

Ontwikkelomgeving in Ubuntu en Windows voor de AT90CAN32 controller

Algemeen

De AT90CAN32 kan worden geprogrammeerd in C en in assembler. Voor het auto-onderwijs wordt (in eerste instantie?) gekozen voor de assembleertaal omdat deze het dichtste staat bij de eigenlijke controller, de hardware. Men heeft voor het programmeren nodig:

- een tekstverwerker
- een assembleerprogramma (eventueel met linker)
- een download programma

Wanneer al deze programma's verwerkt zijn tot 1 programma dan spreekt men wel van een ontwikkelomgeving of IDE (Integrated Development Environment)

Wordt er in C geprogrammeerd dan wordt uiteindelijk door de C-compiler machinetaal gegenereerd, zodat C-compiler en assemblers vaak geïntegreerd zijn. AVR-GCC moet dan wel geïnstalleerd zijn.

Onder Windows is AVR-STUDIO 4 beschikbaar, een schitterende programmeeromgeving (IDE) waarin zowel in C als in assembler kan worden geprogrammeerd. Voor het programmeren in C moet AVR-GCC wel geïnstalleerd zijn. AVR-STUDIO is door de fabrikant Atmel ontwikkeld en is gratis te downloaden. Het is echter (zover bekend) geen vrije of open software.

Onder Linux (Ubuntu) ligt het allemaal wat ingewikkelder. Er zijn (zover ons bekend) een tweetal IDE's beschikbaar:

- Eclipse C/C++ (Dit is een andere versie als de java-programmeeromgeving)
- Kontrollerlab (voor de KDE)

Eclipse C/C++ heeft na installatie een plugin nodig: AVR Eclipse.
We hebben nog (weinig) geen ervaring kunnen opdoen.

Ervaringen met Kontrollerlab zijn toe zeer beperkt. Het ziet er echter wel fraai uit lijkt gemakkelijk in de omgang en lijkt in het gebruik veel op AVR-STUDIO4.

Willen we of moeten we zonder IDE werken dan vallen we terug op de tekstverwerker, de assembler en het downloadprogramma. Bij Linux (Ubuntu) lijkt dat op dit moment noodzaak.

Het programmeren van de AT90CAN32 onder Linux Ubuntu zonder IDE

Omdat Kontrollerlab in wezen niets anders is als een schil om de bestaande ontwikkelprogramma's is enige kennis hieromtrent altijd nuttig. De ontwikkelprogramma's onder Linux zijn:

- kedit of gedit als tekstverwerker
- avr-gcc (bestaande uit: avr-as, avr-ld en avr-objcopy, zijnde de assembler en linker)
- avrdude (ISP download-programma)

De keuze van de tekstverwerker ligt voor de hand. gedit of kedit is standaard aanwezig. (Bij 'tekstkeuring' kan bij 'bronnen' voor C worden gekozen (gedit).

De assembler

gcc is de standaard C-compiler voor Linux. Bij Ubuntu wordt deze compiler automatisch mee-geïnstalleerd. Dit valt te controleren door vanuit de terminal `-$ gcc` in te typen. Een `-$ man gcc` geeft meer informatie. Mocht het programma niet geïnstalleerd zijn dan kan installatie plaatsvinden door `..# apt-get install gcc`

(Je moet dan wel met 'sudo su' jezelf super-user gemaakt hebben en een internetverbinding hebben).

Extra

We zouden nu een C-programma kunnen schrijven. Bijvoorbeeld.

Tik met de tekstverwerker het volgende in en schrijf de file weg als `program1.c`

```
#include <stdio>
int main (void)
{
printf ("hello World");
return 0;
}
```

ga nu vanuit de terminal naar de directory waarin het bestand is opgeslagen en tik in:

```
....$ gcc -o program1 program1.c
```

Er wordt dan een uitvoerbaarbestand `program1` aangemaakt. Probeer dit uit door:

```
...$ ./program1
```

'Hello World' verschijnt dan op het scherm

Nu is er een uitbreiding gemaakt op gcc om gcc geschikt te maken voor de AVR-controllers. Immers een C-programma moet uiteindelijk machinetaal programmeren. Voor AVR-controllers moet dat dus niet de machinetaal worden van de PC (Intel) maar van de AVR-controllers. Deze uitbreiding staat bekend onder `gcc-avr`.

Dit programma staat bekend onder een toolchain omdat het in feite bestaat uit een aantal kleine programma's o.a. een assembler, linker en de C-compiler. `gcc-avr` moet op Ubuntu worden geïnstalleerd. Dit kan gebeuren (nadat je jezelf met `sudo su` superuser hebt gemaakt met `..# apt-get install gcc-avr`. Het programma installeert zich zelf dan automatisch. Om te controleren of dit gelukt is kun je intikken: `avr-gcc` (net omgekeerd dus). Zowel de assembler `avr-as` als de C-compiler `avr-gcc` is nu geïnstalleerd.

De avr-assembler (avr-as)

Informatie kan worden verkregen door: `$ info avr-as` of `$ man avr-as`. Tikken we in:

`avr-as -- help` dan zien we alle ook alle controllers die worden ondersteund. We vinden hier ook tussen de AT90CAN32.

We schrijven nu een programma met de tekstverwerker genaamd test.s Hier volgt de code:

```

/* programma naam: test.s*/
/* testprogramma voor de AT90CAN32*/
/* het programma laat de leds knipperen*/

/* assembleren via gcc-avr, te gebruiken avr-as, avr-ld en avr-objcopy */
/* avr-as -mmcu=at90can32 -al -o test.o test.s*/ /*list file naar scherm*/
/* avr-as -mmcu=at90can32 -al=test.lst -o test.o test.s*/ /*listfile naar file*/
/* avr-ld -o test.elf test.o */ /*linken*/
/* avr-objcopy --output-target=ihex test.elf test.ihex*/ /* maak er een (i)hexfile van*/

/* nu downloaden met: */
/* ..# avrdude -p c32 -c avrispmkII -P usb: xx -e -U flash:w:test.ihex */

.EQU PORTA,0x02
.EQU DDRA,0x01
.EQU PORTC,0x08
.EQU DDRC,0x07
.EQU PINC,0x06
.EQU SPH,0x3E
.EQU SPL,0x3D
.EQU RAMEND,0x08FF

RJMP RESET

/*INITIALISATIE*/

RESET:
    LDI R16,0xFF          ;A op output
    OUT DDRA,R16

    LDI R16,0xFF          ; C inputpoort
    OUT PORTC,R16        ; wordt hier niet gebruikt
    LDI R17,0x00
    OUT DDRC,R17
    NOP

                                ;stackpointer
    LDI R16, 0x08
    OUT SPH, R16
    LDI R16, 0xFF
    OUT SPL,R16

LOOP:
    LDI R16,0x00          ;leds uit
    OUT PORTA,R16
    RCALL WAITABIT

    LDI R16,0xFF          ;leds aan
    OUT PORTA,R16
    RCALL WAITABIT

    RJMP LOOP

WAITABIT:
    LDI R17,0xFF
WAITOUTERLOOP:
    LDI R18,0xFF
WAITINNERLOOP:
    SUBI R18,0x01
    BRNE WAITINNERLOOP
    SUBI R17,0x01
    BRNE WAITOUTERLOOP
    RET

```

In de bovenste regels wordt aangegeven op welke wijze we van de test.s file (aangemaakt door de tekstverwerker komen tot de downloadle test.hex file. Dit gebeurt hier in drie stappen. Ondertussen wordt ook nog een list file aangemaakt. Na dit 3-traps proces zijn de volgende files aangemaakt:

```
test.s
test.o
test.lst
test.elf
test.ihex
```

Van deze files kunnen de .s, de .lst en de .ihex met de tekstverwerker zichtbaar worden gemaakt. De (i)hex file is de file waar het allemaal omgaat. Deze file kan worden gedownload.

(Opm: de hex file die door AVR STUDIO4 wordt gemaakt bevat een extra (eerste) regel. De reden waarom is nog niet duidelijk)

We gebruiken hiervoor het AVRDUDE-programma. Met ...# apt-get install avrdude komt het programma op de computer te staan.

Sluit nu de AT90CAN32 controller (met ISP-interface) aan op de usb-poort van de (linux) computer

Wanneer we even wachten kunnen we met ~\$ lsusb vanaf de command line zien of de AVR-ISP interface is gevonden.

```
desktop:~$ lsusb
Bus 001 Device 004: ID 03eb:2104 Atmel Corp. AVR ISP mkII
Bus 001 Device 003: ID 046d:c00e Logitech, Inc. M-BJ69 Optical Wheel Mouse
```

Wanneer we nu van de command line het volgende intikken (maak jezelf eerst superuser) :

```
/home/ep# avrdude -p c32 -c avrispmkII -P usb: xx
```

dan wordt er geantwoord:

```
avrdude: AVR device initialized and ready to accept instructions
```

```
Reading | ##### | 100% 0.01s
```

```
avrdude: Device signature = 0x1e9581
```

```
avrdude: safemode: Fuses OK
```

```
avrdude done. Thank you.
```

Het downloaden kan nu gebeuren door de volgende regel in te tikken (superuser!):

```
...# avrdude -v -p c32 -c avrispmkII -B0.5 -P usb: xx -e -U flash:w:test1.ihex
```

Om dit proces te bespoedigen zijn er script files (.sh) aangemaakt.

de betekenis hiervan is als het volgt:

- v sta controle op de file toe
- p hierna volgt de code van de controller: hier c32
- c hierna volgt het type download interface: hier avrispmkII
- B0.5 specificeert de klok periode voor de ISP interface
- P geeft de usb poort aan: hier usb: xx
- e wist het flash geheugen
- U hierna volgt de handeling: flash:w:test1.ihex. Schijft (w) de file naar het flashgeheugen

Avrdude kan veel: In het volgende voorbeeld leest avrdude het hfuse register uit.

```
..# avrdude -p c32 -c avrispmkII -P usb: xx -U hfuse:r:hfuse:h
```

Betekenis na -U Lees (read) hfuse uit en schrijf de hexadecimale waarde (h) naar een file met de naam hfuse.

Het is wel belangrijk dat de fuses goed staan ingesteld. Zie onder 'Fuses'. De 'w' optie programmeert de fuses.

Kontrollerlab*

Het kontrollerlab programma bedoelt om te draaien onder kubuntu is de schil om:

- een tekstverwerker (kedit)
- de gcc-avr toolchain (de assembler, linker, c-compiler e.d).
- avrdude voor het downloaden van het geassembleerde programma naar de AT90CAN32 print

Installatie op een (K)Ubuntu-computer via synaptic: Volgorde:

- gcc-avr
- avrdude
- KontrollerLab (via net)

*Op dit moment zijn we er nog niet in geslaagd om kontrollerlab goed te installeren.