General Description

The Watchdog eCos device driver controls the Watchdog IP core and presents a standard interface to the higher layers of eCos. This allows the Watchdog to be accessed with the standard watchdog C functions supported by eCos.

If the Watchdog eCos device driver is to be used on an Evaluator-7T board with an E7T-DBoard fitted, the eCos repository must first be updated to support accesses to the FPGA or the board will freeze when the FPGA is accessed. This update is available from http://www.sweeneydesign.co.uk/downloads.htm

Although this datasheet refers to the Evaluator-7T board with an E7T-DBoard fitted, the Watchdog eCos device driver, without modifications, should be suitable for any eCos platform with a Watchdog core included.

Device Driver Files

The Watchdog eCos device driver consists of 1 file:

watchdog.epk – Watchdog Device Driver eCos Package Distribution File.

Installation

The installation procedure updates the eCos repository to add support for the Watchdog IP core – this only needs to be done once for each eCos repository.

Updating The Repository

This adds the Watchdog Device Driver Package to the eCos repository. The files contained in the eCos Package Distribution File need to be added to the repository. This is done as follows:

For graphical configuration tool users:

1. Start the graphical configuration tool.
2. From the menu select 'Tools -> Administration...'
3. When asked 'This command will close the current document. Do you wish to continue?' click Yes.
4. This will start the eCos Package Administration Tool. Click 'Add...' on this tool.
5. The 'Open eCos Package Files' window will open. Navigate to the watchdog.epk file and click Open.
6. Read the licence agreement and click Yes to accept it.
7. The Watchdog device driver will be added to the eCos repository.
8. Click 'Close' on the eCos Package Administration Tool and then close the configuration tool.
9. Updating The Repository is complete.

For command line configuration tool users:

1. Copy the file, watchdog.epk to the packages/ directory in the eCos repository.
2. At a Bash prompt, cd to the packages/ directory in the eCos repository.
3. At a Bash promp type: **`sh ecosadmin.tcl add watchdog.epk <RETURN>`**
4. Read the licence agreement and press 'y' <return> to accept it.
5. Updating The Repository is complete.

Building eCos With The Watchdog Device Driver

Once the device driver has been installed, eCos may be built with support for the Watchdog IP core. This is achieved as described below.

For graphical configuration tool users:

1. Start the graphical configuration tool.
2. From the menu select 'Build -> Templates...'
3. On the 'Templates' window, select: Hardware 'ARM Evaluator7T board (AEB-2)': Packages 'default'.
4. Click 'OK' on the 'Templates' window. The 'Templates' window will close.
5. Next add the Watchdog device driver. From the menu select 'Build -> Packages...'
6. From the available packages click on 'Watchdog driver for watchdog IP core' and click 'Add >>'.
7. The 'Watchdog driver for watchdog IP core' will move from 'Available packages' to 'Use these packages'.
8. From the available packages click on 'Watchdog IO device' and click 'Add >>'.
9. The 'Watchdog IO device' will move to 'Use these packages'.
10. Click 'OK' on the 'Packages' window. The 'Packages' window will close.
11. On the main configuration tool window, set up the Watchdog device driver configuration options as required. For the pre-built support FPGA image, the default configuration options are okay.
12. From the menu select 'File -> Save' and save the configuration.
13. Build the eCos library by selecting 'Build -> Library' from the menu.

For command line configuration tool users:

1. At a Bash prompt, create a build directory and use the cd command to move to it.
2. Configure eCos for the E7T. At a Bash prompt type: **ecosconfig new e7t default <RETURN>**
3. At a Bash prompt type: **ecosconfig add CYGPKG_DEVICES_WATCHDOG_IPCORES_WATCHDOG <RETURN>**
4. This will add the Watchdog IP core device driver.
5. At a Bash prompt type: **ecosconfig add CYGPKG_IO_WATCHDOG <RETURN>**
6. This will add the generic watchdog driver.
7. Edit the ecos.ecc file to set up the required configuration options for the Watchdog device driver. For the pre-built Support FPGA image, the default configuration options are okay.
8. Create a build tree. At a Bash prompt type: **ecosconfig tree <RETURN>**
9. Build eCos. At a Bash prompt type: **make <RETURN>**

Configuration Options

Once the device driver has been installed, the following configuration options become available. The default values for these options are suitable for use with the pre-built Support FPGA image. These options may be modified using the graphical configuration tool, or by editing the .ecc configuration file (for developers using the command line configuration tool).

cdl_option	CYGDAT_DEVICES_WATCHDOG_IPCORES_WATCHDOG_BASE_ADDR
display	Base address of Watchdog timer device
type	data
default value	0x07A00100
description	This option sets the base address of the Watchdog Timer.

cdl_option	CYGDAT_DEVICES_WATCHDOG_IPCORES_WATCHDOG_STEP
display	Address step between registers in Watchdog
type	data [2, 4]
default value	4
description	This option specifies how much to increment the address to step between Watchdog registers. Set to 2 for 16-bit wide data bus and 4 for 32-bit wide data bus.

cdl_option	CYGDAT_DEVICES_WATCHDOG_IPCORES_WATCHDOG_RESET_PULSE_LENGTH
display	Watchdog reset pulse length in ms
type	data [1 to 15]
default value	2
description	This option sets how long the reset pulse generated by the Watchdog Timer will be.

cdl_option	CYGDAT_DEVICES_WATCHDOG_IPCORES_WATCHDOG_TIMEOUT_PERIOD
display	Watchdog timeout period in ms
type	data [1 to 4094]
default value	600
description	This option sets the minimum time the Watchdog timer will wait without receiving a watchdog reset register write before resetting the processor.

eCos Watchdog API

To use the eCos Watchdog API, the following header file must be included in the application:

#include <cyg/io/watchdog.h>

Controlling the Watchdog IP core with the device driver is very simple, and is achieved by calling just 2 functions. These functions are:

void watchdog_start(void)

This function should be called once at the start of the application to start the Watchdog.

void watchdog_reset(void)

Once the watchdog has been started, this function must be called repeatedly within the watchdog timeout period to prevent the Watchdog core from resetting the processor.

Example Application Code

The C code below is a very simple example of using the Watchdog. This code will work with an eCos library built with the 'default' template.

```
#include <cyg/io/watchdog.h>  /* watchdog_start() & watchdog_reset() */

int main()
{
    /* Start the watchdog timer */
    watchdog_start();

    /* Main application loop */
    while(1)
    {
        /* Reset the watchdog timer */
        watchdog_reset();

        /* Insert application code that should not lock up here */
        /* */
        /* */
    }
}
```

Sweeney Design Ltd

40 Hopkins Close
Cambridge
CB4 1FD
United Kingdom

Tel: +44 (0) 7977 129406
Fax: +44 (0) 7977 045842
E-mail: cores@sweeneydesign.co.uk
URL: www.sweeneydesign.co.uk