Raspberry Pi 2 opdracht 3 – Robot/Car

Inhoud

[1 Opdracht 2](#_Toc66109366)

[2 Versies 2](#_Toc66109367)

[3 Omschrijving 3](#_Toc66109368)

[3.1 Hardware 3](#_Toc66109369)

[3.2 Werking 3](#_Toc66109370)

[3.2.1 HMI 3](#_Toc66109371)

[3.2.2 Opstarten 5](#_Toc66109372)

[3.2.3 Programmeren posities 5](#_Toc66109373)

[4 Opstelling 6](#_Toc66109374)

[4.1 GPIO 6](#_Toc66109375)

[4.2 Schema 7](#_Toc66109376)

[5 Software architectuur 8](#_Toc66109377)

[5.1 V0.1.X 8](#_Toc66109378)

[5.2 V1.0.X 9](#_Toc66109379)

# 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

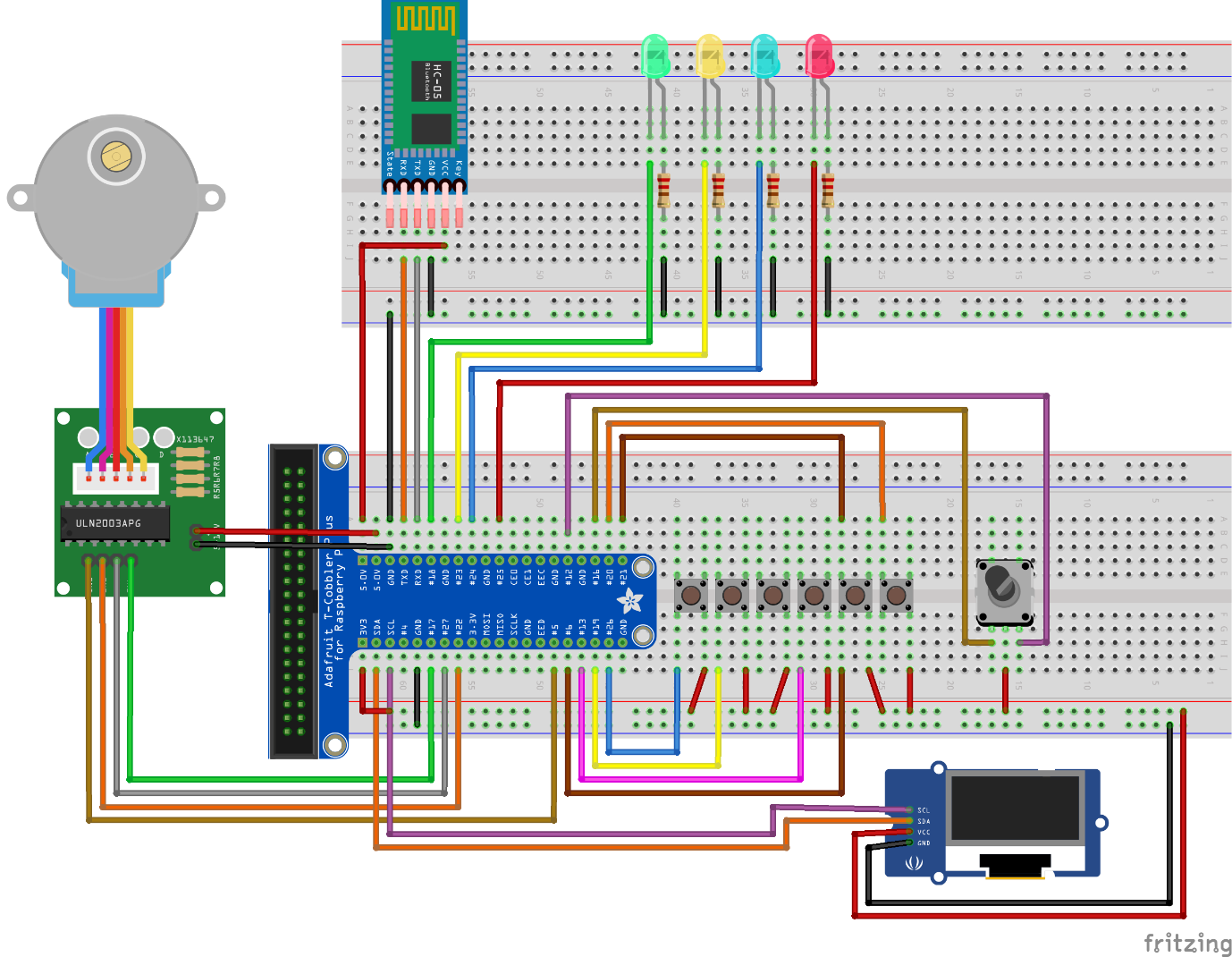
### HMI

#### Hardware

De hardware interface bevat 6 drukknoppen, een rotarty encoder en 4 led ’s . De linkse drukknop is drukknop 1 de rechtse is drukknop 6.

Drukknoppen:

* Drukknop 1 wordt gebruikt als simulatie van de eindeloopsensor.
* Drukknop 2 : verplaatst de motor 1 stap tegenwijzersin
* Drukknop 3 : verplaatst de motor 1 stap wijzersin
* Drukknop 4 : verplaatst de motor naar positie 1 (default 50 stappen wijzersin vanaf de home positie)
* Drukknop 5 : verplaatst de motor naar positie 2 (default 100 stappen wijzersin vanaf de home positie)
* Drukknop 6 : porgrammeren van de posities
* Rotary encoder : 1 stap van de rotary encoder verplaatst de motor in dezelfde richting 1 stap



Figuur 1: overzicht drukknoppen

Led ‘s:

* Groen: Bij het homen en verplaatsen van de motor naar de home positie knippert deze led. Als de motor op de home positie is brand de groene led constang.
* Geel : Led voor positie 1, als de motor op positie 1 staat brand deze constant, bij het verplaatsen naar de positie knippert de led.
* Blauw : Led voor positie 2, als de motor op positie 2 staat brand deze constant, bij het verplaatsen naar de positie knippert de led.
* Rood : Status led voor het programmeren van de positie. Bij het selecteren van de programma functie knippert de rode led. Als er dan een positie wordt geselecteerd om te programmeren (drukknop 4 of 5) brand de rode led constant en knippert de positie led.

#### Tkinter desktop



Figuur 2: hmi desktop

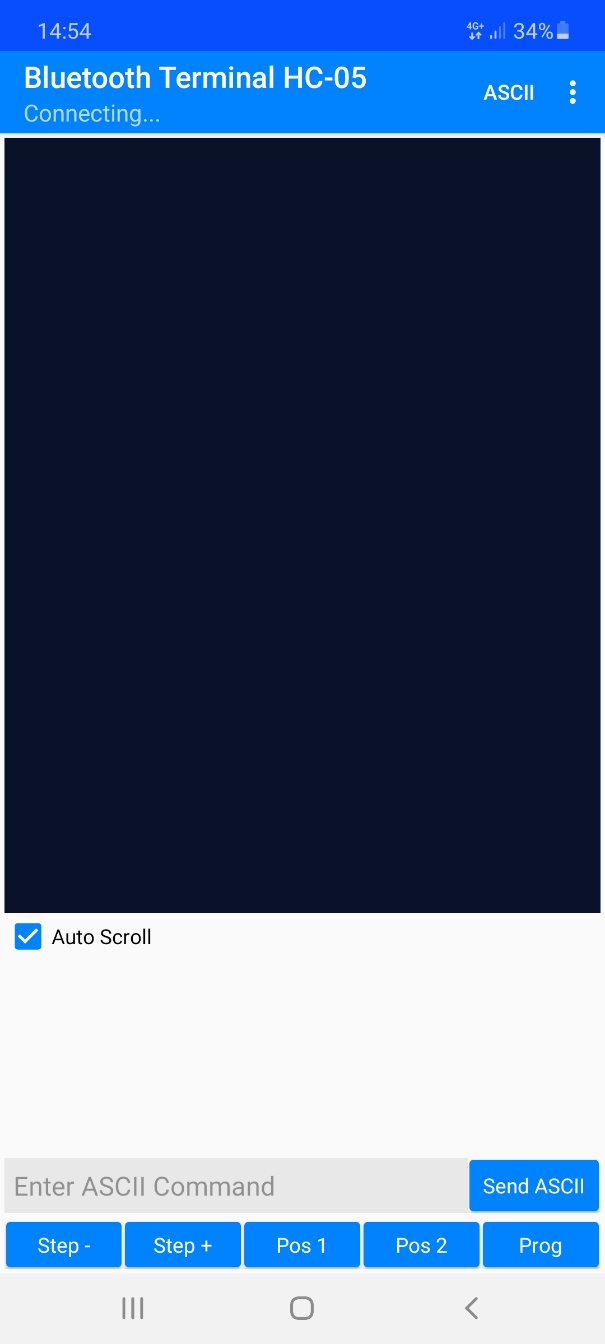
De desktop hmi is opgedeeld in 3 onderdelen, State, Control en Settings.

* Conveyor State: Hierin wordt de status van de motor, de positie van de slede, het aantal stappen van de motor ten opzichte van de home positie en een boodschap getoond.
* Conveyor Control: Met deze knoppen is het mogelijk de slede te sturen.
  + Home : Met deze knop kan de slede naar de home positie worden verplaatst (deze is niet gelijk aan de homing sensor!)
  + <- 1: verzet de motor 1 stap tegenwijzersin
  + Position 1 : verplaatst de slede naar positie 1
  + Position 2 : verplaatst de slede naar positie 2
  + 1 -> : verzet de motor 1 stap wijzersin
* Conveyor Settings: Hierin kan het aantal stappen ten opzichte van de home positie gewijzigd en opgeslagen worden in de settings. Ook kunnen hier de default settings terug gezet worden.

#### Oled

Op het oled scherm worden de state, positie, aantal stappen van de home positie en een boodschap weergegeven. Gelijk aan de Conveyor state van de desktop hmi.

#### Bluetooth



Figuur 3: hmi bluetooth

De knoppen van de bluetooth hebben dezelfde functie als de hardware buttons. In het scherm van de bluetooth applicatie worden de boodschappen getoond.

### 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.

### Programmeren posities

De posities kunnen gewijzigd worden in de desktop hmi door het aantal stappen te wijzigen en op te slaan.

De posities kunnen ook gewijzigd worden door:

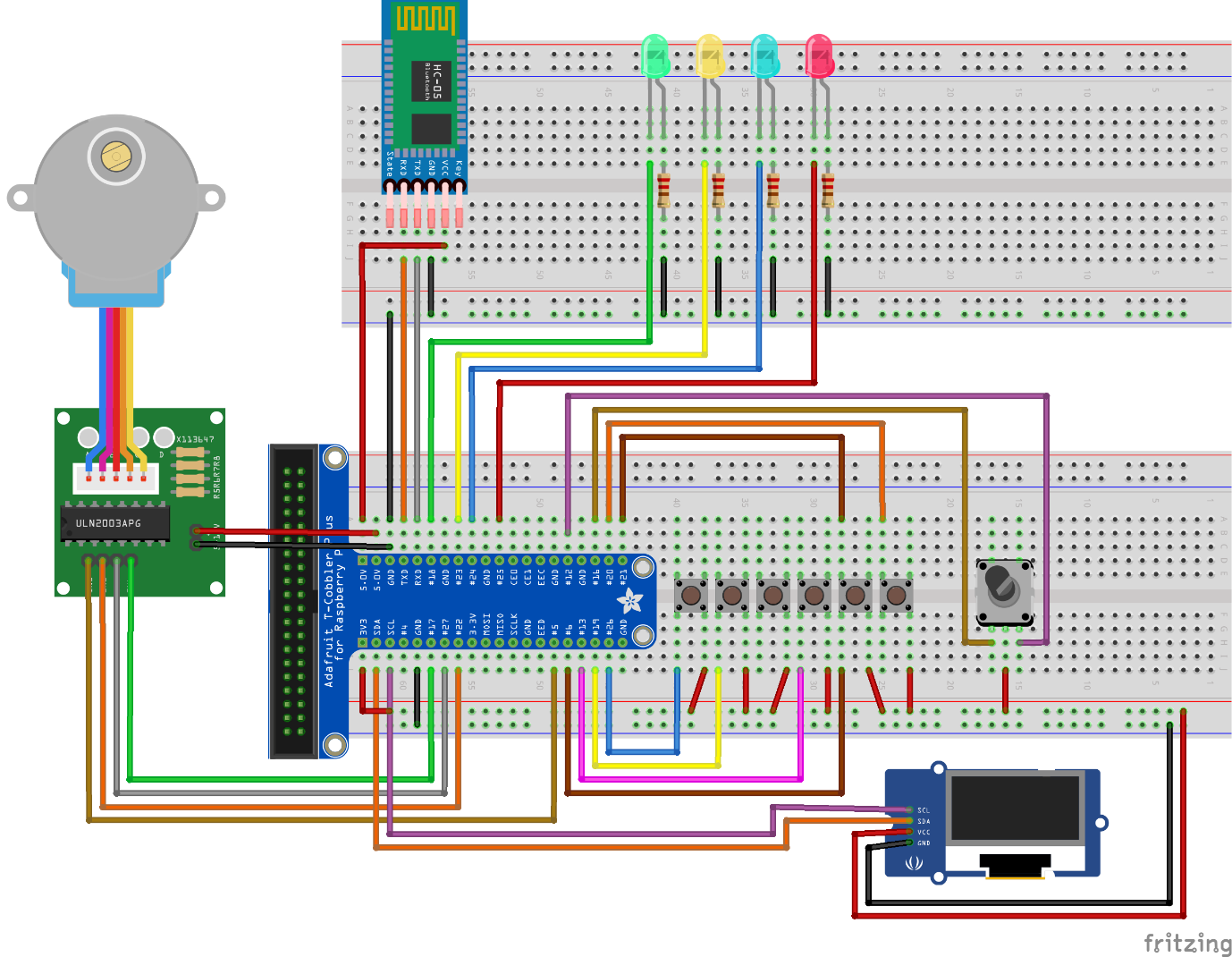
1. knop 6 van de hardware of de “Prog” knop van de bluetooth module.
2. De rode led gaat knipperen.
3. Selecteren van de te wijzigen positie
4. De rode led gaat constant branden en de led van de positie die gewijzigd wordt gaat knipperen.
5. Slede verplaatsen naar de juiste met de drukknoppen 2 / 3 of in bluetooth hmi Step - / Step +
6. Terug de positie knop selecteren, de positie wordt opgeslagen in de settings.txt
7. De rode led gaat terug knipperen
8. Drukknop 6 of Prog knop bedienen om de programmeren functie te verlaten.

# 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
* 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
* OLed display
  + Sla : scla
  + Sll : scll

## Schema

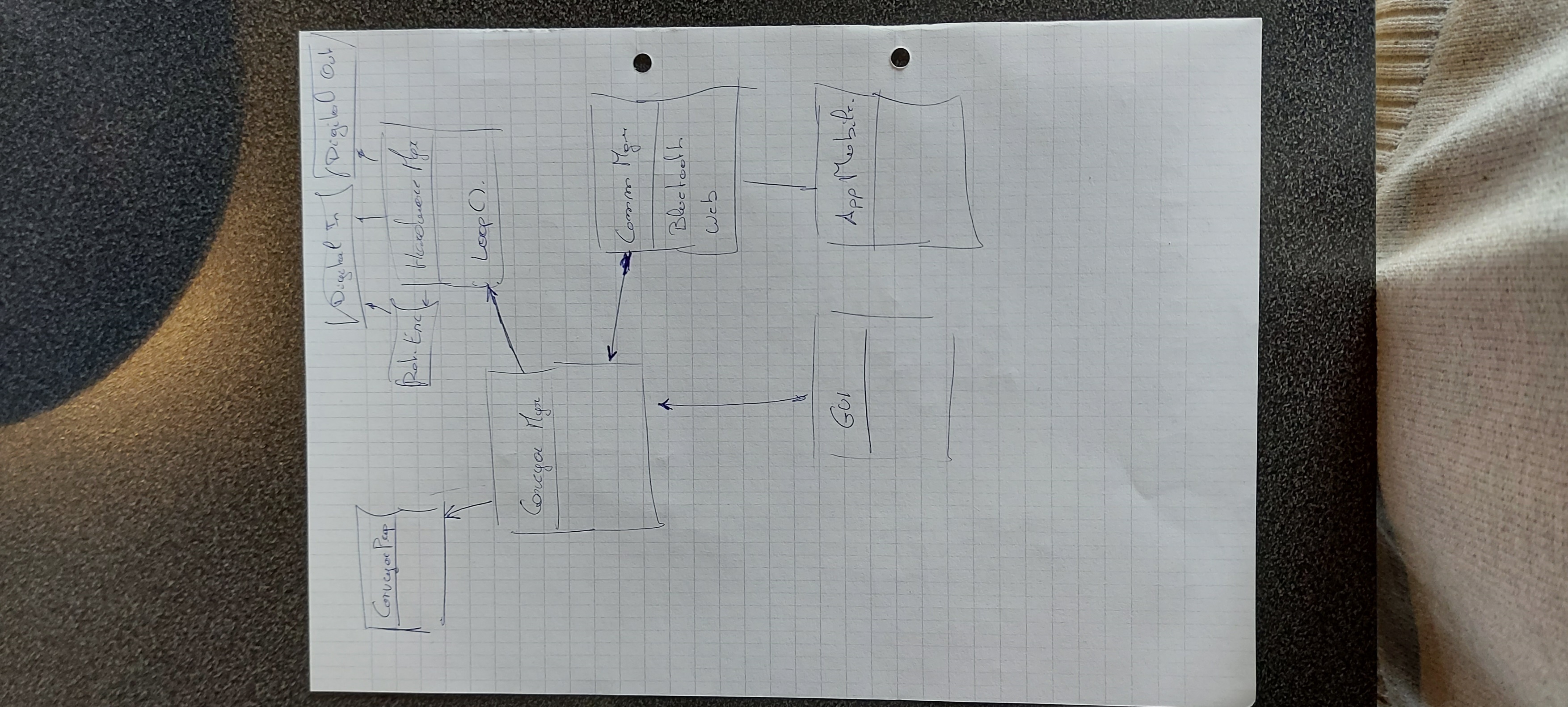


Figuur 4: Schema

# 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 5: 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.