

**ENGINE**



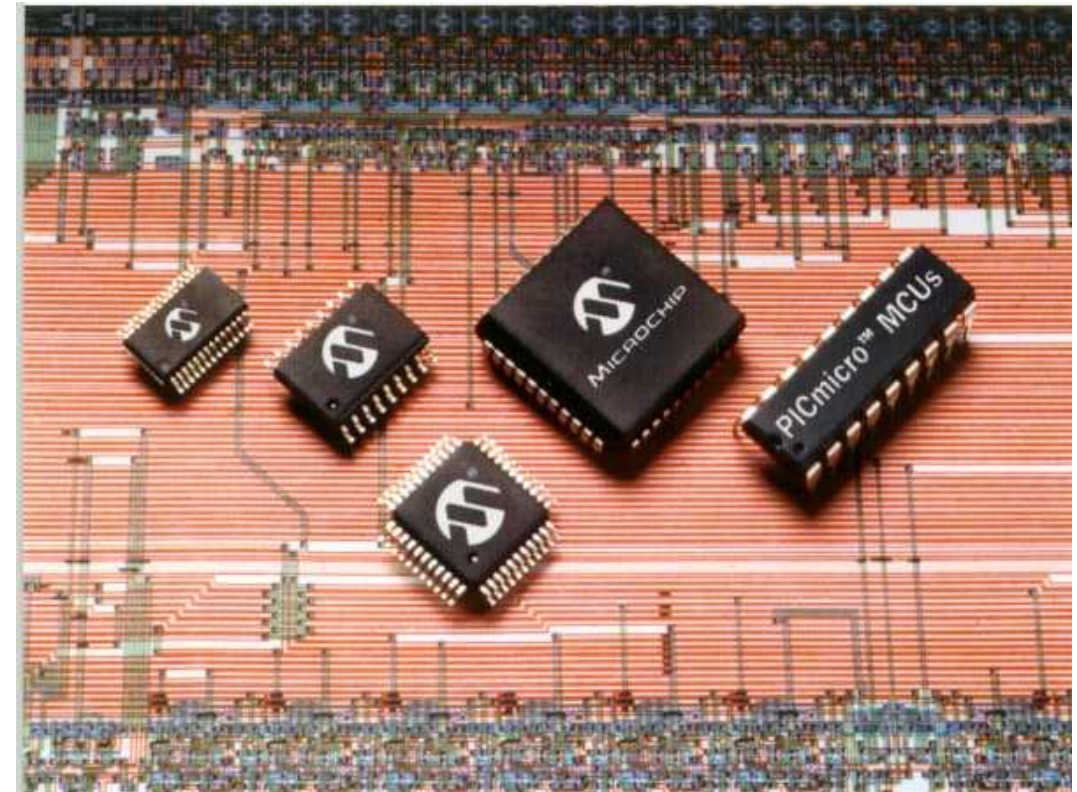
**TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING  
IN HIGHER EDUCATION**

---

## 6. Timers

# Περιεχόμενα

- Εισαγωγή
- Παράδειγμα



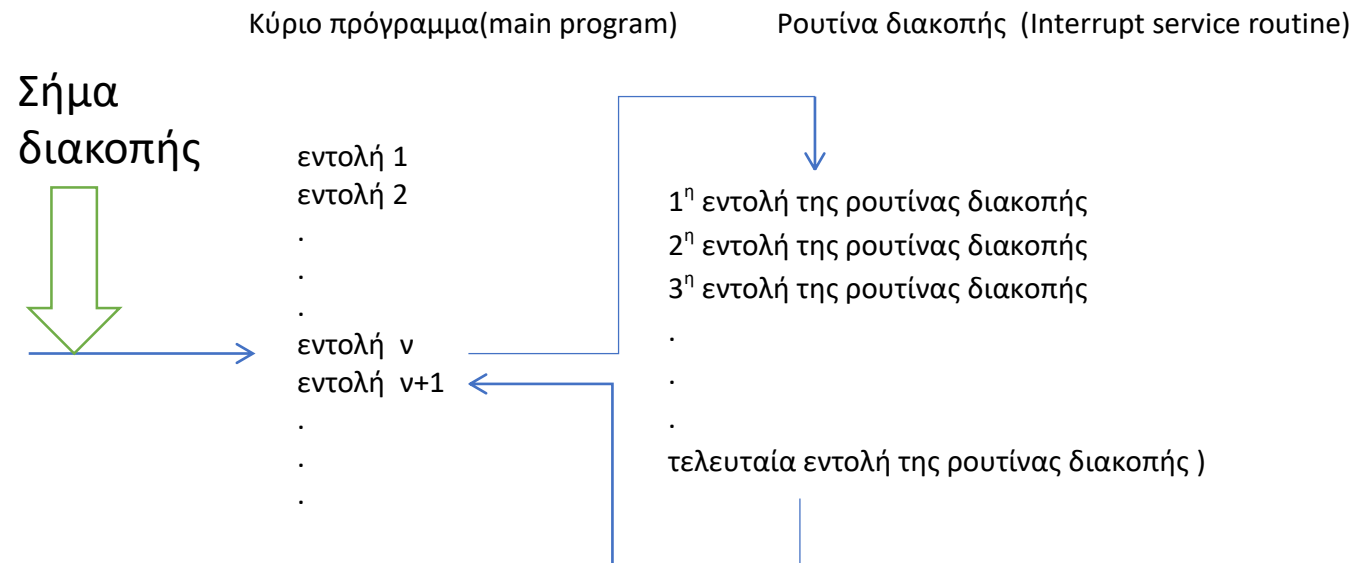
# 6. *Interrupts*

## Εισαγωγή

### Διακοπές (Interrupts)

Τι εννοούμε με τον όρο **διακοπή (Interrupt)** στους μικροελεγκτές;  
Διακοπή(Interrupt) είναι η διακοπή της εκτέλεσης της ροής του προγράμματος του μικροελεγκτή προκειμένου να εκτελεστεί ένα κομμάτι προγράμματος το οποίο ονομάζεται ρουτίνα διακοπής

Παράδειγμα:



Τι είναι οι Timers (Χρονιστές) του μικροελεγκτή;

Είναι κάποιοι καταχωρητές των 8 bit ή των 16 bit των οποίων το περιεχόμενο αυξάνει ανά συγκεκριμένο χρονικό διάστημα.

**• Προσοχή, μετά την τιμή 111 ... 111 (=65535) ο timer παίρνει την τιμή 000 ... 000.**

**• Σ' αυτή την μετάβαση μπορούμε να προγραμματίσουμε να συμβαίνει μια διακοπή (interrupt).**

## 6. *Interrupts* Εισαγωγή

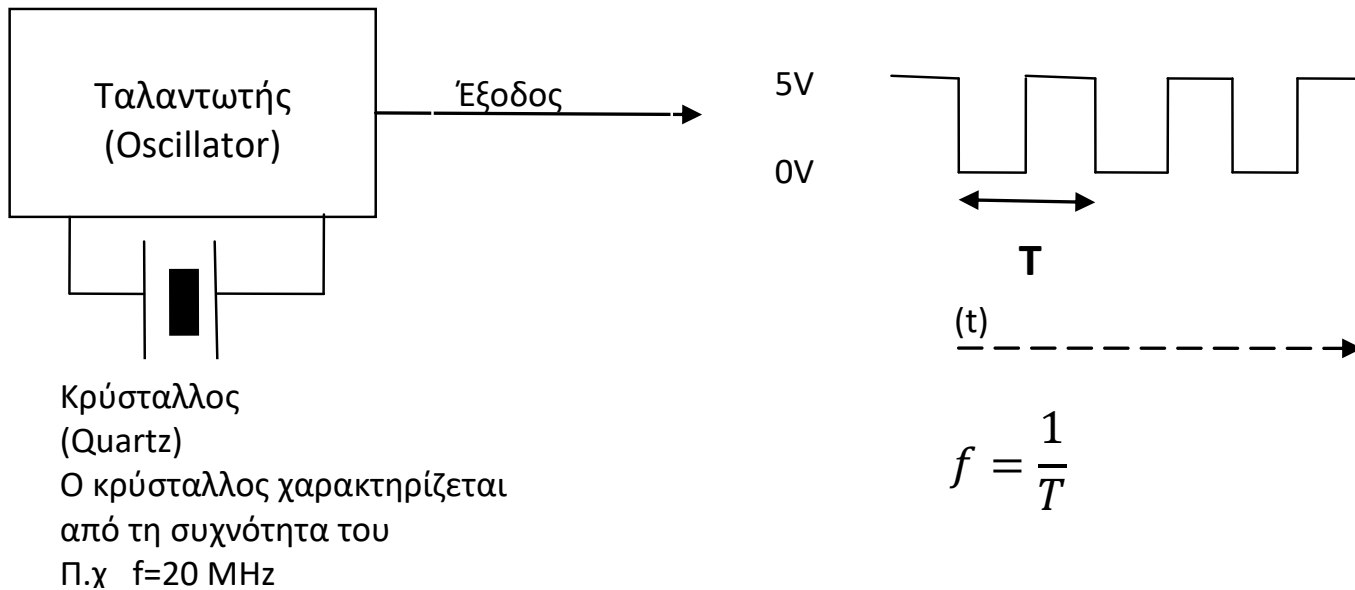
### Ερώτηση:

**Από πού προέρχονται οι παλμοί που αυξάνουν την τιμή του Timer;**

### Απάντηση:

- Μπορεί να προέρχονται από μια εξωτερική πηγή και να εφαρμόζονται σε έναν συγκεκριμένο ακροδέκτη του μικροελεγκτή
- Ή να προέρχονται από ένα κύκλωμα στο εσωτερικό του μικροελεγκτή το οποίο ονομάζεται εσωτερικός ταλαντωτής (Oscillator)
- Στο πρόγραμμα μας με μια εντολή δηλώνουμε την πηγή των παλμών που θέλουμε να χρησιμοποιηθεί

## 6. Interrupts Εισαγωγή



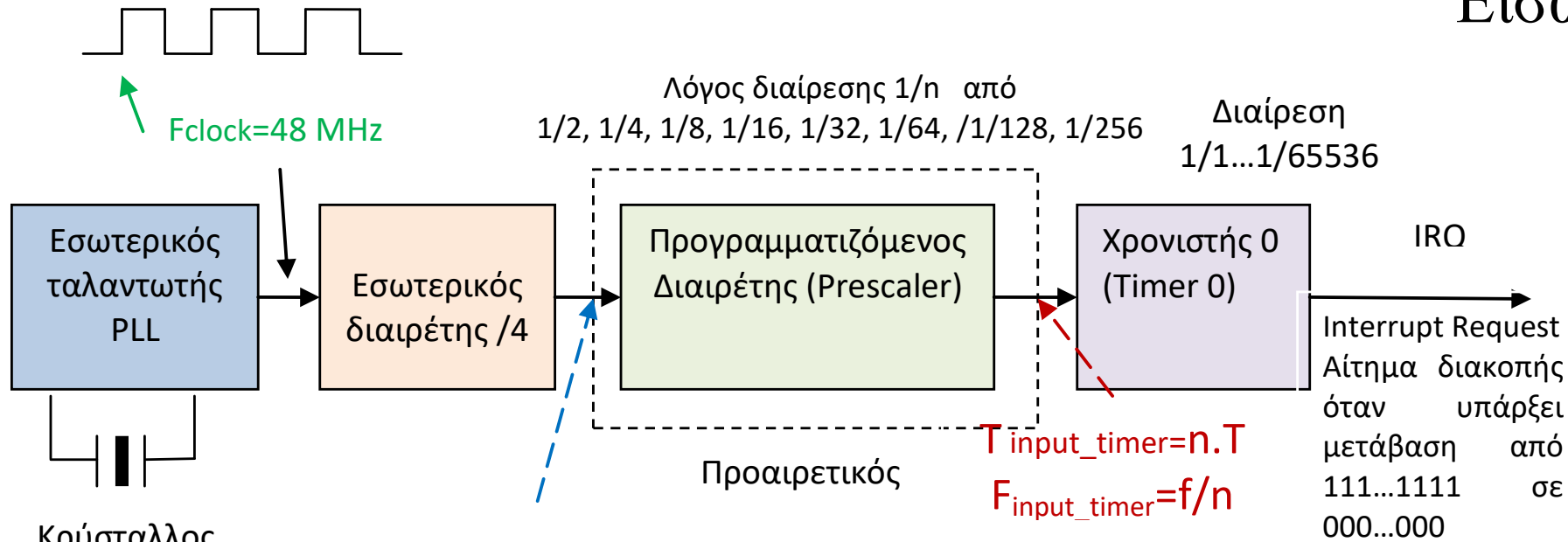
### Timer0

Ο Timer0 είναι ένας timer των 8 bit ή των 16 bit.

Με μια εντολή ρυθμίζουμε πως θα χρησιμοποιηθεί ο timer0. Εάν θέλουμε να χρησιμοποιηθεί σαν timer των 8 bit θα πρέπει να συμπληρώσουμε τη σχετική παράμετρο στην εντολή ρύθμισης του timer0

```
setup_timer_0(TO_INTERNAL|TO_DIV_256|TO_8_BIT);
```

# 6. Interrupts Εισαγωγή



Κρύσταλλος  
20 MHz

$T$  κύκλος μηχανής (Machine Cycle)

$$T = \frac{1}{12\text{MHz}} = \frac{1}{12 \times 10^6 \text{Hz}} = 0,08333 \times 10^{-6} = 83,33 \times 10^{-9} = 83,33 \text{ ns}$$

Ο κύκλος μηχανής MC είναι 83,33 ns      MC → Machine Cycle

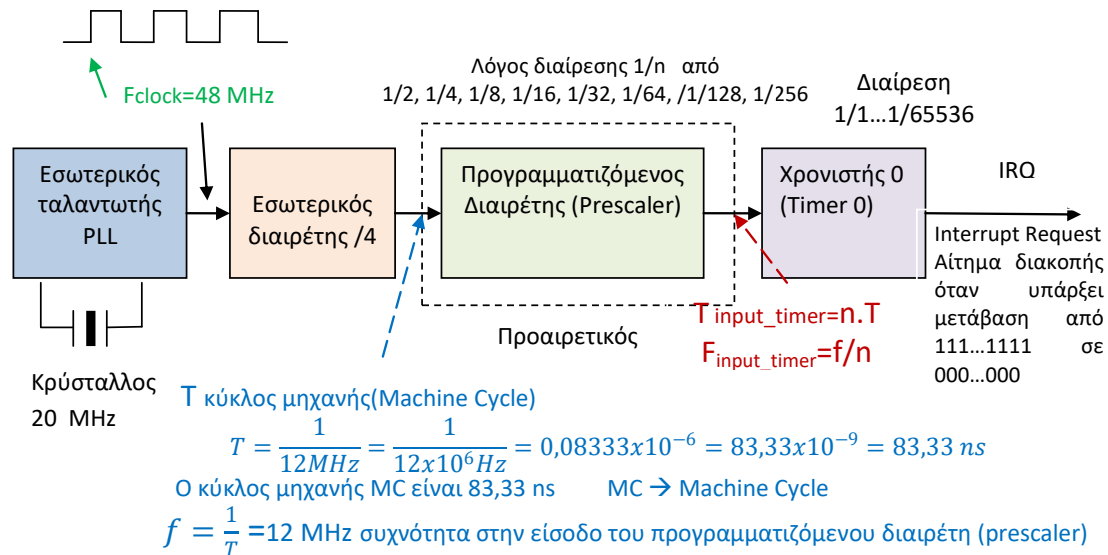
$$f = \frac{1}{T} = 12 \text{ MHz} \text{ συχνότητα στην είσοδο του προγραμματιζόμενου διαιρέτη (prescaler)}$$

Όταν δεν χρησιμοποιείται ο προγραμματιζόμενος διαιρέτης ο timer0 αυξάνει κατά 1 κάθε 83,33 ns.

Όταν ο προγραμματιζόμενος διαιρέτης τεθεί στην τιμή  $\frac{1}{2}$  ο timer0 αυξάνει κατά 1 κάθε  $2 \times 83,33 \text{ ns} = 166,66 \text{ ns}$ .

Όταν ο προγραμματιζόμενος διαιρέτης τεθεί στην τιμή  $\frac{1}{16}$  ο timer0 αυξάνει κατά 1 κάθε  $16 \times 83,33 \text{ ns} = 1333,28 \text{ ns} = 1,33328 \text{ μs}$ .

## 6. Interrupts Εισαγωγή



```
// Ενεργοποίηση της διακοπής από τον timer0  
enable_interrupts(INT_TIMER0);
```

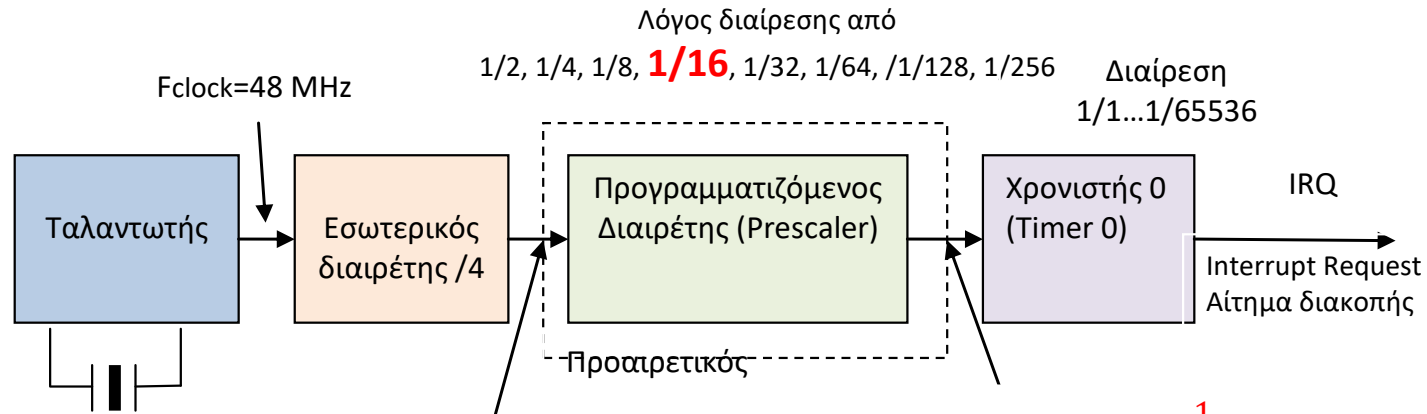
```
// Ενεργοποίηση του γενικού διακόπτη των διακοπών  
enable_interrupts(GLOBAL);
```

```
setup_timer_0(T0_INTERNAL|T0_DIV_256);
```



# Υπολογισμός ώστε να συμβαίνουν διακοπές ανά συγκεκριμένο χρονικό διάστημα για παράδειγμα κάθε 50 ms

## 6. Interrupts Εισαγωγή



T κύκλος μηχανής (Machine Cycle)

$$T = \frac{1}{12 \text{ MHz}} = 83,33 \text{ ns}$$

Επιλέχθηκε λόγος διαίρεσης  $\frac{1}{16}$

Η περίοδος των παλμών στην είσοδο του timer θα είναι:

$$T_{\text{input-timer}} = 16 \times T$$

$$T_{\text{input-timer}} = 16 \times 83,33 \text{ ns} = 1333,28 \text{ ns} = 1,33328 \mu\text{s}$$

Για να έχουμε διακοπές κάθε 50 ms θα πρέπει αρχική τιμή y που τίθεται στον timer μετά από κάθε διακοπή να πληροί τη σχέση:

$$(65536 - y) \times 1,33328 \mu\text{s} = 50000 \mu\text{s} \Leftrightarrow$$

$$65536 - y = \frac{50000}{1,33328} = 37501 \text{ (στρογγυλοποίηση στη ακέραιη τιμή)} \Leftrightarrow$$

$$y = 65536 - 37501 = 28035$$

Αυτή είναι η αρχική τιμή που πρέπει να τίθεται στον Timer ώστε να έχουμε διακοπές κάθε 50 ms

## 6. *Interrupts* Εισαγωγή

Πως θα ρυθμίσουμε ώστε μετά από κάθε διακοπή ο timer να μην παίρνει την αρχική τιμή 0 αλλά 28035;

Μέσα στη ρουτίνα διακοπών θα γράψουμε την εντολή:

```
#INT_TIMER0 // Οδηγία ότι η επόμενη ρουτίνα είναι η ρουτίνα εξυπηρέτησης της
             // από τον Timer0
void timer0_int(void){
set_timer0(28035); //αποδίδεται αρχική τιμή στον timer0 ώστε η επόμενη
                  // διακοπή να συμβεί σε χρόνο ίσο με 50 ms
                  // Η επόμενη διακοπή θα γίνει μετά από 65536-28035 παλμούς
                  // στην είσοδο του timer.

                  // .....
                  // Άλλες εντολές της ρουτίνας διακοπών από τον timer0
                  //.....
} // Αγκύλη κλεισίματος της ρουτίνας διακοπών
```

## ENGINE Partnership

- Warsaw University of Technology (PL) - *coordinator*
- IHU - International Hellenic University (GR)
- EDUMOTIVA - European Lab for Educational Technology (GR)
- University of Padova (IT)
- University of Applied Sciences in Tarnow (PL)



INTERNATIONAL  
HELLENIC  
UNIVERSITY



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA



## CONTACT:



[www.engined.eu](http://www.engined.eu)



[angelika.tefelska@pw.edu.pl](mailto:angelika.tefelska@pw.edu.pl)



[@projectENGINE1](https://twitter.com/projectENGINE1)



[@EUprojectEngine](https://www.facebook.com/EUprojectEngine)



# Erasmus+

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.