



Synchronizing Execution to Real-Time by Polling TMR0

Author: Mark Sullivan Joplin, MO USA msullivan@niobrara.com

INTRODUCTION

Polling TMR0 to synchronize a process with real-time is a common technique. Usually, the timer is reset each time so it repeats a specified period, polling the timer with code like:

WAIT	BTFSS	TMR0,7
	GOTO	WAIT

This method has a slightly unpredictable latency depending on which of the three instruction cycles within the loop the timer expires on. If the timer is reset by code like:

MOVLW	\$80-delay
MOVWF	TMR0,F

The latency error accumulates. Replacing the previous reset code with:

```
MOVLW $80-delay + 3
ADDWF TMR0,F
```

avoids this cumulative drift of real-time synchronization. The +3 to run is required to make up for the latency in the ADDWF TMR0 instruction. So, the complete timersynchronous wait loop is as below:

WAIT	BTFSS GOTO	TMR0,7 WAIT
	MOVLW ADDWF :	\$80-delay + 3 TMR0,F
	:	
	:	
	GOTO	WAIT

Microchip Technology Incorporated, has been granted a nonexclusive, worldwide license to reproduce, publish and distribute all submitted materials, in either original or edited form. The author has affirmed that this work is an original, unpublished work and that he/she owns all rights to such work. All property rights, such as patents, copyrights and trademarks remain with the author.