

Embedded Systems 3

Code: ELEEMS30

Kwartaal 4, Jaar 3

Aantal studiepunten: 6

1 Inleiding

In veel elektrotechnische producten en systemen neemt software een steeds belangrijkere rol in. Voor een (aankomend) elektrotechnisch ingenieur is het dus erg belangrijk om goed te leren programmeren.

In het eerste deel van deze cursus werk je aan het verder ontwikkelen van je programmeervaardigheden in C. Hierbij wordt de nadruk gelegd op het omgaan met dynamisch geheugen, het implementeren van data structuren en het correct implementeren, testen, debuggen en beheren van code.

Programma's in embedded systemen worden steeds omvangrijker. Om er voor te zorgen dat deze programma's uitbreidbaar, aanpasbaar en herbruikbaar blijven wordt vaak gebruik gemaakt van zogenoemde objectgeoriënteerde programmeer- en ontwerp-technieken. In het tweede deel van deze cursus maak je kennis met de objectgeoriënteerde programmeertaal C++ en met een objectgeoriënteerde ontwerpmethode met behulp van UML (Unified Modelling Language).

2 Werkvormen en studielast

In week 1 tot en met 4 zijn er vijf klokuren per week ingeroosterd in twee blokken van twee en een half klokuur. Deze blokken van twee en een half klokuur bestaan uit online labs in Microsoft Teams. Tijdens elk online lab zal eerst de betreffende theorie gepresenteerd worden, waarna gewerkt zal worden aan verschillende labopdrachten waarin de theorie wordt toegepast en uitgebreid.

In week 5 tot en met 8 zijn er theorielessen van twee klokuren per week ingeroosterd (met een half uur pauze ertussen). Tijdens deze theorielessen zal de betreffende theorie van die week gepresenteerd worden. Daarnaast zijn er in deze weken vier klokuren per week ingeroosterd in twee blokken van twee klokuur. Deze blokken van twee klokuren

bestaan uit online labs in Microsoft Teams. Tijdens de online labs zal gewerkt worden aan verschillende labopdrachten waarin de theorie wordt toegepast en uitgebreid.

De verschillende werkvormen zijn weergegeven in de onderstaande tabel.

Werkvorm	Omschrijving	Studielast
Theorielessen	Theorie volgen.	8 klokuren
Practicum	Begin maken met de labopdrachten.	36 klokuren
Zelfstudie	Bestuderen van het studiemateriaal. Uitwerken van alle labopdrachten.	124 klokuren

3 Competenties

Deze cursus draagt bij aan het ontwikkelen van de competentie:

- **analyseren** door de functionele eisen van een systeem weer te geven in use cases (LD6)¹;
- **ontwerpen** door object georiënteerd te ontwerpen met behulp van UML (LD6) en door effectief gebruik te maken van standaard datastructuren en algoritmen (LD5);
- **realiseren** door code te implementeren in C en C++ en deze code systematisch te testen (LD2, LD3 en LD4).
- **beheren** door GIT in groepsverband toe te passen (LD1).

4 Leerdoelen

Deze cursus helpt je bij het bereiken van de volgende doelen:

- een betere C-programmeur worden;
- de basisbeginselen van objectgeoriënteerd programmeren in C++ leren;
- de basisbeginselen van objectgeoriënteerd ontwerpen met UML leren.

De specifieke leerdoelen die getoetst worden staan weergegeven in onderstaande tabel.

¹ LD = leerdoel, zie [paragraaf 4](#).

#	Niveau	Weging	De student is in staat om ...
1	C	5 %	... versiebeheer toe te passen (GIT) zodanig dat de student in groepsverband software revisies kan beheren.
2	C	25 %	... een dynamische datastructuur te implementeren in C met behulp van pointers en de functies malloc en free.
3	C	20 %	... de kwaliteit van code te verbeteren door het toepassen van unit testen, code coverage en coding standards.
4	C	30 %	... gebruik te maken van objectgeoriënteerde programmeertechnieken in C++ zoals encapsulation, templates, overerving en polymorfisme zodanig dat de student programmatuur kan ontwikkelen die uitbreidbaar, aanpasbaar en herbruikbaar is.
5	C	10 %	... op grond van zijn of haar kennis van de big-O-notatie effectief gebruik te maken van de standaard in C++ aanwezige datastructuren en algoritmen.
6	C	10 %	... een objectgeoriënteerd ontwerp te maken van een programma in UML door middel van usecase-, klassen-, sequentie-, toestand- en activiteitendiagrammen.

De beheersingsniveaus van de verschillende leerdoelen zijn afkomstig van de taxonomie van Bloom (met een bewerking van Anderson). A = Kennis, onthouden, B = Inzien, begrijpen, C = Toepassen, gebruiken, D = Problemen oplossen, analyseren, synthetiseren.

5 Toetsing en beoordeling

De leerdoelen worden getoetst op basis van twee praktische opdrachten. Deze opdrachten worden in groepjes van 2 studenten, of als het niet anders kan individueel, uitgevoerd. *Voorwaarde voor het nakijken* is dat alles eigen werk is. Blijkt er niet te zijn voldaan aan deze voorwaarde, dan wordt er onmiddellijk een onvoldoende gegeven en een fraudemelding gedaan bij de examencommissie. Als er twijfel is over de bijdrage van één student van een groepje van twee studenten dan kunnen de docenten deze studenten om een extra mondelinge toelichting vragen.

Toets	Leerdoelen	Weging	Deadline
Opdracht 1	1, 2 en 3	50 %	Lesweek 4, zondag 23.59 uur
Opdracht 2	4, 5 en 6	50 %	Lesweek T4, zondag 23.59 uur

Het eindresultaat is het gewogen gemiddelde van de twee deelresultaten.

Beide opdrachten kunnen, indien onvoldoende, apart van elkaar worden herkanst in HT4. De deadline voor deze herkansingsopdrachten is zondag 23.59 uur van week HT4.

6 Voorkennis

Deze cursus gaat door op de stof van DIS10 en EMS20. Een voldoende voor deze vakken is aan te raden.

7 Aansluiting op verdere studie

Deze cursus biedt veel handige tools om tijdens je minor of het afstuderen te kunnen gebruiken.

8 Verplichte literatuur

Jens Gustedt. *Modern C*. 2de ed. Manning Publications, 2019. ISBN: 978-1-61729-581-2. URL: <https://hal.inria.fr/hal-02383654/document>. Dit gratis boek kun je gebruiken als naslagwerk voor C.

Harry Broeders. *Dictaat Objectgeoriënteerd Programmeren in C++*. 1ste ed. Hogeschool Rotterdam, 2020. URL: https://bitbucket.org/HR_ELEKTRO/cppprog. Dit dictaat wordt gebruikt bij de introductie in C++.

Joost Vennekens. *Objectgericht ontwerpen*. 1ste ed. De Nayer instituut, 2011. URL: <https://people.cs.kuleuven.be/~joost.vennekens/DN/00-UML-cursus.pdf>. Dit dictaat wordt gebruikt bij de introductie in UML.

Bij de lessen maak je gebruik van de MSP-EXP430G2 en de CC3220S-LAUNCHXL ontwikkelborden.

9 Docentenbereikbaarheid

Docent	E-mail	MS Teams chat
R. Bakker	BaRoy@hr.nl	Chat met Roy Bakker
J.Z.M. Broeders	BrojZ@hr.nl	Chat met Harry Broeders
R.T. Verhagen	VeRon@hr.nl	Chat met Ron Verhagen
D. Versluis	VersD@hr.nl	Chat met Daniël Versluis

10 Deadlines en speciale activiteiten

De deadlines voor het inleveren van de opdrachten zijn gegeven in [paragraaf 5](#).

11 Planning

In deze paragraaf wordt de gedetailleerde planning gegeven. De planning kan gedurende het semester nog worden aangepast. Als dit nodig is, dan zal een aangepaste planning worden gepubliceerd.

Week	Werkvorm	Beschrijving
4.1	Les	1: Modules in C , 2: Dynamisch geheugenallocatie
	Zelfstudie	Zie paragraaf 2
4.2	Les	1: Git , 2: Unittesten
	Zelfstudie	Zie paragraaf 2
4.3	Les	1: Memory tests en test coverage , 2: Coding standards en static code analysis
	Zelfstudie	Zie paragraaf 2
4.4	Les	1+2: Werken aan opdracht 1
	Zelfstudie	Zie paragraaf 2
	Opdracht 1	Deadline zondag 23.59 uur
4.5	Les	1: Van C naar C++ , 2: User-defined Data Type in C++

Deze tabel wordt vervolgd op de volgende pagina.

Vervolg van de vorige pagina.

Week	Werkvorm	Beschrijving
	Zelfstudie	Zie paragraaf 2
4.6	Les	1: Templates , 2: Overerving en polymorfisme
	Zelfstudie	Zie paragraaf 2
4.7	Les	1: Datastructuren , 2: Algoritmen, performance en big-O-notatie
	Zelfstudie	Zie paragraaf 2
4.8	Les	1: UML use cases en klassediagram, 2: UML sequentie-, toestands- en activiteitendiagram
	Zelfstudie	Zie paragraaf 2
T4	Opdracht 2	Deadline zondag 23.59 uur
HT4	Herkansing opdracht 1 en 2	Deadline zondag 23.59 uur

Versiehistorie

Datum	Versie	Omschrijving	Auteur
01-2020	1.0	Eerste versie.	BrojZ
06-2020	1.1	Mogelijkheid voor herkansing opdracht 1 in week HT4 toegevoegd.	BrojZ
04-2021	1.2	Alle lessen worden online via Teams gegeven.	BrojZ
04-2022	1.3	Werkvormen en studielast aangepast.	BrojZ
04-2023	1.4	Herkansingsregeling aangepast.	BrojZ