

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

2 Η ΑΣΚΗΣΗ

16/04/2021

**Παπαβασιλείου Μάριος 1059649**

**Ντάκος Γεώργιος 1059569**

**Ανάλυση κώδικα:**

Στην συγκεκριμένη άσκηση πρέπει να υλοποιούσε 3 φανάρια. Ένα φανάρι για τον μεγάλο δρόμο, 1 φανάρι για τον μικρό δρόμο και 1 φανάρι για τους πεζούς. Αυτό θα το πέτυχουμε χρησιμοποιώντας τα 3 PINS από τα Generic Port Pins(βλέπε iom4808.h) ως εξόδους. Στην συνέχεια αρχικοποιούμε τον counter του timer, τον θέτουμε σε normal mode και δίνουμε στο CPM0 την τιμή την οποία όταν φτάσει ο timer να πραγματοποιηθεί interrupt και να σταματήσει ο timer να μετράει. Έπειτα θέτουμε την συχνότητα με την οποία θα λειτουργεί ο timer μας και ενεργοποιούμε τα interrupts. Στην δικιά μας περίπτωση έχουμε βάλει την τιμή ped = 10 και το Nprescaler = 256 οπότε θα έχουμε :

Ο λόγος που έχουμε βάλει να έχει τόσο μικρό χρόνο ο timer μας είναι διότι στο simulation δεν τρέχουμε σε πραγματικό χρόνο πάνω στην πλακέτα , δηλαδή ο timer δεν θα τελειώσει ποτέ στα 128 ms αλλά μετά από μερικά δεκάδες δευτερόλεπτα. Οπότε οι παραπάνω χρόνοι δεν είναι αντιπροσωπευτικοί σε πραγματικούς χρόνους. Οπότε αν δουλεύαμε πάνω στην πλακέτα και θέλαμε ακριβώς 20 δευτερόλεπτα να λειτουργεί ο timer τότε θα κάναμε τα εξής:

Στην συνέχεια ενεργοποιούμε το interrupt του timer και έπειτα βάζουμε σε μόνιμη κατάσταση το μεγάλο φανάρι να είναι πράσινο. Μετά δημιουργούμε το interrupt για τους πεζούς δηλαδή αυτό αντιπροσωπεύει το κουμπί που πρέπει να πατήσουν οι πεζοί για να ανάψει το φανάρι τους και το ενεργοποιούμε. Όσο το κουμπί των πεζών δεν πατιέται(while (interr == 0){…}) αναλογα με το αν έχω αμάξι στον μικρό δρόμο ανάβει πράσινο το φανάρι για τον μικρό δρόμο και γίνεται κόκκινο του μεγάλου δρόμου ή αν δεν έχω αμάξι στον μικρό δρόμο το φανάρι του μεγάλου δρόμου παραμένει πράσινο . Χάρης την μεταβλητή PreviousState καταφέρνουμε να προσπερνάμε γραμμές κώδικα οι οποίες δεν χρειάζονται ανάλογα με την κατάσταση που βρισκόμαστε. Για παράδειγμα έστω ότι είναι πράσινο το φανάρι του μεγάλου δρόμου και ο αισθητήρας στο μικρό δρόμο εντοπίζει αμάξι , τότε θα ανάψει κόκκινο για τον μεγάλο δρόμο και πράσινο για τον μικρό . Τώρα αν ξανά έχει αμάξι στον μικρό δρόμο το πρόγραμμα δεν χρειάζεται να κάνει κόκκινο το φανάρι του μεγάλου δρόμου και πράσινο του μικρού δρόμου αλλά απλά να παραμείνει ως έχει. Με αυτόν τρόπο εξασφαλίζονται όλες οι δυνατές περιπτώσεις και έτσι γλιτώνουμε χρόνο από το να εκτελέσουμε γραμμές κώδικα που δεν είναι αναγκαίες σε κάποια κατάσταση. Τώρα αν κάποιος πεζός πατήσει το κουμπί αυτόματα θα γίνει κόκκινο το φανάρι θα ανάψει πράσινο για τον μικρό δρόμο και για τους πεζούς και ο timer θα αρχίσει να μετράει. Μέχρι να τελειώσει ο χρόνος του timer και να ενεργοποιηθεί το interrupt(while(timercomplete==0){}) το φανάρι των πεζών και του μικρού δρόμου παραμένουν να είναι πράσινα. Έπειτα από αυτό το πέρας του χρόνου ενεργοποιείται το interrupt του timer καθαρίζουμε τον counter του timer για την επόμενη φορά που θα χρησιμοποιηθεί και το πρόγραμμα μας ξεκινάει πάλι από την ετικέτα “start:” και συνεχίζει να κάνει τις λειτουργίες που ζητάει η άσκηση. Τέλος αναφέρουμε ότι ο λόγος που έχουμε βάλει τα delays είναι διότι τα pins πρέπει να προλάβουν να πάρουν τις τιμές που τους αντιστοιχεί κάθε φορά.

**Κώδικας Προγράμματος:**

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define del 0.2

#define ped 10

int interr=0;

int PreviousState=0; // State 0: Big GREEN Small RED. State 1: Big RED Small GREEN (Used to skip lines)

int timercomplete=0; // When timer reaches 20seconds this will be turned to 1 through interrupt

int main(void)

{

time\_t t;

srand((unsigned) time(&t)); // Setting seed for rand() function.

PORTD.DIR |=PIN0\_bm; // PIN 0 (Right bit) --> BIG Traffic Light

PORTD.DIR |=PIN1\_bm; // PIN 1 (Middle bit) --> SMALL Traffic Light

PORTD.DIR |=PIN2\_bm; // PIN 2 (Left bit) --> PEDESTRIANS Light

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

TCA0.SINGLE.CTRLB = 0; // Normal Mode

TCA0.SINGLE.CMP0 = ped; // When reaches this value -> interrupt

TCA0.SINGLE.CTRLA = TCA\_SINGLE\_CLKSEL\_DIV256\_gc; // Clock Frequency/256

TCA0.SINGLE.INTCTRL = TCA\_SINGLE\_CMP0\_bm; // Interrupt Enable

PORTD.OUTSET=PIN0\_bm; // Turning big light to green.

\_delay\_ms(del);

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_BOTHEDGES\_gc; // Pedestrian light button (interrupt)

sei(); // Enable interrupts

start:

while (interr==0)

{

if(rand()%10==0,rand()%10==5,rand()%10==8)

{

if (PreviousState==1) continue;

PORTD.OUTCLR=PIN0\_bm; // Turning big light to red.

\_delay\_ms(del);

PORTD.OUTSET=PIN1\_bm; // Turning small light to green.

\_delay\_ms(del);

PreviousState=1;

}

else

{

if (PreviousState==0) continue;

PORTD.OUTCLR=PIN1\_bm; // Turning small light to red.

\_delay\_ms(del);

PORTD.OUTSET=PIN0\_bm; // Turning big light to green.

\_delay\_ms(del);

PreviousState=0;

}

}

if (PreviousState==1) goto point;

PORTD.OUTCLR=PIN0\_bm; // Turning big light to red.

\_delay\_ms(del);

PORTD.OUTSET=PIN1\_bm; // Turning small light to green.

\_delay\_ms(del);

PreviousState=1;

point:

PORTD.OUTSET=PIN2\_bm; // Turning pedestrian light on (Green).

\_delay\_ms(del);

TCA0.SINGLE.CTRLA |=1; // Start timer of 20s for pedestrians

while(timercomplete==0){} // Stay on while until interrupted by timer

PORTD.OUTCLR=PIN2\_bm; // Turning pedestrian light off (Red).

\_delay\_ms(del);

timercomplete=0;

interr=0;

goto start;

cli(); // Just in case

}

ISR(PORTF\_PORT\_vect) // Pedestrians interrupt

{

int intflags=PORTF.INTFLAGS;

PORTF.INTFLAGS=intflags;

interr=1;

}

ISR(TCA0\_CMP0\_vect){ // Timer interrupt 20s

TCA0.SINGLE.CTRLA = 0; // Disable Timer

int intflags = TCA0.SINGLE.INTFLAGS; // Clear Interrupt Flags

TCA0.SINGLE.INTFLAGS=intflags;

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

timercomplete=1;

}