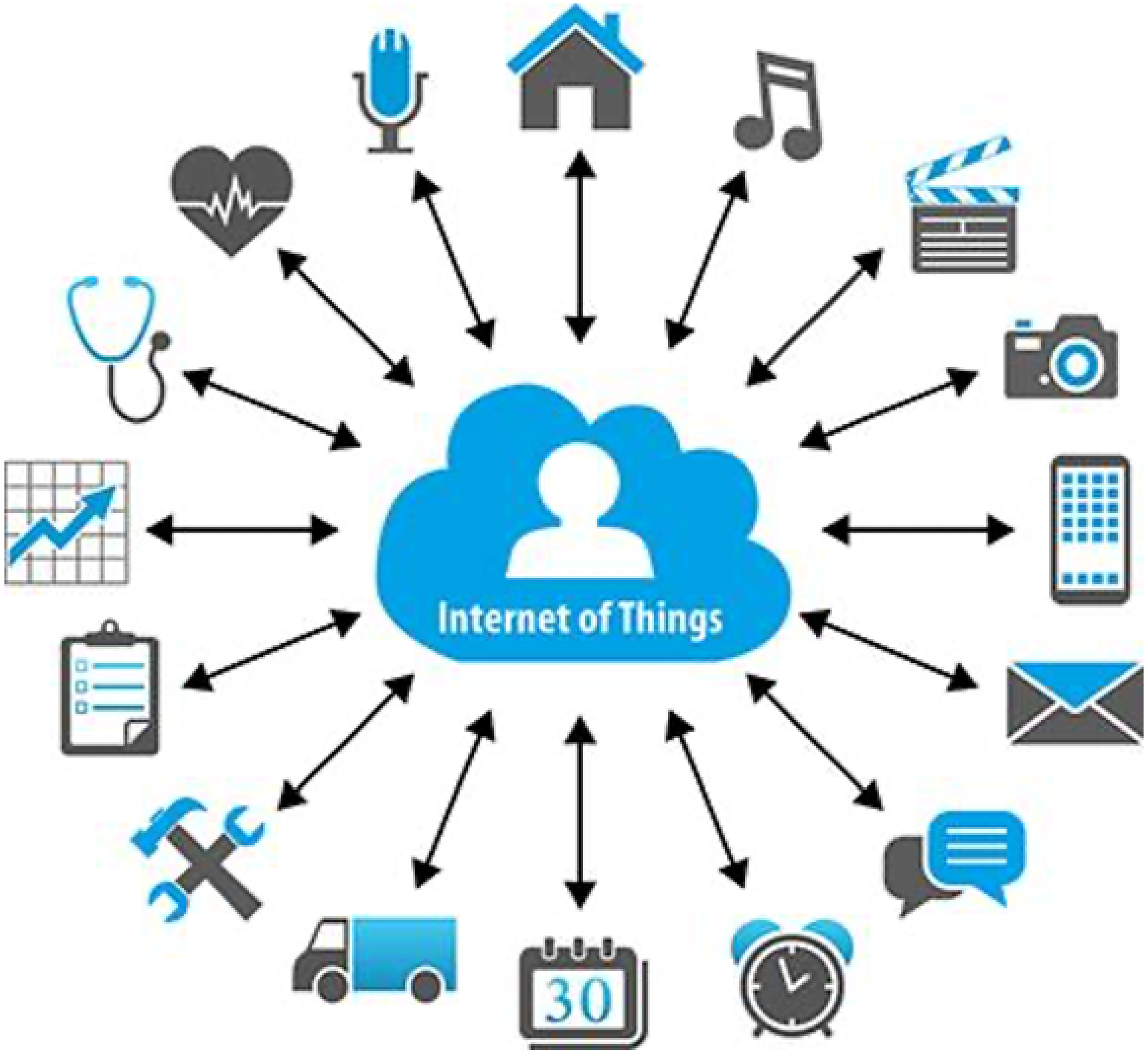
IoT Workshop

IoTPlex
19 September 2020

Alireza Abdeshah

What is The Internet of Things?

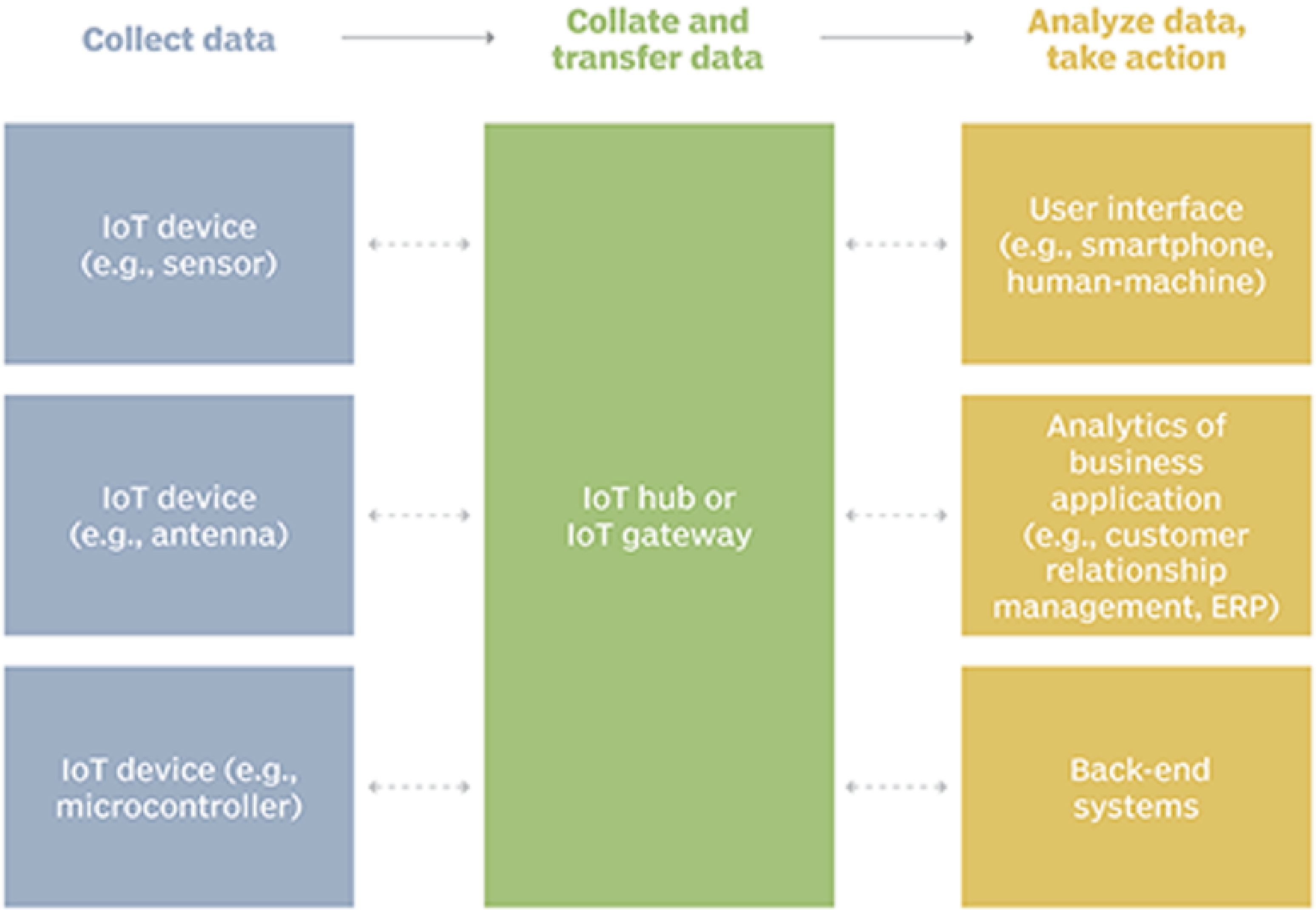


Application of the Internet of Things

According to Wikipedia Application of the Internet of Things:

- Environmental monitoring
- Infrastructure management
- Manufacturing
- Energy management
- Medical and healthcare
- Building and home automation
- Transportation
- Metropolitan scale deployments
- Consumer application

IoT Architecture



Collect data

Collate and transfer data

Analyze data, take action

IoT device (e.g., sensor)

IoT device (e.g., antenna)

IoT device (e.g., microcontroller)

IoT hub or IoT gateway

User interface (e.g., smartphone, human-machine)

Analytics of business application (e.g., customer relationship management, ERP)

Back-end systems

IoT Architecture

- IoT Device
- IoT Device Host (IoT Gateway)
- IoT Application

IoT Device

- ESP8266 & ESP32
- NRF5x
- Mediatek Linkit
- Samsung Artik
- Particle

IoT Gateway

- LoRa
- WiFi
- Bluetooth
- NB-IoT
- Cellular network

IoT Platform

- Mainflux
- Blynk
- Thingsboard
- Kaa Project
- Gobot
- ...

IoT Messaging Protocols

- Mqtt
- Ntas
- rest
- ...

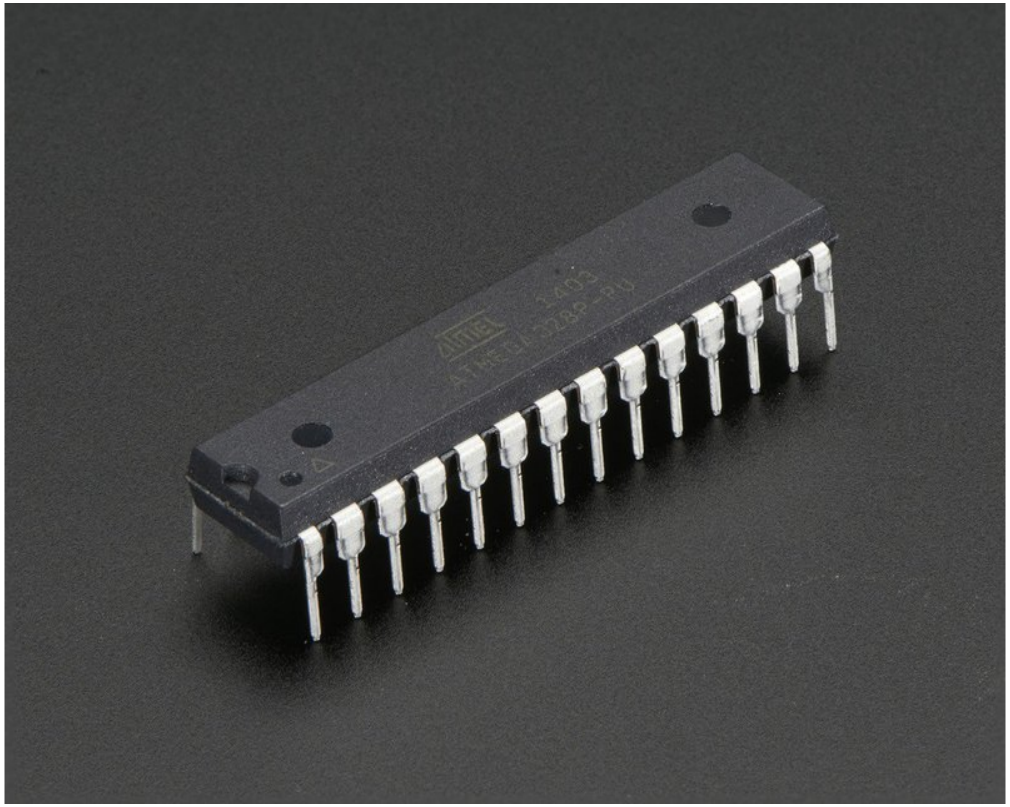
More Resource

https://github.com/HQarroum/awesome-iot

https://github.com/phodal/awesome-iot

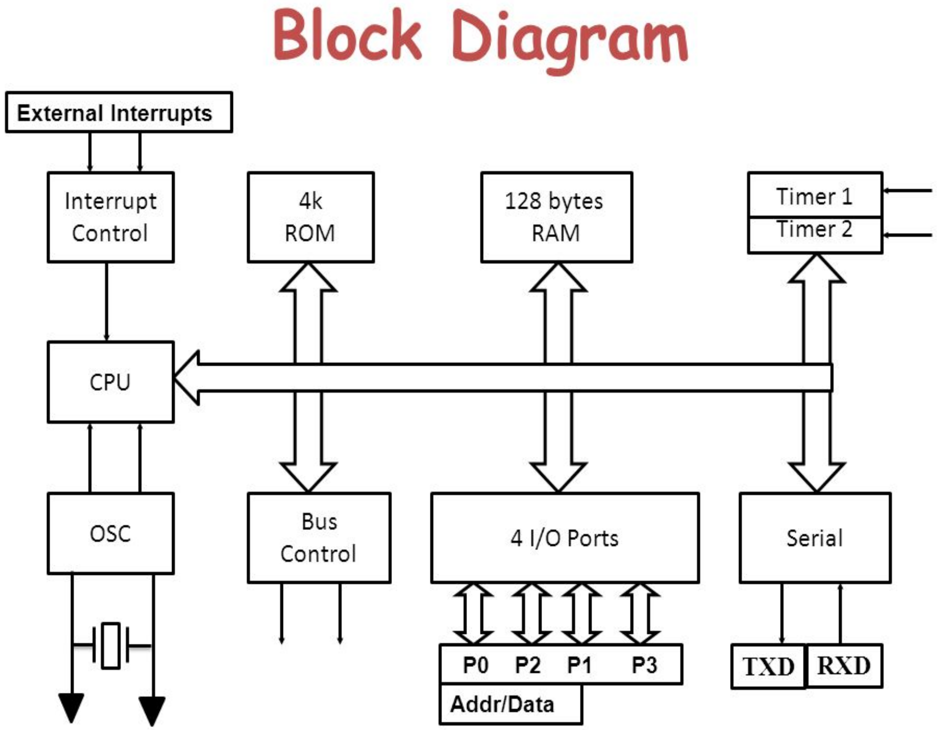
https://github.com/nebgnahz/awesome-iot-hacks

What is Microcontroller?

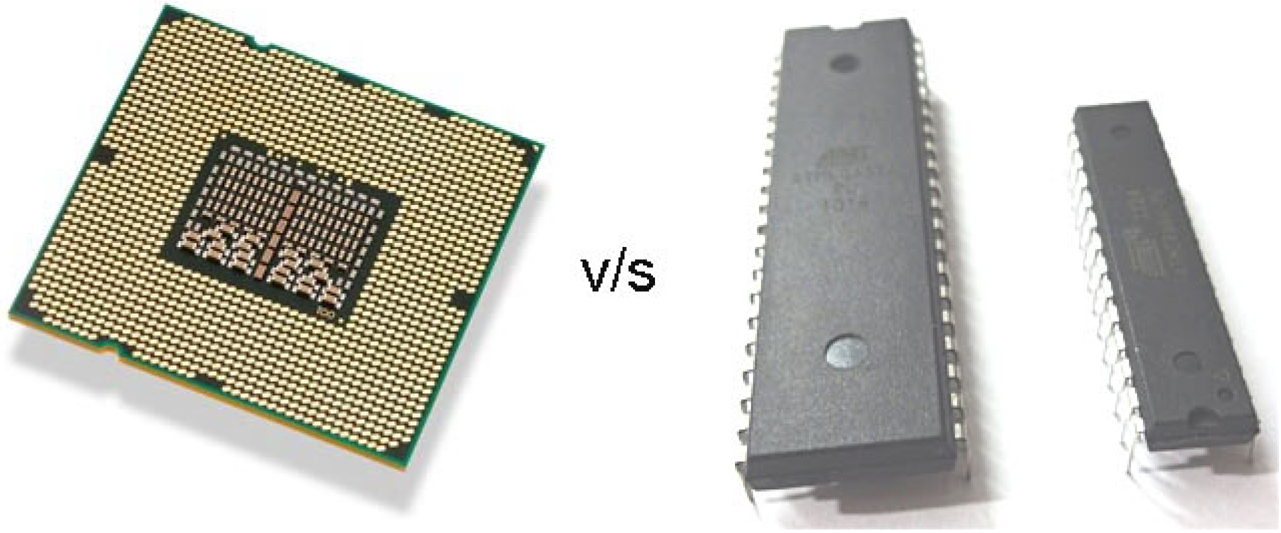


Microcontroller

A microcontroller (or MCU for microcontroller unit) is a small computer on a single integrated circuit



What is the difference between microprocessor and microcontroller?



Microcontroller

- are typically 8-bit, but may be 4-, 16-, or 32-bit
- run at speeds less than 200 MHz
- use very little power
- may provide enough current to operate an LED
- are useful to interface with sensors and motors
- are readily replaced, being inexpensive ($0.10 to $10)
- are really constrained for RAM and persistent storage (flash space)
- are really nice for electronics hobbyists

Microprocessors

- are often at least 16-bit, and typically 32-bit or 64-bit, though 8-bit still has a big market share
- many will be able to do floating point math in hardware
- run at speeds measured in hundreds of MHz
- are designed to be the brains of a system (and need a whole system to support them)
- need special hardware to interface with sensors, motors, LEDs, etc.
- are expensive (think $50 - $250 for 32 or 64-bit)
- are designed for external RAM and persistent storage (hard drives)
- are not as easily worked with by a hobbyist

AVR Microcontroller

AVR is a family of microcontrollers developed by Atmel beginning in 1996. These are modified Harvard architecture 8-bit RISC single-chip microcontrollers. AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

The '328 microcontroller has:

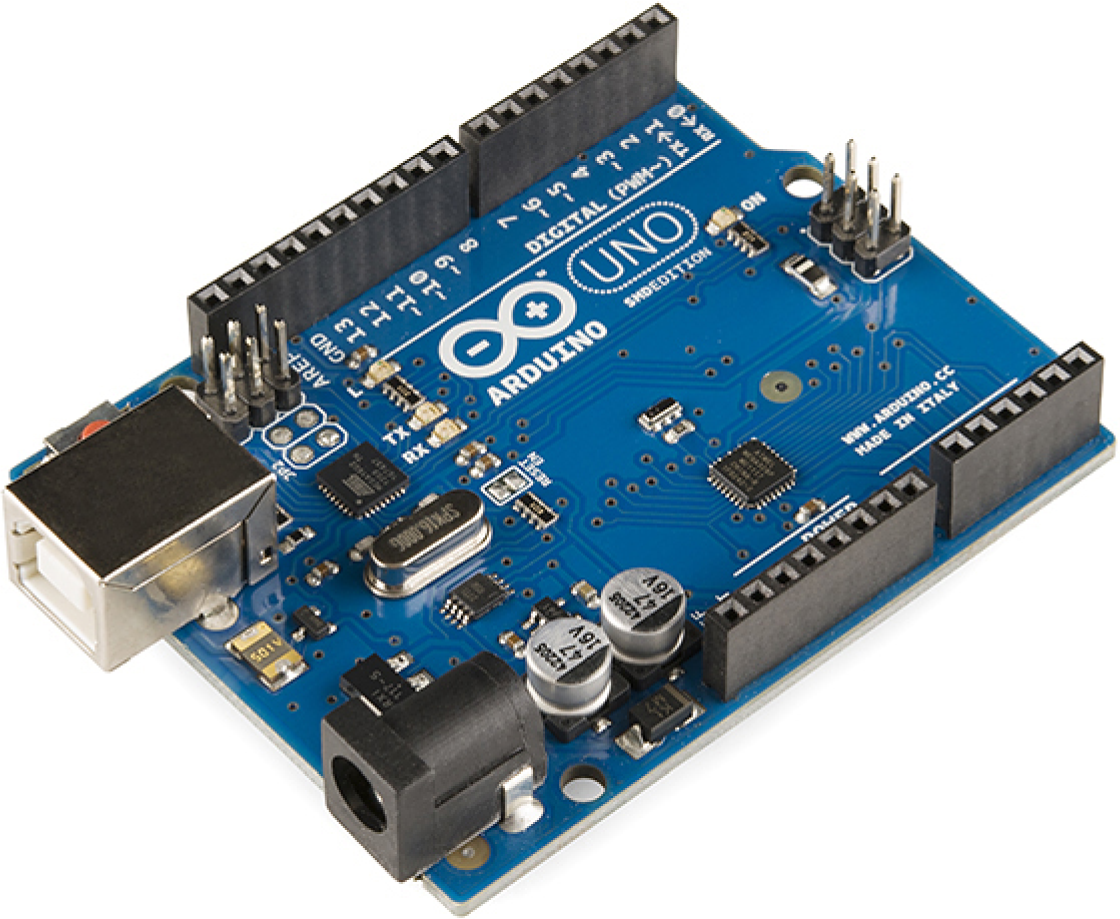
- 28 Pins
- Powered by 3 or 5 Volts
- Requires about 0.1 Watts of power
- Runs at 16 MHz
- 32 KB of flash storage
- 2 KB of RAM
- Costs about $5 per

What is an Arduino?

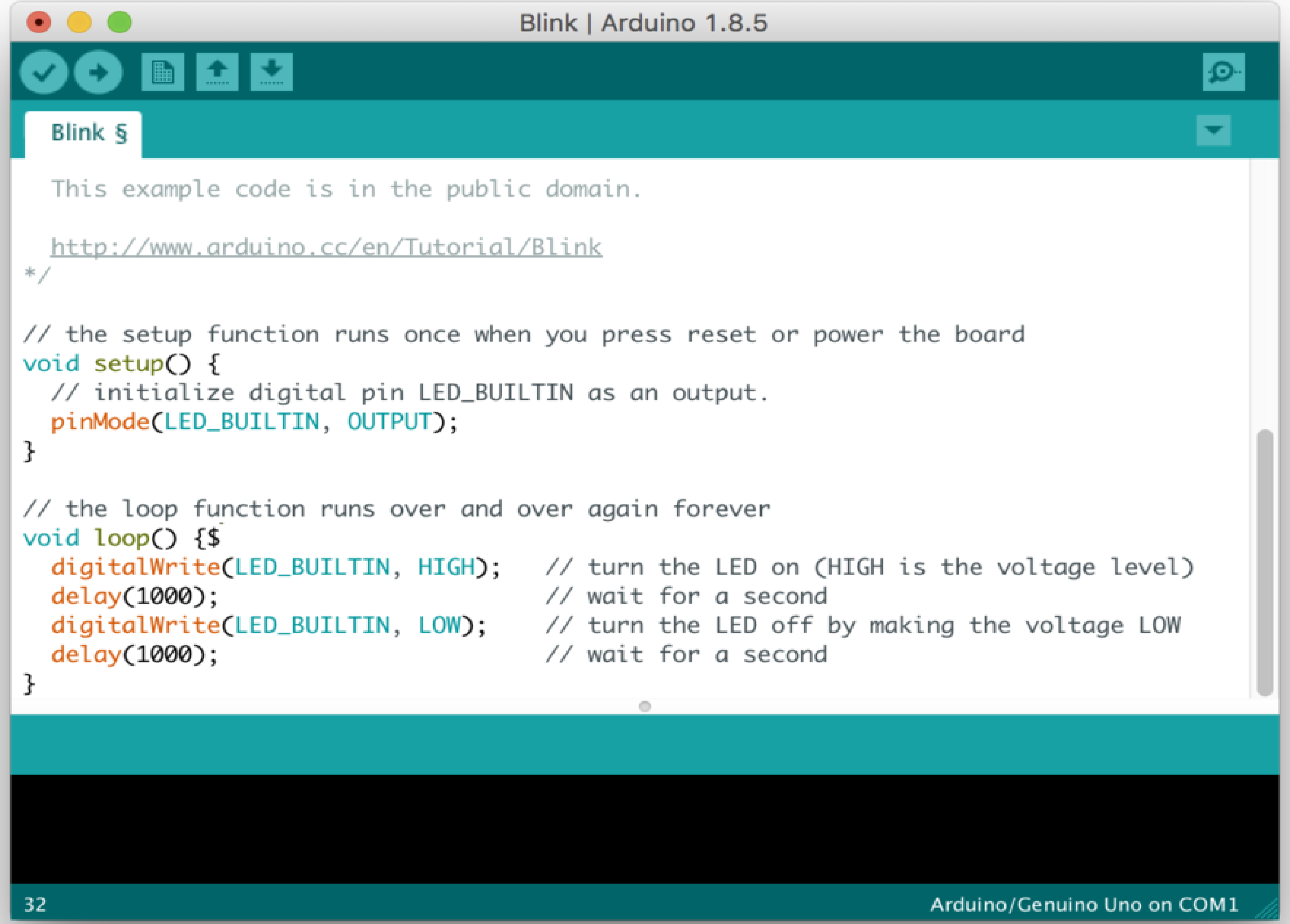
Arduino

Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Arduino Board

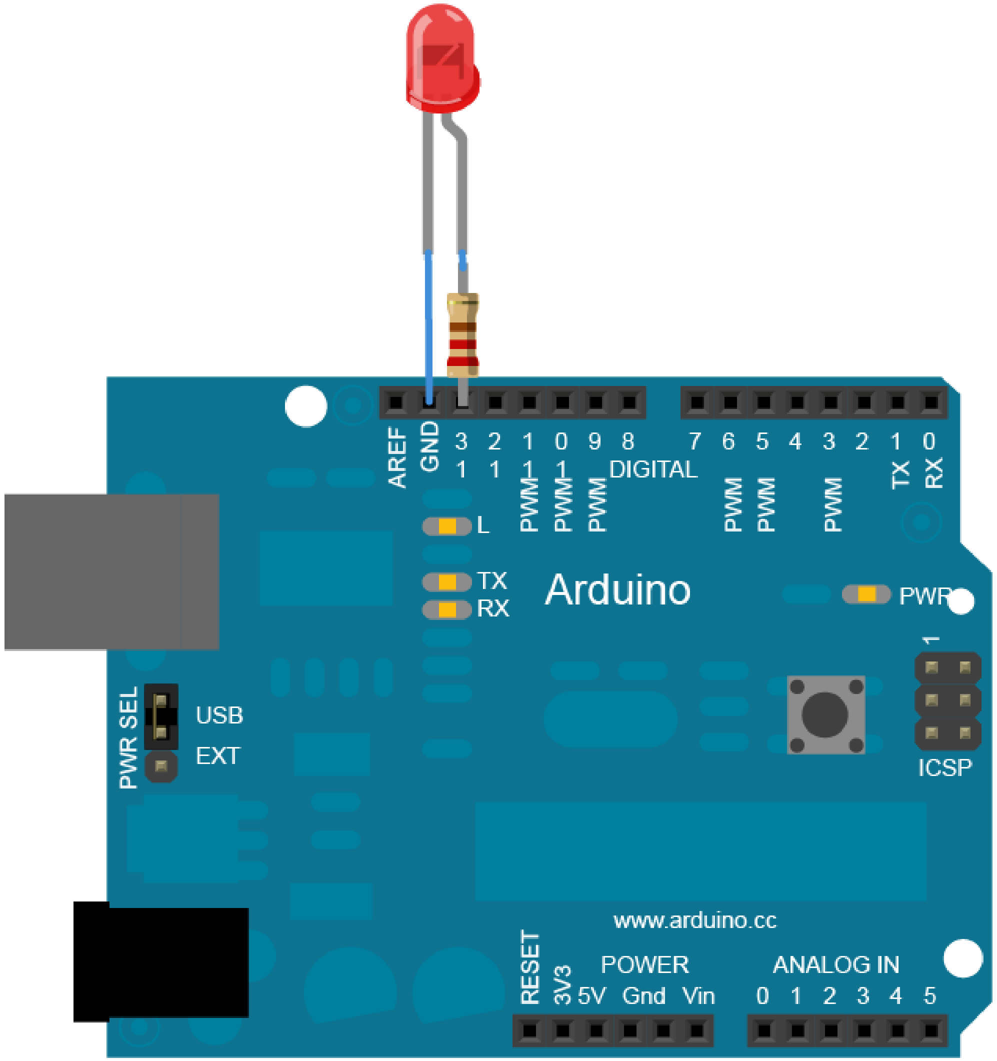


Ardiono IDE

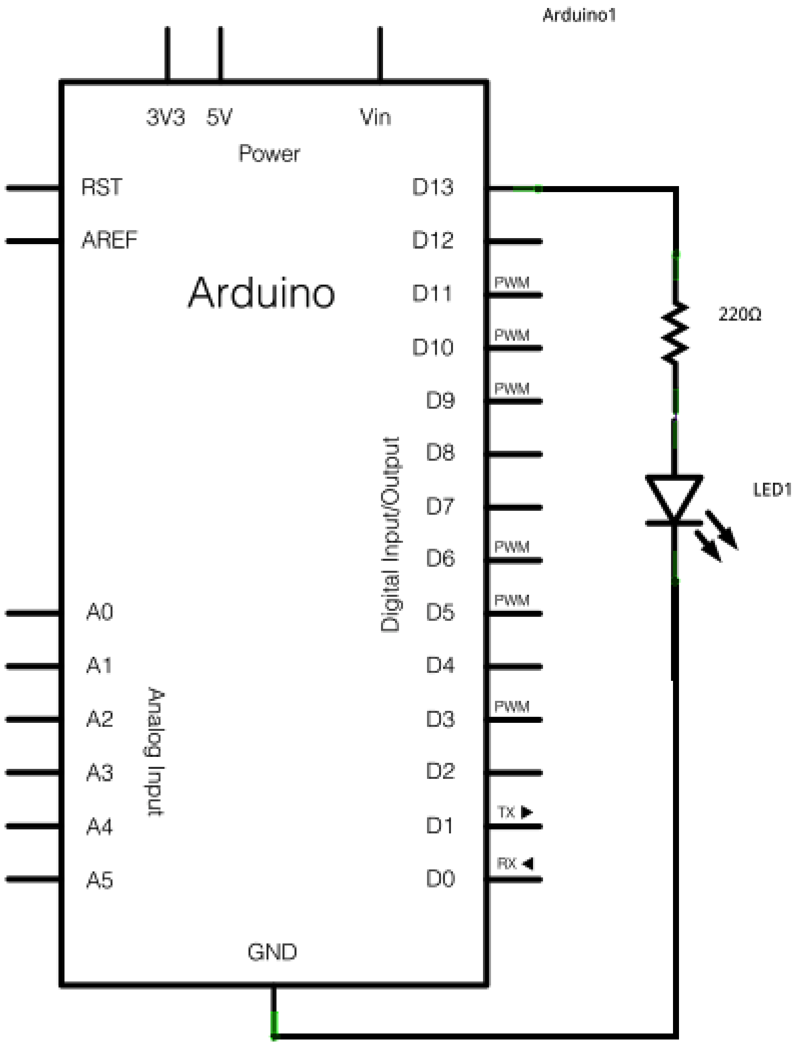


```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

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

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

Blink


AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
1
0
DIGITAL
PWM
PWM
PWM
PWM
PWM
PWM
TX
RX
L
TX
RX
Arduino
PWR
1
ICSP
PWR SEL
USB
EXT
www.arduino.cc
RESET
3V3
POWER
5V
Gnd
Vin
ANALOG IN
0
1
2
3
4
5

Schematic

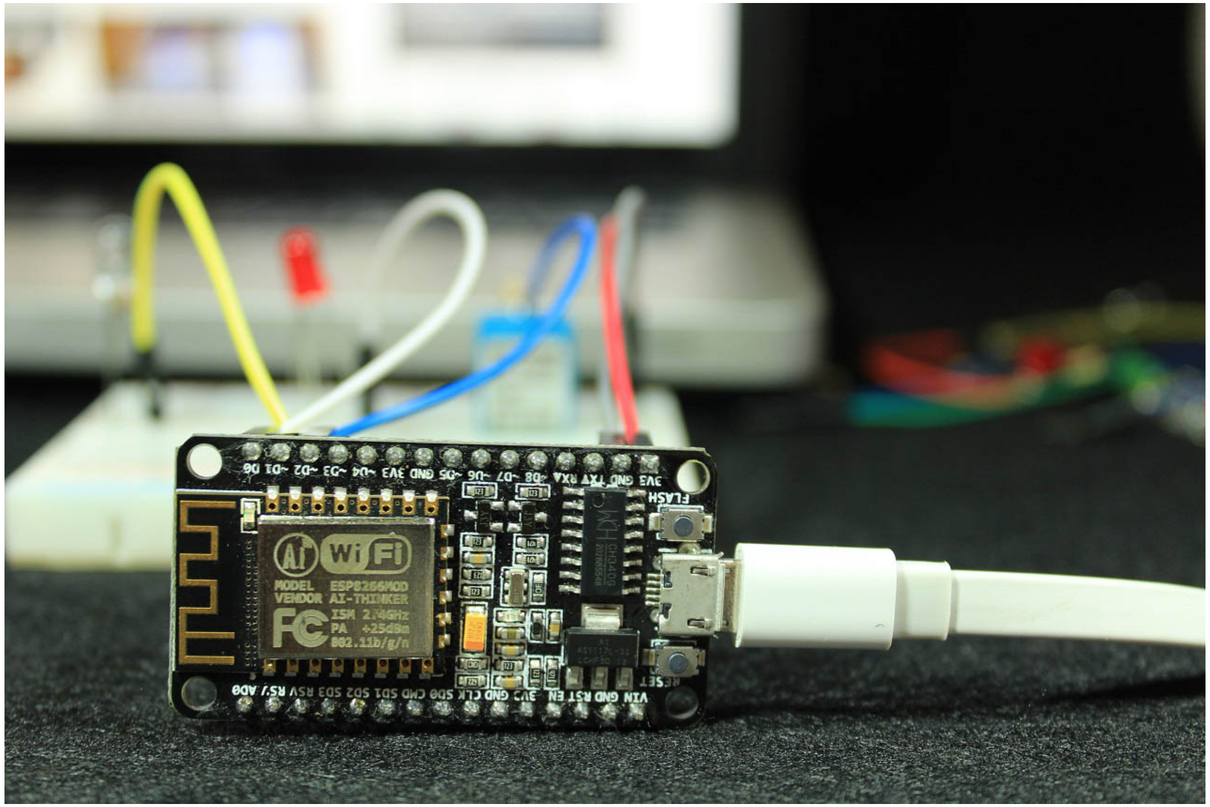

Arduino1
3V3
5V
Vin
Power
RST
AREF
Arduino
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
PWM
TX ▶
RX ◀
Digital Input/Output
A0
A1
A2
A3
A4
A5
Analog Input
GND
220Ω
LED1

Code

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

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

ESP8266

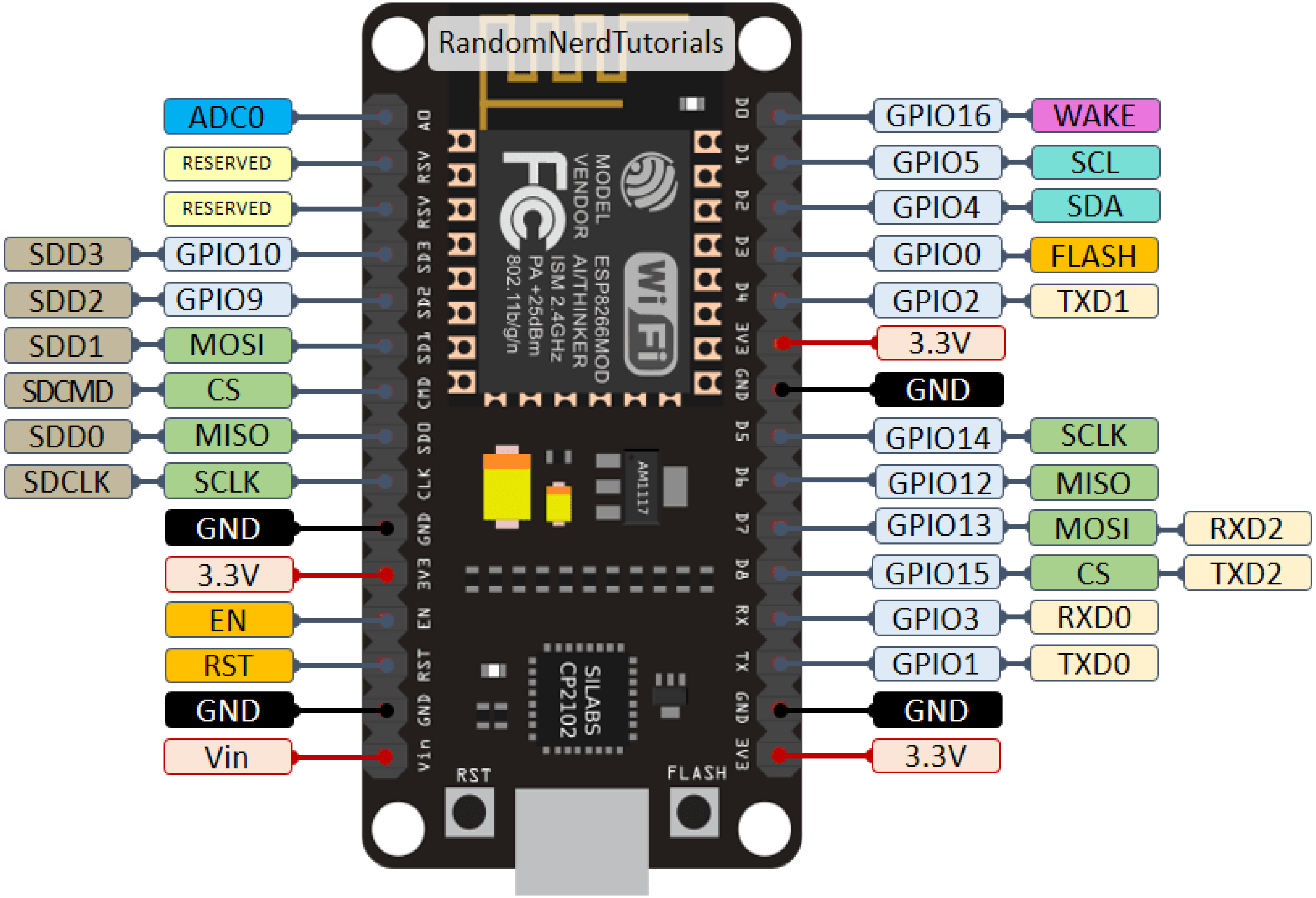
The ESP8266 is a $4 (up to $10) Wi-Fi module. It allows you to control inputs and outputs as you would do with an Arduino, but it comes with Wi-Fi.
So, it is great for home automation/internet of things applications.



ESP8266 Features

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond Standard 106Micro running at 80 MHz
- Memory: 32 KiB instruction RAM ,32 KiB instruction cache RAM ,80 KiB user-data RAM ,16 KiB ETS system-data RAM
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- IEEE 802.11 b/g/n Wi-Fi
- Integrated TR switch, balun, LNA, power amplifier and matching network
- WEP or WPA/WPA2 authentication, or open networks
- 16 GPIO pins
- SPI ,I²C ,I²S
- UART on dedicated pins, plus a transmit-only UART can be enabled on GPIO2
- 10-bit ADC (successive approximation ADC)

ESP-12E NodeMCU Kit Pinout



Programming the ESP8266 Using Arduino IDE

There are several ways to program the ESP8266. We often use Arduino IDE or MicroPython.

- Install the current upstream Arduino IDE at the 1.8.9 level or later. The current version is on the Arduino website.
- Start Arduino and open the Preferences window.
- Enter https://arduino.esp8266.com/stable/package_esp8266com_index.json into the Additional Board Manager URLs field. You can add multiple URLs, separating them with commas.
- Open Boards Manager from Tools > Board menu and install esp8266 platform (and don't forget to select your ESP8266 board from Tools > Board menu after installation).

Thank you

Alireza Abdeshah

eMicro.ir@Gmail.com (mailto:eMicro.ir@Gmail.com)

http://iotplex.ir/ (http://iotplex.ir/)

@eMicro (http://twitter.com/eMicro)