

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs  
CC BY-NC-ND



Codul sursa din acest document este licentiat

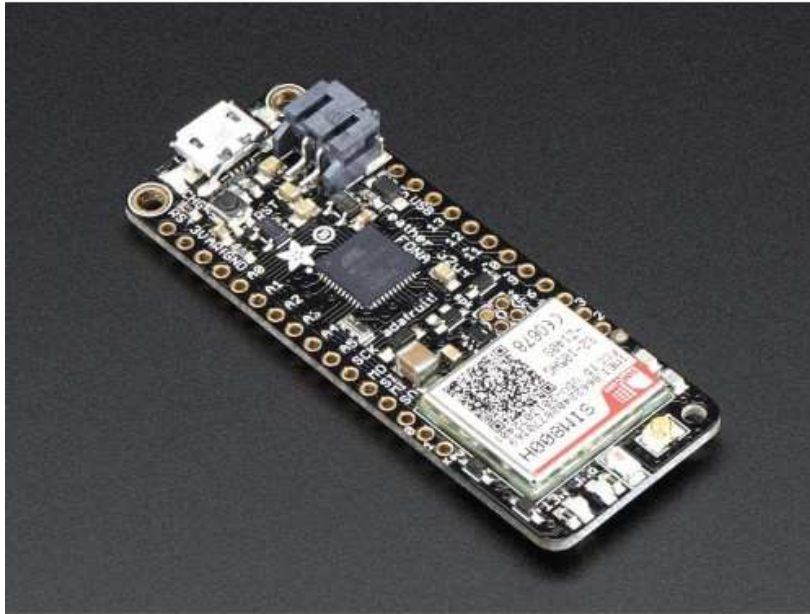
Public-Domain

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 cu ecran OLED și sincronizare GSM

O problemă des întâlnită la ceasuri este acuratețea cu care indică ora. Câți dintre noi au răbdare să potrivească periodic ceasurile de perete astfel încât să nu arate cu câteva minute / zeci de minute înainte sau mai târziu? Există mai multe soluții de sincronizare automată a ceasurilor electronice dar în lecția de față vom utiliza o metodă mai rar întâlnită pentru ceasurile obișnuite: sincronizarea automată dintr-o rețea de telefonie mobilă GSM.

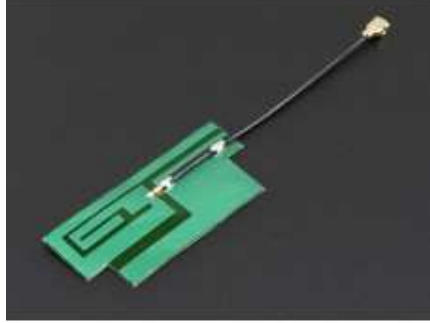
Pentru realizare vom utiliza o placă de dezvoltare Adafruit Feather 32U4 FONA ce include un microcontroler ATmega32U4 (la fel ca și Arduino Leonardo sau alte plăci Feather) și un controler GSM SIM800H (Quad-band 850/900/1800/1900MHz).



<https://www.robofun.ro/feather-32u4-fona>

Pentru utilizarea acestei plăci avem nevoie și de o antenă GSM uFL, un acumulator LiPo 3.7V de minim 500mAh și de un SIM GSM 2G.

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



<https://www.robofun.ro/accesorii/accesorii-antene/slim-sticker-type-gsm-cellular-quad-band-antenna-3dbi-ufi>



[https://www.robofun.ro/surse\\_de\\_alimentare/acumulatori/Acumulator-LiPo-3.7v-800mAh](https://www.robofun.ro/surse_de_alimentare/acumulatori/Acumulator-LiPo-3.7v-800mAh)

Pentru mai multe informații despre funcționarea plăcii Adafruit Feather 32U4 FONA puteți consulta și materialul:

Adafruit Feather 32u4 FONA – Take your Feather anywhere in the world

<https://learn.adafruit.com/adafruit-feather-32u4-fona/>

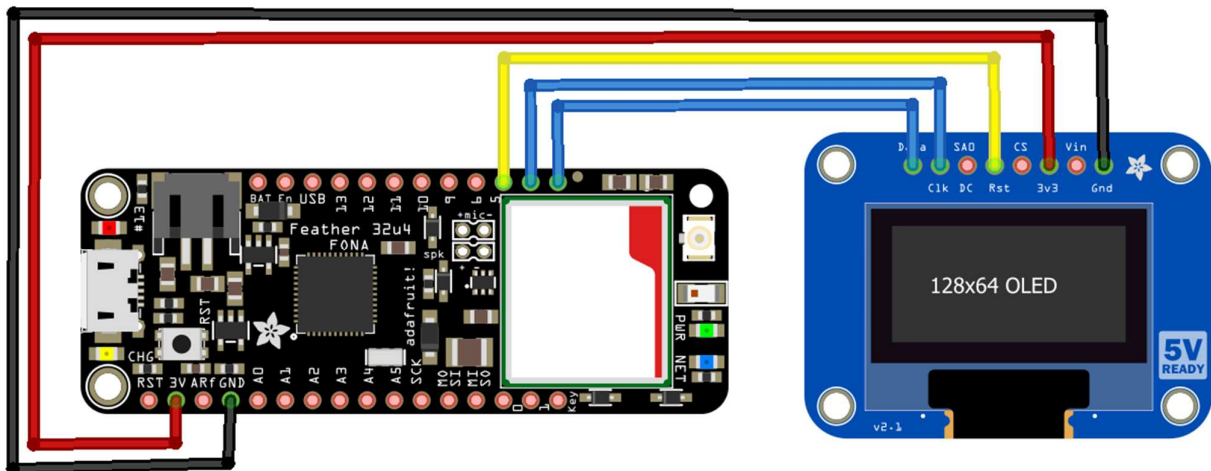
Pentru partea de afișare vom utiliza un afișaj OLED monocrom 1.3” 128x64 pixeli cu conexiune I2C:

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



<https://www.robofun.ro/lcd/monochrome-1-3-128x64-oled-graphic-display>

Schema de interconectare între placa de dezvoltare și ecranul OLED este tipică unei conexiuni I2C (pin Data - pin 2 SDA, pin Clk – pin 3 SCL, pin Rst – pin 5, pin 3v3 – pin 3V, pin GND – pin GND):



Programul necesită instalarea mai multor biblioteci externe mediului de dezvoltare Arduino IDE:

Adafruit FONA – pentru lucrul cu modulul GSM

[https://github.com/adafruit/Adafruit\\_FONA](https://github.com/adafruit/Adafruit_FONA)

Adafruit GFX și Adafruit SSD1306 – pentru lucrul cu ecranul OLED

<https://github.com/adafruit/Adafruit-GFX-Library>

[https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306)

Scopul sistemului este de a prelua data și ora din rețeaua GSM și de o afișa pe ecranul OLED. **Se poate utiliza o cartelă PrePay și nu este nevoie să aibă credit activ – se poate utiliza o cartelă aflată în perioada de grație.** Pe lângă oră și dată pe ecranul sistemului se va afișa și procentul de încărcare a acumulatorului (stânga sus) și Network Status / puterea semnalului GSM (dreapta sus):



În cadrul programului se vor inițializa într-o primă fază obiectele de lucru cu modulul GSM și cu ecranul OLED:

```
#include "Adafruit_FONA.h"
#define FONA_RX 9
#define FONA_TX 8
#define FONA_RST 4
#define FONA_RI 7
char replybuffer[255];
#include <SoftwareSerial.h>
SoftwareSerial fonaSS = SoftwareSerial(FONA_TX, FONA_RX);
SoftwareSerial *fonaSerial = &fonaSS;
Adafruit_FONA fona = Adafruit_FONA(FONA_RST);
uint8_t type;

#include <Wire.h>
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
#define OLED_RESET 5
Adafruit_SSD1306 display(OLED_RESET);
```

Secțiunea setup() cuprinde partea de inițializare a ecranului OLED și a modulului GSM (inclusiv deblocare cartelă SIM, activare sincronizare ceas din rețea).

```
void setup() {

    display.begin(SSD1306_SWITCHCAPVCC, 0x3D);
    display.display();
    delay(2000);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);
    display.println("Initializing...");
    display.display();
    delay(1);

    fonaSerial->begin(4800);
    if (! fona.begin(*fonaSerial)) {
        display.println(F("Couldn't find FONA"));
        display.display();
        delay(1);
        while (1);
    }
    type = fona.type();
    display.println(F("FONA is OK"));
    display.print(F("Found "));
    switch (type) {
        case FONA800L:
```

```

        display.println(F("FONA 800L")); break;
case FONA800H:
        display.println(F("FONA 800H")); break;
case FONA808_V1:
        display.println(F("FONA 808 (v1)")); break;
case FONA808_V2:
        display.println(F("FONA 808 (v2)")); break;
case FONA3G_A:
        display.println(F("FONA 3G (American)")); break;
case FONA3G_E:
        display.println(F("FONA 3G (European)")); break;
default:
        display.println(F("???")); break;
}
display.display();
delay(1);
char imei[15] = {0};
uint8_t imeiLen = fona.getIMEI(imei);
if (imeiLen > 0) {
    display.print("Module IMEI: "); display.println(imei);
    display.display();
    delay(1);
}

```

**Atenție!!!** Codul de deblocare a cartelei SIM GSM trebuie modificat în program în concordanță cu pinul cartelei utilizate.

```

if (! fona.unlockSIM("0000")) {
    display.println(F("Failed"));
} else {
    display.println(F("PIN OK!"));
}

```

```

display.display();
delay(1);
fona.getSIMCCID(replybuffer);
display.print(F("SIM CCID = "));
display.println(replybuffer);
display.display();
delay(1);
if (!fona.enableNetworkTimeSync(true)) {
    display.println(F("Failed to enable"));
    display.display();
    delay(1);
}
delay(5000);
}

```

Secțiunea loop() se ocupă cu citirea din modulul GSM și afișare informațiilor enumerate anterior: procent încărcare acumulator, stare rețea, putere semnal rețea, oră și dată calendaristică.

```

void loop() {
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(0,0);

    uint16_t vbat;
    if (fona.getBattPercent(&vbat)) {
        display.print(vbat);
        display.print(F("          "));
    }

    uint8_t n = fona.getNetworkStatus();
    if (n == 0) display.print(F("N ")); //Not register

```



```

if (n == 1) display.print(F("H ")); //Register (home)
if (n == 2) display.print(F("S ")); //Searching
if (n == 3) display.print(F("D ")); //Denied
if (n == 4) display.print(F("U ")); //Unknown
if (n == 5) display.print(F("R ")); //Register (roamning)

n = fona.getRSSI();
int8_t r;
if (n == 0) r = -115;
if (n == 1) r = -111;
if (n == 31) r = -52;
if ((n >= 2) && (n <= 30))
    { r = map(n, 2, 30, -110, -54); }
display.print(r); display.println(F("dBm"));

display.setTextSize(3);
display.setCursor(20,20);

char buffer[23];
char temp[10];
fona.getTime(buffer, 23);
memcpy(temp,&buffer[10],5);
temp[5]='\0';
display.println(temp);

display.setTextSize(2);
display.setCursor(17,50);

memcpy(temp,&buffer[1],8);
temp[8]='\0';
display.println(temp);

```

```
display.display();  
delay(1);  
  
while (fona.available()) {  
    fona.read();  
}  
delay(10000);  
}
```

Programul a fost dezvoltat și testat cu Arduino IDE 1.6.9, Adafruit AVR Boards 1.4.9, Adafruit FONA Library 1.3.2, Adafruit GFX Library 1.1.5 și Adafruit SSD1306 1.1.0.

Proiectul prezentat poate să stea la baza realizării unui sistem mai complicat de tip telefon mobil. Câteva exemple de astfel de proiecte:

ArduinoPhone

<http://www.instructables.com/id/ArduinoPhone/>

Arduin-o-Phone

<https://learn.adafruit.com/arduin-o-phone-arduino-powered-diy-cellphone/>

My open-source, do-it-yourself cellphone (built with Arduino)

<https://blog.arduino.cc/2013/08/12/diy-cellphone/>

Cell phone text using an Arduino

<http://duino4projects.com/cell-phone-text-using-an-arduino/>