

CAPÍTULO 3

INTERFAZ AL PROTOTIPO

3.1 Comunicación Asíncrona al Computador

Se decidió usar el microcontrolador PIC 16F877 de "Microchip" por sus características de funcionamiento, como la utilidad de señales PWM para control del puente H y la comunicación serial al computador. El modo de operación para la comunicación en base al microcontrolador seleccionado puede ser el modo síncrono o asíncrono para la recepción y transmisión, utilizando el protocolo de comunicación RS 232.

Fue necesario usar un "driver" en este caso un MAX232 para manejar los niveles de voltaje de la comunicación. Por conveniencia se hizo uso de la tarjeta PICDEM2 que trae incorporados distintas funcionalidades como es la interfaz serial y una fuente independiente [20].

Las Principales Características de usar el modo asíncrono son:

- En modo asíncrono no se requiere una señal de reloj para control
- Puede utilizarse para recepción y transmisión al mismo tiempo
- Versatilidad y rapidez

3.1.1 Principio de Funcionamiento del USART en modo asíncrono de ocho bits

Transmisión

El USART (Universal Synchronous Asynchronous Receiver Transmitter) puede ser configurado para transmitir ocho o nueve bits "full duplex" [13]. Lo único que se debe hacer para que la información deseada sea transmitida por la terminal Tx del

microcontrolador, es primeramente configurar la tasa de transmisión y modo de la comunicación entre ambas bases de comunicación, configurar los puertos de transmisión y recepción como salida - entrada respectivamente. Por último la bandera de permiso a transmitir y mandar la información al registro TXREG. Así al detectar que el buffer de transmisión esta lleno el microcontrolador inmediatamente transmitirá el contenido [20].

Recepción

De la misma manera que en la transmisión, en la recepción el USART debe ser configurado para captar ocho o nueve bits de datos. Después de la detección del bit de inicio, los ocho o nueve bits de datos seriales son almacenados de la terminal RX al buffer de recepción y de ahí al registro RCREG, hasta que el bit de paro es reconocido. Este registro tienen la propiedad de almacenar los datos de forma FILO (First In Last Out), el primer dato que entra es el último en salir [20].

Por otra parte el medio de comunicación se mantiene en un nivel lógico alto siempre y cuando no haya transmisión o recepción en progreso. Le sirve al canal de recepción como sincronización de inicio, el primer pulso bajo y consecutivamente los ocho bits de información. Al terminarse la “trama” de información éste activa un bit de paro en alto que regresa al estado inicial activando una bandera. Para la transmisión se debe insertar el dato en el TXREG para ser mandado por el medio y su respectiva bandera indicará que ya fue liberado el buffer de Tx e iniciar otra transmisión. La transmisión y recepción de la información se puede llevar a cabo simultáneamente.

Entre otras cosas el canal de transmisión y recepción están determinados por las terminales del microcontrolador TX y RX del puerto C (RC6-TX y RC7-RX para el microcontrolador 16F877). Y dentro de ellos en conjunto con otros registros de propósito específico, como banderas de recepción de datos y transmisión se puede controlar a la par su funcionamiento a través de ciertas instrucciones y lineamientos de programación.

3.1.2 Diseño de control

Se elaboró el programa en ensamblador para el microcontrolador 16F877 de Microchip con la rutina perteneciente a la comunicación asíncrona entre el PIC y el computador.

1. Inicialización del puerto C con las terminales correspondientes
2. Configuración de la tasa de transferencia
3. Configuración de la transmisión asíncrona
4. Configuración de la recepción asíncrona
5. Rutina de recepción de espera del carácter de validación para inicio
6. Validación del carácter de inicio
7. Rutina de transmisión de los datos cíclica por llamadas

Ya que sabemos el gran contenido de rutinas de control del programa hecho en ensamblador para el PIC 16F877, se da una descripción general de la inicialización de los puertos y los registros de propósito general, para el funcionamiento de cada bloque.

3.2 Inicialización de Puertos y Registros de Función General

Como se detalla en el programa ensamblador del Apéndice B se tiene como primera inicialización la declaración de los registros usados como contadores, comparadores y la habilitación de los puertos utilizados. También se genera la salida del vector de interrupción al finalizar la cuenta del codificador correspondiente. Todo esto para complementar las rutinas, la inicialización de los programas y su funcionamiento.

La declaración fue la siguiente:

1. Declaración de los registros o variables a usar
2. Declaración del vector de interrupción con su respectiva salida
3. Inicialización en el puerto 0
4. Limpieza de los puertos C y D
5. Limpieza del Timer 1 de 16 bits
6. Limpieza de los Contadores
7. Inicialización del Puerto D , C y B

En pocas palabras los registros de propósito general fueron utilizados para generar variables que permitieron delimitar los puntos de rotación de los codificadores. La inicialización de los puertos para la manipulación de la unidad de control de baja potencia, activación y desactivación de los motores. La declaración del vector de interrupción asistió el redireccionamiento de las rutinas de los motores X-Y.

3.3 Generación de la Señal PWM

La señal de PWM introducida en los Puentes H consiste en un ciclo de trabajo en donde se codifica la información de la amplitud y la dirección de la señal de salida. La corriente entregada por el Puente H, es proporcional al ancho de pulso de la señal PWM.

La aplicación del PWM en este proyecto para el funcionamiento de los motores, se realizó bajo pruebas para conocer el mejor desempeño de estos en conjunto con los motores. Es decir, que se puede presentar problemas en cuanto se hagan funcionar si no están caracterizados para los motores usados. El problema observado fue que si la señal de PWM proporcionado al puente H era de un ciclo de trabajo demasiado grande el motor no se movía, y si el ciclo de trabajo fuese muy pequeño el motor revolucionaria a su máximo. Se debe tener en cuenta que el funcionamiento descrito anteriormente se basa en que el acondicionamiento de la señal en el opto acomplador trabaja con lógica inversa, es decir cuando se activa la señal de salida se apaga y viceversa.

Por eso se decidió utilizar una frecuencia de 300 Hz un ciclo de trabajo del 20%, una frecuencia baja en donde el motor del eje Y trabaja a 2.5 vueltas por minuto y poder compatibilizar la frecuencia de operación del sensor, con la transferencia de la información y el control por parte del microcontrolador. Está comprobado que los valores decididos le dan al sistema un buen funcionamiento en cuanto su eficiencia en movimiento sin llegar a una máxima respuesta de operación en conjunto. Esto se puede lograr llevando al límite los diversos componentes, mediante pruebas de funcionamiento en el desplazamiento y aplicación de componentes de alta frecuencia.

En las pruebas realizadas se probó con varias frecuencias de operación y ciclos de trabajo, desde 10Hz hasta 100 kHz variando el ciclo de trabajo del 10% al 90%, por lo que el mejor funcionamiento percibido fue el mencionado anteriormente el de 300 Hz al 20%. Otro semblante a considerar es que si se utilizan frecuencias muy altas en el caso de que se pudiera utilizar, los componentes trabajarían a su máxima respuesta de operación. Sin embargo, las problemáticas resultantes serían conflictos entre el funcionamiento de los componentes. Por eso se decidió trabajar en una frecuencia y ciclo de trabajo bajo para hacer a un lado todas las problemáticas referentes a las frecuencias de operación de los componentes y los conflictos resultantes.

La señalización del PWM para los puentes H se llevó a cabo mediante una rutina en el microcontrolador. En el programa en ensamblador encontramos la parte de la generación del PWM que consiste en la rutina siguiente:

1. Inicialización del CCP1CON y Timer 2
2. Especificar el Prescalador y la frecuencia del cristal
3. Determinar y especificar el ciclo de trabajo para la señal PWM
4. Encender e incrementar timer2
5. Realizar rutinas en ciclo para que funcione repetitivamente el PWM

Dentro de la programación encontramos como parte fundamental el cálculo del PWM en base a los siguientes parámetros.

La declaración de la inicialización del PR2 con la fórmula que a continuación se muestra:

$$\text{PWM periodo} = [(\text{PR2}) + 1] \cdot 4 \cdot \text{TOSC} \cdot (\text{TMR2 valor de preescalador}) [6]$$

Haciendo el cálculo PR2 es igual a,

$$(0.003 / (4 \cdot 2.5 \cdot 10^{-7} \cdot 16)) - 1 = 207.3125$$

* Cálculos realizados para un cristal 4MHz. y para una frecuencia resultante de 300Hz.

Por lo que se declaró los 8 bits del PR2 como:

Bit 7 – Sin implementar

Bit del 6-3 = Posescalador de 1:9 = “1001”

Bit 2 = Timer 2 aprendido = “1”

Bit 1 = Preescalador de 16 = “1”

La declaración de la inicialización para el CCPR1L fue calculada por medio de la siguiente fórmula.

$$\text{PWM ciclo de trabajo} = (\text{DCxB9:DCxB0 bits}) \cdot \text{Tosc} \cdot (\text{TMR2 valor preescalador}) [6]$$

Por lo tanto CCPR1L se definió como:

$$\text{“0010010110”} = 0.0006 / (2.5 \cdot 10^{-7} \cdot 16) = 150$$

* Cálculos para realizar un ciclo de trabajo del 20 %

Y como último para inicializar el funcionamiento del PWM el CCP1CON se declaró como:

$$\text{“00101100”} = \text{CCP1CON}$$

Donde los 8 bits del CCP1CON son:

Bit 5-4 = “10” modo PWM

Bit 3-0 = “1100” modo PWM