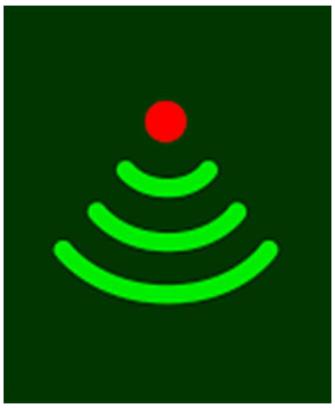
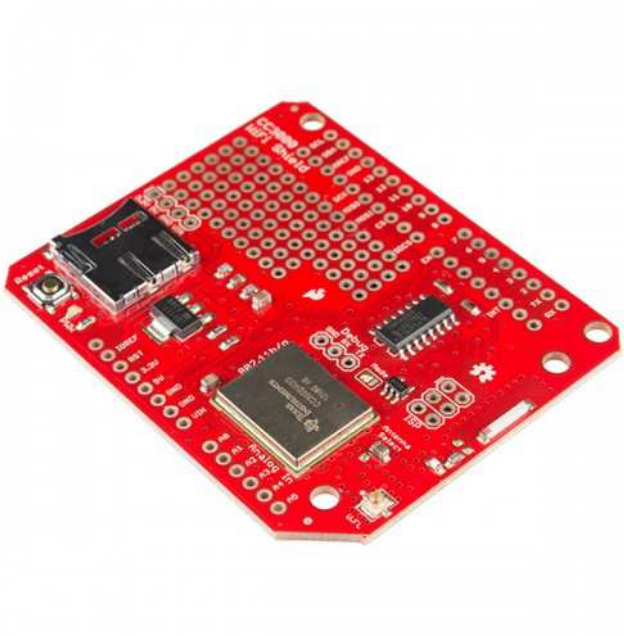
Instalație Crăciun WiFi



Instalațiile de lumini pentru crăciun sunt printre cele mai distractive montaje care se pot realiza cu ajutorul plăcii de dezvoltare Arduino Uno. În cadrul lecției de față vom realiza o instalație de lumini pentru crăciun conectată prin WiFi și controlabilă prin intermediul telefonului mobil.

Pe lângă placa de dezvoltare Arduino Uno vom utiliza shield-ul Sparkfun WiFi CC3000 ce va permite conectarea la rețeaua Internet și recepționarea comenzilor transmise de pe telefonul mobil inteligent.



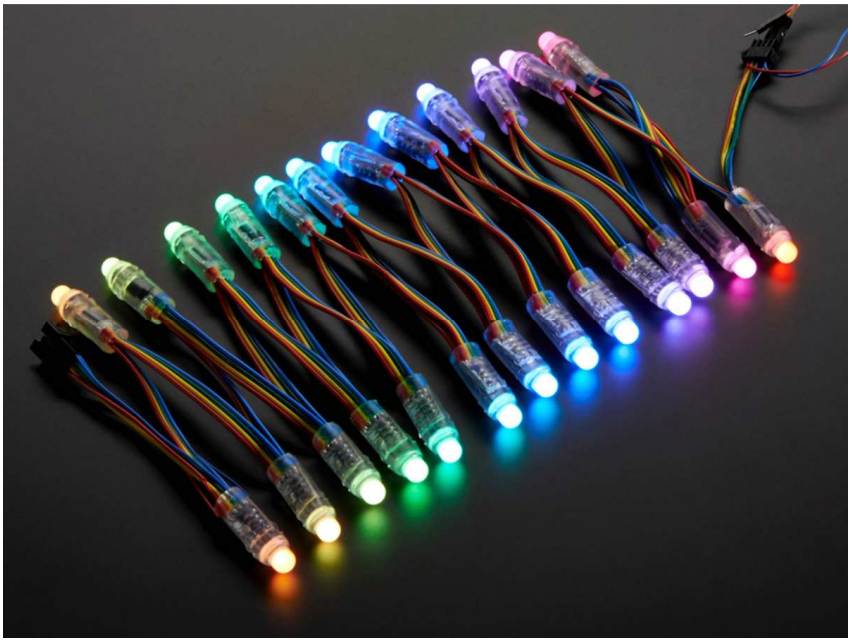
https://www.robofun.ro/wireless/wireless-wifi/cc3000-wifi-shield

Pentru informații legate de utilizarea shield-ului Sparkfun WiFi CC3000 puteți consulta și următorul tutorial:

CC3000 Hookup Guide

https://learn.sparkfun.com/tutorials/cc3000-hookup-guide

Pentru partea de lumini vom utiliza un șir de leduri RGB de 12mm de la Adafruit (se pot utiliza mai multe seturi înseriate atâta timp cât se respectă puterea sursei de alimentare):



https://www.robofun.ro/electronice/led/led-uri-pixeli-rgb-de-12mm

Se recomandă citirea cu atenție a informațiilor de utilizare furnizate de producător:

12mm LED Pixels

https://learn.adafruit.com/12mm-led-pixels

Pentru a realiza conexiunea între sistemul de lumini și telefonul mobil, fără a fi nevoie să scriem o aplicație mobilă personalizată, vom utiliza platforma IoT Blynk:

First drag-n-drop IoT app builder for Arduino, Raspberry Pi, ESP8266, SparkFun boards, and others

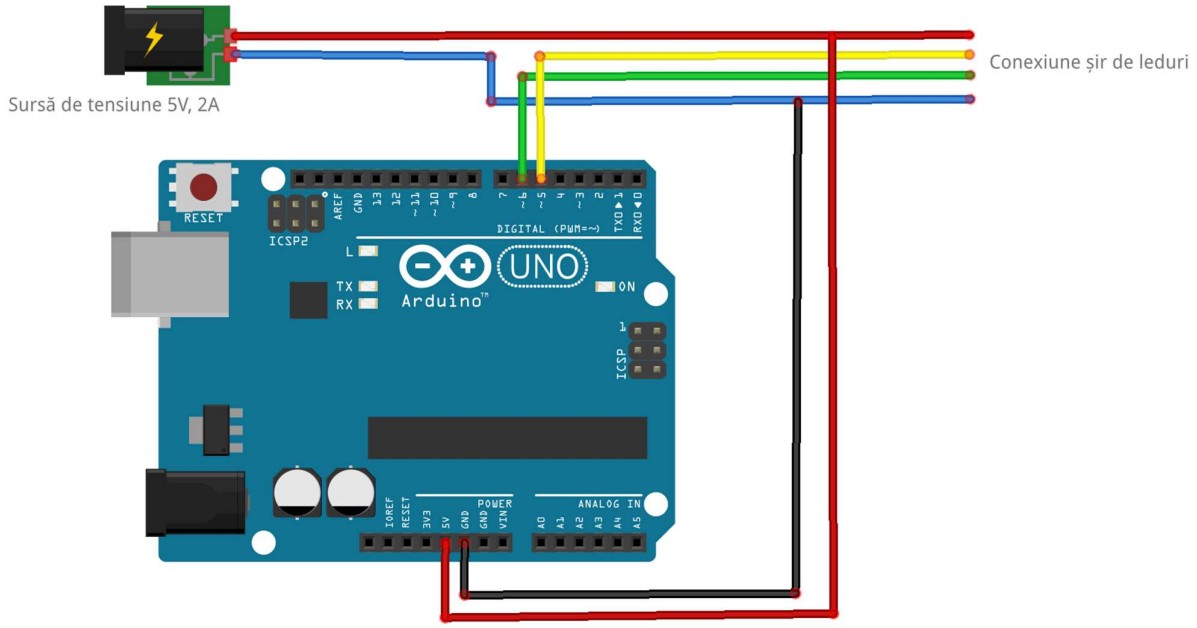
http://www.blynk.cc/

Pentru o mai bună înțelegere a lecției de față se poate revedea și lecția anterioară:

Cum să realizăm un sistem IoT fără să scriem nici o linie de cod?

https://blog.robofun.ro/2016/05/01/cum-sa-realizam-un-sistem-iot-fara-sa-scriem-nici-o-linie-de-cod/

Cea mai importantă grijă în realizarea montajului este alimentarea corectă a acestuia. Trebuie avut în vedere că sistemul va necesita o sursă externă de tensiune de 5V, minim 2A. Schema de interconectare este următoarea:



Chiar dacă schema nu prezintă shield-ul WiFi se presupune că acesta este instalat deasupra plăcii Arduino Uno utilizând un set de conectori Arduino R3:

https://www.robofun.ro/kit_conectori_6_8_10_3

Sursa de tensiune externă va alimenta atât ansamblul placă de dezvoltare Arduino Uno + shield WiFi cât și șirul de 25 de leduri RGB. Comanda de la placa de dezvoltare se va transmite către șirul de leduri pe două fire: pinul 5 – fir galben – date și pinul 6 – fir verde – semnal de ceas.

În cadrul programului Arduino se vor utiliza următoarele biblioteci externe mediului Arduino IDE:

- Blynk Library
  https://github.com/blynkkk/blynk-library
- Adafruit WS2801 Library
  https://github.com/adafruit/Adafruit-WS2801-Library
- Adafruit CC3000 Library
  https://github.com/adafruit/Adafruit_CC3000_Library

```
#include <Adafruit_WS2801.h>

#define ADAFRUIT_CC3000_IRQ   2
#define ADAFRUIT_CC3000_VBAT  7
#define ADAFRUIT_CC3000_CS    10

#include <SPI.h>
#include <Adafruit_CC3000.h>
#include <BlynkSimpleCC3000.h>

int dataPin = 5;
int clockPin = 6;

Adafruit_WS2801 strip =
                Adafruit_WS2801(25, dataPin, clockPin);
```

În program se va personaliza AUTH TOKEN furnizat de aplicația mobilă Blynk la crearea aplicației de comandă:

```
char auth[] = "AUTH TOKEN";
```

și datele de conectarea la rețeaua WiFi locală:

```
char ssid[] = "…";
char pass[] = "…";
int wifi_sec = WLAN_SEC_WPA2;
```

Dacă se dorește urmărirea mesajelor de control ale sistemului Blynk în consola serială se va lăsa următoarea linie necomentată:

```
#define BLYNK_PRINT Serial
```

Următoarele variabile globale permit controlul funcționării sistemului de lumini. Variabila *mode* reține comportamentul de funcționare, există 5 moduri de funcționare: modul 0 – toate ledurile sunt stinse, modul 1 – toate ledurile sunt aprinse având aceiași culoare dictată de aplicația de comandă, modul 2 – toate ledurile sunt aprinse având culori diferite, modul 3 – este aprins un singur led care se plimbă de la un capăt la altul al șirului de leduri și modul 4 – sunt aprinse toate ledurile iar culoarea este dată de accelerometrul telefonului mobil de pe care se face comanda. Variabila *shift* va reține comanda de culoare dată de aplicația mobilă. Variabila *automat* indică declanșarea unui interval de funcționare automată la oră fixă stabilită în aplicația mobilă de comandă. Variabilele *lastmode* și *lastshift* conțin ultimele valori ale variabilelor corespondente pentru ca sistemul să poată să răspundă la modificarea acestora.

```
byte mode = 0;
byte lastmode = 255;
int shift = 0;
int lastshift = -1;
byte automat = 0;
```

În cadrul secțiunii *setup()* se va inițializa conexiunea cu platforma Blynk și obiectul de comandă al șirului de leduri.

```
void setup()
{ Serial.begin(9600);
  Blynk.begin(auth, ssid, pass, wifi_sec);
  strip.begin();
  strip.show(); }
```

Procedurile *BLYNK_WRITE* preiau comenzile primite de la aplicația mobilă prin intermediul unor pini virtuali: pinul V1 va transmite comanda de culoare, pinul V2 va transmite comanda de mod de funcționare, pinul V3 va transmite comanda de funcționare automată și pinul V4 va transmite valorile primite de la accelerometrul telefonului mobil.

```
BLYNK_WRITE(V4) {
  if (mode==4) {
    int r = param[0].asFloat();
```

```
    int g = param[1].asFloat();
    int b = param[2].asFloat();
    shift = Color(r,g,b); }
}

BLYNK_WRITE(V3) {
  automat = param.asInt();
  mode = 3;
}

BLYNK_WRITE(V2) {
  mode = param.asInt();
}

BLYNK_WRITE(V1)
{
  shift = param.asInt();
}
```

În cadrul secțiunii *loop()* este implementat algoritmul de funcționare propriu-zisă a instalației de lumini – cele 5 moduri de funcționare.

```
void loop()
{ Blynk.run();
  if ((lastmode!=mode) || (lastshift!=shift) || automat) {
  switch (mode) {
    case 0:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, 0);
      strip.show();
      break;
    case 1:
```

```
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, Wheel(shift & 255));
      strip.show();
      break;
    case 2:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i,
    Wheel(((i * 256 / strip.numPixels()) + shift) & 255));
      strip.show();
      break;
    case 3:
      for (int i = 0; i < strip.numPixels(); i++) {
        if(i>0) strip.setPixelColor(i-1, 0);
        strip.setPixelColor(i, Wheel(shift & 255));
        strip.show();
        delay(200);
      }
      strip.setPixelColor(strip.numPixels()-1, 0);
      strip.show();
      break;
    case 4:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, shift);
      strip.show();
      break;
  }
  lastmode = mode;
  lastshift = shift;
  }
}
```

Funcțiile *Wheel* și *Color* sunt folosite pentru prelucrarea comenzilor de culoare.

```
uint32_t Wheel(byte WheelPos) {
  if (WheelPos < 85) {
    return Color(WheelPos * 3, 255 - WheelPos * 3, 0);
  } else if (WheelPos < 170) {
    WheelPos -= 85;
    return Color(255 - WheelPos * 3, 0, WheelPos * 3);
  } else {
    WheelPos -= 170;
    return Color(0, WheelPos * 3, 255 - WheelPos * 3);
  }
}

uint32_t Color(byte r, byte g, byte b)
{
  uint32_t c;
  c = r;
  c <<= 8;
  c |= g;
  c <<= 8;
  c |= b;
  return c;
}
```

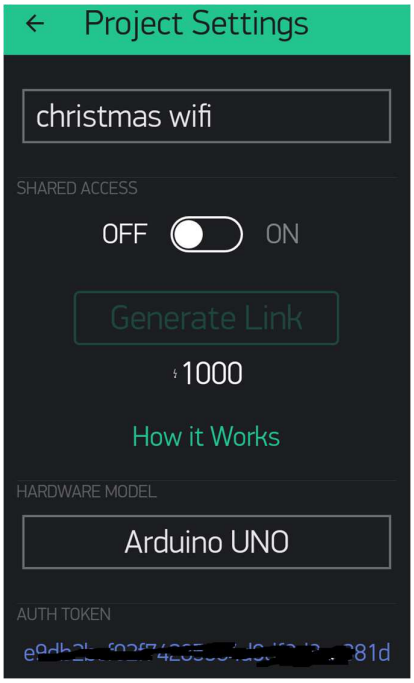
Programul a fost dezvoltat și testat utilizând Arduino IDE 1.6.12, Arduino AVR Boards 1.6.15, Blynk Library 0.4.0, Adafruit CC3000 Library 1.0.3 și Adafruit WS2801 Library 1.0.0.

Bineînțeles, după realizarea și programarea sistemului de lumini este necesară generarea aplicației mobile utilizând aplicația Blynk:

https://play.google.com/store/apps/details?id=cc.blynk

https://itunes.apple.com/us/app/blynk-control-arduino-raspberry/id808760481?ls=1&mt=8

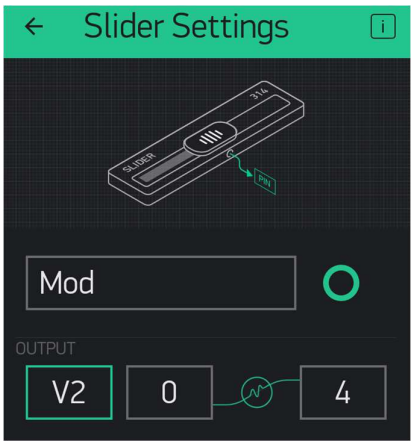
Se generează o aplicație nouă, putem să o botezăm ”cristmas wifi” de exemplu.

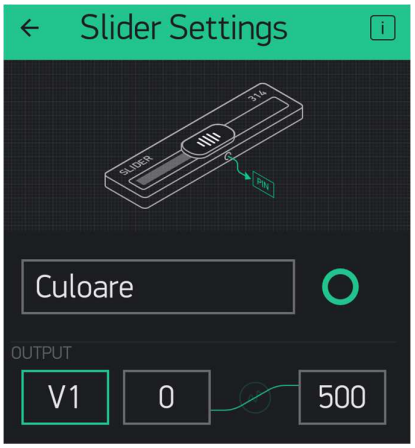


Este important să notăm *AUTH TOKEN* furnizat la crearea aplicației pentru a putea să-l trecem în program.

Aplicația va include 4 controale (Widget Box):

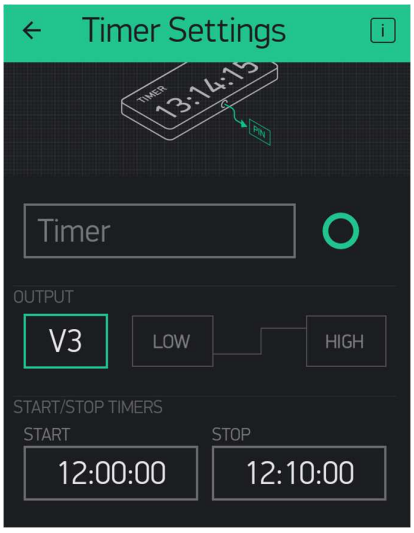
- Un Slider S (200 de credite) conectat la pinul virtual V2 ce va stabili modul de funcționare a sistemului, variază între 0 și 4;



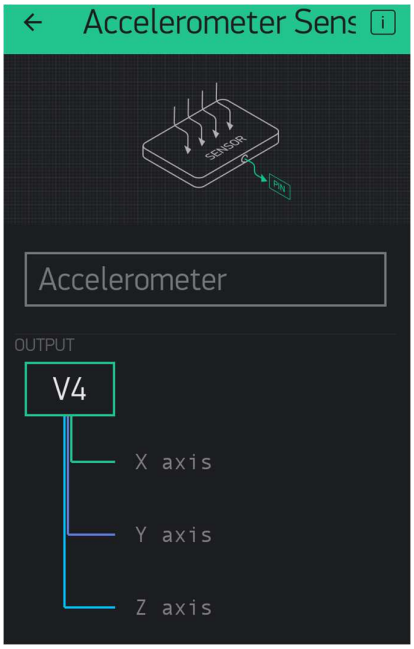


- Un Slider L (200 de credite) conectat la pinul virtual V1 ce va stabili comanda de culoare, variază între 0 și 500;

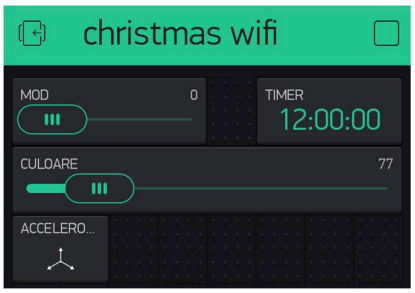
- Un Timer (200 de credite) conectat la pinul virtual V3 ce va declanșa funcționarea automată la oră fixă;



- Un Accelerometer (400 de credite) conectat la pinul virtual V4 ce va transmite valorile pe cele 3 axe ale accelerometrului telefonului mobil.



În final aplicația va arăta în acest fel. Costul total al aplicație 1000 de credite.



La final obținem o instalație de lumini de crăciun la care putem să stabilim culoarea de aprindere, tiparul de aprindere, putem să o programăm să lumineze la oră fixă sau putem pur și simplu să ne facem de cap comandând-o prin intermediul accelerometrului din telefonul mobil. Puteți explora și alte facilități ale platformei Blynk pentru a vă personaliza instalația așa cum doriți.

Se pot consulta și alte metode de comandă WiFi pentru o instalație de lumini dar soluția propusă este una dintre cele mai simple și mai flexibile:

WiFi Controlled LED Christmahanukwanzaa Tree

https://learn.adafruit.com/wifi-controlled-led-christmahanukwanzaa-tree

ESP8266 controlling WS2812 Neopixel LEDs using Arduino IDE - A Tutorial
http://www.instructables.com/id/ESP8266-controlling-Neopixel-LEDs-using-Arduino-ID/

Arduino NeoPixel Wifi
https://create.arduino.cc/projecthub/andres-santos/arduino-neopixel-wifi-d1a93c

Controlling Neopixels with processing via wifi to nodeMCU
http://fablab.ruc.dk/controlling-neopixels-with-processing-via-wifi-to-nodemcu/