

Van PIC tot Pi

Microcontrollers en op microcontrollers gebaseerde platformen zijn belangrijke bouwstenen in de radioamateurwereld. Een overzicht en enkele voorbeelden van toepassingen voor niet-ingewijden tref je hierbij aan. Het overzicht is beperkt en bedoeld als kennismaking met het onderwerp. Dit artikel probeert je over de drempel te tillen en aan te zetten om er mee te gaan experimenteren. Ik garandeer je dat er dan een nieuwe wereld voor je opengaat!

Inleiding

Microcontrollers en daarop gebaseerde platformen worden voor het gemak in dit artikel 'microcontrollers' genoemd.

Microcontrollers worden ingezet om 'iets' te besturen. Het wasprogramma in de wasmachine bijvoorbeeld laat de trommel links en rechts draaien en centrifugeren, laat water in de trommel lopen en pompt het water weg, enzovoorts. In transceivers worden microcontrollers gebruikt om de standen van schakelaars te bepalen, de lcd van informatie te voorzien, oscillatoren om te schakelen en dergelijke functies. De huidige generatie ARDF-zendertjes (voor vossenjachten) wordt trouwens ook door microcontrollers bestuurd; denk aan de callgever en de uitzendperioden. Hoe het besturen gebeurt, ligt vast in het programma dat in de controller zit.

Een klassieke eerste stap om het programmeren van zo'n controller onder de knie te krijgen is een led te laten knipperen. Als dat lukt ben je vertrokken. Microcontrollers zijn relatief goedkoop en erg flexibel, en daardoor bijzonder geschikt voor de experimenterende amateur. Hoe programmeer je een controller? Allereerst moet je natuurlijk weten wat je de microcontroller wilt laten doen. We nemen het klassieke voorbeeld om een led te laten knipperen. Dat is te vertalen in:

```
Begin:   doe de led aan
         wacht 1 seconde
         doe de led uit
         wacht 1 seconde
         ga naar Begin
```

Het bovenstaande programma is wel heel eenvoudig, maar belangrijk is op te merken dat de acties na elkaar plaatsvinden: doe eerst dit en daarna dat. Het programma is een lijst van sequentiële acties. Er moet dus een uitgang van de controller om de tweede beurtelings laag en hoog worden. In de documentatie van de controller wordt uitgezocht welke aansluitingen als uitgang kunnen dienen. Hoe de bewuste uitgang waarop de led aangesloten is reageert, bepaalt het programma in de microcontroller.

Microcontrollerprogramma's zijn op verschillende manieren te schrijven. Voor eenvoudige programma's werd, en wordt soms nog, assembleertaal gebruikt. Dat is een taal die dicht bij de machinetaal van de microcontroller staat. Machinetaal bestaat uit opdrachten in de vorm van een serie bytes of woorden, die direct door de controller begrepen worden, maar die voor ons erg onoverzichtelijk is. Assembleertaal staat vrijwel een op een ten opzichte van machinetaal, terwijl het wel begrijpelijk is voor de mens. Met behulp van een assembler (convertoorprogramma) wordt een programma geschreven

Hans Wagemans ON4CDU
on4cdu@uba.be

Op de *Electron* internetsite <http://electron.veron.nl/> onder 'Links' staat een lijst met de in dit artikel genoemde internetlinks. Zo kun je bij het lezen de link aanklikken en de referentie bekijken. Er zijn daar bovendien nog een aantal extra links vermeld.

in de assembleertaal omgezet in machinetaal. De assembleertaal is microcontrollerafhankelijk. Daarom moet men nogal wat van het inwendige van de microcontroller weten om zo een programma te kunnen schrijven. Marc Simons PE1RRT heeft ons uitgebreid in *Electron* geleerd hoe dat moet. Bekijk dit programma maar eens op [1].

Nu is het schrijven van een programma in assembleertaal (met instructies als `bsf`, `movlw`, `decfsz`..., zie [1]) voor wat ingewikkeldere besturingen en berekeningen nogal arbeidsintensief. Verder moeten we om zo'n programma in assembleertaal te kunnen schrijven ook wat inzicht hebben in hoe de microcontroller inwendig werkt. Vaak is het echter voor de gebruiker niet nodig te weten hoe de microcontroller de instructies intern uitvoert. Zo zijn we er in ons ledvoorbeeld 'alleen' in geïnteresseerd dat de gekozen uitgang van de controller om de seconde hoog en laag gaat.

Bij gebruik van hogere programmeertalen hebben we vaak weinig kennis nodig van het inwendige van de controller. Door gebruik te maken van de hogere programmeertaal C ziet ons ledknipperprogramma er zo uit [2]:

```
void loop()           // oneindige loop
{
    digitalWrite(led, HIGH); // laat de led branden
    delay(1000);         // wacht 1000 milliseconden (de led
                        // brandt nu 1 seconde)

    digitalWrite(led, LOW);  // zet de led uit
    delay(1000);           // wacht 1000 milliseconden (de led
                        // is nu 1 seconde uit)
}
```

Daarbij is de tekst na '/' commentaar voor jezelf om de bedoeling van het statement te verklaren; het is voor het programma niet nodig. Dit programma is natuurlijk veel eenvoudiger te schrijven en te begrijpen dan het programma in assembleertaal. We hebben wel een compiler (vertaalprogramma) nodig die dit programma in C omzet naar de machinetaal voor onze microcontroller.

Nu volgt een kort overzicht over de ontwikkelingen op microcontroller gebied. Bijna alle nieuwe (amateur)apparatuur maakt gebruik van deze wonderdoosjes.

De PIC microcontroller

PIC, een afkorting van Peripheral Interface Controller, is een familie van microcontrollers gefabriceerd door de firma Microchip. Marc PE1RRT heeft in *Electron* van januari 2001 aandacht aan de PIC 16F84 geschonken [1] en hij legde ook uit hoe je die programmeert. We zijn inmiddels meer dan tien jaar verder, maar de basis die in dat artikel beschreven staat geldt nog steeds.

De PIC microcontroller is leverbaar in vele types en groottes, van 8 tot 40-pins DIL en SMD-behuizingen. Welke types er bestaan en de specificaties daarvan vallen buiten het bestek van dit artikel; er zijn beslist amateurs op de afdelingsavond te vinden die je daarmee kunnen helpen. Verder geeft de internetpagina Microchip Design Support veel informatie over de programmeeromgeving voor je desk-

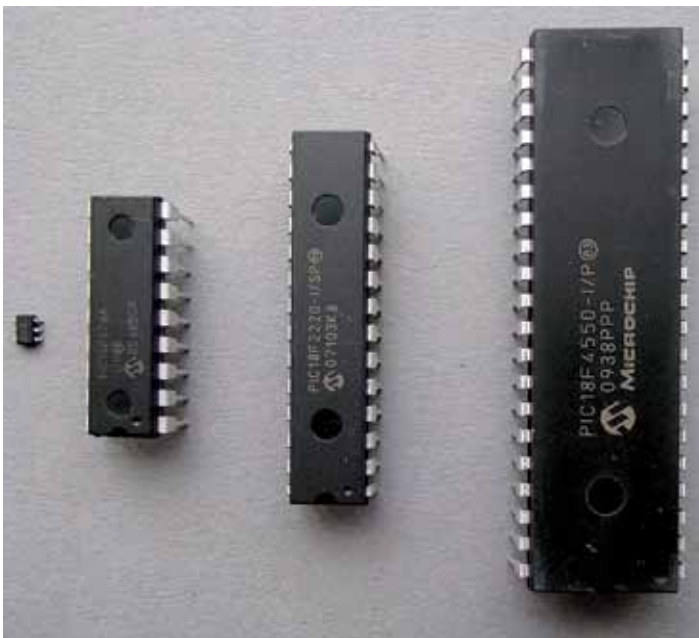


Foto 1 Enkele types PIC microcontrollers: van links naar rechts de 10F200, 16F628A, 18F2220 en 18F4550

laptopcomputer, nodig om zo'n controller te kunnen programmeren. De leercurve om zelfstandig in MPLAB (dit is de programmeeromgeving voor de PIC's) een programma te schrijven en te laden in de PIC controller is tamelijk steil, en dus kan wat hulp van een meer ervaren amateur wel wenselijk zijn. Maar ter geruststelling: mij is het gelukt dankzij de uitstekende artikelen van PE1RRT in *Electron*. MPLAB op de computer zet assembleertaal om in de bitcode die in de microcontroller geladen moet worden. Dat laden in de PIC gebeurt met een PIC-programmer (op eBay staan er veel te koop), of bij bepaalde PIC's via een usb-aansluiting.

Het schrijven van een programma in assembleertaal voor wat ingewikkeldere functies, bijvoorbeeld om uit gps-data een QTH-locator te berekenen en die dan op een display te zetten, is nogal arbeidsintensief. Het gebruik van een hogere programmeertaal zoals C, Basic of JAL is dan aan te raden.

Tot slot van het PIC-overzicht een aantal voorbeelden:

- Eenvoudige projecten om met de PIC te leren omgaan, zie [3]
- Eenvoudige projecten met PIC's in amateurtoepassingen vind je op [4]
- De (Engelstalige) introductie door de Amerikaanse QRP-club, op [5]
- Een aan gps gekoppelde 10 MHz-standaard uitgerust met een PIC is gepubliceerd op [6]

PIC controllers worden ook vaak gebruikt voor de programmering van andere IC's, zoals PLL's, DDS'en, coders en dergelijke. De PIC stuurt het andere IC aan, zodat bijvoorbeeld bij een PLL de juiste beginwaarden, zoals deeltallen, worden ingesteld, of zodat bij een

hoek-encoder de hoekinformatie naar buiten wordt gebracht. Ter illustratie hiervan vind je op [7] een PLL met een LMX2306 waarin een PIC 16F628A als besturing wordt gebruikt.

Dit waren slechts enkele voorbeelden van de toepassing van PIC's. Op het internet zijn er zeer veel meer te vinden. We zien bij de huidige ontwikkelingen dat de PIC wordt verdrongen door platformen als Arduino en Raspberry Pi.

Arduino

De Arduino is een relatief jonge loot aan de stam. Het is een ATMEEL microcontroller op een printje met enkele interface-IC's en wat basisprogrammatuur. Op de website van Arduino [8] staat: "Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments." Dat is stevige taal, maar ze maken dat ook waar. De kracht van de Arduino is dat er zeer gebruikersvriendelijke interfacesoftware voor je desk- of laptopcomputer beschikbaar is, waardoor



Foto 3 Een PLL-schakeling met een LMX2306 bestuurd door een PIC 16F628A



Foto 2 Door de auteur gebouwde gps-gestuurde 10 MHz-standaard van VE2ZAZ met PIC 18F2220



Foto 4 De Arduinoprint

via een usb-poort van je computer met de Arduino wordt gecommuniceerd. Programma's voor de Arduino worden in C geschreven. De vertaalslag naar machinetaal gebeurt door de software op je computer. Verder zijn er bibliotheken van functies beschikbaar, en niet in de laatste plaats is een Arduino goedkoop; voor maar iets meer dan twintig euro heb je er een.

Al deze zaken maken dat de Arduino zeer geschikt is voor een eerste kennismaking met microcontrollertoepassingen. Dat is ook middelbare en hogere scholen en zelfs universiteiten niet ontgaan. Zij gebruiken de Arduino vaak om studenten kennis te laten maken met de mogelijkheden van microcontrollers. Handleidingen, bijvoorbeeld [9] en [10], of interessante filmpjes op YouTube helpen je op weg. Alles bijeen is de Arduino een goede en goedkope opstap om met microcontrollers vertrouwd te worden. Ik stel me zo voor dat je er een weekendproject van maakt om een Arduino te kopen, de software op je computer te installeren en een eerste projectje te maken om een led te laten knipperen [2]. Als je daarin slaagt, zal het niet te moeilijk zijn ook morsetekens uit een luidsprekertje te toveren [11]. Ik garandeer je dat je daarna nog veel andere experimenten zult doen. Je kunt bijvoorbeeld de potmeter van de antennerotor uitlezen, want de Arduino kan ook analoge signalen verwerken.

Conclusie: de Arduino is een concept met een lage drempel, goedkoop in aanschaf en zeer geschikt om mee te beginnen. Iets voor een afdelingsactiviteit?

Raspberry Pi

Vrij recent is er een complete computer ter grootte van een creditkaart op de markt verschenen: de Raspberry Pi [12]. De Raspberry Pi is een Linux (Debian versie) machine en heeft onder andere twee usb-poorten, een HDMI-interface en een ethernet aansluiting. Het operating system wordt op een SD-kaartje gezet. Verder heb je nog een voeding nodig. Hoe het geheel aan de praat te krijgen is, wordt goed beschreven op internet en is niet echt ingewikkeld. Maar het is geen beginnersproject, en wat basiskennis van Unix is echt een pre. Waarom is de Raspberry interessant? Omdat het een complete computer is gebruik je de Raspberry om programma's te ontwikkelen, er is immers een aansluiting voor een computerscherm, en daarna kun je het programma ook op je Raspberry uitvoeren. Verder zijn hogere programmeertalen zoals Perl, Python en C++ gratis beschikbaar, waardoor het ontwikkelen van software een heel stuk eenvoudiger wordt. En niet te versmaden is dat als output een gewoon videoscherm (Xwindows) gebruikt kan worden, zodat er geen moeilijke denkoefeningen nodig zijn om de output op een 16x2 lcd te schrijven. Voor de hogere programmeertalen zijn veel documenten en ook indrukwekkende bibliotheken beschikbaar, zoals de Perl-bibliotheek of die van Python. De Raspberry is pas vrij recent leverbaar, maar in veel tijdschriften en op internet worden nu al toepassingen beschreven, bijvoorbeeld het gebruik van een Raspberry als SDR server. In *Electron* van april stond dat PA3WEG er een testbeeldgenerator voor ATV mee gemaakt heeft, en dat PB0NER een hele bibliotheek van functies in Python klaar



Foto 5 Een Raspberry Pi samen met een DVB-T usb-stick ter vergelijking

heeft. CQ-DL publiceerde in het aprilnummer APRS-toepassingen met de Pi. Zelf test ik momenteel een programma voor uitlezing van de antennepositie in azimut en elevatie, geschreven in Perl. De prijs van een Raspberry Pi is ongeveer 35 euro, maar er is ook nog een SD-kaartje bij nodig. Een scherm, een videokabel, een usb-keyboard en een muis heb je mogelijk al in huis. We gaan beslist nog veel van de Raspberry Pi horen.

Naschrift redactie *Electron*: uw radioamateurtoepassingen met microcontrollers plaatsen we graag in *Electron*!

Internetlinks

- [1] http://www.jancorver.org/info/microchip/rrt/electron/pic_1.pdf
- [2] <http://arduino.cc/en/Tutorial/Blink>
- [3] <http://www.pic-tronics.com/Basic-PIC-Projects.php>
- [4] http://www.qsl.net/dl4yh/#pic_projects
- [5] <http://www.amgrp.org/elmer160/lessons/index.html>
- [6] http://ve2zaz.net/GPS_Std/GPS_Std.htm
- [7] <http://on4cdu.net/publications/MUD2007.pdf>
- [8] <http://www.arduino.cc/>
- [9] http://www.kompanje.nl/arduino/Arduino%20manual%201_0%20NL.pdf
- [10] <http://www.mieni.nl/wp-content/uploads/2011/11/Arduino-UNO-2011b-handleiding.pdf>
- [11] http://liudr.wordpress.com/libraries/phi_morse/
- [12] <http://www.raspberrypi.org/>

Is uw
roepnaam
gewijzigd?

Heeft u een speciale roepnaam
aangevraagd?

Geef het door aan het DQB!