

Real-Time Systems 1

Code: ELERTS10

Kwartaal 1, Jaar 4

Aantal studiepunten: 6

1 Inleiding

Dit studieonderdeel is bestemd voor de studenten van de minor Embedded Systems die wordt verzorgd door de opleiding Elektrotechniek. In deze cursus wordt een introductie gegeven op het ontwerpen van een real-time systeem met behulp van een 32-bit microcontroller.

De microcontroller is uit het hedendaagse leven niet meer weg te denken. Er werden in 2021 naar schatting ongeveer 30 miljard microcontrollers verkocht¹ die veelal gebruikt werden in embedded applicaties. Voor de embedded systems engineer van nu is het dan ook van uitermate groot belang om tot in detail te begrijpen hoe een microprocessor werkt en hoe deze is verbonden met de buitenwereld.

Omdat de Arm architectuur zeer veel toegepast wordt in embedded systemen is besloten om gebruik te maken van een Arm Cortex-M4 microcontroller van STMicroelectronics.

In het begin van de cursus wordt gedetailleerd ingegaan op (een subset) van de instructieset van de Arm Cortex-M4. Daarbij worden onder andere de architectuur, registerset, operating modes, instructieset en timing besproken. Daarna volgt een introductie van het te gebruiken ontwikkelingsplatform. Ook wordt uitgelegd hoe een (eenvoudige versie) van een Cortex-M4 processor hardwarematig is opgebouwd.

Hierna volgt een introductie van verschillende 'scheduling' mogelijkheden en wordt een basis gelegd voor het gebruik van een Real-Time Operating System (FreeRTOS). Daarna leer je hoe je kunt bewijzen dat een real-time systeem zijn deadlines haalt door het berekenen van alle blocking- en responsetijden.

Op dit moment worden bijna alle real-time embedded systemen geprogrammeerd in C of C++². Het gebruik van deze talen zorgt echter regelmatig voor bugs omdat ze niet memory safe en concurrency safe zijn. De taal Rust kan memory bugs voorkomen doordat memory safety in de taal ingebakken is. Bovendien biedt Rust naar eigen zeggen 'fearless concurrency'. Tot slot van deze cursus maak je kennis met Rust en leer je Rust in een embedded omgeving toe te passen.

¹ Bron: www.grandviewresearch.com (2022).

² Bron: www.embedded.com (2019).

2 Werkvormen en studielast

Er zijn een aantal lesuren per week ingeroosterd in twee blokken (lessen). Het eerste blok wordt op de hogeschool gegeven en duurt in week 1.1, 1.2, 1.6 en 1.7 twee en half klokuur en in week 1.3 t/m 1.5 één klokuur. In dit blok wordt de theorie uitgelegd. Het tweede blok van twee en een half klokuur worden online gegeven in de MS Teams omgeving. In dit blok zal gewerkt worden aan de verschillende opdrachten waarin de theorie wordt toegepast en uitgebreid.

De verschillende werkvormen zijn weergegeven in de onderstaande tabel.

Werkvorm	Omschrijving	Studielast
Theorieles	Tijdens deze uren wordt de benodigde theorie uitgelegd.	13 klokuren
Online practicumles	Tijdens deze uren werk je aan de opdrachten.	20 klokuren
Zelfstudie	Bestuderen van de gegeven informatiebronnen en zelfstandig werken aan de opdrachten.	135 klokuren

3 Competenties

Deze cursus draagt bij aan het ontwikkelen van de competentie:

- **analyseren** door het gedrag van een moderne microcontroller te analyseren (LD1)³ en door het gedrag van een real-time systeem te analyseren (LD7);
- **ontwerpen** door software te ontwerpen voor een real-time embedded systeem (LD2 t/m LD6 en LD8);
- **realiseren** door software te implementeren in assembly, C en Rust voor een real-time embedded systeem en door deze software te testen (LD2 t/m LD6 en LD8).

4 Leerdoelen

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

#	Niveau	Weging	De student is in staat om ...
1	C	18,75 %	... de werking van een moderne microcontroller te begrijpen waardoor de student in staat is om een deel van een simpele embedded applicatie in assemblycode te ontwerpen en te realiseren en deze code vanuit C-code aan te roepen en vice versa;

Wordt vervolgd op de volgende pagina.

³ LD = leerdoel, zie [paragraaf 4](#).

Vervolg van de vorige pagina.

#	Niveau	Weging	De student is in staat om ...
2	C	6,25 %	... de pinnen van een 32-bits microcontroller aan te sturen op verschillende abstractieniveaus;
3	C	6,25 %	... een cyclic executive te ontwerpen en te realiseren op basis van een periodieke interrupt;
4	C	6,25 %	... een coöperatieve scheduler te ontwerpen en te realiseren;
5	C	12,5 %	... een pre-emptive scheduler te beschrijven, te ontwerpen en te realiseren;
6	C	12,25 %	... een RTOS te gebruiken inclusief de logische structuren die erbij horen zodanig dat dit gebruikt kan worden bij de realisatie van real-time systemen;
7	D	12,25 %	... te analyseren of een real-time systeem, dat meerdere communicerende taken bevat, zijn deadlines haalt door het berekenen van alle blocking- en responsetijden;
8	C	25 %	... een simpele embedded applicatie of deel van een applicatie in Rust te ontwerpen en te realiseren.

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 verschillende praktische opdrachten. De resultaten van deze opdrachten worden vastgelegd in 3 verslagen die afzonderlijk worden beoordeeld. De opdrachten worden in tweetallen 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
Verslag 1	1 en 2	25 %	Lesweek 1.2, zondag 23.59 uur
Verslag 2	3 t/m 7	50 %	Lesweek 1.6, zondag 23.59 uur
Verslag 3	8	25 %	Lesweek T1, zondag 23.59 uur

Als de cursus niet met een voldoende is afgerond, dan kunnen verslagen die met een onvoldoende zijn beoordeeld, herkanst worden in kwartaal 2. Bij de herkansing zullen

andere opdrachten gemaakt moeten worden. De deadline voor het inleveren van de verslagen over deze herkansingsopdrachten is zondag 23.59 uur van lesweek HT1.

6 Voorkennis

Om deze cursus succesvol te kunnen volgen is kennis van het programmeren in C of C++ en elementaire kennis van microcontrollers noodzakelijk.

7 Aansluiting op verdere studie

De in deze cursus opgedane kennis kan worden toegepast bij de cursus Configurerable System On Chip (CSC10). Studenten kunnen eventueel ook in de afstudeeropdracht te maken krijgen met de onderwerpen die in deze cursus worden behandeld.

8 Verplichte literatuur

Bij de lessen maak je gebruik van het STM32F411E-DISCO ontwikkelbord. Dit kan geleend worden van de hogeschool. Daarnaast wordt gebruik gemaakt van een aantal vrij beschikbare documenten en boeken, die te vinden zijn op de wiki van deze cursus: https://bitbucket.org/HR_ELEKTRO/rts10/wiki/Home.

9 Docentenbereikbaarheid

Docent	E-mail	MS Teams chat
J.Z.M. Broeders	BrojZ@hr.nl	Chat met Harry Broeders
D. Versluis	VersD@hr.nl	Chat met Daniël Versluis

10 Deadlines en speciale activiteiten

Een verslag dat na (onafhankelijk van hoe lang na) de deadline ingeleverd wordt, wordt niet meer nagekeken! De deadline staat vermeld 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
1.1	Lessen	Inleiding microcontrollerarchitectuur en STM32F411E-DISCO
1.2	Lessen Toetsing	Microcontrollerarchitectuur en programmeren in C Verslag 1, deadline zondag 23.59 uur
1.3	Lessen	Cyclic executive en coöperatieve scheduler
1.4	Lessen	Pre-emptive scheduler
1.5	Lessen	FreeRTOS en pthreads
1.6	Lessen Toetsing	Schedulability en response time analyses Verslag 2, deadline zondag 23.59 uur
1.7	Lessen	Inleiding Rust
1.8	Lessen	Embedded Rust
T1	Toetsing	Verslag 3, deadline zondag 23.59 uur
HT1	Herkansing toetsing	Deadline zondag 23.59 uur

Versiehistorie

Datum	Versie	Omschrijving	Auteur
08-2023	2.0	Contacturen beter gespecificeerd.	BrojZ
03-2022	1.0	Eerste versie.	BrojZ