#### 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 <b>adl_tcuTimerBoundaries_t</b> structure format.
tcu_TimerTick	DATA	Timer resolution used by the Accurate Timer Service, using the <b>adl_tcuTimerDuration_t</b> structure format.
tcu_EvDetectUnit	INTEGER	Time granularity used (in µs steps) in the event detection service: for inactivity period settings (_adl_tcuEventDetectionSettings_t::Durati on) for last stable state duration information (_adl_tcuEventDetectionInfo_t::LastStateD uration)
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 <sup>3</sup>	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 <b>adl_busI2CCap_e</b> type values.
i2c_xx_MaxLength	Unsigned INTEGER⁴	The maximum amount of items that can be passed in a I2C read/write operation
spi_NbBlocks <sup>3</sup>	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 <b>adl_busSpiCommonCap_e</b> type values.
spi_xx_ClockDivStep	INTEGER	The number of steps of the clock divider (see <b>adl_busSPISettings_t ::Clk_Speed</b> field description)
spi_xx_MaxLength	INTEGER	The maximum amount of items that can be passed in a SPI read/write operation
spi_xx_DataSizes <sup>2</sup>	INTEGER	Available data sizes for ADL_BUS_CMD_SET_DATA_SIZE IOCtl command
spi_xx_Master_OpcodeSizes <sup>2</sup>	Unsigned INTEGER⁴	Available Opcode sizes for ADL_BUS_CMD_SET_OP_SIZE IOCtl command
spi_xx_Master_AddressSizes <sup>2</sup>	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 as a combination of the <b>adl_busSpiCap_e</b> type
spi_xx_Master_MaxFreqClock	INTEGER	The maximum frequency (in kHz) of the clock in Master mode (see adl_busSPISettings_t:: <b>Clk_Speed</b> 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 <sup>3</sup>
Para_xx_Freq	INTEGER	Current frequency of the Chip select XX

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

#### EXTINT ADL Service

Registry Entry	Туре	Description
extint_NbExternalInterrupt	INTEGER	Number of external interrupt pins
<pre>extint_RisingEdgeSensitivi ty</pre>	INTEGER	Rising edge sensitivity supported
extint_FallingEdgeSensitiv ity	INTEGER	Falling edge sensitivity supported
extint_BothEdgeSensitivity	INTEGER	Both edge detector supported
extint_LowLevelSensitivity	INTEGER	Low level sensitivity not supported
extint_HighLevelSensitivit y	INTEGER	High level sensitivity not supported
extint_BypassMode	INTEGER	Bypass mode supported
extint_StretchingMode	INTEGER	Stretching mode supported
extint_DebounceMode	INTEGER	Debounce mode supported
extint_MaxDebounceDuration	INTEGER	Debounce max duration in ms
extint_DebounceNbStep	INTEGER	Number of step for debounce duration
extint_NbPriority	INTEGER	Available priority levels for the EXTINT service (to be used as a <b>adl_irqPriorityLevel_</b> e value in the IRQ service)

Registry Entry	Туре	Description
dac_NbBlocks	INTEGER	The number of DAC blocks available.
dac_xx_DigitInitValue	INTEGER	Digital value at DAC resource allocation. dac_xx_DigitInitValue is set at -1 if the default value is unknown.
dac_xx_MaxRefVoltage	INTEGER	Reference voltage of the DAC output when the maximal digital value is set.
dac_xx_MinRefVoltage	INTEGER	Reference voltage of the DAC output when the minimal digital value is set.
dac_xx_Resolution	INTEGER	DAC resolution in steps.
dac_xx_DacType	INTEGER	DAC type, see section <b>adl_dacType_e</b> .
dac_xx_InterruptContextUse d	INTEGER	This value is set to 1 if DAC write operations can be called under interrupt context

Note 1 For the registry entry the xx part must be replaced by the number of the instance. Example: if you want the Resolution capabilities of the DAC02 block, the registry entry to use will be dac\_02\_Resolution.

Note 2 DACs will be identified with a number as 0, 1, 2, . . . . dac\_NbBlocks-1. Note 3 For each block, the settling time capabilities are defined in the PTS.

ADL	ADC	Service
-----	-----	---------

Registry Entry	Туре	Description
adc_NbBlocks	INTEGER	The number of ADC blocks available
adc_xx_ResolutionsBits	INTEGER	To get on how many bits, is coded the result.
adc_xx_ MaxInputRange	INTEGER	The minimum input voltage in mV supported by each ADC.
adc_xx_ MinInputRange	INTEGER	The maximum input voltage in mV supported by each ADC.
adc_xx_InterruptContextUse d	INTEGER	This value is set to 1, if ADC read functions can be called under interrupt context

Note 1 For the registry entry the xx part must be replaced by the number of the instance. Example: if you want the Resolution Bits capabilities of the ADC02 block the registry entry to use will be adc\_02\_ResolutionBits.

Note 2 ADCs will be identified with a number as 0, 1, 2, . . . . adc\_NbBlocks-1. Note 3 For each block, the sampling time capability is defined in the PTS.