

METODO BASICO PARA IMPLEMENTAR UN CONTROLADOR DIGITAL PID EN UN MICROCONTROLADOR PIC PARA DESARROLLO DE APLICACIONES A BAJO COSTO

(Aplicaciones en control de Potencia y la Industria)

Ilber Adonayt Ruge Ruge
Universidad de Cundinamarca
Grupo de Investigación en Tecnologías de la Información y las Comunicaciones
GITEINCO
Fusagasuga – Colombia
ilberruge@yahoo.es - iruge@unicundi.edu.co

RESUMEN

Dado el amplio uso de los controladores PID en el ámbito industrial (control de potencia en motores de inducción, control de nivel, caudal y presión en procesos químicos entre otros), el uso de microcontroladores para el desarrollo de este tipo de aplicaciones ha tomado fuerza gracias a la incorporación de lenguajes de alto nivel que facilitan ampliamente este tipo de implementaciones, además de los bajos costos de adquisición de estos dispositivos, distribución de software de desarrollo gratuito y amplia información en la Internet.

Este artículo tiene como propósito mostrar al lector un método de diseño práctico y sencillo en el desarrollo de controladores digitales PID implementados en microcontroladores PIC. Para tal fin, se tomara como ejemplo un sistema de calefacción (*heater*) al cual se le realizara inicialmente un modelamiento basado en el método de curva de reacción, calculo de parámetros y/o constantes de control PID mediante técnica de sintonización Ziegler–Nichols y por ultimo se mostrara la metodología de implementación de este controlador en un microcontrolador PIC16F877A, utilizando como herramienta de validación el software de simulación ISIS Proteus.

Palabras Claves: Controlador PID, Microcontrolador, C CCS Compiler, ISIS Proteus.

CONTROLADOR PID

Es interesante señalar que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID o PID modificado. Los controladores PID analógicos, son principalmente de tipo hidráulico, neumático, electrónico, eléctrico o sus combinaciones. En la actualidad, muchos de estos se transforman en formas digitales mediante el uso de microprocesadores [1]. Se puede indicar que un controlador PID responde a la siguiente ecuación:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{\partial e(t)}{\partial t} \quad (1)$$

donde $e(t)$ es el error de la señal y $u(t)$ es la entrada de control del proceso. K_p es la ganancia proporcional, T_i es la constante de tiempo integral y T_d es la constante de tiempo derivativa.

En el dominio de la frecuencia, el controlador PID se puede escribir como:

$$U(S) = K_p \left(1 + \frac{1}{T_i S} + T_d S \right) E(S) \quad (2)$$

SINTONIZACION DE CONTROLADOR MEDIANTE ZIEGLER-NICHOLS

En lazo abierto, muchos procesos pueden definirse según la siguiente función de transferencia:

$$G(s) = \frac{K_0 e^{-s\tau_0}}{1 + \gamma_0 s} \quad (3)$$

Donde los coeficientes K_0 , τ_0 y γ_0 se obtienen de la respuesta del sistema en lazo abierto a una entrada escalón. Se parte del sistema estabilizado en $y(t) = y_0$ para $u(t) = u_0$. Se aplica una entrada escalón de u_0 a u_1 (el salto debe estar entre un 10% y un 20% del valor nominal) y se registra la respuesta de la salida hasta que se estabilice en el nuevo punto de operación [2].

Los parámetros se pueden obtener de la respuesta mostrada en la Figura 1:

$$\begin{aligned} \tau_0 &= t_1 - t_0 \\ \gamma_0 &= t_2 - t_1 \\ k_0 &= \frac{y_1 - y_0}{u_1 - u_0} \end{aligned} \quad (4)$$

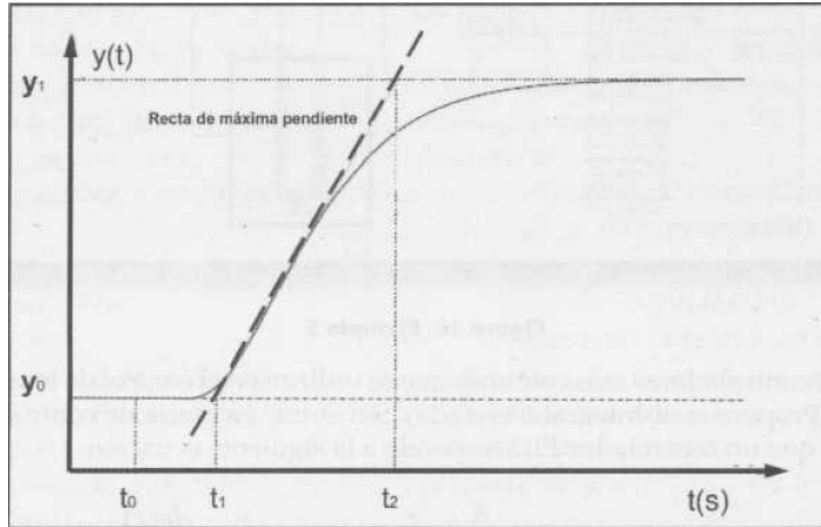


Figura 1. Respuesta de salida ante una entrada escalón.

Según Ziegler-Nichols, la relación de estos coeficientes con los parámetros del controlador son:

$$K_p = 1.2 \frac{\gamma_0}{k_0 \tau_0} \quad T_i = 2\tau_0 \quad T_d = 0.5\tau_0 \quad (5)$$

CONTROLADOR DIGITAL PID

La función de transferencia para el controlador PID digital se convierte en [3]:

$$U(z) = K_p \left[1 + \frac{T}{T_i(1-z^{-1})} + T_d \frac{(1-z^{-1})}{T} \right] E(z) \quad (6)$$

La función de transferencia discreta (6), también puede ser representada como [2]:

$$\frac{U(z)}{E(z)} = a + \frac{b}{1-z^{-1}} + c(1-z^{-1}) \quad (7)$$

Donde:

$$a = K_p \quad b = \frac{K_p T}{T_i} \quad c = \frac{K_p T_d}{T} \quad (8)$$

Existen distintas posibilidades de la realización práctica de un controlador PID, una de las mas habituales es la realización en paralelo:

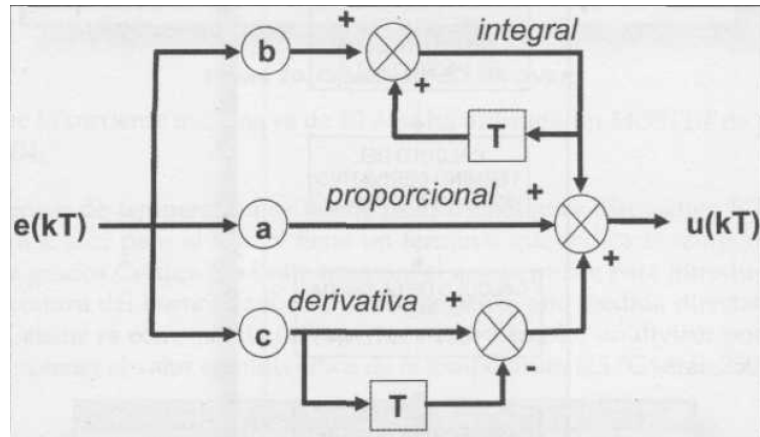


Figura 2. Diseño paralelo de controlador PID.

ALGORITMO DE PROGRAMACION EN MICROCONTROLADOR

El algoritmo utilizado para programar el PIC se muestra en la Figura 3. El muestreo (T) debe ser mayor que el tiempo de establecimiento del sistema en lazo abierto. En el modelo Ziegler-Nichols se toma un valor $T < \tau_0 / 4$.

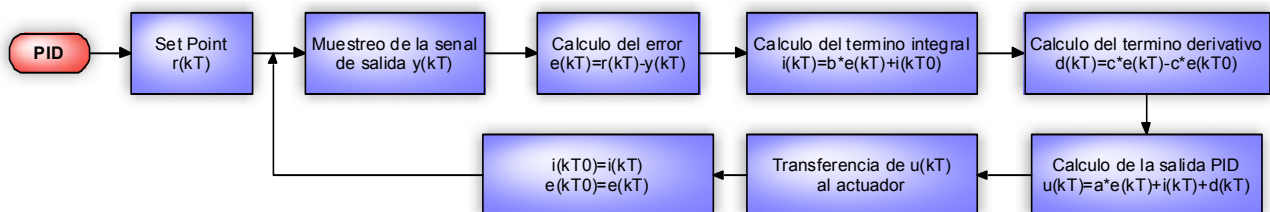


Figura 3. Algoritmo de programación de PID digital en microcontrolador.

Para la implementación y simulación del controlador PID se utilizara el software ISIS Proteus. Este software ofrece el modelo esquemático de un sistema de calefacción denominado **OVEN**, al cual se le pueden variar sus características funcionales tales como: Temperatura Ambiente de trabajo, resistencia térmica, constante de tiempo de establecimiento, constante de tiempo de calentamiento, coeficiente de temperatura y Potencia de calentamiento.

MODELAMIENTO DEL SISTEMA DE CALEFACCION

Para facilidades de simulación se establecerán los siguientes valores de parámetros funcionales del modelo OVEN:

Temperature Ambient (°C)= 25
Thermal Resistance to Ambient (°C/W)= 0.7
Oven Time Constant (sec)= 10
Heater Time Constant (sec)= 1
Temperature coefficient (V/°C)= 1
Heating Power (W)= 120

El modelo esquemático OVEN contiene un terminal sensor **T** que entrega un voltaje proporcional a la temperatura del sistema. De acuerdo a los parámetros establecidos anteriormente, este terminal entregara 1V/°C, es decir, que para una temperatura de 100°C, el terminal **T** entregara 100V.

Para obtener la respuesta del sistema en lazo abierto ante una entrada escalón (curva de reacción), se utiliza el sistema de análisis interactivo de ISIS Proteus *Interactive Analysis* (Graph Mode Tool), el cual interactúa con el sistema OVEN mediante el uso de un *Voltage Probe*¹ OV1(T), según se muestra en la Figura 4. Observar que para realizar la entrada escalon de 0V a 2V se utiliza un interruptor SW1.

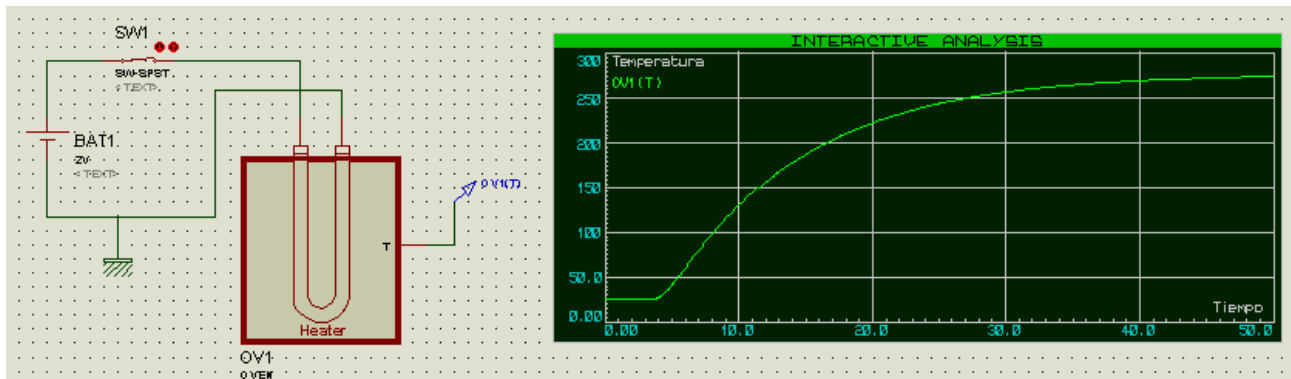


Figura 4. Esquemático para análisis de respuesta ante entrada escalón.

De la recta de máxima pendiente se deducen los parámetros τ_0 , γ_0 y k_0 definidos por el análisis en lazo abierto de Ziegler-Nichols.

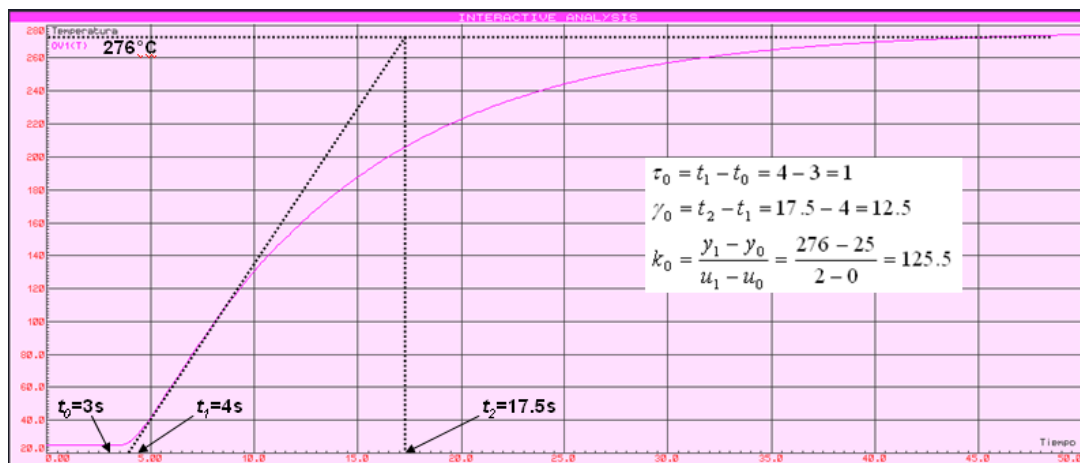


Figura 5. Determinación de los parámetros por método de curva de reacción.

Por tanto el modelo del sistema de calefacción queda definido así:

$$G(s) = \frac{K_0 e^{-s\tau_0}}{1 + \gamma_0 s} = 125.5 \frac{e^{-s}}{1 + 12.5s} \quad (9)$$

¹ Hacer click derecho sobre el Interactive Analysis, seleccionar Add Traces Ctrl+A y anexar el OV1(T).

Los parámetros K_p , T_i y T_d se calculan según la Regla de Sintonización de Ziegler-Nichols basada en la respuesta al escalon [1]:

$$K_p = 1.2 \frac{\gamma_0}{k_0 \tau_0} = 0.1195 \quad T_i = 2\tau_0 = 2 \quad T_d = 0.5\tau_0 = 0.5$$

Reemplazando los valores de K_p , T_i y T_d en las ecuaciones dadas en (8), y estableciendo un periodo de muestreo $T=0.1s$ según criterio $T < \tau_0 / 4$, los parámetros del controlador discreto son:

$$a = K_p = 0.1195 \quad b = \frac{K_p T}{T_i} = 0.0062 \quad c = \frac{K_p T_d}{T} = 0.6215$$

IMPLEMENTACION DE CONTROLADOR DIGITAL PID EN MICROCONTROLADOR

Consideraciones Preliminares

- Se requiere de un conversor Análogo/Digital para ingresar el voltaje entregado por el sensor T del modelo OVEN. Dado que dicho terminal entrega $1V/^{\circ}C$, se utiliza un divisor de voltaje para adecuar la señal del sensor a niveles permitidos por el conversor A/D; en este caso se utiliza un divisor de voltaje con factor 100, es decir, que para una temperatura de $100^{\circ}C$ ($100V$) el conversor recibirá $1V$.
- El control se realiza mediante una señal PWM proporcional a la señal generada por el controlador.

Por tanto el uso del microcontrolador PIC16F877A [6] es pertinente para la aplicación descrita.

Para la realización del código fuente se utilizó el compilador C CCS Compiler [4], [5]. De acuerdo al algoritmo planteado en la Figura 4 para programación de PID en el microcontrolador, el código fuente propuesto es:

CODIGO FUENTE EN LENGUAJE C PARA CONTROLADOR PID DISCRETO

```
#INCLUDE <16F877A.H>
#DEVICE ADC=10
#USE DELAY(CLOCK=4000000)
#FUSES XT,NOWDT

VOID MAIN(){

INT16 valor, control;
FLOAT a,b,c;
FLOAT TEMPERATURA_LIMITE;
FLOAT rt,eT,iT,dT,yT,uT,iT0,eT0;
FLOAT max,min;

//Variables para lectura de ADC y señal de Control a modulo CCP
//Constantes para parámetros de controlador PID
//Referencia de Temperatura
//Variables de controlador PID
//Variables para anti-windup

min=0.0;
max=1000.0;
iT0=0.0;
eT0=0.0;
a=0.1243;
```

```

b=0.0062;
c=0.6215;
TEMPERATURA_LIMITE=1200.0;    //Set Point r(kT)= 120°C

setup_timer_2(t2_div_by_4,249,1);    //Configuracion de Timer 2 para establecer frec. PWM a 1kHz
setup_ccp1(ccp_pwm);    //Configurar modulo CCP1 en modo PWM
setup_adc_ports(all_analog);    //Configurar ADC
setup_adc(adc_clock_internal);
set_adc_channel(0);    //Seleccionar Canal 0 para sensor de Temperatura

while(true){

valor=read_adc();    //Leer ADC
yT=5000.0*valor/1024.0;    //Escalizar señal de salida y(kT)
rT=TEMPERATURA_LIMITE;
eT=rT-yT;    //Calcular senal de error e(kT)
iT=b*eT+iT0;    //Calcular termino integrativo i(kT)
dT=c*(eT-eT0);    //Calcular termino derivativo d(kT)
uT=iT+a*eT+dT;    //Calcular senal de control u(kT)

if (uT>max){    //Anti-windup
uT=max;
}
else {
if (uT<min){
uT=min;
}
}

control=uT;

set_pwm1_duty(control);    //Transferencia de senal de control al actuador
iT0=iT;
eT0=eT;
delay_ms(100);    //Periodo de muestreo T=0.1s

}
}

```

DIAGRAMA ESQUEMATICO Y SIMULACION DE CONTROLADOR PID EN ISIS PROTEUS

Para la generación de perturbaciones se utiliza el conmutador *SW1*, el cual permite cambiar el valor de la fuente de alimentación al sistema OVEN. Para el instante de tiempo $t=0s$. se utilizo la fuente de alimentación BAT2 de 12V, para el instante de tiempo $t=60s$ se conmuta a la fuente de alimentación BAT1 de 24V y para el instante de tiempo $t=105s$ se conmuta nuevamente a la fuente de alimentación BAT2 de 12V. Como se puede apreciar en la Figura 6, el controlador PID responde de manera muy eficiente, siguiendo la referencia de 120°C establecido en el código fuente.

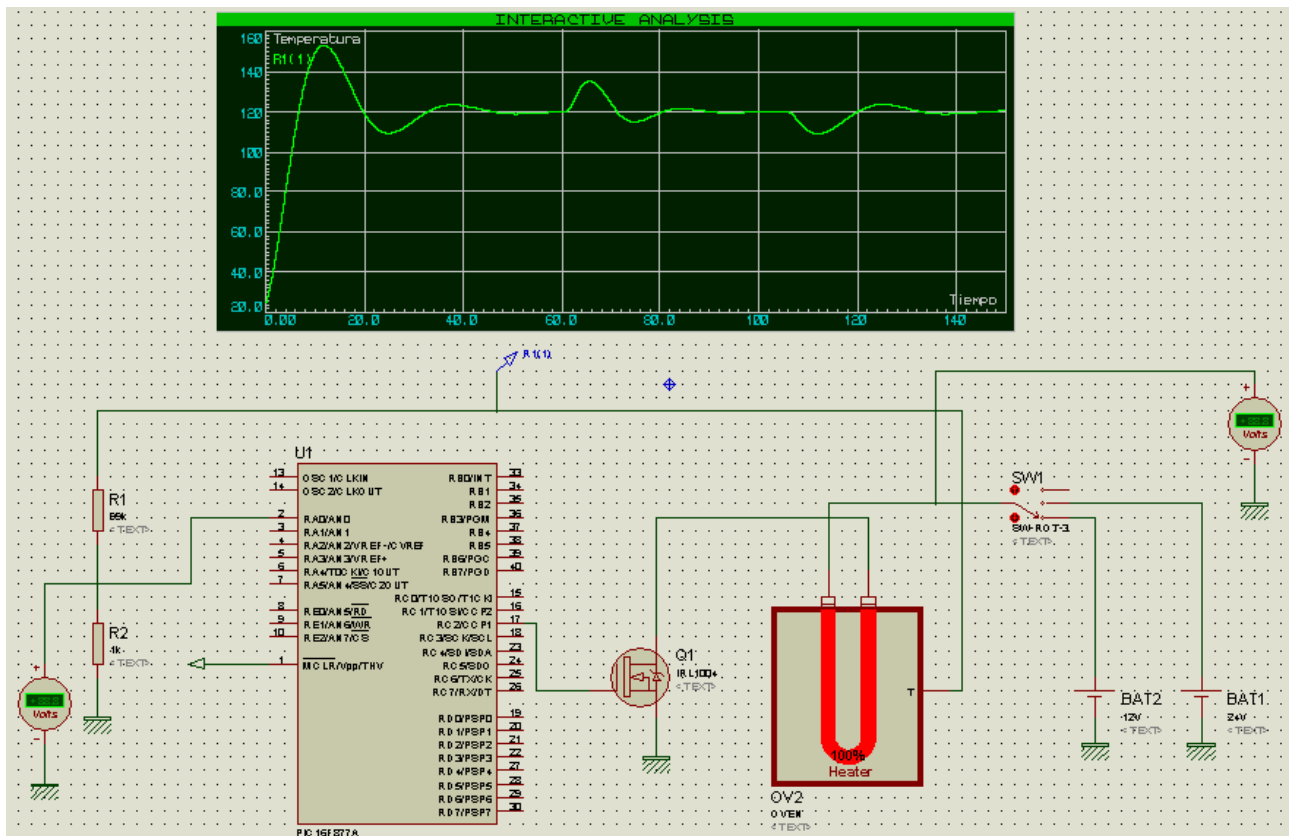


Figura 6. Diagrama esquemático de controlador PID en microcontrolador y resultado de la simulación mediante Interactive Analysis.

OBSERVACIONES

Las resistencias R1 y R2 son de 99k y 1k respectivamente. La configuración mostrada en la Figura 6 permite obtener un divisor de voltaje con factor 100.

La escalización de la señal de salida $y(kT)$ en el código fuente se debe multiplicar por un factor de 100 debido al factor de división que se utilizó para adecuar la señal del sensor del sistema OVEN.

La simulación inicia con una magnitud de temperatura igual a 25°C en $t=0s$, dado que este valor es la temperatura ambiente de trabajo del sistema OVEN.

Si desea hacer mas interactivo el sistema de control propuesto, se puede variar el código fuente para que el valor de referencia $r(kT)$ o Set Point pueda ser ingresado por usuario a través de un teclado matricial o a través de un PC utilizando comunicación RS-232.

CONCLUSIONES

El uso de lenguajes de programación de alto nivel; como es el caso del compilador C CCS Compiler, facilita ampliamente la implementación de controladores PID u otro tipo de esquema de controlador sobre dispositivos microcontrolados, abriendo así una alternativa de desarrollo de sistemas de control a bajo costo, dado que este lenguaje permite fácilmente adicionar periféricos para ingreso de datos y supervisión de variables de estado del sistema controlado; como por ejemplo teclados matriciales y LCD's respectivamente, y además la implementación de sistemas de comunicación tales como USB, CAN, Ethernet, Bluetooth entre otros, ya son soportados por algunas gamas de microcontroladores y pueden ser administrados fácilmente a través de este tipo de compiladores.

Como bien se explica en el desarrollo de este paper, la tarea fundamental para el diseño de un controlador PID esta en la obtención de los parámetros del sistema que se desea controlar (planta), dado que a partir de estos parámetros se utilizan reglas de sintonización para hallar las constantes del controlador K_p , T_i y T_d . Una vez calculadas estas constantes podrá utilizar como Template el código fuente expuesto, y solo requerirá reemplazar el valor de las constantes del controlador PID y por supuesto el respectivo valor de Periodo de Muestreo, el cual esta en términos de solamente un retardo.

BIBLIOGRAFIA

- [1] Ogata, Katsuhiko. Ingeniería de Control Moderna. Controles PID e introducción al control robusto. Tercera edición. Pagina 669. Editorial Prentice Hall. 1998.
- [2] Garcia, Eduardo. Compilador C CCS y simulador Proteus para microcontroladores PIC. Editorial AlfaOmega. Pagina 154. 2008.
- [3] Ogata, Katsuhiko. Sistemas de control en tiempo discreto. Segunda edición. Pagina 116. Editorial Prentice Hall. 1996.
- [4] Canovas, Andrés. Manual de usuario del compilador PCW de CCS. Disponible en internet en http://www.cursos.ucv.cl/eie48700/referencias/CCS_C_Manual.pdf.
- [5] C Compiler Reference Manual March 2009. Disponible en internet en http://www.ccsinfo.com/downloads/ccs_c_manual.pdf.
- [6] PIC 16F87XA Datasheet. Disponible en internet en <http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>.