



WESTERN WEAR
FLAT 30-60% OFF
HURRY!



The Timer Modules - Timer0

Until now, we have cover the basic knowledge of PIC programming. We will start analyzing the built-in modules of the PICs. At this point, i should remind that this tutorial is based on the PIC 16F88, a very powerful 8-bit PIC. For the 8-bit PIC family the modules have more or less the same functionality. As you understand, this tutorial will not work for 16 or 32-bit PICs. I will start with the TIMER modules. These modules are by far the most important to learn.

The Timer0 Module

The PIC 16F88 has 3 built in timers. The first one is the Timer0 module. This is an 8-bit counter, which means that it can count from 0 to 255, and then it will loop back to 0. There are two sources that the Timer0 can increase the count. Each PIC module, can be controlled by bit switches in various registers. The Timer0 module uses some bits from the OPTION_REG register for setting it up.

Setting up and accessing the Timer0 module

The Timer0 module has a dedicated register that can be accessed for read and write operations. This register can be accessed from BANK 0 and BANK 2, and is named "**TMR0**". A direct read (MOVF TMR0,w) will transfer the value of the Timer0 counter directly to the W register, while a direct write (MOVWF TMR0) will transfer the content of the W register to the Timer0 counter.

The operation of the Timer0 module can be determined by specific bits of the OPTION_REG register. Let's see these bits one by one:

OPTION_REG register BIT <5> (T0CS):

This is the "**TMR0 Clock Source Select bit**". The value of this bit will determine if the TMR0 register will be increased on every internal instruction cycle of the PIC (in other words once every 4 clock pulses) ([learn more about the instruction cycle here](#)), or on every pulse on the TOCKI pin input. The possible values for the T0CS bit are:

- **0:** The increment is done on every internal instruction cycle
- **1:** The increment is done on every pulse on the TOCKI pin input of the PIC

OPTION_REG register BIT <4> (T0SE):

This is the "**TMR0 Source Edge Select bit**". This bit has an effect only if the T0CS bit (above) is 1, in other words if the TMR0 register increases with external pulses on the TOCKI pin. The increment is done at the EDGE of each pulse. The value of the T0SE bit will determine if the increment will be done on the POSITIVE GOING transition, or on the NEGATIVE GOING transition of the pulse. The values that this bit can get are:

- **0:** The increment is done on the LOW to HIGH transition of the pulse
- **1:** The increment is done on the HIGH to LOW transition of the pulse





- [Disclaimer](#)
- [Book Contents](#)
- [Discussion forum](#)

Basics

- What will you need
- Choosing the right PIC

The MPLAB

- Getting familiar with the MPLAB environment
- Creating a new project
- Open and close projects
- Creating new files and including them in the project
- Your very first assembly program
- Compile a program and transfer to the PIC

Section 1:
Beginner's theory

More

[Circuits](#)
[Experiments](#)
[Projects](#)
[Theory](#)
[Tech News](#)
[BLOG](#)



OPTION_REG register BIT <3> (PSA):

This is the "**Prescaler Assignment bit**". The prescaler is another module of the PIC. The prescaler divides the input counts before they arrive at the TMR0 register. The prescaler can divide both the pulses from the TOCKI pin (when TOCS=1) as well as the counts from the internal instruction cycles (when TOCS=0). The PSA bit can have the following values:

- **0:** The PSA is assigned to the Timer0 module and the counts are divided before they reach the TMR0 register
- **1:** The PSA is assigned to the Watchdog timer (another module of the PIC) and the counts will arrive directly to the TMR0 register

OPTION_REG register BIT <2:0> (PS):

These 3 bits are the "**Prescaler Rate Select bits**". These bits have an effect on the prescaler division rate all the time, but they have an effect on the Timer0 module **ONLY if the prescaler is assigned to the Timer0 module** (when PSA=0). If the prescaler is assigned to the Timer0 module, then these 3 bits will determine the division rate. If for example the division rate is 64, then the TMR0 module will increase its value by one, after 64 counts. The values that these bits can get are:

- **000:** 1/2 division
- **001:** 1/4 division
- **010:** 1/8 division
- **011:** 1/16 division
- **100:** 1/32 division
- **101:** 1/64 division
- **110:** 1/128 division
- **111:** 1/256 division

The Timer0 Interrupt

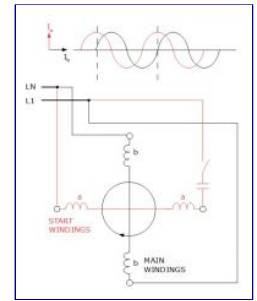
The Timer0 module has an 8-bit register for the counts, the TMR0 register located in BANK0 and BANK2. This means that it can count up to 255. If the TMR0 value is 255 (0xFF) and it is increased once more, then the TMR0 register will become 0 (0x00) and the **TMR0 interrupt** shall be raised. The TMR0IE interrupt can be controlled by the **INTCON** register bit 5 (TMR0IE - TMR0 Interrupt Enabled). When the TMR0 Interrupt is raised, the INTCON bit 2 (TMR0IF - TMR0 Interrupt Flag) is raised to indicate that the TMR0 interrupt is occurred. The TMR0IF must be cleared in software (BCF INTCON,TMR0IF) for the TMR0 interrupt to be re-enabled. [More info about the PIC Interrupts can be found in this link.](#)

Examples with the Timer0 Module

For the sake of understanding, find below some examples on how to setup the TMR0 module.

Example 1 - TMR0 increasing on every LOW to HIGH transition on TOCKI pin

```
BANK1                ; Select Bank 1
BSF OPTION_REG,TOCS  ; Transition on T0CKI pin
BCF OPTION_REG,T0SE   ; Increment on low-to-high transition on T0CKI
BSF OPTION_REG,PSA    ; Prescaler is assigned to the Watchdog
```

Example 2 - TMR0 increasing once every 128 internal instruction cycles

[New Theory: AC electric motor working principle](#)



**NO DEPOSIT
REQUIRED**

Offer valid for 48 hours

**PLAY
RUMMY
NOW**

```

BANK1          ; Select Bank 1
BCF OPTION_REG,TOCS ; Transition on T0CKI pin
BCF OPTION_REG,PSA  ; Prescaler is assigned to the Watchdog
BSF OPTION_REG,2    ;
BSF OPTION_REG,1    ;
BCF OPTION_REG,0    ; PS=110 = 1/128 division rate
BANK0          ; Select Bank 0

```

Example 3 - Read - Add 0x10 and Write the TMR0 register

```

BANK0          ; Select Bank 0
MOVF TMR0,w    ; READ TMR0
ADDLW 0x10     ; Add 0x10
MOVWF TMR0,w   ; WRITE TMR0

```



മിറ്റിങ്ങ് അടുത്ത ദിവസം
കമ്പ്യൂട്ടർ ക്രാഷ്.
എന്തു ചെയ്യും?



ഉണ്ടെങ്കിൽ
കാര്യം നിസ്സാരം

Confirm your knowledge

There is an online test to check your knowledge on this page. You may reveal the test with the following button:

Reveal the test

⬅ Previous page ---- Next page ➡

The Timer Modules - Timer0 ▼

[Go back to the book contents](#)

[Go to the discussion forum of this book](#)

Comments

Name

Email (shall not be published)

Website

☐ Notify me of new posts via email

Write your comments below:

BEFORE you post a comment: You are welcome to comment for corrections and suggestions on this page. But if you have questions [please use the forum](#) instead to post it. Thank you.



Submit your comment

No comment yet...

Be the first to comment on this page!

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise without the prior written permission of the author.

Read the [Disclaimer](#)

All trademarks used are properties of their respective owners.
Copyright © 2007-2009 Lazaridis Giorgos.
All rights reserved.