Apex Embedded Systems Demo GUI Application Quick Start
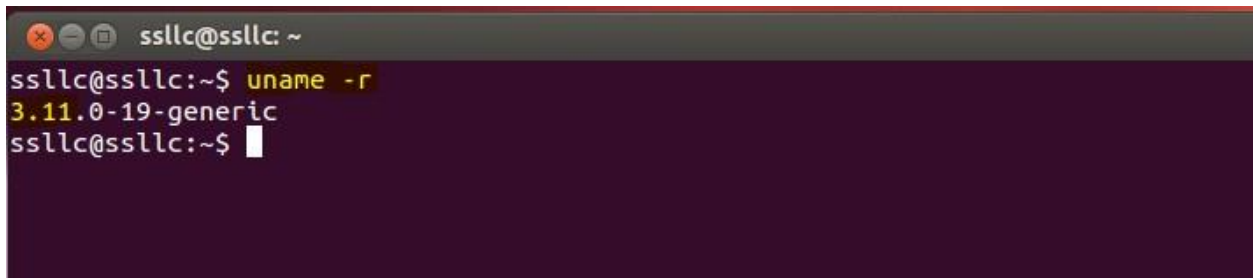
Contents

Prerequisite

In order to utilize the C# Demo GUI Application using Mono, you must be running kernel version 3.2 or later. You can verify your current kernel version by entering the following command

$ uname –r

Scope

This document applies to the aul_tree project being built as a shared resource.

The objective in doing so is to consume the library project from within the accompanying C# executable when run within Mono.

Overview

This application demonstrates using C# and Mono to build a graphical user interface to allow interaction and manipulation of an STX104 PC/104 card.

It is intended to be used to demonstrate utilization of the AUL library interface.
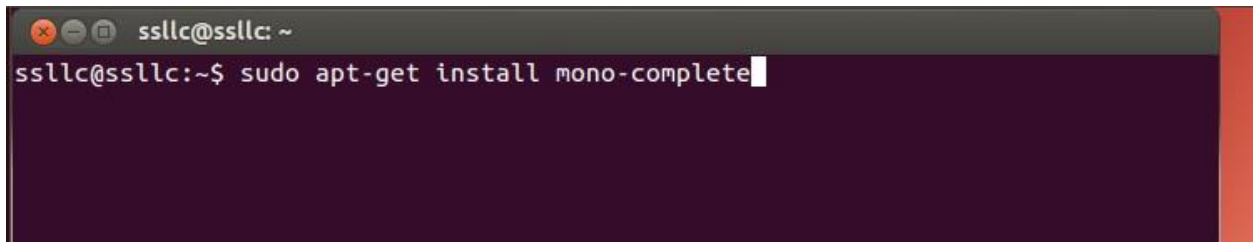
It is not intended to be used for production.

Setup / Installation

-To run the Demo GUI Application for the first time a few setup steps are required. Please see Pre-requisite section before proceeding.

1. Mono Complete must be installed. This is required to run the demo application. This can be accomplished by running:
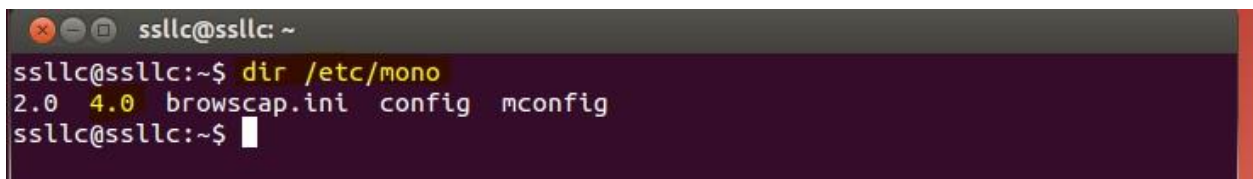
$ sudo apt-get install mono-complete

--Normal installation steps apply. Answer yes when asked if you want to continue. Enter root password to allow install when prompted.

```
⊗⊖▢  ssllc@ssllc: ~
ssllc@ssllc:~$ sudo apt-get install mono-complete▮
```

1.b To verify correct Mono install you need to verify that Mono 4.0 is installed. To do so enter the following command and verify the mono directory contains a 4.0 directory as highlighted below.

$ dir /etc/mono

```
⊗⊖▢  ssllc@ssllc: ~
ssllc@ssllc:~$ dir /etc/mono
2.0  4.0  browscap.ini  config  mconfig
ssllc@ssllc:~$ ▮
```

2. Install the correct library. Scripts have been provided to aid in this step and are located in the _Demo_Csharp directory.

There are two options for the library build. The simulator project that does not require hardware (Option A) and the full library (Option B)

2.(Option A) To use the simulator project use the build_aul_use_simulator script with the following command:

$ sh build_aul_use_simulator

-- Root password will be required to copy library to /usr/lib

2.(Option B)

To install the full library (for use with STX104 driver and hardware installed) use the build_aul_use_driver script with the following command:

sh build_aul_use_driver

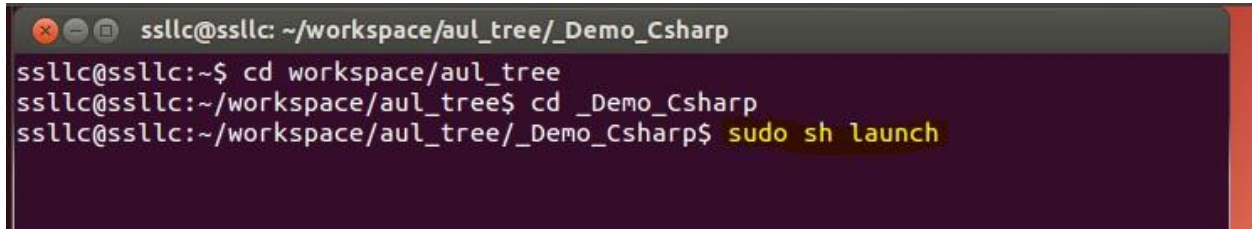-- Root password will be required to copy library to /usr/lib

Run The Demo
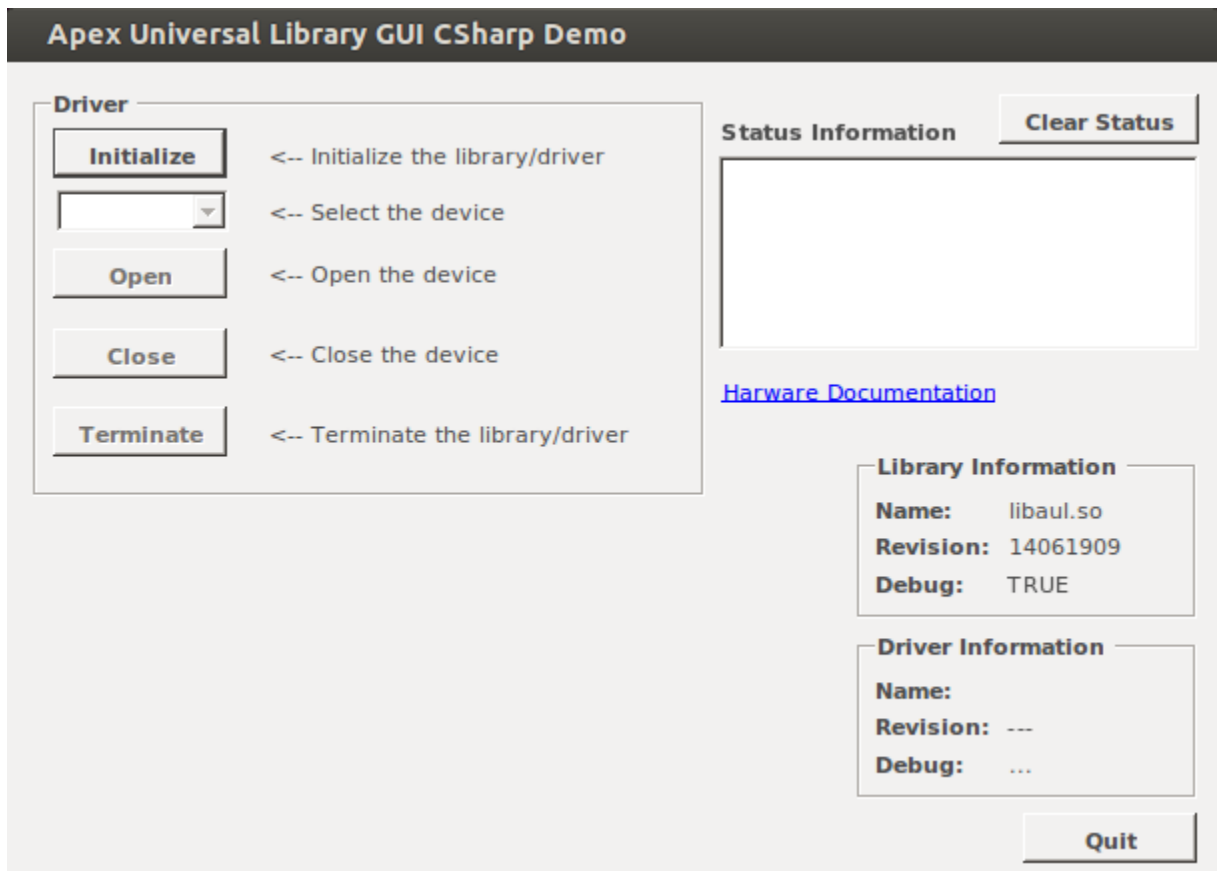A launch script has been provided to aid in setting Mono and starting the application.

To run the application, navigate to the _Demo_Csharp directory and run the following command:

$ sudo sh launch


```
ssllc@ssllc: ~/workspace/aul_tree/_Demo_Csharp
ssllc@ssllc:~$ cd workspace/aul_tree
ssllc@ssllc:~/workspace/aul_tree$ cd _Demo_Csharp
ssllc@ssllc:~/workspace/aul_tree/_Demo_Csharp$ sudo sh launch
```

Doing so will start Mono and the Demo GUI Application.
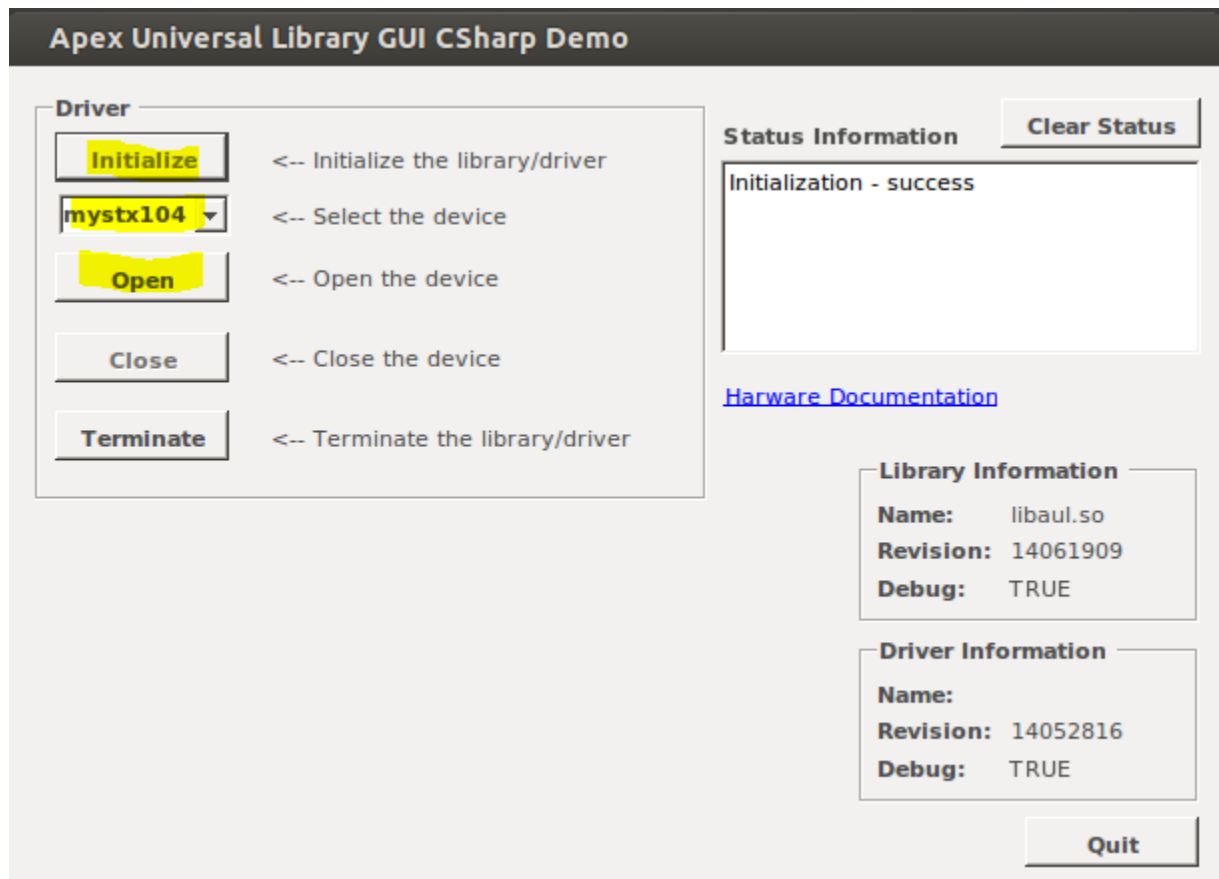

Demo Walk Through

This document provides a simple walk through to configure and demonstrate triggering using DOUT0 within the STX104 Demo Application. Following these steps will allow you to configure the analog input, configure and enable a triggering example, trip the trigger we set up and read and display data from the board.

NOTE: This walk through only works with the full library and hardware installed.

1. Connect digital output 0 to digital input 0 on your STX104 hardware. Done so that we can change the state of DIN0 which is then used to drive triggering state.

2. Run demo GUI application. Initialize, Select Your Device and Open drive. Verification of success is displayed in the Status Information window. NOTICE the Configuration and Data Flow buttons as they will be discussed later.

3. Select the General button under Configuration above. The jumper configuration should match the below screenshot. Communication with the hardware can be verified by a valid Firmware Register Number (i.e. 0x100A Highlighted below). Number such as 0x0000 or 0xFFFF indicates no communications.

STX104: General Information – Configuration

Jumpers   (STX104_GEN_JMP_TYPE)

- ■ M0_mode   (Note: FIFO available, AI-frames)
- ■ M1_mode
- ■ M2_dacs_16bits   (Note: 16-bit resolution)
- ☐ M3_moving_average_filter   (Note: disabled)
- ☐ M4_address_10bit_mode   (Note: 16-bit address decod
- ☐ DMA_ch3   (Note: DMA Ch 1 (*)Deprecated)
- ☐ CLK_1mhz   (Note: 10 MHz to 8254 counter)
- ☐ J9_ai_unipolar   (Note: Analog inputs bipolar)
- ■ J8_ai_differential   (Note: Analog inputs single ended)

■ = Jumper Installed

(*) Few CPUs support DMA and it is very slow.

Firmware	aul_lib_dll.dll	Auldrv.sys
REG: 0x100A ROM: 120328	335810581	335810576

[Close]

4. Select the Analog Inputs button under Configuration. Set the highlighted options to match the screen shot below and select the Apply button.

5. Select the Triggering button under Configuration. Set options as highlighted below and click Apply

6. You are now ready to run the triggering / sampling test. Under Data Flow select the Analog Inputs Display and Digital I/O buttons. You should see the two forms below. Select DO button on the Digital I/O form. If wired correctly Din0 should turn on. These conditions will set Triggering to active and data will be sampled at the 1x per second set

in the analog input configuration. Select the Pause Readout button to pause the display of analog input data. You will notice the data is being collected and added to the FIFO. Select the Resume Readout button (Pause Readout in screen shot) to start display of analog input data. The FIFO count will remain constant until D0 is turned off ending the trigger condition. At this point, the remaining items in the FIFO will be processed.