



Introduction to NodeMCU



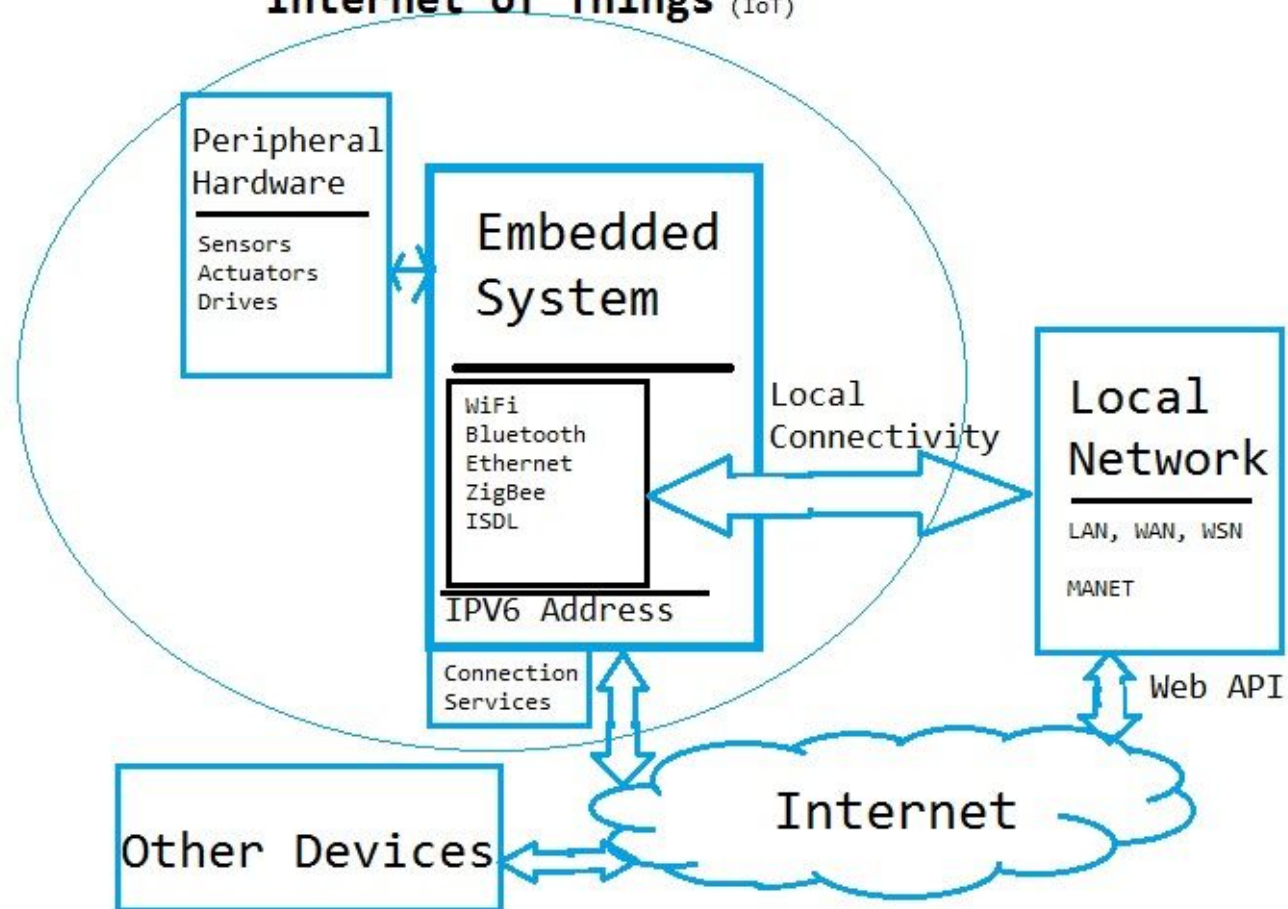
By: Murtaza Ameen
School of Electronics, DAVV, Indore



Brief Introduction to IoT

- Ecosystem of connected physical objects.
- Objects to be sensed or controlled remotely across existing network.
- Things with IP.
- Interaction with Environment.
- Examples: smart microwaves, self-driving cars, wearable fitness device etc.

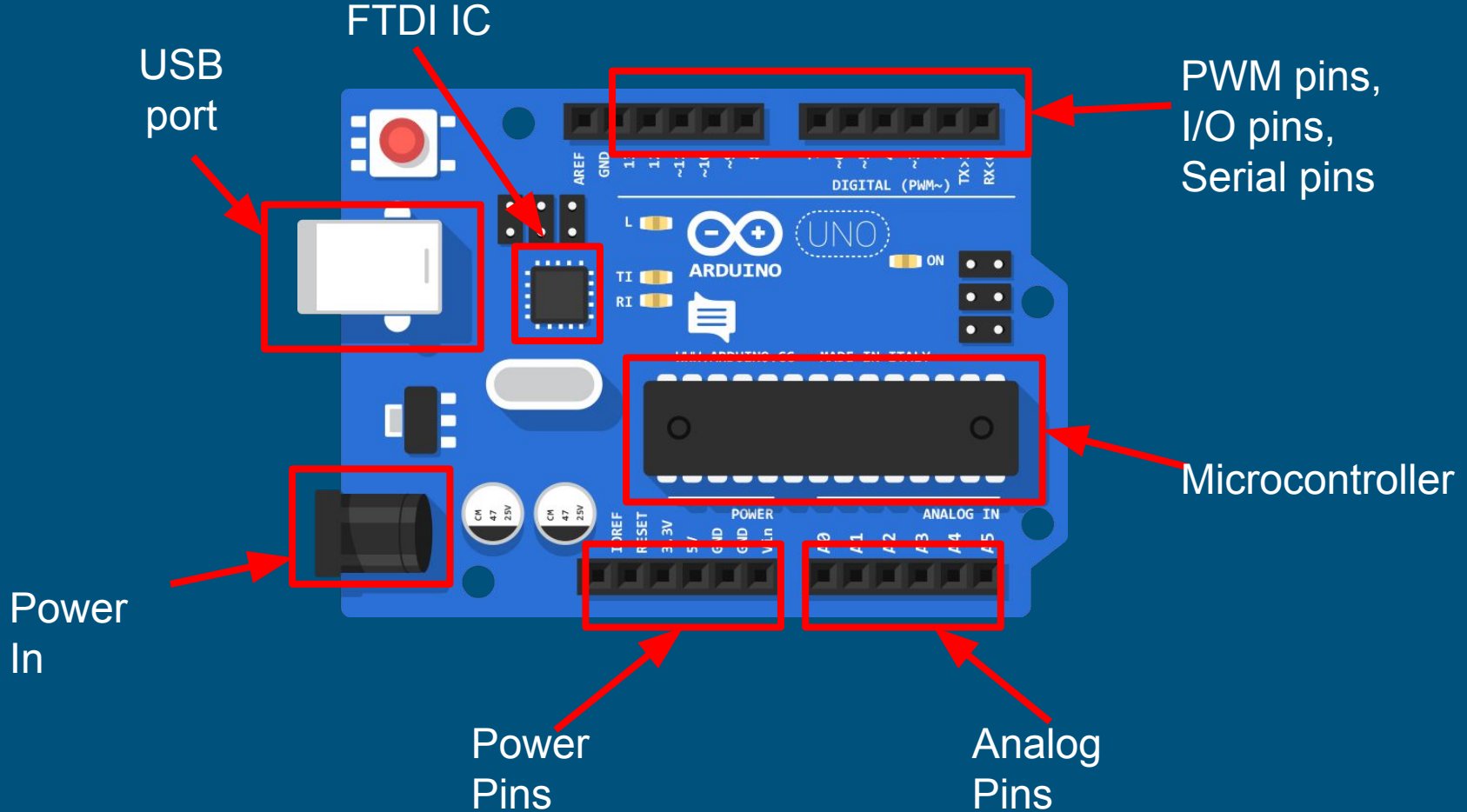
Internet of Things (IoT)



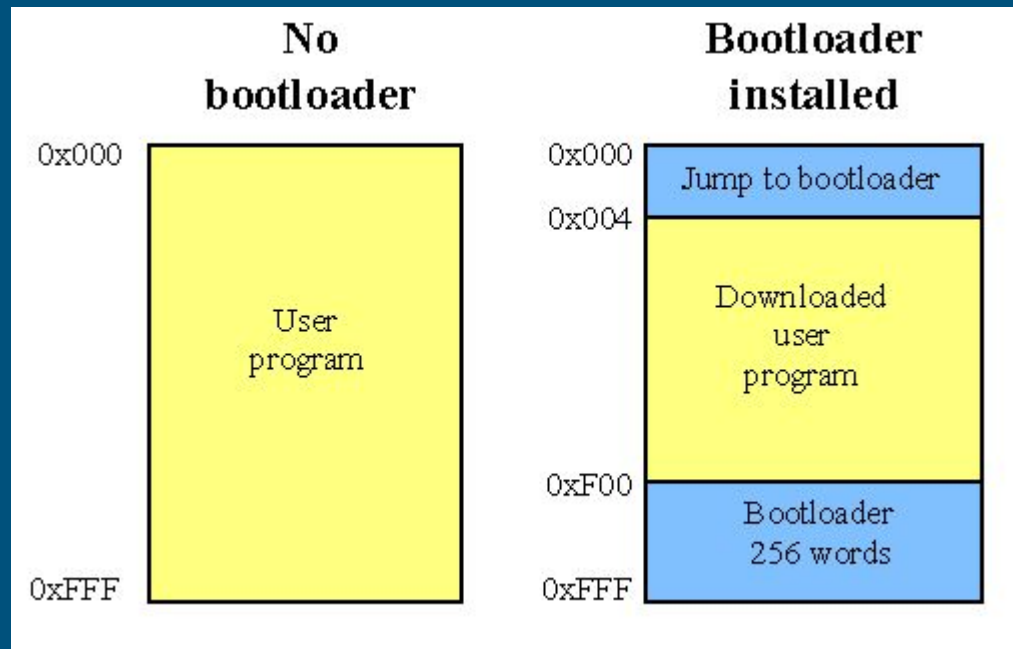
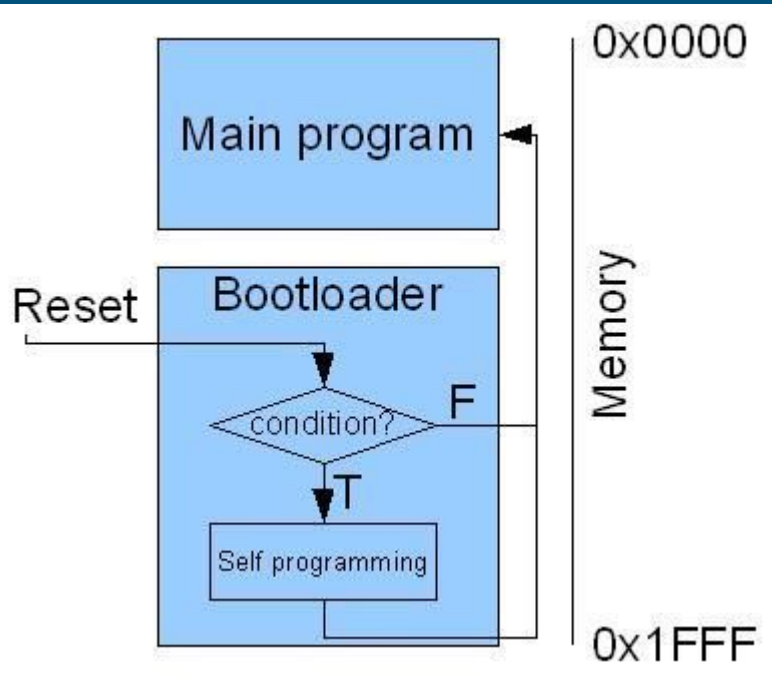
Why we need IoT ?

- The problem is, people have limited time, attention and accuracy.
- they are not very good at capturing data about things in the real world.
- If we had computers that knew everything there was to know about things -- using data they gathered without any help from us -- we would be able to track and count everything and greatly reduce waste, loss and cost.
- We would know when things needed replacing, repairing or recalling and whether they were fresh or past their best

Arduino



Bootloader

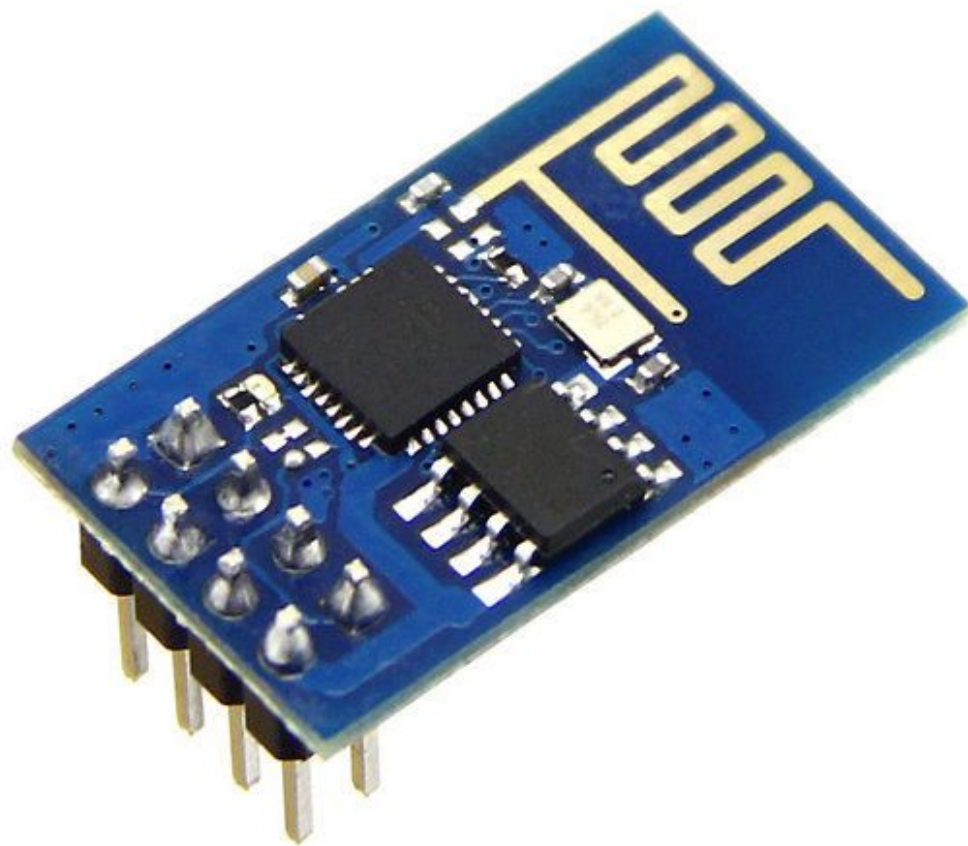


FEATURES OF ARDUINO UNO ATMEGA328

- Microcontroller: **ATmeg328**.
- Operating Voltage: **5V**.
- Input Voltage: **7-12V**.
- Digital I/O Pins: **14** (of which 6 provide PWM output).
- Analog Input Pins: **6**.
- DC Current: **40 mA**.
- DC Current: **50 mA**.
- Flash Memory: **32 KB**.
- SRAM: **2 KB**.
- EEPROM: **1 KB**.
- Clock Speed: **16 MHz**.

Points to remember

- Arduino is a kind of development board for ATmega microcontrollers.
- Arduino IDE provides simple functions to control the I/O pins.
- Bootloader is just a program which helps you to program microcontroller serially.
- FTDI



ESP - 01



❶ GND

❷ GPIO2

❸ GPIO0

❹ RXD

❺ TXD

❻ CH_PD

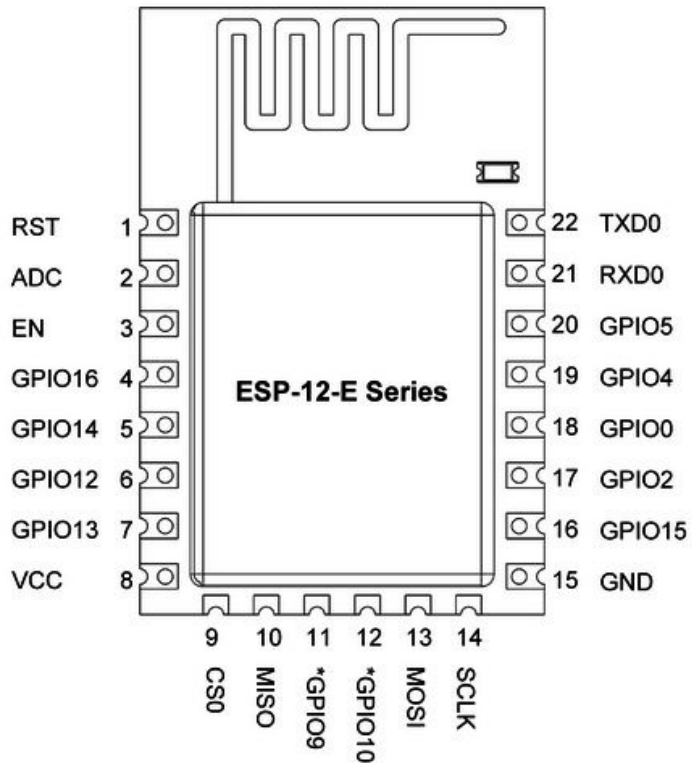
❼ RESET

❽ Vcc

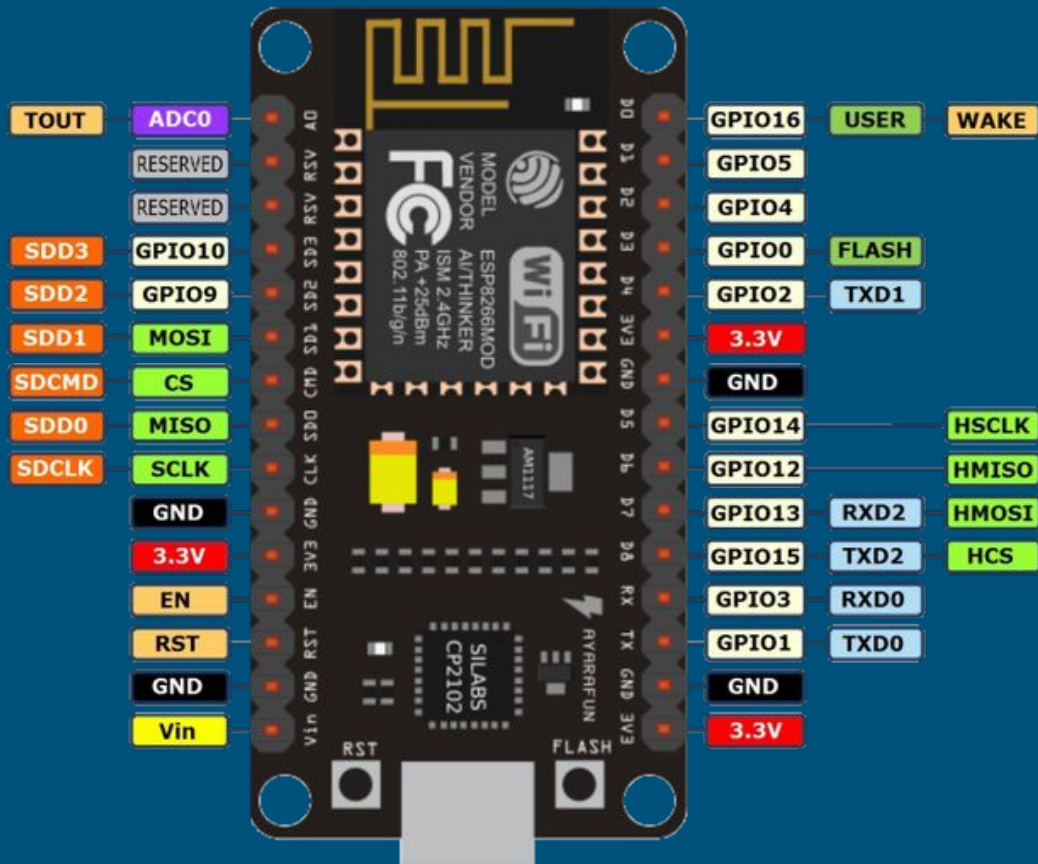
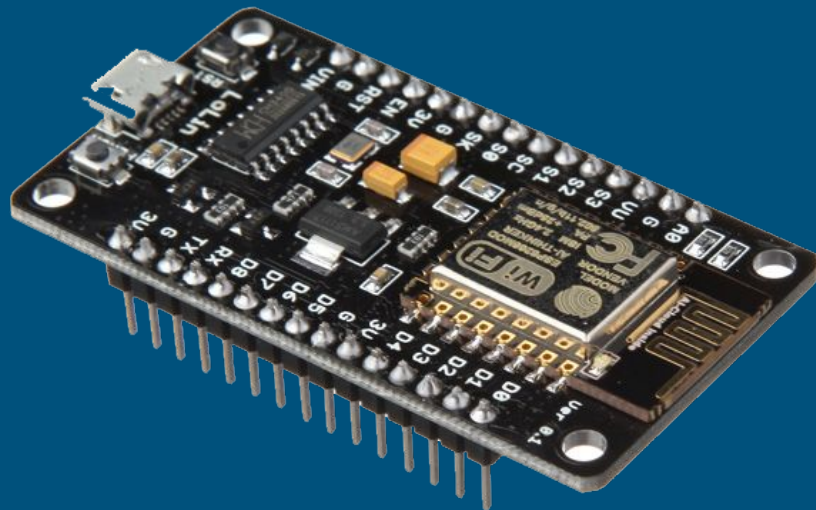
Features

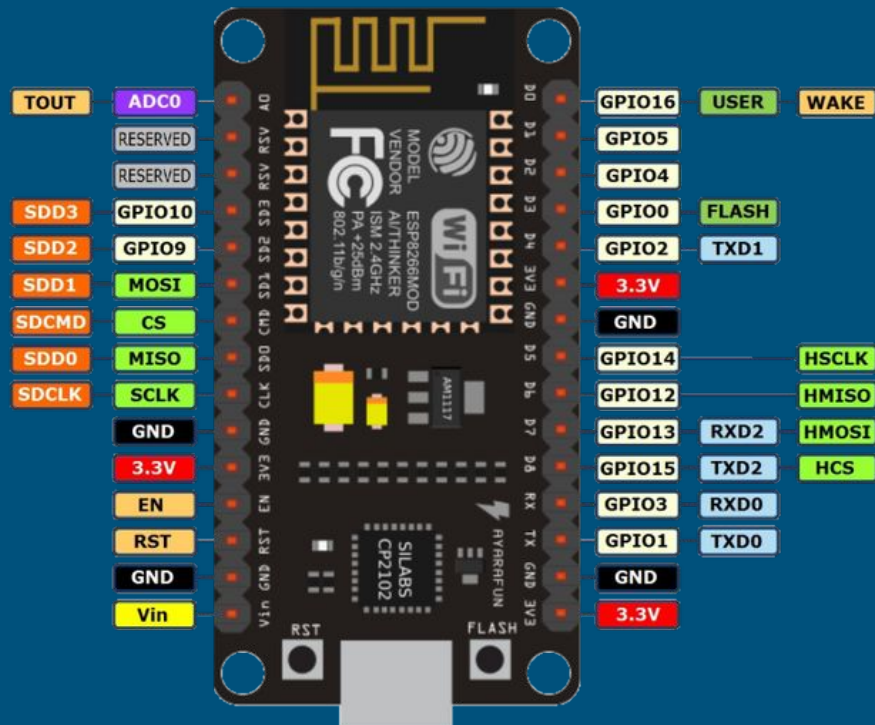
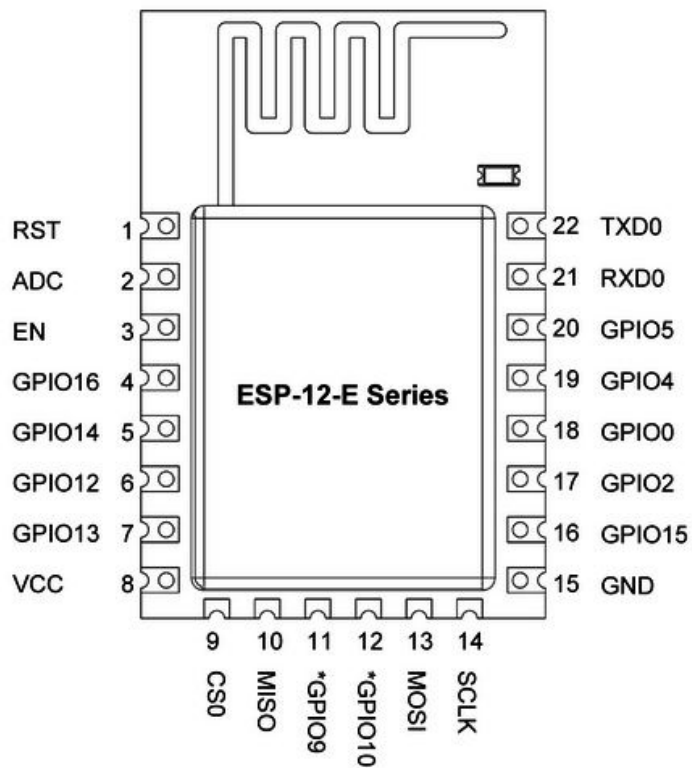
- Two digital pins
- Programmable
- Arduino IDE compatible
- 32 bit controller
- 80MHz clock

ESP - 12E



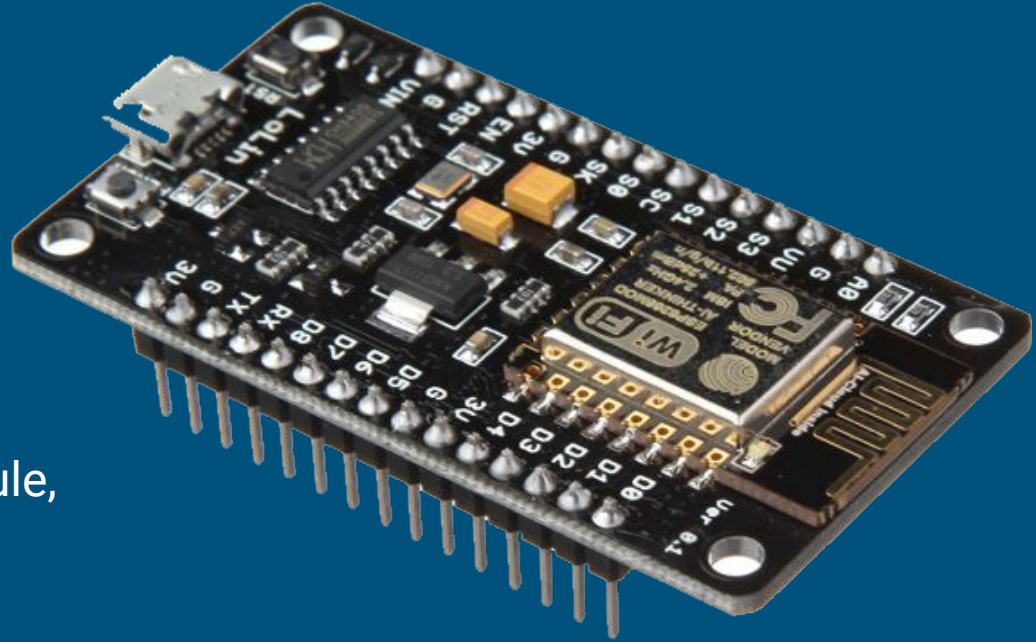
NodeMCU





NodeMCU Devkit 1.0

- Memory - 128kBytes
- Storage - 4MBytes
- Power - USB
- Open Source
- Lua script
- OS - XTOS
- Microcontroller - ESP-12E module, with Espressif ESP8266 32bits
- Operating Voltage - 3.3v
- Clock Speed - 80MHz
- Connectivity - IEEE 802.11 b/g/n Wi-Fi



NodeMCU Programming Using Arduino IDE

Add ESP8266 core Library

Step-1: Open the Arduino IDE and press ctrl + comma.

Step-2: Enter

http://arduino.esp8266.com/stable/package_esp8266com_index.json into
Additional Board Manager URLs field.

Step-3: Goto Tools → Boards → Boards Manager...

Step-4: Type 'esp' in search field and select 'esp8266 by ESP8266 community' and install it.

Step-5: Now check in Tools → Boards, there is a list of esp based boards, select NodeMCU 1.0 (ESP-12E module)

NodeMCU: Blink LED Program

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin D4 as an output.
  pinMode(D4, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(D4, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(D4, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Conclusion

- NodeMCU is a powerful, easy to use, and yet very low-cost solution for IoT application development.
- For IoT beginners, I feel the NodeMCU is one of the best options to go from prototyping all the way to production.