Control de velocidad y de posición de un motor de corriente continua

Téllez Prado, Solón.
Máster en ingeniería Mecatrónica Erasmus Mundus - EU4M
Universidad de Oviedo
Gijón, España

Resumen – A continuación, se describe la implementación de un código escrito en código C en la placa basada en el microcontrolador PIC16F886, PICDEM-MIMUO (tarjeta de demostración basada en microcontrolador PIC del Master en Ingeniería Mecatrónica de la Universidad de Oviedo) para controlar la unidad Feedback 33-100. Se desarrollan los algoritmos necesarios para ejecutar correctamente un control de velocidad en bucle cerrado y un control de posición en cascada. La documentación incluye la validación en el dispositivo y la descripción de todos los módulos y especificadas utilizadas en el microcontrolador PIC16f886.

Palabras clave – CCS C, PIC16f886, Motor DC, PIC16F886, control de posición en cascada.

I. INTRODUCCIÓN

Este documento describe la implementación del diseño de reguladores, desarrollo de algoritmos y diseño de circuitos necesarios para el control de un Unidad Mecánica Feedback, así como toda la documentación necesaria para comprender en términos generales el proceso seguido durante el desarrollo. También se aborda los cambios aplicados en los diversos trabajos previos para su integración y adaptación a los recursos del microcontrolador utilizado. Al final se explora los experimentos físicos en busca de la validación de los diseños implementados.

II. UNIDAD FEEDBACK 33-100

La Unidad Mecánica Feedback 33-100 (UMF a partir de ahora) dispone de un motor de corriente continua cuyo voltaje de funcionamiento va de los -15v a 15v, el motor es controlado por un driver MD22 de forma que con una señal PWM de 5v podemos controlar todo el rango de voltaje posible a proporcionar al motor. Se puede conocer la velocidad de giro del motor en cualquier momento midiendo el voltaje de salida del taco generador.

El motor está conectado por medio de una correa de reducción 32:1 al eje de salida, muestra su posición (8) utilizando una regla graduada de forma que es muy fácil medir la posición actual del mismo. Para realizar la medición de la posición también se cuenta con un potenciómetro conectado al eje de salida que proporciona un voltaje de 0v a 5v para todo el rango de posiciones.

La unidad UMF también cuenta con un potenciómetro que puede ser utilizado por el usuario como señal de referencia para los distintos controles a realizar.

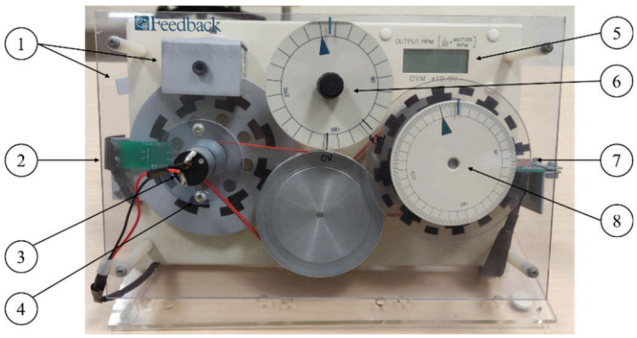

Figura 1. Unidad Mecánica Feedback 33-100

En la Tabla 1 se registran las características de las señales salida de los sensores, así como la señal de entrada.

Tabla 1. Terminales y características de las señales de entrada y salida de la unidad Feedback

SALIDA		
TERMINAL	SEÑAL	RANGO [V]
REF	Señal de salida del potenciómetro	0-5
POS	Señal de salida del transductor analógico de posición	0-5
VEL	Señal analógica emitida por el tacogenerador	-15 - +15
ENTRADA		
TERMINAL	SEÑAL	RANGO
PWM	Señal de entrada al driver del motor DC	0-5 (300Hz – 500Hz)

III. PCB CON MICROCONTROLADOR PIC16F866

Se precisa de una interfaz que permita conectar los distintos periféricos, adaptación de señales, entradas, salidas, así como la conexión entre la UMF y el PIC16f886. La PICDEM-MIMUO (tarjeta de demostración basada en microcontrolador PIC del Master en Ingeniería Mecatrónica de la Universidad de Oviedo) es la encargada de esta tarea.

El diseño de la PICDEM-MIMUO se realizo durante la asignatura de instrumentación electrónica, por tanto, no es necesario conocer a profundidad el proceso de diseño de la

mismo. Si es necesario para la realización de este trabajo conocer las conexiones, el circuito (Figura 2) y de los recursos que dispone.

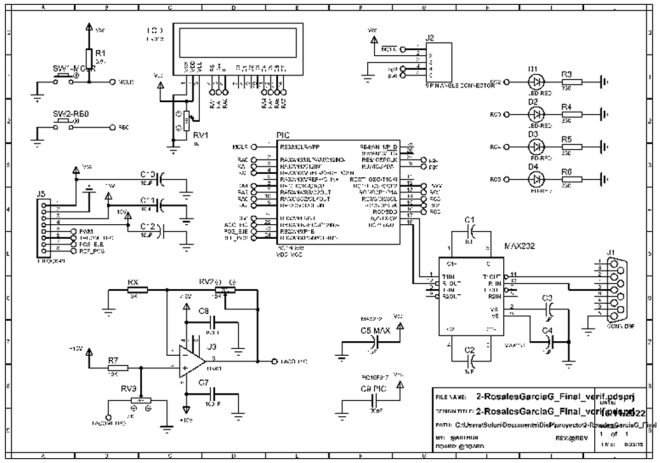

Figura 2. Esquema de conexiones de PICDEM-MIMUO.

Cuenta con un puerto para la conexión con las entradas y salidas de la UMF, un puerto de comunicación serie asíncrona DB9, puerto de conexión para el PicKit3, 4 LEDs indicadores, una pantalla LCD de 16 caracteres por dos líneas con driver integrado y dos pulsadores, uno de reset y otro de propósito general.

En la tabla 2 se describe a detalle las conexiones de cada uno de los componentes.

Tabla 2. Tabla de conexiones al PIC16F886

REFERENCIA	SEÑAL	PIN MICRO
LCD (Liquid Crystal Display)		
RS	Señal de control de memoria del LCD	RA1
RW	Señal de control Lectura/Escritura del LCD	RA2
E	Señal de control Enable del LCD	RA0
D4	Señal de datos 4 del LCD	RA4
D5	Señal de datos 5 del LCD	RA5
D6	Señal de datos 6 del LCD	RA6
D7	Señal de datos 7 del LCD	RA3
ENTRADAS ANALÓGICAS		
TACO_PIC	Señal acondicionada del tacogenerador de la UMF	RB1 / AN10
POS_EJE	Señal analógica del transductor de posición de la UMF	RB2 / AN8
REF_POS	Señal analógica del potenciómetro de la UMF	RB3 / AN9
PICKIT 3		
MCLR	Señal del Master Clear del Pickit 3	RE3/MCLR
PGC	Señal de reloj Pickit 3 (ICSPCLK)	RB6
PGD	Señal de datos Pickit 3 (ICSPDAT)	RB7
SWITCHS		
SW1-MCLR	Señal de entrada Pulsador RESET	RE3/MCLR
SW2-RB0	Señal de entrada Pulsador RB0 (Utiliza resistencia Pull-Up interna del micro)	RB0
LEDs		
D1	Señal de salida LED D1	RC2
D2	Señal de salida LED D2	RC3
D3	Señal de salida LED D3	RC4
D4	Señal de salida LED D4	RC5
COMUNICACIÓN SERIE RS232		
T1IN	Señal de transmisión del MAX232	RC6/TX
R1OUT	Señal de recepción del MAX232	RC7/RX
SEÑAL PWM		
PWM	Señal de salida PWM	RC1/CCP2
ALIMENTACIÓN PIC		
Vcc	Pista de alimentación +5V	Vcc/Pin 20
GND	Pista de masa	GND/Pin 8

IV. CONFIGURACIÓN DE LOS MÓDULOS INTERNOS

El algoritmo de control (detallado en el apartado VI) fue desarrollado para el trabajo de la asignatura de Computadores y Programación. Para que este se ejecute adecuadamente en el PIC16f886 es necesario configurar cada uno de los módulos internos a utilizar.

A. OSCILADOR INTERNO.

Durante la implementación del controlador, será necesario discretizarlo, de forma que realice los cálculos cada Tm segundos, un menor Tm indica que se realizan mas muestras por cada segundo transcurrido, uno de los objetivos de la implementación del algoritmo de control es minimizar en lo posible Tm.

Si bien es cierto que la optimización del código permite reducir el tiempo que toma ejecutar el algoritmo de control, la cual es la cota inferior para nuestro Tm, la velocidad a la que se ejecuta cada instrucción sigue dependiendo de la velocidad del oscilador conectado al PIC, por eso se opto por elegir la máxima frecuencia posible para el oscilador interno.

$$f_{osc} = 8\,MHz$$

B. COMUNICACIÓN SERIE ASÍNCRONA.

Para la comunicación serie fue necesario disponer de un circuito integrado MAX232, que permite establecer la comunicación entre un puerto serie RS232 a señales compatibles con TTL.

Se utilizo el modulo interno del PIC para la comunicación, de esta forma se hace uso del doble buffer físico, la información puede llegar mientras se esta ejecutando el algoritmo de control, el cual tiene la máxima prioridad, y solo leer del buffer cuando el microcontrolador disponga de tiempo para ello.

La comunicación se establece a 9600 baudios, a 8 bits y sin paridad.

Tabla 3. Configuración del módulo USART del PIC

CARACTERÍSTICA	CONFIGURACIÓN	ARGUMENTO
Velocidad de transmisión	9600 baudios	Baud=9600
Pin de transmisión (PIC)	PIN RC6	xmit=PIN_C6
Pin de recepción (PIC)	PIN RC7	rcv=PIN_C7
Tamaño de datos	8 bits	BITS=8
Uso del bit de paridad	Desactivado	PARITY=N
Interrupción por recepción dato serie	Desactivada	-

C. CONFIGURACIÓN DE LOS PUERTOS DE ENTRADA Y SALIDA DE PROPOSITO GENERAL.

En este apartado se hará la diferencia entre los pines de entrada/salida de propósito especial, como los pines RC6 y RC7 que se requieren si se hará uso del módulo interno de comunicación serie, y los pines de entrada y salida que simplemente reciben o envían niveles lógicos de voltajes a través de ellos sin hacer uso de ningún modulo interno (hardware dedicado).

Como lo describe la Tabla 2, se disponen de 4 leds conectados a los pines RC2, RC3, RC4 y RC5 que tienen que ser configurados como salida.

El pulsador de propósito general esta conectado al pin RB0, en el algoritmo este activa una interrupción cuando es pulsado, por la facilidad de detección del flanco fue conectado al pin RB0, pero bien podría ser conectado a los pines RB4 y RB5 que se encuentran libres y también son capaces de detectar interrupciones externas. También es necesario configurarlo como entrada activando las resistencias pull-up interne, ya que este no dispone de una en el circuito.

D. CONFIGURACIÓN DEL TIMER 1 (TMR1)

La una rutina de control es la porción del algoritmo que debe de tener la mayor prioridad, ejecutada a intervalos regulares muy estables.

La mejor elección es utilizar un timer, el PIC16f886 dispone de 3 de ellos. El timer 2 es requerido para el funcionamiento del módulo CCP2. El timer 0 es el que tiene un registro mas pequeño y nos ofrece menos flexibilidad al momento de configurar el tiempo de ejecución del algoritmo de control (relacionado a Tm). La opción ideal en este caso es el timer 1.

Inicialmente se diseñaron reguladores teniendo un Tm de 100ms, durante las pruebas en la PICDEM-MIMUO se determino que era posible reducir el Tm hasta 20ms.

Las características que elegimos para el timer 1 cumpla con el tiempo requerido son las siguientes:

Tabla 4. Configuración del módulo TMR1 del PIC16F886

CARACTERÍSTICA	CONFIGURACIÓN	ARGUMENTO
Señal de reloj	Oscilador interno 8 MHz	T1_INTERNAL
Prescaler	1:1	T1_DIV_BY_1
Interrupción por desbordamiento	Activada	-

Calculamos la pre-carga del TMR 1 para conseguir 20 ms de temporización:

$$Tiempo\ (s) = (2^{16} - PR1) \cdot PR_{TMR1} \cdot T_{instr} \quad (1)$$

Para la configuración elegida (Tabla 4) el tiempo máximo del temporizador es:

El tiempo de instrucción es:

$$T_{instr} = \frac{4}{f_{osc}} = \frac{4}{8\ MHz} = 0.5\ \mu s \quad (2)$$

Para la configuración elegida (Tabla 4) el tiempo máximo del temporizador es:

$$T\_max = (2^{16} - 0) \cdot 1 \cdot T_{instr} = 0.032\ seg = 32.79\ ms \quad (3)$$

Con los valores seleccionados, el tiempo de desbordamiento es mayor a Tm, no será necesario establecer un contador de desbordamientos para el timer 1.

Sustituyendo los parámetros, el nuevo tiempo y despejando PR1 en la ecuación (1) obtenemos que la carga para el timer 1 es:

$$PR1 = 2^{16} - \frac{Tiempo\ (s) \cdot f_{osc}}{4 \cdot PR_{TMR1}} = 65536 - \frac{20 \cdot 10^{-3} s \cdot 8 \cdot 10^{6} Hz}{4 \cdot 1} \quad (4)$$

$$\boldsymbol{PR1 = 25536}$$

E. CONFIGURACIÓN DEL MÓDULO CCP2 – SEÑAL PWM

Para controlar el voltaje aplicado al motor el PIC debe de enviar una onda PWM que recibirá el driver MD22 y actuará proporcional a este.

Como no se requiere de ninguna acción especial, mas que producir la señal PWM y teniendo en cuenta que el PIC trabaja con su reloj interno, se utilizara el modulo CCP2, dejando libre el modulo CCP1 para posibles ampliaciones que si requieran de sus funcionalidades especiales.

Los requerimientos de diseño precisan de una señal comprendida entre 300Hz a 500Hz. Como la diferencia en resolución para un PWM de 300Hz y uno de 500Hz no es significativamente grande, se infiere que la resolución no es una prioridad en esta implementación.

Se opto por configurar una onda de 500Hz, el fabricante especifica que el periodo del PWM del módulo CCP2 se determina con la siguiente ecuación:

$$T_{PWM} = \frac{4}{f_{osc}} \cdot (PR2 + 1) \cdot PR_{TMR2} \quad (5)$$

NOTA: El postscaler del TMR2 no se utiliza para el cálculo del periodo de la onda PWM

$$PR2 = \frac{T_{PWM}}{\frac{4}{f_{osc}} \cdot PR_{TMR2}} = \frac{\frac{1}{500Hz}}{\frac{4}{8 \cdot 10^6 Hz} \cdot 16} - 1 = 249 \quad (6)$$

El timer 2 debe de ser configurado para trabajar con un preescler de 16, un posescaler de 1 y una carga de 249 para generar un PWM de 500Hz.

Para modificar la PWM disponemos de la funcionalidad de modificar su duty cycle. La cantidad bits que almacenan el valor del duty cycle dependerá de la resolución configurada, esto se puede determinar de la siguiente manera:

$$T_{ON_{max}} = 2^N \cdot \frac{1}{f_{osc}} \cdot PR_{TMR2} \quad (7)$$

El número de bits que almacenan el valor del duty cycle es:

$$2^N = T_{ON_{max}} \cdot \frac{f_{osc}}{PR_{TMR2}} = \frac{1}{500\ Hz} \cdot \frac{8 \cdot 10^6 Hz}{16} \quad (8)$$
$$\rightarrow N = 9.96 \cong 10$$

Por tanto, necesitamos 10 bits para generar la onda PWM de 500Hz, pudiendo configurar un duty cycle con un entero que varíe entre 0 y 1023.

F. CONFIGURACIÓN TIMER 2 (TMR2)

El módulo CCP2 hace uso del timer 2 para su funcionamiento, se configura entonces el timer 2 según lo calculado en el apartado del módulo CCP2.

Tabla 5. Configuración del módulo TMR2 del PIC16F886

CARACTERÍSTICA	CONFIGURACIÓN	ARGUMENTO
Prescaler	1:16	T2_DIV_BY_16
Postscaler	1:1	1
Carga	249	249
Interrupción por desbordamiento	Desactivada	

G. ENTRADAS ANALÓGICAS

Es necesario realizar la lectura del taco generador, potenciómetro de referencia y el potenciómetro conectado al eje de salida. Estos dispositivos varían los niveles de voltaje entregado en función de la variable que miden (posición o velocidad), para poder operar con estos valores de voltaje en el PIC se deben de llevar a cabo conversiones analógicas para cada uno de ellos.

Como indica la tabla 2 nuestras entradas citadas en el párrafo anterior están conectadas a los pines AN8, AN9 y AN10. Es necesario indicarle al compilador que estos pines se utilizaran como entradas analógicas.

Recordando durante la ejecución del algoritmo que el PIC16f886 solo cuenta con un circuito de conversión, por lo que será necesario cambiar de cada en cada lectura analógica.

Tabla 6. Configuración del módulo A/D del PIC

CARACTERÍSTICA	CONFIGURACIÓN	ARGUMENTO
Puertos analógicos	AN8, AN9, AN10 activados	sAN8 | sAN9 | sAN10
Reloj de conversión	Oscilador interno	ADC_CLOCK_INTERNAL
Resolución	10 bits	#device adc= 10
Interrupción por fin de conversión A/D	No activada	-
Tensiones de referencia	0V-5V (Por defecto)	-

H. Pantalla LCD

Se utiliza la librería "LCD_PICDEM 2 PLUS." con modificaciones menores para que pueda ser compatible con la PICDEM-MIMUO.

Originalmente la librería estaba escrita para trabajar con el puerto D, es necesario redefinir la dirección del puerto al que hemos conectado la LCD, en este caso el PORTA y su registro de configuración TRISA.

En el caso particular de la LCD utilizada, no era posible escribir en los últimos 8 espacios de cada línea. Fue necesario intercambiar el orden de lectura de los 4 bits de mayor peso con los 4 de menor peso.

Las conexiones utilizadas fueron las que utiliza por defecto la librería.

LCD_ENABLE → PIN_A6
LCD_RS → PIN_A4
LCD_RW → PIN_A5
LCD_DATA4 → PIN_A0
LCD_DATA5 → PIN_A1
LCD_DATA6 → PIN_A2
LCD_DATA7 → PIN_A3

V. CONFIGURACIÓN DE LAS INTERRUPCIONES

Si bien muchos de los módulos utilizados cuentan con la posibilidad de habilitar interrupciones, solamente utilizaremos 3 de ellas, es necesario habilitar estas interrupciones y seguido habilitar la máscara de las interrupciones globales.

VI. ALGORITMO DEL PROGRAMA

El código escrito contiene todas las consideraciones descritas en cada uno de los apartados anteriores, desde las configuraciones iniciales y declaraciones hasta el algoritmo de

control desarrollado en la asignatura de Computadores y los reguladores diseñados en la asignatura de Sistemas de control.

Podríamos entender el código escrito como 3 algoritmos que trabajan en conjunto.

A. ALGORITMO DEL PROGRAMA PRINCIPAL

El programa principal, es el encargado de incluir todas las librerías necesarias, como las utilizadas para el procesamiento de cadenas, manejo de entradas y salidas, definiciones del pic16f886 y el manejo de la LCD. Luego declara las variables globales y define las estructuras a utilizar, se escriben los prototipos de las funciones y se definen las rutinas de interrupción. La mayoría de lo descrito anteriormente se ejecuta en tiempo de compilación.

La segunda parte del programa principal se encuentra en el main y es el encargado de declarar las variables locales a utilizar, inicializar todas las variables, configurar los timers, los tipos de entradas, las salidas, habilitar las interrupciones a utilizar e inicializar la LCD. Todas estas instrucciones solo se ejecutarán una única vez y es al inicio de la ejecución.

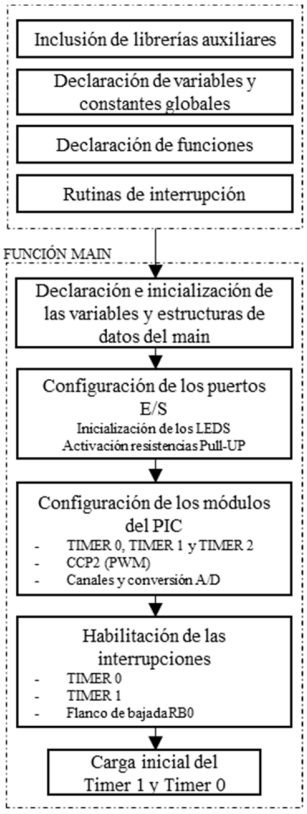

Figura 3 Diagrama de flujo del inicio de programa y la función main

B. ALGORITMO DEL BUCLE DE EJECUCIÓN CONTINUA DEL PROGRAMA PRINCIPAL.

Esta sección del código se ejecuta continuamente, pero con una baja prioridad. Se encarga de recoger los comandos ingresados por el usuario, únicamente cuando se encuentran disponible, procesarlos y guardarlos temporalmente hasta que el pulsador habilite los nuevos cambios.

Continúame este algoritmo verifica si ya se ha cumplido con el tiempo para el refresco de los datos en pantalla y la actualización del estado de los LEDs. El tiempo de refresco de los datos los determina n desbordamientos de la interrupción del timer 0.

Podemos observar que 2 interrupciones son utilizadas en este segmento del codigo, a priori se podría pensar que esto podria interferir con la ejecución del algoritmo de control, pero en el siguiente apartado se validara que realmente no tienen influencia significa en tiempo de ejecución.

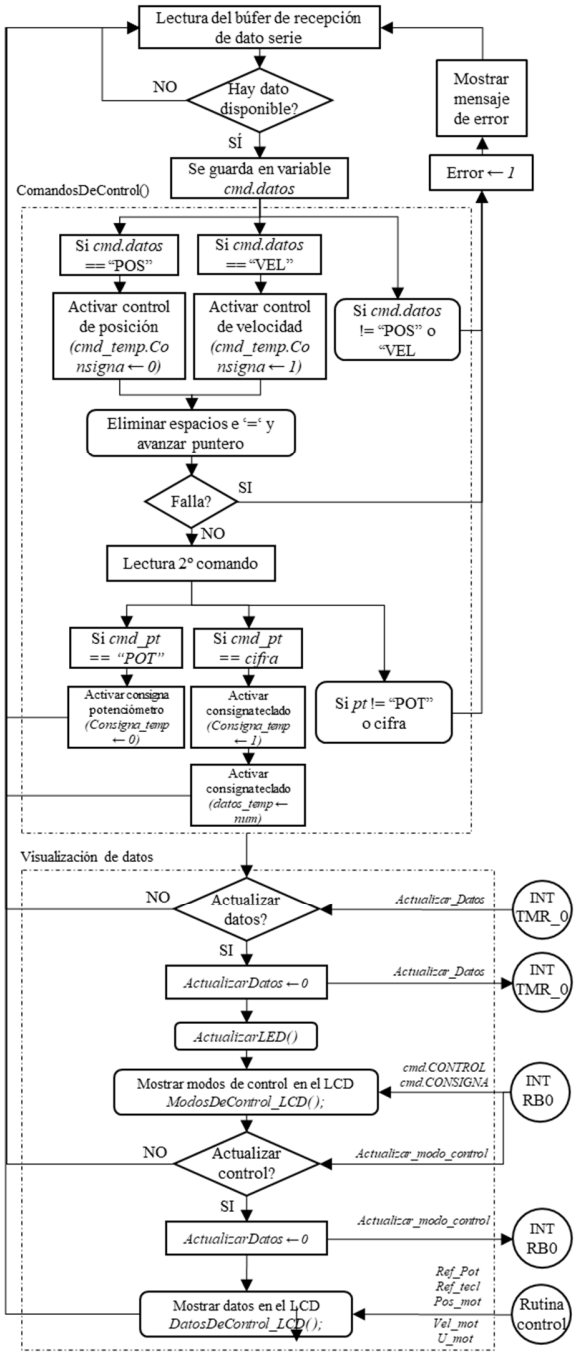

Figura 4. Diagrama de flujo del bucle de ejecución continua del programa principal

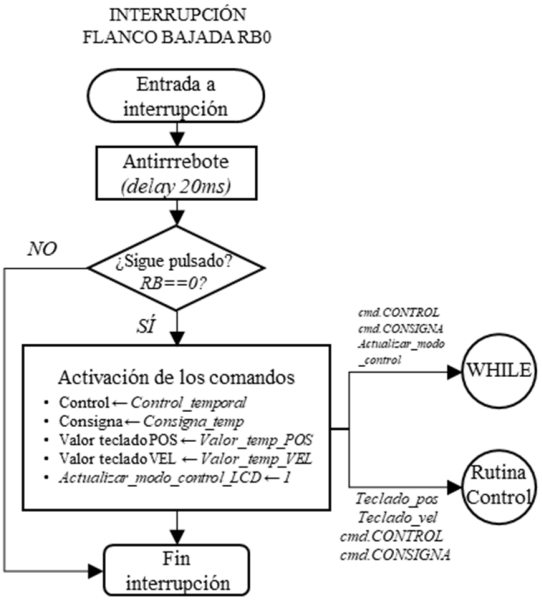

Figura 5. Diagrama de flujo de la interrupción por flanco de bajada del puerto RB0

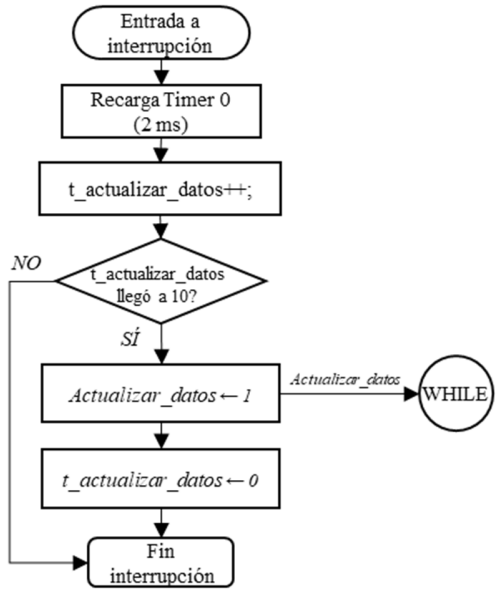

Figura 6. Diagrama de flujo de la interrupción de Timer 0

C. ALGORITMO RUTINA DE CONTROL

Es el algoritmo principal y al que se le da prioridad de ejecución. Este algoritmo recoge los valores de posición. velocidad y referencia de potenciómetro. Valida el tipo de control a utilizar, que tipo de referencia deberá de tomar, realiza las correcciones de paso por cero, busca el camino mas corto, calcula el voltaje a aplica al motor en función del regulador, aplica un limitador de voltaje, corrige la zona muerta de los sensores y del motor.

Para el caso del control posición primero realiza el calculo de este para poder utilizarlo de nuevo como input en el regulador de velocidad, recordar que el control de posición trabaja en cascada con este último.

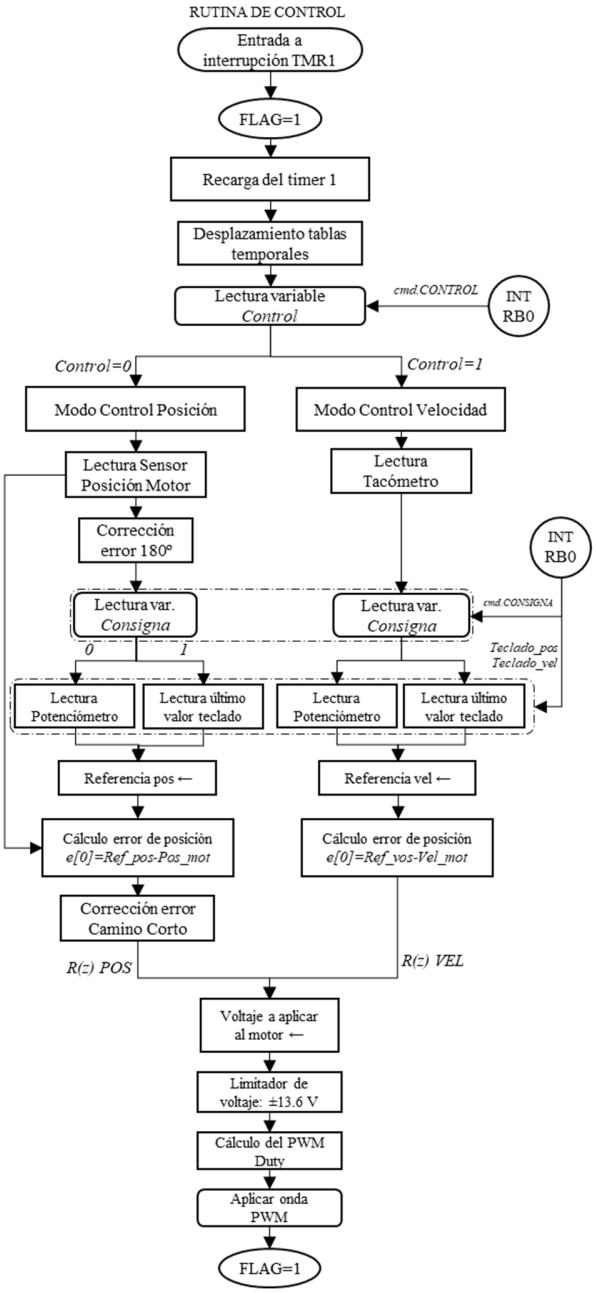

Figura 7. Diagrama de flujo de la rutina de control. Tm=20ms

VII. VALIDACIÓN DE LA IMPLEMENTACIÓN EN LA UMF

Se discretizo el regulador de velocidad diseñado en el anexo "TellezSolon_Reguladores.pdf", con un Tm de 20ms, dando como resultado la siguiente ecuación en diferencias:

$$u_k = u_{k-1} + 0.0742e_k - 0.0658e_{k-1} \qquad (9)$$

El control de posición es de tipo proporcional en cascada con el control de velocidad.

Se realizaron mediciones con osciloscopio y recopilación de datos con MATLAB y se obtuvieron los siguientes resultados en la unidad UMF número 4.

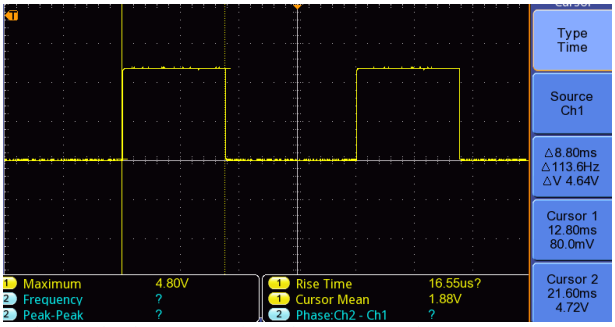

Figura 6. Tiempo de ejecución del algoritmo de control en velocidad=8ms y validación de Tm=20ms.

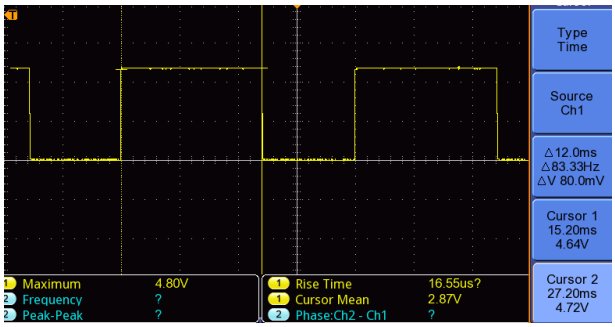

Figura 7. Tiempo de ejecución del algoritmo de control en posición=12ms y validación de Tm=20ms.

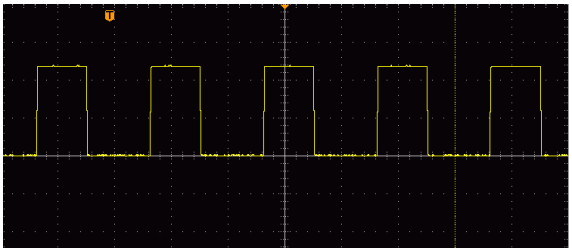

Figura 8. Periodicidad del algoritmo de control (sin interferencias en su periodo de ejecución).

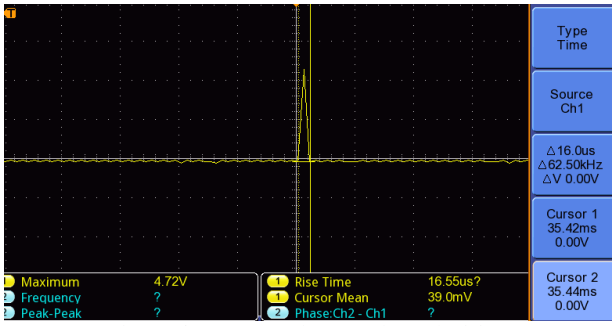

Figura 8. Interrupción del timer 0=16µs.

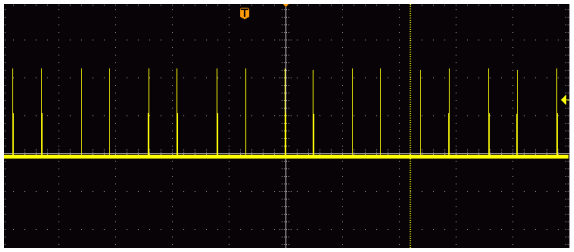

Figura 9. Periodicidad del timer 0 (sin interferencias entre este y el algoritmo de control).

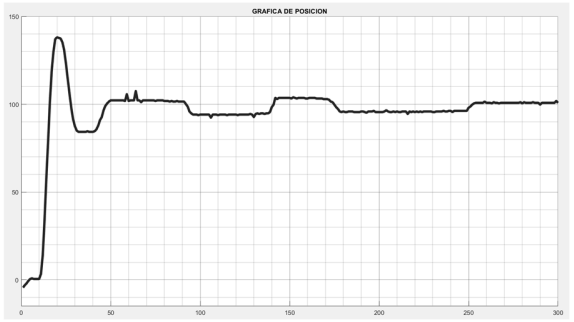

Figura 10. Entrada de un escalón de 100º en la UMF.

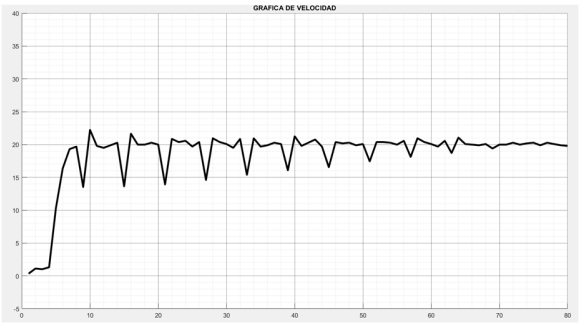

Figura 11. Entrada de un escalón de 20RPM en la UMF.

Dando validez a los resultados esperados durante la etapa de diseño.

VIII. REFERECIAS

[1]TecnoEdu (Tecnología Educativa) s.a., «tecnoedu.com,» [En línea]. Available: https://tecnoedu.com /Feedback/ 33001. php.

[2]Feedback Instruments Ltd, «Entrenador de Servos Analógicos Fundamentales - Modelo 33-002 (SFT154),» Fl Ltd, Crowborough .

[3]I. Álvarez, «Programación de control,» Universidad de Oviedo, Gijón.

[4]CCS Custom Computer Services, «CCS C Compiler Manual,» 2019.

[5]Microchip, «DataSheet PIC16F882/883/884/886/887,» [En línea].Available:http://ww1.microchip.com/downloads/en/devicedoc/41291d.pdf.