IMT SCHOOL

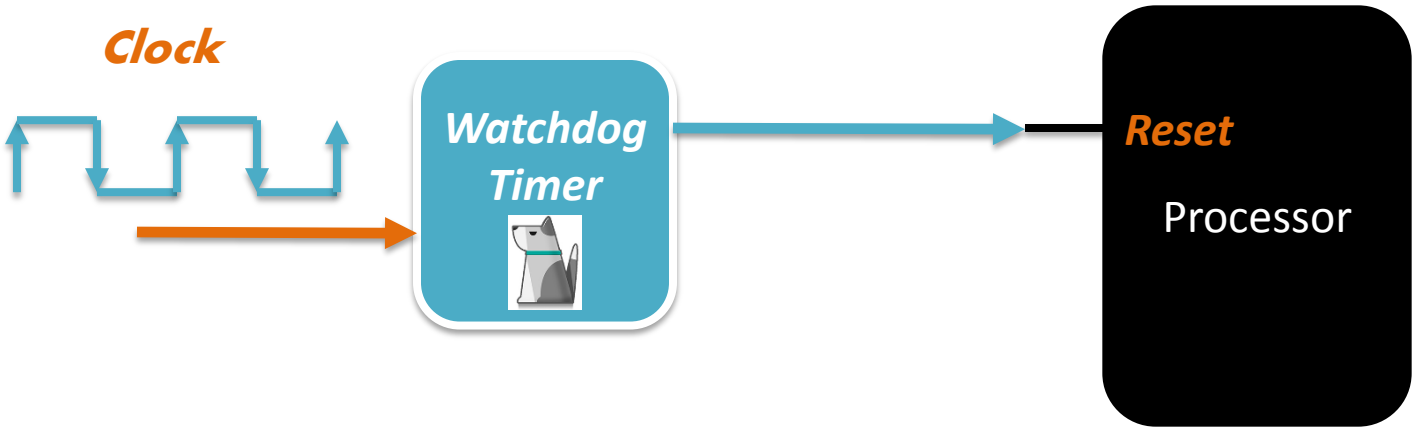
Embedded system interfacing

Lecture Twelve

Watch Dog Timer

Introduction

Watchdog timer is a normal timer peripheral that counts in a register with a certain configured frequency. The difference is that this timer is automatically generates a system reset when it timeout! i.e. when it reaches the maximum value



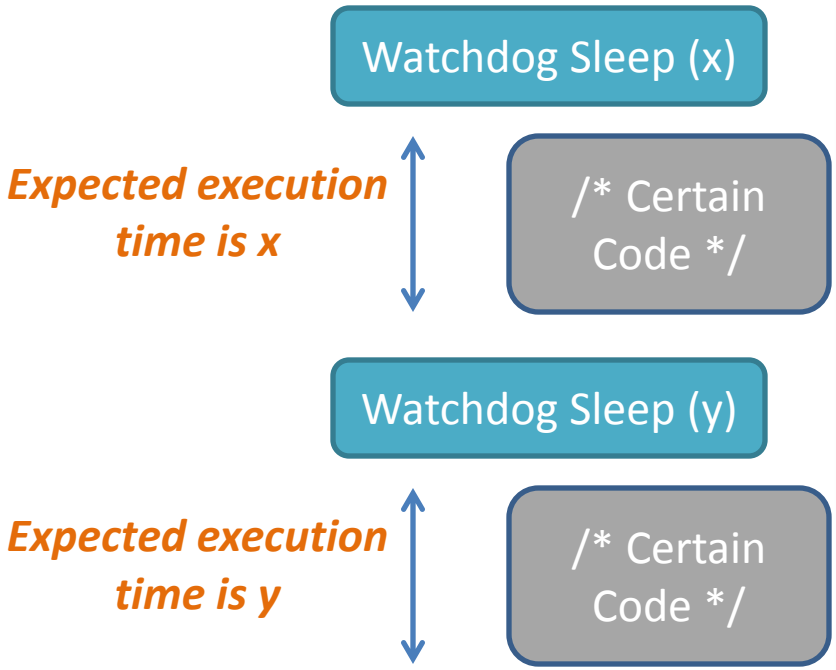
The watchdog timer is used to protect the processor if it halted due to an infinite loop or unexpected long part of the code.

Idea of operation

The watchdog Timeout value is the value that the watchdog time will count before it resets the processor.
Before executing any part of the code, set the timeout period of the watchdog to be the expected execution time for that part, usually the function that sets the timeout is called ***Watchdog_sleep***.

If the code executed in the expected time, then the watchdog would be refreshed again and a new timeout value would be assigned for the next part of the code.

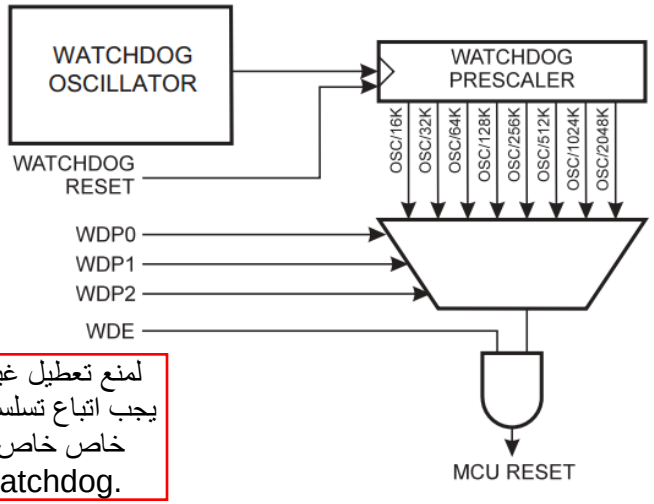
If the code is executed in a time more than expected, then the timeout would be passed and the watchdog would reset the processor.



By this way you will make sure that the processor ***would never halt*** in a code !

AVR Watchdog Timer

- The Watchdog Timer is clocked from a separate On-chip Oscillator which runs at 1MHz.
- To prevent unintentional disabling of the Watchdog, a special turn-off sequence must be followed when the Watchdog is disabled.



لمنع تعطيل غير مقصود يجب اتباع تسلسل إيقاف خاص خاص عندما يتم تعطيل Watchdog.

WDP2	WDP1	WDP0	Number of WDT Oscillator Cycles	Typical Time-out at V_{CC} = 3.0V	Typical Time-out at V_{CC} = 5.0V
0	0	0	16K (16,384)	17.1 ms	16.3 ms
0	0	1	32K (32,768)	34.3 ms	32.5 ms
0	1	0	64K (65,536)	68.5 ms	65 ms
0	1	1	128K (131,072)	0.14 s	0.13 s
1	0	0	256K (262,144)	0.27 s	0.26 s
1	0	1	512K (524,288)	0.55 s	0.52 s
1	1	0	1,024K (1,048,576)	1.1 s	1.0 s
1	1	1	2,048K (2,097,152)	2.2 s	2.1 s

Lab 1

Write a C Code to demonstrate the functionality of the watchdog timer in AVR atmega32 microcontroller, initially the code will turn on a LED for 500msec. Then turn off the LED and enable the watchdog timer with a timeout configured to 2.1 seconds. Then enters an infinite empty loop !

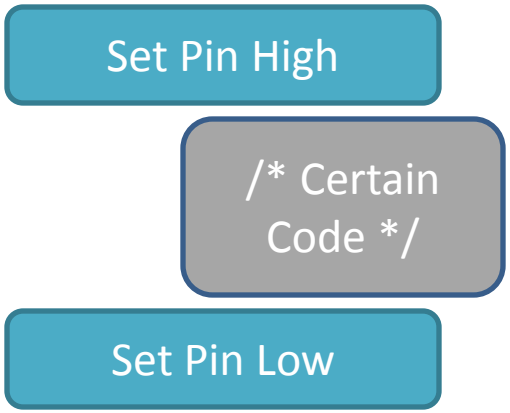
After 2.1 seconds if the LED is powered on again, it means that a system reset has been done.

Execution time measurement

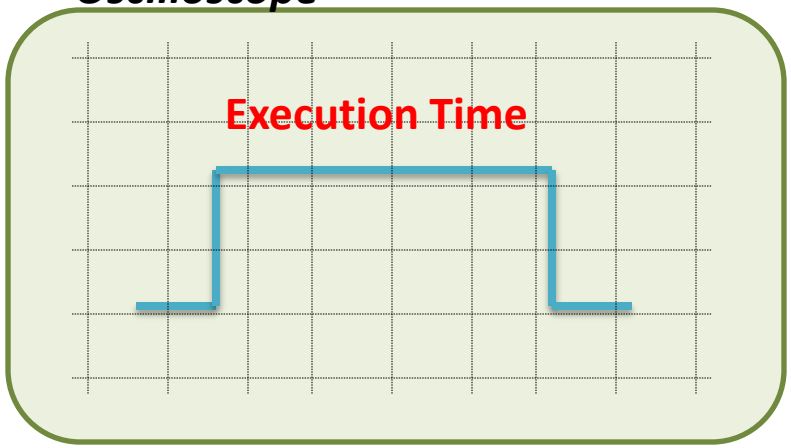
There are many ways to measure the execution time of a certain code, one way is:

By using Oscilloscope

- Set a pin a high before the code to be measured
- Set the pin low after the code to be measured
- Measure the width of the high part which represent the code execution time



Oscilloscope

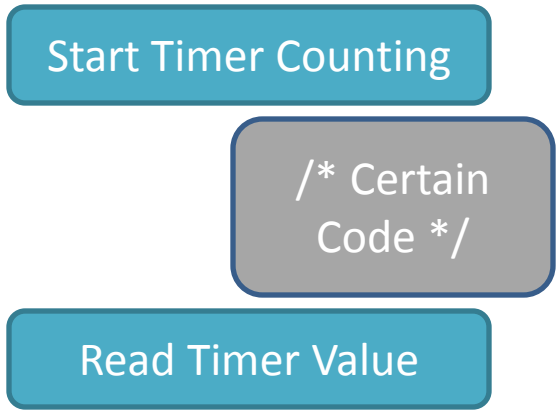


Execution time measurement

There are many ways to measure the execution time of a certain code, one way is:

By using Timer

- Start a timer before the code to be measured
- Read the timer value just after the code to be measured
- Calculate the execution time of the code according to the following equation



$$\boldsymbol{Execution\ Time} = Timer\ Value\ x\ Timer\ Tick\ Time$$

Execution time measurement

There are many ways to measure the execution time of a certain code, one way is:

By using Listing File

The listing file (.lss) is a generated file by the liker that shows the assembly generated for each line of the code, for each generated assembly line, the listing file writes the number of clock cycles for that line.

Count the number of the clock cycles for the part of the code you need to measure, then multiply this count by the period of the clock you use (period of the crystal oscillator used).

The End …

IMT SCHOOL

www.imtschool.com

ww.facebook.com/imaketechnologyschool/