

Uscător de mâini inteligent

Autor: Harea Teodor-Adrian, 333CA

Contact: teodor_adrian.harea@stud.acs.upb.ro

Introducere

Ce face proiectul?

Proiectul este un sistem inteligent care utilizează trei senzori (de mișcare, distanță și infraroșu) pentru a detecta prezența și acțiunile utilizatorilor. Funcționalitățile principale includ:

- Detectarea mișcării cu ajutorul senzorului **PIR HC-SR501**.
- Măsurarea distanței până la obiecte folosind senzorul ultrasonic **HC-SR04P**.
- Controlul sistemului prin telecomandă infraroșu.
- Afișarea informațiilor pe un **ecran LCD** și în consola serială.
- Activarea unui **LED**, a unui **buzzer** și a unui **motor DC** în funcție de intrările senzorilor.

Scopul proiectului

Scopul principal al proiectului este de a demonstra integrarea și utilizarea practică a mai multor senzori într-un sistem coordonat.

Ideea de pornire

Ideea proiectului a pornit de la dorința de a crea un sistem simplu, dar eficient, care să poată detecta și răspunde la acțiunile utilizatorilor. Inspirația a venit din:

- Observarea sistemelor automate din viața de zi cu zi (ex: uscătoare de mâini).
- Nevoia de a combina funcționalități diferite (detectie, afișare, control) într-un singur proiect.

Utilitatea proiectului

- Oferă un exemplu practic de integrare a senzorilor (PIR, ultrasonic, IR) într-un sistem coerent.
- Demonstrează utilizarea unui **microcontroller (ATmega328P)** și a unui **LCD I2C** într-un context real.
- A consolidat cunoștințele despre programarea microcontrolerelor și senzorilor.

- A oferit experiență practică în gestionarea dispozitivelor periferice.
- A dezvoltat abilități de depanare și optimizare a codului embedded.

Descriere generală

Schema sistemului:



Schemă bloc a sistemului:



Listă Componente

Microcontroller

- ATmega328P (16MHz, 32KB Flash)

Senzori

- HC-SR501 - Senzor mișcare PIR ● HC-SR04P - Senzor ultrasonic ● VS1838 - Receptor infraroșu

Afișaje

- LCD 16×2 cu interfață I2C

Dispozitive de acțiune

- Motor DC 3-5V cu elice ● Buzzer activ 5V ● LED cu rezistență 220Ω

Alimentare

● Placa de dezvoltare cu regulator 5V ● Conexiune USB pentru alimentare

Biblioteci software

● LiquidCrystal_I2C (v1.1.2) ● IRremote (v3.6.1)

Hardware Design

Schemă electrică a sistemului:



Bill Of Materials (BOM)

Resurse oficiale și linkuri de cumpărat pentru componente

Componente

1. Microcontroller ATmega328P

Datasheet	Link oficial
Cumpărare	Arduino UNO R3 (ATmega328P)

2. Senzori

Componentă	Datasheet	Cumpărare
-----	-----	-----
LCD 16x2 cu I2C	Link	OptimusDigital
Senzor PIR HC-SR501	Link	OptimusDigital
Senzor ultrasonic HC-SR04P	Link	OptimusDigital
Receptor infraroșu	Link	OptimusDigital

3. Dispozitive de acțiune

Componentă	Datasheet	Cumpărare
-----	-----	-----
Buzzer activ 5V	Link	OptimusDigital
Motor DC	Link	OptimusDigital
LED albastru 3mm	Link	OptimusDigital

Biblioteci Software

Bibliotecă	Documentație
-----	-----
LiquidCrystal_I2C	GitHub
IRremote	GitHub

Software Design

Motivația alegerii bibliotecilor folosite

- **Arduino.h** – Biblioteca de bază pentru orice proiect Arduino. Oferă funcțiile esențiale pentru lucrul cu pini plăcii, temporizări, comunicare serială și alte operații fundamentale. Fără această bibliotecă, programarea directă a plăcii Arduino ar fi mult mai dificilă.
- **Wire.h** – Permite comunicarea pe magistrala I2C între Arduino și alte dispozitive, cum ar fi afișajul LCD. Utilizarea acestei biblioteci reduce numărul de pini necesari pentru conectare și simplifică foarte mult transmiterea datelor către dispozitivele compatibile I2C.
- **LiquidCrystal_I2C.h** – Folosită pentru controlul afișajului LCD cu interfață I2C. Oferă funcții simple pentru afișarea mesajelor pe ecran, gestionarea cursorului și alte operații utile, făcând codul mai clar și mai ușor de întreținut.
- **IRremote.h** – Permite recepționarea și decodificarea semnalelor de la o telecomandă IR. Astfel, sistemul poate fi controlat de la distanță, fără a fi nevoie de implementarea manuală a protocoalelor IR, ceea ce ar fi mult mai complex.

Utilizarea funcționalităților din laborator în cadrul proiectului

- **GPIO** → S-au folosit intensiv funcțiile de configurare și control al pinilor digitali (pinMode, digitalWrite, digitalRead) pentru a interacționa cu senzorul PIR, LED-ul, buzzer-ul și motorul. Acest lucru a permis conectarea și controlul direct al componentelor hardware.
- **UART** → Prin utilizarea funcțiilor Serial.begin și Serial.print, proiectul permite monitorizarea și debugging-ul în timp real, facilitând identificarea rapidă a eventualelor probleme și afișarea valorilor primite de la telecomandă.

- **Timere** → Funcția `millis()` este folosită pentru a implementa temporizări fără blocare (non-blocking delays), esențiale pentru gestionarea stărilor și tranzițiilor între acestea, fără a opri execuția programului.
- **I2C** → Comunicarea cu afișajul LCD se realizează prin protocolul I2C, folosind bibliotecile `Wire.h` și `LiquidCrystal_I2C.h`. Acest lucru reduce numărul de pini utilizați și simplifică transmiterea datelor către LCD.

Despre implementare

Structura software a proiectului este organizată în jurul unei mașini de stări (state machine), fiecare stare reprezentând o etapă distinctă din funcționarea uscătorului de mâini inteligent. Fiecare funcționalitate hardware este integrată și controlată prin cod, iar tranzițiile între stări sunt gestionate pe baza semnalelor primite de la senzori sau de la telecomandă.

Structura proiectului

Inițializarea componentelor hardware (senzori, LCD, motor, LED, buzzer, telecomandă IR) se face în funcția `setup()`. În funcția `loop()`, sistemul verifică în permanență starea curentă și reacționează la evenimente (mișcare detectată, distanță măsurată, comandă IR). Fiecare stare (ex: `CALIBRATING`, `LISTENING`, `MOTION_DETECTED`, `MOTOR_RUNNING`, `PAUSED` etc.) are un bloc dedicat de cod, cu acțiuni specifice și condiții de tranziție către alte stări. Temporizările sunt gestionate cu ajutorul funcției `millis()`, evitând blocarea execuției și permițând răspuns rapid la evenimente.

Interacțiunea dintre funcționalități

Senzorul PIR detectează mișcarea și declanșează trecerea în starea de detecție. Senzorul ultrasonic măsoară distanța față de utilizator și decide dacă motorul trebuie pornit sau dacă utilizatorul este prea departe. Telecomanda IR permite controlul manual al sistemului (pauză, pornire continuă, revenire la ascultare). LCD-ul afișează mesaje relevante pentru fiecare stare, oferind feedback utilizatorului. LED-ul, buzzer-ul și motorul sunt activate/dezactivate în funcție de starea sistemului și de valorile senzorilor.

Validarea funcționalităților

Fiecare funcționalitate a fost testată individual: s-a verificat răspunsul la mișcare, măsurarea corectă a distanței, afișarea mesajelor pe LCD, controlul cu telecomanda IR și funcționarea corectă a motorului, LED-ului și buzzer-ului. S-au folosit mesaje pe Serial Monitor pentru debugging și monitorizare în timp real. Sistemul a fost testat în ansamblu, simulând scenarii reale de utilizare (prezența unui obiect în raza de acțiune a senzorilor, distanță variabilă, comenzi telecomandă), pentru a confirma că toate componentele interacționează corect și că tranzițiile între stări se realizează

conform cerințelor.

Optimizări realizate

- **Eliminarea codului redundant** prin folosirea funcțiilor dedicate pentru fiecare **stare** a automatului, ceea ce crește **lizibilitatea** și **modularitatea** codului.
- Utilizarea **variabilelor statice locale** pentru **debounce** și **temporizări**, evitând poluarea spațiului global și asigurând corectitudinea temporizărilor.
- **Tranzițiile între stări** sunt clar marcate cu mesaje pe **LCD** și în consola **serială**, oferind feedback util pentru utilizator și pentru depanare.
- **Acțiunile pentru senzori și actuatori** sunt izolate în funcții specifice fiecărei stări, facilitând modificarea și extinderea comportamentului sistemului.
- Conversia timpului de la **senzorul ultrasonic** în **centimetri** este realizată eficient cu o **formulă standard**, asigurând acuratețea măsurărilor.
- Structura codului permite **extinderea facilă** cu noi funcționalități sau stări, menținând în același timp **claritatea** și **robustețea** implementării.

Rezultate Obținute

[Videoclip cu proiectul](#)



Concluzii

Proiectul demonstrează o implementare **eficientă** a unui sistem automatizat cu **senzori** și **motor**, controlat printr-o **mașină de stări** clar structurată. Codul este **modular**, ușor de **extins** și **întreținut**. Optimizările asigură **răspuns rapid** și **feedback clar** pentru utilizator. Sistemul poate fi **adaptat** cu ușurință pentru funcționalități suplimentare.

GitHub

[Proiect PM pe GitHub](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2025/iivasciuc/teodor_adrian.harea



Last update: **2025/05/19 17:40**