

Welcome to [E-XFL.COM](https://www.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.

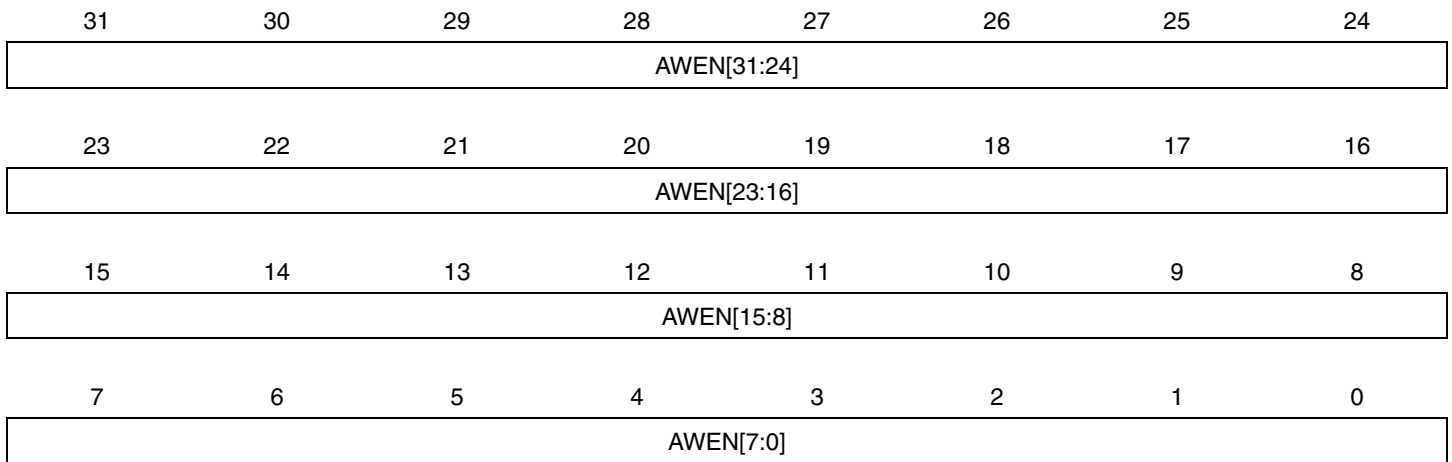
Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Active
Core Processor	AVR
Core Size	32-Bit Single-Core
Speed	50MHz
Connectivity	I ² C, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, POR, PWM, WDT
Number of I/O	36
Program Memory Size	256KB (256K 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-UFLGA Exposed Pad
Supplier Device Package	48-TLLGA (5.5x5.5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atuc256l4u-d3ht

13.7.18 Asynchronous Wake Up Enable Register

Name: AWEN
Access Type: Read/Write
Offset: 0x188
Reset Value: 0x00000000



Each bit in this register corresponds to an asynchronous wake-up source, according to [Table 13-13 on page 240](#).

0: The corresponding wake up is disabled.

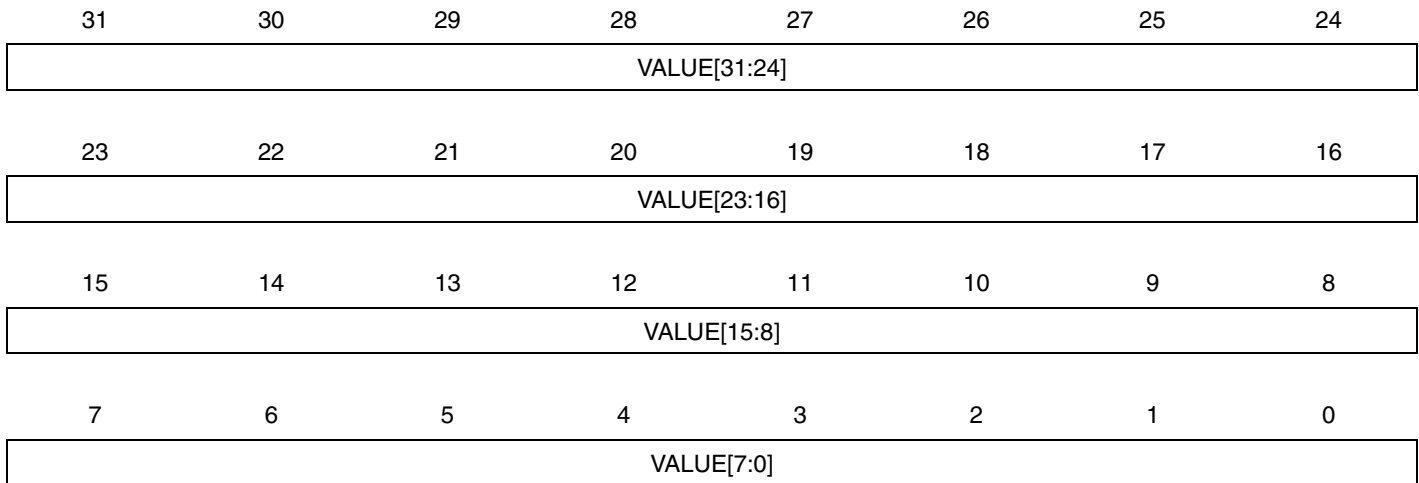
1: The corresponding wake up is enabled

Table 13-13. Asynchronous Wake-up Sources

Bit	Asynchronous Wake-up Source
0	CAT
1	ACIFB
2	ADCIFB
3	TWIS0
4	TWIS1
5	WAKEN
6	ADCIFBPD
7	USBC
31:8	-

15.6.2 Counter Value

Name: CV
Access Type: Read/Write
Offset: 0x04
Reset Value: 0x00000000



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

- **VALUE: AST Value**
The current value of the AST counter.

15.6.16 Event Disable Register

Name: EVD

Access Type: Write-only

Offset: 0x4C

Reset Value: 0x00000000

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

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

Writing a zero to a bit in this register has no effect.

Writing a one to a bit in this register will clear the corresponding bit in EVM.

16.5.3 Disabling the WDT

The WDT is disabled by writing a zero to the CTRL.EN bit. When disabling the WDT no other bits in the CTRL Register should be changed until the CTRL.EN bit reads back as zero. If the CTRL.CEN bit is written to zero, the CTRL.EN bit will never read back as zero if changing the value from one to zero.

16.5.4 Flash Calibration

The WDT can be enabled at reset. This is controlled by the WDTAUTO fuse. The WDT will be set in basic mode, RCSYS is set as source for CLK_CNT, and PSEL will be set to a value giving T_{psel} above 100 ms. Please refer to the Fuse Settings chapter for details about WDTAUTO and how to program the fuses.

If the Flash Calibration Done (FCD) bit in the CTRL Register is zero at a watchdog reset the flash calibration will be redone, and the CTRL.FCD bit will be set when the calibration is done. If CTRL.FCD is one at a watchdog reset, the configuration of the WDT will not be changed during flash calibration. After any other reset the flash calibration will always be done, and the CTRL.FCD bit will be set when the calibration is done.

16.5.5 Special Considerations

Care must be taken when selecting the PSEL/TBAN values so that the timeout period is greater than the startup time of the device. Otherwise a watchdog reset will reset the device before any code has been run. This can also be avoided by writing the CTRL.DAR bit to one when configuring the WDT.

If the Store Final Value (SFV) bit in the CTRL Register is one, the CTRL Register is locked for further write accesses. All writes to the CTRL Register will be ignored. Once the CTRL Register is locked, it can only be unlocked by a reset (e.g. POR, OCD, and WDT).

The CTRL.MODE bit can only be changed when the WDT is disabled (CTRL.EN=0).

Each external interrupt INT_n can be configured to produce an interrupt on rising or falling edge, or high or low level. External interrupts are configured by the MODE, EDGE, and LEVEL registers. Each interrupt has a bit INT_n in each of these registers. Writing a zero to the INT_n bit in the MODE register enables edge triggered interrupts, while writing a one to the bit enables level triggered interrupts.

If INT_n is configured as an edge triggered interrupt, writing a zero to the INT_n bit in the EDGE register will cause the interrupt to be triggered on a falling edge on EXTINT_n, while writing a one to the bit will cause the interrupt to be triggered on a rising edge on EXTINT_n.

If INT_n is configured as a level triggered interrupt, writing a zero to the INT_n bit in the LEVEL register will cause the interrupt to be triggered on a low level on EXTINT_n, while writing a one to the bit will cause the interrupt to be triggered on a high level on EXTINT_n.

Each interrupt has a corresponding bit in each of the interrupt control and status registers. Writing a one to the INT_n bit in the Interrupt Enable Register (IER) enables the external interrupt from pin EXTINT_n to propagate from the EIC to the interrupt controller, while writing a one to INT_n bit in the Interrupt Disable Register (IDR) disables this propagation. The Interrupt Mask Register (IMR) can be read to check which interrupts are enabled. When an interrupt triggers, the corresponding bit in the Interrupt Status Register (ISR) will be set. This bit remains set until a one is written to the corresponding bit in the Interrupt Clear Register (ICR) or the interrupt is disabled.

Writing a one to the INT_n bit in the Enable Register (EN) enables the external interrupt on pin EXTINT_n, while writing a one to INT_n bit in the Disable Register (DIS) disables the external interrupt. The Control Register (CTRL) can be read to check which interrupts are enabled. If a bit in the CTRL register is set, but the corresponding bit in IMR is not set, an interrupt will not propagate to the interrupt controller. However, the corresponding bit in ISR will be set, and EIC_WAKE will be set. Note that an external interrupt should not be enabled before it has been configured correctly.

If the CTRL.INT_n bit is zero, the corresponding bit in ISR will always be zero. Disabling an external interrupt by writing a one to the DIS.INT_n bit will clear the corresponding bit in ISR.

Please refer to the Module Configuration section for the number of external interrupts.

17.6.2 Synchronization and Filtering of External Interrupts

In synchronous mode the pin value of the EXTINT_n pin is synchronized to CLK_SYNC, so spikes shorter than one CLK_SYNC cycle are not guaranteed to produce an interrupt. The synchronization of the EXTINT_n to CLK_SYNC will delay the propagation of the interrupt to the interrupt controller by two cycles of CLK_SYNC, see [Figure 17-2](#) and [Figure 17-3](#) for examples (FILTER off).

It is also possible to apply a filter on EXTINT_n by writing a one to the INT_n bit in the FILTER register. This filter is a majority voter, if the condition for an interrupt is true for more than one of the latest three cycles of CLK_SYNC the interrupt will be set. This will additionally delay the propagation of the interrupt to the interrupt controller by one or two cycles of CLK_SYNC, see [Figure 17-2](#) and [Figure 17-3](#) for examples (FILTER on).

Figure 17-2. Timing Diagram, Synchronous Interrupts, High Level or Rising Edge

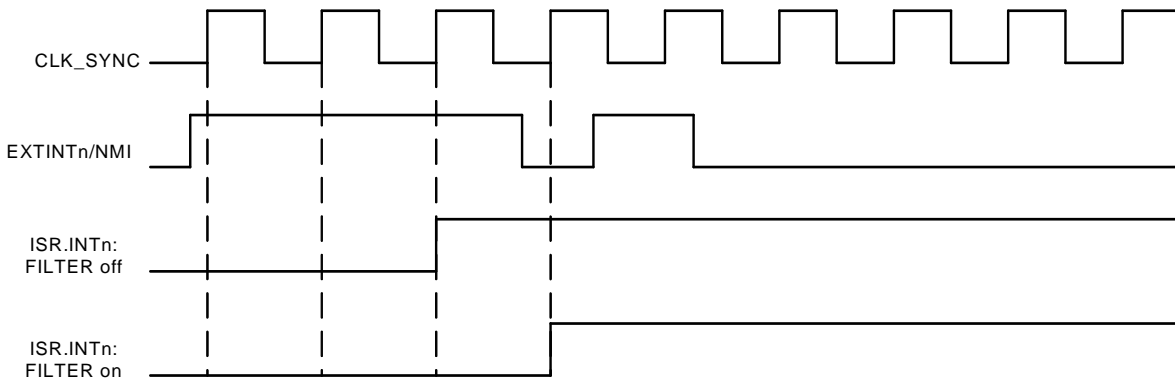
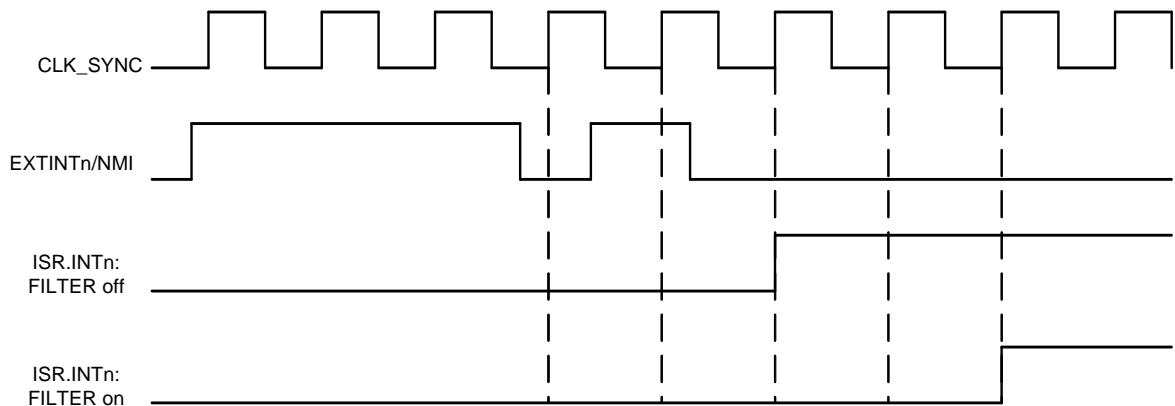


Figure 17-3. Timing Diagram, Synchronous Interrupts, Low Level or Falling Edge



17.6.3 Non-Maskable Interrupt

The NMI supports the same features as the external interrupts, and is accessed through the same registers. The description in [Section 17.6.1](#) should be followed, accessing the NMI bit instead of the INTn bits.

The NMI is non-maskable within the CPU in the sense that it can interrupt any other execution mode. Still, as for the other external interrupts, the actual NMI input can be enabled and disabled by accessing the registers in the EIC.

17.6.4 Asynchronous Interrupts

Each external interrupt can be made asynchronous by writing a one to INTn in the ASYNC register. This will route the interrupt signal through the asynchronous path of the module. All edge interrupts will be interpreted as level interrupts and the filter is disabled. If an interrupt is configured as edge triggered interrupt in asynchronous mode, a zero in EDGE.INTn will be interpreted as low level, and a one in EDGE.INTn will be interpreted as high level.

EIC_WAKE will be set immediately after the source triggers the interrupt, while the corresponding bit in ISR and the interrupt to the interrupt controller will be set on the next rising edge of CLK_SYNC. Please refer to [Figure 17-4 on page 369](#) for details.

17.7.9 Filter Register

Name: FILTER
Access Type: Read/Write
Offset: 0x020
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**
 0: The external interrupt is not filtered.
 1: The external interrupt is filtered.
 Please refer to the Module Configuration section for the number of external interrupts.
- **NMI: Non-Maskable Interrupt**
 0: The Non-Maskable Interrupt is not filtered.
 1: The Non-Maskable Interrupt is filtered.

19.7.15 Glitch Filter Enable Register

Name: GFER

Access: Read/Write, Set, Clear, Toggle

Offset: 0x0C0, 0x0C4, 0x0C8, 0x0CC

Reset Value: -

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

- P0-31: Glitch Filter Enable**

0: Glitch filter is disabled for the corresponding pin.

1: Glitch filter is enabled for the corresponding pin.

NOTE! The value of this register should only be changed when the corresponding bit in IER is zero. Updating GFER while interrupt on the corresponding pin is enabled can cause an unintentional interrupt to be triggered.

CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with NCPHA to produce the required clock/data relationship between master and slave devices.

Table 22-2 lists the compatibility level of the Atmel Two-wire Master Interface and a full SMBus compatible master.

Table 22-2. Atmel TWIM Compatibility with SMBus Standard

SMBus Standard	Atmel TWIM
Bus Timeouts	Supported
Address Resolution Protocol	Supported
Alert	Supported
Host Functionality	Supported
Packet Error Checking	Supported

22.3 List of Abbreviations

Table 22-3. Abbreviations

Abbreviation	Description
TWI	Two-wire Interface
A	Acknowledge
NA	Non Acknowledge
P	Stop
S	Start
Sr	Repeated Start
SADR	Slave Address
ADR	Any address except SADR
R	Read
W	Write

22.4 Block Diagram

Figure 22-1. Block Diagram

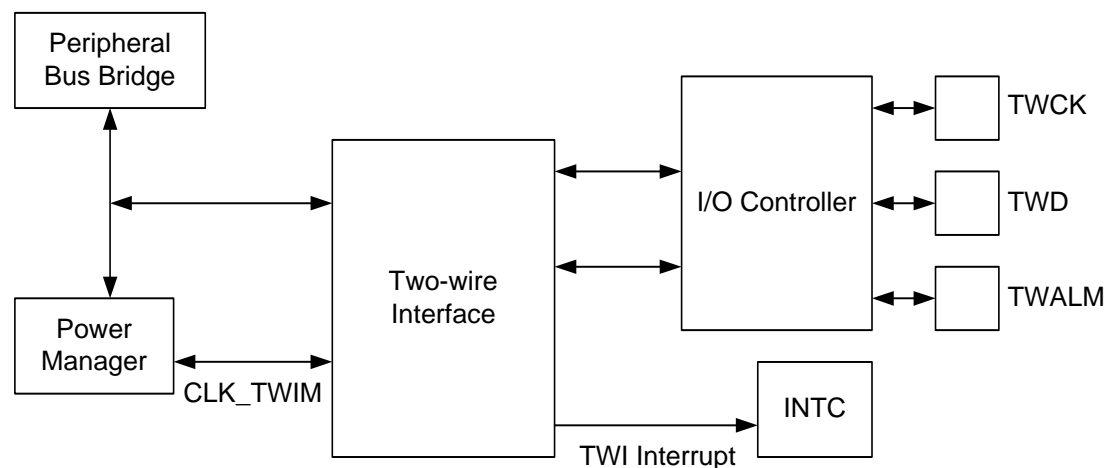


Figure 24-6. Codec Application Block Diagram

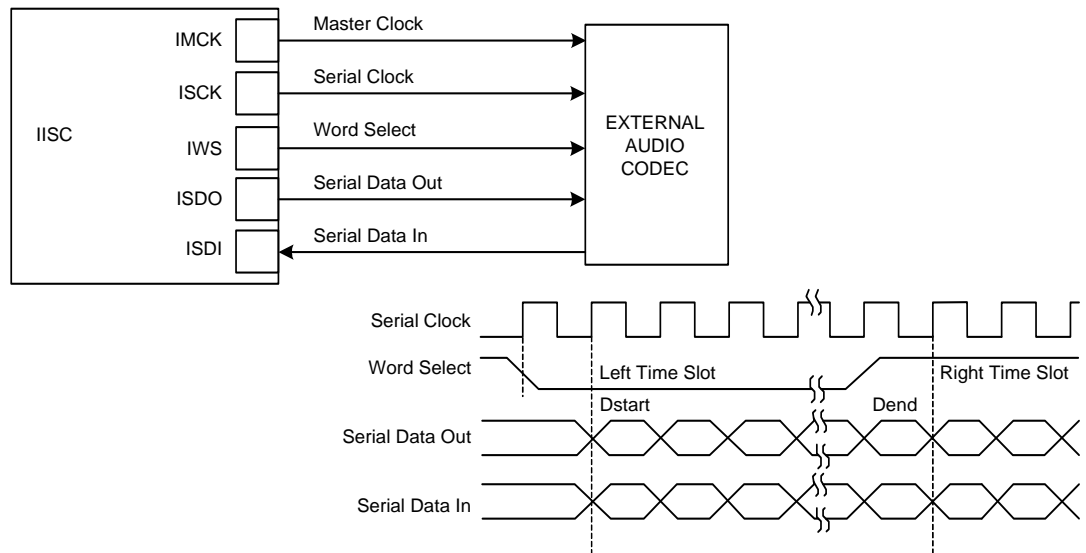
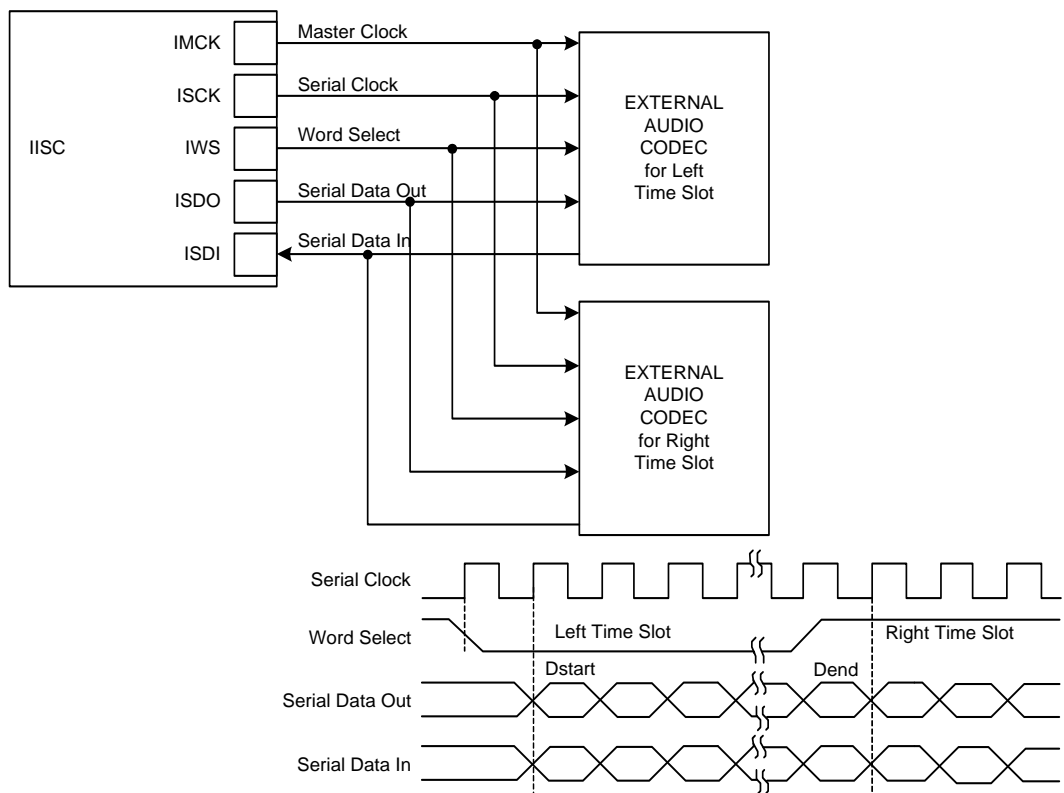


Figure 24-7. Time Slot Application Block Diagram



SPREAD field of CR (CR.SPREAD) is zero). When the timebase counter reaches its effective top value, it restarts counting from zero. The period of the PWMA output waveform is then:

$$T_{PWMA} = (ETV + 1) \cdot T_{CLK}$$

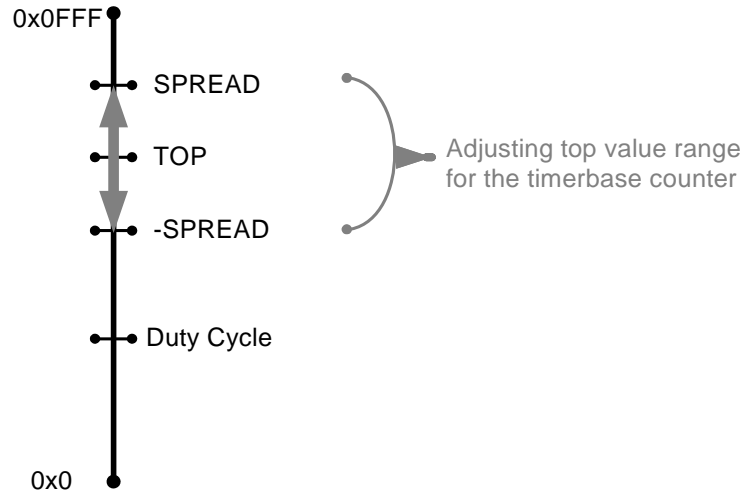
The timebase counter can be reset by writing a one to the Timebase Clear bit in CR (CR.TCLR). Note that this can cause a glitch to the output PWM waveforms in use.

25.6.3 Spread Spectrum Counter

The spread spectrum counter allows the generation of constantly varying duty cycles on the output PWM waveforms. This is achieved by varying the effective top value of the timebase counter in a range defined by the spread spectrum counter value.

When CR.SPREAD is not zero, the spread spectrum counter is enabled. Its range is defined by CR.SPREAD. It starts to count from -CR.SPREAD when the PWMA is enabled or after reset and counts upwards. When it reaches CR.SPREAD, it restarts to count from -CR.SPREAD again. The spread spectrum counter will cause the effective top value to vary from TOP-SPREAD to TOP+SPREAD. [Figure 25-2 on page 618](#) illustrates this. This leads to a constantly varying duty cycle on the PWM output waveforms though the duty cycle values stored are unchanged.

Figure 25-2. PWMA Adjusting Top Value for Timebase Counter



25.6.3.1 Special considerations

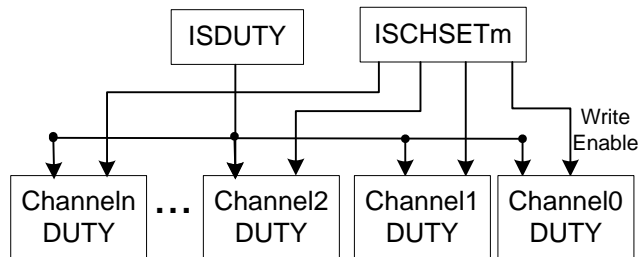
The maximum value of the timebase counter is 0x0FFF. If SPREAD is written to a value that will cause the ETV to exceed this value, the spread spectrum counter's range will be limited to prevent the timebase counter to exceed its maximum value.

If SPREAD is written to a value causing (TOP-SPREAD) to be below zero, the spread spectrum counter's range will be limited to prevent the timebase counter to count below zero.

In both cases, the SPREAD value read from the Control Register will be the same value as written to the SPREAD field.

(ISDUTY) register. Each channel has a corresponding enabling bit in the Interlinked Single Value Channel Set (ISCHSET) register(s). When a bit is written to one in the ISCHSET register, the duty cycle register for the corresponding channel will be updated with the value stored in the ISDUTY register. It can only be updated when the READY bit in the Status Register (SR.READY) is one, indicating that the PWMA is ready for writing. [Figure 25-3 on page 620](#) shows the writing procedure. It is thus possible to update the duty cycle values for up to 32 PWM channels within one ISCHSET register at a time.

Figure 25-3. Interlinked Single Value PWM Operation Flow



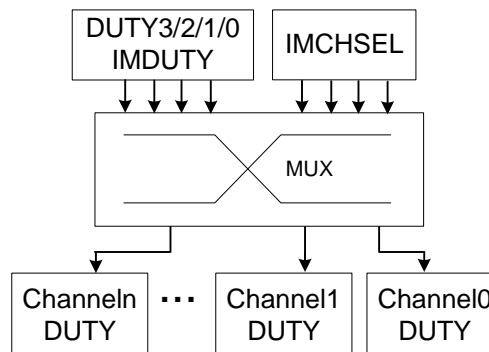
25.6.6.2 Interlinked Multiple Value PWM Operation

The interlinked multiple value PWM operation allows up to four channels to be updated simultaneously with different duty cycle values. The four duty cycle values are required to be written to the four registers, DUTY3, DUTY2, DUTY1 and DUTY0, respectively. The index number of the four channels to be updated is written to the four SEL fields in the Interlinked Multiple Value Channel Select (IMCHSEL) register (IMCHSEL.SEL). When the IMCHSEL register is written, the values stored in the DUTY0/1/2/3 registers are synchronized to the duty cycle registers for the channels selected by the SEL fields. [Figure 25-4 on page 620](#) shows the writing procedure.

Note that only writes to the implemented channels will be effective. If one of the IMCHSEL.SEL fields points to a non-existing channel, the corresponding value in the DUTYx register will not be written. If the same channel is specified multiple times in the IMCHSEL.SEL fields, the channel will be updated with the value referred by the upper IMCHSEL.SEL field.

When only the least significant 8-bits duty cycle value are considered for updating, the four duty cycle values can be written to the IMDUTY register once. This is equivalent to writing the four duty cycle values to the four DUTY registers one by one.

Figure 25-4. Interlinked Multiple Value PWM Operation Flow



In Waveform mode, TIOA is always configured to be an output and TIOB is an output if it is not selected to be the external trigger.

26.6.1.6 *Trigger*

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

The following triggers are common to both modes:

- **Software Trigger:** each channel has a software trigger, available by writing a one to the Software Trigger Command bit in CCRn (CCRn.SWTRG).
- **SYNC:** each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing a one to the Synchro Command bit in the BCR register (BCR.SYNC).
- **Compare RC Trigger:** RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if the RC Compare Trigger Enable bit in CMRn (CMRn.CPCTRG) is written to one.

The channel can also be configured to have an external trigger. In Capture mode, the external trigger signal can be selected between TIOA and TIOB. In Waveform mode, an external event can be programmed to be one of the following signals: TIOB, XC0, XC1, or XC2. This external event can then be programmed to perform a trigger by writing a one to the External Event Trigger Enable bit in CMRn (CMRn.ENETRG).

If an external trigger is used, the duration of the pulses must be longer than the CLK_TC period in order to be detected.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

26.6.1.7 *Peripheral events on TIOA inputs*

The TIOA input lines are ored internally with peripheral events from the Peripheral Event System. To capture using events the user must ensure that the corresponding pin functions for the TIOA line are disabled. When capturing on the external TIOA pin the user must ensure that no peripheral events are generated on this pin.

26.6.2 **Capture Operating Mode**

This mode is entered by writing a zero to the CMRn.WAVE bit.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOA and TIOB signals which are considered as inputs.

[Figure 26-4 on page 649](#) shows the configuration of the TC channel when programmed in Capture mode.

26.6.2.1 *Capture registers A and B*

Registers A and B (RA and RB) are used as capture registers. This means that they can be loaded with the counter value when a programmable event occurs on the signal TIOA.

0: TIOB is used as an external trigger.

- **ETRGEDG: External Trigger Edge Selection**

ETRGEDG	Edge
0	none
1	rising edge
2	falling edge
3	each edge

- **LDBDIS: Counter Clock Disable with RB Loading**

1: Counter clock is disabled when RB loading occurs.

0: Counter clock is not disabled when RB loading occurs.

- **LDBSTOP: Counter Clock Stopped with RB Loading**

1: Counter clock is stopped when RB loading occurs.

0: Counter clock is not stopped when RB loading occurs.

- **BURST: Burst Signal Selection**

BURST	Burst Signal Selection
0	The clock is not gated by an external signal
1	XC0 is ANDed with the selected clock
2	XC1 is ANDed with the selected clock
3	XC2 is ANDed with the selected clock

- **CLKI: Clock Invert**

1: The counter is incremented on falling edge of the clock.

0: The counter is incremented on rising edge of the clock.

- **TCCLKS: Clock Selection**

TCCLKS	Clock Selected
0	TIMER_CLOCK1
1	TIMER_CLOCK2
2	TIMER_CLOCK3
3	TIMER_CLOCK4
4	TIMER_CLOCK5
5	XC0
6	XC1
7	XC2

28.4 I/O Lines Description

Table 28-1. I/O Lines Description

Pin Name	Pin Description	Type
DAC[0]	Output for channel 0	Output
DACN[0]	Inverted output for channel 0	Output
DAC[1]	Output for channel 1	Output
DACN[1]	Inverted output for channel 1	Output
CLK	Clock output for DAC	Output

28.5 Product Dependencies

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

28.5.1 I/O lines

The output pins used for the output bitstream from the Audio Bitstream DAC may be multiplexed with I/O Controller lines.

Before using the Audio Bitstream DAC, the I/O Controller must be configured in order for the Audio Bitstream DAC I/O lines to be in Audio Bitstream DAC peripheral mode.

28.5.2 Clocks

The clock for the ABDACB bus interface (CLK_ABDACB) is generated by the Power Manager. This clock is turned on by default, and can be enabled and disabled in the Power Manager. It is recommended to disable the ABDACB before disabling the clock, to avoid freezing the ABDACB in an undefined state. Before using the Audio Bitstream DAC, the user must ensure that the Audio Bitstream DAC clock is enabled in the Power Manager.

The Audio Bitstream DAC requires a separate clock for the D/A conversion. This clock is provided by a generic clock which has to be set up in the System Control Interface (SCIF). The frequency for this clock has to be set as described in [Table 28-3 on page 697](#). It is important that this clock is accurate and has low jitter. Incorrect frequency will result in too fast or too slow playback (frequency shift), and too high jitter will add noise to the D/A conversion. For best performance one should trade frequency accuracy (within some limits) for low jitter to obtain the best performance as jitter will have large impact on the quality of the converted signal.

28.5.3 DMA

The ABDACB is connected to the Peripheral DMA controller. Using DMA to transfer data samples requires the Peripheral DMA controller to be programmed before enabling the ABDACB.

28.5.4 Interrupts

The ABDACB interrupt request line is connected to the interrupt controller. Using the ABDACB interrupt requires the interrupt controller to be programmed first.

which is linked to the JTAG through a bus master module, which also handles synchronization between the TCK and SAB clocks.

For more information about the SAB and a list of SAB slaves see the Service Access Bus chapter.

34.4.11.1 SAB Address Mode

The MEMORY_SIZED_ACCESS instruction allows a sized read or write to any 36-bit address on the bus. MEMORY_WORD_ACCESS is a shorthand instruction for 32-bit accesses to any 36-bit address, while the NEXUS_ACCESS instruction is a Nexus-compliant shorthand instruction for accessing the 32-bit OCD registers in the 7-bit address space reserved for these. These instructions require two passes through the Shift-DR TAP state: one for the address and control information, and one for data.

34.4.11.2 Block Transfer

To increase the transfer rate, consecutive memory accesses can be accomplished by the MEMORY_BLOCK_ACCESS instruction, which only requires a single pass through Shift-DR for data transfer only. The address is automatically incremented according to the size of the last SAB transfer.

34.4.11.3 Canceling a SAB Access

It is possible to abort an ongoing SAB access by the CANCEL_ACCESS instruction, to avoid hanging the bus due to an extremely slow slave.

34.4.11.4 Busy Reporting

As the time taken to perform an access may vary depending on system activity and current chip frequency, all the SAB access JTAG instructions can return a busy indicator. This indicates whether a delay needs to be inserted, or an operation needs to be repeated in order to be successful. If a new access is requested while the SAB is busy, the request is ignored.

The SAB becomes busy when:

- Entering Update-DR in the address phase of any read operation, e.g., after scanning in a NEXUS_ACCESS address with the read bit set.
- Entering Update-DR in the data phase of any write operation, e.g., after scanning in data for a NEXUS_ACCESS write.
- Entering Update-DR during a MEMORY_BLOCK_ACCESS.
- Entering Update-DR after scanning in a counter value for SYNC.
- Entering Update-IR after scanning in a MEMORY_BLOCK_ACCESS if the previous access was a read and data was scanned after scanning the address.

The SAB becomes ready again when:

- A read or write operation completes.
- A SYNC countdown completed.
- A operation is cancelled by the CANCEL_ACCESS instruction.

What to do if the busy bit is set:

- During Shift-IR: The new instruction is selected, but the previous operation has not yet completed and will continue (unless the new instruction is CANCEL_ACCESS). You may

34.5.2.4 INTEST

This instruction selects the boundary-scan chain as Data Register for testing internal logic in the device. The logic inputs are determined by the boundary-scan chain, and the logic outputs are captured by the boundary-scan chain. The device output pins are driven from the boundary-scan chain.

Starting in Run-Test/Idle, the INTEST instruction is accessed the following way:

1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. In Update-IR: The data from the boundary-scan chain is applied to the internal logic inputs.
5. Return to Run-Test/Idle.
6. Select the DR Scan path.
7. In Capture-DR: The data on the internal logic is sampled into the boundary-scan chain.
8. In Shift-DR: The boundary-scan chain is shifted by the TCK input.
9. In Update-DR: The data from the boundary-scan chain is applied to internal logic inputs.
10. Return to Run-Test/Idle.

Table 34-14. INTEST Details

Instructions	Details
IR input value	00100 (0x04)
IR output value	p0001
DR Size	Depending on boundary-scan chain, see BSDL-file.
DR input value	Depending on boundary-scan chain, see BSDL-file.
DR output value	Depending on boundary-scan chain, see BSDL-file.

34.5.2.5 CLAMP

This instruction selects the Bypass register as Data Register. The device output pins are driven from the boundary-scan chain.

Starting in Run-Test/Idle, the CLAMP instruction is accessed the following way:

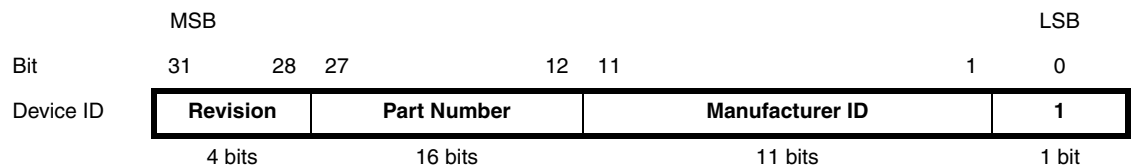
1. Select the IR Scan path.
2. In Capture-IR: The IR output value is latched into the shift register.
3. In Shift-IR: The instruction register is shifted by the TCK input.
4. In Update-IR: The data from the boundary-scan chain is applied to the output pins.
5. Return to Run-Test/Idle.
6. Select the DR Scan path.
7. In Capture-DR: A logic '0' is loaded into the Bypass Register.
8. In Shift-DR: Data is scanned from TDI to TDO through the Bypass register.

34.5.4 JTAG Data Registers

The following device specific registers can be selected as JTAG scan chain depending on the instruction loaded in the JTAG Instruction Register. Additional registers exist, but are implicitly described in the functional description of the relevant instructions.

34.5.4.1 Device Identification Register

The Device Identification Register contains a unique identifier for each product. The register is selected by the IDCODE instruction, which is the default instruction after a JTAG reset.



Revision This is a 4 bit number identifying the revision of the component.
Rev A = 0x0, B = 0x1, etc.

Part Number The part number is a 16 bit code identifying the component.

Manufacturer ID The Manufacturer ID is a 11 bit code identifying the manufacturer.
The JTAG manufacturer ID for ATMEL is 0x01F.

Device specific ID codes

The different device configurations have different JTAG ID codes, as shown in [Table 34-28](#). Note that if the flash controller is statically reset, the ID code will be undefined.

Table 34-28. Device and JTAG ID

Device Name	JTAG ID Code (R is the revision number)
ATUC256L3U	0xr21C303F
ATUC128L3U	0xr21C403F
ATUC64L3U	0xr21C503F
ATUC256L4U	0xr21C603F
ATUC128L4U	0xr21C703F
ATUC64L4U	0xr21C803F

34.5.4.2 Reset Register

The reset register is selected by the AVR_RESET instruction and contains one bit for each reset domain in the device. Setting each bit to one will keep that domain reset until the bit is cleared.



30.9	User Interface	755
30.10	Module Configuration	769
31	Capacitive Touch Module (CAT)	770
31.1	Features	770
31.2	Overview	770
31.3	Block Diagram	771
31.4	I/O Lines Description	771
31.5	Product Dependencies	772
31.6	Functional Description	774
31.7	User Interface	781
31.8	Module Configuration	816
32	Glue Logic Controller (GLOC)	817
32.1	Features	817
32.2	Overview	817
32.3	Block Diagram	817
32.4	I/O Lines Description	818
32.5	Product Dependencies	818
32.6	Functional Description	818
32.7	User Interface	820
32.8	Module Configuration	825
33	aWire UART (AW)	826
33.1	Features	826
33.2	Overview	826
33.3	Block Diagram	826
33.4	I/O Lines Description	827
33.5	Product Dependencies	827
33.6	Functional Description	827
33.7	User Interface	830
33.8	Module Configuration	843
34	Programming and Debugging	844
34.1	Overview	844
34.2	Service Access Bus	844
34.3	On-Chip Debug	847
34.4	JTAG and Boundary-scan (JTAG)	855
34.5	JTAG Instruction Summary	863