// ADC Without interrupt:

unsigned adcvalue, loval, hival;

// Set up ADC

// Write 0 to PRR bit 0 to disable power reduction on the ADC

PRR &= 0b11111110;

// Enable ADC, don't start conversion, disable ADIF and ADIE and

// set ADPS2-0(last 3 bits) to set prescalar value (in this case 111->128)

ADCSRA=0b10000111;

// Set up ADMUX to set reference voltage(REFS:1-0),

// and converting channel(MUX:2-0)

ADMUX=0b01000000;

while(1)

{

// Start ADC conversion by writing a 1 to ADSC bit.

ADCSRA |= 0b01000000;

// Wait for conversion to end.

while(ADCSRA & 0x01000000);

// Clear ADIF

ADSCRA |=0b00010000;

// Read ADC value

loval=ADCL;

hival=ADCH;

adcvalue = hival\*256+loval;

}

// ADC with interrupt

#include <avr/interrupt.h>

/\* irq\_vect can be:

ADC\_vect: handle end-of-conversion interrupt from the A/D Converter.

INT0\_vect: handle IRQ0 (pin3)

INT1\_vect: handle IRQ1 (pin4)

BADISR\_vect:handle cases where an interrupt hsa no ISR

\*/

ISR(irq\_vect)

{

//... ISR body ...

unsigned loval, hival;

// Read the result from the registers

// Read ADC value

loval=ADCL;

hival=ADCH;

adcvalue = hival\*256+loval;

// Re-start the conversion

ADCSRA |= 0b01000000;

}

int main()

{

// Set up ADC

// Write 0 to PRR bit 0 to disable power reduction on the ADC

PRR &= 0b11111110;

// Enable ADC, don't start conversion, enable ADIE (ADC interrupt)

ADCSRA=0b10100111;

// Set up ADMUX to convert from channel.

ADMUX=0b01000000;

// Start ADC conversion by writing a 1 to ADSC bit.

ADCSRA |= 0b01000000;

while(1)

{

// do something here

}

}// main

// timer

#include <avr/io.h>

#include <avr/interrupt.h>

//Initialize Timer0

void InitTimer0(void)

{

//Set Initial Timer value

TCNT0=0;

//Place TOP timer value to Output compare register

OCR0A=249;

//Set Timer Mode (In this case CTC mode)

// and make toggle PD6/OC0A pin on compare match

TCCR0A=0b01000010;

// Enable interrupts.

TIMSK0|=0b10;

}

void StartTimer0(void)

{

//Set prescaler 64, therefore starts timer

TCCR0B=0b00000011;

// Enable global interrupts

sei();

}

// Set up the ISR for TOV0A

ISR(TIMER0\_COMPA\_vect)

{

// do something

}

// Analog for output

#include <avr/interrupt.h>

#include <avr/io.h>

int index=0;

void InitPWM()

{

// Set initial timer value

TCNT0=0;

// Set the initial OCR0A values

OCR0A=0;

// Set TCCR0A to clear OC0A when we reach 255,

// and choose mode 1 Phase correct PWM

TCCR0A=0b10000001;

// Enable compare interrupt

TIMSK0 |= 0b10;

}

// Start PWM signal

void startPWM()

{

// Set prescaler of 0b011, or 64.

TCCR0B=0b00000011;

// Set global interrupts

sei();

}

// ISR

ISR(TIMER0\_COMPA\_vect)

{

// Do something like reset OCR0A

}