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	Obsolete
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K 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/microchip-technology/at32uc3l032-aur

8. Flash Controller (FLASHCDW)

Rev: 1.0.2.0

8.1 Features

- **Controls on-chip flash memory**
- **Supports 0 and 1 wait state bus access**
- **Buffers reducing penalty of wait state in sequential code or loops**
- **Allows interleaved burst reads for systems with one wait state, outputting one 32-bit word per clock cycle for sequential reads**
- **Secure State for supporting FlashVault technology**
- **32-bit HSB interface for reads from flash and writes to page buffer**
- **32-bit PB interface for issuing commands to and configuration of the controller**
- **Flash memory is divided into 16 regions can be individually protected or unprotected**
- **Additional protection of the Boot Loader pages**
- **Supports reads and writes of general-purpose Non Volatile Memory (NVM) bits**
- **Supports reads and writes of additional NVM pages**
- **Supports device protection through a security bit**
- **Dedicated command for chip-erase, first erasing all on-chip volatile memories before erasing flash and clearing security bit**

8.2 Overview

The Flash Controller (FLASHCDW) interfaces the on-chip flash memory with the 32-bit internal HSB bus. The controller manages the reading, writing, erasing, locking, and unlocking sequences.

8.3 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

8.3.1 Power Management

If the CPU enters a sleep mode that disables clocks used by the FLASHCDW, the FLASHCDW will stop functioning and resume operation after the system wakes up from sleep mode.

8.3.2 Clocks

The FLASHCDW has two bus clocks connected: One High Speed Bus clock (CLK_FLASHCDW_HSB) and one Peripheral Bus clock (CLK_FLASHCDW_PB). These clocks are generated by the Power Manager. Both clocks are enabled at reset, and can be disabled by writing to the Power Manager. The user has to ensure that CLK_FLASHCDW_HSB is not turned off before reading the flash or writing the pagebuffer and that CLK_FLASHCDW_PB is not turned off before accessing the FLASHCDW configuration and control registers. Failing to do so may deadlock the bus.

8.3.3 Interrupts

The FLASHCDW interrupt request lines are connected to the interrupt controller. Using the FLASHCDW interrupts requires the interrupt controller to be programmed first.

The page buffer is not automatically reset after a page write. The programmer should do this manually by issuing the Clear Page Buffer flash command. This can be done after a page write, or before the page buffer is loaded with data to be stored to the flash page.

8.5 Flash Commands

The FLASHCDW offers a command set to manage programming of the flash memory, locking and unlocking of regions, and full flash erasing. See Section 8.8.2 for a complete list of commands.

To run a command, the CMD field in the Flash Command Register (FCMD) has to be written with the command number. As soon as the FCMD register is written, the FRDY bit in the Flash Status Register (FSR) is automatically cleared. Once the current command is complete, the FSR.FRDY bit is automatically set. If an interrupt has been enabled by writing a one to FCR.FRDY, the interrupt request line of the Flash Controller is activated. All flash commands except for Quick Page Read (QPR) and Quick User Page Read (QPRUP) will generate an interrupt request upon completion if FCR.FRDY is one.

Any HSB bus transfers attempting to read flash memory when the FLASHCDW is busy executing a flash command will be stalled, and allowed to continue when the flash command is complete.

After a command has been written to FCMD, the programming algorithm should wait until the command has been executed before attempting to read instructions or data from the flash or writing to the page buffer, as the flash will be busy. The waiting can be performed either by polling the Flash Status Register (FSR) or by waiting for the flash ready interrupt. The command written to FCMD is initiated on the first clock cycle where the HSB bus interface in FLASHCDW is IDLE. The user must make sure that the access pattern to the FLASHCDW HSB interface contains an IDLE cycle so that the command is allowed to start. Make sure that no bus masters such as DMA controllers are performing endless burst transfers from the flash. Also, make sure that the CPU does not perform endless burst transfers from flash. This is done by letting the CPU enter sleep mode after writing to FCMD, or by polling FSR for command completion. This polling will result in an access pattern with IDLE HSB cycles.

All the commands are protected by the same keyword, which has to be written in the eight highest bits of the FCMD register. Writing FCMD with data that does not contain the correct key and/or with an invalid command has no effect on the flash memory; however, the PROGE bit is set in the Flash Status Register (FSR). This bit is automatically cleared by a read access to the FSR register.

Writing a command to FCMD while another command is being executed has no effect on the flash memory; however, the PROGE bit is set in the Flash Status Register (FSR). This bit is automatically cleared by a read access to the FSR register.

If the current command writes or erases a page in a locked region, or a page protected by the BOOTPROT fuses, the command has no effect on the flash memory; however, the LOCKE bit is set in the FSR register. This bit is automatically cleared by a read access to the FSR register.

8.5.1 Write/Erase Page Operation

Flash technology requires that an erase must be done before programming. The entire flash can be erased by an Erase All command. Alternatively, pages can be individually erased by the Erase Page command.

The User page can be written and erased using the mechanisms described in this chapter.

9.6.5 Status Register

Name: SR

Access Type: Read-only

Offset: 0x10

Reset Value: 0x00000400

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

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

15	14	13	12	11	10	9	8
-	-	-	-	-	IDLE	SEN	EN

7	6	5	4	3	2	1	0
RTRADR	MBERROR	URES	URKEY	URREAD	CAU	CAS	EXP

- **IDLE**
  This bit is cleared when a read or write operation to the SAU channel is started.
  This bit is set when the operation is completed and no SAU bus operations are pending.
- **SEN: SAU Setup Mode Enable**
  This bit is cleared when the SAU exits setup mode.
  This bit is set when the SAU enters setup mode.
- **EN: SAU Enabled**
  This bit is cleared when the SAU is disabled.
  This bit is set when the SAU is enabled.
- **RTRADR: RTR Address Error**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set if, in the configuration phase, an RTR was written with an illegal address, i.e. the upper 16 bits in the address were different from 0xFFFC, 0xFFFD, 0xFFFE or 0xFFFF.
- **MBERROR: Master Interface Bus Error**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set if a channel access generated a transfer on the master interface that received a bus error response from the addressed slave.
- **URES: Unlock Register Error Status**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set if an attempt was made to unlock a channel by writing to the Unlock Register while one or more error bits were set in SR. The unlock operation was aborted.
- **URKEY: Unlock Register Key Error**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set if the Unlock Register was attempted written with an invalid key.
- **URREAD: Unlock Register Read**
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set if the Unlock Register was read.

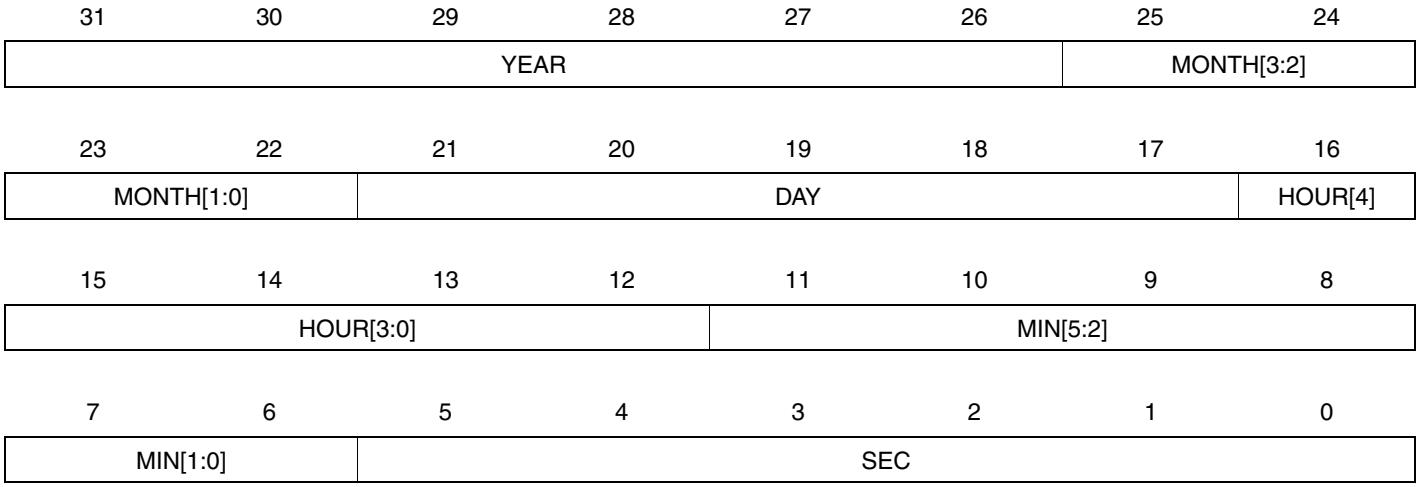
14.6.18 Calendar Value

Name: CALV

Access Type: Read/Write

Offset: 0x54

Reset Value: 0x00000000



When the SR.BUSY bit is set writes to this register will be discarded and this register will read as zero.

- **YEAR: Year**
  Current year. The year is considered a leap year if YEAR[1:0] = 0.
- **MONTH: Month**
  1 = January
  2 = February
  ...
  12 = December
- **DAY: Day**
  Day of month, starting with 1.
- **HOUR: Hour**
  Hour of day, in 24-hour clock format.
  Legal values are 0 through 23.
- **MIN: Minute**
  Minutes, 0 through 59.
- **SEC: Second**
  Seconds, 0 through 59.

14.6.19 Parameter Register

Name: PARAMETER

Access Type: Read-only

Offset: 0xF0

Reset Value: -

31	30	29	28	27	26	25	24
-	-	-	PER1VALUE				

23	22	21	20	19	18	17	16
-	-	-	PER0VALUE				

15	14	13	12	11	10	9	8
PIR1WA	PIR0WA	-	NUMPIR	-	-	NUMAR	

7	6	5	4	3	2	1	0
-	DTEXPVALUE					DTEXPWA	DT

This register gives the configuration used in the specific device. Also refer to the Module Configuration section.

- **DT: Digital Tuner**
  0: Digital tuner not implemented.
  1: Digital tuner implemented.
- **DTREXPWA: Digital Tuner Exponent Writeable**
  0: Digital tuner exponent is a constant value. Writes to EXP field in DTR will be discarded.
  1: Digital tuner exponent is chosen by writing to EXP field in DTR.
- **DTREXPVALUE: Digital Tuner Exponent Value**
  Digital tuner exponent value if DTEXPWA is zero.
- **NUMAR: Number of Alarm Comparators**
  0: Zero alarm comparators.
  1: One alarm comparator.
  2: Two alarm comparators.
- **NUMPIR: Number of Periodic Comparators**
  0: One periodic comparator.
  1: Two periodic comparator.
- **PIRnWA: Periodic Interval n Writeable**
  0: Periodic interval n prescaler tapping is a constant value. Writes to INSEL field in PIRn register will be discarded.
  1: Periodic interval n prescaler tapping is chosen by writing to INSEL field in PIRn register.
- **PERnVALUE: Periodic Interval n Value**
  Periodic interval prescaler n tapping if PIRnWA is zero.

16.7 User Interface

Table 16-2. EIC Register Memory Map

Offset	Register	Register Name	Access	Reset
0x000	Interrupt Enable Register	IER	Write-only	0x00000000
0x004	Interrupt Disable Register	IDR	Write-only	0x00000000
0x008	Interrupt Mask Register	IMR	Read-only	0x00000000
0x00C	Interrupt Status Register	ISR	Read-only	0x00000000
0x010	Interrupt Clear Register	ICR	Write-only	0x00000000
0x014	Mode Register	MODE	Read/Write	0x00000000
0x018	Edge Register	EDGE	Read/Write	0x00000000
0x01C	Level Register	LEVEL	Read/Write	0x00000000
0x020	Filter Register	FILTER	Read/Write	0x00000000
0x024	Test Register	TEST	Read/Write	0x00000000
0x028	Asynchronous Register	ASYNC	Read/Write	0x00000000
0x030	Enable Register	EN	Write-only	0x00000000
0x034	Disable Register	DIS	Write-only	0x00000000
0x038	Control Register	CTRL	Read-only	0x00000000
0x3FC	Version Register	VERSION	Read-only	- (1)

Note: 1. The reset value is device specific. Please refer to the Module Configuration section at the end of this chapter.

16.7.2 Interrupt Disable Register

Name: IDR

Access Type: Write-only

Offset: 0x004

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	INT30	INT29	INT28	INT27	INT26	INT25	INT24
23	22	21	20	19	18	17	16
INT23	INT22	INT21	INT20	INT19	INT18	INT17	INT16
15	14	13	12	11	10	9	8
INT15	INT14	INT13	INT12	INT11	INT10	INT9	INT8
7	6	5	4	3	2	1	0
INT7	INT6	INT5	INT4	INT3	INT2	INT1	NMI

- **INTn: External Interrupt n**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will clear the corresponding bit in IMR.
  Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
  Writing a zero to this bit has no effect.
  Writing a one to this bit will clear the corresponding bit in IMR.

16.7.15 Version Register

Name: VERSION

Access Type: Read-only

Offset: 0x3FC

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
-	-	-	-	VERSION[11:8]			

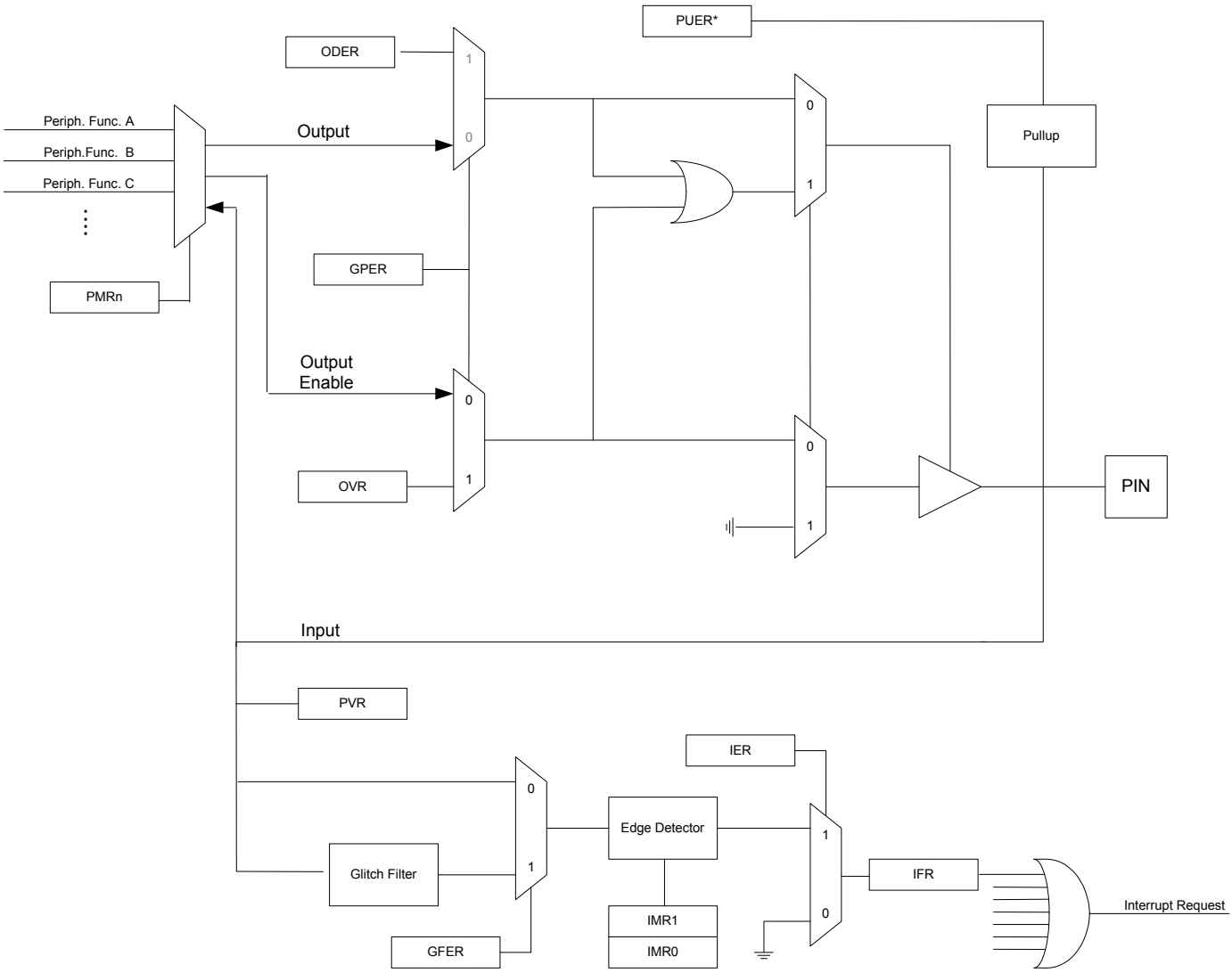
7	6	5	4	3	2	1	0
VERSION[7:0]							

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

18.6 Functional Description

The GPIO controls the I/O pins of the microcontroller. The control logic associated with each pin is shown in the figure below.

Figure 18-2. Overview of the GPIO



*) Register value is overrided if a peripheral function that support this function is enabled

19. Universal Synchronous Asynchronous Receiver Transmitter (USART)

Rev: 4.4.0.6

19.1 Features

- Configurable baud rate generator
- 5- to 9-bit full-duplex, synchronous and asynchronous, serial communication
  - 1, 1.5, or 2 stop bits in asynchronous mode, and 1 or 2 in synchronous mode
  - Parity generation and error detection
  - Framing- and overrun error detection
  - MSB- or LSB-first
  - Optional break generation and detection
  - Receiver frequency over-sampling by 8 or 16 times
  - Optional RTS-CTS hardware handshaking
  - Receiver Time-out and transmitter Timeguard
  - Optional Multidrop mode with address generation and detection
- SPI Mode
  - Master or slave
  - Configurable serial clock phase and polarity
  - CLK SPI serial clock frequency up to a quarter of the CLK_USART internal clock frequency
- LIN Mode
  - Compliant with LIN 1.3 and LIN 2.0 specifications
  - Master or slave
  - Processing of Frames with up to 256 data bytes
  - Configurable response data length, optionally defined automatically by the Identifier
  - Self synchronization in slave node configuration
  - Automatic processing and verification of the “Break Field” and “Sync Field”
  - The “Break Field” is detected even if it is partially superimposed with a data byte
  - Optional, automatic identifier parity management
  - Optional, automatic checksum management
  - Supports both “Classic” and “Enhanced” checksum types
  - Full LIN error checking and reporting
  - Frame Slot Mode: the master allocates slots to scheduled frames automatically.
  - Wakeup signal generation
- Test Modes
  - Automatic echo, remote- and local loopback
- Supports two Peripheral DMA Controller channels
  - Buffer transfers without processor intervention

19.2 Overview

The Universal Synchronous Asynchronous Receiver Transceiver (USART) provides a full duplex, universal, synchronous/asynchronous serial link. Data frame format is widely configurable, including basic length, parity, and stop bit settings, maximizing standards support. The receiver implements parity-, framing-, and overrun error detection, and can handle un-fixed frame lengths with the time-out feature. The USART supports several operating modes, providing an interface to, LIN, and SPI buses and infrared transceivers. Communication with slow and remote devices is eased by the timeguard. Duplex multidrop communication is supported by address and data differentiation through the parity bit. The hardware handshaking feature enables an out-of-band flow control, automatically managing RTS and CTS pins. The Peripheral DMA Controller connection enables memory transactions, and the USART supports chained

21.8.2.1 Clock Generation

The Clock Waveform Generator Register (CWGR) is used to control the waveform of the TWCK clock. CWGR must be written so that the desired TWI bus timings are generated. CWGR describes bus timings as a function of cycles of a prescaled clock. The clock prescaling can be selected through the Clock Prescaler field in CWGR (CWGR.EXP).

$$f_{PRESCALER} = \frac{f_{CLK\_TWIM}}{2^{(EXP+1)}}$$

CWGR has the following fields:

LOW: Prescaled clock cycles in clock low count. Used to time T_{LOW} and T_{BUF}.

HIGH: Prescaled clock cycles in clock high count. Used to time T_{HIGH}.

STASTO: Prescaled clock cycles in clock high count. Used to time T_{HD_STA}, T_{SU_STA}, T_{SU_STO}.

DATA: Prescaled clock cycles for data setup and hold count. Used to time T_{HD_DAT}, T_{SU_DAT}.

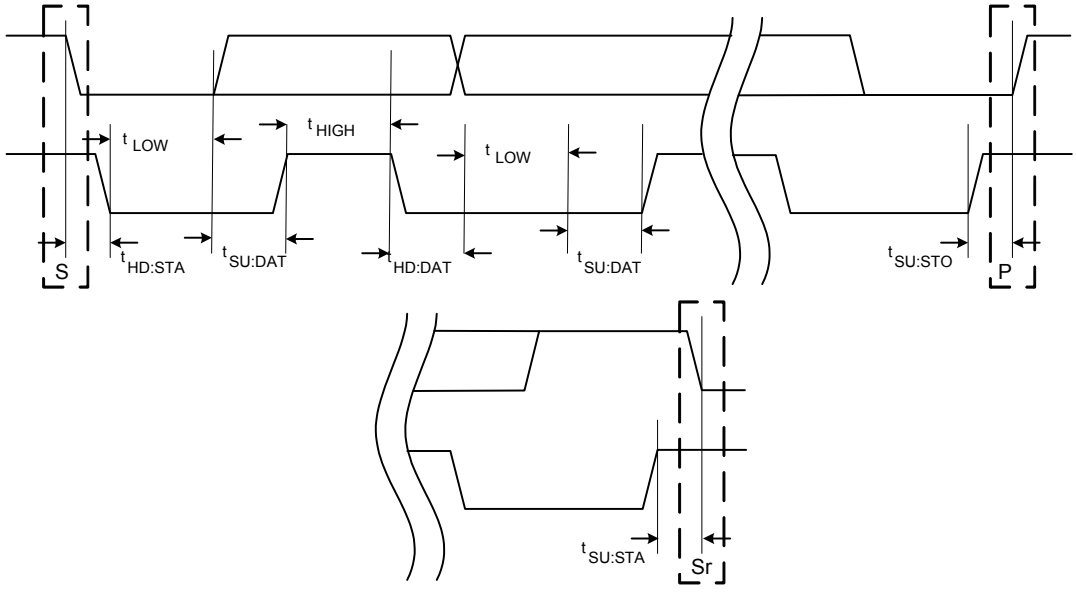
EXP: Specifies the clock prescaler setting.

Note that the total clock low time generated is the sum of $T_{HD\_DAT} + T_{SU\_DAT} + T_{LOW}$.

Any slave or other bus master taking part in the transfer may extend the TWCK low period at any time.

The TWIM hardware monitors the state of the TWCK line as required by the I²C specification. The clock generation counters are started when a high/low level is detected on the TWCK line, not when the TWIM hardware releases/drives the TWCK line. This means that the CWGR settings alone do not determine the TWCK frequency. The CWGR settings determine the clock low time and the clock high time, but the TWCK rise and fall times are determined by the external circuitry (capacitive load, etc.).

Figure 21-5. Bus Timing Diagram



21.9.6 Receive Holding Register

Name: RHR

Access Type: Read-only

Offset: 0x14

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
RXDATA							

- **RXDATA: Received Data**
  When the RXRDY bit in the Status Register (SR) is one, this field contains a byte received from the TWI bus.

23.7.2 Interlinked Single Value Duty Register

Name: ISDUTY

Access Type: Write-only

Offset: 0x04

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
DUTY							

- **DUTY: Duty Cycle Value**

  The duty cycle value written to this field is written simultaneously to all channels selected in the ISCHSET registers.

  If the value zero is written to DUTY all affected channels will be disabled. In this state the output waveform will be zero all the time.

Table 27-2. I/O Line Description

Pin Name	Pin Description	Type
ACBPn	Positive reference pin for Analog Comparator B n	Analog
ACBNn	Negative reference pin for Analog Comparator B n	Analog
ACREFN	Reference Voltage for all comparators selectable for INN	Analog

The signal names corresponds to the groups A and B of analog comparators. For normal mode, the mapping from input signal names in the block diagram to the signal names is given in Table 27-3.

Table 27-3. Signal Name Mapping

Pin Name	Channel Number	Normal Mode
ACAP0/ACAN0	0	ACP0/ACN0
ACBP0/ACBN0	1	ACP1/ACN1
ACAP1/ACAN1	2	ACP2/ACN2
ACBP1/ACBN1	3	ACP3/ACN3
ACAP2/ACAN2	4	ACP4/ACN4
ACBP2/ACBN2	5	ACP5/ACN5
ACAP3/ACAN3	6	ACP6/ACN6
ACBP3/ACBN3	7	ACP7/ACN7

27.5 Product Dependencies

In order to use this module, other parts of the system must be configured correctly, as described below.

27.5.1 I/O Lines

The ACIFB pins are multiplexed with other peripherals. The user must first program the I/O Controller to give control of the pins to the ACIFB.

27.5.2 Power Management

If the CPU enters a sleep mode that disables clocks used by the ACIFB, the ACIFB will stop functioning and resume operation after the system wakes up from sleep mode.

27.5.3 Clocks

The clock for the ACIFB bus interface (CLK_ACIFB) is generated by the Power Manager. This clock is enabled at reset, and can be disabled in the Power Manager. It is recommended to disable the ACIFB before disabling the clock, to avoid freezing the ACIFB in an undefined state.

The ACIFB uses a GCLK as clock source for the Analog Comparators. The user must set up this GCLK at the right frequency. The CLK_ACIFB clock of the interface must be at least 4x the GCLK frequency used in the comparators. The GCLK is used both for measuring the startup time of a comparator, and to give a frequency for the comparisons done in Continuous Measurement Mode, see Section 27.6.

Refer to the Electrical Characteristics chapter for GCLK frequency limitations.

27.9.6 Interrupt Status Register

Name: ISR

Access Type: Read-only

Offset: 0x1C

Reset Value: 0x00000000

31	30	29	28	27	26	25	24
-	-	-	-	WFINT3	WFINT2	WFINT1	WFINT0
23	22	21	20	19	18	17	16
-	-	-	-	-	-	-	-
15	14	13	12	11	10	9	8
SUTINT7	ACINT7	SUTINT6	ACINT6	SUTINT5	ACINT5	SUTINT4	ACINT4
7	6	5	4	3	2	1	0
SUTINT3	ACINT3	SUTINT2	ACINT2	SUTINT1	ACINT1	SUTINT0	ACINT0

- **WFINTn: Window Mode Interrupt Status**
  0: No Window Mode Interrupt is pending.
  1: Window Mode Interrupt is pending.
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding channel pair operating in window mode generated an interrupt.
- **SUTINTn: ACn Startup Time Interrupt Status**
  0: No Startup Time Interrupt is pending.
  1: Startup Time Interrupt is pending.
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the startup time of the corresponding AC has passed.
- **ACINTn: ACn Interrupt Status**
  0: No Normal Mode Interrupt is pending.
  1: Normal Mode Interrupt is pending.
  This bit is cleared when the corresponding bit in ICR is written to one.
  This bit is set when the corresponding channel generated an interrupt.

Because the CAT module is configured with Peripheral DMA Controller capability that can transfer data from memory to MBLEN and from ACOUNT to memory, the Peripheral DMA Controller can perform long acquisition sequences and store results in memory without CPU intervention.

28.6.2 Prescaler and Charge Length

Each QTouch acquisition type (autonomous QTouch, QTouch group A, and QTouch group B) has its own prescaler. Each QTouch prescaler divides down the CLK_CAT clock to an appropriate sampling frequency for its particular acquisition type. Typical frequencies are 1 MHz for QTouch acquisition and 4 MHz for QMatrix burst timing control.

Each QTouch prescaler is controlled by the DIV field in the appropriate Configuration Register 0 (ATCFG0, TGACFG0, or TGBCFG0). The QMatrix burst timing prescaler is controlled by the DIV field in MGCFG0. Each prescaler uses the following formula to generate the sampling clock:

Sampling clock = CLK_CAT / (2(DIV+1))

The capacitive sensor charge length, discharge length, and settle length can be determined for each acquisition type using the CHLEN, DILEN, and SELEN fields in Configuration Registers 0 and 1. The lengths are specified in terms of prescaler clocks. In addition, the QMatrix Cx discharge length can be determined using the CXDILEN field in MGCFG2.

For QMatrix acquisition, the duration of CHLEN should not be set to the same value as the period of any periodic signal on any other pin. If the duration of CHLEN is the same as the period of a signal on another pin, it is likely that the other signal will significantly affect measurements due to stray capacitive coupling. For example, if a 1 MHz signal is generated on another pin of the device, then CHLEN should not be 1 microsecond.

For the QMatrix method, burst and capture lengths are set for each (X,Y) pair by writing the desired length values to the MBLEN register. The write must be done before each X line can start its acquisition and is indicated by the status bit MBLREQ in the Status Register (SR). A DMA handshake interface is also connected to this status bit to reduce CPU overhead during QMatrix acquisitions.

Four burst lengths (BURST0..3) can be written at one time into the MBLEN register. If the current configuration uses Y lines larger than Y3 the register has to be written a second time. The first write to MBLEN specifies the burst length for Y lines 0 to 3 in the BURST0 to BURST3 fields, respectively. The second write specifies the burst length for Y lines 4 to 7 in fields BURST0 to BURST3, respectively, and so on.

The Y and YK pins remain clamped to ground apart from the specified number of burst pulses, when charge is transferred and captured into the sampling capacitor.

28.6.3 Capacitive Count Acquisition

For the QMatrix, QTouch group A, and QTouch group B types of acquisition, the module acquires count values from the sensors, buffers them, and makes them available for reading in the ACOUNT register. Further processing of the count values must be performed by the CPU.

When the module performs QMatrix acquisition using multiple Y lines, it starts the capture for each Y line at the appropriate time in the burst sequence so that all captures finish simultaneously. For example, suppose that an acquisition is performed on Y0 and Y1 with BURST0=53 and BURST1=60. The module will first toggle the X line 7 times while capturing on Y1 while Y0 and YK0 are clamped to ground. The module will then toggle the X line 53 times while capturing on both Y1 and Y0.

Note that the security bit will read as programmed and block these instructions also if the Flash Controller is statically reset.

Other security mechanisms can also restrict these functions. If such mechanisms are present they are listed in the SAB address map section.

31.5.1.1 Notation

Table 31-11 on page 737 shows bit patterns to be shifted in a format like "**peb01**". Each character corresponds to one bit, and eight bits are grouped together for readability. The least significantbit is always shifted first, and the most significant bit shifted last. The symbols used are shown in Table 31-10.

Table 31-10. Symbol Description

Symbol	Description
0	Constant low value - always reads as zero.
1	Constant high value - always reads as one.
a	An address bit - always scanned with the least significant bit first
b	A busy bit. Reads as one if the SAB was busy, or zero if it was not. See Section 31.4.11.4 for details on how the busy reporting works.
d	A data bit - always scanned with the least significant bit first.
e	An error bit. Reads as one if an error occurred, or zero if not. See Section 31.4.11.5 for details on how the error reporting works.
p	The chip protected bit. Some devices may be set in a protected state where access to chip internals are severely restricted. See the documentation for the specific device for details. On devices without this possibility, this bit always reads as zero.
r	A direction bit. Set to one to request a read, set to zero to request a write.
s	A size bit. The size encoding is described where used.
x	A don't care bit. Any value can be shifted in, and output data should be ignored.

In many cases, it is not required to shift all bits through the data register. Bit patterns are shown using the full width of the shift register, but the suggested or required bits are emphasized using **bold** text. I.e. given the pattern "**aaaaaaar** xxxxxxxx xxxxxxxx xxxxxxxx xx", the shift register is 34 bits, but the test or debug unit may choose to shift only 8 bits "**aaaaaaar**".

The following describes how to interpret the fields in the instruction description tables:

Table 31-11. Instruction Description

Instruction	Description
IR input value	Shows the bit pattern to shift into IR in the Shift-IR state in order to select this instruction. The pattern is show both in binary and in hexadecimal form for convenience. Example: **10000** (0x10)
IR output value	Shows the bit pattern shifted out of IR in the Shift-IR state when this instruction is active. Example: peb01

32.9 Timing Characteristics

32.9.1 Startup, Reset, and Wake-up Timing

The startup, reset, and wake-up timings are calculated using the following formula:

$$t = t_{CONST} + N_{CPU} \times t_{CPU}$$

Where t_{CONST} and N_{CPU} are found in Table 32-36. t_{CPU} is the period of the CPU clock. If another clock source than RCSYS is selected as CPU clock the startup time of the oscillator, $t_{OSCSTART}$, must added to the wake-up time in the stop, deepstop, and static sleep modes. Please refer to the source for the CPU clock in the "Oscillator Characteristics" on page 779 for more details about oscillator startup times.

Table 32-36. Maximum Reset and Wake-up Timing[1]

Parameter		Measuring	Max t_{CONST} (in µs)	Max N_{CPU}
Startup time from power-up, using regulator		Time from VDDIN crossing the V_{POT+} threshold of POR33 to the first instruction entering the decode stage of CPU. VDDCORE is supplied by the internal regulator.	2210	0
Startup time from power-up, no regulator		Time from VDDIN crossing the V_{POT+} threshold of POR33 to the first instruction entering the decode stage of CPU. VDDCORE is connected to VDDIN.	1810	0
Startup time from reset release		Time from releasing a reset source (except POR18, POR33, and SM33) to the first instruction entering the decode stage of CPU.	170	0
Wake-up	Idle	From wake-up event to the first instruction of an interrupt routine entering the decode stage of the CPU.	0	19
	Frozen		0	110
	Standby		0	110
	Stop		27 + $t_{OSCSTART}$	116
	Deepstop		27 + $t_{OSCSTART}$	116
	Static		97 + $t_{OSCSTART}$	116
Wake-up from shutdown		From wake-up event to the first instruction entering the decode stage of the CPU.	1180	0

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

32.9.2 RESET_N Timing

Table 32-37. RESET_N Waveform Parameters[1]

Symbol	Parameter	Conditions	Min	Max	Units
t_{RESET}	RESET_N minimum pulse length		10		ns

Note: 1. These values are based on simulation and characterization of other AVR microcontrollers manufactured in the same process technology. These values are not covered by test limits in production.

The TWALM signal in the TWIM is active high instead of active low.
Fix/Workaround
Use an external inverter to invert the signal going into the TWIM. When using both TWIM and TWIS on the same pins, the TWALM cannot be used.

3. **TWIS may not wake the device from sleep mode**
If the CPU is put to a sleep mode (except Idle and Frozen) directly after a TWI Start condition, the CPU may not wake upon a TWIS address match. The request is NACKed.
Fix/Workaround
When using the TWI address match to wake the device from sleep, do not switch to sleep modes deeper than Frozen. Another solution is to enable asynchronous EIC wake on the TWIS clock (TWCK) or TWIS data (TWD) pins, in order to wake the system up on bus events.

4. **SMBALERT bit may be set after reset**
The SMBus Alert (SMBALERT) bit in the Status Register (SR) might be erroneously set after system reset.
Fix/Workaround
After system reset, clear the SR.SMBALERT bit before commencing any TWI transfer.

5. **Clearing the NAK bit before the BTF bit is set locks up the TWI bus**
When the TWIS is in transmit mode, clearing the NAK Received (NAK) bit of the Status Register (SR) before the end of the Acknowledge/Not Acknowledge cycle will cause the TWIS to attempt to continue transmitting data, thus locking up the bus.
Fix/Workaround
Clear SR.NAK only after the Byte Transfer Finished (BTF) bit of the same register has been set.

6. **TWIS stretch on Address match error**
When the TWIS stretches TWCK due to a slave address match, it also holds TWD low for the same duration if it is to be receiving data. When TWIS releases TWCK, it releases TWD at the same time. This can cause a TWI timing violation.
Fix/Workaround
None.

35.2.10 PWMA

1. **BUSY bit is never cleared after writes to the Control Register (CR)**
When writing a non-zero value to CR.TOP, CR.SPREAD, or CR.TCLR when the PWMA is disabled (CR.EN == 0), the BUSY bit in the Status Register (SR.BUSY) will be set, but never cleared.
Fix/Workaround
When writing a non-zero value to CR.TOP, CR.SPREAD, or CR.TCLR, make sure the PWMA is enabled, or simultaneously enable the PWMA by writing a one to CR.EN.

2. **Incoming peripheral events are discarded during duty cycle register update**
Incoming peripheral events to all applied channels will be discarded if a duty cycle update is received from the user interface in the same PWMA clock period.
Fix/Workaround
Ensure that duty cycle writes from the user interface are not performed in a PWMA period when an incoming peripheral event is expected.