**Proiect de Duminică** Autor: Robert Dima (YO8SHU)

Ceas pentru concurs cu Arduino Nano și RTC



De ceva vreme încerc să-mi fac timp pentru un proiect destul de simplu dar în același timp util, la care mă gâdesc de ceva vreme și nu am avut timpul necesar să-l încep darămite să-l și duc până la capăt.

Aveam, prin șmelțurile mele, un arduino nano care a fost achiziționat datorită prețului și nu pentru un proiect anume, iar eu voiam ceva simplu, care să-mi ofere și oarece satisfacție. În ideea că nu am făcut nici măcar inițerea în arduino, teama era destulă dar nefondată. În sfârșit, mi-am pus la punct planul, am achiziționat piesele de care mai era nevoie, și m-am apucat de treabă, puțin copy/paste de aici putină documentare de acolo și uite așa am încropit și firmwareul ce urma să-l uploadez pe arduino. După ce am construit ceasul pe breadboard și am făcut câteva debuguri la firmware totul era pregătit pentru a începe construcția montajului final.

Astfel: am folosit în acest proiect următoarele componente și module:

1 x Arduino Nano ,

1 x RTC cu comunicare I2c,

1 x LCD 16X2 compatibil HD44780,

1 x adaptor I2c la LCD,

1 x R 10K ohm,

1 x R 100K ohm,

1 x Cablaj de test cu dimensiuni 60x80 (din acela cu găurele),

2 x Rigletă cu pini (tată) 1x20,

1 x Rigletă cu pini (mama) 1x20

1 x Baterie 9v

1 x cutie de tablă refolosită (destul de mare sa-ti incapa totul in ea),

1 x comutator ON/OFF,

1 x Push Button,

Ca și scule, am avut următoarele:

1 x ciocan,

1 x șurubelniță „Phillip” („cruce” fară să știe mitropolitul că vrea drepturi de autor),

1 x pilă dreaptă

1 x letcon cu pământare(ca să nu ardem microcontrolerul cand facem reparații, GRIJA MARE CA SE ARDE),

1 x dremel cu pânză pentru tăiat fier,

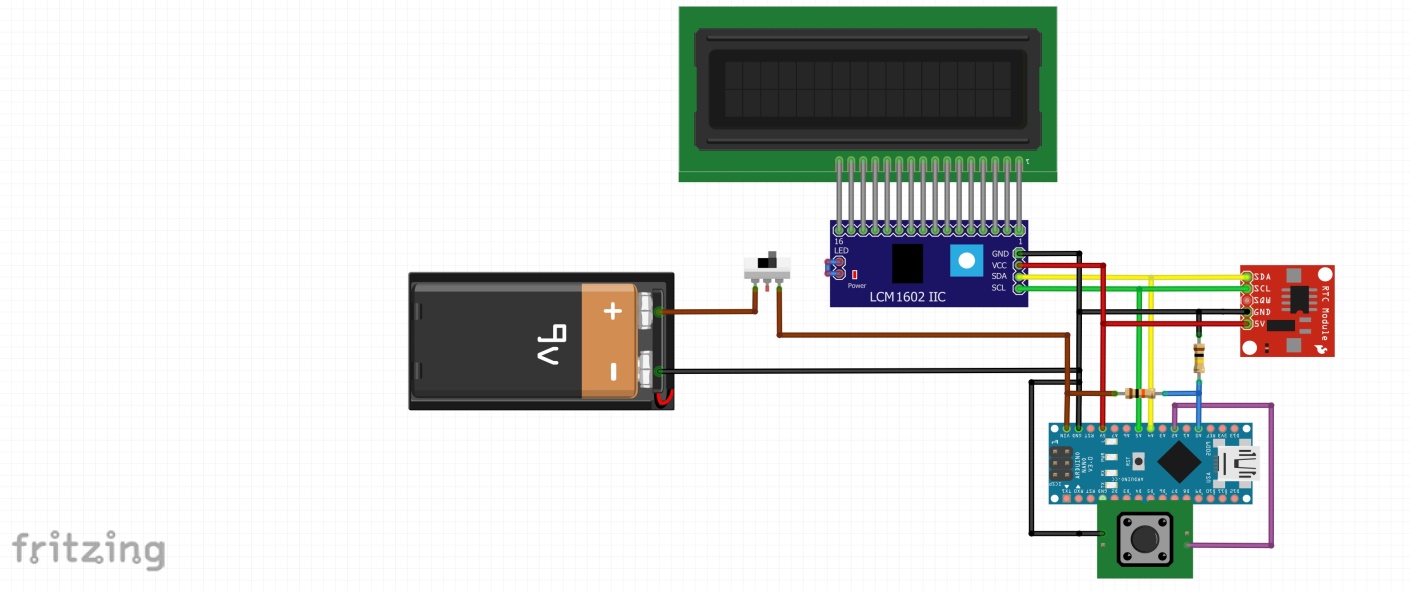
1 x cutter,

1 x ciupitor,

1 x patent,

Fludorul, sacâzul și sârmele cred că le are toata lumea.

Să trecem la treabă:



Am început prin a lipi adaptorul I2c la LCD pe un lcd16x2 de dimensiuni rezonabile...mai mare decât cele standard (recuperat, nu mai știu de pe unde). Pe placa de test am lipit baretele pentru arduino și pentru RTC conform schemei, pe aceeași placă am lipit și firele pentru conectarea lcd-ului, LCD-ul este unul recuperat, cu dimensiuni mari pentru o vizibilitate bună dar care este 100% compatibil cu HD44780, 16 caractere pe 2 rânduri. Apoi am lipit cele două rezistențe ce formează divizorul de tensiune pentru a coborî nivelul măsurat la sub 5v maxim măsurabil de arduino.

Apoi am trasat, cu un marker, forma LCD-ului pe capacul superior al cutiei, urmând imediat și tăierea cu dremelul, apoi am modificat și lărgit găurile de pe spatele cutiei astfel încât să iasă mufele de alimentare și mufa USB a modulului arduino. Am punctat unde vin găurile pentru șuruburi. După ce am montat arduino în interiorul cutiei și mi-am făcut o idee unde și cum montez restul m-am apucat de lipit pinii pe PCB de test urmând imediat și firele aferente conexiunilor.

După cum se vede și din schemă, divizorul de tensiune, se leagă pe pinul A0 al modulului arduino, acesta fiind un pin ANALOG IN ce conține un DAC cu rezoluție de 10 biti, el oferindu-i procesorului o valoare între 0 si 1023, mai mult decât de ajuns pentru a avea o acuratete de 0.01v în măsurarea tensiunii bateriei.

Ultima conexiune cu arduino se face pe pinul A2 unde s-a conectat un buton push cu celălalt capăt la GND.

Alimentare din baterie se face pe pinul VIN al modulului, iar toate perifericele se conectează pe pinul +5v de pe arduino.

Ceasul este gata de utilizare și nu mai este necesar decât să-i reglăm ora.

Pentru aceasta există o porțiune din cod unde se introduce data și ora exactă, UTC, și se „de-comentează” linia de cod semnalizată astfel dupa care facem upload, la cod, în arduino. După ce am văzut că arduino afișează data si ora corect, vom „re-comenta” linia de cod „de-comentată” anterior și facem iarăși upload la cod. Nefiind necesare alte modificări în cod până când vom sincroniza iarăși ceasurile (în funcție de modulul RTC folosit este posibil să nu mai fie necesară o astfel de sincronizare) Butonul aprinde retro-ilumiarea ecranului, pentru economisirea bateriei, stingându-se la 10 secunde dupa eliberarea butonului.

Baftă la construit.

Mai jos aveti codul sursa pe care-l puteti modifica si imbunatati dupa plac.

// Ceas pentru concursurile de radioamatori

// Versiune 3.0

// YO8SHU Robert

// Connectare LCD:

//pin SDA to Arduino Analog pin A4

//pin SCL to Arduino Analog pin A5

//pin GND to Arduino GND

//pin VCC to Arduino 5v

//Conectare RTC:

//pin SDA to Arduino Analog pin A4

//pin SCL to Arduino Analog pin A5

//pin GND to Arduino GND

//pin VCC to Arduino 5v

// Conectare Divizor pin A0

// Conectare buton pin A2

//Librarii

#include <Wire.h> // include libraria Wire.h

#include <LCD.h> // include libraria LCD.h

#include <LiquidCrystal\_I2C.h> // include libraria LiquidCrystal\_I2C.h

#define DS1307\_I2C\_ADDRESS 0x68 // defineste adresa I2C folosita de RTC 0x68

LiquidCrystal\_I2C lcd(0x27,2,1,0,4,5,6,7); // defineste adresa I2C folosita de LCD 0x27

// Configurare Voltmetru

int vBat = A0; //Pinul folosit pentru citirea voltajului de pe baterie, A0.

float vout = 0;

float vin = 0;

float R1 = 103000; // R1 aprox 100K, pentru acuratetea masurarii trebuie introdusa valoarea exacta!

float R2 = 10000; // R2 aprox 10K, pentru acuratetea masurarii trebuie introdusa valoarea exacta!

int value = 0;

// Configurare activare si temporizare retroiluminare

const byte BUTTON = A2; // Pinul folosit pentru butonul ce prinde retroiluminarea, A2

unsigned long buttonPushedMillis = 0; // defineste cand butonul a fost eliberat

unsigned long ledTurnedOnAt = 0; // defineste cand ledul sa aprins

unsigned long turnOnDelay = 250; // defineste cat se asteapta pana se aprinde ledul

unsigned long turnOffDelay = 10000; // defineste cat timp sta aprins ledul

bool ledReady = true; // flag for when button is let go

bool ledState = false; // for LED is on or not.

// Converteste numerele Zecimale in Binare

byte decToBcd(byte val)

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

// Converteste numerele Binare in Zecimale

byte bcdToDec(byte val)

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

// 1) Setarea datei si orei in modulul RTC

// 2) Porneste ceasul

// 3) regleaza afisarea orei in formatul 24 ore

// introduceti cifre corecte

void setDateDs1307( byte second, //ORA: // 0-59

byte minute, // 0-59

byte hour, // 1-23

byte dayOfWeek, //DATA: // 1-7

byte dayOfMonth, // 1-28/29/30/31

byte month, // 1-12

byte year) // 0-99

{Wire.beginTransmission(DS1307\_I2C\_ADDRESS);

Wire.write(0);

Wire.write(decToBcd(second)); // 0 to bit 7 starts the clock

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(0x10); // sends 0x10 (hex) 00010000 (binary) to control register - turns on square wave

Wire.endTransmission();}

// Citirea datei si orei din modulul RTC

void getDateDs1307(byte \*second,

byte \*minute,

byte \*hour,

byte \*dayOfWeek,

byte \*dayOfMonth,

byte \*month,

byte \*year)

// Reset the register pointer

{Wire.beginTransmission(DS1307\_I2C\_ADDRESS);

Wire.write(0);

Wire.endTransmission();

Wire.requestFrom(DS1307\_I2C\_ADDRESS, 7);

\*second = bcdToDec(Wire.read() & 0x7f); // A few of these need masks because certain bits are control bits

\*minute = bcdToDec(Wire.read());

\*hour = bcdToDec(Wire.read() & 0x3f); // Need to change this if 12 hour am/pm

\*dayOfWeek = bcdToDec(Wire.read());

\*dayOfMonth = bcdToDec(Wire.read());

\*month = bcdToDec(Wire.read());

\*year = bcdToDec(Wire.read());}

void setup()

{pinMode(BUTTON, INPUT\_PULLUP); // Setaza butonul ca fiind Pulup

pinMode(vBat, INPUT); // Setaza pinul de masura voltaj baterie ca fiind INPUT

lcd.setBacklightPin(3,POSITIVE); // Seteaza pinul 3 din i2c to lcd ca fiind cel ce controleaza iluminatul

lcd.setBacklight(HIGH); // Aprinde iluminatul

byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

Wire.begin();

// Introduceti mai jos data si ora exacta

// Aceasta informatii se introduc doar atunci cand dorim sincronizarea ceasului cu ora excacta

// In functie de calitatea RTC-ului acest lucru nu trebuie efectuat prea des.

second = 0; // aici se introduce secunda exacta

minute = 12; // aici se introduce minutul exact

hour = 13; // aici se introduce ora exacta

dayOfWeek = 7; // aici se introduce numarul zilei din saptamana

dayOfMonth = 12; // aici se introduce ziua

month = 04; // aici se introduce luna

year = 20; // aici se introduce anul

// Pentru a introduce datele de mai sus in modulul RTC, se decomenteaza urmatoarea linie si apoi se uploadeaza codul in arduino.

// setDateDs1307(second, minute, hour, dayOfWeek, dayOfMonth, month, year);

// Dupa ce sa uploadat codul cu ora exacta reglata, se comentaza la loc linia de deasupra si se face upload la cod din nou.

// daca linia raman decomentata, la fiecare pornire a ceasului informatiile de mai sus se reintroduc in

// modulul RTC ceea ce face ca data si ora sa se reseteze la momentul uploadului initial.

// initializare LCD

lcd.begin(16, 2); // Stabileste dimensiunile LCD-ului

lcd.setCursor(0,1); // Stabileste pozitia cursorului la inceputul primului rand , Stanga sus

lcd.print(" CONTEST TIMER "); // Afiseaza textul ditre paranteze

lcd.setCursor(0,0); // Stabileste pozitia cursorului la inceputul urmatorului rand.

lcd.print("YO8SHU/P Ver. 3"); // Afiseaza textul ditre paranteze, puteti modifica textul dupa plac

delay(5000); // pauza de 5 secunde sa se poata citi textul

lcd.clear();} // sterge ecranul

void loop()

{byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);

BackLight(); //Apeleaza functia backlight

//Afisare DATA si ORA

UTC(); // Afiseaza ora UTC pe primul rand

if (second < 20){CFR(); } // Daca secunde < 20 afiseaza pe randul 2 ora CFR

else if (second < 30){BatVolt(); } // Altfel daca secunde < 30 afiseaza voltajul bateriei pe randul 2

else if (second < 50){ DATA(); } // Altfel daca secunde < 50 afiseaza pe randul doi data

else if (second < 60){BatVolt(); }} // Altfel daca secunde < 60 afiseaza voltajul bateriei pe randul 2

//Functie citire si afisare voltaj baterie

void BatVolt()

{value = analogRead(vBat); // citeste pinul vBat si stocheaza valoarea in value

vout =(value \* 5) / 1024.0; // defineste vout ca fiind value inmulti cu tensiunea de referinta totul impartit la numarul de pasi

vin = vout / (R2/(R1+R2)); // defineste vin ca fiind vout impartit la divizorul de tensiune

if (vin<6) { // daca vin mai mic ca 6

lcd.setCursor(0,1); // stabileste cursorul pe randul 2

lcd.print("Low bat Recharge");} // afiseaza low bat reincarcati

else{ // altfel

lcd.setCursor(0,1); // cursorul pe randul 2

lcd.print("Bat. volt. "); // afiseaza: Bat. Volt.

lcd.print(vin); // afiseaza valoarea lui vin

lcd.print("V");}} // afiseaza: V

// Functie aprindere backlight

void BackLight()

{unsigned long currentMillis = millis(); // deineste surrentMillis ca find ora actuala

if (digitalRead(BUTTON) == LOW) {

buttonPushedMillis = currentMillis; // inoieste milis cand butonul a fost apasat

ledReady = true; }

if (ledReady) { // this is typical millis code here:

if ((unsigned long)(currentMillis - buttonPushedMillis) >= turnOnDelay) { // okay, enough time has passed since the button was let go.

lcd.setBacklight(HIGH); // Backlight on setup our next "state"

ledState = true; // save when the LED turned on

ledTurnedOnAt = currentMillis; // wait for next button press

ledReady = false;}}

if (ledState) { // okay, led on, check for now long

if ((unsigned long)(currentMillis - ledTurnedOnAt) >= turnOffDelay) {

ledState = false;

lcd.setBacklight(LOW);}}} // Backlight off

// functie afisare ora UTC

void UTC()

{byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);

//Afiseaza ORA UTC

lcd.setCursor(0,0);

lcd.print(" UTC ");

if (hour<10){lcd.print("0");}

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(" ");}

// Functie afisare ora CFR

void CFR()

{byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);

//Afiseaza ORA CFR

lcd.setCursor(0,1);

lcd.print(" CFR ");

if (hour+3<10){lcd.print("0");} // se inlocuieste +3 cu +2 daca se foloseste ora de iarna

if (hour+3>24){lcd.print (hour-21,DEC);} // se inlocuieste +3 cu +2 daca se foloseste ora de iarna

else {lcd.print(hour+3, DEC);} // se inlocuieste +3 cu +2 daca se foloseste ora de iarna

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(" ");}

// Functie afisare DATA

void DATA()

{byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

getDateDs1307(&second, &minute, &hour, &dayOfWeek, &dayOfMonth, &month, &year);

//Afiseaza DATA

lcd.setCursor(0,1);

lcd.print(" ");

switch(dayOfWeek){ //AFISEAZA in loc de cifra, numele zilei

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(" ");

if (dayOfMonth<10){lcd.print("0");} //Daca ZIUA mai mica ca 10 afiseaza "0" inainte

lcd.print(dayOfMonth, DEC); //Print ZIUA

lcd.print(".");

if (month<10) {lcd.print("0");} //Daca LUNA mai mica ca 10 afiseaza "0" inainte

lcd.print(month, DEC); //Print LUNA

lcd.print(".20");

lcd.print(year, DEC); //Print ANUL

lcd.print(" ");}