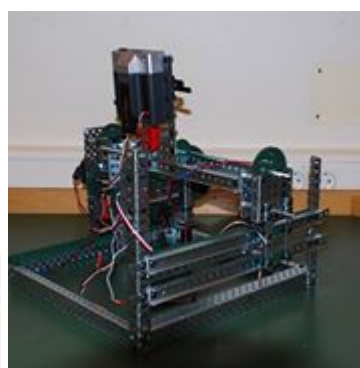


Vélmenni II

Sveinn Máni Jónsson og Vytautas Sipavicius

11. 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	6
8	Viðauki	7
8.1	Kóði Arduino	7

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 viðvörðun 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.



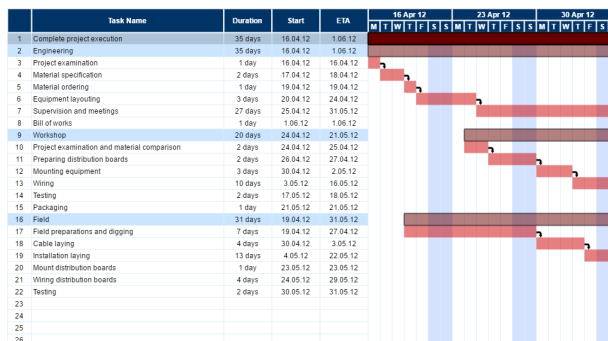
2 Vélbúnaður

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	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

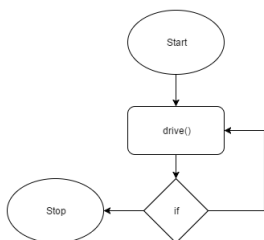
Hér skal gera verkáætlun og tímaáætlun, setja in mynd af henni gerð í <https://draw.io> veljið Flocharts-gant . Þegar þið hafið lokið við grafið farið í export-image og vistið sem 'gant' í skýrsla/img



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 "flowhart". Í Þessu skjali skuluð þið gera sauðakóða dæmi:

```
loop forever{ drive(until done)
ArmUp(30)
armDon(30)
clawOpen()
drive(until done)
}
```



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 sernsor 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 mótora næst til þess að virka. Við notuðum H-Bridge til að stjórna þeim. Í vefhlutanum renydum við að fá takka til þess að stjórna GPIO pinnunum. Það virkaði strax með LED perum en ekki með mótorum. Erum að reyna að láta vefinn keyra console commands. [?]

6 Lokaorð

Hér skal skrifa lokaorð um verkefnið, hvernig gékk, var gaman að vinna það hvað gékk vel og hvað illa. Hvernig var samvinnan :-) [1]

7 Heimildaskrá

Hér skal gera heimildaskrá, skoðið vel skrá sem heitir mybib.bib hér geymi ég allar heimildir mínar, þetta er einskonar gagnagrunnur. Þegar þið eruð að leita af bók eða grein notið þá <https://scholar.google.is>. finnið þar grein eða bók, þegar þið skráið heimildinar veljið þá Cite og afritið það og límið í mybib.bib. Þegar þið notið tilvitnun þá er það gert svona sjá intro.tex :-) Textan fyrir ofan eyðið þið áður en þið compælið og skilið skýrslunni.

Heimildir

- [1] J Dean Brock, Rebecca F Bruce, and Susan L Reiser. Using arduino for introductory programming courses. *Journal of Computing Sciences in Colleges*, 25(2):129–130, 2009.
- [2] MultiMedia LLC. MS Windows NT kernel description, 1999.
- [3] Simon Monk. *Programming Arduino*. McGraw-Hill Companies, USA, 2012.

8 Viðauki

Hér skal vera dagbók frá öllum í verkefninu .

14/12/2016

Bjó til dagbók og uppfærði skýrslu

8.1 Kóði Arduino

Hér hef ég includað kóðan frá arduino sem er forritunarmálið C. Þetta getið þið endurtekið fyrir php kóða sem þið vistið í möppuni php eða python í möppunni 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 = random.randint(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 ()
```