

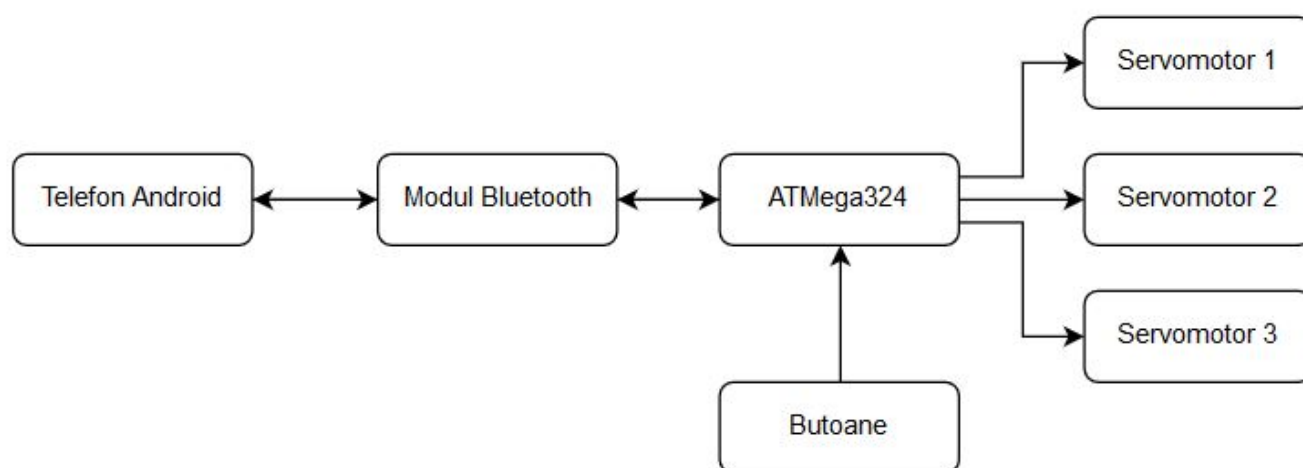
Liviu-Dan SARARIU (66905) - Customizable button pusher comandat vocal

Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

- Proiectul consta intr-un button pusher ce poate fi comandat vocal.
- Button pusherul poate fi customizat pentru aproape orice tip de aparat, utilizatorul setand pozitia butoanelor si cat de tare sa se apese pe ele.
- Button pusherul customizabil comandat vocal este folositor pentru a putea comanda vocal deviceurile ce nu au functionalitatea de comanda vocala si nu pot fi facute usor sa coopereze software cu ale deviceuri de control vocal(de tipul Google Home sau Amazon Echo)
- Proiectul reprezinta o solutie neinvaziva si relativ simpla de a adauga comanda vocala unui home device, chiar daca poate putin prea excentric pentru gustul unora (sa pui un aparat sa iti apese fizic pe butoane e totusi ceva **diferit**, desi in opinia mea, e foarte tare 🤖)

Descriere generală



- Butoanele sunt apasate de un *robot arm* simplist facut din 3 servomotoare(2 realizeaza positionarea XY si al 3-lea realizeaza apasarea butonului).
- Pentru setarea pozitiei butoanelor ce trebuiesc apasate si cat de tare sa se apese pe ele utilizatorul foloseste cate 2 butoane pentru a controla fiecare servomotor(in total 6 butoane de control).
- Dupa ce a setat pozitia servomotoarelor utilizatorul apasa pe un al 7-lea buton pentru a trimite informatiile la aplicatia de pe telefonul android(cu care se comunica prin modulul bluetooth).
- Aplicatia de pe telefon asculta utilizatorul si primeste si interpreteaza comenzile vocale ale lui dupa

care trimite comenzi de control catre servomotoare(tot prin bluetooth).

Hardware Design

Lista de piese:

- 3 servomotoare metalice digitale MG996 (90°)
- modul bluetooth HC-05
- regulator de tensiune LE33
- 7 butoane



Servomotoarele le-am conectat intre ele cu niste bucati de aluminiu standard specifice pentru servourile MG995/6. Bratul e conectat la o baza facuta din doua bucati de placaj/lemn puse in forma de L.

Software Design

Cod microcontroller:

- am controlat 2 dintre servomotoare prin PWM cu timer1
- pentru al 3-lea servo am folosit timer0
- cu butoane modific parametrii pulsurilor
- cu un buton trimit pe bluetooth parametrii pulsurilor/servo-urilor
- am activat intreruperea la receptionare de date pe seriala(blueetooth), pentru a se citi asincron date, doar atunci cand vin(fara a face verificari in bucla sau ceva)
- se primesc prin seriala/bluetooth date ce reprezinta parametrii pulsurilor/servo-urilor

Cod aplicatie Android:

- aplicatie basic ce trimite/primeste date pe bluetooth(parametrii servo-urilor)
- se tine local si o lista de "butoane"(parametrii servo) pe care utilizatorul le poate trimite la microcontroller
- se adauga in lista prin alegerea unui nume pentru buton si primirea de la microcontroller a parametrilor de la servouri, pentru pozitionarea deasupra butonului, si pentru pozitia buton apasat.

Partea de comanda vocala este facuta printr-un combo **IFTTT & Google Assistant & webserver python & Pusher Push-Notifications & Aplicatie Android**

IFTTT(If This Then That) este un service prin care se pot face multe  Pe IFTTT se pot crea appleturi ce au o actiune "This" si o actiune "That". Cand se intampla This, se executa actiunea That. This is That pot fi o sumedenie de chestii, IFTTT avand compatibilitate in ziua de azi cu foarte multe servicii si aplicatii. Eu am folosit IFTTT cu Google Assistant pentru a face partea de recunoastere vocala.

"This"-ul il reprezinta atunci cand Google Assitant aude un keyword("tell/ask button pusher to ...").

This-ul are un parametru, si anume mesajul de dupa keyword("tell/ask button pusher to turn on the toaster").

"That"-ul este un webhook pentru un webserver de python. Webhook-ul este nimic mai mult decat un url ce prin accesarea lui reprezinta trimiterea unei comenzi, cu eventuali parametrii transmisi prin parametrii url-ului. Este trimis prin webhook la webserver comanda receptionala de Google Assistant.

Lucru important e faptul ca folosind Google Assistant, timiteria de comenzi vocale se poate face si de pe un device de tipul Google Home(pretty fancy;-)). Also, IFTTT are suport si pentru Amazon Echo(Alexa) si interfatarea din applet e aproape la fel. Astfel se poate inlocui usor Google Assistant cu Amazon Echo, pentru a se putea trimite comenzi si de la un astfel de device(even more fancier ). Cum eu nu am momentan un Amazon Echo am folosit Google Assistant de pe telefon.

Webserverul python doar primeste comenzile si le face relay la aplicatia de android prin push notifications. Pentru un api mai simplist/usor de folosit am utilizat pentru push notifications Pusher Push Notifications(<https://pusher.com/push-notifications>).

La **Aplicatia Android** doar se ia mesajul de la push notification(ce contine comanda ce trebuie trimisa la button pusher), se uita in comenzile de butoane locale i daca gaseste trimite parametrii pentru servouri la microcontroller.

Rezultate Obținute



Concluzii

A fost un proiect foarte fun din care am invatat multe chestii si de care sunt mandru. Cel mai fun lucru a fost modul de a incerca sa gasesti moduri de a face ceva, combinand diverse chestii intre ele,

jerry-rigging something into existence ☒

Evident, ceva ce e un **must** de stiut, **MEREU** apar chestii neprevazute in proiectele de genul asta, si trebuie efectiv sa fii pregatit sa iti esueze orice ☒, de la partea mecanica(nu gasesti suruburi x, nu ai unealta potrivita, gaura e prea stramta), la cea electrica(nu gasesti componenta x, arzi componenta x, arzi letconul ca il atingi de firul lui de alimentare cat e in priza ☒ si alte minunatii ☒) si la software(api pentru componente ciudat, buguri de nu te miri unde date de o cate o fraza aruncata in datasheet ce iti explica ceva comportament special).

Bottom line is daca faci ceva ce iti place:

- inveti chestii noi
- e foarte rewarding sa vezi proiectul in forma sa finala :thumbs_up:

LE: proiect not done pentru ca trageau prea mult curent servourile(inainte sa le conectez mergeau toate, dar cum servourile under load trag mai mult curent, s-a dovedit a fi prea mult pentru placa. si 2 + bluetooth erau cam mult. putea fi workable cu 1 + bluetooth, dar nici aplicatia android nu e foarte functionala. Reason: la inceput nu mi-am dat seama ca setupul meu nu mergea pentru ca tragea prea mult curent, si credeam ca e de la soft, si am "stricat" destul de mult aplicatia android pana sa imi dau seama ca problema era hardware. So m-am trezit in ziua de pm fair cu o ora inainte sa se termine ca trage prea mult curent si am reusit sa fac ceva working din scraps.)

Download

[button_pusher_avr_code.zip](#)

[buttonpushercontroller_android.zip](#)

Bibliografie/Resurse

- Documentația în format [PDF](#)
- <https://bobparadiso.com/2015/08/03/universal-button-pusher/>
- <https://bobparadiso.com/2016/03/13/linear-vs-rotary-servos-in-button-pusher/>
- <https://ifttt.com/>
- <https://pusher.com/push-notifications>
- <http://www.instructables.com/id/Android-Bluetooth-Control-LED-Part-2/>

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/adraghici/liviu.sarariu>



Last update: **2021/04/14 15:07**