

# Mircea-Adrian TĂNASE (66837) - Bartender

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

## Introducere

### Ce face?

Bartender este, așa cum îi spune și numele, este gândit să vină în ajutorul barmanilor. Device-ul trebuie să prepare cocktail-uri, combinând cantități diferite dintr-o varietate de bauturi și respectând în același timp un rețetar strict. Sistemul în sine este format dintr-o serie de pompe care vor fi conectate la sticlele cu bautura și care vor pompa cantitățile cerute în rețetar din fiecare bautura într-un pahar. Retetele vor fi salvate în memoria microcontroller-ului, iar selecția rețetei va fi făcută printr-o aplicație pe smartphone, care comunică cu microcontroller-ul prin bluetooth.

### Care este scopul?

Scopul sistemului este pe de o parte ușurarea muncii barmanilor, întrucât îi scuteste pe aceștia de efort și garantează faptul că un cocktail anume va fi preparat după aceeași rețetă de fiecare dată. Pe de altă parte, sistemul poate fi folosit pentru a elimina interacțiunea barman-client, întrucât clientul poate folosi el însuși aplicația de pe smartphone pentru a își prepara bautura dorită.

### Care a fost ideea de la care ați pornit?

Ideea de la care am pornit a fost aceea că aglomeratia care se crează la bar într-un club creează neplăceri atât barmanilor -aceștia sunt suprasolicitați- cât și clienților -care așteaptă foarte mult. Inițial ideea a fost un sistem care să ordoneze cumva cererile clienților, însă apoi m-am gândit că cea mai bună soluție ar fi să lasăm barmanul să interacționeze mai mult cu aceștia, și să se ocupe mai puțin de prepararea bauturilor.

### De ce credeți că este util pentru alții și pentru voi?

Soluția mea își propune să elimine cel puțin o parte din aceste neplăceri, oferind o experiență mai puțin solicitantă și mai de încredere. E adevărat că barmanii au un farmec aparte și probabil că nu vor fi vreodată înlocuiți complet de un sistem electronic, însă sistemul vine ca o alternativă la bartending-ul clasic. Principalul atu al unui astfel de device în fața barmanilor este acela că oferă garanție că același cocktail va fi preparat mereu după aceeași rețetă, respectată strict.

## Descriere generală

### Schema bloc:



Prin intermediul aplicatiei smartphone, userul va selecta una dintre retetele existente deja si va trimite prin bluetooth un identificator al acesteia catre modulul bluetooth de pe placa microcontrollerului. Acesta va avea salvate in memorie, corespunzator fiecarei retete, o lista de numere care reprezinta ordinea si timpii in care trebuie actionata fiecare pompa pentru prepararea bauturii selectate. Pompele sunt actionate prin intermediul controllerului si sunt activate pe rand, in intervale de timp corespunzatoare pomparii cantitatii de lichid necesare. Numarul de pompe poate varia, insa aceste modificari aduc cu sine modificari in software.

## Hardware Design

Lista de piese:

- \* Placa de baza cu microcontroller - PM 2017
- \* Drivele motoare (pompe) cu alimentare separata - 3 drivele L298N, unul pentru 2 pompe
- \* Pompe de lichid pentru fiecare sticla - 6 pompe (motoare CC) cu tensiune de alimentare 3V-12V
- \* Fire pentru alimentare si legatura
- \* Modul bluetooth pentru comunicarea cu telefonul - HC-05
- \* Telefonul pe care va rula aplicatia - interfata
- \* Alimentator 240V alternativ - 12V continuu
- \* Furtune pentru lichid

### Schema electrica



Precizari implementare hardware:

-Am folosit un alimentator de 12V (12 W) pentru ca daca foloseam baterii, puterea generata de drivele pentru motoare era prea mica. Astfel, pentru fiecare sursa de alimentare din schema electrica, folosesc "+" si "-" de la alimentator.

## Software Design

In implementarea software-ului am pornit de la urmatoarele **premise**:

- bauturile trebuie sa fie turnate pe rand, deci pompate separat, asadar nu a fost nevoie sa folosesc timere, ci doar functiile din delay.h

- la capacitate maxima debitul pompelor nu e foarte mare, deci nu am folosit PWM. Atunci cand o pompa lucreaza, pe semnalul de PWM al driverului scriu valoarea 1 constant, astfel pompele lucreaza doar la capacitate maxima
- am considerat ca toate retetele se bazeaza pe o cantitate minima din orice bautura de 20ml si ca orice cantitate din orice bautura este multiplu de 20ml
- timpul masurat in care o pompa pompeaza 20ml de lichid este de aproximativ 4 secunde
- trebuie tinut cont de faptul ca la orice pompare e necesar un timp de umplere a furtunului de aproximativ 2 secunde
- o reteta anume se identifica printr-un singur caracter, care va fi primit de telefon via Bluetooth
- pentru comunicarea cu modulul Bluetooth HC-05 am folosit implementariile functiilor USART (fisierele usart.h si usart.c) din Laboratorul 1
- scopul proiectului nu este neaparat implementarea aplicatiei pe Android, asadar am folosit BlueTerm o aplicatie disponibila pe Google Play, care trimite caractere catre un terminal Bluetooth. Mentionez ca aplicatia este **open-source** (vezi resurse, 6.) si free-license.

### Detalii implementare

- pinii care controleaza pompele prin drivere sunt stocati in vectorul pumps
- functia init\_pumps() seteaza toti pinii pentru drivere in mod iesire
- functia pump(pompa, time) activeaza semnalele enable si IN pentru fiecare pompa, asteapta timpul \*time\*, dupa care realizeaza operatiile inverse
- retetele sunt salvate in matricea retetar, care pe coloana i stocheaza reteta pentru cocktail-ul i, in felul urmat: pentru fiecare din cele 6 bauturi, se cunoaste cantitatea care intra in reteta, ca multiplu de 20ml
- functia prepare(cocktail) parcurge linia din retetar corespunzatoare cocktail si activeaza pe rand pompele pentru un timp calculat in functie de debitul pompelor (macro-ul TIME20ML) si cantitatea necesara
- functia usart\_receive() primeste un caracter prin usart, de la modulul bluetooth (Laboratorul 1 PM)
- led-ul de pe pinul PD7 se aprinde atunci cand sistemul este in curs de preparare a unei bauturi
- aplicatia de pe telefon este un proiect open-source, denumit BlueTerm, care stabileste o conexiune cu un modul Bluetooth si are optiunea de a trimite cate un caracter catre acestea
- microcontroller-ul preia caracterul primit, care identifica un cocktail anume si il prepara, dupa care asteapta un nou caracter

Pentru **dezvoltare** am folosit WINAVR, Programmers Notepad si bootloader-ul de la laborator.

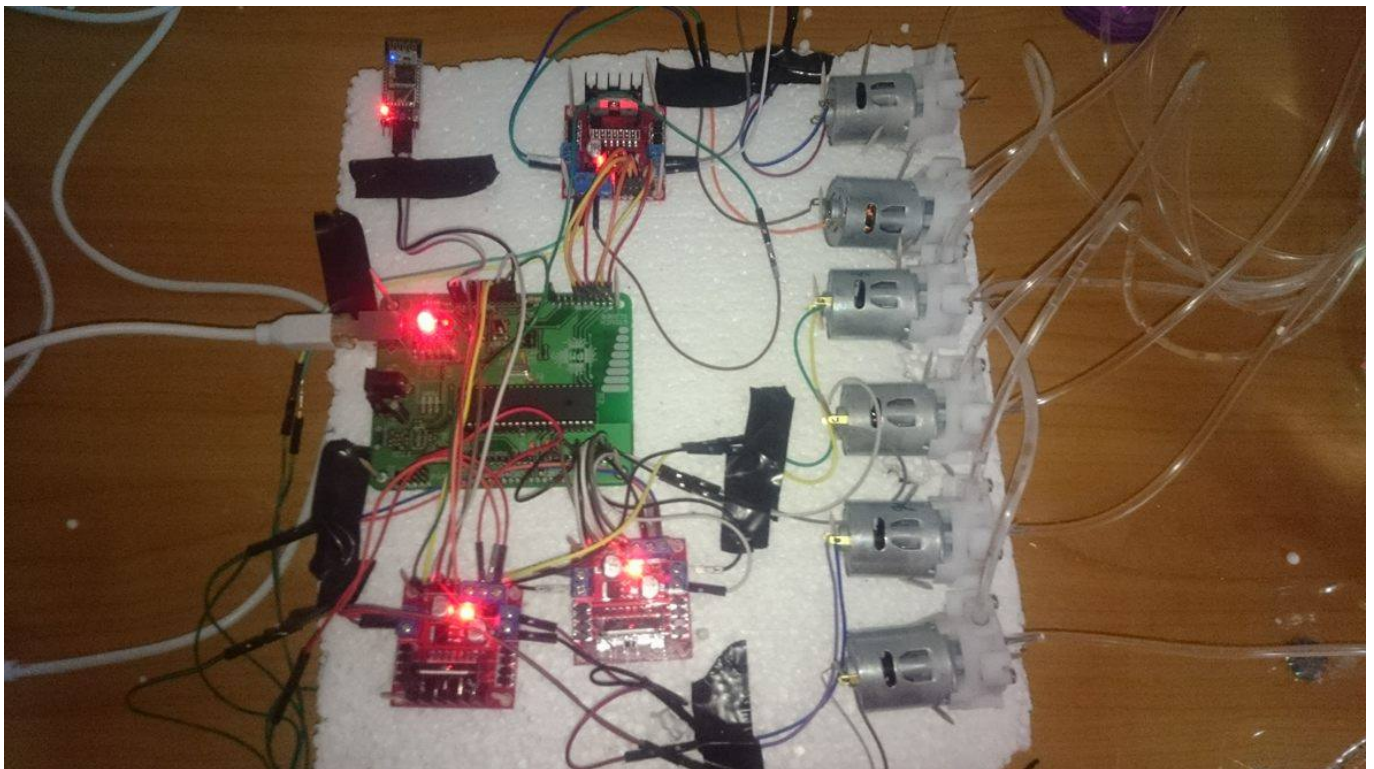
## Concluzii

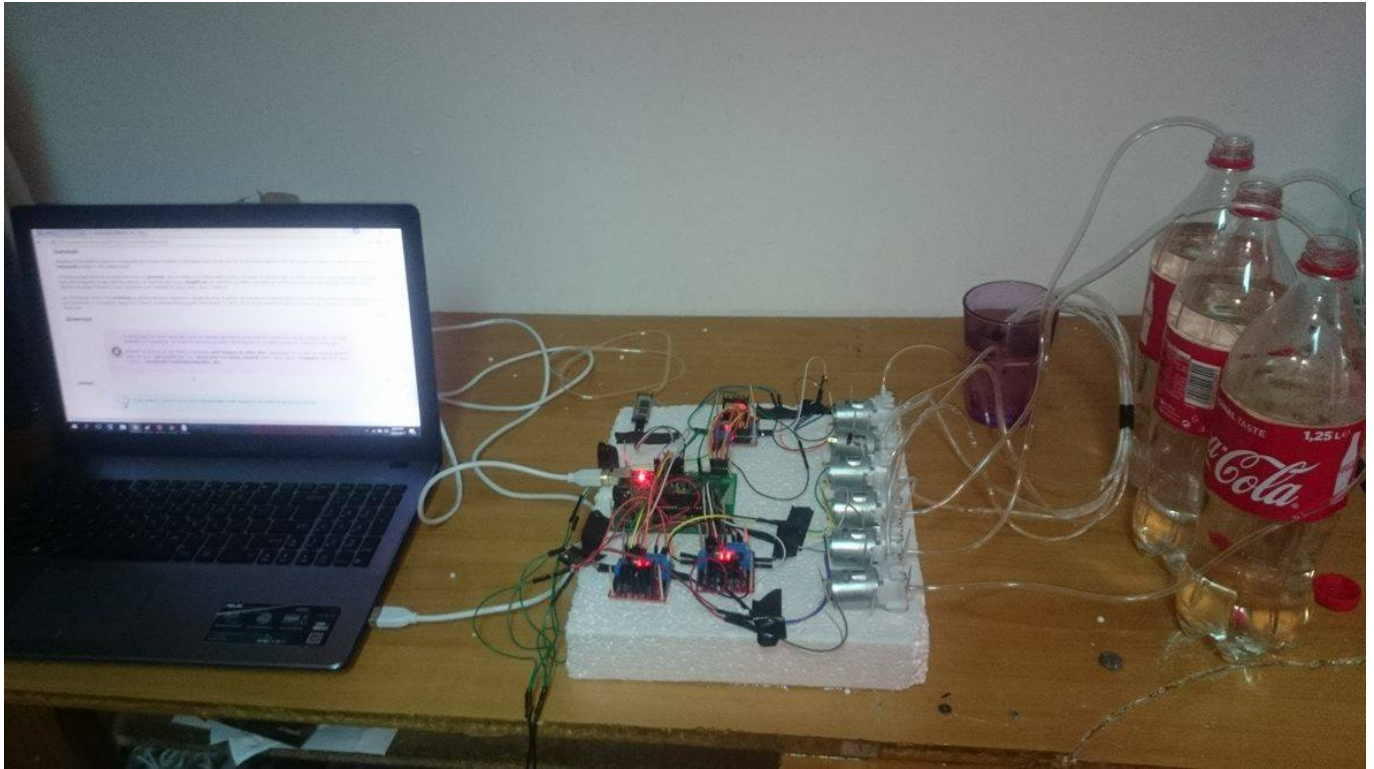
Rezultatul final poate fi observat in imaginile de mai jos. Consider ca am atins scopurile pe care mi le-am propus atunci cand am inceput proiectul, in sensul ca acesta este **functional** si poate fi intr-adevar folosit.

Trebuie precizat faptul ca produsul este doar un **prototip**, care ar trebui mult imbunatatit pentru a fi folosit in situatii reale, in primul rand din punct de vedere al design-ului (vezi imaginile), si apoi datorita faptului ca foloseste mai multe **simplificari** (se considera ca debitul nu depinde de bautura si timpul de umplere a furtunului nu depinde de nivelul lichidului, toate cantitatile sunt multiplii de 20ml, avem doar 6 bauturi).

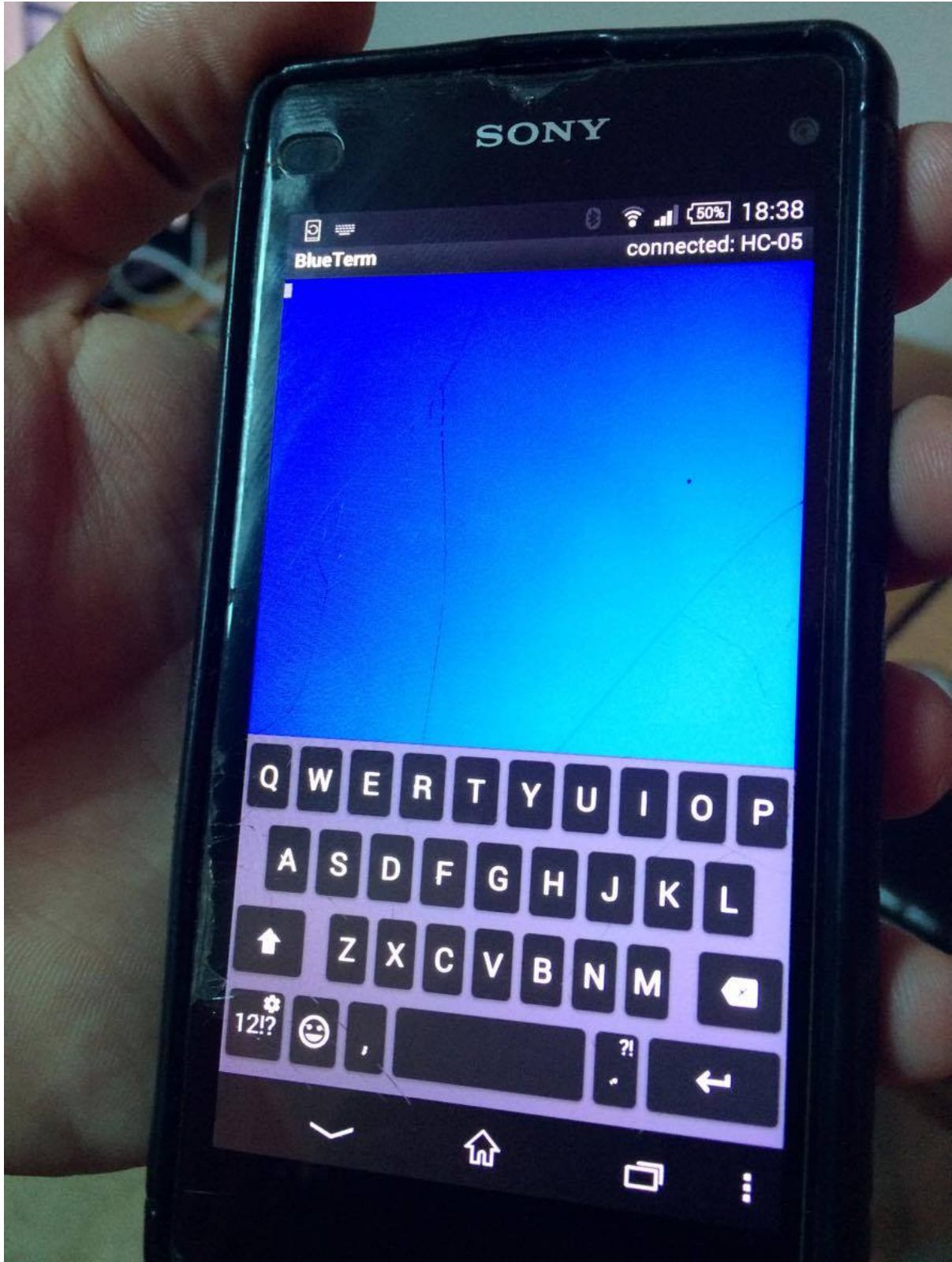
Cu toate astea, consider ca proiectul si-a atins scopul, intrucat am luat contact cu ce inseamna sa cauti cele mai potrivite componente hardware, si mai ales sa construiesti un sistem de la lipituri si pana la software.

Am intampinat foarte mari **probleme** la partea hardware, deoarece in datasheet-urile driverelor de motoare si in specificatiile motoarelor erau precizate mai ales tensiunile de functionare, nu si puterile. Faptul ca a trebuit sa efectuez foarte multe teste (baterii in serie, surse de alimentare) a generat costuri foarte mari atat in timp, cat si financiare.





Interfata aplicatiei BlueTerm:



## Download

Sursele software si schema electrica: [mircea\\_tanase\\_download.rar](http://mircea_tanase_download.rar)

Sursele se afla in folderul proiect. Sursa principala se numeste lab1.c. Pentru compilare se ruleaza comanda make, iar fisierul care trebuie incarcat pe microcontroller se numeste lab1.hex.

Aplicatia BlueTerm: <https://play.google.com/store/apps/details?id=es.pymasde.blueterm&hl=en>

## Bibliografie/Resurse

- 1.Laboratoarele de PM: <http://cs.curs.pub.ro/wiki/pm>
- 2.Datasheet ATMEGA324: [http://cs.curs.pub.ro/wiki/pm/\\_media/doc8272.pdf](http://cs.curs.pub.ro/wiki/pm/_media/doc8272.pdf)
- 3.WinAVR: <https://sourceforge.net/projects/winavr/files/>
- 4.Datasheet driver L298N: [https://www.sparkfun.com/datasheets/Robotics/L298\\_H\\_Bridge.pdf](https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf)
- 5.Datasheet bluetooth HC-05: <http://www.electronicaestudio.com/docs/istd016A.pdf>
- 6.BlueTerm source code: <https://github.com/johnhowe/BlueTerm>

\*Documentația în format [PDF](#)

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

Permanent link:  
<http://ocw.cs.pub.ro/courses/pm/prj2017/anitu/mirceatanase>



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