Capabilities Registry Information

TCU Service

Registry Entry	Туре	Description
tcu_TmrSrvAvailable	INTEGER	Availability of the Accurate Timer service (boolean value)
tcu_CaptSrvAvailable	INTEGER	Availability of the Event Capture service (boolean value)
tcu_DetectSrvAvailable	INTEGER	vailability of the Event Detection service (boolean value)
tcu_EvPinsNb	INTEGER	Number of pins usable to monitor events with the Capture & Detection services
tcu_TimersNb	INTEGER	Maximum number of Accurate Timer service instances which can be running at the same time
tcu_TimerBoundaries	DATA	Minimum & maximum duration values which can be used for the adl_tcuTimerBoundaries_t structure format.
tcu_TimerTick	DATA	Timer resolution used by the Accurate Timer Service, using the adl_tcuTimerDuration_t structure format.
tcu_EvDetectUnit	INTEGER	Time granularity used (in µs steps) in the event detection service: for inactivity period settings (_adl_tcuEventDetectionSettings_t::Duration) for last stable state duration information (_adl_tcuEventDetectionInfo_t::LastStateDuration)
tcu_EvCaptUnit	INTEGER	Time granularity used (in µs steps) in the event capture service, for capture duration setting (_adl_tcuEventCaptureSettings_t::Duration)

Bus Service

Registry Entry	Туре	Description
i2c_NbBlocks ³	INTEGER	The number of i2c blocks managed by the embedded module
i2c_xx_Cap	INTEGER	The capabilities of the block, defined as a combination of the adl_busI2CCap_e type values.
i2c_xx_MaxLength	Unsigned INTEGER⁴	The maximum amount of items that can be passed in a I2C read/write operation
spi_NbBlocks ³	INTEGER	The number of spi blocks managed by the embedded module
spi_xx_Common	INTEGER	The generic capabilities of the block, defined as a combination of the adl_busSpiCommonCap_e typevalues.
spi_xx_ClockDivStep	INTEGER	The number of steps of the clock divider (see <pre>adl_busSPISettings_t ::Clk_Speed field description)</pre>
spi_xx_MaxLength	INTEGER	The maximum amount of items that can be passed in aSPI read/write operation
spi_xx_DataSizes ²	INTEGER	Available data sizes for ADL_BUS_CMD_SET_DATA_SIZE IOCtl command
spi_xx_Master_OpcodeSizes ²	Unsigned INTEGER⁴	Available Opcode sizes for ADL_BUS_CMD_SET_OP_SIZE IOCtl command
spi_xx_Master_AddressSizes ²	Unsigned INTEGER	Available Address sizes for ADL_BUS_CMD_SET_ADD_SIZE IOCtl command
spi_xx_Master_Cap	INTEGER	The capabilities of the block in Master mode, defined asa combination of the adl_busSpiCap_e type
spi_xx_Master_MaxFreqClock	INTEGER	The maximum frequency (in kHz) of the clock in Master mode (see adl_busSPISettings_t:: Clk_Speed field description)
Para_NbBlocks3	INTEGER	The number of parallel bus blocks managed by the embedded module
Para_NbCS	INTEGER	The number of chip select available to the customer
Para_CS	INTEGER	Registry Entry
Para_xx_Addr	INTEGER	i2c_NbBlocks ³
Para_xx_Freq	INTEGER	Current frequency of the Chip select XX

(See Notes below)

Note 1 For the registry entry the xx part has to be replaced by the number of the instance.

Example: if you want the capabilities of the I2C1 block the registry entry to use will be i2c_01_Cap.

Example: if you want the common capabilities of the SPI2 block the registry entry to use will be spi_02_Common.

Note 2 Sizes are coded in a bit field, where size n is available when the n-1 bit is set.

Example: 0x80008003 means sizes 32 bits, 16 bits, 2 bits and 1 bit are available.

- Note 3 A SPI/I2C/Parallel bus block will be identified with a number from 1 to spi_NbBlocks or i2c_NbBlocks or Parallel_NbBlocks.
- Note 4 Entries using the Unsigned INTEGER type have to be casted to an u32 value after being retrieved from adl_regGetHWInteger function.

RTC Service

Registry Entry	Туре	Description
rtc_PreScalerMaxValue	INTEGER	0: No second fractional part xxx: Second fractional part resolution