

Coffee on Demand

Van Java tot Java

MARINUS VAN HEUVELEN | RODERIK EMMERINK



Coffee on Demand

Van Java tot Java

Roderik Emmerink (230903),
Marinus van Heuvelen (230906)

Archytas
de Zaalsteden 18
9461 CT GIETEN

februari - juni, 2004

Voorwoord

Dit rapport over het afstudeerproject 'Coffee on Demand' hebben we geschreven in het kader van de afstudeerstage die deel uitmaakt van de vierjarige opleiding Elektrotechniek aan de Hanzehogeschool Groningen.

Wij zijn beide studenten aan de hiervoor genoemde opleiding en hebben een passie voor internetgestuurde elektronica. We hebben daarom met dit project de kans gegrepen het bedrijf Archytas, dat zich hiermee bezig gaat houden, op te richten en een project dat zowel bedrijfsmatig alsook beroepsmatig interessant is uit te voeren.

Dit rapport is bestemd voor Herman Noback, de begeleidend docent aan de Hanzehogeschool Groningen. Daarnaast is het rapport bedoeld voor Archytas om te functioneren als documentatie van het systeem. Het rapport is enigszins technisch van aard.

Waar mogelijk hebben we gekozen voor uit het Engels voortkomend jargon omdat dit doorgaans meer ingeburgerd is dan de overeenkomstige Nederlandse termen, bovendien is het meteen duidelijk dat het gaat om een technische term en niet om het equivalent in de praktijk (bv. icons op de desktop vs. pictogrammen op het bureaublad). Voor zowel inhoudelijke als tekstuele op- en aanmerkingen houden wij ons aanbevolen.

We willen graag Herman Noback bedanken voor zijn deskundige begeleiding en het meedenken tijdens het project.

Inhoudsopgave

Samenvatting	4
Verklarende woordenlijst	5
1 Inleiding	6
2 Het systeem	7
3 Het front-end	9
3.1 Web front-end	9
3.2 Stand-alone front-end	10
4 De intelligente koffiezetter	11
4.1 Temperatuurregeling	11
4.3 Robotaansturing	14
5 De robot	16
5.1 Ontvangen van opdrachten	16
5.2 Verwerken van de opdrachten	16
5.3 De aandrijving	17
5.3.1 Keuze van de motor	17
5.3.2 Stappenmotorcontroller	17
5.3.3 Overbrenging	19
Conclusie	20
Literatuurlijst	21
Bijlage I: projectplanning	22
Bijlage II: PCB lay-outs	23
Bijlage III: Schema architectuur Atmel microprocessor	24

Samenvatting

Dit rapport behandelt het afstudeerproject 'Coffee on Demand'. Tijdens dit project is een systeem ontwikkeld voor het afleveren van een kopje koffie nadat hiertoe opdracht is gegeven via een website.

Het systeem is op te delen in de volgende onderdelen:

- Een webfront-end.
- Een intelligente koffiezetter.
- Een robotkarretje.

Met het front-end heeft de gebruiker de mogelijkheid om het systeem te bedienen. Hij kan in een grafische interface de opdracht geven een kopje koffie te laten zetten en die op zijn locatie te laten afleveren. Het front-end is uitgevoerd in een webvariant voor in de browser en als een stand-alone applicatie. De koffiezetter bevat de mogelijkheid om een kopje koffie te zetten en is via internet te bedienen via een geïntegreerde embedded webserver. De koffiezetter kan op commando een klep open sturen voor het vullen van het interne waterreservoir. De temperatuur van het water wordt geregeld met een eenvoudig regelsysteem. De koffiezetter geeft het robotje via een UHF-zender instructies voor het afleveren van de koffie bij de gebruiker. Het robotje tot slot, kan een beker koffie ontvangen en deze naar een doellocatie, dat het door heeft gekregen van de koffiezetter via zijn geïntegreerde ontvangst-module, brengen.

Geconcludeerd wordt dat dit een leerzaam afstudeerproject is geweest. Verscheidene disciplines uit de opleiding elektrotechniek zijn toegepast.



Verklarende woordenlijst

client/ server	Communicatie tussen een client en een server.
front-end	In client/server applicaties wordt het clientgedeelte front-end genoemd.
back-end	In client/server applicaties wordt het servergedeelte back-end genoemd.
FTP	File Transfer Protocol, protocol voor het overzetten van bestanden.
HTTP	Hypertext Transfer Protocol, protocol voor het opvragen van webpagina's (en webbestanden)
RTOS	(Realtime Operating System), besturingssysteem ontworpen voor gebruik in een realtime computer systeem
Telnet	Telnet is een netwerk protocol. Het doel van het TELNET protocol is om een tweerichtings acht-bit byte-georiënteerd communicatiemogelijkheid aan te bieden. Het is gewoonlijk gebruikt voor een commando-lijn gestuurde login sessie tussen twee computers op een netwerk tot stand te brengen
IP	Internet Protocol
DHCP	Dynamic Host Configuration Protocol, protocol voor dynamische toekenning van IP adressen.
real-time	Een operatie in een groot dynamisch systeem wordt real-time genoemd als de gecombineerde reactie- en uitvoertijd van een taak korter is dan de maximale tijd die is toegestaan, rekening houdend met invloeden van buitenaf.



1 Inleiding

Wij lopen vele malen per dag naar de koffiezetautomaat om koffie te halen. Ook veel andere mensen doen dat regelmatig, getuige de lange rij voor het apparaat. Wij zijn van mening dat er, behalve een sociale reden, geen reden is om dit niet te automatiseren. Het idee voor een project was geboren, want een dergelijke toepassing past goed in onze opleiding en met name in onze afstudeerrichting informatietechniek.

We willen dus een systeem ontwikkelen waarmee men op de werkplek kan blijven en toch koffie kan krijgen. Een soort van koffieserveerster (m/v) dus. Hiertoe maken we een front-end voor de bediening via de browser, een aangepast koffiezetapparaat om deze opdrachten op te vangen en een karretje om dit naar de gebruiker te brengen. U bestelt koffie via een Java-applet en ontvangt uw Java-koffie van een robotje, kortom van Java tot Java.

Marien, één van onze docenten, kwam met het idee de koffiezetter op een oude computer aan te sluiten die met internet verbonden is. Wij vonden het professioneler en vooral universeler om een simpeler systeem in de koffiezetter zelf te verwerken en zijn uitgekomen bij een embedded webserver. Voor de koffiezetter zelf hebben we een oude kapotte koffiezetter gebruikt, die toevallig aanwezig was op school. Het front-end is gemaakt als programma in Delphi en als applet, zodat het in een website geplaatst kan worden, bijvoorbeeld het intranet. Het karretje hebben we zelf gebouwd en wordt voortgedreven met stappenmotortjes.

Binnen dit project zijn de volgende deelproblemen onderscheidbaar:

- Hoe kan de internetverbinding worden gelegd
- Hoe kan een boodschap doorgegeven worden
- De koffiemachine moet zo 'gereorganiseerd' worden dat hij de koffie kan zetten en op het robotwagentje kan

plaatsen

- Er moet een wagentje gemaakt worden dat naar een bepaalde locatie gestuurd moet kunnen worden
- De koffiemachine moet enkele commando's kunnen sturen naar het wagentje.

Kortom in dit project worden verscheidene disciplines uit onze opleiding toegepast. We hebben onder andere analoge techniek, digitale techniek, informatietechniek, PCB lay-out, telecommunicatie, regeltechniek, mechanica, aandrijftechniek, etc. toegepast.

“

Java

The Dutch planted the first arabica trees in Java early in coffee history, and before the rust disease virtually wiped out the industry, Java led the world in coffee production. Most of this early acreage has been replaced by disease-resistant robusta, but, under the sponsorship of the Indonesian government, arabica has made a modest comeback on several of the old estates originally established by the Dutch.

Java, like New Guinea, shares the low-toned richness of the other Indonesian and New Guinea coffees, but tends to be more obviously acidic, a bit lighter in body, and quicker to finish. Lurking in the acidity is a slight smoky or spicy twist.

Of the revived "old" estates that provide most of the good Java arabica, Djampit is the most likely source of the Java coffee in your specialty store.

”

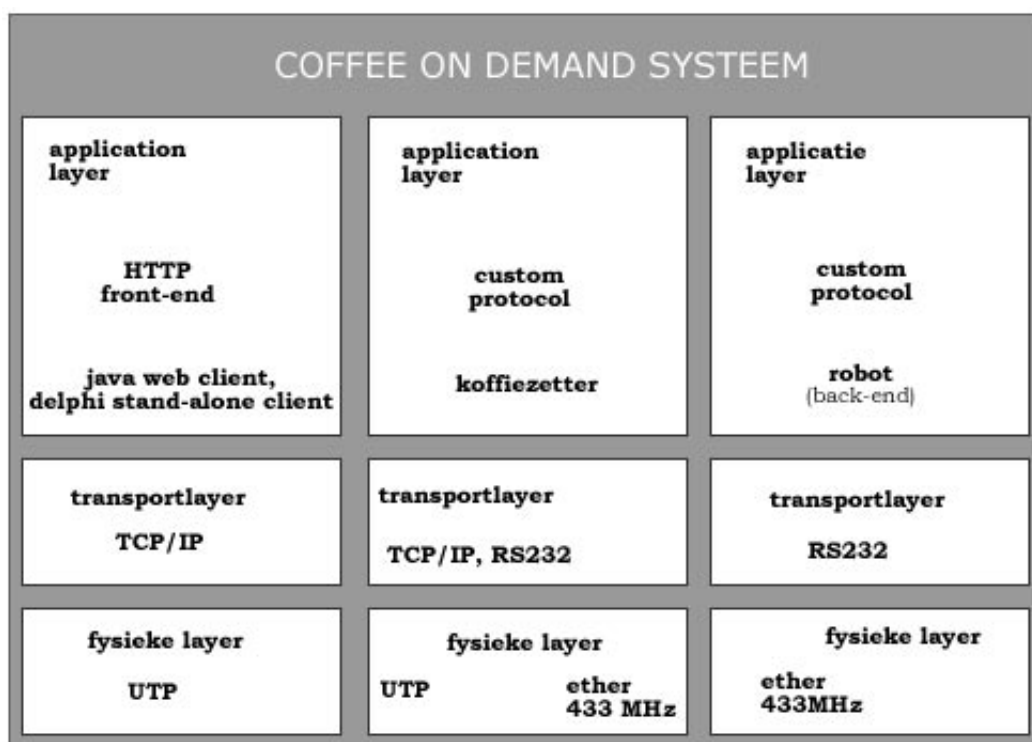
[10] Het waren Nederlanders die de eerste koffieboomen op Java plantten. Weer lopen zij op kop met (voor zover bekend) het eerste Coffee on Demand systeem.



2 Het systeem

In de komende hoofdstukken gaan we nader in op de onderdelen van het systeem. Om dit goed in de algehele context te plaatsen is een behandeling van het systeem als geheel gewenst. In dit hoofdstuk leggen we kort uit waaruit het systeem bestaat en hoe dit dient te functioneren.

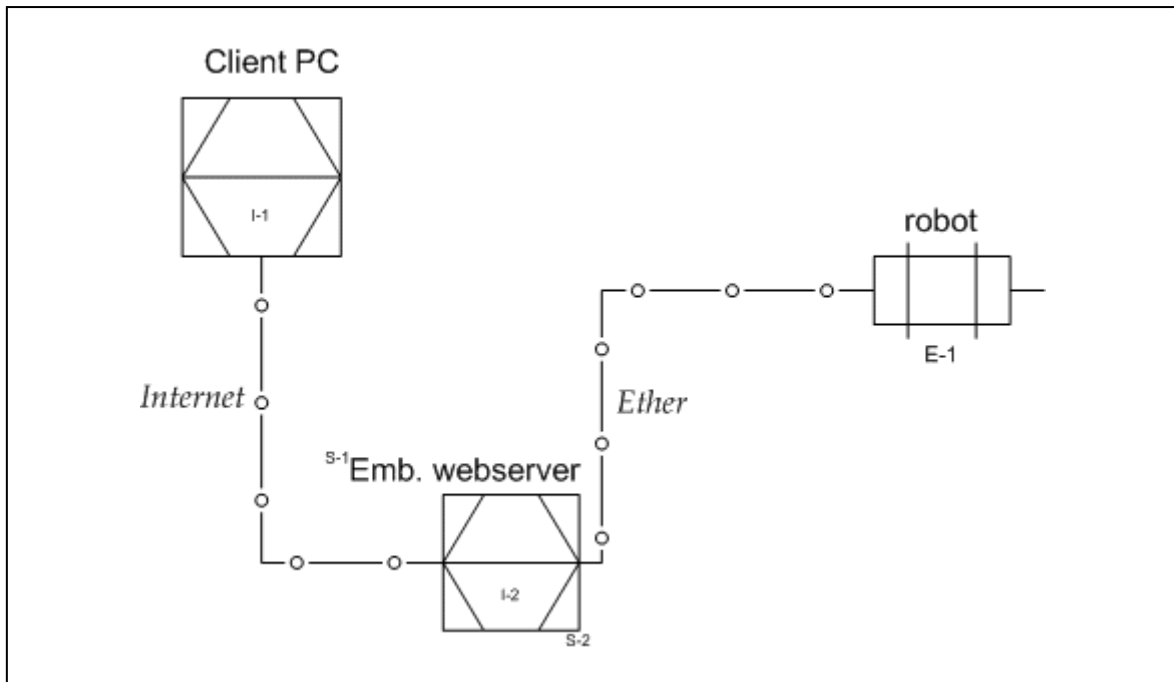
Als we het hebben over het systeem hebben we het over het geheel dat wij realiseren tijdens dit project. Om dit enigszins te verduidelijken hebben we een gelaagd schema gemaakt van het systeem.



Figuur 2.1: Gelaagde weergave van het 'Coffee on Demand' systeem.

Het systeem bestaat uit een front-end om een opdracht, bijvoorbeeld tot het zetten van een kopje koffie, op te geven, een koffiezetter om een kopje koffie klaar te maken en een robotje om het kopje koffie weg te brengen. Het front-end ziet u linksboven in figuur 2.1 en het communiceert via TCP/IP met de webserver in het koffiezetapparaat. Het koffiezetapparaat geeft door wanneer de koffie klaar is en waar het afgeleverd moet worden radiografisch via RS232 aan het robotje rechts in de figuur. Hieronder is een schets gemaakt van het systeem.





Figuur 2.2: Schema van de onderdelen van het systeem.

Samengevat komt het systeem op het volgende neer. Via internet wordt de opdracht tot het zetten en bezorgen naar de koffiezetter gestuurd. De koffiezetter stuurt het bezorgadres radiografisch naar het robotkarretje. Het robotkarretje bezorgt het kopje koffie bij de opdrachtgever.

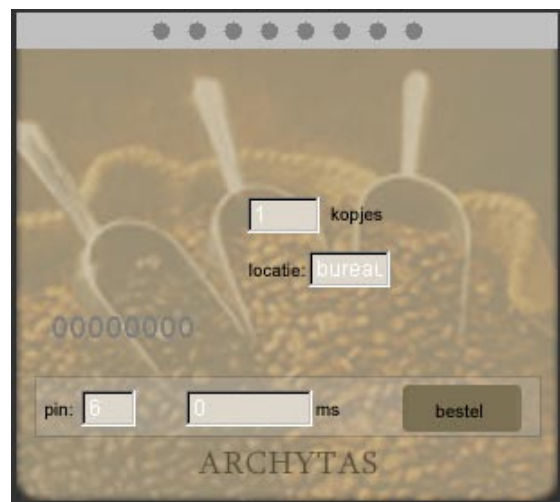


3 Het front-end

Heden ten dage is de trend om via internet systemen aan te sturen. *Archytas* ziet de mogelijkheden. Er staat bijvoorbeeld in the middle of nowhere een geautomatiseerde sluis. Nu zou de overheid kunnen beslissen er iedere dag een ambtenaar naar toe te sturen om naar gelang de hoeveelheid regen die er gevallen is het sluisje te openen c.q. te sluiten. Dit kan natuurlijk veel efficiënter. De ambtenaar kan aan zijn bureau blijven zitten en via een website of internetapplicatie het sluisje openen of sluiten. Dit is in enkele minuten geschied, zodat hij zich kan richten op andere werkzaamheden. Door de toepassing van zijn web front-end, de website, of een internetapplicatie is de toepassing locatie-onafhankelijk. Dit systeem wilden wij in het klein toepassen voor ons koffiesysteem, waar het vooral het gemak dient en door het unieke karakter wellicht een eyecatcher is.

3.1 Web front-end

Voor de bediening van het systeem via een website hebben we een web-frontend ontwikkeld. Voor dit webfront-end hadden we de keuze tussen ActiveX en Java, HTML en Flash. We hebben gekozen voor Java en HTML. ActiveX hebben we niet gekozen omdat dit niet platform-onafhankelijk is, het werkt niet in alle gangbare webbrowsers. Voor ActiveX zijn bovendien beveiligingrechten nodig en dat maakt het gebruik ervan lastiger. Flash hebben we niet gekozen omdat we hier niet veel ervaring mee hadden en we er dan te lang mee bezig zouden zijn om het in Flash te implementeren. Java wordt veel gebruikt, is op alle platformen beschikbaar en het wordt niet alleen voor applets gebruikt, maar ook voor stand-alone programma's en voor mobiele telefoons. Java is niet meteen het eenvoudigste omdat de objectgeïoriënteerde aanpak een vereiste is. We hebben reeds enkele programmaatjes ontwikkeld en wilden ons verder verdiepen in de mogelijkheden van Java. HTML is eenvoudig, maar is enigszins beperkt in de mogelijkheden, zo is het niet gedeeltelijk te refreshen. Een directe aansturing vanuit HTML naar de CGI services van de web-server is wel eenvoudig te realiseren. Het Java web-frontend is uiteindelijk bedoeld voor de eindgebruiker. De grafische gebruikersinterface ervan ziet u hieronder.



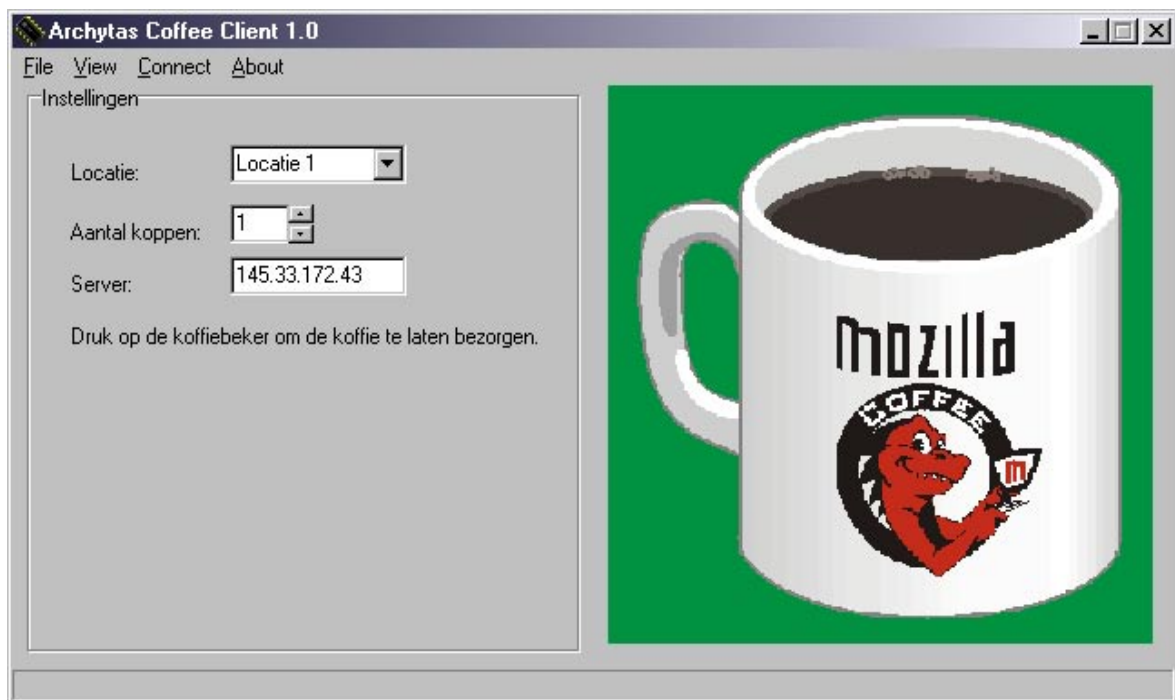
Figuur 3.1.1: Grafische interface van de Java applet

Met het web-frontend kun je een gewenste locatie opgeven en een hoeveelheid kopjes koffie kiezen. Bovendien kun je de status van het systeem bijhouden.



3.2 Stand-alone front-end

Ook is het mogelijk een stand-alone applicatie te ontwikkelen die valt te porten naar andere platformen. Bijvoorbeeld windows 2000, windows 98, windows 95, windows XP en WINE emulaties onder Linux. Het is sneller dan een Java-applet en is minder belastend voor de PC. Een voordeel van ons Delphi programma is dat je de kleuren in de grafische interface naar wens kan aanpassen. Ook kun je je persoonlijke instelling opslaan in het Windows register, iets wat met Java niet mogelijk is.



Figuur 3.2.1: Delphi grafische interface



4 De intelligente koffiezetter

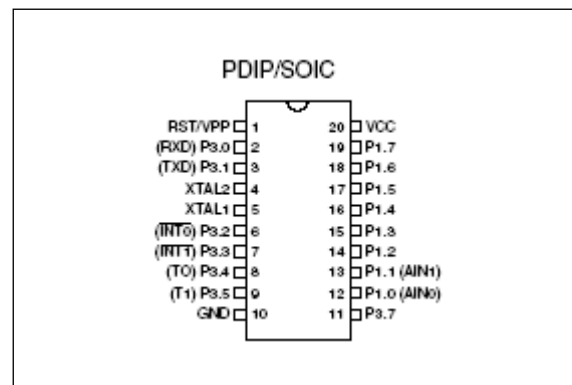
Het hart van het systeem is natuurlijk de koffiezetter. Daarvoor hebben we een Houd koffiezetapparaat bemachtigd dat op de faculteit aanwezig was. Deze koffiezetter bevatte geen elektronica meer. Toch was dit mooi meegenomen, want de hardware ervan hadden we nu vast. We hebben een schakeling toegevoegd voor het regelen van de watertemperatuur. Verder is er een embedded webserver ingebouwd om communicatie via het internet met de koffiezetter mogelijk te maken en een transceiver module is toegevoegd voor de communicatie met de bezorgrobot.

4.1 Temperatuurregeling

Aangezien onze koffiezetter geen elektronica meer bevatte was het noodzakelijk om zelf de elektronica te ontwikkelen. Een zeer belangrijk onderdeel voor het functioneren van de koffiezetter is het regelen van de watertemperatuur. In het waterreservoir dient de watertemperatuur constant bijna 100 graden Celsius te bedragen.

Binnen dit project passen wij enkele malen een Atmel microprocessor toe. We hebben gekozen voor een microprocessor van de firma Atmel omdat wij met deze microprocessors nog geen ervaring hebben opgedaan. Tijdens de opleiding is meestal gewerkt met PIC's van Microchip. Dit afstudeerproject was voor ons de mogelijkheid om onze kennis van microprocessors te verbreden.

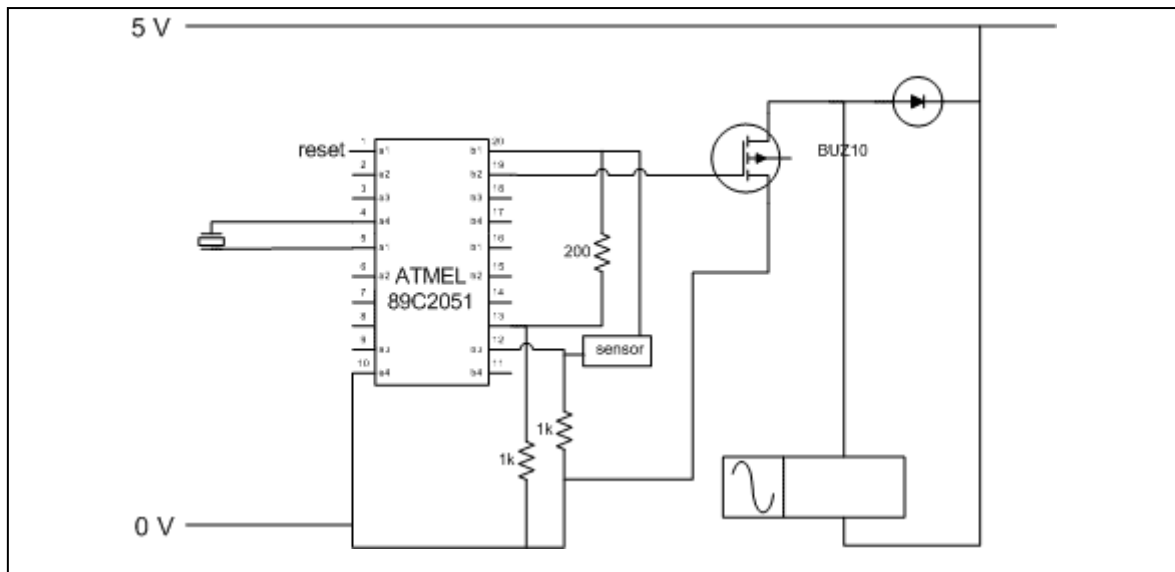
We hebben gekozen voor de 89C2051 chip van Atmel. Deze Atmel microprocessor is redelijk eenvoudig en is goed en goedkoop verkrijgbaar. In Bijlage III treft u een schema aan van de architectuur van deze Atmel microprocessor. Hieronder ziet u een afbeelding van de chip zelf en zijn aansluitingen.



Figuur 4.2.1: Atmel AT89C2051 microprocessor

De Atmel 89C2051 is een low-voltage CMOS 8-bit microcomputer. De chip bevat 2K bytes aan programmeerbaar en wisbaar (PEROM) flashgeheugen. De chip heeft ondanks de slechts 16 pin- nen maar liefst 15 I/O signalen. Verder bevat het twee 16-bit timer/counters, een 5 vector two-level interrupt archi- tectuur, een full duplex serial port, een analoge precisiecomparator en een on- chip oscillator en klok regeling.





Figuur 4.2.2: Temperatuur controller

Bovenaan de pagina ziet u het schema van de door ons ontwikkelde temperatuurcontroller. In de temperatuurcontroller is de Atmel microprocessor opgenomen. Er is gebruik gemaakt van de interne comparator. Een voordeel van de microprocessor is dat het eventueel uitbreidbaar is, bijvoorbeeld om in de toekomst de temperatuur door te geven om een indicatie via internet te realiseren. Voorlopig stuurt de microprocessor uitsluitend het verwarmingselement aan.

De werking van het schema is als volgt. Het kristal zorgt voor de referentiefrequentie waarop de microprocessor werkt. Er is gekozen voor een 11 MHz kristal. De verwerkingssnelheid is hier niet kritiek, maar vertragingen zijn afhankelijk van deze klokfrequentie. In de microprocessor is een procedure (een lusje) dat een halve seconde delay realiseert. De microprocessor vergelijkt de twee ingangen van de comparator (pin 12 en pin 13 in het schema) en stuurt pin 19 aan als de referentiespanning lager is dan de spanning afkomstig van de sensor. In het andere geval wordt pin 19 laag gemaakt. Na een halve seconde herhaalt dit proces zich. Op pin 19 is via een FET en een relais het verwarmingselement aangesloten.



4.2 Internetaansturing

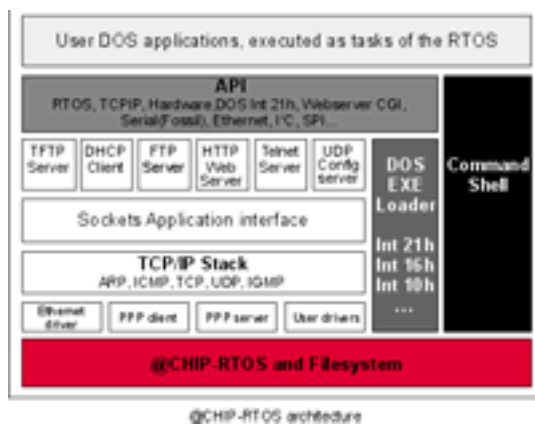
Voor het ontvangen van de instructies hebben we gekozen voor de SC12 single chip embedded webserver. Deze ziet u hieronder.



Figuur 4.1.1 De SC12 chip

De SC12 chip is geplaatst in een schakeling, de DK40, die de interfacing van de chip vereenvoudigt. Op deze embedded webserver draait een realtime operating system. Het realtime operating system zorgt ervoor dat de processen direct worden uitgevoerd zonder vertraging. Bovendien kunnen er meerdere processen gelijktijdig draaien. Iets wat bijvoorbeeld met MS DOS niet mogelijk was.

In de afbeelding hieronder ziet u hoe het realtime operating system is opgebouwd.



Figuur 4.1.2: RTOS architectuur

Het RTOS biedt de mogelijkheid DOS applicaties als taken (tasks) uit te voeren. Deze DOS applicaties kunnen instructies uitvoeren met betrekking tot het RTOS zelf, TCP/IP (voor internet), de hardware (bv. I/O), DosInt21h (starten van interrupts), Webserver CGI (afhandelen van internet requests), seriële communicatie.

Deze chip bevat de volledige TCP/IP stack en ondersteunt DHCP (Dynamic Host Configuration Protocol), dat houdt in dat de chip aan het WAN gekoppeld kan worden en dan een eigen IP-adres krijgt toegewezen.

De DK40 ondersteunt de Intel 80186 instructieset en ondersteunt multitasking tot 16 processen. Dat is een groot voordeel ten opzichte van bijvoorbeeld MS DOS dat geen multitasking ondersteund. De processen kunnen CGI calls ontvangen voor de aansturing ervan.

Ten behoeve van de ontvangst van instructies is er een programma gemaakt dat draait op het RTOS dat CGI requests kan afhandelen. Het programma, gebaseerd op een voorbeeldprogramma kan met een instructie van internet de status van een pin aansturen. Eén van de pinnen is aangesloten op een klep die op een waterreservoir is aangesloten. Met behulp van deze software is het mogelijk deze klep te openen of te sluiten.

We hebben een CGI applicatie geschreven welke de outputs van de DK40 kan sturen. Ook hebben we de externe seriële poort aangestuurd zodat communicatie met het robotkarretje mogelijk wordt. De programma's zijn in C geschreven en bevatten soms assembler.

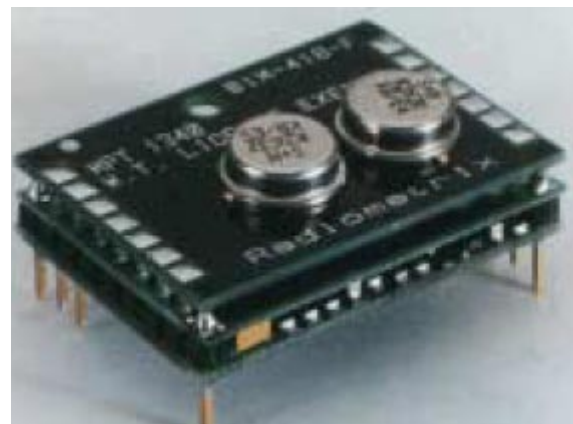


4.3 Robotaansturing

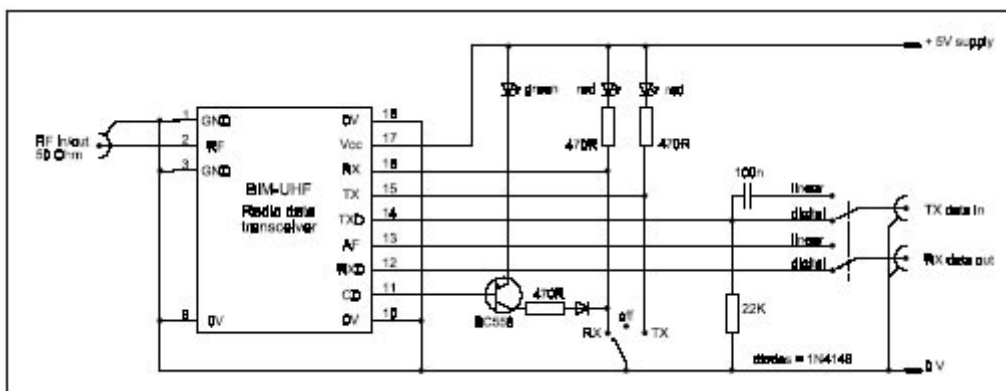
Als het koffiezetapparaat een kopje koffie zet is het van belang dat het robotje deze koffie opvangt en naar de cliënt brengt. Het koffiezetapparaat is door het webfront-end op de hoogte gesteld van de doellocatie en het aantal kopjes koffie dat de cliënt wenst. Er moest een methode gezocht worden om het robotje hiervan op de hoogte te stellen zodat het samen kan werken met het koffiezetsysteem.

In eerste instantie wilden we kiezen voor een zo eenvoudig mogelijke manier van communicatie. Al snel kwam het idee in ons op om door middel van een elektrisch contact tussen koffiezetter en robot in stationaire positie de gegevens over te zenden. Nadeel daarvan is dat de data alleen overgezonden kan worden als de robot tegen de koffiezetter aan staat. Een andere optie voor het overdragen van data was het aanbrengen van ledjes en met behulp van de knipperfrequentie de data door te zenden. Ongeveer zoals dat gebeurt bij een conventionele afstandsbediening. Een derde optie was om van radiocommunicatie gebruik te maken. Door gebruik te maken van radiocommunicatie hoeft het robotje niet tegen de koffiezetter aan te staan, ook als een object zich tussen koffiezetter en robot bevindt kan de communicatie gewoon tot stand worden gebracht. Zodoende hebben we voor het versturen van de opdrachten aan de robot een eenvoudige BiM-433-

40 UHF transceiver gebruikt. Deze transceiver kan half duplex data verzenden tot snelheden van 40 Kbit/s tot een afstand van 30 meter. Wij hebben ervoor gekozen de data op een snelheid van 9,6 Kbit/s te verzenden. Deze snelheid is ruim voldoende, betrouwbaar over te zenden, en is eenvoudig door onze Atmel microprocessor uit te sturen. Op onderstaande afbeelding ziet u de gebruikte transceiver module.



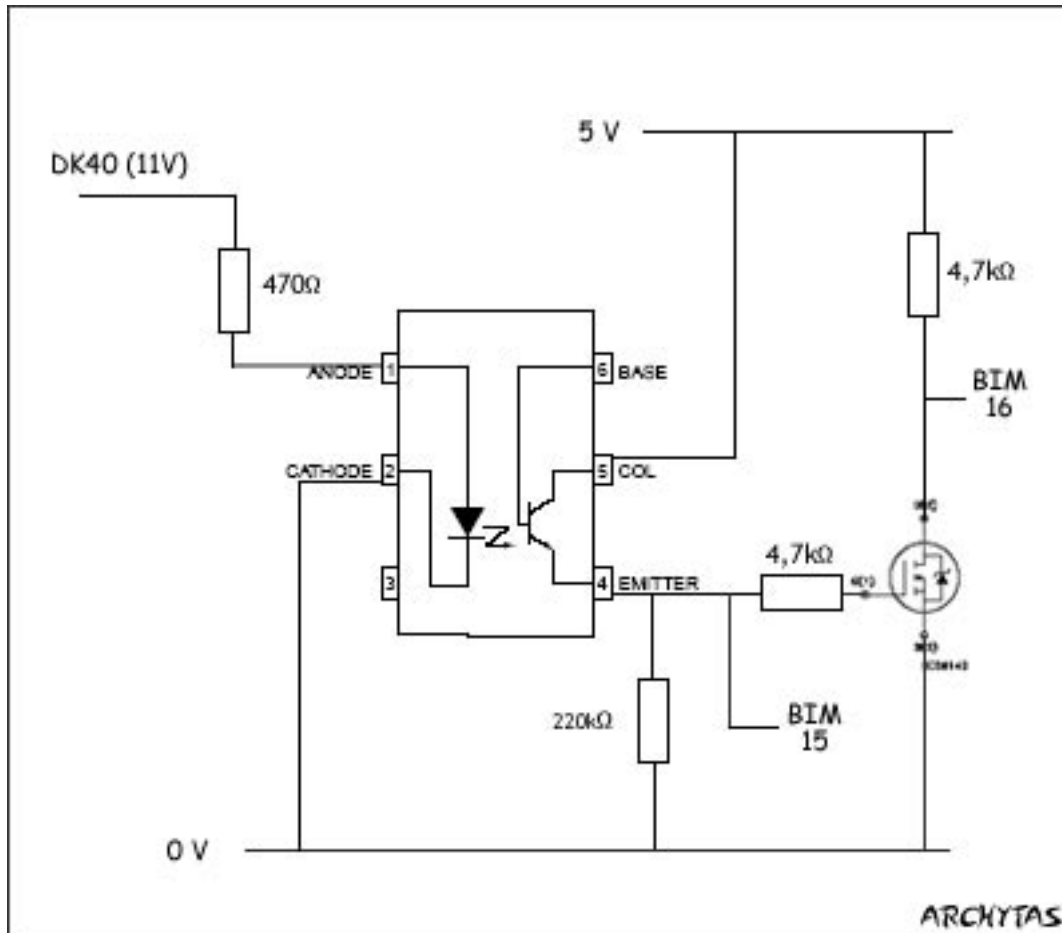
Figuur 4.3.1: BiM-433-40 transceiver module



Figuur 4.3.2: Transceiver circuit



Het transceiver circuit kan ingesteld worden op zend of ontvangst mode. We wilden het mogelijk maken dit met de webserver in te stellen. Een nadeel was dat de webserver een hogere spanning uitstuurt dan gewenst, namelijk ongeveer 11 V. We hebben een schema ontworpen en gemaakt om met een digitale output van de embedded webserver de transceiver module in te stellen. Dit schema ziet u hieronder.



Figuur 4.3.3: schema van de mode-selectorschakeling

Het schema werkt als volgt. Wanneer de DK40 een hoog signaal uitstuurt wil dit zeggen dat de transceiver dient te ontvangen. Met de optocoupler zijn de verschillende positieve spanningen galvanisch van elkaar gescheiden. Pin 16 is altijd de inverse ($\{0,1\}$ resp. $\{0\text{ V}, 5\text{ V}\}$) van pin 15 van de UHF transceiver. De emitter van de fototransistor stuurt een hoog signaal uit aan pin 15 van de transceiver. Ook op de gate van de FET staat een hoog signaal. Dientengevolge komt pin 16 (vrijwel) aan massa en vertegenwoordigt een laag signaal. Stuurt de DK40 een laag signaal uit dan komt op de emitter een laag signaal (deze wordt naar massa getrokken via een weerstand), op de gate staat ook dit lage signaal en pin 16 ligt dan aan 5 volt via een weerstand. De drain source spanning is in spertoestand verwaarloosbaar, dus pin 16 vertegenwoordigt een hoog signaal. De transceiver staat dan in zend mode.



5 De robot

Nu kan het systeem wellicht een kopje koffie maken als men hiertoe via internet opdracht geeft. We wilden echter wat extra's toevoegen. Koffie vanuit je luie stoel, ofwel Coffee on Demand moet natuurlijk gebracht worden. Hiertoe hebben we een robotje ontwikkeld dat koffie ophaalt bij de koffiezetter en brengt naar de gebruiker van het systeem.



Figuur 5.1: robotkarretje i.o.

5.1 Ontvangen van opdrachten

Het robotkarretje is nog niet functioneel op het moment van het schrijven van dit verslag. We beschrijven in dit hoofdstuk hoe het robotje in zijn volle functionaliteit dient te zijn.

De robot bevat een BIM UHF transceiver module waarmee het de opdrachten van de koffiezetter kan ontvangen. Met behulp van deze transceiver module en de transceiver module die in de koffiezetter aanwezig is (zie ook paragraaf 4.3) wordt een half duplex verbinding tot stand gebracht waarmee informatie tussen koffiezetter en robot uitgewisseld kan worden.

Het robotkarretje zendt aan de koffiezetter dat het op de stationaire locatie is (onder de koffiezetter). Op dat moment gaat de koffiezetter transceiver in zend mode. Als de koffiezetter de opdracht krijgt een kopje koffie te maken wacht het totdat deze gevuld is en stuurt dan de doellocatie naar het robotkarretje. Het robotkarretje ontvangt deze locatiecode. Deze locatiecode matcht met een routebeschrijving in het RAM geheugen (voorgeprogrammeerd) en vertrekt direct om het kopje koffie af te gaan leveren. Als het teruggekeerd is begint het verhaal weer van voor af aan.

5.2 Verwerken van de opdrachten

Zoals in de vorige paragraaf vermeld werd ontvangt het robotje een doellocatie op het moment dat het een kopje koffie moet afleveren. De Atmel microprocessor in de robot ontvangt deze informatie van de transceiver module op de robot. In het geheugen zijn de locaties opgeslagen als een instructielijst. Zo'n instructielijst bevat steeds opdrachten bestaande uit een te rijden richting en een tijdsduur dat het in die richting moet rijden.

In de software in de microcontroller moet steeds de tijd bijgehouden worden. Als de tijdsduur van een instructie verlopen is haalt het een nieuwe instructie op. De richting wordt vergeleken met de huidige richting. Het verschil wordt berekend. Voor een bepaalde tijdsduur wordt dan één van de wielen (afhankelijk of hij naar rechts of naar links moet afbuigen) op dubbele toeren gestuurd. Aangekomen op de eindlocatie wacht het 1 minuut en voert de robot de instructie in tegenstelde richting uit om weer terug te keren. In plaats van het linker wiel wordt het rechterwiel qua rotaties verdubbeld en vice versa om het niet in spiegelbeeld uit te voeren. De microprocessor stuurt de stappenmotor aan door pulsjes te zenden met een frequentie overeenkomstig het aantal stapjes dat het moet maken.



De hiervoor beschreven regeling is de basisversie. In deze uitvoering verwachten we namelijk dat het niet vlekkeloos zal functioneren. Het probleem kan bijvoorbeeld optreden dat ten gevolge van oneffenheden, een vuile vloer, punaise op de route het karretje enkele stapjes zal slippen en zodoende in een andere richting komt te staan. Niets vermoedend rijdt het karretje dan verder en botst uiteindelijk tegen een obstakel (bv. een muur) op, of erger, het rijdt zo het trapgat in. Zelfs wanneer het karretje exact op zijn route blijft is het mogelijk dat daar tijdelijk een object is geplaatst of dat er mensen voorlang lopen.

Om deze problemen te voorkomen zou het robotkarretje uitgerust moeten worden met een botsdetectie en zal het op één of andere wijze feedback moeten krijgen vanuit zijn omgeving om af te leiden dat het niet meer in de juiste richting rijdt, en beter nog, om te weten in welke richting het dan wel rijdt (en waar het verzeild is geraakt).

Over deze problemen is nagedacht, maar er is besloten dat dit buiten de scope van dit project ligt. In dergelijke oplossingen kan veel tijd gaan zitten. Bij voorkeur wilden we geen referentiepunten in de omgeving aanbrengen. In dat geval kan je gebruik maken van de referentiepunten die sowieso al aanwezig zijn. Bijvoorbeeld met radar of door gebruik te maken van GPS, maar dan wel verrijd, omdat GPS niet nauwkeurig genoeg is van zichzelf. Een andere mogelijkheid is het beperkt aanbrengen van referentiepunten. Door bijvoorbeeld twee ledjes te plaatsen achterelkaar aan het einde van een gang (of meer algemeen: een richting). Door steeds deze ledjes te detecteren en bij te sturen zodat deze op één lijn liggen kan de robot op zijn route blijven. In plaats van ledjes kunnen ook andere signaalbronnen toegepast worden. Het ontwijken van objecten/obstakels wordt echter te complex. Voorlopig blijft de robot wachten als het een object op zijn route

tegenkomt en hervat zijn route pas als dit verwijderd danwel vertrokken (persoon) is. Als dit lang duurt is een terugkoppeling gewenst, zodat in het front-end daar een melding van gegeven kan worden.

5.3 De aandrijving

5.3.1 Keuze van de motor

Voor de aandrijving van de robot kunnen verschillende motoren gebruikt worden. Een DC-motor en een stappenmotor liggen het meest voor de hand. Nadeel van een DC-motor is dat het toerental afhankelijk is van de belasting. Het toerental van een DC-motor ligt bovendien te hoog om deze direct aan de wielen te monteren, een vertraging met tandwielen is dan noodzakelijk. Een voordeel van de DC-motor is dat deze traploos geregeld kan worden en dat deze doorgaans sterker zijn dan stappenmotoren. We hebben echter gekozen voor een stappenmotor, omdat deze belastingsonafhankelijk is en omdat wij dachten dat wij deze direct op de wielen konden aansluiten. De belastingsonafhankelijkheid is noodzakelijk voor onze oplossing, omdat wij deze dan zonder terugkoppeling kunnen toepassen. In ons basismodel is namelijk geen sprake van terugkoppeling. De door ons toegepaste stappenmotor betreft een 4 polige 7.5 graad nauwkeurige motor.

5.3.2 Stappenmotorcontroller

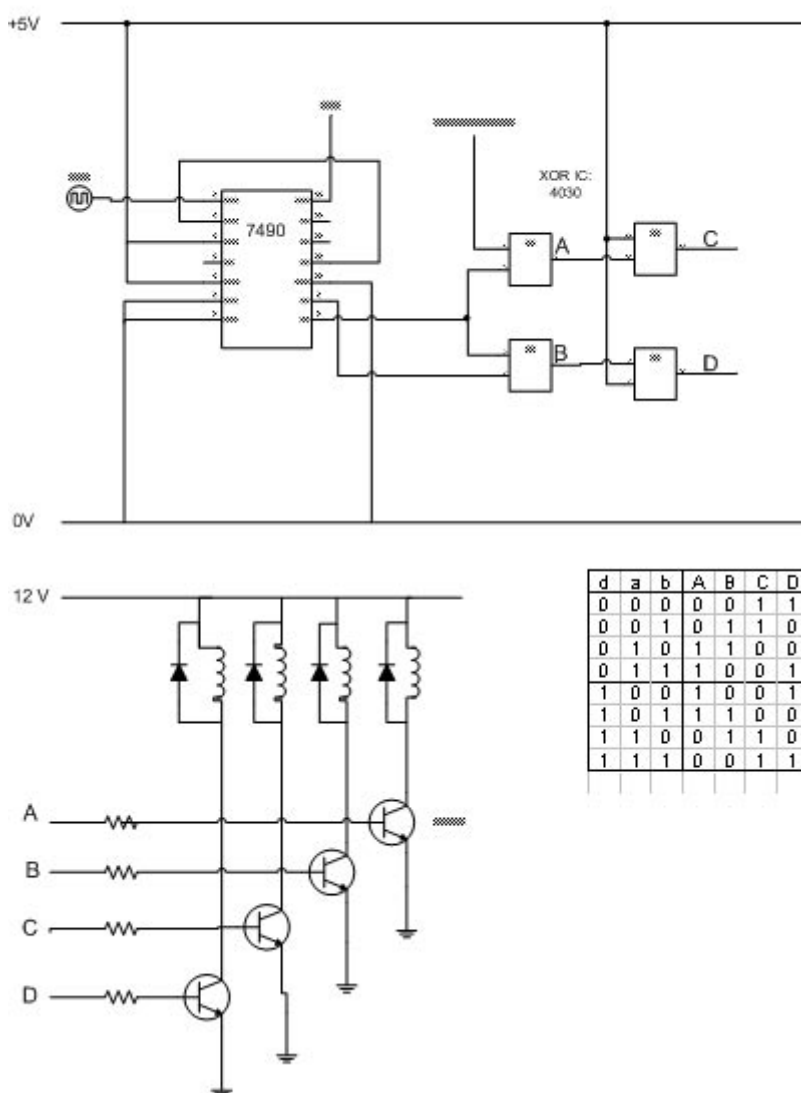
Er bestaan verschillende methodes om een stappenmotor aan te sturen, immers een stappenmotor is niet direct aan te sturen omdat deze een 'waveform' nodig heeft om te gaan draaien. De methodes zijn afhankelijk van het type motor. Bij onze motor waren er twee mogelijkheden, namelijk een single phase en double phase aansturing. Op de markt zijn diverse universele controllers te koop, echter wij wilden liever zelf een controller maken om zo het principe te doorgronden en ervaringen



op te doen omtrent stappenmotoren. Voor ons was dit een extra uitdaging.

Voor de aansturing van de stappenmotoren hebben we een schakeling ontworpen. Deze schakeling ziet u hieronder. Het schema is in twee delen opgesplitst. Het bovenste deel is verantwoordelijk voor het omzetten van de pulsjes van de microcontroller naar de waveform. Het onderste gedeelte is voor het aansturen met deze waveform van de spoelen in de stappenmotor.

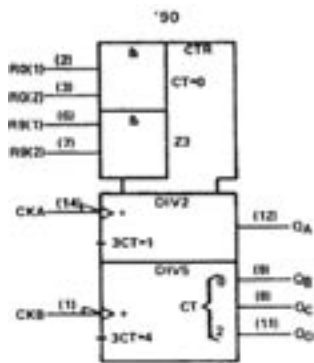
STAPPENMOTORBESTURING



Figuur 5.3.2.1: Schema stappenmotorcontroller

Het onderste deel van het schema spreekt voor zich. Het bovenste deel werkt als volgt: Het symbool van een pulsgenerator staat voor het pulssignaal afkomstig van de microcontroller. De pulsgenerator functioneert als kloksignaal voor de 7490. De 7490 is een decade (10) teller. Een schema van de teller ziet u hieronder.





Figuur 5.3.2.2: 7490 decade teller

We hebben maar één klokingang gebruikt. Daarom is geen gebruik gemaakt van de eerste uitgang en is de telsnelheid gehalveerd. Dit geeft echter niet, dit wordt immers softwarematig geregeld. Als de teller aankomt bij 4, reset het, het betreft hier dus eigenlijk een 4-teller. De telcyclus wordt met behulp van de poortjes efficiënt omgezet naar de gewenste waveform, zoals vermeld in de waarheidstabel. De achterste poortjes zorgen ervoor dat bij een hoog signaal op d (direction) de richting van de cyclus omdraait, zodat de motortjes in tegengestelde richting roteren.

Nadat we deze schakeling op een experimenteerbordje hadden getest hebben we hier een PCB voor ontworpen. Hieronder ziet u de resulterende print waarvan we voor iedere motor één hebben gemaakt.



Figuur 5.3.2.3: resulterende PCB voor de aansturing van de stappenmotoren

5.3.3 Overbrenging

De overbrenging betreft niet ons vakgebied en daarom gaan we er hier slechts beknopt op in. Er is gekozen voor een overbrenging met tandwielen. We zouden natuurlijk kunnen kiezen om direct het wiel te bevestigen aan de stappenmotor. We wilden een vertraging inbouwen, omdat dit resulteert in een groter koppel. Het robotje beweegt zich dan minder snel voort, maar is nog steeds binnen een redelijk periode op zijn bestemming, zodat de koffie nog warm is als het bij de cliënt arriveert.

Op de stappenmotor is een klein tandwiel bevestigd. Dit tandwiel raakt het tweede tandwiel dat op het wiel is bevestigd. Dit tandwiel is zo groot mogelijk gekozen, maar kleiner dan het wiel, zodat het de grond niet raakt en het koppel toch maximaal is. Voor het tandwiel aan de motor hebben we het kleinst mogelijk beschikbare geschikte tandwiel gekozen. Op deze manier hebben we een overbrenging gerealiseerd met een acceptabele vertraging en een koppel dat zo hoog mogelijk is.



Conclusie

Dit rapport beschreef het afstudeerproject 'Coffee on Demand'. Tijdens dit project is een systeem ontwikkeld voor het afleveren van een kopje koffie nadat hiertoe opdracht is gegeven via een website.

Het systeem is op te delen in de volgende onderdelen:

- Een webfront-end.
- Een intelligente koffiezetter.
- Een robotkarretje.

Alle onderdelen zijn tijdens dit project ontwikkeld, alleen het robotkarretje is niet geheel functioneel. Om de onderdelen van het systeem te ontwikkelen en deze aan elkaar te koppelen zijn een groot aantal disciplines toegepast die aan bod zijn geweest bij de opleiding elektrotechniek. Onze ervaring is op een aantal punten verbreed (bv. mechanica, embedded webserver) en verdiept (bv. microcontrollers).

Al met al betrof het een leerzame en interessante afstudeerperiode.



Literatuurlijst

De literatuurlijst geeft een indruk van gebruikte literatuur. In de tekst wordt alleen verwezen naar literatuur wanneer er sprake is van citaten.

- [1] Laszlo, M. J., Object-Oriented Programming featuring Graphical Applications in Java. United States of America, 2001.
- [2] Liang, Y. D., Introduction to Java Programming, 2003
- [3] Robillard, M.J., Microprocessor based Robotics, 1985
- [4] Witowski, M., Boyle, T., Handboek Adobe InDesign
- [5] Ince D., Developing Distributed and E-commerce Applications, Harlow, 2002

Internet

- [6] Beck IPC (embedded webserver)
<http://www.beck-ipc.com/>
- [7] Circuits Online (datasheets, ondersteuning via forum)
<http://www.circuitsonline.net/>
- [8] Archytas afstudeerwebsite (pr, rapportering)
<http://afstuderen.archytas.nl/>
- [9] Too Much Coffee Man (cover art)
<http://www.tmcm.com/>
- [10] Lucidcafe (informatie over java-koffie)
<http://www.lucidcafe.com/>
- [11] Atmel Corporation (documentatie microcontroller)
<http://www.atmel.com/>



Bijlage I: projectplanning

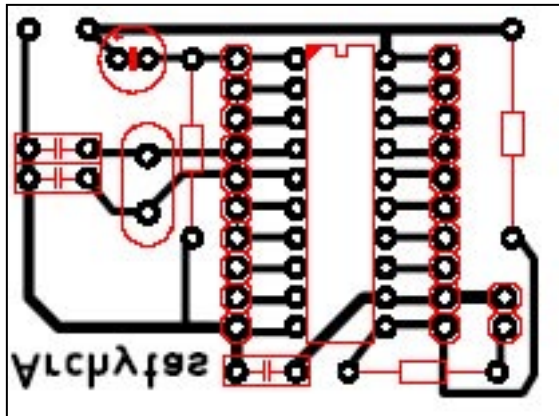
Gedurende het project hebben we wekelijks op het whiteboard opgeschreven wat de doelstellingen zijn voor de komende week/weken. Een globale planning voor de hele periode is ook gemaakt. We hebben dit echter niet gedocumenteerd. Het werkte efficiënt door dit op het whiteboard te doen. De planning is dan ten alle tijden zichtbaar en aan te passen aan de ontwikkelingen.

Bij aanvang van het project hebben we dit als volgt globaal gefaseerd:

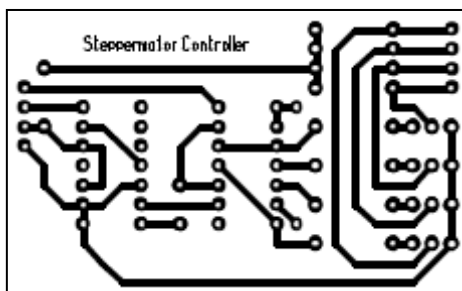
- Functionele specificatie opstellen
- Technische specificatie opstellen
- Onderdelen + kosten opsommen
- Aanschaf componenten (maart, week 1)
- A: ontwikkeling basis robot (maart, week 3)
- B: ontwikkeling koffiezetter + zend/ontvangstunit (april, week 1)
- C: ontwikkeling front-ends Java, Delphi (april, week 3)
- Verbetering A, B, C
- Koppeling alle componenten (mei, week 1)
- Testfase (mei, week 3)
- Acceptatiefase (juni, week 2)
- Verslaggeving is continue, maar is in mei, week 4 afgerond



Bijlage II: PCB lay-outs



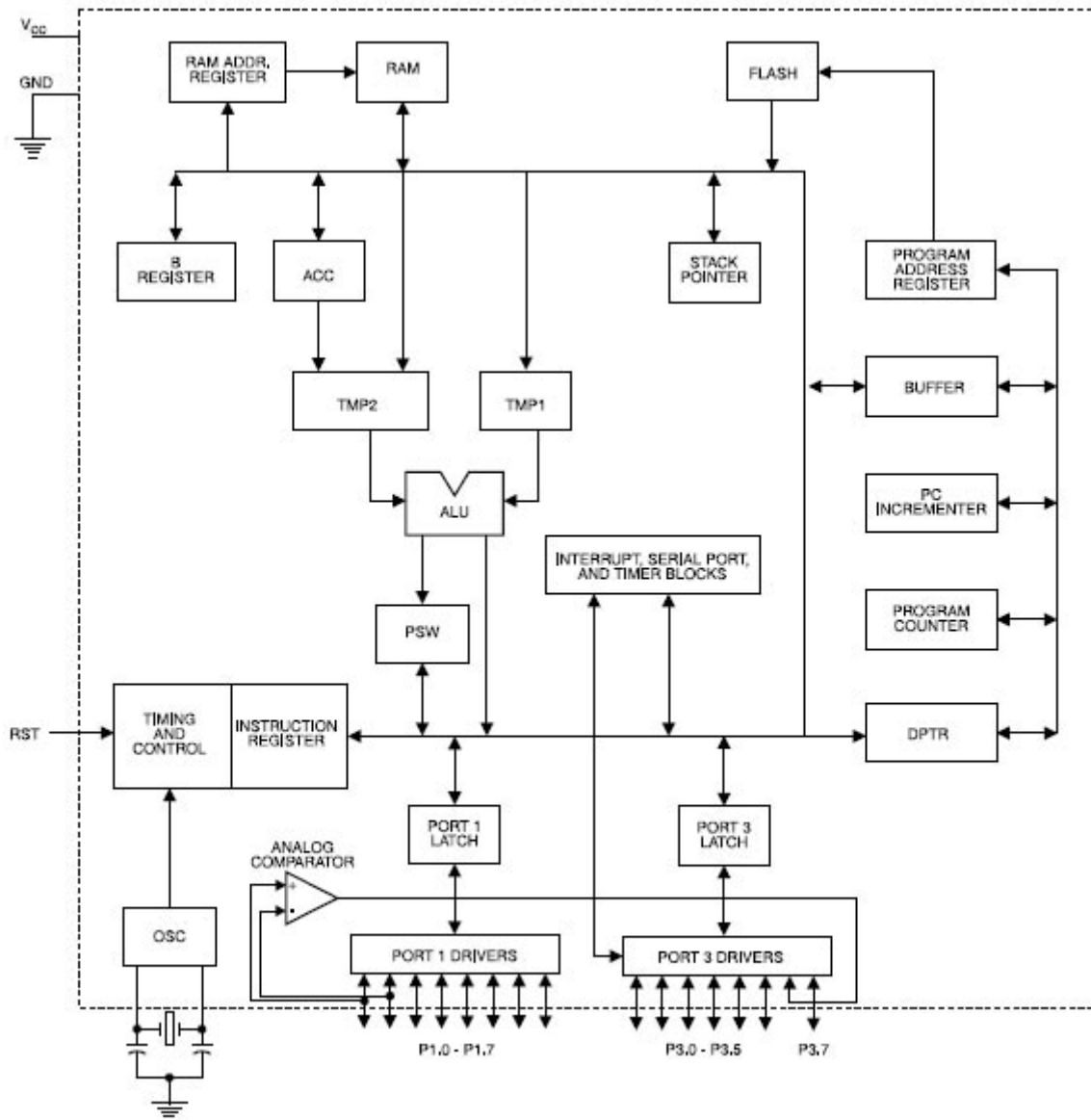
Figuur II.1: PCB voor de Atmel microprocessor.



Figuur II.2: PCB voor de stappenmotor controller



Bijlage III: Schema architectuur Atmel microprocessor



Figuur III.1: Architectuur AT89C2051

