**Προηγμένοι Μικροεπεξεργαστές**

Εργαστηριακή Άσκηση 1

Βασίλειος Ασημακόπουλος – ΑΜ: 1084610

Email: [up1084610@ac.upatras.gr](mailto:up1084610@ac.upatras.gr)

Παντελεήμων Καραγιάννης – ΑΜ: 1084582

Email: [up1084582@ac.upatras.gr](mailto:up1084582@ac.upatras.gr)

Κώδικας εργασίας:

/\*

\* Lab\_ex1.c

\*

\* Created: 15/3/2024 7:58:38 μμ

\* Author : Βασίελιος Ασημακόπουλος, Παντελεήμων Καραγιάννης

\*/

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

void ped\_green();

void car\_green();

void tram\_green();

#define pedTime 20

int PedestInt = 0;

int PedestTime = 0;

int main(void)

{

PORTD.DIR |= PIN0\_bm; //PIN0 is output, fanari0: πεζών

PORTD.DIR |= PIN1\_bm; //PIN1 is output, fanari1: τραμ

PORTD.DIR |= PIN2\_bm; //PIN2 is output, fanari2: αυτοκινητόδρομου

//Αρχικοποίηση φαναριών

PORTD.OUT = PIN0\_bm; //fanari0: κόκκινο

PORTD.OUT = PIN1\_bm; //fanari1: κόκκινο

PORTD.OUTCLR = PIN2\_bm; //fanari2: πράσινο

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; //pullup enable, interrupt with sense

//enabled on both edges

sei(); //enable interrupts

while (1)

{

if(PedestInt == 1)

ped\_green();

car\_green();

tram\_green();

}

cli(); //disable interrupts

}

void ped\_green() {

PORTD.OUTCLR = PIN0\_bm; //fanari0 (πεζών): πράσινο

PORTD.OUTCLR = PIN1\_bm; //fanari1 (τραμ): πράσινο

PORTD.OUT = PIN2\_bm; //fanari2 (αυτοκινητοδρόμου): κόκκινο

//16 bit counter high and low

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

TCA0.SINGLE.CTRLB = 0; //normal mode

TCA0.SINGLE.CMP0 = pedTime; //when reaches this value > interrupt clock\_frequency/1024

TCA0.SINGLE.CTRLA = TCA\_SINGLE\_CLKSEL\_DIV1024\_gc;

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

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; //interrupt enable

while(PedestTime==0) {} //μέχρι να τελειώσει ο counter(να σβήσει το πράσινο των πεζών)

//do nothing

PedestTime = 0; //clear pedTime flag

PedestInt = 0; //clear PedestInt flag

PORTD.OUT = PIN0\_bm; //fanari0 (πεζών): κόκκινο

}

void car\_green() {

PORTD.OUT = PIN0\_bm; //fanari0 (πεζών): κόκκινο

PORTD.OUTCLR = PIN2\_bm; //fanari2 (αυτοκινητόδρομου): πράσινο

if(PedestInt==1)

ped\_green();

}

void tram\_green() {

PORTD.OUTCLR = PIN1\_bm; //fanari1 (τραμ): πράσινο

PORTD.OUT = PIN2\_bm; //fanari2 (αυτοκινήτων): κόκκινο

if(PedestInt==1)

ped\_green();

}

ISR(PORTF\_PORT\_vect) {

//clear interrupt flag

int intflags = PORTF.INTFLAGS;

PORTF.INTFLAGS = intflags;

PedestInt = 1; //set PedestInt flag

}

ISR(TCA0\_CMP0\_vect){

TCA0.SINGLE.CTRLA = 0;

//clear flag

int intflags = TCA0.SINGLE.INTFLAGS;

TCA0.SINGLE.INTFLAGS = intflags;

PedestTime =1;

}

Σχολιασμός κώδικα:

Στον κώδικα μας δεν έχουμε καταφέρει να υλοποιήσουμε των counter TCB, αναλύουμε τον υπόλοιπο κώδικα χωρίς αυτόν. Έχουμε δημιουργήσει 5 συναρτήσεις, 3 από αυτές(ped\_green, car\_green και tram\_green) σχετίζονται με την λειτουργία των φαναριών όπως αυτή ζητείται από την άσκηση και οι άλλες δύο είναι ISR που εξυπηρετούν τον timer και τον interrupt switch. Για αυτές τις δύο συναρτήσεις δηλώσαμε τις μεταβλητές pedestInt και pedestTime και τέλος χρησιμοποιούμε και την σταθερά ped, η οποία αφορά το χρονικό διάστημα του timer. Στο σώμα της main ξεκινάμε αντιστοιχίζοντας τα κατάλληλα pin με τα φανάρια, όπως ζητάει η εκφώνηση και στην συνέχεια ανάβοντας το πράσινο στο φανάρι των αυτοκινήτων. Θέτουμε ως switch interrupt το PIN5 του PORTF, αν το interrupt ενεργοποιηθεί τότε μέσω της αντίστοιχης συνάρτησης λαμβάνει τιμή 1 η μεταβλητή PedestInt, δηλαδή ένας πεζός επιθυμεί να περάσει τον δρόμο. Στο while loop της main γίνεται αρχικά έλεγχος του flag PedestInt, το οποίο αν έχει τιμή 1 εκτελείται η συνάρτηση ped\_green. Η συνάρτηση αυτή αρχικά δίνει στα PIN που αντιστοιχούν στα φανάρια κατάλληλες τιμές που αντιστοιχούν στο πράσινο για τους πεζούς και κόκκινο για τον αυτοκινητόδρομο. Στην συνέχεια αρχικοποιήται και εκτελείται ο TCA0 timer για χρονικό διάστημα που καθορίσαμε μέσω της μεταβλητής ped, το οποίο μόλις τελειώσει η μεταβλητή PedestTime τίθεται μέσω της ISR του timer. Μέσω του while loop ελέγχεται η τιμή της μεταβλητής αυτής και όσο αυτή είναι 0 (δηλαδή όσο τρέχει ο timer) δεν εκτελείται καμία εντολή. Μόλις τελειώσει ο timer η PedestTime λαμβάνει τιμή 1, το πρόγραμμα βγαίνει εκτός του while loop, οι δύο μεταβλητές flag (PedestInt και PedestTime) λαμβάνουν τιμή 0, το φανάρι των πεζών γίνεται κόκκινο , βγαίνουμε από την συνάρτηση και συνεχίζει η εκτέλεση της main. Στην main εκτελούνται οι δύο άλλες συναρτήσεις. Η car\_green στο σώμα της ανάβει το πράσινο για τον αυτοκινητόδρομο και το κόκκινο για το τραμ και στην συνέχεια ελέγχει την μεταβλητή PedestInt, η οποία αν έχει τιμή 1 οδηγεί στην εκτέλεση της συνάρτησης ped\_green, όπως εξηγήθηκε παραπάνω. Με παρόμοιο τρόπο λειτουργεί και η tram\_green όπου στο σώμα της ανάβει το πράσινο των τραμ και το κόκκινο των αυτοκινήτων και στην συνέχεια ελέγχει την μεταβλητή PedestInt.

Τέλος παραθέτουμε το διάγραμμα ροής της άσκησης, το οποίο σχεδιάσαμε με την βοήθεια του εργαλείου draw.io.

Εικόνα που περιέχει κείμενο, διάγραμμα, γραμμή, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα