



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona

Enunciat de la pràctica de laboratori

PWM

(Modulació per amplada de pols)

Pràctica PWM

1. Objectius

L'objectiu d'aquesta pràctica és capacitar a l'alumne en l'ús d'un *Interval Timer* com a dispositiu generador de senyals digitals modulats en amplada de pols (PWM, sigles de *Pulse Width Modulation*).

2. Coneixements previs de l'alumne

L'alumne ja ha de dominar els següents conceptes:

- L'arquitectura del PIC18F45K22
- El simulador PROTEUS
- La programació del PIC en C
- El funcionament dels ports d'E/S del PIC
- Programació de les interrupcions
- El funcionament del controlador del GLCD
- Els *Timers* del PIC18F45K22 com a unitats de mesura de temps

3. Pràctica base

L'objectiu és ser capaços de generar un senyal digital de període 1 mseg i que podem controlar el seu *duty-cycle*.

Per a tal fi, s'utilitzarà la unitat **CCP1** associada al *timer* **TMR2** per generar el senyal PWM mitjançant el maquinari del micro. El resultat haurà de ser visible en el LED RC2 observant que la intensitat de la seva llum és proporcional al % de *duty-cycle*. Aquest efecte de variació de la intensitat es pot observar en un LED real (placa EasyPIC), però malauradament el LED del Proteus no mostra aquest efecte (anirà oscil·lant entre apagat i encès, però no mostra canvis progressius en la intensitat).

Estructurareu la vostra feina en dues fases: una fase preliminar en la que aprendreu a configurar el CCP1 per a generar un PWM a un *duty-cycle* fix. I una segona fase en la que acabareu d'implementar la pràctica per tal que aquest *duty-cycle* vagi oscil·lant segons uns paràmetres.

3.1. Fase 1: *duty-cycle* fix

Com a primer exercici, genereu un senyal de PWM que té el període adequat (1 ms) i el *duty-cycle* és fix a un valor conegut.

El valor del % de *duty-cycle* dependrà de la vostra posició a la llista de subgrups (A o B) i del vostre subgrup:

- Si la vostra posició és parell i sou del subgrup A, el % *duty-cycle* serà 20
- Si la vostra posició és senar i sou del subgrup A, el % *duty-cycle* serà 30
- Si la vostra posició és parell i sou del subgrup B, el % *duty-cycle* serà 40
- Si la vostra posició és senar i sou del subgrup B, el % *duty-cycle* serà 50

Les llistes de subgrups estan al racó publicades el 18/09/2020:

<https://raco.fib.upc.edu/avisos/veure.jsp?id=110678&assig=GRAU-CI>

El primer alumne que apareix del subgrup 11A té la posició 1. El primer alumne del subgrup 11B té la posició 1, i així successivament. Mireu la vostra posició dins del vostre **subgrup**.

3.2. Fase 2: *duty-cycle* oscil·lant

Un cop hagueu comprovat que sabeu generar un PWM a un *duty-cycle* fix, afegiu la següent funcionalitat: el *duty-cycle* ha d'anar oscil·lant entre 0 i 100% cada 1.6 segons. És a dir, durant 0.8s la llum anirà augmentant d'intensitat i durant 0.8s anirà disminuint.

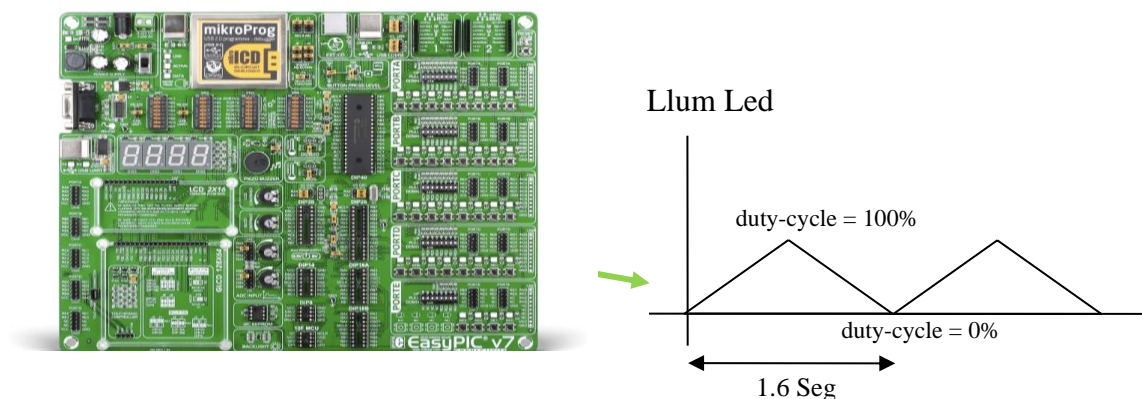


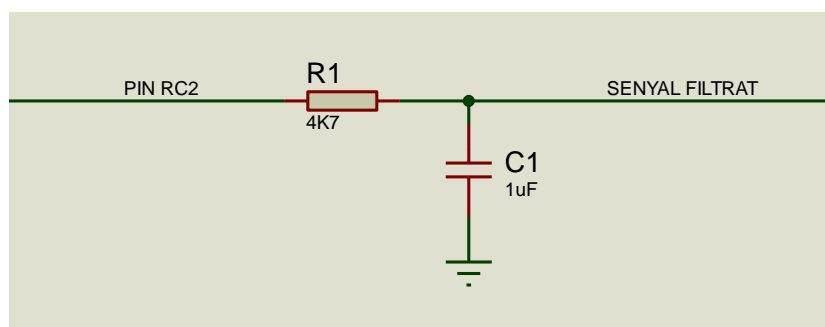
Fig 1. Senyal oscil·latòria per estimulació PWM del LED RC2

4. Treball previ

- **Lectura detallada de les seccions 14.0 i 14.3** (PWM mode) del *DataSheet* del PIC18F45K22 de Microchip. Consultar en general el capítol 14 (CCP module) per a trobar la informació que us calgui.
- **Respondre al qüestionari** adjunt que s'entregarà via telemàtica.
- **Elaborar dos projectes de Proteus** que facin les funcions requerides a cadascuna de les fases de l'enunciat. Simular el seu funcionament amb el Proteus per a comprovar la validesa.
- **Afegiu a cada projecte un oscil·loscopi virtual** per comprovar que el senyal generat té el període adequat (1 ms) i el *duty-cycle* desitjat.

Per a la **FASE 2**: en aquest cas, el *duty-cycle* va oscil·lant entre zero i 100% cada 1.6 segons. Quan mostreu la senyal de PWM a l'oscil·loscopi virtual, veureu que és molt complicat de distingir la oscil·lació que fa el *duty-cycle* cada 1.6 segons. Això és degut a que en 1.6 segons caben 1600 cicles de la nostra senyal PWM i al dibuixar tants cicles, l'oscil·loscopi es satura i no hi podem distingir les variacions amb claredat.

Per tant, per a mostrar la oscil·lació del *duty-cycle*, **posarem un filtre passa-baixes de tipus RC**. Així eliminarem les altes freqüències del PWM (1 kHz), i deixarem passar la baixa freqüència de la oscil·lació triangular (0.625 Hz). Tal com es veu a la següent figura, els valors seran: $R=4K7$, $C=1\mu F$. El condensador el podeu trobar a Proteus amb el nom **CAP**.



Entregueu al Racó com a PREVI un ZIP amb el següent:

- 1) Proteus de l'exercici de la Fase 1
- 2) Captura de pantalla d'oscil·loscopi de la Fase 1 (connectant la senyal del pin RC2). S'ha de distingir amb claredat que el període del PWM és el correcte (1ms) i que el *duty-cycle* és l'adequat segons la vostra posició de subgrup.

- 3) Proteus de l'exercici de la Fase 2
- 4) Captura de pantalla d'oscil·loscopi de la Fase 2 (connectant la senyal filtrada). S'ha de distingir amb claredat que la oscil·lació del *duty-cycle* dura 1.6 segons.
- 5) Qüestionari

5. Pràctica en el laboratori

El treball a realitzar en el laboratori consta dels següents apartats:

- 1) Mostrar el correcte funcionament del programa proposat sobre PROTEUS.
- 2) Comprovar el funcionament dels programes realitzats sobre la placa EASYPIC.
- 3) Comprovar amb l'oscil·loscopi que la freqüència base del senyal PWM és d'1 KHz.
- 4) Realitzar l'ampliació proposada pel professor a classe.

Qüestionari Pràctica PWM

Grup	Noms i Cognoms
21	Ricard Guixaró Tranco

1) Quants bits té el registre complet CCPRx? i el registre CCPRxL?

El registre CCPRX té 10 bits, els 8 de més pes van al CCPRxL, i els altres dos a DCxB de CCPxCON.

2) Quin és el registre amb el que es pot controlar el període del senyal CCPx?

El PRx, en aquest cas el PR2.

3) Quin és el registre amb el que es pot controlar el *duty-cycle* del senyal CCPx?

El CCPRxL (té els 8 bits de més pes) i el DCxB de CCPxCON (té els 2 bits de menys pes).

4) Com estan relacionats el període (T) del senyal PWM i el màxim nombre possible de *duty-cycles* diferents (N) que es poden generar?

A mesura que es fan gran el període, més *duty-cycles* hi podran haver.

5) Amb quin valor heu programat el registre PR2?

Amb el valor 124. $0.001 = (PR2+1)*4*16*(1/8*10^{-6}) \rightarrow PR2 = 124 (0X7C)$

6) Cada quants milisegons incrementeu el valor del *duty-cycle*? En quantes unitats?

No se el número de segons que passen, faig un: `int i = 1000; while(i > 0);`. Si es pogués utilitzar el `_delay(x)`, aleshores ho usaria i sabria la quantitat concreta. Augmenta en 5 unitats.