

Eksamensoppgave Maskin til maskin kommunikasjon (TEK301) Høsten 2018

Kandidatnr: 100018

Om dokumentasjonen

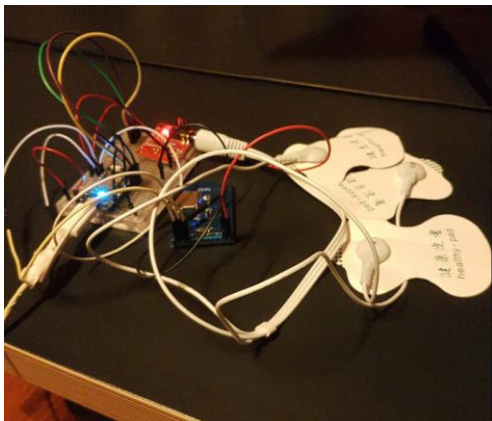
Gjennom dokumentet skal jeg beskrive løsningen, hvordan den settes opp og hvordan den fungerer i detaljer. Jeg skal også forklare hvorfor jeg valgte løsningen google sheets. Video, Fritzing oppkobling og Particle Photon kode inkluderes i zip filen. Jeg logget ikke 30 minutter av data, istedenfor lagret jeg et par sekunder av data for å demonstrere hvordan jeg kom fram til løsningen til oppgaven.

Enheter brukt i løsning

Gjennom hjemme eksamens perioden valgte jeg å benytte AD8232 Heart monitor, denne brukte jeg gjennom arbeidskrav 1 helt fint men den har blitt slit, så når jeg demonstrerte løsningen i Video så koblet jeg den ikke mot kroppen fordi den av og til ikke tok opp min BPM og den var litt upresis her og der. Men når du kobler opp antar jeg at den fungerer som den skal.

Mikrokontrolleren jeg brukte er Particle Photon og akselerometeret som ble benyttet er ST7735 1.8 TFT Skjermen.

Bilde av sluttprodukt oppkobling:



Koden

Gjennom koden brukte jeg biblioteket math.h, TFT akselerometer koden var koden vi benyttet fra tidligere forelesninger og jeg refererte til en tutorial som lærte meg hvordan å sette opp google sheets med webhook.

Jeg lenker disse her:

<https://www.hackster.io/gusgonnet/pushing-data-to-google-docs-02f9c4>

<http://bildr.org/2011/04/sensing-orientation-with-the-adxl335-arduino/>

Koden finner du som en notepad lagt til i zip – filen.

Sette opp løsningen

Jeg forklarer med en kort sammendrag hvordan jeg satt det opp.

Som sagt brukte jeg denne tutorialen for å komme fram til svaret:

<https://www.hackster.io/gusgonnet/pushing-data-to-google-docs-02f9c4>


Det første jeg måtte gjøre var å sette opp en webhook kalt googleDocs som event navn. Det er en URL som blir kjørt gjennom når Particle Publish sender name, bpm, og xyz data. Denne URLen er en script som legger til en ny rad i vår google sheets med informasjonen vi fikk nap av.

googleDocs	{"my_name":"David", "bpm_value": "107", "x":... DavidAmir	11/10/18 at 9:00:39 pm
hook-response/googleDocs/O	{"result":"success","row":52}	particle-internal
		11/10/18 at 9:00:33 pm

Dette fanger mikrokontrolleren og printer eventen. Vi kan også se her at vi fikk ny data i vår google sheet:

11/10/2018 21:00:40	David	107	225	98.899394	351.100606
---------------------	-------	-----	-----	-----------	------------

Merk at tiden som printes i forhold til nå tid er ikke alltid 100% presist. Her kan vi også se Webhook instillingene:

	Event: googleDocs	Target: google.com
ID: 5be4477052e0635aa3229aba	Created: November 8th, 2018	

INTEGRATION INFO

Event Name <i>The Particle event name that triggers the webhook</i>	googleDocs
Full URL <i>The target endpoint that is hit when the webhook is triggered</i>	https://script.google.com/macros/s/AKfycbwfditU5TeO5DVZNRvMveFB5_6mosgkMSJ6AgBNJNb5Z-zFIQA/exec
Request Type <i>The standard web request method used when the webhook is triggered</i>	POST
Request Format <i>How the webhook data will be encoded and passed to the target endpoint</i>	Web Form
Device <i>The device that will trigger the webhook</i>	DavidAmir
Form <i>Form data that will be sent along with the webhook</i>	<pre>{ "name": "{{my_name}}", "BPM": "{{bpm_value}}", "x": "{{x}}", "y": "{{y}}", "z": "{{z}}" }</pre>
Headers <i>HTTP Headers to include when hitting the webhook endpoint</i>	<pre>{ "Content-Type": "application/x-www-form-urlencoded" }</pre>
Enforce SSL <i>Whether your webhook will validate the certificate against its certificate authority chain</i>	Yes

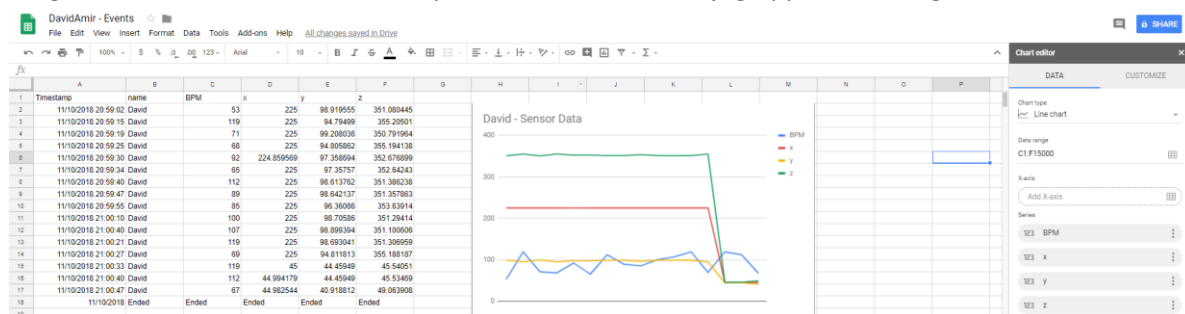
Dataene sendes gjennom en particle publish linje:

```
//Publish the data to the google sheets when everything is calculated
Particle.publish("googleDocs", "{ \"my_name\": \"\" + name + "\", \"bpm_value\": \"\" + String(BPM)
+ "\", \"x\": \"\" + String(x) + "\", \"y\": \"\" + String(y) + "\", \"z\": \"\" + String(z) + "\" }", 60, PRIVATE);
//Show 30 minuter timer progress.
Particle.publish("endingCheck", String(endingCheck), PRIVATE);
```

Siden tanken var at hver episode varte opp til 30 minutter la jeg til en endingCheck int variabel som bruker millis til å tele tiden. Når 30 minutter har gått (1.8millioner millisekunder) vil det ikke printes mer sensor data til google sheeten. Vi vet når det er ferdig når slutten av filen har «Ended» printet flere ganger:

11/10/2018 21:00:40	David	112	44.994179	44.45949	45.53469
11/10/2018 21:00:47	David	67	44.982544	40.918812	49.063908
11/10/2018	Ended	Ended	Ended	Ended	Ended

Det er en del ting som må settes opp manuelt. Kolonne navn, grafen og antall rader google sheeten tar imot. Grafen setter jeg opp som C1:F15000 som betyr at den vil lese 15000 rader ned og legge dette inn. Dette gjør at vi har en «live chart» som oppdateres gjennom prosessen der dataene blir fanget. Kolonne navnene Timestamp, name BPM o.s.v satt jeg opp manuelt også.



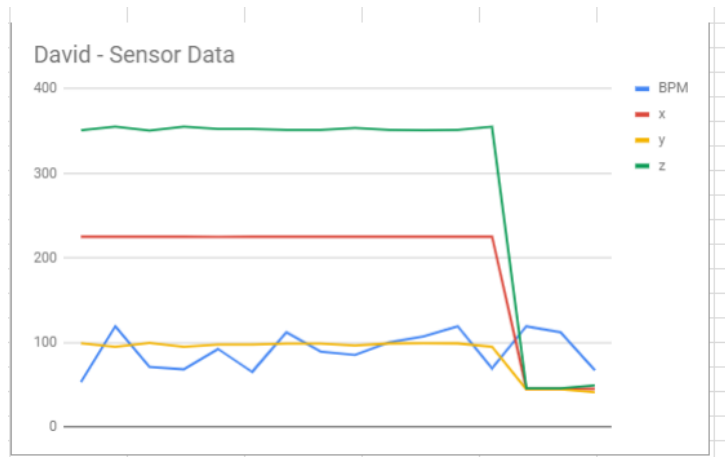
Nå som vi har fått sluttproduktet vi er fornøyd med lagres alle dataene i en separat mappe slik at vi er klare til å lagre sensor data til neste individuell person.



Merk at vi bruker samme google sheet, slik at webhook og koding og mer ikke må gjøres om igjen. Det er litt kronglete å lagre 100 individuelle sheet filer og jeg vurderte å lagre flere personer i samme sheet men det valgte jeg ikke å gjøre.

Hvorfor jeg valgte google sheet som tjeneste til løsningen

Hovedgrunnen jeg valgte å benytte google sheets som løsning var fordi det er lett å ryddig å vise flere rader med data. Spesielt når det lagres data hvert 5 – 10 sekund i 30 minutter. Samtidig kunne jeg sette opp en graf som oppdateres automatisk når data kommer inn, det er mange forskjellige grafer å velge mellom samtidig. Jeg endte opp med å bruke line chart, siden den er det jeg framstilte meg ville presentere denne dataen best. Jeg kunne ha testet med flere andre charts vis jeg ville også.



Siden google sheets er fleksibel kunne jeg også ha tatt tiden og inkludert dette i en chart. Men siden vi har en fiksert 30 minutter tidspunkt kan de som analyserer dataene utelukke dette.

Jeg hadde også som første plan å bruke IFFT til å lagre data i sheets, men måtte gi opp siden måten den setter opp data ville ikke vært mulig å lage til en sheet.

Konklusjon og refleksjon

Jeg er fornøyd med sluttresultatet. Men jeg føler at tjenesten jeg brukte var ikke perfekt nok til løsningen. Siden vi kan ikke ha en gjennomsnittsgraf fordi alle sheetsene er separert kan man ikke benytte en visuel graf med mindre man setter dette opp manuelt. Koden jeg brukte for å lese BPM er ikke helt presis heller og bruker delays. Jeg hadde også muligheten til å legge inn noe «ekstra» men dette valgte jeg ikke å gjøre på grunn av lite tid. Det er flere små detaljer å diskutere men overalt så har jeg lært en god del gjennom denne hjemme eksamen.