

Arbeidskrav 1

David Amirtahmasebi

Oppkobling: Oppkoblingen kan ses på fritzing dokumentet.

Healthy – padene inkluderte jeg ikke i fritzing, dersom jeg antar at det ikke er nødvendig. På begynnelsen av oppgaven fikk jeg problemer med ujevne data verdier og i seriell plotter var den ikke nøyaktig. Men der var fordi jeg koblet ikke healthy padene nøyaktig på kroppen. Jeg benyttet Particle photonen og AD8232 heart rate sensoren i oppgaven.

Kodingen: Jeg fant et par eksempler på nett og brukte det som et utgangspunkt først, men lagde et helt nytt kode struktur etter hvert for å virke til google cloud. Målet var å få data gjennom koden å regne meg fram til den riktige BPMen. Jeg fant ut etter hvert at dataen jeg fikk fra arduino og particle photon var helt annerledes. Particle photon skrev ut data på rundt 2000, mens arduinoen ga en grei mengde rundt 400. Jeg skjønnte etter hvert at dataen var ikke viktig til å regne ut BPM men for å skjønne når hjertet mitt banket i et vis «threshold».

Et problem med å jobbe med photonen var at uten delay ville ikke den skrive ut data. Jeg måtte benytte delay på rundt 400 for å få den til å fungere.

Denne dataen klarte jeg å printe ut som events på particle photonen:

David BPM	75	26002b000447363332363639	10/4/18 at 11:32:33 pm
David BPM	149	26002b000447363332363639	10/4/18 at 11:32:28 pm
David BPM	97	26002b000447363332363639	10/4/18 at 11:32:22 pm
David BPM	97	26002b000447363332363639	10/4/18 at 11:32:16 pm
David BPM	91	26002b000447363332363639	10/4/18 at 11:32:09 pm
David BPM	71	26002b000447363332363639	10/4/18 at 11:32:03 pm
David BPM	79	26002b000447363332363639	10/4/18 at 11:31:56 pm

Jeg benytter millis for å finne forskjellen mellom første puls og andre puls. Dette settes inn i et parameter kalt PulseInterval som skal brukes til å regne ut BPMen. Particle.publish og Particle.subscribe brukes for å kommunisere videre til google cloud.

Bruken av Google cloud og IFFT til Google Sheets:

Når jeg skulle velge et sted å lagre dataen tenkte jeg først på google cloud. Jeg satt opp en bruker og klarte alle stegene bortsett fra de siste. Så jeg hadde ingen måte å vise dataene på med et mer tilfredsstillende visuell måte. Dataene ble fremdeles mottatt fra google cloud som du kan se her. Med bruken av Particle.publish og Particle.subscribe. Men det er ikke en data som kan analyseres behagelig.

David BPM monitor	145	26002b000447363332363639	10/4/18 at 10:55:53 pm
hook-response/David BPM monitor/0	211657422356122	particle-internal	10/4/18 at 10:55:48 pm
hook-sent/David BPM monitor		particle-internal	10/4/18 at 10:55:48 pm

🟢 October 4th, 2018 10:56:12.705 PM

Event

The source event that triggered the webhook


```
{
  "name": "David BPM monitor",
  "data": "78",
  "ttl": 60,
  "published_at": "2018-10-04T20:56:12.395Z",
  "coreid": "26002b000447363332363639"
}
```

Siden Google Cloud ikke visuelt viste data som man hadde muligheten til å lese benyttet jeg heller IFFTT å lagde en applet. Der fant jeg «Log Particle Events to One Spreadsheet» appleten som lagret dataene jeg fikk fra photonen direkte på google sheets.


IFTTT
My Applets
Activity
Search
davam27

Applets

Services


Log Particle Events to One Spreadsheet

On
works with


Get an email with the latest IFTTT updates

On
works with

Denne metoden tilfredstilte arbeidskravet. Slikt at jeg endte opp med å bruke denne. Dataene kan bli analysert videre av det medisinske universitetet.

October 4, 2018	BPM	DavidMess	75		
October 4, 2018	BPM	DavidMess	77		
October 4, 2018	BPM	DavidMess	72		
October 4, 2018	BPM	DavidMess	90		
October 4, 2018	BPM	DavidMess	86		
October 4, 2018	BPM	DavidMess	76		
October 4, 2018	BPM	DavidMess	157		
October 4, 2018	BPM	DavidMess	92		
October 4, 2018	BPM	DavidMess	72		

Løsningen selv og Hvordan å lage løsningen for 10 personer:

Løsningen fungerer godt for å analysere en person sin jevn puls, men burde justeres slik at aktiv bevegelse ikke lager for mye støy. Hvis jeg holder rolig vil jeg få riktig BPM data. Jeg tenker at hvis jeg endret mye på koden ville jeg fått en mer riktig nøyaktig estimering. Denne erfaringen tar jeg med meg videre.

Hvis jeg skulle lage løsningen for 10 personer, kunne jeg brukt IFFFT applet metoden, og laget 10 applets til sammen for å produsere 10 google sheets dokumentasjoner som kan bli analysert av universitetet. Appleten kan også programmeres til å lage flere sheets utifra hvor mye data som er lagret, med dette kan vi lagre data for de forskjellige dagene pasientenes puls blir overvåket.

Istedenfor å printe ut «DavidMess» som er navnet jeg ga min photon, kunne jeg heller endret device navnet på particle appen og navngit de utifra hva pasientene heter eller deres ID.

Konklusjon og lærdom:

Med å bruke Google Cloud og IFFFT appleten for å lagre det til Google sheet bygde jeg utrolig mye erfaring. Jeg hadde også tanker om å benytte azure og firebase for å profesjonelt vise data. Men på grunn av mangel på tid, brukte ikke tid til å researche å prøve meg frem.

Oppgaven var gøy på grunn av at vi gjorde det på nett. Men det var veldig frustrerende også å jobbe med sensoren fordi paddene falt av overtid fordi de ble for slitt, og particle photonen sluttet å fungere tilfeldig mens jeg satt sammen oppgaven. Jeg lærte mye og vil si at jeg har skjønnet konseptet bak å laste opp data til en database er klar for arbeidskrav 2.