Raspberry Pi 2 opdracht 1 – Transportband

Inhoud

[1 Opdracht 2](#_Toc64366630)

[2 Omschrijving 2](#_Toc64366631)

[2.1 Hardware 2](#_Toc64366632)

[2.2 Werking 2](#_Toc64366633)

[3 Opstelling 2](#_Toc64366634)

[3.1 GPIO 2](#_Toc64366635)

[3.2 Schema 3](#_Toc64366636)

[4 Software architectuur 4](#_Toc64366637)

# Opdracht

Maak een flexibele sturing voor een stappenmotor die je bedient met drukknoppen en een rotary encoder of potentiometer. (en eventueel joystick als extra optie)

* drukknop-1 : detecteren startpositie
* drukknop-2 : stuur motor traag naar links
* drukknop-3 : stuur motor traag naar rechts
* drukknop-4 : ga naar positie-1 of programmeer pos-1
* drukknop-5 : ga naar positie-2 of programmeer pos-2
* drukknop-6 : indrukken plaatst drukknop4+5 in programmeer-mode
* led-1 : start pos
* led-2 : pos-1 ingenomen
* led-3 : pos-2 ingenomen
* led-4 : program-mode is ON

GUI toont nuttige info bv : pos = start, pos1, pos2 of moving

Extra:

* GUI met drukknopen en feedback van de led ’s
* Rotary encoder of potentiometer (potentiometer met variabele snelheid)
* Aansturen via BTE

# Versies

V0.1.0 : Start

v1.0.0 : Update architecture

v1.0.1 : Update architecture

* remove hardware repos
* led and display -> observers

v1.0.2 : Update repository

# Omschrijving

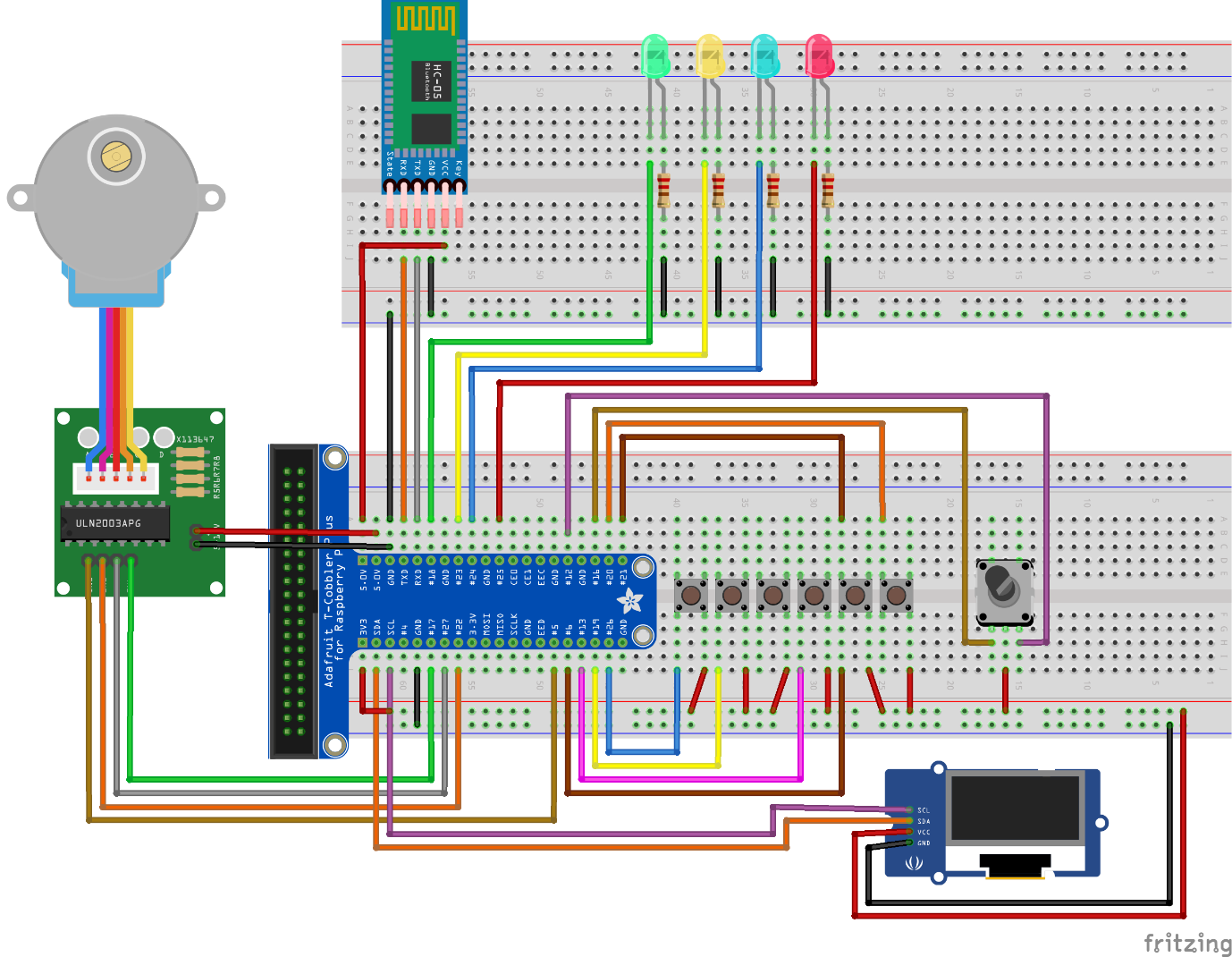
## Hardware

De gebruikte hardware :

* Raspberry Pi 4
* Drukknop (6x)
* Rotary encoder (1x)
* Led (4x)
* Weerstand 220 ohm (4x)
* Bluetooth module
* Stappenmotor
* Oled display

## Werking

### HMI





### Opstarten

Bij het opstarten van de applicatie zal de motor tegenwijzersin roteren tot de motor de home positie heeft bereikt. Deze positie wordt bereikt door drukknop 1 te bedienen.

### Verplaatsen stappenmotor

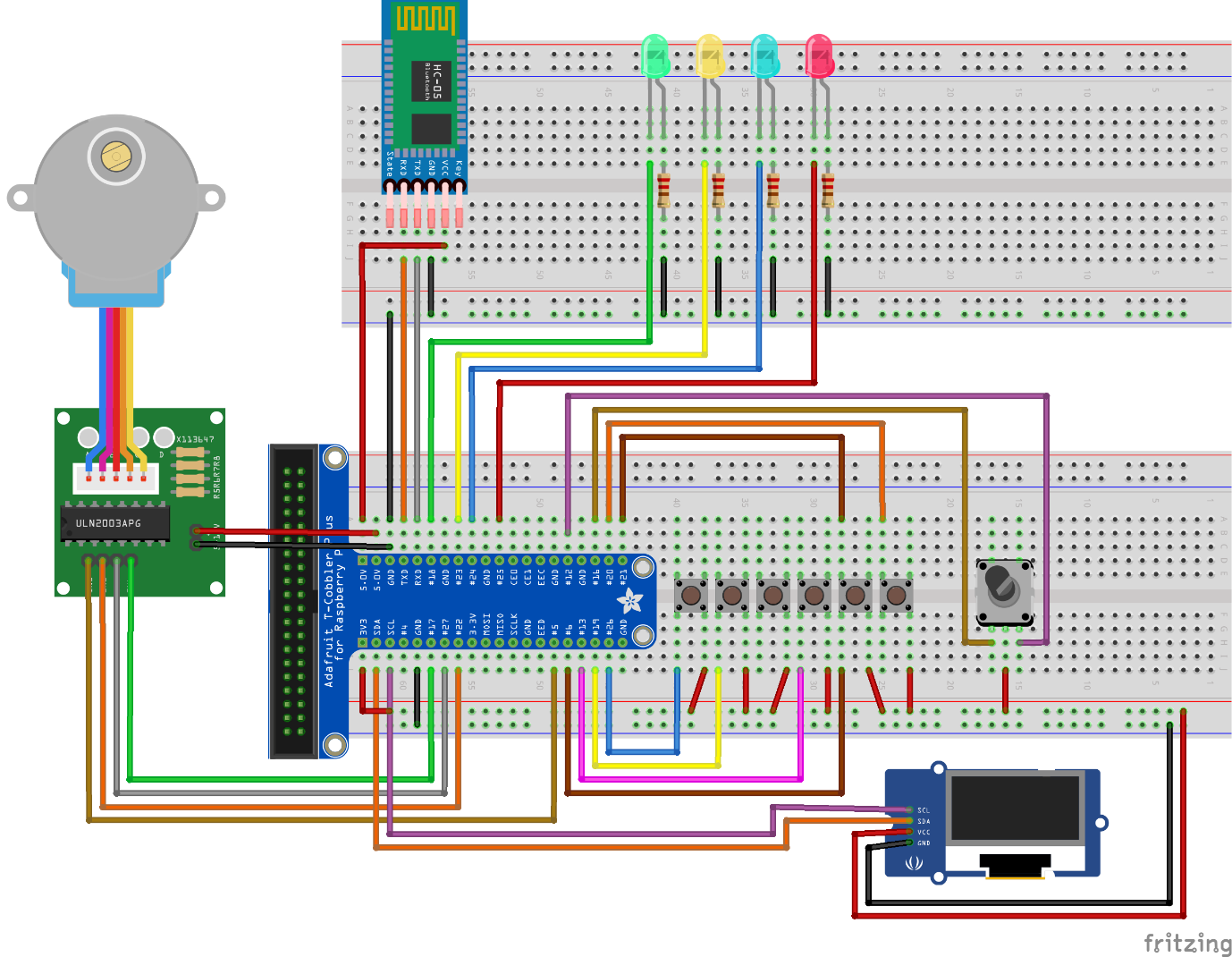
### Programmeren posities

# Opstelling

## GPIO

* Drukknop 1 : GPIO 18
* Drukknop 2 : GPIO 23
* Drukknop 3 : GPIO 25
* Drukknop 4 : GPIO 12
* Drukknop 5 : GPIO 16
* Drukknop 6 : GPIO 26
* Rotary A : GPIO 20
* Rotary B : GPIO 21
* Potentiometer : MCP 3008 channel 1
* MCP3008 :
  + CLK : GPIO 11 / SPIO SCLK
  + Dout : GPIO 9 / SPIO MISO
  + Din : GPIO 10 / SPIO MOSI
  + CS/Sh : GPIO 8 / SPIO CS1
* Led green : GPIO 5
* Led yellow : GPIO 6
* Led blue : GPIO 13
* Led red : GPIO 19
* Bluetooth module
  + RX : UARTO TX
  + TX : UARTO RX
* Stepper motor
  + In 1 : GPIO 17
  + In 2 : GPIO 27
  + In 3 : GPIO 24
  + In 4 : GPIO 22

## Schema

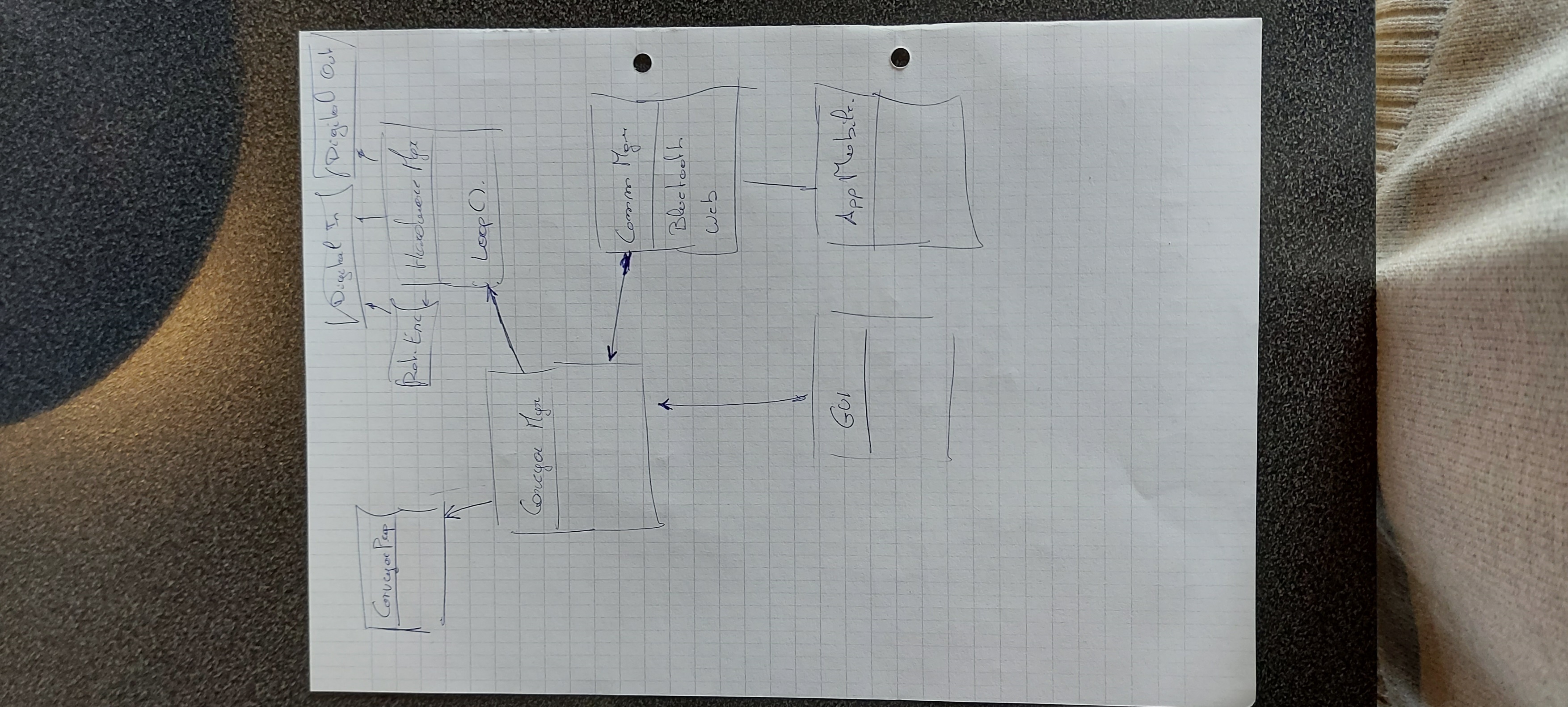


Figuur 1: Schema

# Software architectuur

## V0.1.X

Om de software op te splitsen in logische blokken heb ik op een kladblad een schema getekend. Dit schema is niet volledig en dient enkel om de gedachten te ordenen.



Figuur 2: kladversie software architectuur

## V1.0.X

In v1.0.X heb ik de architectuur aangepast met een MVC (model-view-controller) pattern. Dat houd in dat de interface met de machine, drukknoppen, leds, oled, … gezien worden als frontend van de applicatie en niet meer als service. Dit heeft als voordeel dat controller (het bepalen wat er met de drukknoppen gebeurd) een abstracte basis klasse kan zijn die door andere frontends gedeeld kan worden.