Marian ALDESCU (78298) - Heart Monitor

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Prototipul va masura pulsul si il va afisa sub forma unui grafic. Este util pentru a tine evidenta pulsului in anumite momente ale zilei sau in executia anumitor actiuni.

Descriere generală

Proiectul consta in construirea unui dispozitiv care sa citieasca semnalelele primite de la senzorul de puls, conversia acestora intr-un numar care va reprezenta numarul de pulsuri in unitatea de timp(1 min). Valorile analogice se vor transmite prin intermediul unei conexiuni USART si vor fi afisate sub forma unui grafic pe monitorul laptopului.

De asemenea, va fi primit un feedback luminos si unul sonor de la buzzer pentru fiecare puls masurat.

Mi se pare un subiect util persoanelor care doresc sa isi monitorizeze ritmul cardiac, in acest fel putandu-si adapta activitatile cotidiene si alimentatia in functie de rezultate pentru o stare de bine sporita si consolidarea sanatatii.

Schema bloc

Hardware Design

Lista piese

- Placa de baza
- ATMEGA324A-PU
- Senzor puls
- Display 16×2
- Buzzer
- Cablu USB
- FTDI

Schema electrica

Software Design

Pentru dezvoltarea software am folosit: -Sublime - editor text -avr-gcc - compilare -HIDBootFlash.exe incarcare hex

In principiu am utilizat laboratoarele pentru unele aspecte deja implementate.

Voi prezenta firmware-ul conform flow-lui datelor.

1. Achizitia datelor de la senzor. Senzor de puls este un simplu circuit constituit in principiu dintr-un led si o fotorezistenta. Led-ul emite o lumina puternica astfel incat sa treaca prin piele si muschi, o parte din ea intorcandu-se catre fotorezistor. In functie de cantitatea de sange aflata in vase, se poate estima numarul de batai pe minut. Deci in prima faza este nevoie de citirea unei valori de pe un pin analogic, folosind ADC-ul:

Initializare ADC:

```
void ADC_init(void)
{
    ADMUX = (1 << REFS0);
    ADCSRA = (1 << ADEN) | (5 << ADPS0);
}
```

Citirea valorilor analogice de pe canalul “channel”, in cazul de fata, pinul am folosit pinul PA0, care este conectat la output-ul senzorului.

```
uint16_t ADC_get(uint8_t channel)
{
    ADMUX = (ADMUX & ~(0x1f << MUX0)) | channel;

    ADCSRA |= (1 << ADSC);
    while(ADCSRA & (1 << ADSC));

    return ADC;
    (void)channel;
}
```

Pentru conversia datelor am setat TIMER2 pentru a se declansa la 128 de cicli(2ms) si sa citeasca o valoare de la ADC, care va fi trimisa prin USART, catre calculator. Initializare intrerupere timer: -modul de functionare al timer-ului este CTC, prescaler la 256, top count = 124. Activez intreruperile globale.

```
void interruptSetup() {
  TCCR2A = 0x02;
  TCCR2B = 0x06;
  OCR2A = 0X7C;
  TIMSK2 = 0x02;
  sei();
```

```
}
```

In rutina de tratare a intreruperii, in functie de valorile citite de pe pantele ascendente/descendente ale undelor de puls si durata fiecarei pulsatii, se calculeaza numarul de batai intr-un minut.

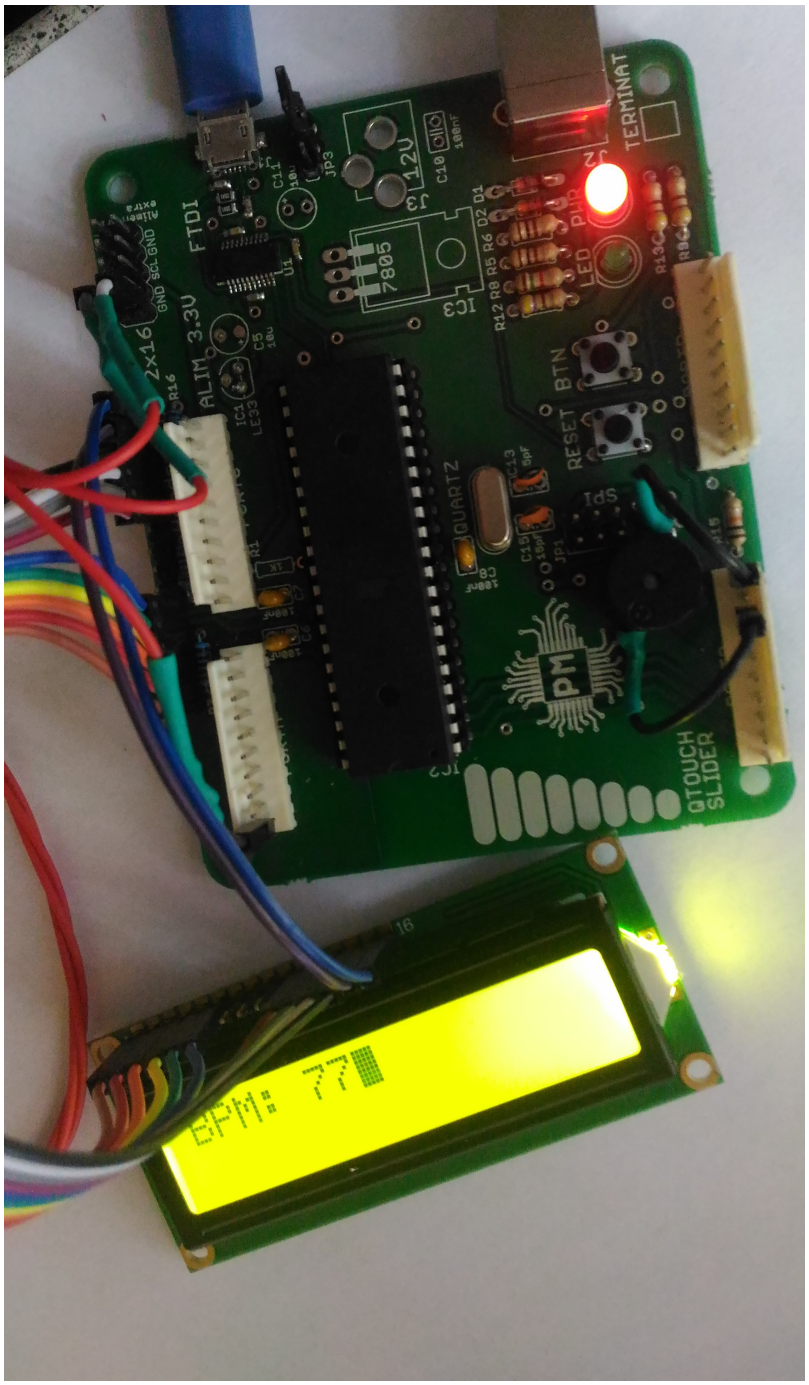
Initializare USART:

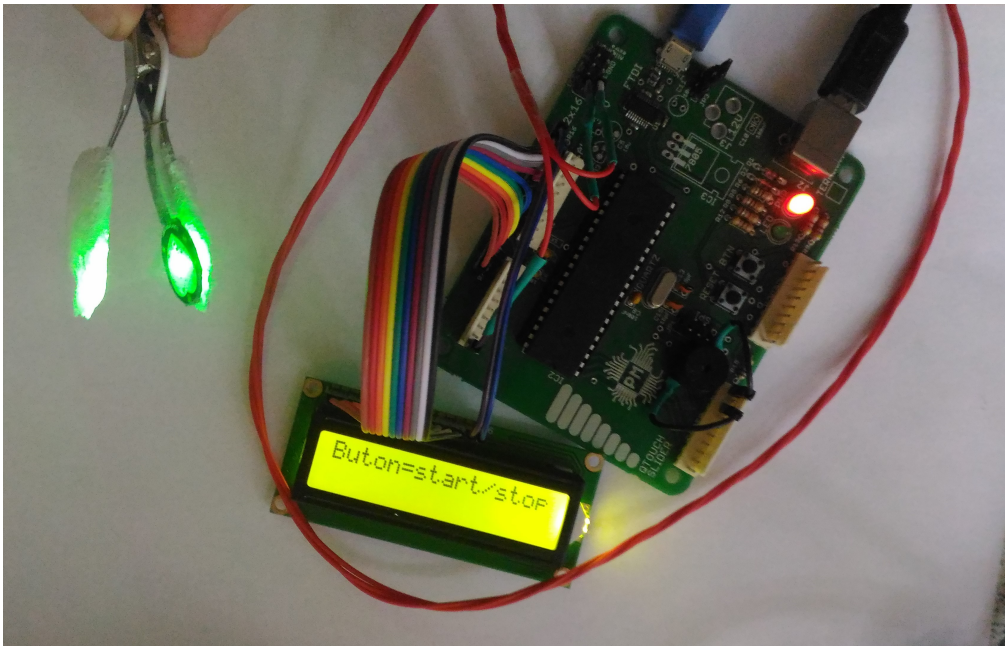
```
void USART0_init()
{
    // seteaza baud rate la 9600
    UBRR0 = 103;

    // porneste transmitatorul
    UCSR0B = (1<<TXEN0) | (1<<RXEN0);

    // seteaza formatul frame-ului: 8 biti de date, 1 biti de stop, paritate
para
    UCSR0C &= ~(1<<USBS0);
    UCSR0C |= (2<<UCSZ00);
    UCSR0C |= (1<<UPM01);
}
```

Pentru transmisia pe USART si afisarea pe LCD, am folosit bibliotecile existente in solutiile laboratorului.

Pa laptop va rula un script in python care va reprezenta grafic valorile primite de la microcontroler. Pentru acest lucru am folosit biblioteca matplotlib.

Rezultate Obţinute


FTDI
ALIM 3.3V
2x16
7805
IC3
QUARTZ
RESET
BTN
LED PIN
TERMINAL
QTOUCH SLIDER
PM
BPM: 77


Buton=start/stop


BPM: 116
Calibrare...

Pentru scurt demo accesati: D_MDOFJPh4w

Concluzii

Pot fi trase o multime de concluzii, insa cuvintele sunt de prisos pentru a descrie traseul (foarte)sinuos de realizare a proiectului. Cert este ca am aprofundat o multime de notiuni din sfera programarii embedded. De aceea, am incercat sa includ in proiectul meu lucrurile de baza pe care le-am invatat la laborator, incepand cu alegerea pieselor corespunzatoare, lipirea lor pe placa de baza, consultarea datasheet-urilor si continuand cu lucrul registrii si cu dezvoltarea soft-ului care va rula pe mC.

Trasand linie, pot spune ca a fost un proiect la care am lucrat cu mare placere.

Download

Cod sursă

Jurnal

2018-04-07 - Alegerea temei proiectului
2018-04-12 - Cautarea componentelor pentru placa de baza
2018-04-19 - Construirea schemei bloc
2018-05-03 - Lipirea componentelor pe placa de baza
2018-05-04 - Alcatuirea chemei electrice (EAGLE)
2018-05-17 - Incarcare bootloader si primul program-test
2018-05-19 - Scriere cod + teste
2018-05-20 - Finalizare documentatie (wiki, README, comentarii cod)

Bibliografie/Resurse

Resurse software:
1. Laboratoare PM: http://cs.curs.pub.ro/wiki/pm/
2. http://www.instructables.com/id/Pulse-Sensor-With-Arduino-Tutorial/

Resurse hardware:
1. https://www.optimusdigital.ro/ro/ (NU RECOMAND)
2. https://www.conexelectronic.ro/
3. https://contactelectric.ro/ (de aici am luat senzorul de puls)

Datasheet-uri:
1. LCD: lcd_16x2.pdf
2. Senzor puls: pulsesensorampedgettingstartedguide.pdf
3. Buzzer: lte12-series.pdf

- Documentația în format PDF