

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

# Norul udă ghivecele cu flori

Chiar dacă un sistem automat de menținere a umidității pentru ghivecelor cu flori rezolvă problema ”am uitat să ud florile” nu rezolvă și problema monitorizării exacte a umidității (necesare pentru diverse tipuri de plante mai pretențioase). Adăugarea unui sistem de jurnalizare a umidității permite vizualizarea și determinarea punctului de umiditate optimă pentru diverse astfel de tipuri de plante. În cadrul lecției de față ne propunem să completăm sistemul prezentat în lecția precedentă (Arduino udă florile) cu un sistem de monitorizare și comandă online. Pentru acest lucru vom utiliza serviciul online (de tip ”nor”) Adafruit IO. Serviciul Adafruit IO este un serviciu specializat IoT gratuit ce pune la dispoziția utilizatorilor atât posibilitatea jurnalizării valorilor preluate de la diverși senzori dar și transmiterea de comenzi către dispozitivele IoT și chiar legătura automată între partea de achiziție și comandă (reguli de conducere automată).



<https://io.adafruit.com>

Pentru implementarea sistemului prezentat în această lecție este necesară crearea unui cont pe platforma Adafruit IO. După conectare trebuie să navigăm în meniul *Settings* și să ne notăm *AIO Keys* (cheia de autentificare) care va fi folosită împreună cu denumirea contului creat în programul sistemului.

Din punct de vedere hardware sistemul va avea în plus un modul Adafruit HUZZAH ESP8266 ce va asigura conectivitatea de rețea necesară accesării serviciului cloud.

<https://www.robofun.ro/forum/>

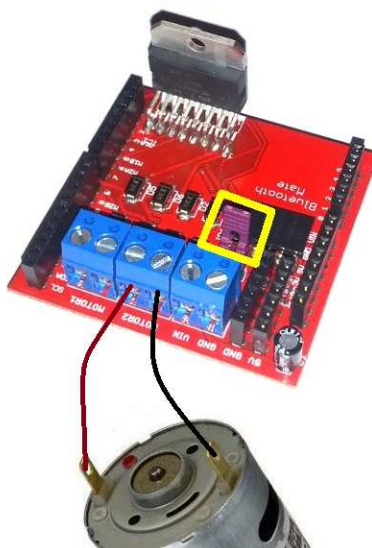


<https://www.robofun.ro/wireless/wireless-wifi/huzzah-esp8266>

Astfel componentele necesare implementării sistemului sunt: [Arduino Uno](#), [shield drivere de motoare L298](#), [senzor umiditate sol](#), [pompa de apă micro](#), [ecran LCD grafic Nokia](#) (opțional) și modul [Adafruit HUZZAH ESP8266](#). Alimentarea sistemului se va face de la un [alimentator de rețea de 9V](#) conectat la placa de dezvoltare Arduino Uno. Alimentatorul va alimenta atât componentele de comandă (placă de dezvoltare, modul WiFi, ecran LCD, senzor de umiditate sol) cât și pompa de apă (prin intermediul shield-ului L298 ce trebuie să aibă jumperul de alimentare comună conectat).

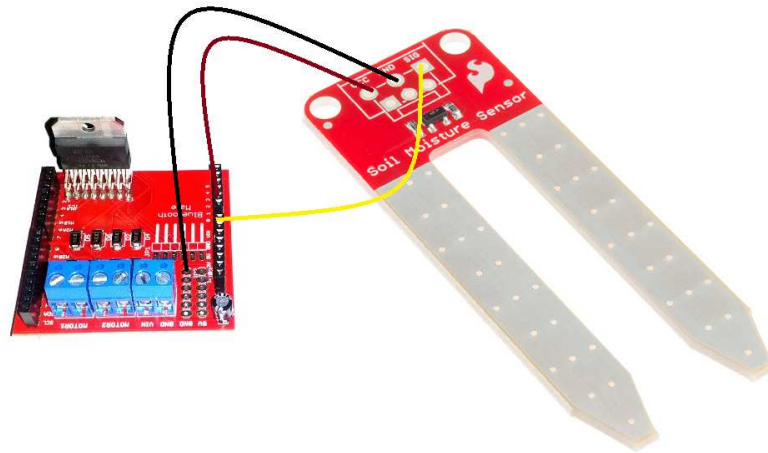
Conexiunile între componentele sistemului sunt următoarele:

- Pompa de apă se conectează la shield-ul L298 pe canalul MOTOR2:

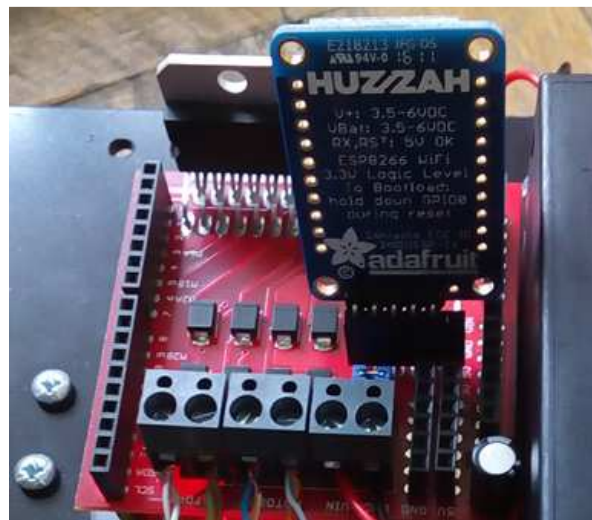


<https://www.robofun.ro/forum/>

- Senzorul de umiditate sol va utiliza pinul A0 pentru a transmite valoarea achiziționată către placa de dezvoltare, pinul A2 pentru alimentare și un pin de GND.



- Ecranul LCD va utiliza următorii pini pentru conectare: LED – 5V (prin intermediul unei rezistențe de 220ohm), SCLK – D13, D/C – D8, RST – D7, SCE – A3 (nu mai putem folosi pinul D4 ca în cazul lecției precedente deoarece acest pin va fi utilizat de modul WiFi), GND – GND, Vcc – 3.3V.
- Modulul WiFi Adafruit HUZZAH ESP8266 se va monta în soclul shield-ului L298 (soclu gândit pentru modul bluetooth dar compatibil și cu modulul WiFi utilizat de noi) – ATENȚIE la polaritate – conectați pinul de GND al modului la pinul de GND al soclului.



Înainte de asamblarea sistemului modulul WiFi trebuie programat separat de programarea plăcii Arduino Uno. Pentru programare este necesară utilizarea unui [cablu USB to TTL](#) sau a unui [conector FTDI](#). Pentru programarea și utilizarea modulului Adafruit HUZZAH este recomandată parcurgerea materialului următor:

Adafruit HUZZAH ESP8266 breakout

<https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout>

Programul pentru modulul WiFi are rolul de a transforma comunicația serială (de la placa Arduino Uno) în mesaje MQTT pentru platforma Adafruit IO. Programul a fost dezvoltat și testat cu Arduino IDE 1.8.1 având instalată extensia ESP8266 Community 2.3.0 și utilizează biblioteca Adafruit MQTT 0.17.0. În program trebuie personalizate datele de conectare la rețeaua WiFi locală (*WLAN\_SSID* și *WLAN\_PASS*) precum și datele de conectare la serviciul cloud Adafruit IO (*AIO\_USERNAME* și *AIO\_KEY*).

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define WLAN_SSID      "... "
#define WLAN_PASS      "... "
#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME    "... "
#define AIO_KEY         "... "

WiFiClient client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER,
    AIO_SERVERPORT, AIO_USERNAME, AIO_USERNAME, AIO_KEY);

Adafruit_MQTT_Publish soilhumidity =
    Adafruit_MQTT_Publish(&mqtt,
        AIO_USERNAME "/feeds/soilhumidity");
Adafruit_MQTT_Subscribe onoffbutton =
    Adafruit_MQTT_Subscribe(&mqtt,
```

```

    AIO_USERNAME "/feeds/onoff");

void onoffcallback(char *data, uint16_t len) {
    String onoff_state = String(data);
    if (onoff_state=="ON") Serial.println("1");
    else if (onoff_state=="OFF") Serial.println("0");
}

void setup() {
    Serial.begin(115200);
    delay(10);
    Serial.println(F("Adafruit MQTT NOR"));
    WiFi.begin(WLAN_SSID, WLAN_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);    }
    onoffbutton.setCallback(onoffcallback);
    mqtt.subscribe(&onoffbutton);
}

uint32_t x;

void loop() {
    MQTT_connect();
    mqtt.processPackets(100);
    if(Serial.available()){
        x=Serial.parseInt();
        soilhumidity.publish(x);
    }
    if(! mqtt.ping()) {
        mqtt.disconnect();
    }
}

```

```

void MQTT_connect() {
    int8_t ret;
    if (mqtt.connected()) {
        return;
    }
    uint8_t retries = 3;
    while ((ret = mqtt.connect()) != 0) {
        mqtt.disconnect();
        delay(10000);
        retries--;
        if (retries == 0) { while (1); }
    }
}

```

Programul pentru placa de dezvoltare Arduino Uno nu diferă foarte mult de varianta prezentată în lecția precedentă. Este necesară modificarea declarării obiectului LCD:

```
Adafruit_PCD8544 display = Adafruit_PCD8544(8, A3, 7);
```

declararea și configurarea comunicației seriale cu modulul WiFi:

```

#include "SoftwareSerial.h"
SoftwareSerial mySerial(2, 4);
void setup()
{
    mySerial.begin(115200);
    ...

```

și de câteva completări în secțiunea *loop()*:

```

void loop()
{

```

```

int command = 0;
int soilhumidity;
...
digitalWrite(SOIL_VCC, LOW);
mySerial.println(soilhumidity);
...
display.println(millis()/1000);
if(mySerial.available()){
    command = mySerial.parseInt();
}
if ((soilhumidity<500)|| (command==1)) {
...

```

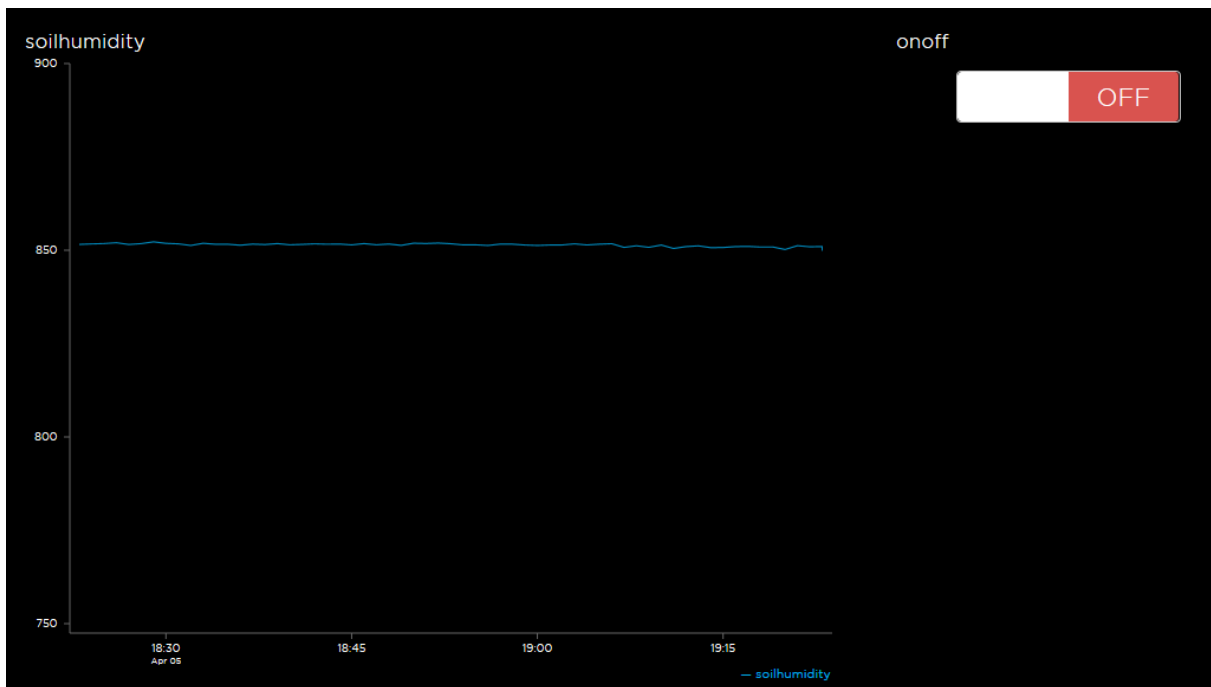
Programul ce rulează pe placa Arduino raportează serial către modulul WiFi valoarea umidității solului iar acesta raportează mai departe către serviciul cloud prin intermediul canalului *soilhumidity*. După prima execuție acest canal va apare automat în interfața Adafruit IO.

<input type="checkbox"/> onoff	onoff	OFF	4 hours ago
<input type="checkbox"/> soilhumidity	soilhumidity	852	a few seconds ago

În plus, sistemul va permite și comanda de la distanță a procesului de udare. Pentru acest lucru este nevoie să creăm manul în interfața Adafruit IO un canal denumit *onoff*. Acest canal va transmite către modulul WiFi comenzi de tipul ON / OFF, acestea vor fi retransmise către placa Arduino Uno ca și comenzi seriale de tip 1 / 0 și vor permite pornirea procesului de udare independent de valoarea umidității solului.

Pentru a putea transmite comenzile din interfața Adafruit IO putem crea o nouă consolă de comandă în secțiunea *Dashboards*. În această consolă de comandă putem plasa graficul de evoluție (*Line Chart*) a parametrului măsurat (umiditate sol) și un buton (*Toggle*) pentru comanda de udare.





În acest moment avem un sistem complet de urmărire și control la distanță a sistemului de udare – interfața web a serviciului Adafruit IO se poate accesa de oriunde de pe Internet.

Bineînțeles, sistemul propus poate fi îmbunătățit prin adăugarea mai multor parametri de supraveghere (putem adăuga senzori de temperatură și umiditate a aerului, senzori de intensitate a luminii solare) completând astfel procesul de urmărire și control. Mai mult decât atât, serviciul Adafruit IO permite definirea de alarme de urmărire (avertizarea prin email când un parametru urmărit atinge un anumit prag) sau chiar definirea de acțiuni asupra elementelor de comandă (putem implementa pragul de declanșare a procesului de udare la nivel de serviciu cloud).

Pentru mai multe detalii legate de funcționarea serviciului Adafruit IO puteți vedea și următoarele proiecte:

Monitor temperature & humidity from anywhere!

<https://learn.adafruit.com/adafruit-io-basics-temperature-and-humidity>

ESP8266 Power Meter Measure the DC power consumption of your devices!

<https://learn.adafruit.com/build-a-cloud-connected-esp8266-power-meter>

Remote Control with the Huzzah + Adafruit.io

<https://learn.adafruit.com/remote-control-with-the-huzzah-plus-adafruit-io>

Log Light Levels to Adafruit IO

<https://learn.adafruit.com/datalogging-hat-with-flora-ble>

Track Your Treats: Halloween Candy GPS Tracker

<https://learn.adafruit.com/track-your-treats-halloween-candy-gps-tracker>

