

Theodor STOICAN (66917) - Smart Health Surveillance

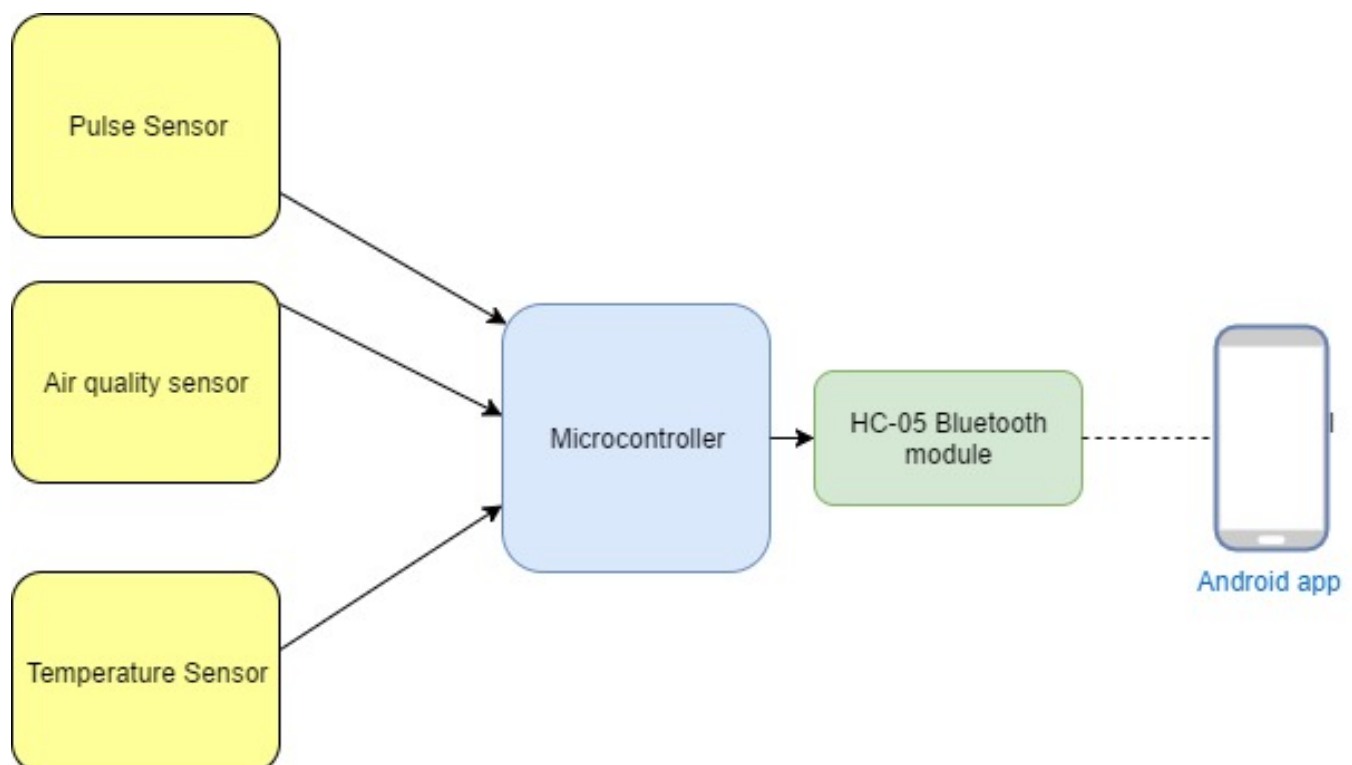
Autorul poate fi contactat la adresa: **Login pentru adresa**

Introducere

Proiectul consta in realizarea unui device care poate monitoriza un pacient intr-un salon obisnuit din spital. Dupa o interventie chirurgicala, spre exemplu, sau in sectiile pentru arsi, fiecarui pacient ii sunt monitorizate semnele vitale si alti cativa parametri, care sunt ulterior afisate pe un monitor, langa patul pacientului. Astfel, un medic poate parcurge un drum destul de lung pana la pacient pentru a verifica starea acestuia. Prin acest proiect, imi propun sa creez un prototip prin care datele culese de la pacient sa fie centralizate, undeva, intr-o camera dedicata (sau pe un device, precum o tableta), asa incat un medic sa poata vedea starea mai multor pacienti simultan.

Descriere generală

Microcontroller-ul va colecta pulsul si date despre mediul ambiant de la cei 3 senzori, va aplica filtre pe datele colectate (pentru a obtine forma pulsului din semnal, de pilda), va trimite printr-un modul Bluetooth datele la o aplicatie Android, unde vor fi afisate intr-un format user-friendly.



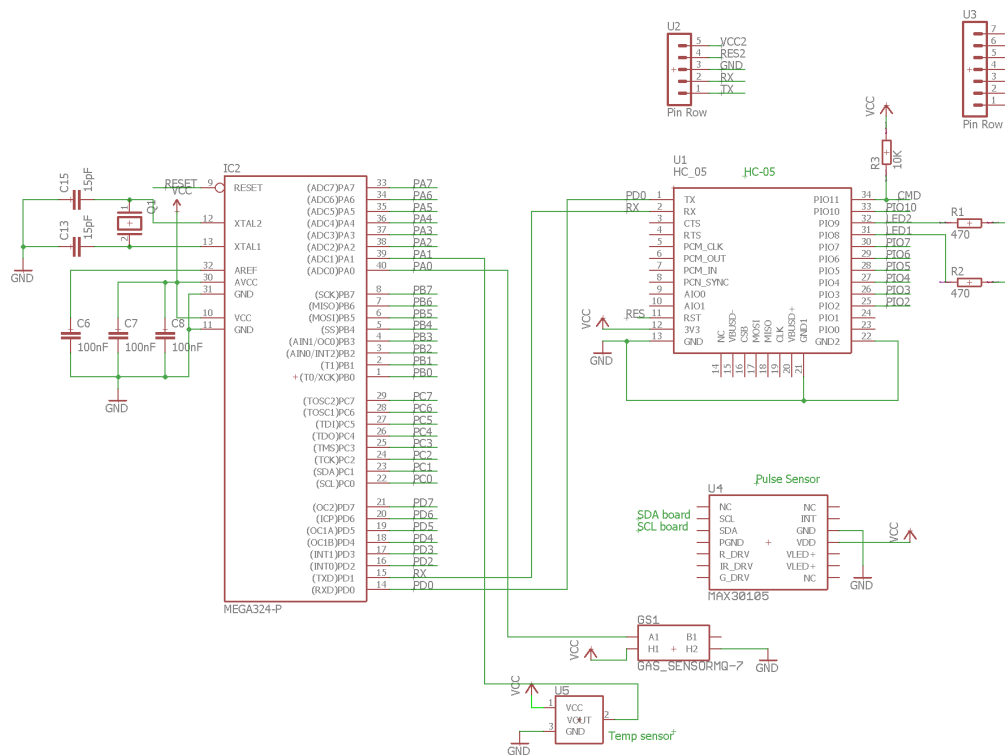
Hardware Design

Lista de piese:

Denumire	Cantitate
HC-05 Bluetooth Module	1
Microcontroller ATmega	1
SEN-11574 (senzor puls)	1
Modulul SNS-MQ135 (calitate aer)	1
Senzor brick pentru temperatura	1

Schema electrica

Am folosit un modul echivalent de la SPARKFUN pentru senzorul de puls, intrucat nu am gasit nicio biblioteca Eagle/schematic Eagle cu senzorul meu. Am notat in schema de mai jos ce reprezinta fiecare componenta.



Software Design

Mediu de dezvoltare

- Programmer's Notepad cu WinAVR
- Android Studio

Biblioteci 3rd-party

- Android GraphView pentru real-time plotting (<http://www.android-graphview.org/>)
- Biblioteci Arduino portate pentru senzori; Link-uri la "Resurse"

Algoritmi and stuff

- Pentru receptionarea pulsului(BPM-beats per minute- + IBI-interbeat interval-), folosesc un algoritm portat din Arduino. Folosesc intreruperi si un timer ca sa esantionez semnalul la fiecare 2ms.
- Pentru ceilalti senzori, fac o mediere a semnalului pe care o trimit la telefon prin Bluetooth.
- Folosesc un protocol pentru comunicatia dintre uC si telefon, implementat ad-hoc, intre telefon si microcontroller. Pe scurt, se trimit 3 valori, la telefon, pulse wave, temperature and air quality. Fiecare este delimitat de 2 caractere(aa, bb, respectiv cc), tocmai pentru a putea parseza la nivelul telefonului acest input. Un sample de payload arata cam asa: a545ab24bc130 (545-puls, 24-temperatura, 130-air quality)

Surse si functii implementate

La nivelul aplicatiei Android, prin UART nu primesc pachete de dimensiune standard. Pur si simplu vin niste bytes, pe care eu trebuie sa ii incapsulez si sa ii trimit thread-ului de UI care ii parseaza si ii afiseaza. Practic, totul functioneaza dupa urmatorul mecanism:

```
int bytes_recv = 0;
receive_bytes(buffer); //un pachet de orice dimensiune
while (i < end(buffer) && buffer[i] != 'c') continue; //caut secventa de
final a payload-ului
send_to_ui_thread(buffer[1:i]); //trimit thread-ului de UI toti acei bytes
pana la secventa de final
for (j = i; j < length(buffer); i++)
{
    buffer[j-i] = buffer[j]; // copiez acei bytes pe care nu i-am trimis la
inceputul buffer-ului
}
```

La nivelul parserului, pur si simplu ma uit dupa secventele delimitate de litera 'a', apoi 'b' si, respectiv, 'c'.

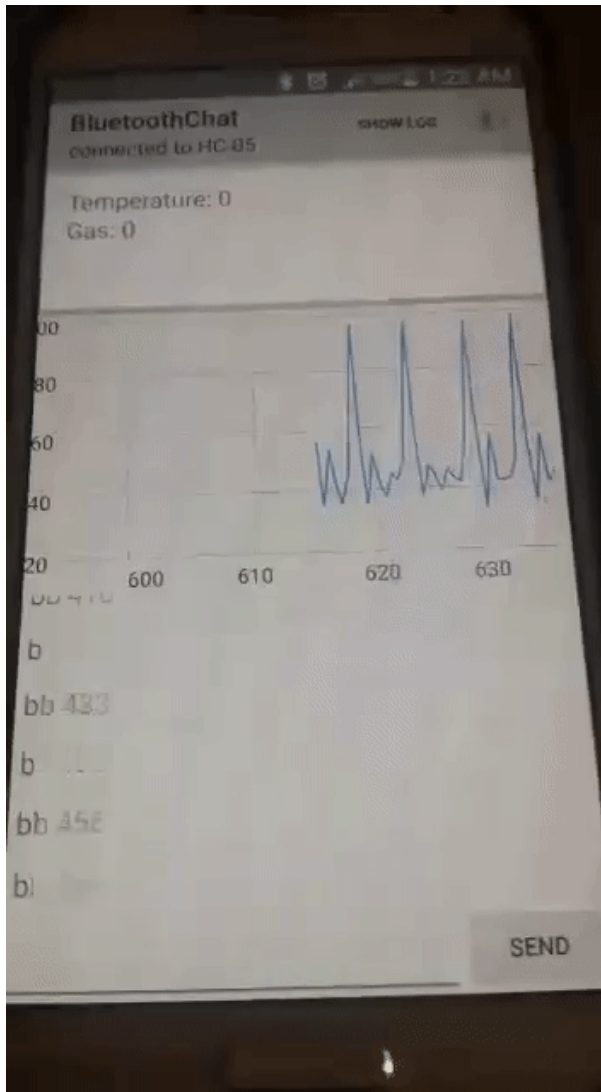
Rezultate Obținute

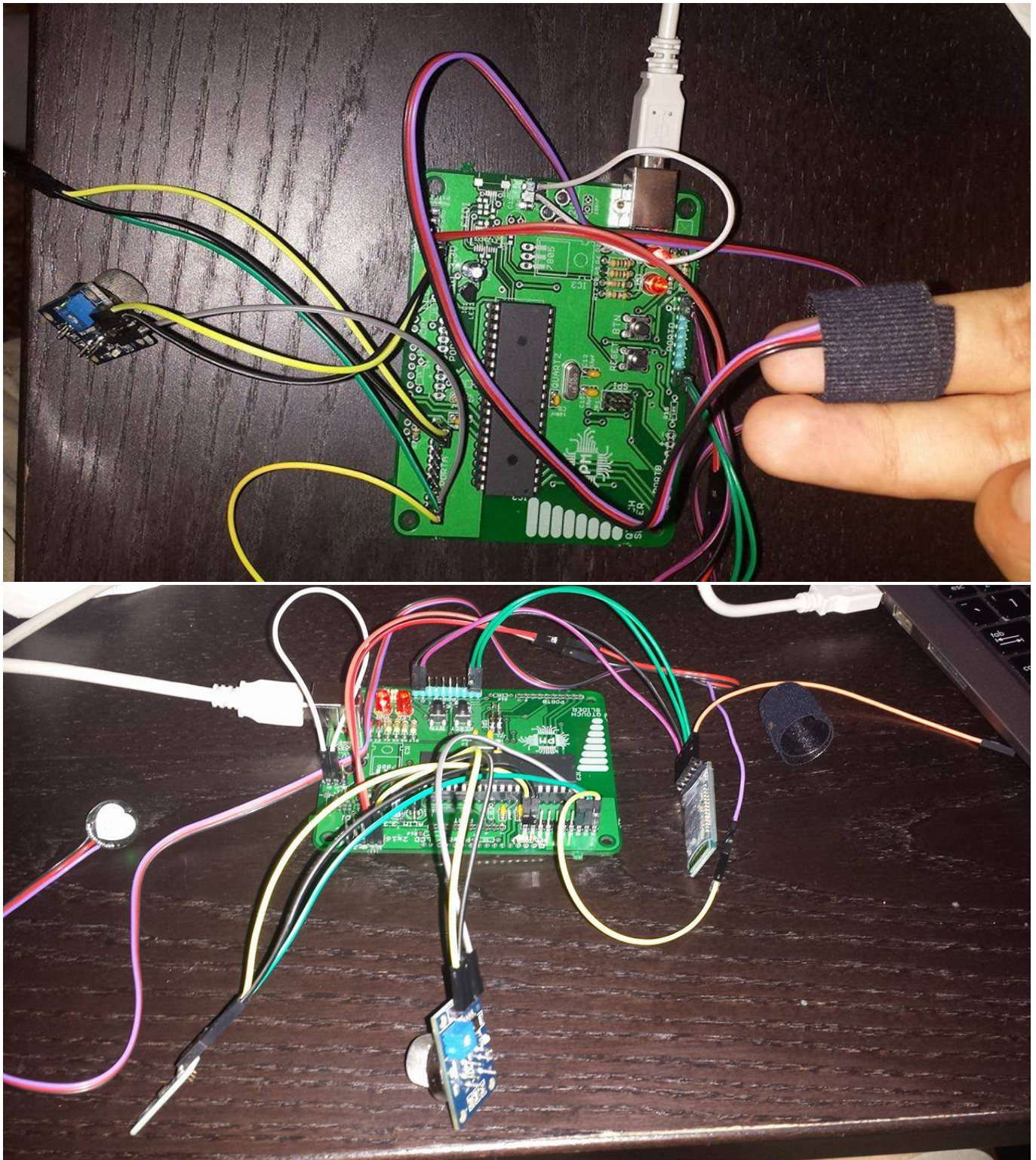
- Proiectul functioneaza, dar, la pornirea placii, senzorul de detectare a calitatii aerului are nevoie de

circa 15-20 minute ca sa se proceseze date utile(practic, pana se incalzeste). De asemenea, senzorul de puls este foarte sensibil, trebuie sa tin mana perfect nemiscata ca sa obtin exact forma de unda a pulsului (care apare si in spitale pe monitoare). Mai mult inca, trebuie sa tin o bandana neagra peste sensor, ca sa umbreasca "terenul" din jurul degetului (el e optic, la baza).

- Pe aplicatia Android, am posibilitatea de a ma conecta la un device folosind Bluetooth (si cerand permisiuni utilizatorului pentru asta). Dupa aceea, graficul se va adapta in real-time in functie de datele primite, precum si datele de la ceilalti 2 senzori (temperatura + calitatea aerului). De asemenea, afisez si payload-ul pe care-l primesc, din motive de debug.

Click on first image(only pulse wave - air quality and temperature disabled):





Concluzii

A fost un proiect interesant, din care am invatat lucruri multiple. Cel mai mult mi-a luat sa-mi dau seama, spre exemplu, ca daca receptionez eu date gresite, nu e necesar sa fac eu o greseala in software. Conform principiului *GARBAGE IN ~ GARBAGE OUT*, senzorul poate fi vinovat de datele pe care le vizualizezi (in cazul meu, trebuia sa tin senzorul foarte fix si sa pun o folie de vinilin peste suprafata lui, pentru ca particulele de transpiratie - in mod surprinzator, daca strangi cu manseta degetul mai tare, transpiri - sa nu perturbe semnalul). Personal, sunt interesant de zona IoT si

in mod particular de aplicatii in lumea medicala, iar acest proiect a fost un "good starting point". Sper sa continui mai departe cu genul acesta de activitati si sa avansez cat mai mult :).

Download

Aplicatia Android ->[bluetoothchat.zip](#)

Sursele (C) pentru uC -> [shs_tstoican.zip](#)

Jurnal

One does not simply finish a project...

- 16 mai: Am instalat cu succes modulul de Wi-Fi ESP8266. Aparent, e usor de programat (poate avea atat functionalitate de client, cat si functionalitate de server). Suporta comenzi AT si a costat putin. L-am configurat in modul client si configurat a ramas.... :). Nici cu pinul de reset activat, nici cu rulare de scripturi in Python pentru a-l configura(apare ca un fisier in /dev pe Linux) ca Access Point nu a mai mers. Din pacate, apar pe Internet informatii conform carora se mai pot bloca (sunt chinezesti, e de inteles).
- 18 mai: Migram pe Bluetooth: Tocmai au adus module noi pe stoc baietii de la Optimus Digital(e ok doctorul, poate sa vada si pe telefon, nu doar pe un monitor oficial intre saloane). HC-05 ftw !!
- Tot 18 mai: HC-05 e usor programabil, suporta tot comenzi AT, dar e mult mai reliable (se incinta usor, but no panic). Avantaj: Ce pui pe UART de pe uC, ajunge direct in telefon. Ca sa-l testez, lucrez cu o aplicatie terminal de Android (luata de pe Play Store).
- 19-20mai: MAX30125 → li invat deja numele pe de rost la cata documentatie caut de 1-2 zile. E de la Sparkfun, au biblioteci pentru Arduino. Iau si eu una sa o "de-portez" din core-ul de Arduino. Schimb driver-ul de I2C cu unul al lui Peter Fleury.
- 20mai: Dupa mai bine de o zi, renunt. E dead end. Desi am fixat clock-ul pt I2C cum e in datasheet, adresa slave-ului s.a.m.d., nu porneste. Nici macar un LED de pe senzor nu pot sa pornesc, din cate se pare. Probabil e de vina driver-ul.
- 21mai: Weekend deja, nu mai am sanse, nu pot face rost nici de osciloscop. Cumpar alt sensor. Merge cu ADC-ul. S-a dus bursa pe luna asta :)
- 22mai: Citesc de pe sensor, folosesc un algoritm de procesare de semnal pentru Arduino. Calculez BPM, IBI si iau si forma semnalului de puls. Inca nu pot sa testez, am nevoie de un grafic.
- 22mai: Lucrez la o aplicatie Android ca sa interfatez lucrurile. Aparent merge ok, am gasit si o biblioteca pentru grafice real-time, GraphView.
- 23mai: Dupa mai multe incercari de calibrare a senzorului, reusesc sa obtin pulse wave-ul pe grafic. Senzorul e foarte sensibil aparent, trebuie sa-l tii intr-o anumita pozitie.
- 24mai: Iau si datele de la ceilalti 2 senzori si ii incorporez in aplicatie. E mult mai usor, gasesc biblioteci mai usor.
- 25mai: Last review....

Bibliografie/Resurse

- <https://www.sparkfun.com/products/11574>
- <https://www.draw.io/> → pentru schema bloc
- <http://www.android-graphview.org/>
- Laboratoarele de PM: <http://cs.curs.pub.ro/wiki/pm>
- WinAVR: <https://sourceforge.net/projects/winavr/files/>
- Biblioteci Eagle open source pentru senzori
- Baietilor de la Google pentru documentatie Android :) → <https://developer.android.com/guide/topics/connectivity/bluetooth.html>
- <https://github.com/WorldFamousElectronics>
- <https://github.com/empierre/arduino/blob/master/AirQuality-MQ135.ino>
- Multumiri speciale lui Cristi Birsan (cablu FTDI, ajutor si sprijin moral :))

- Documentația în format [PDF](#)

From:

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

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2017/cbirsan/82ujk21nd98ay17y4>



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