Εργαστήριο Προηγμένοι Μικροεπεξεργαστές

**Εργαστηριακή Άσκηση 1**

**Στοιχεία Ομάδας :**

**Ταμβάκης Θωμάς-Χρυσοβαλάντης, ΑΜ:1072631**

**Διαμαντόπουλος Ιωάννης , ΑΜ:1064281**

**Ομάδα : Β3**

**Ερώτημα 1.α - Κώδικας για τη λειτουργία των φαναριών του αυτοκινητόδρομου και των πεζών**

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#define ped\_time\_t2 10

#define ped\_time\_t3 20

int x=0;

int main() {

PORTD.DIR |= 0b00000101; //PIN is output

PORTD.OUT |= 0b00000101; //LED for people and cars is off

while(1){

x=0;

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

sei(); //enable interrupts

while (x==0) {

PORTD.OUTCLR= 0b00000100; //LED for cars is on

}

cli(); //disable interrupts

x=0;

PORTD.OUT |= 0b00000100; //LED for cars is off

PORTD.OUTCLR= 0b00000001; //LED for people is on

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

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

TCA0.SINGLE.CMP0 = ped\_time\_t2; //When 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)

PORTF.PIN5CTRL = 0b00000000; //Do not go again to the ISR if button is pressed

sei(); //begin accepting interrupt signals

while (x==0) {

}

cli();

PORTD.OUT |= 0b00000001; //LED for people is off

PORTD.OUTCLR= 0b00000100; //LED for cars is on

x=0;

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

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

TCA0.SINGLE.CMP0 = ped\_time\_t3; //When 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)

PORTF.PIN5CTRL = 0b00000000; //Do not go again to the ISR if button is pressed

sei(); //begin accepting interrupt signals

while (x==0) {

}

cli();

x=0;

}

}

ISR(PORTF\_PORT\_vect){ //ISR if button is pressed

//clear the interrupt flag

int y = PORTF.INTFLAGS;

PORTF.INTFLAGS=y;

x=1;

}

ISR(TCA0\_CMP0\_vect){ //ISR for Timer T2 and Timer T3

TCA0.SINGLE.CTRLA = 0; //Disable

//clear flag

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS=intflags;

x=1;

}

**Ερώτημα 1.β - Κώδικας για τη λειτουργία των φαναριών του αυτοκινητόδρομου, των πεζών και του τραμ**

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#define tram 0xFD

#define t2\_t3 0xFE

#define start 0xFF

int x=0; //logic flag

int t = 0;

int main() {

PORTD.DIR |= 0b00000111; //PIN is output

PORTD.OUT |= 0b00000111; //LED for people and cars is off

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc;

TCA0.SPLIT.CTRLA |= 1;

TCA0.SPLIT.CTRLD = TCA\_SPLIT\_ENABLE\_bm;

TCA0.SPLIT.LCMP0 = t2\_t3; //(T2/T3)

TCA0.SPLIT.HCMP0 = tram; //(T1)

while(1){

//initialize timer for tram

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_HCMP0EN\_bm;

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_HUNF\_bm;

TCA0.SPLIT.HCNT = start; //restart counter

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

x=0;

sei(); //enable interrupts

while(t==0){

//initialize timer for cars and people

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm;

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_LUNF\_bm;

while (x==0) {

PORTD.OUTCLR= 0b00000100; //LED for cars is on

}

cli(); //disable interrupts

x=0;

PORTD.OUT |= 0b00000100; //LED for cars is off

PORTD.OUTCLR= 0b00000001; //LED for people is on

TCA0.SPLIT.LCNT = start; //clear counter, set FF

TCA0.SPLIT.LCMP0 = t2\_t3; //When reaches this value -> interrupt CLOCK FREQUENCY/1024

PORTF.PIN5CTRL = 0b00000000; //Do not go again to the ISR if button is pressed

sei(); //begin accepting interrupt signals

while (x==0) {

}

cli();

PORTD.OUT |= 0b00000001; //LED for people is off

PORTD.OUTCLR = 0b00000100; //LED for cars is on

x=0;

TCA0.SPLIT.LCNT = start; //clear counter, set FF

TCA0.SPLIT.LCMP0 = t2\_t3; //When reaches this value -> interrupt CLOCK FREQUENCY/1024

PORTF.PIN5CTRL = 0b00000000; //Do not go again to the ISR if T3 has no passed

sei(); //begin accepting interrupt signals

while (x==0) {

}

cli();

x=0;

}

PORTD.OUT |= 0b00000100; //LED for cars is off

PORTD.OUTCLR = 0b00000011; //LED for people and tram is on

}

}

ISR(PORTF\_PORT\_vect){ // ISR for button interrupt

//clear the interrupt flag

int y = PORTF.INTFLAGS;

PORTF.INTFLAGS=y;

x=1;

}

ISR(TCA0\_LUNF\_vect){ // ISR for T2-T3

TCA0.SPLIT.CTRLA = 0; //Disable

//clear flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS=intflags;

x=1;

}

ISR(TCA0\_HUNF\_vect){ // ISR for T1

TCA0.SPLIT.CTRLA = 0; //Disable

//clear flag

int intflags = TCA0.SPLIT.INTFLAGS;

TCA0.SPLIT.INTFLAGS=intflags;

t=1;

}

**Ερώτημα 2.α – Επεξήγηση κώδικα και διάγραμμα ροής.**

Σχετικά με τον κώδικα του ερωτήματος 1.α, αρχικά ορίζουμε με την εντολή *PORT.DIR* τα bit που θα χρησιμοποιήσουμε, δηλαδή bit\_0 = 1 (led πεζών), bit\_2 = 1 (led αυτοκινητόδρομου). Στη συνέχεια με την εντολή PORT.OUT απενεργοποιούμε τα δύο προαναφερθέντα led, θεωρώντας ότι τα φανάρια μόλις τοποθετήθηκαν στο συγκεκριμένο σημείο και είναι στο στάδιο του reset. Με την εντολή *PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc* χάρη στο switch 5 του PORTF ενεργοποιείται η PULL UP αντίσταση. Αφού το σύστημα επιτρέπει την αποδοχή των interrupts με την εντολή *sei()*, όσο ο πεζός δεν πατάει το κουμπί για να ενεργοποιηθεί το φανάρι του (πράσινο) και να απενεργοποιηθεί το φανάρι του αυτοκινητόδρομου (κόκκινο), το φανάρι για τα αυτοκίνητα παραμένει ανοιχτό (πράσινο). Σε περίπτωση που ο πεζός πατήσει το κουμπί (στο PORTF, στο INTFLAGS, το 5ο bit) το σύστημα μάς μεταφέρει στην ISR, το x από 0 γίνεται 1, με αποτέλεσμα το πρόγραμμα να εξέρχεται της while loop που είχε μπει και να απενεργοποιεί τη δυνατότητα αποδοχής των interrupts. Το φανάρι των αυτοκινήτων απενεργοποιείται (κόκκινο), ενώ των πεζών ενεργοποιείται (πράσινο). Παράλληλα, έχει ξεκινήσει ο Timer, ο οποίος θα μετράει μέχρι να φτάσει την τιμή που έχουμε κάνει define για το ped\_time, ενώ, με την εντολή *PORTF.PIN5CTRL = 0b00000000* σε περίπτωση που ο πεζός ξανά πατήσει το κουμπί το σύστημα δεν το λαμβάνει υπόψιν και περιμένει μέχρι να περάσει το υπολειπόμενο χρονικό διάστημα που είναι αναμμένο το φανάρι των πεζών. Ομοίως, για το χρονικό διάστημα (Τ3), ο πεζός, πρέπει να περιμένει για να ενεργοποιηθεί το φανάρι του από τη στιγμή που άναψε ξανά το πράσινο φανάρι για τα αυτοκίνητα αξιοποιώντας την εντολή *PORTF.PIN5CTRL = 0b00000000*. Και οι δύο αυτές λειτουργίες, γίνονται ενεργοποιώντας πάλι την δυνατότητα για να δέχεται το σύστημα interrupts. Το πρόγραμμα, στις δύο άνωθεν περιπτώσεις (Timer για T2 και Timer για Τ3) μεταφέρεται στην *ISR(TCA0\_CMP0\_vect)*, το x γίνεται 1, επομένως το σύστημα βγαίνει από τη while loop, και απενεργοποιεί τη δυνατότητα αποδοχής interrupts. Σε περίπτωση που σε ένα από τα δύο χρονικά διαστήματα Τ2 και Τ3, έχει πατηθεί το κουμπί, το σύστημα ενεργοποιεί το interrupt και μεταφέρεται στην *ISR(PORTF\_PORT\_vect)* αφότου έχει παρέλθει το χρονικό διάστημα Τ2 και το χρονικό διάστημα Τ3, αντίστοιχα.

Ακολουθεί το διάγραμμα ροής :

Εικόνα που περιέχει διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

**Ερώτημα 2.β – Επεξήγηση κώδικα και διάγραμμα ροής.**

Σχετικά με τον κώδικα, αρχικά ορίζουμε με την εντολή *PORT.DIR* τα bit που θα χρησιμοποιήσουμε, δηλαδή bit\_0 = 1 (led πεζών), bit\_1 = 1 (led τραμ), bit\_2 = 1 (led αυτοκινητόδρομου). Στη συνέχεια με την εντολή PORT.OUT απενεργοποιούμε τα τρία προαναφερθέντα led, θεωρώντας ότι τα φανάρια μόλις τοποθετήθηκαν στο συγκεκριμένο σημείο και είναι στο στάδιο του reset. Με την εντολή *PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc* χάρη στο switch 5 του PORTF ενεργοποιείται η PULL UP αντίσταση. Επιπρόσθετα, με τις εντολές *TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc;*

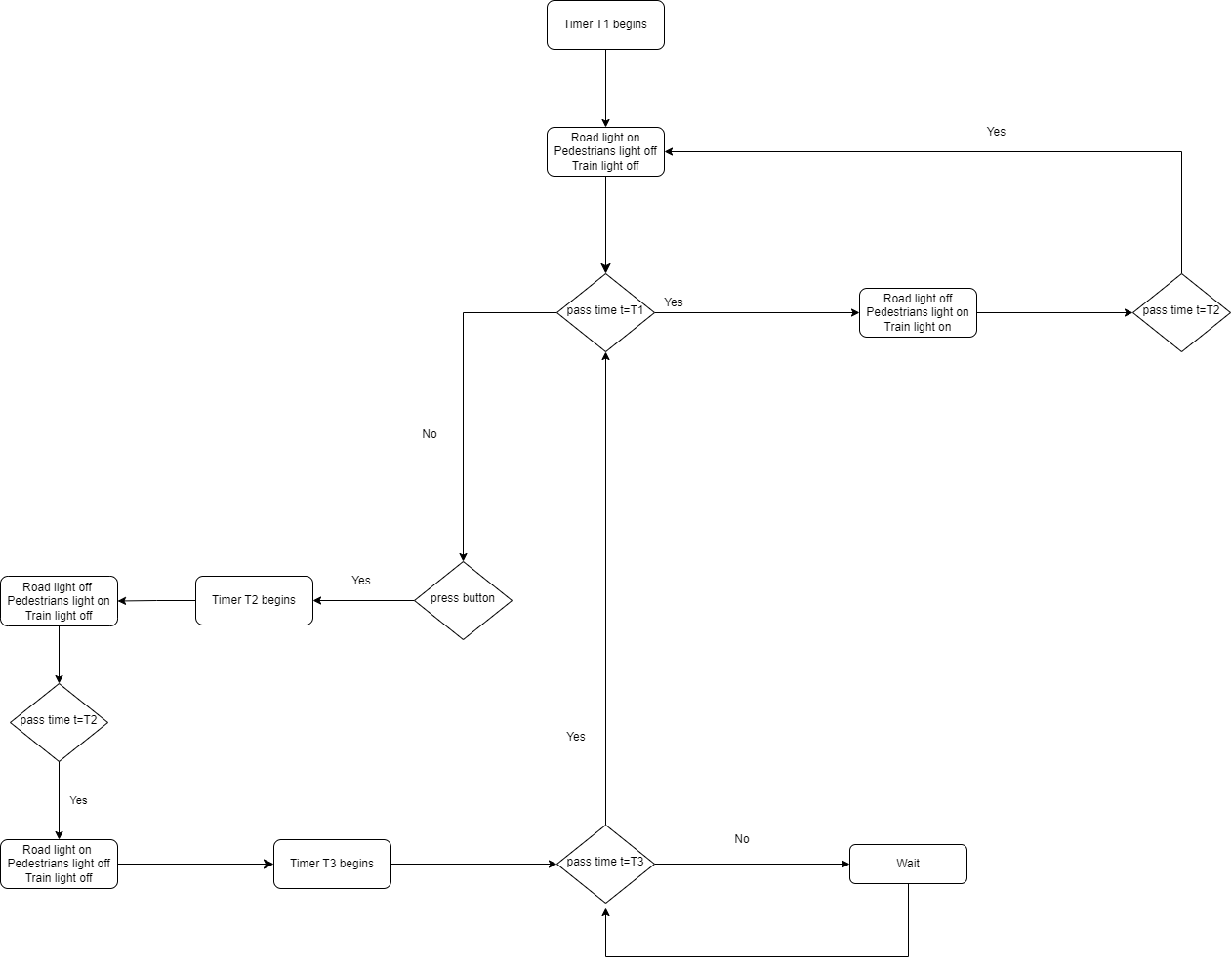
*TCA0.SPLIT.CTRLA |= 1;*

*TCA0.SPLIT.CTRLD = TCA\_SPLIT\_ENABLE\_bm;*

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

*TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_HUNF\_bm | TCA\_SPLIT\_LUNF\_bm;*

χωρίζουμε σε δύο 8 bit μετρητές HUNF και LUNF, τον μετρητή TCA0, ο οποίος είναι των 16 bit. Με την εντολή *TCA0.SPLIT.LCMP0 = t2\_t3* θέτουμε την τιμή που αφορά τα χρονικά διαστήματα T1 και Τ2, ενώ με την εντολή *TCA0.SPLIT.HCMP0 = tram* καθορίζουμε την τιμή που αφορά το χρονικό διάστημα T3, δηλαδή κάθε πότε θα περνάει το τραμ. Σύμφωνα με το manual, όταν κάνουμε Split τον μετρητή αυτό, οι δύο 8 Bit μετρητές που προκύπτουν, μετράνε αντίθετα, δηλαδή η αρχικοποίηση γίνεται στην τιμή 0xFF, και όχι στην τιμή 0. Για αυτό το λόγο, με την εντολή *TCA0.SPLIT.HCNT = start;* θέτουμε την αρχική τιμή για τον έναν από τους δυο μετρητές, στην περίπτωσή μας για το tram, και η αντίστροφη μέτρηση ξεκινάει, ενώ παράλληλα δίνουμε τη δυνατότητα στο σύστημα να αποδέχεται interrupts. Όσων αφορά τη λογική λειτουργίας των φαναριών γίνονται τα εξής: εάν στο χρονικό διάστημα που έχουμε ορίσει για το τραμ (Τ1 = tram – αναμονή για τραμ), ένας πεζός πατήσει το κουμπί, (στο PORTF, στο INTFLAGS, το 5ο bit) το σύστημα, μας μεταφέρει στην ISR, το x από 0 γίνεται 1, με αποτέλεσμα το πρόγραμμα να εξέρχεται της while loop που είχε μπει και να απενεργοποιεί τη δυνατότητα αποδοχής των interrupts. Το φανάρι των αυτοκινήτων απενεργοποιείται (κόκκινο), ενώ των πεζών ενεργοποιείται (πράσινο). Παράλληλα, έχει ξεκινήσει ο Timer (Τ2), ο οποίος έχει αρχική τιμή FF και μετράει αντίστροφα μέχρι την τιμή που έχουμε ορίσει με τις εντολές *TCA0.SPLIT.LCNT = start* και *TCA0.SPLIT.LCMP0 = t2\_t3* αντίστοιχα. Με την εντολή *PORTF.PIN5CTRL = 0b00000000* σε περίπτωση που ο πεζός πατήσει ξανά το κουμπί το σύστημα δεν το λαμβάνει υπόψιν και περιμένει μέχρι να περάσει το υπολειπόμενο χρονικό διάστημα που είναι αναμμένο το φανάρι των πεζών. Ομοίως, για να μπλοκάρεται το πάτημα του κουμπιού των πεζών αν δεν έχει παρέλθει χρονικό διάστημα Τ3 (απαραίτητο διάστημα αναμονής πεζών για να ξανά πατήσουν το κουμπί) αξιοποιούμε την εντολή *PORTF.PIN5CTRL = 0b00000000*. Και οι δύο αυτές λειτουργίες (Timer για Τ2 και Timer για Τ3), γίνονται ενεργοποιώντας πάλι την δυνατότητα να δέχεται το σύστημα interrupts. Το πρόγραμμα, στις δύο άνωθεν περιπτώσεις μεταφέρεται στην *ISR(TCA0\_LUNF\_vect)*, το x γίνεται 1, επομένως το σύστημα εξέρχεται από τη while, και απενεργοποιεί τη δυνατότητα αποδοχής interrupts. Δηλαδή, αν πατηθεί το κουμπί, το σύστημα ενεργοποιεί το interrupt και μεταφέρεται στην *ISR(PORTF\_PORT\_vect)* μόνο στην περίπτωση που έχει παρέλθει και το χρονικό διάστημα Τ2 και το χρονικό διάστημα Τ3.

Ακολουθεί το διάγραμμα ροής :