

# 7 Servomotoren

## 7.1 Einddoel

We sturen 4 servomotoren aan om een kraan te besturen. Je hebt twee vormpjes en die moeten beide van een beginplek naar een eindplek gebracht worden.

Als je klaar bent met de opdracht lever je in It's learning in:

- De definitieve python code.
- Een mp4-filmpje met het verplaatsen van de vormpjes.

Als dit gedaan is laat je de opdracht zien aan de docent zodat deze de opdracht kan goedkeuren.

## 7.2 Kennis

Voor deze opdracht heb je kennis nodig van microPython.

## 7.3 Benodigdheden

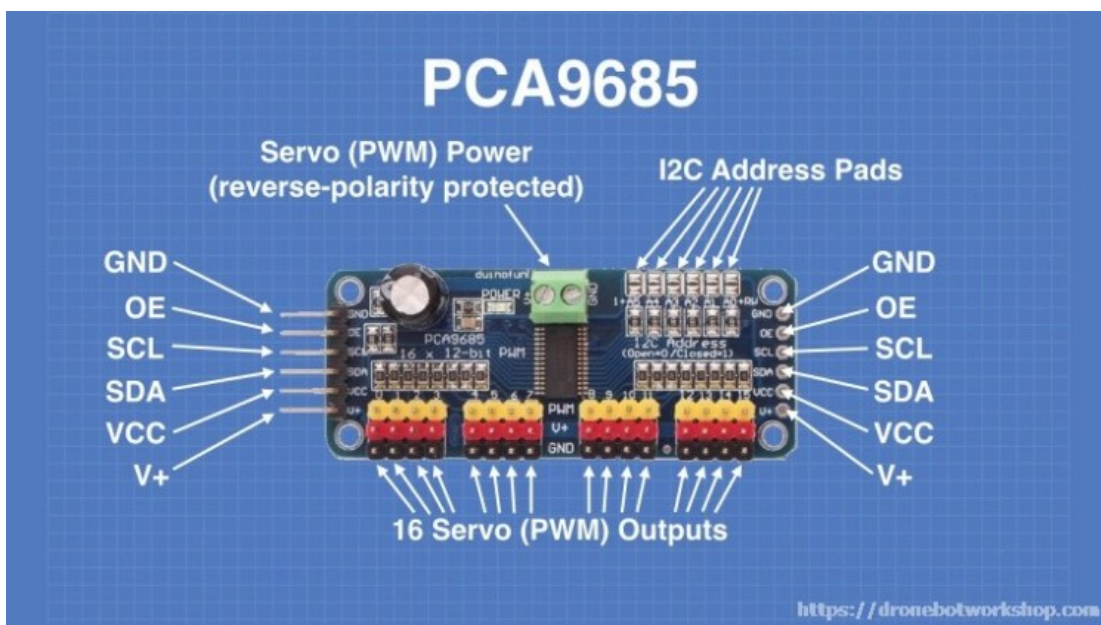
Voor deze opdracht heb je de onderstaande materialen nodig. Controleer aan het begin of al deze spullen aanwezig zijn. Bij het opruimen dien je weer te controleren of alles aanwezig is. Indien er iets defect is geraakt moet je de docent op de hoogte brengen.

- Raspberry Pi Pico – H
- UBS-usb mini kabel
- Opstelling van de kraan
- PCA9685 PWM-controller
- Externe 5V voeding

## 7.4 Opdracht

### 7.4.1 Aansturen van 1 servomotor

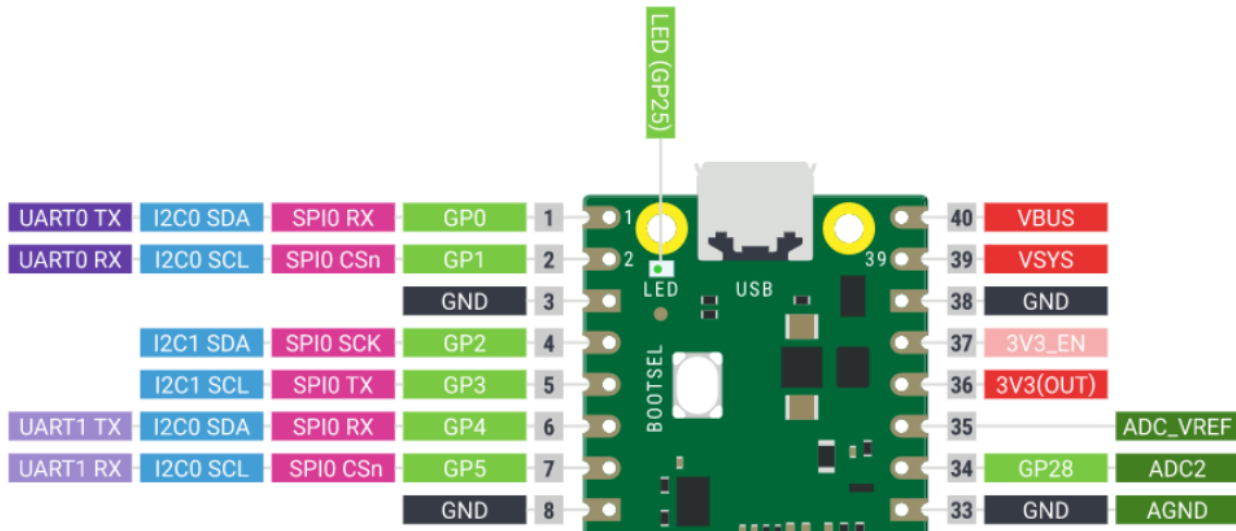
De vier servomotoren van de kraan gaan we aansturen met een PCA9685 PWM controller. De motoren kunnen bij elkaar te veel stroom gebruiken dat de Raspberry Pi Pico niet kan leveren.



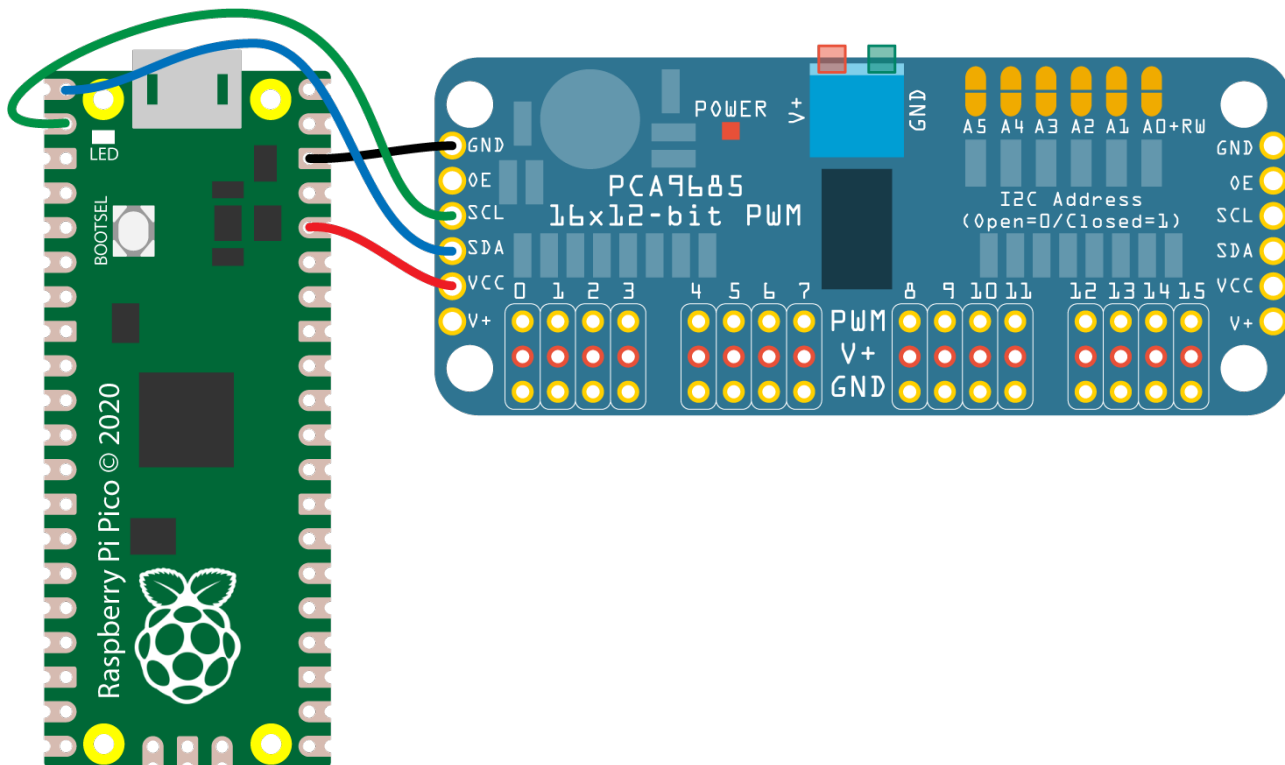
Zoek op internet wat PWM is, dit zal de docent aan het einde van de les vragen of je dit kan uitleggen.

Zowel aan de linkerkant als aan de rechter kant zie je dezelfde aansluitingen. We gebruiken nu nog alleen de connectoren aan de linker kant. Aan de rechterkant kan je nog een PCA9685 of een ander I2C apparaat kunnen aansluiten. Voor deze opdracht gebruiken we maar 4 servomotoren, en deze module kan al 16 servomotoren aansturen.

We praten met dit apparaat via i2c. Dit is een serieel communicatieprotocol. Een serieel communicatieprotocol heeft als kracht dat je tot 256 apparaten kan aansturen met slechts 2 GP-pinnen op je microcontroller. We hoeven alleen de GND en VCC aan te sluiten op respectievelijk GND en 3.3V en we gebruiken de SCL en SDA voor data. Als we naar de Pico kijken zien we in de blauwe vakjes dat dit i2c ook daar ondersteund wordt. De SDA van de PCA9685 moet op de SDA (data) van de pico aangesloten zijn, en de SCL (clock) van de PCA9685 op de SCL van de Pico.



De aansluiting zal er dus als volgt uitzien.



Als we nu onderstaande code uitvoeren dan zien we de naam van het apparaat. (Er worden twee adressen gegeven in dit voorbeeld, alleen de bovenste moeten we gebruiken.

```

1 from machine import I2C, Pin
2
3 i2c = I2C(id = 0, sda=Pin(0), scl=Pin(1))
4 devices = i2c.scan()
5
6 print('Scan i2c bus...')
7 if len(devices) == 0:
8     print("No i2c device !")
9 else:
10     print('i2c devices found:',len(devices))
11
12     for device in devices:
13         print("Decimal address: ",device," | Hexa address: ",hex(device))
14

```

```

shell x
>>> %Run -c $EDITOR_CONTENT

```

```

MPY: soft reboot
Scan i2c bus...
i2c devices found: 2
Decimal address: 64 | Hexa address: 0x40
Decimal address: 112 | Hexa address: 0x70

```

>>>

In It's learning zien we twee bestanden die we als eerste op de pico moeten zetten. Dat zijn de hulpprogramma's om te communiceren met de PWM-module.

- Pca9685.py
- Servo.py

Als we onderstaande code overnemen en we sluiten de servomotor aan op de correcte pinnen dan zien we dan de motor op index 4 (let op we beginnen te tellen met 0) bewegen.

Let op! De externe voeding van de PCA9685 is aangesloten op 5V, dit mag niet vanaf de pico komen.

```

1 from machine import I2C, Pin
2 from servo import Servos
3 from time import sleep
4
5 i2c = I2C(id=0, sda=Pin(0), scl=Pin(1))
6 servo = Servos(i2c=i2c, address=0x40)
7
8 for pos in (0,10,20,30,40,50,60,80,180):
9     servo.position(index=4, degrees=pos)
10     sleep(.50)

```

#### 7.4.2 Bewegen van de 4 motoren in de kraan

De opdracht is nu om de 4 motoren zo aan te sturen dat het object het parcours aflegt...



TODO