**ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ**

**ΤΡΙΤΗ ΑΣΚΗΣΗ**

Στοιχεία μελών ομάδας:

1ο μέλος ομάδας

Ονοματεπώνυμο: Γκουργκούτας Βασίλειος

ΑΜ: 1084667

Έτος: 4ο

2ο μέλος ομάδας

Ονοματεπώνυμο: Κοντογιώργος Αναστάσιος

ΑΜ: 1090084

Έτος: 4ο

**Ερωτήματα Εργαστηριακής Άσκησης 3**

3.1) Ο κώδικας που υλοποιήθηκε για τη λειτουργία ενεργοποίησης του ανεμιστήρα μετά την ενεργοποίηση του διακόπτη (switch 5) είναι ο εξής:

#include <avr/io.h>

#include <avr/interrupt.h>

#define base\_an 40 // Time for the base (2ms) value = T\*f(timer) = 2ms\*19.531Khz = 39.062 --> aproximately 40 (for having normal numbers in our exercise)

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

int button = 0; // This is a variable used for pin 5 of PORTF

int blades = 0; //This is a variable used for counting the pulses of blades

int base = 0; //This is a variable used for counting the pulses of base

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output //This pin is used for the movement of the blades

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

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output //This pin is used for the movement of the base

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

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //pullup enable and Interrupt enabled with sense on both edges

while(button==0){ // We wait until the button pin 5 of PORTF is pressed

sei();

}

if (button==1){ //If pin 5 is pressed then the code inside the if is executed

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SPLIT\_CLKSEL\_DIV1024\_gc

//TCA0.SPLIT.LCNT = 0; //clear counter for low register

TCA0.SPLIT.LPER = base\_an; // This is the period pulse for the base

TCA0.SPLIT.LCMP0 = 24;//When LCMP0 reaches this value -> interrupt

//TCA0.SPLIT.HCNT = 4;

TCA0.SPLIT.HPER = blades\_an; // This is the period pulse for the blades

TCA0.SPLIT.HCMP0 = 10;//When HCMP0 reaches this value -> interrupt

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; // We enable the LCMP0 or the HCMP0

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SINGLE\_CLKSEL\_DIV1024\_gc

TCA0.SPLIT.INTCTRL =0x03; // LUNF HUNF AND INTERRUPT ENABLE

sei();

while (1){

;

}

}

}

ISR(TCA0\_LUNF\_vect){

base = base + 1; // We increase the number of pulses the base has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000001)==1){// We check if the input to the interrupt comes from the pulse given by the movement of the base

if(base%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN1\_bm; // Pin1 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN1\_bm; // Pin1 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS = intflags;

}

ISR(TCA0\_HUNF\_vect){

blades = blades + 1; // We increase the number of pulses the blades has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000010)==2){// We check if the input to the interrupt comes from the pulse given by the movement of the blades

if (blades%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN0\_bm; //Pin0 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN0\_bm; // Pin0 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS=intflags;

}

ISR(PORTF\_PORT\_vect){

int y = PORTF.INTFLAGS; //Procedure to

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

button=1; // We change the variable to 1 if pin 5 of PORTF is pressed

}

**Επεξήγηση του πρώτου κώδικα**

Αρχικά ορίζουμε τα base\_an και blades\_an ώστε να δηλώσουμε τον παλμό περιόδου για την βάση και τις λεπίδες αντίστοιχα. Έπειτα ορίζουμε μία μεταβλητή button η οποία θα αξιοποιηθεί από το Interrupt που προκαλείται από το Pin5 του PORTF. Στην συνέχεια αρχικοποιούμε δύο μεταβλητές (blades και base) οι οποίες θα χρησιμοποιηθούν για να μετράνε πότε η βάση και οι λεπίδες βρίσκονται σε rising edge. Αυτό συμβαίνει όταν ο αριθμός που προκύπτει από τις μεταβλητές blades και base είναι περιττός. Εάν ο αριθμός είναι άρτιος, τότε τα leds των pin1 και pin0 θα είναι κλειστά αντίστοιχα. Μέσα στην main αρχικά οδηγούμε τα pin0 και pin1 του PORTD για την κίνηση των λεπίδων και την κίνηση της βάσης αντίστοιχα. Έπειτα με την εντολή PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; ενεργοποιούμε το pullup καθώς επίσης και τις διακοπές για το Pin5 του PORTF. Μετέπειτα υπάρχει μία while από την οποία θα μπορέσουμε να συνεχίσουμε την εκτέλεση του κώδικα μόνο όταν πατηθεί για πρώτη φορά το κουμπί του pin5. Θα μεταβούμε στην ISR του PORTF όπου η μεταβλητή button θα γίνει ίση με ένα και θα μπορέσουμε να περάσουμε από την συγκεκριμένη while. Στην συνέχεια υπάρχει μία if στην οποία ο κώδικας εκτελείται μόνο όταν η μεταβλητή button γίνει ίση με ένα. Μέσα στην συνθήκη if ορίζουμε δύο χρονιστές των 8-bit ο καθένας με την βοήθεια του split mode από τον 16-bit timer TCA0. Τοποθετούμε στον low register την κυκλική κίνηση της βάσης. Στην συνέχεια τοποθετήσαμε στις εντολές TCA0.SPLIT.LPER = base\_an; και TCA0.SPLIT.HPER = blades\_an; στις οποίες έχουν χρησιμοποιηθεί οι παλμοί περιόδου 2ms και 1ms αντίστοιχα, αλλά για λόγους ευκολίας χρησιμοποιήσαμε λίγο διαφορετικές τιμές για την προσομοίωση. Με τις εντολές TCA0.SPLIT.LCMP0 και TCA0.SPLIT.HCMP0 τοποθετούμε τους κύκλους λειτουργίας (Duty Cycles) της βάσης και του ανεμιστήρα αντίστοιχα. Έπειτα υπάρχει μία while(1) η οποία εκτελείται καθόλη την διάρκεια του προγράμματος. Επιπρόσθετα στην ISR(TCA0\_LUNF\_vect) έχουμε τοποθετήσει έναν μετρητή base ο οποίος μας βοηθάει στο να γνωρίζουμε πότε ο παλμός της βάσης είναι σε rising edge και πότε όχι. Ξεκινάμε από το μηδέν επομένως όταν η μεταβλητή base βρίσκεται σε περιττό αριθμό τότε το led του pin1 ανάβει. Ενώ όταν ο αριθμός της μεταβλητής base είναι άρτιος τότε το pin1 είναι κλειστό. Η παραπάνω λογική υλοποιήθηκε σε μία συνθήκη if else. Επιπρόσθετα υπάρχει μία if((TCA0.SPLIT.INTFLAGS & 0b00000001)==1) στην οποία ελέγχουμε εάν το interrupt προέρχεται από την κίνηση της βάσης. Παρόμοια λογική ακριβώς υλοποιήθηκε στην ISR(TCA0\_HUNF\_vect) με την χρήση της μεταβλητής blades. Τέλος αξιοποιήθηκε η ISR(PORTF\_PORT\_vect) η οποία ενεργοποιείται κατά το πάτημα του pin5 του PORTF.

**Διάγραμμα ροής πρώτης άσκησης**

Εικόνα που περιέχει διάγραμμα, σκίτσο/σχέδιο, Σχέδιο, τεχνικό σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα

3.2) Κώδικας που αφορά τη δεύτερη λειτουργία του παλμού της κυκλικής κίνησης των λεπίδων μέσω της ενεργοποίησης του διακόπτη (switch 5):

#include <avr/io.h>

#include <avr/interrupt.h>

#define base\_an 40 // Time for the base (2ms) value = T\*f(timer) = 2ms\*19.531Khz = 39.062 --> aproximately 40 (for having normal numbers in our exercise)

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

#define blades\_ypo2 10// The period of the pulse of the circular motion of the blades is halved

int button = 0; // This is a variable used for pin 5 of PORTF

int blades = 0; //This is a variable used for counting the pulses of blades

int base = 0; //This is a variable used for counting the pulses of base

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output //This pin is used for the movement of the blades

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

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output //This pin is used for the movement of the base

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

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //pullup enable and Interrupt enabled with sense on both edges

while(button==0){ // We wait until the button pin 5 of PORTF is pressed

sei();

}

while(1){

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SPLIT\_CLKSEL\_DIV1024\_gc

TCA0.SPLIT.LPER = base\_an; // This is the period pulse for the base

TCA0.SPLIT.LCMP0 = 24;//When LCMP0 reaches this value -> interrupt

TCA0.SPLIT.HPER = blades\_an; // This is the period pulse for the blades

TCA0.SPLIT.HCMP0 = 10;//When HCMP0 reaches this value -> interrupt

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; // We enable the LCMP0 or the HCMP0

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SINGLE\_CLKSEL\_DIV1024\_gc

TCA0.SPLIT.INTCTRL =0x03; // LUNF HUNF AND INTERRUPT ENABLE

while(button==2){ // If the button==2 then execute the second phase of fun

TCA0.SPLIT.HPER = blades\_ypo2; // The time of the blades is halved

TCA0.SPLIT.HCMP0 = 5;//When HCMP0 reaches this value -> interrupt

}

}

}

ISR(TCA0\_LUNF\_vect){

base = base + 1; // We increase the number of pulses the base has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000001)==1){// We check if the input to the interrupt comes from the pulse given by the movement of the base

if(base%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN1\_bm; // Pin1 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN1\_bm; // Pin1 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS = intflags;

}

ISR(TCA0\_HUNF\_vect){

blades = blades + 1; // We increase the number of pulses the blades has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000010)==2){ // We check if the input to the interrupt comes from the pulse given by the movement of the blades

if (blades%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN0\_bm; //Pin0 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN0\_bm; // Pin0 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS=intflags;

}

ISR(PORTF\_PORT\_vect){

int y = PORTF.INTFLAGS; //Procedure to

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

button=button +1; // We change the variable to 1 if pin 5 of PORTF is pressed

}

**Επεξήγηση του δευτέρου ερωτήματος**

Όσον αφορά το δεύτερο ερώτημα, αρχικά διαφοροποιήσαμε λίγο την συνάρτηση main. Μέσα στην main αντικαταστήσαμε την if(button==1) με την while(1). Αυτό συνέβη καθώς θέλουμε το πρόγραμμά μας να τρέξει και για δεύτερη φορά όταν πατηθεί το κουμπί του pin5 ξανά εκτελώντας την δεύτερη λειτουργία του παλμού της κυκλικής κίνησης των λεπίδων. Για τον σκοπό αυτό έχουμε αλλάξει στην ISR(PORTF\_PORT\_vect) το button==1 σε button= button+1 ώστε να μπορέσουμε να εισέλθουμε στην καινούρια while(button==2) που υλοποιήσαμε . Τοποθετήθηκε μέσα στην while(1) μία συνθήκη while(button==2) στην οποία έχουν τοποθετηθεί οι εξής εντολές: TCA0.SPLIT.HPER = blades\_ypo2, TCA0.SPLIT.HCMP0 = 5. Με την πρώτη εντολή αρχικά υποδιπλασιάζουμε την περίοδο του παλμού της κυκλικής κίνησης των λεπίδων ενώ με την δεύτερη εντολή αλλάζουμε τον κύκλο λειτουργίας των λεπίδων στο 50% (Duty Cycle = 50%). Ο υπόλοιπος κώδικας παραμένει ως έχει.

**Διάγραμμα ροής δευτέρου ερωτήματος**

Εικόνα που περιέχει διάγραμμα, σκίτσο/σχέδιο, Σχέδιο, τεχνικό σχέδιο

Περιγραφή που δημιουργήθηκε αυτόματα

3.3) Ο κώδικας για την λειτουργία απενεργοποίησης του ανεμιστήρα με δυνατότητα ενεργοποίησής του εκ νέου σε επόμενο πάτημα του διακόπτη είναι ο εξής:

#include <avr/io.h>

#include <avr/interrupt.h>

#define base\_an 40 // Time for the base (2ms) value = T\*f(timer) = 2ms\*19.531Khz = 39.062 --> aproximately 40 (for having normal numbers in our exercise)

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

#define blades\_ypo2 10// The period of the pulse of the circular motion of the blades is halved

int button = 0; // This is a variable used for pin 5 of PORTF

int blades = 0; //This is a variable used for counting the pulses of blades

int base = 0; //This is a variable used for counting the pulses of base

int main(){

PORTD.DIR |= PIN0\_bm; //PIN0\_bm //PIN is output //This pin is used for the movement of the blades

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

PORTD.DIR |= PIN1\_bm; //PIN1\_bm //PIN is output //This pin is used for the movement of the base

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

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //pullup enable and Interrupt enabled with sense on both edges

while(1){

while(button==0){ // We wait until the button pin 5 of PORTF is pressed

sei();

}

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SPLIT\_CLKSEL\_DIV1024\_gc

TCA0.SPLIT.LPER = base\_an; // This is the period pulse for the base

TCA0.SPLIT.LCMP0 = 24;//When LCMP0 reaches this value -> interrupt

TCA0.SPLIT.HPER = blades\_an; // This is the period pulse for the blades

TCA0.SPLIT.HCMP0 = 10;//When HCMP0 reaches this value -> interrupt

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_SPLITM\_bm; //ENABLE TCA SPLIT MODE

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; // We enable the LCMP0 or the HCMP0

//CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc | TCA\_SPLIT\_ENABLE\_bm; //TCA\_SINGLE\_CLKSEL\_DIV1024\_gc

TCA0.SPLIT.INTCTRL =0x03; // LUNF HUNF AND INTERRUPT ENABLE

while(button<2){ // While the button<2 until is it pressed more than one times

sei();

}

while(button==2){ // If the button==2 then execute the second phase of fun

sei();

TCA0.SPLIT.HPER = blades\_ypo2; // The time of the blades is halved

TCA0.SPLIT.HCMP0 = 5;//When HCMP0 reaches this value -> interrupt

}

if(button==3) { // If the button==3 then execute the third phase of fun

TCA0.SPLIT.CTRLA=0; // We disable the TCA0.SPLIT.CTRLA

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

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

blades = 0;

base = 0;

button = 0;

}

}

}

ISR(TCA0\_LUNF\_vect){

base = base + 1; // We increase the number of pulses the base has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000001)==1){// We check if the input to the interrupt comes from the pulse given by the movement of the base

if(base%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN1\_bm; // Pin1 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN1\_bm; // Pin1 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS = intflags;

}

ISR(TCA0\_HUNF\_vect){

blades = blades + 1; // We increase the number of pulses the blades has done

if ((TCA0.SPLIT.INTFLAGS & 0b00000010)==2){ // We check if the input to the interrupt comes from the pulse given by the movement of the blades

if (blades%2==1){ // If the number of pulse is odd then we enable pin1 led

PORTD.OUTCLR = PIN0\_bm; //Pin0 is on

}

else{ //If the number of pulse is even then the pin1 is off

PORTD.OUT |= PIN0\_bm; // Pin0 is off

}

}

//clear the interrupt flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS=intflags;

}

ISR(PORTF\_PORT\_vect){

int y = PORTF.INTFLAGS; //Procedure to

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

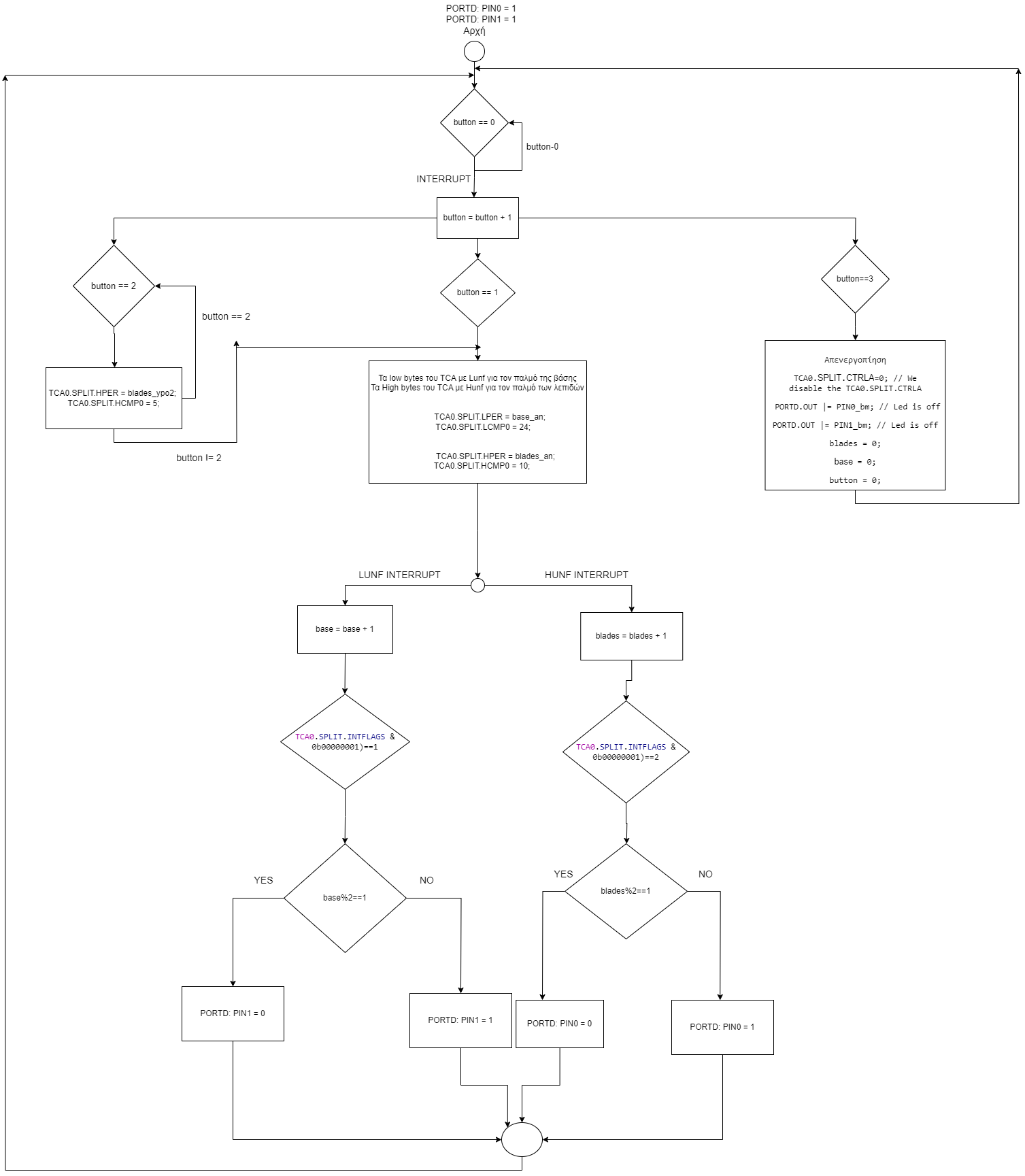
button=button +1; // We change the variable to 1 if pin 5 of PORTF is pressed

}

**Επεξήγηση του τρίτου ερωτήματος**

Κατά την συγγραφή του τρίτου ερωτήματος, τοποθετήθηκε η while(button==0) μέσα στην while(1) έτσι ώστε να μπορέσουμε να αξιοποιήσουμε τον ανεμιστήρα και τις λειτουργίες του εκ νέου. Προστέθηκε μία συνθήκη while(button<2) στην οποία περιμένουμε πριν μεταβούμε στην εκτέλεση της δεύτερης λειτουργίας του παλμού της κυκλικής κίνησης των λεπίδων. Πατώντας το κουμπί του pin5 ξανά αυξάνουμε το button κατά ένα, επομένως η μεταβλητή button θα ισούται με δύο. Επομένως μπρούμε να προσπεράσουμε την while(button<2) και να μεταβούμε στην while(button==2). Αφού εκτελέσουμε την δεύτερη λειτουργία, πατώντας και πάλι το κουμπί του pin5 μπορούμε να μεταβούμε στην ISR(PORTF\_PORT\_vect) όπου η μεταβλητή button θα αυξηθεί κατά ένα και θα γίνει ίση με τρία. Προστέθηκε μία συνθήκη if(button==3) στην οποία προσθέτουμε τη λειτουργία απενεργοποίησης του ανεμιστήρα. Πιο συγκεκριμένα απενεργοποιούμε τον timer TCA0 με την εντολή TCA0.SPLIT.CTRLA=0;. Έπειτα απενεργοποιούμε τα pin0 και pin1 και αρχικοποιούμε εκ νέου τις μεταβλητές blades, base και button. Τέλος επειδή έχουμε ορίσει μία while(1), μπορούμε να επιστρέψουμε στην αρχή με την δυνατότητα ενεργοποίησής του ανεμιστήρα σε επόμενο πάτημα του διακόπτη.

**Διάγραμμα ροής τρίτου ερωτήματος**



**Σχεδιασμός των παλμών**

Εικόνα που περιέχει διάγραμμα, γραμμή, μοτίβο

Περιγραφή που δημιουργήθηκε αυτόματα

Για τον πρώτο παλμό, ο οποίος αφορά την κυκλική κίνηση της βάσης έχουμε ότι το duty cycle ισούται με το 60%.

Για τον δεύτερο παλμό, ο οποίος αφορά την κυκλική κίνηση των λεπίδων έχουμε ότι το duty cycle ισούται με το 50%.

Επίσης η κυκλική κίνηση της βάσης έχει περίοδο T=2ms ενώ η κυκλική κίνηση των λεπίδων έχει περίοδο T=1ms. Για τον λόγο αυτό ο παλμός 2 ολοκληρώνει δύο rising edges ενώ την ίδια χρονική στιγμή ο παλμός 1 έχει ολοκληρώσει μόνο μία rising edge.

Επιπλέον έχουμε ότι η τιμή του CMPx καθορίζει το duty cycle. Εκεί όπου βρίσκεται ο κύκλος γίνεται update δηλαδή προκύπτει rising edge ενώ εκεί όπου έχουμε το x προκύπτει falling edge καθώς γίνεται match με την τιμή του CMP.

**Χρόνοι εκτέλεσης του κώδικα**

* + - 1. Οι τιμές που επιλέχθηκαν για τους χρονιστές είναι οι εξής:
      2. **Κυκλική κίνηση της βάσης:** 40
      3. **Κυκλική κίνηση της λεπίδων:** 20
      4. **Χρόνος για τον υποδιπλασιασμό των λεπίδων:** 10
      5. Κανονικά αξιοποιείται ο τύπος value=T*f(timer).*
      6. *Οι τιμές που υπολογίστηκαν από τον τύπο αυτό είναι οι εξής:* **Κυκλική κίνηση της βάσης:** *value=T1\**f(timer)= 2msec*19.531KHz=19.531*
      7. **Κυκλική κίνηση της λεπίδων:** *value=T2\**f(timer)= 1msec\*19.531KHz=39.062
      8. f(timer)=19.531Khz και Ν=1024.
      9. Για την πιο γρήγορη προσομοίωση της άσκησης χρησιμοποιήθηκαν παρόμοιοι χρόνοι και συγκεκριμένα οι χρόνοι 40,20, 10 που δόθηκαν παραπάνω.