Controlul unei uși cu ajutorul telefonului mobil

Cuprins

Componența echipei


Rotaru Cristian


Băț Mihail


Florică Vera

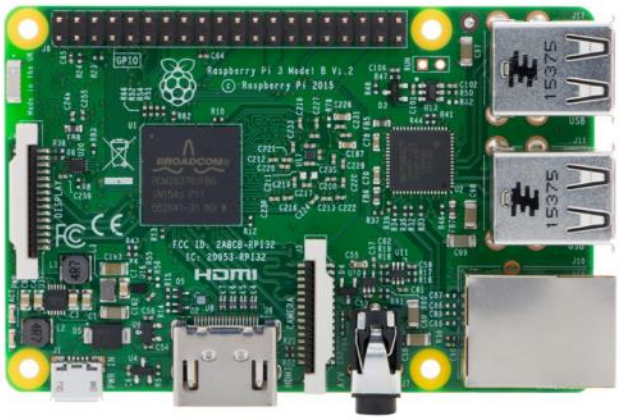
Descrierea problemei

Proiectul presupune implementarea unui mecanism prin care utilizatorul să poată controla ușa unui garaj cu telefonul mobil. De ce anume cu telefonul? Deoarece acesta este dispozitivul cel mai utilizat de către majoritatea oamenilor. E foarte puțin probabil ca cineva să uite (sau să lase) acest dispozitiv acasă atunci când pleacă undeva, spre deosebire de alte lucruri, precum cheia sau telecomanda ușii de la garaj.

Aplicația de pe telefonul mobil trebuie să ofere o interfață prin care accesul la comenzile de control să fie rapid, dar și să ofere un nivel de securitate.

Componente utilizate

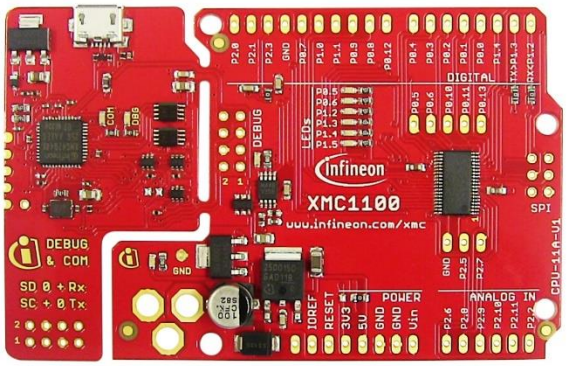
1. Raspberry Pi 3 Model B



Acest Single Board Computer are la bază un SoC Broadcom BCM2837, cu un procesor ARM Cortex-A53 cu 4 nuclee ce rulează la 1,2GHz, 1GB RAM și un controller 100MBit Ethernet.

Rolul acestei componente în cadrul proiectului este rularea unui server TCP care să poată primi comenzi de la distanță (deschidere sau închidere a ușii). Acestea vor fi procesate și trimise unei alte componente care să le execute (un XMC1100).

2. Infineon XMC1100 Boot Kit



Această placă de dezvoltare este bazată pe un microcontroller Infineon construit după arhitectura 32-bit RISC ARM cu procesor ARM Cortex-m0.

Rolul acestei componente va fi controlul motorului ce acționează asupra ușii și a LED-urilor din interiorul garajului. Comenzile vor fi primite de la serverul Raspberry Pi printr-un cablu USB. Alternativ, comenzile vor putea fi primite de la niște butoane.

3. Tower Pro MG995

Acesta este un servomotor care funcționează între 4,8V și 7,2V și poate fi controlat prin semnale PWM.

Acesta va fi atașat la ușa machetei și va fi controlat de placa cu microcontroller XMC1100.

4. LED-uri de putere

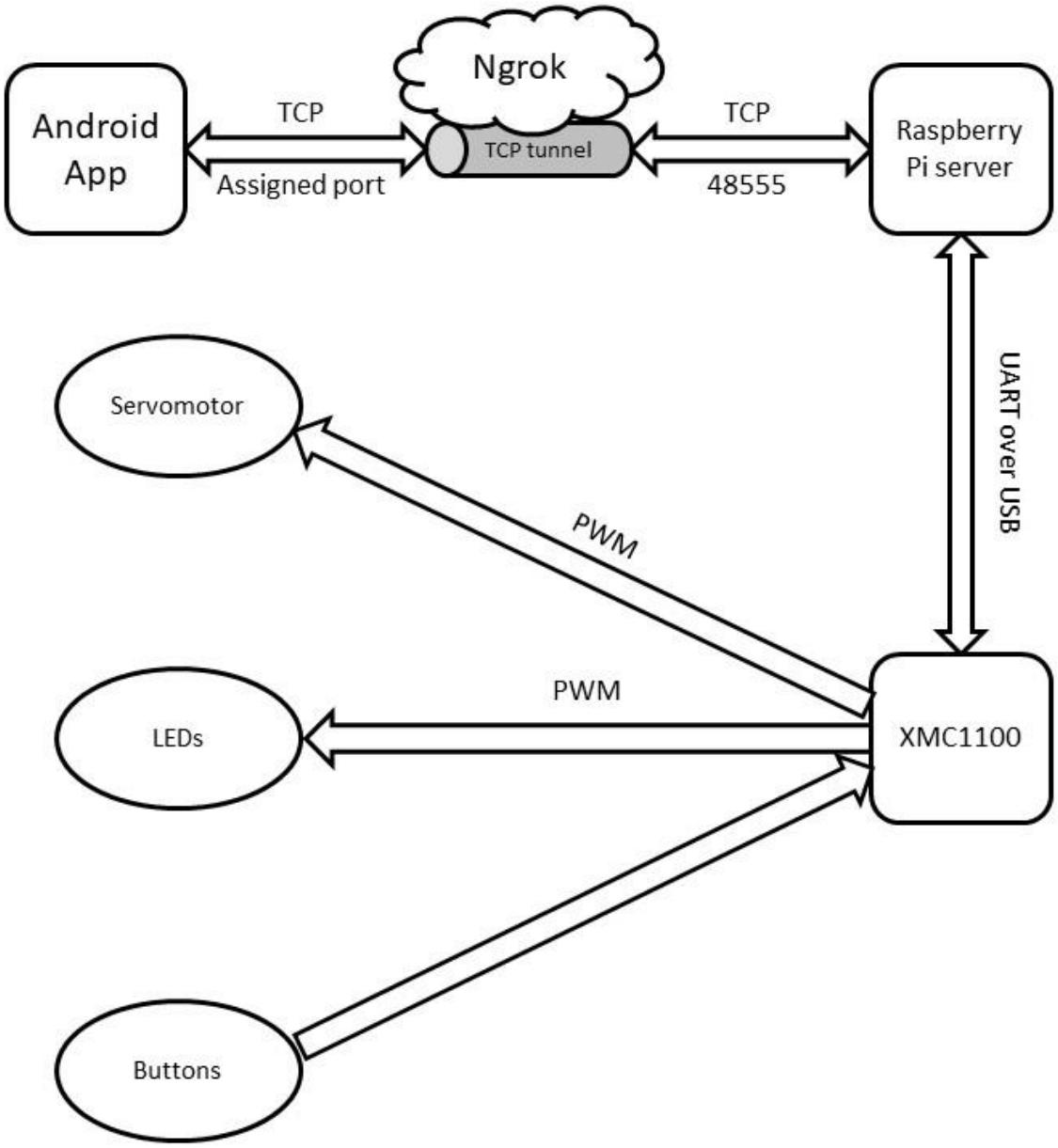
Aceste LED-uri se alimentează la 3,4V și dau o putere de 1W.

În cadrul proiectului se vor folosi două astfel de LED-uri pentru iluminarea interiorului machetei.

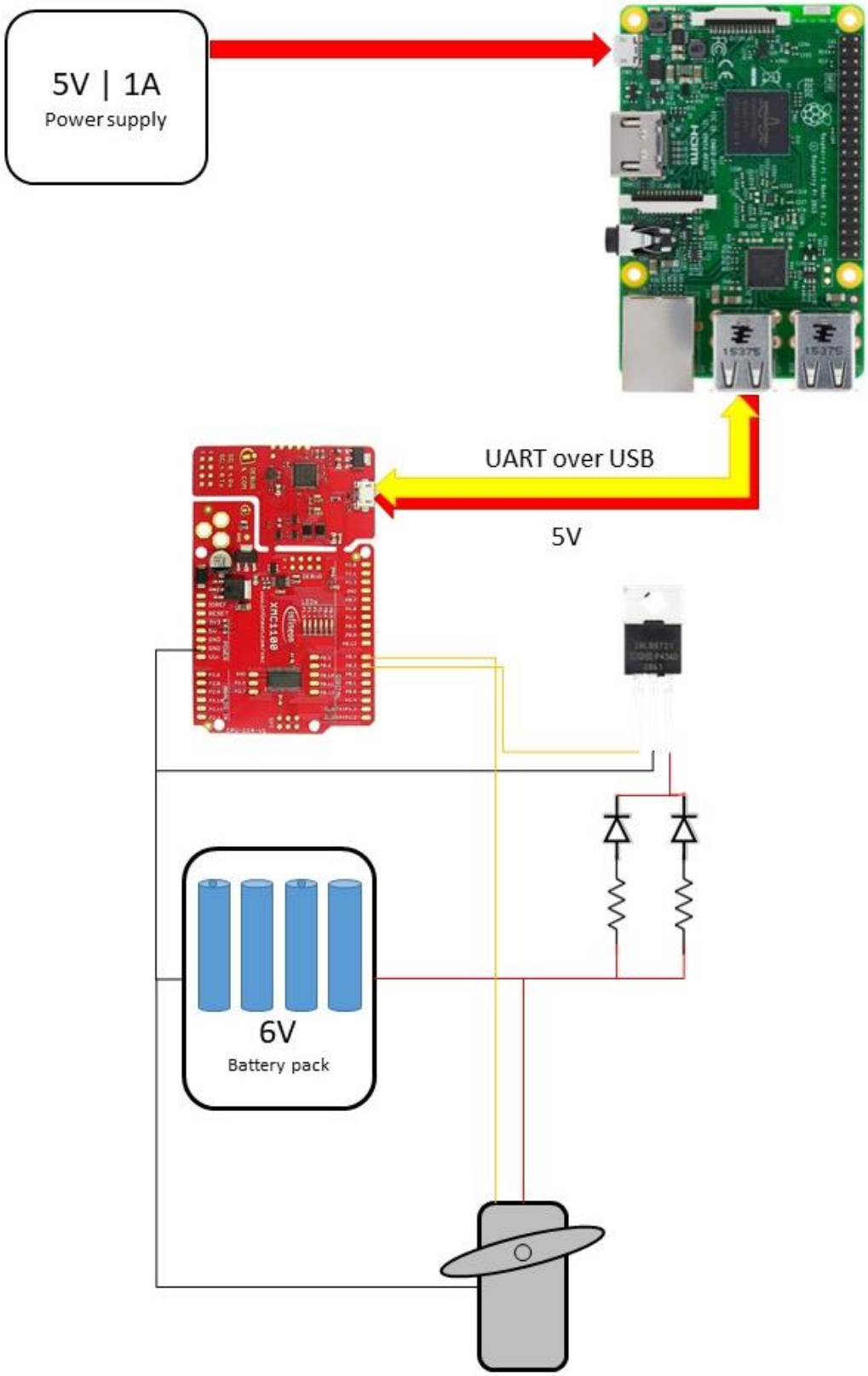
5. D44VH10G

Acesta este un transistor N-P-N de putere ce va fi folosit în proiect pentru controlul LED-urilor. Utilizarea acestuia este necesară deoarece consumul de curent al LED-urilor este ridicat și nu pot fi conectați direct la pinii digitali ai microcontrollerului.

Schema funcțională

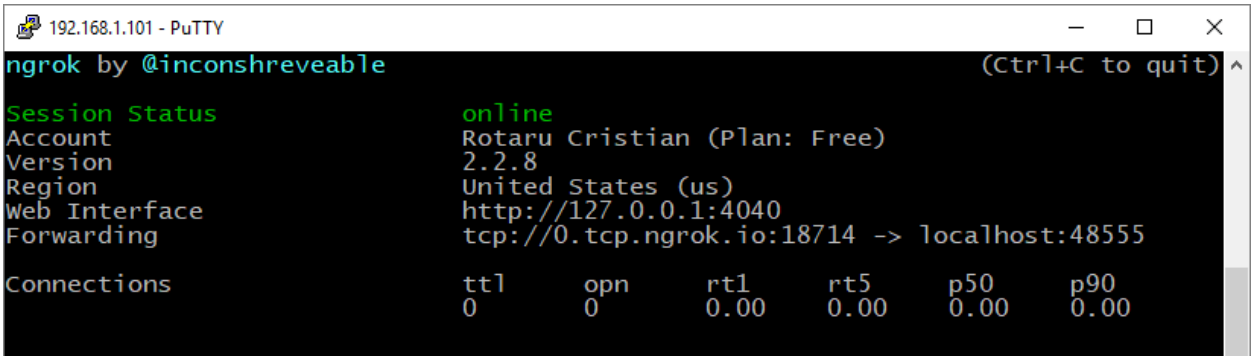


Schema electrică



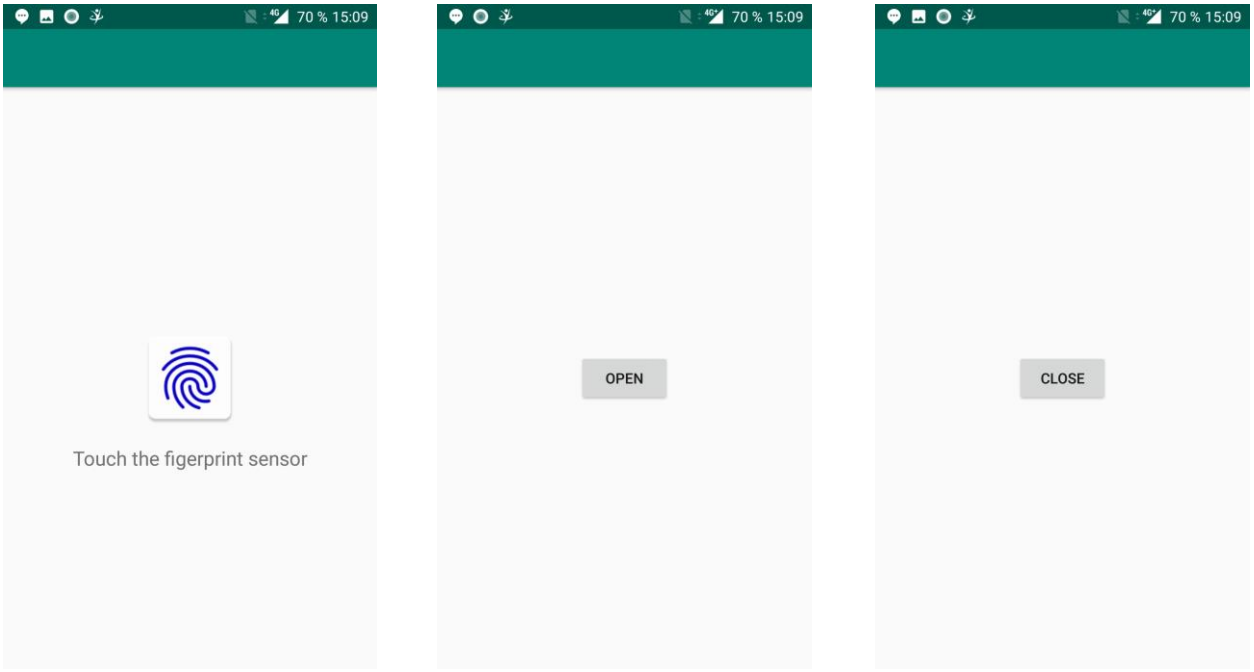
Aplicația server

În cadrul acestui proiect am ales să implementăm un server de tip TCP pentru a facilita trimiterea comenzilor de deschidere/închidere a ușii de la garaj. Pentru a permite serverului să fie vizibil în internet, s-a efectuat tunelare TCP prin serviciul Cloud Ngrok. De asemenea, serverul este proiectat astfel încât să accepte conexiuni doar de la dispozitivele cunoscute. Această funcție este realizată prin verificarea ID-ului unic al clientului, trimis la momentul conectării. Pachetele trimise și primite sunt protejate prin criptare și adăugare de zgomot. Deoarece ne-am dorit să adăugăm un plus de siguranță și să reducem interferențele, am programat serverul să permită conectarea la un moment dat doar a unui singur dispozitiv, încercările de conectare simultană de pe alte dispozitive fiind respinse.



Aplicația client

Clientul reprezintă o aplicație Android (API level 23 | Android 6.0 și mai recente). Ca măsură de securitate locală, autentificarea în aplicație se face prin intermediul cititorului de amprentă a telefonului. După conectarea la server, pe interfața grafică a aplicației va apărea un buton, cu ajutorul căruia utilizatorul poate controla ușa garajului.



Comunicarea între dispozitive

Comunicarea în fiecare dintre dispozitivele incluse în proiect este realizată prin intermediul comenzilor care sunt codificare sub forma unui caracter. Comenzile și răspunsurile posibile sunt reprezentate în tabelele de mai jos:

Comenzi client → server	
codificare	semnificație
'h'	*Handshake* Această comandă este trimisă împreună cu ID-ul unic al dispozitivului imediat după conectarea clientului la server. Această acțiune este necesară pentru a ne asigura că doar dispozitivele cunoscute au acces la interfața de control a ușii.
's'	*Get status* Această comandă este trimisă de către client pentru a afla în ce stare se află ușa la momentul curent.
'o'	*Open* Comandă trimisă de aplicația client pentru a informa serverul că utilizatorul dorește deschiderea ușii.
'c'	*Close* Comandă trimisă de aplicația client pentru a informa serverul că utilizatorul dorește închiderea ușii.

Comenzi server → client	
codificare	semnificație
'a'	*Accepted* Această comandă este trimisă la sfârșitul operațiunii de *Handshake* pentru a informa clientul că aceasta a avut loc cu succes și conexiunea este acceptată.
'd'	*Denied* Această comandă este trimisă ca răspuns la orice comandă a clientului dacă operațiunea de *Handshake* nu a avut succes. Comenzile trimise de client nu vor avea nici un efect.
'o'	*Open* Această comandă este trimisă aplicației client pentru a o informa că ușa este deschisă.
'c'	*Closed* Această comandă este trimisă aplicației client pentru a o informa că ușa este închisă.

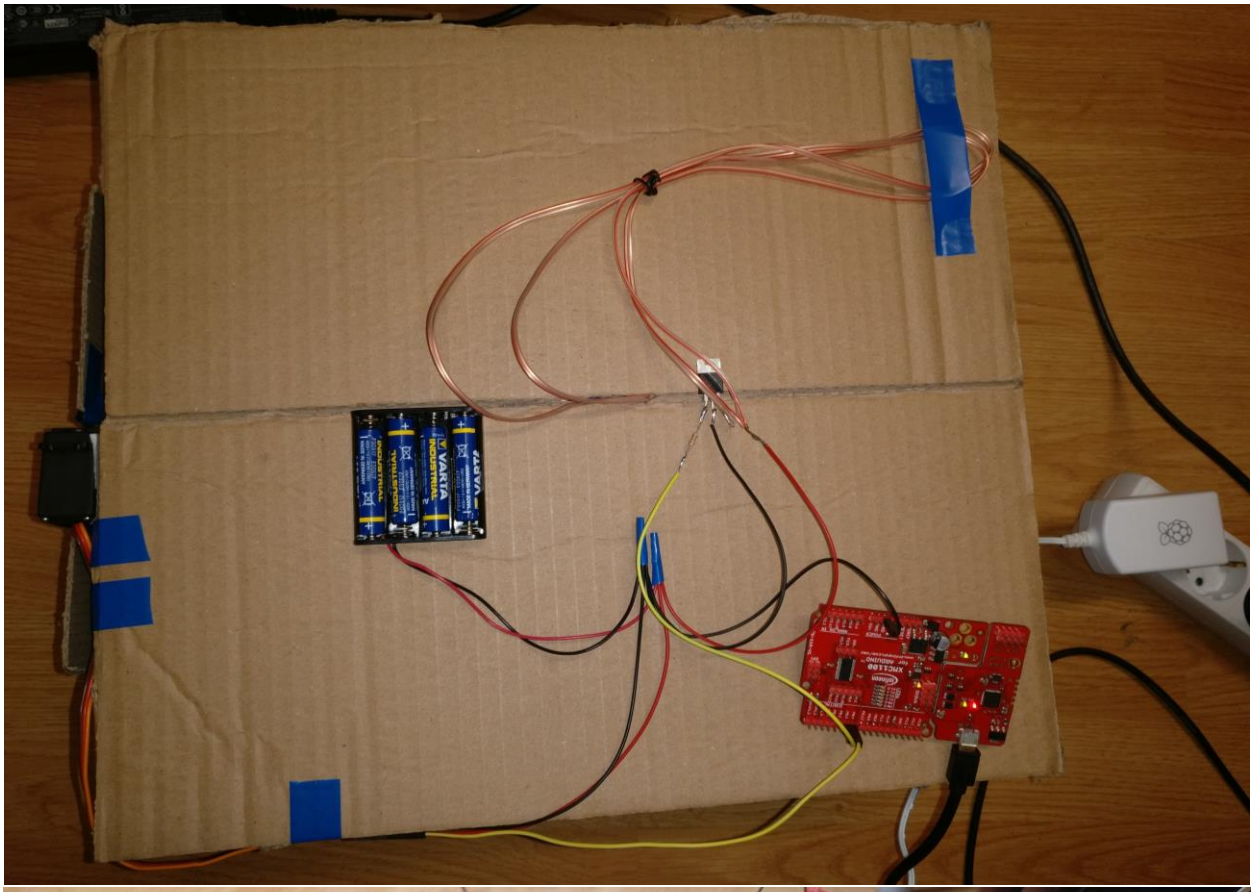
Odată ce conexiunea server-client este realizată prin internet, datele transmise trebuie protejate. Protecția este realizată prin adăugarea de zgomot și prin criptarea datelor înainte de a le trimite. Criptarea este realizată cu o cheie de 32 de biți la care se mai adaugă până la 15 octeți de zgomot. Pachetul este constituit din 21 de octeți (4 octeți de cheie, 16 octeți de date (cu tot cu zgomot) și 1 octet terminator). Împachetarea este reprezentată în imaginea de mai jos:

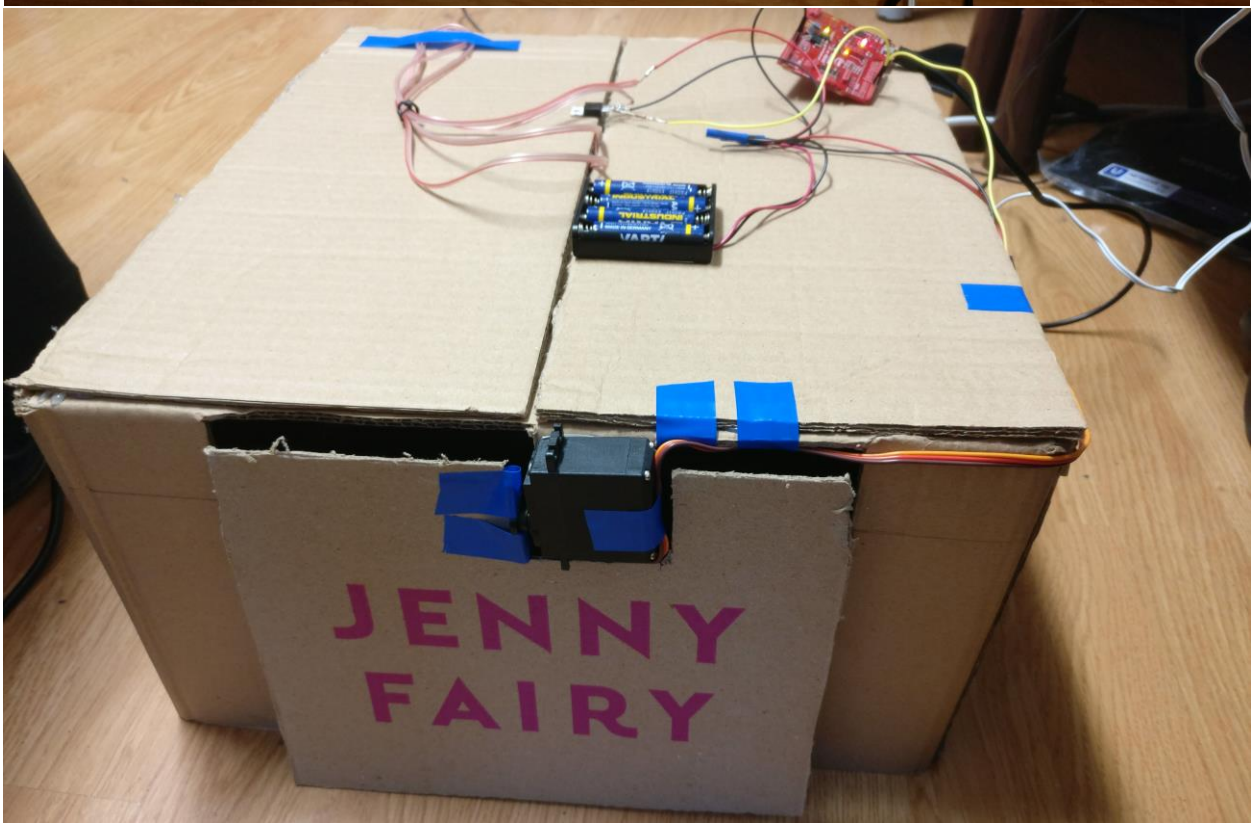
k0	k1	d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	k2	k3	\0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Comenzi server → XMC	
codificare	semnificație
`'s'`	*Get status* Această comandă este trimisă pentru a afla în ce stare se află ușa la momentul curent.
`'o'`	*Open* Această comandă este trimisă pentru a declanșa deschiderea ușii.
`'c'`	*Close* Această comandă este trimisă pentru a declanșa închiderea ușii.

Comenzi XMC → server	
codificare	semnificație
`'o'`	*Open* Această comandă este trimisă ca răspuns la comanda *Get status* pentru a informa serverul că ușa este deschisă.
`'c'`	*Close* Această comandă este trimisă ca răspuns la comanda *Get status* pentru a informa serverul că ușa este închisă.

Macheta





Referințe

[Cod sursă]
[Ngrok]
[Fingerprint authentication]
[TCP sockets in linux]