

General Description

The StdUart eCos device driver controls the StdUart IP core and presents a standard interface to the higher layers of eCos. This allows the StdUart to be accessed in exactly the same way as any other serial interface supported by eCos. The StdUart eCos device driver is actually just a simple wrapper for the generic 16x5x device driver that is contained in the standard eCos distribution.

If the StdUart 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 StdUart eCos device driver, without modifications, should be suitable for any eCos platform with a StdUart core included.

Device Driver Files

The StdUart eCos device driver consists of 2 files:

stduart.epk – StdUart Device Driver eCos Package Distribution File.
stduart.pat – Patch for eCos repository to allow e7t eCos port to use the 16x5x generic driver.

Installation

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

Updating The Repository – Step 1

The first step required is to add the StdUart Device Driver Package to the eCos repository. Firstly, 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 stduart.epk file and click Open.
6. Read the licence agreement and click Yes to accept it.
7. The StdUart 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 - Step 1 is complete.

For command line configuration tool users:

1. Copy the file, stduart.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 prompt type: **sh ecosadmin.tcl add stduart.epk <RETURN>**
4. Read the licence agreement and press 'y' <return> to accept it.
5. Updating The Repository - Step 1 is complete.

Updating The Repository – Step 2

The second step is to apply the patch, stduart.pat. This patch is required to allow the 16x5x generic device driver to be used by the e7t eCos port. This patch is applied at a Bash command prompt:

1. Copy the file, stduart.pat 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 prompt type: **patch -Ni stduart.pat <RETURN>**
4. The file ecos.db will be patched.
5. Updating The Repository - Step 2 is complete.

Building eCos With The StdUart Device Driver

Once the device driver has been installed, eCos may be built with support for the StdUart 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 StdUart device driver. From the menu select 'Build -> Packages...'
6. From the available packages click on 'StdUart serial device driver' and click 'Add >>'.
7. The 'StdUart serial device driver' will move from 'Available packages' to 'Use these packages'.
8. From the available packages click on '16x5x compatible serial device drivers' and click 'Add >>'.
9. The '16x5x compatible serial device drivers' 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, enable the 'Hardware serial device drivers' option to include the serial device drivers in the build.
12. If you are using one of the E7T serial ports to connect Insight/GDB to the board, you will need to disable the serial driver for this port. For port 1, disable 'ARM E7T serial port 1 driver'.
13. On the main configuration tool window, set up the StdUart device driver configuration options as required. For the pre-built StdUart FPGA image, the default configuration options are okay.
14. From the menu select 'File -> Save' and save the configuration.
15. 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_IO_SERIAL_IPCORES_STDUART <RETURN>**
4. This will add the StdUart device driver.
5. At a Bash prompt type: **ecosconfig add CYGPKG_IO_SERIAL_GENERIC_16X5X <RETURN>**
6. This will add the generic 16x5x driver that the StdUart driver uses.
7. Next, modify the configuration as required by editing the ecos.ecc file.
8. Enable the 'Hardware serial device drivers' option by changing the CYGPKG_IO_SERIAL_DEVICES cdl_component in the ecos.ecc file. In this component change the line **# user_value 0** to **'user_value 1'** to enable this option.
9. If you are using one of the E7T serial ports to connect Insight/GDB to the board, you will need to disable the serial driver for this port. For port 1, change the CYGPKG_IO_SERIAL_ARM_E7T_SERIAL1 cdl_component in the ecos.ecc file. In this component change the line **# user_value 1** to **'user_value 0'** to disable the driver for serial port 1.
10. Edit the ecos.ecc file to set up the required configuration options for the StdUart device driver. For the pre-built StdUart FPGA image, the default configuration options are okay.
11. Create a build tree. At a Bash prompt type: **ecosconfig tree <RETURN>**
12. 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 StdUart 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_component	CYGPRI_IO_SERIAL_GENERIC_16X5X_STEP
display	Address step between registers in StdUart
type	data [1, 2 or 4]
default value	4
description	This option specifies how much to increment the address to step between the 8-bit StdUart registers. Set to 1 for 8-bit wide data bus, 2 for 16-bit wide data bus and 4 for 32-bit wide data bus.

cdl_component	CYGPKG_IO_SERIAL_IPCORES_STDUART_CLOCK_FREQ
display	StdUart input clock frequency
type	data
default value	40000000
description	This option specifies the frequency (in Hz) of the clock input to the StdUart.

cdl_component	CYGPKG_IO_SERIAL_IPCORES_STDUART_SERIAL0
display	StdUart serial port driver
type	Boolean
default value	1
description	This option includes the serial device driver for the StdUart port.

cdl_option	CYGDAT_IO_SERIAL_IPCORES_STDUART_BASE_ADDR
display	Base address of StdUart
type	data
default value	0x07A00000
description	This option sets the base address of the StdUart.

cdl_option	CYGDAT_IO_SERIAL_IPCORES_STDUART_SERIAL0_NAME
display	Device name for StdUart serial port driver
type	data
default value	"/dev/stduart"
description	This option specifies the name of the serial device or the StdUart port.

cdl_option	CYGNUM_IO_SERIAL_IPCORES_STDUART_SERIAL0_BAUD
display	Baud rate for the StdUart serial port driver
type	data [50, 75, 110, "134_5", 150, 200, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 14400, 19200, 38400, 57600, 115200, 230400]
default value	38400
description	This option specifies the default baud rate (speed) for the StdUart port."

cdl_option	CYGNUM_IO_SERIAL_IPCORES_STDUART_SERIAL0_BUFSIZE
display	Buffer size for the StdUart serial port driver
type	data [0 to 8192]
default value	128
description	This option specifies the size of the internal buffers used for the StdUart port.

Example Application Code

The C code below is a very simple example of accessing the StdUart. This code will work with an eCos library built with the 'default' template. To see the results a terminal emulator should be connected to the StdUart, set to 38400 baud, 8N1 data with no flow control.

```
#include <stdio.h>

int main()
{
    /* Define a pointer to a file/stream */
    FILE *fptr;

    char name[50];

    /* Open stream to UART */
    printf("Trying to open /dev/stduart\n");
    if ((fptr=fopen("/dev/stduart","r+"))==NULL)
    {
        printf("Error in opening /dev/stduart\n");
        exit(1);
    }

    else
    {
        printf("/dev/stduart opened OK\n");
    }

    /* Write some data to the StdUart */
    fprintf(fptr, "\r\n\r\nHello World!\r\n");

    /* Get some data from the UART */
    fprintf(fptr, "What is your name?\r\n");
    fscanf(fptr, "%s", name);
    fprintf(fptr, "Hello, %s!\r\n", name);

    return(0);
}
```

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