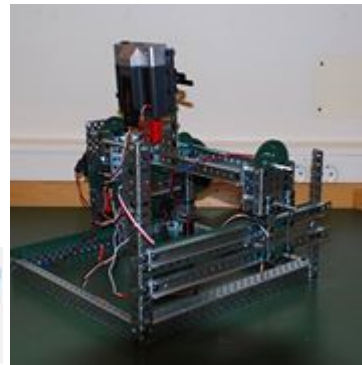


# Vélmenni II

Elmar og Sigurður

20. mars 2018





## Efnisyfirlit

<b>1</b>	<b>Inngangur</b>	<b>3</b>
<b>2</b>	<b>Vélbúnaður</b>	<b>3</b>
<b>3</b>	<b>Verkátun</b>	<b>4</b>
<b>4</b>	<b>Flæðirit og sauðakóði</b>	<b>4</b>
<b>5</b>	<b>Prófanir</b>	<b>4</b>
<b>6</b>	<b>Lokaorð</b>	<b>5</b>
<b>7</b>	<b>Heimildaskrá</b>	<b>5</b>
<b>8</b>	<b>Viðauki</b>	<b>6</b>
<b>9</b>	<b>Dagbók Elmar</b>	<b>6</b>
<b>10</b>	<b>Dagbók Sigurður</b>	<b>6</b>
10.1	Kóði . . . . .	8

## 1 Inngangur

Lestin Rocet! [2]

Markmiðið er að hanna og framleiða lest sem er stírd af tölvu sem er ekki á lestini það gæti þarfnast töluverða vírs forritun til að gera þennan "draum" að veruleika. Við töldum okkur ekki ætlast að fara neit djúft í rafmags fræðina en hún kom til okkar. Við völdum þetta verk út frá ASDF movie "I like trains" og af því að Elon Musk er fyrirmyndin okkar fyrir skritna og sharmandi vinnuhætti sem allir geta nutið. Við áttuðum okkur seinna á því að við þurftum að forrita með vörnum frekar en tölvuni því að allt transmission til tölvulausu lestarinnar er í gegnum ein vír(tein) og ein vír(tein) til að svara um skilavöðin sem fer beint í tölvuna, 3 input og 1 output sem samsvarar áhveðnum skilirðum. Inputin geta verið eitt af þrem skilaboðum sem er í raun bara áhveðin stór spennan eða ekki spennan, og outputið táknar alltaf bara hvort lestin sé á teinunum. Lestin er partur af closed circuit, um leið og lestin fer af teinunum verður allt kerfið open circuit og er ekki lengur functional. Í hvert skifti sem lestin á að vera á ferð, er talvan að lesa á response'ið og segja til um hvort allt er að virka eins og það á að gera samkvæmt áætlun. Hægt er að keyra lestina í ákveðin tíma eða (eins og við ætlum okkur) að keyra þangað til að circuit'ið er rofið með (til dæmis) teipi eða öðrum plast einangrunum, með þessum hætti getum við sagt til um hvar lestin er þegar hún stríkur yfir, aukalega væri hægt (með töluverði forritun) að senda skilaboð með rofningum eins og 2 rofningar í röð innan við 2 sekundur til að merkja við stöð númer 2, eða ef nákvæmnin leifir það binary skilaboð eins og teip, ekki teip, teip, teip væri túlkað sem 1011 bin eða 11 desimal. með þessum hætti væri hægt að senda margvísleg og flókin skilaboð frá lestini til að segja til um hvað talvan á að gera. Við áttuðum okkur ekki á því hversu mikla vinnu við myndum þurfa að setja í vörin til að triggja það að allt rennur vel og smoothly, eingin of hitnun og svoleiðis.

## 2 Vélbúnaður

Hér er tafla sem inniheldur vél og rafbúnað.

Vél/rafbúnaður	Spenna
Raspberry pi 3 [3]	
Breadboard	
H-Bridge [1]	
Tveir NPN resistorar	
Tveir teinar	
Vex 239 motor	7,2V
Lest	
Batterí	9v

### 3 Verkáætlun

Við skiptum okkur verkum, Sigurður sér um hardware hlutann og Elmar sér um software. Við ákvöddum það snemma að skipta því þannig því að Elmar hefur bakgrunn í Python og Sigurður sterka getu í verkfræðinni.

### 4 Flæðirit og sauðakóði

```
while true
```

```
stop
```

```
for run 6 times
```

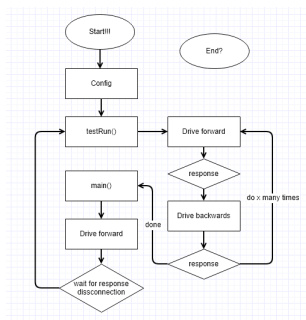
```
reverse=i()/2
```

```
drive(reverse)
```

```
counter+=response
```

```
while responses
```

```
drive
```



### 5 Prófanir

1. prófun var á Raspberry pi tölvuini, hún keyrði allt vel og engin vandræði þar.
2. prófun var á Breadboardinu, lærðum á það í leiðinni. 3 og 4 prófanir voru gerðar á tinkarcad.com og þá voru prófaðar tvær hugmyndir varðandi breadboardið. 5. prófun var gerð á breadboardinu sjálfu, og notað python, prófað var hvort við fáum svar í tölvuna ef stoppað er flæðið í lestina, allt gekk. 6. prófun var á lestina sjálfa

## 6 Lokaorð

Það gekk allt mjög vel í verkefninu en erfitt var að finna góða útfærslu á breadboardinu og einnig gekk illa að keyra lestina með teinunum en það var einungis hardware vandamál sem myndi þurfa algera hardware endurpælingu til að leysa, allt í allt var verkefnið skemmtilegt og gekk vel.

## 7 Heimildaskrá

Hér fyrir neðan sjáið þið heimildir.

### Heimildir

- [1] Miyuki Kanai, Hirokazu Fujimaki, and Takeshi Shimizu. H-bridge circuit, 2011. US Patent 7,902,884.
- [2] Chris McGowan. Rail, steam, and speed: The "rocket" and the birth of steam locomotion, 2010.
- [3] Simon Monk. *Programming the Raspberry Pi: getting started with Python*. TAB Electronics, 2015.

## 8 Viðauki

## 9 Dagbók Elmar

Dagur 1:

kynning hjá kennara um áfangann og raspberry pi.  
aukalega var byrjað að vinna með raspberry pi umhverfið.

Dagur 2:

haldið áfram var að fíkta með smá verkefni í raspberry pi,  
til að venjast umhverfinu.

Dagur 3:

error 404 day not found

dagur 4:

við Sigurður hittumst fyrr í vikuni og ræddum hugmynd um lest,  
í tímanum var svo talað meira um það og hvernig við myndum vinna það

verkefni.

aukalega byrjuðum við að skoða hvernig við getum notað batterí með raspberry

pi

og virkja motor.

Dagur 5:

Áframhald í víraþræðslu

Dagur 6:

testing

Ný vandamál komin

dekk ekki nægilega breið

hlösla í dekkjamótor ekki heppinleg

Dagur 7:

Samsetning kláruð, þarf að fá logic gates á bread board

Dagur 8:

breyting á teinum, fækkað í tvo teina

Dagur 9:

Tími féll niður

Dagur 10:

Endurhönnun á öllum raftengningum

Dagur 11:

veikur

Dagur 12:

Unnið í Tinkercad á samsetningu

Dagur 13:

gerðum forrit sem fær response frá lestini, virkaði, mjög sáttir.

## 10 Dagbók Sigurður

dagur 1:

kynning áfanga frá kennara. Sturtað yfir man upplýsinga um:

rasperryPY  
TexMaker  
resistors  
aukalega far byrjað að fíkta með rasperryPy umhverfið.  
dagur 2:  
fíktað var með sonar, feingum ekki útkomurnar sem við vildum.  
dagur 3:  
Elmar mætti ekki, svaf yfir sig  
komið fram með hugmyndir:  
mezesolver  
járnbrautalest (gerði flæði rit og útskýringar fyrir þetta)  
2fast4U (gearbox)  
dagur 4:  
ransóknir með circitry varðandi rafhlöður og stjórn á hleðsluni.  
rínt var inn í L293D til að ransaka notkun  
dagur 5:  
Áframhald á vírabræðslu í bread-board'inu góða  
dagur 6:  
fokk, ný vandarmál  
dekk ekki nægilega breið  
teingja hleðslu í lestamótorinn er ekki heppilegt  
Lausnir: skifta um motor(Vex) og tengja með gírur.  
nýtt heimaverkefni:  
samsetning á nýrri lest  
dagur 7:  
samansettning var kláruð með öllum moterfunctions  
redisign'aði systemið þannig að breadbordið væri á lestini og bætti við einum  
tein í viðbót til að nota sem response tain (ground-output-input)  
þarf að fá logic gates á breadbordið  
dagur 8:  
Fækað eru teinonum aftur niður í 2 teina, input og output, þarf að nota  
groundið á rafhlöðuni.  
dagur 9:  
Tími fellur niður  
dagur 10:  
endur hönnun á öllum rafteingingum  
dagur 11:  
unnið er á verkefna lýsingu  
dagur 12:  
skapað var hönnun 3 og 4  
3 og 4 eru að nota local cicle, 1 breadboard(ekki á lestinni)  
3: notar Polar transistora  
4: notar nonPolar transistora  
dagur 13:  
Rasperry Pi test með version 4 circitry set up. Virkaði!  
unnið er að koma skipulagi á skjölin.

## 10.1 Kóði

Hér er Python kóðinn sem er notaður í verkefnið.

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
enable = 18
HinputC = 23
HinputCC = 24
trnsC = 20
trnsCC = 21
responseC = 26
responseCC = 19

GPIO.setup(enable,GPIO.OUT)
GPIO.setup(HinputC,GPIO.OUT)
GPIO.setup(HinputCC,GPIO.OUT)
GPIO.setup(trnsC,GPIO.OUT)
GPIO.setup(trnsCC,GPIO.OUT)
GPIO.setup(responseC,GPIO.IN)
GPIO.setup(responseCC,GPIO.IN)

def stop():
    GPIO.output(HinputC,0)
    GPIO.output(HinputCC,0)
    GPIO.output(trnsC,0)
    GPIO.output(trnsCC,0)

def PolarRun(reverse):
    GPIO.output(HinputC,(reverse+1)%2)
    GPIO.output(trnsC,(reverse+1)%2)
    GPIO.output(HinputCC,reverse)
    GPIO.output(trnsCC,reverse)

def listener(reverse):
    if reverse:
        if GPIO.input(responseCC)==1:
            return True
    else:
        if GPIO.input(responseC)==1:
            return True
    return False
```



```

print ("setting_defaults")
stop()

print ("Enableing_H-bridge")

GPIO.output(enable,1)

def testRun(runs):
    sleep(1)
    print ("starting_test")
    sleep(1)
    Counter = 0
    for i in range(runs):
        PolarRun(i%2)
        sleep(0.8)
        response = listener(i%2)
        print ("Current_flowin_g_currently:_%s" % (response))
        if response:
            Counter+=1
        sleep(0.2)
    stop()
    print ("I_got_%s/%s_possible_responses" % (Counter,runs))
    print ("Turning_Rocket_off")
    if Counter==runs:
        return True
    else:
        return False

def main():
    PolarRun(0)
    while listener(0):
        pass
    stop()
    if testRun(6):
        main()

if testRun(6):
    main()
GPIO.cleanup()

```