**Assessment Microcontrollerprogrammeren 2024**

Datum: 8 april 2024

Tijd: 2 uur + 30 min SMF Cijfer: \_\_\_\_\_\_\_\_\_\_

**Toegestane hulpmiddelen**: alle materialen MicroChip, sheets, voorbeeldcode van MicroChip, MPLAB IDE ,MCC, logboek, eigen code, Curiousity Nano AVR128DB48, SMU-bordje, breadboard met bedrading

**NIET Toegestane hulpmiddelen**: Je mag geen gebruik maken van internet en geen gebruik maken van large language models (LLM’s) zoals chat-gpt of copilot.

Algemene Instructie:

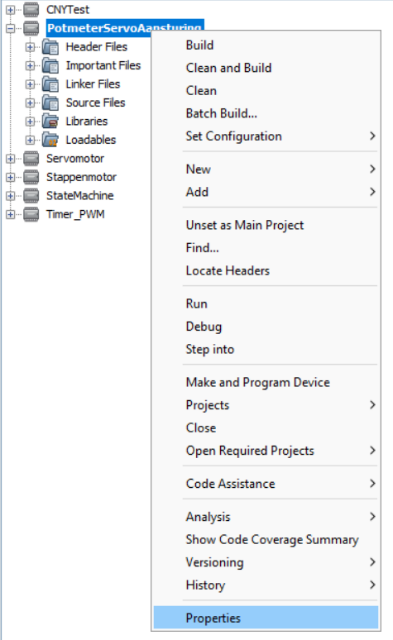
* Dit assessment bevat 4 vragen en heeft 5 pagina’s. Waarschuw de docent wanneer een van deze aantallen niet klopt.
* Becijfering is aangegeven bij de opdracht.
* Je krijgt per vraag 1, 2 of 3 punten. Er zijn in totaal 10 punten te behalen. Je cijfer is gelijk aan het aantal behaalde punten. Het laagst mogelijke cijfer is een 1. Het cijfer telt alleen wanneer er voldaan is aan de practicumverplichting.
* De opdrachten mogen in willekeurige volgorde gemaakt worden. De complexiteit van elke opdracht varieert. Je maakt 1 nieuw project waarin alle code staat. Code van opdrachten die al eerder gemaakt zijn, mag dus behouden blijven en hoeft niet weg gecommenteerd te worden.
* Als je een opdracht geheel af hebt, dan geef je de docent een demo en laat je de code zien. Je doet dit zoveel mogelijk zonder praten en beantwoordt de vragen van de docent. De docent tekent dan de opdracht af, of geeft een aanwijzing waarop de opdracht nog niet voldoet.

Begininstructie

* Maak een nieuw project aan met de naam: assessment\_dagdeel\_snummer. Voorbeeld: assessment\_ochtend\_1234567

Inleverinstructie (aan het eind van het assessment):

1. Rechtermuisknop op je project.
2. Ga naar **Properties** (onderaan het dropdownmenu)
3. Klik op **General** (bovenaan)
4. Kopieer de **Project Location** in je verkenner (windowstoets + e)
5. Selecteer alle bestanden in deze map.
6. Rechtermuisknop
7. Selecteer **Kopiëren naar**
8. **Gecomprimeerde map** (.zip bestand)
9. Noem dit bestand: assessment\_snummer (bijvoorbeeld assessment\_1234567)
10. Lever het .zip bestand in op Brightspace

 Afbeelding met tekst, schermopname, software, scherm

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

# Opdracht 1 (2 punten) Knopjes en Ledje

In deze opdracht ga je laten zien dat je m.b.v. twee knopjes een ledje kan besturen.

1. Stel SW0 (PB2) en SW1 (PE3) in als knop. Zorg ervoor dat pullup resistor aan staat en er een interrupt gegenereerd wordt op de falling edge voor beide knoppen.
2. Stel LED0 in als output
3. Stel de Global Interupt Enable correct in
4. Zet de LED aan in de ISR van SW0
5. Zet de LED uit in de ISR van SW1

Bij een correct geprogrammeerde opdracht zal de LED aan gaan bij het drukken op SW0 en uitgaan bij het drukken op SW1.

# Opdracht 2 (2 punten) UART verbinding en Data Visualiser

Demonstreer dat een bericht via UART verstuurd kan worden naar je laptop. We doen dit als volgt.

1. Stuur een UART bericht met eerste 3 cijfers van je s-nummer naar de laptop. Bijvoorbeeld s-1234567 verstuurt 123.
2. Je mag zelf bepalen of je dit bericht continu verstuurt, of dat dit gebeurt na het indrukken van de knop (zie opdracht 1).
3. Demonstreer dat je het bericht binnen krijgt d.m.v de data visualiser.

# Opdracht 3 (1, 2 of 3 punten) ADC

We gaan een ADC programmeren. Deze ADC moet met behulp van een overflow van de RTC periodiek een sample nemen. Je kunt bij deze opdracht 1, 2 of 3 punten halen. Je krijgt 1 punt voor het correct instellen van de RTC (samplerate), 1 punt voor het correct instellen van de ADC en VREF en 1 punt voor het correct uitlezen van de waarde m.b.v. watch expression of via de UART.  
  
**Voor het correct instellen van de RTC krijg je +1 punt**

1. Stel de RTC zo in dat deze een periode heeft van 0,05 seconden. We samplen dus met een frequentie van 20 Hz. (wanneer dit niet lukt mag je de RTC in standaard instellingen laten staan en doorgaan met de volgende deelopdrachten)

**Voor het correct programmeren van de volgende stappen krijg je +1 punt**

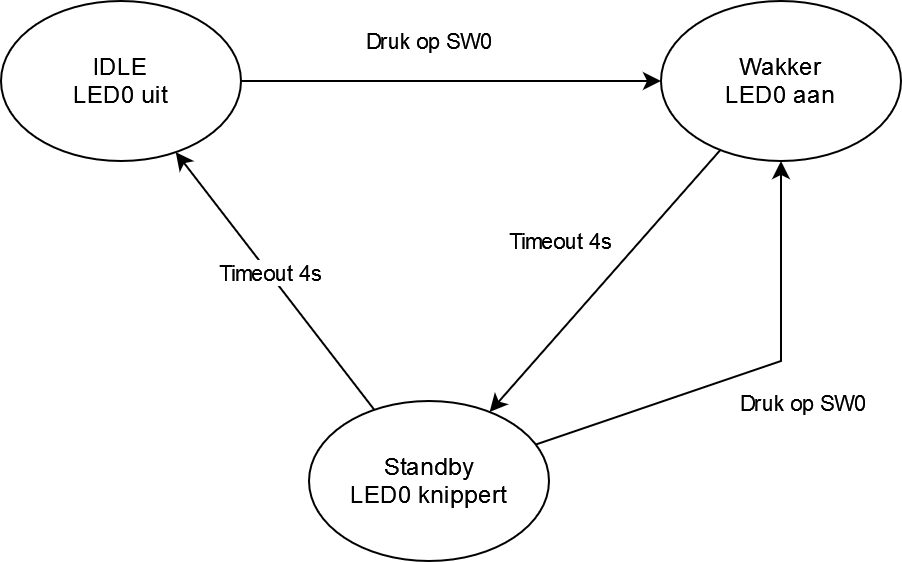
1. Stel de ADC zo in dat deze samples kan nemen op PD4. Laat de ADC een interrupt genereren wanneer deze een sample klaar heeft.
2. Stel VREF op 2.048V in voor de ADC.
3. Koppel de RTC via het event system aan de ADC. Wanneer dit niet lukt, mag je de RTC ook een interrupt laten genereren en via een ISR van de RTC de ADC softwarematig starten.
4. Kopieer het sample dat de ADC genomen heeft in de ISR (van de ADC) over naar een eigen variabele. Deze variabele heeft de naam: adc\_var\_snummer (bijvoorbeeld adc\_var\_1234567).

**Voor het correct laten zien van de ADC waarde krijg je +1 punt**

1. Voeg een breakpoint toe op de plaats van toekenning (dus in je ISR)
2. Run je code in debug mode en laat mbv één van volgende methodes de waarde zien:
   1. Plaats een watch expression op adc\_var\_snummer en laat d.m.v. een breakpoint i.c.m. watch expression zien wat de waarde van de variabele is.
   2. Wanneer een watch expression niet lukt, mag de ADC waarde ook verstuurd worden via de UART verbinding naar de data visualiser (zie opdracht 2).

# Opdracht 4 (1, 2 of 3 punten) State machine met Timer

Programmeer een statemachine. Deze statemachine werkt als volgt:



De statemachine heeft 3 states:   
S\_Idle LED0 staat uit  
S\_Wakker LED0 staat aan   
S\_Standby LED0 knippert, (met een zelfgekozen waarneembare frequentie)

Er zijn 3 events:

E\_druk Na het drukken op SW0 komt dit event  
E\_Timeout Na 4 seconde komt er een timeout event  
E\_no\_event Wanneer bovenstaande events niet zijn opgetreden, is dit het standaard event

**Voor 1 punt (zonder ISR en timer):**Je systeem doorloopt de statemachine continu. Deze zal elke keer checken wat de state is en de bijbehorende actie uitvoeren. Daarnaast wordt elke keer het event gecheckt om de state te updaten. De timeout mag geprogrammeerd worden als een \_delay\_ms(4000). Het knipperen mag gedaan worden door de LED afwisselend aan en uit te laten gaan met een delay van 250 ms. (16x250ms = 4000ms = 4s). De waarde van het knopje mag bepaald worden d.m.v. polling.

**Voor 2 punten (met ISR en timer)**Je systeem doorloopt de statemachine continu. Deze zal elke keer checken wat de state is en de bijbehorende actie uitvoeren. Daarnaast wordt elke keer het event gecheckt om de state te updaten. De timeout wordt nu gecreëerd door een overflow op TCA0, het knipperen wordt gedaan dmv een langzame PWM op TCA1. Het knopje werkt nu m.b.v. een ISR. In de ISR voor SW0 wordt het event aangepast naar E\_druk. In de ISR van TCA0\_OVF\_vect (overflow TCA0) wordt het event aangepast naar E\_Timeout.

**Voor 3 punten (met sleepmode)**  
Laat je systeem slapen in idle mode. Door een ISR wordt het systeem gewekt. Nu blijft het systeem de statemachine doorlopen totdat het weer in de idle state komt.

**Einde Assessment**