**Προηγμένοι Μικροεπεξεργαστές**

**Εργαστηριακή Άσκηση 1**

**Μέλη Ομάδας**

**1ο μέλος: 2ο μέλος:**

**ΟΝΟΜΑ: ΛΟΥΚΑΣ ΟΝΟΜΑ: ΕΥΑΓΓΕΛΟΣ**

**ΕΠΩΝΥΜΟ: ΤΡΙΑΝΤΑΦΥΛΛΟΠΟΥΛΟΣ ΕΠΩΝΥΜΟ: ΜΠΕΝΕΤΟΣ**

**ΕΞΑΜΗΝΟ: 8 ΕΞΑΜΗΝΟ: 10**

**ΑΜ: 1084495 ΑΜ: 1072628**

1) Για το πρώτο ερώτημα υλοποιήσαμε τον κώδικα για ένα πρόγραμμα που εκτελεί την λειτουργία ενός συστήματος οδικής κυκλοφορίας, το οποίο περιλαμβάνει την διέλευση πεζών και αυτοκινήτων. Σε αυτό το σύστημα υπάρχουν δύο διαφορετικές χρονικές καθυστερήσεις. Η Τ2 ενεργοποιείται όταν πατηθεί το κουμπί που είναι για τους πεζούς. Όταν πατηθεί το κουμπί, τότε το LED για τους πεζούς θα ανάψει και το LED για τα αυτοκίνητα θα σβήσει. Αυτή η κατάσταση θα διατηρηθεί για το χρονικό διάστημα Τ2 κι ύστερα θα ανάψει το LED για τα αυτοκίνητα και θα σβήσει για τους πεζούς. Έπειτα θα ξεκινήσει η χρονική διάρκεια Τ3, κατά την οποία, όταν κάποιος πεζός επιχειρήσει να πιέσει το κουμπί, δεν θα ανάψει το LED που του αντιστοιχεί και θα χρειαστεί να περιμένει να περάσει το χρονικό διάστημα Τ3. Επειδή χρησιμοποιούμε single χρονιστή, αξιοποιούμε τον ISR TCA0\_CMP0\_vect. Λαμβάνοντας υπόψιν τις παραπάνω λεπτομέρειες, ο κώδικας θα διαμορφωθεί ως εξής:

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdbool.h>

#define T2 14 //T2, cars' LED off/civilian LED on

#define T3 10 //T3, delay until button can work again

int traffic\_button = 0; //if 1, then the LED of civilians will be on and the cars' will be off

int x = 0; //Flag for reset the ISR

bool aux\_flag = true; //auxiliary flag for the loop

int main()

{

PORTD.DIR|= PIN0\_bm;// Set pin0 as led for civilians

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

PORTD.DIR|= PIN2\_bm;// Set pin2 as led for cars

PORTD.OUT |= PIN2\_bm; //LED for cars is is off

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //Pullup enable bit mask enable bit mask and Sense Both Edges

while(aux\_flag)

{

x = 0;

sei(); //begin accepting interrupt signals

if(traffic\_button == 0)

{

PORTD.OUTCLR = PIN2\_bm; // Turn on the LED for cars

PORTD.OUT |= PIN0\_bm; // Turn off the LED for civilians

}

else

{

PORTD.OUT |= PIN2\_bm; //LED for cars is off

PORTD.OUTCLR= PIN0\_bm; //LED for civilians is on

TCA0.SINGLE.CNT = 0; //Clear Counter

TCA0.SINGLE.CTRLB = TCA\_SINGLE\_WGMODE\_NORMAL\_gc; //Waveform generation mode selection ==> Normal Mode

TCA0.SINGLE.CMP0 = T2; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SINGLE.CTRLA = TCA\_SINGLE\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SINGLE.CTRLA |= 1; //Enable

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; //Compare 0 Interrupt bit mask

while (x==0)

{

; //similar to a nop function

}

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

PORTD.OUTCLR = PIN2\_bm; //LED for cars is off

x = 0;

TCA0.SINGLE.CNT = 0; //Clear Counter

TCA0.SINGLE.CTRLB = TCA\_SINGLE\_WGMODE\_NORMAL\_gc; //Waveform generation mode selection ==> Normal Mode

TCA0.SINGLE.CMP0 = T3; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SINGLE.CTRLA = TCA\_SINGLE\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SINGLE.CTRLA |= 1; //Enable

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; //Compare 0 Interrupt bit mask

while (x==0)

{

; //similar to a nop function

}

}

cli();

}

}

ISR(TCA0\_CMP0\_vect)

{

TCA0.SINGLE.CTRLA = 0; //Disable

int intflags = TCA0.SINGLE.INTFLAGS; //Procedure to

TCA0.SINGLE.INTFLAGS=intflags; //clear interrupt flag

x = 1; //change flag to get out of the loop

}

ISR(PORTF\_PORT\_vect)

{

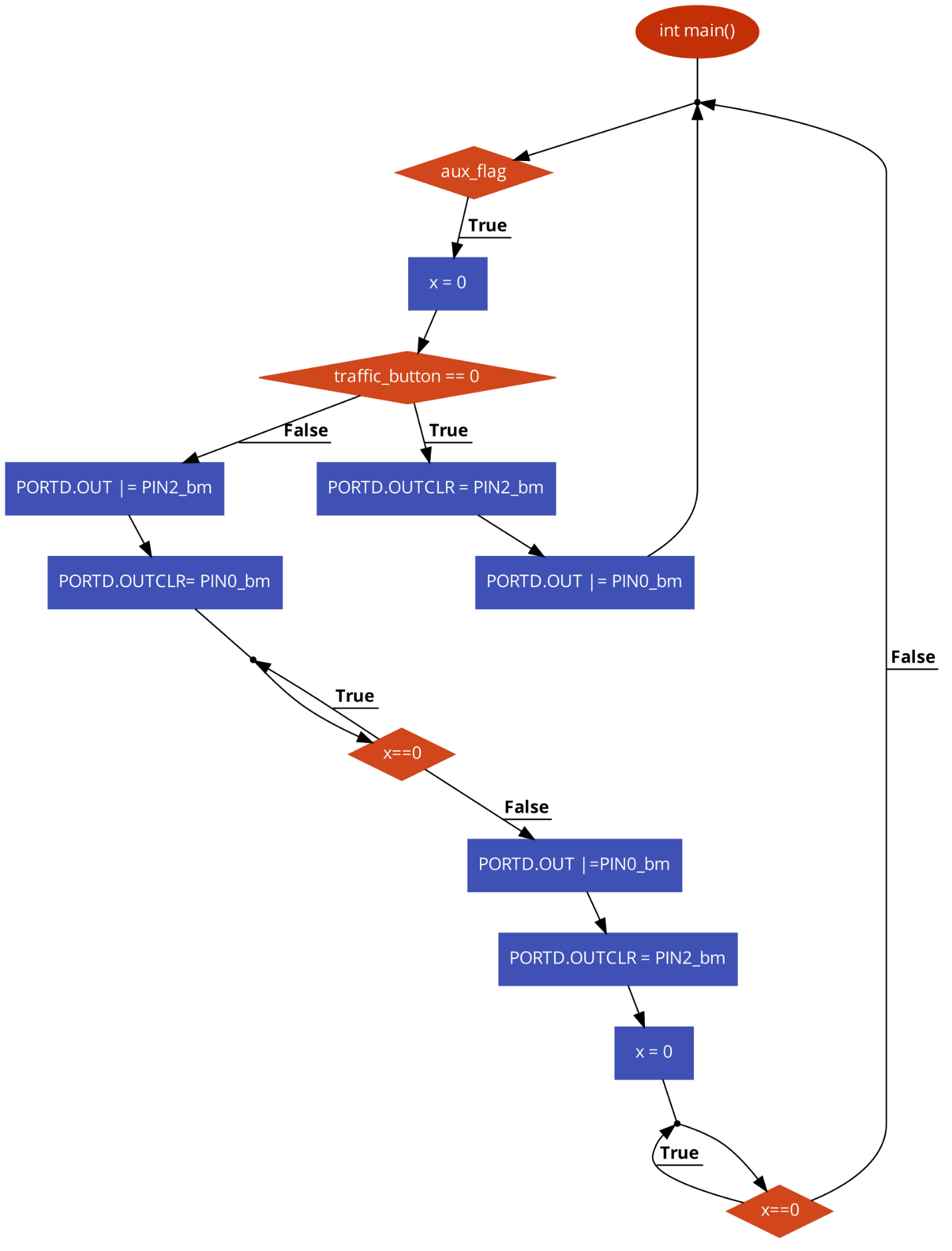
int y = PORTF.INTFLAGS; //Procedure to

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

traffic\_button = 1; //change logic flag to get out of loop

}

Ο παραπάνω κώδικας υλοποιήθηκε με την χρήση 16-bit χρονιστή, τόσο για τα αυτοκίνητα όσο και για τους πεζούς. Στην συνέχεια παραθέτουμε το διάγραμμα ροής που αντιστοιχεί για τον παραπάνω κώδικα.



2) Στην συνέχεια θα προσθέσουμε και το τραμ, το οποίο μας ζητήθηκε για την διεύρυνση του οδικού συστήματος. Για την εκτέλεση του προγράμματος χρησιμοποιήσαμε split χρονιστές (των 8-bit) αντί για single (των 16-bit) διότι θέλαμε να διατηρήσουμε την λογική ότι οι πεζοί θα λάβουν την ίδια μεταχείριση με τα αυτοκίνητα όπως το προηγούμενο ερώτημα και ταυτόχρονα το τραμ να μπορεί να κάνει την διαδρομή ανά τακτά χρονικά διαστήματα Τ1 όπως ζητήθηκε. Συγκεκριμένα βάλαμε ως HCNT τον χρονιστή του τραμ και τον τοποθετήσαμε στην αρχή του κώδικα ώστε να λαμβάνεται πάντα υπόψιν από το πρόγραμμα. Στην συνέχεια, μέσω μεταβλητής Boole (aux\_flag2), εισερχόμαστε στην δομή που αφορά τα αυτοκίνητα και τους πεζούς εάν αυτή είναι ορισμένη ως false. Τα αυτοκίνητα και οι πεζοί χρησιμοποιούν το LCNT. Εάν το κουμπί πατηθεί (INTFLAGS bit5), αυτοκίνητα θα σταματήσουν γιατί έσβησε το LED τους και οι πεζοί θα μπορούν να διασχίσουν το δρόμο γιατί το LED τους άναψε. Έπειτα, αν η aux\_flag2 έχει τιμή true, τότε τα LED των πεζών και του τραμ ανάβουν και των αυτοκινήτων σβήνει για το χρονικό διάστημα Τ2. Μετά το πέρας του Τ2, τα LED των πεζών και του τραμ θα σβήσουν και των αυτοκινήτων θα ανάψει. Επειδή κάναμε χρήση των χρονιστών των 8 bit θα πρέπει να γίνει και χρήση των κατάλληλων ISR (TCA0\_HUNF\_vect και TCA0\_LCMP0\_vect). Με βάση την παραπάνω περιγραφή, ο κώδικας έχει την παρακάτω μορφή:

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdbool.h>

#define T1 12 //T1, train crosses the road periodically

#define T2 14 //T2, cars' LED off/civilian LED on

#define T3 10 //T3, delay until button can work again

int traffic\_button = 0; //if 1, then the LED of civilians will be on and the cars' will be off

int x = 0; //Flag for reset the ISR

bool aux\_flag = true; //auxiliary flag for the inner loop

bool aux\_flag2 = false; //auxiliary flag for the outer loop

int main()

{

PORTD.DIR|= PIN0\_bm;// Set pin0 as led for civilians

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

PORTD.DIR|= PIN1\_bm;// Set pin2 as led for train

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

PORTD.DIR|= PIN2\_bm;// Set pin2 as led for cars

PORTD.OUT |= PIN2\_bm; //LED for cars is is off

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //Pullup enable bit mask enable bit mask and Sense Both Edges

while(aux\_flag)

{

x = 0;

sei(); //begin accepting interrupt signals

//Initialize the split mode for 8-bit counters

//bit masks and bit positions

TCA0.SPLIT.CTRLD |= TCA\_SPLIT\_SPLITM\_bm; //0x01 Split Mode Enable bit mask.

TCA0.SPLIT.HCNT = 0; //Clear Counter

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //0x01 Low Compare 0, 0x10 High Compare 0

TCA0.SPLIT.HCMP0 = T1; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SPLIT.CTRLA |= 1; //Enable

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_HUNF\_bm; //0x02 High Underflow Interrupt Enable bit mask.

while(aux\_flag2 == false)

{

if(traffic\_button == 0)

{

PORTD.OUTCLR = PIN2\_bm; // Turn on the LED for cars

PORTD.OUT |= PIN0\_bm; // Turn off the LED for civilians

}

else

{

PORTD.OUT |= PIN2\_bm; //LED for cars is off

PORTD.OUTCLR= PIN0\_bm; //LED for civilians is on

TCA0.SPLIT.CTRLD |= TCA\_SPLIT\_SPLITM\_bm; //0x01 Split Mode Enable bit mask.

TCA0.SPLIT.LCNT = 0; //Clear Counter

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //0x01 Low Compare 0, 0x10 High Compare 0

TCA0.SPLIT.LCMP0 = T2; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SPLIT.CTRLA |= 1; //Enable

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_LCMP0\_bm; // 0x10 Low Compare 0 Interrupt Enable bit mask.

while (x==0)

{

; //similar to a nop function

}

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

PORTD.OUTCLR = PIN2\_bm; //LED for cars is on

x = 0;

TCA0.SPLIT.CTRLD |= TCA\_SPLIT\_SPLITM\_bm; //0x01 Split Mode Enable bit mask.

TCA0.SPLIT.LCNT = 0; //Clear Counter

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //0x01 Low Compare 0, 0x10 High Compare 0

TCA0.SPLIT.LCMP0 = T3; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SPLIT.CTRLA |= 1; //Enable

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_LCMP0\_bm; // 0x10 Low Compare 0 Interrupt Enable bit mask.

while (x==0)

{

; //similar to a nop function

}

}

}

if(aux\_flag2 == true)

{

PORTD.OUT |= PIN2\_bm; //LED for cars is off

PORTD.OUTCLR= PIN1\_bm; //LED for train is on

PORTD.OUTCLR= PIN0\_bm; //LED for civilians is on

TCA0.SPLIT.CTRLD |= TCA\_SPLIT\_SPLITM\_bm; //0x01 Split Mode Enable bit mask.

TCA0.SPLIT.LCNT = 0; //Clear Counter

TCA0.SPLIT.CTRLB = TCA\_SPLIT\_LCMP0EN\_bm|TCA\_SPLIT\_HCMP0EN\_bm; //0x01 Low Compare 0, 0x10 High Compare 0

TCA0.SPLIT.LCMP0 = T2; //When CMP0 reaches this value, then interrupt will occur ==> CLOCK\_FREQUENCY/1024

TCA0.SPLIT.CTRLA = TCA\_SPLIT\_CLKSEL\_DIV1024\_gc; //Clock Selection mode ==> System Clock/1024

TCA0.SPLIT.CTRLA |= 1; //Enable

TCA0.SPLIT.INTCTRL = TCA\_SPLIT\_LCMP0\_bm; // 0x10 Low Compare 0 Interrupt Enable bit mask.

while (x==0)

{

; //similar to a nop function

}

aux\_flag2 = false;

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

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

PORTD.OUTCLR = PIN2\_bm; //LED for cars is on

x = 0;

}

cli();

}

}

ISR(TCA0\_HUNF\_vect)

{

TCA0.SPLIT.CTRLA = 0; //Disable

int intflags = TCA0.SPLIT.INTFLAGS; //Procedure to

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

aux\_flag2 = true;

}

ISR(TCA0\_LCMP0\_vect)

{

TCA0.SPLIT.CTRLA = 0; //Disable

int intflags = TCA0.SPLIT.INTFLAGS; //Procedure to

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

x = 1; //change flag to get out of the loop

}

ISR(PORTF\_PORT\_vect)

{

int y = PORTF.INTFLAGS; //Procedure to

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

traffic\_button = 1; //change logic flag to get out of loop

}

Με βάση αυτές τις λεπτομέρειες στη συνέχεια παραθέτουμε το διάγραμμα ροής για το διευρυμένο οδικό δίκτυο, το οποίο ακολουθεί την λογική με το προηγούμενο ερώτημα.

