Getting started with the AVR toolchain

Julien TOUS

October 7, 2008

1 introduction

This document is a memo on how to use the gnu AVR toolchain.

2 Installation

For Debian and derivative, just install the following packages: gcc-avr, binutils-avr, gdb-avr, avarice, avr-libc.

`sudo apt-get install gcc-avr binutils-avr gdb-avr avarice avr-libc`

If you wish to use avr6 family (ie atmega2561...) you might want to use patched versions. Sid and Lenny packages are already patched. Patched Ubuntu packages can be found at http://think.objectweb.org/toolchains.html.

3 Compilation

You need to tell avr-gcc which device you want to compile for. Use the -mmcu option.

`avr-gcc -g -c -mmcu=atmega128 riri.c`

4 Link

You can link object files using avr-gcc. All include directory regarding the C runtime (crt0), the libc, and libgcc are then automatically handled.

`avr-gcc -mmcu=atmega128 riri.o fifi.o loulou.o -o DonaldNephews.elf`

Or you can still use avr-ld if you don’t want to use gcc’s defaults.

`avr-ld -m avr5 riri.o loulou.o fifi.o mycert.o -T myldscript`
5  Putting this together

Put those informations and more in a Makefile with the lines :

```bash
CC = avr-gcc
LD = avr-gcc
CFLAGS = -c -g -Wall -O0 -mmcu=atmega2561
LDFLAGS = -mmcu=atmega2561
```

6  Loading

You can load the binary on the target by using avarice.

```bash
avarice -j /dev/ttyS1 -2 --erase --program --file DonaldNephews.elf localhost:6423 --detach
```

Where options :

- `-j /dev/ttyS1`
tells to search for the JTAG on serial port 1. You could use 
  "-j /dev/ttyUSB1" to connect to an usb to serial adapter or 
  "-j usb" to connect to usb (only mkII).

- `-2`
tells you’re using a JTAG mkII (-1 for mkI).

- `--erase --program --file DonaldNephews.elf`
tells to program on the file DonaldNephews.elf on the target and to erase previous program if there was one.

- `localhost:6423`
tells to open a TCP socket on port 6423 on localhost for debugging purpose.

7  Debuging

After starting avarice, you can start avr-gdb :

```bash
$ avr-gdb DonaldNephews.elf
```

Set some usefull settings :

```console
(gdb) set remote addresssize 32
(gdb) set remote Z-packet enable
(gdb) set remote hardware-breakpoint-limit 3
(gdb) set remote hardware-watchpoint-limit 2
```
Connect to the previously created TCP socket:

(gdb) target remote localhost:6423

And debug:

- (gdb) run
  To start the execution

- Ctrl-c
  To pause execution.

- (gdb) hb riri.c:20
  To set a breakpoint on line 20 off file riri.c.

- (gdb) hb getMyUncle
  To set a breakpoint at the beginning of getMyUncle function.

- (gdb) d 1
  To remove breakpoint 1.

- (gdb) p myBrothers
  To show the value of myBrothers variable.

- (gdb) info reg
  To show the status of all registers.

- ...


8 Putting this together

You can install the packet avr-utils from http://think.objectweb.org/toolchains.html and set the serial port you want to use in /etc/default/avr-utils. Running the script ice-gdb will load your kernel start and attach gdb and set its parameters.

$ ice-gdb DonaldNephews.elf