Project on IoT Development using an ESP32 Platform

PERFORMED BY:
PIYUSH BHART
2018UGEC002

Reference no.:
VT20203769

Acknowledgement

Thanks to the company "TATA STEEL" that it gave me an opportunity to do a summer training 20' on

IoT Development using an ESP32 platform.

First of all, I would like to express deep sense of gratitude towards the TATA STEEL Prashikshan Team who permitted me to undergo this summer training 20'. I would like to express my special thanks to *Mr. Prabal Patra (Head of instrumenta*tion and control department) and my project guide *Mr. Chitresh Kundu (Sr. assistant)* who prepared our training schedule & helped us to complete our training.

At last I would like to thank my training batch-mates for their support and co-operation through this training by maintaining peace and decorum in the group.

Content

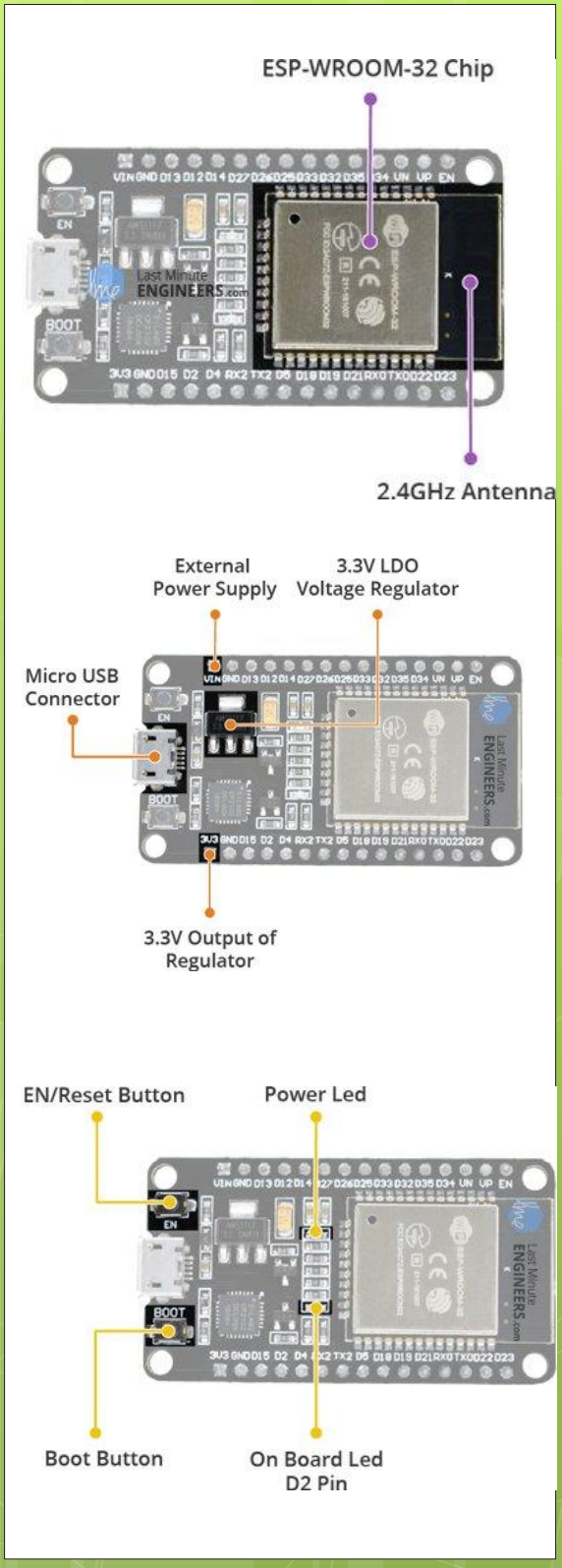
Introduction

Internet of Things (IoT) is a basic concept that brings a mythic world to reality.

With the concept of IoT, certain objects have the ability to transfer data over the network automatically without human intervention and regardless of distance. Manufacturing of Microcontrollers brought this revolution. It is somewhat like a mini computer on a chip that functions as an electronic circuit controller and generally can store programs to perform certain functions.

Few years back, ESP8266 took the embedded IoT world by storm. For less than $3, you could get a programmable, Wi-Fi-enabled microcontroller being able to monitor and control things from anywhere in the world. Now Espressif (The semiconductor company behind the ESP8266) has released a perfect super-charged upgrade: the ESP32

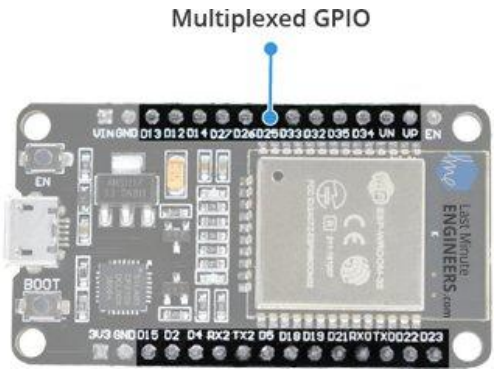
ESP32 is in the series of low power and low cost on chip microcontroller. It comes up with already integrated Wi-Fi support and also features Bluetooth 4.0 (BLE/Bluetooth Smart. It is especially aimed to provide versatility, robustness and reliability in a large number of applications i.e. perfect for just about any IoT project.



About ESP32

The development board equips the ESP-WROOM-32 module containing Tensilica Xtensa® Dual-Core 32-bit LX6 microprocessor. This processor is similar to the ESP8266 but has two CPU cores (can be individually controlled), operates at 80 to 240 MHz adjustable clock frequency and performs at up to 600 DMIPS (Dhrystone Million Instructions Per Second).

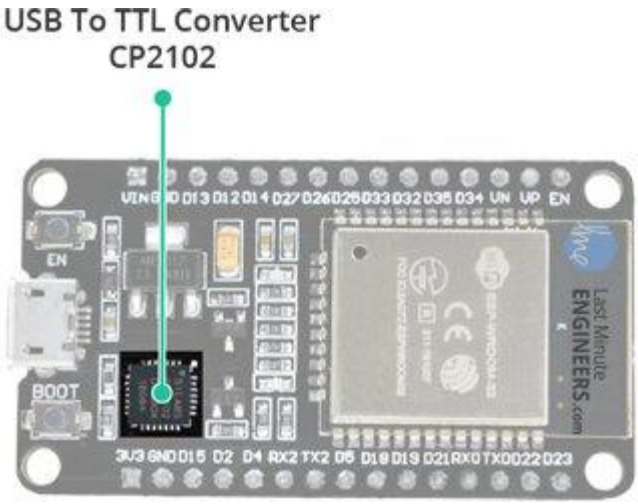
The ESP32 Integrates 802.11 b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a Wi-Fi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it.



Although the ESP32 has total 48 GPIO pins, only 25 of them are broken out to the pin headers on both sides of the development board. These pins can be assigned to all sorts of peripheral duties, including:

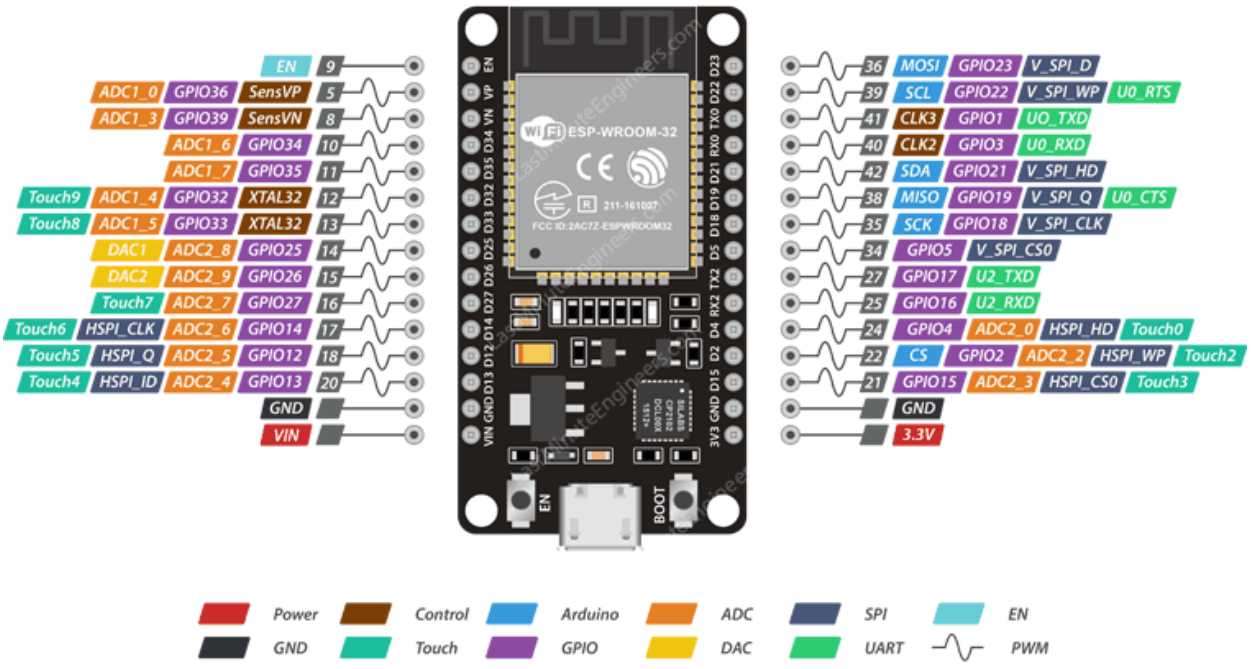
- **15 ADC channels** – 15 channels of 12-bit SAR ADC's. The ADC range can be set, in firmware, to either 0-1V, 0-1.4V, 0-2V, or 0-4V
- **2 UART interfaces** – 2 UART interfaces. One is used to load code serially. They feature flow control, and support IrDA too!
- **25 PWM outputs** – 25 channels of PWM pins for dimming LEDs or controlling motors.
- **2 DAC channels** – 8-bit DACs to produce true analog voltages.
- **SPI, I2C & I2S interface** – There are 3 SPI and 1 I2C interfaces to hook up all sorts of sensors and peripherals, plus two I2S interfaces if you want to add sound to your project.
- **9 Touch Pads** – 9 GPIOs feature capacitive touch sensing.

Thanks to the ESP32's pin multiplexing feature (Multiple peripherals multiplexed on a single GPIO pin). Meaning a single GPIO pin can act as an ADC input/DAC output/Touch pad.

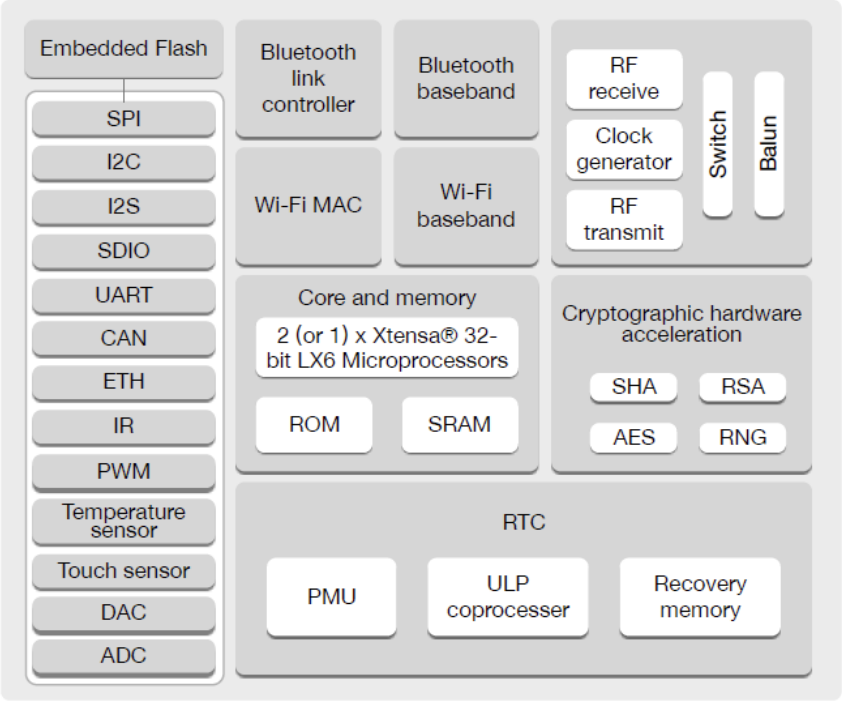


The board includes CP2102 USB-to-UART Bridge Controller from Silicon Labs, which converts USB signal to serial and allows your computer to program and communicate with the ESP32 chip.

ESP32 Development Board Pin out



ESP32 Function Block Diagram



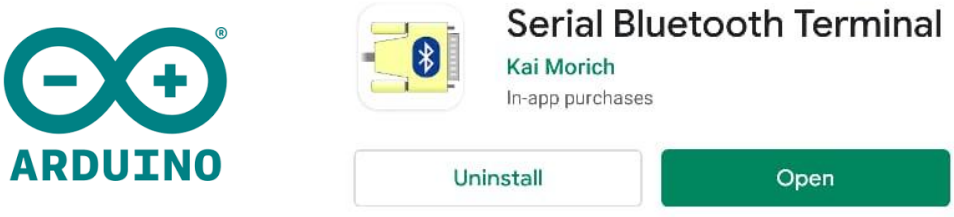
Basic Idea

- Concept of this project is completely based upon the development of IoT in Security field using an ESP32 platform.
- It provides a linkage between the Sever and the Client via Bluetooth.
- ESP32 will provide a Serial Bluetooth Server which is associated with a Lock Door and a Pass Key.
- A Client can get into the range of the server and can easily connect with the server using a Serial Terminal Bluetooth app.
- Client will type the Pass Key on his phone
- If the Pass Key provided by the Client is correct, the door will be unlocked.
- Else providing a wrong Pass Key will not open the door.
- If we can demonstrate the operation of an LED using the same method. Means we can also operate any power driven components like low power motor, smart lock, etc.

Switching an LED ON if and only if the given Pass Key by the client is CORRECT.

Using an Arduino IDE we have coded an ESP32 according to the particular requirements. Following code with proper explanation is provided on the upcoming slides.

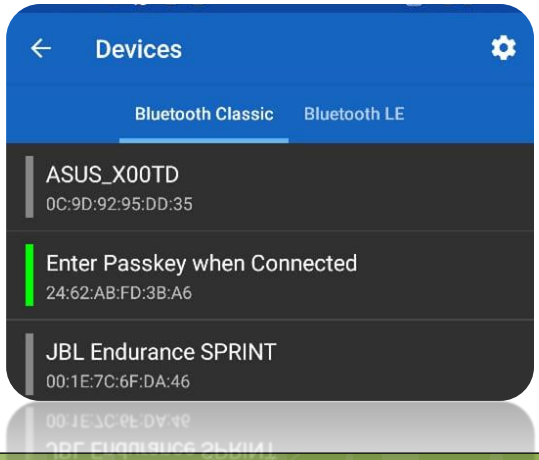
After uploading the code. ESP32 creates a bluetooth server in which clients can connect it easily using a Serial Bluetooth Terminal app which is easily available on playstore for free.

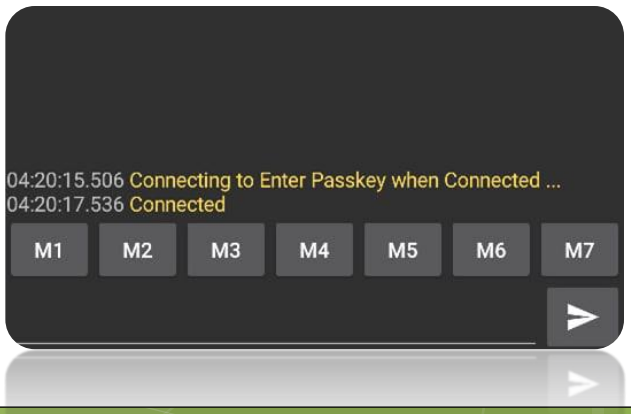


Client has to open the app after installing and select the bluetooth device which is named by the esp32 device name.

In my case it is “Enter Passkey when Connected”.

Connected message appears on the clients device after which he can enter the Pass Key.





Raw Code

```
//This creates a bridge between Server and the Client via Bluetooth

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

int ledpin = 2;
BluetoothSerial SerialBT;

void setup()
 {
  Serial.begin(115200);
  pinMode(ledpin,OUTPUT);
  SerialBT.begin("Enter Pass Key when Connected");
  Serial.println("The device started, now you can pair it with bluetooth!");
  Serial.println("Enter Pass Key");
}
```

```
int i = 0;
int v = 1;
void loop()
{
if (SerialBT.available())
  {
    char x = SerialBT.read();
    Serial.println(x);
    if (x == 'P')
    {
      if (i == 0)
      {
       digitalWrite(ledpin, HIGH);
       SerialBT.println("Correct Password");
        SerialBT.println("Visitor No.:");
        SerialBT.println(v);
       SerialBT.println("Kindly Enter the same to lock it again");
       i =1;
       v++;
      }
      else if (i ==1)
      {
        digitalWrite(ledpin, LOW);
        i = 0;
      }
     }
 }
  delay(20);
}
```

Code Explanation

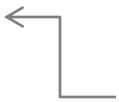
```
//This creates a bridge between Server and the Client via Bluetooth
#include "BluetoothSerial.h" //Library file
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

int ledpin = 2; //GPIO pin2

BluetoothSerial SerialBT;

void setup()
{
  Serial.begin(115200); //baud at which ESP32 works.
  pinMode(ledpin,OUTPUT);
  SerialBT.begin("Enter Passkey when Connected");//Bluetooth device name displayed on the clients device.
  Serial.println("The device started"); // Msg on the  SerialMonitor.
  Serial.println("Enter Pass Key");   // Msg on the  SerialMonitor.
}

int i = 0; //Value responsible for LED or Lock to be HIGH or LOW.
int v = 1; //counting number of times the door is opened only.
void loop()
{
  if (SerialBT.available())
```

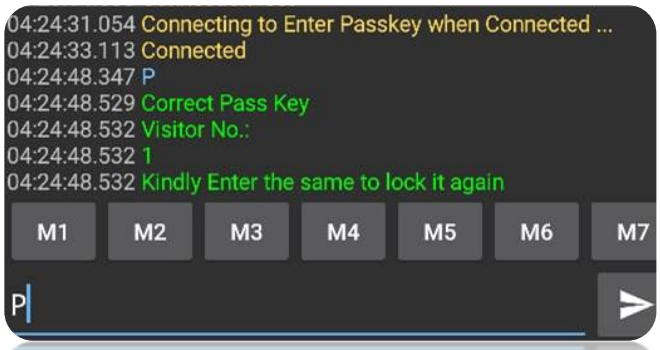


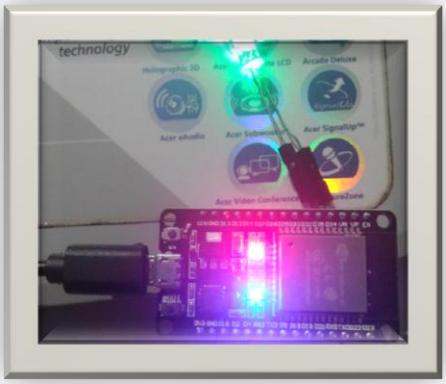
```
void loop()
{
  if (SerialBT.available())
  {
    char x = SerialBT.read(); //Storing the Pass Key recieved by the client.
    Serial.println(x); //Printing the Pass key on the SerialMonitor.
    if (x=='P') // Pass Key to blink the LED is "P".
    {
      if (i==0) //IF the current value of LED is LOW, by entering correct Pass Key.
                //It will Turn the LED HIGH.
      {
      digitalWrite(ledpin, HIGH);
      SerialBT.println("Correct Pass Key");
      SerialBT.println("Visitor No.:");
      SerialBT.println(v);          //Visitor Count.
      SerialBT.println("Kindly Enter the same to lock it again");
      i=1;
      v++;
      }
      else if (i==1) //IF the current value of LED is HIGH, by entering correct Pass Key...
                     //It will Turn the LED LOW again i.e back to normal.
      {
        digitalWrite(ledpin, LOW);
        SerialBT.println("Thank You!");
        i=0;
      }
     } //while entering Incorrect Pass Key, It will show no msg. i.e. You can try again.
  }
  delay(20);
}
```

Results

- Led Anode pin is connected to GPIO pin no. 2 and the cathode is connected to the ground. Initially The LED is LOW as no command is given to the server.

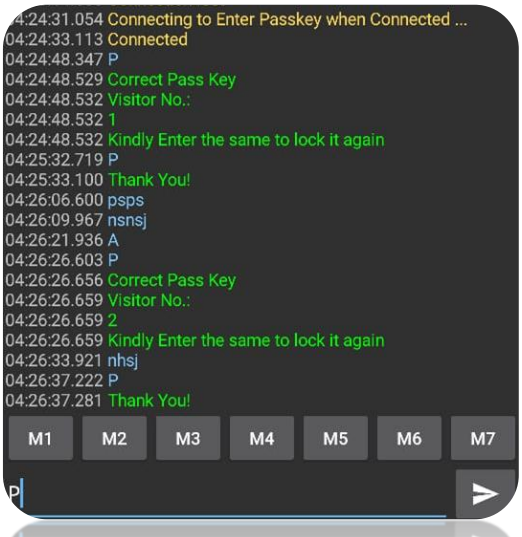
- When I entered Pass Key "P" which is the correct Pass key. The LED turns HIGH also giving the output on the client device i.e. Correct Pass Key and No. of visitors.





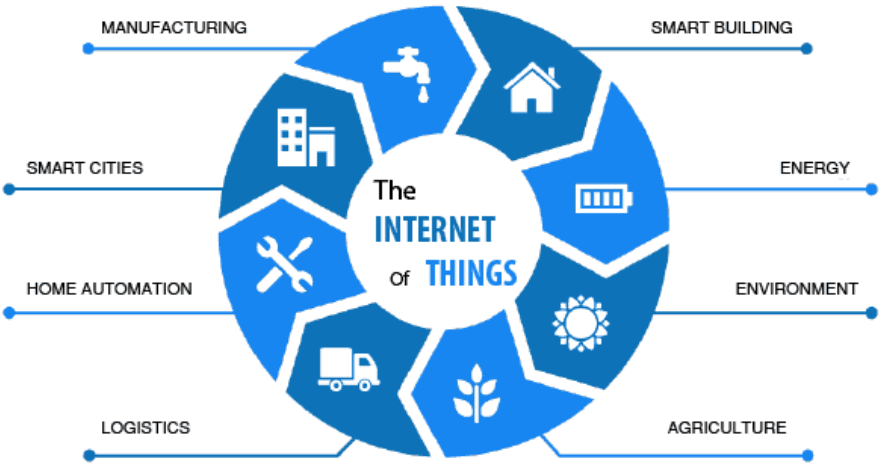
- To turn the LED LOW you again Have to type in the same Pass Key. This will show a thank you msg.

- Any other Pass Key will not respond To anything also showing no msg on the Client device. Examples shown psps, nsnsj, A. Whereas,
Correct Pass Key i.e.:"P" again lights the LED also increasing value of No. of visitors.

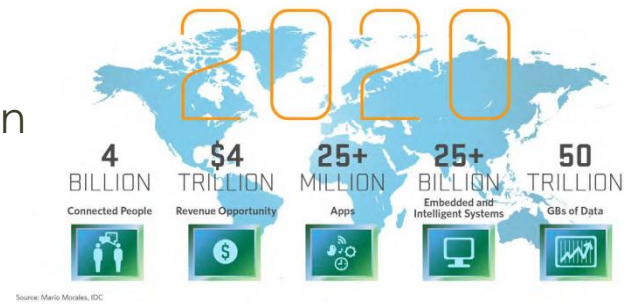


Into the Future

The future of IoT has the potential to be limitless. Advances to the industrial internet will be accelerated through increased network agility, integrated **artificial intelligence** (AI) and the capacity to deploy, automate, orchestrate and secure diverse use cases at hyper scale.



ESP32 microcontroller chip is just another step toward an IoT based world. No doubt In near future we are looking Forward for more such Gadget components that is going to make our day to day life much more easier and smarter.



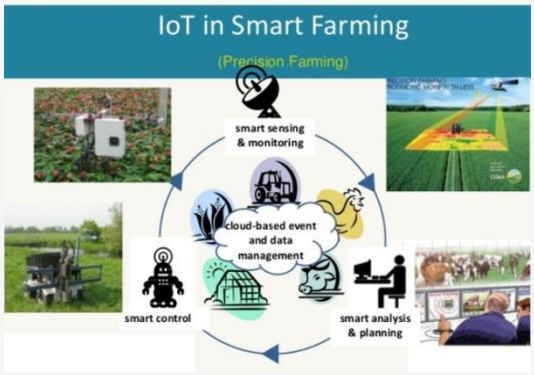
Fields like Medical, Agriculture, industrial are always Looking forward to such inventions holding a great market Place.

Increased demand in smart health care products has give boost to smart healthcare product market.

Products like medicine alarm, fitness band, health monitor, etc. are some leading products

Smart farming is replacing traditional farming to minimize the losses and increasing productivity.

Soil humidity checker, animal tracking devices are few of the applications being used in Agriculture.

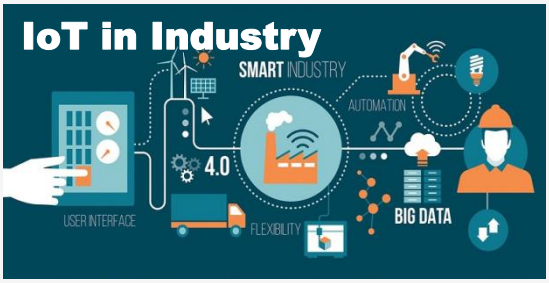


Smart city project has give boost to smart home and smart traffic control system.

From sensor manufacturer to traffic controller everyone is ready to take profit of this growing IoT segment

Replacement of humans with robots are needs of every industry to make it robust and centralized system .

In assembly lines or generation of products.



Conclusion

The Project is all about Learning IoT development using an ESP32.

After this project we saw that IoT is inevitable with a vast future and full of opportunities.

ESP32 is an excellent option for IoT devices due to its performance and price. With the baud of 115200 WROOM32 it features Wi-Fi, Bluetooth and BLE . We saw how we can operate an LED just by operating our phone. Requires just a code that fulfills your requirement.

Also the new BLE feature allows ESP32 to work on low power devices such as Health band, Streaming music, Monitoring things and what not.

With no doubt Inventions of microcontrollers has turned this world to a new phase. And this is going to play a major role in future IoT systems and Embedded projects.