ENGINE

Erasmus+

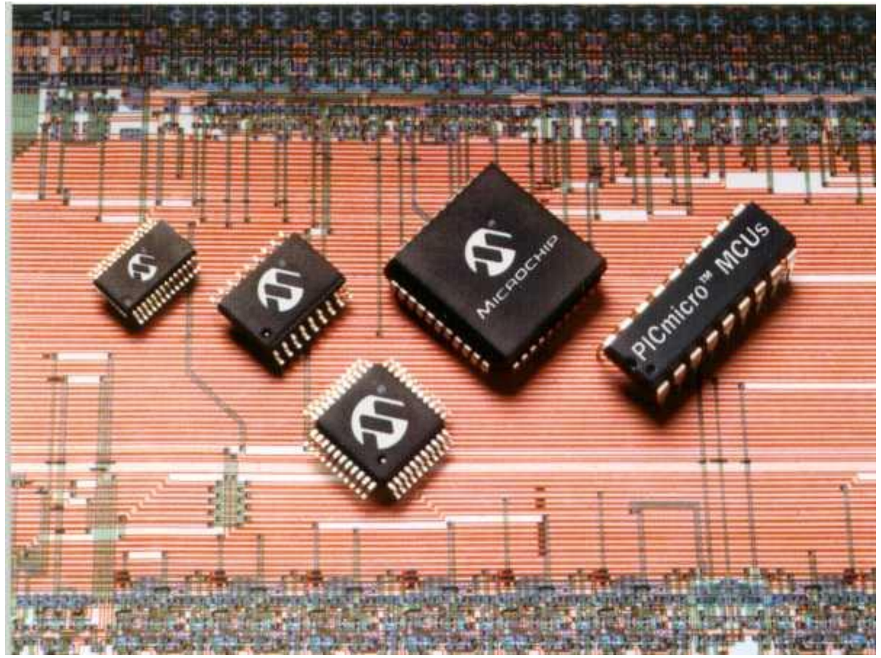
TEACHING ONLINE ELECTRONICS, MICROCONTROLLERS AND PROGRAMMING IN HIGHER EDUCATION

Module_2-8. Timers

PIC18F4550 with Proteus Simulation

Contents

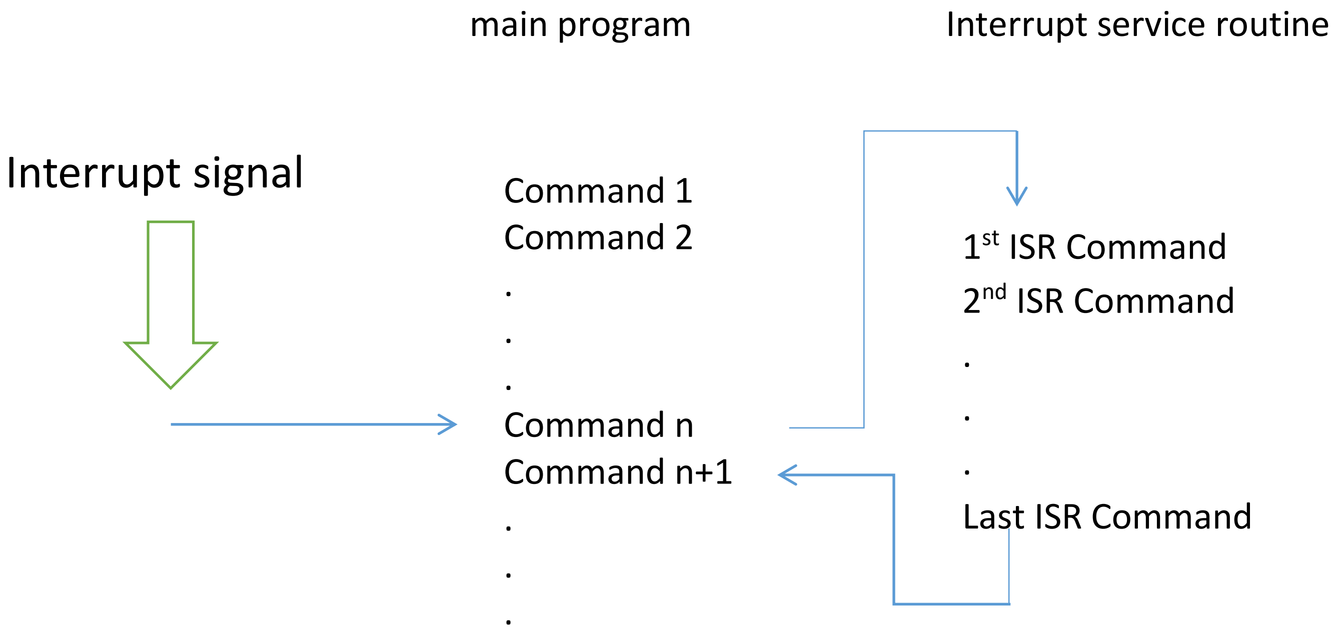
- Interrupt
- Timer



Interrupts

Interrupt is to stop execution of the microcontroller's program flow in order to execute a block of code called an interrupt service routine

Example:



main program

Interrupt service routine

Interrupt signal

Command 1
Command 2
.
.
.
Command n
Command n+1
.
.
.

1st ISR Command
2nd ISR Command
.
.
.
Last ISR Command

Module_2-8. Timers
Timer

Timers are some 8-bit or 16-bit registers whose content is incremented by a specific time interval.

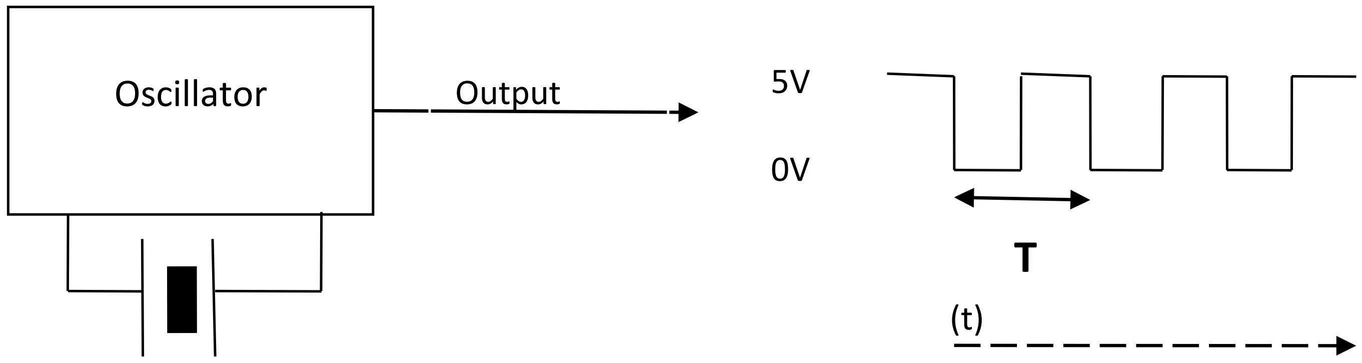
After the value 111 ... 111 the timer takes the value 000 ... 000 (overflow). In this transition we can program an interrupt to occur.

Question: Where do the pulses that increment the Timer value come from?

Answer: They can come from an external source and be applied to a specific pin of the microcontroller
Or come from a circuit inside the microcontroller called an internal oscillator (Oscillator)

In our program with a command we declare the source of the pulses that we want to be used

Module_2-8. Timers
Timer



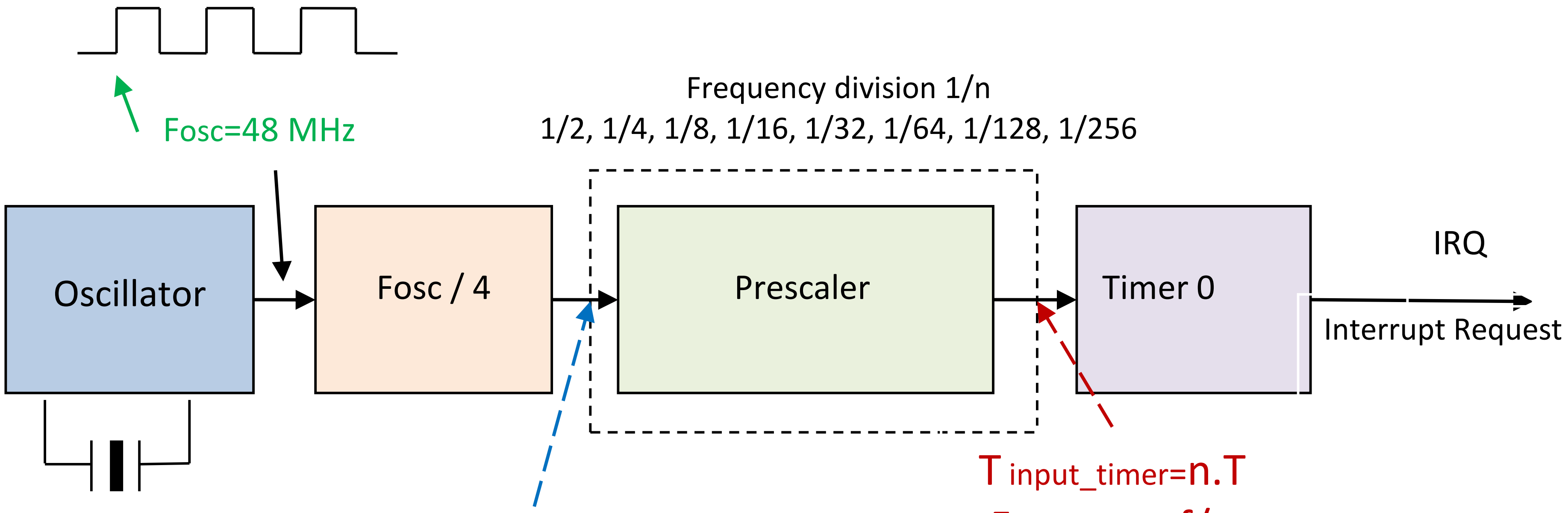
$$f = \frac{1}{T}$$

Timer0

Timer0 is an 8 bit or 16 bit. If we want it to be used as an 8-bit timer, we should fill in the relevant parameter in the timer0 setup command.

```
setup_timer_0(T0_INTERNAL|T0_DIV_256|T0_8_BIT);
```

Module_2-8. Timers
Timer



$T_{\text{input\_timer}}=n.T$

$F_{\text{input\_timer}}=f/n$

T = Machine Cycle

$$T=\frac{1}{12MHz}=\frac{1}{12x10^6Hz}=0{,}08333x10^{-6}=83{,}33x10^{-9}=83{,}33\,ns$$

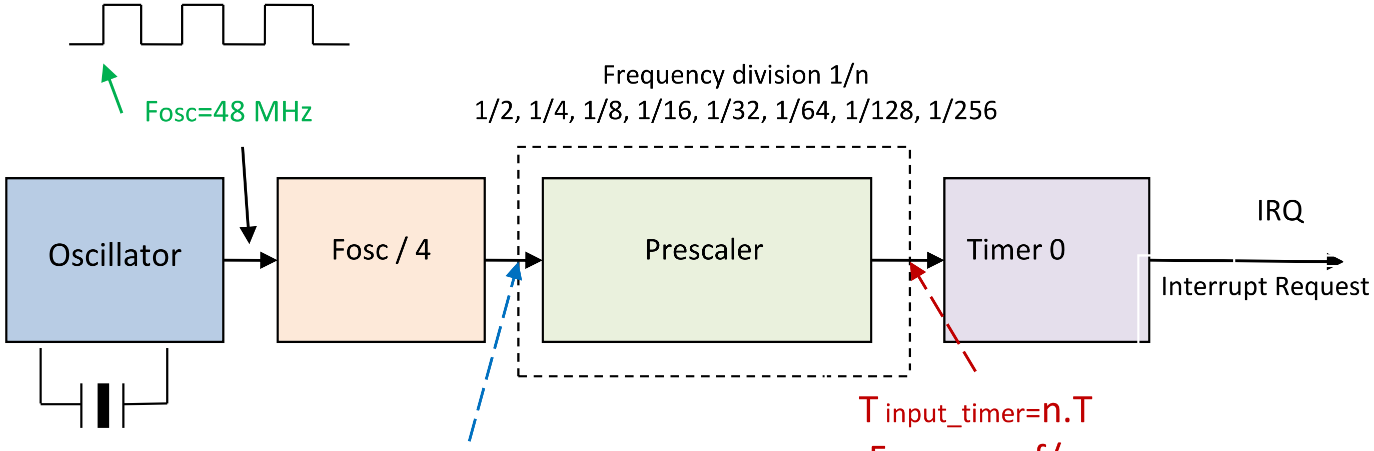
MC = 83,33 ns   MC → Machine Cycle

$f=\frac{1}{T}$ =12 MHz in the input of the prescaler

Prescaler is a configurable clock-divider circuit. It can be used to divide the clock frequency input to the timer module. Prescaler's values: 1, 2, 4, 8, 16, 32, 64, 128, 256

Module_2-8. Timers

Timer



$T_{\text{input\_timer}}$=n.T

$F_{\text{input\_timer}}$=f/n

T = Machine Cycle

$$T = \frac{1}{12MHz} = \frac{1}{12x10^6 Hz} = 0{,}08333x10^{-6} = 83{,}33x10^{-9} = 83{,}33\ ns$$

MC = 83,33 ns MC → Machine Cycle

$f = \frac{1}{T}$ =12 MHz in the input of the prescaler

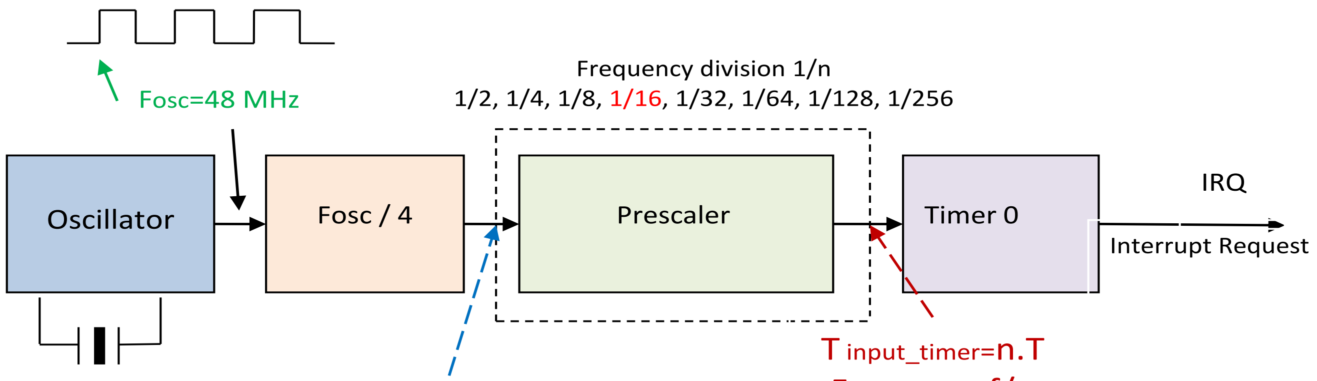
```
enable_interrupts(INT_TIMER0);
// Enable interrupt from timer0

enable_interrupts(GLOBAL);
// enables (if set) all un-masked interrupts or disables (if cleared) all interrupts.

setup_timer_0(T0_INTERNAL|T0_DIV_256);
```

Module_2-8. Timers
Timer

Calculation so that interrupts occur at a specific time interval, for example every 50 ms



T $_{\text{input\_timer}}$=n.T

F$_{\text{input\_timer}}$=f/n

T = Machine Cycle

$$T = \frac{1}{12MHz} = \frac{1}{12x10^6Hz} = 0{,}08333x10^{-6} = 83{,}33x10^{-9} = 83{,}33\ ns$$

MC = 83,33 ns   MC → Machine Cycle

$f = \frac{1}{T}$ =12 MHz in the input of the prescaler

Tinput-timer = 16 x T

Tinput-timer=16 x 83,33 ns=1333,28 ns=1,33328μs

In order to have interruptions every 50 ms, the initial value y set in the timer after each interruption should satisfy the equation:

$$(65536 - y)x1{,}33328\mu s = 50000\mu s \Leftrightarrow 65536 - y = \frac{50000}{1{,}33328} = 37501$$

$$y = 65536 - 37501 = \boxed{28035}$$

This is the initial value that should be set to the Timer so that we have interrupts every 50ms

Module_2-8. Timers

Timer

How do we set it up so that after every interruption the Timer0 does not take the initial value 0 but the value 28035?

```
#INT_TIMER0
// Directive that the next routine is its service routine for Timer0

void timer0_int(void){
  set_timer0(28035)
                                  .......
                                  .......
                                  .......
 }
```

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)

CONTACT:

www.engined.eu

angelika.tefelska@pw.edu.pl

@projectENGINE1

@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.