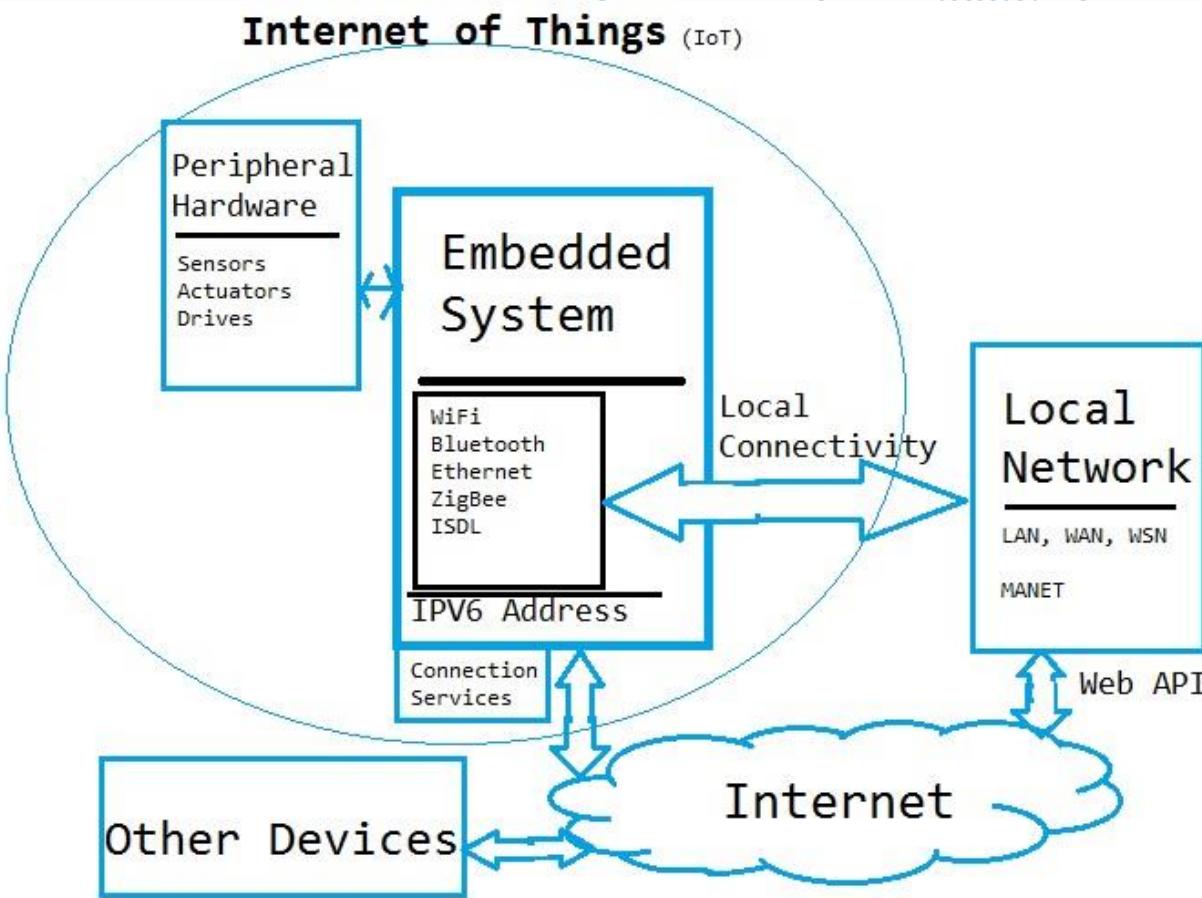


IoT Using NodeMCU

Dr. Swarna Priya RM
SITE

Internet of Things

- “IoT is a **network of physical objects** with **embedded electronics** that **collect and share data**”

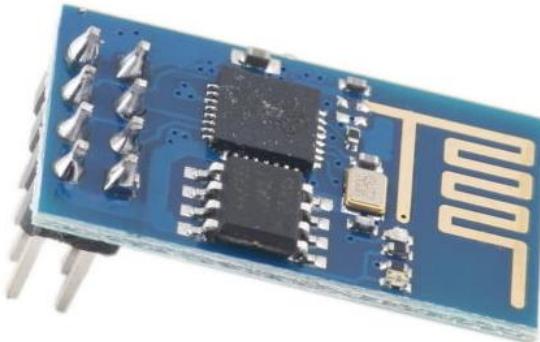


Components for IoT

- ESP8266 module
- NodeMCU 1.0
- Arduino UNO wifi Shield
- OTA WeMos D1

ESP8266 module

- Can be programmed by itself
but usually connected to
Arduino

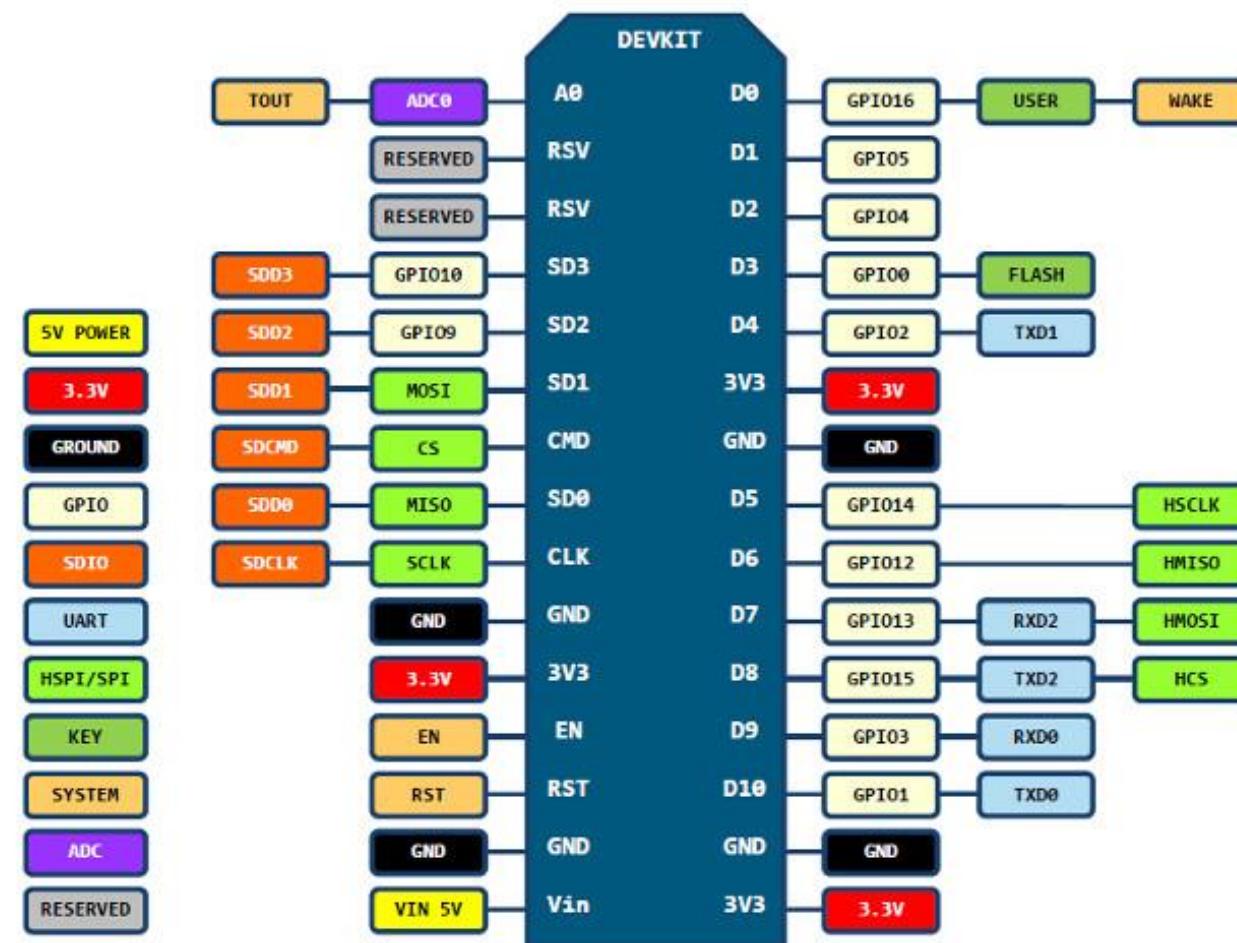


NodeMCU 1.0



- Built-in ESP8266, "ESPduino"
- Works with Arduino IDE, not officially though
- 12 V regulator for input voltage
- 13 GPIO
- **Pins work with 0 - 3,3 V!**
- Good tutorial:
<http://www.makeuseof.com/tag/meet-arduino-killer-esp8266/>

NodeMCU 1.0 Pinout



D0(GPIO16) can only be used as gpio read/write, no interrupt supported, no pwm/i2c/ow supported.

How to connect?

- Cloud
 - NearBus?
- Ready made
 - Blynk
- Server
 - DIY?

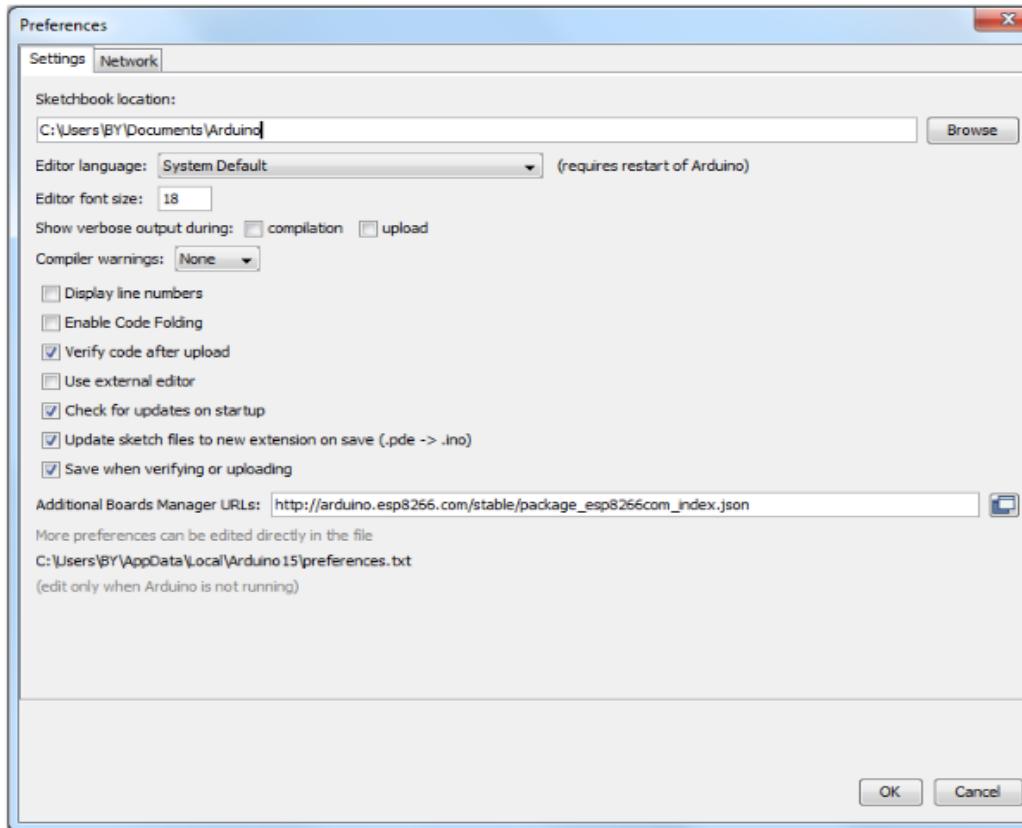
Blynk for phone

- Download the app for your phone



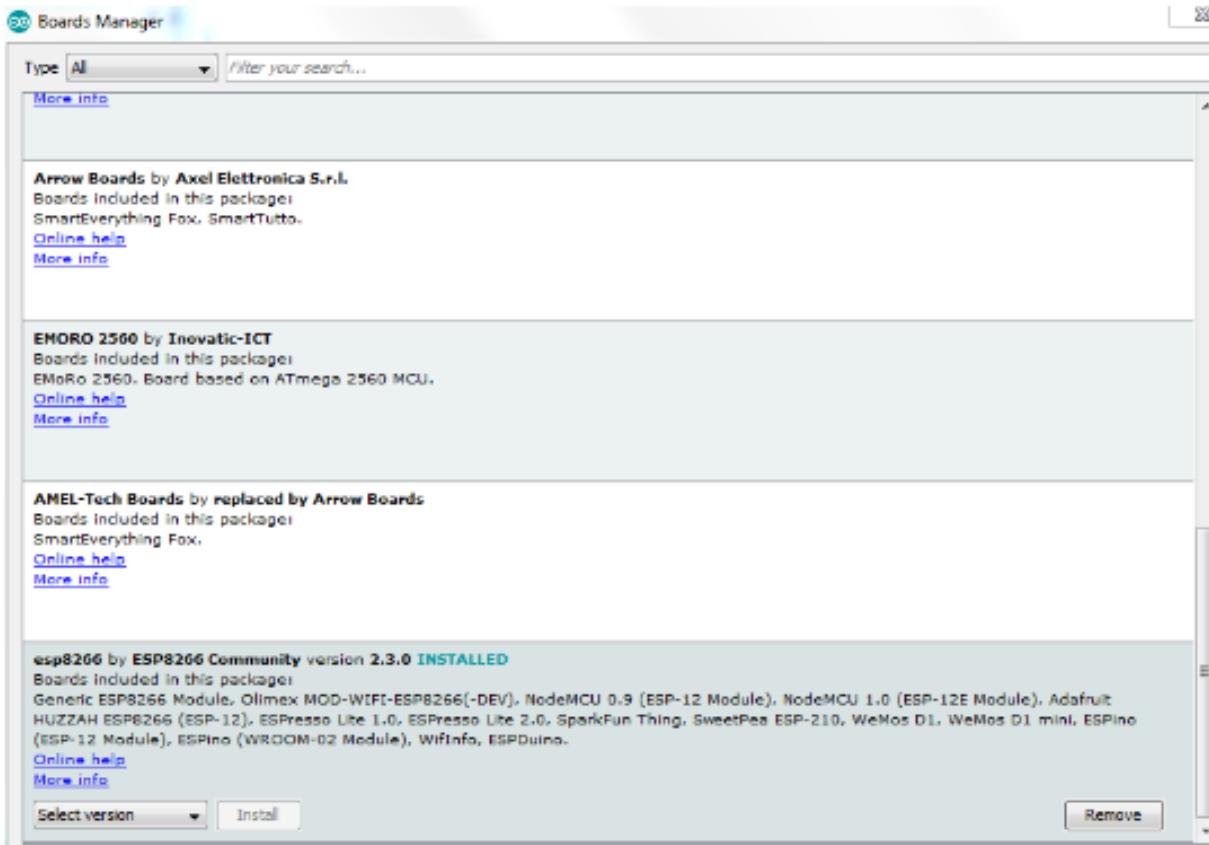
Install ESP8266 Package

- http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Click File -> Preferences
- Paste the above link in Additional Board Managers URL



Installing in Board Manager

- Click Tools -> Boards -> Board Manager
- Search for ESP8266 and click install.
- Click Sketch -> Include Library -> Manage Libraries
- Search for Blynk and click install.



Using NodeMCU

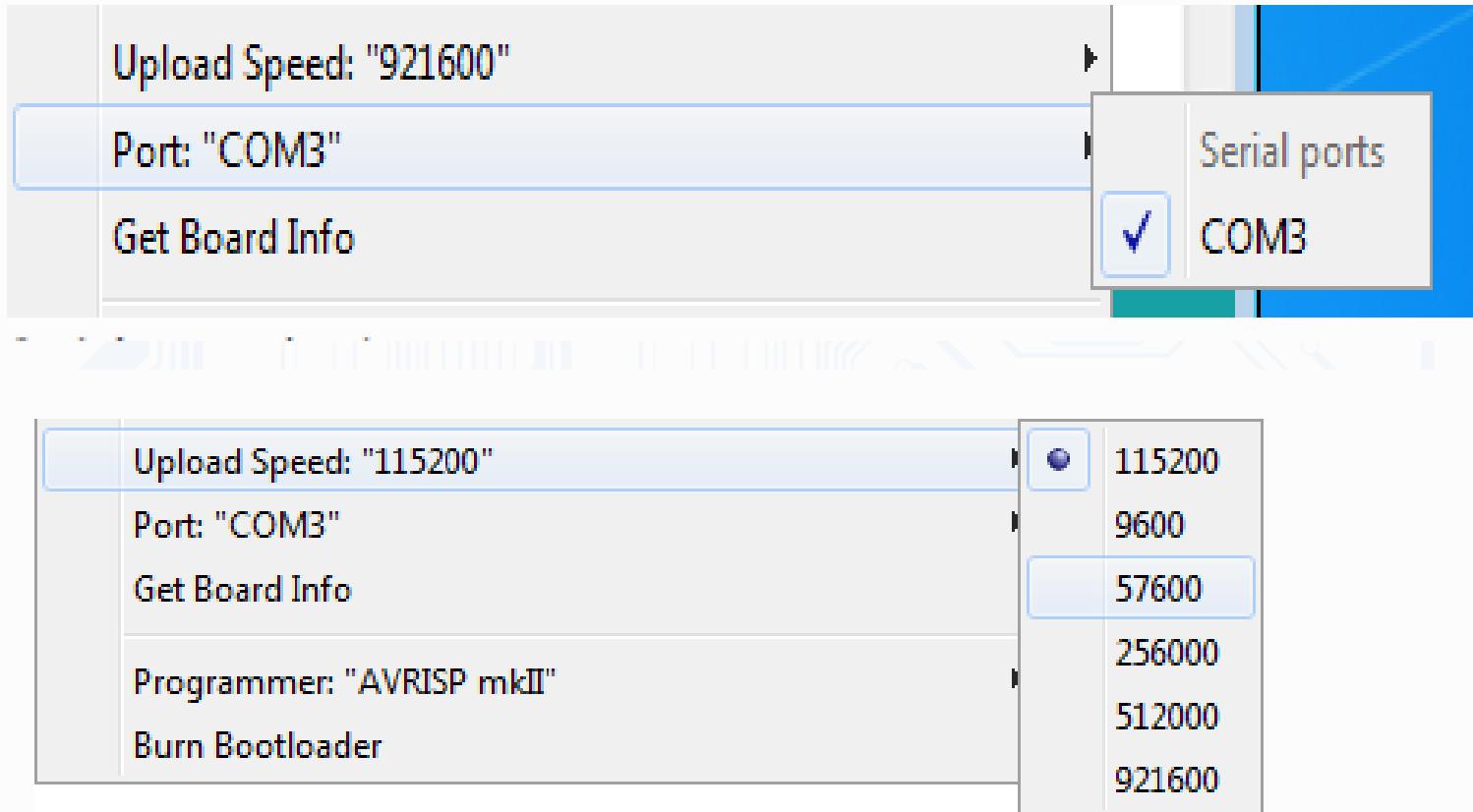
The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Blink | Arduino 1.6.9
- Menu Bar:** File Edit Sketch Tools Help
- Sketch Area:** Displays the `Blink` sketch code.
- Tools Menu:** Shows various options like Auto Format, Archive Sketch, etc., with the **Board:** option highlighted.
- Board Selection Dialog:** A dropdown menu listing various ESP8266 boards, with **NodeMCU 1.0 (ESP-12E Module)** selected.
- Status Bar:** Shows **Uploading...**

Board selection

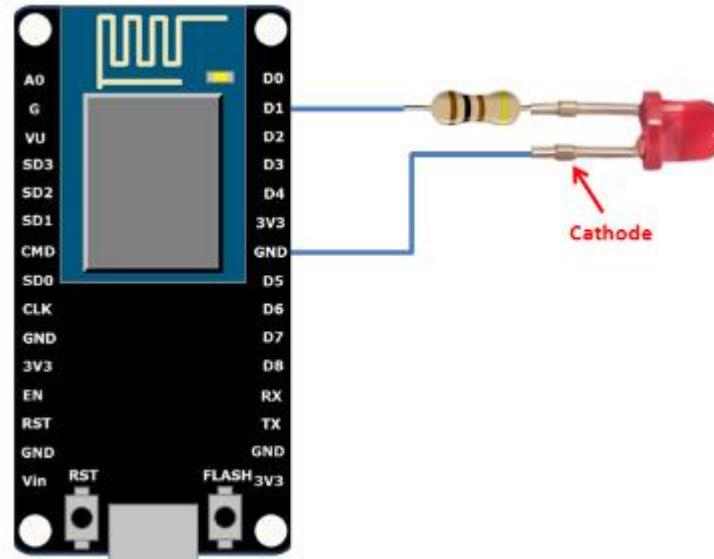
Generic ESP8285 Module
ESPDuino (ESP-13 Module)
Adafruit HUZZAH ESP8266
ESPresso Lite 1.0
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 2.0
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
SparkFun ESP8266 Thing
SparkFun ESP8266 Thing Dev
SweetPea ESP-210
WeMos D1 R2 & mini
WeMos D1(Retired)
ESPino (ESP-12 Module)
ThaiEasyElec's ESPino
WifInfo

Selecting the Port and Upload Speed

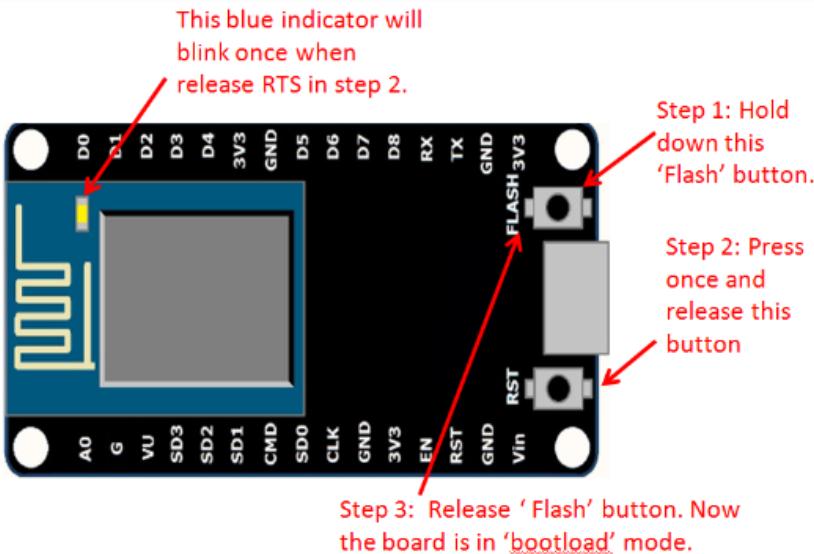


Simple LED Application

```
Void setup()
{
pinMode(5, OUTPUT); // GPIO05, Digital Pin D1
}
void loop()
{
digitalWrite(5, HIGH);
delay(900);
digitalWrite(5, LOW);
delay(500);
}
```



Putting Board in Bootload Mode



The screenshot shows the Arduino IDE interface with a sketch named "blinky". The code is as follows:

```
blinky
void setup() {
  pinMode(5, OUTPUT); // GPIO05, Digital Pin D1
}

void loop() {
  digitalWrite(5, HIGH);
  delay(900);
  digitalWrite(5, LOW);
  delay(500);
}
```

Below the code, the IDE displays the upload process:

Uploading...

WARNING: Spurious .github folder in 'Adafruit IO Arduino' library
WARNING: Spurious .tests folder in 'Adafruit IO Arduino' library
WARNING: Spurious .github folder in 'Adafruit MQTT Library' library
WARNING: Spurious .github folder in 'Adafruit IO Arduino' library
WARNING: Spurious .tests folder in 'Adafruit IO Arduino' library
WARNING: Spurious .github folder in 'Adafruit MQTT Library' library

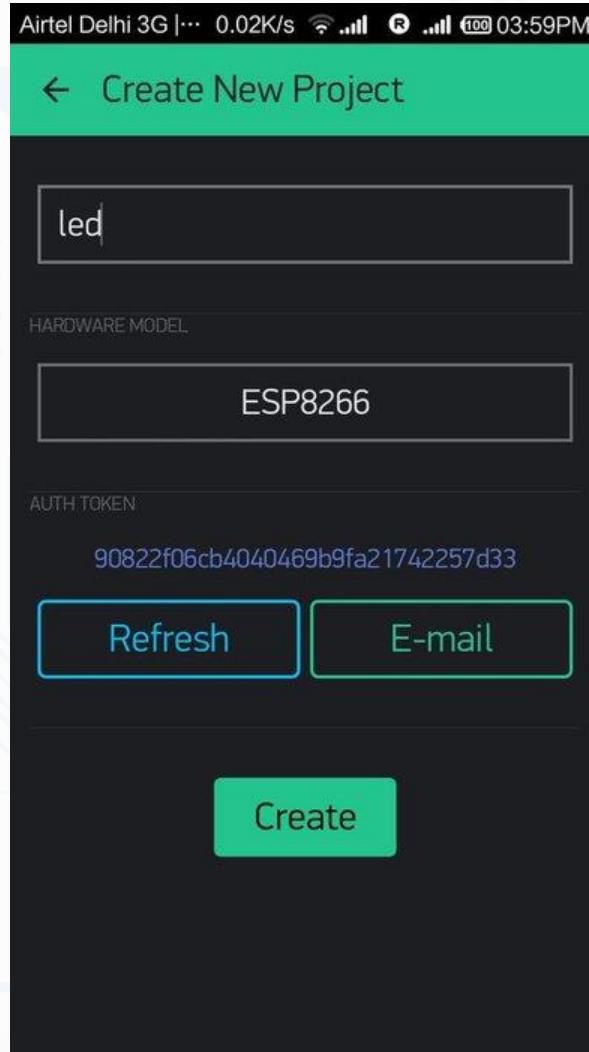
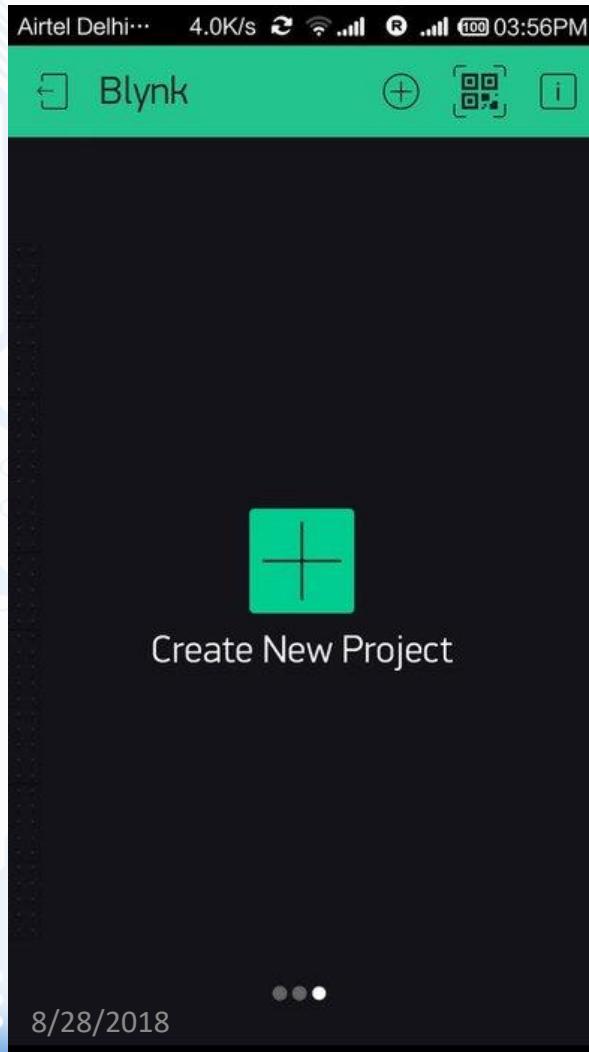
Sketch uses 222,197 bytes (51%) of program storage space. Maximum is 434,160 bytes.
Global variables use 31,572 bytes (38%) of dynamic memory, leaving 50,348 bytes for local v

Uploading 226352 bytes from C:\Users\BY\AppData\Local\Temp\buildb7f3357d9ec338fa2a4043584dd

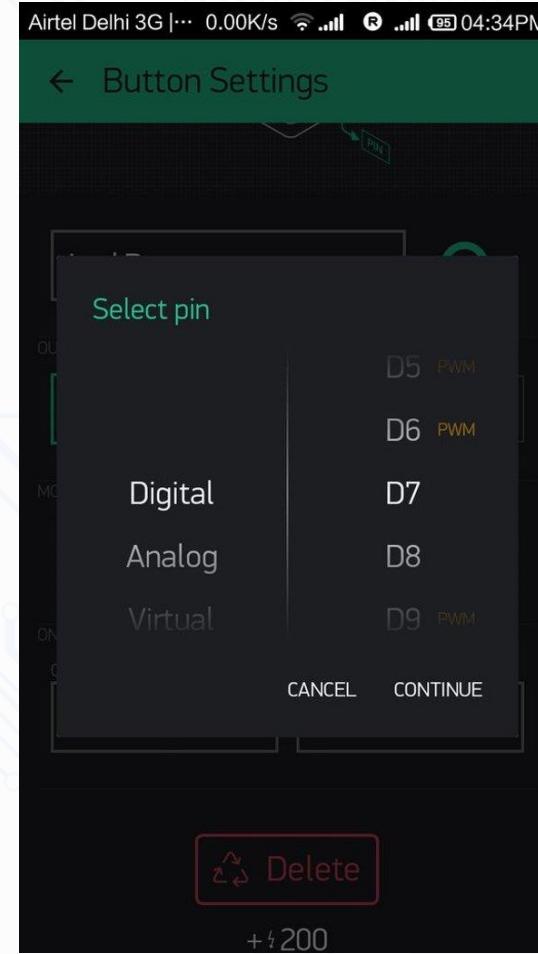
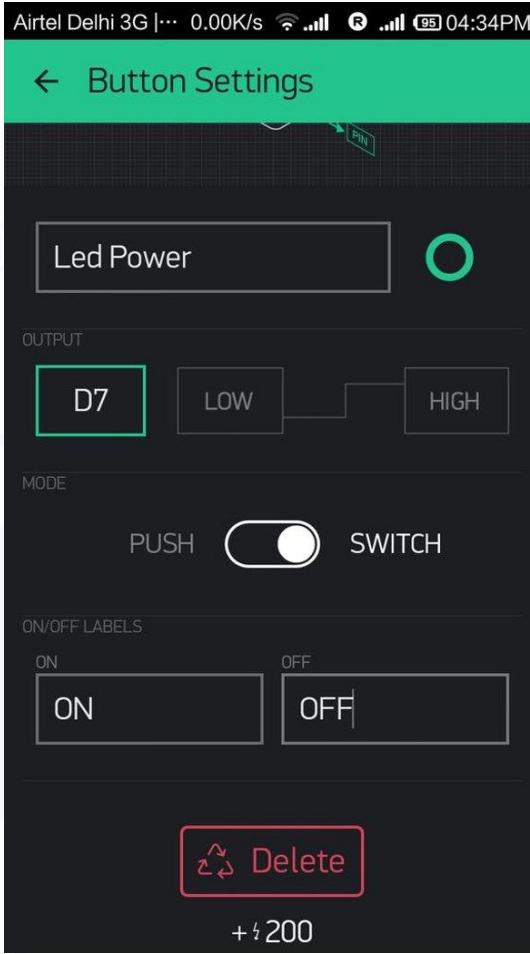
[36%]

Control LED with Mobile

- Setting up Blynk



Setting up Blynk - contd.



```
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
char auth[] = "aa3bc63cf7784a3a958a7f740cbac07e";
char ssid[] = "swp";
char pass[] = "swp123";
```

```
#define ledPin D7
```

```
void setup()
{
    Serial.begin(115200);
    delay(10);
    Blynk.begin(auth, ssid, pass);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
}
```

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