

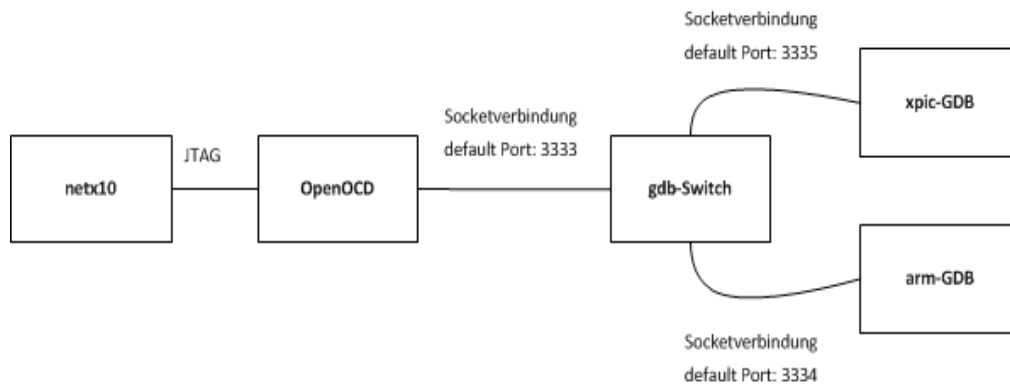


## **gdb-Switch Dokumentation**

Datum	Version	Author	
xx.12.2010	x.x	D. Kramlinger	Initial Version

# Programmbeschreibung

Der *gdb-Switch* dient als Vermittlungsstelle zwischen *OpenOCD* und *xpic-GDB* bzw. *arm-GDB*. Er übernimmt die Aufgabe, Befehle der beiden GDBs nebenläufig zu empfangen, im Falle des *xpic-GDBs* an den *xpic* anzupassen und an *OpenOCD* weiter zu senden.



## Benutzung des Programms

### Starten

Das Programm wird mit **netx10\_gdb\_switch.exe** gestartet. Ohne angegebene Parameter läuft es ohne Konsolenausgaben und benutzt die default Ports 3333 für *OpenOCD*, 3334 für *armGDB* und 3335 für *xpicGDB*.

Werden andere Ports für die jeweilige Verbindung benötigt, können diese durch Angabe folgender Parameter beim Starten des Programms geändert werden:

- a xxxx für den armGDB
- x xxxx für den xpigGDB
- o xxxx für OpenOCD

Mit dem Parameter **-v** wird der verbose Modus aktiviert. In diesem Modus werden in der Konsole Informationen ausgegeben, in welchem Zustand sich das Programm befindet. Z.B. dass eine Socket Verbindung aufgebaut oder beendet wurde.

**Hinweis:** OpenOCD muss bereits gestartet und mit dem Target (netx10) verbunden sein, bevor der Switch gestartet wird.

### Betrieb

**Hinweis:** alle folgenden Befehle werden in einem der beiden Konsolen der GDBs eingegeben. Wenn nicht anders angegeben, gilt der Befehl für beide GDBs. Ansonsten ist angegeben auf welchen GDB (xpig-gdb oder arm-gdb) sich ein Befehl bezieht.

### Verbindung aufbauen

Nachdem der gdb-Switch gestartet wurde, können xpig-GDB und arm-GDB gestartet werden. Für

einen stabilen Betrieb des Switches ist es erforderlich, den Defaultwert für Timeouts in beiden GDBs anzupassen. Dies geschieht mit folgendem Befehl:

**set remotetimeout 10**

Ist dies geschehen, können die GDBs mit dem Switch verbunden werden:

**target remote 3335** (xpica-gdb)

**target remote 3334** (arm-gdb)

Sollten beim Start des Switches andere Ports angegeben worden sein, müssen die Werte 3335/3334 durch die neuen ersetzt werden.

## Laden eines Programms

Ein Program kann mit folgendem Befehl in den GDB geladen werden

**file programmname.elf**

wobei programmname.elf für das entsprechende Programm steht.

Mit dem Befehl

**load**

wird das geladene Programm auf das jeweilige Target gespielt. Beim arm-GDB auf die arm und beim xpica-GDB auf den xpica.

**Hinweis:** bevor ein neues Programm geladen und gestartet werden kann, sollte unbedingt ein Reset der jeweiligen CPU vorgenommen werden. Siehe Kapitel *Reset des Systems*.

## Reset des Systems

Um ein System zu resetten, kann folgender Befehl in dem jeweiligen GDB benutzt werden:

**monitor reset**

Im Falle des arm-GDBs wird die arm resettet und im Falle des xpica-GDBs der xpica.

**Hinweis:** wenn der reset-Befehl vom arm-GDB kommt, wird das gesamte netx10 System resettet. Dies bedeutet, dass nach solch einem Reset der xpica neu initialisiert werden muss. Dies geschieht mit dem Befehl

**monitor init**

und muss vom xpica-GDB aus aufgerufen werden.

## Beenden des gdb-Switchs

Um den gdb-Switch zu beenden, wird der Befehl

**monitor exit**

von der GDB Konsole aus gesendet.

## Zusammenfassung

Starten des Programms:

netx10\_gdb\_switch.exe

mögliche Startparameter:

-a xxxx	– Port armGDB
-x xxxx	– Port xpicGDB
-o xxxx	– Port OpenOCD
-v	– Verbose Mode

Befehle von der GDB Konsole aus:

**set remotetimeout 10** – Timeout anpassen  
**file programmname.elf** – Programm in den GDB laden  
**load** – Programm auf das Target spielen  
**monitor reset** – xpic bzw. arm resettet  
**monitor init** – xpic initialisiern ( nur xpic-gdb)  
**monitor exit** – gdb-Switch beenden

## Beispielsession

```
(gdb) set remotetimeout 10
(gdb) file xpic.elf
Reading symbols from xpic.elf...done.
(gdb) target remote :3335
Remote debugging using :3335
0x00000b98 in main () at new_main.c:408
408      NX10_XpicTimer_Start(0,
NX10_XPICTIMER_CFG_MODE_PRELOAD_AT_0, 0x
fffff);
(gdb) load
Loading section .text_intram, size 0x3efc lma 0x8030000
Loading section .text_local, size 0xb9c lma 0x1018c000
Loading section .data_local, size 0x7c lma 0x1018e000
Start address 0x0, load size 19220
Transfer rate: 17 KB/sec, 4805 bytes/write.
(gdb) c
Continuing.

... übliche GDB Debug Session ...

(gdb) monitor exit
```

## Die Module – Übersicht



Modulname	zugehörige Dateien	Beschreibung
main	netx10gs_main.c netx10gs_main.h	Hauptmodul. Von hier aus werden die Vorinitialisierungen gestartet, bevor das eigentliche Hauptprogramm beginnt. Es werden die benötigten Threads gestartet und auf ihre Beendigung gewartet. Das Hauptmodul kümmert sich um das Starten aller Aufräumarbeiten, bei Beendigung des Switches.
gdbproto	netx10gs_gdbproto.h	Alle relevanten GDB Makros sind hier enthalten.
globals	netx10gs_globals.h	Variablen, Strukturen und Makros, die von mehreren Modulen benötigt werden, sind hier enthalten.
handlearm	netx10gs_handlearm.c netx10gs_handlearm.h	Funktionen, die die arm steuern, bzw. ihren Zustand auslesen.
handlegdbproto	netx10gs_handlegdbproto.c netx10gs_handlegdbproto.h	Steuert den richtigen Ablauf und Bearbeitung des GDB-Protokolls zwischen den beiden GDBs und OpenOCD.
handlexplic	netx10gs_handlexplic.c netx10gs_handlexplic.h	Funktionen, zur Steuerung des xplic. Bsp. Starten, Stoppen oder

		Register lesen.
helper	netx10gs_helper.c netx10gs_helper.h	Jedwede Art von Hilfsfunktionen, die von den anderen Modulen benötigt werden. Siehe Module – Details
sockets	netx10gs_sockets.c netx10gs_sockets.h	Alles was die Initialisierung und Verbindung der Sockets betrifft. Weiter sind hier die Funktionen, die beim Beenden des Switches die Socketverbindungen wieder aufräumen und belegte Ressourcen freigeben.

## Die Module – Details

### Sockets

Funktionsname	Beschreibung
CleanUp	Beendet alle offenen Socketverbindungen und gibt alle benutzten Ressourcen frei.
CleanUpArm	Beendet die Clientsocket Verbindung des ArmGDBs.
CleanUpXpic	Beendet die Clientsocket Verbindung des XpicGDBs.
InitConnectionArmGdb	Initialisiert alle relevanten Einstellungen und fordert Ressourcen an, die nötig sind um Socketverbindungen aufzubauen, zu nutzen und wieder zu beenden.
InitConnectionXpicGdb	Wie InitConnectionArmGDB nur für den XpicGDB.
ConnectToArmGDB	Thread, der auf eine neue ArmGDB Socketverbindung wartet.
ConnectToXpicGDB	Thread, der auf eine neue XpicGDB Socketverbindung wartet.
ConnectToOpenOCD	Baut eine Socketverbindung mit OpenOCD auf.

### Helper

Funktionsname	Beschreibung
CheckArguments	Überprüft die beim gdb-Switch Start übergebenen Parameter auf ihre Richtigkeit.
InitGlobals	Initialisiert alle globalen Variablen.
Char2Hex	Wandelt einen Char-Wert in den entsprechenden Hex-Wert. Bsp.: 'f' → 0xf (dezimal 15)
.GdbVal2UnsignedInt	Wandelt einen übergebenen GDB-Wert in unsigned int um. Bsp.: 78563412 → 305419896 ( 0x12345678 )
CheckAnswer	Überprüft eine übergebene GDB-Antwort. Mögliche Rückgabewerte: OK, UNKNOWN, ERROR.
CalculateChecksum	Berechnet die GDB Checksumme für einen übergebenen Ausdruck.

BuildGdbMsg	Baut eine GDB Nachricht zusammen. Es kann optional eine Zahl übergeben werden, die entweder im GDB oder im normalen Format eingebaut wird. Weiter kann die zugehörige Checksumme mitberechnet und eingebaut werden.
HandleSocketError	Gibt nur eine Fehlermeldung aus, wenn es sich nicht um WSAEWOULDBLOCK handelt. WSAEWOULDBLOCK tritt auf, wenn nicht-blockierend empfangen wird, aber nichts empfangen wurde.
Receive	Plattformunabhängige Socket Empfangsfunktion.
SendAndReceive	Plattformunabhängige haupt Sende- und Empfangsfunktion.
SendKeepAlive	Sendet keepalive Pakete an OpenOCD, um die Verbindung zwischen gdb-Switch und OpenOCD aufrecht zu halten.

## Handlexplic

Funktionsname	Beschreibung
StepBackPC	Setzt den PC des xplic um eine übergebene Anzahl von Schritten zurück.
ReadXpicRegisters	Liest alle Register des xplic aus: r0 – r7, usr0 – usr4, pc, status.
SetSoftwareBreakpoint	Setzt einen Software Breakpoint.
RemoveSoftwareBreakpoint	Löscht einen software Breakpoint.
SetHardwareBreakpoint	Setzt einen Hardware Breakpoint.
RemoveHardwareBreakpoint	Löscht einen Hardware Breakpoint.
HaltXpic	Hält den xplic an.
ResumeXpic	Startet den xplic.
SinglestepXpic	Konfiguriert den xplic für Singlestep und startet ihn.
GetAndReturnAccessKey	Liest das Access Register aus und schreibt den ausgelesenen Wert wieder rein. Nötig zur Ausführung bestimmter Funktionalitäten. Z.B. für die Aktivierung der xplic Clock.
ActivateXpicClock	Aktiviert die xplic Clock.
ResetXpic	Resettet den xplic.



InitNetx10	Initialisiert den xpic. Dazu wird erst die xpic Clock aktiviert und dann der xpic resettet.
ReadSingleXpicRegister	Liest ein einzelnes xpic Register (r0 – r7, usr0 – usr4, pc, status).
WriteSingleXpicRegister	Schreibt ein einzelnes xpic Register (r0 – r7, usr0 – usr4, pc, status).
CheckForSinglestep	Überprüft das xpic System, ob ein Singlestep ausgeführt wurde.
CheckForBreakpoint	Überprüft das xpic System, ob der xpic in einen Breakpoint gelaufen ist.

## Handlegdbproto

Funktionsname	Beschreibung
PrintConsoleMsg	Nur im Debugmodus. Gibt lesbaren Text auf der Konsole aus, der per \$Oxxxx#xx Befehl empfangen wurde.
GetGdbCmd	Analysiert einen übergebenen GDB-String. Zurückgegeben werden der Befehl und evtl. weitere Parameter wie Adresse und/oder Werte.
HandleGdbProto	Hauptthread. Verantwortlich für den richtigen Ablauf und die korrekte Verarbeitung der ein- und ausgehenden Daten der beiden GDBs.

## Handlearm

Funktionsname	Beschreibung
ControlArm	Startet oder stoppt die arm.
CheckArmRunnig	Überprüft, ob die arm läuft oder steht.