Popescu Irina-Elena Bluetooth and RFID door lock

Bluetooth and RFID door lock

Autor

- nume: Popescu Irina-Elena 331CB
- grupa: 331CB

Introducere

Door lock-ul se poate deschide atat prin RFID, folosindu-se de un tag, cat si prin modul de bluetooth, folosindu-se de o aplicatie de telefon. Exista doua functii: lock si unlock door. Cand se va apela una dintre aceste functii, se vor porni led-ul si buzzer-ul, si un servomotor se va roti 180 de grade, pentru a trage de un zavor. Am vrut sa fac un door lock pentru a mi-l pune la usa mea, deoarece daca imi inchid usa, mereu trebuie sa ma ridic sa o deschid pentru altcineva; dar acum, pot sa fac totul de pe telefon.

Descriere generală

- Cand se va da lock, led-ul rosu se va aprinde de 2 ori, si buzzer-ul va suna tot de 2 ori, pe o anumita frecventa; la final, servomotorul se va roti cu 180 de grade.
- Cand se va da unlock, led-ul verde se va aprinde de 3 ori, si buzzer-ul va suna tot de 3 ori, pe o alta frecventa; la final, servomotorul se va roti cu 180 de grade, in sens opus.
- Daca tag-ul este gresit, led-ul albastru se va aprinde de 5 ori, si buzzer-ul va suna tot de 5 ori, pe o alta frecventa.
- Aplicatia de mobil a fost creata cu MIT App Inventor.

Pentru a putea da comenzile de lock/unlock din aplicatia de mobil, trebuie sa se introduca o parola in aplicatie.

Exista un unic tag care va porni functiile de lock/unlock folosind modulul de RFID, tag identificat prin numarul sau serial de 4 bytes.

Schema bloc:

Hardware Design

Lista de componente:

- Arduino UNO ATmega328p
- Breadboard
- Modul Bluetooth Master Slave HC-05
- Modul Tag RFID-RC522
- Micro servo motor SG90
- Led RGB cu catod comun
- Buzzer activ
- Fire tata-tata/mama-tata
- Rezistoare
- Tag RFID

Schema electrica:

Software Design

Am folosit Arduino IDE pentru a dezvolta programul. Bibliotecile folosite au fost:

- AddicoreRFID.h pentru modulul RFID-RC522.
- SPI.h pentru modulul RFID-RC522 - acest comunica prin SPI.
- Servo.h pentru micro servo motor SG90.

Variabilele globale sunt:

- recv_bluetooth: este 1 daca s-au primit date de la bluetooth, si 0 daca nu s-a primit nimic.
- data_bluetooth: byte-ul primit de la modulul de bluetooth.
- door_locked: reprezinta starea actuala a usii; ia valoarea 1/0 daca usa este inchisa/deschisa.
- i, state: variabile auxiliare.
- checksumRFID, strRIFD: variabile folosite in functiile pentru RFID.
- myRFID: obiectul de tip AddicoreRFID.
- myServo: obiectul de tip Servo.

Functii:

setPins()

Setez pinii pe care voi lega componentele de arduino pe **OUTPUT**:

- 7, 8, 9 pentru led-ul RGB.
- 2 pentru buzzer
- 5, 10 pentru RFID

si ii atasez obiectului de tip servo pinul PWM 3.

activatePins()

Scriu pe pinul 10 valoarea LOW si pe pinul 5 valoarea HIGH, pentru a activa RFID-ul.

setupComponents()

Initializez variabilele globale. Folosind *AddicoreRFID_Init()* initializez obiectul myRFID. Folosind *write* setez servo motorul la 0 grade, si variabila *door_locked* este 0, adica usa este deschisa. In plus, variabila *recv_bluetooth* este setata la 0 initial, pentru ca nu se primesc date de la bluetooth la setup.

setupBluetoothInterrupt()

Setez variabilele corespunzatoare pentru intreruperea pe USART, deoarece modulul de bluetooth comunica pe USART. Dezactivez intreruperile prin *cli()*. Setez *UBRR0* pe 103, pentru a configura baud rate-ul la 9600bps. Setez bitii *UCSZ01* si *UCSZ00* pe 1 in *UCSR0C* pentru a seta ca voi recepta/transmite date de 8 biti. Pentru a activa receptia/transmiterea, setez in *UCSR0B* biii *RXEN0* (activeaza receptia pe USART), *TXEN0*(activeaza transmiterea pe USART) si *RXCIE0* pe 1. La final, activez interuperile prin *sei()*.

setup()

Pornesc libraria SPI, deoarece este folosita de RFID. Apelez cele 3 functii descrise anterior si setez bitii corespunzatori pentru a activa interuperea pentru modulul de bluetooth.

set_RGB_led(int red_light, int green_light, int blue_light)

Pornesc led-ul RGB, folosind *analogWrite(pin, freq), unde freq = [0, 255], pin = {7, 8, 9}*.

servo_lock(int startp, int endp)

Pornesc servo motorul, care face o rotire de la unghiul startp la endp, folosindu-se de functia *write*. Pentru a fi o miscare lina, folosesc un for, pentru a trece prin fiecare unghi de la startp la endp, si dupa fiecare miscare fac un *delay*.

Daca as face doar un write(endp), s-ar misca prea repede.

servo_unlock(int startp, int endp)

Aplic aceeasi idee ca la functia anterioara, doar ca in sens opus.

door_routine(int redl, int greenl, int bluel, int delayt, int flickert, int freqb)

Rutina care se executa cand se da lock/unlock. De *flickert* ori, se executa:

- se porneste led-ul RGB
- se porneste buzzer-ul la o frecventa *freqb*, folosind functia *tone*
- delay
- se opreste led-ul RGB
- se opreste buzzer-ul
- delay

lock()

Functia care se apeleaza cand se inchide usa. Se apeleaza functia *door_routine*, pentru a porni led-ul si buzzer-ul, si se porneste servo motorul.

unlock()

Functia care se apeleaza cand se deschide usa. Se apeleaza functia *door_routine*, pentru a porni led-ul si buzzer-ul, si se porneste servo motorul.

wrong_tag()

Daca tag-ul pus la RFID este cel gresit, se apeleaza doar *door_routine*, dar nu se actioneaza si servo motorul.

check_tag()

Se verifica daca tag-ul pus este cel corect. Se verifica fiecare byte din codul tag-ului cu numarul din tag-ul meu. In plus, ca o verificare extra, se verifica si checksum.

Reader-ul de RFID returneaza un numar pe 5 bytes: primii 4 bytes reprezinta numarul unic al tag-ului, iar al 4-lea este checksum.

RFID_routine()

DUpa ce s-a detectat un tag, daca tag-ul este cel corect, se da lock/unlock. Daca tag-ul este gresit, se apeleaza functia corespunzatoare. Daca inainte se da lock, se apeleaza unlock, si invers.

ISR(USART_RX_vect)

Rutina de tratare a interuperii USART pentru bluetooth. Retin byte-ul primit prin *URD0*, si setez variabila *recv_bluetooth* pe 1, asta insemnand ca am primit date de la telefon.

Variabila *recv_bluetoot* este volatila pentru a ma asigura ca valoarea ei se schimba imediat.

loop()

Se verifica daca s-a gasit vreun tag, sau daca se primesc date de la telefon pe bluetooth. Daca s-a gasit tag, se verifica ce tag s-a gasit, si se verifica manual checksum. Dupa ce se apeleaza functia *RFID_routine()*, se da *halt* la modul. Daca s-au primit date de la telefon, se verifica ce functie s-a primit (lock/unlock), se apeleaza functia specifica, si se trimite inapoi la telefon valoarea corespunzatoare (1/0).

In rutina de intrerupere doar am setat variabila *recv_bluetooth* pentru ca lock()/unlock() contin **delay** si nu e bine sa ai delay in interuperi.

Functionare:

Prin RFID, cand se citeste tag-ul bun, automat se da lock(daca starea anterioara a fost unlock), sau

unlock(daca starea anterioara a fost lock). Prin bluetooth(folosindu-se de o aplicatie de mobil pe Android), se poate alege daca se doreste lock sau unlock.

La bluetooth, daca se da lock, desi usa este locked, nu se intampla nimic. (la fel si pentru unlock).

Se retine starea usii printr-o varabila cu valorile posibile **0(unlocked)** sau **1(locked)**.

Indiferent daca s-a dat lock/unlock prin RFID sau bluetooth, se apeleaza aceeasi **rutina** pentru pornit led-ul, buzzer-ul si servo motorul.

Aplicatia de mobil pentru comunicarea cu modulul de bluetooth

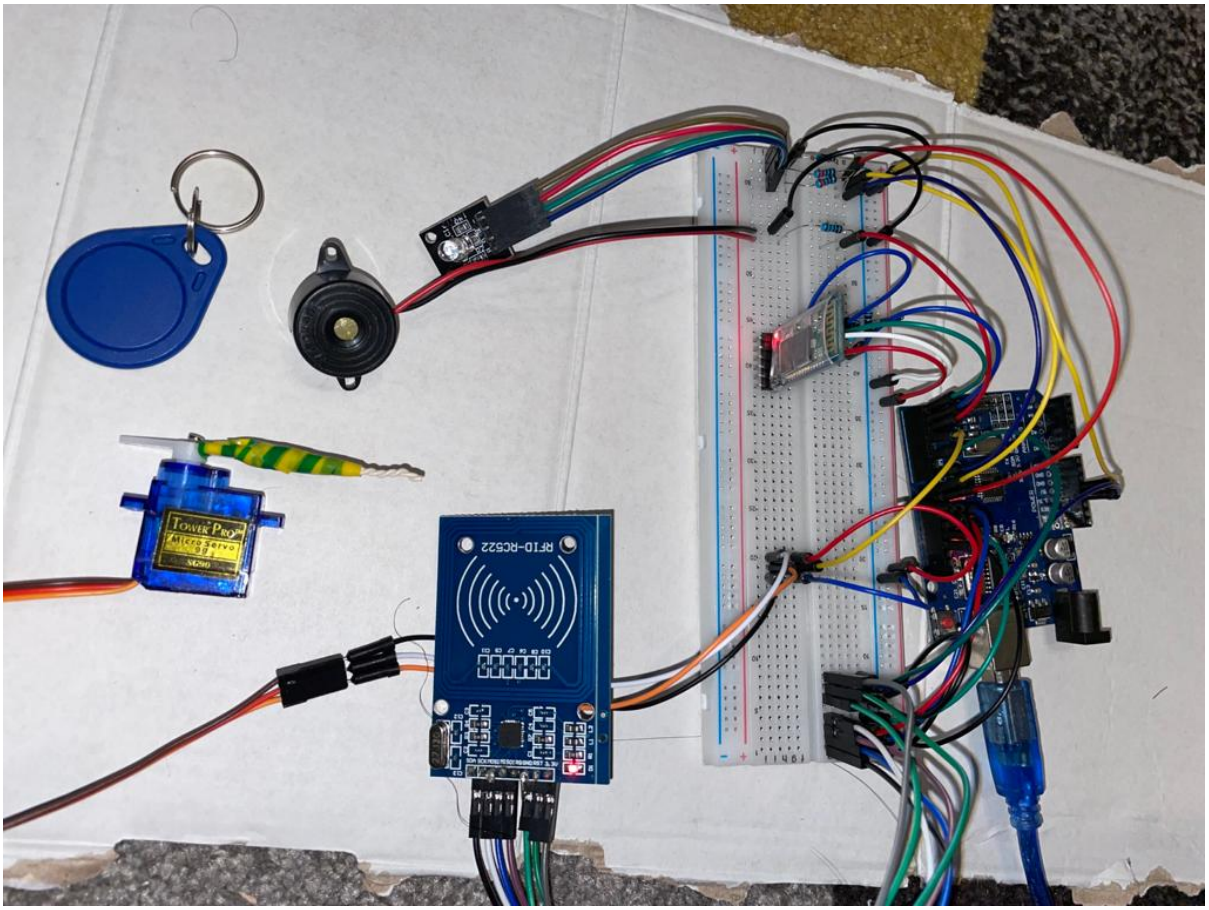
Am folosit MIT App Inventor pentru a crea aplicatia de mobil ce comunica cu modulul de bluetooth.

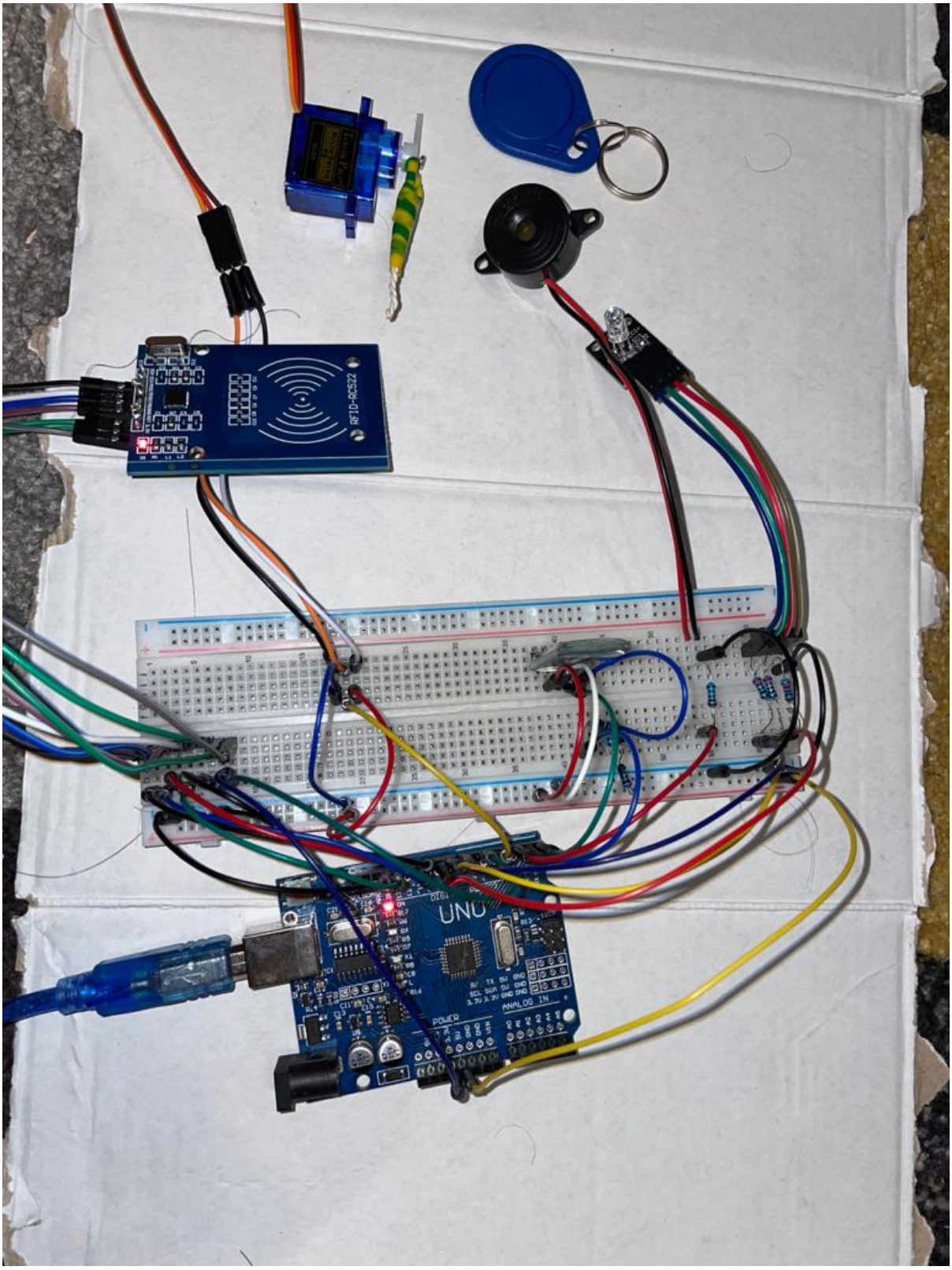
Partea de frontend:

Partea de backend:

Rezultate Obţinute

Asa arata proiectul, cu toate componentele:


TOWER PRO
Micro Servo
SG90
RFID-RC522



Demo-ul care arata cum functioneaza proiectul:demo

Concluzii

REUSITE

- Am reusit sa implementez tot ce mi-am propus, si a iesit exact cum mi-am dorit.
- Am reusit sa fac o aplicatie de mobil functionala.

PROBLEME INTAMPINATE

- Am avut probleme la a face interuperea pentru USART pentru bluetooth. In rutina de interuperea apelam si lock()/unlock(), care contin **delay()**, si din cauza asta, nu se executau corect.
- Nu am reusit inca sa fac un log in care sa retin pe aplicatia de mobil cand s-a deschis/inchis usa mereu. Am creat o baza de date in aplicatie, dar nu le afisez inca cum vreau eu.

CE AM INVATAT DIN PROIECT?

- Am invatat sa folosesc modulul de bluetooth, si in acelasi timp, am inteles mai bine cum se face o intrerupere.
- Am invatat sa citesc mai bine datasheet-ul pentru placa Arduino UNO.
- Am invatat sa fac o logica pentru un proiect pe Arduino ce contine mai multe module.

Download

Arhiva cu codul sursa si README este: arhiva_cod

Jurnal

- 25.04.2021: am ales proiectul si am creat pagina.
- 03.05.2021 - 26.05.2021: lucrat la proiect.
- 26.05.2021: update la pagina + adaugat schema electrica.
- 27.05.2021: modificat functiile din software design + concluzii.
- 30.05.2021: adaugat link demo + arhiva cod + bibliografie

Bibliografie/Resurse

- PM-laborator 0 - scrieri/citiri digitale
- PM-laborator 1 - Intrerupere
- PM-laborator 2 - USART
- PM-laborator 3 - PMW - led RGB, servomotor
- LED RGB
- BUZZER
- Servomotor
- RFID
- Bluetooth HC-05
- PDF