

Entregable 2 (Grups 20 i 50)

Cronograma instruccions ASM (`__delay_ms`)

Quan programeu un microcontrolador en ocasions us caldrà fer esperes d'un cert temps determinat. Quan comenceu a programar en llenguatge C, veureu que el compilador de C us ofereix una funció que permet fer un delay d'un temps especificat en milisegons: aquesta funció es diu "`__delay_ms()`". Per a poder fer-la servir, cal indicar-li al compilador quina és la freqüència F_{osc} a la que fem treballar el nostre micro. Per a indicar-li, haurem de definir una etiqueta anomenada `_XTAL_FREQ` amb el valor en Hz.

El manual del compilador de C (XC8 User Guide) ens diu el següent:

This is an inbuilt function that is expanded by the code generator. When called, **this routine expands to an in-line assembly delay sequence**. The sequence will consist of code that delays for the number of milli seconds specified as the argument. The `__delay_ms()` builtin function **can use loops and the nop instruction to implement the delay**.

A la taula del final d'aquesta pàgina, us he posat el codi ensamblador generat en la compilació de la funció `__delay_ms`, per a un valor de milisegons determinat (el qual no us dic a l'enunciat, i l'haureu de calcular). Si us pregunteu com he obtingut aquest codi: quan compileu, el Proteus genera un fitxer .hex amb el codi hexadecimal que es grava a la memòria de programa. He agafat aquest fitxer .hex i l'he passat per un *disassembler* (programa que tradueix dels fitxers .hex a un llistat llegible d'instruccions ASM).

En aquest exercici es tracta de que:

- 1) dibuixeu el **cronograma dels cicles de Fetch i Execute** d'aquest codi d'instruccions ensamblador, i
- 2) **calculeu quant temps dura l'execució completa d'aquest codi**. Per a poder-ho calcular, necessitareu saber que $F_{osc} = 8 \text{ MHz}$. Per a comptar el temps d'execució total, no comptabilitzeu el primer Fetch.

Address	Opcode	Disassembly
7FF0	0E03	MOVLW 0x3
7FF2	6E01	MOVWF 0x1, ACCESS
7FF4	0E98	MOVLW 0x98
7FF6	2EE8	DECFSZ WREG, F, ACCESS
7FF8	D7FE	BRA 0x7FF6
7FFA	2E01	DECFSZ 0x1, F, ACCESS
7FFC	D7FC	BRA 0x7FF6