

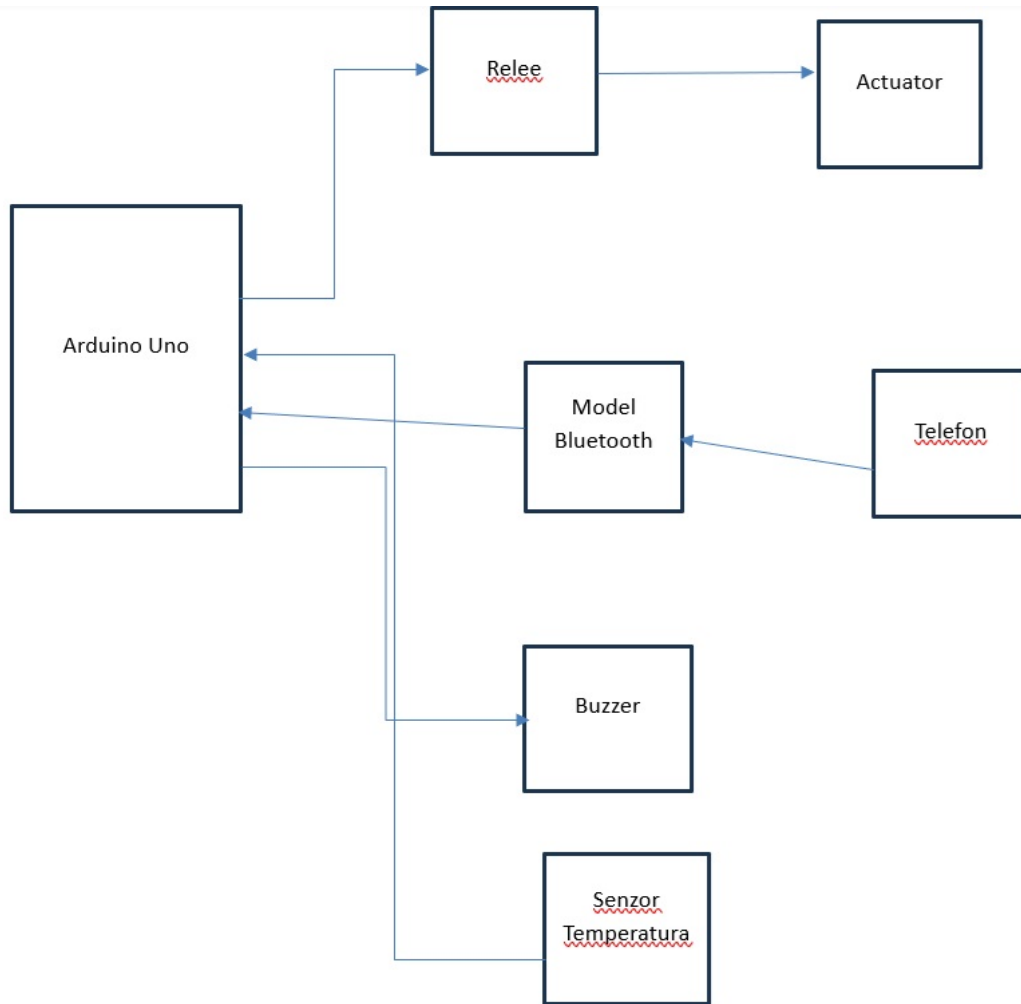
# Voice Command Door Lock

## Introducere

Proiectul meu este un sistem de inchidere si deschidere a zavorului unei usi, controlat prin comenzi vocale. Scopul proiectului este de a oferi o solutie simpla pentru sistemul de inchidere si deschidere a usilor pentru persoanele cu diverse dizabilitati , dar si pentru a facilita intrarea in casa cu un bagaj sau carand ceva de exemplu.

## Descriere generală

Zavorul se actioneaza prin comenzile "Lock" si "Unlock" prin actuator. Modulul Bluetooth legat la placa Arduino este conectat la telefon si prin intermediul sau sunt trimise comenzile . Zavorul va fi actionat doar la auzirea celor 2 comenzi. La introducerea unei comenzi gresite va fi activa buzzerul, de asemenea cand senzorul va detecta ca a trecut de temperatura stabilita va fi activat buzzerul si zavorul va fi deschis.



## 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 de piese:

- \* Arduino UNO
- \* HC-05 Bluetooth Module
- \* 4 x 12volt relee
- \* 2 x BD139 Tranzistoare
- \* Actuator de blocare a ușilor auto

- \* 2.2k rezistoare
- \* 12 Volt Adaptor
- \* Modul cu Buzzer activ
- \* Modul Senzor de Temperatură DHT11 cu LED

## 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 plănuieți să le implementați
- (etapa 3) surse și funcții implementate

Sistemul descris primește comenzi vocale de la un dispozitiv Android prin intermediul unui modul Bluetooth HC-05. Comenzile sunt procesate de un Arduino care acționează asupra diferitelor părți ale sistemului, cum ar fi un LED, un senzor de temperatură DHT11, un motor de blocare/deblocare și un buzzer. Sistemul utilizează biblioteca SoftwareSerial pentru comunicarea serială și biblioteca DHT pentru citirea temperaturii.

Funcționalități și componente principale:

### 1. LED-ul de pe pinul 13:

Folosit pentru a indica primirea de date. LED-ul se aprinde atunci când se primesc date și se stinge după procesarea acestora. Pini 11 și 12:

- Pinul 11 (unlockPin): Folosit pentru deblocarea sistemului.
- Pinul 12 (lockPin): Folosit pentru blocarea sistemului.
- Buzzer-ul conectat la pinul 6:

Folosit pentru a emite sunete în funcție de diferite condiții. Activează buzzerul la frecvența de 400 Hz pentru 1 secundă atunci când se primește o comandă necunoscută sau când temperatura depășește pragul setat.

- Senzorul de temperatură DHT11:

Conectat la pinul 2, acest senzor măsoară temperatura mediului. Dacă temperatura depășește pragul de 30 de grade Celsius, sistemul deblochează ușa și activează buzzerul. Comunicare Bluetooth:

Realizată prin modulul HC-05 conectat la pini 10 și 9 ai Arduino-ului. Sistemul primește comenzi vocale printr-o aplicație Android. Comunicarea cu PC-ul se face prin PuTTY deoarece iPhone-ul nu suportă conectarea directă la HC-05. Explicație detaliată a funcționării: Inițializare:

În funcția setup(), sunt setați toți pini ca ieșiri și se inițializează starea lor la LOW pentru a preveni activarea nedorită a buzzerului la pornire. Se inițializează comunicarea serială la 9600 bps atât pentru Serial cât și pentru SoftwareSerial. Senzorul DHT11 este inițializat pentru a începe măsurătorile de

temperatură. Loop principal:

Dacă există date disponibile pe portul serial SoftwareSerial, LED-ul se aprinde, iar datele sunt citite și trimise înapoi ca ecou. Comenzile lock și unlock sunt procesate pentru a activa pinii de blocare și deblocare, respectiv. Dacă se primește o comandă necunoscută, buzzerul emite un sunet de 400 Hz timp de 1 secundă. Citirea temperaturii:

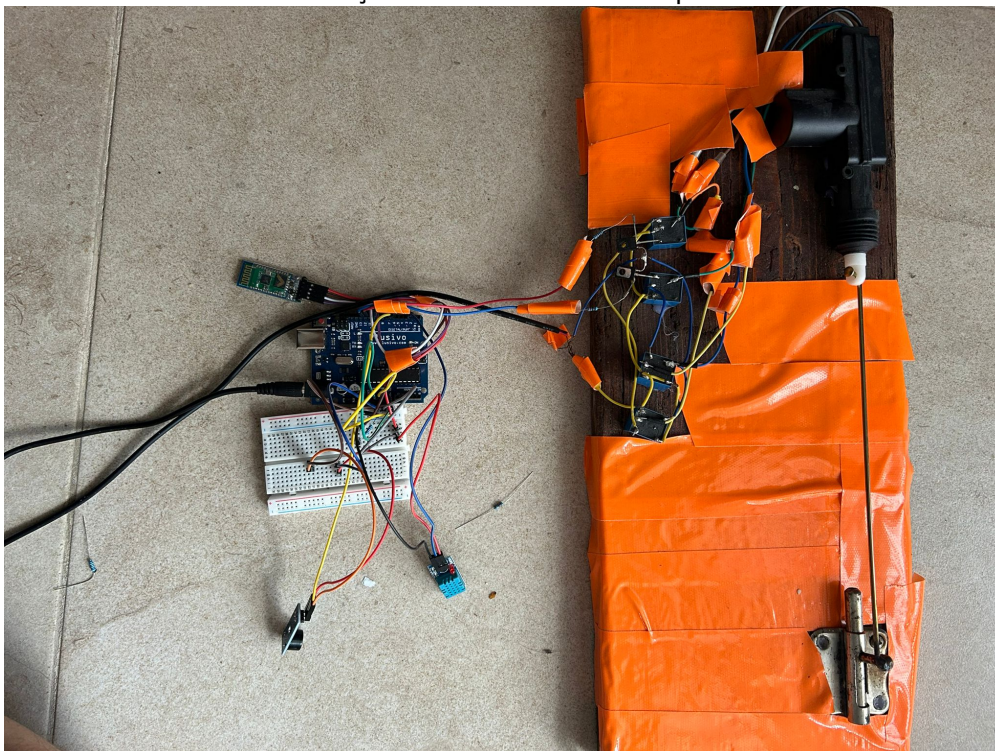
La fiecare iterație a buclei principale, temperatura este citită de la senzorul DHT11. Dacă citirea este validă, temperatura este afișată în monitorul serial. Dacă temperatura depășește pragul de 30 de grade Celsius, sistemul deblochează ușa și activează buzzerul timp de 3 secunde. Mediul de dezvoltare și biblioteci utilizate: Mediu de dezvoltare: Arduino IDE Biblioteci:

- SoftwareSerial.h pentru comunicarea serială cu modulul Bluetooth HC-05.
- DHT.h pentru interfațarea cu senzorul de temperatură și umiditate DHT11.

[sketch\\_may25a.zip](#)

## Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.



VideoclipProiect [videoproiect.zip](#)

## Concluzii

Deși proiectul a atins obiectivele pe care mi le-am setat, nu sunt pe deplin mulțumit de rezultat. Principalul motiv este că iPhone-ul are standarde diferite pentru dispozitivele Bluetooth, ceea ce a făcut imposibilă utilizarea modului HC-05 cu iPhone-ul.

### 1. Incompatibilitatea iPhone-ului cu HC-05:

iPhone-ul are standarde Bluetooth diferite, ceea ce a însemnat că nu am putut folosi modulul HC-05 pentru a trimite comenzi vocale de pe iPhone. A trebuit să folosesc PuTTY pentru a trimite comenzi de pe un PC, ceea ce a complicat lucrurile.

## Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună 😊.

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume\_student** (dacă este cazul).  
**Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru\_alin**.

[{{:pm:prj2024:vstoica:sketch\\_may25a.zip}}](#)

## Jurnal

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

## Bibliografie/Resurse

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

[Export to PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/vstoica/david.paduretu>



Last update: **2024/05/27 10:28**