AvrJtagIce

MKII :

For the impatient :

```
export AVARICE_ARGS='--mkII   --jtag usb'
uisp -dprog=mib510 -dpart=ATmega128  -dserial=/dev/ttyS0 --wr_fuse_h=19
ice-gdb  build/micaz/main.exe
```

you need:

- avarice cvs version
- avr-gdb
- build the program to debug with

```
make micaz debug
```

Or add -ggdb -g -O0 to the CFLAGS in the makefile

- some patience

Configure stuff:

- Tell how to communicate with jtag ice
  - ♦ export AVARICE_ARGS='--mkII --jtag usb'
- Configure fuses to enable jtag debugging. With micaz, mib510 on first serial port :
  - ♦ uisp -dprog=mib510 -dpart=ATmega128 -dserial=/dev/ttyS0 --wr_fuse_h=19

To test if it works :

run : avarice --mkII --jtag usb

you should get something like :

```
JTAG config starting.
Found a device: JTAGICEmkII
Serial number:  00:b0:00:00:08:61
Reported JTAG device ID: 0x9702
Configured for device ID: 0x9702 atmega128
JTAG config complete.
```

To debug:

- ice-gdb build/micaz/main.exe

you should get something like :

```
AVaRICE version 2.5.20061008, Nov 20 2006 16:30:39

Defaulting JTAG bitrate to 1 MHz. Make sure that the target
```

```
frequency is at least 4 MHz or you will likely encounter failures
controlling the target.


JTAG config starting.
Found a device: JTAGICEmkII
Serial number:  00:b0:00:00:08:61
Reported JTAG device ID: 0x9702
Configured for device ID: 0x9702 atmega128
JTAG config complete.
Erasing program memory.
Erase complete.
Preparing the target device for On Chip Debugging.

Disabling lock bits:
  LockBits? -> 0xff

Enabling on-chip debugging:
 Extended Fuse byte -> 0xff
     High Fuse byte -> 0x19
      Low Fuse byte -> 0xff
Downloading FLASH image to
target..............................................


Download complete.
Waiting for connection on port 6423.
GNU gdb 6.5
Copyright (C) 2006 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=avr"...
Connection opened by host 127.0.0.1, port 46688.
0x00000000 in __vectors ()

(gdb)
```

Then use gdb as usual ...

Problems/bugs :

- Reply contains invalid hex digit 79

(then Remote communication error: Connection reset by peer. and putpkt: write failed: Broken pipe.) add :

```
set remotetimeout 60
```

in your gdb init script or the default one in :

```
/usr/share/avarice/gdb-avarice-script
```

- No configuration available for device ID: ffff

the exact error is :

```
AVaRICE version 2.5.20061008, Nov 20 2006 16:30:39

Defaulting JTAG bitrate to 1 MHz. Make sure that the target
frequency is at least 4 MHz or you will likely encounter failures
controlling the target.

JTAG config starting.
Found a device: JTAGICEmkII
Serial number:  00:b0:00:00:08:61
Reported JTAG device ID: 0xFFFF
No configuration available for device ID: ffff

GNU gdb 6.5
Copyright (C) 2006 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=avr"...
/usr/share/avarice/gdb-avarice-script:25: Error in sourced command file:
localhost:6423: Connection refused.
```

you forgot to set or overwrite the fuse bits with uisp ... run again uisp

MKI :

don't have one so can't test but here are some details :
http://www.tinyos.net/tinyos-1.x/doc/nesc/debugging.html