



Case Study

Verslag

IoTessentials ITFactory

Tristan Jacobs 1ITFD

Academiejaar 2019-2020

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

INHOUDSTAFEL

Inhoud

INHOUDSTAFEL	3
1 CORPUS.....	4
1.1 Foto's	4
1.2 Python code	6
1.3 Zelfevaluatie	9
1.4 Link naar Youtube	9
1.5 Conclusie	10

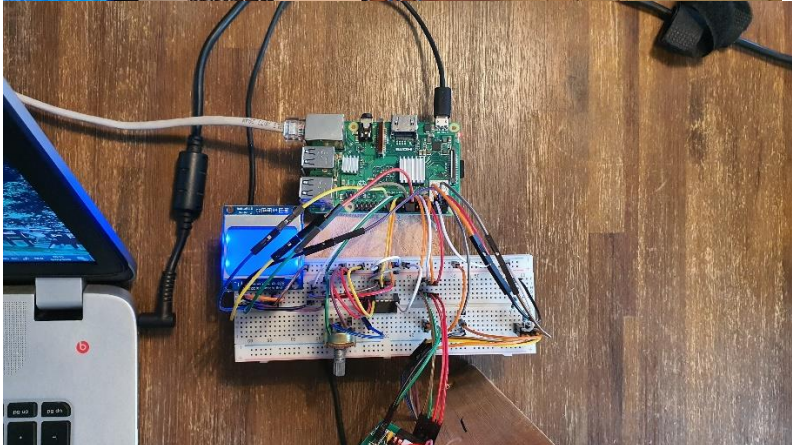
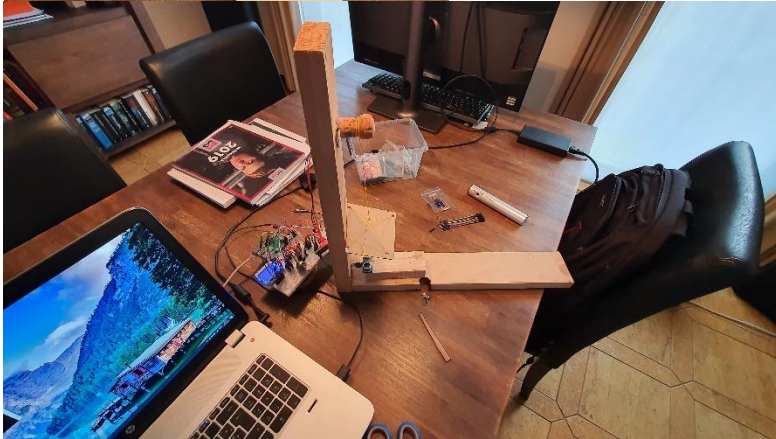
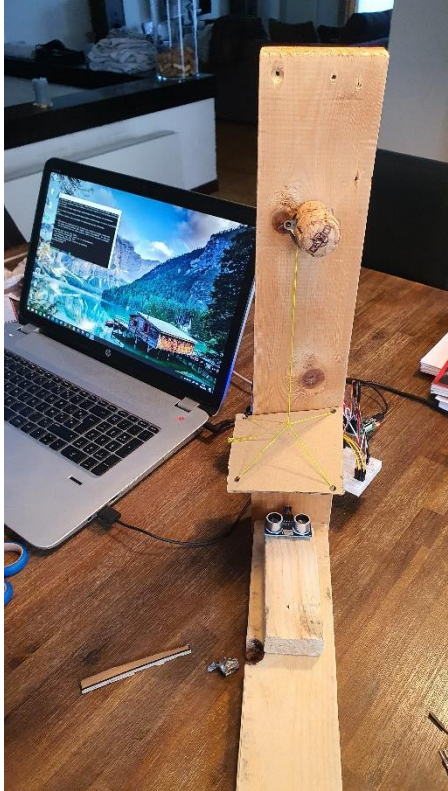
1 CORPUS

1.1 Foto's

Ik heb een constructie gemaakt van hout, in de verticale plank heb ik een gat gemaakt waar ik de steppermotor heb aan vastgevezen. Aan de steppermotor hangt een kurk waar het touw is aan vastgeplakt, dit is het mechanisme om het platform te laten bewegen. Langs de andere kant van de plank kome de kabeltjes eruit en gaan ze via het breadboard naar de Raspberry Pi. Onder het platform ligt de supersonische meter die meet hoe hoog het platform getrokken is, deze kabels gaan door een klein gaatje naar het breadboard. De LCD en de potentiometen zijn allebei bevestigd op het breadboard.

Er zijn een paar kabels die dezelfde pin op de Pi gebruiken, dus ik heb dit opgelost door een kabel van de pin naar het breadboard te trekken en op het breadboard de 2 kabels te verbinden.

Uiteindelijk is alles geconnecteerd en kon ik beginnen met coderen.



1.2 Python code

Dit is mijn Pythoncode. Ik ben begonnen met alles te importeren dat ik nodig heb, daarna heb ik de mode op BCM gezet. Vervolgens ben ik begonnen met het coderen van de steppermotor in fullstop, maar eerst moest ik de poorten instellen, daarna elke pin op modus 'uit' gezet.

Daarna heb ik de potentiometer ingesteld zodat deze de correcte waarde geeft, daarna de LCD en de ultrasonische meter.

Als laatste voor de while loop heb ik de functie voor de steppermotor gemaakt, daarbij heb ik ook een functie gemaakt zodat deze in de andere richting ook kan draaien.

Als laatste komt de while loop, hier staat de volledige code in van de ultrasonische sensor, dit komt omdat deze altijd opnieuw moet meten. Vervolgens heb ik een If-statement gemaakt om ervoor te zorgen dat de steppermotor naar een bepaalde richting draait samenhangend met de potentiometer.

Ik heb het zo ingesteld dat als de potentiometer ongeveer boven 50% zit, de steppermotor naar rechts draait. Als de potentiometer ongeveer onder de 50% zit, draait de steppermotor naar links.

Ten slotte heb ik alles laten weergeven op het LCD scherm. De sterretjes, de procent van de potentiometer en de afstand van de ultrasonische meter. Daarnaast geef ik de afstand en de procent ook weer in de terminal op mijn computer.

```
#steppermotor + potentie
import cgitb ; cgitb.enable()
import spidev
import RPi.GPIO as GPIO
import time
#LCD scherm
import Adafruit_Nokia_LCD as LCD
import Adafruit_GPIO.SPI as SPI
import PIL
from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont

#poorten raspberry pi initialiseren
GPIO.setmode(GPIO.BCM)
#GPIO.setup(17,GPIO.IN, pull_up_down=GPIO.PUD_UP) Als je met een knop
werkt is deze lijn nodig

#steppermotor
# #ingangen op het LED bordje
in1 = 18
in2 = 17
in3 = 27
in4 = 22

#poorten van PI op uit en laag zetten
GPIO.setup(in1, GPIO.OUT)
```

```

GPIO.output(in1,0)

GPIO.setup(in2, GPIO.OUT)
GPIO.output(in2, 0)

GPIO.setup(in3, GPIO.OUT)
GPIO.output(in3,0)

GPIO.setup(in4, GPIO.OUT)
GPIO.output(in4,0)

#code Potmeter
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz=(1000000)
def read_spi(channel):
    spidata = spi.xfer2([1, (8 + channel) << 4,0])
    return ((spidata[1] & 3) << 8) + spidata[2]

#Code LCD
DC = 23 #data control
RST = 24 #reset
SPI_PORT = 0 #spi port 0
SPI_DEVICE = 1 #cs1 pin 26

disp = LCD.PCD8544(DC, RST, spi=SPI.SpiDev(SPI_PORT, SPI_DEVICE,
max_speed_hz=4000000))
image = Image.new('1', (LCD.LCDWIDTH, LCD.LCDHEIGHT))
disp.begin(contrast=60)
disp.clear()
disp.display()
font = ImageFont.load_default()
draw = ImageDraw.Draw(image)

#poorten van ultrasonic
trig = 2
echo = 3

GPIO.setup(trig, GPIO.OUT)
GPIO.setup(echo, GPIO.IN)

#grenzen
max_height = 22
min_height = 4

#functies steppermotor
def fullstop(in1, in2, in3, in4):
    GPIO.output(in1,1)
    GPIO.output(in2,0)
    GPIO.output(in3,1)
    GPIO.output(in4,0)
    time.sleep(0.005)

    GPIO.output(in1,0)
    GPIO.output(in2,1)
    GPIO.output(in3,1)
    GPIO.output(in4,0)
    time.sleep(0.005)

```

```

GPIO.output(in1,0)
GPIO.output(in2,1)
GPIO.output(in3,0)
GPIO.output(in4,1)
time.sleep(0.005)

GPIO.output(in1, 1)
GPIO.output(in2, 0)
GPIO.output(in3, 0)
GPIO.output(in4, 1)
time.sleep(0.005)

def fullstop_reverse(in1, in2, in3, in4):
    GPIO.output(in4, 1)
    GPIO.output(in3, 0)
    GPIO.output(in2, 0)
    GPIO.output(in1, 1)
    time.sleep(0.005)

    GPIO.output(in4, 0)
    GPIO.output(in3, 0)
    GPIO.output(in2, 1)
    GPIO.output(in1, 1)
    time.sleep(0.005)

    GPIO.output(in4,0)
    GPIO.output(in3,1)
    GPIO.output(in2,1)
    GPIO.output(in1,0)
    time.sleep(0.005)

    GPIO.output(in4, 1)
    GPIO.output(in3, 1)
    GPIO.output(in2, 0)
    GPIO.output(in1, 0)
    time.sleep(0.005)

while True:
    #potmeter naar %
    channeldata = read_spi(0)
    pot_procent = (channeldata / 1023) * 100
    pot_procent = round(pot_procent, 1)

    #afstand
    GPIO.output(trig, True)
    time.sleep(0.0001)
    GPIO.output(trig, False)

    while GPIO.input(echo) == 0:
        pass
    PulseStart = time.time()

    while GPIO.input(echo) == 1:
        pass
    PulseEnd = time.time()

```

```

PulseDuration = PulseEnd - PulseStart
Distance = round((PulseDuration * 17150), 2)

#steppermotor richting
if channeldata > 525:
    fullstop(in1, in2, in3, in4)
elif channeldata < 505:
    fullstop_reverse(in4, in3, in2, in1)

#sterretjes
if Distance < 5:
    sterretjes = ""
elif 5 < Distance < 10:
    sterretjes = "*****"
elif 10 < Distance < 20:
    sterretjes = "*****"

#weergave op lcd
draw.rectangle((0, 0, LCD.LCDWIDTH, LCD.LCDHEIGHT), outline=255,
fill=255)
draw.text((1, 0), 'Potmeter: ', font=font)
draw.text((1, 8), (str(pot_procent) + "%"), font=font)
draw.text((1, 16), (str("Afstand: ")), font=font)
draw.text((1, 24), (str(Distance) + "cm"), font=font)
draw.text((1, 32), sterretjes, font=font)
disp.image(image)
disp.display()

time.sleep(0.05)

disp.clear()
disp.display()

#print in terminal
print("Afsand: {}".format(Distance))
print("Potmeter: {}".format(pot_procent))

```

1.3 Zelfevaluatie

Mijn punten gaan redelijk goed zijn vind ik zelf. Ik heb ervoor gezorgd dat mijn constructie een platform naar boven en naar beneden kan laten bewegen. Ook heb ik alle waardes correct laten weergeven op mijn LCD scherm. Dit zijn samen al 12 punten. Ook volgt het platform de potentiometer, dat is 4 punten erbij. Het enige dat niet is gelukt is de visualisatie in Ubeac, dit zijn een paar puntjes kwijtgespeeld. Maar als alles goed is heb ik 16 punten.

1.4 Link naar Youtube

<https://www.youtube.com/watch?v=My5R-GIXJeE>

Dit is de link naar het Youtube filmpje. Hier leg ik kort in uit hoe mijn opstelling werkt en u ziet hoe de opstelling eruit ziet.

1.5 Conclusie

Het was een zeer leuk en uitdagend project om je skills van Python en IOT te testen. Soms was het een beetje zoeken om de kabeltjes juist in te steken maar het is allemaal goed gelukt. Het is leuk dat alles lukt na een lange tijd coderen en debuggen.