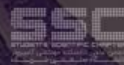


| 9th WSS General Meeting |

IoT WORKSHOP

By:

Mahdi Bahreiny



رند پیشرفت کارگاه

جلسه 2

اتصال ESP به شبکه و
کنترل آن به صورت بی
سیم و لوکال

جلسه 3

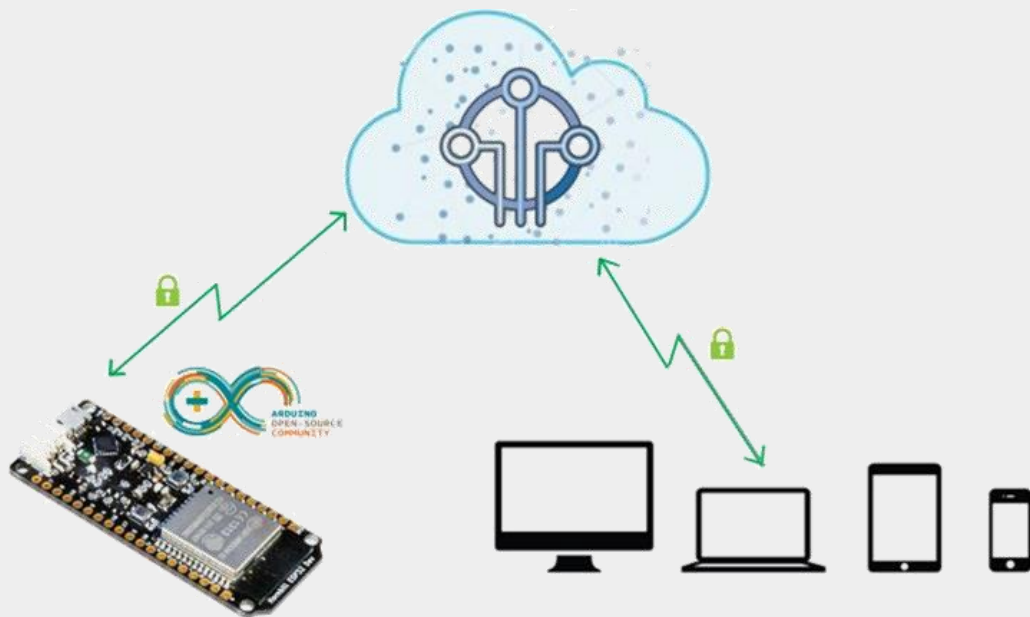
اتصال ESP به اینترنت
و کنترل آن به صورت
بی سیم و از راه دور

جلسه 1

راه اندازی ESP و
کنترل آن باسیم

جلسه 4

راه اندازی پنل کاربری
برای کنترل ESP از
طریق اینترنت



جلسه اول:

آشنایی با مفاهیم IoT و ماژول ESP32

اهمیت IoT

- کاهش هزینه ها
- بهبود کارایی
- بهبود قابلیت حمل

The Internet of Things (IoT)



کاربردهای IoT

- خانه هوشمند
- صنعت پزشکی
- صنعت حمل و نقل
- مصرف هوشمند انرژی



Everyday things



M2M wireless
sensor network



Building management



Agriculture automation



Embedded mobile



energy consumption



security and surveillance



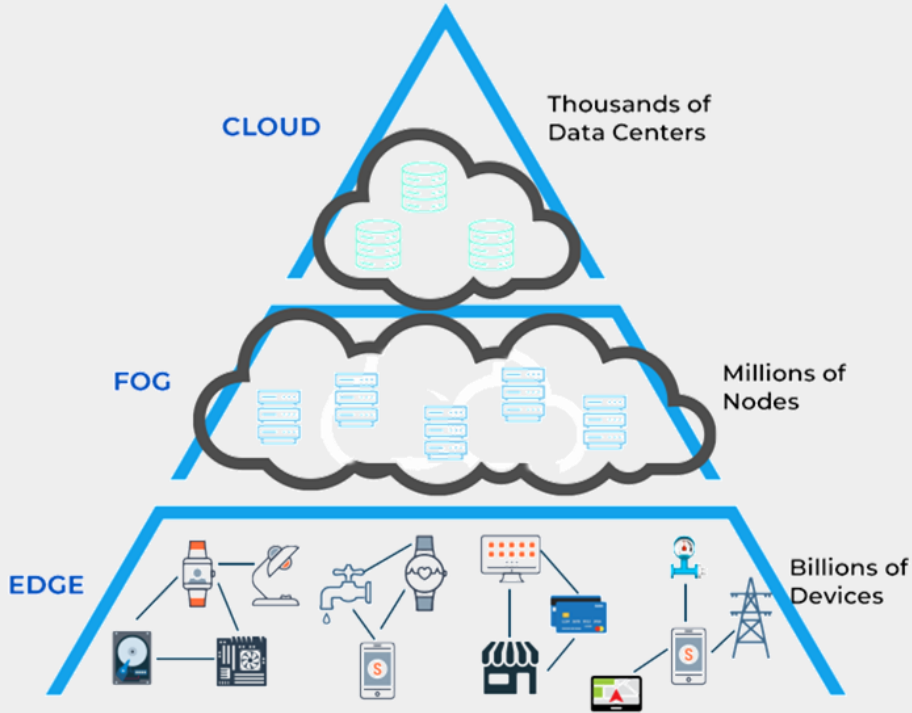
pet, person, vehicle's and asset
monitoring & controlling



smart homes
and cities

ساختار سامانه IoT

CLOUD COMPUTING VS. FOG COMPUTING VS. EDGE COMPUTING

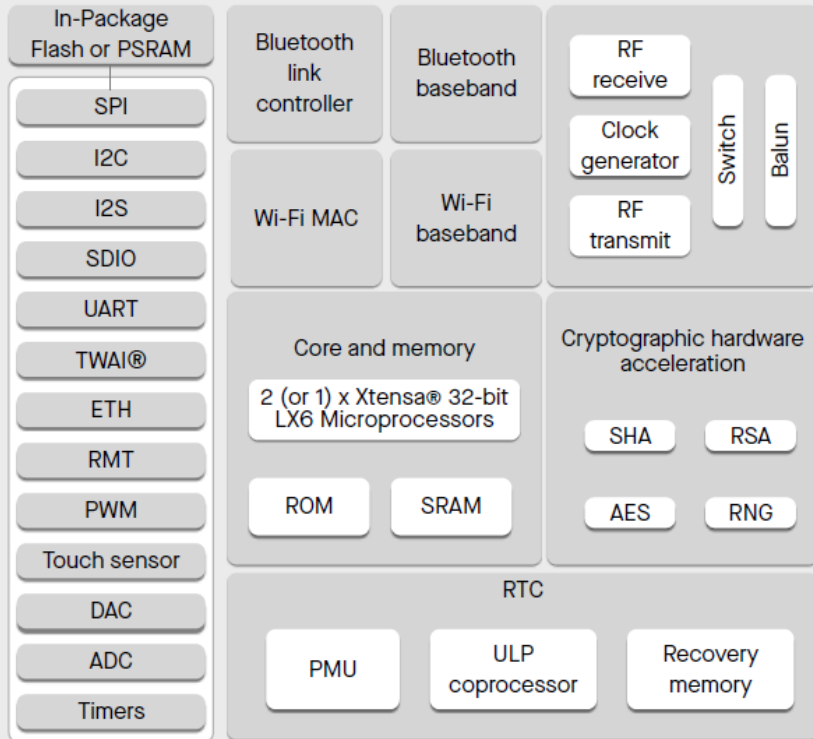


- لایه لبه (Edge)

- لایه مه (Fog)

- لایه ابر (Cloud)

آشنایی با ESP32



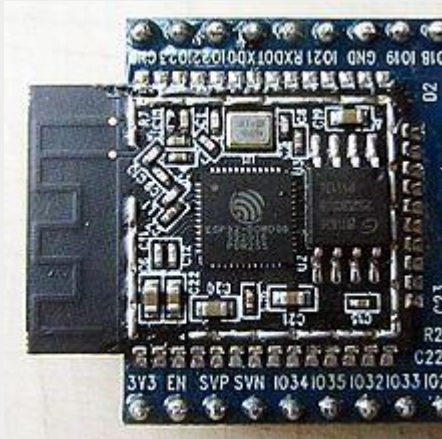
- یک SoC با قابلیت وایفای، بلوتوث دارای امکانات کامل یک میکروکنترلر

آشنایی با WROOM-32



- کریستال جهت عملکرد esp32

- آنتن برای بخش رادیویی



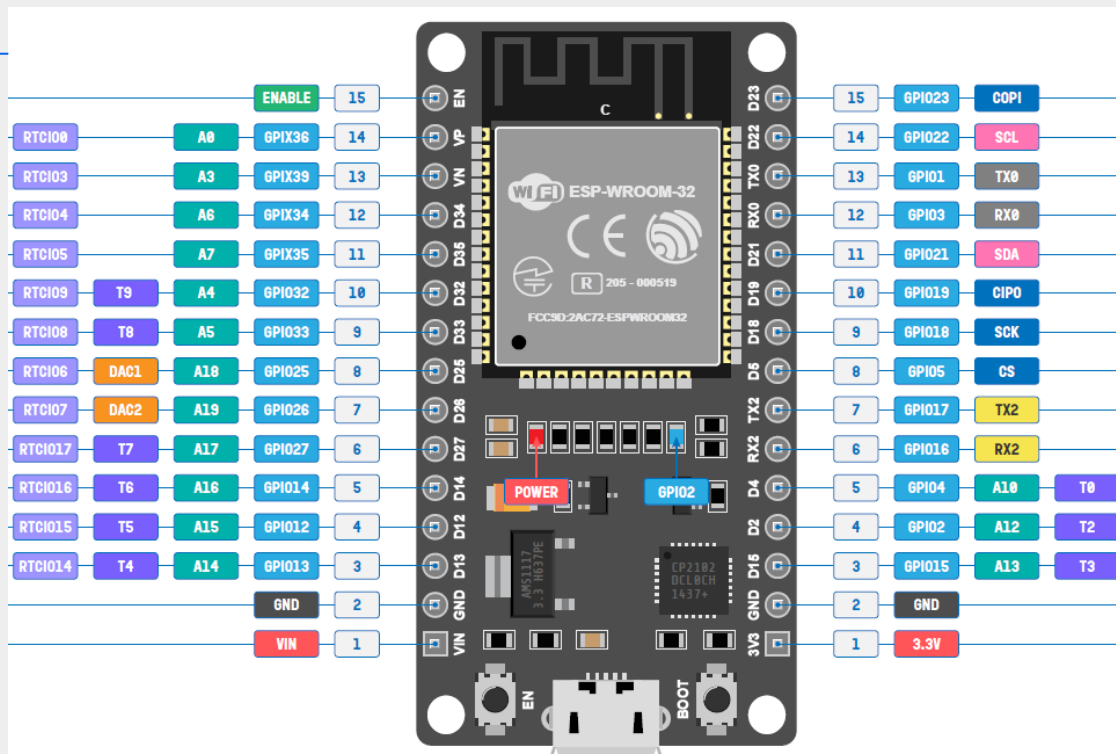
آشنایی با NodeMCU 32S



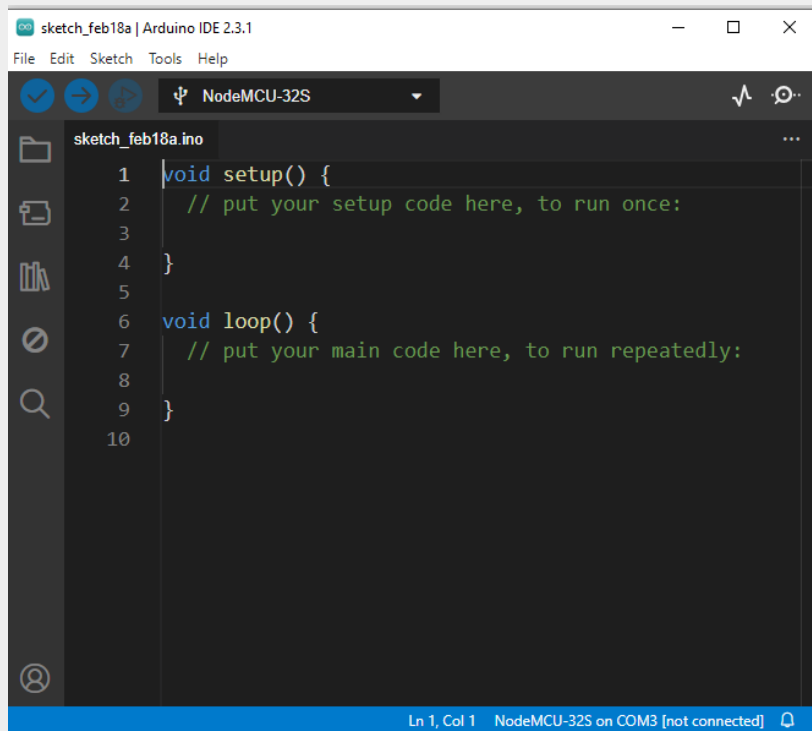
- رگولاتور ولتاژ
- مبدل USB به UART
- LED روی برد
- دکمه های بوت و ریست

پین های برد

- 18 پین ADC
- 3 رابط SPI
- 3 رابط UART
- 2 رابط I2C
- 16 خروجی PWM
- 2 عدد DAC
- 2 رابط I2S
- 10 پین لمس خازنی



نحوه برنامه نویسی برای ESP



```
sketch_feb18a.ino
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
10
```

Ln 1, Col 1 NodeMCU-32S on COM3 [not connected]

ESP-IDF

شیوه پیشنهاد شده توسط شرکت سازنده

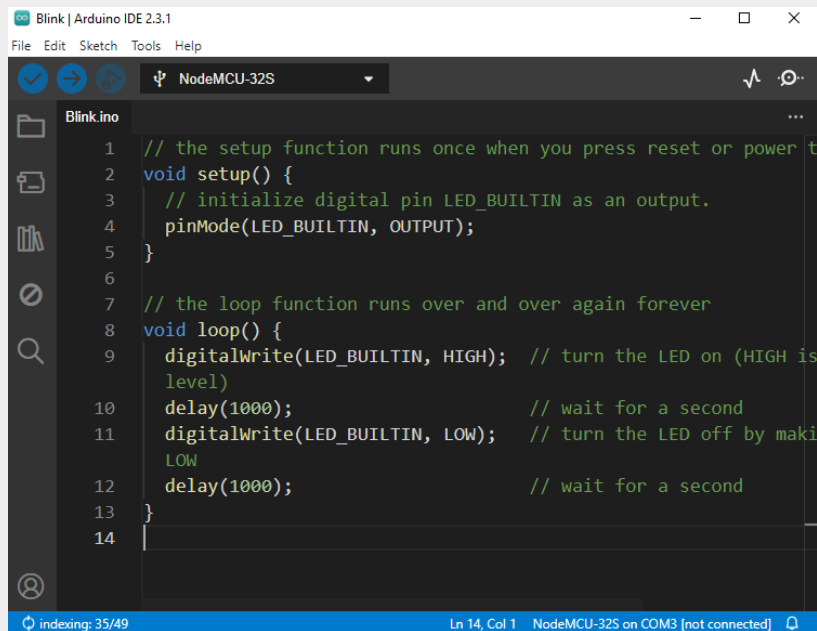
انعطاف بیشتر

Arduino IDE

سهولت بیشتر

سرعت بیشتر توسعه

نوشتن یک برنامه ساده



```
Blink | Arduino IDE 2.3.1
File Edit Sketch Tools Help
NodeMCU-32S
Blink.ino
1 // the setup function runs once when you press reset or power t
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is
   level)
10  delay(1000); // wait for a second
11  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by maki
   LOW
12  delay(1000); // wait for a second
13 }
14
indexing: 35/49 Ln 14, Col 1 NodeMCU-32S on COM3 [not connected]
```

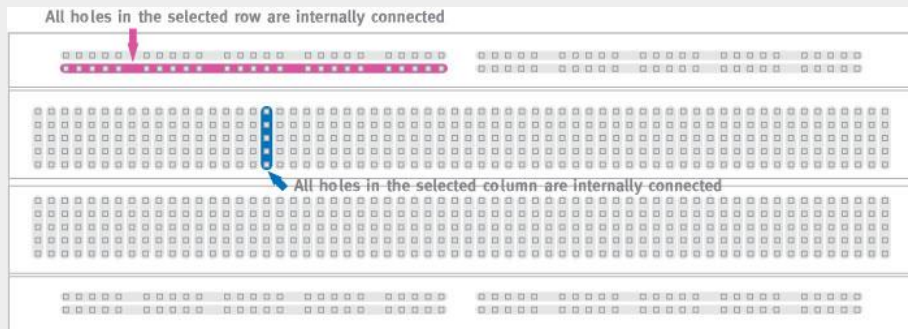
PinMode

تعریف نوع پین

digitalWrite

نوشتن یک مقدار دیجیتال روی پین

برد بورد



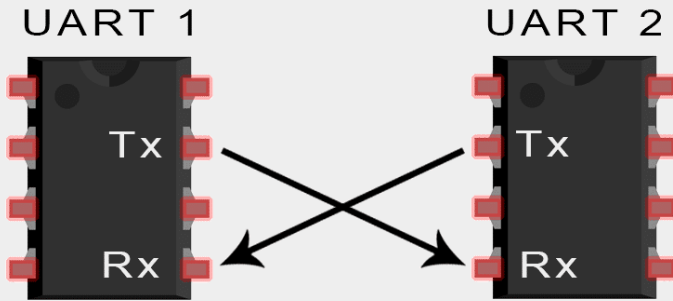
ردیف های کناری

اتصال افقی تمام خانه ها

ردیف های میانی

به صورت ستونی

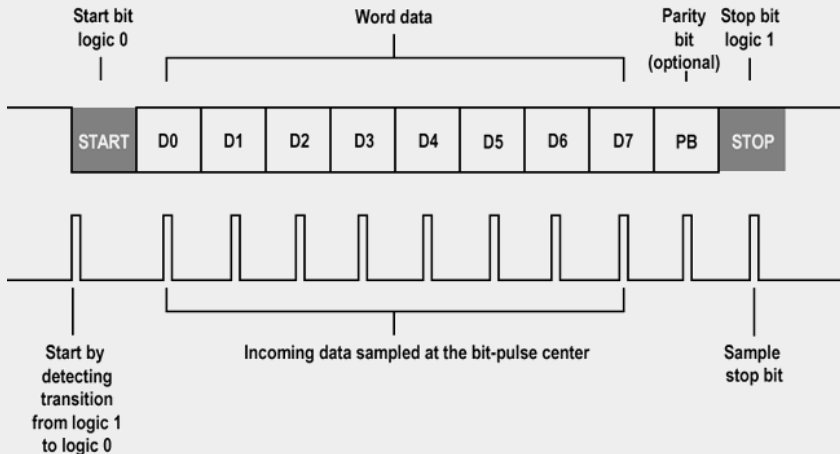
ارتباط سریال UART



سیم کشی

اتصال Tx و Rx به صورت ضربدری

اتصال Gnd



فریم داده و پروتکل

بیت ها و نرخ تبادل

ارتباط سریال

```
1 uint8_t counter = 0;
2 uint32_t lastSavedTime = 0;
3 void setup() {
4     Serial.begin(115200);
5 }
6 void loop() {
7     if(millis() - lastSavedTime > 1000){
8         counter ++;
9         Serial.print("One second past and counter is ");
10        Serial.println(counter);
11        lastSavedTime = millis();
12    }
13 }
```

Serial.begin(baudrate)

راه اندازی و تنظیم نرخ تبادل داده

Serial.println(data)

نوشتن داده در سریال

خواندن ورودی دیجیتال

Polling

در حلقه اصلی مدام ورودی چک می‌شود

`digitalRead(pinNumber)`

- دارای سربار در برنامه اصلی

Interrupt

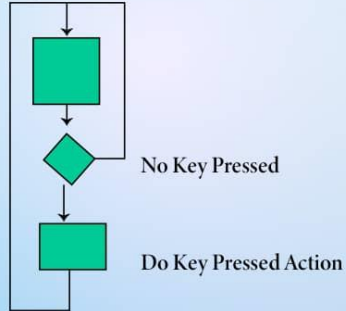
به محض تغییر ورودی تابع ISR اجرا می‌شود

`attachInterrupt(pinNumber, isrFunction, RISING)`

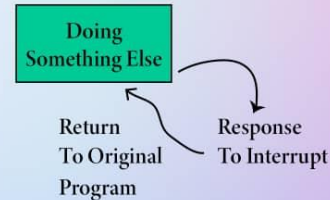
- خارج شدن از روند برنامه اصلی

Polling vs Interrupt

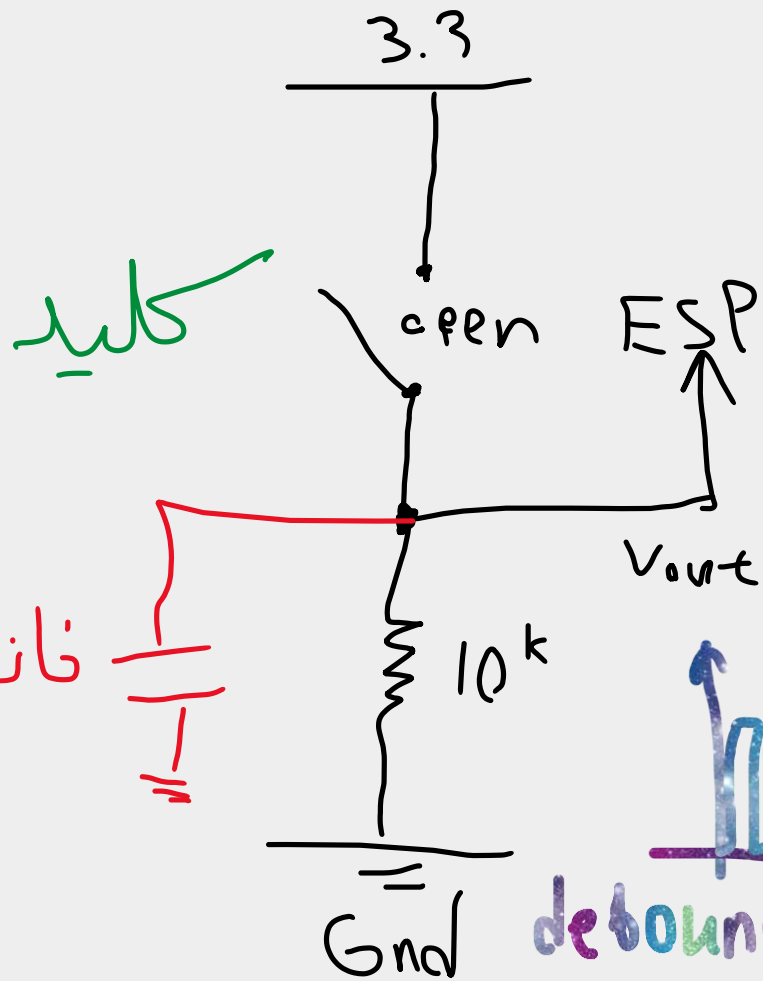
Polling



Interrupt



کلید فشاری



کلید	V_{out}
open	0 (0V)
close	1 (3.3V)



بدون debouncing



با debouncing سخت افزاری

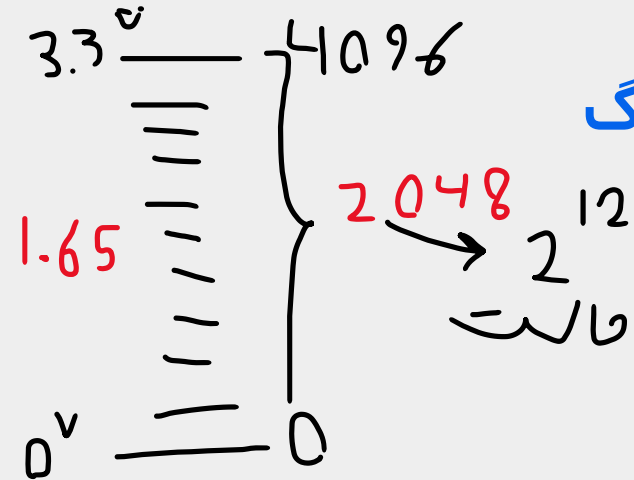
خواندن ورودی دیجیتال (ادامه)

Interrupt

```
1 #define BUTTON_PIN 13
2 uint32_t lastTimeKeyPressed = 0;
3 volatile bool flag = false;
4 void IRAM_ATTR isr() {
5     if (millis() - lastTimeKeyPressed > 500)
6         flag = true;
7 }
8 void setup() {
9     Serial.begin(115200);
10    pinMode(BUTTON_PIN, INPUT);
11    attachInterrupt(BUTTON_PIN, isr, RISING);
12 }
13 void loop() {
14     if (flag) {
15         flag = false;
16         lastTimeKeyPressed = millis();
17         Serial.println("key pressed");
18     }
19 }
```

Polling

```
polling.ino
1 #define BUTTON_PIN 13
2 uint32_t lastTimeKeyPressed = 0;
3 void setup() {
4     Serial.begin(115200);
5     pinMode(BUTTON_PIN, INPUT);
6 }
7 void loop() {
8     if (digitalRead(BUTTON_PIN)) {
9         if (millis() - lastTimeKeyPressed > 500)
10             Serial.println("key pressed");
11         lastTimeKeyPressed = millis();
12     }
13 }
```

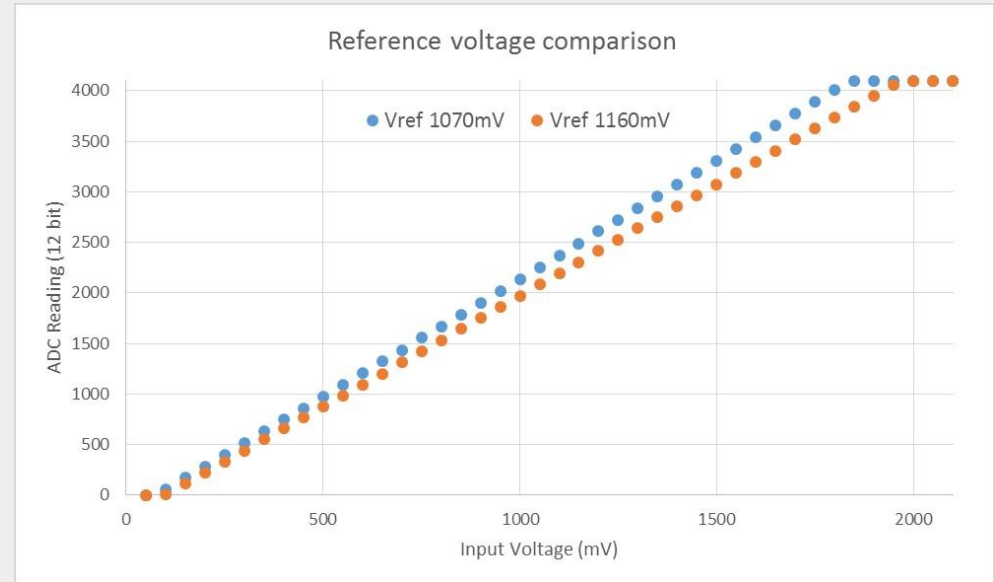
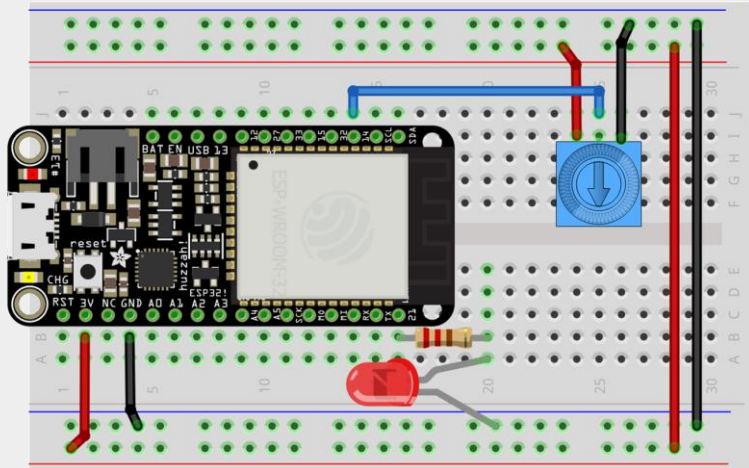


خواندن ورودی آنالوگ

$$\frac{V_i}{4096} \times 3.3 = V_{\text{oltage}}$$

`analogRead()`

واحد 12 بیتی ADC



کادی 6 k
نور زیاد 100
نور کم 100 k

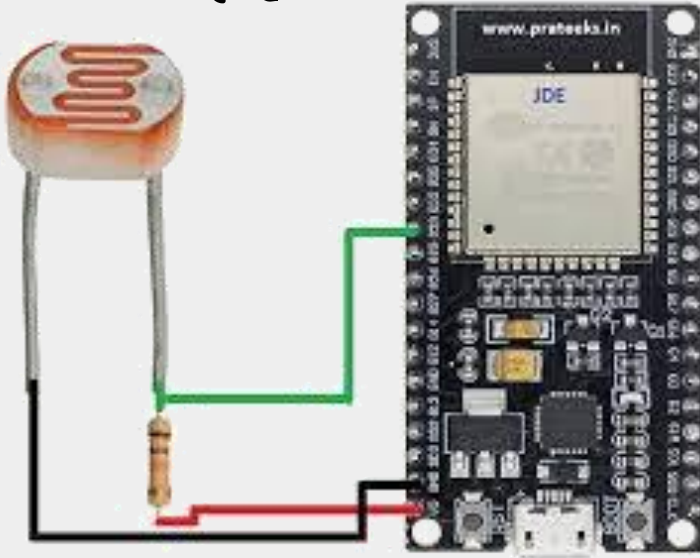
سنسور LDR

LDR

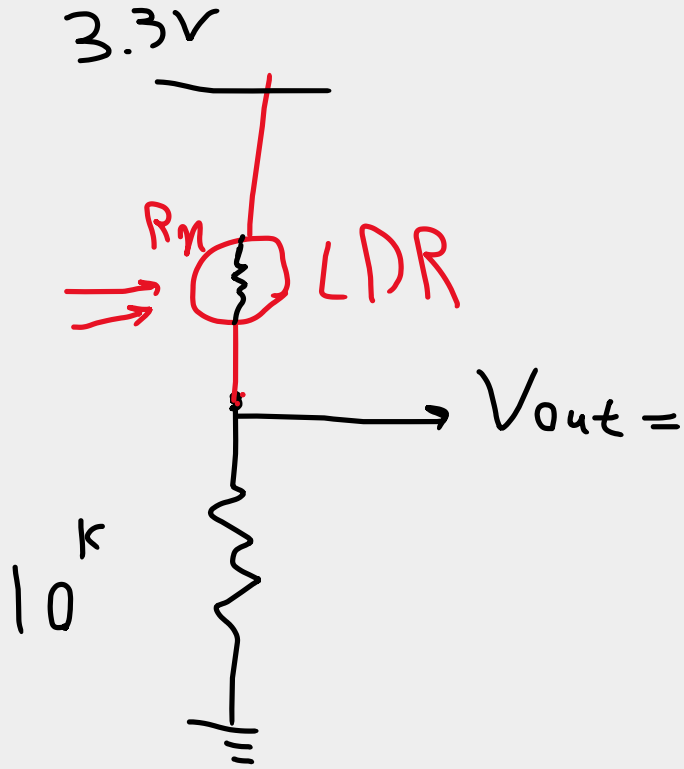
مقاومت متغیر با میزان روشنایی

نحوه استفاده

سری کردن با یک مقاومت و خواندن ولتاژ



LDR sensor

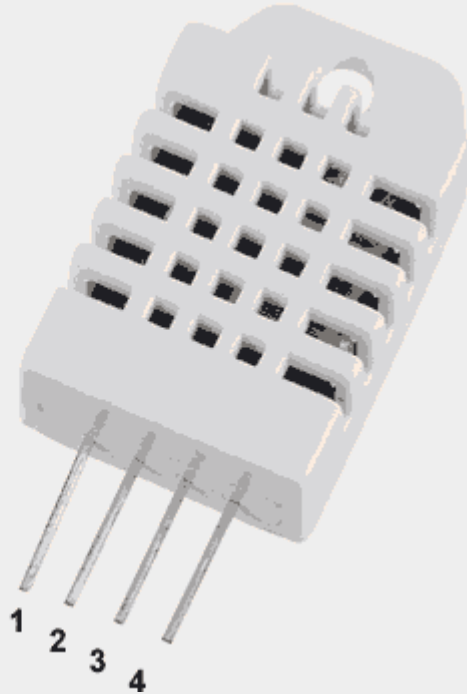


$$V_{out} = \frac{10^k}{10^k + R_L} \times 3.3$$

```
light = analogRead(LDR_PIN);  
String lightStat;  
if (light < 1000)  
  lightStat = "dark";  
else if (light < 3000)  
  lightStat = "normal";  
else  
  lightStat = "bright";
```

سنسور DHT22

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND



دما

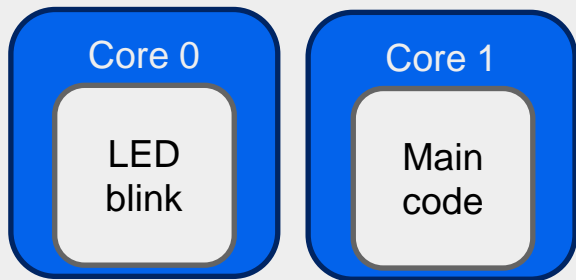
با استفاده از دستور `getTemperature()`

رطوبت

با استفاده از دستور `getHumidity()`

```
#include <DHT22.h>
DHT22 dht22(DHT_PIN);
float temperature, humidity;
void readDhtData(){
    temperature = dht22.getTemperature();
    humidity = dht22.getHumidity();
}
```

مالتی تسکینگ



ایجاد تسک و اختصاص به هسته

استفاده از امکانات FreeRTOS

```
xTaskCreatePinnedToCore(function, "name", 1000, NULL, 1, NULL, coreNumber);
```

استفاده از منبع مشترک

استفاده از semaphore و mutex

```
xSemaphoreTake(xMutex, portMAX_DELAY)
```

```
xSemaphoreGive(xMutex)
```



ممنون از
توجه شما