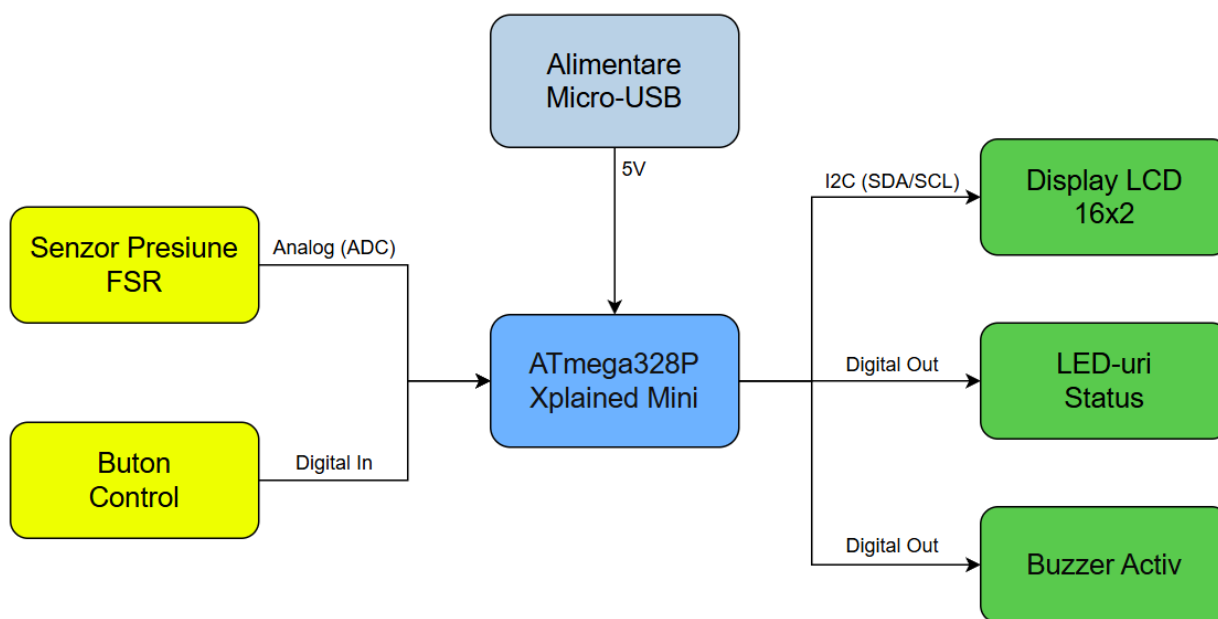


Monitorizare Rata Respiratorie

Introducere

Acest proiect implementeaza un sistem care estimeaza si monitorizeaza rata respiratorie a unui pacient, folosind un senzor de presiune FSR. Sistemul detecteaza mecanic ciclurile de inspiratie si expiratie prin intermediul variatiilor de presiune aplicate pe senzorul FSR si afiseaza rezultatele pe un ecran LCD. De asemenea, semnalizeaza vizual si sonor starea ritmului respirator, cu ajutorul unui buzzer activ si a unor LED-uri.

Descriere generală



Componentele interacționează între ele astfel:

- **Fluxul de date:** Senzorul de presiune FSR transmite un semnal analogic variabil către unitatea centrală. Acest semnal este procesat prin intermediul modulului ADC. În paralel, butonul funcționează ca o întrerupere pentru schimbarea stărilor de funcționare (PORNIT / OPRIT). Faza de calibrare este realizată automat la pornirea sistemului.
- **Procesare Software:** Microcontrolerul identifică momentele de inspirație și expirație, calculând intervalul de timp dintre acestea pentru a determina rata respiratorie (RR).
- **Feedback și Alerte:** Rezultatele numerice sunt transmise prin protocolul I2C către ecranul LCD. Starea de sănătate a ritmului este semnalizată vizual prin 2 LED-uri (Verde - Normal, Rosu - Critic).

Alerta sonora (Buzzer) este activata in cazul valorilor critice.

Hardware Design

Lista de piese

- Microcontroller ATmega328P Xplained Mini
- 1x Senzor FSR (senzor de presiune analogic)
- 1x Display LCD 16×2 cu modul adaptor I2C
- 2x LED-uri (1 rosu, 1 verde)
- 1x Buzzer Activ
- 1x Buton Tactil
- 1x Rezistenta 10KOhm (pentru divizorul de tensiune al senzorului FSR)
- 2x Rezistente 220Ohm (pentru limitarea curentului prin LED-uri)
- Breadboard, fire jumper

Schema electrica



Schema electrica este impartita in sub-circuite conectate prin label-uri:

- **Divizorul de tensiune (Senzor FSR):** Senzorul este conectat in serie cu o rezistenta de 10KOhm. Tensiunea rezultata este transmisa prin nodul central.
- **Iterfata Utilizator:** LED-urile sunt protejate de rezistente de 220Ohm care limiteaza curentul, buzzer-ul activ este conectat direct la pinul digital, iar butonul de comanda controleaza starea sistemului.
- **Display LCD:** Afisarea datelor se realizeaza prin protocolul I2C, utilizand pinii hardware dedicati PC4 si PC5.

Descriere Pini

Componenta	Pin (Rol)	Justificare
Senzor FSR	PC0 (ADC0)	Necesita un convertor analog-digital pentru a translata presiunea in valori numerice
LCD	PC4 (SDA) si PC5 (SCL)	Pini hardware dedicati pentru protocolul I2C
Buton START/STOP	PB0 (PCINT0)	Suporta intreruperi de tip Pin Change
LED Verde	PD3 (GPIO output)	Semnalizare vizuala pentru starea normala
LED Rosu	PD2 (GPIO output)	Semnalizare vizuala pentru starea critica
Buzzer	PD4 (GPIO output)	Alarma sonora pentru starea critica

Software Design

Proiectul a fost dezvoltat utilizand mediul **VS Code** cu extensia **PlatformIO**, codul fiind scris in limbajul C.

Biblioteci standard folosite

- **<avr/io.h>**: Asigura maparea registrilor.
- **<avr/interrupt.h>**: Oferă suport pentru gestionarea intreruperilor hardware.
- **<util/delay.h>**: Utilizata la secventa de boot ecranului LCD.
- **<stdio.h>**: Folosita pentru formatarea sirurilor de caractere prin functia `sprintf()`.

Arhitectura sistemului

- **gpio.h/c**: Initializarea elementelor de I/O (LED-uri, buton, buzzer) si pornirea/oprirea acestora.
- **adc.h/c**: Configurarea ADC-ului pentru citirea senzorului FSR.
- **timer.h/c**: Configurarea Timer0 in mod CTC pentru a genera o intrerupere la 1ms.
- **i2c.h/c** si **lcd.h/c**: Implementarea protocolului TWI pentru comunicarea cu ecranul LCD.
- **usart.h/c**: Configurarea comunicatiei seriale dintre PC si microcontroler.

Functii implementate

Logica este structurata in jurul a doua functii fundamentale apelate in bucla principala.

1. Functia `process_respiration()`

Aceasta functie preia datele de la senzorul FSR si implementeaza state machine-ul algoritmului.

- **Esantionare**: Citirea valorii de la convertorul ADC se realizeaza la un interval fix pentru a nu suprasolicita procesorul.
- **Detectia peak-urilor**:
 - O **inspiratie** este detectata atunci cand valoarea ADC depaseste `THRESHOLD_INSPIRATION` (300) si pacientul se afla in starea de expiratie.
 - O **expiratie** este inregistrata atunci cand valoarea ADC scade sub `THRESHOLD_EXPIRATION` (200).
- **Calculul Ratei Respiratorii**: In momentul detectiei unei respiratii, se calculeaza diferenta de timp fata de respiratia anterioara. Aceasta valoare este introdusa intr-un buffer circular de dimensiune 4. Cat timp bufferul nu este plin, sistemul ramane in starea de **calibrare**. Odata calibrat, se calculeaza media acestor intervale, iar rata respiratorie este calculata ca **$rpm = 60000 / avg_delta$** .
- **Monitorizarea apneei**: La fiecare rulare, functia verifica daca diferenta dintre timpul curent si timpul ultimei respiratii depaseste 8 secunde, caz in care sistemul trece in starea critica.

2. Functia `update_ui()`

Aceasta functie separa logica de calcul de interfata cu utilizatorul.

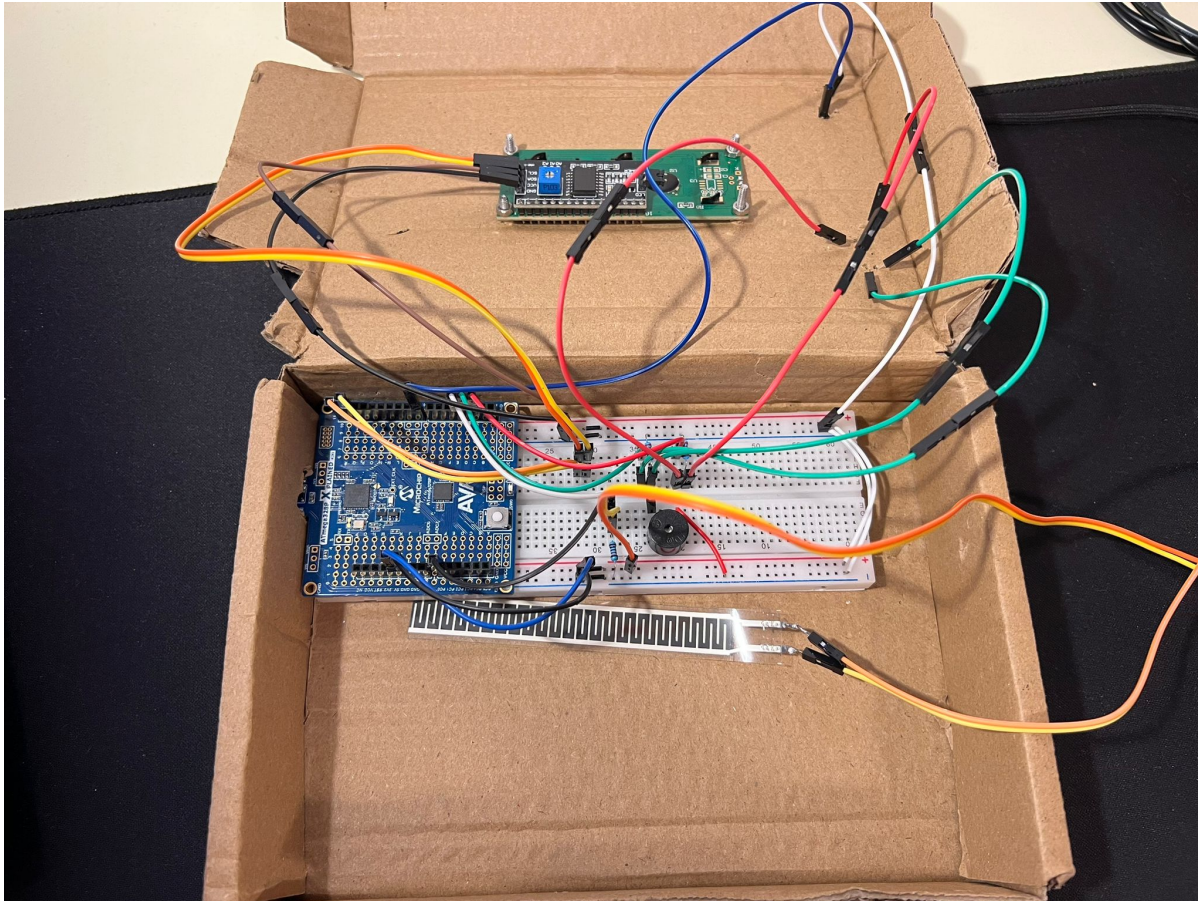
- **Actualizare LCD:** Se formateaza sirurile de caractere transmise prin I2C. In functie de starea sistemului, ecranul afiseaza dinamic:
 - mesajul de repaus, cand sistemul este oprit.
 - textul "Calibrare...", pana cand se umple bufferul circular.
 - valoarea numerica a RPM, alaturi de status (OK sau CRITIC).
- **Controlul alertelor:** Functia analizeaza valoarea calculata a RPM:
 - Daca RPM este in intervalul normal (12-20), pinul PD3 (LED verde) se va aprinde si stinge la interval de 500ms, iar alarma este oprita.
 - Altfel, pinii PD2 (LED rosu) si PD4 (buzzer) sunt activati pentru a declansa alarma.

Rezultate Obținute

Sistemul a fost testat si rezultatele finale demonstreaza indeplinirea tuturor specificatiilor:

- Sistemul porneste si se opreste instant la apasarea butonului de control.
- La pornire, display-ul afiseaza starea de "Calibrare...", pentru a evita rate respiratorii eronate calculate dupa doar 1-2 inspiratii.
- Rata respiratorie este monitorizata in timp real, verificandu-se constant daca aceasta se afla in intervalul normal de 12-20 RPM.
- Starea sanatoasa este semnalata prin LED-ul verde, in timp ce starea critica porneste un sistem de alarma format din LED-ul rosu si buzzer.
- Starea de alarma este pornita daca nu se mai detecteaza nicio inspiratie intr-un interval de 8 secunde.

Imagini



Demo

<https://youtu.be/KPzY-8TT7-Y>

Concluzii

Realizarea acestui proiect a demonstrat posibilitatea construirii unui prototip de monitorizare medicala folosind resurse hardware foarte limitate.

Download

[Repository GitHub](#)

Jurnal

- **Saptamana 9:** Stabilirea listei finale de componente necesare.
- **Saptamana 11:** Crearea paginii OCW. Realizarea montajului hardware pe breadboard si testarea componentelor.
- **Saptamana 12:** Crearea schemei electrice. Descrierea detaliata a pinilor alesi pentru fiecare componenta. Implementare software partiala.
- **Saptamana 13:** Finalizarea implementarii software.

Bibliografie/Resurse

Resurse Hardware

- [ATmega328P Xplained Mini User Guide](#)

Resurse Software

- Laboratoarele de PM (GPIO, USART, Intreruperi, ADC, I2C)

From:
<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2026/tarik_ilhan.omer/antonio.ciocodeica



Last update: **2026/05/26 17:04**