Project Aquarium

Verslag

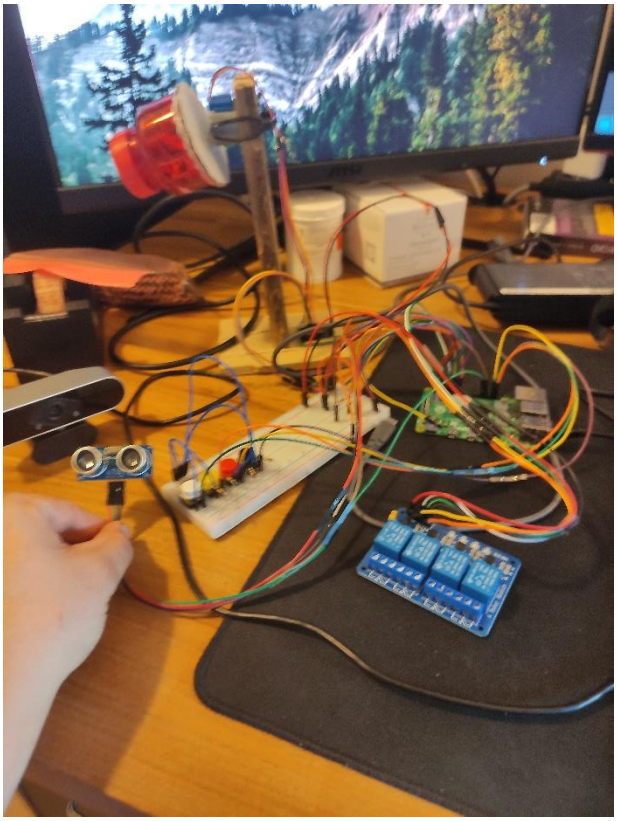
VAK IoT Essentials

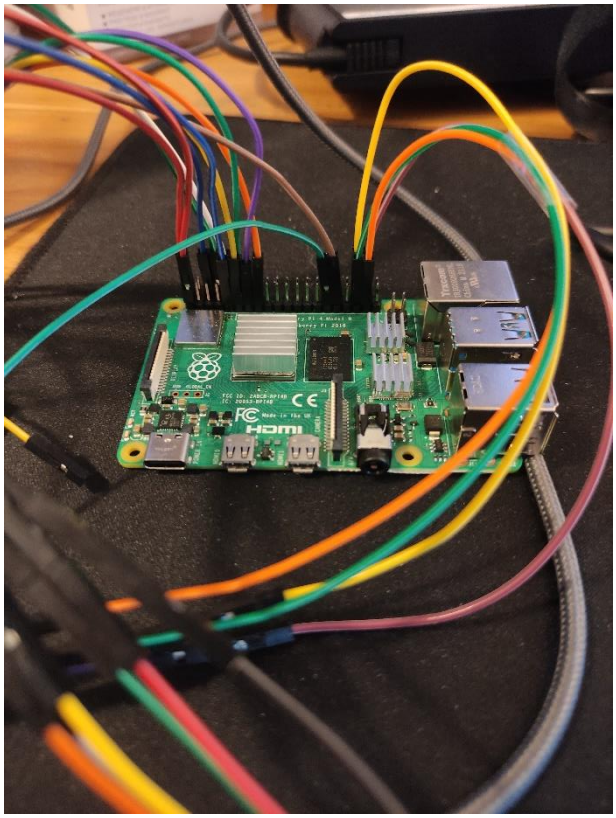
Mathias Wouters 1ITF-13

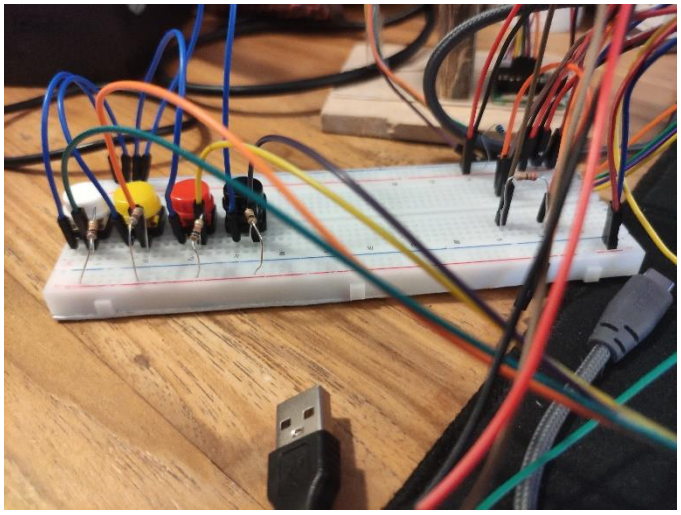
Academiejaar 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Opstelling







Code

```
import threading
import RPi.GPIO as GPIO
import time
from datetime import datetime
import datetime

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(17, GPIO.OUT) #step motor
GPIO.setup(18, GPIO.OUT) #step motor
GPIO.setup(27, GPIO.OUT) #step motor
GPIO.setup(22, GPIO.OUT) #step motor
GPIO.setup(19, GPIO.IN) #button step motor
GPIO.setup(26, GPIO.IN) #button step motor

GPIO.setup(14, GPIO.OUT) #relay light
GPIO.setup(15, GPIO.OUT) #relay pump
GPIO.setup(16, GPIO.IN) #button relay light
GPIO.setup(20, GPIO.IN) #button relay pump

GPIO_TRIGGER = 5 #hc-sr04 trigger
GPIO_ECHO = 6 #hc-sr04 echo
GPIO.setup(GPIO_TRIGGER,GPIO.OUT) # hc-sr04
GPIO.setup(GPIO_ECHO,GPIO.IN) # hc-sr04
GPIO.output(GPIO_TRIGGER, False) # hc-sr04
print ('Waiting a few seconds for the sensor to settle')
time.sleep(2)

feeding_time = "12:00"
light_time = "22:00"

exit_event = threading.Event()


################################################
################################################
########
# Stepper motor

def on(pin):
   GPIO.setup(pin, GPIO.OUT)
   GPIO.output(pin, 1)

def off(pin):
   GPIO.setup(pin, GPIO.OUT)
   GPIO.output(pin, 0)
```

```
def stepper():
    while True:
        now = datetime.now()                    # kijkt hoelaat het is
        current_time = now.strftime("%H:%M") # geeft de juiste format voor tijd
        button_state1 = GPIO.input(19)
        button_state2 = GPIO.input(26)
        if button_state1 == False and button_state2 == True:
            on(17)
            on(18)
            time.sleep(0.01)
            off(17)
            on(27)
            time.sleep(0.01)
            off(18)
            on(22)
            time.sleep(0.01)
            off(27)
            on(17)
            time.sleep(0.01)
            off(22)
            on(18)
        elif button_state2 == False and button_state1 == True:
            on(18)
            off(22)
            time.sleep(0.01)
            on(17)
            off(27)
            time.sleep(0.01)
            on(22)
            off(18)
            time.sleep(0.01)
            on(27)
            off(17)
            time.sleep(0.01)
            on(18)
            on(17)
        if current_time == feeding_time:
            on(17)
            on(18)
            time.sleep(0.01)
            off(17)
            on(27)
            time.sleep(0.01)
            off(18)
            on(22)
            time.sleep(0.01)
            off(27)
            on(17)
            time.sleep(0.01)
            off(22)
            on(18)
        if exit_event.is_set():
            off(17)
            off(18)
            off(22)
            off(27)
            GPIO.cleanup()
```

```
##################################################################
##################################################################
########
# HC-SR04

def hcsr04():
    global distance
    distance = 0
    while True:
        GPIO.output(GPIO_TRIGGER, True)
        time.sleep(0.00001)
        GPIO.output(GPIO_TRIGGER, False)
        while GPIO.input(GPIO_ECHO)==0: # meet de tijd tussen dat de golf
verstuurd wordt en deze terug aankomt
            pulse_start = time.time()

        while GPIO.input(GPIO_ECHO)==1:
            pulse_end = time.time()

        pulse_duration = pulse_end - pulse_start
        distance = pulse_duration * 17165   # berekening om de tijd naar een
afstand om te zetten
        rounddistance = round(distance, 1)
        print ('Distance:',distance,'cm')
        if distance <= 15:
            print("Pump on")
            time.sleep(4)
        else:
            print("Pump off")
            time.sleep(4)
        if exit_event.is_set():
            GPIO.cleanup()
```

```
##################################################
##################################################
########
# Button

def light():
    button_state3 = GPIO.input(16)
    now_light = datetime.now()
    current_time_light = now.strftime("%H:%M")
    if button_state3 == False:
        l = int(1)
        GPIO.output(14,  GPIO.HIGH)
        while l == 1:
            print("Lights on")
            l += 1
    else:
        m = int(1)
        GPIO.output(14,  GPIO.LOW)
        while m == 1:
            print("Lights off")
            m += 1
    if current_time_light == light_time:
        print("Lights on")
    if exit_event.is_set():
        GPIO.cleanup()

def pump():
    button_state4 = GPIO.input(20)
    if button_state4 == False:
        n = int(1)
        GPIO.output(15,  GPIO.HIGH)
        while n == 1:
            print("Pump on")
            n += 1
    else:
        o = int(1)
        GPIO.output(15,  GPIO.LOW)
        while o == 1:
            print("Pump off")
            o += 1
    if distance <= 15:
        while distance < 15:
            GPIO.output(15,  GPIO.HIGH)
            print("Pump on")
    if exit_event.is_set():
        GPIO.cleanup()
```

```
##################################################
##################################################
########

t1 = threading.Thread(target=stepper)
t2 = threading.Thread(target=light)
t3 = threading.Thread(target=pump)
t4 = threading.Thread(target=hcsr04)

t1.start()
t2.start()
t3.start()
t4.start()
```

Zelfevaluatie

Stepper drops food
De vissen worden om 12 uur 's middags gevoederd. Dus dit deeltje is gelukt.

Ultrasonic measurment of depth
Er wordt om de 4 seconden de afstandgemeten en als deze onder 15 cm komt gaat de pomp aan. Er wordt ook in de terminal gezegd wat de afstand is en of dat de pomp aan of uit staat, dit is dus ook gelukt.

Pump and light turn on/off
De pomp gaat dus aan als er te weinig water is in het aquarium. En het licht gaat op een ingesteld uur automatisch aan.

Buttons
De knoppen om de stepper motor in de 2 richtingen te laten draaien werken net zoals de knoppen om de pomp en het licht aan te doen.

LCD
De LCD is me niet gelukt om te laten werken.

Ubeac upload
De Ubeac upload heb ik geprobeerd, maar het is niet gelukt om mijn data door te sturen.

Live stream
De live stream is gelukt door een webcam.

Phone control / Voice control
Is niet gelukt

Really nice physical setup
Ik vindt dat ik wel redelijk wat werk heb gestoken in mijn opstelling.

YouTube

https://youtu.be/dGJX4TkoIcM