



Τμήμα Μηχανικών Η/Υ και Πληροφορικής  
Πανεπιστήμιο Πατρών  
Πολυτεχνική Σχολή

Τομέας Υλικού και Αρχιτεκτονικής των Υπολογιστών

Διδάσκουσα: Μαριλένα Δούναβη

Ακαδημαϊκό Έτος: 2023 – 2024

Ημ/νία Παράδοσης: 02/04/2024

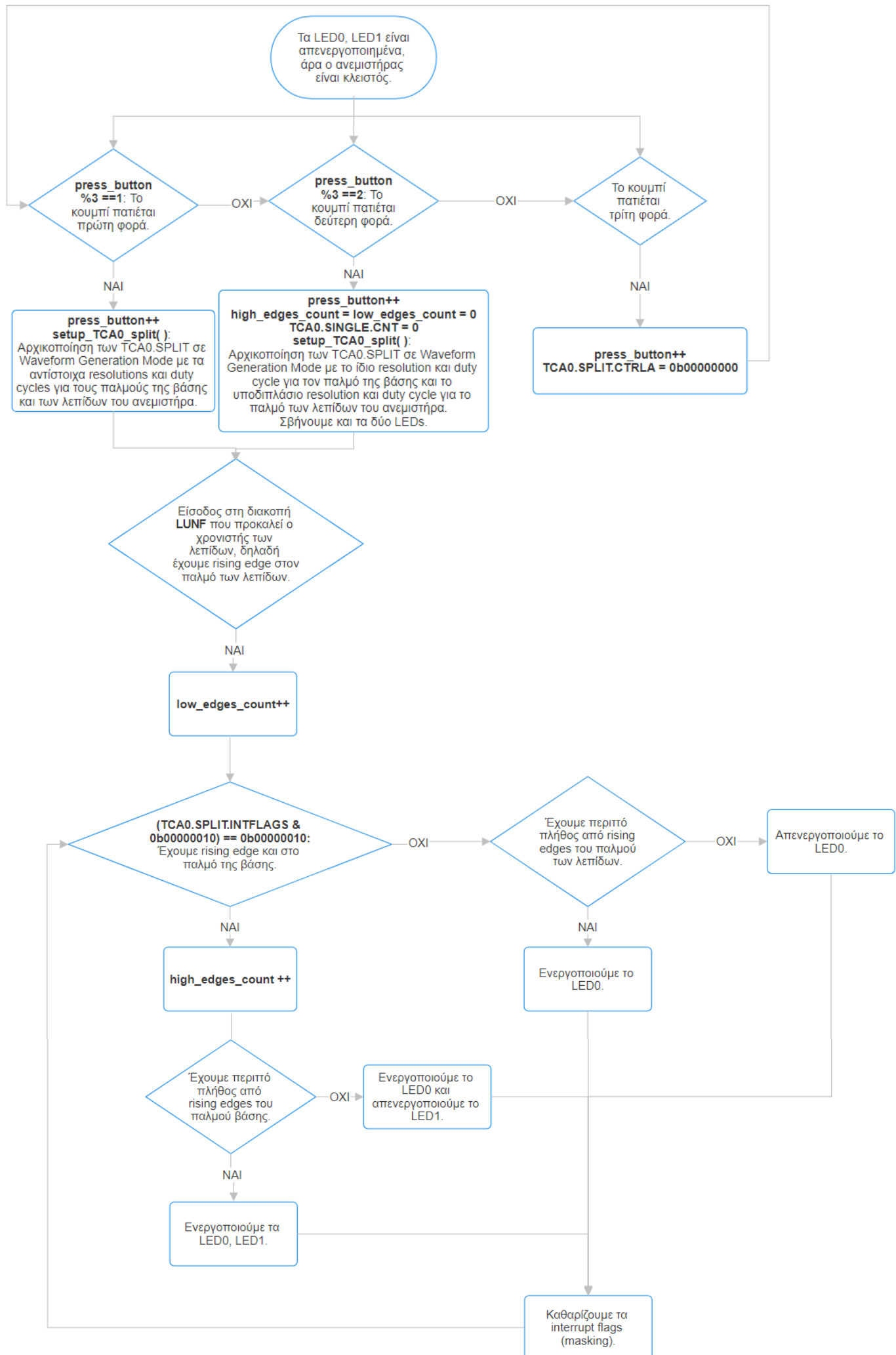
## Εργαστήριο Προηγμένων Μικροϋπολογιστών 3<sup>η</sup> Εργαστηριακή Άσκηση Εξοικείωση με την Πάλμοευρική Διαμόρφωση (Pulse-Width Modulation – PWM)

Μάθημα Κορμού – CEID\_NY463  
Εαρινό Εξάμηνο 2024

Στοιχεία Φοιτητών (ΤΜΗΜΑ Α2):

Ονοματεπώνυμο:	Μηλιτιάδης Μαντές	Χρυσανγή Πατέλη
A.M.:	1084661	1084513
E – mail:	up1084661@ac.upatras.gr	up1084513@ac.upatras.gr
Εξάμηνο:	Η'	Η'

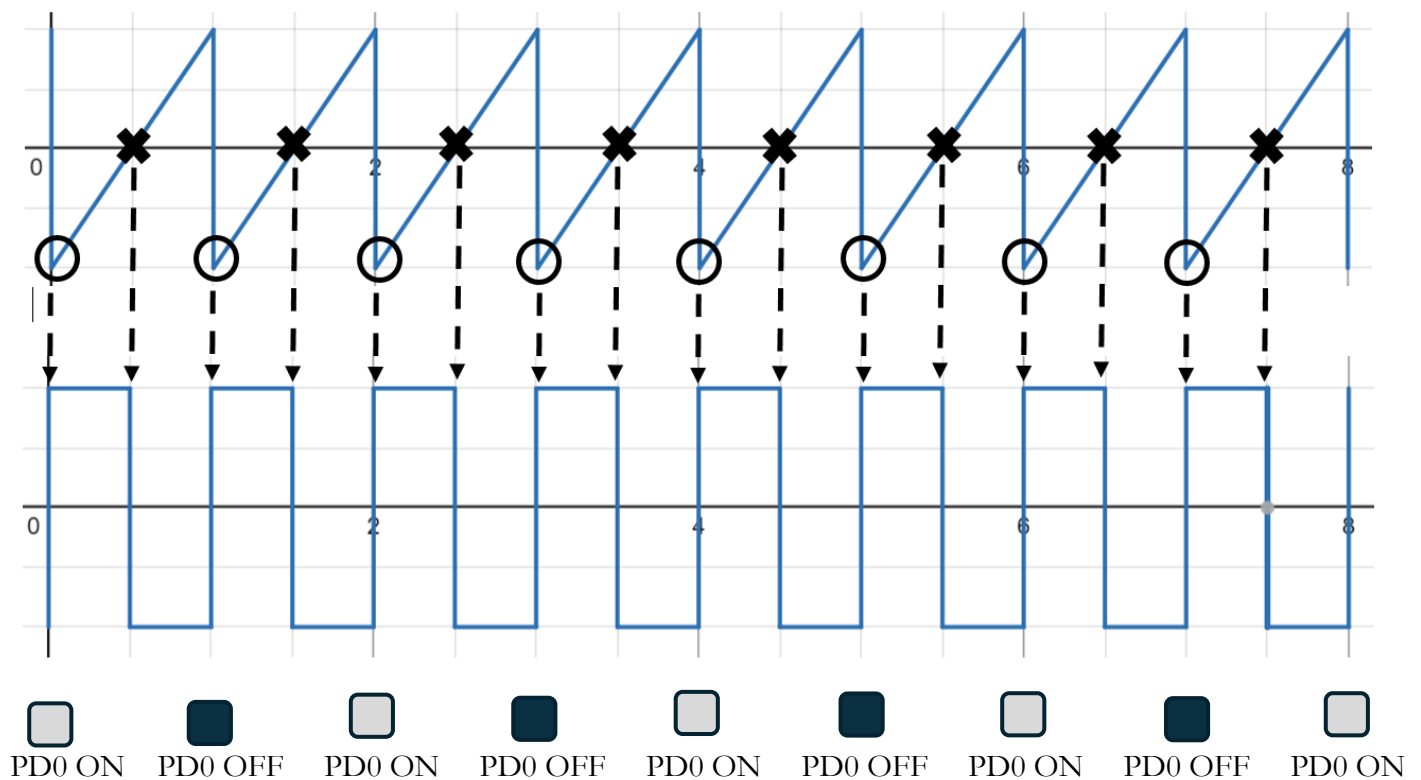
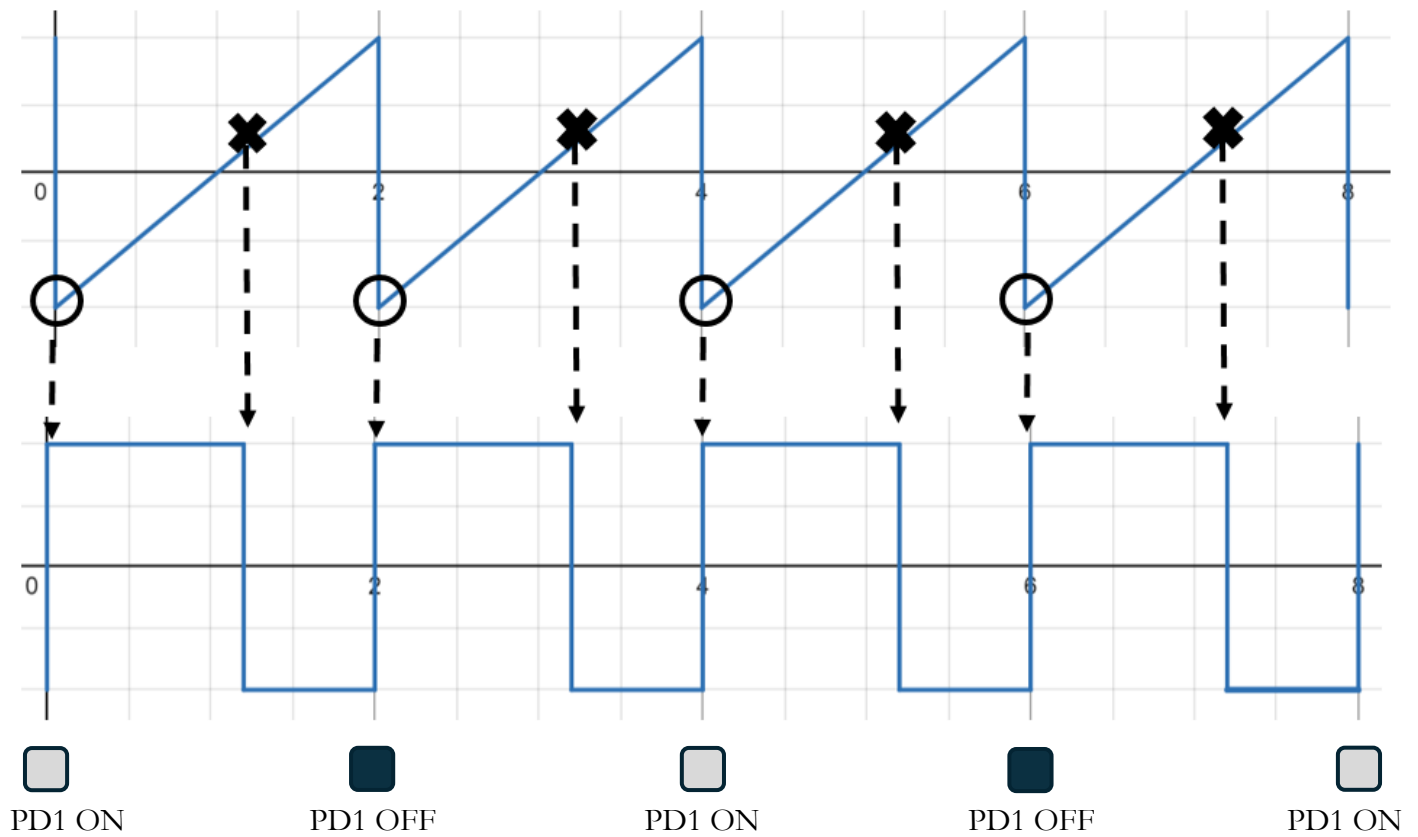
### 3.0 Διάγραμμα Ροής



### 3.1 Σχολιασμός κώδικα

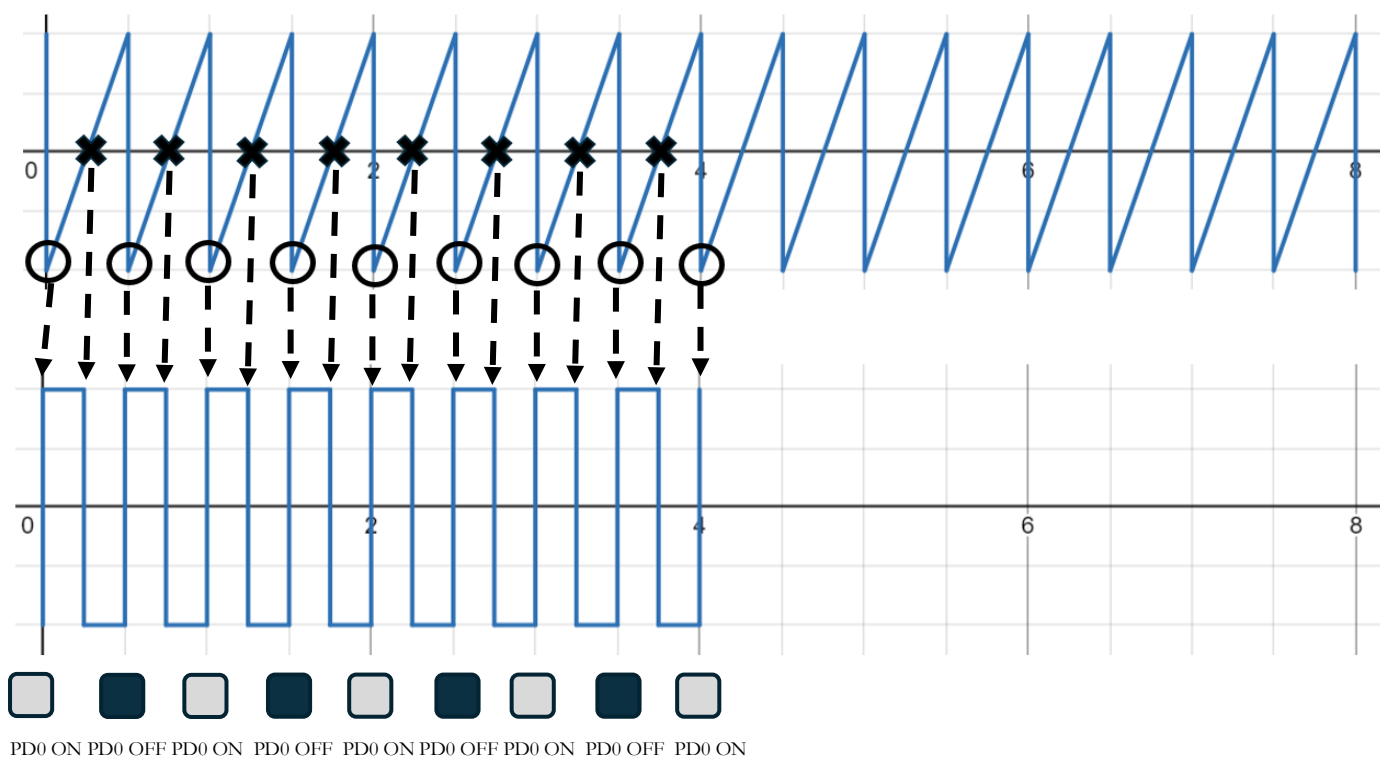
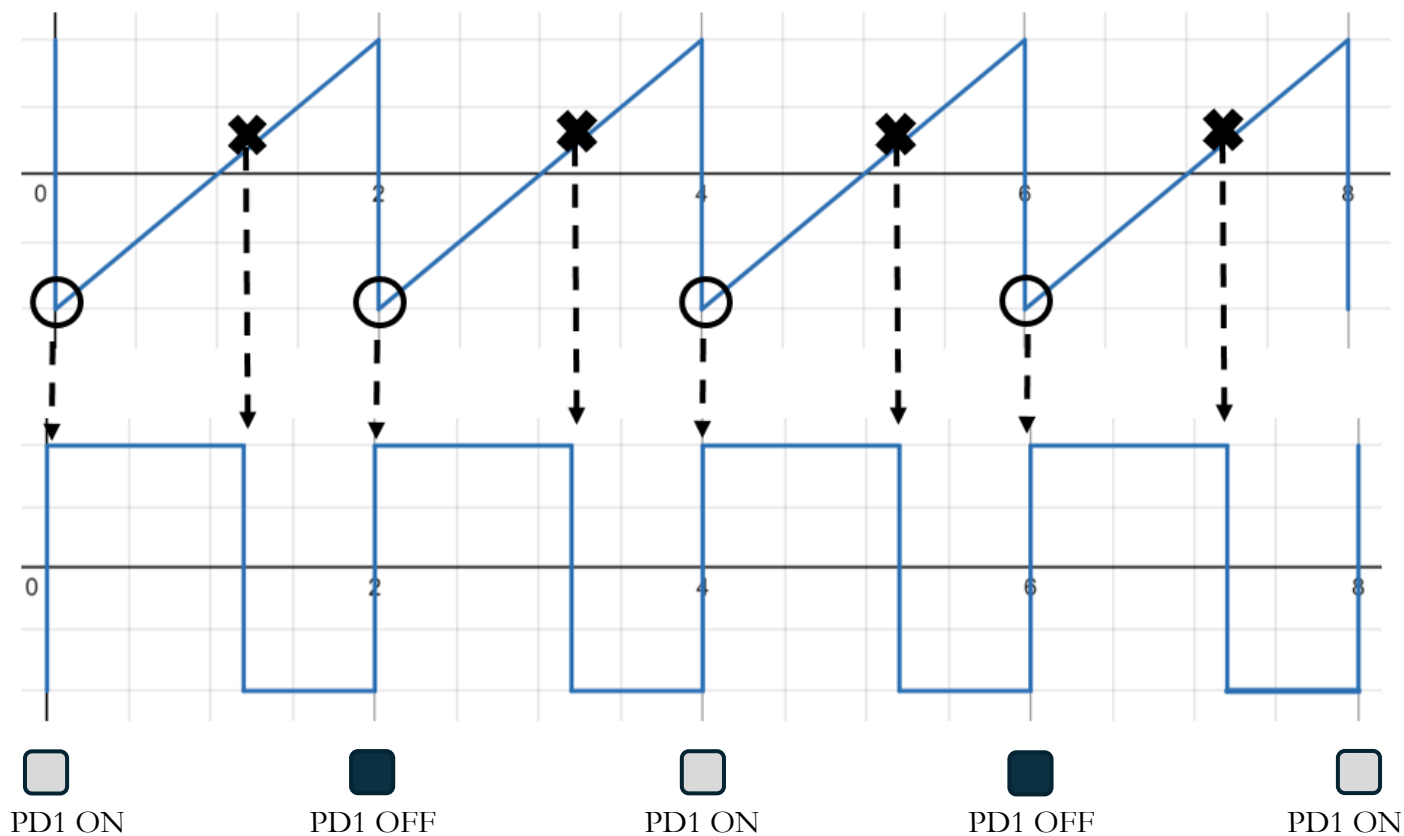
#### Ερώτημα 1°

Παρακάτω εξηγούμε πως προέκυψαν οι ζητούμενες κυματομορφές των παλμών που χρησιμοποιούμε στην εξομοίωση.



## Ερώτημα 2°

Παρακάτω εξηγούμε πως προέκυψαν οι ζητούμενες κυματομορφές των παλμών που χρησιμοποιούμε στην εξομίωση.



Για τον παλμό της βάσης του ανεμιστήρα έχουμε duty cycle 60%, συνεπώς η τιμή του καταχωρητή **HCMP** θα είναι το 60% της τιμής του καταχωρητή **HPER**. Άρα, στη παλμοσειρά που προκύπτει με περίοδο  $T_b = 2 \text{ ms}$  θα έχουμε κατερχόμενη ακμή ανά 1.2 ms. Αντίστοιχα, για τον παλμό των λεπίδων του ανεμιστήρα έχουμε duty cycle 50%, συνεπώς η τιμή του καταχωρητή **LCMP** θα είναι το 50% της τιμής του καταχωρητή **LPER**. Άρα, στη παλμοσειρά που προκύπτει με περίοδο  $T_l = 1 \text{ ms}$  θα έχουμε κατερχόμενη ακμή ανά 0.5 ms. Με “**X**” μαρκάρουμε το σημείο όπου γίνεται ταίριασμα με την αντίστοιχη τιμή του καταχωρητή **CMP**, ενώ με “**O**” κάνουμε ενημέρωση αφού έχει παρέλθει μια περίοδος και προκύπτει ανερχόμενη ακμή.

Ακόμα, για να υπολογίσουμε τη περίοδο της παλμοσειράς κάνουμε τους εξής υπολογισμούς:

- $f_{\text{CLK\_PER}} / N = 20 \text{ MHz} / 1024 = 0.01953125 \text{ MHz} = 19531.25 \text{ Hz}$ , όπου  $f_{\text{CLK\_PER}}$  η συχνότητα του συστήματος και  $N$  η τιμή του prescaler.

Άρα, για τη προσομοίωση των περιόδων και των duty cycles των παλμοσειρών της βάσης και των λεπίδων θα χρησιμοποιήσουμε στο κώδικά μας τις παρακάτω τιμές αντίστοιχα:

### Ερώτημα 1°

- $fb = f_{\text{CLK\_PER}} / N(\text{PER\_high}+1) \Rightarrow 1 / T_b = f_{\text{CLK\_PER}} / N(\text{PER\_high}+1) \Rightarrow 0.5 = 19531.25(1/(\text{PER\_high}+1)) \Rightarrow \text{PER\_high}+1 = 39.0625 \Rightarrow \text{PER\_high} = 38.0625 \Rightarrow \boxed{\text{PER\_high} \approx 40}$
- $fl = f_{\text{CLK\_PER}} / N(\text{PER\_low}+1) \Rightarrow 1 / T_l = f_{\text{CLK\_PER}} / N(\text{PER\_low}+1) \Rightarrow 1 = 19531.25(1/(\text{PER\_low}+1)) \Rightarrow \text{PER\_low}+1 = 19.5312 \Rightarrow \text{PER\_low} = 18.5312 \Rightarrow \boxed{\text{PER\_low} \approx 20}$
- $\text{CMP\_high} = 60\% * \text{PER\_high} = 24$
- $\text{CMP\_low} = 50\% * \text{PER\_low} = 10$

### Ερώτημα 2°

- $fb = f_{\text{CLK\_PER}} / (N(\text{PER\_high}+1)) \Rightarrow 1 / T_b = f_{\text{CLK\_PER}} / (N(\text{PER\_high}+1)) \Rightarrow 0.5 = 19531.25(1/(\text{PER\_high}+1)) \Rightarrow \text{PER\_high}+1 = 39.0625 \Rightarrow \text{PER\_high} = 38.0625 \Rightarrow \boxed{\text{PER\_high} \approx 40}$
- $fl2 = f_{\text{CLK\_PER}} / N(\text{PER\_low2}+1) \Rightarrow 1 / T_{l2} = f_{\text{CLK\_PER}} / N(\text{PER\_low2}+1) \Rightarrow 2 = 19531.25(1/(\text{PER\_low2}+1)) \Rightarrow \text{PER\_low2}+1 = 9.7656 \Rightarrow \text{PER\_low2} = 8.7656 \Rightarrow \boxed{\text{PER\_low2} \approx 10}$
- $\text{CMP\_high} = 60\% * \text{PER\_high} = 24$
- $\text{CMP\_low2} = 50\% * \text{PER\_low2} = 5$

Ξεκινάμε αρχικοποιώντας τις μεταβλητές **PER\_high**, **PER\_low**, **PER\_low2**, **CMP\_high**, **CMP\_low** και **CMP\_low2** με τις κατάλληλες τιμές που υπολογίσαμε παραπάνω. Έπειτα, αρχικοποιούμε με μηδέν όλους τους μετρητές **high\_edges\_count**, **low\_edges\_count** και **press\_button**, καθώς και τη μεταβλητή **x**, η οποία χρησιμοποιείται για να βγούμε από το while loop όταν το κουμπί πατηθεί και τρίτη φορά.

Η συνάρτηση **setup\_TCA0\_split(int PER\_low, int CMP\_low)** χρησιμοποιείται για την αρχικοποίηση των δύο χρονιστών **TCA0.HIGH** και **TCA0.LOW**. Αρχικά, ενεργοποιούμε τον split mode για τον χρονιστή **TCA0**. Στην συνέχεια, το εσωτερικό ρολόι του μικροελεγκτή διαιρείται με παράγοντα 1024 πριν την είσοδο στον χρονιστή. Επιπλέον, ενεργοποιούμε την δημιουργία κυματομορφών. Για τα 8 σημαντικότερα bits του χρονιστή, ορίζουμε την μέγιστη τιμή TOP μέχρι την οποία θα μετρήσει ο παλμός της βάσης και ορίζουμε και τον κύκλο λειτουργίας του παλμού της βάσης. Το ίδιο κάνουμε και για τα 8 λιγότερο σημαντικά bits, ορίζουμε την μέγιστη τιμή TOP μέχρι την οποία θα μετρήσει ο παλμός των λεπίδων και ορίζουμε και τον κύκλο λειτουργίας του παλμού των λεπίδων. Επίσης, ενεργοποιούμε τις διακοπές που προκαλεί το **HCMP0** και **LCMP0**, καθώς και την διακοπή που προκαλεί το **LUNF** κατά την υπερχειλίση του χρονιστή. Τέλος, ενεργοποιούμε τον **TCA0** και ξεκινάμε το ρολόι.

Στη διακοπή **ISR(PORTF\_PORT\_vect)** μεταβαίνουμε κάθε φορά που πατιέται το **SWITCH5** του PORTF. Αρχικά, καθαρίζουμε τις σημαίες διακοπής του PORTF και έπειτα αεζάνουμε το **press\_button** κατά 1. Ανάλογα με τη τρέχουσα τιμή της μεταβλητής μεταβαίνουμε και στον αντίστοιχο βρόχο ελέγχου. Όταν η τιμή του **press\_button** ισούται με 1 όταν βρίσκουμε το mod του με το 3, δηλαδή έχει τις τιμές (1,4,7...) (1ος βρόχος ελέγχου) αρχικοποιούμε μέσω της **setup\_TCA0\_split()** τους δύο χρονιστές **TCA0.HIGH** και **TCA0.LOW** με τις αντίστοιχες τιμές των

**PER** και **CMP** και ξεκινάμε να παράγουμε τις κυματομορφές. Κάθε φορά που συναντάμε ανερχόμενη ακμή στη κυματομορφή του **TCA0.LOW** μεταβαίνουμε και στην αντίστοιχη διακοπή **ISR(TCA0\_LUNF\_vect)**. Όταν η τιμή του **press\_button** ισούται με 2 όταν βρίσκουμε το **mod** του με το 3, δηλαδή έχει τις τιμές (2,5,8...) (2<sup>ος</sup> βρόχος ελέγχου) κάνουμε **restart** το σύστημα αρχικά απενεργοποιούμε και τα δύο pins με την εντολή **PORTD.OUT |= PIN0\_bm | PIN1\_bm**. Στην συνέχεια, μηδενίζουμε τους μετρητές **high\_edges\_count**, **low\_edges\_count** καθώς και το μετρητή **TCA0**, προκειμένου να κάνουμε **restart** της μέτρησης. Έπειτα, αρχικοποιούμε μέσω της **setup\_TCA0\_split( )** ξανά τους δύο χρονιστές **TCA0.HIGH** και **TCA0.LOW** με τις νέες τιμές των **PER** και **CMP** και ξεκινάμε να παράγουμε τις κυματομορφές. Κάθε φορά που συναντάμε ανερχόμενη ακμή στη κυματομορφή του **TCA0.LOW** μεταβαίνουμε όμοια στην αντίστοιχη διακοπή **ISR(TCA0\_LUNF\_vect)**. Τέλος, όταν η τιμή του **press\_button** έχει τις τιμές (3,6,9...) (3<sup>ος</sup> βρόχος ελέγχου) απενεργοποιούμε τον **TCA0** μέσω της εντολής **TCA0.SPLIT.CTRLA = 0b00000000** και απενεργοποιούμε και τα δύο pins με την εντολή **PORTD.OUT |= PIN0\_bm | PIN1\_bm**.

Η διακοπή **ISR(TCA0\_LUNF\_vect)**, ενεργοποιείται κάθε φορά που έχουμε διακοπή από τον καταχωρητή **LUNF**, δηλαδή κάθε φορά που έχουμε ανοδική ακμή στον παλμό των λεπίδων. Κάθε φορά που εισέρχεται το πρόγραμμα στην διακοπή αυξάνουμε το **low\_edges\_count** κατά 1 για να δηλώσουμε σε ποια ανοδική ακμή βρίσκόμαστε. Αν έχει προκύψει διακοπή και από τον καταχωρητή **HUNF**, δηλαδή έχουμε ανοδική ακμή και στον παλμό της βάσης, τότε αυξάνουμε τον μετρητή **high\_edges\_count** κατά 1. Επιπλέον, ελέγχουμε τον μετρητή **high\_edges\_count** αν είναι περιττός αριθμός τότε ανάβουμε και τα δύο LEDs (**LED0**, **LED1**) αλλιώς ανάβουμε το **LED0** και σβήνουμε το **LED1**. Στην περίπτωση που δεν έχει προκύψει διακοπή από το **HUNF** τότε ελέγχουμε τον μετρητή **low\_edges\_count** αν είναι περιττός αριθμός τότε ανάβουμε το **LED0** και το **LED1** παραμένει στην κατάσταση που ήταν, σε διαφορετική περίπτωση σβήνουμε το **LED0** και το **LED1** παραμένει και πάλι στην κατάσταση που ήταν. Τέλος, καθαρίζουμε τις σημαίες διακοπής του **TCA0**.

## 3.2 Κώδικας

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

int PER_high = 40; // Προσομοιώνουμε Tb = 2 ms
int CMP_high = 24; // Προσομοιώνουμε Db = 60% * Tb
int PER_low = 20; // Προσομοιώνουμε Tl = 1 ms
int CMP_low = 10 ; // Προσομοιώνουμε Dl = 50% * Tl
int PER_low2 = 10 ; // Προσομοιώνουμε Tl = 0,5 ms
int CMP_low2 = 5; // Προσομοιώνουμε Dl = 50% * Tl

int high_edges_count = 0; // Μετρητής για την αποθήκευση του πλήθους των rising edges για το παλμό
της βάσης
int low_edges_count = 0; // Μετρητής για την αποθήκευση του πλήθους των rising edges για το παλμό
των λεπίδων
int x=0;
int press_button = 0; // Μετρητής για την αποθήκευση των φορών που πατιέται το κουμπί

void setup_TCA0_split(int PER_low, int CMP_low){
    TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm; // Ενεργοποιούμε το Split Mode για τον TCA0
    TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV1024_gc; // Χρήση παράγοντα διαίρεσης 1024
    TCA0.SPLIT.CTRLB |= TCA_SINGLE_WGMODE_SINGLESLOPE_gc; // Ενεργοποίηση δημιουργίας
κυματομορφών

    TCA0.SPLIT.HPER = PER_high; // Ορίζουμε τη μέγιστη τιμή TOP μέχρι την οποία θα μετρήσει ο
παλμός της βάσης
    TCA0.SPLIT.HCMP0 = CMP_high; // Ορίζουμε το κύκλο λειτουργίας του παλμού της βάσης

    TCA0.SPLIT.LPER = PER_low; // Ορίζουμε τη μέγιστη τιμή TOP μέχρι την οποία θα μετρήσει ο
παλμός των λεπίδων
    TCA0.SPLIT.LCMP0 = CMP_low; // Ορίζουμε το κύκλο λειτουργίας του παλμού των λεπίδων

    TCA0.SPLIT.CTRLB = TCA_SPLIT_HCMP0EN_bm | TCA_SPLIT_LCMP0EN_bm; // Ενεργοποιούμε τη διακοπή
που προκαλεί το HCMP0 και το LCMPO
    TCA0.SPLIT.INTCTRL |= TCA_SPLIT_LUNF_bm; // Ενεργοποιούμε τη διακοπή που προκαλεί το LUNF
κατά την υπερχειλίση του χρονιστή
    TCA0.SPLIT.CTRLA |= TCA_SPLIT_CLKSEL_DIV1024_gc | TCA_SPLIT_ENABLE_bm; // Ενεργοποιούμε τον
TCA0 και ξεκινάμε το ρολόι
}

int main(void)
{
    PORTD.DIR = PIN0_bm | PIN1_bm; // Ορίζουμε τα pins 0, 1 ως εξόδους
    PORTD.OUT = PIN0_bm | PIN1_bm; // Αρχικά και τα δύο pin είναι σβηστά
    PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc; // Ορίζουμε το pin 5 ως είσοδο
sei(); // Ξεκινάμε να δεχόμαστε σήματα διακοπής

    while (x==0)
    {
        ;
    }
    cli(); // Σταματάμε να δεχόμαστε σήματα διακοπής
}

// Διακοπή για το χειρισμό του πατήματος του κουμπιού
ISR(PORTF_PORT_vect){
    int y = PORTF.INTFLAGS; // Καθαρίζουμε τις σημαίες διακοπής
    PORTF.INTFLAGS=y;
    press_button++ ; // Αυξάνουμε τις φορές που πατήθηκε το κουμπί κατά 1

    if(press_button%3==1){
        setup_TCA0_split(PER_low, CMP_low); // Αν το κουμπί πατηθεί 1η φορά αρχικοποιούμε τον
TCA0 με τις αντίστοιχες τιμές
    }
    else if(press_button%3==2){
        PORTD.OUT = PIN0_bm | PIN1_bm; // Σβήνουμε και τα δύο pins
    }
}
```

```

        high_edges_count = 0; // Αρχικοποιούμε και τους δύο μετρητές πάλι με 0 για να κάνουμε
restart
        low_edges_count = 0;
        TCA0.SINGLE.CNT = 0; // Καθαρίζουμε τον TCA0
        setup_TCA0_split(PER_low2, CMP_low2); // Αν το κουμπί πατηθεί 2η φορά αρχικοποιούμε
τον TCA0 με τη καινούρια τιμή του PER_low
    }
    else{
        TCA0.SPLIT.CTRLA = 0b00000000; // Αν το κουμπί πατηθεί 3η φορά απενεργοποιούμε τον
TCA0
        PORTD.OUT |= PIN0_bm | PIN1_bm; // Σβήνουμε και τα δύο pins
    }
}

// Διακοπή για το χειρισμό ανερχόμενης ακμής από τη κυματομορφή των low bits του TCA0 (λεπίδες)
ISR(TCA0_LUNF_vect) {
    low_edges_count++; // Αυξάνουμε το πλήθος των ανερχόμενων ακμών των λεπίδων κατά 1
    if ((TCA0.SPLIT.INTFLAGS & 0b00000010)== 0b00000010) // Ελέγχουμε αν έχουμε ανερχόμενη ακμή
από τη κυματομορφή των high bits του TCA0 (βάση)
    {
        high_edges_count++; // Αυξάνουμε το πλήθος των ανερχόμενων ακμών της βάσης κατά 1
        if(high_edges_count%2==1){ // Ελέγχουμε αν έχουμε περιττό πλήθος ανερχόμενων ακμών της
βάσης

                PORTD.OUTCLR = PIN0_bm | PIN1_bm; // Ανάβουμε και τα δύο pins
            }
            else{
                PORTD.OUT |=PIN1_bm; // Σβήνουμε το pin 1
                PORTD.OUTCLR = PIN0_bm; // Ανάβουμε το pin 0
            }
        }
        else{
            if(low_edges_count%2==1){ // Ελέγχουμε αν έχουμε περιττό πλήθος ανερχόμενων ακμών των
λεπίδων
                PORTD.OUTCLR = PIN0_bm ; // Ανάβουμε το pin 0
            }
            else{
                PORTD.OUT |= PIN0_bm; // Σβήνουμε το pin 0
            }
        }






        int intflags = TCA0.SPLIT.INTFLAGS; // masking
        TCA0.SPLIT.INTFLAGS = intflags;
    }
}

```












### 3.3 Παράδειγμα χρήσης


1. Πατάμε το κουμπί πρώτη φορά. Η εναλλαγή των pins φαίνεται πιο κάτω για χρονικό διάστημα από 0 – 4 ms:

 OUT	0x464	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x02	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x03	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
 OUT	0x464	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

2. Πατάμε το κουμπί δεύτερη φορά και οι δύο μετρητές κάνουν restart. Η εναλλαγή των pins φαίνεται πιο κάτω για χρονικό διάστημα από 0 – 4 ms:

 OUT	0x464	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x02	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x03	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
 OUT	0x464	0x02	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
 OUT	0x464	0x03	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
 OUT	0x464	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

3. Πατάμε το κουμπί τρίτη φορά και οι δύο μετρητές απενεργοποιούνται και τα pins σβήνουν.

 OUT	0x464	0x03	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
---	-------	------	---