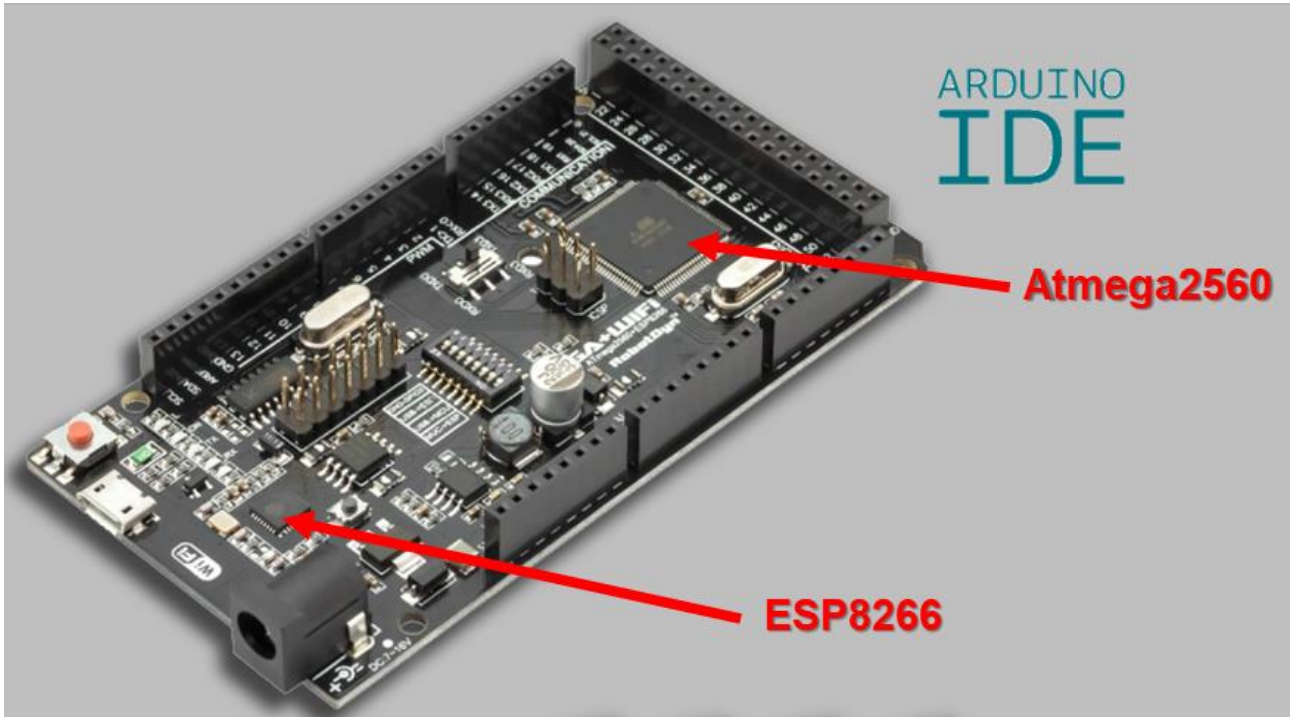
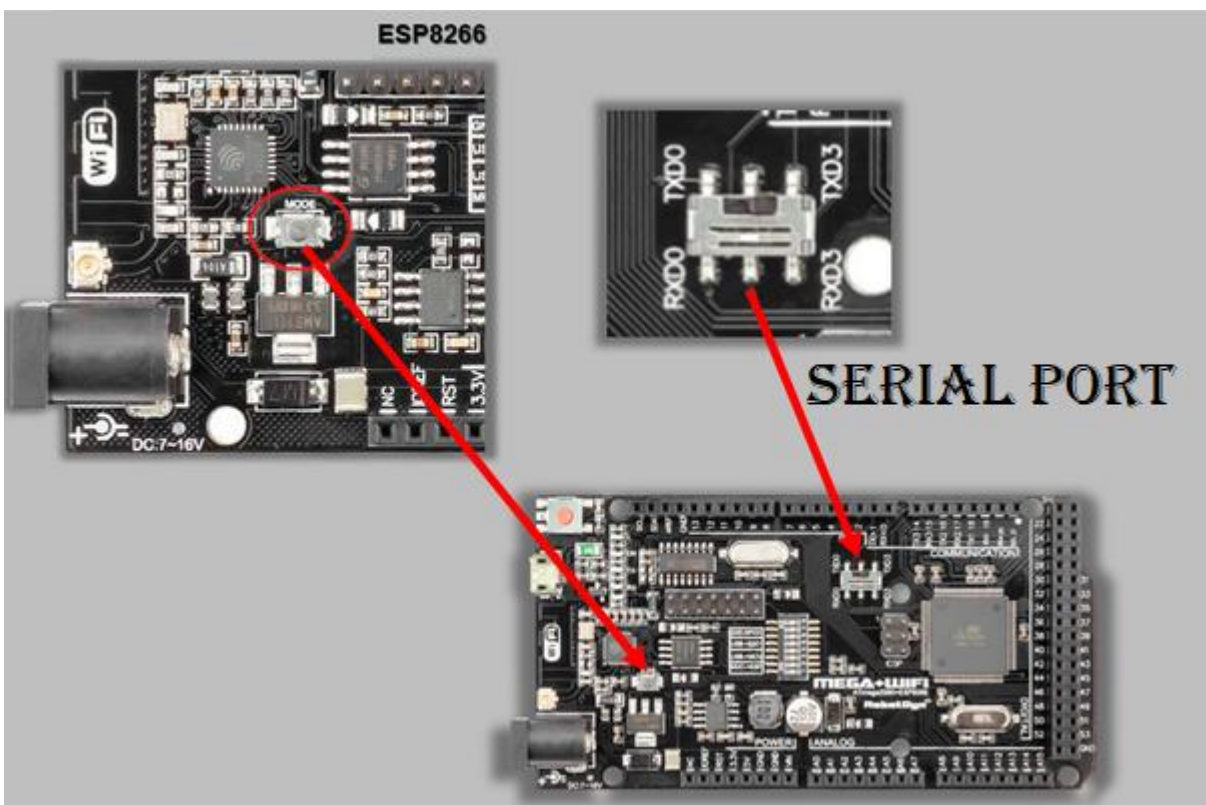


Arduino Mega 2560WiFi

Ide o špeciálnu verziu klasickej dosky ARDUINO MEGA R3. Plná integrácia mikrokontroléra Atmel ATmega2560 a Wi-Fi IC ESP8266 s 32 MB (megabitovou) flash pamäťou a prevodníkom CH340G USB-TTL na jednej doske. Všetky komponenty je možné nastaviť tak, aby spolupracovali alebo jednotlivo.

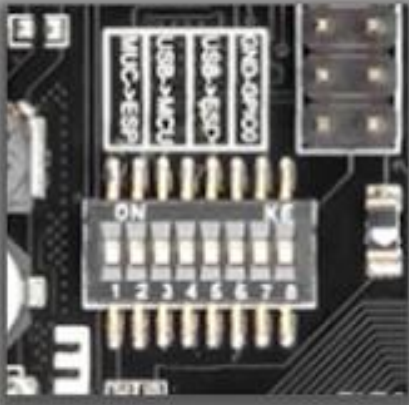


Na tejto doske sú prepínače, ktorými volíme režim dosky:



Popis režimov:

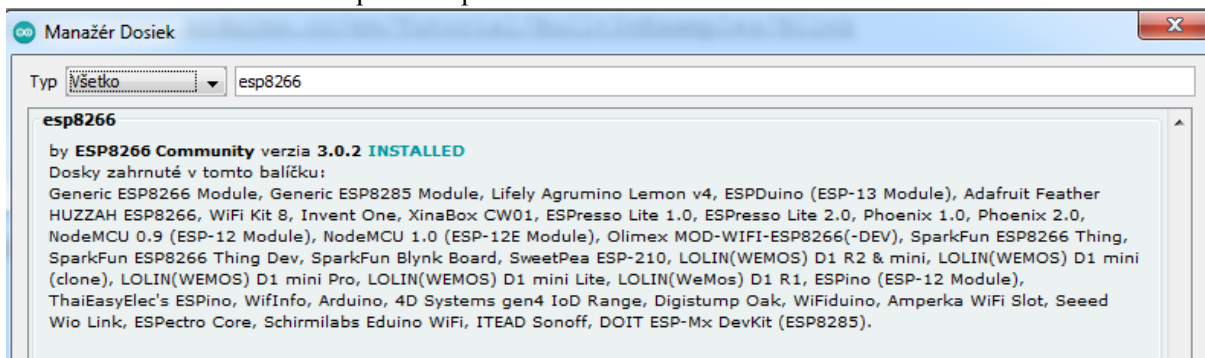
	1	2	3	4	5	6	7	8
CH340 connect to ESP8266 (upload sketch)	OFF	OFF	OFF	OFF	ON	ON	ON	NoUSE
CH340 connect to ESP8266 (connect)	OFF	OFF	OFF	OFF	ON	ON	OFF	NoUSE
CH340 connect to ATmega2560 (upload sketch)	OFF	OFF	ON	ON	OFF	OFF	OFF	NoUSE
CH340 connect to Mega2560 COM3 connect to ESP8266	ON	ON	ON	ON	OFF	OFF	OFF	NoUSE
Mega2560+ESP8266	ON	ON	OFF	OFF	OFF	OFF	OFF	NoUSE
All modules work independed	OFF	OFF	OFF	OFF	OFF	OFF	OFF	NoUSE



Treba zadať do **Súbor-Vlastnosti-Manažér prídavných dosiek URL:**

https://arduino.esp8266.com/stable/package_esp8266com_index.json

Do Manažéra dosiek musíme pridať esp8266



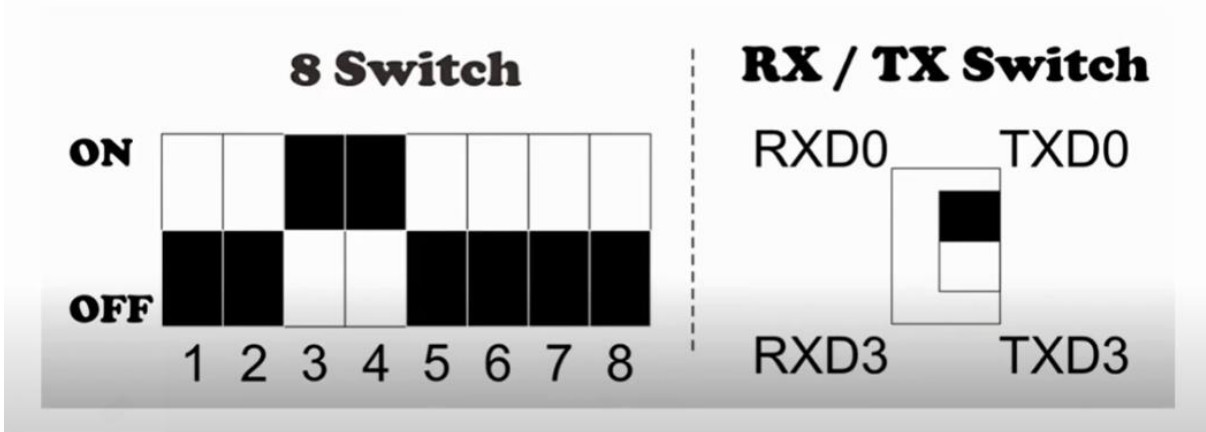
V Arduino IDE sa musí dať prepínať medzi doskou Arduino Mega a LOLIN(WEMOS) D1 R2&mini:



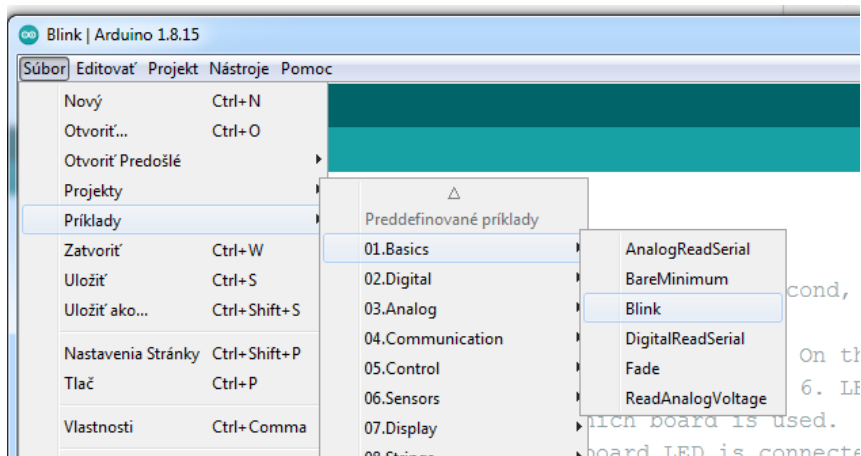
Po každej zmene módu (režimu) treba stlačiť tlačidlo **MODE**.

1. Režim „Len Arduino Mega“

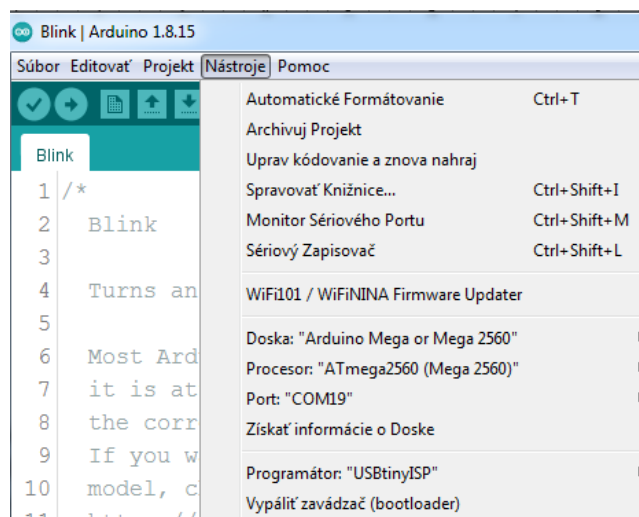
MODE SELECTION ARDUINO MEGA ONLY



Toto je mód pre nahrávanie programu do Arduino Mega 2560. Nastavíme prepínače, stlačíme tlačidlo **MODE**. Typ dosky máme **Arduino Mega**. Môžeme vyskúšať program Blink, ktorý nájdeme v **Súbor-Príklady**:

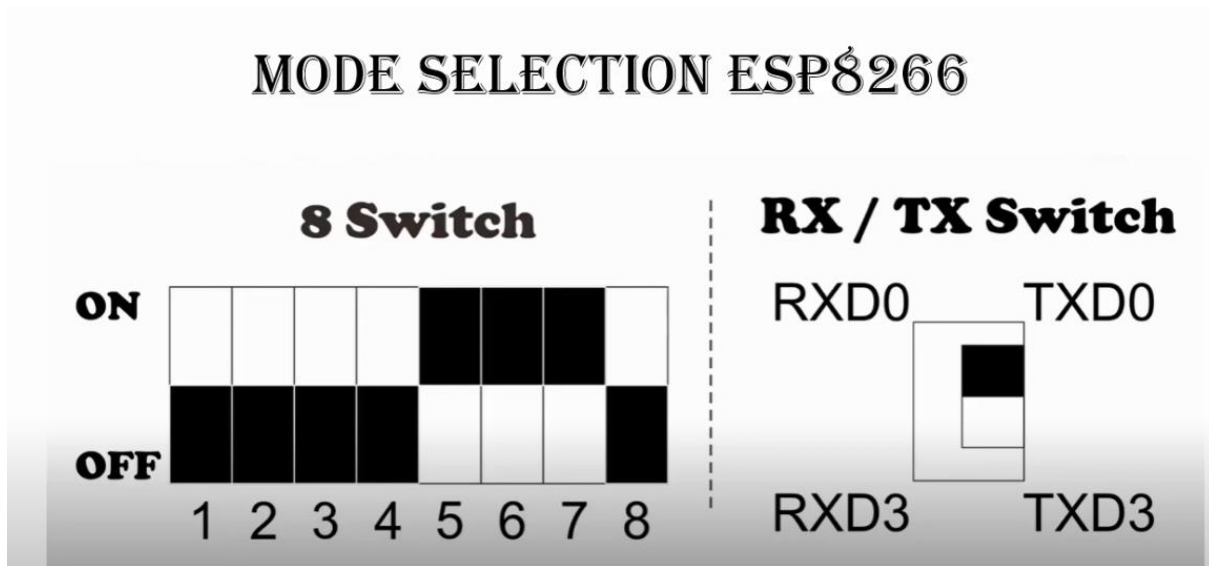


Pred nahratím programu do dosky nastavíme typ dosky a port:

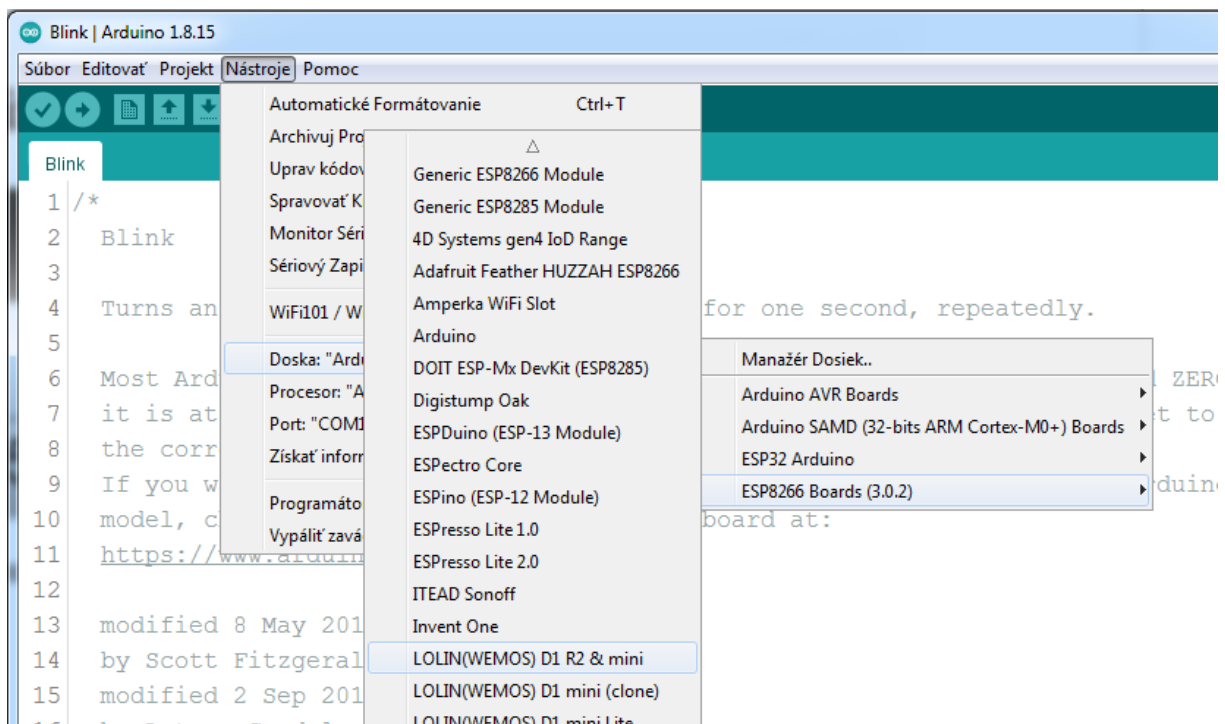


Program bliká internou LEDkou na doske.

2. Režim „ESP8266 nahratie programu“



Toto je mód na nahrávanie programu do ESP8266. Nastavíme prepínače, stlačíme tlačidlo **MODE**. Vyberieme typ dosky **LOLIN(WEMOS) D1 R2&mini**:



Do ESP8266 nahráme tento program - je odtiaľto:

<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw>

Program_blynk_ESP_Test

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "notebook"; //tu zadáme názov svojej siete!
const char* password = "xxxxxxx"; //tu zadáme heslo do svojej siete!
```

```
ESP8266WebServer server(80);
MDNSResponder mdns;

String webPage = "";

int led_pin = 13;

void setup(void){

    // подготовка:
    pinMode(led_pin, OUTPUT);
    digitalWrite(led_pin, LOW);
    Serial.begin(115200);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }

    // информация о контроллере
    Serial.println("");
    Serial.println("ESP8266 board info:");
    Serial.print("\tChip ID: ");
    Serial.println(ESP.getFlashChipId());
    Serial.print("\tCore Version: ");
    Serial.println(ESP.getCoreVersion());
    Serial.print("\tChip Real Size: ");
    Serial.println(ESP.getFlashChipRealSize());
    Serial.print("\tChip Flash Size: ");
    Serial.println(ESP.getFlashChipSize());
    Serial.print("\tChip Flash Speed: ");
    Serial.println(ESP.getFlashChipSpeed());
    Serial.print("\tChip Speed: ");
    Serial.println(ESP.getCpuFreqMHz());
    Serial.print("\tChip Mode: ");
    Serial.println(ESP.getFlashChipMode());
    Serial.print("\tSketch Size: ");
    Serial.println(ESP.getSketchSize());
    Serial.print("\tSketch Free Space: ");
    Serial.println(ESP.getFreeSketchSpace());

    // тело веб-страницы
    webPage += "<h1>ESP8266 Web Server</h1>";
    webPage += "<p>Chip ID: ";
    webPage += ESP.getFlashChipId();
    webPage += "</p>";
    webPage += "<p>Core Version: ";
    webPage += ESP.getCoreVersion();
    webPage += "</p>";
    webPage += "<p>Chip Real Size: ";
    webPage += ESP.getFlashChipRealSize()/1024;
    webPage += " Kbit</p>";
    webPage += "<p>Chip Size: ";
    webPage += ESP.getFlashChipSize()/1024;
    webPage += " Kbit</p>";
    webPage += "<p>Chip Flash Speed: ";
    webPage += ESP.getFlashChipSpeed()/1000000;
    webPage += " MHz</p>";
    webPage += "<p>Chip Work Speed: ";
    webPage += ESP.getCpuFreqMHz();
    webPage += " MHz</p>";
    webPage += "<p>Chip Mode: ";
    webPage += ESP.getFlashChipMode();
```

```
webPage += "</p>";
webPage += "<p>LED state <a href=\"LedON\"><button>ON</button></a>&nbsp;<a href=\"LedOFF\"><button>OFF</button></a></p>";

// подключение к WiFi
WiFi.begin(ssid, password);
Serial.println("");

// ожидание соединения:
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.print(".");
}
Serial.println("");
Serial.print("Connected to "); //
Serial.println(ssid);
Serial.print("IP address: "); //
Serial.println(WiFi.localIP());

// Проверка запуска MDNS
if (mdns.begin("esp8266", WiFi.localIP())) {
  Serial.println("MDNS responder started");
}

server.on("/", [](){
  server.send(200, "text/html", webPage);
});

server.on("/LedON", [](){
  server.send(200, "text/html", webPage);
  digitalWrite(led_pin, HIGH);
  Serial.println("[ON]");
  delay(1000);
});

server.on("/LedOFF", [](){
  server.send(200, "text/html", webPage);
  digitalWrite(led_pin, LOW);
  Serial.println("[OFF]");
  delay(1000);
});

server.begin();
Serial.println("HTTP server started");
}

void loop(void){
  server.handleClient();
}
```



Po nahratí to vyzerá takto:

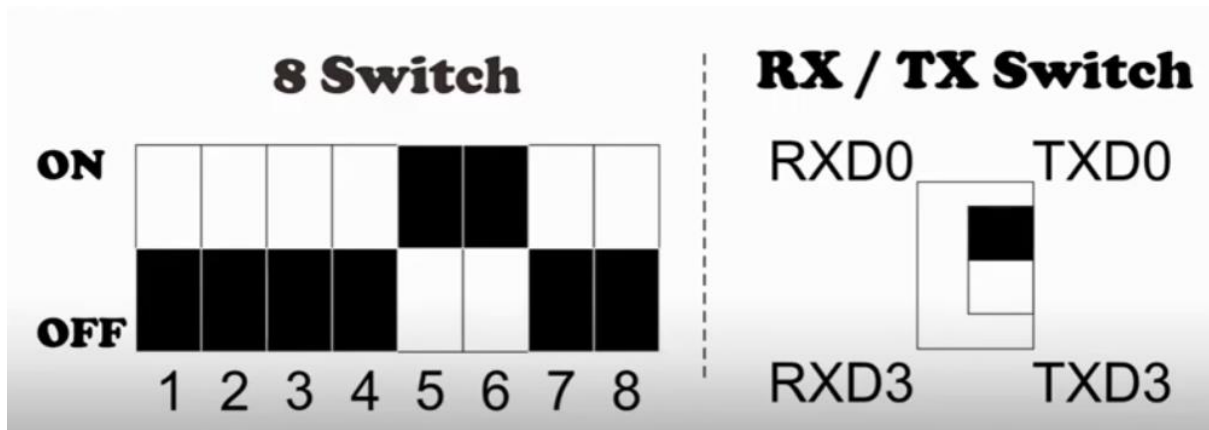
```

Program_blynk_ESP_Test | Arduino 1.8.15
Súbor Editovať Projekt Nástroje Pomoc
Program_blynk_ESP_Test
1 #include <ESP8266WiFi.h>
2 #include <WiFiClient.h>
3 #include <ESP8266WebServer.h>
4 #include <ESP8266mDNS.h>
5
6 const char* ssid = "notebook";
7 const char* password = " ";
8
9 ESP8266WebServer server(80);
10 MDNSResponder mdns;
11
12 String webPage = "";
13
Nahrávanie ukončené.
Writing at 0x00034000... (93 %)
Writing at 0x00038000... (100 %)
Wrote 325808 bytes (234669 compressed) at 0x00000000 in 5.5 seconds (effective 478.0 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
+ 32KB cache + 32KB IRAM (balanced). Use pgm_read macros for IRAM/PROGMEM. 4MB (FS:2MB OTA~1019KB), v2 Lower Memory, Disabled, None, Only Sketch, 921600 na COM19

```

Po nahratí programu zmeníme mód.

3. Režim „ESP8266 spustenie programu“



Toto je mód pre spustenie programu v ESP8266. Nastavíme prepínače, stlačíme tlačidlo **MODE**. Typ dosky máme ešte **LOLIN(WEMOS) D1 R2&mini**. Otvoríme sériový monitor na 115200 baud. Stlačíme RESET tlačidlo na Arduino Mega doske.

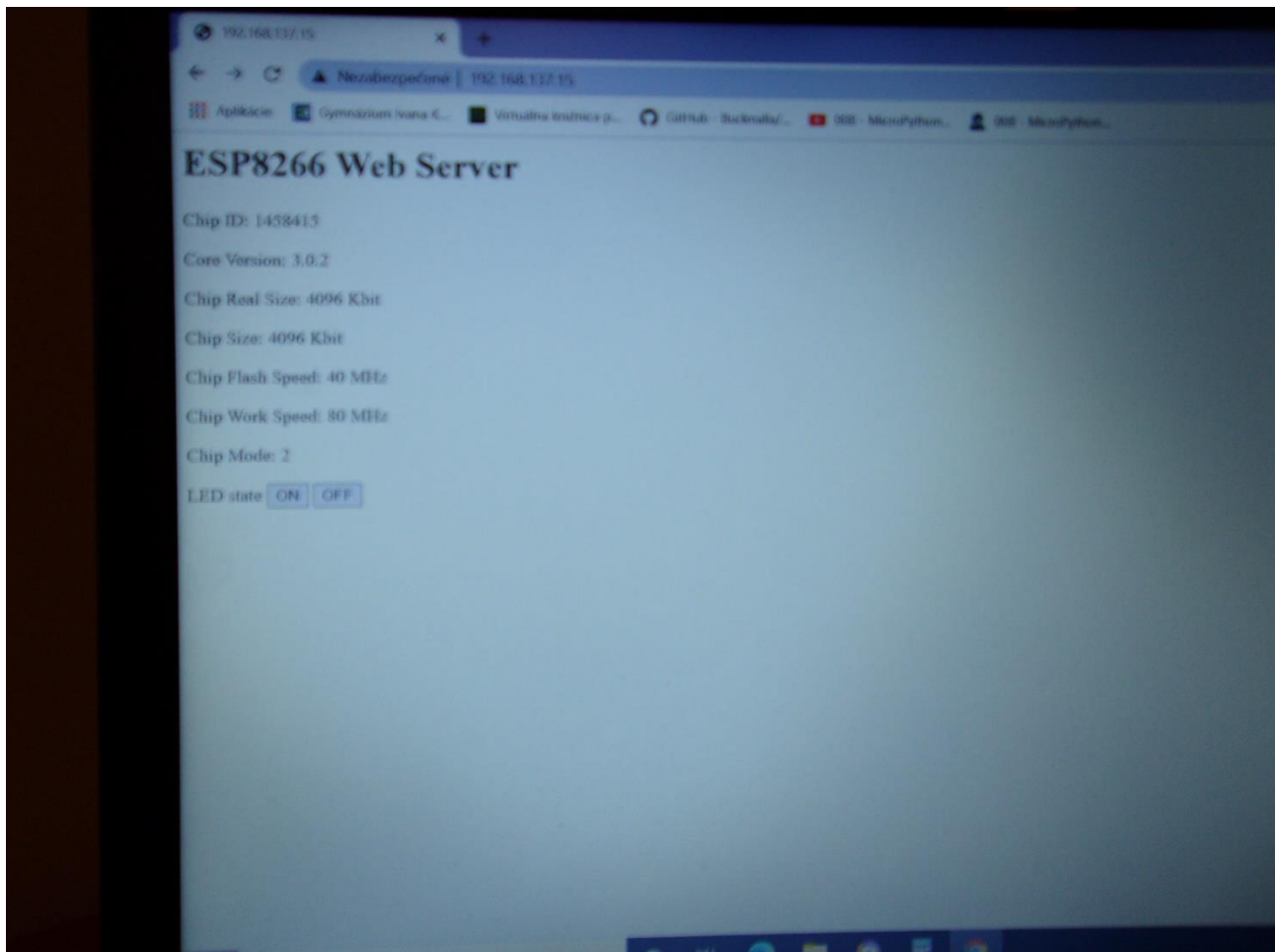


Na sériovom monitore sa vypíše:

```
COM19
ESP8266 board info:
  Chip ID: 1458415
  Core Version: 3.0.2
  Chip Real Size: 4194304
  Chip Flash Size: 4194304
  Chip Flash Speed: 40000000
  Chip Speed: 80
  Chip Mode: 2
  Sketch Size: 325808
  Sketch Free Space: 1769472

....
Connected to notebook
IP address: 192.168.137.15
MDNS responder started
HTTP server started
```

Názov mojej siete, čiže moje SSID je notebook. Dôležitá je IP adresa. Tú IP adresu zadáme do prehliadača na počítači, ktorý je v tej istej sieti.



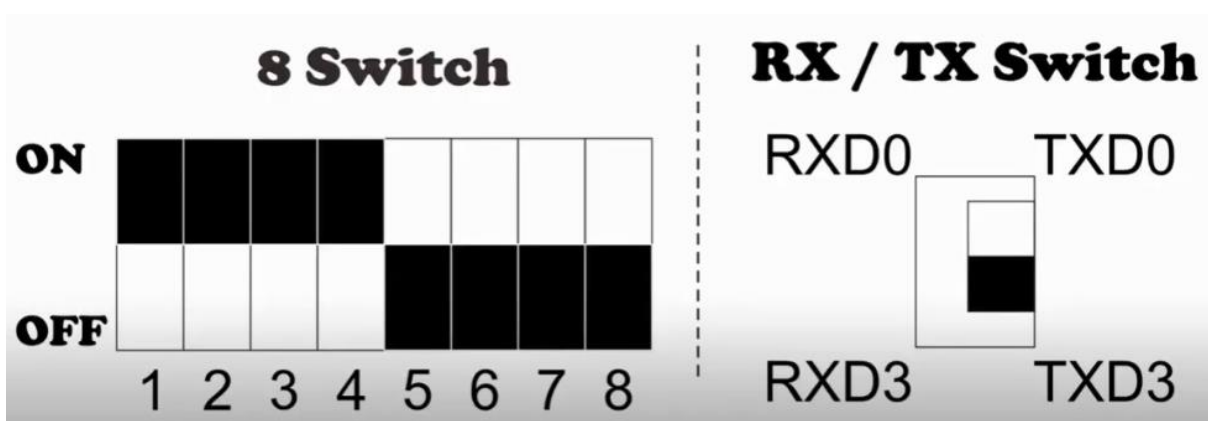
Môžeme stláčať tlačidlá **ON** a **OFF** na tomto webserveri v prehliadači a v sériovom monitore uvidíme, ako sme ich stlačili:


```

COM19
[1 100] [0d0] [0 d0 b<00 [0 s0c0 #00no0$gm000 c p00ds$sdp0g0 [0 1 00 c n0| [0 00 c00o'0 100
ESP8266 board info:
  Chip ID: 1458415
  Core Version: 3.0.2
  Chip Real Size: 4194304
  Chip Flash Size: 4194304
  Chip Flash Speed: 40000000
  Chip Speed: 80
  Chip Mode: 2
  Sketch Size: 325808
  Sketch Free Space: 1769472

....
Connected to notebook
IP address: 192.168.137.15
MDNS responder started
HTTP server started
[OFF]
[ON]
[ON]
[OFF]
[OFF]
[ON]
  
```

4. Režim „Mega2560 aj ESP8266 pripojené“



Toto je mód na pripojenie k MEGA 2560 aj k ESP 8266. Nastavíme prepínače (nezabudnúť aj prepínač RXR/TXD), stlačíme tlačidlo **MODE**. Typ dosky nastavíme na **Arduino Mega**. Nahráme tento program - je odtiaľto:

<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw/folder/3uBXUKKY>

Program_blynk_MEGA_Test

```

#include <MemoryFree.h>
#include <EEPROM.h>
  
```

```
#define PIN_LED 13 // led on pin board
String inString;

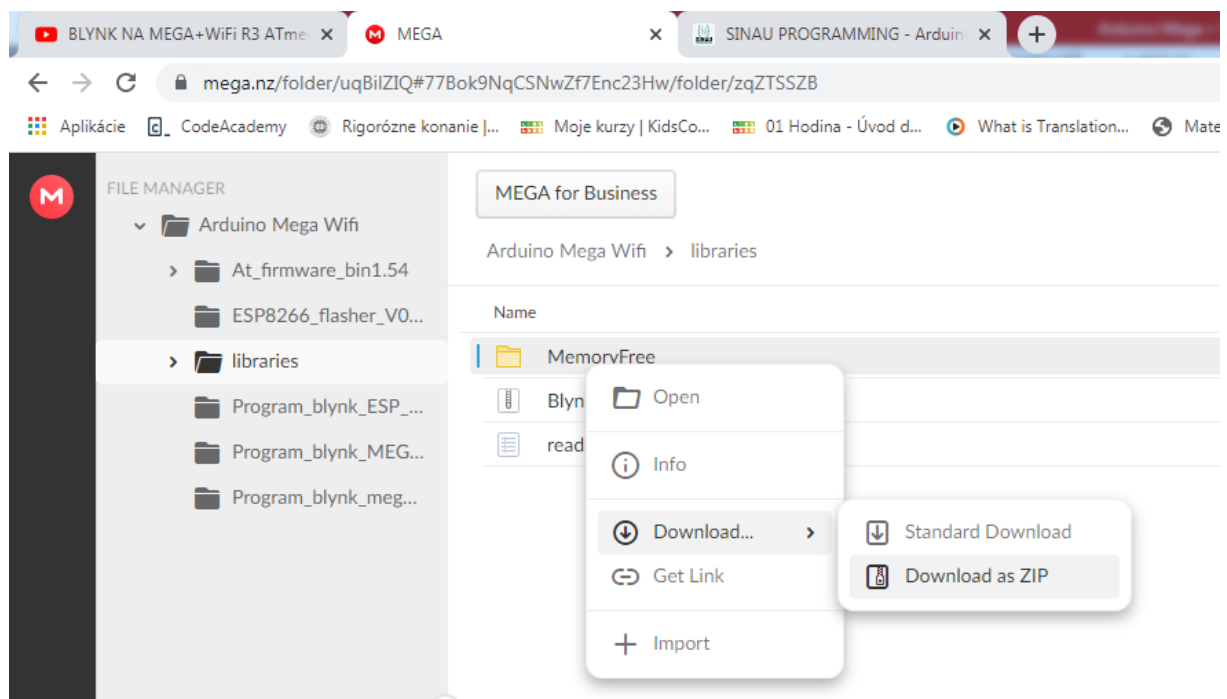
void setup() {
  Serial.begin(115200);
  Serial3.begin(115200);
  pinMode(PIN_LED, OUTPUT);
  digitalWrite(PIN_LED, LOW);
}

void loop() {
}

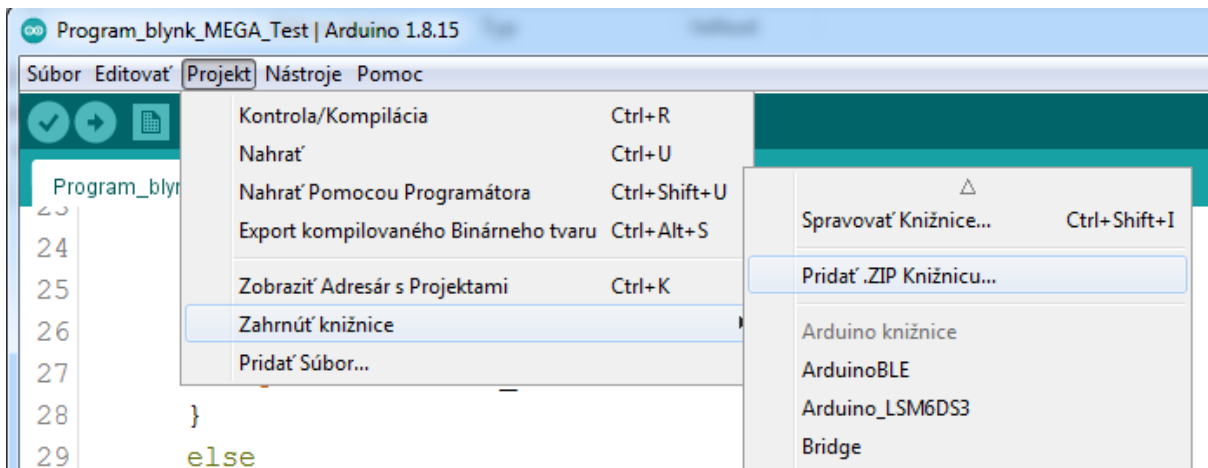
void serialEvent3() {
  while (Serial3.available()) {
    char inChar = Serial3.read();
    Serial.write(inChar);
    inString += inChar;
    if (inChar == ']') {
      if (inString.indexOf("[ON]")>0) {
        digitalWrite(PIN_LED, HIGH);
      }
      else if (inString.indexOf("[OFF]")>0) {
        digitalWrite(PIN_LED, LOW);
      }
      else
      {
        Serial.println("Wrong command");
      }
      inString = "";
    }
  }
}
```

Program potrebuje knižnicu **Memory Free**, stiahneme ju ako ZIP súbor odiaľto:

<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw/folder/zqZTSSZB>



Po stiahnutí ZIP súboru pridáme knižnicu do Arduino IDE takto:



Zavrieme sériový monitor (ak zostal otvorený), skompilujeme program. Ak kompilácia prebehla bez chýb,

nahráme program do Arduina



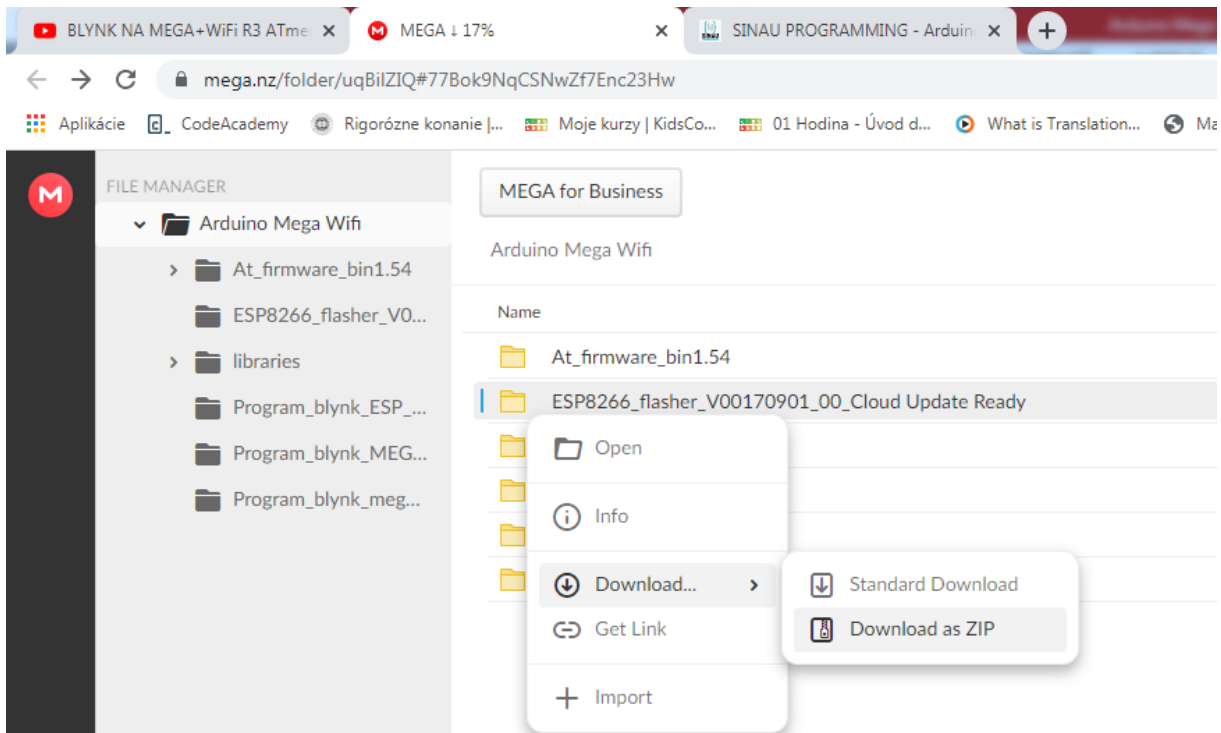
Na počítači, v ktorom máme otvorený webserver, opäť vyskúšame funkciu zobrazených tlačidiel, mali by zažínať a zhasínať LED na doske Arduino Mega.

5. Nahratie firmvéru do ESP8266



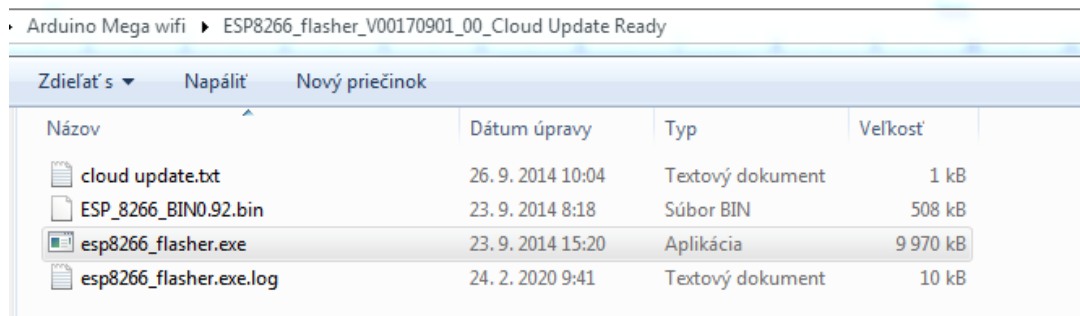
Toto je mód na nahrávanie programu do ESP8266. Nastavíme prepínače (nezabudnúť na RXD/TXD), stlačíme tlačidlo **MODE**. Na nahratie firmvéru do ESP8266 potrebujeme program **ESP 8266 flash Downloader**. Stiahneme si ho ako ZIP odtiaľto:

<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw>

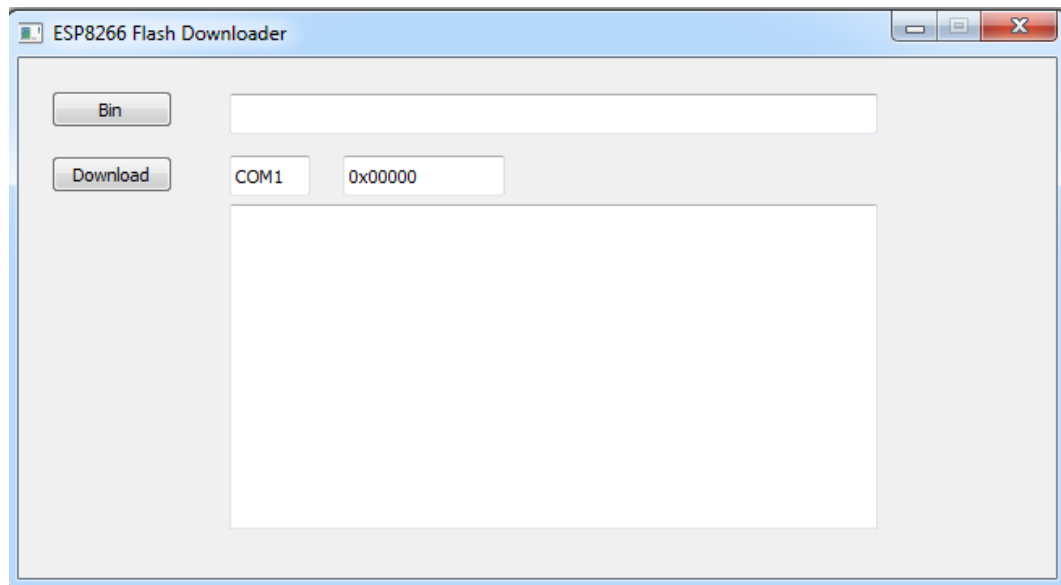


Po stiahnutí si ho premiestnime do požadovaného priečinku, kde ho rozbalíme.

Spustíme program **ESP 8266 flash Downloader**:

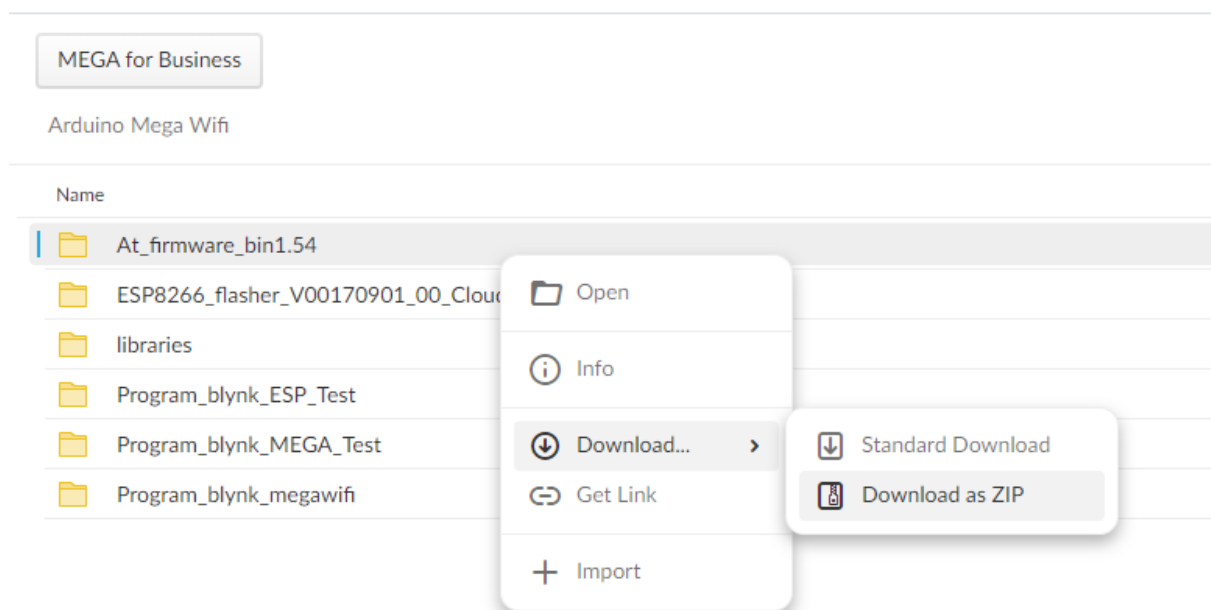


Takto vyzerá po spustení:

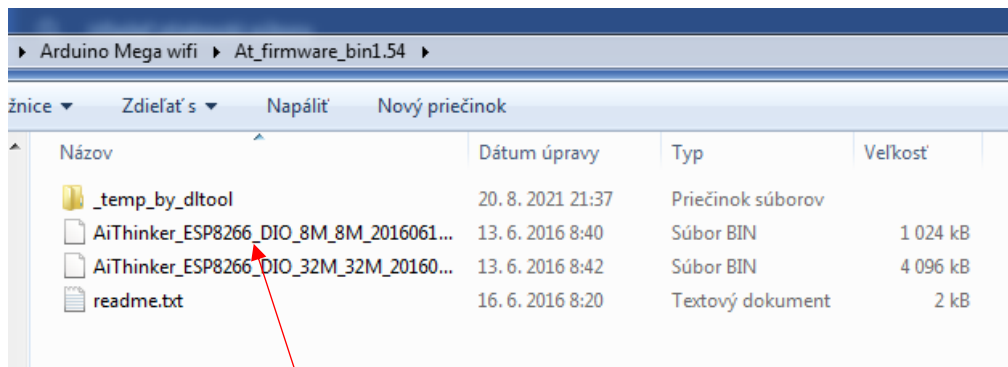


Týmto programom budeme napáľovať firmvér. Ten si stiahneme odtiaľto:

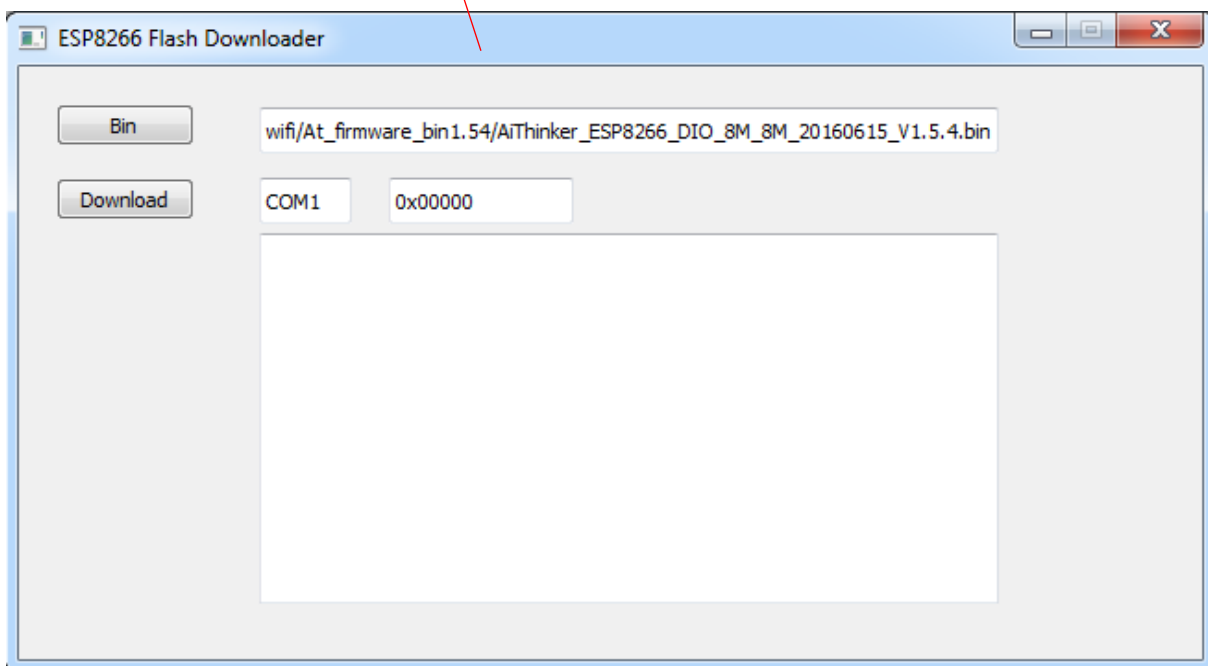
<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw>



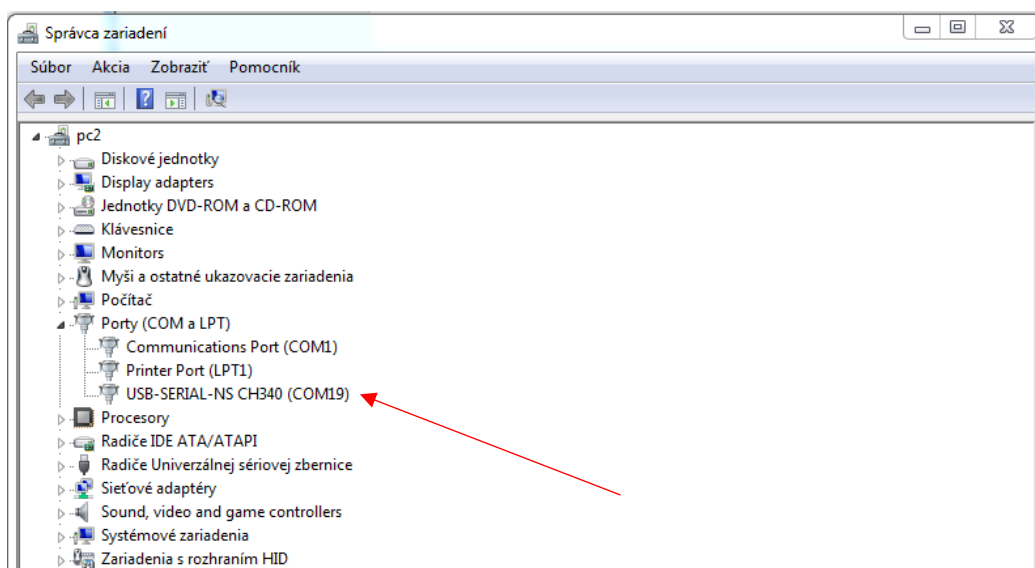
Po stiahnutí si ho premiestnime do požadovaného priečinku, kde ho rozbalíme. Obsahuje tieto súbory:



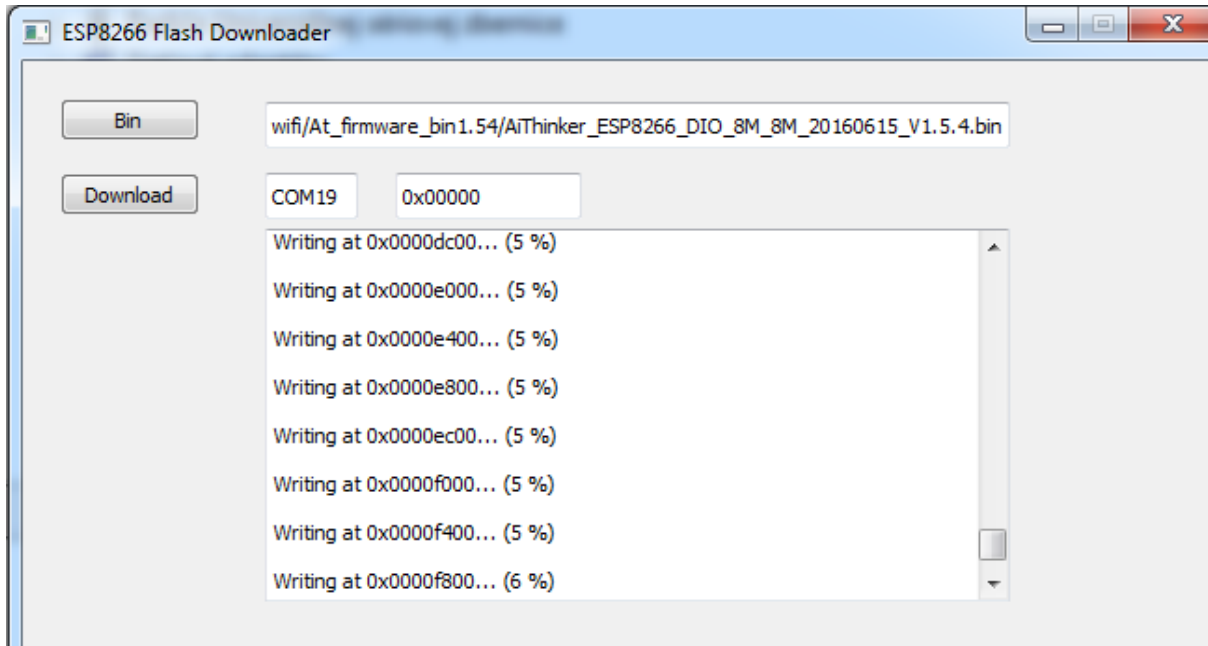
Vrátime sa do programu ESP8266 Flash Downloader. Klikneme na tlačidlo BIN a pohladáme súbor, ktorý sme si pred chvíľou nachystali.



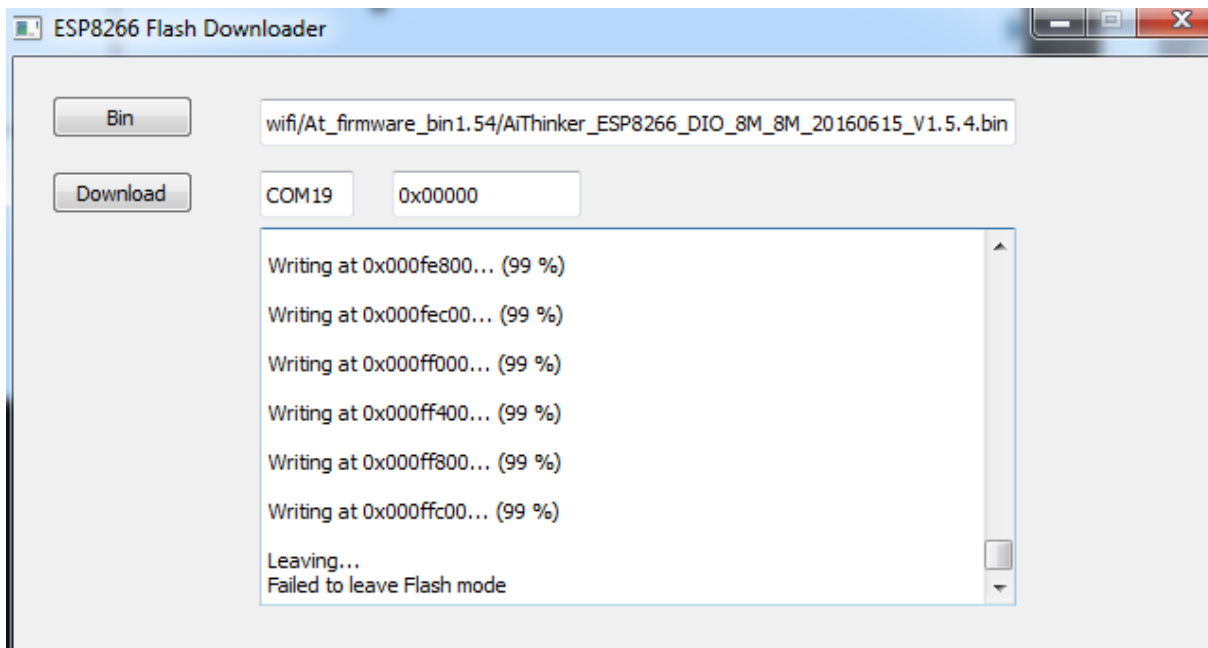
V Správcovi zariadení si pozrieme, na ktorý port je naša doska pripojená:



U mňa je to COM19. Nastavíme tento port v programe ESP8266 Flash Downloader. Potom stlačíme tlačidlo **Download**. Začne sa zapisovanie firmvéru do ESp 8266:

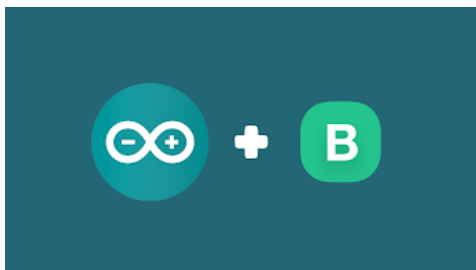


Keď skončí s napáľovaním firmvéru, vypíše:



6. Aplikácia BLYNK

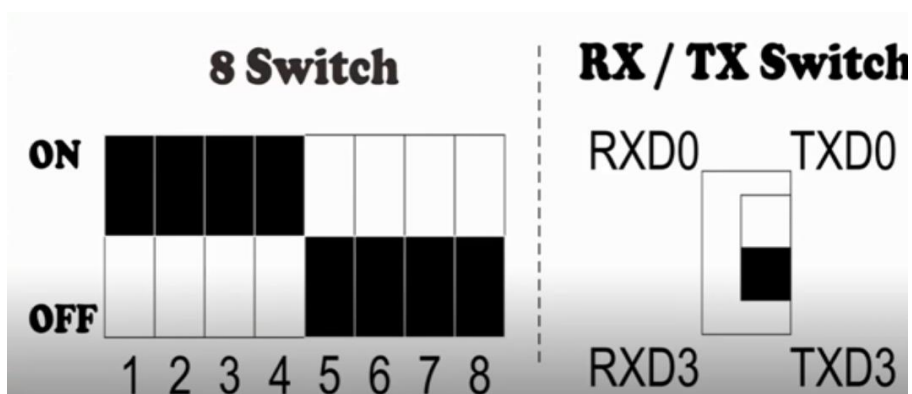
BLYNK je platforma pre aplikácie pre mobilné operačné systémy (iOS a Android), ktorej cieľom je ovládať moduly **Arduino**, **Raspberry Pi**, **ESP8266**, **WEMOS D1** a podobné moduly prostredníctvom internetu. Z tejto aplikačnej platformy môžete ovládať čokoľvek na diaľku, nech ste kdekoľvek a kedykoľvek. So záznamom pripojeným k internetu so stabilným pripojením a tomuto sa hovorí systém internetu vecí (**IoT**).



Postup na nainštalovanie a používanie tejto aplikácie je tu:

<https://www.sinauprogramming.com/2020/10/kontrol-led-dengan-menggunakan-blynk.html>

Našu dosku Arduino Mega2560 WiFi prepeneme do tohto módu:



Je to mód pre pripojenie k Arduino Mega aj k ESP 8266. Po prepnutí stlačíme tlačidlo MODE.

Program **Program_blynk:megawifi** si stiahneme si odtiaľto

<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw/folder/nuljRCCD>

The screenshot shows a web browser window with the Mega.nz file manager interface. The file 'Program_blynk_megawifi.ino' is selected, and a context menu is open over it. The menu options are: Info, Download... (with a sub-menu), Get Link, and Import. The sub-menu for 'Download...' is open, showing 'Standard Download' and 'Download as ZIP'.

Program_blynk:megawifi

```
// LIBRARY-Blynk
//=====
#define BLYNK_PRINT Serial
#include <ESP8266_Lib.h> // insert this library
#include <BlynkSimpleShieldEsp8266.h>
//=====

// Initialitiation
//=====
char auth[] = "2626aXgFo0b3_bDAQURC_FE660sAG-8A";
char ssid[] = "názov siete";
char pass[] = "heslo do siete";
#define EspSerial Serial3
#define ESP8266_BAUD 115200
ESP8266 wifi(&EspSerial);

//=====
// write data from blynk on virtual 2
BLYNK_WRITE(V2) {
  if (param.asInt()) {
    digitalWrite(13, HIGH);
    //turn led on arduino mega wifi
  }
  else {
    digitalWrite(13, LOW);
    //turn led off arduino mega wifi
  }
}
//=====

void setup()
{
  pinMode(13, OUTPUT);
  Serial.begin(115200);
  Serial3.begin(115200);

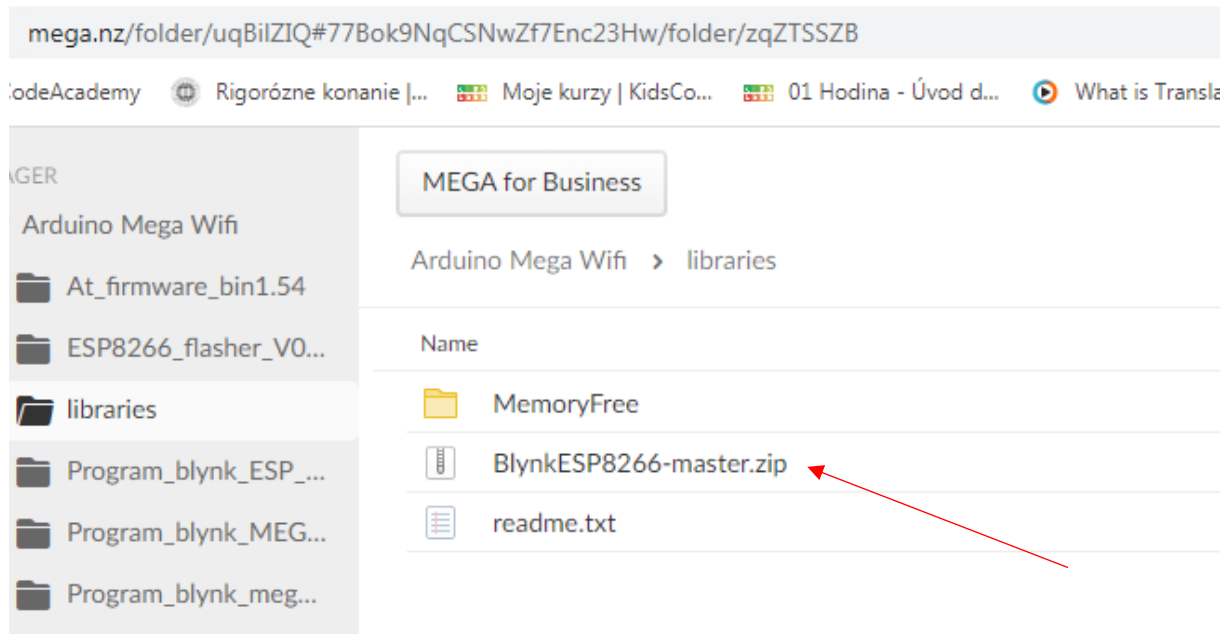
  delay(10);
  EspSerial.begin(ESP8266_BAUD);
  delay(10);

  // Blynk.begin(auth, wifi, ssid, pass); //Reguler server
  Blynk.begin(auth, wifi, ssid, pass,"blynk-cloud.com", 8080); //Local server
}

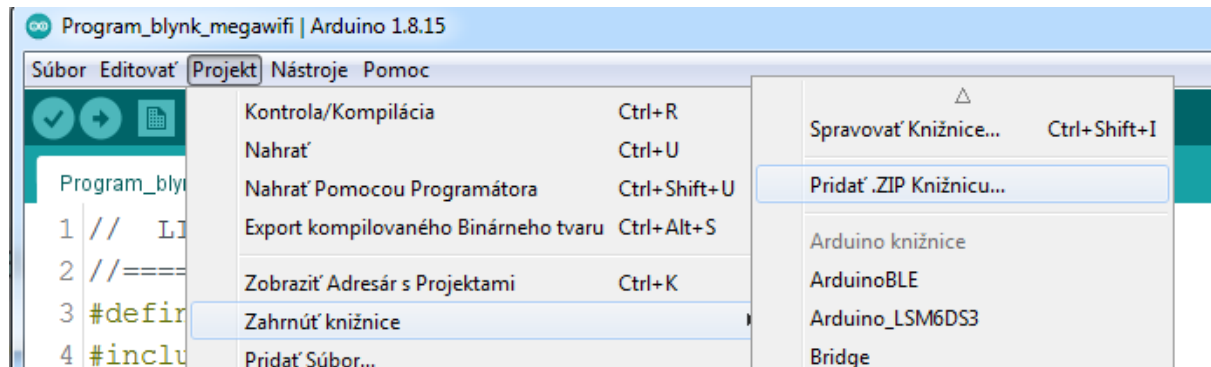
void loop()
{
  Blynk.run();
  if ( Serial3.available() ) {
    Serial.write( Serial3.read() );
  }
  if ( Serial.available() ) {
    Serial3.write( Serial.read() );
  }
}
```

Nezabudneme si v tom programe nastaviť svoju sieť a heslo! Vyskúšame skompilovať, ak mu chýba knižnica **ESP8266_Lib**, tak ju doinštalujeme odiaľto:

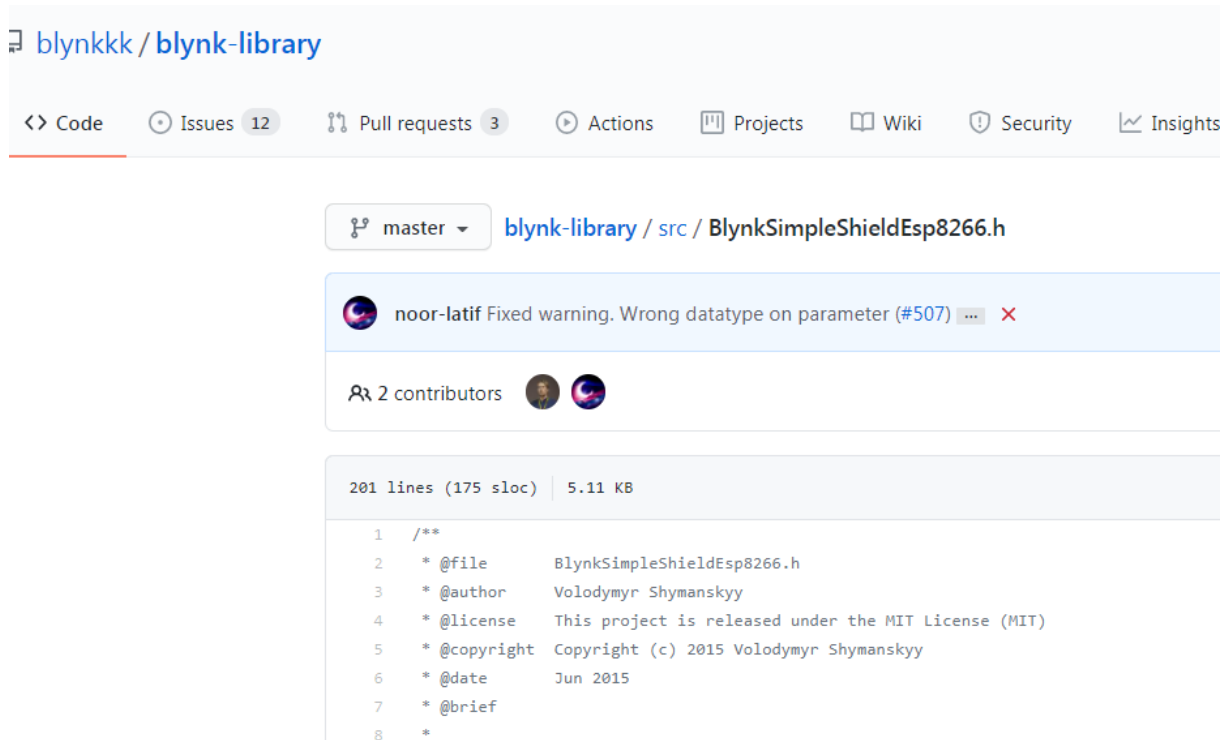
<https://mega.nz/folder/uqBilZIQ#77Bok9NqCSNwZf7Enc23Hw/folder/zqZTSSZB>



Po stiahnutí ju pridáme do Arduino IDE.



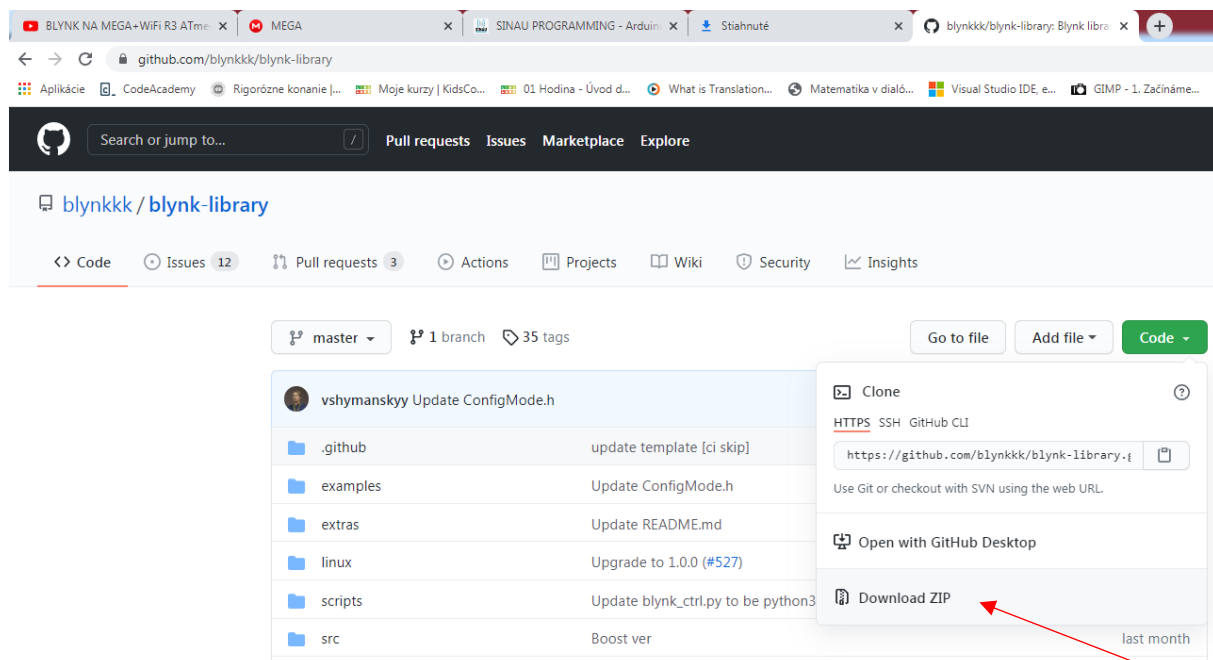
Taktiež bude potrebná knižnica BlynkSimpleShieldEsp8266.h, ktorú stiahneme odtiaľto:
<https://github.com/blynkkk/blynk-library>



The screenshot shows the GitHub repository page for `blynkkk/blynk-library`. The file `BlynkSimpleShieldEsp8266.h` is selected. The repository has 12 issues, 3 pull requests, and 2 contributors. The file content is as follows:

```
1 /**
2  * @file      BlynkSimpleShieldEsp8266.h
3  * @author    Volodymyr Shymanskyy
4  * @license   This project is released under the MIT License (MIT)
5  * @copyright Copyright (c) 2015 Volodymyr Shymanskyy
6  * @date      Jun 2015
7  * @brief
8  *
```

Stiahneme ju ako ZIP súbor:



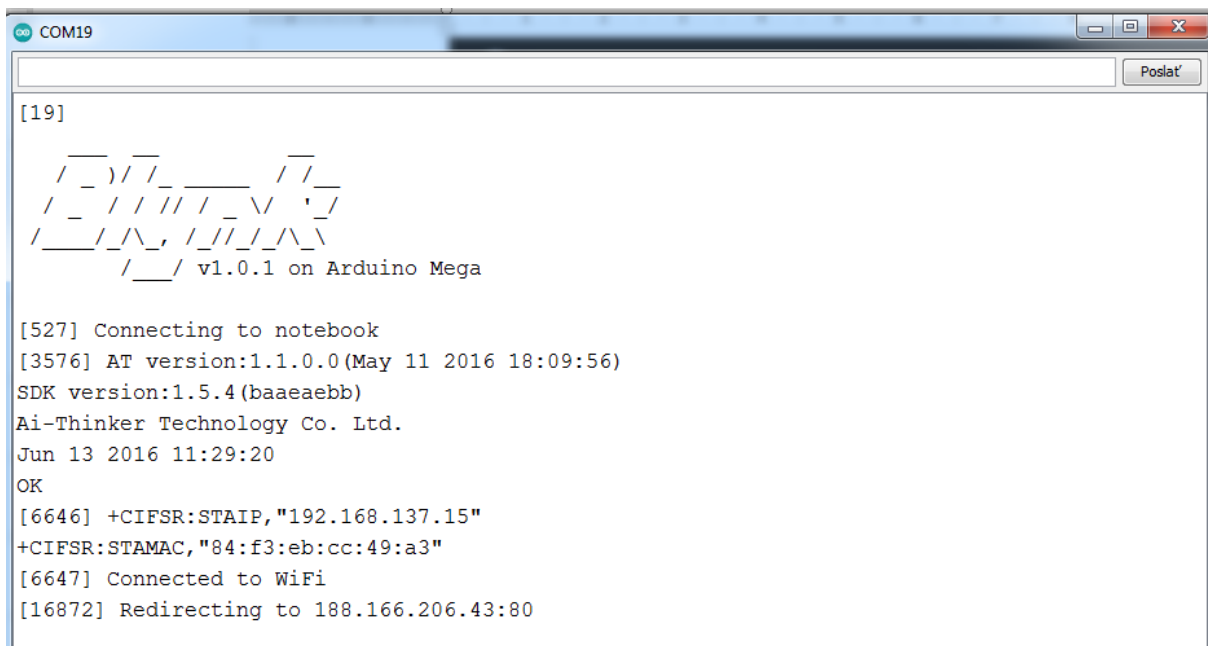
The screenshot shows the GitHub repository page for `blynkkk/blynk-library`. The file `BlynkSimpleShieldEsp8266.h` is selected. The repository has 12 issues, 3 pull requests, and 1 branch. The file content is as follows:

```
1 /**
2  * @file      BlynkSimpleShieldEsp8266.h
3  * @author    Volodymyr Shymanskyy
4  * @license   This project is released under the MIT License (MIT)
5  * @copyright Copyright (c) 2015 Volodymyr Shymanskyy
6  * @date      Jun 2015
7  * @brief
8  *
```

The 'Code' dropdown menu is open, and the 'Download ZIP' option is highlighted with a red arrow.

Aj tú stiahneme ako ZIP a pridáme do Arduino IDE. Potom už kompilácia prebehne v poriadku a môžeme program nahráť do Arduino Mega.

V sériovom monitore uvidíme:



```
COM19
[19]
  / _ ) / / _ _ _ _ / / _
 / _ / / / / / _ \ ' /
 / _ _ / \ _ / / / / \ \
   / _ / v1.0.1 on Arduino Mega

[527] Connecting to notebook
[3576] AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
Ai-Thinker Technology Co. Ltd.
Jun 13 2016 11:29:20
OK
[6646] +CIFSR:STAIP,"192.168.137.15"
+CIFSR:STAMAC,"84:f3:eb:cc:49:a3"
[6647] Connected to WiFi
[16872] Redirecting to 188.166.206.43:80
```

Teraz sa dá LED ovládať cez aplikáciu **Blynk**, nie veľmi podrobný postup je na <https://www.sinauprogramming.com/2020/10/kontrol-led-dengan-menggunakan-blynk.html>