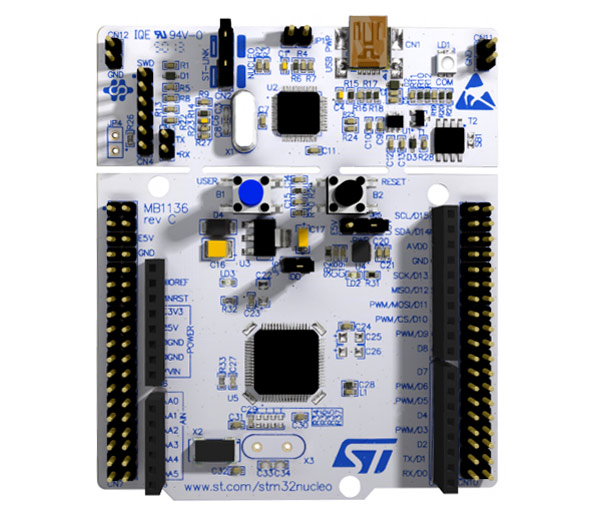
GPIO Challenge

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 waarbij button debounce wordt toegepast?**

## 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. In principe kan je alle ports gebruiken die input en output ondersteunen.

## 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.

Om dit op de proef te zetten heb ik eerst simpele code gemaakt die ervoor zorgt dat ik een led aan kan zetten met een knop zonder nog gebruik te maken van interrupts of timers.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

In deze code word de status van de knop steeds bekeken. Op het moment dat de knop is ingedrukt zal hij de led aanzetten.

## Deelvraag 4

**Hoe implementeer je button debounce op het Stm32 Nucleo bord zonder libraries te gebruiken?**

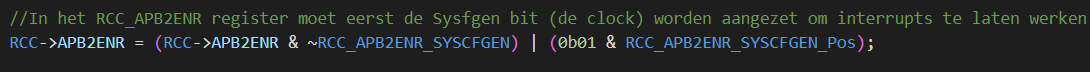
Een button debounce kan je implementeren aan de hand van verschillende variabelen. Door de vorige staat van de button steeds bij te houden, kan je ervoor zorgen dat bijvoorbeeld een lampje pas aangaat als de knop weer opnieuw wordt ingedrukt. Al zou je de knop ingedrukt houden zal de code maar één keer worden uitgevoerd omdat bij de tweede keer de vorige staat van de button hetzelfde is.

## Deelvraag 5

**Hoe kan je interrupts instellen en gebruiken op het Stm32 Nucleo bord?**

Het Stm32 Nucleo bord heeft speciale support voor interrupts, hiervoor moet je wel verschillende dingen instellen voordat je ze daadwerkelijk kan toepassen.

Als eerste moet je bij het RCC\_APB2ENR register de SYSCFGEN bit aanzetten om de clock voor het interrupt register te activeren.



Vervolgens kan je bij het SYSCFG register van het bord instellen welke pins, interruptpins moeten zijn. In mijn geval gebruik ik PINA8 als interrupt pin, dus ik moet hiervoor in EXTICR3 zijn. (in code zijn deze registers geplaatst in een array, ik gebruik daarom EXTICR[2])



Hierna komen nog twee andere bits die moeten worden aangezet om interrupts werkende te krijgen.

EXTI\_IMR\_MR8 is een bit in het EXTI\_IMR register die bepaalt of EXTI lijn 8 is ingeschakeld voor interrupts of niet. De regel EXTI->IMR |= EXTI\_IMR\_MR8; schakelt de interrupt in voor EXTI lijn 8 door de betreffende bit te zetten in de EXTI\_IMR-register.

EXTI\_FTSR\_TR8 is een bit in de EXTI\_FTSR register die bepaalt of de EXTI-controller moet reageren op een falling edge of een rising edge van de pin die verbonden is met EXTI lijn 8. De regel EXTI->FTSR |= EXTI\_FTSR\_TR8; stelt de EXTI-controller in om te reageren op een falling edge, door de betreffende bit te zetten in de EXTI\_FTSR-register.

Afbeelding met tekst, klok

Automatisch gegenereerde beschrijving

Daarna moet je de functie NVIC\_EnableIRQ() gebruiken om de interrupt voor een specifieke NVIC IRQ lijn in te schakelen. In mijn geval, NVIC\_EnableIRQ(EXTI9\_5\_IRQn) schakelt de interrupt in voor de NVIC IRQ lijn van EXTI lijnen 5 tot en met 9. Dit is nodig omdat de EXTI lijn 8 waarop de interrupt zal plaatsvinden, een van de lijnen is binnen deze range. Ik gebruik namelijk PINA8



Als je vervolgens dan een button bijvoorbeeld aansluit op zo’n pin kan je met de interrupt te werk gaan door middel van een interrupthandeler te schrijven. Dit is een methode die wordt uitgevoerd op het moment dat de interrupt optreed.

## Deelvraag 6

**Hoe implementeer je een timer op de STM32 Nucleo om bij te houden hoelang een knop is ingedrukt?**

Bij deze timer mag ik nog gebruik maken van HAL functies. Door deze functies toe te passen op de volgende manier kan ik bijhouden hoelang de knop is ingedrukt: Op het moment dat de knop wordt ingedrukt wordt de interrupt getriggered, die tijd opgeslagen in de “RisignEdgeMilliesTime” variabel. Op het moment dat de knop wordt losgelaten wordt de interrupt opnieuw getriggered en zal het tweede gedeelte van de if-statement van kracht gaan. Hier wordt het verschil berekent tussen het moment dat de knop wordt ingedrukt (Rising edge) en wanneer hij wordt losgelaten (Falling edge). Dit verschil is hoelang de knop is ingedrukt.

Afbeelding met tekst, scherm, overdekt, schermopname

Automatisch gegenereerde beschrijving

# Het Design

# 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>

ChatGPT voor vragen over code:

<https://chat.openai.com>

De Interrupts presentatie van de course based github van semester 3.

