U-Series Support Library

This library provides support for accessing the available functionality of the XMOS U-Series devices.

Features

- ADC support.

Software version and dependencies

This document pertains to version 2.0.0 of this library. It is known to work on version 14.0.0 of the xTIMEcomposer tools suite, it may work on other versions.

This library depends on the following other libraries:

- lib_logging (>=2.0.0)
- lib_xassert (>=2.0.0)

1 A-Series ADC API

Type	**adc_bits_per_sample_t**
Description	Valid bits_per_sample values (8, 16 or 32).
Values	ADC_8_BPS Samples will be truncated to 8 bits. ADC_16_BPS Samples will be placed in the MSB 12 bits of the half word. ADC_32_BPS Samples will be placed in the MSB 12 bits of the word.

Type	**adc_config_t**
Description	Configuration structure for ADCs:.

Fields	`char input_enable`
	`adc_bits_per_sample_t bits_per_sample`
	`unsigned int samples_per_packet`
	`int calibration_mode`

Parameters	`input_enable` An array ints to determine which inputs are active. Each non-zero input will be enabled. `bits_per_sample` Select how many bits to sample per ADC. `samples_per_packet` Number of samples per packet. Must be >0 and <=XS1_MAX_SAMPLES_PER_PACKET. `calibration_mode` When set the ADCs will sample a 0.8V reference rather than the external voltage.

Function	**adc_enable**
Description	Configure and enable the requested ADCs. Will also perform the calibration pulses so that the ADCs are ready to provide data. adc_enable() also checks that the configuration is valid and will raise a trap if attempting to incorrectly configure the ADCs.

Type	`void adc_enable(tileref periph_tile,`
	`                chanend adc_chan,`
	`                out port trigger_port,`
	`                const_adc_config_ref_t config)`

Parameters	`periph_tile` The identifier of the tile containing the ADCs `adc_chan` The chanend to which all ADC samples will be sent. `trigger_port` The port connected to the ADC trigger pin. `config` The configuration to be used.
Returns	ADC_OK on success and one of the return codes in adc_return_t on an error.

Function	**adc_disable_all**
Description	Disable all of the ADCs.
Type	`void adc_disable_all(tileref periph_tile)`

Function	**adc_trigger**
Description	Causes the ADC to take one sample. This function is intended to be used with adc_read(). If used with adc_read_packet() then this function must be called enough times to ensure that an entire data packet will be available before the adc_read_packet() is called.
Type	`void adc_trigger(out port trigger_port)`
Parameters	`trigger_port` The port connected to the ADC trigger pin.

Function	**adc_trigger_packet**
Description	Trigger the ADC enough times to complete a packet.

Type	`void`
	`adc_trigger_packet(out port trigger_port,`
	`                   const_adc_config_ref_t config)`

Parameters	`trigger_port` The port connected to the ADC trigger pin. `config` The ADC ocnfiguration.

Function	**adc_read**
Description	A selectable function to read an ADC sample from the chanend. Any control tokens due to packetization will be discarded silently. Note that the adc_trigger function must have been called before this function will return any data. Note that the configuration must be the same as that used when enabling the ADCs.

Type	`void adc_read(chanend adc_chan,`
	`              const_adc_config_ref_t config,`
	`              unsigned int &data)`

Parameters	`adc_chan` The chanend to which all ADC samples will be sent. `config` The ADC configuration. `data` The word to place the data in.

Function	**adc_read_packet**
Description	A selectable function to read a packet of ADC samples from the chanend. Note that the adc_trigger_packet function must have been called before this function will return any data. Note that the configuration must be the same as that used when enabling the ADCs.

Type	`void adc_read_packet(chanend adc_chan,`
	`                     const_adc_config_ref_t config,`
	`                     unsigned int data[])`

Continued on next page

Parameters	`adc_chan`	The chanend to which all ADC samples will be sent.
	`config`	The ADC configuration.
	`data`	The buffer to place the returned data in. Each sample will be placed in a separate word. The buffer must be big enough to store all the data that will be read (samples_per_packet words).

APPENDIX A - Known Issues

There are no known issues with this library.

APPENDIX B - U-Series support library change log