Werkboek Arduino Duemilanove

A. Kompanje

V1.11 oktober 2009

Werkboek Arduino Duemilanove

De informatie is onder andere verkregen door:

http://www.arduino.nu
http://nl.wikipedia.org

Meer informatie kun je vinden op:

http://www.arduino.cc
http://www.arduino.nu
http://www.wiring.org.co
http://www.antratek.nl
http://nl.wikipedia.org

Bronvermelding:

Bijlage 1: http://nl.wikipedia.org/wiki/Atmel_AVR

Uitgegeven: oktober 2009

Deze tutorial valtonder de licentie: Creative Commons Attribution-Share Alike 2.5.
Een kopie van deze licentie staat op:
http://creativecommons.org/licenses/by-sa/2.5/

Inhoudsopgave

		blz
1.0	Inleiding	4
2.0	Atmega 168/328	5
3.0	De Arduino Duemilanove	7
3.1	Alternatieven	8
3.1.1	De Arduino Mega	9
4.0	Arduino Interface	10
4.1	De Menubar	11
4.1.1	File	11
4.1.2	Edit	11
4.1.3	Sketch	12
4.1.4	Tools	12
4.1.4.1	Board	13
4.1.4.2	De seriële poort	13
4.1.4.3	Installeren USB-Seriële poort Windows	14
4.1.4.4	Installeren USB-Seriële poort MAC	16
4.1.5	Help	19
4.1.6	De Toolbar	20
5.0	Aansluiten en programmeren	21
5.1	Uitbreidingsbord	22
5.2	Assemblage uitbreidingsbord	23
5.3	Schema uitbreidingsbord	26
6.0	Vragen en opdrachten	27
Opdracht 1	Verkeerslicht zonder loop	29
Opdracht 2	Verkeerslicht For-Next	31
Opdracht 3	Verkeerslicht Do-While	33
Opdracht 4	LED schakelen	35
Opdracht 5	Looplicht	37
Opdracht 6	LED snel/langzaam knipperen	39
Opdracht 7	LED regelen 1	41
Opdracht 8	LED regelen 2	43
Opdracht 9	Codeslot	45
Opdracht 10	Serial monitor	47
Opdracht 11	Digitale teller 1	49
Opdracht 12	Digitale teller 2	51
Opdracht 13	Servomotor	53
Opdracht 14	Servomotor twee richtingen	54
Opdracht 15	Reactiespel 1	56
Opdracht 16	Reactiespel 2	58
Opdracht 17	Lijnvolger	60
Bijlage 1	Atmel AVR	63

1.0 Inleiding

De Arduino wordt geprogrammeerd door een open source programma. De gebruikte taal is C. In dit werkboek leer je de mogelijkheden kennen en de basisbeginselen.

Er zijn veel chipfabrikanten die een microcontroller in het assortiment hebben, waaronder Atmel (AVR), Intel (MCS51), Microchip (PIC) en Motorola/Freescale (6800). Hiernaast zijn er haast ontelbaar veel bedrijven die met een specifieke microcontroller of familie van microcontrollers een experimenteertoepassing aanbieden. De meest bekende is Parallax met de Basic Stamp.

De meeste van deze oplossingen zijn echter vrij prijzig (> € 100,=). Een nieuwe telg is hier het Arduino project, dat gebruik maakt van een Atmel ATmega168/328 microcontroller. Bijzonder van Arduino is de ondersteuning van meerdere ontwikkelplatforms (Windows, OS X, Linux), de mogelijkheid om de basisprint zowel voor standalone- als voor programmeertoepassingen te gebruiken, de voeding via USB of via een batterij en de uitgebreide user-base. En de prijs is laag: een compleet ontwikkelsysteem voor minder dan € 30,=.

Dit werkboek is geschreven voor de Arduino Duemilanove. Het is echter 1 op 1 toepasbaar op:

Arduino Diecimila
Arduino Pro
Arduino Mega

LET OP: Er zijn Arduino boards die werken met een maximale spanning van 3,3 V. Door de gebruikte componenten die in dit werkboek zijn beschreven zijn alleen 5 V Arduino boards bruikbaar.

Arduino netwerkmodule

2.0 De Atmega 168/328

De Arduino maakt gebruik van een AVR-microcontroller van de firma Atmel. Het is een 8-bis microcontroller me 16 kB geheugen (Atmel 168) of 32 kB geheugen (Atmel328). Je zie al aan de benaming dat er logica in zit. Atmel328 → 32 kb→ 8 bit → 328.

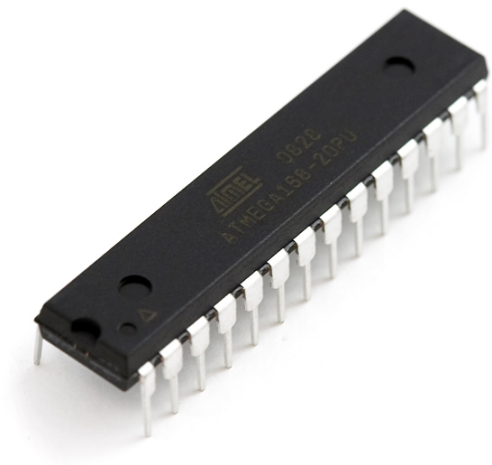


Fig 1, Atmega168

Voor het programmeren van zo'n IC moet er een programma geschreven zijn waarin je in een programmercode kwijt kunt. Dit wordt ook wel een IDE-interface genoemd. IDE staat voor **Integrated Development Environment.** Atmel gebruikt daar voor een programmeeromgeving genaamd: AVR Studio® 4. Deze versie is geschikt voor Windows vanaf versie 2000. Een nadeel van dit programma is dat deze alleen bruikbaar is in assemblertaal. Wil je namelijk snel aan de slag dan is een hogere programmeertaal wenselijk.

Om dit euvel op te heffen heeft een aantal mensen besloten om een programma te schrijven in de open sourcecommunitie waar geen beperkingen aan vast zitten. Deze interface heet gewoon de Arduino IDE.

Atmega-processoren zij erg populair in gebruik vanwege de vele mogelijkheden die zij bezitten. Zo zitten er 14 programmeerbare in- en uitgangen op en analoge uitgangen. Ook kan de processor communiceren via het seriële i^2C protocol.

De processor werkt met een kloksnelheid van 8 MHz.

Vaak moet bij het programmeren van een chip deze eerst uit de schakeling worden gehaald en in een programmeerapparaat worden geplaatst, waarna het programmeren kan starten. Dat is bij de Arduino echter niet het geval.

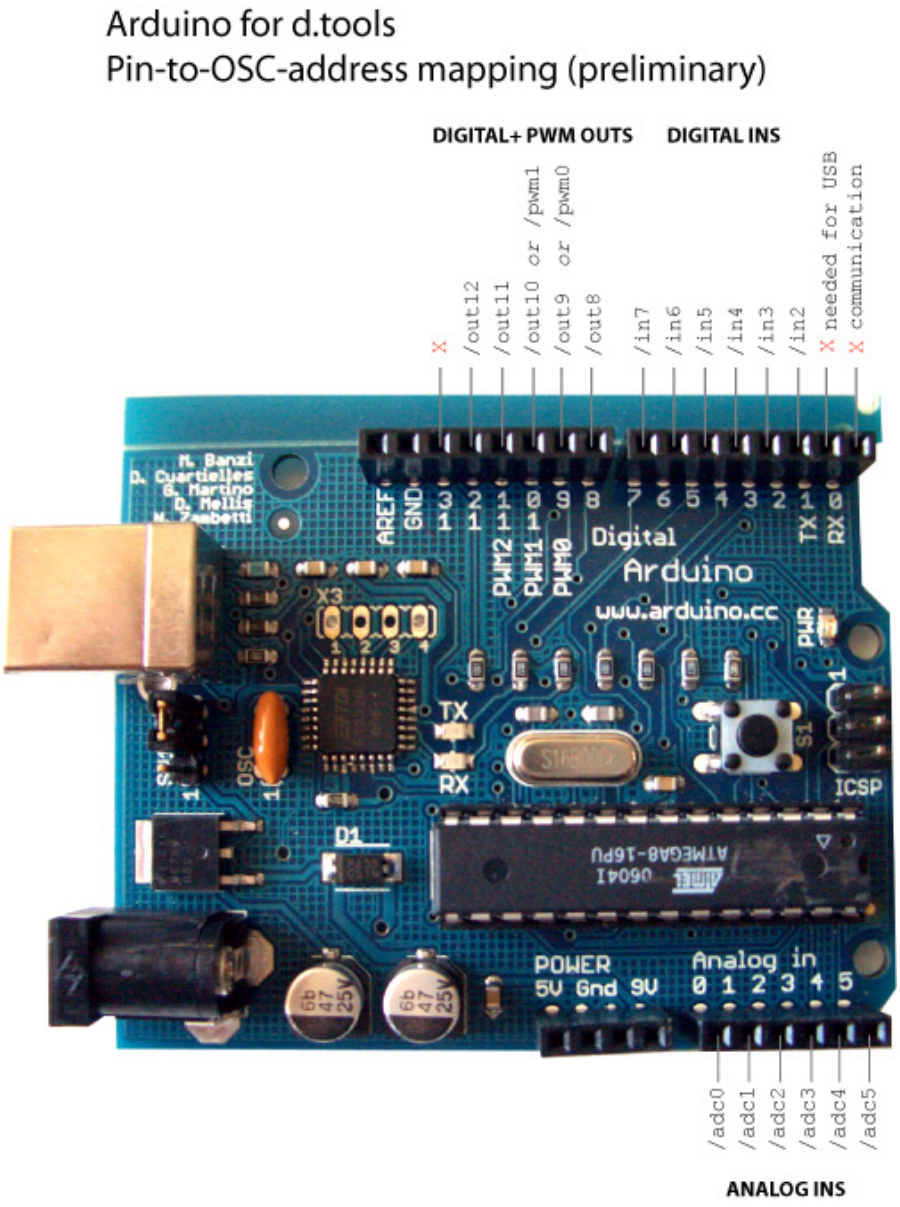
De chip kan in de schakeling blijven zitten. Dit wordt ook wel ISP genoemd (In System Program). ISP heeft wel één nadeel: In de Atmega-chip moet een zogenaamde bootloader zitten. In zo'n bootloader staat geschreven hoe de software waar geplaatst moet worden. Er kan dus nooit een "lege chip"in. Het programmeren van de bootloader in de chip moet dus eenmalig met een programmeerapparaat of de chips moeten kant en klaar, voorzien van bootloader gekocht worden.

Fig 2, Programmeerapparaat

3.0 De Arduino Duemilanove

Arduino Duemilanove. Dit is de laatste versie van het populairste ontwikkelbord en is de opvolger van de Diecimila. Verschil met de Diecimila is dat de Duemilanove automatisch omschakelt tussen USB voeding en een losse adapter. De Duemilanove is de meest eenvoudig te gebruiken versie van Arduino doorat deze direct op de computer kan worden aangesloten en makkelijk te gebruiken aansluitcontacten heeft. Er is een mogelijkheid om zogenaamde shields op het Arduino bord te plaatsen waardoor de mogelijkheden worden uitgebreid. Voorbeelden zijn shields waarmee communicatie mogelijk is tussen 2 Arduino's, een shield voor het besturen van servomotoren, een shield voor het aansturen van RGB LEDs of een ethernet adapter om de Arduino aan te sluiten op het internet. Het is mogelijk om meerdere shields op elkaar te zetten, waardoor een soort flat ontstaat. De geprogrammeerde code wordt geüpload via USB-B.

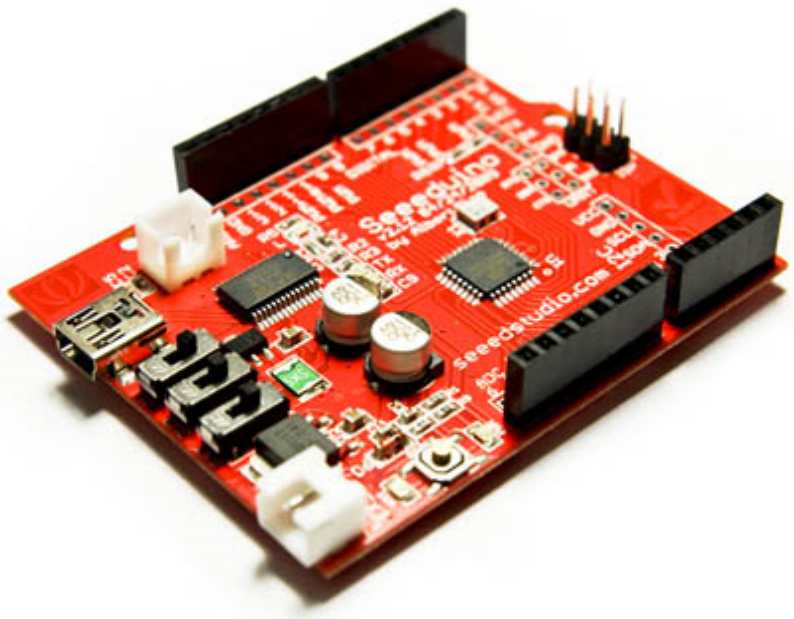
De Atmega chip is meestal geplaatst in een DIP socket, en is makkelijk te vervangen, bijvoorbeeld als de chip overbelast is geraakt door aangesloten randapparatuur.



3.1 Alternatieven

Omdat het Arduinoprojectopensource is zijn er op internet heel veel (legale) varianten te vinden met vaak kleine aanpassingen. Naast deze volledig werkend geleverde versies zijn er ook Arduino bouwpakketten te koop waarbij zelf alle componenten op een printplaat gesoldeerd moeten worden. Voorbeelden zijn de BoArduino van Adafruit of de Bare BonesArduino Board Kit van Makezine. Aangezien dit Amerikaanse versies zijn en Arduino een Europees gemaakt product is, zal voor de Europese gebruiker het zelf bouwen weinig financieel voordeel geven.

Voor meer informatie kijk ook eens op: http://arduino.nu/les6/les6.php

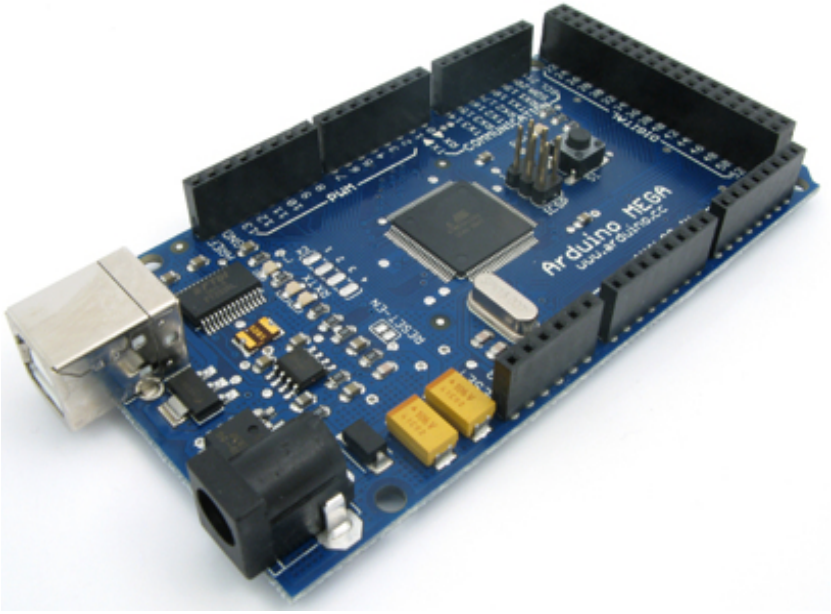


Op de bovenstaande foto zie je een Arduino variant de zogenaamde Seeeduino. Let goed op dat de Atmega chip rechtstreeks op de print gesoldeerd is, dat is goedkoper, maar als de processor stuk is, is deze niet meer te vervangen.

Verder zijn er nog tientallen andere varianten, die variëren in grootte, low budget of gemaakt voor een specifiek doel. Een voorbeeld van dat laatste vind je op: *http://diydrones.com/profiles/blogs/ardupilot-main-page*

3.1.1 De Arduino Mega

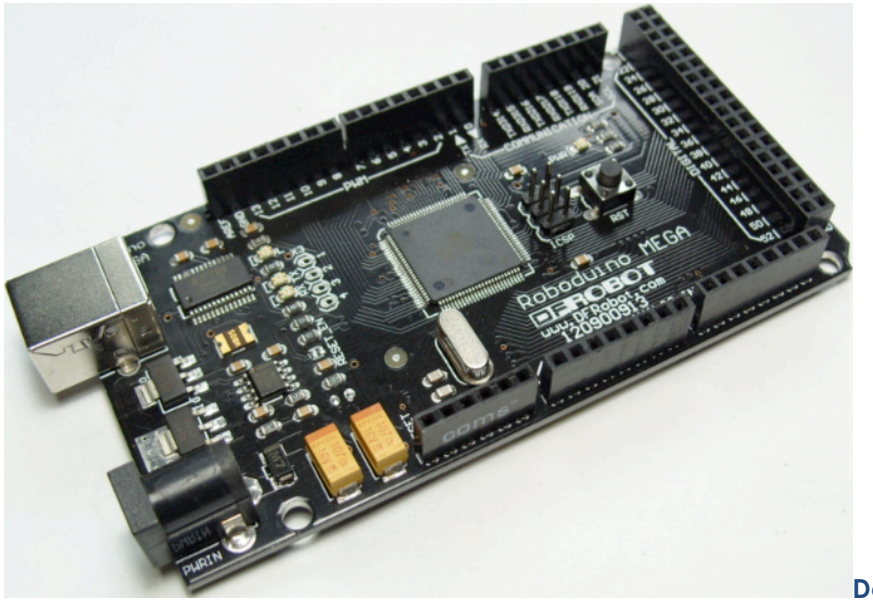
Een ander model uit de Arduino serie is de Arduino Mega. Deze controller is krachtiger dan de Arduino's die hier voor beschreven zijn.



De Arduino Mega

De Arduino Mega is gebaseerd op de ATmega1280. Hij heeft 54 digitale in- en uitgangen, daarvan kunnen er 14 gebruikt worden als PWM uitgangen. Er zitten 16 analoge inputs op en 4 UARTs (Hardwarematige seriële poorten). Het geheel wordt aangedreven door een 16 MHz kristal en is voorzien van een USB-aansluiting. Ondanks het grotere formaat is de Mega volledig compatibel met de ArduinoDuemilanove.

Ook van de Mega is alweer een identiek alternatief te krijgen bij www.dfrobot.com.



De eerste Mega kloon

4.0 Arduino IDE interface

De Arduino IDE interface ziet er als volg uit:

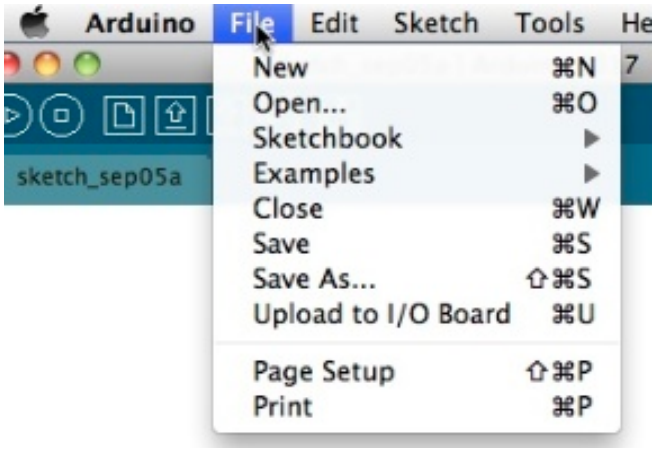
Elk omcirkeld gebied in de foto hier boven heeft een bepaalde functie. Deze zijn:

- **Menu:** Selecteer een commando dat je wilt laten uitvoeren.
- **Toolbar:** Knoppenbalk met de belangrijkste knoppen.
- **Tabs:** Elk programma dat je maakt krijgt zijn eigen tab. Je kunt dus met meerdere applicaties tegelijk werken.
- **Texteditor:** In deze ruimte zet je jouw programmeercode.
- **Berichten gebied:** Hier komen berichten te staan van genomen acties (bijvoorbeeld: Done compiling).
- **Tekst gebied:** Informatie over gecompileerde programmeercode.

4.1 De Menubar

4.1.1 File

Hier vinden bewerkingen plaats met betrekking tot bestanden.



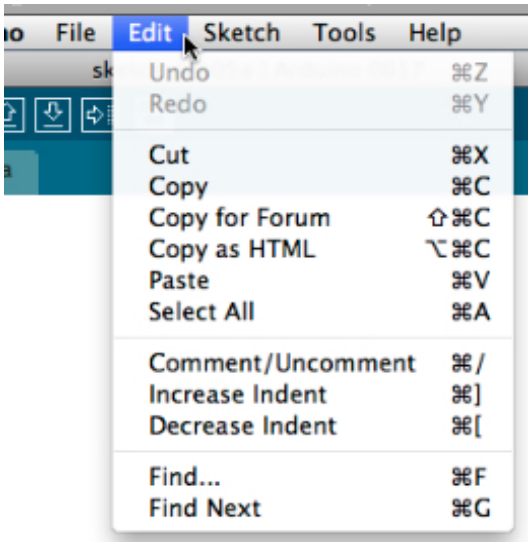
Belangrijk om te weten is dat de Arduino software een programma een Sketch noemt. Sketches die jij eerder gemaakt hebt vind je boven dan terug in de tab: **Sketchbook**.

De Arduino software bezit veel voorbeeldprogramma's. Deze vind je onder de tab: **Examples**.

Met **Upload to I/O Board** kopieer je de geschreven code naar de Atmel-microcontroller.

4.1.2 Edit

Hier voer je acties uit op tekst of code die jij geschreven hebt.



Een aantal tabs verdienen extra aandacht:

De Arduino-omgeving wordt gemaakt in de Open Sourcecommunitie. Daar zitten

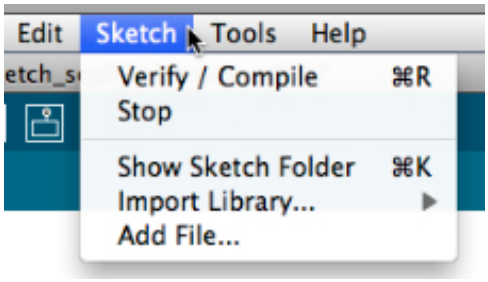
ontwikkelaars in die erg nauwkeurig luisteren wat de gebruikers graag willen. Een van de opties die naar voren kwam was het feit dat voor een vraag op een forum die in de Arduino-omgeving zo overzichtelijk gekleurd was de gekopieerde code zwart werd. Om toch de overzichtelijke gekleurde code te kunnen gebruiken werd de knop **Copy voor Forum** gemaakt. Dat zelfde geldt ook voor **Copy as HTML** waardoor je programmeercode kunt kopieren en rechtstreeks in website software (Een html-editor) kunt plakken.

Met het commando **Comment/Uncomment** kun je gemakkelijk commentaar (die niet mee geprogrammeerd wordt) invoegen.

Tot slot de tabs **Increase Indent** en **Decrease Indent.** Hiermee kun je code laten opschuiven (of terug schuiven) om er voor te zorgen dat programmeercode netjes onder elkaar komt te staan zodat de structuur wat overzichtelijker is.

4.1.3 Sketch

Met programmeercode die klaar is moet iets gebeuren:



De eerste mogelijkheid is het programma **Verifiëren** en te **Compileren**. Bij verifiëren wordt de programmeercode gecontroleerd op syntax fouten.

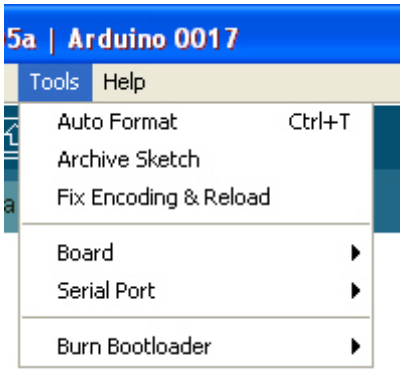
De tab **Stop** laat een programma dat is een oneindige lus zit stoppen.

Bij de **Import Libary…** tab kun je een nieuwe bibliotheek invoeren. Een bibliotheek is een set van instructies die een bepaald apparaat probleemlos laat werken. Je zult hier in het begin weinig tot geen gebruik van maken.

Bij de **Add File…** kun je bij de huidige sketch programmeercode toevoegen uit een gehele file.

4.1.4 Tools

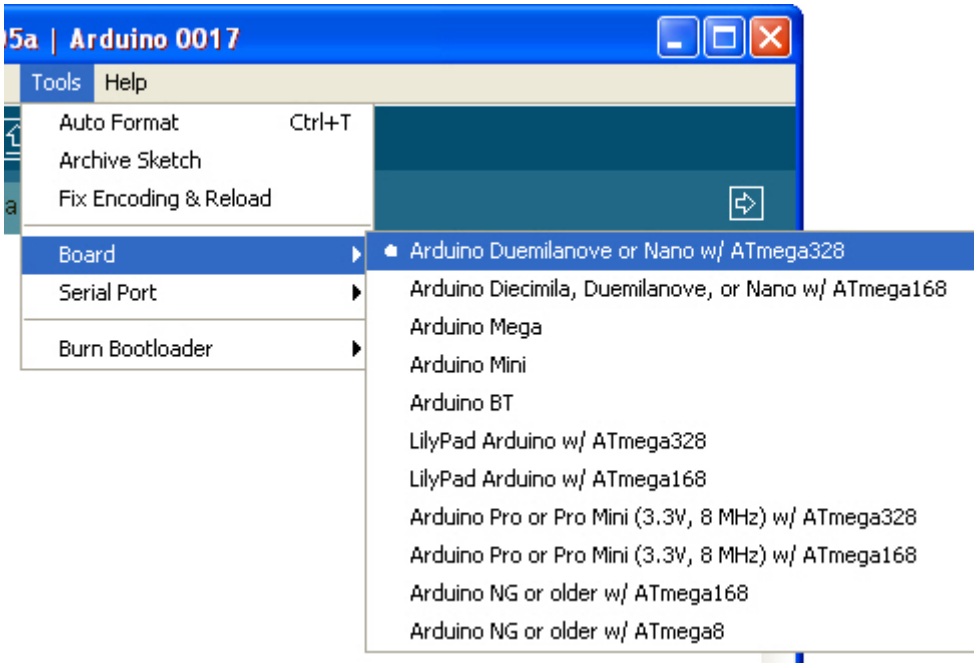
De Tool tab is een belangrijke tab. Deze zul je moeten gebruiken als je voor het eerst een Sketch schrijft.



Uit dit menu zul je voornamelijk te maken krijgen met de knop Board en Serial Port.

4.1.4.1 Board

Het is belangrijk om te weten welk board je gebruikt. De Arduino boards worden met enige (kleine) regelmaat vernieuwd in verschillende vormen waarin je moet denken in goedkoop en duur en veel of weinig kunnen.

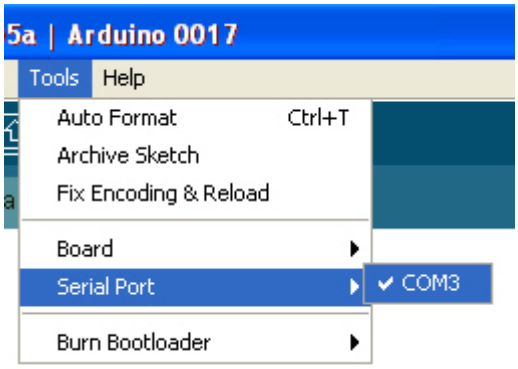


In het bovenstaande plaatje zie je de mogelijkheden met verschillende microcontrollers.

Wij gebruiken de ArduinoDuemilanove met een Atmega328.

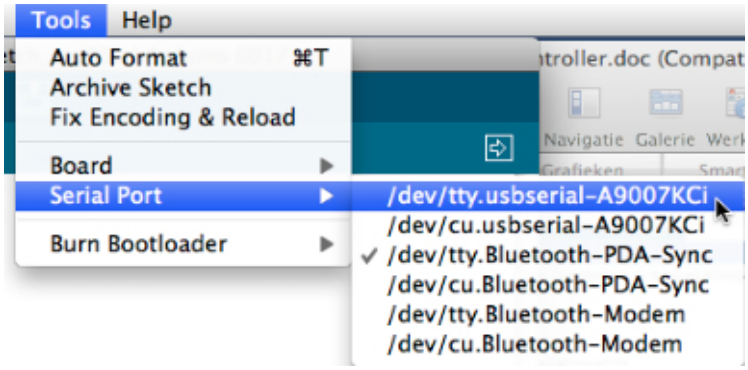
4.1.4.2 De seriële poort

De Arduino werkt alleen op een seriële poort of op een USB poort waarbij een seriële poort gemaakt wordt zodra de computer met de USB kabel aan de Arduino gekoppeld wordt.

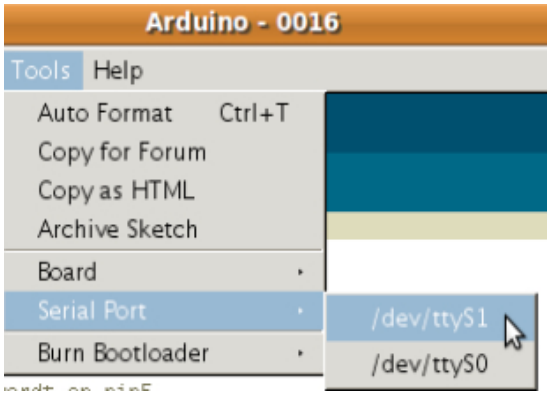


Je zult vooraf aan moeten geven op welke poort jouw Arduino is aangesloten. Zit er op jouw computer geen seriële poort? Of sluit je de Arduino aan op een USB poort dan zul in Windows even in de hardware eigenschappen moeten zoeken welke poort dat is.

Bij MAC OSx ziet dat er als volgt uit:

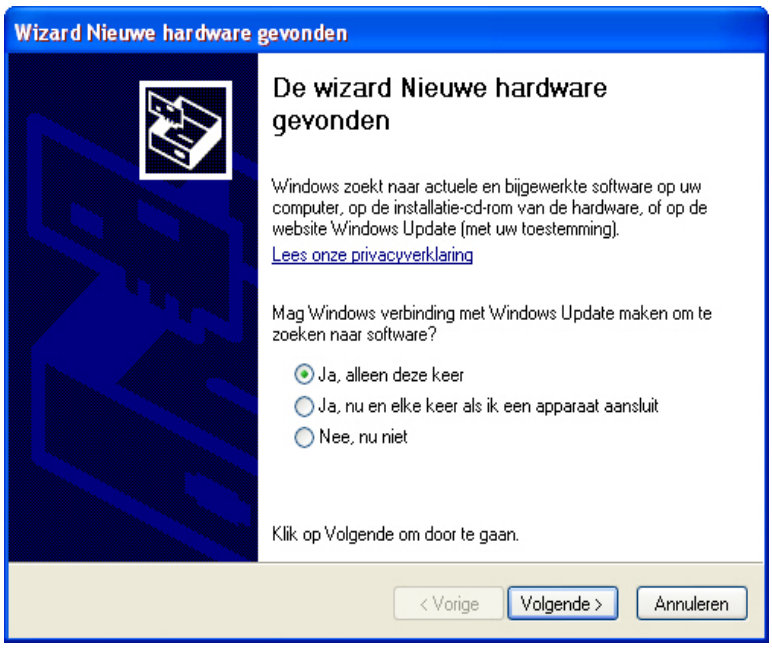


In Linux maak je de onderstaande keuze:



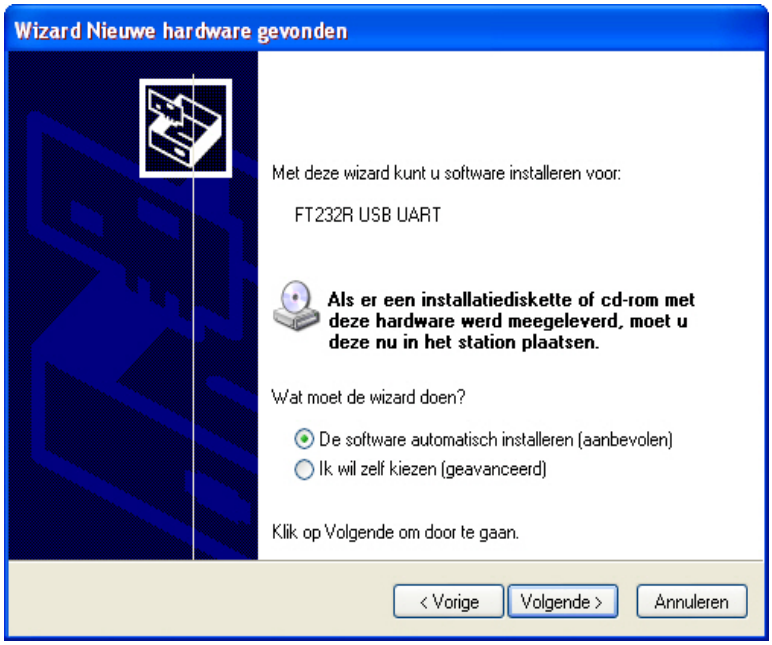
4.1.4.3 Installeren USB-Seriële Poort in Windows

Zodra je de Arduino voor de eerste keer aansluit krijg je het volgende scherm te zien:



Klik Ja, alleen deze keer

Het volgende scherm verschijnt:



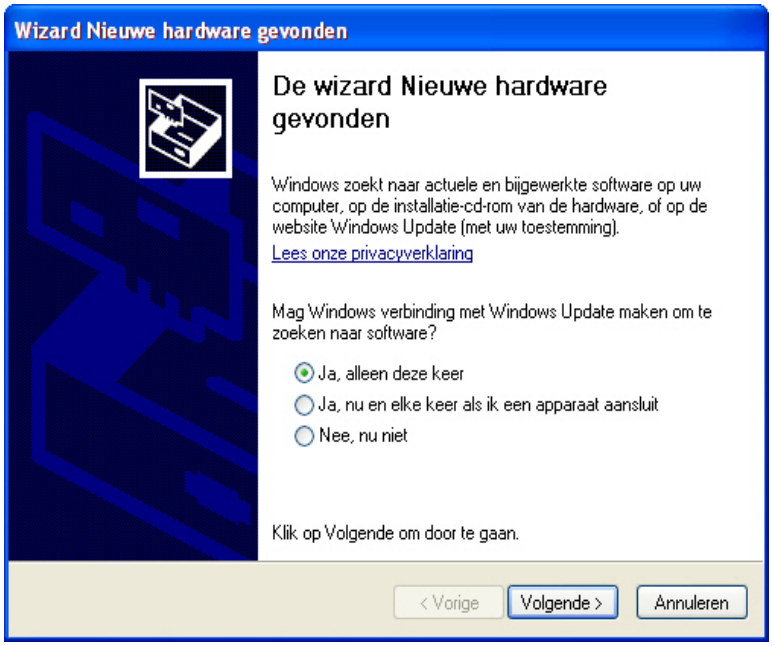
De computer vindt een zogenaamde FT232R USB UART. Dat is een chip die op de Arduino zit en een USB-signaal converteert naar een Serieel signaal. Kies de software automatisch installeren en klik volgende.

Let op: Je moet de software van de Arduino geïnstalleerd hebben op jouw PC (En geUNZIPt).

Als later de melding komt dat Windows hem niet kan vinden herstart de procedure (via de knop terug), kies dan…Ik wil zelf kiezen (geavanceerd) en blader naar de map waar je de Arduino software hebt geïnstalleerd en kies de map Drivers. Klik vervolgens op volgende.

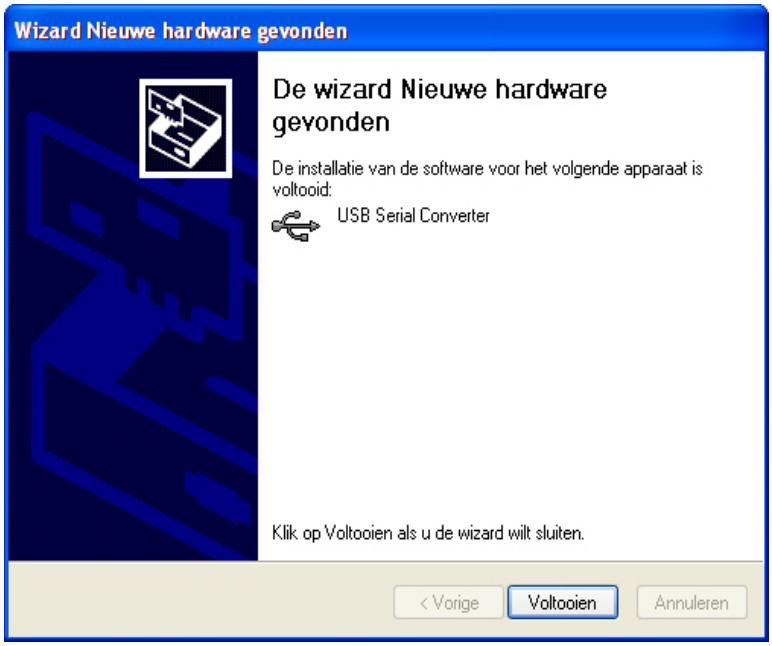
Als alles goed gegaan is verschijnt de melding dat de hardware geïnstalleerd is.

Afhankelijk van de computerconfiguratie verschijnt de mededeling:



Er is nu een seriële poort gevonden (aangemaakt door de hardware die net geïnstalleerd is).

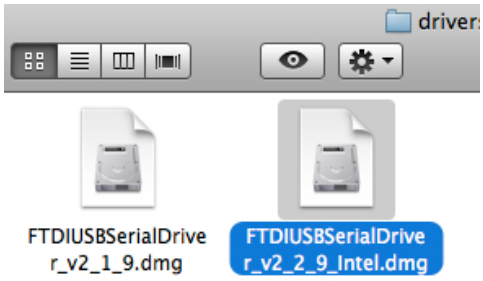
Kies wederom de software automatisch installeren en klik volgende. Na het installeren verschijnt het volgende scherm:



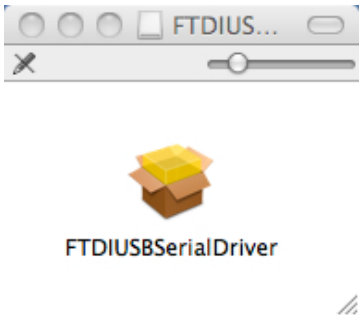
De Arduino is nu klaar voor gebruik.

4.1.4.4 Installeren USB-Seriële Poort in MAC OSx

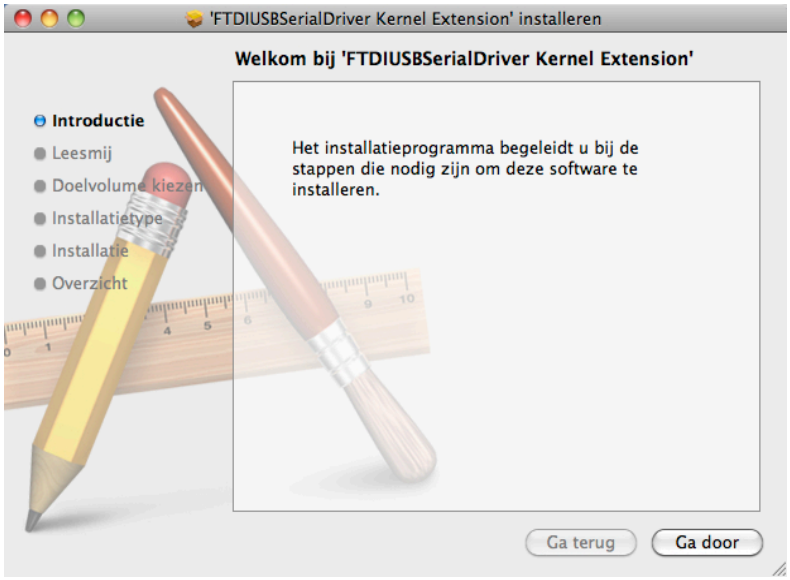
Blader naar de map waar je de Arduino software hebt geïnstalleerd en kies de map Drivers. En dubbelklik op:



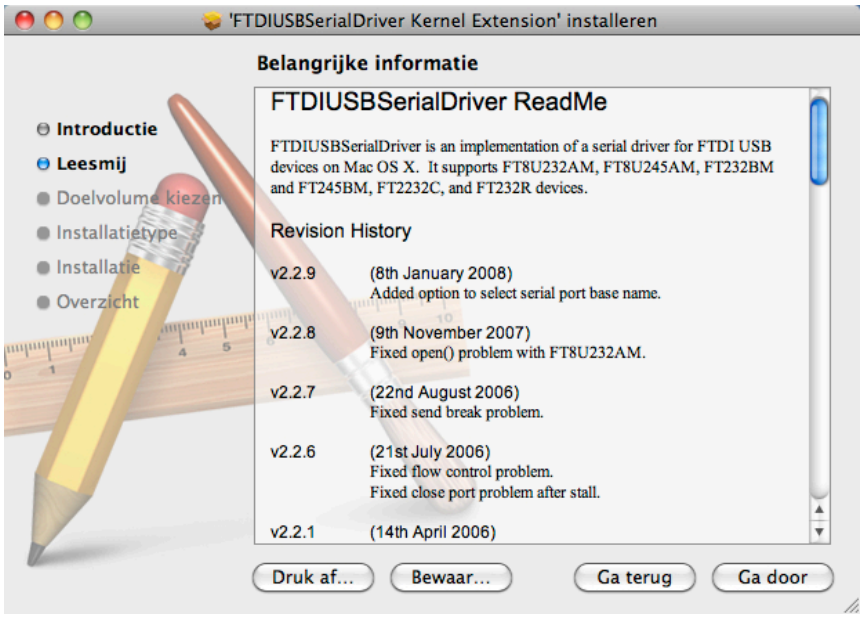
Het volgende scherm verschijnt. Dubbelklik op het icoon.



Het volgende scherm verschijnt:



Klik op Ga door en het volgende scherm verschijnt:

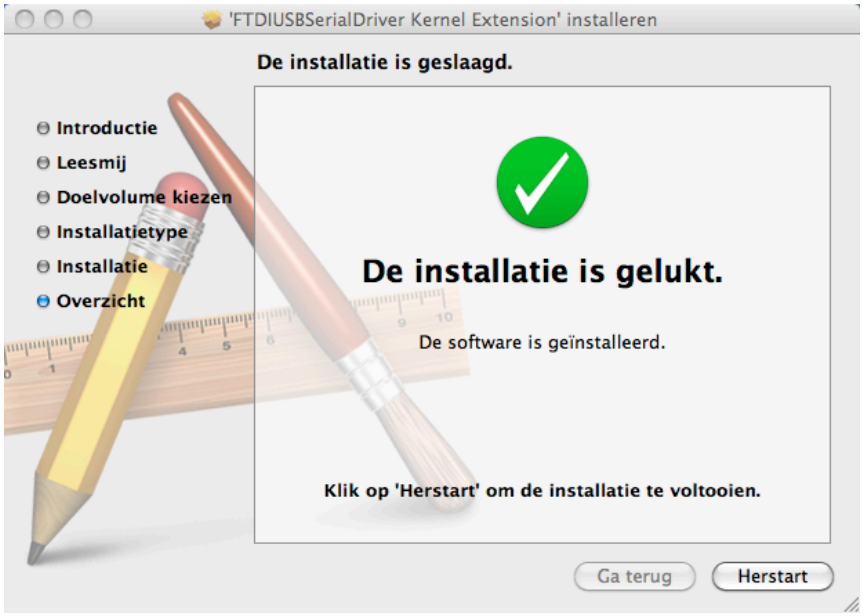


Klik op Ga door en het volgende scherm verschijnt:

Klik op Installeer en geef vervolgens het admistrator wachtwoord op en klik OK.
In het volgende scherm: klik op Ga door met de installatie.

Er verschijnt:

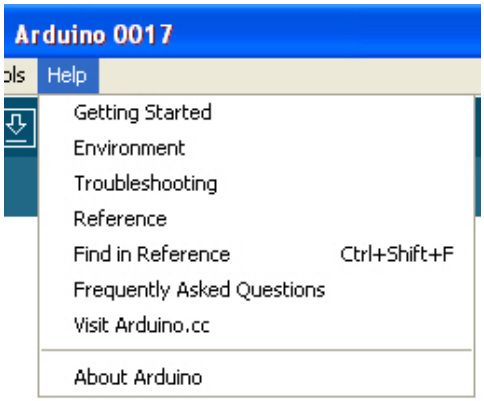
Klik op Annuleer en het volgende scherm verschijnt:



Klik op Herstart.
Komt na herstarten de volgende mededeling weer:

Kies dan wederom annuleer.

4.1.5 Help



Het Helpmenu spreek voor zich. Er is op internet enorm veel informatie te vinden, met als belangrijkste site: (Na http://www.arduino.nu) http://www.arduino.cc

4.1.6 De Toolbar

Verify/Compile: Controleer of er errors zitten in de geschreven code.

Stop: Stop de seriële monitor of deselecteer andere knoppen.

Nieuw: Maak een nieuw programma (sketch).

Open: Open een bestaand programma.

Save: Bewaar het huidige programma.

Upload: Schrijf het programma weg naar het Arduino I/O-bord.

Serial Monitor: Open een window waarin je ziet welke communicatie er plaats vindt.

5.0 Het aansluiten en programmeren van een Arduino.

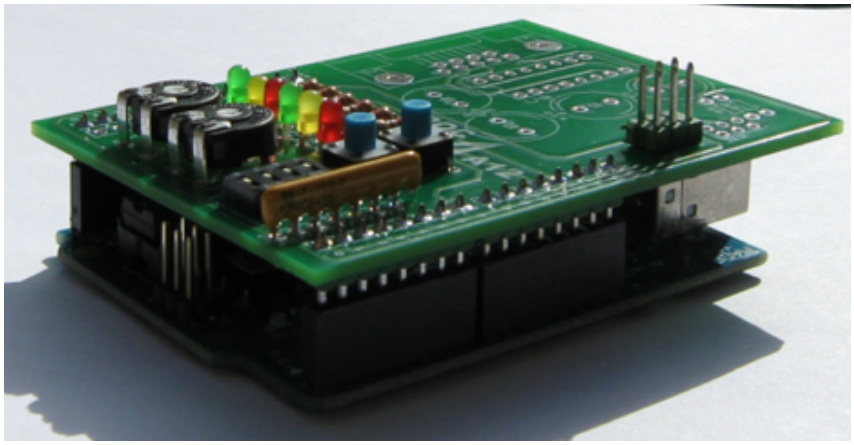
Op de Arduino zit op poort 13 een LED gemonteerd die in serie staat met een weerstand. Je kunt die poort (LED)dus direct al programmeren.

Neem het programma van de Arduino programmeermanual op pagina 26 letterlijk over en laat de LED knipperen.

Teken hier onder de flowchart van het knipperprogramma:

5.1 Uitbreidingsbord

In de volgende opdrachten heb je een “opzetbordje” nodig, handig op de Arduino te programmeren:

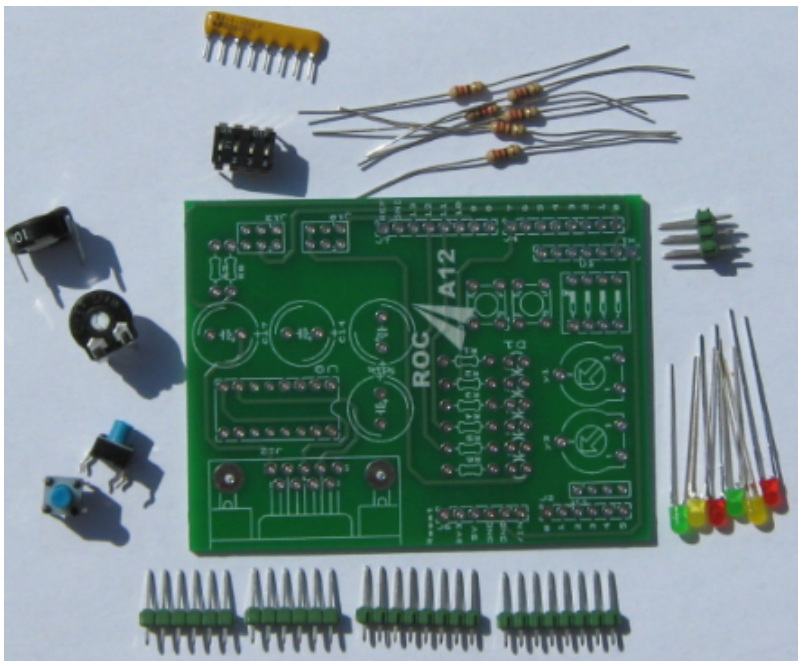


Dat bord kun je rechtstreeks op de Arduino plaatsen waardoor je direct de volgende mogelijkheden hebt:

1. Twee potmeters aangesloten op twee analoge ingangen
2. Zes LED’s aangesloten op zes outputpoorten.
3. Een dipswitch met vier schakelaars aangesloten op de inputpoorten.
4. Twee print-buttons aangesloten op de inputpoorten.
5. Twee connectoren waarop je rechtreeks twee servomotoren kunt aansluiten.

5.2 De assemblage van het uitbreidingsbord

Je krijgt de onderdelen van het uitbreidingsbord:

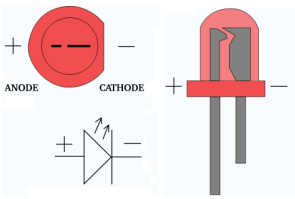
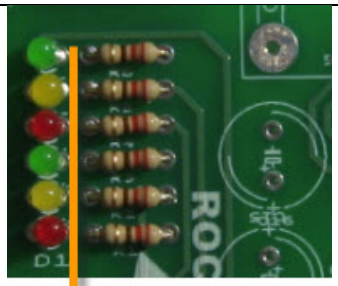


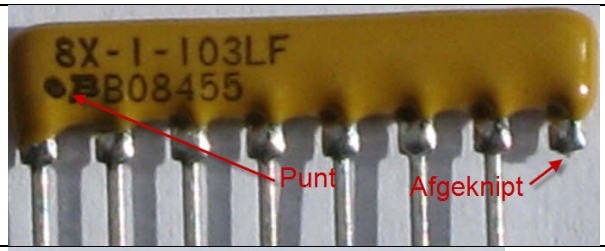
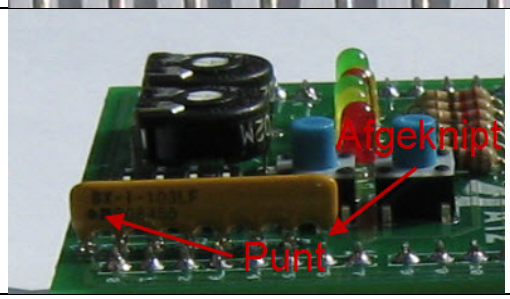
Dat zijn:

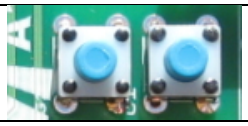
6 x Weerstand 220 Ω
1 x Weerstand array 8 x 10 kΩ
2 x Instelpotmeter 10 kΩ
1 x Dipswitch 4 posities
2 x Tactile print switch
2 x LED rood 3 mm
2 x LED groen 3 mm
2 x LED geel 3 mm
1 x Pin header male 3 pol 2 rijen
2 x Pin header male 6 pol 1 rij
2 x Pin header male 8 pol 1 rij
1 x Printplaat ROC-A12 uitbreidingsprint

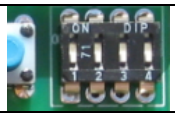
Werkvolgorde

Soldeer eerst de 6 weerstanden van 220 Ω		

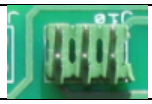
Nu ga je deLEDs solderen. Led daarbij goed op anode het lange pootje is op een LED. Kijk goed naar de twee foto's hiernaast.		 De oranje lijn geeft de anodes weer van de LED's. De anode op een LED is het lagere pootje.

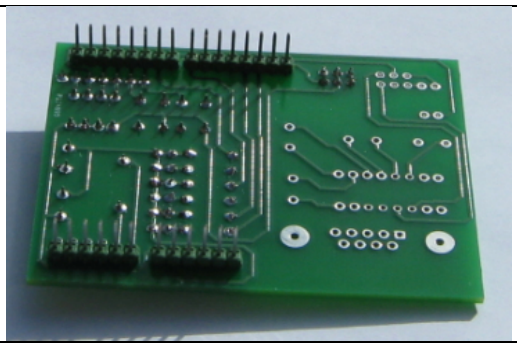
Vervolgens pak je de weerstands-array en knip je het pootje af aan de kant waar geen punt zit. Kijk goed naar de figuur hier naast.		
Dan soldeer je de array op de print. Kijk goed in de foto hier naast waar de punt zit. Het pootje dat afgekipt is rust gewoon op de print en kan verder geen kwaad.		

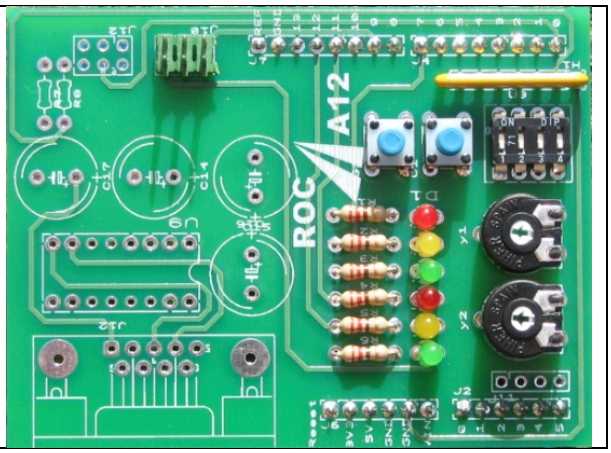
Soldeer nu de twee tactile print switches.		

Soldeer de DIP-switch.		

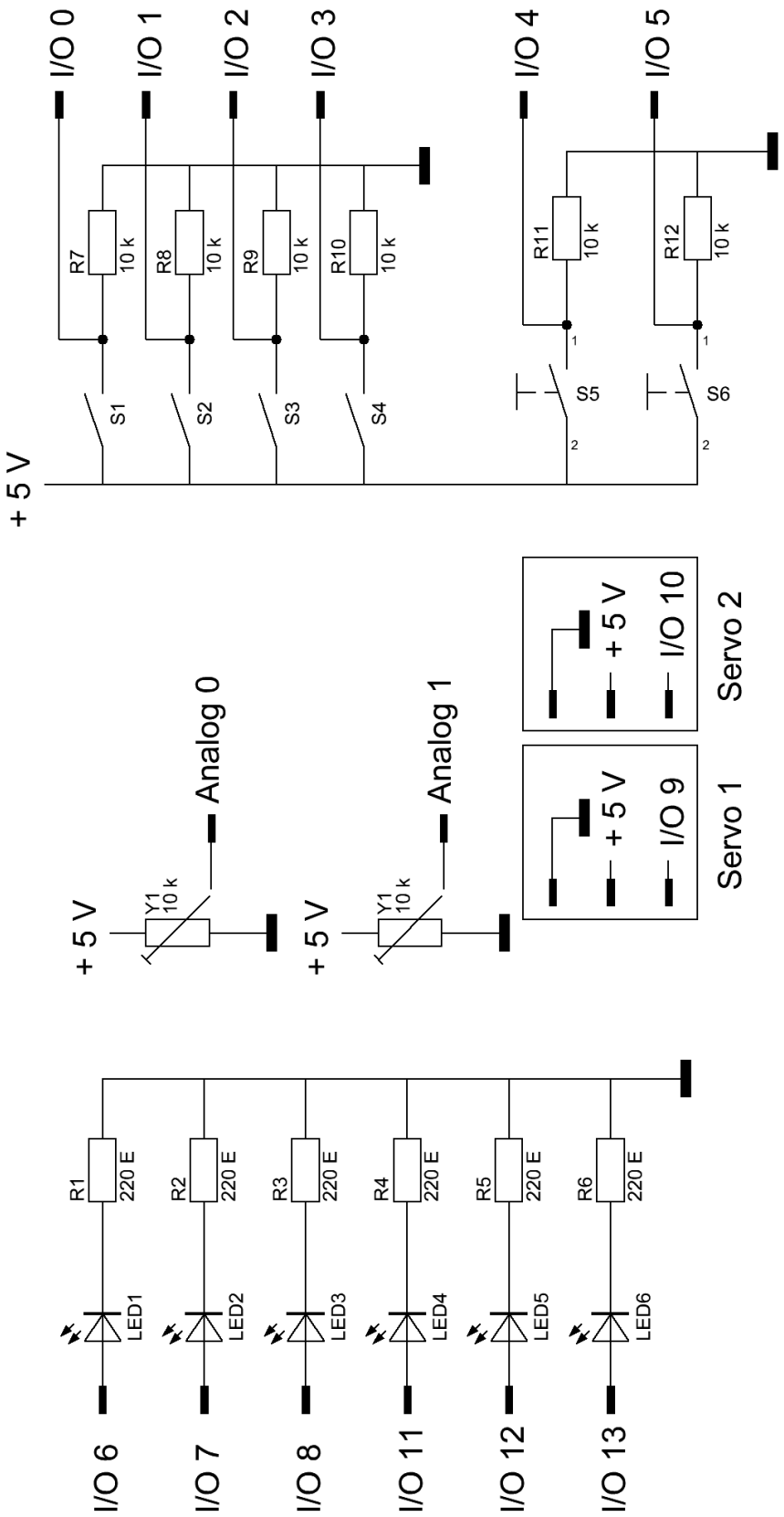
Soldeer de potmeters.		

Soldeer de pinheader op de plek van J10		

Tot slot monteer je aan de onderkant de 6 polige headers en 8 polige header. Deze soldeer je natuurlijk aan de bovenkant vast.		

En klaar is jouw uitbreidingsbord!! Je kunt hem nu op de Arduino zetten. Een groot aantal onderdelen zijn niet gemonteerd. Deze heb je dan ook niet nodig in dit werkboek.		

5.3 Schema Arduino uitbreidingsbord



6.0 Vragen en opdrachten

Beantwoord de volgende vragen uit het schema van de vorige pagina.

Wat is het verschil tussen I/O 0 en I/O 6 ?

Wat is de benaming van R7 ?

Wat is het verschil tussen S4 en S5 ?

Door een LED loopt een stroom van 15 mA. De LED heeft een doorlaatspanning van 1,7 V. Laat door middel van een berekening zien dat de keuze van 220 Ω een juiste keuze is.

Op de Arduino Duemilaove staat bij pin 13 PWM. Wat is PWM en wat doet dat ?

Opdracht 1.

Je gaat nu een verkeerslicht maken met drie LED's. Je maakt geen gebruik van een loop.

Teken eerst een flowchart.

Opdracht 1.

Schrijf het programma.

Opdracht2.

Je gaat nu een verkeerslicht maken met drie LED's. Je maakt gebruik van de loop for-next.

Teken eerst het flowchart.

Opdracht 2.

Schrijf het programma.

Opdracht 3.

Je gaat nu een verkeerslicht maken met drie LED's. Je maakt gebruik van de loop do-while.

Teken eerst een flowchart.

Opdracht 3.

Schrijf het programma.

Opdracht 4.

Je gaat nu een LED laten branden als je S5 indrukt. Als je S5 los laat blijft de LED aan. Met S6 gaat de LED uit.

Teken eerst een flowchart.

Opdracht 4.

Schrijf hier het programma.

Opdracht 5.

Bekijk en het programma dat je hier onder ziet en zet het in de Arduino. Beantwoord daarna de vragen hier onder.

```
/*
 * Looplicht met ROC-A12 Uitbreidingsbord.
 */

int i = 0;
int a = 0;

void setup()
{
  for (i = 6; i < 14; i++)
    {
      pinMode(i, OUTPUT);
    }
}

void loop()
{
   for (i = 6; i < 14; i++)
   {
     a=i;
     if (a < 9 | | a > 10)
       {
         digitalWrite(i, HIGH);
         delay(200);
         digitalWrite(i, LOW);
         delay(200);
       }
   }
}
```

Opdracht 5.

Teken van het programma op de vorige bladzijde een flowchart.

Opdracht 6.

Het volgende programma laat een LED snellen/langzamer knipperen al naar gelang de positie van één potmeter.

Teken eerst een flowchart.

Opdracht 6.

Schrijf het programma.

Opdracht 7.

Bij het volgende programma gaat een LED aan als een potmeter 2/3 van zijn maximale waarde heeft bereikt. Bij het terug draaien van de potmeter gaat de LED pas uit als hij de stand van 1/3 heeft bereikt.

Teken eerst een flowchart.

Opdracht 7.

Schrijf het programma.

Opdracht 8.

Zes LED's gaan aan als potmeter 1 een waarde heeft bereikt van 2/3 en potmeter 2 een waarde heeft van meer dan 1/3.
Zes LED's gaan uit als potmeter 2 een waarde heeft van minder dan 1/3 en potmeter P2 een waarde heeft van minder dan 2/3.
In alle overige situaties knipperen al de zes LED's.

Teken eerst een flowchart.

Opdracht 8.

Schrijf het programma.

Opdracht 9.

Codeslot: Een Led gaat aan als op de dipswitch de code 1011 is ingevoerd.

Teken eerst een flowchart.

Opdracht 9.

Schrijf het programma.

Opdracht 10.

Laat de tafel van 16 zien met behulp van de serial monitor.

Teken eerst een flowchart.

Opdracht 10.

Schrijf het programma.

Opdracht 11.

Maak met 6 LED's een **digitale** teller die om de 0,5 seconde één tel verder gaat.

Teken eerst een flowchart.

Opdracht 11.

Schrijf het programma

Opdracht 12.

Herhaal de vorige opdracht en laat de teller twee keer zo snel open als S1 ingeschakeld is.

Teken eerst een flowchart.

Opdracht 12.

Schrijf het programma.

Opdracht 13.

Sluit een servomotor aan die continue kan draaien op het uitbreidingsbord en laat de servo draaien. Gebruik het programma hier onder:

Teken eerst een flowchart.

Het programma:

servotest | Arduino 0017

File Edit Sketch Tools Help

servotest §

```
#include <Servo.h>
Servo myservo1;  //koppel servo aan myservo1

void setup()
{
  myservo1.attach(9);  // Servo op pin9
}

void loop()
{
   myservo1.write(90);
}
```

Opdracht 14.

Sluit een servomotor aan die continue kan draaien op het uitbreidingsbord en laat de servo draaien. Als S5 ingedrukt is staat de servomotor stil. Met S1 kun je de draairichting veranderen.

Teken eerst een flowchart.

Opdracht 14.

Schrijf het programma.

Opdracht 15. Reactiespel 1: Maak een looplicht van zes LED's. Als S5 wordt ingedrukt en LED 2 is aan dan knipperen alle zes LED's 5 x.
Teken eerst een flowchart.

Opdracht 15.

Schrijf het programma.

Opdracht 16.

Reactiespel 2: Maak een looplicht van zes LED's. Als S5 wordt ingedrukt en LED 2 is aan dan knipperen alle zes LED's 5 x. De snelheid van het looplicht wordt bepaald door de stand van Potmeter P1.

Teken eerst een flowchart.

Opdracht 16.

Schrijf het programma.

Opdracht 17.

Maak een lijnvolger (robot).

Teken eerst het schema:

Opdracht 17.

Teken een flowchart.

Opdracht 17.

Schrijf het programma.

Atmel AVR

Uit Wikipedia, de vrije encyclopedie

Atmel AVR ATmega8.
De **AVR** is een 8-bit RISC microcontroller (µC) ontwikkeld door Atmel in 1996. De AVR was één van de eerste microcontroller-families die gebruik maakte van on-chip flash geheugen voor programma opslag in plaats van PROM, EPROM of EEPROM.

Apparaat Overzicht

De AVR is een Modified Harvard-architectuur apparaat waarbij het programma en de data worden opgeslagen in aparte fysieke geheugensystemen.

Families

AVR's worden algemeen ingedeeld in vier groepen:

- tinyAVR — de ATtiny serie
  - 1–8 kB programma geheugen
  - 6–32-pin behuizing
  - Gelimiteerde peripheral set
- megaAVR — de ATmega serie
  - 4–256 kB programma geheugen
  - 28–100-pin behuizing
  - Uitgebreide instructie set (Vermenigvuldig instructies en instructies voor het gebruik van grotere programma geheugens)
  - Uitgebreide peripheral set
- XMEGA — de ATxmega serie
  - 16–384 kB programma geheugen
  - 44–64–100-pin behuizing (A4, A3, A1)
  - Uitgebreide prestatie eigenschappen, zoals DMA, "Event System" en cryptografie ondersteuning
  - Zeer uitgebreide peripheral set met DAC's
- Applicatie specifieke AVR
  - megaAVRs met speciale mogelijkheden die niet voorkomen bij andere leden van de AVR familie, zoals LCD controllers, USB controllers, geavanceerde PWM, CAN, etc0

Architectuur

Data RAM

Flash, EEPROM, en SRAM zijn allemaal geïntegreerd in één chip, waardoor de noodzaak voor extern geheugen verdwijnt. Sommige apparaten hebben een externe bus voor extra data geheugen.

Programma Geheugen (Flash)

Programma-instructies worden opgeslagen in non-volatile Flash geheugen. Hoewel het 8-bits MCU's zijn, gebruikt elke instructie één of twee 16-bits words.
De grootte van het programma geheugen wordt meestal aangeduid in de naam van het apparaat (vb. de ATmega64 lijn heeft 64 kB Flashgeheugen.
Er is geen mogelijkheid voor extern programma geheugen; alle code uitgevoerd door de AVR kern moet in het on-chip flashgeheugen zitten.

Intern datageheugen
De data-adresruimte bestaat uit het registerbestand, I/O registers en SRAM.

Interne registers
De AVR's hebben 32 single-byte registers en zijn geclassificeerd als 8-bit RISC apparaten.
In de meeste varianten van de AVR architectuur worden de werk registers gemapped in de eerste 32 geheugen adressen (16#0000-16#001F) gevolgd door de 64 I/O registers (16#0020-16#005F).
Het daadwerkelijke SRAM begint na deze registers (adres 16#0060). (De I/O register ruimte kan groter zijn bij sommige uitgebreidere apparaten waarbij dus een deel van het SRAM gedeelte word gebruikt).
Hoewel het register bestand en de I/O registers apart kunnen worden aangesproken met speciale opcodes, kunnen ze ook worden aangesproken en gemanipuleerd alsof ze in het SRAM zitten).

EEPROM
Bijna alle AVR microcontrollers hebben intern "Electrically Erasable Programmable Read Only Memory" (EEPROM) voor semi-permanente data opslag. Net als het flash geheugen behoudt het EEPROM zijn data als de spanning wegvalt.
In de meeste AVR architectuur varianten wordt het interne EEPROM geheugen niet gemapped in de adresseerbare geheugen ruimte van de MCU. Het kan alleen worden aangesproken middels speciale pointer registers en lees/schijf instructies. Hierdoor is het EEPROM geheugen een stuk langzamer dan het interne SRAM.
Doordat het aantal schrijfbewerkingen van EEPROM geheugen beperkt is - Atmel specificeert 100.000 schrijfbewerkingen in zijn datasheets - zou een goed ontworpen EEPROM schrijfroutine moeten controleren of de data afwijkt van de gewenste data alvorens deze wordt geschreven.

Programma-uitvoering
Atmel'sAVR's hebben een tweestaps, enkel niveau pijplijn-ontwerp. Dit houd in dat de volgende machine-instructie wordt opgehaald als de huidige wordt uitgevoerd. De meeste instructies duren slechts één of twee cycli, wat de AVR's relatief snel maakt onder de 8-bits microcontrollers.
De AVR familie van processors werd ontworpen met de efficiënte uitvoering van gecompileerde C code in gedachten en heeft diverse ingebouwde pointers voor de taak.

Instructieset
De AVR instructie set is meer Orthogonaal dan de meeste 8-bits microcontrollers, met name de PIC microcontrollers waarmee AVR concurreert. Het is echter niet compleet regulier:

- Pointer registers X, Y, en Z hebben adresseringsmogelijkheden die van elkaar afwijken.
- Register locaties R0 tot R15 hebben andere addresserings mogelijkheden dan register locaties R16 tot R31.
- I/O poorten 0 tot 31 hebben andere adresseringsmogelijkheden dan I/O poorten 32 tot 63.
- CLR beïnvloedt flags, waar SER dat niet doet, hoewel het de exact omgekeerde instructies zijn. CLR zet alle bits op nul terwijl SER alle bits op één zet.

Verder beïnvloeden sommige chip-specifieke verschillen de code generatie. Code pointers bestaan uit twee bytes op chips met 128 kB of minder flash geheugen, maar drie bytes op de grotere chips; niet alle chips hebben hardware vermenigvuldigers; chips met meer dan 8 kB flash geheugen hebben branch en call instructies met langere ranges; etc.
De grotendeels reguliere instructie set maakt het programmeren middels C (of zelf Ada) compilers vrij eenvoudig. GCC heeft al enige tijd AVR ondersteuning welke veel gebruikt wordt. Atmel vroeg zelfs om input van grote compiler ontwikkelaars voor kleine

microcontrollers, om te bepalen welke instructie set eigenschappen het nuttigste zijn voor een high-level language compiler.

Snelheid

De AVR lijn ondersteund normaal snelheden van 0-20 Mhz, waarbij sommige apparaten tot 32 Mhz reiken. Lager vermogen operatie vereist meestal een gereduceerde clock snelheid. Alle recente (Tiny en Mega, niet de 90S) AVR's hebben een on-chip oscillator, waardoor er geen externe clock of resonantie circuit meer nodig is. Sommige AVR's hebben ook een systeem clock deler, die de clock kan delen met tot 256. Deze prescaler kan worden geherconfigureert tijdens run-time, wat een geoptimaliseerde clock snelheid toestaat. Aangezien alle operaties op registers R0-R31 een enkele cyclus gebruiken, kan de AVR tot 1 MIPS (Miljoen Instructies Per Seconde) per Mhz halen. Laden en opslaan van en naar het geheugen duurt 2 cycli, branching duurt 3 cycli. Branching in de laatste "3-byte PC" onderdelen zoals de ATmega2560 zijn 1 cycli langzamer.

Ontwikkeling

AVR's worden veel gebruikt doordat er gratis en goedkope ontwikkelhulpmiddelen beschikbaar zijn, inclusief redelijk geprijsde ontwikkelborden en gratis ontwikkel software. De AVR's worden verkocht onder verschillende namen die dezelfde basis kern hebben, maar verschillende peripherals en geheugencombinaties hebben. Compatibiliteit tussen chips van elke familie is vrij goed, hoewel I/O controller mogelijkheden variëren.

Mogelijkheden

AVR's bieden een grote reeks mogelijkheden:

- Multifunctie, Bi-directionele algemeen bruikbare I/O poorten met configureerbare pull-up weerstanden.
- Meerdere interne oscillatoren, inclusies RC oscillator zonder externe onderdelen
- Interne, zelf programmeerbare instructie flash geheugen tot 256 KB.
- In-Systeem programmeerbaar middels seriele/parallelelaag-voltage interfaces of JTAG
    - Optionele Boot Code Sectie met onafhankelijke Lock Bits voor beveiliging
- On-Chipdebugging (OCD) ondersteuning middels JTAG of debugWIRE op de meeste apparaten
    - De JTAG signalen (TMS, TDI, TDO en TCK) zijn gemultiplexed op GPIO's. Deze pinnen kunnen worden geconfigureerd voor JTAG of GPIO, afhankelijk van de instelling van de fuse bit, welke middels ISP of HVSP kan worden geprogrammeerd. Standaard staat JTAG aan.
    - Debug WIRE gebruikt de /RESET pin als bi-directioneel communicatiekanaal om het on-chipdebug circuit aan te spreken. Het is aanwezig op apparaten met een laag aantal pinnen, aangezien het maar één pin gebruikt.
- Interne Data EEPROM tot 4 kB
- Interne SRAM tot 8 kB (32 kB bij XMega)
- Externe 64 kB little endian data ruimte op bepaalde modelling, zoals the Mega8515 en Mega162
    - De externe data ruimte is over de interne data ruimte gelegd, zodat de volledige 64 kB adres ruimte niet voorkomt op de externe bus. Een adres hogen dan 16#0100 zal intern RAM aanspreken, niet de externe bus.
    - In bepaalde leden vna de XMEGA serie is de externe data ruimte verbeterd voor ondersteuning van zowel SRAM als SDRAM. Tevens zijn de data adresseringsmodes uitgebreid om directe aanspreking tot 16MB aan data geheugen toe te staan.
    - AVR's hebben doorgaans geen ondersteuning voor uitvoering van code vanaf extern geheugen. Sommige ASSP's (Applicatie Specifieke Standaard Product) ondersteuning dit wel.
- 8-bits en 16-bits timers
    - PWM output

    - Input Capture
- Analoge Comperator
- 10 of 12 bits A/D convertors, met multiplex tot 16 kanalen
- 12 bit D/A convertors
- Verschillende seriële interfaces, zoals
    - I2C
    - Synchrone/Asynchrone Seriële Peripherals (UART/USART) (gebruikt met RS-232, RS-485 en meer)
    - SerialPeripheral Interface Bus (SPI)
    - Universal Serial Interface (USI)
- Brownout Detectie
- Watchdog Timer (WDT)
- Meerdere vermogen besparing slaap modes
- Licht en motor control (PWM)
- CAN controller ondersteuning
- USB controller ondersteuning
    - Volledige Full-speed (12 Mbit/s) hardware &Hub controller met embedded AVR
    - gratis beschikbare Low-speed (1.5 Mbit/s) (HID)
- Ethernet controller ondersteuning
- LCD controller ondersteuning
- Laag-voltage apparaten werken vanaf 1.8v
- picoPower apparaten
- DMA controllers en "Event System" peripheral communicatie
- Snelle cryptografie ondersteuning voor AES en DES

Ontwikkelhulpmiddelen en evaluatiekits

De officiële Atmel AVR ontwikkelhulpmiddelen en evaluatiekits bestaan uit een aantal starterskits en debughulpmiddelen met ondersteuning voor de meeste AVR apparaten:

STK600 Starter Kit

De STK600 starterskit en ontwikkelsysteem is een update van de STK500. De STK600 gebruikt een basisbord, een signaalroutingbord en een targetbord.
Het basisbord is hetzelfde als de STK500 en heeft een spanningsbron, een clock, in-systeem programmeren, RS-232 poorten en connectoren voor de GPIO signalen van het doel apparaat.
Het targetbord heeft ZIF sockets voor DIP, QFN of QFP behuizingen, afhankelijk van het bord.
Het signaalroutingbord zit tussen het basisbord en het targetbord, en verbind de signalen aan de juiste pinnen op het targetbord. Er zijn veel verschillende routingborden die gebruikt kunnen worden met een enkel targetbord, afhankelijk van het apparaat in de ZIF socket.
De STK600 wordt verbonden met de PC via een USB aansluiting, waardoor beide RS-232 poorten beschikbaar zijn voor de microcontroller.

STK500 Starter Kit

De STK500 starterskit en ontwikkelsysteem heeft een ISP en een HVP (High Voltage Programming) voor alle AVR apparaten, ofwel direct of via een uitbreidingsbord. Het bord bevat DIP sockets voor alle beschikbare AVR's in DIP behuizingen.
Er zijn verschillende uitbreidingsmodules beschikbaar voor de STK500:

- STK501 - Voegt ondersteuning toe voor microcontrollers in 64-pins TQFP behuizingen
- STK502 - Voegt ondersteuning toe voor LCD AVR's in 64-pins TQFP behuizingen
- STK503 - Voegt ondersteuning toe voor microcontrollers in 100-pins TQFP behuizingen
- STK504 - Voegt ondersteuning toe voor LCD AVR's in 100-pins TQFP behuizingen
- STK505 - Voegt ondersteuning toe voor 14 en 20-pins AVR's
- STK520 - Voegt ondersteuning toe voor 14 en 20-pins microcontrollers van de AT90PWM familie

- STK524 - Voegt ondersteuning toe voor de ATmega32M1/C1 32-pins CAN/LIN/Motor Control familie
- STK525 - Voegt ondersteuning toe voor de AT90USB microcontrollers in 64-pin TQFP behuizingen
- STLK526 - Voegt ondersteuning toe voor de AT90USB microcontrollers in 32-pins TQFP behuizingen

AVR ISP en AVR ISP mkII

De AVR ISP en AVR ISP mkII zijn goedkope hulpmiddelen om AVR's te programmeren via ICSP De AVR ISP wordt verbonden met de PC via een seriële poort en haalt zijn spanning uit het doelsysteem. De AVR ISP staat gebruik van beide "standaard" ICSP pinouts toe, de 10-pins of de 6-pins connector. De AVR ISP wordt niet meer geproduceerd en is vervangen door de AVR ISP mkII.
De AVR ISP mkII wordt verbonden met de PC via een USB aansluiting en haalt zijn spanning van de USB poort. LED's die zichtbaar zijn door de doorzichtige behuizing geven de status van spanning van het doelsysteem aan.

AVR Dragon

De Dragon is een goedkoop hulpmiddel dan wordt verbonden met de PC via een USB aansluiting. De Dragon kan alle AVR's programmeren via JTAG, HVP of ICSP. De Dragon biedt ook ondersteuning voor het debuggen van AVR's met minder dan 32KB aan programmageheugen, via JTAG of debug WIRE. De Dragon heeft een klein prototypegebied waar een 8, 28 of 40-pins AVR kan worden aangesloten.

JTAGICE mkI en mkII

Het Joint Test Action Group, In Circuit Emulator (JTAGICE) debug hulpmiddel biedt ondersteuning voor on-chipdebugging (OCD) van AVR's met een JTAG interface. De originele JTAGICE mkI gebruikt een RS-232 interface als verbinding met de PC. De JTAGICE mkI wordt niet meer geproduceerd en is vervangen door de JTAGICE mkII.
Het JTAGICE mkIIdebug hulpmiddel ondersteunt on-chipdebugging (OCD) van AVR's middels de JTAG, PDI of debug WIRE interface. De debug WIRE interface gebruikt slecht één pin (de Reset pin), waardoor debuggen mogelijk is voor applicaties met een microcontroller met weinig pinnen.
De JTAGICE mkII wordt verbonden met de PC via een USB aansluiting, maar er is een alternatieve aansluiting via een seriële poort, welke een losse voeding nodig heeft. Naast JTAG ondersteunt de mkII programmeren via ISP (via de 6 of 10-pins adapters). Zowel de USB als de seriële variant gebruikt een versie van het STK500 protocol.

Butterfly Demo Bord

Het zeer populaire AVR Butterfly demonstratiebord is een zelfvoorziend, batterij aangedreven computer die draait op de Atmel AVR ATmega169V microcontroller. Het was gebouwd om te pronken met de AVR familie, in het bijzonder de nieuwe ingebouwde LCD interface. Het bord bevat een LCD scherm, een joystick, een speaker, een seriële poort, realtime clock (RTC), flash geheugenchip en zowel temperatuur als spanningssensoren. Eerdere versie van de AVR Butterfly bevatten ook een CdS photoresistor; deze is niet meer aanwezig op borden geproduceerd na juni 2006 om aan de RoHS norm te voldoen. Het kleine bord heeft een shirt-pin aan de achterkant zodat deze als naamplaatje kan worden gedragen.
De AVR Butterfly komt voorgeladen met software om de mogelijkheden van de microcontroller te demonstreren. De fabrieksfirmware kan je naam laten scrollen, de sensorwaarden weergeven en de tijd weergeven.
De AVR Butterfly demonstreert LCD aansturing door een 14-segments, 6 alpha-numerieke karakter display aan te sturen. De LCD-interface gebruikt echter wel veel van de I/O pinnen. De ATmega169 CPU van de Buttferly kan snelheden tot 8 Mhz aan, maar is door de fabrieksfirmware ingesteld op 2 Mhz om de batterijduur te verlengen. Een voorgeïnstalleerd bootloader programma geeft de mogelijkheid het bord te herprogrammeren via een

standaard RS-232 seriële aansluiting met nieuwe programma's die gebruikers kunnen schrijven met het gratis Atmel IDE hulpmiddel.

AT90USBKey

Dit kleine bord, ongeveer de heft van een creditcard, is iets duurder dan de AVR Butterfly. Het bevat een AT90USB1287 microcontroller met USB-On-The-Go (OTG), 16 MB DataFlash, LED's, een kleine joystick en een temperatuursensor. Via de bijgeleverde software kan het bord zich gedragen als een USB opslagapparaat, een USB joystick en meer. Om als USB Host te werken is een batterij nodig, maar als USB randapparaat gebruikt het alleen de spanning van de USB poort.
Alleen de JTAG aansluiting gebruikt de normale 2.54 mm pinafstand. Alle andere AVR I/O poorten gebruiken compactere 1.27 mm headers.
Doordat de AT90USBKey 128 KB aan flashgeheugen heeft kan een AVR Dragon hem alleen programmeren en niet debuggen. Een JTAGICE kan hem zowel programmeren als debuggen. De processor kan ook worden geprogrammeerd via USB vanaf een Windows of Linux host, middels USB "Device Firmware Update" protocollen.
MyUSB is een gratis software USB protocol stack voor de USBKey gemaakt door derden (MIT license).

Raven Wireless Kit

De RAVEN kit ondersteunt draadloze ontwikkeling middels Atmel's IEEE 802.15.4 chipsets voor ZigBee en andere draadloze stacks. Het stelt één paar draadloze Buttefly kaarten voor plus een draadloze USBKey en kost ongeveer hetzelfde als deze samen (onder $100). Alle borden ondersteunen ontwikkeling via JTAG.
De kit bevat twee AVR Raven borden, beide met een 2.4 GHz transciever met ondersteuning voor IEEE 802.15.4 (en een Zig Beestack met gratis licentie). De radio's worden aangestuurd met ATmega1284p procesoren, welke ondersteund worden door een aangepast gesegmenteerd LCD display aangestuurd door een ATmega3290p processor. Raven peripherals lijken of de Butterfly; piëzo speaker, DataFlash (groter), externe EEPROM, sensoren, 32 KHz kristal voor RTC etc. Deze zijn bedoeld voor het ontwerpen van op afstand aanstuurbare sensor modules, het aansturen van relais, of wat ook nodig.
De USB stick gebruikt een AT90USB1287 voor verbindingen met een USB host en de 2.4 GHz draadloze verbindingen. Ze zijn bedoeld voor het uitlezen en aansturen van draadloze modules, gebruikmakend van spanning van de host i.p.v. batterijen.

Programmers door derden

Er is een groot aantal programmeer- en debug hulpmiddelen beschikbaar van derden. Deze apparaten gebruiken verschillende interfaces, waaronder RS-232, parallelle poort en USB.

Atmel AVR Gebruik

AVR's worden gebruikt in verschillende automotive toepassingen zoals beveiliging, veiligheid en entertainment systemen. Atmel heeft recentelijk een nieuwe publicatie gelanceerd genaamd "Atmel Automotive Compilation" , om nieuwe ontwikkelaars te helpen met automotive toepassingen.
Het Arduino computerplatform is gebaseerd op een ATmega328 microcontroller (ATmega168 of ATmega8 in oudere versies). De ATmega1280, met meer pinnen en geheugen beschikbaar, is ook gebruikt om het Arduino Mega platform te ontwikkelen. Arduino borden kunnen worden gebruikt met zijn eigen taal en IDE, of met meer conventionele programmeeromgevingen (C/ASM/etc) zoals the standaard AVR platforms.
USB-gebaseerde AVR's zijn gebruikt in de Microsoft Xbox hand controllers. De link tussen the controllers en de Xbox is USB.
Verschillende bedrijven produceren AVR-gebaseerde microcontroller borden, bedoeld voor gebruik door hobbyisten, robot bouwers, experimenteurs en klein systeem ontwikkelaars.
System Semiconductor,Inc produceert de M3000 Motor en Motion Control, waarin een Atmel AVR kern en een Advanced Motion Controller zijn opgenomen.

Nieuwe 32-bits AVR's

In 2006 bracht Atmel microcontrollers uit gebaseerd op een nieuwe, 32-bits, AVR32 architectuur. Deze bevatten SIMD en DSP instructies, samen met andere audio en video verwerkingsmogelijkheden. Deze 32-bits familie van apparaten is bedoeld om te concurreren met de op ARM gebaseerde processoren. De instructieset lijkt op andere RISC kernen, maar is niet compatible met de originele AVR of één van de ARM kernen.
Lego heeft recentelijk firmware broncode bestanden uitgebracht voor de 32-bit AVR's om te helpen in de ontwikkeling van de open-source alternatieven van zijn populaire robotica NXT besturingssysteem.