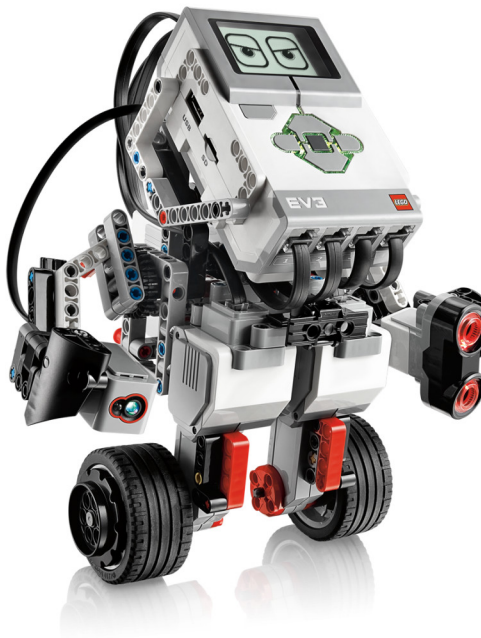


Projecthandleiding Basis Robotica

Lego® Mindstorms® & *First*® Lego® League
Technische Informatica, periode 1 jaar 1



Sander Gieling

Alkmaar, 9 mei 2016

Inhoudsopgave

1. Inleiding	5
2. Voorkennis	7
3. Plaats in het curriculum	9
4. Leerdoelen	11
5. Planning	13
6. Benodigdheden	17
6.1. Materialen	17
6.2. Middelen	18
6.3. Literatuur	18
7. Toetsing	19
8. Evaluatie	21
Appendices	
A. Beoordelingsformulier	25
B. Toetsmatrijs	27
C. Werkwijze Individueel Assessment	29

1. Inleiding

Deze handleiding voorziet je van alle noodzakelijke informatie om het “Project Basis Robotica”¹ uit te voeren en tot een goed einde te brengen. In het kort zul je met een medestudent een robot gaan **ontwerpen, bouwen, programmeren en documenteren**, waarbij je gebruik maakt van Lego®Mindstorms®-onderdelen. Het zou natuurlijk leuk zijn als je uiteindelijk zoiets kunt maken als de *CubeStormer*, te zien in Figuur 1.1. Tijdens het “Project Basis Robotica” zul je ervaren dat bij het bouwen van zelfs een eenvoudige robot veel meer komt kijken dan je zou denken. Met het oog daarop zou je *CubeStormer* kunnen zien als een lange termijn-uitdaging.

Het hoofddoel van het project is het integraal toepassen van de kennis die je in de vakken van deze periode opdoet. Je kunt aan het einde van dit project in teamverband:

- programmacode schrijven voor het uitvoeren van een mechanische taak;
- je projectresultaat met een video-opname demonstreren;
- een onderzoeksverslag vormgeven volgens onderzoeksrichtlijnen en in dat verslag je ontwerpkeuzes en resultaten effectief aan je lezers overbrengen.



Figuur 1.1. “CubeStormer 3” (CS3), tussen 15 maart 2014 en 15 oktober 2015 officieel ’s werelds snelste Rubik’s Cube oplosser. CS3 werd gemaakt met Lego®Mindstorms®-onderdelen - overgenomen van <https://www.youtube.com/watch?v=Fy4Qa0lNfcs>. Copyright © 2014 Michael Grønne.

¹Toetscode: 1914TI116A

2. Voorkennis

Dit project heeft een introducerend karakter. Het is het eerste project waaraan je bij het volgen van de opleiding Technische Informatica (TI) deelneemt. Er is daarom geen voorkennis vereist anders dan de kennis die je in je vooropleiding hebt opgedaan. Kort samengevat rekenen we op de volgende kennis en vaardigheden vanuit je vooropleiding:

- Je bent in staat om een eenvoudig verslag te schrijven en kunt bij het schrijven van een zakelijke tekst structuur aanbrengen in de informatie die je wilt overbrengen.
- Uit je schrijfwerk blijkt dat je de Nederlandse taal in woord en geschrift machtig bent. Als dit niet zo is, bestaat de mogelijkheid om extracurriculair (buiten de reguliere lessen om) in deze periode Nederlands te volgen.
- Je bent in staat om een moderne tekstverwerker te bedienen.
- Je kunt Engels lezen en begrijpen! Veel van de internetpagina's waarop instructies staan voor het aansluiten van apparatuur en het installeren van de bijbehorende software, zijn alleen in het Engels beschikbaar.

Naast de bagage van je vooropleiding zien we een aantal praktische computervaardigheden als zeer wenselijk:

- Het kan handig zijn als je al wat praktische ervaring hebt met het *als administrator* bedienen van computers met daarop één van de veelvoorkomende besturingssystemen (MS Windows, Mac OS X of een Linux-variant).
- Het zou helemaal mooi zijn als je eerder te maken hebt gehad met het aansluiten van hardware (driver-installatie) op een personal computer.
- Als je vroeger (of nog steeds!) hebt gebouwd (bouwt) met Lego[®]Technic[®](of zelfs met Lego[®]Mindstorms[®]), dan is dat mooi meegenomen.
- Je kunt een filmpje maken met je smartphone en je kunt er met videobewerkingssoftware een nette montage van maken. Tijdens het project is er genoeg tijd om je vaardigheid op dit gebied te ontwikkelen.
- Praktische ervaring met (protocollen voor) datacommunicatie is een pré. Hieronder vallen onder andere netwerkcommunicatie (zoals verbindingen via Telnet en Secure Shell) of overige datatransmissie (zoals een seriële poort-terminal). Tijdens het project is er voldoende tijd om je vaardigheden op dit gebied te ontwikkelen.
- GI(N)YF en RTFM. Als moderne internetgebruiker weet je dat deze initiaalwoorden zó 2013 zijn! Maar je bent natuurlijk wél op de hoogte van hun betekenis. Zo niet, GIYF!

3. Plaats in het curriculum

Een technisch informaticus, of *computer engineer*, zoals deze in het Engels wordt genoemd, is in staat om software te ontwikkelen die hardware aanstuurt. De praktijk van de interactie tussen jouw machine en de fysieke wereld is echter weerbarstig. Je zult doorzettingsvermogen moeten kweken om je machine tot op de vierkante bit onder controle te krijgen. Niet “als het werkt, dan is het goed”, maar écht begrijpen wat je doet. Dit project is erop gericht om je van meet af aan te confronteren met deze TI-specifieke problematiek.

Dit project staat niet op zichzelf. Aan het eind van dit project ben je in staat om een eenvoudige robot te programmeren en je vakgenoten er op zinnige wijze iets over te vertellen (zie Hoofdstuk 1). In het tweede studiejaar tillen we deze kennis en vaardigheden naar een hoger niveau, als je meer ervaring hebt met sleutelen en solderen aan hardware¹ en het ontwerpen en schrijven van embedded software². Je gaat dan geheel op eigen kracht een robot realiseren. Niet met Lego[®], maar met zelf gekozen mechanische en elektronische componenten. Dit project is een eerste stap in die richting.

In Hoofdstuk 2 is de voorkennis benoemd die je nodig hebt voor dit project. Het is je misschien opgevallen dat er een aantal zaken missen uit die lijst, bijvoorbeeld programmeervaardigheid. In de vakken die je in deze onderwijsperiode volgt, vul je je voorkennis aan met de zaken die je mist om je project tot een goed einde te brengen. Dit zijn:

- Inleiding Programmeren
Door het volgen van dit vak leer je typische softwareproblemen te analyseren en leer je een schematechniek³ waarin je het ontwerp van een oplossing kunt uitdrukken.
- Praktijk Programmeren
Je leert het omzetten van een programma-ontwerp (een PSD) naar een werkend computerprogramma, uitgedrukt in de programmeertaal C#.
- Algebra
Door het verkrijgen van wiskundig inzicht, vergroot je zowel je abstractievermogen als het vermogen systematisch en mechanisch te redeneren. Hoe dichter je op de hardware werkt, hoe beter je zult zien dat informatica in wezen een “wereld van getallen en logica” is.

¹Het project in de derde periode van het eerste jaar is de volgende stap op de route: je leert dan elektronische schakelingen in elkaar zetten.

²In de eerste twee periodes van het tweede studiejaar ga je aan de slag met microcontrollers en embedded software. Je leert dan “programmeren op de vierkante bit”.

³PSD: Programma Structuur Diagram

3. Plaats in het curriculum

- **Onderzoek 1**
Tijdens het volgen van het vak “Onderzoek 1” leer je het uitvoeren en vastleggen van experimenteel onderzoek. Zo maar iets proberen is leuk, maar door systematisch te experimenteren werk je doelgerichter en maak je je werk controleerbaar. Het afleggen van verantwoording voor gemaakte technische keuzes gaat uiteindelijk een grote rol spelen in je professionele werkhouding.
- **Nederlands (extracurriculair)**
Zoals aangegeven in Hoofdstuk 2 is het mogelijk om buiten de reguliere lessen om je beheersing van de Nederlandse taal aan te scherpen. Dit kan je van pas komen bij het schrijven van je onderzoeksverslag.

De overige vakken die in deze periode worden gegeven zijn ófwel ter voorbereiding op vakken in de volgende periode (Gegevens modelleren), óf betreffen niet-vakinhoudelijke zaken (Studieloopbaanbegeleiding (SLB)), die je studie-effectiviteit moeten vergroten.

4. Leerdoelen

In de vakken die je volgt werk je gedurende je hele opleiding aan vijf beroepscompetenties¹. Deze competenties zijn inhoudelijk van aard. In deze onderwijsperiode worden je ontwerpvaardigheden bijvoorbeeld getoetst in het vak “Inleiding Programmeren” en worden je realisatievaardigheden getoetst in het vak “Praktijk Programmeren”. Door deel te nemen aan projecten werk je aan twee overige competenties: Professionaliseren en Onderzoeken². Wat je leert bij het meedoen aan een project is daarom primair gericht op jouw ontwikkeling als TI-professional.

De competenties professionaliseren en onderzoeken zijn onderverdeeld in een aantal indicatoren, waarop we kunnen meten of je de nodige ontwikkeling hebt doorgemaakt. Om je *professionele ontwikkeling* te meten, kijken we in dit project naar de volgende zaken:

- het verantwoorden van je aanpak;
- het komen tot een resultaat in teamverband;
- rapporteren over het proces en het resultaat;
- communiceren op het niveau van je mede-ingenieurs(-in-wording).

Je *onderzoeksontwikkeling* meten we op basis van deze indicatoren:

- het samenwerken aan een onderzoeksopdracht;
- het onderzoeksresultaat overzichtelijk presenteren.

Deze indicatoren komen niet zomaar uit de lucht vallen. De curriculumcommissie van de opleiding TI heeft bepaald dat dit de aspecten zijn waaraan we je ontwikkeling in deze competenties kunnen aflezen. Er zijn dus ook andere vakken en projecten die via deze indicatoren een bijdrage leveren aan het verwerven van de competenties. Daarom zijn deze indicatoren niet zo heel concreet geformuleerd. Om het meten van je ontwikkeling te relateren aan de inhoud van het project, worden de indicatoren uitgewerkt in **leerdoelen**.

Een leerdoel beschrijft heel concreet wat je moet kunnen als je het project hebt afgerond. Op basis van de genoemde indicatoren en de aard van het project zijn de leerdoelen geformuleerd. In de toetsmatrijs voor het vak kun je de leerdoelen en bijbehorende indicatoren terugvinden. Je vindt het overzicht in Bijlage B. Per leerdoel vind je hieronder een korte toelichting, die je een beeld geeft van wat je zult moeten gaan doen:

¹Analyseren, ontwerpen, adviseren, realiseren en beheren

²Tijdens het volgen van de opleiding Technische Informatica werk je aan vijf beroepscompetenties en twee overige competenties, in totaal dus *zeven* competenties.

4. Leerdoelen

- Je bent in staat om de door jouw toegepaste programmacode toe te lichten en de door jou gekozen oplossing te verantwoorden.

In het kort: je kunt uitleggen wat je programmacode doet en waarom je het zo hebt geschreven.

- Je bent in staat om in de vorm van programmacode een bijdrage aan het project te leveren die in verhouding staat tot de omvang van je projectgroep en het projectresultaat.

In het kort: je hebt je programmeer-aandeel geleverd in het groepswerk

- Je hebt aantoonbaar gewerkt aan je vermogen om samen met andere studenten tot een resultaat te komen.

In het kort: met een videodemonstratie toon je aan dat de robot een aantal opdrachten kan uitvoeren

- Je kunt een onderzoeksverslag vormgeven volgens de richtlijnen voor rapporteren in een experimentele onderzoekscontext.

In het kort: je rapport voldoet aan vormtechnische richtlijnen

- Je bent in staat om je mechanisch robot-ontwerp overzichtelijk in beeld te brengen.

In het kort: er staan duidelijke foto's van je robot in je verslag

- Je kunt je softwarematige ontwerpkeuzes effectief communiceren.

In het kort: er staan zinnige PSD's in je verslag

Het voordeel van een leerdoel is dat we het kunnen gebruiken om af te spreken hoe de toetsing plaatsheeft. In Hoofdstuk 7 laten we zien hoe we de genoemde leerdoelen “examineren”.

5. Planning

Om de leerdoelen te behalen die zijn benoemd in Hoofdstuk 4, hebben we een planning voor je opgesteld, te zien in Tabel 5.1. Als je je aan de planning houdt, kun je er van op aan dat je goed voorbereid je toets tegemoet gaat.

In Tabel 5.1 zijn voor elke lesweek lesdoelen opgenomen. Zij vertellen je wat je die week *moet kunnen*. Daarnaast zijn er activiteiten opgenomen: door ze uit te voeren kun je de lesdoelen bereiken. Ten slotte is per week aangegeven wat je moet inleveren bij of laten zien aan je projectbegeleider (*deliverables*). Hierdoor kun je laten zien dat je op schema zit. Het aantal uren dat je elke week nodig hebt om je activiteiten uit te voeren, is in de laatste twee kolommen genoemd (projectbegeleidingsuren en zelfstudie-uren).

Bij de aanvang van het project komt er veel nieuwe informatie op je af. Dit kan overweldigend zijn en het kan ervoor zorgen dat je niet weet waar je moet beginnen. Om je op weg te helpen, voorzien we je (naast de planning) van een lijst met globale stappen die je *opeenvolgend kunt uitvoeren* om tot een goed projectresultaat te komen:

Week 1:

- Neem tijdens het eerste begeleidingsmoment alle materialen in ontvangst die je door de opleiding ter beschikking worden gesteld voor dit project. Kijk via de meegeleverde materiaallijst na of de legodozen compleet zijn.
- Zoek uit wat [Lego®Mindstorms®EV3](#) is en wat je kunt doen met de EV3 programmeerbare “brick” (een legosteentje met een microcontroller erin)
- Maak jezelf vertrouwd met de [First® Lego® League \(FLL\)](#) en alles wat daarmee samenhangt (de “challenge” van dit jaar, missies, spelregels, veldopbouw, jurering, robotontwerp, hulpstukken, etc.)
- Kies een FLL-missie uit die je met je toekomstige robot wilt gaan uitvoeren op de FLL-wedstrijdtafel
- Zoek uit hoe je met de programmeertaal C# een EV3 brick kunt programmeren - je zult je hiertoe moeten verdiepen in het [MonoBrick-project](#)
- Zoek uit hoe je een verbinding kunt maken tussen je ontwikkel-PC en de EV3 brick - *de verbinding moet het gebruik van C#-programma's ondersteunen* (zie de werk-instructie “Getting Started with MonoBrick”, te vinden op Blackboard)
- Installeer de nodige ontwikkeltools, firmware en drivers op je PC en de EV3 brick (de werk-instructie “Getting Started with MonoBrick” bevat een installatiehandleiding en is te vinden op Blackboard)

5. Planning

Week 2:

- Test de verbinding tussen je PC en de EV3 brick (zie “Getting Started with MonoBrick” op Blackboard)
- Ontwerp en bouw een legorobot die de door jou gekozen missie kan uitvoeren - gebruik hierbij minimaal één motor en minimaal één sensor
- *Denk er telkens aan dat je foto's maakt van je robot als je deze wijzigt - schrijf ook op waarom je de robot wijzigt*
- Test of je robot goed werkt met de MonoBrick-toolchain door er [vanaf je PC een programma naar te verplaatsen / erop te draaien](#) (er is een [testprogramma](#) beschikbaar gesteld binnen het MonoBrick-project)

Week 3:

- Ontwerp een algoritme dat de robot aanstuurt om de door jou gekozen missie uit te voeren - *maak hiervan een PSD*
- Zoek uit hoe je een rapport opmaakt met het zetsysteem \LaTeX - op Blackboard staan een aantal verwijzingen om je op weg te helpen
- Maak een “skelet” voor je onderzoeksverslag in \LaTeX

Week 4:

- Test de verschillende onderdelen van je robot apart van elkaar (dit heet *unit-testing*): maak een klein programma dat de motoren laat draaien en test het, maak daarna een programma dat de sensoren uitleest en de waarden toont op het scherm, waarna je dat ook test, etc.
- Schrijf op basis van het PSD dat je vorige week hebt gemaakt een C#-programma dat de door jou gekozen missie **volgens de spelregels van de FLL** uitvoert - omdat alle projectleden dit moeten doen is het verstandig om apart van elkaar programma's te schrijven van je eigen PSD, waarna je elkaar kunt uitleggen wat je hebt geschreven en waarom je het zo hebt aangepakt
- Bouw in samenwerking met de andere projectgroepen een [wedstrijdtafel](#) op volgens de FLL-richtlijnen (zie [voorbeelden van vorige jaren](#)) in de projectruimte L0-42

Week 5:

- Je bent nu klaar voor je eerste *experiment*: robot gebouwd, verbinding getest, programma ontworpen en geschreven, robot geprogrammeerd, wedstrijdtafel paraat - *voer je eerste experiment uit en leg de resultaten goed vast (filmen, documenteren)*
- Documenteer op basis van je resultaten wat je moet veranderen aan de robot en de programmacode als je experiment niet helemaal volgens plan verloopt (robot herbouwen/verbouwen, nieuwe foto's maken van je verbeterde model, PSD (her)opstellen en programmacode (her)schrijven)

- Vul je onderzoeksverslag aan met de opzet, resultaten en conclusies van je experiment(en).

Week 6:

- Blijf de experimentele cyclus genoemd bij week 5 herhalen tot je de met succes uitgevoerde missie (volgens de spelregels van de FLL) met een videocamera hebt vastgelegd

Week 7 (– 8):

- Schrijf je onderzoeksverslag af op basis van je voortschrijdende inzicht: neem alle experimenten op en verantwoord hoe je tot je uiteindelijke oplossing bent gekomen
- Als je nog voldoende tijd hebt, loont het om meer dan één missie (liefst achter elkaar, ononderbroken gefilmd) uit te gaan voeren - natuurlijk documenteer je je experimenten
- Als je nog nooit eerder een filmpje hebt gemonteerd, is dit een goed moment om daarmee wat te oefenen. Vraag eens rond aan je medestudenten hoe zij dit aanpakken of zoek op internet hoe anderen dat doen. We verwachten niet van je dat je een professional wordt in het gebruik van Final Cut Pro, Sony Vegas Pro, Adobe Premiere Pro, Pinnacle Studio, Blender en/of Lightworks. Windows Movie Maker (Windows), ShotCut (o.a. Linux) of iMovie (Mac) is goed genoeg!

Week 9:

- Monteer je definitieve demonstratiefilmpje, upload het naar DropBox, YouTube, o.i.d. en archiveer een link naar je filmpje op Blackboard - op Blackboard kun je ook wat filmpjes vinden van de studenten uit vorige jaren, misschien heb je er wat aan
- Archiveer je definitieve verslag en de laatste werkende versie van je programmacode (in een zip bestand met daarin je volledige C# project inclusief al je uitgevoerde missies) op Blackboard
- Bereid jezelf en elkaar voor op het individueel assessment door je groepsgenoten te ondervragen over het hoe en waarom van hun programmacode

Als je een stap niet begrijpt, zoek je als projectgroep eerst in de documentatie (voor een lijst met brondocumenten, zie § 6.3). Als dat geen helderheid geeft, kun je tijdens de projectbegeleidingsuren om opheldering vragen. Tips:

- Gebruik deze lijst in combinatie met de planning van Tabel 5.1. Op die manier kun je inschatten welke stappen van de lijst je in welke lesweek uitgevoerd moet hebben en wanneer je welke resultaten aan je projectbegeleider moet laten zien.
- Je begeleidend docent heeft op Blackboard een klassikale projectwiki aangemaakt. Als je dingen ontdekt die niet in de handleiding worden uitgelegd, kun je ze met de rest van de klas delen door een *entry* toe te voegen. Zet je naam erbij, tijdens de toetsing zien je assessoren dit als een geleverde bijdrage!

5. Planning

Tabel 5.1. *Gedetailleerde onderwijsplanning project basis robotica*

Lesweek ¹	Lesdoelen	Activiteiten	Deliverables ²	Contacturen	Zelfstudieuren
1	Na deze les kun je ... • ... uitleggen wat de FLL doet, wat de "Challenge" is en wat "missies" zijn • ... uitleggen waarom je het MonoBrick project nodig hebt • ... de firmware van je EV3 brick aanpassen m.b.v. MonoBrick + een SD kaart	<ul style="list-style-type: none"> • Materialen in ontvangst nemen en op volledigheid controleren • Vertrouwd raken met de FLL en met de "Challenge" van dit jaar • Vertrouwd raken met het MonoBrick project • Toolchain installeren (zie het document "Getting Started with MonoBrick" op Blackboard) 	–	8	6
2	Na deze les kun je ... • ... een bruikbare legorobot bouwen met minimaal één motor en één sensor • ... programma's uploaden van je PC naar de EV3-brick	<ul style="list-style-type: none"> • (Eerste versie van de) robot bouwen die geschikt is voor het uitvoeren van op zijn minst één van de missies (incl. motor/sensor) • Een testprogramma uploaden naar en uitvoeren op de brick 	Een werkende verbinding tussen je ontwikkel-PC en de EV3 (Mono)Brick	8	6
3	Na deze les kun je ... • ... de structuur van je verslag ³ in L ^A T _E X (of Word) opmaken • ... een PSD ⁴ opnemen in je verslag	<ul style="list-style-type: none"> • Je verslag volgens de richtlijnen opmaken in L^AT_EX (of Word) • PSD opstellen van de oplossing voor je missie - <i>je gebruikt minimaal één sensor en één motor!</i> 	Fysieke legorobot	8	6
4	Na deze les kun je ... • ... zelf een C#-programma schrijven en uploaden naar de brick om het te testen • ... je eigen code verantwoorden	<ul style="list-style-type: none"> • Een C#-programma schrijven voor het unit-testen van de diverse onderdelen van je robot + uitvoeren van unit-tests • Vragen stellen over elkaars code (oefenen) 	<ul style="list-style-type: none"> • PSD voor je eerste missie • Skelet van je verslag 	8	6
5	Na deze les kun je een experiment verantwoorden, er foto's van maken en het documenteren	Experimenten uitvoeren (wat moet er anders? → wijzigingen aanbrengen → experiment uitvoeren → vastleggen)	<ul style="list-style-type: none"> • C#-code van je unit-tests • Mondelinge code verantwoording 	8	6
6	Na deze les kun je samen met je groepsleden tot een demonstreerbaar resultaat komen (en een video-opname maken van je uitgevoerde missie(s))	Op de FLL wedstrijdtafel als groep een missie uitvoeren en hiervan een video-opname maken	Verantwoording van een experiment incl. foto's in je verslag	8	6
7	Na deze les kun je experimenten volgens de onderzoekscyclus documenteren in een onderzoeksverslag	<ul style="list-style-type: none"> • Afronden van het onderzoeksverslag • Meer missies programmeren, opnemen en documenteren 	<ul style="list-style-type: none"> • Eerste afgeronde versie van de programmacode • Ruw videomateriaal laten zien 	8	6
8	Lesluwe week (herfstreces)				
9	–	<ul style="list-style-type: none"> • Afmonteren van je projectfilmpje • Voorbereiden op individueel assessment 	–	–	6
10	–	<ul style="list-style-type: none"> • Projectassessment • Inleveren van projectmaterialen 	Definitieve deliverables op Blackboard archiveren	2	6
Subtotaal				58	54
Totaal SBU				112	

¹ Houd zelf in de gaten in welke kalenderweek een lesweek valt!

² Dit moet je laten zien of inleveren tijdens het ingeroosterde projectbegeleidingsmoment van de betreffende lesweek!

³ De richtlijnen voor je onderzoeksverslag leer je bij het vak "Onderzoek 1"

⁴ Het opstellen van PSDs leer je bij het vak "Inleiding Programmeren"

6. Benodigdheden

In de planning van Hoofdstuk 5 heb je kunnen zien dat je onder andere Lego Mindstorms onderdelen nodig hebt om je robot te bouwen, een camera nodig hebt om je resultaten te filmen en veel documentatie nodig hebt om uit te zoeken wat de *First*[®] Lego[®] League, Lego[®] Mindstorms[®] en MonoBrick nu precies inhouden. In dit hoofdstuk brengen we de lijsten van alle materialen, middelen en documentatie die je nodig hebt bij elkaar, zodat je makkelijk kunt vinden en doorgronden wat je nodig hebt om aan de slag te gaan.

6.1. Materialen

De opleiding verstrekt de volgende materialen, die herbruikbaar zijn en na het afronden van het project weer dienen te worden ingeleverd:

- een Lego[®]Mindstorms[®] educatieve basisset ([45544](#)) - dit zijn robotonderdelen;
- een Lego[®]Mindstorms[®] educatieve uitbreidingsset ([45560](#)) - dit zijn robotonderdelen;
- een Lego[®]Mindstorms[®] oplaadbare batterij ([45501](#)) zodat je niet telkens 6 AA batterijen hoeft te plaatsen;
- een Lego[®]Mindstorms[®] laadvoeding ([8887](#)) om de oplaadbare batterij mee op te laden;
- een USB kaartlezer en micro-SD kaart om de MonoBrick firmware op de EV3 brick te kunnen zetten;
- een type A naar mini-B USB-kabel om de EV3-brick bedraad met de PC te verbinden;
- een WNA1100 (of EW-7811UN) USB Wifi-dongle om de EV3-brick draadloos met de PC te verbinden (of gelijkwaardige dongle met Atheros AR9271 chipset);
- een *First*[®] Lego[®] League [wedstrijdtafel](#) met een jaarlijks wisselende inlegmat die de veldopbouw van het thema van het betreffende jaar ondersteunt;
- *First*[®] Lego[®] League kit met Lego[®] props (huisjes, boompjes en beestjes) ter ondersteuning van het thema (de “challenge”) van het betreffende jaar.

6.2. Middelen

Jij als student zorgt ervoor dat je beschikt over de volgende middelen:

- een werkstation waarop je software kunt ontwikkelen - als je zelf niet over een laptop beschikt, kun je gebruik maken van de PC's in het netwerklab (L0-39);
- één van de volgende zaken om een ad-hoc Wifi netwerk op te zetten als je via Wifi met de brick wilt verbinden:
 - ófwel een smartphone met de mogelijkheid voor een wifi-hotspot;
 - ófwel een laptop met daarin een Wifi-kaart die zowel in *listening mode* als in *access point mode* kan worden gezet;
 - ófwel een single board computer, zoals een Raspberry Pi, waarmee je zelf een draadloos access point maakt;
 - ófwel een (in onbruik geraakte) wifi-router of access point die je toevallig nog ergens hebt rondslingeren;
- een fototoestel + videocamera (camera op je smartphone is prima, maar als je daarover niet beschikt, kun je ook een Flip Camera lenen bij het [Servicepunt](#) van Inholland locatie Alkmaar) en videobewerkingssoftware (zie HS 5);
- webruimte waar je je eindfilmpje kunt hosten (YouTube, DropBox o.i.d.);
- de firmware en images die te vinden zijn op de website van [MonoBrick](#);
- een [RNDIS driver](#) als je de voorkeur geeft aan de USB-verbinding en gebruik maakt van Windows 10;
- *utilities* zoals [7-Zip](#), [Win32 Disk Imager](#), [SD Formatter](#) en [WinSCP](#), voor het uitpakken, disk-imagen, formatteren en bedienen van de SD-kaart firmware;
- *build & development tools* zoals het [EV3 Programma](#), [Xamarin Studio](#), het [Mono Framework](#) en de [Xamarin Add-in voor EV3](#).

6.3. Literatuur

De volgende documentatie heb je nodig om aan de slag te gaan:

- de documentatie van de *First*[®] Lego[®] League, te vinden op de [FLL website](#);
- de Lego[®]Mindstorms[®] [EV3 handleiding](#);
- [ontwikkelinstructies](#) en [installatie-instructies](#) voor [MonoBrick](#);
- de werk-instructie “Getting Started with MonoBrick”, te vinden op Blackboard - hierin staat beschreven hoe je de MonoBrick firmware installeert en hoe je een verbinding kunt opzetten tussen je ontwikkel-PC en de EV3 brick;
- voor het installeren en gebruiken van het zetsysteem [L^AT_EX](#) staan er referenties op Blackboard;
- klassikaal kan op de projectwiki op Blackboard kennis gedeeld worden.

7. Toetsing

Als je de planning in Hoofdstuk 5 volgt, ben je in staat om de leerdoelen te behalen die in Hoofdstuk 4 benoemd werden. We hebben je in Hoofdstuk 4 bovendien laten zien dat er vier leerdoelen zijn waarmee we je *professionele ontwikkeling* meten en dat er twee leerdoelen zijn waarmee we je *onderzoeksontwikkeling* meten. Dit hoofdstuk gaat in op de meting (toetsing) van deze zes leerdoelen. Door je inzicht te geven in de manier waarop de toetsing plaatsheeft en hoe zwaar elk onderdeel in de toetsing meeweegt, kun je er doelgericht naar toe werken.

In de toetsmatrijs in Bijlage B kun je zien hoe zwaar elk leerdoel meeweegt in de toetsing (kijk in de kolom “Weging”) en welke vragen ingezet worden om het leerdoel te meten (kijk in de kolom “Vragen”). De vragen zijn geduid met een codering, bijvoorbeeld A_1 of V_2 . Deze codering komt overeen met de vragen op het beoordelingsformulier, dat je kunt vinden in Bijlage A. Jouw examinerator zal dit formulier gebruiken om je te beoordelen. Tijdens het toetsmoment worden *alle* onderdelen van het toetsformulier beoordeeld. De planning van het toetsmoment zal tijdig op de “Berichten”-sectie van dit thema op Blackboard worden gepost. Om het beoordelingsformulier goed te begrijpen, moet je de volgende toelichting naast het formulier houden:

- Bij het onderdeel “Deliverables”
Er zijn drie voorwaarden waaraan je moet voldoen alvorens je aan de toetsing kunt deelnemen. De voorwaarden zijn geduid met codes D_1 t/m D_3 . *Zorg dat je jouw resultaat en verslag op Blackboard hebt gearcheveerd vóór je toetsmoment begint.* V.w.b. punt D_3 : het is verstandig om je robot in gebouwde toestand te houden met het oog op een eventuele herkansing, maar je cijfer wordt pas ingevoerd als je de geleende materialen gesorteerd hebt ingeleverd. **Dit doe je uiterlijk een week na het toetsmoment - daarna is de assessor genoodzaakt een onvoldoende te registreren voor je eerste toetspoging en krijg je na de herkansing nog één kans om de materialen gesorteerd in te leveren!**
- Bij het onderdeel “Assessment”
Tijdens het toetsmoment zal je een individueel assessment worden afgenomen, waarin je beoordeeld wordt op de punten A_1 t/m A_4 . Je verantwoordt mondeling je codetechnische besluiten en laat zien dat je bijdrage van voldoende omvang en door jezelf geschreven is. Voor het afnemen van een individueel assessment is voor programmeerprojecten binnen de opleiding TI een werkwijze opgesteld. Deze procedure vind je in Bijlage C. Voor deze specifieke beoordeling geldt een aanvullende regel:

7. Toetsing

- Als je geen code van anderen (bijv. van internet) hebt gebruikt (A_2), dan wordt je zelf geschreven code (A_1) dubbel meegerekend.
- Bij het onderdeel “Product / resultaat”

Tijdens het toetsmoment wordt het resultaat van je projectgroep (ook wel je “product” genoemd) beoordeeld op de punten P_1 t/m P_3 aan de hand van de gemoniteerde video-opname die je hebt gemaakt van je succesvol uitgevoerde missie(s). **Je scoort een nipte voldoende als je per groepslid precies één succesvol uitgevoerde missie kunt laten zien in je filmpje, waarbij je gebruik maakt van minimaal één motor en minimaal één sensor.** Je kunt extra punten verdienen als ...

 - ...je montage netjes is. Hieronder verstaan we dat je je als projectgroep voor de camera voorstelt, dat de montage goed is afgewerkt en dat het geluid (stemmen, geluidseffecten, muziek) goed verstaanbaar / toepasselijk is;
 - ...je in de opname laat zien dat je meerdere missies achter elkaar ononderbroken kunt uitvoeren - dit betekent dat je je robot zodanig hebt ontworpen en geprogrammeerd, dat deze niet na elke missie terugkeert naar de thuispositie om een hulpstuk te wisselen of opnieuw te worden uitgelijnd, maar dat je in één poging meerdere missies achter elkaar voltooit (minimaal 2 missies);
 - ...uit je opname blijkt dat je meer dan één missie per groepsdeelnemer hebt uitgevoerd, met een maximum van acht missies voor de gehele groep. Meer dan acht missies levert geen extra punten meer op.
- Bij het onderdeel “Onderzoeksverslag”

Tijdens het toetsmoment zal het onderzoeksverslag van je projectgroep worden beoordeeld op de punten V_1 t/m V_4 . Je assessors kijken of je verslag vormtechnisch in orde is¹ en of de verslaglegging van je experimenten zowel foto's van je robot als PSD's van elke uitgevoerde missie bevat. Voor de foto's geldt dat je van je robot voor-, zij-, achteraanzicht en hulpstukken duidelijk in beeld brengt. Het gebruik van L^AT_EX als typesetter (V_1) is optioneel, maar levert wel een hogere score op. Een onvoldoende beoordeling op één van de vragen V_2 , V_3 of V_4 levert automatisch een onvoldoende beoordeling voor dit onderdeel op.

Wanneer één of meerdere van deze vier hoofd-onderdelen als “onvoldoende” (of niet ontvankelijk) worden beoordeeld, krijg je een herstelmogelijkheid. Voor dit herstel wordt (op basis van de motivatie van de onvoldoende) afgesproken wat je moet doen om alsnog de beoordeling “voldoende” te behalen. Voor dit herstel geldt een maximale reparatietijd van twee weken. Aan het einde van deze twee weken moet je de verbeterde onderdelen opnieuw laten beoordelen door je assessoren. Zijn de onderdelen nu wel voldoende, dan is het project alsnog afgerond. Wanneer het alsnog onvoldoende is, zul je het gehele project over moeten doen in het volgende studiejaar.

¹Je verslag wordt inhoudelijk beoordeeld bij het vak “Onderzoeken 1”

8. Evaluatie

Aan het einde van het project, na het assessment, willen we je vragen om een oordeel te geven **over de projecthandleiding** door een aantal stellingen te scoren. Geef aan of je een stelling beoordeelt met -- (helemaal niet mee eens), - (niet mee eens), 0 (neutraal), + (mee eens) of ++ (helemaal mee eens).

Evaluatie projecthandleiding	--	-	0	+	++
Beleving					
Uit deze handleiding kon ik duidelijk opmaken hoe het project ging verlopen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deze handleiding is begrijpelijk en duidelijk geschreven	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ik vind de planning uit de handleiding realistisch / haalbaar	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Cognitie					
Dankzij deze handleiding had ik weinig begeleiding/hulp van anderen nodig	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uit deze handleiding werd mij duidelijk wat ik moest gaan doen / gebruiken / lezen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Uit deze handleiding kon ik duidelijk opmaken hoe ik beoordeeld ging worden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gedrag					
Deze handleiding heeft mijn projectaanpak beïnvloed	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deze handleiding heeft mij gestimuleerd om aan de slag te gaan	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deze handleiding gaf mij genoeg informatie om het project af te ronden	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Impact					
Ik heb anderen gewezen op deze handleiding	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deze handleiding heeft mij aangezet om meer samen te werken	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
De opzet uit deze handleiding ga ik een volgend project weer gebruiken	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Appendices

A. Beoordelingsformulier

Naam: _____

Studentnummer: _____

Datum: _____

Beoordeling Basis Robotica (Project)

Deliverables (deelnamevoorwaarden)		
D_1 : projectresultaat (code, filmlink) gearchiveerd op Blackboard	<input type="text" value="nee"/>	<input type="text" value="ja"/>
D_2 : onderzoeksverslag gearchiveerd op Blackboard	<input type="text" value="nee"/>	<input type="text" value="ja"/>
D_3 : geleende materialen zijn gesorteerd ingeleverd	<input type="text" value="nee"/>	<input type="text" value="ja"/>
Ontvankelijkheid (cijfer wordt ingevoerd)	<input type="text" value="nee"/>	<input type="text" value="ja"/>
Assessment (individueel)		
A_1 : toelichting & verantwoording van de zelf geschreven code	<input type="text" value="O"/>	<input type="text" value="T"/>
A_2 : toelichting & verantwoording van gebruikte code v. internet	<input type="text" value="O"/>	<input type="text" value="T"/>
A_3 : omvang codebijdrage (in verhouding tot projectomvang)	<input type="text" value="O"/>	<input type="text" value="T"/>
A_4 : eigenheid code (vergeleken met code van andere studenten)	<input type="text" value="O"/>	<input type="text" value="T"/>
Deelcijfer C_1	<input type="text"/>	
Product / resultaat (groepsbeoordeling)		
P_1 : video: netjes (voorstellen, montage, audio)	<input type="text" value="nee"/>	<input type="text" value="ja"/>
P_2 : video: meerdere missies ononderbroken uitgevoerd	<input type="text" value="nee"/>	<input type="text" value="ja"/>
P_3 : video: aantal succesvol uitgevoerde missies	<input type="text" value="maximaal 8"/>	
P_4 : groepsgrootte	<input type="text" value="1, 2 of 3"/>	
Deelcijfer C_2	<input type="text"/>	
Onderzoeksverslag (groepsbeoordeling)		
V_1 : \LaTeX is gebruikt als typesetter	<input type="text" value="nee"/>	<input type="text" value="ja"/>
V_2 : vormtechniek (titelblad, pag.numm., hs-numm., figuren/ta-bellen genummerd, inhoudsopgave, zakelijk - geen ik/wij)	<input type="text" value="O"/>	<input type="text" value="T"/>
V_3 : foto's: voor-, zij-, achteraanzicht + hulpstukken	<input type="text" value="O"/>	<input type="text" value="V"/>
V_4 : psd's: voor iedere uitgevoerde missie is een psd gemaakt	<input type="text" value="O"/>	<input type="text" value="V"/>
Deelcijfer C_3	<input type="text"/>	
Eindbeoordeling		
Eindcijfer $C_{eind} = \frac{C_1+C_2+C_3}{3}$	<input type="text"/>	

Legenda Kwalificaties: O = onvoldoende, T = twijfelachtig, V = voldoende, G = goed, E = excellent

B. Toetsmatrijs

Toetsmatrijs Cluster ICT

Opleiding: TI		Datum: 17-07-2014	
Studiejaar: 1		Onderwijsperiode: 1	
Onderwijsseenheid: Project Basis Robotica		Toetscode: 1914TI116A	
Eigenaar: A.M. Gieling			

Codering	Beheersingsindicator	Leerdoel	Weging ¹	Vragen ²	Niveau ³
Professioniseren 6.1.6	Eigen aanpak en handelen verantwoord	De student is in staat om zijn code (door hem zelf geschreven en/of gebruikt van anderen) toe te lichten en de gekozen oplossing te verantwoorden	$16\frac{2}{3} \%$	A ₁ voor 50% A ₂ voor 50%	T ⁴
Onderzoeken 7.1.7	Actief meewerken in een team	De student is in staat om codetechnisch een bijdrage aan het project te leveren die in verhouding staat tot de groepsomvang en het groepsresultaat	$16\frac{2}{3} \%$	A ₃ voor 50% A ₄ voor 50%	T ⁴
Professioniseren 6.1.8	Resultaatgericht werken, individueel en in teamverband	De student is in staat om samen met (een) andere student(en) de robot aantoonbaar een aantal opdrachten te laten uitvoeren	$33\frac{1}{3} \%$	P ₁ voor 5% P ₂ voor 10% P ₃ voor 85%	T ⁴
Professioniseren 6.1.1	Over proces en product adequaat rapporteren conform de richtlijnen	De student is in staat om een onderzoeksverslag netjes vorm te geven en zich daarbij te houden aan de richtlijnen voor rapporteren in een experimentele onderzoekscontext	20 %	V ₁ voor $16\frac{2}{3} \%$ V ₂ voor $83\frac{1}{3} \%$	T ⁴
Onderzoeken 7.1.8	Het gevonden resultaat overzichtelijk in beeld brengen	De student is in staat om het mechanisch ontwerp van de robot overzichtelijk in beeld te brengen	$6\frac{2}{3} \%$	V ₃ voor 100%	T ⁴
Professioniseren 6.1.10	Effectief communiceren rekening houdend met doelgroep en context	De student is in staat om zijn ontwerpbesluiten effectief over te brengen door gebruik te maken van een formele modelleertaal	$6\frac{2}{3} \%$	V ₄ voor 100%	T ⁴
De cesuur van de toets is 55 %					
Toetsvorm van deze toets: Individueel assessment + Videodemonstratie + Onderzoeksverslag					

¹ Het wegingspercentage is telkens aangegeven ten opzichte van 100% (de totale toets)

² Het percentage per vraag is het aandeel dat de vraag levert bij het meten van de onderhavige leereitkomst. Voor vraagcodes zie beoordelingsformulier

³ Met 'niveau' wordt bedoeld op de taxonomie van Bloom

⁴ 'T' staat voor het Bloom-niveau 'Toepassen'

C. Werkwijze Individueel Assessment

Gedurende het hele blok werk je in groepsverband aan een programmeeropdracht. In het individuele assessment wordt niet getoetst of je goed genoeg kunt programmeren: dat is eerder al gebeurd in een tentamen en/of praktijktoets. Wat wél wordt getoetst is of jij voldoende hebt bijgedragen aan het project. Ook wanneer je een onvoldoende behaalde voor het overeenkomstige programmeertentamen, kun je bij het individuele assessment een voldoende halen. Je zal er in overleg met de groep zelf voor moeten zorgen dat je een taak krijgt die je goed kunt uitvoeren.

1. Het individuele assessment vindt groepsgewijs plaats. Telkens wordt één student bevraagd op zijn of haar bijdrage, de andere groepsleden zijn daarbij aanwezig.
2. De tijd die per groep wordt besteed aan het assessment, is gelijk aan 10 minuten keer het aantal groepsleden, plus een extra 10 minuten voor het vaststellen van de cijfers.
3. Alléén de student die een vraag krijgt, mag antwoorden! Je mag elkaar — hoe goed bedoeld ook — *niet* ‘helpen’.
4. Elke student dient programmacode te laten zien die hij of zij zelf geschreven heeft en kan verdedigen en uitleggen. Onder programmacode verstaan we code geschreven in een programmeertaal. Opmaakstatements (bijv. HTML), vormgevingsexpressies (bijv. CSS), databasequeries (bijv. SQL) of tekst uit je verslag zijn dus **NIET** toelaatbaar als programmacode. Het programma wordt niet gedemonstreerd, alleen de code wordt bekeken. Welke code elke student voor zijn of haar rekening neemt, wordt van tevoren door de studenten zelf in overleg vastgesteld.
5. Componenten, methoden, variabelen e.d. dienen een zinvolle naam te hebben. Daar waar het relevant is dient programmacode gedocumenteerd te zijn.
6. Jullie dienen ervoor te zorgen dat je datgene wat je wilt laten zien, gereed hebt en hebt **gearchiveerd op Blackboard** op het moment dat het assessment begint. Het werken met een eigen of gezamenlijke laptop verdient de voorkeur.
7. De assessoren (twee docenten) bepalen samen een voorgesteld cijfer. Dat doen zij door eerst onafhankelijk een cijfer vast te stellen en daarna in overleg met jullie een gezamenlijk oordeel te geven.
8. Aan de groepsleden wordt vervolgens gevraagd welk cijfer ieder aan zichzelf en aan de anderen zou geven. Dit zogenoemde *peerassessment* wordt meegenomen in de definitieve vaststelling van het cijfer. Het eindoordeel ligt geheel in handen van de docenten, maar de praktijk leert dat het vrijwel altijd lukt om zo een eerlijk cijfer vast te stellen.
9. De betekenis van de cijfers is als volgt:
 - 5 of lager: je hebt onvoldoende bijgedragen aan het project
 - 6: je hebt waarschijnlijk voldoende bijgedragen aan het project, maar overtuigt niet in jouw presentatie (je krijgt dus het voordeel van de twijfel)
 - 7: je hebt zonder twijfel voldoende bijgedragen aan het project
 - 8: je hebt duidelijk meer dan gemiddeld bijgedragen aan het project

C. Werkwijze Individueel Assessment

Halve cijfers en cijfers hoger dan 8 wordt zelden toegekend; mocht iemand echt een bijzondere prestatie hebben neergezet, of is aantoonbaar dat diegene iets heeft geprogrammeerd waar de andere projectgroepen ook profijt van hebben gehad, dan wordt daar een 9 voor gegeven, in zeer uitzonderlijke gevallen een 10.

10. Studenten die een 4 of 5 behalen, krijgen binnen twee weken een tweede kans om te laten zien wat ze bijgedragen hebben. Wie aanzienlijk te weinig heeft bijgedragen, komt hiervoor niet in aanmerking. Het betreft vooral projectgroepen en individuele personen die tijdens het assessment nog niet ver genoeg blijken te zijn voor een voldoende cijfer. Een dergelijke tweede kans wordt ook aangeboden aan studenten die ziek zijn of om geldige redenen verhinderd.
11. Wie — ook na een eventueel tweede assessment binnen twee weken — nog steeds geen voldoende heeft, zal het project in zijn geheel over moeten doen. Het is dus niet mogelijk om later alsnog iets ter beoordeling in te leveren.