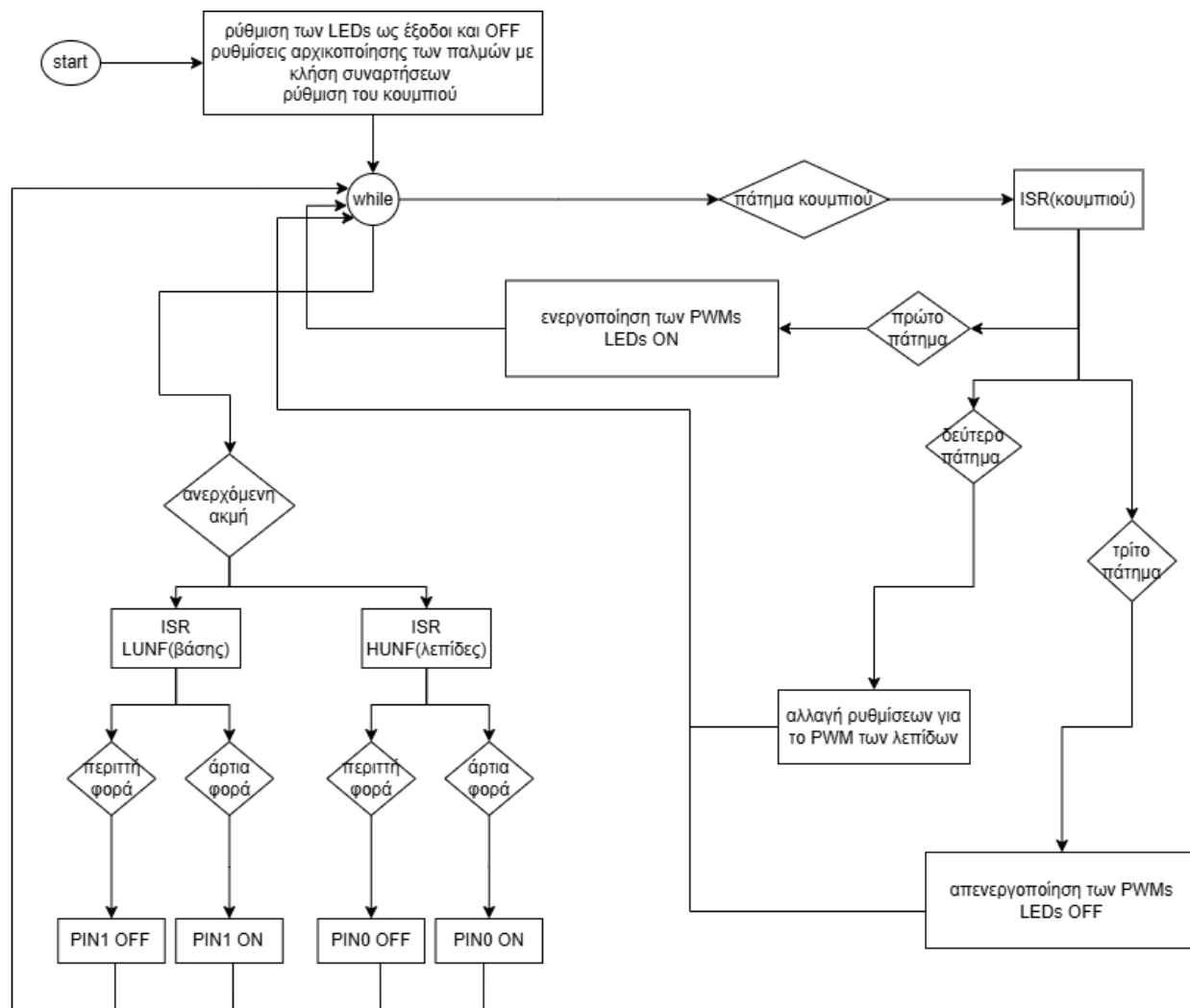


Χαράλαμπος Κωνσταντακόπουλος, 1090059

Εβελίνα Σενή, 1080416

**Διάγραμμα ροής:****Κώδικας:**

```

#include <avr/io.h>
#include <avr/interrupt.h>

int y = 0; //irrelevant
int base = 0;
int lepidex = 0;
int value1;
int value2;

```

```

int button_times = 0; //number of times the button has been pushed

void PWM_base(){

    TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm; //split mode

    // base (Low timer)
    TCA0.SPLIT.CTRLB = TCA_SPLIT_LCMP0EN_bm;
    TCA0.SPLIT.LPER = 125; // Periodos 2ms
    TCA0.SPLIT.LCMP0 = 50; // 40% duty cycle

    TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV16_gc;

    // enable interrupts
    TCA0.SPLIT.INTCTRL = TCA_SPLIT_LUNF_bm;
}

void PWM_lepides(value1, value2){

    // lepides (High timer)
    TCA0.SPLIT.CTRLB |= TCA_SPLIT_HCMP0EN_bm;
    TCA0.SPLIT.HPER = value1; // Periodos
    TCA0.SPLIT.HCMP0 = value2; // duty cycle

    // enable interrupts
    TCA0.SPLIT.INTCTRL |= TCA_SPLIT_HUNF_bm;
}

int main(){
    PORTD.DIR |= PIN0_bm | PIN1_bm; //pins are output
    PORTD.OUT |= PIN0_bm | PIN1_bm; //LED0 and LED1 are off

    //pullup enable and Interrupt enabled with sense on both edges
    PORTF.PIN6CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc;

    PWM_base();
    PWM_lepides (62,37);

    sei();
    while(y==0){
        ;
    }
    cli();
}

ISR(PORTF_PORT_vect){
    cli();
    int y = PORTF.INTFLAGS; //Procedure to
    PORTF.INTFLAGS=y; //clear the interrupt flag

    if(button_times==0){
        // Πρώτο πάτημα: Ενεργοποίηση PWM
        TCA0.SPLIT.CTRLA |= TCA_SPLIT_ENABLE_bm;

        PORTD.OUTCLR= PIN0_bm | PIN1_bm; //LEDs are on
        button_times=1;
    }
    else if(button_times==1){

```

```

        PWM_lepides(31, 16);
        button_times=2;
    }
    else if(button_times==2){
        TCA0.SPLIT.CTRLA &= ~TCA_SPLIT_ENABLE_bm; //disable timer
        PORTD.OUT |= PIN0_bm | PIN1_bm; //LED0 and LED1 are off
        button_times=0; //reset button
    }
    sei();
}

ISR(TCA0_LUNF_vect){
    cli();

    TCA0.SPLIT.INTFLAGS = TCA_SPLIT_LUNF_bm; // clear interrupt flag

    if(base==0){
        PORTD.OUT |= PIN1_bm; //off
        base = 1;
    }
    else if(base==1){
        PORTD.OUTCLR= PIN1_bm; //on
        base = 0;
    }

    sei();
}

ISR(TCA0_HUNF_vect){
    cli();

    TCA0.SPLIT.INTFLAGS = TCA_SPLIT_HUNF_bm; // clear interrupt flag

    if(lepides==0){
        PORTD.OUT |= PIN0_bm; //off
        lepides = 1;
    }
    else if(lepides==1){
        PORTD.OUTCLR= PIN0_bm; //on
        lepides = 0;
    }

    sei();
}

```

### **Αναφορά:**

Ο κώδικας αυτός υλοποιεί τη λειτουργία ενός ανεμιστήρα που προσομοιώνεται με δύο PWM παλμούς, χρησιμοποιώντας τον Timer TCA0 του μικροελεγκτή σε Split Mode. Οι δύο PWM παλμοί αντιπροσωπεύουν δύο διαφορετικές κυκλικές κινήσεις: 1) της βάσης και 2) των λεπίδων του ανεμιστήρα, και η κατάσταση αυτών των σημάτων εμφανίζεται μέσω δύο LED (LED0 και LED1), συνδεδεμένων στο PORTD. Για την εκτέλεση του προγράμματος ορίζουμε τις μεταβλητές base, η οποία χρησιμοποιείται στην ISR(TCA0\_LUNF\_vect) ως flag για το LED της βάσης, lepides η οποία χρησιμοποιείται στην ISR(TCA0\_HUNF\_vect) ως flag για το LED των λεπίδων, value1 & value2, οι οποίες

χρησιμοποιούνται για να ορίσουμε τις επιθυμητές τιμές στον παλμό για τις λεπίδες και τέλος `button_times`, η οποία χρησιμοποιείται ως μετρητής πατημάτων του SW6. Αρχικά ορίζουμε δύο εξόδους στο PORTD, την PIN0 για τις λεπίδες, την PIN1 για την βάση και τις αρχικοποιούμε στην κατάσταση 1 (LED's OFF) . Στη συνέχεια, ορίζεται ο διακόπτης Switch 6, ο οποίος αντιστοιχεί στο PIN6 του PORTF. Ο Timer TCA0 χωρίζεται σε δύο 8-bit Timers, λόγω της εφαρμογής του σε Split mode. Ο LCNT χρησιμοποιείται για να προσομοιώσει την περιστροφή της βάσης με PWM περίοδο 2 ms και duty cycle 40%, ενώ ο HCNT προσομοιώνει την περιστροφή των λεπίδων αρχικά με PWM περίοδο 1 ms και duty cycle 60%. Για την εκκίνηση της λειτουργίας, οι αντίστοιχες παράμετροι ρυθμίζονται μέσω δύο διαφορετικών συναρτήσεων: `PWM_base()` και `PWM_lepides()`. Η ενεργοποίηση και εναλλαγή των καταστάσεων του PWM και των LED ελέγχεται από τον SW6. Όταν ο διακόπτης πατηθεί για πρώτη φορά, και τα δύο PWM σήματα ενεργοποιούνται και τα LED ανάβουν. Στο δεύτερο πάτημα, το `PWM_lepides()` επιταχύνεται, με περίοδο που γίνεται 0.5 ms και duty cycle 50%. Στο τρίτο πάτημα, όλα τα PWM σήματα απενεργοποιούνται, τα LED σβήνουν και το σύστημα επιστρέφει στην αρχική του κατάσταση. Τα interrupts των `TCA0_LUNF_vect` και `TCA0_HUNF_vect` ενεργοποιούνται κάθε φορά που ο αντίστοιχος Timer φτάσει στο underflow (μηδενίζεται). Στο interrupt της βάσης το LED1 εναλλάσσει την κατάστασή του (ON/OFF) κάθε φορά που γίνεται underflow ο Low timer, προσομοιώνοντας έτσι τη σταθερή, περιοδική κυκλική κίνηση της βάσης του ανεμιστήρα. Αντίστοιχα, στο interrupt των λεπίδων το LED0 εναλλάσσει την κατάστασή του (ON/OFF) κάθε φορά που συμβαίνει underflow στον High timer, προσομοιώνοντας τη γρήγορη κυκλική κίνηση των λεπίδων του ανεμιστήρα. Στον συγκεκριμένο κώδικα δεν απαιτείται η προσθήκη επιπλέον καθυστέρησης μεταξύ των παλμών PWM, καθώς η κλήση της `PWM_base()` προηγείται της `PWM_lepides()` και εξασφαλίζεται με αυτό τον τρόπο ο απαραίτητος χρόνος ώστε να μην εκτελεστούν ταυτόχρονα. Με τον τρόπο αυτό, ο κώδικας εκτελείται σωστά και παίρνουμε τα επιθυμητά αποτελέσματα.

Απαραίτητες πράξεις που εκτελέστηκαν:

Έχουμε  $FCPU = 1 \text{ MHz}$  και  $Prescaler = 16$  . Οπότε για τον υπολογισμό της συχνότητας και της περιόδου του Timer θα είναι:

$$F_{timer} = FCPU / Prescaler = 1 \text{ MHz} / 16 = 62.5 \text{ kHz}$$

$$T_{tick} = 1 / f_{timer} = 1 / 62.5 \text{ kHz} = 16 \mu\text{s}$$

- Περίοδος  $T = 2 \text{ ms}$  για την κίνηση της βάσης

$$LPER = 2 \text{ ms} / 16 \mu\text{s} = 2000 / 16 = 125$$

- Duty cycle 40% άρα:

$$LCMP0 = 40\% \times LPER = 0.4 \times 125 = 50$$

Άρα θα έχουμε ότι:

**LPER = 125**

**LCMP0 = 50**

### 1° Πάτημα Κουμπιού

- Περίοδος  $T = 1 \text{ ms}$  για τις λεπίδες.

Υπολογισμός HPER:

$1 \text{ ms} / 16 \mu\text{s} = 1000 / 16 = 62.5 \Rightarrow \text{HPER} = 62$   
(στρογγυλοποιούμε προς τα κάτω γιατί ο Timer μετράει ακέραιες τιμές)

- Duty Cycle 60% άρα:

$$\text{HCMP0} = 60\% \times \text{HPER} = 0.6 \times 62 = 37.2 \Rightarrow \text{HCMP0} = 37$$

Άρα για την πρώτη φάση του παλμού των λεπίδων πρέπει να έχουμε:

**HPER = 62 (value1)**

**HCMP0 = 37 (value2)**

### 2° Πάτημα Κουμπιού

- Περίοδος υποδιπλασιάζεται για τις λεπίδες. Αν στην πρώτη φάση είχαμε HPER = 62, τότε:

$$\text{HPER (νέο)} = 62 / 2 = 31$$

- Duty Cycle 50%, άρα:

$$\text{HCMP0 (νέο)} = 50\% \times 31 = 15.5 \approx 16$$

Άρα για τη δεύτερη φάση του παλμού των λεπίδων πρέπει να έχουμε:

**HPER = 31 (value1)**

**HCMP0 = 16 (value2)**