ชื่อ-สกุล..เลขประจำตัว................................section..........

มหาวิทยาลัยสงขลานครินทร์

คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 2 **ปีการศึกษา**: 2555

วันที่สอบ: 20 กุมภาพันธ์ 2556 **เวลาสอบ**: 13.30-16.30 น.

รหัสวิชา: 241-210 **ห้องสอบ**: หัวหุ่นยนต์

ชื่อวิชา: Microprocessor Architectures and the Assembly Language

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

เวลา: 3 ชั่วโมง (180 นาที)

รายละเอียดของข้อสอบ: ข้อสอบมีทั้งหมด 8 หน้า และ Datasheet อีก 10 หน้า รวม 18 หน้า

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ และเครื่องคิดเลข

ไม่อนุญาต: เครื่องคอมพิวเตอร์โน้ตบุ๊ค แทบเล็ต สมุดจด หนังสือ กระดาษโน้ต และเอกสารใดๆ

คำสั่ง:

- **ให้ทำทุกข้อ** คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ
- ห้ามหยิบยืมสิ่งใดๆ ทั้งสิ้นจากผู้อื่น เว้นแต่ผู้คุมสอบจะหยิบยืมให้
- ห้ามนำส่วนหนึ่งส่วนใดของข้อสอบออกจากห้องสอบ
- ผู้ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ **แต่ต้องไม่น้อยกว่า 30 นาที** ให้ยกมือขออนุญาตจากผู้คุมสอบก่อนจะลุกจากที่นั่ง
- เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
- เขียนชื่อ, รหัสและหมายเลข Section ให้ชัดเจนในข้อสอบ **ทุกแผ่น** แผ่นใดไม่เขียนหรือเขียนไม่ครบจะถูกตัดคะแนนแผ่นละ 1 คะแนน
- อ่านคำสั่งเพิ่มเติมในแต่ละข้อให้ชัดเจน คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด

ทุจริตในการสอบ โทษขั้นต่ำคือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

ผู้ออกข้อสอบ

....

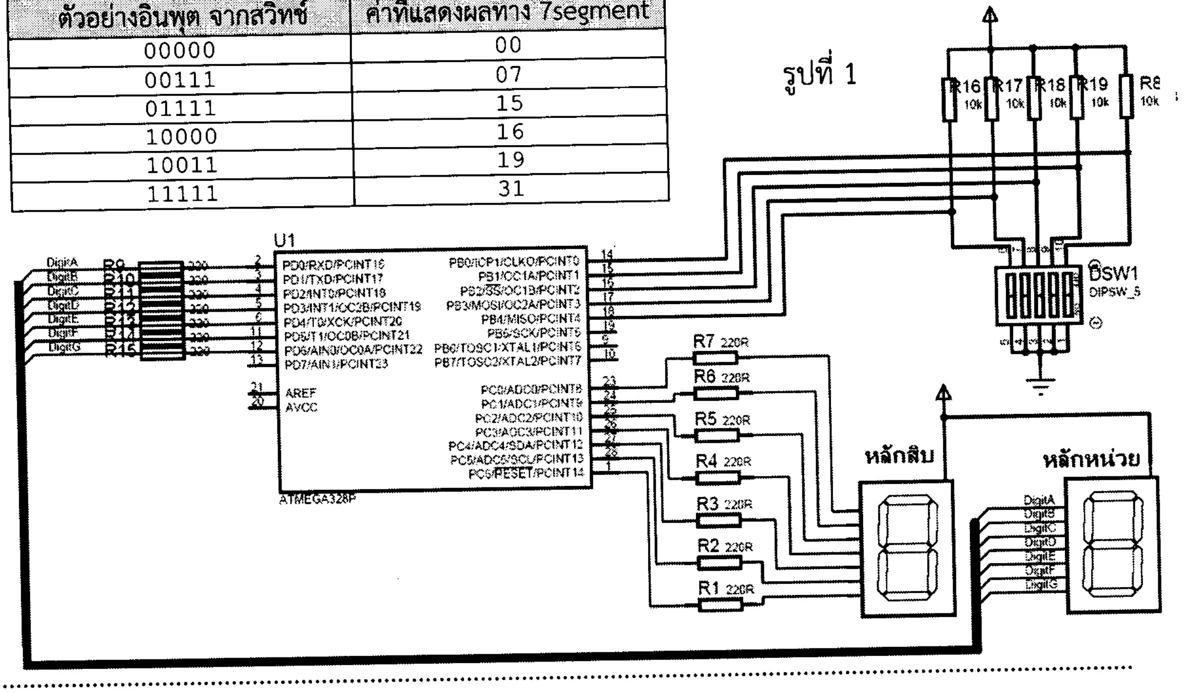
ดร. ปัญญยศ ไชยกาฬ

ชื่อ-สกุล...เลขประจำตัว.................................section..........

1. จากวงจรดังรูปที่ 1 จงเขียนโปรแกรมควบคุมซีพียู AVR เพื่อทำการอ่านค่าจากดิพสวิทช์จำนวน 5 บิตมาแปลงเป็นเลขฐานสิบแบบไม่มีเครื่องหมาย แล้วทำการแสดงผลยังแอลอีดี 7 เซกเมนต์ จำนวน 2 ตัว กำหนดให้ซีพียูทำการตรวจสอบสถานะของสวิทช์ด้วย Pin-change interrupt (ซีพียูไม่ต้องคอยวนลูปอ่านสถานะของสวิทช์) (10 คะแนน)

ตัวอย่างอินพุต จากสวิทช์	ค่าที่แสดงผลทาง 7segment
00000	00
00111	07
01111	15
10000	16
10011	19
11111	31

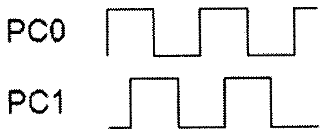
รูปที่ 1



```
#include <avr/io.h>
#include <avr/interrupt.h>

unsigned char LOOKUPTB[] = {  0b00111111, 0b00000110,
                              0b01011011, 0b01001111,
                              0b01100110, 0b01101101,
                              0b01111101, 0b00000111,
                              0b01111111, 0b01101111,
                              0b01110111, 0b01111100,
                              0b00111001, 0b01011110,
                              0b01111001, 0b01110001 };

void do_nothing(void){}
```

ชื่อ-สกุล...เลขประจำตัว...............................section..........

2. จงเขียนโปรแกรมซีพียู AVR เพื่อส่งพัลซ์ความถี่ 60 Hz ออกทางขา PC0 และ PC1 โดยมีเฟสต่างกัน 90 องศา ดังรูปที่ 2 โดยกำหนดให้ความถี่เอาต์พุตผิดพลาดได้ไม่เกิน 1 เปอร์เซ็นต์ กำหนดให้ซีพียูทำงานที่ความถี่ 12 MHz (10 คะแนน)



รูปที่ 2

3. จงเขียนโปรแกรมให้ซีพียู AVR ความเร็ว 12 MHz ติดต่อกับเครื่อง PC ผ่านทางพอร์ตอนุกรม โดยซีพียู AVR นำข้อมูลที่รับได้จาก PC มาตรวจสอบหากพบว่าเป็นเลข 0-9 ให้แสดงผลออกทาง 7-segment LED ชนิด Common Anode ที่ต่ออยู่กับพอร์ต C แต่ถ้าอินพุตจากเครื่อง PC ส่งมาเป็นค่าอื่น ให้แสดงผล 'E' ออกทาง 7-segment กำหนดให้ทำการติดต่อที่ 9600 bps, 8-bit data, Odd parity, 1 stop bits (15 คะแนน)

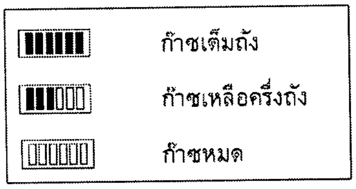
```
#include <avr/io.h>
#include <avr/interrupt.h>

unsigned char LOOKUPTB[] = {  0b00111111, 0b00000110,
                              0b01011011, 0b01001111,
                              0b01100110, 0b01101101,
                              0b01111101, 0b00000111,
                              0b01111111, 0b01101111,
                              0b01110111, 0b01111100,
                              0b00111001, 0b01011110,
                              0b01111001, 0b01110001 };

void do_nothing(void){}
```

ชื่อ-สกุล..เลขประจำตัว...............................section..........

4. สมมุติให้เซนเซอร์ก๊าซ NGV มีค่าแรงดันเอาต์พุตดังตารางที่ 1 หากเราต้องการนำเซนเซอร์ดังกล่าวมาต่อกับซีพียู AVR เพื่อตรวจจับระดับก๊าซในถังและแสดงปริมาณของก๊าซในถังออกทางหลอด LED จำนวน 6 หลอด ดังรูปที่ 3 จงออกแบบวงจรพร้อมทั้งเขียนโปรแกรมควบคุมซีพียู AVR (15 คะแนน)

ปริมาณก๊าซในถัง (%)	ค่าแรงดันเอาต์พุต(V)
100	2.500
75	1.875
50	1.250
25	0.625
0	0.000

ตารางที่ 1



รูปที่ 3

<u>คำแนะนำ</u> ให้ใช้วงจร ADC ภายใน AVR ให้เป็นประโยชน์

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

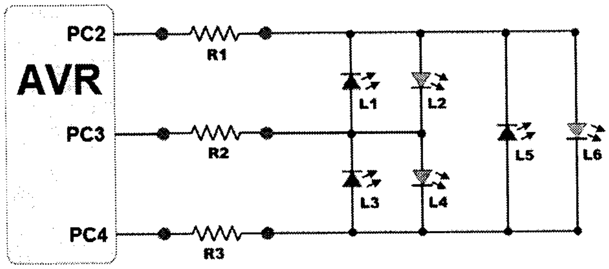
....

....

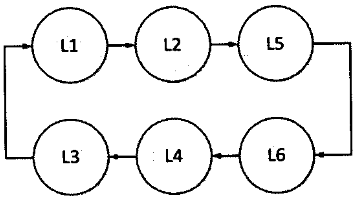
....

ชื่อ-สกุล...เลขประจำตัว.................................section..........

5. จากรูปที่ 4 จงเขียนโปรแกรมภาษาซีเพื่อให้ AVR ทำการสั่งให้แอลอีดีติดสลับกันดังรูปที่ 5 (10 คะแนน)



รูปที่4



รูปที่ 5

----End of Examination----

datasheet page 1

EICRA-External Interrupt Control Register A

The External Interrupt Control Register A contains control bits for interrupt sense control.

Bit	7	6	5	4	3	2	1	0	
(0x69)	–	–	–	–	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:4 – Reserved**

These bits are unused bits in the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P, and will always read as zero.

- **Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0**

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 12-1. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- **Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 12-2. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 12-2. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
0x1D (0x3D)	–	–	–	–	–	–	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Reserved**

These bits are unused bits in the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P, and will always read as zero.

- **Bit 1 – INT1: External Interrupt Request 1 Enable**

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from the INT1 Interrupt Vector.

- **Bit 0 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the External Interrupt Control Register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 Interrupt Vector.

ค่า Interrupt Vector ของ ATMEGA328P

```
/* Interrupt Vectors */
/* Interrupt Vector 0 is the reset vector. */
#define INT0_vect            _VECTOR(1)   /* External Interrupt Request 0 */
#define INT1_vect            _VECTOR(2)   /* External Interrupt Request 1 */
#define PCINT0_vect          _VECTOR(3)   /* Pin Change Interrupt Request 0 */
#define PCINT1_vect          _VECTOR(4)   /* Pin Change Interrupt Request 0 */
#define PCINT2_vect          _VECTOR(5)   /* Pin Change Interrupt Request 1 */
#define WDT_vect             _VECTOR(6)   /* Watchdog Time-out Interrupt */
#define TIMER2_COMPA_vect    _VECTOR(7)   /* Timer/Counter2 Compare Match A */
#define TIMER2_COMPB_vect    _VECTOR(8)   /* Timer/Counter2 Compare Match A */
#define TIMER2_OVF_vect      _VECTOR(9)   /* Timer/Counter2 Overflow */
#define TIMER1_CAPT_vect     _VECTOR(10)  /* Timer/Counter1 Capture Event */
#define TIMER1_COMPA_vect    _VECTOR(11)  /* Timer/Counter1 Compare Match A */
#define TIMER1_COMPB_vect    _VECTOR(12)  /* Timer/Counter1 Compare Match B */
#define TIMER1_OVF_vect      _VECTOR(13)  /* Timer/Counter1 Overflow */
#define TIMER0_COMPA_vect    _VECTOR(14)  /* TimerCounter0 Compare Match A */
#define TIMER0_COMPB_vect    _VECTOR(15)  /* TimerCounter0 Compare Match B */
#define TIMER0_OVF_vect      _VECTOR(16)  /* Timer/Couner0 Overflow */
#define SPI_STC_vect         _VECTOR(17)  /* SPI Serial Transfer Complete */
#define USART_RX_vect        _VECTOR(18)  /* USART Rx Complete */
#define USART_UDRE_vect      _VECTOR(19)  /* USART, Data Register Empty */
#define USART_TX_vect        _VECTOR(20)  /* USART Tx Complete */
#define ADC_vect             _VECTOR(21)  /* ADC Conversion Complete */
#define EE_READY_vect        _VECTOR(22)  /* EEPROM Ready */
#define ANALOG_COMP_vect     _VECTOR(23)  /* Analog Comparator */
#define TWI_vect             _VECTOR(24)  /* Two-wire Serial Interface */
#define SPM_READY_vect       _VECTOR(25)  /* Store Program Memory Read */
```

datasheet page 2

PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	–	–	–	–	–	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Reserved**

These bits are unused bits in the ATmega48A/48PA/88A/88PA/168A/168PA/328/328P, and will always read as zero.

- **Bit 2 – PCIE2: Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT[23:16] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT[23:16] pins are enabled individually by the PCMSK2 Register.

- **Bit 1 – PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT[14:8] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT[14:8] pins are enabled individually by the PCMSK1 Register.

- **Bit 0 – PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT[7:0] pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT[7:0] pins are enabled individually by the PCMSK0 Register.

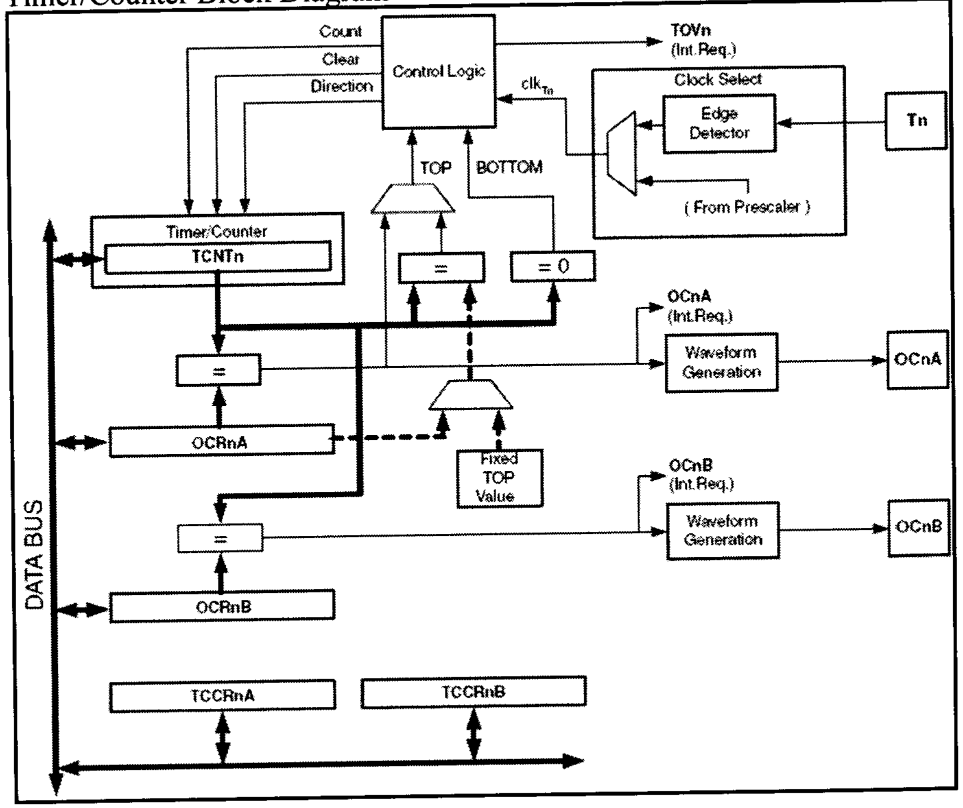
PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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	

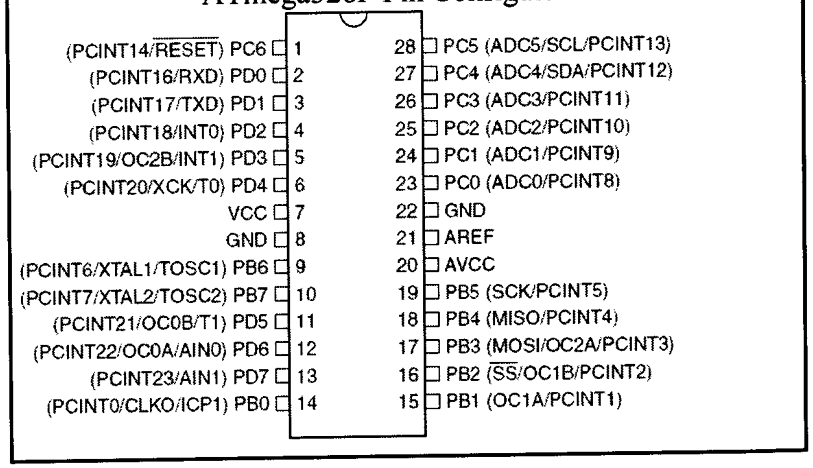
- **Bit 7:0 – PCINT[7:0]: Pin Change Enable Mask 7...0**

Each PCINT[7:0] bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT[7:0] is cleared, pin change interrupt on the corresponding I/O pin is disabled.

Timer/Counter Block Diagram

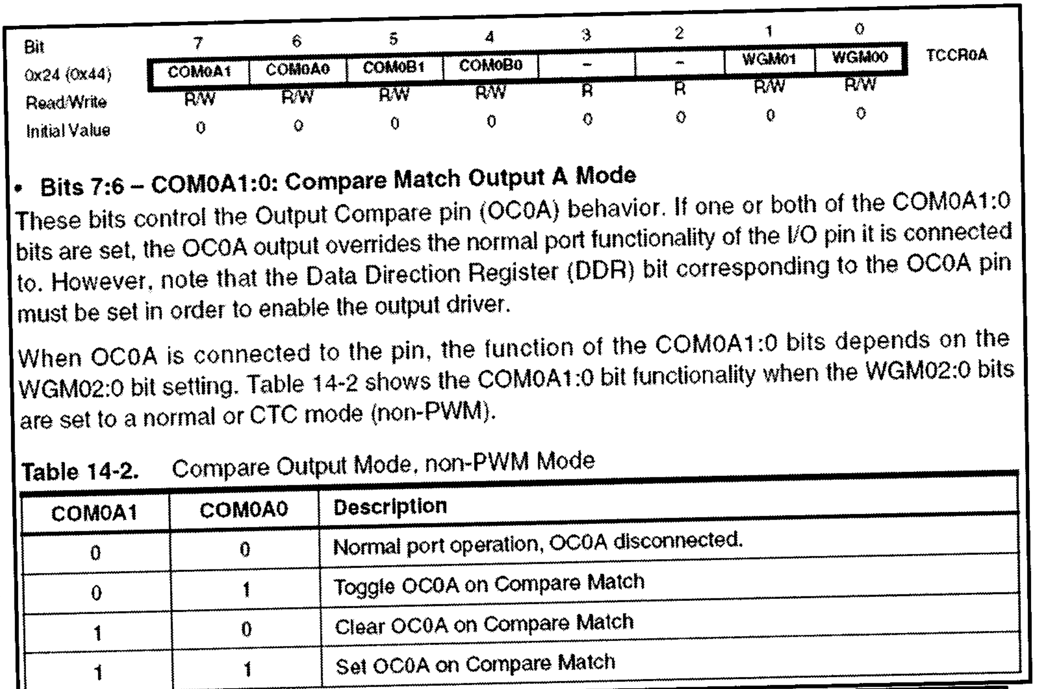


ATmega328P Pin Configuration



datasheet page 3

TCCR0A – Timer/Counter Control Register A



Bit	7	6	5	4	3	2	1	0	
0x24 (0x44)	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	TCCR0A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 – COM0A1:0: Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver.

When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. Table 14-2 shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 14-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

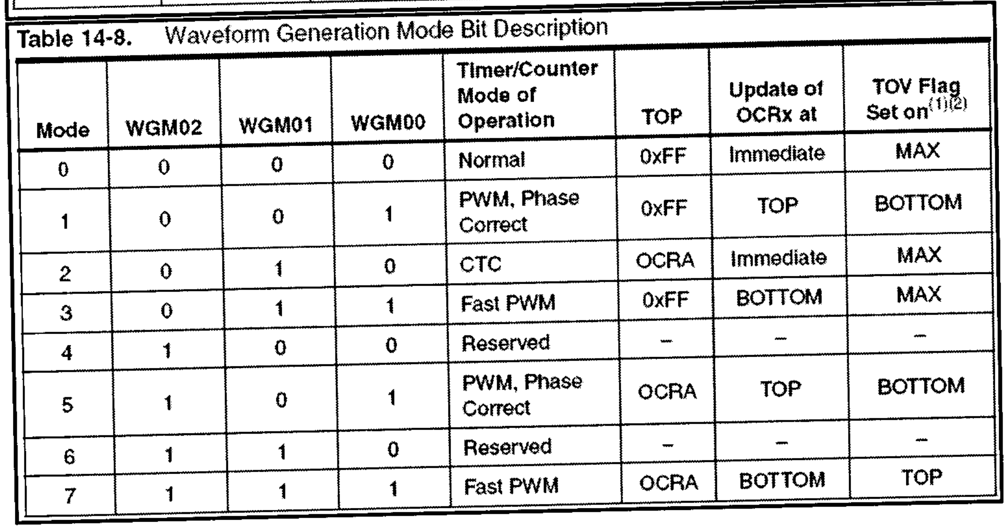
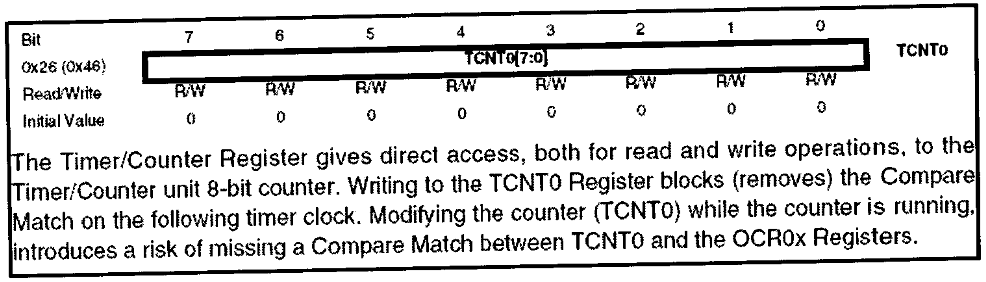


Table 14-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[(1)(2)]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

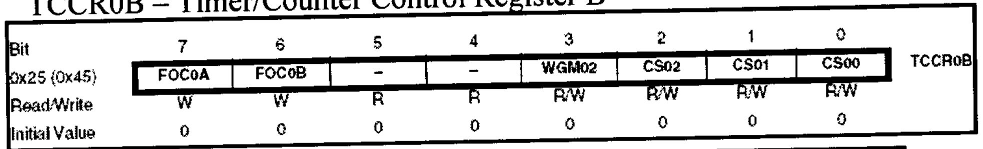
TCNT0



Bit	7	6	5	4	3	2	1	0	
0x26 (0x46)	TCNT0[7:0]								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 Registers.

TCCR0B – Timer/Counter Control Register B



Bit	7	6	5	4	3	2	1	0	
0x25 (0x45)	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

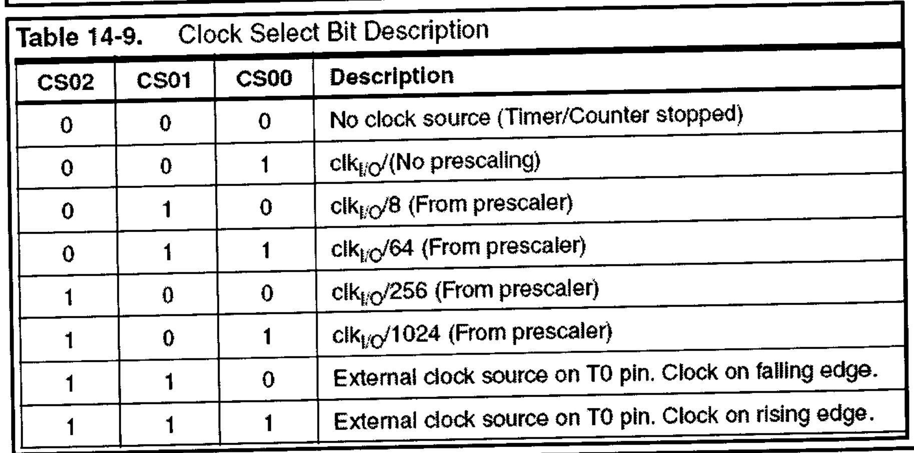


Table 14-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk$_{I/O}$/(No prescaling)
0	1	0	clk$_{I/O}$/8 (From prescaler)
0	1	1	clk$_{I/O}$/64 (From prescaler)
1	0	0	clk$_{I/O}$/256 (From prescaler)
1	0	1	clk$_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

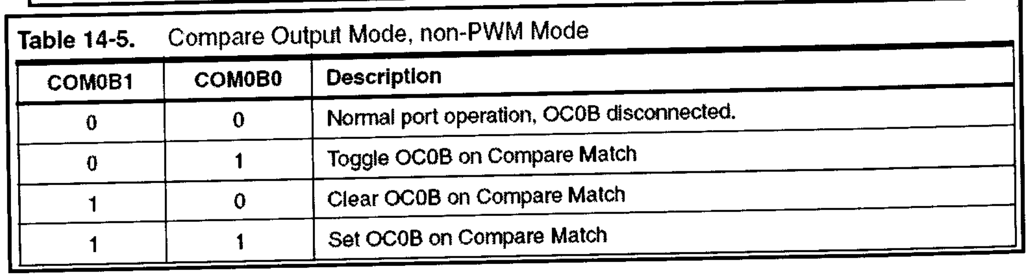


Table 14-5. Compare Output Mode, non-PWM Mode

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Toggle OC0B on Compare Match
1	0	Clear OC0B on Compare Match
1	1	Set OC0B on Compare Match

datasheet page 4

OCR0A – Output Compare Register A

Bit	7	6	5	4	3	2	1	0	
0x27 (0x47)	OCR0A[7:0]								OCR0A
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 Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0A pin.

OCR0B – Output Compare Register B

Bit	7	6	5	4	3	2	1	0	
0x28 (0x48)	OCR0B[7:0]								OCR0B
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 Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0B pin.

TIMSK0 – Timer/Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6E)	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter Interrupt Flag Register – TIFR0.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter 0 Interrupt Flag Register – TIFR0.

การคำนวณหาค่าเริ่มต้นของ Timer0

กรณีทำงานใน Normal Mode

◆กำหนดให้

- ◆TCNT0 = ค่าที่จะต้องเซ็ตเป็นค่าเริ่มต้นให้กับ Timer0
- ◆N = Prescale factor (1, 8, 64, 256, 1024)
- ◆IP = Interrupt Period คาบเวลาที่จะให้ซีพียูเกิดอินเตอร์รัพต์ (หน่วยเป็นวินาที)
- ◆CPUclk = ความถี่สัญญาณนาฬิกาที่ซีพียูทำงาน

$$TCNT0 = 256 - \frac{CPUclk * IP}{N}$$

การคำนวณหาค่าเริ่มต้นของ Timer0

กรณีทำงานใน CTC Mode

◆ กำหนดให้

- ◆N = Prescale factor (1, 8, 64, 256, 1024)
- ◆F_{OC0A} = ความถี่ที่ออกจากขาเอาต์พุต OCR0A
- ◆Fclk_io = ความถี่สัญญาณนาฬิกาอ้างอิงของ Timer

$$F_{OC0A} = \frac{Fclk\_io}{2 * N * (1 + OCR0A)}$$

$$OCR0A = \frac{Fclk\_io}{2 * N * F_{OC0A}} - 1$$

```
unsigned char TB7segment_com_cathode[] =
{       0b00111111,     //0
        0b00000110,     //1
        0b01011011,     //2
        0b01001111,     //3
        0b01100110,     //4
        0b01101101,     //5
        0b01111101,     //6
        0b00000111,     //7
        0b01111111,     //8
        0b01101111,     //9
        0b01110111,     //A
        0b01111100,     //b
        0b00111001,     //C
        0b01011110,     //d
        0b01111001,     //E
        0b01110001,     //F
        0b01000000,     //-
        0b00000000     //all segments off
 };
```

datasheet page 5

TCCR1A – Timer/Counter Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM1A1:0: Compare Output Mode for Channel A**
- **Bit 5:4 – COM1B1:0: Compare Output Mode for Channel B**

Table 15-1. Compare Output Mode, non-PWM

COM1A1/COM1B1	COM1A0/COM1B0	Description
0	0	Normal port operation, OC1A/OC1B disconnected.
0	1	Toggle OC1A/OC1B on Compare Match.
1	0	Clear OC1A/OC1B on Compare Match (Set output to low level).
1	1	Set OC1A/OC1B on Compare Match (Set output to high level).

Table 15-4. Waveform Generation Mode Bit Description[1]

Mode	WGM13	WGM12 (CTC1)	WGM11 (PWM11)	WGM10 (PWM10)	Timer/Counter Mode of Operation	TOP	Update of OCR1x at	TOV1 Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR1A	Immediate	MAX

TCNT1H and TCNT1L – Timer/Counter1

Bit	7	6	5	4	3	2	1	0	
(0x85)	TCNT1[15:8]								TCNT1H
(0x84)	TCNT1[7:0]								TCNT1L
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	

OCR1AH and OCR1AL – Output Compare Register 1 A

Bit	7	6	5	4	3	2	1	0	
(0x89)	OCR1A[15:8]								OCR1AH
(0x88)	OCR1A[7:0]								OCR1AL
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	

TCCR1B-Timer/Counter Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 15-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Timer1 : CTC mode

- กำหนดให้
  - ค่า 16 บิต ที่จะต้องเซ็ตเป็นค่าเริ่มต้นให้กับรีจิสเตอร์ OCR1A
  - N = Prescale factor (1, 8, 64, 256, 1024)
  - f_{OC1A} ความถี่ที่สามารถสร้างขึ้นจากการ Toggle ขาเอาต์พุต OC1A
  - $f_{clk\_I/O}$ = ความถี่สัญญาณนาฬิกาที่ซีพียูทำงาน

$$f_{OC1A} = \frac{f_{clk\ I/O}}{2 \cdot N \cdot (1 + OCR1A)}$$

OCR1BH and OCR1BL – Output Compare Register 1 B

Bit	7	6	5	4	3	2	1	0	
(0x8B)	OCR1B[15:8]								OCR1BH
(0x8A)	OCR1B[7:0]								OCR1BL
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	

datasheet page 6

Timer1 : Normal mode

- ◆กำหนดให้
  - ◆TCNT1 = ค่าที่จะต้องเซ็ตเป็นค่าเริ่มต้นให้กับ Timer1
  - ◆N = Prescale factor (1, 8, 64, 256, 1024)
  - ◆IP = Interrupt Period คาบเวลาที่จะให้ซีพียูเกิดอินเตอร์รัพต์ (หน่วยเป็นวินาที)
  - ◆CPUclk = ความถี่สัญญาณนาฬิกาที่ซีพียูทำงาน

$$TCNT1 = 65536 - \frac{CPUclk * IP}{N}$$

คาบเวลายาวที่สุดที่ซีพียูจับได้

◆กรณีใช้ Timer1 ที่ normal mode

$$IP = (65536 - 0_{TCNT1}) * \frac{1024_N}{CPUclk}$$

◆กรณีใช้ Timer0 ที่ normal mode

$$IP = (256 - 0_{TCNT0}) * \frac{1024_N}{CPUclk}$$

ค่าเวลาสั้นที่สุดที่ซีพียูจับได้

◆กรณีใช้ Timer1 ที่ normal mode

$$IP = (65536 - 65535_{TCNT1}) * \frac{1}{CPUclk}$$

◆กรณีใช้ Timer0 ที่ normal mode

$$IP = (256 - 255_{TCNT0}) * \frac{1}{CPUclk}$$

TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x6F)	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 2 – OCIE1B: Timer/Counter1, Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare B Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 58) is executed when the OCF1B Flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Timer/Counter1, Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Output Compare A Match interrupt is enabled. The corresponding Interrupt Vector (see "Interrupts" on page 58) is executed when the OCF1A Flag, located in TIFR1, is set.

- **Bit 0 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 58) is executed when the TOV1 Flag, located in TIFR1, is set.

SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

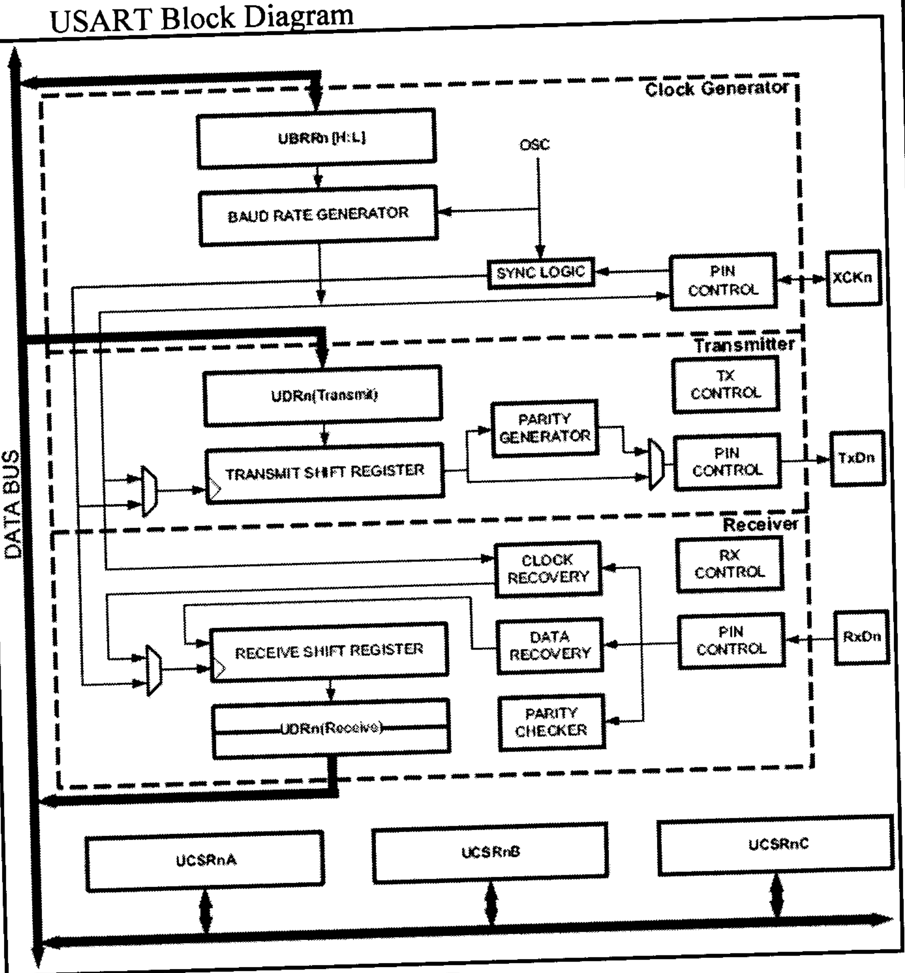
Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

datasheet page 7

USART Block Diagram

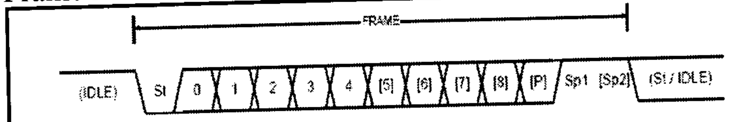


Baud Rate (bps)	f_{osc} = 8.0000 MHz			
	U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%
4800	103	0.2%	207	0.2%
9600	51	0.2%	103	0.2%
14.4k	34	-0.8%	68	0.6%
19.2k	25	0.2%	51	0.2%

Table 19-1. Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate[1]	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$BAUD = \frac{f_{OSC}}{16(UBRRn+1)}$	$UBRRn = \frac{f_{OSC}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{OSC}}{8(UBRRn+1)}$	$UBRRn = \frac{f_{OSC}}{8BAUD} - 1$

Frame Format



St	Start bit, always low.
(n)	Data bits (0 to 8).
P	Parity bit. Can be odd or even.
Sp	Stop bit, always high.
IDLE	No transfers on the communication line (RxDn or TxDn). An IDLE line must be high.

UCSRnA – USART Control and Status Register n A

Bit	7	6	5	4	3	2	1	0	
	RXCn	TXCn	UDREn	FEn	DORn	UPEn	U2Xn	MPCMn	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

- **Bit 7 – RXCn: USART Receive Complete**

This flag bit is set when there are unread data in the receive buffer and cleared when the receive buffer is empty (i.e.. does not contain any unread data). If the Receiver is disabled, the receive buffer will be flushed and consequently the RXCn bit will become zero. The RXCn Flag can be used to generate a Receive Complete interrupt (see description of the RXCIEn bit).

- **Bit 6 – TXCn: USART Transmit Complete**

This flag bit is set when the entire frame in the Transmit Shift Register has been shifted out and there are no new data currently present in the transmit buffer (UDRn). The TXCn Flag bit is automatically cleared when a transmit complete interrupt is executed, or it can be cleared by writing a one to its bit location. The TXCn Flag can generate a Transmit Complete interrupt (see description of the TXCIEn bit).

- **Bit 5 – UDREn: USART Data Register Empty**

The UDREn Flag indicates if the transmit buffer (UDRn) is ready to receive new data. If UDREn is one, the buffer is empty, and therefore ready to be written. The UDREn Flag can generate a Data Register Empty interrupt (see description of the UDRIEn bit). UDREn is set after a reset to indicate that the Transmitter is ready.

datasheet page 8

UCSRnB – USART Control and Status Register n B

Bit	7	6	5	4	3	2	1	0	
	RXCIEn	TXCIEn	UDRIEn	RXENn	TXENn	UCSZn2	RXB8n	TXB8n	UCSRnB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – RXCIEn: RX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the RXCn Flag. A USART Receive Complete interrupt will be generated only if the RXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the RXCn bit in UCSRnA is set.

- **Bit 6 – TXCIEn: TX Complete Interrupt Enable n**

Writing this bit to one enables interrupt on the TXCn Flag. A USART Transmit Complete interrupt will be generated only if the TXCIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the TXCn bit in UCSRnA is set.

- **Bit 5 – UDRIEn: USART Data Register Empty Interrupt Enable n**

Writing this bit to one enables interrupt on the UDREn Flag. A Data Register Empty interrupt will be generated only if the UDRIEn bit is written to one, the Global Interrupt Flag in SREG is written to one and the UDREn bit in UCSRnA is set.

- **Bit 4 – RXENn: Receiver Enable n**

Writing this bit to one enables the USART Receiver. The Receiver will override normal port operation for the RxDn pin when enabled. Disabling the Receiver will flush the receive buffer invalidating the FEn, DORn, and UPEn Flags.

- **Bit 3 – TXENn: Transmitter Enable n**

Writing this bit to one enables the USART Transmitter. The Transmitter will override normal port operation for the TxDn pin when enabled. The disabling of the Transmitter (writing TXENn to zero) will not become effective until ongoing and pending transmissions are completed, i.e., when the Transmit Shift Register and Transmit Buffer Register do not contain data to be transmitted. When disabled, the Transmitter will no longer override the TxDn port.

- **Bit 2 – UCSZn2: Character Size n**

The UCSZn2 bits combined with the UCSZn1:0 bit in UCSRnC sets the number of data bits (Character SiZe) in a frame the Receiver and Transmitter use.

UBRRnL and UBRRnH – USART Baud Rate Registers

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	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	
	0	0	0	0	0	0	0	0	

UCSRnC – USART Control and Status Register n C

Bit	7	6	5	4	3	2	1	0	
	UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC
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	1	1	0	

Table 19-7. UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM)[1]

Table 19-8. UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

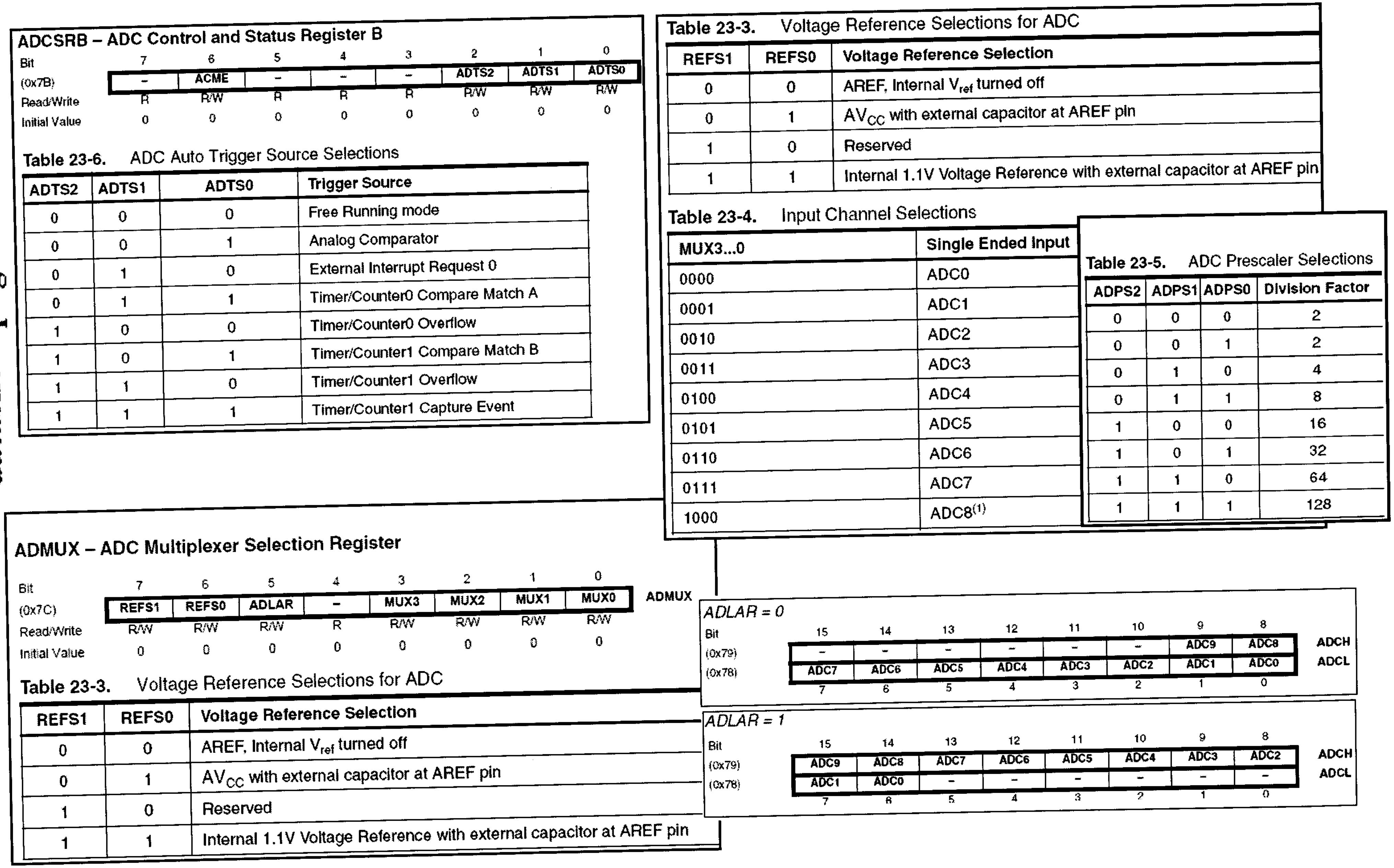
Table 19-9. USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

Table 19-10. UCSZn Bits Settings

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

datasheet page 9



ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Table 23-6. ADC Auto Trigger Source Selections

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match A
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

Table 23-4. Input Channel Selections

MUX3...0	Single Ended Input
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4
0101	ADC5
0110	ADC6
0111	ADC7
1000	ADC8[1]

Table 23-5. ADC Prescaler Selections

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 23-3. Voltage Reference Selections for ADC

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal V_{ref} turned off
0	1	AV_{CC} with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 1.1V Voltage Reference with external capacitor at AREF pin

ADLAR = 0

Bit	15	14	13	12	11	10	9	8	
(0x79)	–	–	–	–	–	–	ADC9	ADC8	ADCH
(0x78)	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	

ADLAR = 1

Bit	15	14	13	12	11	10	9	8	
(0x79)	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
(0x78)	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	

datasheet page 10

ADCSRA – ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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 – ADEN: ADC Enable**

Writing this bit to one enables the ADC. By writing it to zero, the ADC is turned off. Turning the ADC off while a conversion is in progress, will terminate this conversion.

- **Bit 6 – ADSC: ADC Start Conversion**

In Single Conversion mode, write this bit to one to start each conversion. In Free Running mode, write this bit to one to start the first conversion. The first conversion after ADSC has been written after the ADC has been enabled, or if ADSC is written at the same time as the ADC is enabled, will take 25 ADC clock cycles instead of the normal 13. This first conversion performs initialization of the ADC.

ADSC will read as one as long as a conversion is in progress. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect.

- **Bit 5 – ADATE: ADC Auto Trigger Enable**

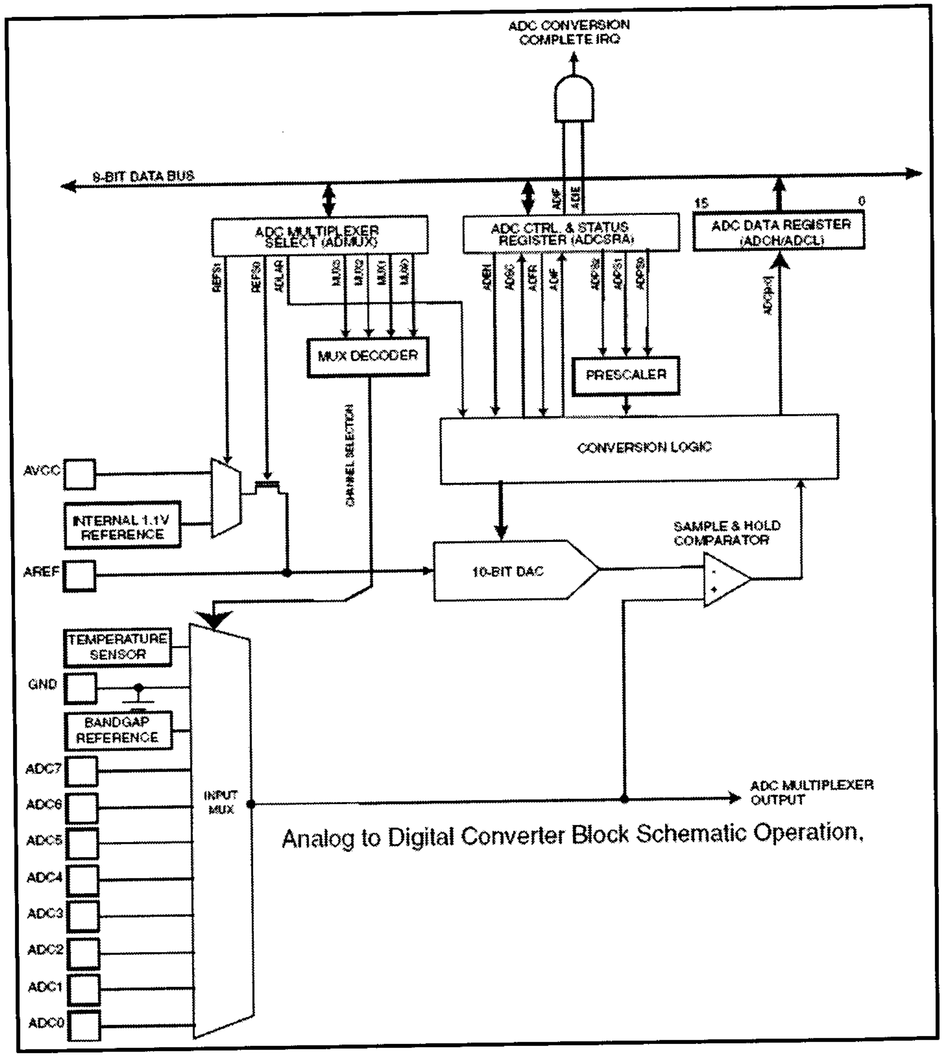
When this bit is written to one, Auto Triggering of the ADC is enabled. The ADC will start a conversion on a positive edge of the selected trigger signal. The trigger source is selected by setting the ADC Trigger Select bits, ADTS in ADCSRB.

- **Bit 4 – ADIF: ADC Interrupt Flag**

This bit is set when an ADC conversion completes and the Data Registers are updated. The ADC Conversion Complete Interrupt is executed if the ADIE bit and the I-bit in SREG are set. ADIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ADIF is cleared by writing a logical one to the flag. Beware that if doing a Read-Modify-Write on ADCSRA, a pending interrupt can be disabled. This also applies if the SBI and CBI instructions are used.

- **Bit 3 – ADIE: ADC Interrupt Enable**

When this bit is written to one and the I-bit in SREG is set, the ADC Conversion Complete Interrupt is activated.



Analog to Digital Converter Block Schematic Operation.

$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$