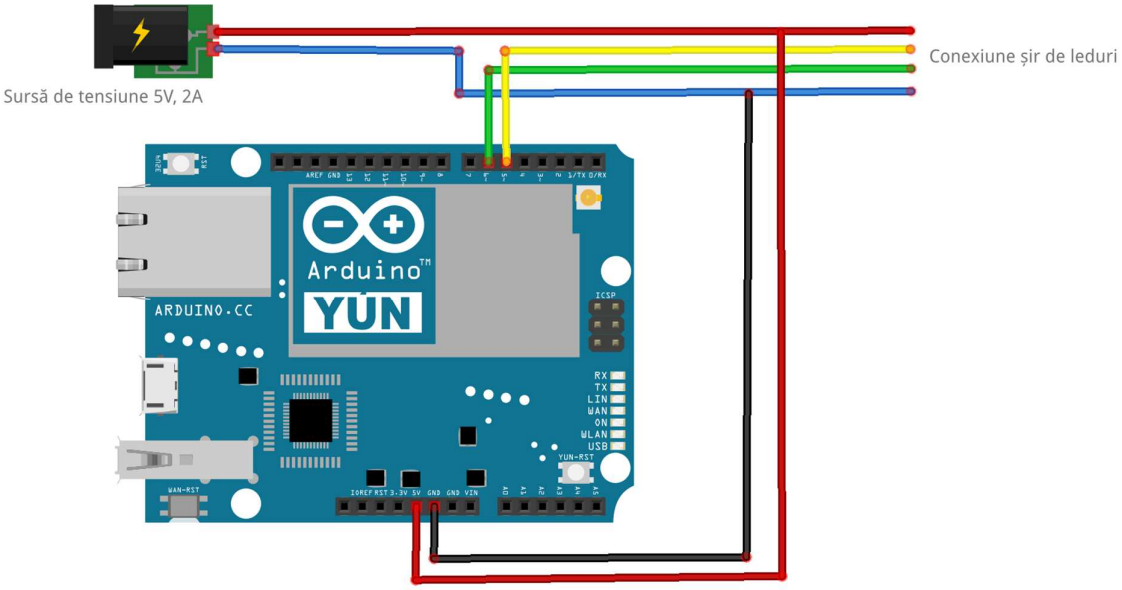
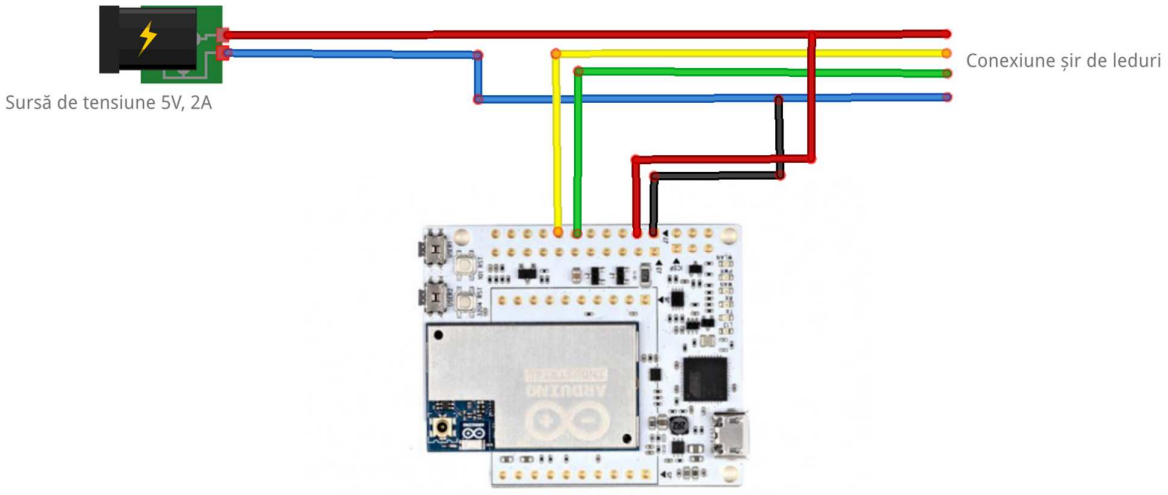
Yun Christmas Lights

Luminițele mele de Crăciun rulează Linux! E un lucru cu care putem uimi chiar și cel mai versat invitat. Mai mult decât atât, putem să le comandăm direct de pe telefonul mobil prin WiFi. Acest lucru este posibil și destul de ușor de implementat utilizând o placă de dezvoltare Arduino Yun (1). Această placă de dezvoltare rulează distribuția Linux OpenWRT (2) și combină conectivitatea de rețea a procesorului Atheros AR9331 cu posibilitatea de comandă în timp real a microcontrolerului ATmega32U4. Pentru lumini vom utiliza 25 de LED-uri RGB WS2801 (3). Aceste LED-uri se comandă simplu prin intermediul a doar două fire (unul de clock și unul de date). Conectarea dintre placa de dezvoltare și LED-uri este următoarea:



Pinul D6 al plăcii de dezvoltare se va conecta la firul verde al șirului de LED-uri (firul de clock) iar pinul D5 la firul galben (firul de date). Alimentarea se va face cu ajutorul unei surse de tensiune de 5V minimum 2A, alimentarea va fi comună pentru placa de dezvoltare și pentru LED-uri. Nu alimentați montajul prin portul USB al plăcii de dezvoltare deoarece un port USB nu poate oferi un curent mai mare de 500mA. Pentru mai multe informații legate de utilizarea șirului de LED-uri puteți consulta și materialul „12mm LED Pixels” (4).

O alternativă mai ieftină pentru sistem este utilizarea plăcii Arduino Industrial 101 (5) – placă mai ieftină și de dimensiuni mai mici. Identică din punct de vedere hardware cu placa Arduino Yun, placa Arduino Industrial 101 expune un număr mai mic de pini ai microcontrolerului ATmega32U4 și nu are interfață ethernet – ambele dezavantaje nu afectează cu nimic montajul propus în lecția de față. Interfațarea cu șirul de LED-uri în cazul plăcii Arduino Industrial 101 este următoarea:



Atât programul cât și configurația conexiunii WiFi sunt identice în cazul ambelor plăci (Arduino Yun și Arduino Industrial 101).

Utilizarea plăcilor de dezvoltare presupune configurarea inițială a conexiunii WiFi (completarea informațiilor de acces WiFi). Utilizarea conexiunii ethernet nu presupune nici o configurație inițială (valabil doar pentru placa Arduino Yun) – interfața ethernet este configurată implicit să obțină o adresă în mod dinamic prin DHCP. Pentru configurarea inițială se recomandă parcurgere materialului „Getting Started with the Arduino Yún LininoOS” (6).

Pentru interfața de comandă mobilă vom utiliza serviciul cloud Blynk (7). Acest serviciu ne va facilita implementarea aplicației de comandă pe dispozitive mobile Android sau iOS. Necesită înregistrarea unui cont gratuit și instalarea aplicației specifice pe dispozitivul mobil. Se poate vedea și materialul „Cum să realizăm un sistem IoT fără să scriem nici o linie de cod?” (8).

Programul pentru placa de dezvoltare a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 având instalate bibliotecile Adafruit WS2801 1.0.0, Blynk 0.4.10 și Time 1.5.0.

```
#include <Adafruit_WS2801.h>
#include <BlynkSimpleYun.h>
#include <TimeLib.h>
#include <WidgetRTC.h>
WidgetRTC rtc;
int dataPin = 5;
int clockPin = 6;
Adafruit_WS2801 strip = Adafruit_WS2801(25, dataPin,
                                clockPin);
```

În cadrul programului trebuie personalizat codul de autentificare în cadrul serviciului Blynk. Acest cod se obține în momentul în care se crează aplicație mobilă de comandă.

```
char auth[] = "…";
```

Sistemul va avea 4 stări posibile (starea va fi stocată în variabila *mode*): starea 0 – LED-urile vor fi stinse, starea 1 – toate LED-urile vor fi aprinse și vor avea aceiași culoare dată de aplicația de comandă, starea 2 – stare implicită, toate LED-uri vor fi aprinse dar vor avea culori diferite, starea 3 – un singur LED va fi aprins și se va plimba de la un cap la altul al șirului de LED-uri. În modul automat (variabila *automat* este implicit *true*) se va închide automat la ora 23:00 și va reporni la ora 10:00 a doua zi, modul automat va putea fi dezactivat din interfața de comandă mobilă.

```
byte mode = 2;
byte lastmode = 255;
int lastcolor = -1;
byte r,g,b;
boolean automat = true;
```

Secțiunea *setup* va inițializa conexiunea cu serviciul cloud Blynk, comunicația cu LED-urile WS2801 și va configura biblioteca Time să comunice la un interval de o oră (3600 de secunde) cu obiectul RTC (definit în interfața mobilă de comandă) pentru sincronizarea ceasului.

```
void setup() {
  Blynk.begin(auth);
  setSyncInterval(60*60);
  strip.begin();
  strip.show();
  r = 0; g = 0; b = 0;
}
```

Procedurile *BLYNK_* tratează evenimentele specifice serviciului cloud preluând comenzile din interfața de comandă mobilă.

```
BLYNK_CONNECTED() {
  rtc.begin();
}

BLYNK_WRITE(V7) {
  b = param.asInt();
}

BLYNK_WRITE(V6) {
  g = param.asInt();
}

BLYNK_WRITE(V5) {
  r = param.asInt();
}

BLYNK_WRITE(V2) {
```

```
  mode = param.asInt();
}

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

În cadrul secțiunii *loop* se implementează comanda efectivă către șirul de LED-uri WS2801 în conformitate cu informațiile primite din interfața mobilă.

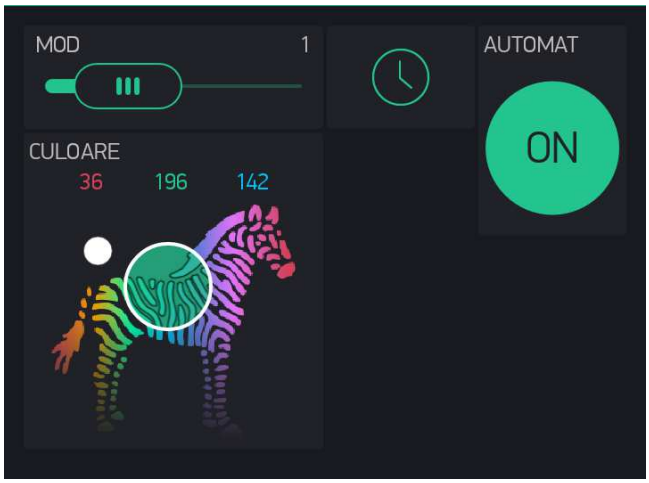
```
void loop()
{
  Blynk.run();
  if (automat) { if ((hour()>22) || (hour()<10)) mode = 0;
  else if (mode==0) mode=2; }
  if ((lastmode!=mode) || (lastcolor!=Color(r,g,b))) {
  switch (mode) {
    case 0:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, 0);
      strip.show();
      break;
    case 1:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, Color(r,g,b));
      strip.show();
      break;
    case 2:
      for (int i = 0; i < strip.numPixels(); i++)
        strip.setPixelColor(i, Wheel(((i * 256 /
             strip.numPixels()) + Color(r,g,b)) & 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, Color(r,g,b));
        strip.show();
        Blynk.run();
        delay(200);
      }
      strip.setPixelColor(strip.numPixels()-1, 0);
      strip.show();
      break;
  }
  lastmode = mode;
  lastcolor = Color(r,g,b);
  }
}
```

Funcțiile Wheel și Color sunt utilizate pentru a calcula comanda de culoare către șirul de LED-uri WS2801.

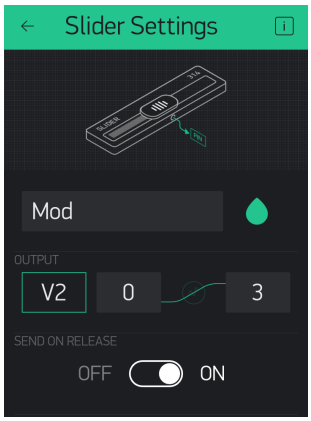
```
uint32_t Wheel(byte WheelPos) {
  if (WheelPos < 85) {
    return Color(WheelPos * 3, 255 - WheelPos * 3, 0);
  } else if (WheelPos < 170) {
    WheelPos -= 85;
    return Color(255 - WheelPos * 3, 0, WheelPos * 3);
  } else {
    WheelPos -= 170;
    return Color(0, WheelPos * 3, 255 - WheelPos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

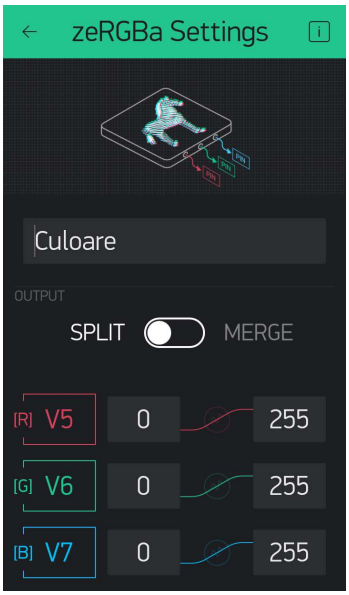


Aplicația mobilă va conține 4 controale (cost total 900 de credite din cele 2000 oferite gratuit):

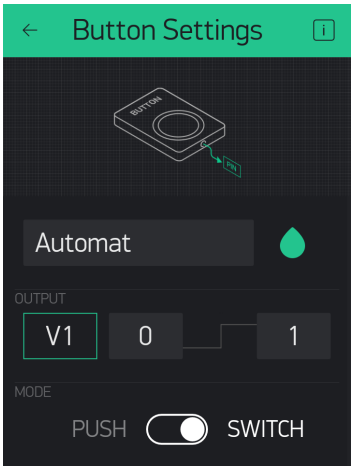
- Un control Slider pentru stabilirea modului de funcționare (cost 200 de credite);
- Un control ZeRGBa pentru stabilirea culorii LED-urilor în modurile de funcționare 1 și 3 (cost 400 de credite);
- Un control Button pentru activarea / dezactivare funcționării automate a sistemului (cost 200 de credite);
- Un control Real Time Clock (RTC) pentru sincronizarea timpului (cost 100 de credite).



Controlul Slider va fi conectat la pinul virtual V2 și va lua valori între 0 și 3 (cele patru moduri de funcționare ale sistemului).



Controlul zeRGBa va fi conectat la pinii virtuali V5, V6 și V7 ce vor trimite către sistem comanda RGB de culoare.



Controlul Button se va configura ca fiind de tip Switch și se va conecta la pinul virtual V1.

Pentru alte idei și alternative de realizare se pot trece în revistă și următoarele proiecte:

- WiFi Controlled RGB LED Strip (9)
- Control RGB LEDs Using Android and Arduino (10)
- Controlling RGB LED using Arduino and Wi-Fi (11)
- Make A WiFi Controllable LED Strip (12)
- ESP8266 based Wifi RGB Controller (H801) (13)

Referințe on-line

(1) Arduino Yun

https://www.robofun.ro/arduino_yun?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(2) OpenWrt

https://openwrt.org/

(3) LED-uri RGB de 12mm (25 pe fir)

https://www.robofun.ro/led-uri-pixeli-rgb-de-12mm?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(4) 12mm LED Pixels

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

(5) Arduino Industrial 101

https://www.robofun.ro/arduino-industrial-101?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(6) Getting Started with the Arduino Yún LininoOS

https://www.arduino.cc/en/Guide/ArduinoYunLin

(7) Blynk

http://www.blynk.cc/

(8) 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/

(9) WiFi Controlled RGB LED Strip

http://www.instructables.com/id/WiFi-Controlled-RGB-LED-Strip/

(10) Control RGB LEDs Using Android and Arduino

https://dzone.com/articles/control-rgb-led-using-android-and-arduino

(11) Controlling RGB LED using Arduino and Wi-Fi
https://circuitdigest.com/microcontroller-projects/controlling-rgb-led-using-wifi-and-arduino

(12) Make A WiFi Controllable LED Strip
http://jakebergamin.com/2016/02/08/wifi-led-strip/

(13) ESP8266 based Wifi RGB Controller (H801)
https://eryk.io/2015/10/esp8266-based-wifi-rgb-controller-h801/