Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	48-TQFP
Supplier Device Package	48-TQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atuc128l4u-aur

Table 3-4. Nexus OCD AUX Port Connections

Pin	AXS=1	AXS=0
EVTO_N	PA04	PA04
MCKO	PA06	PB01
MSEO[1]	PA07	PB11
MSEO[0]	PA11	PB12

3.2.5 Oscillator Pinout

The oscillators are not mapped to the normal GPIO functions and their muxings are controlled by registers in the System Control Interface (SCIF). Please refer to the SCIF chapter for more information about this.

Table 3-5. Oscillator Pinout

48-pin	64-pin	Pin Name	Oscillator Pin
3	3	PA08	XIN0
46	62	PA10	XIN32
26	34	PA13	XIN32_2
2	2	PA09	XOUT0
47	63	PA12	XOUT32
25	33	PA20	XOUT32_2

3.2.6 Other Functions

The functions listed in Table 3-6 are not mapped to the normal GPIO functions. The aWire DATA pin will only be active after the aWire is enabled. The aWire DATAOUT pin will only be active after the aWire is enabled and the 2_PIN_MODE command has been sent. The WAKE_N pin is always enabled. Please refer to Section 6.1.4.2 on page 44 for constraints on the WAKE_N pin.

Table 3-6. Other Functions

48-pin	64-pin	Pin Name	Function
27	35	PA11	WAKE_N
22	30	RESET_N	aWire DATA
11	15	PA00	aWire DATAOUT

4.5.3 Supervisor Calls

The AVR32 instruction set provides a supervisor mode call instruction. The *scall* instruction is designed so that privileged routines can be called from any context. This facilitates sharing of code between different execution modes. The *scall* mechanism is designed so that a minimal execution cycle overhead is experienced when performing supervisor routine calls from time-critical event handlers.

The *scall* instruction behaves differently depending on which mode it is called from. The behaviour is detailed in the instruction set reference. In order to allow the *scall* routine to return to the correct context, a return from supervisor call instruction, *rets*, is implemented. In the AVR32UC CPU, *scall* and *rets* uses the system stack to store the return address and the status register.

4.5.4 Debug Requests

The AVR32 architecture defines a dedicated Debug mode. When a debug request is received by the core, Debug mode is entered. Entry into Debug mode can be masked by the DM bit in the status register. Upon entry into Debug mode, hardware sets the SR.D bit and jumps to the Debug Exception handler. By default, Debug mode executes in the exception context, but with dedicated Return Address Register and Return Status Register. These dedicated registers remove the need for storing this data to the system stack, thereby improving debuggability. The Mode bits in the Status Register can freely be manipulated in Debug mode, to observe registers in all contexts, while retaining full privileges.

Debug mode is exited by executing the *retd* instruction. This returns to the previous context.

4.5.5 Entry Points for Events

Several different event handler entry points exist. In AVR32UC, the reset address is 0x80000000. This places the reset address in the boot flash memory area.

TLB miss exceptions and *scall* have a dedicated space relative to EVBA where their event handler can be placed. This speeds up execution by removing the need for a jump instruction placed at the program address jumped to by the event hardware. All other exceptions have a dedicated event routine entry point located relative to EVBA. The handler routine address identifies the exception source directly.

AVR32UC uses the ITLB and DTLB protection exceptions to signal a MPU protection violation. ITLB and DTLB miss exceptions are used to signal that an access address did not map to any of the entries in the MPU. TLB multiple hit exception indicates that an access address did map to multiple TLB entries, signalling an error.

All interrupt requests have entry points located at an offset relative to EVBA. This autovector offset is specified by an interrupt controller. The programmer must make sure that none of the autovector offsets interfere with the placement of other code. The autovector offset has 14 address bits, giving an offset of maximum 16384 bytes.

Special considerations should be made when loading EVBA with a pointer. Due to security considerations, the event handlers should be located in non-writeable flash memory, or optionally in a privileged memory protection region if an MPU is present.

If several events occur on the same instruction, they are handled in a prioritized way. The priority ordering is presented in Table 4-4 on page 34. If events occur on several instructions at different locations in the pipeline, the events on the oldest instruction are always handled before any events on any younger instruction, even if the younger instruction has events of higher priority

9.8.4 Flash Parameter Register

Name: FPR

Access Type: Read-only

Offset: 0x0C

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	-	PSZ		

7	6	5	4	3	2	1	0
-	-	-	-	FSZ			

- **PSZ: Page Size**
  The size of each flash page.

Table 9-9. Flash Page Size

PSZ	Page Size
0	32 Byte
1	64 Byte
2	128 Byte
3	256 Byte
4	512 Byte
5	1024 Byte
6	2048 Byte
7	4096 Byte

10.6.3 Channel Enable Register High

Name: CERH

Access Type: Read/Write

Offset: 0x08

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	CERH[30:24]						

23	22	21	20	19	18	17	16
CERH[23:16]							

15	14	13	12	11	10	9	8
CERH[15:8]							

7	6	5	4	3	2	1	0
CERH[7:0]							

- **CERH[n]: Channel Enable Register High**
  0: Channel (n+32) is not enabled.
  1: Channel (n+32) is enabled.

14.5.7 System RC Oscillator (RCSYS)

Rev: 1.1.1.0

The system RC oscillator has a startup time of three cycles, and is always available except in some sleep modes. Please refer to the Power Manager chapter for details. The system RC oscillator operates at a nominal frequency of 115kHz, and is calibrated using the Calibration Value field (CALIB) in the RC Oscillator Calibration Register (RCCR). After a Power-on Reset (POR), the RCCR.CALIB field is loaded with a factory defined value stored in the Flash fuses. Please refer to the Fuse setting chapter for more details about RCCR fuses and how to program the fuses.

If the Flash Calibration Done (FCD) bit in the RCCR is zero at any reset, the flash calibration will be redone and the RCCR.FCD bit will be set before program execution starts in the CPU. If the RCCR.FCD is one, the flash calibration will only be redone after a Power-on Reset.

To prevent unexpected writes to RCCR due to software bugs, write access to this register is protected by a locking mechanism. For details please refer to the UNLOCK register description.

Although it is not recommended to override default factory settings, it is still possible to override the default values by writing to RCCR.CALIB.

14.5.8 Voltage Regulator (VREG)

Rev: 1.1.0.0

The embedded voltage regulator can be used to provide the VDDCORE voltage from the internal regulator supply voltage. It is controlled by the Voltage Regulator Calibration Register (VREGCR). The voltage regulator is enabled by default at start-up but can be disabled by software if an external voltage is provided on the VDDCORE pin. The VREGCR also contains bits to control the POR18 detector and the POR33 detector.

14.5.8.1 Register protection

To prevent unexpected writes to VREGCR due to software bugs, write access to this register is protected by a locking mechanism. For details please refer to the UNLOCK register description.

To prevent further modifications by software, the content of the VREGCR register can be set as read-only by writing a one to the Store Final Value bit (VREGCR.SFV). Once this bit is set, software can not change the VREGCR content until a Power-on Reset (POR) is applied.

14.5.8.2 Controlling voltage regulator output

The voltage regulator is always enabled at start-up, i.e. after a POR or when waking up from Shutdown mode. It can be disabled by software by writing a zero to the Enable bit (VREGCR.EN). This bit is set after a POR. Because of internal synchronization, the voltage regulator is not immediately enabled or disabled. The actual state of the voltage regulator can be read from the ON bit (VREGCR.ON).

The voltage regulator output level is controlled by the Select VDD field (SELVDD) in VREGCR. The default value of this field corresponds to a regulator output voltage of 1.8V. Other values of this field are not defined, and it is not recommended to change the value of this field.

The Voltage Regulator OK bit (VREGCR.VREGOK) bit indicates when the voltage regulator output has reached the voltage threshold level.

14.6.20 Temperature Sensor Configuration Register

Name: TSENS

Access Type: Read/Write

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

7	6	5	4	3	2	1	0
-	-	-			-	-	EN

- **EN: Temperature Sensor Enable**
  0: The Temperature Sensor is disabled.
  1: The Temperature Sensor is enabled.

Note that this register is protected by a lock. To write to this register the UNLOCK register has to be written first. Please refer to the UNLOCK register description for details.

14.6.32 PLL Version Register

Name: PLLVERSION

Access Type: Read-only

Offset: 0x03C4

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	VARIANT			

15	14	13	12	11	10	9	8
-	-	-	-	VERSION[11:8]			

7	6	5	4	3	2	1	0
VERSION[7:0]							

- **VARIANT: Variant number**
  Reserved. No functionality associated.
- **VERSION: Version number**
  Version number of the module. No functionality associated.

15.6.7 Interrupt Mask Register

Name: IMR

Access Type: Read-only

Offset: 0x18

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	CLKRDY	-	-	-	READY	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	PER1	PER0
15	14	13	12	11	10	9	8
-	-	-	-	-	-	ALARM1	ALARM0
7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	OVF

0: The corresponding interrupt is disabled.

1: The corresponding interrupt is enabled.

A bit in this register is cleared when the corresponding bit in IDR is written to one.

A bit in this register is set when the corresponding bit in IER is written to one.

16.6.3 Status Register

Name: SR

Access Type: Read-only

Offset: 0x008

Reset Value: 0x00000003

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-

23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-

15	14	13	12	11	10	9	8
-	-	-	-	-	-	-	-

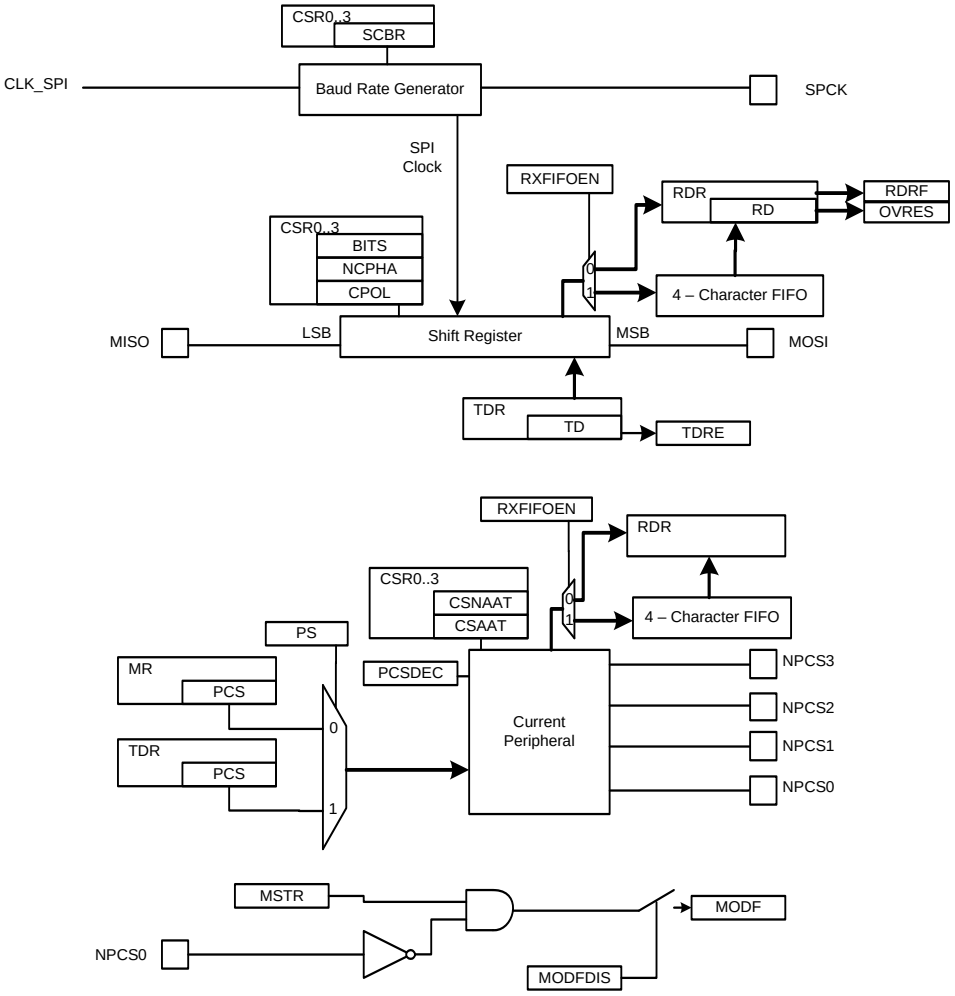
7	6	5	4	3	2	1	0
-	-	-	-	-	-	CLEARED	WINDOW

- **CLEARED: WDT Counter Cleared**
  This bit is cleared when writing a one to the CLR.WDTCLR bit.
  This bit is set when clearing the WDT counter is done.
- **WINDOW: Within Window**
  This bit is cleared when the WDT counter is inside the TBAN period.
  This bit is set when the WDT counter is inside the PSEL period.

Figure 21-5 on page 491shows a block diagram of the SPI when operating in master mode. Figure 21-6 on page 492 shows a flow chart describing how transfers are handled.

21.7.3.1 Master mode block diagram

Figure 21-5. Master Mode Block Diagram



In I²C mode:

- The address in CR.ADR is checked for address match if CR.SMATCH is one.
- The General Call address is checked for address match if CR.GCMATCH is one.

In SMBus mode:

- The address in CR.ADR is checked for address match if CR.SMATCH is one.
- The Alert Response Address is checked for address match if CR.SMAL is one.
- The Default Address is checked for address match if CR.SMDA is one.
- The Host Header Address is checked for address match if CR.SMHH is one.

23.8.2.4 Clock Stretching

Any slave or bus master taking part in a transfer may extend the TWCK low period at any time. The TWIS may extend the TWCK low period after each byte transfer if CR.STREN is one and:

- Module is in slave transmitter mode, data should be transmitted, but THR is empty, or
- Module is in slave receiver mode, a byte has been received and placed into the internal shifter, but the Receive Holding Register (RHR) is full, or
- Stretch-on-address-match bit CR.SOAM=1 and slave was addressed. Bus clock remains stretched until all address match bits in the Status Register (SR) have been cleared.

If CR.STREN is zero and:

- Module is in slave transmitter mode, data should be transmitted but THR is empty: Transmit the value present in THR (the last transmitted byte or reset value), and set SR.URUN.
- Module is in slave receiver mode, a byte has been received and placed into the internal shifter, but RHR is full: Discard the received byte and set SR.ORUN.

23.8.2.5 Bus Errors

If a bus error (misplaced START or STOP) condition is detected, the SR.BUSERR bit is set and the TWIS waits for a new START condition.

23.8.3 Slave Transmitter Mode

If the TWIS matches an address in which the R/$\overline{W}$ bit in the TWI address phase transfer is set, it will enter slave transmitter mode and set the SR.TRA bit (note that SR.TRA is set one CLK_TWIS cycle after the relevant address match bit in the same register is set).

After the address phase, the following actions are performed:

1. If SMBus mode and PEC is used, NBYTES must be set up with the number of bytes to transmit. This is necessary in order to know when to transmit the PEC byte. NBYTES can also be used to count the number of bytes received if using DMA.
2. Byte to transmit depends on I²C/SMBus mode and CR.PEC:
   – If in I²C mode or CR.PEC is zero or NBYTES is non-zero: The TWIS waits until THR contains a valid data byte, possibly stretching the low period of TWCK. After THR contains a valid data byte, the data byte is transferred to a shifter, and then SR.TXRDY is changed to one because the THR is empty again.
   – SMBus mode and CR.PEC is one: If NBYTES is zero, the generated PEC byte is automatically transmitted instead of a data byte from THR. TWCK will not be stretched by the TWIS.
3. The data byte in the shifter is transmitted.

24.8.9 Receive Holding Register

Name: RHR

Access Type: Read-only

Offset: 0x20

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
RHR[31:24]							

23	22	21	20	19	18	17	16
RHR[23:16]							

15	14	13	12	11	10	9	8
RHR[15:8]							

7	6	5	4	3	2	1	0
RHR[7:0]							

- **RHR: Received Word**
  This field is set by hardware to the last received data word. If MR.DATALENGTH specifies less than 32 bits, data shall be right-justified into the RHR field.

tem will only be woken up if the user peripheral generates an interrupt as a result of the operation. This concept is known as SleepWalking and is described in further detail in the Power Manager chapter. Note that asynchronous peripheral events may be associated with a delay due to the need to restart the system clock source if this has been stopped in the sleep mode.

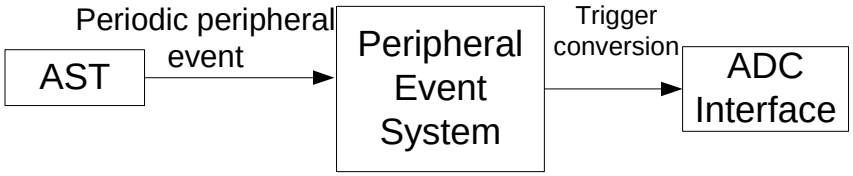
27.5 Application Example

This application example shows how the Peripheral Event System can be used to program the ADC Interface to perform ADC conversions at selected intervals.

Conversions of the active analog channels are started with a software or a hardware trigger. One of the possible hardware triggers is a peripheral event trigger, allowing the Peripheral Event System to synchronize conversion with some configured peripheral event source. From Table 27-3 and Table 27-4, it can be read that this peripheral event source can be either an AST peripheral event, or an event from the PWM Controller. The AST can generate periodic peripheral events at selected intervals, among other types of peripheral events. The Peripheral Event System can then be used to set up the ADC Interface to sample an analog signal at regular intervals.

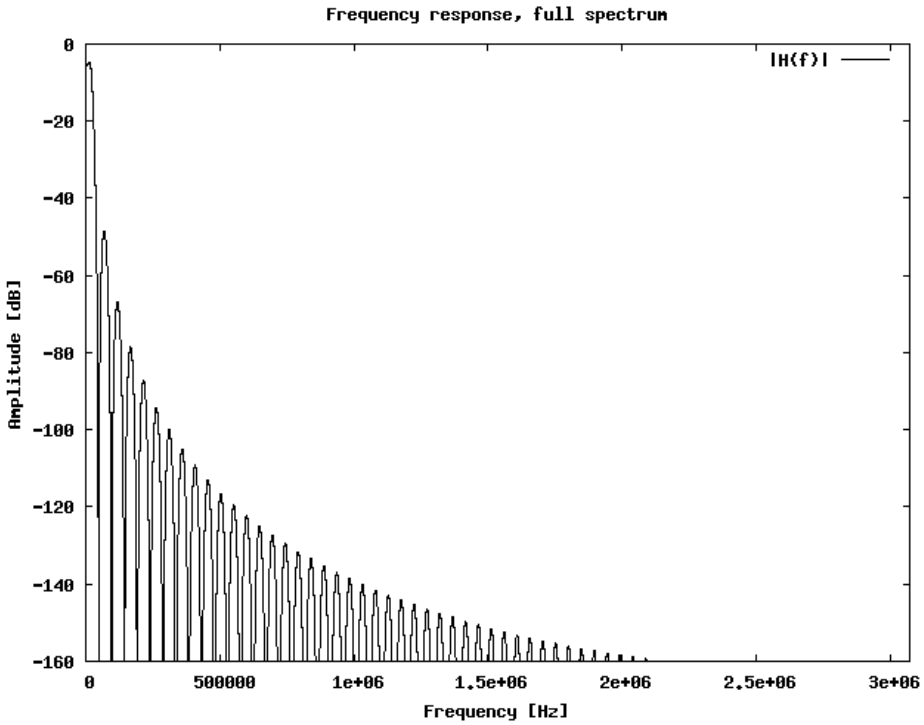
The user must enable peripheral events in the AST and in the ADC Interface to accomplish this. The periodic peripheral event in the AST is enabled by writing a one to the corresponding bit in the AST Event Enable Register (EVE). To select the peripheral event trigger for the ADC Interface, the user must write the value 0x7 to the Trigger Mode (TRGMOD) field in the ADC Interface Trigger Register (TRGR). When the peripheral events are enabled, the AST will generate peripheral events at the selected intervals, and the Peripheral Event System will route the peripheral events to the ADC Interface, which will perform ADC conversions at the selected intervals.

Figure 27-2. Application Example



Since the AST peripheral event is asynchronous, the description above will also work in sleep modes where the ADC clock is stopped. In this case, the ADC clock (and clock source, if needed) will be restarted during the ADC conversion. After the conversion, the ADC clock and clock source will return to the sleep state, unless the ADC generates an interrupt, which in turn will wake up the system. Using asynchronous interrupts thus allows ADC operation in much lower power states than would otherwise be possible.

Figure 28-7. Frequency Response up to 128x Sampling Frequency



Channels are enabled by writing a one to the corresponding bit in the Channel Enable Register (CHER), and disabled by writing a one to the corresponding bit in the Channel Disable Register (CHDR). Active channels are listed in the Channel Status Register (CHSR).

When a conversion sequence is started, all enabled channels will be converted in one sequence and the result will be placed in the Last Converted Data Register (LCDR) with the channel number used to produce the result. It is important to read out the results while the conversion sequence is ongoing, as new values will automatically overwrite any old value and the old value will be lost if not previously read by the user.

If the Analog-to-Digital Converter cell is inactive when starting a conversion sequence, the conversion logic will wait a configurable number of CLK_ADC cycles as defined in the startup time field in the ADC Configuration Register (ACR). After the cell is activated all enabled channels is converted one by one until no more enabled channels exist. The conversion sequence converts each enabled channel in order starting with the channel with the lowest channel number. If the ACR.SLEEP bit is one, the Analog-to-Digital Converter cell is deactivated after the conversion sequence has finished.

For each channel converted, the ADCIFB waits a Sample and Hold number of CLK_ADC cycles as defined in the SHTIM field in ACR, and then instructs the Analog-to-Digital Converter cell to start converting the analog voltage. The ADC cell requires 10 CLK_ADC cycles to actually convert the value, so the total time to convert a channel is Sample and Hold + 10 CLK_ADC cycles.

29.6.10 Analog Compare Mode

The ADCIFB can test if the converted values, as they become available, are below, above, or inside a specified range and generate interrupt requests based on this information. This is useful for applications where the user wants to monitor some external analog signal and only initiate actions if the value is above, below, or inside some specified range.

The Analog Compare mode is enabled by writing a one to the Analog Compare Enable (ACE) bit in the Mode Register (MR). The values to compare must be written to the Low Value (LV) field and the High Value (HV) field in the Compare Value Register (CVR). The Analog Compare mode will, when enabled, check all enabled channels against the pre-programmed high and low values and set status bits.

To generate an interrupt request if a converted value is below a limit, write the limit to the CVR.LV field and enable interrupt request on the Compare Lesser Than (CLT) bit by writing a one to the corresponding bit in the Interrupt Enable Register (IER). To generate an interrupt request if a converted value is above a limit, write the limit to the CVR.HV field and enable interrupt for Compare Greater Than (CGT) bit. To generate an interrupt request if a converted value is inside a range, write the low and high limit to the LV and HV fields and enable the Compare Else (CELSE) interrupt. To generate an interrupt request if a value is outside a range, write the LV and HV fields to the low and high limits of the range and enable CGT and CLT interrupts.

Note that the values written to LV and HV must match the resolution selected in the ADC Configuration Register (ACR).

29.6.11 Interrupt Operation

Interrupt requests are enabled by writing a one to the corresponding bit in the Interrupt Enable Register (IER) and disabled by writing a one to the corresponding bit in the Interrupt Disable Register (IDR). Enabled interrupts can be read from the Interrupt Mask Register (IMR). Active interrupt requests, but potentially masked, are visible in the Interrupt Status Register (ISR). To

30.9.11 Window Configuration Register

Name: CONFWn

Access Type: Read/Write

Offset: 0x80,0x84,0x88,0x8C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	-	-	-	-
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	WFEN
15	14	13	12	11	10	9	8
-	-	-	-	WEVEN	WEVSRC		
7	6	5	4	3	2	1	0
-	-	-	-	-	-	WIS	

- **WFEN: Window Mode Enable**
  - 0: The window mode is disabled.
  - 1: The window mode is enabled.
- **WEVEN: Window Event Enable**
  - 0: Event from awout is disabled.
  - 1: Event from awout is enabled.
- **WEVSRC: Event Source Selection for Window Mode**
  - 000: Event on acwout rising edge.
  - 001: Event on acwout falling edge.
  - 010: Event on awout rising or falling edge.
  - 011: Inside window.
  - 100: Outside window.
  - 101: Measure done.
  - 110-111: Reserved.
- **WIS: Window Mode Interrupt Settings**
  - 00: Window interrupt as soon as the input voltage is inside the window.
  - 01: Window interrupt as soon as the input voltage is outside the window.
  - 10: Window interrupt on toggle of window compare output.
  - 11: Window interrupt when evaluation of input voltage is done.

9, 3, 4, 5, 6, 7, 8, 9, 3, 4, etc. MAXDEV must not exceed the value of (2(DIV+1)), or undefined behavior will occur.

31.6.6 Synchronization

To prevent interference from the 50 or 60 Hz mains line the CAT can trigger acquisition on the SYNC signal. The SYNC signal should be derived from the mains line. The acquisition will trigger on a falling edge of this signal. To enable synchronization for a specific acquisition type, the user must write a one to the SYNC bit in the appropriate Configuration Register 1 (MGCFG1, ATCFG1, TGACFG1, or TGBCFG1).

For QMatrix acquisition, all X lines must be sampled at a specific phase of the noise signal for the synchronization to be effective. This can be accomplished by the synchronization timer, which is enabled by writing a non-zero value to the SYNCTIM field in the MGCFG2 register. This ensures that the start of the acquisition of each X line is spaced at regular intervals, defined by the SYNCTIM field.

31.6.7 Resistive Drive

By default, the CAT pins are driven with normal I/O drive properties. Some of the CSA and CSB pins can optionally drive with a 1k output resistance for improved EMC. The pins that have this capability are listed in the Module Configuration section.

31.6.8 Discharge Current Sources

The device integrates discharge current sources, which can be used to discharge the sampling capacitors during the QMatrix measurement phase. The discharge current sources are enabled by writing the GLEN bit in the Discharge Current Source (DICS) register to one. This enables an internal reference voltage, which can be either the internal 1.1V band gap voltage or VDDIO/3, as selected by the INTVREFSEL bit in the DICS register. If the DICS.INTREFSEL bit is one, the reference voltage is applied across an internal resistor, R_{int}. Otherwise, the voltage is applied to the DIS pin, and an external reference resistor must be connected between DIS and ground. The nominal discharge current is given by the following formula, where V_{ref} is the reference voltage, R_{ref} is the value of the reference resistor, trim is the value written to the DICS.TRIM field, and k is a constant of proportionality:

$I = (V_{ref}/R_{ref})*(1+(k*trim))$

The values for the internal reference resistor, R_{int}, and the constant, k, may be found in the Electrical Characteristics section. The nominal discharge current may be programmed between 2 and 20 µA. The reference current can be fine-tuned by adjusting the trim value in the DICS.TRIM field.

The reference current is mirrored to each Y-pin if the corresponding bit is written to one in the DICS.SOURCES field.

31.6.9 Voltage Divider Enable (VDIVEN) Capability

In many QMatrix applications, the sense capacitors will be charged to 50 mV or more and the negative reference pin (ACREFN) of the analog comparators can be tied directly to ground. In that case, the relatively small input offset voltage of the comparators will not cause acquisition problems. However, in certain specialized QMatrix applications such as interpolated touch screens, it may be desirable for the sense capacitors to be charged to less than 25 mV. When such small voltages are used on the sense capacitors, the input offset voltage of the comparators becomes an issue and can cause QMatrix acquisition problems.

31.7.12 Matrix Group Configuration Register 2

Name: MGCFG2

Access Type: Read/Write

Offset: 0x38

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
ACCTRL	CONSEN			-			

23	22	21	20	19	18	17	16
CXDILEN							

15	14	13	12	11	10	9	8
-				SYNCTIM[11:8]			

7	6	5	4	3	2	1	0
SYNCTIM[7:0]							

- **ACCTRL: Analog Comparator Control**
  When written to one, allows the CAT to disable the analog comparators when they are not needed. When written to zero, the analog comparators are always enabled.
- **CONSEN: Consensus Filter Length**
  For QMatrix sensors, specifies that discharge will be terminated when CONSEN out of the most recent 5 comparator samples are positive. For example, a value of 3 in the CONSEN field will terminate discharge when 3 out of the most recent 5 comparator samples are positive. When CONSEN has the default value of 0, discharge will be terminated immediately when the comparator output goes positive.
- **CXDILEN: Cx Capacitor Discharge Length**
  For QMatrix sensors, specifies how many burst prescaler clock cycles the CAT should use to discharge the Cx capacitor at the end of each burst cycle.
- **SYNCTIM: Sync Time Interval**
  When non-zero, determines the number of prescaled clock cycles between the start of the acquisition on each X line for QMatrix acquisition.

This bit is set when the clock is disabled.

This bit is cleared when the clock is enabled.

- **BUSY: Synchronizer Busy**

  0: The asynchronous interface is ready to accept more data.

  1: The asynchronous interface is busy and will block writes to CTRL, BRR, and THR.

  This bit is set when the asynchronous interface becomes busy.

  This bit is cleared when the asynchronous interface becomes ready.

38. Errata

38.1 Rev. C

38.1.1 SCIF

1. **The RC32K output on PA20 is not always permanently disabled**
   The RC32K output on PA20 may sometimes re-appear.
   Fix/Workaround
   Before using RC32K for other purposes, the following procedure has to be followed in order to properly disable it:
   - Run the CPU on RCSYS
   - Disable the output to PA20 by writing a zero to PM.PPCR.RC32OUT
   - Enable RC32K by writing a one to SCIF.RC32KCR.EN, and wait for this bit to be read as one
   - Disable RC32K by writing a zero to SCIF.RC32KCR.EN, and wait for this bit to be read as zero.

2. **PLLCOUNT value larger than zero can cause PLLEN glitch**
   Initializing the PLLCOUNT with a value greater than zero creates a glitch on the PLLEN signal during asynchronous wake up.
   Fix/Workaround
   The lock-masking mechanism for the PLL should not be used.
   The PLLCOUNT field of the PLL Control Register should always be written to zero.

3. **Writing 0x5A5A5A5A to the SCIF memory range will enable the SCIF UNLOCK feature**
   The SCIF UNLOCK feature will be enabled if the value 0x5A5A5A5A is written to any location in the SCIF memory range.
   Fix/Workaround
   None.

38.1.2 SPI

1. **SPI data transfer hangs with CSR0.CSAAT==1 and MR.MODFDIS==0**
   When CSR0.CSAAT==1 and mode fault detection is enabled (MR.MODFDIS==0), the SPI module will not start a data transfer.
   Fix/Workaround
   Disable mode fault detection by writing a one to MR.MODFDIS.

2. **Disabling SPI has no effect on the SR.TDRE bit**
   Disabling SPI has no effect on the SR.TDRE bit whereas the write data command is filtered when SPI is disabled. Writing to TDR when SPI is disabled will not clear SR.TDRE. If SPI is disabled during a PDCA transfer, the PDCA will continue to write data to TDR until its buffer is empty, and this data will be lost.
   Fix/Workaround
   Disable the PDCA, add two NOPs, and disable the SPI. To continue the transfer, enable the SPI and PDCA.

3. **SPI disable does not work in SLAVE mode**
   SPI disable does not work in SLAVE mode.
   Fix/Workaround
   Read the last received data, then perform a software reset by writing a one to the Software Reset bit in the Control Register (CR.SWRST).