On Chip Debugging mit GALEP-5 und OpenOCD

Die GALEP-5 Programmiergeräte-Serie läßt sich als Hardwareinterface für **OpenOCD** einsetzen. Bei dem Open OCD Projekt handelt es sich um ein freies Software-Projekt, welches Mikrocontroller (MCU) per JTAG-Schnittstelle ansprechen kann. Damit läßt sich Software auf der Original-MCU in der Original-Schaltung debuggen. Hierbei können Hardware- und Software-Breakpoints ausgelöst und das Programm unterbrochen und im Single Step Modus ausgeführt werden… ganz ohne Emulator und - außer GALEP - ohne Zusatz-Hardware. Die JTAG-Funktionalität kann verwendet werden zum:

- Entwickeln von Software auf dem Zielsystem
- Programmieren von Flash-Bausteinen
- Testen von Schaltungen und einiges mehr...

GALEP-5 eignet sich besonders gut als Hardware-Interface für OpenOCD, da auf dem Programmiergerät selbst ein ARM-Linux läuft. Die OpenOCD-Software läuft also nativ auf dem GALEP-5 und stellt über Ethernet oder emuliertem USB-Port eine **gdb** Debugging-Schnittstelle zur Verfügung. Dies erleichtert dem Anwender die Konfiguration und erlaubt komfortables Arbeiten.

GALEP-5 bietet eine Reihe weiterer Features, die im Zusammenhang mit JTAG Debugging sinnvoll sind. So verfügt GALEP-5D über eine Ethernet Schnittstelle für schnellen Datenaustausch. Sein Taster an der Frontseite kann von OpenOCD verwendet werden. Die JTAG-Signale können beliebigen Sockelpins zugeordnet werden. Dieses Merkmal - eine wirksame Waffe gegen Layoutfehler - bietet nur die GALEP-5 Serie.

Inhalt

Überblick

Dieses Kapitel richtet sich an Entwickler, welche die OpenOCD Software benutzen wollen, um ihr Zielsystem zu debuggen oder zu programmieren.

Allgemeiner Hinweis

Wenn Sie GALEP-5 als Interface für OpenOCD verwenden möchten, können Sie auf eine für Sie vorcompilierte OpenOCD Version zurückgreifen. In diesem Fall müssen Sie OpenOCD nicht selbst kompilieren. Diese Version entspricht einem Schnappschuss des OpenOCD Projektes zum Zeitpunkt der Erstellung dieser Release. Da sich erfahrungsgemäß viel am OpenOCD Projekt ändert und ständig neue Funktionen hinzukommen bzw. Fehler behoben werden, kann es sinnvoll sein, die aktuelle Version von OpenOCD selbst zu patchen und zu kompilieren. Dies ist in Teil II dieses Dokuments beschrieben.

Was Sie von uns erhalten

Wir stellen auf **http://www.conitec.net/down/g5ocd.zip** folgendes zur Verfügung:

doc/g5ocd.htm	**Dieses Dokument im html Format.**
g5ocd-bin-svn-xyzM.tar.bz2	**Der GALEP-5 OCD Port, fertig kompiliert für Linux.**
g5ocd-bin-svn-xyzM.tar.zip	**Der GALEP-5 OCD Port, fertig compiliert, Windows™ ZIP format.**

g5ocd-svn-xyzM.patch.bz2	**Patch zum entsprechenden OpenOCD-Trunk (wenn Sie OpenOCD selbst kompilieren möchten).**
g5API-abcM.tar.bz2	**Die GALEP-5 API für Entwickler.**

Die Infixe “xyz” bzw. “abc” stehen dabei für Versionsnummern der entsprechenden Pakete.

Was der GALEP-5 / OpenOCD Port leistet

Dieses Software Paket versetzt den Anwender in die Lage, ein hochwertiges und schnelles Hardware-Interface (das GALEP-5 Programmiergerät) mit einer Open Source Software (**OpenOCD**) zu kombinieren, die in der Lage ist, ein Reihe von MCUs zu debuggen und eine Reihe von Flashs zu programmieren. Generell kann man mit GALEP-5 und OpenOCD einen ähnlichen Komfort erreichen, wie er nur bei sehr hochpreisigen JTAG-Debuggern zu finden ist. Die Software hat sich bereits bei der Fehlersuche in ARM9TDMI und ARM9-EJIS Cores bewährt (zum Zeitpunkt der Erstellung dieses Dokuments war die Unterstützung für EJIS-Cores noch nicht ausgereift).

Was der GALEP-5 / OpenOCD Port nicht leistet

Genau wie OpenOCD ist der GALEP-5 Port ein Open-Source-Projekt, aus dem keine einklagbaren Ansprüche auf Funktionalität und Stabilität abzuleiten sind. Wir bemühen uns, die Software aktuell zu halten, stellen aber auch dem Anwender alle Sources zur Verfügung, die notwendig sind, um die Software selbst zu erweitern, zu aktualisieren und zu verbessern.

GALEP-5 wird zum JTAG Debugger

Damit der JTAG-Debugger optimal mit Ihrem Zielsystem zusammenarbeitet, muss eine Konfigurationsdatei erstellt werden. Diese ist recht einfach aufgebaut und enthält Konfigurationsdaten für das OpenOCD - Programm und dessen Galep-Anbindung (s.u.). Wie eingangs erwähnt, läuft OpenOCD nativ auf dem Programmiergerät. Es muss also zusammen mit der Konfigurationsdatei auf den GALEP übertragen werden, bevor es gestartet werden kann. Die Übertragung erfolgt mit Hilfe eines FTP-Clients, der Start erfolgt mit Hilfe eines Telnet-Clients.

Die Konfigurationsdatei

Die Konfigurationsdatei ist eine Textdatei und enthält Interface-Einstellungen, Target-Einstellungen, und OpenOCD-Einstellungen. Ein Beispiel:

```
# Example that uses the galep5 hardware as ocd interface.
# The socket is configured to interface the arm&eva hardware.
# (See http://www.conitec.com/english/linuxboard.htm)

# interface
interface galep5
interface_speed 0
jtag_speed 100
logic_level 1800
socket_assign TargetDetect 24
socket_assign GND          26
socket_assign GND          27
socket_assign GND          28
socket_assign GND          29
socket_assign GND          30
socket_assign GND          31
socket_assign GND          32
socket_assign GND          33
socket_assign GND          34
socket_assign NTRST        23
socket_assign TDI          22
socket_assign TMS          21
socket_assign TCK          20
socket_assign TDO          18
socket_assign NRESET       17

# daemon configuration
```

```
telnet_port 4444
gdb_port 3333

# use combined on interfaces or targets that can't set TRST/SRST separately
reset_config trst_and_srst

# jtag scan chain
#format L IRC IRCM IDCODE (Length, IR Capture, IR Capture Mask, IDCODE)
jtag_device 4 0x1 0xf 0xe

# target configuration
daemon_startup reset
# target <type> <endianess> <reset mode>
target arm9tdmi little reset_run 0 arm920t
working_area 0 0x200000 0x4000 backup
run and halt time 0 5000

# flash bank <driver> <base> <size> <chip_width> <bus_width> [driver_options ...]
flash bank cfi 0x10000000 0x800000 2 2 0
```

Interface Einstellungen

Zu den Interface-Einstellungen gehören GALEP-5 spezifische Informationen. Die Art der Einstellungen lassen sich am einfachsten anhand des obigen Beispiels erläutern.

Interface-Typ

Zunächst wird mit dem Eintrag **interface = galep5** festgelegt, dass das GALEP-5 5 / OpenOCD - Binding verwendet werden soll. Dieses Binding unterstützt alle GALEP-5 Derivate (GALEP-5M, GALEP-5D und GALEP-5P).

Geschwindigkeit

Es folgen geschwindigkeitsrelevante Einstellungen **interface_speed = 0** und **jtag_speed = 100**. Je höher der Wert von "interface_speed", desto mehr Wartezyklen werden zwischen JTAG-Aktivitäten eingelegt. (Ein Wartezyklus entspricht ca. 1 / 60MHz). Sofern keine Probleme mit der JTAG Kommunikation auftreten brauchen keine Wartezyklen eingelegt werden.

Logikpegel

Mit **logic_level = 1800** werden die Logikpegel der JTAG Signale festgelegt. Sie können in einem Bereich zwischen 1.2V und 4.5V frei definiert werden. Die Angabe erfolgt in Millivolt. **!!** Die Signalparameter, welche für den jeweiligen JTAG Port gelten, sind dem Datenblatt der MCU zu entnehmen. Fehlerhafte Werte führen zur Fehlfunktion und können ggf. Ihr Zielsystem beschädigen!

Sockelzuordnung

Der GALEP-5 erlaubt eine völlig freie Zuordnung zwischen den Sockel-Pins und den erforderlichen JTAG Signalen. Die Zuordnung beginnt mit dem Schlüsselwort **socket_assign** und erwartet einen Signalnamen und eine Pin-Nummer als Argument. Die folgende Tabelle enthält eine Liste möglicher Signalnamen und deren Bedeutung:

Signalname	Bedeutung	Richtung	Signalparameter	Hinweis
TargetDetect	Siehe Text.	Target ► GALEP-5	1.2 - 25V	
VCC1	Siehe Text.	GALEP-5 ► Target	1.2 - 25V	a b
VCC2	Siehe Text.	GALEP-5 ► Target	1.2 - 25V	a b
VCC3	Siehe Text.	GALEP-5 ► Target	1.2 - 25V	a b
GND	Siehe Text.	GALEP-5 ► Target	0V	a
NRESET3v3	Target Reset Signal	GALEP-5 ► Target	3.3V	
NRESET	Target Reset Signal	GALEP-5 ► Target	1.2V - 4.5V	c
NTRST	JTAG Reset	GALEP-5 ► Target	1.2V - 4.5V	c
TDI		GALEP-5 ► Target	1.2V - 4.5V	c
TMS		GALEP-5 ► Target	1.2V - 4.5V	c
TCK		GALEP-5 ► Target	1.2V - 4.5V	c
TDO		Target ► GALEP-5	1.2V - 4.5V	c

[a] Mehrfache Verwendung für verschiedene Pins erlaubt.
[b] Wert hängt vom Parameter **vcc_level VCCx = mV** ab.
[c] Signalparameter (Bereich) hängt vom Parameter **logic_level = mV** ab.

Die JTAG relevanten Signale dürfen nur einmal spezifiziert werden. Die GND und VCC Signale können hingegen mehrfach für verschiedene Pins angegeben werden. Somit sind mehrere GND und VCC Pins am Sockel möglich. Folgende Abbildung veranschaulicht die entsprechende Sockelbelegung:

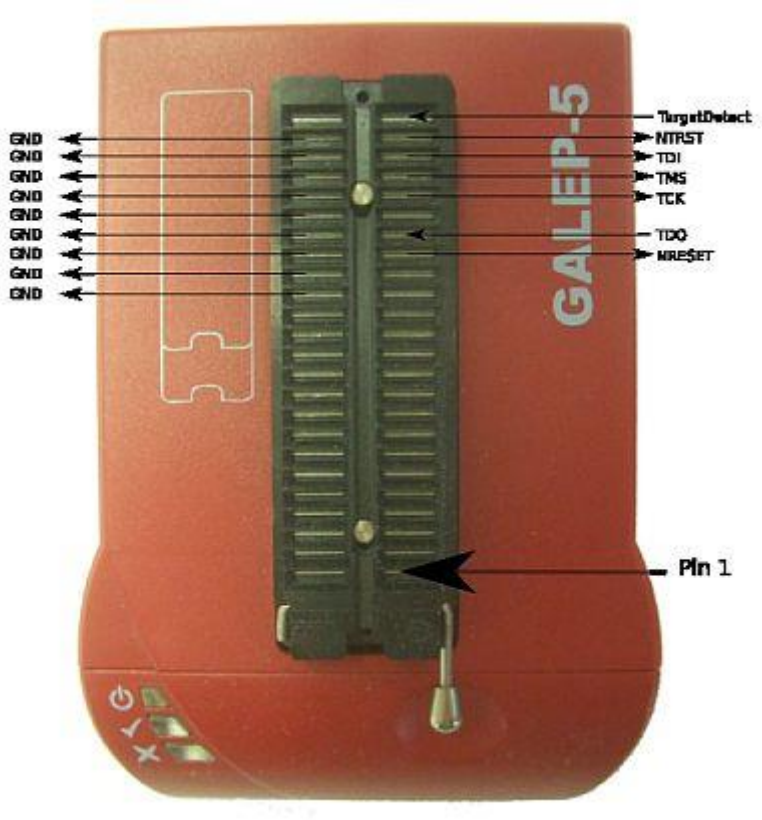

Abb: Sockelbelegung wie in der Beispiel Konfigurationsdatei gegeben.

Zusätzliche Spannungen

Der GALEP-5 kann bis zu drei unterschiedliche Spannungen - **VCC1..VCC3** - am Sockel zur Verfügung stellen. Alle drei Spannungen sind im Bereich vom 1.2V - 25V einstellbar. **!!** Der Strom pro einstellbarer Spannung darf 250mA nicht überschreiten. Weiterhin darf die maximale Leistung pro Spannung von 1.2W nicht überschritten werden! Bei Verwendung des GALEP-5-M Gerätes ist das externe Netzteil zu verwenden, wenn von den einstellbaren Spannungen Gebrauch gemacht wird.

In der Konfigdatei können die Spannungen wie folgt eingestellt werden:

```
# interface

# ...

vcc_level VCC1 3300
vcc_level VCC2 4500
vcc_level VCC3 5000

socket_assign VCC1        1
socket_assign VCC2        2
socket_assign VCC3        3

# ...
```

In dieser Konfiguration werden an den Sockel Pin 1 3.3V, an Pin 2 4,5V und an Pin 3 5V angelegt. Die Spannungen werden sofort nach dem Start der Software eingeschaltet und nach Beenden von OpenOCD ausgeschaltet.

Erkennung des Targets

Das Signal **TargetDetect** beschreibt einem Eingang, welcher vom Zielsystem auf high gezogen werden sollte (z.B. die Betriebsspannung des Zielsystems.) Wenn das Signal mit einem Sockelpin belegt wird (z.B. Konfigurationsparameter **socket_assign TargetDetect = 24**) so prüft die Software beim Start, ob das Zielsystem einen High-Pegel liefert und bricht ggf. mit einem Fehler ab, wenn dem nicht so ist. Ist das Signal in der Konfigurationsdatei nicht angegeben, so wird dieser Test übersprungen.

Belegung des Tasters (Nur GALEP-5D)

Der GALEP-5D verfügt über einen Taster an der Frontseite. Im Zusammenhang mit der OpenOCD -Software kann der Taster dazu verwendet werden bis zu drei Funktionen zu bedienen. Als Funktion ist dabei ein OpenOCD -Skript zu verstehen, welches dieselben Befehle enthalten kann, die auch in der telnet - Konsole gültig sind.

Die Funktionen können über unterschiedlich langes Drücken des Tasters abgerufen werden. Dabei signalisiert eine LED unter dem Taster die jeweilige Funktion. Durch Loslassen des Tasters wird die Funktion ausgeführt.

Taster	**LED**	**Loslassen**
kurz	**Blinkt nicht**	**Funktion 1**
länger	**Blinkt zwei mal**	**Funktion 2**
unglaublich lang	**Blinkt drei mal**	**Funktion 3**

Die Funktionen werden über den Konfigurationsparameter **button_assign** festgelegt. Ein Beispiel:

```
# interface
#                                                                                    ...
button_assign                                                          1        btShort.script
button_assign                                                           3        btLong.script

# ...
```

Die Skripte **btLong.script** und **btShort.script** müssen als Dateien vorliegen, welche die entsprechenden Kommandos enthalten. Es handelt sich um die selben Kommandos / Kommandofolgen, wie sie auch in der Telnet-Verbindung zum OpenOCD-Programm möglich sind (s.u.). Die Ausgabe der Kommandos erfolgt auf der Konsole, von der das OpenOCD-Programm gestartet wurde, und auf der ersten Telnet-Verbindung zum OpenOCD-Programm (sofern eine existiert).

OpenOCD Einstellungen

Alle weiteren Einstellungen sind OpenOCD-spezifisch. Hier sei auf die Dokumentation des OpenOCD Projektes verwiesen.

Die ersten Schritte

Wie eingangs erwähnt, wird OpenOCD auf dem Programmiergerät gestartet und muss daher zuvor, einschliesslich der Konfigurationsdatei, auf den GALEP-5 übertragen werden. Hiefür kann jeder handelsübliche FTP-Client verwendet werden. Folgende Schritte sind erforderlich, dann können Sie mit dem Debuggen beginnen:

1. Auspacken des vorkompilierten Archives (**g5ocd-bin-svn-xyz.tar.bz2**) bzw. der .zip Version in ein Verzeichnis Ihrer Wahl.
2. Hinzufügen Ihrer Konfigurationsdatei (s.o.)
3. Übertragen aller Dateien zum GALEP-5.

4. Einloggen mit einem Telnet-Client auf dem GALEP-5
5. Das Programm“**openocd**” und das Skript “**ocd**” ausführbar machen. Dieser Schritt ist nur erforderlich, wenn Sie unter Windows™ arbeiten, da Dateiattribute wie “ausführbar” weder im ZIP Format abgelegt noch mit Windows™-basierenden FTP Clients übertragen werden können.
6. Ausführen von OpenOCD mit Hilfe der Telnet-Verbindung

Zielsystem mit dem GALEP-5 verbinden

Die Verbindung zum Zielsystem erfolgt über ein Sockelverbindungskabel, welches Sie von Conitec erwerben (**OCD-JTAG-Kabel20**, Bestellnummer 095540) oder selbst zusammenbasteln können. Es hat die Belegung eines Standard-ARM9-JTAG Verbinders, wie sie auch bei anderen JTAG-Debuggern verwendet wird. Da die Zuordnung zwischen Sockel und den JTAG-Leitungen des Zielsystems frei konfigurierbar ist, können Sie auch leicht selbst ein solches Kabel mit seiner spezifschen Belegung herstellen.

Verbindung herstellen

Nachdem der GALEP-5 gestartet wurde und weder die OpenOCD -Software gestartet wurde, noch eine Verbindung zur PC-Galep Software vorliegt, ist der Sockel hochohmig. Es ist also in diesem Zustand möglich, das stromlose Zielsystem mit dem GALEP-5 zu verbinden. **!!** Das Zielsystem sollte stromlos sein, während Sie die Verbindung herstellen. Weiterhin darf der Galep nicht mit der Galep-PC-Software verbunden sein!

Nachdem die Verbindung hergestellt wurde, kann das Zielsystem eingeschaltet werden. Für den Fall, dass Sie das Zielsystem aus dem GALEP-5 mit Strom versorgen, wird es eingeschaltet, sobald Sie die OpenOCD Software starten. Die folgende Abbildung zeigt als Beispiel den GALEP-5M, der mit unserem Arm & Eva - ARM9 Entwicklungsboard verbunden ist

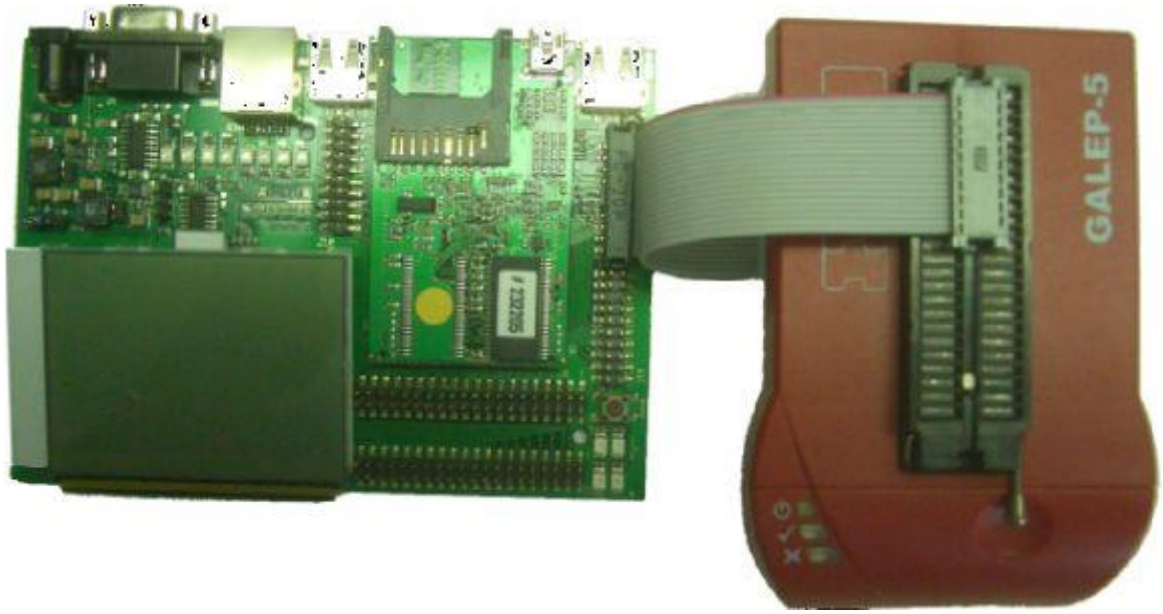

Abb: Verbindung von “Arm&Eva” mit “OCD-JTAG-Kabel20”.

OpenOCD auf dem Galep starten

Auf dem GALEP-5 läuft ein Telnet-Server. Nur der Benutzer “root” ist gültig. Es wird kein Passwort erwartet.

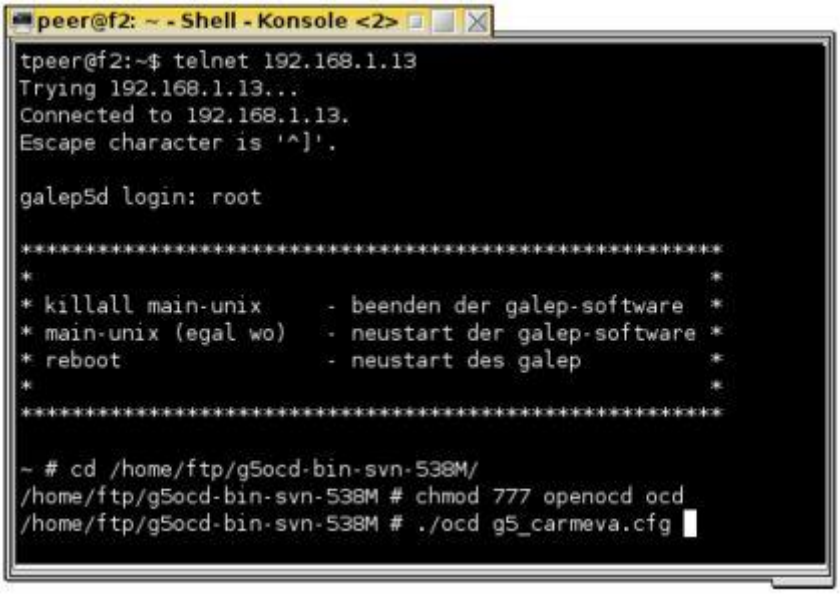

Abb: Anmeldung per Telnet.

Nach der Anmeldung erwartet GALEP Befehle über die Telnetverbindung. Programme, welche mit Hilfe des FTP-Protokolls übertragen wurden, befinden sich im Verzeichnis „/home/ftp" auf dem GALEP. Im obigen Bild sind die folgenden Befehle bereits eingeben worden.

cd /home/ftp
chmod 777 ocd openocd
./ocd g5_carmeva.cfg

Die letzte Zeile startet die OpenOCD -Software. Sie kann mit Ctrl+C beendet- und jederzeit erneut gestartet werden. Wenn das Zielsystem gefunden wurde, erscheint z.B. folgende Meldung:

```
./ocd g5_carmeva.cfg
Open On-Chip Debugger 1.0 (2008-04-04-12:28) svn:538M
$URL: svn://svn.berlios.de/openocd/trunk/src/openocd.c $
Setting interface speed: No delay.
Assigning "TargetDetect"-> Socket pin 24
Assigning "GND"-> Socket pin 26
Assigning "GND"-> Socket pin 27
Assigning "GND"-> Socket pin 28
Assigning "GND"-> Socket pin 29
Assigning "GND"-> Socket pin 30
Assigning "GND"-> Socket pin 31
Assigning "GND"-> Socket pin 32
Assigning "GND"-> Socket pin 33
Assigning "GND"-> Socket pin 34
Assigning "NTRST"-> Socket pin 23
Assigning "TDI"-> Socket pin 22
Assigning "TMS"-> Socket pin 21
Assigning "TCK"-> Socket pin 20
Assigning "TDO"-> Socket pin 18
Assigning "NRESET"-> Socket pin 17
Assigning button function 1 -> exec "btShort.script"...
Assigning button function 3 -> exec "btLong.script"...
Info: options.c:50 configuration_output_handler(): jtag_speed: 100, 100
Setting up socket...
Using 1.8V logic levels.
Target supply detected.
```

Info: jtag.c:1346 jtag_examine_chain(): JTAG device found: 0x05b0203f (Manufacturer: 0x01f, Part: 0x5b02, Version: 0x0)

Schnittstellen von OpenOCD

Die OpenOC -Software öffnet zwei Netzwerk Ports:

- Einen Port, welcher mit einem Telnet-Client bedient werden kann.
- Einen Port, welcher die **gdb** Remote-Spezifikation erfüllt.

Telnet-Port

Es handelt sich um den Port, welcher in der Konfigurationsdatei definiert ist, im obigen Beispiel per Anweisung **telnet_port = 4444**. Die Verbindung erfolgt mit Hilfe eines telnet-Clients, wobei der Port entsprechend der Konfigurationsdatei anzugeben ist. Mit Hilfe dieser Verbindung können Befehle direkt and die OpenOCD -Software gesendet- und der Debugger direkt beeinflusst werden. Befehle sind z.B:

- Prozessorregister anzeigen
- Hardware Breakpoints anzeigen / modifizieren
- Zielsystemreset
- vieles mehr

Ausgaben von OpenOCD Skripten, welche dem GALEP-5-D Taster zugeordnet sind, werden auch auf diese Telnet Verbindung umgeleitet. Genauere Informationen können der OpenOCD Software Dokumentation entnommen werden.

gdb Port

Auch hierbei handelt es sich um einen Netzwerk Port, welcher in der Konfigurationsdatei spezifiziert werden kann: **gdb_port = 3333** setzt ihn beispielsweise auf den Wert 3333. Der GDB (Gnu Debugger) Port wird vom eigentlichen Debugger (gdb) verwendet, um Source Level Debugging zu ermöglichen.

Debuggen mit gdb

Der Debugger ("**gdb**") läuft auf dem Entwicklungsrechner. Er muss mit Zielsystemunterstützung compiliert worden sein und ist normalerweise Bestandteil der Toolchain für ein bestimmtes Zielsystem. Handelt es sich beispielsweise um ein ARM basierendes Zielsystem, benötigen Sie einen gdb, welcher mit ARM Unterstützung compiliert wurde. Da der gdb im Quellcode frei erhältlich ist, steht es jedem frei, den Debugger selbst zu kompilieren. Er läuft auf verschiedenen Platformen und wird teilweise vorcompiliert zum Download angeboten.

Es existieren verschiedene Frontends, welche die Bedienung erleichtern können. Im Folgenden werden einige mögliche Szenarien beleuchtet.

Das Shell-Frontend

Das Shell Frontend des GDB ist textbasiert und auf den ersten Blick spartanisch. Dennoch ist dies oft die praktischste Art zu debuggen, da die grafischen Frontends erfahrungsgemäß oft versagen (zumindest im Hinblick auf JTAG debugging).

Nehmen Sie sich die Zeit, den Debugger kennenzulernen! Es handelt sich um ein äusserst mächtiges und gut funktionierendes Werkzeug.

Vorbereitungen

Starten Sie Ihr Zielsystem. Ja nachdem was Sie debuggen wollen und wie Ihr Programm zur Ausführung kommt, können verschiedene Schritte notwendig sein. Beispiele:

1. Sie übertragen die Software, welche es zu debuggen gilt, per USB.
    1. Nach dem Reset Ihres Zielsystems müssen Sie die Übertragung vornehmen und Ihr Programm ausführen.
    2. Starten Sie die OpenOCD Software auf dem GALEP-5.
    3. Das Zielsystem wird sofort nach dem Start von OpenOCD angehalten (abhängig vom **reset_halt** Parameter des **target** Konfigurationseintrags - s. OpenOCD Dokumentation) und Sie können sich mit gdb verbinden.
2. Ihre Software startet automatisch (z.B. aus einem Flash).
    1. Starten Sie die OpenOCD Software auf dem GALEP-5.
    2. Sie können sich mit gdb verbinden

Unter Umständen kann es Sinn machen, zusätzlich eine Telnet Verbindung zur OpenOCD Software zu öffnen (s.o.).

Verbindung herstellen

```
(gdb) target remote 192.168.1.13:3333
```

Nachdem eine Verbindung zustande gekommen ist, können gdb Befehle wie “continue”, “ss”, “ni” oder “break” verwendet werden, um das Programm zu steuern. An dieser Stelle sei auf eine wirklich gute Kurzreferenz des gdb verwiesen. Sie ist unter http://www.cs.dal.ca/studentservices/refcards/gdbref.pdf abrufbar.

Weitere Frontends für den gdb

Es existieren eine ganze Reihe an Frontends für den gdb:

- kdbg
  Aufruf: kdbg -r 192.168.1.13:3333 ProgrammName
  Vorher sollte in kdbg -> Global Settings der entsprechende gdb konfiguriert werden.
- kdevelop
  Funktioniert mit Hilfe eines Start Skriptes, welches man unter Project ► Project Options ► Debugger einstellt.
- ddd
  Nicht getestet.
- Insight
  Nicht getestet.
- Eclipse
  Nicht getestet.

Unter diesen Entwicklungssystemen ist der gdb einbindbar. Eine detaillierte Beschreibung würde den Rahmen dieses Dokuments sprengen. Für nähere Informationen sei auf das Internet verwiesen.

Entwickeln mit OpenOCD und GALEP-API

Die Firma Conitec Datensysteme hat alle zur Steuerung des GALEP-5 notwendigen Bibliotheken unter der GPL veröffentlicht. Das Entwicklungssystem liegt in Form eines komprimierten Archives (**g5api-abcM.tar.bz2**) vor. Das Archiv enthält, neben dem Quellcode, ebenfalls die compilierte Bibliothek, welche direkt auf dem GALEP-5 lauffähig ist.

OpenOCD selbst compilieren

Dieses Kapitel richtet sich an Entwickler, die mit dem Umgang von Linux vertraut sind. Weiterhin werden Kenntnisse im Umgang mit Cross-Compilern, make, subversion, usw. vorausgesetzt.

Benötigte Software

Um OpenOCD für den GALEP-5 zu kompilieren wird folgendes benötigt:

1. Cross Toolchain für ARM
2. make, automake, autoconf-2.13, ...
3. Die GALEP-5 API (s.o.)
4. Das aktuelle GALEP-5 OCD Patch (s.o.)
5. Die offizielle OpenOCD Distribution.

Cross Toolchain für ARM

Es wird eine Cross Toolchain (Compiler, Linker, ...) für die ARM9 Platform benötigt. Es kann zum Beispiel die Toolchain verwendet werden, welche wir für Linux (32bit) vorkompiliert anbieten und die auch für unser Arm&Eva - Linux Board verwendet wird:

http://armeva.conitec.net/i686/tar_bz2/carmeva-dev-cc-3.4.1-glibc-2.3.3-bin-1.1-9_i386.tar.bz2

GALEP-5 -API und g5ocd-Patch

Diese Softwareteile finden Sie unter **http://www.conitec.net/down/g5ocd.zip**.

Open OCD Sources

Das OpenOCD Projekt stellt ein subversion Repository zur Verfügung, wo Sie die aktuellsten Sourcen beziehen können (Links s.u.).

Kompilieren von OpenOCD mit GALEP-5 Unterstützung

Zunächst werden alle Quellen entsprechend vorbereitet.

1. Erzeugen eines leeren Verzeichnisses “g5ocd”
2. Quellen hinein kopieren, sodass folgender Aufbau entsteht:

```
g5ocd
   |-- g5API
   |   |-- hdw
   |   |   `-- openocd.bin
   |   |-- include
   |   |   |-- ...
   |   |-- lib
   |   |   `-- libg5api.so
   |   |-- src
   |   |   |-- ...
   `-- openocd
       |-- doc
           |-- ...
       |-- ecosflash
           |-- ...
       |-- testing
           |-- ...
       |-- src
           |-- ...
       |-- g5ocd-svn-538M.patch.bz2
```

Patch einspielen

```
cd g5ocd/openocd
bzip2 -dc g5ocd-svn-xyz.patch.bz2 |patch -p0
```

Konfigurieren

```
./bootstrap
./configure --enable-galep5
```

Compilieren

Da OpenOCD für den ARM (Cross-) kompiliert werden muss, kann wie folgt compiliert werden:

```
# We use this for cross compiling...
PREFIX=/usr/local/carmeva/bin/compiler/gcc-3.4.1-glibc-2.3.3/\
        arm-9tdmi-linux-gnu/bin/arm-9tdmi-linux-gnu-
export PREFIX

# Build the stuff...
make CC=${PREFIX}gcc CPP=${PREFIX}gcc CXX=${PREFIX}gcc

# Strip the result...
${PREFIX}strip src/openocd
```

Nach dem Compilieren liegt das Binary “openocd” im “src” Verzeichnis.

Informationen und Details zu OpenOCD

Eine vollständige Beschreibung aller Merkmale von OpenOCD würde den Rahmen dieses Dokuments bei weitem sprengen. Hier einige weiterführende Links:

Die OpenOCD Homepage - Hier finden Sie die vollständige Dokumentation des OpenOCD Projektes: **http://openocd.berlios.de/web/**

Die OpenOCD Mailing Liste - Es handelt sich um eine recht aktive Mailing List, auf der die Entwickler OpenOCD -relevante Fragen beantworten.

Informationen über den GNU Debugger - Alles, was das eigentliche Debugging betrifft (Bedienung und Verhalten des gdb) wird in zahlreichen Dokumenten rund im den gdb behandelt.

- **http://www.cs.dal.ca/studentservices/refcards/gdbref.pdf**
  Eine Befehlsreferenz für den gdb.
- **http://sourceware.org/gdb/documentation/**
  Dokumentation des gdb.