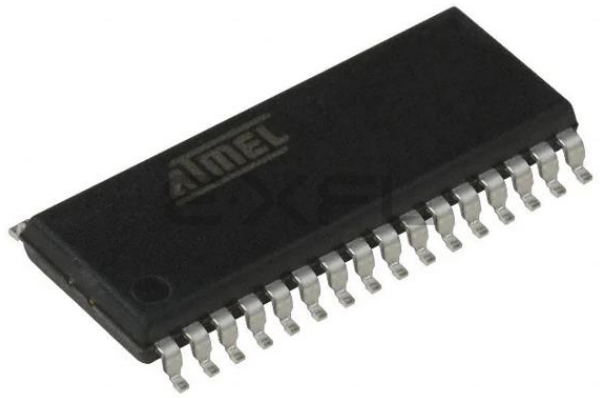


Welcome to E-XFL.COM

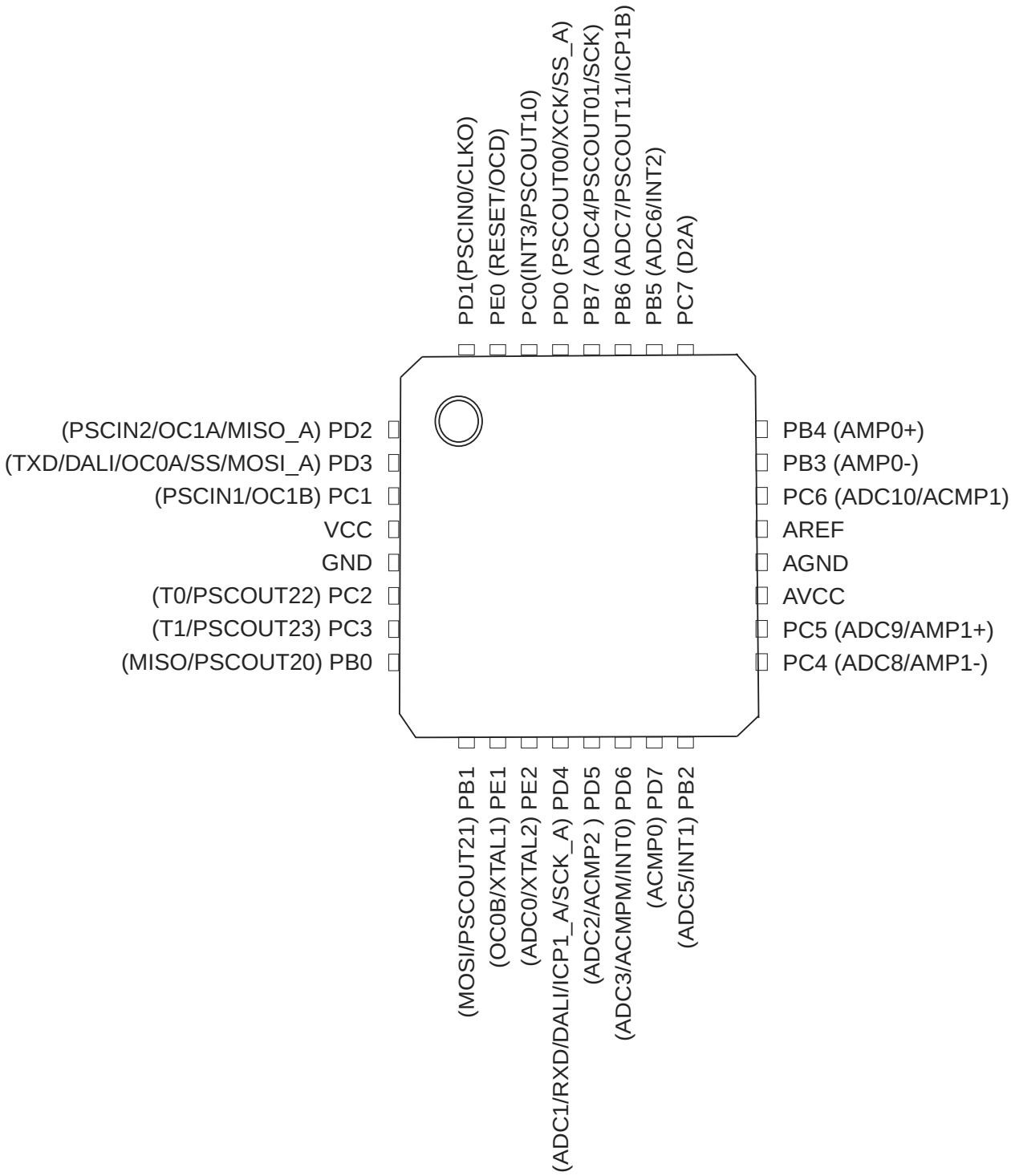
What is "Embedded - Microcontrollers"?

"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"

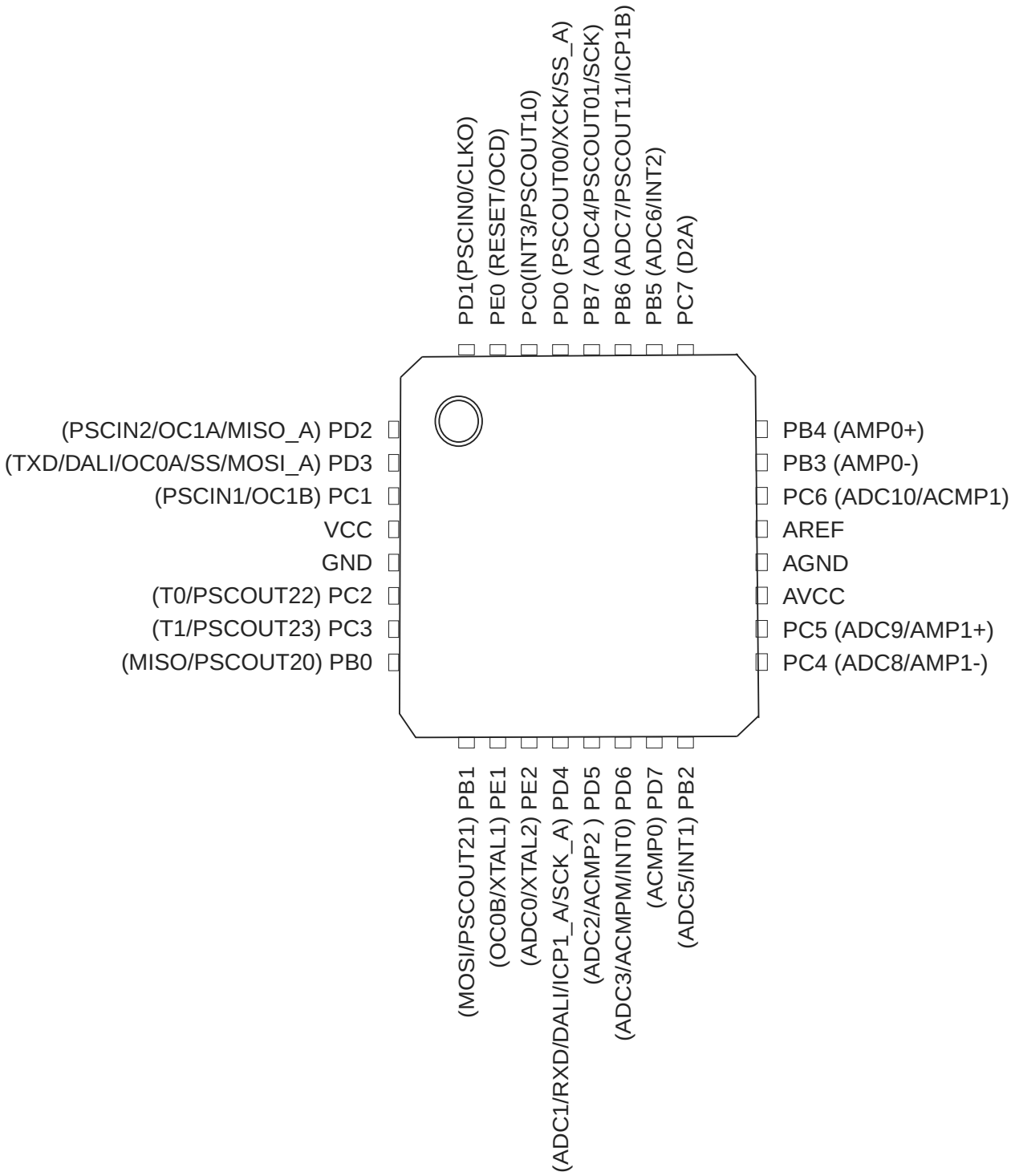
Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	27
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	1K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	32-SOIC (0.295", 7.50mm Width)
Supplier Device Package	32-SOIC
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at90pwm316-16su

Figure 2-3. QFN32 (7*7 mm) Package.



Note: The Center GND PADDLE has to be connected to GND.

Figure 2-3. QFN32 (7*7 mm) Package.



Note: The Center GND PADDLE has to be connected to GND.

2.1 Pin Descriptions

Table 2-1. Pin out description

S024 Pin Number	SO32 Pin Number	QFN32 Pin Number	Mnemonic	Type	Name, Function & Alternate Function
7	9	5	GND	Power	Ground: 0V reference
18	24	20	AGND	Power	Analog Ground: 0V reference for analog part
6	8	4	VCC	power	Power Supply:
17	23	19	AVCC	Power	Analog Power Supply: This is the power supply voltage for analog part For a normal use this pin must be connected.
19	25	21	AREF	Power	Analog Reference: reference for analog converter. This is the reference voltage of the A/D converter. As output, can be used by external analog
8	12	8	PBO	I/O	MISO (SPI Master In Slave Out) PSCOUT20 output
9	13	9	PB1	I/O	MOSI (SPI Master Out Slave In) PSCOUT21 output
16	20	16	PB2	I/O	ADC5 (Analog Input Channel5) INT1
20	27	23	PB3	I/O	AMP0- (Analog Differential Amplifier 0 Input Channel)
21	28	24	PB4	I/O	AMP0+ (Analog Differential Amplifier 0 Input Channel)
22	30	26	PB5	I/O	ADC6 (Analog Input Channel 6) INT 2
23	31	27	PB6	I/O	ADC7 (Analog Input Channel 7) ICP1B (Timer 1 input capture alternate input) PSCOUT11 output (see note 1)
24	32	28	PB7	I/O	PSCOUT01 output ADC4 (Analog Input Channel 4) SCK (SPI Clock)
NA	2	30	PC0	I/O	PSCOUT10 output (see note 1) INT3
	7	3	PC1	I/O	PSCIN1 (PSC 1 Digital Input) OC1B (Timer 1 Output Compare B)
	10	6	PC2	I/O	T0 (Timer 0 clock input) PSCOUT22 output
	11	7	PC3	I/O	T1 (Timer 1 clock input) PSCOUT23 output
	21	17	PC4	I/O	ADC8 (Analog Input Channel 8) AMP1- (Analog Differential Amplifier 1 Input Channel)
	22	18	PC5	I/O	ADC9 (Analog Input Channel 9) AMP1+ (Analog Differential Amplifier 1 Input Channel)
	26	22	PC6	I/O	ADC10 (Analog Input Channel 10) ACMP1 (Analog Comparator 1 Positive Input)
	29	25	PC7	I/O	D2A : DAC output(2)

Table 2-1. Pin out description (Continued)

S024 Pin Number	SO32 Pin Number	QFN32 Pin Number	Mnemonic	Type	Name, Function & Alternate Function
1	1	29	PD0	I/O	PSCOUT00 output(1) XCK (UART Transfer Clock) SS_A (Alternate SPI Slave Select)
3	4	32	PD1	I/O	PSCIN0 (PSC 0 Digital Input) CLKO (System Clock Output)
4	5	1	PD2	I/O	PSCIN2 (PSC 2 Digital Input) OC1A (Timer 1 Output Compare A) MISO_A (Programming & alternate SPI Master In Slave Out)
5	6	2	PD3	I/O	TXD (Dali/UART Tx data) OC0A (Timer 0 Output Compare A) SS (SPI Slave Select) MOSI_A (Programming & alternate Master Out SPI Slave In)
12	16	12	PD4	I/O	ADC1 (Analog Input Channel 1) RXD (Dali/UART Rx data) ICP1A (Timer 1 input capture) SCK_A (Programming & alternate SPI Clock)
13	17	13	PD5	I/O	ADC2 (Analog Input Channel 2) ACMP2 (Analog Comparator 2 Positive Input)
14	18	14	PD6	I/O	ADC3 (Analog Input Channel 3) ACMPM reference for analog comparators INT0
15	19	15	PD7	I/O	ACMP0 (Analog Comparator 0 Positive Input)
2	3	31	PE0	I/O or I	RESET (Reset Input) OCD (On Chip Debug I/O)
10	14	10	PE1	I/O	XTAL1: XTAL Input OC0B (Timer 0 Output Compare B)
11	15	11	PE2	I/O	XTAL2: XTAL OuTput ADC0 (Analog Input Channel 0)

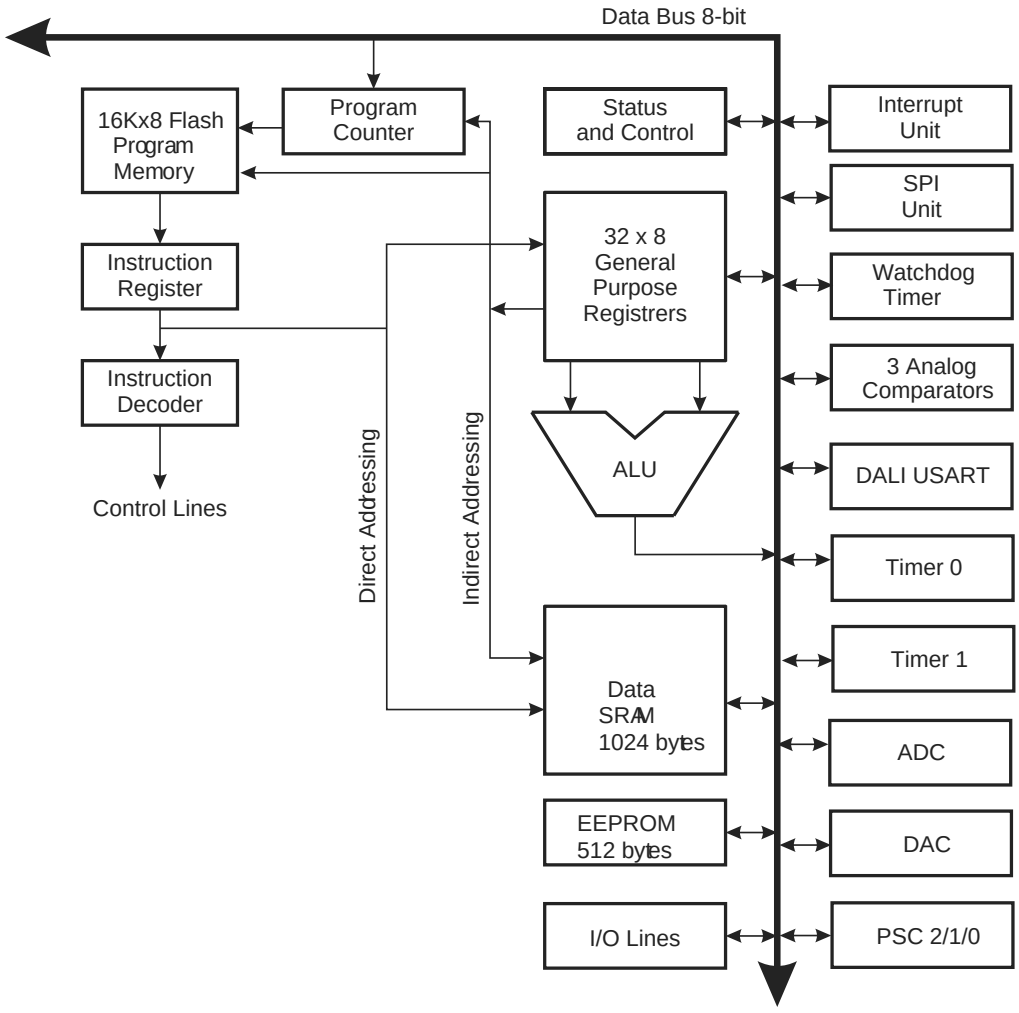
Notes: 1. PSCOUT10 & PSCOUT11 are not present on 24 pins package
2. D2A (DAC Output) not available on AT90PWM261 (SOIC 24-pins)

3. Overview

The AT90PWM216/316 are low-power CMOS 8-bit microcontrollers based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90PWM216/316 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

3.1 Block Diagram

Figure 3-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The AT90PWM216/316 provides the following features: 16K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 1024 bytes SRAM, 53 general purpose I/O lines, 32 general purpose working registers, three Power Stage Controllers, two flexible Timer/Counters with compare modes and PWM, one USART with DALI mode, an 11-channel 10-bit ADC with two differential input stage with programmable gain, a 10-bit DAC, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, an On-chip Debug system and four software selectable power saving modes.

The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI ports and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. The ADC Noise Reduction mode stops the CPU and all I/O modules except ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Res-

3.2.6 Port E (PE2..0) $\overline{\text{RESET}}$/ XTAL1/ XTAL2

Port E is an 3-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

If the RSTDISBL Fuse is programmed, PE0 is used as an I/O pin. Note that the electrical characteristics of PE0 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PE0 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 8-1 on page 41. Shorter pulses are not guaranteed to generate a Reset.

Depending on the clock selection fuse settings, PE1 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PE2 can be used as output from the inverting Oscillator amplifier.

The various special features of Port E are elaborated in Alternate Functions of Port E on page 71 and Clock Systems and their Distribution on page 25.

3.2.7 AVCC

AVCC is the supply voltage pin for the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

3.2.8 AREF

This is the analog reference pin for the A/D Converter.

3.3 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.

8. Errata AT90PWM216/316

8.1 Revision C

DAC Driver linearity above 3.6V

1. DAC Driver linearity above 3.6V
   With 5V V_{CC}, the DAC driver linearity is poor when DAC output level is above V_{CC}-1V. At 5V, DAC output for 1023 will be around 5V - 40mV.

   Work around:
   Use, when Vcc=5V, V_{REF} below V_{CC}-1V

   Or, when V_{REF}=V_{CC}=5V, do not uses codes above 800.

8.2 Revision B

DAC Driver linearity above 3.6V

PSC OCRxx Register update according to PLOCK2 usage

1. DAC Driver linearity above 3.6V
   With 5V V_{CC}, the DAC driver linearity is poor when DAC output level is above V_{CC}-1V. At 5V, DAC output for 1023 will be around 5V - 40mV.

   Work around:
   Use, when Vcc=5V, V_{REF} below V_{CC}-1V

   Or, when V_{REF}=V_{CC}=5V, do not uses codes above 800.

2. PSC OCRxx Register update according to PLOCK2 usage
   If the PSC is clocked from PLL, and if PLOCK2 bit is changed at the same time as PSC end of cycle occurs, and if OCRxx registers contents have been changed, then the updated OCRxx registers contents are not predictable.

   The cause is a synchronization issue between two registers in two different clock domains (PLL clock which clocks PSC and CPU clock).

   Workaround:
   Enable the PSC end of cycle interrupt.

   At the beginning of PSC EOC interrupt vector, change PLOCK value (OCRxx registers can be updated outside the interrupt vector).

   This process guarantees that UPDATE and PLOCK actions will not occur at the same moment.

8.3 Revision A

DAC Driver linearity above 3.6V

PSC OCRxx Register update according to PLOCK2 usage

1. DAC Driver linearity above 3.6V
   With 5V V_{CC}, the DAC driver linearity is poor when DAC output level is above V_{CC}-1V. At 5V, DAC output for 1023 will be around 5V - 40mV.

   Work around:
   Use, when Vcc=5V, V_{REF} below V_{CC}-1V

   Or, when V_{REF}=V_{CC}=5V, do not uses codes above 800.

2. PSC OCRxx Register update according to PLOCK2 usage
   If the PSC is clocked from PLL, and if PLOCK2 bit is changed at the same time as PSC end of cycle occurs, and if OCRxx registers contents have been changed, then the updated OCRxx registers contents are not predictable.

   The cause is a synchronization issue between two registers in two different clock domains (PLL clock which clocks PSC and CPU clock).

   Workaround:
   Enable the PSC end of cycle interrupt.

   At the beginning of PSC EOC interrupt vector, change PLOCK value (OCRxx registers can be updated outside the interrupt vector).

   This process guarantees that UPDATE and PLOCK actions will not occur at the same moment.