Connecting up the Arm Evaluator 7t and Downloading Code

Nigel Roles

Vita Nuova Holdings Limited

10th November 2000

Introduction

This is just a quick note on how to download code to the Evaluator and run it.

Health Warning

I *think* the interrupt and exception handling is OK; it's just that I had problems with supporting two serial ports, interrupt driven at the same time. You have been warned.

Connecting it up

The board has Arm's standard Angel debugger monitor on board. You can read all about this in the documentation. Whether you go to the trouble of installing the Windows tools is up to you, but do read how to operate the monitor.

Anyhow, the board has two serial ports. The one nearest the LEDs is the debug and download port. The other is the console port when running Inferno.

You only get one cable, so beg borrow or steal another cable. These are straight-through 9 way to 9 way male to female cables. I bought a Belkin one from Staples at some horrid price.

Connect both up to the back of your Plan 9 machine, and start terminal emulators on both. For the debug and download you need to run a specially modified version of `vt(1)` which responds to the peculiar answer back sequence that the Arm monitor has decided is appropriate for it to sense baud rate. As it happens, I have modified the version on `doppio`. It should be possible via monitor commands to set the default baud rate of the board to whatever you like. I lost patience after about a day, and hacked `vt(1)` instead. It was easier. So, the board will auto sense baud rate. The highest baud rate which is actually valid and hence will talk to Plan 9 is 57600. So the commands to configure the download window are

```
term% cd /usr/inferno/os/ks32
term% vt
term% con -R -b 57600 /dev/eia0
```

You can choose other speeds, but 57600 is the maximum. For the console port, `vt(1)` is not necessary. The baud rate is fixed in `main.c` at 57600, so the same `con` command is appropriate.

Now plug in the power to the board. You should see the LEDs flick for bit, and then the 7 segment one shows 11. The download window should show

```
Arm Evaluator7T Boot Monitor PreRelease 1.00
Boot:
```

Pressing the reset button (nearer of the two buttons to the CPU), should cause the same message to be displayed.

Building the code

`mk(1)` in directory `/usr/inferno/os/ks32` should build the code. You may need to do

```
bind -b /usr/inferno/Plan9/386/bin /bin
```

to get access to the compilers and tools. The
makes an executable, converts it to the right format, and `uuencode`'s it to create `ievaluator7t.txt`. Note that as shipped, the configuration file (`evaluator7t`) includes the file

`/usr/inferno/usr/nigel/cb.dis` as the main application. This is John Powers' Crackerbarrel program which I used as a benchmark.

Downloading the code

To download to the Evaluator type the command

```
Boot: download
Ready to download. Use 'transmit' option on terminal emulator to download file.
```

break back to the `con(1)` command prompt with Control-\ and type the following command

```
>>> !cat ievaluator7t.txt
```

As a shorthand, you may instead type

```
>>> !squirt
```

The red LED should light indicating download. After a couple of minutes the light goes out and you see

```
Loaded file ievaluator7t.aif at address 00008000, size=254200
Boot:
```

You will type

```
Boot: gos 8080
```

to run the code at address 0x8080. This should cause Inferno to boot and write stuff on the console port (i.e. the other window).

Useful Stuff

There is some useful debugging code in `archevaluator7t.c` and other places.

`setled7ascii()`
: This function takes a single character, and attempts to put it on the LED. Clearly you won't get M's or N's. Read the source.

`iprint()`
: Prints on the debug and download port.