

Lucrarea nr. 4

Utilizarea interfeței I2C / TWI

Obiective:

- Studenții vor lucra cu display-ul alfanumeric 4x20, pe care vor afișa ora și data preluare de la un modul RTC Adafruit.

Hardware necesar:

- Arduino UNO R3,
- Modul timp real RTC cu DS1307; <https://www.arduino-libraries.info/libraries/ds1307-rtc>
- Display I2C LCD 4x20; Biblioteca pentru Arduino poate fi descărcată de aici: <https://github.com/Mcarli201/I2C-Midas-20x4>

Pasul 1:

Puteți refolosi proiectul Proteus de la laboratorul anterior, (Button) sau puteți crea unul nou. Schema de conexiuni pentru lucrare este prezentată în figura de mai jos:

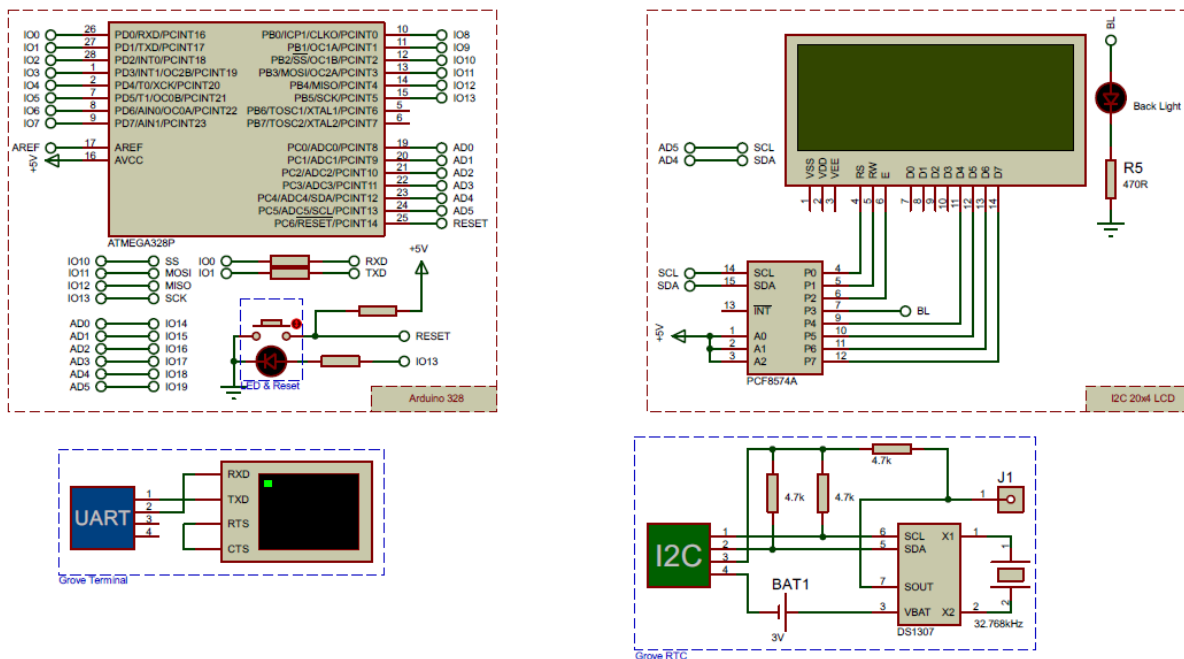


Figura 1: Schema electrică generală, realizată în Proteus.

Se pot utiliza cele 2 Sample Project din Proteus, și anume: "Arduino I2C 4x20 LCD" și "Grove RTC using DS1307".

Ambele device-uri folosesc I2C.

Interfața I²C sau IIC (*Inter-Integrated Circuit*) este un tip de magistrală pentru transmisie de date serială master-slave, utilizată intensiv între circuite integrate digitale (microcontrolere, memorii,

convertoare) și a fost inițial dezvoltată de către firma olandeză Philips (în prezent NXP) în anul 1982.

Magistrala (busul) I²C este compusă din două linii:

- SCL - Serial Clock Line (ro: linie de ceas seriala)
- SDA - Serial Data Line (ro: linie de date seriala)

(ro.wikipedia.org/i2c)

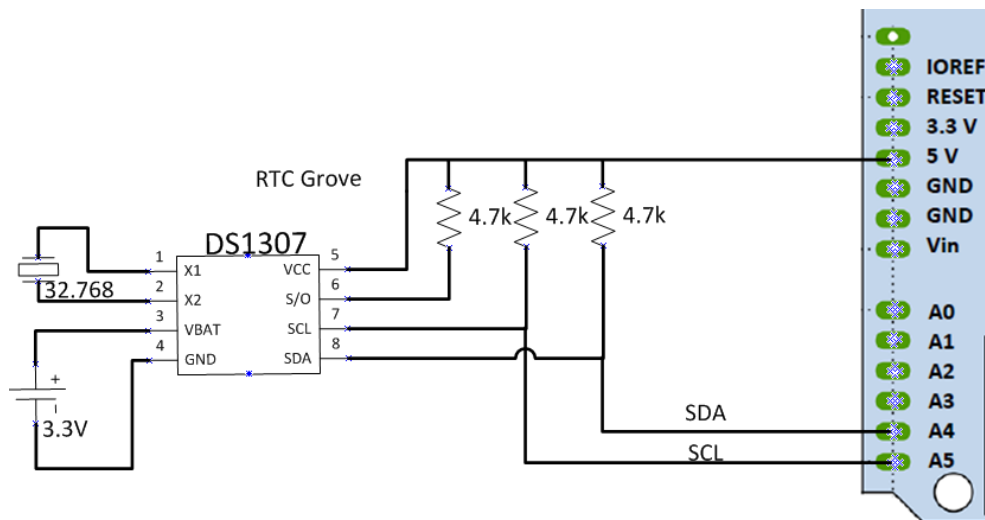


Figura 2: Conectarea liniilor SDA și SCL prin rezistențe de pull-up.

Cele două rezistențe sunt obligatorii, și pot lua valori între 2K și 10K, de obicei se folosește valoarea medie de 4k7.

Liniile SCL și SDA ale Arduino sunt de 5V iar cele ale senzorilor de 3.3V, deci, dacă le-am conecta direct am arde senzorul.

Toate dispozitivele I2c , sau TWI, sunt identificate printr-o adresă hexa, care trebuie identificată înainte de a realiza transmisia de date. Lucrul acesta se poate face cu aplicația i2c_scanner din Arduino IDE: Examples > Wire > i2c_scanner. Valorile adreselor sunt vizualizate cu ajutorul Serial Monitor, și arată ca în figura 3:

```

COM3
Scanning...
I2C device found at address 0x3C !
done

Scanning...
I2C device found at address 0x3C !
I2C device found at address 0x68 !
done
    
```

Figura 3: Adresele dispozitivelor I2C.

Pasul 2: Testarea LCD I2C.

Display-ul folosește o bibliotecă specială, care trebuie încărcată în Arduino IDE de la adresa <https://github.com/Mcarli201/I2C-Midas-20x4>.

Se descarcă arhiva, în Arduino IDE > Sketch > Include Library > Add Zip Library.

Se caută arhiva salvată anterior și se încarcă.

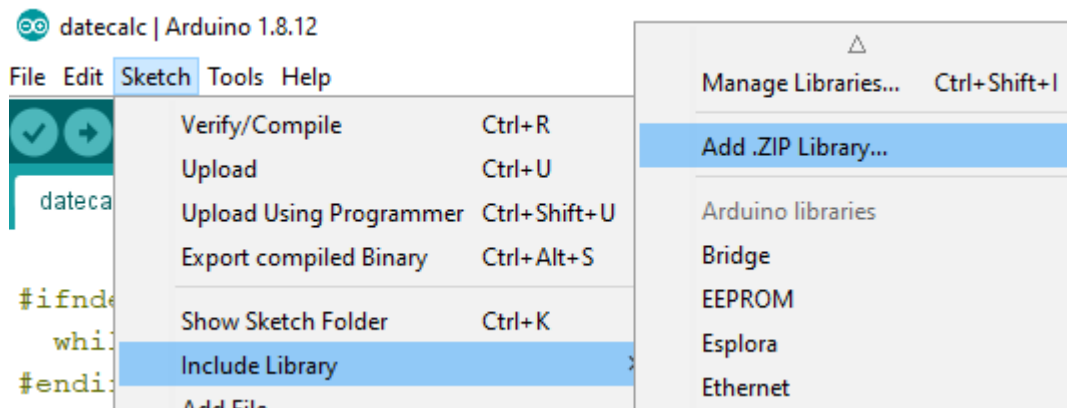


Figura 4: Adăugarea unei librării în Arduino IDE.

Apoi din Examples se deschide fișierul care exemplifică funcționarea LCD-ului, se rulează și se încarcă în aplicația Proteus pentru verificare. Trebuie să verificăm mai întâi adresa hexa a LCD-ului.

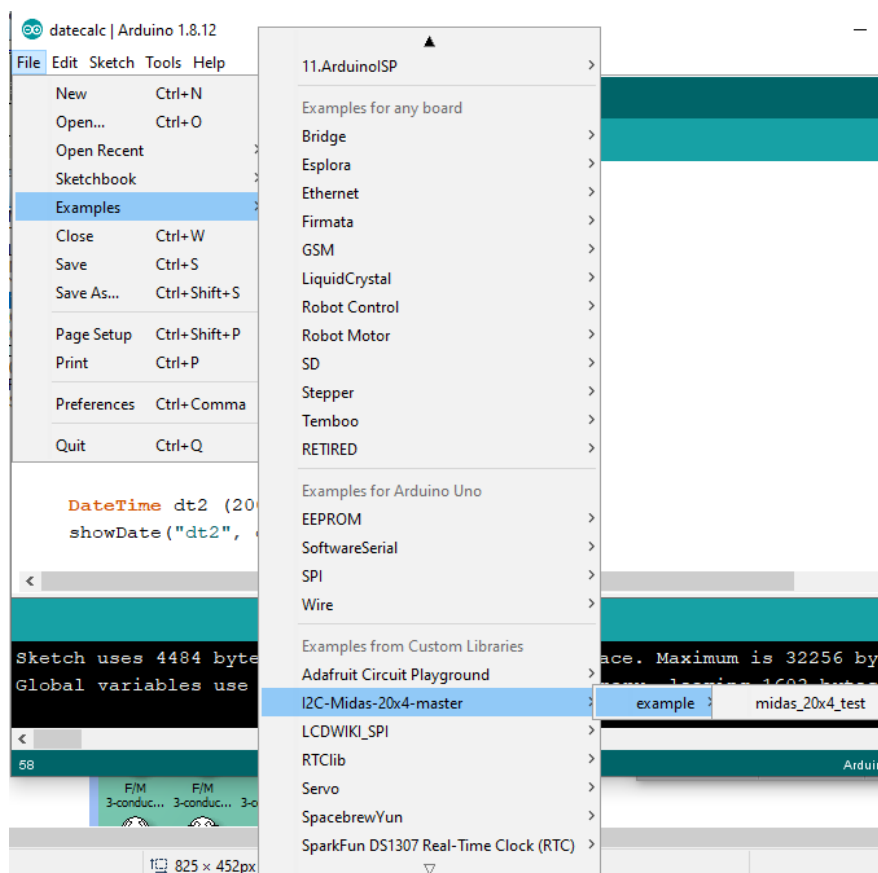


Figura 5: Exemplul pentru LCD i2c.

Pasul 3: Testare modul RTC

ATENȚIE: Pentru cei care fac proiectul pe hardware, trebuie să identifice corect cipul/producătorul modulului RTC și să încarce biblioteca dată de dezvoltatorul produsului.

Trebuie adăugată în Arduino IDE biblioteca corespunzătoare modulului RTC folosit. În cazul de față de la adresa: <https://www.arduino-libraries.info/libraries/ds1307-rtc>.

Se testează funcționarea RTC, utilizând exemplul din Arduino IDE: Examples → SparkFun... → DS1307_RTC_Demo:

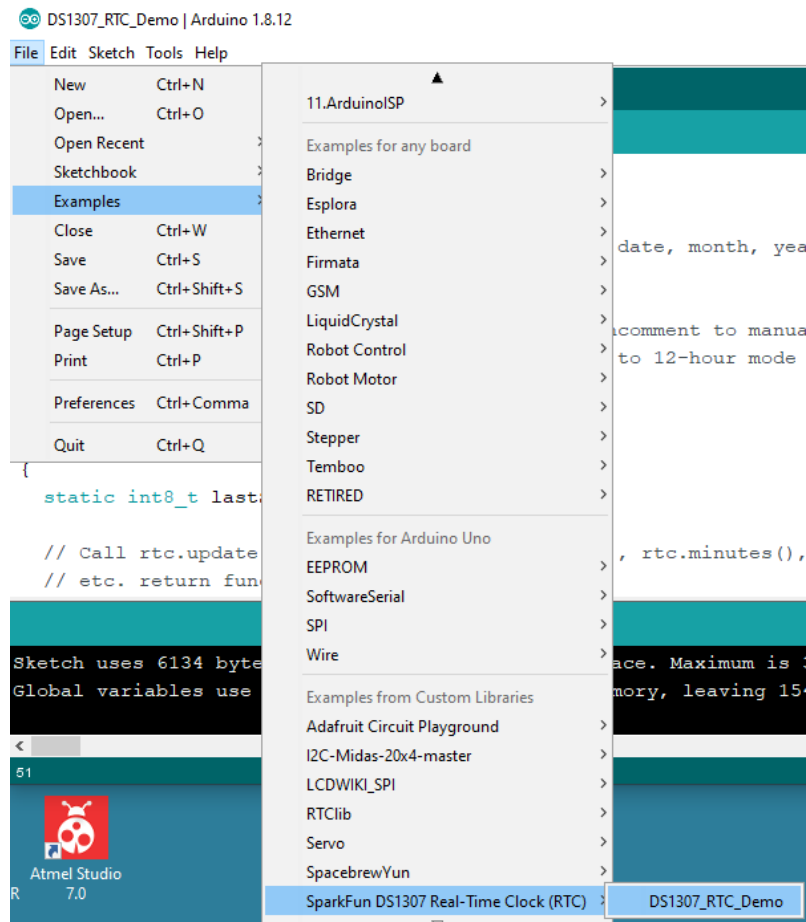


Figura 6: Exemplul "DS1307_RTC_Demo" din Arduino IDE.

Rezultatul se citește pe portul serial, și arată ca în figura de mai jos:

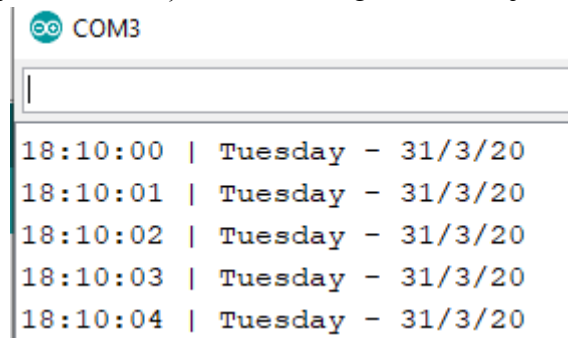


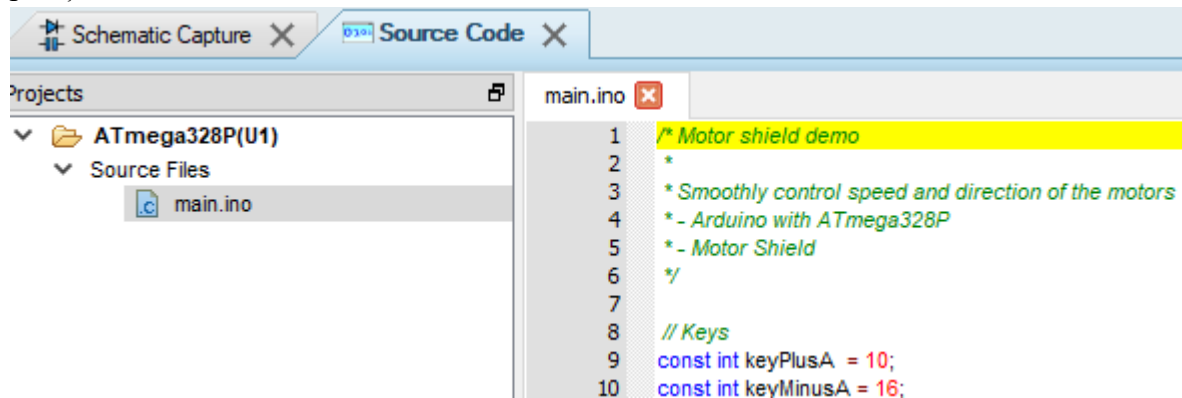
Figura 7: Funcționarea modulului RTC.

Pasul 4:

Combinați cele două exemple de mai sus, astfel încât:

1. pe display-ul LCD i2c să se afișeze ora, și data.
2. Modificați programul astfel încât zilele săptămânii să fie scrise în română, iar data să fie în formatul românesc, și anume: zz:ll:aaaa;
3. Pentru cei pasionați și avansați: adăugați 4 butoane cu ajutorul cărora să puteți regla manual data și ora. Rolul butoanelor va fi:
 - a. Meniu
 - b. Up
 - c. Down
 - d. Set

OBS: pentru butoane, puteți folosi exemplul Arduino Motor Shield din Proteus, unde sunt prezente 6 butoane care controlează viteza up/down a 2 motoare de cc. Codul simulării îl puteți accesa din tabul Source Code:



```
1  /* Motor shield demo
2  *
3  * Smoothly control speed and direction of the motors
4  * - Arduino with ATmega328P
5  * - Motor Shield
6  */
7
8  // Keys
9  const int keyPlusA = 10;
10 const int keyMinusA = 16;
```

Figura 8: Codul sursă al un exemplu de program simulat în Proteus.