

Laborator 8. Arduino și aplicații WiFi - IoT

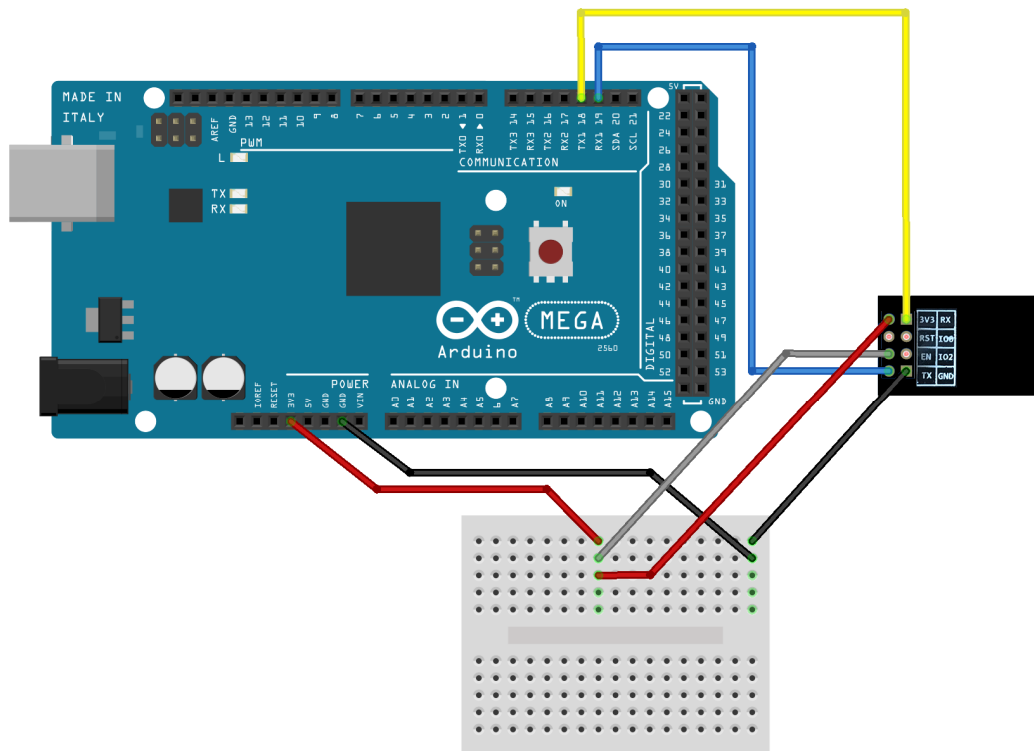
IoT - Internet of Things se referă la conectarea diferitelor dispozitivelor electronice inteligente (inclusiv microcontrollere, senzori) la Internet. În această lucrare de laborator vom utiliza Arduino împreună cu un modul ESP8266 WiFi module. Cu acest dispozitiv, vom controla, dintr-o pagină web găzduită pe Arduino, starea unui LED de pe placă, și vom vizualiza diferite informații.

ESP 8266 este un modul autonom WiFi care poate fi programat folosind mediul de dezvoltare Arduino. Programarea acestui dispozitiv poate fi realizată folosind un programator FTDI, sau o placă Arduino cu chipul ATmega328 detașat. **În acest laborator, vom configura esp8266 prin interfața lui serială, folosind comenzi AT.**

Conectarea modului WiFi

Vom conecta modulul ESP la Arduino folosind interfața UART Serial1. Conectați pinul RX al ESP8266 la pinul **18 (TX1)** Arduino, și pinul TX al ESP8266 la pinul **19 (RX1)**. Conectați pinul GND la un pin GND al Arduino, și pinul VCC/3V3 la pinul Arduino de **3.3 V**. Conectați și pinul EN al ESP8266 tot la **3.3 V**.

Consultați schema de mai jos pentru realizarea conexiunilor:



fritzing

Programul Arduino trebuie să genereze comenzi AT pentru resetarea modului WiFi (“AT+RST”). Următorul pas este configurarea acestui modul ca punct de acces WiFi (“AT+CWMODE=2”). După acest pas, se citește adresa IP a modului, 192.168.4.1, folosind comanda: “AT+CIFSR”, care va tipări și adresa MAC.

Pentru a obține informația de conectare (SSID), vom rula comanda (“AT+CWSAP?”): se va returna numele rețelei și parola (implicit nu există parolă), și apoi putem configura sistemul să accepte conexiuni multiple (“AT+CIPMUX=1”) și vom porni serverul web pe portul 80 (“AT+CIPSERVER=1,80”).

Fiecare comandă AT trebuie terminată cu *carriage return* și *newline* (“\n”).

Mai multe informații privind comenzile AT ale ESP8266 pot fi găsite aici: https://www.espressif.com/sites/default/files/documentation/4a-esp8266_at_instruction_set_en.pdf și aici: https://github.com/espressif/esp8266_at/wiki.

Comenzile sunt trimise folosind interfața serială inițializată din programul Arduino (“**Serial1**”), folosind metoda “*print*”. Răspunsul modulului la comandă este citit și salvat într-un string, și apoi este afișat pe serial monitor, folosind conexiunea dintre Arduino și PC, “**Serial**”. Consultați funcția “**sendData()**” din exemplul de mai jos.

În bucla principală verificăm dacă datele sunt disponibile pe interfața Serial1, și verificăm dacă sunt date de pe rețea (vor include substring-ul “+IPD”). Prima dată trebuie să citim identificatorul conexiunii, deoarece acesta este necesar când transmitem date folosind comanda: “AT+CIPSEND”.

O pagină web este construită sub forma unui string și trimisă la modulul ESP8266. Pagina include text de afișat, două butoane pentru preluarea comenzilor de la utilizator, și apoi alt text de afișat, ce include date generate de Arduino. După ce se transmite comanda AT pentru transmiterea paginii web, se încheie conexiunea folosind: “AT+CIPCLOSE”.

Mecanismul pentru controlul led-ului pe Arduino se bazează pe folosirea a două butoane în pagina web, fiecare indicând un URL diferit. Primul buton va avea atașat URL-ul “/I0”, iar al doilea “/I1”. Prin apăsarea acestor butoane, pagina web încearcă să redirecteze către aceste adrese, și va genera o cerere către server. Pe Arduino, vom primi această cerere citind datele transmise de ESP8266 (în funcția “*sendData()*”) și vom verifica dacă conține substring-urile “/I0” sau “/I1” (ex: “*response.indexOf("/I0") != -1*”).

Datele de la Arduino sunt afișate pe pagina web adăugând rezultatul funcției “*readSensor()*” la string-ul paginii web, și trimiterea lui folosind comenzi AT către modulul WiFi. În exemplul de mai jos, funcția va afișa de fapt rezultatul apelării funcției *millis()*.

ESP 8266 cod exemplu:

```
#define DEBUG true

void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);
  sendData("AT+RST\r\n", 2000, false); // resetare modul
  sendData("AT+CWMODE=2\r\n", 1000, false); // configurare ca access point
  sendData("AT+CIFSR\r\n", 1000, DEBUG); // citește adresa IP
```

```

sendData("AT+CWSAP?\r\n", 2000, DEBUG); // citeste informatia SSID (nume retea)
sendData("AT+CIPMUX=1\r\n", 1000, false); // configurare conexiuni multiple
sendData("AT+CIPSERVER=1,80\r\n", 1000, false); // pornire server pe port 80
}

void loop() {
  if (Serial1.available()) {
    if (Serial1.find("+IPD,") {
      delay(500);
      int connectionId = Serial1.read() - 48; // functia read() returneaza valori zecimal ASCII
      // si caracterul '0' are codul ASCII 48
      String webpage = "<h1>Hello World!</h1><a href=\"/10\"><button>ON</button></a>";
      String cipSend = "AT+CIPSEND=";
      cipSend += connectionId;
      cipSend += ",";
      webpage += "<a href=\"/11\"><button>OFF</button></a>";

      if (readSensor() > 0) {
        webpage += "<h2>Millis:</h2>";
        webpage += readSensor();

      }

      cipSend += webpage.length();
      cipSend += "\r\n";
      sendData(cipSend, 100, DEBUG);
      sendData(webpage, 150, DEBUG);

      String closeCommand = "AT+CIPCLOSE=";
      closeCommand += connectionId; //se adauga identificatorul conexiunii
      closeCommand += "\r\n";
      sendData(closeCommand, 300, DEBUG);
    }
  }
}

String sendData(String command, const int timeout, boolean debug) {
  String response = "";
  Serial1.print(command); // trimite comanda la esp8266
  long int time = millis();
  while ((time + timeout) > millis()) {
    while (Serial1.available()) {
      char c = Serial1.read(); // citeste caracter urmator
      response += c;
    }
  }
  if (response.indexOf("/10") != -1) {
    digitalWrite(LED_BUILTIN, HIGH);
  }
  if (response.indexOf("/11") != -1) {

```

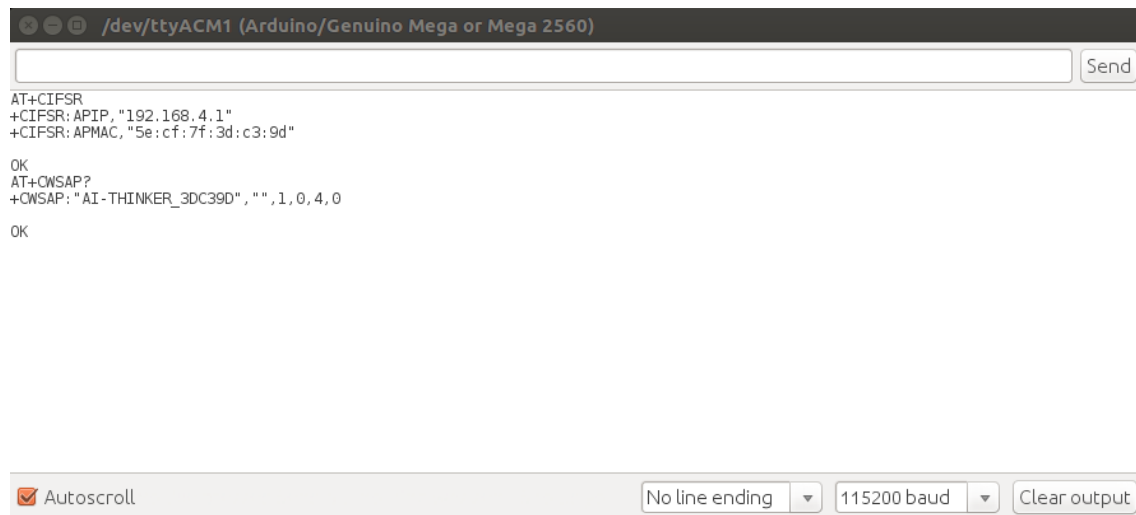
```
digitalWrite(LED_BUILTIN, LOW);
}
if (debug) {
  Serial.print(response);
}
return response;
}
```

```
unsigned long readSensor() {
  return millis();
}
```

Rularea programului

Încărcăți codul pe Arduino. Asigurați-vă că modulul ESP8266 este alimentat la 3.3V, și că pinul EN este conectat tot la 3.3V. **Nu conectați acești pini la 5V, pentru că veți distruge adaptorul!**

Un dispozitiv client convenabil pentru accesarea rețelei este telefonul dvs. Scanați după rețele WiFi, și găsiți numele rețelei care se potrivește cu cel afișat pe Serial Monitor.

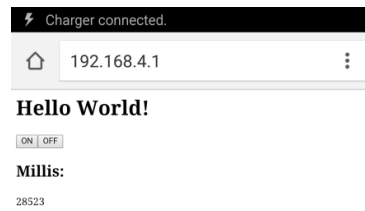
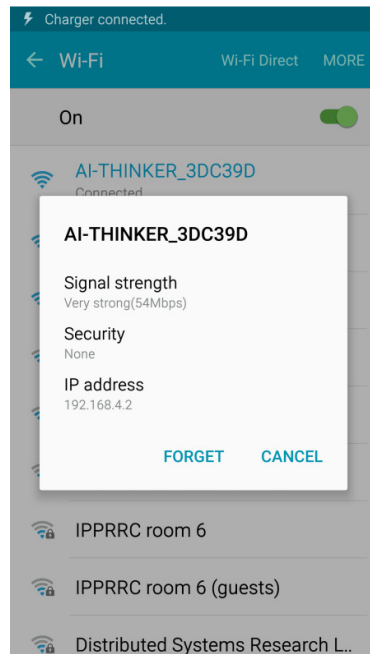


The screenshot shows the Arduino Serial Monitor interface. The title bar reads "/dev/ttyACM1 (Arduino/Genuino Mega or Mega 2560)". The main text area displays the following output:

```
AT+CIFSR
+CIFSR: APIP, "192.168.4.1"
+CIFSR: APMAC, "5e:c7:7f:3d:c3:9d"
OK
AT+CWSAP?
+CWSAP: "AI-THINKER_3DC39D", "", 1, 0, 4, 0
OK
```

At the bottom of the window, there are control elements: a checked "Autoscroll" checkbox, a "No line ending" dropdown menu, a "115200 baud" dropdown menu, and a "Clear output" button.

Nu uitați să schimbați **baud rate-ul din serial monitor la 115200!** După cum se observă, numele rețelei din figura de mai sus este "AT-THINKER_3DC39D", fără parolă. Același nume trebuie să fie vizibil pe dispozitivul dvs. mobil:



Stânga: numele rețelei WiFi a modulului ESP8266, afișat pe telefonul mobil. Dreapta: pagina web cu butoane, afișând valoarea *millis()*.

Notă: fiecare adaptor are numele lui unic de rețea. Conectați-vă la rețeaua adaptorului vostru propriu, nu la rețelele colegilor !

După ce vă conectați la rețeaua WiFi, deschideți pe telefonul mobil un browser web, și scrieți adresa serverului web: **192.168.4.1**. Rezultatul ar trebui să fie cel din imaginea de mai sus, partea dreaptă.

Lucru individual:

1. Rulați exemplul. Asigurați-vă că modulul WiFi este corect conectat, și alimentat la 3.3V.
2. Modificați exemplul pentru a afișa date suplimentare în browser-ul web. Conectați un senzor la Arduino, și, folosind butoanele de pe pagina web, selectați informația de afișat (*millis()* sau datele de la senzor). Asigurați-vă că pagina web informează clar utilizatorul ce fel de informație este afișată.
3. Folosind manualul modulului WiFi, schimbați setările rețelei WiFi: schimbați numele rețelei (SSID), adăugați o parolă și un mod de criptare.