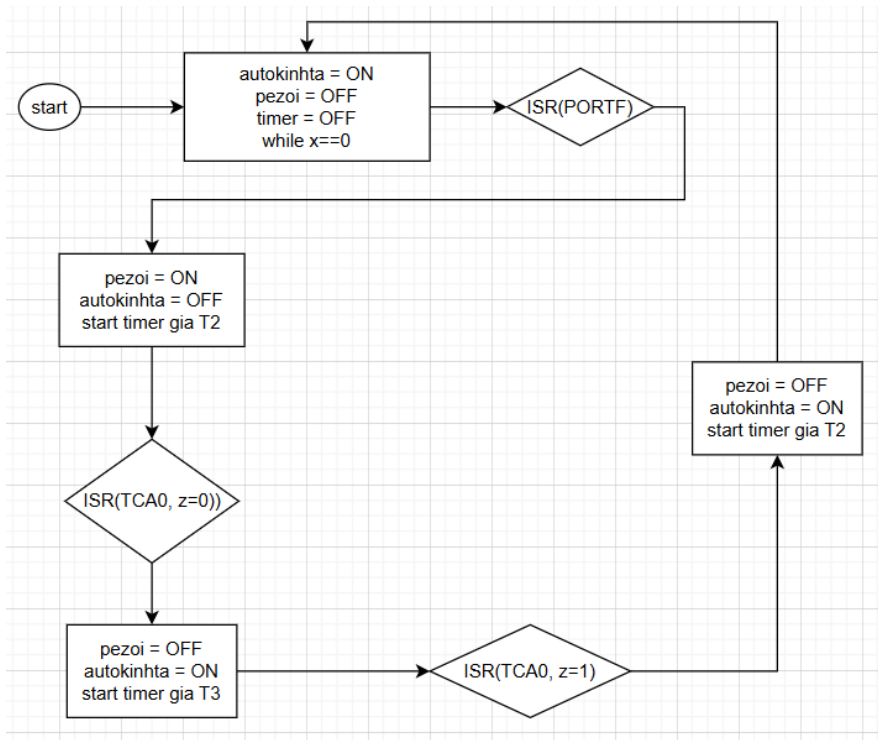


Χαράλαμπος Κωνσταντακόπουλος, 1090059

Εβελίνα Σενή, 1080416

### Ερώτημα 1:

#### Διάγραμμα ροής:



#### Κώδικας:

```

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

#define t2 = 20
#define t3 = 30

int x = 0; //irrelevant
int z = 0;

int main() {

    PORTD.DIR |= PIN0_bm; //PIN0 is output (pezoi)
    PORTD.DIR |= PIN2_bm; //PIN2 is output (autokinhta)
    
```

```

    PORTD.OUTCLR= PIN2_bm; //LED2 ON    ypothetoume oti ksekiname me prasino gia ta
autokinhtha
    PORTD.OUT |= PIN0_bm; //LED0 OFF kai kokkino gia tous pezous

    PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc; // koumpi pezwn

    TCA0.SINGLE.CNT = 0; //clear counter
    TCA0.SINGLE.CTRLB = 0; //Normal Mode
    TCA0.SINGLE.CMP0 = t2; //When CMP0 reaches this value -> interrupt
    TCA0.SINGLE.CTRLA = 0x7<<1; //CLOCK_FREQUENCY/1024
    TCA0.SINGLE.INTCTRL = TCA_SINGLE_CMP0_bm; //Interrupt Enable (=0x10)

    sei();
    //begin accepting interrupt signals
    while (x==0) {
        ;
    }

    cli();
}

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

    PORTD.OUTCLR= PIN0_bm; //pezoi pernoun
    PORTD.OUT |= PIN2_bm; //autokinhtha stamatoun

    TCA0.SINGLE.CTRLA |= 1; //Enable timer gia na metrhsei t2
    sei();
}

ISR(TCA0_CMP0_vect){
    cli();

    // Disable timer TCA0(an to kanw iso me to miden tha svisei kai th rithmisi tou
prescaler)
    TCA0.SINGLE.CTRLA &= ~TCA_SINGLE_ENABLE_bm;
    int intflags = TCA0.SINGLE.INTFLAGS;
    TCA0.SINGLE.INTFLAGS=intflags;

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

    if(z==0)
    {
        PORTD.OUTCLR= PIN2_bm; //LED2 ON
        PORTD.OUT |= PIN0_bm; //LED0 OFF
        //unable to koumpi gia na min patitheis oso einai anenergi i diavasi
        PORTF.PIN5CTRL = 0;
        TCA0.SINGLE.CMP0 = t3;
        //enable gia na metrhsei to xrono pou prepei na einai anenergi i diavasi
        TCA0.SINGLE.CTRLA |= 1;
        z = 1;
    }else if(z==1)
    {

```

```

        int y = PORTF.INTFLAGS; //Procedure to
        PORTF.INTFLAGS=y;      //clear the interrupt flag
        PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc; //enable to
koumpi
        TCA0.SINGLE.CMP0 = t2;
        z=0;
    }
    sei();
}

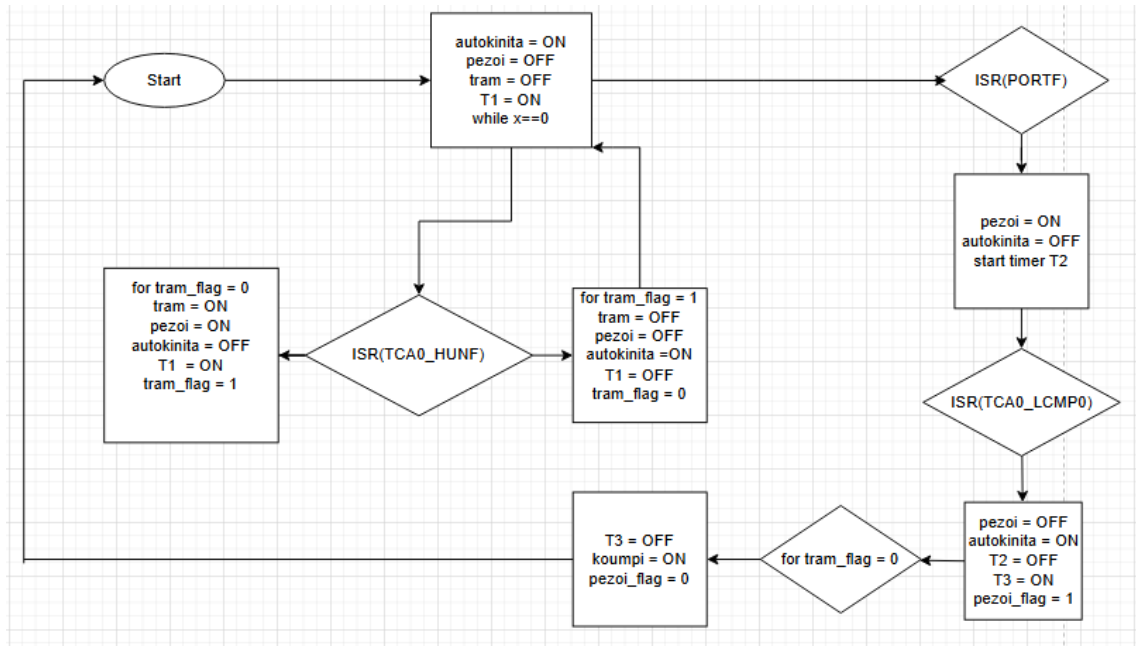
```

### **Αναφορά:**

Κατά την έναρξη λειτουργίας του προγράμματος, υποθέτουμε ότι το φανάρι των αυτοκινήτων είναι πράσινο, ενώ των πεζών κόκκινο. Επίσης, ενεργοποιείται το PIN5 του PORTF ώστε να χρησιμοποιηθεί ως κουμπί ενεργοποίησης διάβασης πεζών. Κατά την κανονική λειτουργία, το πρόγραμμα εισέρχεται σε κατάσταση αναμονής (while loop), όπου περιμένει το πάτημα του πλήκτρου από τους πεζούς. Όταν πατηθεί το κουμπί, το πρόγραμμα «μπάνει» στην `ISR(PORTF_PORT_vect)`. Εντός αυτής της διακοπής, ο κώδικας ανάβει το φανάρι των πεζών και ταυτόχρονα σβήνει των αυτοκινήτων. Ταυτόχρονα, ενεργοποιείται ο timer TCA0 ώστε να ξεκινήσει να μετράει το διάστημα διάβασης πεζών (t2). Όταν ο timer φτάσει στην προκαθορισμένη τιμή του (CMP0=t2), προκαλείται νέα διακοπή, και ενεργοποιείται η ρουτίνα διακοπής `ISR(TCA0_CMP0_vect)`. Εκεί, το πρόγραμμα σταματά τον timer και μηδενίζει τον counter. Εάν η τιμή της z είναι 0 (δηλαδή μόλις ολοκληρώθηκε η διέλευση των πεζών), το πρόγραμμα ανάβει το φανάρι των αυτοκινήτων και σβήνει των πεζών. Ταυτόχρονα, απενεργοποιείται προσωρινά το πλήκτρο των πεζών (ώστε να μην πατηθεί εκ νέου), θέτει τη νέα τιμή για τον timer ίση με t3 και επανενεργοποιεί τον timer, προκειμένου να μετρήσει το χρονικό διάστημα αναμονής πριν να επιτρέψει ξανά τη διέλευση των πεζών. Η μεταβλητή z λαμβάνει πλέον τιμή ίση με 1. Όταν ο timer μετρήσει ξανά και φτάσει πάλι στην προκαθορισμένη τιμή (CMP0 = t3), ενεργοποιείται ξανά η ίδια διακοπή `ISR(TCA0_CMP0_vect)`. Αυτή τη φορά, εφόσον η τιμή της z είναι πλέον 1, το πρόγραμμα ενεργοποιεί ξανά τη δυνατότητα πατήματος του κουμπιού πεζών και ορίζει τη νέα τιμή του timer ίση με t2, επαναφέροντας τη μεταβλητή z στην τιμή 0. Στην συνέχεια επιστρέφει στη while loop μέχρι να πατηθεί ξανά το κουμπί και να επαναληφθεί η παραπάνω διαδικασία.

## Ερώτημα 2:

Διάγραμμα ροής:



**Κώδικας:**

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

int y = 0; //irrelevant
int pezo_i_flag = 0;
int tram_flag = 0;

int main() {

    PORTD.DIR |= PIN0_bm ;
    PORTD.DIR |= PIN1_bm ; // arxikopoihsh tw n fanariwn
    PORTD.DIR |= PIN2_bm ;

    PORTD.OUTCLR = PIN2_bm; // autokinhta ON
    PORTD.OUT |= PIN0_bm ; // pezo_i OFF
    PORTD.OUT |= PIN1_bm ; // tram OFF

    PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc; // koumpi pezw n

    TCA0.SPLIT.CTRLD = TCA_SPLIT_SPLITM_bm; // timer se split mode
    TCA0.SPLIT.CTRLB = TCA_SPLIT_LCMP0EN_bm | TCA_SPLIT_HCMP0EN_bm;

    TCA0.SPLIT.HCNT = 0; // timer tram
    TCA0.SPLIT.HPER = 254; // kathe poso pernaei to tram

    TCA0.SPLIT.INTCTRL = TCA_SPLIT_HUNF_bm; //interrupt enable
```

```

TCA0.SPLIT.CTRLA = TCA_SPLIT_CLKSEL_DIV1024_gc | TCA_SPLIT_ENABLE_bm; //enable
timer

sei();

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

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

    if (tram_flag == 0){
        PORTD.OUTCLR = PIN0_bm; // pezoi ON
        PORTD.OUT |= PIN2_bm;   // autokinhtha OFF
        TCA0.SPLIT.LCNT = 0;    // reset counter
        TCA0.SPLIT.CTRLB = TCA_SPLIT_LCMP0EN_bm;

        TCA0.SPLIT.LCNT = 0;
        TCA0.SPLIT.LPER = 50;   // timer pezwon //den kserw an xreiazetai
        TCA0.SPLIT.LCMP0 = 20;  // xronos meta to patima tou koumpiou
        TCA0.SPLIT.INTCTRL |= TCA_SPLIT_LCMP0_bm;
        PORTF.PIN5CTRL = 0;
    }
    sei();
}

ISR(TCA0_LCMP0_vect){
    cli();
    int y = PORTF.INTFLAGS; //Procedure to
    PORTF.INTFLAGS=y; //clear the interrupt flag

    /*TCA0.SPLIT.CTRLA &= ~TCA_SPLIT_ENABLE_bm; // Disable Timer*/
    TCA0.SPLIT.INTFLAGS = TCA_SPLIT_LCMP0_bm; // clear interrupt flag

    TCA0.SPLIT.LCNT = 0; // reset counter (gia na metrhsei twra to diastima pou exoun
gia na perasoun)

    if(tram_flag == 0){
        if(pezoi_flag == 0){
            PORTD.OUTCLR = PIN2_bm; // Αυτοκίνητα ON
            PORTD.OUT |= PIN0_bm;   // Πεζοί OFF
            pezoi_flag = 1;
        }
    }
    sei();
}

ISR(TCA0_HUNF_vect){
    cli();

```

```

TCA0.SPLIT.INTFLAGS = TCA_SPLIT_HUNF_bm; // clear interrupt flag
TCA0.SPLIT.HCNT = 0; // reset counter

if(tram_flag == 0){

    PORTD.OUTCLR = PIN1_bm | PIN0_bm; // pezoι kai tram ON
    PORTD.OUT |= PIN2_bm; // Αυτοκίνητα OFF
    PORTF.PIN5CTRL = 0;
    TCA0.SPLIT.HPER = 20; // 20 sec pernaei tram
    tram_flag = 1;
}
else if(tram_flag == 1){

    PORTF.PIN5CTRL = 0;
    PORTD.OUT |= PIN0_bm | PIN1_bm; // pezoι kai tram OFF
    PORTD.OUTCLR = PIN2_bm; // Αυτοκίνητα ON
    PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc; // enable
koumpi pezwon
    TCA0.SPLIT.HPER = 100; // mexri na perasei pali to tram
    tram_flag = 0;
    TCA0.SPLIT.LCNT = 0;
}
sei();
}

```

#### **Αναφορά:**

Κατά την έναρξη του προγράμματος, υποθέτουμε ότι το φανάρι των αυτοκινήτων είναι πράσινο (PORTD2 = 0), ενώ τα φανάρια των πεζών (PORTD0) και του τραμ (PORTD1) είναι κόκκινα (PORTD0 = 1, PORTD1 = 1). Παράλληλα, ενεργοποιείται το PIN5 του PORTF ώστε να χρησιμοποιηθεί ως κουμπί διάβασης για πεζούς. Ο χρονοδιακόπτης TCA0 τίθεται σε λειτουργία split mode, ώστε να παρέχει δύο ανεξάρτητους χρονιστές : τον HCNT/HPER για τη διέλευση του τραμ και τον LCNT/LPER για τους πεζούς. Ο χρονιστής του τραμ (HCNT) ξεκινά να μετράει με αρχική τιμή HPER = 254, που αντιστοιχεί στο διάστημα μέχρι την πρώτη διέλευση του τραμ. Όταν ο μετρητής ξεπεράσει την τιμή αυτή, προκαλείται η διακοπή HUNF και ενεργοποιείται η ρουτίνα ISR(TCA0\_HUNF\_vect), η οποία διαχειρίζεται την είσοδο και έξοδο του τραμ. Στην συνέχεια εισέρχεται σε κατάσταση αναμονής (while loop), όπου περιμένει το πάτημα του πλήκτρου από τους πεζούς. Όταν πατηθεί το κουμπί, το πρόγραμμα «μπαίνει» στην ISR(PORTF\_PORT\_vect). Αν δεν διέρχεται τραμ εκείνη τη στιγμή (tram\_flag = 0), τότε το πρόγραμμα σβήνει το φανάρι των αυτοκινήτων (PORTD2 = 1) και ανάβει το φανάρι των πεζών (PORTD0 = 0). Στη συνέχεια, γίνεται μηδενισμός του μετρητή LCNT και ορίζονται οι τιμές του LPER και του CMP0, οι οποίες καθορίζουν το χρονικό διάστημα κατά το οποίο οι πεζοί μπορούν να διασχίσουν τον δρόμο. Η διακοπή LCMPO ενεργοποιείται και το κουμπί απενεργοποιείται προσωρινά για να αποφευχθούν επανεπιλημμένα πατήματα. Όταν λήξει το χρονικό διάστημα που ορίστηκε για τους πεζούς, καλείται η ρουτίνα ISR(TCA0\_LCMPO\_vect). Εφόσον το τραμ δεν είναι ενεργό, το φανάρι των πεζών σβήνει (PORTD0 = 1) και το φανάρι των αυτοκινήτων ανάβει ξανά (PORTD2 = 0). Η μεταβλητή pezoι\_flag γίνεται 1 για να δηλώσει ότι η διέλευση των πεζών ολοκληρώθηκε. Η ρουτίνα ISR(TCA0\_HUNF\_vect) εκτελείται περιοδικά βάσει του χρονισμού HPER και είναι υπεύθυνη για τη διαχείριση του τραμ. Όταν δεν διέρχεται τραμ (tram\_flag = 0), η ρουτίνα ενεργοποιεί το φανάρι του τραμ (PORTD1 = 0) και ταυτόχρονα το φανάρι των πεζών (PORTD0 = 0), ενώ απενεργοποιεί των αυτοκινήτων (PORTD2 = 1). Το κουμπί των πεζών απενεργοποιείται και το HPER τίθεται σε 20, ώστε να οριστεί ο χρόνος διέλευσης του τραμ. Η μεταβλητή tram\_flag γίνεται 1. Όταν το τραμ ολοκληρώσει τη διέλευσή του (στην επόμενη διακοπή HUNF), η ρουτίνα σβήνει τα φανάρια του τραμ και των πεζών (PORTD1 = 1, PORTD0 = 1),

ανάβει το φανάρι των αυτοκινήτων ( $PORTD2 = 0$ ), επανενεργοποιεί το κουμπί των πεζών και θέτει νέο χρονισμό  $HPER = 100$  μέχρι την επόμενη διέλευση του τραμ. Η μεταβλητή `tram_flag` μηδενίζεται ξανά και ο κύκλος συνεχίζεται.