

## **CAPITULO 5. Desarrollo.**

### **5.1 Introducción.**

Existen en el mercado múltiples controladores de motores de inducción del tipo voltaje / frecuencia, todos ellos basados en microcontroladores específicos o en DSP (Digital Signal Processor) que reúnen varias características como:

- a) Regulación desde casi 0 hz hasta 60 hz.
- b) Controlan tanto la velocidad, como el sentido de un motor trifásico o monofásico.
- c) Poseen protección contra sobre corrientes y cortos circuitos.
- d) Vienen especificados de acuerdo a la potencia del motor que pueden manejar.

La tesis propone la construcción de un controlador de este tipo mediante el uso de microcontroladores microchip PIC. De hecho, en la propia página del sitio se encuentra una nota de aplicación referente a un controlador trifásico por relación voltaje / frecuencia mediante la técnica de PWM por muestreo regular (capítulo 4) cuyo autor es Padmaraja Yedamale, de Appliance Group.

Básicamente lo que se hizo fue modificar este programa pues para su uso en un motor monofásico presentaba los siguientes inconvenientes:

1. El proyecto original controlaba 3 fases, defasadas cada una 120 grados respecto a las demás, es decir, generaba 3 señales PWM a 0 hz, 120 hz y 240 hz, respectivamente.

2. La suma de las 3 senoidales debe regenerar los 240 volts RMS originales de la red trifásica, por lo que cada fase tiene una amplitud máxima de 80 volts RMS ( $80 * 3 = 240$  volts).
3. El proyecto original no contenía ningún indicador del voltaje o frecuencia que se estaba generando.
4. Al no poseer una fase referenciada respecto a tierra, no era posible controlar un puente H de transistores IGBT para uso monofásico.

Las modificaciones que se hicieron se describen a continuación:

## **5.2 Generación de una fase senoidal y señales de control para manejar un puente H, el cual controle un motor monofásico de inducción.**

El proyecto original genera 3 señales PWM por medio del PIC18F452, dos de ellas son a través de hardware utilizando los 2 módulos CCP, PWM incluidos en el chip y el tercer PWM es generado por software, utilizando el TIMER1 como referencia para ello. Cada señal PWM es generada de la siguiente manera:

Antes de pasar a la explicación de las señales generadas, es importante mencionar el porqué se uso el PIC 18F452:

- La frecuencia de reloj puede alcanzar los 40 Mhz, y cada instrucción se ejecuta a 100 nanosegundos, en este caso, la frecuencia se estableció en 20 Mhz, con cada instrucción ejecutada en 200 nanosegundos.
- A pesar de que solo posee dos módulos PWMs, la familia 18FXX2 de alto rendimiento posee un set de 75 instrucciones, que permite leer tablas de datos rápidamente, de esta manera se implementó la tabla de

valores senoidales. Además, posee un multiplicador en hardware de 8 x 8 bits, el cual requiere solo 1 ciclo de reloj @ 200 ns comparado contra los 69 ciclos @ 13.8 us si la multiplicación se realizara por software. Posee 18 fuentes de interrupciones las cuales se dividen en alta y baja prioridad, y 8 convertidores análogo – digital de alto muestreo y una resolución de 10 bits.

Todas las características mencionadas anteriormente permiten realizar el proyecto con eficiencia y facilidad sin la necesidad de recurrir a DSPs de alto costo.

Con una frecuencia de reloj de 20 Mhz, el TMR0 se carga con un preescaler de 16, y la tabla de valores senoidales contiene 19 datos, empezando desde 270 grados hasta noventa, en pasos de 10 grados. Como la tabla senoidal sólo contiene la mitad de los datos; se escribe ésta en la RAM dos veces, la primera en orden normal y la segunda al revés, dando todos los valores desde 270 grados hasta 260 grados nuevamente, por lo que se tiene que al final esta formada por 36 valores @ 10 grados cada uno.

El PWM tiene una frecuencia de 20 Khz, el valor de su ciclo útil esta definido por el valor del convertidor ADC multiplicado por los valores de la tabla senoidal, en un factor de 4; con lo cual se obtiene que para una frecuencia específica su voltaje deberá ser 4 veces mayor.

El funcionamiento global de la generación de la señal PWM es como sigue:

- Se lee el voltaje proporcionado por un potenciómetro que actúa como control de velocidad.
- Este voltaje se convierte a un valor digital y se asigna para la mínima velocidad una frecuencia de 5 hz y para la máxima 60 hz.

- El valor de la frecuencia es entonces multiplicado por la tabla de valores senoidales por un factor de 4, de manera cada valor de la senoidal genere el correspondiente ciclo útil del PWM.

### **5.3 Modificaciones al proyecto original.**

Dado que en el proyecto original se generan 3 PWMs separados entre sí 120 grados, era imposible controlar un inversor monofásico de 4 transistores, ya que el defasamiento entre ellas provocaría que los brazos condujeran al mismo tiempo, y esto a su vez provocaría un corto circuito. Para evitarlo, se necesitaba que la salida se manejara por semiciclos, es decir, que en un semiciclo condujera una sola rama del inversor y en el consecutivo, la siguiente rama. De esta manera, cada semiciclo tendría un efecto neto de generar un semiciclo positivo, seguido de uno negativo, produciendo finalmente un voltaje alterno real y en fase con el motor.

Se observó que por cada semiciclo de la señal PWM generada se obtienen 18 desbordamientos del TMR0, estos desbordamientos se cuentan para tener exactamente un pulso que dura lo que dura cada semiciclo.

Por lo tanto se cuentan 18 desbordamientos del TMR0, una vez que estos terminan, cambian de valor 2 bits, usados como control de la señal aplicada al puente H, es decir el primer BIT está en 1 durante el primer semiciclo mientras que en el segundo se encuentra en cero, al contrario sucede con el segundo que primero se encuentra en cero y después en uno, como se puede ver en las gráficas de las figuras 5.1 y 5.2.

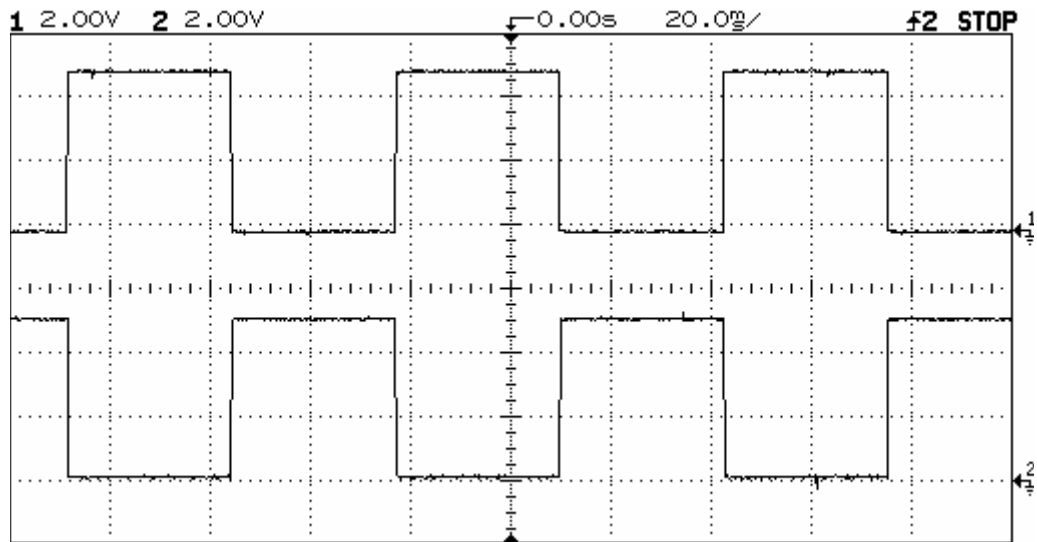


Figura 5.1. Semiciclos generados para controlar la conmutación del inversor.

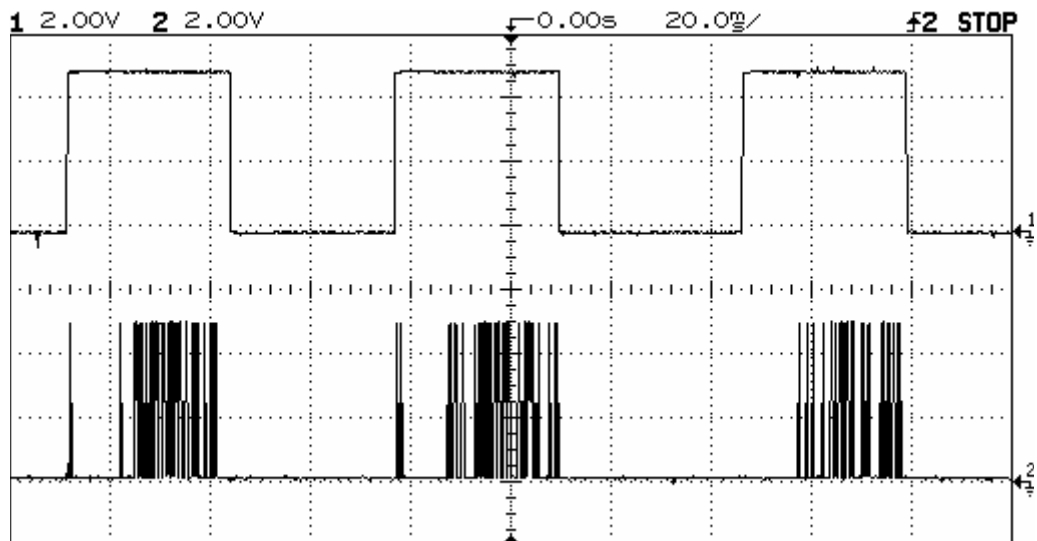


Figura 5.2. Generación del PWM senoidal para una rama del inversor, mediante el uso de los semiciclos.

El resultado neto, aplicado al puente H, genera la señal mostrada en la figura 5.3., que no es mas que una señal PWM senoidal con valores positivos y negativos; la cual, al

aplicarse al driver, y por consiguiente al inversor monofásico controlado por MOSFETs, genera el voltaje RMS deseado para controlar el motor.

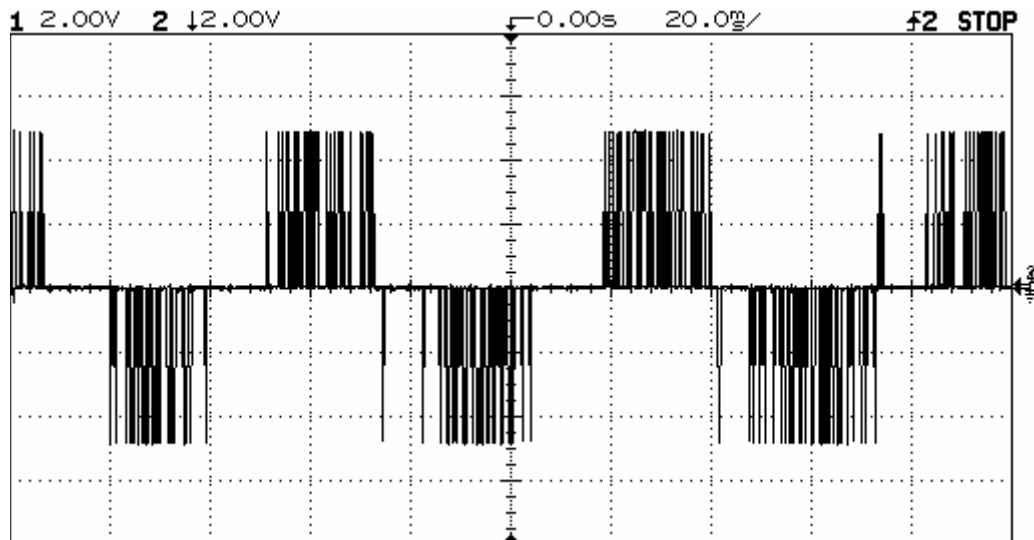


Figura 5.3. Resultado neto a la entrada del inversor.

Una vez implementado el procedimiento, se probó y observamos que la relación voltaje/frecuencia en el motor se cumplía solo para bajas frecuencias: a 60 hz, solamente obteníamos 80 volts RMS. Al principio pensábamos que era la eficiencia del circuito, pero analizando más a detalle, observamos que este valor provenía de las siguientes observaciones:

- a) La frecuencia máxima es de 60 hz, como para un motor trifásico la relación voltaje/frecuencia es de 4 (240v, 60 hz), el convertidor analógico-digital generaba valores desde 20 hasta 240. Este valor se multiplica por la máxima amplitud de la tabla de valores senoidales, la cual era de 222, la multiplicación nos daría 53280, en

binario seria: 1101000000100000; como solo se toman los 8 bits mas significativos, nos queda 11010000, que en decimal seria 208. Este resultado se graba en CCP1RL.

b) Como el ciclo útil del PWM es =  $(CCPR1L:CCP1CON)*Tosc*(TMR2 \text{ preescaler})$

y sólo se utiliza CCP1RL sin el CCP1CON, el máximo duty cycle sería:

$$\text{Sustituyendo: } PWM = 208*4*(1/20000000)*1 = 0.0000416$$

El periodo del PWM es de 0.00005.

Dividiendo para obtener el % nos da:  $0.0000416 / 0.00005 = 0.832 * 100 = 83.2 \%$ .

Si tenemos que el voltaje rectificado de los 110 volts es  $120 * 1.4142 = 169 \text{ V}$ .

Entonces el máximo voltaje multiplicado por el % del duty cycle será de 129 V.

$129 \text{ V} / 1.4142 = 91.5 \text{ VRMS}$  se deberían de obtener en la salida.

Considerando una eficiencia del sistema del 90% a lo mucho tendríamos 91.5

$V_{RMS}$  por 0.9 = 82.36 V RMS, que es lo mas cercano a nuestra hipótesis.

Por lo tanto, se debían subir los valores de la tabla senoidal para obtener un voltaje mayor, el máximo valor cambió de 222 a 245, con lo que rehaciendo cálculos en binario tenemos: 1110010110110000. Como solo se toman los 8 bits mas significativos, nos queda 11100101, que en decimal es 229.

Como el duty cycle del PWM es =  $(CCPR1L:CCP1CON)*Tosc*(TMR2 \text{ preescaler})$

y solo se utiliza CCP1RL sin el CCP1CON, el máximo duty cycle sería:

$$\text{Sustituyendo: } PWM = 229*4*(1/20000000)*1 = 0.0000458.$$

El periodo del PWM es de 0.00005.

Dividiendo para obtener el % nos da:  $0.0000458 / 0.00005 = 0.916 * 100 = 91.6 \%$ .

Si tenemos que el voltaje rectificado de los 120 volts es  $120 * 1.4142 = 169 \text{ V}$ .

Entonces el máximo voltaje multiplicado por el % del duty cycle será de 142 v

$142 \text{ v} / 1.4142 = 100.7 \text{ volts RMS}$  se deberían de obtener en la salida.

Considerando un eficiencia del sistema del 90% a lo mucho, tendríamos

$100.7 V_{\text{RMS}} * 0.9 = 90.68 V_{\text{RMS}}$ , que es lo mas cercano a nuestra hipótesis.

Se modificó la tabla y se volvió a medir el voltaje, esta vez era de 90 volts como lo habíamos predicho, no se alcanzaron los 110 volts RMS, esto es obvio; porque no podemos convertir de 110  $V_{\text{RMS}}$  a un voltaje de dc y luego reconvertir ese voltaje en dc a 110  $V_{\text{RMS}}$ , 60 hz. Si este fuera el caso nuestro controlador tendría una eficiencia del 100%, además, aunque bien podríamos haber subido un poco más los valores de la tabla senoidal, esto no se hizo por que si tuviéramos un duty cycle del 100 %, correríamos el riesgo de dañar el puente H.

Por otra parte, para complementar este proyecto, se implementó una rutina que convirtiese el valor capturado del ADC en código BCD que nos mostrara la frecuencia, desde 5 hasta 60 hertz. Como el valor del ADC va desde 20 hasta 240, lo primero que se hizo fue dividirla entre 4, esto nos devuelve el rango de 5 a 60, de ahí, el algoritmo indica que por cada 9 unidades, se sume 6, de esta manera, se convierte de binario a BCD, no se implementó una rutina que convirtiese de BCD a 7 segmentos, ya que hay que mencionar que la rutina no forma parte activa del programa, como lo es el PWM, sino es una rutina pasiva, que no modifica en nada los algoritmos activos, por lo tanto, implementar la conversión BCD a 7 segmentos requeriría ciclos de instrucciones que son necesarios para otras partes del programa y le restarían desempeño al sistema. Por ello se decodifica el formato BCD a 7 segmentos por hardware por medio de dos integrados 74ls47, que son convertidores de BCD a displays de 7 segmentos.



#### **5.4 Uso de MOSFETS en lugar de IGBTs.**

El proyecto original involucraba el uso de IGBTs para la etapa rectificadora hecha mediante un puente H, la ventaja de los IGBTs es que pueden controlar corrientes más altas que los MOSFETs. La mayor dificultad es obtenerlos en el mercado de nuestro país, aunque se obtuvieron algunos de ellos, en las pruebas preliminares quedaron destruidos y debido a que es muy difícil conseguirlos optamos por utilizar MOSFETS. Se eligió el 6N60, que es un MOSFET de mediana potencia y características similares al IGBT original, sólo que no maneja corrientes altas. Este económico semiconductor es muy utilizado en la etapa de conmutación de las fuentes conmutadas de monitores de video.

Su desempeño fue aceptable, si bien no pudieron manejar las corrientes de un motor de ¼ hp monofásico, manejan bien un motor de 1/32 hp monofásico sin ninguna dificultad. En una ampliación de la tesis presentada se pueden utilizar IGBTs que puedan manejar mayores potencias.

#### **5.5 Características adicionales.**

El driver IR21362 posee una entrada que sensa la corriente consumida por el motor que esta manejando, de esta manera, si la corriente fuera excesiva, el IR 21362 genera un señal de FAULT la cual se conecta al PIC 18F452 y éste cancela la generación de los PWMs. Además, el PIC genera la señal de ENABLE que habilita al IR21362, y posee el interruptor de encendido/apagado.