

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.

Ceas NTP

Protocolul NTP (Network Time Protocol) este folosit de sistemele de calcul de uz general (PC-uri, servere, laptop-uri) pentru a-și sincroniza ceasul intern cu o sursă de încredere. Având în vedere creșterea posibilităților de interconectare a sistemelor embedded acest protocol a devenit accesibil și acestora. În această lecție vom prezenta o soluție de sincronizare a unui ceas cu o sursă NTP Internet. Pentru mai multe detalii legate de protocolul NTP puteți consulta:

What is NTP?

<http://www.ntp.org/ntpfaq/NTP-s-def.htm>

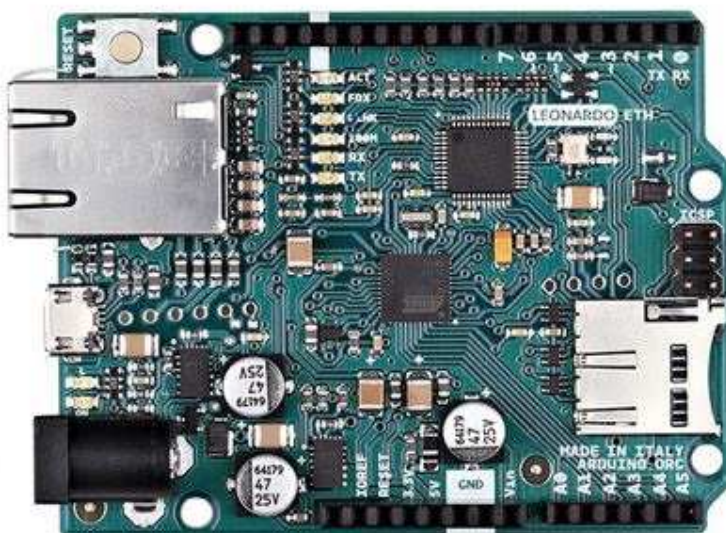
NTP - How does it work?

<http://www.ntp.org/ntpfaq/NTP-s-algo.htm>

Network Time Protocol

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

În cadrul lecției vom utiliza o placă Arduino Leonardo ETH – o versiune mai nouă a clasicii plăci Arduino Ethernet. Această placă este echipată cu un microcontroler ATmega32U4 (ce permite utilizarea plăcii fără a avea nevoie de programator FTDI) și un controler ethernet W5500 (față de W5100 de pe vechea placă Arduino Ethernet).



<https://www.robofun.ro/arduino/arduino-leonardo-eth>

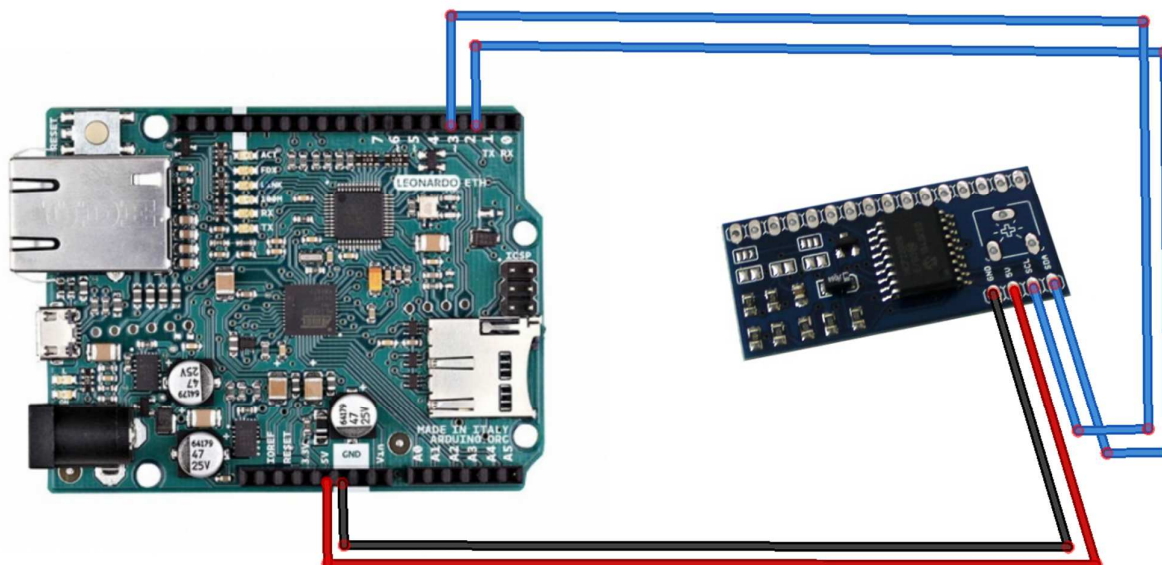
<https://www.robofun.ro/forum/>

Pentru afișare vom utiliza un ecran alfanumeric monocrom de dimensiuni generoase (20x4) cu magistrală I2C pentru simplificarea interconectării:



https://www.robofun.ro/lcd/lcd_20x4_i2c_negru_verde

Schema de interconectare între placa de dezvoltare și ecran este următoarea (SDA-D2, SCL-D3, 5V-5V, GND-GND):



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

Conectarea ecranului alfanumeric prin intermediul magistralei I2C necesită utilizarea unei versiuni modificate a bibliotecii LiquidCrystal:

https://github.com/adafruit/Adafruit_LiquidCrystal

```
#include "Wire.h"
#include "Adafruit_LiquidCrystal.h"
Adafruit_LiquidCrystal lcd(0);
```

Spre deosebire de alte metode de sincronizare a unui ceas electronic, sincronizarea NTP nu permite o interogare frecventă care să permită afișarea ceasului doar pe baza informațiilor primite de la serverul NTP – este o metodă de sincronizare a ceasului nu o sursă ce oferă ora și data curentă. Din acest motiv este necesar să implementăm un mecanism propriu de contorizare a timpului care să ofere informațiile necesare pentru afișarea orei și datei pe ecran – pentru acest lucru vom utiliza biblioteca Time:

<https://github.com/PaulStoffregen/Time>

```
#include <TimeLib.h>
const int timeZone = 3;
```

Biblioteca Ethernet inclusă în mediul Arduino IDE nu este compatibilă cu controlerul ethernet W5500 din acest motiv este necesară instalarea bibliotecii Ethernet2:

<https://github.com/adafruit/Ethernet2>

Placa Arduino Leonardo ETH este produsă exclusiv de arduino.org , biblioteca Ethernet2 este parte a versiunii mediului Arduino IDE dezvoltat de arduino.org dar este posibilă programarea acestei plăci și cu versiunea mediului Arduino IDE dezvoltată de arduino.cc: se selectează Arduino Leonardo și se instalează biblioteca Ethernet2.

```
#include <SPI.h>
#include <Ethernet2.h>
#include <EthernetUdp2.h>
byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02 };
unsigned int localPort = 8888;
// time-a.timefreq.bldrdoc.gov
IPAddress timeServer(132, 163, 4, 101);
```

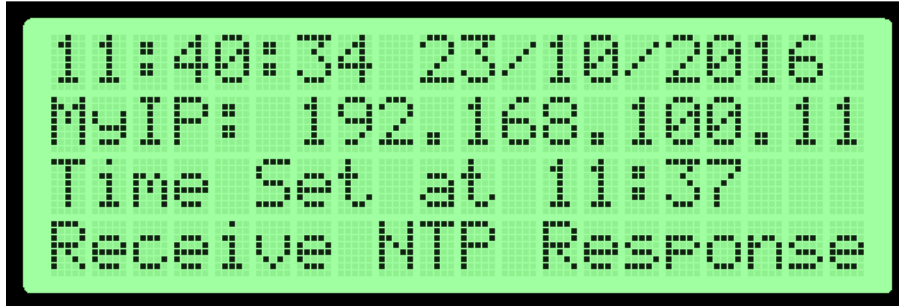
<https://www.robofun.ro/forum/>

```
// time-b.timefreq.bldrdoc.gov
// IPAddress timeServer(132, 163, 4, 102);
// time-c.timefreq.bldrdoc.gov
// IPAddress timeServer(132, 163, 4, 103);
EthernetUDP Udp;
```

În cadrul secțiunii setup() se realizează inițializarea ecranului și a conexiunii de rețea. Este necesară conectarea sistemului într-o rețea ce oferă servicii de configurare automată de tip DHCP.

```
void setup() {
  lcd.begin(20, 4);
  lcd.setBacklight(HIGH);
  lcd.setCursor(0, 1);
  byte n = 1;
  while(!Ethernet.begin(mac)) {
    lcd.print("DHCP failed... ");
    lcd.print(n++);
    lcd.setCursor(0, 1);
    delay(10000);
  }
  lcd.print("MyIP: ");
  for (byte thisByte = 0; thisByte < 4; thisByte++) {
    lcd.print(Ethernet.localIP()[thisByte], DEC);
    if(thisByte<3) lcd.print(".");
  }
  Udp.begin(localPort);
  setSyncProvider(getNtpTime);
}
```

În cadrul secțiunii loop() se realizează partea de afișare propriu-zisă pe ecran a informațiilor de tip oră și dată. Prima linie a afișajului va conține ora și data, a doua linie adresa IP a sistemului, a treia linie ora ultimei sincronizări reușite și a patra linie starea sincronizării NTP.



```
time_t prevDisplay = 0;
time_t lastsyncr;

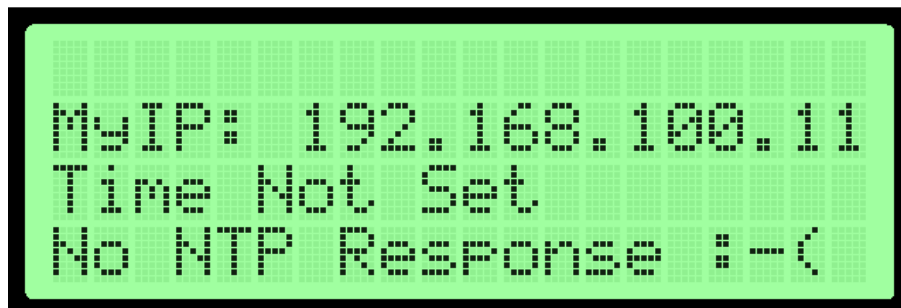
void loop() {
    lcd.setCursor(0,0);
    if (timeStatus() != timeNotSet) {
        if (now() != prevDisplay) {
            prevDisplay = now();
            digitalClockDisplay();
        }
    }
    lcd.setCursor(0,2);
    if(timeStatus()==timeNotSet) lcd.print("Time Not Set");
    if(timeStatus()==timeNeedsSync)
        lcd.print("Time Needs Sync");
    if(timeStatus()==timeSet) {
        lcd.print("Time Set at ");
        printDigits(hour(lastsyncr),false);
        printDigits(minute(lastsyncr),true); }
    delay(1000);
}
```

Procedurile `digitalClockDisplay()` și `prinDigits()` sunt utilizate în afișarea corectă a orei și datei.

```
void digitalClockDisplay(){
    printDigits(hour(),false);
    printDigits(minute(),true);
    printDigits(second(),true);
    lcd.print(" ");
    lcd.print(day());
    lcd.print("/");
    lcd.print(month());
    lcd.print("/");
    lcd.print(year());
}

void printDigits(int digits, boolean dots){
    if (dots) lcd.print(":");
    if(digits < 10)
        lcd.print('0');
    lcd.print(digits);
}
```

Procedurile `getNtpTime()` și `sendNTPpacket()` sunt utilizate de biblioteca `Time` pentru a utiliza protocolul NTP ca sursă de sincronizare. În cazul în care nu se reușește nici o sincronizare (server NTP indisponibil, probleme de conexiune Internet) sistemul nu va fi capabil să afișeze ora și data.



Atenție! În cazul în care sistemul nu are probleme de conexiune dar sincronizarea nu se poate realiza, se poate încerca o resetare completă a sistemului (deconectare alimentare, realimentare) deoarece este posibil ca problema să fie cauzată de o blocare a controlerului ethernet.

```
const int NTP_PACKET_SIZE = 48;
byte packetBuffer[NTP_PACKET_SIZE];

time_t getNtpTime()
{
    while (Udp.parsePacket() > 0) ;
    sendNTPpacket(timeServer);
    uint32_t beginWait = millis();
    while (millis() - beginWait < 1500) {
        int size = Udp.parsePacket();
        if (size >= NTP_PACKET_SIZE) {
            lcd.setCursor(0,3);
            lcd.print("Receive NTP Response");
            Udp.read(packetBuffer, NTP_PACKET_SIZE);
            unsigned long secsSince1900;
            secsSince1900 = (unsigned long)packetBuffer[40]
                                << 24;
            secsSince1900 |= (unsigned long)packetBuffer[41]
                                << 16;
            secsSince1900 |= (unsigned long)packetBuffer[42]
                                << 8;
            secsSince1900 |= (unsigned long)packetBuffer[43];
            lastsyncr = (time_t) (secsSince1900 - 2208988800UL
                                + timeZone * SECS_PER_HOUR);
            return lastsyncr;
        }
    }
    lcd.setCursor(0,3);
```



```

    lcd.print("No NTP Response :-( ");
    return 0;
}

void 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();
}

```

Exemple de alte proiecte de tip ceas cu sincronizare NTP bazate pe platforma Arduino:

Arduino Internet Time Client

<http://www.instructables.com/id/Arduino-Internet-Time-Client/>

Network time synchronized clock for Arduino

<http://www.instructables.com/id/Internet-time-synchronized-clock-for-Arduino/>

Arduino Internet Clock

<http://blog.thiseldo.co.uk/?p=336>

A simple arduino clock using ethernet shield and ntp for beginners

<http://blog.riyas.org/2014/03/a-simple-arduino-led-clock-using-ethernet-ntp-for-beginners.html>

Garbage Mate – Version 2

<http://www.wabbitwanch.net/blog/?p=744>