

TI2720-D: Computer Systems “STAR” Project

projectwijzer

Koen Langendoen
K.G.Langendoen@tudelft.nl

11 november 2011

1 Inleiding

Mensen gebruiken tegenwoordig allerhande apparaten, denk daar aan MP3 spelers, mobiele telefoons, digitale camera's, geldautomaten, wasmachines, enzovoort. Wat men men zich daarbij niet realiseert is dat er in deze apparatuur een (embedded) computer zit om het geheel aan te sturen. De form factor van deze computer is vaak klein, met als gevolg dat de mogelijkheden beperkt zijn (limited resources). De uitdaging voor de programmeur van zulke embedded systems zit in het rekening houden met de hardware limitaties, de noodzaak tot energiezuinigheid (voor mobiele apparaten) en het debuggen zonder gebruikelijke input/output middelen (toetsenbord, scherm). Deze aspecten komen terug in het Software Technology Autonomous Robot (STAR) project.

Voor het project maken we gebruik van een embedded system in de vorm van een rijdende robot die autonoom, mbv. sensoren en actuatoren, een parcours moet kunnen afleggen. In het robotica vakgebied komen zulke autonome robots vaak voor. Deze robots worden dan voorzien van specifieke sensoren om te kunnen navigeren, hun (relatieve) positie te bepalen of om informatie te verzamelen. In dit project ga je in groepsverband software ontwikkelen om een robot uitgerust met oa. ultrasound sensoren te kunnen navigeren en die robot door een doolhof te laten rijden middels de besturing van twee motoren.

2 Leerdoelen

Dit project bouwt voort op vaardigheden en kennis uit het eerste jaar (in het bijzonder de twee projecten, en de vakken Schriftelijk rapporteren, OO-programmeren, en Computerorganisatie) en het vak Operating Systems uit het tweede jaar, maar verlangt ook zelfstudie van relevante onderwerpen (bijvoorbeeld signaalverwerking).

Het doel van dit project is drieledig: ten eerste, het leren ontwerpen en implementeren van programmatuur voor embedded systemen, ten tweede, het verbeteren van groepsvaardigheden vereist in projectmatig werken, en ten derde, het ontwikkelen van academische vaardigheden. Met betrekking tot de aspecten van softwaretechnologie voor embedded systemen die tijdens het project aan bod komen, kun je aan het einde van dit project:

- eenvoudige schema's (data sheets) lezen en (daaruit) sequenties van controle commando's genereren,
- low-level interfaces met digitale randapparatuur benoemen en programmeren,
- een device driver voor nieuwe randapparatuur toevoegen aan Linux,
- de scheduler structuur en I/O interfacing in Linux gebruiken,
- software maken voor het real-time verwerken van digitale signalen, en
- een feedback control loop afregelen.

Met betrekking tot het verbeteren van de samenwerking in het project wordt er middels wekelijkse bijeenkomsten gezorgd dat je aan het eind van dit project in staat bent:

- om in teamverband te werken,
- te reflecteren op de eigen bijdrage aan het groepsproces en -product,
- constructieve feedback kunnen geven op elkaars gedrag, ideeën, meningen, en de bijdragen van de andere leden,
- om (in teamverband) een planning te maken en op tijd te werken, en
- effectief te vergaderen, in het bijzonder de rol te beheersen van voorzitter en notulist.

Met betrekking tot het ontwikkelen van academische vaardigheden verwachten we dat je:

- kort en bondig de essentie van een zaak kunt benoemen.
- zelfstandig problemen in kaart kunt brengen.
- Google, Wikipedia en andere openbare bronnen kunt aanwenden.

3 De opdracht: zoek de weg door een doolhof

De opdracht is om programmatuur te ontwikkelen voor een rijdende robot die autonoom de weg door een doolhof moet zoeken. Deze robot gaat zelfstandig beslissen welke route die moet volgen.

De robot is gebaseerd op de Rogue ATR (All Terrain Robot) base kit en het processing board van een LART draagbare computer ontwikkeld in het Ubicom project. De robot bestaat uit een embedded processor, vier ultrasoon afstandsensoren, een kompas, twee LEDs, twee knoppen, een seriële verbinding, een voeding, twee rups banden en twee motoren. Het geheel is gemonteerd op het robot platform, en staat onder controle van het Linux operating system.

Om zijn weg te kunnen vinden in het doolhof moet de robot gebruik maken van de sensoren om de afstand tot obstakels en rijrichting in te schatten. Op basis van deze informatie kiest de robot snelheid en richting. Het lezen van de sensordata en besturen van de motoren staan dus centraal in deze opdracht.

Gevraagd wordt om programmatuur te ontwikkelen die de robot zo snel mogelijk door het doolhof laat gaan. De werking van de programmatuur zal getest worden op de laatste projectbijeenkomst door de robots te laten rijden door een bepaald parcours.

4 Documentatie en materiaal

De benodigde detail informatie (van handleidingen over projectmatig werken tot data sheets van de sensoren) staat op blackboard. Bestudeer alles grondig:

- deze projectwijzer,
- de documentatie van het processing board van de LART,
- de documentatie van de robot en sensoren,
- een voorbeeld over het gebruik van GPIO,
- een voorbeeld van een kernel module,
- hulp bij het debuggen van C programma's,
- informatie over het werken in teamverband, en
- informatie over het werken in projecten: tips voor teams, het leiden van discussies, notuleren, en presenteren.

Ga daarnaast zelf op zoek naar informatie over de signaalbewerking die voor dit project nodig is om de juiste conclusies te trekken uit de data afkomstig van de sensoren.

week	fase	maandag 08:45-12:30 project	dinsdag	woensdag	donderdag 08:45-12:30 project	vrijdag
46	analyse	14/11 08:45-09:30 KICK-OFF			17/11 13:30 haalbaarheidsstudie	
47	ontwerp	21/11 08:45-09:30 FEEDBACK (haalbaarheid)			24/11 13:30 sensor plots	
48		28/11 feedback per groep			01/12 13:30 ontwerp + peerreview	
49	implementatie	05/12 08:45-09:30 FEEDBACK (ontwerp)				
50						
51					22/12 08:45-12:30 oefenparcours	
02	afronding	09/01 17:00 programmatuur			12/01 08:45-12:30 WEDSTRIJD	13/01 17:00 eindverslag reflectie + peerreview

Tabel 1: Het activiteiten rooster voor het STAR Project. Naast de acht projecturen per week (op maandag en donderdag) zijn er vier verplichte plenaire sessies (aangeduid met hoofdletters). De deadlines voor de diverse ‘producten’ zijn vet gedrukt; de (niet-genoemde) notulen van de wekelijkse projectvergadering dienen uiterlijk de **volgende dag** verspreid te worden.

5 Werkwijze

Er wordt gewerkt in groepen van 4-5 studenten. Elke groep wordt begeleid door een *tutor* (student-assistent of docent) die de procesgang bewaakt. De tutor levert dus *geen* inhoudelijke bijdrage, maar stroomlijnt het groepsmatig werken tijdens de wekelijkse bijeenkomsten. Tijdens deze *verplichte* bijeenkomsten wordt geëvalueerd of de gestelde doelen voor de afgelopen week zijn gehaald (zo nee, waarom?), het werk voor de komende week bepaald, en een planning gemaakt (wie doet welke taak). De diverse teamrollen (voorzitter, notulist, enz.) rouleren; na afloop van elke bijeenkomst evalueren de studenten hun functioneren. De notulen van elke vergadering (inclusief evaluatie) dienen zo snel mogelijk ingeleverd te worden via CPM¹ om de docent een goed inzicht te verschaffen in de voortgang v/d diverse groepen en er, indien noodzakelijk, bij gestuurd kan worden. Alle andere (tussen)producten die tijdens het project vervaardigd worden moeten ook ingeleverd worden via CPM. Op blackboard staat ook achtergrondinformatie over groepswork; bestudeer deze.

6 Fasering

Dit project duurt 7 weken en heeft een omvang van 4 EC punten. Het project is onderverdeeld in vier fasen (analyse, ontwerp, implementatie en afronding) die elk in een “product” resulteren (zie tabel 1).

De bedoeling van de analysefase is helder te krijgen *wat* de embedded software nu precies dient te doen, onder welke omstandigheden, enz. Het resultaat wordt vastgelegd in een haalbaarheidsstudie. In de ontwerpfase moet vervolgens duidelijk worden *hoe* één en ander gerealiseerd zal gaan worden. Omdat er met meerdere mensen tegelijk aan de embedded software gewerkt zal moeten worden is het van belang om het geheel in componenten (modules) op te splitsen en tussenliggende interfaces goed te documenteren in een systeemontwerp. Verder moet er ook gedefinieerd worden welke signaalbewerking op de ruwe sensor data noodzakelijk is. In de implementatiefase volgt dan de eigenlijke ontwikkeling van de programmatuur

¹Complete Project Monitoring (CPM): <https://cpm.ewi.tudelft.nl>

en de oplevering van een werkend systeem (device driver + user-land applicatie). In de afrondingsfase is een verplichte bijeenkomst gepland waarin de deelnemende groepen in een onderlinge wedstrijd op een testparcours uitmaken welk product het beste presteert. Ook moet er een eindverslag ingeleverd worden over het gehele project (van haalbaarheidsstudie t/m implementatie).

Tijdens het project wordt er tweemaal gevraagd om middels een (anoniem) **peerreview** te beoordelen hoe de afzonderlijke groepsleden functioneren. Je moet hierbij ook jezelf een waardering geven. De resultaten worden gebruikt om het groepsproces te stroomlijnen (functioneert iedereen zoals hij/zij denkt?) en de docent te informeren over de verhoudingen tussen de studenten in de groep. De peerreviews worden ingeleverd via Studentrate². Ter afsluiting wordt nog gevraagd een persoonlijke **reflectie** te schrijven over hoe het project verlopen is, en wat er geleerd is.

7 Producten en beoordelingscriteria

Algemeen: De verslagen dienen in het PDF formaat ingeleverd te worden en *beknopt* te zijn. De volgende harde limieten gelden voor het document zonder eventuele bijlagen. Dit document moet ook zonder die bijlagen goed te begrijpen zijn.

notulen	≤ 2 pagina's
haalbaarheidsstudie	2 pagina's
sensor plots	2 pagina's
ontwerp	≤ 5 pagina's
eindverslag	≤ 5 pagina's
reflectie	1 pagina p.p.

De notulen (7 stuks) dienen uiterlijk de dag na de wekelijkse groepsvergadering ingeleverd te worden via CPM. De deadlines voor de diverse (tussen)producten staan vermeld in het faseringsschema in tabel 1. De agenda dient verspreid te worden **voor** elke vergadering aan de groepsleden en tutor.

Notulen: De volgende elementen dienen in ieder geval aanwezig te zijn:

- Aanwezig/afwezigheid groepsleden.
- Rolverdeling (voorzitter, notulist).
- Agenda.
- Weergave van de behandeling van de agendapunten:
 - voortgang (zijn actiepunten van de vorige keer wel/niet gedaan?)
 - inhoudelijke discussie
 - bijstelling projectplanning (indien van toepassing)
- Actiepunten (wie doet wat komende week).
- Reflectie op de vergadering (in het bijzonder de rol van de voorzitter).

Haalbaarheidsstudie: De haalbaarheidsstudie is als voorstudie bedoeld om de randvoorwaarden waarbinnen en de eisen waaraan de besturingsprogrammatuur moet voldoen te definiëren. Denk hier bijvoorbeeld aan hoe je de sensor data leest, de robot autonoom laat rijden, en de (kortste) weg uit een doolhof kan vinden. Denk goed na over wat realistisch is gegeven de eigenschappen vd hardware en de beoogde functionaliteit vd software, en over de implicaties die dat heeft voor het ontwerp/implementatie traject.

²Studentrate: <http://studentrate.ewi.tudelft.nl>

Sensor plots: Schrijf een programma om de sensoren uit te lezen (ultrasoon sensoren en kompas). Schrijf een programma om de robot te laten rijden. Maak plots van de ruwe data van de sensoren en een schets situatie waarin je laat zien wat voor pad je robot heeft gevolgd en waar de obstakels gepositioneerd waren. Geef een korte toelichting op de situatie schetsen en sensorplots.

Ontwerp: Gebruik je plot van sensor waarden om vast te stellen welke signaalverwerking noodzakelijk is. Houd ook rekening met het feit dat mobiele apparaten altijd zuinig om moeten gaan met hun resources.

Geef vervolgens een functionele decompositie van de (te implementeren) software. Gebruik hiervoor de methodieken van het OO-programmeren: Klassendiagrammen en Volgordediagrammen. Merk op dat er hier geen sprake is van meerdere objecten van dezelfde klasse. Houdt hier rekening mee bij het maken van de diagrammen. Geef enige details over het navigatie algoritme (hint: gebruik een Toestandsdiagram).

Maak een realistisch tijdsplanning voor de rest van het project (ontwerpen, programmeren, software integreren, afronden en verslag schrijven).

Programmatuur: Enige vuistregels voor het schrijven van robuuste, leesbare (C) programma's.

- Houd functies en procedures kort, zeg maximaal 30 regels, door complexe taken op te splitsen.
- Dupliceer geen code, maar schrijf een hulpfunctie die meerdere keren aangeroepen wordt.
- Controleer de return waarden van system calls, en geef nette foutmeldingen indien noodzakelijk.
- Specificeer en controleer invarianten
- Wees matig in het gebruik van globale (klasse) variabelen.
- Gebruik een consistente lay-out.³
- Voorzie code van commentaar en geef aan welk deel (functie) door wie is geschreven in de functieheader.

Eindverslag: Geef geen chronologische opsomming van wat er gebeurd is tijdens het project, maar een resultaatgericht verhaal: probleem, aanpak, oplossing, ervaring. Vermeld verder in een bijlage wat de werkverdeling geweest is (wie heeft welk onderdeel geprogrammeerd, wie heeft wat aan het eindverslag geschreven, enz.). De beoordelingscriteria die gehanteerd worden voor het verslag zijn gebaseerd op die van het vak "Schriftelijk rapporteren" uit het eerste jaar. Bekijk de eisen van dit vak.

Reflectie: Geef jouw persoonlijke visie op het project: wat je hebt geleerd (in relatie tot de leerdoelen vermeld in sectie 2), en hoe het projectmatig werken verliep in de groep en jouw rol daarin. Lever je reflectieverslag in op CPM.

Beoordeling

Het eindcijfer is een combinatie van een groeps cijfer voor het uiteindelijke product (embedded software) en een individueel cijfer voor de persoonlijke bijdrage aan het groepsproces. Het 'productcijfer' wordt vastgesteld door de docenten aan de hand van de verslagen (haalbaarheidsstudie, productontwerp en eindverslag), de software (structuur, leesbaarheid, enz.) en de 'wedstrijd' resultaten (onder anderen nauwkeurigheid en robuustheid). Criteria voor de beoordeling van het 'procescijfer' zijn inzet en bijdrage aan de *verplichte* groepsbijeenkomsten zoals deze te lezen zijn in de wekelijkse notulen; ook de tutor fungeert als bron voor de beoordeling van het procescijfer.

$$\text{eindcijfer} = \text{productcijfer} + \text{procescijfer} - \frac{\sum_{i=1}^N \text{procescijfer}_i}{N}$$

Het procescijfer wordt dus genormaliseerd ten opzichte van het rekenkundig gemiddelde van de procescijfers van de gehele groep.

³Je kunt hiervoor bijvoorbeeld het programma `indent` gebruiken.

8 Praktische zaken

- Aanwezigheid bij de plenaire sessies (onder andere de kick-off en de wedstrijd) is *verplicht*, net als bij de wekelijkse groepsvergadering.
- De taakverdeling binnen de groep is gebonden aan twee randvoorwaarden:
 1. iedereen zit een keer een vergadering voor en maakt een keer notulen, en
 2. iedereen programmeert een deel van de software (in C).
- Ter ondersteuning van het maken van software in teamverband stelt de faculteit een subversion server ter beschikking (<https://svn.ewi.tudelft.nl>). Elke groep wordt hier geacht een account aan te maken, en tevens de tutor toegang te geven zdd. de voortgang ook inhoudelijk bewaakt kan worden.
- De plenaire sessies vinden plaats in **zaal EWI-CZ E** (14/11/2011, 21/11/2011, 05/12/2011) en **zaal CT-IZ 1.96** (12/01/2012).
- Op de maandag en donderdag is de **projectzaal DW-PC 0.010** aan de Drebbelweg 5 (gebouw 35) beschikbaar om met de robot te werken. Iedere groep kan dan beschikken over een tafel met PCs en een “eigen” robot. De robot wordt verstrekt door de aanwezige student-assistent en dient na afloop weer ingeleverd te worden. De student-assistent dient tevens als vraagbaak voor allerlei praktische zaken betreffende de robot en bijbehorende software.
- Elke groep dient in overleg met zijn tutor af te spreken wanneer en waar de wekelijkse groepsbijeenkomst plaatsvindt.
- Notulen en groepsproducten (haalbaarheidsstudie, ontwerp, enz.) dienen via CPM⁴ ingeleverd te worden.
- De peerreviews dienen via Studentrate⁵ ingeleverd te worden.

⁴CPM: <https://cpm.ewi.tudelft.nl>

⁵Studentrate: <http://studentrate.ewi.tudelft.nl>