

Truck alcohol safety check

Introducere

Proiectul constă în implementarea unui sistem de siguranță pentru camioane, șoferul fiind nevoit să treacă un test de alcool pentru a putea porni motorul, pentru demonstrație conține un servo motor. Acesta utilizează o cartelă RFID pentru a verifica identitatea șoferului, dispune de un ecran LCD pentru afișarea informațiilor, un buzzer pentru a alerta utilizatorul în timpul procesului și un card SD pentru stocarea rezultatelor și prevenirea fraudei. (Ideal) Cardul SD este plasat strategic în interiorul camionului, astfel încât să fie inaccesibil.

Implementarea acestui sistem de siguranță pentru camioane este utilă pentru prevenirea accidentelor rutiere și protejarea vieților, cerința de a trece un test de alcool înainte de a porni motorul contribuind la asigurarea că șoferii sunt în stare bună și în deplină capacitate de a conduce.

Consider că proiectul reprezintă un prototip viabil, care, cu ajustări minime, ar putea fi adaptat și implementat în sistemul unui camion.

Descriere generală

O schemă bloc cu toate modulele proiectului vostru, atât software cât și hardware însoțită de o descriere a acestora precum și a modului în care interacționează.

Exemplu de schemă bloc: <http://www.robs-projects.com/mp3proj/newplayer.html>



Pasii pe care șoferul trebuie să îi urmeze sunt:

Șoferul trebuie să aștepte până când senzorul de alcool este pregătit (senzorul MQ3 are un timp necesar pentru încălzire). Șoferul validează cardul sau la cititorul RFID pentru a confirma identitatea. Dacă identitatea este confirmată atunci se va afișa numele sau se va începe testul. Senzorul de alcool MQ3 preia aerul suflat spre el și transmite rezultatul la arduino. Arduino-ul procesează rezultatul primit de la senzor și după procesarea calculului există 2 situații: nivelul de alcool este egal cu 0 sau mai mare decât 0. Pentru testul negativ se transmite prin USB "ENGINE_STARTING", se afișează pe LCD că se poate porni motorul și se pornesc servo motorul. În cazul în care testul este pozitiv la alcool, atunci șoferul nu poate porni motorul, este afișat un mesaj conform pe LCD și primește un delay de 15 minute până când va putea reîncepe procesul.

Hardware Design

Aici puneți tot ce ține de hardware design:

- listă de piese
- scheme electrice (se pot lua și de pe Internet și din datasheet-uri, e.g. <http://www.captain.at/electronic-atmega16-mmc-schematic.png>)
- diagrame de semnal
- rezultatele simulării

Lista componente:

- 1 x arduino UNO
- 1 x breadboard
- 1 x IIC 1602 LCD
- 1 x RFID Module RC522
- 1 x RFID keychain
- 1 x RFID White card
- 1 x senzor alcool MQ3
- 1 x Placa de stocare Micro SD TF Card reader
- 1 x rezistenta 220
- 1 x buzzer pasiv
- 1 x servo motor SG90

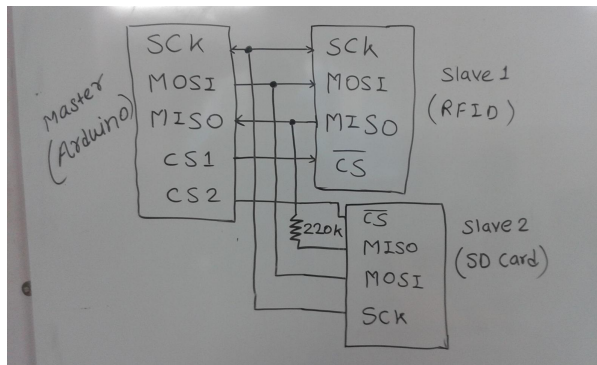
Schema electrica:



Explicatii pini:

- 1) Pentru motorul servo SG90 am ales drept pin de semnal, pinul 6 digital PWM (orice pin digital PWM era bun).
- 2) Pentru senzorul de alcool MQ3 am ales sa il folosesc in mod analogic, astfel am conectat la A0, respectiv A0 la arduino (orice pin analog era ok).
- 3) Pentru LCD I2C am urmat varianta standard de conectare. (https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615). Am folosit pinii A4 si A5 pentru a obtine SDA respectiv SCL.
- 4) Buzzerul pasiv l-am conectat la pinul digital 2 deoarece imi era mai accesibil. Orice pin digital liber este perfect okay.

Problema a fost la RFID si SD card reader, ambele folosind SPI deci aveau MISO, MOSI si SCK comun. O solutie de pe internet care m-a ajutat in conectarea si cardului SD, dar si a RFID-ului simultan: <https://forum.arduino.cc/t/sd-card-and-rfid-not-working-together-spi-conflict/995287/6>



Astfel:

5) Pentru RFID am conectat la pinii digitali: 11 pentru MOSI, 12 pentru MISO, 13 pentru SCK, 10 pentru SDA si 9 pentru RST.

6) Pentru SD card reader am conectat la pinii digitali: 11 pentru MOSI, 12 pentru MISO, 13 pentru SCK, am adaugat rezistenta de 220 in plus pentru MISO, si 4 pentru CS.

PENTRU DEMO: Am pus link cu video pe YT la "Rezultate obtinute"

Software Design

Descrierea codului aplicației (firmware):

- mediu de dezvoltare (if any) (e.g. AVR Studio, CodeVisionAVR)
- librării și surse 3rd-party (e.g. Procyon AVRlib)
- algoritmi și structuri pe care plănuți să le implementați
- (etapa 3) surse și funcții implementate

Mediul de dezvoltare folosit este Arduino IDE. Este o alegere excelenta pentru un incepator, avand un setup foarte simplu.

Stadiul actual al implementarii: Proiectul implementeaza tot ce mi-am propus, acesta are drept functionalitati software:

- Detecteaza si citeste carduri RFID. In functie de cardul prezentat, valideaza identitatea si permite accesul catre pasii urmatoari de testare. Totodata, daca accesul este valid, afiseaza si numele utilizatorului (soferului).
- Utilizarea senzorului de alcool MQ3. Senzorul analizeaza cantitatea de alcool din aerul suflat spre el si in functie de valoarea calculata si **aproximata** (proces descris mai jos) permite pornirea motorului.
- Servo motorul SG90 realizează o mișcare de 180 de grade, deplasându-se complet către stânga și apoi complet către dreapta.
- Sd card reader. Este folosit pentru a salva intr-un fisier text, concentratia de alcool alaturi de identitatea soferului (un camion are mai multi soferi daca e pe echipaj).
- LCD I2C. Este folosit la fiecare etapa. Afiseaza stadiul procesului de testare, precum si rezultatele

(apt de condus sau nu).

- Buzzer pasiv. Este folosit pentru a acompania fiecare etapa, prin furnizarea de feedback.

Motivatia alegerii bibliotecilor:

- Pentru RFID am ales sa folosesc [MFRC522.h](#). Biblioteca este special concepută pentru cititorul RFID RC522, care este unul dintre cele mai comune și accesibile module RFID disponibile. Am vazut-o recomandata peste tot si majoritatea tutorialelor foloseau aceasta librerie.
- Pentru servo motor am folosit [Servo.h](#). Biblioteca Servo este utilizată pentru a controla servo motorul SG90. Permite setarea unghiului de rotație al servo motorului și controlul acestuia prin semnale PWM.
- Pentru sd card reader am folosit [SD.h](#). Biblioteca SD este folosită pentru a interacționa cu cardurile SD. Permite inițializarea cardului, deschiderea, citirea și scrierea fișierelor pe cardul SD. Am testat toate functionalitatile sale, si am ramas doar cu scrierea intr-un fisier.
- Pentru lcd i2c am folosit [LiquidCrystal_I2C.h](#). Această bibliotecă este folosită pentru a controla afișajul LCD cu interfață I2C. Concret am folosit mult metodele de print, clear si setCursor.
- Pentru RFID si SD card am avut nevoie si de libraria [SPI.H](#). Biblioteca SPI se ocupă de complexitățile configurării vitezei magistralei, ordinii datelor și modului, care pot diferi între dispozitive. Bibliotecile [SD.h](#) si [MFRC522.h](#) depind de aceasta biblioteca.

Elementul de noutate al proiectului: unicitatea este ideea de integrare a unui sistem in plus de siguranta, pentru a preveni condusul sub influenta alcoolului.

Utilizarea functionalitatilor din laborator in cadrul proiectului:

- Din laboratorul de GPIO am folosit cunostintele generale, care au oferit baza pentru folosirea unor componente simple, precum buzzerul pasiv. (setarea unui pin digital pe high, pe low)
- Din laboratorul de PWM am folosit cunostintele pentru intelegerea controlului unui motor servo. Desi am folosit librerie ajutatoare, cunostintele acumulate au ajutat in intelegerea utilizarii unui motor folosind PWM.
- Din laboratorul de SPI am folosit mai mult pentru partea hardware. Explicatiile oferite in conectarea mai multor dispozitive folosind SPI, mai precis **2.1 Conectarea mai multor dispozitive** a fost de ajutor, am gasit si pe internet aceiasi solutie. A fost util ca am mai lucrat cu cardul SD la laborator, acasa m-am descurcat relativ usor. (**Debugging enervant**, scoate cardul sd, бага cardul sd, laptopul n-are cititor sd, foloseste laptopul vechi).
- Din laboratorul de I2C am invatat modul în care dispozitivele master și slave comunică folosind doar două linii: SDA (linia de date) și SCL (linia de ceas). Aceasta informatie a fost utila, pentru a avea o minima intelegere despre cum libraria LiquidCrystal lucreaza in spate.

Scheletul proiectului:

setup(): am initializat lcd-ul, sd card reader-ul si rfid-ul. Am setat pinii pentru servo motor si pentru buzzer. Am afisat un mesaj de informare, ca trebuie asteptate 3 beep-uri. Am setat un delay pentru a astepta sa se incalzeasca senzorul de alcool. Dupa delay am folosit functia tone() pentru a reda 3 sunete, pentru a alerta utilizatorul ca senzorul este gata.

loop(): am folosit functia PICC_IsNewCardPresent() pentru a detecta aparitia unei cartele RFID. Cu ajutorul PICC_ReadCardSerial() este citita valoarea cartelei. Daca este egala cu o anumita valoare (cunoscuta), afisez "Hello Marian" si varsta random de 22 de ani. (Marian e ales arbitrar) Ulterior actualizez lcd-ul si apelez functia performBreathalyzerTest(). Am incercat sa folosesc functii pentru fiecare componenta pentru un coding style mai bun. Functia foloseste analogRead() si citeste valoarea de la senzorul de alcool. Aici intervine logica explicata in **calibrarea elementelor de**

senzoristica. Dupa intoarcerea rezultatului functiei, se apeleaza `saveDataToSD()` care verifica initializarea cardului SD si folosind `SD.open()` obtine un `dataFile`, pe care apeleaza metoda `print()` si `println()` pentru a scrie concentratia de alcool si UID-ul soferului in fisier. Dupa salvarea rezultatului, daca valoarea a fost zero atunci apelez functia `servoSpin()`, care folosind 2 for uri realizeaza 10 miscari ale servo motorului. In schimb, daca valoarea este pozitiva, atunci afisez un mesaj si pun un delay de 2 minute, care folosind un while afiseaza live cat a mai ramas.

Calibrarea elementelor de senzoristica:

Ideea mea a fost sa iau o valoare de baza, senzorul depinde si de camera in care te afli, poluarea din ea, pentru ca el nu analizeaza strict alcoolul. Astfel, salvez valoarea intiala si apoi intr-un while extrag valori. Daca diferenta este mai mare de un prag intre cele doua (pragul setat este de 100) atunci inseamna ca e alcool detectat.

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Video proiect: https://www.youtube.com/watch?v=3NGuXG7dtdw&ab_channel=OnelAlexandruGabriel

Video proiect final: <https://youtu.be/h2ZpldwdIC8>

Concluzii

Probleme intalnite:

- ⇒ M-am chinuit cu prima componenta pusa, lcd-ul are pe spate un potentiometru si a trebuit sa il setez cu surubelnita. Problema a fost ca nu stiam ca e de la asta, a durat ceva sa caut pe net.
- ⇒ Am mai avut problema cand am conectat si RFID-ul si SD card reader-ul. Cand alegi componentele proiectului, este bine sa verifici cum stai cu pinii inainte, pentru a nu fi nevoie sa multiplexezi.
- ⇒ Trebuie sa gasesti adresa lcd-ului, important de stiut daca nu iti merge din prima.

Download

Codul se poate descarca de mai jos.

[proiect_pm_onel_alexandru_332cb.zip](#)

Jurnal

Puteți avea și o secțiune de jurnal în care să poată urmări asistentul de proiect progresul proiectului.

M-am încadrat în toate deadline-urile la timp. Inițial, când aveam ideea deja lucrăm la proiect și a fost gata înainte de deadline hard și soft. Ulterior, m-am ocupat de aranjarea site-ului de pe ocw.

Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

Resurse hardware:

- <https://forum.arduino.cc/t/sd-card-and-rfid-not-working-together-spi-conflict/995287/6>
- <https://www.instructables.com/Arduino-Wiring-and-Programming-of-RFID-Sensor/>
- <https://www.instructables.com/Micro-SD-Card-Tutorial/>
- https://projecthub.arduino.cc/arduino_uno_guy/i2c-liquid-crystal-displays-5eb615

Resurse software:

- <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>
- <https://www.arduino.cc/reference/en/libraries/mfrc522/>
- <https://www.arduino.cc/reference/en/libraries/sd/>

Laboratoare:

- <https://ocw.cs.pub.ro/courses/pm/lab/lab0-2023>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab5-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab6-2023-2024>
- <https://ocw.cs.pub.ro/courses/pm/lab/lab3-2023-2024>

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/avaduva/alexandru.onel0512>



Last update: **2024/05/24 13:52**