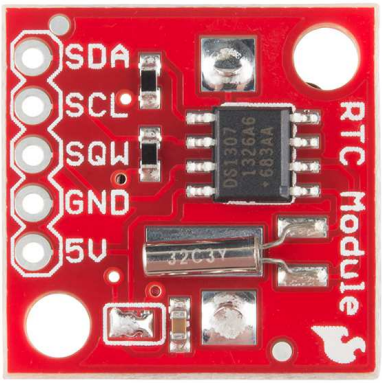
Ceas/calendar cu termometru și comandă prin bluetooth

Chiar dacă la prima vedere realizarea unui ceas electronic pare o sarcină simplă există anumite aspecte practice reale ce îngreunează implementarea unui astfel de sistem.

Prima problemă pe care o ridică implementarea unui ceas este contorizarea timpului: la nivel de secundă (dependentă de acuratețea ceasului intern al microcontrolerului) și la nivel de minut, oră, zi, săptămână, lună, an (număr diferit de zile ale săptămânii, calculul zilei săptămânii, an bisect etc.). Pentru a simplifica acest aspect vom utiliza în cele ce urmează un circuit specializat RTC (Real Time Clock) DS1307 ce se va ocupa în mod precis de ambele aspecte de contorizare.



https://www.robofun.ro/module/module-rtc/real_time_clock

Un al doilea aspect delicat este interfața utilizator de comandă prin care se ”potrivește ceasul” – în mod tradițional implementată de un set de butoane pentru setarea orei, datei, alarmei etc. În cadrul unui proiect DIY butoanele sunt simplu de conectat la o placă de dezvoltare dar ridică dificultăți de ordin mecanic și estetic când vine vorba de montat într-o carcasă. Din acest motiv proiectul de față nu va utiliza butoane pentru interacțiunea cu sistemul ci un modul bluetooth prin care se vor recepționa comenzi echivalente. Modulul utilizat este Bluetooth Mate Silver dar se poate utiliza orice modul bluetooth serial:

https://www.robofun.ro/wireless/wireless-bluetooth/bluetooth_arduino

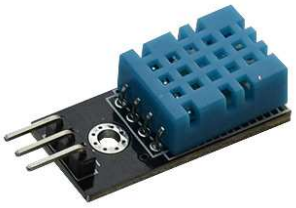
Ca și placă de dezvoltare vom utiliza o placă Arduino Uno (sau echivalentă) și pentru afișare un clasic afișaj LCD alfanumeric monocrom 16x2:

https://www.robofun.ro/arduino/arduino_uno_v3

https://www.robofun.ro/lcd/lcd_16x2_negru_verde

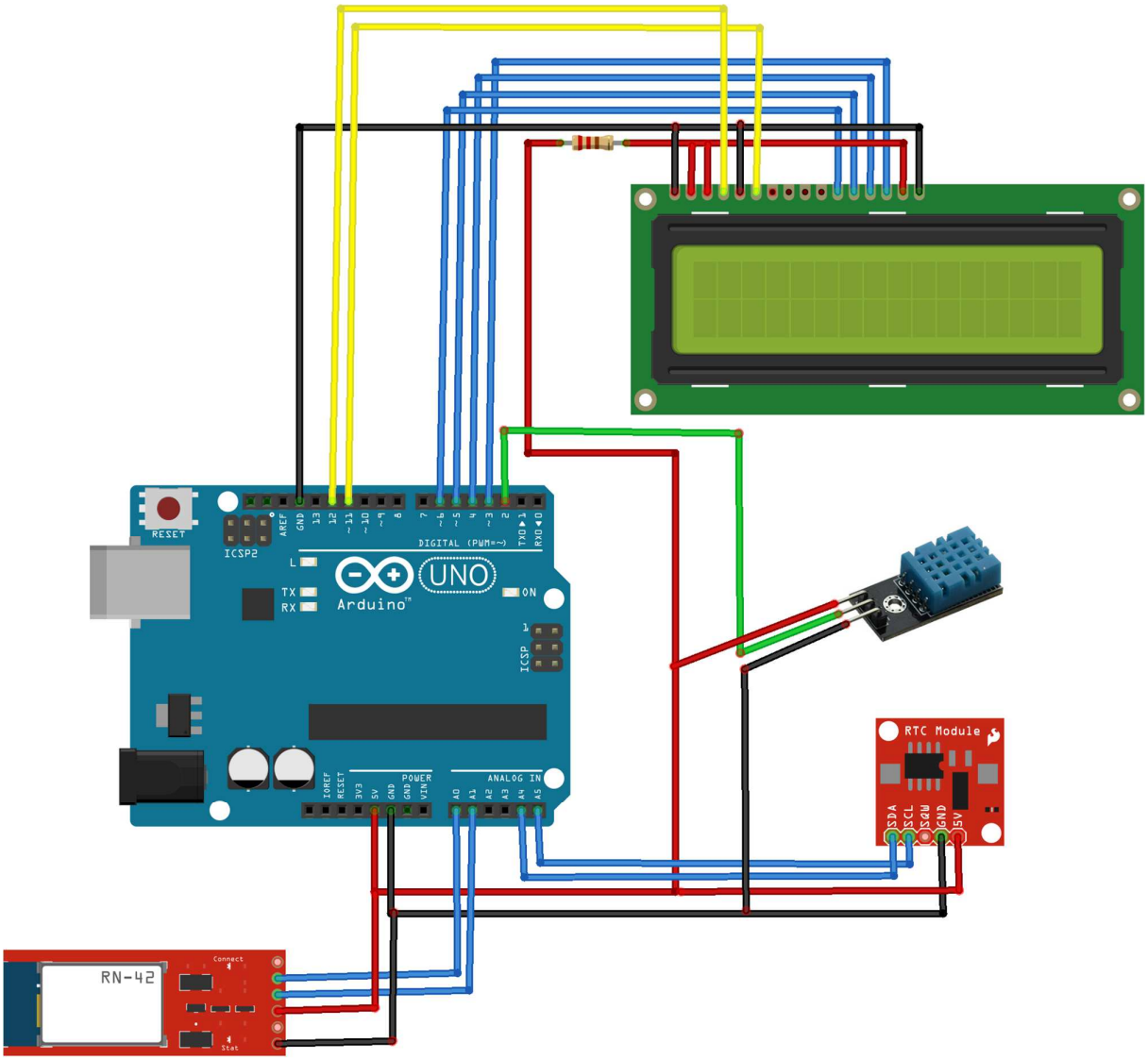
Pentru a diversifica funcționalitatea sistemului, pe lână afișarea orei, a zilei săptămânii

și a datei calendaristice, vom implementa și afișarea temperaturii și umidității mediului ambiental utilizând un senzor digital SNS-DH11 (sau DHT11):



https://www.robofun.ro/senzori/vreme/senzor-temperatura-umiditate-sns-dh11

Schema de interconectare a componentelor este următoarea:



LCD-ul se interconectează cu placa de dezvoltare utilizând patru linii de date și două

linii de comandă (pinii D3, D4, D5, D5, D11, D12) și are lumina de fundal aprinsă permanent și contrastul la maxim:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,6,5,4,3);
```

Dacă se dorește simplificarea conectării afișajului la placa de dezvoltare se poate opta pentru un modul I2C:

https://www.robofun.ro/lcd/lcd-16x2-i2c-negru-verde

Modulul RTC se conectează pe magistrala I2C a plăcii de dezvoltare, pinii A4 și A5, și are adresa fixă 0x68.

```
#include <Wire.h>
#define DS1307_I2C_ADDRESS 0x68
```

Modulul bluetooth se conectează la pinii A1 și A0 deoarece se utilizează biblioteca SoftwareSerial ce permite utilizarea a oricăror doi pini ca port de comunicație serială.

```
#include "SoftwareSerial.h";
int bluetoothTx = A1;
int bluetoothRx = A0;
SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);
```

Senzorul de temperatură și umiditate se conectează la pinul D2 al plăcii de dezvoltare și necesită utilizarea bibliotecii SimpleDHT:

https://github.com/winlinvip/SimpleDHT

```
#include <SimpleDHT.h>
int pinDHT11 = 2;
SimpleDHT11 dht11;
```

Programul folosește două variabile globale pentru controlul afișării: sec_afis – număr

de secunde ce permite alternarea afișării temperaturii / umidității – datorită numărului limitat de caractere acestea sunt afișate alternativ la 10 secunde; afis – permite reîmprospătarea ecranului nu mai des de o secundă.

```
volatile int sec_afis=1;
volatile boolean afis=true;
```

În cadrul secțiunii setup() se inițializează componentele din sistem și timerul 1 intern ce va fi utilizat în incrementarea variabilei sec_afis.

```
void setup() {
  bluetooth.begin(9600);
  digitalWrite(9,LOW);
  Wire.begin();
  lcd.begin(16, 2);
  lcd.noCursor();
  lcd.clear();
  TIMSK1 = 0x01;
  TCCR1A = 0x00;
  TCNT1 = 0x0BDC;
  TCCR1B = 0x04;
}
```

În cadrul secțiunii loop() se preiau de la modulul RTC valorile de oră și dată (getDateDs1307)

```
void loop(){
  byte temp, hum;
  byte second, minute, hour, dayOfWeek, dayOfMonth, month,
                      year;
  getDateDs1307(&second, &minute, &hour, &dayOfWeek,
                    &dayOfMonth, &month, &year);
```

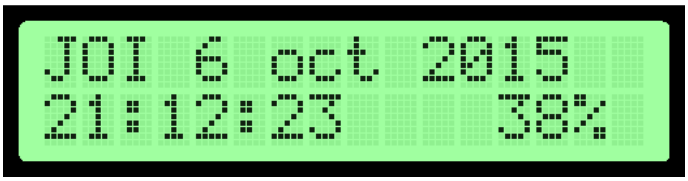
și apoi se verifică dacă se primește comandă pe conexiunea bluetooth – comenzile se pot trimite cu orice program de tip terminal bluetooth, se trimite mai întâi un caracter care să indice ce valoare dorim să modificăm (y – an, m – lună, d – zi, w – zi a

săptămânii, h – oră, t – minut) și după aceia + sau – pentru incrementare sau decrementare.

```
if(bluetooth.available()) {
  char comanda = (char)bluetooth.read();
  switch (comanda) {
  case 'y':
    while (bluetooth.available()==0) { delay(10); }
    comanda = (char)bluetooth.read();
    if (comanda=='+') year++;
    else if (comanda=='-') year--;
    setDateDs1307(second, minute, hour, dayOfWeek,
              dayOfMonth, month, year);
    break;
  case 'm':
    while (bluetooth.available()==0) { delay(10); }
    comanda = (char)bluetooth.read();
    if (comanda=='+') month++;
    else if (comanda=='-') month--;
    setDateDs1307(second, minute, hour, dayOfWeek,
              dayOfMonth, month, year);
    break;
  case 'd':
    while (bluetooth.available()==0) { delay(10); }
    comanda = (char)bluetooth.read();
    if (comanda=='+') dayOfMonth++;
    else if (comanda=='-') dayOfMonth--;
    setDateDs1307(second, minute, hour, dayOfWeek,
                      dayOfMonth, month, year);
    break;
 case 'w':
    while (bluetooth.available()==0) { delay(10); }
```

```
      comanda = (char)bluetooth.read();
      if (comanda=='+') dayOfWeek++;
      else if (comanda=='-') dayOfWeek--;
      setDateDs1307(second, minute, hour, dayOfWeek,
                    dayOfMonth, month, year);
      break;
    case 'h':
      while (bluetooth.available()==0) { delay(10); }
      comanda = (char)bluetooth.read();
      if (comanda=='+') hour++;
      else if (comanda=='-') hour--;
      setDateDs1307(second, minute, hour, dayOfWeek,
                    dayOfMonth, month, year);
      break;
    case 't':
      while (bluetooth.available()==0) { delay(10); }
      comanda = (char)bluetooth.read();
      if (comanda=='+') minute++;
      else if (comanda=='-') minute--;
      setDateDs1307(second, minute, hour, dayOfWeek,
                    dayOfMonth, month, year);
      break;
  }}
```

Următoarea parte din secțiunea loop() se ocupă cu partea de afișare efectivă a informațiilor pe LCD – rezultatul fiind ca cel din captura de mai jos.



```
lcd.setCursor(0,0);
switch (dayOfWeek) {
 case 1:
   lcd.print("LUN");
   break;
 case 2:
   lcd.print("MAR");
   break;
 case 3:
   lcd.print("MIE");
   break;
 case 4:
   lcd.print("JOI");
   break;
 case 5:
   lcd.print("VIN");
   break;
 case 6:
   lcd.print("SAM");
   break;
 case 7:
   lcd.print("DUM");
   break;
}
lcd.print(" ");
lcd.print(dayOfMonth, DEC);
lcd.print(" ");
switch (month) {
 case 1:
   lcd.print("ian");
   break;
 case 2:
```

```
  lcd.print("feb");
  break;
case 3:
  lcd.print("mar");
  break;
case 4:
  lcd.print("apr");
  break;
case 5:
  lcd.print("mai");
  break;
case 6:
  lcd.print("iun");
  break;
case 7:
  lcd.print("iul");
  break;
case 8:
  lcd.print("aug");
  break;
case 9:
  lcd.print("sep");
  break;
case 10:
  lcd.print("oct");
  break;
case 11:
  lcd.print("noi");
  break;
case 12:
  lcd.print("dec");
  break;
```

```
}
lcd.print(" 20");
lcd.print(year, DEC);
lcd.setCursor(0,1);
if (hour<10) lcd.print(" ");
lcd.print(hour, DEC);
lcd.print(":");
if (minute<10) lcd.print("0");
lcd.print(minute, DEC);
lcd.print(":");
if (second<10) lcd.print("0");
lcd.print(second, DEC);
lcd.print("   ");
   }
if (((sec_afis%10)==0) && afis)
{
  byte chk = dht11.read(pinDHT11, &temp, &hum, NULL);
  lcd.setCursor(12,1);
  switch(chk)
  {
      case 0:
      if (sec_afis==10) {
      lcd.print(temp);
      lcd.print(char(223));
      lcd.print("C");
      }
      else
      {
       lcd.print(" ");
       lcd.print(hum);
       lcd.print("%");
      }
```

```
        break;
      default:
        lcd.print("Error");
        break;
  }
   afis=false;
  }
}
```

Subrutina de ceas se ocupă cu incrementarea variabilei sec_afis și cu alternarea (true/false) conținutului variabilei afis, valoarea maximă a variabilei sec_afis este 21 deoarece se doreste afișarea combinată (10 secunde + 10 secunde) a celor două valori: temperatură și umiditate.

```
ISR(TIMER1_OVF_vect) {
  TCNT1=0x0BDC;
  sec_afis++;
  afis=true;
  if (sec_afis==21) sec_afis=1;
}
```

Ultimele proceduri din program sunt necesare pentru comunicația cu modulul RTC – proceduri de preluare și modificare a valorilor interne ale modulului RTC.

```
byte bcdToDec(byte val) { return ( (val/16*10) + (val%16)
); }

byte decToBcd(byte val) { return ((val/10*16) + (val%10));
}

void getDateDs1307(byte *second,
          byte *minute,
          byte *hour,
          byte *dayOfWeek,
```

```
          byte *dayOfMonth,
          byte *month,
          byte *year)
{
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  Wire.write(0);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
  *second     = bcdToDec(Wire.read() & 0x7f);
  *minute     = bcdToDec(Wire.read());
  *hour       = bcdToDec(Wire.read() & 0x3f);
  *dayOfWeek  = bcdToDec(Wire.read());
  *dayOfMonth = bcdToDec(Wire.read());
  *month      = bcdToDec(Wire.read());
  *year       = bcdToDec(Wire.read());
}

void setDateDs1307(byte second,
          byte minute,
          byte hour,
          byte dayOfWeek,
          byte dayOfMonth,
          byte month,
          byte year)
 {
      Wire.beginTransmission(DS1307_I2C_ADDRESS);
      Wire.write(byte(0));
      Wire.write(decToBcd(second));
      Wire.write(decToBcd(minute));
      Wire.write(decToBcd(hour));
      Wire.write(decToBcd(dayOfWeek));
      Wire.write(decToBcd(dayOfMonth));
```

```
  Wire.write(decToBcd(month));
  Wire.write(decToBcd(year));
  Wire.write(byte(0));
  Wire.endTransmission();

}
```

Programul a fost dezvoltat și testat cu Arduino IDE 1.6.9, Arduino AVR Boards 1.6.14 și SimpleDHT 1.0.2.