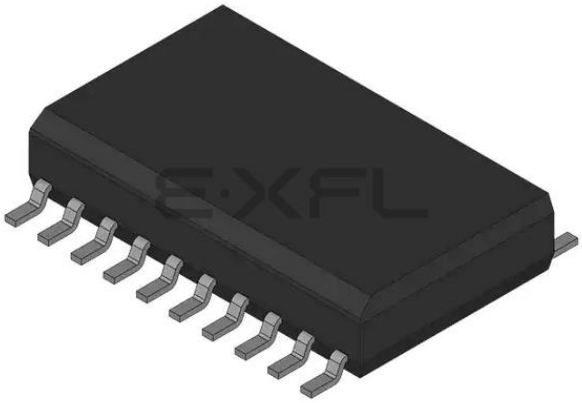


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI
Peripherals	Brown-out Detect/Reset, PWM, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	20-SOIC (0.295", 7.50mm Width)
Supplier Device Package	20-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/atmel/at90pwm81-16sn

- **Bits 5..4 – EEPM1 and EEPM0: EEPROM Programming Mode Bits**

The EEPROM Programming mode bit setting defines which programming action that will be triggered when writing EEWE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in Table 4-1. While EEWE is set, any write to EEPMn will be ignored. During reset, the EEPMn bits will be reset to 0b00 unless the EEPROM is busy programming.

Table 4-1. EEPROM mode bits.

EEPM1	EEPM0	Programming time	Operation
0	0	3.4ms	Erase and write in one operation (atomic operation)
0	1	1.8ms	Erase only
1	0	1.8ms	Write only
1	1	–	Flush temporary EEPROM page buffer

- **Bit 3 – EERIE: EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEWE is cleared. The interrupt will not be generated during EEPROM write or SPM.

- **Bit 2 – EEMWE: EEPROM Master Write Enable**

The EEMWE bit determines whether setting EEWE to one causes the EEPROM to be written. When EEMWE is set, setting EEWE within four clock cycles will write data to the EEPROM at the selected address If EEMWE is zero, setting EEWE will have no effect. When EEMWE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEWE bit for an EEPROM write procedure.

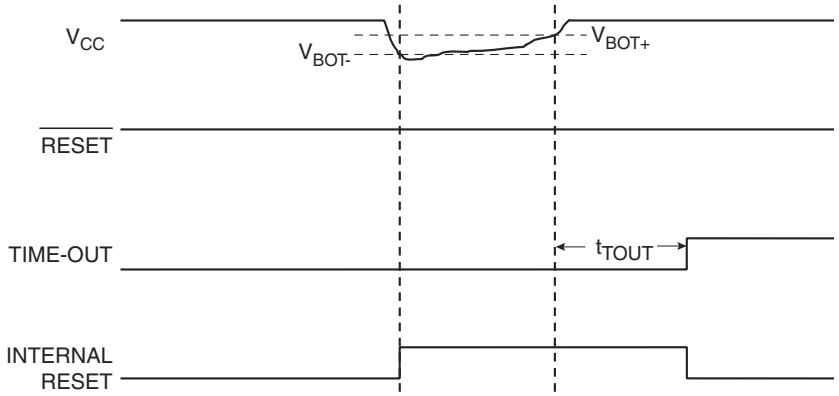
- **Bit 1 – EEWE: EEPROM Write Enable**

The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be written to one to write the value into the EEPROM. The EEMWE bit must be written to one before a logical one is written to EEWE, otherwise no EEPROM write takes place. The following procedure should be followed when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEWE becomes zero.
2. Wait until SPMEN (Store Program Memory Enable) in SPMCSR (Store Program Memory Control and Status Register) becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMWE bit while writing a zero to EEWE in EECR.
6. Within four clock cycles after setting EEMWE, write a logical one to EEWE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted. See “Boot Loader Support – Read-While-Write Self-Programming” on page 233 for details about Boot programming.

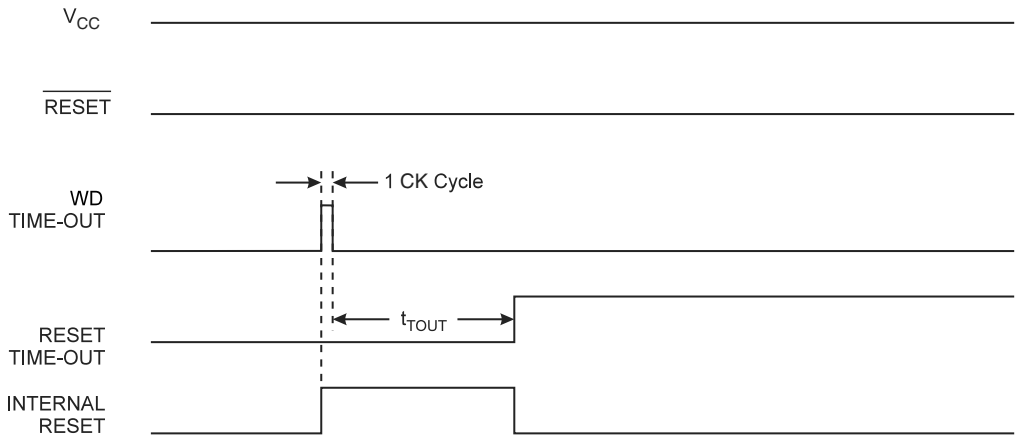
Figure 7-5. Brown-out reset during operation.



7.1.6 Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 56 for details on operation of the Watchdog Timer.

Figure 7-6. Watchdog reset during operation.



7.2 System Control registers

7.2.1 MCUSR - MCU Status Register

The MCU Status Register provides information on which reset source caused an MCU reset.

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

According to PSC n Input A Control Register (see “PFRCnA - PSC n Input A Control Register” on page 141), PSC n input A can act as a Retrigger or Fault input.

Each part A or B can be triggered by up to four signals as defined per Table 12-18 on page 139 and Table 12-19 on page 139.

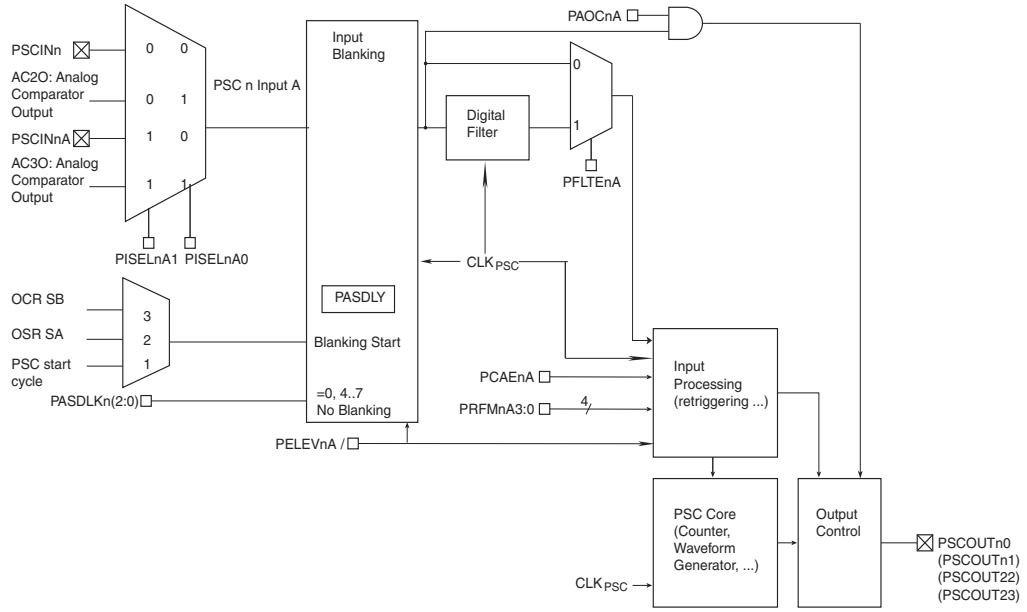
Part A of PSC has also a blanking module allowing to cancel unwanted transitions which may appear on the PSC n input A during a certain period of time.

The blanking start is defined by the bits PASDLKn(2:0) as per Table 12-14 on page 138.

The blanking duration is defined by the register PASDLYn. If the blanking is selected by the corresponding PASDLKn(2:0) bit, all transitions which may appears from the blanking start until a time period are ignored.

Blanking is level sensitive, that is, a pulse started in the blanking window and still at active level after the window will generate a valid retriggering event.

Figure 12-14. PSC input module A.



PSC input module B is shown on Table 12-15 on page 116.

According to PSC n Input B Control Register (see “PFRCnB - PSC n Input B Control Register” on page 141), PSC n input B can act as a Retrigger or Fault input.

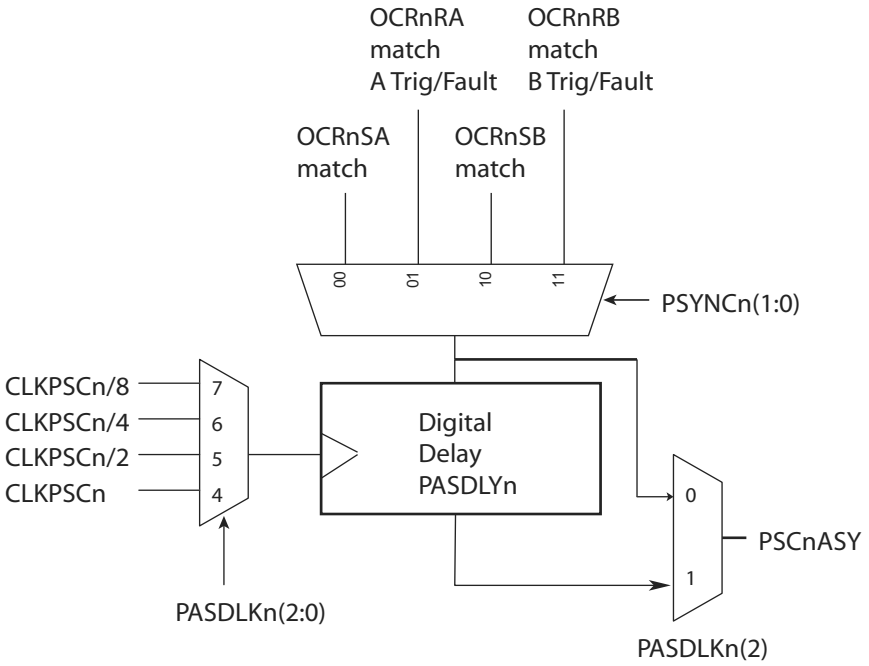
12.20 Analog Synchronization

PSC generates a signal to synchronize the sample and hold or the ADC start; synchronization is mandatory for measurements.

This signal can be selected between all falling or rising edge of PSCn0 or PSCn1 outputs as defined per Table 12-11 on page 134 and Table 12-12 on page 135.

The signal can be shifted by a digital delay defined by the register PASDLY. The shifting clock can be either Clkpsc or Clkpsc/4, as described per Bit 7, 6, 5– PASDLKn(2:0): Analog Synchronization Output Delay or Input Blanking select on page 137.

Figure 12-40. Analog synchronization.



12.21 Interrupt Handling

As each PSC can be dedicated for one function, each PSC has its own interrupt system.

List of interrupt sources:

- Counter reload (end of On Time 1)
- End of Enhanced Cycle
- PSC Input event (active edge or at the beginning of level configured event)
- PSC Mutual Synchronization Error

12.25.10 PFRCnA - PSC n Input A Control Register

Bit	7	6	5	4	3	2	1	0	
	PCAEnA	PISELnA0	PELEVnA0	PFLTEnA	PRFMnA3	PRFMnA2	PRFMnA1	PRFMnA0	PFRCnA
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	

12.25.11 PFRCnB - PSC n Input B Control Register

Bit	7	6	5	4	3	2	1	0	
	PCAEnB	PISELnB0	PELEVnB0	PFLTEnB	PRFMnB3	PRFMnB2	PRFMnB1	PRFMnB0	PFRCnB
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 Input Control Registers are used to configure the 2 PSC's Retrigger/Fault block A & B. The 2 blocks are identical, so they are configured on the same way.

- **Bit 7 – PCAEnx: PSC n Capture Enable Input Part x**

Writing this bit to one enables the capture function when external event occurs on input selected as input for Part x (see PISELnx1:0 bit in the same register).

- **Bit 6 – PISELnx0: PSC n Input Select for Part x**

Together with PISELnx1 in PCNFEn register, defines active signal on PSC module A. See Table 12-18 on page 139 and Table 12-19 on page 139.

- **Bit 5 –PELEVnx0: PSC n Edge Level Selector of Input Part x**

Together with PELEVnx1 n PCNFEn register, defines active edge & level on PSC part x; See Table 12-16 on page 138 and Table 12-17 on page 139.

- **Bit 4 – PFLTEnx: PSC n Filter Enable on Input Part x**

Setting this bit (to one) activates the Input Capture Noise Canceler. When the noise canceler is activated, the input from the retrigger pin is filtered. The filter function requires four successive equal valued samples of the retrigger pin for changing its output. The Input Capture is therefore delayed by four oscillator cycles when the noise canceler is enabled.

- **Bit 3:0 – PRFMnx3:0: PSC n Fault Mode**

These four bits define the mode of operation of the Fault or Retrigger functions.

(see Table 12-7 on page 120 for more explanations).

Table 12-21. Level sensitivity and Fault Mode operation.

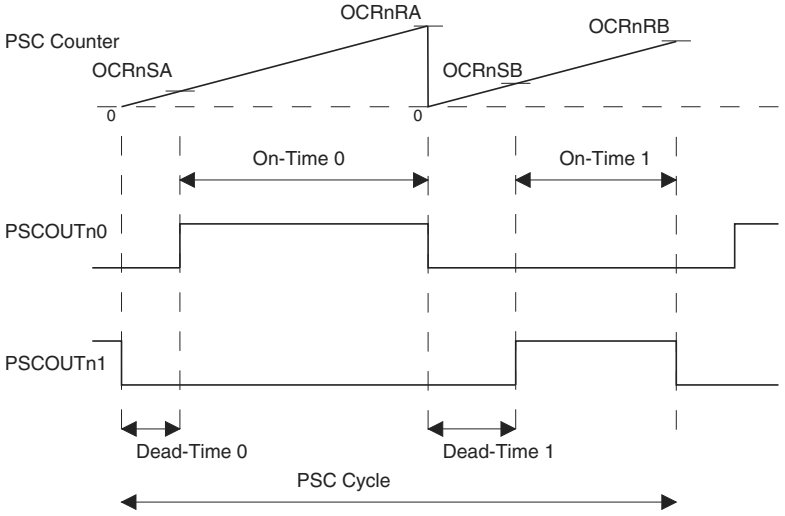
PRFMnx3:0	Description
0000b	No action, PSC Input is ignored
0001b	"PSC Input Mode 1: Stop signal, Jump to Opposite Dead-Time and Wait", page 121
0010b	"PSC Input Mode 2: Stop signal, Execute Opposite Pulse and Wait", page 122
0011b	"PSC Input Mode 3: Stop signal, Execute Opposite Pulse while Fault active", page 123

13.5.2.2 Two Ramp Mode

In Two Ramp mode, the whole cycle is divided in two moments:

- One moment for PSCr0 description with OT0 which gives the time of the whole moment
- One moment for PSCr1 description with OT1 which gives the time of the whole moment

Figure 13-5. PSCr0 & PSCr1 basic waveforms in Two Ramp mode.



PSCOUTr0 and PSCOUTr1 signals are defined by On-Time 0, Dead-Time 0, On-Time 1 and Dead-Time 1 values with:

On-Time 0 = (OCRrRAH/L - OCRrSAH/L) × 1/Fclkpsc

On-Time 1 = (OCRrRBH/L - OCRrSBH/L) × 1/Fclkpsc

Dead-Time 0 = (OCRrSAH/L + 1) × 1/Fclkpsc

Dead-Time 1 = (OCRrSBH/L + 1) × 1/Fclkpsc

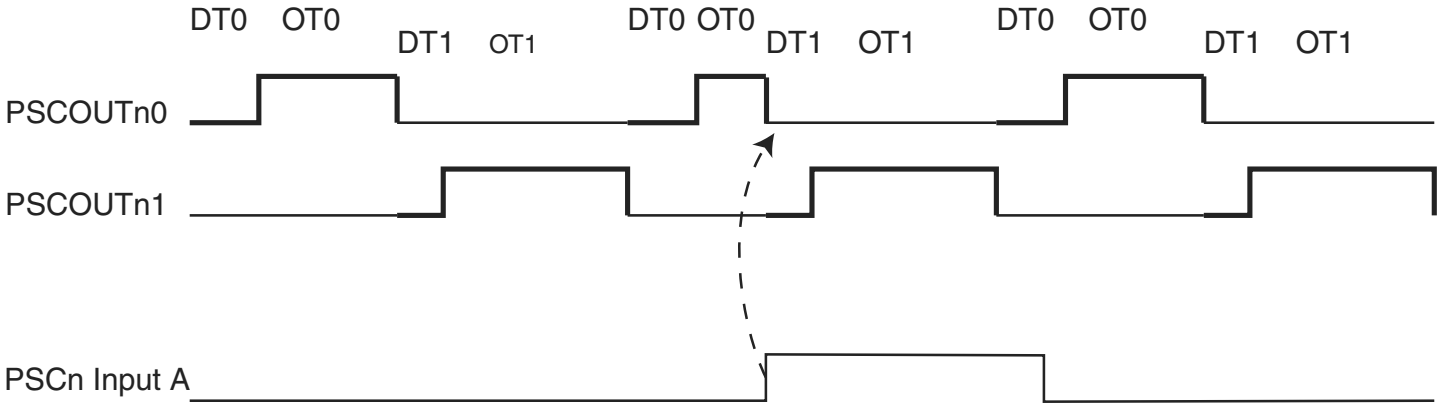
Note: Minimal value for Dead-Time 0 and Dead-Time 1 = 1/Fclkpsc.

13.5.2.3 One Ramp Mode

In One Ramp mode, PSCOUTr0 and PSCOUTr1 outputs can overlap each other.

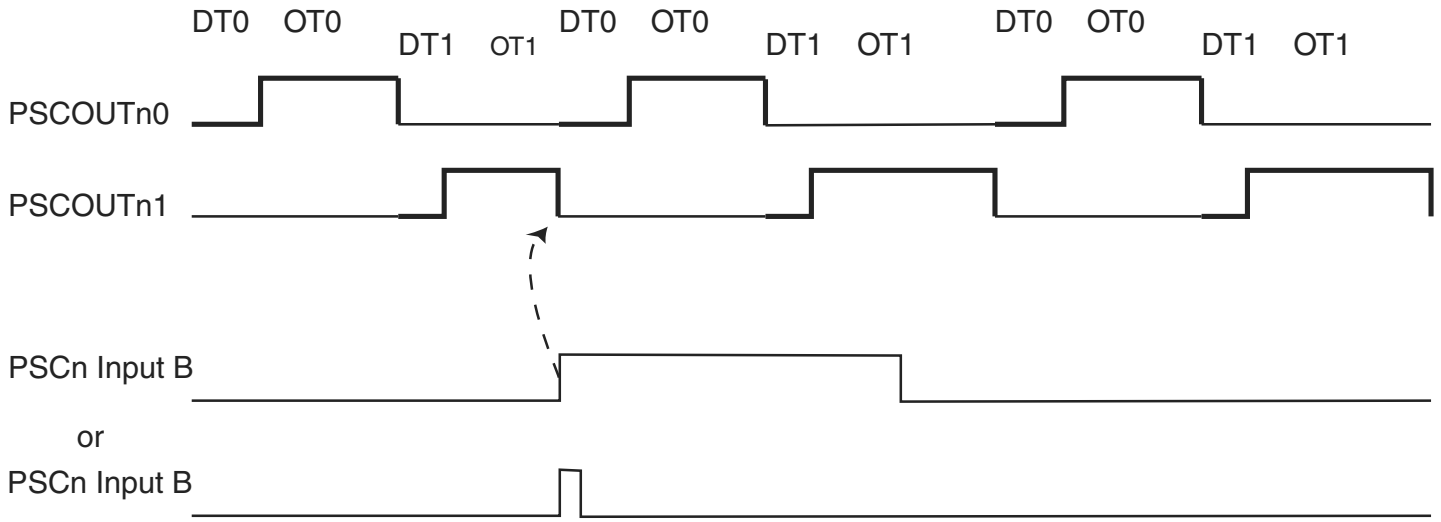
13.16 PSCR Input Mode 8: Edge Retrigger PSC

Figure 13-26. PSCR behavior versus PSCr Input A in Mode 8.



The output frequency is modulated by the occurrence of significative edge of retriggering input.

Figure 13-27. PSCR behavior versus PSCr Input B in Mode 8.



The output frequency is modulated by the occurrence of significative edge of retriggering input.

The retrigger event is taken into account only if it occurs during the corresponding On-Time.

Note: In one ramp mode, the retrigger event on input A resets the whole ramp. So the PSCR doesn't jump to the opposite dead-time.

Table 13-6. Available input modes according to running modes. (Continued)

Input mode number:	1 Ramp mode	2 Ramp mode	4 Ramp mode
10	Do not use		
11			
12			
13			
14	Valid	Valid	Valid
15	Do not use		

13.18.2 Event Capture

The PSCR can capture the value of time (PSCR counter) when a retrigger event or fault event occurs on PSCR inputs. This value can be read by software in PICRrH/L register.

13.18.3 Using the Input Capture Unit

The main challenge when using the Input Capture unit is to assign enough processor capacity for handling the incoming events. The time between two events is critical. If the processor has not read the captured value in the PICR1 Register before the next event occurs, the PICR1 will be overwritten with a new value. In this case the result of the capture will be incorrect.

When using the Input Capture interrupt, the PICR1 Register should be read as early in the interrupt handler routine as possible. Even though the Input Capture interrupt has relatively high priority, the maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

13.19 Analog Synchronization

PSCR generates a signal to synchronize the sample and hold; synchronization is mandatory for measurements.

This signal can be selected between all falling or rising edge of PSCr0 or PSCr1 outputs.

13.20 Interrupt Handling

List of interrupt sources:

- Counter reload (end of On Time 1)
- PSCR Input event (active edge or at the beginning of level configured event)

13.21 PSC Clock Sources

PSCR must be able to generate high frequency with enhanced resolution.

The PSCR has two clock inputs:

- CLK PLL from the PLL
- CLK I/O

13.23.7 PCTL0 - PSCR Control Register

Bit	7	6	5	4	3	2	1	0	
	PPRE01	PPRE00	PBFM01	PAOC0B	PAOC0A	PBFM00	PCCYC0	PRUN0	PCTL0
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 – PPRE01:0 : PSCR Prescaler Select**

This two bits select the PSCR input clock division factor. All generated waveform will be modified by this factor.

Table 13-12. PSCR prescaler selection.

PPRE01	PPRE00	Description
0	0	No divider on PSCR input clock
0	1	Divide the PSCR input clock by 4
1	0	Divide the PSCR input clock by 32
1	1	Divide the PSCR clock by 256

14. Serial Peripheral Interface – SPI:

14.1 Features

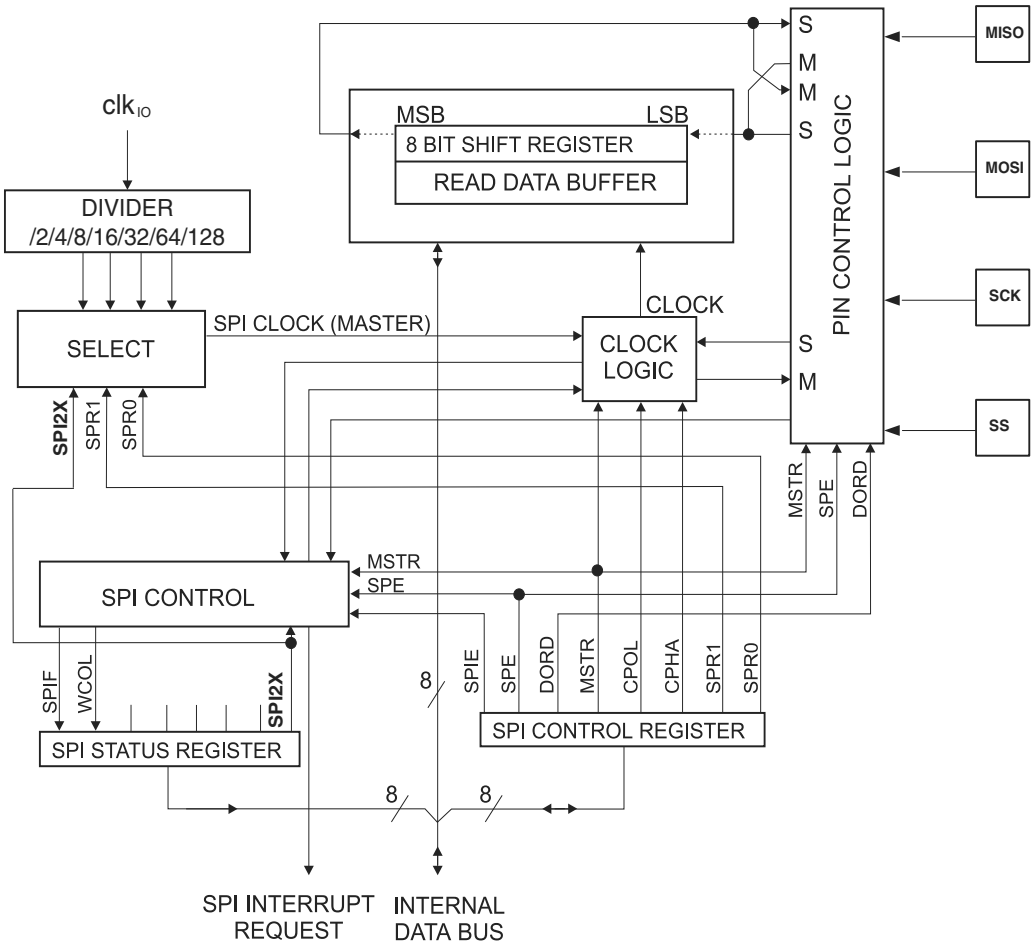
- **Full-duplex, three-wire synchronous data transfer**
- **Master or Slave operation**
- **LSB first or MSB first data transfer**
- **Seven programmable bit rates**
- **End of transmission interrupt flag**
- **Write collision flag protection**
- **Wake-up from idle mode**
- **Double speed (CK/2) Master SPI mode**

14.2 Overview

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT90PWM81/161 and peripheral devices or between several AVR devices.

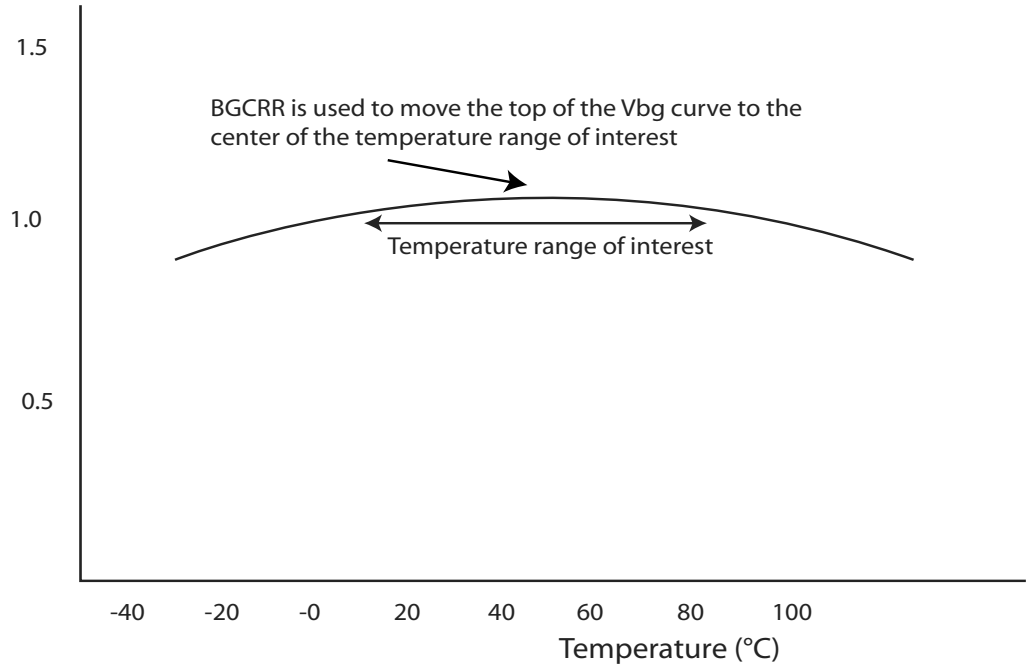
The AT90PWM81/161 SPI includes the following features.

Figure 14-1. SPI block diagram [1].



Note: 1. Refer to Figure 2-1 on page 3, and Table 9-3 on page 75 for SPI pin placement.

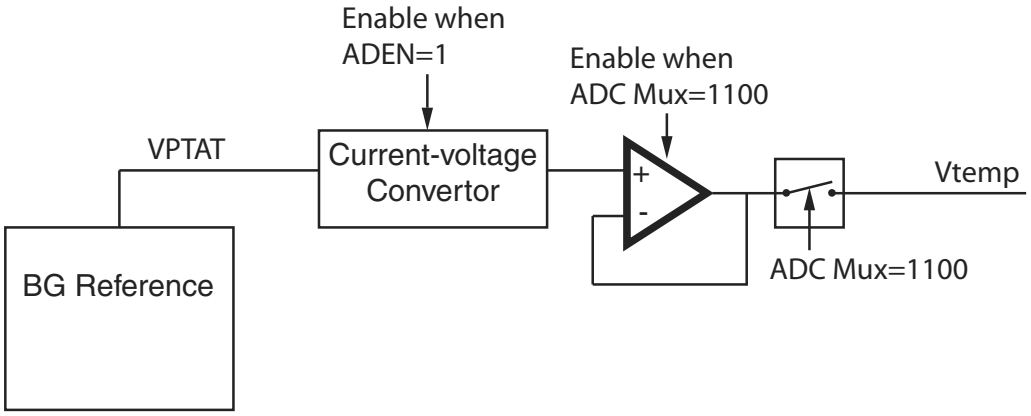
Figure 15-2. Illustration of Vbg as a function of temperature.



15.4 Temperature Measurement

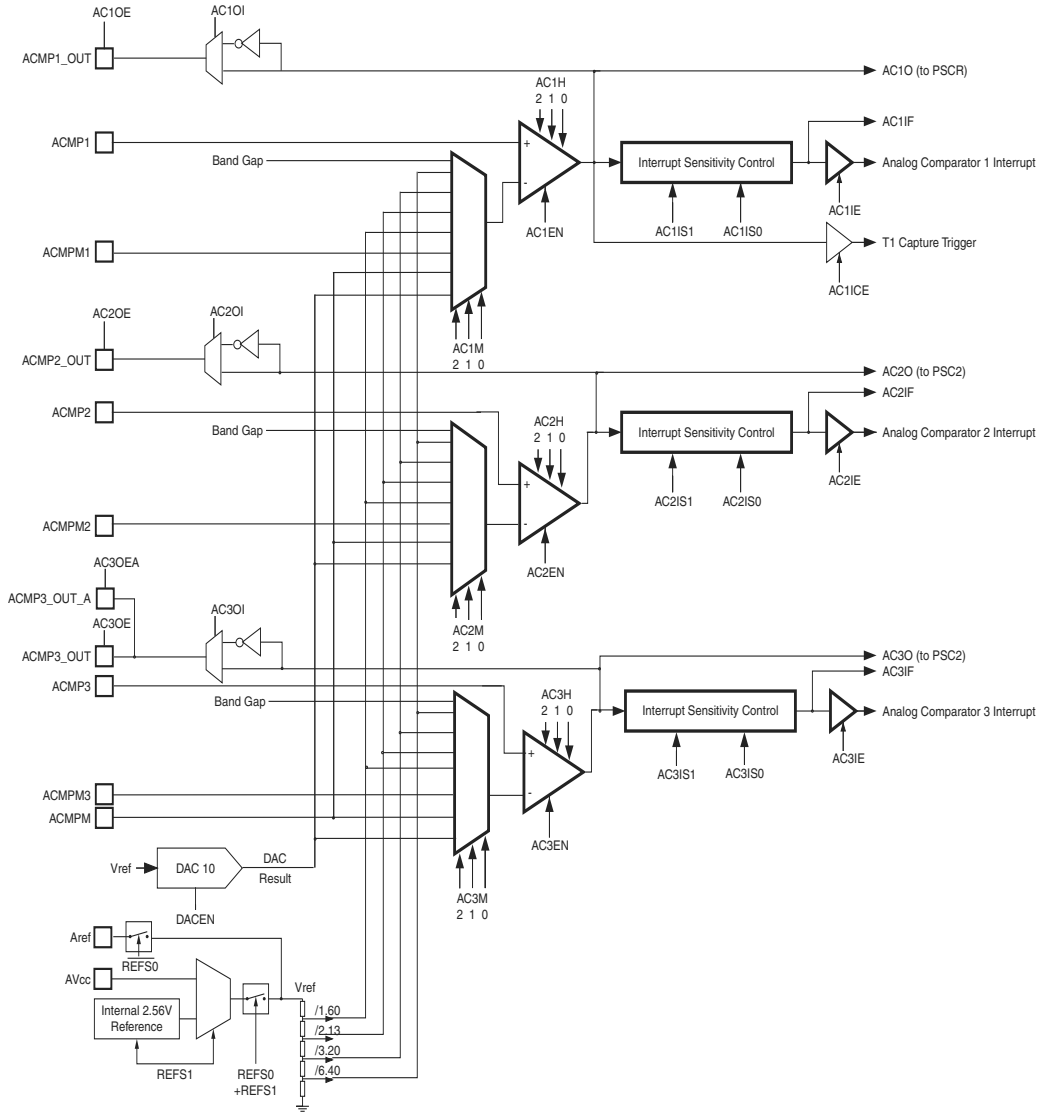
The temperature measurement is based on an on-chip temperature sensor that is coupled to a single ended ADC12 channel, as shown on Figure 15-3.

Figure 15-3. Temperature sensor circuitry.



Selecting the ADC12 channel by writing the MUX3..0 bits in ADMUX register to “1100” enables the temperature sensor (see “ADMUX - ADC Multiplexer Register” on page 217). The recommended ADC voltage reference source is the internal 2.56V voltage reference for temperature sensor measurement. When the temperature sensor is enabled, the ADC converter can be used in single conversion mode to measure the voltage over the temperature sensor. The amplifier allows to charge the ADC sample capacitor at full CKadc clock speed. The measured voltage has a linear relationship to temperature as described in Table 15-1 on page 193. When the voltage reference equals 2.56V, the conversion result has approximately a 1LSB/°C (or 2.5mV/°C)

Figure 16-1. Analog comparator block diagram.



Notes: 1. Refer to Figure 2-1 on page 3 and Figure 2-2 on page 4 for Analog Comparator pin placement.

2. The voltage on V_{REF} is defined in Table 17-3, “ADC voltage reference selection.,” on page 218.

16.4.4 ACnECON - Analog Comparator n Extended Control Register

Bit	7	6	5	4	3	2	1	0	
			ACnOI	ACnOE	AC1ICE	ACnH2	ACnH1	ACnH0	ACnECON
Read/Write			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– Reserved**

- **Bit 5– AC1OI: Analog Comparator n Output Invert**

Set this bit to invert the analog comparator n output.
Clear this bit to keep the analog comparator n output.

- **Bit 4– AC1OE: Analog Comparator n Output Enable**

Set this bit to enable the analog comparator n output pin.
Clear this bit to disable the analog comparator n output pin.

- **Bit 3 – AC1ICE: Analog Comparator 1 Interrupt Capture Enable bit**

Set this bit to enable the input capture of the Timer/Counter1 on the analog comparator event. The comparator output is in this case directly connected to the input capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the ICIE1 bit in the Timer Interrupt Mask Register (TIMSK1) must be set.

In case ICES1 bit (“TCCR1B - Timer/Counter1 Control Register B” on page 97) is set high, the rising edge of AC3O is the capture/trigger event of the Timer/Counter1, in case ICES1 is set to zero, it is the falling edge which is taken into account.

Clear this bit to disable this function. In this case, no connection between the Analog Comparator and the input capture function exists.

- **Bit 2, 1, 0– ACnH2, ACnH1, ACnH0: Analog Comparator n Hysteresis select**

These 3 bits determine the hysteresis value of the analog comparator.
The different setting are shown in Table 16-7.

Table 16-7. Analog cComparator n hysteresis selection.

AC1M2	AC1M1	AC1M0	Description
0	0	0	No hysteresis
0	0	1	Hysteresis + 10mV
0	1	0	Hysteresis - 10mV
0	1	1	Hysteresis ±10mV
1	0	0	Reserved
1	0	1	Hysteresis + 25mV
1	1	0	Hysteresis - 25mV
1	1	1	Hysteresis ±25mV

17. Analog to Digital Converter - ADC

17.1 Features

- **10-bit resolution**
- **0.5LSB integral non-linearity**
- **±2LSB absolute accuracy**
- **8µs - 250µs conversion time**
- **Up to 120kSPS at maximum resolution**
- **11 multiplexed single ended input channels**
- **One differential input channels with accurate (5%) programmable gain 5, 10, 20, and 40**
- **Optional left adjustment for ADC result readout**
- **0 - V_{CC} ADC input voltage range**
- **Selectable 2.56V ADC reference voltage**
- **Free running or single conversion mode**
- **ADC start conversion by auto triggering on interrupt sources**
- **Interrupt on ADC conversion complete**
- **Sleep mode noise canceler**
- **Temperature sensor**

The AT90PWM81/161 features a 10-bit successive approximation ADC. The ADC is connected to an 15-channel Analog Multiplexer which allows eleven single-ended input. The single-ended voltage inputs refer to 0V (GND).

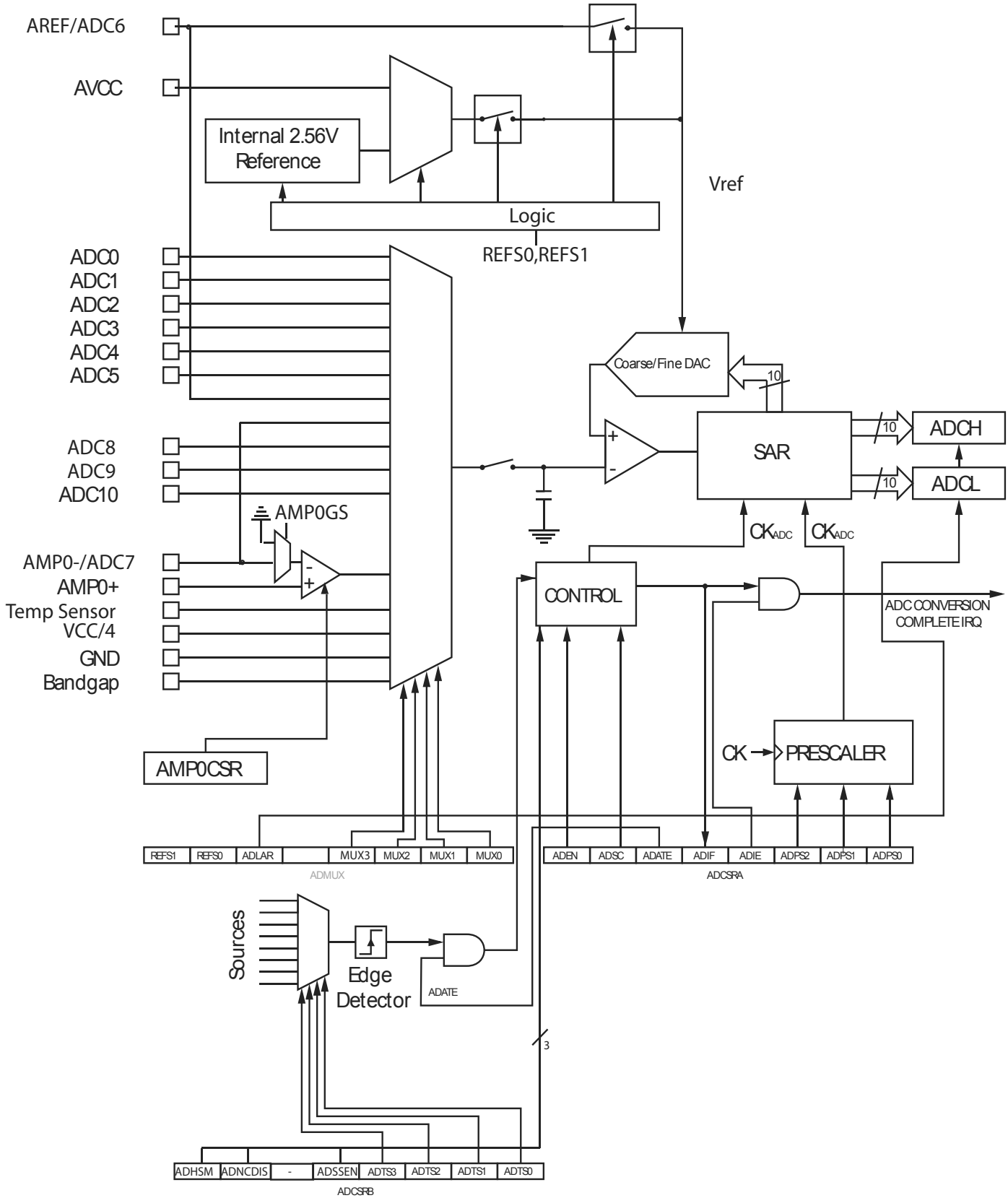
The device also supports 2 differential voltage input combinations which are equipped with a programmable gain stage, providing amplification steps of 14dB (5×), 20dB (10×), 26dB (20×), or 32dB (40×) on the differential input voltage before the A/D conversion. On the amplified channels, 8-bit resolution can be expected.

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 17-1 on page 204.

The ADC has a separate analog supply voltage pin, AV_{CC}. AV_{CC} must not differ more than ±0.3V from V_{CC}. See the paragraph “ADC Noise Canceler” on page 210 on how to connect this pin.

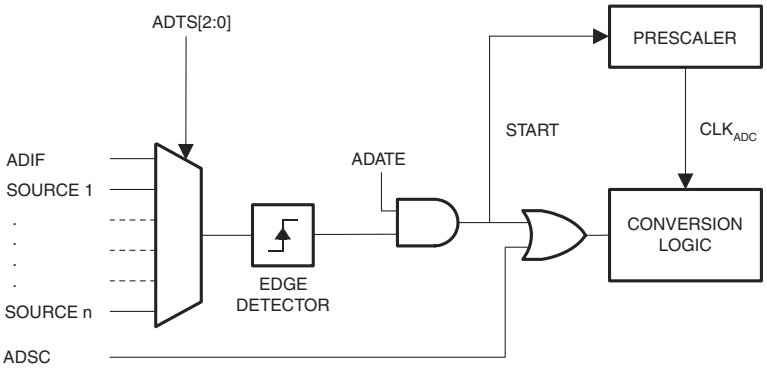
Internal reference voltages of nominally 2.56V or AV_{CC} are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.

Figure 17-1. Analog to digital converter block schematic.



vated by setting the bit ADSSEN in ADCSRB register. In this case the synchronization signal is blocked until the ADCH registed is read.

Figure 17-2. ADC auto trigger logic.

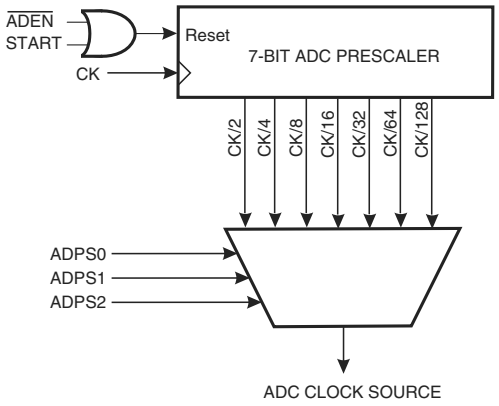


Using the ADC Interrupt Flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in Free Running mode, constantly sampling and updating the ADC Data Register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA. In this mode the ADC will perform successive conversions independently of whether the ADC Interrupt Flag, ADIF is cleared or not. The free running mode is not allowed on the amplified channels.

If Auto Triggering is enabled, single conversions can be started by writing ADSC in ADCSRA to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

17.4 Prescaling and Conversion Timing

Figure 17-3. ADC prescaler.



By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 2MHz to get maximum resolution. If a lower resolution than 10 bits is needed, the input clock frequency to the ADC can be higher than 2MHz to get a higher sample rate.

The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100kHz. The prescaling is set by the ADPS bits in ADCSRA.

- **Bit 5, 4– AMP0G1, 0: Amplifier 0 Gain Selection Bits**

These two bits determine the gain of the amplifier 0.
The different setting are shown in Table 17-7.

Table 17-7. Amplifier 0 gain selection.

AMP0G1	AMP0G0	Description
0	0	Gain 5
0	1	Gain 10
1	0	Gain 20
1	1	Gain 40

To ensure an accurate result, after the gain value has been changed, the amplifier input needs to have a quite stable input value during at least four Amplifier synchronization clock periods.

- **Bit 3– AMP0GS: Amplifier 0 Ground Select of AMP0**

This bit select negative input of the amplifier:

Set this bit to ground the Amplifier 0 negative input.
Clear this bit to normally use the Amplifier 0 differential input.

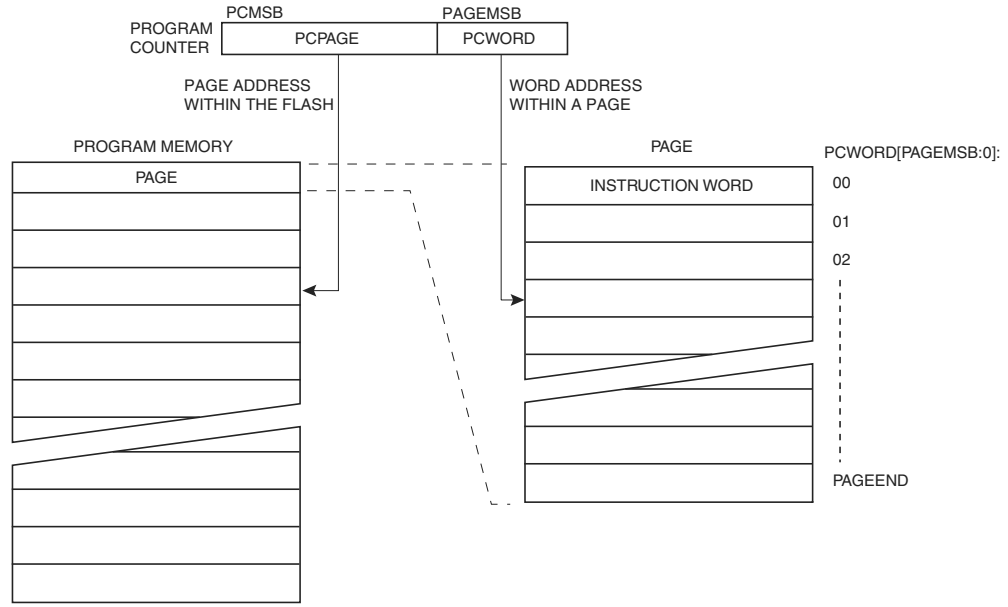
- **Bit 1, 0– AMP0TS1, AMP0TS0: Amplifier 0 Trigger Source Selection Bits**

In accordance with the Table 17-8, these two bits select the event which will generate the trigger for the amplifier 0. This trigger source is necessary to start the conversion on the amplified channel.

Table 17-8. AMP0 auto trigger source selection.

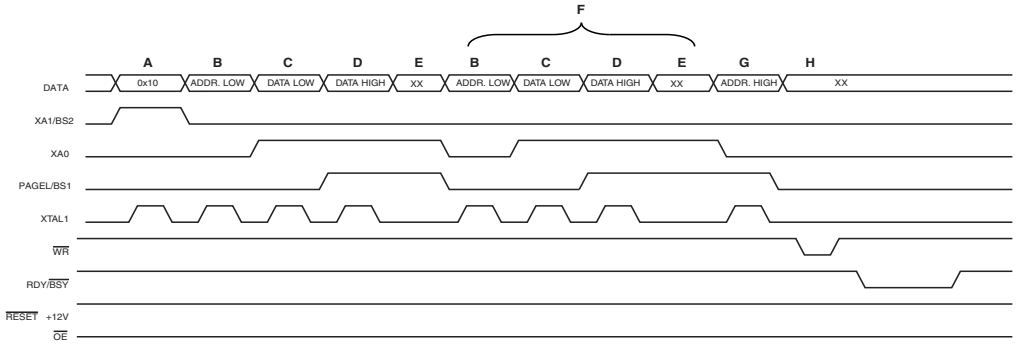
AMP0TS1	AMP0TS0	Description
0	0	Auto synchronization on ADC Clock/8
0	1	Trig on PSCRASY
1	0	
1	1	Trig on PSC2ASY

Figure 21-2. Addressing the flash, which is organized in pages [1].



Note: 1. PCPAGE and PCWORD are listed in Table 21-12 on page 254.

Figure 21-3. Programming the flash waveforms [1].



Note: 1. "XX" is don't care. The letters refer to the programming description above.

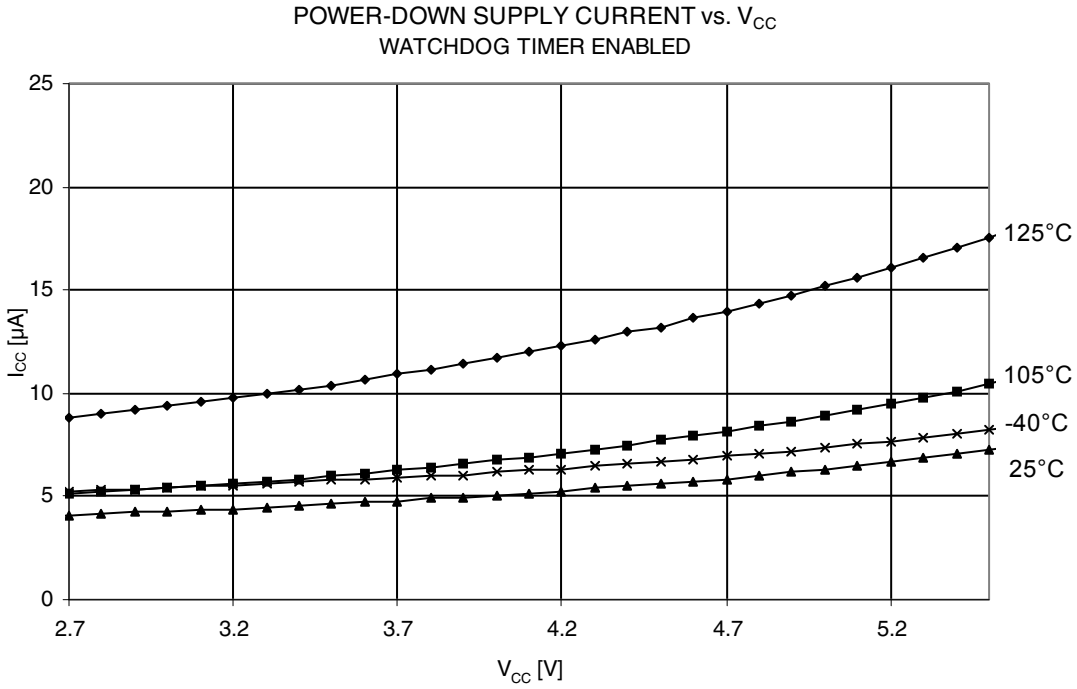
21.7.5 Programming the EEPROM

The EEPROM is organized in pages, see Table 21-13 on page 254. When programming the EEPROM, the program data is latched into a page buffer. This allows one page of data to be programmed simultaneously. The programming algorithm for the EEPROM data memory is as follows (refer to "Programming the Flash" on page 255 for details on Command, Address and Data loading):

1. A: Load Command "0001 0001".
2. G: Load Address High Byte (0x00 - 0xFF).
3. B: Load Address Low Byte (0x00 - 0xFF).
4. C: Load Data (0x00 - 0xFF).
5. E: Latch data (give PAGEL a positive pulse).

K: Repeat 3 through 5 until the entire buffer is filled

Figure 23-10. Power-down supply current vs. V_{CC} (watchdog timer enabled).



23.4 Pin Pull-up

Figure 23-11. I/O pin pull-up resistor current vs. input voltage (V_{CC} = 5V).

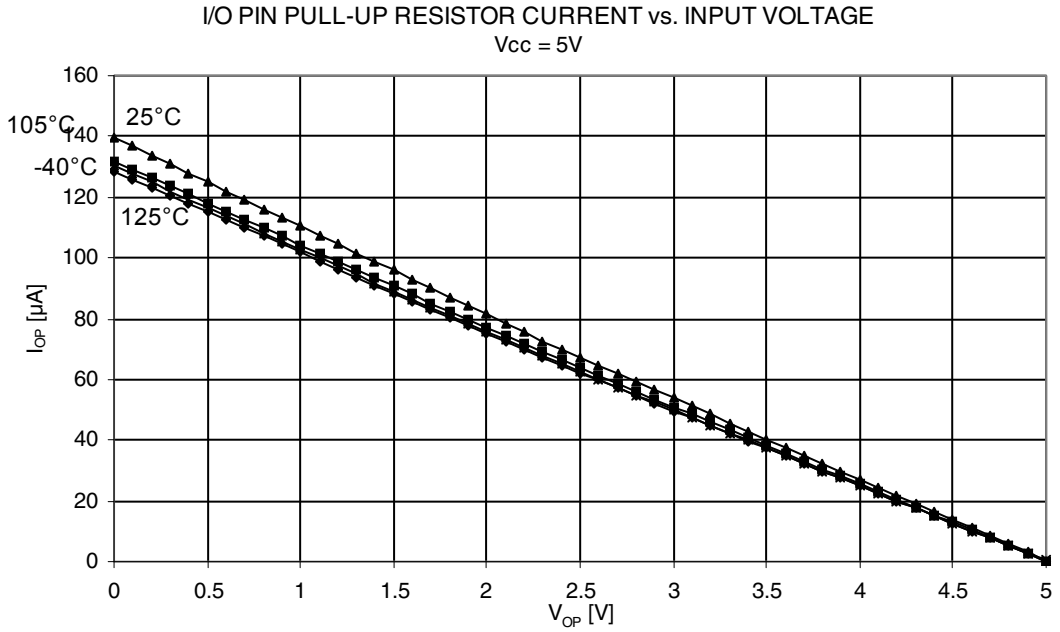


Table Of Contents