

**PROIECT DE DIPLOMĂ**

**Conducător științific:**

**Prof. dr. ing. NICULA Dan**

**Absolvent:**

**Tudorache Robert-Cosmin**

**BRAȘOV, 2023**

**Departamentul: Electronică și Calculatoare**

**Programul de studii: Calculatoare**

***TUDORACHE Robert-Cosmin***

Sistem inteligent de deschidere a ușii

**Conducător științific:**

Prof. dr. ing. *NICULA Dan*

**Cuprins**

Cuprins

[Listă de figuri, tabele și coduri sursă 4](#_Toc138792193)

[Listă de acronime 7](#_Toc138792194)

[1 Introducere 9](#_Toc138792195)

[1.1 Tema proiectului 9](#_Toc138792196)

[1.2 Scopul proiectului 9](#_Toc138792197)

[1.3 Stadiul actual 10](#_Toc138792198)

[2 Arhitectura hardware 18](#_Toc138792199)

[2.1 Placa de dezvoltare Arduino Uno 18](#_Toc138792200)

[2.2 Componente utilizate 22](#_Toc138792201)

[2.2.1 Modulul RFID RC522 23](#_Toc138792202)

[2.2.2 Modulul Bluetooth HC-05 25](#_Toc138792203)

[2.2.3 Servomotor SG90 Micro Servo 28](#_Toc138792204)

[2.2.4 OLED Display I2C SSD1306 29](#_Toc138792205)

[2.2.5 LED 31](#_Toc138792206)

[2.2.6 Buzzer activ 5V 32](#_Toc138792207)

[2.2.7 Componente auxiliare 33](#_Toc138792208)

[3 Arhitectura software 35](#_Toc138792209)

[3.1 Configurare Arduino 35](#_Toc138792210)

[3.1.1 Mediul de dezvoltare Arduino IDE 35](#_Toc138792211)

[3.1.2 Programare microcontroller 36](#_Toc138792212)

[3.1.3 Comenzi AT proprii 41](#_Toc138792213)

[3.2 Aplicația Mobilă 42](#_Toc138792214)

[3.2.1 Mediul de dezvoltare Mit App Inventor 42](#_Toc138792215)

[3.2.2 Realizarea aplicatiei prin Mit App Inventor 43](#_Toc138792216)

[4 Implementare și testare 52](#_Toc138792217)

[5 Concluzii 58](#_Toc138792218)

[6 Bibliografie 59](#_Toc138792219)

[Rezumat 62](#_Toc138792220)

[Abstract 63](#_Toc138792221)

[Cod sursa 64](#_Toc138792222)

# Listă de figuri, tabele și coduri sursă

FIGURI

Figura 1. Diagramă a funcționării etichetei RFID în contact cu cititorul aferent.

Figura 2. Microcipul și antena din interiorul unui tag RFID.

Figura 3. Exemplu de cablare a modulului RFID la placa de dezvoltare Arduino Uno.

Figura 4. Citirea unei etichete RFID folosind fișierul de test MFRC522.

Figura 5. Sisteme biometrice de deschidere a ușii prin recunoaștere facială, recunoaștere a amprentelor și recunoaștere a irisului.

Figura 6. Sistem inteligent de deschidere a ușii folosind recunoașterea vocală.

Figura 7. Sistem inteligent de deschidere a ușii folosind codul PIN.

Figura 8. Sistem inteligent de deschidere a ușii folosind aplicații mobile, tehnologia NFC.

Figura 9. Diagrama de ansamblu a proiectului.

Figura 10. Arduino Uno.

Figura 11. Modulul RFID RC522 cu pinii aferenți.

Figura 12. Modulul Bluetooth HC-05 cu pinii aferenți.

Figura 13. Stabilire comunicare între telefon mobil și modulul Bluetooth HC-05.

Figura 14. SG90 Micro Servo 9g cu firele aferente.

Figura 15. OLED Display I2C SSD1306 cu pinii aferenți.

Figura 16. LED-uri normale 5mm de culoarea verde, respectiv roșu.

Figura 17. Buzzer activ cu alimentare la 5V.

Figura 18. Firele jumper utilizate și breadboard.

Figura 19. Apariția mediului de dezvoltare Arduino IDE.

Figura 20. Librăriile folosite pentru realizarea proiectului.

Figura 21. Definirea pinilor, obiectelor și variabilelor globale după inițializarea bibliotecilor corespunzătoare.

Figura 22. Configurare stare inițială a sistemului.

Figura 23. Funcția void loop().

Figura 24. Funcția readRFID() din interiorul funcției principale loop().

Figura 25. Funcții pentru citirea tagului modulului RFID RC522.

Figura 26. Verificarea modificărilor aduse modulului Bluetooth.

Figura 27. Mediul de programare vizual MIT App Inventor.

Figura 28. Crearea unui proiect nou utilizând MIT App Inventor.

Figura 29. Interfața Designer a mediului de programare vizual MIT App Inventor.

Figura 30. Instanțiere “Label”.

Figura 31. Rezultatul configurării.

Figura 32. Interfața grafică finală a aplicației.

Figura 33. Interfață blocuri MIT App Inventor.

Figura 34. Inițializare ecran.

Figura 35. Selectare dispozitiv Bluetooth.

Figura 36. Actualizare text ”Label1”.

Figura 37. Trimitere text către plăcuța Arduino utilizând Bluetooth.

Figura 38. Obținerea textului utilizând recunoașterea vocală.

Figura 39. Verificarea textului obținut prin recunoaștere vocală.

Figura 40. Implementare buton de deconectare de la modulul Bluetooth.

Figura 41. Cablare modul RFID la Arduino.

Figura 42. Cablare LED-uri și Buzzer.

Figura 43. Cablare OLED.

Figura 44. Cablare servomotor.

Figura 45. Cablare modul Bluetooth.

TABELE

Tabel1. Listă comenzi AT pentru modul Bluetooth HC-05.

# Listă de acronime

AC - Alternating Current

APK - Android Package

APP - Application

AT - Attention

AWG - American Wire Gauge

BLE - Bluetooth Low Energy

BPS - Bits Per Second

CMOS - Complementary Metal-Oxide-Semiconductor

DC - Direct Current

EDR - Enhanced Data Rate

EEPROM - Electrically Erasable Programmable Read-Only Memory

EML - Emissive Layer

ETL - Electron Transport Layer

FIFO - First In First Out

GFSK - Gausian Frequency-Shift Keying

GND - Ground

GPS - Global Positioning System

GSM - Global System for Mobile

HID - Human Interface Device

HTL - Hole Transport Layer

I/O - IN/OUT

I2C - Inter-Integrated Circuit

IC - Integrated Circuit

ICSP - In-Circuit Serial Programming

ID - Identification

IDE - Integrated Development Environment

IoT - Internet of Things

ISM - Industrial, Scientific and Medical

ITO - Indium Tin Oxide

LCD - Liquid Crystal Display

LED - Light Emitting Diode

MCU - Microcontroller Unit

MISO - Master In Slave Out

MIT - Massachusetts Institute of Technology

MOSI - Master Out Slave In

MP3 - Mpeg Audio Layer-3

NFC - Near-Field Communication

OLED - Organic Light-Emitting Diode

OSCIN - Oscillator In

OSCOUT - Oscillator Out

PC - Personal Computer

PID - Proportional-Integral-Derivative controller

PIN - Personal Identification Number

PLED - Polimeric Light-Emitting Diode

PWM - Pulse Width Modulation

RAM - Random Access Memory

RFID – Radio Frequency Identification

RST - Reset

RX - Receiver

SCK - Serial Clock

SCL - Serial Clock Line

SDA - Serial Data

SPI - Serial Peripheral Interface

SPP - Serial Port Profile

SRAM - Static Random Access Memory

SS - Slave Select

TX - Transmitter

UART - Universal Asynchronous Receiver/Transmitter

UID - Unique Identifier

USB - Universal Serial Bus

VCC - Voltage Collector Current

VIN - Input Voltage

# Introducere

## Tema proiectului

Punctul central al temei mele de licență constă în crearea unui sistem inteligent de deschidere a ușii. Tehnologia pe care am ales să o folosesc pentru a construi un sistem de securitate și acces eficient include Arduino, RFID, Bluetooth și aplicații mobile.

Obiectivul principal al proiectului este de a dezvolta un sistem care să permită intrarea într-o zonă restricționată. Un card de acces pentru utilizator va fi programat pentru a fi citit de modulul RFID. Placa de dezvoltare Arduino Uno va activa un dispozitiv de acționare pentru a deschide ușa atunci când detectează un card valid.

Am creat o aplicație mobilă și am integrat un modul Bluetooth pentru a simplifica interacțiunea utilizatorului cu sistemul. Utilizatorii pot interacționa cu placa de dezvoltare Arduino Uno și pot transmite comenzi pentru deschiderea ușii prin intermediul aplicației mobile.

De asemenea, aplicația le oferă utilizatorilor posibilitatea de a utiliza comenzi vocale pentru a opera dispozitivul, ceea ce face ca procesul să fie mai simplu și mai ușor de utilizat. Această idee a pornit de la necesitatea de a oferi persoanelor, care își rătăcesc sau își uită tokenul de acces, o modalitate convenabilă și funcțională de a evita riscul de a rămâne blocate în astfel de circumstanțe neprevăzute.

Am adăugat, de asemenea, un buzzer și LED-uri care oferă un feedback atât sonor, cât și vizual. În funcție de eticheta RFID citită, soneria emite un sunet diferit pentru a indica accesul permis sau respins, iar LED-urile arată dacă ușa s-a deschis sau nu.

## Scopul proiectului

Scopul proiectului meu este de a dezvolta un sistem inteligent de deschidere a ușii care să ofere o metodă modernă și eficientă de control al accesului în zone limitate, cum ar fi propria locuință. Obiectivele principale ale acestui sistem sunt creșterea confortului și a securității utilizatorului.

Scopul punerii în aplicare a acestui sistem este de a face accesul autorizat mai ușor și mai sigur. Utilizatorii vor avea carduri de acces preprogramate pe care modulul RFID le poate citi. Ușa va putea fi deschisă doar de persoanele autorizate. Acest lucru reduce posibilitatea apariției unor situații nedorite și oferă o securitate mai puternică împotriva accesului neautorizat.

Un alt obiectiv al proiectului este acela de a oferi utilizatorilor o experiență simplă și directă de gestionare a sistemului. Aceștia vor putea să pornească sistemul și să deblocheze ușa cu ajutorul unei aplicații pentru smartphone. În plus, integrarea comenzilor vocale facilitează interacțiunea hands-free cu sistemul și îmbunătățește experiența utilizatorului.

Sistemul nu va oferi doar funcționalitatea fundamentală, ci și feedback sonor și vizual pentru a arăta starea sistemului și accesul autorizat sau neautorizat. Obiectivul final al proiectului este de a demonstra potențialul și utilitatea unor tehnologii precum Arduino, RFID, Bluetooth și aplicațiile mobile în domeniul securității și al accesului inteligent.

Scopul este de a oferi un sistem care poate fi implementat simplu, economic și care să poate fi personalizat în funcție de cerințele unice ale fiecărui utilizator.

## Stadiul actual

Sistemul de acces cu card RFID este una dintre caracteristicile și funcționalitățile actuale utilizate în sistemele inteligente de deschidere a ușilor.

RFID este o rețea electrică care permite comunicarea wireless, prin intermediul undelor radio, a două dispozitive. Un sistem RFID are două părți principale : un trasponder, cunoscut și sub numele de etichetă, care este plasat pe obiectul pe care dorim să-l identificăm, și un emițător-receptor, cunoscut și sub numele de cititor.[1]

O bobină de antenă care generează un câmp electromagnetic de înaltă frecvență, un dispozitiv de control și un modul de radiofrecvență alcătuiesc un cititor RFID. Eticheta, pe de altă parte, este de obicei echipată doar cu o antenă și un microcip electronic, ceea ce o face un element pasiv. Ca urmare, inducția determină formarea unei tensiuni în bobina antenei atunci când eticheta intră în contact cu câmpul electromagnetic al emițătorului-receptorului, care alimentează microcipul. [1]

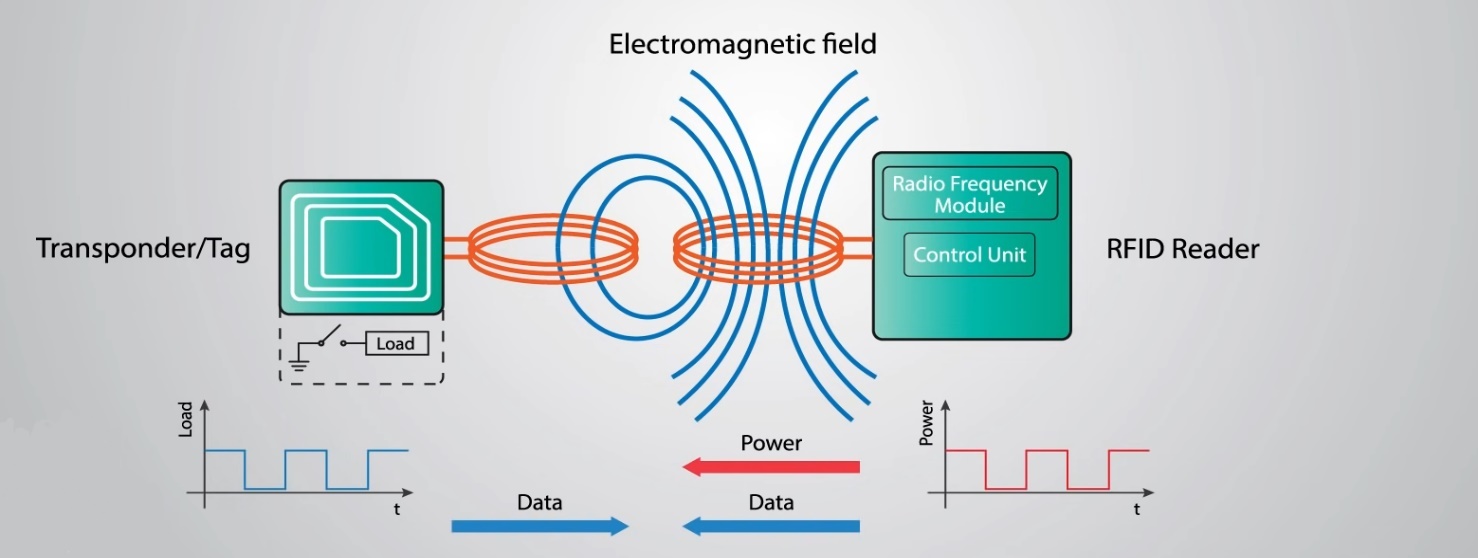


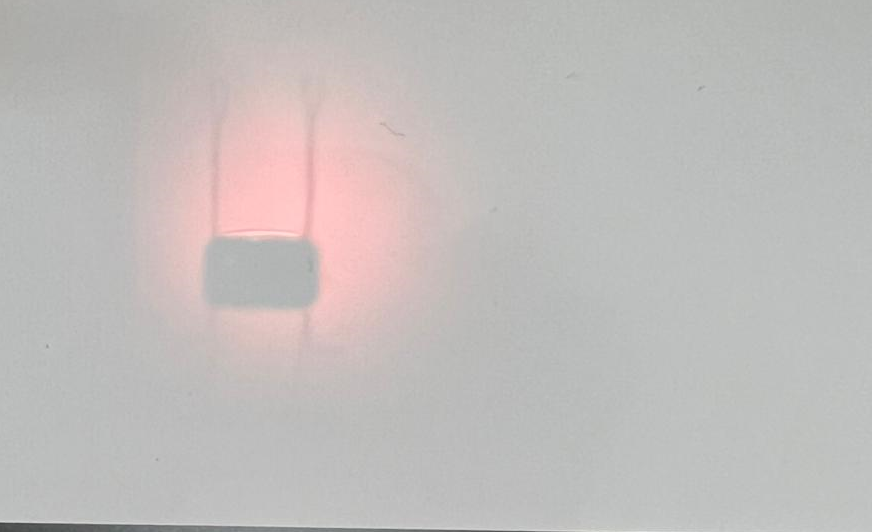
Figura 1. Diagramă a funcționării etichetei RFID în contact cu cititorul aferent. [2]

Odată ce eticheta a fost alimentată, aceasta poate extrage mesajul pe care cititorul l-a transmis și îl poate trimite înapoi la cititor printr-o tehnică numită manipulare a încărcăturii. Putem determina câtă energie consumă antena cititorului prin măsurarea căderii de tensiune care apare atunci când o sarcină este activată și dezactivată la antena etichetei. În acest mod, datele sunt trimise de la etichetă la cititor, variațiile de tensiune fiind înregistrare sub formă de unu și zero. [2]

Metoda de cuplare prin retrodifuziune este o altă abordare utilizată în paralel pentru a transmite date de la etichetă la cititor. În acest caz, o parte din energia absorbită de etichetă este utilizată pentru a produce un câmp electromagnetic suplimentar pe care antena cititorului îl poate capta. [2]

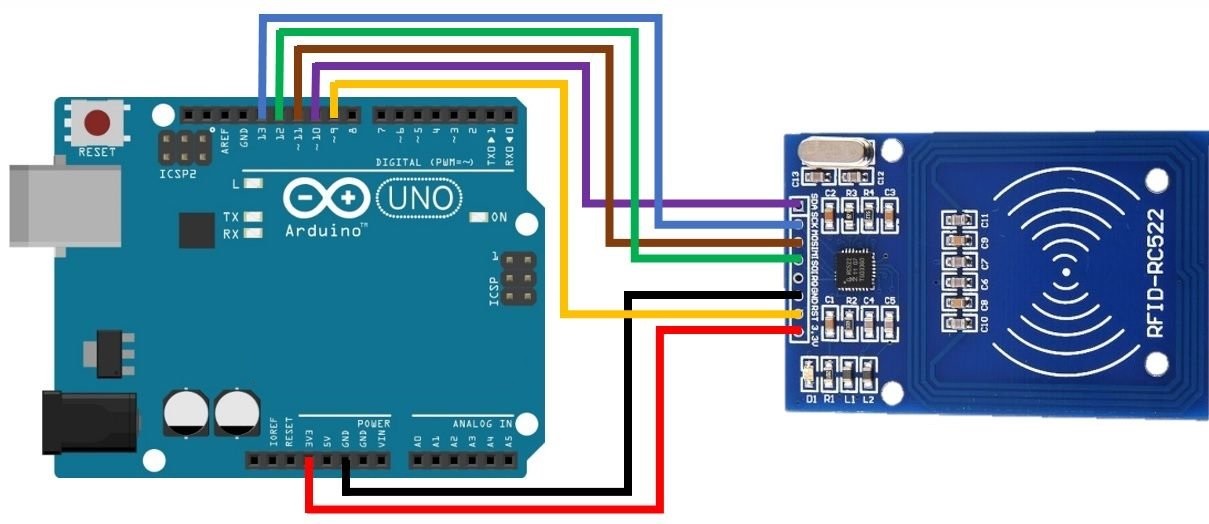
După ce am stabilit principiile fundamentale de funcționare, vom examina acum modul în care RFID poate fi utilizat împreună cu Arduino. Pentru a construi un astfel de sistem utilizând Arduino, se va achiziționa un modul RFID care să conțină etichete MIFARE și cititorul asociat.

Aceste etichete au un microprocesor cu 1KB de memorie și capacitatea de a efectua funcții matematice. În funcție de geometria antenei, acestea pot funcționa la o distanță de până la 10cm, la frecvență de 13,56Mhz. Putem vedea antena și microcipul unei astfel de etichete dacă o ținem în fața unei surse de lumină. [2]



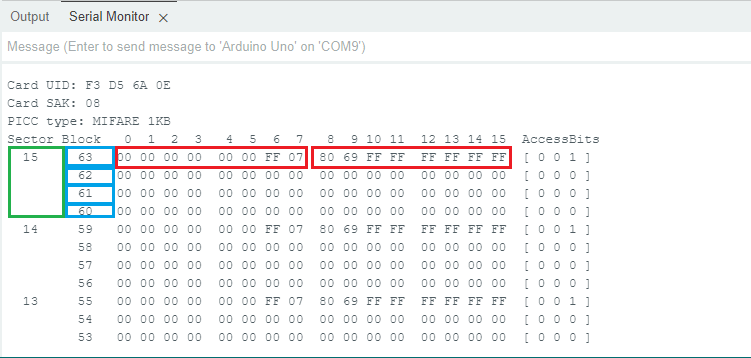
*Figura 2. Microcipul și antena din interiorul unei etichete RFID.*

Protocolul SPI este utilizat de modulul cititor RFID pentru a se conecta la placa de dezvoltare Arduino Uno. VCC-ul modulului trebuie să fie conectat la sursa de alimentare de 3,3V a plăcii de dezvoltare Arduino Uno. [2]

**

*Figura 3. Exemplu de cablare a modulului RFID la placa de dezvoltare Arduino Uno.*

Odată ce modulul este conectat, trebuie descărcată biblioteca aferentă tipului de modul RFID. Pentru a verifica dacă sistemul funcționează corespunzător, putem încărca pe placa de dezvoltare Arduino Uno exemplul din biblioteca aferentă modulului RFID, care se află, de obicei, sub denumirea de “Dumpinfo”. Cititorul va începe să citească eticheta și toate datele de pe aceasta vor fi afișate pe monitorul serial din mediul de dezvoltare Arduino IDE imediat ce eticheta este plasată în apropierea modulului



*Figura 4. Citirea unei etichete RFID folosind fișierul de test MFRC522.*

Figura 4 arată numărul UID al etichetei, precum și faptul că memoria de 1KB a acesteia este împărțită în 16 sectoare, fiecare dintre acestea fiind compus din 4 blocuri care pot stoca până la 2 bytes de date. [2]

În continuare vom vorbi despre sisteme inteligente de deschidere a ușii ce folosesc senzori biometrici. Aceste sisteme se bazează pe tehnologii de recunoaștere biometrică, care permit identificarea și autentificarea persoanelor pe baza unor trăsături fizice distinctive, cum ar fi amprentele digitale, recunoașterea facială, scanarea irisului și alte caracteristici biometrice.

Utilizarea senzorilor biometrici pentru a capta și stoca caracteristicile biometrice ale utilizatorilor autorizați este un aspect major al acestor sisteme inteligente de deschidere a ușii.

Acești senzori pot fi dispozitive specializate, precum cititoare de amprente, camere pentru recunoașterea facială sau scanere de iris, care colectează și procesează informațiile necesare pentru identificare.



*Figura 5. Sisteme biometrice de deschidere a ușii prin recunoaștere facială, recunoaștere a amprentelor și recunoaștere a irisului. [3] [4] [5]*

Pentru scanarea amprentelor digitale se utilizează un cititor de amprente care folosește un senzor optic sau un senzor capacitiv. Senzorul măsoară variațiile de presiune sau scanează suprafața încrețită a pielii de pe vârful degetelor pentru a obține o imagine a amprentei. Această imagine este apoi prelucrată și convertită într-o reprezentare digitală a amprentei, numită "șablon". [6]

Scanarea facială implică utilizarea unei camere sau a unui sistem de camere pentru a captura imagini ale feței unei persoane. Aceste imagini sunt prelucrate pentru a identifica trăsăturile distinctive, cum ar fi forma feței, ochii, nasul și buzele. Aceste caracteristici sunt extrase cu ajutorul unor algoritmi complexi, reprezentate sub formă de șabloane sau modele, iar apoi sunt comparate cu șabloanele păstrate în baza de date pentru a vedea dacă există o potrivire. [6]

Pentru scanarea irisului, se utilizează un scanner specializat care utilizează lumina infraroșie pentru a obține o imagine detaliată a irisului ochiului. Irisul este structura colorată și texturată a ochiului și are trăsături unice pentru fiecare persoană. Imaginile capturate sunt procesate pentru a extrage caracteristicile relevante ale irisului și pentru a crea un șablon sau o reprezentare digitală a acestora. [7]

Prin urmare, procesul de recunoaștere biometrică în sistemele de deschidere a ușii implică compararea caracteristicilor biometrice prezentate cu cele stocate într-o bază de date securizată. Dacă există o potrivire acceptabilă între caracteristicile biometrice prezentate și cele stocate, accesul este autorizat și ușa se deschide automat. În caz contrar, accesul este respins și ușa rămâne închisă.



*Figura 6. Sistem inteligent de deschidere a ușii folosind recunoașterea vocală. [8]*

Conform unui studiu privind preferințele consumatorilor în materie de securitate a locuinței, piața deschizătoarelor inteligente de uși care utilizează identificarea vocală este în continuare în expansiune.

Realizarea unui model digital al vocii unei persoane, care poate fi folosit ca șablon sau profil salvat pentru acea amprentă vocală, face posibilă tehnologia de recunoaștere vocală. Tiparele de frecvență care alcătuiesc cuvintele si frazele descriu în mod colectiv modelul distinctiv de vorbire al unei persoane. Șabloanele sunt păstrate în baza de date pentru a fi comparate cu alte tipuri de informații biometrice. [9]

Aceste sisteme, care pot fi utilizate, de exemplu, pentru a controla accesul, pot fi dependente sau independente de text (rareori o combinație). În cazul primei opțiuni, cuvintele sau numerele care sunt rostite ca parole pot fi comparate cu un eșantion de cuvinte care a fost colectat în timpul înscrierii. Tehnologia independentă de text scanează discursul liber al vorbitorului pentru a găsi trăsături vocale distinctive, în loc să solicite o frază de trecere. [9]

Pe lângă faptul că sunt acceptate în mod universal de către clienți, dispozitivele de verificare vocală sunt, de asemenea, ușor de utilizat și, comparativ, mai puțin costisitoare decât alte opțiuni biometrice.

Analizând piața, observăm că există și sisteme inteligente de deschidere a ușii ce folosesc codul PIN. Acestea sunt soluții comune și practice pentru accesul securizat în diverse medii. Aceste sisteme se bazează pe faptul că utilizatorul autorizat este singurul care cunoaște codul numeric secret utilizat pentru a debloca ușa.

Introducerea codului, validarea codului și obținerea accesului sunt etapele necesare procesului de deblocare a ușii cu ajutorul codului PIN.

Introducerea codului PIN presupune ca utilizatorul să introducă codul PIN într-un dispozitiv sau o tastatură numerică dedicată, numită și tastatură PIN sau tastatură numerică. Aceasta poate fi o tastatură fizică montată lângă ușă sau o tastatură virtuală afișată pe un panou de control. Validarea codului PIN presupune ideea că dispozitivul de acces verifică corectitudinea codului PIN introdus de utilizator. Codul introdus este comparat cu unul care este stocat intern într-o bază de date sau cu alte strategii de autentificare, cum ar fi algoritmii de criptare. Accesul autorizat presupune ca sistemul de acces să deblocheze ușa, permițând utilizatorului să intre în spațiu, dacă codul PIN introdus corespunde celui stocat. În caz contrar, intrarea este interzisă, iar ușa rămâne închisă. [10]

În concluzie, sistemele de deschidere a ușilor bazate pe coduri PIN reprezintă o abordare utilă și eficientă pentru a oferi acces securizat în diferite medii. Acestea oferă un nivel rezonabil de securitate, sunt simplu de utilizat și permit o flexibilitate în gestionarea accesului.



Figura 7. Sistem inteligent de deschidere a ușii folosind codul PIN. [11]

Un alt sistem de acces foarte popular pe piață sunt sistemele inteligente de deschidere a ușii ce folosesc smartphone-ul și aplicațiile mobile.

De cele mai multe ori, controlul mobil al accesului utilizează Bluetooth pentru a conecta telefoanele la cititoarele instalate pe uși. În prezent, nu mai este necesară împerecherea manuală a dispozitivelor. De exemplu, tehnologia Bluetooth Low Energy (BLE) se va ocupa de această operațiune destul de iritantă. Accesul prin Bluetooth oferă un avantaj semnificativ față de tehnologiile tradiționale de carduri RFID, prin faptul că o conexiune sigură poate fi realizată de la o distanță de până la câțiva metri, spre deosebire de raza medie de citire de aproximativ cinci centimetri pentru tehnologiile convenționale de carduri RFID. Cardurile de control al accesului nu mai sunt necesare datorită controlului mobil al accesului.

Controlul accesului mobil mai utilizează frecvent și tehnologia NFC (Near-Field Communication), aceasta fiind asociată de cele mai multe ori cu aplicațiile de plată. Ca și în cazul Bluetooth, un telefon se poate conecta la un cititor, însă un dezavantaj al NFC este că interzice conexiunile la distanță. Aceasta are doar o rază de acțiune de câțiva centimetri. [12]



*Figura 8. Sistem inteligent de deschidere a ușii folosind aplicații mobile, tehnologia NFC**. [12]*

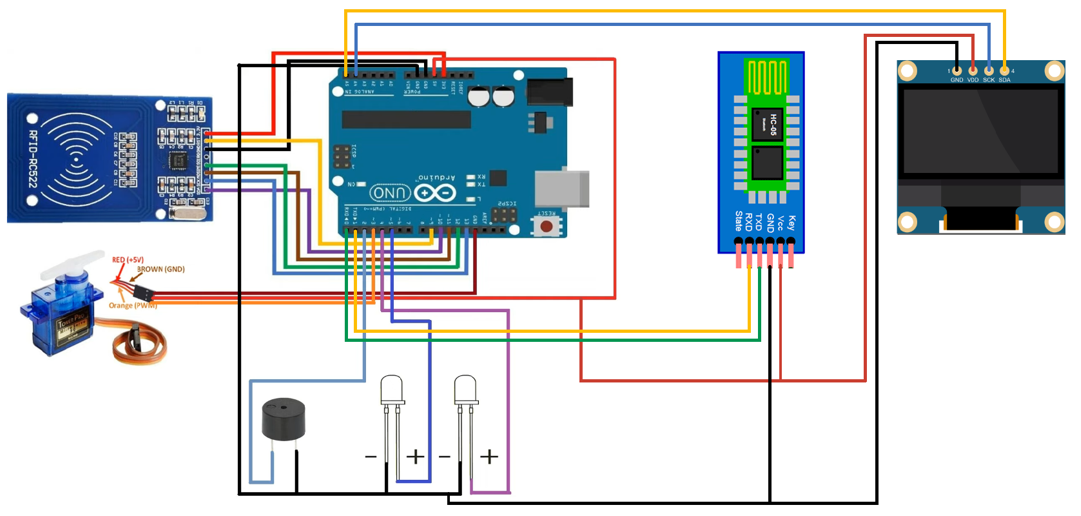
# Arhitectura hardware

## Placa de dezvoltare Arduino Uno

Arhitectura hardware a proiectului meu de licență se bazează pe utilizarea plăcii de dezvoltare Arduino Uno, ce găzduiește un microcontroller ATmega328p, un modul RFID-RC522, două LED-uri, un buzzer și un servomotor. Comunicația dintre plăcuța de dezvoltare Arduino Uno și aplicația mobilă se realizează prin intermediul unui modul Bluetooth, mai exact modulul HC-05. Pe un afișaj I2C OLED sunt prezentate mesajele de acces.

Sistemul este alimentat prin conectarea plăcii de dezvoltare la portul USB al unui laptop, care creează tensiunea de 5V necesară pentru funcționarea microcontrollerului. Modulul Bluetooth, OLED display-ul, buzzerul și LED-urile sunt alimentate la 5V, acestea fiind distribuite pe un breadboard, iar modulul RFID este alimentat la tensiune de 3,3V de la pinul aferent plăcuței de dezvoltare.

În imaginea de mai jos (Figura 9.), am reprezentat diagrama de ansamblu a sistemului și modul în care am interfațat toate componentele. Partea de aplicație mobilă prin Bluetooth va fi dezvoltată ulterior într-un alt capitol, modulul Bluetooth HC-05 fiind cel care realizează această legătură între sistemul hardware și aplicația software. Viziunea mea a fost ca dispozitivul să aibă o dimensiune cât mai compactă și să ofere o experiență aproape de dispozitivele existente pe piață.



*Figura 9. Diagrama de ansamblu a proiectului.*

Arduino Uno este o placă de dezvoltare cu dimensiuni reduse care poate fi integrată într-o multitudine de proiecte. Aceasta a fost creată de Arduino.cc în Italia și funcționează cu procesorul ATmega328p, care are capacități de comunicare USB, UART, I2C și SPI, pe lângă o multitudine de porturi I/O digitale și analogice. [13]

ATmega328p dispune de un modul UART (Universal Asynchronous Receiver/Transmitter) care permite comunicarea serială asincronă. Acesta este utilizat frecvent pentru a comunica cu dispozitive externe, cum ar fi senzori, GPS sau module GSM și poate fi folosit pentru a trimite și primi date în modul serial. [14]

Microcontrollerul oferă și suport pentru interfața I2C (Inter-Integrated Circuit), care este un protocol serial sincron utilizat pentru comunicații pe scurtă distanță între microcontrollere și diferite componente, cum ar fi senzori, afișaje LCD sau module de memorie. Pe aceeași linie de comunicare pot fi utilizate mai multe dispozitive, iar comunicarea bidirecțională este suportată prin intermediul acestei interfețe. [14]

Interfața periferică serială (SPI), un protocol de comunicare serială sincronă utilizat pentru transferul rapid de date între dispozitive precum senzori, afișaje grafice sau module de stocare, este de asemenea inclusă în ATmega328p. Comunicarea full-duplex este suportată de SPI, permițând transmiterea și recepționarea simultană a datelor. [14]

De asemenea, ATmega328p oferă și pini digitali de intrare/ieșire care pot fi configurați pentru comunicație serială prin software. Acest lucru înseamnă că microcontrollerul poate fi programat pentru a implementa protocoale de comunicație personalizate sau pentru a interacționa cu dispozitive care utilizează protocoale specifice dezvoltate de utilizator. [14]

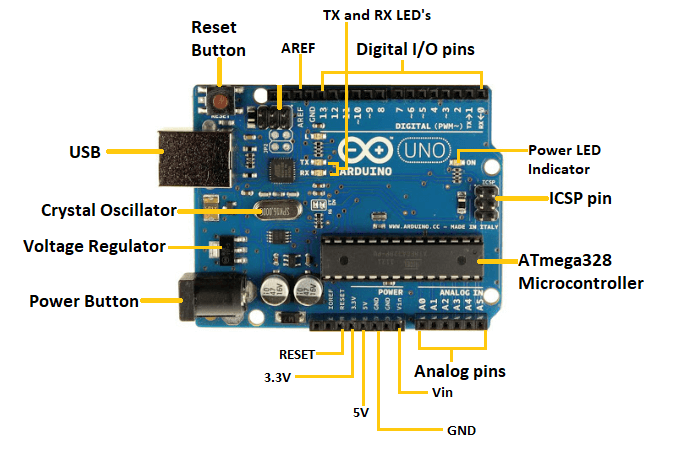
Placa de dezvoltare Arduino Uno are, de asemenea, o serie de caracteristici încorporate, cum ar fi un regulator de tensiune, un buton de resetare și un LED de stare. Placa include, de asemenea, un conector ICSP (In-Circuit Serial Programming) care permite programarea directă a microcontrollerului cu ajutorul unui programator extern. Utilizarea unui programator extern poate oferi o flexibilitate suplimentară și un control mai mare asupra procesului de programare. Acest lucru poate fi util în dezvoltarea avansată, depanarea sau în situațiile în care este necesară o programare personalizată. Cu toate acestea, în majoritatea cazurilor, programarea prin intermediul mediului Arduino IDE și al cablului USB este suficientă și ușor de utilizat fără a necesita un programator extern separat. [13]

Pentru a ușura dezvoltarea proiectelor, Arduino Uno suportă programarea C/C++ și oferă o gamă largă de biblioteci și exemple de coduri. Compatibilitatea sa cu mai multe sisteme de operare, inclusiv Windows, macOS și Linux, oferă utilizatorilor libertatea de a selecta mediul de programare preferat. Utilizatorii pot construi o gamă largă de proiecte cu placa Arduino Uno, de la dispozitive de robotică și automatizare la sisteme de monitorizare și control al mediului. Arduino Uno oferă o modalitate economică și atractivă de a descoperi și experimenta în domeniul electronicii și al programării cu ajutorul comunității Arduino și al resurselor online. [13]

Arduino Uno dispune de o memorie Flash de 32KB și funcționează la o frecvență de 16MHz. Acesta poate fi alimentat printr-o conexiune USB sau printr-un adaptor de alimentare extern. Platforma Arduino Uno este compatibilă cu o varietate de module și senzori, oferind utilizatorilor posibilitatea de a crea proiecte electronice complexe. Arduino Uno este instrumentul perfect pentru a experimenta și a crea proiecte electronice de orice complexitate, datorită asistenței unei comunități de utilizatori vibrante și a mediului de programare simplu și ușor de înțeles oferit de Arduino IDE. [13]

Caracteristici generale Arduino Uno : [16]

* Microcontroller: ATmega328p.
* Tensiune de operare: 5V.
* Tensiune de intrare recomandată: 7-12V.
* Tensiune de intrare limită: 6-20V.
* Pini digitali I/O: 14, din care 6 pot fi utilizați ca ieșiri PWM .
* Pini analogici de intrare: 6.
* Curent DC (Direct Current) per pin I/O: 20mA.
* Curent DC pentru pinul 3.3V: 50mA.
* Memorie Flash: 32 KB (din care 0.5KB sunt utilizați pentru bootloader).
* SRAM: 2KB.
* EEPROM: 1KB.
* Viteza de ceas (Clock Speed): 16 MHz.



*Figura 10. Arduino Uno. [15]*

Aceste specificații descriu caracteristicile hardware ale plăcii Arduino Uno. Microcontrollerul ATmega328p este inima plăcii, asigurând procesarea instrucțiunilor și gestionarea interacțiunii cu componente externe. Tensiunea de operare de 5V indică tensiunea de lucru a plăcii și a componentelor conectate la ea. Tensiunea de intrare recomandată este intervalul de tensiuni la care placa funcționează optim, iar tensiunea de intrare limită este intervalul în care placa poate funcționa în siguranță.

Plăcuța de devoltare Arduino Uno oferă un număr generos de 14 pini digitali de intrare/ieșire, care pot fi utilizați pentru a citi sau controla semnale digitale. Dintre aceștia, 6 pot fi utilizați ca ieșiri PWM, permițând controlul precis al intensității sau vitezei unor dispozitive. De asemenea, există 6 pini analogici de intrare care permit citirea valorilor analogice de la senzori sau alte surse. [16]

Curentul DC pe pinul I/O indică cantitatea maximă de curent pe care un pin digital poate furniza sau accepta. Este important să nu se depășească această valoare pentru a proteja placa și componentele externe. Pinul de 3,3V poate furniza o tensiune de 3,3V și un curent de până la 50 mA, potrivit pentru alimentarea componentelor care necesită această tensiune redusă. [16]

Placa de dezvoltare Arduino Uno dispune de 32KB de memorie Flash, care este utilizată pentru stocarea codului sursă al programelor. Din această capacitate, 0,5KB sunt rezervați pentru bootloader, care permite încărcarea programelor în placa Arduino. De asemenea, placa are 2KB de memorie SRAM pentru stocarea datelor temporare și 1KB de memorie EEPROM, care poate păstra date chiar și după întreruperea alimentării. [16]

Viteza de ceas de 16MHz indică frecvența de bază la care microcontrollerul procesează instrucțiunile și execută operațiunile. Aceasta determină rata de execuție a programelor și influențează temporizările și sincronizările în aplicații. [16]

Aceste specificații tehnice oferă o imagine de ansamblu a capacităților și limitărilor plăcii Arduino Uno, permițând dezvoltatorilor și entuziaștilor să programeze și să interacționeze cu o varietate de componente și senzori pentru a crea diverse aplicații și proiecte electronice.

Am ales plăcuța de dezvoltare Arduino Uno pentru proiectarea sistemului inteligent de deschidere a ușii pentru că aceasta are un cost accesibil, iar microcontroller-ul ATmega328p este arhisuficient pentru dezvoltarea acestei aplicații.

Descrierea pinilor Arduino Uno: [16]

* **VIN**, pin folosit pentru a alimenta placa Arduino Uno de la o sursă externă cu o tensiune de 5V.
* **5V**, este un pin care funcționează ca un regulator de tensiune al plăcii Arduino, folosit pentru a controla și a alimenta componente externe.
* **3.3V**, este un pin asemănător cu cel de 5V, dar la o tensiune mai mică.
* **GND**, este un pin care are rolul de împământare. Arduino Uno prezintă doi astfel de pini.
* **RST**, este un pin folosit pentru a reseta placa. Este foarte util atunci când un program încărcat pe placă este foarte complex, încât este nevoie de resetare.
* **Pinii cu intrare analogică**, sunt numerotați de la A0 la A5 și sunt folosiți pentru a furniza o intrare analogică în intervalul 0-5V.
* **Pinii I/O**, pinii digitali de la 0 la 13 care pot fi folosiți ca pini de ieșire sau de intrare.
* **RX,TX**, sunt doi pini folosiți pentru comunicarea pe serială. TX-ul reprezintă transmisia de date, iar RX-ul reprezintă receptorul de date.

Pentru comunicația I2C sunt folosiți pinii A4 și A5. Pinul A4 reprezintă linia de date seriale (SDA) cu rolul de a transporta datele pe bus-ul I2C, iar pinul A5 reprezintă linia de ceas (SCL), care este generată de dispozitivul master și care ajută la sincronizare. Pentru comunicația pe SPI sunt folosiți pinii 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). [16]

## Componente utilizate

Pentru a crea sistemul inteligent de deschidere a ușii, am integrat o serie de componente care lucrează împreună pentru a oferi funcționalitatea dorită. Pentru a permite accesul utilizatorilor, se utilizează un modul RFID care este capabil să scaneze și să recunoască etichetele RFID autorizate. Aceste etichete pot fi, de exemplu, brelocuri de chei sau carduri unice. Sistemul trimite un semnal către dispozitivul de acționare pentru a deschide ușa atunci când detectează o etichetă validă.

În scopul facilitării interacțiunii și oferirii unui feedback adecvat utilizatorului, proiectul implementează un dispozitiv acustic, cunoscut sub denumirea de buzzer, care generează semnale sonore distincte pentru a indica starea de deblocare sau respingere a ușii. În plus, două LED-uri sunt utilizate pentru a furniza feedback vizual suplimentar utilizatorului. Aceste LED-uri sunt activate într-un mod specific, indicând astfel accesul permis sau refuzat. Astfel, prin utilizarea acestor elemente, se asigură un feedback intuitiv și clar pentru utilizator în ceea ce privește starea și rezultatul procesului de acces la ușă.

În vederea facilitării comunicării și interacțiunii cu utilizatorii, proiectul utilizează un modul Bluetooth HC-05, care oferă o interfață wireless pentru conectarea dispozitivelor mobile. Acest modul permite transferul datelor și comenzilor între sistemul de control și dispozitivul mobil, permițând astfel controlul și monitorizarea ușii inteligente prin intermediul unei aplicații mobile.

De asemenea, sistemul este echipat cu un ecran OLED I2C SSD1306 care prezintă informații relevante și mesaje către utilizatori. Acest display oferă un afișaj compact și ușor de citit, permițând afișarea clară a textelor și a altor elemente grafice.

### Modulul RFID RC522

RC522 RFID este un modul de citire-scriere RFID (Radio Frequency Identification) care se utilizează în proiecte electronice pentru comunicarea cu etichetele RFID. Este un modul compact și ușor de integrat, care oferă funcționalități de bază pentru citirea și scrierea datelor pe etichetele RFID. (Figura 11.)

Acesta este alcătuit din 3 componente cheie, acestea fiind un circuit integrat MFRC522, un oscilator cu cristal de 27,12MHz și o antenă.

Circuitul integrat MFRC522 reprezintă componenta centrală a modulului RFID RC522 și are rolul de a permite citirea și scrierea cardurilor RFID. Acesta a fost dezvoltat de compania NXP și utilizează un canal de comunicare fără contact la o frecvență de 13,56MHz. Avantajele sale constă în consumul redus de energie, costurile și dimensiunile reduse. Circuitul integrat MFRC522 este compatibil cu diverse tipuri de etichete RFID, inclusiv MIFARE 1K, MIFARE 4K, MIFARE Mini și altele care se bazează pe protocolul ISO/IEC 14443. Este capabil să realizeze comunicare fără contact cu o viteză de până la 424kbps, oferind astfel o performanță rapidă în transferul datelor. Funcționarea circuitului MFRC522 se realizează la o frecvență de 13,56 MHz, iar raza de acțiune este de aproximativ 50 mm, în funcție de dimensiunea și frecvența antenei utilizate. Pentru comunicarea cu gazda, în cazul nostru microcontrollerul ATmega328p, circuitul integrat MFRC522 suportă protocoalele de comunicare seriale SPI, UART și I2C. [17]

Un cristal de cuarț cu frecvența de 27,12 MHz este utilizat în conexiune cu pinurile OSCIN și OSCOUT ale cipului oscilatorului intern. Acest cristal oferă semnalul de bază pentru generarea impulsului de ceas în cadrul sistemului. Prin divizarea acestui impuls de ceas de 27,12 MHz la jumătate, se obține frecvența dorită, 13,56Mhz, pentru operațiunile de comunicare și transfer de date în sistem. [17]

Placa de circuit imprimat a modulului RFID RC522 conține o bobină NFC care funcționează ca o antenă pentru transmiterea unui câmp electromagnetic de înaltă frecvență de 13,56MHz. Această antenă este proiectată să interacționeze cu componente pasive care sunt compatibile cu frecvența de 13,56MHz. Etichetele RFID, cum ar fi cardurile și brelocurile de chei, sunt exemple de astfel de componente pasive. Acestea nu au o sursă de energie internă și funcționează prin intermediul unui microcip și a bobinei NFC (antenă). Datele sunt stocate în microcip, iar modulul cititor RFID, precum RC522, primește aceste date prin intermediul bobinei NFC. Etichetele MIFARE 1K sunt compatibile cu modulul cititor RFID RC522. Pentru a stoca date individuale, aceste etichete au o memorie de 1 kilobyte. [17]

În particular, etichetele MIFARE 1K sunt compatibile cu modulul cititor RFID RC522 și oferă o capacitate de stocare de 1 kilobyte pentru date individuale. Prin conectarea modulului RC522 la un microcontroller, cum ar fi Arduino, se deschid posibilități extinse de dezvoltare a diverselor aplicații. Aceste aplicații pot include sisteme de acces cu cartelă, control al prezenței, identificarea produselor, sisteme de securitate și multe altele. [17]

Caracteristici RFID RC522: [17]

* Tensiune de alimentare: 3,3V.
* Curent inactiv: 10 - 13mA.
* Curent de sleep: 80uA.
* Curent maxim: 30mA.
* Frecvență de funcționare: 13,56MHz.
* Carduri suportate: S50, S70, UltraLight, Pro și Desfire.
* Versiune firmware: 0x12.
* Buffer FIFO (First In – First Out).
* Moduri flexibile de întrerupere.
* Temperatură optimă de funcționare: -25 °C - +85 °C.
* Protocoale de comunicare:
* RS232 Serial UART : până la 1228.8 kBd.
* SPI: până la 10MBit/s.
* I2C: până la 400 kBd în Fast Mode și până la 3400 kBd în High-Speed Mode



*Figura 11. Modulul RFID RC522 cu pinii aferenți. [18]*

Descrierea pinilor modulului RFID RC522: [17]

* **VCC** - Acesta este pinul care furnizează energie modulului. Intervalul de tensiune al modulului variază de la 2,5 la 3,3 volți. Îl putem conecta la pinul de ieșire de 3,3V al microcontrollerului (Arduino).
* **RST** - O intrare pentru resetarea și oprirea modulului este pinul de resetare. Dezactivarea este activată atunci când acest pin este la nivel redus. Acest lucru dezactivează toate sursele interne de curent, inclusiv oscilatorul, iar pinii de intrare ai modulului sunt deconectați exterior.
* **GND** - Acesta este pinul de masă al modulului. Trebuie conectat la pinul de masă al microcontrollerului (Arduino).
* **IRQ** - Acesta este un pin de întrerupere care informează microcontrollerul să trezească modulul atunci când o etichetă RFID intră în raza sa de acțiune. Acesta ajută modulul să treacă în modul de veghe, care economisește energie.
* **MISO/SCL/Tx** - Acest pin funcționează ca Master-In-Slave-Out atunci când este activată interfața SPI; atunci când este activată interfața I2C, funcționează ca un ceas serial; și atunci când este activată interfața UART, funcționează ca o ieșire de date seriale.
* **MOSI** – Master Out Slave In - pin pentru comunicare SPI.
* **SCK** - Acesta este pinul de ceas serial. Acceptă impulsurile de ceas furnizate de magistrala SPI, adică de microcontrolerul (Arduino).
* **SS/SDA/Rx** - Acest pin acționează ca intrare serială (SS) în timpul comunicării SPI, SDA în timpul I2C și RX în timpul UART.

### Modulul Bluetooth HC-05

HC-05 reprezintă un modul Bluetooth ce oferă capacități de funcționare atât ca Master, cât și ca Slave, permițând astfel comunicarea wireless între dispozitive prin intermediul tehnologiei Bluetooth. Acest modul, caracterizat prin dimensiuni compacte și ușurința în utilizare, este larg utilizat în proiecte electronice pentru a adăuga funcționalitatea de comunicare Bluetooth la dispozitivele existente. [19]

Modulul HC-05 este dotat cu un indicator LED roșu, care își asumă rolul de a indica starea de conectivitate Bluetooth. Înainte de a se realiza o conexiune cu modulul HC-05, LED-ul roșu clipește într-un mod constant și periodic. Odată ce conexiunea Bluetooth este stabilită cu succes între modulul HC-05 și un alt dispozitiv Bluetooth, ritmul de clipocire al LED-ului roșu se reduce la o frecvență de două secunde. [19]

Modulul HC-05 utilizează standardul Bluetooth 2.0+EDR (Enhanced Data Rate) pentru a permite comunicația wireless între dispozitive. Această caracteristică permite modulului HC-05 să se conecteze și să comunice cu alte dispozitive Bluetooth, precum telefoane mobile, tablete, computere sau alte module Bluetooth disponibile. Prin configurarea adecvată, modulul HC-05 poate asuma atât rolul de Master, când preia controlul în procesul de comunicare, cât și de Slave, când acceptă comenzi și se supune comenzilor primite de la dispozitivele Master. [19]

Modulul HC-05 se conectează la un microcontroller, în cazul nostru Arduino, prin intermediul interfeței seriale UART. Acesta permite transmiterea și recepționarea datelor între microcontroller și dispozitivul conectat prin Bluetooth. [19]

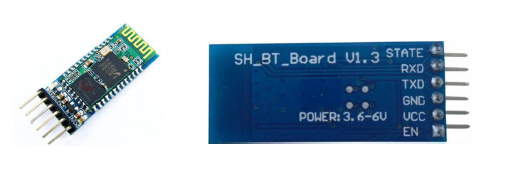
Modulul HC-05 prezintă o rază de acțiune nominală de aproximativ 10 metri în condiții ideale, cu mențiunea că acestă valoare poate varia în funcție de factori externi precum prezența obstacolelor fizice sau interferența radio. Distanța de comunicare poate fi influențată negativ de prezența obiectelor solide între dispozitive sau de surse de interferență radio, ceea ce poate reduce semnalul și, implicit, raza de acoperire a modulelor Bluetooth. În ceea ce privește transferul datelor, modulul HC-05 suportă o gamă de rate de transfer configurabile. Acestea includ valorile de 9600, 19200, 38400, 57600, 115200, 230400, 460800 și 921600 biți pe secundă (bps). Utilizatorii pot alege una dintre aceste rate de transfer pentru a se potrivi cu necesitățile lor specifice de comunicare și de viteză a transferului de date între modulele HC-05 și dispozitivele conectate. [19]

Modulul HC-05 poate fi utilizat într-o varietate de aplicații cum ar fi transmiterea și recepționarea datelor între dispozitive, controlul dispozitivelor prin intermediul unei aplicații mobile, comunicarea wireless între dispozitive IoT (Internet of Things), implementarea sistemelor de control la distanță și multe altele.

Prin intermediul modulului HC-05 este posibilă integrarea capacității de comunicație Bluetooth în proiectele electronice, oferind o modalitate convenabilă de a realiza comunicarea wireless între dispozitive.

Specificații ale modulului Bluetooth HC-05: [19]

* Versiunea Bluetooth: 2.0 + EDR.
* Frecvență: bandă ISM (Industrial, Scientific and Medical) de 2,4GHz.
* Modulație: GFSK (Gaussian Frequency Shift Keying).
* Puterea de transmisie: Clasa 2 (până la 4 dBm).
* Sensibilitate: -80 dBm tipic.
* Raza de acțiune: aproximativ 10 metri.
* Profiluri acceptate: SPP (Serial Port Profile), HID (Human Interface Device) și altele.
* Tensiune de funcționare: 3,3V până la 5V DC.
* Curent de funcționare: mai puțin de 50mA.
* Curent de așteptare: mai puțin de 2,5mA
* Curent de repaus: mai puțin de 1mA
* Interfață: UART
* Rate de transfer: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400 și 460800 bps.
* Temperatura de funcționare: -20°C până la 75°C.



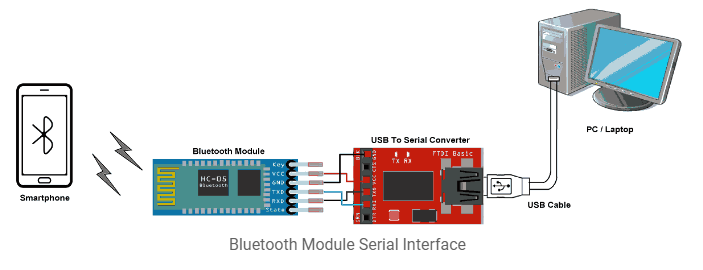
*Figura 12. Modulul Bluetooth HC-05 cu pinii aferenți. [19]*

Pentru realizarea comunicării dintre smartphone și modulul Bluetooth HC-05, este necesară utilizarea unei aplicații de terminal Bluetooth pe smartphone pentru a facilita transmiterea și primirea datelor. [19]

Pentru a stabili comunicarea între smartphone și modulul HC-05, modulul HC-05 trebuie să fie conectat la un computer prin intermediul unui convertor serial la USB.

Înainte de a iniția comunicarea între cele două dispozitive Bluetooth, este necesară asocierea modulului HC-05 cu smartphone-ul. Acest proces implică configurarea smartphone-ului pentru a recunoaște și a se conecta la modulul HC-05. După asociere, smartphone-ul și modulul HC-05 vor putea comunica între ele în mod eficient și se pot transmite și recepționa date prin intermediul conexiunii Bluetooth stabilită.

Pentru a realiza asocierea (împerecherea) între un dispozitiv mobil și modulul Bluetooth HC-05, utilizatorul poate iniția procesul prin căutarea dispozitivelor Bluetooth disponibile pe telefonul mobil. În urma căutării, utilizatorul va identifica un nou dispozitiv Bluetooth cu denumirea "HC-05", atunci când modulul Bluetooth HC-05 este alimentat corect și activ. După identificarea modulului HC-05 în lista dispozitivelor Bluetooth disponibile, utilizatorul poate selecta opțiunea "Connect/Pair" asociată dispozitivului HC-05. În general, codul de asociere predefinit pentru modulul HC-05 este “1234” sau “0000”, care trebuie introdus de utilizator în procesul de asociere.



*Figura 13. Stabilire comunicare între telefonul mobil și modulul Bluetooth HC-05. [19]*

Modulul Bluetooth se poate programa folosind diferite comenzi specifice, după cum se poate observa în exemplul următor:

Tabel1. Listă comenzi AT pentru modul Bluetooth HC-05. [19]

|  |  |  |
| --- | --- | --- |
| **Comandă** | **Răspuns comandă** | **Descriere comandă** |
| AT | OK | Verificare comunicație |
| AT+PSWD=XXXX | OK | Setare parolă |
| AT+NAME=XXXX | OK | Setare nume dispozitiv |
| AT+UART=BAUDRATE, bit  de stop, bit de paritate | OK | Schimbare baudrate |
| AT+VERSION? | +Versiune: XX-OK | Versiune Bluetooth |
| AT+ORGL | Parametri: tipul dispozitivului, parametri seriali, cheie de trecere,  etc. | Trimite detalii despre configurarea cu succes a producătorului |

Modulul Bluetooth HC-05 conține 6 pini distribuiți în felul următor (Figura 12.): [19]

* **Enable/Key** - este utilizat pentru a aduce modulul Bluetooth în modul de comenzi AT (Attention / AT command set). Dacă pinul Key/EN este alimentat la o tensiune, atunci acest modul va funcționa în modul de comandă. În caz contrar, în mod implicit, acesta se află în modul date. Viteza de transmisie implicită a HC-05 în modul de comandă este de 38400 bps și de 9600 bps în modul de date. Modulul Bluetooth HC-05 are două moduri:
* Mod de date cu rol în schimbul de date dintre dispozitive.
* Mod comandă ce folosește comenzi AT care sunt folosite pentru a schimba setările modulului Bluetooth HC-05. Pentru a trimite aceste comenzi ne folosim de portul serial (UART).
* **VCC** - alimentează modulul. Se conectează la +5V tensiune de alimentare.
* **GND** - Pinul de masă al modulului, conectat la masa sistemului.
* **TXD -- Transmitter** - transmite date seriale. Tot ceea ce se primește prin Bluetooth va fi transmis de acest pin ca date seriale.
* **RXD -- Receiver** - primirea datelor seriale. Fiecare dată serială transmisă la acest pin va fi transmisă prin Bluetooth.
* **State** - pinul de stare este conectat la LED-ul de la modul. Acesta poate fi utilizat ca feedback pentru a verifica dacă modulul Bluetooth funcționează corect.

Alegerea modulului HC-05 este motivată de dorința de a beneficia de funcționalitatea Bluetooth în proiectul meu, iar acest modul îndeplinește cerințele în ceea ce privește utilizarea ușoară și integrarea flexibilă în diferite proiecte. Prin intermediul comunicării Bluetooth, pot realiza transferul datelor într-un mod convenabil și eficient, asigurând o conexiune stabilă între dispozitivul fizic și aplicația mobilă asociată.

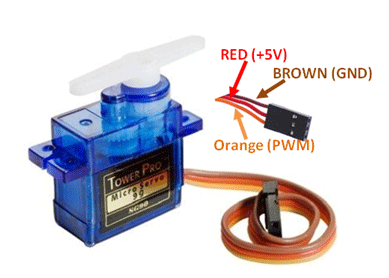
### Servomotor SG90 Micro Servo

Servomotorul reprezintă un dispozitiv electromecanic utilizat pentru a realiza controlul precis al poziției unui obiect, în baza semnalelor de control pe care le primește. În cadrul sistemului meu, servomotorul are rolul de a efectua mișcarea fizică a ușii în momentul în care se detectează o etichetă RFID validă, se apasă butonul de deschidere sau se primește o comandă vocală prin intermediul aplicației mobile. [20]

Un servomotor standard este compus dintr-un motor electric, un sistem de reducere a vitezei și un circuit de control integrat. O caracteristică cheie a servomotoarelor constă în capacitatea lor de a fi controlate cu precizie în ceea ce privește poziția sau unghiul de rotație.

Pentru a obține controlul asupra unui servomotor, se trimit semnale de control sub forma unor impulsuri de lățime variabilă, cunoscute sub denumirea de modulare în lățime a impulsului (PWM). Durata impulsului PWM determină poziția sau unghiul de rotație la care servomotorul se va poziționa. În mod obișnuit, semnalul PWM utilizat are o frecvență de aproximativ 50 Hz. [20]

În vederea asigurării compatibilității cu platforma Arduino și alte microcontrollere, majoritatea servomotoarelor sunt proiectate cu o interfață standard compusă din trei pini. Această interfață simplă permite conectarea facilă și comunicația cu microcontrollerul.



*Figura 14. SG90 Micro Servo 9g cu firele aferente. [21]*

Caracteristici și dimensiuni ale SG90 Micro Servo 9g: [21]

* Tensiune de operare: 4,8V - 6V.
* Viteza de operare: 0,12 sec/60° (fără sarcină).
* Cuplu de blocare: 1,8kg/cm (4,8V); 2,2kg/cm (6V).
* Lățime zonă neutră: 5 μs.
* Greutate: 9 grame.
* Dimensiuni: 23mm x 12.2mm x 29mm.
* Unghi de rotație: 180°.

Configurație fire SG90 Micro Servo 9g: [21]

* **Brown Wire**: reprezintă firul de împământare ce se conectează la împământarea sistemului.
* **Red Wire**: alimentarea motorului, de obicei utilizându-se tensiune de +5V.
* **Orange Wire**: semnalul PWM este dat în acest fir pentru a controla motorul.

### OLED Display I2C SSD1306

În proiectul meu, OLED (Organic Light Emitting Diode) Display I2C SSD1306 joacă un rol esențial oferind o interfață vizuală interactivă pentru utilizatori. Acesta afișează informații importante legate de procesul de deschidere a ușii, oferind feedback utilizatorului în timp real.

Prin intermediul afișajului OLED, utilizatorii primesc mesaje de acces sau respingere în funcție de validitatea tagului RFID detectat. Aceasta permite o comunicare eficientă și intuitivă cu sistemul de deschidere a ușii, oferind o experiență mai interactivă și informativă pentru utilizatori. De asemenea, ecranul OLED poate fi utilizat pentru a afișa alte informații relevante, precum stări ale sistemului sau alte mesaje personalizate.

Cu ajutorul interfeței I2C, integrarea și controlul ecranului OLED cu Arduino Uno și restul componentelor proiectului meu devine simplă și eficientă. Astfel, utilizatorii pot primi feedback vizual clar și concis pentru a înțelege starea și rezultatele procesului de deschidere a ușii în timp real.

Alegerea acestui ecran a constat în numărul redus de pini disponibili pe plăcuța de dezvoltare Arduino Uno utilizată.

SSD1306 este un driver OLED/PLED CMOS (Complementary Metal-Oxide-Semiconductor) cu un singur cip, conceput pentru a controla un sistem de afișare grafică dot-matrix de rezoluție 128×64 pixeli. Acesta utilizează tehnologia OLED/PLED, care se referă la diodele emițătoare de lumină organică/polimerică, pentru a genera imaginile pe ecran. SSD1306 este un circuit integrat care operează cu un afișaj OLED de tip catod comun, având o configurație specifică de 128 de segmente și 64 de comune. Cu alte cuvinte, acesta gestionează controlul și comunicația între cele 128 de segmente orizontale și cele 64 de comune verticale ale panoului OLED. [22]

Structura de bază a unui OLED constă într-un strat subțire de materiale organice poziționate între un catod și un anod. Anodul este de obicei realizat dintr-un oxid de indiu și staniu (ITO) transparent și conductiv, în timp ce catodul poate fi dintr-un metal cu potențial redus. Materialele organice sunt dispuse într-o peliculă subțire cu mai multe straturi, acestea incluzând stratul de transport al găurilor (HTL - Hole Transport Layer), stratul de emisie (EML - Emissive Layer) și stratul de transport al electronilor (ETL - Electron Transport Layer). Aceste straturi joacă rolul de a facilita transferul de încărcături (găuri și electroni) între anod și catod și de a permite formarea excitonilor în stratul de emisie. În momentul în care excitonii revin la stările de energie inferioare, emisia de lumină (electroluminescență) are loc. Calitatea componentelor OLED este influențată de alegerea adecvată a materialului de transport, a materialului stratului de emisie și a electrozilor utilizati. Acești factori determină eficiența energetică, stabilitatea, durabilitatea și reproducerea culorilor în cadrul OLED-urilor. [22]

SSD1306 integrează funcții precum controlul contrastului, memoria RAM pentru afișare și un oscilator, reducând astfel necesitatea unui număr mare de componente externe și consumul de energie. Controllerul permite ajustarea luminozității în 256 de trepte distincte. Datele și comenzile necesare pentru afișare sunt transmise de către unitatea de microcontrol (MCU - Microcontroller Unit) către SSD1306 prin intermediul unei interfețe paralele compatibile cu seria 6800/8000 a interfeței I2C sau interfeței seriale periferice, selectabile prin configurarea hardware. SSD1306 este potrivit pentru o varietate de aplicații portabile și compacte, cum ar fi sub-afișajul telefoanelor mobile, playere MP3, ceasuri inteligente, dispozitive de monitorizare a sănătății, calculatoare și altele. [22]



*Figura 15. OLED Display I2C SSD1306 cu pinii aferenți. [22]*

Caracteristici OLED Display I2C SSD1306: [22]

* Rezoluție: 128 x 64 pixeli.
* Dimensiune ecran: 0.96 inch.
* Unghi de vizualizare: >160°.
* Tensiune de intrare: 3,3V - 6V.
* Dimensiune modul: 27 x 27 x 3.5mm.

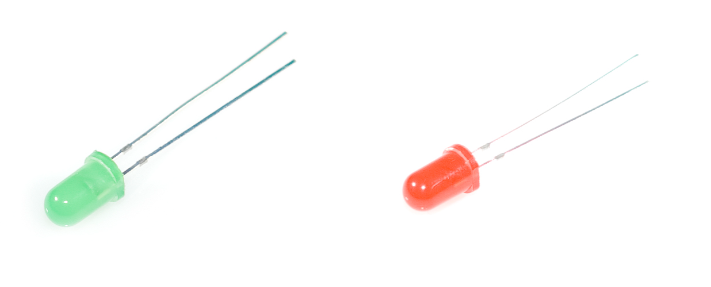
Caracteristici ale pinilor pentru OLED Display I2C SSD1306: [22]

* **GND** - pin pentru împământare.
* **VCC (VDD, 5V)** - un singur pin de alimentare pentru toate intrările de alimentare.
* **SCK (CLK, SCL)** - realizează un semnal de ceas comun pentru Arduino cu OLED.
* **SDA (MOSI, DI)** - primește datele de la dispozitivul de control (în cazul nostru fiind plăcuța de dezvoltare Arduino Uno).

### LED

Un LED (Light-Emitting Diode), în română diodă emițătoare de lumină, este un dispozitiv electronic semiconductor care convertește curentul electric în lumină. LED-urile sunt compuse dintr-un strat subțire de material semiconductor, care poate fi dopat pozitiv (tip p) și dopat negativ (tip n). Când un curent electric este aplicat la LED, electronii se recombină cu găurile electronice în zona de joncțiune dintre cele două tipuri de dopaj, eliberând energie sub formă de fotoni. Această energie emisă sub formă de lumină determină luminozitatea LED-ului. [23]

LED-urile folosite în proiectul meu, în număr de două (un LED de culoare verde și un LED de culoare roșu), au rol în a oferi feedback vizual corespunzător pentru fiecare din răspunsurile primite la citirea etichetei RFID. În cazul acționării servomotorului din aplicația mobilă, doar LED-ul de culoare verde este utilizat.



*Figura 16. LED-uri normale 5mm de culoarea verde, respectiv roșu. [27] [28]*

Caracteristici LED: [23]

* Dimensiune: 5mm.
* Tensiune nominală: de obicei, LED-urile de 5 mm au o tensiune nominală între 1,8V și 3,3V, în funcție de tipul de LED și culoarea emisă.
* Curent nominal: LED-urile de 5 mm au, de regulă, un curent nominal între 5 mA și 20 mA, în funcție de tipul de LED și luminozitatea dorită.
* Unghi de vizualizare: LED-urile de 5 mm au de obicei un unghi de vizualizare între 30° și 60°, care poate varia în funcție de tipul de LED și designul optic.

### Buzzer activ 5V

Un buzzer este un dispozitiv electronic utilizat pentru a genera sunete sau semnale acustice. Un electromagnet cu o membrană magnetică este plasat pe carcasă. La trecerea curentului electric prin magnet se formează un câmp magnetic alternativ, sub influența căruia membrana începe să oscileze, creând astfel un semnal sonor. [24]

Există două categorii principale de buzzere: active și pasive. Diferența principală între ele constă în modul în care generează sunetul. [24]

Buzzerele active sunt dispozitive autonome care generează sunet atunci când li se aplică o tensiune. Ele au un circuit intern care produce sunetul și nu necesită un semnal extern pentru a funcționa. Acestea sunt mai comune și mai ușor de utilizat în proiecte, deoarece pot fi activate direct prin alimentarea cu tensiune corespunzătoare. Buzzerele active oferă un nivel ridicat de control asupra sunetului produs deoarece pot fi programate pentru a genera diferite tonuri și modele de sunete. [24]

Buzzerele pasive sunt dispozitive care necesită un semnal extern pentru a produce sunet. Ele sunt, de fapt, tipuri speciale de difuzoare, care răspund la semnalele electrice aplicate pentru a genera sunet. Pentru a funcționa, un semnal electric trebuie să fie generat de un circuit extern și aplicat la buză. Buzzerele pasive sunt mai puțin utilizate în proiecte electronice de bază și pot necesita mai mult efort de configurare și control. [24]

Buzzerul ales pentru implementarea proiectului prezintă caracteristici active și funcționează la o tensiune nominală de 5V. Principalul său scop este acela de a furniza un feedback auditiv în momentul în care accesul este permis sau respins, aducând astfel o componentă acustică în procesul de interacțiune cu utilizatorul.



*Figura 17. Buzzer activ cu alimentare la 5V. [26]*

Caracteristici Buzzer activ 5V: [25]

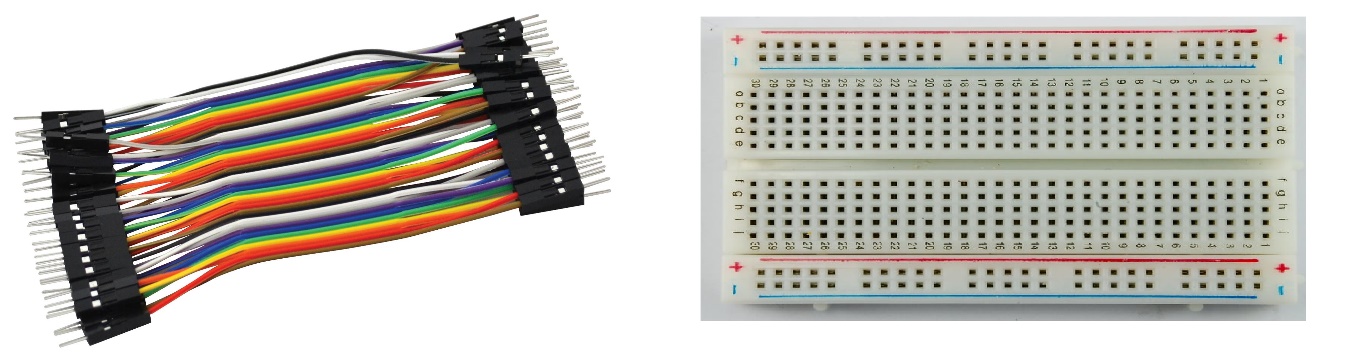
* Tensiune nominală: 6V DC.
* Tensiune de funcționare: de la 4V până la 8V DC.
* Curent nominal : ≤30mA.
* Ieșirea sunetului la 10 cm : ≥85dB.
* Frecvența rezonantă: 2300Hz ± 300Hz.
* Tonalitate : Continuă.
* Temperatura de funcționare: -25°C până la +80°C.

### Componente auxiliare

În cadrul proiectului de licență, atât breadboard-ul cât și firele jumper își dovedesc relevanța prin multiplele lor beneficii practice și funcționale, devenind astfel elemente indispensabile în realizarea și implementarea acestui sistem inteligent de deschidere a ușii.

**Breadboard**-ul reprezintă un instrument crucial în dezvoltarea prototipurilor de circuite electrice, oferind un cadru organizat și eficient pentru dispoziția componentelor. Cu ajutorul acestuia, am realizat conexiunile electrice necesare ce pot fi stabilite fără a necesita sudare, lucru ce facilitează modificarea și optimizarea circuitului în timpul procesului de dezvoltare. În cazul proiectului prezent, breadboard-ul a permis o explorare rapidă și flexibilă a diferitelor configurații de circuite, esențială în etapa de proiectare și testare a sistemului.

**Firele jumper**, pe de altă parte, constituie un alt element esențial, având rolul de a asigura interconexiunea componentelor dispuse pe breadboard. Întrucât acestea permit crearea unui traseu electric personalizat, ele își dovedesc utilitatea în multiple scenarii, indiferent de complexitatea circuitului sau de numărul de componente implicate. În contextul proiectului, firele jumper au fost utilizate pentru a lega diferitele module și componente hardware (modulul RFID, servomotorul, modulul Bluetooth, LED-urile, display-ul OLED și buzzerul) cu placa Arduino, asigurând astfel funcționarea corectă și eficientă a întregului sistem.



*Figura 18. Firele jumper utilizate și breadboard. [29] [30]*

Caracteristici Breadboard: [31]

* Benzi de distribuție: breadboard-ul are două dungi de distribuție, care sunt utilizate în mod obișnuit pentru a furniza conexiuni de alimentare sau masă la mai multe puncte pe placa de bază.
* Dimensiunea cablului: dimensiunea recomandată a cablului pentru acest breadboard este între 21 și 26 AWG (American Wire Gauge). Aceasta indică gama de grosimi a cablurilor care sunt potrivite pentru utilizare cu breadboard-ul.
* Puncte de legare: breadboard-ul are în total două sute de puncte de legare. Punctele de legare sunt acele puncte, sau găuri individuale, de pe breadboard în care pot fi inserate și conectate componente și cabluri.
* Tensiunea suportată: breadboard-ul poate suporta o tensiune de până la 1.000V AC. Această specificație indică tensiunea maximă pe care breadboard-ul o poate gestiona în siguranță fără a se produce defecțiuni sau deteriorări electrice.
* Puncte de legare pentru circuite integrate (IC): există 630 de puncte de legare special concepute pentru acomodarea circuitelor integrate (IC). Aceste puncte de legare sunt, de obicei, aranjate în grupuri și sunt conectate intern pentru a se potrivi cu configurația pinilor celor mai comune tipuri de circuite integrate.
* Rezistența la izolație: rezistența la izolație este de DC = 500V sau 500MΩ. Acest lucru indică rezistența electrică a breadboard-ului la o tensiune de 500V DC, măsurată în ohmi.
* Dimensiune: dimensiunea breadboard-ului este de 16,5 x 5,4 x 0,9 cm (înălțime, lățime, grosime).
* Putere nominală: Puterea nominală a breadboard-ului este de 5 A.

Caracteristici ale firelor Jumper: [32]

* Lungime: 240mm x 4, 200mm x 4, 150mm x 8 și 110mm x 49.
* Conectori: mufă-mufă (de la pin la pin).
* Culoare: 7 culori diferite (roșu, negru, portocaliu, verde, galben, maro, alb).
* Material: plastic.

# Arhitectura software

## Configurare Arduino

### Mediul de dezvoltare Arduino IDE

Deoarece putem utiliza o serie de biblioteci software gratuite care sunt disponibile în mediul IDE, precum și biblioteci open-source, care au fost produse de alți utilizatori pentru a ajuta la optimizarea proiectelor folosind placa de dezvoltare Arduino, am ales aplicația Arduino IDE (v2.1.0) în locul altor platforme, precum Atmel Studio sau AVR Studio.

Arduino IDE (Integrated Development Environment) este un mediu de dezvoltare integrat puternic și versatil, conceput special pentru programarea și dezvoltarea proiectelor utilizând plăcile de dezvoltare Arduino. Este un software open-source, disponibil gratuit, care oferă o interfață intuitivă și ușor de utilizat pentru programatori de toate nivelurile de experiență. Programul IDE este compatibil cu o varietate de sisteme de operare, inclusiv Windows, MacOS și Linux. C și C++ reprezintă limbajele de programare compatibile Arduino IDE. [13]

Mediul de dezvoltare integrat (IDE) Arduino este o aplicație cross-platform bazată pe Java. A luat naștere din mediile de dezvoltare pentru limbajele de programare Processing și Wiring. [13]

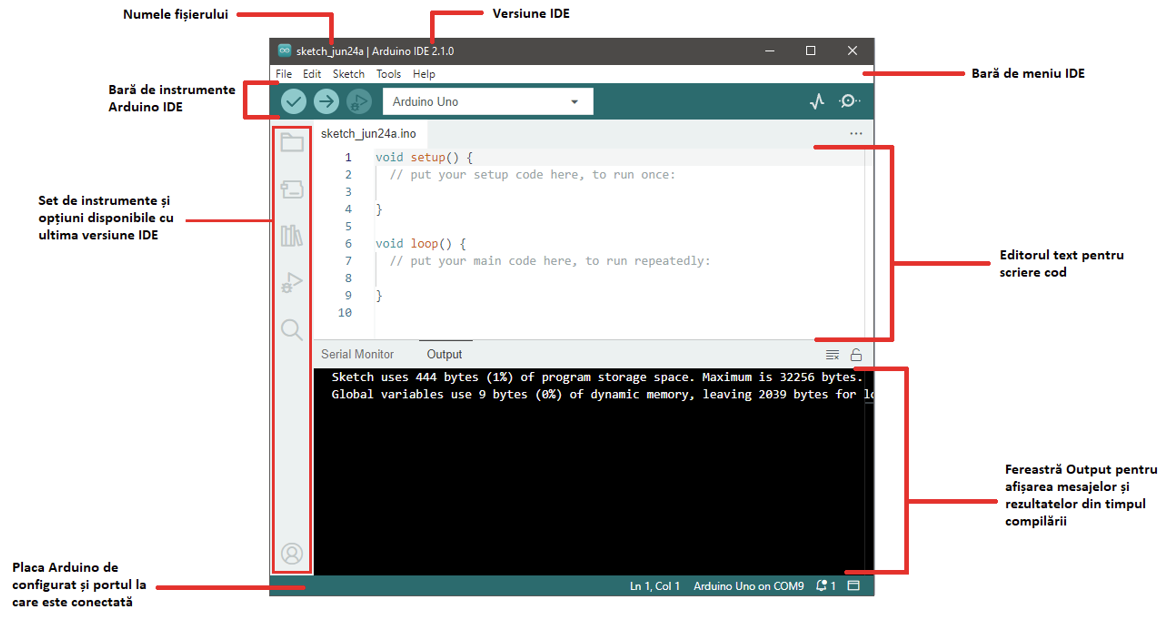
Cu Arduino IDE, putem crea și edita codul sursă al proiectelor într-un editor de text simplu și funcțional. Acesta vine cu o varietate de funcții utile, precum evidențierea sintaxei, completarea automată a codului și indentarea automată, care ne ajută să scriem și să organizăm codul într-un mod eficient. [13]

Încărcarea codului pe placa Arduino este un proces simplu în Arduino IDE. Cu doar câteva clickuri, putem compila și încărca codul pe placa conectată la calculator prin intermediul unui cablu USB. IDE-ul se ocupă de procesul de generare a codului binar și de încărcarea acestuia pe placa Arduino în mod automat. [13]

Un alt avantaj al Arduino IDE este biblioteca sa standard bogată, care include o varietate de funcționalități și module pregătite pentru utilizare. Această bibliotecă ne permite să accesăm funcționalități avansate ale plăcii Arduino, cum ar fi controlul pinilor, comunicarea serială, gestionarea senzorilor și multe altele, fără a fi nevoie să scriem codul de la zero.

IDE-ul include, de asemenea, un monitor serial integrat, care ne permite să comunicăm cu placa Arduino în timp real. Putem transmite și primi date prin portul serial, ceea ce este util pentru depanarea și monitorizarea proiectelor. [13]

Pe lângă aceste funcții esențiale, Arduino IDE oferă și multe alte facilități și opțiuni de configurare. De exemplu, putem selecta placa Arduino pe care o utilizăm, specificând tipul de microcontroller și setările corespunzătoare. De asemenea, putem adăuga și gestiona biblioteci externe pentru a extinde funcționalitățile și opțiunile disponibile în viitoarele proiecte. [13]



*Figura 19. Apariția mediului de dezvoltare Arduino IDE.*

### Programare microcontroller

Acest proiect folosește mai multe biblioteci pentru a facilita funcționarea diferitelor componente. Codul este scris în limbajul de programare C++, și este destinat să fie rulat pe un microcontroller Arduino Uno.

În continuare, voi descrie bibliotecile folosite, configurarea pinilor de intrare/ieșire și funcțiile cele mai importante ale codului utilizat pentru crearea acestui proiect.

Biblioteci folosite:

* **Biblioteca MFRC522:** este o librărie open-source pentru Arduino ce permite comunicarea cu modulul de citire a cardurilor RFID MFRC522.

Librăria MFRC522 include un set de funcții și metode ce facilitează comunicarea cu modulul RC522 prin interfața SPI și permite realizarea unor operații precum: inițializarea modulului, verificarea prezenței unui card RFID, citirea cardului RFID, oprirea criptării (după citirea cardului), determinarea tipului de card, etc.

Aceasta librărie este esențială pentru orice proiect ce implică modulul RC522, simplificând interacțiunea cu acesta și reducând semnificativ complexitatea codului scris. [33]

* **Biblioteca Adafruit\_GFX:** este o bibliotecă open-source pentru Arduino care oferă o colecție de funcții de bază pentru desenarea grafică pe display-uri. Aceasta este o librărie "core" folosită de multe alte biblioteci Adafruit pentru display-uri de diferite tipuri.

Funcțiile din Adafruit\_GFX includ capacitatea de a desena pixeli, linii, dreptunghiuri, cercuri și text pe ecran. De asemenea, permite scalarea și rotirea textului.

Biblioteca Adafruit\_GFX este adesea folosită împreună cu alte biblioteci Adafruit specializate pentru tipuri specifice de ecrane, precum Adafruit\_SSD1306 pentru ecranele OLED. În cazul meu, voi putea să comunic cu ecranul OLED conectat la Arduino prin intermediul acestei librării. [34]

* **Biblioteca Adafruit\_SSD1306:** este o bibliotecă software pentru Arduino care facilitează comunicarea cu modulele de afișare OLED bazate pe chipsetul SSD1306.

Biblioteca Adafruit\_SSD1306 furnizează o colecție de funcții care simplifică programarea acestor module de afișare OLED. De exemplu, aceasta include funcții pentru inițializarea ecranului, afișarea de text, desenarea de linii, cercuri, pătrate, și multe altele. [35]

* **Biblioteca SPI.h:** este o bibliotecă software pentru Arduino care facilitează comunicarea SPI între placa Arduino și alte dispozitive sau module hardware. SPI este un protocol de comunicare serială sincron, utilizat frecvent pentru a comunica cu diferite periferice, cum ar fi memoriile flash, senzorii, convertorii de analog la digital și multe altele.

Biblioteca SPI.h oferă un set de funcții care simplifică utilizarea protocolului SPI în proiectele Arduino. De exemplu, biblioteca include funcții pentru inițierea comunicației SPI (**SPI.begin()**), setarea modului SPI (**SPI.setDataMode()**), setarea vitezei de ceas SPI (**SPI.setClockDivider()**), și multe altele. [13]

* **Biblioteca Servo.h:** este o bibliotecă Arduino standard utilizată pentru controlul servomotoarelor.

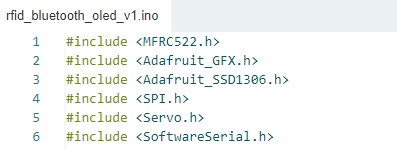
Biblioteca Servo.h permite controlul până la 12 servomotoare pe majoritatea plăcilor Arduino folosind doar funcții simple. Pentru a controla un servomotor, trebuie să creăm un obiect “Servo” și apoi să utilizăm funcțiile disponibile în bibliotecă. De exemplu, funcția **attach()** este folosită pentru a specifica pinul pe care este conectat servomotorul, iar funcția **write()** este folosită pentru a seta poziția servomotorului. [13]

* **Biblioteca SoftwareSerial.h:** este o bibliotecă Arduino care permite comunicația serială (RS232) prin utilizarea oricărui pin digital ca pin de transmisie sau recepție. Aceasta este extrem de utilă atunci când avem nevoie de mai multe conexiuni serial pe placă sau atunci când dorim să eliberăm porturile hardware serial pentru alte scopuri.

În proiectul meu, am utilizat biblioteca SoftwareSerial.h pentru a stabili o conexiune serială cu un modul Bluetooth prin intermediul pinilor digitali 0 și 1. Am creat un obiect SoftwareSerial numit “Blue” și l-am inițializat cu pinii TX și RX în funcția setup(). În funcția loop(), am folosit ”Blue.available()” pentru a verifica dacă există date pentru citire din modulul Bluetooth și ”Blue.read()” pentru a citi acele date. [13]

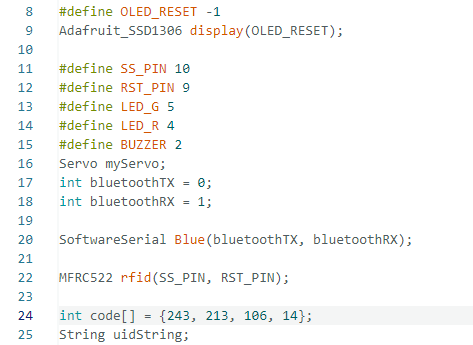
Dintre bibliotecile utilizate, doar trei dintre ele au nevoie de instalare separată pentru a funcționa în mediul de dezvoltare Arduino. Bibliotecile “Spi.h, Servo.h, SoftwareSerial.h” sunt biblioteci predefinite de IDE-ul Arduino.

Bibliotecile ”Adafruit\_GFX.h, Adafruit\_SSD1306.h, MFRC522.h” au fost descărcate și instalate direct din IDE-ul Arduino în felul următor: după instalarea și pornirea Arduino IDE, vom naviga către bara de meniu a IDE-ului (Figura 19.). Din bara de meniu vom selecta butonul “Sketch”, după care “Manage Libraries”, iar în fereastra ce se va deschide vom căuta numele bibliotecilor pe care dorim să le integrăm în proiectul nostru. După descărcare, vom inițializa librăriile ca în figura de mai jos.



*Figura 20. Librăriile folosite pentru realizarea proiectului.*

În continuare, după includerea bibliotecilor, vom defini pinii, variabilele globale și obiectele care vor fi folosite în restul programului ca în figura de mai jos.



*Figura 21. Definirea pinilor, obiectelor și variabilelor globale după inițializarea bibliotecilor corespunzătoare.*

**#define OLED\_RESET -1** : Definește pinul de reset pentru afișajul OLED ca fiind valoarea ”-1” pentru a indica faptul că nu este folosit niciun pin de reset.

**Adafruit\_SSD1306 display(OLED\_RESET);** : Creează o instanță a clasei “Adafruit\_SSD1306”, care controlează afișajul OLED. Acest obiect, numit “display”, este inițializat cu pinul de reset pentru a reseta ecranul într-un statut inițial.

**#define SS\_PIN 10** : Definește pinul SS (Slave Select) pentru cititorul RFID ca fiind pinul digital 10.

**#define RST\_PIN 9** : Definește pinul de reset pentru cititorul RFID ca fiind pinul digital 9.

**#define LED\_G 5**, **#define LED\_R 4** și **#define BUZZER 2** : Definesc pinii pentru LED-ul verde, LED-ul roșu și implicit buzzer.

**Servo myServo;** : Creează o instanță a clasei Servo, care va fi folosită pentru a controla servomotorul.

**int bluetoothTX = 0;** și **int bluetoothRX = 1;** : Definesc pinii TX și RX pentru modulul Bluetooth. Acestea sunt setate pe pinii 0 și 1, respectiv, care sunt pinii de comunicație serială pe plăcuța Arduino.

**SoftwareSerial Blue(bluetoothTX, bluetoothRX);** : Creează o instanță a clasei SoftwareSerial, care va fi folosită pentru comunicarea serială a plăcuței de dezvoltare Arduino Uno cu modulul Bluetooth HC-05. Conexiunea este setată astfel încât pinul de transmisie (TX) al Arduino Uno este conectat la pinul de recepție (RX) al modulului Bluetooth și invers. Acest lucru este specificat prin parametrii bluetoothTX și bluetoothRX, care au fost declarați anterior ca 0 și 1.

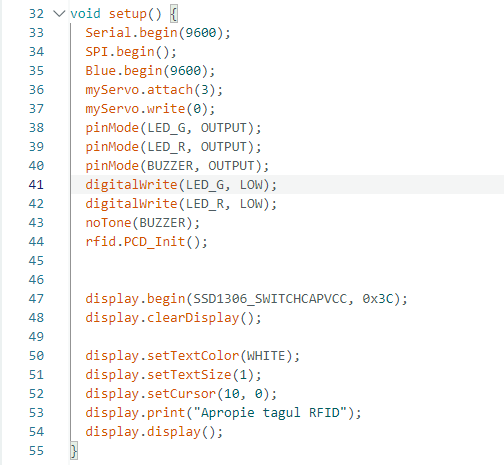
**MFRC522 rfid(SS\_PIN, RST\_PIN);** : Această linie creează o instanță a clasei ”MFRC522”, numită “rfid”. “MFRC522” este clasa care gestionează comunicarea cu modulul RFID. Această clasă necesită doi parametri pentru a funcționa corect, care sunt pinul SS (Slave Select) și pinul RST (Reset).

**int code[] = {243, 213, 106, 14};** : Definește un array cu codul UID al cardului RFID autorizat.

**String uidString;** : Definește un șir de caractere care va fi folosit pentru a stoca UID-ul cardului RFID citit.

În Arduino IDE, cele două funcții principale sunt **void setup()** și **void loop()**. Acestea sunt esențiale pentru orice program scris pentru placa Arduino și sunt chemate automat de către IDE.

1. **void setup():** Această funcție este apelată o singură dată la pornirea plăcii Arduino și este utilizată pentru inițializarea setărilor și configurațiilor inițiale. De asemenea, aceasta este echivalentă cu funcția **main()** din alte limbaje de programare.

****

*Figura 22. Configurare stare inițială a sistemului.*

**Serial.begin(9600); -** inițializează portul serial cu o viteză de transfer de date de 9600 biți pe secundă, pentru comunicarea cu computerul sau cu alte dispozitive.

**SPI.begin(); -** inițiază comunicația SPI necesară pentru comunicația cu modulul ”MFRC522 RFID”.

**Blue.begin(9600);** - inițiază comunicația serială software pentru modulul Bluetooth la o viteză de 9600 de biți pe secundă.

**myServo.attach(3);** și **myServo.write(0);** - atașează servomotorul la pinul digital 3 și îl rotește în poziția inițială (0 grade).

**pinMode(LED\_G, OUTPUT);**, **pinMode(LED\_R, OUTPUT);**, **pinMode(BUZZER, OUTPUT);** - Configurarea pinilor pentru LED-urile verde și roșu și pentru buzzer ca pini de ieșire.

**digitalWrite(LED\_G, LOW);**, **digitalWrite(LED\_R, LOW);**, **noTone(BUZZER);** - Aducerea LED-urilor și a buzzerului în stare 0 (închise).

**rfid.PCD\_Init();** - inițializarea modulului MFRC522.

*Inițializarea și configurarea ecranului OLED:*

**display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);** - Inițializarea ecranului OLED cu adresa I2C 0x3C.

**display.clearDisplay();** - Curăță afișajul OLED.

**display.setTextColor(WHITE);** - Setează culoarea ALB textului ce va fi afișat.

**display.setTextSize(1);** - Setează dimensiunea textului la 1.

**display.setCursor(10, 0);** - Plasează cursorul care este setat să înceapă de la 10 pixeli pe orizontală și 0 pixeli pe verticală.

**display.print("Apropie tagul RFID");** - Afișează mesajul "Apropie tagul RFID" pe ecranul OLED.

**display.display();** - Actualizează ecranul cu modificările făcute în linia de cod anterioară.

1. **void loop():** Această funcție în Arduino este partea principală a oricărui cod unde se execută logica principală a programului. După ce funcția setup() este apelată o dată la început, pentru a configura setările inițiale, funcția loop() este apelată într-o buclă infinită. Principalele funcții prezente în funcția **void loop()** sunt reprezentate în imaginea următoare.

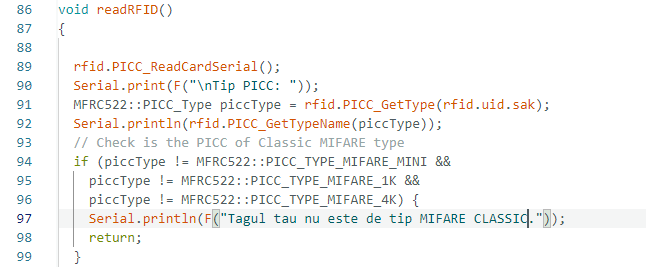


*Figura 23. Funcția void loop().*

**rfid.PICC\_IsNewCardPresent() -** se verifică dacă există un card RFID nou prezent. Dacă acesta există, se va apela funcția “readRFID()”.

**Blue.available() -** se verifică dacă există date disponibile de la conexiunea Bluetooth.

**Blue.read()** - dacă există date disponibile, se citesc aceste date utilizând această funcție. Dacă datele primite sunt 'A', se execută acțiunile corespunzătoare pentru accesul autorizat.



*Figura 24. Funcția readRFID() din interiorul funcției principale loop().*

**rfid.PICC\_ReadCardSerial()**: Această instrucțiune citeste datele cardului RFID.

**rfid.PICC\_GetType(rfid.uid.sak)**: Se obține tipul cardului RFID citit și se stochează în variabila piccType.

După acestea, se face verificarea tipului cardului RFID prin a se verifica dacă tipul cardului nu este MIFARE Classic Mini, MIFARE Classic 1K sau MIFARE Classic 4K. Dacă este un alt tip de card, se afișează un mesaj de eroare și funcția se oprește prin utilizarea instrucțiunii “return;”.



*Figura 25. Funcții pentru citirea tagului modulului RFID RC522.*

**printDec(rfid.uid.uidByte, rfid.uid.size);** : Se afișează ID-ul cardului RFID (reprezentat ca un șir de octeți) în format decimal.

**uidString = String(rfid.uid.uidByte[0]) + " " + String(rfid.uid.uidByte[1]) + " " + String(rfid.uid.uidByte[2]) + " " + String(rfid.uid.uidByte[3]);** : Această linie de cod construiește un șir de caractere uidString care reprezintă ID-ul cardului RFID. ID-ul este obținut din obiectul rfid.uid.uidByte, care conține un șir de octeți reprezentând fiecare byte al ID-ului.

Fiecare octet al ID-ului cardului RFID este convertit într-un șir de caractere folosind constructorul String(byte) și apoi este concatenat cu șirul " ". Astfel, rezultă un șir de caractere format din octetul 0, urmat de un spațiu, apoi octetul 1, alt spațiu, octetul 2 și, în final, octetul 3.

**rfid.PICC\_HaltA()**: Se oprește comunicarea cu cardul RFID.

**rfid.PCD\_StopCrypto1()**: Se oprește criptarea cu cardul RFID.

### Comenzi AT proprii

Pentru a configura comenzile personalizate ale modulului Bluetooth, am efectuat următorii pași. În primul rând, am conectat modulul Bluetooth în modul normal, alimentat la tensiunea de 5V a plăcii, și am adăugat un fir suplimentar de la pinul "Enable" al modulului Bluetooth la tensiunea de 3,3V a plăcii Arduino Uno. Ulterior, am deconectat plăcuța de la sursa de current, am ținut apăsat butonul modulului Bluetooth și în același timp am conectat din nou placa la sursa de alimentare (PC). Apoi, am utilizat mediul de dezvoltare Arduino IDE pentru a deschide monitorul serial prin combinația de taste "CTRL + SHIFT + M". Am modificat viteza de transmitere a datelor (BaudRate) la 38400 și am schimbat opțiunea "New line" în "Both NR & CL" din meniul asociat BaudRate-ului. Pentru a verifica dacă modulul Bluetooth este în modul de comandă, am introdus comanda "AT" în secțiunea de text a monitorului serial. Dacă modulul se afla în modul de comandă, monitorul serial afișa mesajul "OK". Ulterior, am utilizat următoarele comenzi în secțiunea de text a monitorului serial pentru a schimba numele modulului Bluetooth și parola.

* AT+NAME=Licenta
* AT+PSWD=”2546”

Pentru a confirma aplicarea modificărilor, am utilizat următoarele două comenzi. Acestea pot fi folosite și înainte de comenzile menționate anterior pentru a afla numele actual al modulului și parola configurate.

* AT+NAME?
* AT+PSWD?

După efectuarea acestor operațiuni, deconectăm și reconectăm placa pentru a vedea dacă modificările făcute au fost salvate.



*Figura 26. Verificarea modificărilor aduse modulului Bluetooth.*

## Aplicația Mobilă

### Mediul de dezvoltare Mit App Inventor

MIT App Inventor este un mediu de programare vizual care permite tuturor utilizatorilor să creeze aplicații complet funcționale pentru telefoane Android, iPhone și tablete Android/iOS. A fost dezvoltat de către Massachusetts Institute of Technology (MIT) și este disponibil gratuit pentru utilizare.

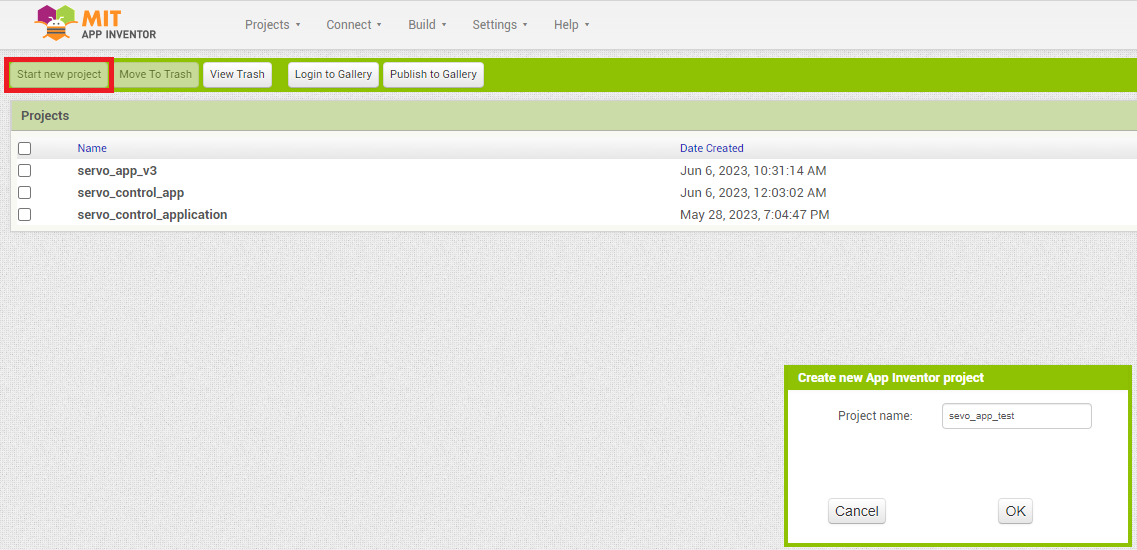
MIT App Inventor oferă o experiență de dezvoltare prietenoasă, care permite utilizatorilor să experimenteze logica programării și să vadă rezultatele în timp real. Emulatorul integrat ne permite să rulăm și să testăm aplicațiile pe un dispozitiv virtual, iar apoi să le instalăm și să le utilizăm pe propriul dispozitiv Android sau iOS. Acest instrument facilitează crearea de aplicații complexe, de mare impact, într-un timp semnificativ mai scurt decât mediile de programare tradiționale. [36]



*Figura 27. Mediul de programare vizual MIT App Inventor. [37]*

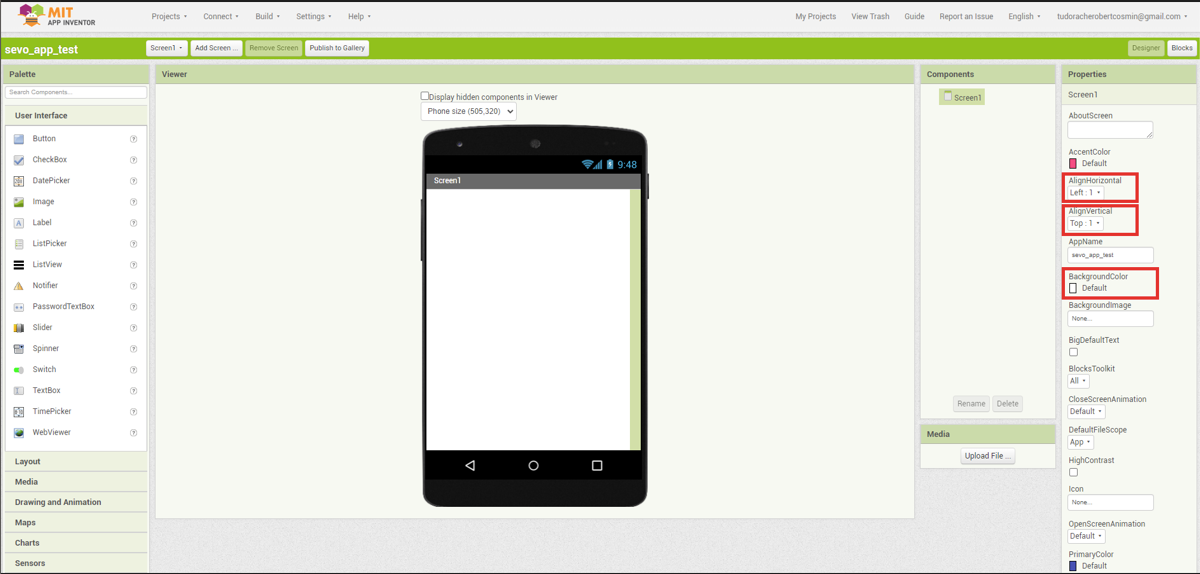
### Realizarea aplicatiei prin Mit App Inventor

Pentru a dezvolta aplicația mobilă utilizând mediul de programare vizual MIT App Inventor, am accesat site-ul oficial al platformei și ne-am autentificat folosind una dintre opțiunile disponibile pentru autentificare. După autentificare, am creat un nou proiect și i-am atribuit un nume relevant, așa cum este ilustrat în imaginea de mai jos.



*Figura 28. Crearea unui proiect nou utilizând MIT App Inventor.*

După crearea unui proiect nou, ni se va afișa interfața de design următoare unde vom crea aspectul vizual al viitoarei noastre aplicații mobile.



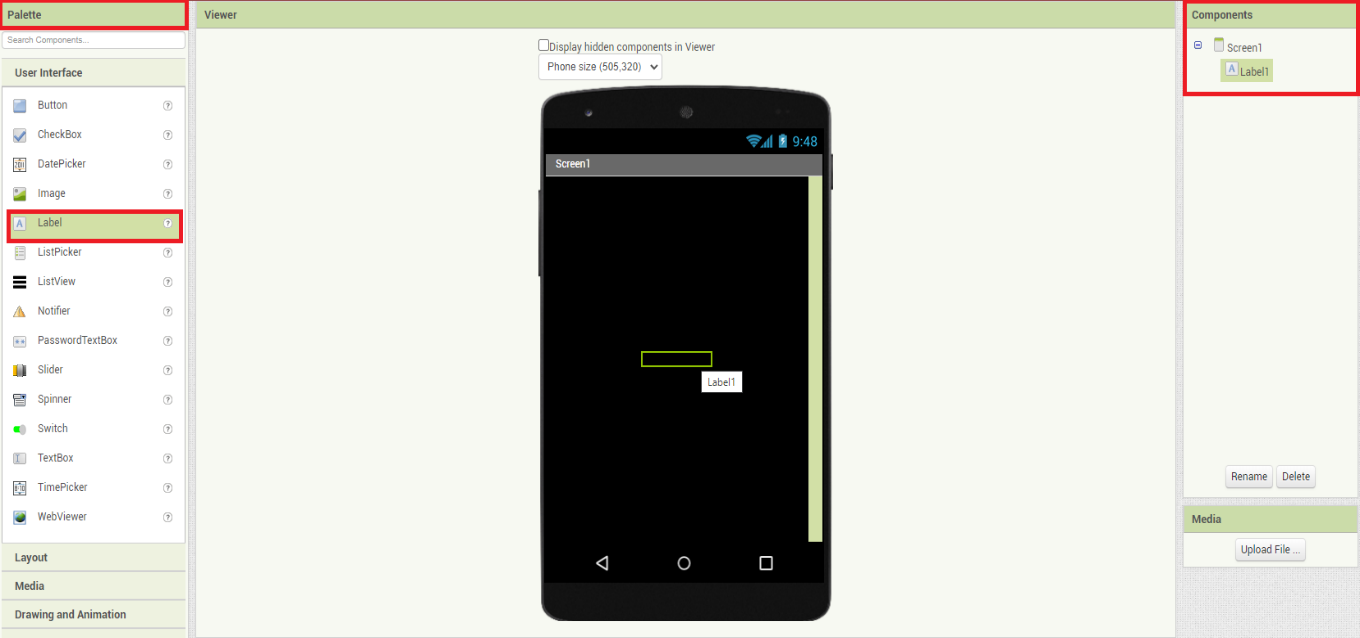
*Figura 29. Interfața Designer a mediului de programare vizual MIT App Inventor*

După cum am evidențiat în imaginea de mai sus, vom modifica alinierea orizontală și verticală a viitoarelor butoane și etichete ce vor fi afișate pe ecranul aplicației. Acestea sunt setate implicit și le vom modifica în felul următor:

* AlignHorizontal - Center : 3 (Left : 1 este setarea inițială).
* AlignVertical - Center : 2 (Top : 1 fiind setarea inițială).
* BackgroundColor - Black (alb fiind setarea inițială).

După modificarea ecranului, ar trebui să obținem fundalul de culoare negru și aliniererea care nu se poate observa până nu introducem componentele necesare.

Din meniul ”Pallete”, selectăm sub-meniul “User Interface” și facem vom trage obiectul ”Label” pe “Screen1” obținând rezultate ca în imaginea de mai jos.



*Figura 30. Instanțiere “Label”.*

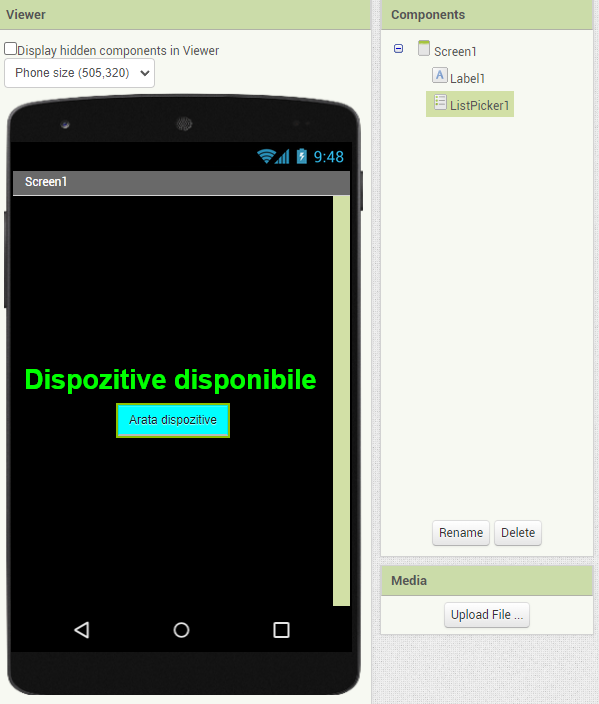
Observăm faptul că obiectul adus pe Screen1 se poate vedea, selecta și modifica pe parcursul dezvoltării aplicației din meniul “Components” afișat în partea stânga. După aducerea etichetei în Screen1, selectăm “Label1” din meniul “Components” și în meniul “Properties” vom face următoarele setări:

* FontBold - vom bifa această opțiune pentru a ne fi afișat textul îngroșat.
* FontSize - 30 (14 mărime inițială a textului).
* Text - vom introduce textul “Dispozitive disponibile”.
* TextColor - Green (culoare inițială fiind negru).

În continuare vom adăuga, din același meniu “Pallete” și sub-meniu “User Interface”, butonul “ListPicker” mai jos de “Label1”. Acesta este un buton care odată apăsat ne va arăta o listă de texte din care putem alege. Acest buton joacă un rol important în selecția dispozitivelor Bluetooth disponibile. Acesta va fi modificat astfel:

* BackgroundColor - Cyan (culoarea inițială fiind negru).
* ItemBackgroundColor - None (negru fiind inițial).
* Text - vom introduce textul “Arata dispozitive”.

După aceste implementări și setări, ar trebui să obținem rezultate ca în imaginea de mai jos.



*Figura 31. Rezultatul configurării.*

În continuarea dezvoltării aplicației, vom adăuga 3 butoane sub “ListPicker1”, unul sub altul, acestea găsindu-se în același meniu și sub-meniu menționate anterior.

Acestea vor fi modificate în felul următor:

* FontBold - bifat (nebifat inițial) pentru toate butoanele.
* BackgroundColor - Orange pentru primele două butoane, Yellow pentru al treilea buton.
* Text - pentru primul buton “Deschide”, al doilea buton “Vorbeste”, al treilea buton “Deconectare”.

Din același sub-meniu vom adăuga și componenta “Notifier”. Această componentă non-vizibilă are rolul de a afișa mesaje, mesaje temporare de alertă, instrucțiuni, etc.

Mai departe vom naviga in sub-meniul “Media” din meniul “Palette” și vom trage către ecran componenta “SpeechRecognizer”. Această componentă este non-vizibilă și o vom folosi pentru conversia vocii în text.

Din sub-meniul “Sensors” vom adduce în proiect componenta “Clock”. Această componentă are rolul de a furniza un mecanism de sincronizare pentru acțiunile comenzilor aplicației mobile.

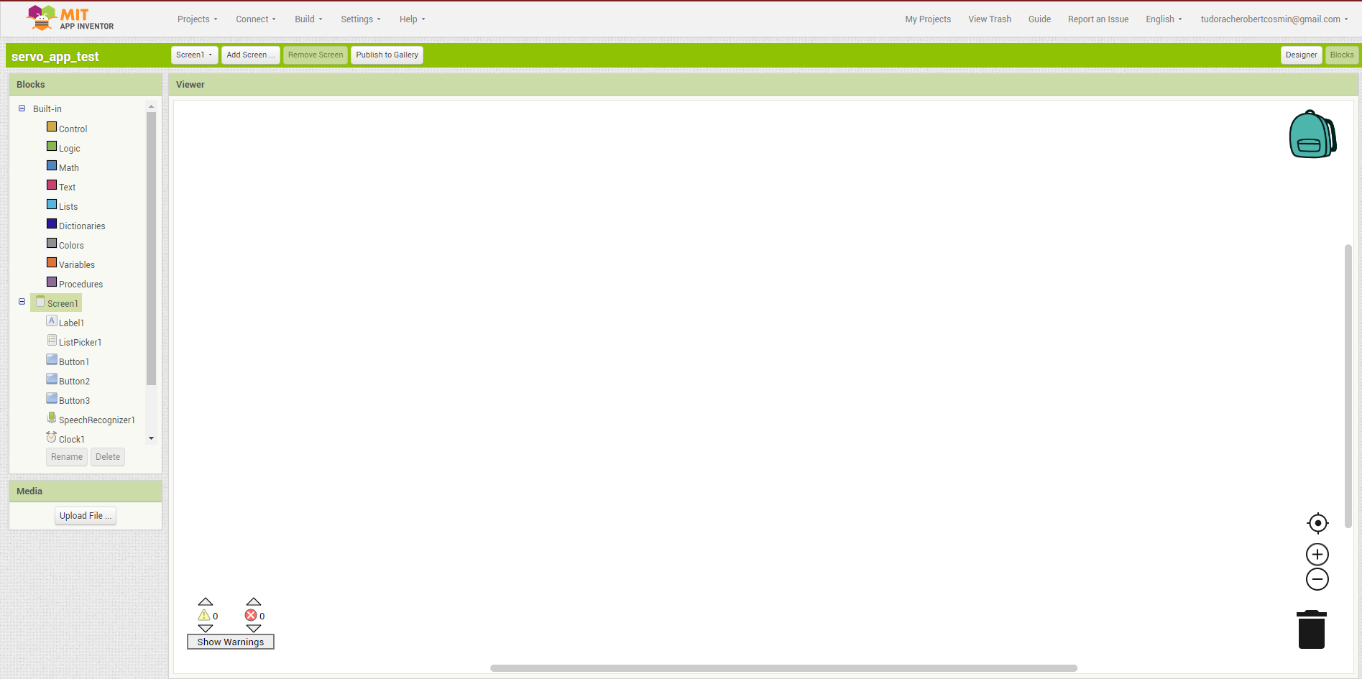
În ultimă fază, pentru designul aplicației mobile, vom naviga către sub-meniul “Connectivity” și adăuga componenta “BluetoothClient”, de asemenea non-vizibilă aplicației.

Urmând acești pași, interfața grafică ar trebui să arate ca în figura de mai jos.



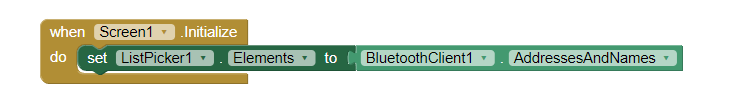
*Figura 32. Interfața grafică finală a aplicației.*

După realizarea interfeței grafice, mai departe vom realiza funcționalitatea aplicației. În partea din dreapta sus, sub contul nostru cu email-ul înscris pe site, observăm un buton intitulat “Blocks”. Navigând pe el, site-ul ne va pune la dispoziție o nouă interfață, aceasta fiind locul în care are vom realiza funcționalitățile aplicației folosind blocuri.



*Figura 33. Interfață blocuri MIT APP INVENTOR.*

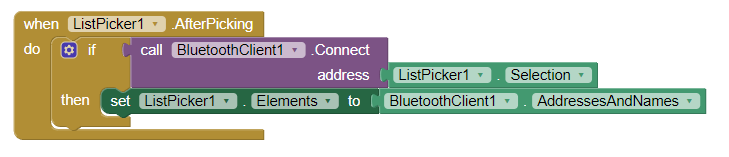
Vom realiza funcționalitatea în felul următor:



*Figura 34. Inițializare ecran.*

Această secvență de blocuri reprezintă un eveniment în Mit App Inventor în care componenta "Screen1" se inițializează. Aceasta declanșează codul asociat atunci când ecranul este încărcat sau creat. Codul din interiorul blocului de evenimente setează elementele componente "ListPicker1" la lista de adrese și nume ale dispozitivelor Bluetooth obținute de la componenta "BluetoothClient1".

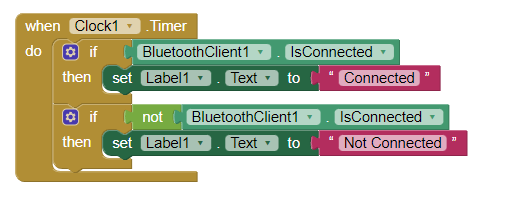
Pentru a crea acest eveniment, vom selecta ”Screen1” din meniul din stânga “Blocks” și vom aduce blocul ”When Screen1.Initialize do” din imaginea de mai sus pe ecran. În continuare vom selecta “ListPicker1” din meniul “Blocks” și vom aduce pe ecran blocul “set ListPicker1.Elements to”. Din același meniu de blocuri, vom merge pe componenta “BluetoothClient1” de unde selectam blocul “BluetoothClient1.AddressesAndNames”. Toate aceste blocuri le vom conecta ca în imaginea de ansamblu prezentată mai sus.



*Figura 35. Selectare dispozitiv Bluetooth.*

Această secvență de blocuri reprezintă un eveniment care are loc după ce utilizatorul selectează o opțiune din “ListPicker1”. În interiorul acestui eveniment, se verifică dacă apelul către ”BluetoothClient1.ConnectAddress” cu selecția din ”ListPicker1” este ADEVĂRAT. Dacă este adevărat, atunci se actualizează elementele componente „ListPicker1” cu lista de adrese și nume de la ”BluetoothClient1.AddressesAndNames”.

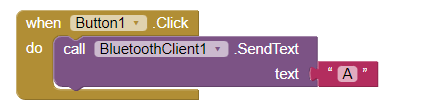
Pentru a crea acest eveniment, vom selecta ”ListPicker1” din meniul din stânga “Blocks” și vom aduce blocul ”When ListPicker1.AfterPicking do” din imaginea de mai sus pe ecran. Din meniul ”Blocks”, vom merge pe control și vom aduce blocul “IF/THEN”. În continuare vom selecta “BluetoothClient1” din meniul “Blocks” și vom aduce pe ecran blocul “call BluetoothClient1.Connectaddress”. Din același meniu de blocuri, vom merge pe componenta “ListPicker1” de unde selectam blocul “ListPicker1.Selection”. Pentru partea “THEN” a acestui eveniment pur și simplu vom copia partea de “DO” a evenimentului anterior exemplificat în Figura 34. Toate aceste blocuri le vom conecta ca în imaginea de ansamblu prezentată mai sus.



*Figura 36. Actualizare text Label1.*

Această secvență de blocuri reprezintă un eveniment care se activează la fiecare impuls al cronometrului Clock1. În interiorul acestui eveniment, se verifică dacă “BluetoothClient1” este conectat prin intermediul blocului ”BluetoothClient1.IsConnected”. Dacă este adevărat, atunci se actualizează textul etichetei ”Label1” cu valoarea "Connected".

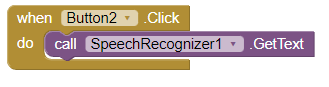
Pentru crearea acestui eveniment, vom selecta ”Clock1” din meniul din stânga “Blocks” și vom aduce blocul ”When Clock1.Timer do” din imaginea de mai sus pe ecran. Din meniul ”Blocks”, vom merge pe ”Control” și vom aduce 2 blocuri “IF/THEN”. În continuare vom selecta “BluetoothClient1” din meniul “Blocks” și vom aduce pe ecran blocul “BluetoothClient1.IsConnected” pe care-l vom conecta la un bloc “IF/THEN” pe partea “IF”. Din același meniu de blocuri, vom merge pe componenta “Label1” de unde selectam blocul “set Label1.Text to” și-l vom aduce în proiect. Acesta îl vom seta pentru partea de “THEN” a funcției și vom atașa din meniul de blocuri “Text” prima componentă în care vom scrie textul “Connected”. Pentru al doilea bloc “IF/THEN” vom copia primul bloc “IF/THEN” și vom adăuga și modifca următoarele: înaintea secvenței “IF” vom adăuga un “not”, bloc pe care-l găsim în secțiunea “Logic” al meniului “Blocks”, iar pentru partea de “THEN” vom modifica textul “Connected” la “Not connected”.



*Figura 37. Trimitere text către plăcuța Arduino utilizând Bluetooth.*

Când este apăsat butonul "Button1", folosește blocul "call BluetoothClient1.SendText" pentru a trimite textul "A" prin “BluetoothClient1”. Acest bloc de acțiune va trimite textul "A" către plăcuța Arduino utilizând Bluetooth. Dacă acest text este recepționat, va fi comparat cu caracterul “A” din codul sursă Arduino și va începe să execute instrucțiunile din interiorul acelui bloc (Figura 23.).

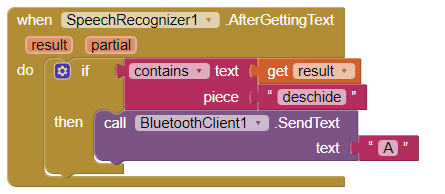
Pentru realizarea acestui eveniment, vom aduce din secțiunea “Button1”, al meniul “Blocks”, componenta ”when Button1.Click do”. Din același meniu vom selecta secțiunea ”BluetoothClient1” și vom conecta la funcția butonului componenta ”call BluetoothClient1.SendText text”. Din secțiunea ”Text” vom aduce prima componentă conectând-o la blocul ”BluetoothClient1.SendText text” redenumind-o cu caracterul ”A”.



*Figura 38. Obținerea textului utilizând recunoașterea vocală.*

Când este apăsat butonul "Button2", utilizează blocul "call SpeechRecognizer1.GetText" pentru a obține textul recunoscut prin intermediul “SpeechRecognizer1”. Acest bloc de acțiune va porni recunoașterea vocală și va obține textul recunoscut din vorbirea utilizatorului.

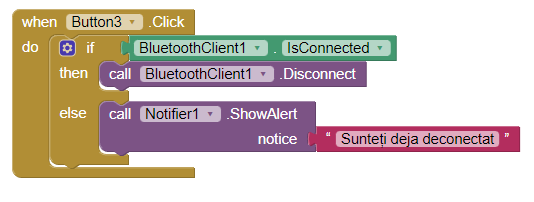
Pentru realizarea acestei funcții, vom aduce din secțiunea “Button2” blocul “when Button2.Click do” și vom conecta la acesta blocul “call SpeechRecognizer1.GetText” pe care-l găsim în secțiunea “SpeechRecognizer1”.



*Figura 39. Verificarea textului obținut prin recunoaștere vocală.*

Blocul “SpeechRecognizer1.AfterGettingText” folosește blocul ”IF” pentru a verifica dacă textul rezultat din recunoașterea vocală conține expresia "deschide". Dacă este adevărat, atunci apelează blocul "call BluetoothClient1.SendText" pentru a trimite textul "A" prin BluetoothClient1. Această acțiune va asigura că atunci când recunoașterea vocală returnează un rezultat complet, verificăm dacă textul conține expresia specifică și, în caz afirmativ, trimitem textul "A" prin ”BluetoothClient1”.

Pentru realizarea acestei funcții, vom naviga în secțiunea ”SpeechRecognizer1” din meniul ”Blocks” și vom aduce în zona de lucru blocul ”when SpeechRecognizer1.AfterGettingText do”. Din secțiunea ”Control” vom aduce un bloc ”IF/THEN” pe care-l vom atașa de blocul menționat anterior. Pentru partea de ”IF” vom atașa ”contains text piece” pe care-l găsim în secțiunea ”Text”. Vom atașa pentru partea de ”text”, a acesui bloc, blocul ”get result” pe care-l obținem prin aducerea cursorului pe butonul ”result” al blocului ”when SpeechRecognizer1.AfterGettingText do” și tragerea acestuia în interiorul zonei de lucru. Pentru partea de “piece”, vom aduce din secțiunea “Text” primul bloc, redenumindu-l “deschide”, cuvânt în baza căruia să se poată face verificarea pentru a acționa, în cazul nostru, servomotorul. Pentru partea ”THEN”, vom copia apelul către Bluetooth pentru trimiterea caracterului “A”, apel făcut în Figura 37.



*Figura 40. Implementare buton de deconectare de la modulul Bluetooth.*

Când se face clic pe “Button3” se utilizează blocul IF pentru a verifica dacă BluetoothClient1 este conectat. Dacă este adevărat, atunci apelează blocul "call BluetoothClient1.Disconnect" pentru a se face deconectarea. În caz contrar, apelează blocul "call Notifier1.ShowAlert" pentru a afișa o notificare cu mesajul "Sunteți deja deconectat". Această acțiune va permite verificarea stării de conectare a BluetoothClient1 și afișarea unui mesaj corespunzător, fie că este vorba despre deconectare sau notificarea că deja sunteți deconectat.

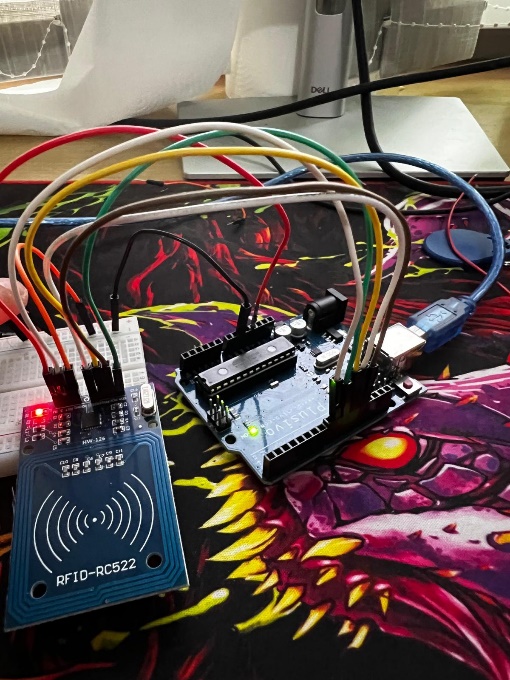
Pentru realizarea funcționalității acestui buton, vom naviga în secțiunea ”Button3” unde vom aduce în zona de lucru blocul ”when Button3.Click do”. Vom atașa la acesta un bloc ”IF/THEN/ELSE” care se găsește în secțiunea ”Control” al meniului ”Blocks”. Pentru partea de ”IF” vom atașa ”BluetoothClient1.IsConnected”, bloc pe care-l putem copia din funcționalitatea actualizării ”Label1” (Figura 36.). Pentru partea ”THEN” vom aduce blocul ”call BluetoothClient1.Disconnect”, bloc disponibil în secțiunea ”BluetoothClient1”, iar pentru partea de ”ELSE” vom atașa blocul ”call Notifier1.ShowAlert notice” pentru care vom atașa și un bloc text, din secțiunea ”Text”, modificându-l cu mesajul ”Sunteti deja deconectat”.

Pentru partea de deploy a aplicației, din bara de instrumente MIT APP INVENTOR vom selecta opțiunea “Build”. În continuare vom selecta opțiunea “Android App (.apk)”, această opțiune făcând referire la fișierul de instalare al unei aplicații dezvoltate pentru sistemul de operare Android. Acesta este un fișier cu extensia ".apk" (Android Package), care conține toate resursele, codul sursă și fișierele necesare pentru a instala și rula aplicația pe dispozitive Android. Selectând această opțiune, vom descărca pe computerul personal fișierul .apk al aplicației careia îi vom da deploy prin intermediul conectării telefonului la computer printr-un cablu USB. După introducerea fișierului într-un folder al telefonului, vom accesa telefonul pentru a căuta fișierul și instala aplicația. O metodă mai simplă de deploy a aplicației ar fi descărcarea din “Magazin Play” a aplicației MIT APP INVENTOR și scanarea codului QR care se generează automat după compilarea fișierului .apk. Scanarea acestui cod QR din aplicație ne va trimite către un portal pentru a descărca fișierul .apk.

# Implementare și testare

Pentru partea de implementare și testare a aplicației, voi reface conexiunile componentelor și voi prezenta motivul conectării în acest fel.

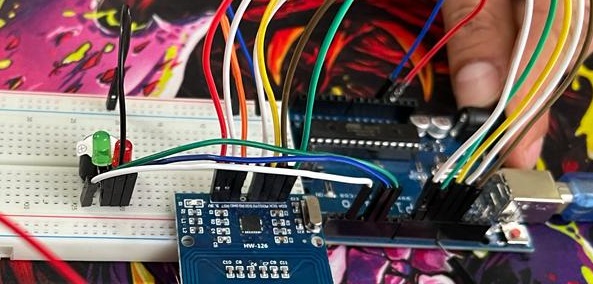
După ce avem componentele necesare, menționate în capitolul 2, vom începe cu configurarea modulului RFID.



*Figura 41. Cablare modul RFID la Arduino.*

Imaginea de mai sus prezintă modul în care am atașat pe breadboard și legat modulul RFID la placa Arduino. Conexiunile sunt prezentate în felul următor:

* Pinul 3,3V al modulului RFID este conectat la pinul 3,3V al plăcuței Arduino - asigură furnizarea adecvată a tensiunii necesare modulului RFID.
* Pinul RST al modulului RFID este conectat la pinul 9 al plăcuței Arduino - se permite controlul resetării modulului RFID direct de către programul sau codul implementat pe plăcuța Arduino. Acest lucru oferă flexibilitate în gestionarea și controlul modulului RFID în cadrul sistemului de deschidere a ușii.
* Pinul GND al plăcuței Arduino a fost introdus în șina de masă a breadboardului pentru a furniza o conexiune comună la masă pentru toate componentele sistemului.
* Pinul GND al modulului RFID a fost conectat la un pin al șinei de masă - realizează o legătură electrică directă între cele două componente, astfel încât potențialul de referință al modulului RFID să fie aliniat cu potențialul de referință al plăcuței Arduino. Aceasta asigură că tensiunile și semnalele transmise între modulul RFID și plăcuța Arduino sunt interpretate în mod corect și coerent.
* Pinul MOSI al modulului RFID a fost conectat la pinul 12 al plăcuței Arduino - pinul MOSI este utilizat de plăcuța Arduino în calitate de master pentru a trimite date către modulul RFID, care acționează în calitate de slave. Prin conectarea pinului MOSI al modulului RFID la pinul 12 al plăcuței Arduino, se stabilește o conexiune directă între aceste două componente, prin care plăcuța Arduino poate transmite date către modulul RFID.
* Pinul MISO al modulului RFID a fost conectat la pinul 11 al plăcuței Arduino - pinul MISO este folosit de modulul RFID în calitate de slave pentru a transmite datele către plăcuța Arduino, care acționează în calitate de master. Prin conectarea pinului MISO al modulului RFID la pinul 11 al plăcuței Arduino, se stabilește o legătură directă între cele două componente, prin care modulul RFID poate trimite date către plăcuța Arduino.
* Pinul SCK al modului RFID a fost conectat la pinul 13 al plăcuței Arduino - se realizează o legătură directă între cele două componente pentru transmiterea semnalului de ceas. Plăcuța Arduino generează semnalul de ceas pe pinul 13 și modulul RFID îl utilizează pentru a sincroniza transmiterea și recepționarea datelor.
* Pinul SDA al modulului RFID a fost conectat la pinul 10 al plăcuței Arduino - se stabilește o conexiune directă între cele două componente pentru a permite transferul de date serial de la modulul RFID către plăcuța Arduino.



*Figura 42. Cablare LED-uri și Buzzer.*

Figura 42. prezintă modul în care am amplasat LED-urile și buzzerul pe breadboard, și conectat aceste componente la placa Arduino. Pentru a asigura un mod mai optim de cablare, am așezat pe orizontal fiecare pin de GND al acestor componente. În continuare, avem nevoie doar de un fir conectat la șina de masă pentru pinii de GND al acestor componente. Pinii pentru tensiune a componentelor au fost distribuiți separat pentru cablarea corectă pe placa Arduino.

Conexiunea LED-ului roșu s-a făcut în felul următor:

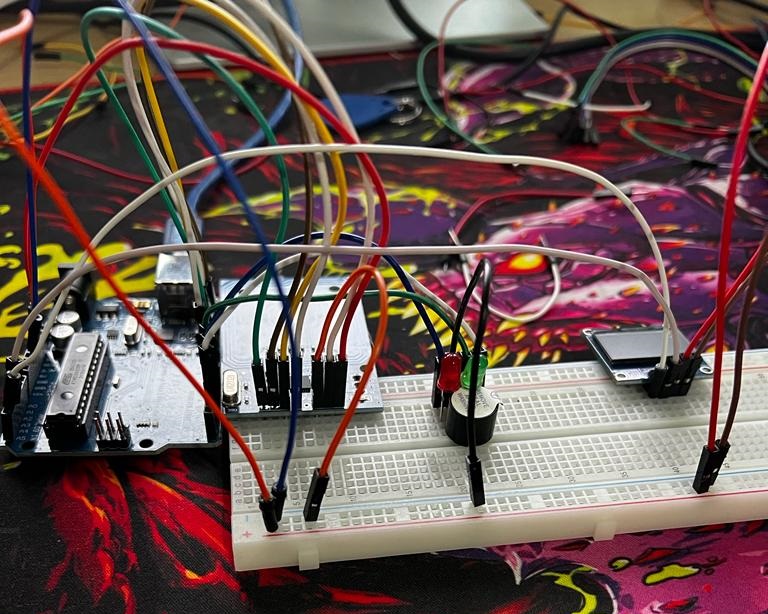
* Catodul LED-ului a fost conectat la șina de masă a breadboardului - pentru a completa circuitul și a permite fluxul curentului în mod corespunzător prin LED.
* Anodul LED-ului a fost conectat la pinul 4 al plăcuței Arduino - se creează o conexiune directă între pinul digital și partea pozitivă a LED-ului. Astfel, când se aplică un semnal digital de tensiune (HIGH) pe pinul 4, se permite fluxul curentului prin LED, ceea ce duce la iluminarea acestuia. Atunci când semnalul digital este în starea de tensiune scăzută (LOW), se întrerupe curentul și LED-ul se va stinge.

Conexiunea LED-ului verde s-a făcut în felul următor:

* Catodul LED-ului a fost conectat la șina de masă a breadboardului.
* Anodul LED-ului a fost conectat la pinul 5 al plăcuței Arduino.

Conexiunea buzzer-ului s-a făcut în modul următor:

* Pinul GND al buzzer-ului a fost conectat la șina de masă a breadboardului - pentru a completa circuitul și a permite fluxul curentului necesar pentru a genera sunet prin buzzer.
* Pinul VCC al buzzer-ului a fost conectat la pinul 2 al plăcuței Arduino - se asigură faptul că buzzer-ul primește tensiunea necesară pentru a funcționa corect. Aceasta permite ca semnalul de tensiune generat de pinul 2 să alimenteze buzzer-ul și să îi permită să genereze sunetul dorit.



*Figura 43. Cablare OLED.*

Figura 43. prezintă modul în care am făcut conexiunile ecranului OLED cu plăcuța de dezvoltare Arduino Uno. Pentru a realiza alimentarea, am conectat un fir de la alimentarea de 5V a plăcuței Arduino la șina de alimentare a breadboard-ului.

Conexiunile s-au efectuat în modul următor:

* Pinul GND al OLED-ului a fost conectat la șina de masă a breadboard-ului - se asigură o legătură directă cu sursa de tensiune de referință sau cu masa sistemului. Această conexiune permite ca curentul să circule în mod corect prin componentă și să se stabilească nivelul de referință al tensiunii pentru semnalele electrice care alimentează OLED-ul.
* Pinul VCC al OLED-ului a fost conectat la șina de alimentare a breadboard-ului - se furnizează sursa de tensiune necesară pentru alimentarea OLED-ului. Șina de alimentare a breadboard-ului este utilizată pentru a distribui tensiunea de alimentare către mai multe componente conectate la acea șină.
* Pinul SCL al OLED-ului a fost conectat la pinul A5 al plăcuței Arduino - se stabilește o legătură de comunicație I2C între cele două dispozitive, astfel Arduino poate transmite și recepționa date de la OLED prin intermediul acestei linii de ceas sincronizată. Conexiunea liniei SCL permite Arduino să controleze și să sincronizeze transferul de date către OLED, asigurând o comunicare corectă și fiabilă între cele două dispozitive.
* Pinul SDA al OLED-ului a fost conectat la pinul A4 al plăcuței Arduino - se stabilește o legătură de comunicație I2C între cele două dispozitive, astfel Arduino poate transmite și recepționa date de la OLED prin intermediul acestei linii de date. Conexiunea liniei SDA permite transferul de informații între Arduino și OLED, inclusiv comenzi și date necesare pentru controlul și afișarea informațiilor pe ecranul OLED.



*Figura 44. Cablare servomotor.*

În Figura 44. am cablat servomotorul în felul următor:

* Pinul VCC al servomotorului a fost conectat la șina de alimentare (5V) a breadboard-ului - se asigură că motorul primește tensiunea necesară pentru a funcționa corect. Astfel, servomotorul va putea executa mișcările și rotațiile dorite în funcție de semnalele de control pe care le primește.
* Pinul GND al servomotorului a fost conectat la șina de masă a breadboard-ului - se realizează o conexiune directă între potențialul de masă al servomotorului și placa Arduino. Acest lucru asigură că semnalele de control și tensiunile furnizate de Arduino sunt referite la aceeași masă sau potențial de referință.
* Pinul PWM al servomotorului a fost conectat la pinul 3 al plăcuței Arduino - se permite generarea semnalelor PWM necesare pentru controlul precis al poziției servomotorului. Prin modificarea duratei impulsurilor PWM trimise pe pinul 3, se poate regla unghiul la care se rotește servomotorul. Astfel, prin programarea corespunzătoare a plăcuței Arduino și transmiterea semnalelor PWM adecvate pe pinul 3, putem controla cu precizie poziția și viteza de rotație a servomotorului.



*Figura 45. Cablare modul Bluetooth.*

În figura de mai sus este prezentat modul în care am cablat modulul Bluetooth la plăcuța de dezvoltare Arduino Uno. Conexiunile au fost făcute în felul următor:

* Pinul TXD al modulului Bluetooth a fost conectat la pinul 0 (RX) al plăcuței Arduino - într-un canal de comunicație serială, precum Bluetooth, datele sunt transmise prin intermediul unui fir sau a unui canal virtual, format din două direcții - transmitere (TX) și recepție (RX). Pinul TXD al modulului Bluetooth este responsabil pentru transmiterea datelor, iar pinul RX al plăcuței Arduino este responsabil pentru recepționarea acestora.
* Pinul RXD al modulului Bluetooth a fost conectat la pinul 1 (TX) al plăcuței Arduino - se realizează o conexiune directă între pinul de recepție al modulului Bluetooth și pinul de transmitere al plăcuței Arduino. Astfel, plăcuța Arduino poate recepționa datele trimise de către modulul Bluetooth și le poate utiliza în aplicația sau programul în care este implementat sistemul inteligent de deschidere a ușii.
* Pinul VCC al modulului Bluetooth a fost conectat la șina de alimentare (5V) a breadboard-ului - se furnizează tensiunea necesară pentru a alimenta modulul Bluetooth și a-l pune în funcțiune.
* Pinul GND al modulului Bluetooth a fost conectat la șina de masă a breadboard-ului - se asigură faptul că toate semnalele și tensiunile de la modulul Bluetooth sunt referite la aceeași bază de referință electrică.

Pentru partea software a sistemului meu inteligent de deschidere a ușii am folosit mediul de dezvoltare Arduino IDE. Pentru implementarea acestuia, am atașat codul la finalul documentației în secțiunea **Cod sursă**.

Bibliotecile utilizate și instalarea acestora a fost prezentată în secțiunea **3.1.2 Programare microcontroller.**

Modificările aduse modulului Bluetooth au fost explicate în secțiunea **3.1.3 Comenzi AT proprii.**

Aplicația mobilă realizată cu ajutorul mediulului de programare vizual MIT App Inventor se poate găsi în secțiunea **3.2.2 Realizarea aplicației prin Mit App Inventor.**

# Concluzii

Având la bază un set de cunoștințe și experiențe acumulate, am realizat un proiect deosebit care implică crearea unui sistem inteligent de deschidere a ușii utilizând Arduino Uno.

Proiectul meu a trecut prin multiple etape, de la configurarea componentelor hardware, programarea plăcuței Arduino, implementarea algoritmilor necesari pentru citirea tag-urilor RFID și controlul servomotorului, până la dezvoltarea unei interfețe mobile intuitive pentru utilizator prin intermediul aplicației.

Scopul principal al proiectului a fost de a asigura accesul controlat și securizat într-un anumit loc prin utilizarea tehnologiei RFID. Sistemul creat poate recunoaște și valida tag-urile RFID autorizate, permițând deschiderea ușii prin acționarea servomotorului. LED-urile și buzzer-ul au fost utilizate pentru a furniza feedback vizual și auditiv utilizatorului cu privire la starea sistemului și rezultatul procesului de autentificare.

Integrarea modulului Bluetooth în proiect a oferit utilizatorului posibilitatea de a controla sistemul prin intermediul aplicației mobile, adăugând astfel flexibilitate și comoditate.

În continuare, există potențial pentru dezvoltarea ulterioară a proiectului, prin adăugarea unor funcționalități suplimentare și optimizarea performanței. Se poate considera implementarea unui algoritm PID pentru a obține un control mai precis al servomotorului și menținerea constantă a parametrilor.

În concluzie, proiectul meu a constituit o oportunitate de a aplica cunoștințele și abilitățile dobândite într-un mod practic și inovativ. Am învățat nu doar aspecte teoretice și practice din domeniul tehnologiilor IoT și programării hardware, ci am și dezvoltat abilități de gestionare a proiectelor și de comunicare cu diferite componente și tehnologii. Prin acest proiect, am dobândit experiență valoroasă și am demonstrat capacitatea de a crea și implementa soluții tehnologice complexe și eficiente.

# Bibliografie

[1] "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication" de Klaus Finkenzeller.

[2] Fernando, Dejan, Smithens, B., Dan, Marc, Williams, D., Tom, Ron, Guerrero, J., Seppe, Nedelkovski, D., Khan, J. M., Baidar, Tobo, Moroisa, Moroisa, Facundo, Deddy, Hewitt, L., … Arif, T. (2022, February 17). How RFID works and how to make an Arduino based RFID Door Lock. How To Mechatronics. <https://howtomechatronics.com/tutorials/arduino/rfid-works-make-arduino-based-rfid-door-lock/>.

[3] Best face recognition door locks. Swiftlane. (2023, June 20). <https://www.swiftlane.com/blog/best-face-recognition-locks/>.

[4] Buek, E. (2020, May 21). Iris Scanning Technology &amp; Biometric Vault Security. Safe Haven Private Vaults. <https://safehavenvaults.com/iris-scanning-technology/>.

[5] Ellsworth, L. (2023, April 26). 8 must-have physical data center security features. TierPoint, LLC. <https://www.tierpoint.com/blog/8-must-have-physical-data-center-security-features/>

[6] “Biometric Authentication: A Machine Learning Approach" de Ravindra Das

[7] "Handbook of Iris Recognition" de Kevin W. Bowyer, Mark J. Burge și John H. Connell

[8] CNET. (n.d.-b). 5 reasons you need a smart lock right now - video. CNET. <https://www.cnet.com/videos/5-reasons-you-need-a-smart-lock-right-now/>

[9] "Speaker Verification: A Tutorial" de Douglas A. Reynolds și Richard Rose

[10] Security management: A publication of ASIS International. SM Homepage. (n.d.). <http://www.securitymanagement.com/>

[11] REMOTE ACCESS CONTROLLING SOLUTION, <https://www.kincony.com/remote-access-controlling-solution.html>

[12] Web-based Door Access Control System. Kisi. (n.d.). <https://www.getkisi.com/guides/web-based-access-control>

[13] Home. Arduino. (n.d.-a). <https://www.arduino.cc/>

[14] ATMEGA328P - Microchip Technology. (n.d.). <https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf>

[15] Arduino Uno - JavaTpoint. www.javatpoint.com. (n.d.). <https://www.javatpoint.com/arduino-uno>

[16] Team, T. A. (n.d.). Arduino Uno. Arduino Documentation. <https://www.arduino.cc/en/Main/ArduinoBoardUno>

[17] Admin. (2022, April 15). What is RFID RC522 module: How does it work " electroduino. ElectroDuino. <https://www.electroduino.com/what-is-rfid-rc522-module-how-does-it-work/>

[18] MFRC-522 RFID Reader Module. CE Store. (2023, June 22). <https://www.cestore-mm.com/product/mfrc-522-rfid-reader-module/>

[19] Bluetooth module HC-05 Pinout, at Commands &amp; Arduino programming .. ElectronicWings. (n.d.). <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05->

[20] Collado, E., Dejan, James, B., Suyi, M., Sabith, Bart, Nedelkovski, D., &amp; Anand. (2022, May 17). How to control servo motors with Arduino - Complete Guide. How To Mechatronics. <https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

[21] moont087, &amp; Instructables. (2020, July 21). Memory button for potentiometer servo setup on Arduino: Motion memory loop. Instructables. <https://www.instructables.com/Memory-Button-for-Potentiometer-Servo-Setup-on-Ard/>

[22] Fuller, J. (2022, August 1). SSD1306 128x64 mono 0.96 inch I2C OLED display. Datasheet Hub. <https://datasheethub.com/ssd1306-128x64-mono-0-96-inch-i2c-oled-display/>

[23] Notes, E. (n.d.). Understanding light emitting diode specifications &amp; characteristics. Electronics Notes. <https://www.electronics-notes.com/articles/electronic_components/diode/light-emitting-diode-led-datasheet-specifications-parameters-characteristics.php>

[24] Agarwal, T. (2021, July 26). Buzzer : Working, types, circuit, Advantages &amp; Disadvantages. ElProCus. <https://www.elprocus.com/buzzer-working-applications/>

[25] Active passive buzzer. Components101. (n.d.). <https://components101.com/misc/buzzer-pinout-working-datasheet>

[26] 15pcs super loud active passive piezo buzzer alarm 12mm DIA DC 5v 2 Terminals Electronic Continuous Sound Buzzer for arduino: Buy online at best price in UAE. Buy Online at Best Price in UAE - Amazon.ae. (n.d.). <https://www.amazon.ae/Passive-Terminals-Electronic-Continuous-Arduino/dp/B07D8MQ9ZW>

[27] LED 5mm green. Makestore. (2023, May 16). <https://www.makestore.in/product/led-5mm-green-led/>

[28] LED - basic red 5mm. COM. (n.d.). <https://www.sparkfun.com/products/9590>

[29] Amazon.in. (n.d.). <https://www.amazon.in/BUD-Industries-BC-32627-Colored-Ribbon/dp/B0137JZJZG>

[30] Amazon.ae. (n.d.). <https://www.amazon.ae/Arduino-Half-size-breadboard/dp/B07MCQZVHX>

[31] WatElectronics. (2021, June 19). Breadboard : Types, connections, Advantages &amp; Disadvantages. WatElectronics.com. <https://www.watelectronics.com/breadboard-construction-types-working/>

[32] Male to male jumper wire. Cytron Technologies. (n.d.). <https://www.cytron.io/p-male-to-male-jumper-wire>

[33] Miguelbalboa. (n.d.). RFID: Arduino Rfid Library for MFRC522. GitHub. <https://github.com/miguelbalboa/rfid>

[34] Adafruit GFX Graphics Library - Adafruit Industries. (n.d.-a). <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-gfx-graphics-library.pdf>

[35] Adafruit. (n.d.). Adafruit/ADAFRUIT\_SSD1306: Arduino Library for SSD1306 monochrome 128x64 and 128x32 oleds. GitHub. <https://github.com/adafruit/Adafruit_SSD1306>

[36] Mit App Inventor. About Us. (n.d.). <https://appinventor.mit.edu/explore/about-us.html>

[37] Elena Lulcheva. Coding Girls. (n.d.). <https://www.coding-girls.com/blog/building-android-applications-mit-app-inventor>

# Rezumat

Proiectul meu a constat în crearea unui sistem inteligent de deschidere a ușii utilizând Arduino Uno și diverse componente electronice.

Am implementat modulul RFID pentru autentificarea utilizatorilor și controlul servomotorului pentru deschiderea ușii. Am dezvoltat o aplicație mobilă pentru interacțiunea cu sistemul prin intermediul modulului Bluetooth. Am integrat un display OLED pentru afișarea informațiilor și am utilizat LED-uri și un buzzer pentru feedback vizual și auditiv. Am realizat conexiuni corecte între componentele electronice și am programat Arduino Uno folosind Arduino IDE.

Proiectul a presupus abordarea atât a aspectelor tehnice, cum ar fi configurarea și controlul componentelor hardware, cât și a aspectelor software, inclusiv programarea algoritmilor de citire RFID și controlul servomotorului.

Am obținut o soluție funcțională care asigură accesul controlat și securizat într-un anumit loc, cu posibilitatea de a controla sistemul de la distanță prin intermediul aplicației mobile.

# Abstract

My project consisted of creating a smart door opener using the Arduino Uno and various electronic components.

I have implemented the RFID module for user authentication and servo motor control for opening the door. I have developed a mobile app for interacting with the system via Bluetooth module. I have integrated an OLED display for information display and used LEDs and a buzzer for visual and auditory feedback. I have made the correct connections between the electronic components and programmed the Arduino Uno using the Arduino IDE.

The project involved addressing both technical aspects, such as configuring and controlling hardware components, and software aspects, including programming RFID reading algorithms and servo motor control.

I have achieved a functional solution that provides a controlled and secure access to a specific location, with the ability to control the system remotely via the mobile app.

# Cod sursa

#include <MFRC522.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#include <SPI.h>

#include <Servo.h>

#include <SoftwareSerial.h>

#define OLED\_RESET -1

Adafruit\_SSD1306 display(OLED\_RESET);

#define SS\_PIN 10

#define RST\_PIN 9

#define LED\_G 5

#define LED\_R 4

#define BUZZER 2

Servo myServo;

int bluetoothTX = 0;

int bluetoothRX = 1;

SoftwareSerial Blue(bluetoothTX, bluetoothRX);

MFRC522 rfid(SS\_PIN, RST\_PIN);

int code[] = {243, 213, 106, 14};

String uidString;

void setup() {

  Serial.begin(9600);

  SPI.begin();

  Blue.begin(9600);

  myServo.attach(3);

  myServo.write(0);

  pinMode(LED\_G, OUTPUT);

  pinMode(LED\_R, OUTPUT);

  pinMode(BUZZER, OUTPUT);

  digitalWrite(LED\_G, LOW);

  digitalWrite(LED\_R, LOW);

  noTone(BUZZER);

  rfid.PCD\_Init();

  display.begin(SSD1306\_SWITCHCAPVCC, 0x3C);

  display.clearDisplay();

  display.setTextColor(WHITE);

  display.setTextSize(1);

  display.setCursor(10, 0);

  display.print("Apropie tagul RFID");

  display.display();

}

void loop() {

  if (rfid.PICC\_IsNewCardPresent()) {

    readRFID();

  }

  if (Blue.available() > 0) {

    unsigned int servopos = Blue.read();

    if (servopos == 'A') {

      display.clearDisplay();

      display.setCursor(10, 0);

      display.print("Acces autorizat");

      display.display();

      digitalWrite(LED\_G, HIGH);

      tone(BUZZER, 500);

      delay(300);

      noTone(BUZZER);

      myServo.write(180);

      delay(3000);

      myServo.write(0);

      digitalWrite(LED\_G, LOW);

      display.clearDisplay();

      display.setCursor(10, 0);

      display.print("Apropie tagul RFID");

      display.display();

    }

  }

}

void readRFID()

{

  rfid.PICC\_ReadCardSerial();

  Serial.print(F("\nTip PICC: "));

  MFRC522::PICC\_Type piccType = rfid.PICC\_GetType(rfid.uid.sak);

  Serial.println(rfid.PICC\_GetTypeName(piccType));

  // Check is the PICC of Classic MIFARE type

  if (piccType != MFRC522::PICC\_TYPE\_MIFARE\_MINI &&

    piccType != MFRC522::PICC\_TYPE\_MIFARE\_1K &&

    piccType != MFRC522::PICC\_TYPE\_MIFARE\_4K) {

    Serial.println(F("Tagul utilizat nu este unul tip MIFARE Classic."));

    return;

  }

  clearUID();

  Serial.println("Scanned PICC's UID:");

  printDec(rfid.uid.uidByte, rfid.uid.size);

  uidString = String(rfid.uid.uidByte[0]) + " " + String(rfid.uid.uidByte[1]) + " " + String(rfid.uid.uidByte[2]) + " " + String(rfid.uid.uidByte[3]);

  printUID();

  int i = 0;

  bool match = true;

  while (i < rfid.uid.size) {

    if (!(rfid.uid.uidByte[i] == code[i])) {

      match = false;

    }

    i++;

  }

  if (match) {

    Serial.println("\nTAG existent in baza de date");

    display.clearDisplay();

    display.setCursor(10, 0);

    display.print("Acces autorizat");

    display.display();

    printUID();

    digitalWrite(LED\_G, HIGH);

    tone(BUZZER, 500);

    delay(300);

    noTone(BUZZER);

    myServo.write(180);

    delay(3000);

    myServo.write(0);

    digitalWrite(LED\_G, LOW);

    display.clearDisplay();

    display.setCursor(10, 0);

    display.print("Apropie tagul RFID");

    display.display();

    clearUID();

  } else {

    Serial.println("\nTag necunoscut");

    display.clearDisplay();

    display.setCursor(10, 0);

    display.print("Acces respins");

    display.display();

    printUID();

    digitalWrite(LED\_R, HIGH);

    tone(BUZZER, 300);

    delay(1000);

    digitalWrite(LED\_R, LOW);

    noTone(BUZZER);

    display.clearDisplay();

    display.setCursor(10, 0);

    display.print("Apropie tagul RFID");

    display.display();

    clearUID();

  }

  rfid.PICC\_HaltA();

  rfid.PCD\_StopCrypto1();

}

void printDec(byte\* buffer, byte bufferSize) {

  for (byte i = 0; i < bufferSize; i++) {

    Serial.print(buffer[i] < 0x10 ? " 0" : " ");

    Serial.print(buffer[i], DEC);

  }

}

void clearUID() {

  display.setTextColor(BLACK);

  display.setTextSize(1);

  display.setCursor(30, 20);

  display.print(uidString);

  display.display();

}

void printUID() {

  display.setTextColor(WHITE);

  display.setTextSize(1);

  display.setCursor(0, 20);

  display.print("UID: ");

  display.setCursor(30, 20);

  display.print(uidString);

  display.display();

}