# Vak: Hard- en Software Besturingssystemen — credits: 10

| | |
|---|---|
| **Vakcode** | ENDH19HSB |
| **Naam** | Hard- en Software Besturingssystemen |
| **Studiejaar** | 2025-2026 |
| **ECTS credits** | 10 |
| **Taal** | Nederlands |
| **Coördinator** | C.E.J. van Someren |
| **Werkvormen** | Onderwijs |
| **Toetsen** | Chip Ontwerp - Opdracht<br>Programmeren 3 - Opdracht |

**Leeruitkomsten**

| | |
|---|---|
| **LOL 2 Hard- en Software Besturingssystemen** | **De student programmeert in de programmeertaal C, analyseert problemen en creëert en formuleert hiervoor een methodische, stapsgewijze oplossingsmethodiek (algoritme). De student weegt aan de hand van gangbare criteria af, een product-functionaliteit hardware-, softwarematig of met combinaties daarvan te implementeren en u ontwerpt, modelleert en synthetiseert deze digitale elektronica (deel)ontwerpen met een hardware beschrijvingstaal, implementeert de algoritmes in de hardware en hanteert testmethodieken.** |
| De toetsen waarmee deze LOL wordt getoetst | 1 Chip Ontwerp |
| Weging | 50% |
| De leeruitkomst behorende tot deze LOL die in deze toetsmethode worden getoetst | De student kan aan de hand van gangbare criteria, zoals kosten, performance en betrouwbaarheid, afwegen om een product-functionaliteit hardwarematig, softwarematig of met combinaties daarvan te implementeren.<br><br>De student ontwerpt, modelleert en synthetiseert digitale elektronica (deel)ontwerpen met een hardware beschrijvingstaal.<br><br>De student implementeert op systematische manier algoritmes in hardware.<br><br>De student hanteert systematisch testmethodieken. |
| Toetsvorm | Opdracht |
| Toetsmoment | Tijdens module |
| Indicatoren / Beoordelingscriteria: | Programmeerbare CPLD- en FPGA-chips.<br><br>De geavanceerde Quartus II ontwikkelomgeving.<br><br>Talen voor het beschrijven van elektronicasystemen, in het bijzonder de taal VHDL.<br><br>Simulatie en testen van een hardware-ontwerp met de taal TCL.<br><br>Timing, in het bijzonder volledig synchrone |

**Inhoud**

**M6.1 High Tech Systems & Materials Elektro**

**Waar gaat de module over?**

In de module ‘High Tech Systems & Materials’ (HTSM) wordt het thema Smart Industry uitgewerkt. Smart Industry past in de huidige wereld van snelle veranderingen.
Zo zet de globalisering onverminderd door en dwingt de economische werkelijkheid bedrijven om competitief te blijven. Consumenten en afnemers verwachten steeds meer op maat gesneden oplossingen.
Daarnaast maken technologische ontwikkelingen als internet, diepgaande veranderingen mogelijk. De vergaande digitalisering van de industrie zorgt ervoor dat machines onderling met elkaar verbonden zijn en slim worden aangestuurd. Niet alleen binnen de fabriek, maar ook tussen bedrijven onderling en tussen bedrijven en klanten. Het draait om een combinatie van de inzet van productietechnologie, digitalisering en een netwerkaanpak (www.smartindustry.nl).

De digitalisering is hiermee tot in de haarvaten van onze samenleving doorgedrongen en de ontwikkelingen op het gebied van ‘High Technology’ zijn op veel punten afhankelijk van deze digitalisering. Basis van deze digitalisering is de computertechnologie die in een aantal decennia van grote ‘mainframes’ is uitgekomen bij microchips die overal ingebouwd zitten en veelal met elkaar en ‘de cloud’ communiceren.

In deze module wordt bij **‘Programmeren 3’** ingegaan op de essentie van ‘imperatief’ programmeren, waar in een voor mensen toegankelijke taal algoritmes stap voor stap de computerchip instrueren wat er gedaan moet worden.
Behalve op de programmeertaal C, wordt aandacht gegeven aan *de essentie van programmeren*: wat is nu eigenlijk programmeren? Al snel wordt gedacht aan applicaties of apps met schermen met mooie interfaces en dergelijke, maar ook daar zit onder de motorkap altijd de echte code die algoritmes uitvoert op inputdata, en daarmee outputdata genereert.
Programmeren kan worden gezien als een diepgaande manier van systematisch denken om een bepaald probleem op te lossen, een competentie die voor elke ingenieur van groot belang is. Het oefenen in het ontwikkelen van een stap-voor-stapmethode om een probleem op te lossen, is nodig bij het maken van computerprogramma’s, maar ook in tal van andere vakdisciplines.
De taal is ‘slechts’ een middel dat gebruikt wordt, maar natuurlijk ook een middel dat geleerd moet en kan worden.
Het onderdeel ‘Programmeren in C’ is opgedeeld in vier hoofdonderdelen:

1. Inleiding Programmeren in C.
2. Datatypes, functies, bit-operatoren en debuggen.
3. Arrays, pointers, strings en file IO.
4. Datastructuren, dynamisch geheugen en OS-functies.

Bij **‘Chip Ontwerp’** draait het om het ontwerpen van eigen digitale elektronicaschakelingen. Startpunt is basiskennis over digitale

| | |
|---|---|
| | schakelingen.<br><br>Digitaal systemontwerp met een controller en datapadarchitectuur.<br><br>Hardware versus software. |
| Minimaal oordeel | 5,5 |
| De toetsen waarmee deze LOL wordt getoetst | 2 Programmeren 3 |
| Weging | 50% |
| De leeruitkomsten behorende tot deze LOL die in deze toetsmethode worden getoetst | De student *programmeert* in de programmeertaal C programma's met taalconstructies, rekenkundige, logische en bit-operatorenfuncties, enkel- en meervoudige variabelen, pointers en datastructuren, bestanden en OS-interfacing.<br><br>De student *analyseert* een probleem en *creëert* en *formuleert* een methodische, stapsgewijze oplossingsmethodiek en daarvoor de ontwikkeling van algoritmes. |
| Toetsvorm | Opdracht |
| Toetsmoment | Tijdens module |
| Indicatoren / beoordelingscriteria | Gebruik van de CodeBlocks ontwikkelomgeving voor het maken van C-programma's, inclusief het gebruik van beschikbare debug-faciliteiten voor het vinden en oplossen van verschillende soorten programmeerfouten.<br><br>Gebruik van gangbare C-taalelementen (beslissings- en lus-structuren), datatypes en complexe data(structuren), gebruikers IO en het lezen en schrijven van databestanden in de programmeertaal C.<br><br>Het leggen van de relatie tussen data in een C-programma en de representatie daarvan in het geheugen van een computersysteem.<br><br>Het aandacht geven aan leesbaarheid, documentatie en veiligheid van geschreven C-code.<br><br>Ontwikkeling van algoritmen en de realisatie daarvan in de taal C voor oplossing van gegeven problemen. |
| Minimaal oordeel | 5,5 |

techniek, zoals poorten en flipflops. Er wordt kort gememoreerd hoe het traditionele ontwerpproces verliep, en waarom die methode niet meer haalbaar is voor complexe systemen, waarbij een enorme integratiedichtheid bereikt is: tot vele miljoenen poorten in een enkel ontwerp.
Daarvoor wordt tegenwoordig gebruik gemaakt van geavanceerde ontwikkelgereedschappen, zoals het gebruikte Quartus II softwarepakket.
Een van de meest cruciale mogelijkheden daarvan is het gebruik van een 'hogere taal', zoals VHDL, die het mogelijk maakt om digitale elektronica steeds meer functioneel te beschrijven, waarna een 'compiler' deze beschrijving omzet naar de juiste poortjes en verbindingen daartussen maakt. Wel blijft de mogelijkheid om als ontwerper precies de hardware die gegeneerd wordt te specificeren, zodat het voordeel van eigen ontworpen elektronica blijft.
Vervolgens wordt een ontwerp getest door middel van simulatie, met een krachtige taal om testpatronen te maken en uit te voeren.
In een **project** is er ruimte om de hierboven beschreven kennis te gebruiken en te verdiepen. Dit gebeurt door een besturingsprobleem dat zowel hardware- als softwarematig geïmplementeerd kan worden. In dit project wordt de context van de energietransitie als case genomen, waarmee een koppeling wordt gelegd tussen het programmeren van een besturing en de energietransitie.

**Wat kunt u aan het eind van deze module?**

**1. Hard- en Software Besturingssystemen**
U *programmeert* in de programmeertaal C, *analyseert* problemen en *creëert* en *formuleert* hiervoor een methodische, stapsgewijze oplossingsmethodiek (algoritme).
U weegt aan de hand van gangbare criteria af, een product-functionaliteit hardware-, softwarematig, of met combinaties daarvan te implementeren en u ontwerpt, modelleert en synthetiseert deze digitale elektronica (deel)ontwerpen met een hardware beschrijvingstaal, implementeert de algoritmes in de hardware en hanteert testmethodieken.

**2. Ontwerp en Realisatie Besturingssystemen**
Het *analyseren* van een besturingsprobleem binnen energietransitie en het *ontwerpen* en *formuleren* van de architectuur en algoritmes voor de oplossing, het *realiseren* van een software-applicatie in de programmeertaal C, het testen van het gemaakte product om het op overtuigende en professionele wijze op te leveren.

**3. Uw inzicht in uw professionele vaardigheden is vergroot**, dat wil zeggen dat u laat zien hoe u in uw werkomgeving de kennis, vaardigheden en methoden m.b.t. de onderwerpen van het gevolgde onderwijs toepast en reflecteert op uw ontwikkeling en handelen in de beroepspraktijk.

**Opgenomen in opleiding(en)**
Elektrotechniek

**School(s)**
Instituut voor Engineering

share your talent. move the world.

De ECTS onderwijscatalogus van de Hanzehogeschool Groningen wordt met de grootst mogelijke zorg samengesteld. Het is echter mogelijk dat de inhoud van de catalogus -en de daarin vervatte informatie- verouderd, incompleet of onjuist is. Aan de inhoud van de catalogus kunnen dan ook geen rechten worden ontleend.