



دانشکده فنی و حرفه‌ای شهید شمسی پور

گزارش پروژه

عنوان پروژه: IOT

دانشجو: مصطفی دست غیبی شیرازی

استاد راهنما: دکتر محمد خانی

مقطع: کارشناسی نرم افزار

سنة الفجر

# فهرست

فهرست	۳
چکیده	۴
مقدمه	۵
مفهوم اینترنت اشیا	۶
قطعات مورد استفاده در پروژه	۷
Wemos esp8266 D1 Mini	۷
ماژول رله ۵ ولت دو کاناله	۱۱
ماژول کارت خوان RC522	۱۳
شرح و اهداف پروژه	۱۵
زبان های مورد استفاده در این پروژه	۱۶
نحوه کار پروژه	۱۷
لینک ها	۱۸
تصاویر	۱۹
کد های esp8266	۲۲

## چکیده

در این پروژه قصد داریم وسایل مختلفی از جمله درب های ورودی ، روشنایی ، درب پارکینگ و دستگاه های مختلف را کنترل و هوشمند سازی کنیم . همچنین کاربر هنگامی که بخواهد از وب استفاده کند یک احراز هویتی صورت می گیرد.

در این پروژه کاربر می تواند از دو طریق کنترل داشته باشد :

۱-بر بستر وب

۲-RFID

کلمات کلیدی :

- lot
- internet of things
- هوشمند سازی
- اینترنت اشیاء

## مقدمه

مصطفی دست غیبی شیرازی با شماره دانشجویی ۰۰۲۲۱۰۳۳۷۰۵۰۰۵ هستم که در ترم ۴ مقطع کارشناسی رشته نرم افزار در دانشگاه شهید شمس پور مشغول به تحصیل هستم و واحد پروژه را با موضوع iot (اینترنت اشیا) اخذ کردم .

با تشکر از استاد راهنما دکتر محمدخانی که مرا در این پروژه راهنمایی کردند .

در این پروژه من سعی کردم تمام علاقه مندی ها و دانش و مهارت های خودم که شامل (نرم افزار ، الکترونیک ، شبکه، سخت افزار) بود را مورد استفاده و بهره برداری قرار دهم و این پروژه را به نحو احسن و توانم به پایان رسانم .

## مفهوم اینترنت اشیا

اینترنت اشیا (Internet of Things) که گاهی به نام اینترنت اجسام (Internet of Objects) یا به اختصار I.O.T نیز نامیده می‌شود، به ارتباط اینترنتی بین اشیا و تجهیزاتی می‌گویند که در محیط پیرامون ما قرار دارند.

این لوازم یا اشیا متصل به شبکه اینترنت می‌توانند با استفاده از نرم‌افزارهای موجود در تلفن‌های هوشمند، تبلت‌ها، رایانه‌ها، گجت یا ابزارک‌ها، ساعت‌های هوشمند، تلویزیون‌ها و هر شیء دیگری به صورت کنترل از راه دور مدیریت شوند.

اینترنت اشیا به معنای ارتباط گیرنده‌ها و دستگاه‌ها با شبکه اینترنت است که از طریق این ارتباط و تعامل بین لوازم متصل به شبکه و کاربران دارای دسترسی مجاز به این شبکه، امکان دیدن و کنترل لوازم متصل به شبکه برای کاربران آن فراهم می‌شود. این مفهوم می‌تواند به سادگی ارتباط یک گوشی هوشمند با تلویزیون یا به پیچیدگی نظارت بر زیرساخت‌های شهری و ترافیکی باشد. این شبکه بسیاری از دستگاه‌های اطراف ما را از جمله ماشین لباس شویی و یخچال گرفته تا پوشاک را دربرمی‌گیرد.

به عبارتی دیگر اینترنت اشیا یعنی مجموعه‌ای از اشیا و تجهیزات که به وسیله اتصال به یک شبکه با یکدیگر ارتباط برقرار می‌کنند. این فناوری می‌تواند به مصرف کنندگان خدمات گوناگونی ارائه کند، برای مثال می‌توان به کنترل برخی از وسیله‌ها از راه دور، اطلاع از شرایط محیطی خانه و غیره اشاره کرد.

## قطعات مورد استفاده در پروژه

### Wemos esp8266 D1 Mini

برد Wemos D1 Mini یک برد توسعه یافته برای اجرای برنامه‌های کاربردی برای اینترنت اشیا IoT است. این برد بر اساس میکروکنترلر ESP8266 طراحی شده و دارای ابعاد کوچکی است که به راحتی در پروژه‌های کوچک و متوسط قابل استفاده است. Wemos D1 Mini از مجموعه‌ای از پایه‌ها و ماژول‌ها تشکیل شده است که شامل ورودی/خروجی دیجیتال و آنالوگ، ارتباط WiFi، پورت USB، و رابط‌های شیلد قابل ارتقا و اتصال است. این برد با استفاده از زبان برنامه‌نویسی Arduino IDE و یا MicroPython قابل برنامه‌ریزی است. از ویژگی‌های و مومس دی ۱ مینی می‌توان به قیمت ارزان، کارایی بالا، طراحی کوچک، مصرف کم انرژی، و کتابخانه‌های آماده اشاره کرد که موجب شده است برای کاربرانی که می‌خواهند به سادگی پروژه‌های IoT را بر روی اینترنت بیاورند یکی از انتخاب‌های مناسب باشد. به عبارت دیگر اگر بردی با امکانات Nodemcu ولی کوچکتر نیاز دارید، این برد انتخاب مناسبی خواهد بود. این برد فقط دارای یک پایه آنالوگ با حداکثر ولتاژ ورودی ۳/۲ ولت است. از پورت micro usb بهره‌برده و با میکروپایتون و آردوینو قابل راه‌اندازی است. ابعاد این برد در حدود ۳۴/۲ × ۲۵/۶ میلی‌متر است. با توجه به دیتاشیت این برد سرعت کلاک ۸۰/۱۶۰ MHz است.

## مشخصات فنی برد wemos D1 Mini

ولتاژ عملیاتی: ۳/۳ ولت DC

ورودی 5 DC: ولت DC از طریق پورت USB

ورودی آنالوگ: ۱ پورت ورودی آنالوگ با دقت ۱۰ بیت (ADC) با ورودی حداکثر ۳/۲ ولت

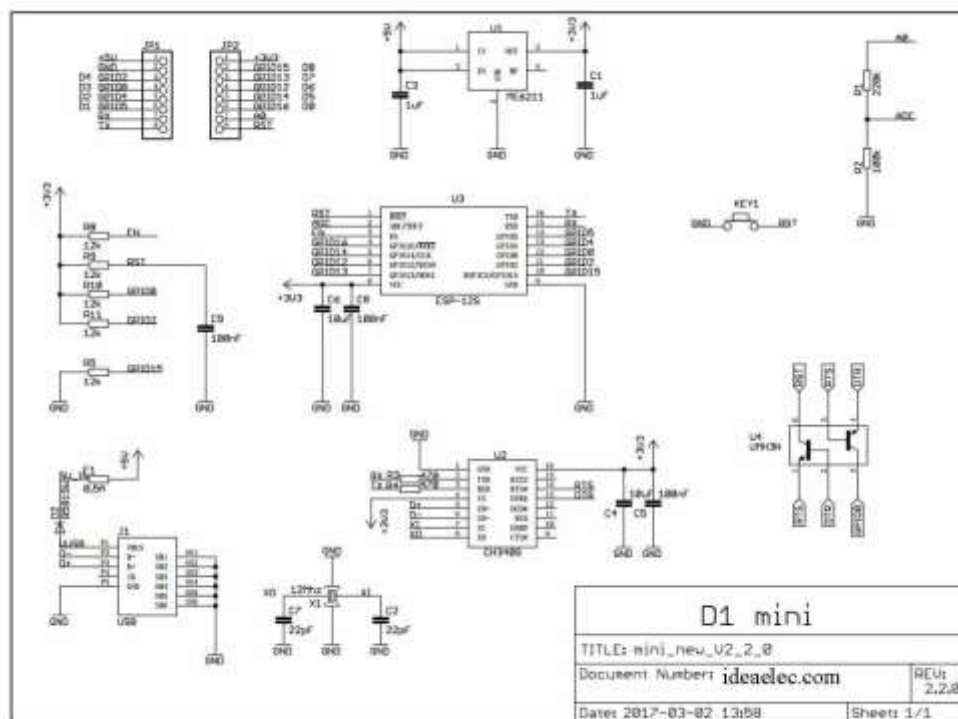
ورودی دیجیتال: ۱۱ پورت ورودی/خروجی دیجیتال با پشتیبانی از سطوح ولتاژ ۳/۳ ولت

رابط شبکه Wi-Fi 802.11 b/g/n :

پورت USB: Micro USB برای اتصال به کامپیوتر و تغذیه از طریق آن

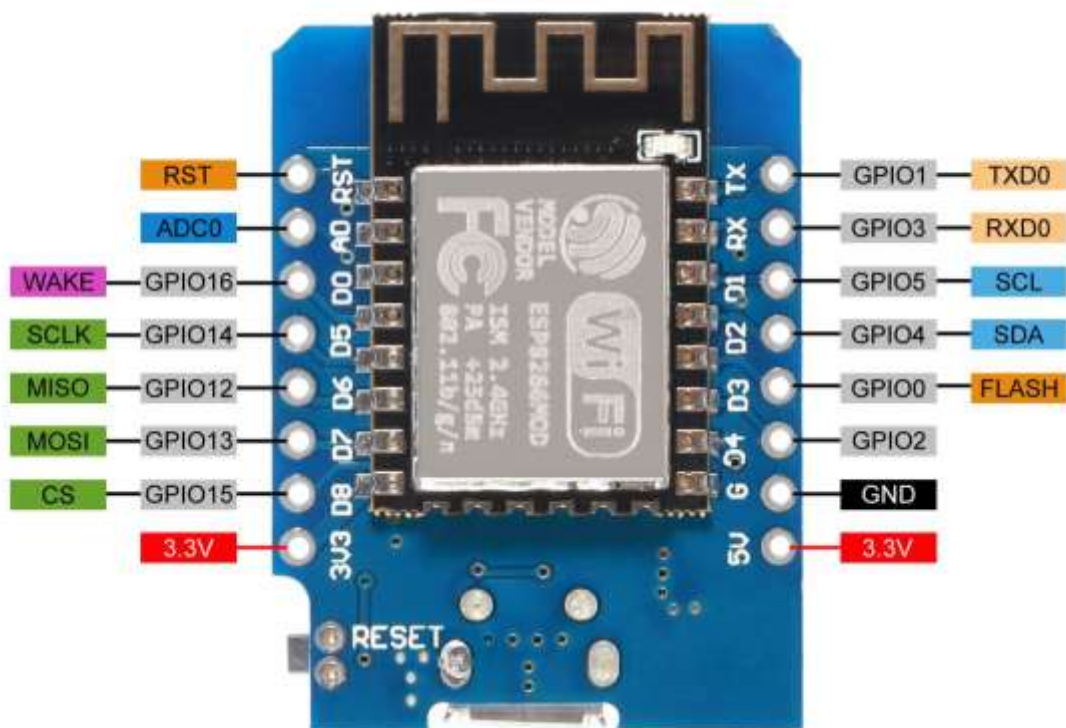
حافظه: ۴ MB فلش داخلی

ابعاد: ۳۴/۲ mm x 25.6mm x 7.2mm





## بررسی پایه های ماژول Wemos D1 mini



VCC: منبع تغذیه ۳/۳+ ولتی

GND: زمین

D0: پایه تغذیه (خروجی با دقت ۳/۳ ولت)

D1: I2C برای رابط SCL و همچنین پایه GPIO5

D2: I2C برای رابط SDA و همچنین پایه GPIO4

پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

استفاده می Boot ، همچنین برای راه اندازی با کد فلش در حالت GPIO0 پایه D3: شود.

به صورت LED و همچنین خروجی ۳/۳ ولتی که برای کنترل GPIO2 پایه D4: پیشفرض در نظر گرفته شده است.

SPI برای رابط CLK و همچنین پایه GPIO14 پایه D5:

SPI برای رابط MISO و همچنین پایه GPIO12 پایه D6:

SPI برای رابط MOSI و همچنین پایه GPIO13 پایه D7:

، همچنین می توان آن را برای SPI برای رابط SS و همچنین پایه GPIO15 پایه D8: کنترل خروجی های دیجیتال ورودی/خروجی استفاده کرد.

پایه ورودی سریال RX:

پایه خروجی سریال TX:

پایه ورودی آنالوگ A0:

## ماژول رله ۵ ولت دو کاناله

ماژول رله دو کانال، یک دستگاه الکترونیکی است که دارای دو کانال خروجی رله است. هر کانال از این ماژول می‌تواند برای کنترل یک مدار الکتریکی یا الکترونیکی استفاده شود. رله‌ها از نوع سوئیچ هستند که به وسیله‌ی جریان الکتریکی، می‌توانند یک مدار را روشن یا خاموش کنند. ماژول رله دو کانال می‌تواند به عنوان یک واسطه بین مدارهای الکتریکی و کنترل کننده‌ی آنها مورد استفاده قرار گیرد. استفاده از ماژول رله دو کانال در سیستم‌های خانگی، صنعتی و تجاری بسیار رایج است و به کاربران امکان کنترل دستگاه‌های مختلف را با استفاده از میکروکنترلر، ماژول‌های سنسور و دیگر وسایل الکترونیکی می‌دهد.

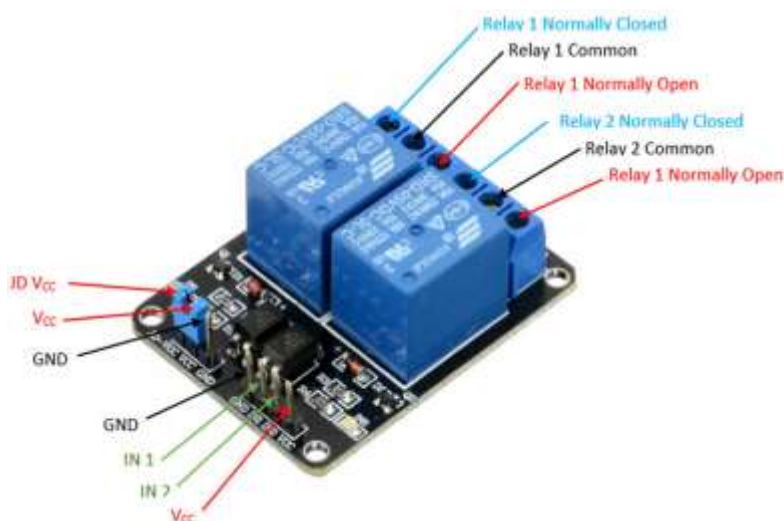
## ویژگی‌های ماژول رله ۵ ولت دو کاناله

- دارای دو کانال جهت کنترل دو وسیله برقی
- قابلیت تحمل تا ۲۵۰ ولت AC در خروجی رله
- مجهز به اپتوکوپلر جهت ایزولاسیون و حذف نویز
- دارای ولتاژ ورودی ۵ ولت جهت فرمان به خروجی
- قابلیت تحمل تا ۱۰ آمپر جریان خروجی به ازای هر کانال
- تعداد کانال: دو کانال
- نوع رله: رله‌های نیمه‌هادی (solid-state)
- ولتاژ کاری: ۵ ولت DC یا ۱۲ ولت DC
- جریان کاری: تا ۱۰ آمپر برای هر کانال
- ولتاژ خروجی: حداکثر ۲۵۰ ولت AC و ۳۰ ولت DC برای هر کانال
- تحمل جریان بار: حداکثر ۱۰ آمپر برای هر کانال
- زمان واکنش: کمتر از ۱۰ میلی‌ثانیه

## بررسی پایه های ماژول رله دو کانال

پایه های ماژول رله دو کانال بسته به نوع ماژول و تولید کننده متفاوت باشد، اما برای بیان یک نمونه، پایه های ماژول رله دو کانال را به صورت زیر می توان بیان کرد:

- VCC: پایه ای که ولتاژ تغذیه به ماژول رله دو کانال متصل می شود. معمولاً این پایه به ولتاژ ۵ ولت یا ۱۲ ولت DC نیاز دارد.
- GND: پایه ای که به زمین متصل می شود.
- IN1: پایه ورودی برای کنترل رله اول. این پایه به یک پورت دیجیتال از پایه مشترک برای رله اول.
- NO1: پایه خروجی آزاد برای رله اول. اگر رله فعال باشد، این پایه به COM1 متصل می شود.
- NC1: پایه خروجی بسته برای رله اول. اگر رله غیرفعال باشد، این پایه به COM1 متصل می شود.
- IN2: پایه ورودی برای کنترل رله دوم. این پایه به یک پورت دیجیتال از میکروکنترلر، برد آردوینو یا سایر سیستم های کنترلی متصل می شود.
- COM2: پایه مشترک برای رله دوم.
- NO2: پایه خروجی آزاد برای رله دوم. اگر رله فعال باشد، این پایه به COM2 متصل می شود.



- NC2: پایه خروجی بسته برای رله دوم. اگر رله غیرفعال باشد، این پایه به COM2 متصل می شود.

## ماژول کارت خوان RC522

ماژول کارت خوان RFID RC522 یک ماژول کارت خوان NFC/RFID با فرکانس ۱۳/۵۶ مگاهرتز است که به راحتی می‌تواند انواع کارت‌های RFID و NFC را تشخیص دهد و اطلاعات موجود در آن‌ها را بخواند. این ماژول از یک الگوریتم تشخیص و ارتباط با کارت‌های RFID و NFC با نام MFRC522 پشتیبانی می‌کند و با اتصال به میکروکنترلرهایی مانند Arduino و ESP8266، به راحتی می‌تواند از آن برای پروژه‌های مختلفی استفاده کرد. از کاربردهای این ماژول می‌توان به پروژه‌های امنیتی و اعتبارسنجی مثل درب بازکن اتوماتیک، سیستم حضور و غیاب و سیستم‌های دسترسی فیزیکی اشاره کرد. همچنین این ماژول در صنایع مختلفی مانند حمل و نقل، پیشرفته‌سازی شهری و هوشمندسازی ساختمان‌ها نیز کاربرد دارد.

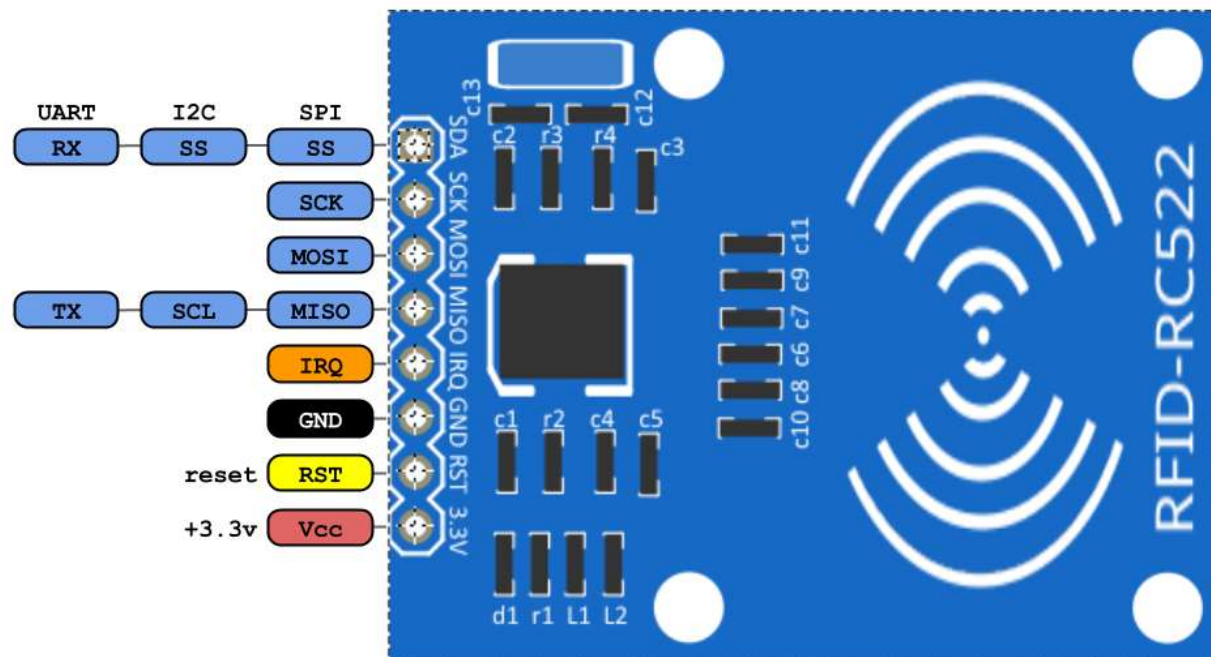
### مشخصات فنی RFID RC522

- فرکانس عملیاتی: ۱۳/۵۶ مگاهرتز
- محدوده فاصله از کارت: تا ۳ سانتی‌متر
- پشتیبانی از کارت‌های RFID: MIFARE 1 S50، MIFARE 1 S70، MIFARE Ultralight، MIFARE Pro و NTAG203
- پشتیبانی از کارت‌های NFC: NTAG213 و NTAG215
- پشتیبانی از رابط SPI با سرعت بالا
- دارای یک آنتن کوچک داخلی جهت ارتباط با کارت‌ها
- دارای ۸ پین ورودی و خروجی برای ارتباط با میکروکنترلرها
- ولتاژ کاری: ۳/۳ ولت
- جریان مصرفی: کمتر از ۱۳ میلی‌آمپر
- ابعاد فیزیکی: ۴۰x۶۰ میلی‌متر

## بررسی پایه های RFID RC522

پایه های برد RFID RC522 به شرح زیر می باشد:

- GND: اتصال به منبع منفی (گراند)
- IRST: اتصال به یک پایه دلخواه در آردوینو
- ۳/۳ V: اتصال به منبع تغذیه با ولتاژ ۳/۳ ولت
- SDA: اتصال به پایه دیجیتال دلخواه در آردوینو
- SCK: اتصال به پایه دیجیتال دلخواه در آردوینو
- MOSI: اتصال به پایه دیجیتال دلخواه در آردوینو
- MISO: اتصال به پایه دیجیتال دلخواه در آردوینو
- IRQ:
- اتصال به یک پایه دلخواه در آردوینو



## شرح و اهداف پروژه

در این پروژه قصد داریم در حوزه اینترنت اشیاء (iot) کار کنیم هدف از این پروژه کنترل درب بر بستر وب و RFID می باشد .

البته که درب در اینجا یک مثال است پس از اتمام پروژه می توانیم هر وسیله ای که با برق 220V کار می کند را به رله وصل و آن را کنترل کنیم .

لازم به ذکر است که هنگامی که کاربر قرار است از وب استفاده کند باید مورد احراز هویت (Authentication) قرار گیرد.

با توجه به موضوع پروژه من بهترین و مناسب ترین برد از نظر (دقت، کیفیت ، هزینه ، کامل بودن امکانات و کتابخانه ) برد esp8266 بود که من هم این برد را انتخاب کردم .

همچنین من برای کنترل وسایل برقی که در ایران از ولتاژ 220v استفاده می کنند باید رله هم برای قطع و وصل جریان استفاده می کردم .

## زبان های مورد استفاده در این پروژه

این پروژه ۰ تا ۱۰۰ کد نویسی بوده و از هیچ CMS بهره نگرفته شده البته که ب راحتی میتوان از آن ها بهره برد تا هم سرعت بالا رفته و هم عملکرد بهینه تری داشته باشیم.

### برخی از زبان های مورد استفاده در پروژه:

- HTML برای ساخت وب سایت
- CSS برای ساخت وب سایت
- PHP برای backend
- C++ برنامه نویسی 82266 esp
- SQL برای مدیریت کاربران



## نحوه کار پروژه

ابتدا که برق ۵ ولت را به برد می‌رسانیم دستگاه شروع به کار کرده و در عرض چند میلی ثانیه (با روشن شدن چراغ سبز) آماده به کار می‌باشد.

همچنین در حال جستجو در ssid های اطراف خود برای برقراری اتصال به شبکه (اینترنت) می‌باشد.

با نزدیک کردن nfc و یا کنترل از طریق وب می‌توان رله را کنترل کرد.

پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

لینک ها

آدرس ایمیل : [mostafa.ms607@gmail.com](mailto:mostafa.ms607@gmail.com)

آدرس وب سایت برای کنترل پروژه :

<https://payner.ir/Project/>

**User:**admin

**password:**admin

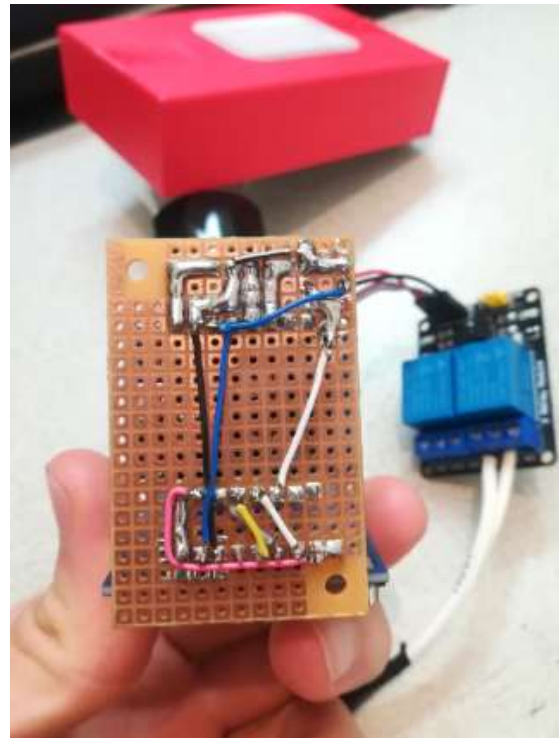
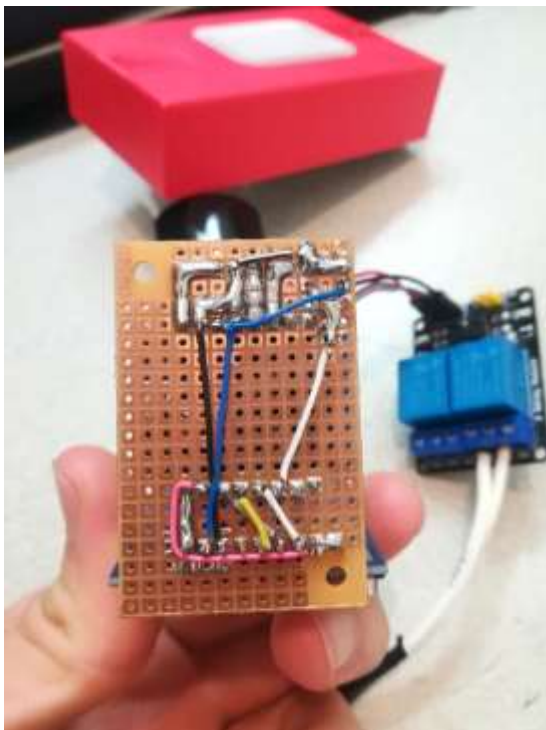
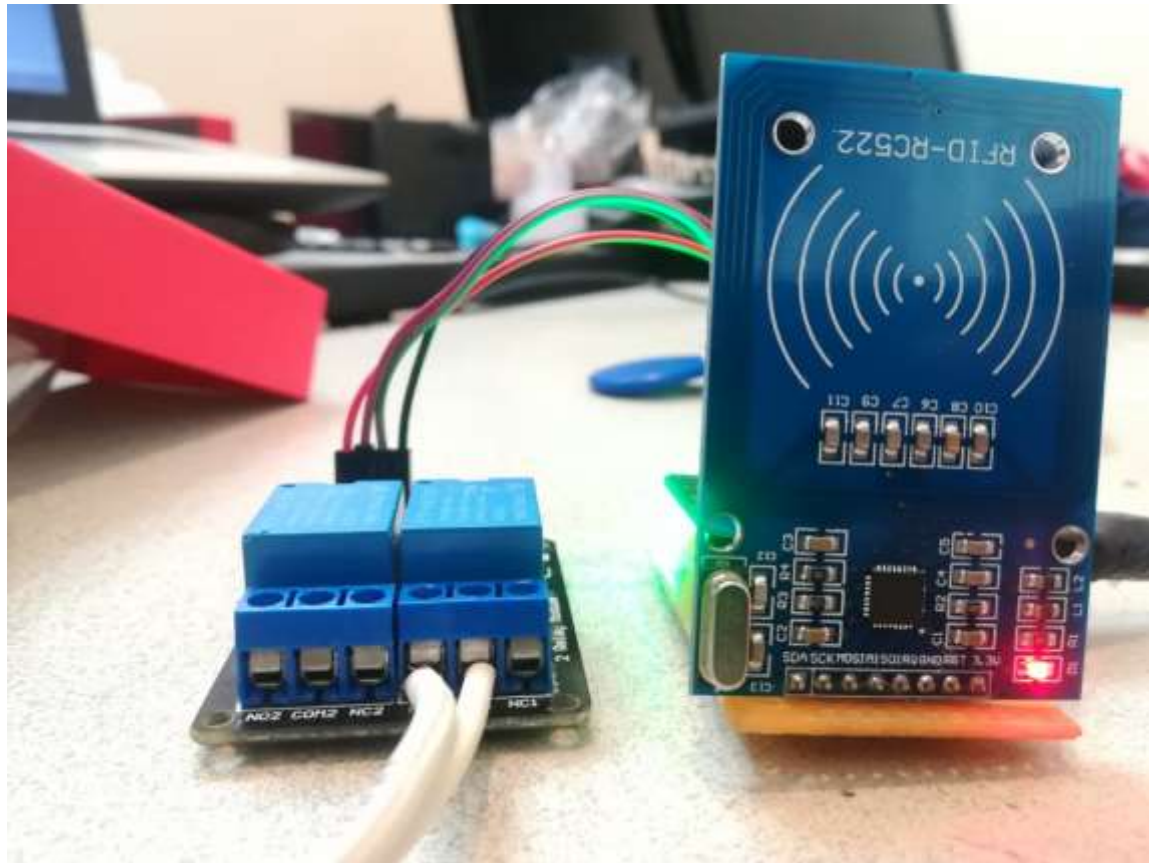
آدرس گیت هاب برای دریافت کدها :

<https://github.com/mostafadastghibi/IoT>

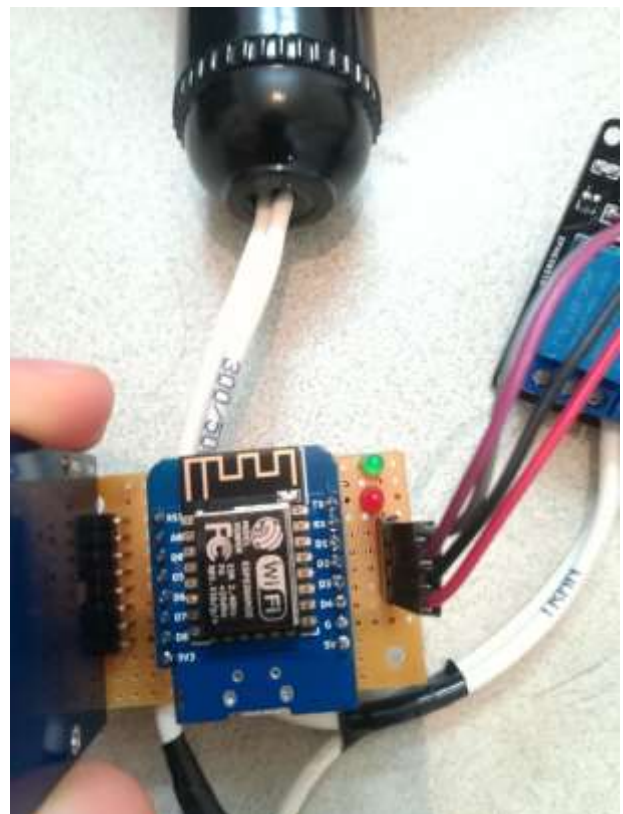
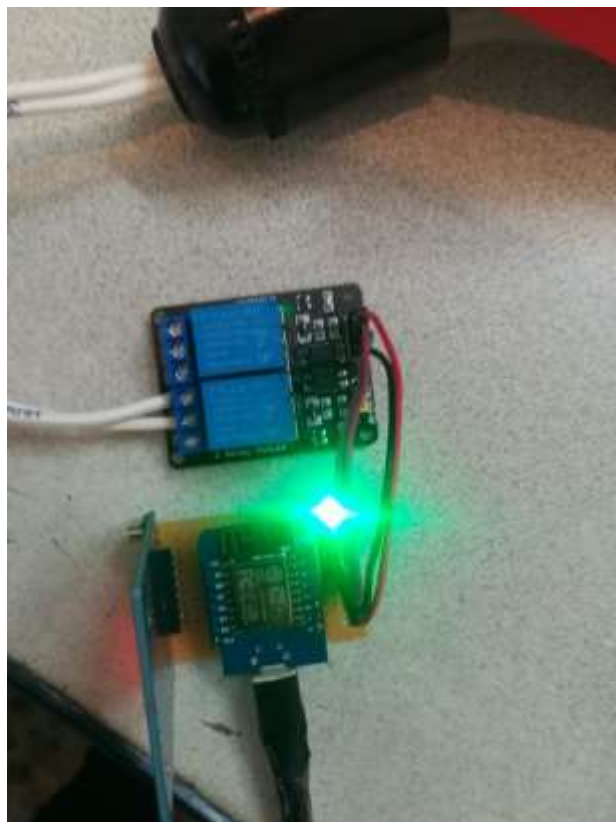
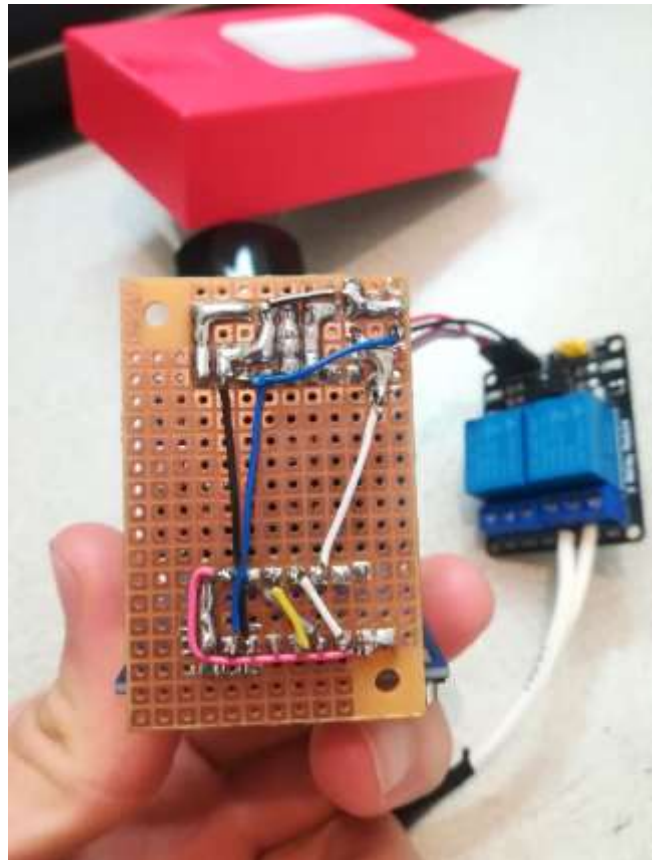
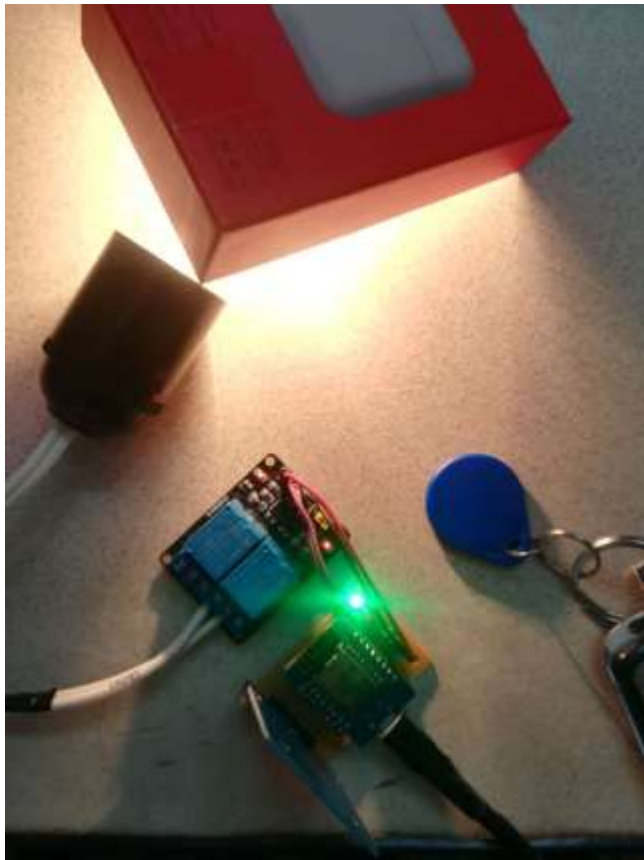
مشخصات wi-fi

**Ssid:**" NTW"

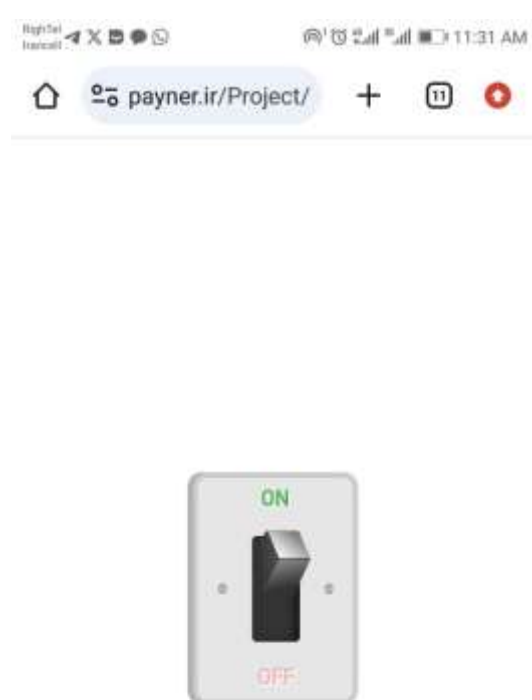
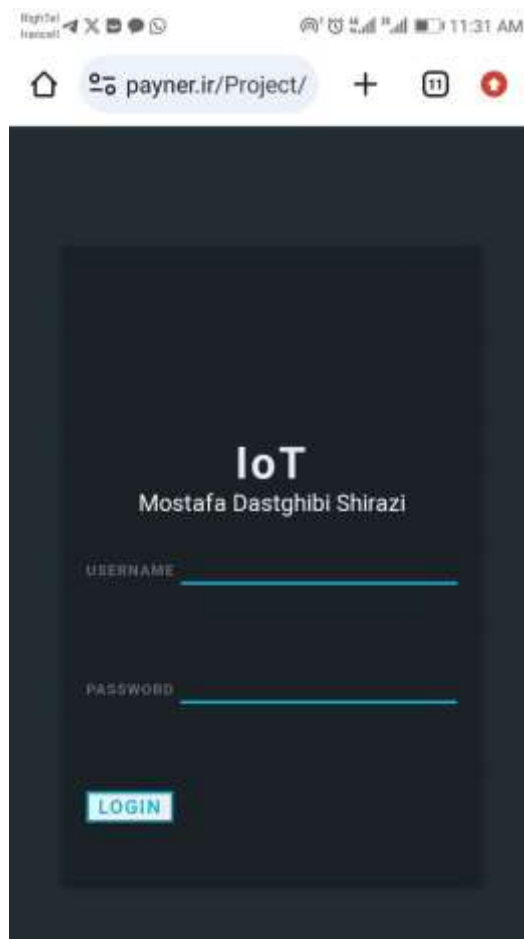
**Password:**" 0303301306004"



## پروژه اینترنت اشياء-مصطفی دست غیبی شیرازی



## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی



//وارد کردن کتابخانه ها

#include <EEPROM.h> // کتابخانه حافظه eeprom

#include <ESP8266WiFi.h> // کتابخانه esp8266

#include <MFRC522.h> // کتابخانه ماژول ارای دی

#include <SPI.h>

#include <ESP8266HTTPClient.h> // کتابخانه راه اندازی درخواست http

#include <WiFiClient.h>

// تعریف پایه های رله و ال ای دی و پایه های ماژول ارای دی

#define RST\_PIN D3

#define SS\_PIN D8

#define RELAY\_1\_PIN D1

#define RELAY\_2\_PIN D2

#define LED\_GREEN\_PIN D4

#define LED\_RED\_PIN D0

// رمز و پسورد وای فای مورد نظر

const char\* ssid = "NTW"; // The SSID (name) of the Wi-Fi network you want to connect to

const char\* password = "0303301306004"; // The password of the Wi-Fi network

// کارت خوانده شده

byte presentedCard[4];

boolean state = false;

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

// آدرس سرور

```
const char* serverAddress = "http://www.payner.ir:80/Project/process.php"; //Your Domain name  
with URL path or IP address with path
```

```
unsigned long lastTime = 0;
```

```
//Set timer to 5 seconds (5000)
```

```
unsigned long timerDelay = 8000;
```

// مقدار دهی اولیه مازول ا ر اف ای دی

```
MFRC522 rf(SS_PIN, RST_PIN);
```

```
void setup() {
```

// راه اندازی سریال در بادریت مورد نظر

```
Serial.begin(115200);
```

```
delay(100);
```

```
EEPROM.begin(512);
```

```
delay(100);
```

// تعیین ورودی یا خروجی بودن پین ها

```
pinMode(RELAY_1_PIN, OUTPUT);
```

```
pinMode(RELAY_2_PIN, OUTPUT);
```

```
pinMode(LED_GREEN_PIN, OUTPUT);
```

```
pinMode(LED_RED_PIN, OUTPUT);
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

// وضعیت اولیه پین

```
digitalWrite(RELAY_1_PIN, HIGH);
```

```
digitalWrite(RELAY_2_PIN, HIGH);
```

```
digitalWrite(LED_RED_PIN, LOW);
```

station در مود esp8266 // راه اندازی

```
WiFi.mode(WIFI_STA);
```

// اتصال به وای فای مورد نظر

```
WiFi.begin(ssid, password);
```

// ۱۰ ثانیه چشمک زدن چراغ قرمز به منظور اتصال به وای فای

```
int count = 0;
```

```
while(count < 10){
```

```
    digitalWrite(LED_RED_PIN, HIGH);
```

```
    delay(500);
```

```
    digitalWrite(LED_RED_PIN, LOW);
```

```
    delay(500);
```

```
    count++;
```

```
{
```

rfid // شروع کار ماژول

```
SPI.begin();
```

```
rf.PCD_Init();
```

```
{
```



## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
void loop() {  
  // شرط برقراره در صورتی که کارتی به مازول نزدیک نشه - اگه کارتی تشخیص داده بشه وارد شرط نمیشه  
  if ( ! rf.PICC_IsNewCardPresent() || ! rf.PICC_ReadCardSerial() ) {  
    اگه از تایمی کی بالا تعریف کردیم گذشته بود وارد شرط میشه //  
    if ((millis() - lastTime) > timerDelay) {  
      digitalWrite(LED_GREEN_PIN, LOW);  
      if(WiFi.status() == WL_CONNECTED) { // بررسی اتصال به وای فای  
        digitalWrite(LED_RED_PIN, HIGH);  
        String response = httpGETRequest(serverAddress); // ایجاد درخواست  
        Serial.print("Server -> ");  
        Serial.println(serverAddress);  
        Serial.print("Response -> ");  
        Serial.println(response);  
        if(response != "{}") {  
          if(response == "off"){  
            state = false;  
          }  
          else if (response == "on"){  
            state = true;  
          }  
          {  
            control();  
          }  
          {  
            else {  
              digitalWrite(LED_RED_PIN, LOW);  
            }  
          }  
        }  
      }  
    }  
  }  
}
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
lastTime = millis();  
  
{  
  
digitalWrite(LED_GREEN_PIN, HIGH);  
  
delay(50);  
  
return;  
  
{  
  
for (byte i = 0; i < rf.uid.size; i++) { // خواندن کارت  
    presentedCard[i] = rf.uid.uidByte[i];  
  
    {  
  
        if (checkID(presentedCard)) { // بررسی وجود کارت در حافظه - اگه بود رله رو کنترل کن  
            control();  
  
            { else { // اگه نبود بیا برو تو حالت افزودن کارت  
                int count = 0;  
  
                while(! rf.PICC_IsNewCardPresent() || ! rf.PICC_ReadCardSerial()) { // تا ۵ ثانیه اگه کارت قدیمی نزد  
                    کارت جدیدو اضافه نکن  
  
                    digitalWrite(LED_RED_PIN, LOW);  
  
                    digitalWrite(LED_GREEN_PIN, LOW);  
  
                    delay(250);  
  
                    digitalWrite(LED_RED_PIN, HIGH);  
  
                    digitalWrite(LED_GREEN_PIN, HIGH);  
  
                    delay(250);
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
if (count >= 10) {  
    return;  
}  
count++;  
  
{  
  
byte userCard[4];  
for (byte i = 0; i < rf.uid.size; i++) {  
    userCard[i] = rf.uid.uidByte[i];  
}  
  
if(checkID(userCard)){  
    addNewUsers(presentedCard);  
}  
  
{  
  
    delay(400);  
}  
  
// تابع کنترل رله ها  
void control() {  
    if(!state) {  
        digitalWrite(RELAY_1_PIN, HIGH);  
        digitalWrite(RELAY_2_PIN, HIGH);  
        state = true;  
    }  
}
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
{ else {  
    digitalWrite(RELAY_1_PIN, LOW);  
    digitalWrite(RELAY_2_PIN, LOW);  
    state = false;  
    {  
    {  
  
    // تابع بررسی آی دی کارت خوانده شده در حافظه  
    boolean checkID( byte card[] ) {  
        byte users = EEPROM.read(0);  
        byte storedCard[4];  
  
        if(users <= 0){  
            addNewUsers(card);  
            return true;  
        }  
  
        for ( byte user = 0 ; user < users ; user++) {  
            for ( byte i = 0 ; i < 4 ; i++) {  
                storedCard[i] = EEPROM.read( user * 4 + 1 + i );  
            }  
            boolean match = true;  
            for ( byte i = 0; i < 4; i++ ) {  
                if ( card[i] != storedCard[i]) {
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
match = false;

break;

{

{

    if(match){

        return true;

    }

    {

        return false;

    }

}

// تابع افزودن کاربر جدید در حافظه فلش
void addNewUsers(byte newUser[]) {

    byte users = EEPROM.read(0);

    users += 1;

    EEPROM.write(0, users);

    for ( byte i = 0 ; i < 4 ; i++ ) {

        EEPROM.write( (users - 1) * 4 + 1 + i , newUser[i] );

    }

    if (!EEPROM.commit()) {

        Serial.println("ERROR! EEPROM commit failed");

    }

}
```

## پروژه اینترنت اشیاء-مصطفی دست غیبی شیرازی

```
{
```

```
get به صورت http// تابع درخواست
```

```
String httpGETRequest(const char* serverName) {
```

```
    WiFiClient client;
```

```
    HTTPClient http;
```

```
// تایم اوت
```

```
    http.setTimeout(3000);
```

```
    http.begin(client, serverName);
```

```
// Send HTTP GET request
```

```
    int httpResponseCode = http.GET();
```

```
    String payload = "{}";
```

```
    if (httpResponseCode == 200) { // اگر کد ۲۰۰ بود اطلاعات صفحه رو بخون
```

```
        payload = http.getString();
```

```
{
```

```
    http.end();
```

```
    return payload;
```

```
{
```