

# Studiehandleiding (uC)

## Microcontrollerprogrammeren

Studiejaar 2024 – 2025

Leerjaar	1
Periode	3
Vakcode	EDPR.22 pa1 (EDPR.22)
Aantal studiepunten	5 (samen met PLC-programmeren)
Opgesteld door	Douwe Schotanus en Marco Winkelman
Docenten	Douwe Schotanus
Laatste wijziging	3 februari 2025

### Doelstelling

Het aanleren van basisvaardigheden op het gebied van microcontrollerprogrammeren. Dit wordt beschreven met de volgende leeruitkomsten:

“De student ontwerpt en realiseert software dat kan executeren op een microcontroller, waarbij rekening wordt gehouden met de in de microcontroller aanwezige architectuur. Er wordt in de software gebruik gemaakt van toestanden, timing en interrupts. Hiermee kan een eenvoudige autonome sturing worden opgebouwd, waarbij gebruik wordt gemaakt van de microcontroller, sensoren en actuatoren.”

### Ontwerpen

- student kan een eenvoudig programma voor een microcontroller ontwerpen waarbij een sturing met tijd- en toestandsafhankelijke beslissingen kan worden gerealiseerd middels de in de microcontroller aanwezige architectuur
- student kan een programma ontwerpen waarmee specifieke analoge en digitale sensoren kunnen worden uitgelezen en actuatoren kunnen worden aangestuurd via de in de microcontroller aanwezige interfaces
- student kan de opbouw en werking van het programma illustreren aan de hand van een blokschema en een toestandsdiagram

### Realiseren

- student kan het programma schrijven in een geschikte IDE op de host en het programma executeren op de met de host verbonden target microcontroller
- student kan het programma via de IDE debuggen en testen op de target microcontroller door breakpoints te zetten en de variabelen te monitoren.
- student kan met een microcontroller analoge en digitale componenten uitlezen en aansturen (zoals analoge en digitale sensoren, actuatoren, PWM motorcontrol)

## Organisatie lessen

- 6x 1,5 uur voorbereiding
- 6x 3-uurs practicum
- 1x 3-uurs inhaalpracticum
- De les start met een demonstratie en uitleg over de opdrachten, daarna gaan de studenten aan het werk met de practicumopdrachten
- De studenten gaan steeds meer zelfstandig op zoek naar de benodigde theorie en voorbeelden o.b.v. de datasheets en application notes van AVR/Microchip en example code

## Inhoud vak

- Basiskennis van een microcontroller (hiervoor wordt de AVR128db48 gebruikt)
- Programmeren van een microcontroller m.b.v. een ontwikkelbord (Curiosity Nano)
- Basiskennis gebruik van een IDE als ontwikkelplatform (MPLAB X IDE)
- Basiskennis gebruik van een configuratietool (MCC)
- Registerbased programmeren (aansturen van de hardwareregisters van de uC in C)
- Debuggen, foutcontrole en verbeteren van het geschreven programma (in C)
- General Purpose IO – Werking, instellen, aansturen en uitlezen
- Interrupts – Werking, instellen, afhandelen en integreren binnen project
- Clocks – Werking, instellen en gebruiken
- Timers – Werking en instellen als bron voor PWM
- PWM – Instellen en doorkoppelen aan verschillende peripherals
- Analog-to-digital conversion – Werking, instellen en uitlezen
- Event System – Werking en instellen via MCC
- Uitlezen van data met behulp van diagnostische tools (Datavisualiser)
- Communicatieprotocollen I2C, UART, SPI – Werking, instellen en gebruik om mee te communiceren
- Statemachines – Werking en theorie, maken statemachine diagram, programmeren van een statemachine
- Integreren van peripherals – koppelen van peripherals zodat deze elkaar kunnen aansturen.
- Koppeling van sensoren en actuatoren aan de microcontroller die geleerd worden bij hardware design
- Leren gebruiken van datasheets en andere naslagwerken
- Extra: Leren programmeren van een losse throughhole microcontroller (zonder ontwikkelbord) (AVR128db28)

## Literatuur

- Practicumopdrachten beschrijving
- De programmeertaal C – Kelley & Pohl
- Datasheets en getting started application notes van de gebruikte AVR uC
- Tutorials van AVR <https://mu.microchip.com/page/all-courses>
- Code examples voor de AVR128DA48 of AVR128DB48 beschikbaar via de MPLAB X IDE en op github <https://github.com/microchip-pic-avr-examples>
- Video's van Atmel Studio op Youtube: <https://youtube.com/playlist?list=PL9B4edd-p2ajbkbFpi47P9PfMtQuV0OwQ>
- Video's van Atmel over vergelijkbare AVRTiny [https://www.youtube.com/playlist?list=PLtQdQmNK\\_0DQ8KGcZ1BOPv-3RDPvtqJ1H](https://www.youtube.com/playlist?list=PLtQdQmNK_0DQ8KGcZ1BOPv-3RDPvtqJ1H)
- AVR LibC <https://www.nongnu.org/avr-libc/user-manual/modules.html>
- Datasheets van de gebruikte sensoren en actuatoren
- Extra: Computersystemen en embedded systemen, 5e druk, Leo van Moergestel

## Materiaal

Neem de volgende materialen in principe altijd mee!

- AVR128DB48 Curiosity Nano met het SMU board (incl. OLED) + USB-kabel
- Laptop met MPLAB X IDE + MCC
- Diverse sensoren en actuatoren van project DXC
- Breadboard
- Componentenset
- Gesoldeerde motordriverbord (Prototyping realiseren soldeerworkshop)

## Relatie met andere vakken

Dit vak gaat verder op het vak Project Digitaal (EDPD.24) uit periode 1. We gaan veel dieper en grondiger in op verschillende programmeer- en architectuuronderdelen van de uC. uC-Programmeren vormt samen met Hardware Design de belangrijkste kennis die nodig is voor project DXC (EDDXC.22) in periode 3 en 4. Dit vak bereidt voor op de Professionele Profilerings: Embedded and Wireless Systems.

### Toetsing:

De deelloets (pa1) van uC-programmeren bestaat uit een practicumdeel en een assessmentdeel. Het practicumdeel en assessmentdeel vormen samen het deeltcijfer (ook wel deelloets) uC-programmeren (pa1). uC-programmeren kan gecompenseerd worden met PLC-programmeren. Hiervoor zijn de regels, zoals beschreven in de OER (artikel 6 lid 2) van toepassing. In de basis geldt: voor beide deelloetsen (PLC-programmeren en uC-programmeren) dient minimaal een 4,0 gehaald te worden, en het gemiddelde van deze deelloetsen dient ten minste een 5,5 te zijn. Let op! Met de deelloets (pa1) wordt de gehele afronding van uC-programmeren bedoeld! De regels in de OER zijn altijd leidend. Wanneer deze verandert, terwijl de studiehandleiding achterloopt, gelden de regels van de OER. Lees dus deze regels zelf goed door!

### [Opleidingsdeel OER Elektrotechniek](#)

Hieronder staat vermeld hoe aan beide onderdelen van uC-programmeren voldaan dient te worden, om de deelloets uC-programmeren af te ronden.

### Practicumgedeelte

Tijdens uC-programmeren volg je 6 practica waarvoor je 5 voorbereidingen maakt. Je levert na afloop het practicum + voorbereiding van die week in. Om het practicumgedeelte af te sluiten dien je aan de volgende voorwaarden te voldoen:

1. Je hebt 6 practica bijgewoond. Wanneer je 1x mist, kun je dat inhalen in de responsieles
2. Je hebt 5 voorbereidingen gemaakt en af laten tekenen door de docent. Je mag dit 1x vergeten van te voren. Bij de 2<sup>e</sup> keer word je verzocht eerst de voorbereiding op de gang te maken.
3. Je hebt de practicumuitwerkingen van alle 6 weken ingeleverd op Brightspace

### Assessment

Het assessment bestaat uit een aantal opdrachten die je zelfstandig uitvoert. Voor het assessment neem je dezelfde materialen als tijdens de practica mee: microcontroller (avr128db48 curiosity nano) mee, aansluit kabels, je laptop en je componentensetje. Het assessment bestaat uit een aantal opdrachten die in willekeurige volgorde gemaakt mogen worden.

### Deeltcijfer uC-programmeren

Het cijfer voor de deelloets van uC-programmeren (pa1) wordt bepaald op basis van de volgende regels:

1. Neem je niet deel aan de practica, of voldoe je niet aan de practicumverplichting zoals hierboven beschreven dan is het cijfer voor de deelloets pa1 een 1.
2. Neem je deel aan de practica en voldoe je aan de practicumverplichting zoals hierboven beschreven, dan is het cijfer pa1 gelijk aan je assessmentcijfer.

**Periodeplanning:**

Week	Onderwerpen die aan bod komen
Week 1 – Introductie en	Introductie Microcontroller architectuur + peripherals MPLAB X IDE + MCC Debuggen, breakpoints en monitoring Programma flow
Week 2 – GPIO	GPIO m.b.v. knopje en ledje
Week 3 – Interrupts (ISR)	Interrupts (ISR) Processor- en energieverbruik Sleepmode
Week 4 – Clocks, Timers & PWM	Clock (CLK) Timers (TCA) Pulsewidth modulation (PWM) Aansturing LEDs en motorcontroller
Week 5 – Analog to Digital Conversion (ADC)	Uitlezen van een analoge waarde mbv van ADC Doorsturen van meetwaarden naar laptop mbv UART Data visualiser Koppelen van peripherals mbv event system
Week 6 – Integratie & State Machines	Statemachine als raamwerk voor programma Integratie en koppeling van sensor, controller en actuator Blok-schema's en State Diagrams om programma en systeem mee te modelleren
Week 7 – Uitloop	
Week 8 – Assessment	