

P&O 2

Samenvatting voor individuele vraag

Materiaal, Sensoren, Motoren, andere delen.

- USB Webcam 1080P → normale webcam
→ USB-port
→ afstand + detectie
→ Rechtstreeks met pc verbonden
- Micro Metal Gear 100:1 HP
- Arduino nano 33 iot → microcontroller
→ Verbinding tot pc en alle geprogrammeerde elektronica (motoren, pomp en waterflauwenvol)
→ Programmeen via software van Arduino (USB)
→ Levert ook stroom (aangesloten op pc)
- Breadboard Full-size → gebruikt voor de tijdelijke circuits
→ Arduino + Dual Diode hierop bevestigen
en zo verbinden met motoren, pomp, ... zonder te solderen
- Batterij 9V → Nodig voor relais (5V)
→ Voltage regulator om 9V → 5V te gaan
- Nebraan pomp 12V 4,8 bar → verbonden met universele voeding van 12V (met externe relai)
- Jerrycan 10L
- Makrobeams → 10 mm dik
→ l = 10 of 30 cm
→ + Hoekverbindingen 90° (binnen/buiten)
- Flexible slang → Ø 10 mm
→ + slangenklemmen

- MDF → platformen + draaiend platform
- Plasticen aamlijstjes → arm (licht)
- Universele hub voor 3mm assen → motoren zo makkelijk te bevestigen ($a_s = 3\text{mm}$)
- Relais → Soort van schakelaar
→ Regelt stroomtoevoer naar de pomp
- Dual Drive → Motoren zijn hieraan verbonden (interface)
 - zelf verbonden ^{met} Arduino (zorgt ook voor stroom)
 - Regelt de spanning over de motor
 - Dual → 2 motoren mogelijk
 - Kan richting v/d motoren omdraaien
- Whadda waterflowsensor → bevestigt bevestigen de arm
→ Meet de snelheid van het water dat passeert en ook het debiet
 - Nodig voor de afstandberekeningen
 - Weet dus ook hoeveel water er geparseerd is
- Step-Down Voltage Regulator → 9V → 5V
→ Nodig voor de voeding v/d relais

Berekeningen hoek waterstraal

Berekening

- Maximale afstand tot cilinder: $x^2 = 3^2 + 10^2 \Rightarrow x \approx 10,45$

- Hoek θ berekenen om een afstand x te halen:

$$x = \cos \theta \cdot v \cdot t$$

$$x_{\text{act}} = R_{\text{platform}} + R_{\text{arm}} + \sin \theta \cdot v \cdot t - \frac{1}{2} \cdot g \cdot t^2$$

\downarrow
 $= \text{ring}$

$\Leftrightarrow x \rightarrow \text{door camera}$

$v \rightarrow \text{waterflow-sensor}$

$$\Rightarrow t \Rightarrow \theta$$

Code

- functie HoekV(waterdebit, afstandBeker)

\Leftrightarrow python gemaakt, gebruikt in Arduino

\Leftrightarrow waterdebit in l/min \rightarrow waterflowsensor

\Leftrightarrow afstand beker \rightarrow camera

\Leftrightarrow bovenstaande vergelijkingen gecodeerd

Berekeningen en code Arduino

Code

- ① Eerst alle variabelen toekennen en pins van de Arduino toekennen aan pomp, motoren, --

② Setup-functie

- pins gedeclareerd
- Variabelen gelijk stellen aan 0 (wan nodig)
- Serialconnectie gestart met pythoncode

③ Loopfunctie

- Elke seconde waterdebit + totaal volume gevoerd water berekend (mbr waterflowsensor)
- Startsignaal pythoncode nodig
 ⇒ horizontale motor draait
- Als nieuw signaal (\neq start) van python komt start een if-lus:
 - # branden + 1
 - Horizontale motor gestopt
 - Verticale motor draait ... rec afhang. w/d hoek θ
 - Pomp begint te pompen
- Nieuwe if-lus:
 - Zorgen voor juiste hoeveelheid gepassend water
 - Dus pomp dan uitzetten + water op 0 zetten
 - Verticale motor draait terug naar startpositie
- Sub-iflus:
 - Als # branden < 3 \rightarrow code herhalen
 - Anders code stoppen \rightarrow stilstaan

Berekeningen en code (camera)

Berekeningen

- Camera draait mee met draaiend platform (Noot erop)
- Berekening op 2 manieren

$$\textcircled{1} \quad d = f \cdot \frac{bw}{bs}$$

- bw = werkelijke breedte v/h object
- bs = breedte v/h object op het beeld v/d camera)

berekend door: # pixels in x-richting mal 0,00015833...

- f = focal lengte → eigen aan ieder toestel
- gewonden met deze formule met gekende afstand d tot het object

$$\textcircled{2} \rightarrow \text{hoek meten tss middelste pixel } pm \text{ en buitenste pixel } pb$$

- Deze uitkomst vermenigvuldigen met ggp (= graden/pixel)
 - ↳ $ggp = \frac{\text{breedte v/h gezichtsveld v/d camera} (=60^\circ)}{\text{breedte v/h beeld} (=640 \text{ pixels})}$

⇒ hoek α

$$\rightarrow d = \frac{bw}{\tan \alpha}$$

bw = werkelijke breedte object

- * Nu gemiddelde van ① en ② nemen \Rightarrow nauwkeuriger

Code

- beelden v. webcam analyseren in Python dankzij cv2
- beeld van BRG naar HSV
(Blue-Green-Red) (Hue-Saturation-Value)
- boven- en onderlimiet opleggen voor het kleur (rood v/d led)
- Output = zwart beeld met gevonden kleuren aangeduid
- Camera draait mee, dus al object in midden v/h beeld staat naast de afstandberekening (om stand dan te volgen)

Labs View