

Using OpenOCD server and IAR Embedded Workbench

(a simple tutorial)

Installing Device Drivers

Please follow the instructions in **README.txt** file in the drivers folder included in the package. There is nothing specific about the installation process if you have installed Windows drivers before.

Installing IAR Embedded Workbench

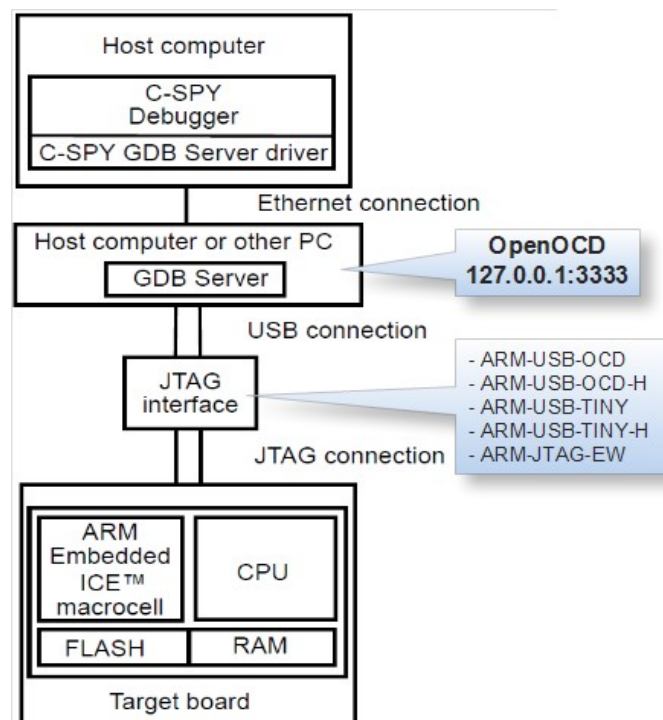
Please note that IAR Embedded Workbench is a commercial product not available for free use. However, there are packages with time limited functionality which may be used for evaluation purposes of the described configuration. Visit :

<http://www.iar.com>

and get the desired product. Installation guide should be available inside the package.

The IAR C-SPY GDB Server Driver

Using the IAR GDB Server driver, C-SPY can connect to the available GDB Server-based JTAG solutions, currently OpenOCD. JTAG is a standard on-chip debug connection available on most ARM processors. The C-SPY GDB Server driver communicates with the GDB Server via an Ethernet connection, and the GDB Server communicates with the JTAG interface module over a USB connection. The JTAG interface module, in turn, communicates with the JTAG module on the hardware.



C-SPY GDB Server communication overview

Configuring the OpenOCD Server

The server executable file is called: **openocd-libftdi.exe**. It is ready to run and needs no installation!

OpenOCD server needs a configuration file upon startup. The default is '**openocd.cnf**' so do not get upset if you start '**openocd-libftdi.exe**' and end up with an error message. A typical solution is to override the search for the configuration file and provide one or more configuration files with different names. For your convenience configuration files are divided into 2 types, one to configure the JTAG interface being used and another to configure the target processor. Interface configuration files for the supported JTAG adapters are placed inside the directory of OpenOCD executable.

Sample target configuration scripts are located in the 'OpenOCD\tcl\target' directory.

Enough long explanations, to start the server open a console and navigate to 'OpenOCD' folder. If you are using ARM-OCD-H and a LPC2124 processor, for example, type:

```
> openocd-libftdi.exe -f olimex-arm-usb-ocd-h.cfg -f ./tcl/target/lpc2124.cfg
```

```
D:\OpenOCD_Olimex\OpenOCD>ls
arm-jtag-ew.cfg      olimex-arm-usb-ocd.cfg      openocd-libftdi.exe
config.log           olimex-arm-usb-tiny-h.cfg  openocd_UG.pdf
olimex-arm-usb-ocd-h.cfg  olimex-jtag-tiny.cfg      tcl

D:\OpenOCD_Olimex\OpenOCD>openocd-libftdi.exe -f olimex-arm-usb-ocd-h.cfg -f ./tcl/target/lpc2124.cfg
Open On-Chip Debugger 0.4.0 (2010-11-23-14:20)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.berlios.de/doc/doxygen/bugs.html
trst_and_srst srst_pulls_trst srst_gates_jtag trst_push_pull srst_open_drain
jtag_nrst_delay: 100
jtag_ntrst_delay: 100
1000 kHz
Info : max TCK change to: 30000 kHz
Info : clock speed 1000 kHz
Info : JTAG tap: lpc2124.cpu tap/device found: 0x4f1f0f0f (mfg: 0x787, part: 0xf1f0, ver: 0x4)
Info : Embedded ICE version 4
Info : lpc2124.cpu: hardware has 2 breakpoint/watchpoint units
```

Screenshot with OpenOCD server started.

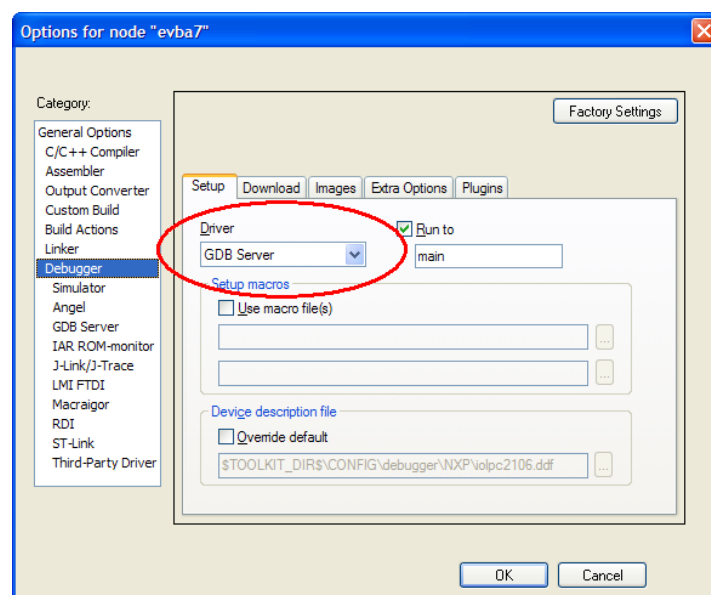
The **-f** command explicitly provides a configuration file to the OpenOCD server. The first file tells OpenOCD that you will be using ARM-OCD-H JTAG adapter, and the second file describes the LPC2124 processor.

You will generally be more than unlucky to find no configuration file matching your target. Then you should write one using a file describing a platform closest to yours. Please consult OpenOCD user's guide for explanation of the various sections of such a file. Of course you will need the datasheet of the target device and some patience to try things until you set them right.

Configuring IAR IDE

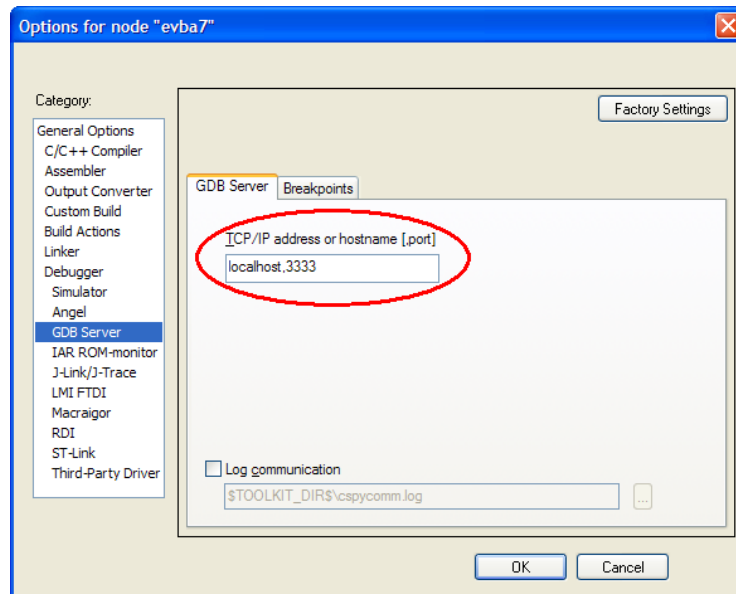
With OpenOCD running in a command prompt window you now need to configure IAR IDE.

1. Start IAR IDE and open a project for your target. There are quite a lot example projects for quick starters to play around – **Help>Information Center>Example projects**.
2. Compile with no errors (and no warnings if possible ;)).
3. Choose **Project>Options>Debugger** and select the debugger to use:



Select "GDB Server" in the list of available debuggers.

4. Choose **Project>Options>Debugger>GDB Server** and specify the location of the OpenOCD server. If the server is located on the host computer, specify 'localhost' with port 3333, otherwise specify the host name or IP address. For the example configuration type: **localhost,3333**



Choose IP address and port of the OpenOCD server.

Note that you are prompted to enter a dot after the IP address, bu IAR actually expects a space or comma character (boggles the mind, huh :))!

5. Start the debug session (**Project>Download and Debug**) and click the Run button when downloading has finished. You should be able to use breakpoints, step execution and all the other stuff IAR provides.

Conclusion

Please note that OpenOCD is sometimes pretty tricky to work with so be aware of configuration and runtime issues. Well, if things get very ugly you should stop the server, reconnect the JTAG adapter and start over.

If in need of more information check out any of the following:

- <http://openocd.berlios.de/web/> - the official web page of the OpenOCD project.
- <http://www.iar.com/> - home of the IAR development tools.
- <http://www.olimex/dev/> - manufacturer of development boards and tools. That's us :).

THE END