LAUTERBACH
DEVELOPMENT TOOLS

Run Mode Debugging Manual QNX

MANUAL

Run Mode Debugging Manual QNX

Run Mode Debugging Manual QNX

Version 06-Jun-2024

Basic Concepts

For Integrated Run & Stop Mode Debugging, Stop Mode Debugging via the JTAG interface is extended by:

- pdebug as debug agent on the target.
- An ethernet communication between TRACE32 and the debug agent.

pdebug

pdebug is the process level debugger for QNX. It provides access to process-level debugging from a remote host. The pdebug agent may either be included in the image and started in the image start-up script or started later from any available file system.

Switching to Run Mode Debugging

After TRACE32 was started and configured for Stop Mode Debugging switching to Run Mode debugging is performed as follows:

```
SYStem.PORT 10.1.2.99:8000

SYStem.MemAccess QnxMON

Go.MONitor
```

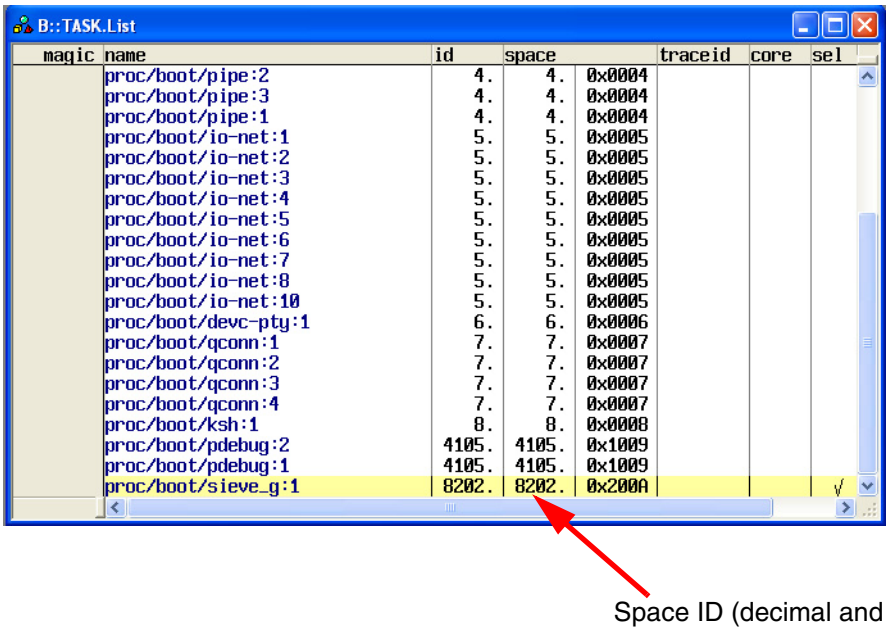
Go.MONitor	Switch to Run Mode Debugging
SYStem.MemAccess QnxMon	Select QNX as Monitor
SYStem.PORT *<ip>*:*<port>*	Configure ethernet communication *<ip>* is the target IP address *<port>* is the used port number

After the communication is configured, debugging can be performed completely via the TRACE32 PowerView GUI.

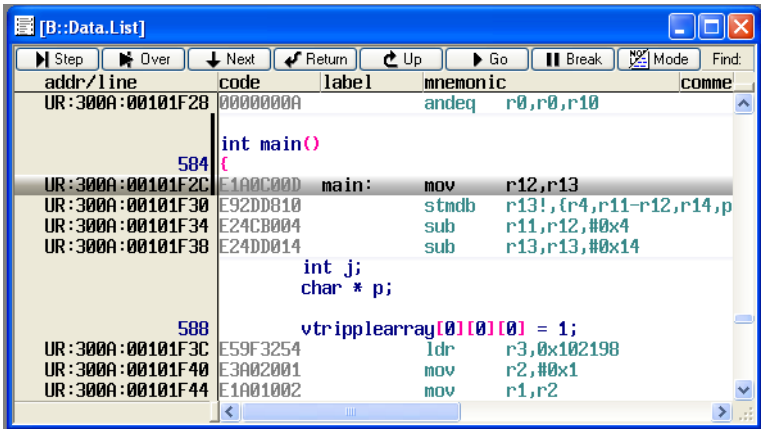
The Space ID for Run Mode Debugging

Processes of QNX may reside virtually on the same addresses. To distinguish those addresses, the debugger uses an additional space ID that specifies to which virtual memory space an address refers. In Run Mode Debugging the space ID is equal to the process ID.

The command **SYStem.Option.MMUSPACES ON** enables the additional space ID.



A source code listing for the process *sieve_g* is displayed as follows:



To debug a process proceed as follows:

1. **Check if the process is already running.**

TASK.List.tasks	List all running processes

```
TASK.List.tasks
```

2. **Load the process for debugging or attach to it.**

TASK.RUN *<process>*	Load *<process>*
TASK.select *<id>*	Attach to the process with PID *<id>*

If the process is not running, the command **TASK.RUN** can be used to load the process for debugging.

```
; Load process sieve from the QNX file system and prepare it for
; debugging

TASK.RUN /proc/boot/sieve
```

If the process is already running, the command **TASK.SELect** can be used to attach to it.

```
TASK.select 8280.
```

3. Load the symbol and debug information for the process.

Data.LOAD.*<file_format> <file> <space_id>***:0 /NoCODE /NoClear /NOREG**

Since processes of QNX may reside virtually on the same addresses, the symbol and debug information has to be loaded for the address space of the process by using the *<space_id>*.

NoCODE - load only symbol information.

NoClear - obtain the symbol information loaded for other processes.

NOREG - avoid any unintended change to the CPU registers.

```
; Data.LOAD.Elf <file> <space_id>:0 /NoCODE /NoClear /NOREG
Data.LOAD.Elf sieve.elf 0x91:0 /NoCODE /NoClear /NOREG

; Stop sieve at main and display source listing
Go main
Data.List
```

TASK.PROC.SPACEID(*<process>***)** This function returns the *<space_id>* of a process. This is required for PRACTICE scripts.

Example for a PRACTICE script:

```
…

TASK.RUN /bin/sieve

Data.LOAD.Elf sieve.elf TASK.PROC.SPACEID(sieve):0 /NoCODE /NoClear /NOREG
```

Quick Start Example for ARM

Integrated Run & Stop Mode Debugging requires that **Stop Mode Debugging** is working properly before Run Mode Debugging can be activated. The following commands represent a basic TRACE32 setup for Stop Mode Debugging. It is assumed that the target setup and QNX booting is performed by the code in the boot FLASH.

```
SYStem.CPU ARM920                       ; Select the target CPU

SYStem.Option.MMUspaces ON              ; Extend logical addresses by
                                        ; space ID

TrOnchip.Set DABORT OFF                 ; Debug mode is not entered at data
                                        ; abort exception used by QNX
                                        ; for page miss!

TrOnchip.Set PABORT OFF                 ; Debug mode is not entered at
                                        ; prefetch abort exception used by
                                        ; QNX for page miss!

TrOnchip.Set UNDEF OFF                  ; Debug mode is not entered at an
                                        ; UNDEF instruction may be used by
                                        ; QNX for FPU detection

SYStem.MemAccess QnxMON                 ; Set the monitor type

SYStem.Mode Attach                      ; Establish the communication
                                        ; between the debugger and the CPU

Data.LOAD.Elf procnto.sym /NoCODE       ; Load the kernel symbols

TASK.CONFIG qnx                         ; Enable the TRACE32 QNX
                                        ; awareness

MMU.FORMAT QNX                          ; Define an MMU page table format
                                        ; for QNX

TRANSlation.COMMON 0xC0000000--         ; Upper memory pages are valid for
0xffffffff                              ; all space IDs

TRANSlation.on                          ; Switch debugger MMU to ON
```

The target setup and the preparations for QNX debugging might be more complex for your system. It is strongly recommended to refer to **"Arm Debugger"** (debugger_arm.pdf) and for details.

To configure Run Mode Debugging:

1. Make sure pdebug is running on your target.

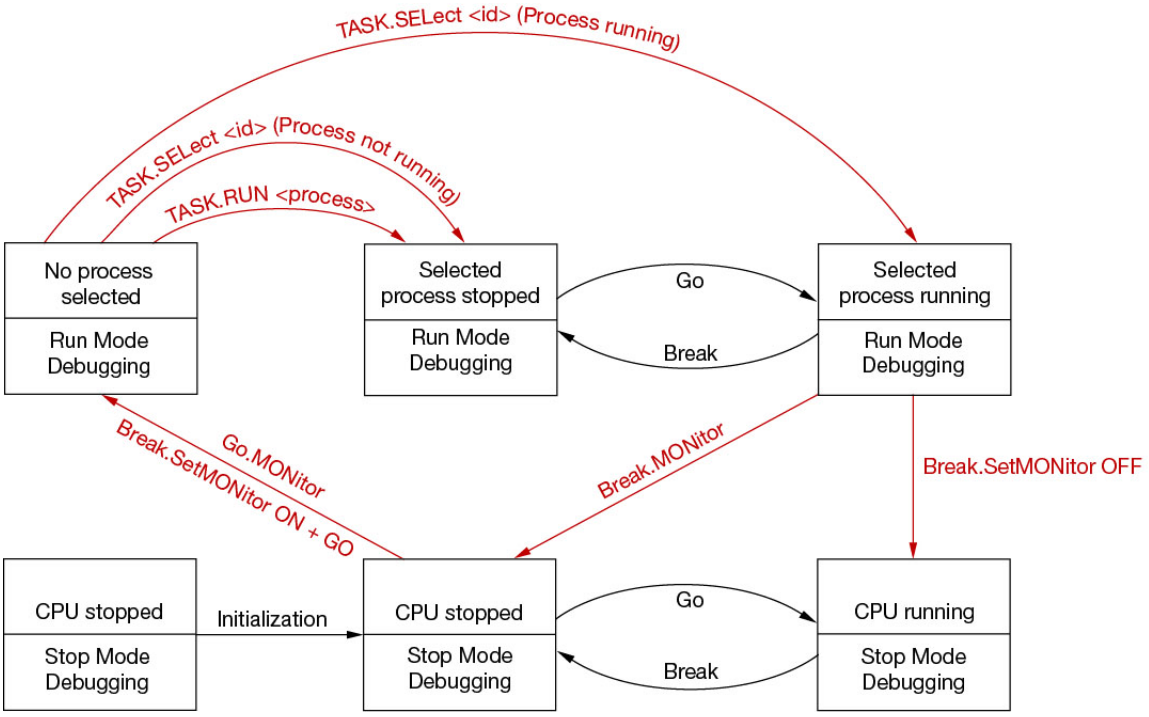
2. Configure your ethernet connection:

```
SYStem.PORT <ip>:<port>      ; target ip and port number
```

3. Then continue to enter the following commands in TRACE32:

```
Go.MONitor                   ; Switch to Run Mode Debugging
```

Switching between Run & Stop Mode Debugging



The graphic above shows a simple schema of the switching between Run Mode Debugging and Stop Mode Debugging. Not all transitions are covered.

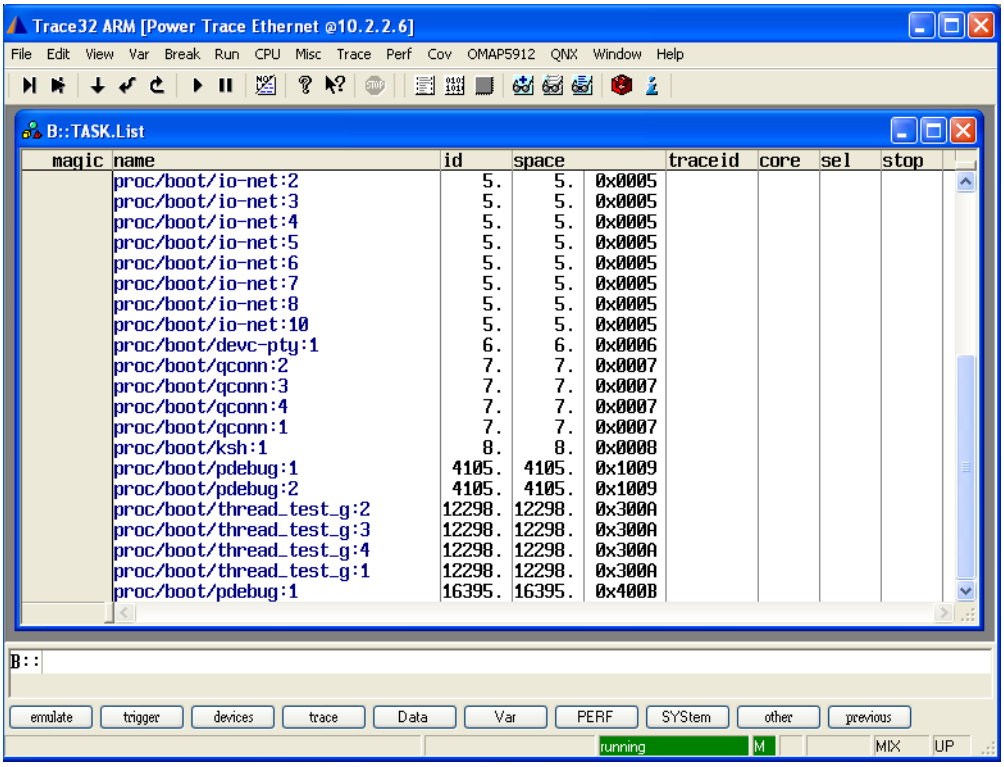
The following commands are used to switch between Run & Stop Mode Debugging:

Break.MONitor	Switch to Stop Mode Debugging and stop the program execution on CPU.
Break.SetMONitor OFF	Switch to Stop Mode Debugging. If the selected process was running or no process was selected, the CPU stays running in Stop Mode. If the selected process was stopped, the CPU is stopped in Stop Mode.
Break.SetMONitor ON	Switch to Run Mode Debugging with the next **Go**. In Run Mode Debugging no process is selected for debugging.
Go.MONitor	If the CPU is stopped, the program execution is started. Switch to Run Mode Debugging. In Run Mode Debugging no process is selected for debugging.

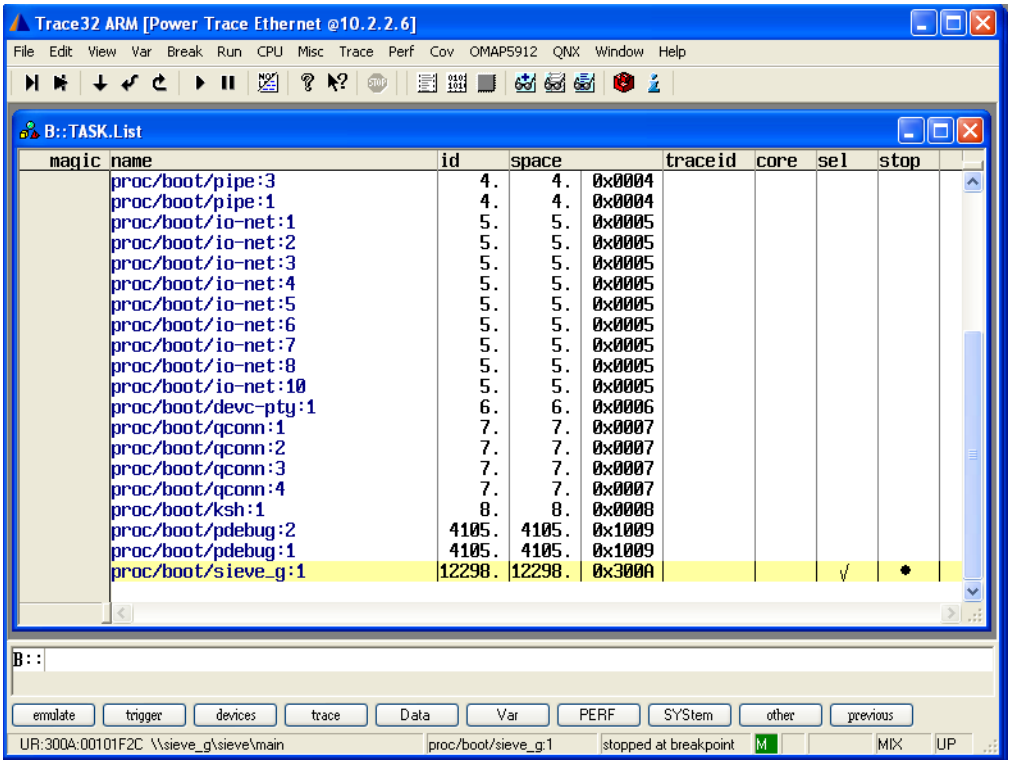
If Run Mode Debugging is active, a **green M** is displayed in the state line of TRACE32.

The following states are possible in Run Mode Debugging:

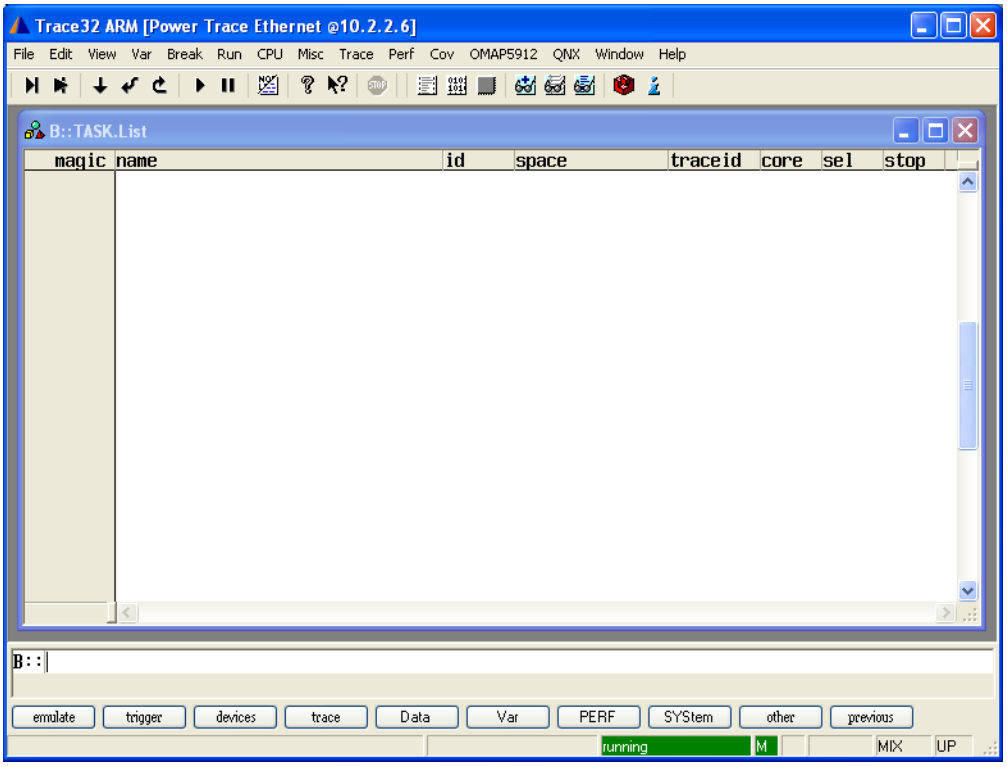
1. Run Mode Debugging active (**green M**), no process selected (see **TASK.List.tasks**).



2. Run Mode Debugging active (**green M**), selected process (sieve_g) stopped.



3. Run Mode Debugging active (**green M**), selected process (sieve_g) running. When the selected process is running, the task list cannot be displayed.



TASK.List.tasks	List all running processes
TASK.RUN *<process>*	Load a process for debugging

TASK.DETACH *<id>*	Detach from the process
TASK.select *<id>*	Select a process and attach to it

TASK.KILL *<id>*	Request pdebug agent to end the process

Only processes that have been started with a **TASK.RUN** or that have been attached with **TASK.SELect** can be killed..

TASK.COPYDOWN *<src> <dest>*	Copy a file from the host into the target
TASK.COPYUP *<src> <dest>*	Copy a file from the target into the host

Breakpoint Conventions

For Integrated Run & Stop Mode Debugging please keep the following breakpoint convention:

- Use on-chip breakpoints for Stop Mode Debugging

  If an on-chip breakpoint is hit in Run Mode Debugging, the CPU is stopped in Stop Mode debugging (only for ARM).

- Use software breakpoints for Run Mode Debugging

Examples for Stop Mode Debugging:

```
; Break.Set <space_id>:<address> /Program /Onchip
Break.Set 0x0:0x4578 /Program /Onchip

Break.Set error /Program /Onchip
```

Examples for Run Mode Debugging:

```
; Break.Set <space_id>:<address> /Program /SOFT
Break.Set 0x2bc:0x0xd0065789 /Program /SOFT

Break.Set 0x2bc:main /Program /SOFT
```