



What's IOT ?

Project Idea

IOT Alert System

Strategy

Code



Final Project

# What's IOT ?

The internet of things, or IoT, is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

# How Does it work ?

An IoT ecosystem consists of web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environments. IoT devices share the sensor data they collect by connecting to an IoT gateway or other edge device where data is either sent to the cloud to be analyzed or analyzed locally. Sometimes, these devices communicate with other related devices and act on the information they get from one another. The devices do most of the work without human intervention, although people can interact with the devices -- for instance, to set them up, give them instructions or access the data.

# Our Idea ?

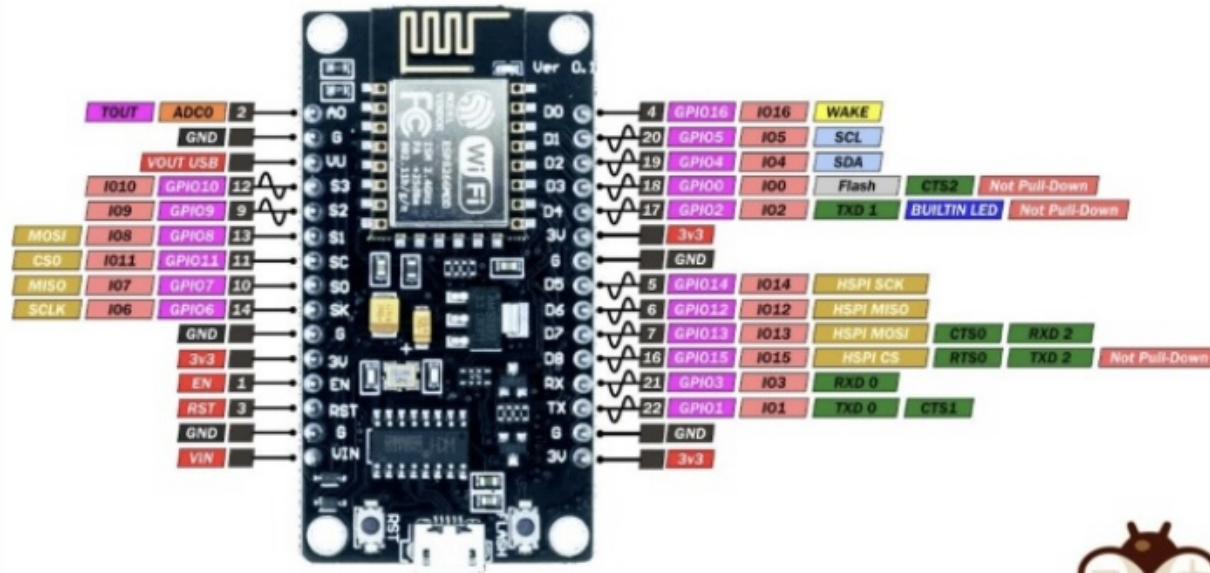
Creating Alert system for every press so when the production technician is about to finish, he sends an alert to the supervisor to prepare the next die for him.

This will help reducing changeover time waste and increase Press capacity.

# ESP8266 Board Pinout

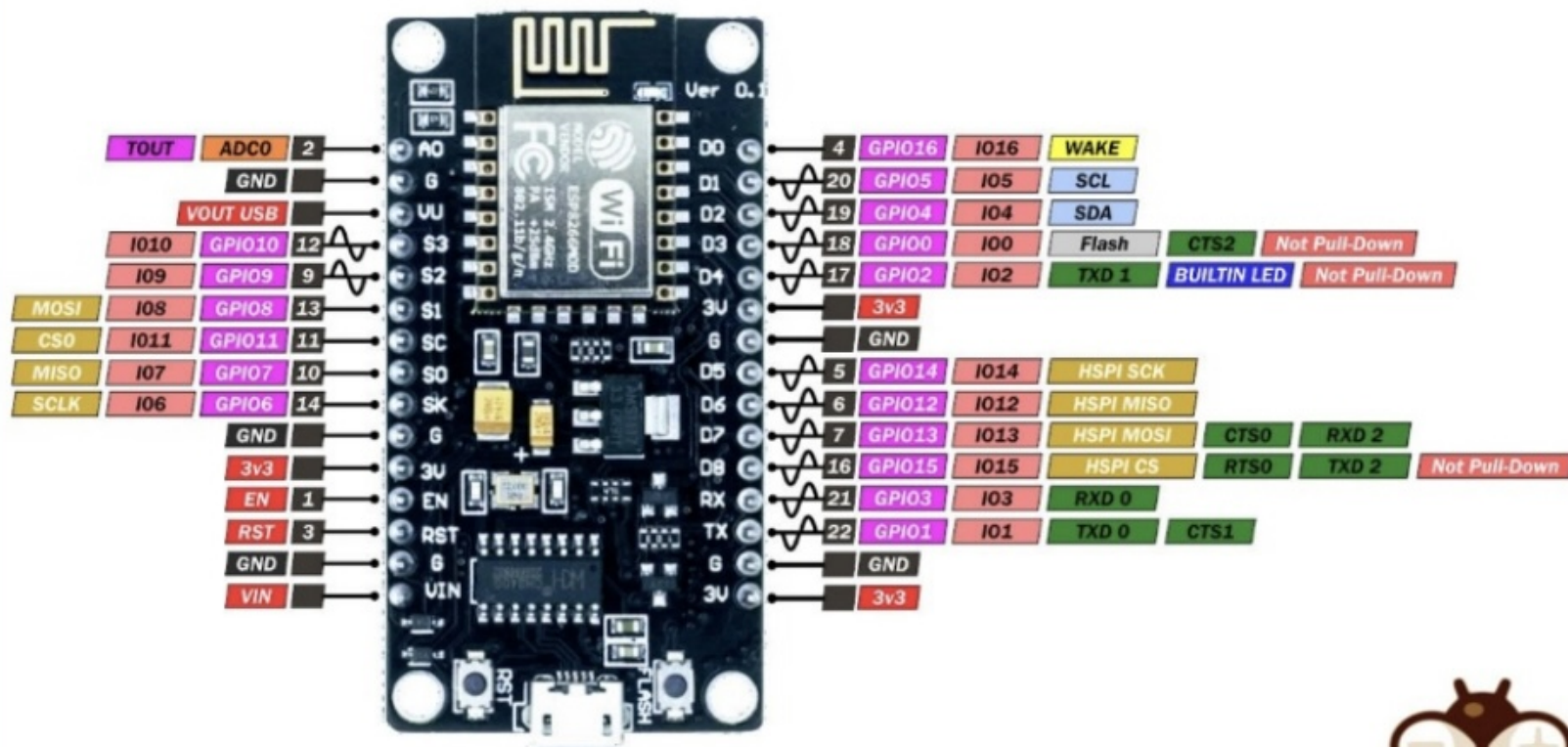
NodeMCU v3 CH340

PINOUT



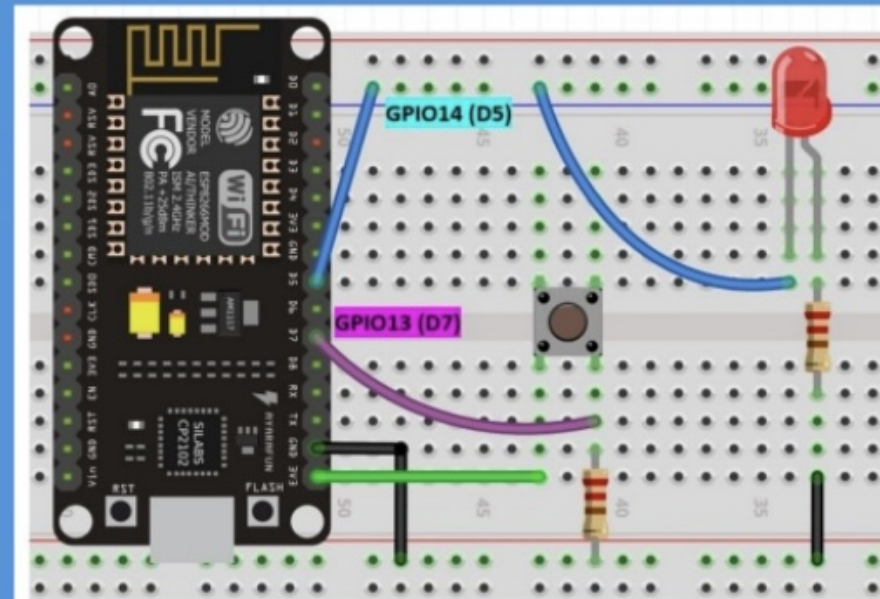


## PINOUT



# Project Components?

- ESP8266 NodeMCU x 1
- Push Button Switches x 2
- 330 Ohms Resistor x 2
- 10K Ohms Resistor x 1
- Led (RED and GREEN) x 2
- Breadboard
- Jumper wires
- 5V/1A mobile charger as power supply



# What we will do ?

- Create Alert Buttons for every press to alert the supervisor.
- Adjust Mobile App for Supervisors to receive alerts from labors to reduce changeover time waste and increase capacity.
- Connect Mobile app to all presses wirelessly using Wifi to communicate.



# Project Code

```
/*
*****
Simple push notification example

App project setup:
  Push widget

Connect a button to pin 2 and GND...
Pressing this button will also push a message! ;)
*****/

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID      "TMPLxxxxxx"
#define BLYNK_DEVICE_NAME     "Device"
#define BLYNK_AUTH_TOKEN      "YourAuthToken"

// Comment this out to disable prints and save space
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "YourNetworkName";
char pass[] = "YourPassword";

void notifyOnButtonPress()
{
  // Invert state, since button is "Active LOW"
  int isButtonPressed = !digitalRead(2);
  if (isButtonPressed) {
    Serial.println("Button is pressed.");
  }
}
```

```
// Note:
// We allow 1 notification per 5 seconds for now.
Blynk.notify("Yaaay... button is pressed!");

// You can also use {DEVICE_NAME} placeholder for device name,
// that will be replaced by your device name on the server side.
//Blynk.notify("Yaaay... {DEVICE_NAME} button is pressed!");
}

}

void setup()
{
  // Debug console
  Serial.begin(115200);

  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);

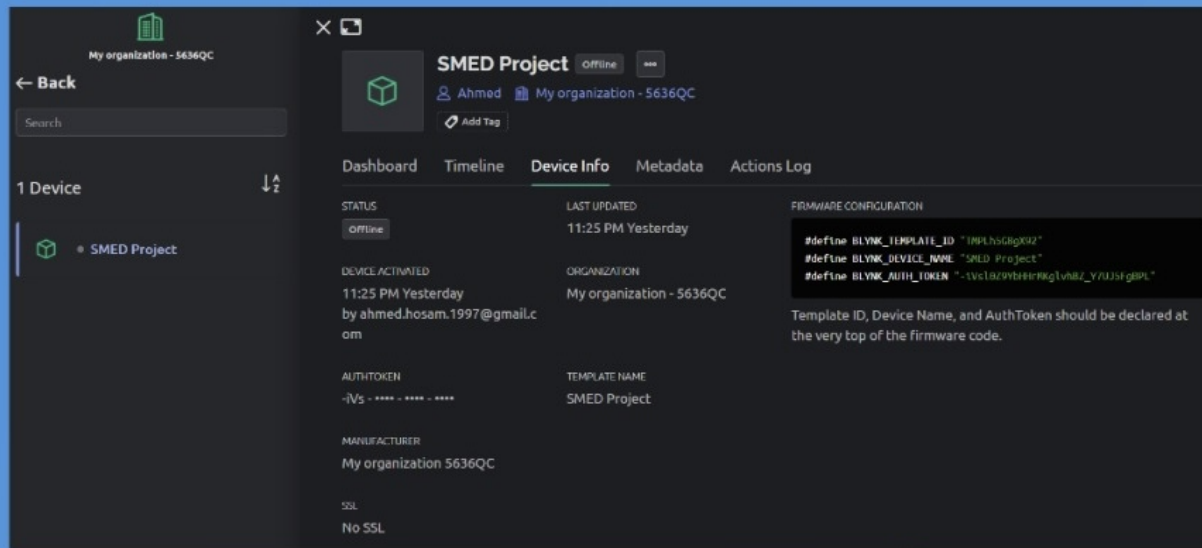
  // Setup notification button on pin 2
  pinMode(2, INPUT_PULLUP);
  // Attach pin 2 interrupt to our handler
  attachInterrupt(digitalPinToInterrupt(2), notifyOnButtonPress, CHANGE);
}

void loop()
{
  Blynk.run();
}
```

# Code Explanation

```
// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID "TMPLXXXXXX"
#define BLYNK_DEVICE_NAME "Device"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
```

This Part must be included first in our code and it's taken from the device info in Blynk Website



The screenshot displays the Blynk web interface for a device named 'SMED Project'. The interface is in a dark theme. On the left, there's a sidebar with a 'Back' button and a search bar. Below the search bar, it shows '1 Device' and a list item for 'SMED Project'. The main content area has tabs for 'Dashboard', 'Timeline', 'Device Info' (which is active), 'Metadata', and 'Actions Log'. The 'Device Info' tab shows the following details:

- STATUS:** Offline
- LAST UPDATED:** 11:25 PM Yesterday
- DEVICE ACTIVATED:** 11:25 PM Yesterday by ahmed.hosam.1997@gmail.com
- ORGANIZATION:** My organization - 5636QC
- AUTH TOKEN:** -IVs - \*\*\*\* - \*\*\*\* - \*\*\*\*
- TEMPLATE NAME:** SMED Project
- MANUFACTURER:** My organization 5636QC
- SSL:** No SSL

On the right side of the 'Device Info' tab, there's a 'FIRMWARE CONFIGURATION' section. It contains a code block with the following definitions:

```
#define BLYNK_TEMPLATE_ID "TMPL5G8gX02"
#define BLYNK_DEVICE_NAME "SMED Project"
#define BLYNK_AUTH_TOKEN "-1Vc1a291b44F8g1vMBZ_YRUS1g8PL"
```

Below the code block, a note states: 'Template ID, Device Name, and AuthToken should be declared at the very top of the firmware code.'

Activate Windows  
Go to Settings to activate Windows.

# Code Explanation

```
char auth[] = BLYNK_AUTH_TOKEN;  
  
// Your WiFi credentials.  
// Set password to "" for open networks.  
char ssid[] = "YourNetworkName";  
char pass[] = "YourPassword";
```

This Part of code is responsible for the WIFI Credentials configurations. we insert first the authorization code we got from device info, then we insert WIFI SSID and Password.

# Code Explanation

```
void notifyOnButtonPress()
{
    // Invert state, since button is "Active LOW"
    int isButtonPressed = !digitalRead(2);
    if (isButtonPressed) {
        Serial.println("Button is pressed.");

        // Note:
        // We allow 1 notification per 5 seconds for now.
        Blynk.notify("Yaaay... button is pressed!");

        // You can also use {DEVICE_NAME} placeholder for device name,
        // that will be replaced by your device name on the server side.
        //Blynk.notify("Yaaay... {DEVICE_NAME} button is pressed!");
    }
}
```

This is a function to be called in ISR for interrupt, as it has the logic we need to send notification from the button.