ICT-technology

# Specificaties

## Werkstukken

Goedgekeurd:

* Kleur 🡪 grijs & blauw
* Hoogte 🡪 50 & 54

Afgekeurd:

* Kleur 🡪 Zwart
* Hoogte 🡪 46 & 58

## Aanvoer werkstukken

* Werkstukken door operator met hand op aangeboden raster geplaatst
* Bij start programma wordt er gewacht op een signaal van de operator met een van de schakelaars.
* Er verschijnt een tweede boodscap en er wordt gewacht op een signaal van de operaterator met een tweede schakelaars

## Robotprogramma randvoorwaarden

* Manipulatie werkstukken op pallet 🡪 robot 1 punt aanleren en werken met offsets t.o.v van het punt
* Kleurmeting punt staat al vast
* Bewegingmodus 🡪 linear

## Meting

Er wordt in een rapport informatie over de batch gestoken

Informatie:

* Aantal van elke kleur
* Aantal van elke toegestaande hoogte
* Aantal afgekeurde blokken

# Routines

## Main()

Bij start-up worden alle persistent variables op 0 gezet, zodat de volgende batch kan ingelezen worden. waitOnSetup en startProgram worden opgeroepen. Er wordt over het grid gelooped, vervolgens wordt telkens de grab functie opgeroepen zodat elke blok gemeten en gesorteerd worden. Op het einde wordt alle informatie afgedrukt.

## Reset\_()

Alle persistent variables worden op 0 gezet, zodat de volgende batch kan ingeladen worden. Als we gewoon zouden overschrijven, zou het programma mid-run niet mogen stoppen, want anders heb je foutieve data.

## WaitOnSetup()

Er wordt een melding getoond dat de batch gevuld kan worden op de flexpendant en gewacht tot de eerste schakelaar aangezet wordt.

### Test strategie

Verwachting:

* Er komt een melding op de flexpendant dat de batch opgevuld kan worden en de rest van het programma wordt niet uitgevoerd tot de schakkelaar aangezet wordt.

Resultaat:

* Programma wordt niet uitgevoerd tot de schakelaar aangezet wordt.

## StartProgram()

Er wordt nogmaals een melding getoond op de flexpendant en gewacht tot de tweede schakelaar aangezet wordt. Vervolgens wordt de robot naar de begin positie gebracht.

### Test strategie

Verwachting:

* Er komt een melding op de flexpendant dat de tweede schakelaar aangezet moet worden en de rest van het programma wordt niet uitgevoerd tot de schakkelaar aangezet wordt. Vervolgens gaat de robot naar de eerste positie

Resultaat:

* Programma wordt niet uitgevoerd tot de schakelaar aangezet wordt.
* Als de schakelaar aangezet wordt, gaat de robot naar de eerste positie

## Grab(num x, num y)

Parameters:

* X 🡪 x-waarde van robot op de eerste positie
* Y 🡪 y-waarde van robot op de eerste positie

Eerst wordt de robot gepositioneerd boven de blok met een offset, vervolgens gaat de robot naar beneden en probeert de grootste blok dat toeglaten is, 58 cm, te grijpen met de zuignap. Als er iets gededecteerd wordt, wordt deze later afgekeurt. Mocht het kleinrer zijn dan 58cm dan gaat de robot naar de volgende positie van 54cm

Als DO\_ManoMeter\_Out1 0 is, dan gaan we ervan uit dat de blok 50 cm is en wordt het ventiel uitgezet. Dezelfde grijp procedure wordt uitgevoerd en als er nogsteeds niets gegrepen wordt, dan is het een blok dat afgekeurd is en slagen we de hoogte en present op 0.

Nu wordt measure\_color opgeroepen om de kleur te bepalen.

Op het einde, wordt de blok terug naar zijn orginele positie gebracht en neergezet. Het ventiel wordt uitgezet en er wordt 2 seconden gewacht, zo dat we zeker zijn dat de blok niet omvalt bij het neerzetten.

### Test strategie

Verwachting:

* De robot positioneerd zich boven de blok en gaat naar beneden om deze te grijpen
* Er wordt gecheckt of het een geldige hoogte heeft, deze hoogte wordt opgeslagen
  + Als het blokje kleinder is dan de toegestaande hoogte, wordt de hoogte en present op 0 gezet
* Er wordt terug naar de orginele posite gegaan en 2 seconden gewacht

Tussentijds Resultaat:

* Blokje viel om 🡪 Robot ging te snel en geen wait

Resultaat:

* hoogtes worden opgeslagen
* hoogtes worden afgekeurd

## Measure\_color(num x, num y, num offset)

Num offset 🡪 de z\_waarde om de blok terug te zetten

* Verschillende o.b.v de hoogte van de blok

Als er toch een blok gegerepen is, dan gaan we naar het kleurmeetpunt. Voor het punt wordt 3 seconden gewacht, de kleurensensor wordt dan uitgelezen en de kleur van de blok wordt opgeslagen.

Toegelaten kleuren:

* Rood 🡪 Q1
* Grijs 🡪 Q2
* Blauw 🡪 Q3

Als het Q4 is, dan gaan we ervan uit dat het zwart is, wat niet toegestaan is.

### Test strategie

Verwachtingen:

* Blokje wordt gebracht naar de kleurensensor
* Er wordt gecheckt of het een geldige kleur heeft, deze kleur wordt opgeslagen
  + Als het blokje zwart is dan wordt dit blokje afgekeurd

Tussentijds resultaat:

* De kleur wordt niet goed herkent 🡪 lag aan de kleurensensor
* Als we 5 seconden voor de kleurensensor wouden wachten, dan kwam er een error op de flexpendant dat dit niet ging 🡪 verandert naar 3 seconden

Resultaat:

* Toegestaande kleuren worden opgeslagen
* Niet toegestaande kleuren worden afgekeurd

## Validate()

Er wordt hier over de arrays: blockColor, blockHeight & blockPresent, gelooped om de waardes hier uit te halen en te bereken hoeveel er zijn afgekeurd en goedgekeurd. Er wordt ook nog weergegeven hoeveel blokken 50 en 54 cm zijn en bijhorende kleur.

### Test strategie

Verwachtingen:

* Alle variables worden met de correcte informatie opgevuld

Resultaat:

* Alle variables worden met de correcte hoeveelheid weergegeven

## OPC

We hebben de eind variables geconnecteerd met OPC, waardoor de gebruiker een overzicht krijgt over:

* Grootte
* Kleur
* Afgekeurd
* Goedgekeurd