Poseidon

# The project

We need to develop a way to control light and pumps on terminal places

# What are the requirements (pr 2022-08-25)

## Outputs

There is a need for 9 outputs all relay

4 for controlling pumps

4 for controlling red/green lights

1 for resetting the E-stop

## Inputs

There is a need for 12 input all switchable

We need to connect to Uptime for reading inputs and controlling the outputs

This system must run on its own like a second layer of security, so we will not be able to connect to existing Sapphire terminal. So this is a parallel system to the existing Sapphire terminal.

# Solution for a prototype

We suggest that we use a relay board with 16 relays and 16 inputs.

We found one where we can change the esp32 microcontroller, so we can put in our own firmware (the one mounted on the board is locked so we can’t reprogram it)

Price for this board is 580dkk + import tax

We can control this board in many ways, but we suggest that we control it via TTL 3v3.

We need to make our own protocol on how to communicate between uptime and the relay board

Firmware for the ESP32 MCU, will take approx. 5 days (In high level languish, Arduino, by Jesper)

We suggest a simple communication protocol like this:

Start byte, Data length, To Terminal Address, Data….., Stop byte

And the respond like this

Start byte, Data length, From Terminal Address, Data….., Stop byte

# Send to relay board

## Send relay data like this: (To relay board)

Relay1, Relay2, ……. Relay15, Relay16, Input1, Input2, …… Input15, Input16 (32 bytes)

## Send to relay byte can be like this

Bit Name Values

7..4 DelayTime 0-15 milli second

2..3 DelayTime multiplier 00=\*10, 01=\*100, 10=\*1000, 11=\*10000

1..0 Function (00=off, 01=on, 10=on time, 11 blink time)

So if you want the relay to be off, then you set the to bit (1,0) to 00, and the relay turns off.

If you want the relay to blink every second then you set bit (7-4) to 1010, bit (2,3) to 01, bit (1,0) to 11, and the relay will start blinking every second

Like this:

0B1010 10 11 = blink with a delay of 1000 millisecond

## Send input byte

The input byte is not used for now and will be ignored, but is then ready for future functions like how often to measure inputs.

# Response from the relay board

## And the response like this (Input data)

Relay1, Relay2, ……. Relay15, Relay16, Input1, Input2, …… Input15, Input16 (32 bytes)

## Relay byte, the current state of the relay

Bit Name Values

7..1 N/A There is space for new functionality here

0 Current State (00=off, 01=on)

## Input byte, the current state of the input

Bit Name Values

7..1 N/A There is space for new functionality here

0 Current State (00=off, 01=on)

We suggest that we use the Advantech board with an RS232 to TTL3v3 adaptor price 590dkk

<https://www.digikey.dk/da/products/detail/advantech-corporation/BB-232LPTTL33/3770643>

If we can’t use the Advantech board then we can use any pc available with a USB to TTL cable, but the other solution is to prefer because the software has been running on this for a long time, so we know it works.

The Advantech board can be connected to a Cellular router like the ones we already use.

The software on the Advantech board will then connect to optime to ask for data, optime responds with “send data to relay board” and gets a “Response from the relay board”

There must be made some modifications to part of the software on the Advantech board, this will take approx. 38 hours (by Jacob)

# So the BOM will look like this:

Relay board 590dkk

Esp32 MCU 30dkk

Rs232 to TTL3v3 600dkk

Advantech board in house

12v power supply in house

Gsm modem in house

Box for the build ?

Div. 100dkk

Programming firmware JM 40 hours

Programming software JJ 38 hours

Programming Optime ?

Build time 20 hours

Support documentation 20 hours