

Capítulo 3: Diseño del mecanismo y desarrollo funciones

3.1 Diseño del mecanismo

Como se explicó en las primeras páginas de este escrito, la finalidad de este trabajo es la documentación de las diferencias entre los microcontroladores PIC y los ARM, por lo tanto, el mecanismo diseñado es solo para propósitos de demostración.

El mecanismo propuesto se trata de un brazo robótico con tres grados de libertad. La razón para elegir este mecanismo es debido a su simplicidad, basta con suministrar el PWM correspondiente a cada servomotor para que se ubique en la posición deseada. Dicho brazo robótico cuenta con un display LCD para observar la posición de los servomotores, un teclado matricial para proporcionar la posición deseada a cada servo, un joystick para poder ubicar los servomotores si es que no se sabe la posición deseada y un switch para cambiar entre los modos de operación.

Una particularidad de este brazo es que se acomode en la posición deseada de manera suave, por lo que podría utilizarse en conjunto con una cámara, ya sea para fotografía o video. A continuación, se presenta el prototipo realizado.

3.2 Utilizando Keil

El entorno de desarrollo *IDE* de este proyecto fue Keil Uvision 5. Este programa es suministrado por parte de Keil una empresa dedicada al desarrollo de software embebido perteneciente a ARM [18].

El uso de Keil Uvision es sumamente sencillo, pues te proporciona los archivos para programar de acuerdo a tus necesidades. Para poder trabajar con este IDE es necesario crear un *Nuevo Proyecto*, con esto aparecerá una ventana en la que especificamos el integrado a utilizar. Sin embargo, al instalar por primera vez Keil viene sin archivos de configuración para los diferentes integrados, por lo que hay que actualizarlo mediante la instalación de paquetes. En las siguientes imágenes se muestra el proceso de crear un proyecto e instalación de paquetes.

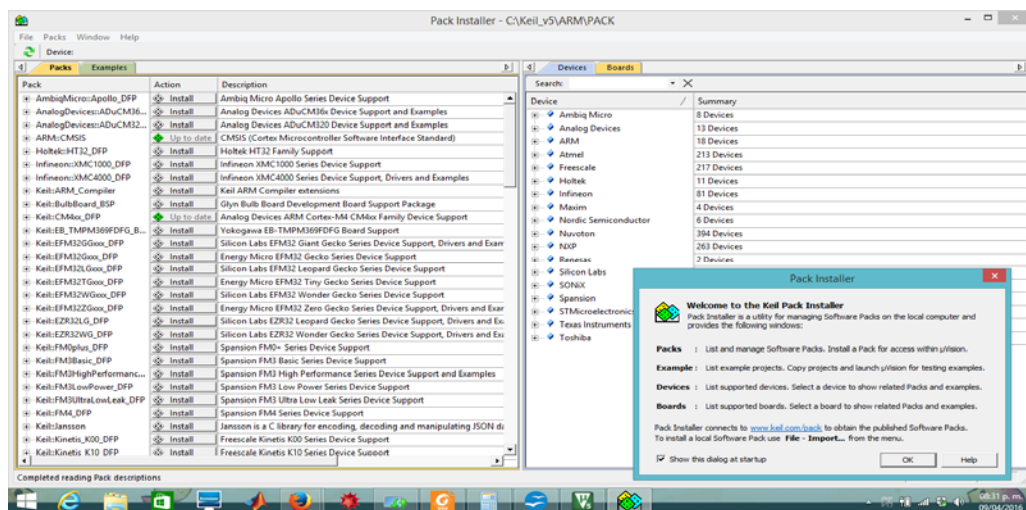


Figura 3.1 Instalación de paquetes en KEIL Uvision

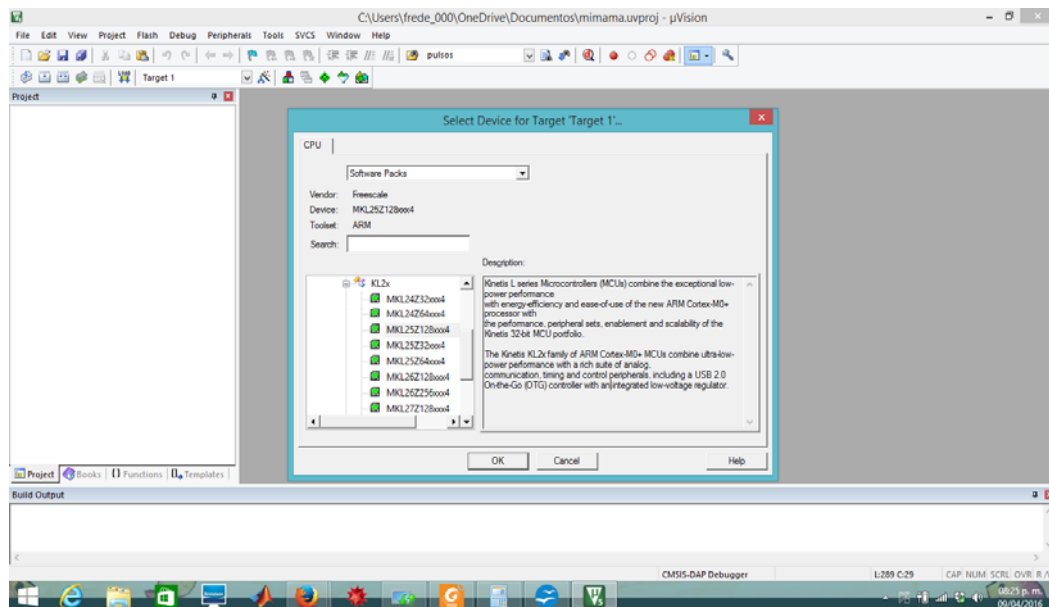


Figura 3.2 Selección del integrado

Una vez elegido el integrado a utilizar aparecerá otra ventana, esta ventana llamada *Manage Run-Time Environment* es en donde se eligen los archivos necesarios para la realización del proyecto, por lo que elegiremos Core y Startup. En estos archivos Keil tiene como opción default que el microcontrolador funcione a 41.94 MHz, pero se puede elegir otra opción, Keil trae las funciones necesarias para configurar el microcontrolador en PEE o BLPE con tan solo cambiar un número.

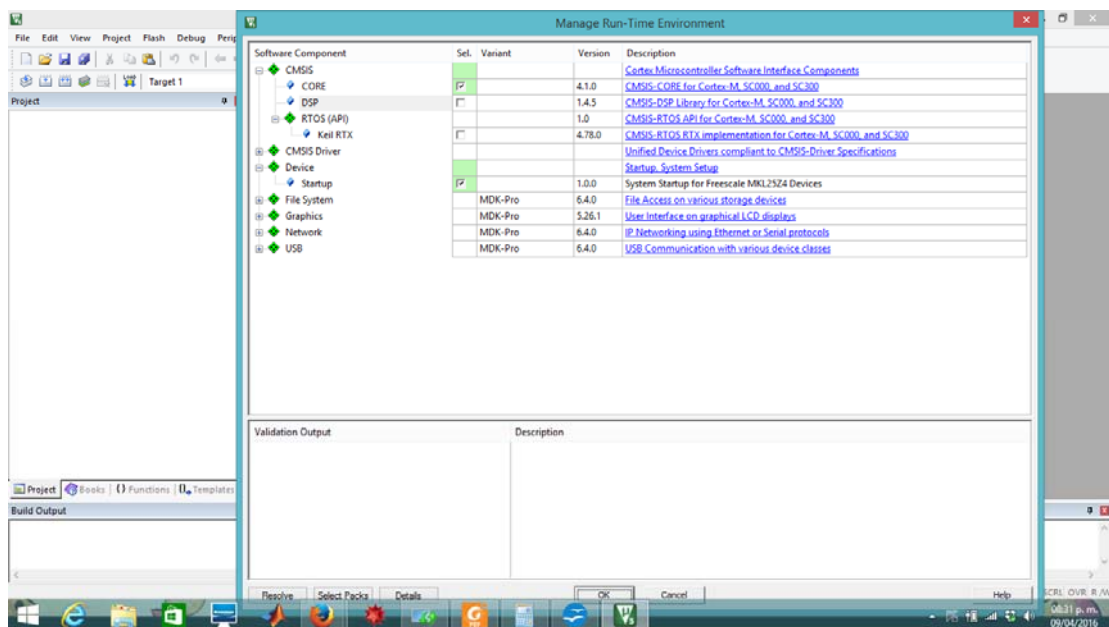


Figura 3.3 Selección de archivos de inicialización.

Como se pudo observar en las imágenes anteriores, Keil Uvision cuenta con una base de datos actualizada, pues se encuentran los integrados por fabricante e incluso se pueden encontrar ejemplos. Una vez realizado lo anterior podemos escribir nuestros códigos como es costumbre, se compila y se graba en la tarjeta.

3.3 Configuración de Módulos

Para la realización de este trabajo se utilizaron los siguientes módulos de la tarjeta.

- Módulo TPM *Timer PWM Module*
- GPIO *General Purpose Input Output*
- I2C *Interconnected Integrated Circuit*
- ADC *Analog Digital Converter*

Debido a la cantidad de funciones realizadas se recurrió a concentrarlas en diferentes archivos, los archivos creados fueron:

- timers.c y timers.h
- I2C.c e I2C.h
- LCD.c y LCD.h
- puertos.c y puertos.h
- varios.c y varios.h

Hay que aclarar que antes de empezar a configurar los módulos con los que cuenta el integrado hay que activar el clock gating, esto es debido a que al reiniciar la tarjeta todos los módulos son deshabilitados. Escribir a los registros sin habilitar el clock gating generara una excepción *Hard Fault*, por lo que la tarjeta quedara atascada en dicha interrupción. Para habilitar el clock gating de este o cualquier otro modulo se hace a través del *SIM System*

Integration Module. Este módulo lo implementa Freescale. Asimismo, si se requiere modificar algún parámetro general de un módulo hay que revisar si está presente dicha opción en SIM.

3.3.1 Modulo TPM

Este módulo es parecido al CCP de los PIC. La tarjeta cuenta con 3 módulos TPM, TPM0, TPM1 y TPM2. Cada módulo tiene 6 canales, por lo que podrían tenerse teóricamente 18 salidas de PWM. Para utilizar este módulo como timer basta con configurar la señal de reloj del timer, seleccionar pre escalador y poner el periodo de interrupción en ticks en el Registro MOD. En el caso de generar señales PWM, OUTPUT COMPARE hay que configurar los pines especificados mediante el registro PCR del pin, poner el modo del canal a utilizar, y poner el valor del canal.

Para evitar tener que recordar valores para cada timer a la hora de utilizar las funciones se recurrió a utilizar una enumeración. Además, se crearon funciones para las siguientes tareas:

- Clock gating de los timers
- Habilitación y deshabilitación del modulo
- Selección de la fuente de reloj
- Limpiar interrupción del canal o del timer
- Configuración de modos en canales

- Establecer el periodo del timer
- Establecer los valores de los canales

3.3.2 GPIO

Los puertos que soportan interrupciones son el PORTA y PORTD. Asimismo, cada uno de los puertos con interrupción cuenta con su propio vector de interrupción, por lo que en la interrupción se tendrá que obtener el pin donde ocurrió la misma.

Debido a que los pines tienen diferentes usos estos se encuentran multiplexados con los diferentes módulos, por lo que se creó una función que ayuda a configurarlos en cualquiera de sus funciones. Con la creación de las funciones para este módulo se buscó cumplir con las siguientes tareas:

- Clock Gating módulo GPIO
- Leer y escribir un pin de un puerto dado
- Configurar el modo de interrupción de un pin
- Detectar que pin causó interrupción en un puerto
- Configuración del pin según modulo a utilizar

3.3.3 I2C

Las funciones para el uso del I2C se realizaron mediante polling, es decir, no se utilizó su interrupción. Este componente fue un poco difícil de configurar debido a que presenta

muchas más configuraciones, tales como el modo de 10 bits, el modo multi-maestro y la configuración de las distintas frecuencias de operación por medio de tablas. Las funciones de este módulo tratan de cubrir lo siguiente:

- Obtención de MUL e ICR para determinar la frecuencia de uso actual
- Inicialización de los módulos I2C0 e I2C1 a 48.51 kHz, incluyendo clock gating
- Lectura y escritura sencilla
- Lectura y escritura en modo Burst

El I2C0 está conectado internamente con el acelerómetro, utilizando PTE24 y PTE25. Además, se creó una enumeración para detectar el error en la utilización de estos módulos.

3.3.4 ADC

Este módulo consta de 24 canales simples y 4 pares de canales diferenciales. Las funciones desarrolladas para este módulo se basaron en canales simples y el uso de las interrupciones. Para poder utilizar este módulo hay que recurrir a una rutina de auto calibración, la cual utiliza valores almacenados en sus registros para calcular un valor de calibración y aplicarlo al registro PG del ADC0. La función de calibración se especificó como static y se introdujo dentro de la función habilitadora, por lo que no se puede llamar de manera explícita dentro del main. Las funciones realizadas para este módulo tienen como objetivo crear las siguientes tareas:

- Clock Gating ADC0
- Configuración de ADC0 con canales simples
- Habilitación del ADC0, esta función trae la calibración y la habilitación de la interrupción en NVIC

3.3.5 UART0

Este módulo se configuró en modo transmisor para visualizar datos a través de un terminal serial como Teraterm con el fin de depurar el código generado. Las funciones que se crearon para el uso de este módulo fueron para poder observar los resultados de las operaciones realizadas en la tarjeta, por lo tanto, cubren las siguientes tareas:

- Clock Gating del UART0
- Habilitación del UART0, utilizando modo simple sin bit de paridad
- Deshabilitar el UART0
- Configuración de frecuencia de operación
- Envío de cadenas

Para visualizar la información se utilizaron cadenas de texto, para convertir un número a texto se ocupó la función `sprintf` que viene en la librería `stdio`.

3.4 Dispositivos externos

3.4.1 Display LCD

El display LCD es un elemento común hoy en día, pues el precio es barato y la programación es sencilla. En este trabajo el display utilizado tenía el integrado 1602A compatible con el controlador HD44780 de Hitachi por parte de SHENZHEN EONE ELECTRONICS. El display LCD se configuro en 16x2 sin blinking e incremento en el cursor en cada escritura. A continuación, se presenta el display LCD utilizado y una tabla que muestra las conexiones realizadas entre el display LCD y la tarjeta.



Figura 3.4 Display LCD

Pines LCD	RS	RW	E	D0	D1	D2	D3	D4	D5	D6	D7
Pines Tarjeta	PTB0	PTB1	PTB2	PTC2	PTC3	PTC4	PTC5	PTC6	PTC7	PTC8	PTC9

Tabla 3.1 Conexión del display LCD

Para llevar un control de la posición del cursor se creó la variable *ac* y en cada instrucción de escritura se aumentaba en 1, en las funciones de limpieza del display esta variable se pone a 0 y en la función que limpia el ultimo digito se excrementaba en 1 la variable *ac*.

3.4.2 Teclado Matricial

Para la introducción de datos entre el microcontrolador es muy usual encontrar un teclado matricial. El teclado matricial no es más que un arreglo de pulsadores tal y como se muestra en la siguiente figura.

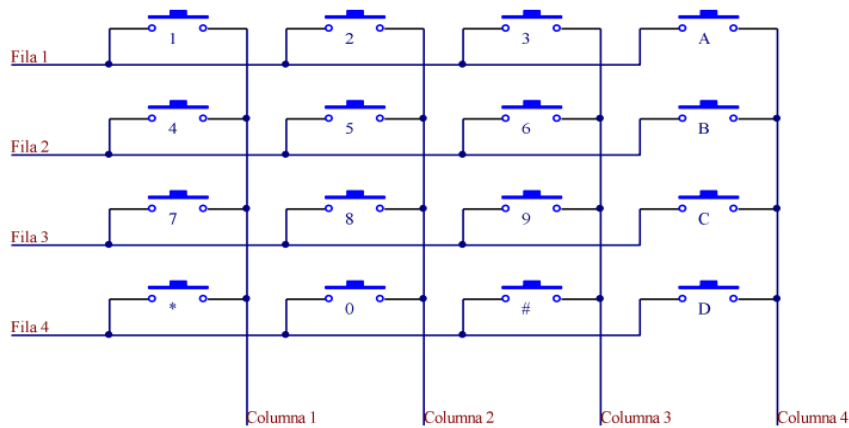


Figura 3.5 Arreglo de pulsador 4x4

Para poder detectar la tecla oprimida se configuraron las columnas para estar en alto mediante conexiones pull up, asimismo se configuro las interrupciones en flanco de bajada a las columnas, por otro lado, se configuraron los renglones para tener 0.

En la rutina de interrupción se guarda la columna en donde ocurrió la interrupción, después se procede a realizar un barrido en cada uno de los renglones, manteniendo 1 bit de los renglones en alto mientras los otros son 0, es ahí que si se detecta un cambio de 0 a 1 de una columna cuando se sabe la posición de la tecla que se encuentra apretada, también se le agrego un pequeño delay para dejar pasar el tiempo debouncing del switch. A continuación, se muestra el teclado ocupado en este trabajo.



Figura 3.6 Teclado matricial utilizado

3.4.3 Joystick

El joystick se conforma de dos potenciómetros, con lo cual podríamos saber el que posición se encuentra, sin embargo, este no se comporta de manera 'lineal' en los extremos. Por lo que se recurrió a dividirlo en sectores para asignar los grados que avanza o retrocede el servo, tal y como se muestra en la Figura 3.7.

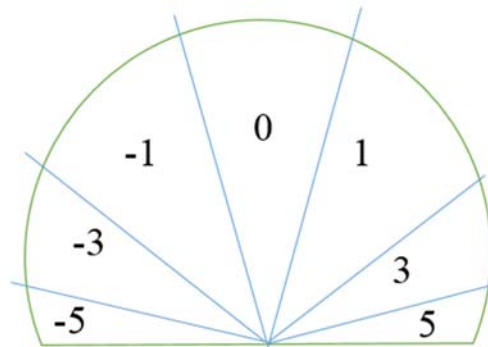


Figura 3.7 Distribución del potenciómetro

Además de esto se observó que para la variable X el potenciómetro aumentaba en la dirección X+, mientras que para la variable Y el potenciómetro aumentaba en la dirección Y-, por lo que se procedió a realizar un pequeño arreglo dentro del código, invirtiendo los rangos para asignar los grados que debe avanzar el brazo. A continuación, se muestra el joystick utilizado.



Figura 3.8 Joystick utilizado

3.4.4 Encoder en cuadratura

El uso de este dispositivo es para el movimiento del tercer eje de libertad de la versión con joystick. Este tipo de encoder permite saber la distancia recorrida y la dirección de giro. Para el uso de este dispositivo se utilizaron los pines PTA1 y PTA2 y se configuraron sus interrupciones en ambos flancos de la señal. El encoder en cuadratura funciona tal como se muestra en la siguiente figura.

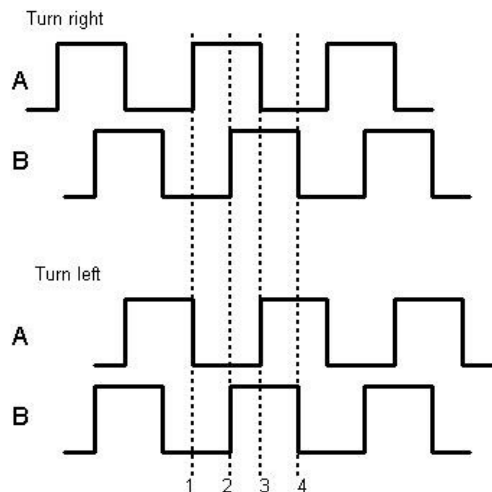


Figura 3.9 Funcionamiento del encoder

En este caso las señales se llaman CLK y DT respectivamente. Como se puede observar la secuencia de valores generados por el encoder es código grey. Para tener un control exacto de su movimiento se utilizó la siguiente matriz, donde a partir del valor previo y al valor actual en los pines se puede determinar en qué dirección se ha girado, tal como se muestra en la siguiente figura.

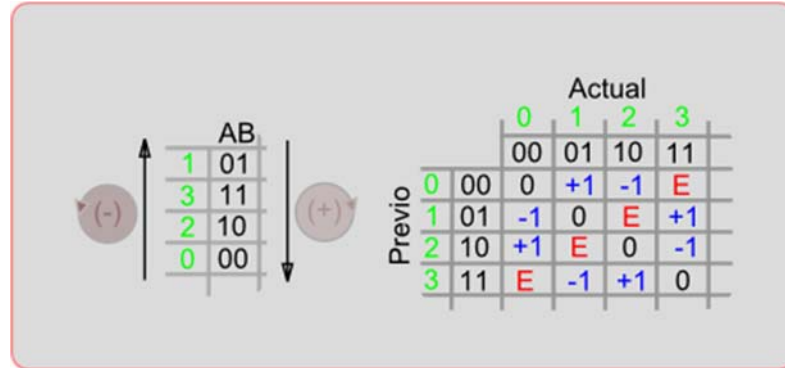


Figura 3.10 Matriz de control encoder

3.4.5 Servomotores

El servomotor es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición. Está conformado por un motor, una caja reductora y un circuito de control. Los servomotores se controlan mediante señales PWM, usualmente el periodo de esta señal es de 20 milisegundos. A continuación, se presenta el servomotor utilizado.



Figura 3.11 Servomotor MG995

Este servomotor tiene muy buenas características en cuanto a torque, pues cuenta con un torque de 8.5 kg-cm, además su hoja de datos dice que permite un rango de movimiento de 180 grados.