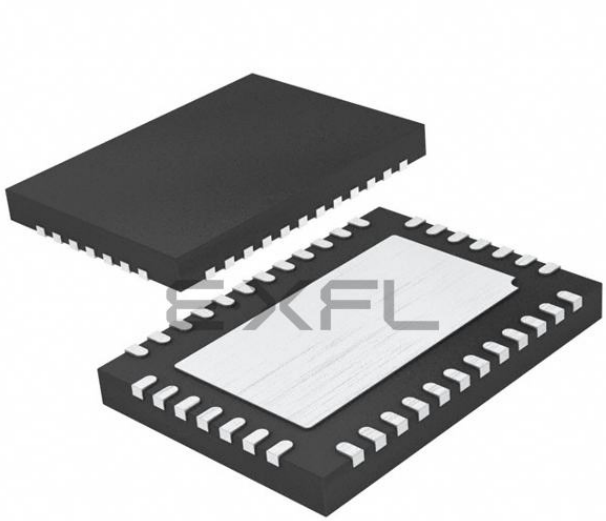


Welcome to E-XFL.COM

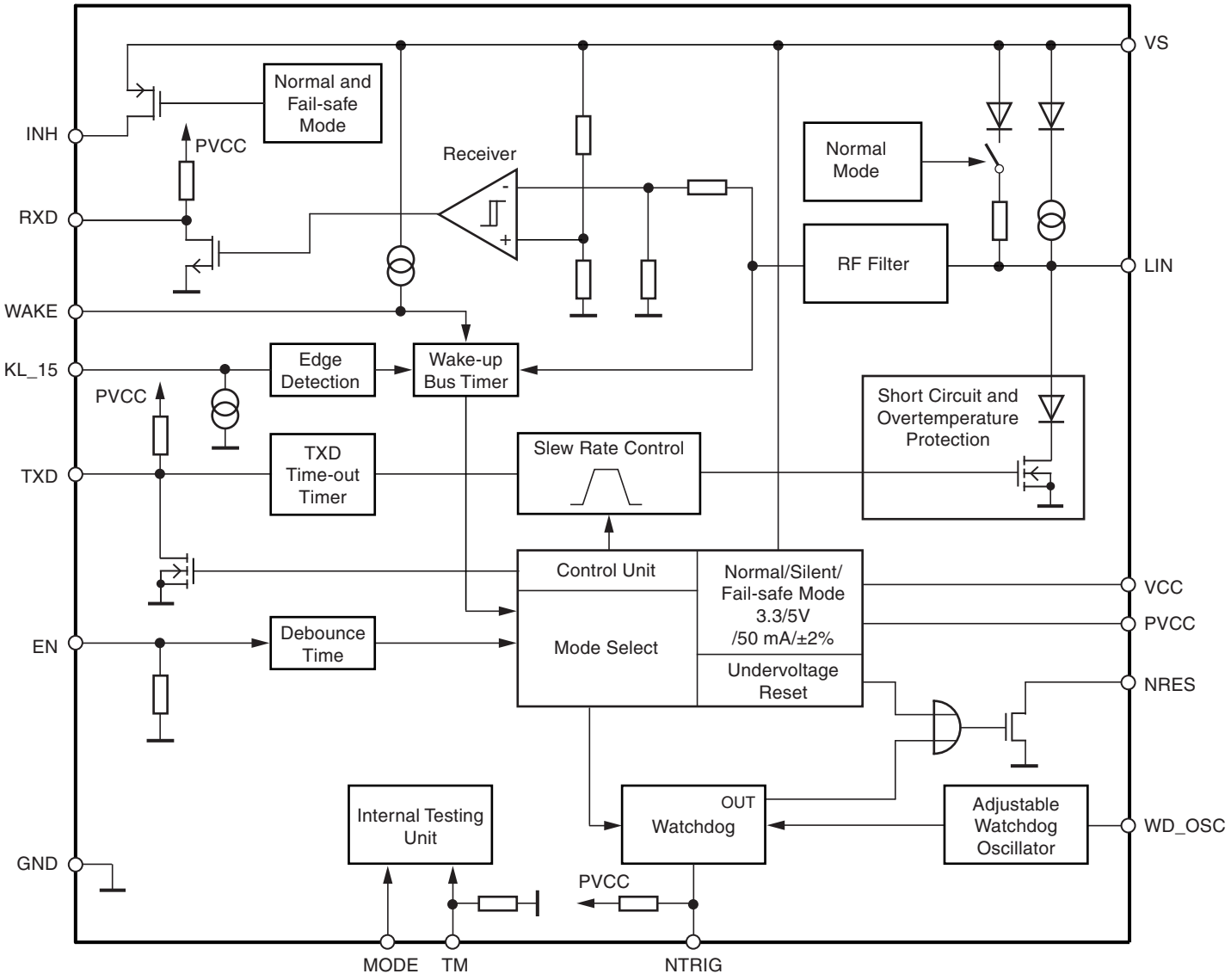
What is "Embedded - Microcontrollers"?

"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	8-Bit
Speed	16MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TC)
Mounting Type	Surface Mount
Package / Case	38-VFQFN Exposed Pad
Supplier Device Package	38-VQFN (5x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/ata6616-p3qw

Figure 3-1. Block Diagram



3.8 Electrical Characteristics (Continued)

5V < V_S < 27V, –40°C < T_{case} < 125°C, –40°C < T_j < 150°C, unless otherwise specified. All values refer to GND pins

No.	Parameters	Test Conditions	Pin	Symbol	Min.	Typ.	Max.	Unit	Type*
13	**Watchdog Oscillator**								
13.1	Voltage at WD_OSC in Normal Mode	I_{WD_OSC} = –200µA $V_{VS} \geq$ 4V	WD_OSC	V_{WD_OSC}	1.13	1.23	1.33	V	A
13.2	Possible values of resistor		WD_OSC	R_{OSC}	34		120	kΩ	A
13.3	Oscillator period	R_{OSC} = 34kΩ		t_{OSC}	10.65	13.3	15.97	µs	A
13.4	Oscillator period	R_{OSC} = 51kΩ		t_{OSC}	15.68	19.6	23.52	µs	A
13.5	Oscillator period	R_{OSC} = 91kΩ		t_{OSC}	26.83	33.5	40.24	µs	A
13.6	Oscillator period	R_{OSC} = 120kΩ		t_{OSC}	34.2	42.8	51.4	µs	A
14	**Watchdog Timing Relative to t_{osc}**								
14.1	Watchdog lead time after reset			t_d		7895		cycles	A
14.2	Watchdog closed window			t_1		1053		cycles	A
14.3	Watchdog open window			t_2		1105		cycles	A
14.4	Watchdog reset time NRES		NRES	t_{nres}	3.2	4	4.8	ms	A
15	**KL_15 Pin**								
15.1	High-level input voltage R_V = 47 kΩ	Positive edge initializes a wake-up	KL_15	V_{KL_15H}	4		V_S + 0.3V	V	A
15.2	Low-level input voltage R_V = 47 kΩ		KL_15	V_{KL_15L}	–1		+2	V	A
15.3	KL_15 pull-down current	V_S < 27V V_{KL_15} = 27V	KL_15	I_{KL_15}		50	65	µA	A
15.4	Internal debounce time	Without external capacitor	KL_15	Tdb_{KL_15}	80	160	250	µs	A
15.5	KL_15 wake-up time	R_V = 47kΩ, C = 100nF	KL_15	Tw_{KL_15}	0.4	2	4.5	ms	C
16	**WAKE Pin**								
16.1	High-level input voltage		WAKE	V_{WAKEH}	V_S – 1V		V_S + 0.3V	V	A
16.2	Low-level input voltage	Initializes a wake-up signal	WAKE	V_{WAKEL}	–1		V_S – 3.3V	V	A
16.3	WAKE pull-up current	V_S < 27V V_{WAKE} = 0V	WAKE	I_{WAKE}	–30	–10		µA	A
16.4	High-level leakage current	V_S = 27V V_{WAKE} = 27V	WAKE	I_{WAKEL}	–5		+5	µA	A
16.5	Time of low pulse for wake-up via WAKE pin	V_{WAKE} = 0V	WAKE	I_{WAKEL}	30	70	150	µs	A

*) Type means: A = 100% tested, B = 100% correlation tested, C = Characterized on samples, D = Design parameter

4.2.7 About Code Examples

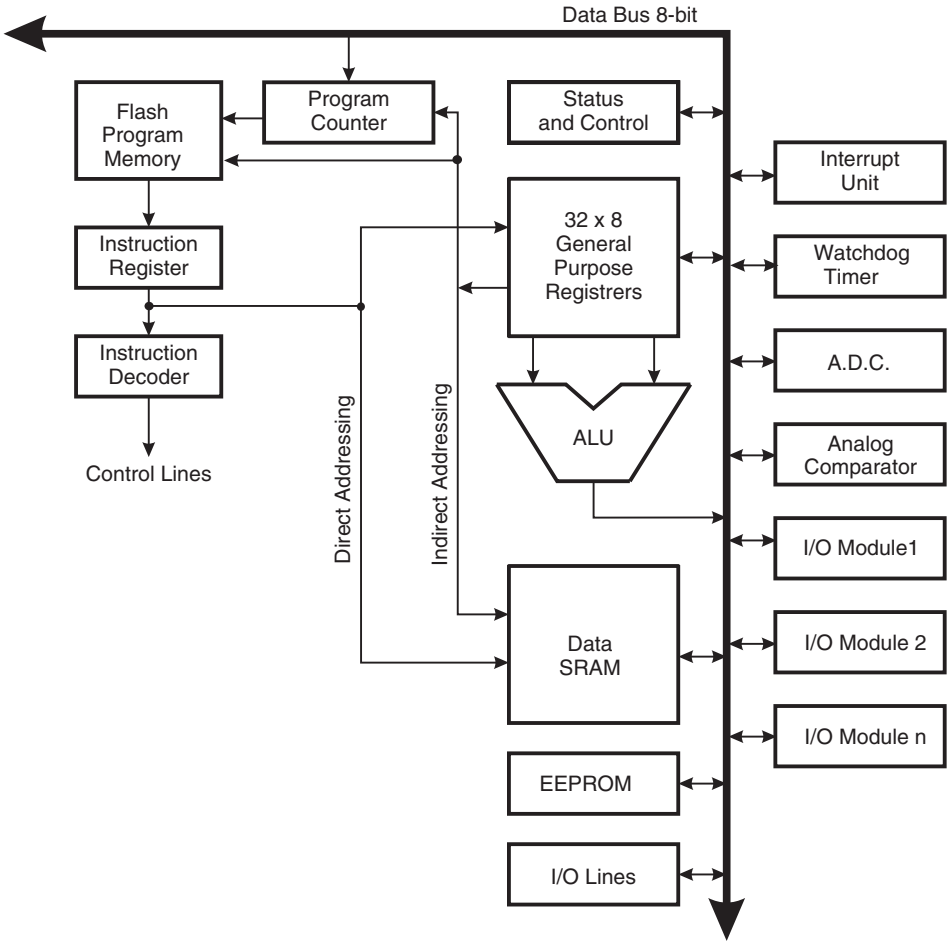
This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

4.3 AVR CPU Core

4.3.1 Overview

This section discusses the AVR® core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Figure 4-2. Block Diagram of the AVR Architecture



4.4.3.1 *EEPROM Read/Write Access*

The EEPROM Access Registers are accessible in the I/O space.

The write access times for the EEPROM are given in Table 4-4. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken. In heavily filtered power supplies, Vcc is likely to rise or fall slowly on Power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See “Preventing EEPROM Corruption” on page 44 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to “Atomic Byte Programming” on page 42 and “Split Byte Programming” on page 42 for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

4.4.3.2 *Atomic Byte Programming*

Using Atomic Byte Programming is the simplest mode. When writing a byte to the EEPROM, the user must write the address into the EEARL Register and data into EEDR Register. If the EEPMn bits are zero, writing EEPE (within four cycles after EEMPE is written) will trigger the erase/write operation. Both the erase and write cycle are done in one operation and the total programming time is given in Table 1. The EEPE bit remains set until the erase and write operations are completed. While the device is busy with programming, it is not possible to do any other EEPROM operations.

4.4.3.3 *Split Byte Programming*

It is possible to split the erase and write cycle in two different operations. This may be useful if the system requires short access time for some limited period of time (typically if the power supply voltage falls). In order to take advantage of this method, it is required that the locations to be written have been erased before the write operation. But since the erase and write operations are split, it is possible to do the erase operations when the system allows doing time-critical operations (typically after Power-up).

4.4.3.4 *Erase*

To erase a byte, the address must be written to EEAR. If the EEPMn bits are 0b01, writing the EEPE (within four cycles after EEMPE is written) will trigger the erase operation only (programming time is given in Table 1). The EEPE bit remains set until the erase operation completes. While the device is busy programming, it is not possible to do any other EEPROM operations.

4.4.3.5 *Write*

To write a location, the user must write the address into EEAR and the data into EEDR. If the EEPMn bits are 0b10, writing the EEPE (within four cycles after EEMPE is written) will trigger the write operation only (programming time is given in Table 1). The EEPE bit remains set until the write operation completes. If the location to be written has not been erased before write, the data that is stored must be considered as lost. While the device is busy with programming, it is not possible to do any other EEPROM operations.

The calibrated Oscillator is used to time the EEPROM accesses. Make sure the Oscillator frequency is within the requirements described in Section 4.5.5.1 “OSCCAL – Oscillator Calibration Register” on page 61.

The following code examples show one assembly and one C function for erase, write, or atomic write of the EEPROM. The examples assume that interrupts are controlled (e.g., by disabling interrupts globally) so that no interrupts will occur during execution of these functions.

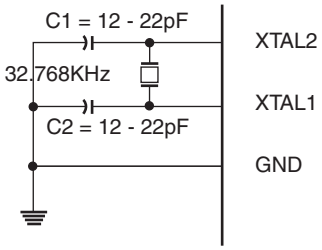
Assembly Code Example

```
EEPROM_write:
  ; Wait for completion of previous write
  sbic   EECR,EEPE
  rjmp   EEPROM_write
  ; Set Programming mode
  ldi    r16, (0<<EEPM1)|(0<<EEPM0)
  out    EECR, r16
  ; Set up address (r18:r17) in address register
  out    EEARH, r18
  out    EEARL, r17
  ; Write data (r16) to data register
  out    EEDR, r16
  ; Write logical one to EEMPE
  sbi    EECR,EEMPE
  ; Start eeprom write by setting EEPE
  sbi    EECR,EEPE
  ret
```

C Code Example

```
void EEPROM_write(unsigned char ucAddress, unsigned char ucData)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEPE))
    ;
  /* Set Programming mode */
  EECR = (0<<EEPM1)|(0<<EEPM0);
  /* Set up address and data registers */
  EEAR = ucAddress;
  EEDR = ucData;
  /* Write logical one to EEMPE */
  EECR |= (1<<EEMPE);
  /* Start eeprom write by setting EEPE */
  EECR |= (1<<EEPE);
}
```

Figure 4-12. Low-frequency Crystal Oscillator Connections



12 - 22pF capacitors may be necessary if parasitic impedance (pads, wires & PCB) is very low.

When this oscillator is selected, start-up times are determined by the SUT fuses or by CSUT field as shown in Table 4-12.

Table 4-12. Start-up Times for the Low Frequency Crystal Oscillator Clock Selection

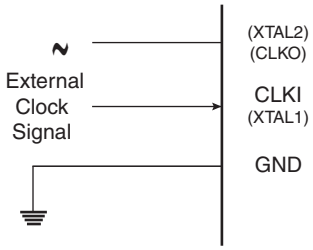
SUT1..0[1] CSUT1..0[2]	Start-up Time from Power-down/save	Additional Delay from Reset (Vcc = 5.0V)	Recommended Usage
00	1K (1024) CK[3]	4.1ms	Fast rising power or BOD enabled
01	1K (1024) CK[3]	65ms	Slowly rising power
10	32K (32768) CK	65ms	Stable frequency at start-up
11	Reserved		

Notes:
1. Flash Fuse bits
2. CLKSELR register bits
3. These options should only be used if frequency stability at start-up is not important for the application

4.5.2.6 *External Clock*

To drive the device from this external clock source, CLKI should be driven as shown in Figure 4-13. To run the device on an external clock, the CKSEL Fuses or CSEL field must be programmed as shown in Table 4-5 on page 49.

Figure 4-13. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses or CSUT field as shown in Table 4-13 This external clock can be used by the asynchronous timer if the high or low frequency Crystal Oscillator is not running (See ”Enable/Disable Clock Source” on page 57.). The asynchronous timer is then able to enable this input.

Notes: 1. ONLY the reset (watchdog reset included) disables this function. The Watchdog System Reset Flag (WDRF bit of MCUSR register) can be used to monitor the reset cause.

2. ONLY clock frequencies ≥ (4 * WatchDog Clock frequency) can be monitored.

Here is a “light” C-code of a clock switching function using automatic clock monitoring.

C Code Example

```
void ClockSwiching (unsigned char clk_number, unsigned char sut) {

#define CLOCK_RECOVER   0x05
#define CLOCK_ENABLE    0x02
#define CLOCK_SWITCH    0x04
#define CLOCK_DISABLE   0x01
#define WD_ARL_ENABLE   0x06

#define WD_2048CYCLES   0x07

unsigned char previous_clk, temp;

  // Disable interrupts
    temp = SREG; asm ("cli");
  // Save the current system clock source
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_RECOVER;
    previous_clk = CLKSELR & 0x0F;
  // Enable the new clock source
    CLKSELR = ((sut << 4 ) & 0x30) | (clk_number & 0x0F);
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_ENABLE;
  // Wait for clock validity
    while ((CLKCSR & (1 << CLKRDY)) == 0);

  // Enable the watchdog in automatic reload mode
    WDTCSR = (1 << WDCE) | (1 << WDE);
    WDTCSR = (1 << WDE ) | WD_2048CYCLES;
    CLKCSR = 1 << CLKCCE;
    CLKCSR = WD_ARL_ENABLE;

  // Switch clock source
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_SWITCH;
  // Wait for effective switching
    while (1){
        CLKCSR = 1 << CLKCCE;
        CLKCSR = CLOCK_RECOVER;
        if ((CLKSELR & 0x0F) == (clk_number & 0x0F)) break;
    }
  // Shut down unneeded clock source
    if (previous_clk != (clk_number & 0x0F)) {
    CLKSELR = previous_clk;
    CLKCSR = 1 << CLKCCE;
    CLKCSR = CLOCK_DISABLE;
}
  // Re-enable interrupts
    SREG = temp;
}
```

4.6.2 BOD Disable

When the Brown-out Detector (BOD) is enabled by BODLEVEL fuses, Table 4-69 on page 251, the BOD is actively monitoring the power supply voltage during a sleep period. To save power, it is possible to disable the BOD by software for some of the sleep modes, see Table 4-16. The sleep mode power consumption will then be at the same level as when BOD is globally disabled by fuses. If BOD is disabled in software, the BOD function is turned off immediately after entering the sleep mode. Upon wake-up from sleep, BOD is automatically enabled again. This ensures safe operation in case the Vcc level has dropped during the sleep period.

When the BOD has been disabled, the wake-up time from sleep mode will be approximately 60 µs to ensure that the BOD is working correctly before the MCU continues executing code.

BOD disable is controlled by BODS bit (BOD Sleep) in the control register MCUCR, see "MCUCR – MCU Control Register" on page 70. Setting it to one turns off the BOD in relevant sleep modes, while a zero in this bit keeps BOD active. Default setting keeps BOD active, i.e. BODS is cleared to zero.

Writing to the BODS bit is controlled by a timed sequence and an enable bit, see "MCUCR – MCU Control Register" on page 70.

4.6.3 Idle Mode

When the SM1..0 bits are written to 00, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing the SPI, Analog Comparator, ADC, USI start condition, Asynchronous Timer/Counter, Watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the SPI interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

4.6.4 ADC Noise Reduction Mode

When the SM1..0 bits are written to 01, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupts, the USI start condition, the asynchronous Timer/Counter and the Watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart from the ADC Conversion Complete interrupt, only an External Reset, a Watchdog System Reset, a Watchdog Interrupt, a Brown-out Reset, a USI start condition interrupt, an asynchronous Timer/Counter interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT0 or INT1 or a pin change interrupt can wake up the MCU from ADC Noise Reduction mode.

4.6.8 Minimizing Power Consumption

There are several possibilities to consider when trying to minimize the power consumption in an AVR® controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

4.6.8.1 Analog to Digital Converter

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to Section 4.18 "ADC – Analog to Digital Converter" on page 214 for details on ADC operation.

4.6.8.2 Analog Comparator

When entering Idle mode, the Analog Comparator should be disabled if not used. When entering ADC Noise Reduction mode, the Analog Comparator should be disabled. In other sleep modes, the Analog Comparator is automatically disabled. However, if the Analog Comparator is set up to use the Internal Voltage Reference as input, the Analog Comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to Section 4.19 "AnaComp - Analog Comparator" on page 236 for details on how to configure the Analog Comparator.

4.6.8.3 Brown-out Detector

If the Brown-out Detector is not needed by the application, this module should be turned off. If the Brown-out Detector is enabled by the BODLEVEL Fuses, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to Section 4.7.1.5 "Brown-out Detection" on page 74 for details on how to configure the Brown-out Detector.

4.6.8.4 Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detection, the Analog Comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and it will not be consuming power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to Section 4.7.2 "Internal Voltage Reference" on page 76 for details on the start-up time.

Output the internal voltage reference is not needed in the deeper sleep modes. This module should be turned off to reduce significantly to the total current consumption. Refer to Section 4.17.3.1 "AMISCR – Analog Miscellaneous Control Register" on page 213 for details on how to disable the internal voltage reference output.

4.6.8.5 Internal Current Source

The Internal Current Source is not needed in the deeper sleep modes. This module should be turned off to reduce significantly to the total current consumption. Refer to Section 4.17.3.1 "AMISCR – Analog Miscellaneous Control Register" on page 213 for details on how to disable the Internal Current Source.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

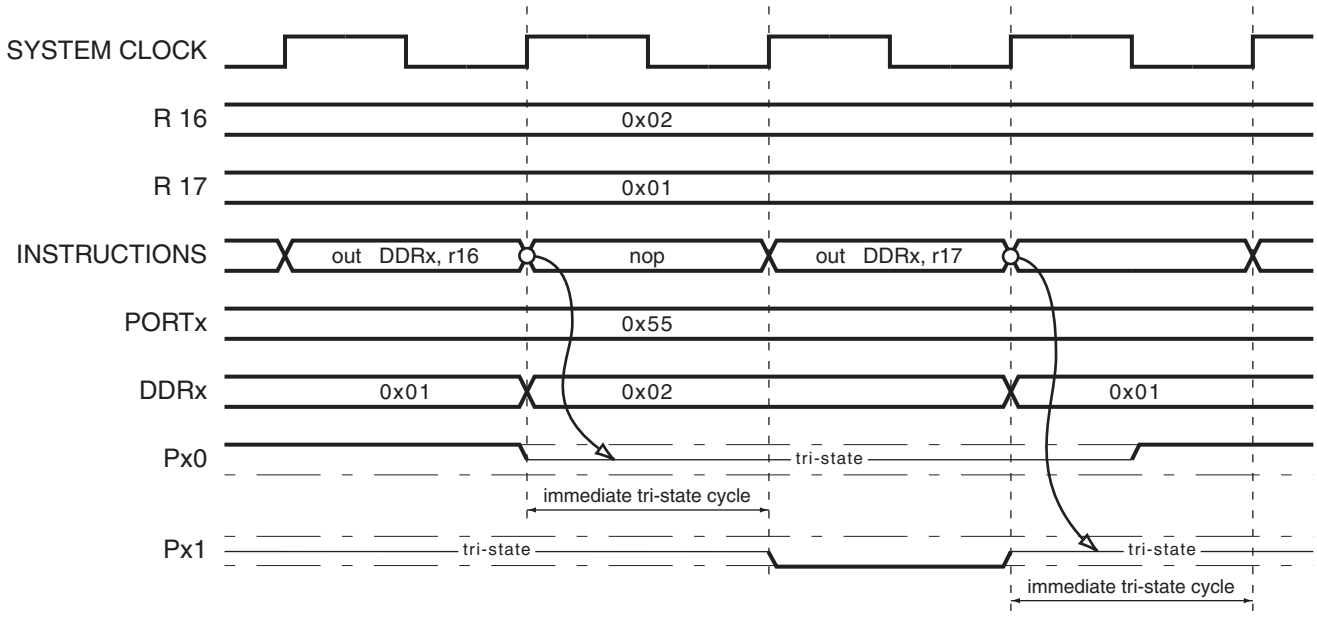
4.10.2.2 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI assembler instruction can be used to toggle one single bit in a port.

4.10.2.3 Break-Before-Make Switching

In the Break-Before-Make mode when switching the DDRxn bit from input to output an immediate tri-state period lasting one system clock cycle is introduced as indicated in Figure 4-26. For example, if the system clock is 4MHz and the DDRxn is written to make an output, the immediate tri-state period of 250ns is introduced, before the value of PORTxn is seen on the port pin. To avoid glitches it is recommended that the maximum DDRxn toggle frequency is two system clock cycles. The Break-Before-Make is a port-wise mode and it is activated by the port-wise BBMx enable bits. For further information about the BBMx bits, see “Port Control Register – PORTCR” on page 99. When switching the DDRxn bit from output to input there is no immediate tri-state period introduced.

Figure 4-26. Break Before Make, Switching between Input and Output



4.11.11.2 Timer/Counter0 Control Register B – TCCR0B

Bit	7	6	5	4	3	2	1	0	
	FOC0A	–	–	–	–	CS02	CS01	CS00	TCCR0B
Read/Write	W	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A: Force Output Compare A**

The FOC0A bit is only active when the WGM bits specify a non-PWM mode.

However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written when operating in PWM mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare.

A FOC0A strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0A as TOP.

The FOC0A bit is always read as zero.

- **Bit 6:3 – Res: Reserved Bits**

These bits are reserved in the Atmel® ATtiny87/167 and will always read as zero.

- **Bit 2:0 – CS02:0: Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Table 4-34.

Table 4-34. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk_{T0S} (No prescaling)
0	1	0	clk_{T0S}/8 (From prescaler)
0	1	1	clk_{T0S}/32 (From prescaler)
1	0	0	clk_{T0S}/64 (From prescaler)
1	0	1	clk_{T0S}/128 (From prescaler)
1	1	0	clk_{T0S}/256 (From prescaler)
1	1	1	clk_{T0S}/1024 (From prescaler)

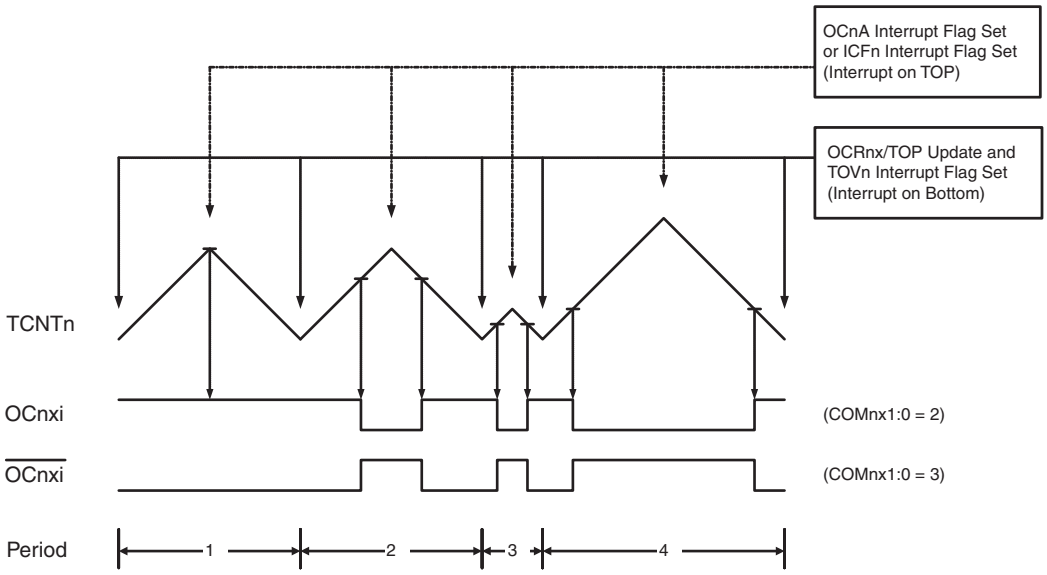
4.11.11.3 Timer/Counter0 Register – TCNT0

Bit	7	6	5	4	3	2	1	0	
	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0x Register.

In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM13:0 = 8), or the value in OCR1A (WGM13:0 = 9). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct and frequency correct PWM mode is shown on Figure 4-53. The figure shows phase and frequency correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1A/B and TCNT1. The OC1A/B interrupt flag will be set when a compare match occurs.

Figure 4-53. Phase and Frequency Correct PWM Mode, Timing Diagram



The Timer/Counter Overflow Flag (TOV1) is set at the same timer clock cycle as the OCR1A/B Registers are updated with the double buffer value (at BOTTOM). When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 flag set when TCNT1 has reached TOP. The interrupt flags can then be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the TCNT1 and the OCR1A/B.

As Figure 4-53 shows the output generated is, in contrast to the phase correct mode, symmetrical in all periods. Since the OCR1A/B Registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed by changing the TOP value, using the OCR1A as TOP is clearly a better choice due to its double buffer feature.

In phase and frequency correct PWM mode, the compare units allow generation of PWM waveforms on the OC1A/B pins. Setting the COM1A/B1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1A/B1:0 to three (See Table on page 158). The actual OC1A/B value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1A/B) and OC1A/Bi is set. The PWM waveform is generated by setting (or clearing) the OC1A/B Register at the compare match between OCR1A/B and TCNT1 when the counter increments, and clearing (or setting) the OC1A/B Register at compare match between OCR1A/B and TCNT1 when the counter decrements. The PWM frequency for the output when using phase and frequency correct PWM can be calculated by the following equation:

$$f_{OCnxPFCPWM} = \frac{f_{clk\_I/O}}{2 \times N \times TOP}$$

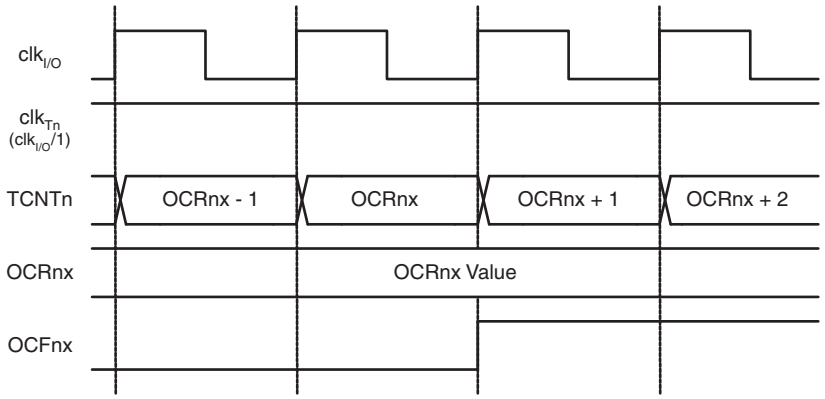
The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1A/B Register represents special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1A/B is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be set to high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

4.13.10 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T1}) is therefore shown as a clock enable signal in the following figures. The figures include information on when interrupt flags are set, and when the OCR1A/B Register is updated with the OCR1A/B buffer value (only for modes utilizing double buffering). Figure 4-54 shows a timing diagram for the setting of OCF1A/B.

Figure 4-54. Timer/Counter Timing Diagram, Setting of OCF1A/B, No Prescaling



4.15.3.5 *Start Condition Detector*

The start condition detector is shown in Figure 4-67. The SDA line is delayed (in the range of 50 to 300 ns) to ensure valid sampling of the SCL line. The start condition detector is only enabled in Two-wire mode.

The start condition detector is working asynchronously and can therefore wake up the processor from the Power-down sleep mode. However, the protocol used might have restrictions on the SCL hold time. Therefore, when using this feature in this case the Oscillator start-up time set by the CKSEL Fuses (see Section 4.5.1 “Clock Systems and their Distribution” on page 48) must also be taken into the consideration. Refer to the USISIF bit description on page 182 for further details.

4.15.4 Alternative USI Usage

When the USI unit is not used for serial communication, it can be set up to do alternative tasks due to its flexible design.

4.15.4.1 *Half-duplex Asynchronous Data Transfer*

By utilizing the USI Data Register in Three-wire mode, it is possible to implement a more compact and higher performance UART than by software only.

4.15.4.2 *4-bit Counter*

The 4-bit counter can be used as a stand-alone counter with overflow interrupt. Note that if the counter is clocked externally, both clock edges will generate an increment.

4.15.4.3 *12-bit Timer/Counter*

Combining the USI 4-bit counter and Timer/Counter0 allows them to be used as a 12-bit counter.

4.15.4.4 *Edge Triggered External Interrupt*

By setting the counter to maximum value (F) it can function as an additional external interrupt. The Overflow Flag and Interrupt Enable bit are then used for the external interrupt. This feature is selected by the USICS1 bit.

4.15.4.5 *Software Interrupt*

The counter overflow interrupt can be used as a software interrupt triggered by a clock strobe.

Table 4-45. Relations between USIWM1..0 and the USI Operation

USIWM1	USIWM0	Description
0	0	Outputs, clock hold, and start detector disabled. Port pins operates as normal.
0	1	Three-wire mode. Uses DO, DI, and USCK pins. The *Data Output* (DO) pin overrides the corresponding bit in the PORT Register in this mode. However, the corresponding DDR bit still controls the data direction. When the port pin is set as input the pins pull-up is controlled by the PORT bit. The *Data Input* (DI) and *Serial Clock* (USCK) pins do not affect the normal port operation. When operating as master, clock pulses are software generated by toggling the PORT Register, while the data direction is set to output. The USITC bit in the USICR Register can be used for this purpose.
1	0	Two-wire mode. Uses SDA (DI) and SCL (USCK) pins(1). The *Serial Data* (SDA) and the *Serial Clock* (SCL) pins are bi-directional and uses open-collector output drives. The output drivers are enabled by setting the corresponding bit for SDA and SCL in the DDR Register. When the output driver is enabled for the SDA pin, the output driver will force the line SDA low if the output of the USI Data Register or the corresponding bit in the PORT Register is zero. Otherwise the SDA line will not be driven (i.e., it is released). When the SCL pin output driver is enabled the SCL line will be forced low if the corresponding bit in the PORT Register is zero, or by the start detector. Otherwise the SCL line will not be driven. The SCL line is held low when a start detector detects a start condition and the output is enabled. Clearing the Start Condition Flag (USISIF) releases the line. The SDA and SCL pin inputs is not affected by enabling this mode. Pull-ups on the SDA and SCL port pin are disabled in Two-wire mode.
1	1	Two-wire mode. Uses SDA and SCL pins. Same operation as for the Two-wire mode described above, except that the SCL line is also held low when a counter overflow occurs, and is held low until the Counter Overflow Flag (USIOIF) is cleared.

Note: 1. The DI and USCK pins are renamed to *Serial Data* (SDA) and *Serial Clock* (SCL) respectively to avoid confusion between the modes of operation.

- **Bit 3:2 – USICS1:0: Clock Source Select**

These bits set the clock source for the USI Data Register and counter. The data output latch ensures that the output is changed at the opposite edge of the sampling of the data input (DI/SDA) when using external clock source (USCK/SCL). When software strobe or Timer/Counter0 Compare Match clock option is selected, the output latch is transparent and therefore the output is changed immediately. Clearing the USICS1:0 bits enables software strobe option. When using this option, writing a one to the USICLK bit clocks both the USI Data Register and the counter. For external clock source (USICS1 = 1), the USICLK bit is no longer used as a strobe, but selects between external clocking and software clocking by the USITC strobe bit.

Table 4-46 on page 185 shows the relationship between the USICS1..0 and USICLK setting and clock source used for the USI Data Register and the 4-bit counter.

4.16.5.4 *Configuration*

Depending on the mode (LIN or UART), LCONF[1..0] bits of the LINCR register set the controller in the following configuration (Table 4-50):

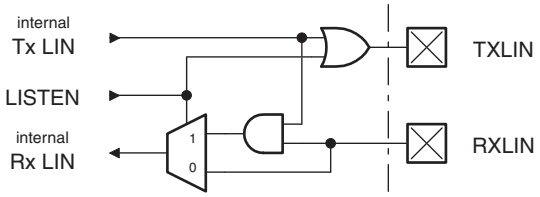
Table 4-50. Configuration Table versus Mode

Mode	LCONF[1..0]	Configuration
LIN	00_b	LIN standard configuration (default)
	01_b	No CRC field detection or transmission
	10_b	Frame_Time_Out disable
	11_b	Listening mode
UART	00_b	8-bit data, no parity & 1 stop-bit
	01_b	8-bit data, even parity & 1 stop-bit
	10_b	8-bit data, odd parity & 1 stop-bit
	11_b	Listening mode, 8-bit data, no parity & 1 stop-bit

The LIN configuration is independent of the programmed LIN protocol.

The listening mode connects the internal Tx LIN and the internal Rx LIN together. In this mode, the TXLIN output pin is disabled and the RXLIN input pin is always enabled. The same scheme is available in UART mode.

Figure 4-73. Listening Mode

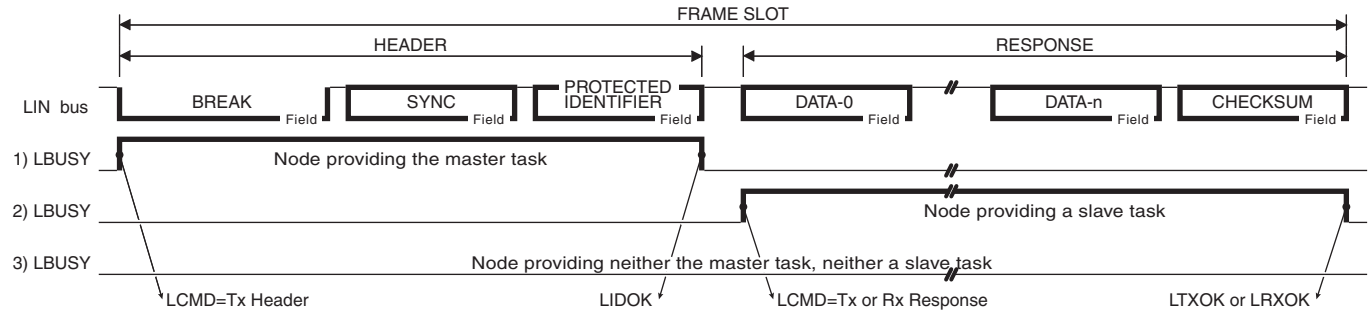


4.16.5.5 *Busy Signal*

LBUSY bit flag in LINSIR register is the image of the BUSY signal. It is set and cleared by hardware. It signals that the controller is busy with LIN or UART communication.

Busy Signal in LIN Mode

Figure 4-74. Busy Signal in LIN Mode



4.16.6.7 LIN Data Length Register - LINDLR

Bit	7	6	5	4	3	2	1	0	
	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	LINDLR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:4 - LTXDL[3:0]: LIN Transmit Data Length**

  In LIN mode, this field gives the number of bytes to be transmitted (clamped to 8 Max).

  In UART mode this field is unused.

- **Bits 3:0 - LRXDL[3:0]: LIN Receive Data Length**

  In LIN mode, this field gives the number of bytes to be received (clamped to 8 Max).

  In UART mode this field is unused.

4.16.6.8 LIN Identifier Register - LINIDR

Bit	7	6	5	4	3	2	1	0	
	LP1	LP0	LID5 / LDL1	LID4 / LDL0	LID3	LID2	LID1	LID0	LINIDR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 - LP[1:0]: Parity**

  In LIN mode:

```
LP0 = LID4 ^ LID2 ^ LID1 ^ LID0
LP1 = ! ( LID1 ^ LID3 ^ LID4 ^ LID5 )
```

  In UART mode this field is unused.

- **Bits 5:4 - LDL[1:0]: LIN 1.3 Data Length**

  In LIN 1.3 mode:

  – 00 = 2-byte response,
  – 01 = 2-byte response,
  – 10 = 4-byte response,
  – 11 = 8-byte response.

  In UART mode this field is unused.

- **Bits 3:0 - LID[3:0]: LIN 1.3 Identifier**

  In LIN 1.3 mode: 4-bit identifier.

  In UART mode this field is unused.

- **Bits 5:0 - LID[5:0]: LIN 2.1 Identifier**

  In LIN 2.1 mode: 6-bit identifier (no length transported).

  In UART mode this field is unused.

4.23.2 DC Characteristics (Continued)

T_A = –40°C to +125°C, Vcc = 2.7V to 5.5V (unless otherwise noted)

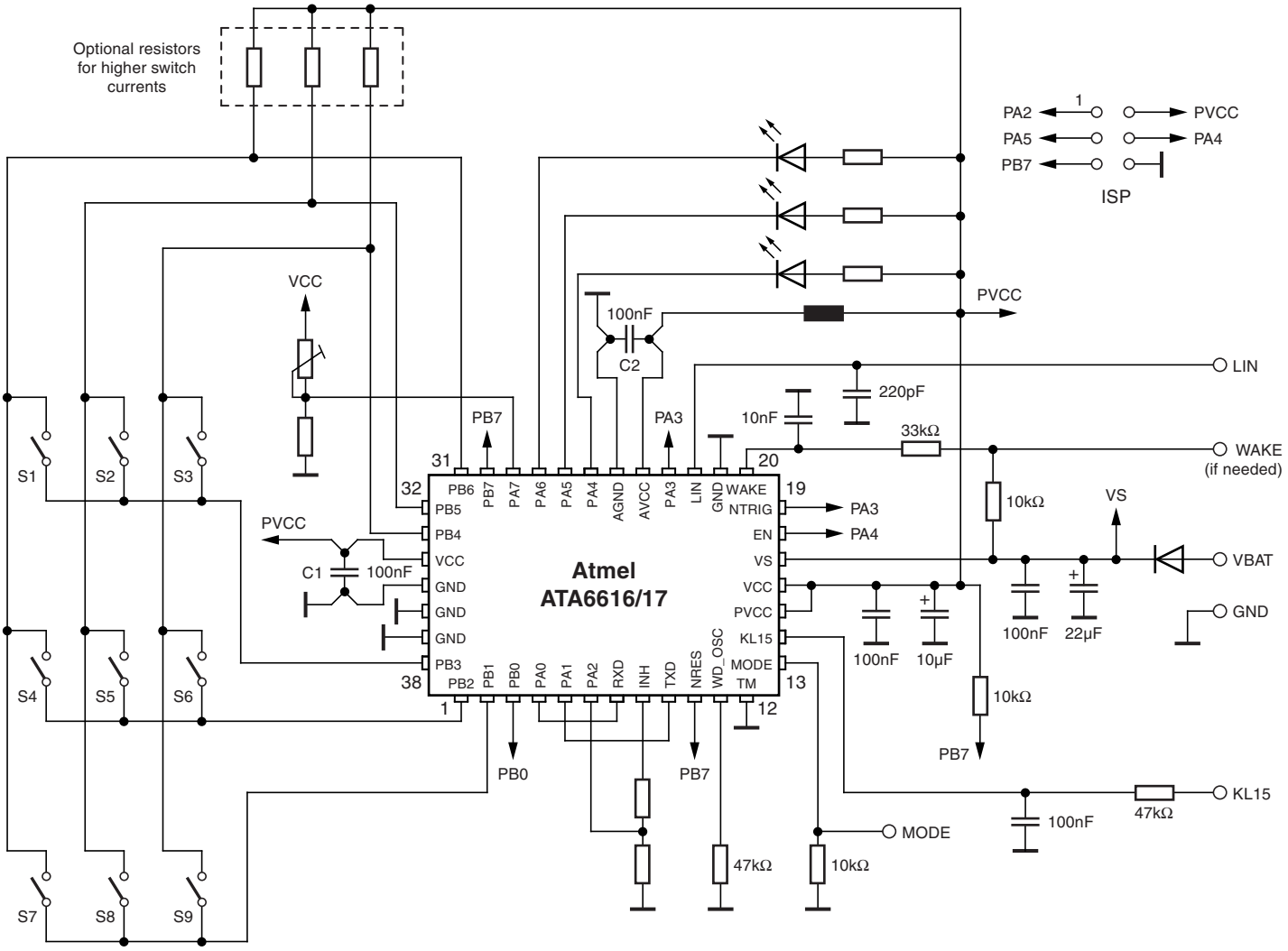
Parameters	Test Conditions	Symbol	Min.	Typ.(1)	Max.	Unit
Power Supply Current(6) Active Mode (external clock)	16MHz, Vcc = 5V	I_{CC}		10	13	mA
	8MHz, Vcc = 5V			5.5	7.0	mA
	8MHz, Vcc = 3V			2.8	3.5	mA
	4MHz, Vcc = 3V			1.8	2.5	mA
Power Supply Current(6) Idle Mode (external clock)	16MHz, Vcc = 5V			3.5	5.0	mA
	8MHz, Vcc = 5V			1.8	2.5	mA
	8MHz, Vcc = 3V			1	1.5	mA
	4MHz, Vcc = 3V			0.5	0.8	mA
Power Supply Current(7) Power-down Mode	WDT enabled, Vcc = 5V			7	100	µA
	WDT disabled, Vcc = 5V			0.18	70	µA
	WDT enabled, Vcc = 3V			5	70	µA
	WDT disabled, Vcc = 3V			0.15	45	µA
Analog Comparator Input Offset Voltage	Vcc = 5V V_{in} = Vcc/2	V_{ACIO}	–10	+10	+40	mV
Analog Comparator Input Leakage Current	Vcc = 5V V_{in} = Vcc/2	I_{ACLK}	–50		+50	nA
Analog Comparator Propagation Delay Common Mode Vcc/2	Vcc = 2.7V	t_{ACID}		170		ns
	Vcc = 5.0V			180		ns

Notes:
1. “Typ.”, typical values at 25°C. Maximum values are characterized values and not test limits in production.
2. “Max.” means the highest value where the pin is guaranteed to be read as low.
3. “Min.” means the lowest value where the pin is guaranteed to be read as high.
4. Although each I/O port can sink more than the test conditions (10mA at Vcc = 5V, 5mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   The sum of all IOL, for all ports, should not exceed 120mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
5. Although each I/O port can source more than the test conditions (10mA at Vcc = 5V, 5mA at Vcc = 3V) under steady state conditions (non-transient), the following must be observed:
   The sum of all IOH, for all ports, should not exceed 120mA.
   If IOH exceeds the test condition, VOH may exceed the related specification. Pins are not guaranteed to source current greater than the listed test condition.
6. Values using methods described in Section 4.6.8 “Minimizing Power Consumption” on page 68. Power Reduction is enabled (PRR = 0xFF) and there is no I/O drive.
7. BOD Disabled.

4.27 Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ←Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ←Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ←K	None	1
LD	Rd, X	Load Indirect	Rd ←(X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ←(X), X ←X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ←X - 1, Rd ←(X)	None	2
LD	Rd, Y	Load Indirect	Rd ←(Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ←(Y), Y ←Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ←Y - 1, Rd ←(Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ←(Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ←(Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ←(Z), Z ←Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ←Z - 1, Rd ←(Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ←(Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ←(k)	None	2
ST	X, Rr	Store Indirect	(X) ←Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ←Rr, X ←X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ←X - 1, (X) ←Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ←Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ←Rr, Y ←Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ←Y - 1, (Y) ←Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ←Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ←Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ←Rr, Z ←Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ←Z - 1, (Z) ←Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ←Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ←Rr	None	2
LPM		Load Program Memory	R0 ←(Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ←(Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ←(Z), Z ←Z+1	None	3
SPM		Store Program Memory	(Z) ←R1:R0	None	-
IN	Rd, P	In Port	Rd ←P	None	1
OUT	P, Rr	Out Port	P ←Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ←Rr	None	2
POP	Rd	Pop Register from Stack	Rd ←STACK	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

Figure 5-5. 3 × 3 LIN Slave Switch Module Matrix Decoder with the Atmel® ATA6616/ATA6617



Note: Note: All open pins of the SiP can be used for application-specific purposes.
AVR®: TXD, RXD, NRES and EN connected for LIN Master. The software must be appropriately programmed for the connection between the LIN-SBC and the AVR. PA7 and PA2 are used as analog inputs. In addition, the application supports a 3 x 3 switch matrix. PA4, 5, 6 are used as PWM outputs to control the brightness of light emitting diodes. System clock controlled by external crystal.
LIN-SBC: LIN slave application, 1k Master resistance connected via diode to VBAT, local wake up via pin WAKE; watchdog is enabled.
RF emissions: best results for RF emissions will be achieved by connecting the blocking capacitors of the microcontroller supply (C1 and C2) between the microcontroller pins and the GND/PVCC line. See also Figure 5-5.