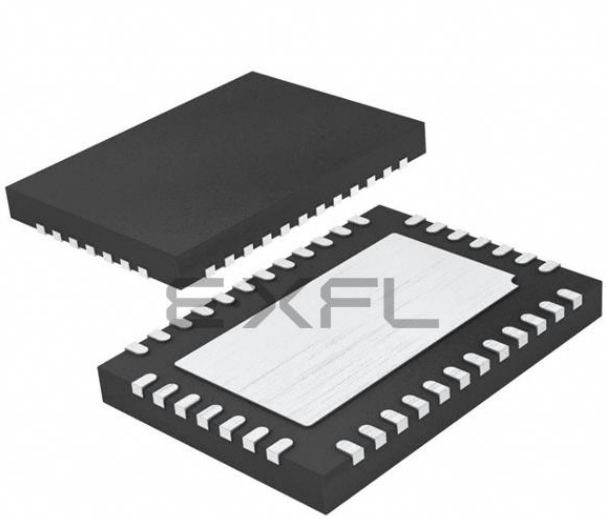


Welcome to E-XFL.COM

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	16KB (16K 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/ata6617-p3pw

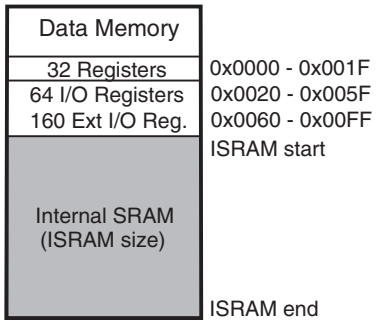
3.8 Electrical Characteristics (Continued)

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*
10.4	TXD dominant time-out time	V_{TXD} = 0V	TXD	t_{dom}	6	13	20	ms	A
10.5	Time delay for mode change from Silent Mode into Normal Mode via EN	V_{EN} = 5V	EN	t_{s_n}	5	15	40	µs	A
10.6	Duty cycle 1	$TH_{Rec(max)} = 0.744 \times V_S$ $TH_{Dom(max)} = 0.581 \times V_S$ V_S = 7.0V to 18V t_{Bit} = 50µs D1 = $t_{bus\_rec(min)}/(2 \times t_{Bit})$	LIN	D1	0.396				A
10.7	Duty cycle 2	$TH_{Rec(min)} = 0.422 \times V_S$ $TH_{Dom(min)} = 0.284 \times V_S$ V_S = 7.6V to 18V t_{Bit} = 50µs D2 = $t_{bus\_rec(max)}/(2 \times t_{Bit})$	LIN	D2			0.581		A
10.8	Duty cycle 3	$TH_{Rec(max)} = 0.778 \times V_S$ $TH_{Dom(max)} = 0.616 \times V_S$ V_S = 7.0V to 18V t_{Bit} = 96µs D3 = $t_{bus\_rec(min)}/(2 \times t_{Bit})$	LIN	D3	0.417				A
10.9	Duty cycle 4	$TH_{Rec(min)} = 0.389 \times V_S$ $TH_{Dom(min)} = 0.251 \times V_S$ V_S = 7.6V to 18V t_{Bit} = 96µs D4 = $t_{bus\_rec(max)}/(2 \times t_{Bit})$	LIN	D4			0.590		A
10.10	Slope time falling and rising edge at LIN	V_S = 7.0V to 18V	LIN	t_{SLOPE_fall} t_{SLOPE_rise}	3.5		22.5	µs	A
11	**Receiver Electrical AC Parameters of the LIN Physical Layer LIN Receiver, RXD Load Conditions C_{RXD} = 20pF**								
11.1	Propagation delay of receiver (Figure 3-10 on page 26)	V_S = 7.0V to 18V t_{rx_pd} = max(t_{rx_pdr} , t_{rx_pdf})	RXD	t_{rx_pd}			6	µs	A
11.2	Symmetry of receiver propagation delay rising edge minus falling edge	V_S = 7.0V to 18V $t_{rx\_sym} = t_{rx\_pdr} - t_{rx\_pdf}$	RXD	t_{rx_sym}	–2		+2	µs	A
12	**NRES Open Drain Output Pin**								
12.1	Low-level output voltage	$V_S \geq 5.5V$ I_{NRES} = 1mA I_{NRES} = 250µA	NRES	V_{NRESL}			0.2 0.14	V V	A
12.2	Low-level output low	10kΩ to V_{CC} V_{CC} = 0V	NRES	V_{NRESLL}			0.2	V	A
12.3	Undervoltage reset time	$V_S \geq 5.5V$ C_{NRES} = 20pF	NRES	t_{reset}	2	4	6	ms	A
12.4	Reset debounce time for falling edge	$V_S \geq 5.5V$ C_{NRES} = 20pF	NRES	t_{res_f}	1.5		10	µs	A

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

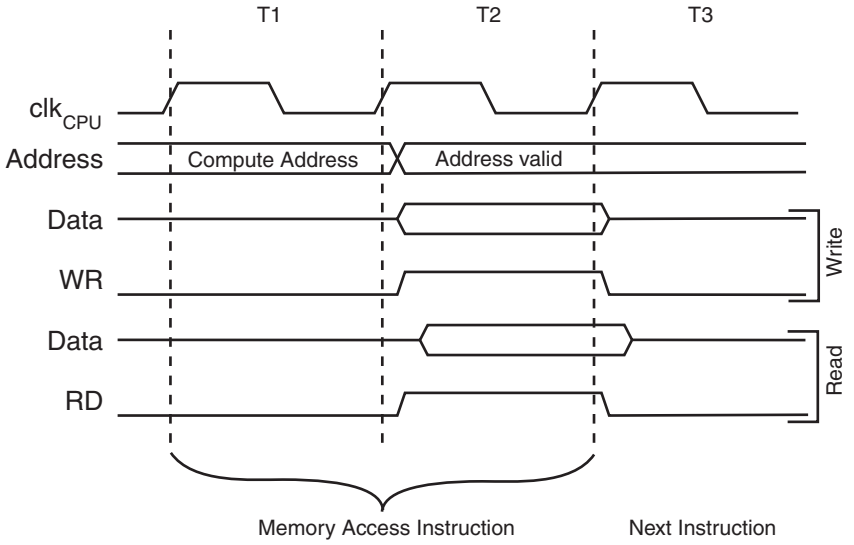
Figure 4-8. Data Memory Map



4.4.2.1 *Data Memory Access Times*

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 4-9.

Figure 4-9. On-chip Data SRAM Access Cycles



4.4.3 EEPROM Data Memory

The Atmel® ATtiny87/167 contains EEPROM memory (see “E2 size” in Table 4-3 on page 39). It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles in automotive range. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register and the EEPROM Control Register.

Section 4.22 “Memory Programming” on page 250 contains a detailed description on EEPROM programming in SPI or Parallel Programming mode.

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);
}
```

When the CPU wakes up from Power-down or Power-save, or when a new clock source is enabled by the dynamic clock switch circuit, the selected clock source is used to time the start-up, ensuring stable oscillator operation before instruction execution starts.
When the CPU starts from reset, there is an additional delay allowing the power to reach a stable level before commencing normal operation. The Watchdog Oscillator is used for timing this real-time part of the start-up sequence. The number of WDT Oscillator cycles used for each time-out is shown in Table 4-6.

Table 4-6. Number of Watchdog Oscillator Cycles

Typ. Time-out (Vcc = 5.0V)	Typ. Time-out (Vcc = 5.0V)	Number of Cycles
4.1ms	4.3ms	512
65ms	69ms	8K (8,192)

4.5.2.1 *Default Clock Source*

At reset, the CKSEL and SUT fuse settings are copied into the CLKSELR register. The device will then use the clock source and the start-up timings defined by the CLKSELR bits (CSEL3..0 and CSUT1:0).

The device is shipped with CKSEL Fuses = 0010_b, SUT Fuses = 10_b, and CKDIV8 Fuse programmed. The default clock source setting is therefore the Internal RC Oscillator running at 8MHz with the longest start-up time and an initial system clock divided by 8. This default setting ensures that all users can make their desired clock source setting using an In-System or High-voltage Programmer. This set-up must be taken into account when using ISP tools.

4.5.2.2 *Calibrated Internal RC Oscillator*

By default, the Internal RC Oscillator provides an approximate 8.0MHz clock. The frequency is nominal at 5V and 25°C. Though voltage and temperature dependent, this clock can be accurately calibrated by the user. See Table 4-81 on page 269 and Section 4.25.7 “Internal Oscillator Speed” on page 288 for more details.

If selected, it can operate without external components. At reset, hardware loads the pre-programmed calibration value into the OSCCAL Register and thereby automatically configuring the RC Oscillator. At 5V and 25° C, this calibration gives a frequency of 8MHz ±1%. The tolerance of the internal RC oscillator remains better than ±10% within the whole automotive temperature and voltage ranges (4.5V to 5.5V, –40°C to +125°C). The accuracy of this calibration is shown as Factory calibration in Table 4-81 on page 269.

By adjusting the OSCCAL register in software, see Section 4.5.5.1 “OSCCAL – Oscillator Calibration Register” on page 61, it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in Table 4-81 on page 269.

The Watchdog Oscillator will still be used for the Watchdog Timer and for the Reset Time-out even when this Oscillator is used as the device clock. For more information on the pre-programmed calibration value, see the section Section 4.22.4 “Calibration Byte” on page 252.

4.6.8.6 *Watchdog Timer*

If the Watchdog Timer is not needed in the application, the module should be turned off. If the Watchdog Timer is enabled, 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 4.7.3 “Watchdog Timer” on page 76 for details on how to configure the Watchdog Timer.

4.6.8.7 *Port Pins*

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section Section 4.10.2.6 “Digital Input Enable and Sleep Modes” on page 96 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to Vcc/2, the input buffer will use excessive power.

For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to Vcc/2 on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Registers (DIDR1 and DIDR0). Refer to Section 4.18.12.6 “DIDR1 – Digital Input Disable Register 1” on page 234 and Section 4.18.12.5 “DIDR0 – Digital Input Disable Register 0” on page 234 for details.

4.6.8.8 *On-chip Debug System*

If the On-chip debug system is enabled by the DWEN Fuse and the chip enters sleep mode, the main clock source is enabled and hence always consumes power. In the deeper sleep modes, this will contribute significantly to the total current consumption.

4.6.9 Register Description

4.6.9.1 *SMCR – Sleep Mode Control Register*

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	**SM1**	**SM0**	**SE**	**SMCR**
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 Res: Reserved Bits**

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

- **Bits 2..1 – SM1..0: Sleep Mode Select Bits 1, and 0**

These bits select between the four available sleep modes as shown in Table 4-17.

Table 4-17. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle
0	1	ADC Noise Reduction
1	0	Power-down
1	1	Power-save

- **Bit 0 – SE: Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

4.6.9.2 *MCUCR – MCU Control Register*

Bit	7	6	5	4	3	2	1	0	
	–	BODS	BODSE	PUD	–	–	–	–	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – BODS: BOD Sleep**

The BODS bit must be written to logic one in order to turn off BOD during sleep, see Table 4-16 on page 65. Writing to the BODS bit is controlled by a timed sequence and an enable bit, BODSE in MCUCR. To disable BOD in relevant sleep modes, both BODS and BODSE must first be set to one. Then, to set the BODS bit, BODS must be set to one and BODSE must be set to zero within four clock cycles.

The BODS bit is active three clock cycles after it is set. A sleep instruction must be executed while BODS is active in order to turn off the BOD for the actual sleep mode. The BODS bit is automatically cleared after three clock cycles.

- **Bit 5 – BODSE: BOD Sleep Enable**

BODSE enables setting of BODS control bit, as explained in BODS bit description. BOD disable is controlled by a timed sequence.

4.6.9.3 *PRR – Power Reduction Register*

Bit	7	6	5	4	3	2	1	0	
	–	–	PRLIN	PRSPI	PRTIM1	PRTIM0	PRUSI	PRADC	PRR
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	

- **Bit 7 - Res: Reserved bit**

This bit is reserved in Atmel® ATtiny87/167 and will always read as zero.

- **Bit 6 - Res: Reserved bit**

This bit is reserved in Atmel ATtiny87/167 and will always read as zero.

4.10.3.1 MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
	–	BODS	BODSE	PUD	–	–	–	–	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 4 – PUD: Pull-up Disable**

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0, 1). See “Configuring the Pin” on page 91 for more details about this feature.

4.10.3.2 Port Control Register – PORTCR

Bit	7	6	5	4	3	2	1	0	
	-	-	BBMB	BBMA	-	-	PUDB	PUDA	PORTCR
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 5, 4 – BBMx: Break-Before-Make Mode Enable**

When these bits are written to one, the port-wise Break-Before-Make mode is activated. The intermediate tri-state cycle is then inserted when writing DDRxn to make an output. For further information, see “Break-Before-Make Switching” on page 92.

- **Bits 1, 0 – PUDx: Port-Wise Pull-up Disable**

When these bits are written to one, the port-wise pull-ups in the defined I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0, 1). The Port-Wise Pull-up Disable bits are ORed with the global Pull-up Disable bit (PUD) from the MCUCR register. See “Configuring the Pin” on page 91 for more details about this feature.

Figure 4-55 shows the same timing data, but with the prescaler enabled.

Figure 4-55. Timer/Counter Timing Diagram, Setting of OCF1A/B, with Prescaler ($f_{clk\_I/O}/8$)

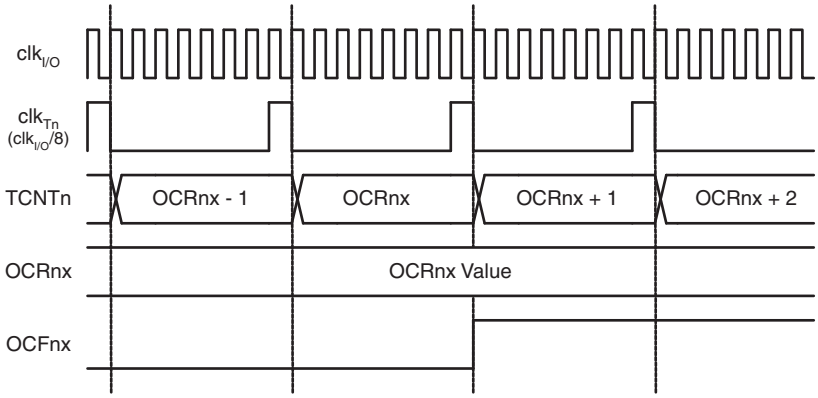


Figure 4-56 shows the count sequence close to TOP in various modes. When using phase and frequency correct PWM mode the OCR1A/B Register is updated at BOTTOM. The timing diagrams will be the same, but TOP should be replaced by BOTTOM, TOP-1 by BOTTOM+1 and so on. The same renaming applies for modes that set the TOV1 flag at BOTTOM.

Figure 4-56. Timer/Counter Timing Diagram, no Prescaling

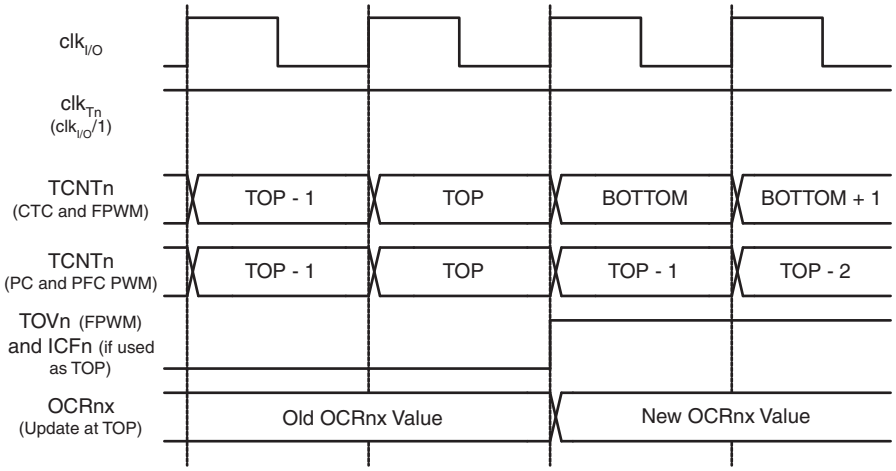


Table 4-46. Relations between the USICS1..0 and USICLK Setting

USICS1	USICS0	USICLK	USI Data Register Clock Source	4-bit Counter Clock Source
0	0	0	No Clock	No Clock
0	0	1	Software clock strobe (USICLK)	Software clock strobe (USICLK)
0	1	X	Timer/Counter0 Compare Match	Timer/Counter0 Compare Match
1	0	0	External, positive edge	External, both edges
1	1	0	External, negative edge	External, both edges
1	0	1	External, positive edge	Software clock strobe (USITC)
1	1	1	External, negative edge	Software clock strobe (USITC)

- **Bit 1 – USICLK: Clock Strobe**

Writing a one to this bit location strobes the USI Data Register to shift one step and the counter to increment by one, provided that the USICS1..0 bits are set to zero and by doing so the software clock strobe option is selected. The output will change immediately when the clock strobe is executed, i.e., in the same instruction cycle. The value shifted into the USI Data Register is sampled the previous instruction cycle. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1), the USICLK function is changed from a clock strobe to a Clock Select Register. Setting the USICLK bit in this case will select the USITC strobe bit as clock source for the 4-bit counter (see Table 4-46).

- **Bit 0 – USITC: Toggle Clock Port Pin**

Writing a one to this bit location toggles the USCK/SCL value either from 0 to 1, or from 1 to 0. The toggling is independent of the setting in the Data Direction Register, but if the PORT value is to be shown on the pin the DDB2 must be set as output (to one). This feature allows easy clock generation when implementing master devices. The bit will be read as zero.

When an external clock source is selected (USICS1 = 1) and the USICLK bit is set to one, writing to the USITC strobe bit will directly clock the 4-bit counter. This allows an early detection of when the transfer is done when operating as a master device.

4.15.5.5 USIPP – USI Pin Position

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	-	-	**USIPOS**	**USIPP**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:1 – Res: Reserved Bits**

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

- **Bit 0 – USIPOS: USI Pin Position**

Setting or clearing this bit changes the USI pin position.

When the busy signal is set, some registers are locked, user writing is not allowed:

- “LIN Control Register” - LINCR - except LCMD[2..0], LENA & LSWRES,
- “LIN Baud Rate Registers” - LINBRRL & LINBRRH,
- “LIN Data Length Register” - LINDLR,
- “LIN Identifier Register” - LINIDR,
- “LIN Data Register” - LINDAT.

If the busy signal is set, the only available commands are:

- LCMD[1..0] = 00_b, the abort command is taken into account at the end of the byte,
- LENA = 0 and/or LCMD[2] = 0, the kill command is taken into account immediately,
- LSWRES = 1, the reset command is taken into account immediately.

Note that, if another command is entered during busy signal, the new command is not validated and the LOVRERR bit flag of the LINERR register is set. The on-going transfer is not interrupted.

Busy Signal in UART Mode

During the byte transmission, the busy signal is set. This locks some registers from being written:

- “LIN Control Register” - LINCR - except LCMD[2..0], LENA & LSWRES,
- “LIN Data Register” - LINDAT.

The busy signal is not generated during a byte reception.

4.16.5.6 Bit Timing

Baud rate Generator

The baud rate is defined to be the transfer rate in bits per second (bps):

- BAUD: Baud rate (in bps),
- $fclk_{i/o}$: System I/O clock frequency,
- LDIV[11..0]: Contents of LINBRRH & LINBRRL registers - (0-4095), the pre-scaler receives $clk_{i/o}$ as input clock.
- LBT[5..0]: Least significant bits of - LINBTR register- (0-63) is the number of samplings in a LIN or UART bit (default value 32).

Equation for calculating baud rate:

$$BAUD = fclk_{i/o} / LBT[5..0] \times (LDIV[11..0] + 1)$$

Equation for setting LINDIV value:

$$LDIV[11..0] = (fclk_{i/o} / LBT[5..0] \times BAUD) - 1$$

Note that in reception a majority vote on three samplings is made.

Re-synchronization in LIN Mode

When waiting for Rx Header, LBT[5..0] = 32 in LINBTR register. The re-synchronization begins when the BREAK is detected. If the BREAK size is not in the range (10.5 bits min., 28 bits max. — 13 bits nominal), the BREAK is refused. The re-synchronization is done by adjusting LBT[5..0] value to the SYNCH field of the received header (0x55). Then the PROTECTED IDENTIFIER is sampled using the new value of LBT[5..0]. The re-synchronization implemented in the controller tolerates a clock deviation of ± 20% and adjusts the baud rate in a ±2% range.

The new LBT[5..0] value will be used up to the end of the response. Then, the LBT[5..0] will be reset to 32 for the next header.

The LINBTR register can be used to (software) re-calibrate the clock oscillator.

The re-synchronization is not performed if the LIN node is enabled as a master.

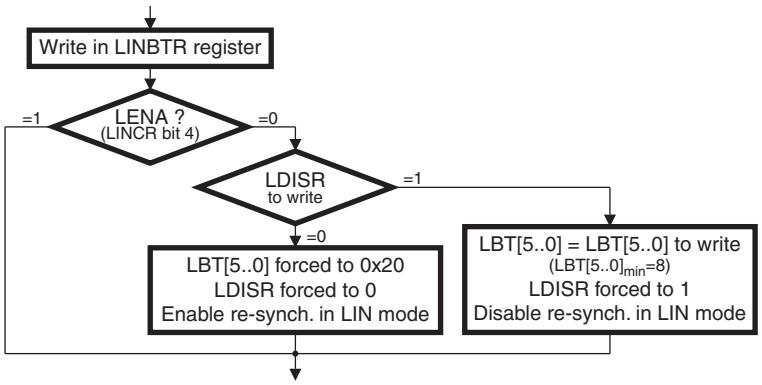
Handling LBT[5..0]

LDISR bit of LINBTR register is used to:

- Disable the re-synchronization (for instance in the case of LIN MASTER node),
- To enable the setting of LBT[5..0] (to manually adjust the baud rate especially in the case of UART mode). A minimum of 8 is required for LBT[5..0] due to the sampling operation.

Note that the LENA bit of LINCR register is important for this handling (see Figure 4-75 on page 197).

Figure 4-75. Handling LBT[5..0]



- **Bit 2 - LENIDOK: Enable Identifier Interrupt**
  - 0 = Identifier interrupt masked,
  - 1 = Identifier interrupt enabled.
- **Bit 1 - LENTXOK: Enable Transmit Performed Interrupt**
  - 0 = Transmit performed interrupt masked,
  - 1 = Transmit performed interrupt enabled.
- **Bit 0 - LENRXOK: Enable Receive Performed Interrupt**
  - 0 = Receive performed interrupt masked,
  - 1 = Receive performed interrupt enabled.

4.16.6.4 LIN Error Register - LINERR

Bit	7	6	5	4	3	2	1	0	
	LABORT	**LTOERR**	**LOVERR**	**LFERR**	**LSERR**	**LPERR**	**LCERR**	**LBERR**	**LINERR**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 - LABORT: Abort Flag**
  - 0 = No warning,
  - 1 = LIN abort command occurred.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 6 - LTOERR: Frame_Time_Out Error Flag**
  - 0 = No error,
  - 1 = Frame_Time_Out error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 5 - LOVERR: Overrun Error Flag**
  - 0 = No error,
  - 1 = Overrun error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 4 - LFERR: Framing Error Flag**
  - 0 = No error,
  - 1 = Framing error.

  This bit is cleared when LERR bit in LINSIR is cleared.
- **Bit 3 - LSERR: Synchronization Error Flag**
  - 0 = No error,
  - 1 = Synchronization error.

  This bit is cleared when LERR bit in LINSIR is cleared.

4.18.12 Register Description

4.18.12.1 ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
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	

- **Bit 7:6 – REFS1:REFS0: Voltage Reference Selection Bits**

These bits and AREFEN bit from the Analog Miscellaneous Control Register (AMISCR) select the voltage reference for the ADC, as shown in Table 4-58. If these bits are changed during a conversion, the change will not go in effect until this conversion is complete (ADIF in ADCSRA register is set). Whenever these bits are changed, the next conversion will take 25 ADC clock cycles. If active channels are used, using AVCC or an external AREF higher than (AVcc - 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin.

Table 4-58. Voltage Reference Selections for ADC

REFS1	REFS0	AREFEN	Voltage Reference (V_{REF}) Selection
X	0	0	AVcc used as Voltage Reference, disconnected from AREF pin.
X	0	1	External Voltage Reference at AREF pin (AREF ≥ 2.0V)
0	1	0	Internal 1.1V Voltage Reference.
1	1	0	Internal 2.56V Voltage Reference.

- **Bit 5 – ADLAR: ADC Left Adjust Result**

The ADLAR bit affects the presentation of the ADC conversion result in the ADC Data Register. Write one to ADLAR to left adjust the result. Otherwise, the result is right adjusted. Changing the ADLAR bit will affect the ADC Data Register immediately, regardless of any ongoing conversions. For a complete description of this bit, see “ADCL and ADCH – The ADC Data Register” on page 232.

- **Bits 4:0 – MUX4:0: Analog Channel and Gain Selection Bits**

These bits select which combination of analog inputs are connected to the ADC. In case of differential input, gain selection is also made with these bits. Refer to Table 4-59 for details. If these bits are changed during a conversion, the change will not go into effect until this conversion is complete (ADIF in ADCSRA register is set).

4.18.12.7 AMISCR – Analog Miscellaneous Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	AREFEN	XREFEN	ISRCEN	AMISCR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, they must be written to zero when AMISCR is written.

- **Bit 2 – AREFEN: External Voltage Reference Input Enable**

When this bit is written logic one, the voltage reference for the ADC is input from AREF pin as described in Table 4.18.11 on page 228. If active channels are used, using AVcc or an external AREF higher than (AVcc - 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin. It is recommended to use DIDR register bit function (digital input disable) when AREFEN is set.

- **Bit 1 – XREFEN: Internal Voltage Reference Output Enable**

When this bit is written logic one, the internal voltage reference 1.1V or 2.56V is output on XREF pin as described in Table 4.18.11 on page 228. It is recommended to use DIDR register bit function (digital input disable) when XREFEN is set.

Bit	7	6	5	4	3	2	1	0
Rd (Z=0x0003)	FHB7	FHB6	FHB5	FHB4	FHB3	FHB2	FHB1	FHB0

Similarly, when reading the Extended Fuse byte (EFB), load 0x0002 in the Z-pointer. When an LPM instruction is executed within three cycles after the RFLB and SPMEN bits are set in the SPMCSR, the value of the Extended Fuse byte will be loaded in the destination register as shown below. See Table 4-69 on page 251 for detailed description and mapping of the Extended Fuse byte.

Bit	7	6	5	4	3	2	1	0
Rd (Z=0x0002)	–	–	–	–	–	–	–	EFB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

4.21.2.4 *Reading the Signature Row from Software*

To read the Signature Row from software, load the Z-pointer with the signature byte address given in Table 4-65 on page 246 and set the SIGRD and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the SIGRD and SPMEN bits are set in SPMCSR, the signature byte value will be loaded in the destination register. The SIGRD and SPMEN bits will auto-clear upon completion of reading the Signature Row Lock bits or if no LPM instruction is executed within three CPU cycles. When SIGRD and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Table 4-65. Signature Row Addressing

Signature Byte	Z-Pointer Address
Device Signature Byte 0	0x0000
Device Signature Byte 1	0x0002
Device Signature Byte 2	0x0004
8MHz RC Oscillator Calibration Byte	0x0001
TSOFFSET - Temp Sensor Offset	0x0003
TSGAIN - Temp Sensor Gain	0x0005

Note: All other addresses are reserved for future use.

4.21.2.5 *Preventing Flash Corruption*

During periods of low Vcc, the Flash program can be corrupted because the supply voltage is too low for the CPU and the Flash to operate properly. These issues are the same as for board level systems using the Flash, and the same design solutions should be applied.

A Flash program corruption can be caused by two situations when the voltage is too low.

- First, a regular write sequence to the Flash requires a minimum voltage to operate correctly.
- Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR® RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD) if the operating voltage matches the detection level. If not, an external low Vcc reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
2. Keep the AVR core in Power-down sleep mode during periods of low Vcc. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR Register and thus the Flash from unintentional writes.

4.21.2.6 *Programming Time for Flash when Using SPM*

The calibrated RC Oscillator is used to time Flash accesses. Table 4-66 shows the typical programming time for Flash accesses from the CPU.

Table 4-66. SPM Programming Time

Symbol	Min Programming Time	Max Programming Time
Flash write (Page Erase, Page Write, and write Lock bits by SPM)	3.7ms	4.5ms

```
Rdloop:
  lpm     r0, Z+
  ld      r1, Y+
  cpse    r0, r1
  rjmp    Error
  sbiw    loophi:looplo, 1          ; use subi for PAGESIZEB<=256
  brne    Rdloop

  ; To ensure compatibility with devices supporting Read-While-Write
  ; Return to RWW section
  ; Verify that RWW section is safe to read

Return:
  in      temp1, SPMCSR
  sbrs    temp1, RWWSB     ; If RWWSB is set, the RWW section is not ready
yet
  ret
  ; Clear temporary page buffer
  ldi     spmcsrval, (1<<CPTB) | (1<<SELFPGEN)
  call    Do_spm
  rjmp    Return

Do_spm:
  ; Check for previous SPM complete
Wait_spm:
  in      temp1, SPMCSR
  sbrc    temp1, SELFPGEN
  rjmp    Wait_spm
  ; Input: spmcsrval determines SPM action
  ; Disable interrupts if enabled, store status
  in      temp2, SREG
  cli
  ; Check that no EEPROM write access is present
Wait_ee:
  sbic    EECR, EEPE
  rjmp    Wait_ee
  ; SPM timed sequence
  out     SPMCSR, spmcsrval
  spm
  ; Restore SREG (to enable interrupts if originally enabled)
  out     SREG, temp2
  ret
```

F. Repeat B through E until the entire buffer is filled or until all data within the page is loaded.

While the lower bits in the address are mapped to words within the page, the higher bits address the pages within the FLASH. This is illustrated in Figure 4-99 on page 257. Note that if less than eight bits are required to address words in the page (pagesize < 256), the most significant bit(s) in the address low byte are used to address the page when performing a Page Write.

G. Load Address High byte

1. Set XA1, XA0 to “0,0”. This enables address loading.
2. Set BS1 to “1”. This selects high address.
3. Set DATA = Address high byte (0x00 - 0xFF).
4. Give XTAL1 a positive pulse. This loads the address high byte.

H. Program Page

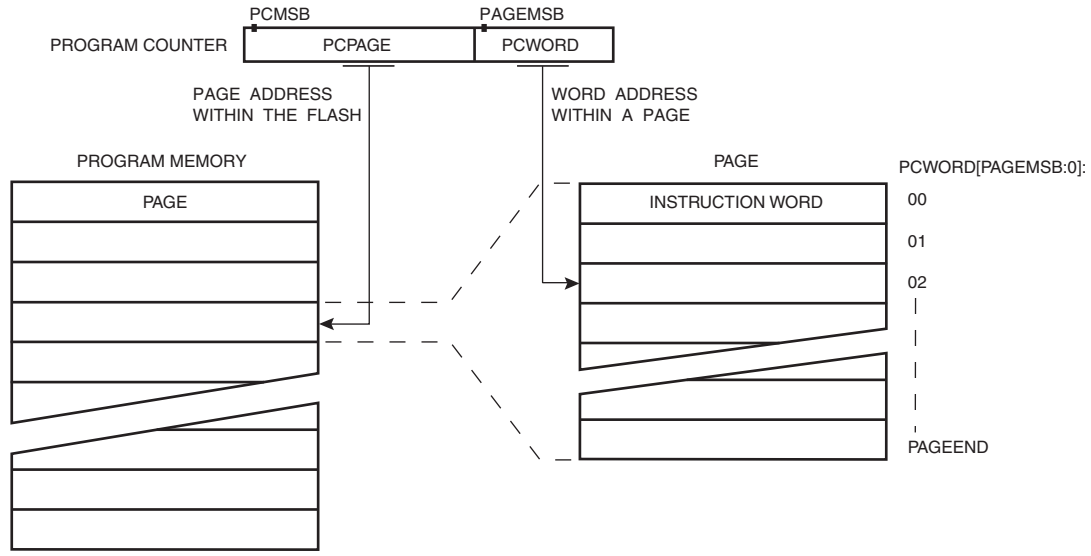
1. Give $\overline{WR}$ a negative pulse. This starts programming of the entire page of data. RDY/$\overline{BSY}$ goes low.
2. Wait until RDY/$\overline{BSY}$ goes high (See Figure 4-100 for signal waveforms).

I. Repeat B through H until the entire Flash is programmed or until all data has been programmed.

J. End Page Programming

1. Set XA1, XA0 to “1,0”. This enables command loading.
2. Set DATA to “0000 0000 $_b$”. This is the command for No Operation.
3. Give XTAL1 a positive pulse. This loads the command, and the internal write signals are reset.

Figure 4-99. Addressing the Flash Which is Organized in Pages



Note: PCPAGE and PCWORD are listed in

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

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600

Atmel Asia Limited
Unit 01-5 & 16, 19/F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
JAPAN
Tel: (+81) (3) 3523-3551
Fax: (+81) (3) 3523-7581