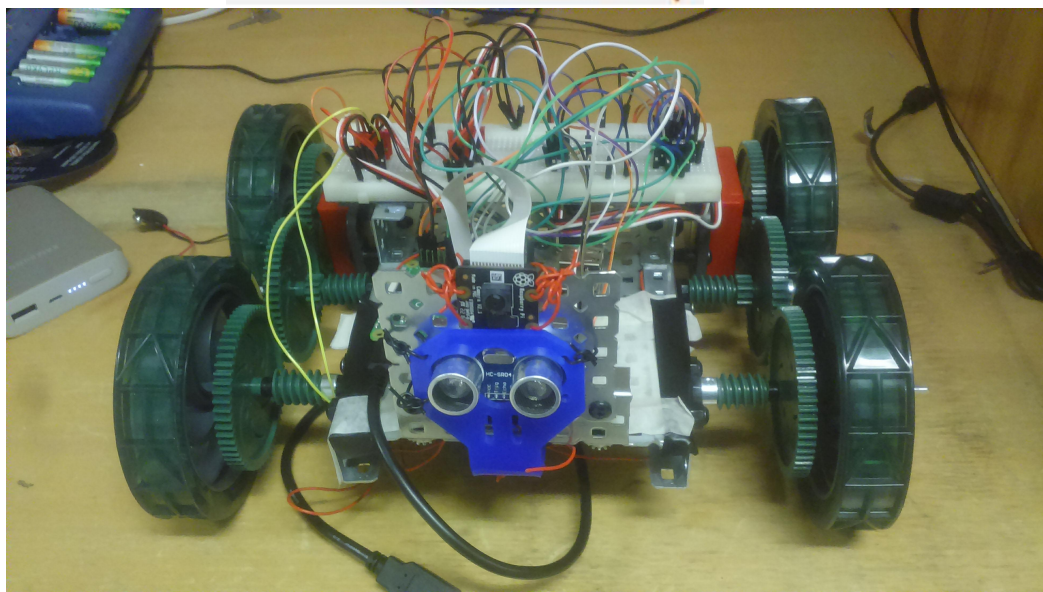


Vélmenni II

Sveinn Máni Jónsson og Vytautas Sipavicius

15. maí 2018



Efnisyfirlit

1	Inngangur	3
2	Vélbúnaður	3
3	Verkátun	4
4	Flæðirit og sauðakóði	5
5	Prófanir	5
6	Lokaorð	5
7	Heimildaskrá	7
8	Viðauki	8
8.1	Dagbók Sveinn	8
8.2	Dagbók Vytautas	9
8.3	Kóði Python	10
8.4	Kóði Index síða	13
8.5	Kóði Vefþjónn	14

1 Inngangur

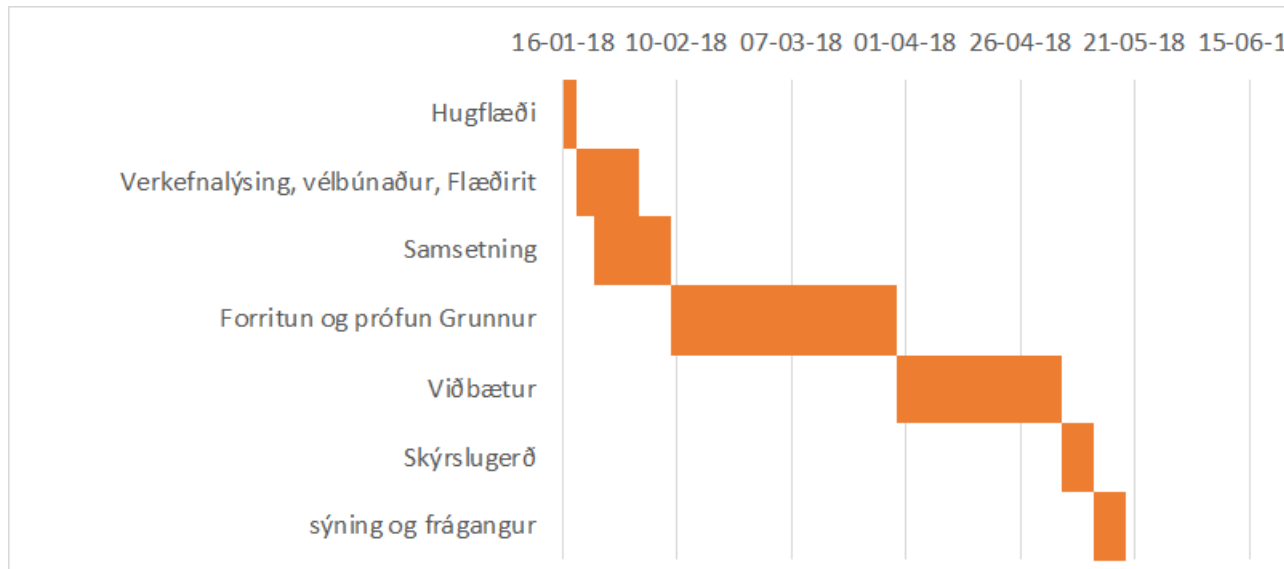
Við ákváðum að búa til eftirlits róbot. Í grunninn á vélmennið að fara hring í kringum tiltekið svæði og taka upp allt sem verður í vegi þess. Ástæðan fyrir því að við völdum þetta verkefni er því það er mjög sveigjanlegt og ekki flókið að fá grunn virknina til að virka. Síðan þegar það er komið er hægt að bæta við allskonar eiginleikum sem bæta virkni vélmennisins. Þær hugmyndir sem við höfum fengið og munum bæta við eftir því hversu mikin tíma við höfum eru t.d. leyfa manneskju að stýra vélmenninu með appi, láta myndavélina geta hreyfst í margar áttir, Gera vélmenninu kleift að keyra sjálfur án þess að klessa á neitt og komast aftur á rétta braut ef það þurfti að sveigja hjá einhverju, láta vélmennið þekkja sitt svæði og senda út viðvörun ef hann skynjar einhverja breytingu. Mismunandi og misflóknar aðferðir við að leysa margt að þessu sem gerir þetta að skemmtilegu og krefjandi verkefni án þess þó að enda með ekki neitt í höndunum. [2]



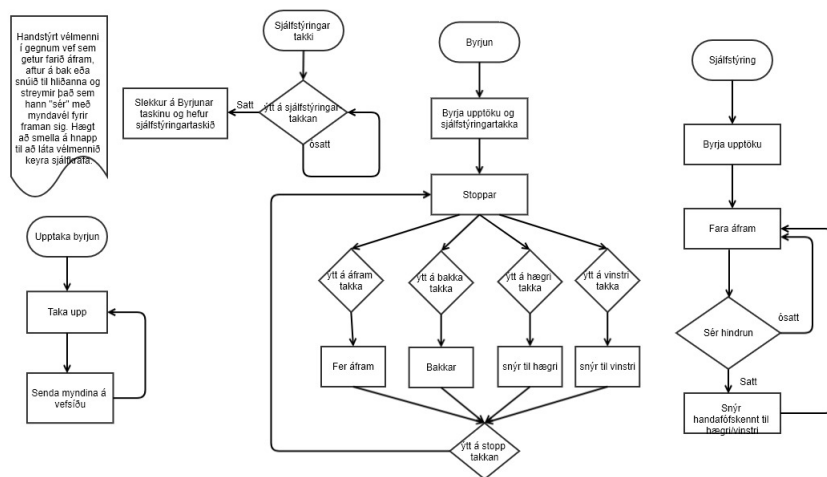
2 Vélbúnaður

Vél/rafbúnaður	Spenna	Viðnám
Raspberry Pi 3 Model B		
Breadboard		
HC-SR04 UltraSonic Sensor	5V	
Vex 2-Wire Motor 393	7,2V	
Vex Line Tracking Sensor		
Vefmyndavél		
Rafhlöður		

3 Verkáætlun



4 Flæðirit og sauðakóði



5 Prófanir

Áður en við byrjuðum að setja vélmennið okkar saman þá prófuðum við alla partana sem við ætluðum að nota fyrir þetta verkefni. Fyrsti parturinn sem við reyndum að fá til þess að virka var Ultrasonic sersnor sem við notum fremst á vélmenninu svo það keyri ekki á vegg. Kóðinn sem við notuðum til að athuga með sonar fengum við frá kennaranum. Við ætluðum að nota shaft encoder fyrir verkefnið og fengum það til þess að virka en hættum við að nota þá. Fengum mótor næst til þess að virka. Við notuðum H-Bridge til að stjórna þeim. Í vefhlutanum reyndum við að fá takka til þess að stjórna GPIO pinnunum. Það virkaði strax með LED perum en ekki með mótórum. Erum að reyna að láta vefinn keyra console commands.

6 Lokaorð

Verkefnið gekk mjög brösulega hjá okkur aðallega útaf tæknilegra erfiðleika með py tölvuna. Hún byrjaði mjög snemma að hitna mjög mikið og undir lokin á verkefninu þá hélt hún gangandi í mjög stuttan tíma þannig við náðum ekki að taka upp lokaafresturinn. Annars náðum við að framkvæma þá parta af verkefninu sem við vildum fá til að virka og þó það hafi tekið sinn tíma og mismunandi aðferðir með t.d. vefinn og kóðan þá virkaði allt saman undir lokin. Við lærðum að skipuleggja og setja saman verkefni á faglegri hátt en áður og einnig fengum betri innsýn á linux kerfum sem og nota python til að stjórna GPIO pinnunum. Samvinnan í hópnum gekk mjög vel og áttum við létt

með að skipta niður verkefnum en hjálpuðum þó hver öðrum þegar við lentum í mismunandi vandræðum. Allt í allt erum við sáttir með loka útkomuna með það í huga að þetta hafði getað verið mjög flott vélmenni hefðum við ekki lent í vandræðum með tölvuna.

7 Heimildaskrá

Heimildir

- [1] How to give your raspberry pi a static ip address, 2016.
- [2] Svetlana Cheusheva. How to make a gantt chart in excel 2010, 2013 and excel 2016, 2017.
- [3] hrvoje. Ultimate rotary encoder switch decoder, 2016.
- [4] Abdul Hannan Mustajab. How to control gpio pins and operate relays with the raspberry pi, 2017.
- [5] The pi hut. How to install / use the raspberry pi camera, 2014.
- [6] Ritvikdave. Controlling direction and speed of dc motor using raspberry pi, 2016.

8 Viðauki

8.1 Dagbók Sveinn

9.1.18 og 12.1.18

Fengum Raspberry pi í þessari viku og leystum fyrsta æfingaverkefnið þar sem við tengdum

led peru, servo og ultrasonic sensor við RPi með breadboard.

16.1.18 og 19.1.18

Við skrifuðum niður nokkrar hugmyndir fyrir verkefnið. Við byrjuðum á verkefnalýsingu og gerðum sauðakóðann

þegar við vorum búnir að velja verkefni. Settum upp TexWorks svo við þurfum ekki að pæla í því seinna.

23.1.18 og 26.1.18

Við völdum parta fyrir vélmennið, ultrasonic sensor, tvo mótora frá Vex, byggingarefni frá Vex. Byrjuðum einnig að setja saman vélmennið

sjálft. Reyndum að átta okkur á því hvernigvið getum fengið gildi frá shaft encoder svo við getum notað hann fyrir vélmennið.

30.1.18 og 2.2.18

Náði að koma shaft encoder í lag. Vorum líka að vinna í því að setja vélmennið saman.

6.2.18 og 9.2.18

Í þessari viku vorum við að mest að klára samsetninguna á vélmenninu. Byrjaði á því að keyra mótora.

13.2.18 og 16.2.18

Kláruðum grunnsamsetninguna á vélmenninu. Náði að keyra báða mótora. Við erum byrjaðir á því að bæta myndavél við vélmennið. Reyndum að leysa vesnið með raspberry pi-inn þar sem hann var að ofhitna við ekkert álag.

20.2.18 og 23.2.18

Setti upp webserver á pi-inn til þess að geta stjórnað honum gegnum netið. Nota Apache fyrir það og ætla að nota php með því. Er buinn að atti mig á því hvernig maður stjórnar gpio pinnunum af piinum gegnum php vefsíðu.

27.2.18 og 2.3.18

Bættum við bita fremst á vélmennið til að festa myndavélina og sonar sensorinn.

6.3.18 og 9.3.18

Byrjuðum á því að tengja saman alla partana við breadboard og reyndum að fá vélmennið til þess að keyra.

13.3.18 og 16.3.18

Lærði hvernig h-bridge virkar. Reyndi að fá encoderinn til þess að virka, þarf að gera það í næstu viku.

10.4.18 og 13.4.18

Reyna að ná stýringu á vélmenninu gegnum vefinn okkar. Einnig að finna útúr því hvernig hægt er að keyra python skrár gegnum vefinn.

24.4.18

Er farinn að vinna í kóðanum sem laetur vélmennið keyra sjálfkrafa þegar ytt er á takka á síðunni.

17.4.18

Fékk php skjal til að keyra shell command í gegnum terminal en það virkar ekki gegnum vefinn okkar.

4.5.18

Var að vinna í kóðanum svo vélmennið gæti keyrt áfram án þess að keyra á vegg.

8.5.18 og 11.5.18

Pi tölvan bilaði. Drap alltaf á sér eftir smá tíma. Aftengdum vélmennið af fengum nýjann. Erum að klára skýrsluna.

[3] [4] [6]

8.2 Dagbók Vytatas

"09/01/2018

Kynntist námsefni áfangans og fengum raspberry pi tölvuna í hendurnar og fengum led ljós

til að tengjast henni.

12/01/2018

Tengdum Sonar sensor og servo við raspberry pi og fengum allt til að virka saman.

16/01/2018

Fengum upplýsingar um verkefni áfangans og ákváðum hvað við ætlum að gera.

19/01/2018

Bjó til verkefnalýsingu, skýringarmynd af verkefninu og sauðakóða fyrir grunnverkefnið.

23/01/2018

Byrjuðum að plana og byggja skelina á vélmenninu

26/01/2018

Kláraði fulla verkefnalýsingu

06/02/2018

Héldum áfram að byggja skelina

09/02/2018

Kláruðum að byggja skelina og ég fór í það að klára gantt ritið

13/02/2018

Kláraði gantt ritið og við tengdum og fengum myndavélina til að virka

02/03/2018

Löguðum til tölvuna þannig að hún virki og kláruðum að byggja vélmennið

06/03/2018

Skóðaði möguleika fyrir stream og ákvað að nota motion

09/03/2018

Setti upp streamið þannig við fáum mynd

06/04/2018

Lagaði til streamið þannig það keyrir betur

10/04/2018
Setti streamið upp þannig það keyri við startup og tengist rétt við vefsíðuna
13/04/2018
Skoðuðum hvernig við fáum takkana á vefsíðunni til að virka og löguðum til
kóðann til að láta vélmennið keyra
20/04/2018
Breytti yfir á python webserver backend client
24/04/2018
Lagaði til serverinn og setti allt upp þannig að takkanir virki almennilega og
að réttu hlutanir ræsists við bootup"
[2] [5] [1]

8.3 Kóði Python

```
#Libraries
import RPi.GPIO as GPIO
import random
import time

#GPIO MOde (Board / BCM)
GPIO.setmode(GPIO.BCM)
#print("Something something darkside")
#set GPIO Pins
GPIO_TRIGGER = 5
GPIO_ECHO = 6
Motor1A = 10
Motor1B = 9
Motor1E = 11
Motor2A = 17
Motor2B = 27
Motor2E = 22
#set GPIO direction (IN / OUT)
GPIO.setup(GPIO_TRIGGER, GPIO.OUT)
GPIO.setup(GPIO_ECHO, GPIO.IN)
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(Motor1E,GPIO.OUT)
GPIO.setup(Motor2A,GPIO.OUT)
GPIO.setup(Motor2B,GPIO.OUT)
GPIO.setup(Motor2E,GPIO.OUT)

def rndm5050(a, b):
    test = random.random()
    if test < 0.5: return a
    else: return b
```

```
def motorForward():
    #Setting the first motor to move forward
    print("Driving_forward")
    GPIO.output(Motor1A, GPIO.HIGH)
    GPIO.output(Motor1B, GPIO.LOW)
    GPIO.output(Motor1E, GPIO.HIGH)

    #Setting the other motor to move forward
    GPIO.output(Motor2A, GPIO.HIGH)
    GPIO.output(Motor2B, GPIO.LOW)
    GPIO.output(Motor2E, GPIO.HIGH)

def motorBackward():
    #Setting the first motor to move backwards
    print("Backing_up")
    GPIO.output(Motor1A, GPIO.HIGH)
    GPIO.output(Motor1B, GPIO.LOW)
    GPIO.output(Motor1E, GPIO.HIGH)

    #Setting the other motor to move backwards
    GPIO.output(Motor2A, GPIO.HIGH)
    GPIO.output(Motor2B, GPIO.LOW)
    GPIO.output(Motor2E, GPIO.HIGH)

def motorLeft():
    #Setting the first motor to move forwards
    print("Turn_left")
    GPIO.output(Motor1A, GPIO.HIGH)
    GPIO.output(Motor1B, GPIO.LOW)
    GPIO.output(Motor1E, GPIO.HIGH)

    #Setting the other motor to move backwards
    GPIO.output(Motor2A, GPIO.HIGH)
    GPIO.output(Motor2B, GPIO.LOW)
    GPIO.output(Motor2E, GPIO.HIGH)

def motorRight():
    #Setting the first motor to move backwards
    print("Turn_right")
    GPIO.output(Motor1A, GPIO.LOW)
    GPIO.output(Motor1B, GPIO.HIGH)
    GPIO.output(Motor1E, GPIO.HIGH)

    #Setting the other motor to move forward
    GPIO.output(Motor2A, GPIO.HIGH)
    GPIO.output(Motor2B, GPIO.LOW)
```

```

GPIO.output(Motor2E, GPIO.HIGH)

def motorStop():
    GPIO.output(Motor1A, GPIO.LOW)
    GPIO.output(Motor1B, GPIO.LOW)
    GPIO.output(Motor1E, GPIO.LOW)

    GPIO.output(Motor2A, GPIO.LOW)
    GPIO.output(Motor2B, GPIO.LOW)
    GPIO.output(Motor2E, GPIO.LOW)

def distance():
    #set Trigger to HIGH
    GPIO.output(GPIO_TRIGGER, True)
    #set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(GPIO_TRIGGER, False)

    StartTime = time.time()
    StopTime = time.time()

    #save StartTime
    while GPIO.input(GPIO_ECHO) == 0:
        StartTime = time.time()
    #save time of arrival
    while GPIO.input(GPIO_ECHO) == 1:
        StopTime = time.time()

    #time difference between start and arrival
    TimeElapsed = StopTime - StartTime
    #multiply with the sonic speed (34300 cm/s)
    #and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2
    return distance

try:
    while True:
        dist = distance()
        print("Measured_Distance_=%_if_cm" % dist)
        #Herna kemur kodinn fyrir verkefnið
        if dist > 1000: #A eftir að meta hversu langt talan á að vera
            motorForward()
        else:
            motorStop()
            time.sleep(1)

```

```

        direct = rndm5050(1,2)
        if(direct == 1):
            motorRight()
        else:
            motorLeft()
        time.sleep(1)
#Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Measurement_stopped_by_User")
        GPIO.cleanup()

```

[6]

8.4 Kóði Index síða

```

<html>
    <head>
        <title>Security system</title>
        <link rel="stylesheet" type="text/css" href="css/style.css">
    </head>
    <body>
        <?php
        phpinfo();
        ?>
        
        <form action ="/forward" method="POST">
            <input type="submit" value="forward">
        </form>
        <form action ="/backward" method="POST">
            <input type="submit" value="backward">
        </form>
        <form action ="/left" method="POST">
            <input type="submit" value="left">
        </form>
        <form action ="/right" method="POST">
            <input type="submit" value="right">
        </form>
        <form action ="/stop" method="POST">
            <input type="submit" value="stop">
        </form>

        <button id="autorun" type="submit">Run code</button>
        <!--<script type="text/javascript" src="http://code.jquery.com/j
        <script src="javascript/main.js"></script -->
        <script type="text/javascript">
            var ip = "<?php_echo_$_SERVER['SERVER_ADDR'];>";
            alert(ip);

```

```

</script>
</body>
</html>

```

8.5 Kóði Vefþjónn

```

from bottle import route, run, template
from motor2 import *
@route('/')
def index():
    return template('/var/www/html/index.php')
@route('/forward', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.forward()
    return template('/var/www/html/index.php')
@route('/backward', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.backward()
    return template('/var/www/html/index.php')
@route('/left', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.left()
    return template('/var/www/html/index.php')
@route('/right', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.right()
    return template('/var/www/html/index.php')
@route('/right', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.right()
    return template('/var/www/html/index.php')
@route('/stop', method=['POST'])
def tralala():
    x = MotorKeyra()
    x.stop()
    return template('/var/www/html/index.php')
run(host='0.0.0.0', port=8000)

```