

Elektronika digjitale dhe sinjalet

ARDUINO & MATLAB: Heartbeat signal

Detyra:

Te lidhet nje apo me shume senzore ne arduino pllake, sipas mundesise elektrodave ECG sinjalit ose EEG sinjalit. Arduino te lidhet ne PC dhe te mundesohet qe sinjali te paraqitet ne MATLAB ne domenin kohore. Poashtu, aplikacioni duhet te ketë mundesine qe sinjali te ndalet me nje STOP buton (ne GUI) dhe te inqizohet. Pasi te merren X samples, te mundesohet qe te shihet edhe frequency domain i sinjalit.

Studentet:
Kushtrim Pacaj
Kujtim Hyseni
Kujtim Krasniqi

Profesori:
prof.Dr. Zhilbert Tafa

Përmbajtja:

<u>1.Matësi i rrahjeve të zemrës</u>	<u>3</u>
<u>1.a. Përshkrim</u>	<u>3</u>
<u>1.b. Saktësia</u>	<u>5</u>
<u>2. Pajisjet e përdorura</u>	<u>6</u>
<u>2.a. Arduino Nano</u>	<u>6</u>
<u>2.b. MAX3012 Heart Rate Monitor</u>	<u>8</u>
<u>3. Arkitektura e brendshme hardverike e sensorit</u>	<u>9</u>
<u>4. Skema e lidhjes së Arduinos me MAX30102</u>	<u>10</u>
<u>5. MATLAB</u>	<u>12</u>
<u>5a. Komunikimi me Arduino nga Matlab</u>	<u>12</u>
<u>5b. Krijimi i UI të programit</u>	<u>13</u>
<u>5c. Marrja e të dhënave nga Arduino dhe shfaqja e sinjalit në domenin kohor në plot</u>	<u>14</u>
<u>5d. Transformimi i sinjalit në domenin frekuencor dhe shfaqja grafike.</u>	
<u>Llogaritja e rrahjeve të zemrës.</u>	<u>16</u>

Abstrakt

Në këtë punim kemi zhvilluar një pajisje për monitorimin e rrahjeve të zemrës. Pajisja përbëhet nga një mikrokontroller arduino dhe një qark i integruar MAX3012 e cila mundëson monitorimin e rrahjeve të zemrës. Sinjali merret nga Arduino në Matlab përmes portit serik, dhe bëhet shfaqja e sinjalit në domen kohor, si dhe në domen frekuencor. Përfundimisht bëhet llogaritja e frekuencës së sinjalit, si dhe duke përdorur atë gjindet numri i të rrahurave të zemrës në minutë.

1. Matësi i rrahjeve të zemrës

1.b. Përshkrim

Zemra si organ kryesor në trupin e njeriut mundëson qarkullimin e gjakut, e pompon gjakun nëpër trup me një ritëm e cila determinohet nga një grup i qelizave në nyjën sinoatriale. Ky grup e krijon një rrymë e cila mundëson që zemra të kontraktohet. Sipas CVD, sëmundjet e zemrës janë shkaku kryesor i shumë vdekjeve nëpër botë që nga viti 2008 duke zënë 30% të vdekjeve në mbarë botën. Në mënyrë që të kontrollohen sëmundjet e zemrës është shfaqë nevoja që të krijohet një pajisje e cila mundëson monitorimin e saj. Përmes kësaj pajisje matet ritmi i rrahjeve të zemrës, pastaj mund të përcaktohet se nëse pacienti vuan nga ndonjë sëmundje e zemrës. Kryesisht në mjekësi monitorimi i zemrës bëhet përmes Elektro Kardio Diagramit (EKG), e cila e matë aktivitetin elektrik të zemrës. Përmes EKG mund të përcaktohet nëse një person vuan nga sëmundjet e ndryshme të zemrës. Ritmi i zemrës është numri i rrahjeve të zemrës për njësi të kohës e cila shprehet si rrahjet për minutë(BPM). Rrahjet e zemrës mund të jenë të ndryshme për shkak se nevoja e trupit për të thithur oksigjen dhe për të nxjerr dioksidin e karbonit gjatë ushtrimeve ose gjumit ndryshon[1].

Në vitin 1901 Willem Einthoven, një fizikanë holandez, ka zbuluar Elektro Kardio Grafin e parë praktik duke përdorur një galvanometër me tel. Në vitin 1927 Kompania General Electric (GE), zhvilloi versionin e saj të makinës EKG duke përdorur tuba vakumi amplifikator me një llambë dhe pasqyrë që gjurmonin impulset elektrike në film. Duke përdorur të njëjtat parime themelore të atyre makinave të hershme, pajisjet e sotme janë evoluar në sisteme kompakte. EKG është një makinë e lëvizshme e cila ka 12 tuba, si tela të bashkangjitura në elektroda ngjitëse. Këto vendosen në pjesët e caktuara rreth zemrës, në duar dhe këmbë. Këto elektroda i ndiejnë impulset elektrike që vijnë nga drejtime të shumëfishta. EKG nuk krijon elektricitet ajo e përçon dhe e matë aktivitetin elektrik.

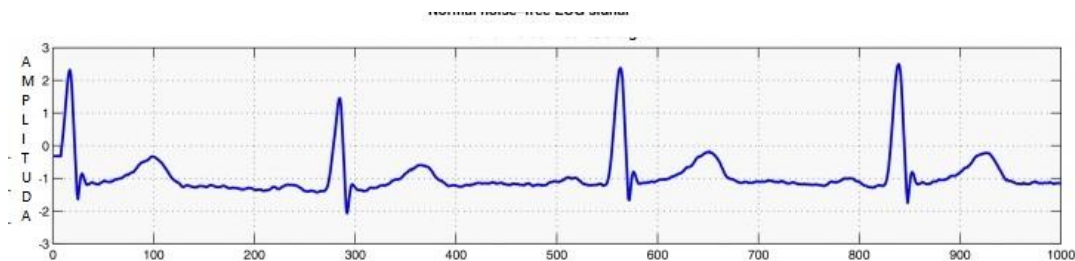


Fig.1. Sinjali i EKG

Përveç përdorimit në mjekësi, pajisja për monitorimin e rrahjeve të zemrës ka përdorim shumë të madhe edhe në sport. Në vitin 1977 Seppo Säynäjäkangas nga kompania Polar në Finlandë, ka zbuluar pajisjen e parë EKG wireless, kompania e tij ishte e popullarizuar në krijimin e pajisjeve për sportist nëpër botë gjatë viteve 1980. Një monitor pa tela i rrahjeve të zemrës me

saktësi EKG, përdor një rrip rreth gjoksit që ka sensorë elektrodë që mund të zbulojnë aktivitetin elektrik të zemrës kur rreh. Më pas i transmeton këto të dhëna tek një marrës i cili përpunon dhe shfaq sinjalin si rrahje në minutë.

Fillimisht, sinjali u dërgua nga valët e radios te marrësi. Teknologjia është zhvilluar për ta dërguar atë përmes ANT ose Bluetooth. Kjo u mundëson pajisjeve të tilla si telefonat celularë dhe pajisjet e tjera mobile të përdorin aplikacione për të marrë të dhëna nga sensorët e monitorit të rrahjeve të zemrës. Marrësi mund të jetë një ekran ore, pajisje që mund të vishet, brez fitnesi, ekran i montuar për një biçikletë, monitrore në shirita vrapues dhe makina të tjera kardio ose aplikacione të telefonit celular.

Shumica e pajisjeve që vishen në treg për momentin kanë brenda tyre monitorë optikë të rrahjeve të zemrës. Ky është një monitor që përdor LED të ndezur për të depërtuar në lëkurë dhe për të zbuluar rrjedhjen e gjakut. Kur drita reflektohet nga rrjedha e gjakut, ajo kapet nga sensorët. Algoritmi pastaj fillon të punojë për të prodhuar të dhëna për rrahjet e zemrës. Por, nuk është aq i saktë sa mund të jetë - veçanërisht kur përfshihen çështje mjekësore.

1.c. Saktësia

Gjurmuesit e aktivitetit që mbani në kyçin e dorës tuaj kanë sensorë photoplethysmography që përdorin dritën për të matur pulsën tuaj. Ato funksionojnë kështu: LED të vegjël, në pjesën e poshtme të tyre shkëlqejnë dritën në lëkurën e kyçit të dorës tuaj. Drita përthihet nga gjaku që rrjedh poshtë lëkurës dhe një tjetër sensor në gjurmues e përkthen atë informacion në një lexim të pulsit. Disa monitorë kanë ekrane të vogla shfaqjeje, por të tjerët duhet të jenë të lidhur me një orë sportive ose smartphone. Modelet themelore tregojnë vetëm rrahjet e zemrës suaj, ndërsa ato më të përparuara ju lejojnë të programoni nivelet e dëshiruara dhe të lëshoni një alarm nëse rrahjet e zemrës suaj shkojnë shumë të larta ose shumë të ulëta. Saktësia e tyre sipas studimit JAMA Cardiology në Janar të vitit 2017, e ka krahasuar një monitor EKG me 4 gjurmues të aktivitetit, ku EKG ka qenë e saktë 99.6%, ndërsa katër gjurmuesit e tjerë kanë qenë të saktë prej 83% deri në 91%, e cila mund të jetë 15 deri në 34 rrahje për minutë dhe varet nga aktiviteti. Për shembull, gjurmuesit e aktivitetit kanë qenë të saktë kur personi ka qenë duke pushuar ose duke vrapuar në shirita vrapues në fitnes, por kanë qenë me pak të saktë kur personi ka qenë duke ngarë biçikletën ose duke përdorur biçikletë eliptike.

Në punimin shkencorë të Babiker, Sharief me titull *“Microcontroller based Heart Rate Monitor using Fingertip Sensors”*, kanë marrë vlerat nga pajisja gjurmuese e aktivitetit që e kanë zhvilluar dhe nga osciloskopi dhe kanë krahasuar gabimet që kanë ndodhur në pajisje matëse. Poshtë tabela.

Gjinia	Mosha	HR në monitor	HR në Osciloskopë	Gabimi %
Mashkull	22	97	96	1.03
Mashkull	22	83	81	2.41

Mashkull	20	78	78	0
Mashkull	22	90	87	3.33
Mashkull	20	80	79	1.25
Femër	22	77	77	0
Femër	22	104	103	0.96
Femër	19	75	75	0
Femër	20	69	71	2.81
Femër	22	83	85	2.35

Tabela 1. Matja e rrahjeve të zemrës duke përdorur pajisjen e zhvilluar dhe përmes një osciloskopi[1]

Kur jemi duke pushuar zemra pompon sasinë më të ulët të gjakut për të furnizuar nevojat e trupit tuaj me oksigjen. Për shumicën e personave të shëndetshëm rrahjet e zemrës gjatë pushimit variojnë nga 60 deri 100 rrahje në minutë. Shpejtësia me të cilën rrah zemra juaj kur punon më shumë për të përmbushur nevojat e oksigjenit të trupit tuaj është rrahja maksimale e zemrës suaj. Shkalla maksimale e rrahjes së zemrës tuaj luan një rol të madh në vendosjen e kapacitetit tuaj aerobik - sasinë e oksigjenit që jeni në gjendje të konsumoni. Në tabelën e mëposhtme mund të shihni vlerat e rrahjes së zemrës për moshë të ndryshme dhe për ushtrime të ndryshme.

Mosha	50%-75% ushtrime mesatare	70%-85% ushtrime të vështira	Maksimumi i rrahjeve të zemrës
20	100-140 bpm	140-170 bpm	200 bpm
30	95-133 bpm	133-161.5 bpm	190 bpm
40	90-126 bpm	126-153 bpm	80 bpm
50	85-119 bpm	119-144.5 bpm	70 bpm
60	80-112 bpm	112-136 bpm	160 bpm
70	75-105 bpm	105-127.5 bpm	150 bpm
80	70-98 bpm	98-119 bpm	140 bpm
90	65-91 bpm	91-110.5 bpm	130 bpm

Tabela 2. Rrahjet e zemrës tek personat në moshë të ndryshme për ushtrime të ndryshme[5].

2. Pajisjet e përdorura

2.a. Arduino Nano

Arduino Nano është një pllakë zhvillimore e bazuar në mikrokontrollerin ATmega329 nga Atmel. Këto pllaka janë jashtëzakonisht të përdorura për shkak të lehtësisë së përdorimit si dhe mbështetjes së madhe që ka nga komuniteti.

Arduino nano ka 14 pina digjital për hyrje/dalje prej të cilëve 5 kanë aftësi për PWM (Pulse Width Modulation). PIN0 dhe PIN1 gjithashtu mund të shërbejnë për komunikim serik pasi që PIN0 është RX dhe PIN1 është TX. Prej PIN10 deri në PIN13 duke e përfshirë edhe këtë të fundit, mund të përdoren për komunikim SPI (MISO, MOSI, CS, CLK). Kjo pllakë gjithashtu ka edhe 8 pin-a analog, ku PIN-i A4 dhe PIN-i A5 mund të përdoren për I2C komunikim. Arduino Nano ka të gjitha funksionet e Arduino UNO përveq hyrjes DC për furnizim, ku si zëvendësim ka USB portin.

Për të përcaktuar nëse një pin i caktuar do të veprojë si hyrje apo dalje përdoret funksioni `pinMode(PIN X, OUTPUT/INPUT)` ku X është numri i pin-it dhe pastaj zgjedh ndonjërin nga OUTPUT ose INPUT. Për të shkruajtur në pin-at digjital apo analog përdoren funksionet `digitalWrite()`, `digitalRead()`, `analogWrite()` dhe `analogRead()`.

Atmega328 ka 32 kB të flash memory e cila përdoret për t'a ruajtur kodin të cilin përdoruesi e shkruan në të. Ndërsa 2 KB të memories janë të rezervuar për bootloader. SRAM i ka 12 KB ndërsa EEPROM i ka gjithsej 1KB. SRAM është vendi ku Arduino i ruan variablat të cilat i përdorë dhe manipulon me to në RUNTIME. Ndërsa EEPROM është hapësira e memories e cila përdoret për t'i ruajtur informatat për kohë më të gjatë. Flash memoria dhe EEPROM nuk janë volatile, duke na dhënë të kuptojmë që të dhënat në këto hapësira memoruese nuk fshihen me ndaljen e mikrokontrollerit.



Fig.2. Pamja e pllakës Arduino Nano

Arduino Nano mund të programohet duke përdorur Arduino IDE i cili mund të shkarkohet nga faqja zyrtare. Arduino IDE është mjaft e kuptueshme dhe e thjeshtë. Përveq që i jep mundësi user-it të shkruaj kod, mundëson edhe shkruajtjen e kodit në flash memorie duke përdorur USB portin e kompjuterit edhe mini USB portin e pllakës. Gjithashtu, kjo IDE mundëson edhe një dritare për komunikim serik me user-in e cila është mjaft e rëndësishme gjatë testimit dhe

debuggimit. Perveq dritares për komunikim serik, kjo IDE e ka të implementuar edhe një dritare për plot-im të të dhënave që dërgohen në portin serik.



Fig.3. Pamja e Arduino IDE dhe shpjegimi i dy ikonave dhe funksionet kryesore

2.b. MAX3012 Heart Rate Monitor

MAX30102 është një modul elektronik që mundëson monitorimin e rrahjeve të zemrës. Përmbanë LED të brendshëm, fotodetektor, elemente tjera optike, një qark elektronik për eliminim të dritës së ambientit, etj.

Komunikimi me këtë modul bëhet përmes I2C protokolit. Pra, ky modul përmbanë edhe konverterin ADC (Analog to Digital Conversion) pasi që I2C protokoli është protokoll digjital i komunikimit. I2C është master/slave komunikim. SDA kanali mundëson shkëmbimin e të dhënave, ndërsa SCL kanali siguron clock-un. Ka disa mode të komunikimit sa i përket shpejtësisë:

- Standard Mode - që kryen transferin e të dhënave me shpejtësi 100 kbps
- Fast Mode - që kryen transferin e të dhënave me shpejtësi 400 kbps (kjo shpejtësi është përdorur në projektin tonë)
- High Speed Mode - që kryen transferin e të dhënave me shpejtësi 3.4 Mbps
- Ultra Fast Mode - që kryen transferin e të dhënave me shpejtësi 5 Mbps

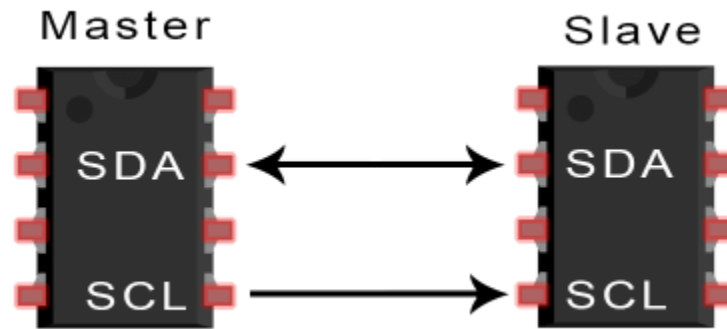


Fig.4. Ilustrim i I2C komunikimit

I2C është një komunikim sinkron, ku outputi i bitave është i sinkronizuar me clock-un që si burim ka kanalin SCL dhe i cili përdoret prej master dhe slave. Clock-u kontrollohet gjithmonë vetëm nga master.

3. Arkitektura e brendshme hardverike e sensorit

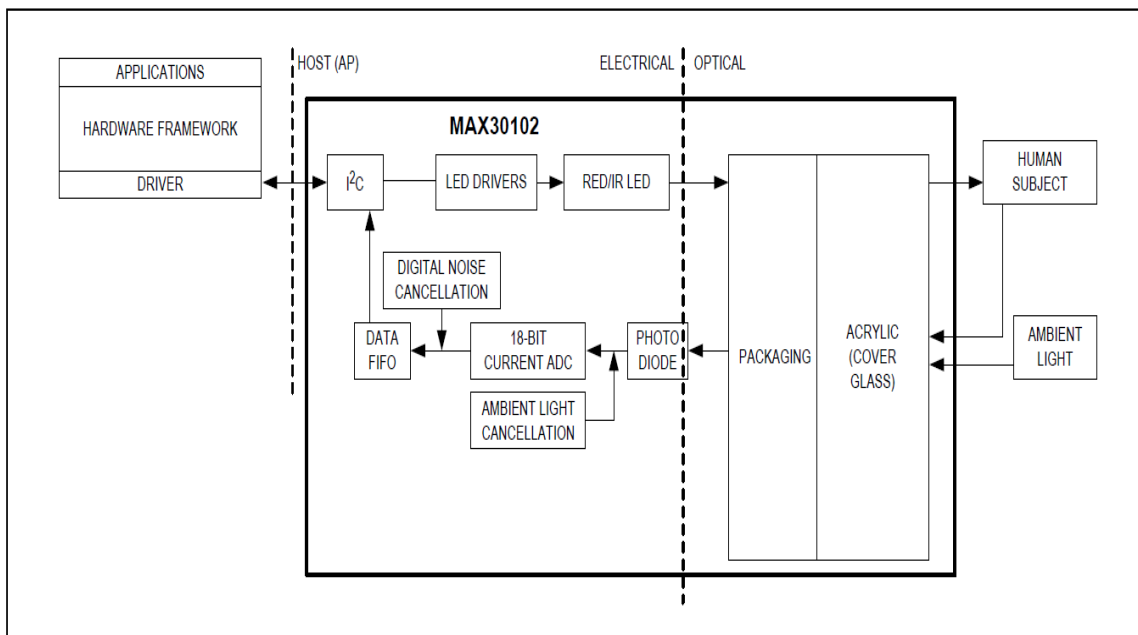


Fig.5. Skema e brendshme e sensorit MAX30102

Siç shihet edhe nga figura e mësipërme, janë tri pjesë kryesore të modulit:

1. Komunikimi me pajisjet periferale
2. Pjesa elektronike që mundëson konvertimin e të dhënave, kontrollimin e LED-ave si dhe anulimin e dritës së ambientit
3. Pjesa e marrjes së të dhënave nga useri apo pjesa optike

4. Skema e lidhjes se Arduinos me MAX30102

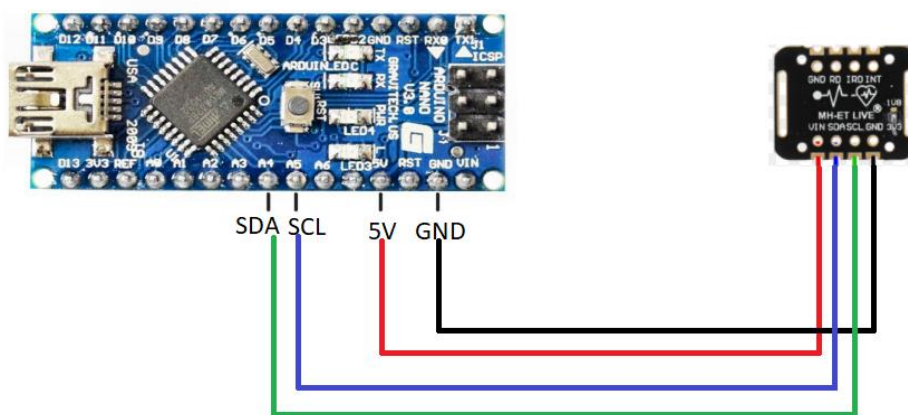


Fig.6. Skema e lidhjes ndërmjet sensorit dhe Arduinos

Në figurën e mësipërme është paraqitur figurativisht lidhja ndërmjet Arduinos dhe sensorit. Mund të shihet që totalisht janë 4 tela të përdorur, 2 për rrymë dhe 2 për komunikim I2C.

Për të lehtësuar komunikimin, është përdorur një librari e gatshme e suportuar nga komuniteti i Arduinos “Wire”. Gjithashtu është përdorur edhe libraria MAX3015 për të kontrolluar sensorin me komandat e ofruara si të gatshme nga kjo librari.

Duhet mbajtur në mend gjatë programimit me Arduino që ekzistojnë dy funksione kryesore të cilat nuk mund të fshihen e të cilat janë: funksioni “**void setup()**” dhe funksioni “**void loop()**”. Përfshirja e librarive dhe deklarimi i variablave globale duhet të bëhet jashtë këtyre funksioneve. Përfshirja e librarive në projekt pas instalimit të tyre bëhet me këto komanda:

```
#include <Wire.h>
#include "MAX30105.h"    //MAX3010x library
#include "heartRate.h"    //Heart rate calculating algorithm
```

Deklarimi i objekteve dhe variablave globale është bërë në këtë mënyrë:

```
MAX30105 particleSensor;

const byte RATE_SIZE = 4; //Increase this for more averaging. 4 is good.
byte rates[RATE_SIZE]; //Array of heart rates
byte rateSpot = 0;
long lastBeat = 0; //Time at which the last beat occurred
float beatsPerMinute;
int beatAvg;
```

Funksioni i parë i cili thirret është funksioni **void setup()**. Ky funksion thirret vetëm në fillim dhe nuk thirret më gjatë ekzekutimit. Këtë funksion e shfrytëzojmë që t'i bëjmë setup komunikimin I2C me sensorin, t'a lëshojmë sensorin në punë si dhe të konfigurojmë komunikimin serik me kompjuter duke përcaktuar edhe baudrate. Në rastin tonë, ky funksion duket kështu:

```
void setup() {

  // Initialize sensor
  particleSensor.begin(Wire, I2C_SPEED_FAST); //Use default I2C port, 400kHz speed
  particleSensor.setup(); //Configure sensor with default settings
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
  Serial.begin(9600);
}
```

Siç mund të shohim nga kodi i mësipërm, kemi përdorur objektin “particleSensor” për të konfiguruar komunikimin me sensorin duke përdorur dy libraritë e sipër përmendur si “Wire” dhe “MAX30105”.

Funksioni i cili thirret më pas është **void loop()**. Ky funksion thirret vazhdimisht derisa të ndalet funksionimi i mikrokontrollerit. Në rastin tonë, ky funksion duket si më poshtë:

```
void loop() {

  long irValue = particleSensor.getIR(); //Reading the IR value it will permit us to know if there's a finger on the sensor or not

  irValue = max(irValue,105000);
  irValue = min(irValue,120000);
  irValue = irValue - 105000;

  Serial.println(irValue);
}
```

Siç mund të shihet edhe nga kodi i mësipërm, getIR() funksioni i cili vjen nga libraria MAX30102 na ka mundësuar t'a lexojmë vlerën IR të sensorit. Pas disa testeve, kemi vërejtur që vlerat po sillen nga 105000 deri në 120000 atëherë kur gishti i përdoruesit gjendet mbi sensor, i kemi përdorur funksionet max() dhe min() për të përcaktuar kufirin e poshtëm dhe të epërm të kësaj variable. Për t'a rritur rezolucionin e plotimit dhe për ti zvogeluar kufijte gjatë plotimit, atë

vlerë e kemi zbritur me 105000 në mënyrë që vlerat nën 105000 të vijne si zero, dhe të tjerat të plotohen me rezolucion të lartë duke mundësuar të shohim më shume e me qartë gjatë plotimit.

5. MATLAB

5a. Komunikimi me Arduino nga Matlab

Komunikimi ndërmjet Matlab dhe Arduino bëhet përmes portit serik. Pasi që një kompjuter mund të ketë më shumë se një port serik, atëherë fillimisht duhet t'i listojmë të gjitha portet, dhe më pas ta filtrojmë atë listë, dhe ta zgjedhim portin e dëshiruar. Ky funksionalitet në programin tonë është arritur përmes kodit vijues:

```
ports = serialportlist("available");
usbSerialPort = ports(contains(ports, "/dev/cu.usbserial"));

if(usbSerialPort.length == 0)
    disp("No serial port connected");
    return;
else
    disp("Serial port: " + usbSerialPort(1))
end

disp("Setting up serial port connection");
```

Pasi që të bëhet zgjedhja e portit, bëhet krijimi i lidhjes me të. Si argument extra është edhe shpejtësia e komunikimit, apo baud rate, që në këtë rast e kemi 9600. Poashtu bëhet definimi i Data vargut dhe skalarit Count për ti mbajtur të dhënat e lexuara.

```
arduinoObj = serialport(usbSerialPort(1),9600);
flush(arduinoObj);
arduinoObj.UserData = struct("Data",[],"Count",1);
```

5b. Krijimi i UI të programit

Sipas specifikacioneve të detyrës, duhet që programi të shfaqë këto gjëra:

1. Graf të sinjalit të marrë nga sensori (time-domain)
2. STOP buton që do e ndalë programin
3. Graf te domenit frekuencor të sinjalit (frequency-domain)

Fillimisht duhet që ta krijojmë dritaren e aplikacionit (*uifigure*), dhe kjo bëhet përmes:

```
% Krijo UI te programit
figure = uifigure('Position',[ 500 500 600 600]);
movegui(figure,'center');
```

Më pas, e krijojmë *axis*, apo sistemin koordinativ , brenda te cilit do bëjmë më vonë grafikimin e sinjalit.

```
% Krijo 'axis' - Sistemin koordinativ ku do shfaqim te dhenat
plotAxis = uiaxes('Parent',figure, 'Position', [50, 50, 500, 500]);
```

Së fundi, e krijojmë *STOP* butonin përmes kodit vijues. Ne klikim te butonit, e bëjmë variablën globale *stopNow* në true. Këtë variabël do e përdorim më vonë.

```
% Krijome nje button
stopBtn = uibutton(figure,'push',...
    'Text', 'Stop',...
    'Position',[100, 550, 100, 25],...
    'ButtonPushedFcn', @stopButtonPushed);

function stopButtonPushed(~,~)
    global stopNow
    stopNow = true;
end
```

5c. Marrja e të dhënave nga Arduino dhe shfaqja e sinjalit në domenin kohor në plot

Pasi ta kemi krijuar lidhjen me Arduino, përdorim funksionin *readline*, te *serialobject* në mënyrë që ta lexojmë të dhënën e radhës.

```
tic
while stopNow == false
    data = str2double(strtrim(arduinoObj.readline));

    arduinoObj.UserData.Data(end+1) = data;
    arduinoObj.UserData.Count = arduinoObj.UserData.Count + 1;
    plot(plotAxis,arduinoObj.UserData.Data(2:end));

end
timeElapsed = toc;
```

Të dhënën e marre e ruajmë në vargun *Data* dhe re ri-vizatojmë atë varg. Kjo loop do ndalet në momentin që *stopNow* bëhet *false* (dmth, kur të klikohet butoni *STOP*).

Me *tic* dhe *toc* matim kohën për të cilën marrim dhe ruajmë sinjal. Duke ditur këtë, si dhe numrin e samples të marra, mund ta llogaritim *sampleRate* mesatar të sinjalit.

Më poshtë është një me një sinjal nga testimi që kemi bërë:

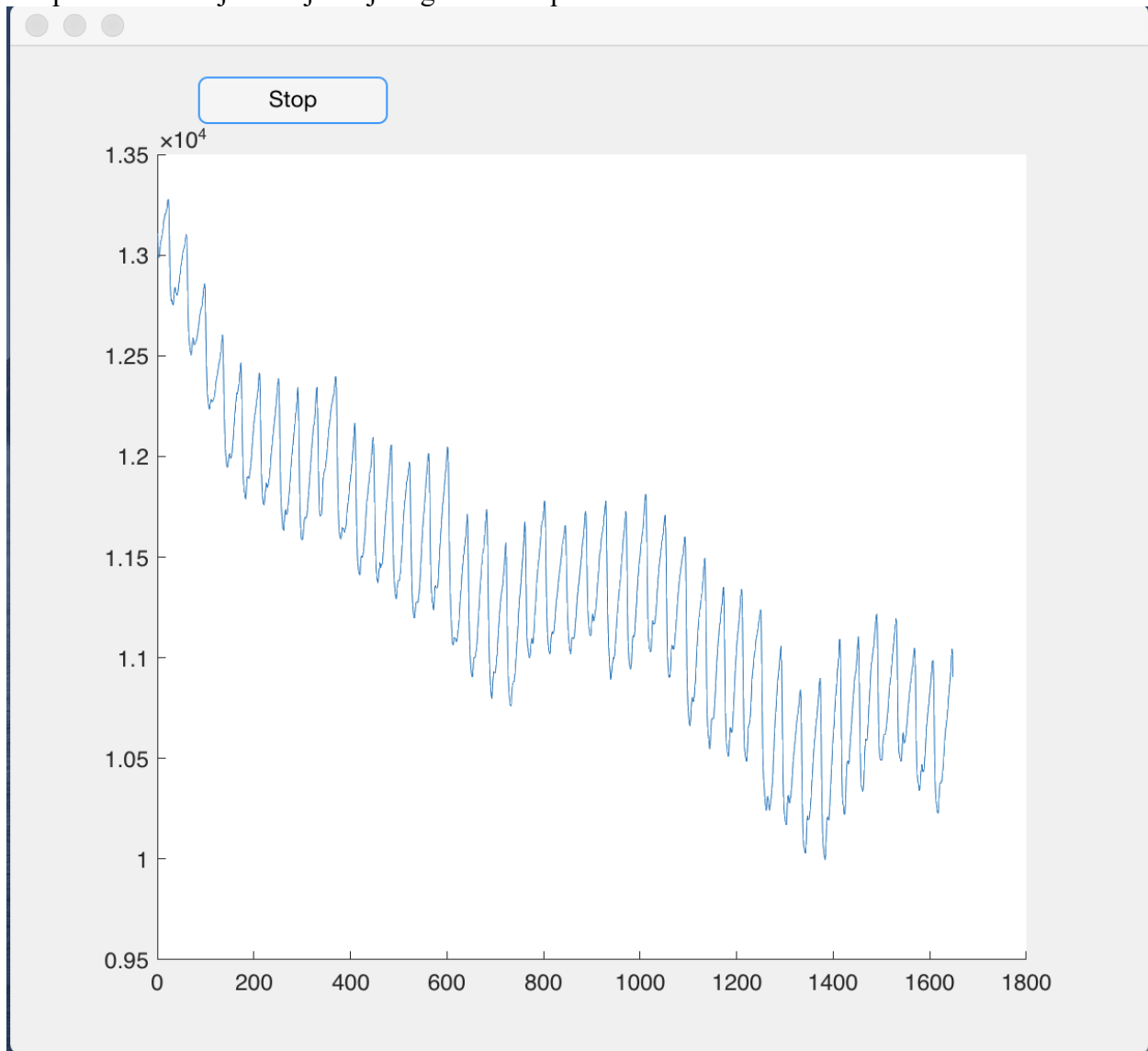


Fig.7. Sinjali i heartbeat në domenin kohor

Përfundimisht, para se ta llogaritim *frequency domain*, do e ndreqim pak sinjalin, që të jetë me i mirë për procesim. Do e bëjmë shift sinjalin që të jë rreth $Y=0$. Këtë e bëjmë duke llogaritur mesataren, dhe duke zbritur Y komponentën për aq. Dmth nëse sinjali do ishte me amplitude minimale 10k dhe maksimale 15k, atëherë mesatarja do ishte 12.5k. Pas zbritjes për 12.5k, sinjali do ishte me amplitude minimale -2.5k, dhe maksimale 2.5k.

```
%llogarit mesataren e amplitudave dhe zbrite gjithë sinjalin per aq  
shiftedTD = timeDomainData - mean(timeDomainData);
```

```
% thirr funksionin për paraqitje të domenit frekuencor.  
plotFrequencyDomain(shiftedTD,timeElapsed);
```

5d. Transformimi i sinjalit në domenin frekuencor dhe shfaqja grafike.

Llogaritja e rrahjeve të zemrës.

Më tutje do shpjegojmë implementimin funksionit për plotim të domenit frekuencor të thirrur më lartë. Si argumente të tij kemi sinjalin në domen kohor, si dhe kohën për të cilën është marrë ai sinjal.

Si *sampleCount* të sinjalit kemi numrin e të dhënave të lexuara nga Aurdiono. Dukë përdorur edhe kohën për të cilën është marrë sinjali, e llogarisim *sampleRate* të sinjalit.

Ky sample rate mund të varet nga shumë faktorë, por në rastin tonë më së shumti varet nga sensori i zgjedhur. Në testimet tona, vlera e tij sillet rreth **48**, dmth aq herë në sekond ka dhënë vlerë.

```
function plotFrequencyDomain(timeDomainData,timeElapsed)  
close all;  
  
sampleCount = length(timeDomainData);  
  
samplingFreq = sampleCount/timeElapsed; % samples per second  
  
%heq frekuencat e uleta ( nën 0.8Hz ) nga sinjali  
timeDomainData = highpass(timeDomainData,0.8,samplingFreq);  
  
% .... Funksioni vazhdon në poshtë në copeza tjera të kodit.
```

Më tutje kemi zgjedhur që sinjalin ta kalojmë përmes një highpass filteri, që i eliminon frekuencat e vogla (nën 0.8Hz).

Në raste ideale të testimit nuk do kishim nevojë për këtë filter, po fatkeqësisht është gati e pamundshme të marrim sinjal ideal gjatë testimit, sepse ose e levizim gishtin pak, ose e ndryshojmë nivelin e shtypjes në sensor me gisht.

Kjo na e shton njëfarë zhurme me frekuencë të ulët. *Highpass* filter i aplikuar i heq zhurmat duke anuluar sinjalin me frekuencë më të vogël se 0.8Hz.

Për ta llogaritur grafën e sinjalit në domen frekuencor, përdorim FFT:

```
X = fftshift(fft(timeDomainData));  
dF = samplingFreq/sampleCount;  
f = -samplingFreq/2:dF:samplingFreq/2-dF;
```

```
%shfaq vetem Hz pozitive. Pjesa negative eshte simetrike me kete
f = f(floor(length(f)/2):end);
X = X(floor(length(X)/2):end);
```

Kurse plotimi bëhet si më poshte:

```
figure;
plot(f,abs(X)); % ploto vlerat reale ( jo imagjinare) te transformimit Furie
xlabel('Frekuenca (ne Hz)');
xticks(0:1:length(f)); % ne legjende poshte, shfaq numrat cdo 1Hz
title('Magnituda');
xlim([0, f(end)]); % starto x prej 0 ne plot
```

Duke vazhduar shembullin e marrë me lartë, e gjemë domenin kohor të sinjalit të shfaqur në foton e mëparshme.

Siç shihet nga fotoja e mëposhtme, frekuenca mbizotëruese në sinjalin e marrë është 1.22Hz.

Këtë vlerë e gjejmë me kod kështu:

```
[maxValue,indexOfMax] = max(abs(X));
maxFreq = abs(f(indexOfMax));
disp('Max freq ' + maxFreq + " Hz");

text(maxFreq,maxValue,"<-- " + maxFreq + "Hz");
```

Dmth, gjejmë vlerën maksimale tek koordinatat Y, dhe marrim më pas vlerën në X se sa është, vlerë e cila na jep frekuencën.

Pasi sinjali i matur është shprehur në frekuencë Hz, që i bie kemi 1.22 rrahje të zemrës për një sekondë. Nëse këtë vlerë e shumëzojmë me 60, atëherë e fitojmë numrin e rrahjeve për minutë, që është zakonisht se si shprehet numri i rrahjeve të zemrës.

Në shembullin tim, vlerat e marra dhe të llogaritura janë:

```
timeElapsed = 33.3711
sampleCount = 1648
samplingFreq = 49.3840
-----
Max freq 1.2286 Hz
bpm: 73.7165
```

DMTH, sinjali në këtë rast na tregon se zemra ka rrahur rreth 73 herë për minutë.

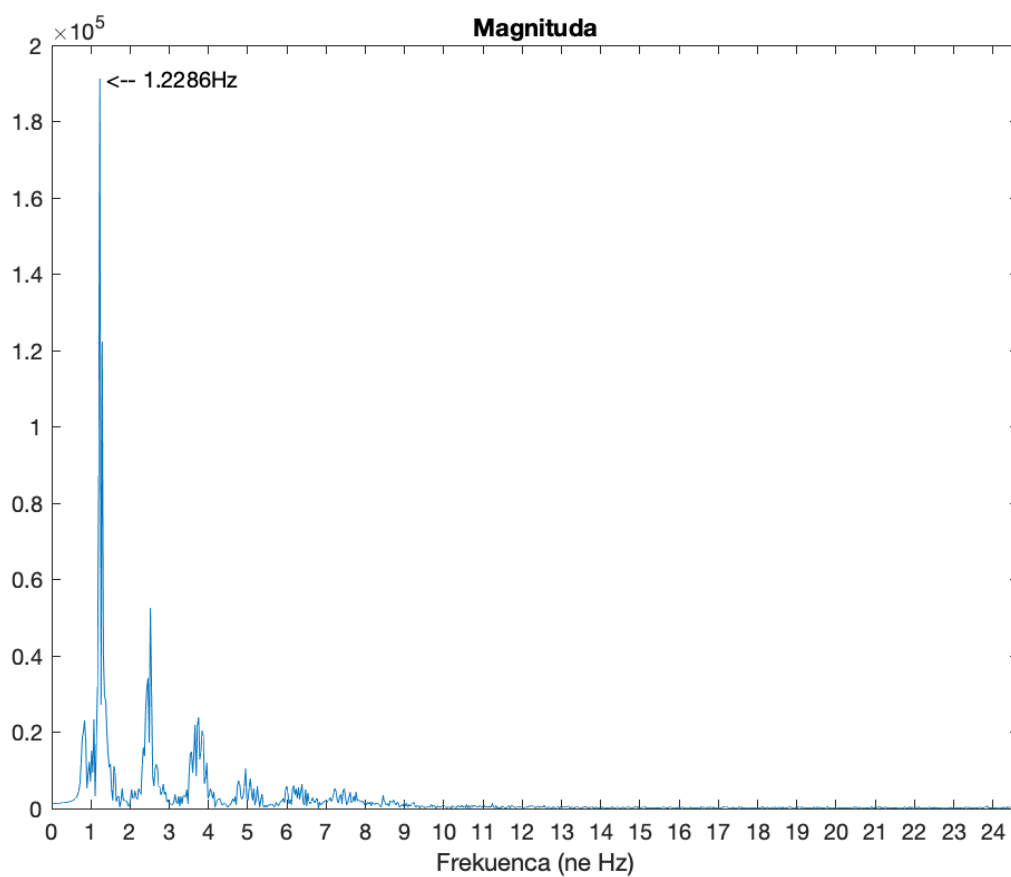


Fig.8. Sinjali i heartbeat në domenin frekuencor

Referencat

[1]. Babiker, Sharief & Abdel-Khair, Liena. (2012). Microcontroller based Heart Rate Monitor using Fingertip Sensors. UofK Engineering Journal.

[2]. <https://www.healthline.com/health/abnormal-ekg#ekg-process>

[3]. <https://www.wearable.com/health-and-wellbeing/ecg-heart-rate-monitor-watch-guide-6508>

[4]. <https://www.verywellfit.com/heart-rate-monitor-3436583>

[5]. <https://www.yourobservers.com/article/understanding-heart-rate-and-health>

Tabela e figurave

Fig.1. Sinjali i EKG

Fig.2. Pamja e pllakës Arduino Nano

Fig.3. Pamja e Arduino IDE dhe shpjegimi i dy ikonave dhe funksionet kryesore

Fig.4. Ilustrim i I2C komunikimit

Fig.5. Skema e brendshme e sensorit MAX30102

Fig.6. Skema e lidhjes ndërmjet sensorit dhe Arduinos

Fig.7. Sinjali i heartbeat në domenin kohor

Fig.8. Sinjali i heartbeat në domenin frekuencor

Tabela 1. Matja e rrahjeve të zemrës duke përdorur pajisjen e zhvilluar dhe përmes një osciloskopi[1]

Tabela 2. Rrahjet e zemrës tek personat në mosha të ndryshme për ushtrime të ndryshme[5].