## Άσκηση 6 Γ. Ασκήσεις πάνω στους χρονιστές(Timers)

**Εφαρμογή για την δημιουργία λογικού probe με χρήση διακοπών από τον Timer0**

**Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού C για τον μικροελεγκτή PIC 18F4550 με το οποίο ο μικροελεγκτής θα λειτουργεί σαν λογικό probe(ακροδέκτης με τον οποίο ανιχνεύουμε σε ένα σημείο του κυκλώματος την ύπαρξη του λογικού 0 ή του λογικού 1).**

**Η είσοδος του λογικού probe θα είναι ο ακροδέκτης RD0.**

**Αν ο ακροδέκτης RD0 είναι σε λογικό 0 θα πρέπει να αναβοσβήνει το LED0 το οποίο είναι συνδεδεμένο στον ακροδέκτη RB0 με συχνότητα 2 Hz.**

**Αν ο ακροδέκτης RD0 είναι σε λογικό 1 θα πρέπει να αναβοσβήνει το LED7 το οποίο είναι συνδεδεμένο στον ακροδέκτη RB7 με συχνότητα 5 Hz.**

**Να χρησιμοποιηθεί η μέθοδος της διακοπής από υπερχείλιση του timer0.**



Κύκλωμα άσκησης 6Γ. Ο ακροδέκτης RD0 χρησιμοποιείται σαν λογικό probe(ακροδέκτης από τον οποίο ανιχνεύουμε την ύπαρξη σε κάποιο σημείο του κυκλώματος του λογικού 1 ή του λογικού 0.

Λύση:

**Α)**

Έστω ότι έχει ρυθμιστεί οι διακοπές(interrupts) από τον timer0 να συμβαίνουν κάθε 50 ms.

**Αν RD0=0** θα πρέπει το LED0 που συνδέεται στο RB0 να αναβοσβήνει με συχνότητα 2 HZ.

Η περίοδος θα είναι: .

To LED0 θα πρέπει να αλλάζει κατάσταση ανά χρονικό διάστημα:

***Δηλαδή αν RD0=1, τότε το LED0 θα πρέπει να αλλάζει κατάσταση κάθε 5 διακοπές.***

**Αν RD0=1** θα πρέπει το LED0 που συνδέεται στο RB7 να αναβοσβήνει με συχνότητα 5 HZ.

Η περίοδος θα είναι: .

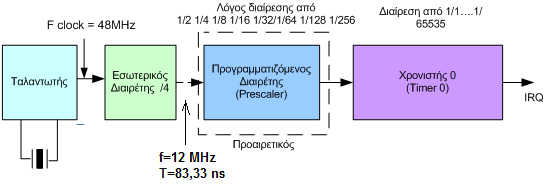
To LED7 θα πρέπει να αλλάζει κατάσταση ανά χρονικό διάστημα:

***Δηλαδή αν RD0=1, τότε το LED7 θα πρέπει να αλλάζει κατάσταση κάθε 2 διακοπές.***

***Β)***

***Υπολογισμός της αρχικής τιμής του timer0 ώστε να συμβαίνουν διακοπές κάθε 50 ms.***  *Επιλέγεται τιμή του προγραμματιζόμενου διαιρέτη ίση με .*

Υπολογισμός της αρχικής τιμής του timer0 για να έχουμε διακοπές κάθε 50 ms.



50.000.000 nsec = 83,333 nsec \* 64 \* (65536 – αρχική τιμή του μετρητή )⇔

Αρχική τιμή μετρητή==56161

***Με αρχική τιμή στον timer0=56161 θα έχουμε διακοπές κάθε 50 ms***

Διάγραμμα ροής:

Αρχή

Αρχικές ρυθμίσεις, είσοδοι, έξοδοι, ρυθμίσεις διακοπών. init()

Μεταφορά της κατάστασης του ακροδέκτη RD0 στην μεταβλητή flag

Καθυστέρηση 50 ms

Απόδοση αρχικής τιμής στον timer0 ώστε να γίνονται διακοπές κάθε 50 ms

Ελάττωση του μετρητή διακοπών(counter) κατά 1

Είναι counter=0;

NAI

OXI

Είναι flag =1;

OXI

NAI

counter=5

counter=2

Σβήσιμο του LED0

Αλλαγή κατάστασης του LED7

Σβήσιμο του LED7

Αλλαγή κατάστασης του LED0

Αρχή Ρουτίνας αρχικοποίησης init()

Πόρτα D είσοδος

Πόρτα Β έξοδος

Ενεργοποίηση του γενικού διακόπτη διακοπών

Ενεργοποίηση διακοπών από τον timer0

Τοποθέτηση του prescaler σε 1/64

Αρχική τιμή στον timer0 ώστε να έχουμε διακοπές σε 50 ms

Απόδοση αρχικής τιμής στον μετρητή διακοπών(counter)

Τέλος ρουτίνας αρχικοποίησης

#include <main.h>

#byte PORTB =0xF81 //ορισμός των θυρών με την θέση τους στην μνήμη

#byte PORTD =0xF83

**// Προσοχή στους παρακάτω ορισμούς πράξεων Toggle\_Led0 και Toggle\_Led7**

#define Toggle\_Led0 PORTB=PORTB^0x01; // αλλαγή κατάστασης του bit RB0. ^ σημαίνει exclusive OR.

#define Toggle\_Led7 PORTB=PORTB^0x80; // αλλαγή κατάστασης του bit RB7. ^ σημαίνει exclusive OR.

// Με το Toggle\_Led0 αλλάζουμε την κατάσταση του bit 0 της πόρτας Β

// Με το Toggle\_Led7 αλλάζουμε την κατάσταση του bit 7 της πόρτας Β

**// Προσοχή στους παρακάτω ορισμούς πράξεων clear\_Led0 και clear\_Led7**

#define clear\_Led0 PORTB=PORTB&0b11111110; // μηδενισμός του bit RB0

#define clear\_Led7 PORTB=PORTB&0b01111111; // μηδενισμός του bit RB7

// Με το clear\_Led0 μηδενίζουμε το bit 0 της πόρτας Β και αφήνουμε τα άλλα bit αμετάβλητα

//Με το clear\_Led7 μηδενίζουμε το bit 7 της πόρτας Β και αφήνουμε τα άλλα bit αμετάβλητα

int8 counter; // μεταβλητή για ρύθμιση συχνότητας αναβοσβησίματος του led. Μετρητής //διακοπών. Είναι global μεταβλητή, τοποθετείται πάνω από την main()

int1 flag=0; //τιμή της μεταβλητής flag=1 οταν RD0=1 και flag=0 όταν RD0=0

//Είναι global μεταβλητή, τοποθετείται πάνω από την main()

// Δήλωση συναρτήσεων

void timer0\_int(void); //Οι συναρτήσεις που θα χρησιμοποιηθούν δηλώνονται πάνω από την main()

void init (void);

// Κύρια συνάρτηση. Ελέγχει την είσοδο του probe(RD0) και αλλάζει την κατάσταση του flag σε 1 ή σε 0

void main()

{

init();

while (TRUE){

flag=input(PIN\_D0); //μεταβλητή στην οποία αποδίδεται η κατάσταση

// του ακροδέκτη RD0 στην μεταβλητή flag

delay\_ms(50); //καθυστέρηση για αποφυγή φαινομένου αναπηδήσεων

//κατά την αλλαγή κατάστασης της εισόδου από το RD0

}

} //κλείνει η main

// Ρουτίνα διακοπής από τον timer0. Αλλάζει την κατάσταση του led0 ή του led7.

#INT\_TIMER0 // Directive. Οδηγία προς τον Compiler. Δηλώνει ότι η επόμενη ρουτίνα είναι ρουτίνα

// διακοπών από τον timer0

void timer0\_int(void){

set\_timer0(56161); // αρχική τιμή του μετρητή

// για να συμβεί η επόμενη

// διακοπή σε 50ms

counter--; // ελαττώνεται ο μετρητής διακοπών

//τι συμβαίνει όταν o counter γίνει 0 και το probe είναι σε λογικό 0.

if (counter==0 && flag==0){ //τι συμβαίνει όταν το probe(RD0) είναι 0

counter=5;

Toggle\_Led0; //Αλλαγή κατάστασης του led0

//της πόρτας Β

clear\_Led7; //σβήνει το led7

}

//τι συμβαίνει όταν το probe(RD0) γίνει ο και το probe είναι σε λογικό 1.

if (counter==0 && flag==1){ //τι συμβαίνει όταν το probe(RD0) είναι 1

counter=2;

Toggle\_Led7; //Αλλαγή κατάστασης του led7

//της πόρτας Β

clear\_Led0; //σβήνει το led0

}

}

// Ρουτίνα αρχικοποίησης

void init (void){

set\_tris\_b(0x00); // Καθορισμός της πόρτας Β ως εξόδου

set\_tris\_d(0xff); // Καθορισμός της πόρτας D ως εισόδου

PORTB = 0;

counter=5; // Αρχική τιμή του counter που αντιστοιχεί σε RD0=0

SETUP\_TIMER\_0(T0\_INTERNAL | T0\_DIV\_64 ); //Prescaler=64

set\_timer0(56161); // Αρχική τιμή του μετρητή timer0 για διακοπές

//κάθε 50 ms

enable\_interrupts(INT\_TIMER0); // Ενεργοποίηση της

//διακοπής του timer0

enable\_interrupts(GLOBAL); // Ενεργοποίηση του γενικού

// διακόπτη των διακοπών

}