Hönnunar Skýrsla Robo2  
 Eftir Ármann Hallbert Jónsson og Gylfa Þór Helgaso

Efnisyfirlit

Lýsing á verkefni hér.

Vélbúnaður

Verkáætlun

Flæðirit og sauðakóði

Prófanir

Viðauki

Heimildir

Lýsining á verkefni

Lýsing.

Ætlum að búa til frumgerð að Rasberry Pi bíl sem notar ultra sonic til að forðast hættur sem eru á vegi hans. Pæling að hafa ljósaskynjara sem myndi kveikja og drepa á honum. Síðan hafa line follower til láta fara eftir braut en myndi nota sonic til að forðast hættur og finna línu/brautina aftur.

Hægt að nota í.

* Frumgerð að bíla sem gæti keyrt leið eða um götu. Prófa forrit áður en það er sett í stærri bíl.
* Hægt að bæta við frumgerðina til að lesa merkingar eins og skilti.
* Nota til að sýna fólk til að kveikja áhugan á bíl sem keyrir sig sjálfur.

Verkefni valið.

Við völdum þetta verkefni því við höldum að bílar sem keyra sig sjálfir eru framtíðinn. Að hafa reynt að gera frumgerð myndi auka þekkingu okkur á þessu sviði. Myndi vera okkur gangleg eftir tölvunarfræði í háskóla ef maður myndi vilja stefna áttina að tölvukeyrðum bílum.

Vélbúnaður

1x PCA9685 PWM Driver  
linkur : <https://www.adafruit.com/product/815>  
  
1 \* TB6612 Motor Driver  
linkur : https://www.amazon.com/Adafruit-TB6612-Stepper-Driver-Breakout/dp/B00VY32VU4a

1 \* 5-CH Line Follower Module  
linkur : <http://www.robotpark.com/5-Channel-Line-Tracking-Sensor-Module-BFD-1000>

1 \* Ultrasonic Obstacle Avoidance Module  
linkur : <https://www.amazon.com/Measuring-Ultrasonic-Distance-Avoidance-Raspberry/dp/B01N1KVIOD>  
  
1 \* Light Follower Module  
linkur : <https://www.amazon.com/Solu-TCRT5000-Infrared-Reflective-Photoelectric/dp/B00XDM4ZMQ/ref=sr_1_1?s=industrial&ie=UTF8&qid=1509976021&sr=1-1&keywords=Light+Follower+Module>

2 \* 18650 Battery Holder  
linkur : <https://www.amazon.com/SMAKN-Plastic-Battery-Storage-Holder/dp/B00EONNLTG/ref=sr_1_1?s=industrial&ie=UTF8&qid=1509976088&sr=1-1&keywords=18650+Battery+Holder>

2 \* DC Gear Motor  
linkur :

1 SunFounder SF0180 Servo   
linkur : <https://www.amazon.com/SunFounder-Digital-Helicopter-Airplane-Controls/dp/B01M5LIKLQ/ref=sr_1_1?s=industrial&ie=UTF8&qid=1509976218&sr=8-1&keywords=SunFounder+SF0180+Servo>

4 x Hjól   
  
1 \* Raspberry Pi 3  
linkur : https://www.amazon.com/Raspberry-Pi-Starter-Model-Heatsink/dp/B06Y6291VS/ref=sr\_1\_1\_sspa?s=industrial&ie=UTF8&qid=1509976278&sr=1-1-spons&keywords=raspberry+pi+3&psc=1

Verkáætlun

Vika 1: Hugmyndasmiði

Vika 2:Hugmyndasmiði

Vika 3: Hönnun og uppsetning

Vika 4: Hönnun og uppsetning

Vika 5: Hönnun og uppsetning

Vika 6: Hönnun og uppsetning

Vika 7: kóðun og test

Vika 8: kóðun og test

Vika 9: kóðun og test

Vika 10: kóðun og test

Vika 11: kóðun og test

Vika 12: kóðun og test

Vika 13: Prófanir

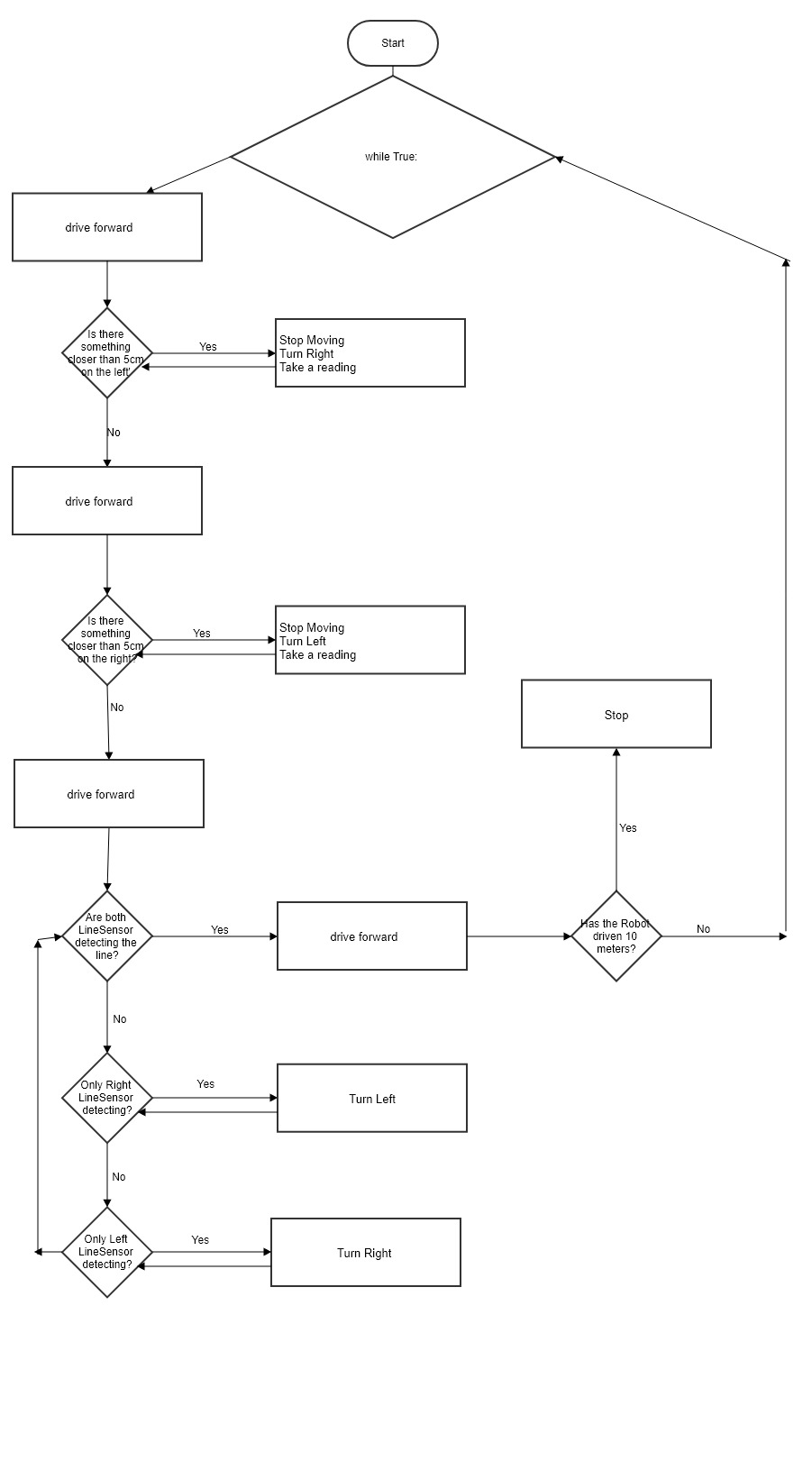
Vika 14: Prófanir

Vika 15: Sýning

Flæðirit og sauðakóði

While True:   
 Drive Forward()  
 if Right IR sensor detects an object  
 Turn Left  
 if Left IR sensor detects an object  
 Turn Right

Drive Forward()  
 if LineSensor detects nothing on Left  
 Turn Right  
 if LineSensor detects nothing on Right  
 Turn Left   
 Drive Forward()  
 if drive.length > 10meter  
 Stop()



Prófanir

Hugmynd að gera reglur fyrir bílinn eins og hann væri að keyra í borg, myndi fatta þegar leið hans væri lokuð og myndi finna aðra leið. Prófanir væru þá reglur sem hann myndi flygja í rétt röð eftir vægi þeirra.

1. Aðal hlutinn er auðvita að hann fari leiðina sem við viljum. Í byrjun er hún gerð með límbandi á jörðinni sem hann fer eftir. Þá nota sensor til að fara eftir henni. Fyrsta prófun er að hann fari eftir henni og pæli ekki í örðu.
2. Annar hluti sem við prófum er sensor sem passar að við keyrum ekki á hluti. Láttum bíl fara eftir línu þangað til hlutur kemur fyrir framan hann. Þá á hann að hæga á sér eða stoppa.
3. Þriðji hluti sem við prófum er að finna hvort bílinn komist framhjá hluti sem er á leið hann.

Ef hann kemst framhjá hindrun finnur hann línuna aftur og heldur síðan áfram.

1. Ef hann kemst ekki fram hjá hluti þarf að beina honum á aðra leið. Semsagt snúa við og reyna finna annan lit. Væri gert með að hafa til dæmis gulan tengdan við grænu brautina á öðrum stað.
2. Eftir að hann hefur farið gulu brautina þá og hittir á grænu að kveikja á því að elta hana.
3. Þegar hann kemur á enda punkt á að drepa á honum, eða hafa hann í park þangað annað er sagt.

Lokaorð

Verkefnið gekk upp og niður yfir önnina, lentum eins og flestir í eitthverju vesen að finna út hvar við vildum enda með þetta verkefni. Samvinnan okkar var með því besta eins og vanalega, þótt að github var ekki okkar besti vinur í byrjun.

Róbo gaf okkur aðeins meiri skilning á hvað það þarf að huga að miklu þegar kemur tækum sem keyra með bara með tölvukerfi. Hvað allt þarf að vera fullkomið í hönnun á kóða og framkvæmd hans. Ein villa gæti sent allt kerfið útum gluggan.

Myndi ekki segja að róbotinn okkar mun hafa eitthver hagnýtt gildi fyrir utan að kenna okkur. En þetta kveikti minnsta kosti mun meira áhuga á framtíðinn í tölvukeyrðum bílum eins og Teslu og fl. Þetta mun gefa betri sjón varðandi framtíðar nám ef maður skyldi velja að stefna í áttina að forrita faratæki eða álíka.

Viðauki

Dagbók

06.09.17

Settum upp dagbók. Panta aukahlut af amazon https://www.amazon.com/HiLetgo-Driver-Expansion-Control-Shield/dp/B00M0F25LK

18.09.17

Unnum að verklýsingu

25.09.17

Gerðum skjal með Íhlutum

09.10.17

Setti inn Sauðakóða

16.10.17

Setti inn Flæðirit

23.10.17

Uppfærðum Verklýsingu að ósk kennara

30.10.17

Hönnu á kóða

Robo sett saman

06.11.17

Uppfærðum íhluti og bættum í flowchart

kóði uppfærður

13.11.17

Byrjað að vinna í prófanir file.

Prófanir hannaðir

Byrjað að yfirfara allt

20.11.17

Heimildir

Prófanir byrjaðir

Uppfært Skýrslu

Sudocode og Flowchart lagað

Sumum skilað inná innu

27.11.17

Kóði

|  |
| --- |
| import RPi.GPIO as GPIO |
|  | import time |
|  | GPIO.setmode(GPIO.BOARD) |
|  | GPIO.setup(5,GPIO.OUT) #Left motor input A |
|  | GPIO.setup(7,GPIO.OUT) #Left motor input B |
|  | GPIO.setup(11,GPIO.OUT) #Right motor input A |
|  | GPIO.setup(13,GPIO.OUT) #Right motor input B |
|  | GPIO.setup(2,IO.IN) #GPIO 2 -> Left IR out |
|  | GPIO.setup(3,IO.IN) #GPIO 3 -> Right IR out |
|  | GPIO.setup(4,IO.OUT) #GPIO 4 -> Motor 1 terminal A |
|  | GPIO.setup(14,IO.OUT) #GPIO 14 -> Motor 1 terminal B |
|  | GPIO.setup(17,IO.OUT) #GPIO 17 -> Motor Left terminal A |
|  | GPIO.setup(18,IO.OUT) #GPIO 18 -> Motor Left terminal B |
|  | GPIO.setwarnings(False) |
|  |  |
|  | while True: |
|  | if(GPIO.input(2)==True and GPIO.input(3)==True): #both while move forward |
|  | GPIO.output(4,True) #1A+ |
|  | GPIO.output(14,False) #1B- |
|  | GPIO.output(17,True) #2A+ |
|  | GPIO.output(18,False) #2B- |
|  | print "Rotating both motors in clockwise direction" |
|  | i=GPIO.input(3) #Reading output of right IR sensor |
|  | j=GPIO.input(16) #Reading output of left IR sensor |
|  | GPIO.output(5,1) |
|  | GPIO.output(7,0) |
|  | GPIO.output(11,1) |
|  | GPIO.output(13,0) |
|  | time.sleep(1) #One second delay |
|  | if i==0: #Right IR sensor detects an object |
|  | print "Obstacle detected on Left",i |
|  | time.sleep(0.1) |
|  | elif j==0: #Left IR sensor detects an object |
|  | print "Obstacle detected on Right",j |
|  | time.sleep(0.1) |
|  |  |
|  | elif(GPIO.input(2)==False and GPIO.input(3)==True): #turn right |
|  | GPIO.output(4,True) #1A+ |
|  | GPIO.output(14,True) #1B- |
|  |  |
|  | GPIO.output(17,True) #2A+ |
|  | GPIO.output(18,False) #2B- |
|  |  |
|  | elif(GPIO.input(2)==True and GPIO.input(3)==False): #turn left |
|  | GPIO.output(4,True) #1A+ |
|  | GPIO.output(14,False) #1B- |
|  |  |
|  | GPIO.output(17,True) #2A+ |
|  | GPIO.output(18,True) #2B- |
|  |  |
|  | else: #stay still |
|  | GPIO.output(4,True) #1A+ |
|  | GPIO.output(14,True) #1B- |
|  |  |
|  | GPIO.output(17,True) #2A+ |
|  | GPIO.output(18,True) #2B- |
|  |  |
|  | print "Rotating both motors in anticlockwise direction" |
|  | GPIO.output(5,0) |
|  | GPIO.output(7,1) |
|  | GPIO.output(11,0) |
|  | GPIO.output(13,1) |
|  | time.sleep(1) |

Heimildir

Github áfanga.   
<https://github.com/eirben>

Leitarvél   
<https://google.com>

Hugmynd   
<https://diyhacking.com/raspberry-pi-robot/>