ADC Module Monitor

The Talon-DX on-board ADC module is the Maxim Integrated MAX11615 low-power 8 channel I2C 12-bit ADC in ultra-small package device.

The I2C mapping for this device is:

busID = **0**
i2cAddress = **0x33 (51)**
mod = **i2c**
compat = **maxim,max11615**
name = **max11615**
hw = **iio:device0**

Control for the ADC module is via the ADC driver. The ADC driver is configured within the device tree. The device tree file for the ADC device contains the following information:

```
adc@33 {
     compatible = "maxim,max11615";
     reg = <0x33>;
     #address-cells = <1>;
     #size-cells = <0>;
};
```

As identified by the devicetree entry, a compatible linux driver is provided via the ***max1363.c*** driver. This driver exposes all relevant and required attributes for control of the module via the sysfs mechanism. The specific compatibility string ***max11615*** is listed within the driver explicitly. The driver is located within the linux kernel source code at

~/drivers/iio/adc/max1363.c

Upon linux boot, the sysfs mount point for the module becomes:

/sys/bus/i2c/devices/0-0033

Below this directory, the device attributes are organized under a directory entitled ***iio:device#***, where *#* is the enumeration for the specific device. Because many device drivers may share this enumeration construct (iio:device) it should be noted that this prefix is appended at mount time (linux boot time) with an enumeration. The enumeration (and subsequently the mount point for the device attributes) may differ from one boot to the next. Asa result, the TalonDXBSMCBase class contains a function called ***get_hw_path*** which given the appropriate filter (in this case, iio:device) detects the enumeration, and returns the correct path for access to attributes for the current boot configuration. There is no

indication that the enumeration of the device path will change after boot time.

Since the driver exposes attributes for a range of Maxim ADC devices, significantly more attributes are exposed than are utilized by the specific ***max11615*** device. Many of these attributes have been coded into the TANGO device server for the ADC device, however are not utilized at this time and remain dormant.

The physical device is capable of receiving 8 distinct ADC signals. In the Talon-DX implementation, only channels 0 through 5 (six channels) are connected to external board input. As such, channels 6 and 7 are handled by the TANGO device server, but ignored by the GUI.

The GUI for this device is – at this time – a simplified display of the raw ADC value for each channel. This value is the raw voltage received by the device, scaled by the provided scaling factor.

TANGO DEVICE SERVER

Base Class

The TANGO device server is built upon the TalonDXBSMCBase class. This class provides basic information for the device under control, which is stored in the TANGO database as device properties. The list of device properties in the base class are as follows:

- **busID** – the I2C bus address for the HW device
- **i2cAddress** – the I2C device address for the HW device
- **mod** – the device module name (as referenced in the device tree file)
- **compat** – the device compatibility string as detected by the Linux driver (if available)
- **name** – the device name (not required)
- **deviceID** – internal index for device (if multiple devices)
- **hwFilter** – a string used to construct the *hwPath* variable within the TANGO device server.
- **HwPrefix** – a string used to construct the *hwPath* variable within the TANGO device server.

One additional device property in the base class is ***hwPath***. This property is constructed within the *init* method of the TANGO device server. This is the path where device attributes are exposed to the user space within the HPS Linux distribution.

A suite of TANGO device commands accompany these attributes within the base class, providing access to the values to any software able to access the device.

Device Class

The TANGO device server class responsible for monitoring the ADC device is named **TalonDXBSMCADC**. The class provides monitoring of the following attributes:

- **inputVoltageRaw** – an 8 element array of DevULong storing raw voltage measurements from 8 different channels provided by the physical ADC device. This attribute is READ ONLY and polled every 3s.
- **InputVoltageScale** – a DevDouble scalar value storing the voltage scaling factor for all device input channels. This attribute is READ ONLY.

Attributes

Each active ADC channel has an attribute file named:

in_voltage#_raw

where the # denotes the channel index (0-7).

One further attribute is the voltage scale value:

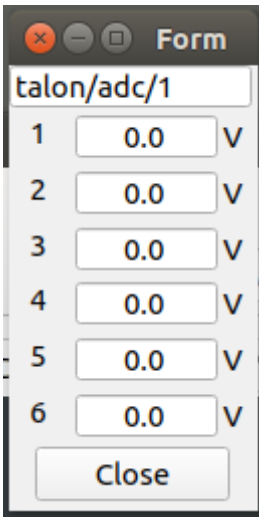
in_voltage_scale

These attribute files are updated when read by the TANGO device server via the linux driver. The value contained within the ***in_voltage#_raw*** file is an integer representing the raw voltage value for the channel. The raw voltage is provided in mV. The value contained within the ***in_voltage_scale*** file is a floating value representing the scale factor to be applied to the raw voltage.

The raw voltage values for all channels (0-7) as well as the voltage scale value are polled by the TANGO device server every 3s.

The GUI subscribes to the raw voltage periodic event generated by the TANGO device server. When an event is detected, the GUI reads all valid raw voltage values for channels 0-5. The GUI then reads the current voltage scale factor. The scale factor is applied to the raw value, and the new value is displayed within the GUI, expressed in V.

Note that within the GUI, numbering of the channels begins at 1. This may be changed as required in future versions.



The GUI is fairly basic at this point, as specific functionality pertaining to the ADC values for each channel have yet to be identified.