GPIO Challenge

Tiemon Steeghs

Inhoud

[1. Inleiding 3](#_Toc126832780)

[2. Onderzoek 4](#_Toc126832781)

[2.1. Hoofdvraag 4](#_Toc126832782)

[2.2. Deelvraag 1 4](#_Toc126832783)

[2.3. Deelvraag 2 4](#_Toc126832784)

[2.4. Deelvraag 3 4](#_Toc126832785)

[3. Het Design 5](#_Toc126832786)

[4. Testen 6](#_Toc126832787)

[5. Conclusie 7](#_Toc126832788)

# Aanleiding

Ik maak deze opdracht om nieuwe kennis op te doen over het bewerken van registers. In deze opdracht maak ik een eigen Manual Control Panel (MCP). De MCP bestaat uit twee knoppen en twee led’s. De MCP moet voldoen aan verschillende requirements. Om de opdracht te realiseren ga ik uitgebreid onderzoek doen over de verschillende onderdelen die aanbod komen.

De requirements waaraan het project moet voldoen zijn hieronder te vinden:

**Functional**  
1. BA Only: (M) Short and long presses on the buttons or left and right turns of the  
rotary encoder must be distinguishable by the LED lighting pattern.  
2. BA Only: (M) Short and long presses on the buttons or left and right turns of the  
rotary encoder must be distinguishable on the serial monitor.

**Non-functional**  
1. (M) The MCP must function properly with buttons or a rotary encoder with a  
bouncing time of less than 20 ms. When using capacitors this requirement is fulfilled.  
2. (M) Button presses or rotary encoder turns must never be missed i.e. detecting  
presses or turns must be done without polling.  
3. (M) Your solution must be scalable i.e. adding more buttons or LEDs should be easy.  
4. (S) When no button is pressed or the rotary encoder is not turned, the power  
consumption of the MCP must be minimal.

## Probleemstelling

Het aansturen van bijvoorbeeld led’s deed ik voorheen altijd met al bestaande methodes. Het nadeel van deze methodes is dat ze slomer zijn dan een handmatige bewerking van de bits. In dit onderzoek ga ik dit probleem oplossen door een zo efficient mogelijk manier te maken om in dit geval led’s aan te sturen.

# Onderzoek

## Hoofdvraag

**Hoe kan je aan de hand van registers een led aansturen met een knop op het Stm32 Nucleo bord?**

## Deelvraag 1

**Wat is een hardware register?**

Hardware registers maken het mogelijk om doormiddel van software data te schrijven en op te vragen van een apparaat. Het register heeft een bepaalde opslag met een adres. Door op dit adres te werk te gaan kan je dus verschillende eigenschappen van het apparaat bewerken.

## Deelvraag 2

**Welke ports moet ik gebruiken op het Stm32 Nucleo bord voor het aansturen van twee led’s en knoppen?**

Op het Stm32 Nucleo bord zijn veel verschillende ports aanwezig. Als je de user manual raadpleegt kan je een port layout vinden. Hierop is te zien welke ports je kan gebruiken voor het aansturen van de knoppen en led’s. In de bronnenlijst van dit document is deze layout terug te vinden.

## Deelvraag 3

**Hoe kan je codematig een register bewerken?**

Een register is altijd opgeslagen in een stuk geheugen. Door op het adres van het register bits aan te passen kan je het register bewerken en dus bijvoorbeeld een pin aflezen. In code zou dit beteken dat je bit operators gaat gebruiken. Met de OR operator | kan je makkelijk bits aanzetten. Hiermee zou ik dus een gewenst bit aan kunnen zetten.

# Het Design

# Testen

# Conclusie

# Bronvermelding

Wikipedia hardware register:

<https://en.wikipedia.org/wiki/Hardware_register>

Hardware register op academickids:

<http://academickids.com/encyclopedia/index.php/Hardware_register>

Stm32 nucleo pinindeling:

<https://microcontrollerslab.com/stm32-nucleo-board-pinout-features-applications-programming-peripherals/>

Stm32 nucleo user manual:

<https://www.st.com/resource/en/user_manual/um1724-stm32-nucleo64-boards-mb1136-stmicroelectronics.pdf>

Stm32 nucleo driver:

<https://www.st.com/en/development-tools/stsw-link009.html#overview>