**ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ**

**ΤΕΤΑΡΤΗ ΑΣΚΗΣΗ**

Στοιχεία μελών ομάδας:

1ο μέλος ομάδας

Ονοματεπώνυμο: Γκουργκούτας Βασίλειος

ΑΜ: 1084667

Έτος: 4ο

2ο μέλος ομάδας

Ονοματεπώνυμο: Κοντογιώργος Αναστάσιος

ΑΜ: 1090084

Έτος: 4ο

**Κώδικας για το πρώτο ερώτημα:**

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output // LED0 Indicates that the plants need watering

PORTD.OUT |= PIN0\_bm; //PIN0\_bm //LED is off

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output // LED1 Indicates that the humidity has exceeded the desired levels and must be lowered through a ventilation system

PORTD.OUT |= PIN1\_bm; //PIN1\_bm //LED is off

// Initialize the ADC for single conversion mode

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

ADC0.DBGCTRL |= ADC\_DBGRUN\_bm; //Enable Debug Mode

ADC0.WINLT |= 5; //Set threshold

ADC0.WINHT |= 10; //Set threshold

ADC0.INTCTRL |= ADC\_WCMP\_bm; //Enable Interrupts for WCM

ADC0.CTRLE |= 0x4; //Interrupt when RES<WINLT or RES>WINHT

sei(); // Enable Interrupts

ADC0.COMMAND |= ADC\_STCONV\_bm; //Start Conversion

while(1){

;

}

}

ISR(ADC0\_WCOMP\_vect){

cli();

int intflags = ADC0.INTFLAGS;

ADC0.INTFLAGS = intflags;

if (ADC0.RES<ADC0.WINLT){ // We check if the temperature is below the threshold WINLT and we enable the LED0

PORTD.OUTCLR = PIN0\_bm;

PORTD.OUT |= PIN1\_bm;

}

else if (ADC0.RES>ADC0.WINHT){ // We check if the temperature is over the threshold WINHT and we enable the LED1

PORTD.OUTCLR = PIN1\_bm;

PORTD.OUT |= PIN0\_bm;

}

sei();

}

**Επεξήγηση του κώδικα για το πρώτο ερώτημα**

Αρχικά ορίζουμε τα pin που πρόκειται να χρησιμοποιήσουμε για την άσκηση μας. Το pin0 θα χρησιμοποιηθεί ώστε να ενεργοποιήσουμε το led0 για να υλοποιηθεί η ένδειξη η οποία σηματοδοτεί ότι τα φυτά του θερμοκηπίου πρέπει να ποτιστούν. Με το pin1 και το led1 δηλώνουμε πως η υγρασία έχει υπερβεί τα επιθυμητά επίπεδα και πρέπει να κατέβει μέσω ενός

συστήματος αερισμού. Έπειτα αρχικοποιούμε τον ADC ώστε να μπορούμε μέσω δύο κατωφλίων να καταλαβαίνουμε ποια ένδειξη έχει ενεργοποιηθεί. Ορίζουμε με την εντολή ADC0.WINLT |= 5; το κατώφλι σύμφωνα με το οποίο όταν η τιμή που δίνουμε στο RES είναι μικρότερη του κατωφλίου αυτού, τότε θα ενεργοποιείται το led0 που υποδεικνύει ότι τα φυτά του θερμοκηπίου χρειάζονται πότισμα. Επιπλέον με την εντολή ADC0.WINHT |= 10; ορίζουμε ένα δεύτερο κατώφλι. Εάν το RES είναι μεγαλύτερο του συγκεκριμένου κατωφλίου, τότε θα ενεργοποιείται το led1 το οποίο δείχνει ότι η υγρασία είναι υψηλή και το θερμοκήπιο χρειάζεται αερισμό. Με την εντολή ADC0.CTRLE |= 0x4; ενεργοποιούμε το interrupt το οποίο ελέγχει και τις δύο συνθήκες RES<WINLT και RES>WINHT. Ανάλογα με το ποια συνθήκη έχει προκαλέσει το interrupt μεταβαίνουμε στην ISR του ADC. Μέσα στην ISR ελέγχουμε με μία συνθήκη if-else το αποτέλεσμα που έχουμε στον RES με τα κατώφλια. Εάν το RES<WINLT τότε ενεργοποιείται το led0 ενώ εάν το RES>WINHT τότε το led1 ανάβει.

**Διάγραμμα ροής για το πρώτο ερώτημα**

Εικόνα που περιέχει διάγραμμα, κείμενο, γραμμή, σκίτσο/σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα

**Κώδικας για το δεύτερο ερώτημα:**

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#define T1 20 // Time for the T1 (1ms) value = T\*f(timer) = 1ms\*19.531Khz = 19.531 --> aproximately 20

// Variables

int led0\_on = 0; //This variable is used to for indicate that the LED0 is on

int led1\_on = 0; //This variable is used to for indicate that the LED1 is on

int timer = 0; //This variable is used to store the time of the timer TCA

int flag = 0; //This variable is used to the ISR of TCA

int flag1= 0; //This variable is used to the ISR of PWM

int counter = 0; //This variable is used to count if the palm is in rising edge or in the falling edge

int count\_to\_four = 0; //This variable is used to count the four rising edges

int watering = 0; //This variable is used to indicate that the PIN5 is enable

int ventilation = 0; //This variable is used to indicate that the PIN6 is enable

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output // LED0 Indicates that the plants need watering

PORTD.OUT |= PIN0\_bm; //PIN0\_bm //LED is off

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output // LED1 Indicates that the humidity has exceeded the desired levels and must be lowered through a ventilation system

PORTD.OUT |= PIN1\_bm; //PIN1\_bm //LED is off

PORTD.DIR |= PIN2\_bm; //PIN2\_bm //PIN is output // LED2 Indicates that ventilation system is enabled

PORTD.OUT |= PIN2\_bm; //PIN2\_bm //LED is off

//pullup enable and Interrupt enabled with sense on both edges

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc;

//pullup enable and Interrupt enabled with sense on both edges

PORTF.PIN6CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc;

while(1){

if(watering==1 && led0\_on==1) { // If PIN5 and LED0 is on then we enable the watering system

timer = ADC0.WINLT-ADC0.RES;

TCA0.SINGLE.CNT = 0; //clear counter

TCA0.SINGLE.CTRLB = 0; //Normal Mode (TCA\_SINGLE\_WGMODE\_NORMAL\_gc )

TCA0.SINGLE.CMP0 = timer;//When CMP0 reaches this value -> interrupt

//CLOCK\_FREQUENCY/1024

TCA0.SINGLE.CTRLA = 0x7<<1; //TCA\_SINGLE\_CLKSEL\_DIV1024\_gc

TCA0.SINGLE.CTRLA |= 1; //Enable

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; //Interrupt Enable (=0x10)

sei();

while(flag==0){ // We remain this while until the timer TCA finished

;

}

watering=0;

flag=0;

}

else if(ventilation==1 && led1\_on==1){ // If PIN6 and LED1 is on then we enable the ventilation system

//prescaler=1024

TCA0.SINGLE.CTRLA=TCA\_SINGLE\_CLKSEL\_DIV1024\_gc;

TCA0.SINGLE.PER = T1; //select the resolution

TCA0.SINGLE.CMP0 = 10; //select the duty cycle

//select Single\_Slope\_PWM

TCA0.SINGLE.CTRLB |= TCA\_SINGLE\_WGMODE\_SINGLESLOPE\_gc;

//enable interrupt Overflow

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_OVF\_bm;

TCA0.SINGLE.CTRLA |= TCA\_SINGLE\_ENABLE\_bm; //Enable

sei();

while(flag1==0){ //We wait until the four rising edges are completed

;

}

ventilation=0;

flag1=0;

}

while(watering==0 && ventilation==0){ // We remain inside this while until the watering and ventilation system is enabled

// Initialize the ADC for single conversion mode

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

ADC0.DBGCTRL |= ADC\_DBGRUN\_bm; //Enable Debug Mode

ADC0.WINLT |= 5; //Set threshold

ADC0.WINHT |= 10; //Set threshold

ADC0.INTCTRL |= ADC\_WCMP\_bm; //Enable Interrupts for WCM

ADC0.CTRLE |= 0x4; //Interrupt when RES<WINLT or RES>WINHT

sei(); // Enable Interrupts

ADC0.COMMAND |= ADC\_STCONV\_bm; //Start Conversion

}

}

}

ISR(ADC0\_WCOMP\_vect){

cli();

int intflags = ADC0.INTFLAGS;

ADC0.INTFLAGS = intflags;

if (ADC0.RES<ADC0.WINLT){ // We check if the temperature is below the threshold WINLT and we enable the LED0

PORTD.OUTCLR = PIN0\_bm;

PORTD.OUT |= PIN1\_bm;

led0\_on = 1;

led1\_on = 0;

}

else if (ADC0.RES>ADC0.WINHT){ // We check if the temperature is over the threshold WINHT and we enable the LED1

PORTD.OUTCLR = PIN1\_bm;

PORTD.OUT |= PIN0\_bm;

led1\_on = 1;

led0\_on = 0;

}

ADC0.CTRLA &= ~ADC\_ENABLE\_bm; //Disable ADC

sei();

}

ISR(TCA0\_CMP0\_vect){ // ISR that is used for the watering system

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS=intflags; //clear interrupt flag

flag = 1;

led0\_on = 0;

PORTD.OUT |= PIN0\_bm;

TCA0.SINGLE.CTRLA = 0; //Disable

}

ISR(PORTF\_PORT\_vect){ // ISR that checks which button are pressed PIN5 or PIN6

if((PORTF.INTFLAGS & 0b00100000)==0x20 && led0\_on==1){

watering = 1;

ventilation = 0;

}

else if((PORTF.INTFLAGS & 0b01000000) ==0x40 && led1\_on==1){

ventilation = 1;

watering = 0;

}

int y = PORTF.INTFLAGS;

PORTF.INTFLAGS=y; //clear the interrupt flag

}

ISR(TCA0\_OVF\_vect){ //This ISR is used for the PWM of the ventilation system

//clear the interrupt flag

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS = intflags;

PORTD.OUTCLR |= PIN1\_bm; //LED1 is on

counter++;

if(counter==1 ) // With these if - else statements we check when the PWM is in the rising edge

{

count\_to\_four++;

PORTD.OUTCLR = PIN2\_bm; //LED2 is on

}

else if(counter==2 ){

count\_to\_four++;

PORTD.OUT |= PIN2\_bm; //LED2 is off

}

else if(counter==3 ){

count\_to\_four++;

PORTD.OUTCLR = PIN2\_bm; //LED2 is on

}

else if(counter==4 ){

count\_to\_four++;

PORTD.OUT |= PIN2\_bm; //LED2 is off

}

if (count\_to\_four==4){ // We check if the PWM has done the four rising edges

PORTD.OUT |= PIN1\_bp; //LED1 is off

PORTD.OUT |= PIN2\_bm; //LED2 is off

flag1=1;

TCA0.SINGLE.CTRLA = 0; //Disable

counter=0;

count\_to\_four=0;

}

}

**Επεξήγηση του κώδικα για το δεύτερο ερώτημα**

Για το δεύτερο ερώτημα που αφορά την προσθήκη της λειτουργίας του ποτίσματος και του αερισμού, αρχικά ορίσαμε ένα ακόμη pin2 το οποίο αφορά την ένδειξη η οποία δηλώνει ότι το σύστημα εξαερισμού έχει ενεργοποιηθεί. Έπειτα ενεργοποιούμε τα pins 5 και 6 του portf για να αξιοποιήσουμε τις δύο λειτουργίες του ποτίσματος και του εξαερισμού. Στην συνέχεια υλοποιούμε μέσα σε μία συνεχόμενη while(1) μία συνθήκη if η οποία έχει σαν ορίσματα δύο μεταβλητές. Τα ορίσματα αυτά αλλάζουν τιμή μέσα στην Isr του ADC (led0\_on) και στην Isr του PORTF (watering). Η μεταβλητή led0\_on γίνεται ένα όταν έχουμε ενεργοποιήσει το led0 για το πότισμα. Επιπλέον η μεταβλητή watering γίνεται ίση με ένα όταν έχουμε πατήσει το pin5 του portf και μεταβούμε στην Isr του. Εκεί γίνεται έλεγχος με τα intflags του portf και κατάλληλο bit masking ((PORTF.INTFLAGS & 0b00100000)==0x20) και εάν η τιμή είναι ίση με το 0x20 και η μεταβλητή που έχουμε ορίσει για το led0 (led0\_on=1) είναι ένα, τότε η μεταβλητή watering ισούται με ένα. Εφόσον και οι δύο αυτές συνθήκες πληρούνται ((PORTF.INTFLAGS & 0b00100000)==0x20 && led0\_on==1), τότε εισερχόμαστε στην if όπου έχουμε ορίσει έναν timer TCA0 ο οποίος εκτελείται για χρονικό διάστημα που δίνεται από την αφαίρεση της τελευταίας τιμής του RES από το κατώφλι του WINLT. Έπειτα ενεργοποιείται η Isr του Timer TCA0 (TCA\_CMP0\_vect) όπου απενεργοποιείται το led0 και ο χρονιστής και επιστρέφουμε στην αρχική λειτουργία του συστήματος. Στην συνέχεια έχουμε ορίσει μία else if στην οποία πρέπει να είναι οι μεταβλητές ventilation και led1\_on ίσες με ένα ώστε να μπορέσουμε να εισέλθουμε στην συνθήκη. Η μεταβλητή ventilation γίνεται ίση με ένα όταν πατήσουμε το pin6 και μεταφερθούμε στην Isr του portf. Εκεί γίνεται έλεγχος με τα intflags του portf και κατάλληλο bit masking ((PORTF.INTFLAGS & 0b01000000) ==0x40) και εάν η τιμή είναι ίση με το 0x40 και η μεταβλητή που έχουμε ορίσει για το led1 (led1\_on=1) είναι ένα, τότε η μεταβλητή ventilation ισούται με ένα. Η μεταβλητή led1\_on γίνεται ίση με ένα όταν η τιμή του RES είναι μεγαλύτερη από το κατώφλι WINHT. Εάν και οι δύο συνθήκες πληρούνται (((PORTF.INTFLAGS & 0b01000000) ==0x40 && led1\_on==1), τότε μετακινούμαστε μέσα στην else if όπου έχουμε ορίσει τον timer TCA0 για λειτουργία PWM παλμού με περίοδο Τ=1ms και Duty Cycle=50%. Γίνεται interrupt και μεταβαίνουμε στην Isr με TCA\_OVF\_vect. Μέσα στην συγκεκριμένη Isr έχουμε ορίσει μια μεταβλητή counter η οποία μετράει πότε βρισκόμαστε σε rising edge. Επειδή θέλουμε να ολοκληρωθούν 4 rising edges και μετά να εξέλθουμε από την λειτουργία του εξαερισμού και ταυτόχρονα να ανάβει και να σβήνει εναλλάξ το led2 στα rising edges, έχουμε ορίσει συνθήκη με δομή if-else η οποία ανάβει και σβήνει το led2 ανάλογα με την τιμή της μεταβλητής counter. Τέλος υπάρχει μία συνθήκη if η οποία ελέγχει εάν έχουν ολοκληρωθεί 4 rising edges με μία μεταβλητή που έχουμε ορίσει (count\_to\_four), η οποία όταν φτάσει την τιμή τέσσερα τότε απενεργοποιούμε τα led 1 και 2 και απενεργοποιούμε επίσης τον timer.

**Διάγραμμα ροής για το δεύτερο ερώτημα**

Εικόνα που περιέχει σκίτσο/σχέδιο, διάγραμμα, μοτίβο

Περιγραφή που δημιουργήθηκε αυτόματα

**Κώδικας για το τρίτο ερώτημα:**

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#define T1 20 // Time for the T1 (1ms) value = T\*f(timer) = 1ms\*19.531Khz = 19.531 --> aproximately 20

// Variables

int led0\_on = 0; //This variable is used to for indicate that the LED0 is on

int led1\_on = 0; //This variable is used to for indicate that the LED1 is on

int timer = 0;//This variable is used to store the time of the timer TCA

int flag = 0;//This variable is used to the ISR of TCA

int flag1= 0;//This variable is used to the ISR of PWM

int counter = 0;//This variable is used to count if the palm is in rising edge or in the falling edge

int count\_to\_four = 0;//This variable is used to count the four rising edges

int watering = 0;//This variable is used to indicate that the PIN5 is enable

int ventilation = 0;//This variable is used to indicate that the PIN6 is enable

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output // LED0 Indicates that the plants need watering

PORTD.OUT |= PIN0\_bm; //PIN0\_bm //LED is off

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output // LED1 Indicates that the humidity has exceeded the desired levels and must be lowered through a ventilation system

PORTD.OUT |= PIN1\_bm; //PIN1\_bm //LED is off

PORTD.DIR |= PIN2\_bm; //PIN2\_bm //PIN is output // LED2 Indicates that ventilation system is enabled

PORTD.OUT |= PIN2\_bm; //PIN2\_bm //LED is off

//pullup enable and Interrupt enabled with sense on both edges

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc;

//pullup enable and Interrupt enabled with sense on both edges

PORTF.PIN6CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc;

while(1){

if(watering==1 && led0\_on==1) {// If PIN5 and LED0 is on then we enable the watering system

timer = ADC0.WINLT-ADC0.RES;

TCA0.SINGLE.CNT = 0; //clear counter

TCA0.SINGLE.CTRLB = 0; //Normal Mode (TCA\_SINGLE\_WGMODE\_NORMAL\_gc )

TCA0.SINGLE.CMP0 = timer;//When CMP0 reaches this value -> interrupt

//CLOCK\_FREQUENCY/1024

TCA0.SINGLE.CTRLA = 0x7<<1; //TCA\_SINGLE\_CLKSEL\_DIV1024\_gc

TCA0.SINGLE.CTRLA |= 1; //Enable

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; //Interrupt Enable (=0x10)

sei();

while(flag==0){// We remain this while until the timer TCA finished

;

}

watering=0;

flag=0;

}

else if(ventilation==1 && led1\_on==1){// If PIN6 and LED1 is on then we enable the ventilation system

//prescaler=1024

TCA0.SINGLE.CTRLA=TCA\_SINGLE\_CLKSEL\_DIV1024\_gc;

TCA0.SINGLE.PER = T1; //select the resolution

TCA0.SINGLE.CMP0 = 10; //select the duty cycle

//select Single\_Slope\_PWM

TCA0.SINGLE.CTRLB |= TCA\_SINGLE\_WGMODE\_SINGLESLOPE\_gc;

//enable interrupt Overflow

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_OVF\_bm;

TCA0.SINGLE.CTRLA |= TCA\_SINGLE\_ENABLE\_bm; //Enable

sei();

while(flag1==0){//We wait until the four rising edges are completed

;

}

ventilation=0;

flag1=0;

}

while(watering==0 && ventilation==0){ // We remain inside this while until the watering and ventilation system is enabled

// Initialize the ADC for single conversion mode

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

ADC0.DBGCTRL |= ADC\_DBGRUN\_bm; //Enable Debug Mode

ADC0.WINLT |= 5; //Set threshold

ADC0.WINHT |= 10; //Set threshold

ADC0.INTCTRL |= ADC\_WCMP\_bm; //Enable Interrupts for WCM

ADC0.CTRLE |= 0x4; //Interrupt when RES<WINLT or RES>WINHT

sei(); // Enable Interrupts

ADC0.COMMAND |= ADC\_STCONV\_bm; //Start Conversion

}

}

}

ISR(ADC0\_WCOMP\_vect){

cli();

int intflags = ADC0.INTFLAGS;

ADC0.INTFLAGS = intflags;

if (ADC0.RES<ADC0.WINLT){ // We check if the temperature is below the threshold WINLT and we enable the LED0

PORTD.OUTCLR = PIN0\_bm;

PORTD.OUT |= PIN1\_bm;

led0\_on = 1;

led1\_on = 0;

}

else if (ADC0.RES>ADC0.WINHT){ // We check if the temperature is over the threshold WINHT and we enable the LED1

PORTD.OUTCLR = PIN1\_bm;

PORTD.OUT |= PIN0\_bm;

led1\_on = 1;

led0\_on = 0;

}

ADC0.CTRLA &= ~ADC\_ENABLE\_bm; //Disable ADC

sei();

}

ISR(TCA0\_CMP0\_vect){ // ISR that is used for the watering system

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS=intflags; //clear interrupt flag

flag = 1;

led0\_on = 0;

PORTD.OUT |= PIN0\_bm;

TCA0.SINGLE.CTRLA = 0; //Disable

}

ISR(PORTF\_PORT\_vect){ // ISR that checks which button are pressed PIN5 or PIN6

if(( (PORTF.INTFLAGS & 0b00100000) == 0x20) && led0\_on==1){

watering = 1;

ventilation=0;

}

else if(( (PORTF.INTFLAGS & 0b01000000) == 0x40) && led1\_on==1){

ventilation = 1;

watering=0;

}

else if( (PORTF.INTFLAGS == 0x20 && led1\_on==1) || (PORTF.INTFLAGS == 0x40 && led0\_on==1) ){

PORTD.OUTCLR = PIN0\_bm | PIN1\_bm | PIN2\_bm;

PORTD.OUT |= PIN0\_bm | PIN1\_bm | PIN2\_bm;

}

int y = PORTF.INTFLAGS;

PORTF.INTFLAGS=y; //clear the interrupt flag

}

ISR(TCA0\_OVF\_vect){ //This ISR is used for the PWM of the ventilation system

//clear the interrupt flag

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS = intflags;

PORTD.OUTCLR |= PIN1\_bm; //LED is on

counter++;

if(counter==1 ) // With these if - else statements we check when the PWM is in the rising edge

{

count\_to\_four++;

PORTD.OUTCLR = PIN2\_bm; //LED2 is on

}

else if(counter==2 ){

count\_to\_four++;

PORTD.OUT |= PIN2\_bm; //LED2 is off

}

else if(counter==3 ){

count\_to\_four++;

PORTD.OUTCLR = PIN2\_bm; //LED2 is on

}

else if(counter==4 ){

count\_to\_four++;

PORTD.OUT |= PIN2\_bm; //LED2 is off

}

if (count\_to\_four==4){ // We check if the PWM has done the four rising edges

PORTD.OUT |= PIN1\_bp; //PIN1\_bm //LED1 is off

PORTD.OUT |= PIN2\_bm; //PIN2\_bm //LED2 is off

flag1=1;

TCA0.SINGLE.CTRLA = 0; //Disable

counter=0;

count\_to\_four=0;

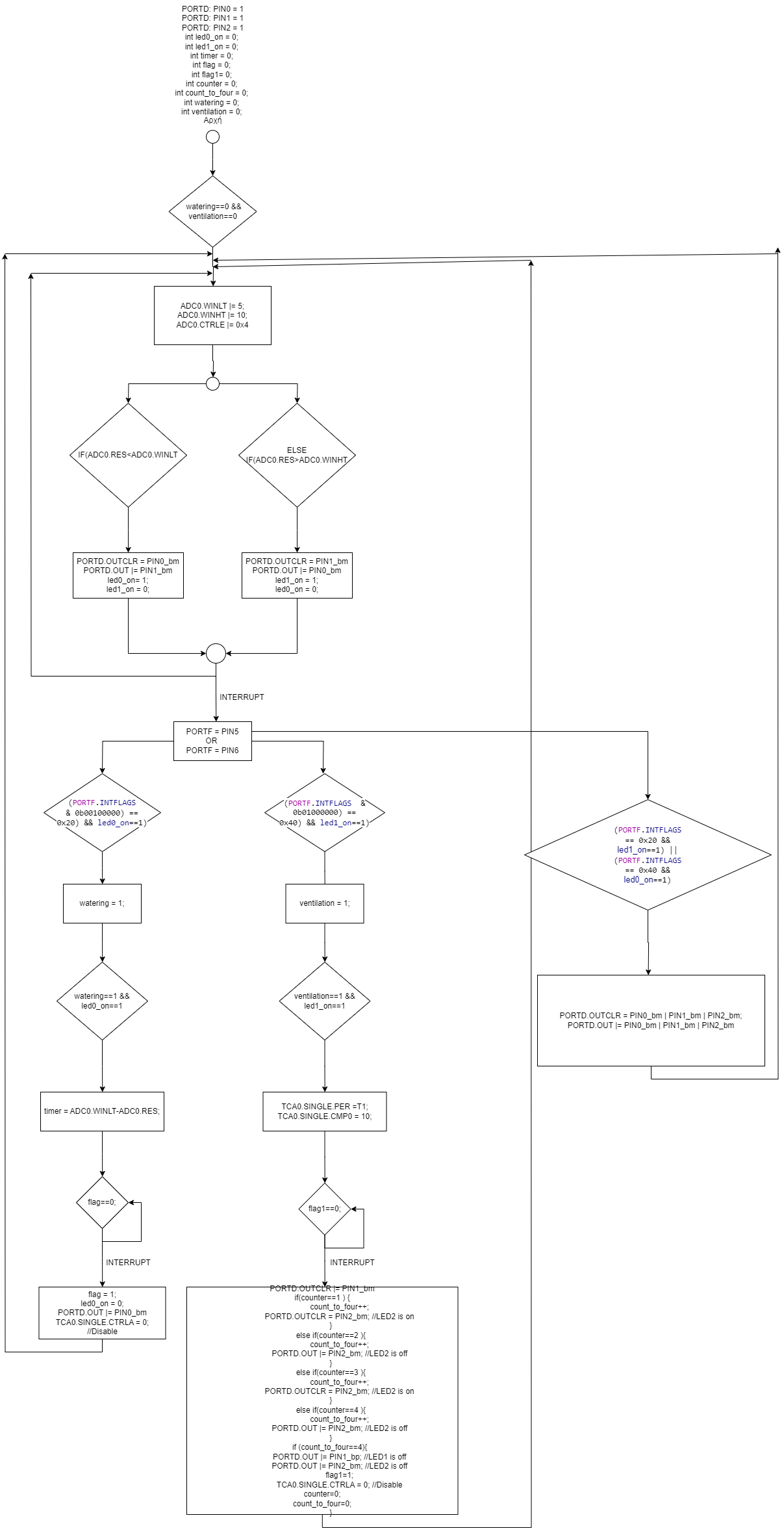
}

}

**Επεξήγηση του κώδικα για το τρίτο ερώτημα**

Για το τρίτο ερώτημα που αφορά την υλοποίηση της ένδειξης που ειδοποιεί τον χρήστη για λάθος εισαγωγή κουμπιού, έχουμε προσθέσει στην Isr του PORTF (PORTF\_PORT\_vect) την συνθήκη else-if η οποία ελέγχει εάν έχει ενεργοποιηθεί το λαμπάκι led0 (με την χρήση της μεταβλητής led0\_on) που δηλώνει ένδειξη για το πότισμα των φυτών και ταυτόχρονα έχει πατηθεί το pin6 που υποδεικνύει ότι πρέπει να γίνει εξαερισμός του θερμοκηπίου ή έχει ενεργοποιηθεί το led1 (με την χρήση της μεταβλητής led1\_on) που είναι η ένδειξη για τον εξαερισμό και ταυτόχρονα έχει πατηθεί το κουμπί pin5 που υποδεικνύει ότι πρέπει να ποτιστούν τα φυτά. Εάν συμβεί μία από τις δύο περιπτώσεις, τότε εισερχόμαστε μέσα σε αυτή την συνθήκη όπου ενεργοποιούμε τα leds 0,1,2 για να δηλώσουμε την ένδειξη λάθος λειτουργίας και έπειτα με βηματική εκτέλεση απενεργοποιούμε τα συγκεκριμένα leds.

**Διάγραμμα ροής για το τρίτο ερώτημα**



**Χρόνοι εκτέλεσης του κώδικα**

* + - 1. Η τιμή που επιλέχθηκε για τον χρονιστή είναι η εξής:
      2. **Χρονικό διάστημα για τον εξαερισμό:** 20

Κανονικά αξιοποιείται ο τύπος value=Tf(timer).



Η τιμή που υπολογίστηκε από τον τύπο αυτό είναι η εξής: Χρονικό διάστημα για τον εξαερισμό:

value=T1\*f(timer)= 2msec\*19.531KHz=19.531 (Περίπου 20).

* + - 1. Με f(timer)=19.531Khz και Ν=1024.