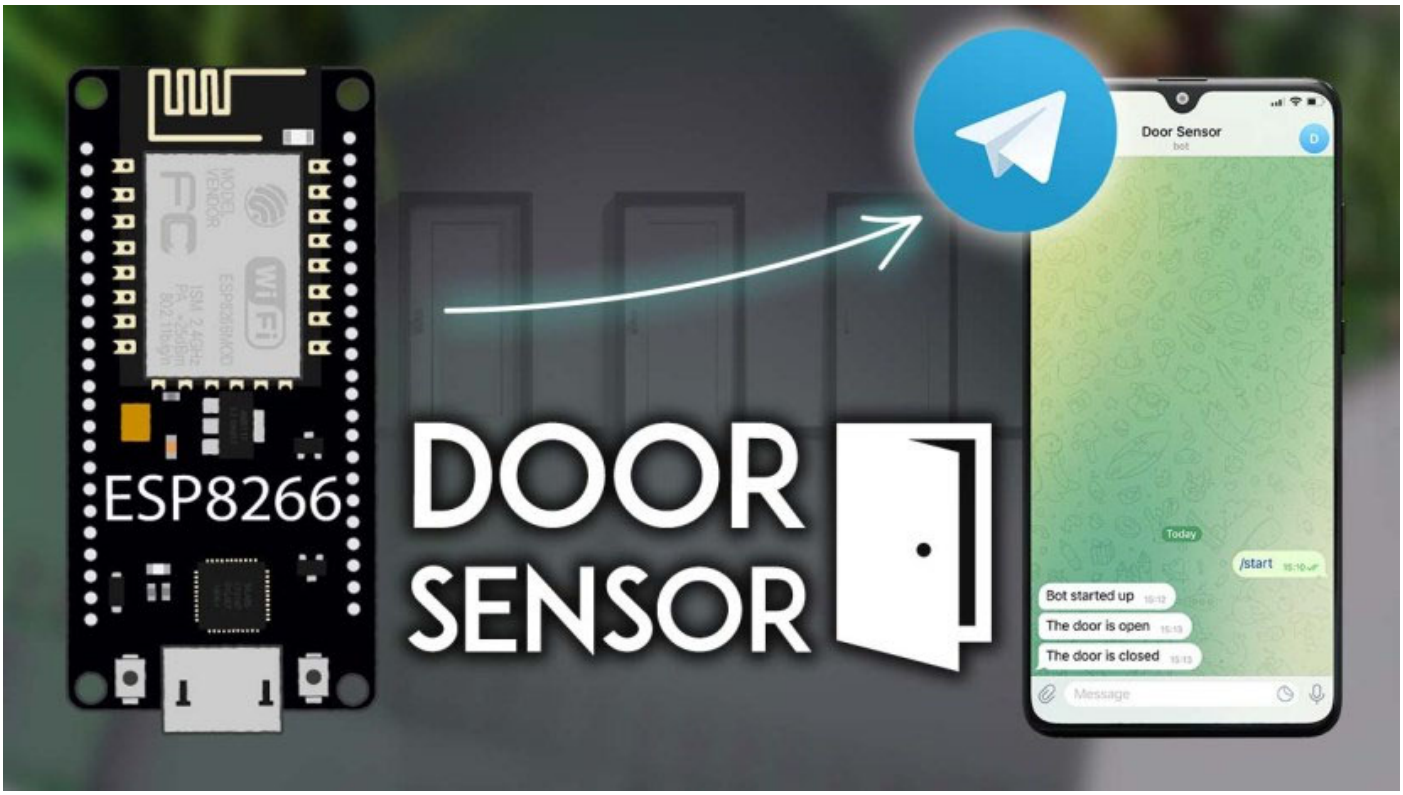
ESP8266 帶有 Telegram 通知的 NodeMCU 門狀態監控器

ESP8266 NodeMCU 門狀態監控器與 Telegram 通知 |隨機書教程 (randomnerdtutorials.com)

在本專案中，您將使用 ESP8266 NodeMCU 板和磁簧開關監控門的狀態。每當門改變狀態時，您都會在您的 Telegram 帳戶中收到一條消息：打開或關閉。只要您可以在智慧手機上訪問互聯網，無論您身在何處，都會收到通知。ESP8266 板將使用 Arduino IDE 進行程式設計。



我們有一個類似的教程，它發送電子郵件而不是 Telegram 消息：

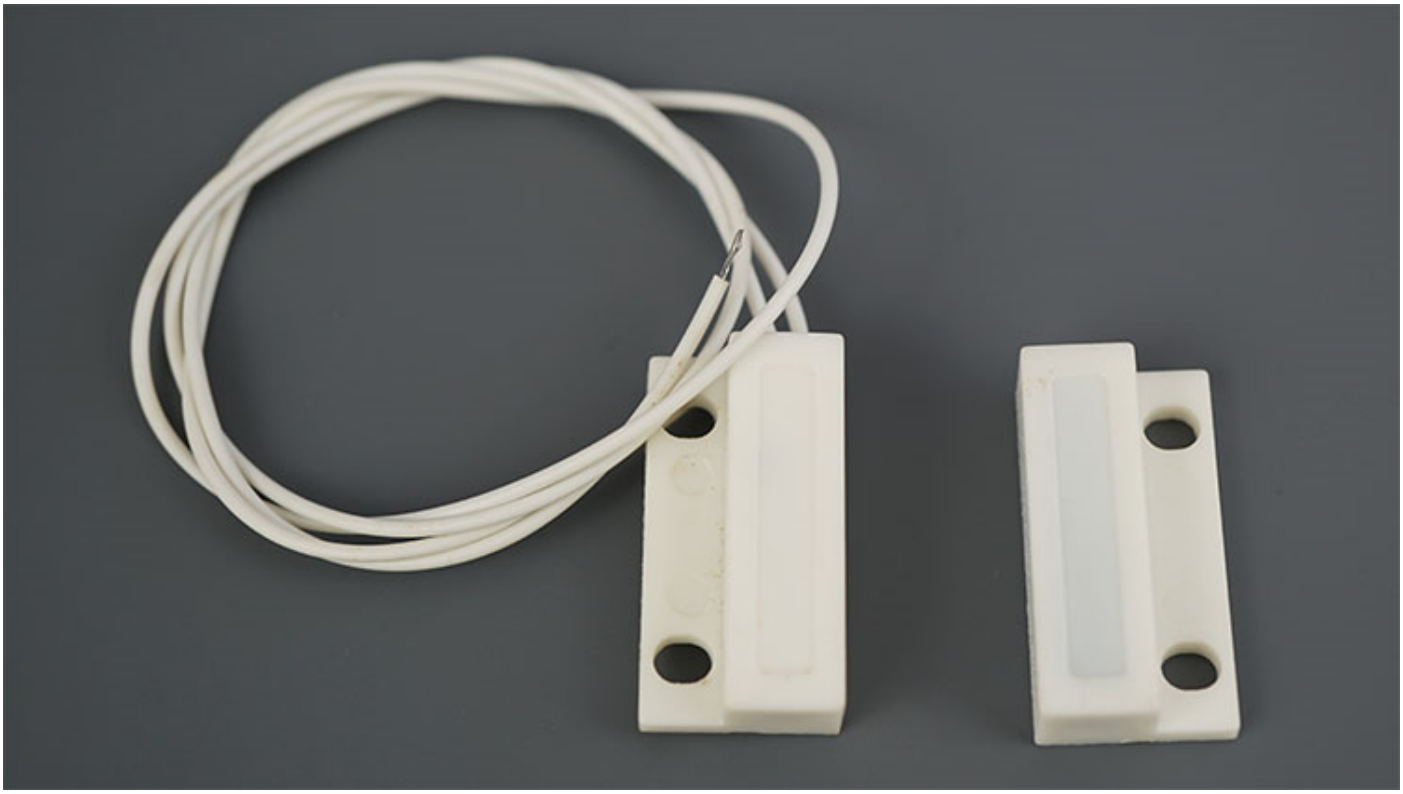
- ESP8266 帶電子郵件通知的 NodeMCU 門狀態監控器 （IFTTT）

閱讀 ESP32 指南：帶有 Telegram 通知的門狀態監控器

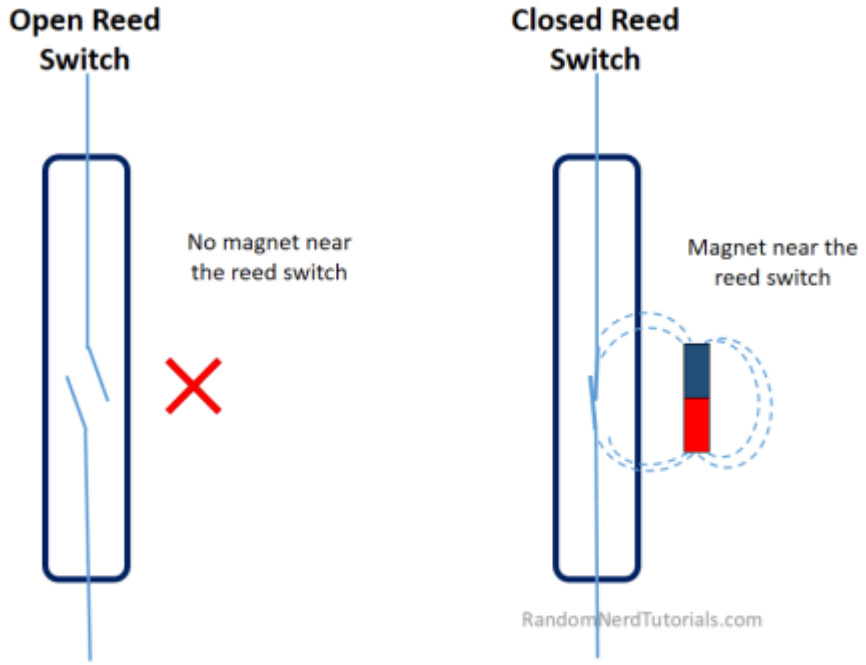
專案概況

在這個專案中，我們將創建一個 Telegram Bot，每當門改變狀態時，它都會向你的 Telegram 帳戶發送消息。為了檢測變化，我們將使用磁性觸點開關。

磁性接觸開關基本上是封裝在塑膠外殼中的簧片開關，因此您可以輕鬆地將其應用於門、窗或抽屜，以檢測它是打開還是關閉。



當磁鐵靠近開關時，電路閉合 - 門關閉。當磁鐵遠離開關時（門打開），電路打開。見下圖。



我們可以將簧片開關連接到 ESP8266 GPIO 來檢測其狀態的變化。

Telegram 簡介

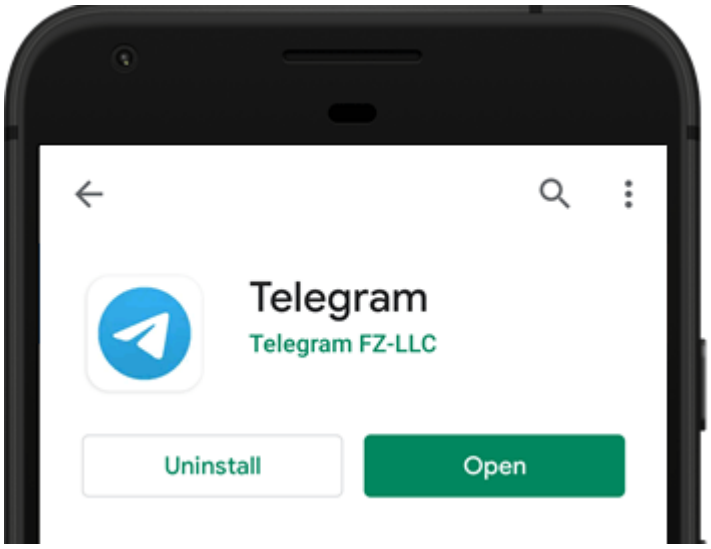
電報 Messenger 是一種基於雲的即時消息和 IP 語音服務。您可以輕鬆地將其安裝在智能手機（Android 和 iPhone）或電腦（PC、Mac 和 Linux）上。它是免費的，沒有任何廣告。Telegram 允許您創建可以與之交互的機器人。

"爬蟲程式是在 Telegram 中運行的第三方應用程式。用戶可以通過向機器人發送消息、命令和內聯請求來與機器人交互。您使用對 Telegram Bot API 的 HTTPS 請求來控制您的機器人"

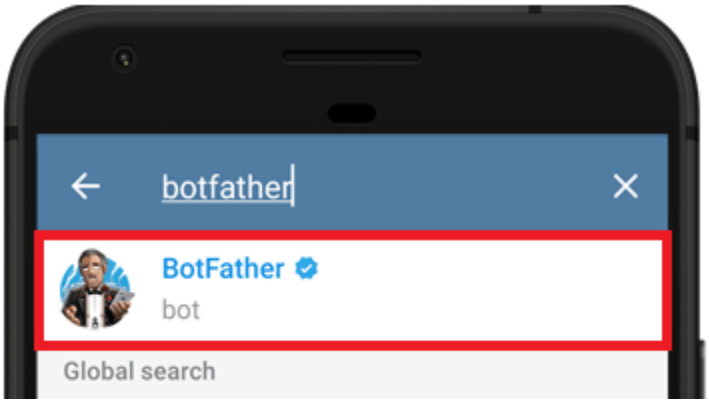
ESP8266 將與 Telegram 機器人交互，以將消息發送到您的 Telegram 帳戶。每當門改變狀態時，您都會在智能手機上收到通知（只要您可以訪問互聯網）。

創建 Telegram 機器人

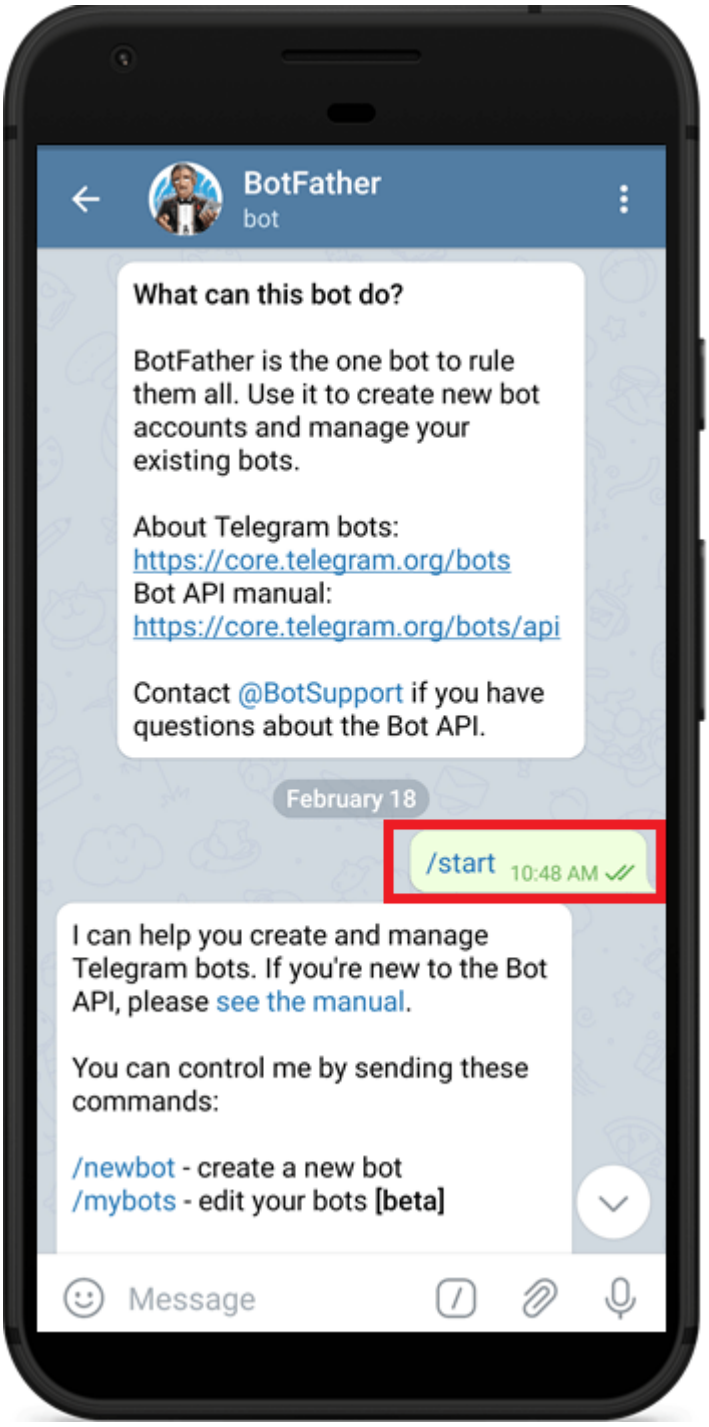
前往 Google Play 或 App Store，下載並安裝 **Telegram**。



打開 Telegram 並按照以下步驟創建 Telegram 機器人。首先，搜索 **"botfather"** 並按兩下 BotFather，如下所示。或者 t.me/botfather 智慧手機打開此連結。

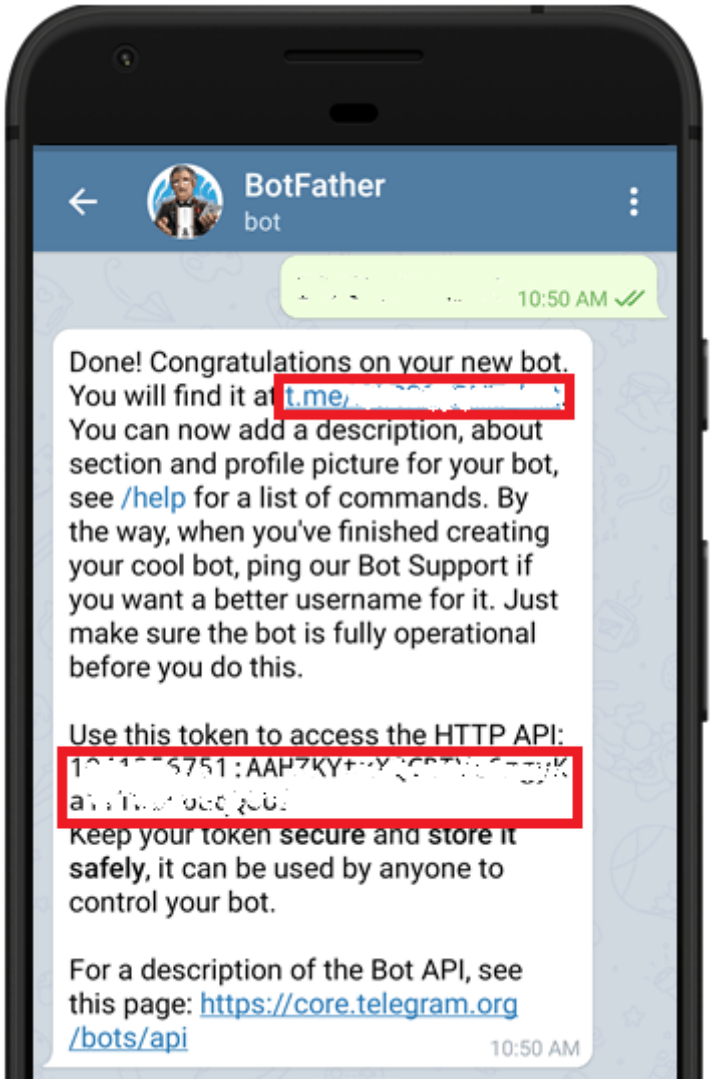


以下視窗應打開，系統將提示您按兩下**開始**按鈕。



鍵入 **/newbot** 並按照說明創建您的機器人。為其指定 name 和 username。我的叫門感測器，使用者名為 `ESPDoorSensorBot` 機器人.

如果您的機器人創建成功，您將收到一條消息，其中包含用於訪問機器人和**機器人令牌**的連結。保存機器人令牌，因為您將需要它，以便 ESP8266 可以與機器人交互。

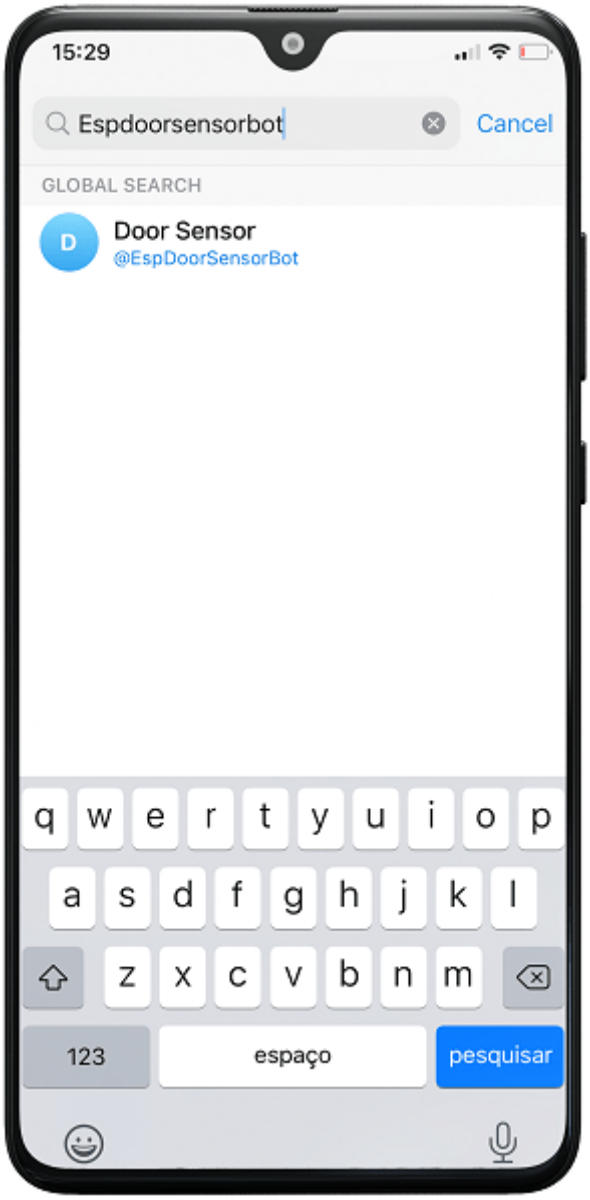


向 Bot 發送消息

這一步非常重要。不要錯過。否則，專案將無法運行。

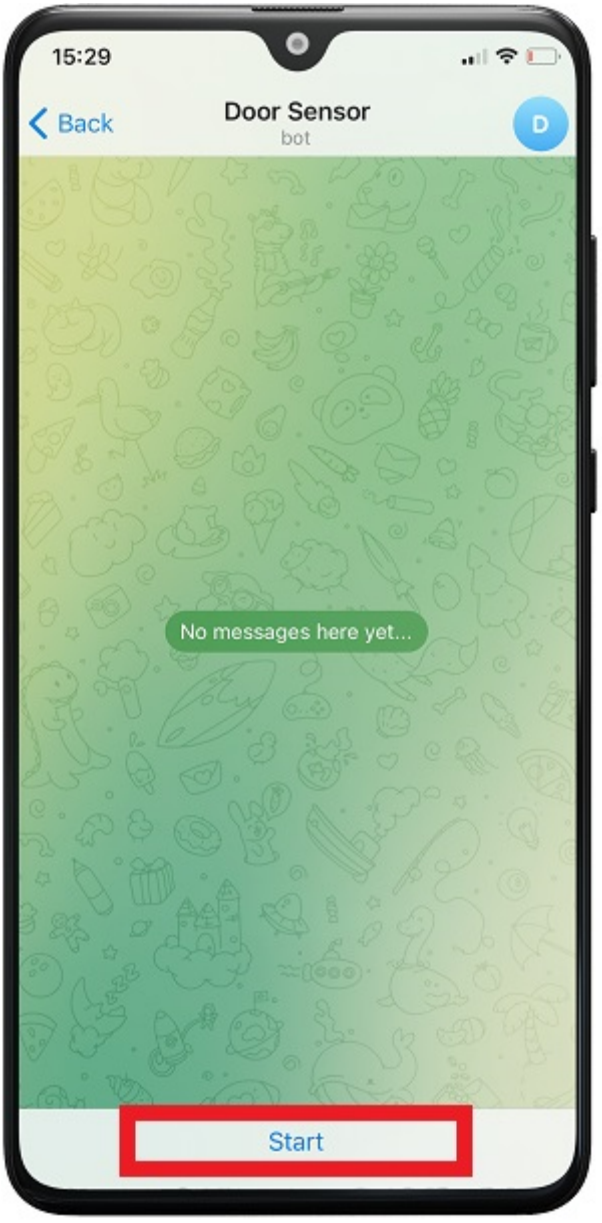
您必須先從您的 Telegram 帳戶向您的 Telegram Bot 發送消息，然後才能向您發送消息。

1）傳回 chats （聊天） 選項卡，然後在搜索欄位中鍵入 bot 的使用者名。



2）選擇您的機器人以開始對話。

3）按兩下「開始」 連結。

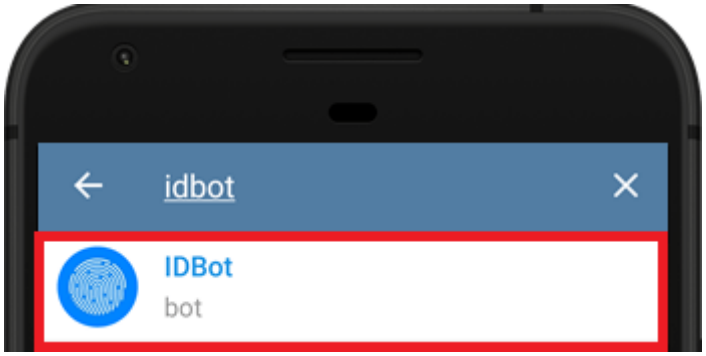


就是這樣！您可以繼續下一部分。

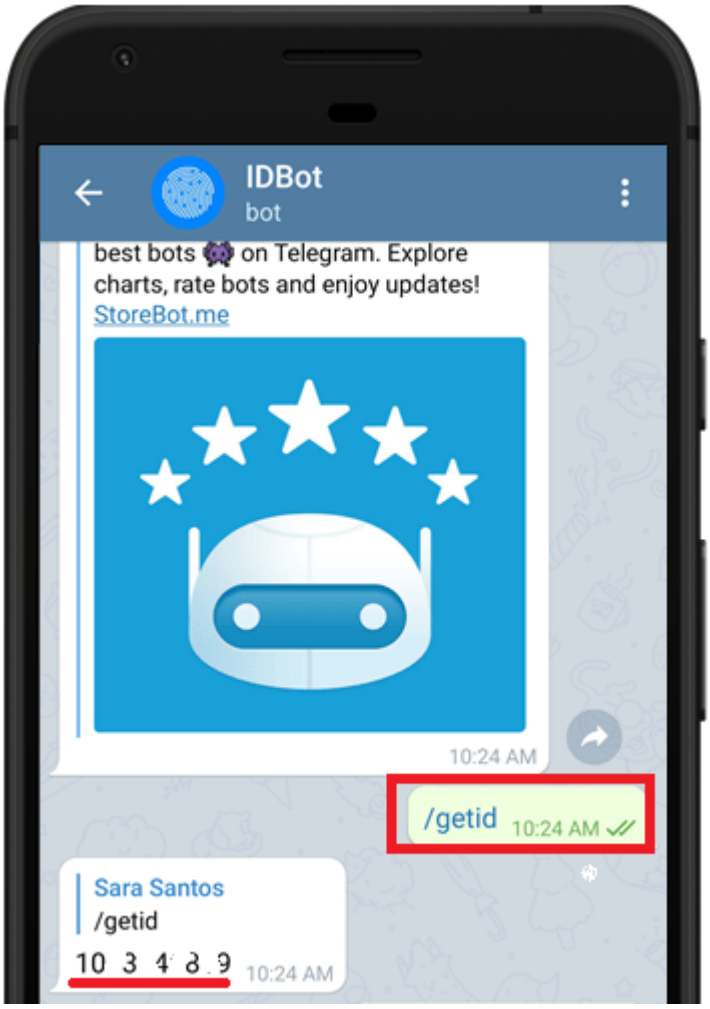
獲取您的 **Telegram** 使用者 **ID**

要向您的 Telegram 帳戶發送消息，機器人需要知道您的使用者 ID。

在您的 Telegram 帳戶中，搜索「**myidbot**」或在您的智慧手機上打開此連結 t.me/myidbot。



開始與該機器人的對話並鍵入 **/getid**。您將收到包含您的使用者 ID 的回復。請儲存該**使用者 ID**，因為在本教程的後面部分將需要用到它。



準備 Arduino IDE

我們將使用 Arduino IDE 對 ESP8266 板進行程式設計，因此請確保您已將其安裝在 Arduino IDE 中。

- 在 Arduino IDE 中安裝 ESP8266 板（Windows、Mac OS X、Linux）

通用 Telegram 機器人庫

為了與 Telegram 機器人交互，我們將使用由 Brian Lough 建立的通用 Telegram 機器人庫，該庫為 Telegram 機器人 API 提供了一個簡單的介面。
按照後續步驟安裝最新版本的庫。

1. 按兩下此處下載 Universal Arduino Telegram Bot 庫。
2. 轉到 **Sketch > Include Library > Add.ZIP Library...**
3. 添加您剛剛下載的庫。

重要提示：請勿通過 Arduino Library Manager 安裝庫，因為它可能會安裝已棄用的版本。

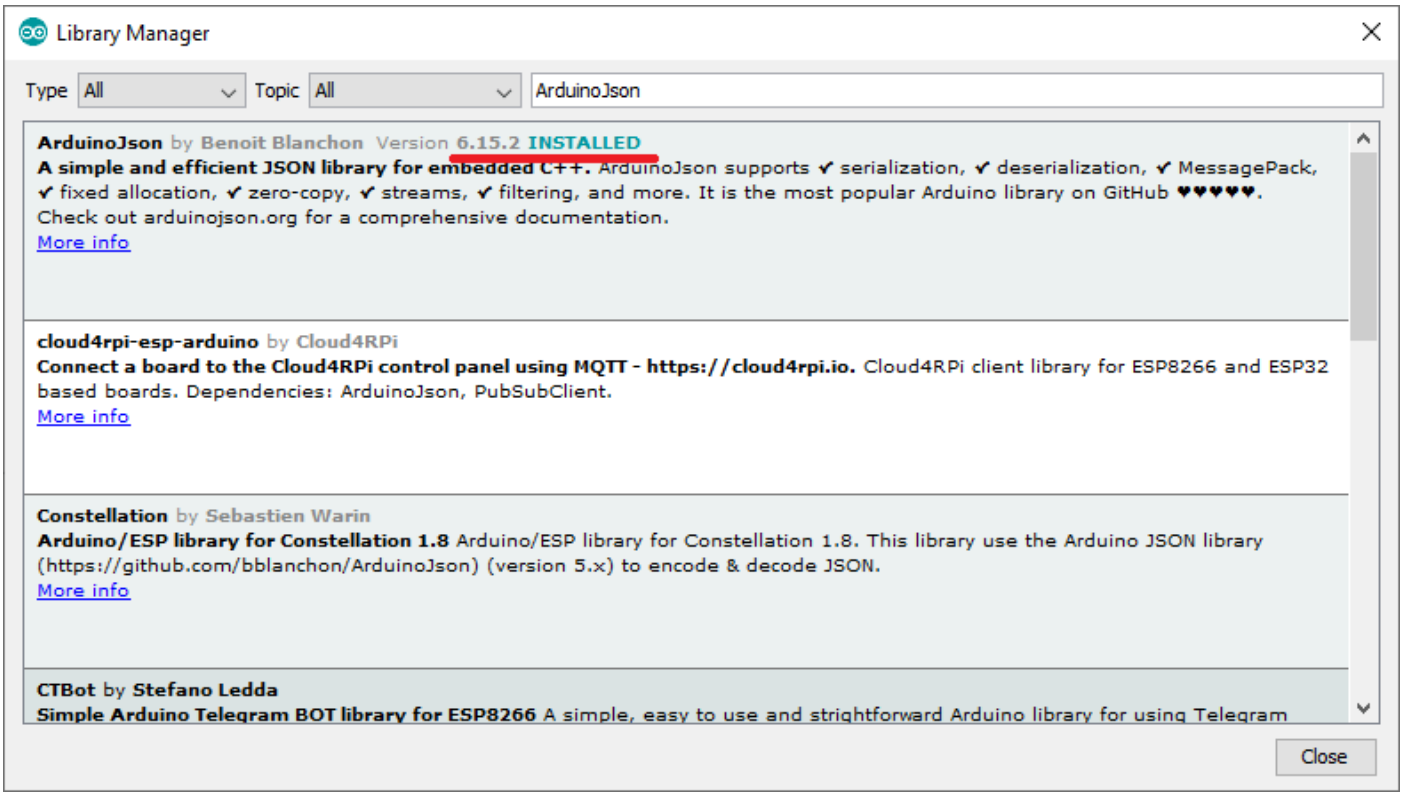
有關該庫的所有詳細資訊，請查看通用 Arduino Telegram 機器人庫 GitHub 頁面。

ArduinoJson 庫

您還必須安裝 ArduinoJson 庫。按照後續步驟安裝庫。

1. 轉到 **Skech > 包括庫 > 管理庫**。
2. 搜索 “ArduinoJson”。
3. 安裝庫。

我們使用的是 ArduinoJson 庫版本 6.15.2。



所需零件

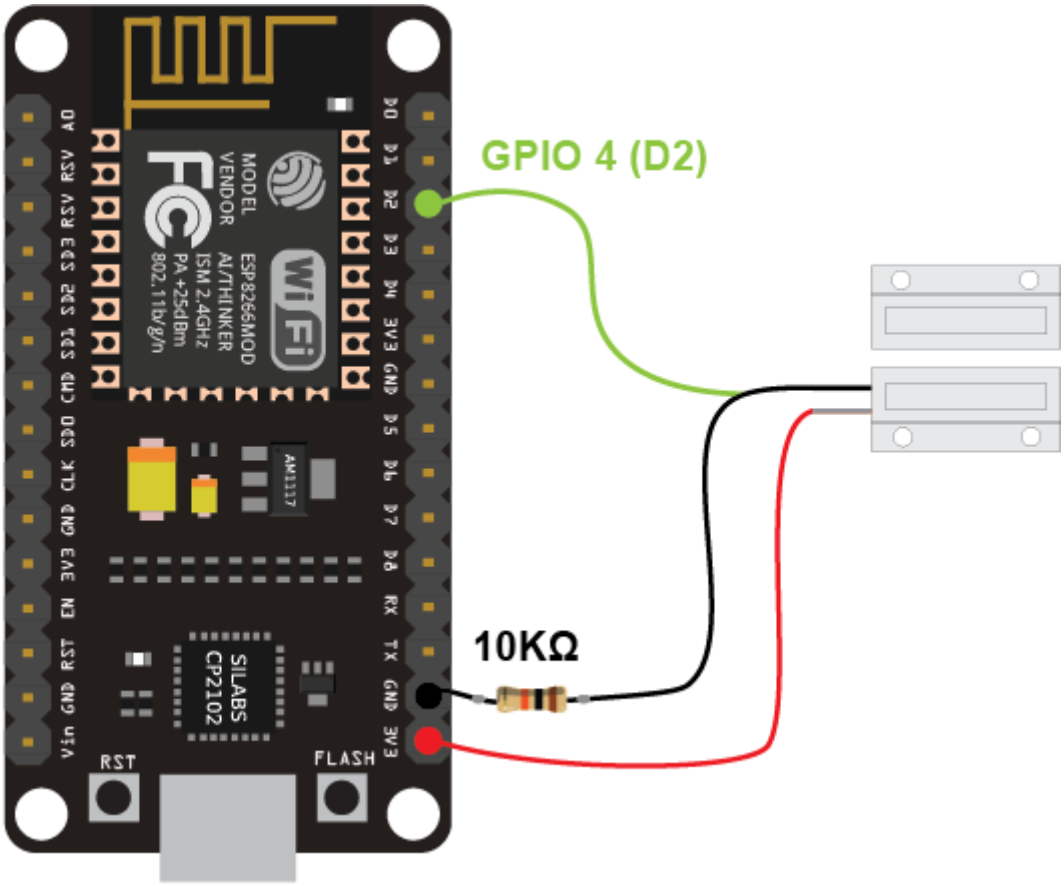
以下是完成此專案所需的硬體：

- ESP8266 – 閱讀最佳 ESP8266 開發板
- 1× 磁簧開關
- 1× 10kΩ 電阻
- 1× 麵包板
- 跳線

您可以使用前面的連結或直接訪問 MakerAdvisor.com/tools 以最優惠的價格找到適合您專案的所有零件！

原理圖 – 帶簧片開關的 ESP8266

我們將簧片開關連接到通用輸出 4（D2），但您可以將其連接到任何合適的 GPIO。



法典

將下面的草圖複製到您的 Arduino IDE。將 SSID、密碼、BOT 令牌和使用者 ID 替換為您的憑證。

```
/*
  Rui Santos
  Complete project details at https://RandomNerdTutorials.com/esp8266-nodemcu-door-status-telegram/
```

```
  Permission is hereby granted, free of charge, to any person
obtaining a copy
  of this software and associated documentation files.

  The above copyright notice and this permission notice shall be
included in all
  copies or substantial portions of the Software.
*/

#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

// Set GPIOs for LED and reedswitch
const int reedSwitch = 4;
const int led = 2; //optional

// Detects whenever the door changed state
bool changeState = false;

// Holds reedswitch state (1=opened, 0=close)
bool state;
String doorState;

// Auxiliary variables (it will only detect changes that are 1500
milliseconds apart)
unsigned long previousMillis = 0;
const long interval = 1500;

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Initialize Telegram BOT
#define BOTtoken "XXXXXXXXXX:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"  //
your Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
#define CHAT_ID "XXXXXXXXXX"
```

```
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
UniversalTelegramBot bot(BOTtoken, client);

// Runs whenever the reedswitch changes state
ICACHE_RAM_ATTR void changeDoorStatus() {
  Serial.println("State changed");
  changeState = true;
}

void setup() {
  // Serial port for debugging purposes
  Serial.begin(115200);
  configTime(0, 0, "pool.ntp.org");      // get UTC time via NTP
  client.setTrustAnchors(&cert); // Add root certificate for
api.telegram.org


  // Read the current door state
  pinMode(reedSwitch, INPUT_PULLUP);
  state = digitalRead(reedSwitch);

  // Set LED state to match door state
  pinMode(led, OUTPUT);
  digitalWrite(led, state);

  // Set the reedswitch pin as interrupt, assign interrupt function
and set CHANGE mode
  attachInterrupt(digitalPinToInterrupt(reedSwitch), changeDoorStatus,
CHANGE);

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
      delay(500);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");

    bot.sendMessage(CHAT_ID, "Bot started up", "");
```

```
}

void loop() {
  if (changeState){
    unsigned long currentMillis = millis();
    if(currentMillis - previousMillis >= interval) {
      previousMillis = currentMillis;
      // If a state has occured, invert the current door state
        state = !state;
        if(state) {
          doorState = "closed";
        }
        else{
          doorState = "open";
        }
        digitalWrite(led, state);
        changeState = false;
        Serial.println(state);
                      Serial.println(doorState);

        //Send notification
        bot.sendMessage(CHAT_ID, "The door is " + doorState, "");
    }
  }
}
```

代碼的工作原理

繼續閱讀以了解代碼的工作原理，或繼續閱讀演示部分。
首先，包含所需的庫。

```
#include <ESP8266WiFi.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
```

設置簧片開關和 LED 的 GPIO（板載 LED 為 GPIO 2）。當門打開時，我們將點亮板載 LED。

```
const int reedSwitch = 4;
const int led = 2; //optional
```

這更改狀態 boolean 變數指示門是否已更改狀態。

```
bool changeState = false;
```
這州變數將保持簧片開關狀態，而 `doorState`，顧名思義，將保持門狀態 - closed 或 opened。
```
bool state;
String doorState;
```
以下 timer 變數允許我們去抖動 switch。僅考慮它們之間至少間隔 1500 毫秒的更改。

```
unsigned long previousMillis = 0;
const long interval = 1500;
```
在以下變數中插入您的 SSID 和密碼，以便 ESP8266 可以連接到互聯網。

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```
插入您的 Telegram Bot Token——您在此步驟中獲得的令牌。
```
#define BOTtoken "XXXXXXXXXX:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```
插入您的聊天 ID，即您在此步驟中獲得的 ID。
```
#define CHAT_ID "XXXXXXXXXX"
```
使用創建新的 Wi-Fi 用戶端 `WiFiClientSecure`.
```
X509List cert(TELEGRAM_CERTIFICATE_ROOT);
WiFiClientSecure client;
```
使用之前定義的令牌和用戶端創建機器人。

```
UniversalTelegramBot bot(BOTtoken, client);
```
這變更門狀態（）函數將在 Door 狀態上檢測到更改時運行。此函數只是將更改狀態變數設置為真.然後，在 `loop（）` 的我們將處理 state 發生變化時會發生什麼（反轉之前的 door 狀態並向你的 Telegram 帳戶發送消息）。
```
ICACHE_RAM_ATTR void changeDoorStatus() {
  Serial.println("State changed");
  changeState = true;
}
```

設定 （）

在設定 （），初始化 Serial Monitor 以進行調試：
```
Serial.begin(115200);
```
為 api.telegram.org 添加根證書。

```
client.setTrustAnchors(&cert); // Add root certificate for
api.telegram.org
```
將簧片開關設置為輸入.並在 ESP8266 首次啟動時保存當前狀態。
```
pinMode(reedSwitch, INPUT_PULLUP);
state = digitalRead(reedSwitch);
```

將 LED 設置為輸出並設置其狀態以匹配簧片開關狀態（電路閉合且 LED 關閉;電路打開且 LED 亮起）。

```
pinMode(led, OUTPUT);
digitalWrite(led, state);
```

- 門關閉 – >ESP8266 讀取 HIGH 信號 – >關閉板載 LED（發送 HIGH 信號*）
- 門打開 – >ESP8266 讀取 LOW 信號 – >打開板載 LED（發送 LOW 信號*）

* ESP8266 板載 LED 與反相邏輯一起工作——發送 HIGH 信號將其關閉，發送 LOW 信號將其打開。

設置中斷

將簧片開關設置為中斷。

```
attachInterrupt(digitalPinToInterrupt(reedSwitch), changeDoorStatus, CHANGE);
```

要在 Arduino IDE 中設置中斷，請使用 `attachInterrupt()` 函數，它接受以下參數：GPIO 中斷引腳、要執行的函數的名稱和 mode。
第一個參數是 GPIO 中斷。您應該使用 `digitalPinToInterrupt (GPIO)` 將實際的 GPIO 設置為中斷引腳。
的第二個參數 `attachInterrupt()` function 是每次觸發中斷時將調用的函數的名稱 – 中斷服務程式 （ISR）。在本例中，它是更改門狀態功能。
ISR 函數應盡可能簡單，以便處理器快速返回主程式的執行。

第三個參數是 mode。我們將其設置為改變每當引腳更改值時觸發中斷 - 例如，從 HIGH 到 LOW，從 LOW 到 HIGH。
要瞭解有關 ESP8266 中斷的更多資訊，請閱讀以下教程：

- ESP8266 使用 Arduino IDE 的中斷和計時器

初始化 Wi-Fi

以下線路將 ESP8266 連接到 Wi-Fi。

```
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
```

```
}
Serial.println("");
Serial.println("WiFi connected");
```
向您的 Telegram 帳戶發送消息，通知您機器人已啟動。

```
bot.sendMessage(CHAT_ID, "Bot started up", "");
```

loop（）

在 `loop（）`，我們將讀取更改狀態變數，如果發生更改，我們將向您的 Telegram 帳戶發送消息。
首先，檢查是否發生了更改：

```
if (changeState){
```
然後，檢查自上次狀態更改以來是否至少過去了 1500 毫秒。

```
if(currentMillis - previousMillis >= interval) {
```
如果這是真的，請重置 timer 並反轉當前的 switch 狀態：

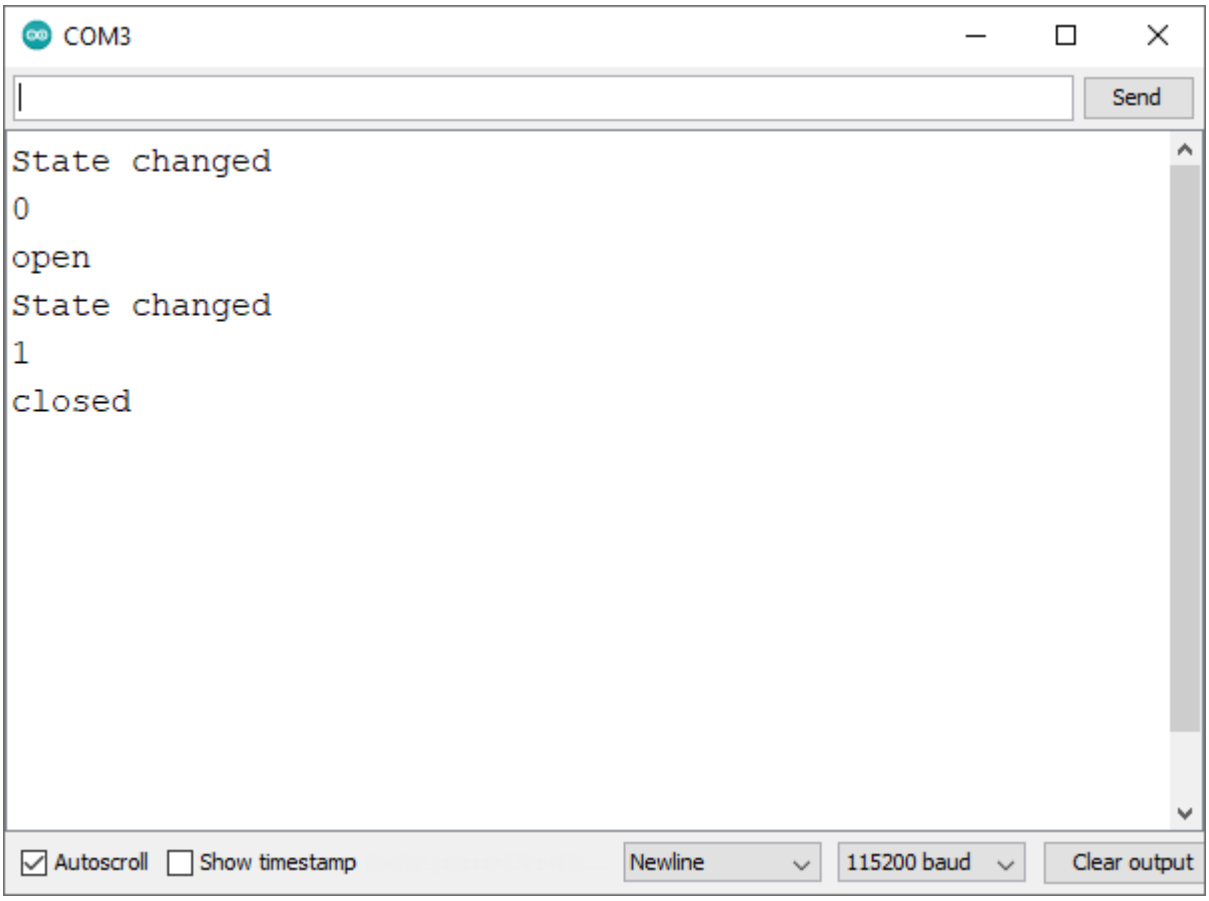
```
state = !state;
```
如果磁簧開關狀態為 `1`（true） 時，門已關閉。因此，我們將 `doorState` 變數設置為閉.
```
if(state) {
  doorState = "closed";
}
```
如果是 `0`（false） 時，門已打開。
```
else{
  doorState = "open";
}
```
相應地設置 LED 狀態，並在 Serial Monitor 中列印門狀態。

```
digitalWrite(led, state);
changeState = false;
Serial.println(state);
Serial.println(doorState);
```
最後，以下行向您的 Telegram 帳戶發送通知，其中包含當前 door 狀態。

```
bot.sendMessage(CHAT_ID, "The door is " + doorState, "");
```

示範

修改草圖以包含您的網路憑證、機器人令牌和使用者 ID 后，將其上傳到您的 ESP8266。轉到 **Tools > Board** 並選擇您的 ESP8266 板。然後，前往 **工具 > 埠** 並選擇 ESP8266 連接的 COM 埠。
以 115200 的波特率打開串行監視器，檢查是否檢測到更改。



對於原型設計/測試，您可以使用威扣將磁簧開關安裝到您的門上。

現在，當有人打開/關閉您的門時，您會在您的 Telegram 帳戶中收到一條消息。

結束語

在本教程中，您學習了如何在簧片開關更改狀態時向您的 Telegram 帳戶發送通知。這對於檢測門、窗或抽屜是打開還是關閉非常有用。