

obligatoriske oppgaven.

Programmet var skrevet i Python

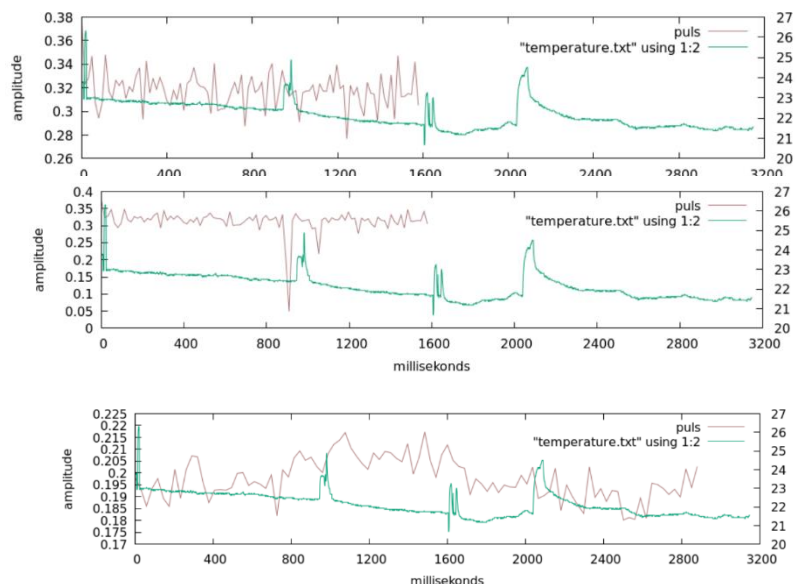
```
#!/usr/bin/python3
import Adafruit_BBIO.GPIO as GPIO
import Adafruit_BBIO.ADC as ADC
import time

# led lyser stabil
GPIO.setup('P9_14', GPIO.OUT)
GPIO.output('P9_14', GPIO.HIGH)
#GPIO.output('P9_14', GPIO.LOW)
GPIO.cleanup()
time.sleep(0.5)
ADC.setup()

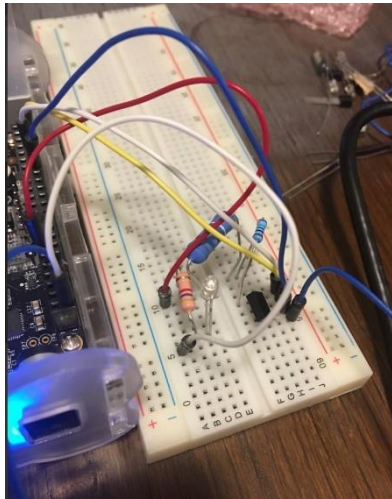
# åpner fil data.txt.
f = open('data.txt', 'a')
# x = 1 første nedtelling
x = 1
while True:
    value = ADC.read("P9_38")
    print(value)
    f.write(str(round(x, 1)) + ' ' + str(value) + '\n')
    f.flush()
    x += 3 # steg
    # hvor ofte måler vi data
    time.sleep(0.2)
```

Krets 1

Jeg koblet elementer for å få data ved å måle puls ved å endre blodvolumet i fingerarterien. Denne metoden refererer til fotopletysmografi. Fotopletysmografi er en ikke-invasiv metode for å måle endringer i blodvolum i vev ved hjelp av en lyskilde og fotodetektor. Siden endringen i blodvolum er synkronisert med hjerterytmen, kan denne metoden brukes til å beregne hjerterefrekvensen. Det er to hovedtyper av fotopletysmografi: den ene er basert på lystransmisjon, den andre er basert på refleksjon. Jeg brukte fotoresistor i den kretsen som tilsvarer lystransmisjon. For den krets brukte jeg P9 pins bare sånn – P9_38 som svarer analog inputs. P9_34 og P9_32 som tilsvarer Power, jord, og reset, til den kretsen involverte jeg en analog sensor fotoresistor. For å lage stabilt lys brukte jeg krets fra første

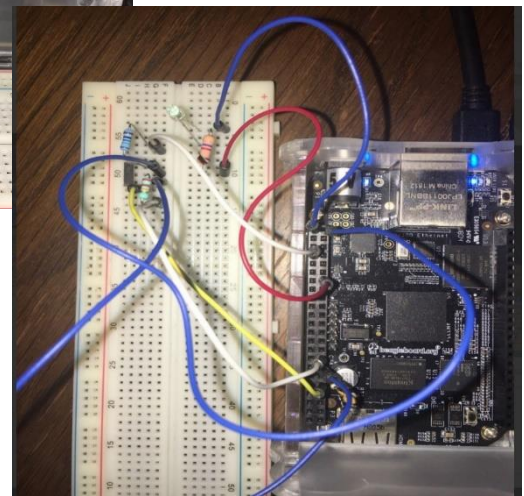
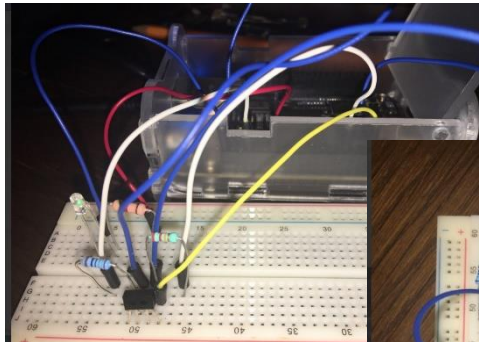


Den krets var enkelt å bygge. Så derfor er det bare et forsøk. Mye vanskelig er å måle data, hver gang de er helt forskjellige. Data er tilstrekkelig ikke til å finne antall slag per min. Sensor reagerer på relative endringer i lysintensitet. Hvis mengden lys som faller på sensoren forblir konstant, vil signalverdien forbli på 0,26. Mer lys og signalet stiger. Mindre lys - faller.



Krets 2

Det var flere forsøker med feil krets Et bilde med feil hardwareoppsett.



Krets 3 og 4

Fungerende hardwareoppsett

I denne kretsen brukte jeg en TCRT1000-sensor

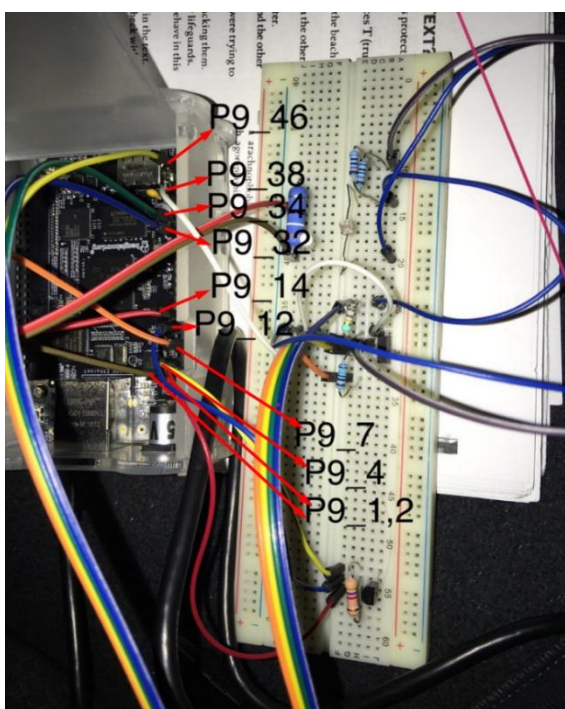
For å bygge hardwareoppsett med to analog sensor og en digital sensor

Sensorkortets forsyningsspenningsområde, lik 3-5 V

For å bestemme hjerterefrekvensverdien, må du først beregne utgangsfrekvensen og deretter multiplisere den resulterende verdien med 60.

Eller Minste hjerterefrekvens bestemmes av varigheten av det største intervallet R - R, og den maksimale pulsen bestemmes av det minste intervallet R - R. Beregningen av hjerterefrekvensen utføres i henhold til formelen:

$$\frac{60}{R - R'}$$



Noen informasjon var vurdert fra <https://robotclass.ru/tutorials/arduino-pulse-sensor/>
<https://studfile.net/preview/4623977/page:19/>

I prosjektet bruker jeg også en spesiell TCRT1000-sensor, som forenkler kretsen og utformingen, siden LED og fototransistor er plassert i en kompakt skjermet pakke. I det første tilfellet ledes lysstrålen gjennom en del av menneskekroppen (gjennom fingeren), og fotodetektoren bestemmer den resulterende lysintensiteten, slik at strålingskilden og mottakeren er plassert overfor hverandre den er basert på refleksjon. I det andre tilfellet er lyskilden og fotodetektoren plassert på samme side, og det reflekterte signalet bærer informasjon om pulsen. Avhengig av blodvolumet i fingeren absorberes mer eller mindre lys, derfor endres intensiteten til det reflekterte lyset. En grafisk fremstilling av avhengigheten av signalendringer i tid er signalet fra fotopletysmografi. Fotopletysmogrammet har flere komponenter; det registrerer bølger av første, andre og tredje orden. Bølger av andre og tredje orden refererer til langsomme svingninger (de kan kalles en konstant komponent). Første ordens bølger er raske svingninger og er relatert til puls (kan kalles variabel komponent). De gjenspeiler bevegelsen av blodvolum ved det målte punktet under systole og diastole og kan brukes som kilde til pulsinformasjon. For å trekke ut dette signalet, trenger du effektive forsterknings- og signalbehandlingskretser. Vel, faktisk fungerer det som et EKG. Du setter finger på, fotoresistoren leser endringer i blodtrykket, etter det kan man måle hvor lang tid det går mellom toppene.

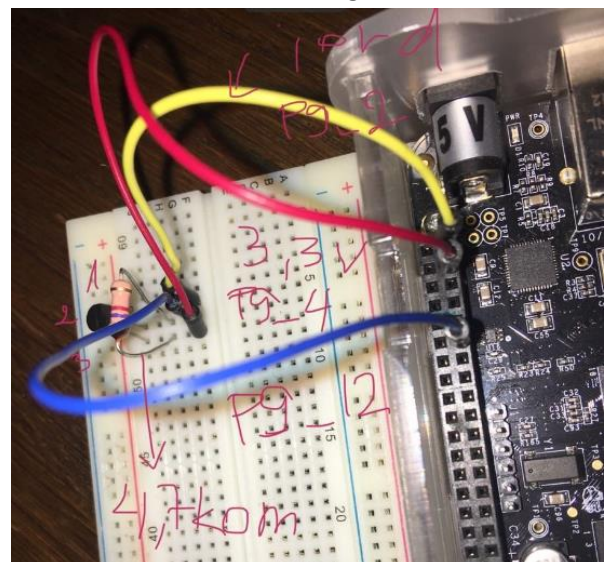
Jeg fikk noen resultat ved å bruke TCRT1000-sensor og digital sensor DS18B20(+), Som jeg kunne visualisere i gnu plot. Siden data er rådata så gjør det ikke mulig å telle puls. Koden nede er for å ta dato fra digital sensor, data skreves i txt filer: temperature.txt. Jeg brukte den lenken for å bygge den kode <https://stackoverflow.com/questions/4180867/make-datepicker-and-timepicker-smaller>. DS18B20(+), Sensoren opererer over et spenningsområde på 3,0 V til 5,5 V og måler temperatur fra -55 ° C til + 125 ° C (-67 ° F til + 257 ° F) med en nøyaktighet på ± 0,5 ° C (-10 ° C opptil + 85 ° C). i gnu plot kan vi se at den målte temperatur bra, temperaturer viser nøyaktig temperatur som tilsvare romtermometer. Når jeg sette finger på, får jeg høyretemperatur.

```
import time, re
# function: read and parse sensor data file
def read_sensor(path):
    value = "U"
    try:
        f = open(path, "r")
        line = f.readline()
        if re.match(r"([0-9a-f]{2}){9}: crc=[0-9a-f]{2} YES", line):
            line = f.readline()
            m = re.match(r"([0-9a-f]{2}){9}t=([+]?[0-9]+)", line)
            if m:
                value = str(float(m.group(2)) / 1000.0)
            f.close()
    except IOError:
        print(time.strftime("%x %X"), "Error reading", path)
    return value

# define pathes to 1-wire sensor data
pathes = (
    "/sys/bus/w1/devices/28-0000053853f7/w1_slave",
)

f = open('temperature.txt', 'a')
```

feil tilkobling



```

# read sensor data
x = 1
while True:
    for path in pathes:
        data = read_sensor(path)
        print(data)
        f.write(str(x)+' '+data+'\n')
        f.flush()
        x += 1
        time.sleep(2)

```

så etter på for å ta data fra to forskjellige sensorer (digital og analog) skrev jeg den koden, koden kan samle data fra to sensorer og separat skrive dem i to filer.txt

```

import time, re
import Adafruit_BBIO.GPIO as GPIO
import Adafruit_BBIO.ADC as ADC
def read_sensor(path):
    value = "U"
    try:
        f = open(path, "r")
        line = f.readline()
        if re.match(r"([0-9a-f]{2}){9}: crc=[0-9a-f]{2} YES", line):
            line = f.readline()
            m = re.match(r"([0-9a-f]{2}){9}t=([+]?[0-9]+)", line)
            if m:
                value = str(float(m.group(2)) / 1000.0)
        f.close()
    except IOError:
        print(time.strftime("%x %X"), "Error reading", path)
    return value

```

```

# define pathes to 1-wire sensor data
pathes = (
    "/sys/bus/w1/devices/28-0000053853f7/w1_slave",
)
GPIO.setup('P9_14', GPIO.OUT)
GPIO.output('P9_14', GPIO.HIGH)
#GPIO.output('P9_14', GPIO.LOW)
GPIO.cleanup()
time.sleep(0.5)
ADC.setup()

```

```

f1 = open('data.txt', 'a')
f2 = open('temperature.txt', 'a')

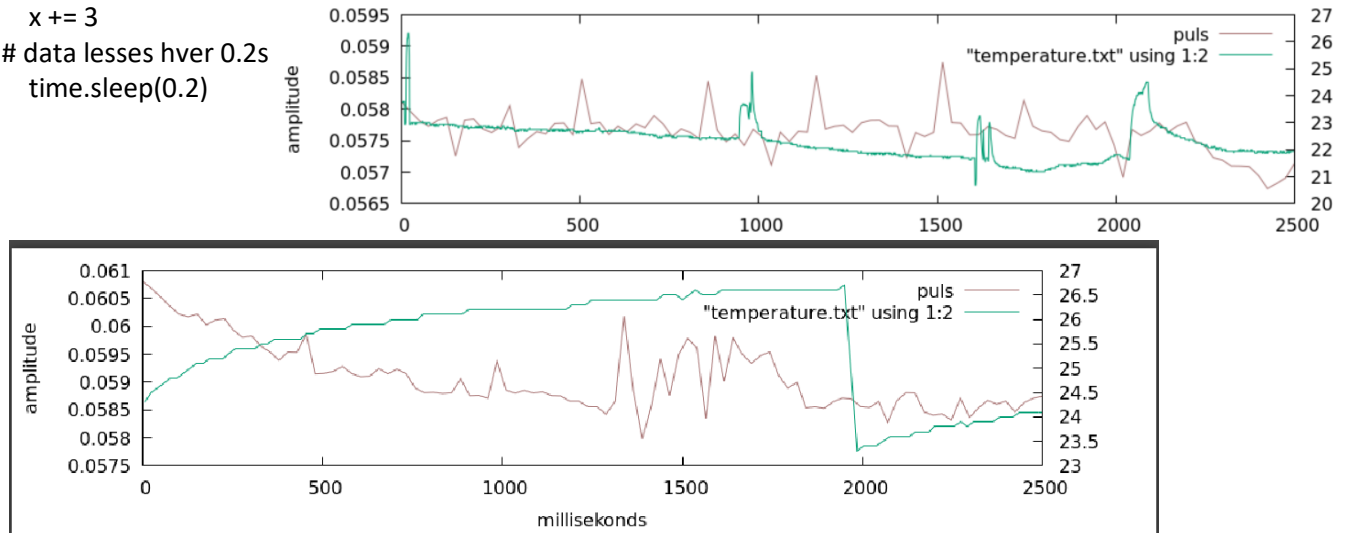
x = 1
while True:
    value = ADC.read("P9_38")
    print(value)
    f1.write(str(round(x, 1)) + ' ' + str(value) + '\n')
    f1.flush()
    if(x % 4 == 0):

```

```

for path in pathes:
    data = read_sensor(path)
    print(data)
# data fra digital signal skrives 1 tall etter koma.
    f2.write(str(x*1.5)+' '+str(round(float(data), 1))+'\n')
    f2.flush()
x += 3
# data leses hver 0.2s
time.sleep(0.2)

```

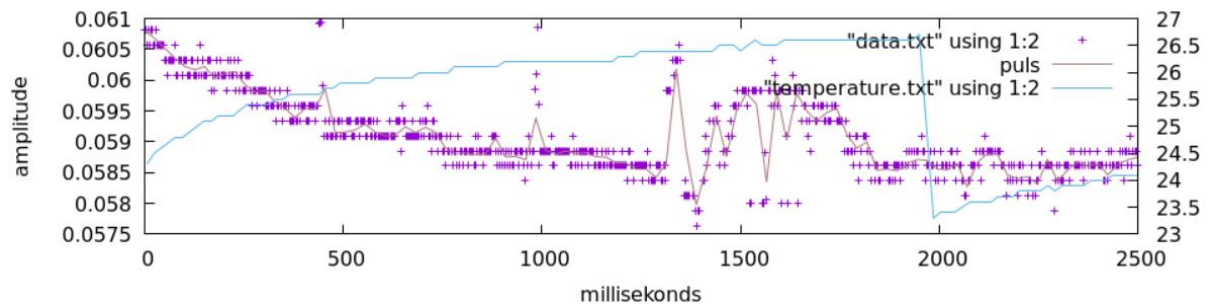


```

# separate value by space
set datafile separator " "
# output to PDF
set terminal pngcairo size 1000,250 font "Helvetica" background rgb "white"
set output "average.png"
set xlabel 'millisekonds'
set xrange [0:2500]
set ytics nomirror
# ny separate Y aksel
set y2tics
#set x2tics nomirror
set xtics 0,500, 2500
set ylabel 'amplitude'
# auto generate stats for the "data.txt" file to use later
stats "data.txt" using 1:2 name "stats"
#stats "temperature.txt" using 1:2 name "stats2"
# funktion returning how many data points we have processed
# capped to at least 1 and max 5
# ved bruk av h.h.v 5 punkter av pulsmåling
samples(x) = $0>4?5 : ($0+1)
# function to put current data point in back1, previous back1 value into back2, ets...
shift5(x)=(back5=back4, back4=back3, back3=back2, back2=back1, back1=x)
# function that calls shift above, than calculate avarage of all backX values (last 1-5 data points)
avg5(x)=(shift5(x), (back1+back2+back3+back4+back5)/samples($0))
# function to initialise values to 0 at start
init(x)=(back1=back2=back3=back4=back5=sum=0)
# actual plot (Note: Using stats, not the date file) og rådata
plot "data.txt" using 1:2 with points pointtype 1 pointsize 1 axis x1y1, sum = init(0), \

```

" using (\$1 - stats_min_x):(avg5(\$2)) with lines smooth csplines lc rgb '#aa7777' title 'puls' axis x1y1, "temperature.txt" using 1:2 with lines axis x1y2



0.8571428656578064
0.8583638668060303
0.8578754663467407
0.857631266117096
0.8578754663467407
0.8571428656578064
0.8571428656578064
0.8568986654281616
0.8571428656578064
0.059584859758615494
0.05934065952897072
0.05909645929932594
0.05909645929932594
0.05885225906968117
0.05885225906968117
0.05885225906968117
0.05909645929932594
0.05909645929932594
0.05909645929932594
0.05909645929932594
0.05885225906968117
0.05836385861039162
0.05860805884003639
0.05860805884003639
0.05860805884003639
0.05860805884003639
0.05860805884003639
0.05836385861039162
0.05836385861039162
0.05836385861039162
0.05836385861039162
0.05860805884003639
0.06568986922502518
0.9025641083717346
0.9028083086013794
0.9023199081420898
0.9025641083717346
0.9020757079124451
0.9023199081420898
0.9023199081420898
0.9023199081420898
0.9025641083717346

Hvis vi ser på disse data som jeg tokk vi kan se hvor mye de forandret seg hvis vi tar i utgangspunkt i null hypotese. Det var umulig å lage helt mørkt i rommet... derfor det er ikke helt null hypotese. Men vi kan se nøyaktig hvor mye varierer voltverdi dersom jeg målte i mørkerom. Brukt TCRT 1000.

Alle andre filer som koder, data, gnu plot filer, skal bli plassert i git.