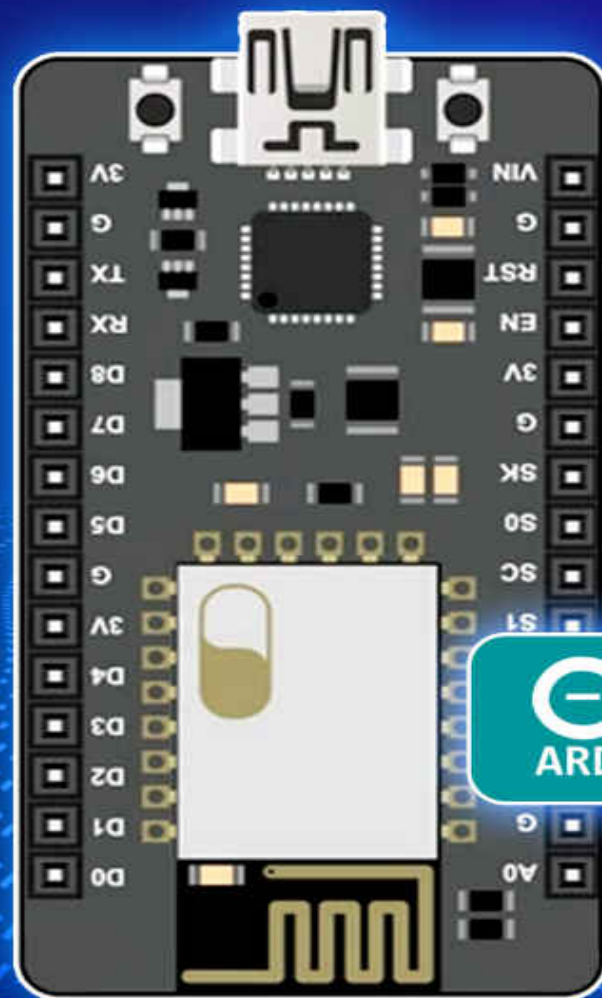


LEARN ESP32 PROJECTS WITH ARDUINO

ESP32-Cam, Robot, 360 Degree Surveillance Camera,
Bluetooth Fundamentals, Arduino Makers, BLE Mesh



LEARN ESP32 PROJECTS WITH ARDUINO

**ESP32-Cam, Robot, 360 Degree Surveillance
Camera, Bluetooth Fundamentals, Arduino
Makers, BLE Mesh**

**By
Jansa Selvam**

TABLE OF CONTENTS

[WHAT IS ESP32](#)

[BLUETOOTH LOW ENERGY](#)

[ADVANTAGES AND LIMITATIONS OF BLE](#)

[ROLE OF BLE IN THE WIRELESS DOMAIN](#)

[OVERALL ARCHITECTURE OF BLE](#)

[APPLICATION, HOST, AND CONTROLLER](#)

[UNDERSTANDING THE BLE STACK](#)

[CONNECTIONS, AND SERVICES IN BLE](#)

[CHARACTERISTICS, PROFILES, AND ATTRIBUTES IN BLE](#)

[SECURITY IN BLE PART 1 PAIRING AND BONDING](#)

[SECURITY IN BLE PART 2 SECURITY KEYS, MODES AND LEVELS](#)

[WHAT IS BLE MESH](#)

[FUNDAMENTAL ARCHITECTURE OF BLE MESH](#)

[BLE MESH CORE CONCEPTS](#)

[BLE MESH PROVISIONING PROCESS](#)

[SECURITY IN BLE MESH](#)

[BLE THROUGHPUT MEASURING IMPLEMENTATION USING ESP32](#)

[BLE MESH NODE, PROVISIONER _ CONSOLE IMPLEMENTATION](#)

[BLE MESH WI-FI COEXISTENCE IMPLEMENTATION](#)

[ARCHITECTURE OVERVIEW](#)

[IMPLEMENTATION](#)

[KNOWING ABOUT THE HARDWARE ESP32-CAM](#)

[UNDERSTANDING THE SCHEMATIC DESIGN OF ESP32](#)

[HARDWARE REQUIRED FOR THE COURSE](#)

[SETTING UP THE HARDWARE](#)

[FLASH YOUR FIRST APPLICATION ON TO ESP32](#)

[UNDERSTANDING DEFAULT APPLICATION](#)

[ESP32-CAM AS ACCESSPOINT](#)

[ESP32-CAM AS STATION WITH STATIC IP](#)

[WRITING YOUR OWN STREAMING WEBSERVER](#)

[USING ANYCAMIO](#)

[USING FFMPEG](#)

[INTRODUCTION TO EEPROM AND MEMORY CARDS ON ESP32-CAM](#)

[PROGRAMMING ESP32-CAM TO ACCESS EEPROM AND MMC](#)

[CREATING TIME LAPSE VIDEO USING ESP32-CAM WITH THE HELP OF SD-MMC](#)

[INTRODUCTION TO PWM FOR LED BRIGHTNESS CONTROL](#)

[BINDING LED CONTROL ACTIONS TO HTML BUTTONS](#)

[DEMONSTRATION](#)

[DIY ROBOT CAR](#)

[PROGRAMMING ESP32-CAM TO CONTROL DIY CAR OFFLINE](#)

[DEMO AND CAR ASSEMBLY BRIEFED](#)

[PROGRAMMING ESP32-CAM TO CONTROL THE DIY CAR OVER IP](#)

[DEMONSTRATION](#)

[ROBOT CAR CONTROL - ESP32 CAM AS ACCESS POINT](#)

[ASSEMBLING PAN AND TILT SYSTEM](#)

[CODING](#)

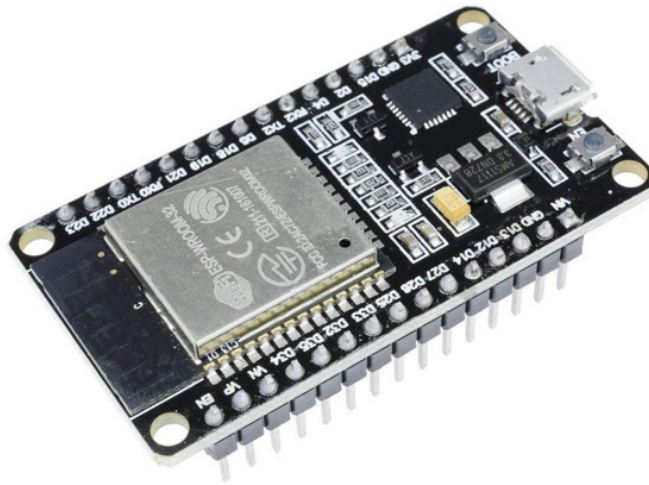
[CODING](#)

[DEMONSTRATION](#)

[INTRODUCTION AND CODING](#)

WHAT IS ESP32

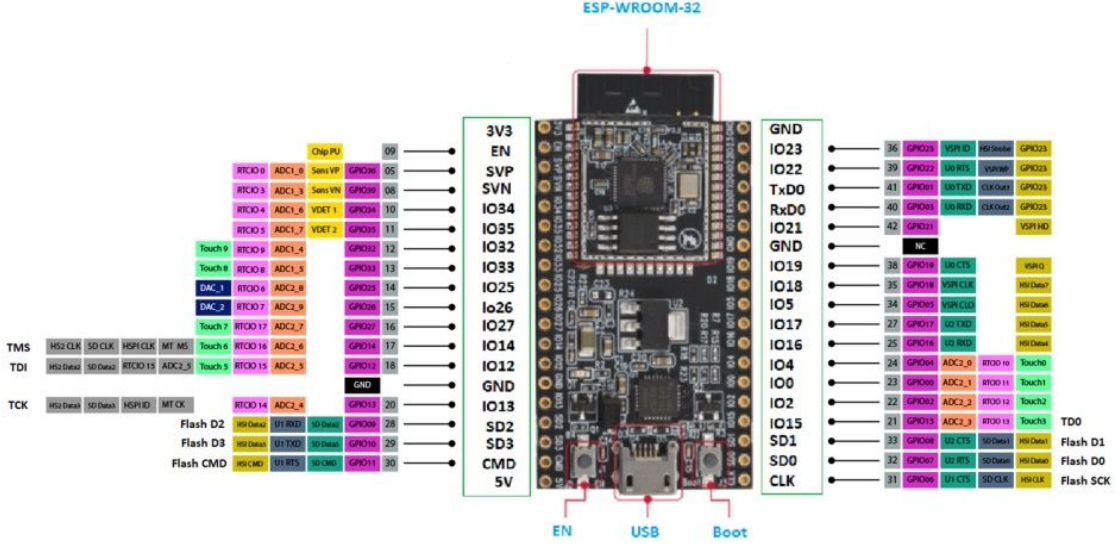
If you are into building your own electronic devices with the help of the microcontrollers, for example, with at megabase or or STM 32, and you still do not know what ESP, ESP 32 east, Please bear in mind that the topic of the ESP 32 is maybe not the freshest topic in the ward because the board itself, the chip set itself appeared on the market four years ago, because somewhere around 2016, but the amount of the possibilities in the development. Put into this board and how cheap and available they are and which incredible functions they really do offer is just like almost mind-blowing to be honest, I always liked to create me something with the Arduino and he goes, well, it's cheap. It's there. It's it's simple. Add that there was a lot of libraries. Well, however, one at one point I discovered that the old ATmega Arduinos clause okay.



Clause of the hardware. Well, they really kind of suck, not too many serial ports, not too many flash, not too much Ram. Everything is so bloody complicated. When you try to build something more competition. Requiring from, from the board. And if you, for example, want to connect this to the internet somehow, then the problem starts to amplify ears. I know. for example, now a day from the earth, you know, you can buys those new fancy with the S a M D architecture, but they are kind of like expensive,

expensive. And if you compare the expensiveness of those boards with the fact that you can get the ESP therapy, Whereas the wifi, With the wifi, with the Bluetooth. And for example, with the installed, LoRa module running at 868 megahertz for really long wrench communication for around 10 bucks, maybe 15, if you add the OAD to it, then it kind of. You'll see. There is a difference. And with ESP, you do not only get the internet connectivity out of the box. You also get a goodies, like for example, three serial ports out of nowhere for SPI. Am I correct? For SBI buses, you can get, two ITC buses. You can get the, as the current driver out of. Nowhere 16 PWM outputs. Yes, they are software PWM, PWM outputs, and many, many, many other cool features. Like for example, 36 GPA ELLs, not every single one can be used as an output. And the input about this as a completely different, a different topic and all of that for like three bucks, you don't kinda like really start to wonder maybe this is the real. Yeah, the real way to go forward and to make things even more interesting. All the ESP 30 twos come with 160 megahertz clock.

ESP32 PINOUTS



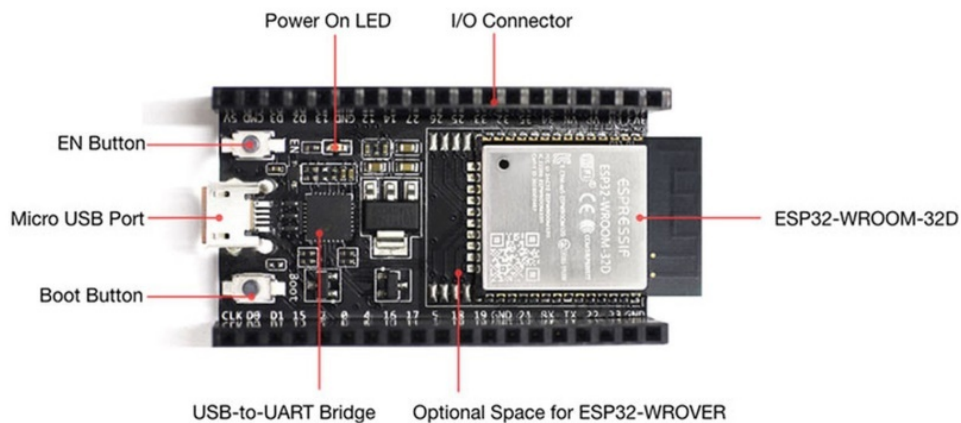
More than 500 kilobytes of Ram and two Cortes. Then they are running the free real time operators. We've took course, and you can have two things running at once on them and everything is there. And by the way, you can program this thing with everything you want, because it will run or do you know? you've got a fresh Arduino not program with Arduino, not the

program are doing about the, are doing the environment. you can do. C just program everything we've seen, just like you wanted, you can use Python and probably know JS and everything. Everything, because the power that lies inside of the ESP cheap stuff is kind of like massive. They are not perfect. They are not perfect because for example, majority of them lacks any flash memory. So if you want to have a place to store your cult, you have to have the board and the board has to have external. SBA connected flash memory, but. 3 10, 15 bucks. Okay. 20, ah, probably the most expensive ESP 32 board I found is for like 25 bucks, but this one has the power management USP, 32 Bluetooth wifi of course, or led display and LoRa module, and also a GPS connected and, liti on 18, 650. In there.



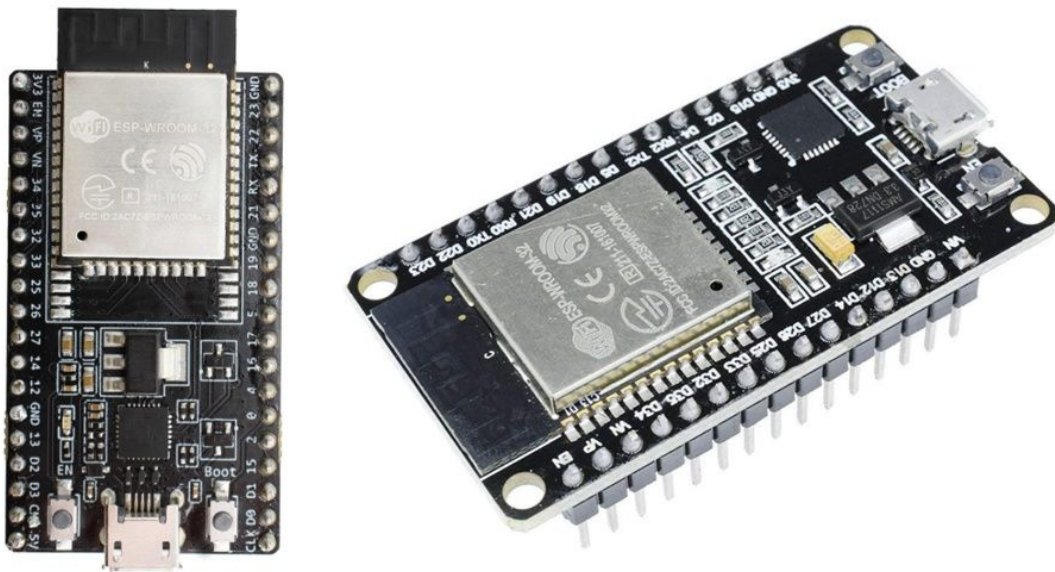
So I'm like, come on. It's like almost a full-blown device that you can do whatever you want with this. And that Skype is the link. I will show some of the examples of the boards that I. Here not everything because for example, I do not have on me the example of the simplest one without the only display, I have no idea where I used all of them. Probably every time I just need something running a microcontroller. I just put the ESP 32 over there. And from my stack of probably five or six, I got zero. And I have no idea where the rest they don't. Maybe they. Who knows. And then DM and in the future, there also be a video of how to start your adventure with ESP 32 Arduino environment. And this is really cool. Okay. So, let's see what I

do have on my work bench today. Let's begin our adventure with ESP 32 boards. What I have over here is probably the most interesting, Value for price board that the Reese right now on the market, which is the ESP 32 in the 30 8:00 PM. Very Sean, which has USB micro was be connected on my site. Two buttons. One of which is, and then the second one can be used just as the button on the board or led display. And on the bottom side, we can see there is the wifi and turn up. There is the flash memory. There is a voltage regulation for everything. It accepts five volts and this very interesting connector over here. Thanks to disconnect, or you can connect any lipo or lithium battery sink one S 3.7 volts that just fit somewhere here and have this think running as the battery operated device, because this thing has a charging unit. Every time you plug in the USB port, it will just charge the battery and then allow you to use this thing.



Oh, without being black for everyone, there is also very similar boards to this one, which is lacking the led display. But if you want to have this thing kind of like interacting with the user, this is rather way to go. Of course you can do. I'll only display to the very Shaun room 32, I think without early D that's the price difference is not that high. not that big. And if you get yourself feeling. ESP 30 twos and external outlets. You can just as well get one with the integrated bear in mind that the way of working with the early deacon, different between boards, for example, some of them require you to

keep a pin, digital, IO 16 high to, for the, for this to work, some are, flipped and you kind of sometimes need a special versions of the library, but. It works. It also has enough flash enough memory and enough computational power to really run a smaller web server without any problems you can connect this thing to the internet. There were certainly plenty of examples, how to use wifi integrated on it, make the request to a server, get the data, or just act as the server to have this thing really working together with your smartphone. You just either connect to the same network or. Make access point of the ESP and everything is fine. And then if this is not enough for you, then if you want some kind of the connectivity between two ESPs for a longer range that rugged, longer ranges than the wifi or Bluetooth allows you to do, there is this version and something like that. There are of course more variants, but those are the, basically this thing. Laura integrated here on the bottom site over here, you can find the, some tech. 1278 or six. I never remember which exactly. And of course the antenna, on, on the pig tail LoRa connectivity.



So you can talk really for like incredible distance as few kilometers, at least if there is, at least some elevation between the unintended us and have. Spread the network of devices, doing incredible stuff, access to the Laura mash, et cetera, et cetera, et cetera, really sky is the limit. I'm currently using a lot of those devices for my project that uses NTSP 32. And Laura, I

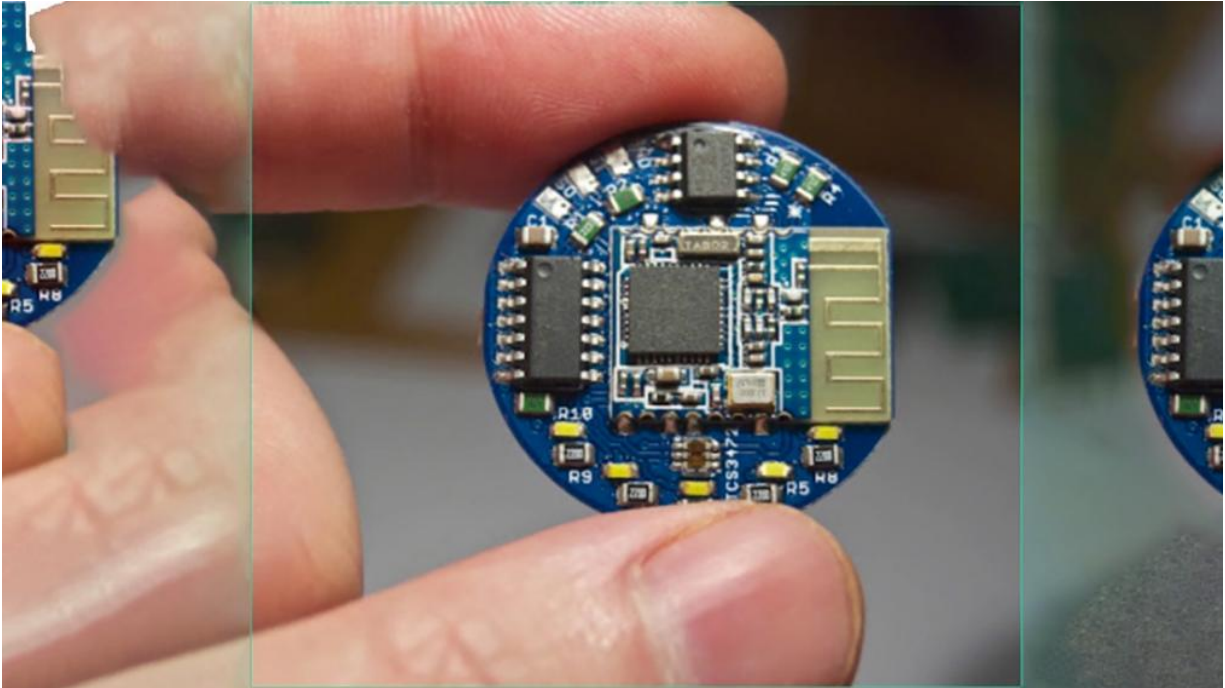
was using this for the location because I was using this also for the long green shred. Your system really sky's the limit there. You can also get very shows running on the. 33 megahertz. And I bet that quite soon, there also will be the versions with the LoRa module running on 2.4 gigahertz that will more or less allow you to build your own version of immersion or see ghost or fly sky FRM three. There are also kind of more advanced versions because for example, what I have over here is the ESP 32, again, but equipped with the color LCD. If I plug the battery in, that's what, then you see, it's a color LCD. I hope it will be visible in the, is it visible? Yeah, it's a color LCD. So if you only want to use. Mainly the color display. And this one, for example, should be out of the box programmed with. But this is not, of course not the only, only choice you have. There, there are also the versions of the ESP 32. I do not have at hand. The first one was this phone factor about not sorry, this form factor, but without the early display, there is also much smaller, very strong call to D one or something like that, which has much less and you cannot connect everything. Connections are just not the broken out. However, those things has really like super simple, super cheap and small, and you can put every, there, there are also variations with the paper displays. There are versions with the Laura or early D DPS power distribution system and a lead to your holder. One more time. Nothing can me, but I do own two of those. And for 25 bucks, Almost everything, everything you want. If you want to know what you can do with ESP through deduce and what kinds of incredible hardware you can get, then AliExpress really has everything you really type ESP 32. And you only pick. And speaking of which I will probably have to make some purchases right now, because like I said, I'm out of few versions of. Those sports very mind. The quality of them kind of sometimes can be disappointing. for example, all the versions with the early installed the early deep cracks very easily, and they have like a small. For three without the old, because they just cracked, it was not working. And I think that's all for today. I hope I got your interest in the ESP still T2. And when you will be thinking about the hardware to use in your next electronic project, you will think about that stuff because it's really amazing for the price.

BLUETOOTH LOW ENERGY

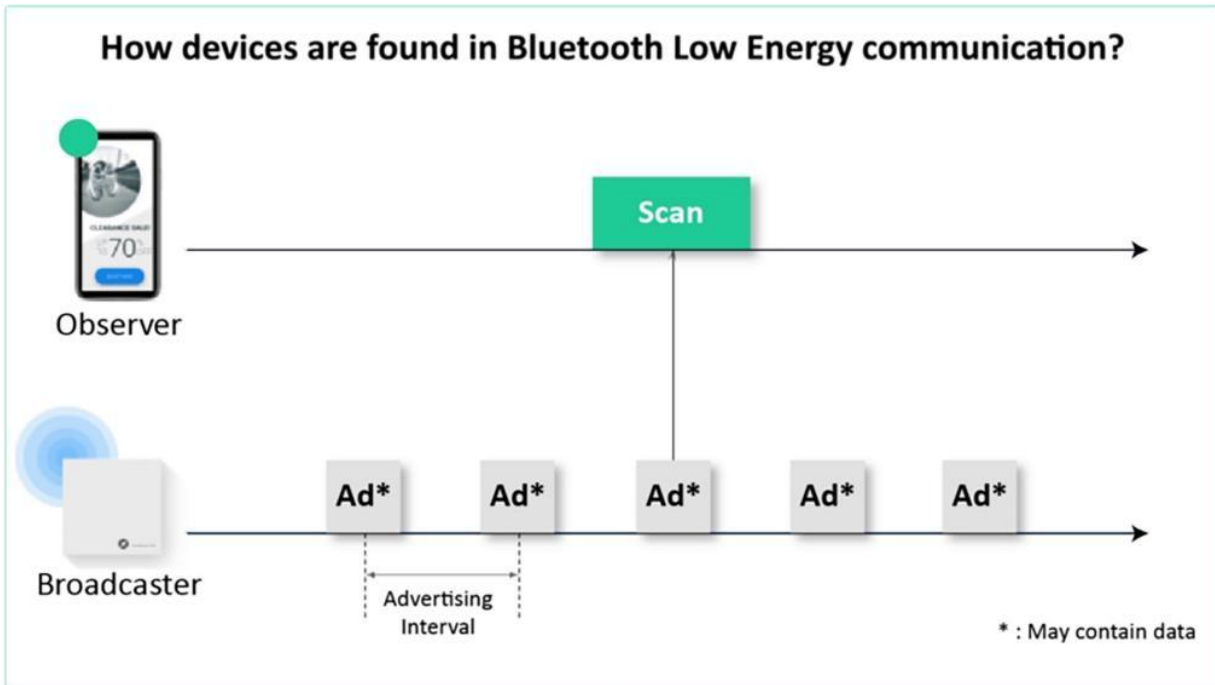
You have heard of Bluetooth, right?! I guess by now, almost every person on the earth is familiar with what Bluetooth is, right? For example, you use Bluetooth, also known as Bluetooth Classic, when using a wireless headset to listen to music on your smartphone. Yes, that is an application of Bluetooth. listen to music on your smartphone. Yes, that is an application of Bluetooth. As you are a beginner, you can see BLE as a type of Bluetooth that uses less energy. As you are a beginner, you can see BLE as a type of Bluetooth that uses less energy. It is an efficient form of Bluetooth that has been developed specifically for the "Internet of Things."

Before we look at what is the Role of BLE in IoT, let's have a look at what IoT is. The Internet of Things is the connection of devices to other devices and to the Internet to help them communicate and share data across large wireless networks. BLE's energy efficiency has made it a preferred option for IoT. Also, its low data rate makes it extremely suitable for use in cases where only state data has to be transferred, such as sensors. suitable for use in cases where only state data has to be transferred, such as sensors.

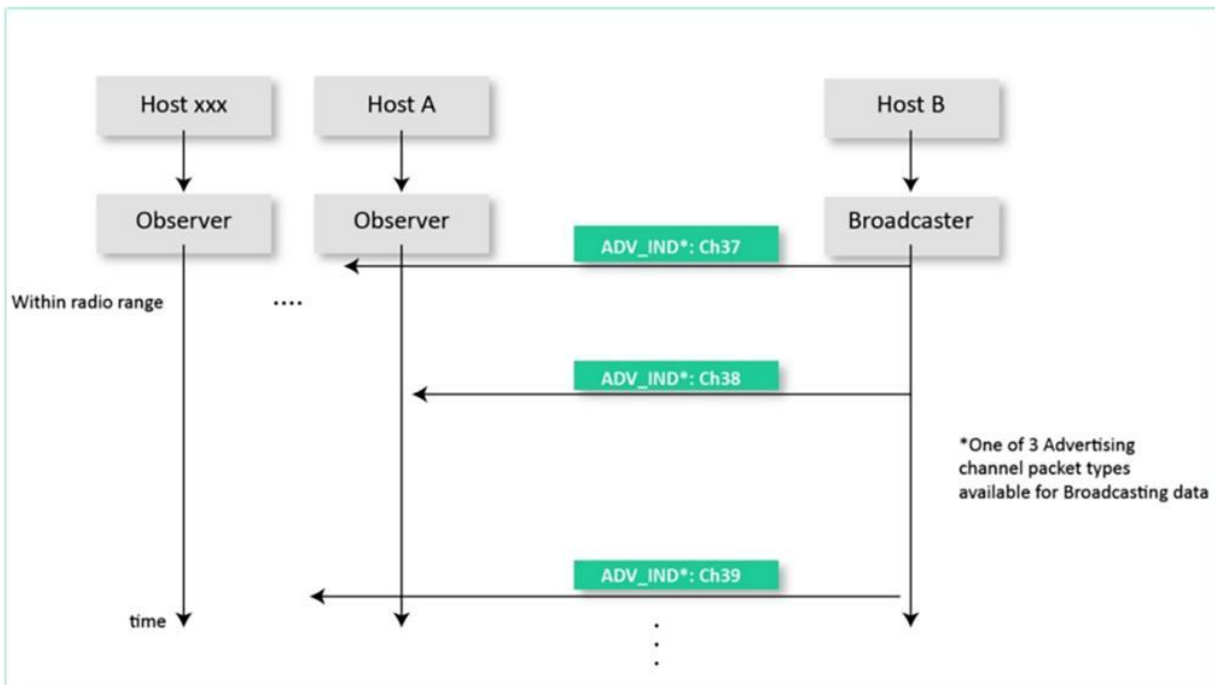
What makes BLE so unique is that it can communicate with a large number of mobile devices found today, like phones which run Android, OS X, iOS, and BlackBerry, as well as Linux, and Windows 10 all support BLE. and Windows 10 all support BLE. This means that you can integrate your project easily to make a multi-platformed communication.



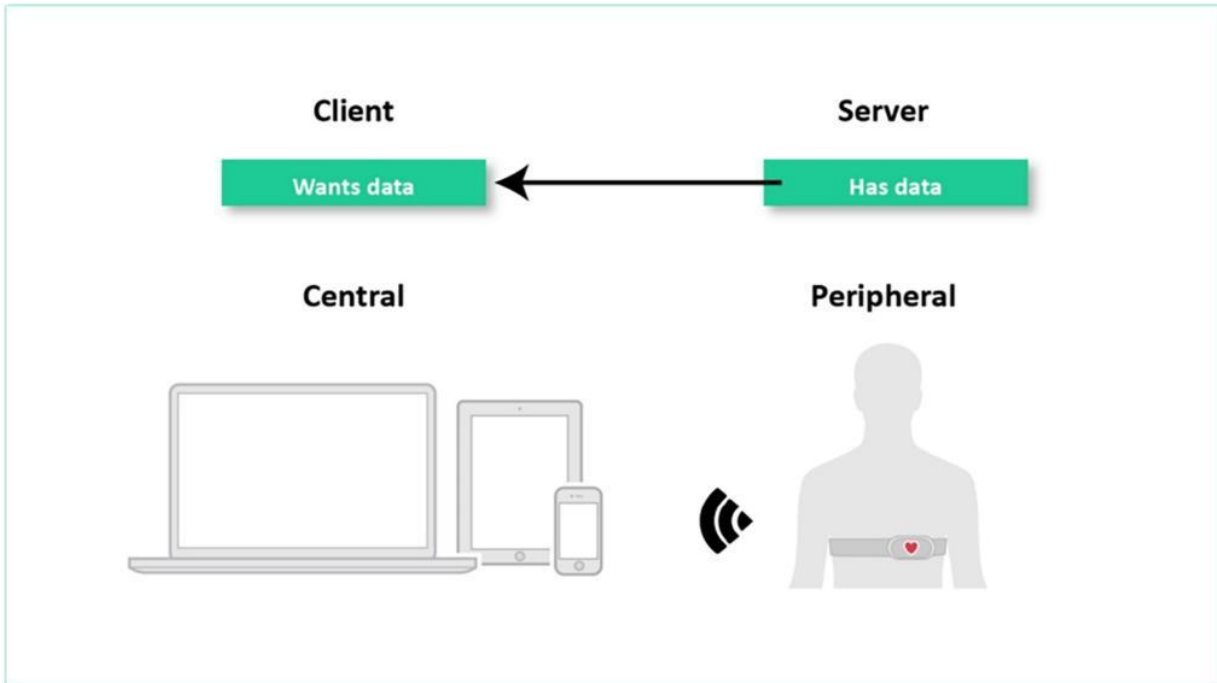
So imagine this: a radio module that is fit for any data that is needing to be sent while being energy-efficient and connectable to all major devices on the market. It is easy to see why BLE is the choice for low-energy radio communication for most of today's engineers. It is easy to see why BLE is the choice for low-energy radio communication for most of today's engineers. We will now look at an overview of how BLE works. We will get in-depth into this in later project. A BLE device can talk to other devices in one of two ways: Either by Broadcasting or by Connections.



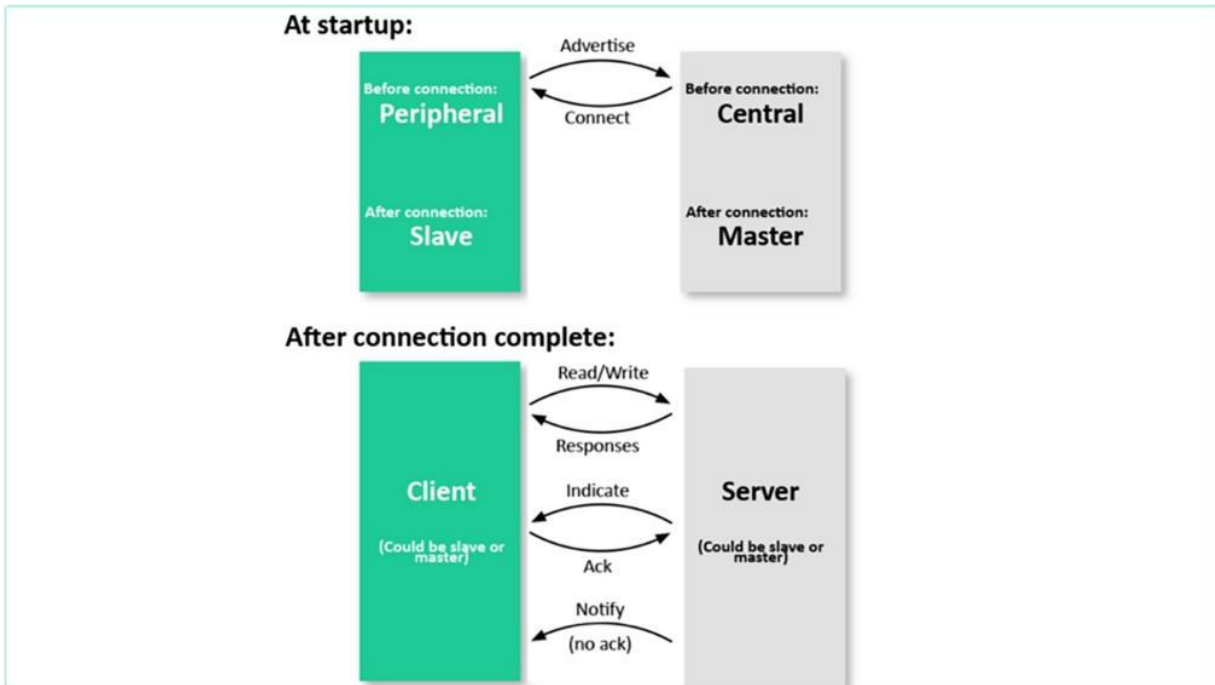
Broadcasting is the process of sending data out to all the listening devices. When talking about Broadcasting, we need to define two device roles: Broadcaster and Observer. The Broadcaster sends advertising packets periodically to anyone who is willing to listen. observer repeatedly scans the area in order to receive the advertising packets.



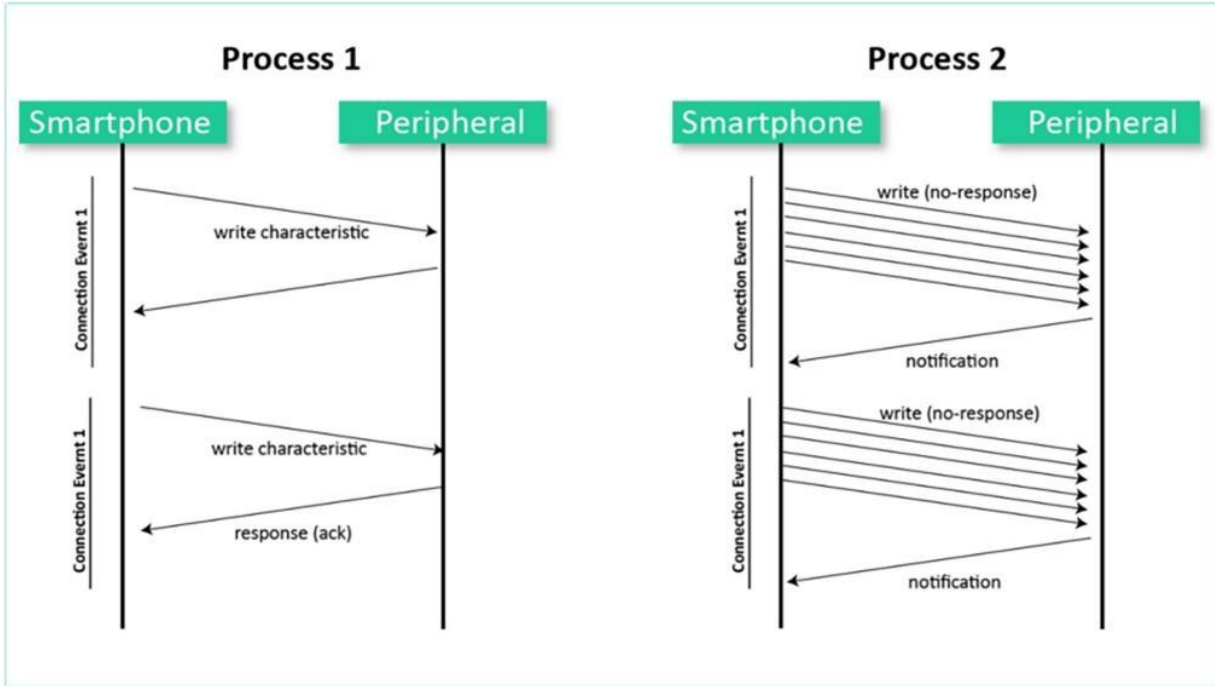
While an Observer repeatedly scans the area in order to receive the advertising packets. Now, when the Observer receives the Advertising packet, it can request the Scan Response Data. It is important to note that in BLE, Broadcasting is the only way a device can transmit data to more than one device at a time.



Now let's look at the second way of communication. A Connection in BLE is defined as a permanent, periodical data exchange of packets between two devices. The master, also known as the central device, scans the frequencies for advertising packets and initiates a connection.



Once the connection is established, the central device manages the timing and initiates the periodical data exchanges. The slave, also known as the peripheral device, sends advertising packets periodically and accepts incoming. The slave, also known as the peripheral device, sends advertising packets periodically and accepts incoming the peripheral follows the central timings and exchanges data regularly with it then connected.



Thus two devices can power up, exchange data, and then go to sleep until the next connection event, which is one of the reasons for BLE's energy efficiency. Thus two devices can power up, exchange data, and then go to sleep until the next connection event, which is one of the reasons for BLE's energy efficiency. Thus two devices can power up, exchange data, and then go to sleep until the next connection event, which is one of the reasons for BLE's energy efficiency. Thus two devices can power up, exchange data, and then go to sleep until the next connection event, which is one of the reasons for BLE's energy efficiency. Now let's look at some technical facts about BLE. BLE is not compatible with Bluetooth, although BLE versions are backward compatible with each other BLE has a typical range of 10 to 30 meters The maximum data rate supported as per Bluetooth Version 5 is 2 Mbps BLE has a typical range of 10 to 30 meters The maximum data rate supported as per Bluetooth Version 5 is 2 Mbps BLE Occupies the 2.400-2.4835 GHz Frequency spectrum. It is segmented into 40 2MHz wide channels. BLE power consumption can vary based on the application, and The peak current consumption of a BLE chipset during radio transmission is typically under 15 mA. BLE Occupies the 2.400-2.4835 GHz Frequency spectrum. It is segmented into 40 2MHz wide channels. BLE power consumption can vary based on the application, and The peak current consumption of a BLE chipset during radio transmission is typically under 15 mA. BLE Occupies the 2.400-2.4835 GHz Frequency spectrum. It is segmented into 40 2MHz wide channels. BLE power consumption can vary based on the application, and The peak current consumption of a BLE chipset during radio transmission is typically under 15 mA. BLE Occupies the 2.400-2.4835 GHz Frequency spectrum. It is segmented into 40 2MHz wide channels. BLE power consumption can vary based on the application, and The peak current consumption of a BLE chipset during radio transmission is typically under 15 mA. Security is optional in BLE Communication, and it is up to the developers to implement it. For all encryption operations, BLE uses AES CCM with a 128-bit key. For all encryption operations, BLE uses AES CCM with a 128-bit key.

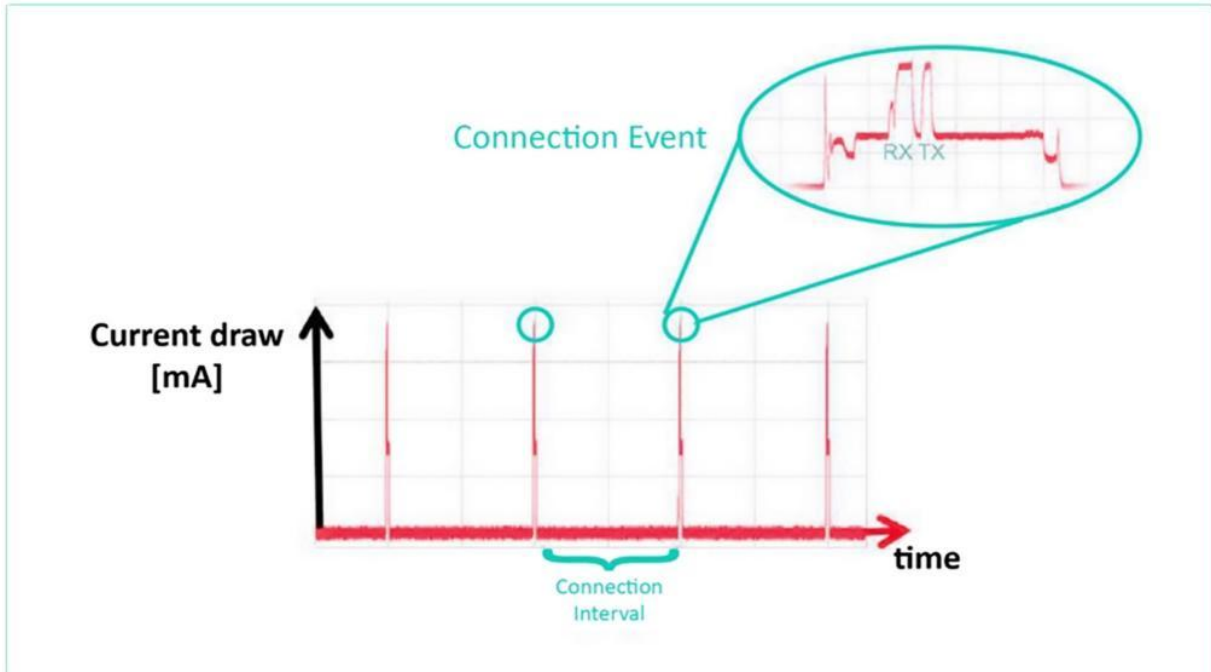
Bluetooth Classic	BLE
Used for streaming applications such as Audio streaming, file transfers, and headsets	Used for sensor data, control of devices, And low-bandwidth applications
Not optimized for low power, but has a Higher data rate (3Mbps maximum Compared to 2Mbps for BLE)	Meant for low power, low duty data Cycles
Operates over 79 RF (radio frequency) Channels	Operates over 40 RF channels
Discovery occurs on 32 channels	Discovery occurs on 3 channels, leading to quicker discovery and connections than Bluetooth Classic

Now let's compare Bluetooth Classic with BLE. Bluetooth Classic is used for streaming applications such as audio streaming, file transfers, and headsets. BLE, on the other hand, is used for sensor data, control of devices, and low-bandwidth applications such as audio streaming file transfers and headset's. Bluetooth Classic is used for streaming applications such as audio streaming, file transfers, and headsets. BLE, on the other hand, is used for sensor data, control of devices, and low-bandwidth applications Bluetooth Classic is used for streaming applications such as audio streaming, file transfers, and headsets. BLE, on the other hand, is used for sensor data, control of devices, and low-bandwidth applications Bluetooth Classic is not meant for low power but has a higher data rate. Bluetooth classic has a maximum data rate of 3Mbps as compared to 2Mbps for BLE as BLE is meant for low power, low duty data cycle applications Bluetooth Classic is not meant for low power but has a higher data rate. Bluetooth classic has a maximum data rate of 3Mbps as compared to 2Mbps for BLE as BLE is meant for low power, low duty data cycle applications Bluetooth Classic operates over 79 RF Channels while BLE Operates over 40 RF Channels Bluetooth Classic operates over 79 RF Channels while BLE Operates over 40 RF Channels In Bluetooth Classic, the discovery operation occurs on 32 channels, while on BLE, it occurs on three channels, which leads to quicker discovery and connections than

Bluetooth Classic In Bluetooth Classic, the discovery operation occurs on 32 channels, while on BLE, it occurs on three channels, which leads to quicker discovery and connections than Bluetooth Classic In Bluetooth Classic, the discovery operation occurs on 32 channels, while on BLE, it occurs on three channels, which leads to quicker discovery and connections than Bluetooth Classic.

ADVANTAGES AND LIMITATIONS OF BLE

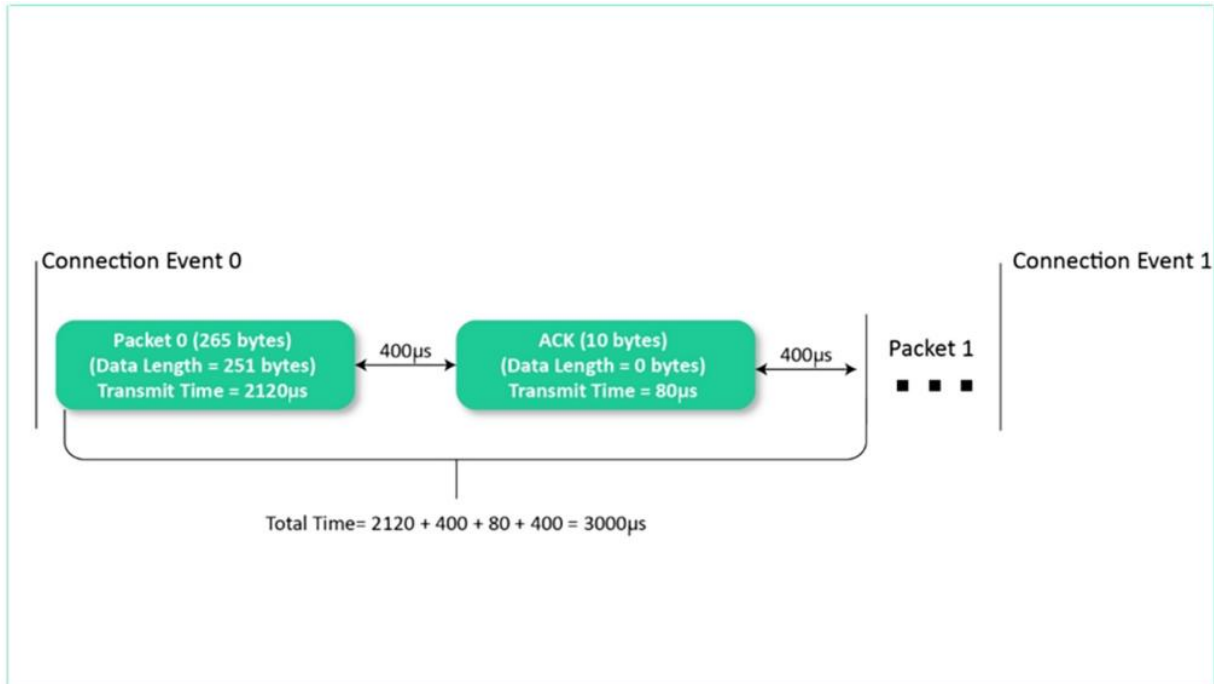
Low power consumption: Even when compared to other low-power technologies, BLE achieves a lower power consumption than its competitors.



It's optimized and less power consumed, by turning the radio off as much as possible, in addition to sending small amounts of data at low data rates. It's optimized and less power consumed, by turning the radio off as much as possible, in addition to sending small amounts of data at low data rates. It's optimized and less power consumed, by turning the radio off as much as possible, in addition to sending small amounts of data at low data rates.

Zero cost to access official specification documents: With most other wireless protocols and technologies, you would have to become a member of the official group or consortium for that standard in order to access the specification. Becoming a member of those groups can cost a significant amount of up to thousands of dollars per year. With BLE, the major version specification documents are available to download from the Bluetooth website for free.

Lower cost of modules and chipsets when compared to other similar technologies.



Real-time operation: A connection between BLE devices A connection between B and C devices can be made, data transferred and the connection shut down in can be made, data transferred, and the connection shut down in just 3ms. This agility is quite unique to BLE. Compatibility is another important advantage of BLE. BLE's existence in almost all smartphones in the market is probably the biggest advantage BLE has over its competitors such as ZigBee, Z-Wave, and Thread.

Now let us move on to some of the limitations of BLE. In my opinion, knowing the limitations of technology is more important than knowing its advantages So without wasting any time, let's dive deep into it. Data Rate: The data Rate of BLE is limited by the physical radio's data rate. This rate depends on the Bluetooth version that is used.

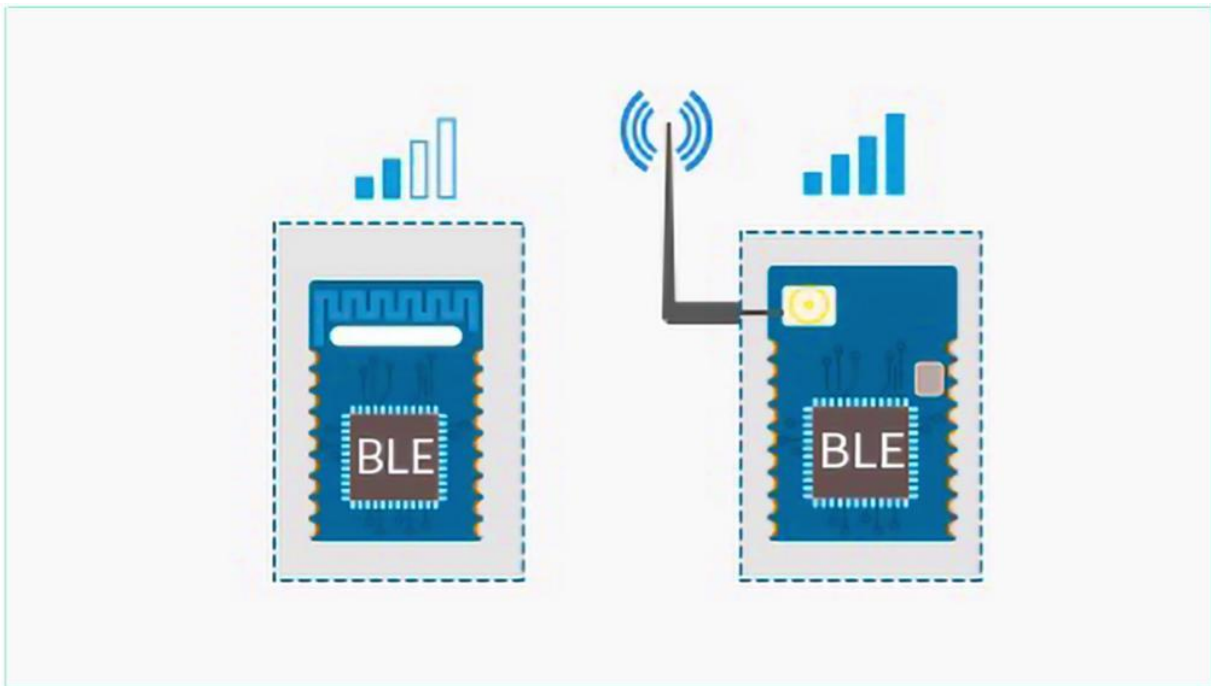
Version	4.0	4.1	4.2	5.0 & 5.1
Multi-Roles	No	Yes		
PDU Payload	Up to 31 bytes		Up to 255 bytes	
LE Secure	No		Yes	
IoT Support	Limited		Medium	High
Advertising Channels	3 Channels			3 Primary Ch. 37 Secondary Ch.
Data Rate	1 Mbps			2 Mbps
Effective Range	50 m (Line of Sight) 10 m (Indoor)		200 m (Line of Sight) 40 m (Indoor)	
Battery Life	Shorter			Longer

For Bluetooth 4.2 and earlier, the rate is 1 Mbps. For Bluetooth 5 and later, the rate varies depending on the mode and stack configuration being used.

PHY	Modulation scheme	Coding scheme		Data rate
		Access Header	Payload	
LE 1M (Current BLE)	1 Ms/s GFSK modulation	Uncoded	Uncoded	1 Mbps
LE2M	2 Ms/s GFSK modulation	Uncoded	Uncoded	2 Mbps
LE Coded	1 Ms/s GFSK modulation	S=8 S=8	S=8 S=2	125 kbps 500 kbps

The rate can be at 1 Mbps like earlier versions or 2 Mbps when utilizing the high-speed feature. When utilizing the long-range feature the rate drops to either 500 or 125 Kbps. Range: BLE is designed for short-range

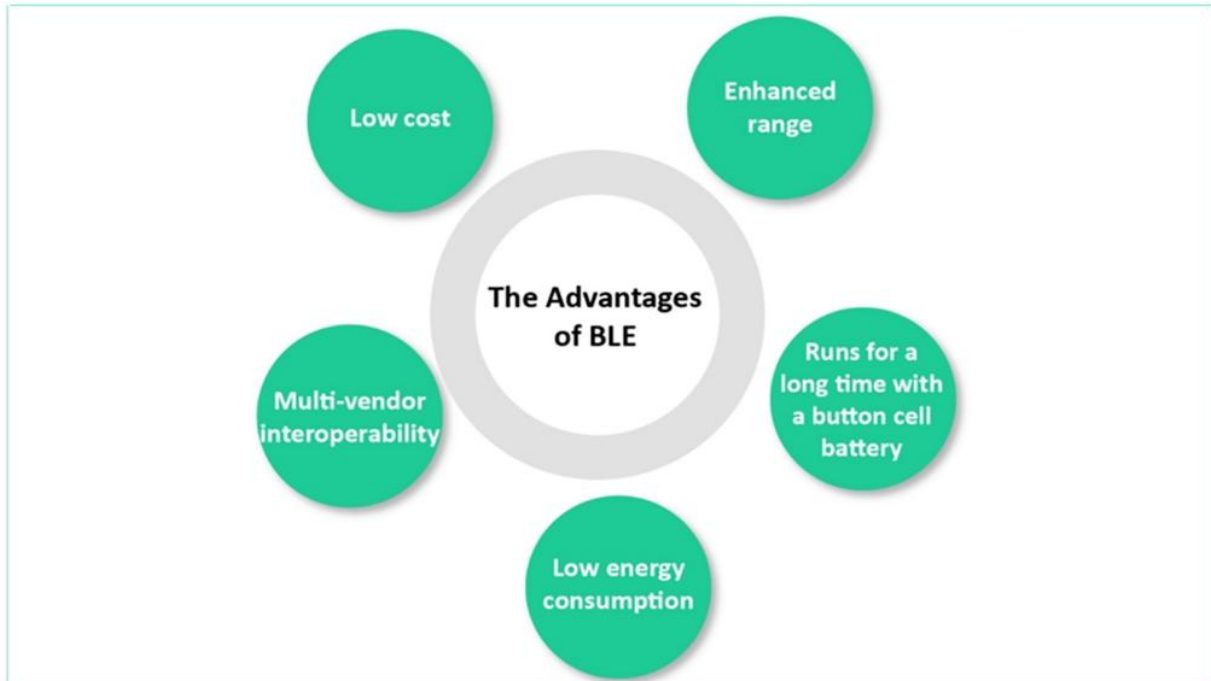
applications, and thus its range of operation is limited to a maximum of 30 meters.



There are a few factors that limit the range of BLE. The range depends upon the Antenna design, Antenna orientation, physical enclosure of the device, and interference due to obstacles. The range depends upon the Antenna design, Antenna orientation, physical enclosure of the device, and interference due to obstacles. Gateway Requirements for Internet Connectivity is another big disadvantage of BLE. In order to transfer data from a BLE only device to the Internet, another BLE device that has an IP connection is needed to receive this data and then, in turn, relay it to the Internet.

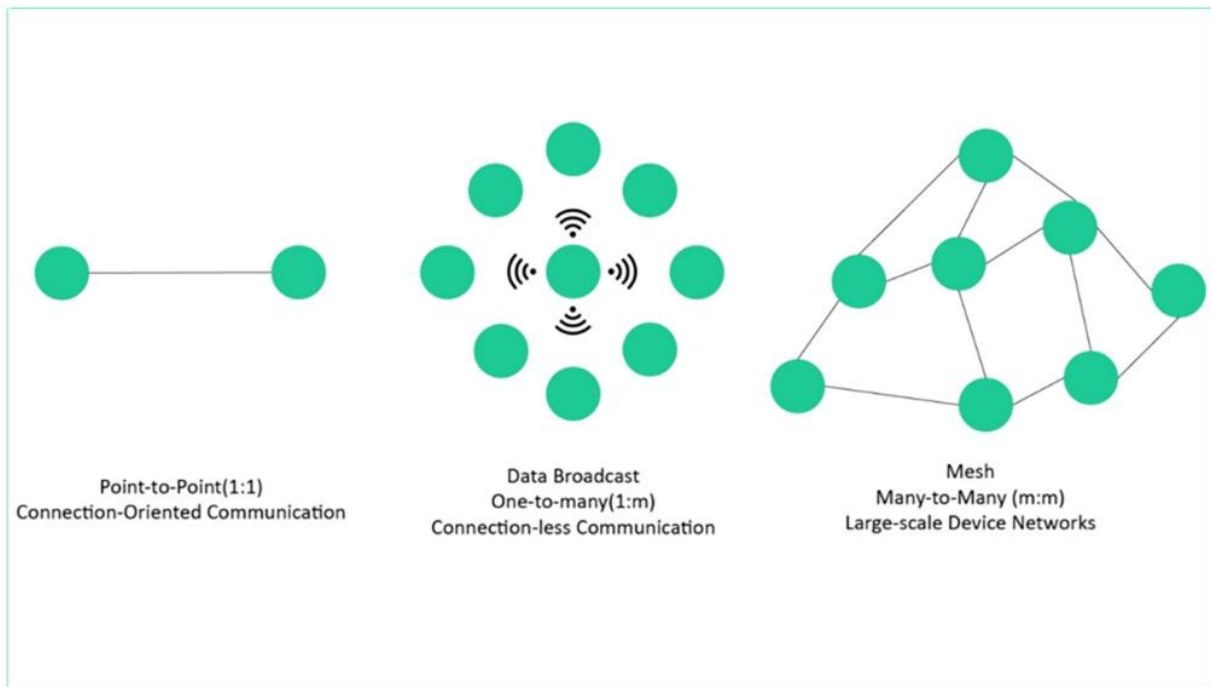
ROLE OF BLE IN THE WIRELESS DOMAIN

IoT being a launchpad for standalone business-driven innovations, it still struggles with interoperability and unification. As we have already seen, Bluetooth Low Energy is one of such unifying standards that have already settled down in the IoT world.



As opposed to the classic Bluetooth, it offers considerably reduced power consumption with a similar communication range, with an approximate real-life maximum of 30 m.

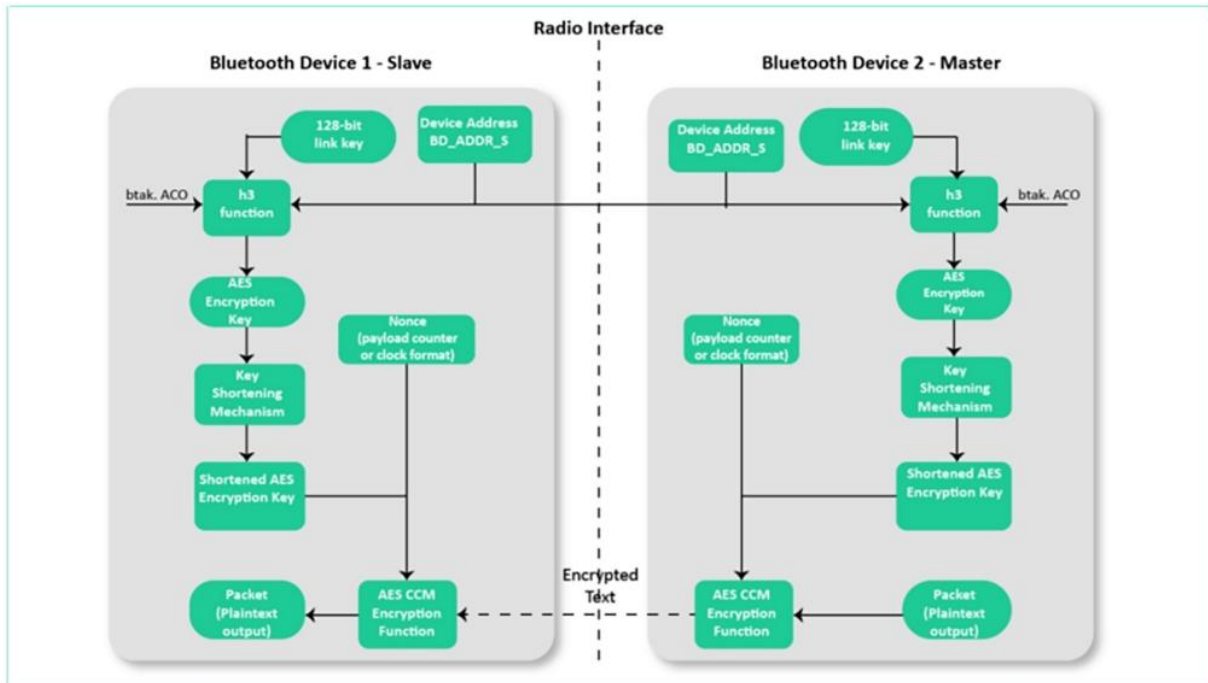
BLE technology offers a variety of possible applications: from wearables and smart homes to healthcare, from connected cars to large-scale enterprise sensor networks and industrial IoT. The advantages of this technology are simplicity, compatibility with other devices, and low cost in both development and support. The advantages of this technology are simplicity, compatibility with other devices, and low cost in both development and support.



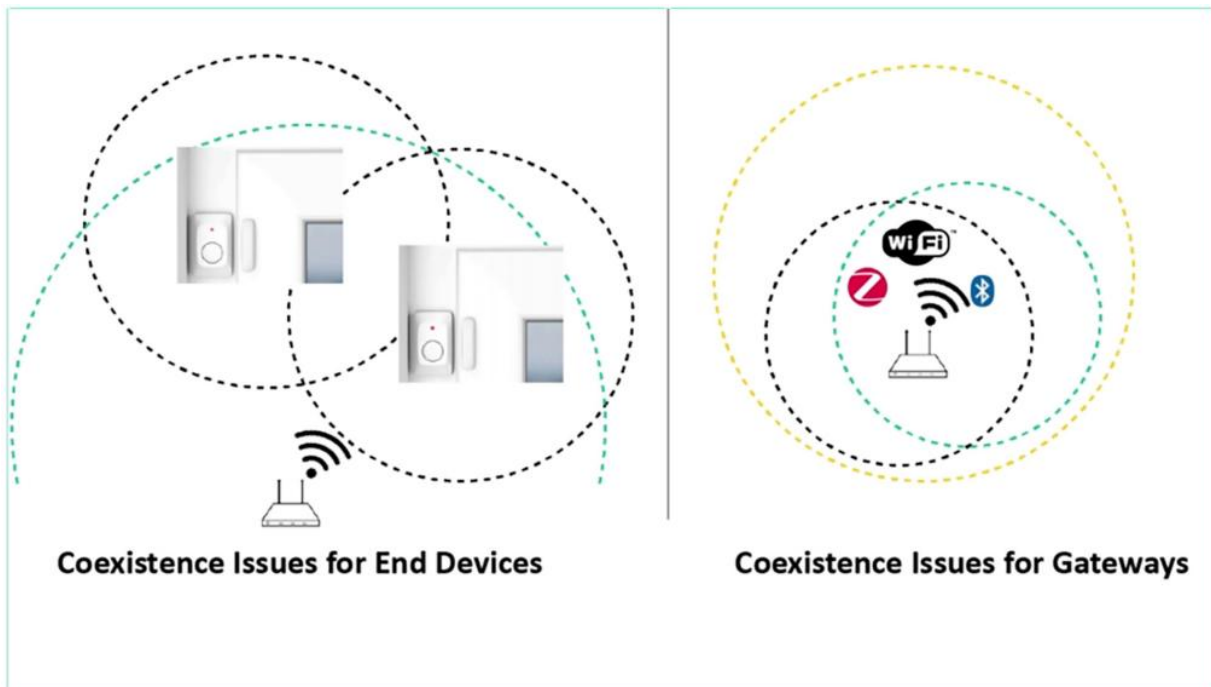
There are 3 types of connection that could be implemented in your next Internet of Things project using BLE. There are 3 types of connection that could be implemented in your next Internet of Things project using BLE. One-to-One is widely used in the health & fitness domain, where low-power consumption devices like fitness trackers and smartwatches are used to track daily activities and send this data to users??? smartphones. One-to-One is widely used in the health & fitness domain, where low-power consumption devices like fitness trackers and smartwatches are used to track daily activities and send this data to users??? smartphones.

One-to-Many is applied in the areas involving item-tracking and way-finding beacons, as well as point-of-interest beacons. Smartphones have become the ultimate devices for the said purposes, where sending small bits of data using little energy is enough to perform the required action. Smartphones have become the ultimate devices for the said purposes, where sending small bits of data using little energy is enough to perform the required action. Many-to-Many presuppose home automation and sensor networks as the most promising areas of use, well-suited for developers and system integrators in the IoT sphere. Many-to-Many presuppose home automation and sensor networks as the most promising areas of use, well-suited for developers and system integrators in the IoT sphere.

For any type of IoT system, perhaps the main issue is security: passive eavesdropping, identity tracking, and man-in-the-middle attacks are a threat if not taken care of.



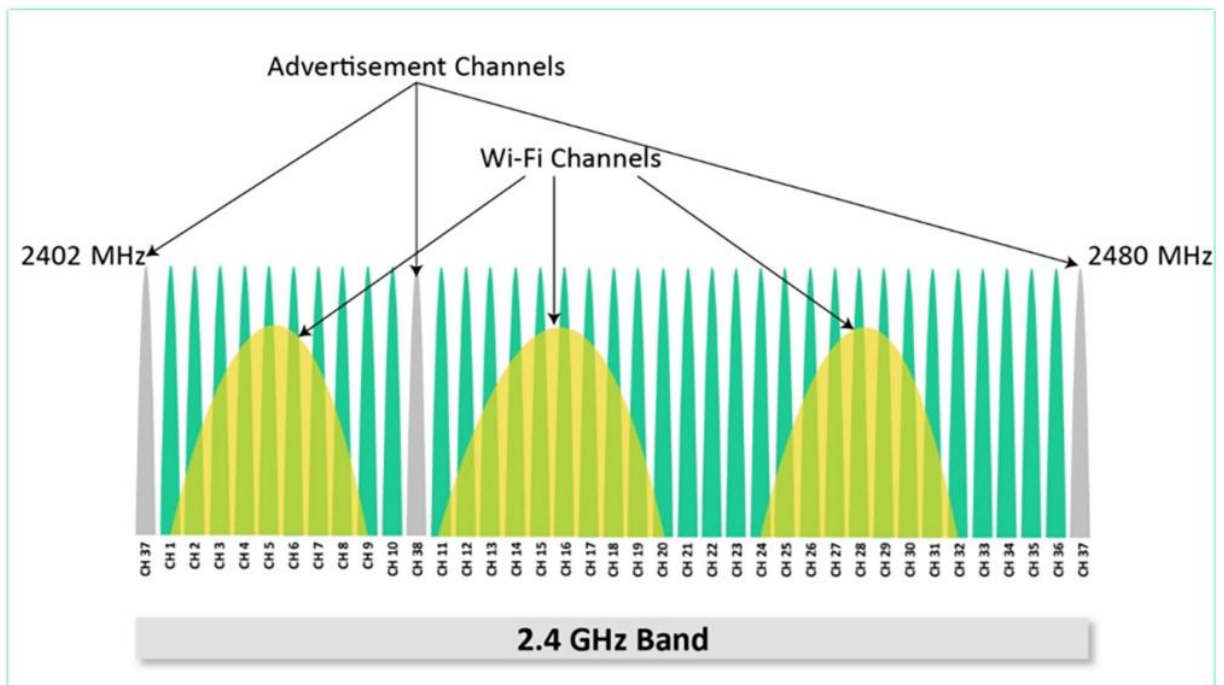
BLE supports AES-CCM cryptography, which works to protect the transferred data and provide an extra level of security. BLE supports AES-CCM cryptography, which works to protect the transferred data and provide an extra level of security. BLE supports AES-CCM cryptography, which works to protect the transferred data and provide an extra level of security.



Now that we have seen how BLE is important for the world of IoT, let's look at its wireless coexistence capabilities with other standards. Wireless coexistence is the ability of one system to perform a task in a given shared environment where other systems performing tasks may, , or may not, be using the same set of rules or standards.

As there are different types of wireless networks operating in the same unlicensed frequency band, it is possible for two different networks to interfere with each other. operating in the same unlicensed frequency band, it is possible for two different networks to interfere with each other. This interference can cause transmission failures in both networks. There is no standardized algorithm to achieve the coexistence of two different wireless networks. This interference can cause transmission failures in both networks. There is no standardized algorithm to achieve the coexistence of two different wireless networks. As we have already covered, BLE works in 2.4- to the 2.483-GHz frequency band. As we have already covered, BLE works in 2.4- to the 2.483-GHz frequency band. Thus let's say there is 2.4 GHz Wi-Fi in the vicinity of the BLE devices, which is often the case in IoT applications Thus let's say there is 2.4 GHz Wi-Fi in the vicinity of the BLE devices, which is often the case in IoT applications In such cases, by common sense, we would assume that there would be a lot of interference.

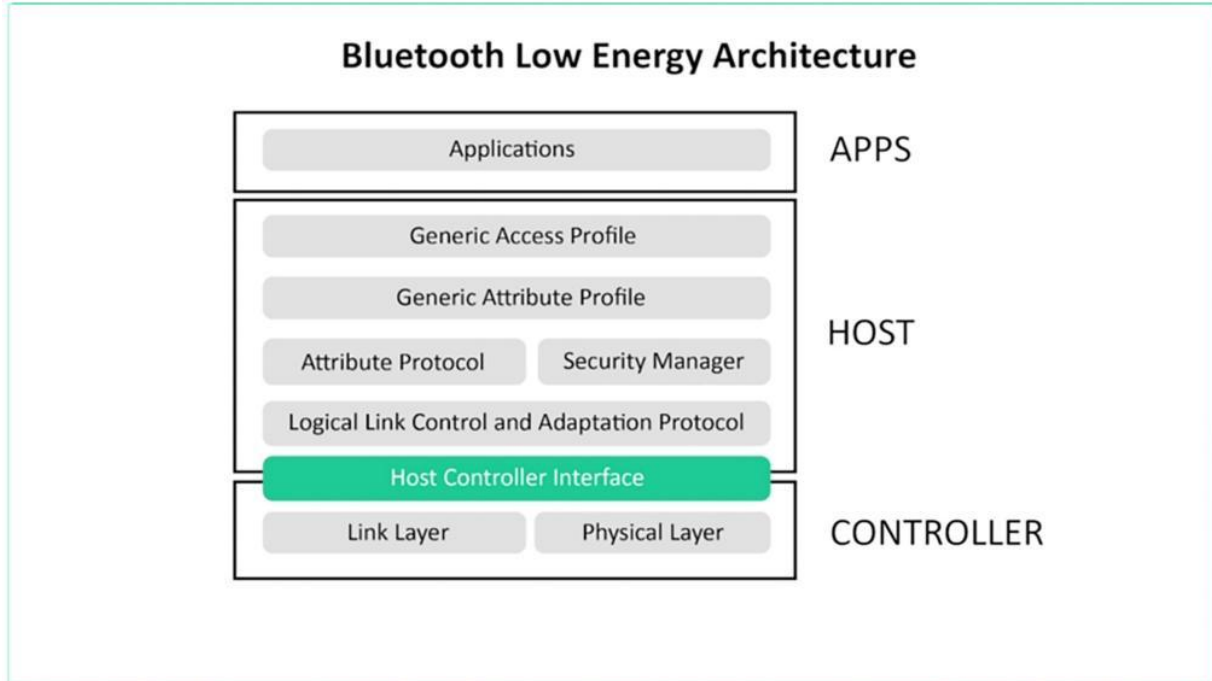
In such cases, by common sense, we would assume that there would be a lot of interference. But this is not the case.



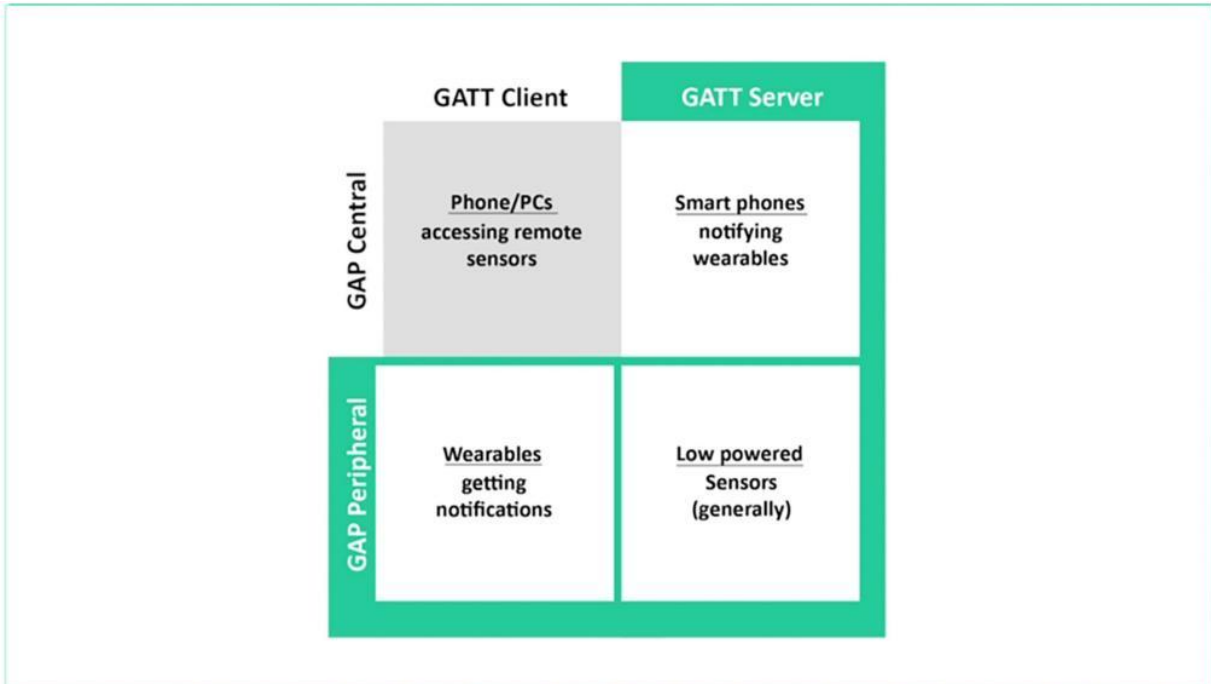
This is because most other devices that interfere with BLE are narrow bands. BLE Advertising channels, in particular, were placed between Wi-Fi channels in such a way so as to avoid the Wi-Fi signal. The wide spacing of the advertisement channels helps BLE better manage the interference from Wi-Fi, Classic Bluetooth, Microwaves, Baby Monitors, etc. to ensure that advertisements succeed. If you want to learn more about the wireless coexistence of BLE with other Wireless standards, If you want to learn more about the wireless coexistence of BLE with other Wireless standards.

OVERALL ARCHITECTURE OF BLE

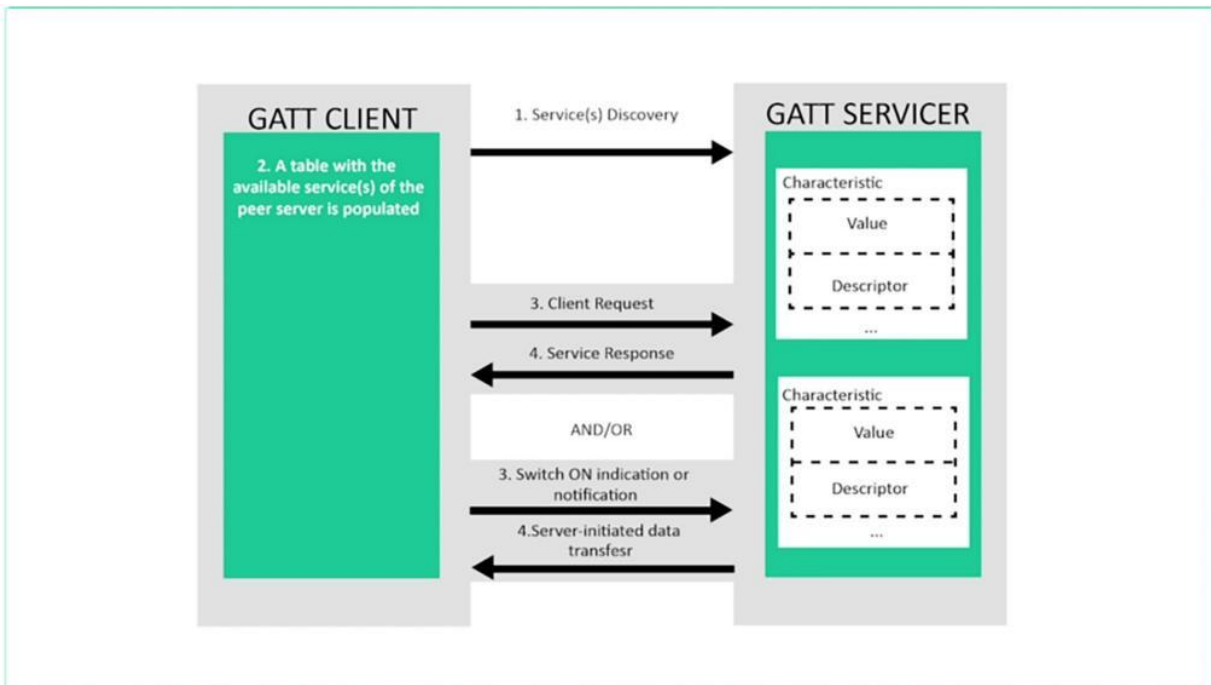
we will cover the following topics Layered Architecture Components of BLE Let us get right into the different layers of the Architecture of BLE.



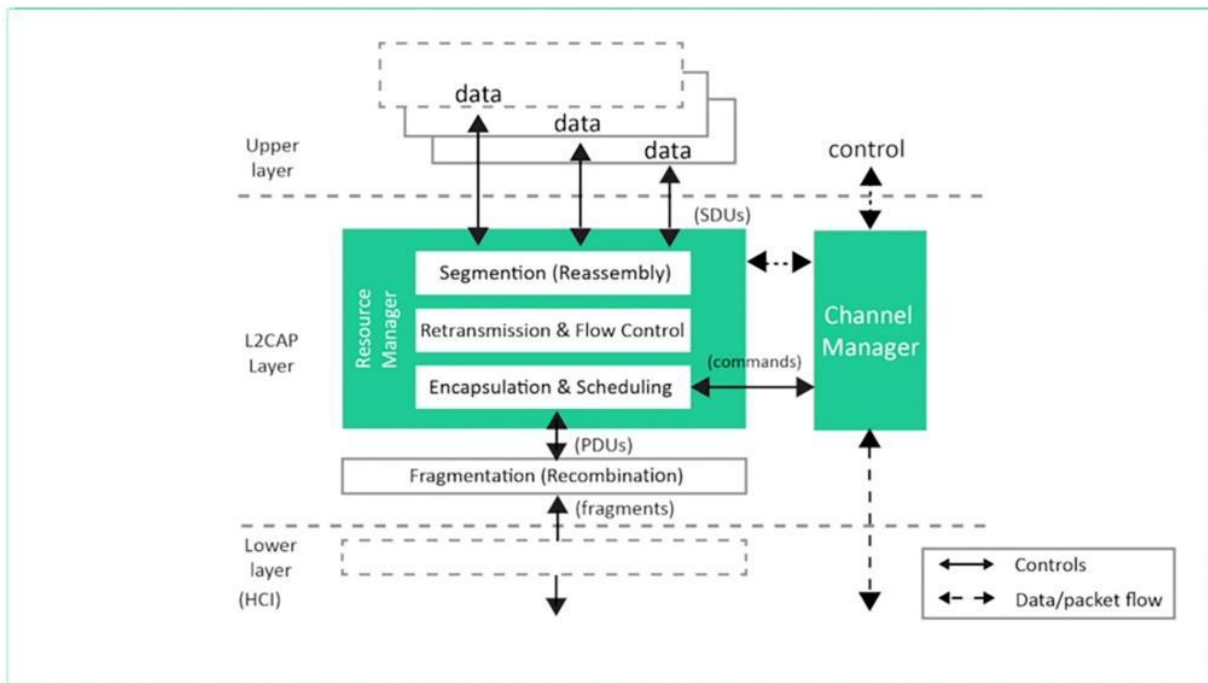
The following figure shows the different layers within the architecture of BLE. The three main blocks in the architecture of a BLE device are: the application, the host, and the controller. The three main blocks in the architecture of a BLE device are: the application, the host, and the controller.



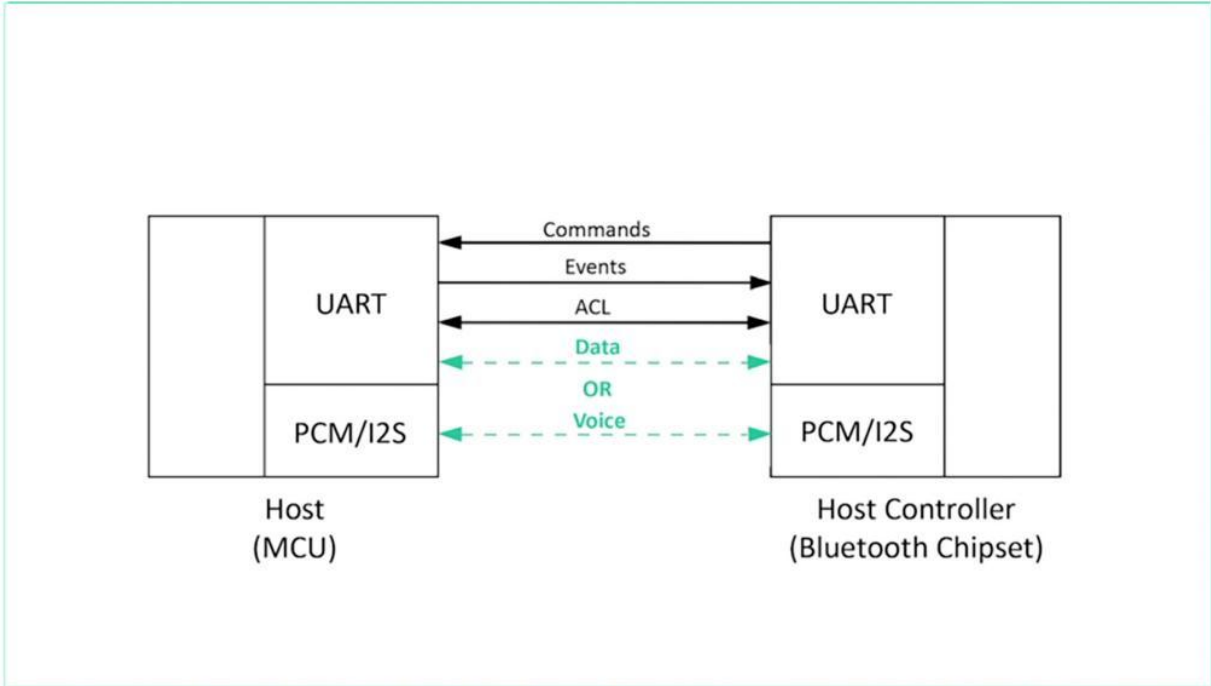
Application layer The application layer is use-case dependent and refers to the implementation on top of the Generic Access Profile and Generic Attribute Profile ??? it??s how your application handles data received from and sent to other devices and the logic behind it



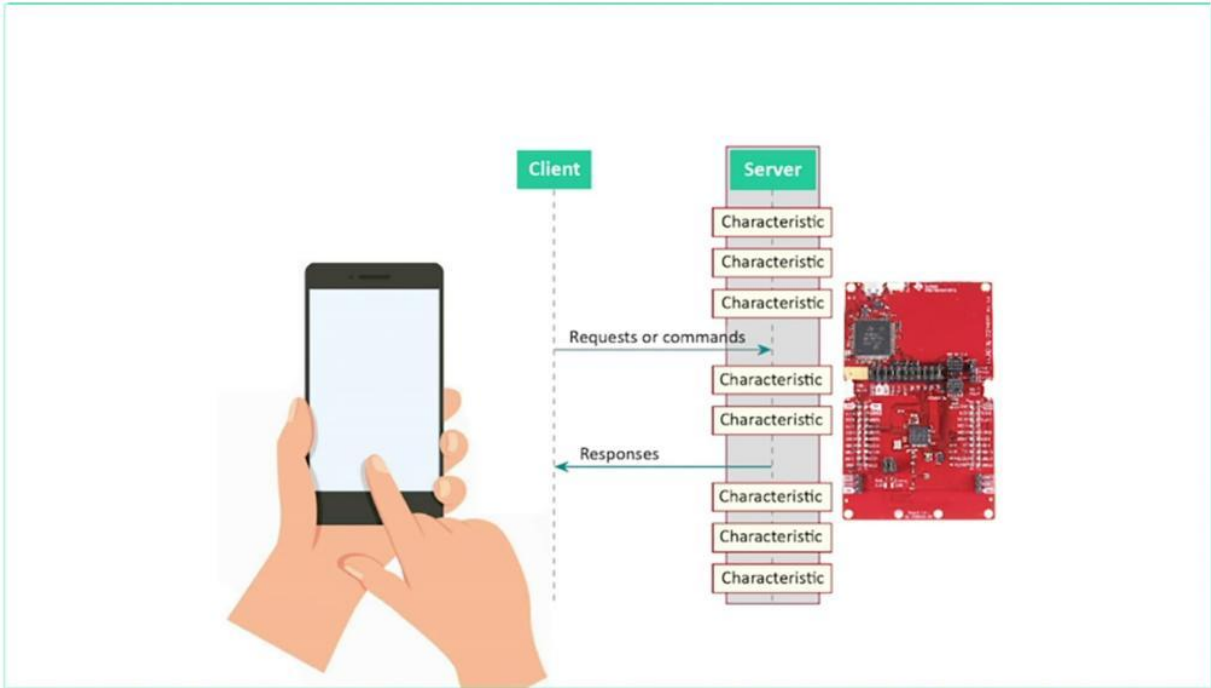
This portion is the code that you would write for your specific BLE application and is generally not part of the BLE stack for the platform which you develop. This portion is the code that you would write for your specific BLE application and is generally not part of the BLE stack for the platform which you develop. Host side Generic Access Profile (GAP) Generic Attribute Profile (GATT) Attribute Protocol (ATT) Security Manager (SM) Logical Link Control and Adaptation Protocol (L2CAP) Host Controller Interface (HCI) ??? Host side Host Layer enables communication between the host and the controller via a serial interface.



Host Layer enables communication between the host and the controller via a serial interface. Since the controller deals with hard real-time requirements and contact with the physical layer, it is good to separate this part from the host. Host mostly responsible for more complex implementations but with less stringent timing.

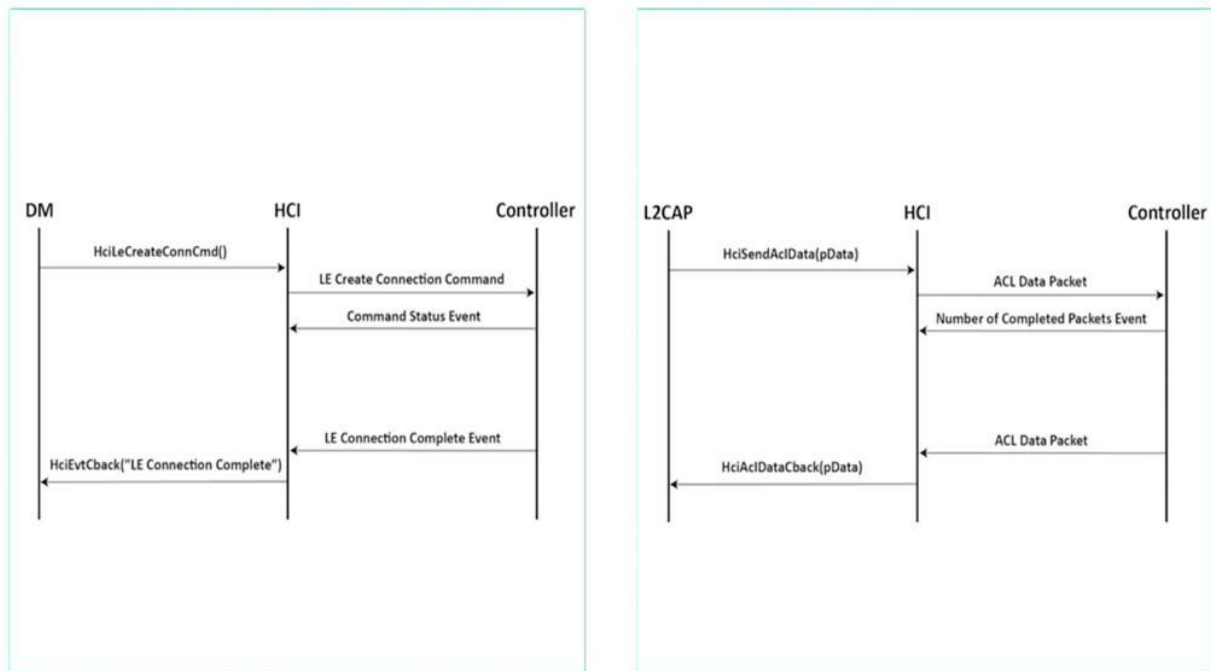


Generally, the host runs in the main CPU, while the controller is mostly about the hardware chip connected via the hardware IO ports.



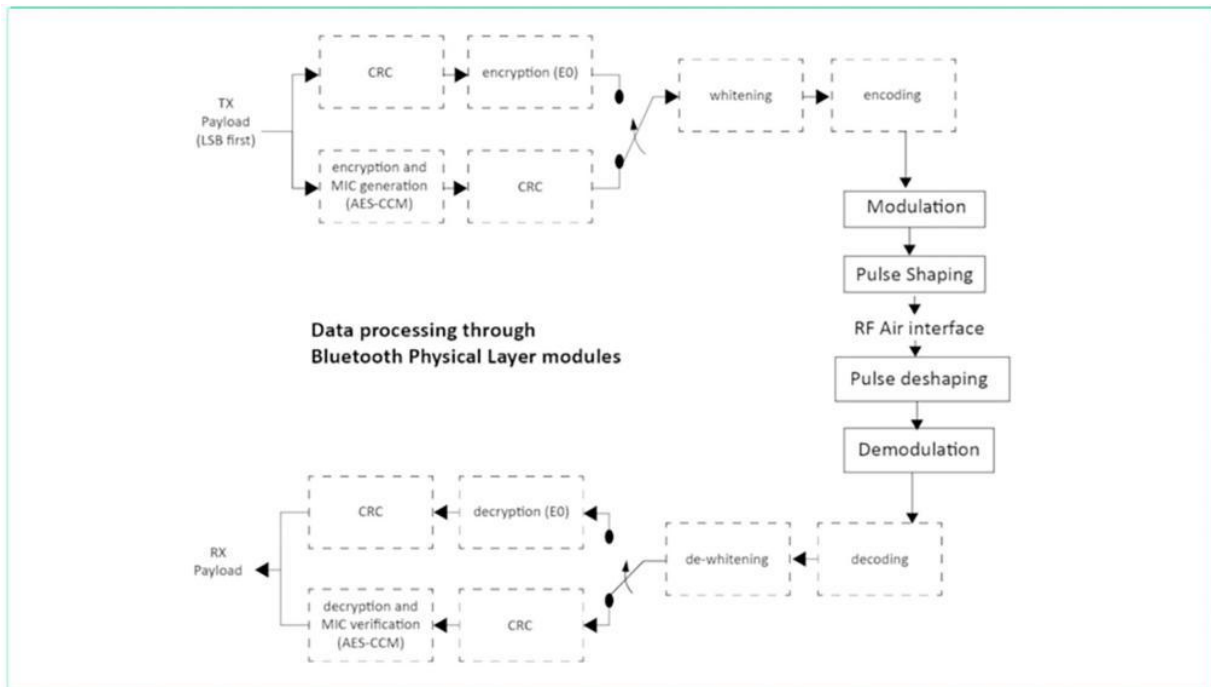
Take the smartphone for example, the controller is basically the Bluetooth chipset embedded inside the smartphone, whereas the host provides an

abstract interface for the application developer to interact with the Bluetooth operations.



The Bluetooth specification defines HCI as a set of commands and events for the host and the controller to interact with each other, along with a data packet format and a set of rules for flow control.

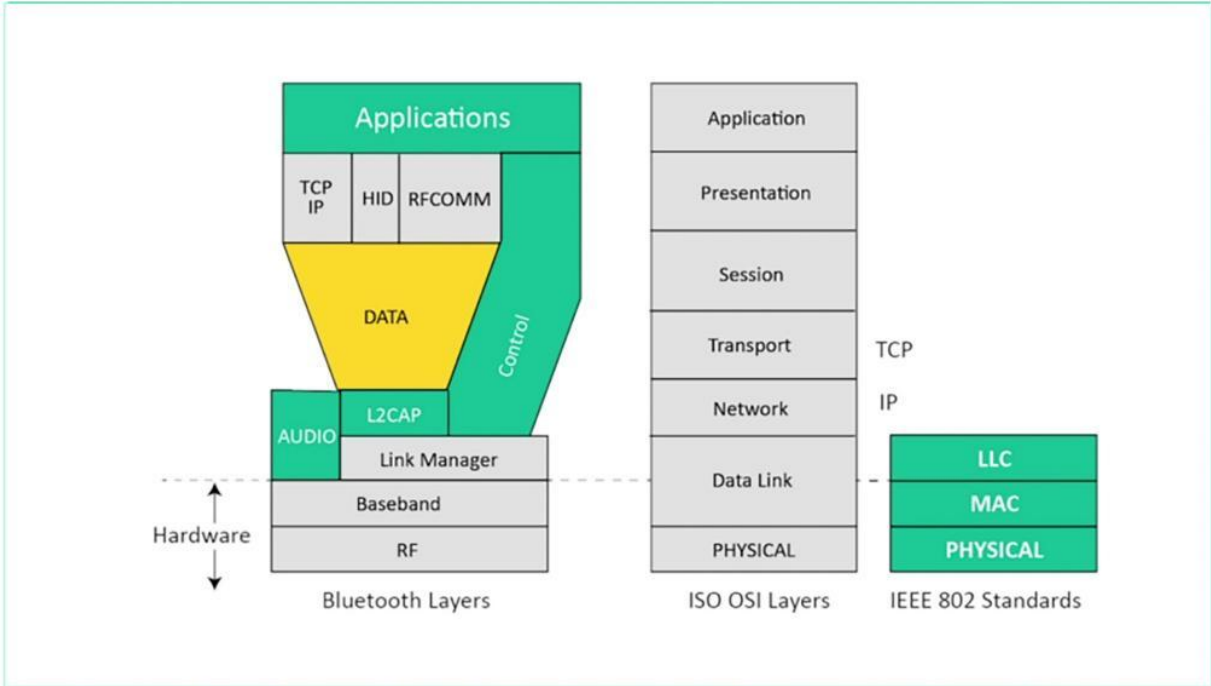
The Controller layer contains the following sublayers, which we will get in depth in later project: Host Controller Interface (HCI) ??? Controller side, Link Layer (LL), Physical Layer (PHY)



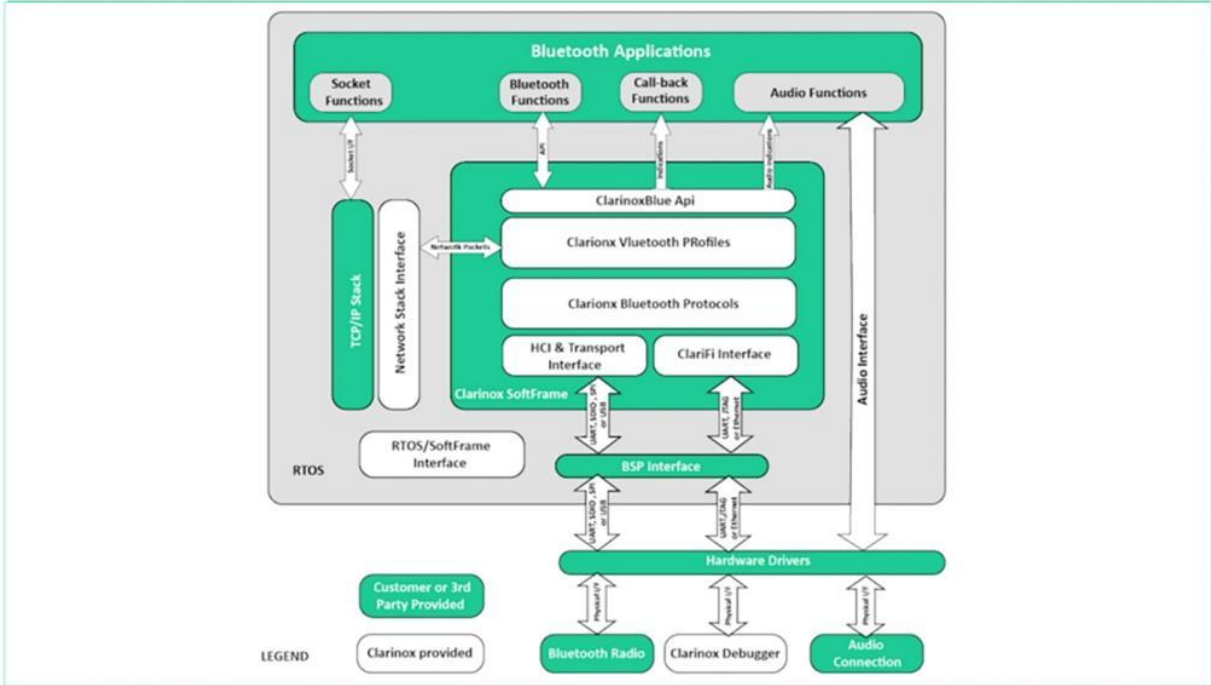
The Controller Layer handles the operation involving analog communications. Specifically, it defines the modulation and demodulation of analog signals and applies source coding to transform the signals into digital symbols. It also defines certain automated functionalities to avoid overloading the central processing unit that runs all of the software layers in the stack. These functionalities include Preamble, Access Address, and air protocol framing CRC generation and verification Data whitening Random number generation AES encryption CRC generation and verification Data whitening Random number generation AES encryption CRC generation and verification Data whitening Random number generation AES encryption The controller layer also defines the different roles of BLE Devices, like advertiser, scanner, master and slave.

APPLICATION, HOST, AND CONTROLLER

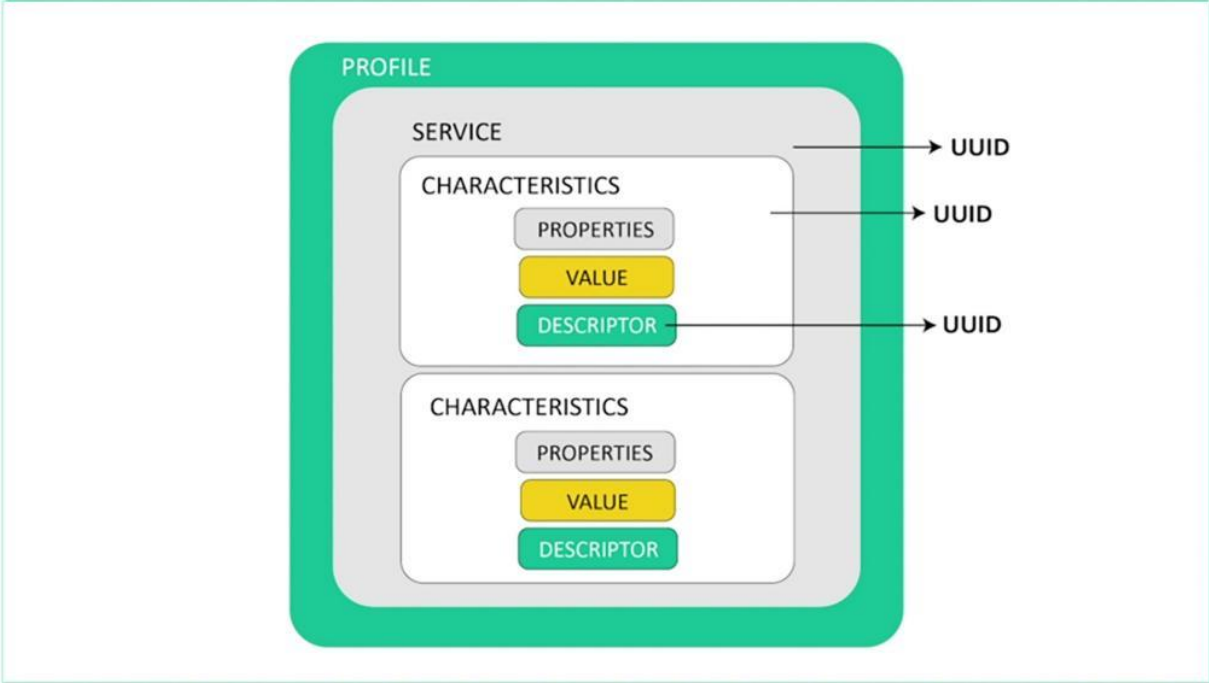
we will go much deeper into the roles and features of each layer. Lets kick off with the Application Layer.



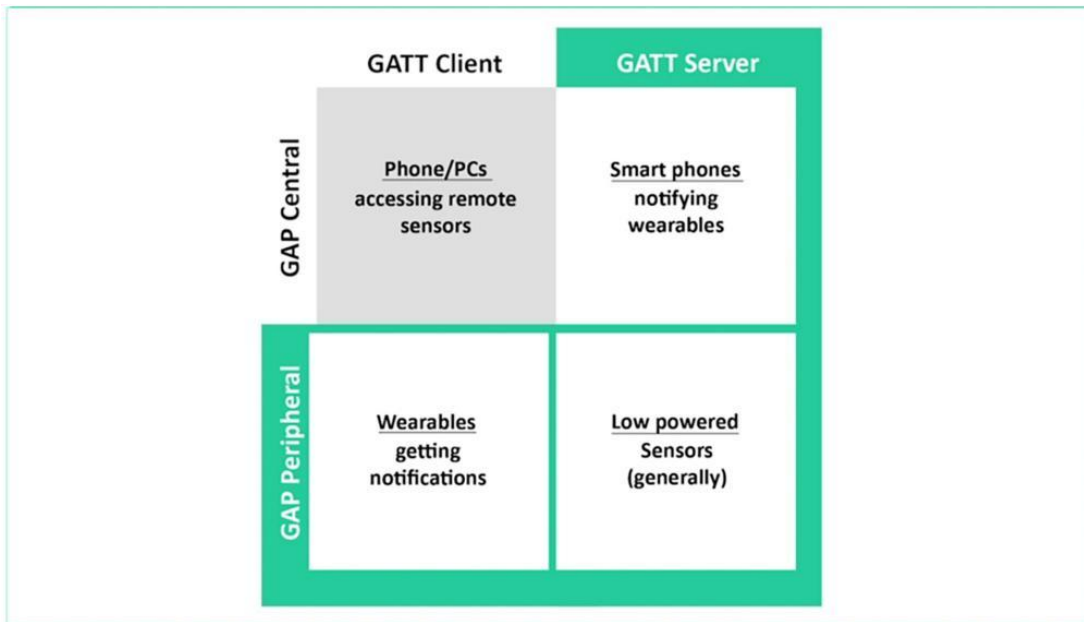
The application, like in all other types of systems, is the highest layer and the one responsible for containing the logic, user interface, and data handling of everything related to the actual use-case case that the application implements



the architecture of an application is highly dependent on each that the application implements The architecture of an application is highly dependent on each particular implementation. The profile defines a vertical interaction between the layers, as well as the peer to peer interactions of specific layers between devices. This consists of profiles and services. The profile defines the vertical interactions between the layers as well as the peer-to-peer interactions of specific layers between devices.

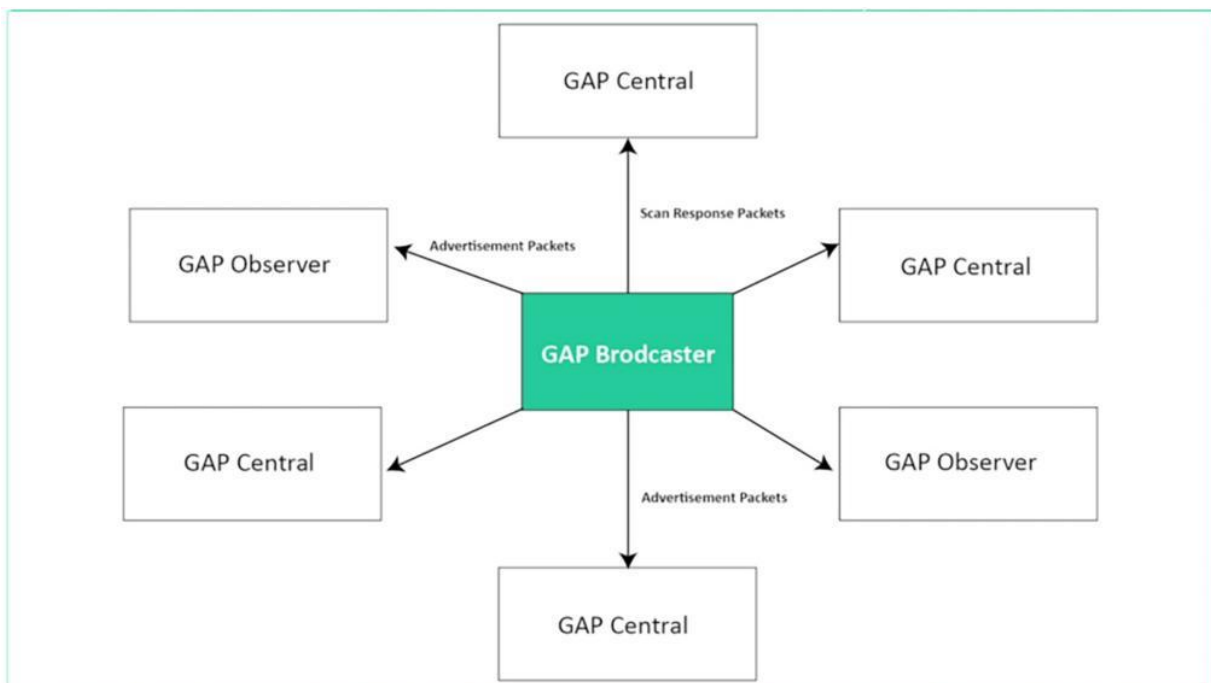


A profile composed of one or more services to address a particular use case. A service consists of characteristics or references to other services.

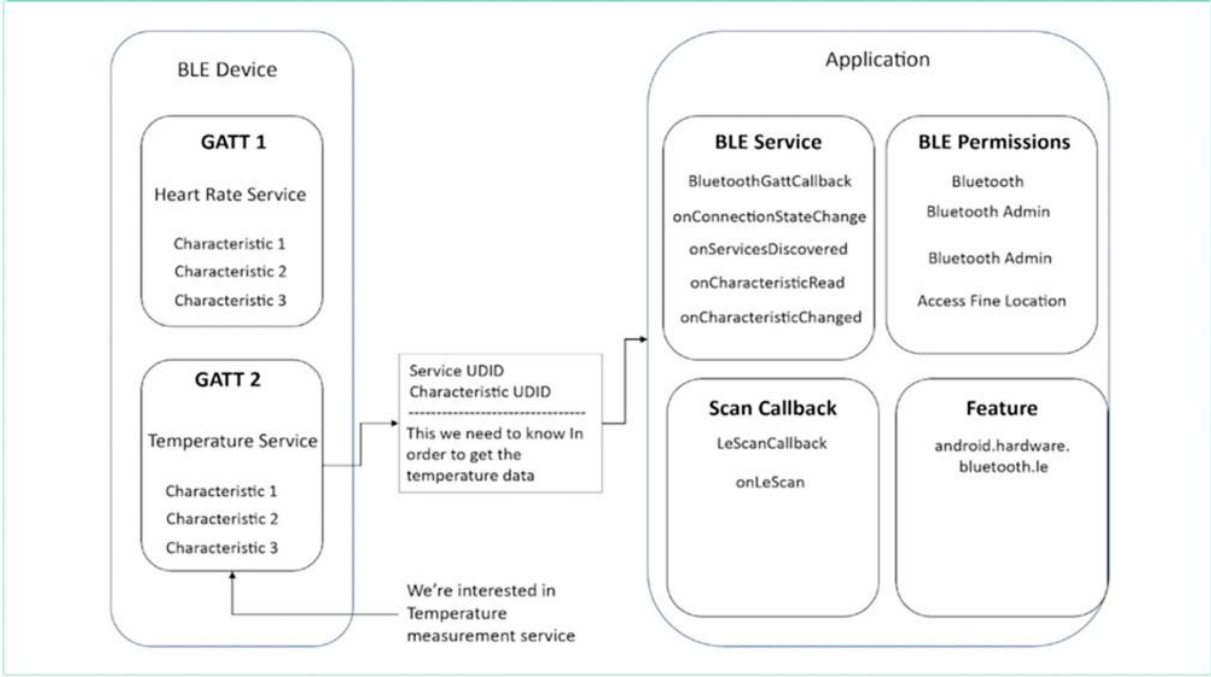


Any profiles/applications run on top of GAP/GATT layers of BLE protocol stack. It handles device discovery and connection related services for the BLE device.

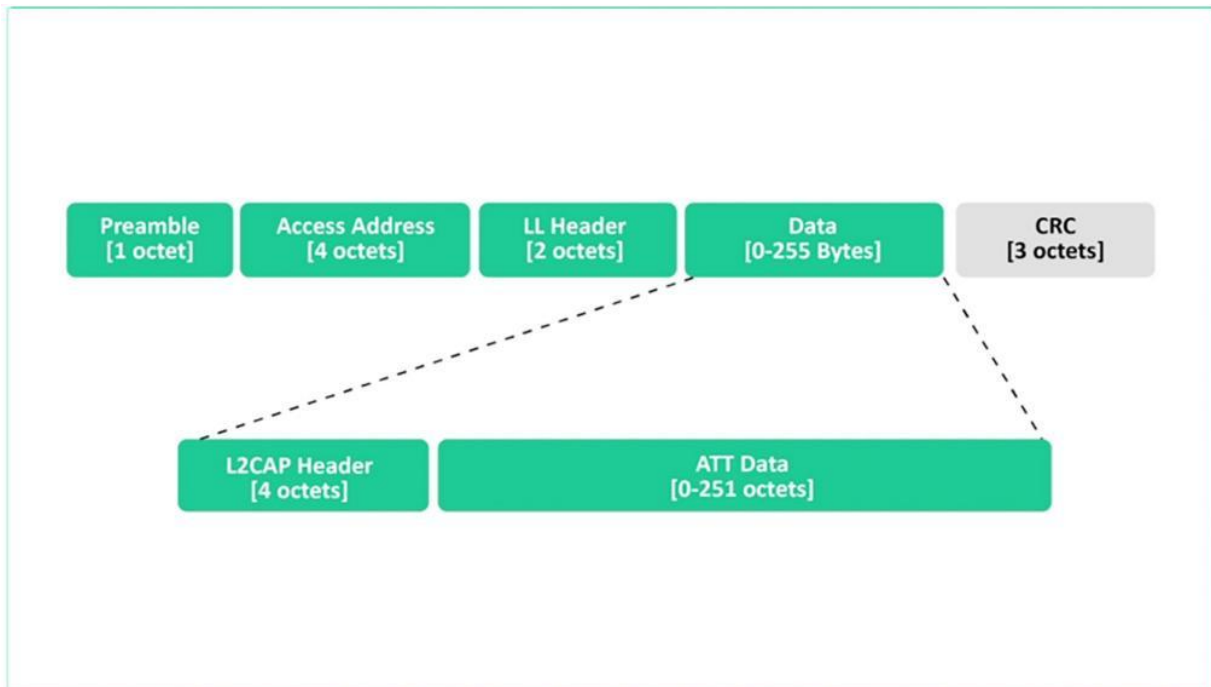
Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles. Next Let's look at the Host layer. As discussed in the last project, this layer consists of various sublayers, each with its own functions.



First let's look at GAP also known as Generic Access Profile. This layer directly interfaces with the application layer and/or profiles on it. It handles device discovery and connection related services for BLE devices. It also takes care of initiation of security features.

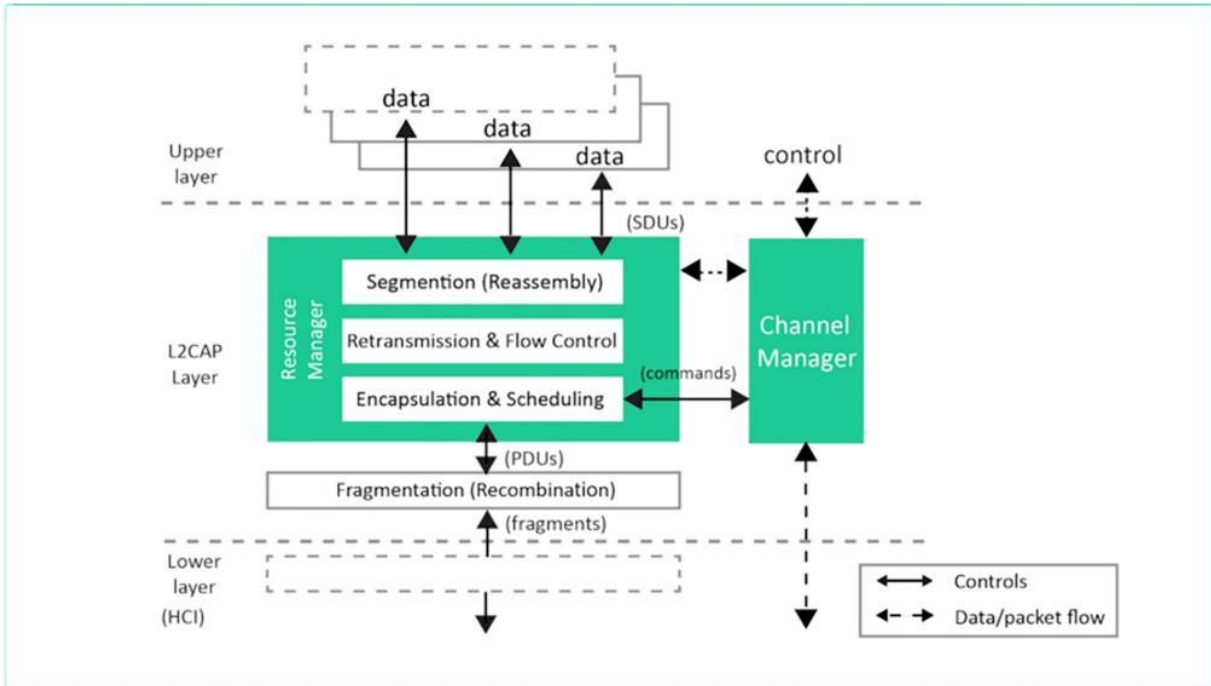


Next is GATT also known as Generic Attribute Profile This layer is a service framework which specifies sub-procedures to use ATT. Data communications between two BLE devices are handled through these sub-procedures. This layer is a service framework which specifies sub-procedures to use ATT. Data communications between two BLE devices are handled through these sub-procedures. The applications and/or profiles will use GATT directly. The applications and our profiles will use that directly.

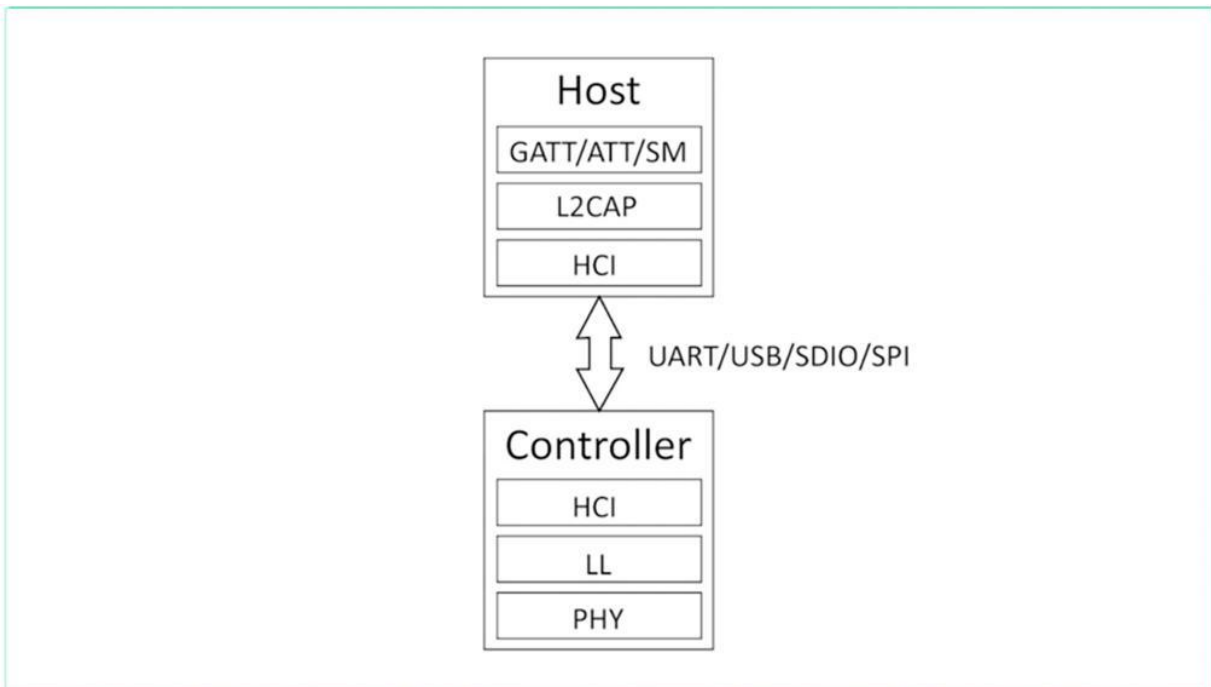


Next is ATT also known as Attribute Protocol. This layer allows BLE devices to expose certain pieces of data or attributes. This layer allows BEARLY devices to expose certain pieces of data or attributes.

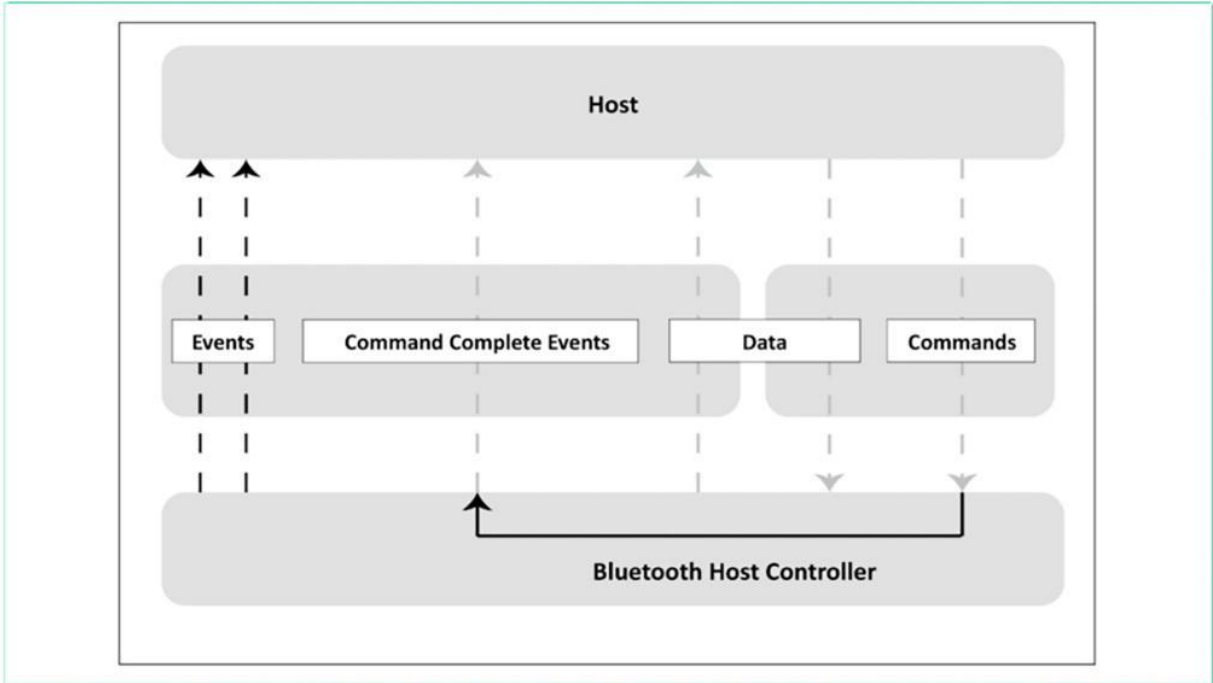
The next important sublayer is the Security Manager. This security Manager layer provides methods for device pairing and key distributions. It offers services to other protocol stack layers in order to securely connect and exchange data between BLE devices. It offers services to other protocol stack layers in order to securely connect and exchange data between BLE devices.



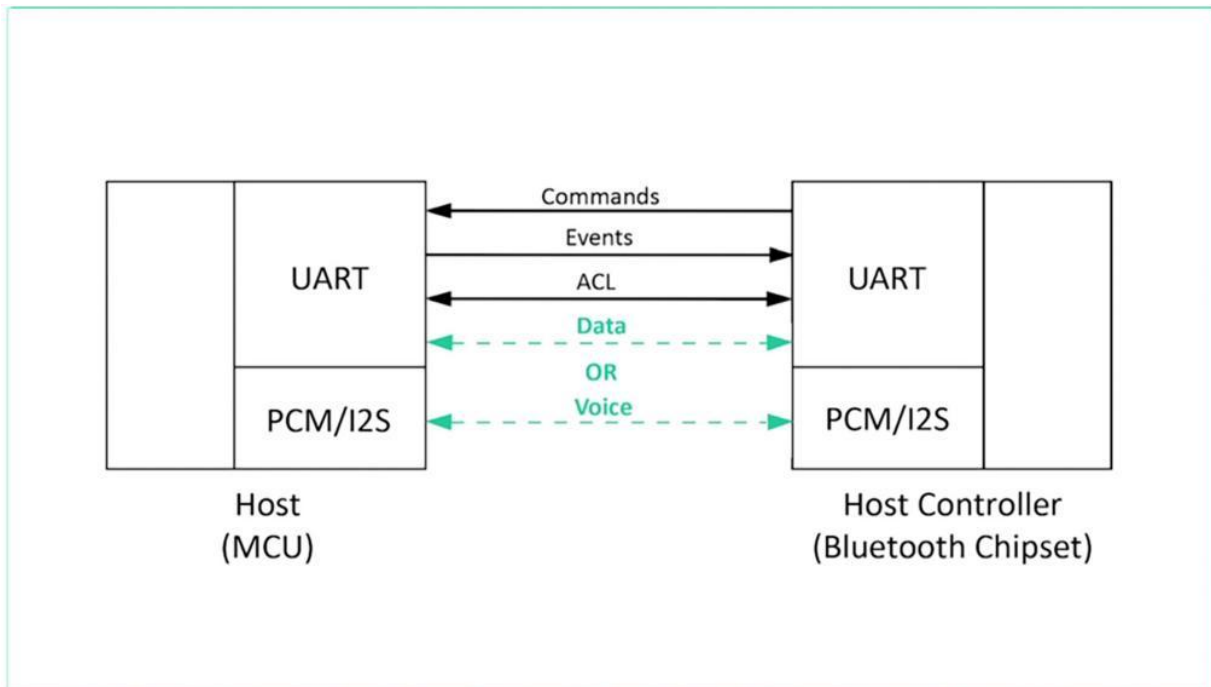
The next important sublayer is the L2CAP also known as Logical Link Control and Adaptation Protocol. This layer offers data encapsulation services to upper layers. This allows logical end to end data communication. This layer offers data encapsulation services to upper layers. This allows logical end to end data communication.



The final sublayer of the host layer that we are going to discuss is a shared sublayer. The HCI also known as the Host Controller Interface is the common sublayer shared among the host layer and the Controller Layer. Let call the HCI in the host layer as Host Side HCI and one in the controller layer as Controller Side HCI side HCA.



HC provides communication between controller and host through standard interface types. This HCI layer can be implemented either using API or by interfaces such as UART/SPI/USB.

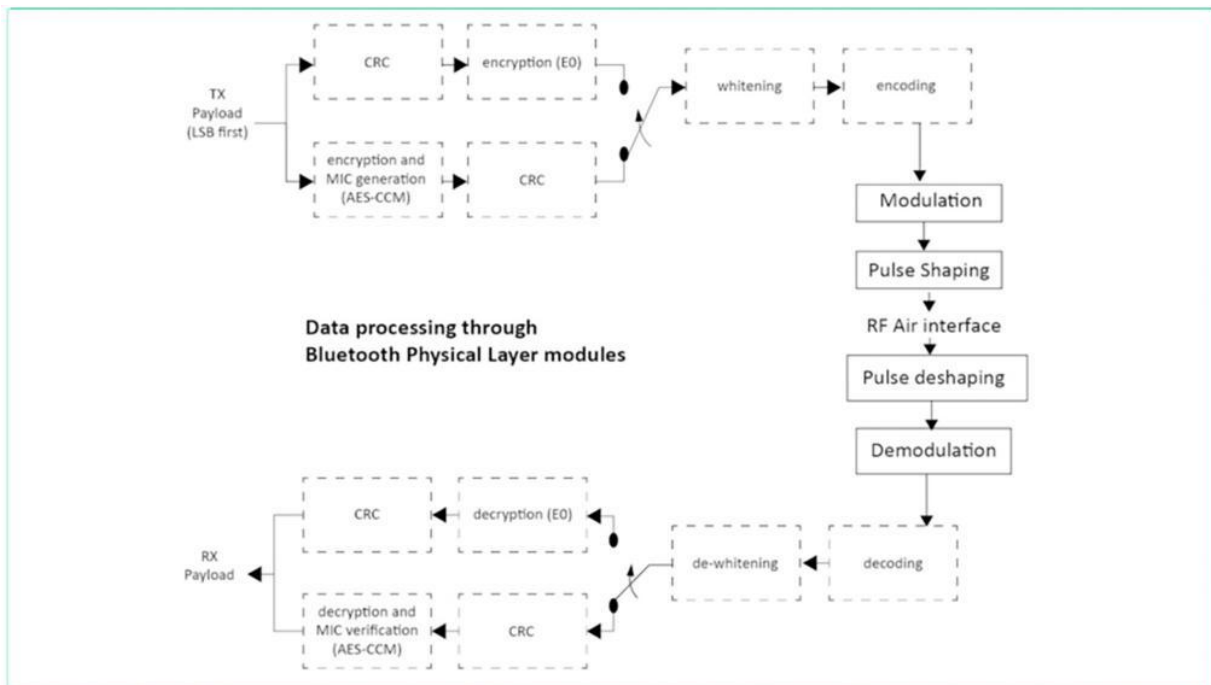


Standard HCI commands and events are defined in the bluetooth specifications. The Bluetooth specification defines HCI as a set of commands and events for the host and the controller to interact with each other, along with a data packet format and a set of rules for flow control and other procedures. to interact with each other, along with a data packet format and a set of rules for flow control and other procedures. Inside the controller layer, the first layer to learn about is the Link Layer also known as LL.

This layer sits above the Physical layer. It is responsible for advertising, scanning, and creating/maintaining connections. The role of BLE devices changes in peer to peer (i.e. Unicast) or broadcast modes. The common roles are Advertiser/Scanner (Initiator), Slave/Master or Broadcaster/Observer.

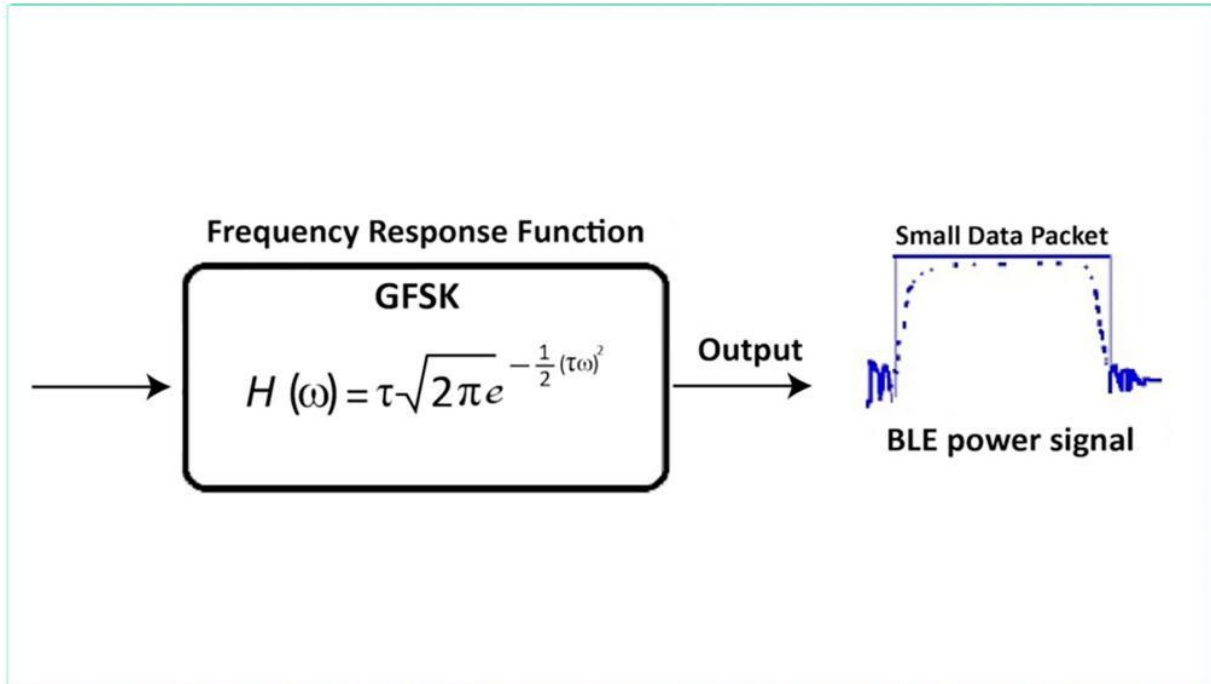
Link layer states are defined in this image The Link Layer is usually implemented as a combination of custom hardware and software. It is also the only hard real time constrained layer of the whole protocol stack, since it is responsible for complying with all of the timing requirements defined by the specification. It is therefore usually kept isolated from the higher layers of the protocol stack by means of a standard interface. It is therefore usually kept isolated from the higher layers of the protocol stack by means

of a standard interface. Now let's look at the physical it be. Its weight is short for physical, as in the physical lab.



This is the bottom layer of the belly stack and is responsible for actually transmitting and receiving information over there via radio waves.

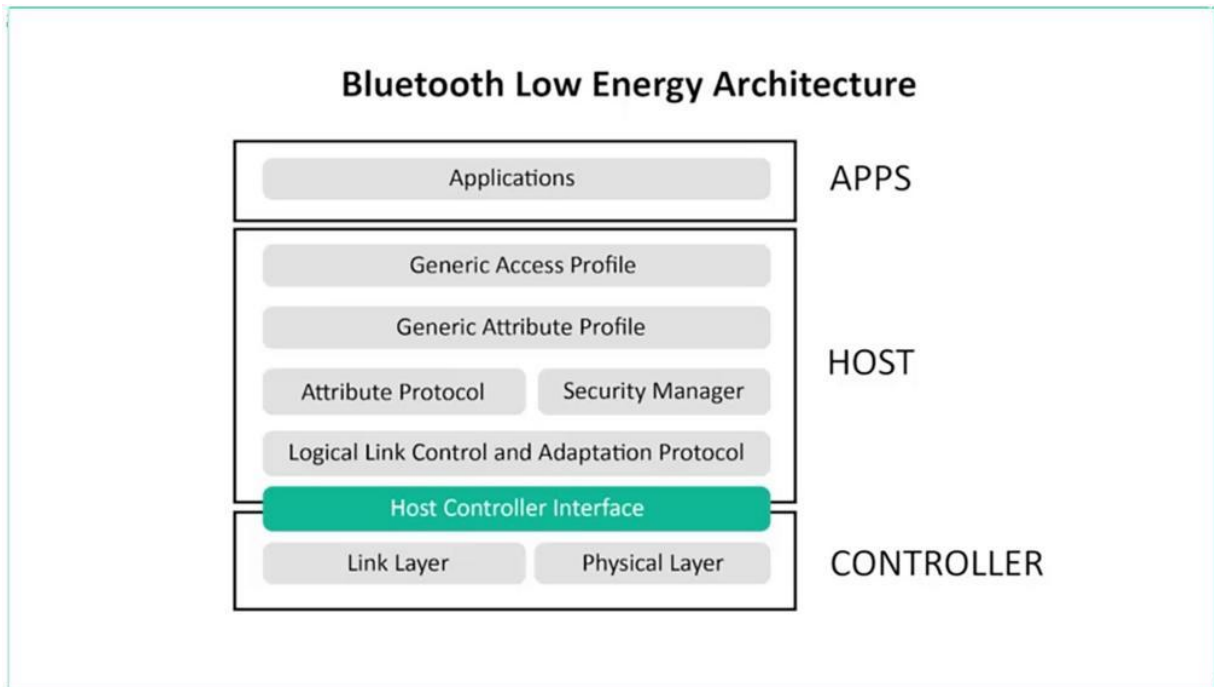
Bially in particular transmits on the two point four gigahertz band, also known as the Industry and Scientific and Medical Band. There are a number of advantages to transmitting on this band. Mainly, it is a licensed free band that is essentially free to use for short range applications. It's also the only licensed free band that's the same in every country. Unfortunately, these benefits make the two point four giga. It's been a popular choice, and as a result, overcrowding and condition are common issues.



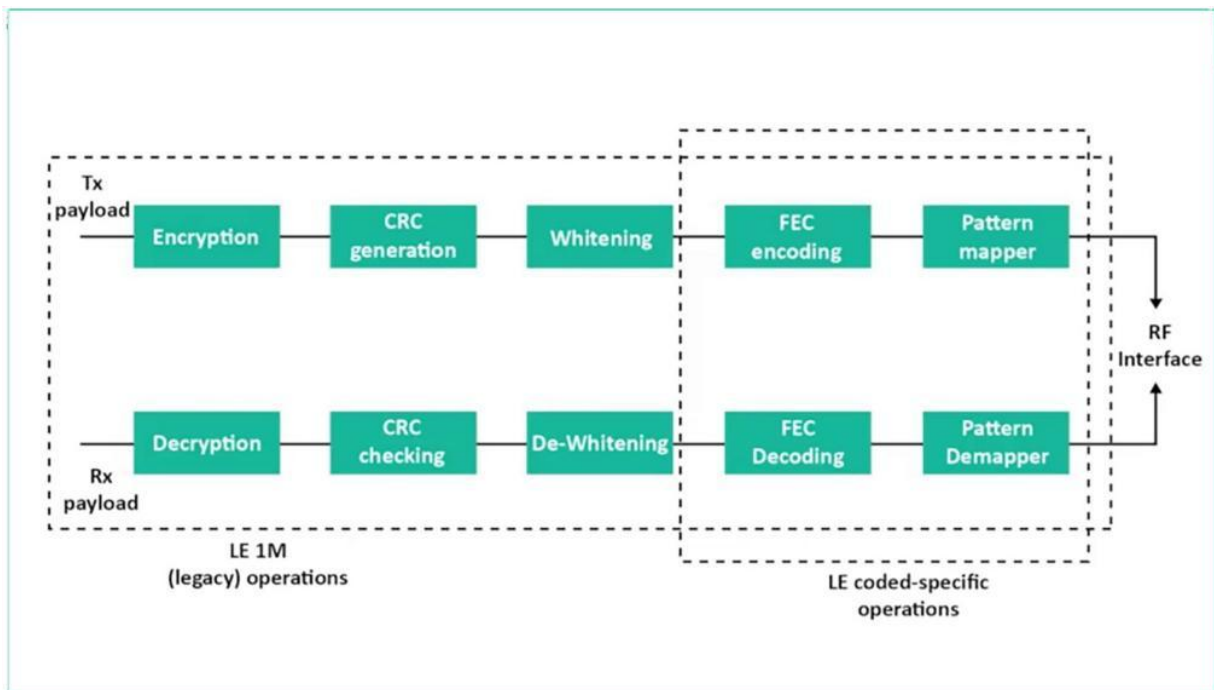
This is a refinement of Frequency Shift Keying (FSK), where information is coded by shifting the frequency up and down. This is a refinement of Frequency Shift Keying (FSK), where information is coded by shifting the frequency up and down. This is a refinement of Frequency Shift Keying (FSK), where information is coded by shifting the frequency up and down.

UNDERSTANDING THE BLE STACK

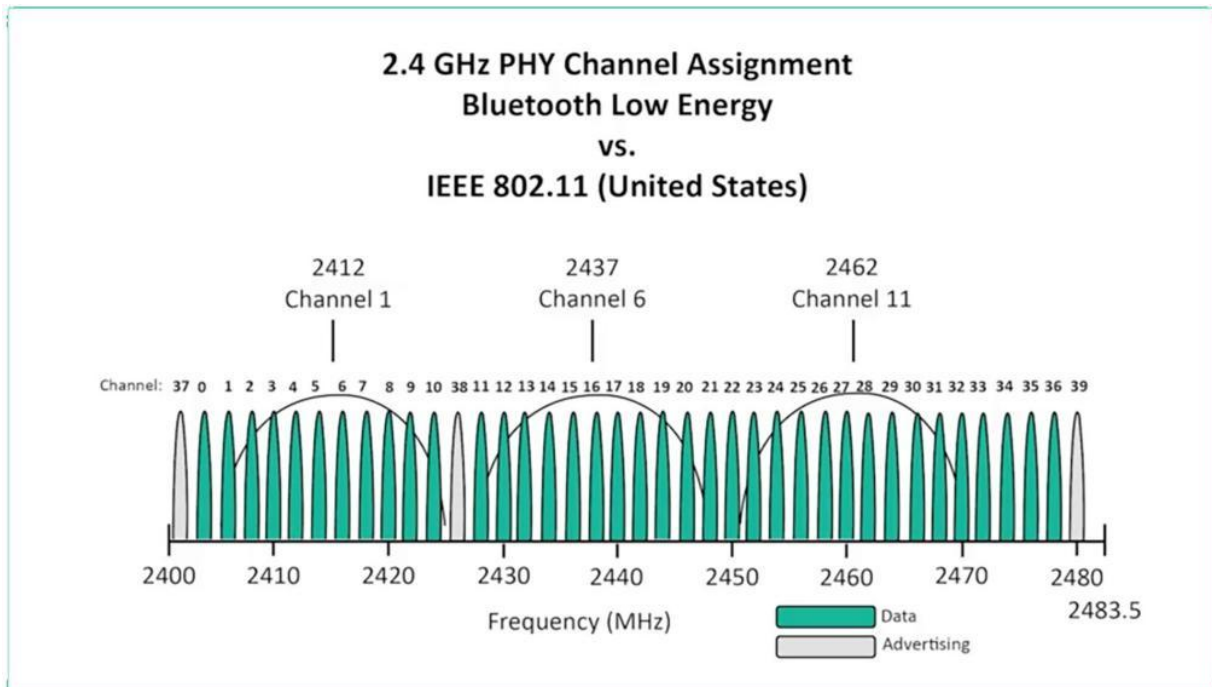
we just had a basic overview of the different sublayers and its functions.



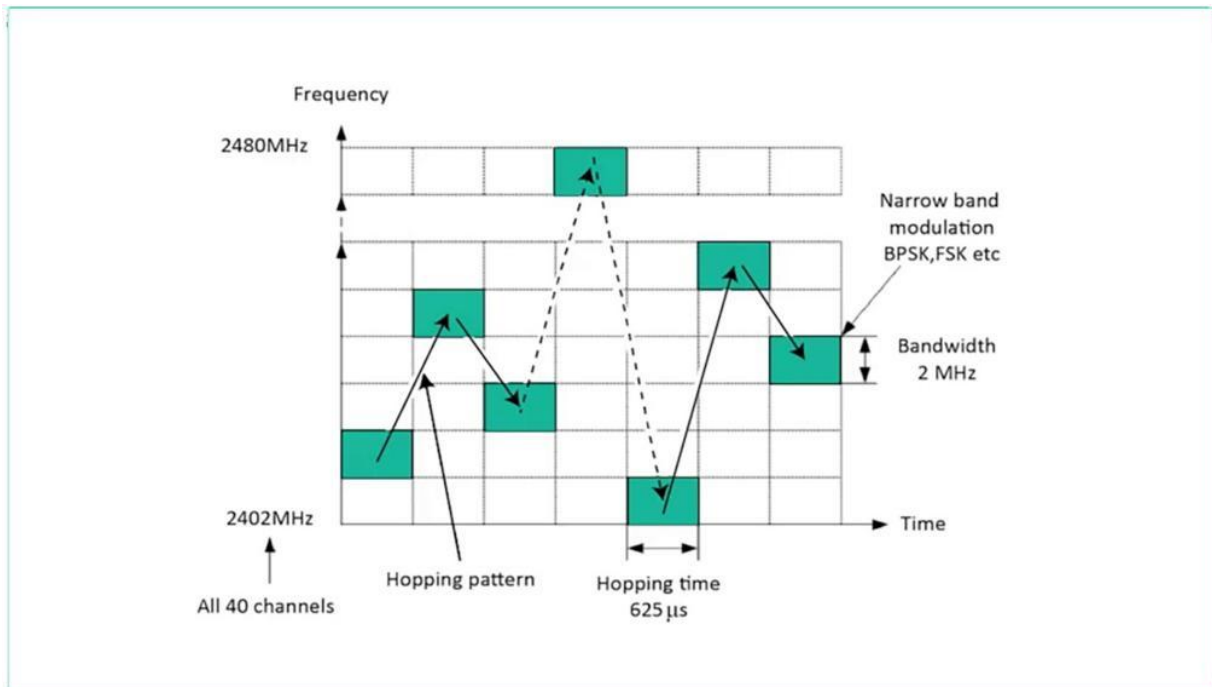
we will learn in depth about different aspects of the most important layers of the BLE Stack.



Let us look at the PHY layer in depth. As you already know, The physical layer (PHY) refers to the radio hardware used for communication and for modulating/demodulating the data.



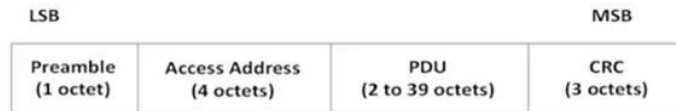
BLE operates in the ISM band (2.4 GHz spectrum), which is segmented into 40 RF channels, each separated by 2 MHz (center-to-center), as shown in the following figure which is segmented into 40 RF channels, each separated by 2 MHz (center-to-center), as shown in the following figure. Three of these channels are called the Primary Advertising Channels, while the remaining 37 channels are used for Secondary Advertisements and for data transfer during a connection. Let's look at some technical facts about PHY now;



It uses Frequency Hopping Spread Spectrum (FHSS), which allows the two communicating devices to switch to randomly (agreed-on) selected frequencies for exchanging data. It uses Frequency Hopping Spread Spectrum (FHSS), which allows the two communicating devices to switch to randomly (agreed-on) selected frequencies for exchanging data.

In older versions of Bluetooth (4.0, 4.1, and 4.2), the data rate was fixed at 1 Mbps. The physical layer radio (PHY) in this case is referred to as the 1M PHY and is mandatory in all versions including Bluetooth 5. With Bluetooth 5, however, two new optional PHYs were introduced: bps PHY, used to achieve twice the speed of earlier versions of Bluetooth. Coded PHY, used for longer range communication. bps PHY, used to achieve twice the speed of earlier versions of Bluetooth. Coded PHY, used for longer range communication. bps PHY, used to achieve twice the speed of earlier versions of Bluetooth. Coded PHY, used for longer range communication. The link layer is the layer that interfaces with the physical layer (radio) and provides the higher-level layers an abstraction and a way to interact with the radio through an intermediary level called the HCI layer which we will discuss later in this project.

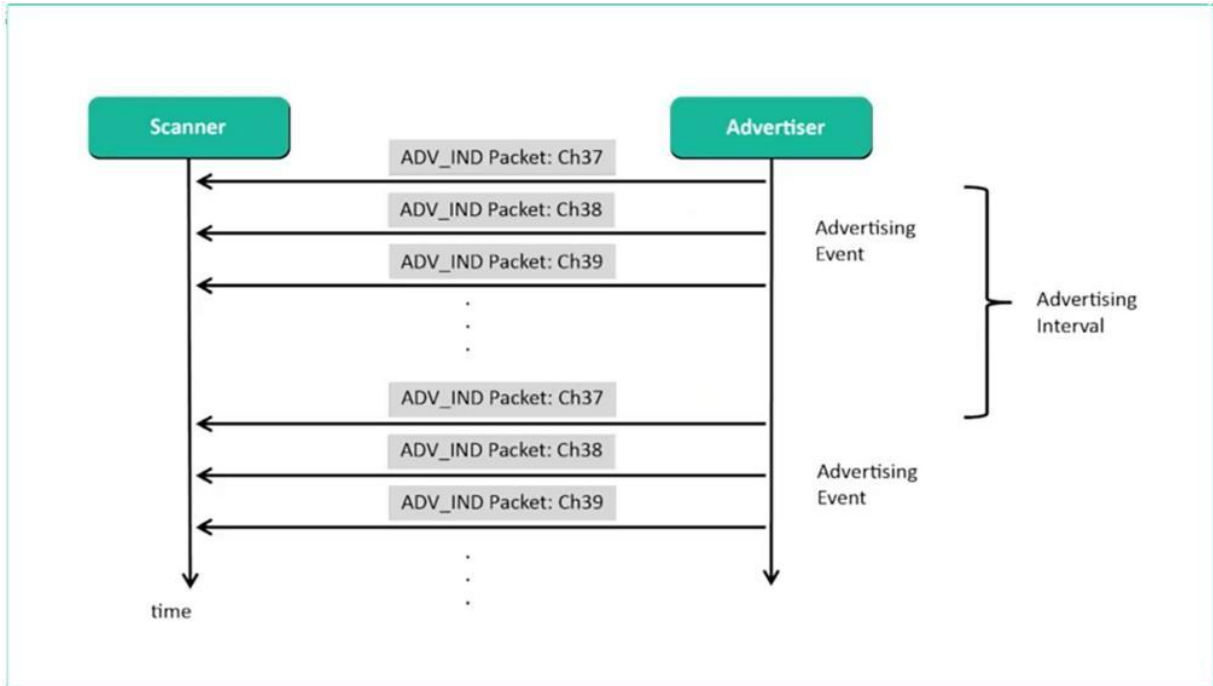
Link Layer



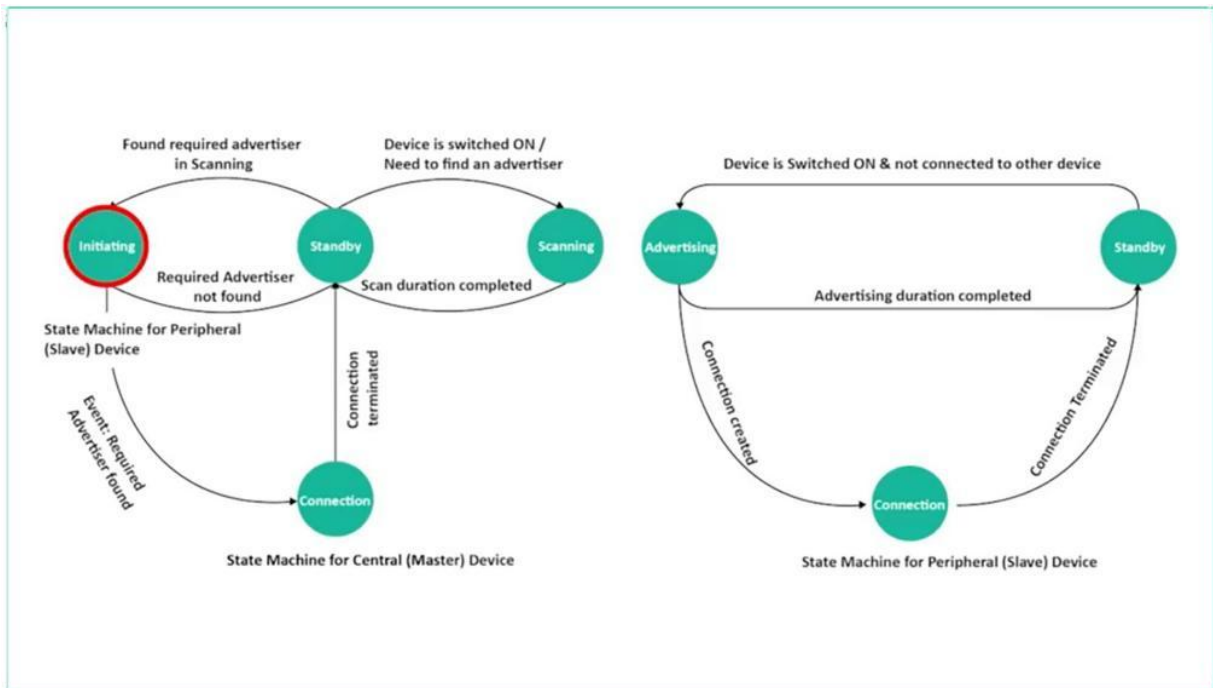
- **Preamble:** an alternating binary sequence for synchronization
- **Access Address:** unique identifier which defines a particular connection
 - Fixed value for communications in advertising channel : 0x8E89BED6
- **PDU:** protocol data unit which is the actual payload (variable length)
- **CRC:** for error checking
 - depends on CRC Init and the PDU
 - Computed using Linear Feedback Shift Register (LFSR)
- Whitening is applied to the PDU and CRC
 - Not complicated as it depends only on channel number
 - Computed using LFSR
- Each Bluetooth device has a unique MAC address

It is responsible for managing the state of the radio as well as the timing requirements necessary for satisfying the BLE specification. It is also responsible for managing hardware accelerated operations such as: CRC, random number generation, and encryption. It is also responsible for managing hardware accelerated operations such as: CRC, random number generation, and encryption. The link layer manages the different states of the radio, shown in the following figure: Standby: the default state in which the radio does not transmit or receive any data.

Advertising: the state in which the device sends out advertising packets for other devices to discover and read.



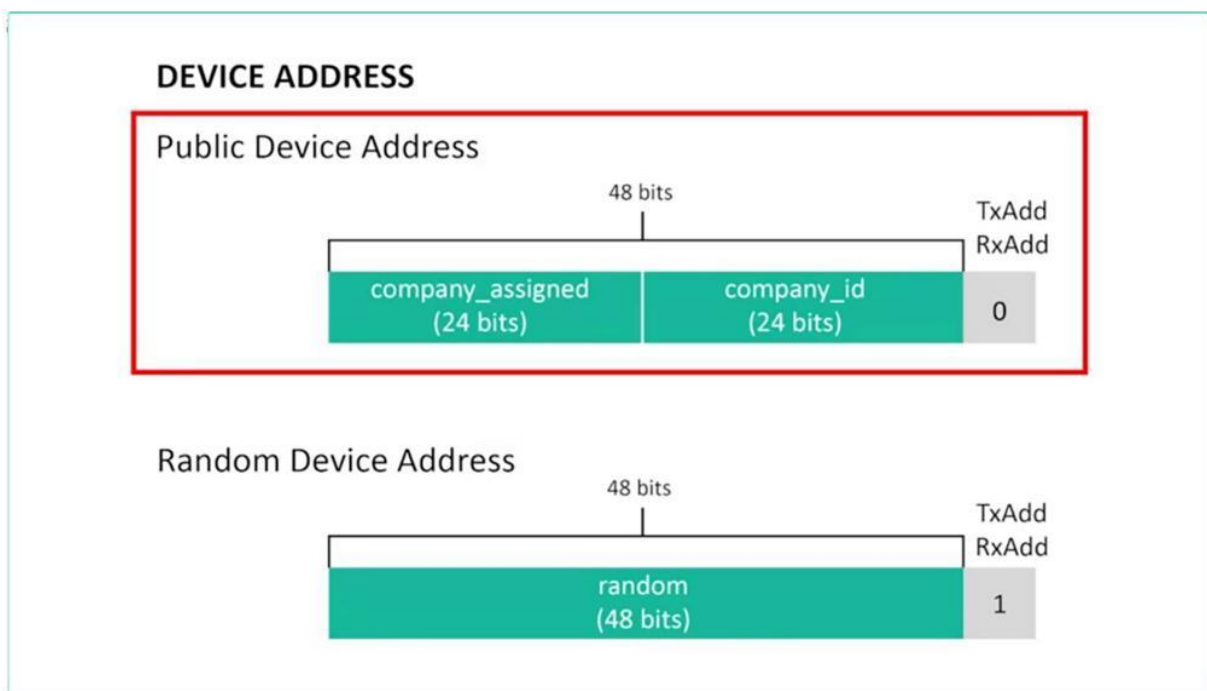
Scanning: the state in which the device scans for devices that are Advertising



Initiating: the state in which a scanning device decides to establish a connection with a device that is advertising. Connection the state in which

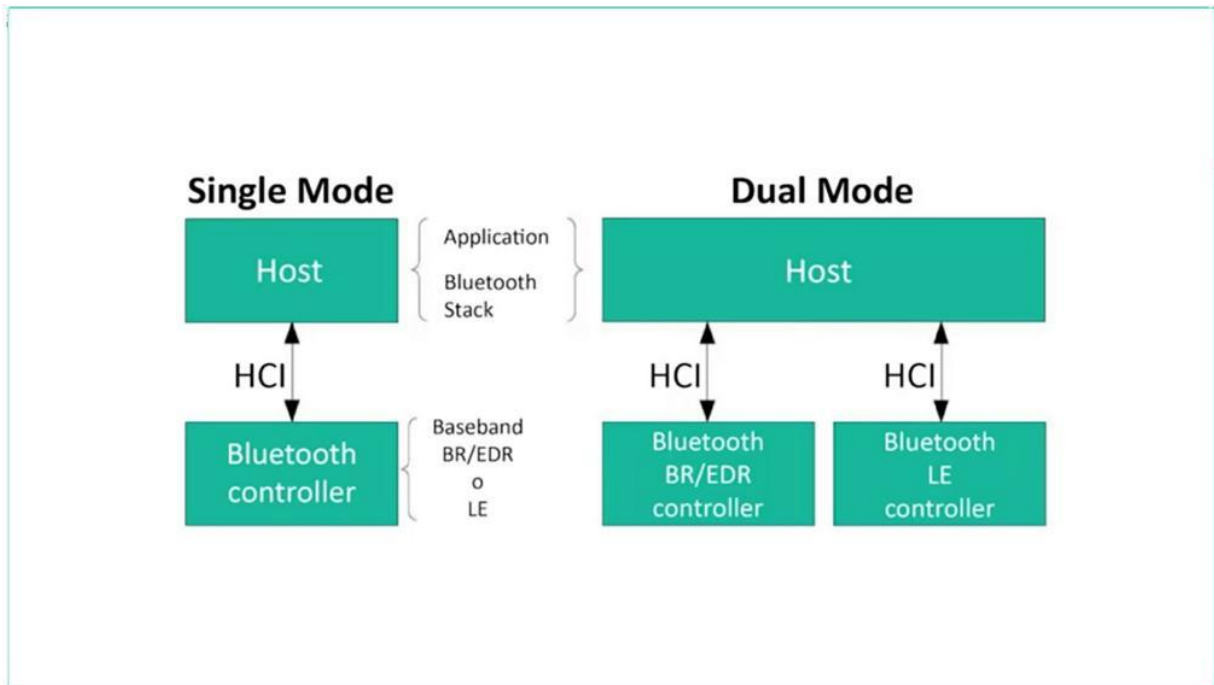
a device has an established link with another device and regularly exchanges data with this other device.

The Link Layer is also responsible for managing the Different Addresses in BLE. Bluetooth devices are identified by a 48-bit address, similar to a MAC address. There are two main types of addresses: Public Addresses and Random Addresses. in BLE. Bluetooth devices are identified by a 48-bit address, similar to a MAC address. There are two main types of addresses: Public Addresses and Random Addresses. in BLE. Bluetooth devices are identified by a 48-bit address, similar to a MAC address.

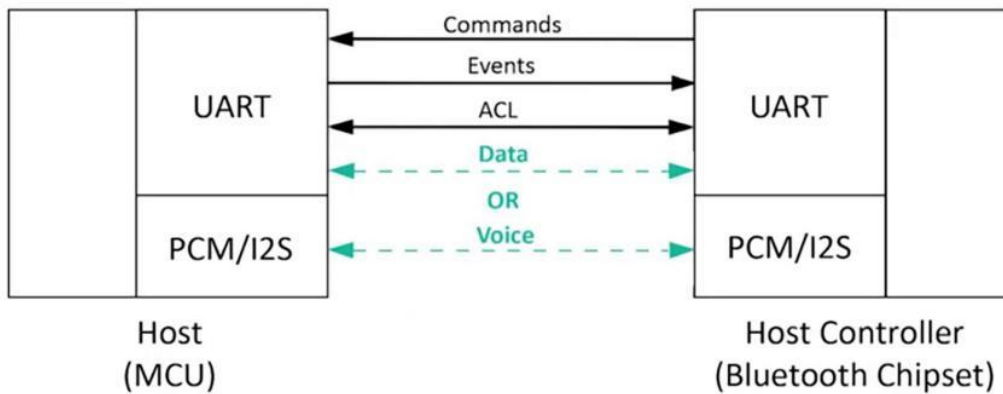


There are two main types of addresses: Public Addresses and Random Addresses. in BLE. Bluetooth devices are identified by a 48-bit address, similar to a MAC address. There are two main types of addresses: Public Addresses and Random Addresses. Public Address This is a fixed address that does not change and is factory-programmed. It must be registered with the IEEE (similar to a WiFi or Ethernet device MAC address). Public Address This is a fixed address that does not change and is factory-programmed. It must be registered with the IEEE (similar to a WiFi or Ethernet device MAC address). Random Address Since manufacturers have a choice on what type of address to use (Random vs. Public), Random addresses are more popular since they do not require registration with the

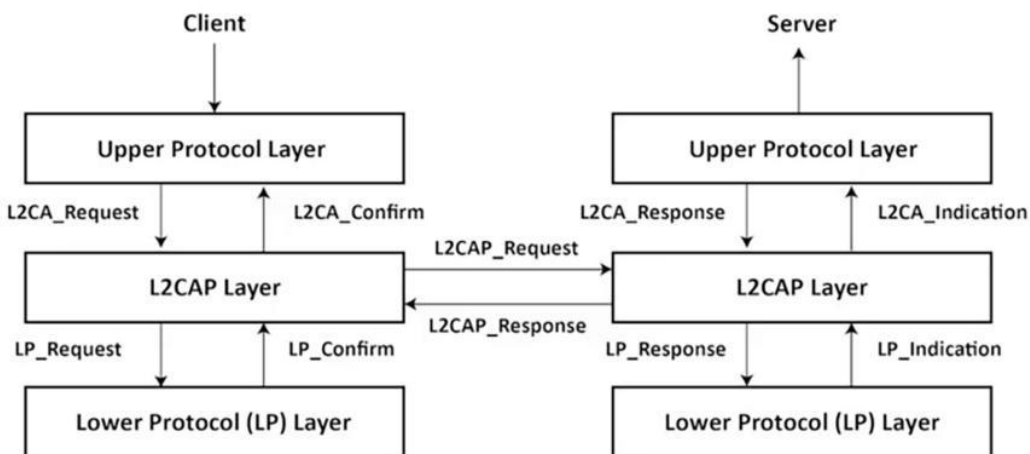
IEEE. It can be one of the two subtypes. It is either static address or private address. Next Lets look at the HCI Layer also known as Host Controller Interface. The HCI layer is a standard protocol defined by the Bluetooth specification that allows the host layer to communicate with the controller layer The HCI layer is a standard protocol defined by the Bluetooth specification that allows the host layer to communicate with the controller layer



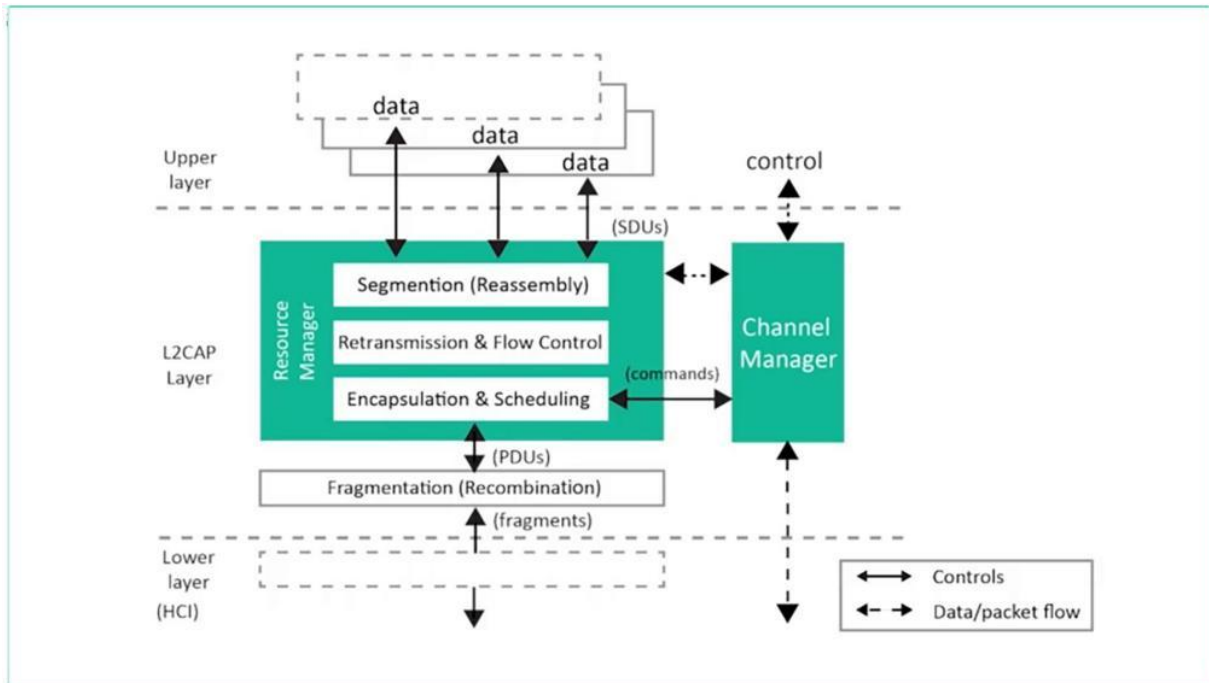
These layers could exist in separate chipsets, or they could exist in the same chipset. In this sense, it also allows interoperability between chipsets, so a device developer can choose two Bluetooth certified devices, a controller and a host, and be 100% confident that they are compatible with each other in terms of communication between the host and controller layers are compatible with each other in terms of communication between the host and controller layers implemented over a physical communication interface.



In the case where the host and controller are in separate chipsets, the HCI layer will be implemented over a physical communication interface. The job of the HCI layer is to relay commands from the host down to the controller and send events back up from the controller to the host. Next let's look at the L2CAP Layer: The Logical Link Control and Adaptation Protocol (L2CAP) layer acts as a protocol multiplexing layer.



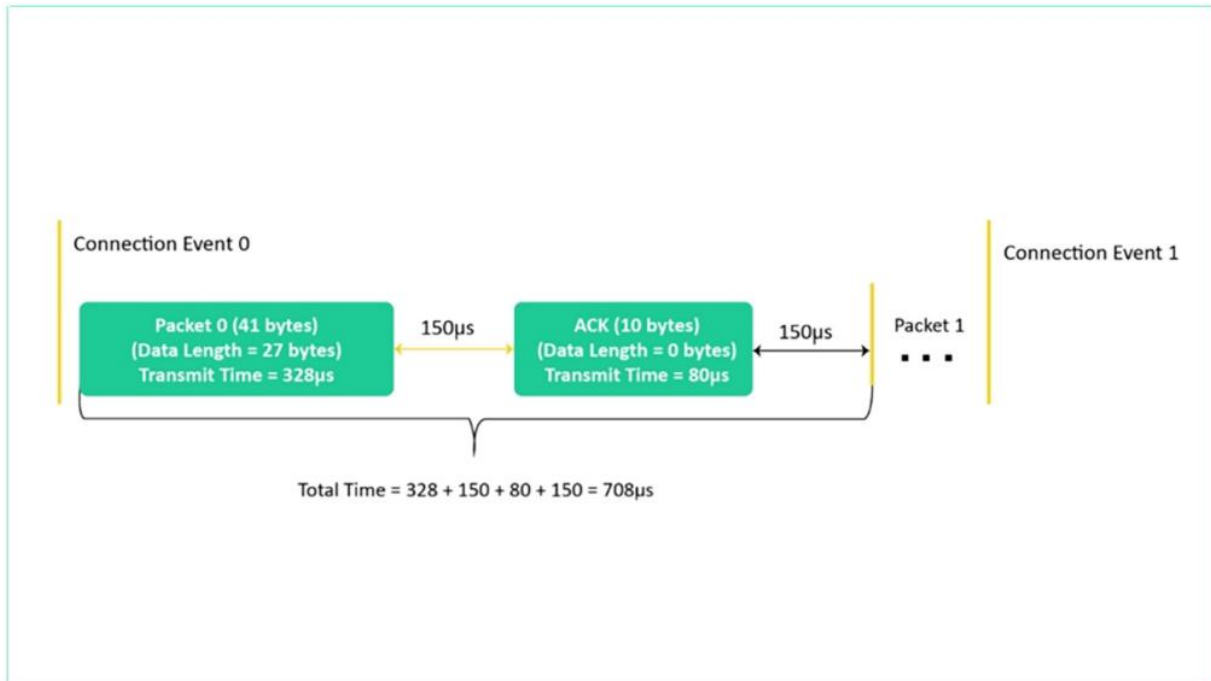
It is borrowed from the Bluetooth Classic standard, and performs the following tasks in the case of BLE: Takes multiple protocols from the upper layers and places them in standard BLE packets that are passed down to the lower layers beneath it. packets that are passed down to the lower layers beneath it to handle fragmentation and recombination.



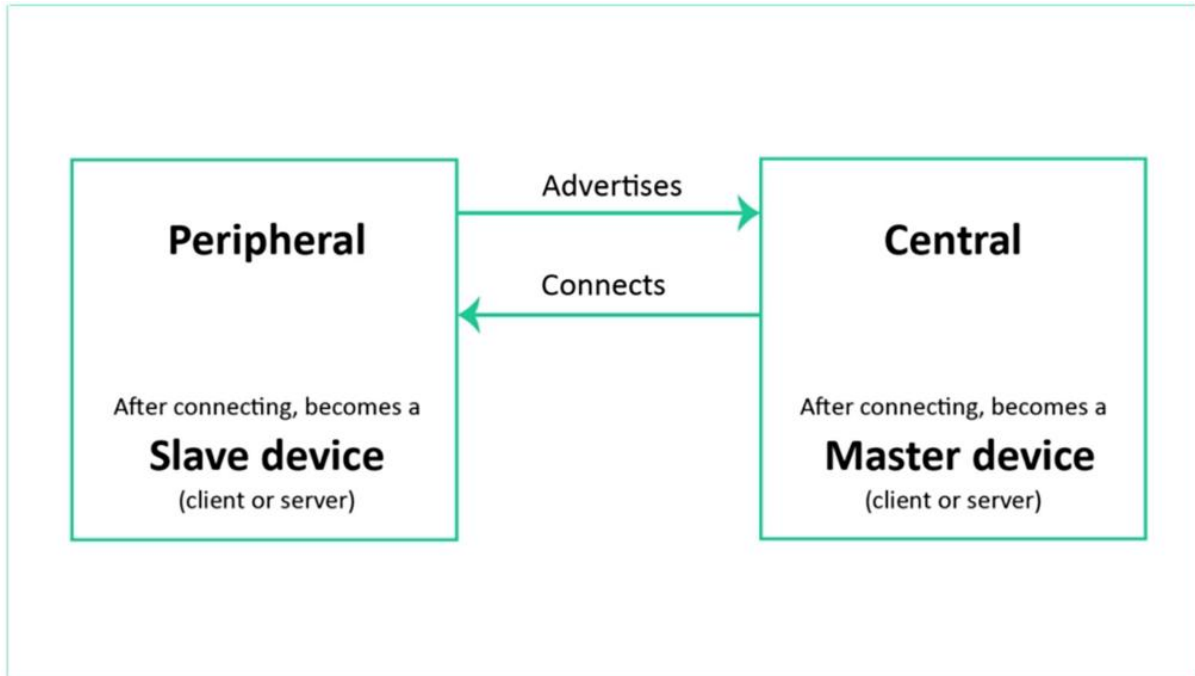
Handles fragmentation and recombination. payload size supported for transmission. It takes the larger packets from the upper layers and splits them into chunks that fit into the maximum BLE payload size supported for transmission. On the receiver side, it takes multiple packets and combines them into one packet that can be handled by the upper layers.

CONNECTIONS, AND SERVICES IN BLE

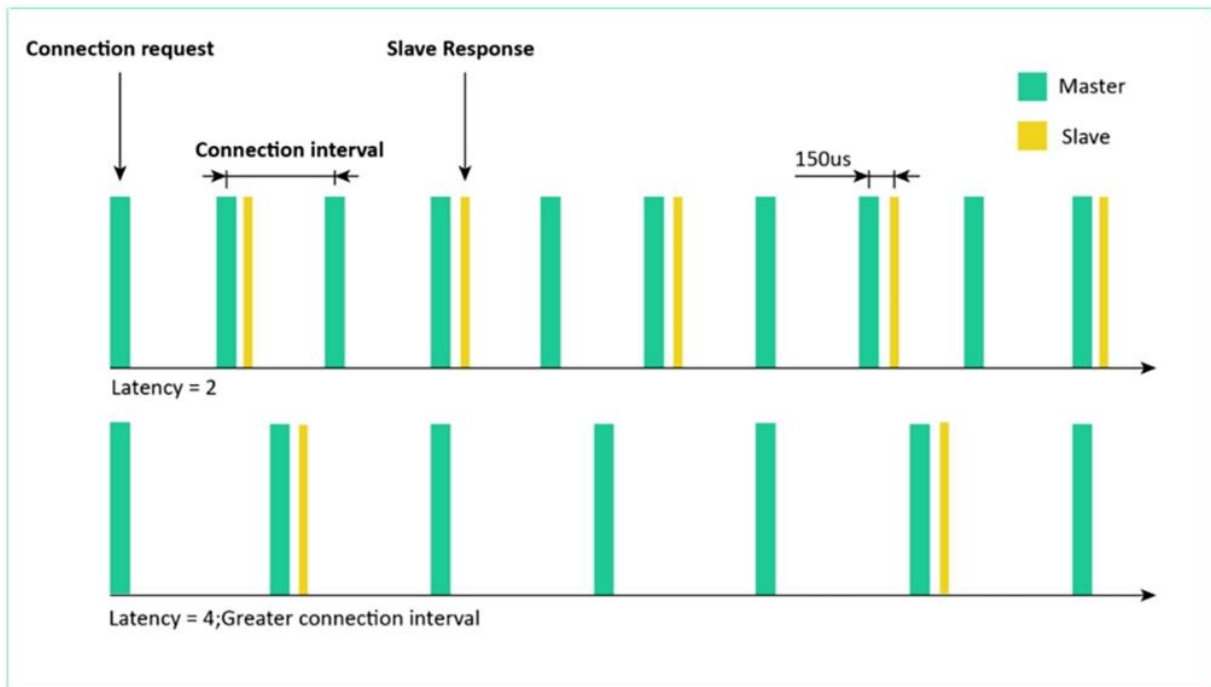
Here are a few aspects of connections that are very important to know: A connection event occurs periodically and continuously until the connection is closed or lost.



A connection event contains at least one packet sent by the master. The slave always sends a packet back if it receives a packet from the master

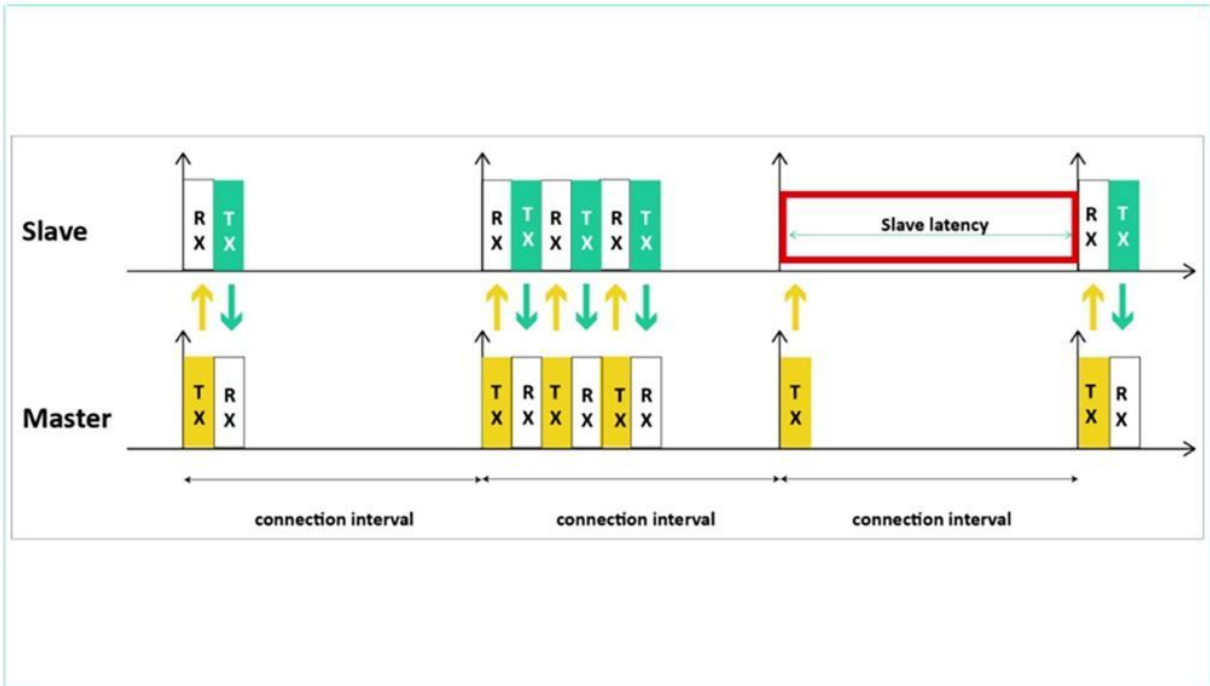


If the master does not receive a packet back from the slave, the master will close the connection. If the master receives a packet from the slave, the master will not close the connection. If the master does not receive a packet back from the slave, the master will close the connection. If the master receives a packet from the slave, the master will not close the connection. The connection event can be closed by either side. The connection event can be closed by either side.

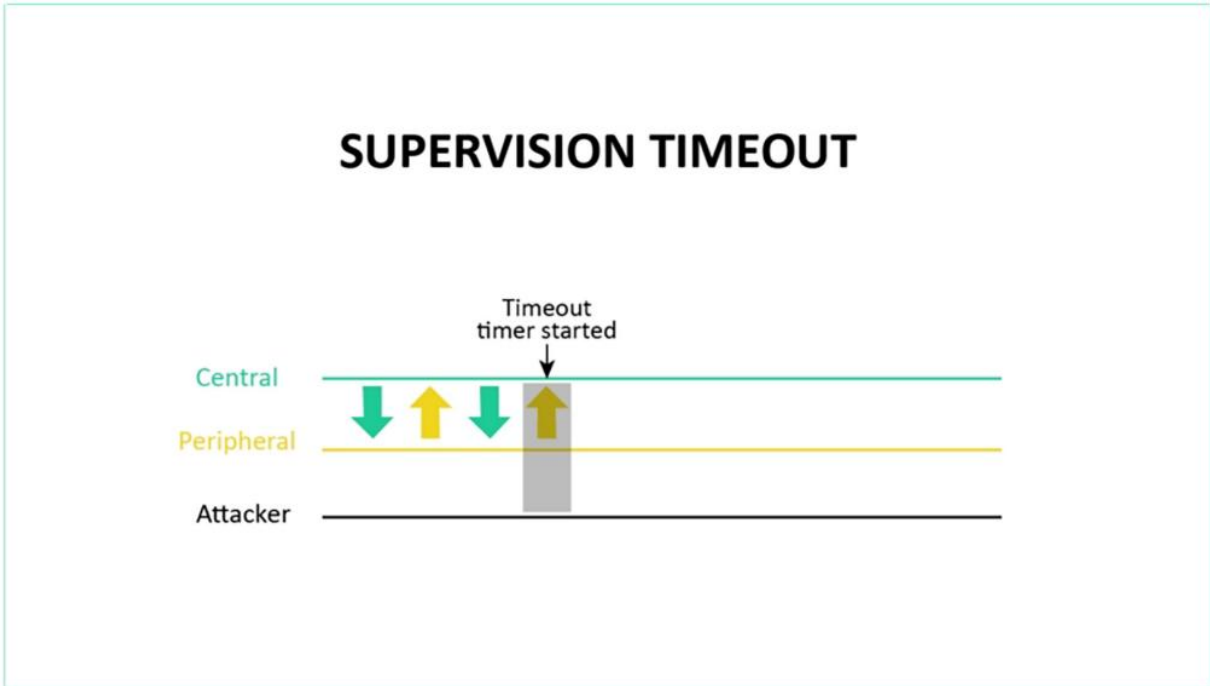


The starting points of consecutive Connection Events are spaced by a period of time called the connection interval. end of it.

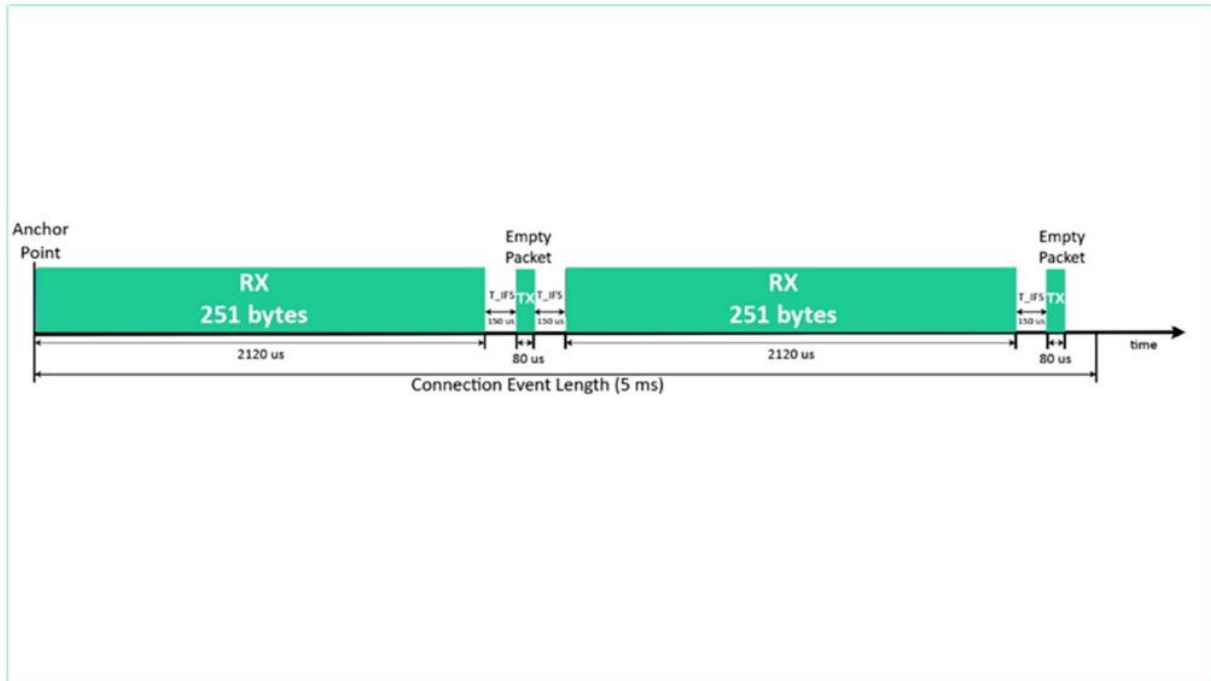
Now let's look at some of the important connection parameters Connection Interval: The Connection Interval value ranges between 7.5 milliseconds - 4.0 seconds in increments of 1.25 milliseconds. It is set by the central device in the connection request packet Connection Interval: The Connection Interval value ranges between 7.5 milliseconds - 4.0 seconds in increments of 1.25 milliseconds. It is set by the central device in the connection request packet five milliseconds. Connection Interval: The Connection Interval value ranges between 7.5 milliseconds - 4.0 seconds in increments of 1.25 milliseconds.



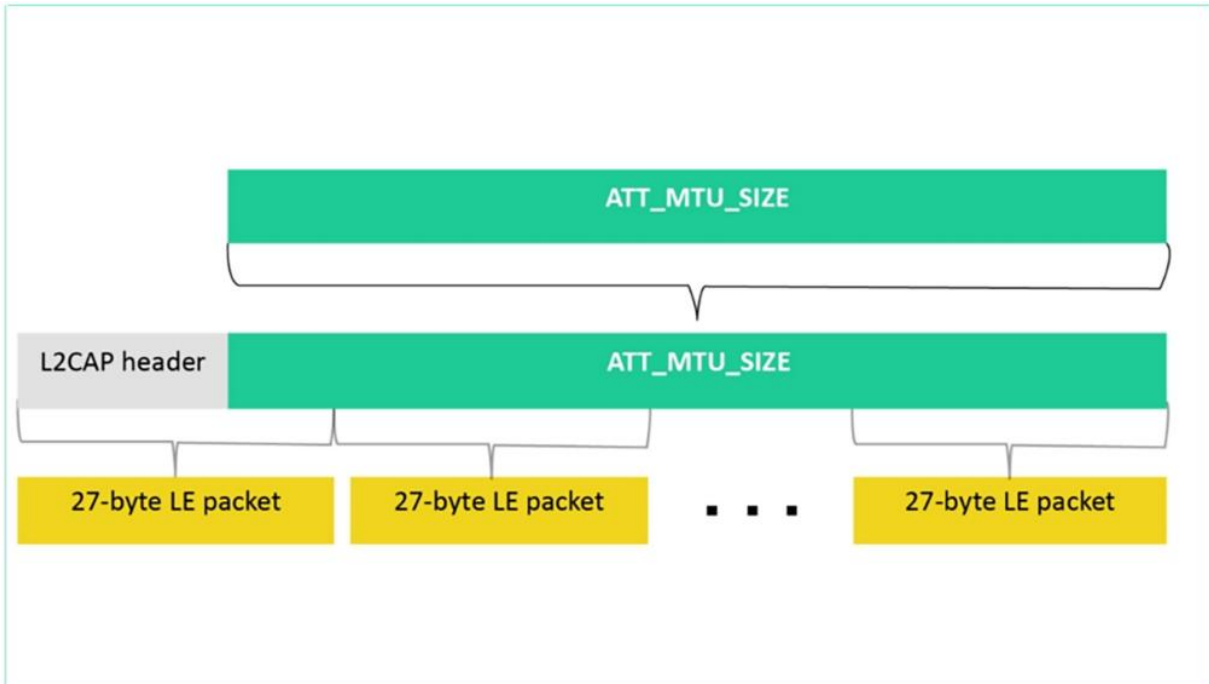
It is set by the central device in the connection request packet Slave Latency: The slave latency parameter allows the peripheral to skip a number of consecutive connection Events and not listen to the central at these connection events without compromising the connection. This allows the peripheral to sleep for longer periods of time, potentially reducing power consumption.



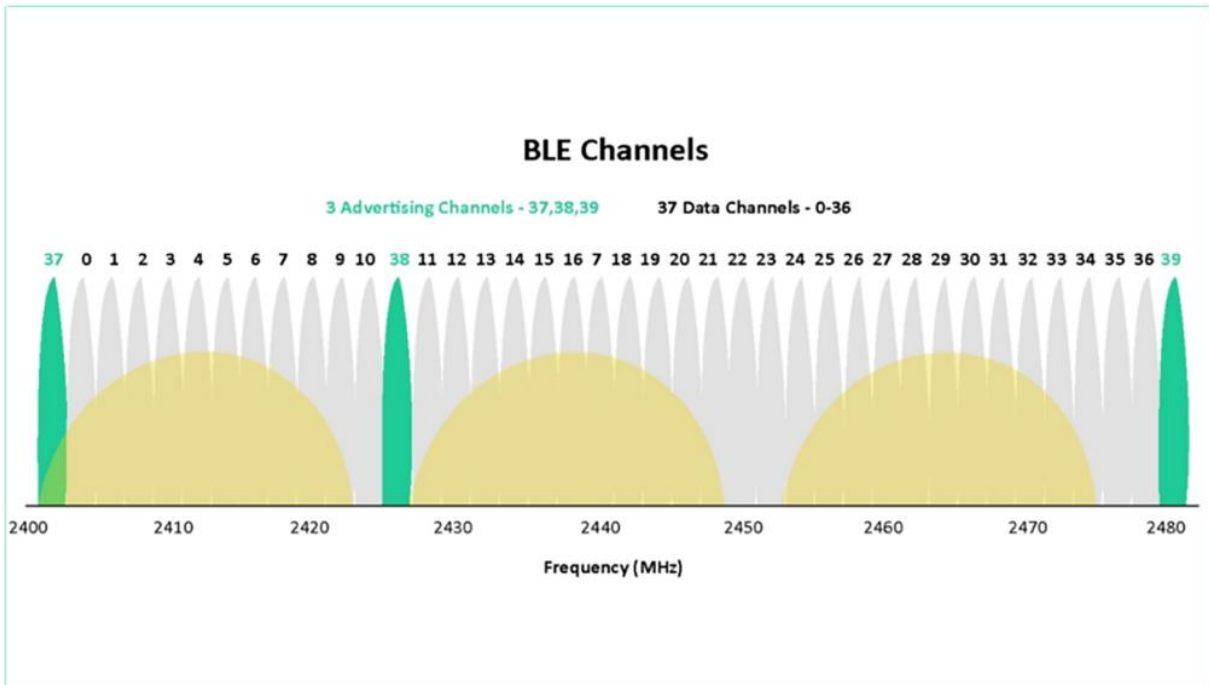
Supervision Timeout: The supervision timeout is used to detect a loss in connection. It is defined as the maximum time between two received data packets before the Connection is considered lost. It is defined as the maximum time between two received data packets before the Connection is considered lost.



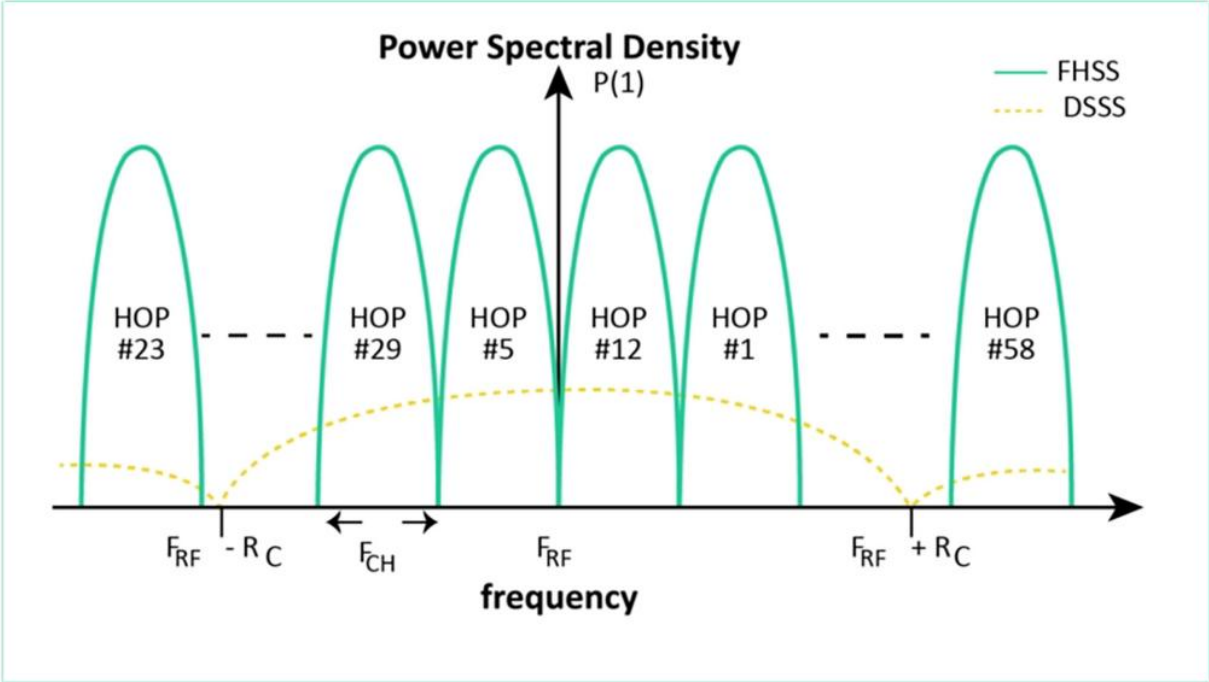
Data Length Extension (DLE) This is a setting that can be enabled or disabled. It allows the packet size to hold a larger amount of payload up to 251 bytes vs. 27 when disabled. This feature was introduced in version 4.2 of the Bluetooth specification.



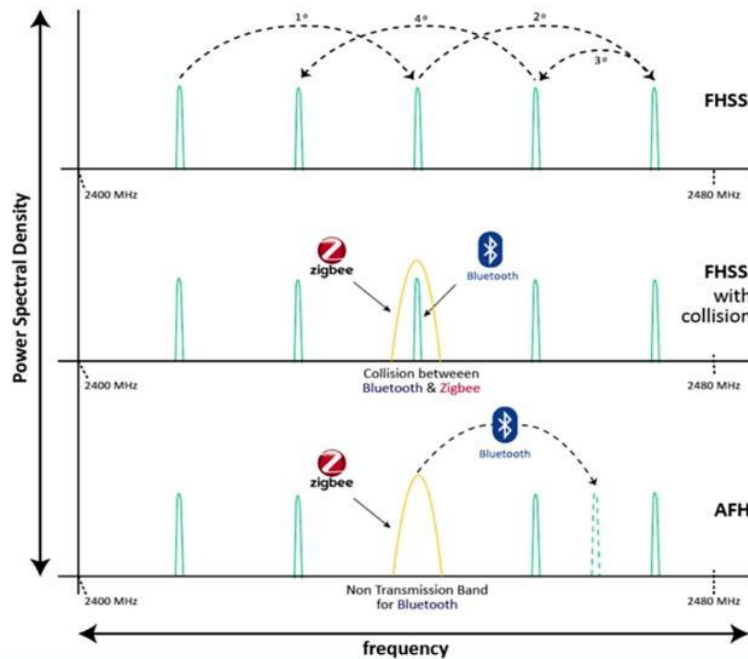
Maximum Transmission Unit (MTU) MTU stands for Maximum Transmission Unit and is used in computer networking to define the maximum size of a Protocol Data Unit (PDU) that can be sent by a specific protocol. the maximum size of a Protocol Data Unit (PDU) that can be sent by a specific protocol.



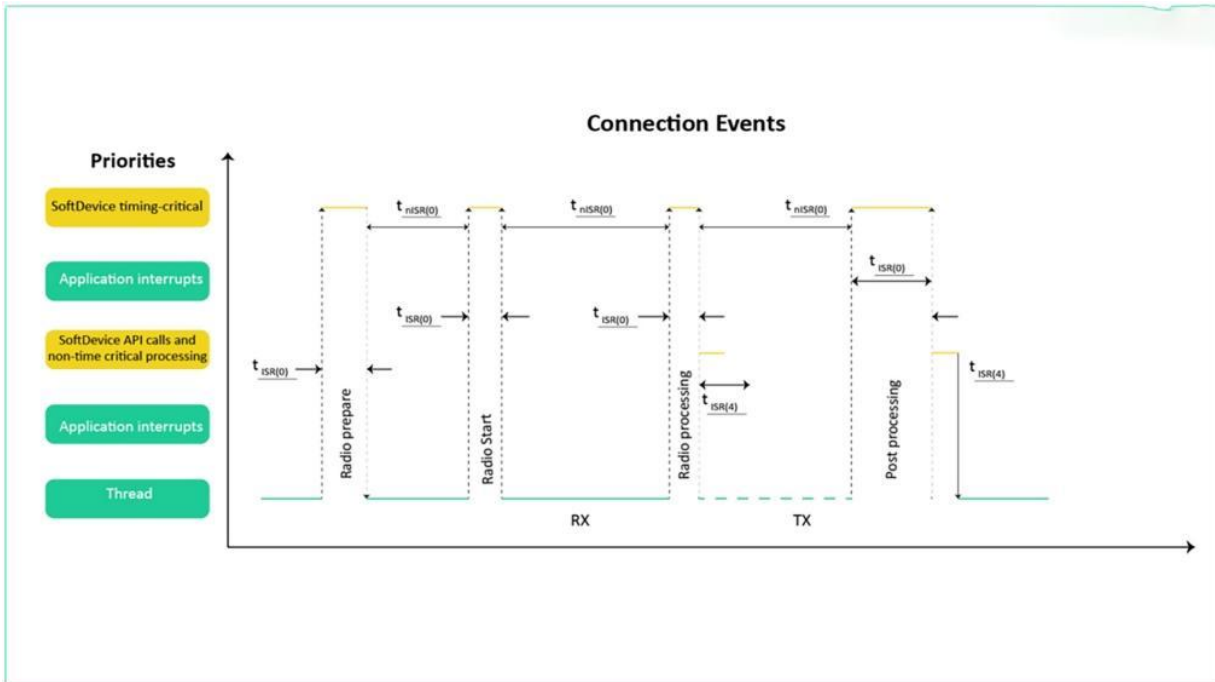
there are 37 RF channels utilized for transmitting data packets during a connection. However, not all 37 channels are necessarily used during a connection.



Channel hopping is used in Bluetooth to reduce interference and improve throughput. Use of the unlicensed 2.4GHz ISM band by several wireless technologies causes increased interference and results in retransmissions to correct errors in received packets. .

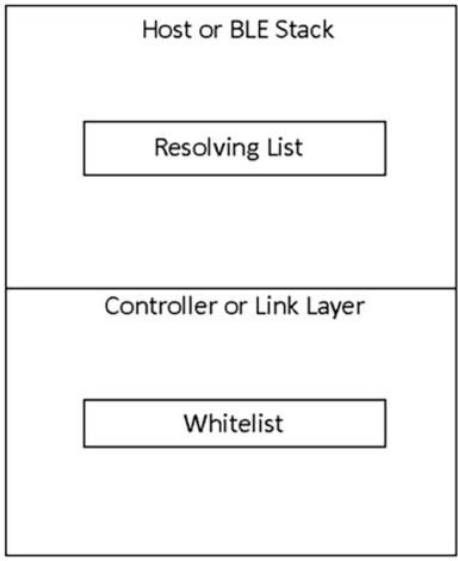


Since BLE is a low energy oriented protocol, it is more susceptible to interference. BLE uses channel hopping to combat the impact of interference. When one channel is completely blocked due to interference, devices can still continue to communicate with each other on other channels. A connected device changes to a new data channel. For every connection, even a connection event is a sequence of data packet exchanges between two connected devices.

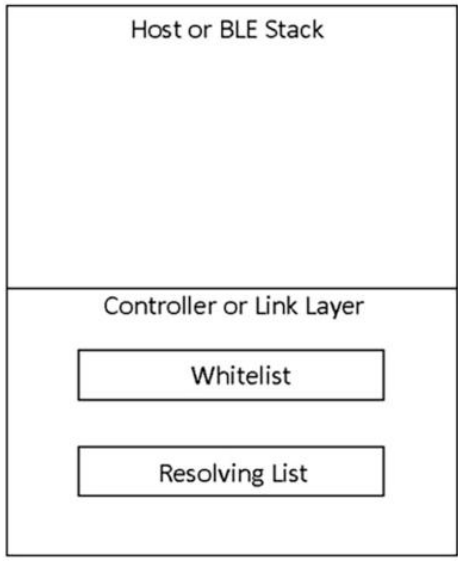


The connection events occur periodically with an interval called connection. In all, the packets within a connection event are transmitted on the same day that's entered. A new connection event uses a new data standard.

Bluetooth 4.1



Bluetooth 4.2



Finally, let's look at whitelist and device filtering. A white list is a list of addresses and address types of specific devices. It is used for determining

which devices a particular device is interested in an entry for an anonymous device and allows matching all advertisements sent with no address. BLE supports device filtering for procedures related to: the advertising state, the scanning state, and the initiating state (for establishing connections)

Device Filtering gets processed at the link layer in the controller (the lower layer of the Bluetooth stack), Device Filtering gets processed at the link layer in the controller (the lower layer of the Bluetooth stack), which saves time and overhead from being performed at the host (the upper layer of the stack). However, the host is responsible for configuring the white list.

This filter policy defines how the advertiser processes both scan and connection requests. The different configurations include:

- Process scan and connection requests only from devices in the white list.
- Process scan and connection requests from all devices (white list not used).
- Process scan requests only from devices in the white list, while processing connection requests from all devices.
- Process connection requests only from devices in the white list, while processing scan requests from all devices.

- **Scanning State Filter Policy** (central side)

This filter policy defines how the scanner processes advertising packets. The different configurations include:

- Process advertising packets from all devices (white list not used).
- Process advertising packets only from devices in the white list.

- **Initiating State Filter policy** (central side)

This filter policy defines how a connection initiator processes advertising packets. The different configurations include:

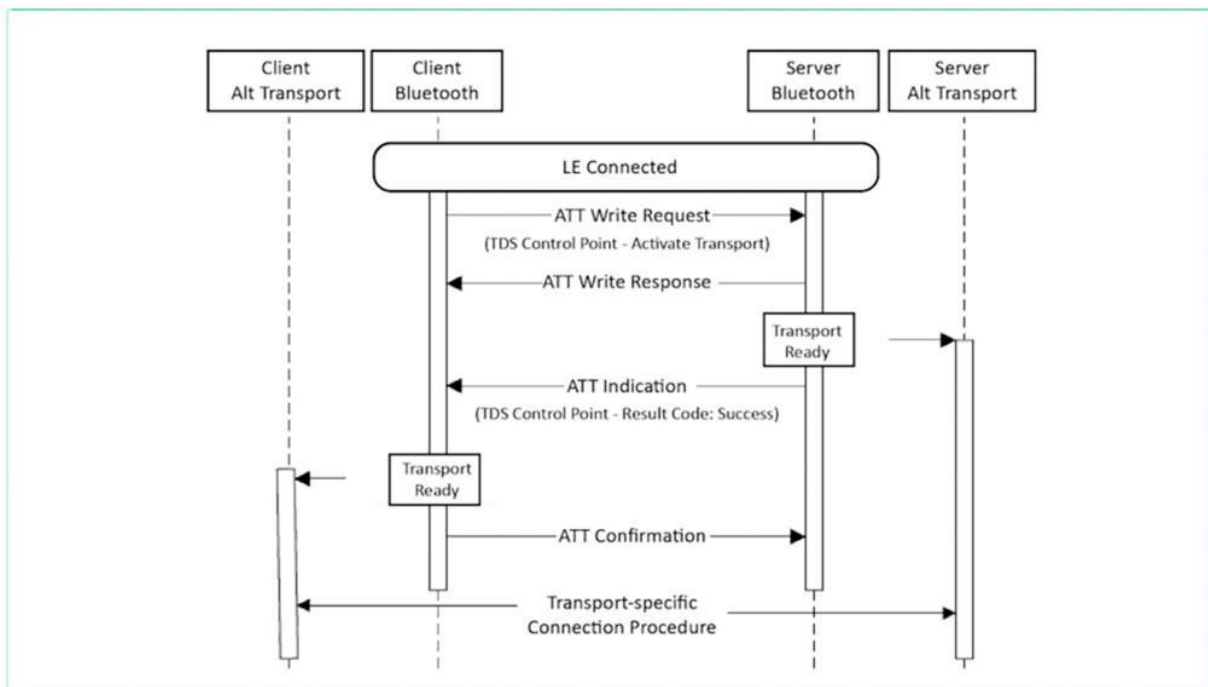
- Process and initiate a connection to all devices listed in the white list.
- Process and initiate a connection only to a device specified by the host.

CHARACTERISTICS, PROFILES, AND ATTRIBUTES IN BLE

Packet format for Uncoded LE data packets

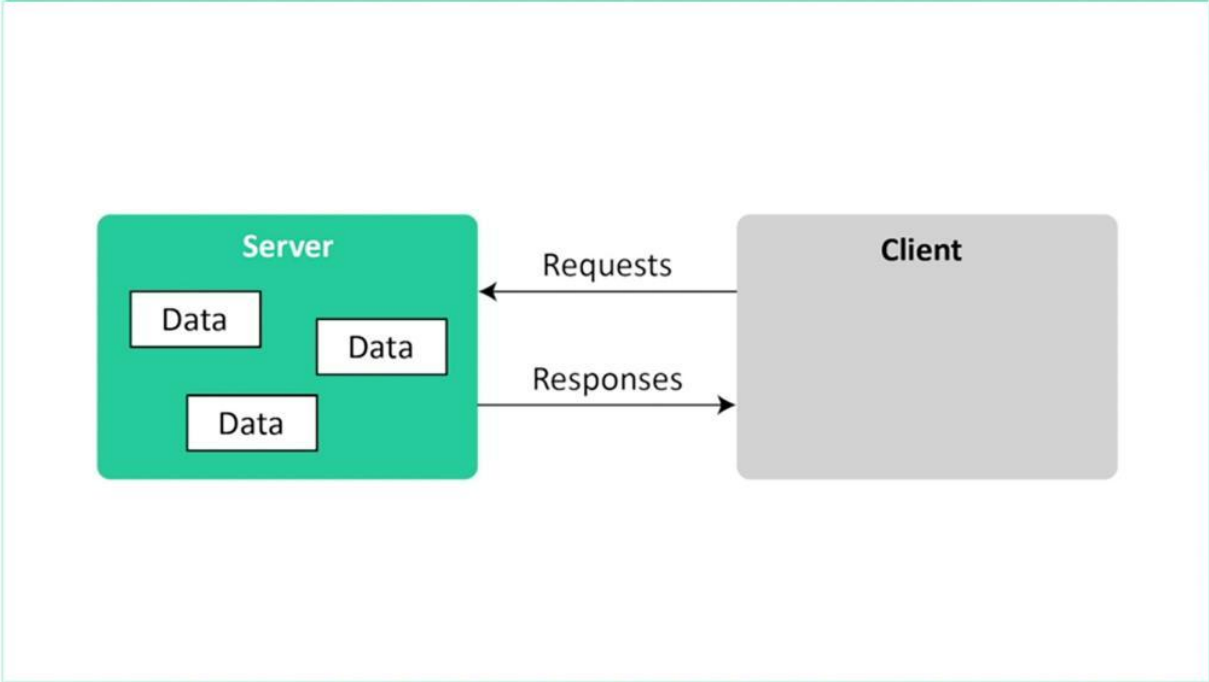
Preamble	Access Address	PDU (2-257 bytes)				CRC	
1 byte (1M PHY) 2 bytes (2M PHY)	4 bytes	LL Header	Payload (0-251 bytes)			MIC (Optional)	3 bytes
		2 bytes	L2CAP Header	ATT Data (0-247 bytes)			
			4 bytes	ATT Header	ATT Payload		
				Op Code	Attribute Handle	Up to 244 bytes	
			1 byte	2 bytes			

Attribute Protocol defines how a server exposes its data to a client and how this data is structured.

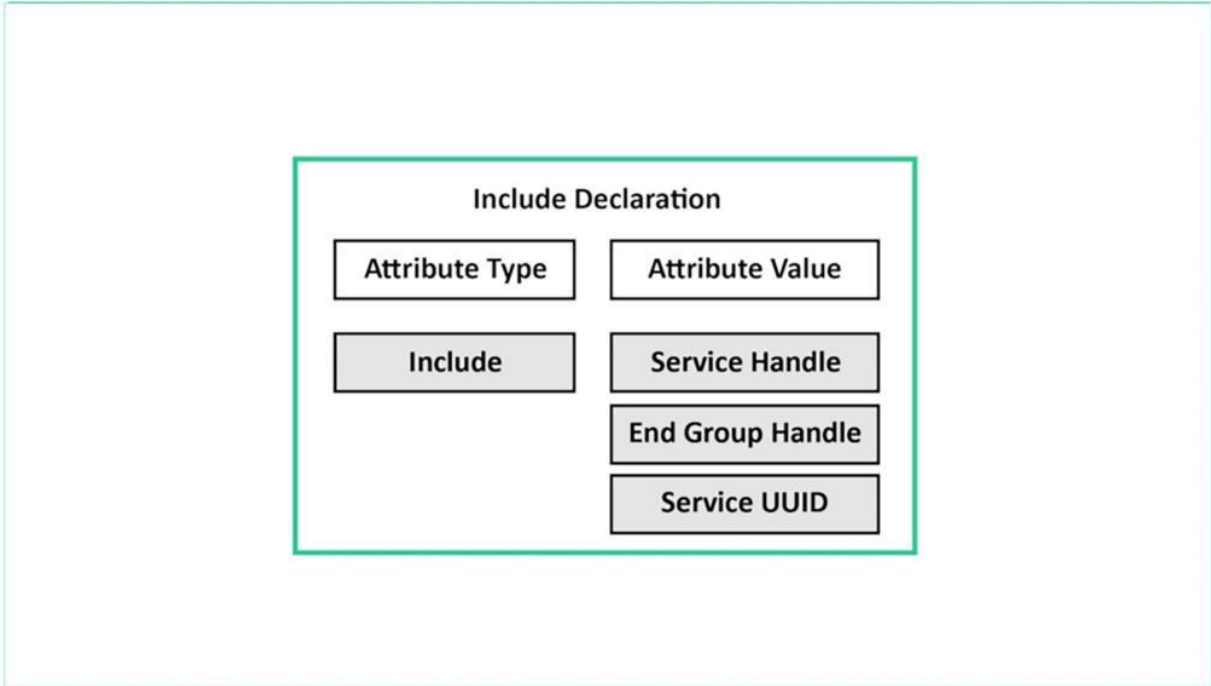


There are two roles within the ATT: Server: It is the device that exposes the data it controls or contains and some other aspects of server behavior that other devices may control. It is the device that accepts incoming commands

from a peer device & sends responses, notifications, and indications. It is the device that accepts incoming commands from a peer device & sends responses, notifications, and indications.



Client - This is the device that interfaces with the Server to read the server's exposed data and/or controlling the server's behavior. Client - This is the device that interfaces with the Server to read the server's exposed data and/or controlling the server's behavior.the servers behavior. It is the device that sends commands & requests and accepts incoming notifications and indications.



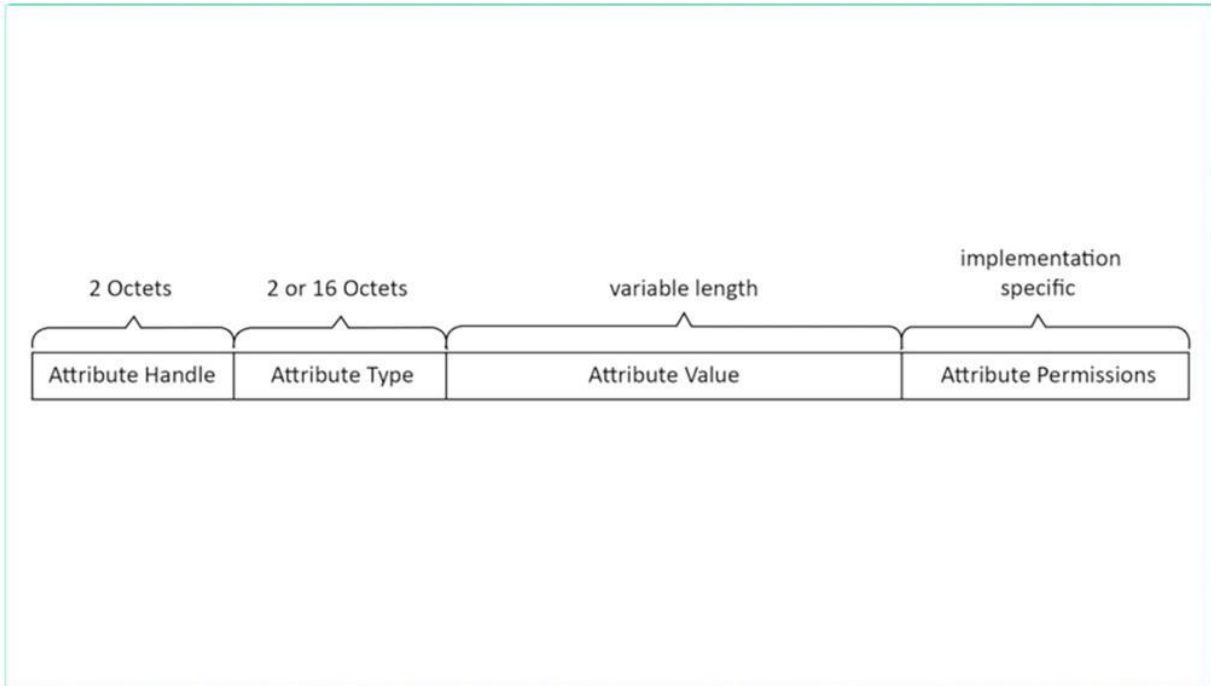
The data that the Server exposes is structured as attributes. An attribute is the generic term for any type of data exposed by the Server and it defines this data structure.

Attribute Handle	Attribute Type	Attribute Value			Attribute Permissions
0xNNNN	0x2803-UUID for <<Characteristic>>	Characteristic properties	Characteristic Value	Characteristic UUID	Read only, No Authentication, No Authorization

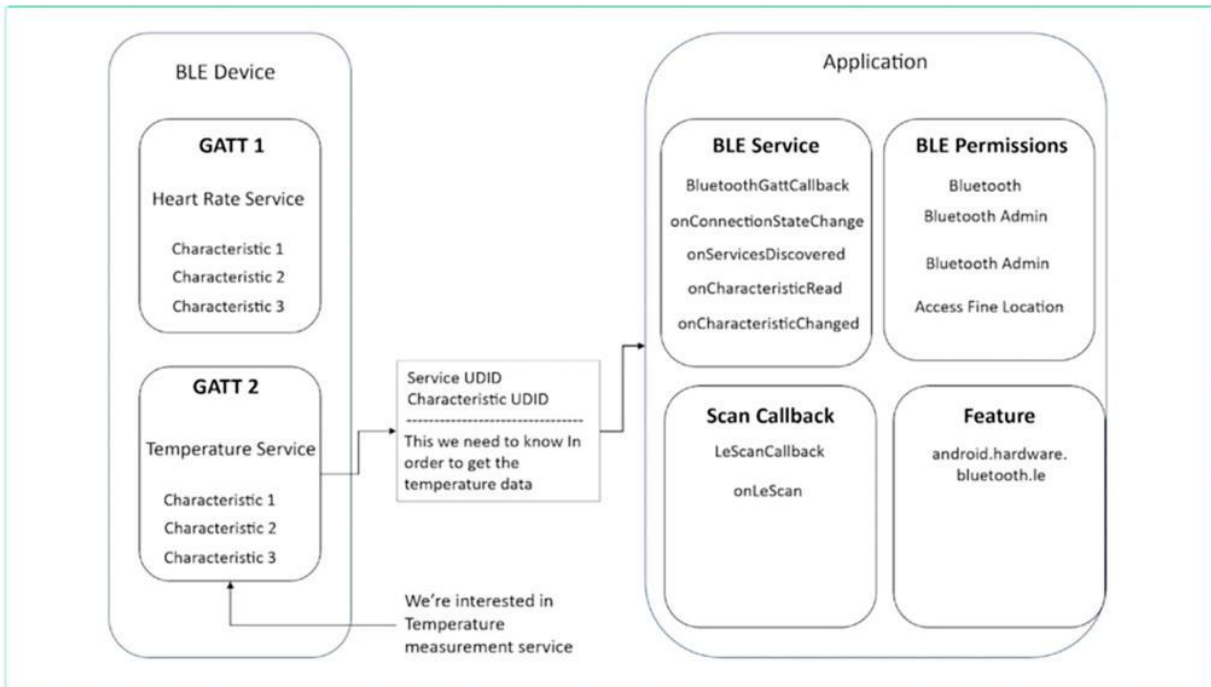
For example, services and characteristics are types of attributes. Attributes are made up of the following: For example, services and characteristics are

types of attributes. Attributes are made up of the following:

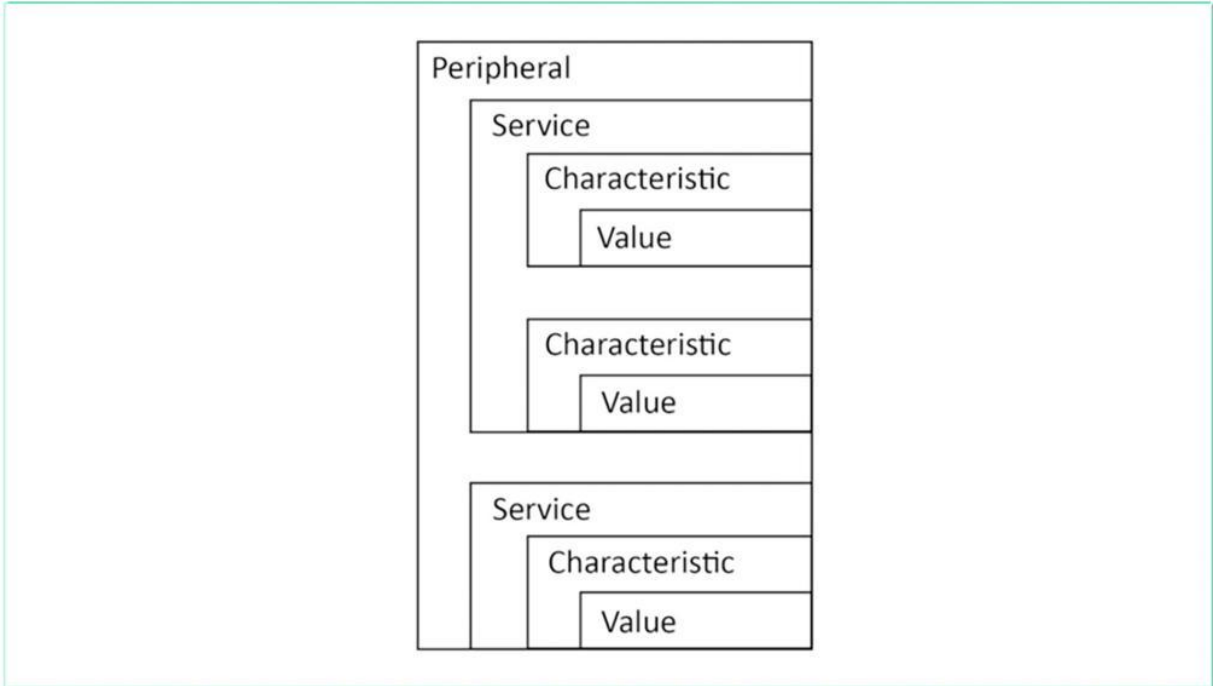
- Attribute type (Universally Unique Identifier or UUID): This is a 16-bit number in the case of Bluetooth SIG-Adopted Attributes or a 128-bit number in the case of vendor-specific UUID
- Attribute type (Universally Unique Identifier or UUID): This is a 16-bit number in the case of Bluetooth SIG-Adopted Attributes or a 128-bit number in the case of vendor-specific UUID
- Attribute Handle. It is a 16-bit value that the Server assigns to each of its attributes. This Value is used by the client to reference a specific attribute
- Attribute Handle. It is a 16-bit value that the Server assigns to each of its attributes. This Value is used by the client to reference a specific attribute
- Attribute Handle. It is a 16-bit value that the Server assigns to each of its attributes. This Value is used by the client to reference a specific attribute
- Attribute Permissions. It determines whether an attribute can be read or written to. Whether it can be notified or indicated & what security levels are required for each of these operations.
- Attribute Permissions. It determines whether an attribute can be read or written to. Whether it can be notified or indicated & what security levels are required for each of these operations.
- Attribute Permissions. It determines whether an attribute can be read or written to. Whether it can be notified or indicated & what security levels are required for each of these operations.
- Attribute Permissions. It determines whether an attribute can be read or written to. Whether it can be notified or indicated & what security levels are required for each of these operations.
- Attribute Permissions. It determines whether an attribute can be read or written to. Whether it can be notified or indicated & what security levels are required for each of these operations.



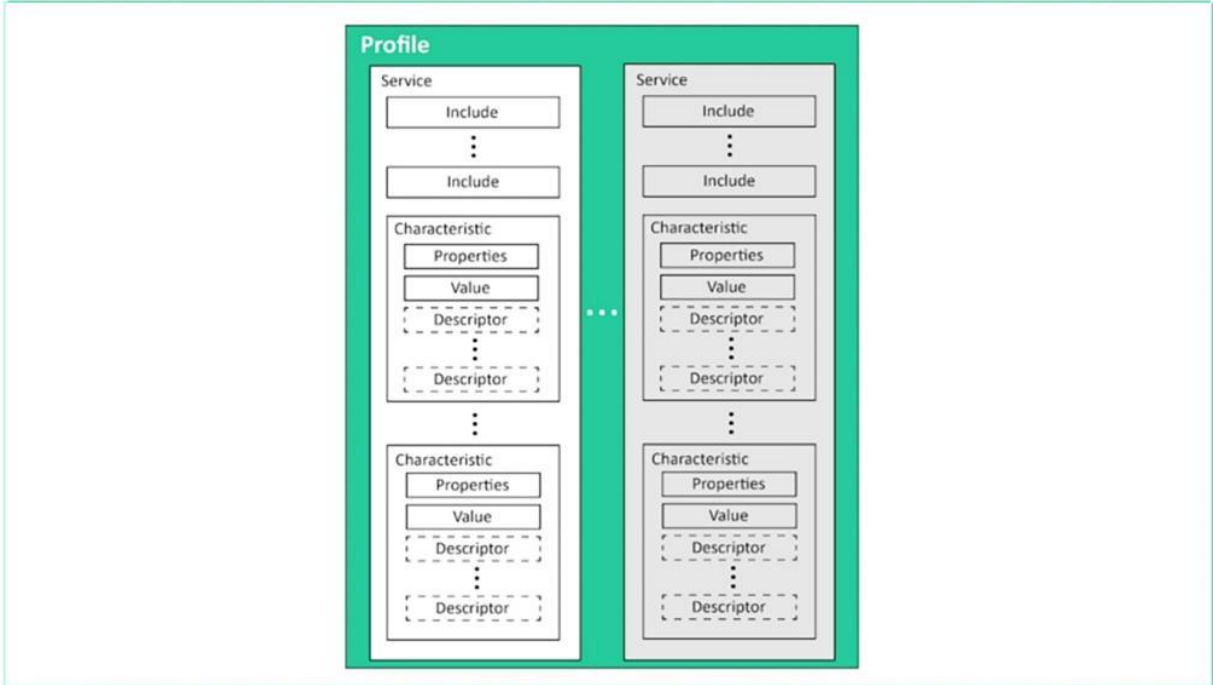
This figure shows a logical representation of an Attribute. Now let us look at the generic attribute profile target. Now let us look at Generic Attribute Profile or GATT.



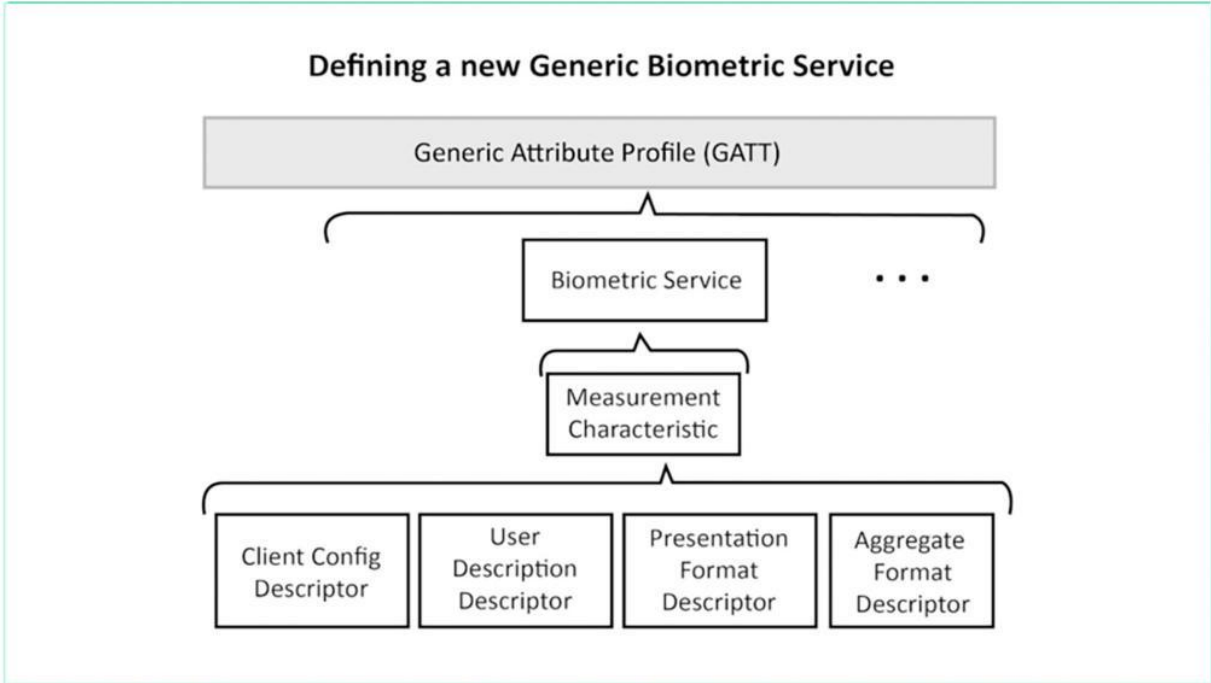
Now that we've covered the concept of attributes, we will go over 3 important concepts in BLE:



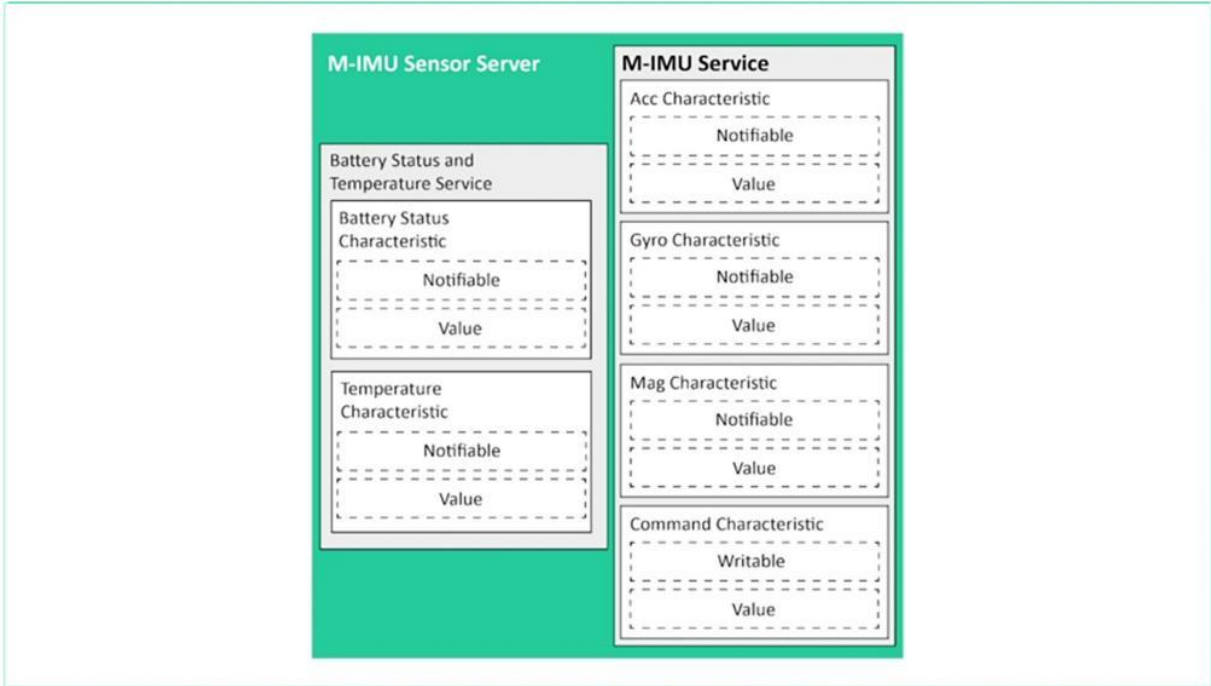
Services, Characteristics and Profiles These concepts are used specifically to allow hierarchy in the structuring of the data exposed by the Server. These concepts are used specifically to allow hierarchy in the structuring of the data exposed by the Server.



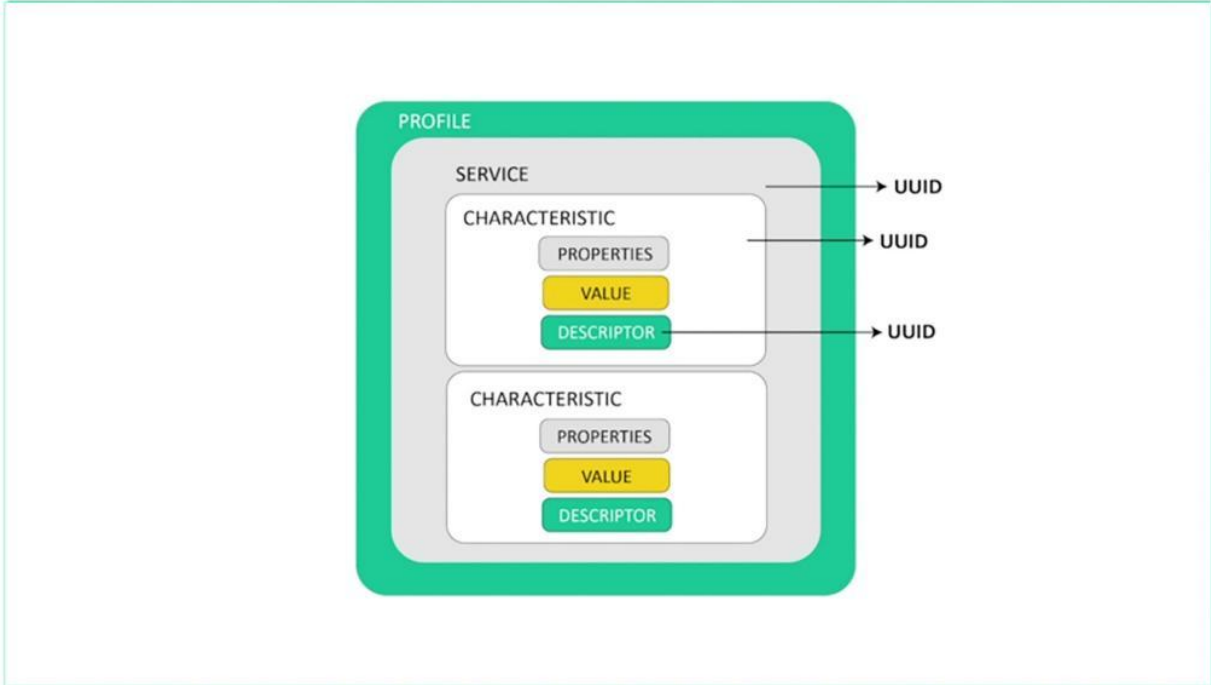
The GATT defines the format of services and their characteristics and the procedures that are used to interface with these attributes, such as service discovery, characteristic reads, characteristic writes, notifications, and indications. such as service discovery, characteristic reads, characteristic writes, notifications, and indications. writes notifications and indications.



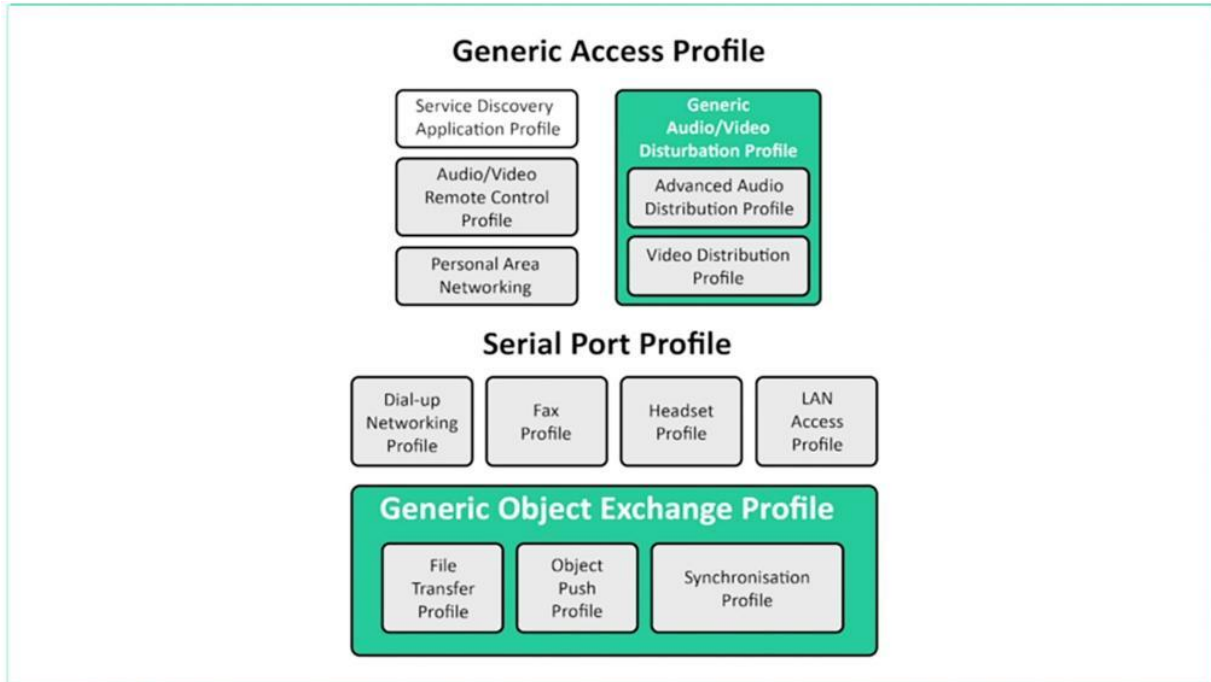
A service is a grouping of one or more attributes, some of which are characteristics. It's meant to group together related attributes that satisfy a specific functionality on the Server. This is how a service looks like of information that the server wants to expose to a client.



For example, the battery level characteristic represents a Bartley's remaining power level in a device that a client can read. The characteristic has other attributes that help define the value it holds. They include properties and descriptions, properties.



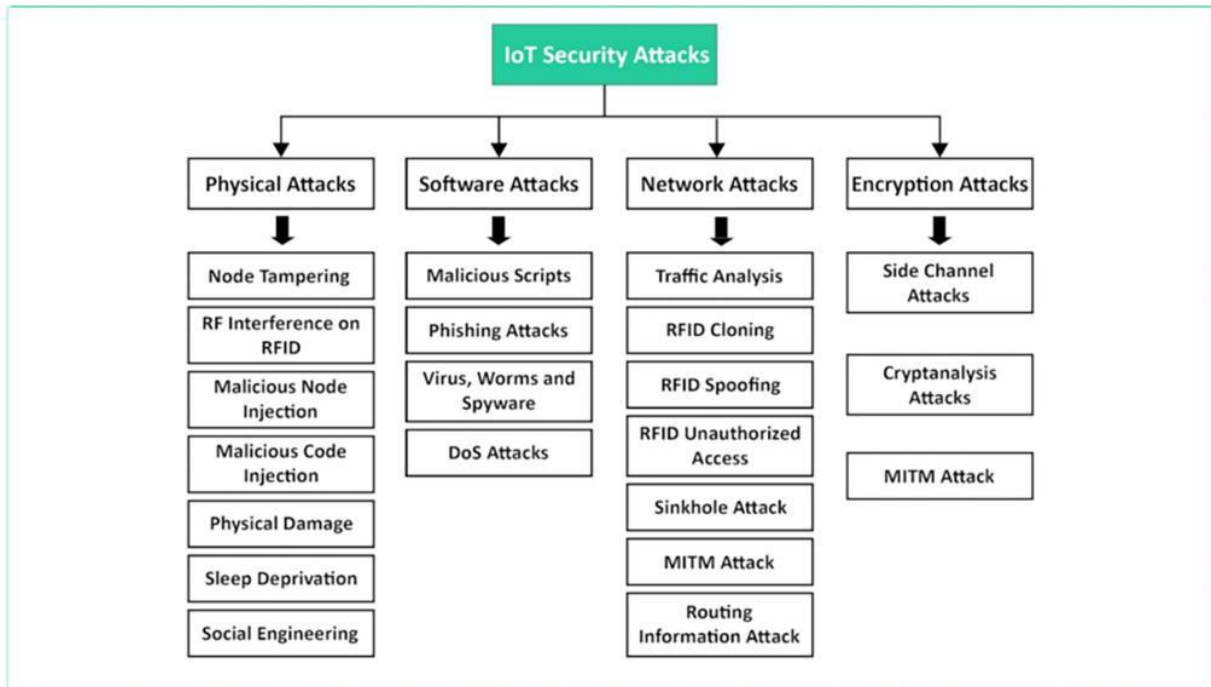
It is represented by a number of objects. It defines how a characteristic value can be used. Some examples include read, write, write with our response, notify indicate descriptors. It is used to contain the latest information about the characteristic value. Some examples include extended properties, user description fields, use for subscribing to notification and indications, and a field that defines the presentation of the value such as the format and the unit of the value.



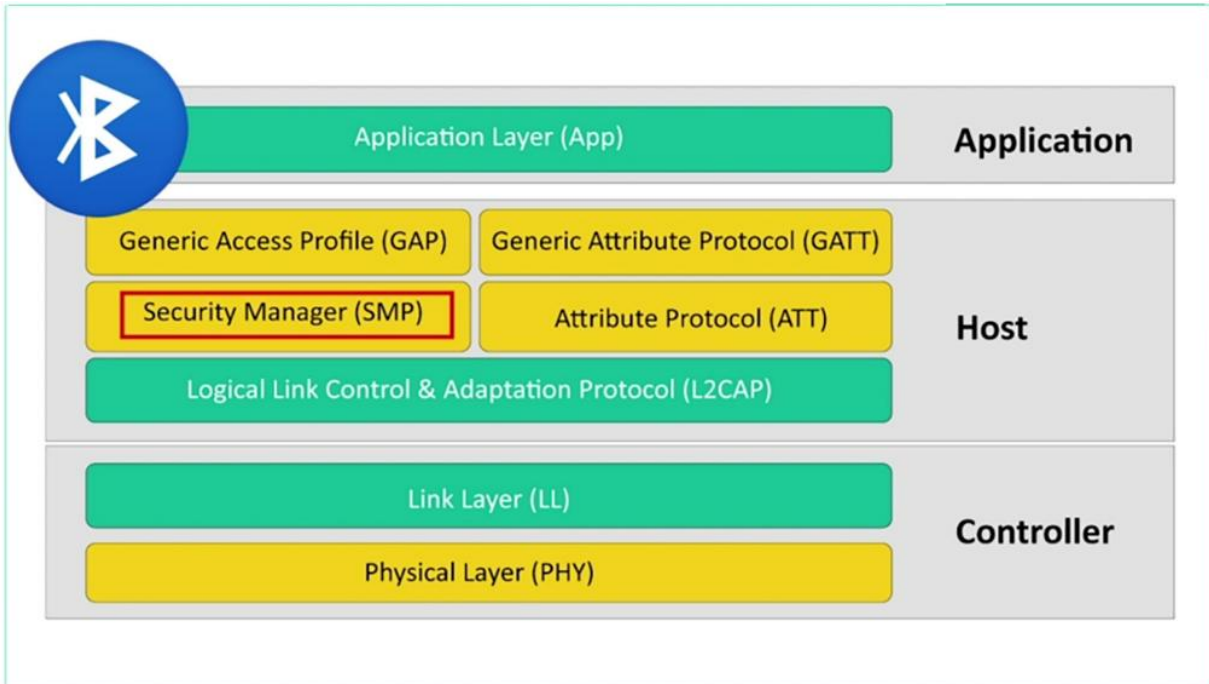
Profiles are much broader in definition than services. They are concerned with defining the behavior of both the client and server when it comes to services, characteristics, and even connections and security requirements. Services and their specifications, on the other hand, deal with the implementation of these services and characteristics on the server-side only.

SECURITY IN BLE PART 1 PAIRING AND BONDING

we will cover the following topics Security Threats and Types of Attacks Security Manager in BLE Pairing Process Bonding Process Security Manager in BLE Pairing Process Bonding Process Security has become one of the most voiced concerns about IoT systems.

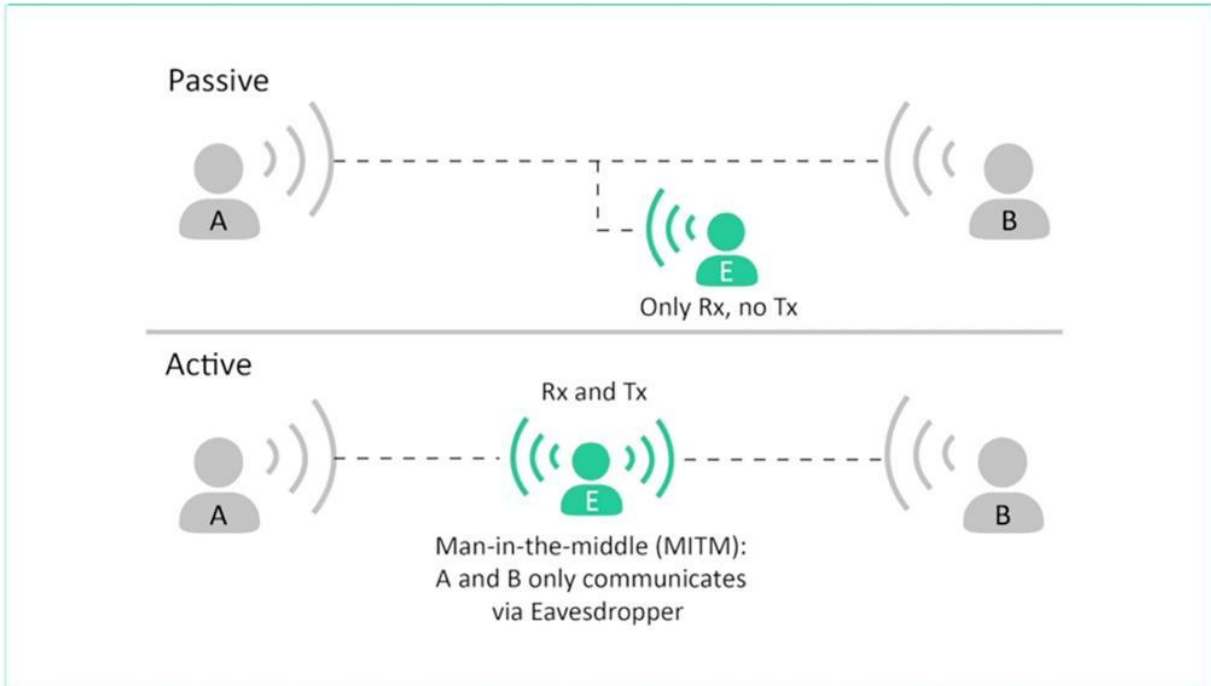


With all the headline news that mentions hacks and vulnerabilities discovered in many IoT products has become one of the major concerns for Iot devices, manufacturers and developers.

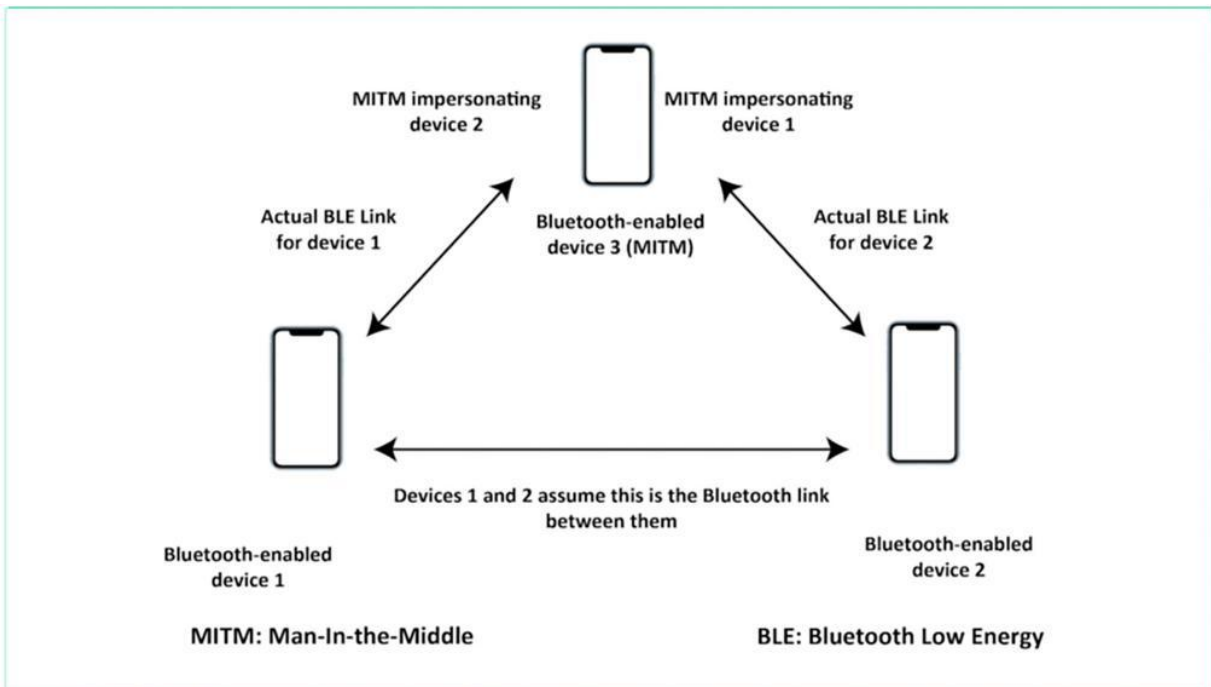


Security in Bali is handled by the security manager layer of the architecture.

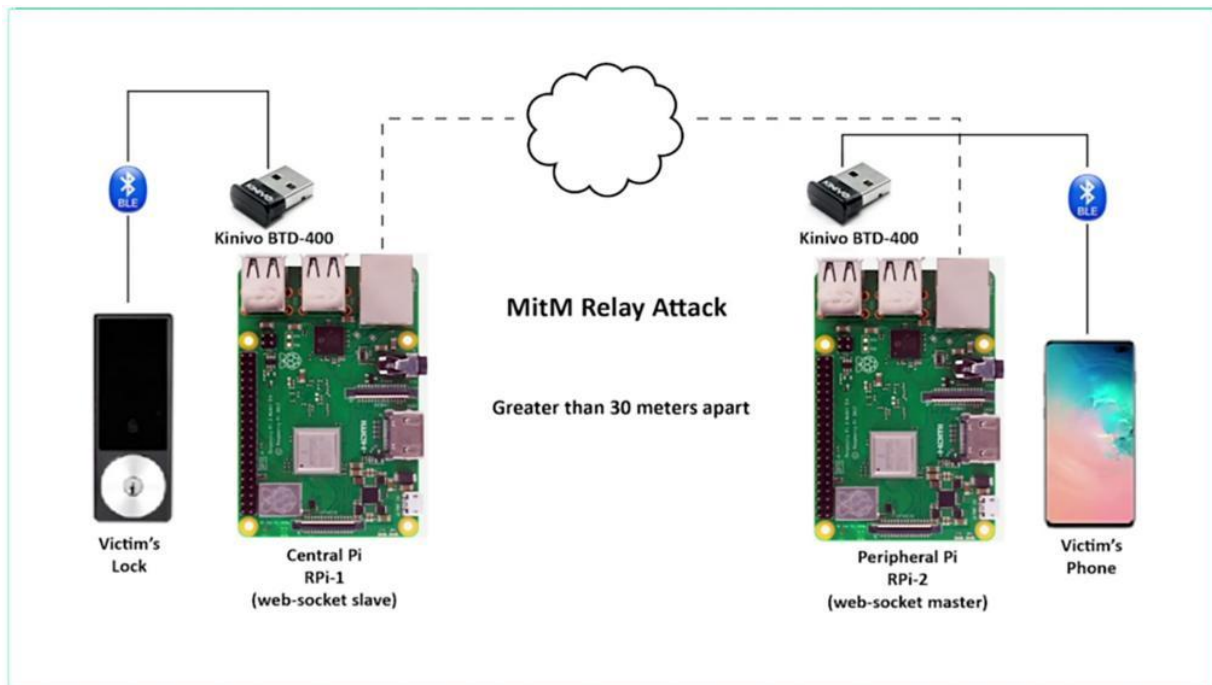
The most common security concerns with any system include one authentication to the integrity of the message received. Three confidentiality of data for privacy. Based on the concerns mentioned above, there are different types of attacks that a malicious device or person may implement.



Let us look at the passive eavesdropping. This describes a malicious device listening in on the communication between two devices and can understand the data. It is usually done by gaining access to the encryption key active eavesdropping.

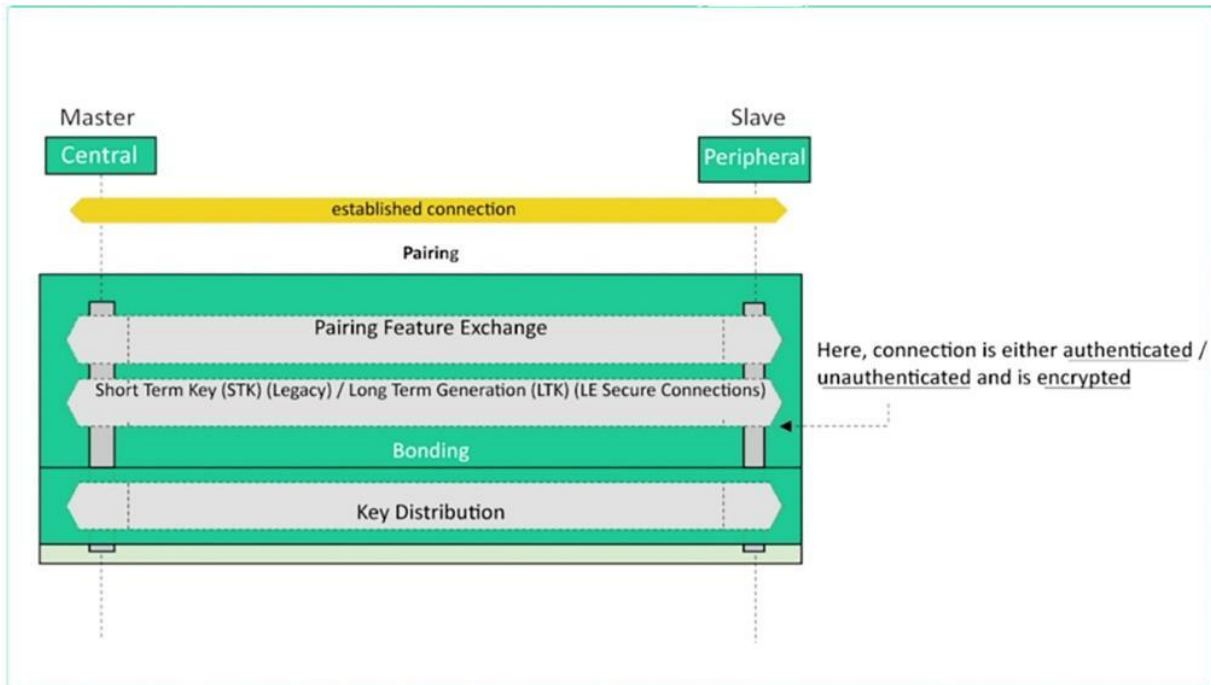


This is also known as a man in the middle attack. In this attack, the malicious device impersonates both the peripheral and the central.

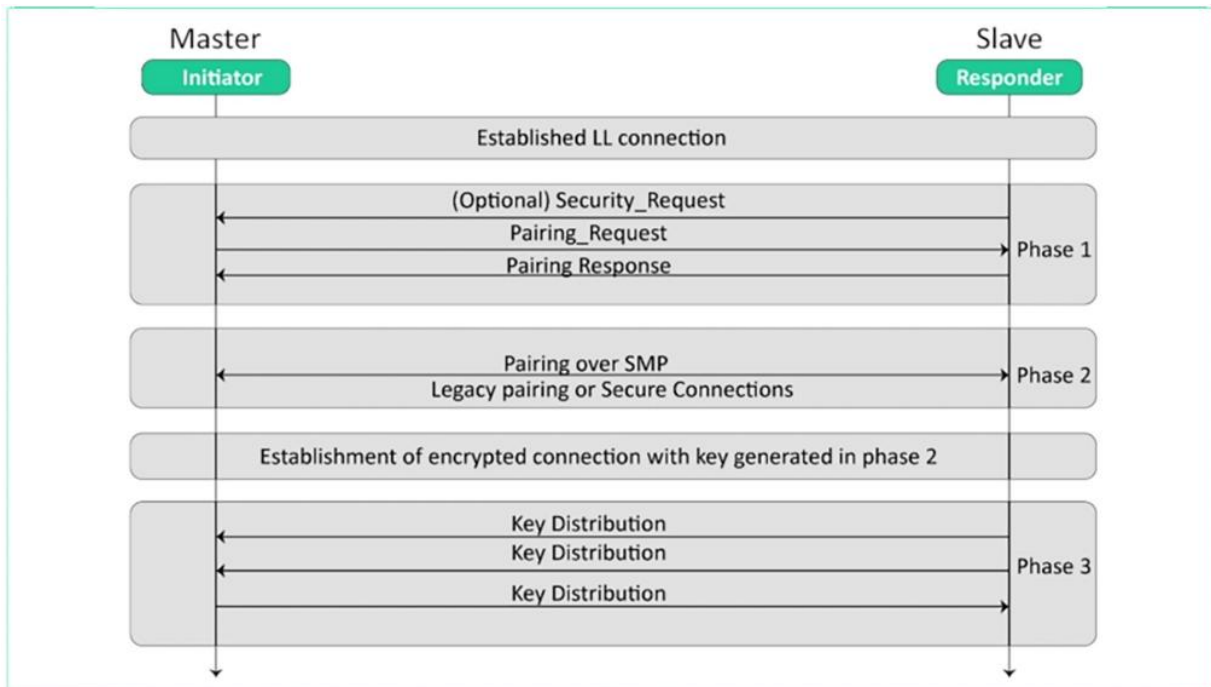


It could then intercept the communication between them and root it so that they do not realize that the attack is happening and possibly even inject data into the packets. Privacy and identity tracking in this attack devices and users are tracked by the Bluetooth address. This can reveal their location and correlated with their behavior. The security manager defines the protocols and algorithms for generating and exchanging keys between two devices. It involves five security features, one bearing. It is a process of creating shared secret keys between two devices to bonding. It is a process of creating and storing shared secret keys on central and peripheral devices for use in subsequent connections between the devices. Three Authentication. It is a process of verifying that the two devices share the same secret keys for encryption. It is a process of encrypting the data exchanged between the devices encryption and BEARLY uses the one to deal with a IS encryption standard, which is a symmetric algorithm. This means that the same keys used to encrypt and decrypt the data on both sides. Five Message Integrity. It is a process of signing the data and verifying the signature at the other end.

With these five security features, the security manager addresses the different security concerns as follows. One confidentiality via encryption to authentication via pairing and bonding. Three, privacy, whereas all of your private addresses for integrity were digital signatures.

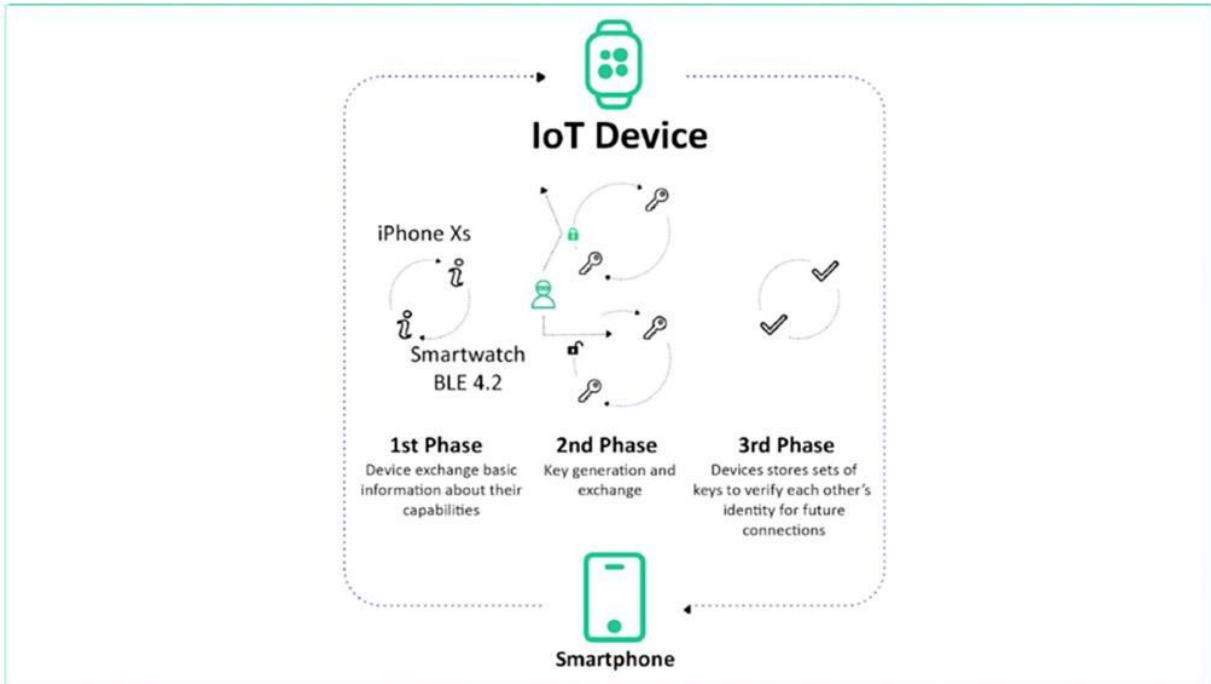


Now let's understand the whole process of security implementation in Bearly in BEARLY. The Master device is the initiator of security procedures. This little device may request the start of a security procedure by sending a security request message to the master. Still, it is up to the master to then send the packet that officially starts the security process to better understand the whole process.

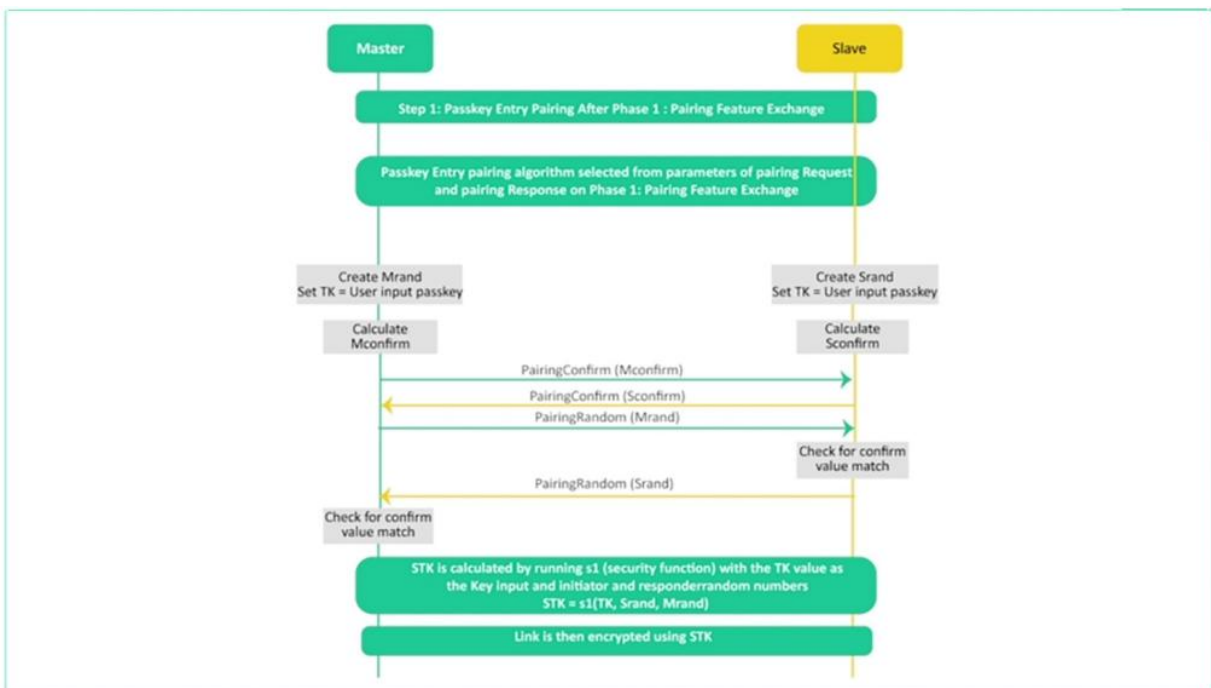


The sequence diagram showing the security process is divided into three phases. In phase one, the slave may request the start of the bidding process. The master initiates the bidding process by sending a pairing request message to the slave, which then responds with a bearing response message. The bearing request and the bearing response messages represent an exchange of the features supported by each device and the security requirements for each device.

These security requirements include authentication requirements, maximum encryption keys and the different security keys each device is requesting to use in phase two. The process may differ depending on which method is used.



At least secure connections are only legacy connections in secure connections. The pairing method does not involve exchanging keys over there between the two devices. Rather, the devices utilize the ecdysiaist protocol to generate a public private keypad. The devices then exchange the public use only and from that generate a shared secret. We call the long term GUI, also known as LDK.



In the case of legacy connections, there are two keys used the temporary chebeague and the short term key SDK the BeeGees used, along with other values exchanged between the two devices to generate the SDK. Phase 3 represents the bonding process This is an optional phase that??s utilized to avoid the need to re-pair on every connection to enable a secure communication channel. The result of bonding is that each device stores a set of keys that can be used in each subsequent connection and allows the devices to skip the pairing phase. These keys are exchanged between the two devices over a link that??s encrypted using the keys resulting from phase two.

SECURITY IN BLE PART 2 SECURITY KEYS, MODES AND LEVELS

1 is concerned with encryption, whereas security mode 2 is concerned with data signing. There are two security modes in BLE: Security mode 1 and security mode 2. Security mode 1 is concerned with encryption, whereas security mode 2 is concerned with data signing. One. There are four levels. Level one, no security. That is, no authentication and no encryption level to unauthenticated bearing with encryption. Level three authenticated bearing with encryption. Level four authenticated a secure connection sparing with encryption and security more two. There are two levels.

```

Bluetooth Security Manager Protocol
  Opcode: Pairing Request (0x01)
  IO Capability: Keyboard, Display (0x04)
  OOB Data Flags: OOB Auth. Data Not Present (0x00)
  AuthReq: 0x00, Secure Connection Flag, MITM Flag, Bonding Flags: Bonding
    0000 .... = Reserved: 0x0
    ...0 .... = Keypress Flag: False
    ....1... = Secure Connection Flag: True
    ....1.. = MITM Flag: True
    ....01 = Bonding Flags: Bonding (0x1)
  Max Encryption Key Size: 16
  Initiator Key Distribution: 0x0f, Link Key, Signature Key (CSRK), Id Key (IRK), Encryption Key (LTK)
    0000 .... = Reserved: 0x0
    ....1... = Link Key: True
    ....1.. = Signature Key (CSRK): True
    ....1.. = Id Key (IRK): True
    ....1.. = Encryption Key (LTK): True
  Responder Key Distribution: 0x0f, Link Key, Signature Key (CSRK), Id Key (IRK), Encryption Key (LTK)
    0000 .... = Reserved: 0x0
    ....1... = Link Key: True
    ....1.. = Signature Key (CSRK): True
    ....1.. = Id Key (IRK): True
    ....1.. = Encryption Key (LTK): True
  
```

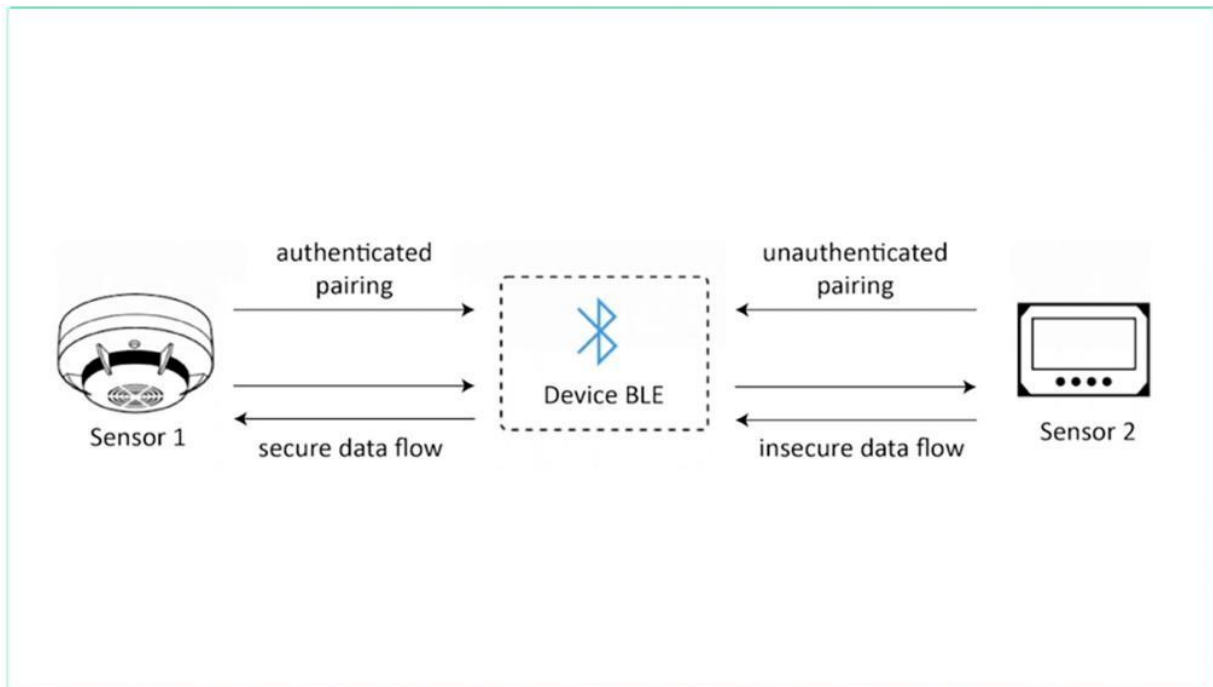
```

Bluetooth Low Energy Link Layer
  Bluetooth L2CAP Protocol
    Length: 65
    CID: Security Manager Protocol (0x0006)
  Bluetooth Security Manager Protocol
    Opcode: Pairing Public Key (0x0c)
    Public Key X: 284496b90d1580a689dd5ed7591554ad37d87b0af7d3f2df...
    Public Key Y: eed448e2acb78cb9198b644eed8b1d8559e7cb67e8ead91d...
  
```

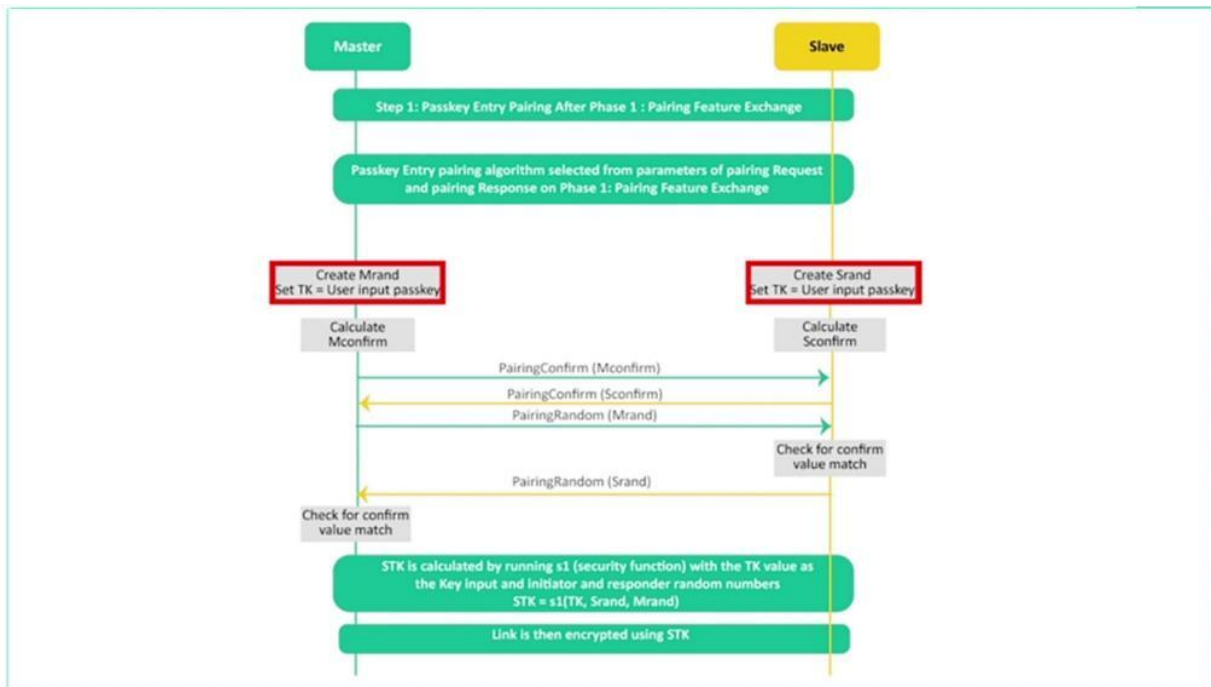
```

Bluetooth Low Energy Link Layer
  Bluetooth L2CAP Protocol
    Length: 65
    CID: Security Manager Protocol (0x0006)
  Bluetooth Security Manager Protocol
    Opcode: Pairing Public Key (0x0c)
    Public Key X: 2f7c29e311dc430d94a64cb57184d589741703b427f7d660...
    Public Key Y: 006a63328ed66cefbeda1bebf5f9cda99a5825b3c0256593...
  
```

Level one unauthenticated pairing with data signing level to authenticate it, bearing that data, saying a link is either considered authenticated or unauthenticated based on the method of pairing used a link between two devices always operates in one security mode only, but it can operate at different levels within the same mode.



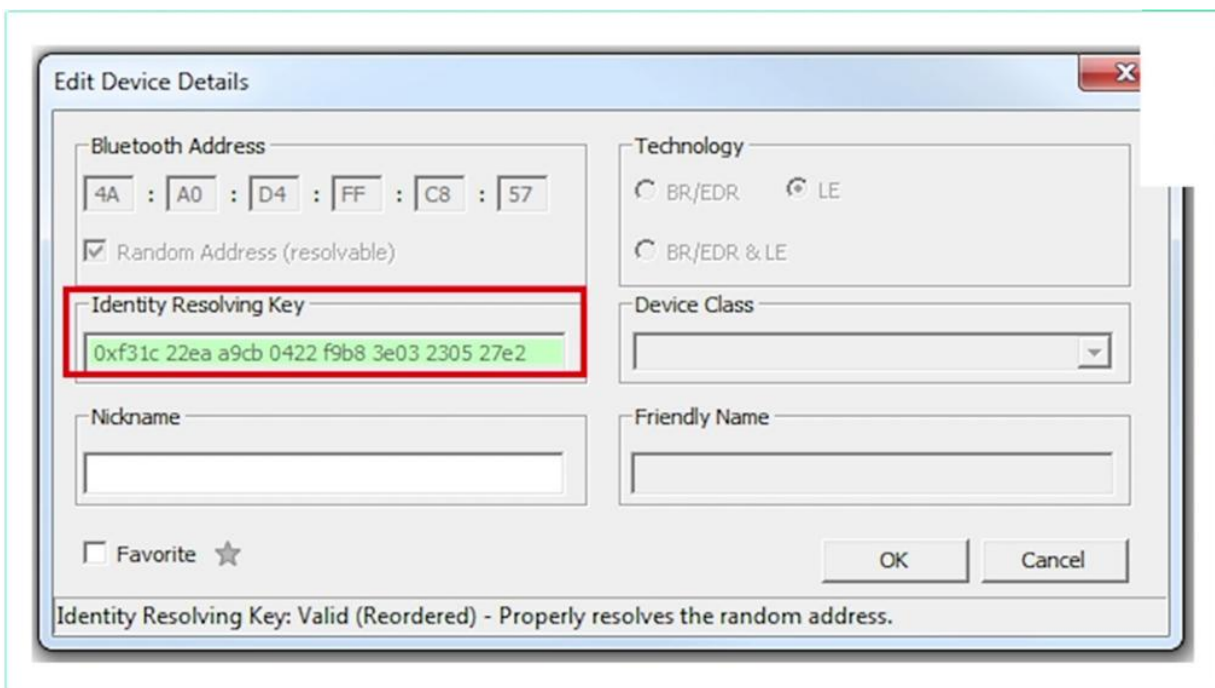
This is because different characteristics may require different levels of security. For example, read access of one characteristic may require level one. The same characteristic for Texas may require level three. Also another characteristic may require level four for both read and write access. Now that the Morton levels are clear, let's do an overview of the different security keys. Houston bearly one Tambra Ricky.



The temporary RPK gets then rated each time the pairing process occurs. The DCS used in legacy connections only to short term key. This key is generated from the ticket exchange between the devices. The Short-Term Key, or SDK, gets generated each time the pairing process occurs and is used to encrypt the data. Throughout the current connection, the SDK is used in legacy connections. Only three long term GUI, also known as LDK. A key gets generated and stored during phase three of the security process in legacy connections and during phase two in a secure connections.



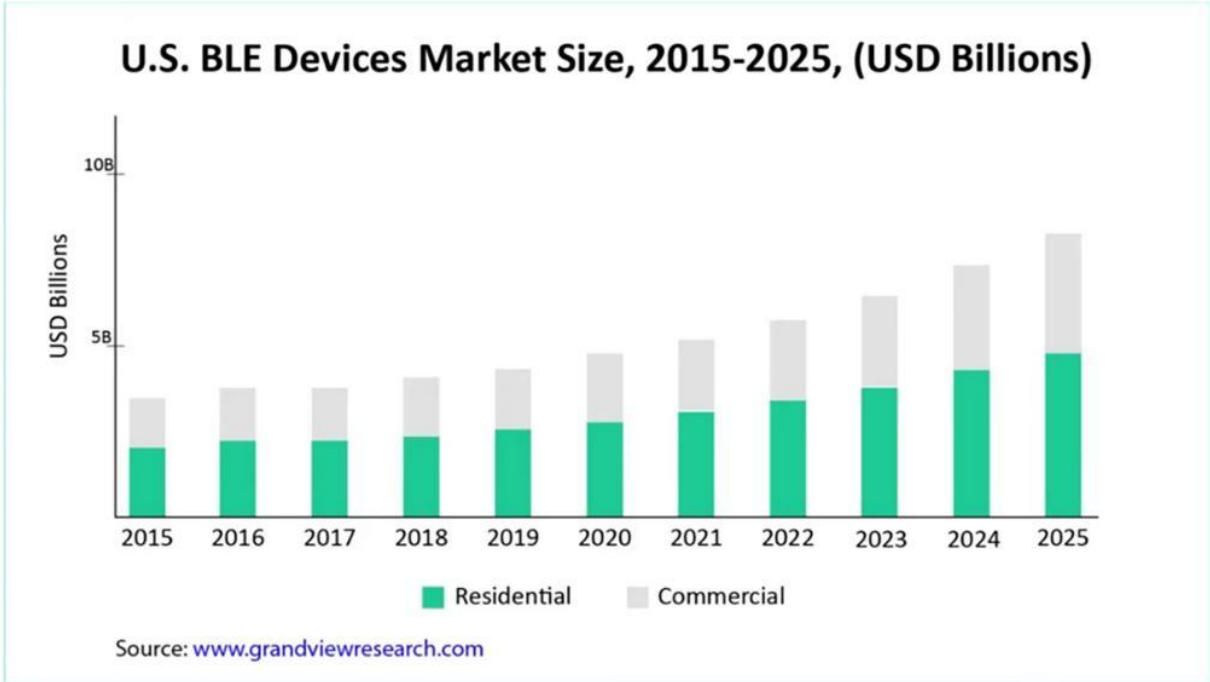
It gets stored on each of the two devices that are bonded and used in subsequent connections between the two devices for Connection Signator Resolving Key, also known as the Yasaka, it is used to assign data and verify the signature attached to the data.



At the other end, this key stored on each of the two bonded devices fight identity resolving key, also known as Ayaka. It is used to resolve random private addresses. This key is unique, but a device so the masters are key will get stored on the slave site and the slaves are key will be stored on the master.

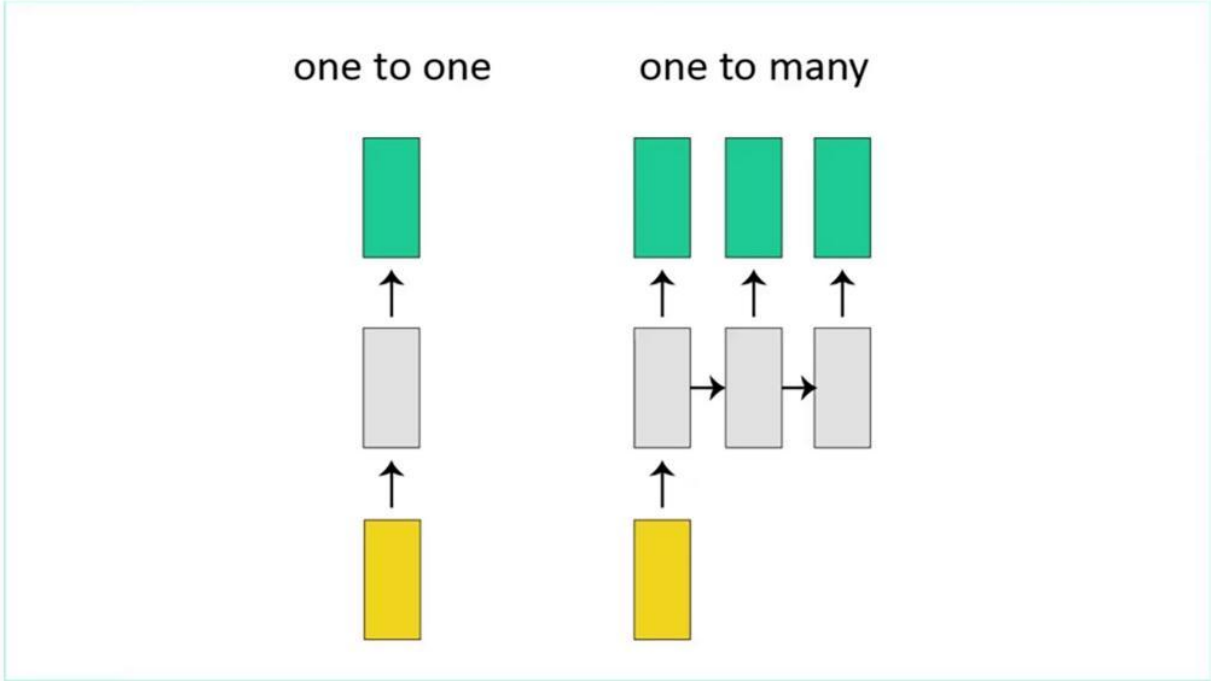
WHAT IS BLE MESH

we will cover the following topics Basics of BLE Mesh Device Types in BLE Mesh Important Features & Advantages

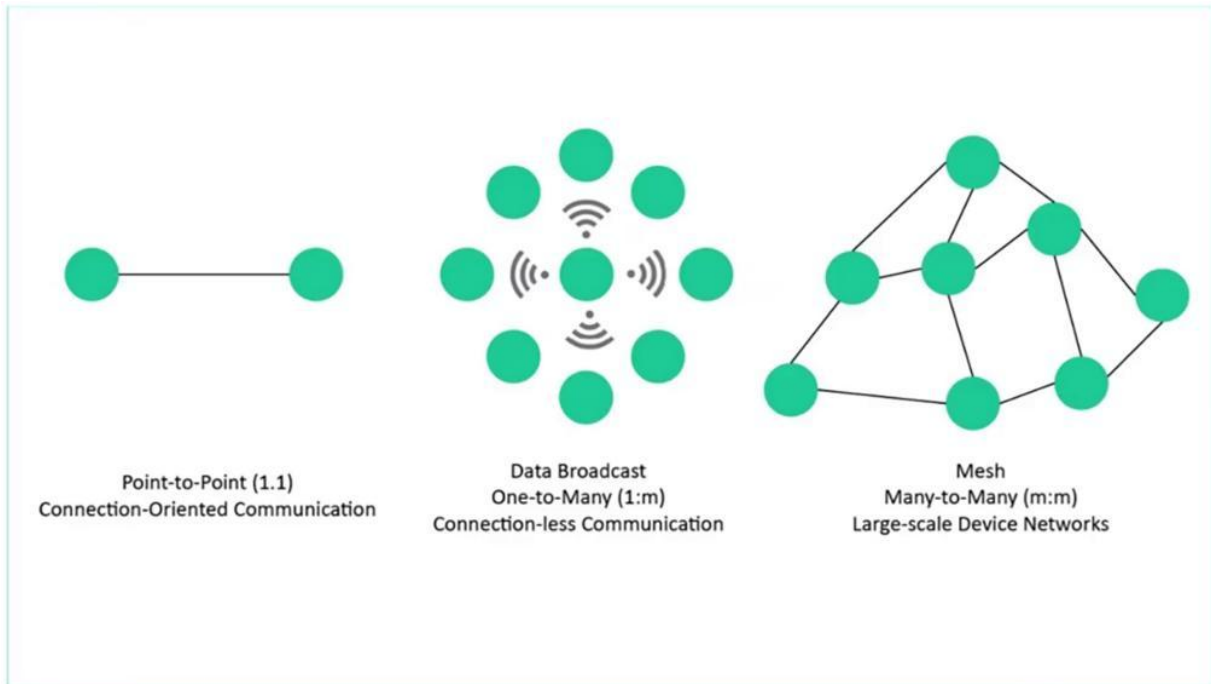


The Bluetooth low energy market was estimated at \$4.9 billion in the year 2020 and is anticipated to grow at 19.7% per year from 2021 to 2025. This

indicates that the popularity of BLE is growing rapidly among the other wireless communication protocols in the IoT domain. The network connection style used in classic Bluetooth is a traditional point-to-point connection that looks like the hub-and-spoke model known as a star topology. Devices can only talk to the central controller and not directly to each other.



BLE was capable of one-to-one and one-to-many topology applications with limited range coverage till July 2017.

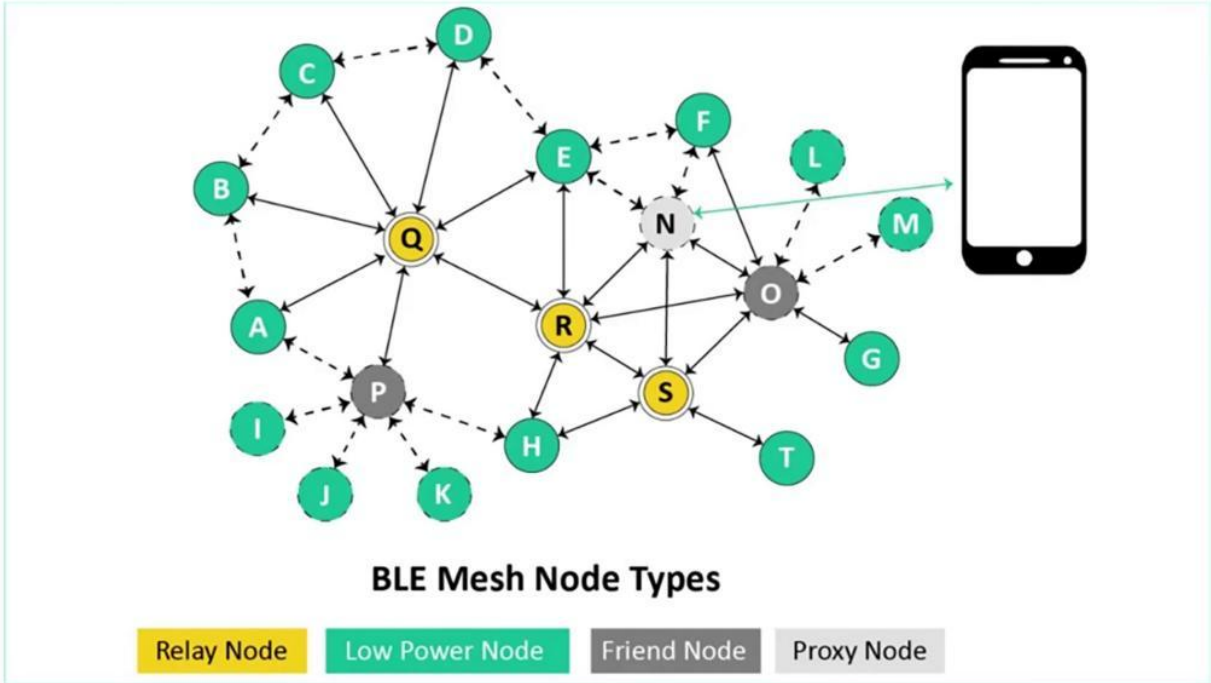


With the addition of mesh topology in July 2017, the BLE applications are further extended to many-to-many topologies for industries where a wide range of power-efficient sensor networks are required. BLE Mesh is a many-to-many style of communication with provisions for message relay from one device to another to create a flexible mesh topology with multiple potential paths between nodes to ensure message delivery. to another to create a flexible mesh topology with multiple potential paths between nodes to ensure message delivery.

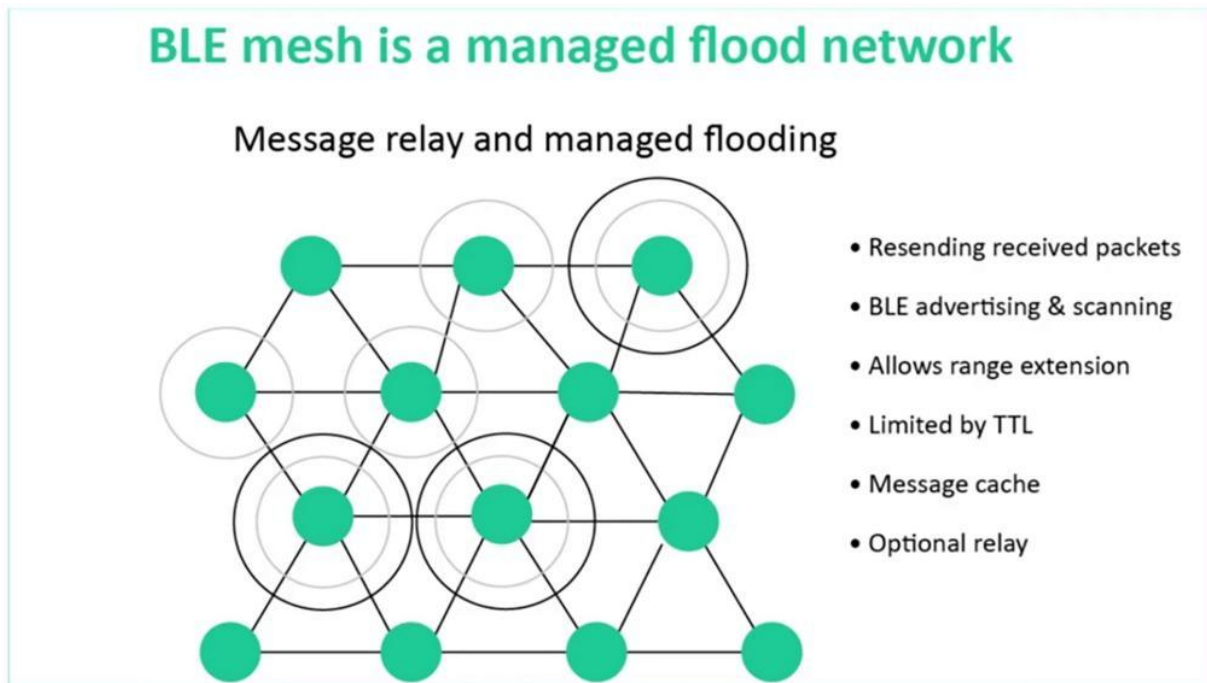
Mesh functionality is a vital update for new applications in lighting, sensor networking, predictive maintenance, asset tracking, and positioning. Mesh functionality is a vital update for new applications in lighting, sensor networking, predictive maintenance, asset tracking, and positioning.



Now let's look at how BLE Mesh works. The first process of any BLE Mesh network is to add devices to the network. This process is called provisioning, and the device, which helps a new device to get connected, is called a provisioner. The provision of each device connected to the mesh network is called the node. The nodes use messages to communicate with each other.



There are four types of nodes in any mesh network. They are as follows one relay node. This node relays messages for others. It acts as an intermediate node, which helps mesh networks to achieve large coverage and reliability to proxy node. This node is an intermediate and that enables the communication between mesh networks and be alienable devices that do not have mesh capabilities. Three Low power node. This is a power sensitive node that can skip being active all the time and wake itself as a program and receive messages at an active time. For friend node, this node has a friendship relation with Lupul. A. A message for a low power node is sent to a related friend which stores messages for the low power node while it's asleep. When the loop our node becomes active, it receives messages from related friend node.

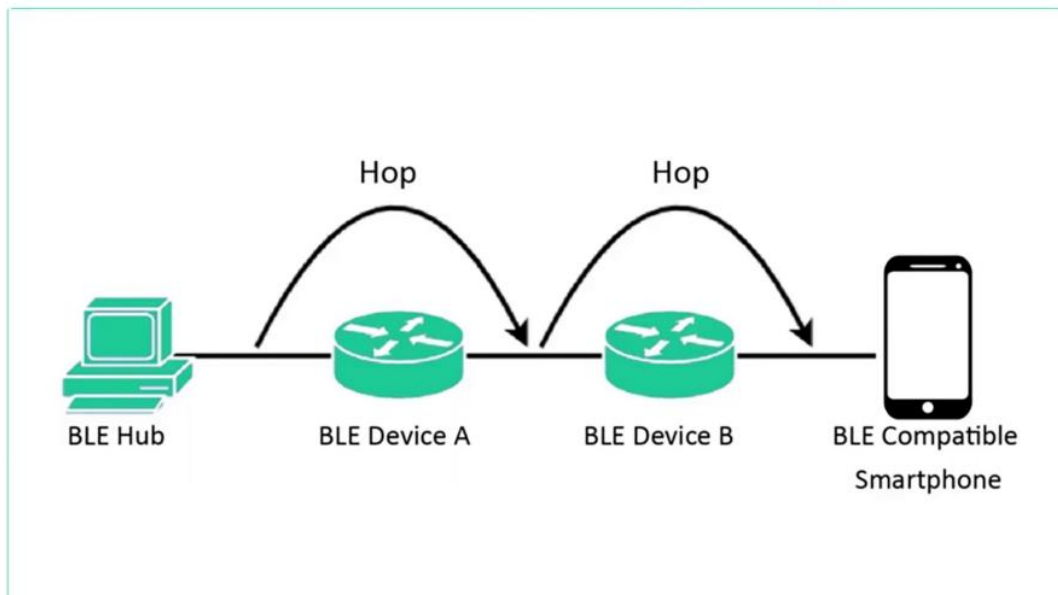


There are different methods of distributing data across mesh networks. Bearly mesh users manage flooding mesh, which is a simple and reliable approach to distribute messages in larger networks. Reliability is ensured with multiple parts from source to destination, and there is no single point of failure. The reliability of message delivery is ensured in bearly through the following three features one heartbeat.

Each node sends heartbeat messages periodically to inform other nodes that it is active. And it also helps recipient nodes to calculate the number of jobs

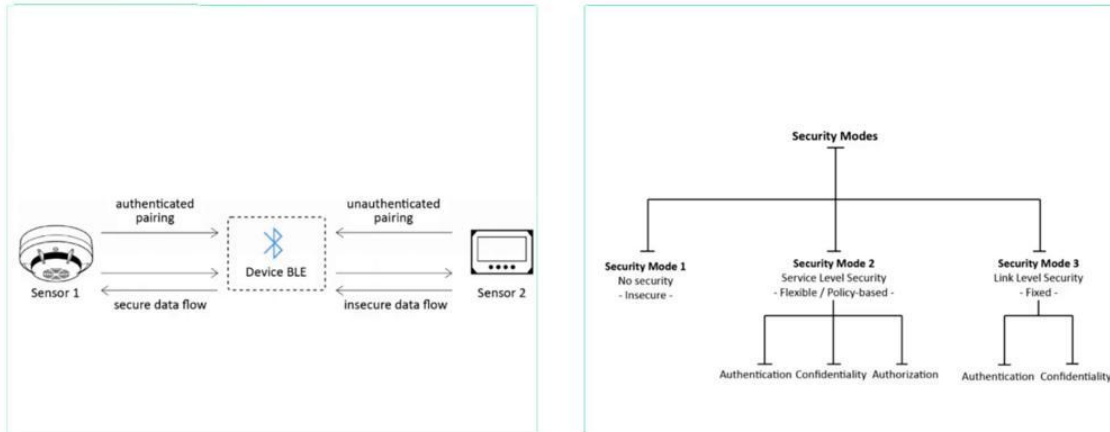
required to send messages to the heartbeat source node to detail you.

Time-to-Live (TTL)



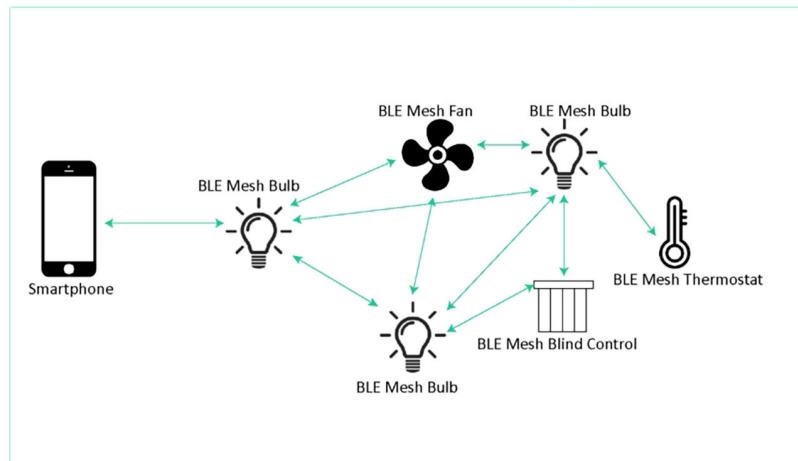
Each message contains a time to live, also known as DTL Value, that restricts the number of times the message can be relayed in the network. Each time the messages are received and relayed the detailed values DiClemente by one heartbeat message helps the source node to set the correct DPL value three message Geshe. Each node contains a message cache for recently received messages. Whenever a node receives the message, it takes the locally cached list of relayed messages and ignores if already present. Finally, let's look at why be element enabled solutions are preferred. One Global interoperability. The products developed by one vendor works with products manufactured by another vendor seamlessly without additional efforts to lead backward compatibility with the help of the proxy nodes. The element is backward, compatible with our bearly enabled devices that do not support mesh. Protocol three, Cost-Effectiveness more smartphone support bearly 4.0 or 5.0. Hence bearly mesh enabled solutions can easily leverage mobile phones without needing customers to purchase additional hardware devices for end to end security.

End-to-End Security



In BLE mesh, security is mandatory and can't be switched off. It provides complete process security design starting from the provisioning process to processing messages. The processing messages fight poverty, efficiency being limited, uses Bluetooth, low energy, even the friendship concept to keep the power consumption optimized compared to other technologies.

Extended Range

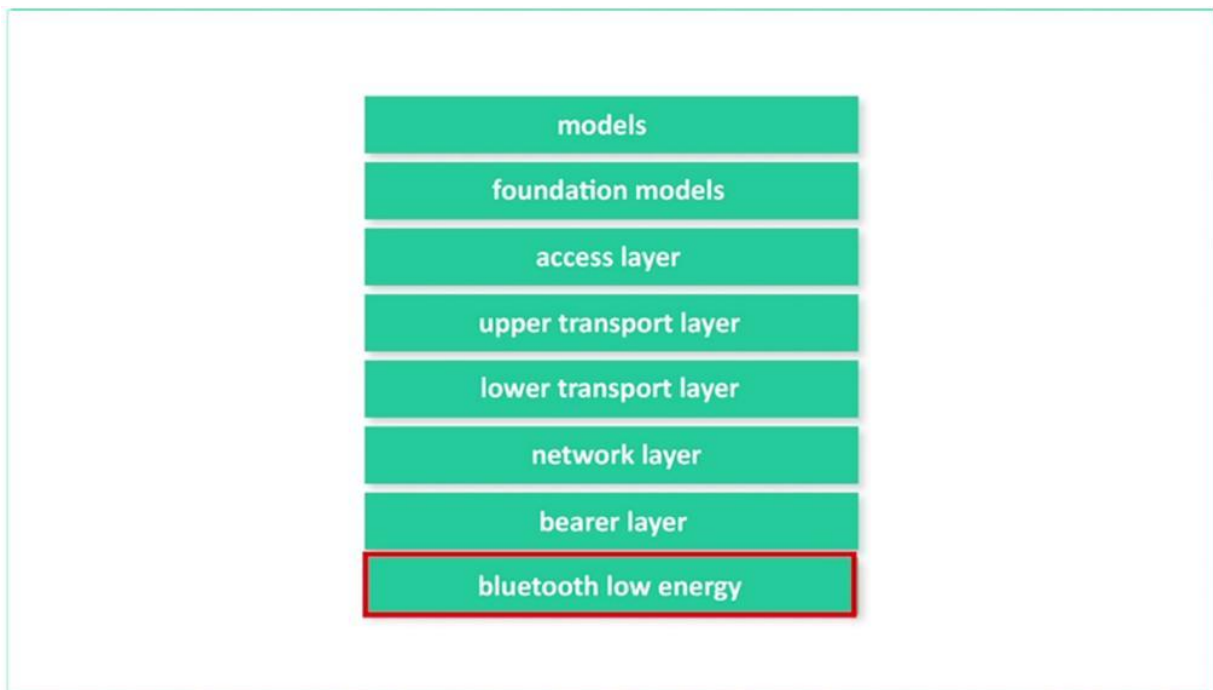


Six extended range since North Candlemas messages to far NY other nodes in between them. This allows the network to extend its range and expand

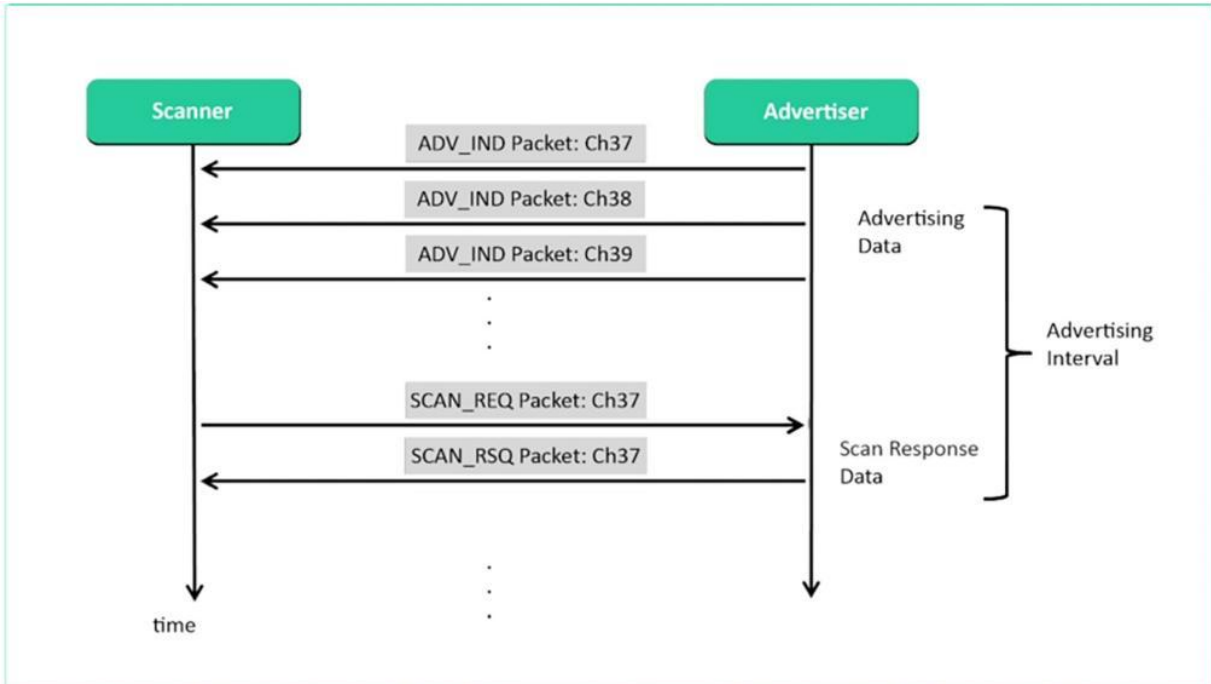
the reach of devices. Seven Self-healing Capabilities Self-heating refers to the fact that there is no single point of failure. If a node drops from the mesh network, the other nodes can still participate and send messages to one another.

FUNDAMENTAL ARCHITECTURE OF BLE MESH

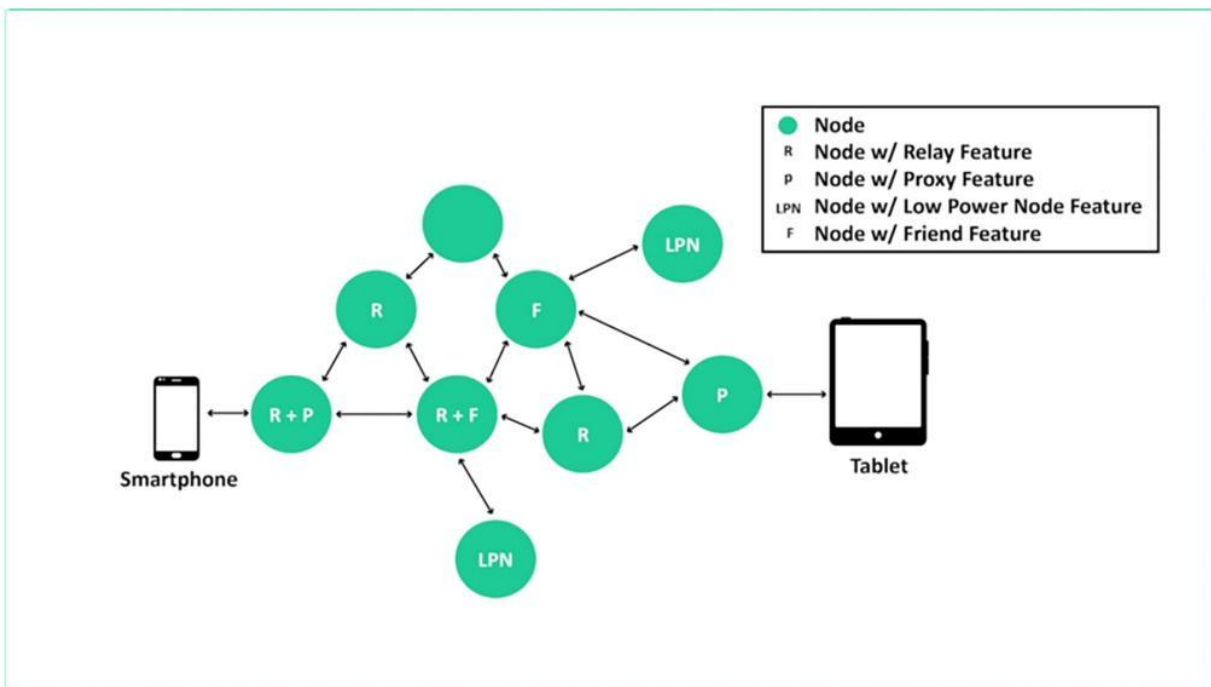
It specifically utilizes the advertising state of BLE devices. Devices within a Bluetooth mesh network do not connect to each other like traditional BLE devices. Rather, they use the advertising and scanning states to relay messages to each other.



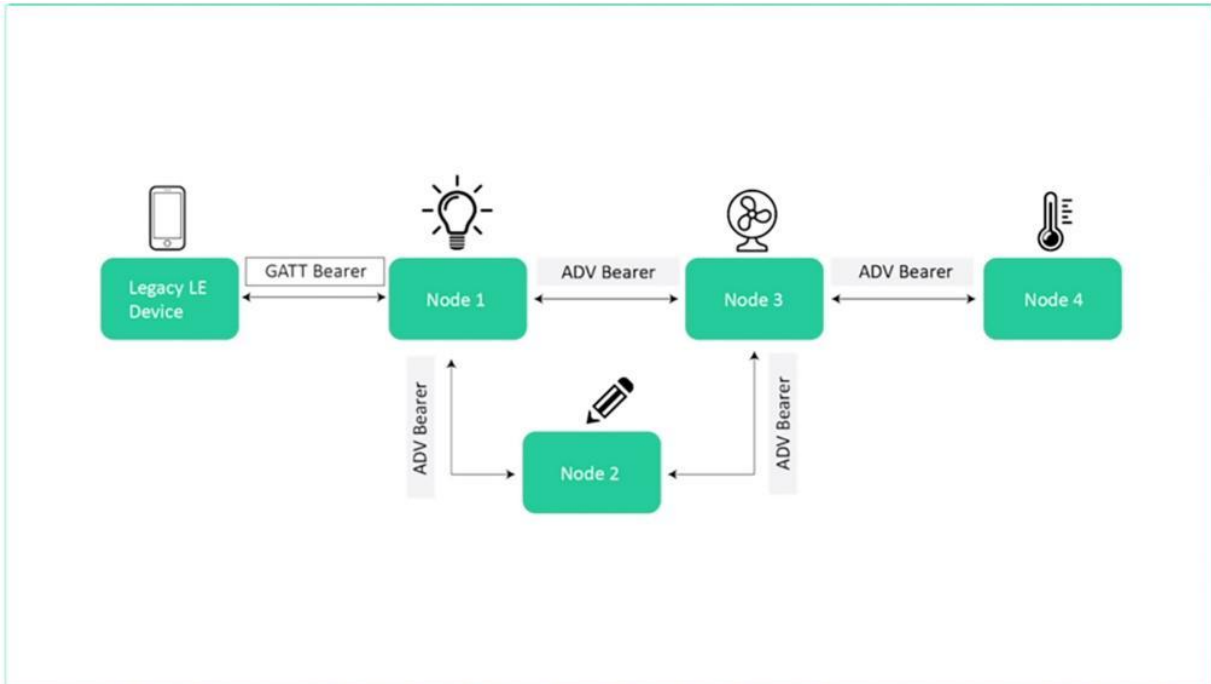
This is the BLE Mesh Architecture. Let's discuss each layer starting from the bottom one by one. Bluetooth Low Energy layer As we mentioned before, Bluetooth mesh builds on top of BLE, so it requires a full BLE stack to be running on the device.



It utilizes the advertising and scanning states for sending and receiving messages the mesh network.

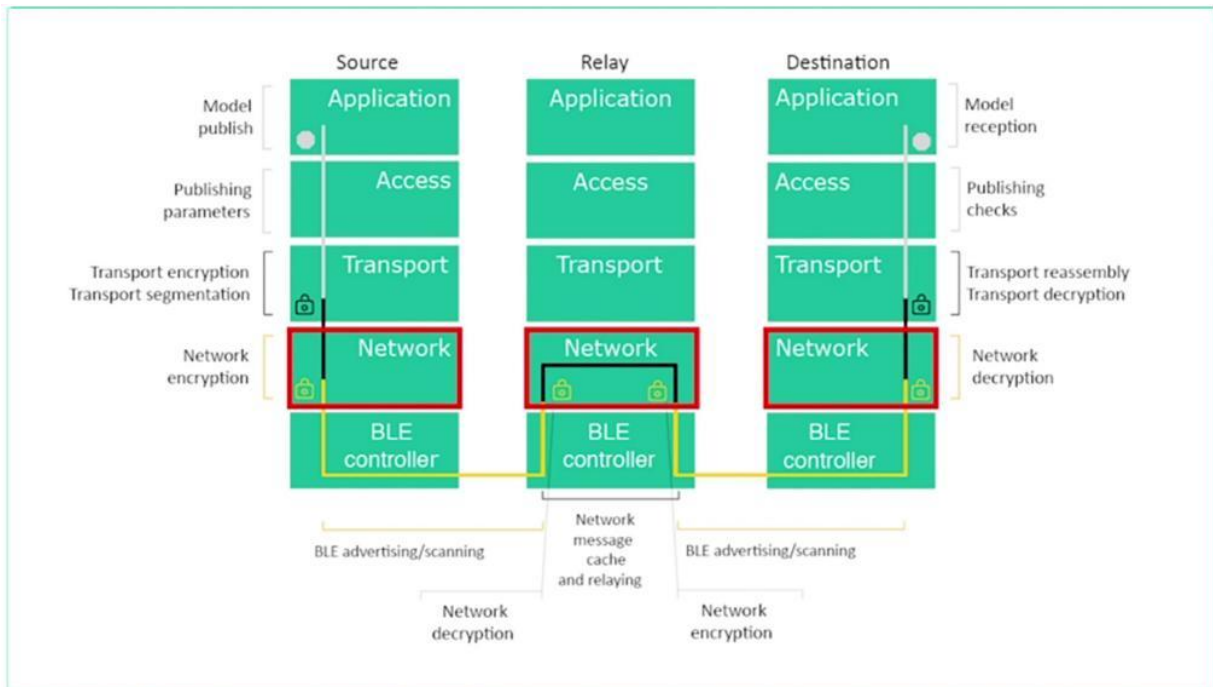


It also supports the connected state and get for special devices called proxy nodes to parallel the Barrowland defines how the different mesh packets are handled.



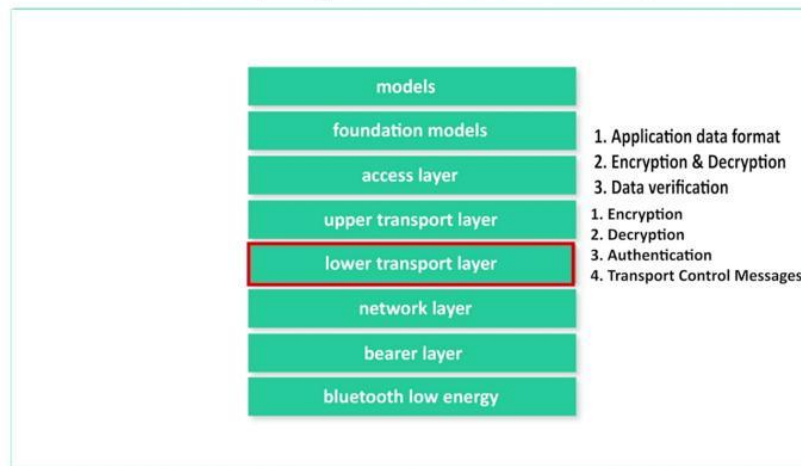
There are two types of Bearers in Bluetooth Mesh advertising. The first is GATT Bearer and the second is ADV Bearer.

The GATT Bearer utilizes the advertising and scanning states of the devices. The ADV Bearer better utilizes the connections state of Bluetooth Mesh devices. It allows non-mesh supporting devices to interact with the mesh network via GATT operations. This is accomplished via a special node called the Proxy Node or a network LED.



This lab defines how transport layer messages are addressed towards one or more elements for lower transport layer and upper transport layer.

Segmentation of packets from upper layer
Reassembly of packets from lower layer

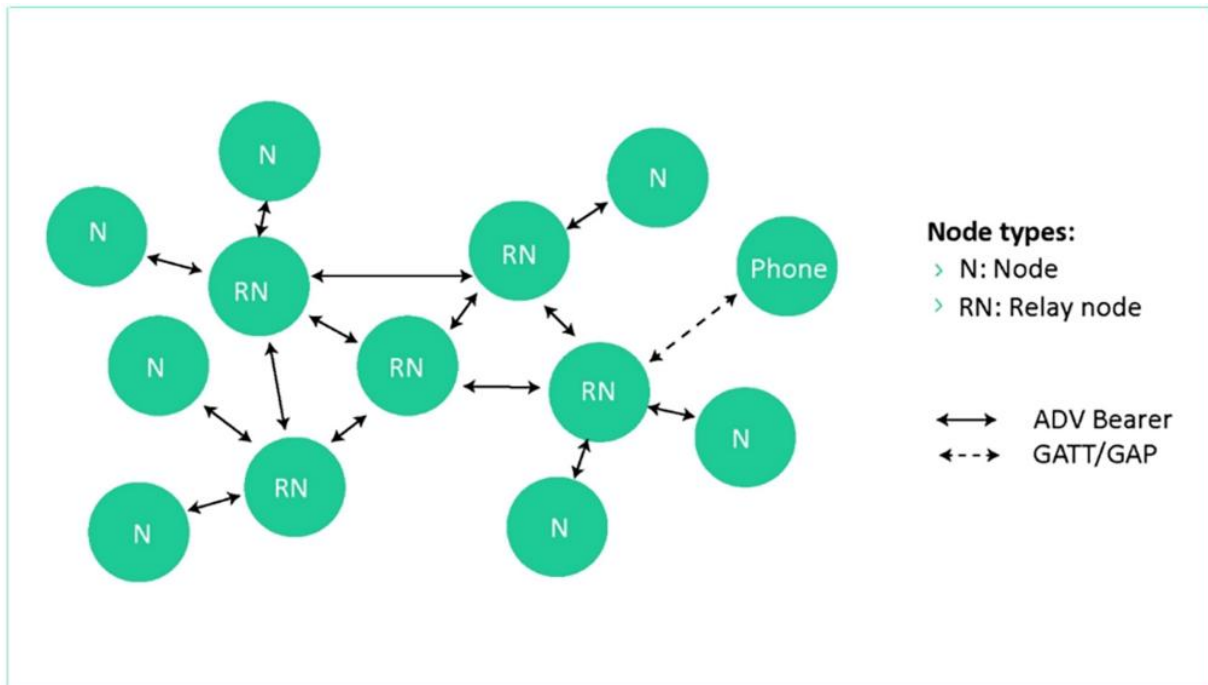


Lower transport layer handles two main tasks segmentation of packets from the larible and reassembly of packets from the layer below. Upper transport layer is responsible for the following functionalities. Encryption, decryption authentication, transport control messages fight access later. This letter

defines how the application uses the transport layer. It handles the following tasks application, data format, encryption and decryption database edification. Foundation Models & Models layer: Foundation layer is concerned with the network configuration and network management models Models layer addresses the implementation of models including behaviors, messages, states, and state bindings. There are three types of models. Server model, Client Model, and Control model.

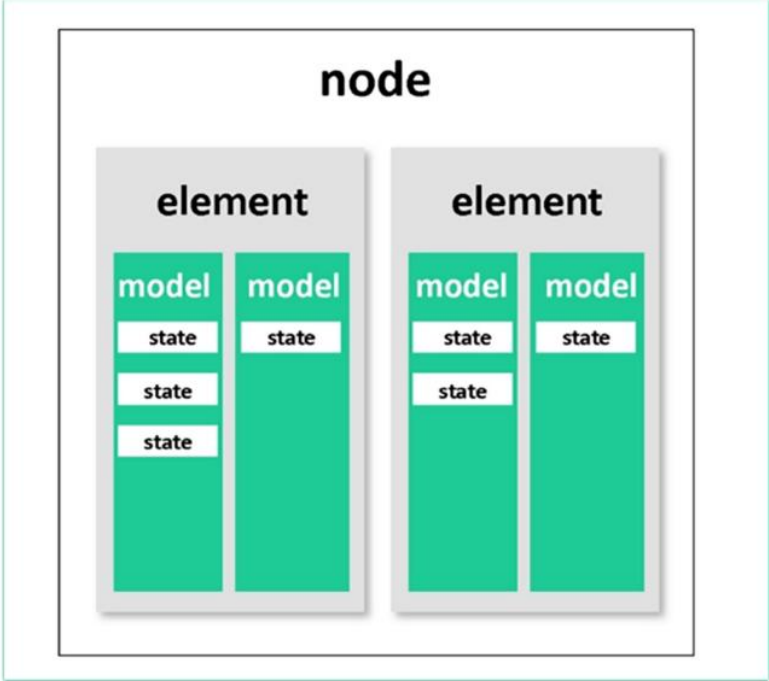
BLE MESH CORE CONCEPTS

we will cover the following topics Nodes Elements States Properties Messages Addresses Publish-Subscribe Managed Flooding and Scenes Let us go over some of the most important concepts and terminologies in Bluetooth mesh.

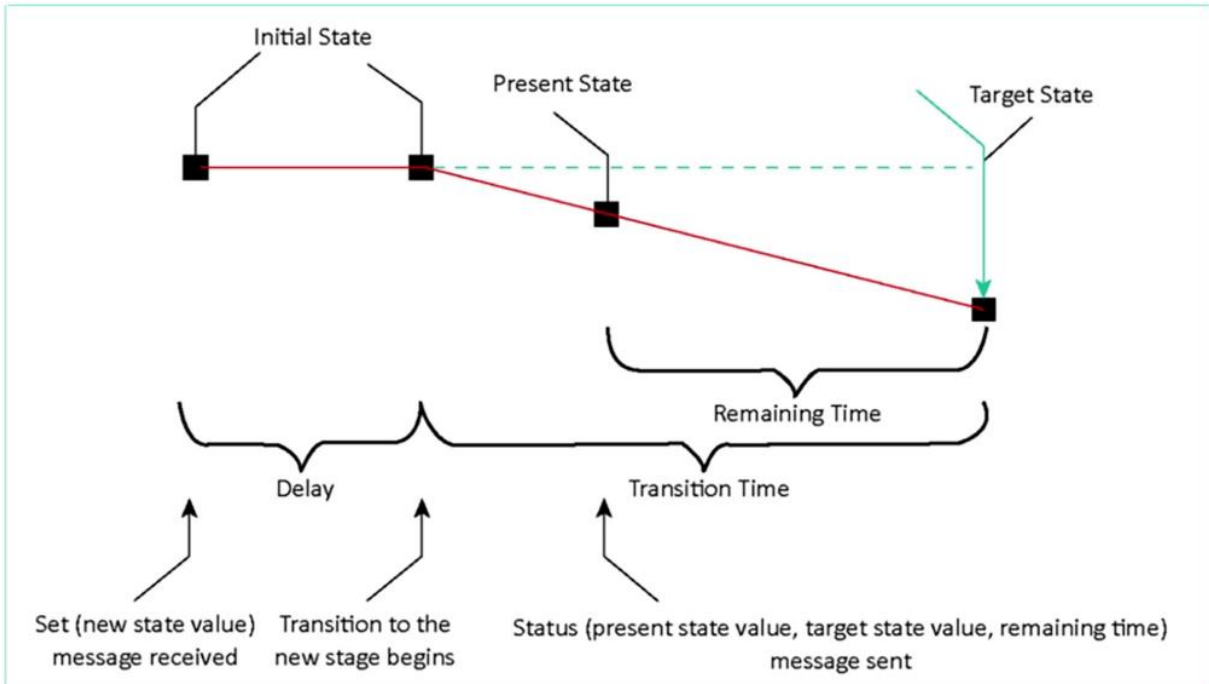


Nodes: Devices that are part of a Bluetooth mesh network are called nodes. Devices that are not part of the network are called unprovisioned devices. Once an unprovisioned device gets provisioned, it joins the network and becomes a node

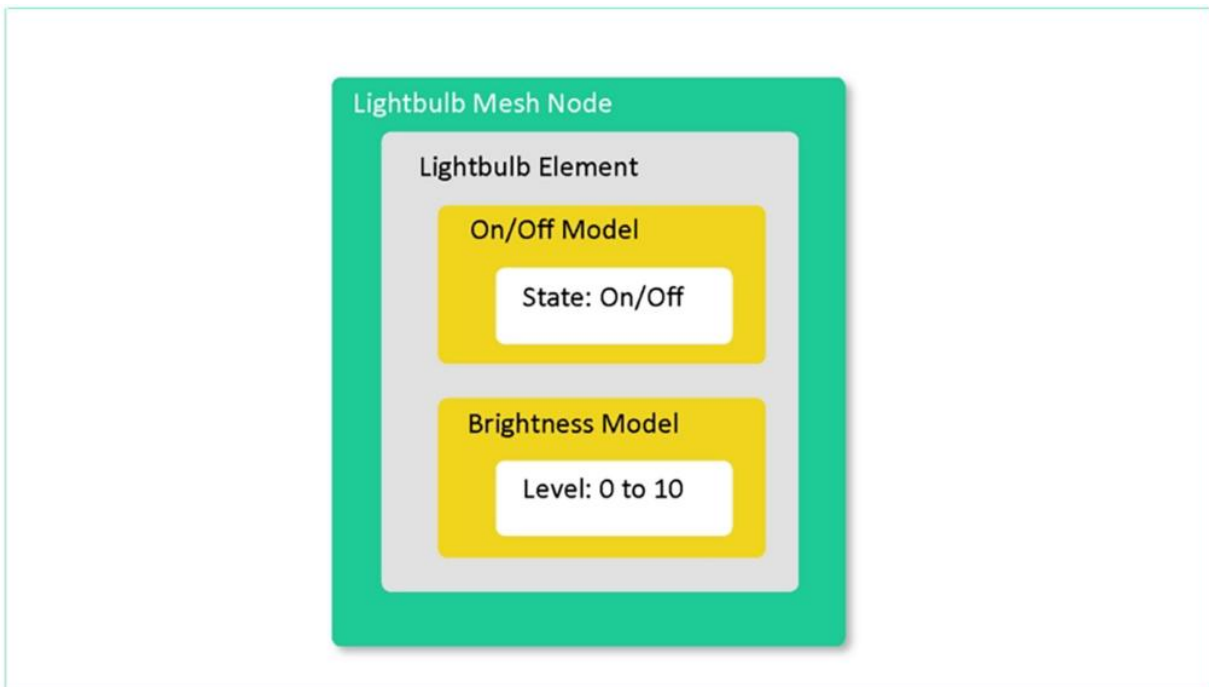
Elements: A node may contain multiple parts which can be controlled independently. For example, a light fixture may contain multiple lightbulbs which can be turned on/off independently. These different parts of a single node are referred to as elements.



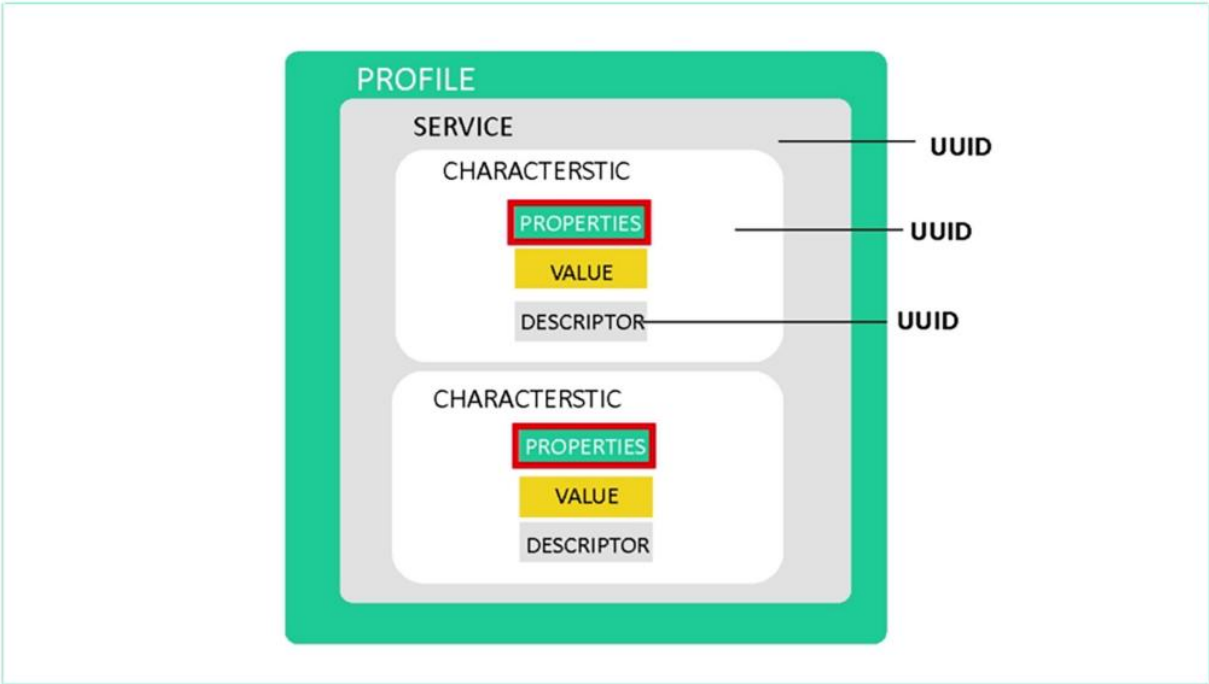
States: Elements can be in various conditions, represented by state values. For example, on and off are states of a lightbulb within a light fixture.



A change from one state to another is called a state transition. This can be instantaneous, or it can occur over time after what's called a transition period. When a state change occurs, it is likely to cause a change in the behavior of an element.



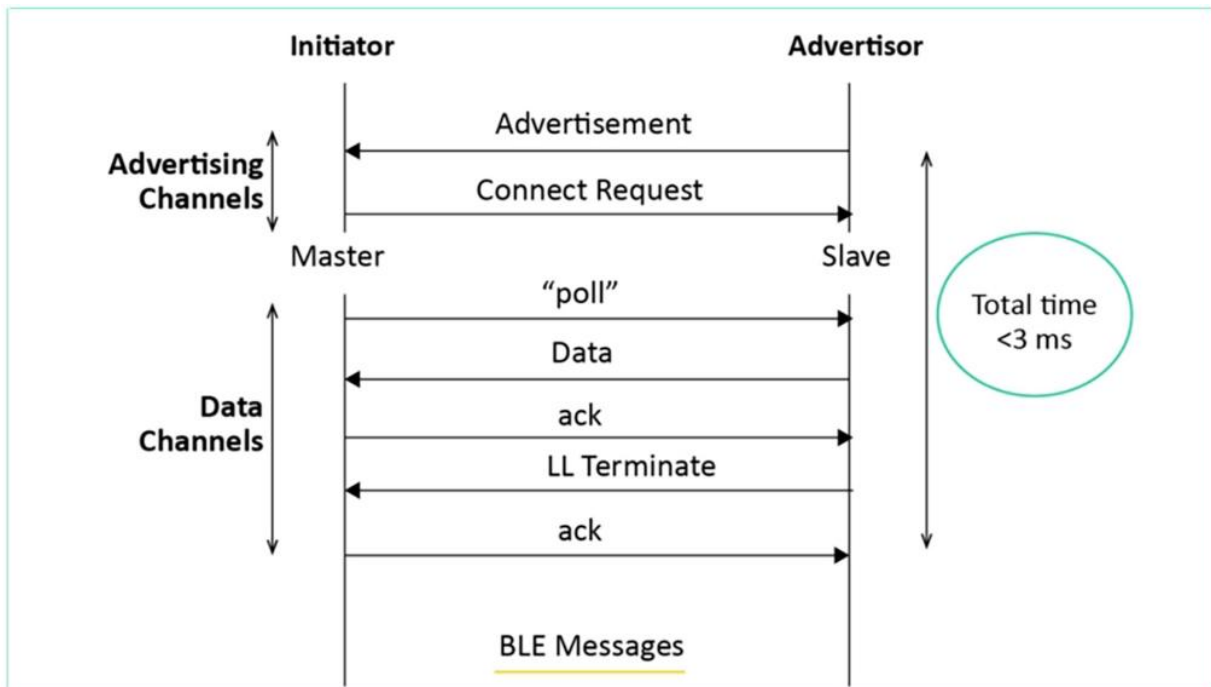
Some states may be bound to each other, meaning that a change in one state triggers a change in the other.



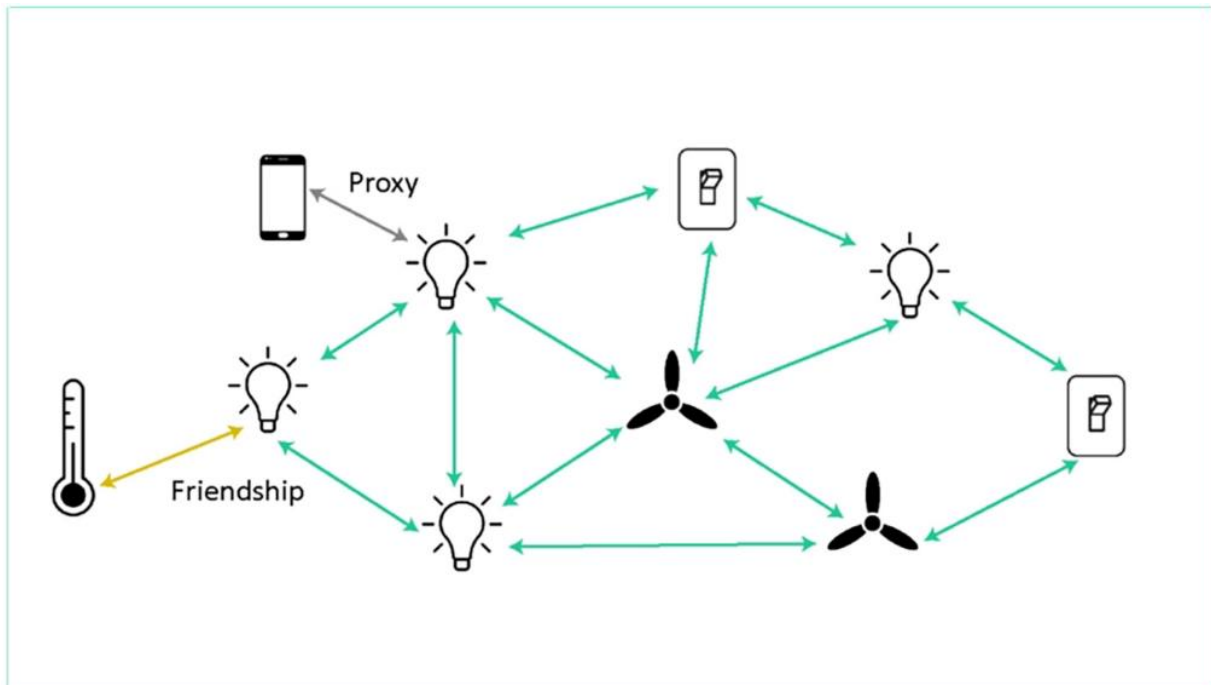
Properties, properties add some context to state value, for example, defining that a temperature There may be two or more states bound to each other.

Properties: Properties add some context to a state value. For example, defining that a temperature value is an outdoor or indoor temperature.

There are two types of properties: Manufacturer property: provides read-only access Admin property: provides read-write access Manufacturer property: provides read-only access Admin property: provides read-write access



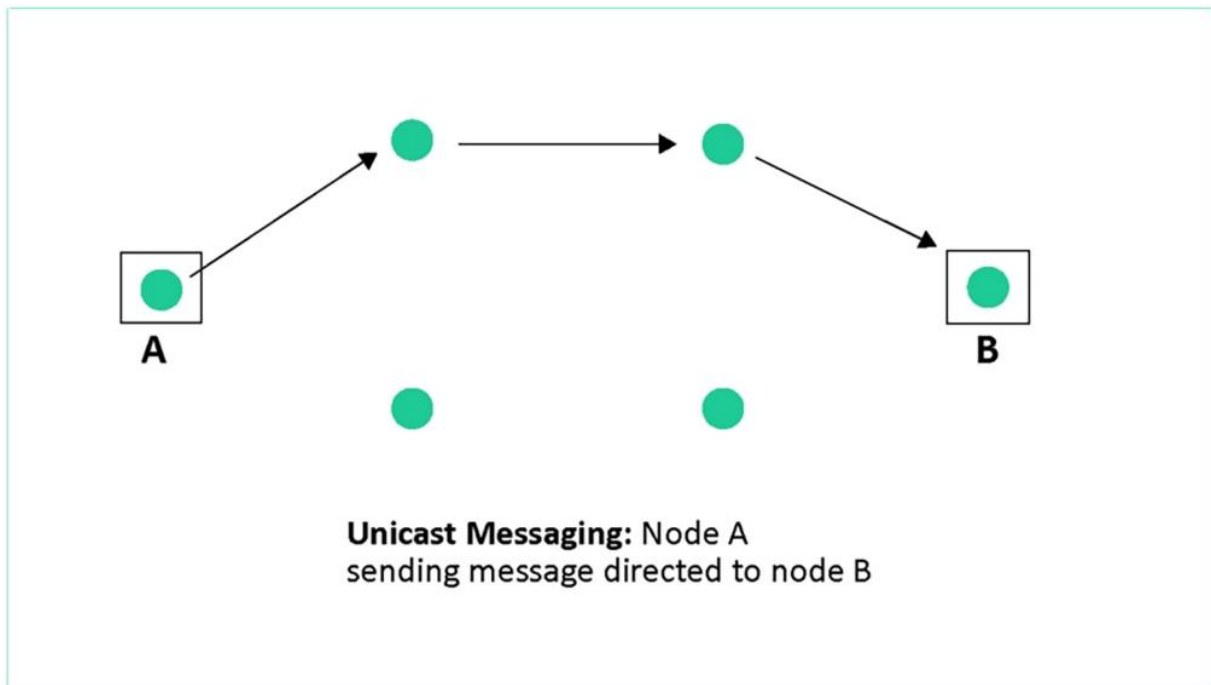
Messages: In Bluetooth mesh, all communications within the network are message-oriented, and nodes send messages to control or relay information to each other.



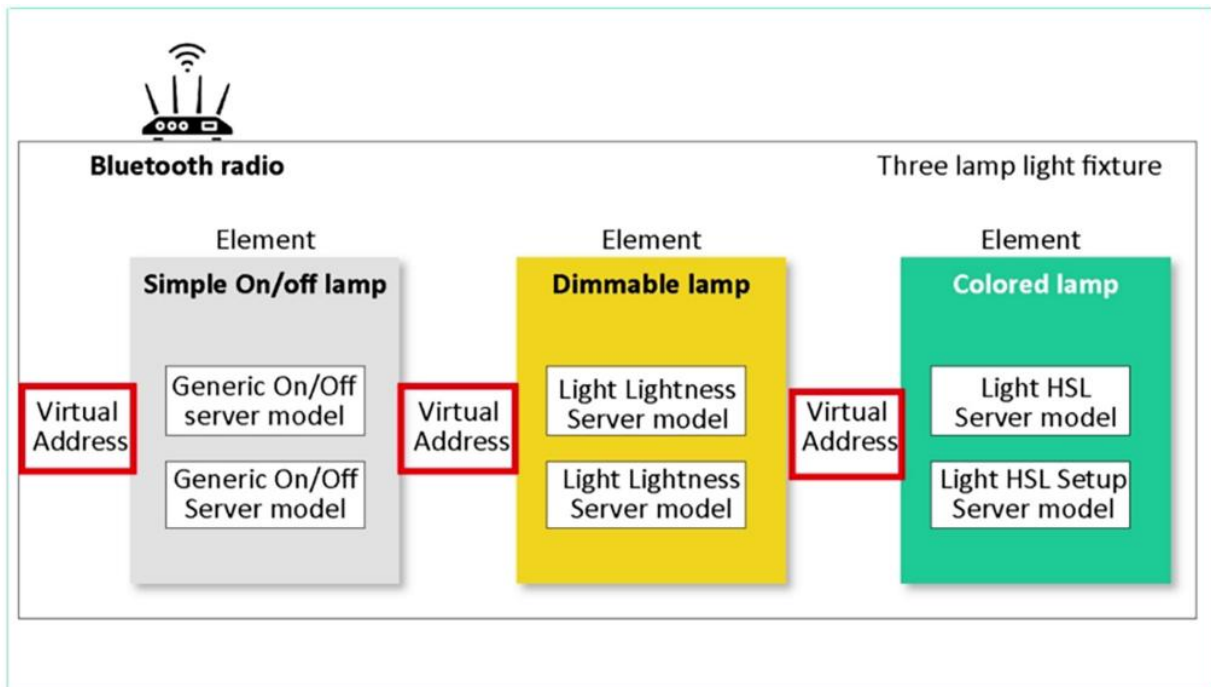
Messages are the mechanism by which operations on nodes are invoked. If a node needs to report its status, it also sends it via a message A given

message type represents an operation on a state or collection of multiple state values There are three types of messages in Bluetooth mesh, each of which is defined by a unique opcode (operation code):

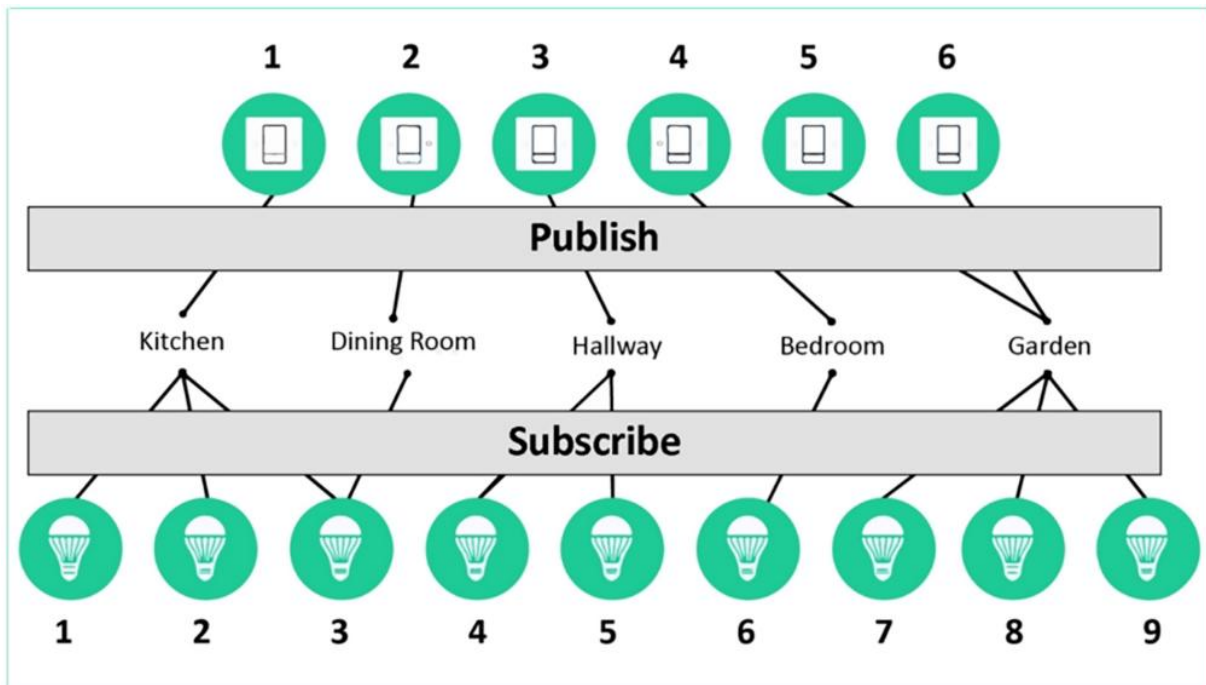
A good message is a message to request the state from one or more nodes. A state message is a message to change the value of a given state. A status message is used to send the element status addresses. Messages in the network must be sent to and from an address. There are three types of addresses.



One Unicast Address and others that uniquely identifies a single node assigned during the provisioning process to group, address and address used to identify a group of nodes. A group address usually reflects a physical grouping of nodes such as all nodes within a specific Group



three virtual address and address that may be assigned to one or more elements spanning one or more nodes. This acts as a label and takes the form of a 128 with unity with which an element can be associated. Publish subscribe. The way messages are exchanged in a limited network is via the public subscribe button. Publishing is the act of sending a message. Subscribing is a configuration used to allow selected messages to be sent to specific addresses for processing.



Here is an example of a mesh network in a home that's composed of six light switches and nine light bulbs, the network utilizes the public subscribe method to allow Nords to send messages to each other. Typically, messages are addressed to group or virtual addresses. The benefit of using group or virtual addresses is that adding or removing nodes does not require reconfiguration of nodes. The benefit of using group or virtual addresses is that adding or removing nodes does not require reconfiguration of nodes.

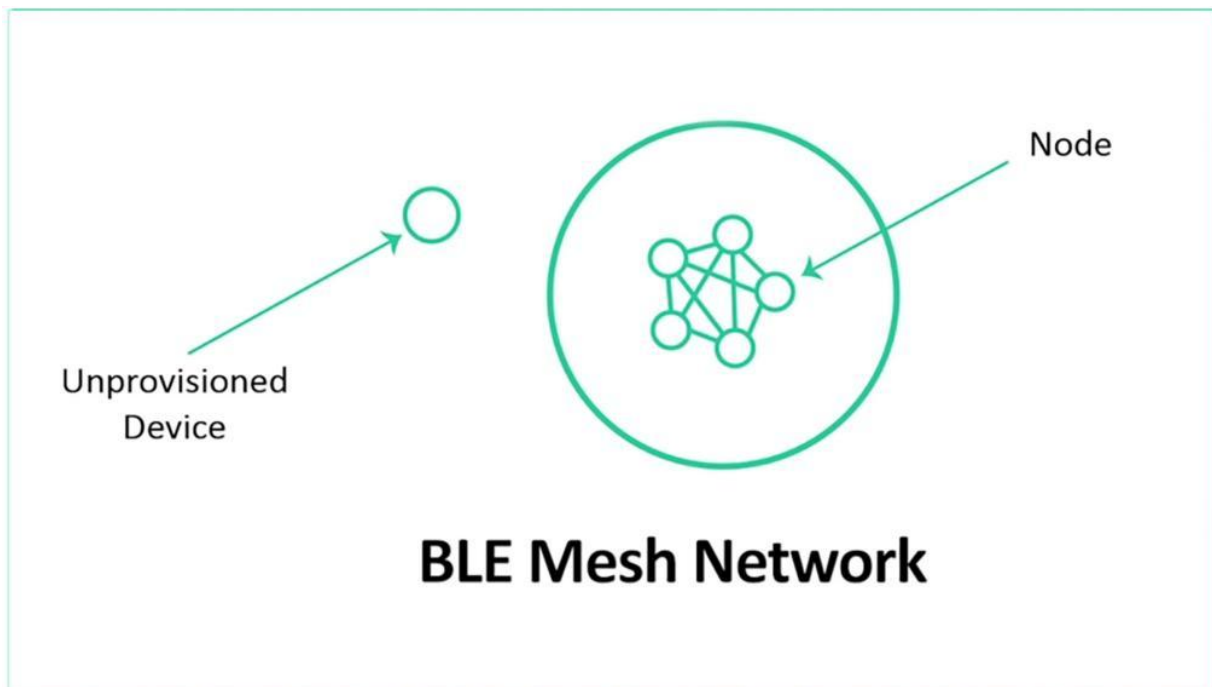
Managed Flooding: Managed flooding relies on broadcasting messages to all nodes within range of the sender node, with a few added optimizations such as

1. Messages have a TTL assigned. TTL stands for time-to-live, which limits the number of hops a message can take across multiple nodes within the mesh network
2. Messages are cached
3. Heartbeat messages are sent periodically to indicate to other nodes that the sender is alive and active within the network.

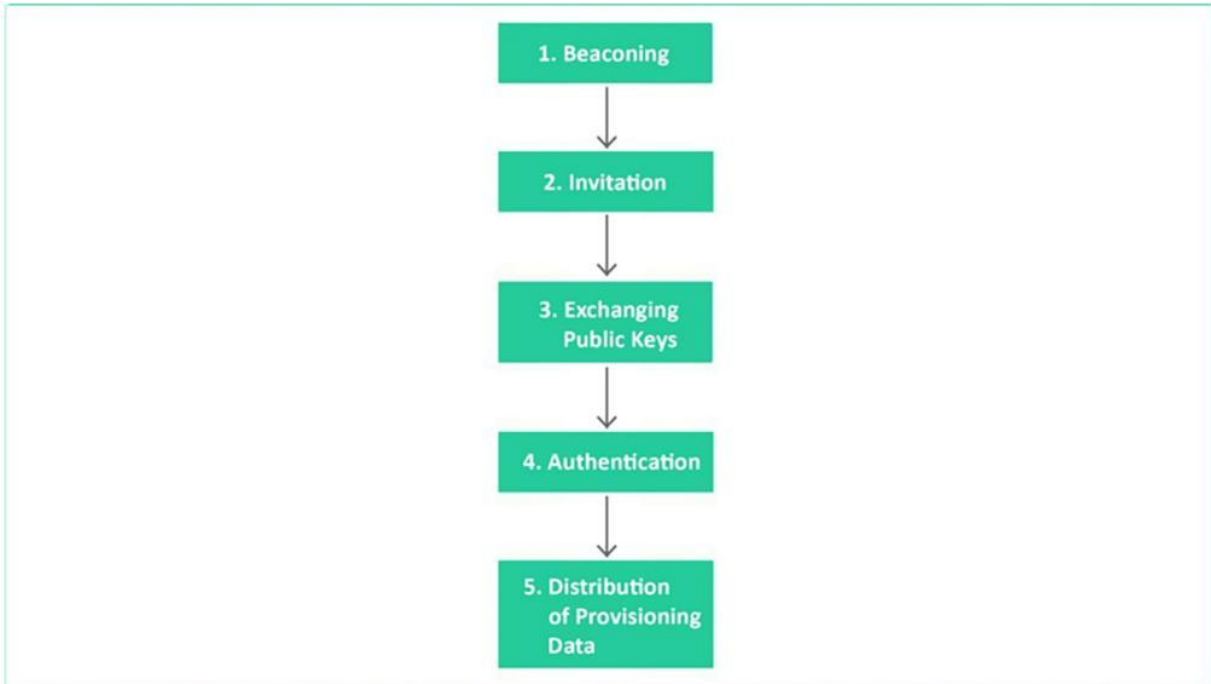
Scenes: The final concept we want to cover is the concept of scenes in a Bluetooth mesh network. A scene is a stored collection of states and is identified by a 16-bit number which is unique within the mesh network. Scenes allow triggering one action to set multiple states of different nodes. They can be triggered on-demand or at a specified time. For example, a scene may be configured to set the temperature of a room to 72 degrees the living room lights to a certain brightness level, and the window blinds to close.

BLE MESH PROVISIONING PROCESS

The provisioning process is one of the most important concepts in Bluetooth mesh. It is used for adding devices to the mesh network.

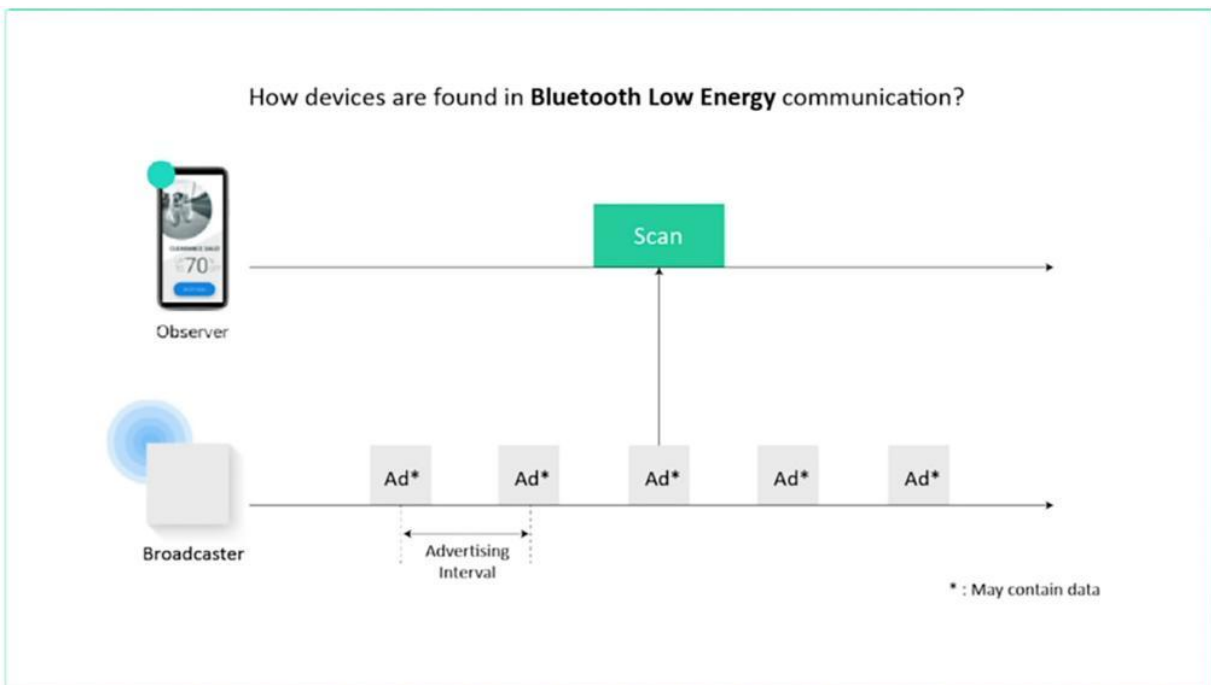


A device that gets added to the network is called a node, and the device used to add a node to the network is called the provisioner usually a tablet, smartphone, or a PC.



This process involves five steps: The process involves five steps. Step one beginning.

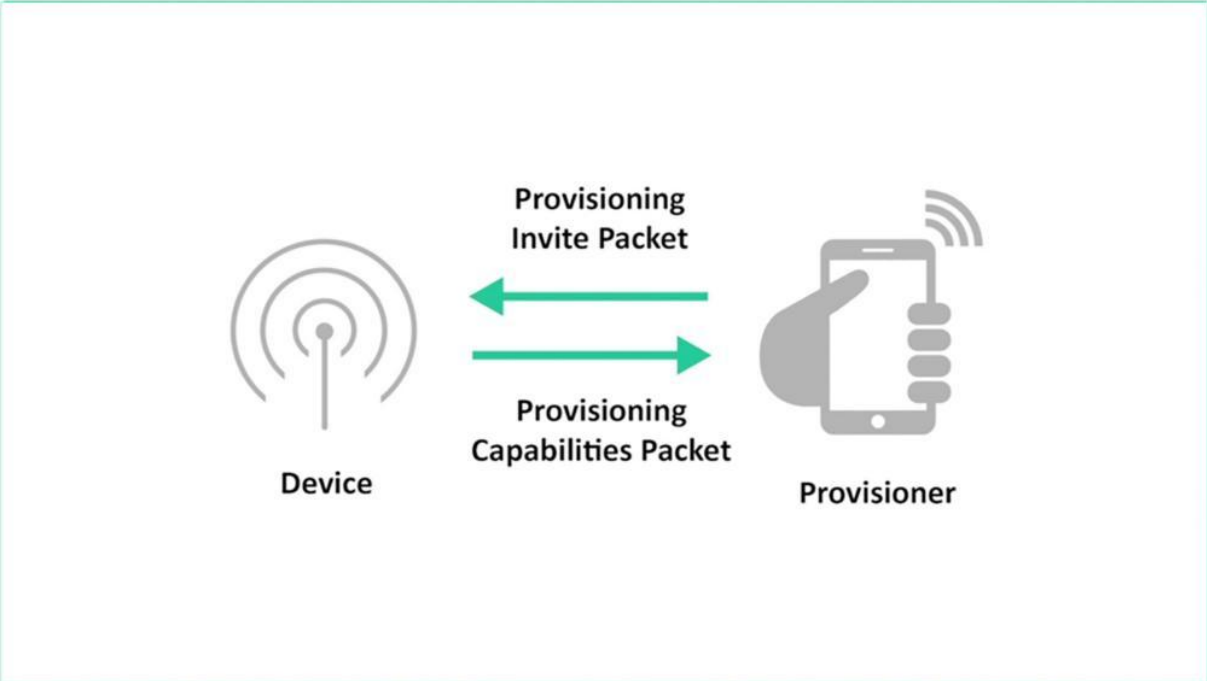
Step one involves what is called beaconsing, where the provision the device announces its availability to be provisioned by sending the message back advertisements in the advertisement packets.



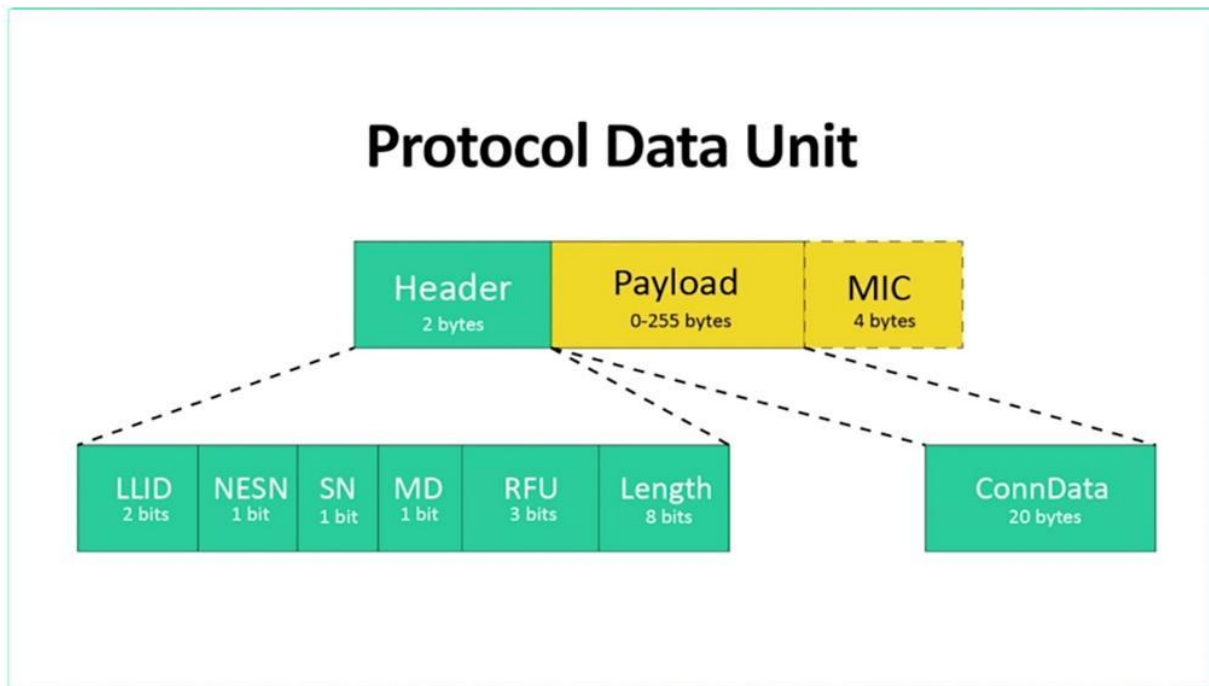
This is a new type of advertisement data type introduced in the bill standard.



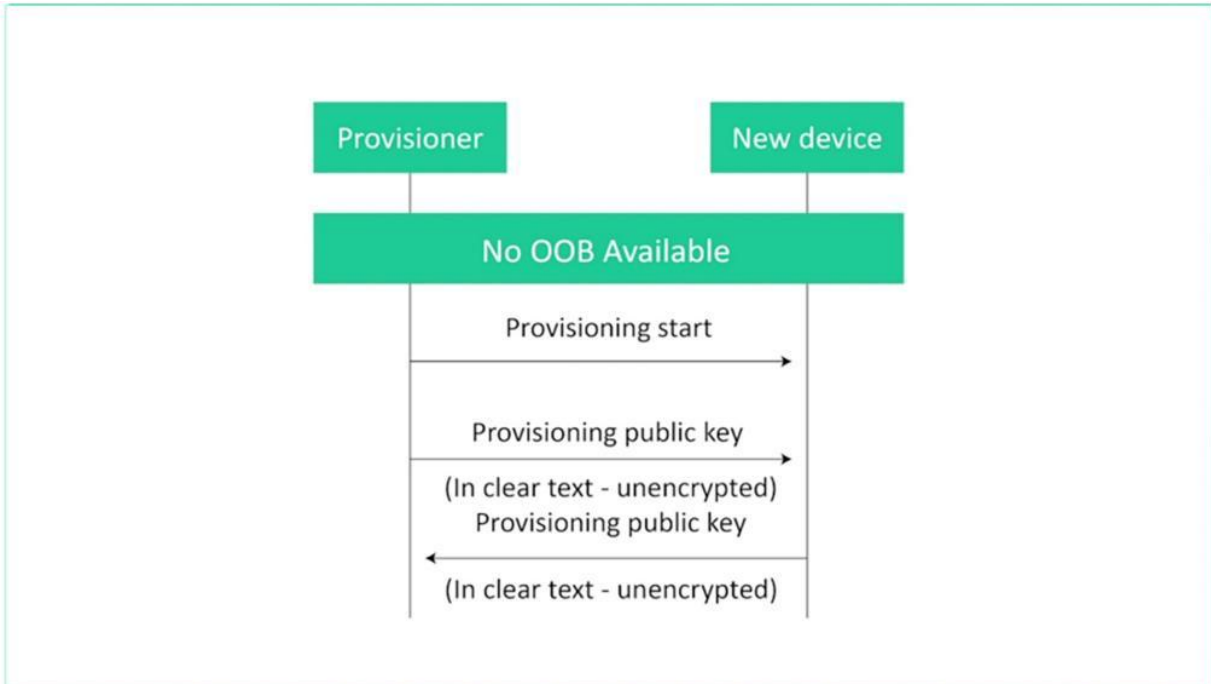
A common way to trigger this process is via a defined sequence of button presses on the unprovisioned device.



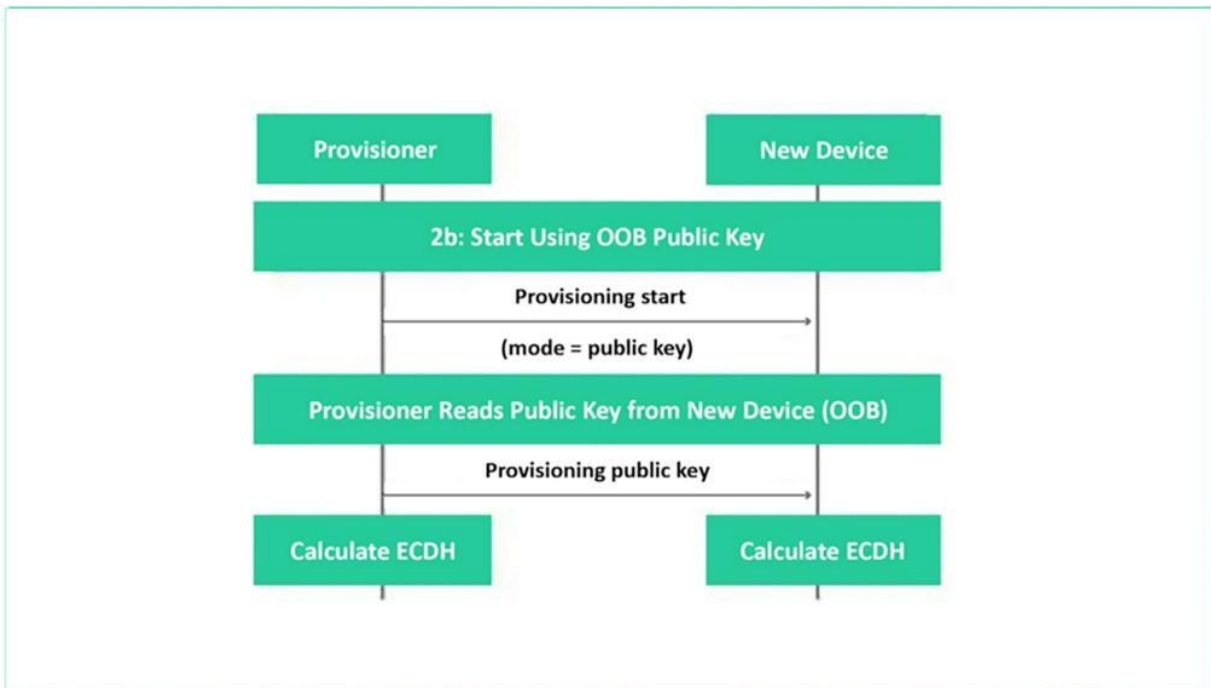
Step 2: Invitation When the provisioner discovers the unprovisioned device via the beacons that were sent, it sends an invitation to this unprovisioned device When the provisioner discovers the unprovisioned device via the beacons that were sent, it sends an invitation to this unprovisioned device This uses a new type of PDU introduced in Bluetooth mesh called the provisioning invite PDU.



This uses a new type of PDU introduced in Bluetooth mesh called the provisioning invite PDU. The unprovisioned device then responds with information about its capabilities in a provisioning capabilities PDU, which includes: The number of elements the device supports. The set of security algorithms supported. The availability of its public key using an Out-of-Band (OOB) technology. The ability of this device to output a value to the user. The ability of this device to allow a value to be input by the user. The number of elements the device supports. The ability of this device to allow a value to be input by the user. Step 3: Public Key Exchange Security in Bluetooth mesh involves the use of a combination of symmetric and asymmetric keys such as the Elliptic-curve Diffie-Hellman (ECDH) algorithm.

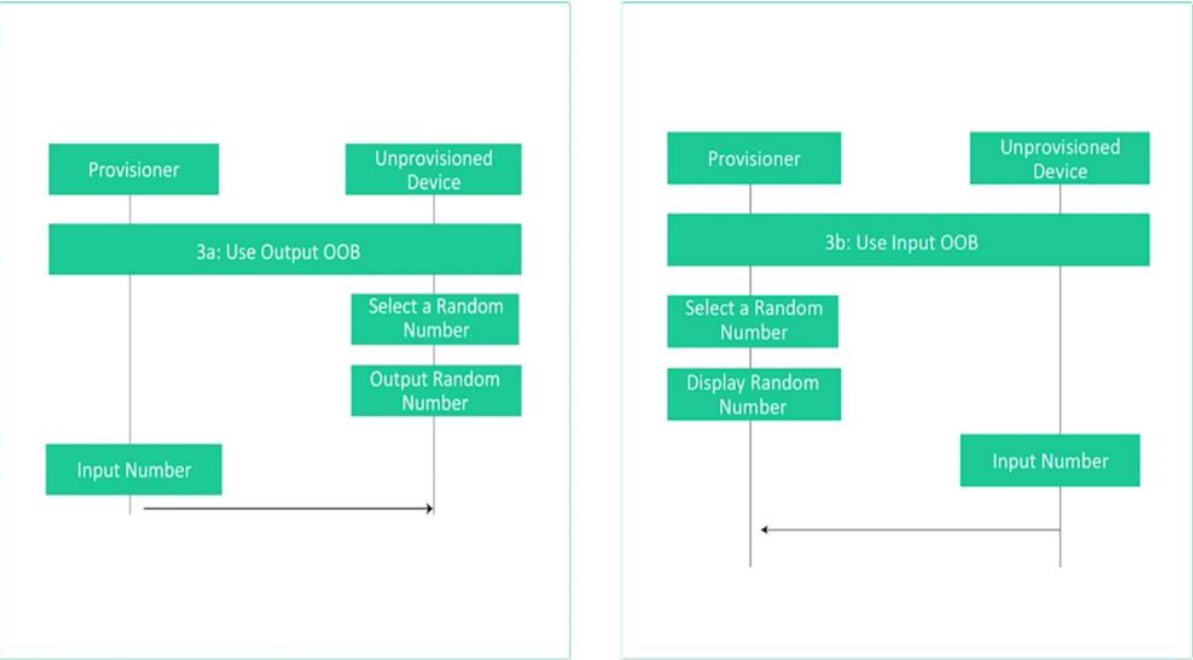


In ECDH, public keys are exchanged between the provisioner and the device to be provisioned.



This is done either directly over BLE or via an out-of-band (OOB) channel. Step 4: Authentication The next step is to authenticate the unprovisioned device. This usually requires an action by the user by interacting with both

the provisioner and the unprovisioned device. The authentication method depends on the capabilities of both devices used. The authentication method depends on the capabilities of both devices used. In one case, called the output OOB, the unprovisioned device could output a random single or multiple-digit number to the user in some form, such as blinking an LED a number of times. That number is then input into the provisioning device via some input method. such as blinking an LED a number of times. That number is then input into the provisioning device via some input method. Other cases include an input OOB, where the number is generated by the provisioner and entered into the unprovisioned device, a static OOB, or no OOB at all. Other cases include an input OOB, where the number is generated by the provisioner and entered into the unprovisioned device, a static OOB, or no OOB at all.

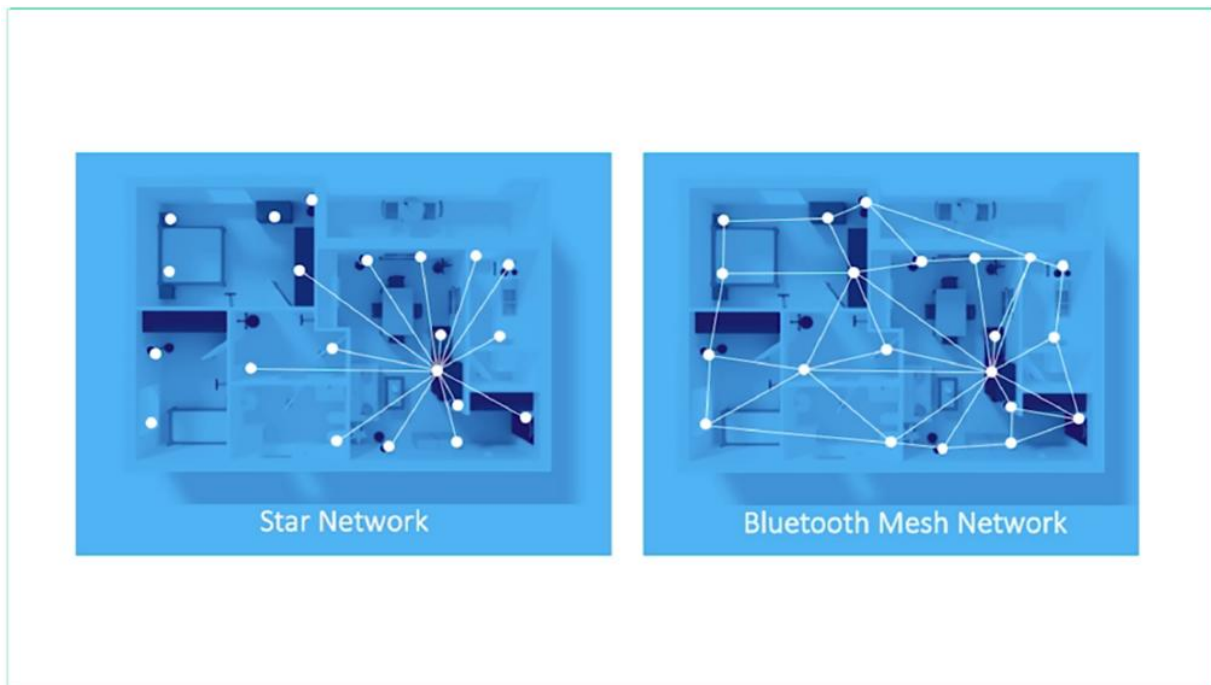


Regardless of the authentication method used, the authentication also includes a confirmation value generation step and a confirmation check step. Regardless of the authentication method used, the authentication also includes a confirmation value generation step and a confirmation check step. Step 5: Provision Data Distribution After authentication is complete, each device derives a session key using their private key and the public key sent to it from the other device. Step 5: Provision Data Distribution After

authentication is complete, each device derives a session key using their private key and the public key sent to it from the other device. The session key is then used to secure the connection for exchange of additional provisioning data including the network key (NetKey), a device key, a security parameter known as the IV index, and a unicast address which is assigned to the provisioned device by the provisioner. After this step, the unprovisioned device becomes known as a node.

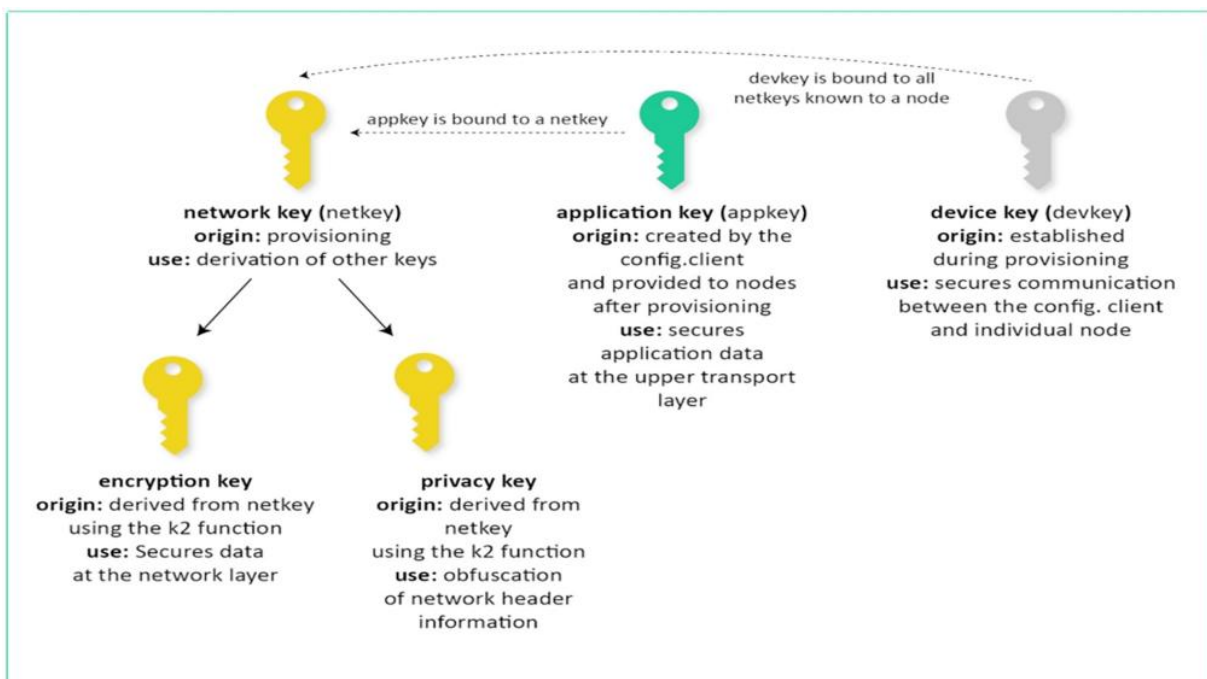
SECURITY IN BLE MESH

The first important note regarding security in BLE mesh is that it is mandatory.



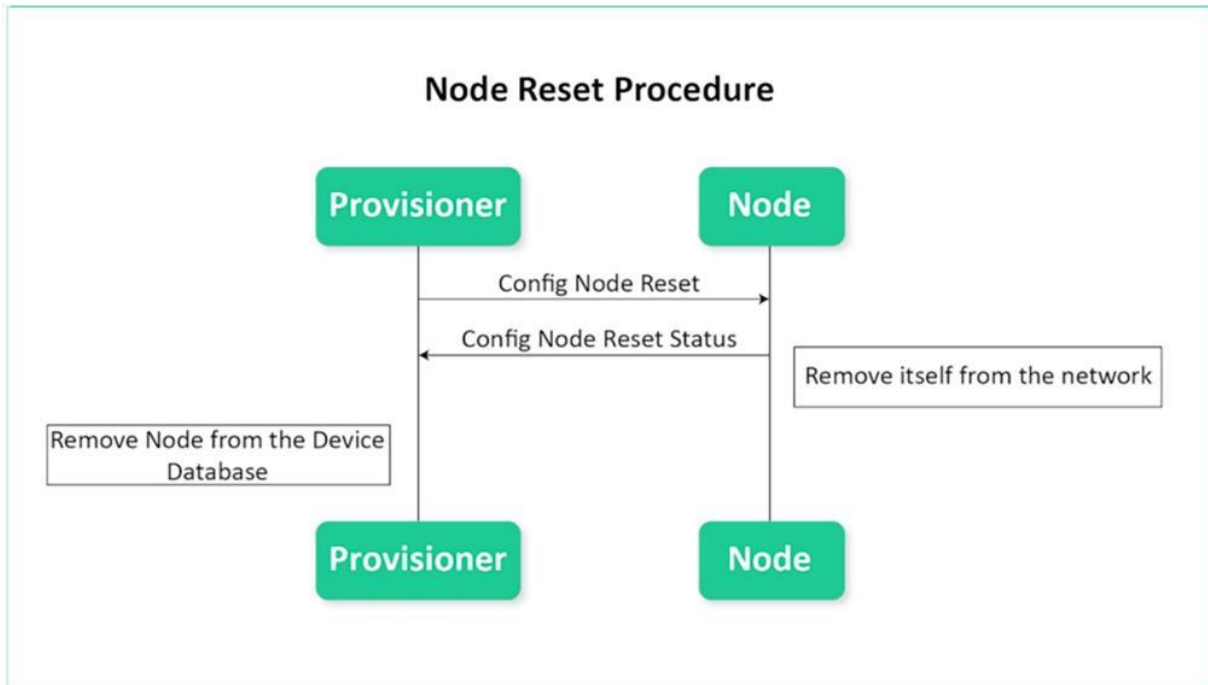
This is compared to BLE where it is optional and left to the developer to decide whether to include it or not. Let's look at some basics of security in BLE Mesh. include it or not. All mesh messages are encrypted and

authenticated Network security, application security, and device security are all handled independently. Security keys can be changed during the life of the mesh network All mesh messages are encrypted and authenticated Network security, application security, and device security are all handled independently. Security keys can be changed during the life of the mesh network Due to the separation of security between the network, application, and device levels, there are three types of security keys for BLE Mesh. Three security keys can be changed during the life of mesh network due to the separation of security between the network application and device levels. There are three types of security keys for Bielema.



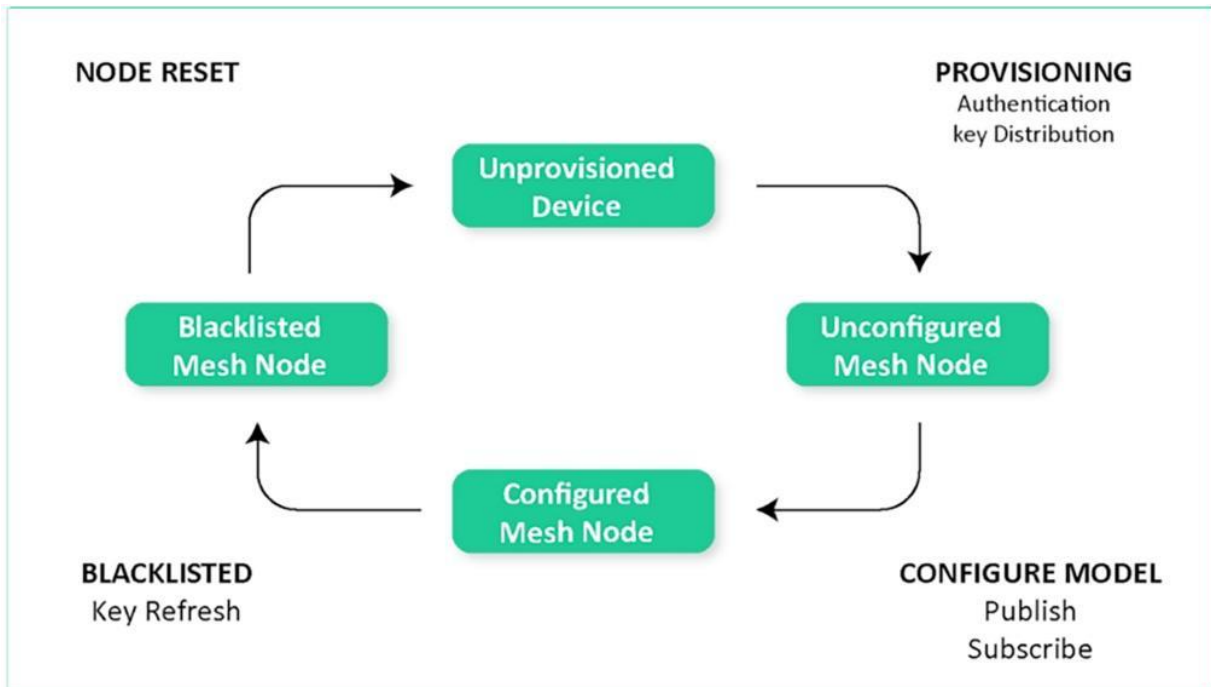
One network position of the shared key makes the device part of the network, also known as a node. There are two keys derived from the network the network encryption key and the basic position of the network allows the node to decrypt and authenticate up to the network layer, allowing the relay of messages but no application data decryption to application. This is a key that is shared between a subset of nodes within a mesh network. Normally, those that participate in the common application. For example, a lighting system application key would be shared between light switches and light bulbs, but not with a thermostat or a motion sensor. An application is used to decrypt and authenticate messages at the

application level, but it is only valid within one mesh network, not across multiple networks, three devices. This is a device specifically that is used during the provisioning process for securing communication between the provision device and the provision.

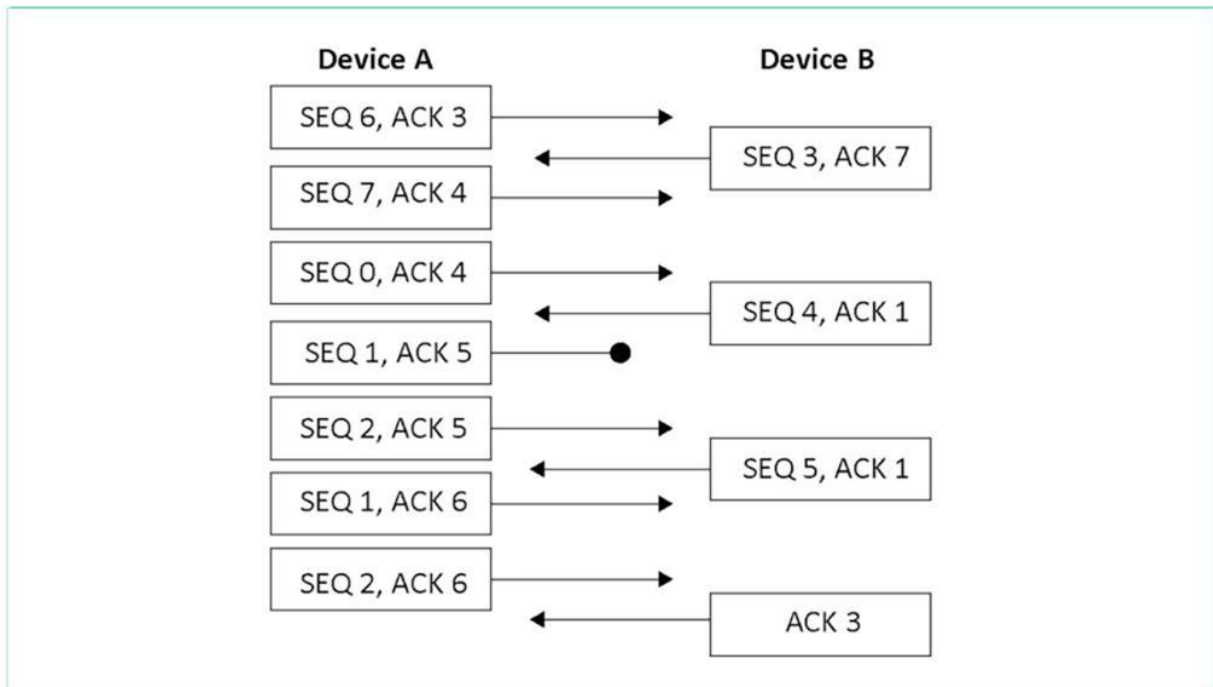


Now let's look at some security concerns in Bill eMESH, nor the removal, privacy and lepley attacks out of concern in Bill Image. First, let's look at Northmoor with one major concern with the mesh network is gaining access to a network via a discarded or remove the device that used to be the part of the network.

This can be accomplished by gaining physical access to the keys stored within the device in order to protect against such an attack.



BLE defines a procedure for removal of a node where the device is added to a blacklist and the keys are refresh. This process distributes new network keys, application keys and other relevant data to all nodes except those in the blacklist. Now let's look at privacy. Another concern is privacy. The way privacy is addressed in this via the use of a privacy key that is used to obfuscate the message that the privacy is derived from the network. For example, the source address could be obfuscated to prevent tracking of a device via its address. Finally, let's look at replay attacks. The last security topic we want to cover is replay attacks. A replay attack is when one or more messages are stored and replayed later by a malicious device.



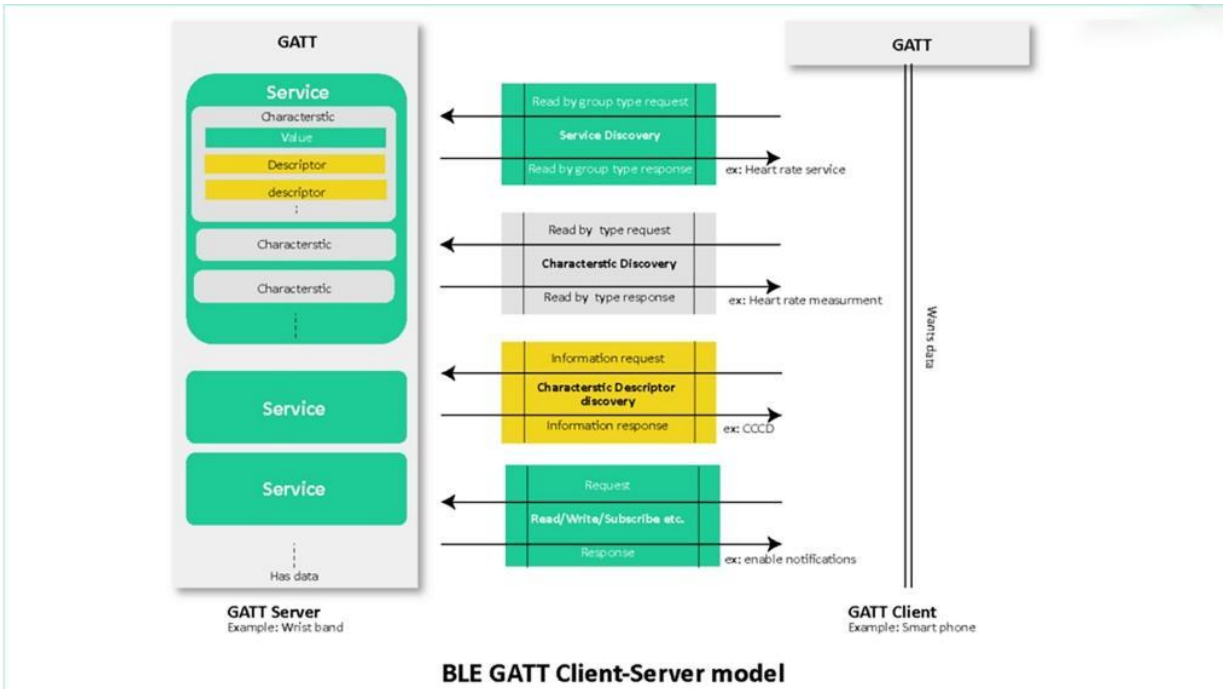
BLE provides protection against replay attacks by utilizing a sequence number also known as Ithica. Elements implement. They seek to value every time they publish a message, a note receiving a message from an element which contains a siku value that is less than or equal to the one in the last valid message we discarded.

IVI	NID	CTL	TTL	SEQ	SRC	DST	Transport PDU	NetMIC
Field Name	Bits	Notes						
IVI	1	Least significant bit of IV Index						
NID	7	Value derived from the NetKey used to identify the Encryption Key and Privacy Key used to secure this PDU						
CTL	1	Network Control						
TTL	7	Time To Live						
SEQ	24	Sequence Number						
SRC	16	Source Address						
DST	16	Destination Address						
TransportPDU	8 to 128	Transport Protocol Data Unit						
NetMIC	32 or 64	Message Integrity Check for Network						

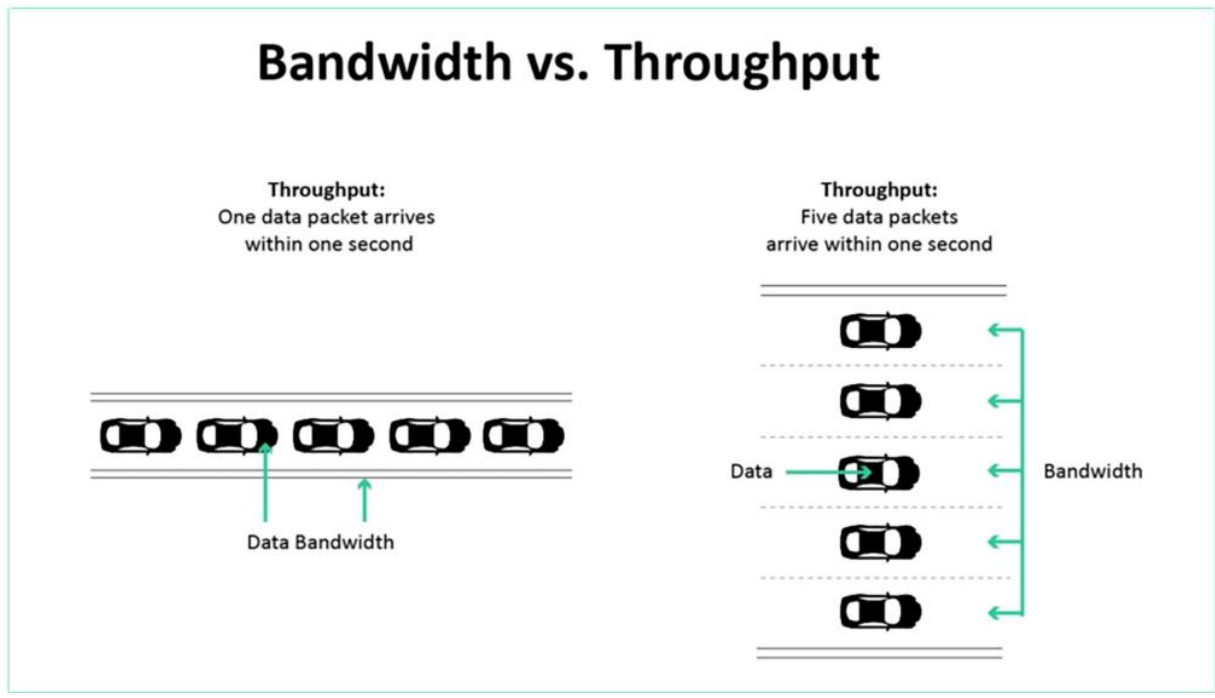
Another way element provides protection against replay attacks is by using an incrementing IVI index, which is an additional value that also gets validated when the message is received.

BLE THROUGHPUT MEASURING IMPLEMENTATION USING ESP32

will cover the following topics Project Overview Setting up 2 ESP32s
Coding, Implementation, and Testing

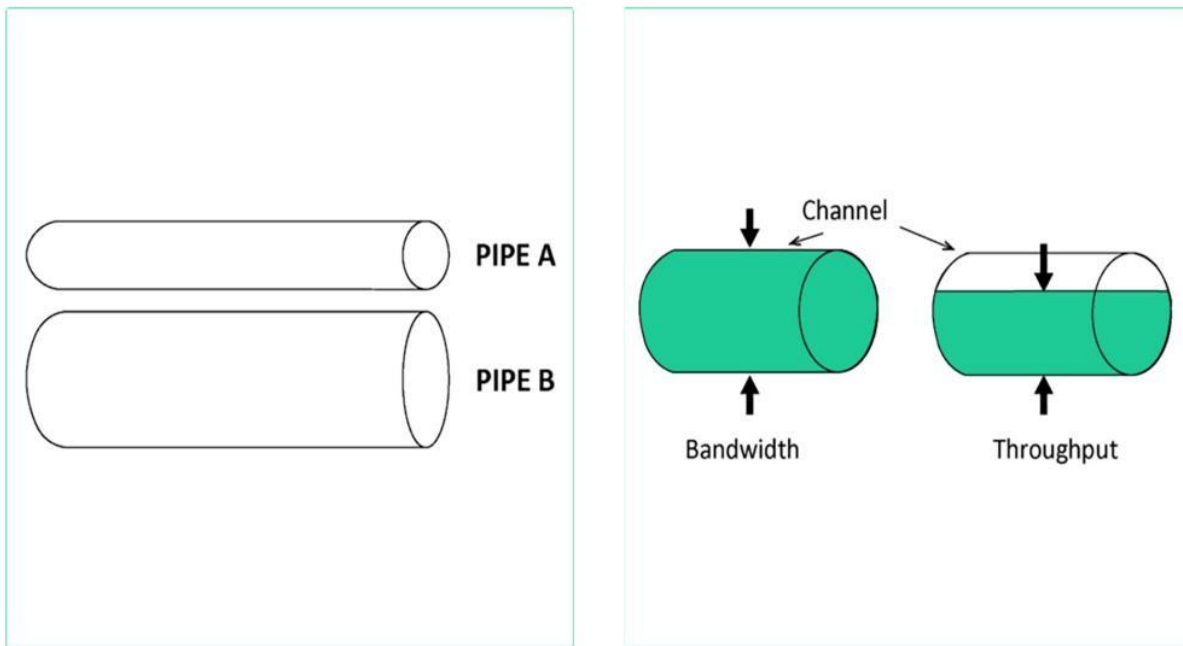


In this project, we are going to test the actual transfer rate of data between a GATT Server and a Client.



it with bandwidth to explain this difference. Let's use an analogy of water flowing through pipes, cancelling the two pipes shown here. Which pipe do

you think will pass more water through it? The default answer is baby, because that is the fact pipe.



However, the real answer to the question is that it depends if water is flowing at maximum capacity through both pipes, then Pipe B will carry more water through at a particular time. But what if much more water is coming in to pipe it than baby? Or what if there is debris in baby that is restricting the flow of water inside the pipe? Using the analogy about bandwidth, which can be compared to the flatness of the pipe, that is the maximum and theoretical capacity of the pipe, while throughput is the actual amount of water that flows through per unit.

Unit	Abbrev.	Equivalent
Bits per second	bps	(Basic unit of bandwidth)
Kilobits per second	Kbps	1,000 bps
Megabits per second	Mbps	1,000 kbps / 1,000,000 bps
Gigabits per second	Gbps	1,000,000,000 bps / 1,000,000 Kbps / 1,000 Mbps

Therefore, even the bandwidth will set a limit on throughput. Throughput can be affected by a host of other factors. Throughput is usually measured in bits per second.

	BLE 4.2	BLE 5	BLE 5 Long Range (S=2)	BLE 5 Long range (S=8)
Connection speed	1 Mbps	2 Mbps	1 Mbps	1 Mbps
Network data rate	1 Mbps	2 Mbps	500 Kbps	125 Kbps
Data throughput	800 Kbps	1400 Kbps	380 Kbps	109 Kbps
Error Correction	None	None	FEC	FEC
Bluetooth 5 Requirement	Mandatory	Optional	Optional	Optional

In this project, we are actually going to measure the actual throughput of the value system in real life. In real life applications, the maximum

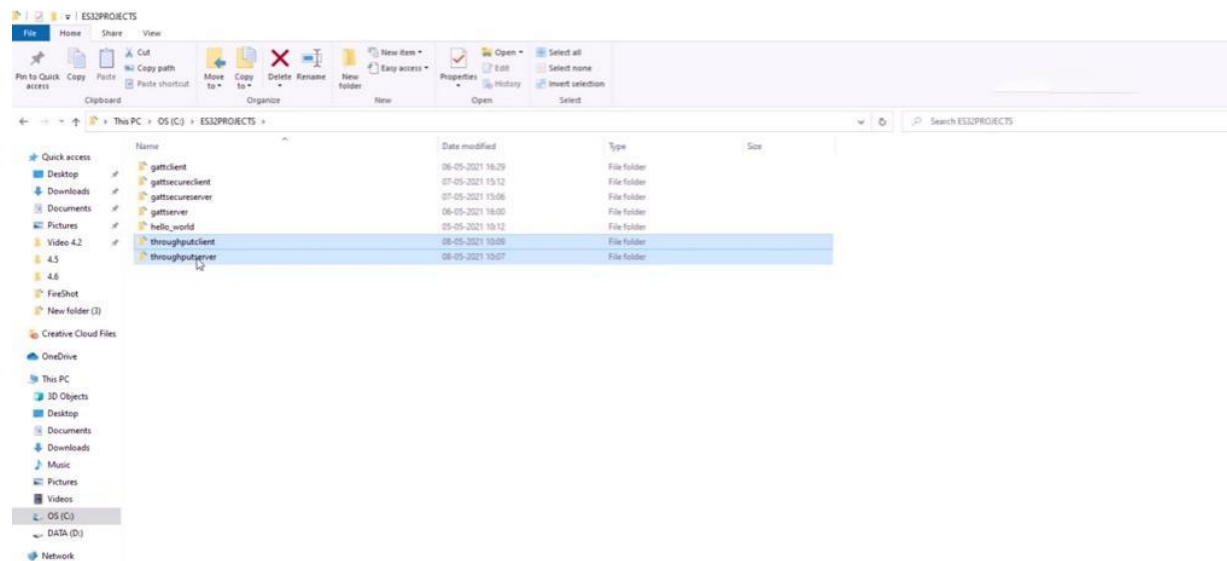
throughput won't reach the theoretical maximum throughput of eight hundred kilobits per second due to various reasons.

```
void throughput_server_task(void *param)
{
    #if (CONFIG_GATTS_NOTIFY_THROUGHPUT)
        if (!can_send_notify) {
            int res = xSemaphoreTake(gatts_semaphore, portMAX_DELAY);
            assert(res == pdTRUE);
        } else {
            if (is_connect) {
                int free_buff_num = esp_ble_get_cur_sendable_packets_num(g1_profile_tab[PROFILE_A_APP_ID].conn_id);
                if (free_buff_num > 0) {
                    for( ; free_buff_num > 0; free_buff_num--) {
                        esp_ble_gatts_send_indicate(g1_profile_tab[PROFILE_A_APP_ID].gatts_if, g1_profile_tab[PROFILE_A_APP_ID].conn_id,
                                                    g1_profile_tab[PROFILE_A_APP_ID].char_handle,
                                                    sizeof(indicate_data), indicate_data, false);
                    }
                } else { //Add the vTaskDelay to prevent this task from consuming the CPU all the time, causing low-priority tasks to not be executed at all.
                    vTaskDelay( 10 / portTICK_PERIOD_MS );
                }
            }
        }
    }

}

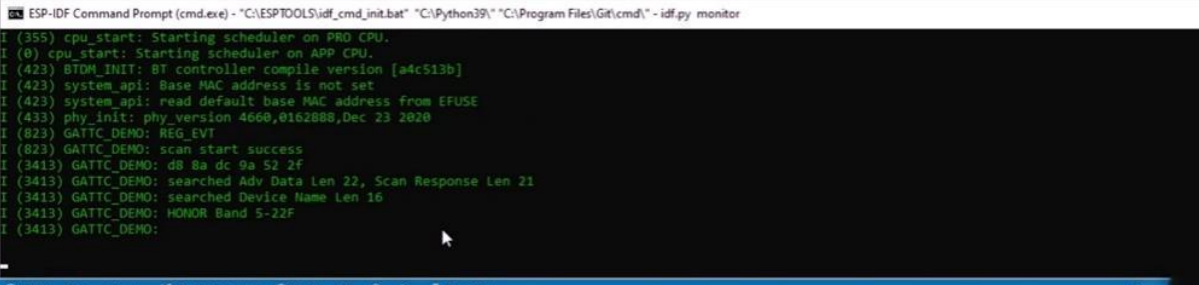
void throughput_client_task(void *param)
{
    #if (CONFIG_GATTS_NOTIFY_THROUGHPUT)
        vTaskDelay(2000 / portTICK_PERIOD_MS);
        if(is_connect){
            uint32_t bit_rate = 0;
            if (start_time) {
                current_time = esp_timer_get_time();
                bit_rate = notify_len * SECOND_TO_USECOND / (current_time - start_time);
                ESP_LOGI(GATTC_TAG, "Notify Bit rate - %d Byte/s, - %d bit/s, time - %ds",
                        bit_rate, bit_rate<<3, (int)((current_time - start_time) / SECOND_TO_USECOND));
            } else {
                ESP_LOGI(GATTC_TAG, "Notify Bit rate - 0 Byte/s, - 0 bit/s");
            }
        }
    }
}
```

The only change we need to do in the Gatesville and Gatland code is to add these functions on the server and timecode respectively.



Now, just like earlier open, each of the code in Visual Studio could connect both the ESP 32 to your PC.

```
11 * This is the demo to test the BLE throughput. It should be used together with throughput_se
12 *
13 *****
14
15 #include <stdint.h>
16 #include <string.h>
17 #include <stdbool.h>
18 #include <stdio.h>
19 #include "nvs.h"
20 #include "nvs_flash.h"
21
22 #include "esp_bt.h"
23 #include "esp_gap_ble_api.h"
```



ESP-IDF Command Prompt (cmd.exe) - "C:\ESPTOOLS\idf_cmd_init.bat" "C:\Python39\" "C:\Program Files\Git\cmd\" - idf.py monitor

```
(355) cpu_start: Starting scheduler on PRO CPU.
(0) cpu_start: Starting scheduler on APP CPU.
(423) BTDM_INIT: BT controller compile version [a4c513b]
(423) system_api: Base MAC address is not set
(423) system_api: read default base MAC address from EFUSE
(433) phy_init: phy_version 4660,0162888,Dec 23 2020
(823) GATT_DEMO: REG_EVT
(823) GATT_DEMO: scan start success
(3413) GATT_DEMO: dB 8a dc 9a 52 2f
(3413) GATT_DEMO: searched Adv Data Len 22, Scan Response Len 21
(3413) GATT_DEMO: searched Device Name Len 16
(3413) GATT_DEMO: HWMOR Band 5-22F
(3413) GATT_DEMO:
```

Then set the target device. Bill. And monitor the Gansevoort and Gatland called on the two year Peter, respectively. I will fast forward these steps as we have done these steps already in earlier project.

GATT Server	GATT Client
I (54118) GATT...	status I (10833) GATT_DEMO: Notif...
I (54128) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 3576 bit/s, time = 9s
I (54128) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (12833) GATT_DEMO: Notify Bit rate = 83543 Byte/s, = 68
I (54138) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 8344 bit/s, time = 11s
I (54138) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (14833) GATT_DEMO: Notify Bit rate = 84822 Byte/s, = 67
I (54148) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 8576 bit/s, time = 13s
I (54158) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (16833) GATT_DEMO: Notify Bit rate = 85761 Byte/s, = 68
I (54158) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 6088 bit/s, time = 15s
I (54168) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (18833) GATT_DEMO: Notify Bit rate = 86479 Byte/s, = 69
I (54168) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 1832 bit/s, time = 17s
I (54178) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (20833) GATT_DEMO: Notify Bit rate = 87046 Byte/s, = 69
I (54178) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 6368 bit/s, time = 19s
I (54178) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (22833) GATT_DEMO: Notify Bit rate = 87505 Byte/s, = 70
I (54188) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 0040 bit/s, time = 21s
I (54198) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (24833) GATT_DEMO: Notify Bit rate = 87885 Byte/s, = 70
I (54198) GATT_DEMO: ESP_GATTS_CONF_EVT,	status 3080 bit/s, time = 23s
I (54208) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (26833) GATT_DEMO: Notify Bit rate = 88204 Byte/s, = 705632 bit/s,
I (54208) GATT_DEMO: ESP_GATTS_CONF_EVT,	status ime = 25s
I (54178) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (28833) GATT_DEMO: Notify Bit rate = 88475 Byte/s, = 707800 bit/s,
I (54188) GATT_DEMO: ESP_GATTS_CONF_EVT,	status ime = 27s
I (54198) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (30833) GATT_DEMO: Notify Bit rate = 88710 Byte/s, = 709680 bit/s,
I (54198) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 29s
I (54208) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (32833) GATT_DEMO: Notify Bit rate = 88914 Byte/s, = 711312 bit/s,
I (54208) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 31s
I (54218) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (34833) GATT_DEMO: Notify Bit rate = 89093 Byte/s, = 712744 bit/s,
I (54218) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 33s
I (54228) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (36833) GATT_DEMO: Notify Bit rate = 89252 Byte/s, = 714016 bit/s,
I (54238) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 35s
I (54238) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (38833) GATT_DEMO: Notify Bit rate = 89393 Byte/s, = 715144 bit/s,
I (54248) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 37s
I (54248) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (40833) GATT_DEMO: Notify Bit rate = 89521 Byte/s, = 716168 bit/s,
I (54258) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 39s
I (54258) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (42833) GATT_DEMO: Notify Bit rate = 89635 Byte/s, = 717080 bit/s,
I (54268) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 41s
I (54268) GATT_DEMO: ESP_GATTS_CONF_EVT,	status I (44833) GATT_DEMO: Notify Bit rate = 89739 Byte/s, = 717912 bit/
I (54278) GATT_DEMO: ESP_GATTS_CONF_EVT,	status time = 43s

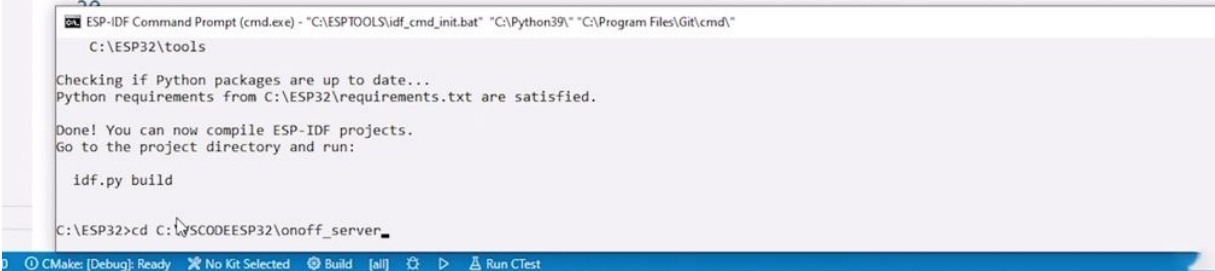
After the process is done, just like earlier, open to put determinists each for each E.S.P 32, the gas LSP 32 is on the left and the Gatland DP 32 is on the right for us to reset the Gatesville board. And after a few seconds reset the Gatland, the 32. Now let the correction complete and wait for some time for the throughput to be calculated. You can now indeed see that the throughput in both bytes per second and bits per second running it for some time reveals that it averages between 600 kilobits per second to 750 kilobits per second, which is the expected result in real life applications.

BLE MESH NODE, PROVISIONER _ CONSOLE IMPLEMENTATION

we are going to demonstrate the process of setting and operating a small ESP-BLE-MESH network of three nodes. This process will cover device provisioning and node configuration, and then sending on/off commands to Generic OnOff Server Models on specific nodes using the nRF Mesh app console. Before we go through the code explanation, let's see the project in action.

Let's set up the smart ESP network of three nodes and control the onboard ality based on the status of the node in the network.

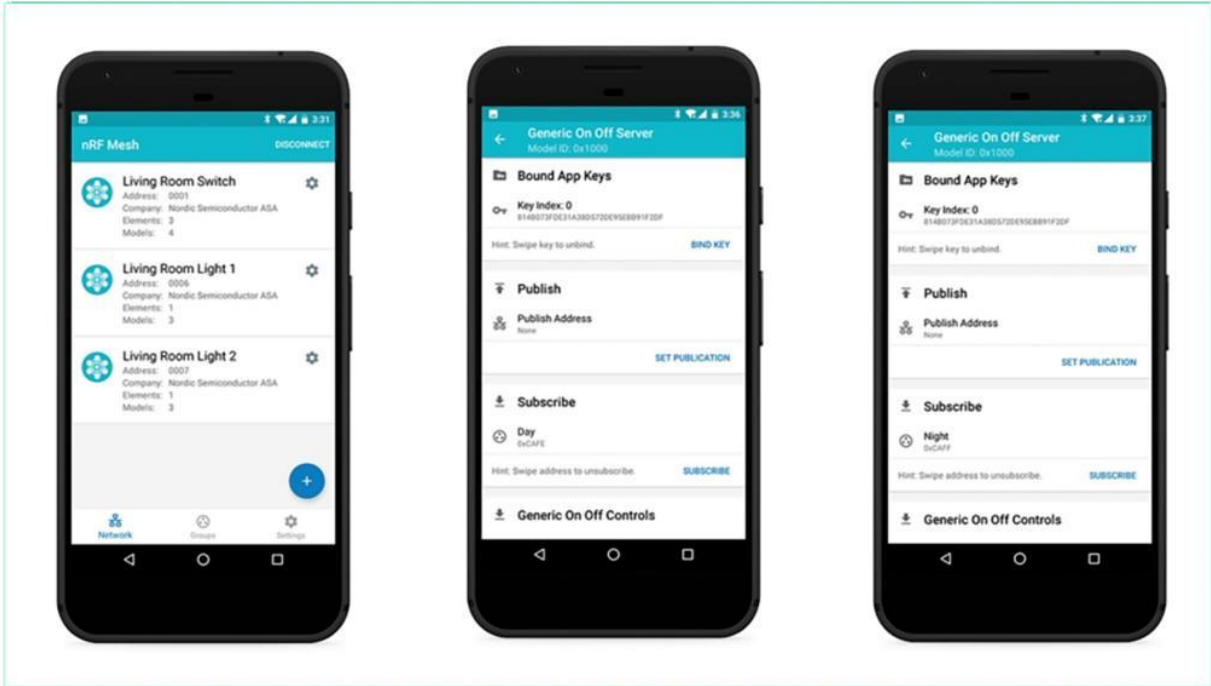

```
17 #include "esp_ble_mesh_common_api.h"
18 #include "esp_ble_mesh_networking_api.h"
19 #include "esp_ble_mesh_provisioning_api.h"
20 #include "esp_ble_mesh_config_model_api.h"
21 #include "esp_ble_mesh_generic_model_api.h"
22 #include "esp_ble_mesh_local_data_operation_api.h"
23
24 #include "board.h"
25 #include "ble_mesh_example_init.h"
26
27 #define TAG "EXAMPLE"
28
29 #define CID_ESP 0x02E5
```



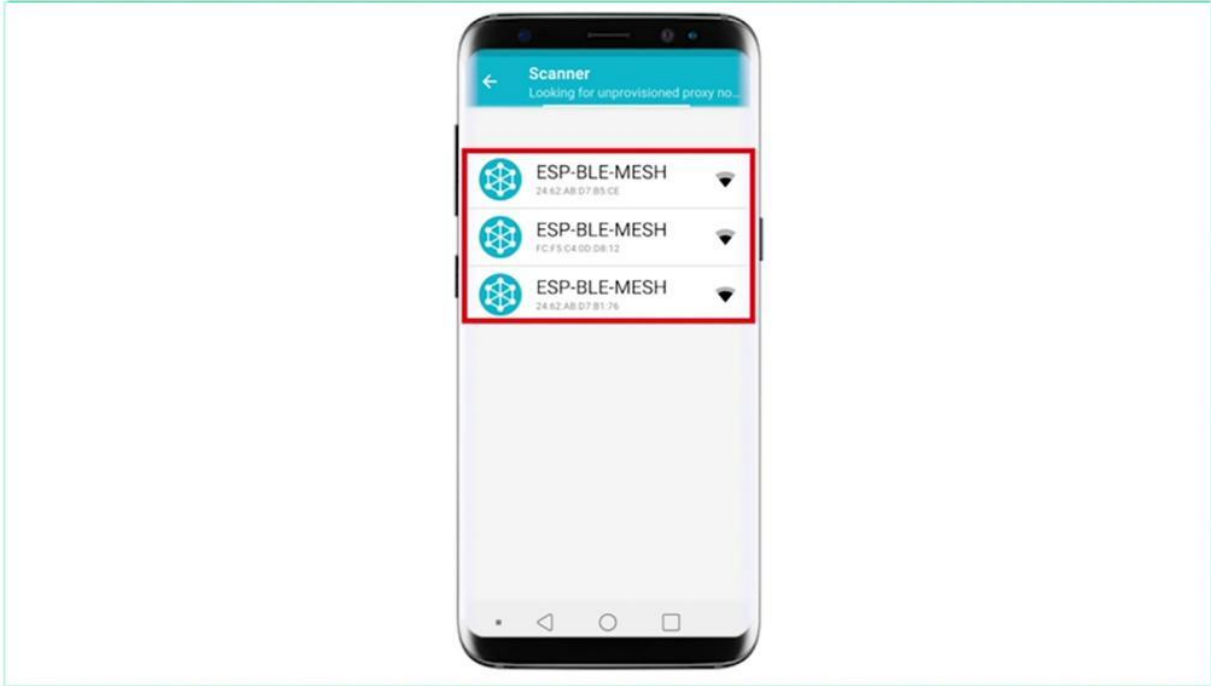
```
ESP-IDF Command Prompt (cmd.exe) - "C:\ESPTOOLS\idf_cmd_init.bat" "C:\Python39\cmd"
C:\ESP32\tools
Checking if Python packages are up to date...
Python requirements from C:\ESP32\requirements.txt are satisfied.
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:
idf.py build
C:\ESP32>cd C:\SCODEESP32\onoff_server_
```

open the Golden Visual Studio Code and load the main source file. Keep the E.S.P idea of terminal open and change the directory to the project folder. Now, as always, after connecting your ISP 32 set the target will the code and flash it. Now, disconnect the 32 and connect the second E.S.P third to unflashy, do the same for the dirtiest 32 also now power all the boats using power banks.

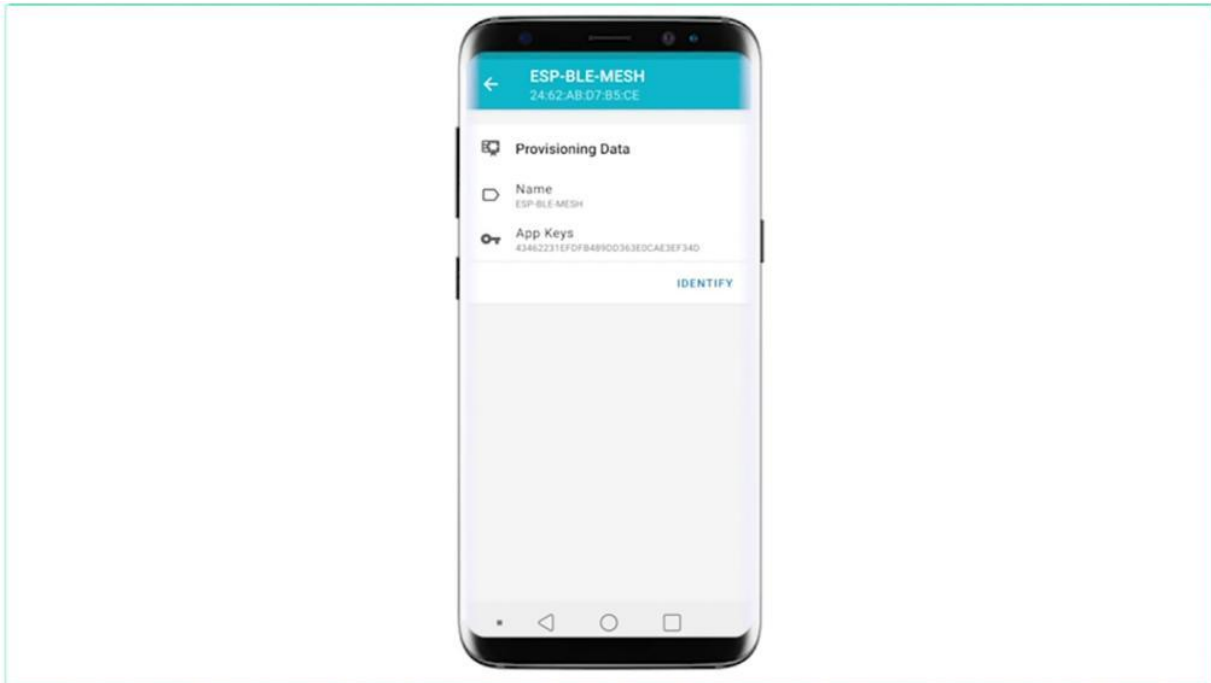
We will use the end of Android app to demonstrate how to prove an unproven device. You will need to download and install the NRA app to your smartphone. Also, this is available for both Android and iOS.



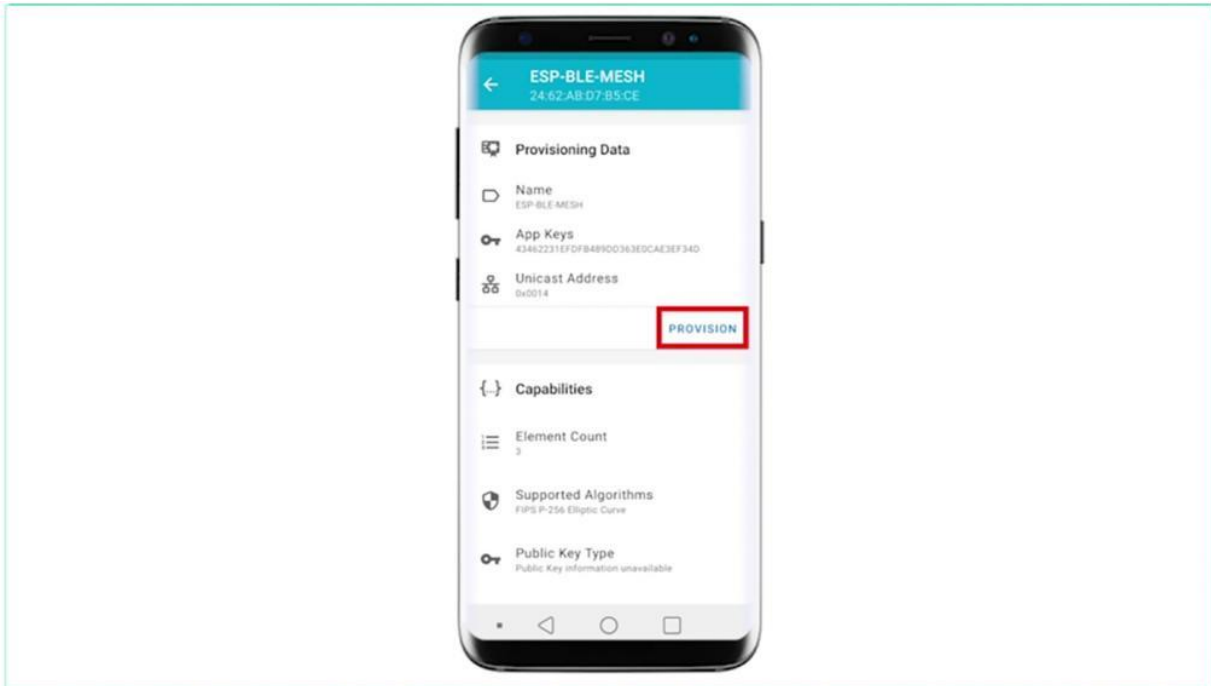
Now, after making sure that all the tools are powered or make sure that all the three ladies are on, this signifies that the devices are configured as nodes and are waiting to be broken into a mesh network. The code is configured in such a way that once the nodes are scanned, identified and provisioned into the network, the blue ality will be switched off.



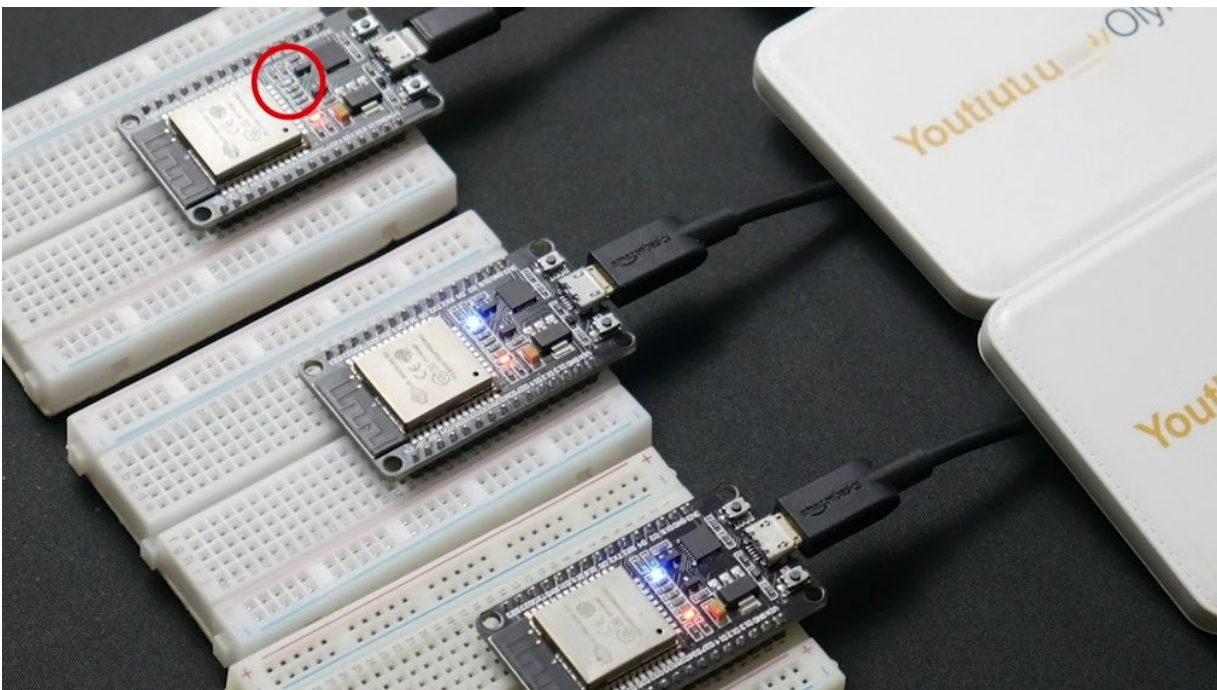
Open the inner mesh application on your smartphone and click the plus icon to scan for unproven devices in the range. You can see that the three elements enabled unbroken nodes indeed pop up in the can. Users can select any unproven device then the app and try to set up a connection with this device.



Click on the first and provisioned E.S.P 32 node to connect to it after the BEARLY connection is established successfully and proper ABCs have set, users can see the identify interface button on the screen.

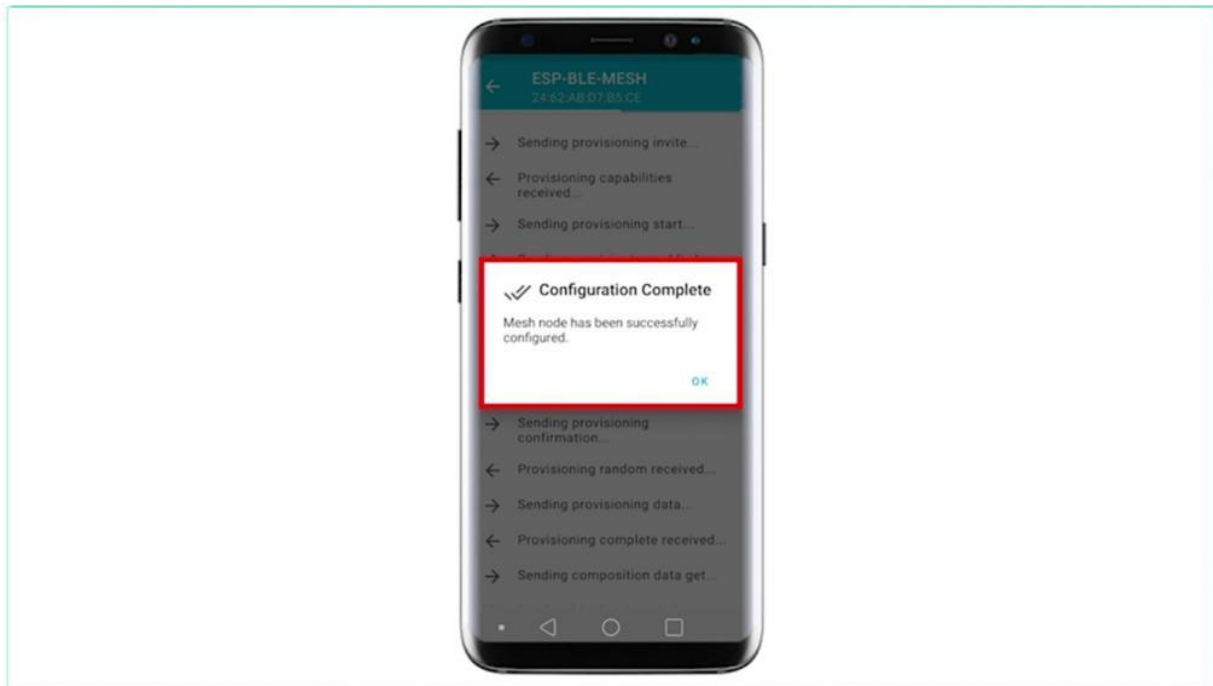


The identify operation can be used to tell users which device is going to be provisioned. Now it loads all the capabilities of the node. Also, you can see that the provision option also pops up in the interface. Click this button and now the app will try to position the unbroken device.

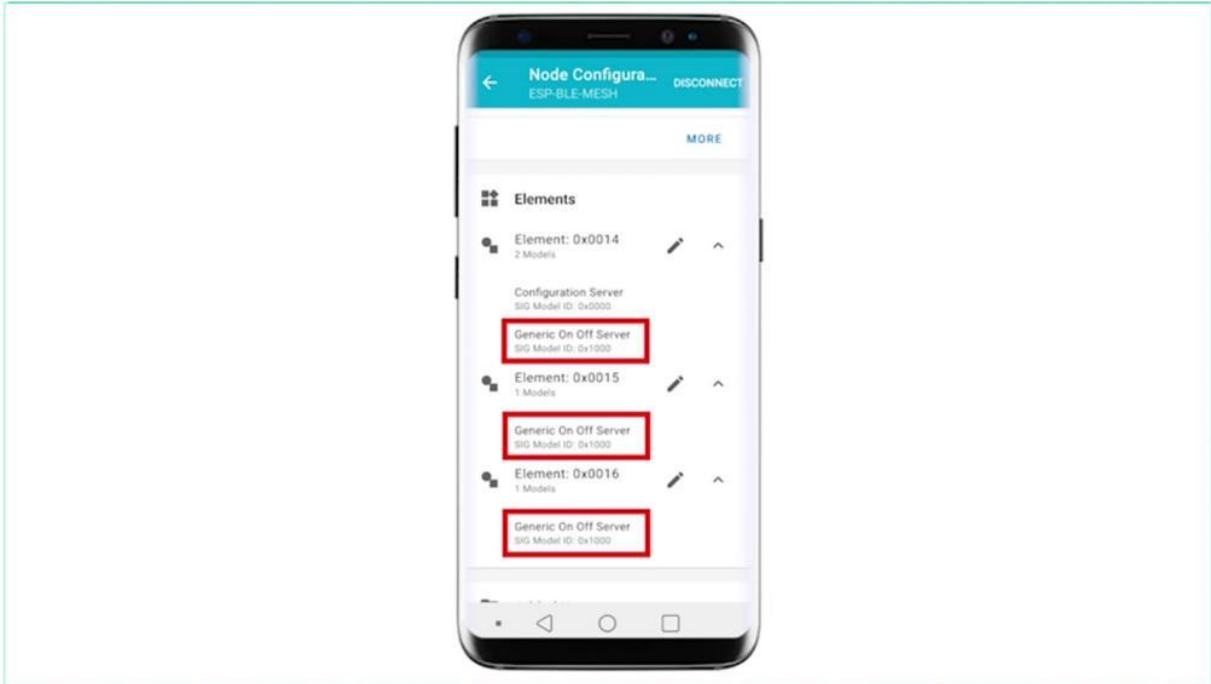


When the devices provision successfully, the blue lady on the board will turn off and the app will implement the following procedures, disconnect with the note, try to reconnect with the note, connect successfully and discover SBB Gazzo, get composition data of the note and add up key to it. But all the procedures are finished.

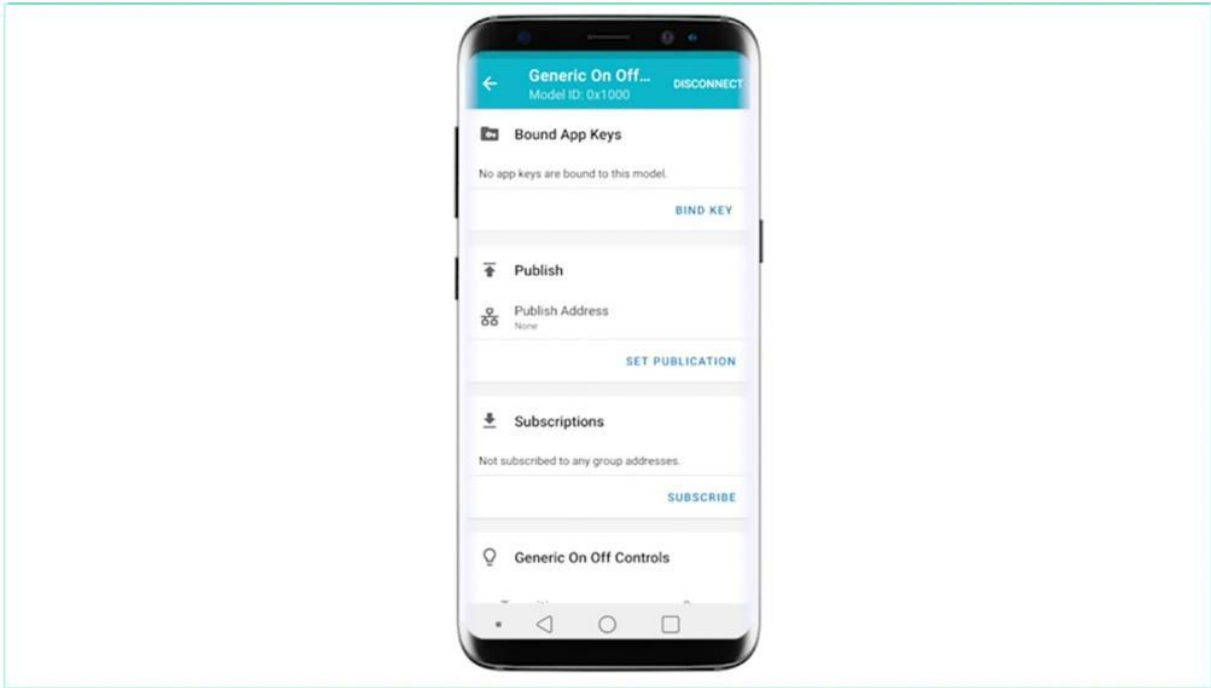
The note is configured properly and you will get the following message.



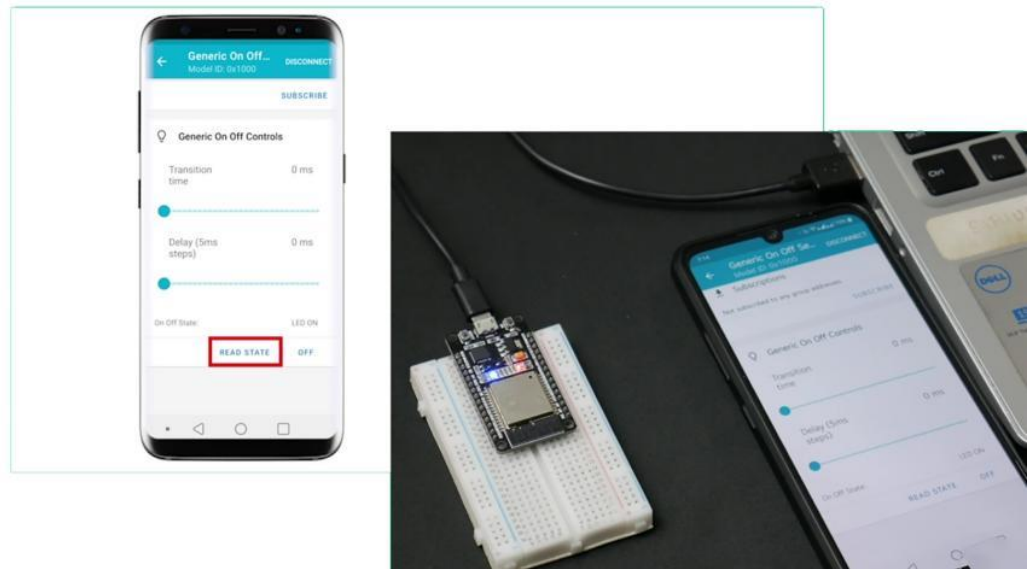
Now you can proceed, okay. And when you click, more users can see that the unicast address is assigned and the elements and its more details are decoded along with the network and the app. Repeat this whole process for the rest of the to be notes. When provisioning and initial configuration of all nodes are finished, users can start to configure the node.



For example, let's go to one of the provisioned nodes in the network and open elements to view the generic ARNAV server models under different elements, the generic ARNAV server model that comes under the second element is mapped to the blue alyda on board now to bind the application key to this click binkie and select the key.



Now scroll down to the controls console here. You can read it and control the state of any on plurality or any ESP 32 node in the same mesh network.



When you click this button you can see that the concerns date changing. They lead to happen. On top of that, we are also able to read the state of the ality through this console as the project is over.

```
void app_main(void)
{
    int err;

    ESP_LOGI(TAG, "Initializing...");

    board_init();

    err = bluetooth_init();

    if (err) {
        ESP_LOGE(TAG, "esp32_bluetooth_init failed (err %d)", err);
        return;
    }

    /* Initializes the Bluetooth Mesh Subsystem */
    err = ble_mesh_init();
    if (err) {
        ESP_LOGE(TAG, "Bluetooth mesh init failed (err %d)", err);
    }
}
```

Now let's discuss the logic behind the code. The main loop takes care of the E.S.P 32 system initialization, specifically Bluetooth and the initialization as shown here.

```
static esp_err_t ble_mesh_init(void)
{
    int err = 0;

    memcpy(dev_uuid + 2, esp_bt_dev_get_address(), BLE_MESH_ADDR_LEN);

    // See comment 1
    esp_ble_mesh_register_prov_callback(esp_ble_mesh_prov_cb);
    esp_ble_mesh_register_custom_model_callback(esp_ble_mesh_model_cb);

    err = esp_ble_mesh_init(&provision, &composition);
    if (err) {
        ESP_LOGE(TAG, "Initializing mesh failed (err %d)", err);
        return err;
    }

    esp_ble_mesh_node_prov_enable(ESP_BLE_MESH_PROV_ADV | ESP_BLE_MESH_PROV_GATT);

    ESP_LOGI(TAG, "BLE Mesh Node initialized");

    board_led_operation(LED_G, LED_ON);

    return err;
}
```

The code block below shows the declaration of an element structure. This code implements and defines a modern structure.

```
/** Abstraction that describes a Mesh Model instance.
 * This structure is associated with bt_mesh_model in mesh_access.h
 */
struct esp_ble_mesh_model {
    /* Model ID */
    union {
        const uint16_t model_id;
        struct {
            uint16_t company_id;
            uint16_t model_id;
        } vnd;
    };

    /* The Element to which this Model belongs */
    esp_ble_mesh_elem_t *element;

    /* Model Publication */
    esp_ble_mesh_model_pub_t *const pub;

    /* AppKey List */
    uint16_t keys[CONFIG_BLE_MESH_MODEL_KEY_COUNT];

    /* Subscription List (group or virtual addresses) */
    uint16_t groups[CONFIG_BLE_MESH_MODEL_GROUP_COUNT];

    /* Model operation context */
    esp_ble_mesh_model_op_t *op;

    /* Model-specific user data */
    void *user_data;
};
```


The next code block represents the definition of the model structure, and the router underscored more than is used for indicating the number of existing model structures.

```
static esp_ble_mesh_model_t root_models[] = {
    ESP_BLE_MESH_MODEL_CFG_SRV(&config_server),
    ESP_BLE_MESH_SIG_MODEL(ESP_BLE_MESH_MODEL_ID_GEN_ONOFF_SRV, onoff_op,
        &onoff_pub, &led_state[0]),
};
```

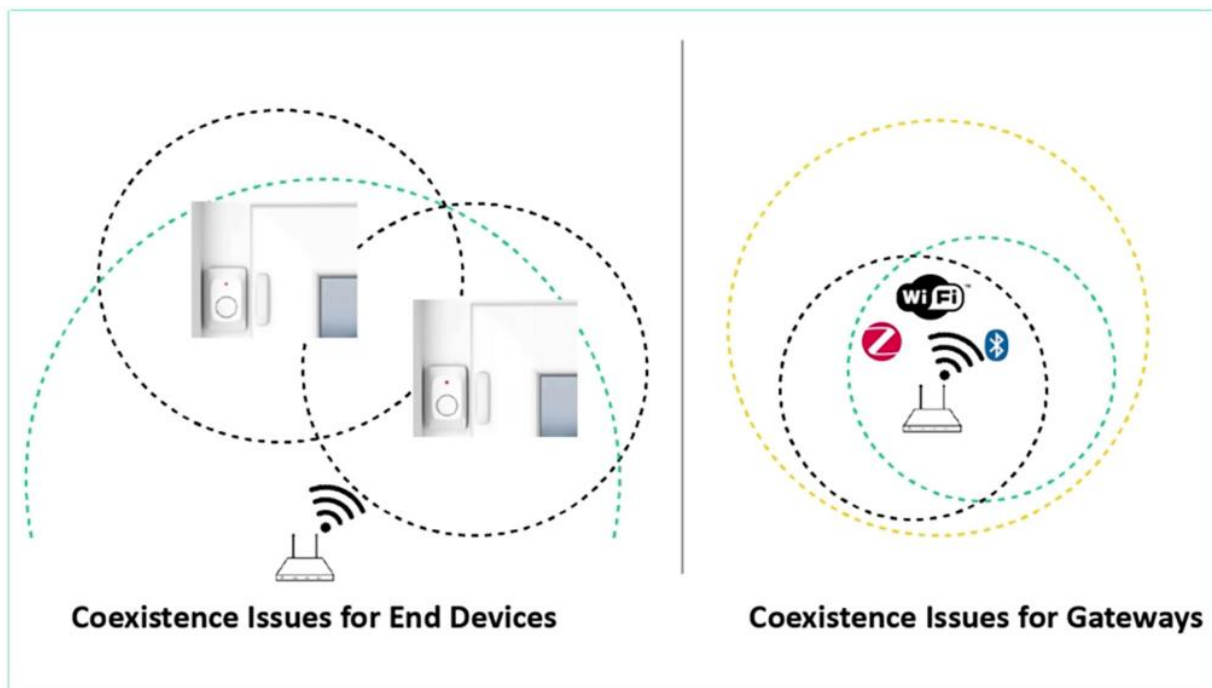
This block shows the operation structure defined by the On-Off server in this demo that covers the basic overview of the code.

```
static esp_ble_mesh_model_op_t onoff_op[] = {
    { ESP_BLE_MESH_MODEL_OP_GEN_ONOFF_GET, 0, 0},
    { ESP_BLE_MESH_MODEL_OP_GEN_ONOFF_SET, 2, 0},
    { ESP_BLE_MESH_MODEL_OP_GEN_ONOFF_SET_UNACK, 2, 0},
    /* Each model operation struct array must use this terminator
     * as the end tag of the operation uint. */
    ESP_BLE_MESH_MODEL_OP_END,
};
```

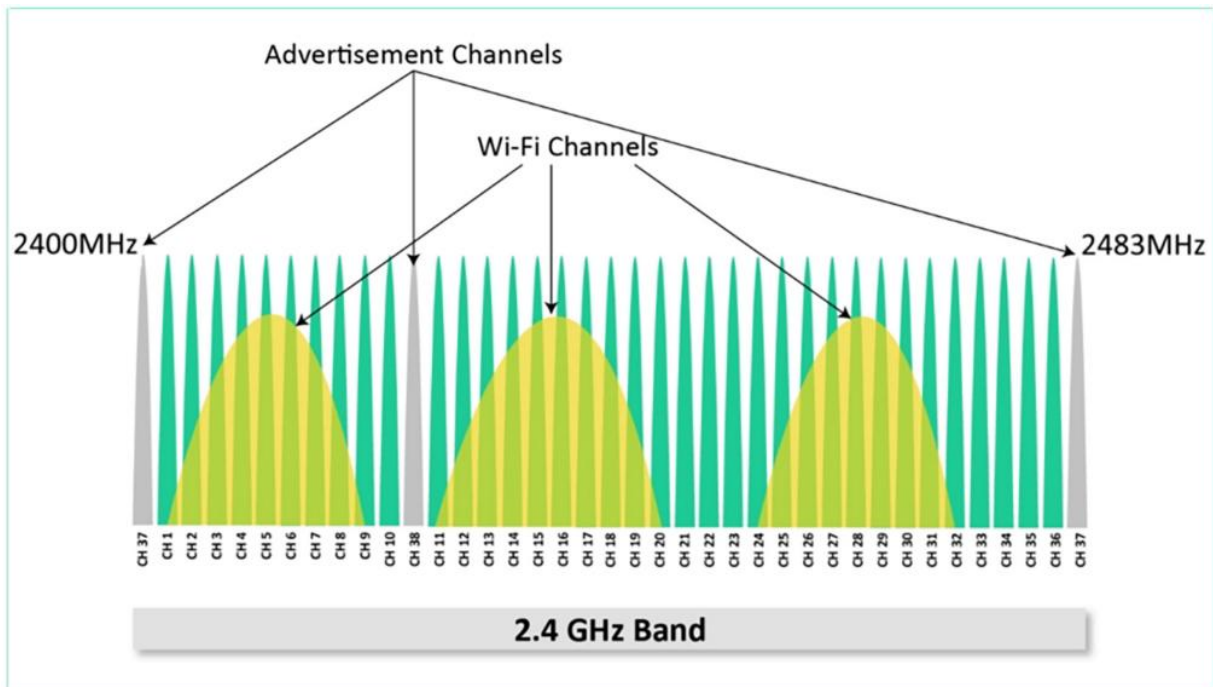
If you want to learn more about the code.

BLE MESH WI-FI COEXISTENCE IMPLEMENTATION

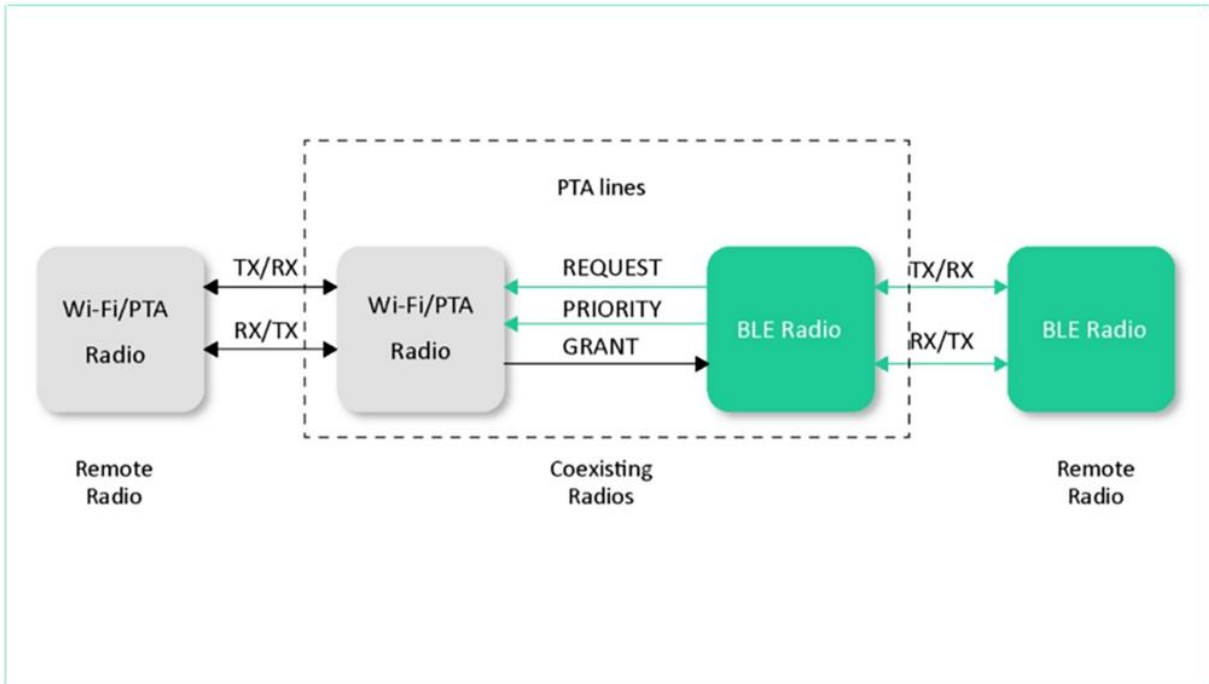
we will cover the following topics What is Wireless Coexistence What is the need for BLE MESH Wi-Fi Coexistence Implementation of Project using Blufi, to set network parameters using BLE



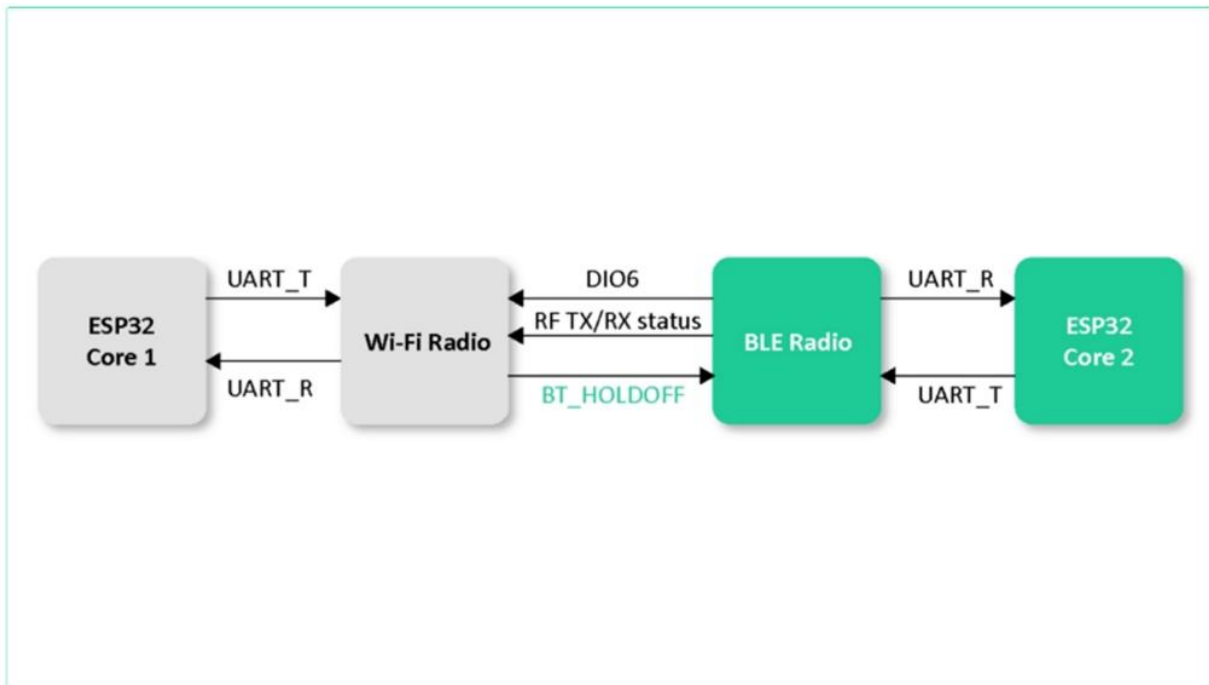
Wireless coexistence is the ability of one system to perform a task in a given shared environment where other systems performing tasks may, or may not, be using the same set of rules or standards. The ESP32 has both Wi-Fi and BLE capabilities on the same chip thus as there are different types of wireless networks operating in the same unlicensed frequency band, it is possible for two different networks to interfere with each other. This interference can cause transmission failures in both networks. There is no standardized algorithm to achieve the coexistence of two different wireless networks.



As we have already covered, This interference can cause transmission failures in both networks. There is no standardized algorithm to achieve the coexistence of two different wireless networks. As we have already covered, This interference can cause transmission failures in both networks. There is no standardized algorithm to achieve the coexistence of two different wireless networks. As we have already covered, we have already covered. BLE works in 2.4- to the 2.483-GHz frequency band Thus, let's say that there is an application in which bearyl and Wi-Fi have to work together, which is often the case in Iot applications. In such cases, by common sense, we would assume that there would be a lot of indifference. But this is not the case. This is because most other devices that interfere with bearyl are bands, bearyl advertising channels in particular. We are placed between Wi-Fi channels in such a way so as to avoid Wi-Fi signal. The wide spacing of the advertisement is held barely better manage the indifference from white flight. This covers the concept of violent coexistence in the context of SB 32,

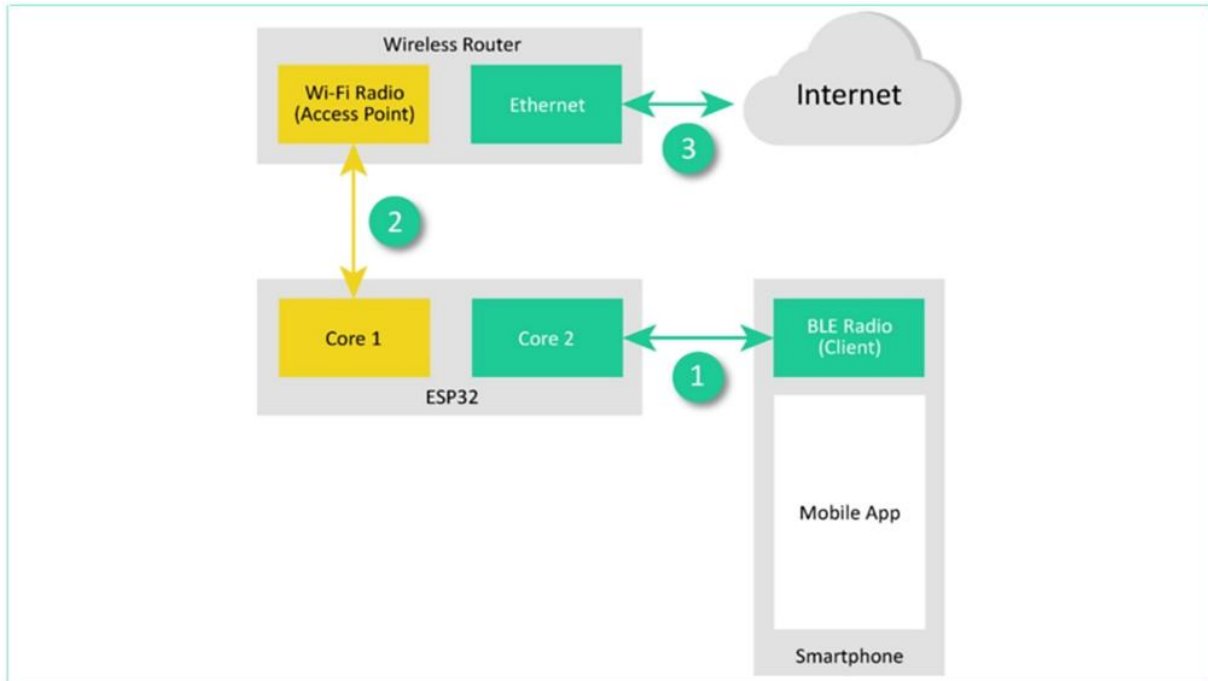


SB 32 allows you to use Wi-Fi and full BEARLY capabilities, which includes mesh capabilities at the same time. This is possible in SB 32 due to two reasons.

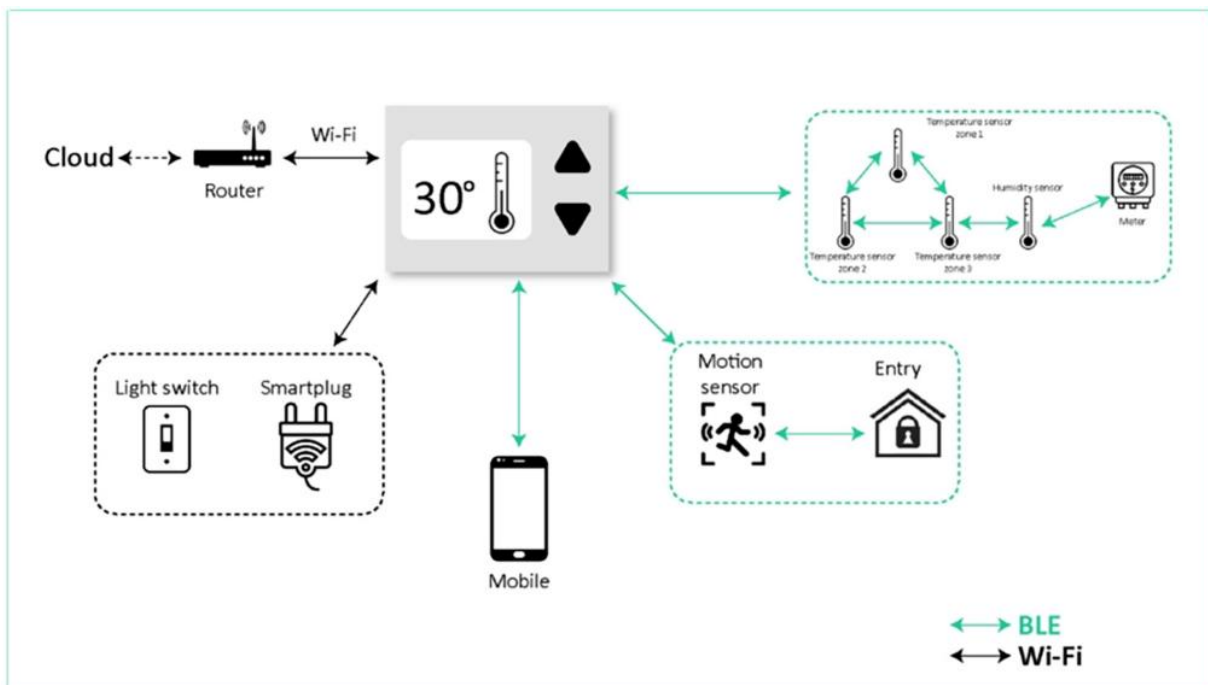


The first reason is that the SB 32 chip contains Ducos, thereby allowing us to run bearly related task on one goal while the Wi-Fi related task on the

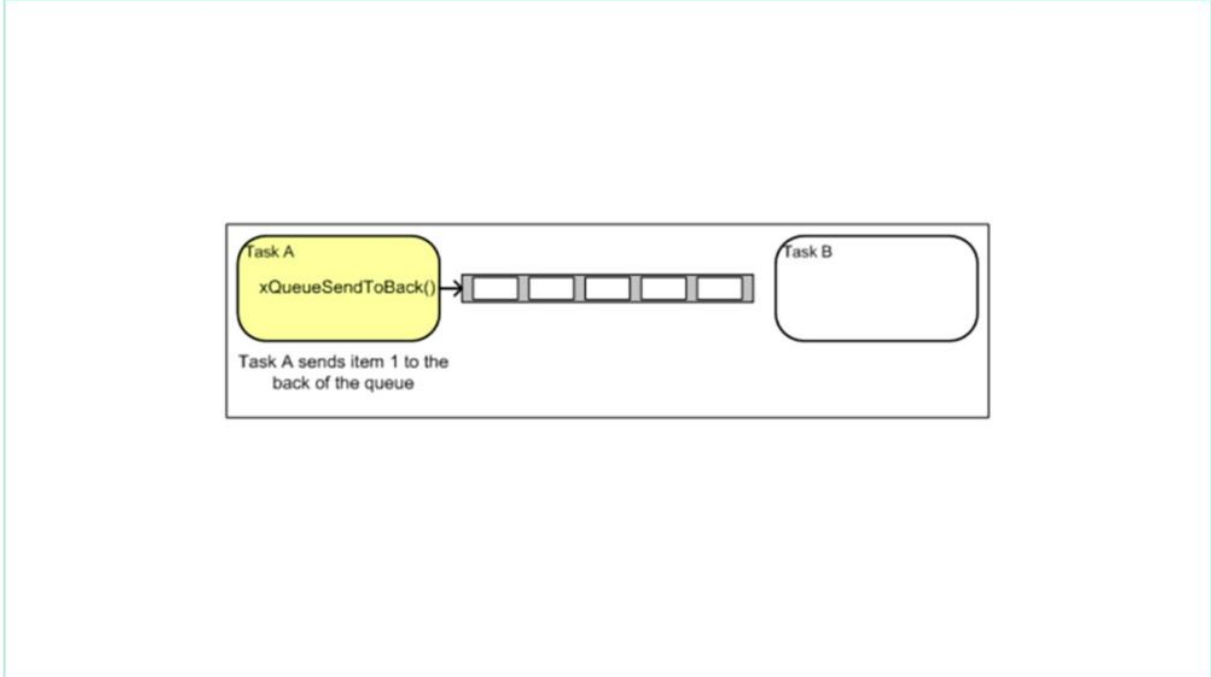
other.



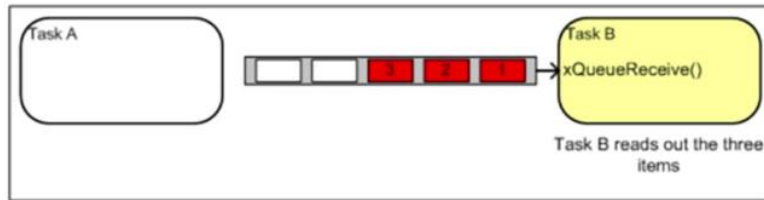
The second reason is that SB 32 supports free Arktos, thereby allowing us to do in their task communication between concurrently running tasks which here can be the Wi-Fi and the bialy tasks



in most of the BEARLY network be the normal one. Automation enabled one at one point or another. We may want to stream the packets to a cloud service in such cases instead of implementing a separate bearly to cloud gateway. We can use the coexistence functionality to convert a node to a gate gateway.



The inverters communication can convert the yearly packets to IP packets for the wi fi task running on the second goal, which will then handle the cloud connection.



Another application for coexistence involves scenarios in which the nodes have the ability to switch between the Wi-Fi obviously implementation to conserve power and improve battery life.

We can have a general overview of the code before flashing and implementing it,

```
void app_main(void)
{
    esp_err_t err;

    ESP_LOGI(TAG, "Initializing...");

    err = board_init();
    if (err) {
        ESP_LOGE(TAG, "board_init failed (err %d)", err);
        return;
    }

    err = bluetooth_init();
    if (err) {
        ESP_LOGE(TAG, "esp32_bluetooth_init failed (err %d)", err);
        return;
    }

    /* Initialize the Bluetooth Mesh Subsystem */
    err = ble_mesh_init();
    if (err) {
        ESP_LOGE(TAG, "Bluetooth mesh init failed (err %d)", err);
        return;
    }

    wifi_console_init();
}
```

the code block shown here, first initializes the board, then it's Bluetooth related functions, including the Bluetooth and bearyl mesh, and then the Wi-Fi cancel.

```
led_action_queue = xQueueCreate(60, sizeof(struct _led_state));  
ret = xTaskCreate(led_action_thread, "led_action_thread", 4096, NULL, 5, NULL);
```

This code creates a task for controlling the status of the ality and then creates a queue for storing status data.

```
static void led_action_thread(void *arg)  
{  
    struct _led_state led = {0};  
  
    while (1) {  
        if (xQueueReceive(led_action_queue, &led, (portTickType)portMAX_DELAY)) {  
            ESP_LOGI(TAG, "%s: pin 0x%04x onoff 0x%02x", __func__, led.pin, led.current);  
            /* If the node is controlled by phone, add a delay when turn on/off led */  
            if (fast_prov_server.primary_role == true) {  
                vTaskDelay(50 / portTICK_PERIOD_MS);  
            }  
            gpio_set_level(led.pin, led.current);  
        }  
    }  
}
```


The ality action task continuously sets the status of the leading indicator by using the status data obtained from the queue.

```
static esp_err_t ble_mesh_init(void)
{
    esp_err_t err;

    /* First two bytes of device uuid is compared with match value by Provisioner */
    memcpy(dev_uuid + 2, esp_bt_dev_get_address(), 6);

    esp_ble_mesh_register_prov_callback(example_ble_mesh_provisioning_cb);
    esp_ble_mesh_register_custom_model_callback(example_ble_mesh_custom_model_cb);
    esp_ble_mesh_register_config_client_callback(example_ble_mesh_config_client_cb);
    esp_ble_mesh_register_config_server_callback(example_ble_mesh_config_server_cb);

    err = esp_ble_mesh_init(&prov, &comp);
    if (err != ESP_OK) {
        ESP_LOGE(TAG, "%s: Failed to initialize BLE Mesh", __func__);
        return err;
    }
    ...
    return ESP_OK;
}
```

Now the functionality is initialized using this set of then the code calls the Wi-Fi related functions.

```
initialise_wifi();
initialize_console();

/* Register commands */
esp_console_register_help_command();
register_wifi();
```

It first initializes the Wi-Fi, then enables the Wi-Fi console, and then enables some debugging related function for wi fi. This covers the basic overview of the code.

```
53 static uint8_t dev_uuid[16] = { 0xdd, 0xdd };
54 static uint8_t prov_start_num = 0;
55 static bool prov_start = false;                                     = _OP_FAST_
56
57 static const esp_ble_mesh_client_op_pair_t fast_prov_cli_op_pair[] = {
58     { ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_INFO_SET,      ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_I
59     { ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_NET_KEY_ADD,  ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_N
60     { ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_NODE_ADDR,    ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_N
61     { ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_NODE_ADDR_GET, ESP_BLE_MESH_VND_MODEL_OP_FAST_PROV_N
62 };
63
64 /* Configuration Client Model user_data */
65 esp_ble_mesh_client_t config_client;
```



```
ESP-IDF Command Prompt (cmd.exe) - "C:\ESPTOOLS\idf_cmd_init.bat" "C:\Python39" "C:\Program Files\Git\cmd" - idf.py build
-- Build files have been written to: C:\VSCODEESP32\ble_mesh_wifi_coexist\build
C:\VSCODEESP32\ble_mesh_wifi_coexist>idf.py build
Executing action: all (aliases: build)
Running ninja in directory c:\vscodesp32\ble_mesh_wifi_coexist\build
Executing "ninja all"...
[43/1292] Generating ../partition_table/partition-table.bin
Partition table binary generated. Contents:
*****
# ESP-IDF Partition Table
# Name, Type, SubType, Offset, Size, Flags
nvs_data,data,nvs,0x9000,16K,
otadata,data,ota,0xd000,8K,
phy_init,data,phy,0xf000,4K,
factory_app,factory,0x10000,2M,
*****
```

Now please download the code from the resources of this vide . Open the fast code in VS code, and load the main source file. Keep the ESP IDF terminal open, and change directory to the project folder Now as always after connecting your ESP32, set the target, build the code, and flash it. Now as always after connecting your ESP32, set the target, build the code, and flash it. Build the code and flash it.

```
COM - PUTTY
I (777) BTDM_INIT: BT controller compile version [a4c513b]
I (778) system_api: Base MAC address is not set
I (779) system_api: read default base MAC address from EFUSE
I (787) phy_init: phy_version 4660,0162888,Dec 23 2020
I (3582) EXAMPLE: ESP_BLE_MESH_PROV_REGISTER_COMP_EVT, err_code: 0
I (3589) EXAMPLE: ESP_BLE_MESH_NODE_PROV_ENABLE_COMP_EVT, err_code: 0
I (3590) EXAMPLE: BLE Mesh Wi-Fi Coexist Node initialized
I (3601) wifi_init: rx ba win: 32
I (3602) wifi_init: tcpip mbox: 64
I (3602) wifi_init: udp mbox: 64
I (3605) wifi_init: tcp mbox: 64
I (3609) wifi_init: tcp tx win: 65534
I (3614) wifi_init: tcp rx win: 65534
I (3618) wifi_init: tcp mss: 1440
I (3622) wifi_init: WiFi IRAM OP enabled
I (3627) phy_init: phy_version 4660,0162888,Dec 23 2020

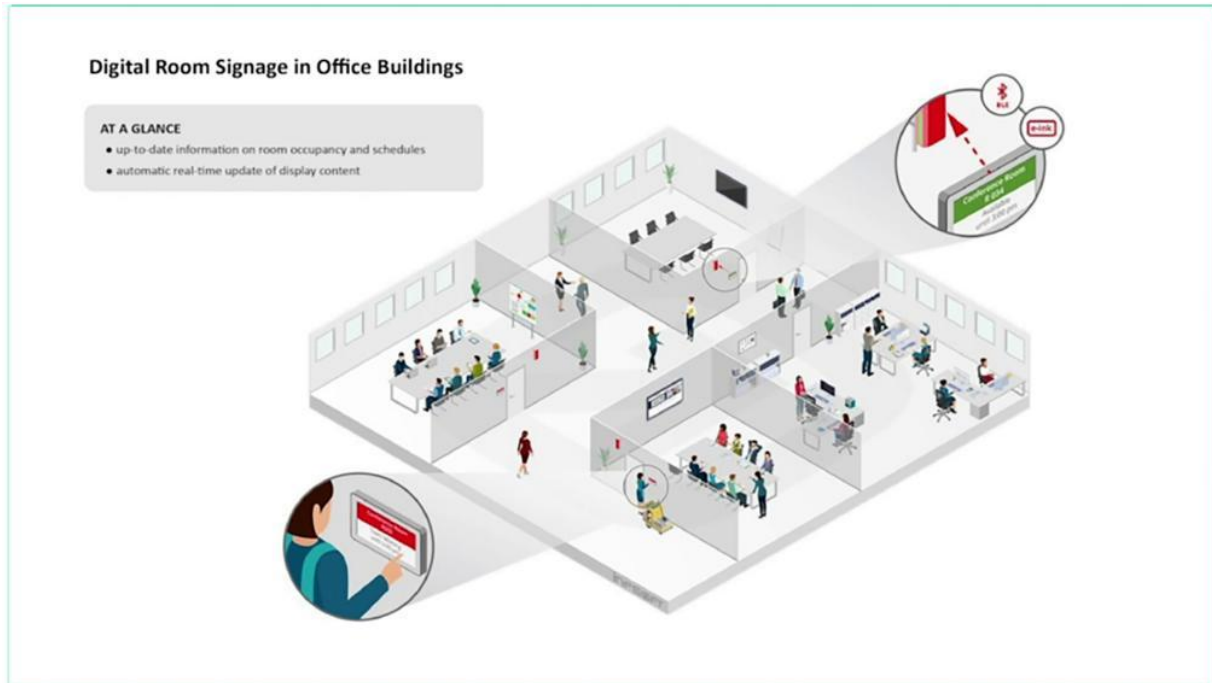
Type 'help' to get the list of commands.
Use UP/DOWN arrows to navigate through command history.
Press TAB when typing command name to auto-complete.

=====
|           Steps to test WiFi throughput           |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1. Print 'help' to gain overview of commands   |
| 2. Configure device to station or soft-AP      |
| 3. Setup WiFi connection                       |
| 4. Run iperf to test UDP/TCP RX/TX throughput |
|-----|-----|-----|-----|-----|-----|-----|-----|
|                                     1                                     |
esp> sta Work
I (14487) iperf: sta connecting to 'Work'
I (15592) iperf: L2 connected
I (17094) esp_netif handlers: sta ip: 192.168.0.100, mask: 255.255.255.0, gw: 192.168.0.1
I (17096) iperf: goT ip
esp> █
```

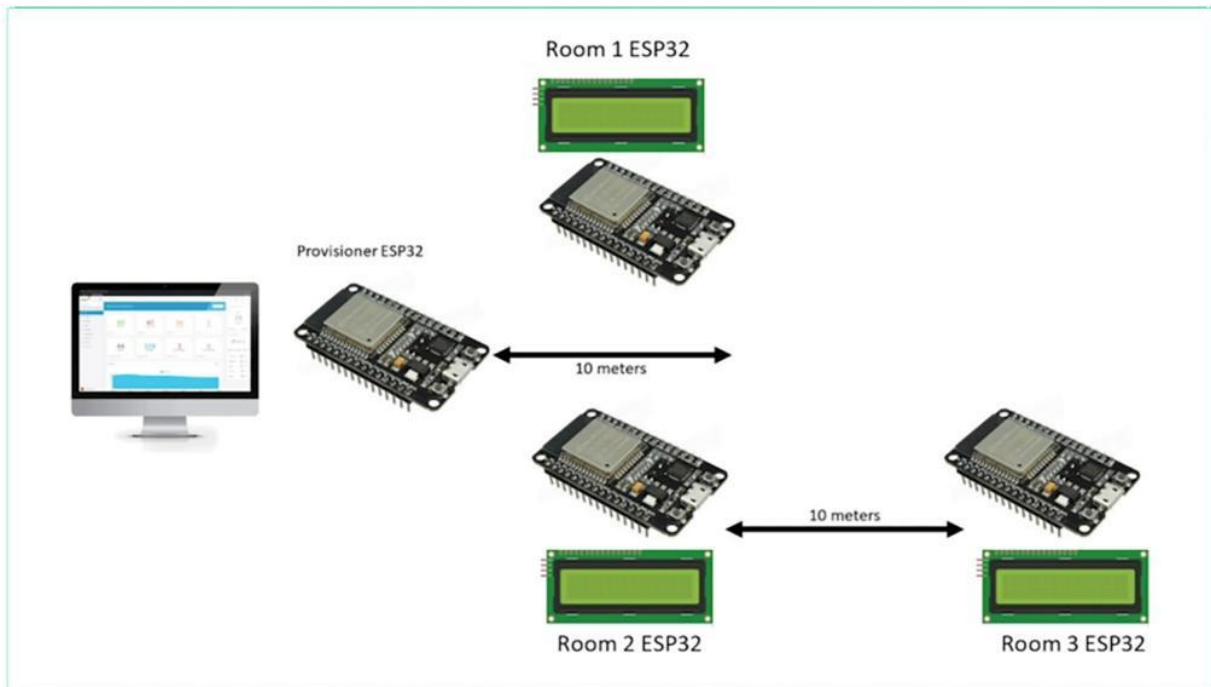
After doing this, open the serial port of the ESP32 by using PuTTY. Now enter the SSID and Password of the access point you wish to connect to. You can open the end of mishap scan for the device first and provision it after provisioning. After this is done, start a new TCP server by entering the following command in your serial port. This will print the current transfer rate of the Wi-Fi network the board connects to While this is happening you can open the nRF Mesh app, scan for the device first and provision it. After provisioning you can do the proper binding procedure and then control the onboard LED, while the TCP server is running. Just click the on button and you can see the Onboard LED light up.

ARCHITECTURE OVERVIEW

Digital Room Labeling is increasingly becoming popular as customers are increasingly benefiting from numerous possibilities of electronic shelf labeling technology.



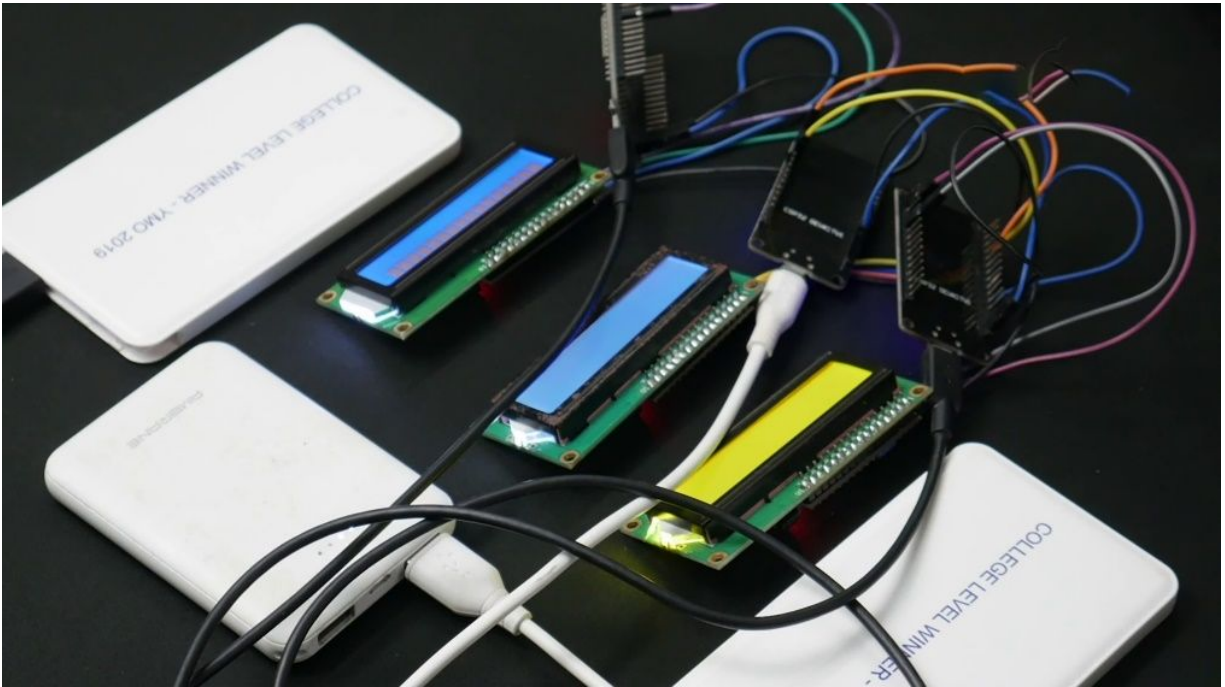
In addition to digital room management, digital room signage can be found in a wide variety of areas. These include, among others, industry, trade, events, museums and tourism. The use of digital room signage in meeting rooms, conference rooms, medical facilities, hotels or restaurants can greatly facilitate the day-to-day work of employees as well as customers. Within seconds, new bookings or reservations are displayed on the digital labels, which are attached to the meeting rooms. This way your employees and your guests are always up to date. This visit, your employees and your guests are always up to date. We will implement a project using 4 ESP32 to make a Mesh Enabled Notice Board System for Hotels. In this we assume that there are 3 rooms that need to be digitally labelled.



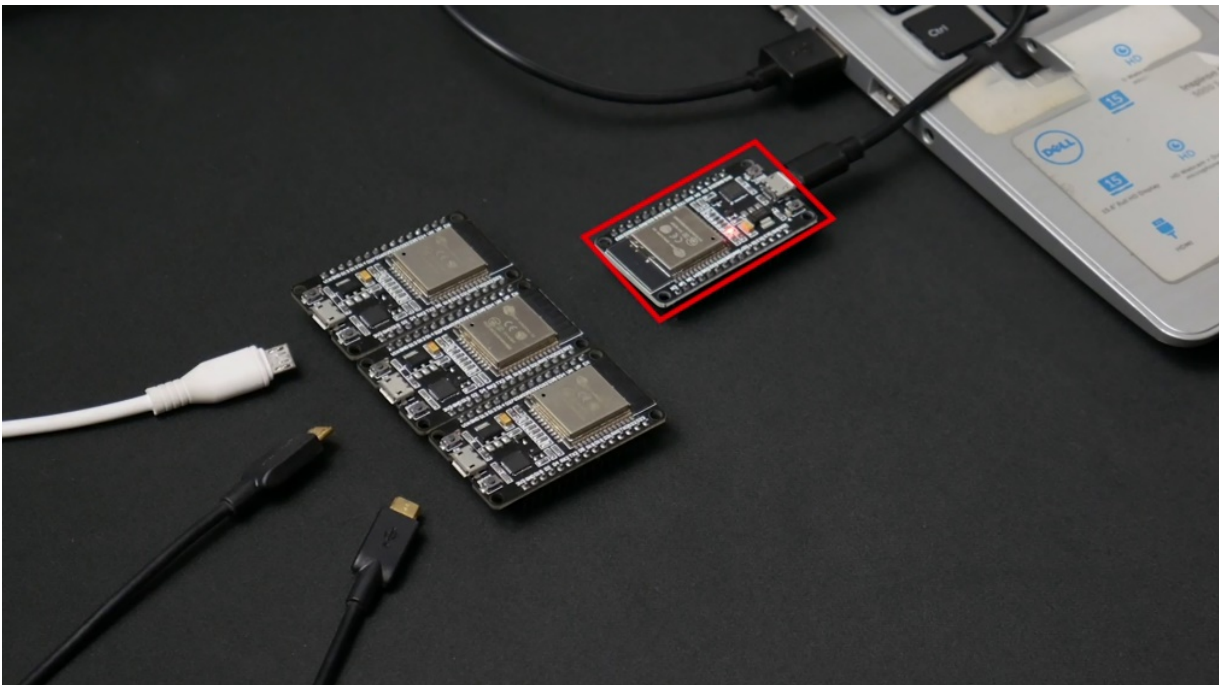
To enable labelling, we will interface a 16x2 LCD with each of 3 ESP32s for the room. The information on each of the LCDs can be changed from a central provisioner. The information on each of the LCDs can be changed from a central provisioner ESP32 which is accessed by the receptionist of the hotel. In this project we assume that the central provisioner at any point in time only is in the range of a maximum of 2 ESP32 Client nodes. In this project, we assume that the sender provisioner at any point in time or place is in the range of a maximum of two E.S.P. The Deucalion notes. This configuration helps us to explore and leverage the BLE Mesh capabilities of the system.



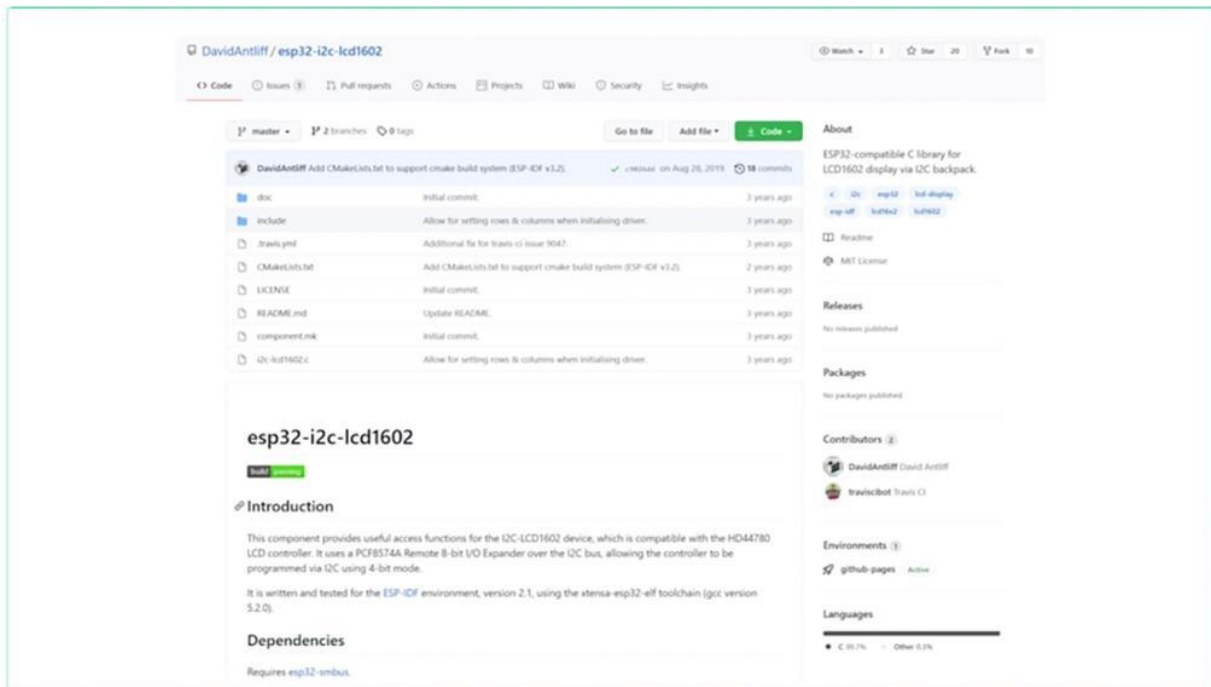
In the last section, as we have already worked with the sensor for the implementation, this project gives you the opportunity as we have already worked with a sensor, for the BLE Mesh implementation, this project gives you the opportunity to work with an actuator.



Furthermore, this project will help you understand the BLE Mesh provisioning process in an actual project, and the way the BLE messages are moving inside the mesh network. Now let's look at the hardware requirements for this project. You will need a total of 4 ESP32 for this project. Make sure all of them are similar variants from the same vendor. Make sure all of them are similar variants from the same vendor. You will need 3 16x2 I2C LCD Modules. You will also need 3 separate power banks to make each node a standalone device. You will also need 3 separate power banks to make each node a standalone device.



The provisioner node will be powered by the PC as we will be using it to send messages to be displayed on the LCD. You will obviously need some jumper wires and 4 micro usb cables for the ESP32s. as we will be using it to send messages to be displayed on the LCD. You will obviously need some jumper wires and 4 micro usb cables for the ESP32s.

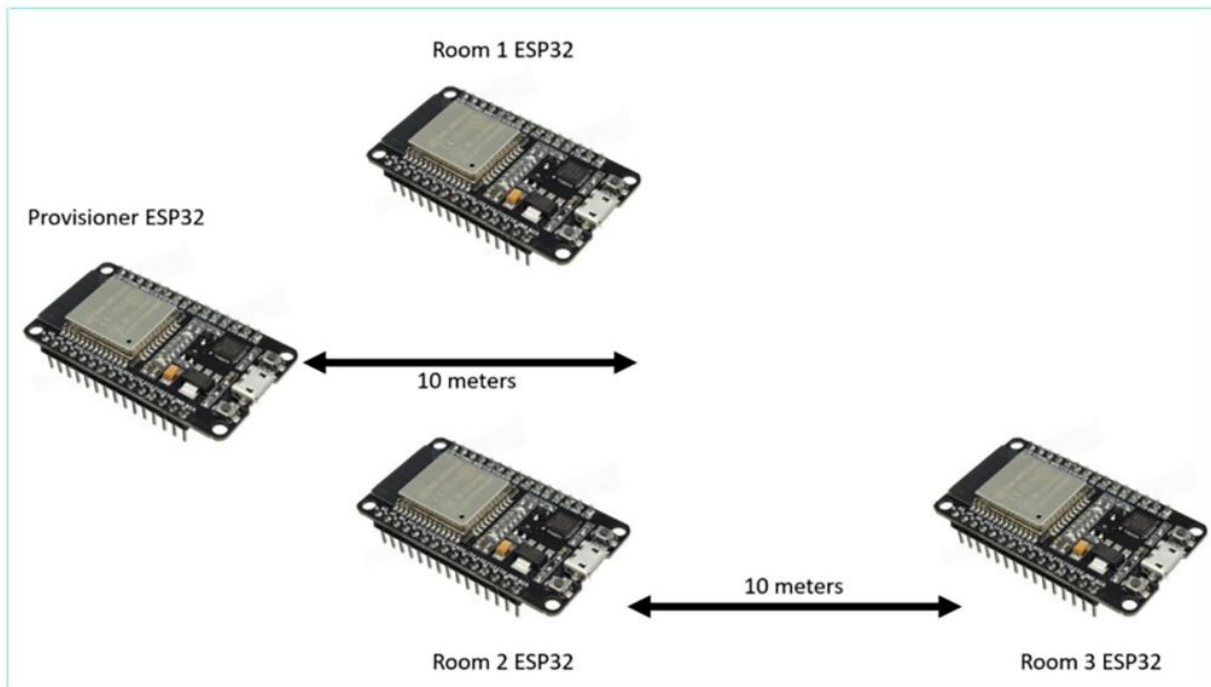


Now let's look at software requirements for this project. Apart from using the latest VS Code, Python and ESP-IDF, you will also need to download a library for the 16x2 I2C LCD, and PuTTY, to enter the messages needed to be shared.

You may also keep the nRF Mesh app that we used in the last project ready on your smartphone, to do any basic debugging of the mesh network.

IMPLEMENTATION

we will cover the following topics Implementation Testing and Explanation Further Improvements Lets just right away jump into the implementation. Make sure that you power your client ESP32s with power banks..



Now make sure that the 3rd ESP32 client node is placed in this test setup in such a way that it is not in the range of the provisioner ESP32, but is in the range of either the second or third client ESP32. Please note that the Provisioner node BLE has a range of around 10 meters. It should be noted that both the 1st and 2nd ESP32 client nodes should be inside the range of the provisioner node. This setup is implemented to test the mesh capabilities of the project. Now connect the provisioner ESP32 to your PC, and open a terminal and then reset the board and wait for the fast provisioning to be completed.

```
COM - PUTTY
I (0) cpu_start: Starting scheduler on APP CPU.
I (763) EXAMPLE: Initializing...
I (789) BTDM_INIT: BT controller compile version [a4c513b]
I (790) system_api: Base MAC address is not set
I (790) system_api: read default base MAC address from EFUSE
I (798) phy_init: phy version 4660,0162888,Dec 23 2020
I (2779) EXAMPLE: ESP_BLE_MESH_PROV_REGISTER_COMP_EVT, err_code: 0
I (2786) EXAMPLE: ESP_BLE_MESH_NODE_PROV_ENABLE_COMP_EVT, err_code: 0
=====REPL Opened=====

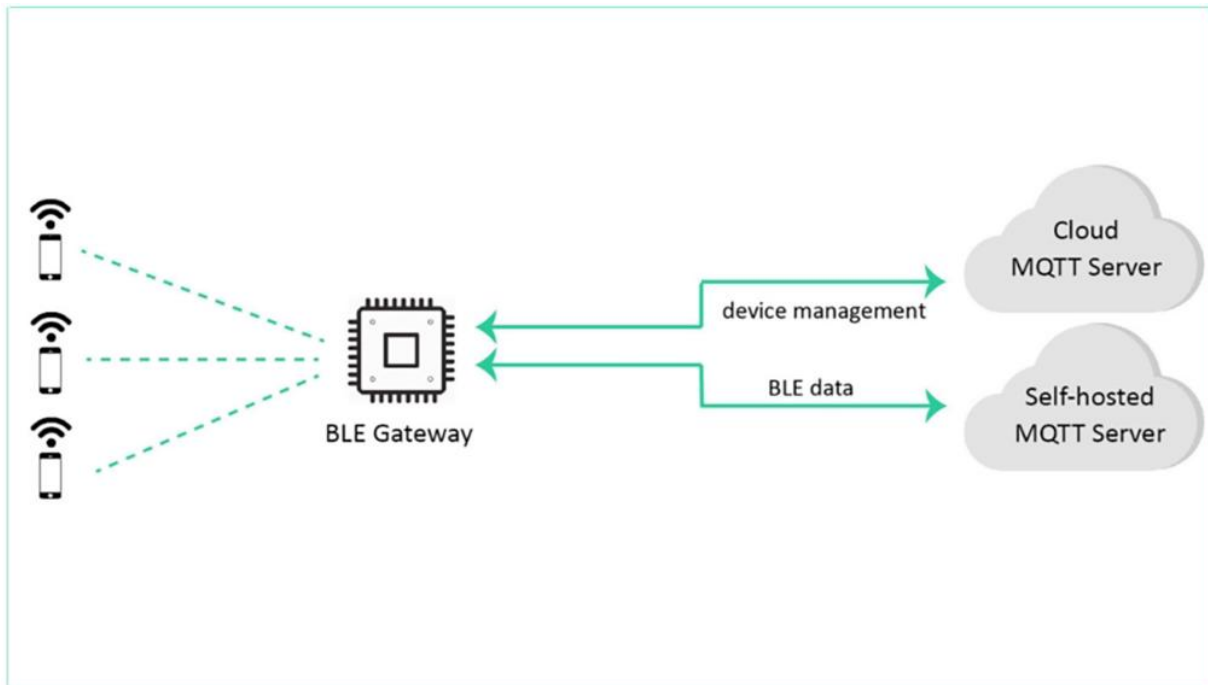
  Steps to Test Project
  |
  | 1. Enter the client device ID. Either 1,2 or 3
  | 2. Enter the message to be sent and displayed
  |
  |=====REPL Opened=====

esp> 3
esp> Room 3: Occupied
esp> █
```

Once it's completed the REPL will open in the terminal. Once the flashing is completed, the rappell console will open indeterminate now and select the Now in the REPL, select the third client node by entering 3 and then enter the message you want to be displayed on its LCD



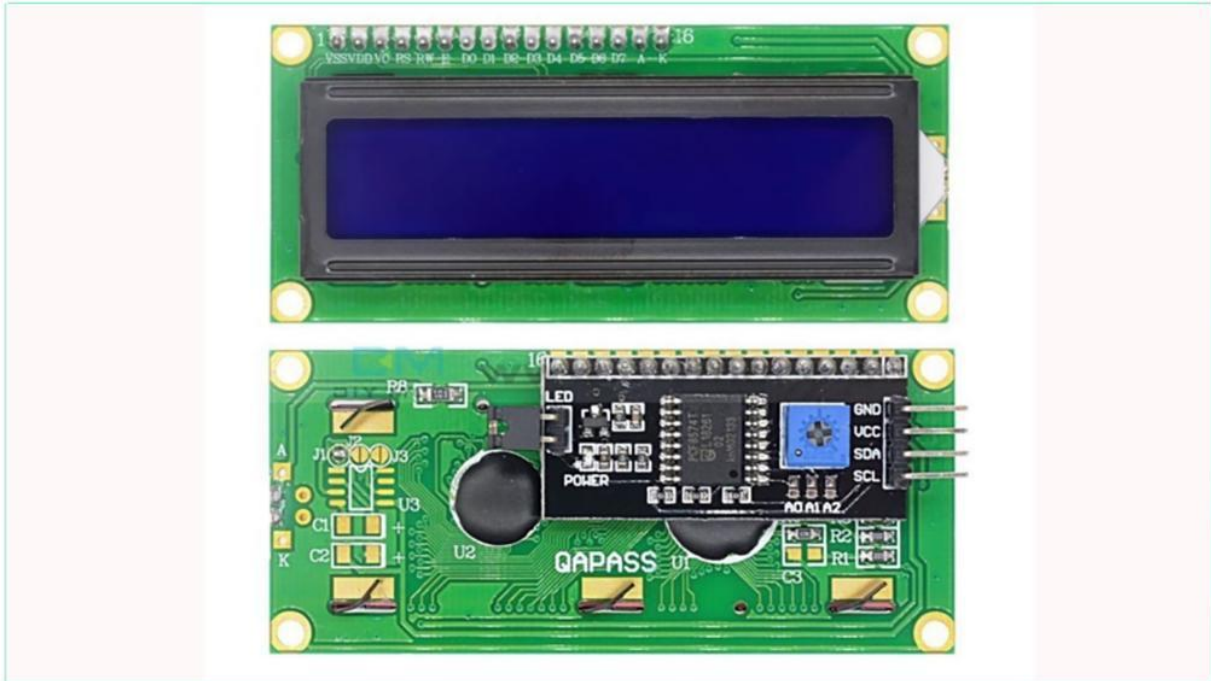
Now you can see that indeed the message is received and displayed on the LCD. Now you can see that indeed the message is received and displayed on the LCD. Now why don't you try sending messages to other nodes also to test whether all nodes are working as intended. We have successfully implemented the capstone project. Now I will give you some ideas to improve this project. This will be part of assignments for this section. This will be part of assignments for this section. We have earlier covered the concept of WiFi BLE MESH Coexistence and implemented this feature.



Integrating this feature with the provisional node in this project can allow us to convert the portion This will be part of assignments for this section. We have earlier covered the concept of WiFi BLE MESH Coexistence and implemented this feature. This will be part of assignments for this section. We have earlier covered the concept of WiFi BLE MESH Coexistence and implemented this feature. Now let's discuss another improvement that can be done to this project.

This opens up the possibility of issuing updates remotely from anywhere in the world. Now let's discuss another improvement that can be done to this

project.



The 16x2 I2C LCD that we used in this project consumes a considerable amount of power. But all those gains will be lost due to the 16 by two LCD power consumption instead of the 16 by two One of the main reasons to choose BLE implementation for such projects is due to the low power consumption. But all those gains will be lost due to the 16x2 LCD???'s power consumption. This is because for the specific application, we will not be updating the display regularly.



Instead of the 16x2 I2C LCD if we use E-ink based displays, the overall power budget will be reduced considerably. In this section, as an assignment of this section, why don't you research on E-ink Displays, and mention 3 reasons why E-ink Displays are best suited for this application. Submit in the assignments of this section.



As an assignment of this section , why don't you research on E-Ink Displays, and mention 3 reasons why E-Ink Displays are best suited for this application.

KNOWING ABOUT THE HARDWARE ESP32-CAM

why not some other USB camera models like VFP, which is very cheap, provided from passive systems, the volume of S3 modules itself. And what are the basic building blocks of our E.S.P digital camera? I mean, I will really jump into the schematic portion and will try to explain you what are the blocks available on the model and what is the main image sensor used here as an Collingridge sensor? I would call it sensor module. Because it combines with the sensor plus the lens. Associated lens for imaging and how the sensory interface tree is better to do. And what are the other miscellaneous interfaces available on it? So first of all, when you compare both the hospital to Cam and E.S.P, I would look similar.

ESP32-CAM vs ESP-EYE

ESP32-CAM	ESP-EYE
<ul style="list-style-type: none">▶ Board Not directly made by espressif, but hobby board companies like AI-Thinker▶ OV2640(2Mpx) image sensor connectivity▶ Onboard PSRAM▶ On Board SD Card Connector with support up to 4GB▶ Rst Button and Flash LED▶ I/O Pins Available for Interfacing Additional Hardware(Great Adv.)▶ Arduino Community Support▶ Lacks<ul style="list-style-type: none">▶ On Board USB to UART Conv.▶ Rst Button and boot buttons separately, and flash LEDs	<ul style="list-style-type: none">▶ Board featured by espressif for imaging aps▶ OV2640(2Mpx) image sensor connectivity▶ On Board PSRAM▶ No SD Card but Onboard Flash Memory▶ No I/O s available to Interface any hardware▶ No Arduino Support▶ Extras:<ul style="list-style-type: none">▶ On Board USB to UART Conv.▶ On board Mic▶ Rst Button and boot buttons separately, and flash LEDs

The only big difference I see is the eye opens if I go more into detail. This is the competition you can have. So the hospital to the camp is not made by likely by the volume like explicit William Anderson's original equipment

manufacturer. It's not made by Aspecific, which is the exclusive maker, but rather it is made by some hoverboard companies like Tinker. Currently in this course, we will consider Legazpi the hospital care model, which is made by Edinger, especially the sensor here used in both of the camera models is the same Oveta 642, which is the two megapixels. And both do have a strong. I will discuss about what is better in the coming late, and both do have, uh, a kind of specifications more the same. But the main difference, what I do find is the SD card, which is available to me is petrochem. And the I will be an extra 10 percent on the year spectrogram without I will because we can't really do anything on the with the hospital care model except that to image again, we have to find another hardware like our inner border or some other things to achieve the objectives of controlling the, uh, motors or the step of motor drivers or the ADC Motors. So it's a great idea to have our viewpoint on it and also both. I have the same kind of voting process, but the only disadvantage I find with the hospital, do you have to person who will be a reset button in order to enter into good. But whereas any S.P.I, you can just press that button in order to enter into Bootmen, you get through the camp. We have to show for you. I open. I will show the process in the coming straight.



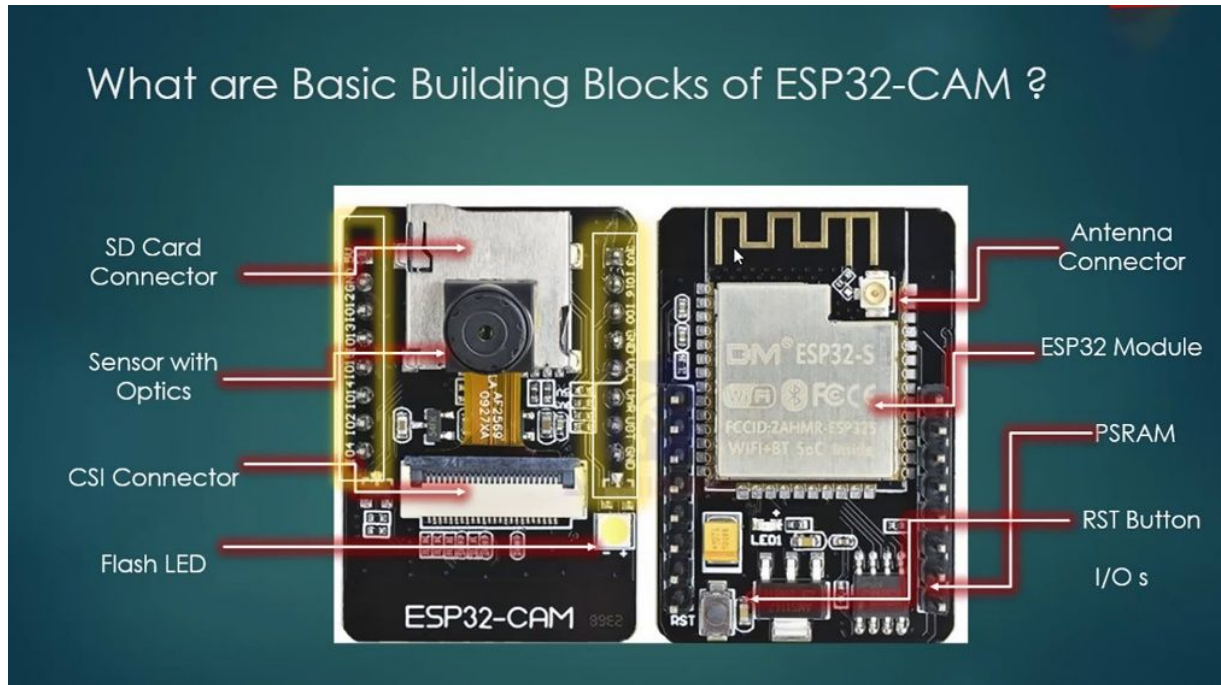
So apart from the FBI, we also have some other modules from companies like Digo, which is a familiar name in the Chinese border making companies. So they do have some more deals where they provided biosensor like this white collar center and on the serial USPI, the serial conversion with one one three zero six LCD display and some other options of Lintas.

So they come at a factor of increased cost, so as because instead of spending some more bucks for that particular models, it's better we go with the hospital to come and spend those remaining bucks for buying the model drivers. And these are role models and the models for driving over surveillance robot application.



So my final conclusion is, even though there are plenty of millions of years bitter to come in emerging market, this is the reason why I have chosen to use people to go from the article, which has both. I will be coming out and has a good camera support how the importance of support. So come into the building blocks of E.S.P digital camera. Basically you have a sensor which is Interface NCSA connected to the hospital to model, which is the source here and into this SWC model is connected to the grid and also Yeslam. We'll come to that what is based and why it is used and the respective power supply secretary is also present on board the capital required for the

sensor to be powered up. Uh, like the firewalled are taken from the header and it's been converted to gravity and further to three three three point three is converted to 2.8 in one point two required for the sensor. And also the speed model for this was a from that will also have the Wi-Fi or Bluetooth BCB antenna as like shown here.



So if you see Motorola available, this is the secret character I was speaking about and this is the sensor with optics. So that is a call center module and Skynet laboratory is a character. So earlier I was telling you about what is a sensor module, which is this we do this for the rover here, represents the equipment manufacturer, name, form division. And there is also a flashily for you to use for in case of hydrogen applications. Oh, it's not exactly an admission, but in night time, if you want to surveillance, do the surveillance, if there is any other allied soldiers to be switched on separately. Instead, we can switch on the flashlight and try to see the imaging. And you do have our antenna character on the back of the boat. This is different side of the boat. The issues are not exposed to the difference, whether they're exposed on the backside and the names of them, you can find it on the top itself. Like I was 16, I was zero ground. You see, all these are available. All the names are available only to observe and interact as well as the bottom, saying you have a reset button and you have

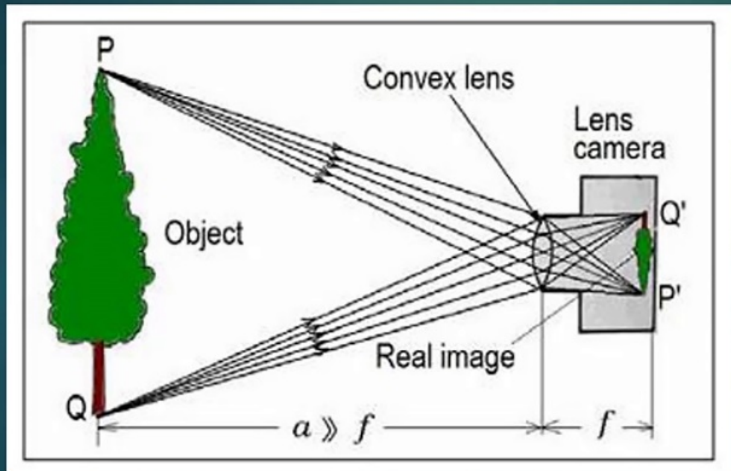
respective, um, power converters on the DVD and the bathroom. This is the same. And this is in Minnesota, and you have an external antenna connected here, you can see this beam, you can see this as it is shot to BCB and this is a shared back. Once if you remove this resistor and connected to these parts, the antenna will be functional. Uh, I mean, that were extraordinary, will be functional orders in this position, the PCV and misfortunate.

So what exactly is there? And so these people. So, yes, it is a very complex, as you see, a kind of beast in terms of operations was looking for the dollars you. So it's a very cheap as well. That is the reason why you are trying to get speed to try to get more religious freedom fighters back. If you try to see the same performance and a kind of resources available are like much costlier. But this particular one being very base, having the connectivity to Wi-Fi and also Bluetooth, and it has its own cryptographic acceleration unit, which can be very useful in terms of security applications. On the other side. It doesn't satisfy you in terms of embedded protocol interfaces like the sticker interface or the sound interface or the HWC interface, different interfaces.

UNDERSTANDING THE SCHEMATIC DESIGN OF ESP32

When coming to the cameras, sensors are the sensor model, which is used, it is the movie two six four zero, which is a model given by the volume of revision where the sensor along with the sensor.

How Images are Created



We also have the lens model. So here we can have multiple variants of lenses in the market, but for this cause, I prefer this particular lens model, which is having certain philosophy. This is the feature length where we will have much lens philosophy around like 140, 130 degrees. This will be like a normal what we are using currently will have a regular philosophy around 40, 40, 50 degrees. As for our mobile phones, so how are images created, for example? This is the tree which we want to see in the image. So basically, we have a lens which will help us to concentrate the reflections from the trees because, you know, whatever we see are the reflections of the sunlight which are falling on the object. And we are able to see. As because there is elimination of light on an object and that reflection is falling on the right, every eye, the human eye will have a lens of its own which tries to focus the image on the retina. Here you can imagine the image inside of the retina, which further sends the signals to the brain to form an image. So similarly in our senses. Also, we do have a lens which focuses images, I mean the reflections onto the center and creates the image digitally.

So what is the sense that he two years or over two, six four zero, so make sure that you also use the same sensor? Succumbing to the of specifications, you don't really need to be very clear about his incentive specification, but the reason why I am trying to focus this particular aspect here is you should

understand what is actually provided by the sensor, what is actually done by the hospital to associate. So this particular center used here works on a call to different point walls and has maximum pixels of sixteen hundred twelve hundred pixels. It's just an imagery. These many pixels will be horizontal and vertical, say. So by default, the center gives you different formats like the hajib and Wikipedia formats. These are different emerging forwards and coming later, we'll see why. What are these formats and why they are being used? So therefore, every answer there is also called the frame rate, 50 frames per second. So the full resolution, like the sixteen hundred one hundred is the U.S. resolution. It can only give you a throughput of 15 frames per second. That means 15 images. Whereas if you reduce the resolution to even to visually 640 480, you can go to 60 FPS. This is what support president. And it does has a very good dynamic range of treatment, which is sufficient for us later on in the sessions. We will try to tweak the camera parameters and adjust it for our purposes of imaging.

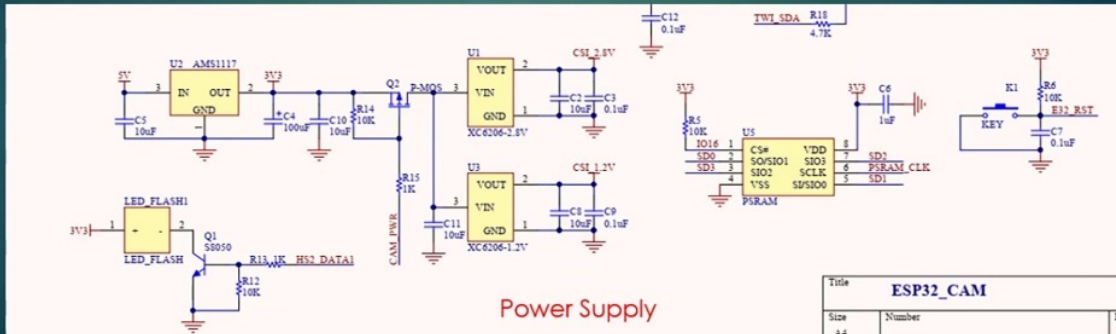
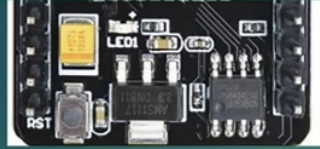
So how many is it that it is able to handle the image processing of that image? Actually, if you see the Center for Windows XP or Zero Datasheet Ship, which is attached in the section for you in this lecture session itself, I have provided as an attachment to this particular datasheet, if you see a cinder block diagram of this sensor internal. There is an imagery, which is the sixteen hundred, one hundred, so you have columns and rows which has been split, the imagery splintered into columns and rows and they are being sampled. At a particular frame that you have selected and those pixels are amplified and given to analog analog to digital converter, which further converts it into a digital format and gives it to the image. Processing block. Which predominantly consists of converting the pixels into a representable formats, so we will discuss in the coming leches how we are actually converting the images into a representation format in a digital format, a completely made how it is created. Oh, we'll see. Pretty basic, certainly, uh, Latron will jump into programming it. Bertolino But understanding this interests will be very critical and useful to you to work in an image processing applications. And this will be a very useful data for you in on any different projects that you do with. So this particular DP block here will do the image creation and bad pixel versions, not all pixels which are very present in the imagery, are perfectly all right. There will be some

manufacturing defects which will make them a bad pixel, which will give you a wrong illumination level. So those will be corrected by the processing bloggers and in the center of disability center, which is tightly attached to the processing, they're processing electronics, not actually the complete image processing, but at least up to the sensor level. Deformity corrections are available in the sensor modulated. So this will reduce the burden on the hospital to do that, it will be able to do and take the leverage of the processing power of hospitals to do, and we are not burdening it by trying to do most of the operations minimalistic. And I'm taking on the center of public attention. So even the GP competition, which is required. In order to. The raw data into a very less articulate, that compression would be a very great factor. We will also see a compression in our communities, how we will help you in storing the data. OK. So I will do a brief of why this competition is required. The GOP competition will be a rough calculation, uncensored, full resolution at 16 a little bit. All right. That means we have that many pixels, each one after the other in terms of rows and columns. And it is a resolution can be a bit of time, but we can program it. And we do have three colors, which is red, green and blue are the if you see the format. You can support RGV, fight, fight, fight, fight, fight, fight, which means each bid, each candidate has been given fireboats. In different formats, like where you each space is given no space to remember it's been for you will see kind of space is also in our communities. So I suppose if you consider full three made three full resolution, three being 15 frames per second, which is what their leadership qualities. So, Rodica, you may need to almost 16 and bet 100 with a being amount of data from the Alysse. And you have tricolours for 15 frames per second. You will recover 82 and be 82 megabytes of data for one second. Can you just imagine that we'll be able to handle this much amount of data needed? We have that much avam available with the hospital and that is why we use the compression technique concept or the other encoding techniques like Wisecup to bring down the data. By a large extent approximately, you would convert this 82 M.B into eight point twenty percent, which is what makes it transferable or transmittable as a value for benefit, because 80 be being transmitted over Wi-Fi and being collected from some access point and being displayed. It's really choking the network so instant. If you do a compression image compression exhibit will be compressing the data by one tenth. And you

will see plenty of that, so if you revisit the speed that you do, then you'll clearly understand again after being told these facts, wages are ridiculous. This so it is quite simplistic and there is not much image processing happening on the digital except the image processing algorithms. Then the images, signal processing, for example, when I say mechanical processing, all these things which take huge computation, power like the production or the bad interactions or the non-uniform decryption where different pixels behave differently for different eliminations. All this had been taken care of, and that is we were able to keep the architecture very simple. All right.

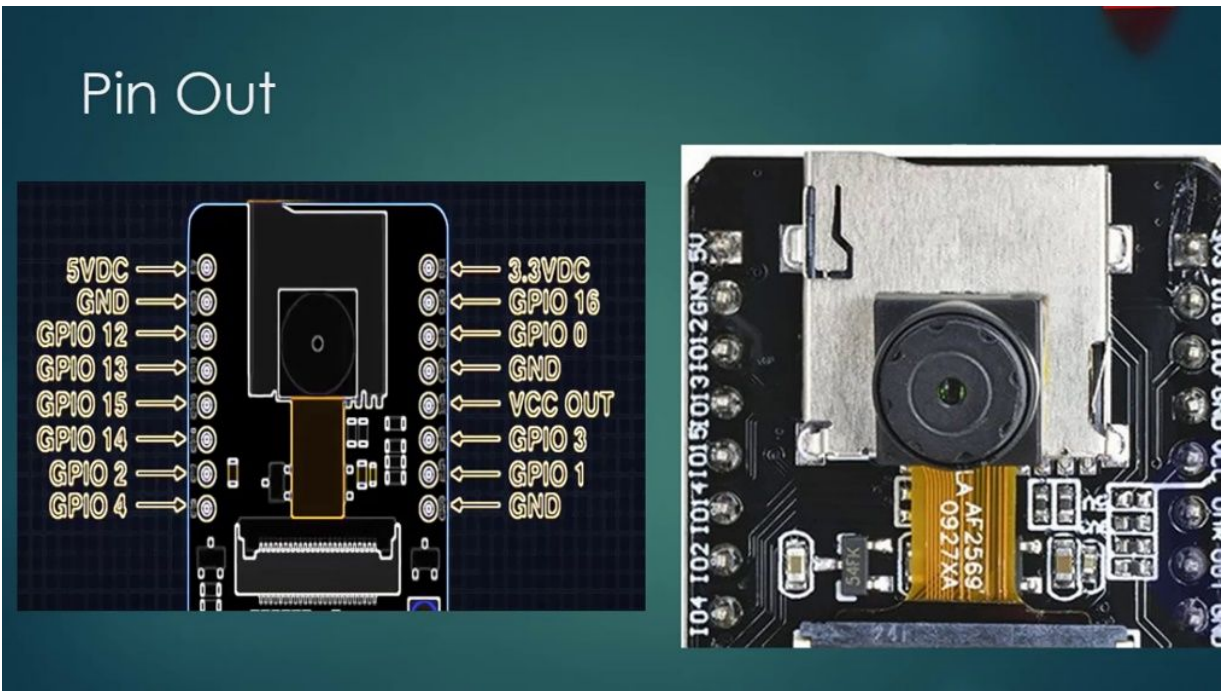
So we will try to get into this schematic. So this is the classic schematic. So you have cameras that data connections and block and a few horizontal and vertical sync signals coming from the center, really. So the communication of U.S., how does it happen? These are all been taken at a level taken care of by the ordinary. So you are in a safe hands to communicate with the center. By making use of the ordinary British. And the liabilities provided by the specific and these are the characters which are exposed. Which are mainly to give you access of the occupants later on in this course, we'll be using the legal means for controlling the murders of sergeant. And result is the character here, which is interfaced, some of these lines are shared, is they willing to share? I see. I just wanted to say that I want to present to you and also present here.

Snapshots of Schematic



So these are shared islands depending upon the usage, and we can decide. Also, another thing, we do have an injury flash and also if I want to be with taken, but we do this and we have after the perimeter is converted, it is again converted down to two point guard and one point for the TSA over this, which will be shared to the center of TSA entropies. So these are the there is one more Doninger. You can see these these are the two bags which would provide power by taking the power from the back and then you would be distributed to the sensor. I would describe additional to this. We have also PSM. So from previous slides we had been waiting for to know what is being used.

RAM is nothing but as good as a slow start again, which tries to behave as a static. In fact, not it's not a static. So a static line is a technology where it will help you for uh, in case of roughly to to say very, very briefly a ram random access memory in your computer is very required. Very much required because you need a faster memory when compared to other people. So these are all dynamics which will help you in image formation or image compression or image conversion or image processing operations. Image processing operations inherently require more than one person. So this PFM is interfaced externally to help the algorithms which are on to the hospital



So if you see the up, you can clearly see that this is if I want any ground. And three with three. So if you power the firewalls, you can actually take the territory. So if you want to power the system with liberty, you can even do that. But especially in this course, I prefer to power the module with firewalls. Because it will provide you with sufficient current, which is required for any kind of processing applications. Apart from that, you want have the lupin's. I will fall. I will do. I would run 13 miles. Importantly, a battery if you have you zero to use arm. These are the basically what transmitter and weapons, which are very much required for us to program the U.S.A.. Harman flashing the spitted.

HARDWARE REQUIRED FOR THE COURSE

Now we will see what other things required for us to complete these projects and let the very first primary thing we use to determine anything.

Things Needed

- ▶ 1. ESP32-CAM – AI Thinker Module



4.5 \$ - 10\$
Depends on Country
Custom Charges ,
Shipping Charges, etc

The image shows two components: an ESP32-CAM module on the left and a camera module on the right. The ESP32-CAM module is a black PCB with a camera lens and various components. The camera module is a smaller black PCB with a camera lens and a gold-colored ribbon cable.

Which comes along with this insurmountable. So that would cost you around four point five to ten dollars. It depends on the country you're living in and what are the Customs Service and shipping to this? Some miscellaneous charges. So it depends on that. But roughly, you'll get it around some six dollars. Five dollars? I hope so.

Things Needed

▶ 2. USB – UART TTL Converter Cable



1.5 \$ - 4\$
Depends on Country
Custom Charges ,
Shipping Charges, etc

And the next important thing is we will see what the deal would discover that I would prefer better if it is out of. Like you plug in model where you can have bigger waves coming over, you can easily insert into the, you know, the camera model and show you how to do it. And then Xitian.

Things Needed

▶ 3. L293D Module



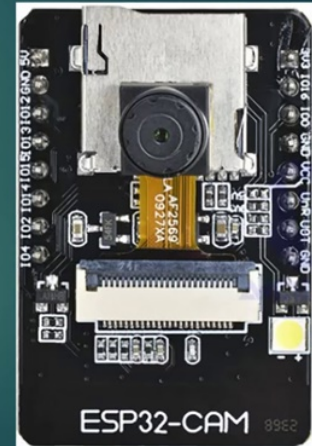
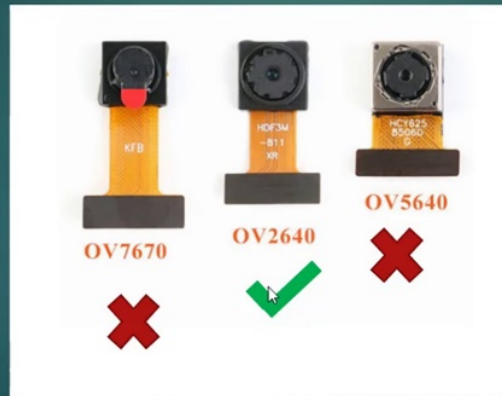
1.5 \$ - 4\$
Depends on Country
Custom Charges ,
Shipping Charges, etc

And importantly, the Altuve and the. This will drive our voters of the Woodcourt. Due to his health care. So this would cost you around seven point ninety four dollars and the US retail convert is going to be costing us more or less the same. So here the beauty is like you have the riding pants. Exposed don't on the Syrian side of the BCB, and these are the more the connection between. And basically, to the contrary, is the kind of driver which will away and it's currently motors taking it from the alternate current source. And the control is happening from the U.S. that took.

And one more important thing, you requited. Two pieces of genetic. My concern was these are two microsurgions. A very narrow minded bracket, so this would cost you around some four point five to ten dollars, including the motus sometimes if you see the price online, if it is, let's understand that just the brackets or the service included. So make sure you're comparing the service also.

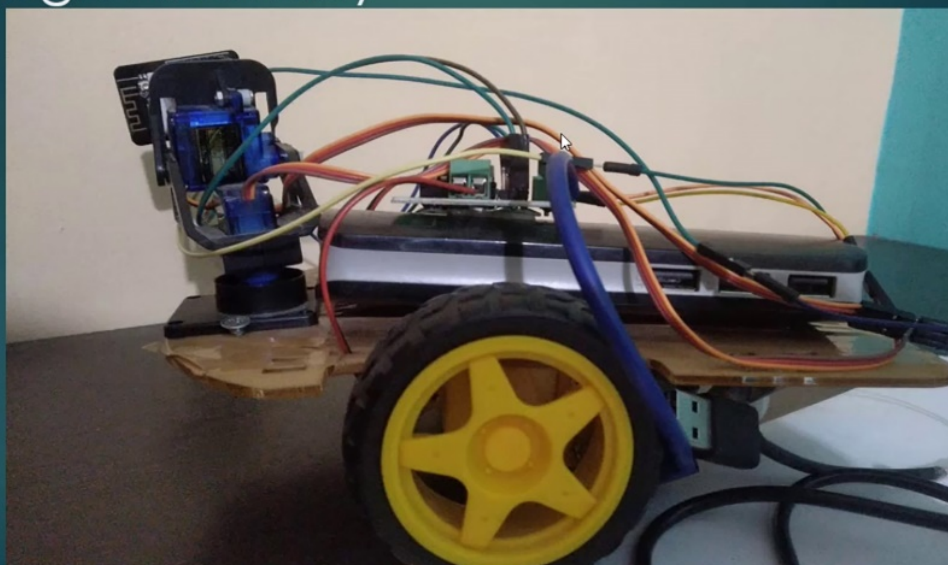
And the one more important thing is to watch, as are the borders of the carpet, which consists of to listen to these models and one Castroville schizophrania. So once it is assembled that we will do this once it is supposed to be something like this, actually if I ever played a big role so, you know, somebody will also be discussed in these sessions. So you are raising money until you understand, not usually most of the models, companies that are simply using the most politically self-contained, you can assemble it with the tools which are given in say that with a small screwdriver will be sufficient. And three, he was being put by one each of at least one parent, but are basically they come in from form factor of three, which will be sufficient for us. So this would cost some 10 to 15 dollars. And just like most of you will be having home. And this will cause you to make good relatives. And some Zimbabwean's. And I say that tape, which will help you in assembling of the 32 camera to the front assembly. So this is the part that you have to assemble these people who can do this, but how to do it? This is where the debris will come into the news to us. This I have followed my own hack so you can try to figure out how to loot sorry for the sound system.

Don't Buy Wrong Hardware, If sensor became faulty



And there is also an auction also alerted I would suggest you to take, because this will help you in learning it's going to be useful to six four only for a camera in case if you damage these are the contents of which comes with the hospital to Kathmandu. You can just, by the way, be this for little mountain and replace it.

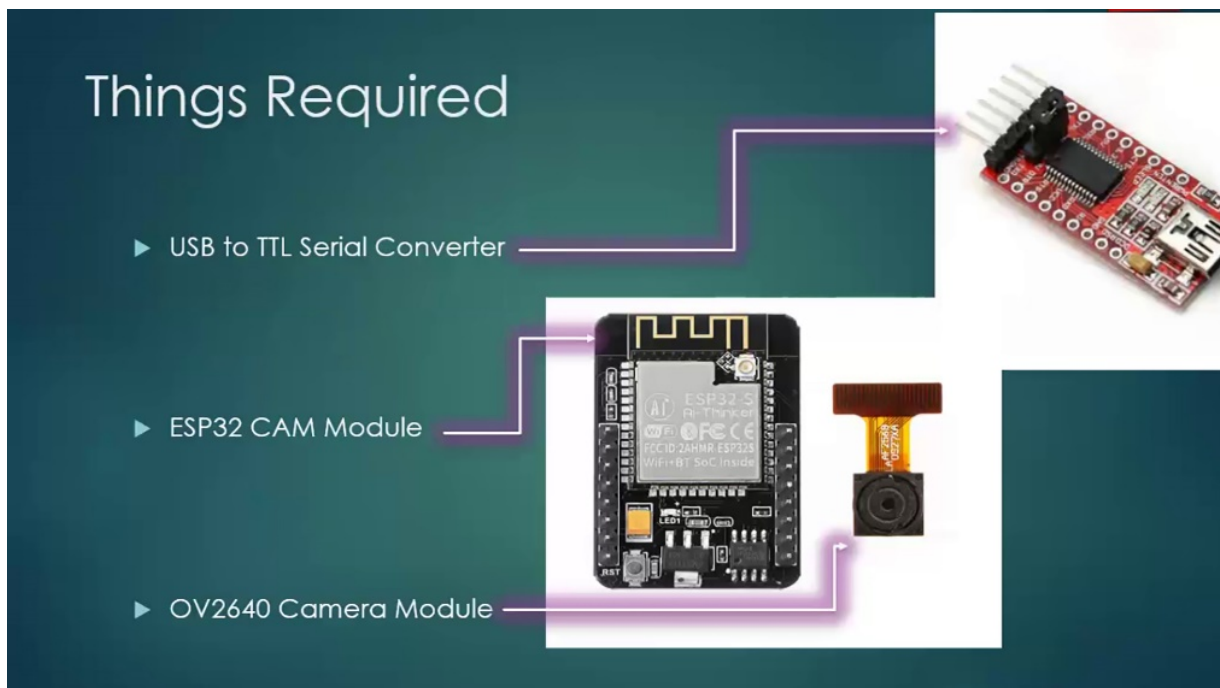
Rough Assembly



So if you see the final assembly with all the conference, what we have, it looks something like this, sorry for it being a little bit. Clumsy. This is the phantom assembly. I have put on the. Jesse, this is tragic. It's called Jesse because this is the main body which is actually making the cut and the camera is being molded here with a double. I got to show you, how did they do that? And this is the bank and I have was from the accidentally, uh, input parcel's. I mean, from here, I have good deals being able to go out and I have just plugged it into the end zone and really input and one other I have used for. So was. OK. These are things which are must ensure equal footing. Hope you gather them all this sense that decisions are.

SETTING UP THE HARDWARE

we'll see how to set the harder it is federal computer programming or uploading the code or another flashing the U.S. can take it from the remaining days. We will be seeing what are the things required for operating the code and how to configure to upload a code. And how to program with the that?

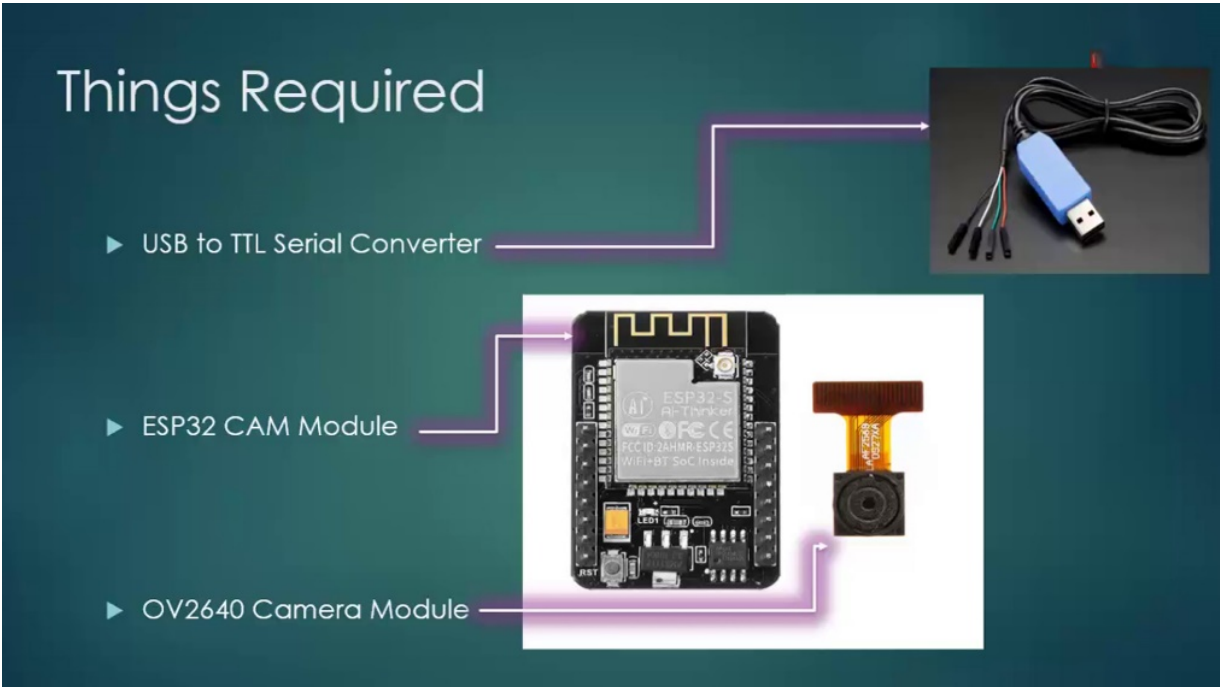


So things which are required is a bit of detail, can you speak to the camera and do something more on it? So basically I also show you how to connect

the sensor. Sometimes the sensor module comes separately from the camera mounted to the shipping issue.

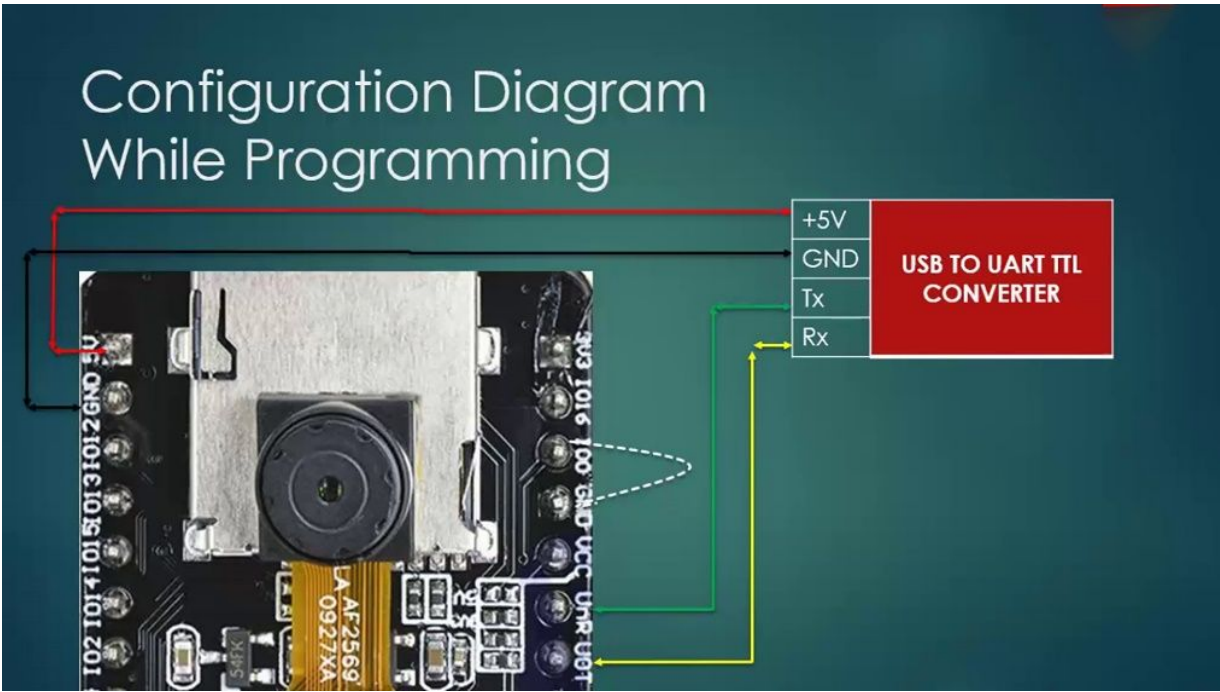
Things Required

- ▶ USB to TTL Serial Converter
- ▶ ESP32 CAM Module
- ▶ OV2640 Camera Module



So in my case, I use a lot of paper in the speed of detail and it's already inserted, it is handy.

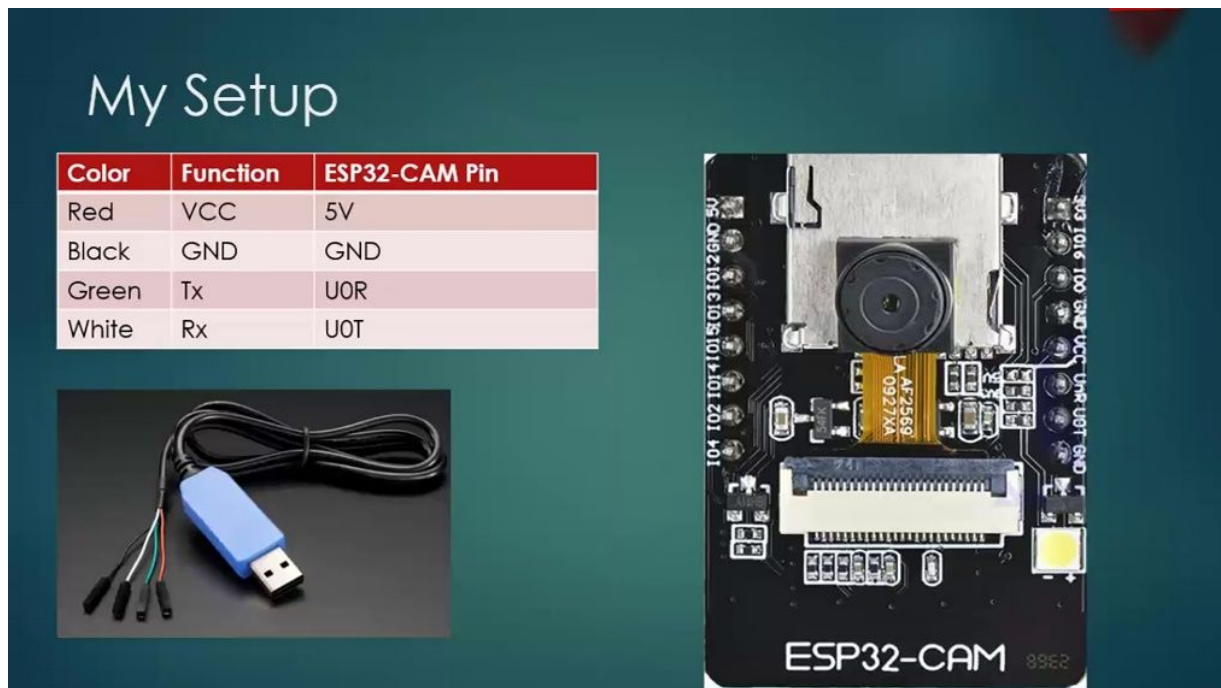
Configuration Diagram While Programming



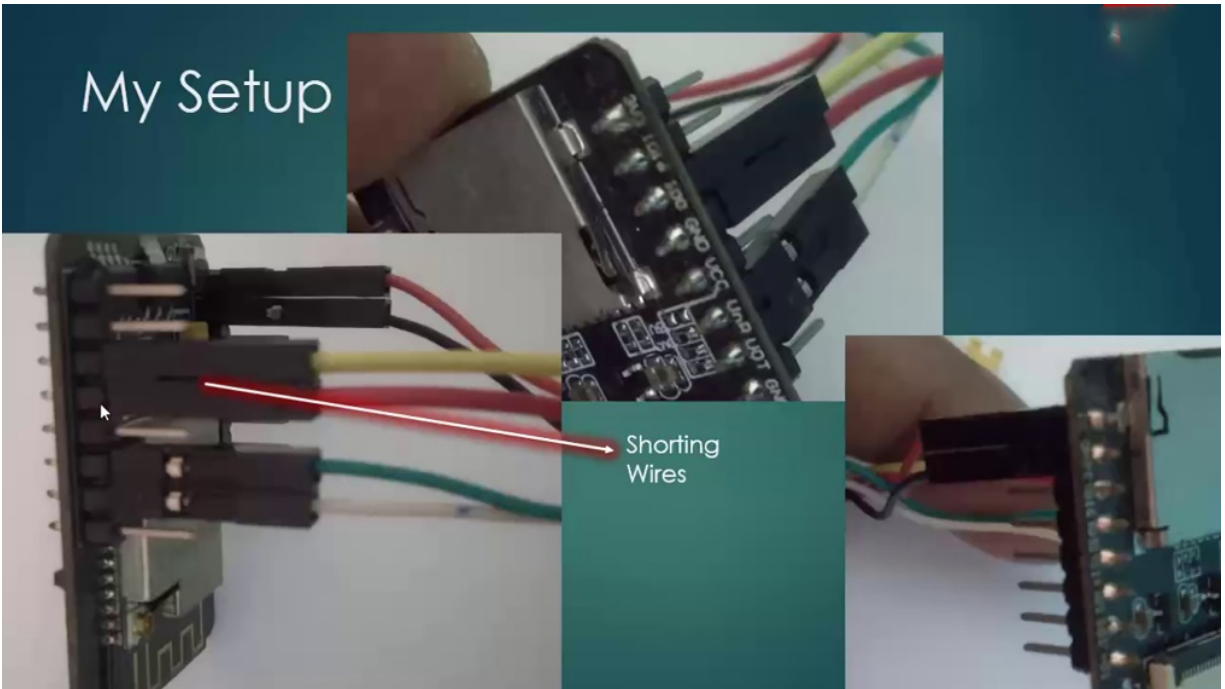
+5V	USB TO UART TTL CONVERTER
GND	
Tx	
Rx	

Labels on the ESP32 CAM module: 102 101 410150131012GND 5V, AF2569, LA 0927XA, 3V3 101 6 100 GND VCC UART USB

So when you're the configuration diagram, when your programming is you have to speak to, you are detailed computer models, if I will, just phonetically, if I born and grown to be, grown to be and then voice. It is already provided by default and the thing is being connected to the receiver, it's being transported. And this is very important, this particular the Iowa zero being connected to ground will push the digital camera into Google. Which makes it to. Take. So while an operation won't start, but including some quick check for I was illustrator's when I was illustrator's being zero, which means that it goes into programming mode aren't normal, Budiman. So in the process, if you go back to the executing the main program or do so in operation, we just remove it so that it takes a program like that.



So in my second, I have the red color, white, green and white, white and black. So these were white and green, or sometimes the white is replaced by a little sometimes. So basically, the functions you have to identify yourself, you can just have the dictionary and see what the programming is happening on relatively easy, technically, pretty sure that what is your reaction? And most of the things the one who is selling this to you, they will tell you how that happens, functionalities so you can likewise connected.



So before going to door, you just show you how well my setup is. So this is all Internet. These are the shocking words, what I have said. So what are you going to do? Some people try to take a jumper and attach it because it is going to always putting a damper on it at this time. So what I do is I put Frehley hanging batteries. We talk to them and I just kind of them by. Charged them and arrested is important. There is a reason to press the reset button. I'm going to be programming more. Are you still waiting for Gwendolyn to call them so to step up the charges and press the reset button and then only need to go into the program? These are the oddest exploits that have connected to the market. And this is Mrs. Graham. OK, this is the little. This is in front of us, so these exotic antiques. This is where I in the new holiday. I just got me a speed that you do. Come on, I'll bring you back from one of the wounded and we just it if you're actually on it, there are two things that you can find. One is the hospital took a little water and sent it to. So. I was telling you earlier, some people will try to use a short deal, but I refer to Chardy I was hearing when I was zero, you can observe here and between. Know the charges, some people will try to charge them, but this is also an effort to it, but I don't personally prefer it. Rather, I would prefer to take chattiness. Like this. What happened is these are two males. Disconnected to Milward and. Nick and the Greens are back, so these are two separate ways, so this is the females. I thought it was going to be

female side, too. I was, you know, and groan, OK, now, when we are trying to program what you can simply do as you can just catch these two women, which is essentially taunting and press the reset button. This is the reset button.

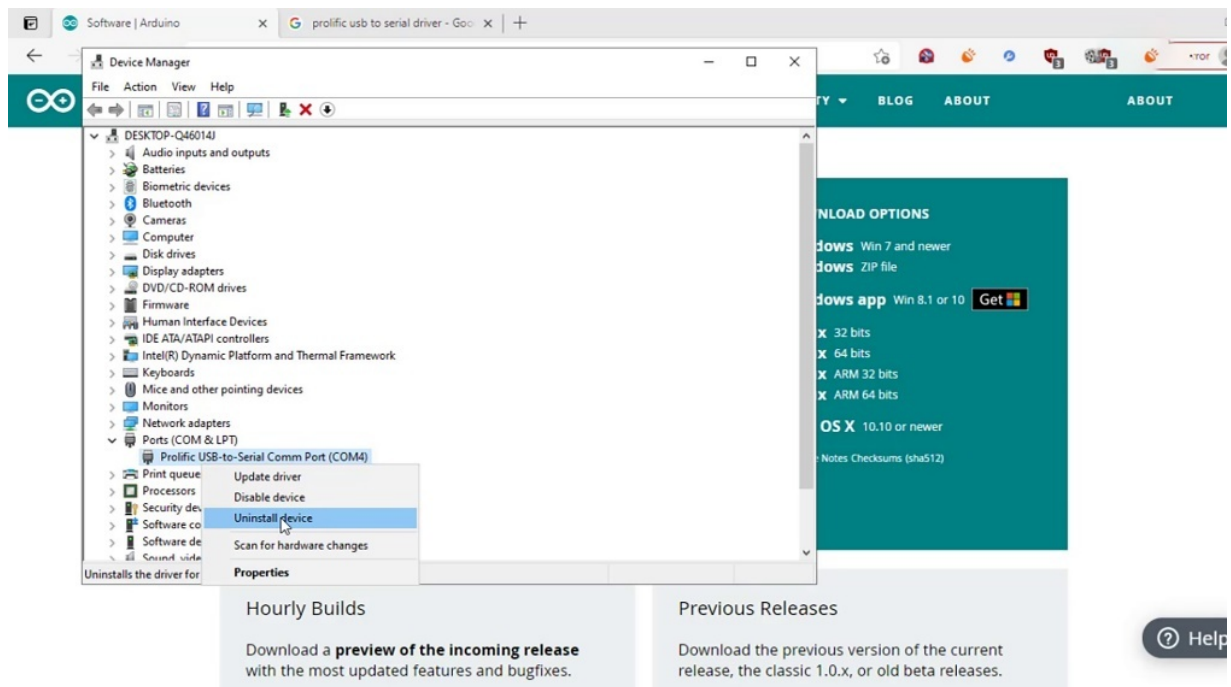


You can just press the reset button like this, which will make it to enter into programming mode. So is so then it just remove these guys and let us insert this sense of like the protecting cover would be like that only hurts most of the dust particles will enter into the lens, which will decrease the image quality. So if you see the model very closely. There is this character you have to just put your nail or something and just drag this backward, you are just flipping it up and repeated. Once to disclose it is like this, if you have to open it, just flip it back very, very gently. So now you take this center and very carefully inserted tropica. Into the kind of trouble like this, just generally don't push it too much until. Once you have boosted just the discount window, which makes it complete, most of this century will be getting a double take on the central bank so that you can stick these and stick to the model here. So when you're inserting it, you can just remove this uncertainty and you can just take it so that it will stay so that your energy level to be stable when you even if you're moving. OK, so let me

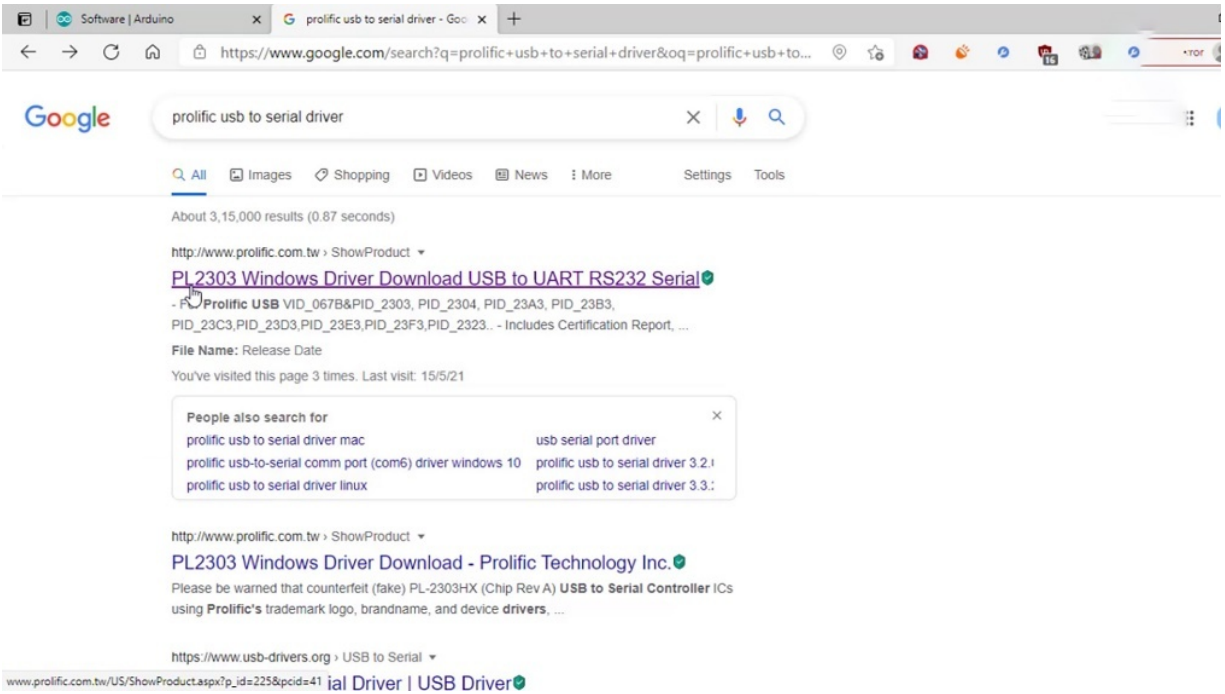
take the this ingrown connected to the wires here. This is the red and the green is always top of the shopping. Will this do something worse? I will also tell you all the things you have to do to set up the software like the film three zero three drivers for the U.S. and Canada and also the ordinance at. Once we do it.

FLASH YOUR FIRST APPLICATION ON TO ESP32

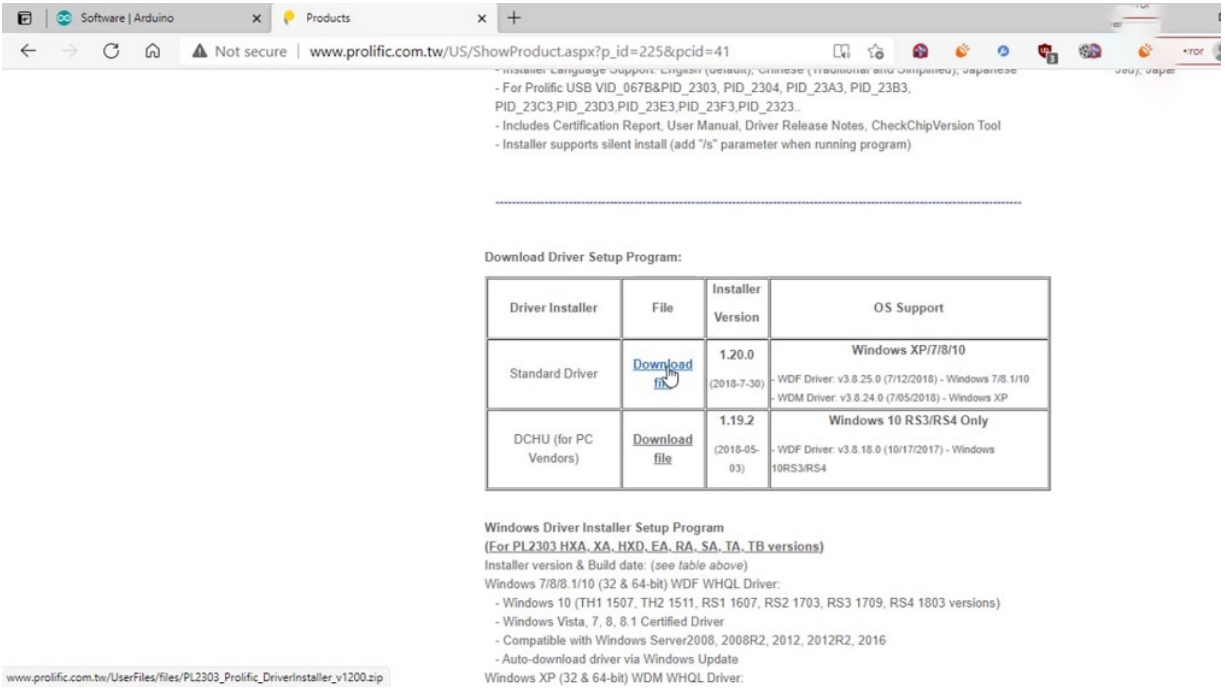
the download section of the software that I identified in the latest version, which is is stable, so you can just click on Windows and download, I only have one senator downloading. One more important thing to tell you is and you're using this, you can work and you just plug it in if you just plug it in.



So I have called this control for detector the to civil converter, which is having a prolific part of it. So I have to install that. I was corresponding to that sometimes, for example, if I tried to uninstall Littlewoods. And they show you how it looks like. See, as doesn't value the control, because you will see some kind of problem here.

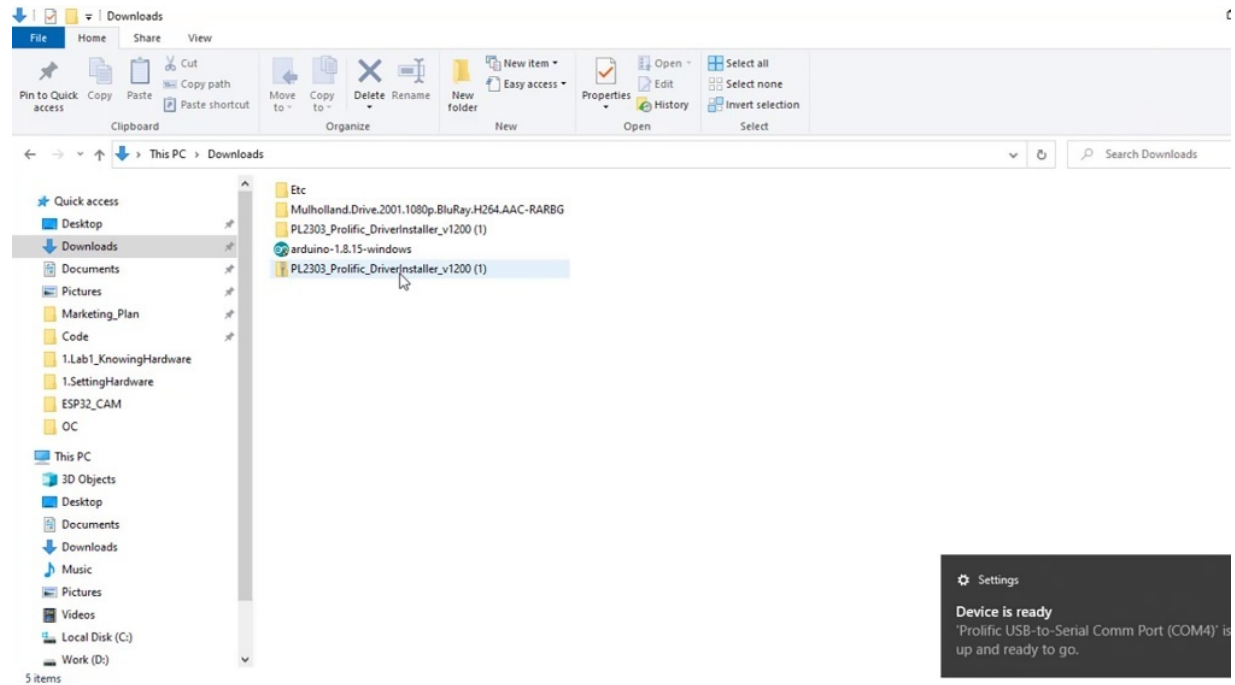


So you can download qualifications online, it's bill two three zero three under.

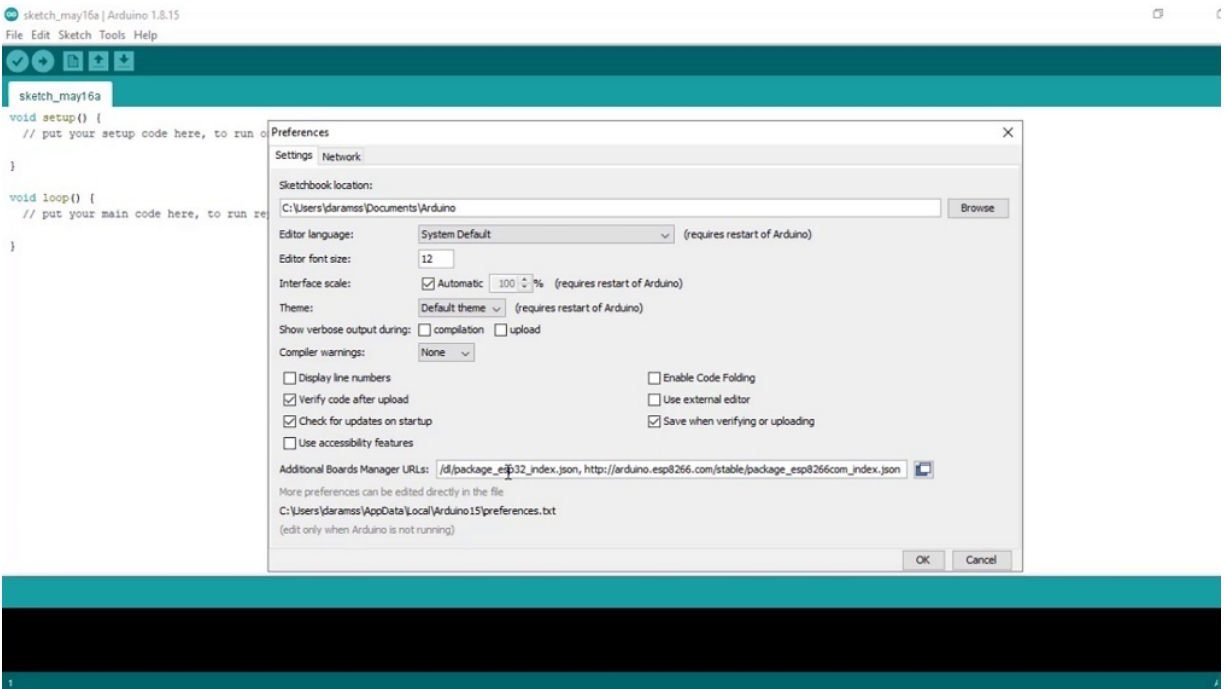


So the standard would have to be persuaded here, you can just download that somehow downloaded both are. Only know and to three times. S.. As

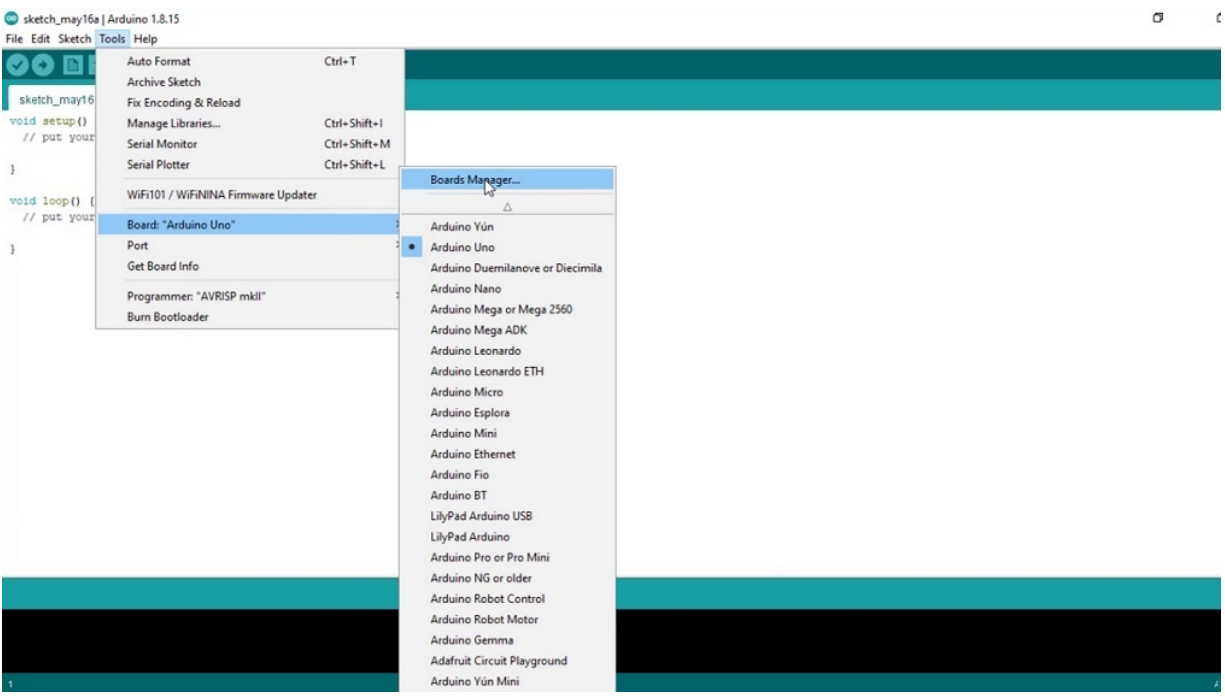
because I have already installed and was able to autodidactic on, let's suppose this is not happening.



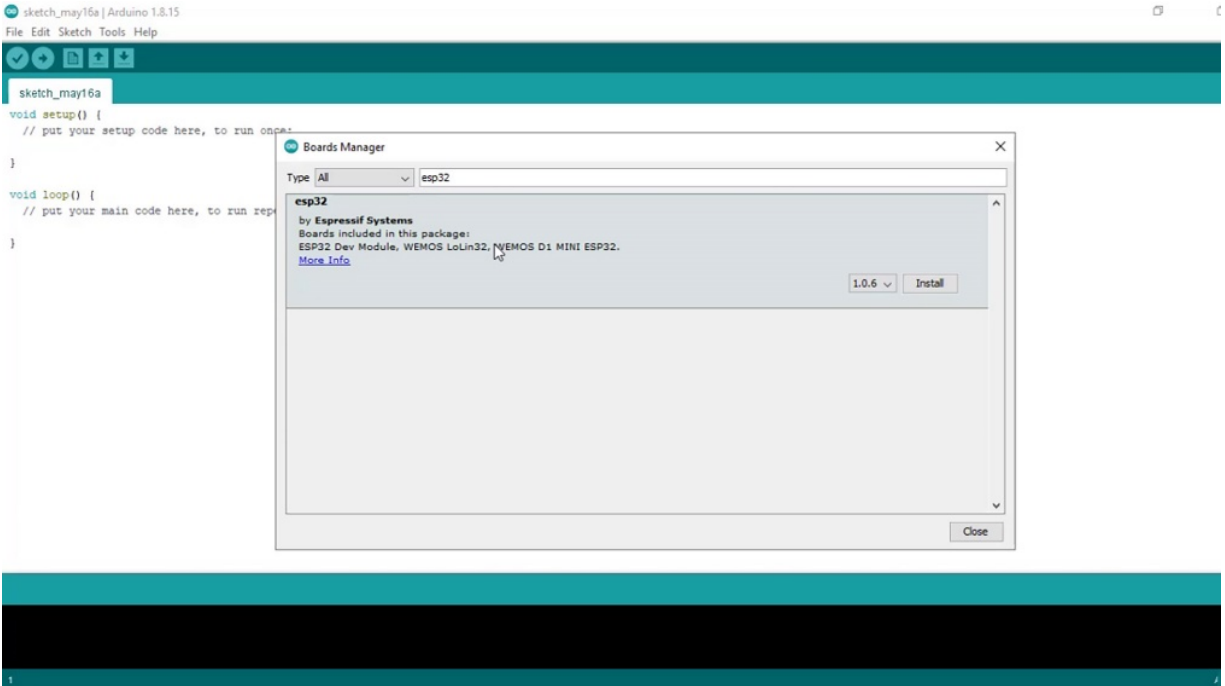
You have to install the bill two, three, two, three, zero to try to just stop it. I'm just reinstalling it doesn't make any difference, and we shall install the ordinance of. This is Ardino. Agree to be on. So usually the default folder program is. The installation is completed, its antics are the truck that will be great, London's this government. If you have already installed audio, so you will get the previous preferences, all things done. If you have done anything for a because it takes on you speak like you do other countries, so this mighty heart of it is not detected. This is a problem that can get.



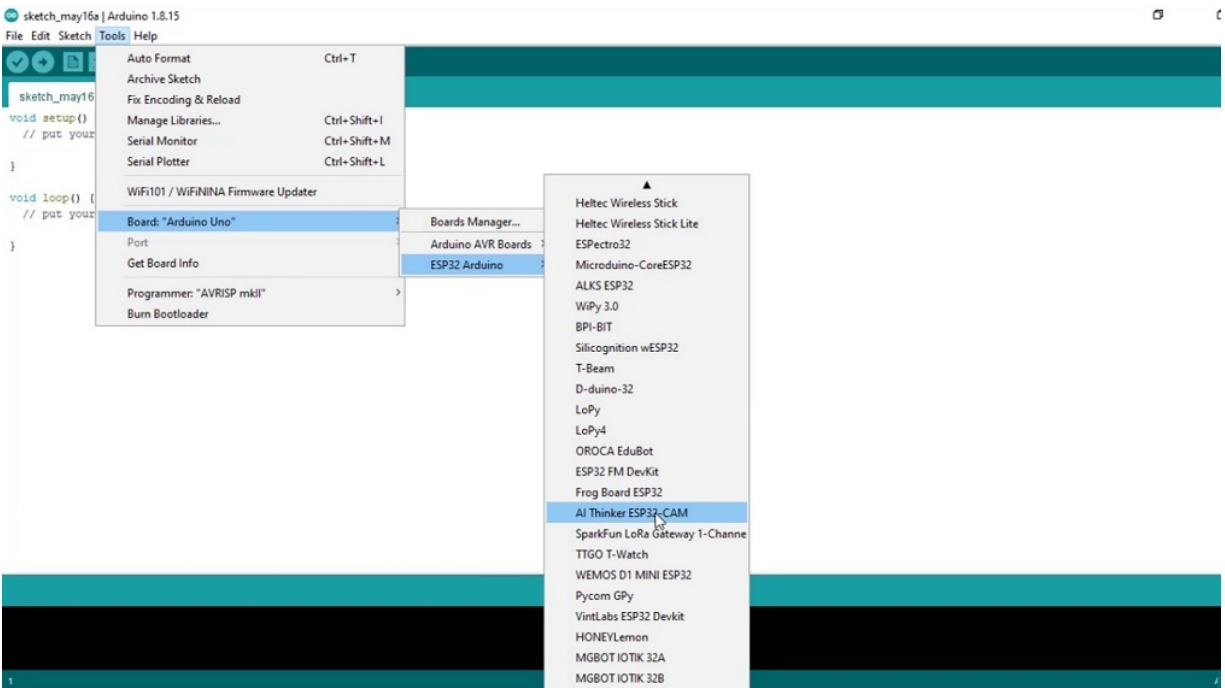
So that was four years ago, so you go to conferences. So given that. There are two important worlds we talk about which are becoming. Once you have done this, you cannot and Whirlpool's.



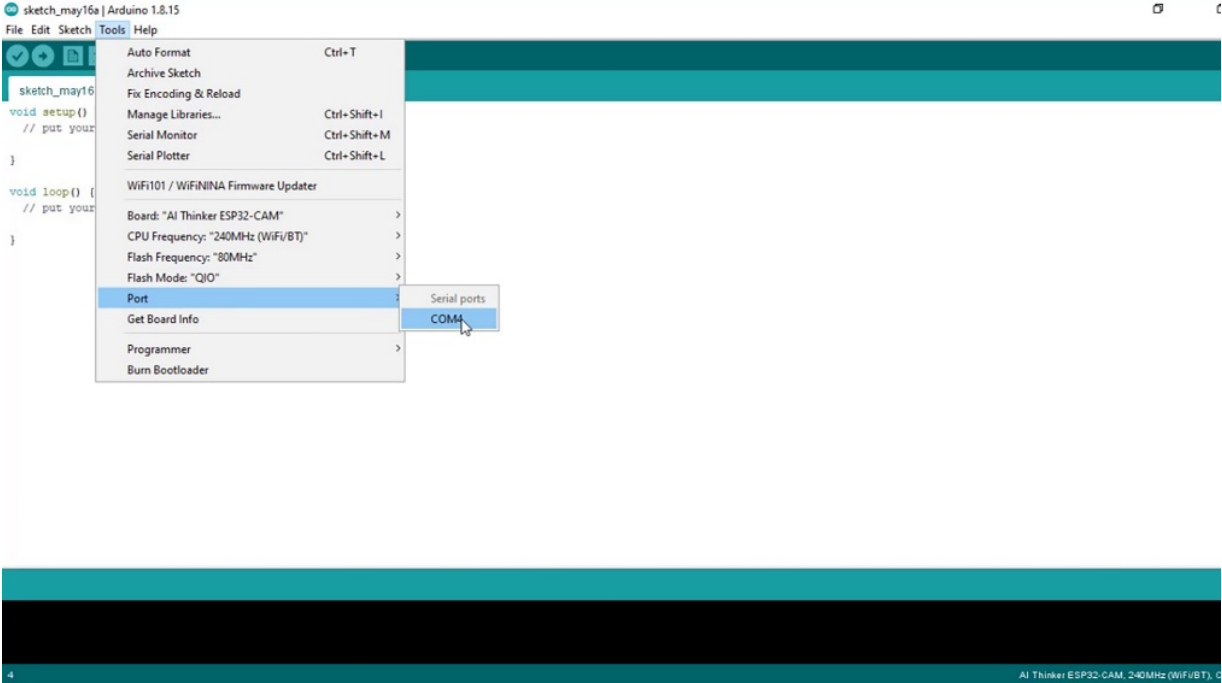
Go for bone management, because we have to find a lot more money to just go E.S.P. Patrick.



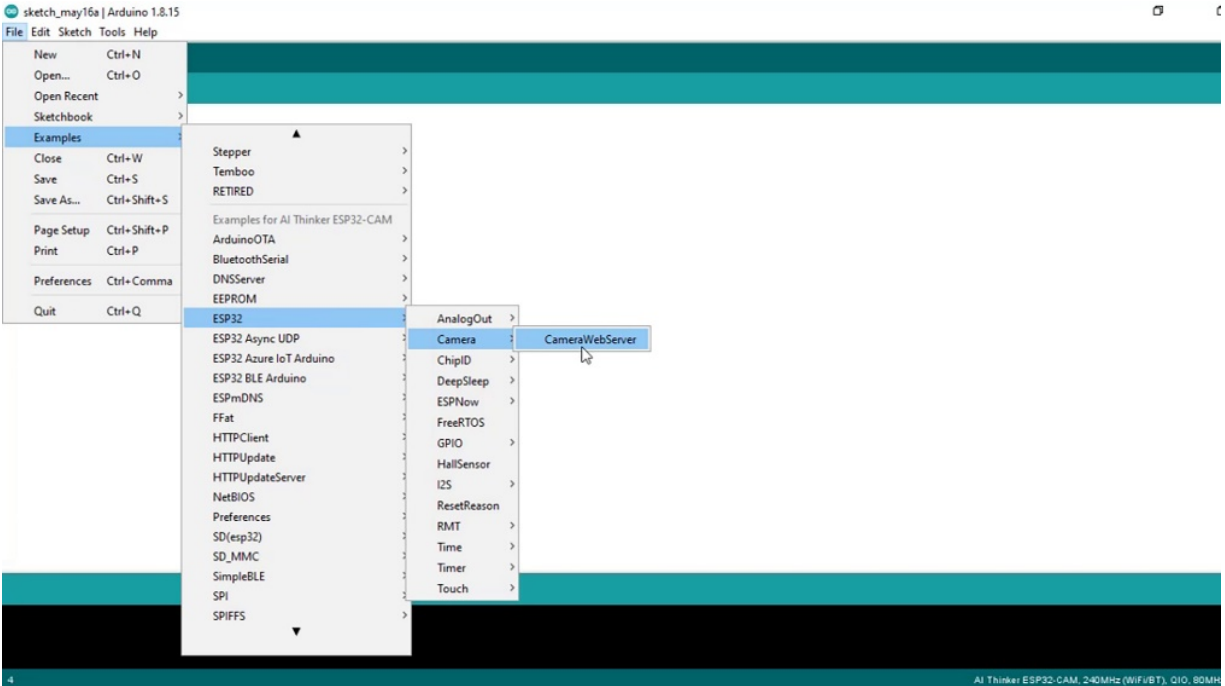
Because we are following this particular way, because these are unofficial alternate packages, but not being officially supported. So this is the reason why we have to give accustoming of the website. So you can install those I tonight, even in the slightest bit, so we shall wait for that to complete. Yeah, they have completed the installation of the robots, so you can see something like installed in his closet. No.



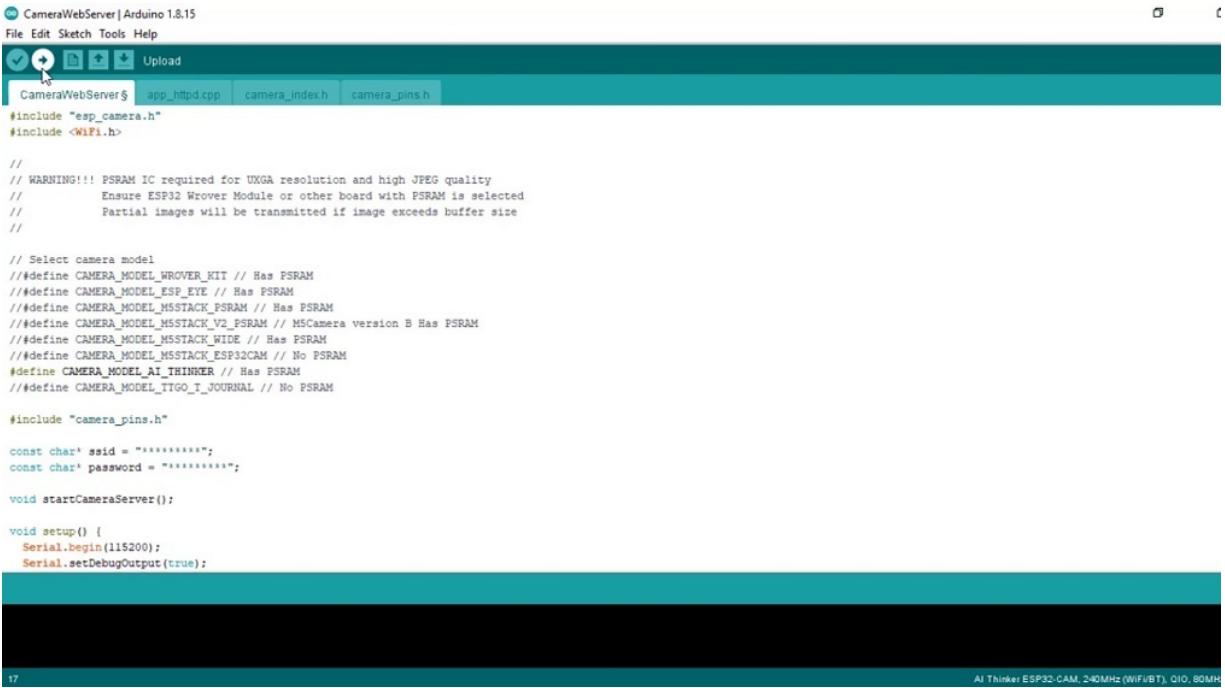
You can go to Tool's. Jack Fury is bitterly divided, so the board but what we are using is the. Yes, but there are two council meeting this particular. You can just celebrate. Once you've selected that, the frequency is automatically selected, flash frequency and the flash more.



Positive vote to come for us, because this is the only point available now for my laptop because I know things are connected, so I would select comfort for it is compulsory to sleep.

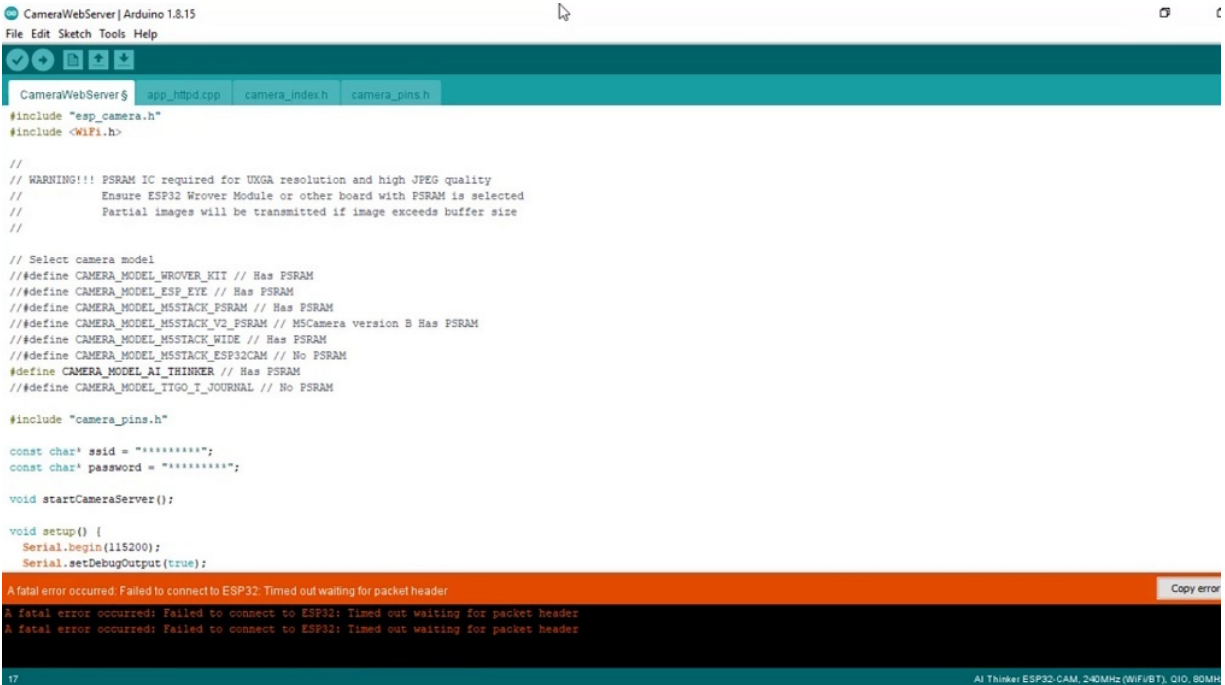


So let me search for some examples. OK, let us move the cameras and just try to learn something. This is a camera obscura example. So beautiful.



You're seeing many macros being created by a particular model, is used to camera from every angle. So let me just uncharming that and this is the button that you have to use this this will verify and this will upload. Now

you can observe and trying to connect to it, but it's not able to get into programming mode, so if you doesn't short and doesn't press the reset button, this is the state which will eventually happen if you keep on waiting for it. Let's see what happens.



```
CameraWebServer | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer$ app_httpd.cpp camera_index.h camera_pins.h
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//           Ensure ESP32 Wrover Module or other board with PSRAM is selected
//           Partial images will be transmitted if image exceeds buffer size
//
// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TIGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "*****";

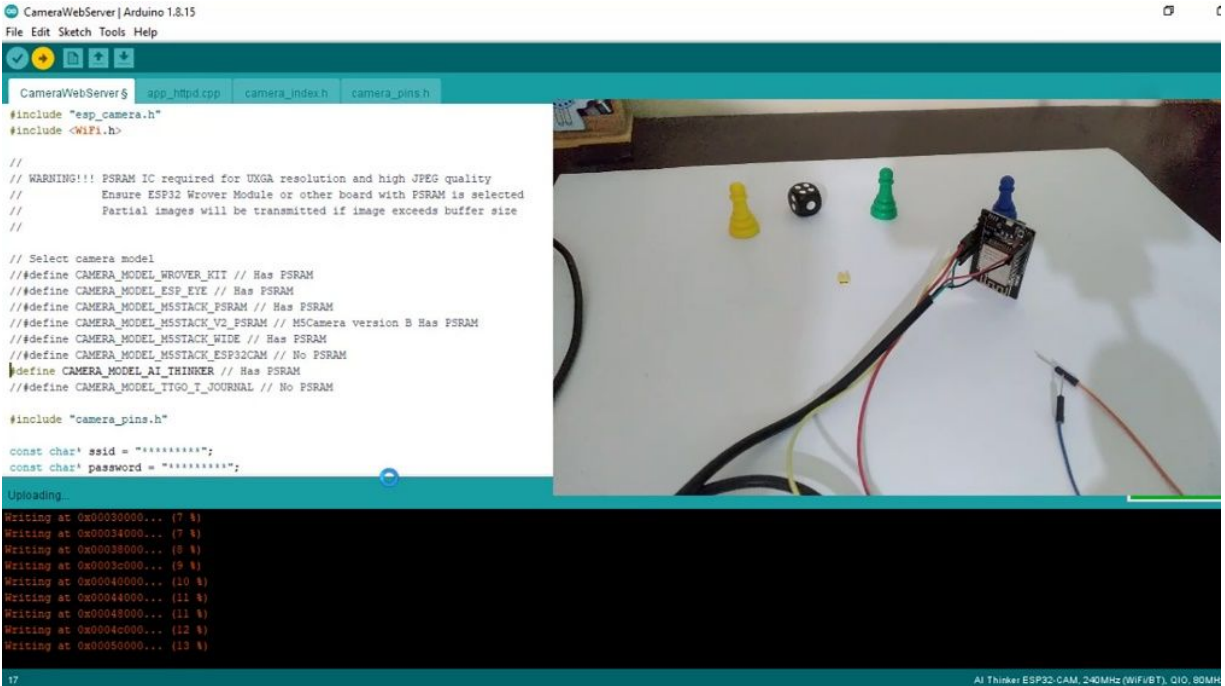
void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
}

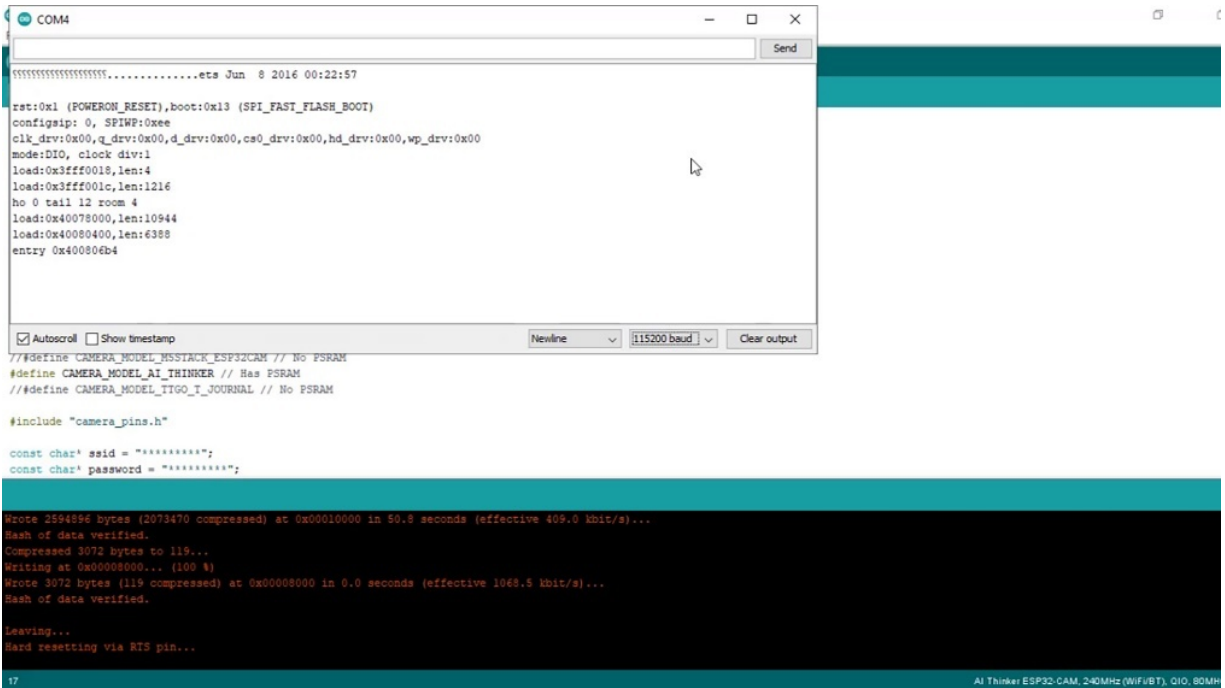
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header
A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

17 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MH
```

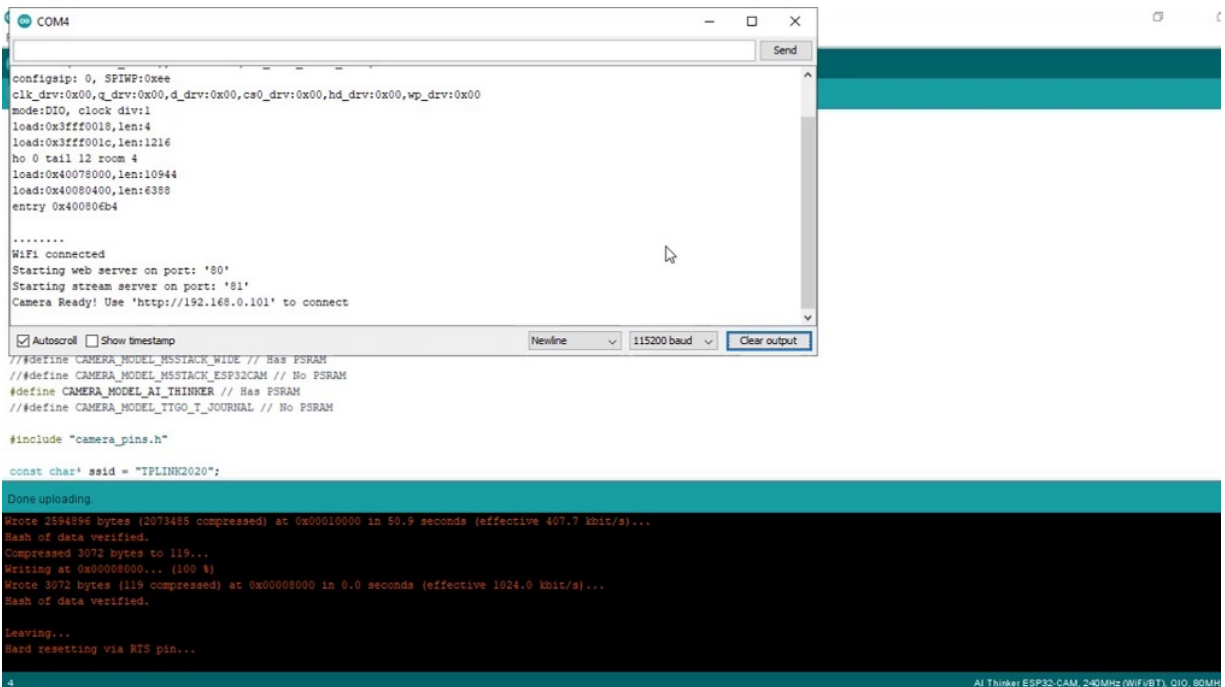
So this is what happens. That means there is no response from it. Now, let me upload again and this time.



On the hardware side, I will connect the talking points and I would Presti. This is what I have reset and just left the Fleischmann's. It start programming, surprisingly, should not happen, so at. Yes, I said it and I did not properly. So you can see the program is being loaded, the default application, which is coming along with that, it's quite heavy. So it takes a considerable amount of time. The program flashing is completed.



Now, what we can do is can simply. And the U.S. media once and again plug it and go to balls and go for monitor. So that is something coming the. Consecutive violent fight to moderate. So let me press the reset button on. Is it? So that is a problem, we have them here, you in the default application, you have to provide the society and password without which we can go further. So let me provide the this idea you have so you will be your wife reconnections. And this is only available, which is what I'm going to tell them. So let me introduce them in person and again, being pleased.



```
COM4
config:ip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,bd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4

.....
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.101' to connect

[Autoscroll] [Show timestamp] [Newline] [115200 baud] [Clear output]

// #define CAMERA_MODEL_ESP32CAM // Has PSRAM
// #define CAMERA_MODEL_AI_THINKER // Has PSRAM
// #define CAMERA_MODEL_TIGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

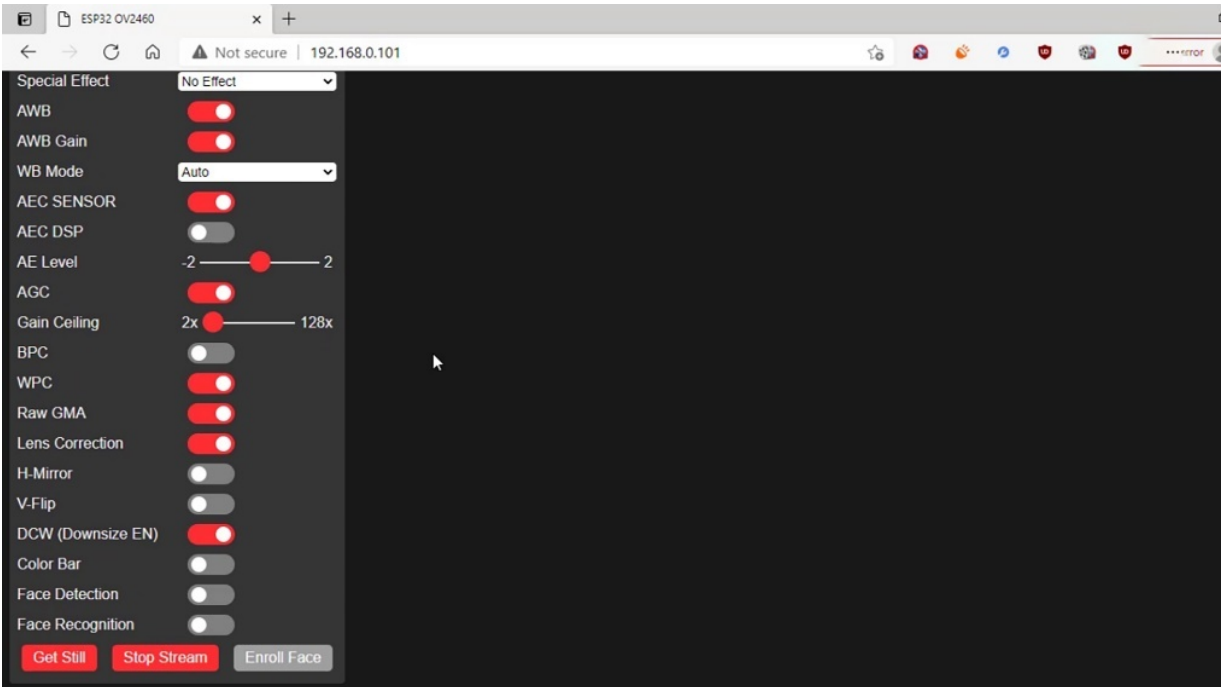
const char* ssid = "TPLINK2020";

Done uploading.
Wrote 2594896 bytes (2073485 compressed) at 0x00010000 in 50.9 seconds (effective 407.7 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 119...
Writing at 0x00080000... (100 %)
Wrote 3072 bytes (119 compressed) at 0x00080000 in 0.0 seconds (effective 1024.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

4 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz
```

So the programming is complete, go to Edwards because he didn't want the output and he said. So now NBC and the hardware will try to connect to the access point. So the Wi-Fi is connected? No, this is the IP address I have to use to connect to the camera. So there are no problems in front of the camera. Copy this IP address going to your browser just based this IP address.



Here comes the default to the website where we just posted on this pathetic attempt, you can just start stream. Actually, they're doing. Because.

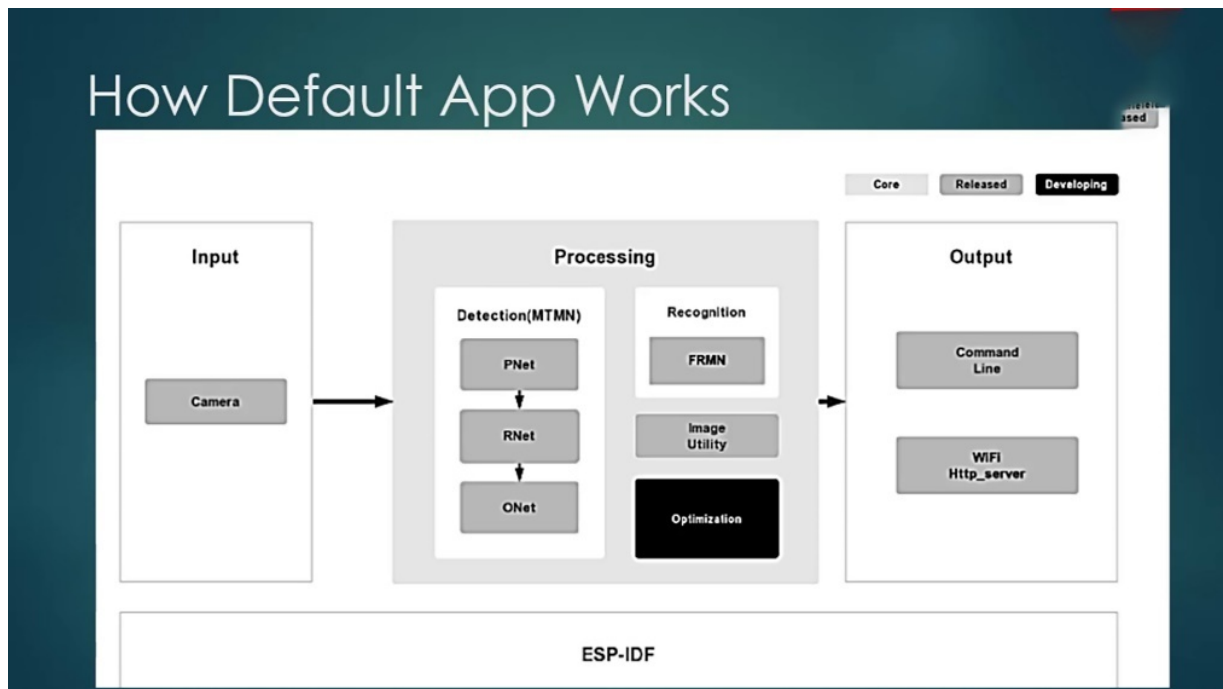
UNDERSTANDING DEFAULT APPLICATION

E.S.P digital camera as a station now there are two points here to understand, that is something called the station.



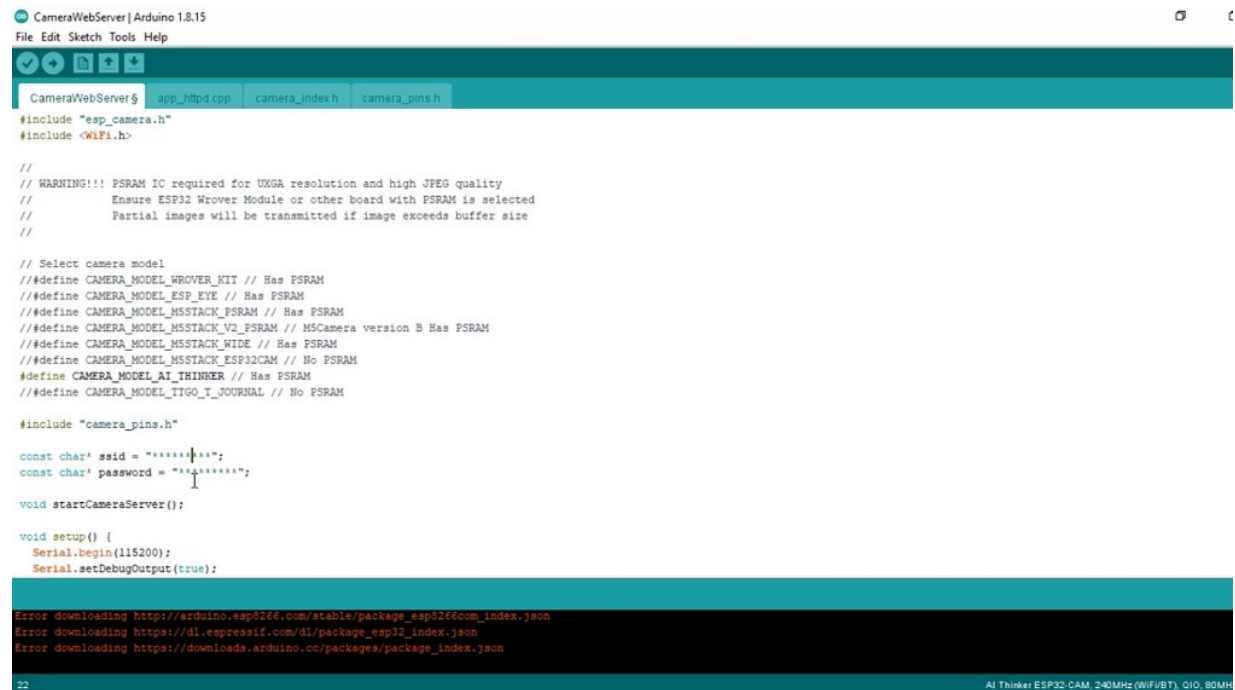
There is something called as an access point. So when you're actually connecting programming your default application, if you remember, we have given the credentials of the router that we want to connect to and from either a laptop or a mobile phone, we are connected to the router and we are accessing the webpage. Basically, these are multiple nodes which are connected to a single access point and the access point is actually allowed, it can route your request to and from some other nodes. So there is a reason why when you try to access it through a PC, you are able to create better to turn the page. The Web page is hosted on to this node and you are able to log into that. I mean, access that and seemed so different here is called as an access point and this is called station. There are two ways we can actually create our own access point using this particular camp. And instead of, for example, if you don't have a router, then what to do then? The best thing to do is to. Configure Urias petrochem as an access point and connect directly from your phone to the access point and you can access. Which is he is better. There is no need of Dr.. That is one technique we can follow. So here in this, I am trying to explain the default application, how it works, the default application works as extension. So in are coming close. We will try to change the default application into an access point. We do that. So before that, we'll see how the station works. So I will give you only a brief. I will not go deep into the default application because the default application is a

very complex application for a beginner level. How does the world that works this default app when you have switched on your speed dial again, it connects to the access point and when you try to access it through your phone with the world of Inspector Cookham that you are seeing on the screen, put it so we can actually make it static. Also incoming. Listen. I'll tell you how to make it static because every time you should be, you will not be trying to open the border and see what is the access point. Sorry, what is it? You get it. So instead we can put it a static IP and always open the same. Then start on. And. This is all the royal commission is working, but what is happening when you try to access the can, you are able to see a control page, completely legitimate page, which is hosted on the hospital. Again, that is sent to your phone. As a complete. Data, including the image and also video stream.



So I was saying about the first eviction, face recognition and first addiction algorithm, which is running, right. So I'm just giving a brief. The FBI is doing that. So it has a model called MDMA model, which is doing it. But still, I personally prefer not to use it better to camp for face recognition detection applications because I have found it very, very unsuitable. I will show you in the coming up samples also how it will feel. Most of the time it doesn't work, even if it's some Web pages or some Internet pointedly

showing you it is working, that in a very rare, optimal, proper lighting conditions warrant. It's difficult to.



```
CameraWebServer | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer$ app_httpd.cpp camera_index.h camera_pins.h
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//           Ensure ESP32 Wrover Module or other board with PSRAM is selected
//           Partial images will be transmitted if image exceeds buffer size
//
// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "123456789";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json

22 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), Q10, 80MH
```

OK, now we will go back to our call. Yes, so this is a different application. What we have seen, the primary steps, what we have done is we have committed are not required accountable being the aid we have for the username and password.


```
CAUsers\daramss\Desktop\temp.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp.c temp2.c
sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://";
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

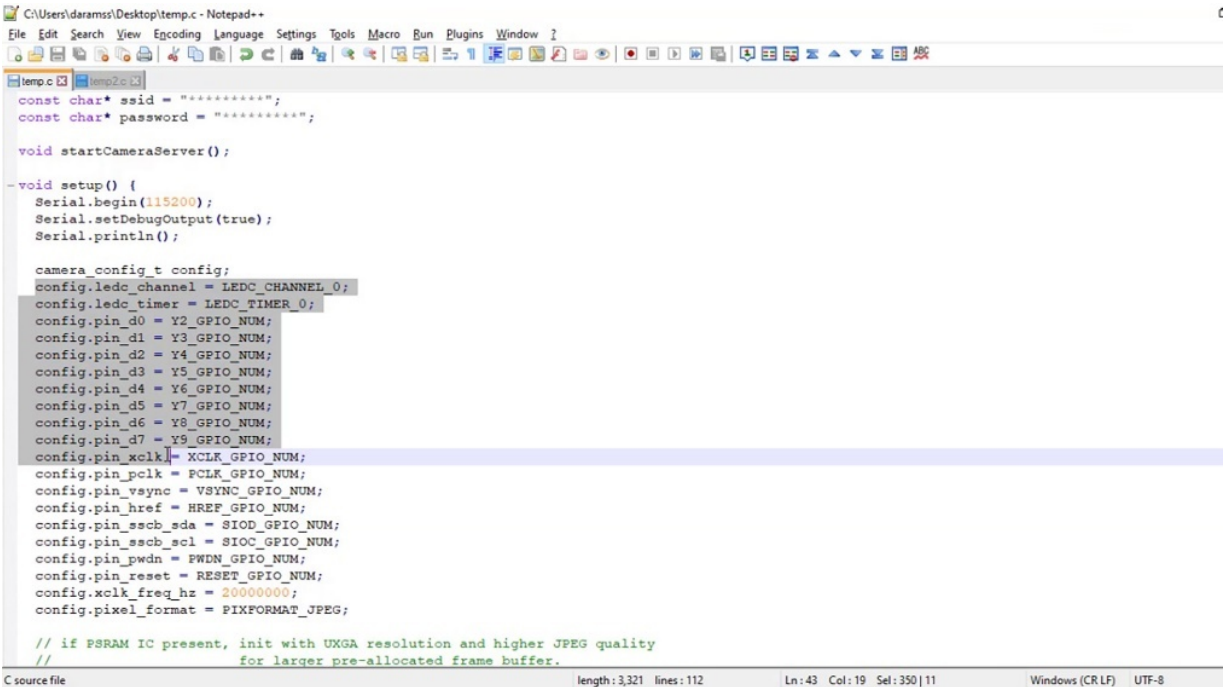
+void loop() {
```

Right. So in order to pass the code properly, I just open it in a notepad. This is not part of an internal battle. This is a very good, wonderful tool.

```
CAUsers\daramss\Desktop\temp.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp.c temp2.c
#include "e
#include <W
//
// WARNING!
//
//
//
// Select c
#define CAM
#define C
#define C
#define C
#define C
#define C
#define C
#define C
#include "c
const char*
const char*
void startC
void setup(
Serial.be
Serial.se
Serial.pr
camera_co
config.le
config.lea
```

So you can go to view and collapse, our fallen fall will fall. The main function calls, so that'll be easy for you to understand. So if you see the

orange, look, there isn't a thing in the room. But what is there in the setup before going is that, you know, only the camera model selection and the username and password veraciously password is good.



```
CAUsers\darams\Desktop\temp.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp.c temp2.c
const char* ssid = "*****";
const char* password = "*****";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sccb_sda = SIOD_GPIO_NUM;
  config.pin_sccb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  // for larger pre-allocated frame buffer.
}
C source file length: 3,321 lines: 112 Ln: 43 Col: 19 Sel: 350 | 11 Windows (CR LF) UTF-8
```

Right after that we configure the. Object called config, which is campaign camera configured as code, where we mention what are the pins of the spirit again, because in the years to come, liability for the specified Philippines, which are to be told that this particular pin is responsible for functional desex or pin number, did not in the label. So where do we define them?

```

CameraWebServer - camera_pins.h | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer $ app_httpd.cpp camera_index.h camera_pins.h
#define VSYNC_GPIO_NUM 22
#define HREF_GPIO_NUM 26
#define FCLK_GPIO_NUM 21

#ifdef CAMERA_MODEL_AI_THINKER
#define FWDN_GPIO_NUM 32
#define RESET_GPIO_NUM -1
#define XCLK_GPIO_NUM 0
#define SIOD_GPIO_NUM 26
#define SIOC_GPIO_NUM 27

#define Y9_GPIO_NUM 35
#define Y8_GPIO_NUM 34
#define Y7_GPIO_NUM 39
#define Y6_GPIO_NUM 36
#define Y5_GPIO_NUM 21
#define Y4_GPIO_NUM 19
#define Y3_GPIO_NUM 18
#define Y2_GPIO_NUM 5
#define VSYNC_GPIO_NUM 25
#define HREF_GPIO_NUM 23
#define FCLK_GPIO_NUM 22

#elif defined(CAMERA_MODEL_TTGO_T_JOURNAL)
#define FWDN_GPIO_NUM 0
#define RESET_GPIO_NUM 15
#define XCLK_GPIO_NUM 27
#define SIOD_GPIO_NUM 25
#define SIOC_GPIO_NUM 23

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json
110 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MH

```

If you go to the ordinal again, there is this camera up in storage file in which four corresponding camera, for example, asks the Edinger. So in Edinger, what are the PIN numbers? So this is the PIN number for winning this number. This you can check back with this and it will see exactly what is of using signal. What are the hits referencing?

```

CameraWebServer | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer $ app_httpd.cpp camera_index.h camera_pins.h
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//           Ensure ESP32 Wrover Module or other board with PSRAM is selected
//           Partial images will be transmitted if image exceeds buffer size
//

// Select camera model
#ifdef CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "*****";

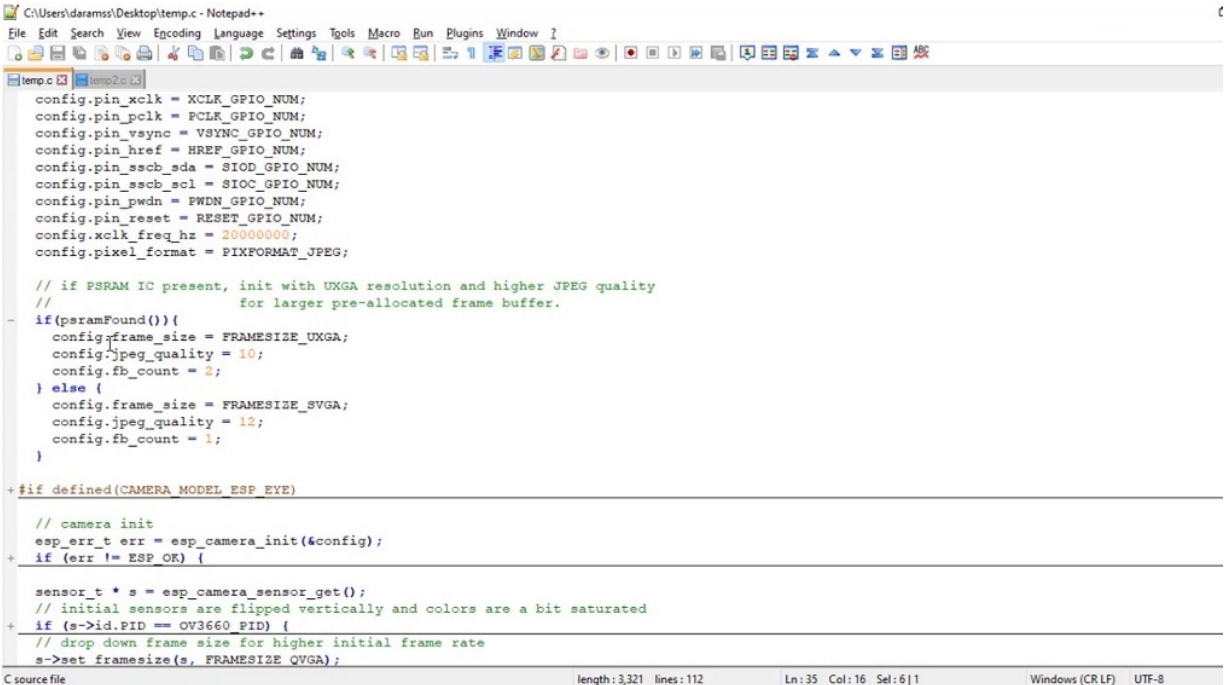
void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json
17 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MH

```

All this will be defined in the interface. So when we define this micro, those get included and the same that configured here, or if you wrongly configure, you will face problems. Are all the problems that you feel separately. Explain one more session.



```
temp.c
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 30000000;
config.pixel_format = PIXFORMAT_JPEG;

// if PSRAM IC present, init with UXGA resolution and higher JPEG quality
// for larger pre-allocated frame buffer.
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

+#if defined(CAMERA_MODEL_ESP_EYE)

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);
}

C source file          length: 3,321  lines: 112          Ln: 35  Col: 16  Sel: 6|1          Windows (CR LF)  UTF-8
```

Let's go back to Christmas. Yeah, so there is one check here. We have the bathroom. So once it is found, these are the configuration parameters, which will be the same like the frame size and what is the data quality as much lower that much better. And I do believe the parameters, if suppose if you are using any other Mordialloc SBA, there are some additional different.

```
CAUsers\daramss\Desktop\temp.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp.c temp2.c
)
-#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
- if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); // flip it back
  s->set_brightness(s, 1); // up the brightness just a bit
  s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);
1

+ #if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)

  WiFi.begin(ssid, password);

+ while (WiFi.status() != WL_CONNECTED) {
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

C source file length: 3,321 lines: 112 Ln: 80 Col: 30 Sel: 10 | 1 Windows (CR LF) UTF-8
```

So. If you go to some other camera models there, they change the brightness and saturation levels.

```
CAUsers\daramss\Desktop\temp.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp.c temp2.c
- if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); // flip it back
  s->set_brightness(s, 1); // up the brightness just a bit
  s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

-#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

- while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

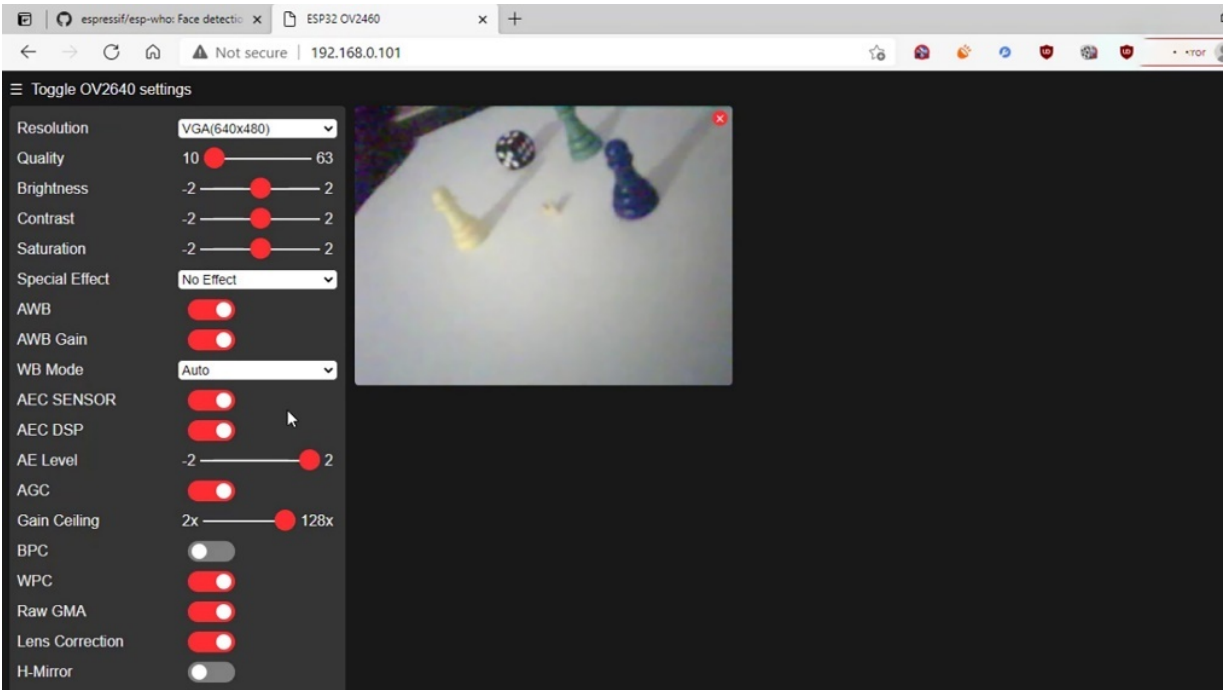
Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

-void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}

C source file length: 3,321 lines: 112 Ln: 102 Col: 20 Sel: 17 | 1 Windows (CR LF) UTF-8
```

So predominantly after that, we connect to WiFi. And then we start the camera said. That's the main program is simple, slow start cameras. What is

this? This is actually defined in the. Your application. This is the main application, right?



So when you open the hospital again, if you see the webpage now, I have put a very low resolution of sexology before it. So whatever you are seeing, the webpage is actually hoster onto the digital camera. Does that mean this Web server you are claimed were accessing from the laptop? The Web server is actually hosted onto the USB digital camera device. For example, if I try to perform Birkerts incursion or I try to do any operation or vertical flip, yeah, it happened a little bit. So whenever you do this operation, you are actually selling the control signal. From your phone or your laptop, and then he is with the camera performing this operation and updating the image and sending it back to your webpage. So this is all happening on the flight. So you're trying to control debate using the webpage and the functionalities on the other side are being also shown to you on the Web page. So any change you do in the webpage is getting reflected on image and it's been reverted back to you as a status. OK, but this is already working.

```
CameraWebServer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer $ app_httpd.cpp camera_index.h camera_pins.h

fb_gfx_drawFastVLine(fb, x+w-1, y, h, color);

#if 0
// landmark
int x0, y0, j;
for (j = 0; j < 10; j+=2) {
  x0 = (int)boxes->landmark[i].landmark_p[j];
  y0 = (int)boxes->landmark[i].landmark_p[j+1];
  fb_gfx_fillRect(fb, x0, y0, 3, 3, color);
}
#endif
}

static int run_face_recognition(dl_matrix3du_t *image_matrix, box_array_t *net_boxes){
  dl_matrix3du_t *aligned_face = NULL;
  int matched_id = 0;

  aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
  if(!aligned_face){
    Serial.println("Could not allocate face recognition buffer");
    return matched_id;
  }
  if (align_face(net_boxes, image_matrix, aligned_face) == ESP_OK){
    if (is_enrolling == 1){
      int0_t left_sample_face = enroll_face(sid_list, aligned_face);

      if(left_sample_face == (ENROLL_CONFIRM_TIMES - 1)){
        Serial.printf("Enrolling Face ID: %d\n", id_list.tail);
      }
    }
  }

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json

1 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz
```

So if I just put it up, so this Web page where I can see the e-mail call, the e-mail code of the Web page is actually compressed format. So this is the code that hits the mail page, which is considered. Compressed using a mother and then converted to Hex and then been given, as it were, as because the page is heavy, so the developers have tried to compress it and provide it as it compressed hex format.

```
CameraWebServer - camera_index.h | Arduino 1.8.15
File Edit Sketch Tools Help

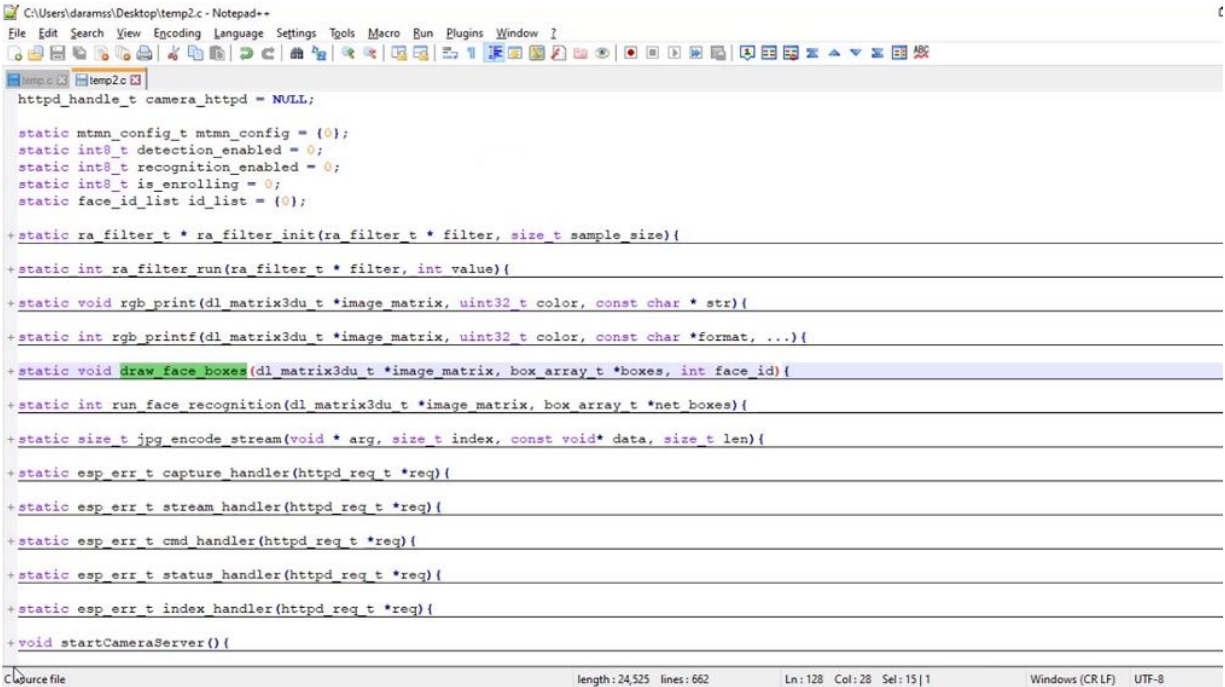
CameraWebServer $ app_httpd.cpp camera_index.h camera_pins.h

//File: index_ov2640.html.gz, Size: 4316
#define index_ov2640_html_gz_len 4316
const uint8_t index_ov2640_html_gz[] = {
0x1F, 0x8B, 0x08, 0x08, 0x50, 0x5C, 0xAE, 0x5C, 0x00, 0x03, 0x69, 0xE, 0x4, 0x65, 0x78, 0x5F,
0xF, 0x76, 0x32, 0x36, 0x34, 0x30, 0x2E, 0x68, 0x74, 0x6D, 0x6C, 0x00, 0xE3, 0x5D, 0x7B, 0x73,
0xD3, 0xC6, 0x16, 0xFF, 0x9F, 0x4F, 0x21, 0x04, 0x25, 0xF6, 0x34, 0x76, 0x6C, 0x7, 0x84, 0xE0,
0xDA, 0xE2, 0x42, 0x08, 0xD0, 0x19, 0x5E, 0x25, 0x2D, 0x74, 0xA6, 0xD3, 0x81, 0xB5, 0xB4, 0xB2,
0x55, 0x64, 0xC9, 0x95, 0x56, 0x76, 0x52, 0x26, 0x9F, 0xE3, 0x7E, 0xA0, 0xFB, 0xC5, 0xEE, 0xD9,
0x7, 0xA4, 0x95, 0xBC, 0x7A, 0xD8, 0x26, 0x36, 0x97, 0xEB, 0xCC, 0x14, 0xD9, 0xDA, 0x73, 0xF6,
0x9C, 0xF3, 0x3B, 0xAF, 0x5D, 0x3D, 0x3A, 0xBC, 0x6D, 0xF9, 0x26, 0xB9, 0x9A, 0x63, 0xED, 0x4A,
0x66, 0xAE, 0x71, 0xEB, 0xC8, 0xFF, 0xD1, 0xE0, 0x33, 0x9C, 0x62, 0x64, 0xF1, 0x43, 0xF6, 0x75,
0x86, 0x09, 0xD2, 0xCC, 0x29, 0x0A, 0x42, 0x4C, 0x46, 0x7A, 0x44, 0xEC, 0xD6, 0xA9, 0x9E, 0x3F,
0xED, 0xA1, 0x19, 0x1E, 0xE9, 0x0B, 0x07, 0x2F, 0xE7, 0x7E, 0x40, 0x74, 0xCD, 0xF4, 0x3D, 0x82,
0x3D, 0x18, 0xBE, 0x74, 0x2C, 0x32, 0x1D, 0x59, 0x78, 0xE1, 0x98, 0xB8, 0xC5, 0xBE, 0x1C, 0x3A,
0x9E, 0x43, 0x1C, 0xE4, 0xB6, 0x42, 0x13, 0xB9, 0x78, 0xD4, 0x95, 0x79, 0x11, 0x87, 0xB8, 0xD8,
0x38, 0xBF, 0x78, 0x7B, 0xDC, 0xD3, 0xDE, 0xBC, 0xEF, 0xF5, 0x4F, 0x3A, 0xC3, 0x23, 0xFF, 0x5B,
0x3A, 0x26, 0x24, 0x57, 0xF2, 0x77, 0xFA, 0x19, 0xFB, 0xD6, 0x95, 0xF6, 0x25, 0xF3, 0x13, 0xFD,
0xD8, 0x20, 0x44, 0xCB, 0x46, 0x33, 0xC7, 0xBD, 0x1A, 0x68, 0x8F, 0x03, 0x98, 0xF3, 0xF0, 0x05,
0x76, 0x17, 0x98, 0x38, 0x26, 0x3A, 0x0C, 0x91, 0x17, 0xB6, 0x42, 0x1C, 0x38, 0xF6, 0x4F, 0x2B,
0x84, 0x63, 0x64, 0x7E, 0x9E, 0x04, 0x7E, 0xE4, 0x59, 0x03, 0xED, 0x4E, 0xF7, 0x94, 0xFF, 0xAD,
0xE, 0x32, 0x7D, 0xD7, 0x0F, 0xE0, 0xFC, 0xF9, 0x33, 0xFA, 0xB7, 0x7A, 0x9E, 0xCD, 0x1E, 0x3A,
0xFF, 0xE0, 0x81, 0xD6, 0x3D, 0x99, 0x5F, 0x66, 0xCE, 0x5F, 0xDF, 0xCA, 0x7C, 0x9D, 0xF6, 0x3A,
0xA4, 0x17, 0xF4, 0xA7, 0xE5, 0xF4, 0x21, 0x36, 0x89, 0xE3, 0x7B, 0xED, 0x19, 0x72, 0x3C, 0x05,
0x27, 0xCB, 0x09, 0xE7, 0x2E, 0x02, 0x1B, 0xD8, 0x2E, 0x2E, 0xE5, 0x73, 0x67, 0x86, 0xBD, 0xE8,
0xB0, 0x82, 0x1B, 0x65, 0xD2, 0xB2, 0x9C, 0x80, 0x8F, 0x1A, 0x50, 0x3B, 0x44, 0x33, 0xAF, 0x92,
0xED, 0x99, 0x5C, 0x9E, 0xEF, 0x61, 0x85, 0x01, 0xE9, 0x44, 0xCB, 0x00, 0xCD, 0x9, 0x00, 0xFA,
0xEF, 0xEA, 0x90, 0x99, 0xE3, 0x71, 0xA7, 0x1A, 0x68, 0xC7, 0xFD, 0xCE, 0xFC, 0xB2, 0x02, 0xCA,
0xE3, 0x13, 0xFA, 0xB7, 0x3A, 0x68, 0x8E, 0x2C, 0xCB, 0xF1, 0x26, 0x03, 0xED, 0x54, 0xC9, 0xC2,

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json

1 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz
```

Into the circle, so insanely so we actually decompress it and to show it to the youth, so we are not going into detail of this particular technique because this is very clumsy and like very complex way to do things. But in your recommendations, I will try to write our own history, will page our own application of actually streaming the video.



```
CAUsers\darams\Desktop\temp2.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp2.c temp2.c
httpd_handle_t camera_httpd = NULL;

static mtmn_config_t mtmn_config = {};
static int8_t detection_enabled = 0;
static int8_t recognition_enabled = 0;
static int8_t is_enrolling = 0;
static face_id_list id_list = {};

+static ra_filter_t * ra_filter_init(ra_filter_t * filter, size_t sample_size){
+static int ra_filter_run(ra_filter_t * filter, int value){
+static void rgb_print(dl_matrix3du_t *image_matrix, uint32_t color, const char * str){
+static int rgb_printf(dl_matrix3du_t *image_matrix, uint32_t color, const char *format, ...){
+static void draw_face_boxes(dl_matrix3du_t *image_matrix, box_array_t *boxes, int face_id){
+static int run_face_recognition(dl_matrix3du_t *image_matrix, box_array_t *net_boxes){
+static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len){
+static esp_err_t capture_handler(httpd_req_t *req){
+static esp_err_t stream_handler(httpd_req_t *req){
+static esp_err_t cmd_handler(httpd_req_t *req){
+static esp_err_t status_handler(httpd_req_t *req){
+static esp_err_t index_handler(httpd_req_t *req){
+void startCameraServer(){

Source file length: 24,525 lines: 662 Ln: 128 Col: 28 Sel: 15 | 1 Windows (CR LF) UTF-8
```

OK, if you see the application, I just open it in the back. It's part of the problem. So start observe. It is actually present here.


```
CAUsers\daramss\Desktop\temp2.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp2.c temp2.c
+static int run_face_recognition(dl_matrix3du_t *image_matrix, box_array_t *net_boxes){
+static size_t jpg_encode_stream(void * arg, size_t index, const void* data, size_t len){
+static esp_err_t capture_handler(httpd_req_t *req){
+static esp_err_t stream_handler(httpd_req_t *req){
+static esp_err_t cmd_handler(httpd_req_t *req){
+static esp_err_t status_handler(httpd_req_t *req){
+static esp_err_t index_handler(httpd_req_t *req){

-void startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
+    httpd_uri_t index_uri = {
+    httpd_uri_t status_uri = {
+    httpd_uri_t cmd_uri = {
+    httpd_uri_t capture_uri = {
+    httpd_uri_t stream_uri = {

    ra_filter_init(&ra_filter, 20);

    mtmn_config.type = FAST;
    mtmn_config.min_face = 80;
    mtmn_config.pyramid = 0.707;
    mtmn_config.pyramid_times = 4;

C source file length: 24,525 lines: 662 Ln: 594 Col: 26 Sel: 9 | 1 Windows (CR LF) UTF-8
```

So what is happening here is so there are multiple Yubari handlers for indexing. There is one handler for starters. There is one candidate for command. There is one handler for capturing the design for streaming. There is a handler, for example. Capturing this then gets delice. Activity, it stops the stream so you can just see it, so you can just against our district.

```
CAUsers\daramss\Desktop\temp2.c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp2.c temp2.c
    ra_filter_init(&ra_filter, 20);

    mtmn_config.type = FAST;
    mtmn_config.min_face = 80;
    mtmn_config.pyramid = 0.707;
    mtmn_config.pyramid_times = 4;
    mtmn_config.p_threshold.score = 0.6;
    mtmn_config.p_threshold.nms = 0.7;
    mtmn_config.p_threshold.candidate_number = 20;
    mtmn_config.r_threshold.score = 0.7;
    mtmn_config.r_threshold.nms = 0.7;
    mtmn_config.r_threshold.candidate_number = 10;
    mtmn_config.o_threshold.score = 0.7;
    mtmn_config.o_threshold.nms = 0.7;
    mtmn_config.o_threshold.candidate_number = 1;

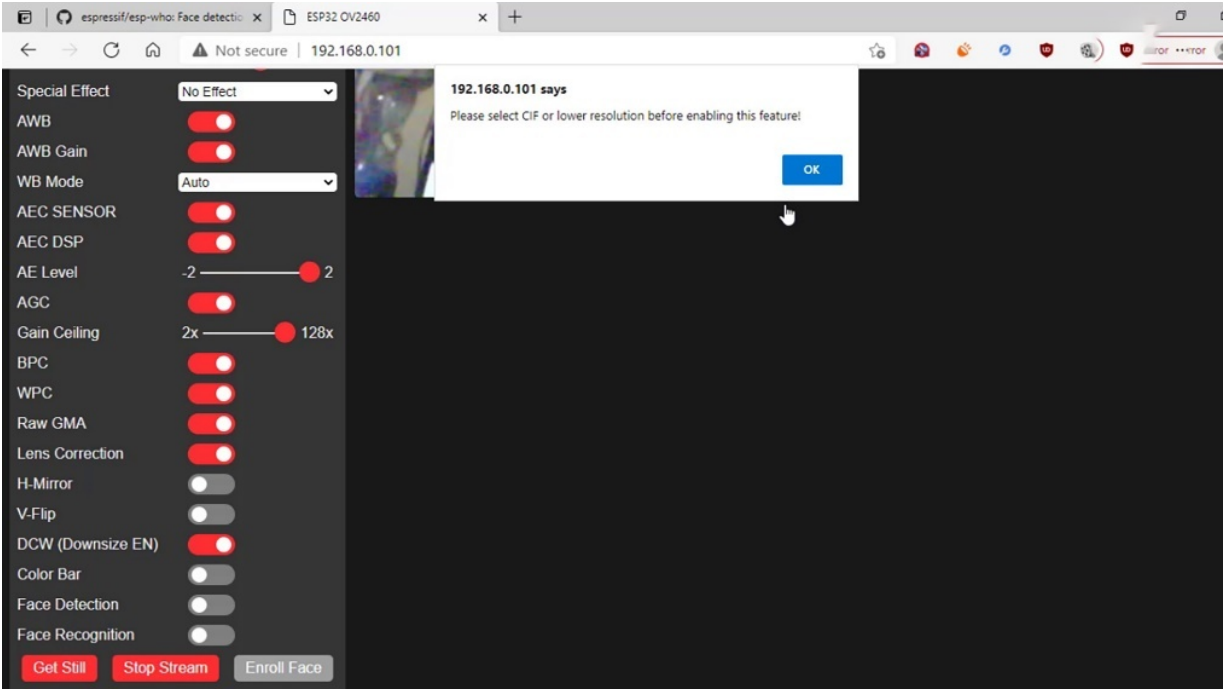
    face_id_init(&id_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);

    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(camera_httpd, &index_uri);
        httpd_register_uri_handler(camera_httpd, &cmd_uri);
        httpd_register_uri_handler(camera_httpd, &status_uri);
        httpd_register_uri_handler(camera_httpd, &capture_uri);
    }

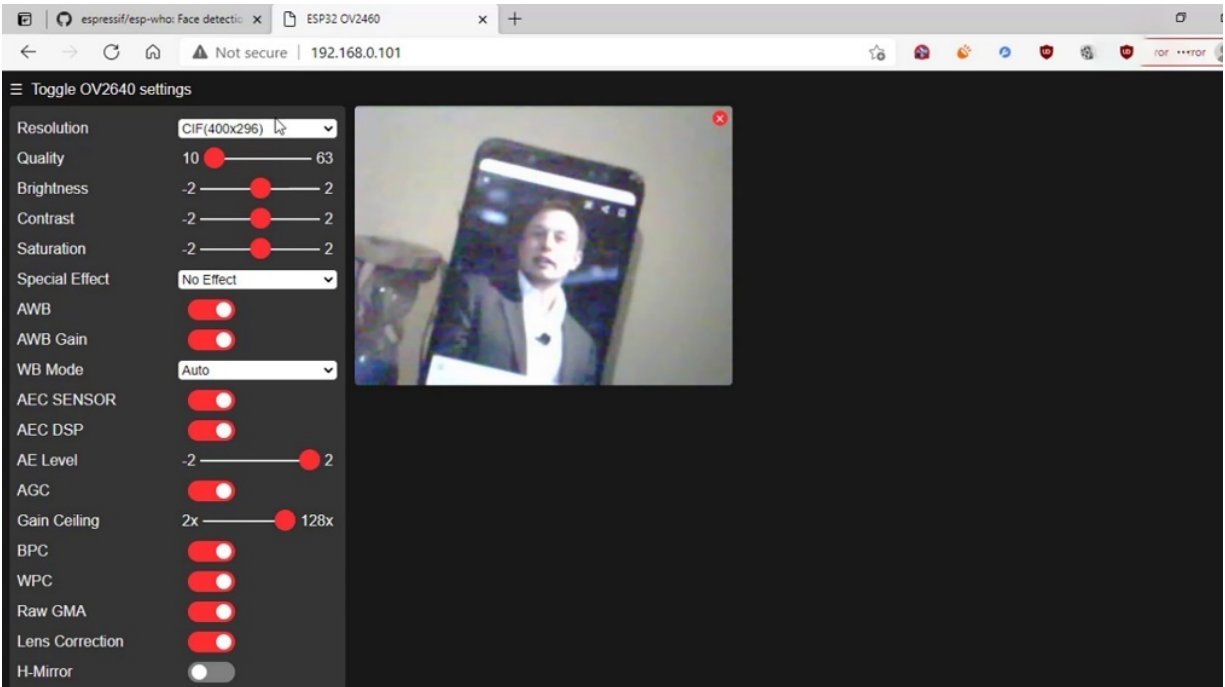
    config.server_port += 1;
    config.ctrl_port += 1;
    Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
    if (httpd_start(&stream_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(stream_httpd, &stream_uri);
    }
}

C source file length: 24,525 lines: 662 Ln: 622 Col: 26 Sel: 10 | 1 Windows (CR LF) UTF-8
```

OK, so these things are there and for example, yeah, these all these are being operated under the single umbrella. And apart from that, we have is streaming over different body. So these are maintaining a single and this is the streaming is meeting on a different footing. So these are all the abdomen functions, the medium and configuration and diamond object they was showing in the paper. So these are the first reaction and the recognition barometer's face recognition letters enable you to ask me to change the resolution.



Yes, OK, I love it. You know, this is a resolution where the basic condition starts. It's a very, very low resolution. You cannot do resolutions and and very high resolutions.



And it's unrestrained. It changes parameters so that it can do this restriction. Mm hmm. No, it's to do so. This is a typical example, like I suppose it is any raspberry by Damara or something else. It would definitely do it immediately. But does he did it again not meant for those kind of applications. So that is why I don't recommend my personal experience is very bad. So I'm going to do some pregnancy and think happens. No, I mean you start the application again with the same resolution restriction. Well, I just mean. No, even if a particular was not able to get any it. So this is where it is OK to go back to the cold.

```
CAUsers\daramss\Desktop\temp2c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp2c temp2c
ra_filter_init(&ra_filter, 20);

mtmn_config.type = FAST;
mtmn_config.min_face = 80;
mtmn_config.pyramid = 0.707;
mtmn_config.pyramid_times = 4;
mtmn_config.p_threshold.score = 0.6;
mtmn_config.p_threshold.nms = 0.7;
mtmn_config.p_threshold.candidate_number = 20;
mtmn_config.r_threshold.score = 0.7;
mtmn_config.r_threshold.nms = 0.7;
mtmn_config.r_threshold.candidate_number = 10;
mtmn_config.o_threshold.score = 0.7;
mtmn_config.o_threshold.nms = 0.7;
mtmn_config.o_threshold.candidate_number = 1;

face_id_init(&id_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);

Serial.printf("Starting web server on port: '%d'\n", config.server_port);
if (httpd_start(&camera_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(camera_httpd, &index_uri);
    httpd_register_uri_handler(camera_httpd, &cmd_uri);
    httpd_register_uri_handler(camera_httpd, &status_uri);
    httpd_register_uri_handler(camera_httpd, &capture_uri);
}

config.server_port += 1;
config.ctrl_port += 1;
Serial.printf("Starting stream server on port: '%d'\n", config.server_port);
if (httpd_start(&stream_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(stream_httpd, &stream_uri);
}
}

C source file length: 24,525 lines: 662 Ln: 660 Col: 35 Sel: 26 | 1 Windows (CR LF) UTF-8
```

So these are different handlers and the cameras are what is considered so gnarliest, given the handler mean, OK, and the handler has been assigned to a particular delivery, is particularly resistant. These are highly that. So whenever a request comes, what should be done will be addressed by the handler as the normal interpellation routines that you do in not only controllers.

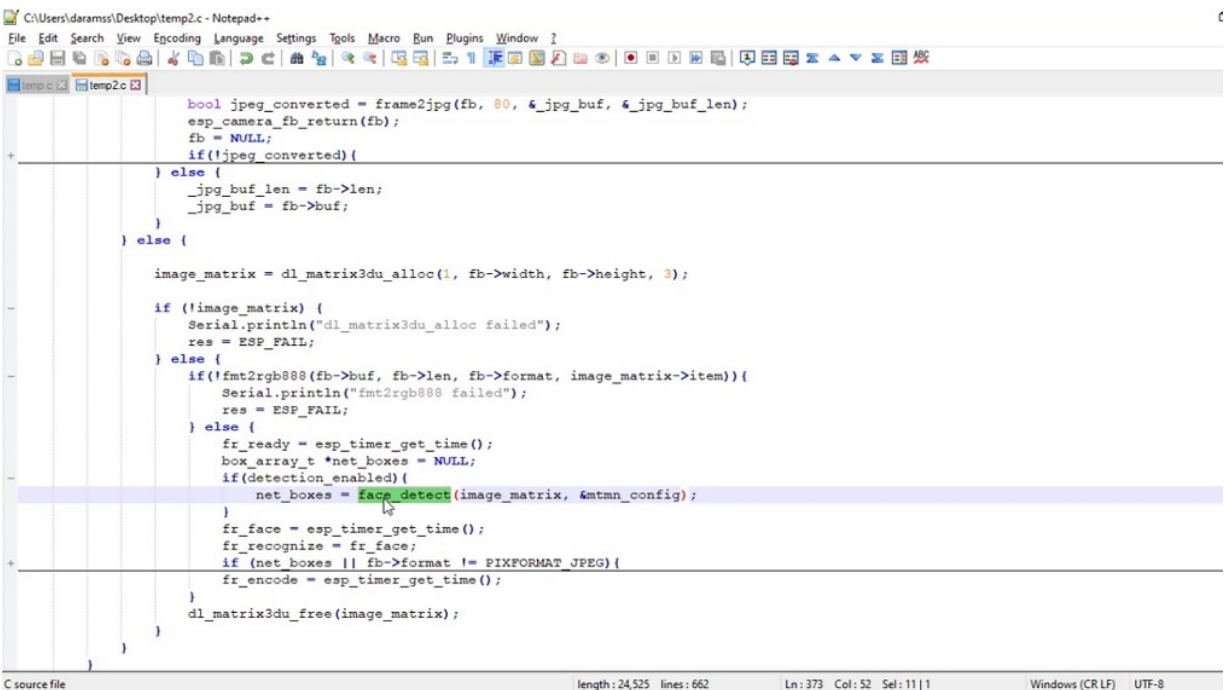
```
CAUsers\daramss\Desktop\temp2c - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
temp2c temp2c
if (res != ESP_OK) {
    return res;
}

httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "");

while(true) {
    detected = false;
    face_id = 0;
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    } else {
        fr_start = esp_timer_get_time();
        fr_ready = fr_start;
        fr_face = fr_start;
        fr_encode = fr_start;
        fr_recognize = fr_start;
        if (!detection_enabled || fb->width > 400) {
            if (fb->format != PIXFORMAT_JPEG) {
                bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
                esp_camera_fb_return(fb);
                fb = NULL;
                if (!jpeg_converted) {
                    } else {
                        _jpg_buf_len = fb->len;
                        _jpg_buf = fb->buf;
                    }
                } else {
                    image_matrix = dl_matrix3du_alloc(1, fb->width, fb->height, 3);
                    if (!image_matrix) {

```

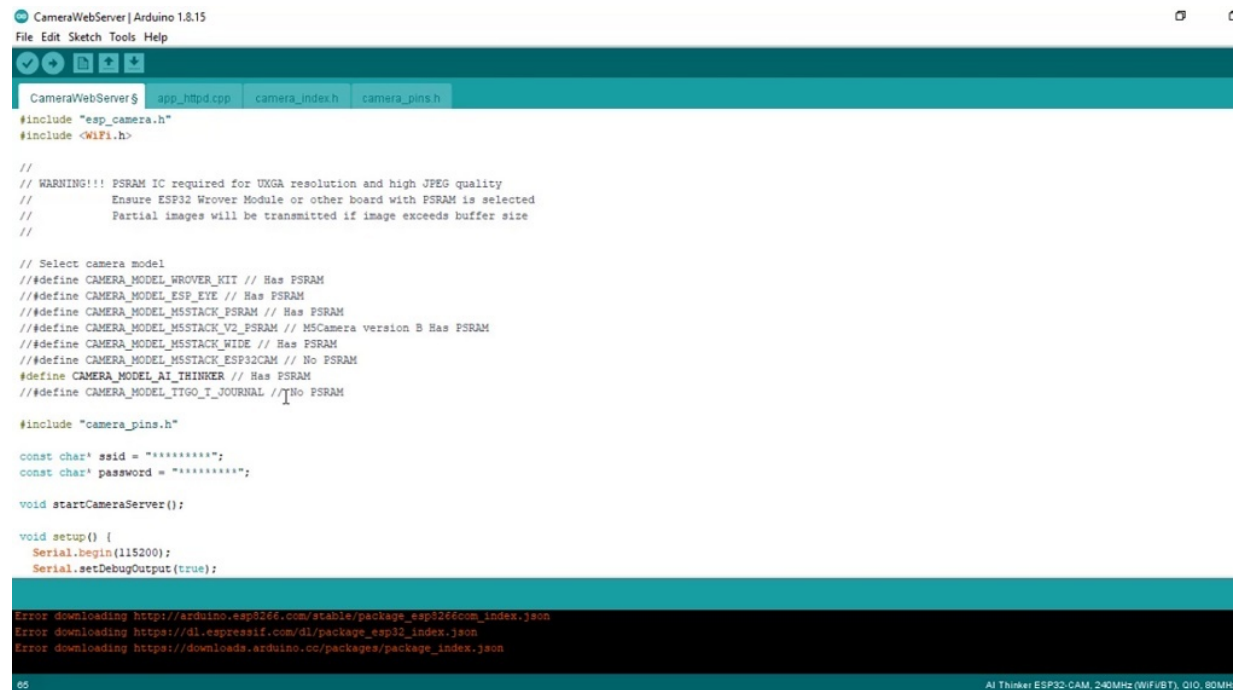
So coming to that, for example, let us see what has happened in a streamlined, streamlined look. Typically they get the frame of the frame from the camera and then we check further reductions in employment and then hopefully we can work differently. And then if any detection is enabled, we try to create a box set on that and pass it through the screen. Can we do our job and then do the detection objectives like or take these function calls, have access to from the hospital visits when they will be and then whatever results come from that these boxes will be tried as a matrix and we try it as a line onto the onto the image and we again send it back to the plane.



```
bool jpeg_converted = frame2jpg(fb, 80, &_jpg_buf, &_jpg_buf_len);
esp_camera_fb_return(fb);
fb = NULL;
if(!jpeg_converted){
} else {
    _jpg_buf_len = fb->len;
    _jpg_buf = fb->buf;
}
} else {
    image_matrix = dl_matrix3du_alloc(1, fb->width, fb->height, 3);
    if (!image_matrix) {
        Serial.println("dl_matrix3du_alloc failed");
        res = ESP_FAIL;
    } else {
        if(!fmt2rgb888(fb->buf, fb->len, fb->format, image_matrix->item)){
            Serial.println("fmt2rgb888 failed");
            res = ESP_FAIL;
        } else {
            fr_ready = esp_timer_get_time();
            box_array_t *net_boxes = NULL;
            if(detection_enabled){
                net_boxes = face_detect(image_matrix, &mtmn_config);
            }
            fr_face = esp_timer_get_time();
            fr_recognize = fr_face;
            if (net_boxes || fb->format != PIXFORMAT_JPEG){
                fr_encode = esp_timer_get_time();
            }
            dl_matrix3du_free(image_matrix);
        }
    }
}
```

All of these are like readings. This is being done. But sometimes it works, sometimes it doesn't work. As I said, the lighting conditions, multiple factors. OK, so this is how on a nutshell, if we like to briefly see how this different application is working. So it does all things implemented on it, but it's a very hefty application. So we will not try to use this. We will try to build everything from scratch in the coming lectures. So it is OK and coming lectures will see how to treat the camera model. How do understand those? What are the parameters? How can it be? How can I add just my quality to the optimum level and what each parameter corresponds to and

how to grab onto a memory card and of how to control your models, how to control your samples and then do the surveillance.



```
CameraWebServer | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer$ app_httpd.cpp camera_index.h camera_pins.h
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
//           Ensure ESP32 Wrover Module or other board with PSRAM is selected
//           Partial images will be transmitted if image exceeds buffer size
//
// Select camera model
#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
#define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TIGO_T_JOURNAL //No PSRAM

#include "camera_pins.h"

const char* ssid = "*****";
const char* password = "*****";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);

Error downloading http://arduino.esp8266.com/stable/package_esp8266com_index.json
Error downloading https://dl.espressif.com/dl/package_esp32_index.json
Error downloading https://downloads.arduino.cc/packages/package_index.json

55 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), Q10, 80MH
```

And we can also we'll also see how to record your video and even live streaming already. You always want to see the just the wonderful part of it without any help. But trust me, we do not face recognition and face detection purposes, though it can do some kind of level of color recognition objectives. I can see, but it's not meant for that segment for recognition and detection. But do we require much government sensors? But maybe we can do the things and be watching your friends, maybe someone from the government if we include this in security, but. It's way beyond the cost in operations per second world. OK, so let us go forward and I do the surveillance camera applications, which is this device is a perfect storm.

ESP32-CAM AS ACCESSPOINT

in the previous application, we have seen how the hospital is configured as in the station and we were connecting to the road. Now in this application, are this a small cottage? We'll try to see how we can make it better. You can be here as access point. And we the stations from our mobile phones are from the laptops. We try to connect to the access point and all the controls. Difference between generator directly happen between the cavity and the last straightaway jumper to the core. It's very simple. Before that, I just want you to download this notepad. Plus you can just go to the downloads of whatever version which is compatible for you. This is the latest release point in point site.



You can go for like 30 minutes and can download it and you can install it. So that's a very hard data which will help you to understand what the basic difference, what we have done with the calls. OK, so let me open up our website. Overapplication there. I have made. Yes, better to be here as access point.

```
D:\MIUbi\ESP32_CAM\Code\1.Lab2_CameraWebServer_Default\CameraWebServer\CameraWebServer.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CameraWebServer_AP.ino CameraWebServer.ino
}
sensor_t * s = esp_camera_sensor_get();
//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); //Flip it back
  s->set_brightness(s, 1); //up the blightness just a bit
  s->set_saturation(s, -2); //lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_OVGA);
#if defined(CAMERA_MODEL_MSSTACK_WIDE)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
startCameraServer();
Serial.print("Camera Ready! Use 'http://";
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}
void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

C++ source file length: 2,841 lines: 107 Ln: 74 Col: 1 Sel: 75 | 1 Windows (CR LF) UTF-8

So let me open notepad placeless. So I have already opened the files, but see, I am trying to compare here our previous file. I mean, the previous record where we have seen the default in this lecture, we just had a small three lines of code which makes the default application to be here. But at this point.

```
D:\MIUbi\ESP32_CAM\Code\1.Lab2_CameraWebServer_Default\CameraWebServer\CameraWebServer.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CameraWebServer_AP.ino CameraWebServer.ino
}
sensor_t * s = esp_camera_sensor_get();
//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1); //Flip it back
  s->set_brightness(s, 1); //up the blightness just a bit
  s->set_saturation(s, -2); //lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_OVGA);
#if defined(CAMERA_MODEL_MSSTACK_WIDE)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
startCameraServer();
Serial.print("Camera Ready! Use 'http://";
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}
void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

Plugins Admin

Available Updates Installed

Search: comp Next Install

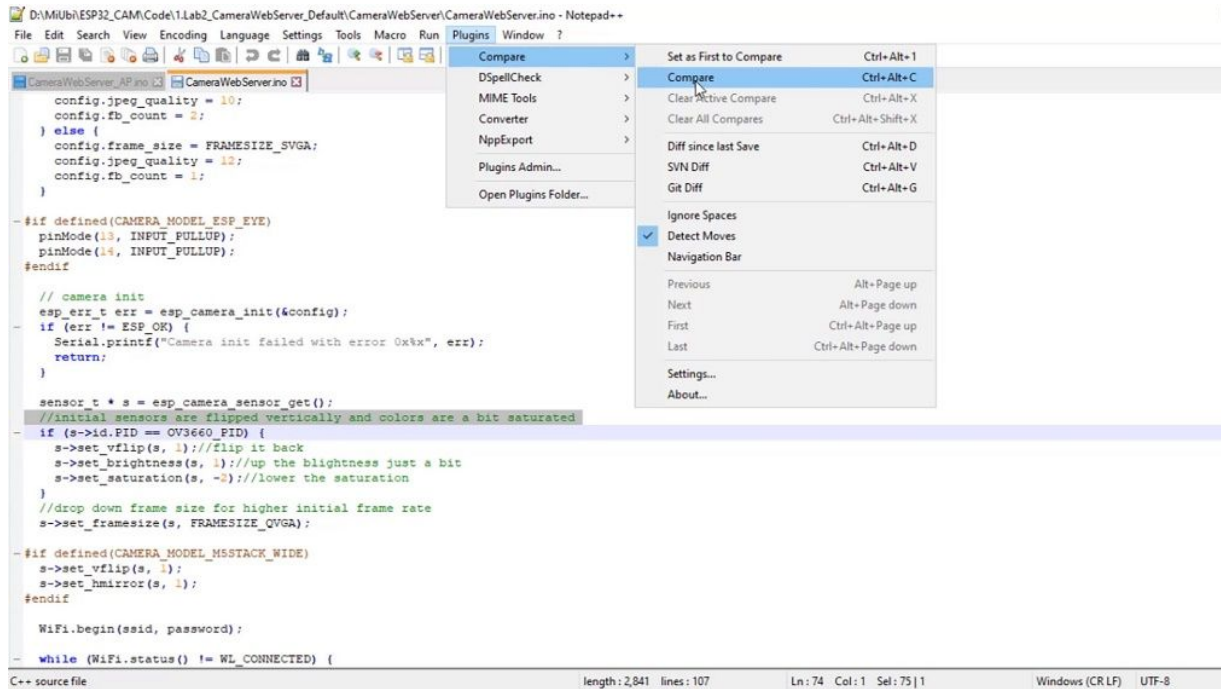
Plugin	Version
<input type="checkbox"/> BigFiles - Open Very Large Files	0.1.3
<input type="checkbox"/> Bookmarks@Dook	2.1
<input type="checkbox"/> CADdyTools	1.1.3.7
<input type="checkbox"/> Code Alignment	14.1.107
<input type="checkbox"/> Code:Stats	1.0.1
<input type="checkbox"/> ColumnTools	1.4.1.1
<input type="checkbox"/> Comment Wrap	1.0.0.3
<input checked="" type="checkbox"/> Compare	2.0.1

Shows the differences between 2 files (side by side).
Author: Ty Landercaasper, Jean-Sebastien Leroy, Pavel Nidev
Homepage: <https://github.com/jleroy/compare-plugin>

Close

C++ source file length: 2,841 lines: 107 Ln: 74 Col: 1 Sel: 75 | 1 Windows (CR LF) UTF-8

OK, so I just want to tell you the comparison, how I read it. So I just go to your to install a plugin called Come Back to see whether what we have done. So this is a very handy tool. That's why I'm telling you, just go to Twitter and Instagram, just search for comfort and just trying to install it. Yes, install it.



Yeah. Now, if you see plug ins, there is a tool called combat where you can just go to compare and compare. These two will be over with it right now.

```
D:\MIUb\ESP32_CAM\Code\1.Lab2_CameraWebServer_Default\CameraWebServer\CameraWebServer.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CameraWebServer_AP.ino CameraWebServer.ino
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
// or another board which has PSRAM enabled
//

// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_ESP_EYE
// #define CAMERA_MODEL_MSSTACK_PSRAM
// #define CAMERA_MODEL_MSSTACK_WIDE
#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

const char* ssid = "ESP32-CAM";
const char* password = "password";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
}

C++ source file length: 2,883 lines: 111 Ln: 18 Col: 1 Pos: 418 Windows (CR LF) UTF-8
```

Here, the society remains saying no, this is essentially that what we are meant to become the access point, obviously, through the cap and password to log in to that you better get. Which is acting as a access point. Right. This is the only core difference with default that you'll find is there.

```
D:\MIUb\ESP32_CAM\Code\1.Lab2_CameraWebServer_Default\CameraWebServer\CameraWebServer.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
CameraWebServer_AP.ino CameraWebServer.ino
//initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
  s->set_vflip(s, 1);//flip it back
  s->set_brightness(s, 1);//up the blightness just a bit
  s->set_saturation(s, -2);//lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);

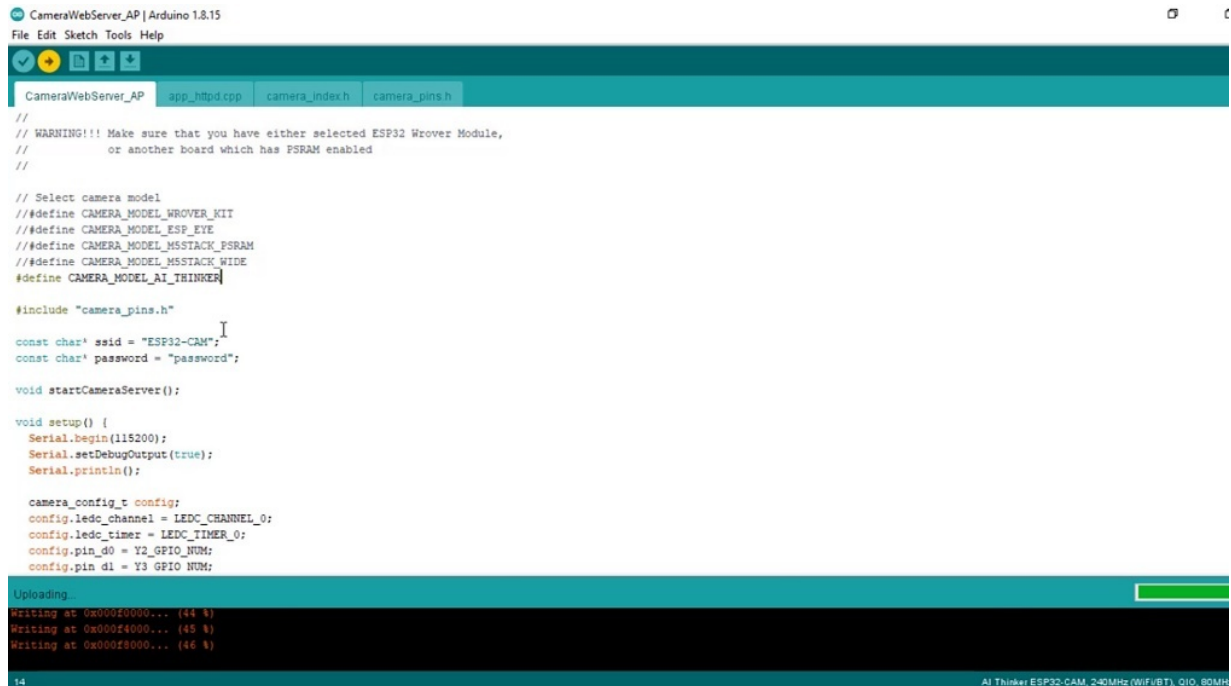
#if defined(CAMERA_MODEL_MSSTACK_WIDE)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif
WiFi.softAP(ssid, password);
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);

startCameraServer();

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}

C++ source file length: 2,883 lines: 111 Ln: 91 Col: 27 Sel: 24 | 1 Windows (CR LF) UTF-8
```

You are using the function called call, if you don't begin with the Saudi person here, instead of using Wi-Fi, it would be used by traders of. OK, which will create a Nexus point with Spotify necessarily and Facebook, so as soon as you get the IP I'm trying to put into the IP and I started to start everything. So dysplastic all so all the applications, everything we say, nothing changes.



```
CameraWebServer_AP | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer_AP app_httpd.cpp camera_index.h camera_pins.h

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
// or another board which has PSRAM enabled
//

// Select camera model
// #define CAMERA_MODEL_WROVER_KIT
// #define CAMERA_MODEL_ESP_EYE
// #define CAMERA_MODEL_MSSTACK_PSRAM
// #define CAMERA_MODEL_MSSTACK_WIDE
// #define CAMERA_MODEL_AI_THINKER
#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

const char* ssid = "ESP32-CAM";
const char* password = "password";

void startCameraServer();

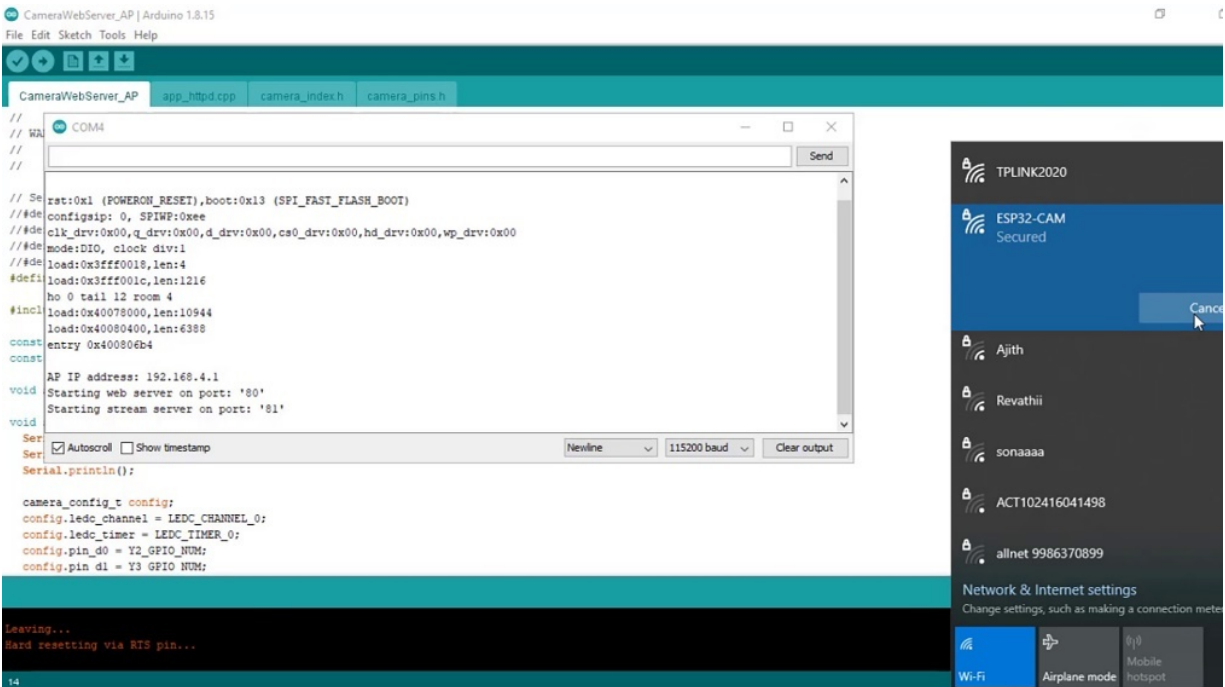
void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;

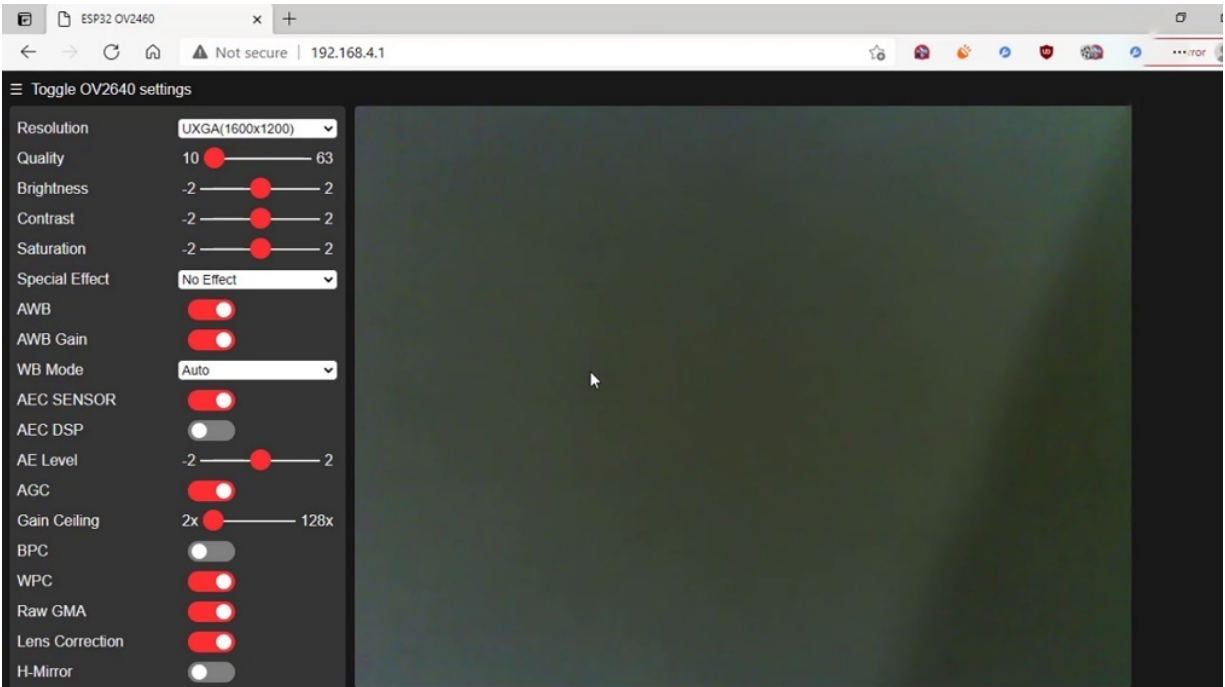
  Uploading
  Writing at 0x00020000... (44 %)
  Writing at 0x00040000... (45 %)
  Writing at 0x00020000... (46 %)

14 AI Thinker ESP32-CAM, 240MHz (WIFI/BT), Q10, 80MHz
```

OK. What about Flash? Please do know that you have to select the correct. A bull market. Restudying, says he, so not coming lectures, we will see also how to pull the static. This is at access point. We will again modify our default application to be functional within a static so that you don't need to again and again see this report and see the what is the allotted IP address further. You can try to listen to it away, bookmark your IP and you can just log into that. And hopefully a completed.



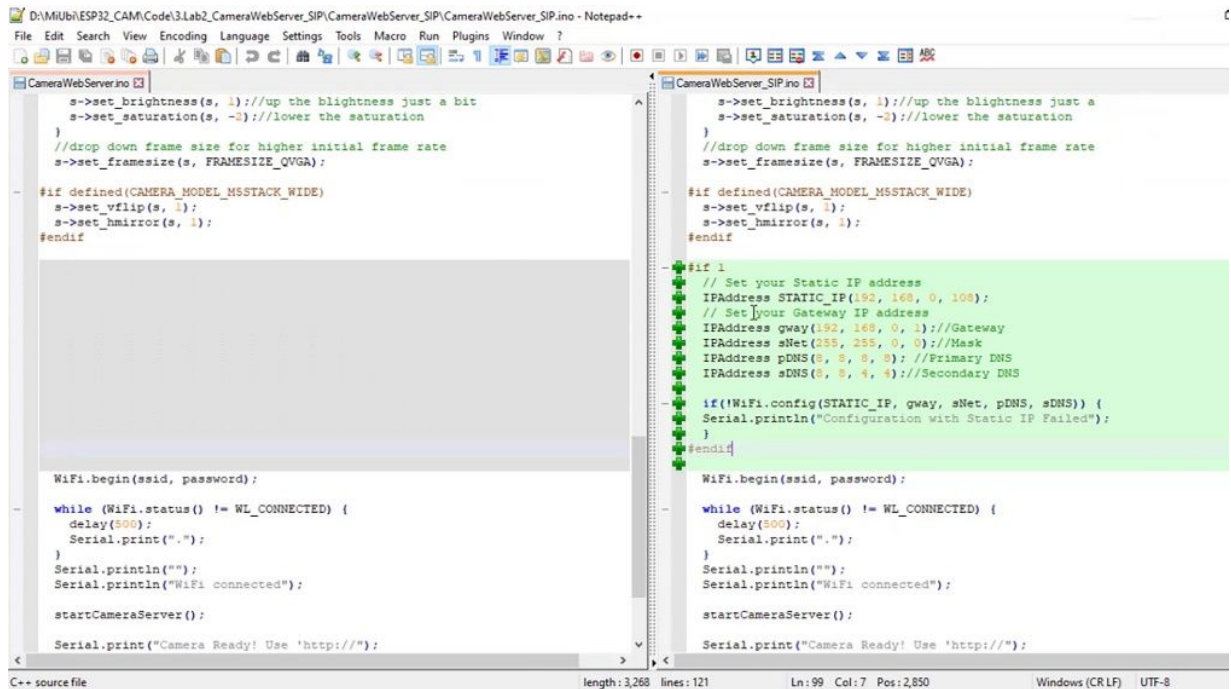
Yes, no, we'll just go to see monitor, we'll see what is printed that. Just press the reset button. Yeah, the program started and I think is displayed here, you have to go to the Wi-Fi connections. You see these people look like you just disconnect that. So you just kind of disconnect. You're ready for it. But out of connection and log in to. He's not asking the password. The password is password. Or whatever you wish. There's a ticking clock requirement. Something is happening here. So you see the IP address of the access point is 190 to 168 for one. OK, the Web server is streaming through the 81. So let us see by going to the this.



On the browser. What is happening on these little. For one, yes, right over this time stream. Higher resolution, Let me put it on the table. So the default settings are being posted on others, so do it again, ceiling. Which brings much clarity in my picture. Everything is fine, as usual, perfect, this is how we create the access point on the hospital again just by changing the three lines of code. Next, we will try to create a static IP, are assigned a static IP to the hospital again. And what we will be connecting, but really through the alternative, if you feel like there are plenty of latencies and and we don't require the Internet to be connected and connected in the way of access, point is much better and much faster.

ESP32-CAM AS STATION WITH STATIC IP

previously we have tried to create at this point, I mean, this current little waterloo's I just open the Web server default application and this is a Web server where I told you we will make a static IP.



```
CameraWebServer.ino
s->set_brightness(s, 1);//up the blightness just a bit
s->set_saturation(s, -2);//lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);
- #if defined(CAMERA_MODEL_MSSTACK_WIDE)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);
- while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");

CameraWebServer_SIP.ino
s->set_brightness(s, 1);//up the blightness just a
s->set_saturation(s, -2);//lower the saturation
}
//drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_QVGA);
- #if defined(CAMERA_MODEL_MSSTACK_WIDE)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif
- #if 1
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 100);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1);//Gateway
IPAddress sNet(255, 255, 0, 0);//Mask
IPAddress pDNS(8, 8, 8, 8); //Primary DNS
IPAddress sDNS(8, 8, 4, 4);//Secondary DNS
- if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) {
  Serial.println("Configuration with Static IP Failed");
}
- #endif
WiFi.begin(ssid, password);
- while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
```

So these are the only set of. Codes which are required to make your hospital began to have a static level, so very defined. So if you really feel there is not much difference, exactly the same. So we will create an optical address and we do assign a static IP 190 to 160 zero one zero eight and divide. And we also are seeing the gateway. And a mosque, subnet mask and primary interconnectedness. That's it, and then we go for confirmation and then rudeboy for begin with the smaller, should we actually make the IP to be static? In fact, as like you will do your TCP IP in your network setting and do your manual settings the similar way. This also. So these are like default values, what we usually do for networks, 190 to 168 01 one hour later. So for example, if you have three or four years petrochem, you can take this IP to one or did one or nine 110 like multiple IP games you can have and you can have a computer billing system of four or five or six candidates like that. That is also possible. That is very much required when you are connecting to the network. Most of the time that entertains to the IP.

```
CameraWebServer_AP | Arduino 1.8.15
File Edit Sketch Tools Help

CameraWebServer_AP app_httpd.cpp camera_index.h camera_pins.h
#if defined(CAMERA_MODEL_ESP_EYE)
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
  Serial.printf("Camera init failed with error 0x%x", err);
  return;
}

sensor_t * s = esp_camera_sensor_get();
// Set the required Frame Rate
s->set_framesize(s, FRAMESIZE_QVGA);

WiFi.softAP(ssid, password);
IPAddress IP = WiFi.softAPIP();
Serial.print("AP IP address: ");
Serial.println(IP);

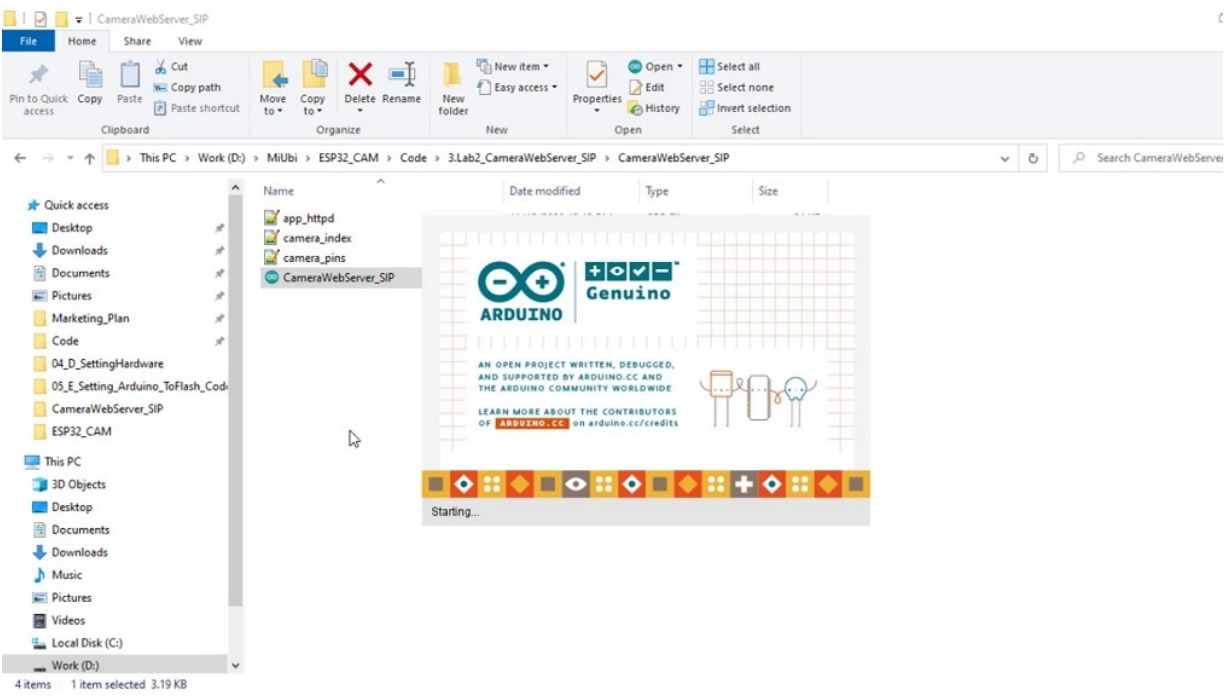
startCameraServer();
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}

Leaving...
Hard resetting via RTS pin...

14 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz
```

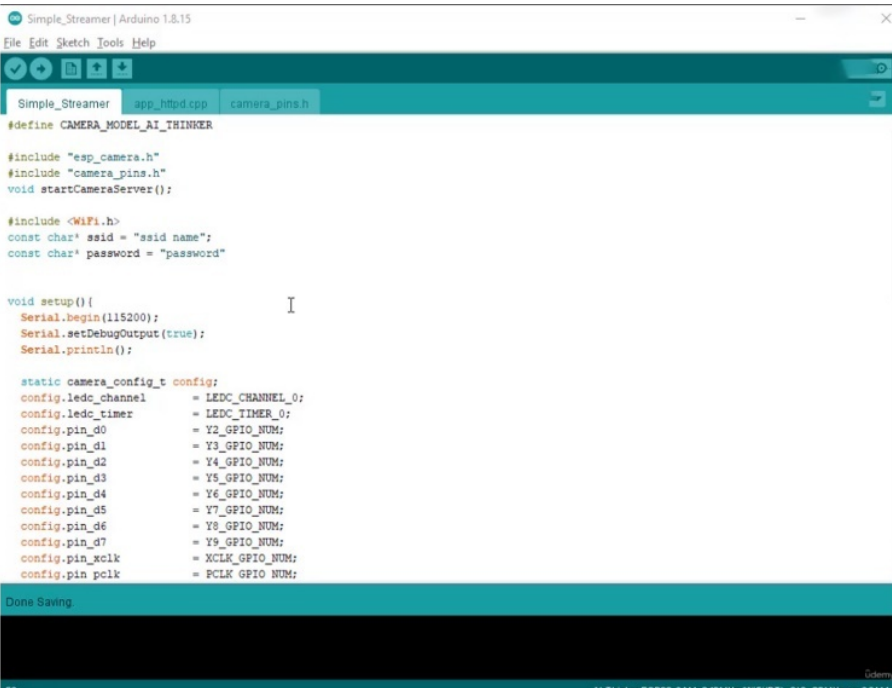
So here you can fix your own IP and quickly call easily, so let us flash this application with a small tune that we have done and let us see what happens. Yeah, I'm just thinking, oh, this is the access point application. Make it. Yeah, super. OK, now let me just flesh it. These, again, have application. OK, we're done programming, I just want to reset the program.



We are connected. Oh, if you remember, we have got a two year period of time and again came back, no, let me go for 100 to 160, zero to 100 Hirose cities. So just timestream, everything looks it no problem. Everything, exactly. Without any problem, everything looks like this, so we do it, so in the coming later, we'll try to write our own default. Streaming service. From scratch.

WRITING YOUR OWN STREAMING WEBSERVER

we are done with writing the application. Well, we will try to move on to a radio streaming service.



```
Simple_Streamer | Arduino 1.8.15
File Edit Sketch Tools Help
Simple_Streamer app_httpd.cpp camera_pins.h
#define CAMERA_MODEL_AI_THINKER

#include "esp_camera.h"
#include "camera_pins.h"
void startCameraServer();

#include <WiFi.h>
const char* ssid = "ssid name";
const char* password = "password"

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  static camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = FCLK_GPIO_NUM;
}

Done Saving
```

OK. So here you have tried to connect to the WiFi. All access point and also try to configure the center right, know what will be there and the key portion of the code, which is streaming because the request will be coming from the browser. Do you know from his little brother who and you have to service the request with the email stream and tell the browser that there is an image going to come and it has to get Gitari or equipped with a recording, the image and displaying for the user. Right.

```
Simple_Streamer | Arduino 1.8.15
File Edit Sketch Tools Help
Simple_Streamer app_httpd.cpp camera_pins.h
if(err != ESP_OK) {
  Serial.print("Camera Init Failed with error : %k",err);
  return;
}

IPAddress STATIC_IP(192,168,0,108);
IPAddress gway(192, 168,0,1);
IPAddress sNet(255,255,0,0);
IPAddress pDNS(8,8,8,8);
IPAddress sDNS(8,8,4,4);

if(! WiFi.config(STATIC_IP,gway,sNet,pDNS,sDNS)) {
  Serial.println("Configuration with Static IP Failed");
}

WiFi.being(ssid,password);

while(WiFi.status() != WL_CONNECTED){
  delay(500);
  Serial.print(".");
}

startCameraServer();

Serial.print("Wifi and Camera are ready to go")

}

void loop(){
Done Saving
```

So the very first thing will be we have to define the function of. Whatever name you find here, start Gamasutra like.

```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
Simple_Streamer app_httpd.cpp camera_pins.h
#include "esp_http_server.h"
#include "esp_camera.h"
#include "Arduinio.h"

static esp_err_t stream_handler(httpd_req_t *req){
  camera_fb_t * fb = NULL; // Initialize the Frame Buffer
  esp_err_t res = ESP_OK; // Initialize the result variable
  size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
  uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
  char * part_buf[64]; // Temporary buffer for string creation

  res = httpd_resp_set_type(req, "text/html");
}

startCameraServer(){
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();
  config.server_port = 80;
  httpd_uri_t stream_uri = {
    .uri = "/",
    .method = HTTP_GET,
    .handler = stream_handler,
    .user_ctx = NULL
  };
}

if(httpd_start(&stream_httpd,&config)== ESP_OK){
  httpd_register_uri_handler(stream_httpd,&stream_uri)
}
}

Done Saving
```

So this is on what will be the border, so it doesn't accept anything, written, anything. So it can start like this right before that, so we should have at least like the previous floor we have tried to include, you had to find similarly,

we have to include a few important things here which are essential and going forward, which is the history of it, because we are trying to create a hazard so as to decide what the main function calls which are present in the library will be declared in this survey. So this is required for the program. Look, and apart from that, I was included has become it. And I also included it all in order because I have to cross-reference the serial volume calls for debugging. OK. So what will be present in the stock over? So we as Americans, we have created a contact object, so we have to create a conflict object and also. As deeply underscored, the default conflict, so you don't need to worry too much here. There is a strategic function available from the with the local library, which will configure your server for default settings and in common. In his statement, as per standard, we have to use it for. Further communication between the client in the server, which is reported and just put it on. So we will continue the work and then I have to define whether what kind of request is going to come. Basically, the request, which is going to be. Coming, the user request. It is going to come is, you know, the request for stream, but we don't know whether it is extremely questionable because the user will try to or you yourself from the browser, you will be opening simply a. You are so what should happen in the world is open. So what is the prime out of the address is the we don't have any such policy, we just put it. OK, see, here we are trying to create the structure of the defining object. Define a declaration. What we are doing, the same thing, that is like refining it here on it. So what is the matter? As we have seen, the it has had its own structure called liberty. You are in this country. So we have to specify a few things. What is the type of, uh, request which is going to come? So it is going to be difficult, and when I get request comes to the server, there should be handled right. So what should be the handler doing? It has to pass the handle to some to some streamlined level. Right. So if you have seen the earlier walk through a the application, we have seen a registering of the different functions, like what happens when somebody asks for a string? What happens when somebody presses a button? So the different settings are there. So different something that. So here we are trying to see only a small portion of it. So there is one one murderer to be declared as. Right. So I just define the structure of that stability, this university or controlling the different variables is used internally, which is critical. So I have just made it and not care that much

about it now. So once this object is created, I'll be happy to. Start this server with the all these other configurations which are to be applied. And the configuration variable we constructed, it was. So if everything goes well and there is no other. And we are happy. If not, we have to tell that. There is a problem like. So if me OK. Then we have to go for registering the. So this is this dream handler. Heartstring request. So there is a function called. Canula. He's been registered. Hide. It's not just this dream, you have to get the complete structure of the. Mr. Lose part of that, so now we are registering for a request to get what is it in the be it has to call extreme. OK, no, no, the job is still not complete. We are missing the main part of the stream and the stream. We have to have this data stream that will stream that. So here is to level as the debate of will become and what are you going to do? So what are we going to do is will be required in the stream. And the stream is like it has to tell us it's somewhere like. What is the output of this streamlining function? So that he gets better. So Streamliner will accept only one thing. The request of. Or whatever the request which is coming, so I've created a request variable of these two entities separately. So initially to go forward, we require a few variables. One is a camera before variable and the variable looks like the result of the status of. And also the length of the Geopark before. And I pointed to before. And some interpreted labels subtitles. OK, so we have a frame of appointer, we have a thing, and we we want to see the end of the book and the pointer to hold the table before an alternate interministerial. Now, what is the next thing? The next thing is you have to send the response to the client, whichever is asking for you to send some data. So what do you think of the response? Not immediately the data, but you have to tell what type of is. So this is a function, call call response type. So for the request, you have to pass the request object, that means you have got a request for this request. I want to send something. So I want to say something like, oh, I just want to define or send some something that the request is the statement, then only the server will understand. So they can really understand what that is going to become is what is going to be the state.

```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

Simple_Streamer app_httpd.cpp camera_pins.h

#include "esp_http_server.h"
#include "esp_camera.h"
#include "Arduino.h"

#define PART_BOUNDARY "1234567890000000000000987654321" // A Unique Header to sperate payloads , which are JPEG Images in our case
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL; // Initialize the Frame Buffer
    esp_err_t res = ESP_OK; // Initialize the result variable
    size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
    uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
    char * part_buf[64]; // Temporory buffer for string creation

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
    while(true){
        fb = esp_camera_fb_get();
    }
}

startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    config.server_port = 80;
}
```

Done Saving.

Supposedly the problem that I have directly I don't know what is the stream content type? The stream content is a character that I want to define out it as a like that. It's a multi-party and it's having a boundary. It's not just a single part, it's a multi partnership. These are the regulations already by default. This is, um, we are working. So I'm justifying Padawan so bad boundaries. And I think we're really just a unique string which separates each pillar. What is the pillar here? The image, the content is with some bad boundaries. So what is bad boundary? So I have to define the part boundary as a unique string in making the bar boundary. I have made sure that we not repeat in Egypt extreme, which is this kind of specific and extreme. So you can also try with the other, because this is Mr. Proven mostly volume, so I use that to stream boundary is a different thing. We'll come to that.

```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

Simple_Streamer app_httpd.cpp camera_pins.h

#include "esp_http_server.h"
#include "esp_camera.h"
#include "Arduino.h"

#define PART_BOUNDARY "12345678900000000000000907654321" // A Unique Header to sperate payloads , which are JPEG Images in our case
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;

static esp_err_t stream_handler(httpd_req_t *req){
  camera_fb_t * fb = NULL; // Initialize the Frame Buffer
  esp_err_t res = ESP_OK; // Initialize the result variable
  size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
  uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
  char * part_buf[64]; // Temporary buffer for string creation

  res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
  if(res != ESP_OK){
    return res;
  }
  while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      res = ESP_FAIL;
    } else {
      _jpg_buf_len = fb->len; // Fill the Buffer length variable with size of compressed image
      _jpg_buf = fb->buf; // Pointer to the frame buffer is assigned to variable
    }
  }
}
```

Let us define the candidate first, one of the candidates is final, the next important step for us to keep on streaming images. And now the next step is continuously. We have to take notice. The continuous image is nothing. Whatever you think, you are trying to stream a video from the hospital to come over the voter. Right. So let us. Grab the frame from the camera, ESP8266, the score camera on the score, fearless quirked is the function which grabs the image. And gives them access to the FBI. What is a different offer which will have the access to the image? So I just want to check whether freimaurer is valid or not, if it is valid and will go forward, if it is not valid, I will just ask from you, OK, if this is valid. Then I'll go forward. It is not valid and I could do something that I have to just return the result and let this industry and at last I'll be returning from the function Google again.

```
esp32-camera/esp_camera.h at · x +
https://github.com/espressif/esp32-camera/blob/master/driver/include/esp_camera.h

114
115     int jpeg_quality;           /*!< Quality of JPEG output. 0-63 lower means higher quality */
116     size_t fb_count;           /*!< Number of frame buffers to be allocated. If more than one, then each frame will be a
117     camera_grab_mode_t grab_mode; /*!< When buffers should be filled */
118 } camera_config_t;
119
120 /**
121  * @brief Data structure of camera frame buffer
122  */
123 typedef struct {
124     uint8_t * buf;             /*!< Pointer to the pixel data */
125     size_t len;                /*!< Length of the buffer in bytes */
126     size_t width;              /*!< Width of the buffer in pixels */
127     size_t height;            /*!< Height of the buffer in pixels */
128     pixformat_t format;       /*!< Format of the pixel data */
129     struct timeval timestamp; /*!< Timestamp since boot of the first DMA buffer of the frame */
130 } camera_fb_t;
131
132 #define ESP_ERR_CAMERA_BASE 0x20000
133 #define ESP_ERR_CAMERA_NOT_DETECTED (ESP_ERR_CAMERA_BASE + 1)
134 #define ESP_ERR_CAMERA_FAILED_TO_SET_FRAME_SIZE (ESP_ERR_CAMERA_BASE + 2)
135 #define ESP_ERR_CAMERA_FAILED_TO_SET_OUT_FORMAT (ESP_ERR_CAMERA_BASE + 3)
136 #define ESP_ERR_CAMERA_NOT_SUPPORTED (ESP_ERR_CAMERA_BASE + 4)
137
138 /**
139  * @brief Initialize the camera driver
140  *
141  * @note call camera_probe before calling this function
142  *
143  * This function detects and configures camera over I2C interface,
144  * allocates framebuffer and DMA buffers,
145  * initializes parallel I2S input, and sets up DMA descriptors.
146  */
```

So the very first one we got is he has been a good candidate, so the frame buffer will have access to the land, the buffer appointed from the facility that we can hide and format all these things.

```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
Simple_Streamer app_httpd.cpp camera_pins.h

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL; // Initialize the Frame Buffer
    esp_err_t res = ESP_OK; // Initialize the result variable
    size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
    uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
    char * part_buf[64]; // Temporary buffer for string creation

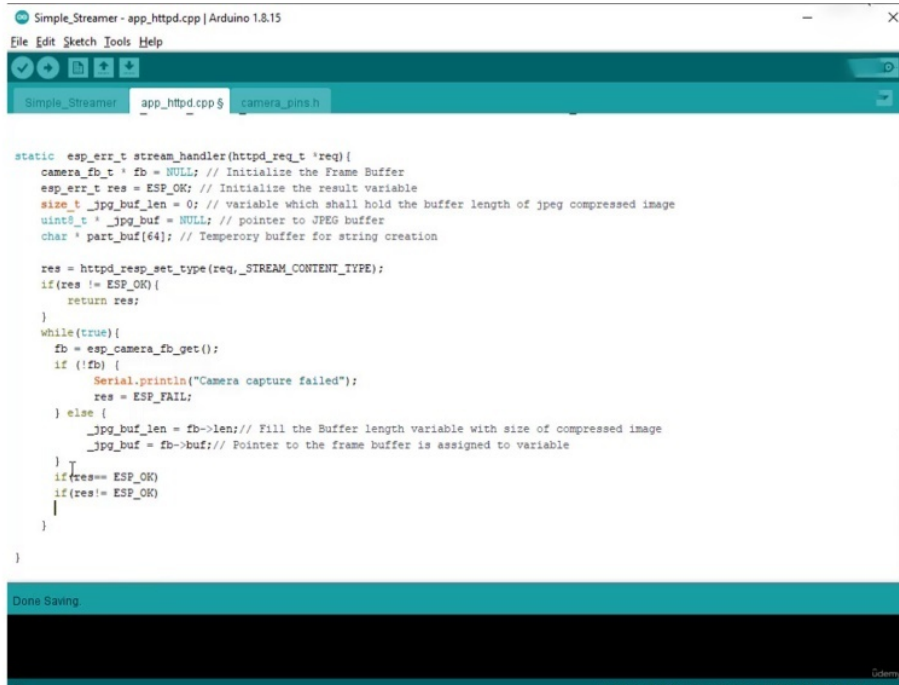
    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
    while(true){
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        } else {
            _jpg_buf_len = fb->len; // Fill the Buffer length variable with size of compressed image
            _jpg_buf = fb->buf; // Pointer to the frame buffer is assigned to variable
        }
    }
}

startCameraServer(){
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
}

Done Saving
```

So this will give me access to that I. So that I was able to get into the it and also the buffet point, so I've just given it to the little girl and so make sure

that the dipping buffer in there that they have declared it should do the same thing, whatever is different inside. A little other important thing to send in it, how we be sending it. So we have to tell the browser that I'm going to send a kind of stream. Right. I have already told that it's a multipotent, but I have to tell something else also so far, which is the same style of function call I will try to like if suppose if there is a result from the results of the previous function, call is okay.



```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

Simple_Streamer app_httpd.cpp camera_pins.h

static esp_err_t stream_handler(httpd_req_t *req){
  camera_fb_t * fb = NULL; // Initialize the Frame Buffer
  esp_err_t res = ESP_OK; // Initialize the result variable
  size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
  uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
  char * part_buf[64]; // Temporary buffer for string creation

  res = httpd_resp_set_type(req, STREAM_CONTENT_TYPE);
  if(res != ESP_OK){
    return res;
  }
  while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
      Serial.println("Camera capture failed");
      res = ESP_FAIL;
    } else {
      _jpg_buf_len = fb->len; // Fill the Buffer length variable with size of compressed image
      _jpg_buf = fb->buf; // Pointer to the frame buffer is assigned to variable
    }
    if(res== ESP_OK)
      if(res!= ESP_OK)
      |
  }
}
```

Done Saving

That means it should not be finished. Well I have to get up. So let me go right now. So suppose the result is. Nautical rules. I'm expecting it to be OK. That means everything is OK if it is not OK. I have to break the loop. If it is OK, I will move forward.


```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

Simple_Streamer app_httpd.cpp camera_pins.h
uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
char * part_buf[64]; // Temporary buffer for string creation

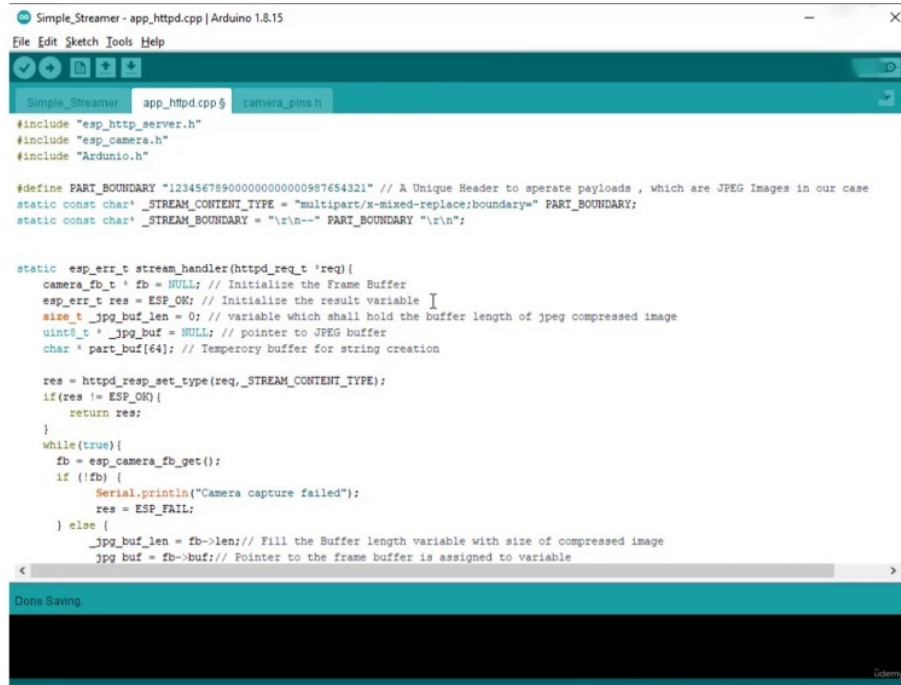
res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
if(res != ESP_OK){
    return res;
}
while(true){
    fb = esp_camera_fb_get();
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    } else {
        _jpg_buf_len = fb->len; // Fill the Buffer length variable with size of compressed image
        _jpg_buf = fb->buf; // Pointer to the frame buffer is assigned to variable
    }
    if(res== ESP_OK){
        size_t hlen = sprintf((char *)part_buf, 64, _STREAM_PART, _jpg_buf_len);
        res = httpd_resp_send_chunk(req, (const char*)part_buf, hlen);
    }

    if(res== ESP_OK){
        res = httpd_resp_send_chunk(req, (const char *)_jpg_buf, _jpg_buf_len); // Send the JPEG Image
    }

    if(res== ESP_OK){
        res = httpd_resp_send_chunk(req, _STREAM_BOUNDARY, strlen(_STREAM_BOUNDARY)); // Separate each JPEG Image with a Unique Boundary
    }
}
```

So three things we have to do here. One is we have to tell the client, the browser, which is asking for a buffer from us. The content is the image. And decoded. And the boundary. We are in fact, we have three things I want to make sure are not in the loop. So the first thing is we have to tell the browser that how much amount of data is going to come? What is the type of data are you that the type of mainstream or the like within the boundary in order to get the length of the buffer? The function call we have here is the equivalent. Right. Which have already accessed using. I've been in Scotland, so there is a function called here called, uh. Aspirin, which will try to contain it. And this settlement is to be sent to the requesting client. So I'm just trying to check the result again from the wells and request to see discontinuance taking off. A result is because we are trying to do the communication. At any point of time, the communication may fail. So I just want to make sure I know what the result will be happy, just not simply sending wrong information. So been. Response from. So here we are trying to create a string, right, and also get Dylan, the string we just created, which will be stolen. So we have to pass the part of here. So this is part of which is a temporary. So I'm just trying to send the Depardieu's and the length of it. So the length of this particular string. String, which is created by the Bardwell's, should be typecasts. And to character this, this, that. And then we have to say the. Then after once you've sent home and your mother is

willing to come, then I tell her. Actual bottom, which is the GOP image to. So the same function called. Then I'm going to hand it over. So the debate before this debate is by this point and is holding it this way. OK. And again, we're taking four little ones, Dylan, and also the buffer. For the next part, we have to separate the upcoming back of that bombing bill. With a boundary that is nothing but extremely so, but with some kind of special characters included, that's it.



```
Simple_Streamer - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

Simple_Streamer app_httpd.cpp camera_pins.h

#include "esp_http_server.h"
#include "esp_camera.h"
#include "Arduino.h"

#define PART_BOUNDARY "1234567890000000000000987654321" // A Unique Header to sperate payloads , which are JPEG Images in our case
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";

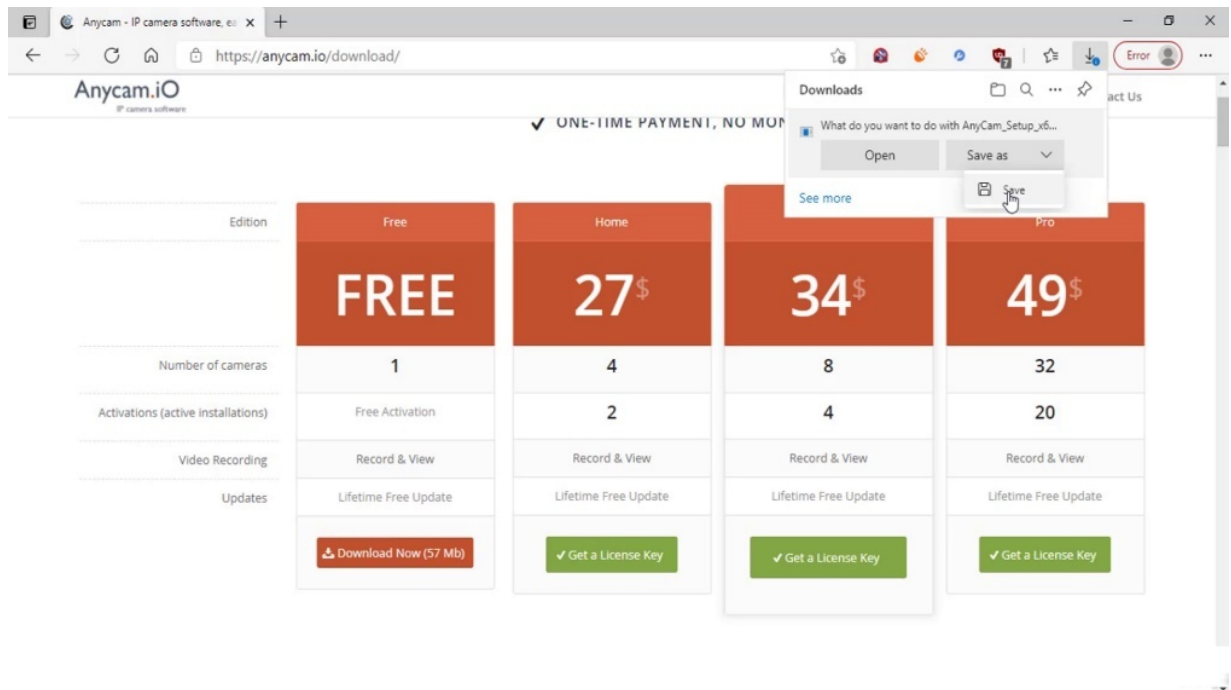
static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL; // Initialize the Frame Buffer
    esp_err_t res = ESP_OK; // Initialize the result variable
    size_t _jpg_buf_len = 0; // variable which shall hold the buffer length of jpeg compressed image
    uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
    char * part_buf[64]; // Temporary buffer for string creation

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
    if(res != ESP_OK){
        return res;
    }
    while(true){
        fb = esp_camera_fb_get();
        if (!fb) {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        } else {
            _jpg_buf_len = fb->len; // Fill the Buffer length variable with size of compressed image
            _jpg_buf = fb->buf; // Pointer to the frame buffer is assigned to variable
        }
    }
}
```

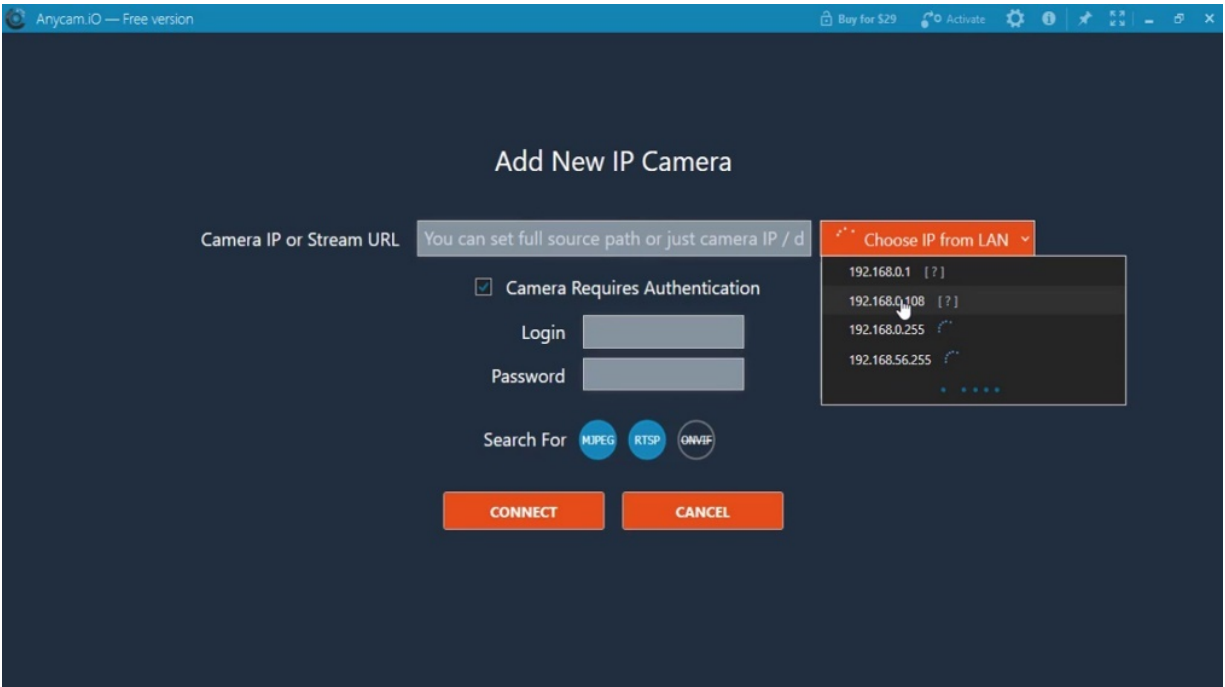
That we are just trying to include a new name so that it understands that Paderborn refundable Uihlein is nothing but. It's a new image stream. So this is much, much quite simplistic than the default application, what you seen, so I will again be able to do so. In the meanwhile, I will set up an hour loop in this attempt to configure the civil war, then the camera, then the Wi-Fi and then the strategy server. When you started this whole internal server, you are trying to create a server and registering the handler, so whenever a request comes from the browser, it will actually call stream handler. So what is the handler function? It keeps on continuously since the image request for it to be displayed.

USING ANYCAMIO

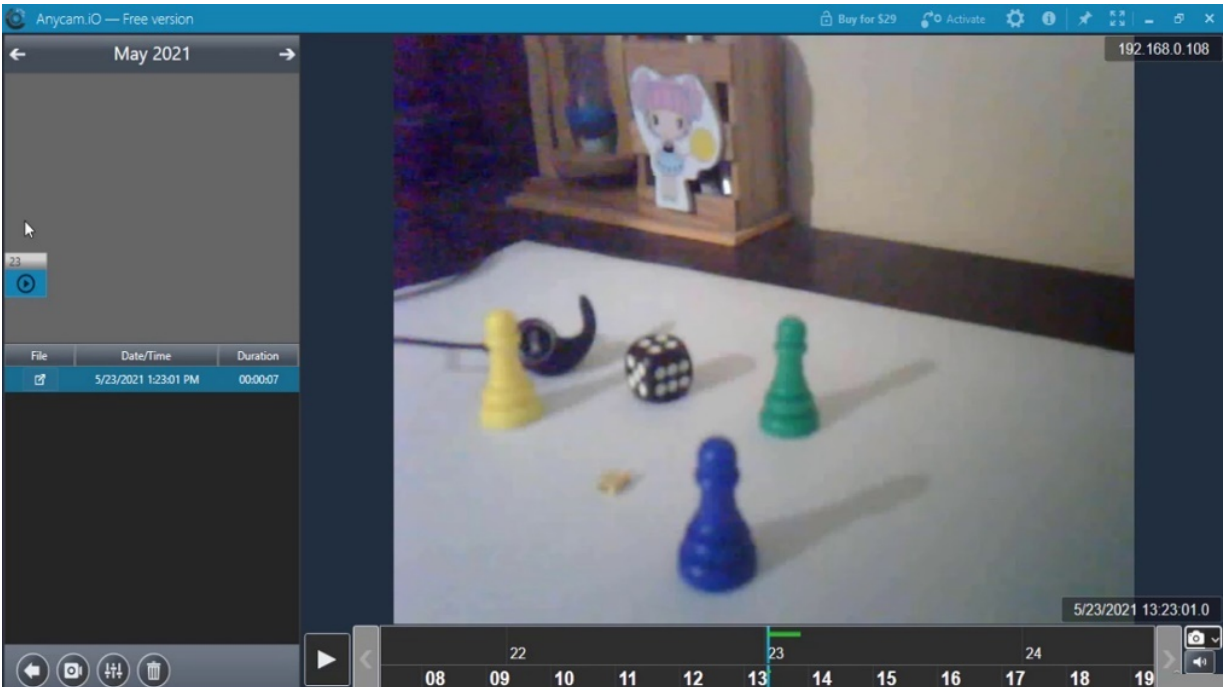
we'll see how to create a camcorder using any kind of cam,



you can just Google any camera and go to the download section and go for downloading it. I have only a version of it. I'm not able to. I'm not going to download it.



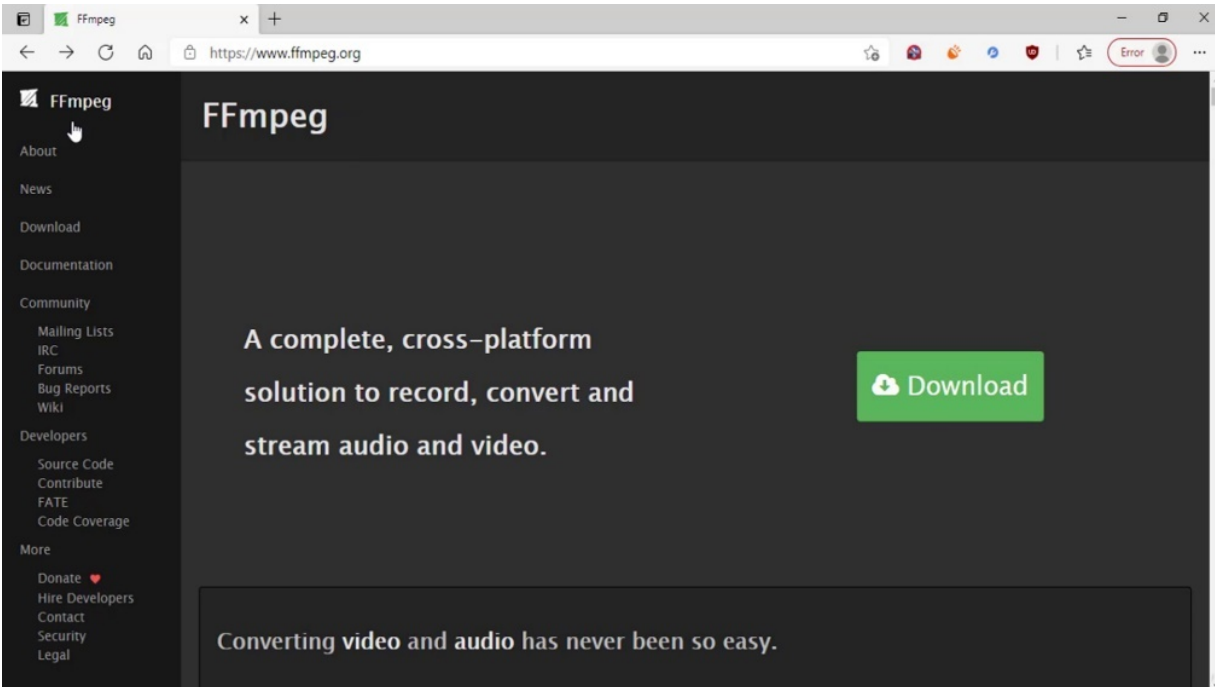
So there are multiple plans that a user free plan and just after installing and go to your start menu and tap any I it opens up and then use your sections to hide a camera. If you are connected to the network, you will find the IP of the camera automatically. That's the great thing. You have to just take the camera because authentication because you don't have username and password and click on to. Straightaway, you can see the camera coming from the IP Street on the right bottom, you have two sections to your recordings and you start recording.



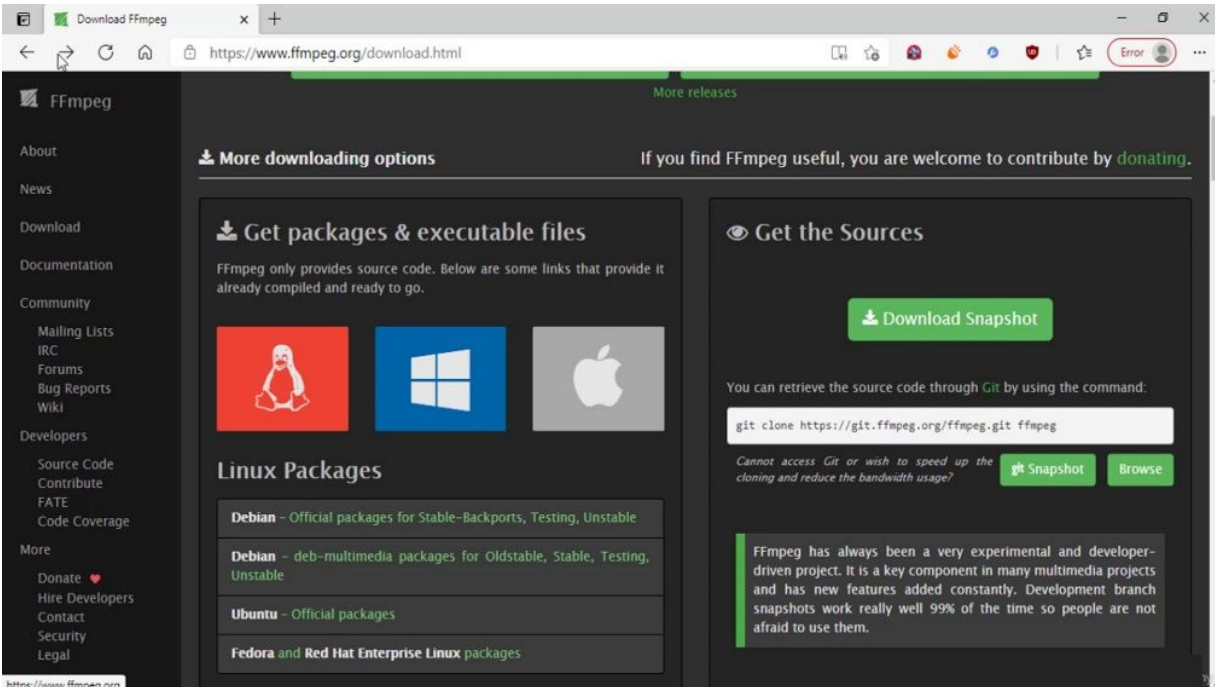
Just try to start recording. You can do capture also. If you just click on a record, I'm just recording for a few seconds, let me just go to open recordings and you can just see the recorded file in a different format. That's great. Apart from this, we also have a calendar in the end. And again, my little portion, if I go back to the enigma that I can see your recordings so that I can finally zoom of particular video portion also, which is a great tool. And we trying something, they can see what is happening if you just play button. So we will do whatever is recorded and you can just pause. It is a very good, wonderful tone we can stream as people began using this. Hope this helps.

USING FFMPEG

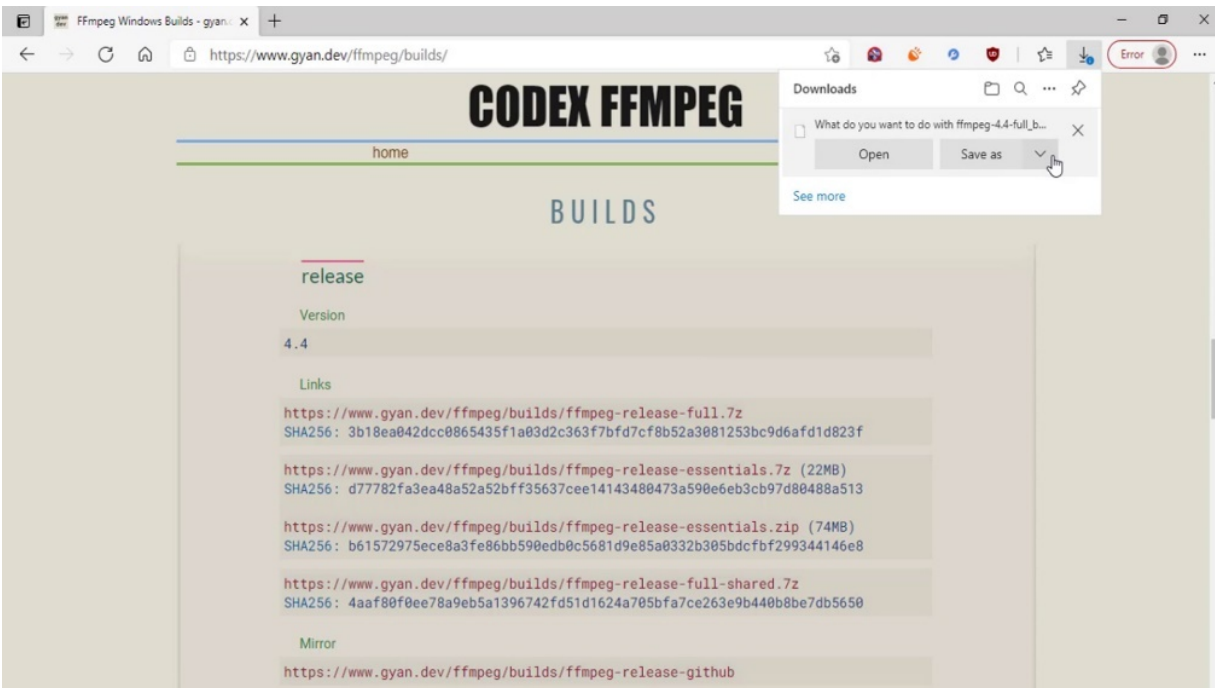
try to create a camcorder using something, and you can search for something under the.



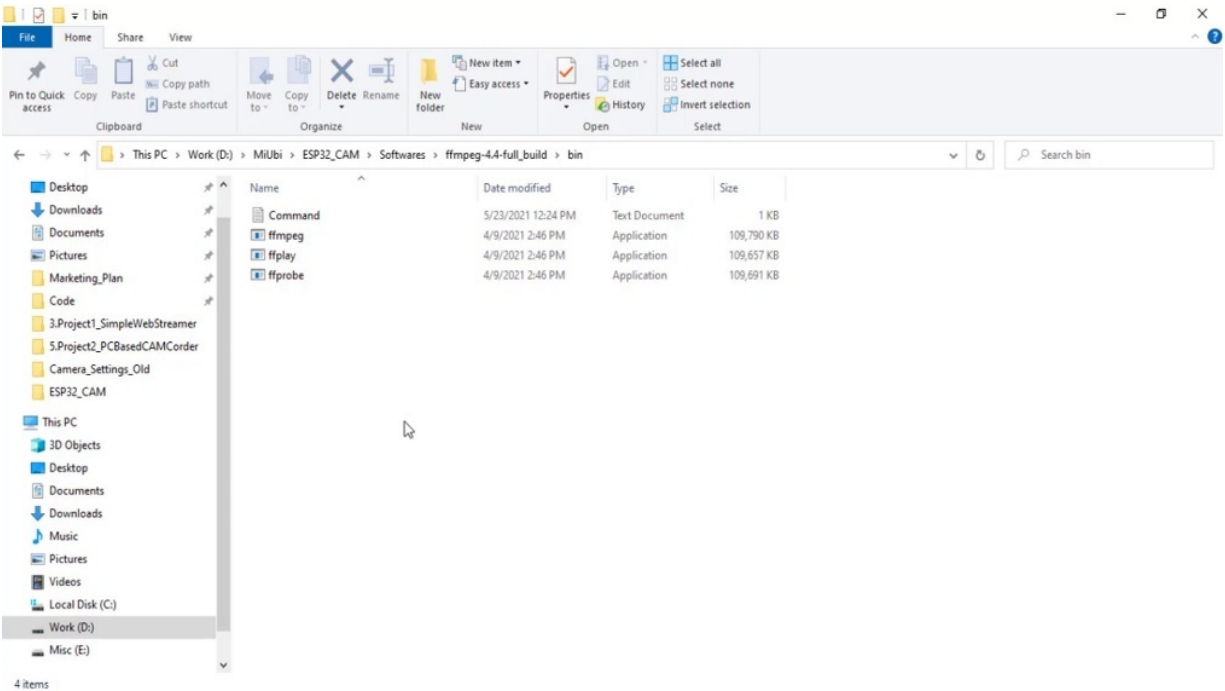
You on the first page and go to download. And there is you'll find this kind of page where you can see all the options for the backers of Linux, Windows and Apple, like you can just click on Windows and there is only Windows means from this particular site. So if you go to that, you'll see something like this.



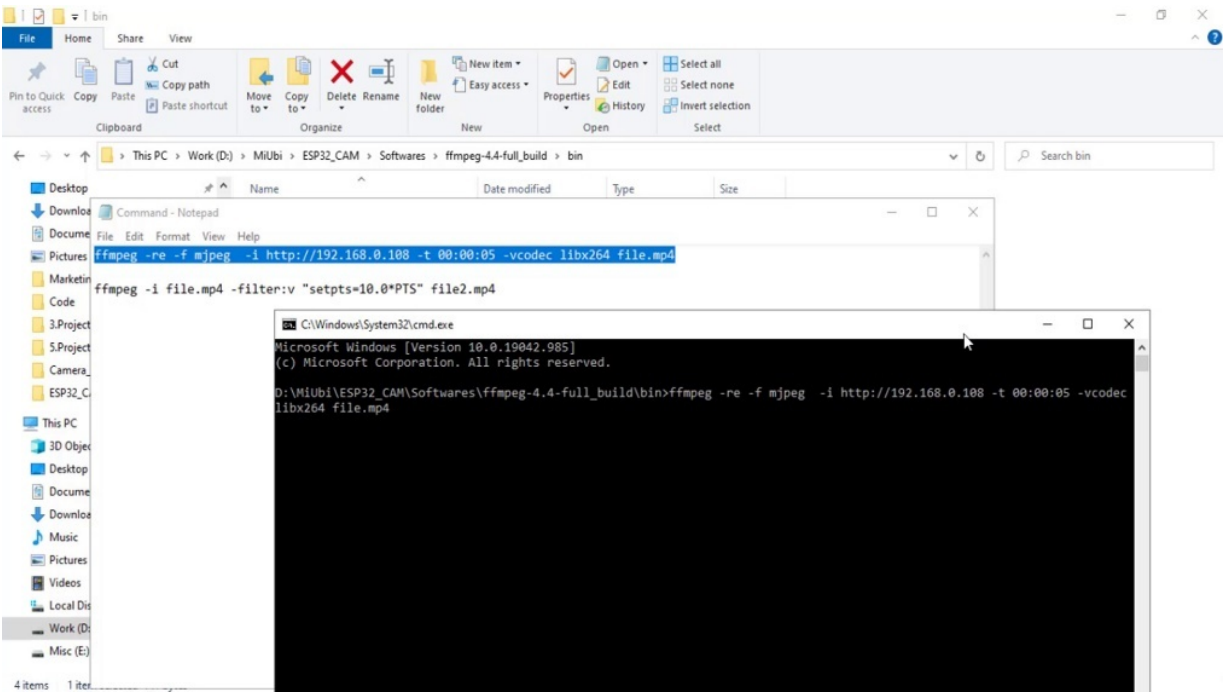
And not as a fan base, so in this country is. Invalid's, you just don't know. Look, this is basically Cenziper and the some of the companies define you can just click on it and download.



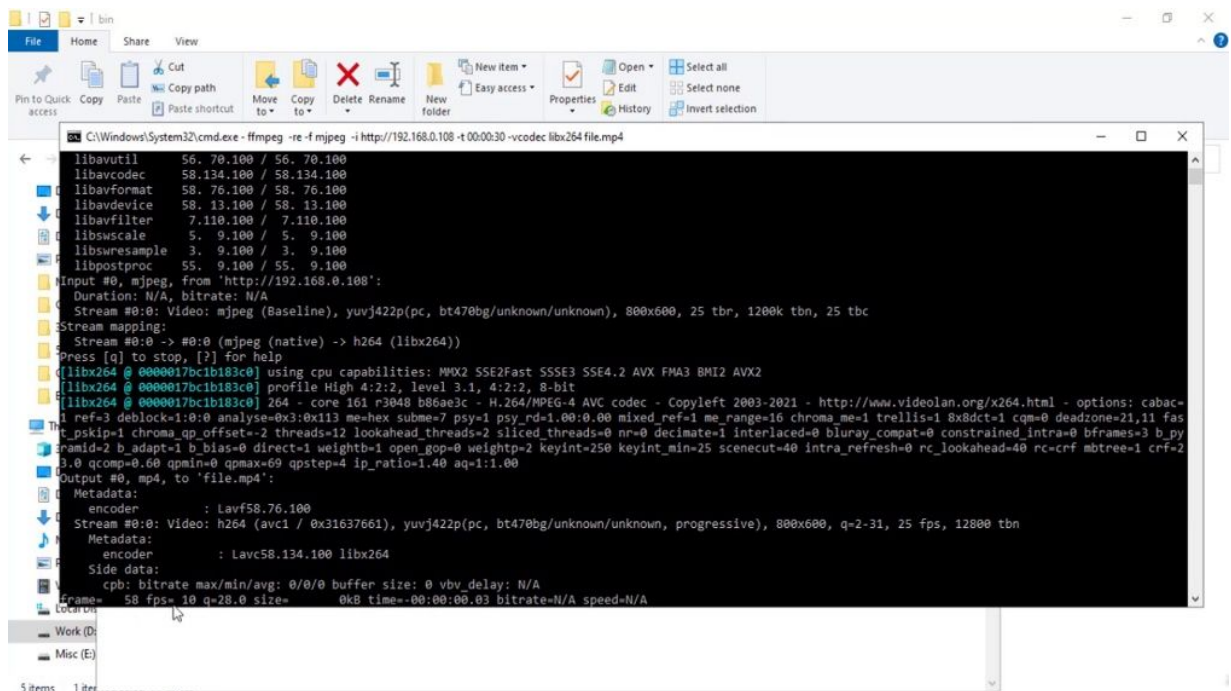
Look, I have already downloaded one. I just canceled it. So I think I'm good on those. This is the one you can just go to seven zip. And they started.



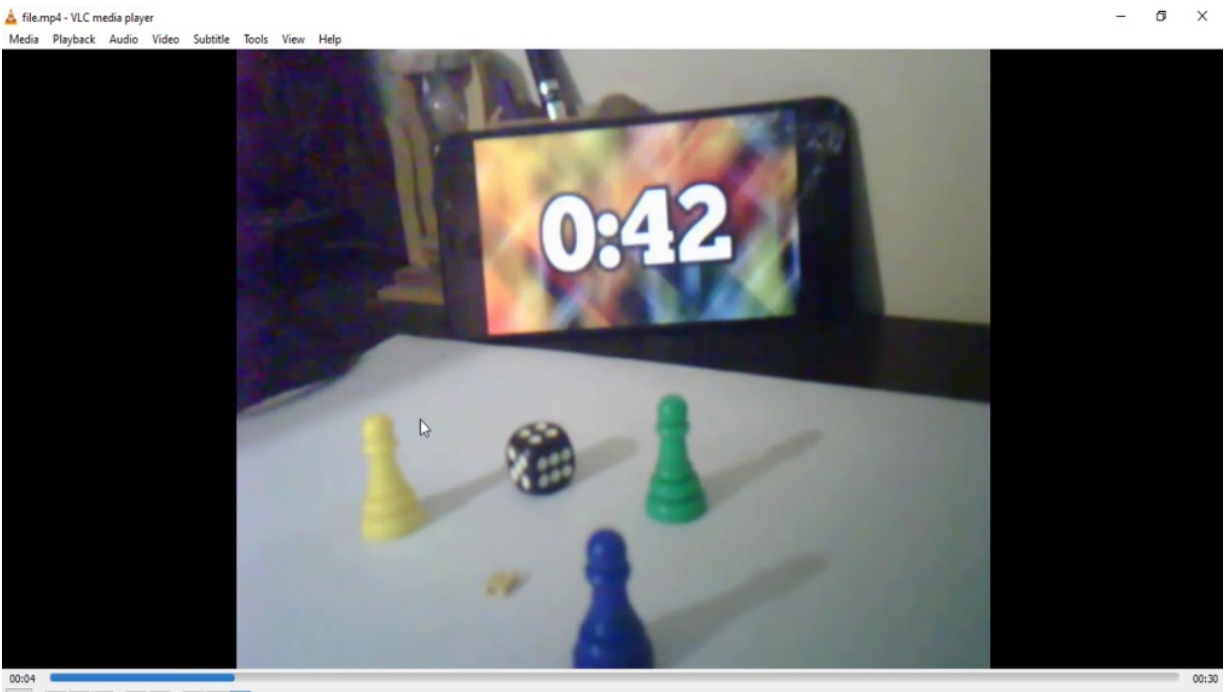
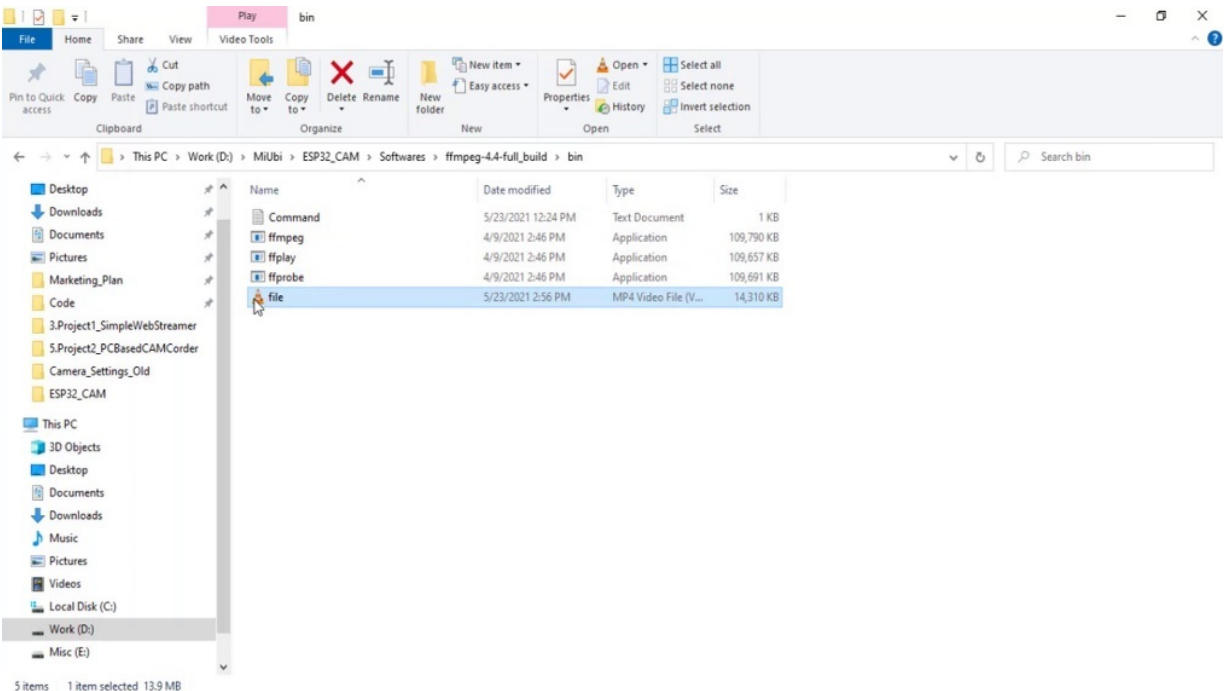
OK, here we go into the bin, this is where you will find this bit right here in this explore this tape and click play catch so you will open the command window to and from the current, but like so acetyl and Fagih, the software will help you to record.



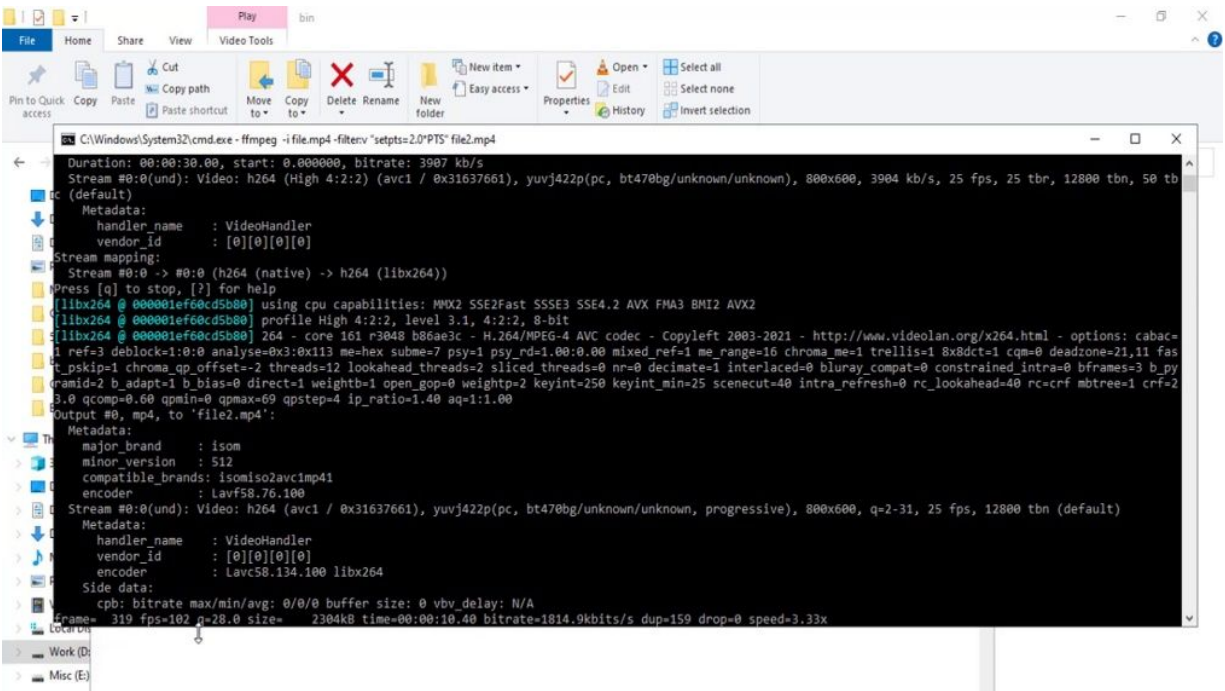
So I have carried out to commence and Cooperman's the very first commanders regarding in video, which is coming from the IP camera. So I just finished it. Now what is it doing this? So you have the IP camera link lined up to 116 on it. You are giving that IP stream of input and saying that the streams and get the stream and you could come to the Peterle and you want to record for a second time, not later on continuously. I want to record for five seconds. So let me just make it to some 30 seconds. So essentially, I want to record that sentence. OK, and when I competed, when the streams, which I am recording, I wanted to restore order and before coming farming, the decompression became pointless for as one disease, for all the live export exporters, the competition for three years before that. I just want to switch on one thing on my screen.



Yeah. So once you click on Enter, this is what will happen. So there are frames coming up, the frames per second, essentially around in and frames per second is the rate at which the frames are coming from is BITRATE again. So it has completed the time of the. Seconds, four seconds.



This is the file we recorded. Yeah, I put it in one of the alternate, so if you see that it's actually moving very fast, right? The time it is going very fast.



So it's completed, but we are increasing the number of frames. It is through military means in. And frames for frames will increase so you can take more time to play. So let me see it. This is not perfect recipe for stagnation, but still it's OK. So once you record with the hospital, you can you can correctly reduce them using this. If you can slow down the video much more by making a ten point five, fifteen point, if you want to see the much detail of the fast moving things is available to. Right. So with this, you can actually record a number of hours of video. So for 30 seconds of video, it was condemning me around. The file was around 14. So you can calculate and see how much side will be taken for you and you're going to accordingly. So this is how we use the big hope. This helps you to use the hospital.

INTRODUCTION TO EEPROM AND MEMORY CARDS ON ESP32-CAM

we will see how to access the improvement as the memory card, which is present on the bottom, is actually present inside as the memory card access

is given out. So I will see a short comparison between it from memory cards or the multimedia cards. And formatting them and the spectacted.

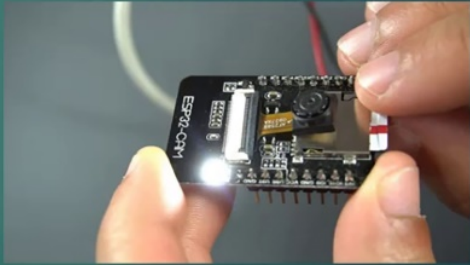
So if you see the comparison between the prime memory card, he promises like a political erasable programmable read-only memory in contrast with random access memory ramp's, this is an unrelated memory, but certainly as was. Whereas his demons, he is sitting on the module of the hospital camp outside the associate interface, you have been said from America on onto the modern. So, like memory cards are available, it's basically micro declared secure digital multimedia card. Are like different variants are available, maybe anyone would work, so we have to just format it with the greatest.

EEPROM & SD-MMC on ESP32-CAM

EEPROM

- ▶ The EEPROM is an internal memory of the ESP32 microcontroller that allows to keep in memory data after restarting the board. When working with microcontrollers, it is interesting to keep data in memory especially when the card turns off whether it is wanted or not, as in the case of a loss of electrical power.

SD-MMC



So this is the place where you should insert the America, you can just push it, it's like push it once it is pushed it once if you push it so. This is there is no need to interpret or interpret anything about the promise president entirely associate.

Going Ahead

Adapter



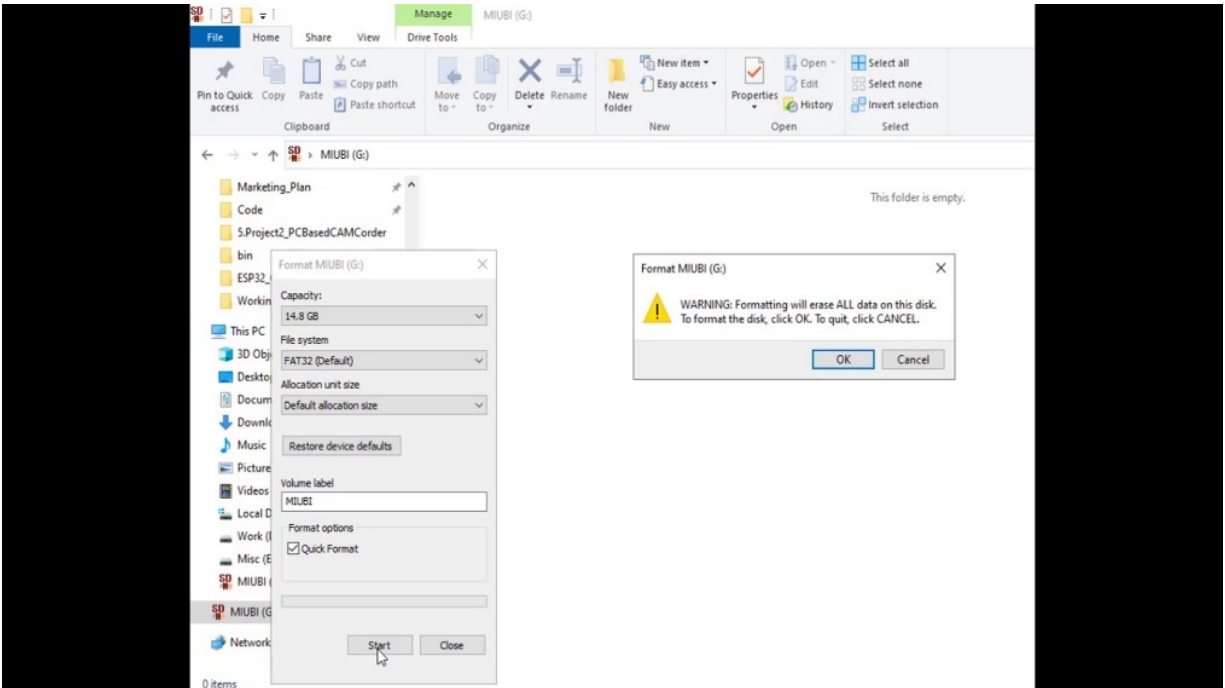
USB Reader



So once you have insight into this in order to review the files back or to see to the record, you should have you got a letter which can mean to tell them or you related for.

PROGRAMMING ESP32-CAM TO ACCESS EEPROM AND MMC

So this is my sticker, I just I click on it to format.



Totally false. It should be. OK. Is it? Then for Margaret, this is one job it - has done. I just want to do it first, then we'll start to go to court. So, as I've said, we have to deal with filesystems.


```
sketch_may23b | Arduino 1.8.15
File Edit Sketch Tools Help

sketch_may23b $
#include "FS.h"
#include "SD_MMC.h"
#include <EEPROM.h>
#define EEPROM_SIZE 1
int FileNumber = 0;
void setup() {
  // put your setup code here, to run once:
  //Necessary Serial Function call to be done
  Serial.begin(115200); //Set Baud Rate
  Serial.setDebugOutput(true); // Let Debug Messages Appear
  Serial.println(); //Just Prints a NewLine

  if(!SD_MMC.begin())
  {
    Serial.println("SD Mounting Failed");
  }

  unit8_t cardType = SD_MMC.cardType();
  if(cardType == CARD_NONE)
  {
    Serial.println("No Card Attached");
  }

}

void loop() {
  // put your main code here, to run repeatedly:
}

Done Saving
24 AI Thinker ESP32-CAM 240MHz (WiFi/BT), QIO, 80MHz on COM4
```

So our leader has already implemented the file system, so we are just trying to access it and also just the memory card is also implemented in our own accord. So I'm just trying to include the function calls, declarations, which will be helpful for me. Apart from that, I also include the routines, which will be responsible for people access. So apart from that, I also define sites for people. I mean, one might also want to define a variable global variable. This is a number which is initialized for the first time being zero. So in setup, as usual, I want to see the debug messages and associated with messages. So, you know, those serializable calls as they code NMC as like this. So if there is a failure in the value, so we have to tell our serial port. That that is something from IRS demanding food. After that, so I just want to after Cardamone, things happen, so I want to inquire about the car or just to check whether the car is mounted properly or not. It should be even a tape type. It will be accessed from as during the court NMC dot com type. And so if the car is having a problem. Like, if it is cardinal, then that is a problem. So up to here, we can do inquiring about ticker-Tape and also accessing the card and mounting Montegut, but creating access, obtaining all access permission to read or write or open things onto the sticker.

```
sketch_may23b | Arduino 1.8.15
File Edit Sketch Tools Help

sketch_may23b
-----
{
  Serial.println("No Card Attached");
}

EEPROM.begin(EEPROM_SIZE);
FileNumber= EEPROM.read(0) + 1;

String path = "/text"+String(FileNumber)+".txt"
Serial.printf("New file name: %s\n", path.c_str());

File file = SD_MMC.open(path.c_str(), FILE_WRITE);
if(!file){//File Creation or Access Error
  Serial.println("Failed to open file in writing mode");
}
else{
  file.println("This is Text File");
  Serial.printf("Saved file to path: %s\n", path.c_str());//Path of File Saved
  EEPROM.write(0,FileNumber);
  EEPROM.commit();
}
file.close();
delay(2000);
}

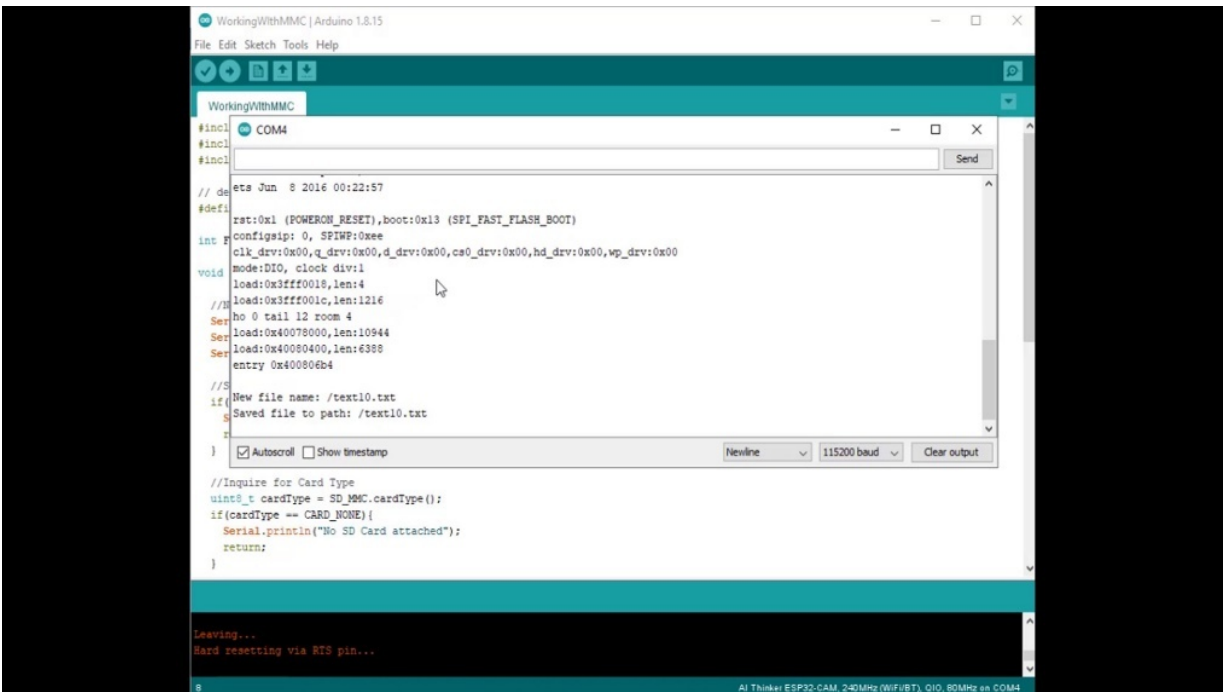
void loop() {
  // put your main code here, to run repeatedly:
}

Done Saving

AI Thinker ESP32-CAM 240MHz (WiFi/BT) QIO, 80MHz on COM4
```

In fact, trying to get a handle on it now will come to you from watching. Just like the current system we have from the beginning. So we will begin as like we begin with a report for the year, we begin from with the signs we want to allocate in our case. This is just one way. What I just wanted to say is I want to create a dummy file onto political. And usually the final bill will be zero. So. I was like, I have already defined it, so here it is, because we have nothing in. So I'm just. So it starts at 1:00 and keeps on increasing because next time I'm going to like after this is complete, my final creation is complete, then I'll just incrementally if no. And I will ride onto the promise it is going to happen before that, I want to. Create a path pathway through which I want to store the file on American. So I want to create a string, but. I want to create a text file. OK, I'll be fine, no. So there will be a string. Whose name is the final? That means. Text one, two to three and the extinction of the file will be next. So this is the final. That they want to create. Once that is done. Then I want to again know we have to create the so as like a normal index or other file systems, where are the C programming file operations? So if you try to open a file indictment, you all know that it would create the file. If it is already not the same happens here. So I'm just trying to open a file which is which should be present and the part that file. Right operation. OK, this will be different. And also we will

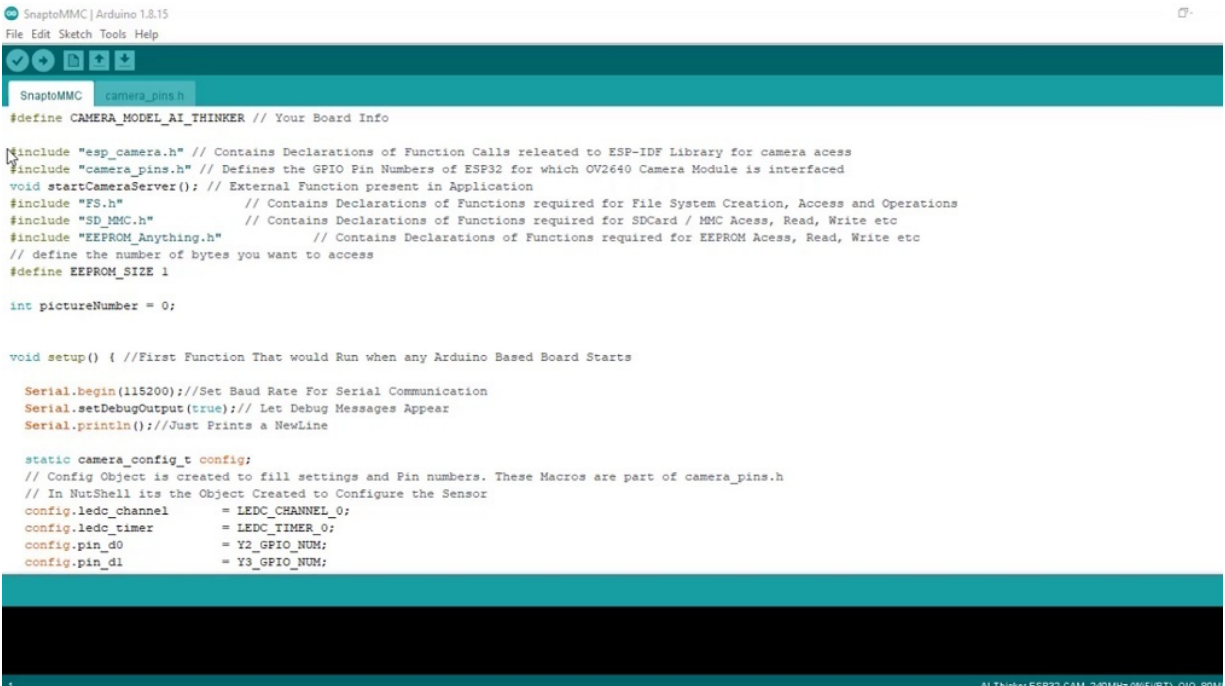
take the file handler, but if there is any error in the file handler, then there is a problem in opening. If there is no problem in opening the file, then I am free to write some letter here. So to write to the file and just access the file handler, not print that little printer line into the file. I wish to put into the big message that we have saved, if I like. Then after here, we will see that pushing what I was earlier speaking, I want to update defining Lepidopterist that the findings should be updated, admits it should not again create the same problem. So this is a current running fine. No, no, when I come for the first time next time, I have to update it, right. So then I come for the next time. What I would be doing if I No. Sorry, this assignment, so it is not over, we are doing file one. OK, I have already told one that you will read one again and it comes back one plus one will become the new phone number will come. It will actually apply the changes. So once the FedExes is complete, we usually have the, uh, relinquishing the date allocated to the handler and some delay of some two seconds would be sufficient to do the closing operation. Also, is the correct US operations to smoothly complete. So what we're actually doing here is practicing the prop, creating a initial value or number one, that and when we come back, again, reading it and creating a file through the text. And so OK now and see if there is any problem.



So we are done with the gold. So let me go to tools and go to see the monitor. And I will try to see for the comfort. Yeah, that's right. Now, I have written something, text eight or so, let me start again and you will see plus one fixed line of text. Then again, next Wednesday and 4th, we will further use this memory card to score our images in the coming tempters.

CREATING TIME LAPSE VIDEO USING ESP32-CAM WITH THE HELP OF SD-MMC

we'll try to take snapshots to the memory card. So the court is the same as being this. There is nothing will change the system. Access Economics is everything is there. But instead of writing some garbage text.



```
SnaptoMMC | Arduino 1.8.15
File Edit Sketch Tools Help

SnaptoMMC camera_pins.h
#define CAMERA_MODEL_AI_THINKER // Your Board Info

#include "esp_camera.h" // Contains Declarations of Function Calls related to ESP-IDF Library for camera access
#include "camera_pins.h" // Defines the GPIO Pin Numbers of ESP32 for which OV2640 Camera Module is interfaced
void startCameraServer(); // External Function present in Application
#include "FS.h" // Contains Declarations of Functions required for File System Creation, Access and Operations
#include "SD_MMC.h" // Contains Declarations of Functions required for SDCard / MMC Access, Read, Write etc
#include "EEPROMAnything.h" // Contains Declarations of Functions required for EEPROM Access, Read, Write etc
// define the number of bytes you want to access
#define EEPROM_SIZE 1

int pictureNumber = 0;

void setup() { //First Function That would Run when any Arduino Based Board Starts

  Serial.begin(115200); //Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true); // Let Debug Messages Appear
  Serial.println(); //Just Prints a NewLine

  static camera_config_t config;
  // Config Object is created to fill settings and Pin numbers. These Macros are part of camera_pins.h
  // In NutShell its the Object Created to Configure the Sensor
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
```

To the end, what we will do is I have as late in the previous lab, I have created a camera object and then find all the settings and then also tweak

the settings.



```
SnapttoMMC | Arduino 1.8.15
File Edit Sketch Tools Help

SnapttoMMC camera_pins.h

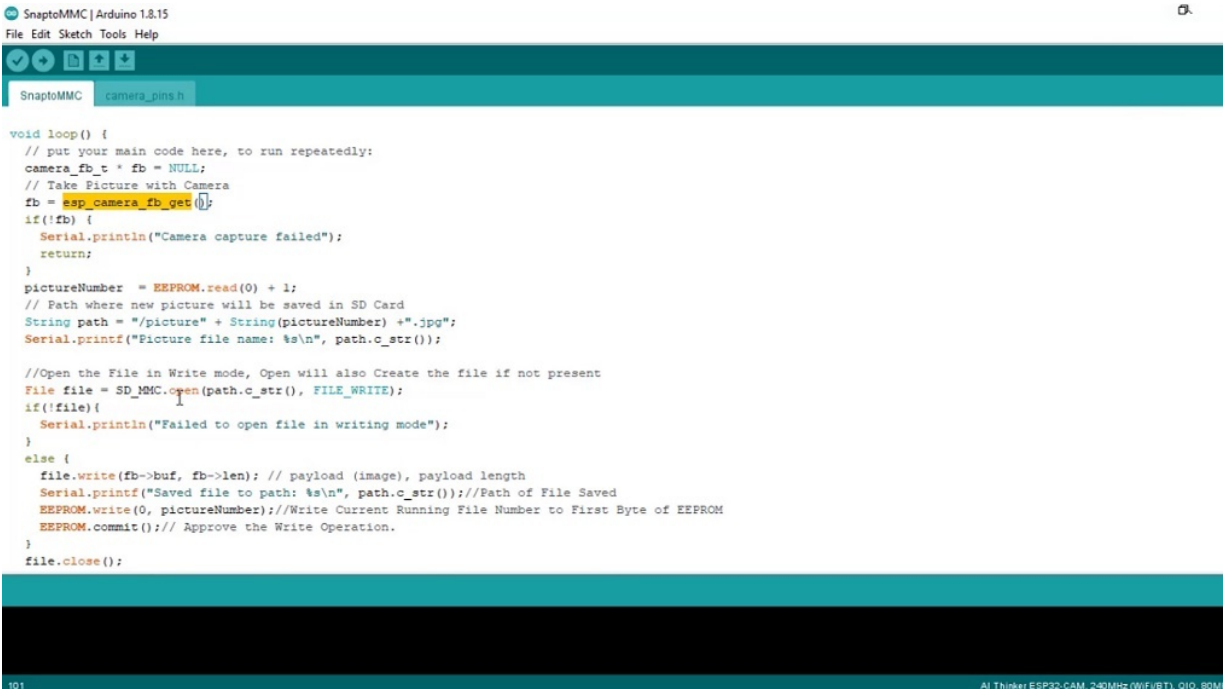
s->set_vflip(s,0); // 0 = disable , 1 = enable
s->set_dcw(s, 1); // 0 = disable , 1 = enable
s->set_colorbar(s, 0); // 0 = disable , 1 = enable
#endif

//Serial.println("Starting SD Card");
if(!SD_MMC.begin()){
  Serial.println("SD Card Mount Failed");
  return;
}
uint8_t cardType = SD_MMC.cardType();
if(cardType == CARD_NONE){
  Serial.println("No SD Card attached");
  return;
}
// Initialize EEPROM with predefined size, start at specified address.
EEPROM.begin(EEPROM_SIZE);

}

void loop() {
  // put your main code here, to run repeatedly:
  camera_fb_t * fb = NULL;
  // Take Picture with Camera
  fb = esp_camera_fb_get();
  if(!fb) {
```

And then I started in America and you said. And also the income. No, the only difference, I mean, is insulting, do I want to take pictures for every five seconds?



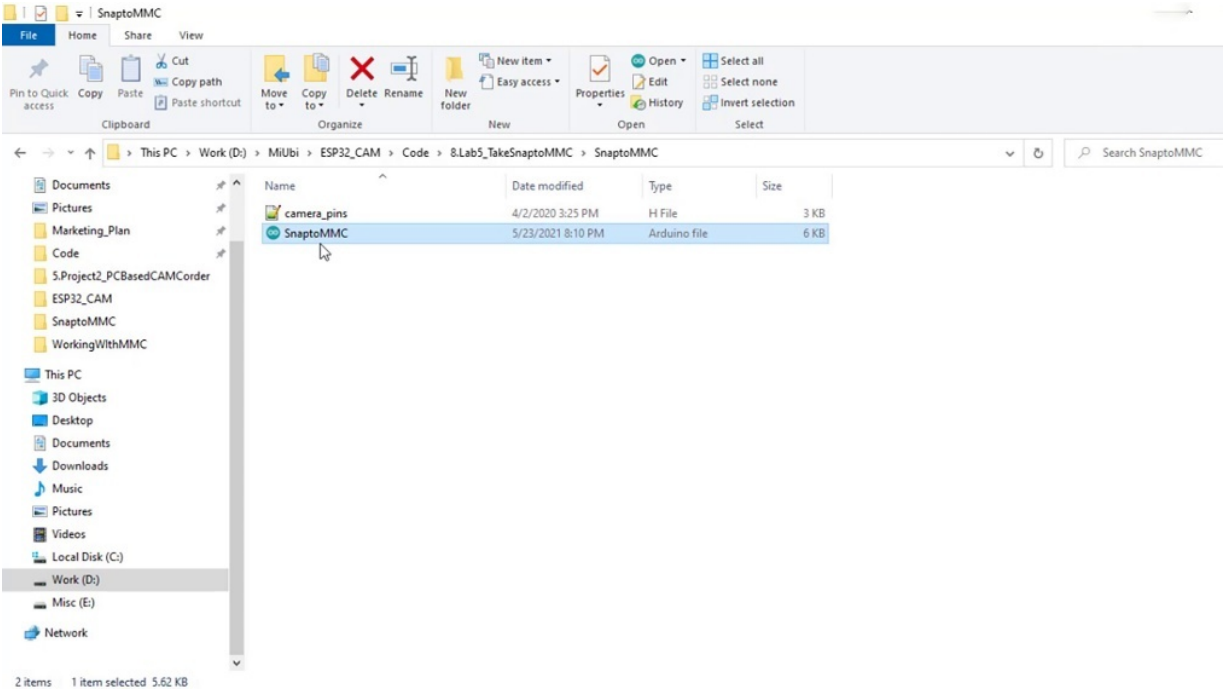
```
SnapttoMMC | Arduino 1.8.15
File Edit Sketch Tools Help

SnapttoMMC camera_pins.h

void loop() {
  // put your main code here, to run repeatedly:
  camera_fb_t * fb = NULL;
  // Take Picture with Camera
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    return;
  }
  pictureNumber = EEPROM.read(0) + 1;
  // Path where new picture will be saved in SD Card
  String path = "/picture" + String(pictureNumber) + ".jpg";
  Serial.printf("Picture file name: %s\n", path.c_str());

  //Open the File in Write mode, Open will also Create the file if not present
  File file = SD_MMC.open(path.c_str(), FILE_WRITE);
  if(!file){
    Serial.println("Failed to open file in writing mode");
  }
  else {
    file.write(fb->buf, fb->len); // payload (image), payload length
    Serial.printf("Saved file to path: %s\n", path.c_str()); //Path of File Saved
    EEPROM.write(0, pictureNumber); //Write Current Running File Number to First Byte of EEPROM
    EEPROM.commit(); // Approve the Write Operation.
  }
  file.close();
}
```

So for that. I use a camera and get the picture and check whether the print book is valid and then update might be. And then opened in America and started writing a number that previously no, I'll wait the buffer picture Aislinn. The below this image, and it's an the image. So. This is how I do it, instead of having their dirtiest extension, so we will have a deputy extensional. Is as simple as that. So if you have two more convictions, so what are the two key things which have been done from the normal camera program to mapping to America so far, that what we will do is we'll try to go to.



We will open this program and. And one more example is their. Working with America. OK, so I will try to come back to the.

```

D:\MIUbn\ESP32_CAM\Code\5.Lab5_TakeSnapttoMMC\SnaptoMMC\SnaptoMMC.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
SnaptoMMC.ino
if(cardType == CARD_NONE){
  Serial.println("NO SD Card attached");
  return;
}
// initialize EEPROM with predefined size, start at specified address.
EEPROM.begin(EEPROM_SIZE);
}
}
void loop() {
  // put your main code here, to run repeatedly:
  camera_fb_t * fb = NULL;
  // Take Picture with Camera
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    return;
  }
  pictureNumber = EEPROM.read() + 1;
  // Path where new picture will be saved in SD Card
  String path = "/pictures" + String(pictureNumber) + ".jpg";
  Serial.printf("Picture file name: %s\n", path.c_str());
  //Open the File in Write mode, Open will also Create the file if not present
  File file = SD_MMC.open(path.c_str(), FILE_WRITE);
  if(!file){
    Serial.println("Failed to open file in writing mode");
  }
  else {
    file.write(fb->buf, fb->len); // payload (image), payload length
    Serial.printf("Saved file to path: %s\n", path.c_str()); //Path of File Sav
    EEPROM.write(0, pictureNumber); //Write Current Running File Number to First
    EEPROM.commit(); // Approve the Write Operation.
  }
  file.close();
}
}

WorkingWithMMC.ino
if(cardType == CARD_NONE){
  Serial.println("NO SD Card attached");
  return;
}
// initialize EEPROM with predefined size
EEPROM.begin(EEPROM_SIZE);
//Read the First Byte of EEPROM to know the running file number before
FileNumber = EEPROM.read() + 1;
}
}
// Path where new file will be saved in SD Card
String path = "/pics" + String(FileNumber) + ".txt";
Serial.printf("New file name: %s\n", path.c_str());
//Open the File in Write mode, Open will also Create the file if not
File file = SD_MMC.open(path.c_str(), FILE_WRITE);
if(!file){ //File Creation or Access Error
  Serial.println("Failed to open file in writing mode");
}
else { //If No Error
  file.println("This is Text File"); // random text Written to File
  Serial.printf("Saved file to path: %s\n", path.c_str()); //Path of File
  EEPROM.write(0, FileNumber); //Write Current Running File Number to EEPROM
  EEPROM.commit(); // Approve the Write Operation.
}
file.close(); //Relinquish the memory allocation done to file handler
}
}
C++ source file length: 5,815 lines:136 Ln:113 Col:50 Sel:13|1 Windows (CR LF) UTF-8

```

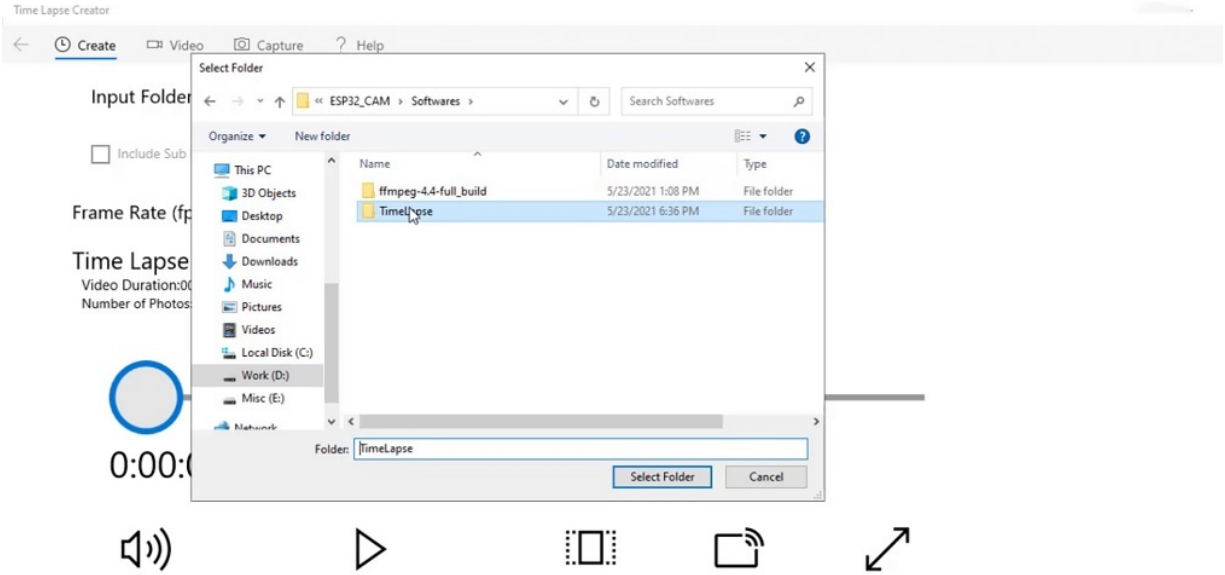
So if you see essentially the camera settings, give me an extra. So this camera settings. I was told everything is in the Abrams's. There we were calling for a number here we are calling like. Everything is exactly the same. Instead of writing. Fine, no, no. So the declaration of credibility, no desire to see it.

```

D:\MIUbn\ESP32_CAM\Code\5.Lab3_Tweak_Camera_Settings\Tweak_Camera\Tweak_Camera.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
SnaptoMMC.ino
//Serial.println("Starting SD Card");
if(!SD_MMC.begin()){
}
Serial.println("SD Card Mounting Failed");
return;
}
uint8_t cardType = SD_MMC.cardType();
if(cardType == CARD_NONE){
  Serial.println("NO SD Card attached");
  return;
}
// initialize EEPROM with predefined size, start at specified address.
EEPROM.begin(EEPROM_SIZE);
}
}
void loop() {
  // put your main code here, to run repeatedly:
}
}

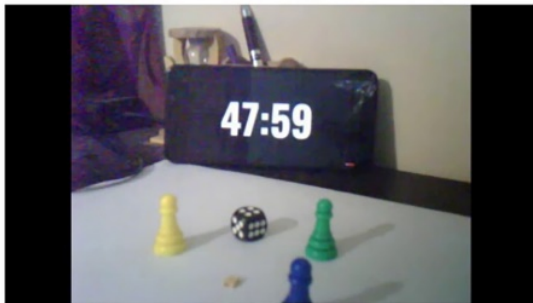
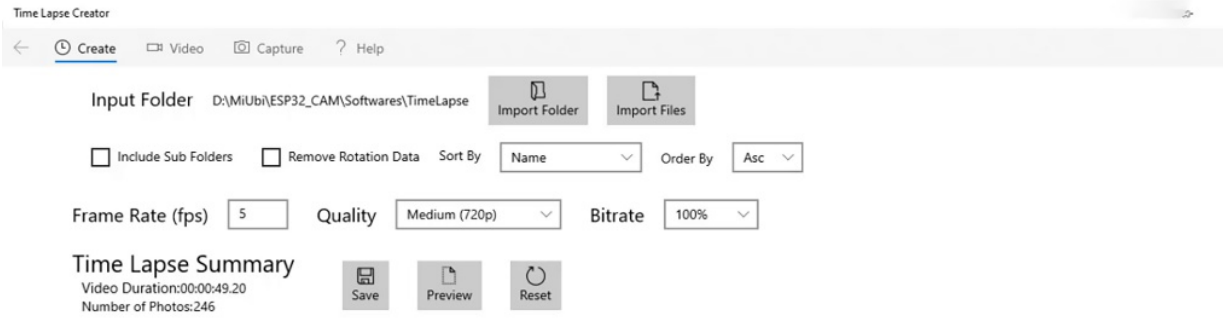
Tweak_Camera.ino
#if 1
// Set your Static IP address of ESP32-CAM Node / Stw Node /
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway Address, in our case the Gat
IPAddress sNet(255, 255, 0, 0); //Sub Net Mask, Default as we have no
IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's
if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Attempt for C
  Serial.println("Configuration with Static IP Failed");
}
#endif
WiFi.begin(ssid, password); // Attempt for Wifi Connection Establishme
while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Connection A
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
startCameraServer(); // Start the Camera Server / Streamer Explained i
Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP()); // This will give you the IP Assigned to
Serial.println("' to connect");
}
}
void loop() {
  // put your main code here, to run repeatedly:
}
}
C++ source file length: 5,964 lines:146 Ln:6 Col:1 Pos:353 Windows (CR LF) UTF-8

```

Status: **Importing files....**

OK, and just launching it and import this folder of complete maps. So I am having 255 images that. So I have left the camera like that and put of that. So let me just play what happens. So these images are taken for every five seconds. I have put a of that so that I was wrong, so it is trying to show it as a medium.



Status: **Time Lapse Created Successfully.**

So you would have seen timelapse we use where people try to show the complete weather. For example, I want to take different images or two hundred images, so I have to set my timer too much greater length. Let's see here. We have put it on in seconds. So we have to put some positions to somewhere around like let's say some eight hours of our time is sufficient so you can take four or five miles per hour or you can put the timer to some. 30 minutes or 40 minutes and then take image and then clear a time for you. That would be a marvelous. So this is how it is. We are taking pictures and story and can with the time frame.

```

SnaptoMMC | Arduino 1.8.15
File Edit Sketch Tools Help

SnaptoMMC camera_pins.h
if(!fb) {
  Serial.println("Camera capture failed");
  return;
}
pictureNumber = EEPROM.read(0) + 1;
// Path where new picture will be saved in SD Card
String path = "/picture" + String(pictureNumber) + ".jpg";
Serial.printf("Picture file name: %s\n", path.c_str());

//Open the File in Write mode, Open will also Create the file if not present
File file = SD_MMC.open(path.c_str(), FILE_WRITE);
if(!file){
  Serial.println("Failed to open file in writing mode");
}
else {
  file.write(fb->buf, fb->len); // payload (image), payload length
  Serial.printf("Saved file to path: %s\n", path.c_str()); //Path of File Saved
  EEPROM.write(0, pictureNumber); //Write Current Running File Number to First Byte of EEPROM
  EEPROM.commit(); // Approve the Write Operation.
}
file.close();
esp_camera_fb_return(fb);

delay(1*60*1000UL); // 1 min
}

```

176 AI Thinker ESP32-CAM 2-SDMMC (R) (F) (E) (T) QIO_80M

So in the current Napranum, he could have just made a small change to put a delay of six decision, I take photographs, but it is 60 seconds. So this delay continues and they wanted to create a time lapse video from this.



So 60 seconds almost, I have captured around 250 images as my variable, 250 for a limited. This is the video, what you're seeing. It came pretty good. I have connected it through the hospital. What kind of word and it was the power is also coming from that cable and I have just left it at the window and it has done a pretty good job of creating a time lapse video. So you can you can play this.

INTRODUCTION TO PWM FOR LED BRIGHTNESS CONTROL

So before we go into the action of how to do the binding of actions over the heads, Tamalpais, that we have built in the previous letter, let us understand a little bit on Boltwood Modulation, which will be useful.

What is PWM

- ▶ Technique to reduce output power by toggling the digital pin at varied duty cycle

- ▶ Imp. parameters

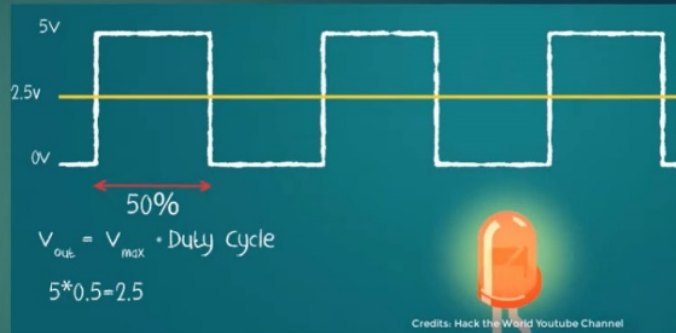
- ▶ Frequency
- ▶ Duty Cycle

In Brief its about Changing the avg.

Power you provide For the output pin by Changing the PW

Note that this is done Completely over a Digital pin, not analog

- ▶ Finds Usage in applications like motor speed control, LED Dimming,
- ▶ multiple color generation using RGB LEDs, where R G B intensities are varied with each other to create multiple colors



I heard in our lectures, Birlsworth modulation is nothing, but you are trying to essentially control the power that is being. Delivered to have been in a positive manner, instead of giving a complete hey, you give a pulse that you recycle later 50 percent or so, which means you are delivering the 50 percent of the power to the bin. So here's the important parameter, is the frequency and the duties. So, for example, if there is a signal which is ranging from zero to firewall's, so the essential vote in this case, if we try to maintain 50 percent of the signal, will be 2.5 words, which would corresponding a certain brightness of value.

What is PWM

- ▶ Technique to reduce output power by toggling the digital pin at varied duty cycle

- ▶ Imp. parameters

- ▶ Frequency
- ▶ Duty Cycle

In Brief its about Changing the avg.

Power you provide For the output pin by Changing the PW

Note that this is done Completely over a Digital pin, not analog

- ▶ Finds Usage in applications like motor speed control, LED Dimming,
- ▶ multiple color generation using RGB LEDs, where R G B intensities are varied with each other to create multiple colors



So if you keep on reducing the duty cycle, it will reduce the brightness of delivery. If you increase the brightness of the if you increase the duty cycle of the Beltway for it will increase the power delivered over that.

What is PWM

- ▶ Technique to reduce output power by toggling the digital pin at varied duty cycle

- ▶ Imp. parameters

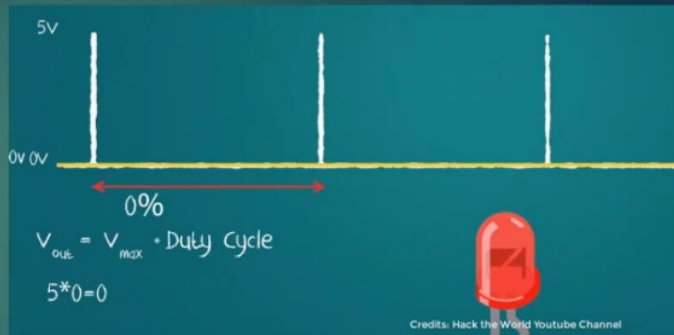
- ▶ Frequency
- ▶ Duty Cycle

In Brief its about Changing the avg.

Power you provide For the output pin by Changing the PW

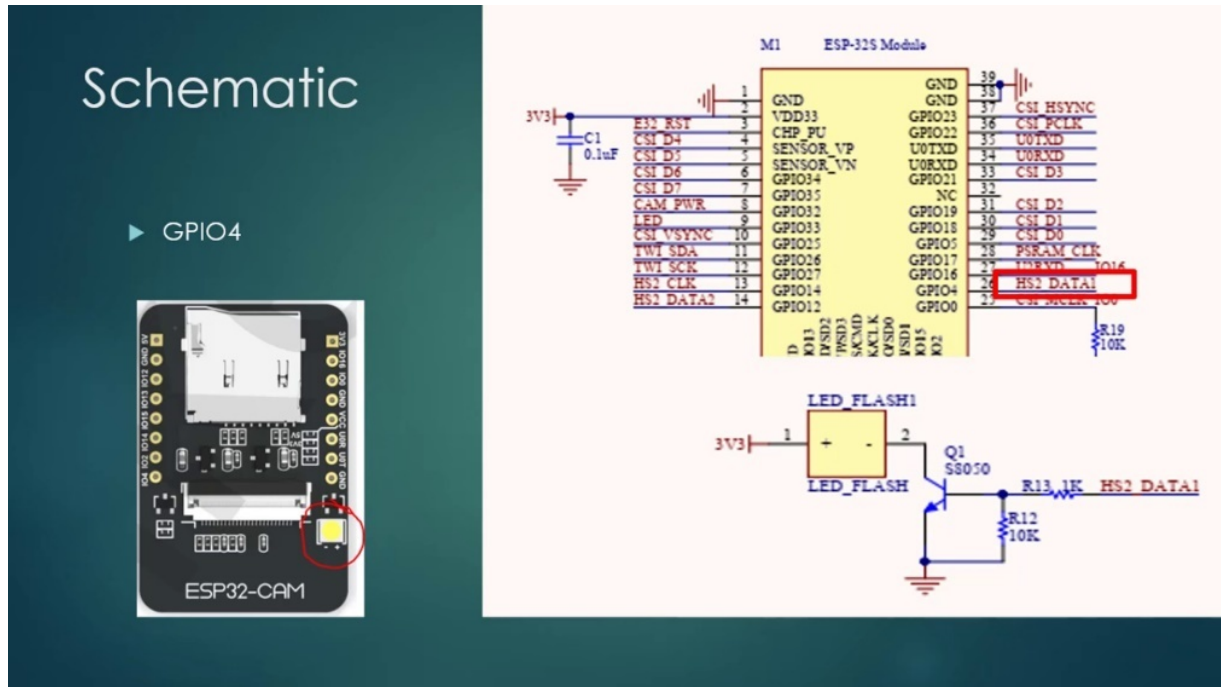
Note that this is done Completely over a Digital pin, not analog

- ▶ Finds Usage in applications like motor speed control, LED Dimming,
- ▶ multiple color generation using RGB LEDs, where R G B intensities are varied with each other to create multiple colors



And essentially the elderly also will glow break. This is, in short, the pulse with modulation where the power delivered is actually being put on the pin.

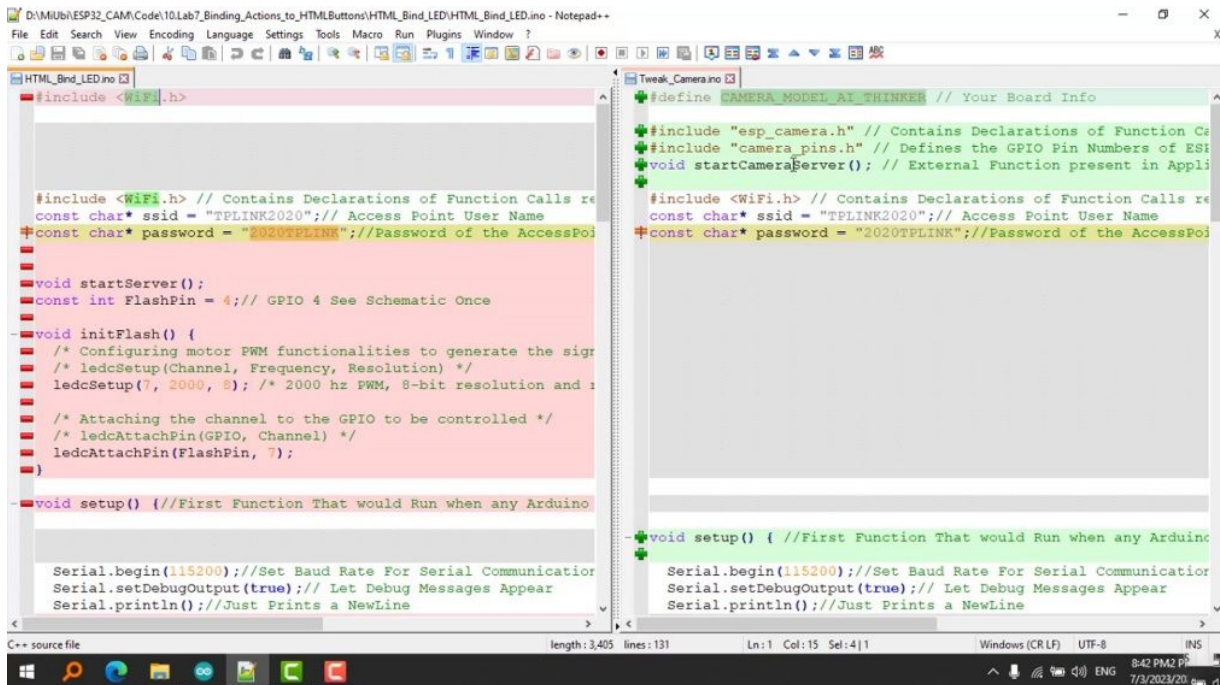
So it will be useful even in luminary applications like and also inductive applications like moulders where you can control the speed of water by doing things with modulation, essentially controlling the power, deliver to the motor, and then which produces the motor speed. If you give more power to the motor, it will run faster. If you give less power to the motor, which will run slow, the city will be controlled more the speed also. So this is one of the things we will be using our lab in this lab.



So come into the schematic. There is a flagellated on the hospital to Kathmandu. And if you see the schematic, it just tried to pin it just to that one. OK, so this latest 231 is actually a to of four. So we will be playing with TPA before in this lecture and we create a pulse channel and that will which will be assigned to certain frequency read. So let us go to the code.

BINDING LED CONTROL ACTIONS TO HTML BUTTONS

let us closely observe the quarterback. This is the first section of the code. Which is the main program. So this is the three camera example, what you have used previously and this is our current program.



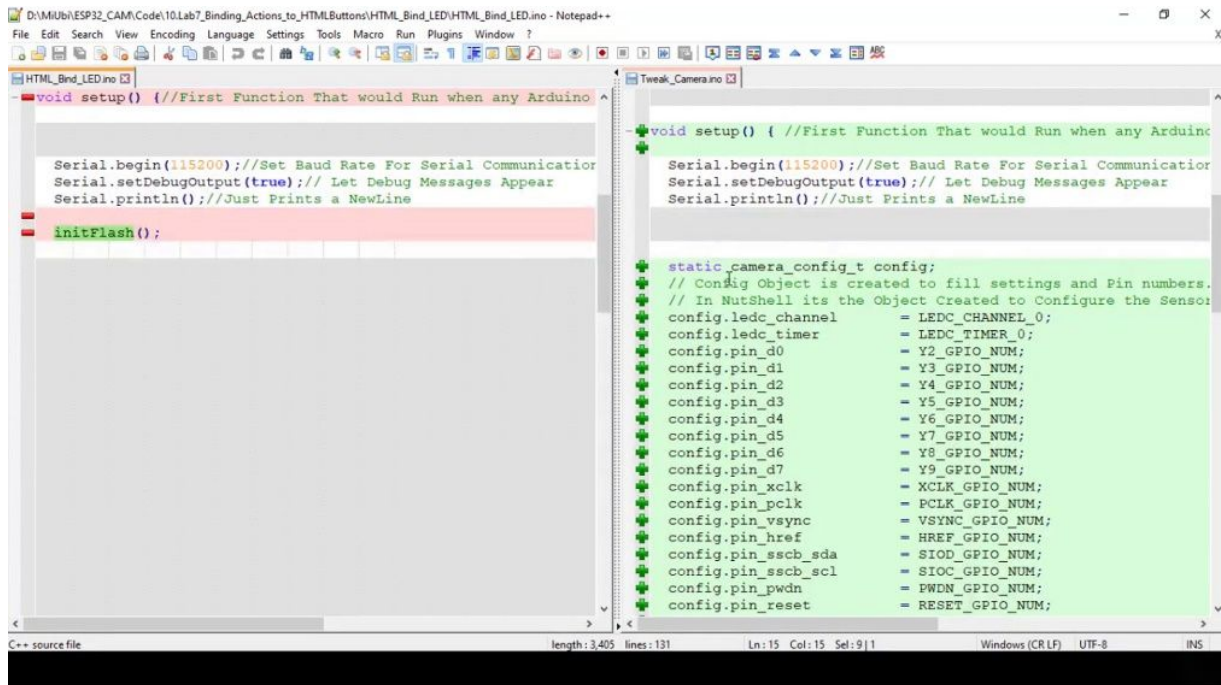
```
#include <WiFi.h>

#include <WiFi.h> // Contains Declarations of Function Calls re
const char* ssid = "TPLINK2020";// Access Point User Name
const char* password = "2020TPLINK";//Password of the AccessPo
void startServer():
const int FlashPin = 4;// GPIO 4 See Schematic Once
void initFlash() {
/* Configuring motor PWM functionalities to generate the sign
/* ledcSetup(Channel, Frequency, Resolution) */
ledcSetup(7, 2000, 8); /* 2000 hz PWM, 8-bit resolution and
/* Attaching the channel to the GPIO to be controlled */
/* ledcAttachPin(GPIO, Channel) */
ledcAttachPin(FlashPin, 7);
}
void setup() (//First Function That would Run when any Arduino
Serial.begin(115200);//Set Baud Rate For Serial Communicator
Serial.setDebugOutput(true);// Let Debug Messages Appear
Serial.println();//Just Prints a NewLine

#define CAMERA_MODEL_AI_THINKER // Your Board Info
#include "esp_camera.h" // Contains Declarations of Function Ca
#include "camera_pins.h" // Defines the GPIO Pin Numbers of ES
void startCameraServer(); // External Function present in Appli
#include <WiFi.h> // Contains Declarations of Function Calls re
const char* ssid = "TPLINK2020";// Access Point User Name
const char* password = "2020TPLINK";//Password of the AccessPo
void setup() ( //First Function That would Run when any Arduino
Serial.begin(115200);//Set Baud Rate For Serial Communicator
Serial.setDebugOutput(true);// Let Debug Messages Appear
Serial.println();//Just Prints a NewLine
```

So I just wanted to compare because we did not make any big changes here. So if you see the comparison, I have removed the camera requirements because we are not working with any kind of camera here. So we will have the effect of touching that. But we will not have the camera up in the declarations and start cameras that will kind of function calls. Instead, we will have to begin servitor of Stateville function called, which will be taking any visual society and password will be there. And this is the important code. I want you to see this particular section where we have started anyways and we have taken a Pinkel flashpoint, which is DPL for and in the initial edition of The Flagellating, what I am doing is there is a function called called ADC Setup. Here I am assigning a maximum frequency of the words, which is to Killens and a Channel seven with the resolution of eight big. This is basically configuring a petroleum entity and then after I'm assigning that problem channel. To the flashpoint, which has been for so

whatever problem functionalities you do will go through this channel to that particular spin, that's nothing more than that. So the set up coliseum in this attack, we actually calling this unit flesh. Initial invasion of the flashily,



```
HTML_Bind_LED.ino
void setup() { //First Function That would Run when any Arduino
  Serial.begin(115200); //Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true); // Let Debug Messages Appear
  Serial.println(); //Just Prints a NewLine
  initFlash();
}

Tweak_Camera.ino
void setup() { //First Function That would Run when any Arduino
  Serial.begin(115200); //Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true); // Let Debug Messages Appear
  Serial.println(); //Just Prints a NewLine

  static_camera_config_t config;
  // Config Object is created to fill settings and Pin numbers.
  // In NutShell its the Object Created to Configure the Sensor
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
}
```

so we will not have this camera bin declarations in our current phase so we can skip them and we will have the IP declaration, stratigraphy declaration as it is, which is there is no change in that.


```

// Set your Static IP address of ESP32-CAM Node / Station.
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway Address, in our case
IPAddress sNet(255, 255, 0, 0); //Sub Net Mask, Default as we
IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Atten
Serial.println("Configuration with Static IP Failed");
}
#endif

WiFi.begin(ssid, password); // Attempt for Wifi Connection Est

while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Conr
delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

```

```

s->set_lenc(s, 1); // 0 = disable , 1 = enable
s->set_hmirror(s, 0); // 0 = disable , 1 = enable
s->set_vflip(s, 0); // 0 = disable , 1 = enable
s->set_dcw(s, 1); // 0 = disable , 1 = enable
s->set_colorbar(s, 0); // 0 = disable , 1 = enable
#endif
#endif

// Set your Static IP address of ESP32-CAM Node / Station.
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway Address, in our case
IPAddress sNet(255, 255, 0, 0); //Sub Net Mask, Default as we
IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Atten
Serial.println("Configuration with Static IP Failed");
}
#endif

WiFi.begin(ssid, password); // Attempt for Wifi Connection Est

while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Conr
delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

```

And instead of calling it cameras that would make all of the normal salary at, we start that service and print the IP and go back to the important parties, we have to see what is exactly inside this structure.

```

IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Atten
Serial.println("Configuration with Static IP Failed");
}
#endif

WiFi.begin(ssid, password); // Attempt for Wifi Connection Est

while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Conr
delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

startServer(); // Start the Server

Serial.print("Web Server Ready! Use 'http://");
Serial.print(WiFi.localIP()); // This will give you the IP Ass
Serial.println("' to connect");
}

void loop() {
// put your main code here, to run repeatedly:
}

```

```

IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Atten
Serial.println("Configuration with Static IP Failed");
}
#endif

WiFi.begin(ssid, password); // Attempt for Wifi Connection Est

while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Conr
delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

startCameraServer(); // Start the Camera Server / Streamer Exp

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP()); // This will give you the IP Ass
Serial.println("' to connect");
}

void loop() {
// put your main code here, to run repeatedly:
delay(10000);
}

```

Now coming to the court, as we have previously discussed in today's society, password.

```
HTML_Bind_LED | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED $ app_httpd.cpp
#include <WiFi.h> // Contains Declarations of Function Calls related to ESP-IDF and Arduino HookUps for
const char* ssid = "TPLINK2020";// Access Point User Name
const char* password = "2020TPLINK";//Password of the AccessPoint

void startServer();
const int FlashPin = 4;// GPIO 4 See Schematic Once

void initFlash() {
  /* Configuring motor PWM functionalities to generate the signal */
  /* ledcSetup(Channel, Frequency, Resolution) */
  ledcSetup(7, 2000, 8); /* 2000 hz PWM, 8-bit resolution and range from 0 to 255 */

  /* Attaching the channel to the GPIO to be controlled */
  /* ledcAttachPin(GPIO, Channel) */
  ledcAttachPin(FlashPin, 7);
}

void setup() { //First Function That would Run when any Arduino Based Board Starts

  Serial.begin(115200);//Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true);// Let Debug Messages Appear
  Serial.println();//Just Prints a NewLine
}
```

And then we configurability and then we set up the IP and we go for start. Now let us right from start server in this. So it is quite a little bit different from whatever we have written previously. Like we have tried to return a streaming server where there was a straightaways streaming handler.

```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED $ app_httpd.cpp
#include "esp_http_server.h"
#include "Arduino.h"

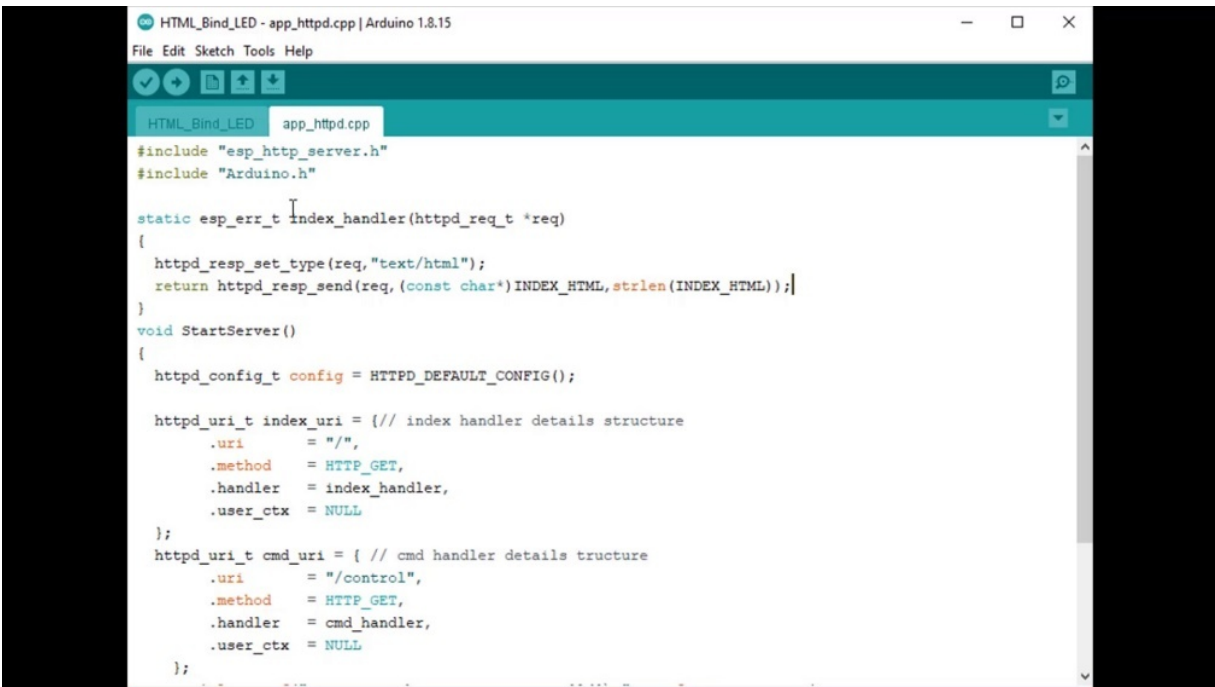
void StartServer()
{
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();

  httpd_uri_t index_uri = { // index handler details structure
    .uri = "/",
    .method = HTTP_GET,
    .handler = index_handler,
    .user_ctx = NULL
  };

  httpd_uri_t cmd_uri = { // cmd handler details structure
    .uri = "/control",
    .method = HTTP_GET,
    .handler = cmd_handler,
    .user_ctx = NULL
  };

  Serial.printf("Starting web server on port: '%d'\n", config.server_port);
  if (httpd_start(&server_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(server_httpd, &index_uri); // Register index handling structure
    httpd_register_uri_handler(server_httpd, &cmd_uri); // Register cmd handling structure
  }
}
```

Similarly, we will have you here. Come on, handler. And also in the handler index handler is nothing but the handler coumarin handler is nothing but whatever request you are getting from the next page. So let us start from the ground where we first include the essential liabilities. What are the essential liabilities now required as we are already dealing with the server? So having to decide whether or not it is essential and also essential and also for general usage, we should need the order in order to. Write the story, make us all make things clear. No, the very first is we wanted to write several great. So what is that and say to these types of. Primarily, we'll have the strategic reconfiguration, which you already know very well defined clearly, configuration object. All right, and then the important thing is I want to create as like we have created in an extreme. I'm trying to create a structure here. You don't mind. Yeah. This is the index handler details. Structure, so this structure will consist of the main where is the world goods coming from? What is the matter? See, there are two methods here. Usually to get an activity post posters. And I think what you are trying to say, something to be said, were outlined in the history to be communication. These are the typical things used to be Gutman's were trying to get some request. But here, as we begin this hour, we are trying to see what is the thing which is coming, whether they get request or opposed to it, see it in terms of service, because we are writing this over. OK. No, there are as like I said, there are two handlers here. One is the handler. When is the next big thing? There is nothing but the handler. CAMAC is the Handey, which is required for you to understand the incoming commands from the page and take necessary actions. So what is there here as we previously registered our stream handlers? So we are again registering our index handler structure and command structure to the register. You are. OK, this is the architecture of the framework which is already existing with Arduino for. Now we we're left with two things to do. Your next handler do what should mean. Come on. So it makes handler is quite simply only.



```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp
#include "esp_http_server.h"
#include "Arduino.h"

static esp_err_t index_handler(httpd_req_t *req)
{
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char*) INDEX_HTML, strlen(INDEX_HTML));
}

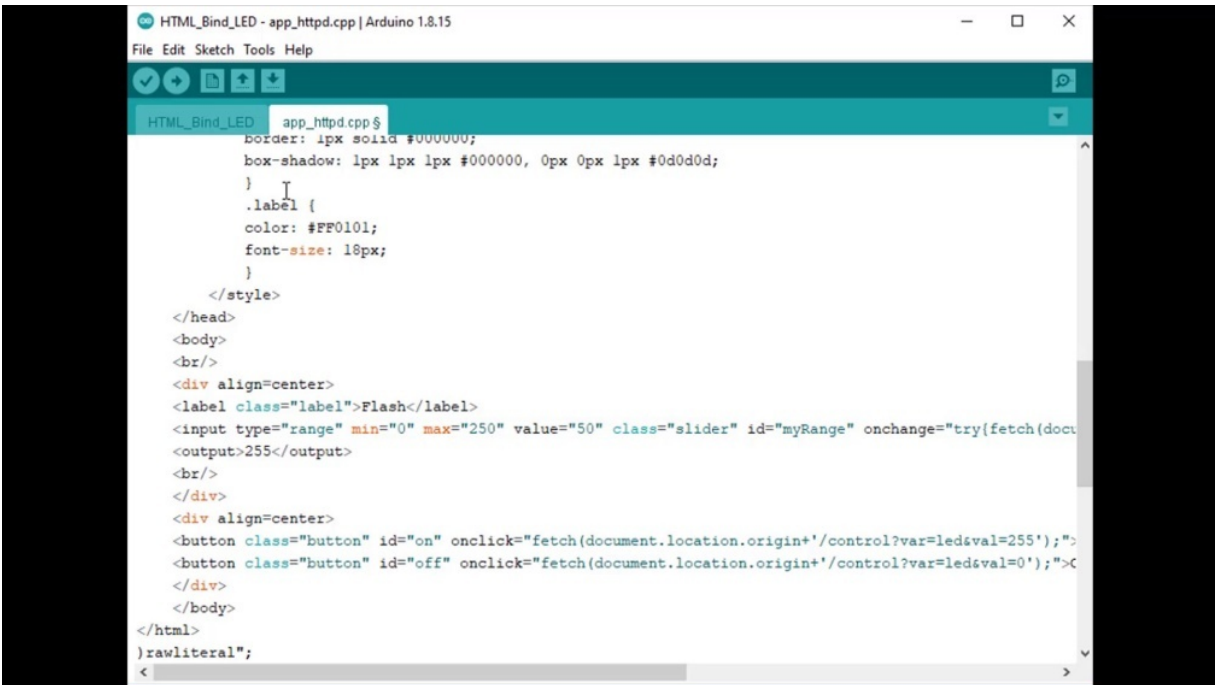
void StartServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    httpd_uri_t index_uri = { // index handler details structure
        .uri      = "/",
        .method   = HTTP_GET,
        .handler  = index_handler,
        .user_ctx = NULL
    };

    httpd_uri_t cmd_uri = { // cmd handler details structure
        .uri      = "/control",
        .method   = HTTP_GET,
        .handler  = cmd_handler,
        .user_ctx = NULL
    };
};
```

Let me read the exchange function here. So the index handler is articulate and bright, which usually returns, etc. by. Ethics handler, make sure that you write the same function in whatever you trying to register in this structure, it receives basically a request from the page, which is our primary question is quarter. We collect that in the request and then. The second response, they say basically we are trying to here service the function of a request and sending back the same e-mail page to the browser so that it can be and his team will be doing it. That is what is in this whole. So it really underscores points on this course. So first of all, we have to tell the coming or the data which is going to come is a statement and it's been sent in a text format. And. There certainly handler tippity response and. He was threatening to start this volume of the sorry request by Caster's to a constant character type, by the way, they usually do this so that during the process of handling it, the expert should not be tampered or indicated, should not be made any change, any change in the original pitch. Right. So you are trying to take the statement of completing the statement, which. OK, now here comes the. You are saying that we are sending the next of stimulus, but where is the stimulus? So what are is simply and trying to. Now, I want to bring the complete stimulus package into this program, which has stimulated the same stimulus that we have developed in the last election. I'm trying to bring it as a complete string because the testimony format

cannot be directly given in the program. So we give it in a text format, albeit with a string format.



```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp $
border: 1px solid #000000;
box-shadow: 1px 1px 1px #000000, 0px 0px 1px #0d0d0d;
}
.label {
color: #FF0101;
font-size: 18px;
}
</style>
</head>
<body>
<br/>
<div align=center>
<label class="label">Flash</label>
<input type="range" min="0" max="250" value="50" class="slider" id="myRange" onchange="try(fetch(docu
<output>255</output>
<br/>
</div>
<div align=center>
<button class="button" id="on" onclick="fetch(document.location.origin+'/control?var=led&val=255');">
<button class="button" id="off" onclick="fetch(document.location.origin+'/control?var=led&val=0');">
</div>
</body>
</html>
)rawliteral";
<
```

So let me bring this completely legitimate page here. Right. I have brought this to my village and I see this. This is indeed the string. This is starting on this tour, so we are declaring the world literally OK, everything, whatever it Tamalpais we had previously. But but there is one single change.

```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

HTML_Bind_LED app_httpd.cpp

FF0101;
: 18px;

<
<label>Flash</label>
<input type="range" min="0" max="250" value="50" class="slider" id="myRange" onchange="try(fetch(document.location.origin
put>

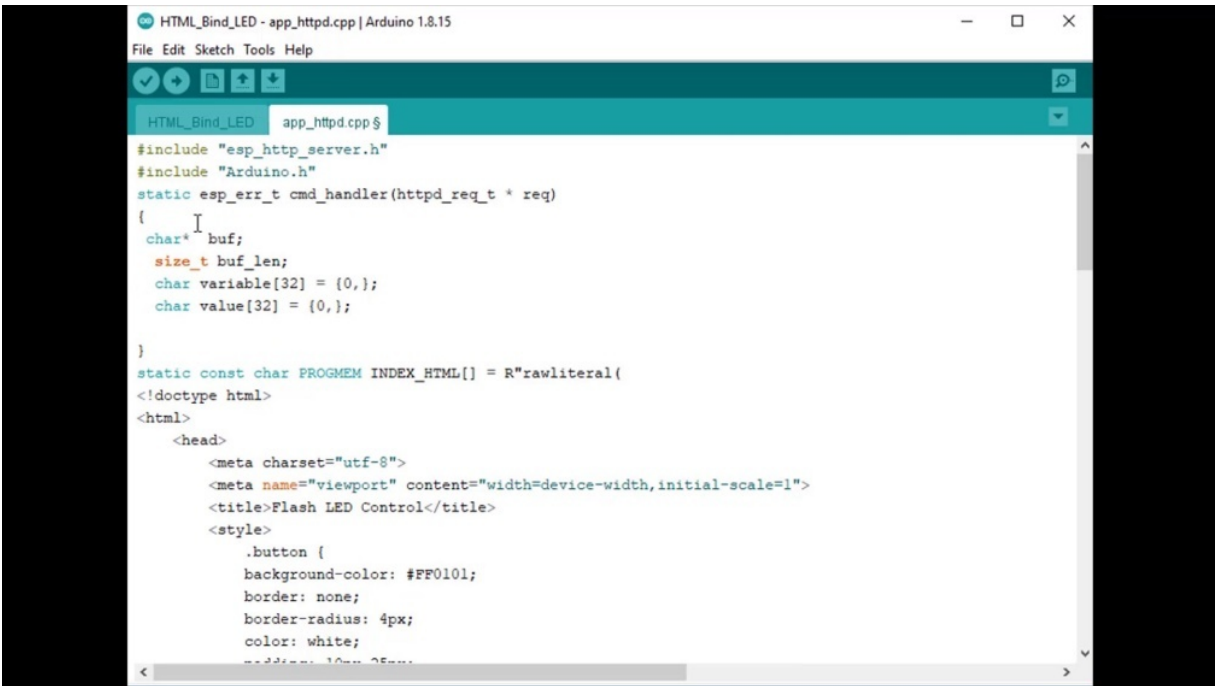
<
<button id="on" onclick="fetch(document.location.origin+'/control?var=led&val=255');">ON</button>
<button id="off" onclick="fetch(document.location.origin+'/control?var=led&val=0');">OFF</button>

http_handler(httpd_req_t *req)

send(req, "text/html");
<
```

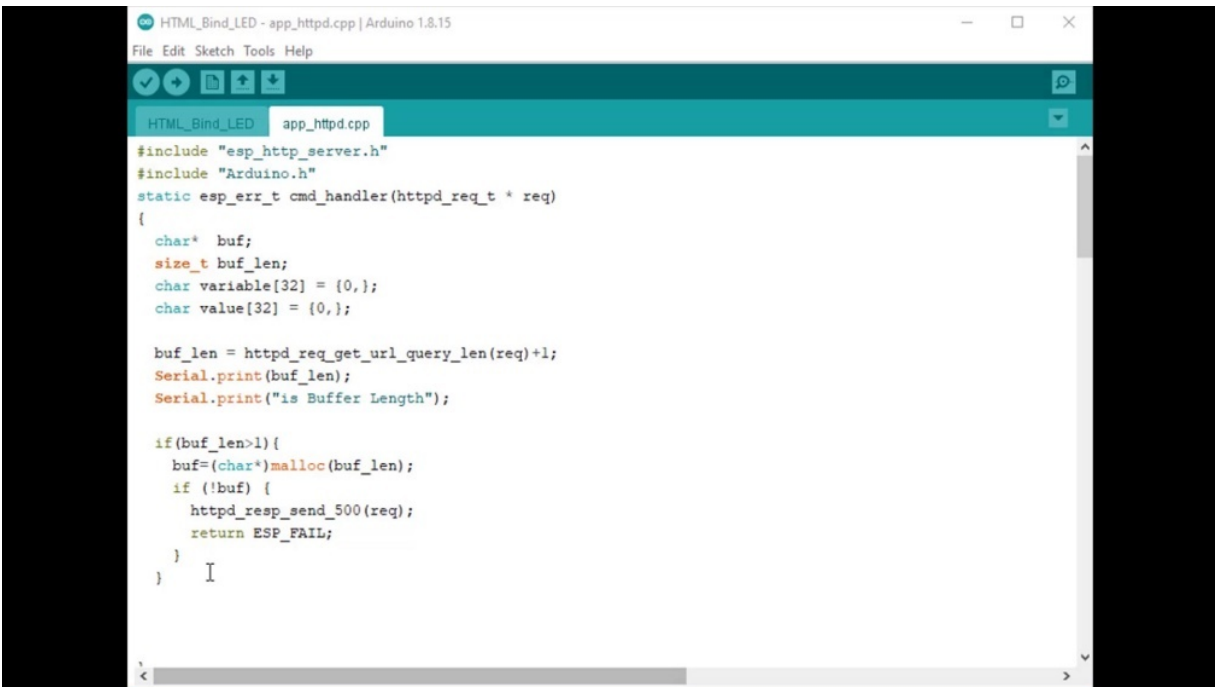
There is a function call here called Unclick. On change unclick is nothing, but whenever somebody presses on this button clicks on this button, the page has tries to generate a request. See, here it is trying to do something called fetch documents, not location, not origin. This is nothing, but it tries to fix the current liberal or in our case, it is the 190 to 168 zeroed out on our date and then spending it with astronomical control. Remember? Just control, we will come back to this and we are also creating this thing, I mean, creating a variable variable is a. And value is to. This is what we are trying to create and send it as request here on the Slater case and in trying to create the same thing. But we are trying to give the very flash here the variable. So in order to discriminate the basically you were response handler or the command handler should understand whether it is a request or a button request. And that is why we are saying is available on my current variable name is Allouni. That means I have to take it function or taking action on the alien, either go on or off. If it comes from a slider, it comes with a variable called flash and the value just is not valid, which means in the slightest current. Well, you know, as I said you previously, just concentrate on this phrase called control press call control slash control is my Web page. Control page are the command handling page. So if a request comes to you like something like 190 to 168, not zero node one or slash control like this question mark, which means that it is his duty to request

state of it with the opinion. So the Web page will send this request upon clicking on this request will descend upon clicking off. If you click on off button, this will send you a value zero. If we click on on button, this is only available 255 in case of later misleaders. Current value will be sent in the request. No, this request has come from this system will be just been sent to the browser by Nixonland. No what next.



```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp $
#include "esp_http_server.h"
#include "Arduino.h"
static esp_err_t cmd_handler(httpd_req_t * req)
{
    char buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};
}
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Flash LED Control</title>
    <style>
      .button {
        background-color: #FF0101;
        border: none;
        border-radius: 4px;
        color: white;
        padding: 10px 20px;
      }
    </style>
  </head>
  <body>
    <div class="button">ON</div>
    <div class="button">OFF</div>
  </body>
</html>
)rawliteral";
```

And the next thing is your in handy and the next thing is your command handling. So how are we going to bring the command hangi functionality to handle the commands? So we have to write our command handler now, write the function, call what are we going to do and say this is primarily to go ahead. We define some variables which are our character buffer, which we take, and philately, which we take the request and fill it to the buffer and also ambivalent variable and also variable in value as like we are sending to barometer's call back and Val. Right. So we are just trying to take a few character type relevance to this. No. I have to take the request, so in order to store the request, I have to have a boyfriend,

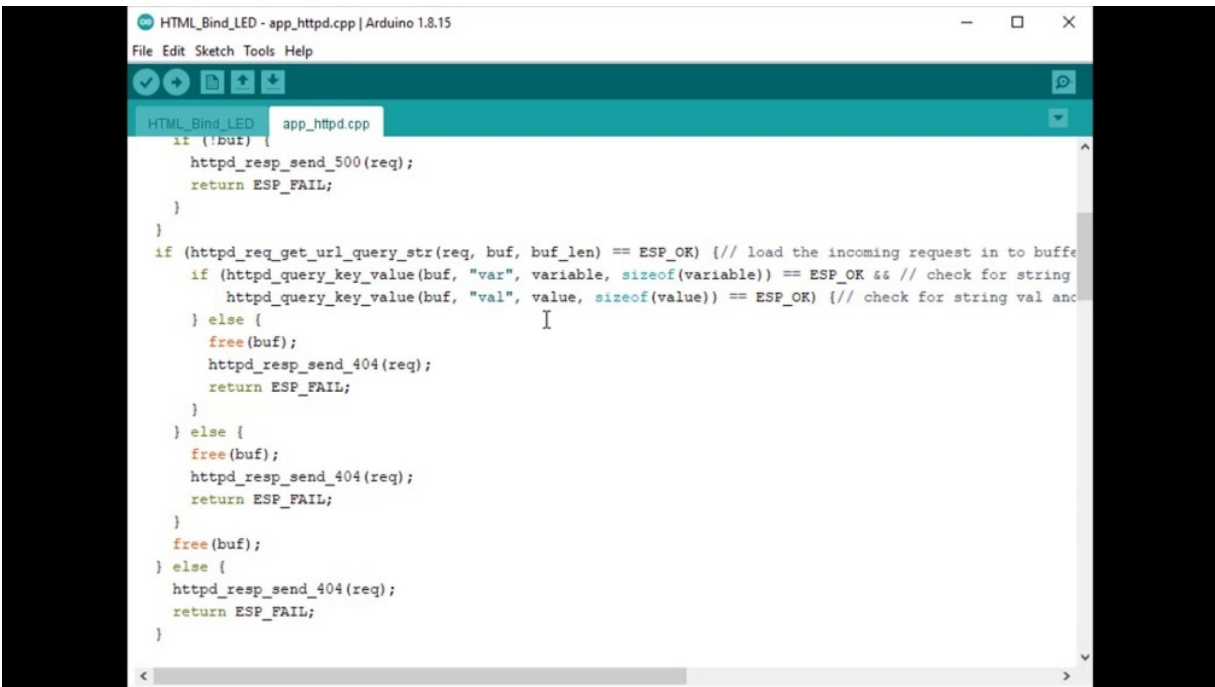


```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp
#include "esp_http_server.h"
#include "Arduino.h"
static esp_err_t cmd_handler(httpd_req_t * req)
{
    char* buf;
    size_t buf_len;
    char variable[32] = {0,};
    char value[32] = {0,};

    buf_len = httpd_req_get_url_query_len(req)+1;
    Serial.print(buf_len);
    Serial.print("is Buffer Length");

    if(buf_len>1){
        buf=(char*)malloc(buf_len);
        if (!buf) {
            httpd_resp_send_500(req);
            return ESP_FAIL;
        }
    }
}
```

so let me know. So what is the government of the request which has come which will be retained by the facility? Deputy underscore the request get. So we are trying to get the. And what is the length of the quota which has come to me. Just add one to it because it starts from zero. So serial or print print Ferland so that I am if I did not receive anything, my preference should be zero. So I just want to print what is my boyfriend currently so that I can do the advertising properly. As I have already told you, that we have only one way of debugging in ordinal. Is printing and taking the government, if it is greater than one and getting them ready for it, if it is not greater than one, then I just do something which is sending you for your response. Right. He. Now, here comes a very important job of seeing. What has a response? So what has come at a cost? So this is a typical small thing which I want to put on the table. And so this is the one I just forwarded the video.



```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp
if (!buf) {
  httpd_resp_send_500(req);
  return ESP_FAIL;
}
if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) { // load the incoming request in to buffer
  if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) == ESP_OK && // check for string
      httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK) { // check for string val and
  } else {
    free(buf);
    httpd_resp_send_404(req);
    return ESP_FAIL;
  } else {
    free(buf);
    httpd_resp_send_404(req);
    return ESP_FAIL;
  }
  free(buf);
} else {
  httpd_resp_send_404(req);
  return ESP_FAIL;
}
```

So this particular thing, we are trying to get people to land and create a. Here we are just trying to really buffer Ferland of elements of really buffer land, of elements from the request to our buffer between now in this, I'm trying to query the key. What is our key? The key is variable. We are. And what are those values? Those will be stored in the variables, variable and also value. It is everything is fine here, it is OK, or is there is a. We have to just send your response to the page back and then get out of here right now. I know we have understood what has come to us. Let us assume we have passed all this. We did not feel that means our string, which is great. It is also proper. And we were able to really the values until properly and the both of them are being true. And then we'll go ahead take.

```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp $
    return ESP_FAIL;
}
free(buf);
} else {
    httpd_resp_send_404(req);
    return ESP_FAIL;
}

int val = atoi(value);
Serial.print("Variable : ");
Serial.print(variable);
Serial.print(" With Value : ");
Serial.print(val);

if (!strcmp(variable, "flash"))
{
    ledcWrite(7, val); //Write the PWM value directly, which is equivalent to controlling brightness
}

}
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
```

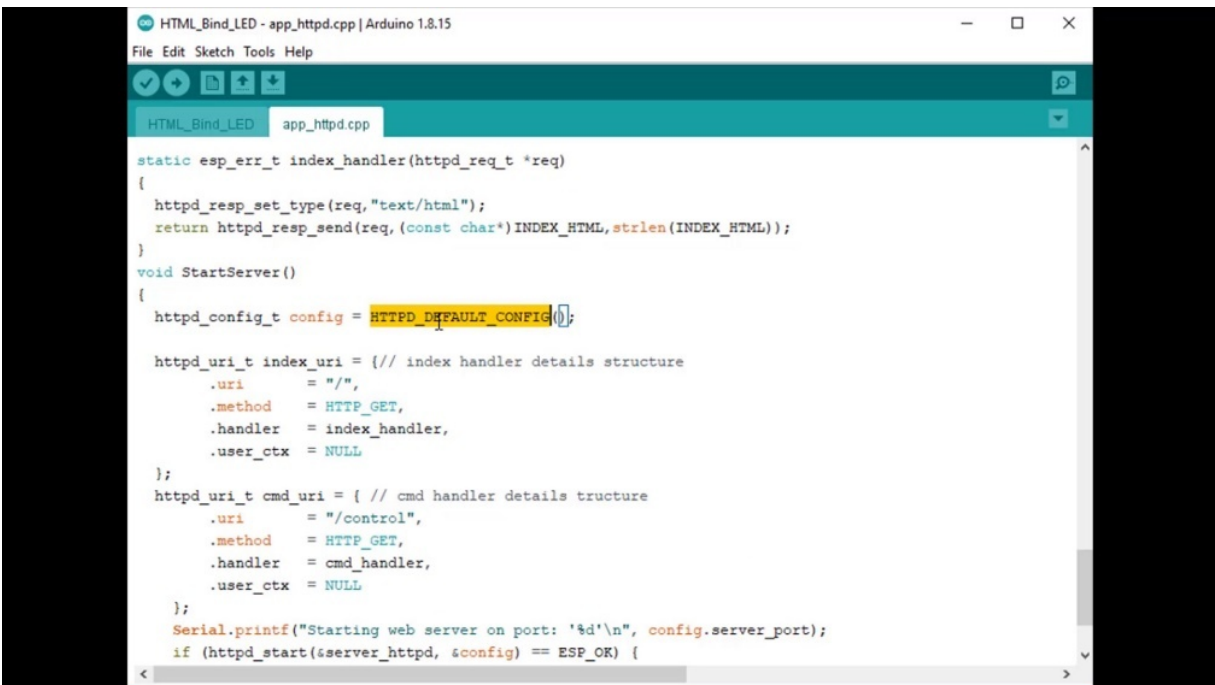
And I will try to convert Disvalue, which is right from aski during visit. Because remember, we have made this character, so that is why we are converting in this. So now I have read what is the value which has come to me. So it is always better to bring to the values what we received and what we have read. No, the request of a variable which has come to you is essentially a type flash just because your page is sending either flash.

```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp $
}

int val = atoi(value);
Serial.print("Variable : ");
Serial.print(variable);
Serial.print(" With Value : ");
Serial.print(val);

if (!strcmp(variable, "flash"))
{
    ledcWrite(7, val); //Write the PWM value directly, which is equivalent to controlling brightness
}
if (!strcmp(variable, "led"))
{
    ledcWrite(7, val);
}
else
{
    Serial.println("variable");
}
httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "");
return httpd_resp_send(req, NULL, 0);
}
<
```

Or. So let me check first whether it is a fake flash, if it is of type flash, then what should I do? And this comparing my variable, which is loaded with this string flash. So if it is the string flash, then the Channel seven. Which is the problem? Jenny, I want to raise the value straightaway into that. Right along with the variable, we also get you. So what is the value? It is the current value, so which is nothing but indirectly corresponds to the brightness of the reality that we want to set. Right. So the range of zero to 255. So we are just trying to set the brightness here. So we are just the value here. Which is essentially would lower the value, lower the lower, the higher the value, higher, the higher the brightness rate. Similarly. We will have to see whether the incoming variable and the incoming request which has come is obviously late if it is a failure. It is obviously delayed if it is a failure. Then again, they develop and try to it. And it can be either zero or 250 for the zero is off and for this 250, it is on straight. If something else comes, but in doing so, it. That's a. So let us return to this. From this function. But.



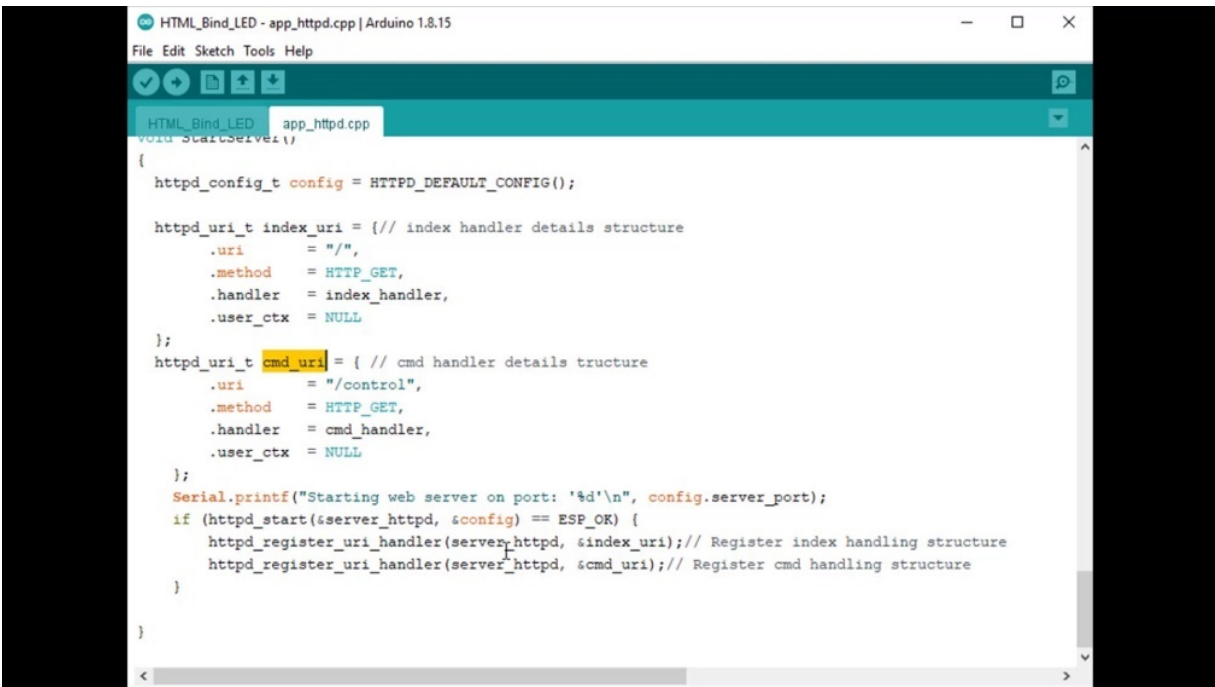
```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp
static esp_err_t index_handler(httpd_req_t *req)
{
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char*) INDEX_HTML, strlen(INDEX_HTML));
}
void StartServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG(0);

    httpd_uri_t index_uri = { // index handler details structure
        .uri      = "/",
        .method   = HTTP_GET,
        .handler  = index_handler,
        .user_ctx = NULL
    };

    httpd_uri_t cmd_uri = { // cmd handler details structure
        .uri      = "/control",
        .method   = HTTP_GET,
        .handler  = cmd_handler,
        .user_ctx = NULL
    };

    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&server_httpd, &config) == ESP_OK) {
```

Now we have almost the latest. Let me first call back again and see what other things we have done, so we have started here. Starts with this creating and mixing in command and structures are created and they are being fed to the handler.



```
HTML_Bind_LED - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
HTML_Bind_LED app_httpd.cpp
void startServer()
{
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();

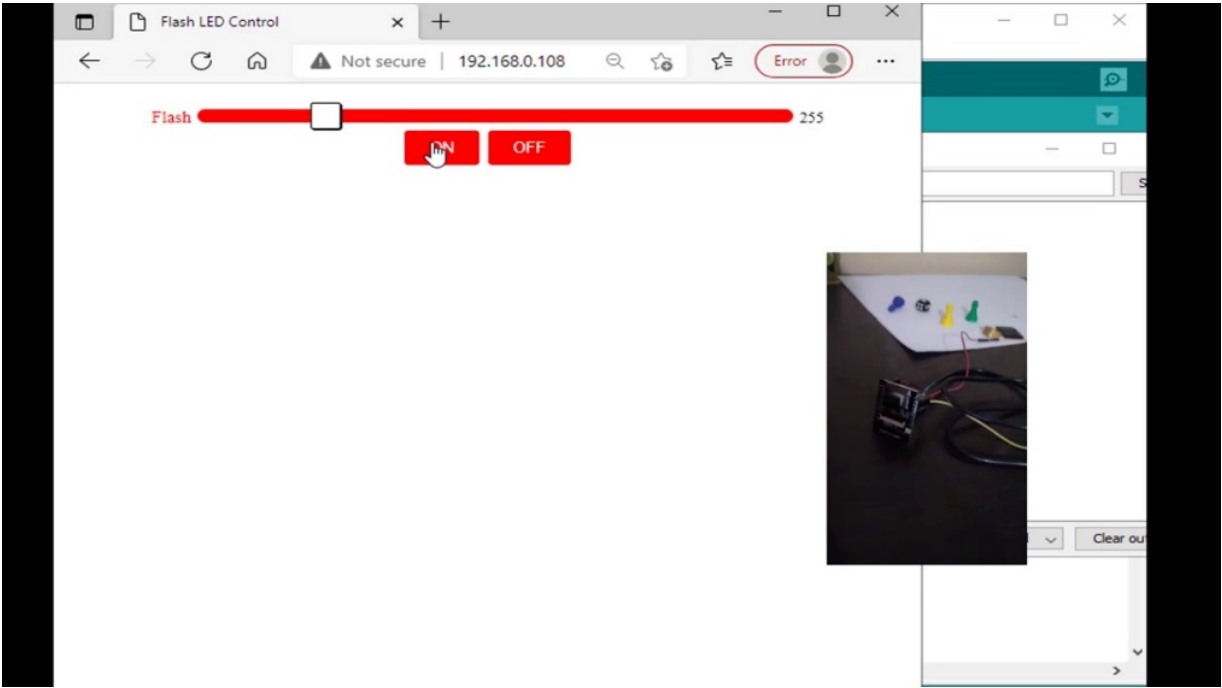
  httpd_uri_t index_uri = { // index handler details structure
    .uri      = "/",
    .method   = HTTP_GET,
    .handler  = index_handler,
    .user_ctx = NULL
  };

  httpd_uri_t cmd_uri = { // cmd handler details structure
    .uri      = "/control",
    .method   = HTTP_GET,
    .handler  = cmd_handler,
    .user_ctx = NULL
  };

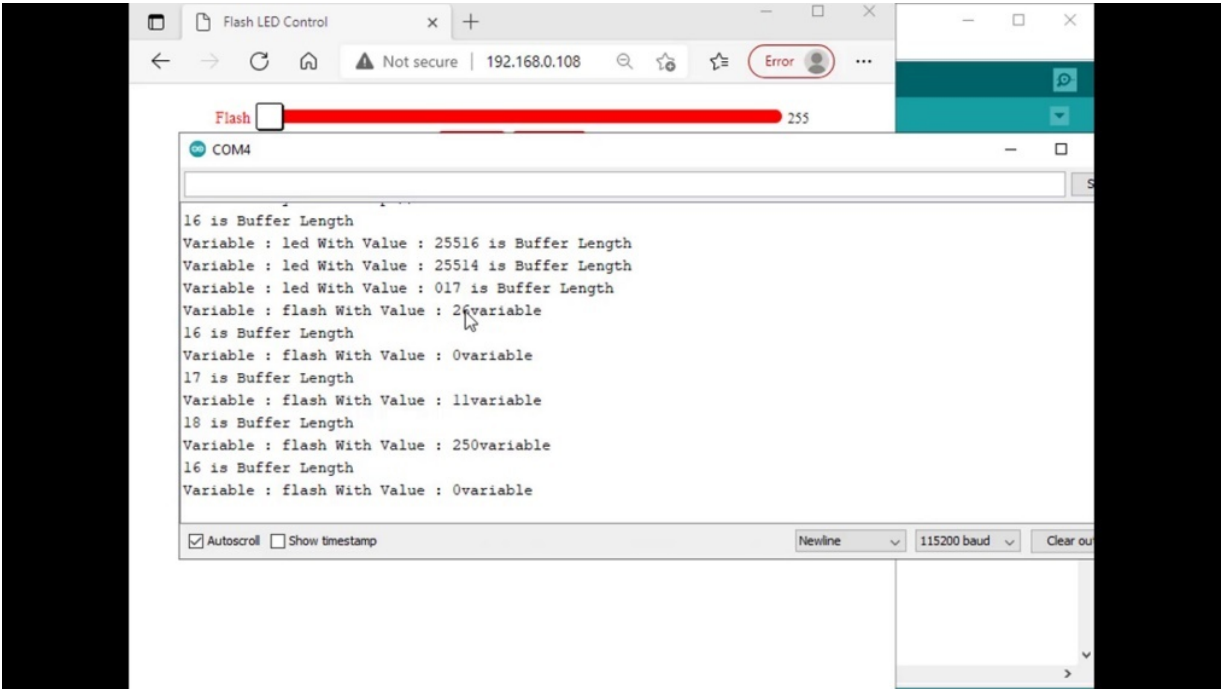
  Serial.printf("Starting web server on port: '%d'\n", config.server_port);
  if (httpd_start(&server_httpd, &config) == ESP_OK) {
    httpd_register_uri_handler(server_httpd, &index_uri); // Register index handling structure
    httpd_register_uri_handler(server_httpd, &cmd_uri); // Register cmd handling structure
  }
}
```

You are a handler. Function calls you have to destroy them. And then we have written the actual handlers, which is the handler and command handler, and left nothing. What does the communications page from this network to decline? And the CommonHealth tries to take the request, which is come from the page. It can be of two types, it can be a flash input or it can be a. Oh, El-Erian, right, you have to see that this can be off type. Flash or ality, if it comes, it can have two value to 250, the flash comes, it will have the current existing requestor slated value that we will use to evaluate whether the values which are coming at proper or not. And then convert it into the string, into a value and then take the decision. But we're still we're trying to ride straight over to the Wall Street, to the pulse with modulated. And contrariness. No, let see dysplastic go and see the demonstration.

AP, let me open my browser. Two, upon clicking on the listen, upon clicking off, it is off, even I can control the brightness.



This is low brightness. As you can read more, this is much higher and reducing it when I'm doing this stuff, I was able to see the values which have been coming over of serial production.

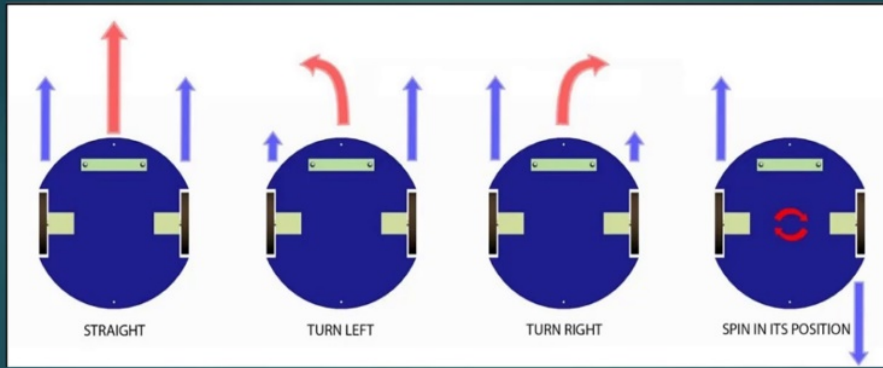


OK. Of this explains you properly how to find actions. Off to his Timal buttons on the website of.

Basics of a Car Control

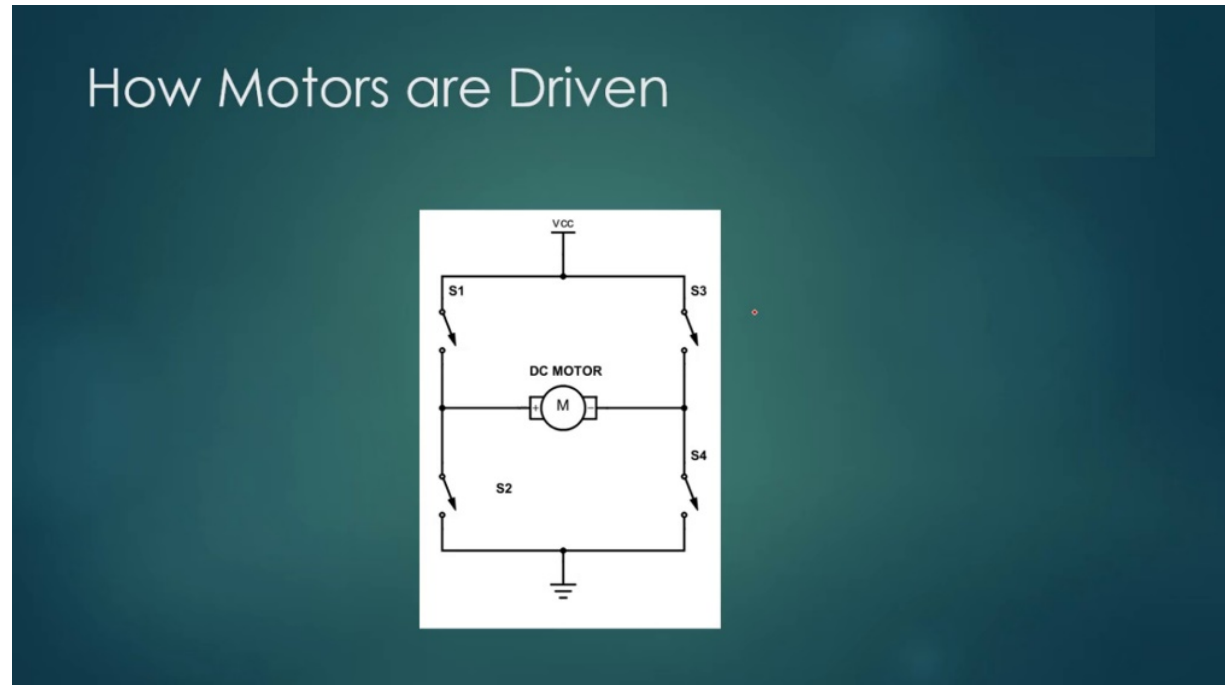
we'll see about motor driving. So before actually trying to drive the motor home, we should understand how this complete technology of a robot car assembly works. Even if you are already familiar with the robots movement and how to control the motors, this lecture is not for you. So you can skip this. So for people who are very new to motor driving and also doesn't know about how to assemble the robot cars, they have to have to go through this. So we are going to make something like this not exactly the same, because it depends on the whatever you have ordered. So but most of the cheap or the low cost robots someplace will be more or less the same. We'll have to build a new Castroville. This is called Castroville because it's a freewheeling, free and free running wheel. So based on the movement of the robot, it can move in any direction. So with which we will reduce the number of vehicles and the power required also. So I would suggest you do go over to Villarrubia simply with gasoline. So if you're not familiar with how to assemble this battery, you by a fully assembled robot instead of buying a robot which is completely dismantled and it comes to you in pieces. You have to join them to make a robot. So in the next, we will show you how to do it, how to assemble a robot. If you don't wish to get into that, you can straight away by robots which are assembled. So in our case, we will not use this battery pack. Instead, we will be using a power bank. Rest of the assembly will be looking thing here, will have a Castroville here we will be having in our system or the party system. The parental system will hold, the U.S. military can fight.

How to move a Car



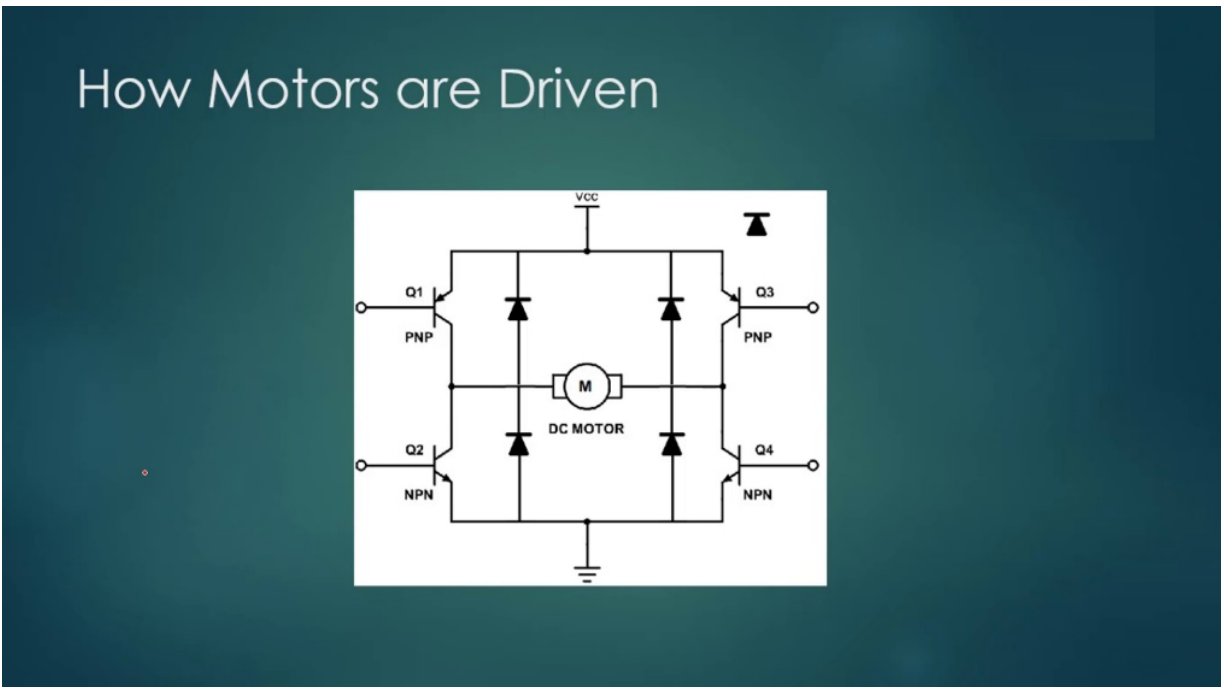
No, coming to moving a car, how to move a car, how to move to Will. So imagine this is the Castroville position, OK? And this is the left wing. This is the right wing. OK, so let me take the pen. Yeah, in order to move the robot to the right, first of all, let see straight, so if you move this robot where the Fosterville and also the second will be the same speed boat will move forward without any problem. Let us say you want to take a left turn then. Either you stop the motion of this will or you move it very slowly and you move this right between much faster. With which what happens is as this rule is moving much faster and this really is stationary, the robots left, that means to turn the robot left, you have to use the right will. And you have to keep the left will stationary. The same applies whatsoever to the direction of going right. They want to go right. You make the right really they're stationary or make little speed. And the left will so that it can take a right then. So similarly, if you want to take a back, you have to run the both of the wheels to the back if you want to rotate or spin the position. You turn this forward, you turn this backward, just use your imagination, what happens, this will keep keeps on turning, they say. At the same time, Israel keeps to turn this thing. With which you will get to spinning, so instead of taking a dark turn, like usually cars do the cars like this, it will usually take a battle and then come like this. But instead, here you have advantage. You can just spin the position of the complete car. Right now, here in this

process, you are able to do all this just because of your law to it, because you have the advantage of Castroville. If it is a case of four wheels, the spinning will be different. So in a formal case, also the same happens, you have to stop, they'll have to do is. And find the right tools to decorate, the similar thing happens.



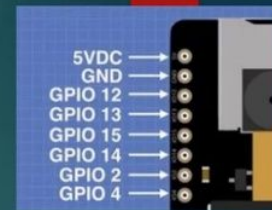
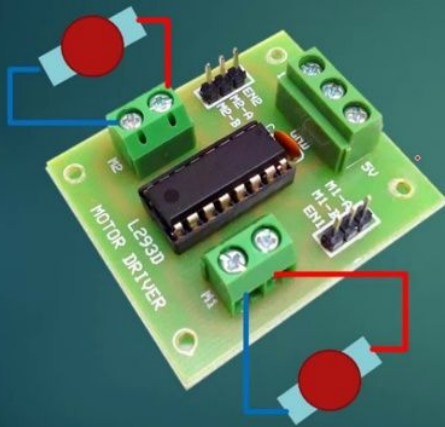
So how old is actually the motor control? This is driving a car. So this is required when you're trying to right the control program. No, but actually how you grade the model itself. For example, let's say we have a firewall model. OK, so this firewall model requires some certain current right? So this cannot be driven directly by your. Debate opens up the country. So what we do is we have a separate physical model driver. And the poverty, Margaret River with the mother. And the power asymmetries in. And so remodeler driveable have certainly some kind of differential in Darlington base are the current multipliers, which will provide the required sufficient current. For remodelers. OK. This will these moderate rebels will help us. So how does this moderate rebel internally works? Usually the moderate rebels are what in principle with the principle called Haertsch Bridge. If you see the hatchet here, you have a switch here, you have a switch here, you have a switch here, you have a switch. This is what the. OK, let us say there are which is like this in the his and if you close switch one and all the

switch for then your positive rail. Gets connected to the positive of the motor and negative rail, gets connected to the negative terminal of the motor with which the common principle is the motor has two polarities. More positive. And motorboating. So at any point of time, if you take your positive terminal to a more positive terminal model, turning clockwise direction. If you reverse the polarity, it will turn into antiglobalization. All right, so here what is happening, if you wanted on the waterfront, I mean forward, so you have to drive it in a positive bias. If you want to turn the motor backward, you have to pass in the legislation for which being able to switch tree and switch to and disable the switch one and switch between what happens, the current version will be reversed. Right. So the motive comes in and the cooperation. So this is the motor drives work. So you will be given control of so you will be given control of the switches in common with both of the switches are connected or the single pin and the switches also control the. Single. Right, you will have a election called A and B, and if you make it one zero, you will have an election. If you make it zero one in one direction, if you make it one one, it will stop like that. If you make it zero zero, nothing happens still in a party traditionally.



So different cases, a positive note in a typical model driver, I see this, which is a replacement European patent system promotion and protection Lobet.

Motor Driver Board & Schematic



ESP32-CAM	Driver	
GPIO12	M1_A	Motor 1 Pin A
GPIO13	M1_B	Motor 1 Pin B
GPIO14	M2_A	Motor 2 Pin A
GPIO15	M2_B	Motor 2 Pin B
+5V	EN_1	Enable Driver1
+5V	EN_2	Enable Driver2

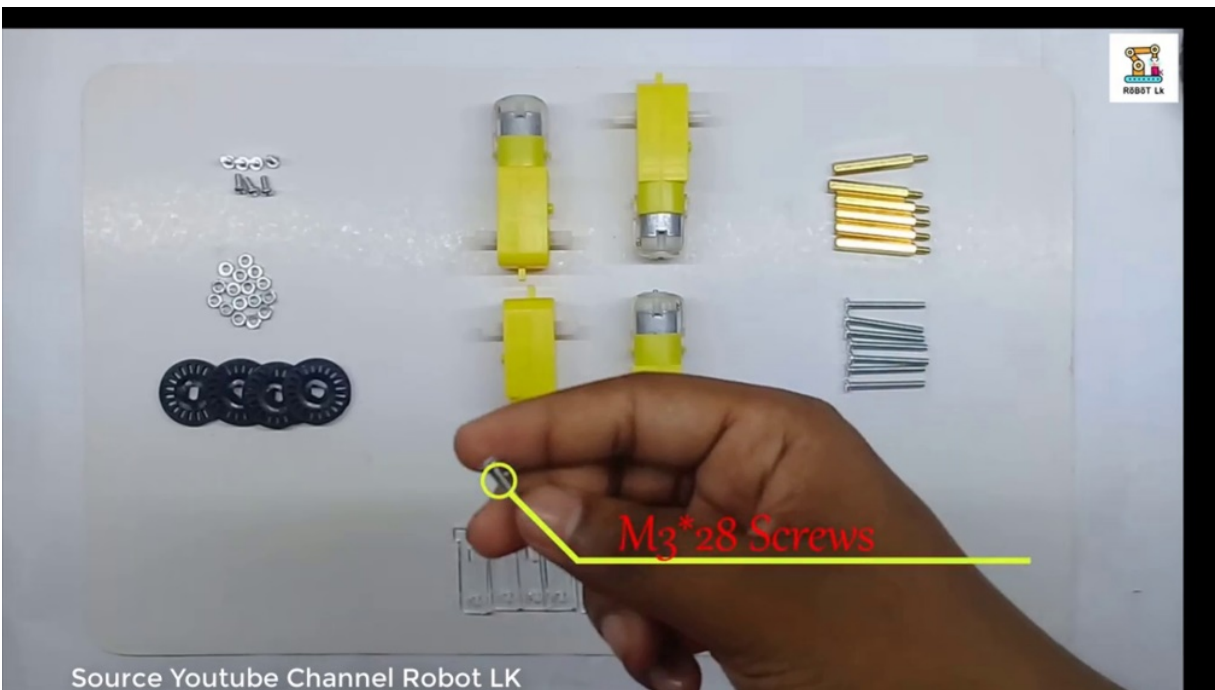
Note : In order to reduce number of Wires from ESP32 CAM, Enables are tied to +5V, motors are driven in simple way than driving through PWM ways, so motors operate at maximum speed

And now coming to the Madrid River border, the schematic, so I use this particular rail to the best model driver, this is very popularly available where you have terminals to terminal blocks to connect the motors so you can cannot see the polarities. You can connect once. If you feel like the motors are being driven into antiglobalization, you have to reverse the polarity. Or you can play with the girls or you can just be open to 10 percent. So here you have the final connection. I want you to connect the power back here and one of them will be grown. And this militant may be ground zero, this may be twelvefold, this is firewalled, so if I want if you are using five Almendras mostly the motors were what we buy, usually for the purpose. Mostly the motors that we buy from reporters that we buy for the purpose are final motors, so you can start buying those motors here. Whatever motors I have shown you in the starting pick, these are usually very well motors. These are priceless DC motors. These are priceless, these images. No, along with that, we will have it and it will be. This one. Support the martyrs, for example, as like we have been the problem earlier, so you can play with enable pins to do the problem. But in our case, I don't want to do problem to the motor control because the number of IEDs which are running from me is petrochem billion more. So I want to avoid that because we also have to promote to be connected for moving the north stepper motors. These are role models to be connected for moving the pedal

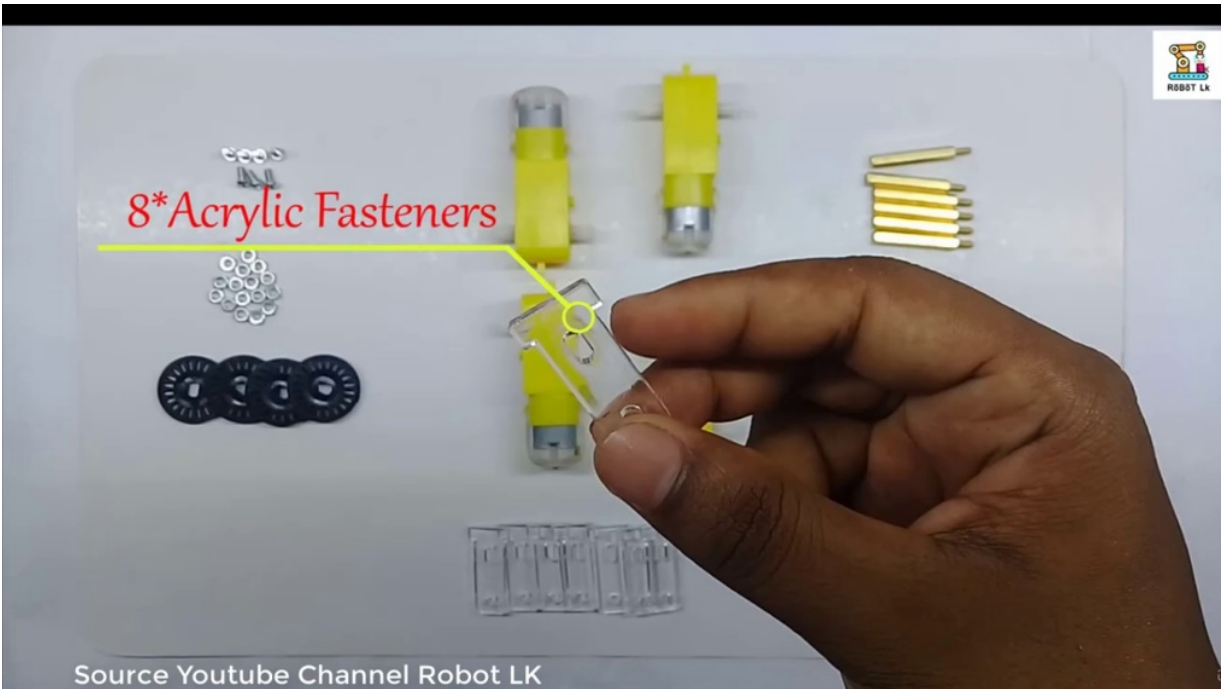
system. Right, so you want to save some beans, I don't want to control the motors using petroleum, so I wish to directly to the control using the motor pins itself and one year corresponds to motor control of motor one is an Tobins correspond to. OK, so you have to connect this in one year. And then when we were 12, 13 and 22, between 14 and 15 of these particular camp. OK, and the enabled pins either you can it. Aldrin or you can simply take a while and connect it to the fugitive. Late. The best way I feel is if you don't have to use a wire and shut it, if you have solar, you straightaway take the wire and shoulder the pin on the bottom side to the fabled. This is very important without any Bilpin being connected to firewalls, the motors won't be driven. Keep it in mind, this is very, very important. OK, this is a hardworking American, hardworking, which to be followed, and we will go ahead seeing how this unbelievable then we will to.

DIY ROBOT CAR

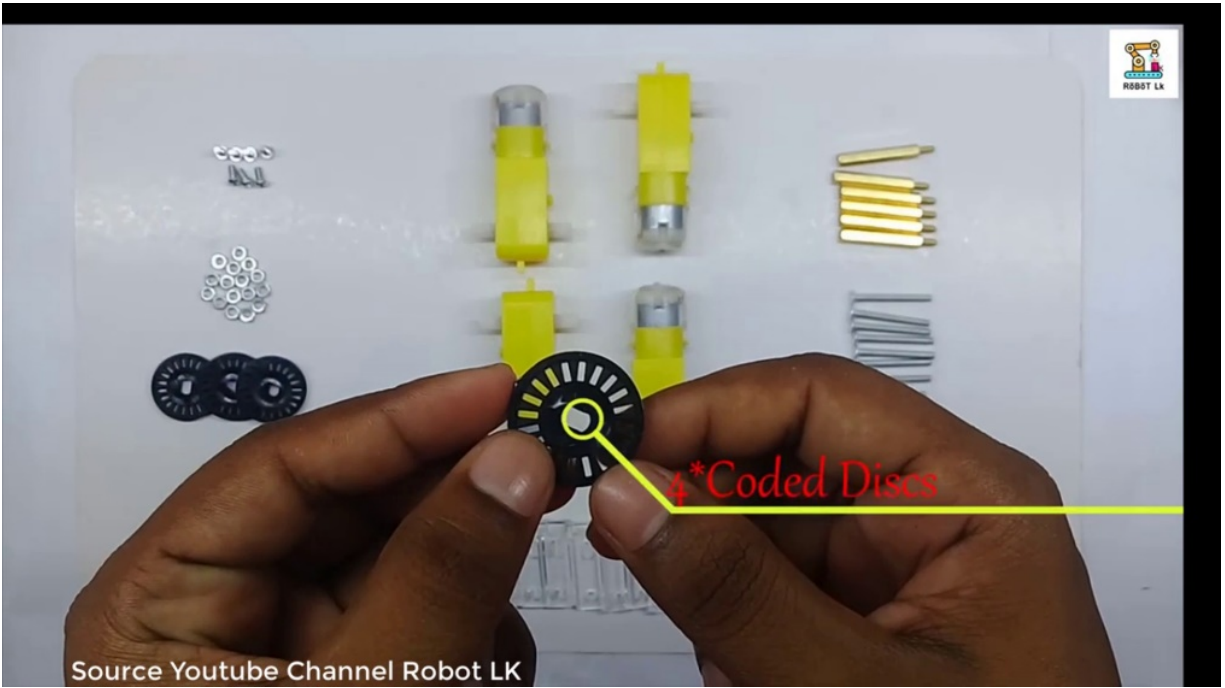
there are few things in common, the interest groups we should be given and also the important part here to assemble the is the fastener, the wheel, the more directly to the body, to the tazi, and some is going to be there.



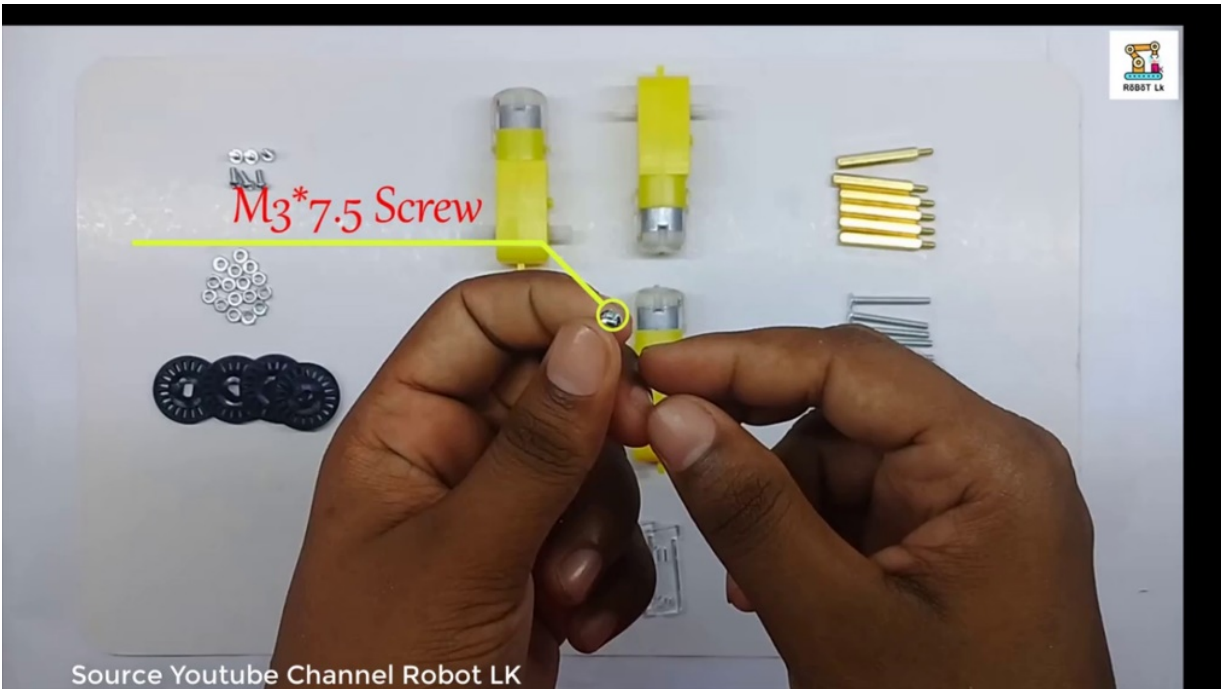
This would be your choice. You can use it or not. And there are these so important feminist groups which will be used to sustain the.



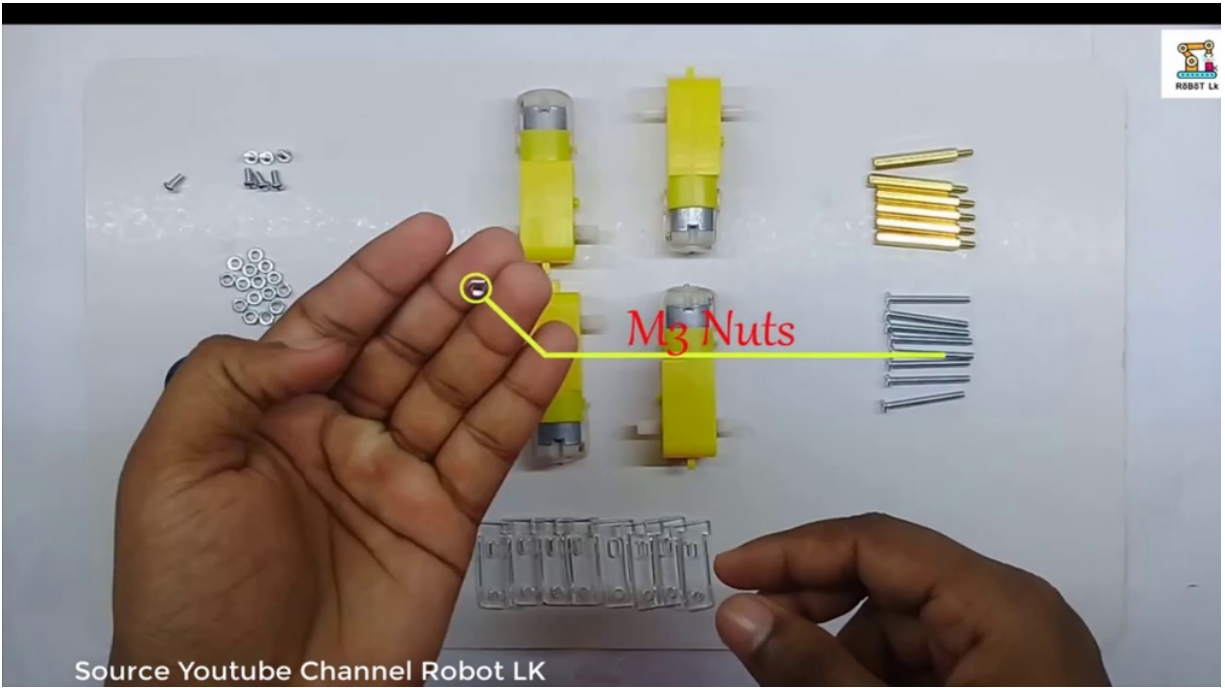
Castroville And then it's so in our case, there are only two General Motors, you can also sometimes you will get troubles with Foreign Motors.



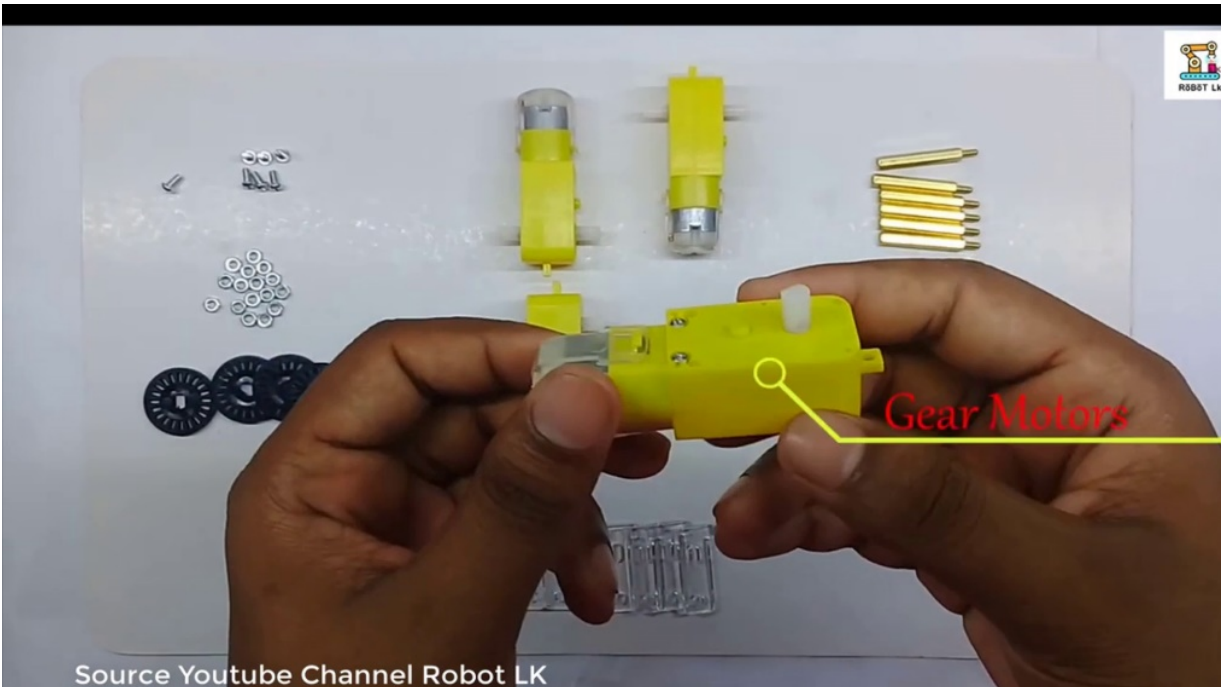
So you have to take this helping slaves and then using that, you have to screw the motor to the test. Tases the what is the body which holds all these motors? Everything.



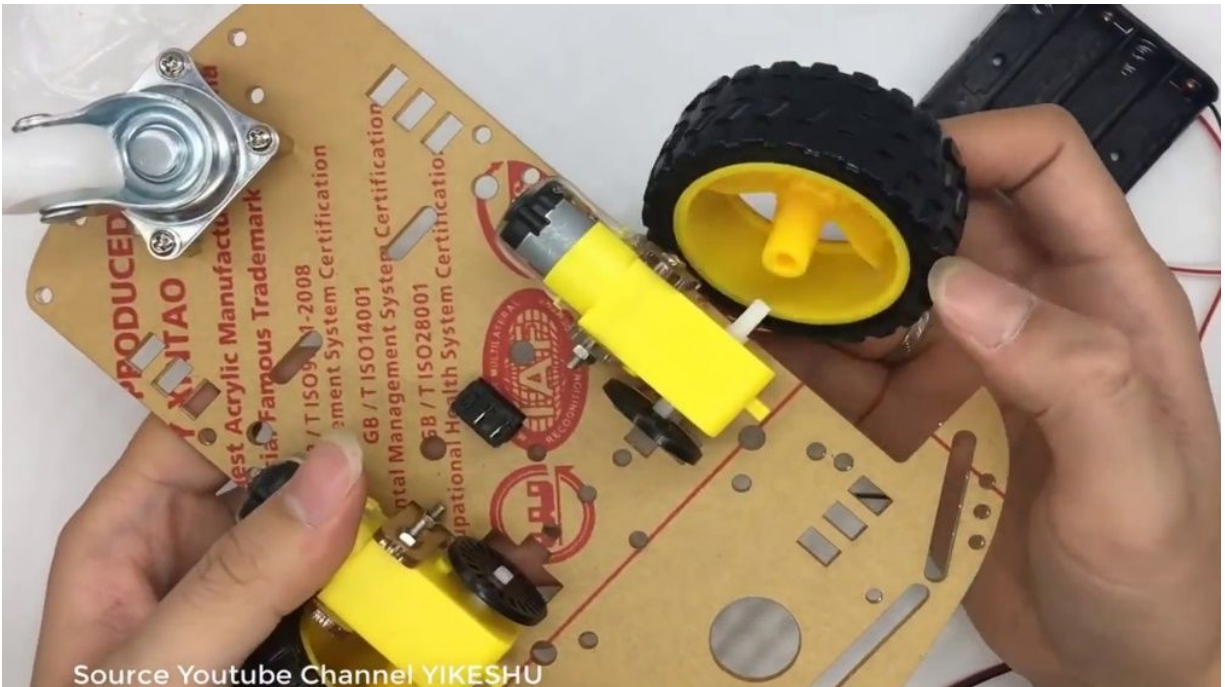
Like every car or even any kind of formula will have to feel like authorities will have this kind of test.



So the charge is the one which holds everything to which we screw it. So, first of all, recycling motors.



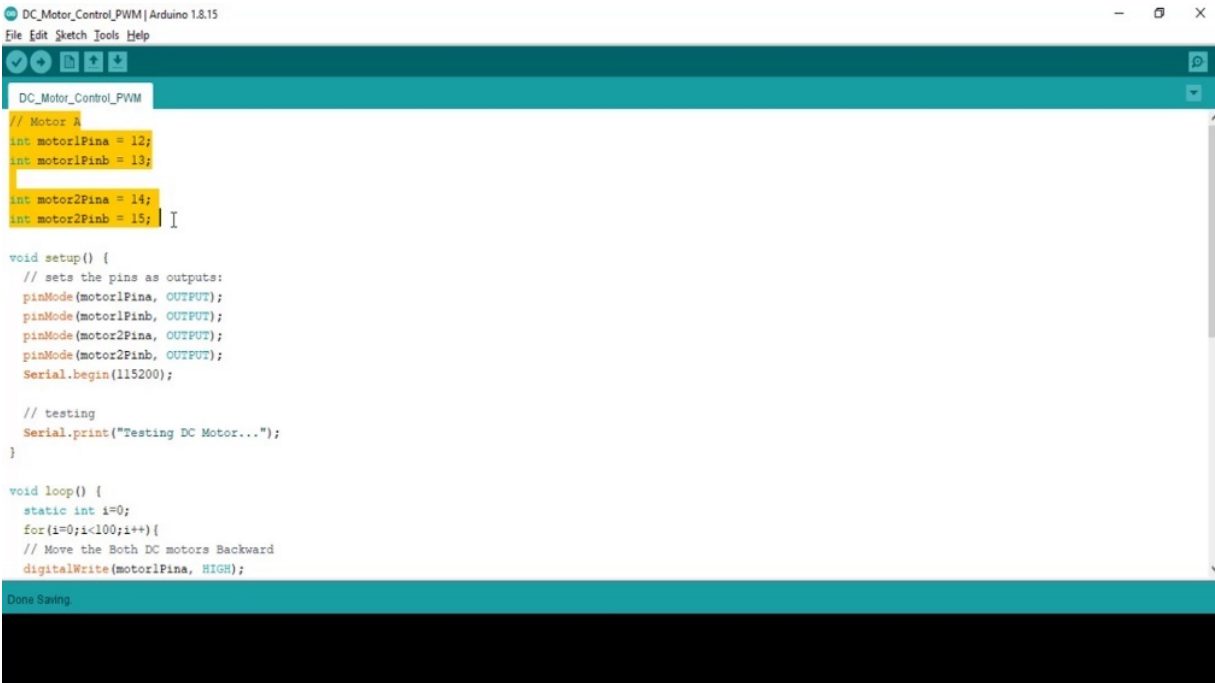
It's not that we don't need. Once these things are done, screwing the motors to the tightening them properly, one more important thing is before actually doing this, but this all the way to the moderates of. Like, you have determinants of motors to themselves, whether you sold the device before screwing. If you are a prince or you can do it afterwards also. I don't comment on that, but most of the modest ones you get also sometimes come with Soledad, who we're just not coming. It's already better to let them. And they go from the halls before the tazi in this area, the reserves, which don't require that speech. So those fast in the. Schools were the Castroville. Not that we are putting the Castroville on one side, on the other side.



We will place over P.A. system or the pantless. So in coming weeks, I'll show you how to assemble the power and how to assemble the mantle, sometimes you will get the same, a complete assemble panel that you can buy that, or if you want to do it by yourself, you can do anything else. You'll save some bucks. Assembled one will be a little costly. So unfastening Castrovile is over and now you're done, you take all the words from the six DVDs before that they go the ways from the model onto the top said. Hope this helps you to realize invaluable.

PROGRAMMING ESP32-CAM TO CONTROL DIY CAR OFFLINE

let's see the court portion of the D.C. motor control, and this is really quite simple.



```
DC_Motor_Control_PWM | Arduino 1.8.15
File Edit Sketch Tools Help

DC_Motor_Control_PWM
// Motor A
int motor1Fina = 12;
int motor1Finb = 13;

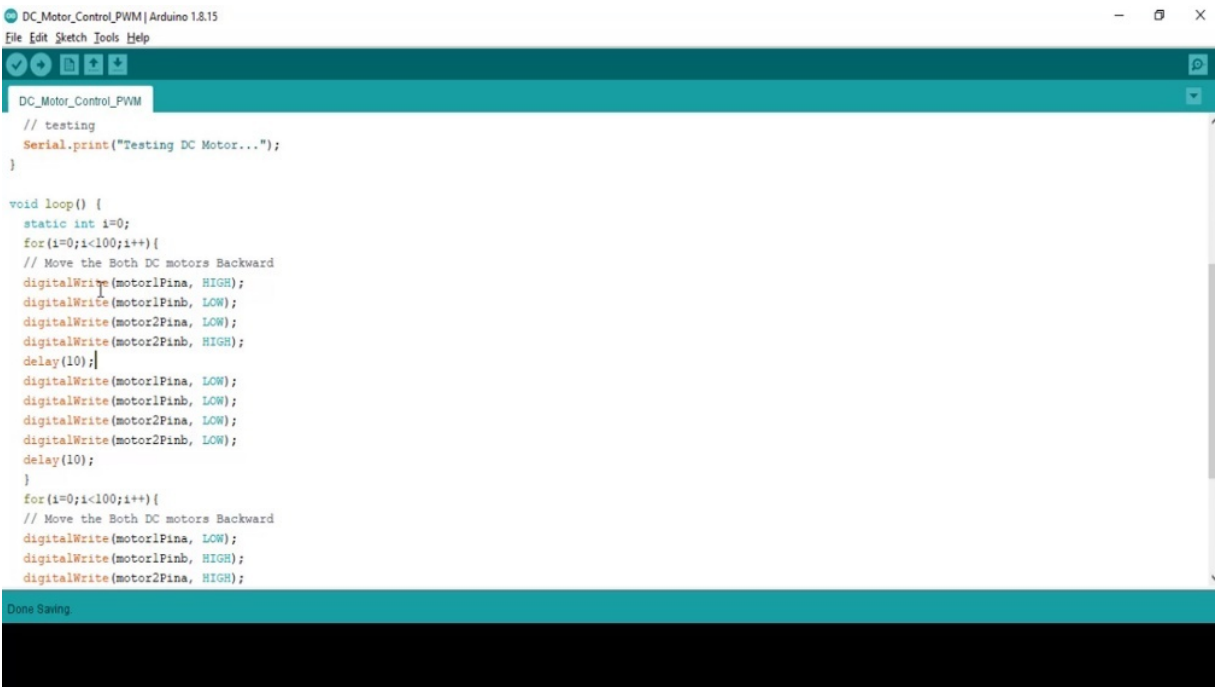
int motor2Fina = 14;
int motor2Finb = 15;

void setup() {
  // sets the pins as outputs:
  pinMode(motor1Fina, OUTPUT);
  pinMode(motor1Finb, OUTPUT);
  pinMode(motor2Fina, OUTPUT);
  pinMode(motor2Finb, OUTPUT);
  Serial.begin(115200);

  // testing
  Serial.print("Testing DC Motor...");
}

void loop() {
  static int i=0;
  for(i=0;i<100;i++){
    // Move the Both DC motors Backward
    digitalWrite(motor1Fina, HIGH);
  }
}
```

First of all, we define the motor pins that we have the opens, 12, 13, 14, 15 in the center. We set all these pincers of buttons and then we do. This will begin just for people just to know where is my control flow of the program.



```
DC_Motor_Control_PWM | Arduino 1.8.15
File Edit Sketch Tools Help

DC_Motor_Control_PWM
// testing
Serial.print("Testing DC Motor...");
}

void loop() {
  static int i=0;
  for(i=0;i<100;i++){
    // Move the Both DC motors Backward
    digitalWrite(motor1Fina, HIGH);
    digitalWrite(motor1Finb, LOW);
    digitalWrite(motor2Fina, LOW);
    digitalWrite(motor2Finb, HIGH);
    delay(10);}
    digitalWrite(motor1Fina, LOW);
    digitalWrite(motor1Finb, LOW);
    digitalWrite(motor2Fina, LOW);
    digitalWrite(motor2Finb, LOW);
    delay(10);
  }
  for(i=0;i<100;i++){
    // Move the Both DC motors Backward
    digitalWrite(motor1Fina, LOW);
    digitalWrite(motor1Finb, HIGH);
    digitalWrite(motor2Fina, HIGH);
  }
}
```

So in the U.S., what we do is we want to move the decimal forward rate, so I make the more the 128. Let us assume some the first and second bill in basically been one, I mean, first Bill immediately and second Bill, I'm a little. Which means I am basking in positive popularity for my mother to move it forward. For example, you were Motor Pinza slabbed or you don't know what is the polarity of them. You can simply change the direction so that the polarity will be reversed. Right, how you can change the motability instead of putting here Binyon, you can put PIN as like here I have my polarities reverse here. So what I did is I made a point to it as lo and behold, be born to be my model, to be as high, which means the second model also is for best means positive. Positive rule of this applies given the positive terminal of the model and negative rate of the supply, which is the ground is given to the negative turn of the model with which this will move forward. They know when it moves forward, then we are not doing the pulse with modulation or the speed control. This moves at the maximum speed consuming as much current it can consume far from that subject right now in order to reduce the speed. What I have done is I am running the models for ten milliseconds and then after immediately I'm switching off the orders and making them all. Then again, I waiting for 10 milliseconds. I'm repeating the four hundred code, which means fifty percent of the time I am moving forward and 50 percent of the time and staying calm or just not moving at all. But this time is very mild for you. Right? So in a sense, I am trying to do a slow, pancit modulation with 50 percent do. Exactly. And the overall time is OK. This Metronet like maybe two or three seconds. It depends on how many instructions are executed in that time.

```
DC_Motor_Control_PWM | Arduino 1.8.15
File Edit Sketch Tools Help

DC_Motor_Control_PWM
digitalWrite(motor2Fina, LOW);
digitalWrite(motor2Finb, HIGH);
delay(10);
digitalWrite(motor1Fina, LOW);
digitalWrite(motor1Finb, LOW);
digitalWrite(motor2Fina, LOW);
digitalWrite(motor2Finb, LOW);
delay(10);
}
for(i=0;i<100;i++){
// Move the Both DC motors Backward
digitalWrite(motor1Fina, LOW);
digitalWrite(motor1Finb, HIGH);
digitalWrite(motor2Fina, HIGH);
digitalWrite(motor2Finb, LOW);
delay(10);
digitalWrite(motor1Fina, LOW);
digitalWrite(motor1Finb, LOW);
digitalWrite(motor2Fina, LOW);
digitalWrite(motor2Finb, LOW);
delay(10);
}
}
}

Done Saving
```

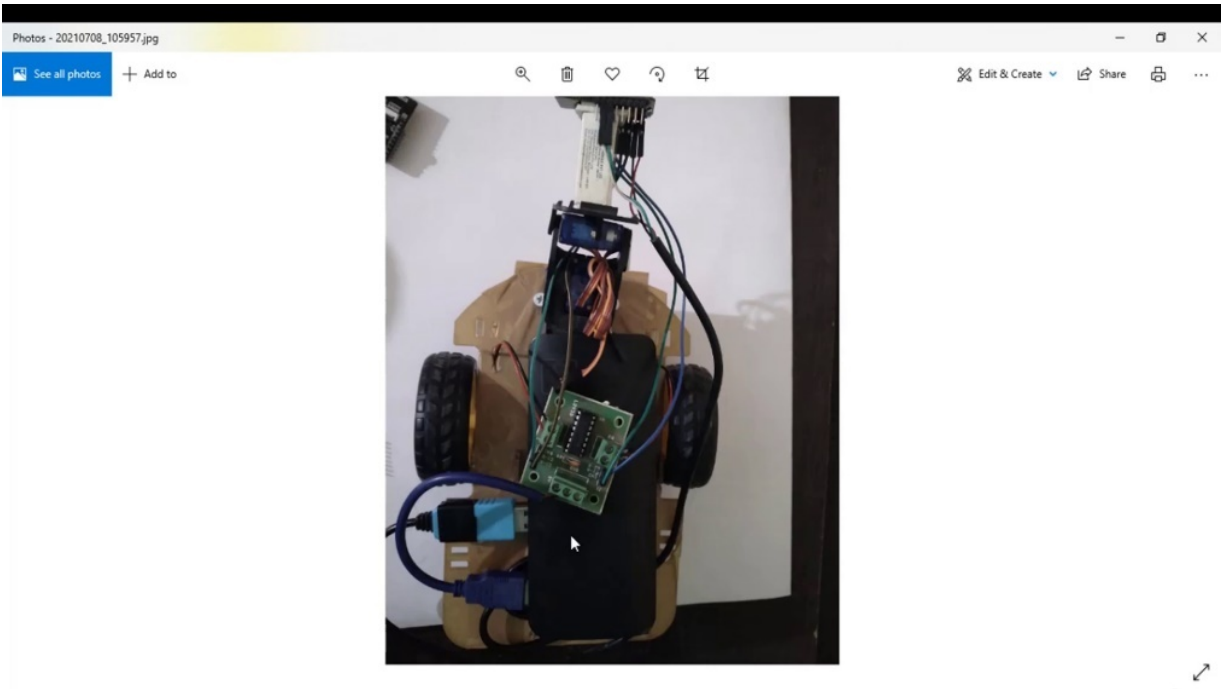
Similarly, this is completely moving forward and I just want to move backward again. So I have reversed the polarity. See here, bindu polarities are changed. Now, these models, these two models, model one and model to move backward. Again, the same principle for a couple of hundred half of the time I am moving backward half of the time and stay low, which means I'm not moving, but this being very low. So you will have intermediate stoppages which will make the motor movement slow. I ask because we don't want to move in a very high speed, we are slowing down the motor speed by airplane, 60 percent of the. This is a way of doing it, so this will keep it simple. So I don't want to do the opposite modulation. I just want to tell you the other way possible. So this explains you very well how to do remote control. So in the coming year, we will move to controlling the mode of spangly.

DEMO AND CAR ASSEMBLY BRIEFED

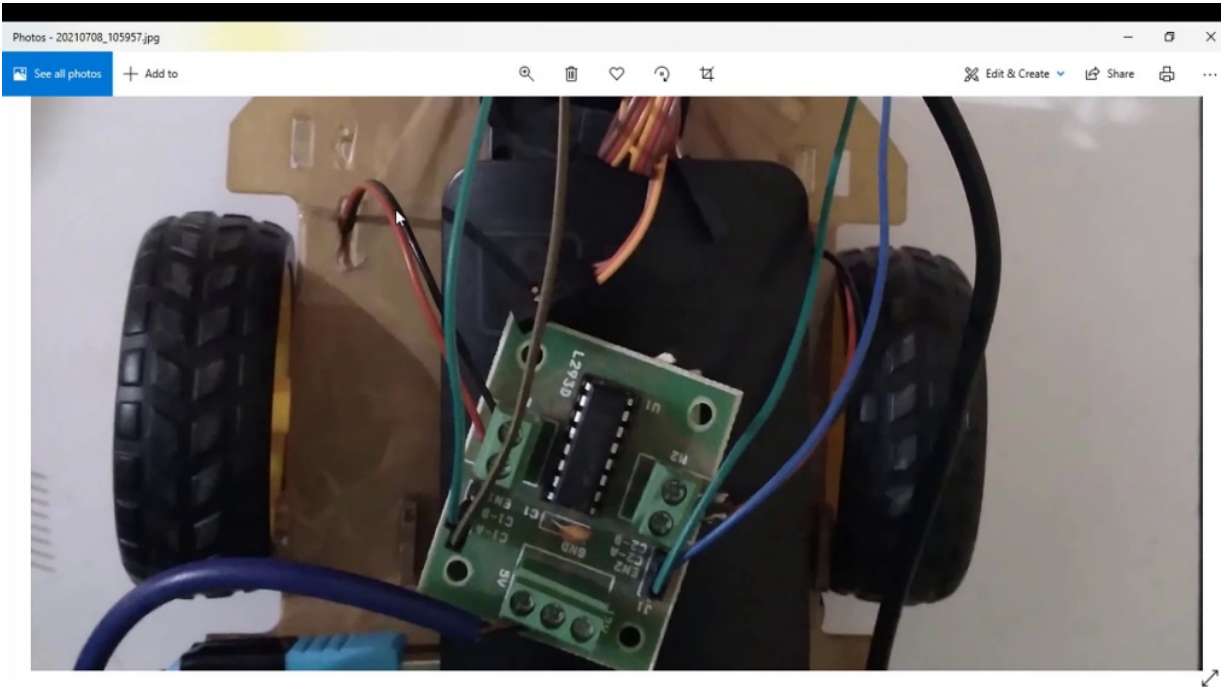
Don't get carried away by this.



This actually happened, the Chattooga completely broken during shipping. So now let me show you the motor terminals. See this? I have plastered all over that. And so it's looking like that. Fine. I have taken or demoted or not and given it to be a minimum two blocks of the module. I can see and these are the driving bills are the Gilpin's which are coming from the hospital. Look at now, if you see in the front repeater system, I've done a small hack, I did not get a block. So I use the CHAKMA block to place in between the years petrochem. I sustained this to the as a friend. So you can look now the wires are coming from the back side of this 12, 13, 14, 15 pins. And I have seen them from the British system to the British system to that. But there is no connection between the system and the motor driveable. And you can also see the top that will block. I have stripped the USB wire. Plus and minus, I never connected that so that you can directly connect the power back there and place the place it all the tazi. So one more important thing is the use of our bank, which has at least three slots. Then I put up our double sided tape on it and I stick it to stick it on the backside of the monitor. You can clearly see I have Solarte enabled us to enable one and two. See?



Now let's have a closer look at the robot assembly.

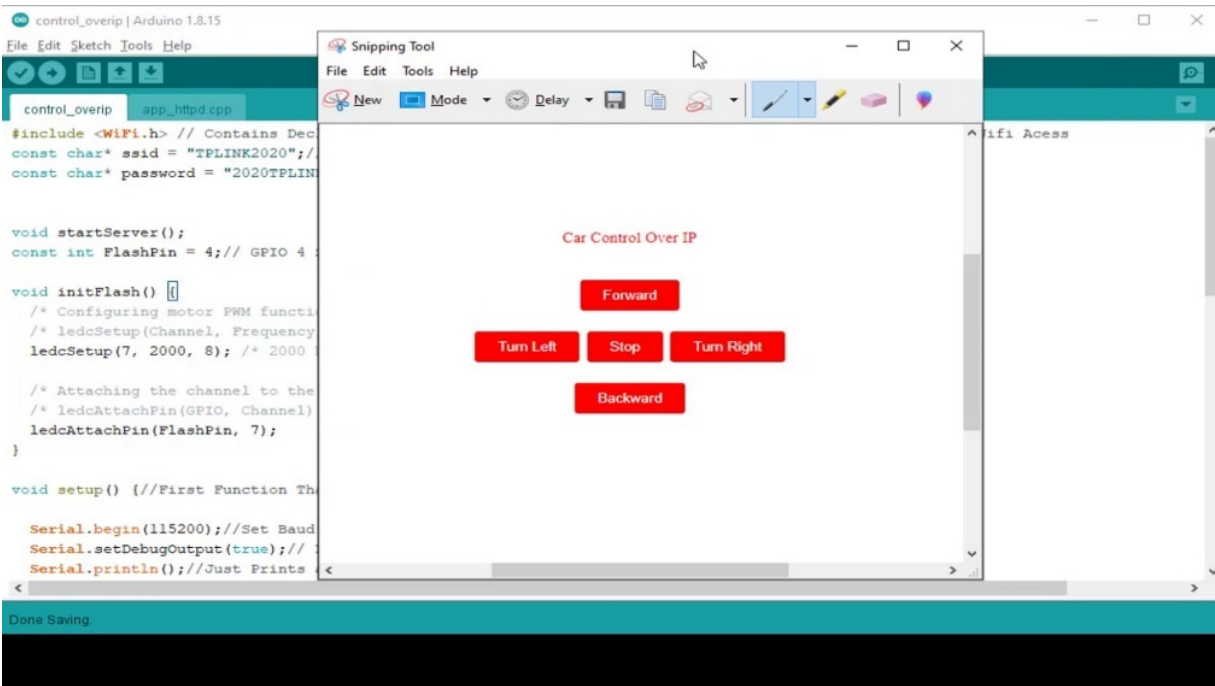


If you see the assembly, clearly the robot wheels from the robot, wheels from the bottom, say you are getting the motor to go into the tunnel blocks late. And these are the pins which are connected to the table, which are. Modern weapons, which are connected on the bottom side of this modern

building, like the if you take is petrochem facing camera, that you can consider them as Saipan's, OK. The and ground and these are the programming skills, the word. Right. And this goes to the same. And actually empowering the better to come from the same is building what, like you can directly plug into the biobank so that the power will power for the board will come from that. And we're is powerful. The driver will control the other people because you have firewalls and have the abandoned current around the three groups. That would be sufficient. If your models are not moving, it means that the current is not sufficient. So now I have left the banking system without connection. I join the next hour, usually connected. I don't by some very heavy or hyphy stuff or really bring something. So I have to do some really terrible level hack where I have to korshak my block in order the sharpness to sharpen the pencil to use this. So I've done I have a double sided tape on the back side of the hospital again. I have put out into the space in between because I did not get a perfect block which fits in between. Usually I do the things like this on and on the other side. I have put the double sided tape onto the panel system topside. This is this so far, the third party resistance, forget the shrapnel and just take your digital camera, kind of the same connections. Just leave it on the robot. Little for this example of the latter example. I'll show you how to fix the British system on the right. So for now, this picture speaks louder so you can take the supply so these people can directly use the word model directly connected to the power back. The other slot will be given giving power to the motor driver. Right. So this would make a simple offline moment of the problem. So in the latter example, what we will do is they will host a similar application, a stable server on the Liesbeth road again and will control the robot using the Web browser or a mobile browser from the mobile phone.

PROGRAMMING ESP32-CAM TO CONTROL THE DIY CAR OVER IP

we will try to create a small thing where it's a duet with duet as like we have built for a little control.



So similarly, we will do the same for driving the robot car over IP. So I call it car control over IP. So you'll have a forward button, backward button left and right and stop. OK, now let's jump to the. So if you see this code you may be familiar with. Yes, this is the same code we have used for Hillary Clinton, if you remember. So where we have common handler and we have a slider button and on and off buttons for the flashily control, we will make some small changes to this and build the RoboCup control application overate. So stocks that were built in the U.S. so there is no flashily here. So I don't need this flaccidity control. Coulter let me. So also, I will not be having any class requirement or anything else that starts or anyways has to happen. So apart from this, the previous program, whatever we have done is offline mode.

```
control_overip | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip $ app_httpd.cpp
#include <WiFi.h> // Contains Declarations of Function Calls related to ESP-IDF and Arduino HookUps for Wifi Access
const char* ssid = "TPLINK2020";// Access Point User Name
const char* password = "2020TPLINK";//Password of the AccessPoint

void startServer();
int motor1Pina = 12;
int motor1Pinb = 13;

int motor2Pina = 14;
int motor2Pinb = 15;

void setup() { //First Function That would Run when any Arduino Based Board Starts

  Serial.begin(115200);//Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true);// Let Debug Messages Appear
  Serial.println();//Just Prints a NewLine
  //sets the pins as outputs:
  pinMode(motor1Pina, OUTPUT);
  pinMode(motor1Pinb, OUTPUT);
  pinMode(motor2Pina, OUTPUT);
  pinMode(motor2Pinb, OUTPUT);
}

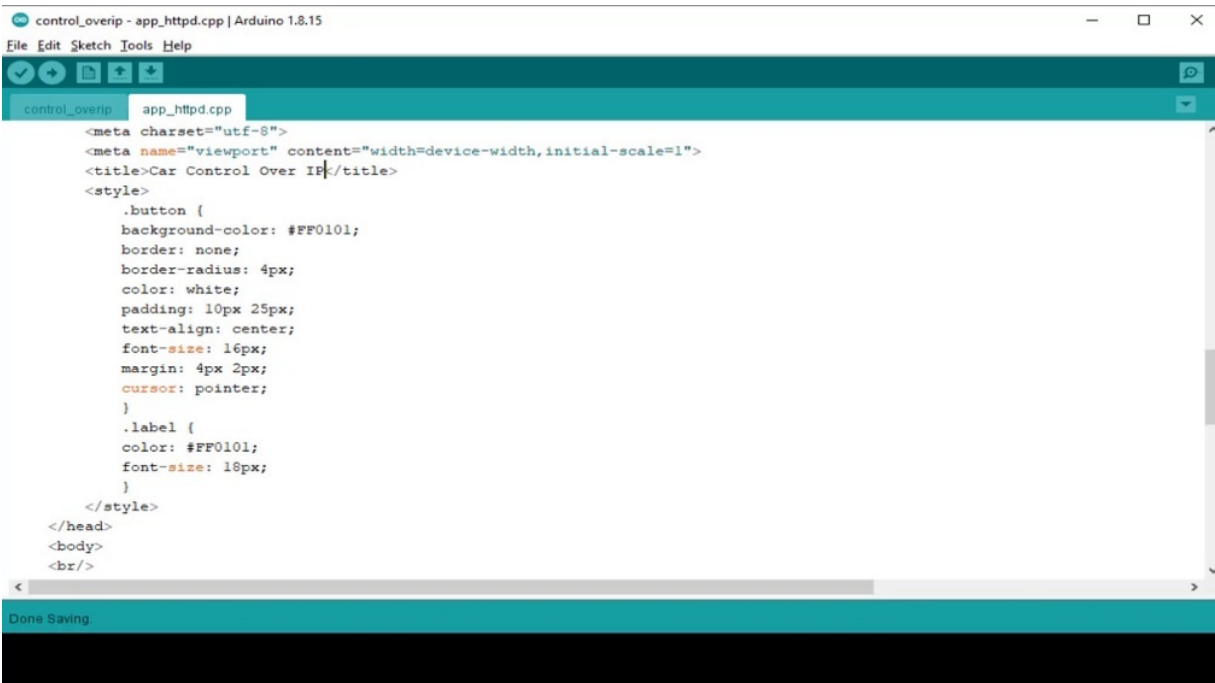
Done Saving.
```

So let us define the model here in this code and in the setup. Let me just make them. Outputs like the murder one bin have been demoted to be. Like the other things I've seen exactly. Nothing will change now when it comes to application. As we jump to the application here, also, we will be having loves.

```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
<body>
<br/>
<div align=center>
<label class="label">Flash</label>
<input type="range" min="0" max="250" value="50" class="slider" id="myRange" onchange="try{fetch(document.location.origin+'/cc
<output>255</output>
<br/>
</div>
<div align=center>
<button class="button" id="on" onclick="fetch(document.location.origin+'/control?var=led&val=255');">ON</button>
<button class="button" id="off" onclick="fetch(document.location.origin+'/control?var=led&val=0');">OFF</button>
</div>
</body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req)
{
  httpd_resp_set_type(req, "text/html");
  return httpd_resp_send(req, (const char*) INDEX_HTML, strlen(INDEX_HTML));
}
void StartServer()
{
Done Saving.
```


Right. Here also, we will be having the Olympics and everything will be done, and as you know, the only change we have to do is we have to clear a few extraversts. Illegal immigration and write the corresponding hand in the corresponding variable and also value it should be coming from that group. So, I mean, the handler in handler puts a earlier. We had this later. So let's review this later that we will not have in Virginia. Right. So let me remove this later. So the Web page becomes much simpler. There is Monero, Slater here, there is only button class will be available because we only have buttons.



```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,initial-scale=1">
<title>Car Control Over IP</title>
<style>
.button {
background-color: #FF0101;
border: none;
border-radius: 4px;
color: white;
padding: 10px 25px;
text-align: center;
font-size: 16px;
margin: 4px 2px;
cursor: pointer;
}
.label {
color: #FF0101;
font-size: 18px;
}
</style>
</head>
<body>
<br/>
```

Done Saving

And instead of it being absolutely controlled, it will be car control over IP. Right. Let me know what extra things might happen, so the highest bid starts like this car can be anywhere, but unless I know so we don't have this that let me remove this label. So don't do.

```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
color: #FF0101;
font-size: 18px;
}
</style>
</head>
<body>
<br/>
<div align=center>
<button class="button" id="forward" onclick="fetch(document.location.origin+'/control?var=ccoip&val=1');" >Forward</button>
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" onclick="fetch(document.location.origin+'/control?var=ccoip&val=2');" >Turn Left</button>
<button class="button" id="stop" onclick="fetch(document.location.origin+'/control?var=ccoip&val=3');" >Stop</button>
<button class="button" id="turnright" onclick="fetch(document.location.origin+'/control?var=ccoip&val=4');" >Turn Right</button>
</div>
<br/>
<div align=center>
<button class="button" id="backward" onclick="fetch(document.location.origin+'/control?var=ccoip&val=5');" >Backward</button>
</div>
</body>
</html>
)rawliteral";
<
```

Done Saving

Yeah, no, the on and off buttons, which are already present, so there is no need of that. So let me put my four buttons on suburban school where I have five buttons. One is having any forward address having left stop don't backward. These are these are just for our ID. But the text, which is going to be displayed is between this is forward. Don't let. Stop the light backward. I'm following this. So it's your choice to put any other buttons also. So as like the previous one, there is an action on it, on what should happen. I have to face the current origin, which is the word control, because we are doing it over the control interface and because this is a name we have given for the candidate what we have defined in the command center and things like that. So we are following that. And suppose you give cordiale, it should be clear here.

```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
}
</style>
</head>
<body>
<br/>
<div align=center>
<button class="button" id="forward" onclick="fetch(document.location.origin+'/control?var=ccoip&val=1');" >Forward</button>
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" onclick="fetch(document.location.origin+'/control?var=ccoip&val=2');">Turn Left</button>
<button class="button" id="stop" onclick="fetch(document.location.origin+'/control?var=ccoip&val=3');">Stop</button>
<button class="button" id="turnright" onclick="fetch(document.location.origin+'/control?var=ccoip&val=4');">Turn Right</button>
</div>
<br/>
<div align=center>
<button class="button" id="backward" onclick="fetch(document.location.origin+'/control?var=ccoip&val=5');">Backward</button>
</div>
</body>
</html>
)rawliteral";

static esp_err_t index_handler(httpd_req_t *req)
```

OK, so in the control I to which we are and vehicle. So I have to change the variable name here. Earlier we were using something else, I think some values to justify your zero based on the control right here. What we use is a different variable car controller, IPCC or IP. I was using a short form so far forward. I was hearing that while you call one, that means when reality one comes, the guard should move left. Sorry, forward. OK, four left turn left. I use number two for stop at three four right this forward. Four backward it is right now when somebody tries to press these buttons. These are the values which are in which you are going to get in the and like.

```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
int val = atoi(value);
Serial.print("Variable : ");
Serial.print(variable);
Serial.print(" With Value : ");
Serial.print(val);

httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, NULL, 0);
}
static const char PROGMEM INDEX_HTML[] = R"rawliteral(
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Car Control Over IP</title>
    <style>
      .button {
        background-color: #FF0101;
        border: none;
        border-radius: 4px;

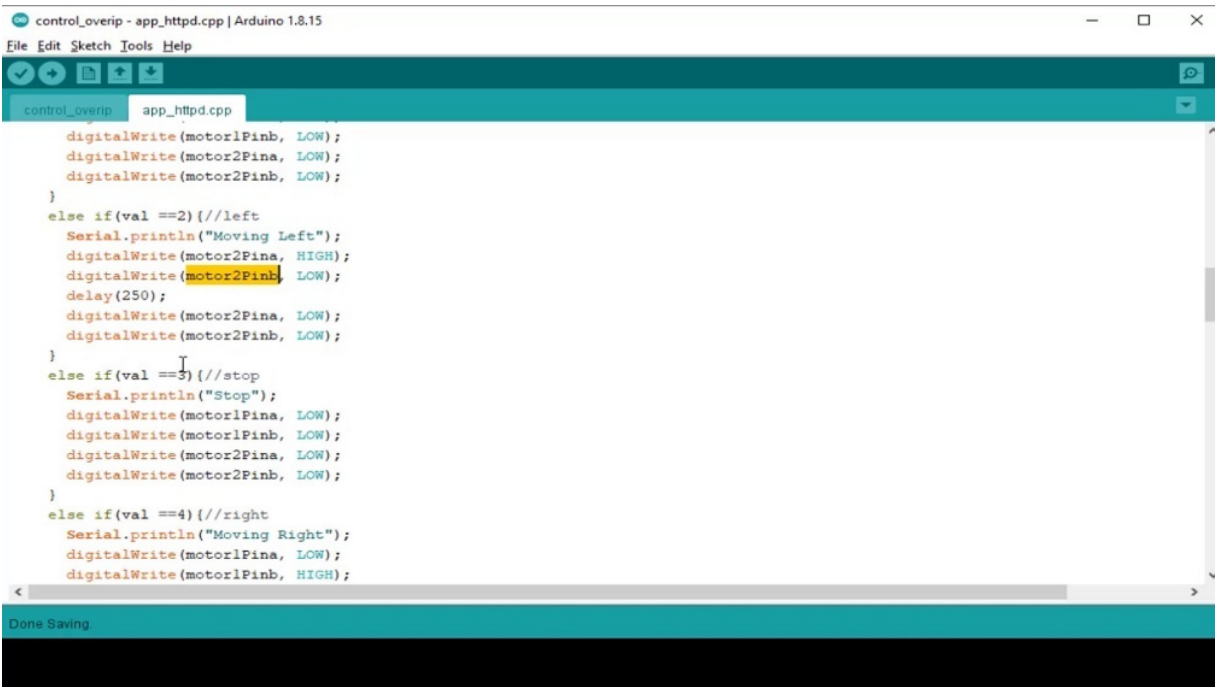
```

Now, in the command center, if you come earlier, we were handling. We are taking the val. And also, the variable here is a flight control will. Let me remove this country and place our motor control from our previous call simply so in our previous call, we have already written how to move forward, how to move it backward.

```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
} else {
  httpd_resp_send_404(req);
  return ESP_FAIL;
}

int val = atoi(value);
Serial.print("Variable : ");
Serial.print(variable);
Serial.print(" With Value : ");
Serial.print(val);
Serial.print(val);
if (!strcmp(variable, "ofoip"))
{
  if (val == 1) { //forward
    Serial.println("Moving Forward");
    digitalWrite(motor1Pina, LOW);
    digitalWrite(motor1Pinb, HIGH);
    digitalWrite(motor2Pina, HIGH);
    digitalWrite(motor2Pinb, LOW);
    delay(250);
    Serial.println("Stop");
    digitalWrite(motor1Pina, LOW);
    digitalWrite(motor1Pinb, LOW);
    digitalWrite(motor2Pina, LOW);
  }
}
```

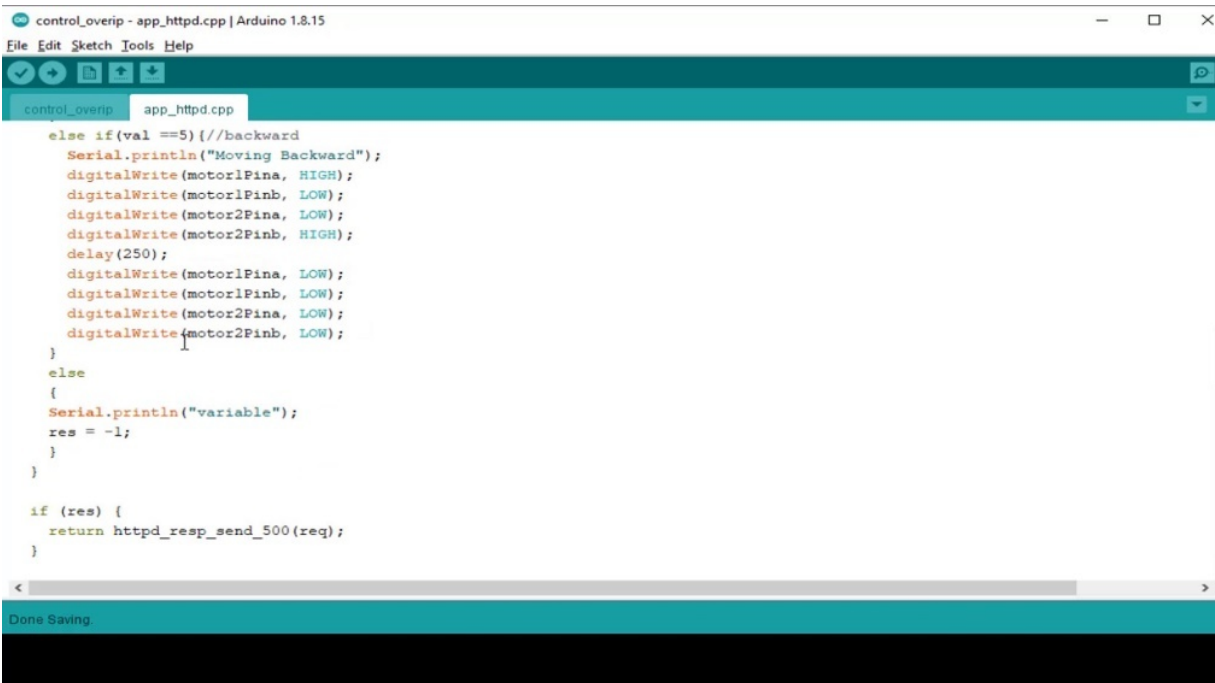
I'm just bringing my code one second. We are not allowed to see whether the variable we got is quality or not control over a Belmont, once that is true, then we have a possibility of forward as value being one and possibility of left, possibility of rain stop and backward so far forward. What are we doing so far forward? Simply, we are moving the left motor and also right motor and clockwise direction. It depends on the orientation of your vehicle, which direction you decide. So it goes forward for a forward base and high, high and low in my case. OK, then after I provide a little delay and then stop them on this, right. So please note that when somebody gives you forward command, you should not be continuously going in for direction. Rather, I look forward for a while and then stop even here until I go forward for 250 milliseconds, sort of left for 250 milliseconds and then stop so far left. I have already told you the left is already stuck. So you have to move the right one, which is Michael's motor to like, I'm just moving.



```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
digitalWrite(motor1Pinb, LOW);
digitalWrite(motor2Pina, LOW);
digitalWrite(motor2Pinb, LOW);
}
else if(val ==2){//left
Serial.println("Moving Left");
digitalWrite(motor2Pina, HIGH);
digitalWrite(motor2Pinb, LOW);
delay(250);
digitalWrite(motor2Pina, LOW);
digitalWrite(motor2Pinb, LOW);
}
else if(val ==3){//stop
Serial.println("Stop");
digitalWrite(motor1Pina, LOW);
digitalWrite(motor1Pinb, LOW);
digitalWrite(motor2Pina, LOW);
digitalWrite(motor2Pinb, LOW);
}
else if(val ==4){//right
Serial.println("Moving Right");
digitalWrite(motor1Pina, LOW);
digitalWrite(motor1Pinb, HIGH);
}
Done Saving
```

So you have to take this by your own. So you have particular terminals. Which direction is actually your car is moving once you give it right to this school. And then if you observe, is there any for example, if you're trying to move your motor to the left, a legacy, it is moving backward, then you have to reverse the quality of the model instead of if you are trained for the left

right motors, sorry if it is taking a right and then you have to change the motor in the court itself that is efficient. So you have to do the swapping. This you will get to once you are actually trying to see whether the functionality is moving on or you can just grab the bits that is OK and stopping is when they stop coming comes to you, you have to just use or disable all the motors and make all the same. When you are moving right. You have to stop the right motor and move the left. But similarly, like to tell right and then stop. So when I was being asked to move backward, I will rotate the models in Antiglobalization for 250 milliseconds and then I stop.



```
control_overip - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
control_overip app_httpd.cpp
else if(val ==5){//backward
  Serial.println("Moving Backward");
  digitalWrite(motor1Pina, HIGH);
  digitalWrite(motor1Pinb, LOW);
  digitalWrite(motor2Pina, LOW);
  digitalWrite(motor2Pinb, HIGH);
  delay(250);
  digitalWrite(motor1Pina, LOW);
  digitalWrite(motor1Pinb, LOW);
  digitalWrite(motor2Pina, LOW);
  digitalWrite(motor2Pinb, LOW);
}
else
{
  Serial.println("variable");
  res = -1;
}
}

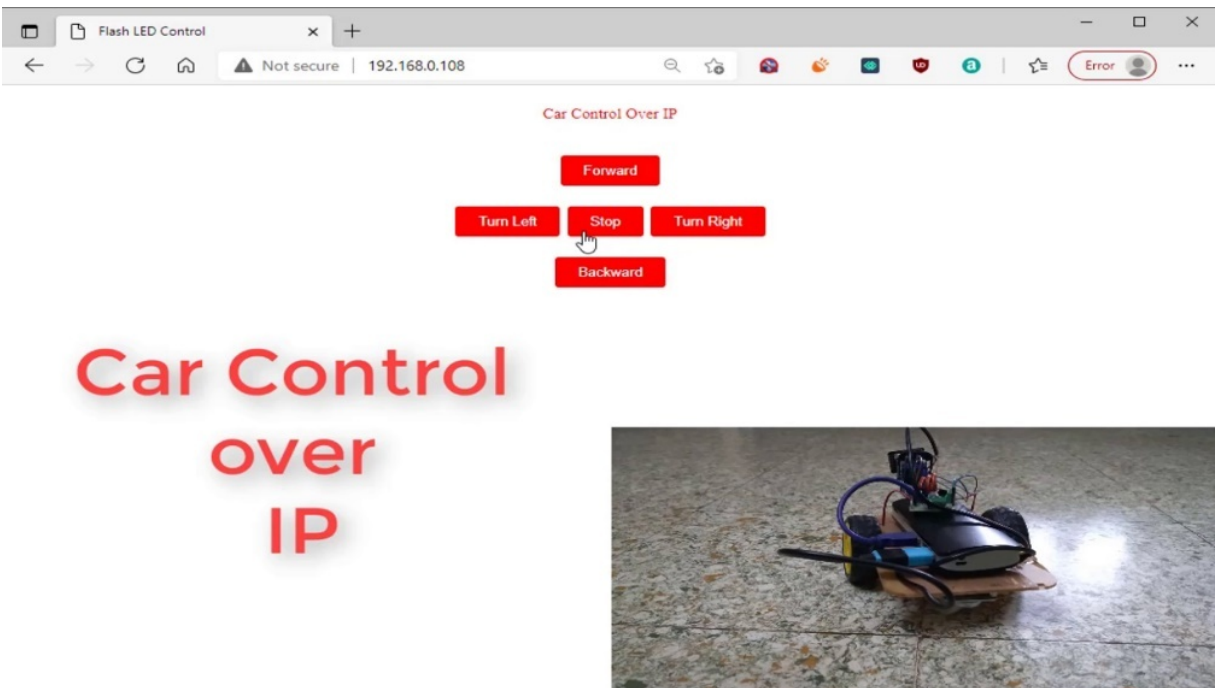
if (res) {
  return httpd_resp_send_500(req);
}
```

Done Saving

Supposedly in unvalidated variable comes, I give a result minus one, and if the result is minus one, then I go for sending the response of whatever political system.

DEMONSTRATION

so I will move the robot to the forward signal that we have opened this Web page on our PC because we are using the additional buttons as unclick, not on touch.

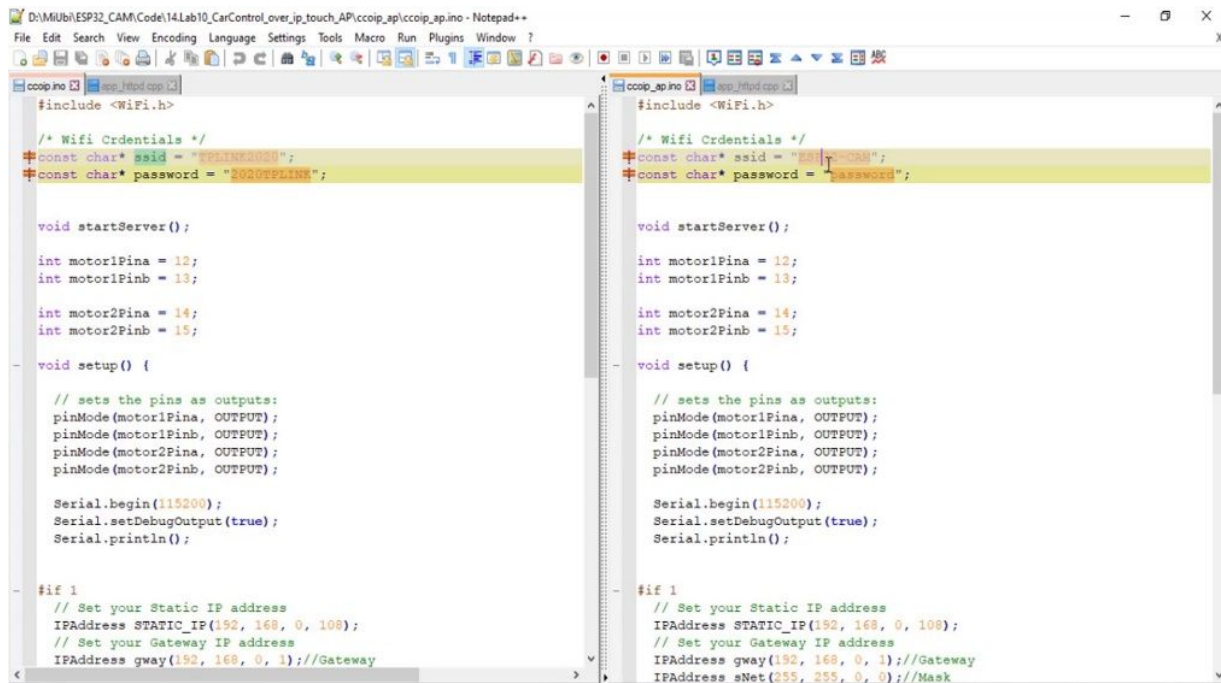


The image shows a web browser window with the title 'Flash LED Control' and the address bar displaying '192.168.0.108'. The page content is titled 'Car Control Over IP' and features five red buttons: 'Forward', 'Turn Left', 'Stop', 'Turn Right', and 'Backward'. A mouse cursor is hovering over the 'Stop' button. Below the browser window, the text 'Car Control over IP' is displayed in a large, red, stylized font. To the right of this text is a photograph of a small, custom-built robot on a tiled floor. The robot has a black top deck, a camera lens, and various electronic components and wires visible on its back.

So if you open this on a mobile phone, this one may not be working. This is forward. Backward. They were right. Forward by putting. Hope you can build some applications around this and in this current project, whatever you're seeing, the yes, but the look at is actually mounted on the back end. So for this lecture, you can simply leave it on the car assembly itself at one create really an issue.

ROBOT CAR CONTROL - ESP32 CAM AS ACCESS POINT

They will do a very slight difference or slight change from our previous call, for example. This is our car controller right on the left side. This is a new code which I'm trying to develop, which is creating an access point and trying to control the robot using that access point. We're going to over the phone. So in essence, I am trying to create a remote control using all the Wi-Fi network. You don't need to connect to the law. You can connect to the hospital to get mobility and try to control little.



```
ccoop.ino | esp32/ssid.cop | cooip_ap.ino | esp32/ssid.cop |
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
D:\Mikub\ESP32_CAM\Code\14Lab10_CarControl_over_ip_touch_API\ccoop_ap\ccoop_ap.ino - Notepad++
#include <WiFi.h>

/* Wifi Crdentials */
const char* ssid = "ESP32-CAM";
const char* password = "2020EPLINK";

void startServer();

int motor1Pina = 12;
int motor1Pinb = 13;

int motor2Pina = 14;
int motor2Pinb = 15;

void setup() {

// sets the pins as outputs:
pinMode(motor1Pina, OUTPUT);
pinMode(motor1Pinb, OUTPUT);
pinMode(motor2Pina, OUTPUT);
pinMode(motor2Pinb, OUTPUT);

Serial.begin(115200);
Serial.setDebugOutput(true);
Serial.println();

#if 1
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1);//Gateway
#endif
}

ccoop_ap.ino | esp32/ssid.cop |
#include <WiFi.h>

/* Wifi Crdentials */
const char* ssid = "ESP32-CAM";
const char* password = "password";

void startServer();

int motor1Pina = 12;
int motor1Pinb = 13;

int motor2Pina = 14;
int motor2Pinb = 15;

void setup() {

// sets the pins as outputs:
pinMode(motor1Pina, OUTPUT);
pinMode(motor1Pinb, OUTPUT);
pinMode(motor2Pina, OUTPUT);
pinMode(motor2Pinb, OUTPUT);

Serial.begin(115200);
Serial.setDebugOutput(true);
Serial.println();

#if 1
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1);//Gateway
IPAddress sNet(255, 255, 0, 0);//Mask
#endif
}
```

So the only difference what we have done is I have very clear up to the Texas Point Project, which we have been very initially and also our previous project, QIP. So what are the details? They're very, very minor things. So instead of giving your Rhodesia's alien buzzword here, you give your own name and password for the module and then after the static electricity is also seen.


```
pinMode(motor2Pinb, OUTPUT);

Serial.begin(115200);
Serial.setDebugOutput(true);
Serial.println();

#if 1
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1);//Gateway
IPAddress sNet(255, 255, 0, 0);//Mask
IPAddress pDNS(0, 0, 8, 8);//Primary DNS
IPAddress sDNS(0, 0, 4, 4);//Secondary DNS

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) {
  Serial.println("Configuration with Static IP Failed");
}
#endif
/* Connecting to WiFi */
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startServer();/* Starting camera server */

Serial.print(WiFi.localIP());
```

But while configuring instead of using that concept, you have to use software which creates makes the hospital can behave as an access point.

ASSEMBLING PAN AND TILT SYSTEM

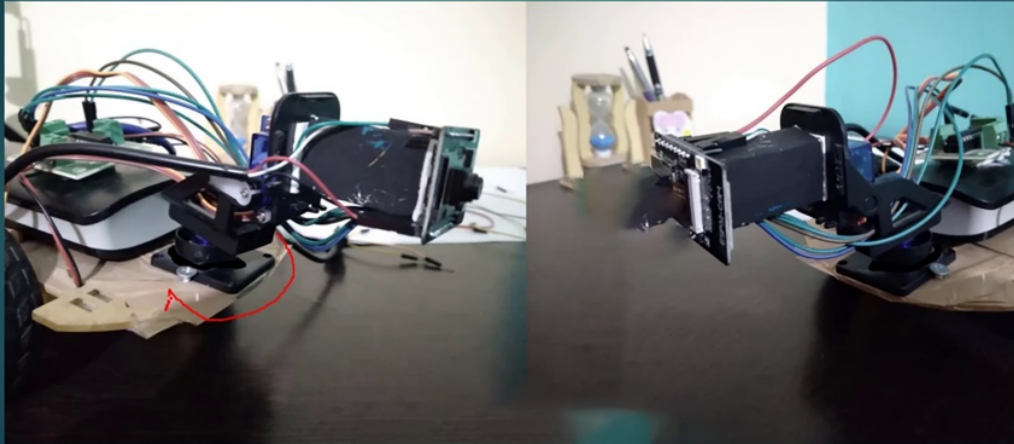
if your panel there simply doesn't look like this and if there's some other alternative, you have to find out from the assembly manual of that particular system how to assemble it.

So going forward, usually the typical kit contains two role models. Now, in my case, it is a 90 and based bracket to say brackets and the top bracket and respected more base are the side brackets holding screws and also a few couplets which come from the motor shaft shaft. And this is the one which extracts out from the motor the rotating position, I mean, the rotating output motion which is connected to the base so that holding it the motor or the complete assembly rotates. Next, the passingly. This is the. Let me take this is the. This, but first one, these are the same brackets and this is the top bracket. So the very first thing is you have to take the best part and take the capital city, which is in a staff format like this, and then basically when it

comes, some records will have a out here. Exactly here. Some won't help. So we have to cut exactly. So that it fits or goes into the circuit. After cutting, it becomes something like this, make sure where you're cutting your balls. I mean, symmetry, that means your balls have to come exactly up here so that you can fix screw holes from the bottom. After this, it becomes like this. So from the bottom side, you have to screw the Kappler to the base.

Now, the second portion is you have to couple your model. I was speaking about the shaft. This is the shaft. And this should be fixed in this position. So we drop in there and fix the model from the back. So using a scoop and this is a way we are actually coupling our model to the base earlier. We are coupling the copper accessory to the base. Right. Then after place, the side brackets from the sides and there are groups here and the mortar also exactly goes into the grooves since properly and assemble the left bracket and they'll brackets properly. And no wonder both the brackets using a school drop this helicopter to this age bracket, to the right side, Muslims and the screw the cobbler to the body using a school as like we are done with the bottom and then take the top bracket and screw the. Top Model, which is the building model, the Baltimore Police Department, which fixes to the body and build models, the top one, so no detailed model wanted to fix it to the top bracket, drop it into the bottom like here. Now, this particular shaft of the top model goes into the cable. And we have to hold it from the side using it screw, this is the last step that will. Once that is done, dismantled assembly, this can move in 180 degrees from zero being here and 180 being here. And this particular base assembly, if I hear this a basis for start from zero and it can added up to 180, so zero is here, the one is here. So if you want to banned motion, you have to move this site. This will be static. If you want to take the motion, you have to move this. Like.

FIXING ESP32 CAM TO PAN TILT ASSEMBLY



I know this particular band motion is performed in this way until the motion is performed in this. Oh, I have fixed one camouflaged shark, not block this with two double sided dates. And I fixed it. I will say the tapes to the hospital. This is the only block I could get exactly. Which fits into the hospital. You can find your own alternating blocks. It can be some wooden block or a plastic block or any legal block, whatever it can be, because without this we cannot create a separation or space to bring our wires to the camera model. One more important step is to outlaw 60 percent of assembly to the robot assembly. You will find in the forward portion of the backward portion to screws where you can fix the pedal assembly like this to the robot with this complete our complete assembly to the robot car to let us have a relook of the assembly. This is the baseplate and design plates. And then we have the public. So when we take the baseplate also take these couples are deeply assembly accessories, which are very crucial for this assembly, and then fix the capital to the base, which is in a star shape and make sure wildcatting is the whole appropriate so that we have the school once the cutting is over, scraped from the bottom and fix it properly to the base, the base, Kappler, the base plate. This discolor will be again bonded with the base model, which is the band model. So the Scrooged by short shape that we're seeing in which we will drop the shaft, that means the extruding portion of the model. Into the base and then fix the school from the bottom

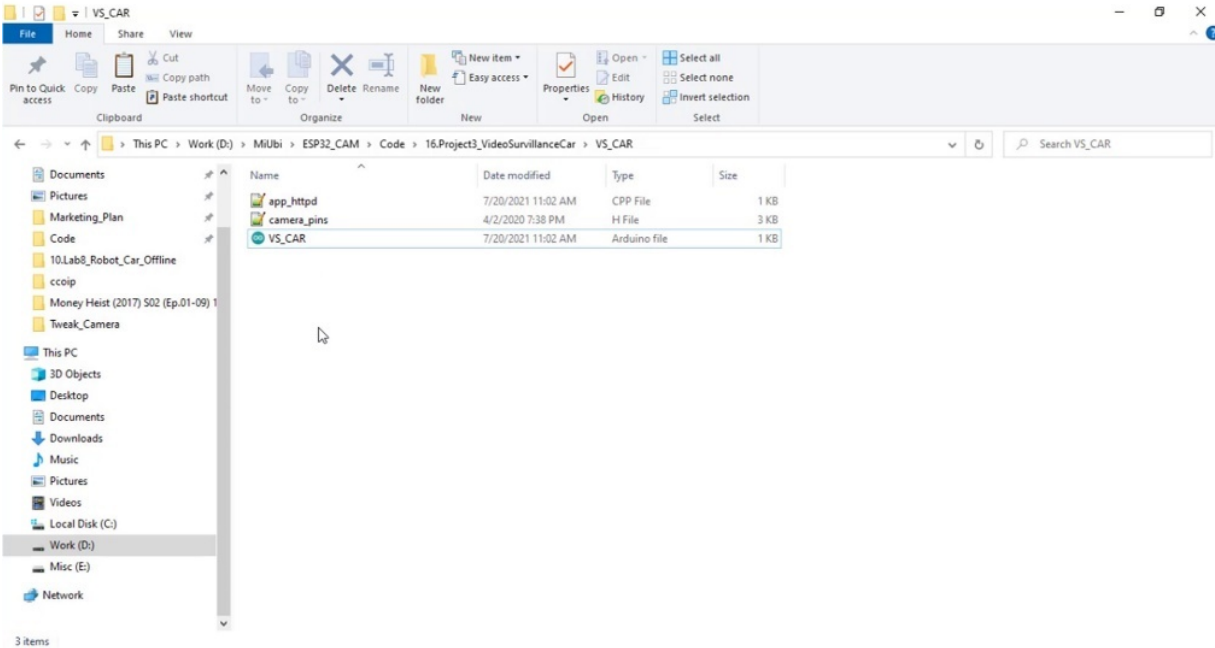
side so that the mortar perfectly sticks to the base without any problem. Once this is complete, we then take the side brackets and drop them exactly into the grooves properly. And screw the side brackets properly. So we attach the same brackets and between these records, I like holding these higher brackets together, we put the screws and screw them properly, noticeably greater volume probably around the bottom model, which is the band model. Then after take this one more complete accessory for the top model, which is the tilt model in the same category so that it falls properly in the shape. And then from the other side, you screwed the cobbler to the base as like we are born with the bottom of the. Sorry, bottom line. So now this top cabinet is also scrutiny body properly. Now the cabinet needs to be considered with the panel before that, the top person. I mean, the top bracket. We fix it to the Panmure properly. Once the top bracket is fixed to the band model properly, we take this complete assembly, the assembly, and then drop it into the bottom brackets and the bottom assembly, which fixes it properly. Now, this will make the complete with the system.



The final step is you have to take one more of a screw and tighten the top model, which is to build the model properly to the bracket which makes this complete bank and assembly complete.

CODING

the building of the project of project surveillance car, you can just copy your previous project, which is project to.



We are the only for a limited you know, if a name is changed, so accordingly to I live for the moment or two years, it is a little scary. So, no, what is we have to fill in the car. So let me open it. INAUDIBLE. So the application also I have nullified and they are also nullified. So the camera means that it would remain safe.

```
#define CAMERA_MODEL_AI_THINKER // Your Board Info

#include "esp_camera.h" // Contains Declarations of Function Calls related
#include "camera_pins.h" // Defines the GPIO Pin Numbers of ESP32 for
void startCameraServer(); // External Function present in Application

#include <WiFi.h> // Contains Declarations of Function Calls related
const char* ssid = "TFLINK2020"; // Access Point User Name
const char* password = "2020TFLINK"; // Password of the AccessPoint

void setup() { // First Function That would Run when any Arduino Based
  Serial.begin(115200); // Set Baud Rate For Serial Communication
  Serial.setDebugOutput(true); // Let Debug Messages Appear
  Serial.println(); // Just Prints a NewLine

  static camera_config_t config;
  // Config Object is created to fill settings and Pin numbers. These
  // In NutShell its the Object Created to Configure the Sensor
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
}

void startServer();

int motor1Pina = 12;
int motor1Pinb = 13;

int motor2Pina = 14;
int motor2Pinb = 15;

void setup() {
  // sets the pins as outputs:
  pinMode(motor1Pina, OUTPUT);
  pinMode(motor1Pinb, OUTPUT);
  pinMode(motor2Pina, OUTPUT);
  pinMode(motor2Pinb, OUTPUT);

  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();
}
```

So now let's first start with building the oenophile. So I have opened two projects. As I said, the first one is the You camcorder oenophile and the next one is the car controller, IP Ivanovo, which is all previous. So let us see what are the differences between both if you immediately see the differences. The only thing what comes extra is a of more pins and the configuration in the second. And they literally where he was in the previous lap, we were using a soft baby. Now we don't require to use it, so we have to just add the motor pins to the main camera application or the camera.

```
D:\Mikubi\ESP32_CAM\Code\6Project_PC Based IP Cam Recorder\Tweak_Camera - Copy\Tweak_Camera.ino - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Tweak_Camera.ino
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's
if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Attempt for
Serial.println("Configuration with Static IP Failed");
}
#endif
WiFi.begin(ssid, password); // Attempt for Wifi Connection Establish
while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Connection
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
startCameraServer(); // Start the Camera Server / Streamer Explained
Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP()); // This will give you the IP Assigned
Serial.println("' to connect");
void loop() {
// put your main code here, to run repeatedly:
delay(1000);
}
ccoop_ap.ino
if(!WiFi.softAPConfig(STATIC_IP, gway, sNet)) {
Serial.println("Configuration with Static IP Failed");
}
#endif
/* Connecting to WiFi */
WiFi.softAP(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
startServer(); /* Starting camera server */
Serial.print(WiFi.localIP());
Serial.println("' to connect");
void loop() {
// put your main code here, to run repeatedly:
}
C++ source file length: 5,384 lines: 125 Ln: 1 Col: 1 Sel: 5,384 | 124 Windows (CR LF) UTF-8
```

So we copy the camera completely front from TV newscast. And, you know, this is your project. So you have the camera hydrofoils, everything in this we have to at least our server will be replaced with that camera. So I will not change it.

```
VS_CAR | Arduino 1.8.15
File Edit Sketch Tools Help
VS_CAR $ app_httpd.cpp camera_pins.h
void setup() { //First Function That would Run when any Arduino Based Board Starts
Serial.begin(115200); //Set Baud Rate For Serial Communication
Serial.setDebugOutput(true); // Let Debug Messages Appear
Serial.println(); //Just Prints a NewLine
// sets the pins as outputs:
pinMode(motor1PinA, OUTPUT);
pinMode(motor1PinB, OUTPUT);
pinMode(motor2PinA, OUTPUT);
pinMode(motor2PinB, OUTPUT);
static camera_config_t config;
// Config Object is created to fill settings and Pin numbers. These Macros are part of camera_pins.h
// In NutShell its the Object Created to Configure the Sensor
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
Done Saving
29 AI Thinker ESP32-CAM, 240MHz (WiFi@1), GPIO, 80MHz (WiFi), 80MHz
```

But instead we can declare the motor pins in the space here. Now, one more thing is set up in this setup. We are setting the pins as well. But anyway, this really work is only enabled. Just go to the setup and do that. Ten also controllers and then. What the changes only that nothing is exactly the same. So now let's go to opening the an application. This is the. Gargantua, right, Beattyville application, you can see the board here, UAP, let me open the.

```

D:\MIUbi\ESP32_CAM\Code\6.Project2_PC Based IP Cam Recorder\Tweak_Camera - Copy\app_httpd.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
app_httpd.cpp
#include "esp32-hal-ledc.h"

#include "esp_http_server.h"
#include "esp_timer.h"
#include "Arduino.h"

httpd_handle_t server_httpd = NULL;

extern int motor1Pina ;
extern int motor1Pinb ;

extern int motor2Pina ;
extern int motor2Pinb ;

static esp_err_t cmd_handler(httpd_req_t *req)
{
    char* buf;
    size_t buf_len;
    char variable[10] = {0};

#include "esp_http_server.h" // Contains Declarations of Function Calls
#include "esp_camera.h" // Contains Declarations of Function Calls related to camera
#include "Arduino.h" // Contains Declarations of Function Calls related to Arduino

#define PART_BOUNDARY "12345678900000000000000987654321" // A Unique Header
// String which would be passed to HTML Browser for indicating the stream content type
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace";
// Unique Header discussed will be replaced in place of macro
static const char* _STREAM_BOUNDARY = "--" PART_BOUNDARY "--\r\n";
// String which would be passed to HTML Browser for indicating the stream content type
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\n\r\n";
httpd_handle_t stream_httpd = NULL;

// Stream Handler Function mainly, takes images from camera and stream
static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL; // Initialize the Frame Buffer
    esp_err_t res = ESP_OK; // Initialize the result variable
    size_t _jpg_buf_len = 0; // variable which shall hold the buffer length
    uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
    char * part_buf[1024]; // Temporary buffer for string creation

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE); // When the type is set
    if(res != ESP_OK){
        return res;
    }

    while(true){
        fb = esp_camera_fb_get(); // Get the JPEG Frame from Image Sensor
        if (!fb) {
            return ESP_FAIL;
        }
    }
}

```

Project to Gambacorta application and let us compare both to see what is the big difference? The same thing, coffee here, the same application for this. Here, here. OK, let us come back now. What is the difference we are seeing now in order to access the model? Because we need the extent of the militants which is required. So let us do that.


```
D:\MIUBiESP32_CAM\Code\14Lab10_CarControl_over_ip_touch_API\ccoop_ap - Copy\app_httpd.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
app_httpd.cpp [3]
extern int motor2Pina ;
extern int motor2Pinb ;

static esp_err_t cmd_handler(httpd_req_t *req)
{
    char* buf;
    size_t buf_len;
    char variable[30] = {.,.};
    char value[30] = {0.};
    int res = 0;

    buf_len = httpd_req_get_url_query_len(req) + 1;
    Serial.println();
    Serial.print(buf_len);
    Serial.println(" is Buffer Length");

    if (buf_len > 0) {
        buf = (char*)malloc(buf_len);
        if (!buf) {
            httpd_resp_send_500(req);
            return ESP_FAIL;
        }
        if (httpd_req_get_url_query_str(req, buf, buf_len) == ESP_OK) {
            if (httpd_query_key_value(buf, "var", variable, sizeof(variable)) == ESP_OK) {
                httpd_query_key_value(buf, "val", value, sizeof(value)) == ESP_OK;
            } else {
                Serial.println("Camera capture failed");
                res = ESP_FAIL;
            }
        } else {
            Serial.println("Camera capture failed");
            res = ESP_FAIL;
        }
    }
}

esp_err_t res = ESP_OK; // Initialize the result variable
size_t _jpg_buf_len = 0; // variable which shall hold the buffer length
uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
char * part_buf[64]; // Temporary buffer for string creation

res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE); // When the type is set
if (res != ESP_OK) {
    return res;
}

while(true){
    fb = esp_camera_fb_get(); // Get the JPEG Frame from Image Sensor
    if (!fb) {
        Serial.println("Camera capture failed");
        res = ESP_FAIL;
    }
} else {
}
}

source file length: 5,683 lines: 208 Ln: 51 Col: 15 Sel: 685|23 Windows (CR LF) UTF-8
```

In the previous application, what we were doing is streaming and little, so here you have a common handler, so copy the complete command handler of your previous lab.

```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
VS_CAR app_httpd.cpp $ camera_pins.h
}
else if (val ==5) //backward
    Serial.println("Moving Backward");
    digitalWrite(motor1Pina, HIGH);
    digitalWrite(motor1Pinb, LOW);
    digitalWrite(motor2Pina, LOW);
    digitalWrite(motor2Pinb, HIGH);
}
else {
    Serial.println("variable");
    res = -1;
}
}

if (res) {
    return httpd_resp_send_500(req);
}

httpd_resp_set_hdr(req, "Access-Control-Allow-Origin", "*");
return httpd_resp_send(req, NULL, 0);
}

Done Saving
116 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz(BT), QIO, 80M
```

Copy the complete command and look and put it before our next to the screen and let you know coming to the IT in the previous project. We don't

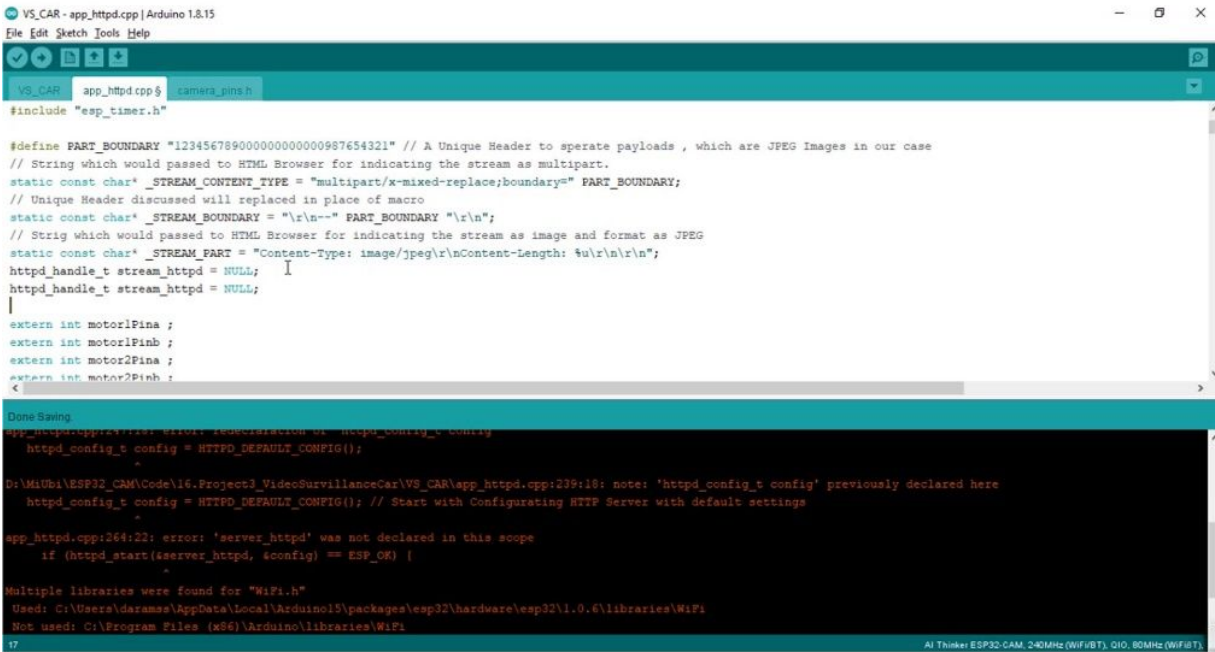
have any testimony, which was the screaming.

```
app_httpd.cpp
<button class="button" id="backward" onclick="fetch(document.
</div>
<br/>
</body>
</html>
rawliteral";
static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}
void startServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    httpd_uri_t index_uri = {
        .uri = "/",
        .method = HTTP_GET,
        .handler = index_handler,
        .user_ctx = NULL
    };
    httpd_uri_t cmd_uri = {
        .uri = "/control",
        .method = HTTP_GET,
        .handler = cmd_handler,
        .user_ctx = NULL
    };
    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&server_httpd, &config) == ESP_OK) {
        httpd_config_t config = HTTPD_DEFAULT_CONFIG(); // Start with Config
        config.server_port = 80; // What is Special about Port 80, its the
        httpd_uri_t stream_uri = {
            .uri = "/", // Just Root, no sub folder or nodes
            .method = HTTP_GET, // Request which come to server from cli
            .handler = stream_handler, // Function that shall be executed
            .user_ctx = NULL
        };
        if (httpd_start(&stream_httpd, &config) == ESP_OK) {
```

So let me put his name on it. No index. And there was also one of the portion which is present. Nokia is. So here in distractible. So in the camera's over. So we have an additional stream handler. So we forgot to register the handler and excel.

```
app_httpd.cpp
static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}
void startServer()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    httpd_uri_t index_uri = {
        .uri = "/",
        .method = HTTP_GET,
        .handler = index_handler,
        .user_ctx = NULL
    };
    httpd_uri_t cmd_uri = {
        .uri = "/control",
        .method = HTTP_GET,
        .handler = cmd_handler,
        .user_ctx = NULL
    };
    Serial.printf("Starting web server on port: '%d'\n", config.server_port);
    if (httpd_start(&server_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(server_httpd, &index_uri);
        httpd_register_uri_handler(server_httpd, &cmd_uri);
        httpd_config_t config = HTTPD_DEFAULT_CONFIG(); // Start with Config
        config.server_port = 80; // What is Special about Port 80, its the
        httpd_uri_t stream_uri = {
            .uri = "/", // Just Root, no sub folder or nodes
            .method = HTTP_GET, // Request which come to server from cli
            .handler = stream_handler, // Function that shall be executed
            .user_ctx = NULL
        };
        if (httpd_start(&stream_httpd, &config) == ESP_OK) {
            httpd_register_uri_handler(stream_httpd, &stream_uri); // Register
        }
    }
}
```

This copy the structures of command handler Nixonland Ystem here. Right, and then you have to register both of these government and commanding inmates, including the if statement that they said before that so we were doing the registrations tremendous good. It's really pretty. But certainly as good as GDP is not a. So let me say this tremendous stupidity which is declared null.



```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp $ camera_pins.h
#include "esp_timer.h"

#define PART_BOUNDARY "1234567890000000000000987654321" // A Unique Header to sperate payloads , which are JPEG Images in our case
// String which would passed to HTML Browser for indicating the stream as multipart.
static const char* _STREAM_CONTENT_TYPE = "multipart/x-mixed-replace;boundary=" PART_BOUNDARY;
// Unique Header discussed will replaced in place of macro
static const char* _STREAM_BOUNDARY = "\r\n--" PART_BOUNDARY "\r\n";
// Strig which would passed to HTML Browser for indicating the stream as image and format as JPEG
static const char* _STREAM_PART = "Content-Type: image/jpeg\r\nContent-Length: %u\r\n\r\n";
httpd_handle_t stream_httpd = NULL;
httpd_handle_t stream_httpd = NULL;

extern int motor1Pina ;
extern int motor1Pinb ;
extern int motor2Pina ;
extern int motor2Pinb ;

Done Saving
app_httpd.cpp:239:128: error: redeclaration of 'httpd_config_t config'
httpd_config_t config = HTTPD_DEFAULT_CONFIG();
^
D:\MIUBi\ESP32_CAMERA\Code\16.Project3_VideoSurveillanceCar\VS_CAR\app_httpd.cpp:239:18: note: 'httpd_config_t config' previously declared here
httpd_config_t config = HTTPD_DEFAULT_CONFIG(); // Start with Configuring HTTP Server with default settings
^
app_httpd.cpp:244:22: error: 'server_httpd' was not declared in this scope
if (httpd_start(&server_httpd, &config) == ESP_OK) {
^
Multiple libraries were found for "WiFi.h"
Used: C:\Users\darama\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.6\libraries\WiFi
Not used: C:\Program Files (x86)\Arduino\libraries\WiFi

17 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi/BT)
```

So let this clear one more object, pieces of paper handle. So now we have the handles in and streamline everything, all the buttons. So forward, backward, everything. That is great. Sounds great. Now, what is missing somehow on the will be you have to display the project, which is coming right there. The only difference as far as I see.

```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
VS_CAR app_httpd.cpp camera_pins.h
<div align=center>
<button class="button" id="forward" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=1');"ontouchend="fetch(document.location.origin+'/control?
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=2');"ontouchend="fetch(document.location.origin+'/control;
<button class="button" id="stop" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=3');">Stop</button>
<button class="button" id="turnright" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=4');"ontouchend="fetch(document.location.origin+'/control
</div>
<br/>
<div align=center>
<button class="button" id="backward" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=5');"ontouchend="fetch(document.location.origin+'/control;
</div>
<br/>
</body>
</html>
)rawliteral";
static esp_err_t index_handler(httpd_req_t *req){
    httpd_resp_set_type(req, "text/html");
    return httpd_resp_send(req, (const char *)INDEX_HTML, strlen(INDEX_HTML));
}
void startCameraServer(){
Done Saving
217 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi/QIO, 80M
```

Right. This is please note that this is the touch based we are doing on that site in the so I want the project feed to be present.

```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help
VS_CAR app_httpd.cpp camera_pins.h
    color: #FF0101;
    font-size: 18px;
}
</style>
</head>
<body>
<br/>
<div align=center>
<label class="label">Video Surveillance Car</label>
<br/>
<br/>
<br/>
<div align=center>
<button class="button" id="forward" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=1');"ontouchend="fetch(document.location.origin+'/control?
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=2');"ontouchend="fetch(document.location.origin+'/control;
<button class="button" id="stop" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=3');">Stop</button>
<button class="button" id="turnright" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=4');"ontouchend="fetch(document.location.origin+'/control
</div>
<br/>
Done Saving
210 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi/QIO, 80M
```

Before this morning's meeting, there will also be Óscar. Something like video surveillance caught right before the forward button I used to create the illusion of extra division, of displaying the beauty of it. So let us do that.

```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp camera_pins.h

    .label {
      color: #FF0101;
      font-size: 18px;
    }
  }
</style>
</head>
<body>
<br/>
<div align=center>
<label class="label">Surveillance Car</label>
<br/>
<div align=center> </div>
<br/>
<div align=center>
<button class="button" id="forward" ontouchstart="fetch(document.location.origin+'/control?var=ccolp&val=1');"ontouchend="fetch(document.location.origin+'/control?";
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" ontouchstart="fetch(document.location.origin+'/control?var=ccolp&val=2');"ontouchend="fetch(document.location.origin+'/control";
<button class="button" id="stop" ontouchstart="fetch(document.location.origin+'/control?var=ccolp&val=3');">Stop</button>
<button class="button" id="turnright" ontouchstart="fetch(document.location.origin+'/control?var=ccolp&val=4');"ontouchend="fetch(document.location.origin+'/control";
</div>
<br/>
<div align=center>

```

Done Saving

Leaving...
Hard resetting via RTS pin...

150 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi), O

Yeah, one important thing is in the image. So the way we feel, the image few of them depict the our project is we use a parameter called AMG and the source being your IP address and reportedly being 81 string.

```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp camera_pins.h

    .method = HTTP_GET,
    .handler = index_handler,
    .user_ctx = NULL
  };
};

httpd_uri_t cmd_uri = {
  .uri = "/control",
  .method = HTTP_GET,
  .handler = cmd_handler,
  .user_ctx = NULL
};

httpd_uri_t stream_uri = {
  .uri = "/stream", //Just Root, no sub folder or nodes
  .method = HTTP_GET, // Request which come to server from client have the method as HTTP_GET, GET is used by clients to request data.
  .handler = stream_handler, // Function that shall be executed for GET request from clients
  .user_ctx = NULL
};

Serial.printf("Starting web server on port: '%d'\n", config.server_port);
if (httpd_start(&server_httpd, &config) == ESP_OK) {
  httpd_register_uri_handler(server_httpd, &index_uri);
  httpd_register_uri_handler(server_httpd, &cmd_uri);
}

config.server_port += 1;
config.ctrl_port += 1;
if (httpd_start(&stream_httpd, &config) == ESP_OK) {

```

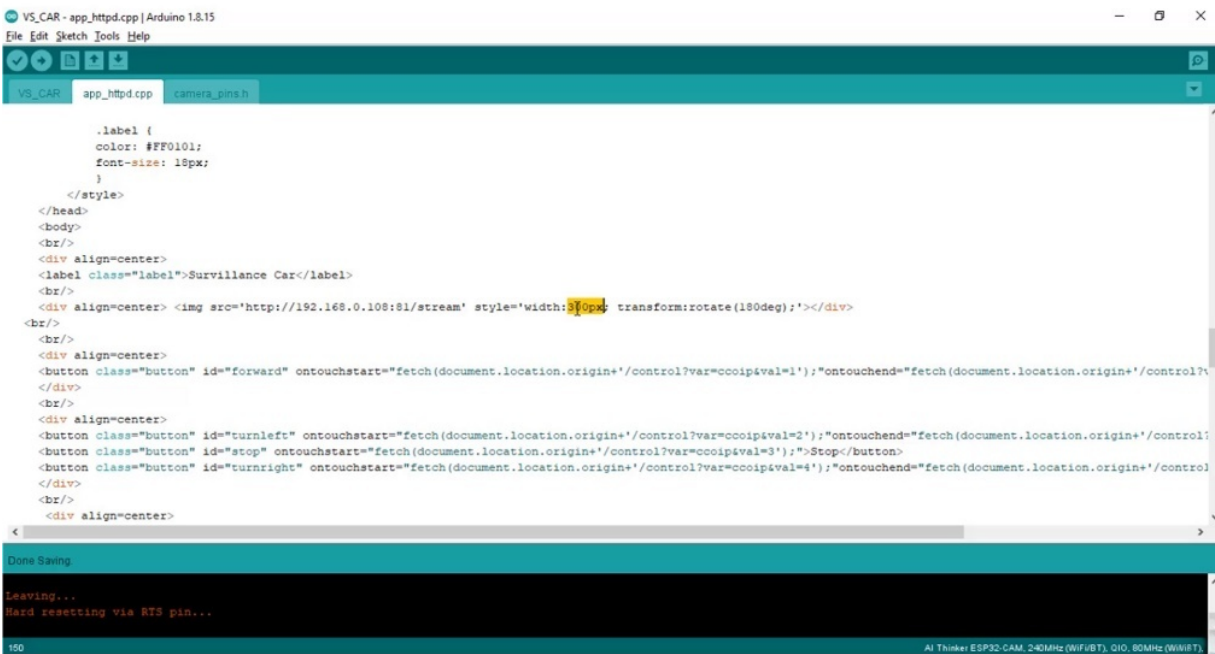
Done Saving

Leaving...
Hard resetting via RTS pin...

241 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi), O

No one important change that we should definitely do without miss is. Changing our previous cold stream, you are right, which was earlier

backlash, just the backlash. Now we have to include celestially. So this comment no longer applies right then after before considering before actually trying to restrain the handler for signore. Or starting his credibility for this stream, his credibility configuration, we have to incrementally put because we cannot do all the things over a single poll. So both 80, which has been initialized, that will be used for both control and index handling, whereas plus one, which is a silver bullet, the one in control, but one those will handle the workers streaming.



```
VS_CAR - app_httpd.cpp | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp camera_pins.h

    .label {
      color: #FF0101;
      font-size: 18px;
    }
  }
</style>
</head>
<body>
<br/>
<div align=center>
<label class="label">Surveillance Car</label>
<br/>
<div align=center> </div>
<br/>
<br/>
<div align=center>
<button class="button" id="forward" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=1');"ontouchend="fetch(document.location.origin+'/control?
</div>
<br/>
<div align=center>
<button class="button" id="turnleft" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=2');"ontouchend="fetch(document.location.origin+'/control:
<button class="button" id="stop" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=3');">Stop</button>
<button class="button" id="turnright" ontouchstart="fetch(document.location.origin+'/control?var=ccoop&val=4');"ontouchend="fetch(document.location.origin+'/control
</div>
<br/>
<div align=center>
```

Done Saving
Leaving...
Hard resetting via RTS pin...

150 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), QIO, 80MHz (WiFi/BT)

So that is the reason why I have put a division here. Right. So this will destroy Netflix pixel currently because it is the best of this to see on a mobile phone. You can increase it to some 450 or 500 based on your mobile phone device. Whether if you are using a tablet, you can increase it much more. So this rotation by her index of 180 degrees because of my camera has it.

```
VS_CAR | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp camera_pins.h
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000; //SPI Clock Frequency set to 20MHz
config.pixel_format = PIXFORMAT_JPEG; //Compression format from Sensor is selected as JPEG
config.frame_size = FRAMESIZE_VGA; // Resolution VGA
config.jpeg_quality = 12; //lowest is better, lowest possible is 10
config.fb_count = 1; //Number of Frame Buffers at a time

// camera init with required settings and pin numbers
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) { //This Flag is Set Internally by Init Function in case of failure
  Serial.printf("Camera init failed with error 0x%x", err); //error number would tell us type of error
  return;
}
#if 1
//Acquire Access to the sensor
sensor_t * s = esp_camera_sensor_get(); // Get Access to tweak camera settings
s->set_brightness(s, 0); // -2 to 2
s->set_contrast(s, 0); // -2 to 2
#endif

Done Saving
Leaving...
Hard resetting via RTS pin...

65 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), Q10, 80MHz, 80IO, 8.
```

So here you can play with the resolution. So if you want to get this stream very far better, we can play with video resolution or less because lower resolutions have less data and. You will have much compression.

```
VS_CAR | Arduino 1.8.15
File Edit Sketch Tools Help

VS_CAR app_httpd.cpp camera_pins.h
s->set_aec2(s, 1); // 0 = disable , 1 = enable
s->set_ae_level(s, 0); // -2 to 2
s->set_aec_value(s, 1200); // 0 to 1200
s->set_gain_ctrl(s, 1); // 0 = disable , 1 = enable
s->set_agc_gain(s, 30); // 0 to 30
s->set_gainceiling(s, (gainceiling_t)6); // 0 to 6
s->set_bpc(s, 1); // 0 = disable , 1 = enable
s->set_wpc(s, 1); // 0 = disable , 1 = enable
s->set_raw_gma(s, 1); // 0 = disable , 1 = enable
s->set_lenc(s, 1); // 0 = disable , 1 = enable
s->set_hmirror(s, 0); // 0 = disable , 1 = enable
s->set_vflip(s, 0); // 0 = disable , 1 = enable
s->set_dcw(s, 1); // 0 = disable , 1 = enable
s->set_colorbar(s, 0); // 0 = disable , 1 = enable
#endif

#if 1
// Set your Static IP address of ESP32-CAM Node / Station.
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway Address, in our case the Gate way is router, this IP is common for any routers
IPAddress sNet(255, 255, 0, 0); //Sub Net Mask, Default as we have no network divisions
IPAddress pDNS(8, 8, 8, 8); //Primary DNS - This is Google's
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS - This is Google's

if (!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) { // Attempt for Configuration of Wifi Hardware with Settings
  Serial.println("Configuration with Static IP Failed");
}

Done Saving
Leaving...
Hard resetting via RTS pin...

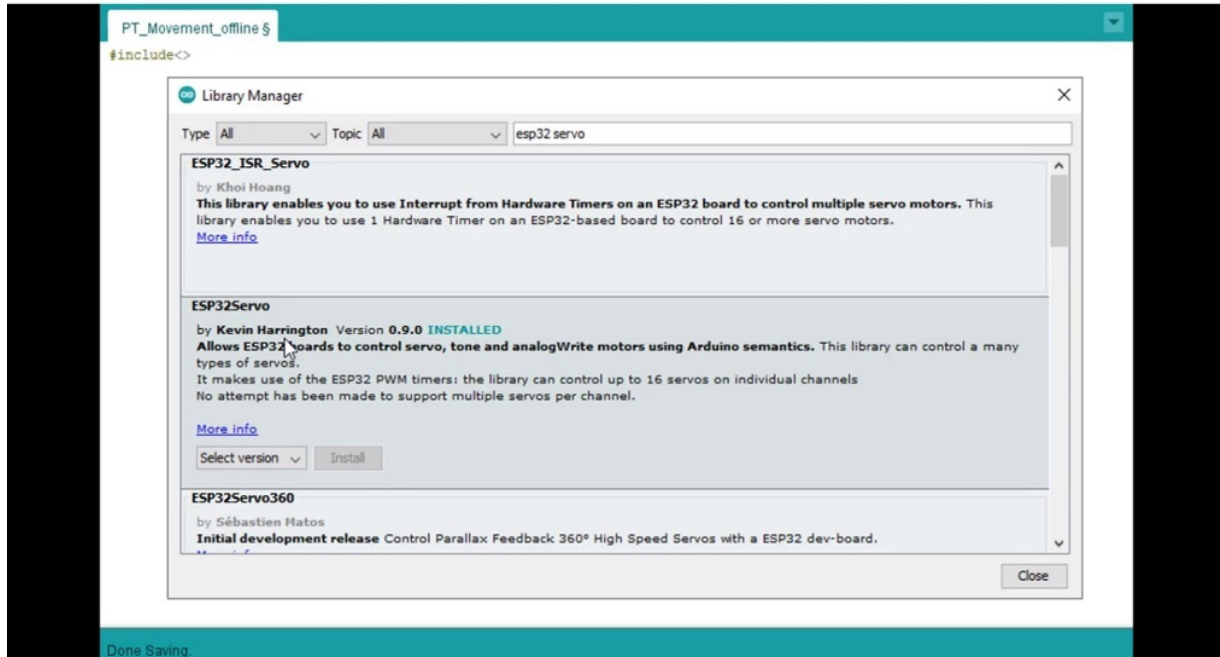
80 AI Thinker ESP32-CAM, 240MHz (WiFi/BT), Q10, 80MHz (WiFi/BT).
```

And you can also play with the sleeping. The vertical and horizontal flip you can play in that also, or you can play with the stimulus a little because

the browser itself will flip it. So that the major formation will be faster and imagining to be faster. So I have compiled this code and this is pretty much fine.

CODING

start calling it a moment of severe line of sight and include disorder.



Yeah, before that, one important thing to do is go to manage laboratories and years to better data space. So. So there is one library from Kevin Haddington. This one is better. So it makes you Sofia's better to be. The library can control of the 16 rules on individual terms. No attempt has been made to support multiple sources. But the reason why I have chosen this particular library is it has a control over particular. If you use a normal service library, it will not have the channel control, which means you will be destroying the only existing use to be of use, the little with which it misbehaves. So in order to avoid that misbehavior, you have to love the tiger, which means that in fact, OK, so you have to install this version. I have already installed zero point nineteen point zero whatever, as late as we can install it. So I'm telling you, it is better to do so. You can just use better toothpaste, so. By Kevin Harinder, OK, once that is done,


```
PT_Movement_offline $
#include <Servo.h>

Servo myservo_pan;
Servo myservo_tilt;

int pos = 90;

int servoPin_pan = 16;
int servoPin_tilt = 2;

void setup()
{
  myservo_pan.attach(servoPin_pan,2);//Timer 0,1,2,3
  myservo_tilt.attach(servoPin_tilt,3);//Timer 0,1,2,3

  myservo_pan.write(pos);
  myservo_tilt.write(pos);

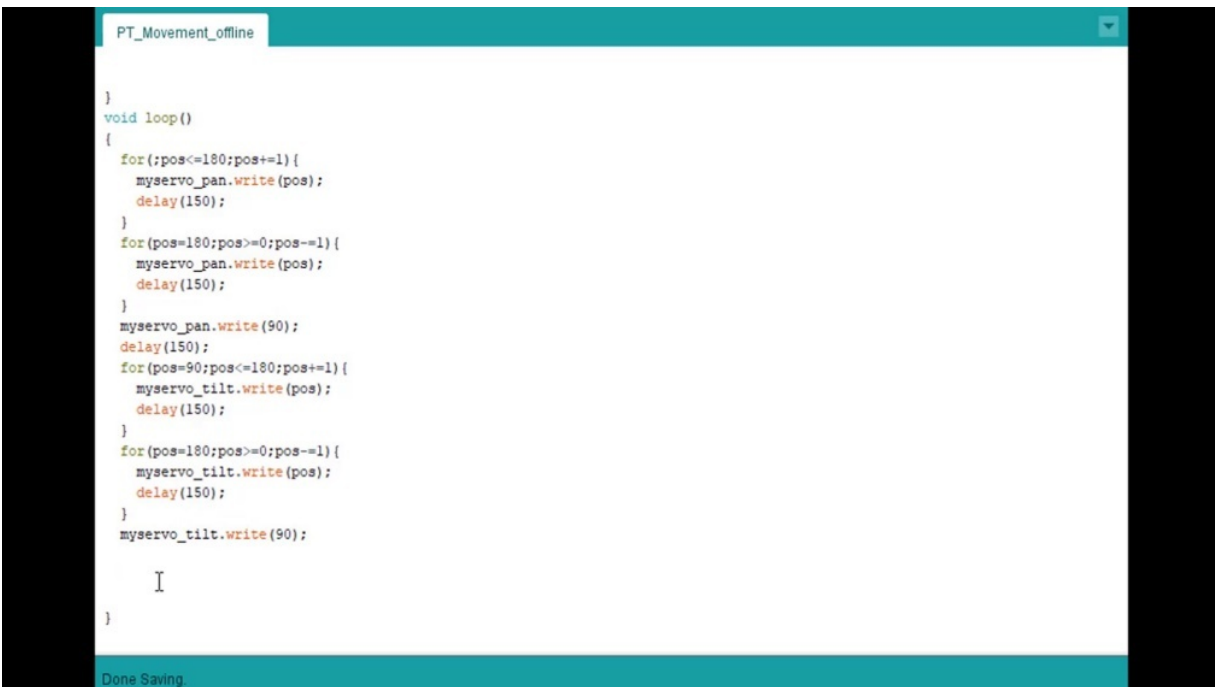
  pinMode(servoPin_pan, OUTPUT);
  pinMode(servoPin_tilt, OUTPUT);
  digitalWrite(servoPin_pan, LOW);
  digitalWrite(servoPin_tilt, LOW);
}

void loop()
{
}

Done Saving
```

you can just include the Salvador file and then we have to create a third. I can see myself and let's go. MAN one for the band. Model one for the Mizer one this quarter. OK, let me initialize the position with the variable called initialization of the position variable with ninety ninety degrees. And then let us also declare these are open for band position. For controlling the band, I meant 15 or 16 as as. One more thing for debate over the second debate would be that once the system does go to the second. War in Syria, which is a regular function call, is done in our and I've been worried look. And the important portion here of this court is you take this object. And in that. Access, the function, edge and use are open. Which is what you want to control and as I'm going to do it. See GPO, sorry, Taimyr zero one, two, three are available only as digital, along with three, just four is better. The camera, while Zeland one are internally used for WiFi connections, are also the leftover ones too. So as because we are not currently using the Channel three all the time. Three, you can attach your model. So for which you have access the object and then access the attached column and assign the Philippine military. And then you can initialize your servo motors plants are also using mizer water from the school band of. So this will light the position. So it's a positive rate. Ninety, it will move your server to the position, if you like, 180. It will move this to. Position 180. So both actually initially initialism the position very well, Richard. Then I have to make this

all serpens as occupant's. And also make the output look just for the initialization cos nothing else.



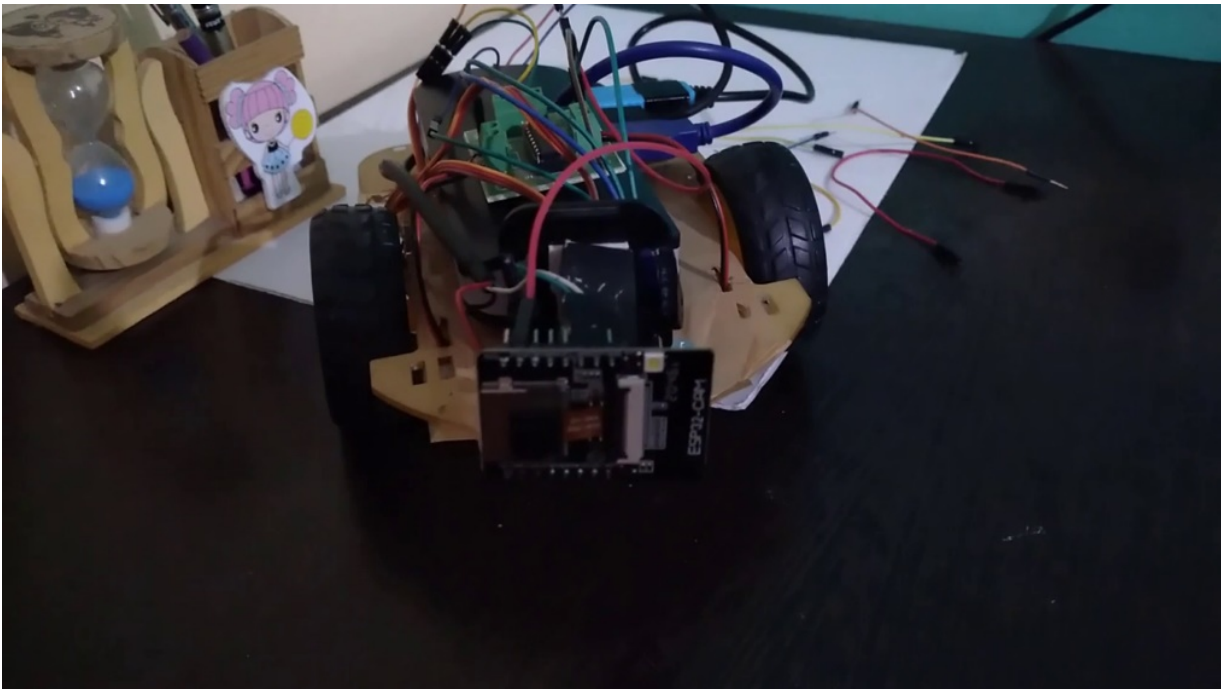
```
PT_Movement_offline
}
void loop()
{
  for(;pos<=180;pos+=1){
    myservo_pan.write(pos);
    delay(150);
  }
  for(pos=180;pos>=0;pos-=1){
    myservo_pan.write(pos);
    delay(150);
  }
  myservo_pan.write(90);
  delay(150);
  for(pos=90;pos<=180;pos+=1){
    myservo_tilt.write(pos);
    delay(150);
  }
  for(pos=180;pos>=0;pos-=1){
    myservo_tilt.write(pos);
    delay(150);
  }
  myservo_tilt.write(90);
  I
}
Done Saving
```

So let us come into the loop. So what to do in the loop? So I want to just do a loop where the position doesn't exceed 180. You know, this kind of a statement, this position plus the one innocense position, the position plus one. So in this, I want to write. So here are my circle on this Turpan Dautry. The position, so it starts from the main debate and goes to 180. And while the positional movement is happening, I would suggest you do give a delay of at least 150 milliseconds so that when you from the current position, the motor has to move to a certain position. So it reflects on certain time eligible. So keep the delay of 150 milliseconds so that we will give time for the fine motor Pancevo motor to reach that position thereafter. Again, that's a 180. And then. Making them the position that it goes to 180 comes back to zero again. So the position should be the false statement, that is the it should come out of the loop when it is great negative. It means the position is going less than zero. Then I should come out of the loop. Then I will complete the by movement from zero to one later to zero. The competition is taking off for know what I'm doing is so I'll make the position again to 90 and I'll bring the. Or let me make the position come to Ninety-One once again, because myself and the Sculpin don't, right. So that at this minute.

OK, give it a little one to speak so that you get something. So no starting position, maybe. The more details, more than. But then after once it reaches 180 again, you start from the bottom. And go up. Like, once again, when the loop starts, you have to repeat the same as like you have done with the. You have to do it with the kid. That means then find the one at the moment is happening. There should be 20, 30, 90. Is just for observation purposes or else the final will be in the previous position itself. And the other more keeps on moving, which doesn't you not be able to make out properly because it is individuals like I like the call and let us have a look at the demo.

DEMONSTRATION

Asplen DeSario first initialised as interlaces to go to the main due to the position, and then it'll move from Bangladesh and zero to 180. Like. Then after when it reaches 180, it comes back from 180. To zero again.

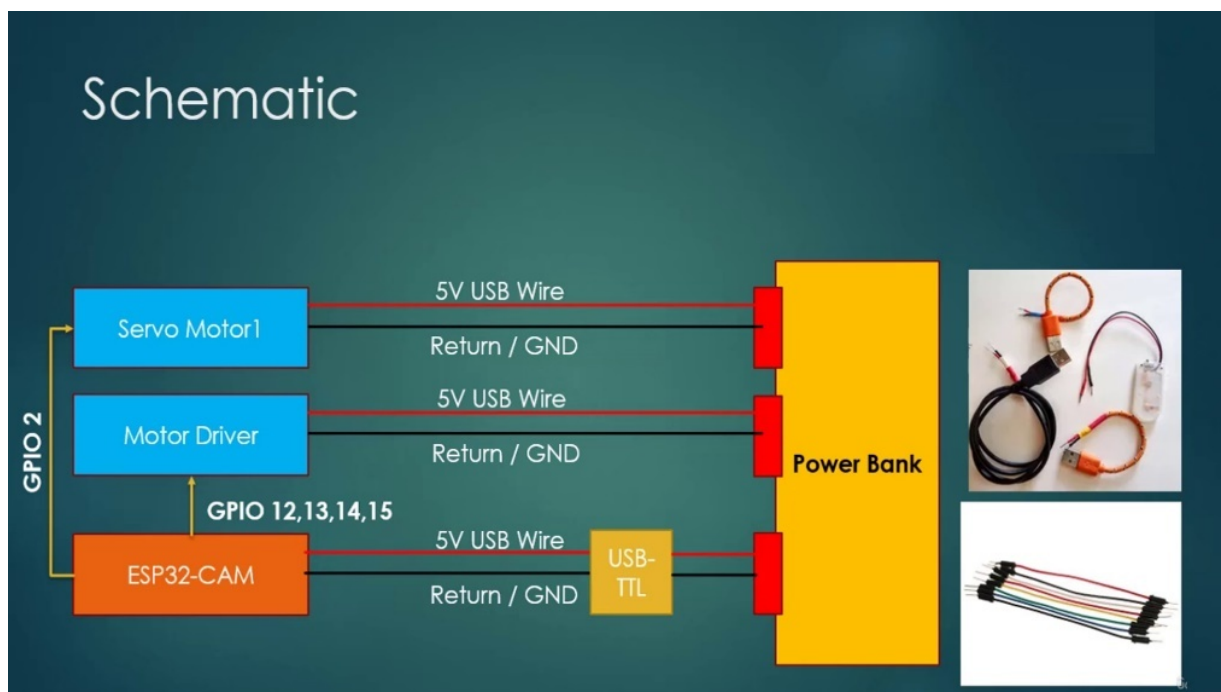


The horizontal position is always called Lisburn, whereas the vertical position and movement is called. So once the movement is completed, they will move to zero in 90. And from there we go to the minus positions, which is the down side, and then slowly will come up. So you can also see here, however, the schematic of the British system where a separate power

supply is taken for the old model and separate parts of plastic and supermodel and separate supply for the inspector to camera.

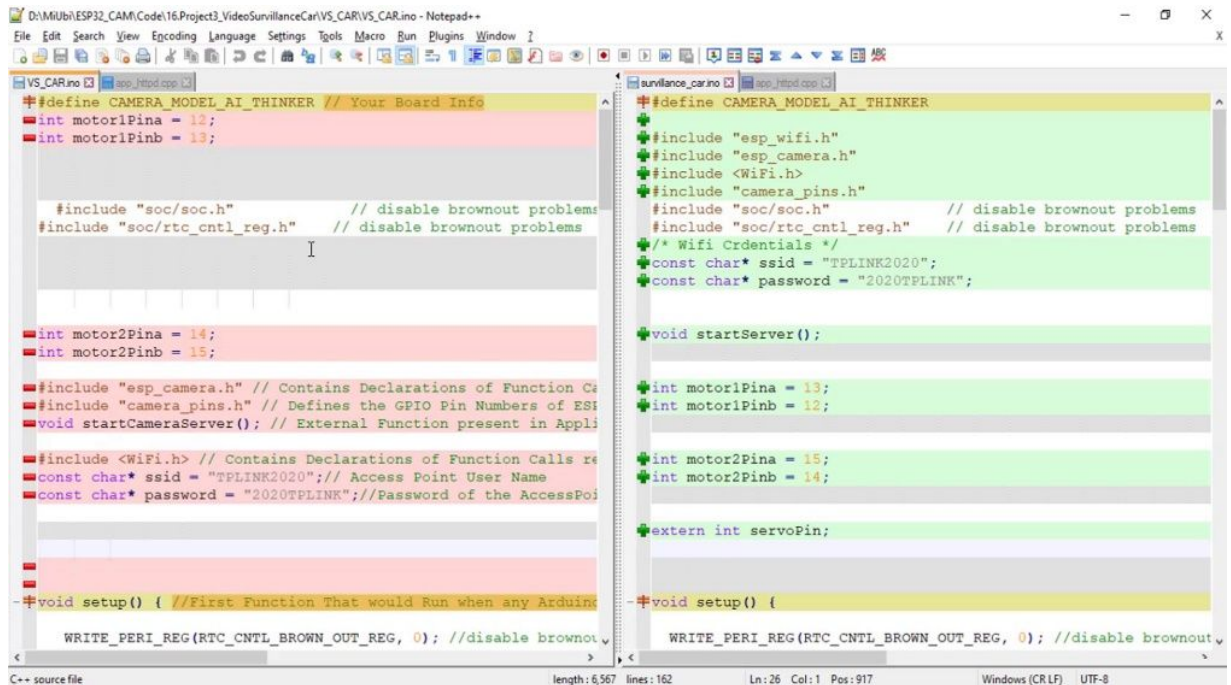
INTRODUCTION AND CODING

we'll try to achieve the objective of releasing a 360 degree video surveillance Karolinska. So the notable important things, which I want to mention as a disclaimer is the camera module, obviously digital camera occupies the time entry, which you want to use for the old model department of the bathroom. Jews already occupied, whereas the whitefly modules already occupied zero in one time. That's. And the only leftover Paymer with. Oh, yes, Peter, the camel is the channel to all the time. With which we are trying to play with the element so the extra water cannot be accommodated, so we use the robot itself for left and right movement so the robot can rotate completely up and down motions. You can use the toilet so that you can achieve the 360 degree. Roland.



Coming to the schematic. So the motor driver board is connected with the five children grown from the power bank and several motors apparently connected to what motives do you wish to separate the motor water and motor driver separately? Because there will be river systems which are you don't want to be entering do VOICE-OVER So drivers and the hospital can

pay was 12, 13, 14, 15 and go escort the motor driver and the GP to go to this full tilt operation. This is a typical setup.



```
#define CAMERA_MODEL_AI_THINKER // Your Board Info
int motor1Pina = 12;
int motor1Pinb = 13;

#include "soc/soc.h" // disable brownout problems
#include "soc/rtc_cntl_reg.h" // disable brownout problems

int motor2Pina = 14;
int motor2Pinb = 15;

#include "esp_camera.h" // Contains Declarations of Function Calls
#include "camera_pins.h" // Defines the GPIO Pin Numbers of ESP8265
void startCameraServer(); // External Function present in Application

#include <WiFi.h> // Contains Declarations of Function Calls
const char* ssid = "TPLINK2020"; // Access Point User Name
const char* password = "2020TPLINK"; // Password of the Access Point

void setup() { // First Function That would Run when any Arduino Board starts
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detection
}
```

```
#define CAMERA_MODEL_AI_THINKER

#include "esp_wifi.h"
#include "esp_camera.h"
#include <WiFi.h>
#include "camera_pins.h"
#include "soc/soc.h" // disable brownout problems
#include "soc/rtc_cntl_reg.h" // disable brownout problems
/* Wifi Credentials */
const char* ssid = "TPLINK2020";
const char* password = "2020TPLINK";

void startServer();

int motor1Pina = 13;
int motor1Pinb = 12;

int motor2Pina = 15;
int motor2Pinb = 14;

extern int servoPin;

void setup() {
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); //disable brownout detection
}
```

Now let us move to the code. Here I have opened the files of our previous project, which is Project three, the producer LINSCOTT, and now currently the 360 degree surveillance. What is the core difference? I have done I have included extra sarcopenia. All things have seen, nothing has changed, the more turbines which have that, they're there as it is. The Ford Motor Pins makes Streamwood Motor businesses are open. I have made that as an output, but that is the only difference I have done in the Restall streaming the image settings.

```

VS_CARino | app_httpd.cpp |
s->set_aec2(s, 1); // 0 = disable, 1 = enable
s->set_ae_level(s, 0); // -2 to 2
s->set_aec_value(s, 1200); // 0 to 1200
s->set_gain_ctrl(s, 1); // 0 = disable, 1 = enable
s->set_agc_gain(s, 30); // 0 to 30
s->set_gainceiling(s, (gainceiling_t)6); // 0 to 6
s->set_bpc(s, 1); // 0 = disable, 1 = enable
s->set_wpc(s, 1); // 0 = disable, 1 = enable
s->set_raw_gma(s, 1); // 0 = disable, 1 = enable
s->set_lenc(s, 1); // 0 = disable, 1 = enable
s->set_hmirror(s, 0); // 0 = disable, 1 = enable
s->set_vflip(s, 0); // 0 = disable, 1 = enable
s->set_dcw(s, 1); // 0 = disable, 1 = enable
s->set_colorbar(s, 0); // 0 = disable, 1 = enable
#endif

#if 0
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway
IPAddress sNet(255, 255, 0, 0); //Mask

#endif

if(!WiFi.softAPConfig(STATIC_IP, gway, sNet)) {
Serial.println("Configuration with Static IP Failed");
}

WiFi.softAP(ssid, password);

surveillance_carino | app_httpd.cpp |
s->set_aec2(s, 1); // 0 = disable, 1 = enable
s->set_ae_level(s, 0); // -2 to 2
s->set_aec_value(s, 1200); // 0 to 1200
s->set_gain_ctrl(s, 1); // 0 = disable, 1 = enable
s->set_agc_gain(s, 30); // 0 to 30
s->set_gainceiling(s, (gainceiling_t)6); // 0 to 6
s->set_bpc(s, 1); // 0 = disable, 1 = enable
s->set_wpc(s, 1); // 0 = disable, 1 = enable
s->set_raw_gma(s, 1); // 0 = disable, 1 = enable
s->set_lenc(s, 1); // 0 = disable, 1 = enable
s->set_hmirror(s, 0); // 0 = disable, 1 = enable
s->set_vflip(s, 0); // 0 = disable, 1 = enable
s->set_dcw(s, 1); // 0 = disable, 1 = enable
s->set_colorbar(s, 0); // 0 = disable, 1 = enable

#if 1
// Set your Static IP address
IPAddress STATIC_IP(192, 168, 0, 108);
// Set your Gateway IP address
IPAddress gway(192, 168, 0, 1); //Gateway
IPAddress sNet(255, 255, 0, 0); //Mask
IPAddress pDNS(8, 8, 8, 8); //Primary DNS
IPAddress sDNS(8, 8, 4, 4); //Secondary DNS

#endif

if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) {
Serial.println("Configuration with Static IP Failed");
}

```

Everything remains the same if you want to tweak anything, you can tweak. Right. Nothing has changed. These are the changes of make who want to make as a static IP. I just have included it. That is not a difference.

```

VS_CARino | app_httpd.cpp |
if(!WiFi.config(STATIC_IP, gway, sNet, pDNS, sDNS)) // Attempt
Serial.println("Configuration with Static IP Failed");
}
WiFi.begin(ssid, password); // Attempt for Wifi Connection Est
while (WiFi.status() != WL_CONNECTED) { // Result of Wifi Conn
delay(500);
Serial.print(".");
}
#endif

Serial.println("");
Serial.println("WiFi connected");

startCameraServer(); // Start the Camera Server / Streamer Exp

Serial.print("Camera Ready! Use 'http://'");
Serial.print(WiFi.localIP()); // This will give you the IP Add
Serial.println(" to connect");

}

void loop() {
// put your main code here, to run repeatedly:
delay(10000);
}

surveillance_carino | app_httpd.cpp |
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");

startServer(); // Starting camera server */

Serial.print(WiFi.localIP());
Serial.println(" to connect");

}

void loop() {
// put your main code here, to run repeatedly:
}

```

Everything everything's in the structure with only the names that cameras that will stretch the. It. Let's talk things that seem inclusion of widespread

outage in society, camera phones use the camera, everything, everything reminds me of it. But when it comes to the application, we have to include an extra skin, I mean, extra pain control of the treatment. And also I use a button called Home for bringing the data model to the plus name to position.

```
httpd_handle_t stream_httpd = NULL;
httpd_handle_t server_httpd = NULL;

extern int motor1Pina ;
extern int motor1Pinb ;

extern int motor2Pina ;
extern int motor2Pinb ;

// Stream Handler Function mainly, takes images from camera and
static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL; // Initialize the Frame Buffer
    esp_err_t res = ESP_OK; // Initialize the result variable
    size_t _jpg_buf_len = 0; // variable which shall hold the
    uint8_t * _jpg_buf = NULL; // pointer to JPEG buffer
    char * part_buf[64]; // Temporary buffer for string creation

    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE); // W

httpd_handle_t stream_httpd = NULL;
httpd_handle_t camera_httpd = NULL;

int pos=90;

extern int motor1Pina ;
extern int motor1Pinb ;

extern int motor2Pina ;
extern int motor2Pinb ;

Servo myservo; // create servo object to control a servo
int servoPin = 2;

static esp_err_t stream_handler(httpd_req_t *req){
    camera_fb_t * fb = NULL;
    esp_err_t res = ESP_OK;
    size_t _jpg_buf_len = 0;
    uint8_t * _jpg_buf = NULL;
    char * part_buf[64];
    res = httpd_resp_set_type(req, _STREAM_CONTENT_TYPE);
```

So let us see the application code, so in the application code to start from the starting on inclusion, said that everything is the perfect. You have created a set object myself, and the solution is to make the stream is as it is. There is no change. And the campaign handlers, as it is, there is no change.

```
D:\MUBH\ESP32_CAM\Code\16.Project3_VideoSurveillanceCar\VS_CAR\app_httpd.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
VS_CARino [1] app_httpd.cpp [2]
}
else
{
Serial.println("variable");
res = -1;
}
}

surveillance_carino [1] app_httpd.cpp [2]
}
else
{
Serial.println("Result -1");
res = -1;
}
}
if (!strcmp(variable, "pt"))
{
if (val == 1) //forward
Serial.println("Moving Tilt - ");
pos = pos + STEP;
if (pos >= 180)
pos = 180;
myservo.write(pos);
delay(50);
}
else if (val == 2) //Tilt +
Serial.println("Moving Tilt + ");
pos = pos - STEP;
if (pos <= 0)
pos = 0;
myservo.write(pos);
delay(50);
}
else if (val == 3) //Home
Serial.println("Going Home");
pos = 90;
myservo.write(pos);
delay(250);
}
}
}

C++ source file length: 10,408 lines: 328 Ln: 93 Col: 2 Sel: | 1 | 1 Windows (CR LF) UTF-8
```

Here there is a small change of extra variable being monitored, which is beating so fast, we'll move to move to the webpage and then we'll come back here. How are we doing this?

```
D:\MUBH\ESP32_CAM\Code\18.Project4_360degreeIPSurveillanceCar\surveillance_car\app_httpd.cpp - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
VS_CARino [1] app_httpd.cpp [2]
<button class="button" id="turnleft" ontouchstart="fetch(docu
<button class="button" id="stop" ontouchstart="fetch(docume
<button class="button" id="turnright" ontouchstart="fetch(c
</div>
<br/>
<div align=center>
<button class="button" id="backward" ontouchstart="fetch(d
</div>
<br/>
</body>
</html>
)rawliteral";
static esp_err_t index_handler(httpd_req_t *req){
httpd_resp_set_type(req, "text/html");
return httpd_resp_send(req, (const char *)INDEX_HTML, STRLE
}
void startCameraServer() {

surveillance_carino [1] app_httpd.cpp [2]
<button class="button" id="turnleft" ontouchstart="fetch(docu
<button class="button" id="stop" ontouchstart="fetch(docume
<button class="button" id="turnright" ontouchstart="fetch(c
</div>
<div align=center>
<button class="button" id="servo_n" ontouchstart="fetch(doc
<button class="button" id="home" ontouchstart="fetch(docume
<button class="button" id="servo_p" ontouchstart="fetch(doc
</div>
<br/>
</body>
</html>
)rawliteral";
static esp_err_t index_handler(httpd_req_t *req){
httpd_resp_set_type(req, "text/html");
return httpd_resp_send(req, (const char *)INDEX_HTML, STRLE
}
void startServer()
{
myservo.attach(servoPin,2); // attaches the servo on pin to th
myservo.write(90);
delay(150); // waits 150ms for the servo to reach

C++ source file length: 9,613 lines: 328 Ln: 273 Col: 28 Pos: 7,982 Windows (CR LF) UTF-8
```

So here the streaming you are elevating is him, so I have added extra three buttons, so negative, so positive and all of this are going negative. Which

means moving the motor to the negative side, moving the motor to the positive.

```
VS_CARino [ app_httpd.cpp ]
/control?var=ccoip&val=2');"ontouchend="fetch(document.location)
trol?var=ccoip&val=3');">Stop</button>
/control?var=ccoip&val=4');"ontouchend="fetch(document.location)

/control?var=ccoip&val=5');"ontouchend="fetch(document.location)

surveillance_carino [ app_httpd.cpp ]
/control?var=ccoip&val=2');"ontouchend="fetch(document.location)
trol?var=ccoip&val=3');">Stop</button>
/control?var=ccoip&val=4');"ontouchend="fetch(document.location)

/control?var=ccoip&val=5');"ontouchend="fetch(document.location)

control?var=pt&val=1');">TILT - </button>
trol?var=pt&val=3');">HOME</button>
control?var=pt&val=2');">TILT + </button>

ected timer
```

So imagine 90 is your straight position if you want to move down. So you have to subtract when increment the position 90 plus one plus two plus three up to 180. If you want to move the motor upside, you have to decrease. So I have for this button separately have given values. One, two, three. So when you go to the control, the command handle here, the request of one comes, which means. We want to move the motor to the negative state. That means don't say so. The current position is mainly because, I mean, it's closer to the 90. So being reminded based here, what step is I do the exact value. I have put it 15 so that it is not one in order to create faster motion. I have been to 15 so that it can move vastly. Because the camera has to and we love you so. You can come back later. It's not a problem, so I move to the downside upon pressing the negative for positive. I move to the top side so that I have to resort to the zero to position like. And appropriately, delay after every positional request coming to home is nothing, but moving the position may be right. This is the only difference. What we have done for a security surveillance can now let us move to the demonstration of the same.

THE END