ΕΞΕΤΑΣΗ ΕΡΓΑΣΤΗΡΙΟΥ ΠΡΟΗΓΜΕΝΟΙ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ

Μαρίνος Γκίζας 1054348

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define del 1000

ISR(ADC0\_WCOMP\_vect)  
{   
int intflags = ADC0.INTFLAGS;   
ADC0.INTFLAGS = intflags;   
PORTC.OUTCLR= PIN5\_bm; //LED5 is on   
  
}

ISR(PORTF\_PORT\_vect)  
{   
//clear the interrupt flag   
int intflags = PORTF.INTFLAGS;   
PORTF.INTFLAGS=intflags;

PORTD.OUTCLR= PIN1\_bm; //Pin is on   
\_delay\_ms(del);   
PORTD.OUT |= PIN1\_bm; //Pin is off   
\_delay\_ms(del);  
   
}

int main(){

PORTC.DIR |= PIN5\_bm; //PIN5 is output

PORTC.OUT |=PIN5\_bm; //PIN5 is off  
PORTD.DIR |= PIN3\_bm; // PIN3 is output  
PORTD.OUT |= PIN3\_bm; //PIN3 is off

ADC0.CTRLA |= ADC\_RESSEL\_10BIT\_gc; //10-bit resolution   
ADC0.CTRLA |= ADC\_FREERUN\_bm; //Free-Running mode enabled   
ADC0.CTRLA |= ADC\_ENABLE\_bm; //Enable   
ADC ADC0.MUXPOS |= ADC\_MUXPOS\_AIN7\_gc;   
//The bit   
//Enable Debug Mode   
ADC0.DBGCTRL |= ADC\_DBGRUN\_bm; //Window Comparator Mode   
ADC0.WINHT |= 30; //Set threshold   
ADC0.INTCTRL |= ADC\_WCMP\_bm; //Enable Interrupts for   
WCM ADC0.CTRLE |= ADC\_WINCM0\_bm; //Interrupt when RESULT > WINHT sei(); ADC0.COMMAND |= ADC\_STCONV\_bm; //Start Conversion

PORTF.PIN2CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc;   
sei(); //enable interrupts

}