

**SISTEMA DE MEDICION ANGULAR MICROCONTROLADO, DE ALTA RESOLUCIÓN,
UTILIZANDO UN *SYNCHRO***

T-G: 0417

**LUIS GUILLERMO AYALA RAMÍREZ
RAFAEL HUMBERTO CÓRDOBA QUINTERO
ALVARO MARTÍNEZ ALVERNIA**

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.**

2004

**SISTEMA DE MEDICION ANGULAR MICROCONTROLADO, DE ALTA RESOLUCIÓN,
UTILIZANDO UN *SYNCHRO***

**LUIS GUILLERMO AYALA RAMÍREZ
RAFAEL HUMBERTO CÓRDOBA QUINTERO
ALVARO MARTÍNEZ ALVERNIA**

**Trabajo de grado presentado como
requisito parcial para optar al título de
Ingeniero Electrónico**

**Director:
CAMILO OTÁLORA SÁNCHEZ
Ingeniero Electrónico**

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE ELECTRÓNICA
BOGOTÁ D.C.**

2004

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA ELECTRÓNICA

RECTOR MAGNIFICO: R.P. GERARDO REMOLINA S.J.

DECANO ACADÉMICO: Ing. ROBERTO ENRIQUE MONTOYA VILLA

DECANO DEL MEDIO UNIVERSITARIO: R.P. ANTONIO JOSÉ SARMIENTO NOVA S.J.

DIRECTOR DE CARRERA: Ing. JUAN CARLOS GIRALDO CARVAJAL

DIRECTOR DEL PROYECTO:

ARTICULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

"La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia".

TABLA DE CONTENIDO

1.	INTRODUCCION	1
2.	MARCO TEÓRICO	4
2.1.	SENSORES Y TRANSDUCTORES DE POSICIÓN ANGULAR	4
2.1.1.	Encoder incremental	4
2.1.2.	Synchro	6
2.2.	FORMAS DE CONVERSIÓN DE UN SYNCHRO	8
2.2.1.	Tracking o seguimiento	8
2.2.2.	Aproximaciones Sucesivas	10
2.2.3.	Conversión synchro-digital mediante submuestreo	11
2.2.4.	Conversión synchro-digital mediante sobremuestreo	12
2.2.4.1.	Filtro FIR pasabanda y decimación	13
2.2.4.2.	Función arcotangente	14
2.2.4.3.	Malla cerrada de interpolación y ajuste de velocidad	15
2.3.	PROTOCOLO MODBUS	16
2.3.1.	Modo ASCII	16
2.4.	MICROCONTROLADOR PIC18F4521	17
2.5.	MOTORES ELÉCTRICOS	20
2.6.	PUENTE H	21
3.	ESPECIFICACIONES	23
3.1.	DESCRIPCIÓN GENERAL	23
3.2.	DIAGRAMA EN BLOQUES	25
3.2.1.	Etapa de procesamiento analógico	25
3.2.2.	Etapa de procesamiento digital	25
3.2.3.	Etapa de control	27

3.2.4.	Etapa de interfaz gráfica	27
3.2.5.	Etapa de aplicación	27
4.	DESARROLLOS	28
4.1.	DESCRIPCION GENERAL	29
4.2.	ETAPA DE PROCESAMIENTO ANALÓGICO	29
4.2.1.	Filtro	29
4.2.2.	Circuito transformador de Scott	30
4.2.3.	Manejador de RS-232	32
4.3.	ETAPA DE PROCESAMIENTO DIGITAL	33
4.3.1.	Cuadrada 400Hz	34
4.3.1.1.	Interrupción Timer 1	34
4.3.2.	Control	35
4.3.3.	Algoritmo de sobremuestreo	37
4.3.3.1.	Interrupción Timer 0	37
4.3.3.2.	Filtro Pasabajos	37
4.3.3.2.1.	Implementación del filtro FIR pasabanda en MATLAB	38
4.3.3.2.2.	Implementación del filtro FIR pasabanda en SIMULINK	39
4.3.3.2.3.	Implementación del filtro FIR pasabajos en MATLAB	41
4.3.3.2.4.	Implementación del filtro FIR pasabanda en SIMULINK	42
4.3.3.2.5.	Implementación del filtro IIR pasabajos en MATLAB	43
4.3.3.2.6.	Implementación del filtro IIR pasabajos en SIMULINK	44
4.3.3.2.7.	Comparación entre los filtros FIR e IIR pasabajos	45
4.3.3.3.	Función arcotangente	46
4.3.3.4.	Malla cerrada de interpolación	47
4.3.3.4.1.	Filtro IIR	48
4.3.3.4.2.	Controlador Proporcional-Integral de la malla de interpolación	50
4.3.3.4.3.	Integrador	51
4.3.4.	Módulo MODBUS esclavo	51
4.3.4.1.	Interrupción por recepción en la transmisión serial	51
4.3.5.	Generación de cuadratura	52
4.3.5.1.	Interrupción (Timer3)	52
4.3.6.	DAC y PWM	53
4.4.	ETAPA DE CONTROL	55

4.5.	ETAPA DE INTERFAZ GRÁFICA	58
4.6.	ETAPA DE APLICACIÓN	60
4.7.	OTROS DESARROLLOS DEL PROYECTO	61
4.7.1.	Circuitos Impresos	61
4.7.1.1.	Tarjeta Analógica	61
4.7.1.2.	Tarjeta digital	62
5.	ANALISIS DE RESULTADOS	63
5.1.	PRUEBA DE CONSTANTE DE TRANSFORMACION DEL SYNCHRO	63
5.2.	PRUEBAS DEL ALGORITMO DE SOBREMUESTREO	64
5.2.1.	Respuesta del filtro IIR pasabajos	64
5.2.2.	Respuesta del filtro IIR de la malla de interpolación	65
5.2.3.	Respuesta de la función arcotangente	65
5.2.4.	Respuesta de la malla cerrada de interpolación del ángulo	66
5.2.5.	Respuesta final del algoritmo de sobremuestreo	70
5.3.	PRUEBA DE COMPARACIÓN CON ENCODER DE CUADRATURA INCREMENTAL	77
5.4.	PRUEBA DE ESTABIILIDAD CON LA TEMPERATURA	79
5.5.	PRUEBA DE REPETICIÓN DE LA MEDIDA ANGULAR	80
5.6.	PRUEBA GEOMÉTRICA DE PRECISIÓN	82
5.7.	PRUEBA DE SALIDA DE CUADRATURA DEL SISTEMA	83
5.8.	PRUEBA DE LA APLICACIÓN DE CONTROL DE POSICIÓN	84
5.9.	CAPACIDAD DE PROCESAMIENTO DEL SISTEMA	85
5.10.	ANALISIS DE COSTOS	86
6.	CONCLUSIONES	88
7.	BIBLIOGRAFÍA	91
	ANEXOS	92

TABLA DE ANEXOS

ANEXO A. FOTOGRAFÍAS DEL MONTAJE FINAL Y APLICACIÓN DE CONTROL DE POSICIÓN.

ANEXO B: HOJA DE ESPECIFICACIONES DEL DAC7513

ANEXO C: DIAGRAMA CIRCUITAL DE LA TARJETA ANALÓGICA

ANEXO D: DIAGRAMA CIRCUITAL TARJETA DIGITAL

ANEXO E: CÓDIGO DEL PROGRAMA IMPLEMENTADO EN EL MICROCONTRLADOR PIC18F452

ANEXO F: CÓDIGO DE LA INTERFAZ GRÁFICA

TABLA DE FIGURAS

Figura 1.1. Señales en cuadratura incremental	5
Figura 2.2.a) Presentación comercial de los encoder. b) Grafica del funcionamiento.	5
Figura 2.3.a) Presentación Comercial del Synchro. b) Señales del Synchro en función del ángulo	6
Figura 2.4. Malla de seguimiento	9
Figura 2.5. Diagrama básico de la conversión mediante submuestreo	11
Figura 2.6. Diagrama en bloques de la conversión mediante sobremuestreo	13
Figura 2.7. Respuesta en magnitud del filtro FIR de Decimación	14
Figura 2.8. Malla de interpolación y ajuste de velocidad	15
Figura 2.9. Trama de Modbus ASCII	17
Figura 2.10. Distribución de terminales del PIC18F452	19
Figura 2.11. Configuración de un motor DC	20
Figura 2.12. Configuración de un Puente H	21
Figura 3.1. Esquema general de entradas y salidas del sistema	23
Figura 3.2. Diagrama en bloques	26
Figura 4.1. Esquema de desarrollo del sistema de medición angular	28
Figura 4.2. Diagrama en bloques definitivo del algoritmo implementado	29
Figura 4.3. Filtro Pasabajos para la señal cuadrada de 400Hz	30
Figura 4.4. Circuito transformador de Scott	31
Figura 4.5. Señales entregadas por el synchro.	31
Figura 4.6. Interconexión para el manejador de RS-232	33
Figura 4.7. Diagrama de flujo del programa principal	34
Figura 4.8. Diagrama de flujo de la interrupción del Timer 1	35
Figura 4.9. Diagrama de flujo de la etapa de control	36
Figura 4.10. Diagrama de la interrupción del Timer 0	37
Figura 4.11. Esquema básico del filtro FIR.	38
Figura 4.12. Simulación de la respuesta en frecuencia para el filtro FIR	39

Figura 4.13. Diagrama en bloques del filtro FIR pasabanda en SIMULINK	40
Figura 4.14. Respuesta a entrada paso para el filtro FIR, pasabanda, mediante simulación en SIMULINK.	40
Figura 4.15. Respuesta real a entrada paso, del microcontrolador, para el filtro FIR	40
Figura 4.16. Respuesta en frecuencia para el filtro FIR pasabajos	42
Figura 4.17. Diagrama en bloques del filtro FIR pasabajos en SIMULINK	42
Figura 4.18. Respuesta a entrada paso para el filtro FIR, pasabanda, mediante simulación en SIMULINK	43
Figura 4.19. Esquema básico de un filtro IIR	43
Figura 4.20. Respuesta en frecuencia para el filtro IIR pasabajos	44
Figura 4.21. Diagrama en bloques del filtro IIR pasabajos en SIMULINK	44
Figura 4.22. Respuesta a entrada paso para el filtro IIR, pasabajos, mediante simulación en SIMULINK	45
Figura 4.23. Esquema de comparación para los filtros FIR e IIR pasabajos	45
Figura 4.24. Comparación de respuestas a entrada paso, para filtros IIR y FIR pasabajos	46
Figura 4.25. Comportamiento con respecto al ruido, ante una entrada paso, para filtros IIR y FIR pasabajos	46
Figura 4.26. Esquema de la malla cerrada de interpolación y ajuste de velocidad	47
Figura 4.27. Malla cerrada de interpolación implementada en simulink	48
Figura 4.28. Respuesta en frecuencia para el filtro IIR pasabajos de la malla cerrada de interpolación	49
Figura 4.29. Diagrama en bloques del filtro IIR, en SIMULINK, de la malla cerrada de interpolación	49
Figura 4.30. Respuesta a entrada paso para el filtro IIR, de la malla de interpolación, mediante simulación en SIMULINK	50
Figura 4.31. Diagrama en bloques de la malla de interpolación con el controlador P+I	50
Figura 4.32. Diagrama en bloques del integrador de la malla de interpolación	51
Figura 4.33. Diagrama de flujo de la interrupción por recepción en la transmisión serial	52
Figura 4.34. Diagrama de flujo por interrupción del Timer 3	53
Figura 4.35. Diagrama de flujo de conversión de medida digital-analógica	54
Figura 4.36. Acople mecánico 1:1, entre los rotores del synchro y motor DC, para el grado de libertad del eje a) vertical y b) horizontal	55

Figura 4.37. Modelo para um motor DC	56
Figura 4.38. Diagrama del controlador para el motor DC	57
Figura 4.39. Respuesta a entrada paso para el controlador del motor DC	57
Figura 4.40. Esquema de simulación de respuesta final del controlador	58
Figura 4.41. Respuesta a entrada paso para la malla definitiva de control	58
Figura 4.42. Pantalla de Inicio de interfaz gráfica	59
Figura 4.43. Pantalla de medición de la interfaz gráfica.	59
Figura 4.44. Pantalla de control de la aplicación mecánica	60
Figura 4.45. Montaje mecánico de dos grados de libertad, para aplicación de control de posición	61
Figura 5.1. Respuesta lineal de la respuesta del transformador	64
Figura 5.2. Respuesta real a entrada paso para el filtro IIR del microcontrolador	64
Figura 5.3. Respuesta Entrada paso para el filtro IIR de la malla de interpolación	65
Figura 5.4. Error de la función arcotangente implementada en el microcontrolador	66
Figura 5.5. Esquema de simulación, de la malla cerrada de interpolación, ante entrada paso	67
Figura 5.6. Esquema de simulación para respuesta a entrada rampa, de la malla de interpolación	70
Figura 5.7. Respuesta de la malla de interpolación a entrada rampa	70
Figura 5.8. Esquema de simulación, del algoritmo de sobremuestreo, para entrada paso	71
Figura 5.9. Esquema de simulación para respuesta a entrada rampa, del algoritmo definitivo	74
Figura 5.10. Encoder Incremental de la casa RENCO CORP, utilizado para prueba con el sistema de medición angular	77
Figura 5.11. Detalle del montaje mecánico entre el encoder y el synchro para prueba de comparación	78
Figura 5.12. Error del encoder incremental vs. algoritmo de sobremuestreo	79
Figura 5.13. Variación del error del ángulo con respecto a la temperatura	80
Figura 5.14. Pruebas de repetición de la medida angula, para diferentes valores	81
Figura 5.15. Detalle de la parte externa del synchro	82
Figura 5.16. Esquema para medida geométrica de mínima variación	82
Figura 5.17. Error en la medida del ángulo ante prueba geométrica	83
Figura 5.18. Error de la salida de cuadratura incremental	83

Figura 5.19. Sistema electrónico utilizado para medir las señales de cuadratura	84
Figura 5.20. Respuesta de la aplicación de control de posición ante una entrada paso	85

TABLA DE TABLAS

Tabla 2.1. Resolución de algunos transductores de ángulo	8
Tabla 2.2. Características de microcontrolador PIC18F452	18
Tabla 3.1. Principales características del sistema de medición	24
Tabla 5.1. Relación de voltajes entre la referencia y las salidas del synchro	63
Tabla 5.2. Simulación de resolución angular y tiempo, para la malla de interpolación, de establecimiento ante variaciones de K_i y K_p	69
Tabla 5.3. Respuesta simulada y real, del algoritmo, para diferentes entrada paso	73
Tabla 5.4. Respuestas, simulada y real, del algoritmo, para diferentes entrada rampa	76
Tabla 5.5. Medida de tiempos de procesamiento del algoritmo	85
Tabla 5.6. Costos para el desarrollo total del Sistema de Medición Angular Microcontrolado de Alta Resolución Utilizando un Synchro	87

*A la Virgen de Torcoroma bendita,
al Cristo Petrolero y al Señor de los Milagros...*

“ama et quod vis fac”

San Agustín.

1. INTRODUCCIÓN

La rapidez y simplicidad de la vida actual, probablemente han hecho que el hombre, en su afán de vivir más cómodamente, olvide todos aquellos detalles que hacen que ello pueda ser posible. Un tiempo exacto, las menores distancias, la solución mas eficiente, el dato más preciso o más conveniente, son apenas valores que hacen más llevadera nuestra vida, pero quizá sean la herramienta más importante para quienes pretenden, con su ciencia, crear los ambientes ideales para tal fin.

Medir fue, es y seguirá siendo, para algunos, una más de las actividades cotidianas; para otros, el camino que lleve a la consecución de resultados óptimos o esa distancia que puede separar lo bueno, de lo excelso. Lo que muchos ignoran es que a veces puede llegar a ser, para otros cuantos, una obsesión.

Los sistemas actuales de medición son la prueba veraz de la evolución de esta disciplina. El cuerpo fue, con toda certeza, el primer utensilio práctico para cuantificar la esencia de la distancia. Con él nació lo que hoy conocemos como pulgada, pié, yarda, braza, cuarta y también el ánimo de hacer tal muestra, antes que homogénea, respetuosa de las diferencias notables entre uno y otro. Con tal dedicación, nació el metro y con él, surgió el detonante de un cúmulo de unidades derivadas, ayudadas de otras básicas como las de masa y tiempo, originadas a partir de la misma necesidad.

Con la medida de la distancia, e incluso en muchos casos sin ella, se capitalizó la idea de una gran cantidad de genios fanáticos de la geometría, empeñados en publicar los resultados que años de trabajo habían arrojado acerca de lo que se ha venido conociendo como semirrecta, plano, área, volumen y ángulo, entre otros. Llegar a la menor medida de cualquiera de estos conceptos, es el trabajo que perturba a quienes pretenden laborar

sobre el legado de los que hicieron florecer esta labor, en el Antiguo Egipto, Sumeria, Babilonia y Grecia.

La ciencia, sus técnicas y todo lo que esto implique, nunca serán suficientes para cuantificar y corroborar con veracidad, la menor porción absoluta de lo que ya esta medido, pero algo probablemente sí sea cierto: cada vez que exista la necesidad de tener la más precisa de las mediciones, el hombre pondrá su empeño para lograrlo, lo que ya es garantía en el resultado.

Al sistema descrito en este documento, se le ha puesto el mayor empeño para obtener, a partir de las señales entregadas por un instrumento de medición (*Synchro*), la medida angular de alta resolución de su posición, elaborada sobre el procesamiento analógico y digital de las señales suministradas por dicha herramienta. Para lograr esta técnica, se ensamblan múltiples procesos prácticos de la electrónica, reseñados con detalle en el presente material, hasta llegar a la consecución de los objetivos trazados para este proyecto, como son: crear un sistema de medición de alta resolución (12 bits – 5 minutos de arco – 0.087°) implementado en un microcontrolador; un conjunto de salidas digitales y analógicas para futuras aplicaciones del sistema de medición (salida analógica a través de un convertidor analógico digital DAC, emulación de cuadratura incremental, salida digital de 12 bits y salida por modulación de ancho de pulso –PWM); desarrollar un dispositivo de dimensiones pequeñas; crear una interfaz gráfica de amable entorno, en la cual se pueda visualizar el resultado de la medida angular y finalmente, un protocolo de comunicaciones para este sistema, como lo es el estándar MODBUS ASCII.

El sistema de medición angular, se implementa en un microcontrolador, unidad central de procesamiento, de las señales digitales que intervienen en su resultado; estas señales son entregadas por un *Synchro*, instrumento capaz de transformar el movimiento de su eje, en señales que contienen la medida de la posición angular del mismo, con una resolución dependiente del algoritmo bajo el cual se analicen. El resultado del proyecto: un proceso altamente puntual (12 bits – 0.087° de resolución) sobre el producto real de la medida, gracias a la selección del algoritmo más conveniente para tal fin, del que hasta el momento, solo se tenía efectos gracias a simulaciones.

Durante la lectura de este libro, se recorrerá el camino trazado para hacer, de lo que fue un proyecto, la realidad que hoy se expone. En su segundo capítulo, se resume la información que es considerada más pertinente para la comprensión del trabajo realizado. El tercer capítulo contiene la descripción detallada del sistema y de cada una de las partes que lo conforman, como especificaciones, características y etapas de funcionamiento que van acompañadas de diagramas y gráficos para facilitar su entendimiento. En el capítulo cuarto se encontrarán los desarrollos realizados para obtener el completo funcionamiento del sistema. Finalmente, en los capítulos quinto y sexto, será expuesto un profundo análisis de resultados, dando paso a las conclusiones, que no son más que la recopilación de las metas y el fruto exitoso, de los objetivos trazados durante este proyecto.

2. MARCO TEÓRICO

El Sistema de Medición Angular Microcontrolado de Alta Resolución utilizando un *Synchro* ha sido diseñado para encontrar la posición angular de cualquier eje, que pueda ser acoplado a su rotor, mediante el método de conversión digital de *sobremuestreo*, explicado con detenimiento en este capítulo. Por lo tanto, se hace estrictamente necesaria la aclaración dada a los siguientes conceptos, para su correcta comprensión.

2.1. SENSORES Y TRANSDUCTORES DE POSICIÓN ANGULAR

Un transductor es un dispositivo al que se aplica cierta energía de entrada y devuelve otra de salida; ésta energía de salida suele ser diferente al tipo de energía de entrada, con una relación directa entre una y otra.

Existen varias formas de obtener la medida angular de un sistema. Dentro de los transductores y sensores más conocidos y utilizados, para tal fin, se encuentran los *encoder*, *synchro* y potenciómetros. Los potenciómetros generalmente son poco usados para aplicaciones de gran precisión y resolución angular, debido a las variaciones resistivas con respecto a la temperatura y a que comercialmente su comportamiento no es lineal, sin contar que carecen de robustez mecánica. A continuación se hará una descripción de los *encoder* y *synchro*.

2.1.1. **Encoder incremental**²: Su nombre se debe a que la señal generada está en formato de cuadratura incremental, es decir, dos señales (A y B) desfasadas 90° y con una señal de índice que cambia cada vez que se registre un giro completo (ver figura 2.1).

² CRUZ, Agustín. Control de Movimiento P.I.D. digital para servomotor de 0.5kW. Centro Tecnológico de Chihuahua. 2001.

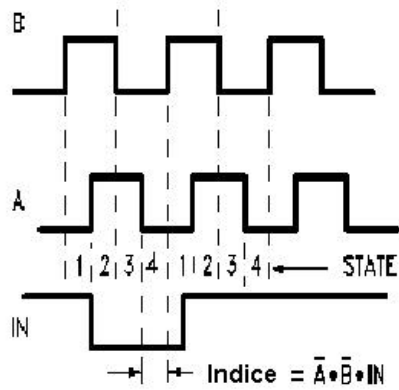


Figura 2.1. Señales en cuadratura incremental.

Un encoder incremental consta, básicamente, de un disco dividido en sectores oscuros y transparentes frente a una fuente de luz ubicada de uno de los lados del disco y un detector de luz del otro lado, como se muestra en la figura 2.2. Si el disco gira el fotorreceptor se encenderá y apagará alternadamente, produciendo un tren de pulsos, señal con la cual se puede obtener la medida angular de la posición. Generalmente dichos encoder logran alrededor de 100 o 6000 conteos por vuelta del disco y suele agregárseles un par adicional de foto transmisor y receptor para obtener la dirección en la cual gira el disco, y si es preciso, un tercer par con el fin de lograr un contador de vueltas.



Figura 2.2. a.

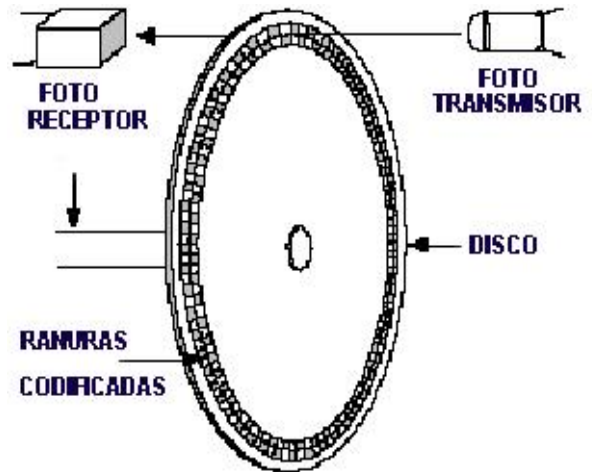


Figura 2.2. b.

Figura 2.2. a) Presentación comercial de los encoder. b) Grafica del funcionamiento.

Adicionalmente suele ubicarse una ranura como marcador de vuelta que indica desde dónde debe iniciar a contar en caso de que se pierda el suministro de energía. Sin

embargo hoy se fabrican encoders absolutos que resuelven este último problema, gracias a la distribución interna especial de las ranuras que permiten el paso o no de la luz. Es claro, entonces, la necesidad de un procesamiento digital de las señales entregadas por un encoder para obtener la medida del ángulo del sistema acoplado a él, con resultados en la resolución de no más 0.183° , a nivel comercial.

2.1.2. **Synchro**¹: Synchro es el término que denota a una familia de transductores, los cuales pueden ser conectados de varias formas para obtener la medida del cambio de ángulo o posición en un sistema. En apariencia, se asemejan a pequeños motores AC (Figura 2.3.a.) y son similares en construcción; tienen un rotor con uno o tres devanados (dependiendo del tipo de Synchro), capaces de rotar dentro del estator fijo. Mediante una señal de referencia AC (U_0) producen señales proporcionales, en amplitud, al ángulo de rotación del eje y con la misma frecuencia que la señal de referencia, como se observa en la figura 2.3.b. (U_1 , U_2 , U_3).

Para obtener el valor de la medida de ángulo entregada por el synchro, es necesario una forma de conversión específica de estas señales, que puede ser digital o analógica, como se describe más adelante.



Figura 2.3. a.

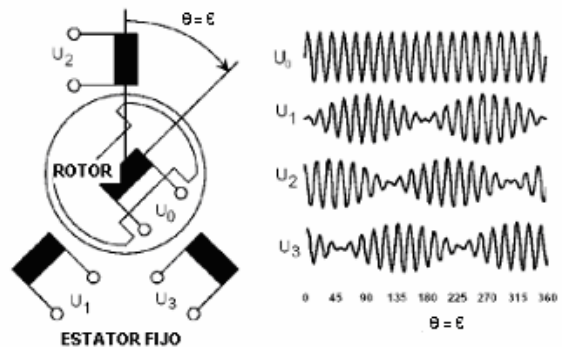


Figura 2.3.b.³

Figura 2.3. a) Presentación Comercial del **Synchro**. b) Señales del **Synchro** en función del ángulo.

¹ **BOYES**, Geoffrey. Synchro and Resolver Conversion. NAI Editors INC and Memory Devices. United Kingdom. 1980. 191 p.

³ **MUÑOZ**, Andrea., **PRIETO**, Alberto., **TINJACA**, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.

Dentro de la familia de los Synchro se encuentran los **"Torque Receiver"** (comúnmente llamados Synchro Receiver). Su apariencia es similar a la de cualquier otro synchro, pero con diferencia en las señales entregadas debido a su construcción interna, con respecto a otros tipos de la misma familia. Como su nombre lo indica, este tipo de synchro recibe un movimiento (torque) a través de su eje rotor y transforma la señal de referencia en tres señales con las que se obtendrá la medida de dicho movimiento angular.

Comparando el uso de los diferentes transductores, resultaría más simple hacer una manipulación analógica de las señales AC entregadas por un synchro, perdiendo con ello la capacidad de resolución que podrían entregar, si se procesaran digitalmente, como es el objetivo de este trabajo de grado. De forma digital, el desarrollo es más sencillo, debido a que siempre están disponibles la medida seno y coseno de la posición angular, a diferencia de los encoder (ya sea incremental o absoluto) que requieren de un procesamiento especial para obtener tal resultado. Los potenciómetros, por ejemplo, presentan un inconveniente diferente y es que requieren de un suministro de energía muy preciso de acuerdo a la resolución que se quiera obtener a la salida.

Los synchro y los encoder presentan mucho menos fricción dinámica o rozamiento entre el rotor y el estator durante el movimiento, en comparación con un potenciómetro o un encoder de contacto, aunque su costo sea más elevado que estos últimos. Pero a la hora de hablar de condiciones de operación, como son humedad, impactos, temperatura y vibración, los synchro son una muy buena alternativa y una muestra de ello es que vienen siendo utilizados desde hace aproximadamente 40 años en la industria militar y aeroespacial, con excelentes resultados.

La siguiente tabla muestra el comportamiento en cuanto a resolución de los diferentes tipos de transductores de ángulo.

Tipo de Transductor	Precisión	
Synchro Multipolar	7 Arcseg	0.019°
Synchro (Alta Precisión)	20 Arcseg	0.056°
Encoder Absoluto	23 Arcseg	0.064°
Synchro Estándar	3-7 Arcmin	0.05°-0.12°
Potenciómetro	7 Arcmin	0.12°
Encoder Incremental	11 Arcmin	0.183°
Encoder de contacto	26 Arcmin	0.433°

Tabla 2.1. Resolución de algunos transductores de ángulo.

2.2. FORMAS DE CONVERSIÓN DE UN SYNCHRO

Con el objeto de obtener los valores digitales del ángulo en el synchro, y de esta forma poder almacenar los valores correctos de seno y coseno de los ángulos, se requiere de técnicas para convertir de formato synchro a señales digitales. Existen principalmente cuatro tipos de conversores de synchro:

- *Tracking* o seguimiento.
- Aproximaciones Sucesivas.
- Conversión mediante *submuestreo*
- Conversión mediante *sobremuestreo*

2.2.1. **Tracking o seguimiento**¹: La conversión por tracking básicamente es una malla de seguimiento tipo 2, como la mostrada en la figura 2.4, que es encargada de generar el ángulo φ digital encargado de realizar el seguimiento de un ángulo θ de entrada.

¹ **BOYES**, Geoffrey. Synchro and Resolver Conversión. NAI Editors INC and Memory Devices. United Kingdom. 1980. 191 p.

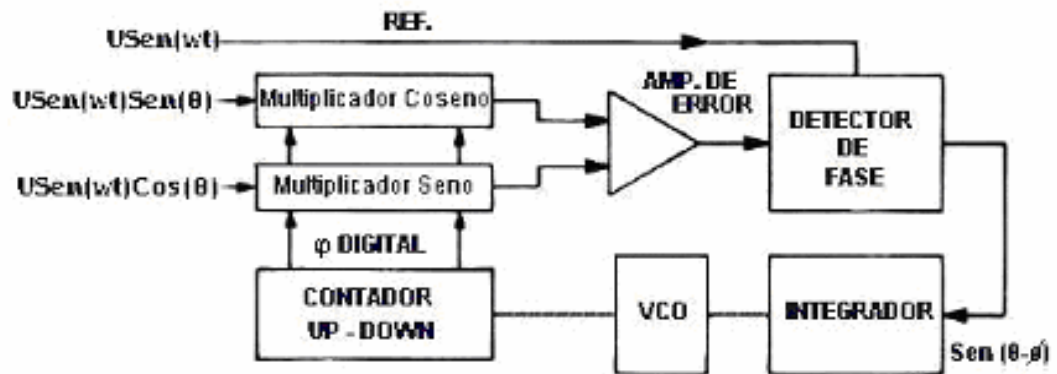


Figura 2.4. Malla de seguimiento.

Las siguientes son las señales a la salida del Synchro:

$$U_1(\theta, t) = U_o \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot \text{Sen}(\theta)$$

$$U_2(\theta, t) = U_o \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot \text{Cos}(\theta),$$

El coseno y el seno del ángulo φ , es aplicado al multiplicador, respectivamente, de lo cual se obtienen las siguientes señales:

$$U \cdot \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot \text{Sen}(\theta) \cdot \text{Cos}(\varphi)$$

$$U \cdot \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot \text{Cos}(\theta) \cdot \text{Sen}(\varphi)$$

El amplificador de error proporciona la siguiente resta de señales:

$$U \cdot \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot (\text{Sen}(\theta) \cdot \text{Cos}(\varphi) - \text{Cos}(\theta) \cdot \text{Sen}(\varphi))$$

Este resultado, por identidad trigonométrica, es igual a:

$$U \cdot \text{Sen}(\omega_{\text{ref}} \cdot t) \cdot \text{Sen}(\theta - \varphi)$$

Este error AC es demodulado por el detector sensitivo de fase que utiliza la señal de referencia. La señal entregada por el dicho detector es proporcional a:

$$\text{Sen}(\theta - \varphi)$$

Con este error se alimenta un integrador, cuya salida alimenta un oscilador controlado por voltaje (VCO). Este VCO genera tantos pulsos al contador como sean necesarios hasta que el error se hace cero, debido a que su frecuencia contiene la información de la diferencia entre los ángulos θ y φ . Lo anterior ocurre cuando los dos ángulos son iguales ($\theta = \varphi$), para obtener finalmente, un ángulo digital que sigue a la entrada sin tener que utilizar una instrucción de algún conversor externo. Si la entrada no cambia, el conversor no realiza ninguna operación.

Incluir un integrador en esta malla, genera un sistema de tipo 2, los cuales se caracterizan por tener error cero para movimientos de velocidad constante o para entradas estacionarias. La señal de error está presente únicamente durante períodos de aceleración o desaceleración.

2.2.2. **Aproximaciones Sucesivas**¹: En los conversores por aproximaciones sucesivas, la señal del synchro es muestreada en el valor máximo de la señal de referencia por amplificadores de muestreo y retención que proporcionan dos voltajes DC, por ejemplo:

$$V \text{ Sen}\theta \text{ y } V \text{ Cos}\theta$$

Estos voltajes son procesados por el conversor para dar una representación digital del ángulo θ , por un método similar al utilizado por el conversor analógico-digital mediante aproximaciones sucesivas. Los conversores por aproximaciones sucesivas son a menudo usados en aplicaciones en las que un gran número de canales de información de synchro, tienen que ser procesados. La información analógica de cada canal es mantenida sobre los amplificadores de muestreo y retención antes de la conversión. Un sistema de

¹ **BOYES**, Geoffrey. Synchro and Resolver Conversión. NAI Editors INC and Memory Devices. United Kingdom. 1980. 191 p.

multiplexing (cambiar constantemente el valor de la entrada y/o salida) es usado para pasar de una a otra de las salidas de muestreo y retención dentro del conversor central.

En ciertos aspectos la conversión es similar al conversor de tracking o seguimiento, en el hecho de que es una malla formada por multiplicadores de tipo seno y coseno, pero en este caso, la señal de error es anulada por el uso de un registro de aproximaciones sucesivas en vez de un VCO y un contador *up-down* (que se incrementa o se disminuye). La malla no actúa en un modo de seguimiento continuo sino que requiere de una señal externa para iniciar cada conversión.

La principal desventaja de los conversores por aproximaciones sucesivas es que los datos del synchro tienen que ser mantenidos por los amplificadores de muestreo y retención en el pico de la señal de referencia. Otro problema es su falta de inmunidad al ruido. Cualquier ruido altera la relación de los voltajes del synchro causando errores. La interferencia de ruido presenta con frecuencia gran cantidad de picos y aunque algunos picos ocurran en tiempos diferentes al de muestreo, esto afecta directamente la precisión de la salida.

2.2.3. **Conversión synchro-digital mediante *submuestreo***⁵: El método básicamente esta descrito en la figura 2.5. Las salidas seno y coseno moduladas, u_1 y u_2 del synchro, deben ser muestreadas a la misma frecuencia de referencia del synchro. Esto, llamado submuestreo, demodula ambas señales, por lo que las muestras digitales de $u_1(n)$ y $u_2(n)$ son el seno y el coseno de el ángulo ϵ , respectivamente.

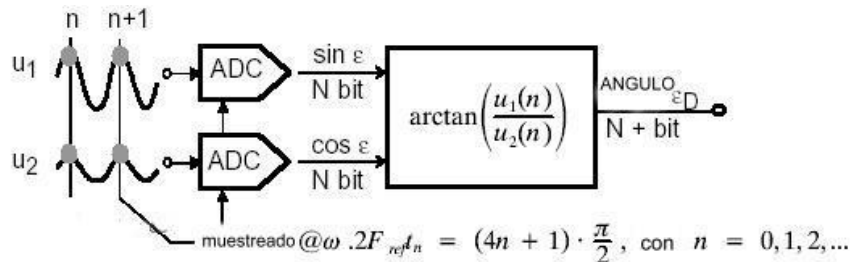


Figura 2.5. Diagrama básico de la conversión mediante submuestreo

⁵ TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed. Application Report. Texas Instruments. February 2000. <http://www.ti.com>

La posición puede ser determinada por una función arcotangente del cociente de las muestras de la función seno y la función coseno, de la siguiente forma:

$$\varepsilon(n) = \begin{cases} \arctan\left(\frac{u_1(n)}{u_2(n)}\right), & \text{si } u_2(n) \geq 0 \\ \pi + \arctan\left(\frac{u_2(n)}{u_1(n)}\right), & \text{si } u_2(n) < 0 \end{cases}$$

Para asegurar la precisión, ambas señales, u1 y u2 del synchro, tienen que ser muestreadas simultáneamente, sincronizadas con la frecuencia de referencia.

Con el fin de implementar en la realidad este algoritmo, existe un inconveniente principal y es que necesita un conversor analógico digital que tenga, por lo menos, la resolución requerida de la medida final, lo que hace el trabajo ineficiente, por la obligación de procesar externamente la conversión y el muestreo, y posteriormente, la medida del ángulo. Todo lo anterior, puede hacerse en un mismo circuito integrado, por ejemplo en un microcontrolador, que ejecute internamente la conversión y todas las labores adicionales, como incrementar la resolución, filtrado e inmunidad a ruido, etc.

2.2.4. Conversión synchro-digital mediante sobremuestreo⁵: Básicamente la figura 2.6. describe el funcionamiento del algoritmo de sobremuestreo. La resolución del ángulo digitalmente medido, la proporciona la malla final de ajuste de velocidad e interpolación de la posición, como se explicará más adelante.

⁵ TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed. Application Report. Texas Instruments. February 2000. <http://www.ti.com>

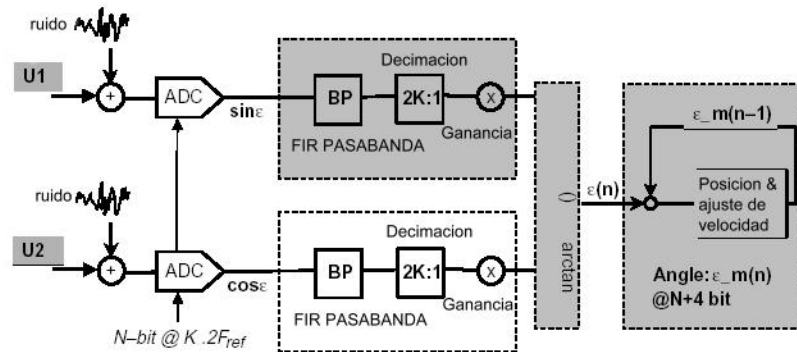


Figura 2.6. Diagrama en bloques de la conversión mediante sobremuestreo

Las señales seno y coseno moduladas entregadas por el synchro, $u1$ y $u2$, son muestreadas a $2K$ -veces la frecuencia de referencia. Esto es el equivalente a sobremuestrear K -veces. Por tal razón el algoritmo recibe el nombre de “sobremuestreo”, debido a que se muestrea K veces por encima de la frecuencia de Nyquist, criterio que establece que para que una señal muestreada pueda ser reconstruida en su totalidad, necesita mínimo dos muestras en un período. El ruido es añadido para asegurar, en simulación, que el ruido de cuantización no esté correlacionado con las señales entregadas por el synchro. En la práctica, dicho ruido está implícito en el desarrollo, debido al tratamiento analógico que requieren las señales de entrada, así como los efectos mecánicos que inciden directamente sobre el movimiento del rotor del synchro. El filtro FIR pasabanda actúa como un filtro *anti-aliasing* digital, para evitar la imposibilidad de reconstrucción de la señal por traslape y mejora la resolución de las muestras en 0.5 bits debido a la reducción del ancho de banda de f_{ref} a $(f_{ref}/2)$. En un algoritmo que utilice sobremuestreo, la decimación consiste en tomar solamente la $2K$ -ésima muestra en un punto del algoritmo, para reducir la frecuencia de muestreo a f_{ref} y de esta forma, bajar $2K$ -veces su tiempo de procesamiento. Bajar dicha frecuencia de muestreo no implica el incumplimiento del criterio de Nyquist, debido a que se supone una velocidad baja en el movimiento del rotor del synchro, es decir, la frecuencia de la señal envolvente es mucho menor a la frecuencia de la señal de referencia. En el caso particular del algoritmo de sobremuestreo, la decimación se realiza a la salida de los filtros de entrada. De esta forma ya se tienen las señales seno y coseno filtradas. Ahora, para obtener la posición angular se realiza la función arcotangente del cociente de las señales seno y coseno demoduladas. Para mejorar la calidad de la medida, se implementa una malla cerrada de interpolación, que cumplirá esta función dependiendo de su ancho de banda, como se

explicará a continuación. Otra función de esta malla es compensar el *group delay* – *retardo de grupo* - (velocidad de cambio de fase con respecto a la frecuencia angular) que generan los filtros FIR pasabanda.

2.2.4.1. **Filtro FIR pasabanda y decimación:** El filtro pasabanda de decimación es un filtro FIR simétrico de fase lineal. El orden del filtro depende principalmente de la tasa de sobremuestreo. La frecuencia central del filtro FIR es igual a la frecuencia de referencia del Synchro, como se muestra en la figura 2.7.

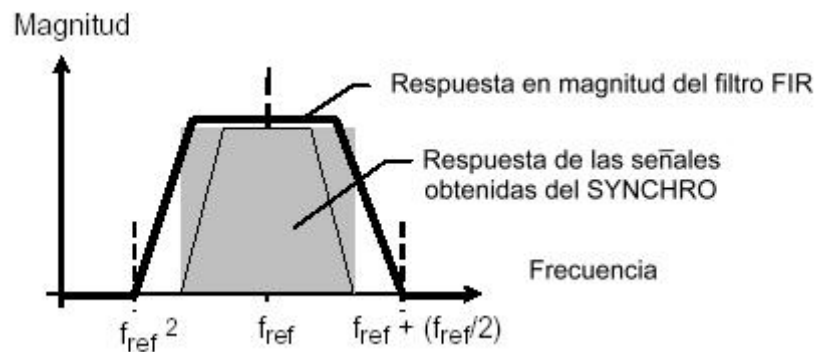


Figura 2.7. Respuesta en magnitud del filtro FIR de Decimación

La tarea del filtro FIR es reducir el ancho de banda a $[f_{ref} - (f_{ref}/2)]$, lo que es necesario para que la decimación cumpla el criterio de *Nyquist* (la frecuencia de muestreo debe ser mayor a dos veces la máxima frecuencia de la señal a muestrear). El filtro FIR es, de hecho, un filtro digital anti-alias el cual adicionalmente mejora la resolución entre la banda de interés, en 0.5 bits. Para sobremuestreo k -veces, la decimación en tiempo se hace mas precisa tomando solamente muestras cada $2k$ -veces de la salida del filtro. Esto es equivalente a filtrar solo la $2k$ -ésima muestra de la señal de entrada. La decimación en tiempo demodulará las señales dentro de la banda base $[0 - f_{ref}/2]$.

Por otra parte las señales son retrasadas en tiempo. Esto es debido al *group delay* constante del filtro del FIR simétrico.

Para un orden N del filtro FIR pasabanda, el *group delay* es exactamente $(N-1)/2$ muestras. Cuando el filtro FIR tiene un orden de $(4K+1)$, el *group delay* es equivalente a un periodo de la frecuencia de referencia.

2.2.4.2. **Función arcotangente:** El ángulo se obtiene de el cociente del las señales seno y coseno filtradas por el filtro FIR.

$$\varepsilon_{FIR}(n) = \begin{cases} \arctan\left(\frac{u_{1,FIR}(n)}{u_{2,FIR}(n)}\right), & u_{2,FIR}(n) \geq 0 \\ \pi + \arctan\left(\frac{u_{2,FIR}(n)}{u_{1,FIR}(n)}\right), & u_{2,FIR}(n) < 0 \end{cases}$$

Comparado con el método básico o de submuestreo, el ángulo digitalizado ahora tiene una mayor resolución dada por el filtro FIR. Sin embargo, el ángulo digitalizado tiene un retardo debido al group delay de dicho filtro. El error del ángulo digitalizado, en la malla de interpolación, es proporcional a la velocidad de rotación del sistema.

2.2.4.3. **Malla cerrada de interpolación y ajuste de velocidad:** La figura 2.8. muestra la función de la malla de interpolación de ángulo y ajuste de velocidad.

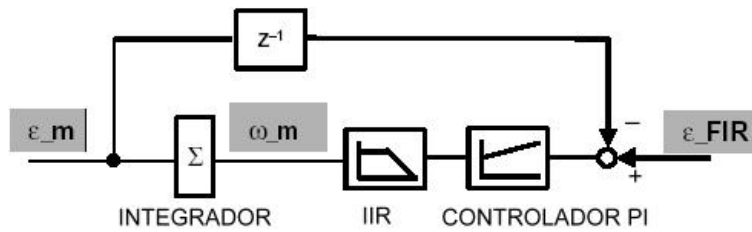


Figura 2.8. Malla de interpolación y ajuste de velocidad

La malla cerrada consta de un controlador P+I, un filtro IIR de primer orden y un integrador. Todo reunido, es básicamente un filtro pasabajos, el cual determina la resolución final del algoritmo. La resolución aumenta con la disminución del ancho de banda de este filtro, que a su vez, implica un mayor tiempo de establecimiento de la malla. Dichas características hacen que no tenga error integral, debido a que la malla no sigue la medida sino que la interpola, es decir, obtiene valores intermedios de las variaciones de la entrada, por su característica de filtro pasabajos. El efecto “doble” integral de la malla garantiza error cero a entrada rampa, lo que compensa los retardos generados por los filtros de entrada. Los parámetros de dicha malla definen su ancho de banda.

Entonces, dentro de las principales propiedades de la malla de interpolación, están:

- Mejorar la precisión de la medida angular ε_m (medida angular entregada por la malla de interpolación como resultado al tratamiento realizado sobre la medida ε_{FIR} entregada por la función arcotangente – ver figura 2.8). El incremento en la resolución depende del ancho de banda seleccionado.
- Compensar el retraso del filtro FIR, por lo tanto ε_m no sufre retraso de velocidad.
- Integrar la velocidad angular ε_m .

El resultado con esta malla es una mayor precisión de la posición angular. La señal ε_m es retrasada con el mismo tiempo que el group delay de los FIR. La señal retrasada $\varepsilon_{m(n-1)}$ (como resultado de retrasar ε_m un período de muestreo, a través del bloque Z^{-1}) es comparada con la señal retrasada por el filtro FIR (ε_{FIR}). La malla fuerza a ambas medidas a ser iguales. Para una velocidad constante, la salida de la malla ε_m es idéntica al ángulo en el que se encuentra el synchro.

2.3. PROTOCOLO MODBUS ⁴

Modbus es un protocolo de comunicación, utilizado por todos los equipos controladores de la casa *Modicon*. Este protocolo define una estructura de mensaje especial que, los controladores que lo usen, reconocerán y manejarán en cualquier tipo de red de comunicaciones. En él se describe el proceso que un equipo controlador utiliza para acceder a otro dispositivo, cómo responderá a un llamado de otro dispositivo e igualmente determinar cuántos errores se detectarán y reportarán en la transmisión. Modbus establece un formato estándar de representación y contenido para los campos que deben ser establecidos antes de la transmisión.

⁴ Modicon Modbus Protocol Reference Guide. Modicon Inc., Industrial Automation Systems. <http://www.modbus.org>

Durante una comunicación por medio de protocolo Modbus, se define qué controladores o dispositivos Maestros, desean información sobre otro dispositivo, denominado esclavo. En esta comunicación se determina de qué dirección necesita información y la forma de reconocer el mensaje que se encuentra en esa dirección, qué tipo de acción debe ejecutar y extrae cualquier otro tipo de información de dicho mensaje. Si se requiere contestar el mensaje, el esclavo debe construir un mensaje y enviarlo, también, con el formato de Modbus.

Los controladores pueden ser configurados para cualquiera de dos modos posibles: RTU y ASCII. El usuario selecciona este modo de comunicación y otros parámetros del puerto serial como tasa de transmisión (*baud rate*), paridad (*parity*), entre otros.

2.3.1. **Modo ASCII:** Cuando los controladores se configuran para comunicarse mediante Modbus ASCII (*American Standard Code for Information Interchange*), un mensaje de byte de 8 bits, es enviado como uno o dos caracteres ASCII. La principal ventaja de este modo es que permite intervalos de tiempo de más de un segundo entre comunicación y comunicación, sin causar error alguno.

INICIO	DIRECCIÓN	FUNCIÓN	DATO	LRC	FIN
1 Carácter :	2 Caráct.	2 Caráct.	n Caráct.	2 Carácter	2 Caráct. CRLF

Figura 2.9. Trama de Modbus ASCII.

La figura 2.9. muestra la trama de la transmisión de Modbus en modo ASCII. Los caracteres con que se pueden llenar los espacios de la trama, deben ser hexadecimales (0-9 y A-F). Un bit de inicio que debe ser el carácter ":" con su debida representación en hexadecimal. El campo de dirección contiene dos caracteres que indican el registro del módulo esclavo a manipular, de un rango decimal entre 0 y 247. Cuando el esclavo envía su respuesta, pone su propia dirección en el campo de dirección para que el maestro entienda quien está contestando. El campo de función se constituye de dos caracteres ASCII de un rango entre 1- 255 decimal, con los cuales se especifican funciones como la #03 (*READ HOLDING REGISTER*), en la cual se lee la información de la dirección de

registro descrita en la trama, y la #06 (*PRESET SINGLE REGISTER*), en la que se escribe al registro en dicha dirección del módulo esclavo, entre otras funciones.

El campo de datos es construido usando pares de dígitos hexadecimales, en el rango de 00 a FF hexadecimal. En él se ubica la información requerida por la acción enviada en el campo “función” del mensaje maestro. El campo LRC contiene dos caracteres como resultado de un Chequeo de Redundancia Longitudinal (*Longitudinal Redundancy Check*), calculado sobre el contenido del mensaje completo. Un par de caracteres, finalmente cierran la trama (“**CRLF**”).

2.4. MICROCONTROLADOR PIC18F452³

El microcontrolador PIC18F452 de Microchip cuenta con un conjunto de 75 instrucciones de 16 y 32 bits, y una alta velocidad de procesamiento gracias a que realiza simultáneamente el ciclo de ejecución (*execute*) de una instrucción y el de búsqueda (*fetch*) de la siguiente. A este sistema de procesamiento paralelo de instrucciones se le conoce como *pipelining*. Un circuito de enganche de fase interno puede ser utilizado para multiplicar la frecuencia del oscilador externo por 4, lo que puede resultar útil para evitar algunos problemas de interferencias electromagnéticas que presentan los osciladores de alta frecuencia.

³ MUÑOZ, Andrea., PRIETO, Alberto., TINJACA, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.

Característica	Valor
Frecuencia de Operación	DC a 40MHz
Memoria de Programa (Bytes)	32 K
Memoria de Programa FLASH	16384
Memoria de Datos (Bytes)	1536
Memoria EEPROM (Bytes)	256
Fuentes de Interrupción	18
Puertos de Entrada/Salida	5
Instrucciones	75
Temporizadores	4
Módulos de PWM	2
Comunicaciones seriales	SPI, I ² C, USART
Canales de Conversión Analógica-Digital	8 canales de 10 bits

Tabla 2.2. Características de microcontrolador PIC18F452

Las características más importantes del PIC18F452 se describen brevemente en la tabla 2.2. Existen tres bloques de memoria dentro de este microcontrolador: la memoria FLASH, para almacenamiento del programa, la memoria RAM, para almacenamiento temporal de datos, y la memoria EEPROM, para almacenamiento permanente de datos. Las memorias de datos y de programa utilizan buses separados, por lo que es posible acceder a estos dispositivos de almacenamiento al mismo tiempo.

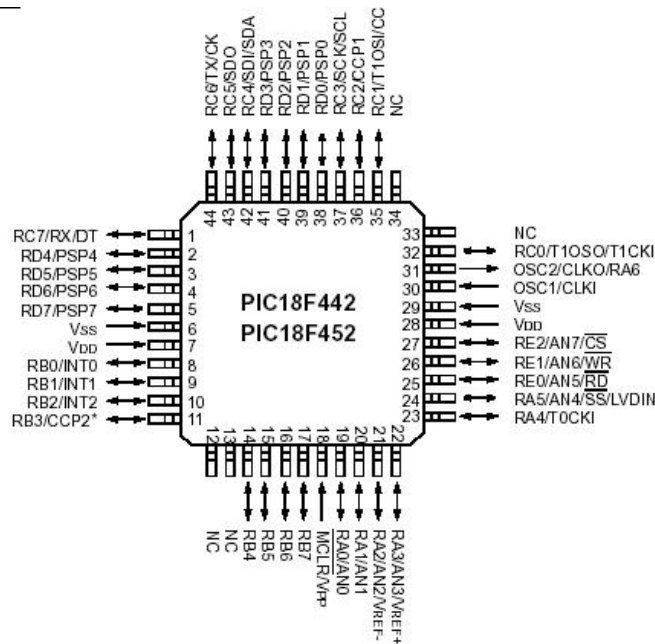


Figura 2.10. Distribución de terminales del PIC18F452

Cuenta con 32K bytes de memoria FLASH, a los cuales se les puede asignar una dirección con un registro contador de programa de 21 bits. La memoria EEPROM tiene una capacidad de 256 bytes. El PIC18F452 puede tener hasta 18 fuentes de interrupción, provenientes ya sea de periféricos internos o externos. A cada una de estas interrupciones, es posible asignarle uno de once niveles de prioridad diferentes.

Consta de cinco puertos de entrada/salida, y cada uno de ellos puede ser configurado de diferentes maneras para utilizar los periféricos. La figura 2.10. muestra la distribución de terminales dentro del microcontrolador. Los terminales 0 a 5 del puerto A, y 1 y 2 del puerto E pueden ser configurados como entradas analógicas a los canales de conversión analógica-digital de 10 bits. Los terminales 0, 1, 2 y 4 del puerto B pueden ser usados para realizar interrupciones desde señales externas. Para comunicación serial, en modo SPI o I2C, pueden ser utilizados los terminales 3, 4 y 5 del puerto C. Los terminales de reloj y transmisión para comunicación serial por USART corresponden a los terminales RC6 y RC7, respectivamente. Las salidas de los módulos de PWM internos se pueden configurar en los terminales RC1 y RC2. El terminal RC0 puede ser una salida de tiempo generada desde el microcontrolador. Los demás terminales no pertenecen a ningún periférico, sino que están designados a otras funciones especiales como programación, oscilador externo y puerto esclavo.

2.5. MOTORES ELÉCTRICOS³

Existen diferentes tipos de motores eléctricos. Entre ellos se encuentran: motores de corriente continua (DC), motores de paso y motores de corriente alterna (AC).

Probablemente el uso más sencillo sea el de motores DC, dado que es relativamente fácil encontrar algunos de este tipo con un bajo consumo de potencia y lograr un movimiento adecuado para el proceso a controlar, gracias a que es posible acoplarlos a motorreductores, además son fáciles de controlar a partir de los módulos de PWM de microcontroladores comerciales. El diseño de los motores DC es bastante simple: un campo magnético permanente es creado en el estator por un imán permanente, o por bobinas electromagnéticas. Opuesto al campo del estator está el campo de la armadura, el cual es generado por un flujo electromagnético cambiante que viene de las bobinas localizadas sobre el rotor.

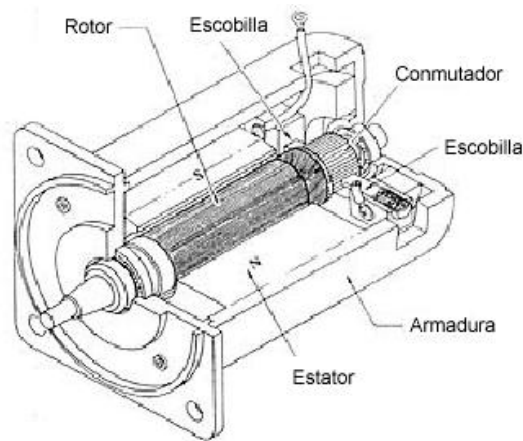


Figura 2.11. Configuración de un motor DC.

Los polos magnéticos del campo de la armadura intentarán alinearse con los polos magnéticos opuestos, generados por el campo del estator. La sección del rotor donde la electricidad entra en los embobinados del rotor se llama conmutador. La electricidad es conducida entre el rotor y el estator por las escobillas de cobre o los conductos de grafito

³ MUÑOZ, Andrea., PRIETO, Alberto., TINJACA, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.

(localizados sobre el rotor), los cuales están en contacto con los anillos que se encuentran sobre el estator.

El motor gira hacia el punto de alineación de los polos, por ende, las escobillas saltan a través de un espacio entre los anillos del estator. La velocidad adquirida lleva al motor a adelantar este espacio. Cuando las escobillas consiguen estar al otro lado del espacio, ellas entran de nuevo en contacto con los anillos del estator y se invierte la polaridad del voltaje en este conjunto de anillos. El motor nuevamente empieza a acelerarse con el objeto de conseguir el conjunto opuesto de polos (La velocidad adquirida lleva al motor a pasar el polo original del punto de alineamiento). Esto continúa sucediendo mientras el motor está en movimiento. El sentido de rotación de eje de un motor DC es controlado por la dirección del flujo electromagnético generado por las bobinas localizadas en el rotor. Es decir, el movimiento del motor puede ser alterado modificando dicho flujo. Para este caso particular el flujo magnético de una bobina puede ser modificado variando la corriente que circula a través de la misma. Como el efecto deseado es el de inversión de marcha, basta con invertir el sentido de circulación de la corriente por las bobinas del rotor. Si el motor es alimentado por una fuente de voltaje, al intercambiar los terminales de alimentación, la corriente circulará en sentido contrario y se generará una inversión de marcha en el motor.

2.6. PUNTE H³

El circuito denominado puente H, suministra la corriente necesaria para mover un motor DC, utiliza una sola fuente de alimentación y recibe su nombre debido a que su esquema se asemeja a la forma de la letra "H".

³ **MUÑOZ**, Andrea., **PRIETO**, Alberto., **TINJACA**, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.

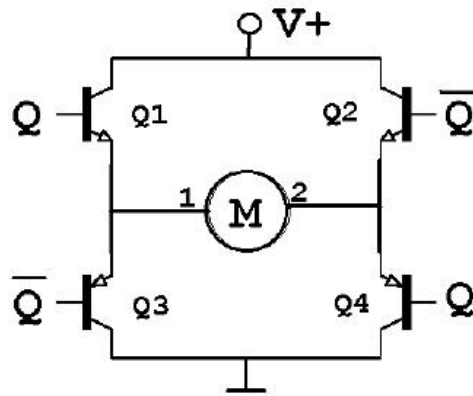


Figura 2.12. Configuración de un Puente H.

Como se observa en la figura 2.12, la señal de control Q controla el paso de corriente al motor de la siguiente forma: cuando la señal Q es uno lógico, $Q2$ y $Q3$ están inactivos, mientras la corriente circula por $Q1$ y $Q4$, fluyendo de los terminales 1 al 2 del motor; cuando Q es cero lógico ocurre lo contrario, $Q4$ y $Q1$ están inactivos, mientras la corriente circula por $Q2$ y $Q3$, fluyendo de los terminales 2 al 1 del motor, haciendo que el eje del motor cambie su sentido de giro.

3. ESPECIFICACIONES

3.1. DESCRIPCIÓN GENERAL

El sistema de medición angular microcontrolado, de alta resolución, utilizando un synchro, recibe 3 señales sinusoidales desfasadas entre sí y cuya amplitud, en conjunto, corresponde al ángulo del movimiento del rotor del synchro. Dichas señales son manipuladas analógicamente para obtener una señal seno y otra coseno. Estas últimas son tratadas digitalmente dentro del microcontrolador, mediante el algoritmo de sobremuestreo, de la siguiente forma: primero se muestrean, luego se filtran y seguidamente se procede a calcular la función arcotangente del cociente entre las muestras seno y coseno respectivamente. El resultado es involucrado en una malla de control digital para la mejora de la resolución del ángulo finalmente obtenido. Dicha medida se manipula para obtener las salidas en cuadratura incremental, PWM, protocolo MODBUS ASCII y digital de 12 bits para ser convertida a analógica mediante un DAC. La señal de PWM es entregada a un Puente H para mover un motor DC acoplado al synchro y dicho movimiento realimentará una malla de control (aplicación del sistema de medición). Los ángulos de control de dicha aplicación, son establecidos mediante una interfaz gráfica que se comunica con el sistema a través del protocolo Modbus ASCII.

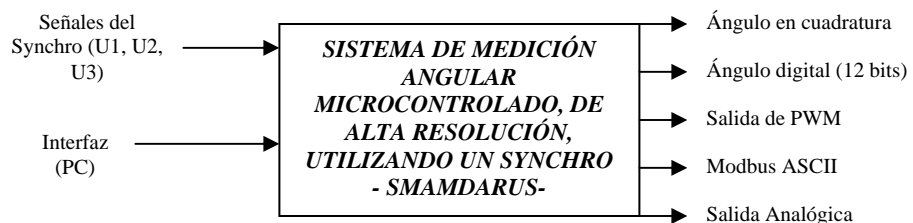


Figura 3.1. Esquema general de entradas y salidas del sistema.

En la siguiente tabla, se encontrarán las características más relevantes del sistema de medición angular.

CARACTERÍSTICAS ELÉCTRICAS		
PARÁMETROS	CONDICIONES	VALOR
Alimentación externa		120 V _{RMS}
Alimentación interna	Temp = 25°C , I _{máx} =68 mA	Tarjeta analógica: 5 V _{DC} Tarjeta digital: 9.8 V _{DC}
Potencia disipada (sin aplicación)	Temp = 25°C , I _{máx} =68 mA	680 mW
Frecuencia de Procesamiento		40MHz
Frecuencia de Muestreo del algoritmo		400 Hz
Temperatura máxima de operación		60°C
CARACTERÍSTICAS MECÁNICAS		
Aplicación		2 grados de libertad
Acople entre motor y synchro		1:1
Resolución en plano horizontal		12 bits – 0.087° (5 minutos de arco)
Resolución en plano vertical		12 bits – 0.087° (5 minutos de arco)
CARACTERÍSTICAS DE LA INTERFAZ GRÁFICA		
Software de programación		Visual Studio 6.0
Lenguaje		Basic
Pantallas		<ul style="list-style-type: none"> ▪ 1) Medición ▪ 2) Control
Comunicación		Serial - Modbus ASCII
Requisitos		Windows© 95 o superior.
CARACTERÍSTICAS DE CONTROL		
Sensor		Synchro
Medida del error		Digital
Variable controlada		Posición
Actuador		Puente H
CARACTERÍSTICAS DE COMUNICACIÓN		
Formato de Modbus		ASCII
Velocidad		19200 baudios/s
Paridad		Ninguna
Bit de parada		0
Puertos		Serial COM1 y COM2
Timeout		1000 ms
Funciones		<ul style="list-style-type: none"> ▪ #03 READ HOLDING REGISTER ▪ #06 PRESET SINGLE REGISTER

Tabla 3.1. Principales características del sistema de medición.

Esta tabla contiene información eléctrica, mecánica, de comunicaciones y otras, necesarias para la utilización final del sistema y su completa comprensión. Se han destacado los parámetros más importantes de las diferentes perspectivas desde las cuales, puede ser analizado el trabajo.

3.2. DIAGRAMA EN BLOQUES

El diagrama de la figura 3.2. muestra el esquema general del Sistema de Medición Angular Microcontrolado de Alta Resolución Utilizando un Synchro. A continuación se describe brevemente cada una de sus partes

- 3.2.1. **Etapas de procesamiento analógico:** En esta etapa, se recibe la señal cuadrada de 400 Hz generada por el microcontrolador, para generar una señal seno, de la misma frecuencia, con la cual se alimentará la referencia del synchro. En ella también se encuentra el circuito transformador de *Scott*, encargado de recibir las tres señales desfasadas 120° entregadas por el synchro, para generar una señal seno y otra coseno; asimismo, un circuito integrado manejador del estándar RS-232, encargado de la transmisión serial entre el módulo maestro del protocolo Modbus y el esclavo.
- 3.2.2. **Etapas de procesamiento digital:** La etapa de procesamiento digital tiene varias funciones; una de ellas, es la de generar una señal cuadrada de 400Hz con la que se alimentará un filtro analógico para la generación de la señal de referencia del synchro.
- **Algoritmo de sobremuestreo:** En él se procesan las muestras de las señales seno y coseno entregadas por la etapa analógica y sobre las cuales se calculará la medida digital de la posición angular del giro del rotor del synchro. Es quizá, la parte central de este proyecto.

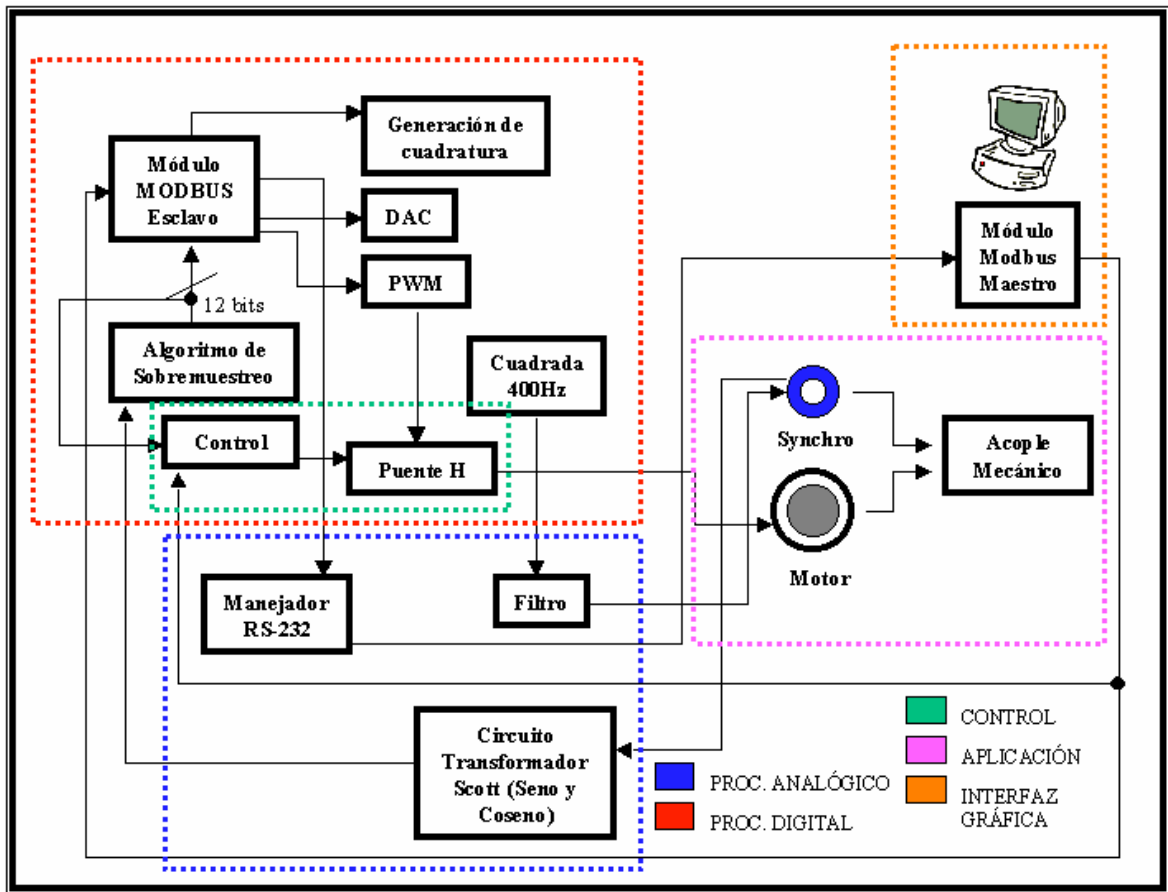


Figura 3.2. Diagrama en bloques.

- **Módulo MODBUS Esclavo:** Este módulo es el encargado de procesar la trama de datos enviada mediante el protocolo MODBUS tipo ASCII para que, durante la medición y en la malla de control, el sistema de medición sea entendido como Esclavo por parte del módulo Maestro.
- **Control:** En esta parte se ubica un controlador digital proporcional integral (P+I) encargado de procesar la medida angular del montaje mecánico, para conseguir controlar la aplicación en un nivel de referencia establecido desde la interfaz gráfica.
- **Generación de Cuadratura:** Mediante esta sección se recibe la medida digital entregada por el algoritmo de sobremuestreo para procesarla digitalmente y obtener las tres señales que conforman las componentes de la medida en cuadratura incremental.
- **DAC:** El convertidor Digital-Analógico es el encargado de pasar la medida entregada por el algoritmo de sobremuestreo en una señal entre 0 V y 5 V.

- PWM: El modulador por ancho de pulso alimenta el puente H, encargado del movimiento del motor en la aplicación de control.
- Puente H: Recibe la señal del PWM para habilitar su conducción de corriente y mediante otro terminal, recibe la señal que controla la dirección de giro del motor.

3.2.3. **Etapa de control:** En esta etapa reúnen los componentes necesarios para establecer el control de la aplicación mecánica de dos grados de libertad, es decir, que pueda moverse sobre un plano vertical y otro horizontal. Está conformado por la etapa de control y el puente H, pertenecientes a la etapa de procesamiento digital.

3.2.4. **Etapa de interfaz gráfica:** Es la encargada de mostrar, en un esquema agradable, la medida angular recibida por el módulo Maestro, que en este caso es cualquier computador bajo plataforma Windows, capaz de conectarse al sistema de medición a través del puerto serial, para iniciar la transmisión de datos mediante protocolo Modbus. En la interfaz pueden habilitarse dos funciones de este protocolo para leer o escribir sobre los registros disponibles en el módulo esclavo (sistema de medición), que en este caso son siete diferentes. También en ella, pueden establecerse los niveles de referencia para el control de la aplicación de dos grados de libertad y visualizar su resultado.

3.2.5. **Etapa de aplicación:** Para verificar el funcionamiento del sistema de medición angular como sensor de posición dentro de un lazo de control, se diseñó y construyó un montaje mecánico de dos grados de libertad, para el cual se acopló, con una relación de 1:1, un motor DC a un synchro, con el fin de que este último haga las veces de intérprete de la posición angular del motor. Con dicha medida y los niveles de referencia establecidos en la interfaz gráfica, se realiza el control digital de la posición del montaje mecánico.

4. DESARROLLOS

El sistema de medición angular, en su totalidad, fue realizado como se muestra en la figura 4.1, de acuerdo con las partes mencionadas en el capítulo 3. Los módulos de funcionamiento están básicamente separados en una tarjeta analógica, que contiene las etapas que procesan este tipo de señales; una tarjeta digital que contiene al microcontrolador y los demás circuitos encargados de procesar las señales discretas y la etapa mecánica sobre la cual convergen los resultados de las dos fases anteriores.

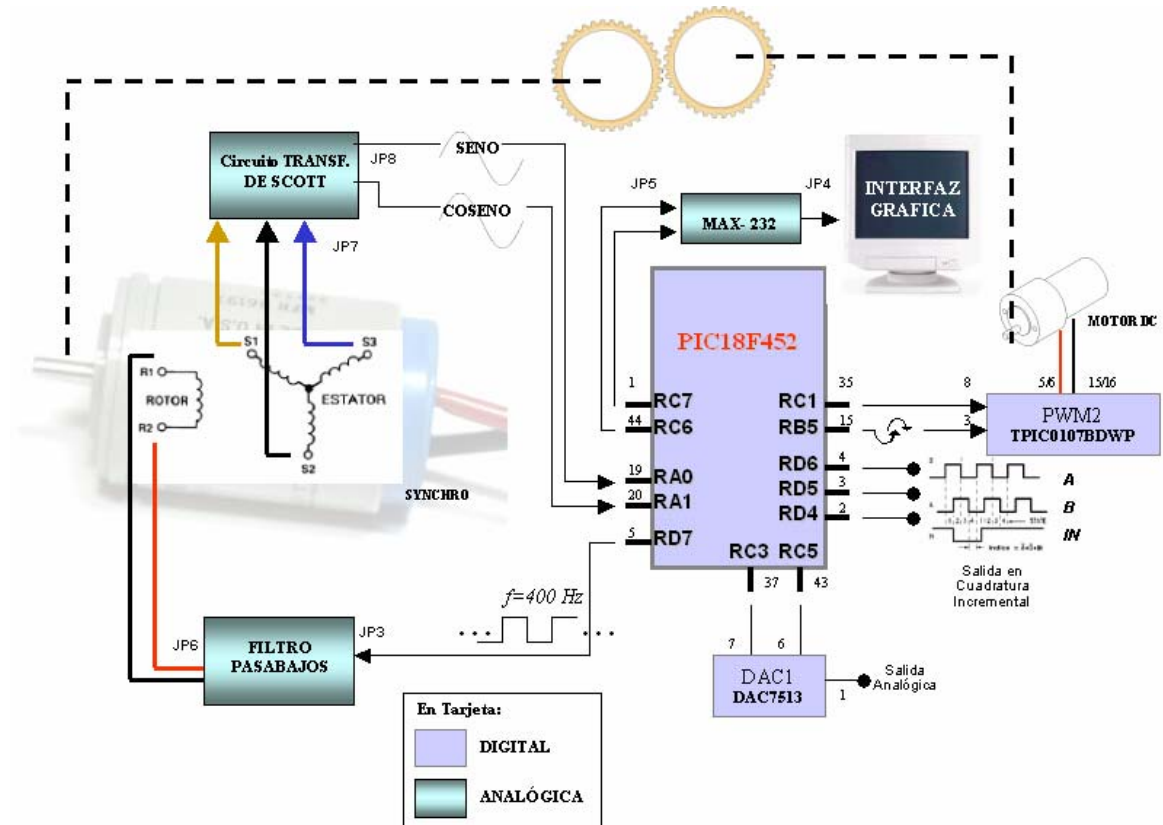


Figura 4.1. Esquema de desarrollo del sistema de medición angular.

A continuación se describe el camino seguido para generar la medida digital del ángulo, a partir del movimiento del rotor del sincro.

4.1. DESCRIPCION GENERAL

En esta sección, se explicará con detenimiento, la forma como se implementó cada una de las funciones de los bloques que constituyen este proyecto, es decir, las etapas digital y analógica, así como las salidas, interfaz gráfica, aplicación de control y algoritmo que se decidió desarrollar, de los descritos en el capítulo 2. El algoritmo escogido definitivamente fue el de sobremuestreo, debido a la necesidad de obtener la resolución de 12 bits, a partir de un convertidor analógico digital de 10 bits. El diagrama de bloques final, del algoritmo utilizado, es el mostrado en la figura 4.2, y se describe en la etapa de procesamiento digital.

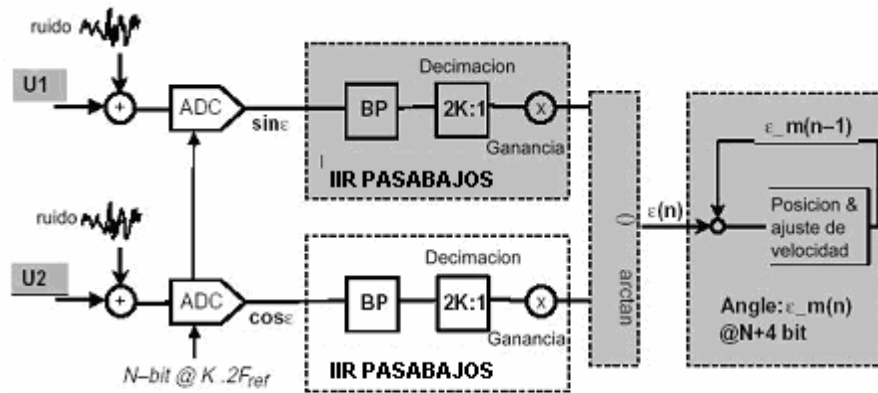


Figura 4.2. Diagrama en bloques definitivo del algoritmo implementado

4.2. ETAPA DE PROCESAMIENTO ANALÓGICO

El desarrollo de las señales analógicas que necesita el sistema de medición para su completo funcionamiento, se destaca a continuación.

4.2.1. **Filtro:** La señal cuadrada de 400 Hz, generada por el microcontrolador de la forma explicada en la etapa de procesamiento digital, puede ser filtrada a través de un filtro pasabanda o un pasabajos, como el utilizado finalmente con la configuración mostrada en la figura 4.3.

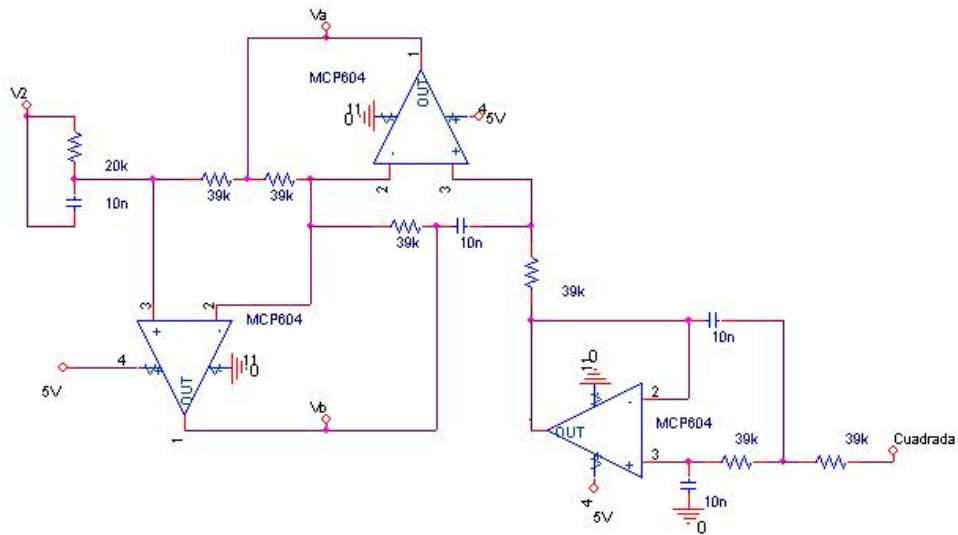


Figura 4.3. Filtro Pasabajos para la señal cuadrada de 400Hz

Este filtro se diseñó con las configuraciones activas *SallenKey* y *Antoniou* de segundo orden (cada uno) en cascada, hasta obtener un ancho de banda de 600 Hz.

La opción de un filtro activo pasabanda perdió fuerza debido a que las variaciones registradas con cambios de temperatura son mucho mayores a las de un filtro pasabajos, ya que en el primero, la banda de paso es muy selectiva a diferencia del segundo, usado finalmente. La comparación se realizó experimentalmente, mediante el montaje físico de cada uno de los filtros (pasabajos y pasabanda) y su calentamiento.

La señal seno entregada por el filtro se toma entre V_a y V_b ($V_a - V_b$) y ésta es la señal de referencia ($R1 - R2$, ver figura 4.1) con la que se alimentará el synchro. Aplicando esta señal como referencia, a una frecuencia de 400 Hz, se obtienen las tres señales seno desfasadas, $S1$, $S2$ y $S3$, del synchro. (Ver figura 4.1)

4.2.2. Circuito transformador de Scott: Para realizar la conversión de señales de synchro (desfasadas 120°) a señales seno y coseno, se implementó la función que cumple la manipulación trigonométrica descrita a continuación, mediante amplificadores operacionales y que generalmente se conoce como Circuito Transformador de Scott. Esta configuración se muestra en la figura 4.4.

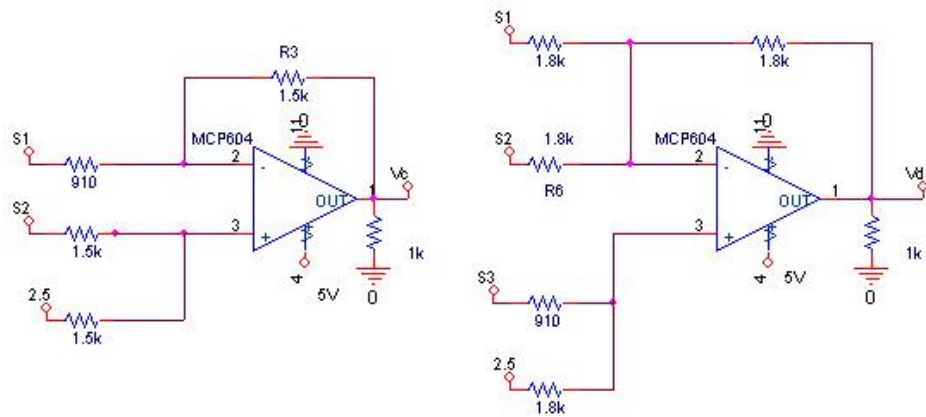


Figura 4.4. Circuito transformador de Scott.

Las señales del Synchro, mostradas en la figura 4.5, pueden escribirse de la siguiente forma:

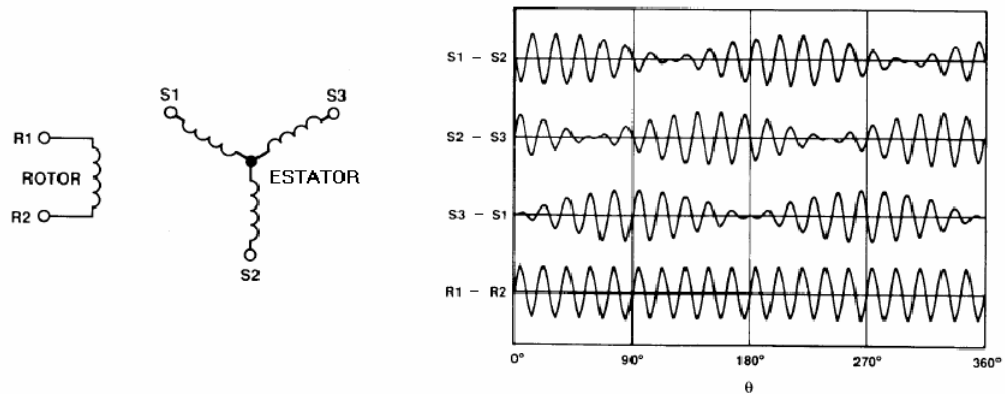


Figura 4.5. Señales entregadas por el synchro.

$$V_{S1-S2} = KV_{ref}(\text{Sen } \theta) = V_c$$

$$V_{S2-S3} = KV_{ref} (\text{Sen } (\theta + 120))$$

$$V_{S3-S1} = KV_{ref} (\text{Sen } (\theta + 240))$$

$$V_{ref} = V_a - V_b = V_o(\text{Sen } (wt))$$

Donde K es la constante de transformación del transductor del eje del ángulo. $V_{ref} = V_a - V_b$ es la señal de excitación del synchro y está ajustada a la frecuencia de funcionamiento del mismo (400Hz). Para la conversión, el primer operacional actúa como un buffer diferencial que resta las señales de S1 y S2, produciendo la señal seno de cuadratura

(Vc). La señal coseno se consigue, mediante la siguiente combinación lineal de las señales del synchro:

$$V_d = V_{S3-S1} - V_{S2-S3} = KV \text{ Sen}(wt)(\text{Sen} [\theta - 120^\circ] - \text{Sen} [\theta + 120^\circ])$$

Usando la identidad trigonométrica

$$\text{sen} (A + B) = \text{sen} A \cos B + \cos A \text{sen} B$$

Es entonces posible mostrar que

$$V_{S3-S1} - V_{S2-S3} = -KV \sqrt{3} \text{ Sen}(wt) \text{Cos} \theta = V_d$$

Como el amplificador invierte y realiza la diferencia entre las respectivas señales del synchro, la ganancia del amplificador se debe ajustar para adaptar la salida y así obtener las respectivas salidas seno y coseno con la misma amplitud.

Esta es la parte encargada de pasar las señales analógicas de salida del synchro, a las funciones seno y coseno del ángulo en formato digital. Las tres señales del synchro corresponden al seno de la posición angular medida y desfasada 120° y 240°.

4.2.3. **Manejador de RS-232** : Para comunicar la interfaz gráfica, desde la cual se realizan las mediciones del sistema, y el control de la aplicación mecánica, se utilizó una transmisión serial de datos mediante protocolo Modbus ASCII. Para tal objetivo, se escogió el estándar de comunicaciones RS-232.

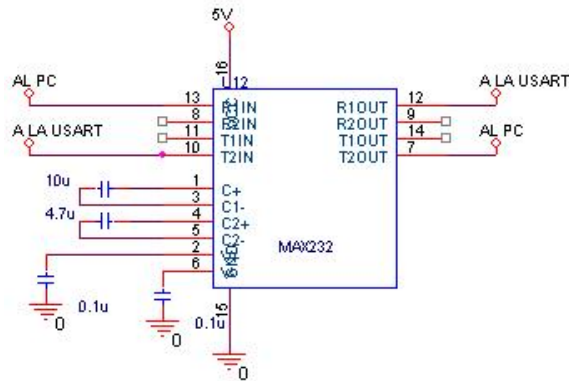


Figura 4.6. Interconexión para el manejador de RS-232

Con el fin de tener compatibilidad entre los niveles de voltaje con que operan el sistema y el computador, fue necesario usar un manejador de dicho estándar como lo es el circuito integrado MAX232 que se conectó directamente a los terminales de comunicación serial (USART) del microcontrolador como se observa en la figura 4.6.

4.3. ETAPA DE PROCESAMIENTO DIGITAL

Cada una de las partes que conforman la etapa de procesamiento digital, se implementaron en el microcontrolador PIC18F452, bajo el compilador de lenguaje C. El ANEXO E contiene el código fuente para cada una de las partes descritas a continuación. Este microcontrolador fue escogido, debido a sus características, comentadas en el capítulo 2, como velocidad de procesamiento, conversor analógico digital de 10 bits con al menos dos canales, PWM interno, puertos de entrada – salida, manejo de punto flotante, capacidad de memoria y demás requisitos necesarios para el proyecto.

Como puede apreciarse en la figura 4.7, el programa principal tiene como función inicializar las variables que se manipularán durante el desarrollo del algoritmo y las diferentes etapas; configurar los puertos a utilizar como salidas, así como los Timer que realizarán las interrupciones; configurar la USART para la transmisión y recepción de información del protocolo Modbus y finalmente esperar, constantemente, alguna de las interrupciones de los Timer que se han establecido para las diferentes funciones.

4.3.1. **Cuadrada 400Hz:** Para generar la señal de referencia de 400Hz, se utilizó el puerto RD7 del Microcontrolador, para generar una señal cuadrada entre 0V y 5V interrumpiéndolo cada 1.25 ms, con un vector de interrupción de máxima prioridad.

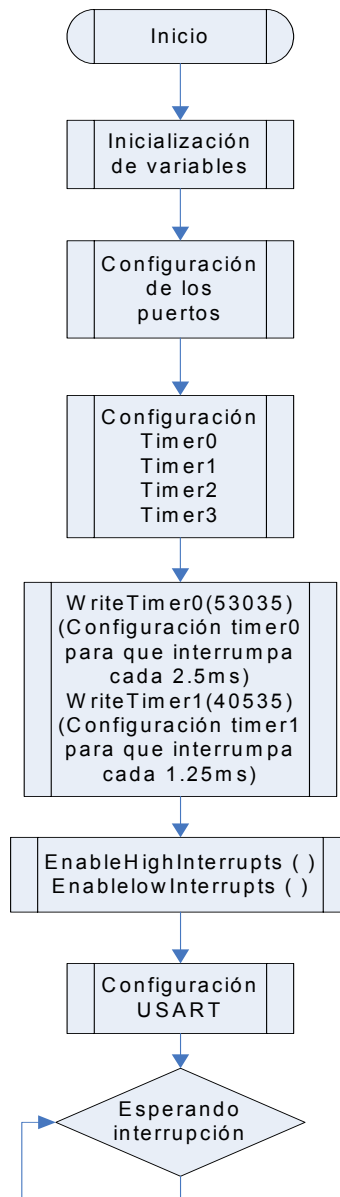


Figura 4.7. Diagrama de flujo del programa principal

4.3.1.1. **Interrupción Timer 1:** El Timer 1 interrumpe al microcontrolador cada 2.5 ms para realizar el procesamiento digital de las señales seno y coseno (algoritmo

de sobremuestreo). La figura 4.8. describe el funcionamiento de la interrupción por Timer 1.

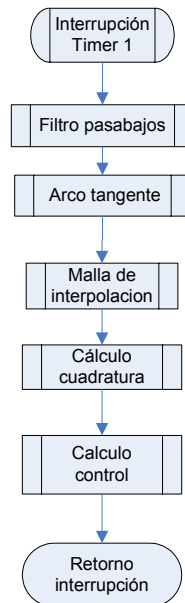


Figura 4.8. Diagrama de flujo de la interrupción del Timer 1

4.3.2. **Control:** El controlador de la aplicación, con el cual se corrobora el funcionamiento del sistema, fue implementado en el microcontrolador de acuerdo al diagrama mostrado en la figura 4.9. El nivel de referencia de la malla de control es ajustado desde el computador y se realiza la resta de dicho nivel, con el dato medido por el sistema, para obtener el error angular a controlar. A dicho error se le realiza una acción de control, cuyo resultado es adaptado a un valor entre 0 y 1023, proporcional al ciclo útil de la salida del PWM.

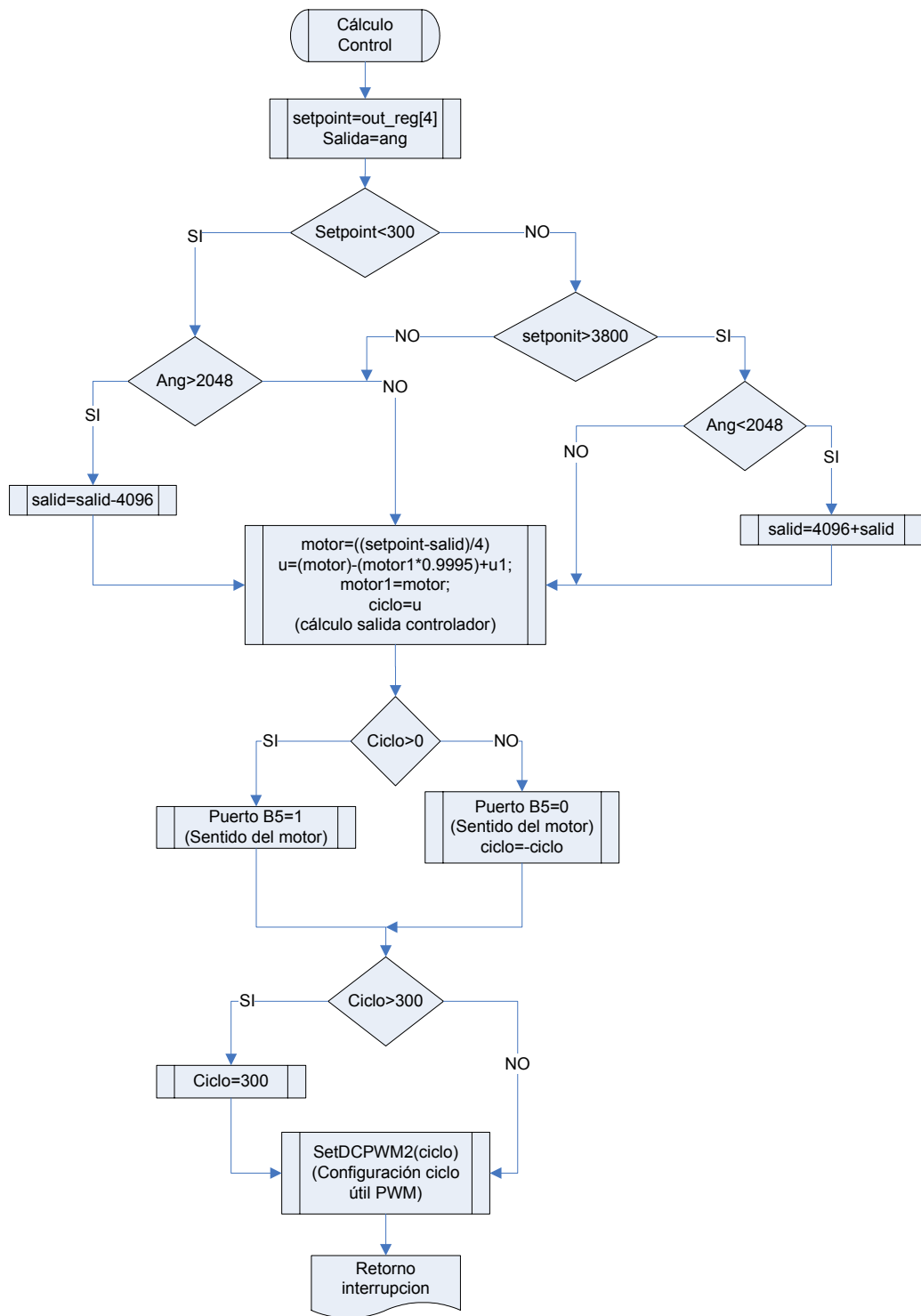


Figura 4.9. Diagrama de flujo de la etapa de control.

Existe una bandera de dirección que es activada dependiendo del signo del error. Esta bandera es asignada al puerto RB5 del microcontrolador, conectado al pin de dirección del puente H para proporcionar el sentido de giro al motor.

4.3.3. **Algoritmo de sobremuestreo:** Para calcular el algoritmo de sobremuestreo, sobre la señal seno y coseno, se sigue el procedimiento mostrado en la figura 4.8. Con el fin de completar el cálculo del algoritmo es necesario generar la señal cuadrada de 400 Hz con la que se obtendrá la señal de referencia del synchro y finalmente, realizar el muestreo de las señales entregadas por éste. Ésta es la función del Timer 0, descrita en la figura 4.10.

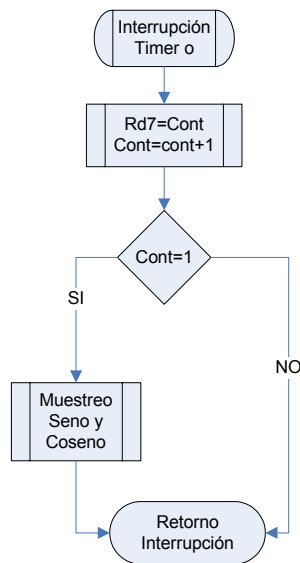


Figura 4.10. Diagrama de la interrupción del Timer 0.

4.3.3.1. **Interrupción Timer 0:** El Timer 0 se configuró a una frecuencia de 800Hz. Éste interrumpe dos veces cada 2.5 ms. En la primera interrupción se genera un voltaje alto (5V) en el puerto RD7 – terminal RD7. Para la segunda interrupción es un voltaje bajo (0V). De esta forma se genera la señal cuadrada de 400Hz, con un ciclo útil de 50%.

4.3.3.2. **Filtro Pasabajos:** Se seleccionó un filtro FIR, por ser un filtro digital anti-aliasing que además incrementa la resolución en 0.5 bits (3dB) cada vez que se disminuye el ancho de banda a la mitad, en este caso, de f_{ref} a $f_{ref}/2$, en

donde f_{ref} es 400 Hz, que es la frecuencia de referencia a la que funciona el synchro. Para el orden del filtro se debe tener en cuenta la decimación y el group delay, ya que cuando el orden es igual a $(4k+1)$, con k igual al número de veces de sobremuestreo, el group delay es igual a un período de la frecuencia de referencia. La selección final de un filtro pasabajos, reemplazando al FIR pasabanda del algoritmo original, se describe a continuación.

4.3.3.2.1. **Implementación del filtro FIR pasabanda en MATLAB:** Se escogió el orden del filtro igual a 21 con el fin de obtener una mayor selectividad en frecuencia (es decir, a mayor frecuencia, más selectivo es el filtro) y para poder reducir correctamente el ancho de banda a $\pm f_{ref}/2$ debido a que si se utilizaba un orden menor, no se llegaba al resultado esperado. Con este orden resulta que $k=5$, y por ende, la frecuencia de muestreo $F_{muestreo}=2(K)(F_{ref})$ con el fin de cumplir con el algoritmo de sobremuestreo. La frecuencia de muestreo es, entonces, de 4kHz.

Para generar los veintiún coeficientes del filtro, como se muestra en la figura 4.11, y simular su resultado, se utilizó MATLAB como herramienta de diseño digital, a través de la función **FIR1**, por el método de **Kaiser**, que devuelve dichos coeficientes ($b_0, b_1, b_2 \dots b_{20}$). Se usó una frecuencia de muestreo de 4kHz, la función y el método mencionados, debido a que ofrecieron el mejor resultado con respecto a selectividad en frecuencia, comparado con los métodos de **Chebwin** y **Win** disponibles para la misma función.

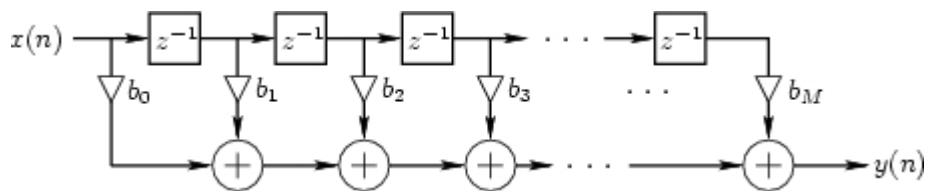


Figura 4.11. Esquema básico del filtro FIR.

Los coeficientes obtenidos en punto flotante, son los siguientes:

[0.0584 0.0543 0.0234 -0.0260 -0.0743 -0.0988 -0.0848 -0.0339
 0.0350 0.0934 0.1161 0.0934 0.0350 -0.0339 -0.0848 -0.0988
 -0.0743 -0.0260 0.0234 0.0543 0.0584]

Para adaptarlos al procesamiento digital, se escogió el máximo como 127 ya que en el microcontrolador la implementación del filtro es en punto fijo para disminuir el tiempo de procesamiento. Los nuevos coeficientes son los siguientes:

[64 59 26 -28 -81 -108 -93 -37 38 102 127 102 38 -37 -93 -108 -81 -28 26
 59 64]

Mediante las funciones *Freqz* y *Plot*, se obtuvo la respuesta ideal mostrada en la figura 4.12.

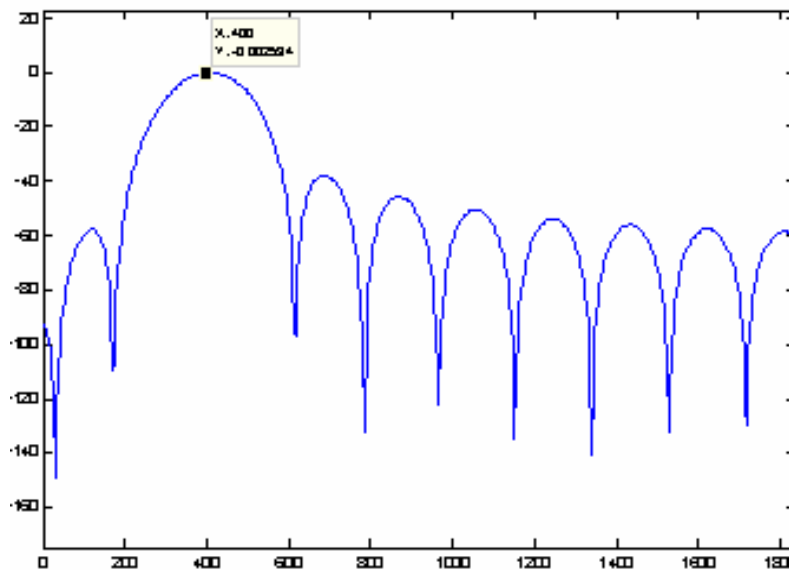


Figura 4.12. Simulación de la respuesta en frecuencia para el filtro FIR (Magnitud [dB] vs Frecuencia [Hz])

4.3.3.2.2. Implementación del filtro FIR pasabanda en SIMULINK: Se llevó a cabo la implementación en bloques del filtro FIR por medio de simulink, como se muestra en la figura 4.13, utilizando un tiempo de muestreo de 1/400Hz, debido a la decimación, que es equivalente a tomar la 2K-ésima muestra y filtrarla, es decir, tomar la décima muestra a una frecuencia de 4KHz. Con este diagrama se obtuvo la respuesta a entrada paso, mostrada en la figura 4.14,

utilizando punto fijo de 32 bits (*int32*), tipo de dato implementado en el microcontrolador.

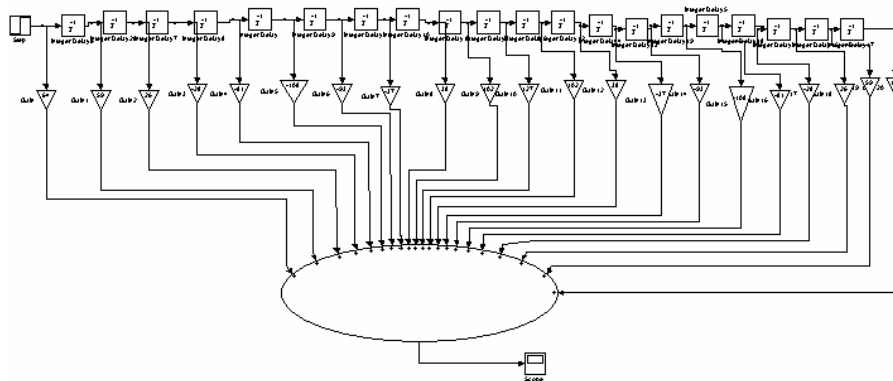


Figura 4.13. Diagrama en bloques del filtro FIR pasabanda en SIMULINK.

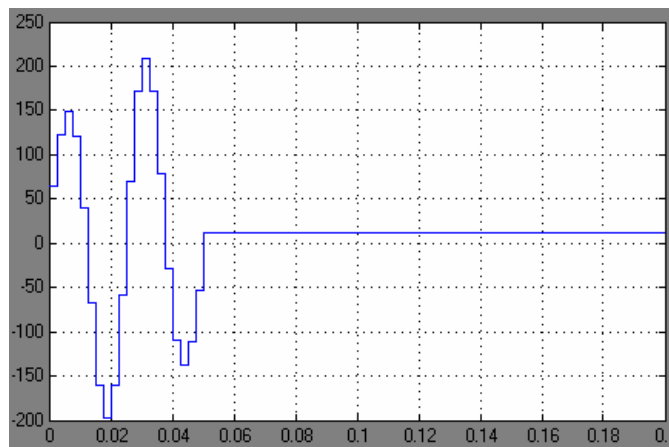


Figura 4.14. Respuesta a entrada paso para el filtro FIR, pasabanda, mediante simulación en SIMULINK.
(Amplitud vs Tiempo [s])

En el microcontrolador se desarrolló el diagrama en bloques de la figura 25 y se obtuvo respuesta a entrada paso de la figura 4.15.

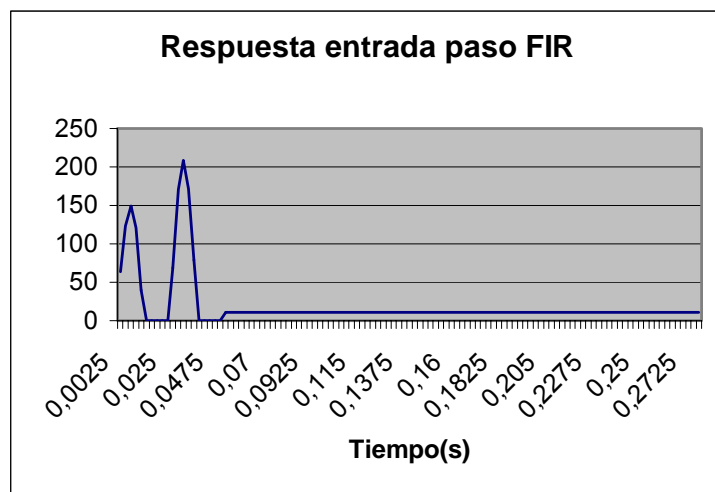


Figura 4.15. Respuesta real a entrada paso, del microcontrolador, para el filtro FIR.

Esta respuesta oscilatoria que presenta el filtro, altera el funcionamiento de la medida, debido a que los esfuerzos por mejorarla, deterioraban el requerimiento de reducción del ancho de banda a $\pm f_{ref}/2$.

Una solución para disminuir el efecto de la respuesta a entrada paso del filtro FIR, fue implementar uno pasabajos antes del mismo. Esta idea eleva el tiempo de procesamiento del algoritmo ya que el orden del filtro pasa bajos también debía ser alto.

La función del el filtro pasabanda FIR era la de aumentar la resolución en 0.5bits y filtrar el ruido, por esta razón se optó por implementar un filtro pasabajos que lo reemplazara, sin importar que la resolución no se aumentara en el valor ya mencionado, pues la malla de interpolación es adecuada para compensar estos efectos. Este filtro pasabajos tenía dos alternativas: IIR o FIR, con una frecuencia de corte suficientemente baja, para poder hacer una atenuación exitosa del ruido. Se experimentó con varios filtros FIR de distinto orden y frecuencia de corte bajo, hasta llegar al de mejor desempeño según las características deseadas. Con los filtro IIR surgía un problema debido a su fase no lineal, a diferencia de los FIR. No obstante se diseñaron, ya que no afectan el desempeño del algoritmo, pues su variación de fase es la misma para el seno y el coseno, que a su vez tienen la misma frecuencia. Se usaron dos filtros idénticos para la señal seno y coseno respectivamente, luego a dicha frecuencia, desfasan lo mismo. Los filtros IIR tienen una gran ventaja frente a los FIR y es que, con un menor orden, se puede

alcanzar la misma respuesta de un FIR de orden mucho mayor, sin contar que su implementación en el microcontrolador puede ser llevada a cabo en punto flotante, al contrario de la del FIR, que debe ser realizada en punto fijo, por el alto grado de procesamiento requerido, ya que el filtro FIR simulado era de orden 21 lo que hace necesario definir 42 variables dentro del microcontrolador (21 por filtro) haciendo imposible su implementación en el microcontrolador escogido.

4.3.3.2.3. **Implementación del filtro FIR pasabajos en MATLAB:** Se utilizó de nuevo la función *fir1* para generar los coeficientes correspondientes para la simulación en tiempo y frecuencia. Se especificó una frecuencia de corte de 2Hz, con la cual, ante una variación de 2 bits en la medida del ADC, se atenúa el ruido hasta obtener una variación de 1 bit, como se puede ver en la figura 4.25., todo esto teniendo en cuenta un tiempo de muestreo de 4kHz y orden 21 con el fin de ser coherentes con lo especificado en el filtro FIR pasabanda.

Se utilizó el método de *Win* con el cual se obtuvo la mejor respuesta. Los coeficientes obtenidos fueron los siguientes y la respuesta en frecuencia para este diseño se detalla en la figura 4.16.

[8 10 15 24 35 48 62 74 84 92 96 96 92 84 74 62 48 35 24 15 10
8]

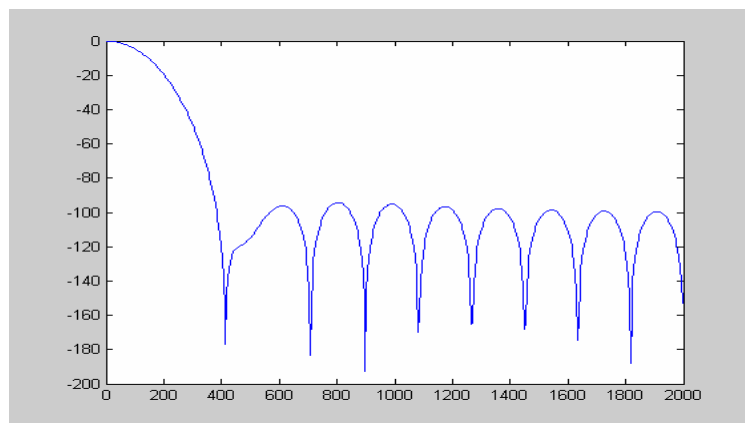


Figura 4.16. Respuesta en frecuencia para el filtro FIR pasabajos (Magnitud [dB] vs Frecuencia [Hz]).

4.3.3.2.4. **Implementación del filtro FIR pasabanda en SIMULINK:** La simulación en tiempo se llevó a cabo en SIMULINK, con el diagrama en bloques mostrado en la figura 4.17, para el cual se utilizó el *toolbox* de *Signal Processing Blockset*. A esta simulación se le añadió ruido para observar el comportamiento con respecto a este. En la figura 4.18 se observa el resultado ante una entrada paso.

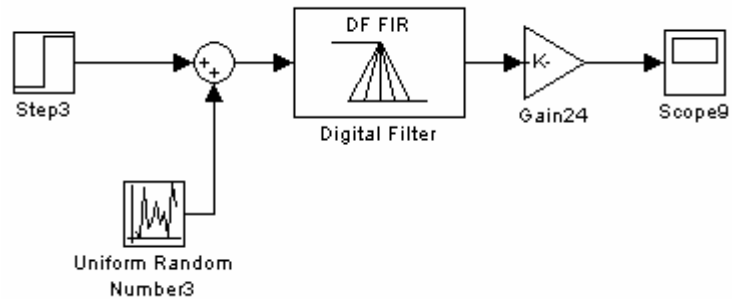


Figura 4.17. Diagrama en bloques del filtro FIR pasabajos en SIMULINK.

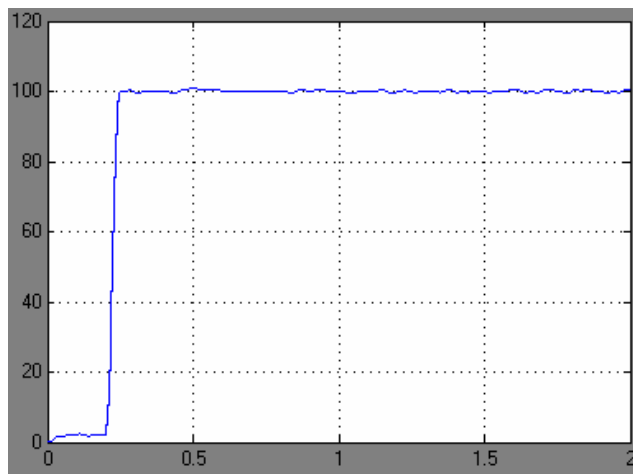


Figura 4.18. Respuesta a entrada paso para el filtro FIR, pasabanda, mediante simulación en SIMULINK.
(Amplitud vs Tiempo [s])

4.3.3.2.5. **Implementación del filtro IIR pasabajos en MATLAB:** Para implementar un filtro pasabajos IIR, que tuviera una frecuencia de corte de 2Hz, se utilizó nuevamente MATLAB. Después de observar la respuesta a entrada paso y con respecto al ruido, se llegó a que con un orden de cuatro (4) se obtenía una

respuesta similar a la del FIR de orden 21, en los dos aspectos antes citados, como se observa en las figuras 4.24 y 4.25.

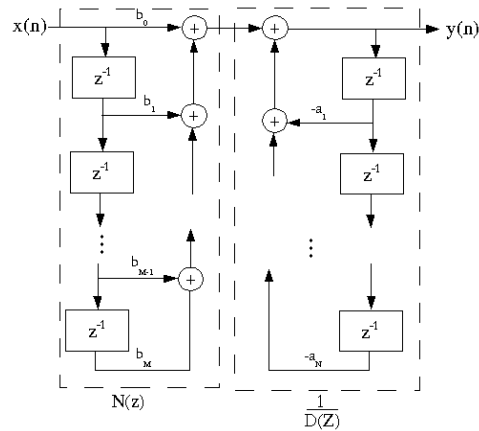


Figura 4.19. Esquema básico de un filtro IIR

Utilizando la función **Yulewalk** se desarrolló el filtro, ajustando la respuesta en frecuencia con un corte de 2Hz y un tiempo de muestreo de 4KHz, para el que se obtuvieron los siguientes coeficientes:

$$A=1.0000 \quad -3.3265 \quad 4.2475 \quad -2.4608 \quad 0.5443$$

$$B= 0.0031 \quad -0.0038 \quad 0.0041 \quad -0.0015 \quad 0.0014$$

La respuesta en frecuencia para este diseño se detalla en la figura 4.20.

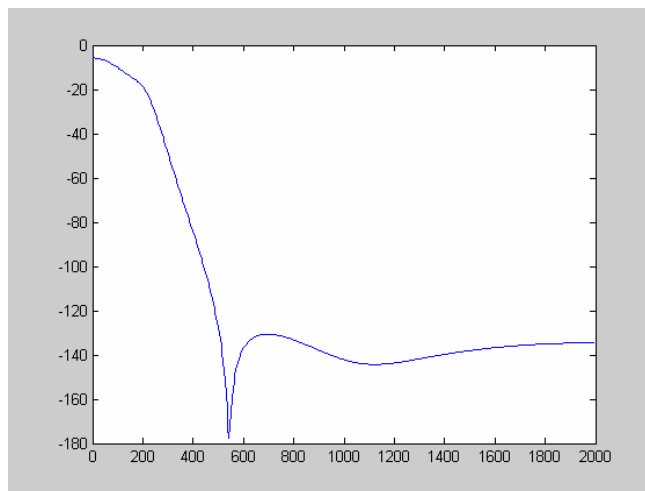


Figura 4.20. Respuesta en frecuencia para el filtro IIR pasabajos (Magnitud [dB] vs Frecuencia [Hz]).

4.3.3.2.6. **Implementación del filtro IIR pasabajos en SIMULINK:** La simulación en tiempo se llevó a cabo en SIMULINK, con el diagrama en bloques mostrado en la figura 4.21. A esta simulación se le añadió ruido para observar el comportamiento con respecto a este. En la figura 4.22 se observa el resultado para una entrada paso.

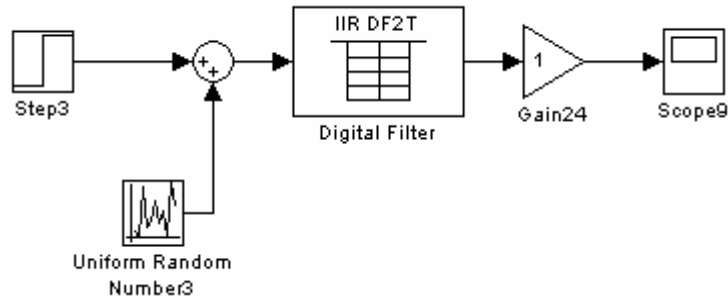


Figura 4.21. Diagrama en bloques del filtro IIR pasabajos en SIMULINK.

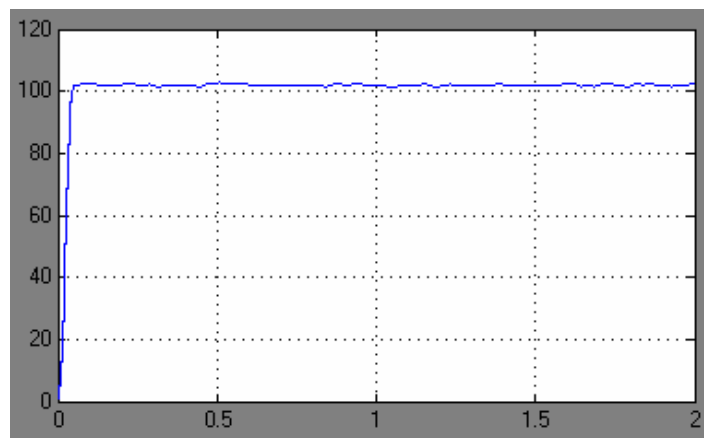


Figura 4.22. Respuesta a entrada paso para el filtro IIR, pasabajos, mediante simulación en SIMULINK (Amplitud vs Tiempo [s]).

4.3.3.2.7. **Comparación entre los filtros FIR e IIR pasabajos:** La respuesta obtenida para ambos filtros fue muy similar, así que se decidió compararlos mediante el esquema de la figura 4.23.

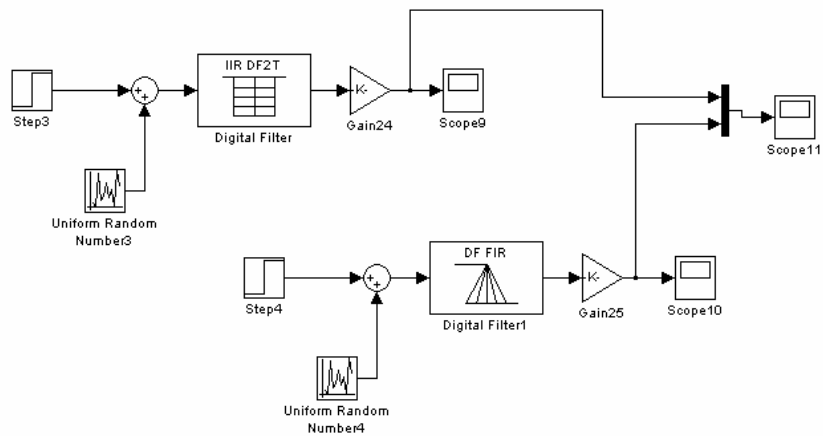


Figura 4.23. Esquema de comparación para los filtros FIR e IIR pasabajos.

Para este diagrama se obtuvo la respuesta a entrada paso, agregando ruido a dicha entrada, con los resultados mostrados en la figura 4.24.

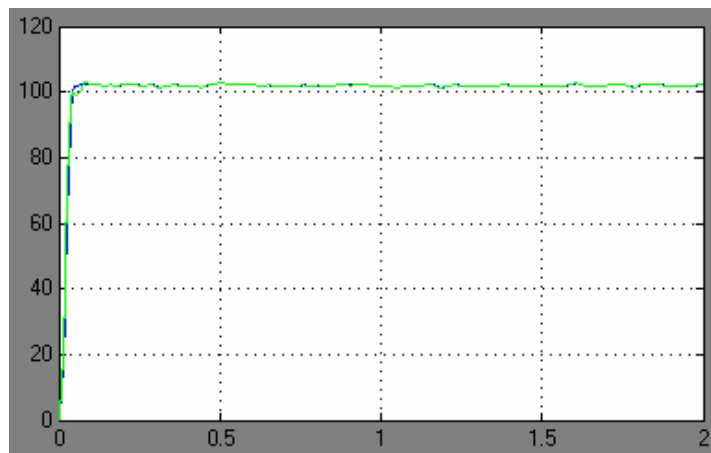


Figura 4.24. Comparación de respuestas a entrada paso, para filtros IIR y FIR pasabajos (Amplitud vs Tiempo [s]).

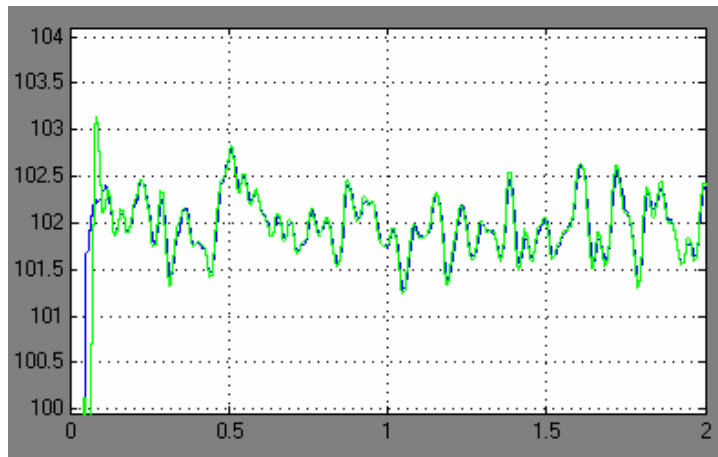


Figura 4.25. Comportamiento con respecto al ruido, ante una entrada paso, para filtros IIR y FIR pasabajos. (Amplitud vs Tiempo [s]).

Las gráficas verdes, de las figuras 4.24 y 4.25, son las respuestas de filtro FIR y las azules corresponden a la del IIR. El comportamiento con respecto al ruido se puede apreciar en la figura 4.25. Como los resultados son casi idénticos, con respecto a respuesta a entrada paso y comportamiento al ruido, se decidió implementar definitivamente el filtro IIR ya que su orden es de 4, mientras que el del FIR es de 21. Por esta razón, el filtro IIR puede realizarse en punto flotante, contrario al FIR que debe hacerse en punto fijo.

4.3.3.3. **Función arcotangente:** Para la realización de la función arcotangente se tuvieron varios criterios de selección del algoritmo a utilizar, dentro de los cuales el más importante es la precisión del resultado. Seguidamente se tuvo en cuenta un algoritmo, desarrollado en punto fijo, pero los encontrados sólo funcionaban para ángulos pequeños y su resolución era muy baja, así que se optó por un algoritmo en punto flotante, basado en iteraciones (Ver ANEXO E). La función arcotangente desempeña un papel muy importante en el sistema en general, debido a que el desarrollo del algoritmo de sobremuestreo, está hecho en punto flotante, lo que satisface las necesidades de precisión y rango de ángulos, deseadas.

4.3.3.4. **Malla cerrada de interpolación :** La malla cerrada descrita por bloques, mostrada en la figura 4.26, fue implementada en SIMULINK (figura 4.27) para

analizar su respuesta a entrada paso. Esta malla se realiza por la necesidad de mejorar la resolución, dado que el conversor analógico-digital del microcontrolador es de 10bits y la resolución propuesta por este proyecto, de 12bits. Su realización real en el microcontrolador, se basó en la malla descrita en la nota de aplicación de *Texas Instruments*⁵, en la cual se garantiza un aumento hasta de 4 bits en la resolución. La malla propuesta tiene un rango variable de incremento de resolución, dependiendo de su ancho de banda, debido a que en conjunto la malla es un filtro pasa bajos.

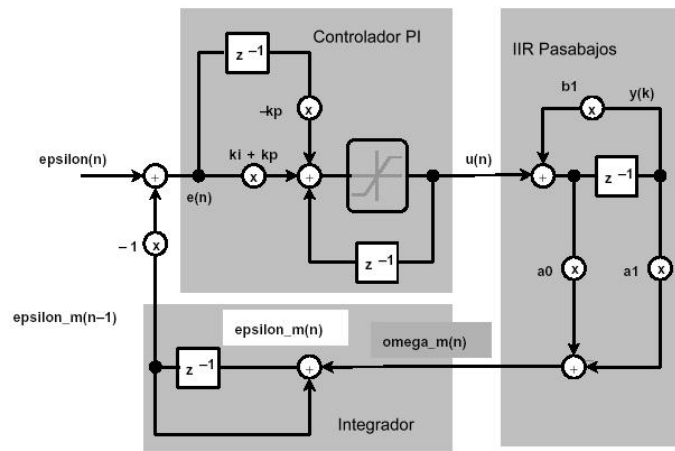


Figura 4.26. Esquema de la malla cerrada de interpolación y ajuste de velocidad

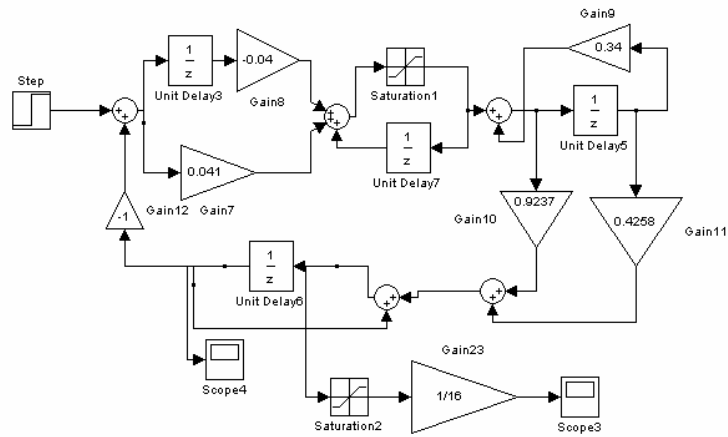


Figura 4.27. Malla cerrada de interpolación implementada en simulink.

⁵ TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed. Application Report. Texas Instruments. February 2000. <http://www.ti.com>

La malla real se realizó en punto flotante para obtener una resolución de 12 bits.. Esta malla consta de un controlador proporcional integral, un filtro IIR pasa bajos, y un integrador.

4.3.3.4.1 **Filtro IIR:** Para el desarrollo del filtro IIR, de forma digital, nuevamente se utilizó MATLAB, mediante el comando *Yulewalk*, para obtener los coeficientes pasándole como parámetros el orden y las frecuencias de corte y central.

El criterio para escoger la respuesta del filtro se basó en la resolución obtenida finalmente. Si el filtro se calcula con un frecuencia de corte muy baja, la resolución aumenta pero a la vez el tiempo de establecimiento crece mucho, así que se decidió no ponerle a éste, una frecuencia de corte muy baja. Con una frecuencia de 400Hz de muestreo y utilizando las funciones descritas se obtuvieron los siguientes coeficientes y la respuesta en frecuencia mostrada en la figura 4.28.

$$A= 1.0000 \quad 0.3400$$
$$B= 0.9237 \quad 0.4258$$

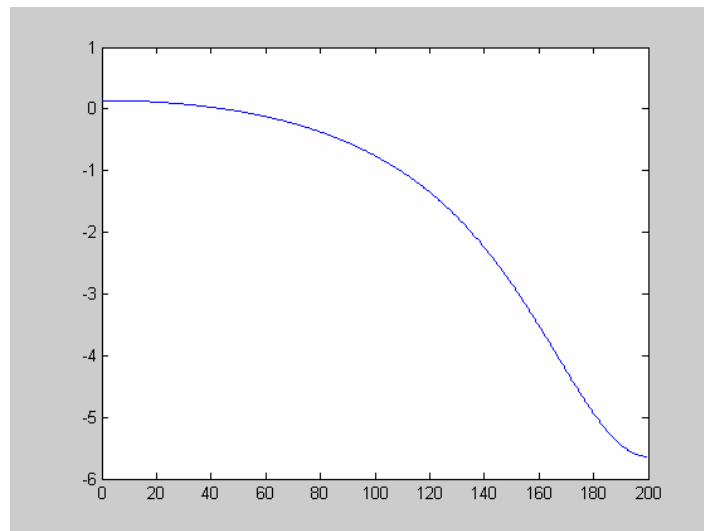


Figura 4.28. Respuesta en frecuencia para el filtro IIR pasabajos de la malla cerrada de interpolación (Magnitud [dB] vs Frecuencia [Hz]).

El filtro fue simulado en SIMULINK, mediante el diagrama de la figura 4.29, para obtener la respuesta a entrada paso, utilizando una frecuencia de muestreo de 400Hz y con los coeficientes anteriormente hallados. El cálculo se efectuó en punto flotante para ser

coherentes con lo implementado en el microcontrolador. El resultado se observa en la figura 4.30.

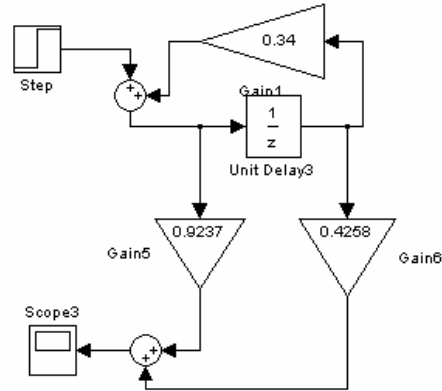


Figura 4.29. Diagrama en bloques del filtro IIR, en SIMULINK, de la malla cerrada de interpolación.

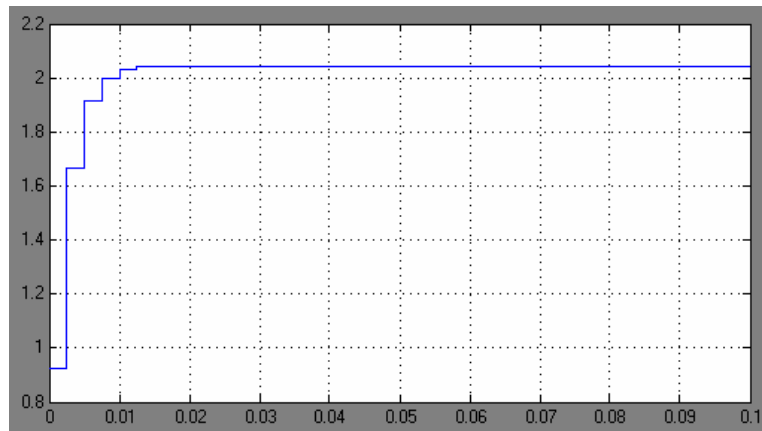


Figura 4.30. Respuesta a entrada paso para el filtro IIR, de la malla de interpolación, mediante simulación en SIMULINK (Amplitud vs Tiempo [s]).

Es importante tener en cuenta que el tiempo de establecimiento de este filtro no puede ser muy alto, pues su efecto repercute sobre el tiempo total de establecimiento de la malla.

4.3.3.4.2. **Controlador Proporcional-Integral de la malla de interpolación:** Para este desarrollo, se diseñó un controlador proporcional integral discreto, con la siguiente función de transferencia en el dominio de z:

$$\frac{(k_i + k_p - k_p Z^{-1})}{1 - Z^{-1}}$$

El controlador se implementó en SIMULINK, en conjunto con la malla de interpolación, con una frecuencia de muestreo de 400Hz en punto flotante.

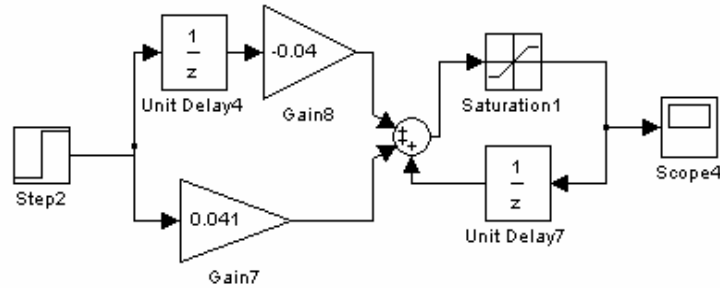


Figura 4.31. Diagrama en bloques del controlador P+I.

Los valores de las constantes finalmente utilizados fueron de $K_p=0.04$ y $K_i=0.001$. Estos valores se encontraron experimentalmente teniendo en cuenta el tiempo de establecimiento y resolución del sistema completo, ya que la sintonización de la malla depende estos dos valores y no hay un método matemático que permita hacer el cálculo de la resolución, en función de estos valores .

4.3.3.4.3. **Integrador:** Su función es compensar el error de velocidad, garantizando que sea igual a cero y no existan problemas de retardo ante una entrada rampa.

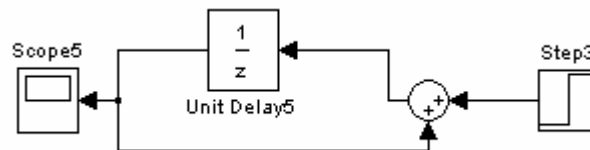


Figura 4.32. Diagrama en bloques del integrador de la malla de interpolación.

4.3.4. **Módulo MODBUS esclavo:** Este módulo, para efectos de la comunicación y el protocolo, es el microcontrolador (sistema de medición angular), en el cual se ajustó la rutina de atención para que se decodifique la trama enviada por el

módulo maestro (computador) y así reconstruir la trama con la respuesta a la función enviada.

4.3.4.1. **Interrupción por recepción en la transmisión serial:** Esta es la interrupción de mayor prioridad. Se realiza cuando el computador se comunica con el microcontrolador (módulo MODBUS esclavo). A través de ella se recibe el dato enviado por el Dispositivo Maestro del Protocolo Modbus ASCII (Computador), se procesa dicha información y se responde con la trama correspondiente a la función enviada, todo de acuerdo al protocolo. La figura 4.32 describe el funcionamiento de la interrupción por transmisión serial.

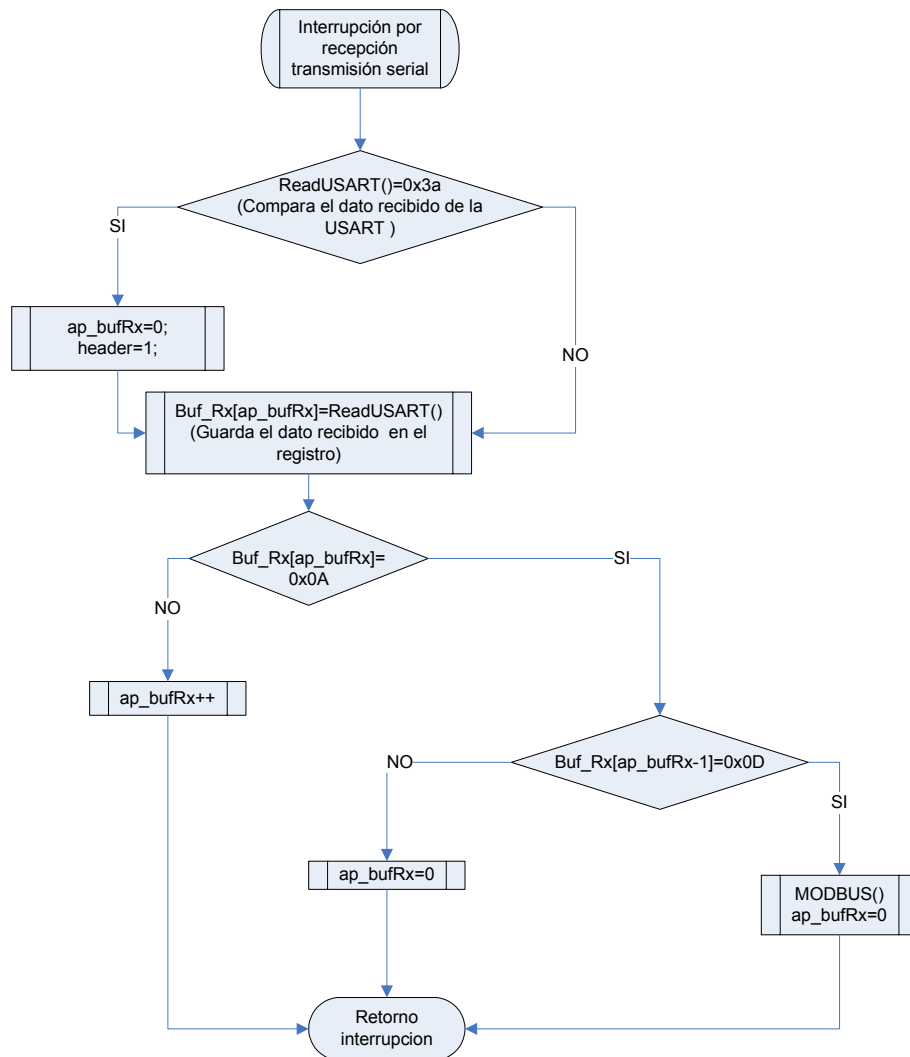


Figura 4.33. Diagrama de flujo de la interrupción por recepción en la transmisión serial.

4.3.5. **Generación de cuadratura:** Para obtener la salida en cuadratura incremental, fue necesaria la interrupción del programa principal, a través del Timer 3 y utilizando las salidas RD4, RD5 y RD6 para las señales que la conforman.

4.3.5.1. **Interrupción (Timer3):** La función de Timer 3 es generar la señal de salida en cuadratura incremental. Cada vez que se obtiene la medida del ángulo, se calcula el número de pulsos que deben generarse, equivalente a dicha medida.

Con estos pulsos se configura el Timer3 para que interrumpa 4 veces por el número de pulsos calculados. La rutina de atención a esta interrupción genera un voltaje alto (5V) o un bajo (0V) en el puerto D- pin RD5 y de igual forma en RD6. En el pin RD4 se genera la señal de índice, que se cuenta sólo cuando se complete una vuelta.

La señal del puerto RD5 está desfasada 90° con respecto a la de RD6. Para tal fin se interrumpe cuatro veces el número de pulsos calculados y se generan las señales de RD5 y RD6 a través de una máquina de estados. La figura 4.34 describe el funcionamiento de la interrupción por Timer 3.

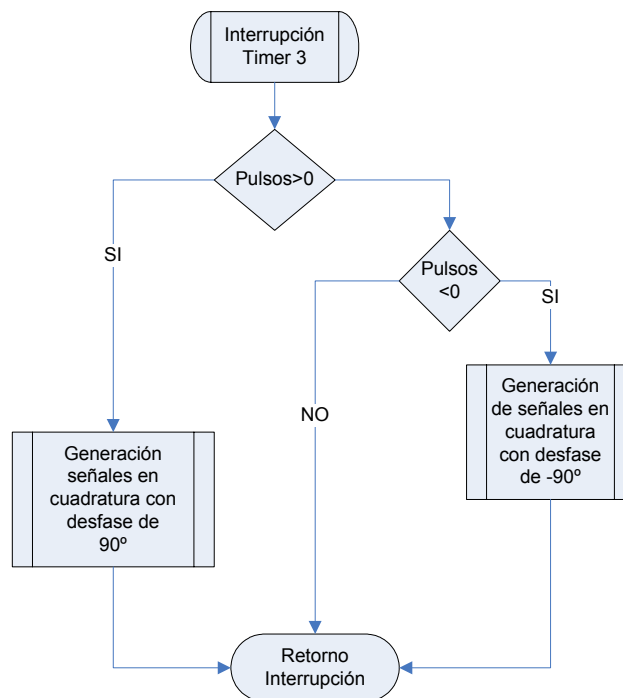


Figura 4.34. Diagrama de flujo por interrupción del Timer 3.

4.3.6. **DAC y PWM:** Las salidas del DAC y del PWM necesitan sólo la información digital de la medida del ángulo, en formato de 12 bits, para realizar su función, debido a que cada una de ellas se procesa internamente, de la siguiente forma: en el microcontrolador, para el caso del PWM y en el circuito integrado de conversión digital-analógica, para el caso de la salida de DAC. En este último se genera una salida constante entre 0 y 5V y para construirla se utilizó el esquema mostrado en la figura 4.35 de acuerdo, a su hoja de especificaciones (ver ANEXO B).

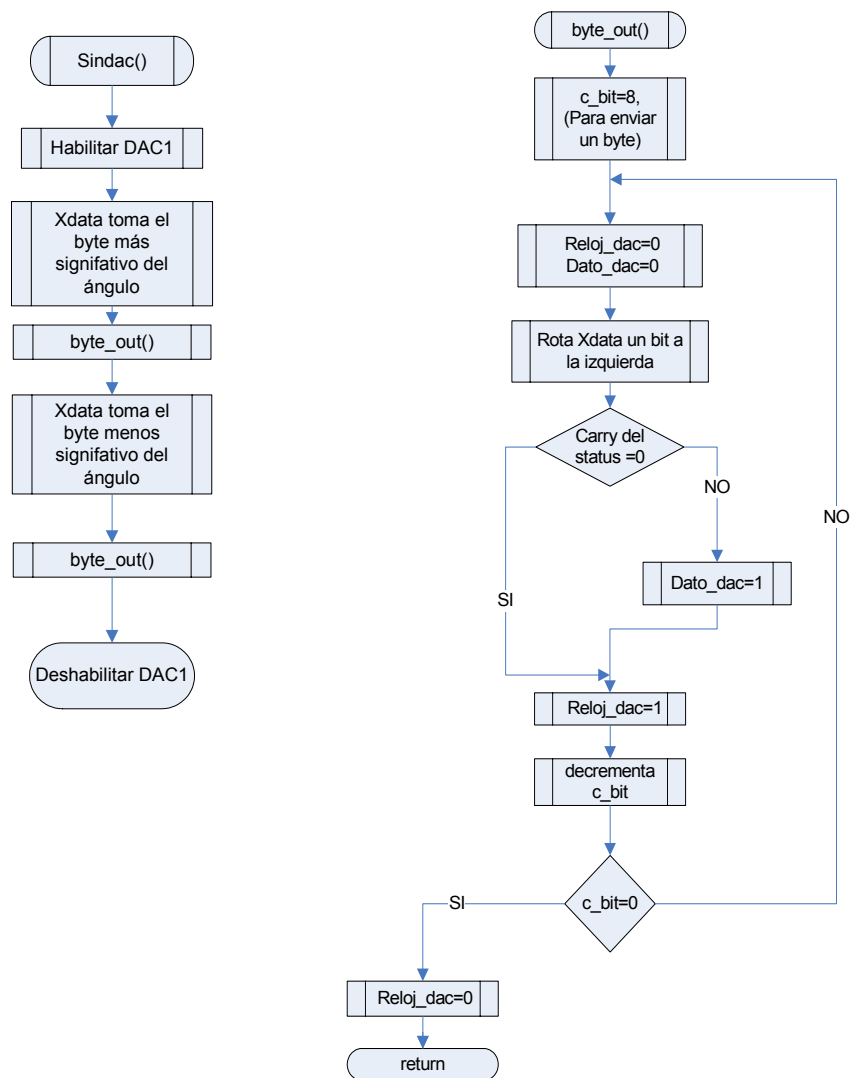


Figura 4.35. Diagrama de flujo de conversión de medida digital-analógica

Para generar la salida del PWM, se utilizó una función del compilador, configurada de acuerdo a los requerimientos de ciclo útil y frecuencia de su señal de salida.

4.4. ETAPA DE CONTROL

Con el fin de constatar el funcionamiento del Sistema de Medición, se realizó un montaje mecánico de dos grados de libertad, para el cual se acopló un synchro a un motor, por cada grado de libertad, con una relación de 1:1 (figura 4.36), para garantizar que el movimiento de los rotores de cada uno, sea idéntico.

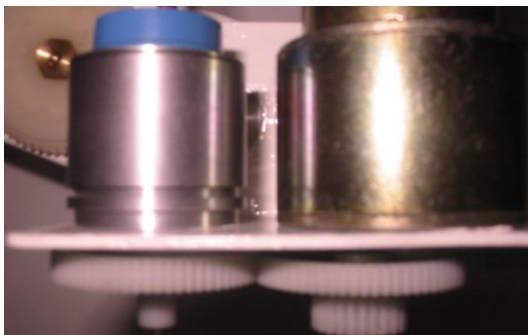


Figura 4.36.a.

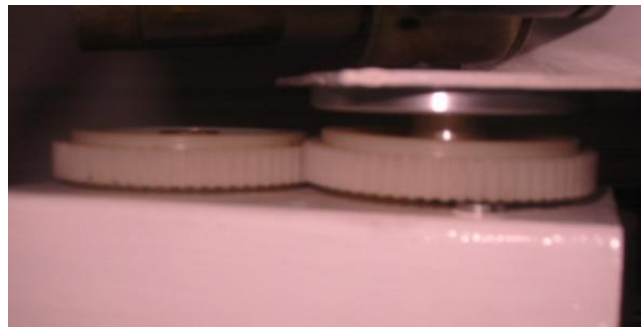


Figura 4.36.b.

Figura 4.36. Acople mecánico 1:1, entre los rotores del synchro y motor DC, para el grado de libertad del eje a) vertical y b) horizontal

Para controlar este mecanismo, se modeló el motor a utilizar (**9234 Pittman**), con base en sus características, de la siguiente forma: (figura 4.37)

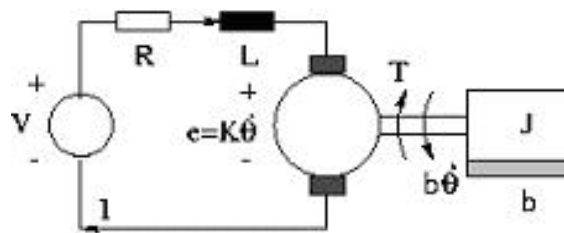


Figura 4.37. Modelo para un motor DC.

Donde:

- Momento de inercia del rotor: J [kg-m²/s²]
- Coeficiente de fricción: b = [N-s/m]
- Constante electromotriz: $K = K_e = K_t$ [N-m/A]
- Resistencia de armadura: R [Ω]
- Inductancia de armadura: L [H]

La función de transferencia obtenida es de la siguiente forma:

$$\frac{\theta(s)}{V(s)} = \frac{K}{s[(Js + b)(Ls + R) + K^2]}$$

Los valores de estas variables, para el motor real, de acuerdo a su hoja de especificaciones, son los siguientes:

$$J = 5.9 \times 10^{-4}$$

$$b = 2.6 \times 10^{-6}$$

$$K = 10.556$$

$$R = 10.556$$

$$L = 10 \times 10^{-3}$$

Con estos valores se pudo obtener una función de transferencia para el motor como la siguiente:

$$\frac{73.2e-3}{41.7e-9s^3 + 43.99e-6s^2 + 5.38e-3s}$$

Transfer Fcn2

Como la planta (motor) a controlar es analógica, se hizo el cálculo del controlador en frecuencia y se convirtió a un cálculo discreto, utilizando la función *c2d* de MATLAB por medio del método de *ZOH*, con una frecuencia de muestreo de 400Hz, para obtener el controlador discreto a programar en el microcontrolador. Primero se sintonizó la planta ,

hasta obtener una respuesta aceptable ante una entrada paso. El resultado de dicha sintonización se muestra en la figura 4.38.

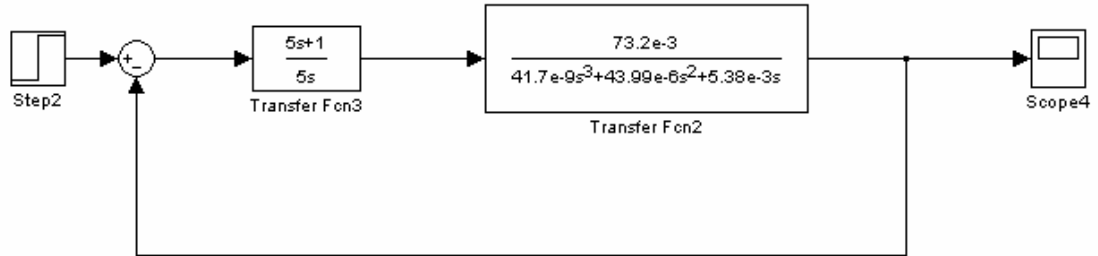


Figura 4.38. Diagrama del controlador para el motor DC.

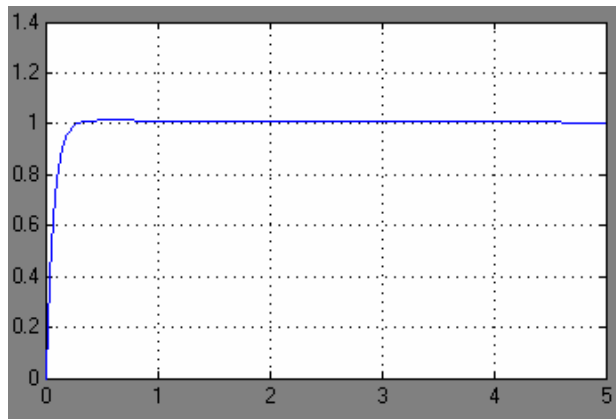


Figura 4.39. Respuesta a entrada paso para el controlador del motor DC (Amplitud vs Tiempo [s]).

La anterior respuesta se obtuvo con un $K_p=1$ y $\tau =5$, en el controlador analógico. Seguidamente se obtuvo la respuesta discreta de este controlador, con una frecuencia de muestreo de 400Hz. La salida del controlador final debe ser digital para ser implementada en el microcontrolador. El siguiente es el controlador obtenido en el dominio de z:

$$\frac{1-0.9995z^{-1}}{1-z^{-1}}$$

Discrete Filter

Como no es posible simular el funcionamiento del Synchro, la malla de control final fue simulada como se muestra en la figura 4.40, para poder conocer la respuesta del sistema a una entrada paso.

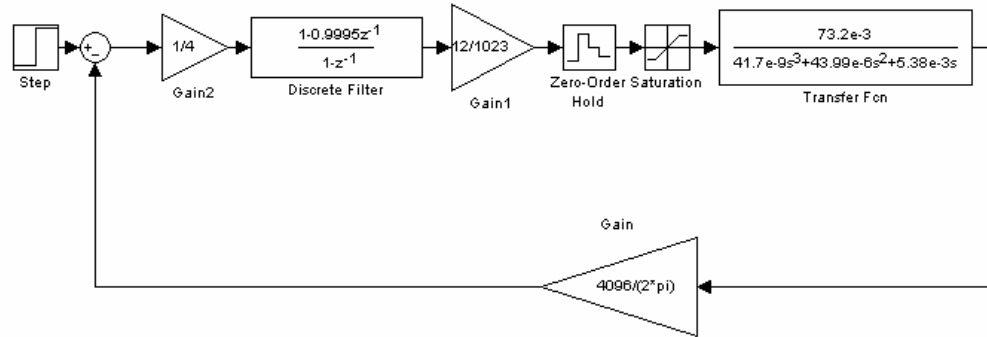


Figura 4.40. Esquema de simulación de respuesta final del controlador.

En anterior esquema se simuló a una frecuencia de muestreo de 400Hz, utilizando punto flotante ya que de esta forma se implementó en el microcontrolador y se obtuvo la respuesta a entrada paso mostrada en la figura 4.41.

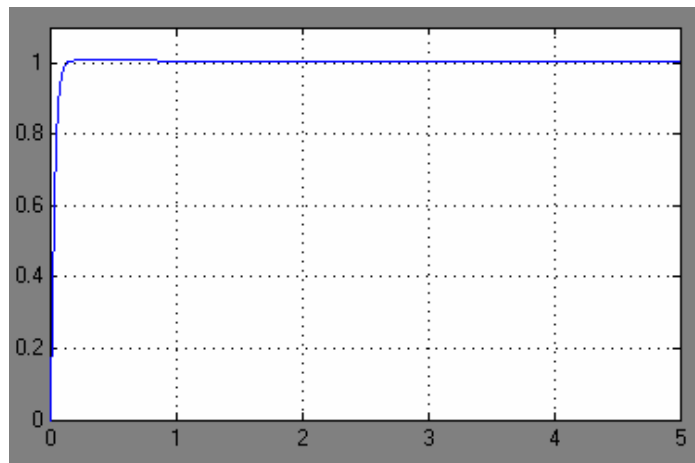


Figura 4.41. Respuesta a entrada paso para la malla definitiva de control (Amplitud vs Tiempo [s]).

4.5. ETAPA DE INTERFAZ GRÁFICA

La interfaz gráfica del sistema de medición fue creada con el fin de visualizar el resultado de la medida angular procesada digitalmente por el microcontrolador. Esta interfaz fue diseñada en lenguaje *Basic*, mediante el software *Visual Studio 6.0*. La información entre el sistema de medición y el computador es enviada por medio de la USART del microcontrolador.



Figura 4.42. Pantalla de Inicio de interfaz Gráfica

La interfaz gráfica, pretende hacer amigable la medición angular para el usuario, a través de una pantalla que cumple esta función, como es mostrada en la figura 4.43. En ella se visualiza la medida en dos formatos: de 0° a 360° y de 0-4096 (Medida angular de 12 bits).



Figura 4.43. Pantalla de medición de la interfaz gráfica.

En estas dos medidas se pueden apreciar los incrementos y las mínimas variaciones, entre una y otra posición de la aplicación mecánica. Esta pantalla cuenta además, con un botón de “iniciar/parar” medición para comenzar la comunicación de lectura entre los

módulos del protocolo Modbus. También unas casillas de dirección de registro a leer, cantidad de registros a leer e intervalo de activación entre lectura y lectura, que pueden ser configurados conociendo el protocolo de comunicación.

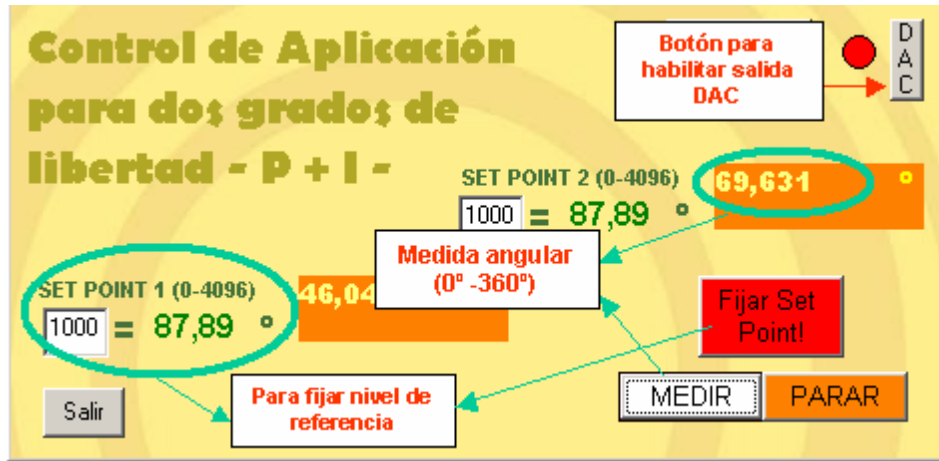


Figura 4.44. Pantalla de control de la aplicación mecánica.

Con el botón de controlar aplicación, se puede acceder a la pantalla que cumple la función de iniciar el control del montaje mecánico (figura 4.44), en los niveles de referencia que pueden establecerse, para cada grado de libertad, en esta pantalla. Adicionalmente, puede hacerse un seguimiento de la posición, oprimiendo el botón "medir", con el cual se realizará la lectura del registro donde se almacena la posición y será mostrado entre 0° y 360°, al lado del nivel de referencia establecido. En esta pantalla también se controla la salida del DAC. Mediante el botón ubicado en la esquina superior derecha, se puede habilitar o deshabilitar esta función del microcontrolador.

4.6. ETAPA DE APLICACIÓN

La aplicación, básicamente consta de un montaje mecánico de dos grados de libertad en el cual se acopló un motor DC a un Synchro y se conectaron a un sistema de medición angular microcontrolado, por cada eje de movimiento (vertical y horizontal), como se detalla en la figura 4.45. En el ANEXO A se encuentran las fotografías de este montaje desde otras perspectivas.

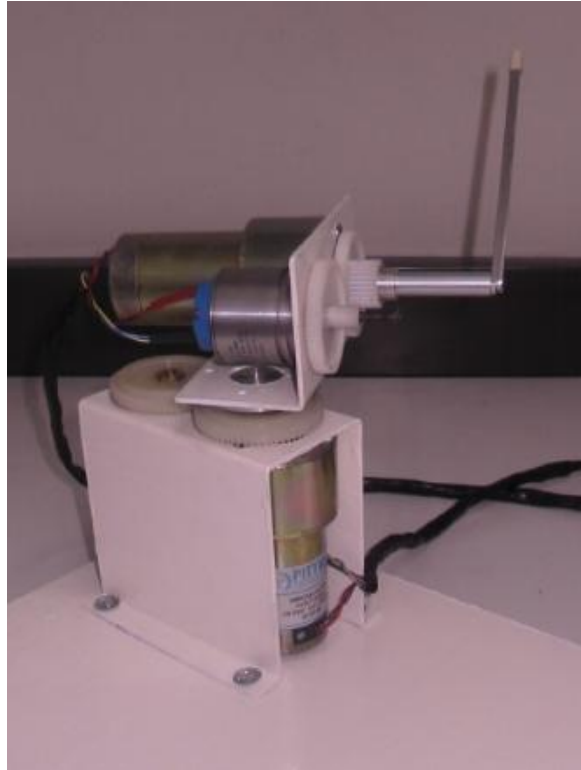


Figura 4.45. Montaje mecánico de dos grados de libertad, para aplicación de control de posición.

4.7. OTROS DESARROLLOS DEL PROYECTO

4.7.1. **Circuitos Impresos:** Para desarrollar el proyecto se utilizó el software Orcad 9.1, con el fin de diseñar una tarjeta analógica, encargada de procesar este tipo de señales, y otra digital en la que se ubicó el microcontrolador y las salidas que necesita este tipo señal. En los ANEXOS C y D se detallan los esquemáticos definitivos de las tarjetas usadas en este proyecto.

4.7.1.1. **Tarjeta Analógica:** Esta tarjeta se diseñó con el fin de efectuar la comunicación entre la tarjeta digital y los componentes necesarios para su funcionamiento, como es el caso de los filtros y el manejador de RS-232. Para su diseño se tuvo en cuenta:

- **Tamaño:** Las dimensiones no exceden las de la tarjeta digital, debido a que han de ubicarse una sobre la otra, requisito propuesto para este proyecto.
- **Conectores:** Varias señales entregadas por la tarjeta analógica, tienen relación directa con el procesamiento del microcontrolador. Por lo tanto se realizaron muchas

conexiones entre una tarjeta y otra, por lo que la distribución de los conectores se hizo de acuerdo con estas necesidades.

- Características eléctricas: Este es quizá el requisito más importante para el diseño de cualquier circuito impreso. Para él se tuvo en cuenta el ancho de las pistas que conducen tierra y fuente, destacando que el consumo de corriente del sistema es de 68mA, así como también la separación entre las pistas y la distribución de los componentes de acuerdo a la necesidad de cada bloque de operación (filtros, manejador, conectores, etc).
- Dependencia entre una tarjeta y otra: La fuente de alimentación de la tarjeta analógica, se toma de la tarjeta digital. Por lo tanto, ambas se encuentran eléctricamente acopladas.
- Circuitos Integrados: El filtro activo y el transformador de Scott, se diseñaron con el fin de trabajar con un valor positivo de fuente (5V) y tierra (0V), por lo que se hace necesario utilizar un tipo de amplificador operacional denominado *RAIL-TO-RAIL*, que tiene la particularidad de entregar la señal entre los valores máximos de polarización y que hace más eficiente el manejo de las señales analógicas.

4.7.1.2. **Tarjeta digital:** Esta tarjeta fue suministrada por el director de este proyecto. En ella se encuentra ubicado el microcontrolador PIC18F452 que realiza todo el procesamiento digital de las señales provenientes de la tarjeta analógica, y que además cuenta con dos DAC, dos puente H , y una fuente de alimentación propia que debe ser alimentada con no menos de 9V_{DC} .

5. ANALISIS DE RESULTADOS

Con el fin de medir de alguna forma, los resultados entregados por el proyecto, es básico realizar las siguientes pruebas, que en su mayoría, dependen de mecánica de precisión, por características propias del synchro, o por efectos de acoples entre éste y el motor.

5.1. PRUEBA DE CONSTANTE DE TRANSFORMACION DEL SYNCHRO

Para determinar la constante de transformación del synchro entre la inductancia de la señal de referencia y las inductancias de s1-s2, s2-s3 y s1-s3, se tomo la medida RMS de varios voltajes aplicados a la referencia y de igual forma el voltaje obtenido entre dichos terminales secundarios. La relación de transformación es igual entre todos los terminales. Los resultados están consignados en la siguiente tabla:

V_{ref} (V rms)	$V_{s1-V_{s2}}$ (V rms)
0.5	0.223
1	0.459
2	0.910
3	1.373
4	1.816
5	2.283

Tabla 5.1. Relación de voltajes entre la referencia y las salidas del synchro.

La constante de transformación puede hallarse, realizando la grafica correspondiente y la respectiva respuesta lineal. El resultado fue una constante de $K= 0.456$ como puede apreciarse en la figura 5.1.

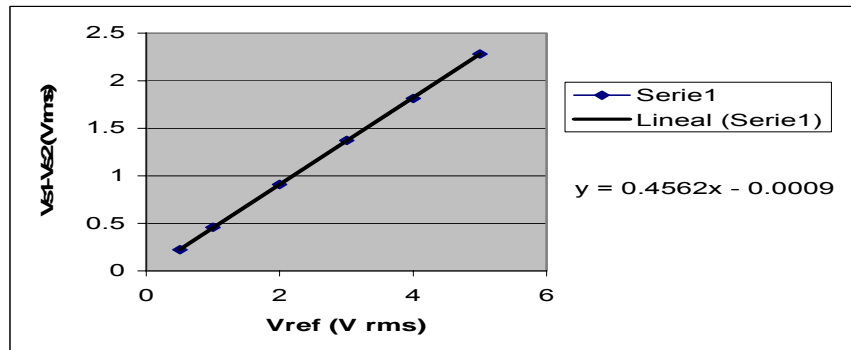


Figura 5.1. Respuesta lineal de la respuesta del transformador.

La gráfica muestra la linealidad en la respuesta del transformador, lo que garantiza el funcionamiento continuo, del circuito transformador de Scott, para generar las señales seno y coseno.

5.2. PRUEBAS DEL ALGORITMO DE SOBREMUESTREO

El funcionamiento del algoritmo puede ser verificado por etapas o de manera completa, como se describe a continuación.

5.2.1. **Respuesta del filtro IIR pasabajos:** Se implementó finalmente en el microcontrolador el filtro IIR, dado a las ventajas mencionadas en el capítulo 4. En la figura 5.2 se detalla la respuesta a entrada paso adquirida por medio del puerto serial.

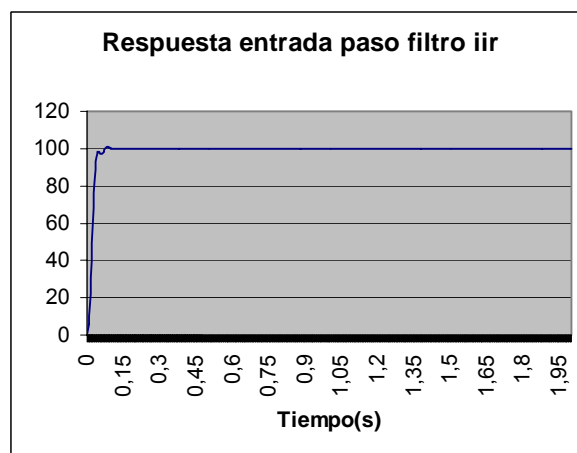


Figura 5.2. Respuesta real a entrada paso para el filtro IIR del microcontrolador.

Nótese que el tiempo de establecimiento del filtro, ante la variación en la entrada, es de aproximadamente 0.15 s. El resultado es idéntico a la simulación realizada en MATLAB, en cuanto a tiempo de establecimiento y máximo sobrepico, lo que garantiza que la medida final tiene la confiabilidad prevista durante el desarrollo.

5.2.2. Respuesta del filtro IIR de la malla de interpolación: El orden seleccionado fue de uno (1) igual al implementado en el nota de aplicación que describe el algoritmo de sobremuestreo. La función de la malla es mejorar la resolución entregada por la arcotangente y mejorar la estabilidad de la medida, por lo tanto no es prioritario cumplir con un requisito de selectividad alta para el filtro.

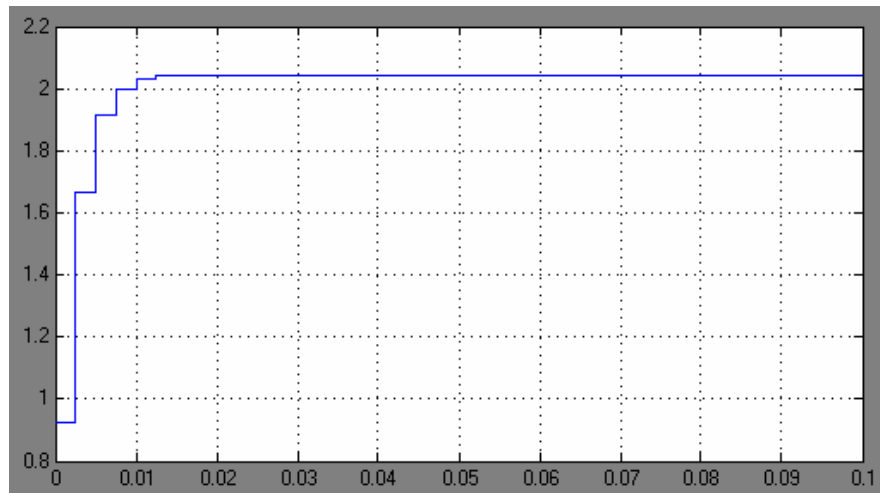


Figura 5.3. Respuesta Entrada paso para el filtro IIR de la malla de interpolación.

La figura 5.3. corresponde a la respuesta del filtro IIR de la malla de interpolación, a una entrada paso. El tiempo de establecimiento debe ser rápido para que su efecto sobre el tiempo de establecimiento de la medida total, sea mínimo como se observa en esta gráfica.

5.2.3. Respuesta de la función arcotangente: La figura 5.4 muestra la gráfica de 100 datos adquiridos de forma serial desde el microcontrolador, correspondientes a la función arcotangente. El error es la diferencia entre el dato obtenido y el resultado de la función arcotangente en una hoja de cálculo de Excel para dichos datos.

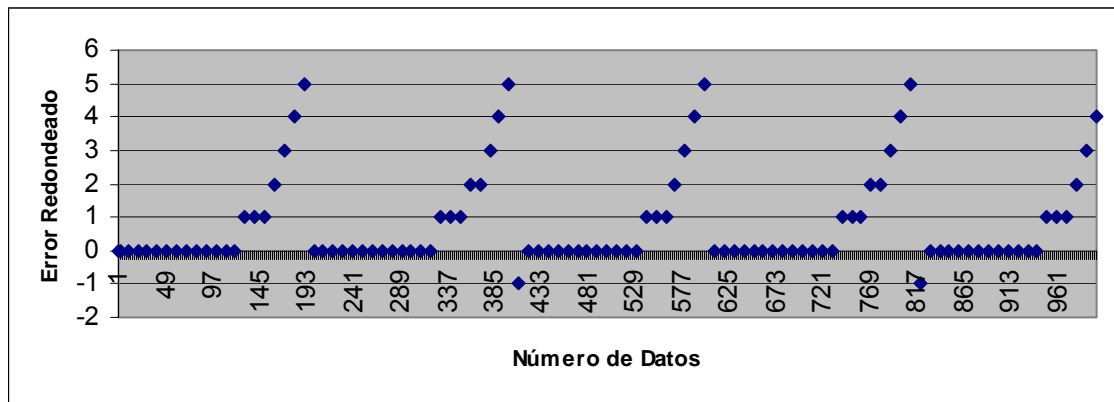


Figura 5.4. Error de la función arcotangente implementada en el microcontrolador.

Se puede apreciar la periodicidad en el incremento del error de dicha función. El error máximo es de 5 lo que corresponde a un cambio de 0.0274° en la medida. Dicho cambio no afecta la resolución de la medida digital. Pero también puede apreciarse que de 20 datos, 7 no tienen error de cero, por lo tanto, aproximadamente el 65% del resultado de la arcotangente implementada en el microcontrolador, comparada con la de Excel, tienen un error de cero. El resto de datos tienen un error mayor a 0 pero menor a 5 (dentro del rango de 0.0274° máximo). Esta máxima variación del error se encuentra por debajo de la mínima medida entregada por el sistema desarrollado (0.087°).

5.2.4. Respuesta de la malla cerrada de interpolación del ángulo: A continuación se muestran las respuestas de la malla de interpolación, bajo simulaciones hechas en MATLAB.

Para tal efecto, se realizaron varias pruebas en punto flotante, debido a la implementación final en el microcontrolador. Su sintonización se realizó variando K_p y K_i y utilizando los coeficientes para el filtro IIR calculados en el capítulo 4, y que son:

$$A = \begin{matrix} 1.0000 & 0.3400 \\ 0.9237 & 0.4258 \end{matrix}$$

Esta sintonización se realizó ajustando K_p y K_i hasta obtener la resolución de 12 bits y un tiempo de establecimiento mínimo. Para decidir las constantes más acertadas, se observó

la respuesta de malla con mejor interpolación de la medida. Para ello, se añadió a la entrada paso números aleatorios y se esperó por el mejor comportamiento.

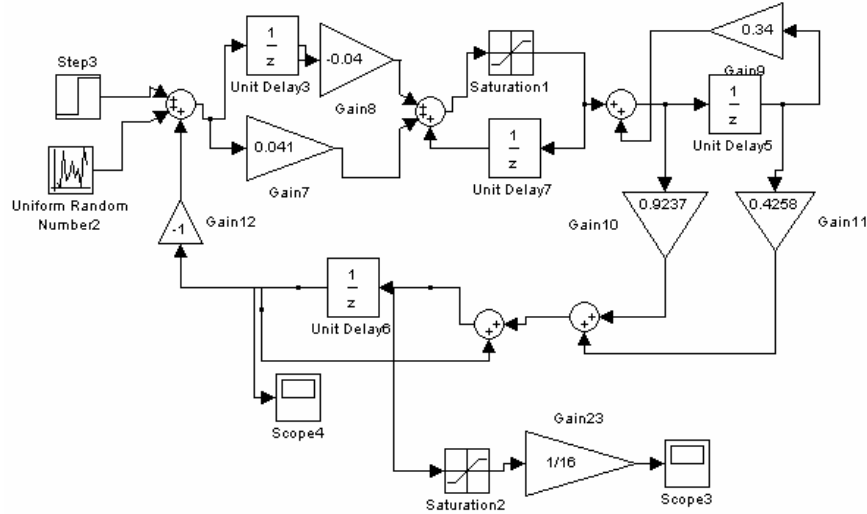
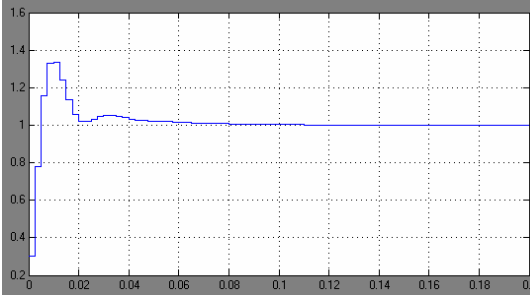
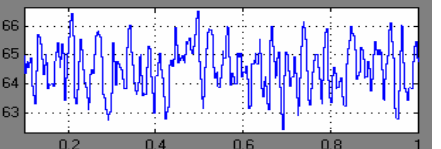
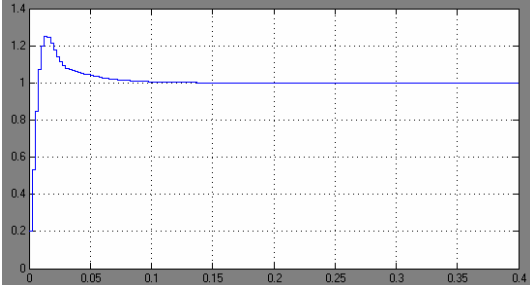
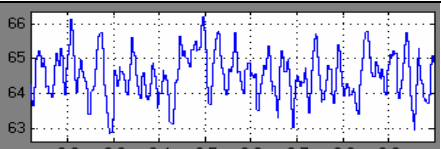
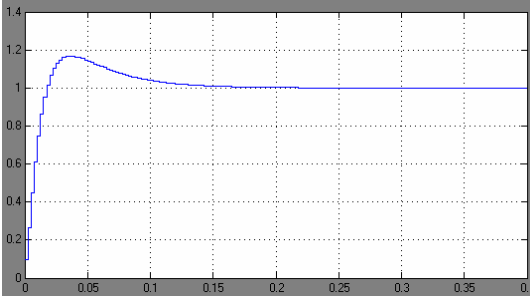
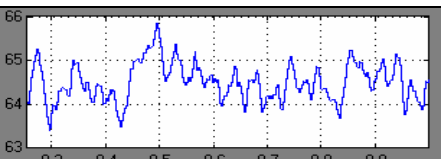
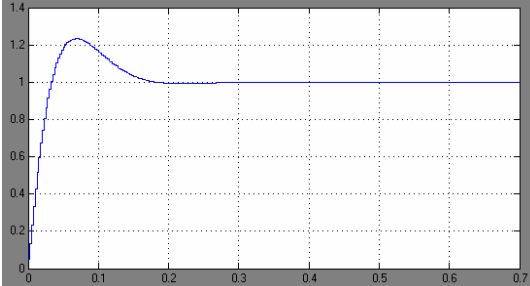
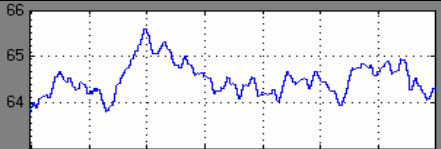


Figura 5.5. Esquema de simulación, de la malla cerrada de interpolación, ante entrada paso.

Cada una de las simulaciones se realizó añadiéndole un número aleatorio variable de 0-64 ($0-2^6$), lo que simula una entrada con resolución de 10bits equivalente a la entregada por el ADC y con un valor máximo de 64, ya que el valor que entra a la malla está adaptado a 16bits. Como se puede ver en las siguientes gráficas, a medida que se van cambiando las constantes, la interpolación va siendo cada vez mejor, hasta llegar a una variación mínima, como es la de la malla con las constantes $K_p=0.04$ y $K_i=0.001$, entendiéndose variación mínima como “no” tener variación, adaptando la salida a 12 bits. Para esta respuesta se puede notar que la medida de la malla con resolución de 12 bits permanece siempre en un mismo valor, es decir que a una entrada con resolución de 10bits (variando 1 LSB, es decir variación de $64(2^6)$ para 16bits), la salida no cambia, lo que indica que la malla interpoló y arrojó el resultado con resolución de 12bits. La entrada de la malla fue un paso de 1000 (en 16bits), lo que indica que la salida en 12bits debería ser $62 \left(\frac{1000}{2^4} \right)$. La malla interpola la variación de 0-64 y finalmente su salida se establece en 64. Ahora, si obtenemos el valor promedio a esta variación y la adaptamos a 12 bits, el resultado es 2, $\left(\frac{64}{2 \times 16} = 2 \right)$, y esto, sumado al resultado de 62, debido a la entrada paso de 1000, da como resultado final 64 que es el registrado. Esto último se aprecia en la gráfica final de la tabla 5.2, que es con la que se obtiene resolución de 12bits en la

simulación. Por todos estos motivos, las constantes seleccionadas para la simulación, fueron definitivas.

VARIABLES	RESPUESTA	CARACTERÍSTICAS Y GRÁFICA DE INTERPOLACIÓN
$K_i = 0.026$ $K_p = 0.3$		<p>Tiempo de estab = 120ms</p> <p>Max vel. Entrada rampa = 6.6 Hz</p> <p>Resolución < 12 bits</p> 
$K_i = 0.016$ $K_p = 0.2$		<p>Tiempo de estab = 150ms</p> <p>Max vel. Entrada rampa = 5 Hz</p> <p>Resolución < 12 bits</p> 
$K_i = 0.05$ $K_p = 0.1$		<p>Tiempo de estab = 200ms</p> <p>Max vel. Entrada rampa = 2 Hz</p> <p>Resolución < 12 bits</p> 
$K_i = 0.0025$ $K_p = 0.05$		<p>Tiempo de estab = 200ms</p> <p>Max vel. Entrada rampa = 1.66 Hz</p> <p>Resolución < 12 bits</p> 

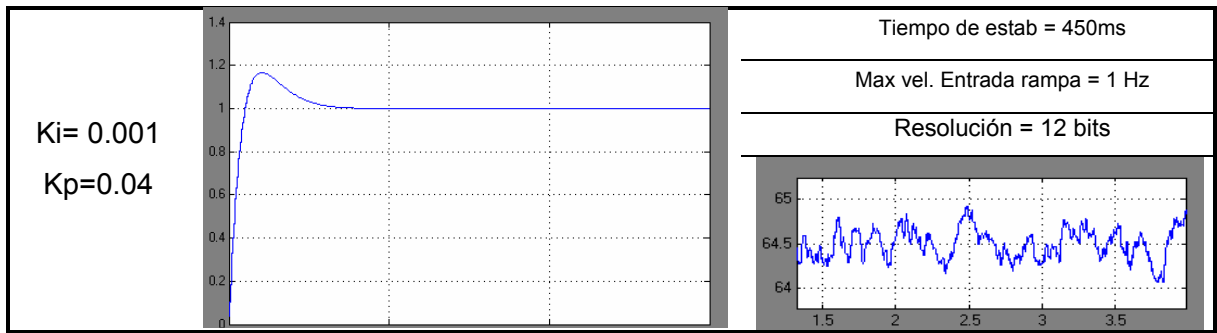


Tabla 5.2. Simulación de resolución angular y tiempo, para la malla de interpolación, de establecimiento ante variaciones de K_i y K_p .

La tabla anterior muestra las variaciones de la respuesta en tiempo de la malla de interpolación. Como puede apreciarse, el tiempo de establecimiento y la resolución depende directamente de K_p y K_i . A mayor resolución, mayor tiempo de establecimiento y por ende, la máxima velocidad que puede seguir el sistema, ante una entrada rampa disminuye. Esto es debido a que la forma de medición es absoluta, es decir, desde 0° a 360° al completar un giro. Por lo tanto, cuando la entrada es una rampa y se completa una vuelta, habrá un salto en la medida que para efectos de la malla, es una entrada paso de 360° en magnitud. Por esta razón, para que la malla continúe siguiendo la rampa, después del paso de 360° , deberá tardarse por lo menos, su tiempo de establecimiento. La última de las respuestas corresponde a los valores definitivos: $K_p=0.04$ y $K_i=0.001$.

La prueba final, para determinar la velocidad de respuesta de la malla de interpolación, es la respuesta ante una entrada rampa, es decir, cambiar constantemente el nivel de referencia de la medida a interpolar. El resultado de esta simulación es comparado con la entrada, lo que nos permitió observar que la malla puede seguir la entrada rampa a la frecuencia de entrada. En la figura 5.6 se muestra el esquema utilizado para la simulación.

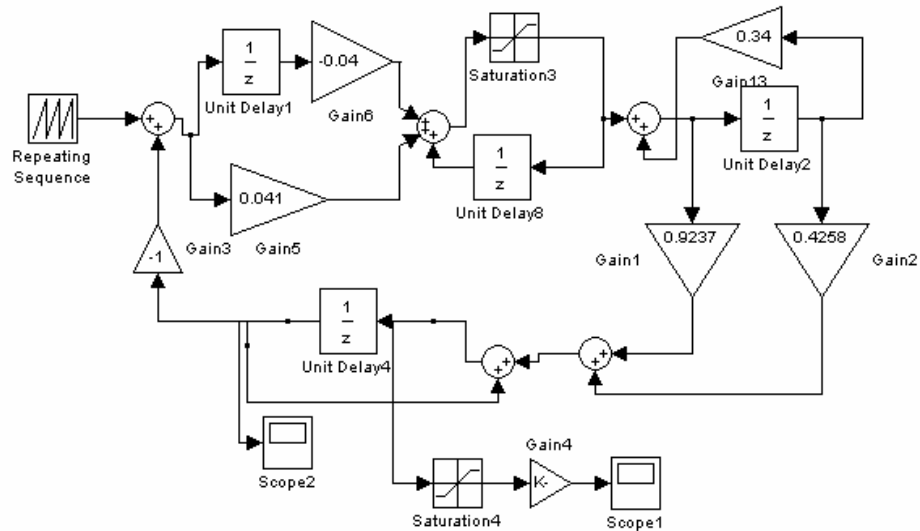


Figura 5.6. Esquema de simulación para respuesta a entrada rampa, de la malla de interpolación.

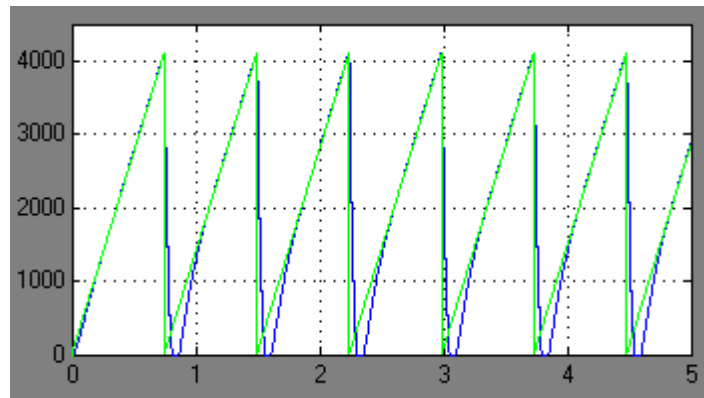


Figura 5.7. Respuesta de la malla de interpolación a entrada rampa.

En la figura 5.7 se observa que la malla sigue la entrada rampa en su totalidad a una frecuencia de 1.34Hz, es decir 482°/s.

5.2.5. Respuesta final del algoritmo de sobremuestreo: En la siguiente tabla se consigna la respuesta real del sistema completo, ante una entrada paso. Los datos reales mostrados fueron adquiridos con el osciloscopio digital Tektronix TDS-3012.

El algoritmo total se simuló tomando como frecuencia de muestreo 400Hz y utilizando punto flotante. Se usaron todos los filtros descritos en el capítulo 4 y se implementaron

mediante el *toolbox* de *signal processing blockset* de SIMULINK. La función arcotangente de MATLAB utiliza punto flotante y la malla de interpolación también es la calculada anteriormente. En la figura 5.8, se muestra el esquema general bajo el cual, se simuló el desempeño final del algoritmo ante una entrada paso.

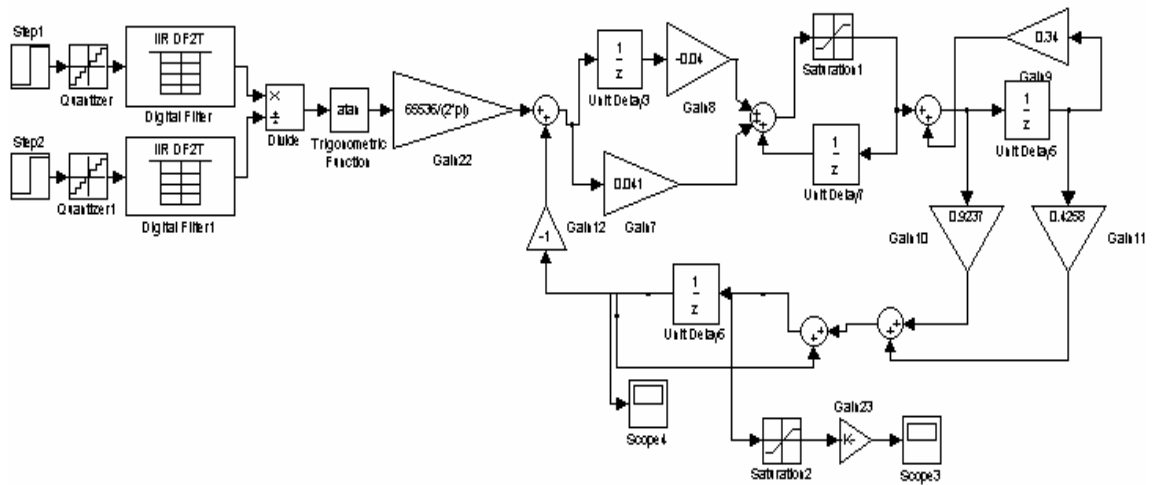
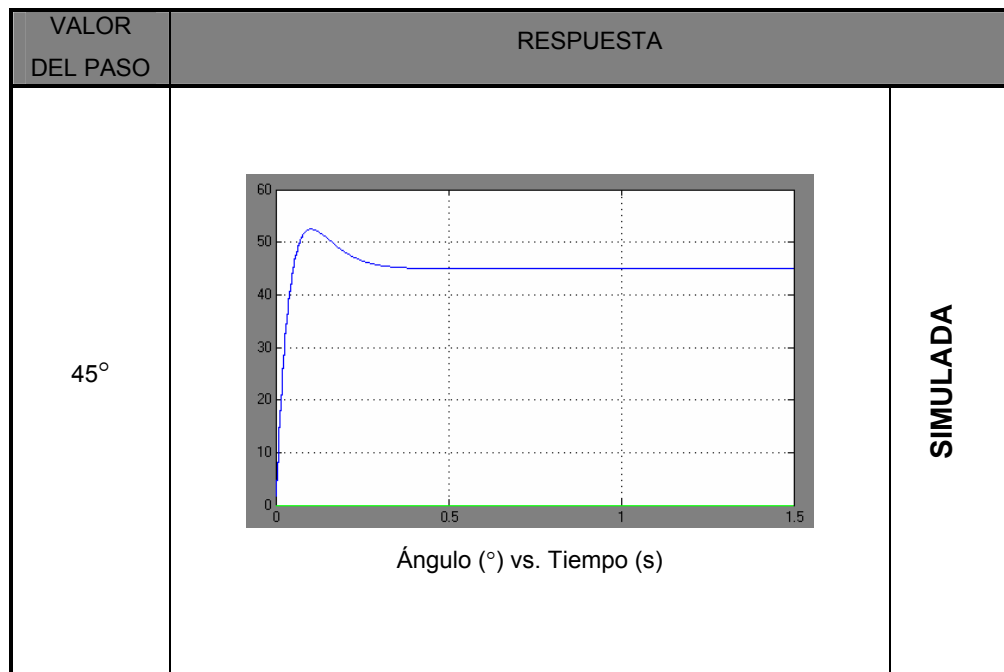


Figura 5.8. Esquema de simulación, del algoritmo de sobremuestreo, para entrada paso.



		REAL
135°	<p>Ángulo (°) vs. Tiempo (s)</p>	SIMULADA
		REAL
225°	<p>Ángulo (°) vs. Tiempo (s)</p>	SIMULADA

		REAL
315°		SIMULADA
		REAL

Tabla 5.3. Respuesta simulada y real, del algoritmo, para diferentes entrada paso.

Con lo anterior se puede corroborar que las funciones implementadas en el microcontrolador y en general, el algoritmo de sobremuestreo, es efectivamente el resultado simulado, comprobando los tiempos de establecimiento y sobre picos obtenidos.

Otra prueba importante, para determinar la velocidad del sistema, es la respuesta ante una entrada rampa, es decir, cambiar constantemente el nivel de referencia del algoritmo de sobremuestreo, teniendo como entrada de la simulación, una función seno y otra coseno. Se realizó la simulación en SIMULINK, para ver el seguimiento del sistema. La

	<p style="text-align: center;">Respuesta entrada rampa (0.398Hz)</p>	REAL
0.776Hz		SIMULADA
	<p style="text-align: center;">Repuesta entrada rampa(0.776Hz)</p>	REAL
1.34Hz		SIMULADA
	<p style="text-align: center;">Respuesta entrada rampa(1.34Hz)</p>	REAL

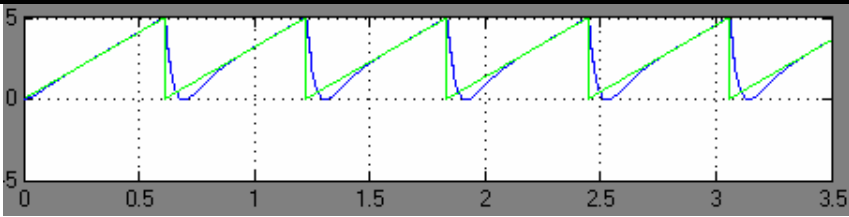
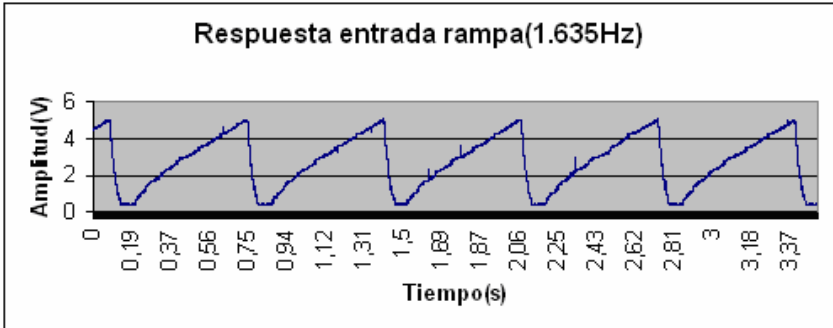
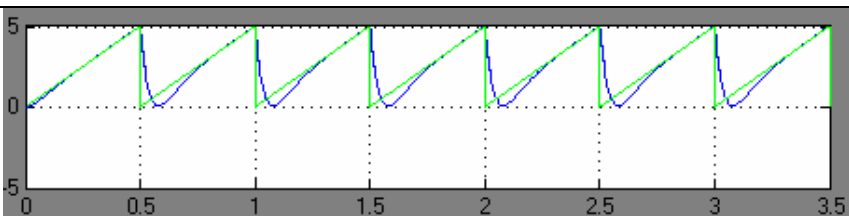
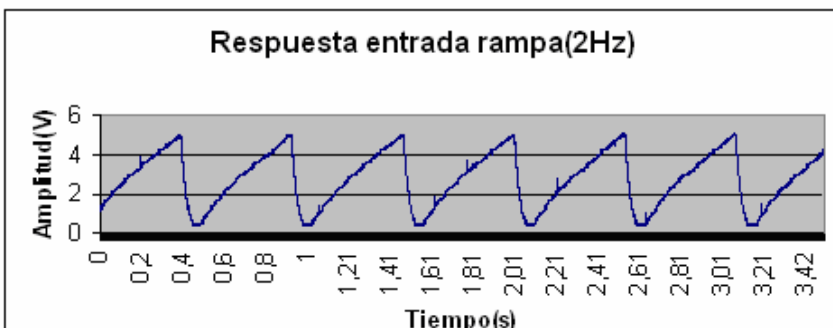
1.635Hz	 <p>Voltaje (V) vs. Tiempo (s)</p>	SIMULADA
	 <p>Respuesta entrada rampa(1.635Hz)</p> <p>Amplitud(V)</p> <p>Tiempo(s)</p>	REAL
2Hz	 <p>Voltaje (V) vs. Tiempo (s)</p>	SIMULADA
	 <p>Respuesta entrada rampa(2Hz)</p> <p>Amplitud(V)</p> <p>Tiempo(s)</p>	REAL

Tabla 5.4. Respuestas, simulada y real, del algoritmo, para diferentes entrada rampa.

Las gráficas de la tabla anterior muestran el seguimiento del sistema ante una entrada rampa. Para las diferentes frecuencias se observa que el sistema comienza retardarse, con respecto a la entrada, a medida que se incrementa la frecuencia. Para entradas

menores a 1.34Hz la respuesta es aceptable, pero no lo es para frecuencias mayores. Por tal motivo, la velocidad máxima de seguimiento del sistema es 80 rpm aproximadamente, velocidad a la cual el sistema responde linealmente a la entrada. (Ver tabla 5.4)

5.3. PRUEBA DE COMPARACIÓN CON ENCODER DE CUADRATURA INCREMENTAL

El encoder de cuadratura incremental, es quizá, el único instrumento de medición angular con el cual puede compararse nuestro sistema. Por tal motivo en esta prueba se acopló mecánicamente el rotor del synchro al del encoder, con el fin de observar los cambios en la medida de uno y otro. El error entre la medida del encoder y la del sistema de medición angular con synchro, se detalla en la figura 5.13.



Figura 5.10. Encoder Incremental de la casa *RENCO CORP*, utilizado para prueba con el sistema de medición angular.

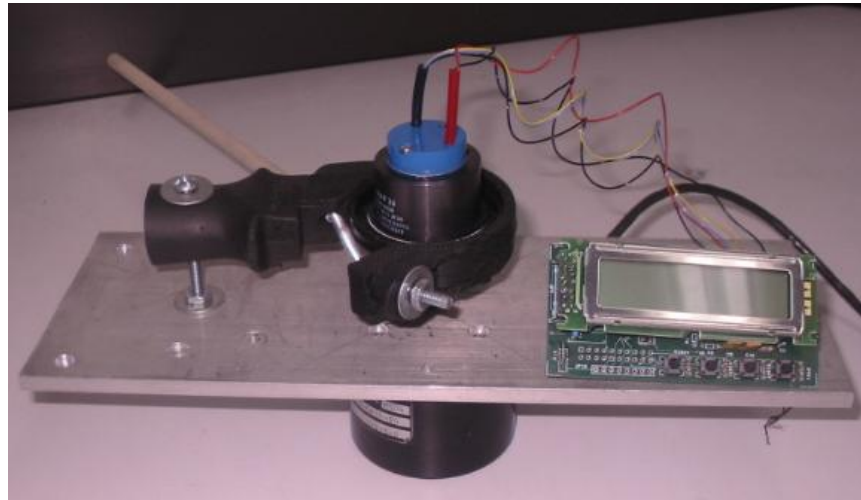


Figura 5.11.a.

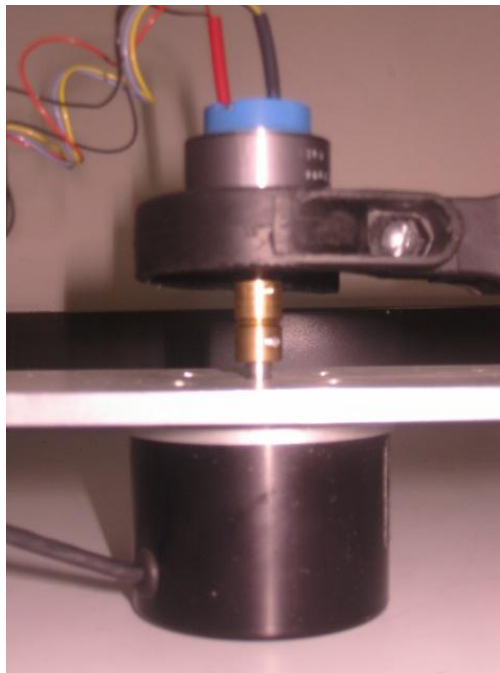


Figura 5.11.b.

Figura 5.11. Detalle del montaje mecánico entre el encoder y el synchro para prueba de comparación.

El encoder utilizado para la prueba (ver figura 5.10), tiene como resolución 500 pulsos por vuelta. Se utilizó una tarjeta de medición para encoder (suministrada por el director del proyecto – ver figura 5.19), que realiza el conteo de pulsos, incrementado por cuatro la resolución, es decir, 2000 pulsos por vuelta. Dicha tarjeta de medición cuenta los pulsos según el estado de las señales en cuadratura que recibe, dependiendo de cómo estén los bordes de subida, bajada y los niveles de ambas señales. Esta tarjeta a su vez, presenta

estos datos en una pantalla de cristal líquido, o también pueden enviarse por el puerto serial de comunicaciones, al computador.

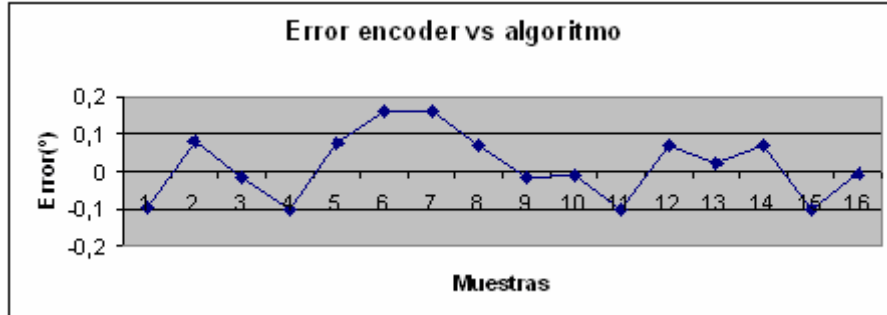


Figura 5.12. Error del encoder incremental vs. algoritmo de sobremuestreo.

Las muestras (figura 5.12) indican el número de veces que se realizó la medida para un número determinado de pulsos, que representan la variación en la medida del ángulo. Todos los errores de esta comparación, se encuentran dentro del máximo error permisible, o mínima medida del encoder, que es de 0.18° (encoder de 2000 pulsos por vuelta).

5.4. PRUEBA DE ESTABILIDAD CON LA TEMPERATURA

Esta prueba se realizó con el fin de garantizar el funcionamiento del sistema dentro de un rango de temperatura determinado. La figura 66 muestra la variación del error angular para temperaturas entre 25°C y 60°C , dentro de las cuales se garantiza mínimo cambio en la medida.

Para la prueba de temperatura se calentaron los circuitos por medio de un secador y se registró la temperatura con un termómetro digital. Las mediciones se efectuaron cada 5°C , desde 25 a 65°C . Mientras se calentaba el circuito, se registró la medida del ángulo a través de la interfaz grafica. Los resultados se observan en la figura 5.13.

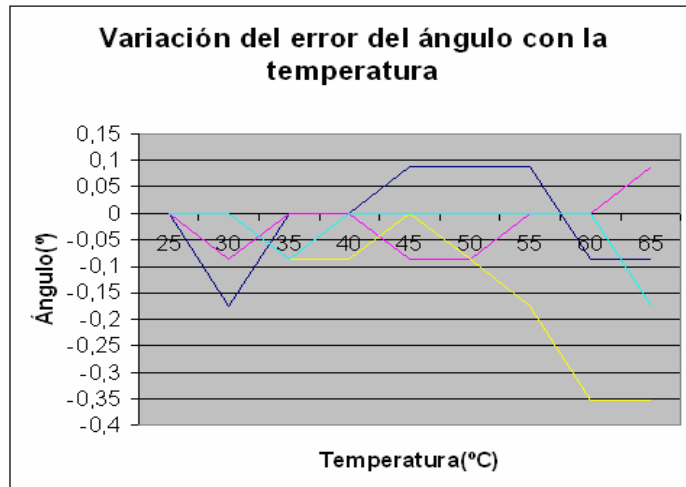


Figura 5.13. Variación del error del ángulo con respecto a la temperatura.

Las variaciones registradas con colores diferentes, corresponden a diferentes medidas de ángulo y el error se obtiene referido a la medida realizada a una temperatura de 25°C (aproximadamente temperatura ambiente). Nótese que cuando se incrementa la temperatura, a partir de 60°C, una de las medidas sufre una variación de 0.35°, que se sale del promedio de variaciones para temperaturas inferiores a ésta.

5.5. PRUEBA DE REPETICIÓN DE LA MEDIDA ANGULAR

En la figura 5.14 se muestran los resultados de las pruebas de repetición para varias medidas. Para ello se ubicó en repetidas ocasiones el rotor del synchro, mecánicamente, en la misma posición y se adquirieron, durante un mismo intervalo de tiempo, los datos de dicha posición. El histograma muestra la medida y sus variaciones para la frecuencia de la medida con respecto al número total de ellas. Nótese que la mayor frecuencia ocurre para un ángulo que está separado, aproximadamente, la mínima medida permisible para resolución de 12 bits ($\approx \pm 0.087^\circ$). Los puntos de frecuencia cercana a cero (0), no tuvieron una ocurrencia comparable con las medida máximas de cada histograma.

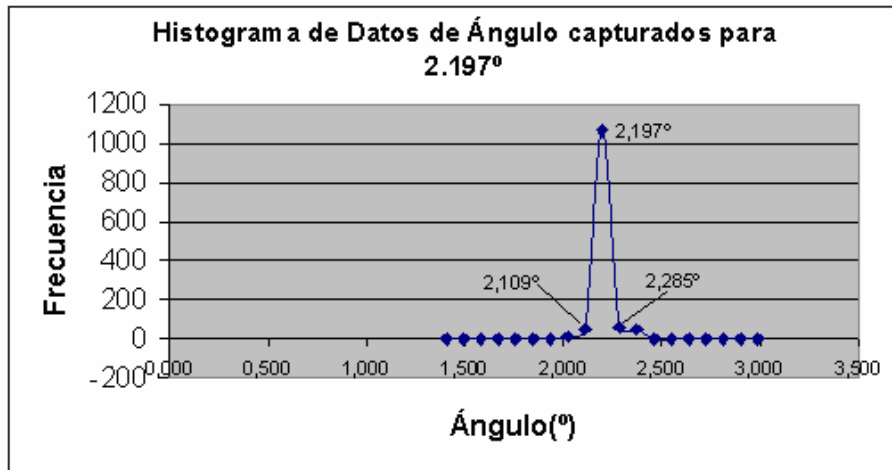


Figura 5.14 a.

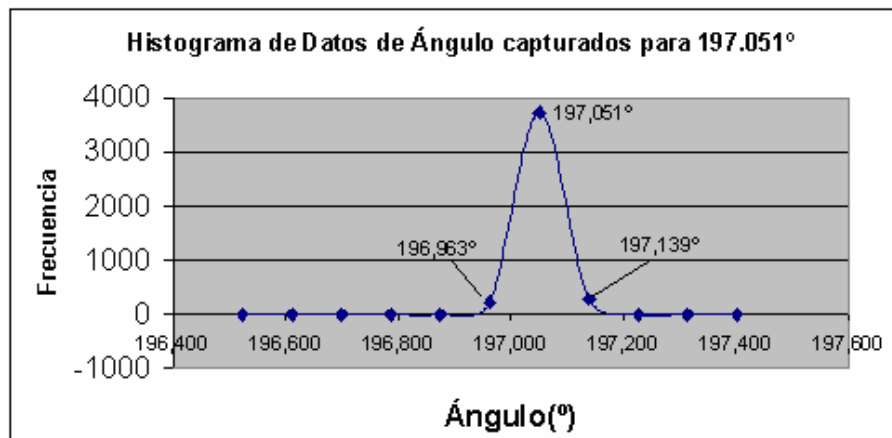


Figura 5.14 b.

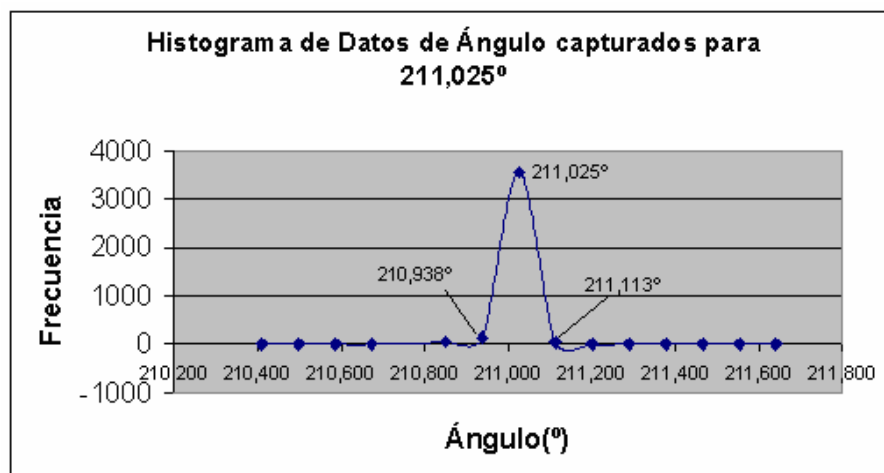


Figura 5.14 c.

Figura 5.14. Pruebas de repetición de la medida angular, para diferentes valores.

5.6. PRUEBA GEOMÉTRICA DE PRECISIÓN



Figura 5.15. Detalle de la parte externa del synchro.

La prueba geométrica permite registrar las variaciones pequeñas del ángulo, y sus efectos a largas distancias. Para esto se ubicó un apuntador láser sobre uno de los grados de libertad del montaje mecánico para la aplicación de control, con el fin de poder observar esos movimientos mínimos de ángulo, a una distancia considerable, como se muestra en la figura 5.16. Un inconveniente presentado durante esta prueba, y en general para todos los acoples mecánicos realizados, es el detallado en la figura 5.15. El eje del motor señalado en esta gráfica, es de forma de cono truncado y no cilíndrica, como los ejes convencionales de cualquier tipo motor, sensor, encoder, etc. Por lo tanto, para realizar el acople mecánico entre el rotor del synchro y cualquier otro eje, es indispensable realizar un trabajo mecánico supremamente preciso, que consiga minimizar los efectos producidos por esta forma.

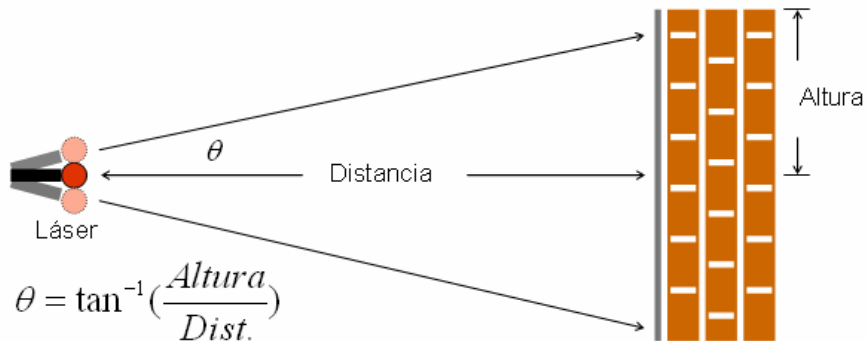


Figura 5.16. Esquema para medida geométrica de mínima variación.

La distancia para estas mediciones fue de 2237 cm y con ella, se registraron las variaciones de la altura para el mínimo movimiento angular posible del sistema y para otros. En la figura 5.17 se observa el resultado de esta prueba.

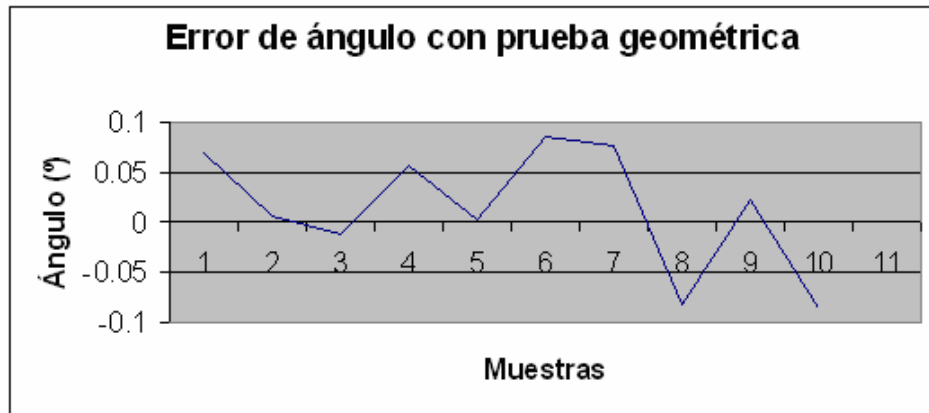


Figura 5.17. Error en la medida del ángulo ante prueba geométrica.

Nuevamente, el resultado permite observar que el máximo error se encuentra dentro del permitido para el sistema de medición.

5.7. PRUEBA DE SALIDA DE CUADRATURA DEL SISTEMA

El resultado de esta prueba, mostrado en la figura 5.18, permite apreciar el error entre la salida de cuadratura del sistema y el ángulo calculado por el algoritmo.

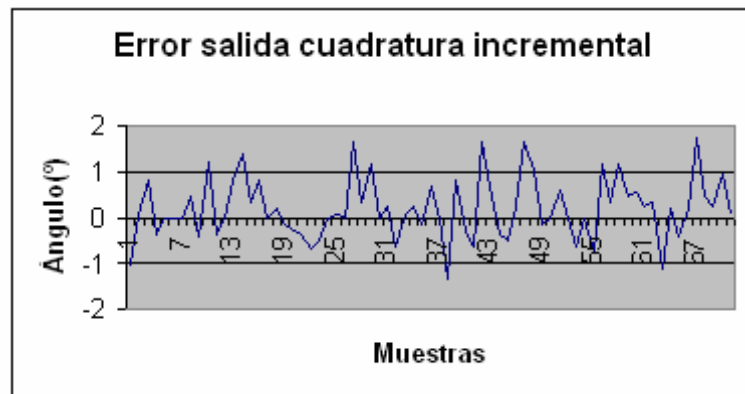


Figura 5.18. Error de la salida de cuadratura incremental



Figura 5.19. a.



Figura 5.19.b.

Figura 5.19. Sistema electrónico utilizado para medir las señales de cuadratura

Por medio de las medidas realizadas a través de un sistema electrónico para medir posición angular, de señales en cuadratura incremental, suministrado por el director del proyecto (figura 5.19), se puede obtener el error promedio y el error cuadrático medio de dicha distribución de muestras, de la siguiente forma:

$$\text{Error cuadrático medio} = \Delta x = \sqrt{\frac{\sum_{i=1}^n (x_i - \langle x \rangle)^2}{n(n-1)}} = 0,081^\circ$$

$$\text{Error promedio} = \langle x \rangle = \frac{x_1 + x_2 + \dots + x_n}{n} = \frac{\sum_{i=1}^n x_i}{n} = 0,200^\circ$$

5.8. PRUEBA DE LA APLICACIÓN DE CONTROL DE POSICIÓN

Para observar la respuesta de la aplicación de control de posición, se utilizó la interfaz gráfica (pantalla de control), para establecer un cambio en el nivel de referencia (entrada paso) y se registró la salida del DAC (señal analógica) mediante el osciloscopio TDS-3012. El resultado de esta prueba se muestra en la figura 5.20.

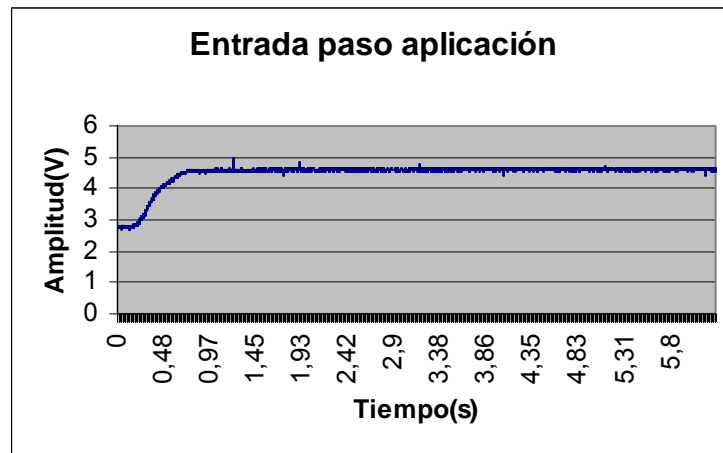


Figura 5.20. Respuesta de la aplicación de control de posición ante una entrada paso.

En la respuesta se puede apreciar que el tiempo de establecimiento del algoritmo de sobremuestreo, junto con la respuesta del controlador digital establecido para esta aplicación, tiene el resultado esperado durante los desarrollos y simulaciones hechas para su implementación final (figura 4.41).

5.9. CAPACIDAD DE PROCESAMIENTO DEL SISTEMA

La capacidad de procesamiento fue calculada utilizando los Timer del microcontrolador. A cada una de las interrupciones se le fue tomado el tiempo de ejecución con otro Timer que fue configurado para que cada incremento, correspondiera a un período del reloj del microcontrolador, los resultados que se obtuvieron se consignan en la siguiente tabla:

	Ciclos de reloj	Tiempo(s)
Timer1 (algoritmo)	18277	1.82ms
Filtro IIR entrada	2427	242.7 μ s
Arcotangente	11878	1.18ms
Malla interpolación	2180	218 μ s
Control	1334	133.4 μ s
Timer0(cuadrada)	439	43.9 μ s
Timer3(cuadratura)	97	9.7 μ s

Tabla 5.5. Medida de tiempos de procesamiento del algoritmo.

Con los datos mostrados en la tabla anterior se puede hacer el cálculo de la capacidad de procesamiento de todo el programa, debido a que el algoritmo se ejecuta cada 2.5ms, de la siguiente forma:

$$\frac{1.82ms}{2.5ms} \times 100 = 72.8\%$$

5.10. ANALISIS DE COSTOS

En la siguiente tabla se presentan los costos del trabajo final presentado.

RECUROS HUMANOS	HORAS X PERSONA	VALOR HORA	TOTAL
Director del proyecto	22 semanas x 2 horas	35.000	1'540.000
Asesor	22 semanas x 2 horas	30.000	660.000
Estudiantes	22 semanas x 6 horas	10.000	19'800.000
TOTAL			22'000.000

RECUROS FÍSICOS Y COMPONENTES	CANTIDAD	VALOR UNIDAD	TOTAL
Energía (mes)	6	70.000	420.000
Agua (mes)	6	60.000	360.000
Transporte (mes)	6	60.000	360.000
Conexión a internet (mes)	6	100.000	600.000
Alquiler de computador con licencias de software y equipo de laboratorio (mes)	6	500.000	3'000.000
Microcontrolador (Montaje Superficial)	3	40.000	120.000
Componentes	-	-	300.000
Circuito Impreso	4	15.000	60.000
Alquiler de equipo – osciloscopio digital, multímetro digital, frecuencímetro, analizador de estados y otros (mes)	6	1'500.000	9'000.000
Alquiler tarjeta digital de desarrollo (mes)	6	50.000	300.000
Montaje Mecánico	-	-	600.000
Diseño físico	-	-	200.000
TOTAL			15'320.000

RECURSOS MATERIALES	CANTIDAD	VALOR UNIDAD	TOTAL
Cartuchos de tinta	4	120.000	480.000
Fotocopias	-	-	50.000
Papel (resmas)	3	10.000	30.000
Discos Compactos	5	1.500	7.500
Empastes y velobind	-	-	60.000
TOTAL		.	627.500

TOTAL COSTOS			37'947.500
---------------------	--	--	-------------------

Tabla 5.6. Costos para el desarrollo total del Sistema de Medición Angular Microcontrolado de Alta Resolución Utilizando un Synchro.

Los costos aquí presentados, se diferencian ligeramente de los presupuestados, debido a que en estos últimos, no se tuvieron en cuenta los gastos por efectos del montaje mecánico de la aplicación, alquiler de la tarjeta digital y diseño físico del sistema.

6. CONCLUSIONES

Los resultados arrojados por proyectos anteriores sobre medición angular, utilizando alguna de las técnicas ya citadas, reflejaron la necesidad de mejorar estos efectos. La forma de conversión synchro a digital mediante el algoritmo de sobremuestreo, pudo ser implementada, en su totalidad en un microcontrolador, obteniendo excelentes efectos, con resolución de 12 bits.

Todos los objetivos trazados para este proyecto, se han cumplido a plenitud: se estableció una interfaz gráfica, bajo plataforma Windows, de fácil manejo en la cual se visualiza, el resultado generado por el sistema; la comunicación, entre la interfaz gráfica y el sistema de medición se realizó a través de un protocolo estándar de control, como lo es Modbus ASCII; las dimensiones físicas del sistema son lo suficientemente pequeñas, para hacer de él, un dispositivo portátil y manejable; el resultado de este sistema se pudo corroborar, gracias a una aplicación mecánica de dos grados de libertad, en la cual pueden establecerse los niveles de control, con excelentes resultados.

Las pruebas realizadas para verificar la resolución del sistema de medición angular, muestran que ante un instrumento para tal fin, como lo es el encoder de cuadratura incremental, el desempeño del algoritmo de sobremuestreo y en general del sistema, es superior ya que se pueden obtener medidas intermedias a la mínima medida posible con el encoder. Esta afirmación se ratifica en la segunda prueba realizada con el láser, denominada prueba geométrica, en la que se registraron máximas variaciones, de la mínima medida posible con resolución de 12 bits. Como la resolución del encoder es de aproximadamente 11 bits, y por las razones anteriormente expuestas, se puede reiterar que el funcionamiento del sistema de medición angular microcontrolado, comparado con el primero, es preeminente.

Para filtrar las señales de entrada del algoritmo de sobremuestreo, previo al cálculo de la función arcotangente, se obtuvieron excelentes resultados implementando filtros digitales

pasabajos, IIR y FIR, con características superiores, en cuanto a respuesta a entrada paso, comparado con el filtro FIR pasabanda, inicialmente propuesto por dicho algoritmo.

La decisión de desarrollar finalmente un filtro IIR digital pasabajos, se debe a que con un orden menor (orden 4) al del filtro FIR (orden 21), se puede obtener una respuesta equivalente a la de este último, facilitando su ejecución en punto flotante con el fin de garantizar un óptimo desempeño del sistema. Una potencial desventaja del filtro IIR con respecto al FIR, como es su fase no lineal a diferencia del segundo, resultó no ser un inconveniente debido a que las señales introducidas a los filtros tienen la misma frecuencia, lo que corresponde a un desfase idéntico sobre las señales.

La resolución de la malla de interpolación la proporciona el ancho de banda de la misma, por ser su característica global, la de un filtro pasabajos. Si se quisiera la mayor precisión posible en la medida, se debería reducir al máximo el ancho de banda, que está directamente relacionado con un incremento en el tiempo de establecimiento del sistema. Por lo tanto, la exactitud de la medida angular está limitada a la respuesta temporal que se quiera obtener de esta técnica, es decir, a mayor resolución, menor ancho de banda y mayor tiempo de establecimiento de la medida, lo que desmejora el seguimiento de la malla ante una entrada de variación constante, y la del proceso en general (respuesta a entrada rampa).

Durante todas y cada una de las pruebas realizadas para corroborar el mínimo cambio en la medida, se obtuvieron variaciones de no más de 0.087° en promedio, lo que significa que se logró desarrollar un sistema de medición de alta exactitud.

Es necesario que la medida del ángulo esté disponible en varios formatos, para aplicaciones futuras de este proyecto. Para tal fin se logró generar la salida de cuadratura incremental, simulando el funcionamiento de un encoder. Asimismo la medida digital y de PWM están disponibles, como también la medida entregada por el DAC, proporcional a la digital.

El montaje mecánico realizado para la aplicación de control del sistema de medición angular, demuestra el funcionamiento del mismo, y permite hacer palpable el resultado

planteado teóricamente por el algoritmo de sobremuestreo, pero a la vez, resalta el inconveniente presentado con mayor frecuencia, que es no tener un montaje mecánico que permita hacer movimientos, por lo menos, con la precisión que el algoritmo provee. Hacer de éste montaje un trabajo mecánico de alta fidelidad, incrementa los costos del proyecto, lo que se sale del presupuesto establecido para tal fin.

El microcontrolador actualmente está trabajando al 72.8% de su capacidad. Por esto, otras mejoras en cualquier aspecto, sea control, algoritmo o salidas, podrían ser implementadas.

7. BIBLIOGRAFÍA

LIBROS

[1] **BOYES**, Geoffrey. Synchro and Resolver Conversión. NAI Editors INC and Memory Devices. United Kingdom. 1980. 191 p.

Trabajos de Grado

[2] **CRUZ**, Agustín. Control de Movimiento P.I.D. digital para servomotor de 0.5kW. Centro Tecnológico de Chihuahua. 2001.

[3] **MUÑOZ**, Andrea., **PRIETO**, Alberto., **TINJACA**, Ivan. Control de Posición y Seguimiento para Telescopio con Montura Azimutal. Pontificia Universidad Javeriana. 2003.

Tomado de Internet

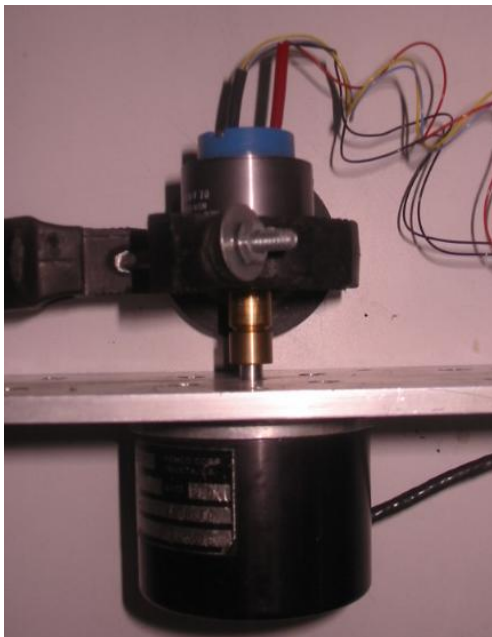
[4] Modicon Modbus Protocol Reference Guide. Modicon Inc., Industrial Automation Systems. <http://www.modbus.org>

[5] TMS320F240 DSP Solution for Obtaining Resolver Angular Position and Speed. Application Report. Texas Instruments. February 2000. <http://www.ti.com>

ANEXO A.
FOTOGRAFÍAS DEL MONTAJE FINAL Y APLICACIÓN DE CONTROL DE POSICIÓN.



Montaje mecánico de dos grados de libertad para control de posición.



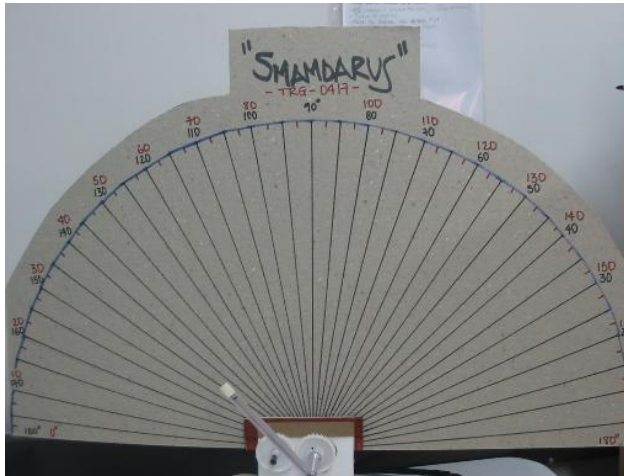
Acople mecánico entre encoder y synchro.



Hardware y montaje mecánico, para aplicación de control de posición, definitivos.



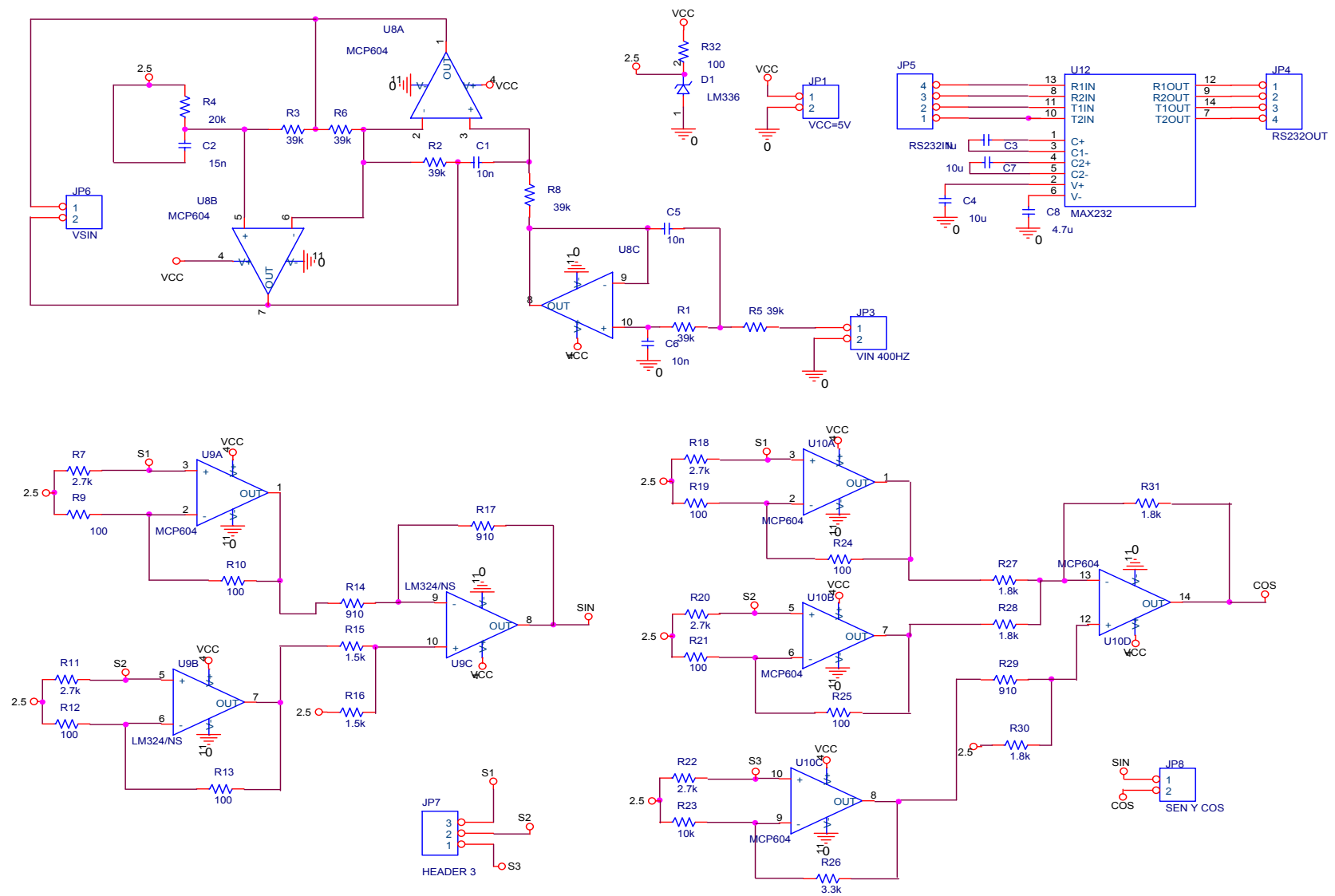
Hardware en detalle: cada una de las dos cajas contiene un sistema de medición angular, uno por cada grado de libertad.



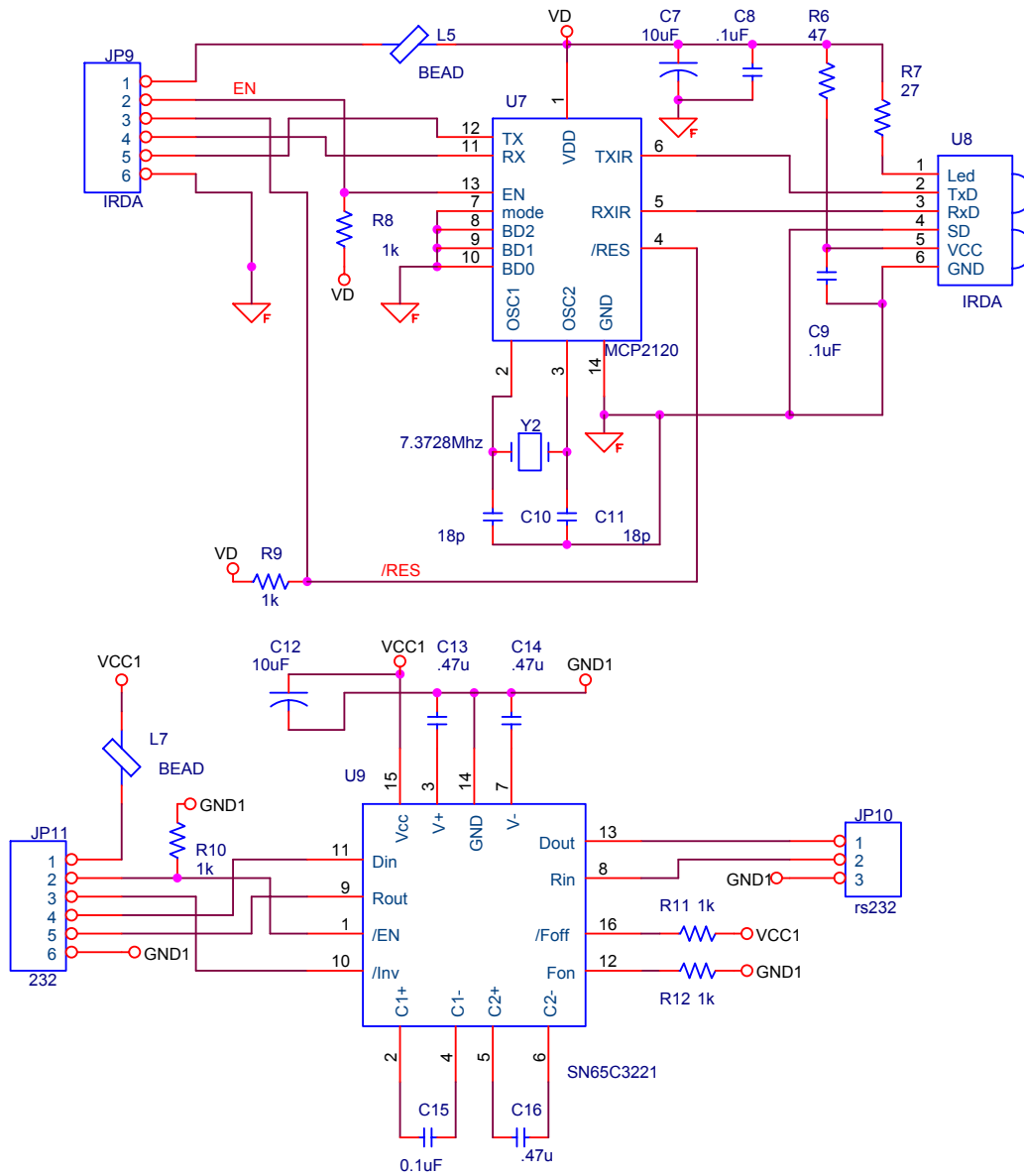
Transportador a gran escala, para verificación de ubicación de los ángulos dentro de una circunferencia.

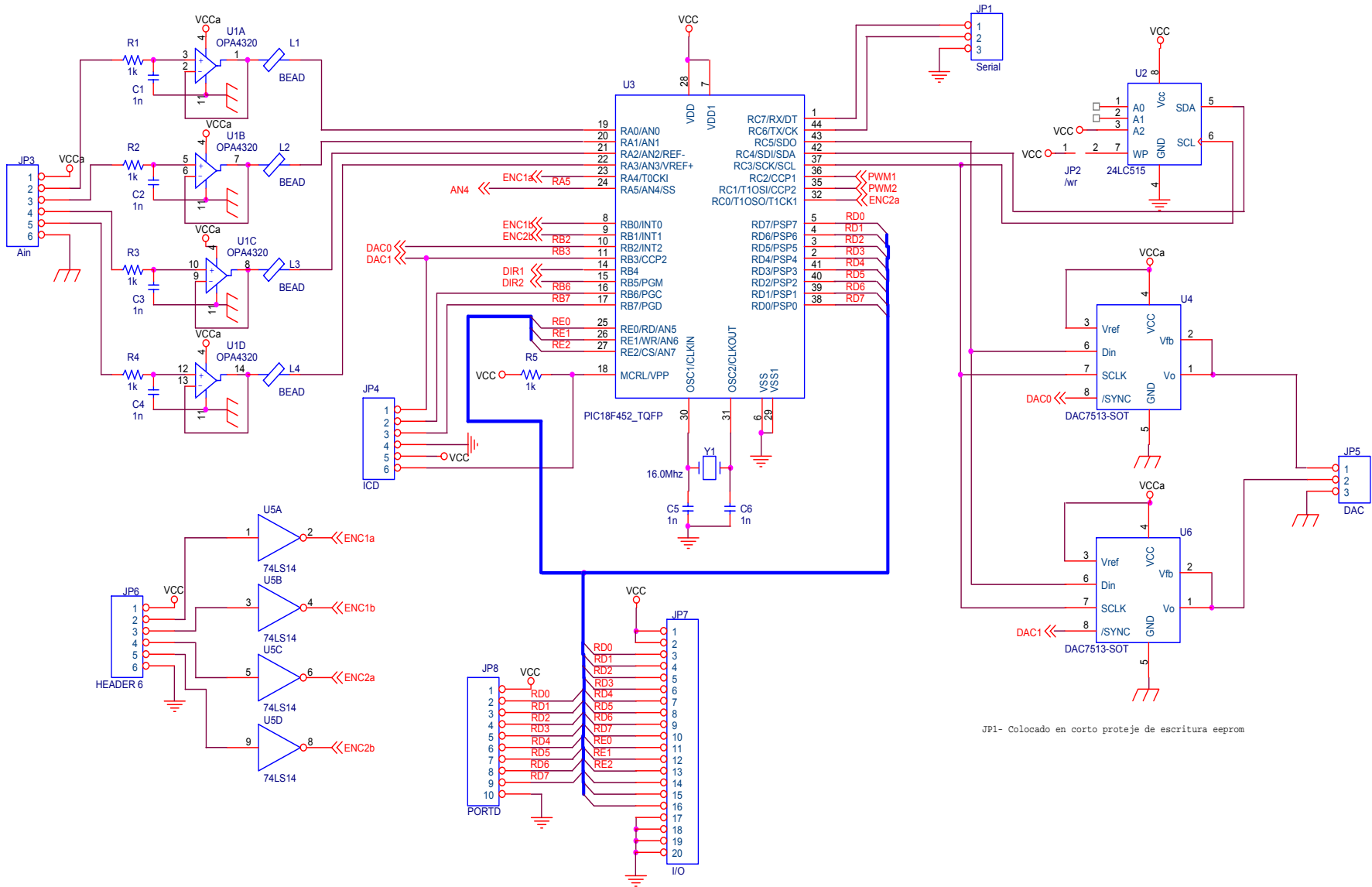
ANEXO B:
HOJA DE ESPECIFICACIONES DEL DAC7513

ANEXO C: DIAGRAMA CIRCITAL DE LA TARJETA ANALÓGICA

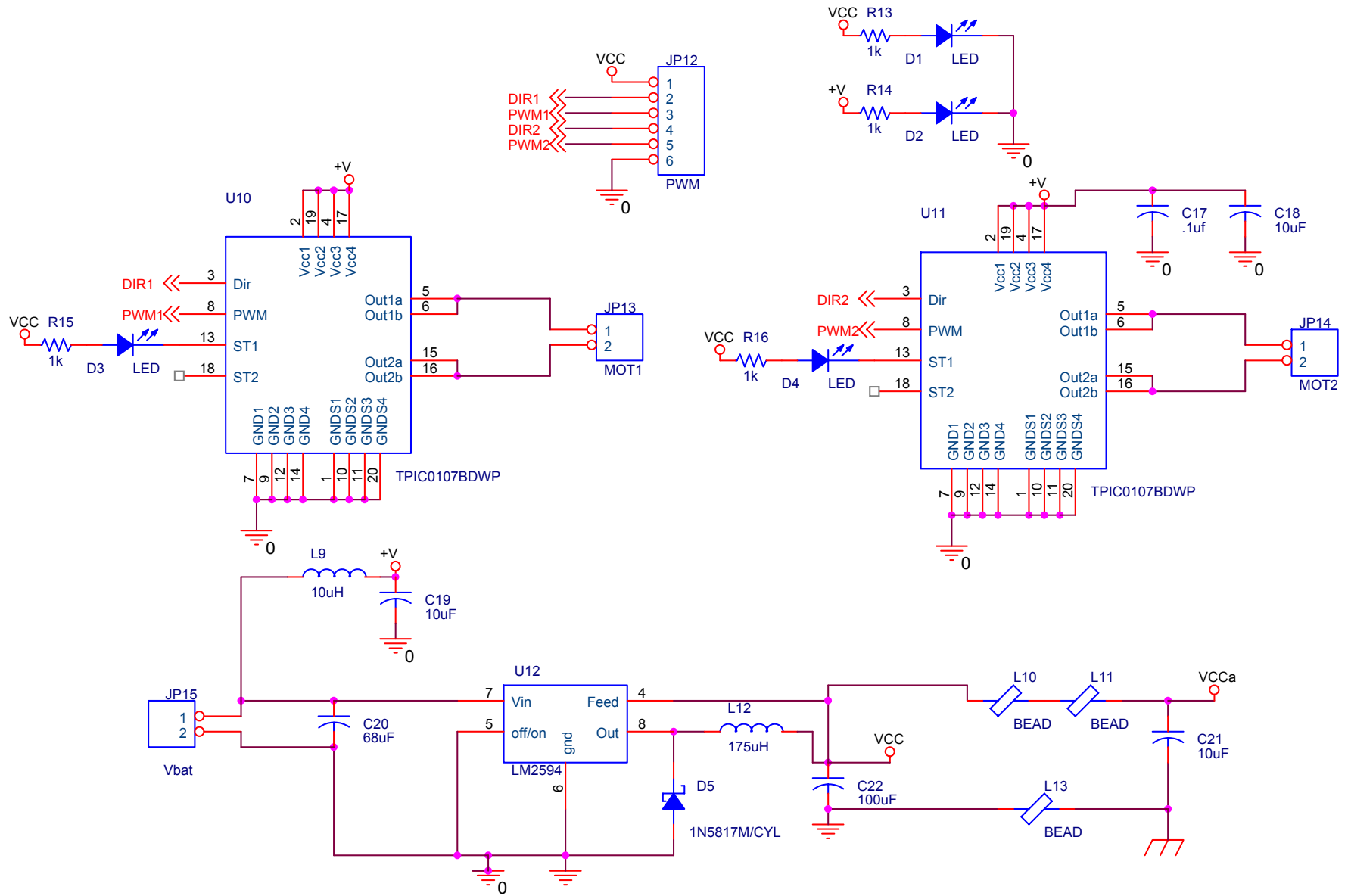


ANEXO D: DIAGRAMA CIRCUITAL DE LA TARJETA DIGITAL





JP1- Colocado en corto proteje de escritura eeprom



ANEXO E:
CÓDIGO DEL PROGRAMA IMPLEMENTADO EN EL MICROCONTRLADOR
PIC18F452

```

SINDAC()

#include <P18F452.INC>
                                UDATA_ACS

prog    CODE
extern Xdata;
extern DAC_H;
extern DAC_L;
extern c_bit;

#define DAC1    PORTB,3          ; Enable para dac 1.
#define DAC0    PORTB,2          ; Enable para dac 0.

sindac
    bcf    DAC1
    bsf    DAC0    ; Enable en cero.
    movf   DAC_H,0
    movwf  Xdata
    call   byte_out ; salida de datos a DAC
    movf   DAC_L,0
    movwf  Xdata
    call   byte_out ; transmite LSB a DAC.
    bsf    DAC0    ;enable en uno
    bsf    DAC1
    return

#define s_clk    PORTC,3          ; Salida de reloj a DAC.
#define s_dat    PORTC,5          ; salida de datos.
byte_out
    movlw  8
    movwf  c_bit
loopout
    bcf    s_clk    ; clk en 0
    bcf    s_dat    ; dato en 0
    rlcf   Xdata
    btfsc  STATUS,C
    bsf    s_dat    ; si bit a tx en uno dato en uno
                                nop
    bsf    s_clk    ; borde de subida en clk.
    decfsz c_bit
    goto  loopout    ; a enviar siguiente bit.
    bcf    s_clk    ; clk en 0
    return

```

; salida a conversor DAC, dato a la salida 12 bits en DAC_H y DAC_L (justificado a la derecha).
; canal de salida en dac_ch.
; tiempo aproximado de salida a DAC 40 uSeg @ 16Mhz.

```

        global sindac
    end

PROGRAMA
#include <p18f452.h>
#include <adc.h>
#include <stdlib.h>
#include <delays.h>
#include <pwm.h>
#include <timers.h>
#include <math.h>
#include <usart.h>
const rom char MaxBufRx = 20 ; //maximo de bytes en buffer RX.
const rom char MaxBufTx = 20 ; //maximo de bytes en buffer TX.
volatile char j,tt ;
volatile unsigned char k;
volatile unsigned int          out_reg[7];          // memoria para holding registers modbus.
400000.
volatile char Slave_hi,Slave_lo; // para numero de esclavo.
volatile unsigned char hex_buf[16];              // maneja convercion a HEX de datos recibidos.
volatile unsigned char lrc;                      // para manejo lrc.
volatile char header; // llevo caracter de inicio de secuencia ':'.
volatile unsigned char Buf_Rx[18]; // buffer para Rx y Tx serial.
volatile char ap_bufRx,ap_bufTx; // Apuntan a buffers de RX,TX.
volatile char ix,NB_tx;                          // No de caracteres a transmitir del buffer TX.
volatile int sin,cos,sw;
volatile double g,h,b,pi,g1,g2,g3,g4,y11,y12,y13,y14,h2,h1,h3,h4,y21,y22,y23,y24;
volatile int x1,ciclo;
volatile long aux,enc,enc1,angulo,pulsos;
volatile unsigned int time,ang,ang1,ind;
volatile double y1,y2,a1,d,d1,a,ep11,ep1,ep,motor,motor1,u,u1,setp,salid;
volatile unsigned char sig,lm,ust,ust1,sec,late1,late2,late11,late12,DAC_H,c_bit,Xdata,DAC_L;
extern sindac(void); //Funcion que envia los 12bits al DAC
/*-----*/
-----*/
char asci_hex ( char asc ) // convierte el dato ascii en HEX.
{
    if (asc < 0x3a) asc=asc-0x30;
    else asc=asc-0x37;
    return(asc);
}
unsigned char hex_asci ( unsigned char hex ) // convierte el dato HEX en ascii.
{
    if (hex < 0xa) hex=hex+0x30;
    else hex=hex+0x37;
    return(hex);
}
/*-----*/
-----*/
void iniciartx(void) { // Transmision de la trama para responder al maestro.
ap_bufTx=0;

for( NB_tx=0; NB_tx<15; NB_tx++){
    WriteUSART(Buf_Rx[NB_tx]);
    while(TXSTAbits.TRMT!=1);
}
}

```

```

    }
/*-----*/

void modbus(void)          // procesa comando de modbus.
                          // se recibe : X X Y Y R R R R Z Z Z Z L L CR LF
                          // XX No esclavo
                          // YY Comando.
                          // RRRR No de registro.
                          // ZZZZ munero de registros a leer o dato en registro.
{
    if((Buf_Rx[1]==Slave_hi) && (Buf_Rx[2]==Slave_lo)) // corresponde a # esclavo.
    {
        k=0;
        j=1;    // apunta a incio datos en ascii, despues de ':'.

        while ( k <= 6) // para pasar el buffer ascii a HEX.
        {
            hex_buf[k]=(asci_hex(Buf_Rx[j]));
            hex_buf[k]=hex_buf[k]*16;
            tt= asci_hex(Buf_Rx[ j+1 ]);
            hex_buf[k]=hex_buf[k] +tt;
            k++;
            j=j+2;
        }

        k=0;
        lrc=0;
        while ( k <= 5)          // para calcular LRC.
        {
            lrc=lrc+hex_buf[k];
            k++;
        }

        lrc=(0xff-lrc)+1;

        if (lrc==hex_buf[6]) // verifica si LRC OK
        {
            if(hex_buf[1]==0x06)    // funcion de escritura de registro
            {
                out_reg[hex_buf[3]]=hex_buf[4]*256+hex_buf[5]; // coloca el valor del registro.

                NB_tx=18; // numero de bytes a Tx.
                ap_bufTx=0;
                for( NB_tx=0; NB_tx<18; NB_tx++){
                    WriteUSART(Buf_Rx[NB_tx]); // Coloca dato a tx en TxReg. e incrementa apuntador.
                    while(TXSTAbits.TRMT!=1);
                }
                ap_bufTx=0;
            }

            else if(hex_buf[1]==0x03)    // funcion de lectura de registro
            {

                Buf_Rx[7]=hex_asci(((out_reg[hex_buf[3]])/256)/16);
                Buf_Rx[8]=hex_asci(((out_reg[hex_buf[3]])/256)&0b00001111);
                Buf_Rx[9]=hex_asci(((out_reg[hex_buf[3]]&0b0000000011110000)/16));
            }
        }
    }
}

```

```

Buf_Rx[10]=hex_asci((out_reg[hex_buf[3]]&0b0000000000001111));
    Buf_Rx[6]=0x34;
    Buf_Rx[5]=0x30;
    Buf_Rx[13]=0xd;
    Buf_Rx[14]=0xa;
    k=0;
    j=1;
    while ( k <= 4) // para pasar el buffer ascii a HEX.
    {
        hex_buf[k]=(asci_hex(Buf_Rx[j]));
        hex_buf[k]=hex_buf[k]*16 ;
        tt= asci_hex(Buf_Rx[ j+1 ]);
        hex_buf[k]=hex_buf[k] +tt;
        k++;
        j=j+2;
    }

    k=0;
    lrc=0;
    while ( k < 5)          // para calcular LRC.
    {
        lrc=lrc+hex_buf[k];
        k++;
    }
    lrc=(0xff-lrc)+1;
    Buf_Rx[11]=hex_asci(lrc/16);
    Buf_Rx[12]=hex_asci(lrc&0b00001111);
    NB_tx=18; // numero de bytes a Tx.
    iniciartx();          // inicia TX de respuesta.
    ap_bufTx=0;
    sec=0;
    sw=0;
    }
}

}

/*-----*/
-----*/
double powerd( double xm, int ym)//Eleva un numero xm al potencia ym (xm^ym)
{
int im=0;
double ans=1;
if(ym==0) return 1;
else
{
while( im < ym)
{
im++;
ans = ans * xm;
}
}
return ans;
}
double arctan( double xz) //Funcion arcotangente
{

```

```

int iz=0; /* counter for terms in binomial series */
int jz=1; /* sign of nth term in series */
int kq=0;
int sign = 1; /* sign of the input x */
double yz = 0.0; /* the output */
double deltay = 1.0; /* the value of the next term in the series */
double addangle = 0.0; /* used if arctan > 22.5 degrees */
if (xz < 0.0)
{
xz = -1 * xz;
sign = -1;
}

while( xz > 0.3249196962 )
{
kq++;
xz = (xz - 0.3249196962) / (1 + xz * 0.3249196962);
}
addangle = kq * 0.31415926;
while( powerd( deltay, 2) > 0.0001 )
{
iz++;
deltay = jz * powerd( xz, (2*iz - 1)) / (2*iz - 1);
yz = yz + deltay;
jz = -1 * jz;
}
return (sign * (yz + addangle) );
}
/*-----*/
-----*/
//Declaracion rutina de alta prioridad interrupcion
void toggle (void);
#pragma code HIGH_INTERRUPT_VECTOR = 0x8
void high_ISR (void)
{
_asm
goto toggle //Asignacion direccion rutina de atencion a alta prioridad
_endasm
}
#pragma code
#pragma interrupt toggle

void toggle (void)//Rutinas de alta prioridad
{
if(PIR2bits.TMR3IF==1){ //Rutina atencion timer3 para generar salida cuadratura
PIR2bits.TMR3IF=0; //Baja bandera de overflow, para habilitar una nueva interrupcion
if(sig==1){
WriteTimer3(time); //Reinicia el timer para generar los pulsos
if(pulsos>0){ //Salida en cuadratura con desfase de 90°
LATDbits.LATD5=late2;
LATDbits.LATD6=!late1;
late1=LATDbits.LATD5;
late2=LATDbits.LATD6;
}else { //Salida en cuadratura con desfase de -90°
LATDbits.LATD6=late12;
LATDbits.LATD5=!late11;
}
}
}
}
}

```

```

late11=LATDbits.LATD6;
late12=LATDbits.LATD5;
}
}
if(INTCONbits.TMR0IF==1){//Rutina atencion timer0 para generar cuadrada
WriteTimer0(53035); //Reinicia el timer para otros 1.25ms
INTCONbits.TMR0IF=0; //Baja bandera de overflow, para habilitar una nueva interrupcion
PORTDbits.RD7=lm; //Salida de la cuadrada
lm=lm+1;
if(lm==2){
lm=0;
}
if(sw<4){
sw=sw+1;
}else sw=4;

if(PORTDbits.RD7==1){ //Muestreo de seno y coseno
/*Conversion ADC*/
OpenADC( ADC_FOSC_64 & ADC_RIGHT_JUST & ADC_8ANA_0REF,//Configuracion ADC para seno
ADC_CH0 & ADC_INT_OFF );
Delay10TCYx( 5 ); // Retardo de 50 ciclos de reloj
ConvertADC(); // Comienzo de conversion
while( BusyADC() ); // Esperando por conversion
sin = ReadADC(); // Leer resultado
CloseADC(); // Deshabilitar conversor

OpenADC( ADC_FOSC_64 & ADC_RIGHT_JUST & ADC_8ANA_0REF,//Configuracion ADC para
coseno
ADC_CH1 & ADC_INT_OFF );
Delay10TCYx( 5 ); // Retardo de 50 ciclos de reloj
ConvertADC(); // Comienzo de conversion
while( BusyADC() ); // Esperando por conversion
cos = ReadADC(); // Leer resultado
CloseADC();// Deshabilitar conversor

if(out_reg[5]==1){ //Habilitar salida DAC
DAC_H=ang/4096;
DAC_L=ang/16;
sindac(); //Generar salida DAC
}

}
}
/*-----*/
if(PIR1bits.RCIF==1){ //Rutina atencion recpcion transmision serial para Modbus
sec=1;
if(ap_bufRx>=MaxBufRx)
{
ap_bufRx=0; // si pasa maximo de datos, pasa a cero.
}

if(ReadUSART()==' ') // header!=0 // detecta encabezado.
{
ap_bufRx=0; // si detecta inicio reinicializa apuntador.
header=1;
}
}

```

```

Buf_Rx[ap_bufRx]=ReadUSART();      // almacena dato recibido.
PIR1bits.RCIF==0;
if(Buf_Rx[ap_bufRx]==0x0a)
    {
        header=0;                    //para que espere siguiente encabezado.
        if(Buf_Rx[ap_bufRx-1]==0x0d) // recibio CR + LF
            {
                modbus();

                ap_bufRx=0; // reinicializa ap-buffer.
            }
        else ap_bufRx=0; // recibe LF pero no hay CR. ERROR.
    }
else ap_bufRx++; // No ha recibido LF (final de dato);
sw=0;
}
}

//Declaracion rutina de baja prioridad interrupcion
void low(void);
#pragma code low_vector=0x18
void low_interrupt(void) {
    _asm goto low _endasm //Asignacion direccion rutina de atencion a baja prioridad
}
#pragma code

#pragma interruptlow low
void low (void) { //Rutina de baja prioridad (calcula del angulo)
WriteTimer1(40535); //Reinicia el timer para otros 2.5ms
PIR1bits.TMR1IF=0; //Baja bandera de overflow, para habilitar una nueva interrupcion
if(sw==4){
    ind=ang; //Generacion indice salida cuadratura
    ind=ind/32768;
    LATDbits.LATD4=ind;
    out_reg[1]=sin;
    out_reg[2]=cos;
    g=sin-511; //Para quitar el nivel DC de las señales
    h=cos-511; //Para quitar el nivel DC de las señales
    y1=g*0.0031-0.0038*g1+0.0041*g2-0.0015*g3+0.0014*g4; //Calculo salida Filtro pasabajos IIR para el seno
    y1=y1+y11*3.3265-y12*4.2475+y13*2.4608-y14*0.5443;
    g1=g;
    g2=g1;
    g3=g2;
    g4=g3;
    y11=y1;
    y12=y11;
    y13=y12;
    y14=y13;
    y2=h*0.0031-0.0038*h1+0.0041*h2-0.0015*h3+0.0014*h4; //Calculo salida Filtro pasabajos IIR para el coseno
    y2=y2+y21*3.3265-y22*4.2475+y23*2.4608-y24*0.5443;
    h1=h;
    h2=h1;
    h3=h2;
}
}

```

```

h4=h3;
y21=y2;
y22=y21;
y23=y22;
y24=y23;

//Calculo arcotangente
if((y1>=0) & (y2>=0)){//Cuando el angulo esta en el primer cuadrante
g=y1;
h=y2;
b=(g)/h;
g=arctan(b);
}
if((y1>=0) & (y2<0)){//Cuando el angulo esta en el segundo cuadrante
g=y1;
h=y2*-1;
b=(g)/h;
g=pi-arctan(b);
}
if((y1<0) & (y2<0)){//Cuando el angulo esta en el tercer cuadrante
g=y1*-1;
h=y2*-1;
b=(g)/h;
g=arctan(b)+pi;
}
if((y1<0) & (y2>=0)){//Cuando el angulo esta en el cuarto cuadrante
g=y1*-1;
h=y2;
b=(g)/h;
g=2*pi-arctan(b);
}
if((sin==0) & (cos==0)){
g=0;
}
b=0;
b=g*10430.37835; //Escalizacion para dejar angulo en 16bits
ep=b;
//Malla de interpolacion
a=(ep-ep1)*(0.041)-(ep11)*0.04; //Calculo salida PI
ep11=ep-ep1; //Error
a=a+a1;
//Filtro IIR malla de interpolacion
a1=a;
d=a+d1*0.34;
a=0;
a=d*0.9237+d1*0.4258+ep1; //Calculo salida IIR y el integrador
d1=d;
ep1=a;
if(sec!=1){
ang=0;
ang1=0;
if(a<=65535){ //Limitador a la salida para que el angulo quede 0°-360°
ang=a;
}else{
ang=65535;
}
}

```

```

if(a<0){
ang=0;
}
out_reg[0]=ang; //Guarda el angulo calculado en registro 0 de Modbus
salid=ang/16;
enc=ang/64; //Escalizacion del angulo a 10bits
enc1=enc/64; //Escalizacion del angulo a 10bits
pulsos=enc1-enc;
enc1=ang;
if((pulsos>-8) & (pulsos<8)){ //Calculo pulsos para cuadratura
    if(pulsos==0){
        sig=0;
        WriteTimer3(0);

        }else{ sig=1;
            time=65535-4687/pulsos;
            if(pulsos<0){
                time=65535-4687/(-1*pulsos);
            }

            WriteTimer3(time);
        }

}

}else { WriteTimer3(0);
        sig=0;
    }
//Controlador Aplicacion
setp=out_reg[4];
if(setp<300){
    if(ang>32768){
        salid=salid-4096;
    }

}

if(setp>3800){
    if(ang<32768){
        salid=4096+salid;
    }

}

}
motor=((setp-salid)/4);
u=(motor)-(motor1*0.9995)+u1; //Calculo salida controlador
motor1=motor;
if(u>20000){
ciclo=300;
u=20000;
}
if(u<-20000){
ciclo=-300;
u=-20000;
}
ciclo=u;
u1=u;
if(ciclo>0){ //Direccion del motor
LATBbits.LATB5=1;
}else{ LATBbits.LATB5=0;
}

```

```

        ciclo=-1*ciclo;
    }
    if(ciclo>300){
    ciclo=300;
    }
    SetDCPWM2(ciclo);    //Configuracion ciclo util PWM
    }
    }
    }
    void EnableHighInterrupts (void)
    {
    RCONbits.IPEN = 1; /* Habilita niveles de prioridad para interrupcion */
    INTCONbits.GIEH = 1; /* Habilita todas las interrupciones de alta prioridad*/
    IPR1bits.RCIP=1;    //Configura como alta prioridad recepcion serial
    IPR2bits.TMR3IP=1;    //Configura como alta prioridad interrupcion timer3
    }
    void EnablelowInterrupts (void)
    {
    RCONbits.IPEN = 1; /* Habilita niveles de prioridad para interrupcion */
    INTCONbits.GIEL = 1; /* Habilita todas las interrupciones de baja prioridad*/
    IPR1bits.TMR1IP=0; //Configura como alta prioridad interrupcion timer1
    }
    void main (void)
    {
    //Inicializacion de variables
    a1=0;
    d=0;
    d1=0;
    a=0;
    ep11=0;
    ep1=0;
    ep=0;
    sw=4;
    lm=0;
    motor1=0;
    motor=0;
    out_reg[3]=0;
    pi=3.1415926535;
    late11=0;
    late12=0;
    out_reg[5]=0;
    ap_bufRx=0;
    ap_bufTx=0;
    NB_tx=0;    // inicializa variables para manejo serial.
    Slave_hi='0';
    Slave_lo='1'; // coloca direccion.
    header=0;    // indica que no hay caracter de inicio.
    out_reg[4]=1000;
    lm=0;
    sec=0;
    //Configuracion del todos los puertos a utilizar como salidas
    TRISD=0;
    TRISCbits.TRISC6=0;
    TRISDbits.TRISD7=0;
    TRISCbits.TRISC5=0;

```

```

TRISCbits.TRISC1=0;
TRISBbits.TRISB5=0;
TRISCbits.TRISC3=0;
TRISBbits.TRISB2=0;
TRISBbits.TRISB3=0;
//Configuracion de los Timer
OpenTimer0( TIMER_INT_ON &
T0_16BIT &
T0_SOURCE_INT &
T0_PS_1_1 );
OpenTimer1( TIMER_INT_ON &
T1_16BIT_RW &
T1_SOURCE_INT &
T1_PS_1_1 &
T1_OSC1EN_OFF &
T1_SYNC_EXT_OFF);
OpenTimer2( TIMER_INT_OFF &
T2_PS_1_16 &
T2_POST_1_8 );
OpenTimer3( TIMER_INT_ON &
T3_16BIT_RW &
T3_SOURCE_INT &
T3_PS_1_1 &
T3_OSC1EN_OFF &
T3_SYNC_EXT_OFF);
OpenPWM2(255); //Configuracion de la frecuencia del PWM

```

```

//Configuracion de los tiempos de los timers

```

```

WriteTimer0(53035);
WriteTimer1(40535);
WriteTimer3(0);
//Configuracion USART
OpenUSART( USART_TX_INT_OFF &
USART_RX_INT_ON &
USART_ASYNC_MODE &
USART_EIGHT_BIT &
USART_CONT_RX &
USART_BRGH_HIGH,
129 );

```

```

//Habilita interrupciones
EnableHighInterrupts ( );
EnablelowInterrupts ( );

```

```

while(1){
}
}

```

ANEXO F: CÓDIGO DE LA INTERFAZ GRÁFICA

```
'VENTANA DE CONTROL DE LA APLICACION
Option Explicit
'Declaracion de varibales utilizadas por esta ventana
Dim g As Boolean
Dim h As Boolean
Dim i As Boolean
Dim j As Boolean
Dim k As Boolean
Dim n As Boolean
Dim o As Boolean
Dim l As Boolean
Dim m As Boolean
Dim x As Double
Dim y As Double
Dim v As Double
Dim w As Double
Dim u As Double
Dim error As Integer
Function CloseConnection()
End Function
Private Sub Command3_Click()
'*****
'MBAXP es una herramienta para usar un control ActiveX
'sobre Visual Basic, Delphi, Excel y otros software que
'contengan aplicaciones OLE, con el fin de acceder a los datos
'de un dispositivo esclavo conectado a un PC, del protocolo
'MODUBS. Esta aplicación soporta modos ASCII, RTU y TCP/IP.
'MAYOR INFORMACION www.modbustools.com
'*****
'Lectura del registro del esclavo que contiene el angulo
Mbxp3.CloseConnection
Mbxp4.CloseConnection
Mbxp5.CloseConnection
Mbxp6.CloseConnection
'Se cierran las conexiones innecesarias
Command3.BackColor = &HFFFFFF 'blanco
Command4.BackColor = &H80FF& 'naranja
'Configuracion de la aplicacion MBAXP.
Mbxp1.Connection = 1 'Puerto serial 1
Mbxp1.BaudRate = 7 '19200 Baudios
Mbxp1.DataBits = 1 'Dato de 8 bits
Mbxp1.Parity = 0 'Paridad ninguna
Mbxp1.StopBits = 0 '1 bit de parada
Mbxp1.ProtocolMode = 1 'Mode ASCII
Mbxp1.Timeout = 1000
i = Mbxp1.OpenConnection 'Para abrir aplicacion MBAXP1
i = Mbxp1.ReadHoldingRegisters(1, 1, 0, 1, 4000) 'Funcion de
'Lectura de MODBUS ASCII
Mbxp1.UpdateEnable (1)
```

```

'De igual forma se abre la aplicacion MBAXP2
Mbaxp2.Connection = 2
Mbaxp2.BaudRate = 7
Mbaxp2.DataBits = 1
Mbaxp2.Parity = 0
Mbaxp2.StopBits = 0
Mbaxp2.ProtocolMode = 1
Mbaxp2.Timeout = 1000
j = Mbaxp2.OpenConnection
j = Mbaxp2.ReadHoldingRegisters(1, 1, 0, 1, 4000) 'Lectura
Mbaxp2.UpdateEnable (1)
End Sub
Private Sub Command5_Click()
'Escriura del valor de habilitacion del DAC sobre el registro
'destinado para tal fin. (1 en este caso para activacion)
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
Mbaxp4.CloseConnection
Mbaxp5.CloseConnection
Mbaxp6.CloseConnection
Command6.Visible = True
Command5.Visible = False
Mbaxp3.Connection = 1
Mbaxp3.BaudRate = 7
Mbaxp3.DataBits = 1
Mbaxp3.Parity = 0
Mbaxp3.StopBits = 0
Mbaxp3.ProtocolMode = 1
Mbaxp3.Timeout = 1000
Shape2.Visible = True 'Ocultar indicacion de encendido
Shape1.Visible = False 'Ocultar indicacion de apagado
m = Mbaxp3.OpenConnection
Mbaxp3.Register(1, 0) = 1
m = Mbaxp3.PresetSingleRegister(1, 1, 5, 4000)
'Escriura de registros (FUNCION MODBUS)
Mbaxp3.UpdateOnce (1)
End Sub
Private Sub Command6_Click()
'Escriura del valor de habilitacion del DAC sobre el registro
'destinado para tal fin. (0 en este caso para desactivacion)
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
Mbaxp3.CloseConnection
Mbaxp5.CloseConnection
Mbaxp6.CloseConnection
Command5.Visible = True
Command6.Visible = False
Shape2.Visible = False
Shape1.Visible = True
Mbaxp4.Connection = 1
Mbaxp4.BaudRate = 7
Mbaxp4.DataBits = 1

```

```

Mbaxp4.Parity = 0
Mbaxp4.StopBits = 0
Mbaxp4.ProtocolMode = 1
Mbaxp4.Timeout = 1000
k = Mbaxp4.OpenConnection
Mbaxp4.Register(1, 0) = 0
k = Mbaxp4.PresetSingleRegister(1, 1, 5, 4000)
Mbaxp4.UpdateOnce (1)
End Sub
Private Sub Form_Load()
'Acciones a realizar cuando se cargue el formulario
Command3.BackColor = &H80FF& 'naranja
Command4.BackColor = &HFFFFFF 'blanco
Command6.Visible = False
Command5.Visible = True
Shape2.Visible = False
Shape1.Visible = True
Label2.Visible = False
Label3.Visible = False
Label10.Visible = False
Label11.Visible = False
Label12.Visible = False
Label13.Visible = False
End Sub
Private Sub Command1_Click()
'Para salir de la ventana de control
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
Mbaxp3.CloseConnection
Mbaxp4.CloseConnection
Mbaxp5.CloseConnection
Mbaxp6.CloseConnection
Unload ControlAp
ANGUGONO.Show
Unload Me
End Sub
Private Sub Command2_Click()
'Para genera la medida del set point en grados y establecer
'el valor en el registro destinado para tal fin.
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
Mbaxp3.CloseConnection
Mbaxp4.CloseConnection
Mbaxp5.Connection = 1
Mbaxp5.BaudRate = 7
Mbaxp5.DataBits = 1
Mbaxp5.Parity = 0
Mbaxp5.StopBits = 0
Mbaxp5.ProtocolMode = 1
Mbaxp5.Timeout = 1000
g = Mbaxp5.OpenConnection
v = Text1.Text

```

```

    x = FormatNumber((v * (360 / 4096)), 2)
If (v > 4096) Or (x < 0) Then
    error = MsgBox("El SET POINT debe estar entre 0(0°) y 4096(360°)", vbOKOnly + vbDefaultButton1 +
vbApplicationModal + vbExclamation, "ERROR")
    Text1.Text = ""
Else
    Mbaxp5.Register(0, 0) = v
    g = Mbaxp5.PresetSingleRegister(0, 1, 4, 4000)
    'Escritura del set point del primer grado de libertad.
    Mbaxp5.UpdateOnce (0)
    Label10.Visible = True
    Label4 = x
End If

Mbaxp6.Connection = 2 'Serial port 2
Mbaxp6.BaudRate = 7 '19200 Baud
Mbaxp6.DataBits = 1 '8 Data bits
Mbaxp6.Parity = 0 'None parity
Mbaxp6.StopBits = 0 '1 Stop bits
Mbaxp6.ProtocolMode = 1 'ASCII Mode
Mbaxp6.Timeout = 1000
    h = Mbaxp6.OpenConnection
    u = Text6.Text
    y = FormatNumber((u * (360 / 4096)), 2)
If (u > 4096) Or (u < 0) Then
    error = MsgBox("El SET POINT debe estar entre 0(0°) y 4096(360°)", vbOKOnly + vbDefaultButton1 +
vbApplicationModal + vbExclamation, "ERROR")
    Text6.Text = ""
Else
    Mbaxp6.Register(0, 0) = u
    h = Mbaxp6.PresetSingleRegister(0, 1, 4, 4000)
    'Escritura del set point del segundo grado de libertad.
    Mbaxp6.UpdateOnce (0)
    Label11.Visible = True
    Label7 = y
End If
End Sub
Private Sub Mbaxp1_ResultOk(ByVal Handle As Integer)
'Rutina de comprobación de la respuesta del esclavo del puerto 1
If Handle = 1 Then
    If Mbaxp1.Register(1, 0) >= 0 Then
        Label12.Visible = True
        Label2.Visible = True
        'Para mostrar la medida entre 0 y 360 grados
        Label2 = FormatNumber(Mbaxp1.Register(1, 0) * (360 / 65536), 3)
    Else
        Label12.Visible = True
        Label2.Visible = True
        Label2 = FormatNumber((Mbaxp1.Register(1, 0) + 65536) * (360 / 65536), 3)
    End If
End If
End Sub

```

```

Private Sub Mbaxp2_ResultOk(ByVal Handle As Integer)
  If Handle = 1 Then
    'Rutina de comprobación de la respuesta del esclavo del puerto 2
    If Mbaxp2.Register(1, 0) >= 0 Then
      Label13.Visible = True
      Label3.Visible = True
      Label3 = FormatNumber(Mbaxp2.Register(1, 0) * (360 / 65536), 3)
    Else
      Label13.Visible = True
      Label3.Visible = True
      Label3 = FormatNumber((Mbaxp2.Register(1, 0) + 65536) * (360 / 65536), 3)
    End If
  End If
End Sub

Private Sub Mbaxp1_ResultError(ByVal Handle As Integer, ByVal error As Integer)
  If Handle = 1 Then
    If error = 10 Then
      Label2.Visible = True
      Label2 = "Sin Medida"
    End If
  End If
End Sub

Private Sub Mbaxp2_ResultError(ByVal Handle As Integer, ByVal error As Integer)
  If Handle = 1 Then
    If error = 10 Then
      Label3.Visible = True
      Label3 = "Sin Medida"
    End If
  End If
End Sub

Private Sub Command4_Click()
  'Para parar la medicion:
  Command3.BackColor = &H80FF& 'naranja
  Command4.BackColor = &HFFFFFF 'blanco
  Label2.Visible = False
  Label3.Visible = False
  Label10.Visible = False
  Label11.Visible = False
  Label12.Visible = False
  Label13.Visible = False
  Mbaxp1.CloseConnection
  Mbaxp2.CloseConnection
  Mbaxp3.CloseConnection
  Mbaxp4.CloseConnection
  Mbaxp5.CloseConnection
  Mbaxp6.CloseConnection
End Sub

'VENTANA DE MEDIDA DEL ANGULO
Option Explicit
'Declaracion de variables a utilizar
Dim f As Boolean

```

```

Dim e As Boolean
Function CloseConnection()
End Function
Private Sub Command4_Click()
'Se cargan las aplicaciones MBAXP necesarias
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
ControlAp.Show
Unload ANGUGONO
End Sub
Private Sub Command1_Click()
Label2.Visible = False
Command1.Visible = False
Command2.Visible = True
Label17.Visible = True
'Configuracion de la aplicacion MBAXP
'Para el puerto serial 1 (Synchro 1)
Mbaxp1.Connection = 1 'Puerto Serial 1
Mbaxp1.BaudRate = 7 '19200 Baudios
Mbaxp1.DataBits = 1 'Dato de 8 bits
Mbaxp1.Parity = 0 'Paridad, ninguna
Mbaxp1.StopBits = 0 '1 bit de parada
Mbaxp1.ProtocolMode = 1 'Modo ASCII
Mbaxp1.Timeout = 1000
'Para abrir el puerto serial
e = Mbaxp1.OpenConnection
'Funcion de lectura de la medida del angulo
e = Mbaxp1.ReadHoldingRegisters(1, 1, Text3, Text4, Text5) 'slave 1
Mbaxp1.UpdateEnable (1)
'Para el puerto serial 2 (Synchro 2)
Mbaxp2.Connection = 2
Mbaxp2.BaudRate = 7
Mbaxp2.DataBits = 1
Mbaxp2.Parity = 0
Mbaxp2.StopBits = 0
Mbaxp2.ProtocolMode = 1
Mbaxp2.Timeout = 1000
f = Mbaxp2.OpenConnection
f = Mbaxp2.ReadHoldingRegisters(1, 1, Text3, Text4, Text5) 'slave 1
Mbaxp2.UpdateEnable (1)
End Sub
Private Sub Command2_Click()
'Para abrir la ventana de control del angulo
Label11.Visible = False
Label13.Visible = False
Label14.Visible = False
Label4.Visible = False
Label15.Visible = False
Label16.Visible = False
Label2.Visible = True
Command1.Visible = True
Command2.Visible = False

```

```

Label17.Visible = False
Mbaxp1.CloseConnection
Mbaxp2.CloseConnection
End Sub
Private Sub Form_Load()
'Acciones a realizar cuando se cargue el formulario
Label1.Visible = False
Label4.Visible = False
Label17.Visible = False
Label15.Visible = False
Label16.Visible = False
Label13.Visible = False
Label14.Visible = False
Command2.Visible = False
End Sub
Private Sub Mbaxp2_ResultError(ByVal Handle As Integer, ByVal error As Integer)
'Comprobacion del resultado de la aplicacion MBAXP
If Handle = 1 Then
If error = 10 Then
Label4.Visible = True
Label4 = "Timeout"
If Text3 = 0 Then
Label14.Visible = True
Label14 = "Sin Medida"
End If
End If
End If
End Sub
Private Sub Mbaxp2_ResultOk(ByVal Handle As Integer)
If Handle = 1 Then
If Mbaxp2.Register(1, 0) >= 0 Then
Label4.Visible = True
'Para mostrar el resultado en resolucion 12 bits (0-4096)
Label4 = FormatNumber(Mbaxp2.Register(1, 0) * (4096 / 65536), 0)
If Text3 = 0 Then
Label16.Visible = True
Label14.Visible = True
'Para mostrar el resultado de 0 - 360 grados
Label14 = FormatNumber(Mbaxp2.Register(1, 0) * (360 / 65536), 3)
End If
Else
Label4.Visible = True
Label4 = FormatNumber((Mbaxp2.Register(1, 0) + 65536) * (4096 / 65536), 0)
If Text3 = 0 Then
Label16.Visible = True
Label14.Visible = True
Label14 = FormatNumber((Mbaxp2.Register(1, 0) + 65536) * (360 / 65536), 3)
End If
End If
End If
End Sub
Private Sub Mbaxp1_ResultError(ByVal Handle As Integer, ByVal error As Integer)

```

```

If Handle = 1 Then
  If error = 10 Then
    If Text3 = 0 Then
      Label13.Visible = True
      Label13 = "Sin Medida"
    End If
    Label1.Visible = True
    Label1 = "Timeout"
  End If
End If
End Sub
Private Sub Mbaxp1_ResultOk(ByVal Handle As Integer)
If Handle = 1 Then
  If Mbaxp1.Register(1, 0) < 0 Then
    Label1.Visible = True
    Label1 = FormatNumber((Mbaxp1.Register(1, 0) + 65536) * (4096 / 65536), 0)
    If Text3 = 0 Then
      Label15.Visible = True
      Label13.Visible = True
      Label13 = FormatNumber((Mbaxp1.Register(1, 0) + 65536) * (360 / 65536), 3)
    End If
  Else
    Label1.Visible = True
    Label1 = FormatNumber(Mbaxp1.Register(1, 0) * (4096 / 65536), 0)
    If Text3 = 0 Then
      Label15.Visible = True
      Label13.Visible = True
      Label13 = FormatNumber(Mbaxp1.Register(1, 0) * (360 / 65536), 3)
    End If
  End If
End If
End Sub

```

```

'Ventana de INICIO
Option Explicit
Private Sub Form_Click()
ANGUGONO.Show
Unload Me
End Sub
Private Sub Form_KeyPress(KeyAscii As Integer)
ANGUGONO.Show
Unload Me
End Sub
Private Sub Form_Load()
lblVersion.Caption = "Versión " & App.Major & "." & App.Minor & "." & App.Revision
End Sub
Private Sub Frame1_Click()
ANGUGONO.Show
Unload Me
End Sub
Private Sub Timer1_Timer()

```

Unload Me
ANGUGONO.Show
End Sub