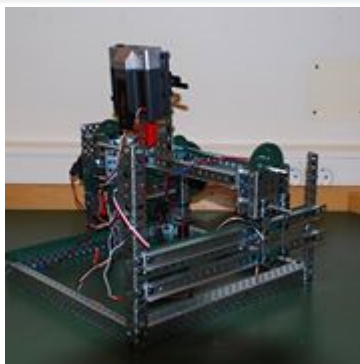


Vélmenni II

Mikael Máni og Jóel Snær

11. maí 2018



Efnisyfirlit

1	Inngangur	3
2	Vélbúnaður	3
3	Verkátun	4
4	Flæðirit og sauðakóði	5
5	Prófanir	5
6	Lokaorð	6
7	Heimildaskrá	7
8	Viðauki	8
8.1	Dagbók Mikaela	8
8.2	Dagbók Jóels	9
8.3	Kóði Python	9

1 Inngangur

Vélmennið sem við ætlum að byggja og forrita á þessari önn verður vélmenni sem er hægt að stjórna í gegnum síma með vefappi, við ætlum að nota RaspberryPi sem gruninn af vélmenninu. Kóðinn sem vélmennið keyrir verður að mestu leyti python kóði og það er tungumálið sem við völdum að nota til að kóða vélmennið en hinsvegar kóðinn á bakvið vefappið mun að mestu leyti vera JavaScript kóði. Vélmennið verður með Raspberry Pi myndavél sem mun vera notuð til að leyfa notandanum að sjá það sem vélmennið er að horfa á. Það mun stream-a live feed til vefappið þannig að stjórnandinn getur séð hvað hann er að gera um leið og hann er að keyra vélmennið. Við ætlum að nota motora úr Vex kittinu og stjórna þeim með Raspberry Pi. Við ætlum að setja upp myndavél og flygja [6] til að fá hana til að virka. Á vefappinu mun notandinn geta séð það sem myndavélin er að stream-a og hann á líka að geta stjórnað vélmenninu með snerti stýringum á símanum sínum. Við ákvöðum að nota Raspberry Pi yfir hinum valmöguleikunum, Arduino og Vex, vegna þess að það er mjög einfalt og þægilegt að nota það. Raspberry Pi er með built-in Python þannig við getum notað það til að forrita vélmennið og vegna þess að Raspberry Pi-ið er sín eigin lítil tölvu með net tengingu lætur það einfaldara fyrir okkur að ná að tengja það við vefappið þannig við getum stjórnað vélmenninu.

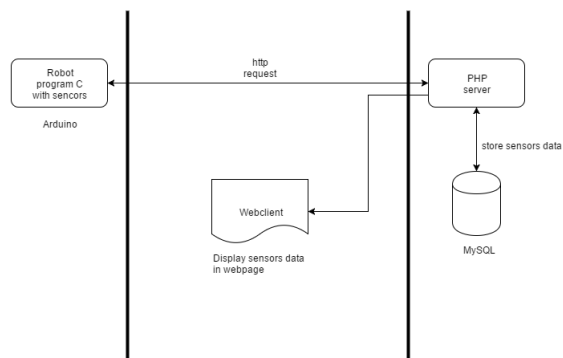
Við ákvöðum að nota Python til að forrita vélmennið útaf nokkrum einföldum ástæðum. Vegna þess að Python er mjög létt forritunartungumál að kóða í, það er léttar fyrir okkur að skrifa út dæmi og prófa hugmyndir okkar í Python meðað við önnur tungumál eins og til dæmis C. Raspberry Pi er mjög samhæft við python og það verður létt að nota Python til að forrita og láta það virka með Raspberry Pi-inu. Við ætlum að nota JavaScript fyrir vefappið vegna þess að JavaScript er létt og þægilegt tungumál að vinna með.

Svona tegund af vélmenni hefur mjög magvísilegt notgildi. Vélmennið gæti verið notað sem eftirlits vélmenni sem þú getur tengst þegar þú ert út úr húsi og vilt fylgjast með og athuga að allt sé eðlilegt og að allt sé á réttum stað heima hjá þér meðan þú ert í burtu. Hann getur líka verið notaður til að skoða staði sem fólk er of stórt til að komast inn léttilega eins og til dæmis loftræsikerfi. Þú gætir látt vélmennið inn í þar inn og keyrt um til að sjá ef það er eitthvað fast þar. Við gætum minnkað hversu stórt vélmennið er og látt það eins lítið og mögulega hægt er og svona mini vélmenni gætu verið notaður af sérsveit sem er að reyna bjarga húsi sem er búið að hertaka, vélmennið gæti verið sent inn á undan og fundið meira út um húsið og hvernig það er þar inni.

Við byggðum vélmennið eins lítin og við gátum, notuðum stál grind til að búa till lítinn kassa sem við tengdum allt við sá grind. 2 Mótórar sem keyra 2 hjól hver og svo er Raspberry Pi-ið fastur á sínum eigin pall ofaná vélmenninu.

2 Vélbúnaður

Hér skal gera töflu eða lista yfir allan búnað sem notaður er gott væri að þið nýttuð ykkur töfluna hér fyrir neðan:



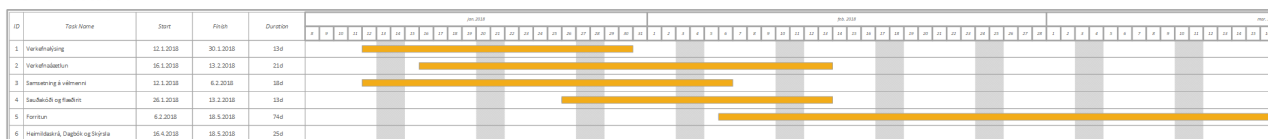
Vél/rafbúnaður	Spenna	Fjöldi
RaspberryPi		1
Cabstone PocketPower 11.2		1
RPi Camera Module		1
Breadboard		1
Jumper cable		11
H Bridge		1
Alkaline Battery	9V	1
Vex 239 motor	7,2V	2
VEX 2.75" Wheel		4
Miðstærð Tannhjól		4
Lítil Tannhjól		2

3 Verkáætlun

Hér skal gera verkáætlun og tímaáætlun, setja in mynd af henni gerð í <https://draw.io> veljið Flocharts-gant . Þegar þið hafið lokið við grafið farið í export-image og vistið sem 'gant' í skýrsla/img

ROB2B3U

Verkefnaáætlun



4 Flæðirit og sauðakóði

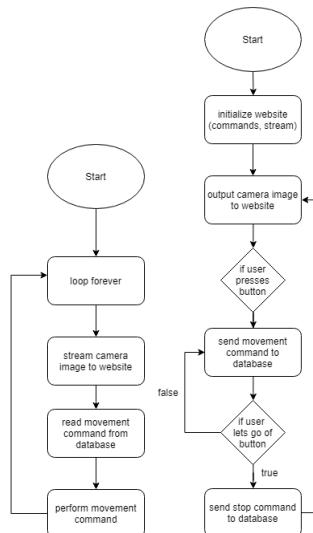
Hér skal gera flæðirit og sauðakóða nýtið ykkur <https://draw.io>. Þegar þið hafið lokið að gera flæðiritið farið í export-image og vistið grafið í skýrsla/img meðnafni "flowchart". Í þessu skjali skuluð þið gera sauðakóða dæmi:

robot code:

```
loop forever{ stream camera image
read newest movement command from database
perform movement command
}
```

website code:

```
loop forever{ output camera image to website
create control pad for user to use
if user presses a button send movement command to database
when user lets go of button send stop command to database
}
```



5 Prófanir

1. Prófanir á robot
Prófanirnar á robot Búið til og tengt robot. Tengt alla mótora og sett upp Enable-a SSH og camera á raspberry Pi-inu. Með því að fara í preferences->raspberrypi configuration->interfaces->velja enable á camera og ssh Sett upp Apache PHP og MySQL. - [8] Sett upp PHPmyadmin - [9] RPI Cam web interface - [7] Setja upp vefsíðu. Búa til python file, láta það keyra á startup með það að keyra það í /etc/rc.local Farið í gegnum þetta ferli þrisvar vegna þess að Pi-ið dó tvisvar, fyrsta skipti fór það í endalausa loop-u að restarta sér og í seinna skiptið vildi það ekki launcha stýrikerfinu vegna þess að það corrupt-aðist eitthvað

2. Prófanir á vefsíðu Prófanirnar á vefsíðu voru á mestu leiti bara að reyna að ná camera til að virka á vefinum. Þegar það var búið að ná því til að virka var það bara að búa til vefsíðu sem birtir camera stream-ið og tekur inn input frá notanda til að stjórna því. Mikael gerði function sem keyrði event listener á örvatókkunum þannig ef það er ýtt á þá er notað ajax til að uploada commands á database-inn. Vefsíðan virkaði öll frekar vel frá byrjun, Mikki lagaði stylesheet-ið og lagaði villu í python file-inu sem var ekki að lesa stop command rétt.

6 Lokaorð

Verkefnið virkaði vel, við náðum að gera það sem við ætluðum, vélmenni sem væri stjórnað remote í gegnum vefsíðu. Allt sem var tengt robot-inum gekk frekar vel fyrir utan að hann dó tvisvar og við þurftum að setja allt á honum upp aftur. Vefsíðu hönnunin gekk vel og hafði lítið af vandamálum. Samvinnan var góð og við unnum það vel. Mikael gerði mikið af set upp-inu sem tengist vélmenninu því ég var að missa af mörgum tímum vegna áreksturs. Ég(Jóel) skrifaði upp alla skýrsluna og fyllti inn í öll skjölin. Þetta var skemmtilegt verkefni að fylgjast með og vinna í.

7 Heimildaskrá

[6] [5] [2] [3] [7] [1] [4] [8] [9]

Heimildir

- [1] Jason Barnett. Controlling dc motors using python with a raspberry pi, 2014.
- [2] Raspberry Pi Foundation. Getting started with picamera, 2016.
- [3] Raspberry Pi Foundation. Setting up an apache wev server on a raspberry pi, 2016.
- [4] Scott Kildall. Raspberry pi: Launch python script on startup, 2014.
- [5] The PiHut. How to install / use the raspberry pi camera, 2014.
- [6] Sanjana Prasad, P Mahalakshmi, A John Clement Sunder, and R Swathi. Smart surveillance monitoring system using raspberry pi and pir sensor. *Int. J. Comput. Sci. Inf. Technol*, 5(6):7107–7109, 2014.
- [7] Unknown. Rpi-cam-web-interface, 2018.
- [8] Ste Wrightl. Tutorial – install apache, php and mysql on raspberry pi, 2012.
- [9] Ste Wrightl. Tutorial – install phpmyadmin on your raspberry pi, 2012.

8 Viðauki

Hér skal vera dagbók frá öllum í verkefninu .

8.1 Dagbók Mikael's

18/1/2018

clone-aði rob2b3u af github og gerði mitt eigið repository

25/1/2018

samsetning af róbota

1/2/2018

byrjaði að tengja motorana við robotan

9/2/2018

Lagaði alla víra og náði robotanum til að keyra

13/2/2018

Bætt við myndvél og fengið hana til að virka

23/2/2018

fundið leið til að stream-a videoinu frá pi-inu til web browser

2/3/2018

configureað video streamið

Lagað vírana á pi-inu.

9/3/2018

fokkaði upp myndavélinni

fokkaði upp pi-inu

reinstallaði os á pi-ið

lagaði myndavélina á pi-inu nokkurn veginn.

14/3/2018

lagað myndavélina og er núna stable

16/3/2018

setti upp phpmyadmin lenti í smá vandræðum með það

þar sem til að logga inn notarú ekkert pw en þú þarft

að setja inn pw til að ekki fá error

21/3/2018

lenti í vandræðum með motorona en lagaði það

6/4/2018

fann leið til að tengjast við sql server með python

11/4/2018

bjó aftur til prufu forrit sem keyrir motorona með python

þar sem ég gleymdi að uploada því á github og það eyddist út

13/4/2018

breytti um dekk á robotinum

byrjaði á aðal forritinu sem robotinn keyrir

8.2 Dagbók Jóels

26/01/2018

Unnið í verkefnalýsingu

30/01/2018

Klárað verkefnalýsingu

2/02/2018 - 20/2/2018

Unnið í verkafnaáætlun, lýsingu á vélbúnað osfrv.

23/02/2018

Byrjað að skrifa þessa dagbók, hlutir sem eru á undan þessum dag eru skrifuð eftir það sem ég pushaði á github

02/03/2018

Unnið í vefsíðunni sem verður notuð til að stjórna robotinum

06/03/2018

Skrifað upp basic sauðakóða fyrir hvað robotinn gerir, unnið í vefsíðunni meira

09/03/2018

Árekstrur var í öðrum áfanga

13/03/2018-04/10/2018

Gleymt að skrifa upp í dagbók

10/04/2018

Bætt við sauðakóða fyrir hvernig vefsíðan og robotinn mun virka, unnið meira í robot sjálfum

13/04/2018

Árekstur

17/04/2018

Breytt hardware skýrslu, update-að dagbók, unnið á html og css

20/04/2018

Árekstur

24/04/2018

Hjálpað Mikka með að gera vinna verkefnið

27/04/2018

Árekstur

01/05/2018

Frí dagur

04/05/2018

Hjálpað Mikael með að laga vefsíðuna sem hann gerði, náð robot til að virka

08/05/2018

Klárað allar skýrslur og gert þær til að prenta út

8.3 Kóði Python

Python kóði sem við keyrum á robotinum

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

```

GPIO.setmode(GPIO.BOARD)
#Ports for the left motor
Motor1A = 22
Motor1B = 18
Motor1E = 16
#Ports for the right motor
Motor2A = 37
Motor2B = 35
Motor2E = 33
#Set up the motors
#the left motors
GPIO.setup(Motor1A,GPIO.OUT)
GPIO.setup(Motor1B,GPIO.OUT)
GPIO.setup(Motor1E,GPIO.OUT)
#the right motors
GPIO.setup(Motor2A,GPIO.OUT)
GPIO.setup(Motor2B,GPIO.OUT)
GPIO.setup(Motor2E,GPIO.OUT)

while True:
    #Connect to database
    db = MySQLdb.connect(host="localhost", user="root", passwd="admin", db="Gagn")
    # you must create a Cursor object. It will let you execute all the queries y
    cur = db.cursor()
    # Fetching data from the database
    cur.execute("SELECT _action _FROM _direction _ORDER _BY _id _DESC _LIMIT _1")
    # print all the first cell of all the rows
    data = cur.fetchall()
    print(data[0][0])
    if data[0][0] == "stop":
        GPIO.output(Motor1E, GPIO.LOW)
        GPIO.output(Motor2E, GPIO.LOW)

    elif data[0][0] == "forward":
        GPIO.output(Motor1A,GPIO.HIGH)
        GPIO.output(Motor1B,GPIO.LOW)
        GPIO.output(Motor1E,GPIO.HIGH)

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

    elif data[0][0] == "backward":
        GPIO.output(Motor1A,GPIO.HIGH)
        GPIO.output(Motor1B,GPIO.HIGH)
        GPIO.output(Motor1E,GPIO.LOW)

```

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

elif data[0][0] == "left":
    GPIO.output(Motor1A,GPIO.HIGH)
    GPIO.output(Motor1B,GPIO.LOW)
    GPIO.output(Motor1E,GPIO.HIGH)

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

elif data[0][0] == "right":
    GPIO.output(Motor2A,GPIO.HIGH)
    GPIO.output(Motor2B,GPIO.LOW)
    GPIO.output(Motor2E,GPIO.HIGH)

    GPIO.output(Motor1A,GPIO.HIGH)
    GPIO.output(Motor1B,GPIO.HIGH)
    GPIO.output(Motor1E,GPIO.LOW)

db.close()
```