

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursă din acest document este licențiat

Public-Domain

Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără nici un fel de limitări dar autorii nu își asumă nici o răspundere pentru pagubele pricinuite de implementările realizate de utilizatori. Schemele și codul sursă au un rol educativ și nu sunt gândite pentru a fi utilizate în mediu de producție (industrial, casnic sau comercial).

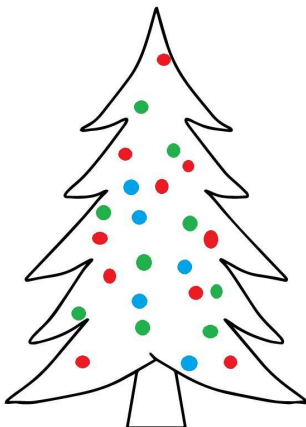
# Calendar de Crăciun

Calendarul de Crăciun sau Calendarul de Advent ([1](#)) este o modalitate de a face să treacă timpul mai ușor în așteptarea sărbătorii de Crăciun. Acest calendar indică câte zile au trecut și câte zile mai sunt (în luna decembrie bineînțeles) până în dimineața zilei de Crăciun. Pentru copii există variante care oferă zilnic mici surprize (bomboane sau ciocolată) pentru a atenua nerăbdarea așteptării cadourilor de Crăciun dar, în cadrul lecției de față, propunem construirea unui calendar de Crăciun bazat pe 24 de LED-uri: un calendar de Crăciun cu luminițe.

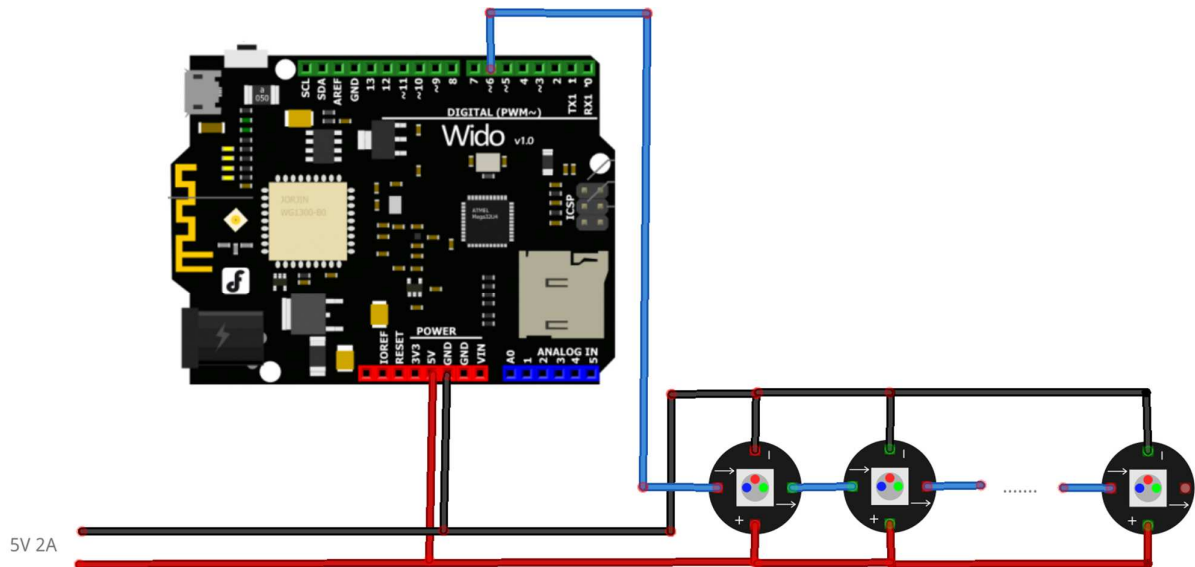


Pentru a comanda mai ușor cele 24 de LED-uri vom utiliza LED-uri RGB adresabile WS2812. Acestea nu necesită comandă individuală, este suficientă o singură linie de comandă între cele 24 de LED-uri și placa de dezvoltare. Se pot utiliza LED-uri NeoPixel ([2](#)), ([3](#)) sau module WS2812 breakout ([4](#)). Pentru mai multe detalii legate de funcționarea LED-urilor WS2812 se recomandă parcurgerea materialului: „Adafruit NeoPixel Überguide” ([5](#)).

Pentru comandă vom utiliza placa de dezvoltare WiDo ([6](#)) echipată cu un microcontroler ATmega32U4 (prezent și pe placa Arduino Leonardo) și un controler WiFi CC3000. Această combinație este perfectă pentru implementarea sistemului nostru: microcontrolerul ATmega32U4 va asigura comanda LED-urilor la o tensiune de 5V iar controlerul WiFi va fi utilizat pentru a putea ști în ce zi calendaristică ne aflăm (sincronizare de timp de tip NTP).



Conexiunile dintre placa de dezvoltare și LED-uri sunt prezentate în schema de pe pagina următoare. Sistemul necesită o alimentare de 5V minim 2A (pentru placa de dezvoltare și cele 24 de LED-uri). Pinul de comandă este pinul D6. Comanda se va transmite serial de la un LED la următorul. LED-urile pot fi aranjate pe un carton pictat cu un brad de Crăciun sau orice alt suport doriți (o căsuță de poveste sau o ilustrată de Crăciun).



Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.3 și bibliotecile Adafruit NeoPixel 1.1.3 și o versiune modificată a bibliotecii Adafruit CC3000 (7). Placa se va programa în Arduino IDE ca o placă Arduino Leonardo obișnuită.

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
    #include <avr/power.h>
#endif

#define PIN                6
#define NUMPIXELS          24

Adafruit_NeoPixel pixels = Adafruit_NeoPixel(NUMPIXELS,
                                              PIN, NEO_GRB + NEO_KHZ800);

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

#define WiDo_IRQ    7
#define WiDo_VBAT   5
```

```
#define WiDo_CS      10
```

```
Adafruit_CC3000 WiDo = Adafruit_CC3000(WiDo_CS, WiDo_IRQ,  
                                         WiDo_VBAT,SPI_CLOCK_DIVIDER);
```

În cadrul programului trebuie personalizate datele de conectare la rețeaua WiFi (*WLAN\_SSID* și *WLAN\_PASS*).

```
#define WLAN_SSID      ""  
#define WLAN_PASS      ""  
#define WLAN_SECURITY  WLAN_SEC_WPA2
```

```
Adafruit_CC3000_Client client;
```

```
#include <TimeLib.h>
```

Pentru sistemul final, directiva debug se poate comenta pentru a suprima mesajele din consola serială.

```
#define debug
```

```
void setup() {  
  #ifdef debug  
    SerialUSB.begin(115200);  
    while(!SerialUSB) { ; }  
    SerialUSB.println(F("Calendar de Craciun\n"));  
    SerialUSB.println(F("\nInitialising the CC3000 ..."));  
  #endif  
  if (!WiDo.begin()) {  
    #ifdef debug  
      SerialUSB.println(F("Unable to initialise the  
                           CC3000! Check your wiring?"));  
    #endif  
  }  
}
```

```

        #endif
        while(1);    }
    if
    (!WiDo.connectToAP(WLAN_SSID,WLAN_PASS,WLAN_SECURITY)) {
        #ifdef debug
            SerialUSB.println(F("Failed to connect to AP!"));
        #endif
        while(1);    }
    #ifdef debug
        SerialUSB.println(F("Connected to AP!"));
        SerialUSB.println(F("Request DHCP"));
    #endif
    while (!WiDo.checkDHCP()) {    delay(100); }
    while (! displayConnectionDetails()) {
        delay(1000);
    }
    setSyncProvider(getTime);
    pinMode(6,OUTPUT);
    pixels.begin();
    randomSeed(analogRead(0));
    startprogram();
}

bool displayConnectionDetails(void)
{
    uint32_t ipAddress, netmask, gateway, dhcpserv, dnsserv;
    if(!WiDo.getIPAddress(&ipAddress, &netmask, &gateway,
                        &dhcpserv, &dnsserv)) {
        #ifdef debug
            SerialUSB.println(F("Unable to retrieve the IP
                                Address!\r\n"));
        #endif
    }
}

```

```

        return false;
    }
    else
    {
        #ifdef debug
            SerialUSB.print(F("\nIP Addr: "));
            WiDo.printIPdotsRev(ipAddress);
            SerialUSB.print(F("\nNetmask: "));
            WiDo.printIPdotsRev(netmask);
            SerialUSB.print(F("\nGateway: "));
            WiDo.printIPdotsRev(gateway);
            SerialUSB.print(F("\nDHCPsrv: "));
            WiDo.printIPdotsRev(dhcpserve);
            SerialUSB.print(F("\nDNSServ: "));
            WiDo.printIPdotsRev(dnsserv);
            SerialUSB.println();
        #endif
        return true;
    }
}

```

Procedura *startprogram()*, apelată o singură dată în secțiunea *setup()*, aprinde toate LED-urile de trei ori în trei culori diferite: roșu, verde, albastru. Este o procedură de verificare a montajului ce se execută la fiecare pornire a sistemului.

```

void startprogram() {
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(255,0,0));
        pixels.show();
        delay(100);
    }
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0,0,0));
    }
}

```

```

    }
    pixels.show();
    delay(100);
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0,255,0));
        pixels.show();
        delay(100);
    }
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0,0,0));
    }
    pixels.show();
    delay(100);
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0,0,255));
        pixels.show();
        delay(100);
    }
    for (int i=0; i<NUMPIXELS; i++) {
        pixels.setPixelColor(i, pixels.Color(0,0,0));
    }
    pixels.show();
    delay(100);
}

```

Secțiunea *loop()* va aștepta realizarea sincronizării de timp după care va verifica o dată la 10 minute data calendaristică. Între 1 decembrie și 24 decembrie va aprinde un număr de LED-uri egal cu data calendaristică. Culoarea va fi de fiecare dată generată aleatoriu. În data de 25 decembrie programul va executa un joc de lumini pe toată perioada zilei (va aprinde toate cele 24 de LED-uri treptat în culori aleatorii). În restul anului LED-urile vor rămâne stinse.

```

void loop() {

```

```

#ifdef debug
    if(timeStatus()==timeNotSet)
        SerialUSB.println("Time Not Set");
    if(timeStatus()==timeNeedsSync)
        SerialUSB.println("Time Needs Sync");
#endif
while(timeStatus()==timeNotSet) delay(1000);
for (int i=0; i<NUMPIXELS; i++) {
    pixels.setPixelColor(i, pixels.Color(0,0,0));
}
pixels.show();
delay(100);
if(month()==12) {
    if (day(<25) {
        #ifdef debug
            SerialUSB.print("Days to Christmas: ");
            SerialUSB.println(25-day());
        #endif
        for (int i=0; i<NUMPIXELS; i++) {
            pixels.setPixelColor(i, pixels.Color(0,0,0));
        }
        pixels.show();
        delay(100);
        for (int i=0; i<day(); i++) {
            pixels.setPixelColor(i,
                pixels.Color(random(0,255),
                    random(0,255),random(0,255)));
        }
        pixels.show();
    }
}
if (day()==25) {
    for (int i=0; i<NUMPIXELS; i++) {

```



```

        pixels.setPixelColor(i,
            pixels.Color(random(0,255),
                random(0,255),random(0,255)));
        pixels.show();
        delay(1000);
    }
}
}
delay(600000);
}

```

Funcția *getTime()* este folosită de biblioteca Time pentru sincronizarea de timp NTP.

```

const unsigned long
    connectTimeout  = 15L * 1000L,
    responseTimeout = 15L * 1000L;

unsigned long getTime(void) {
    uint8_t      buf[48];
    unsigned long ip, startTime, t = 0L;
    #ifdef debug
        SerialUSB.print(F("Locating time server..."));
    #endif
    if(WiDo.getHostByName("pool.ntp.org", &ip)) {
        static const char PROGMEM
            timeReqA[] = { 227,  0,  6, 236 },
            timeReqB[] = {  49, 78, 49,  52 };
        #ifdef debug
            SerialUSB.println(F("\r\nAttempting
                                connection..."));
        #endif
        startTime = millis();
    }
}

```

```

do {
    client = WiDo.connectUDP(ip, 123);
} while ((!client.connected()) &&
        ((millis() - startTime) < connectTimeout));
if(client.connected()) {
    #ifdef debug
        SerialUSB.print(F("connected!\r\nIssuing
                           request..."));
    #endif
    memset(buf, 0, sizeof(buf));
    memcpy_P( buf, timeReqA, sizeof(timeReqA));
    memcpy_P(&buf[12], timeReqB, sizeof(timeReqB));
    client.write(buf, sizeof(buf));
    #ifdef debug
        SerialUSB.print(F("\r\nAwaiting response..."));
    #endif
    memset(buf, 0, sizeof(buf));
    startTime = millis();
    while ((!client.available()) &&
            ((millis() - startTime) < responseTimeout));
    if(client.available()) {
        client.read(buf, sizeof(buf));
        t = (((unsigned long)buf[40] << 24) |
              ((unsigned long)buf[41] << 16) |
              ((unsigned long)buf[42] << 8) |
              (unsigned long)buf[43]) - 2208988800UL;
        #ifdef debug
            SerialUSB.print(F("OK\r\n"));
        #endif
    }
    client.close();
}

```

```
}  
#ifdef debug  
    if(!t) SerialUSB.println(F("error"));  
#endif  
return t;  
}
```

Pentru idei suplimentarea în realizarea artistică a aranjamentului se pot vedea și următoarele două proiecte:

- Christmas Advent Calendar (Arduino powered) // How-To – YouTube ([8](#))
- DIY WIFI Arduino ESP8266 Advent Calendar ([9](#))

## Referințe on-line

(1) Ce este calendarul de Advent? | Povești săsești

<https://povestisasesti.com/2013/12/01/1-in-asteptarea-craciunului-ce-este-calendarul-de-advent/>

(2) LED RGB NeoPixel cu Driver Incorporat - Pachet de 10

[https://www.robofun.ro/neopixel-5050-rgb-led-with-integrated-driver-chip-10-pack?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/neopixel-5050-rgb-led-with-integrated-driver-chip-10-pack?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(3) Flora RGB LED Smart NeoPixel WS2812 - kit 4 bucati

[https://www.robofun.ro/flora-rgb-smart-neopixel-version-2-pack-4?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/flora-rgb-smart-neopixel-version-2-pack-4?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(4) WS2812 RGB LED Breakout

[https://www.robofun.ro/electronic/led/ws2812-rgb-led-breakout?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/electronic/led/ws2812-rgb-led-breakout?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(5) Adafruit NeoPixel Überguide

<https://learn.adafruit.com/adafruit-neopixel-uberguide>

(6) WiDo - Arduino Iot

[https://www.robofun.ro/wido-arduino-iot-internet-of-thing-board?utm\\_source=newsletter&utm\\_medium=email&utm\\_content=productLink&utm\\_campaign=CURS\\_EMAIL](https://www.robofun.ro/wido-arduino-iot-internet-of-thing-board?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL)

(7) DFRobot\_WiDo\_Arduino\_IOT\_Board\_DFR0321

[https://github.com/Arduinolibrary/DFRobot\\_WiDo\\_Arduino\\_IOT\\_Board\\_DFR0321/](https://github.com/Arduinolibrary/DFRobot_WiDo_Arduino_IOT_Board_DFR0321/)

(8) Christmas Advent Calendar (Arduino powered) // How-To - YouTube

<https://www.youtube.com/watch?v=A3rJ5Um3lS4>

(9) DIY WIFI Arduino ESP8266 Advent Calendar

<https://oshlab.com/advent-calendar/>