

**Bouwsteen
Embedded Systems 1
Microcontrollers en de taal C**

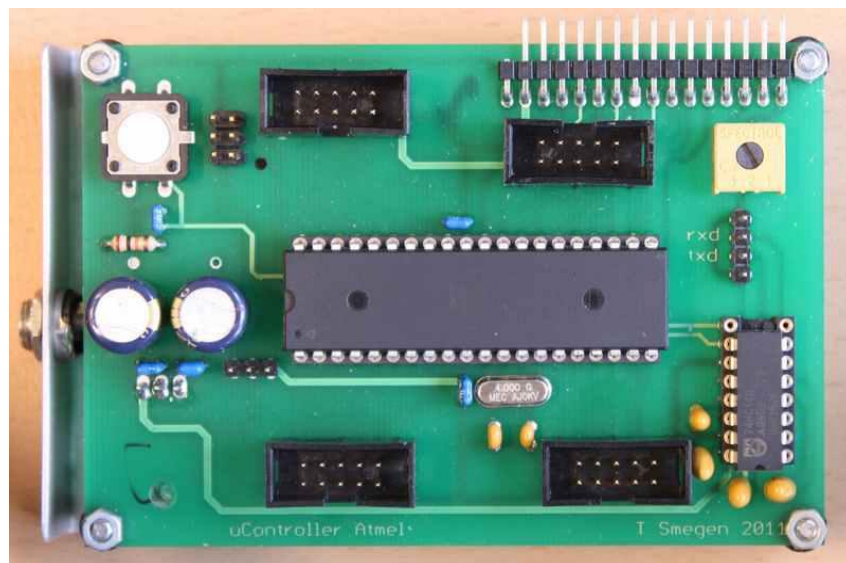
Studiejaar 2
Code ES1

Stenden University
Education
Van Schaikweg 94
Po box 2080
7801 CB Emmen
tel: 0591 853100
www.stenden.com

Embedded Systems 1

Microcontrollers en de taal C

Version 1.6: 8 juli 2013



Studiejaar 2013-2014

Bouwsteen coördinator:	Auteurs:
Naam: Bert Meijerink	Naam: Bert Meijerink
Code: Mk	Code: Mk
Room: 1.025	Room: 1.025
Tel: 0591-853270	Tel: 0591-853270
e-mail: bert.meijerink@stenden.com	e-mail: bert.meijerink@stenden.com

Voorwoord

In het eerste jaar heb je al kennis gemaakt met een microcontroller en wel met de AVR ATmega328. Je hebt toen software geschreven in een taal (nauw verwant aan C), die speciaal door Arduino is geschreven. De software werd door middel van een zogenaamde bootloader in de microcontroller geplaatst.

Zoals de ondertitel van deze bouwsteen al aangeeft, ga je je nu verder bekwaamen in het programmeren in de taal C. De taal C verscheen al in 1973 en is de meest gebruikte taal bij het programmeren van microcontrollers. C is een algemene procedurele programmeertaal en geen object georiënteerde taal, zoals Java of C++. C is effectief en snel en staat dicht bij de hardware. Je kunt je de taal C snel eigen maken, doordat de basis al is gelegd bij het programmeren in Java. Deze programma's zul je nu echter zonder bootloader rechtstreeks in de microcontroller plaatsen.

Ook wordt er veel aandacht besteed aan de mogelijkheden en de opbouw van microcontrollers. Kortom je leert hoe je een microcontroller kunt gebruiken. Dit alles doe je door een boek te bestuderen en daarnaast vooral door opdrachten uit te voeren. De reden daarvan is, dat het programmeren en het werken met een microcontroller een vaardigheid is, die je alleen maar onder de knie kunt krijgen door veel te oefenen.

Het primaire doel van deze bouwsteen is, dat je een goede theoretische basis krijgt met betrekking tot de mogelijkheden van microcontrollers en deze kunt toepassen in een hardwareomgeving. Na afloop van de bouwsteen heb je een basis gelegd voor een goede start van het project in de 2e periode van het 2e jaar en de vervolgbouwsteen Embedded systems 2 in de 3^e periode van het 2e jaar.

In het latere werkveld zul je als ICT-er Technische Informatica in de embedded systemen richting in contact komen met microcontrollers.

De studielast voor deze bouwsteen bedraagt 84 uur (3 EC).

Uit de verschillende toegepaste methoden op de Stenden hogeschool is in het verleden gebleken, dat de werkwijze van de bouwsteen Embedded systems 1 de meest geschikte methode is voor het aanleren van vaardigheden, die betrekking hebben op microcontrollers en een programmeertaal. Door evaluaties van studenten is het programma zodanig aangepast, dat er meer tijd beschikbaar is voor Embedded systems 1.

Bert Meijerink
Emmen, 8 juli 2013

Inhoud

1	INTRODUCTIE	7
1.1	BOUWSTEEN RATIONALE	7
1.2	COMPETENTIES	8
1.3	BOUWSTEEN THEMA	9
1.4	BOUWSTEEN DOELSTELLINGEN	9
1.5	VOORKENNIS	9
1.6	CONVENTIES	9
1.7	VERSIEBEHEER	10
2	TOETSING	11
2.1	ALGEMEEN	11
2.2	BEOORDELING EINDTOETS	11
2.3	ACTIEVE PARTICIPATIE	12
2.4	BEOORDELING WERKCOLLEGES	12
2.5	BOUWSTEEN HERKANSING	12
3	PROGRAMMA.....	13
3.1	INTRODUCTIE	13
3.1.1	<i>Zelfstudie</i>	<i>14</i>
3.1.2	<i>Werkcolleges.....</i>	<i>14</i>
3.2	PROGRAMMAOVERZICHT.....	15
3.3	WEEKPROGRAMMA'S	16
3.3.1	<i>Introductiecollege</i>	<i>16</i>
3.3.2	<i>Zelfstudie 1 (Boek hoofdstuk 1 t/m 9)</i>	<i>16</i>
3.3.3	<i>Zelfstudie 2 (Boek hoofdstuk 10 t/m 11).....</i>	<i>17</i>
3.3.4	<i>Werkcollege 1 (Boek hoofdstuk 10 t/m 11)</i>	<i>17</i>
3.3.6	<i>Werkcollege 2 (Boek hoofdstuk 10 t/m 11)</i>	<i>18</i>
3.3.7	<i>Zelfstudie 4 (Boek hoofdstuk 12)</i>	<i>18</i>
3.3.5	<i>Zelfstudie 3 (Boek hoofdstuk 10 t/m 11).....</i>	<i>18</i>
3.3.8	<i>Werkcollege 3 (Boek hoofdstuk 12)</i>	<i>19</i>
3.3.9	<i>Zelfstudie 5 (Boek hoofdstuk 13)</i>	<i>19</i>
3.3.10	<i>Werkcollege 4 (Boek hoofdstuk 13)</i>	<i>19</i>
3.3.11	<i>Werkcollege 5, 6 en 7.....</i>	<i>19</i>
4	STRUCTUUR & ORGANISATIE	20
4.1	BOUWSTEEN CONTACT UREN	20
5	LITERATUUR/PROGRAMMATUUR	21
5.1	VERPLICHTE LEERSTOF	21
5.2	NASLAGWERK.....	21
5.3	PROGRAMMATUUR.....	21
5.4	VOORBEELD PROGRAMMA'S	21
6	BOUWSTEEN EVALUATIE.....	22
7	BIJLAGEN.....	23

BIJLAGE 1:	AANVULLENDE LEERSTOF TALSTELSELS	24
B1.1	Voorstelling van gegevens.....	24
B1.2	Talstelsels.....	24
BIJLAGE 2:	INSTALLEREN EN GEBRUIK AVR STUDIO	27
BIJLAGE 3:	BESCHRIJVING MOEDERBOARD EN UITBREIDINGSPRINTEN.....	29
B3.1	Moederboard	29
B3.2	Uitbreidingsprint 1	31
B3.3	Uitbreidingsprint 2	33
BIJLAGE 4:	PRACTICUM 1	35
BIJLAGE 5:	PRACTICUM 2	36
BIJLAGE 6:	PRACTICUM 3	37
BIJLAGE 7:	PRACTICUM 4	38
BIJLAGE 8:	PRACTICUM 5	39
BIJLAGE 9:	PRACTICUM 6	40
BIJLAGE 10:	PRACTICUM 7	41
BIJLAGE 11:	INDIVIDUELE EINDOPDRACHT	42
BIJLAGE 12:	SCORING RUBRICS	45

1 Introductie

1.1 Bouwsteen rationale

Microcontrollers en (programmeren in) de taal C zijn basiselementen voor de ICT-er Technische Informatica. Het programmeren zal in dit geval aangeboden worden in de taal C, de standaard taal voor microcontrollers.

Door de grote diversiteit aan microcontrollers is het moeilijk een keuze te maken. Hier is gekozen voor de AVR ATmega32. Factoren die een rol hebben gespeeld, zijn de beschikbaarheid van (les) materiaal en wat er in het bedrijfsleven wordt gebruikt.

Kenmerkende beroepssituatie

Pieter werkt sinds een anderhalf jaar bij **Heat!** als embedded software engineer. Pieter heeft een HBO Technische Informatica opleiding achter de rug waar hij veel programmeerkennis heeft opgedaan evenals het werken in projecten.

Daarnaast heeft hij verschillende cursussen gevolgd om zijn kennis te verbreden zodat hij ook in staat is om (elektronische) hardware te ontwikkelen.

Als embedded software engineer wordt hij bij verschillende bedrijven ingezet om software te ontwerpen. Op dit moment doet Pieter dat voor een grote producent van lithografiemachines. In teamverband maakt hij software voor de besturing van een vacuümketel. Met zijn software kan de ketel volautomatisch op de juiste druk worden gebracht.

Maar Pieter heeft niet alleen te maken met software. Hij werkt in een multidisciplinair team, waarin bijvoorbeeld ook mechatronica en fysica een aandeel hebben. Ze maken samen één product, dus het werk van verschillende disciplines moet naadloos op elkaar aansluiten. Door het samenwerken steken zij veel van elkaar op.

Pieter kan zelfstandig en in teams werken. Hij kan communiceren met de verschillende partijen. Hij heeft een brede technische kennis en een goed inlevingsvermogen met betrekking tot wat de klant wil. Hij kan zich op verschillende gebieden snel inwerken en beheerst verschillende ontwerpmethoden, programmeertalen en ontwikkeltools om software te ontwikkelen.

1.2 Competenties

Binnen deze bouwsteen wordt als beginnend beroepsprofessional gewerkt aan een vijftal competenties die betrekking hebben op het analyseren, adviseren, ontwerpen, realiseren en beheren van hardware interfacing op niveau 1.

	Analyseren	Adviseren	Ontwerpen	Realiseren	Beheren
Gebruikers interactie					
Bedrijfsprocessen					
Software					
Infrastructuur					
Hardware interfacing	<ul style="list-style-type: none"> • Architectuur van een embedded systeem beschrijven. • Functionele specificaties voor een embedded systeem en een bijbehorende acceptatietest opstellen. <p>(niveau 1)</p>	<ul style="list-style-type: none"> • Een initiële architectuur op grond van een gegeven systeemconfiguratie bepalen en een technisch advies uitbrengen. <p>(niveau 1)</p>	<ul style="list-style-type: none"> • Een eenvoudig embedded systeem ontwerpen. <p>(niveau 1)</p>	<ul style="list-style-type: none"> • Een eenvoudig embedded systeem realiseren voorzien van actuatoren en sensoren, inclusief de hardware opleveren en de driversoftware schrijven en testen. <p>(niveau 1)</p>	<ul style="list-style-type: none"> • Een compileromgeving ten behoeve van een microprocessor en bijbehorende periferie inrichten. <p>(niveau 1)</p>

1.3 Bouwsteen thema

Binnen deze modulperiode is geen centraal thema aanwezig en is de bouwsteen Embedded Systems 1 een op zichzelf staand onderdeel.

1.4 Bouwsteen doelstellingen

Na afloop van deze bouwsteen kan de student:

1. globaal de mogelijkheden van een microcontroller aangeven;
2. de diverse onderdelen van een microcontroller benoemen;
3. de architectuur van de ATmega32 beschrijven;
4. een eenvoudig en efficiënt embedded systeem ontwerpen en realiseren;
5. een eenvoudig en efficiënt programma ontwerpen en realiseren in de taal C met de structuren die zijn behandeld bij de bouwstenen PHP en Java;
6. vanuit een gegeven situatie en probleemstelling een programma modulair ontwerpen;
7. de koppeling aanbrengen tussen de microcontroller en allerlei andere uitbreidingskaarten;
8. omgaan met de AVR Studio ontwikkelomgeving en bijbehorende software.

1.5 Voorkennis

De student dient:

1. de bouwsteen Stenden Duurzaam Weerstation (OPR04) met een voldoende te hebben afgerond.

1.6 Conventies

Een paar conventies met betrekking tot lettertypen zijn opgesteld om onderscheid te kunnen maken tussen tekst en programmeercode.

Normale tekst	Verdana. (tekengrootte 10)
Programma code	Courier new. (tekengrootte 10)
Programma output	Courier new. (tekengrootte 10)

1.7 Versiebeheer

Versie	Datum	Auteur	Omschrijving
1.0	22-08-2010	B. Meijerink	Initieel document
1.1	14-10-2010	B. Meijerink	Opname versiebeheer; Vervangen variabele pulsenmotor door toongenerator met TTL uitgang; Aanvulling individuele eindopdrachtdarcht m.b.t. problemen van code in de while lus in de main methode.
1.2	17-06-2011	B. Meijerink	Scoring rubrics aangepast.
1.3	12-09-2011	B. Meijerink	Ontwikkelomgeving aangepast naar versie 5.
1.4	23-10-2011	B. Meijerink	Ingangseisen aangepast; Nieuwe printen toegevoegd.
1.5	21-08-2012	B. Meijerink	Aanpassing data studiejaar.
1.6	08-07-2013	B. Meijerink	Aanpassing data studiejaar.

2 Toetsing

2.1 Algemeen

Om te beoordelen of je de doelstellingen van deze bouwsteen hebt behaald wordt er een individuele eindtoets (praktijktoets) gehanteerd.

Om te mogen deelnemen aan de eindtoets moeten alle practicumopdrachten uit de bouwsteen Embedded Systems 1 zijn afgetekend. Zie ook paragraaf 2.2.

Tabel 2.1 geeft een overzicht van de eindtoets met relevante informatie over de normering, punten en credits.

Tabel 2.1 Overzicht toetsing

Test method	Max. points	Norm %	Norm in points	Credits	Deadline	Resit
Praktijktoets	100	55%	55	3 EC		
Total	100	55%	55	3 EC		

2.2 Beoordeling eindtoets

De eindbeoordeling is een praktijktoets waar de student een programma moet maken achter de computer voor de microcontroller. Deze eindtoets zal dezelfde moeilijkheidsgraad hebben als de behandelde stof en zal binnen de gestelde tijd moeten worden afgerond.

Bij deze eindtoets mag de student gebruik maken van de volgende middelen: Boeken, bouwsteenboek, aantekeningen en handleidingen. **(Andere als genoemde middelen, zoals bijvoorbeeld Internet zijn niet toegestaan en niet beschikbaar.)**

De beoordeling van de praktijktoets zal plaatsvinden aan de hand van het beoordelingsformulier (Scoring Rubrics) welke te vinden is in bijlage 12.

2.3 Actieve participatie

De mogelijkheid voor het aftekenen van de practicumopdrachten heeft de student tijdens de ingeroosterde werkcolleges.

2.4 Beoordeling werkcolleges

Tijdens de werkcolleges zullen de volgende onderdelen beoordeeld worden:

1. De participatie van de student;
2. De practicumopdrachten;

Alle practicumopdrachten moeten met een voldoende zijn afgerond in de weken dat het desbetreffende werkcollege op het rooster (spoorboekje) staat.

(Slechts bij onvoorziene omstandigheden (ziekte etc.) kan hiervan afgeweken worden en na overleg met de docent.)

Deze 2 onderdelen moeten met een voldoende zijn beoordeeld voor men kan deelnemen aan de afsluitende eindtoets.

2.5 Bouwsteen herkansing

Studenten die niet slagen voor de praktijktoets, kunnen een herkansing doen.

3 Programma

3.1 Introductie

Microcontrollers en (programmeren in) de taal C zijn basiselementen voor de ICT-er Technische Informatica. Het programmeren zal in dit geval aangeboden worden in de taal C, de standaard taal voor microcontrollers.

Wij gaan werken met de AVR ATmega32 en gebruiken daarbij het boek "Microcontrollers en de taal C" van Wim Dolman. Bij dit boek hoort een website van de Hogeschool van Amsterdam met een groot aantal labs en oefenopdrachten. Een paar lab opdrachten moet je gaan uitvoeren. In het programmaoverzicht is aangegeven welke opdrachten je moet uitvoeren.

Het boek bevat een flink aantal behapbare hoofdstukken. De hoofdstukken 1 en 2 zijn een introductie van de Microcontroller en de taal C. De hoofdstukken 3 t/m 9 van het boek bevatten een groot aantal elementen, die jullie al gehad hebben bij het programmeren in Java. Door de hoofdstukken 1 t/m 9 kunnen jullie dan ook snel heen.

Naast de hierboven genoemde labs zijn er practicumopdrachten. Deze **practicumopdrachten** moeten alle **gemaakt** worden door de student **en afgetekend** worden door de docent.

Deze bouwsteen bevat een aantal (wekelijkse) werkcolleges. De student dient de hoofdstukken zelfstandig door te werken. Dit dient te gebeuren door middel van zelfstudie in combinatie met het werkcollege.

Tijdens de werkcolleges heeft de student de mogelijkheid om problemen met de opdrachten te bespreken met de docent en tevens kunnen dan opdrachten afgetekend worden.

Het geheel zal afgesloten worden met een praktijktoets.

3.1.1 Zelfstudie

Het is de bedoeling dat de student door middel van zelfstudie de verschillende onderdelen doorneemt en uitvoert welke in het programmaoverzicht te vinden zijn.

3.1.2 Werkcolleges

Het is de bedoeling dat de student door middel van zelfstudie de werkcolleges voorbereidt en daar eventuele vragen kan stellen. Ook worden werkcolleges gebruikt om de voortgang van de student te toetsen.

Tijdens de werkcolleges heeft de student de mogelijkheid om de onderdelen van de bijbehorende week af te laten tekenen. Het kan vanzelfsprekend wel eens voorkomen, dat een student het werk van een week niet heeft afgekregen en dit kan dan in de week daaropvolgend worden afgetekend. Het is echter niet mogelijk om aan het eind van de periode nog een groot gedeelte te laten aftekenen.

De docent is **voor delen** van het werkcollege beschikbaar.

Lab- en oefenopdrachten

In (en voorafgaand aan) de werkcolleges kunnen er zogenaamde Labs (lab opdrachten) en oefenopdrachten worden uitgevoerd. Deze zijn te vinden op de website van Wim Dolman. De rechtstreekse link is:

<http://oege.ie.hva.nl/~dolwe/mic/extra/index.php>

Practicumopdrachten

De practicum opdrachten zijn te vinden in bijlagen. Deze **practicumopdrachten** moeten alle **gemaakt worden en daarna worden afgetekend** door de docent.

3.2 Programmaoverzicht

In onderstaand overzicht is een overzicht te vinden van de wekelijkse activiteiten.

Wk	Taak nr.	Studie activiteit
1	3.3.1	Introductiecollege.
	3.3.2	Zelfstudie 1. Hoofdstuk 1 t/m 9.
2	3.3.3	Zelfstudie 2. Hoofdstuk 10 t/m 11.
	3.3.4	Werkcollege 1. Practicum 1 en 2.
3	3.3.5	Zelfstudie 3. Hoofdstuk 10 t/m 11.
	3.3.6	Werkcollege 2. Practicum 3.
4	3.3.7	Zelfstudie 4. Hoofdstuk 12
	3.3.8	Werkcollege 3. Practicum 4 en 5.
5	3.3.9	Zelfstudie 5. Hoofdstuk 13
	3.3.10	Werkcollege 4. Practicum 6 en 7
6	3.3.11	Werkcollege 5. Individuele eindopdracht.
7	3.3.11	Werkcollege 6. Individuele eindopdracht.
8	3.3.11	Werkcollege 7. Individuele eindopdracht.
9		Beoordeling en afronding eindtoets Embedded Systems 1. Deelname alleen toegestaan als alle practicumopdrachten zijn afgetekend.

3.3 Weekprogramma's

3.3.1 Introductiecollege	
Week:	1
Werkvorm:	Hoorcollege
Duur:	1
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt een overzicht van de inhoud van de bouwsteen Embedded Systems 1. • De student kent de werkwijze en de beoordeling binnen de bouwsteen Embedded Systems 1.
Inhoud:	Tijdens het introductiecollege krijg je instructie over onder andere de werkwijze/werkvormen, beoordeling, materiaal en inhoud van de bouwsteen Embedded Systems 1.
Vorbereiden:	
Individuele opdrachten:	

3.3.2 Zelfstudie 1 (Boek hoofdstuk 1 t/m 9)	
Week:	1
Werkvorm:	Zelfstudie
Duur:	n.v.t.
Lesdoelen:	<ul style="list-style-type: none"> • De student verkrijgt inzicht in een aantal basisbegrippen van de taal C.
Inhoud:	Tijdens deze zelfstudie wordt er aandacht besteed aan: <ul style="list-style-type: none"> • De microcontroller; • De taal C; • Declaraties; • Functies; • In- en uitvoer; • Talstelsels; • Voorwaardelijke opdrachten; • Herhalingsopdrachten; • Structuur en opmaak; • Datatypen en operatoren.
Vorbereiden:	Bestudeer hoofdstuk 1 t/m 9. Bestudeer de aanvullende leerstof uit bijlage 1.
Individuele opdrachten:	Bestudeer hoofdstuk 1 t/m 9. Bestudeer de aanvullende leerstof uit bijlage 1.

3.3.3 Zelfstudie 2 (Boek hoofdstuk 10 t/m 11)	
Week:	2
Werkvorm:	Zelfstudie
Duur:	n.v.t.
Lesdoelen:	<ul style="list-style-type: none"> De student verkrijgt inzicht in een aantal basisbegrippen van de ATmega32 microcontroller.
Inhoud:	Tijdens deze zelfstudie wordt er aandacht besteed aan: <ul style="list-style-type: none"> De ATmega32; Led Blink.
Voorbereiden:	Bestudeer hoofdstuk 10 t/m 11.
Individuele opdrachten:	Bestudeer hoofdstuk 10 t/m 11.

3.3.4 Werkcollege 1 (Boek hoofdstuk 10 t/m 11)	
Week:	2.
Werkvorm:	Werkcollege
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student installeert de AVR studio ontwikkelomgeving inclusief aanvullende software; De student oefent de met de ontwikkelomgeving AVR studio; De student leert structuren te gebruiken voor het opslaan van bestanden; De student leert C programma's te maken op de PC d.m.v. de AVR studio omgeving om deze vervolgens te kunnen runnen op een AVR microcontroller; De student leert hoe hij een tijdsafhankelijk element maakt.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> De ontwikkelomgeving AVR studio; De ATmega32; Led Blink.
Voorbereiden:	Installeer AVR studio zoals omschreven in bijlage 2. Bestudeer hoofdstuk 10 t/m 11.
Individuele opdrachten:	Installeer AVR studio zoals omschreven in bijlage 2. Bestudeer de aanvullende leerstof uit bijlage 3. Voer practicum 1 uit bijlage 4 uit. Voer practicum 2 uit bijlage 5 uit.

3.3.5 Zelfstudie 3 (Boek hoofdstuk 10 t/m 11)	
Week:	3
Werkvorm:	Zelfstudie
Duur:	n.v.t.
Lesdoelen:	<ul style="list-style-type: none"> De student verkrijgt inzicht in een aantal basisbegrippen van de ATmega32 microcontroller.
Inhoud:	Tijdens deze zelfstudie wordt er aandacht besteed aan: <ul style="list-style-type: none"> De ATmega32; Led Blink.
Voorbereiden:	Bestudeer hoofdstuk 10 t/m 11.
Individuele opdrachten:	Bestudeer hoofdstuk 10 t/m 11.

3.3.6 Werkcollege 2 (Boek hoofdstuk 10 t/m 11)	
Week:	3.
Werkvorm:	Werkcollege
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student installeert de AVR studio ontwikkelomgeving inclusief aanvullende software; De student oefent de met de ontwikkelomgeving AVR studio; De student leert structuren te gebruiken voor het opslaan van bestanden; De student leert C programma's te maken op de PC d.m.v. de AVR studio omgeving om deze vervolgens te kunnen runnen op een AVR microcontroller; De student leert hoe hij een tijdsafhankelijk element maakt.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> De ontwikkelomgeving AVR studio; De ATmega32; Led Blink.
Voorbereiden:	Bestudeer hoofdstuk 10 t/m 11.
Individuele opdrachten:	Voer practicum 3 uit bijlage 6 uit.

3.3.7 Zelfstudie 4 (Boek hoofdstuk 12)	
Week:	4
Werkvorm:	Zelfstudie
Duur:	n.v.t.
Lesdoelen:	<ul style="list-style-type: none"> De student verkrijgt inzicht in het interruptmechanisme van de ATmega32 microcontroller.
Inhoud:	Tijdens deze zelfstudie wordt er aandacht besteed aan: <ul style="list-style-type: none"> Interrupts.
Voorbereiden:	Bestudeer hoofdstuk 12.
Individuele opdrachten:	Bestudeer hoofdstuk 12.

3.3.8 Werkcollege 3 (Boek hoofdstuk 12)	
Week:	4.
Werkvorm:	Werkcollege
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student leert hoe het interruptmechanisme werkt.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Interrupts
Voorbereiden:	Bestudeer hoofdstuk 12.
Individuele opdrachten:	Voer practicum 4 uit bijlage 7 uit. Voer practicum 5 uit bijlage 8 uit.

3.3.9 Zelfstudie 5 (Boek hoofdstuk 13)	
Week:	5
Werkvorm:	Zelfstudie
Duur:	n.v.t.
Lesdoelen:	<ul style="list-style-type: none"> De student verkrijgt inzicht in de verschillende timers van de ATmega32 microcontroller.
Inhoud:	Tijdens deze zelfstudie wordt er aandacht besteed aan: <ul style="list-style-type: none"> Timers.
Voorbereiden:	Bestudeer hoofdstuk 13.
Individuele opdrachten:	Bestudeer hoofdstuk 13.

3.3.10 Werkcollege 4 (Boek hoofdstuk 13)	
Week:	5.
Werkvorm:	Werkcollege
Duur:	2 uur.
Lesdoelen:	<ul style="list-style-type: none"> De student leert hoe de verschillende timers werken.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Timers.
Voorbereiden:	Bestudeer hoofdstuk 13.
Individuele opdrachten:	Voer practicum 6 uit bijlage 9 uit. Voer practicum 7 uit bijlage 10 uit.

3.3.11 Werkcollege 5, 6 en 7	
Week:	6, 7 en 8.
Werkvorm:	Werkcollege
Duur:	2 uur elk.
Lesdoelen:	<ul style="list-style-type: none"> De student kan openstaande onderdelen afronden.
Inhoud:	Tijdens dit werkcollege wordt er aandacht besteed aan: <ul style="list-style-type: none"> Openstaande onderdelen uit de weken 1 t/m 6.
Voorbereiden:	
Individuele opdrachten:	Voer de individuele eindopdracht uit bijlage 11 uit.

4 Structuur & Organisatie

4.1 Bouwsteen contact uren

Het onderstaande schema geeft een overzicht van alle contacturen in deze bouwsteen.

Daarnaast wordt van studenten verwacht, dat zij hun eigen (project)bijeenkomsten plannen, waar zij kunnen werken aan de opdrachten. Dit geldt tevens voor de tijd, die de student nodig heeft om (individuele) opdrachten voor te bereiden en te maken. Tevens geeft dit schema een goed overzicht van de verwachte studiebelasting per student.

Table 4.1: Student contact uren (SCU) per week:

Bouwsteen Embedded Systems 1: Student Contacturen en Student Belastinguren																					
Werkvorm	Aantal groepen	Week 1		Week 2		Week 3		Week 4		Week 5		Week 6		Week 7		Week 8		Week 9		Totaal	
		SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU	SCU	SBU
HC	4	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1,00	3,00
WC	2	2	6	2	6	2	6	2	6	2	6	2	6	2	6	2	6	0	0	16,00	48,00
ET	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	9	3,00	9,00
ZS	1	0	0	0	3	0	3	0	3	0	3	0	0	0	0	0	0	0	0	-	12,00
		3	9,00	2	9,00	2	9,00	2	9,00	2	9,00	2	6,00	2	6,00	2	6,00	3	9,00	20,00	72,00

HC = Hoorcollege
 WC = Werkcollege
 ET = Eind toets

SCU = Student Contact Uren (45 minuten)
 SBU = Studie Belasting Uren (60 minuten)

5 Literatuur/programmatuur

5.1 Verplichte leerstof

- Dit bouwsteenboek
- Boek:
Dolman, Wim (2009)
Microcontrollers en de taal C, Derde druk.
Culemborg: Hogeschool van Amsterdam
- ISBN bestelgegevens
ISBN-10: -.
ISBN-13: 978-90-484-0835-1.

5.2 Naslagwerk

- Wim Dolman website: <http://mic.dolman-wim.nl/>
- AVR 8-bits microcontrollers:
http://www.atmel.com/dyn/products/product_card.asp?part_id=2014

5.3 Programmatuur

- Netbeans programmeer omgeving: <http://www.netbeans.org/>
- MinGW GNU compiler: <http://www.mingw.org/>
- AVR Studio:
http://www.atmel.com/microsite/avr_studio_5/default.asp?source=redirect

5.4 Voorbeeld programma's

- Voorbeeldprogramma's zijn te vinden onder de course Embedded Systems 1.

6 Bouwsteen evaluatie

De bouwsteen zal worden geëvalueerd door middel van een vragenlijst aan het eind van de bouwsteen. Deze vragenlijst bevat alle onderdelen van de bouwsteen inclusief organisatorische aspecten, inhoud, kwaliteit van onderwijzend personeel, etc.

Wij willen je vriendelijk verzoeken deel te nemen aan deze evaluatie. De resultaten van deze evaluatie worden gebruikt om de volgende versie van deze bouwsteen te verbeteren.

7 Bijlagen

Bijlage 1: Aanvullende leerstof talstelsels

Bij het programmeren van de AVR-microcontroller komen de volgende talstelsels kijken: Decimaal, Binair en Hexadecimaal.

B1.1 Voorstelling van gegevens

Bij het voorstellen van gegevens maakt een computer gebruik van het binaire talstelsel. Hierbij worden de volgende begrippen gebruikt:

- Een **bit** is de grootte van een elementaire geheugencel. Hij kan één binair (tweewaardig, 0 of 1) cijfer bevatten.
- Om een teken (letter, cijfer, leesteken) op te slaan heeft men acht bit nodig. Zulk een eenheid van 8 bit wordt een **byte** (samentrekking van **by eight**) genoemd. Elke bit van een byte kan twee waarden aannemen, zodat een byte $2^8 = 256$ verschillende waarden kan hebben. Dit volstaat in het algemeen om alle gebruikelijke tekstuele tekens op te slaan.

Positieve gehele getallen, negatieve gehele getallen en reële (niet-gehele) getallen worden op een verschillende wijze voorgesteld. Omdat een computer een beperkt geheugen heeft kan hij geen enkele van deze drie klassen getallen volledig voorstellen. Steeds zal er een kleinste en een grootste getal zijn, dat kan worden voorgesteld. Valt het resultaat van een berekening buiten deze grenzen, dan treedt overflow op. Daarenboven zullen vele niet-gehele getallen slechts benaderd kunnen worden voorgesteld (dit is ook het geval in het decimale talstelsel).

B1.2 Talstelsels

De moderne mens stelt getallen voor in het **decimale** talstelsel, waarbij 10 cijfers worden gebruikt (0 tot 9).

De bij computers gebruikte technologieën (vroeger schakelaars en lampen, nu transistors en magneten) kunnen het best grootheden met twee waarden (aan/uit) voorstellen. Zij maken daarom gebruik van het tweewaardige of **binaire** talstelsel, dat werd ontwikkeld door Gottfried Leibnitz in de 17de eeuw. In het binaire talstelsel worden enkel de cijfers 0 en 1 gebruikt.

Het binaire talstelsel is juist zoals het decimale talstelsel een **positioneel talstelsel**. Dit betekent dat de waarde van een cijfer bepaald wordt door zijn plaats.

Zo hebben de cijfers in het decimaal voorgestelde getal **1999** respectievelijk de waarden 1000 (1 maal 10^3), 900 (9 maal 10^2), 90 (9 maal 10^1) en 9 (9 maal 10^0).

In het **binaire** talstelsel wordt het getal 1999 voorgesteld door **111 1100 1111**. De binaire cijfers 1 hebben hier respectievelijk de waarden 1024 (2^{10}), 512 (2^9), 256 (2^8), 128 (2^7), 64 (2^6), 8 (2^3), 4 (2^2), 2 (2^1) en 1 (2^0), waarbij $1024 + 512 + 256 + 128 + 64 + 8 + 4 + 2 + 1 = 1999$.

Alle moderne talstelsels zijn positioneel, dit in tegenstelling tot het Romeinse talstelsel.

Naast het binaire en het decimale talstelsel worden in de informatica ook nog het **hexadecimale** en het **BCD** (binary coded decimal) talstelsel gebruikt.

In het zestienwaardige of **hexadecimale** talstelsel worden zestien cijfers gebruikt. Naast de cijfers 0 tot 9 worden ook de hoofdletters A tot F gebruikt, met als waarden

10 tot 15. De positie van een hexadecimaal cijfer bepaalt de macht van 16 waarmee het moet worden vermenigvuldigd.

In het **hexadecimale** talstelsel wordt het getal 1999 voorgesteld door **7CF**. De hexadecimale cijfers hebben hier respectievelijk de waarden 1792 (7 maal 162), 192(12 maal 161) en 15 (15 maal 160), waarbij $1792 + 192 + 15 = 1999$. Het binaire talstelsel heeft als nadeel dat getallen snel erg veel cijfers bevatten en daardoor onleesbaar worden. Omdat met elk groepje van vier binaire cijfers steeds exact één hexadecimaal cijfer overeenkomt is de omzetting van een binair getal naar een hexadecimaal getal erg eenvoudig. Doordat hexadecimale getallen 16 verschillende cijfers kunnen gebruiken nemen ze daarenboven weinig plaats in, zelfs minder dan decimale getallen.

Het **BCD** talstelsel is een variant van het (zuivere) binaire talstelsel, waarbij elk cijfer van een decimaal getal (0 tot 9) wordt omgezet in de vier binaire cijfers die hiermee overeenkomen. Op deze wijze is de omzetting van een decimaal naar een BCD getal en omgekeerd even eenvoudig als die van een hexadecimaal naar een zuiver binair getal. Het rekenen met BCD getallen is voor de computer echter moeilijker dan het rekenen met zuiver binaire getallen. Daarentegen zijn BCD getallen nog langer dan zuiver binaire.

Decimaal	BCD	Hexadecimaal	Binair (zuiver)
0	0000	0	0
1	0001	1	1
2	0010	2	10
3	0011	3	11
4	0100	4	100
5	0101	5	101
6	0110	6	110
7	0111	7	111
8	1000	8	1000
9	1001	9	1001
10	0001 0000	A	1010
11	0001 0001	B	1011
12	0001 0010	C	1100
13	0001 0011	D	1101
14	0001 0100	E	1110
15	0001 0101	F	1111
16	0001 0110	10	10000
17	0001 0111	11	10001
18	0001 1000	12	10010
19	0001 1001	13	10011
20	0010 0000	14	10100

tabel b1.1: Voorstelling van de getallen 0 tot 20 in de verschillende talstelsels

In het **BCD** talstelsel wordt het getal 1999 voorgesteld door 0001 1001 1001 1001. De binaire cijfers hebben hier per groepje van vier de waarden 1000 (1 maal 10^3 ; 0001 binair = 1 decimaal), 900 (9 maal 10^2 ; 1001 binair = 9 decimaal), 90 (9 maal 10^1) en 9 (9 maal 10^0).

Om de leesbaarheid van getallen te vergemakkelijken groepeer men decimale getallen per drie cijfers, binaire en BCD getallen per vier cijfers, en hexadecimale getallen (soms) per twee cijfers. Hierbij worden de cijfers steeds van rechts naar links gegroepeerd (bij gehele getallen).

In tabel b1.1 worden getallen 0 tot 20 in de vier talstelsels weergegeven.

Merk hierbij op dat niet-significante nullen vooraan op de gebruikelijke wijze kunnen worden weggelaten bij decimale, hexadecimale en (zuiver) binaire getallen, maar dat bij BCD getallen elk groepje van vier binaire cijfers meestal expliciet wordt vermeld.

In tabel b1.2 worden een aantal machten van twee en zestien in het decimale, hexadecimale en binaire talstelsel weergegeven. Dit is interessant bij de omzetting van grote getallen. Daarenboven worden ook de uitdrukkingen met behulp van de symbolen voor kilo (1 **K** = 2^{10}), mega (1 **M** = 2^{20}) en giga (1 **G** = 2^{30}) vermeld (zie ook infra).

macht van 2		decimaal	hexadecimaal	(zuiver) binair
2^0	1	1	1	1
2^1	2	2	2	10
2^2	4	4	4	100
2^3	8	8	8	1000
2^4	16	16	10	1 0000
2^5	32	32	20	10 0000
2^6	64	64	40	100 0000
2^7	128	128	80	1000 0000
2^8	256	256	1 00	1 0000 0000
2^9	512	512	2 00	10 0000 0000
2^{10}	1 K	1 024	4 00	100 0000 0000
2^{11}	2 K	2 048	8 00	1000 0000 0000
2^{12}	4 K	4 096	10 00	1 0000 0000 0000
2^{13}	8 K	8 192	20 00	10 0000 0000 0000
2^{14}	16 K	16 384	40 00	100 0000 0000 0000
2^{15}	32 K	32 768	80 00	1000 0000 0000 0000
2^{16}	64 K	65 536	1 00 00	1 0000 0000 0000 0000
2^{20}	1 M	1 048 576	10 00 00	1 0000 0000 0000 0000 0000
2^{24}	16 M	16 777 216	1 00 00 00	1 0000 0000 0000 0000 0000 0000

tabel b1.2: Voorstelling van een aantal machten van 2 in de verschillende talstelsels

Bijlage 2: Installeren en gebruik AVR Studio

Om programma's te kunnen schrijven, compileren en uploaden in de AVR microcontroller heb je aanvullende (gratis) software nodig.

De software die je nodig bent is:

- AVR Studio, de ontwikkelomgeving;
(http://www.atmel.com/microsite/avr_studio_5/default.asp?source=redirect)

Deze software is vrij beschikbaar en gewoon te downloaden vanaf internet. Ook zijn er verschillende handleidingen van deze softwarepakketten te vinden op het Internet.

Download de **meest recente** versies van bovenstaand pakket en installeer deze. Let hierbij op het volgende:

- Ten tijde van het schrijven van dit boek werkte AVR Studio 5 nog niet helemaal goed. Dit heeft onder meer uit te staan met de programmeeradapter (mySmartUSB light). Let er goed op dat je gelijk de juiste seriële poort selecteert. Doe je dat niet dan kun je dat helaas niet wijzigen en moet AVR Studio 5 uninstallen en opnieuw installeren.
Om dit te voorkomen kun je het beste even de programmeeradapter in je PC steken en dan kijken via het Control Panel (Hardware and Sound -> Devices and Printers) en kijken op welke poort hij zit.
Ga vervolgens naar AVR Studio en kies daar Tools. Vervolgens kun je daar kiezen Add STK500 en kies daar de poort die je eerder hebt gevonden.

Vervolgens kun je een project aanmaken. Op het volgende Youtube filmpje (<http://www.youtube.com/watch?v=uUwythZLASg&feature=related>) is uitstekend te zien hoe je dit doet. Je moet echter het volgende anders doen:

- Na 47 seconden is op het filmpje te zien dat de optimalisatie wordt gewijzigd in -Os. **Doe dit echter niet. Laat hem op -O0 (None, geen optimalisatie) staan.**
- De optimalisatie wil soms nog wel eens wat roet in het eten gooien. Bij latere opdrachten kan het nodig zijn om hem **juist wel in te schakelen**, bijvoorbeeld op -Os.

Denk daarnaast nog aan het volgende:

- Het is, om een goed overzicht te houden van belang dat je een duidelijk structuur aanhoudt in het opslaan van bestanden. Wij eisen dan ook dat je een bepaalde werkwijze hanteert.
Maak op de schijf waar je normaal je bestanden opslaat (wij gaan nu even uit van de C-drive) een directory aan met de naam **EMBSYS_1**. In deze directory moet je alle projecten opslaan die bij deze bouwsteen horen. Maak

voor elke opdracht steeds een nieuw project aan met een zinvolle naam. Zorg er voor dat het eerste project met 01_ begint, het tweede project met 02_ etc.

Als laatste kun je het programma uploaden in de microcontroller. Dit kun je als volgt doen:

- Uploaden is niet zo lastig. Kies bovenin **Project** en vervolgens 01_Lab7 Properties en zorg er voor dat onder **debugging** bij de debugger de STK500 is gekozen. Vervolgens kun je op **ALT+F5** drukken (of het blauwe pijlke) en het programma wordt geladen in het kitje.

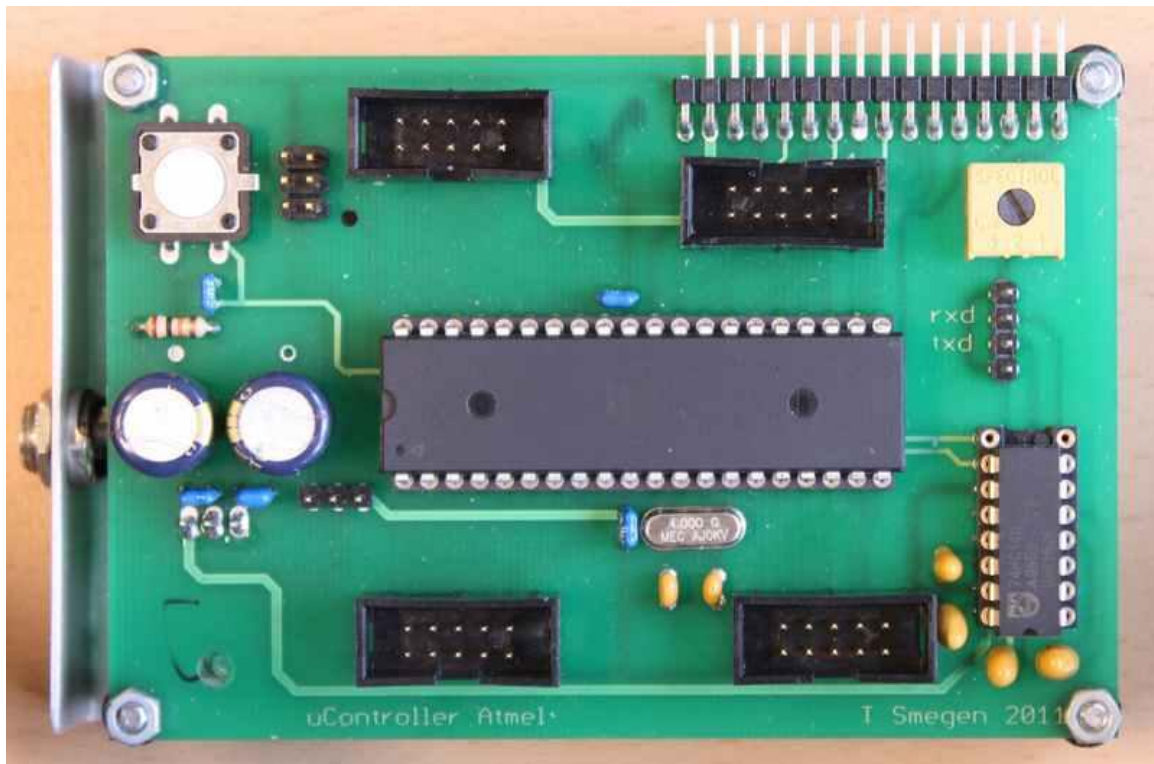
Bijlage 3: Beschrijving moederboard en uitbreidingsprinten

Er zijn een aantal uitbreidingskaarten aanwezig voor het moederboard. In deze bijlage kun je de functies en mogelijkheden zien van de verschillende printen.

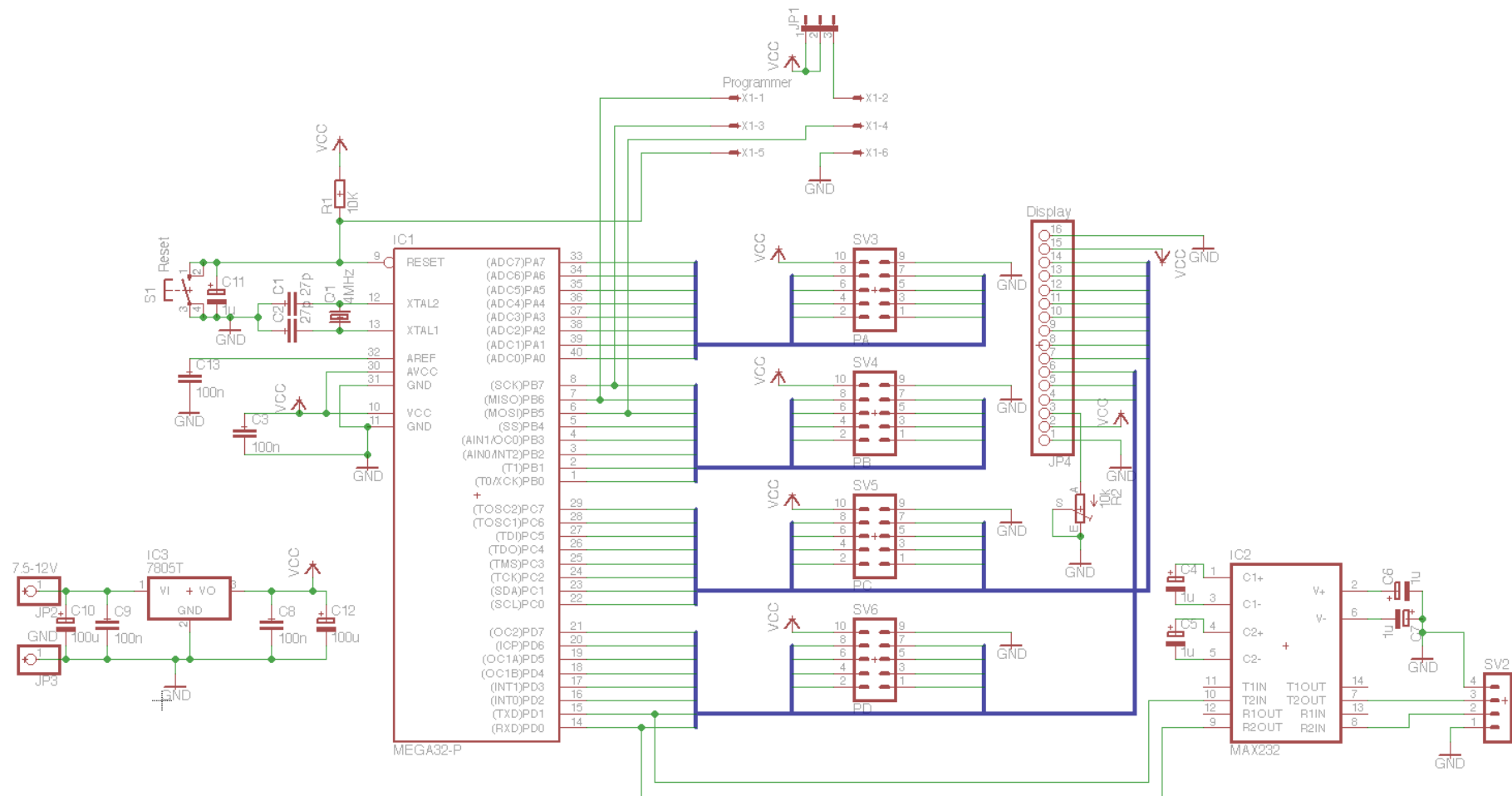
B3.1 Moederboard

Een foto van het moederboard is te vinden in figuur b3.1 en het logisch schema van het moederboard is te vinden in figuur b3.2. Op dit moederboard zijn onder andere aanwezig:

- De ATmega32 microcontroller;
- Voeding;
- Oscillator;
- RS232 convertor;
- Diverse headers waarop de uitbreidingen kunnen worden aangesloten.



Figuur b3.1



Figuur b3.2

B3.2 Uitbreidingsprint 1

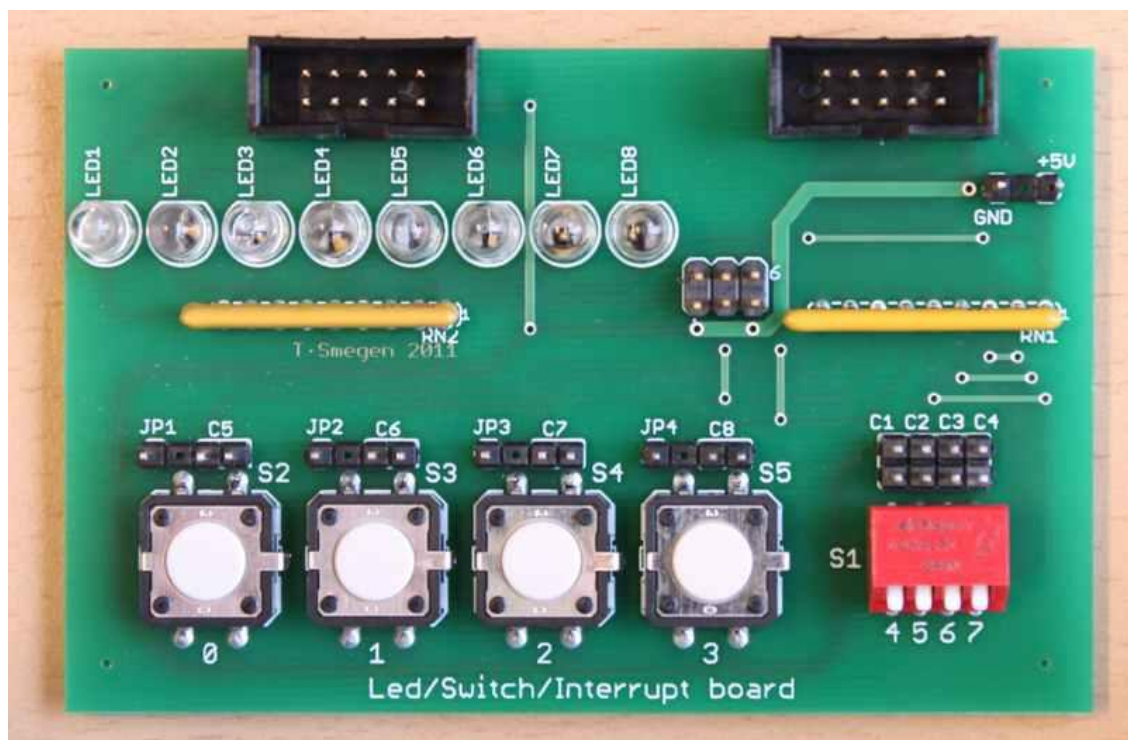
Uitbreidingsprint nummer 1 betreft een print met 8 LED's, 4 dipswitches en 4 drukknopjes. Daarnaast zijn de ingangen waar de drukknopjes op zitten via 2 mm contactbussen naar buiten uitgevoerd worden.

Specificaties:

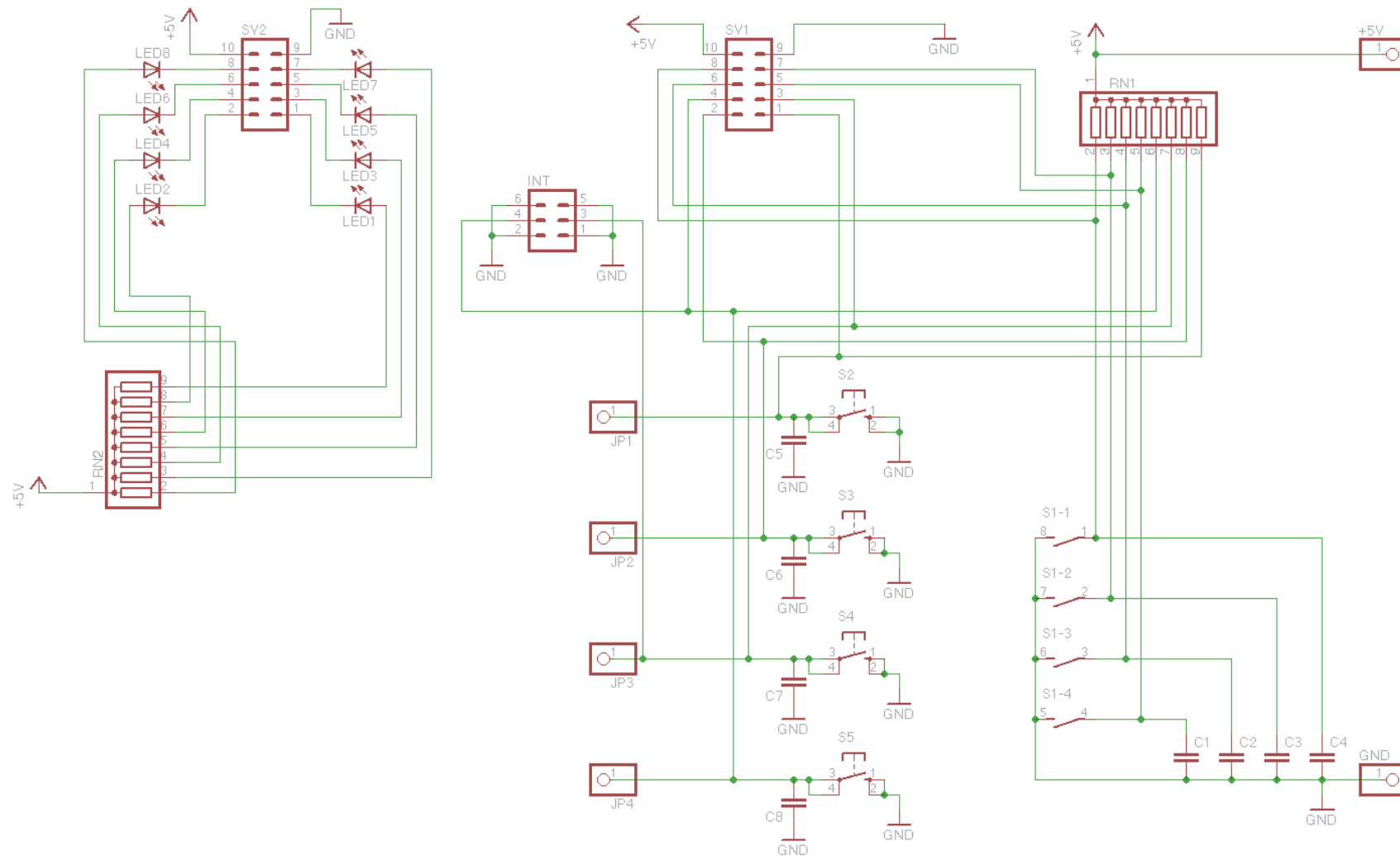
De LED's worden laag actief aangestuurd. De ingangen via de drukknopjes liggen via een weerstand aan de voedingsspanning. Hierdoor is ook dit signaal laag actief.

Afbeeldingen:

Een foto van uitbreidingsprint 1 is te vinden in figuur b3.3 en het logisch schema van uitbreidingsprint 1 is te vinden in figuur b3.4.



Figuur b3.3



Figuur b3.4

B3.3 Uitbreidingsprint 2

Uitbreidingsprint nummer 2 bevat 4*7-segmentsdisplays. En 4 LED's.

Specificaties:

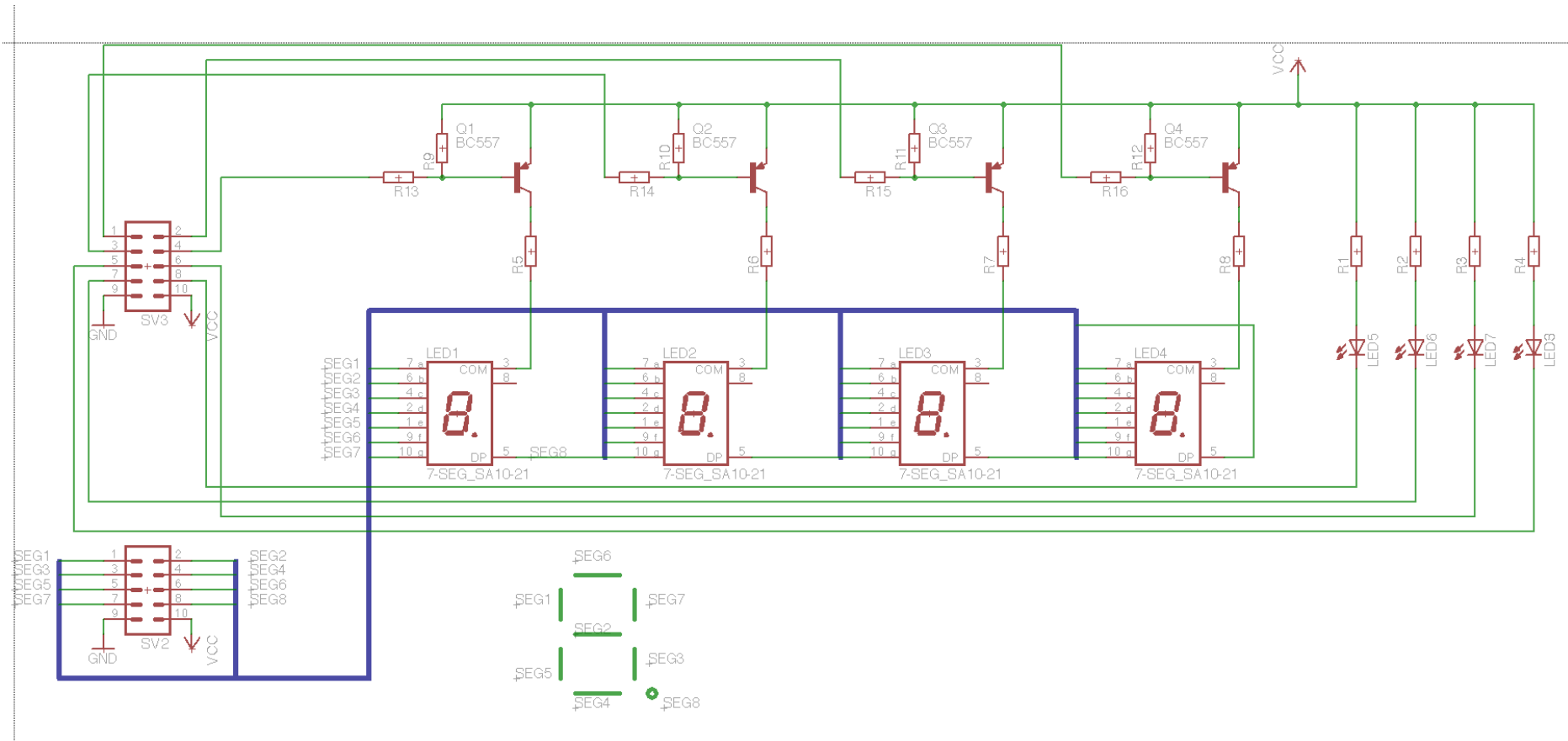
Alle datalijnen moeten laag actief aansturen.

Afbeeldingen:

Een foto van uitbreidingsprint 2 is te vinden in figuur b3.5 en het logisch schema van uitbreidingsprint 2 is te vinden in figuur b3.6.



Figuur b3.5



Figuur b3.6

Bijlage 4: Practicum 1

Bij dit eerste practicum ga je output bedrijven met de microcontroller.

Na dit practicum kan de student:

- Eenvoudige programma's schrijven voor de ATmega32;
- Eenvoudige programma's debuggen;
- Programma's in de ATmega32 laden;
- Poorten aansturen.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;

Uitvoering:

Plaats linker connector uitbreidingsprint 01 op poort A van het moederboard.

Jullie moet bij deze opgave een opdracht uitvoeren van de Website van Wim Dolman. Deze opdracht is te vinden op website van Wim Dolman -> Microcontrollers -> labs -> Micro-C -> Lab7. (<http://oege.ie.hva.nl/~dolwe/mic/labs/lab7.php>)

Voer de opdrachten uit zoals beschreven op de website. Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 5: Practicum 2

Bij dit tweede practicum ga je IO bedrijven met de microcontroller.

Na dit practicum kan de student:

- Poorten uitlezen;
- Poorten aansturen;
- Met verschillende lussen werken;
- Tijdvertragingen maken;
- Met bitoperaties werken (AND, OR, etc).

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;

Uitvoering:

Het is de bedoeling dat de student een rij LED's aanstuurt. Met de dipswitches moet de gewenste modus kunnen worden ingesteld. Voor de mode geldt:

- Mode 00: LEDs 'bewegen' van links naar rechts (Er brandt steeds maar 1 LED.);
- Mode 01: als mode 00, maar van rechts naar links (Er brandt steeds maar 1 LED.);
- Mode 10: Modes 00 & 01 om en om. Zodra de rechterkant is bereikt gaan de LED's weer aan naar links en zodra de linkerkant is bereikt gaan de LED's weer aan naar de rechterkant; (Er brandt steeds maar 1 LED.)
- Mode 11: Eigen idee.

De tijdvertraging in het programma moet op een nette manier zijn uitgevoerd dus niet met eigengemaakt lusconstructie.

Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 6: Practicum 3

Bij dit derde practicum ga je meerdere cijfers op een display zetten door middel van multiplexen.

Na dit practicum kan de student:

- 7-segment displays aansturen;
- Multiplexen;

Benodigd:

- Moederboard;
- Uitbreidingsprint 02, 4 * 7-segmentsdisplay;

Uitvoering:

Uitbreidingsprint 02 bevat 4 * 7-segmentsdisplays. Toon de cijfers 8 7 6 5 naast elkaar op de displays.

Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 7: Practicum 4

Bij dit vierde practicum ga je externe interrupts verkennen met de microcontroller.

Na dit practicum kan de student:

- Externe interrupts verwerken in een interrupt service routine.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;

Uitvoering:

Plaats linker connector uitbreidingsprint 01 op poort A van het moederboard.

Plaats rechter connector uitbreidingsprint 01 op poort D van het moederboard.

Drukknop met nummer 2 zit aangesloten op INT0. Deze kun je gebruiken voor deze opdracht.

Jullie moet bij deze opgave een opdracht uitvoeren van de Website van Wim Dolman. Deze opdracht is te vinden op website van Wim Dolman -> Microcontrollers -> labs -> Micro-C -> Lab9. (<http://oege.ie.hva.nl/~dolwe/mic/labs/lab9.php>)

Voer de opdrachten uit zoals beschreven op de website. Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 8: Practicum 5

Bij dit vijfde practicum ga je een pulsenteller maken met een 4 * 7-segmentsdisplay.

Na dit practicum kan de student:

- 4 * 7-segment displays aansturen;
- Externe interrupts verwerken in een interrupt service routine.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;
- Uitbreidingsprint 02, 4 * 7-segmentsdisplay + 4 LED's;
- Eventueel toongenerator met TTL uitgang.

Uitvoering:

De pulsen dienen op een interrupt ingang worden ingelezen waarmee een interne teller wordt opgehoogd of verlaagd afhankelijk van de ingestelde modus. De waarde van de teller moet op het 4 * 7-segmentsdisplay worden weergegeven. Met de dipswitches moet de gewenste modus kunnen worden ingesteld. Voor de mode geldt:

- Mode 00: De teller moet optellen. Zodra de waarde 9999 is bereikt moet de teller weer op 0 beginnen;
- Mode 01: De teller moet aftrekken. Zodra de waarde 0 is bereikt moet de teller weer op 9999 beginnen;
- Mode 10: De teller moet zich bewegen tussen 0 en 100. Dit houdt in dat als de teller de waarde 100 heeft bereikt dat hij weer moet aftrekken en zodra de teller de waarde 0 heeft bereikt moet hij weer optellen.
- Mode 11: Eigen idee.

Daarnaast geldt dat er geen onnodige nullen getoond mogen worden. Al je dus bijvoorbeeld 57 moet tonen dan mag er geen 0057 staan. De intensiteit van de displays mag niet variëren.

Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Om snel pulsen achter elkaar te maken kun je gebruik maken van de toongenerator met TTL uitgang.

Bijlage 9: Practicum 6

Bij dit zesde practicum ga je timers verkennen met de microcontroller.

Na dit practicum kan de student:

- Tijdsafhankelijke elementen maken d.m.v. timers.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;

Uitvoering:

Plaats linker connector uitbreidingsprint 01 op poort A van het moederboard.

Jullie moet bij deze opgave een opdracht uitvoeren van de Website van Wim Dolman. Deze opdracht is te vinden op website van Wim Dolman -> Microcontrollers -> labs -> Micro-C -> Lab10. (<http://oege.ie.hva.nl/~dolwe/mic/labs/lab10.php>)

Attentie: De oscillator frequentie moet van tevoren aangegeven worden in het programma. Dit staat niet in de programmafragmenten op de website. Dat kun je als volgt doen:

```
#define F_CPU 1000000UL          // Oscillator frequency

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
```

Voer de opdrachten uit zoals beschreven op de website. **Je mag de onderdelen die betrekking hebben op de debugger overslaan.** Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 10: Practicum 7

Bij dit zevende practicum ga je een toerenteller maken met een 4 * 7-segmentsdisplay.

Na dit practicum kan de student:

- 4 * 7-segment displays aansturen;
- Externe interrupts verwerken in een interrupt service routine;
- Tijdsafhankelijke elementen maken d.m.v. timers.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;
- Uitbreidingsprint 02, 4 * 7-segmentsdisplay + 4 LED's;
- Variabele toongenerator met TTL uitgang.

Uitvoering:

De pulsen dienen op een interrupt ingang te worden ingelezen. Het toerental moet op het 4 * 7-segmentsdisplay worden weergegeven.

Daarnaast geldt dat er geen onnodige nullen getoond mogen worden. Al je dus bijvoorbeeld 57 moet tonen dan mag er geen 0057 staan. De intensiteit van de displays mag niet variëren.

Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 11: Individuele eindopdracht

Bij deze individuele opdracht moet je een eenvoudig kilometertellergedeelte van een auto maken. Hierin moeten onder andere aanwezig zijn:

- Een snelheidsmeter, snelheid in km/h;
- Een kilometerteller, totaal afgelegde weg in kilometers;
- Een dagteller, telt de kilometers met een nauwkeurigheid van 100 meter.
Tevens moet de dagteller gereset kunnen worden;

Daarnaast moet er een maximum snelheid ingesteld kunnen worden.

Let op het volgende: Er kunnen problemen optreden als je code opneemt in de while lus die in de main routine staat. Een oplossing is om deze code in een aparte methode te plaatsen en vanuit de while lus deze methode aan te roepen.

Na dit practicum kan de student:

- Alle onderdelen uit de voorgaande practica integreren in een groter geheel.

Benodigd:

- Moederboard;
- Uitbreidingsprint 01, 8 LED's, schakelaars, dipswitches etc;
- Uitbreidingsprint 02, 4 * 7-segmentsdisplay + 4 LED's;
- Toongenerator met TTL uitgang.

Uitvoering:

De uitvoer van de data zal moeten plaatsvinden op de 7-segment displays en de 4 LED's.

Op het bordje zitten vier drukknoppen, waarvan er drie worden gebruikt en welke de volgende functies hebben:

- Met 1 drukknop moet de displaymode gekozen kunnen worden;
- Met 1 drukknop moet er data gereset kunnen worden;
- Met 1 drukknop (vanaf nu up knop genoemd) moeten waarden verhoogd kunnen worden.

Door middel van de displaymode drukknop moet er een mode ingesteld kunnen worden en een LED moet aangeven welke mode is ingeschakeld zodat de gebruiker weet waar de getoonde data bij hoort.

De verschillende modi en hun eisen zijn:

- Mode 1:
LED nummer 1 brandt;
Snelheidsmeter, de actuele snelheid wordt getoond in km/h.;
Wordt de ingestelde maximum snelheid overschreden dan dient het display te gaan knipperen;

Drukt men op de displaymode knop dan moet er overgeschakeld worden naar mode 2;

Drukt men op de andere knoppen als de displaymode knop dan mogen die geen invloed hebben.

- Mode 2:

LED nummer 2 brandt;

Kilometerteller, de totaal afgelegde weg wordt getoond;

Wordt de ingestelde maximum snelheid overschreden dan dient het display te gaan knipperen;

Drukt men op de displaymode knop dan moet er overgeschakeld worden naar mode 3;

Drukt men op de andere knoppen als de displaymode knop dan mogen die geen invloed hebben.

- Mode 3:

LED nummer 3 brandt;

Dagteller, de afgelegde weg sinds men op de reset knop heeft gedrukt wordt getoond;

Wordt de ingestelde maximum snelheid overschreden dan dient het display te gaan knipperen;

Drukt men op de displaymode knop dan moet er overgeschakeld worden naar mode 4;

Drukt men op de reset knop dan moet de dagteller op 0 gezet worden;

Drukt men op de andere knoppen als de displaymode of reset knop dan mogen die geen invloed hebben.

- Mode 4:

LED nummer 4 brandt;

Instellen maximum snelheid;

Drukt men op de up knop dan moet de ingestelde maximum snelheid met 10 verhoogd worden. Men moet herhaaldelijk op de up toets kunnen drukken en wordt de waarde meer als 300 dan dient de ingestelde maximum snelheid terug te gaan naar 0;

Drukt men op de displaymode knop dan moet er overgeschakeld worden naar mode 1;

Drukt men op de reset knop dan moet de ingestelde maximum snelheid op 0 gezet worden;

Drukt men op de andere knoppen als de displaymode, reset knop of up knop dan mogen die geen invloed hebben.

De pulsen die de snelheid aangeven dienen op een interrupt ingang te worden ingelezen. De pulsen geven het aantal cm's per seconde aan. Voorbeeld: 60 km/h is 1000 meter (1 km) per minuut en dat is $1000/60 = 16,66666$ meter per seconde. En dit is weer 1666,6666cm per seconde. Staat de toongenerator dus op 1667 dan moet het display dus 60 aangeven. Voor een displaywaarde van 30 moet de toongenerator dus op 834 staan en voor een waarde van 100 moet de toongenerator dus op 2778 staan.

Daarnaast geldt dat er geen onnodige nullen getoond mogen worden. Al je dus bijvoorbeeld 57 moet tonen dan mag er geen 0057 staan. De intensiteit van de displays mag niet variëren.

Maak voor de opslag van het programma gebruik van de werkwijze zoals beschreven in de bijlage 2 van dit bouwsteenboek.

Bijlage 12: Scoring rubrics

De toets heeft een aantal gebieden waarop hij beoordeeld zal worden namelijk: Correctheid v.d. code, specificaties, efficiëntie, Code conventies, leesbaarheid en documentatie.

Daarnaast krijgt de student als bonus 10 punten zodat het maximum aantal te behalen punten voor de toets 100 is. Om het eindcijfer te berekenen dient dit puntenaantal door 10 gedeeld te worden.

De verdeling van bovengenoemde punten zal verdeeld worden over de opgaven. Het gewicht dat een opgave meetelt wordt aangegeven op de toets.

De bonuspunten zullen per opgave worden toegekend.

Eindbeoordeling toets van student:			met studentnummer:		
Onderdeel	Punten opgave 1	Punten opgave 2 (indien van toepassing)	Punten opgave 3 (indien van toepassing)	Punten opgave 4 (indien van toepassing)	Totaal aantal punten onderdeel
Totaal (totaal maximaal 90)					A
Bonuspunten (totaal 10)					B
Voor het maken van de eindbeoordeling zal gebruik worden gemaakt van de beoordelingsformulieren welke op de volgende pagina zijn te vinden.					Totaal A+B
					Delen door 10
					Eindcijfer

Beoordelingsformulieren van Student:**studentnummer:**

Beoordelingsformulier opgave:			
Onderdeel	Opmerkingen	Maximum aantal punten	Behaalde punten
Algemeen			
Bonus			

Beoordelingsformulier opgave:			
Onderdeel	Opmerkingen	Maximum aantal punten	Behaalde punten
Algemeen			
Bonus			