

"Wiggling" Using arm-elf-gdb/insight-gdb

A small "step-by-step"-guide by Martin Thomas, Kaiserslautern, Germany

Intro

There have been a lot of questions about using Wiggler-type JTAG-adaptors with LPC2000 ARM controllers. I hope this guide will at least help with the first steps. The Olimex ARM-JTAG ("Wiggler-clone") has been used but there are other schematics available in the internet which are simpler and seem to work more reliably. See the LPC-yahoo-groups file-archive for schematics.

Hard- and Software

- WinARM 4/2005 (arm-elf-gcc 4.0.0, insight-gdb 5.2.1)
- Macraigor's ocdremote 2.11 (ocdremote.exe needs the cygwin1.dll, not need to install the complete cygwin-environment, just download the cygwin1.dll and place it into the same directory as ocdremote.exe)
- Olimex "Wiggler" ARM-JTAG with additional internal connection from pin 8 to 15 on the "PC-side"
- Olimex Board LPC-P2129 (Philips LPC2129 ARM7DTMI-S) with DEBUG jumper set
- PC with Windows 2000 OS, CPU: PII-400, Printerport set to EPP in BIOS

But method should be the same for other version of the gnu-tools, other Wiggler-clones (and of cause the original) and different target CPUs.

Example Software

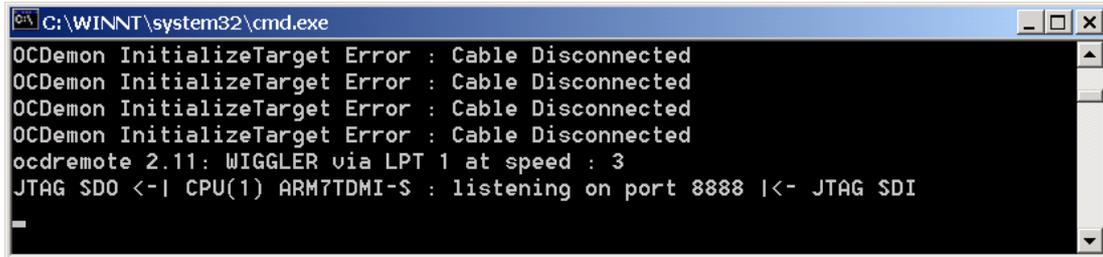
The example program is a simple "LED-Blinky" which turns the two LEDs on a Olimex LPC-P2129-Boad (Philips LPC2129) on and off. The source is available in the zip-archive from http://www.siwawi.arubi.uni-kl.de/avr_projects/arm_projects/. The important makefile-settings for this test are:

- RUN_MODE=RAM_RUN : Code must be compiled "for RAM"
- OPT = 0 : to avoid compiler optimisation
- DEBUG = stabs : dwarf-2 seems to be not supported by the old gdb version

"Step-by-Step"

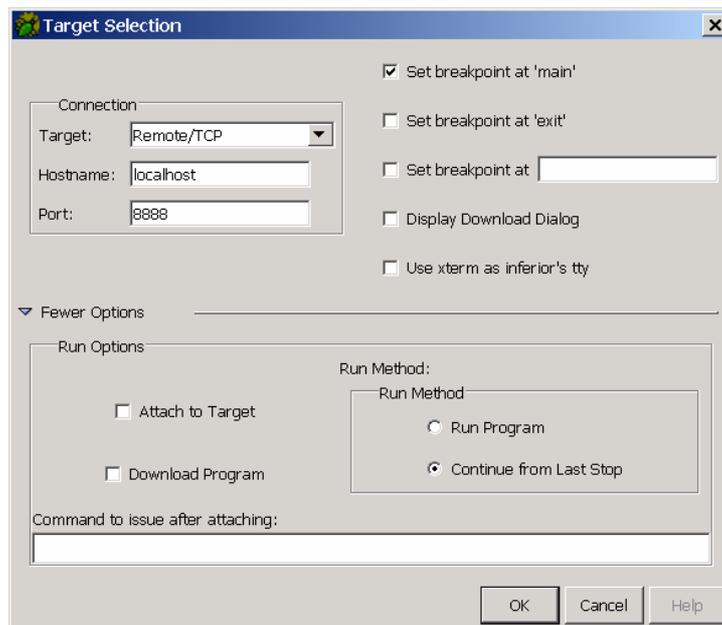
1. Compile and link the source (make all)
2. Configure settings in the script (batch-file) start_wiggler.cmd according to your system. The script is included in the source archive.

3. Start start_wiggler.cmd. This will try to establish a connection to the target CPU. If it fails verify the settings. If it still fails lower the speed by increasing the SPEED-value (maximum: 8 = slowest connection). Around 30 retries were "normal" on the test-system using an Olimex "ARM-JTAG" (sometimes it connects on the first try...). The maximum speed I have reached has been 3. Other Wiggler-clones are know to work at speed-setting 1, but so far I have no other adaptor here for tests. The final output has to be like this:

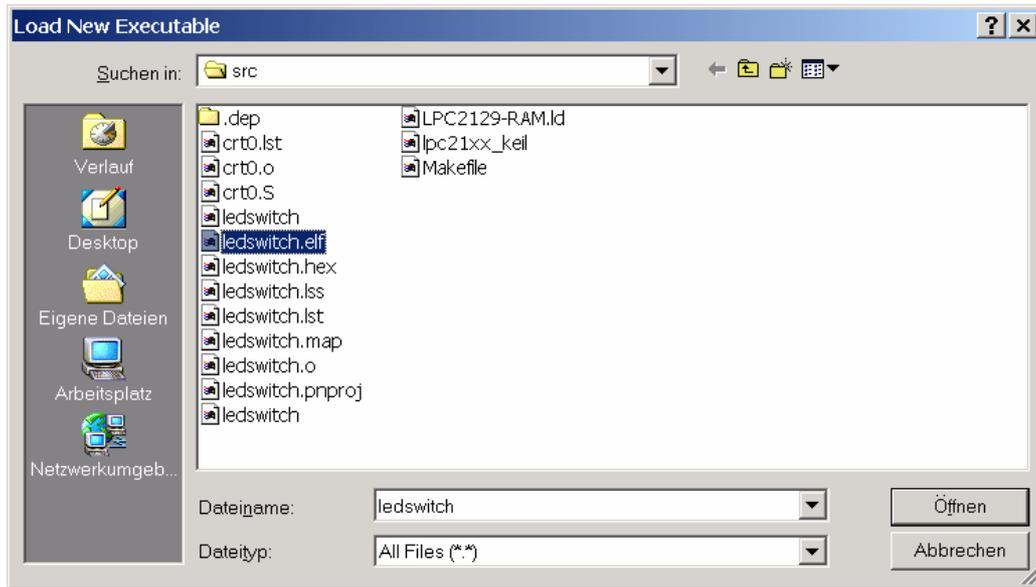


```
C:\WINNT\system32\cmd.exe
OCDEmon InitializeTarget Error : Cable Disconnected
ocdremote 2.11: WIGGLER via LPT 1 at speed : 3
JTAG SDO <-| CPU(1) ARM7TDMI-S : listening on port 8888 |<- JTAG SDI
```

4. Configure settings in the script (batch-file) start_insight.cmd. The script is included in the source archive
5. Start start_insight.cmd
6. Select File/Target Settings... and configure like shown in the following picture.



7. Select File/Open, browse in the source-dir and select the elf-File to debug (ledswitch.elf in this example):



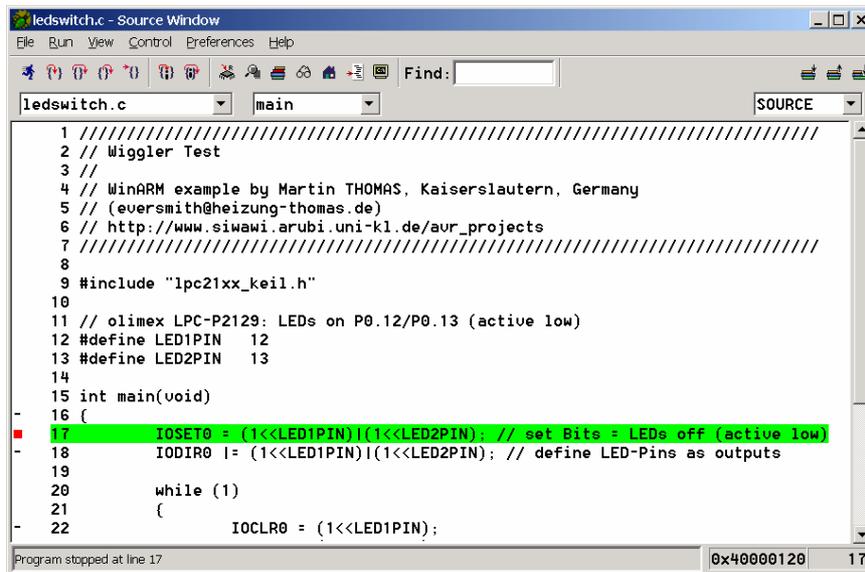
8. Select Run/Connect to target. A message will appear in the "ocdremote-window":

```
C:\WINNT\system32\cmd.exe
OCDEmon InitializeTarget Error : Cable Disconnected
ocdremote 2.11: WIGGLER via LPT 1 at speed : 3
JTAG SDO <-| CPU(1) ARM7TDMI-S : listening on port 8888 |<- JTAG SDI
CPU[1] Accepted gdb connection on port 8888.
```

9. Select Run/Download. Verify in the status bar that the binary has been downloaded:



10. Select Run/Run (or use the "running man"-button in the toolbar)
11. Since "Set breakpoint at main" has been enabled in the target selection dialog the program will halt at the first line in main:



12. Now the menu "Control" or the toolbar-buttons can be used to step through the code.
Useful information is available from the options in the "View" menu (or equivalents in the toolbar)