Εργαστήριο Μικρουπολογιστών

Παπαδόπουλος Κωνσταντίνος el20152
Σκούρτης Παύλος el20052

ΑΣΚΗΣΗ 1

.include "m328PBdef.inc"	
.equ FOSC_MHZ=16 ;MHz	
.equ DEL_mS=100 ;mS	
.equ DEL_NU= FOSC_MHZ*DEL_mS	
.def dc_value=r27	
TABLE:	
.DW 0x0005,0x001A,0x002E,0x0042,0x0057,0x006B,0x007F,0x0094,0x00A8,0x00BD	Πίνακας με τις προϋπολογισμένες τιμές
.DW 0x00D1,0x00E6,0x00FA	
reset:	
ldi r24,low(RAMEND)	
out SPL,r24	
ldi r24,high(RAMEND)	
out SPH,r24	
ldi r26,0b00111111	
out DDRB, r26	;PORTB=out
clr r26	
out DDRD,r26	;PortD == in
ldi r16,(1 << WGM10) | (1 << COM1A1)	;set TMR1Ain fast PWM 8-bit mode with non-inverted output and prescale=8
sts TCCR1A, r16	
ldi r16,(1 << WGM12) | (1 << CS10)	
sts TCCR1B, r16	

ldi ZH,HIGH(TABLE*2)	;the table with the dc_values is set on Z and the first dc_value is 50%
ldi ZL,LOW(TABLE*2)	;so its the 7th value on the TABLE
adiw ZL,12	
lpm	
mov dc_value,r0	
main:	
lpm	
mov dc_value,r0	
start:	
sts OCR1AL,dc_value	
in r26, PIND	Έλεγχος εισόδου
sbrs r26,1	
rjmp syn	
sbrs r26,2	
rjmp meion	
rjmp start	
syn:	Πατήθηκε αύξηση φωτεινότητας
ldi r24, low(DEL_NU)	
ldi r25, high(DEL_NU)	
rcall delay_mS	Χρησιμοποιείται για debouncing
cpi dc_value,0xFA	Έλεγχος ορίων
breq start	
adiw ZL,2	
rjmp main	
meion:	Πατήθηκε μείωση φωτεινότητας
ldi r24, low(DEL_NU)	
ldi r25, high(DEL_NU)	

rcall delay_mS	Χρησιμοποιείται για debouncing
cpi dc_value,0x05	Έλεγχος ορίων
breq start	
sbiw ZL,2	
rjmp main	
delay_mS:	
ldi r23, 249	
loop_inn:	
dec r23 ; 1 cycle	
nop ; 1 cycle	
brne loop_inn ; 1 or 2 cycles	
sbiw r24 ,1 ; 2 cycles	
brne delay_mS ; 1 or 2 cycles	
ret ;4 cycl	

ΑΣΚΗΣΗ 2

#define F_CPU 16000000UL	
#include "avr/io.h"	
#include "util/delay.h"	
void adc_init(){	
return;	
}	
uint8_t table [9]={0, 32, 64, 96. 128, 160, 192, 224,255};	
void open(){	
ADCSRA |=(1 << ADSC);	
uint8_t temp;	
while(1){	
temp=ADCSRA;	
temp=temp & 64; //0b01000000; apomonosi ADSC	
if(temp==0)break;	
}	
uint8_t adcvalueh,adcvaluel;	
adcvaluel=ADCL;	
adcvalueh=ADCH;	
if(adcvalueh>0 && adcvalueh<=32) {PORTD=0b00000001;}	Κατηγοροιοποίηση σταθμών τάσης και αντίστοιχη εξόδου
if(adcvalueh>32 && adcvalueh<=64) {PORTD=0b00000010;}	
if(adcvalueh>64 && adcvalueh<=96) {PORTD=0b00000100;}	
if(adcvalueh>96 && adcvalueh<=128) {PORTD=0b00001000;}	

if(adcvalueh>128 && adcvalueh<=160) {PORTD=0b00010000;}	
if(adcvalueh>160 && adcvalueh<=192) {PORTD=0b00100000;}	
if(adcvalueh>192 && adcvalueh<=224) {PORTD=0b01000000;}	
if(adcvalueh>224 && adcvalueh<=255) {PORTD=0b10000000;}	
return;	
}	
int main(){	
TCCR1A =(1 << WGM10) | (1 << COM1A1);	
TCCR1B =(1 << WGM12) | (1 << CS11);	
ADMUX =0b01100001;	
ADCSRA =0b10000111;	
uint8_t duty=134;	
OCR1AL=duty;	
DDRB |=0b00000010;	//B input
DDRD |=0b11111111;	//D output
while(1){	
uint8_t inpout;	
input = PINB;	
_delay_ms(100);	debouncing
input = input & 48;	Απομόνωση button p5-p6 as 0b00110000 = 48
if(input==32) {if(duty<245){duty+=20;OCR1AL=duty;} }	Μείωση button pb5

else if(input==16) {if(duty>25){duty-=20;OCR1AL=duty;} }	Αύξηση button pb6
open();	
}	
}	

Άσκηση 3

#define F_CPU 16000000UL	
#include "avr/io.h"	
#include "util/delay.h"	
uint8_t duty=134;	
void adc_init(){	
return;	
}	
uint8_t table [9]={0, 32, 64, 96, 128, 160, 192, 224,255};	
void mode1(){	Παρόμοιο με ασκ2
uint8_t input;	
input = PIND;	
_delay_ms(100);	//delay for debouncing
input = input & 6;	//απομόνωση button pd1-pd2 as 0b00000110 = 6
if(input==4) {if(duty<245){duty+=20;} }	//pb1 —> αύξηση
else if(input==2) {if(duty>25){duty-=20;} } // != exw arnitiki logiki	//pb2 —> μείωση
OCR1AL=duty;	
}	
void mode2(){	
ADCSRA |=(1 << ADSC);	
uint8_t temp;	
while(1){	
temp=ADCSRA;	

temp=temp & 64;	//0b01000000; απομόνωση ADSC
if(temp==0)break;	
}	
uint8_t adcvalueh,adcvaluel, duty2;	
adcvaluel=ADCL;	
adcvalueh=ADCH;	
if(adcvalueh>0 && adcvalueh<=32){duty=250;}	Κατηγοροιοποίηση σταθμών τάσης και αντίστοιχη εξόδου
if(adcvalueh>32 && adcvalueh<=64){duty=200;}	
if(adcvalueh>64 && adcvalueh<=96){duty=170;}	
if(adcvalueh>96 && adcvalueh<=128){duty=140;}	
if(adcvalueh>128 && adcvalueh<=160){duty=110;;}	
if(adcvalueh>160 && adcvalueh<=192){duty=80;}	
if(adcvalueh>192 && adcvalueh<=224){duty=50;}	
if(adcvalueh>224 && adcvalueh<=255){duty=20;}	
OCR1AL=duty;	
return;	
}	
int main(){	
TCCR1A =(1 << WGM10) | (1 << COM1A1);	

TCCR1B =(1 << WGM12) | (1 << CS11);	
ADMUX =0b01100000;	// adc0 —>ποντεσιόμετρο
ADCSRA =0b10000111;	
DDRD |=0b00000000;	//D input
DDRB |=0b11111111;	//B output
PORTB=0b00000000;	
int mode=0;	
while(1){	
uint8_t input;	
input = PIND;	
_delay_ms(100);	//delay for debouncing
input = input & 192;	//button p6-p7 as 0b11000000 = 192
if(input==128) {mode=1;}	// PRESSED P7
else if(input==64) {mode=2;}	// PRESSED P6
if (mode==1){	
mode1();	
}	
if(mode==2) {	
mode2();	
}	
}	
}	