

Textul și imaginile din acest document sunt licențiate

Attribution-NonCommercial-NoDerivs

CC BY-NC-ND



Codul sursă din acest document este licențiat

Public-Domain

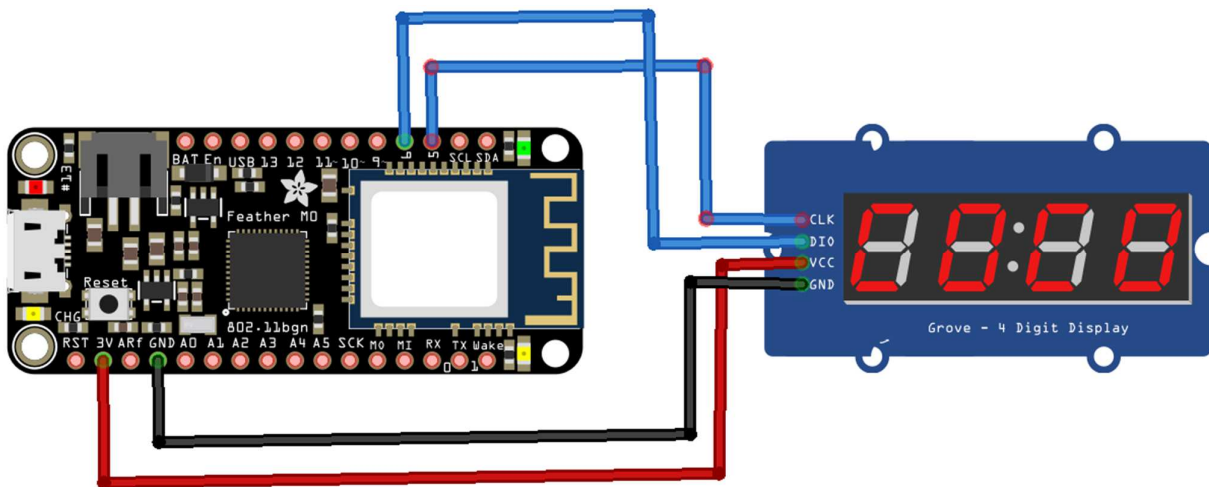
Ești liber să distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, tipărire, sau orice alt mijloc), atât timp cât nu aduci nici un fel de modificări acestuia. Codul sursă din acest document poate fi utilizat în orice fel de scop, de natură comercială sau nu, fără 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).

New Year Countdown

Un element important al oricărui revelion este, bineînțeles, numărătoarea inversă până la trecerea în noul an - cronometrul care ne arată cât timp mai este până la deschiderea sticlei de șampanie și până la pornirea focurilor de artificii. În cadrul lecției de față ne propunem să implementăm un sistem care să ne arate cu precizie cât timp mai este până la un anume moment de timp, poate să fie trecerea în noul an, o aniversare sau orice alt moment important.



Pentru a putea calcula cu exactitate timpul rămas trebuie să știm în primul rând data și ora la momentul actual. Pentru acest lucru vom utiliza o placă cu conectivitate WiFi ce ne va permite sincronizare de timp NTP (1) și anume placa de dezvoltare Adafruit Feather M0 WiFi (2). Această placă este similară cu placa Arduino MKR1000 (3), montajul propus poate fi realizat cu oricare dintre ele. Pe lângă conectivitatea WiFi, ambele plăci de dezvoltare permit alimentarea de la un acumulator LiPo de 3.7V permițând astfel realizarea simplă a unui montaj portabil. Pentru afișare vom utiliza un modul cu afișaj pe 7 segmente cu 4 caractere și circuit de comandă TM1637 (4). Schema de interconectare dintre placa de dezvoltare și modulul de afișare este:



Modulul de afișare se va alimenta la 3.3V (pinul VCC se va conecta la pinul de 3V al plăcii de dezvoltare iar pinul GND la pinul de GND). Pinul CLK se va conecta la pinul 5 al plăcii de dezvoltare și pinul DIO la pinul 6. Comunicația între placa de dezvoltare și modulul de afișare se face serial.

Programul a fost dezvoltat și testat utilizând Arduino IDE 1.8.5 având instalate bibliotecile TM1637Display ([5](#)), Time 1.5.0, WiFi101 0.14.5 și extensia Adafruit SAMD Boards 1.0.21.

```
#include <TM1637Display.h>
#define CLK 5
#define DIO 6
TM1637Display display(CLK, DIO);

#include <SPI.h>
#include <WiFi101.h>
#include <WiFiUdp.h>
```

În cadrul programului trebuie personalizate informațiile de conectare WiFi (*ssid[]* și *pass[]*). Constanta *timeZone* conține decalajul de fus orar specific zonei geografice în care ne aflăm (2 pentru ora de iarnă în România).

```
int status = WL_IDLE_STATUS;
char ssid[] = "...";
char pass[] = "...";

unsigned int localPort = 2390;
IPAddress timeServer(129, 6, 15, 28);
const int NTP_PACKET_SIZE = 48;
byte packetBuffer[ NTP_PACKET_SIZE];
WiFiUDP Udp;

#include <TimeLib.h>
const int timeZone = 2;
tmElements_t TheDay;
```

Structurile *SEG_DONE* și *dy* conțin configurația de segmente pentru mesajul *done* și pentru literele *d* și *Y*. Mesajul *done* va fi afișat după ce numărătoare inversă se va

încheia. Litele *d* și *Y* vor fi utilizate când se vor afișa anii și zilele rămase pâna la evenimentul programat.

```
const uint8_t SEG_DONE[] = {
    SEG_B | SEG_C | SEG_D | SEG_E | SEG_G,          // d
    SEG_A | SEG_B | SEG_C | SEG_D | SEG_E | SEG_F,  // o
    SEG_C | SEG_E | SEG_G,                          // n
    SEG_A | SEG_D | SEG_E | SEG_F | SEG_G           // e
};

const uint8_t dy[] = {
    SEG_B | SEG_C | SEG_D | SEG_E | SEG_G,          // d
    SEG_B | SEG_E | SEG_F | SEG_G ,                 // Y
};
```

În cadrul secțiunii *setup* se realizează conectarea la rețeaua WiFi și programarea evenimentului dorit (în exemplul nostru 1 ianuarie 2018 ora 0:00, anul este înregistrat ca diferență față de anul 1970). Instrucțiunea *WiFi.setPins* este necesară pentru placa Adafruit Feather M0, se va șterge în cazul plăcii Arduino MKR1000.

```
void setup() {
    WiFi.setPins(8,7,4,2);
    if (WiFi.status() == WL_NO_SHIELD) {
        while (true);
    }
    while ( status != WL_CONNECTED) {
        status = WiFi.begin(ssid, pass);
        delay(10000);
    }
    Udp.begin(localPort);
    setSyncProvider(getNtpTime);
    setSyncInterval(3600);
    TheDay.Second = 0;
```

```

TheDay.Minute = 0;
TheDay.Hour = 0;
TheDay.Day = 1;
TheDay.Month = 1;
TheDay.Year = 48;
}

```

Secțiunea *loop* implementează partea de verificare a diferenței de timp între momentul actual și data evenimentului (diferența se calculează în secunde și este stocată în variabila *dif*) și partea de afișare. Dacă diferența este negativă înseamnă că evenimentul a trecut și se afișează *done*. Dacă diferența este mai mare de 31556926 secunde (adică un an) pe ecran se va afișa diferența în ani întregi, pe trei caractere urmate de litera *Y*. Dacă diferența este mai mare de 86400 secunde (adică o zi) pe ecran se va afișa diferența în zile, pe trei caractere urmate de litera *d*. Dacă diferența este mai mică de o zi dar mai mare de o oră (3600 secunde) pe ecran se va afiș diferența în ore:min. În final, când avem un interval mai mic de o oră, pe ecran se va afișa diferența în format min:sec.

```

void loop() {
    time_t dif;
    dif = makeTime(TheDay) - now();
    display.setBrightness(0x0f);
    if (dif < 0) display.setSegments(SEG_DONE);
    else if (dif>31556926L) {
        display.showNumberDec(((1970+TheDay.Year) -
                                (year()+1)),true,3,0);
        display.setSegments(dy+1, 1, 3); }
    else if (dif>86400L) {
        display.showNumberDec((dif/86400L),true,3,0);
        display.setSegments(dy, 1, 3); }
    else if (dif>3600) {
        display.showNumberDecEx((dif/3600L),0xFF,true,2,0);
        display.showNumberDec(((dif%3600L)/60),true,2,2); }
    else {
        display.showNumberDecEx((dif/60),0xFF,true,2,0);

```

```

        display.showNumberDec((dif%60L),true,2,2); }
    delay(1000);
}

```

Funcții `getNtpTime` și `sendNTPpacket` sunt utilizate de biblioteca `Time` pentru sincronizarea de timp.

```

time_t getNtpTime() {
    sendNTPpacket(timeServer);
    delay(1000);
    if ( Udp.parsePacket() ) {
        Udp.read(packetBuffer, NTP_PACKET_SIZE);
        unsigned long highWord = word(packetBuffer[40],
            packetBuffer[41]);
        unsigned long lowWord = word(packetBuffer[42],
            packetBuffer[43]);
        unsigned long secsSince1900 = highWord << 16 | lowWord;
        const unsigned long seventyYears = 2208988800UL;
        unsigned long epoch = secsSince1900 - seventyYears;
        return (time_t) (secsSince1900 - 2208988800UL +
            timeZone * SECS_PER_HOUR);
    }
    else return 0;
}

```

```

unsigned long sendNTPpacket(IPAddress& address) {
    memset(packetBuffer, 0, NTP_PACKET_SIZE);
    packetBuffer[0] = 0b11100011;
    packetBuffer[1] = 0;
    packetBuffer[2] = 6;
    packetBuffer[3] = 0xEC;
    packetBuffer[12] = 49;
}

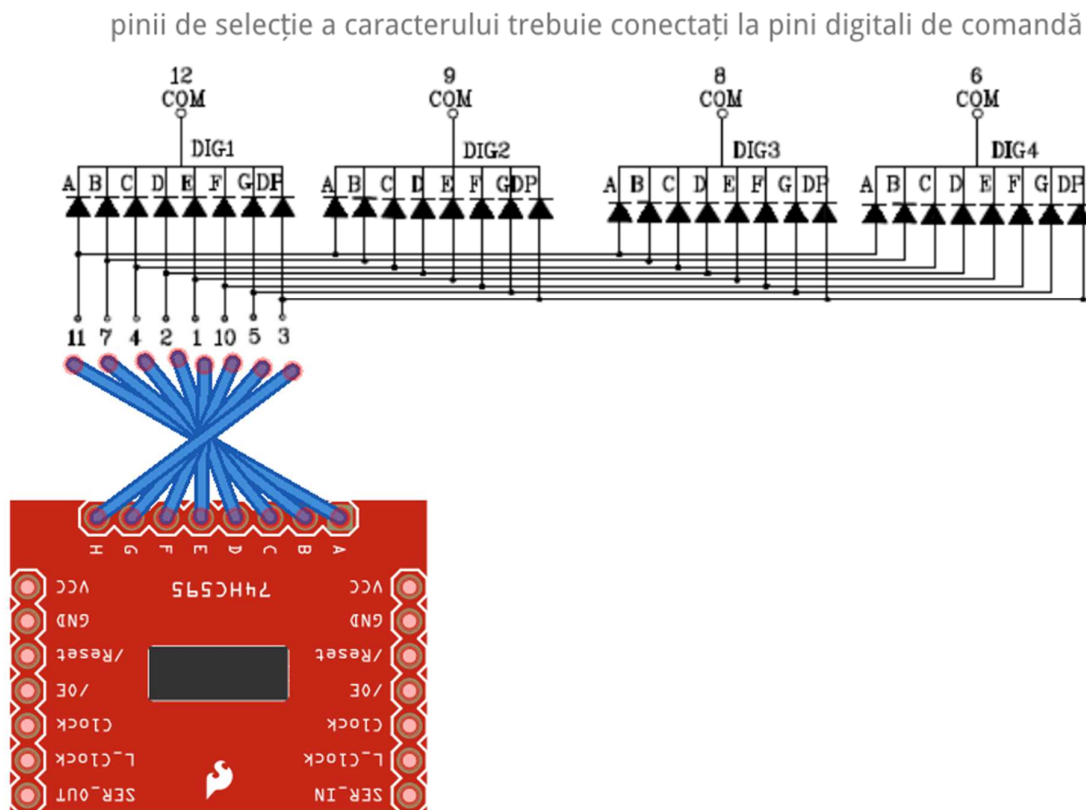
```

```

packetBuffer[13]  = 0x4E;
packetBuffer[14]  = 49;
packetBuffer[15]  = 52;
Udp.beginPacket(address, 123);
Udp.write(packetBuffer, NTP_PACKET_SIZE);
Udp.endPacket();
}

```

Dacă se dorește implementarea unui afișaj pe 7 segmente utilizând o componentă proprie (un afișaj de mari dimensiuni de exemplu (6)) se poate utiliza un circuit de tip shift register (7) cu o schemă de interconectare de genul:



Pentru circuitul 74HC595 se poate avea în vedere utilizarea bibliotecii *SparkFun 74HC595 Arduino Library* (8) pentru implementarea comenzii către caracterele pe 7 segmente.

Referințe on-line

(1) Network Time Protocol

https://en.wikipedia.org/wiki/Network_Time_Protocol

(2) Feather M0 WiFi - ATSAM21 + ATWINC1500

https://www.robofun.ro/feather-m0-wifi-atsamd21-atwinc1500?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(3) Arduino MKR1000

https://www.robofun.ro/arduino-mkr1000?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(4) Grove - 4-Digit Display

<https://www.seeedstudio.com/grove-4digital-display-p-1198.html>

(5) GitHub - avishorp/TM1637: Arduino library for TM1637 (LED Driver)

<https://github.com/avishorp/TM1637>

(6) Display 7 segmente - 20mm Albastru

https://www.robofun.ro/display-7-segmente-20mm-albastru?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(7) Shift Register - 74HC595

https://www.robofun.ro/sparkfun-shift-register-breakout-74hc595?utm_source=newsletter&utm_medium=email&utm_content=productLink&utm_campaign=CURS_EMAIL

(8) SparkFun 74HC595 Arduino Library

https://github.com/sparkfun/SparkFun_74HC595_Arduino_Library/tree/V_1.1.0