A PROJECT REPORT

on

“ **Application of ESP32 on Wireless Networks** ”

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN
INFORMATION TECHNOLOGY

BY

Nikhil Raj 1605577

Shivam Singh 1605151

UNDER THE GUIDANCE OF
Dr. MANOJ KUMAR MISHRA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

KIIT Deemed to be University
School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"Application of ESP32 on Wireless Networks"

submitted by

Nikhil Raj	1605577
Shivam Singh	1605151

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-enc & Engineering) at KIIT Deemed to be university,Bhubaneswar. This work is done during year 2018-2019, under our guidance.

Date: 03 /07 / 2019

Dr. MANOJ KUMAR MISHRA
Project Guide

Acknowledgement

I would like to express deep gratitude towards **Dr. Manoj Mishra** for giving me the opportunity to work on this project. Also I would like to thank him for his guidance, encouragement and gracious support throughout the internship. I would also like to thank KIIT UNIVERSITY for providing the facilities that helped in the completion of this project.

Nikhil Raj

Shivam Singh

Abstract

Arduino ESP WROOM 32 is developed by the espressif systems.This is one of the latest arduino boards in the market. The chip comes with features like Wi-Fi, bluetooth connectivity. The chip also contains built-in sensors like touch, temperature and hall effect sensor. It can be powered through an USB cable or battery.The chip is compatible with a variety of hardware such as sensors, motors, etc.In all the projects that follow, Arduino ESP WROOM 32 is used.The chip can be set in a SoftAP or station mode or a Combo mode and it can carry up to 250 byte payload.The chip is used to transfer small data to its peer.The data reaches the central server through its peer in a hop by hop manner.

There are 2 parts of the project .In the first the chip uses the external wifi and in the second it uses its built in internal wifi where it tries to connect with its peer through the physical address.

Contents

List of Figures

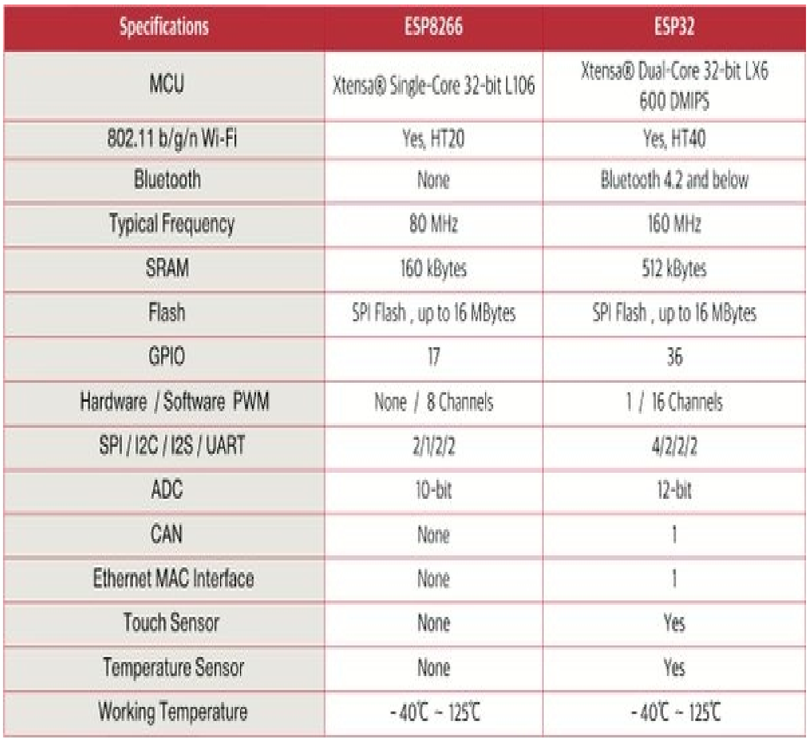
INTRODUCTION

Arduino is an open-source hardware. It is a single-board micro controller chip that is both cheap and easy to program. The chip can be programmed using the Arduino Software(IDE) that is a cross platform application. The Arduino programming language is used for programming the micro controller board. There are a wide variety of Arduino micro controllers available in the market. Each chip can be used to create objects that can sense and control objects in the physical and digital world. Due to its low cost and several features, the Arduino boards are used in a variety of innovative projects. The boards can be connected to different types of both hardwares and softwares. The micro controller board contains several digital and analog pins that are used for connection. The board also has serial port for serial communication with the computer. It can be programmed manually using this serial connection.

In all the projects that follow, Arduino ESP WROOM 32 is used. This is one of the latest arduino boards in the market. It is developed by the espressif systems. The chip comes with features like Wi-Fi, bluetooth connectivity. The chip also contains built-in sensors like touch, temperature and hall effect sensor. It can be powered through an USB cable or battery. The chip contains the latest bluetooth 4.0 and also the bluetooth low energy(BLE). It also contains a special low power Wi-Fi protocol called esp-now. Esp-now is developed by the espressif systems so that the chips can communicate with each other without depending on any external resources. This is the same technology that is used in wireless mice. To lower the power consumption, the chip has two sleep modes, the light sleep and deep sleep. The chip has a storage of 32 MB. Additional data can be stored by connecting a SD card to the chip. The chip is compatible with a variety of hardware such as sensors, motors, etc. The chip has more features than any of the previous arduino chips and thus can be used for a variety of applications.

Fig1: ESP32 chip

Fig2: Comparison between esp32 and earlier board (esp8266)



Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	- 40℃ ~ 125℃	- 40℃ ~ 125℃

ESP 32 Pin out diagram

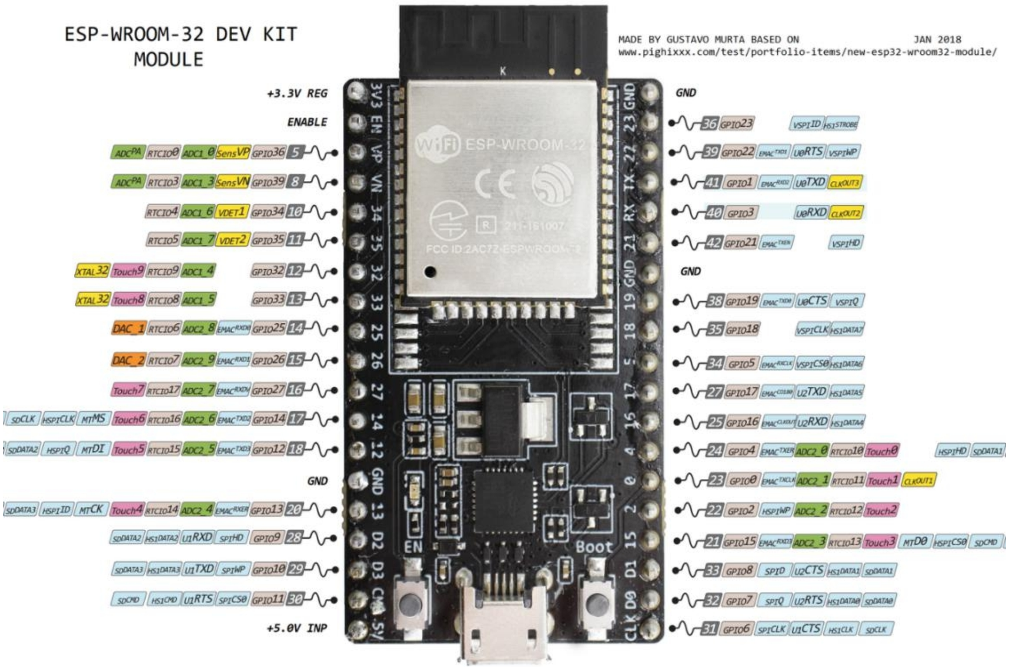


FIG 3:- ESP 32 Pin out diagram

LED Blink

Objective: Turn on and off the built-in led every second.
About the project: It is a simple project that is used to control the built-in led of the arduino board. The arduino board contains a blue built-in led alongside the red power led. The blue led can be controlled through programming. The project turns the led on and off every alternative second.
Requirements: Arduino ESP32, USB cable
Algorithm1: To blink the built-in LED of the board

1.Begin
2.declare the led to blink
3.while (in main loop) do
4.increase counter
5.if (counter is even) then
6.turn off led

7.end
8.Otherwise
9.turn on led
10.End
11. delay of 1 second
12.end
13.End

LED Touch

Objective: Control the built-in led through touch
About the project: The project utilizes the built-in touch sensors of the arduino board. The board has 10 touch pins that can be used for this purpose. The pin read a lower value when touched. So, the led light is turned on when the pin reading is below a certain value(50).

Requirements: Arduino ESP32, USB cable

Algorithm2: To respond to user’s touch by lighting the built-in LED

1. begin
2. declare the led to turn on
3. declare the touch pin to use
4. while (in main loop) do
5. read touch pin data
6. if (touch pin data < 50) then
7. turn on led
8. end
9. otherwise
10. turn off led
11. end
12. end
13. End

Temperature Sensor

Objective: Read the temperature from the internal temperature sensor.

About the project: This project uses the internal temperature sensor that is already available in the board. The built-in temperature sensor gives the current temperature of the chip.This temperature increases with the increase in the time for which the chip is being used. The temperature reading is in Fahrenheit and it needs to be converted to Celsius before it is displayed. The data from the chip is sent to the computer through the serial port to which the chip is connected. The data is then displayed on the serial monitor of the Arduino IDE.

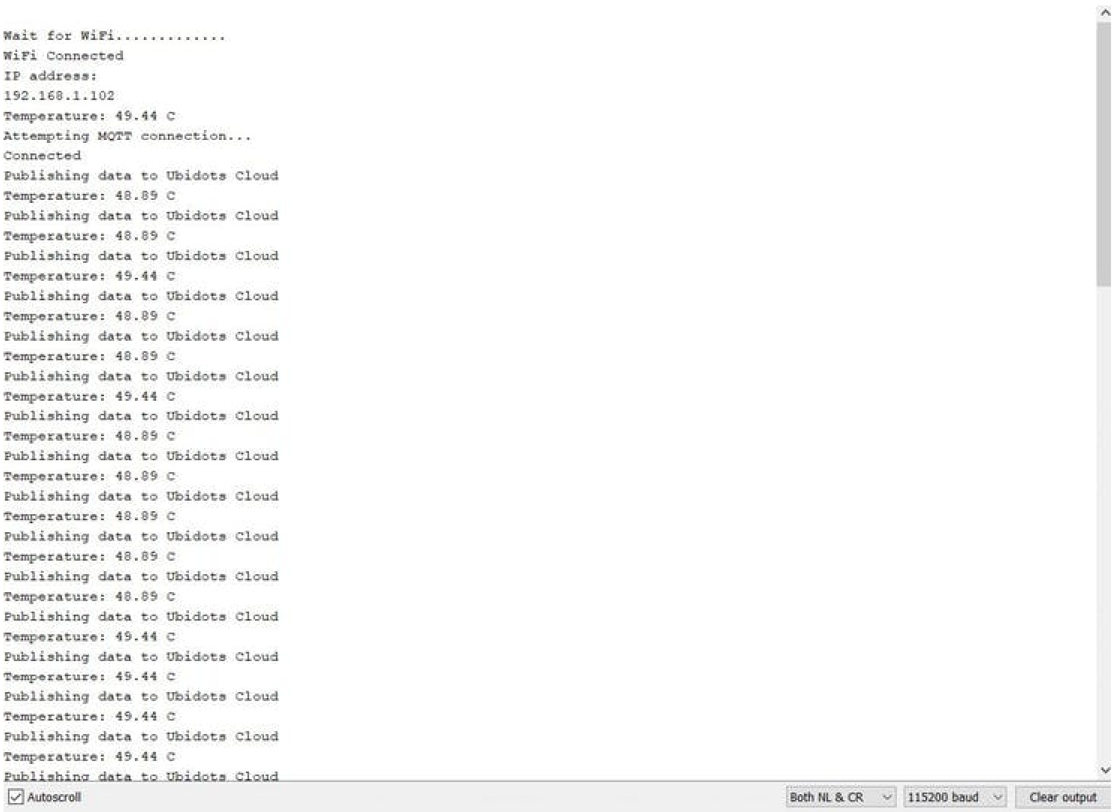


FIG 4:- OUTPUT OF TEMPERATURE OF DEVICE

Requirements: Arduino ESP32, USB cable

Algrithm3: To read the data from internal temperature sensor

1.begin
2.while (in main loop) do
3.read the internal temperature sensor
4.convert the temperature to Celsius
5.print the data on serial monitor
6.delay of 1 second
7.End
8.end

DHT Sensor

Objective: To read the temperature and humidity of the atmosphere

About the project: The project uses an external DHT sensor to record the atmospheric temperature and humidity. The DHT sensor is powered by the 3.3V pin of the arduino. The data pin is connected to the sensor that can be read when the sensor reading is required. The DHT sensor has two varieties, DHT11 and DHT22. The DHT22 is used in this project as it is more reliable and long lasting than the other sensor.

FIG 5:- Output of temperature of atmosphere

Algorithm4: To measure atmospheric temperature and humidity using an external
DH

1.begin
2.declare the data pin to use
3.declare the DHT type used
4.while (in main loop) do
5.read the temperature from the data pin
6.read the humidity from the data pin
7.print the temperature on serial monitor
8.print the humidity on serial monitor
9.delay of 1 second
10.end
11.end

Data Transfer through Wi-Fi

Objective: To transfer data from one chip to another by using a wireless connection like Wi-Fi.

About the project: The project uses the Wi-Fi connectivity of the arduino board. The board has built-in Wi-Fi that operates at 2.4 GHz frequency. To communicate through Wi-Fi, the chip needs to be connected to an external internet. The two chips can communicate with each other if one of them acts as a server chip while the other acts as a client chip. The server chip scans the network range to find if any client chip is nearby. On finding a client chip, it transfers the data to its client every 5 seconds. The client chip broadcasts a local IP so that it can be found by the server. On connection with the server, the client displays the data that it receives on the serial monitor. The project can be extended to create a network with more than 2 chips.

Requirements: 2 Arduino ESP32, USB cable

Algorithm6: To make the board act as server

1.begin
2.start serial communication
3.connect to external network
4.start the Wi-Fi server
5.while (in main loop) do
6.if (client is present) then
7.connect to the client
8.send data to the client
9.print the MAC address of the client
10.close connection
11.end
12.end
13.end

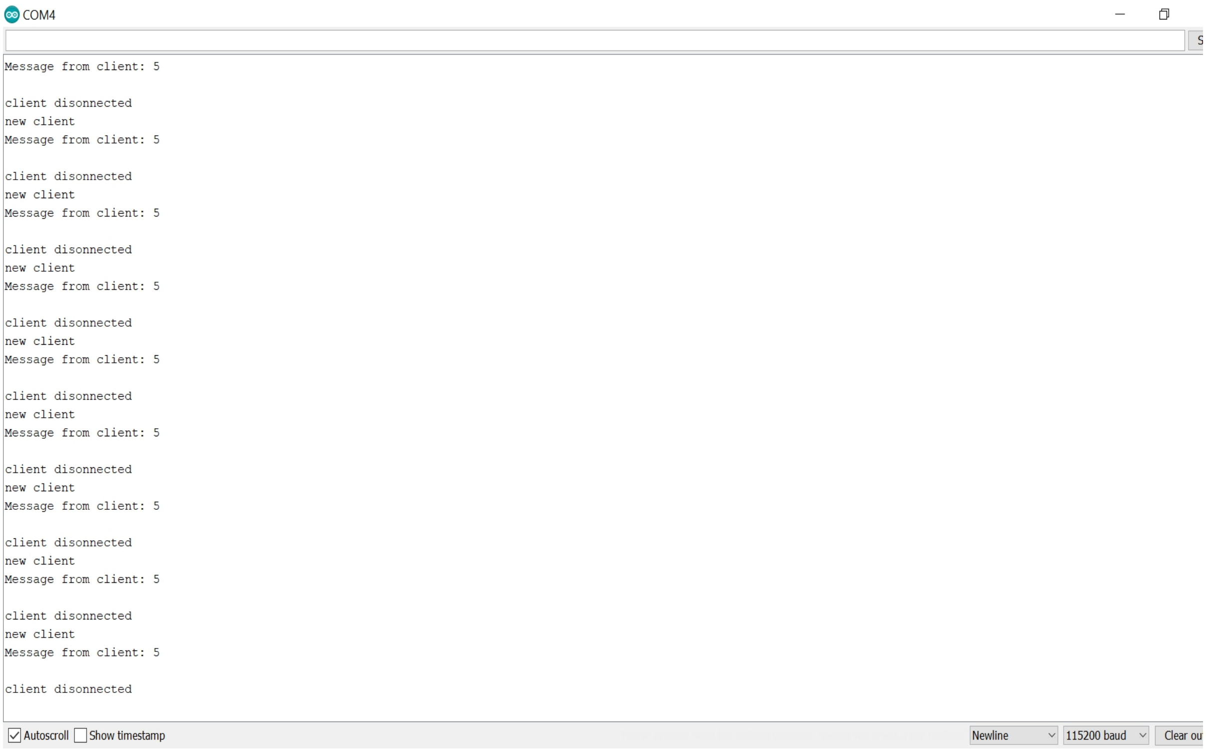


FIG 6:- output of code receiving message

Algorithm7 : To make the board act as receiver

1.begin
2.start serial communication
3.connect to external network
4.broadcast the local IP address of the client
5.while (in main loop) do
6.if (server is connected) then
7.read the line from the server
8.print the line on serial monitor
9.end
10.end
11.End

Application: The chips can be used to send the data from any sensors connected to it wirelessly over Wi-Fi. This will help us in taking sensor data from many different locations without being physically present in all places.

Ad-hoc Communication

Objective: To create a wireless ad-hoc network using the arduino boards as nodes.

About the project: The project uses the arduino boards as the nodes of a wireless ad-hoc network. An ad-hoc network does not require routers or access points like other wireless network. It is decentralized and each node is responsible for the data routing without depending on an external infrastructure. The network is easy to setup and the failure of a few nodes does not affect the working of the network. That is, the nodes can change their routes on failure of a few neighbour nodes.

We create the network with many nodes all of which can both send and receive data. The role of a node in this network is to send its data to its neighbours at regular intervals. Each node also relays any message that it receives, that is, it receives the data from one of its neighbour and forwards the data to its peers. This way the data can have multiple hops before it arrives at a destination which is the receiver node.

The chip is suitable for this purpose as it can be connected to external sensors easily. We do not use the normal Wi-Fi to transfer data as it needs to be connected to an external internet for

transferring data. For this purpose we use the ESP-NOW wifi protocol to transfer data from one chip to another. The esp-now protocol is developed by espressif for peer to peer communication using Wi-Fi. The esp-now is low power and provides a higher range of communication than bluetooth. The protocol uses low power 2.4 GHz that is used with devices like wireless mice. The advantage of esp-now over normal wifi is that it does not require an access point for connecting to the network. Therefore, the nodes can be placed anywhere without worrying about the wifi connectivity. The esp-now has three modes: Master, Slave and Combo. The master mode is responsible to transfer data to the neighbouring slaves. It scans the area in it range to check how many soft access points(softAP) are available. If the master finds the softAP of a slave, it adds it to the list of peers. It can then communicate to it directly without any external resources. The slave mode receives the data from the master. The slave mode creates a softAP that can be identified by any master in its range. The combo mode can act as both a master and a slave. We use this mode of the esp-now for the nodes and the slave mode for the receiver.

Applications of this project: It can be used to make a static as well as dynamic ad-hoc network. The arduino chips act as the nodes of the network. The individual chips can be connected to sensors and the various sensor data can be sent to the receiver node. We explore such an application of this project in which we create a static and a dynamic network using these arduino boards. Each node has an Arduino ESP32 chip and a pollution sensor(MQ-135). These pollution sensors can be attached to the vehicles that can read the pollution caused by the vehicle. The receiver node can be stationary and receive data from different vehicles as they form a vehicular ad-hoc network around the receiver. The receiver node can be connected to a central server and the data from the vehicles can be monitored by the authorities. When a vehicle shows an abnormal rise in the pollution caused by it, the owner of the vehicle can be informed and vehicle can be taken for servicing.

Static network:

A static wireless network has all the nodes stationary. Each node has information about the neighbouring nodes. When each node starts, it scans its neighbourhood to find the surrounding nodes. The nodes can identify the other nodes within its range through their mac addresses. Each node stores the list of the mac address of the neighbouring nodes as the peer information. This scan needs to be done only once in case of a static network. Once the scan is complete, the nodes have two functions: 1. Transmit the data recorded by its sensors at regular time intervals to its neighbors through peer to peer communication. 2. Receive data from its neighbouring node and relay it to the other nodes in the network. The node ignores if it receives multiple copies of the same data. Each node is given an unique id number. This is used by the receiver to find the path travelled by the data to reach it. When data is sent from a node, the node sends it's id number along with the data. The receiver separates the data from the id number of the intermediate nodes. The receiver chip can be connected to the computer so that it can transfer the received information serially to the computer. The serial monitor shows the following details about a data received - the sender node id, the receiver node id, the data value, the time of receiving data and the path(intermediate nodes) the data has travelled to reach the receiver. The static nodes do less work as they scan for neighbours only once making them more energy efficient. In case of any collision of data, the sender node receives a bad acknowledgement from the peer and the data is transferred again.

Dynamic network:

In a dynamic network, the nodes of the network change positions with time. The role of the nodes changes slightly than in the case of a static network. The number of nodes in the range of a node changes with time in case of dynamic network. For example, if the vehicles moving on a road are considered as the nodes of the network, the vehicles may move at different speeds causing a change in the number of vehicles surrounding a single one. Thus, the node in a dynamic network needs to scan for the nodes in its range after a specific cycle time. After the scan is complete the node can easily communicate with its peers. The node stores the addresses of its peers in a list which gets updated at each cycle time. The receiver does not need to move like the nodes for receiving the data as the data can come through multiple hops to the receiver.

Requirements: Arduino ESP32, USB cable

Algorithm8: To make the board node of the static network

1. begin
2. start the node in combo mode
3. scan the network for the neighbouring nodes
4. store the details of the peers in the list
5. register sending callback function
6. register receiving callback function
7. while (in main loop) then
8. read data from the sensor
9. send this data to its peers at equal intervals
10. if (data is received from any neighbour) then
11. forward the data to the peers
12. end
13. end
14. end

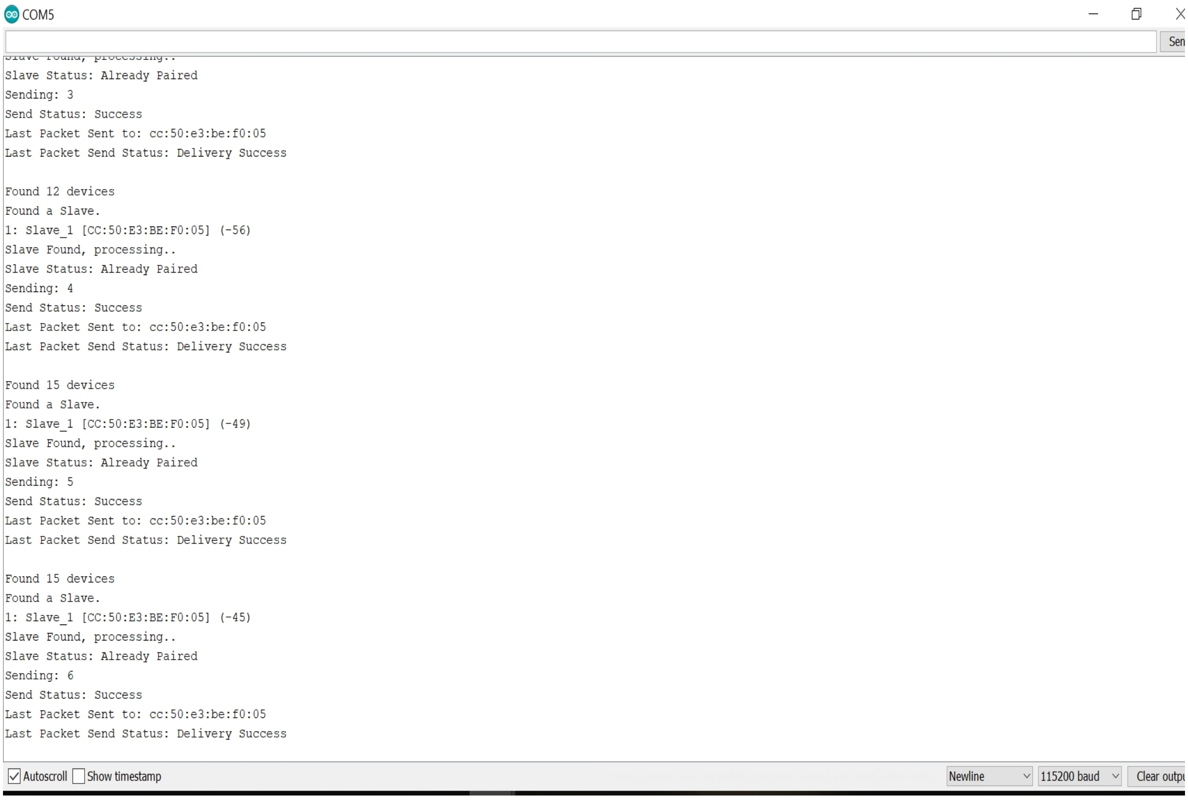


FIG 7:- Output of Master Slave

Algorithm9: To make the board node of the dynamic network

1. begin
2. start the node in combo mode
3. scan the network for the neighbouring nodes
4. store the details of the peers in the list
5. register sending callback function
6. register receiving callback function
7. while (in main loop) then
8. read data from the sensor
9. send this data to its peers at equal intervals
10. scan for peers after cycle time
11. if (data is received from any neighbour) then
12. forward the data to the peers
13. end
14. end
15. end

Algorithm10: To make the board receiver node

1. begin
2. start the node in soft AP mode
3. register receiving callback function
4. while (in main loop) then
5. if (data is received from any neighbour) then
6. check if the data has already been received
7. if (data already received) then
8. drop the data
9. end
10. otherwise
11. display the data
12. display the time of receiving data
13. display the sender node id
14. display the receiver node id
15. display the path the data has travelled
16. end
17. end
18. end
19. end

Conclusion

As we have seen the arduino ESP32 has a wide variety of applications. We have tried to explore some of these applications. These boards can be helpful in the several fields like military, traffic control, pollution control, fire detection etc. In the military, the nodes can be connected to camera and other sensors to get information about the enemy. In the fire detection, the nodes can be connected to the temperature sensor to detect a sudden fire. In the field of pollution control, the nodes can be connected with pollution sensors and gas detectors so that atmospheric pollution can be sensed and controlled. The projects done above can be extended in the future for more applications. The possibilities of using these micro controller boards are infinite and I would like to work on them in the future as well.