



**Osciloscopio y medidor virtual inalámbrico,
implementado con tecnología .NET, utilizando
sensores y microcontrolador ARM-M0+**

T E S I S

Que para Obtener el Título de Ingeniero en Comunicaciones y Electrónica

P R E S E N T A N:

Aguilar Peñaloza Jorge David

Martínez Gómez Karina Isabel

Asesores:

Ing. Armando Mancilla León

M. en C. Roberto Galicia Galicia

M. en C. Genaro Zavala Mejía

Índice

Introducción	17
Planteamiento del problema	19
Justificación del proyecto	19
Objetivos	21
Objetivo general.....	21
Objetivos específicos	21
Capítulo 1.....	23
1. Antecedentes históricos	25
1.1 La medición en la historia	25
1.2 El osciloscopio	27
1.3 Los instrumentos de medición de diferentes magnitudes físicas en la historia	27
1.3.1 Temperatura.....	27
1.3.2 Presión.....	28
1.3.3 Humedad	31
1.3.4 Densidad de flujo magnético	31
1.4 Plataforma de desarrollo visual.....	32
Capítulo 2.....	33
2. Estado del arte	35
2.1 Tecnologías actuales usadas en osciloscopios.....	35
2.1.1 Visual Persistence Oscilloscope (VPO)	35
2.1.2 Tecnología HDO4096	35
2.1.3 Memoria inteligente MegaZoom IV	35
2.1.4 SiGe	35
2.1.5 Rendimiento puramente hardware	36
2.1.6 Digital band with interleave	36
2.1.7 Entrelazado de frecuencia.....	36
2.1.8 RealEdge	36
2.1.9 DSP potenciado	37
2.2 Herramientas de desarrollo.....	37
2.2.1 InfiniiView N8900A	37
2.2.2 Simulink	37
2.2.3 LabView	38
2.2.4 Matlab	38
2.2.5 Visual C#.....	38
2.2.6 DSP System Toolbox	38
2.2.7 Code Warrior.....	39
2.2.8 MPLAB X	39
2.3 Estudio del mercado	39

**Osciloscopio y Medidor Virtual Implementado con Tecnología .NET,
Utilizando Sensores y Microcontrolador ARM-M0+**

2.3.1 Osciloskopios de banco	39
2.3.1.1 Osciloscopio GDS3000	39
2.3.1.2 InfiniiVision series 3000x y series 4000x.....	40
2.3.1.3 LeCroyWaveSurfer serie B.....	41
2.3.1.4 Osciloskopios HDO4104	42
2.3.1.5 Infinium serie 90000 Q	42
2.3.1.6 LabMaster 10 Zi	43
2.3.2 Osciloskopios con interfaz gráfica	43
2.3.2.1 Osciloscopio USB con interfaz gráfica en LabView.....	43
2.3.2.2 PicoScope 2104 y PicoScope 2105	44
2.3.2.3 Hantek DSO2090	46
2.3.2.4 Osciloscopio registrador para PC SU1000.....	46
2.3.2.5 Hantek DSO3064	47
2.3.2.6 Fluke 190 series II ScopeMeter Test Tool.....	48
Capítulo 3.....	49
3. Marco teórico	51
3.1 El osciloscopio	51
3.1.1 Métodos de muestreo	52
3.1.1.1 Tiempo real	52
3.1.1.2 Tiempo equivalente.....	53
3.1.1.3 Muestreo aleatorio repetitivo	53
3.1.1.4 Muestreo secuencial	53
3.1.1.5 Submuestreo.....	54
3.1.2 Magnitudes físicas	54
3.2.1 Temperatura.....	54
3.2.2 Presión.....	55
3.2.3 Humedad	56
3.2.4 Densidad de flujo magnético	56
3.3 Sensores	57
3.3.1 Temperatura.....	57
3.3.1.1 Termopar.....	57
3.3.1.2 Termistor	58
3.3.1.3 Termoresistencia.....	59
3.3.2 Presión	60
3.3.2.1 Sensor capacitivo.....	60
3.3.2.2 Sensor piezoeléctrico	60
3.3.2.3 Sensor resistivo.....	61
3.3.2.4 Sensor magnético de inductancia variable.....	61
3.3.3 Humedad	62
3.3.3.1 Sensor de bulbos húmedo y seco	62
3.3.3.2 Sensor capacitivo	63

3.3.3.3 Sensor piezoelectrónico	64
3.3.4 Densidad de flujo magnético	64
3.3.4.1 Sensor de salida lineal	64
3.3.4.2 Sensor de salida digital	65
3.4 Convertidor Analógico-Digital (ADC).....	66
3.4.1 Tipos de ADC.....	68
3.4.1.1 ADC de aproximaciones sucesivas.....	68
3.4.1.2 ADC flash o paralelo	70
3.5 Amplificadores Operacionales (AO).....	71
3.5.1 Amplificador ideal.....	72
3.5.2 Amplificador Operacional como inversor.....	72
3.5.3 Amplificador Operacional en configuración sumador/restador no inversor	73
3.5.4 Amplificador Operacional en configuración diferencial.....	73
3.6 Bluetooth.....	74
3.6.1 Funcionamiento del estándar	74
3.7 Universal Asynchronous Receiver/Transmitter (UART).....	75
3.8 Protocolo serial RS-232	77
3.8.1 Convertidor de niveles RS-232	78
3.9 Interfaz Gráfica de Usuario (IGU).....	79
Capítulo 4.....	81
4. Desarrollo del Hardware	83
4.1 Selección de dispositivos	84
4.1.1 Microcontrolador	84
4.1.2 Sensores	85
4.1.2.1 Sensor de temperatura LM61CIZ	85
4.1.2.2 Sensor de presión MPX10DP	86
4.1.2.3 Sensor de humedad HMZ-433A1.....	86
4.1.2.4 Sensor de efecto Hall (detector de densidad de flujo magnético) UGN3503	87
4.2 Análisis e implementación de los sensores.....	87
4.2.1 LM61CIZ	87
4.2.2 MPX10DP	88
4.2.3 HMZ-433A1.....	89
4.2.4 UGN3503	89
4.3 Desarrollo del hardware para la entrada de una señal externa	90
4.3.1 Diseño del circuito de obtención de frecuencia	90
4.3.2 Circuito para voltajes de entrada de 20 a 120 V_{rms}	91
4.3.3 Circuito para voltajes de entrada de 3 a 20 V_{rms}	92
4.3.4 Circuito para voltajes de entrada de 300 mV_{rms} a 3 V_{rms}	93
4.4 Diseño del prototipo	93
4.4.1 Diagrama eléctrico de conexión	93
4.4.2 Circuito impreso	93

4.5 Fuente simétrica 5V	96
4.6 Módulo de comunicación serial inalámbrica (Bluetooth HC06)	97
Capítulo 5.....	99
5. Desarrollo del software	101
5.1 Desarrollo del software para el microcontrolador	101
5.1.1 Entorno de desarrollo CodeWarrior y herramienta ProcessorExpert.....	101
5.1.1.1 Ambiente del software CodeWarrior.....	102
5.1.2 Elección de los componentes usados en CodeWarrior, para el desarrollo del programa del microcontrolador	103
5.1.3 Configuración de los componentes empleados.....	104
5.1.3.1 Componente ConsoleIO.....	104
5.1.3.2 Configuración del componente ADC	105
5.1.3.3 Configuración de las interrupciones externas (ExtInt)	106
5.1.3.4 Configuración del temporizador	108
5.1.4 Algoritmo de funcionamiento para el microcontrolador	108
5.1.5 Creación del código fuente, para el funcionamiento del microcontrolador	110
5.1.6 Grabado del microcontrolador.....	113
5.2 Desarrollo del software, para la aplicación a PC	115
5.2.1 Elección del entorno de programación.....	115
5.2.1.1 Ambiente de la plataforma de desarrollo Visual C#	116
5.2.2 Elección de los componentes utilizados, para el diseño de la aplicación a PC.....	116
5.2.3 Diseño de la aplicación	117
5.2.3.1 Ventana de selección de puerto.....	117
5.2.3.2 Ventana de selección de magnitud a medir	119
5.2.3.3 Ventana de graficación de magnitud seleccionada	121
5.2.3.3.1 Para los sensores	121
5.2.3.3.2 Para el osciloscopio	129
5.2.3.4 Ventana del generador de funciones	131
Capítulo 6.....	135
6. Pruebas de funcionamiento	137
6.1 Para el medidor virtual	137
6.1.1 Medición de temperatura	137
6.1.2 Medición de humedad	139
6.1.3 Medición de densidad de flujo magnético	140
6.1.4 Medición de presión	142
6.2 Para el osciloscopio	145
6.2.1 Señal rectangular de 30 Hz.....	145
6.2.2 Señal senoidal de 60 Hz	146
6.2.3 Señal rampa de 120 Hz	147
6.2.4 Señal senoidal de 2.5 KHz	148
6.2.5 Señal pulso cuadrado a 5 KHz con 30% de ciclo útil	149

6.2.6 Señal triangular 4 KHz	150
6.3 Para generador de funciones	151
Conclusiones	155
Mejoras	157
Anexo A	161
Anexo A.1: Tablas comparativas de sensores	163
Anexo A.2: Tabla comparativa de características de diferentes microcontroladores	166
Anexo B	167
Anexo B.1: Hojas de especificaciones del sensor de temperatura LM61CIZ	169
Anexo B.2: Hojas de especificaciones del sensor de presión MPX10DP	178
Anexo B.3: Hojas de especificaciones del sensor de humedad HMZ-433A1	183
Anexo B.4: Hojas de especificaciones del sensor de efecto Hall UGN3503	188
Anexo C	193
Anexo C.1: Código del programa principal del microcontrolador	195
Anexo C.2: Código fuente de los métodos de los subsistemas del microcontrolador	200
Anexo C.3: Programa de la primera ventana de la aplicación a PC	202
Anexo C.4: Programa de la segunda ventana de la aplicación a PC	204
Anexo C.5: Programa de ventana de graficación, para los sensores	206
Anexo C.6: Programa de ventana de graficación, para el osciloscopio	212
Anexo C.7: Programa de ventana del generador de funciones	219
Anexo D	233
Anexo D.1: Evaluación económica	235
Referencias	241

Índice de figuras

Fig. 1: Termoscopio.....	27
Fig. 2: Diseños diferentes de barómetros de acuerdo a la presión a medir. (a) De caldera tradicional. (b) De extremo abierto a bajas presiones. (c) De extremo abierto a altas presiones. (d) De extremo cerrado.....	29
Fig. 3: Barómetro aneroide.....	30
Fig. 4: Barómetro de Burdon.....	30
Fig. 5: Medidor de efecto Hall (pinzas Hall)	32
Fig. 6: Osciloscopio GDS3000.....	39
Fig. 7: Osciloscopio InfiniiVision 3000x series.....	40
Fig. 8: Osciloscopio LeCroyWaveSurfer B.....	41
Fig. 9: Osciloscopio HDO4104.....	42
Fig. 10: Osciloscopio Infiniium 90000 Q.....	42
Fig. 11: Osciloscopio LabMaster 10 Zi.....	43
Fig. 12: Panel de Control.....	43
Fig. 13: Panel de recepción de datos.....	43
Fig. 14: Panel de control de gráfica.....	44
Fig. 15: Interfaz gráfica, osciloscopio USB con Labview.....	44
Fig. 16: PicoScope 2104.....	45
Fig. 17: Hantek DSO2090.....	46
Fig. 18: Osciloscopio SU1000.....	47
Fig. 19: Hantek DSO3064.....	47
Fig. 20: Fluke 190.....	48
Fig. 21: Diagrama a bloques de un osciloscopio.....	51
Fig. 22: Esquema de presión resultante a causa de una fuerza aplicada a una superficie.....	55
Fig. 23: Principio de Seebeck.....	57
Fig. 24: Termopar.....	58
Fig. 25: Conexión directa de un voltímetro y un termopar.....	58
Fig. 26: Conexión adecuada de un termopar con un voltímetro.....	58
Fig. 27: Sensor de presión capacitivo.....	60
Fig. 28: Sensor de presión piezoelectrónico y funcionamiento.....	60
Fig. 29: Sensor de presión resistivo.....	60
Fig. 30: Puente de Wheatstone.....	61
Fig. 31: Sensor de presión magnético de inductancia variable.....	61
Fig. 32: Diagrama de relación de variables.....	62
Fig. 33: Sensor de bulbo húmedo y seco.....	62
Fig. 34: Concentración de agua en las placas.....	63
Fig. 35: Diagrama de relación de variables.....	64
Fig. 36: Sensor piezoelectrónico y diagrama de relación de variables.....	64
Fig. 37: Sensor de efecto Hall de salida lineal.....	65

**Osciloscopio y Medidor Virtual Implementado con Tecnología .NET,
Utilizando Sensores y Microcontrolador ARM-M0+**

Fig. 38: Transferencia del sensor lineal.....	65
Fig. 39: Sensor de salida digital.....	65
Fig. 40: Señal del sensor.....	66
Fig. 41: Composición de un ADC.....	67
Fig. 42: Diagrama a bloques de un ADC de aproximaciones sucesivas.....	69
Fig. 43: Diagrama interno de un ADC flash.....	70
Fig. 44: Símbolo del Amplificador Operacional.....	71
Fig. 45: Alimentación de un AO.....	71
Fig. 46: Circuito conectado en cascada con un AO.....	72
Fig. 47: Circuito esquemático de un AO como inversor.....	72
Fig. 48: Circuito esquemático de un AO en configuración sumador/restador no inversor.....	73
Fig. 49: Circuito esquemático de un AO en configuración diferencial.....	73
Fig. 50: Diagrama de bloques de un UART.....	75
Fig. 51: Esquema de transmisión de un UART.....	76
Fig. 52: Diagrama a bloques del transmisor del USART0 del microcontrolador MKL25Z128.....	76
Fig. 53: Diagrama a bloques del receptor del USART0 del microcontrolador MKL25Z128.....	77
Fig. 54: Protocolo de comunicación RS232.....	78
Fig. 55: Convertidor de niveles TTL a RS-232 utilizando el MAX3232.....	79
Fig. 56: Diagrama a bloques del hardware del proyecto.....	83
Fig. 57: Tarjeta de desarrollo FRDM-KL25Z.....	84
Fig. 58: Sensor de temperatura LM61CIZ.....	85
Fig. 59: Sensor de presión MPX10DP.....	86
Fig. 60: Sensor de humedad HMZ-433A1.....	86
Fig. 61: Sensor de efecto Hall UGN3503.....	87
Fig. 62: Conexión del sensor de temperatura LM61CIZ.....	88
Fig. 63: Acondicionamiento de señal e implementación del sensor de presión MPX10DP.....	88
Fig. 64: Configuración del sensor de humedad HMZ-433A1.....	89
Fig. 65: Implementación del sensor de efecto Hall UGN3503.....	90
Fig. 66: Diagrama eléctrico para la obtención de la frecuencia de la señal.....	90
Fig. 67: Diagrama eléctrico, para entrada de señal externa entre 20 y 120 Vrms.....	91
Fig. 68: Diagrama eléctrico, para entrada de señal externa entre 3 y 20 Vrms.....	92
Fig. 69: Diagrama eléctrico, para entrada de señal externa entre 300 mVrms y 3 Vrms.....	93
Fig. 70: Diagrama esquemático del prototipo.....	94
Fig. 71: Diseño de las pistas del PCB del prototipo.....	95
Fig. 72: Diseño de la placa del prototipo con los dispositivos correspondientes.....	95
Fig. 73: Placa del prototipo.....	95
Fig. 74: Placa del prototipo con los dispositivos soldados.....	96
Fig. 75: Diagrama de conexión eléctrica de la fuente simétrica de 5V para alimentación de AO.....	96
Fig. 76: Pistas para la placa de la fuente simétrica de 5V.....	96
Fig. 77: Placa de la fuente simétrica de 5V, sin transformador.....	97
Fig. 78: Placa de la fuente simétrica de 5V, con el transformador conectado.....	97

**Osciloscopio y Medidor Virtual Implementado con Tecnología .NET,
Utilizando Sensores y Microcontrolador ARM-M0+**

Fig. 79: Diagrama eléctrico de conexión del módulo Bluetooth.....	98
Fig. 80: Diseño de la placa del módulo Bluetooth HC06.....	98
Fig. 81: Placa de adaptación, para módulo Bluetooth HC06.....	98
Fig. 82: Entorno CodeWarrior.....	102
Fig. 83: Elementos agregados, utilizados en el software CodeWarrior.....	104
Fig. 84: Configuración del componente de serial.....	105
Fig. 85: Configuración del componente ADC.....	106
Fig. 86: Configuración del componente de interrupción externa ExtInt1.....	107
Fig. 87: Configuración del componente de interrupción externa ExtInt.....	107
Fig. 88: Configuración del componente de interrupción externa ExtInt3.....	107
Fig. 89: Configuración del componente Timer.....	108
Fig. 90: Diagrama UML para el proceso a ejecutar por el microcontrolador.....	109
Fig. 91: Métodos usados, del componente ADC.....	111
Fig. 92: Métodos del componente de interrupción externa.....	111
Fig. 93: Métodos del componente Timer.....	112
Fig. 94: Generación del código para cada uno de los componentes y métodos usados.....	112
Fig. 95: Compilación del programa creado en el software CodeWarrior.....	113
Fig. 96: Grabado del microcontrolador.....	114
Fig. 97: Grabado proyecto CodeWarrior en el microcontrolador.....	114
Fig. 98: Ejecución del programa en el microcontrolador.....	114
Fig. 99: Diagrama a bloques del software del proyecto.....	115
Fig. 100: Ejecución del programa en el microcontrolador.....	116
Fig. 101: Ventana para la selección de puerto COM.....	117
Fig. 102: Diagrama de clases para la ventana de selección de puerto COM.....	118
Fig. 103: Ejecución de la primera parte de la aplicación.....	118
Fig. 104: Diseño de la ventana de selección de magnitud a analizar.....	120
Fig. 105: Diagrama de clases para la ventana de selección de magnitud a graficar.....	120
Fig. 106: Diseño de la ventana de graficación de temperatura.....	122
Fig. 107: Diseño de la ventana de graficación de presión.....	122
Fig. 108: Diseño de la ventana de graficación de humedad.....	123
Fig. 109: Diseño de la ventana de graficación de efecto Hall.....	123
Fig. 110: Plano cartesiano.....	123
Fig. 111: Plano en pantalla.....	124
Fig. 112: Forma de graficación en el plano cartesiano.....	125
Fig. 113: Forma de graficación en el plano en pantalla.....	126
Fig. 114: Diagrama de clases, para la venta de graficación de magnitud proveniente de un sensor.....	128
Fig. 115: Diseño de la ventana del osciloscopio.....	129
Fig. 116: Diagrama de clases, para la venta de graficación, del osciloscopio.....	130
Fig. 117: Diseño de la ventana del generador de funciones.....	131
Fig. 118: Funcionamiento de la ventana del generador de funciones.....	133
Fig. 119: Diagrama de clases del generador de funciones.....	133

Fig. 120: Señal de 10 muestras del sensor de temperatura.....	137
Fig. 121: Lectura de voltaje para el sensor de temperatura.....	138
Fig. 122: Calentamiento del sensor de temperatura.....	138
Fig. 123: Respuesta del sensor de temperatura al ser calentado, mostrado en la interfaz gráfica creada	139
Fig. 124: Señal de 10 muestras del sensor de humedad.....	139
Fig. 125: Lectura del voltaje para el HMZ4331.....	140
Fig. 126: Señal de 10 muestras del sensor de efecto Hall.....	140
Fig. 127: Lectura del voltaje del sensor de efecto Hall.....	141
Fig. 128: Inducción de flujo magnético al sensor de efecto Hall.....	141
Fig. 129: Respuesta del sensor de efecto Hall, mostrado en la interfaz gráfica creada, al variar la densidad de flujo magnético.....	142
Fig. 130: Pruebas para el sensor de presión.....	142
Fig. 131: Señal de 10 muestras, del sensor de presión diferencial.....	143
Fig. 132: Nivel de presión uno.....	143
Fig. 133: Nivel de presión dos.....	144
Fig. 134: Respuesta del sensor, al variar la presión, mostrado en la interfaz gráfica creada.....	144
Fig. 135: Señal rectangular de 30 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	145
Fig. 136: Señal rectangular de 30 Hz, mostrada en el proyecto desarrollado.....	145
Fig. 137: Señal senoidal de 60 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	146
Fig. 138: Señal senoidal de 60 Hz, mostrada en el proyecto desarrollado.....	146
Fig. 139: Señal rampa de 120 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	147
Fig. 140: Señal rampa de 120 Hz, mostrada en el proyecto desarrollado.....	147
Fig. 141: Señal seno de 2.5 KHz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	148
Fig. 142: Señal seno de 2.5 KHz, mostrada en el proyecto desarrollado.....	148
Fig. 143: Pulso cuadrado de 5 KHz, con 30% de ciclo útil, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	149
Fig. 144: Pulso cuadrado a 5KHz, con 30% de ciclo útil, mostrada en el proyecto desarrollado.....	149
Fig. 145: Señal triangular de 4 KHz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.....	150
Fig. 146: Señal rampa a 4 KHz, mostrada en el proyecto desarrollado.....	150
Fig. 147: Mensaje de error, al iniciar graficación y no haber seleccionado alguna función.....	151
Fig. 148: Generación de función coseno, graficada en un rango de -6 a 4, sin variaciones de amplitud, tiempo o velocidad.....	151
Fig. 149: Generación de función seno, graficada en un rango de -3 a 14, con variación de amplitud.....	152
Fig. 150: Generación de una señal pulso, graficada en un rango de 0 a 22, con periodo de 7, ciclo útil de 2 y variación de amplitud.....	152

**Osciloscopio y Medidor Virtual Implementado con Tecnología .NET,
Utilizando Sensores y Microcontrolador ARM-M0+**

Fig. 151: Generación de una señal pulso, graficada en un rango de 0 a 22, con periodo de 7, ciclo útil de 2, y variación de amplitud, tiempo y velocidad.....	153
Fig. 152: Generación de la función rampa, graficada en un rango de -5 a 6, con periodo de 4, y variación de amplitud, tiempo y velocidad.....	153
Fig. 153: Generación de la función rectangular, graficada en un rango de -3.3 a 4.9, con periodo de 3.8, y variación de amplitud, tiempo y velocidad.....	154
Fig. 154: Circuito de protección, para señales de entrada externa.....	159

Índice de tablas

Tabla 1: Cuadro comparativo entre materiales, según su rango de operación y precisión.....	59
Tabla 2: Proceso de un ADC de 10 bits cuando $V_{in} = 4\text{ V}$	69
Tabla 3: Cuadro comparativo entre ADC de aproximaciones sucesivas y ADC flash o paralelo.....	71
Tabla 4: Sensores seleccionados para el prototipo.....	85
Tabla 5: Correspondencias carácter-proceso.....	110
Tabla 6: Correspondencias terminal-subsistema-circuito.....	113
Tabla 7: Cuadro comparativo entre coordenadas físicas y coordenadas de pantalla.....	124
Tabla A.1.1: Comparación de características principales de diferentes sensores de temperatura.....	163
Tabla A.1.2: Comparación de precios, proveedores y disponibilidad en sensores de temperatura.....	163
Tabla A.1.3: Comparación de características principales de diferentes sensores de presión.....	164
Tabla A.1.4: Comparación de precios, proveedores y disponibilidad en sensores de presión.....	164
Tabla A.1.5: Comparación de características principales de diferentes sensores de humedad.....	164
Tabla A.1.6: Comparación de precios, proveedores y disponibilidad en sensores de humedad.....	164
Tabla A.1.7: Comparación de características principales de diferentes sensores de efecto Hall.....	165
Tabla A.1.8: Comparación de precios, proveedores y disponibilidad en sensores de efecto Hall.....	165
Tabla A.2.1: Comparación de características principales de diferentes microcontroladores.....	166
Tabla D.1.1: Comparación de características entre el proyecto creado y el Osciloscopio USB con Interfaz Gráfica en LabView.....	235
Tabla D.1.2: Desglose de costo del proyecto.....	236
Tabla D.1.3: Desglose de gastos en la realización del proyecto.....	238

Introducción

En la educación media superior y superior, especialmente, en la rama de la electrónica, los estudiantes día a día, se enfrentan ante diversos problemas, uno de ellos, es la necesidad de analizar el comportamiento de diferentes magnitudes físicas, requiriendo así, hacer uso de gran cantidad de instrumentos, los cuales suelen ser robustos, complejos de utilizar, y de precio elevado, encontrando otro problema, ya que, para un alumno, no es fácil adquirir y poseer en casa uno de estos instrumentos de medición, disminuyendo la velocidad de aprendizaje.

Ya que nos encontramos en la era digital, en este escrito, se propone la computadora portátil o de escritorio, como una solución a la falta de instrumentos de medición en casa, pues la mayoría de los estudiantes cuenta con una de éstas en casa. Para ello, se hace el uso de los lenguajes de programación, sin embargo, a su vez, se requiere de un sistema de adquisición de datos, para el procesamiento de las diversas señales que se desean analizar.

El presente trabajo, plantea la creación de un osciloscopio para PC, con interfaz inalámbrica (Bluetooth), de bajo costo, y bajo consumo energético, utilizando la plataforma de desarrollo “Visual C#”, para la creación de la interfaz gráfica, implementando un sistema de adquisición de datos, con la ayuda de un microcontrolador de 32 bits, y un sistema de acondicionamiento de señales, para un buen y correcto funcionamiento. De esta manera, el prototipo puede ser utilizado en aplicaciones o sistemas, tales como:

- Control de temperaturas
- Termómetros
- Alarmas indicadoras de incongruencia
- Sistemas de aire acondicionado
- Higrómetros
- Control de humedad de un recinto
- Velocímetros
- Tacómetros
- Medidores de RPM (revoluciones por minutos)
- Circuitos de protección magnética
- Altímetros
- Sistemas de control ambiental
- Indicadores de nivel
- Indicadores de fugas de fluidos
- Instrumentación médica
- Control neumático

Entre muchas otras que el estudiante creativo puede realizar.

Planteamiento del problema

La necesidad de complementar el equipo de medición electrónica y de señales (osciloscopio) en los laboratorios de E.S.I.M.E. Zacatenco, aunado a la importancia de actualizar dichos instrumentos, los cuales son ocupados uno por dos o más alumnos al mismo tiempo, impidiendo la realización en tiempo de prácticas y con ello, una adquisición de conocimientos lenta o incompleta.

Además, es necesario contar con un instrumento de medición especial, para cada una de las magnitudes físicas a medir y/o analizar (se debe contar con un termómetro para medir temperatura, un voltímetro para medir voltaje, un osciloscopio para analizar señales, etc.), generando así, gastos muy elevados para la institución.

Por otra parte, debido a los precios elevados de cada instrumento de medición, es casi imposible que un alumno pueda adquirir uno de ellos en su hogar, en consecuencia, es necesario buscar un equipo disponible dentro de la institución en horas extra clase.

Justificación del proyecto

Este proyecto, al ser un medidor virtual, reduce la posibilidad de descomponerse con el paso del tiempo o por mal uso, como lo hacen los medidores físicos, puede ser utilizado en escuela u hogar puesto que sólo es necesaria su instalación en computadora.

Posee una interfaz gráfica amigable, contiene los controles, botones, barras deslizables y pantalla virtuales similares a los de un osciloscopio físico, haciéndolos lo más parecidos posible, permitiendo al usuario una familiarización fácil y rápida con el funcionamiento y manejo del medidor virtual.

Además de ser instrumento de medición electrónica y de señales, este proyecto contiene una serie de sensores capaces de medir y graficar diferentes magnitudes físicas (temperatura, presión, humedad y densidad de flujo magnético), permitiendo observar y analizar su comportamiento en tiempo real. Por otra parte, debido a que se trata de una herramienta virtual, su precio es menor que el del osciloscopio físico, con la ventaja de que éste jamás se descompondrá y que puede ser actualizado o modificado si es necesario.

Objetivos

Objetivo general

Diseñar y construir un osciloscopio para PC, un medidor de diferentes magnitudes (temperatura, presión, humedad y densidad de flujo magnético), por medio de sensores, y un generador de funciones virtual. Mostrar las mediciones efectuadas a través de software, con adquisición de datos en tiempo real.

Objetivos específicos

Cumplir con las siguientes características:

- Interfaz gráfica de usuario amigable, fácil de utilizar y parecida a un osciloscopio de banco, con controles virtuales parecidos a los físicos.
- Dimensiones y peso razonables como para ser transportable (10 x 12 x 5 cm. aprox.).
- De bajo consumo energético.
- Estar dirigido a señales de control (Frecuencia < 5 KHz).

Capítulo 1:

Antecedentes

Históricos

1. Antecedentes históricos

1.1 La medición en la historia

Desde la aparición del ser humano en la tierra, la necesidad de explorar nuevos territorios en busca de mejores condiciones de vida, le llevó a tomar las primeras mediciones realizadas, las cuales estuvieron relacionadas con la masa, la longitud y el tiempo, para medir dichas magnitudes, se tomaba como referencia las jornadas solares y las medidas corporales (pies, brazas, palmas, etc.). [1] Desde hace 2,500 años A.C., se conocen en el cercano oriente algunos planos y mapas esbozados de las mediciones que en ese tiempo se realizaban.

Posteriormente, comenzaron las medidas de volumen y ángulo como una necesidad debido a las primeras construcciones realizadas por el hombre. De igual modo, en los intercambios comerciales donde el trueque suponía intercambiar unos productos por otros, era necesario conocer la cantidad exacta del producto que se pretendía intercambiar, así comenzaron las mediciones en los productos alimenticios y de objetos de valor como el oro y la plata. [2]

Con lo anterior, empezaron a surgir los primeros instrumentos de medición. Se estima que los comienzos de la balanza se remontan al año 5,000 A.C. En Mesopotamia y Egipto comienzan a utilizarse en el año 3,000 A.C. teniendo como valores, múltiplos de una unidad común: "el peso de un grano de trigo". Por otra parte, la ciencia griega, a partir del año 500 A.C. tuvo necesidad de instrumentos de precisión para determinar la pureza de metales preciosos. [2]

Aun así, con todo y los primeros instrumentos y herramientas de medición, medir resultaba algo muy complicado. Debido a que medir es simplemente comparar, y cada persona, cada pueblo, cada país comparaba las cosas que poseía, hubo gran conflicto pues resulta obvio que todos los seres humanos no tienen los pies ni las manos del mismo tamaño, ni todos los granos de trigo o maíz pesan lo mismo aun siendo de la misma mazorca. Entonces surgen los sistemas de medición. [1]

Los sistemas más raros de medición coexistían hasta la revolución francesa por el año 1789. En esta época de tumulto y grandes cambios, los franceses, decidieron fundar un sistema de mediciones racional y único que fuera superior a todos los demás. Entonces, la Asamblea Nacional francesa le encomendó en 1790 a la Academia de Ciencias que creara este nuevo sistema, el cual tenía que:

- a) Estar basado en cosas que permanecieran estables (constantes) en la naturaleza, que fueran lo mismo a pesar del lugar y del tiempo, y no en cosas variables o que dependieran de algo más.
- b) Estar basado en pocas formas de medir, que se conectaran unas con otras de manera lógica. Por ejemplo, una vez definido el centímetro, se define al litro como el volumen de algo que entra en un cubo de 10 cm de lado, y se define el kilogramo como el peso de un litro de agua.

- c) Ser un sistema decimal, es decir, donde los múltiplos de las unidades variaran de 10 en 10. Así, un decámetro es igual a 10 metros, un hectómetro es igual a 10 decámetros, y así sucesivamente. [2]

Una vez que se tuvieron bastantes instrumentos de medición para diferentes magnitudes y un estándar en cuanto a los objetos de comparación (unidades de medida), entonces surgió una nueva necesidad: observar la variación de diferentes *magnitudes*¹ con el paso del tiempo, es decir, su comportamiento.

El primer método de la historia para crear una imagen de una forma de onda, era a través de un minucioso y laborioso proceso, en donde se media la tensión o corriente de un *rotor*² giratorio en puntos específicos alrededor del eje del rotor, y se efectuaba a mano, trazando puntos de acuerdo a las mediciones efectuadas con un *galvanómetro*³, en intervalos de tiempo deseados.

El segundo método fue automático, de papel pintado por un oscilógrafo, utilizado por primera vez en un galvanómetro para mover una pluma, a través de un rollo de papel, la captura de los patrones de onda en el papel se realizaba con el movimiento continuo de la aguja del galvanómetro, mientras que la pluma que contenía adosada iba marcando sobre el rollo de papel, y a su vez este último giraba conforme el movimiento de la aguja, para poder trazar la onda y no marcar todo en una sola superficie. Debido a la relativamente alta frecuencia de las ondas, en comparación con el tiempo de reacción lenta de los componentes mecánicos, la imagen de forma de onda no se tomaba directamente, sino que se construía a lo largo de un período de tiempo mediante la combinación de pequeños trozos de muchas diferentes formas de onda, para crear un forma total promedio.

El tercer método se hizo con el desarrollo de la bobina móvil, el oscilógrafo inventado por el ingeniero británico electro-físico William Duddell, que en los tiempos modernos también se conoce como un espejo del galvanómetro. Esto redujo el dispositivo de medición a un pequeño espejo que podía moverse a gran velocidad, para que la imagen creada por la aguja del galvanómetro y el papel coincidiera con la forma de onda de la medición.

Con la invención del tubo de rayos catódicos (CRT) desarrollado en el siglo XIX. El físico alemán Karl Ferdinand Braun inventó el osciloscopio como curiosidad física en 1897, mediante la aplicación de una señal oscilante de *deflector*⁴ de placas con carga eléctrica en un fósforo recubierto de electrones, en ese momento, el invento estaba destinado principalmente para demostrar y explorar la física de los electrones

¹ *Magnitudes*: Cualidades de un sistema o cuerpo a las que puede asignársele un número por comparación con la unidad de medida que le corresponda. [Diccionario Encyclopédico Salvat, 1985, Vol. 17, Varios Autores, pg. 2378]

² *Rotor*: Parte giratoria de un motor eléctrico. [Diccionario Encyclopédico Salvat, 1985, Vol. 23, Varios Autores, pg. 3274]

³ *Galvanómetro*: Instrumento para medir intensidad de corriente, de elevada sensibilidad que generalmente requiere de una calibración previa a la medición. [Diccionario Encyclopédico Salvat, 1985, Vol. 12, Varios Autores, pg. 1710]

⁴ *Deflector*: Placa o superficie, de formas muy diversas, destinada a desviar la dirección de corriente en un fluido. [Diccionario Encyclopédico Salvat, 1985, Vol. 8, Varios Autores, pg. 1140]

(entonces conocidos como rayos catódicos), pero después se fueron ocupando para observar el comportamiento de algunos dispositivos electrónicos y circuitos complejos. [3]

1.2 El osciloscopio

En la historia de las mediciones eléctricas y electrónicas, el instrumento que ha producido mayor impacto ha sido el osciloscopio, debido a que es de utilidad no solamente para medir algunas variables de los fenómenos eléctricos (voltaje, corriente, frecuencia, período, etc.), sino que también permite observar el desarrollo en el tiempo de dichos fenómenos, con lo cual se pueden conocer y estudiar en forma más detallada. Debido a esto, el osciloscopio sigue siendo uno de los instrumentos de mayor aplicación en el trabajo diario. [3]

El Osciloscopio se convirtió en una herramienta mucho más útil y poderosa en 1946, cuando los ingenieros y científicos estadounidense Howard Vollum y Jack Murdock inventaron el osciloscopio de barrido disparado (El Tektronix modelo 511). [3]

1.3 Los instrumentos de medición de diferentes magnitudes físicas en la historia

1.3.1 Temperatura

Los primeros científicos entendían la diferencia entre “frío” y “caliente”, pero no tenían un método para *cuantificar*⁵ los diferentes grados de calor. En 1597, el astrónomo, filósofo, matemático y físico italiano Galileo Galilei inventó el primer instrumento para medir la temperatura, un simple termoscopio, que consistía en un largo tubo de cristal invertido en una jarra sellada que contenía agua y aire. Cuando la jarra era calentada, el aire se expandía y empujaba hacia arriba el líquido en el tubo. El nivel del agua en el tubo podía ser comparado a diferentes temperaturas, para mostrar los cambios relativos cuando se añadía o se retiraba calor, pero el termoscopio no permitía cuantificar la temperatura fácilmente.

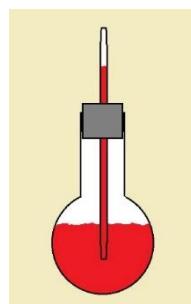


Fig. 1: Termoscopio.

⁵ Cuantificar: Expresar numéricamente el valor de una magnitud. [Diccionario Encyclopédico Salvat, 1985, Vol. 8, Varios Autores, pg. 1085]

Varios años después, el físico e inventor Italiano Santorio Santorio mejoró el diseño de Galileo añadiendo una escala numérica al termoscopio. Estos primeros termoscopios dieron paso al desarrollo de los termómetros usados hoy en día, que funcionan sobre la base de la tendencia de algunos líquidos a expandirse cuando se calientan.

Cuando el *fluido*⁶ dentro de un termómetro absorbe calor, se expande, ocupando un volumen mayor y forzando la subida del nivel del fluido dentro de un tubo, cuando el fluido se enfria, se contrae, ocupando un volumen menor y causando la caída del nivel de dicho fluido. [4]

1.3.2 Presión

El barómetro es el instrumento que sirve para medir la presión atmosférica, es decir, la fuerza por unidad de superficie que ejerce un líquido o un gas *perpendicularmente*⁷ a dicha superficie.

Los primeros barómetros estaban formados por una columna de mercurio encerrado en un tubo. El peso de la columna de líquido comprimido era exactamente el peso de la atmósfera. [5]

Desde las primeras versiones surgidas del barómetro con los experimentos del físico y matemático italiano Torricelli, y la observación por parte del matemático y filósofo francés Blaise Pascal de la relación entre la presión del aire atmosférico con la *altitud*⁸, este instrumento hizo uso de la tradicional columna de mercurio para la medición de la presión atmosférica. La utilización de este principio, que pareció ser suficiente por un espacio de dos siglos, comenzó a evidenciar ciertas desventajas a partir de la primera mitad del siglo XIX.

A las evidentes limitaciones relacionadas con la dificultad para transportar estos instrumentos, y la necesidad de realizar las lecturas en frágiles tubos de vidrio, se unió la altura de las columnas de mercurio requeridas. Mientras para presiones bajas bastaba con un diseño usual, para otras razonablemente altas, era necesario realizar modificaciones que incluían sistemas, formas o dimensiones diferentes, como por ejemplo las mostradas en la Fig. 2 que resultaban indispensables.

⁶ *Fluido*: Del cuerpo en el que las fuerzas de cohesión entre sus moléculas son poco intensas y estas pueden desplazarse unas respecto de otras, por lo que adopta la forma del recipiente que lo contiene. Son fluidos, por lo tanto, los líquidos y los gases. [Diccionario Enciclopédico Salvat, 1985, Vol. 12, Varios Autores, pg. 1627]

⁷ *Perpendicularmente*: Que se coloca con respecto a una referencia de tal manera que forman un ángulo de 90° entre sí. [Diccionario Enciclopédico Salvat, 1985, Vol. 21, Varios Autores, pg. 2946]

⁸ *Altitud*: Altura de un punto de la Tierra con respecto del nivel del mar. [Diccionario Enciclopédico Salvat, 1985, Vol. 1, Varios Autores, pg. 142]

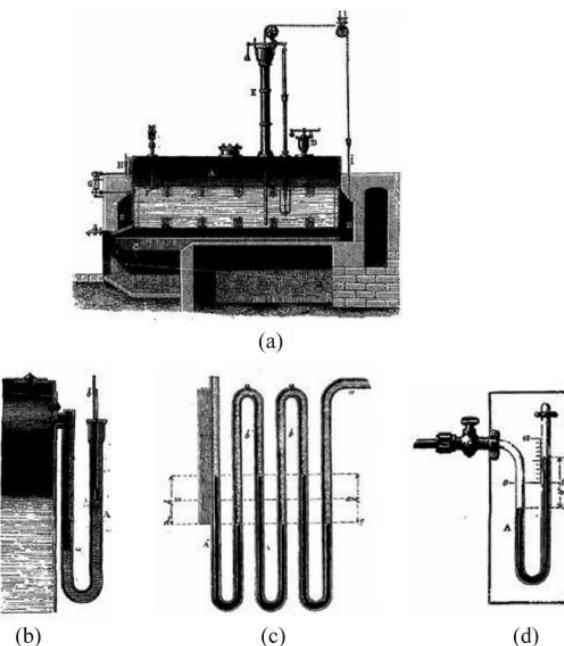


Fig. 2: Diseños diferentes de barómetros de acuerdo a la presión a medir. (a) De caldera tradicional. (b) De extremo abierto a bajas presiones. (c) De extremo abierto a altas presiones. (d) De extremo cerrado.
 [A. Armengaud, *Traité Théorique et Pratique des Moteurs à Vapeur*, v. I (Chez Lateur, Paris, 1861), pg. 256-264]

Diferentes desarrollos posteriores a los diseños originales, sólo lograron superar parcialmente las limitaciones mencionadas anteriormente. En 1749 el geólogo y meteorólogo suizo Jean André De Luc eliminó la necesidad de un depósito anexo de mercurio al construir el primer barómetro portátil de sifón⁹. [6]

Por más de un siglo fueron propuestas modificaciones de distinta índole, privilegiando la protección del equipo y, por ende, el mejoramiento de la portabilidad por encima de otras consideraciones, incluso de su misma *precisión*¹⁰. Otros diseños presentados en las primeras décadas del siglo XIX, incorporando la utilización de sustancias diferentes al mercurio, por ejemplo, ácido sulfúrico coloreado, resolvían parcialmente los problemas asociados con el tamaño y la dificultad de transporte, pero fueron rápidamente descontinuados, debido a la continua necesidad de *calibración*¹¹ en virtud de la *volatilidad*¹² de los líquidos utilizados. La modificación del sistema de tubo abierto por el uso de aire comprimido (Fig. 2d) eliminaba el problema de las elevadas dimensiones, pero reducía la precisión de las lecturas.

⁹ Sifón: Tubo doblado para el traspase de líquido ya sea entre dos recipientes o depósitos o para extraerlo de uno de ellos. Para hacerlo funcionar hay que cebarlo, es decir, llenar todo el tubo de líquido por aspiración desde un extremo. El flujo tiene lugar por la acción de la presión atmosférica. [Diccionario Encyclopédico Salvat, 1985, Vol. 24, Varios Autores, pg. 3397]

¹⁰ Precisión: fidelidad, concisión y puntualidad en la ejecución de algo, cercanía con lo real.
[\[http://es.thefreedictionary.com/precision\]](http://es.thefreedictionary.com/precision)
 Consulta 27/06/2013]

¹¹ Calibración: Determinar, por comparación con un patrón, el valor justo de la lectura de la escala de un instrumento de medida, dejándolo ajustado con la precisión deseada. [Diccionario Encyclopédico Salvat, 1985, Vol. 5, Varios Autores, pg. 664]

¹² Volatilidad: Tendencia de un cuerpo sólido o líquido a disiparse, a transformarse en vapor o gas. [Diccionario Encyclopédico Salvat, 1985, Vol. 26, Varios Autores, pg. 3705]

La adjudicación en 1844 de una patente británica (No. 10,157) al abogado y científico francés Lucien Vidie, por la invención del primer barómetro mecánico, cambió definitivamente la historia de los elementos de medición de presión [7]. Este nuevo instrumento hizo necesario el reemplazo del uso del mercurio por un disco metálico previamente evacuado.

Vidie, logró diseñar el llamado barómetro *aneroide*¹³. El instrumento principal descrito en la patente, consistía de una caja que contenía una capsula metálica sellada, fabricada originalmente en bronce, de la cual algo de aire había sido retirado, creando un vacío parcial, y provista de un diafragma corrugado sostenido por una serie de 33 resorte *helicoidales*¹⁴ mantenidos en su base, y un elemento interior encargado del registro. El principio de funcionamiento es el cambio en la altura de la cámara metálica, la cual posee superficies flexibles. [8]

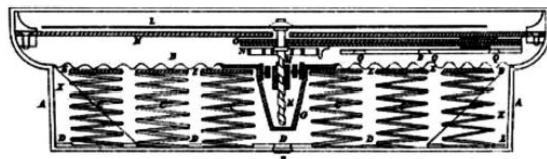


Fig. 3: Barómetro aneroide. [8]

En Junio 18 de 1849 el Ministerio de Agricultura y Comercio del Gobierno Francés le concedió a Bourdon una patente por su trabajo titulado: "Manómetro metálico, aplicable a barómetros y manómetros", en el cual Bourdon planteó los principios de los que consideraba dependía la acción del instrumento propuesto: las relaciones de proporcionalidad entre el espesor de un tubo aplanado dispuesto de manera circular y la variación de su radio de curvatura por un lado, y el desplazamiento de sus extremos y la magnitud de la presión aplicada por el otro. [9]

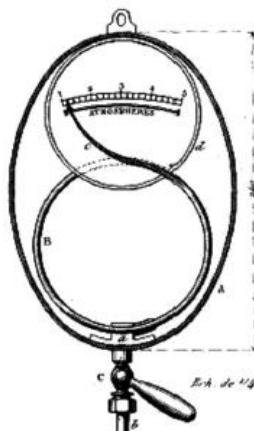


Fig. 4: Barómetro de Burdon.

[E. Bourdon, in Minutes of the Proceedings Institution of Civil Engineers 11, 1852.]

¹³ Aneroide: Que no contiene ningún fluido dentro. [http://etimologias.dechile.net/?aneroide, Consulta 27/06/2013]

¹⁴ Helicoidal: En forma de espiral. [Diccionario Encyclopédico Salvat, 1985, Vol. 14, Varios Autores, pg. 1906]

1.3.3 Humedad

Los primeros métodos de medición de *humedad relativa*¹⁵ se basaron en el uso de cabellos, ya que su longitud varía en función de la humedad ambiente, o los psicrómetros, en los cuales se medía la humedad relativa en función de la diferencia de temperaturas entre un termómetro que mide la temperatura ambiente y otro embebido en un algodón con agua destilada. Actualmente, los instrumentos modernos de medición de humedad se basan en dos tecnologías:

- Medición por espejo de *punto de rocío*¹⁶: En la cual se enfriá un espejo hasta que se alcanza la temperatura de punto de rocío. Presenta la ventaja de tener una muy alta exactitud, pero requiere de un gran cuidado debido a su fragilidad y alto costo, motivo por el cual se utiliza casi exclusivamente en mediciones de laboratorio. Es la referencia de todo laboratorio de calibración.
- Medición por sensor capacitivo: Por su alta exactitud, estabilidad a largo plazo, reducidas dimensiones y costo es la opción más utilizada en instrumentos de medición. Su tecnología de medición se basa en el cambio de la capacidad de un sensor en función de la humedad. Incorporan un sensor de temperatura para compensar los efectos de variación de capacidad por variación de temperatura. [10]

1.3.4 Densidad de flujo magnético

En octubre de 1879, el físico estadounidense Edwin Hall descubrió el efecto que lleva su nombre (efecto Hall). Encontró que si se aplica un campo magnético elevado a una fina lámina de oro por la que circula corriente, se produce un voltaje en la lámina perpendicularmente a como fluye la corriente, este voltaje se llama voltaje Hall.

La técnica de medición más utilizada para la determinación de los portadores de carga y resistividad en un semiconductor es la *técnica de Van Der Pauw*¹⁷, conocida también como técnica de cuatro puntas, siendo las pinzas de Hall las que utilizan dicha técnica para la medición del efecto. [11]

¹⁵ *Humedad relativa*: Es un porcentaje que indica que cantidad del total de vapor de agua está actualmente en el ambiente. Dado que la cantidad máxima posible de vapor de agua está en relación con la temperatura, siempre se relacionan ambos parámetros según su coeficiente porcentual. [10]

¹⁶ *Punto de rocío*: Es la temperatura a la cual, el vapor de agua presente en el aire se condensa. Si bien es una medida de humedad, se la expresa con unidades de temperatura. °Ctd o °Ctpd -bajo presión-. [10]

¹⁷ *Técnica de Van Der Pauw*: El objetivo en el experimento de Van Der Pauw es determinar la densidad de portadores de carga midiendo el voltaje de Hall (VH). Para medir la tensión Hall (VH), una corriente (I) es forzada a fluir través del par opuesto de contactos (puntas, esquinas) 1 y 3 y el voltaje VH es medido en el par de contactos cruzados restante 2 y 4. [http://www.uv.es/martined/tecweb/efecto_hall.pdf, Consulta 26/06/2013].



Fig. 5: Medidor de efecto Hall (pinzas Hall).

[<http://electricidad-viatger.blogspot.mx/2008/04/medida-de-la-intensidad-de-corriente-en.html>]

1.4 Plataforma de desarrollo visual

El origen de las plataformas de desarrollo visual, se remontan a Massachusetts Institute of Technology (MIT) durante la segunda guerra mundial, donde Vannevar Brush propuso una herramienta llamada “Memex”, la cual permitiría a la información ser almacenada de tal manera que se pudiera tener un acceso más fácil, ligable a hipervínculos y programable.

En 1963, Iván Sutherland, un estudiante de postgrado de MIT, creó Sketchpad un programa que permitía la manipulación de objetos gráficos en un monitor de *Tubo de Rayos Catódicos* (TRC), usando un lápiz táctil. Pero gran parte del progreso sucedió en el centro de investigación estadounidense Xerox de Palo Alto, donde se estableció un sistema en 1970, hasta que en 1984, el empresario estadounidense Steve Jobs, cofundador de la empresa Apple, fue inspirado fuertemente por los avances de Xerox y decidió incorporar estas tecnologías a sus computadoras, mientras que la empresa Microsoft lanzó una versión inicial de su sistema operativo con IGU, llamado Windows 1.0 en 1985. Actualizando ambas, su S.O. año con año. [12]

Capítulo 2:

Estado del Arte

2. Estado del arte

2.1 Tecnologías actuales usadas en osciloscopios

2.1.1 Visual Persistence Oscilloscope (VPO)

La tecnología de procesamiento digital llamada Osciloscopio de Persistencia Visual (VPO), permite desplegar formas de onda con varias escalas de grises basado en las frecuencias de ocurrencia, a diferencia de los osciloscopios analógicos. Puede obtener los valores de voltajes, corrientes y periodo o frecuencia. La alta velocidad de procesamiento de datos paralelo, proporciona el análisis de eventos rápidos como: video, *fluctuación*¹⁸, video juegos, etc. VPO se puede encontrar en el osciloscopio GDS 3000. [13]

2.1.2 Tecnología HDO4096

La tecnología de alta definición HD4096, aplicada a los osciloscopios HDO series 4000 y 6000, se basa en una arquitectura con Convertidores Analógico a Digital (ADC) de 12-bits de alta velocidad, amplificadores con alta relación señal-ruido y un sistema de bajo ruido. Esta tecnología permite a los osciloscopios de alta definición, capturar y visualizar señales de hasta 1Ghz con alta velocidad de muestreo y 16 veces más *resolución*¹⁹ que otros osciloscopios. [13]

2.1.3 Memoria inteligente MegaZoom IV

La tecnología personalizada MegaZoom IV, combina las características de un osciloscopio, un analizador lógico, un analizador de protocolos y un generador de funciones en un único equipo compacto. La cuarta generación de tecnología MegaZoom permite obtener las mayores velocidades de actualización de formas de onda, con adquisiciones de datos propias de una memoria profunda de alta respuesta. Ésta tecnología se encuentra en el osciloscopio InfiniiVision Serie 4000 X. [13]

2.1.4 SiGe

Esta tecnología, incluida en el LabMaster 10 Zi, está basada en el semiconductor compuesto por Silicio y Germanio (SiGe), con el cual es posible realizar componentes electrónicos altamente resistentes, tanto a grandes variaciones de temperatura como a las radiaciones en el espacio. [14]

¹⁸ Fluctuación: Oscilación, crecer y disminuir alternadamente. [Diccionario Enciclopédico Salvat, 1985, Vol. 12, Varios Autores, pg. 1627]

¹⁹ Resolución: Exactitud o claridad en la reproducción de una imagen. Parámetro que determina el número de pixeles (punto luminoso) en un área determinada. [<http://es.thefreedictionary.com/resoluci%C3%B3n>, Consulta 26/06/2013]

2.1.5 Rendimiento puramente hardware

La manera más difícil de obtener un *ancho de banda*²⁰ superior a 20 GHz es utilizando únicamente el rendimiento hardware. Para tener un rendimiento puramente hardware superior a 20 GHz, un fabricante de osciloscopios debe invertir en múltiples chips que sean compatibles con estos anchos de banda. El proceso requiere la utilización de transistores con una frecuencia de corte superior a 150 GHz. Este proceso es costoso y poco común, encontrado en el osciloscopio DSAX93204A. [15]

2.1.6 Digital band with interleave

Divide el ancho de banda en distintas partes para así poder realizar un mayor procesamiento de la señal ya sea *demultiplexando*²¹ o *multiplexando*²² la señal de entrada, éste proceso está contenido en el osciloscopio LabMaster 10 Zi. [13]

2.1.7 Entrelazado de frecuencia

A pesar de no ser ampliamente conocida, ha sido una técnica utilizada en el sector de radiofrecuencia (RF) durante muchos años. El entrelazado de frecuencia es diferente al entrelazado convencional de los Convertidores Analógico-Digital (ADC por sus siglas en inglés) utilizado por los fabricantes de osciloscopios. Todos los fabricantes de osciloscopios entrelazan actualmente recursos de canal, como la memoria y ADCs para obtener velocidades de muestreo más altas y mayor profundidad de memoria.

El entrelazado de frecuencia requiere la utilización de hardware adicional y un avanzado procesamiento digital de señal para obtener el rendimiento de ancho de banda óptimo. Desarrollando la técnica de entrelazado de frecuencia, los fabricantes de osciloscopios consiguen producir osciloscopios de mayor ancho de banda, sin tener que esperar al desarrollo de costosos chips de preamplificador. [15]

2.1.8 RealEdge

RealEdge, contenida en el Infiniium Serie 90000 Q, se compone de una combinación de nuevas arquitecturas, microcircuitos de última generación y de una aplicación avanzada del proceso de semiconductores de fosfuro de indio. Esta nueva tecnología hace posible la capacidad de alta frecuencia al tiempo, que mantiene los niveles de ruido y medida de fluctuación bajos. [13]

²⁰ Ancho de Banda: Cantidad de información o datos que se puede enviar a través de una conexión en un periodo de tiempo dado. [<http://www.masadelante.com/faqs/ancho-de-banda>, Consulta 27/06/2013]

²¹ Demultiplexar: Separar dos o más señales que fueron previamente combinadas y transmitidas a través de un canal. [Diccionario Enciclopédico Salvat, 1985, Vol. 8, Varios Autores, pg. 1150]

²² Multiplexar: Combinar dos o más señales para ser transmitidas. [Diccionario Enciclopédico Salvat, 1985, Vol. 19, Varios Autores, pg. 2627]

2.1.9 DSP potenciado

El Procesador Digital de Señales (DSP por sus siglas en inglés) potenciado es una técnica de procesamiento donde el contenido de alta frecuencia de un osciloscopio se potencia mediante software, usando este es posible amplificar los componentes de mayor frecuencia de la señal. [15]

2.2 Herramientas de desarrollo

2.2.1 InfiniiView N8900A

Primera aplicación que migra la *interfaz*²³ de usuario completa de los osciloscopios de tiempo real a PC, para acelerar la navegación y obtener una visualización y unos análisis más potentes. Ofrece una resolución de pantalla de hasta 1680 x 1200 pixeles (px), para que la visualización de las formas de onda en monitores grandes sea mejor.

Esta aplicación incorpora ventanas acopiables independientes para medidas de tiempo, frecuencia y protocolo, con el fin de crear vistas personalizables por el usuario, contiene licencias transportables y basadas en servidor para que compartir resulte más sencillo. Contiene marcadores inteligentes que actualizan dinámicamente los valores con los ejes anotados para acelerar la *depuración*²⁴. [16]

2.2.2 Simulink

Es un entorno de diagrama de bloques para simulación multidominio y diseño basado en modelos. Es compatible con el sistema de nivel de diseño, simulación, generación automática de código y prueba continua y verificación de *sistemas embebidos*²⁵.

Ofrece un editor gráfico, las bibliotecas de bloques personalizables y solucionadores para el modelado y simulación de sistemas dinámicos. Está integrado con MATLAB, y así le permite incorporar algoritmos de MATLAB en modelos y exportar los resultados de simulación para MATLAB para análisis posteriores. [17]

²³ *Interfaz*: Medio de conexión de dos o más elementos diseñado para facilitar la comunicación entre ambos. [Diccionario Encyclopédico Salvat, 1985, Vol. 15, Varios Autores, pg. 2085]

²⁴ *Depuración*: Proceso en el que se traducen las instrucciones de un programa escrito (lenguaje fuente) a código máquina. [<http://es.thefreedictionary.com/compilaci%C3%B3n>, Consulta 27/06/2013]

²⁵ *Sistemas Embebidos*: Sistemas completos, complejos, que contiene dentro de sí varios sistemas separables. [<http://www.mastermagazine.info/termino/6694.php>, Consulta 27/06/2013]

2.2.3 LabView

El software LabView es ideal para cualquier sistema de medidas y control. Integra todas las herramientas que los ingenieros y científicos necesitan para construir una amplia variedad de aplicaciones en poco tiempo. NI LabView es un entorno de desarrollo para resolver problemas, productividad acelerada y constante innovación.

Es una plataforma de programación gráfica que ayuda a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes. Ofrece herramientas para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente. [18]

2.2.4 Matlab

Es un lenguaje de alto nivel y un entorno interactivo para cálculo numérico, visualización y programación. Con MATLAB se puede: analizar datos, desarrollar algoritmos, y crear modelos y aplicaciones. El lenguaje, las herramientas y funciones incorporadas de matemáticas le permiten explorar múltiples enfoques y llegar a una solución más rápida, que con las hojas de cálculo o lenguajes de programación tradicionales (C/C++ o Java).

Se puede utilizar MATLAB para una gama de aplicaciones, incluyendo el procesamiento de señales y comunicaciones, procesamiento de imágenes y vídeo, sistemas de control, prueba y medición. [16]

2.2.5 Visual C#

Microsoft Visual C# es un lenguaje de programación diseñado para crear una amplia gama de aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Con sus diversas innovaciones, C# permite desarrollar aplicaciones rápidamente.

Visual Studio admite Visual C# con un editor de código completo, plantillas de proyecto, diseñadores, asistentes para código, un depurador eficaz y fácil de usar, además de otras herramientas. La biblioteca de clases .NET Framework ofrece acceso a una amplia gama de servicios de sistema operativo y a otras clases útiles y adecuadamente diseñadas que aceleran el ciclo de desarrollo de manera significativa. [19]

2.2.6 DSP System Toolbox

Proporciona algoritmos para el diseño y la simulación de sistemas de procesamiento de señales. La caja de herramientas del sistema incluye métodos de diseño especializado, técnicas para procesar el flujo de datos y la creación de prototipos en tiempo real. Se puede: generar señales, analizar espectros y visualizar de forma interactiva el comportamiento y rendimiento del sistema. [20]

2.2.7 Code Warrior

Es una herramienta basada en un Entorno de Desarrollo Integrado (IDE). Desarrollado por Metrowerks, CodeWarrior integra edición de archivos, compilación (depuración) optimizada, simulación de código y programación de diferentes dispositivos.

Este programa trae la versatilidad de programar en lenguaje ensamblador, C, C++, e incluso permite combinaciones de código en los tres tipos mencionados. [21]

2.2.8 MPLAB X

El IDE MPLAB X es una herramienta gráfica, con depuración integrada creada para todos los más de 800 microcontroladores de 8-bit, 16-bit y 32-bit, controladores de señales digitales y dispositivos de memoria. Incluye un editor rico en funciones, un depurador a nivel de fuente, director del proyecto, simulador de software, y es compatible con populares herramientas de Microchip de hardware. Basado en la plataforma de código abierto NetBeans, MPLAB X se ejecuta en Windows OS, el sistema operativo Mac y Linux, soporta muchas herramientas de terceros. [23]

2.3 Estudio del mercado

2.3.1 Osciloscopios de banco

2.3.1.1 Osciloscopio GDS3000



Fig. 6: Osciloscopio GDS3000. [24]

Características:

- Precio de \$5014 dlls.
- 500/350/250/150MHz de 2 a 4 Canales.
- Muestreo desde 5 G muestras/s a 100G muestras/s.
- Memoria independiente por canal.
- Tecnología VPO (Persistencia Visual).
- Display de 800x600.

- Función de división de pantalla.
- *Impedancias*²⁶ de entrada ($50\Omega/75\Omega/1M\Omega$).
- Software de análisis de potencia.
- Análisis de bus serie para I2C, SPI y UART. [24]

2.3.1.2 InfiniiVision series 3000x y series 4000x



Fig. 7: Osciloscopio InfiniiVision 3000x series. [25]

Características InfiniiVision Series 3000x:

- Costo: \$5014 dlls.
- 2 a 4 Canales analógicos.
- Ancho de banda ampliable 70MHz, 100MHz, 200MHz, 1GHz ó 1.5GHz.
- 8 canales digitales.
- Muestreo máximo de 2 G muestras/s.
- Velocidad de actualización de onda > 50000 ondas por segundo.
- Display táctil capacitivo de 8.5 pulgadas.
- Función de canal simple.
- Voltímetro digital integrado.
- Análisis de protocolo serial (CAN, I2C, LIN, SPI, UART/RS232).
- Tecnología VPO. [25]

Características InfiniiVision Series 4000x:

- Costo: \$6006 dlls.
- 2 a 4 Canales analógicos.
- Ancho de banda ampliable 200MHz, 350MHz, 500 MHz, 1GHz, 1.5GHz.
- 16 canales digitales.
- Muestreo máximo de 5 G muestras/s.
- Velocidad de actualización de onda > 1M ondas por segundo.
- Display táctil capacitivo de 12.1 pulgadas.
- Zona de disparo táctil estándar.

²⁶ Impedancia: Resistencia aparente de un circuito eléctrico al paso de la corriente alterna.
[<http://www.wordreference.com/definicion/impedancia>, Consulta 47/05/2013]

- Voltímetro digital integrado.
- Búsqueda y navegación.
- Análisis de protocolo serial (ARINC 429, CAN, FlexRay, I2C, I2S, LIN, MIL-STD-1553, SPI, UART/RS232, USB 2.0).
- Memoria segmentada estándar.
- Memoria inteligente MEGAZOOM4.
- Análisis de potencia.
- Prueba de calidad de señal USB 2.0.
- Análisis de TV de alta definición.
- Conectividad USB2.0, LAN, salida de video. [25]

2.3.1.3 LeCroyWaveSurfer serie B



Fig. 8: Osciloscopio LeCroyWaveSurfer B. [26]

Características:

- Costo: \$11,499.99 dls.
- Memoria amplia y una alta velocidad de muestreo.
- Anchos de banda entre los 200 MHz a 1 GHz.
- Interfaz de pantalla táctil de 10,4 pulgadas.
- Cinco puertos USB.
- Velocidad de muestreo de hasta 10 G muestras/s.
- Memoria de hasta 25 Mpts.
- Modo secuencial que permite al usuario segmentar la memoria, con adquisiciones combinadas hasta 10,000 segmentos, optimizando el uso de la memoria tan solo haciendo las capturas cuando se presentan los datos.
- La función “WaveScan” reduce el tiempo de detección de problemas.
- La función “WaveStream” facilita la medida de anomalías.
- La función “LabNotebook” permite la captura de imágenes para ser enviado directamente por email o almacenarlo con formato “.doc”, “.pdf” o “HTML”.
- 4 canales analógicos y 18 canales digitales que cubren anchos de banda analógicos hasta 1 GHz y digitales hasta 250 MHz y con velocidades de muestreo de 1 G muestra/s que posibilitan la captura a altas velocidades de señales digitales. [26]

2.3.1.4 Osciloscopios HDO4104



Fig. 9: Osciloscopio HDO4104. [27]

Características:

- Costo: \$16,399.96 dlls.
- Osciloscopio digital.
- Ancho de banda $\leq 1\text{GHz}$.
- 4 canales.
- Longitud de record de memoria 12,5 Mpts/canal.
- Muestreo de hasta 2,5 G muestras/s.
- Display LCD 12,1 pulgadas de pantalla táctil.
- Tensión de entrada máxima de 400V.
- Impedancia de entrada $1\text{M}\Omega/15\text{pF}$.
- Dimensiones (mm): 292 x 400 x 131. [27]

2.3.1.5 Infinium serie 90000 Q



Fig. 10: Osciloscopio Infinium 90000 Q. [28]

Características:

- Costo: \$286,000 dlls.
- Ancho de banda de 63 GHz a 3 dB.
- Tasa de muestreo de 2 canales de 160 G muestras/s.
- Tasa de muestreo de 4 canales de 80 G muestras/s.

- Tiempo de subida y bajada 7 ps.
- Impedancia de entrada: $50\Omega \pm 3\%$.
- Acoplamiento de CD.
- Sensibilidad de 1 mV/div a 1 V/div.
- Resolución vertical de 8 a 12 bits. [28]

2.3.1.6 LabMaster 10 Zi



Fig. 11: Osciloscopio LabMaster 10 Zi. [29]

Características:

- Costo: \$252,900 dlls.
- Herramientas Matemáticas.
- Ancho de banda 65 GHz.
- Impedancia de entrada de $50\Omega \pm 2\%$.
- 80 canales de entrada.
- Resolución vertical de 8 bits.
- Rango de offset- $\pm 500\text{mV}$
- Tasa de muestreo de 80 G muestras/s por canal.
- Para señales repetitivas 200 G muestras/s. [29]

2.3.2 Osciloscopios con interfaz gráfica

2.3.2.1 Osciloscopio USB con interfaz gráfica en LabView

Es un osciloscopio de un canal que está compuesto por tres elementos:

1. Una fuente de alimentación de +15 y +32 Voltios
2. Un circuito acondicionador de señal (CAS)
3. Interfaz gráfica en Labview

Funcionamiento:

Después de que el dispositivo ha sido conectado y reconocido por la PC, se lee desde el CAS la tensión de referencia y el valor de tensión de la señal, se almacena el dato del ADC en una variable entera, la cual es transformada para ser enviada a la Computadora.

La interfaz gráfica (realizada en Labview 2009), consta de tres paneles con los cuales se configuran y ordenan todos los datos para su análisis y graficación, estos paneles son: el panel de control, el panel de recepción de datos y el de control de gráfica. [30]



Fig. 12: Panel de Control. [30]



Fig. 13: Panel de recepción de datos. [30]



Fig. 14: Panel de control de gráfica. [30]



Fig. 15: Interfaz gráfica, osciloscopio USB con Labview. [30]

2.3.2.2 PicoScope 2104 y PicoScope 2105

Osciloscopio de un canal con interfaz gráfica de usuario aplicable a computadoras de escritorio, laptop y mini-laptop con entrada USB 2.0 y compatible con USB 1.1, gráficas obtenidas en tiempo real.



Fig. 16: PicoScope 2104. [31]

Características PicoScope 2104:

- Precio de \$206.25 dlls.
- Ancho de banda de 10 MHz.
- 50 M muestras/s de tasa de muestreo en señales simples.
- 1 G muestras/s de tasa de muestreo en señales repetitivas.
- Rango dinámico de 48 dB.
- 8,000 muestras de tamaño de buffer.
- Precisión de $\pm 3\%$.
- Temperatura de operación entre 0 y 45 °C.
- Dimensiones (mm): 220 X 3 X 32.
- Requerimientos:
 - Memoria: 32 MB mínimo.
 - Espacio en HD: 10 MB mínimo.
 - Sistema Operativo: Microsoft Windows ME, 2000, XP, Vista y 7. [31]

Características PicoScope 2105:

- Precio de \$328.35 dlls.
- Ancho de banda de 25 MHz.
- 100 M muestras/s de tasa de muestreo en señales simples.
- 2 G muestras/s de tasa de muestreo en señales repetitivas.
- Rango dinámico de 48 dB.
- 24,000 muestras de tamaño de buffer.
- Precisión de $\pm 3\%$.
- Temperatura de operación entre 0 y 45 °C.
- Dimensiones (mm): 220 X 3 X 32.
- Requerimientos:
 - Memoria: 32 MB mínimo.
 - Espacio en HD: 10 MB mínimo.
 - Sistema Operativo: Microsoft Windows ME, 2000, XP, Vista, 7. [31]

2.3.2.3 Hantek DSO2090

Es un osciloscopio digital USB de dos canales, liviano y portable, no necesita alimentación externa, se alimenta directamente por USB lo que permite ser utilizado para trabajo de campo. Utiliza la PC como interfaz gráfica. Alta performance, 100 M muestras/s muestreo en tiempo real y ancho de banda de 40 MHz. Más de 23 funciones de medición. [32]



Fig. 17: Hantek DSO2090. [32]

Características:

- Precio de \$147.54 dlls.
- Interface USB 2.0
- Adecuado para usar con cualquier computadora de escritorio o laptop.
- Fácil de transportar.
- Dimensiones (mm): 190 X 100 X 35.
- Sistema operativo: Windows 98, Windows ME, Windows NT, Windows 2000, Windows XP, Vista y Windows 7.
- 23 funciones de medición.
- Promedio de forma de onda, persistencia, intensidad, inversión, adición, sustracción, multiplicación, división, grafico X-Y.
- Permite guardar la forma de onda como: Archivo de texto, archivo gráfico JPG/BMP, Excel o Word.
- Pueden conectarse más de un equipo a la misma PC extendiendo fácilmente la cantidad de canales.

2.3.2.4 Osciloscopio registrador para PC SU1000

Es un osciloscopio de dos canales, una entrada de disparo (trigger) externa y una cuota de medición en tiempo real de hasta 50 M muestras/s. La memoria por canal es de 4,000 muestras. El software ofrece, además de la función osciloscopio, un analizador de espectro y un registrador transitorio. También tiene a disposición funciones matemáticas para sumar o restar canales e invertir el canal 2. Además se le pueden mostrar directamente los 21 parámetros de las formas de onda.



Fig. 18: Osciloscopio SU1000. [33]

Características:

- Precio de \$166.47 dlls.
- Interfaz USB 1.1 y 2.0
- Dos canales de entrada.
- Alimentación y transmisión de datos únicamente mediante USB.
- Ancho de banda de 60 MHz.
- Acoplamientos de entrada por Condensador (Capacitor) de Acoplamiento (AC) y Acoplamiento Directo (CD).
- Tensión máxima de entrada de 30 V pico-pico.
- Dimensiones (mm): 55 x 205 x 175. [33]

2.3.2.5 Hantek DSO3064

Es un osciloscopio USB de 4 canales con un 200 M muestras/s velocidad de muestreo real, un ancho de banda de 60MHz con una resolución de 8 bits. Diseñado para probar una señal general y aplicaciones de automotores. El DSO3064 tiene un tamaño pequeño y bajo consumo de energía y alta fiabilidad. Con interfaz gráfica LabView.



Fig. 19: Hantek DSO3064. [34]

Características:

- Precio de \$465.46 dlls.
- 4 canales.
- En tiempo real 200 M muestras/s velocidad de muestreo.
- 10k-16M de memoria por canal de profundidad.
- Ancho de banda de 60 MHz.
- Contador de frecuencia, análisis de espectro FFT.
- USB 2.0 plug and play de interfaz, LAN y WiFi opcional.

- Más de 20 tipos de funciones de medición automática.
- Interfaz similar con un osciloscopio de banco.
- El soporte de software: Windows NT, Windows 2000, Windows XP, Vista y Windows 7.
- Se suministra con el código DEMO (VC, VB, LabView) y soporte técnico. [34]

2.3.2.6 Fluke 190 series II ScopeMeter Test Tool

Osciloscopio de dos canales diseñado para entornos industriales CAT 4. Amplias capacidades de resolución de problemas mediante la inspección visual de amplitud de señales, tiempo, forma y características de perturbación o distorsión.

La serie 190 de Fluke combina la potencia de un osciloscopio de alto rendimiento y un multímetro. Combina los más altos niveles de seguridad y portabilidad resistente a las altas prestaciones de un osciloscopio de banco. Este instrumento de medida incluye un Multímetro, TrendPlot y ScopeRecord (grabación en memoria) y operación manos libres con Connect-and-View.



Fig. 20: Fluke 190. [35]

Características:

- Precio de \$5,349.45 dlls.
- 2 canales.
- 500 MHz de ancho de banda.
- Frecuencia de muestreo de hasta 5 G muestras/s.
- Memoria de 10,000 muestras por canal.
- Captura de datos de hasta 48 horas seguidas.
- Alimentación por cargador externo.
- Adquisición de datos vía USB.
- Temperatura de operación de 0 a 45 °C.
- Dimensiones (mm): 270 X 190 X 170. [35]

Capítulo 3:

Marco

Teórico

3. Marco teórico

3.1 El osciloscopio

El osciloscopio es un instrumento que permite visualizar fenómenos transitorios (variantes en el tiempo) así como formas de ondas en circuitos eléctricos y electrónicos. [36]

Partes fundamentales de un Osciloscopio:

1. El tubo de rayos catódicos (TRC).
2. El amplificador vertical.
3. La base de tiempo.
4. El circuito de disparo (Trigger).
5. El amplificador horizontal.
6. El amplificador de control de intensidad (Gate amplifier).
7. La línea de retardo.

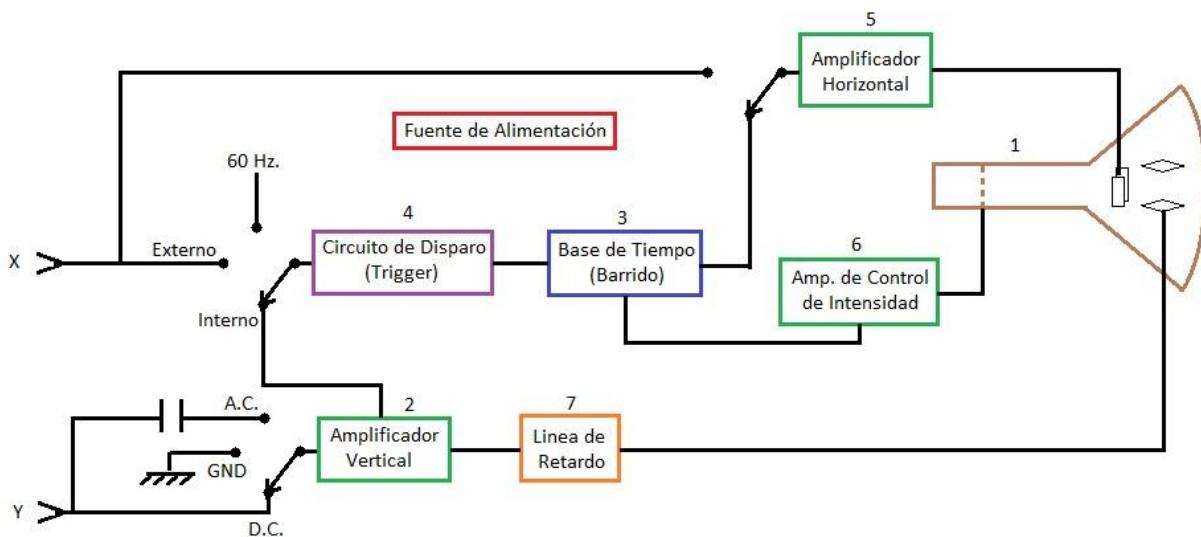


Fig. 21: Diagrama a bloques de un osciloscopio.

En un osciloscopio existen, básicamente, dos tipos de controles que son utilizados como reguladores, los cuales ajustan la señal de entrada y permiten medir en la pantalla, de esta manera se puede ver la forma de la señal medida. El primer control regula el eje X (horizontal) y aprecia fracciones de tiempo (segundos, milisegundos, microsegundos, etc.), el segundo regula el eje Y (vertical), controlando la tensión de entrada (en voltios, milivoltios, microvoltios, etc.). Estas regulaciones determinan el valor de la escala cuadricular que divide la pantalla, permitiendo saber cuánto representa cada cuadrado de ésta para conocer el valor de la señal a medir, tanto en tensión como el periodo. [37]

En los osciloscopios existen tres tipos de mecanismos de disparo (trigger) según la finalidad con la que se utilice. Estos son:

- Mecanismo de disparo básico: Este es el trigger que usan los osciloscopios analógicos para poder mantener fija las imágenes que se muestra en la pantalla.
- Mecanismo de disparo por detección de transitorios: Utilizado por los osciloscopios digitales (con memoria) para capturar eventos *anómalos*²⁷ de una señal y desplegar la forma de la señal en el momento en que estos ocurren.
- Disparo externo: Este es el mecanismo de disparo que permite observar lo que ocurre en una de las puntas del osciloscopio cuando llega un disparo (pulso, transitorio, etc) en la otra punta.

3.1.1 Métodos de muestreo

3.1.1.1 Tiempo real

- Método ideal para $2f_s < f_m$ (f_s = freq. señal , f_m = freq max del ADC).
- Única forma de capturar transitorios.
- Interpolación:
 - Lineal: unir puntos con líneas.
 - Senoidal: $\frac{\sin(x)}{x}$ preferido para $3f_s < f_m < 5f_s$

Este método de muestreo permite digitalizar señales no periódicas y transitorias. Cada muestra y el tiempo en que fue tomada tienen una correspondencia directa con su equivalente en tiempo real. A mayor tasa de muestreo en comparación al ancho de banda de la señal, se obtiene una mayor definición en el resultado.

El ingeniero eléctrico sueco Harry Nyquist desarrolló un teorema, que dice que para reconstruir una señal de frecuencia f_s , se debe muestrear a un índice mayor a $2f_s$. Sin embargo, esta teoría se aplica solamente a señales de ancho de banda limitado que no contienen ningún componente sobre la frecuencia f_m , y los bordes rápidos de las señales encontradas en circuitos digitales de alta velocidad, pueden contener armónicos significativos sobre sus frecuencias fundamentales. Más aún, cuando uno muestrea una señal, no lo hace por un tiempo infinito, sino que esta señal se ve acotada en el tiempo. Al recortar esta señal se le agregan componentes de frecuencia más altas, por lo tanto, no se puede muestrear sólo al doble de la frecuencia f_m , sino que hay que hacerlo a una frecuencia mayor. Por lo tanto, no importa lo rápido que se muestree, nunca se podrá recomponer esta señal a la perfección. [38]

²⁷ Anómalo: Irregular, extraño. [Diccionario Encyclopédico Salvat, 1985, Vol. 2, Varios Autores, pg. 215]

3.1.1.2 Tiempo equivalente

- Única forma para $2 f_s > f_m$.
- Solo para señales periódicas.
- La señal se va construyendo en barridos sucesivos.
- 3 tipos de barridos:
 - Aleatorio repetitivo
 - Secuencial
 - Submuestreo

3.1.1.3 Muestreo aleatorio repetitivo

Este tipo de muestreo se utiliza para aumentar la frecuencia máxima de medición de la señal de entrada, captura datos sobre la forma de onda adquiriendo puntos en más de una ocurrencia del trigger. Esto significa que la forma de onda en sí misma debe ser periódica y no un acontecimiento transitorio, puesto que la tasa de muestreo en estos casos es típicamente demasiado baja para señales transitivas de alta frecuencia.

En cada ocurrencia del disparador se adquieren algunos puntos de referencia, luego todos los puntos muestreados se juntan en un cuadro compuesto de la forma de onda. Cada punto es puesto en su lugar apropiado midiendo el tiempo transcurrido entre el trigger y la propia muestra.

Ofrece mayor ancho de banda que el muestreo en tiempo real, sin embargo, este método no es apto para mediciones de señales no periódicas y de alta velocidad. [39]

3.1.1.4 Muestreo secuencial

Este método captura una muestra por ciclo, pero con un determinado tiempo, entre el disparador y el punto de captura. Para este tipo de muestreo es necesaria una señal periódica. Cada muestra es tomada pasado cierto intervalo de tiempo luego de disparado el trigger. Para la siguiente captura, el intervalo de espera es incrementado, y por lo tanto dicha muestra va a ser tomada un instante de tiempo después en el ciclo que la muestra anterior.

Los puntos son tomados siempre en puntos diferentes del ciclo de la señal de entrada, sin importar en qué ciclo fue tomada la muestra. De este modo, al finalizar la captura de todas las muestras, cada una de ellas es posicionada en un único ciclo, pero en la posición que le corresponde a partir del tiempo transcurrido desde el trigger.

Ofrece el mayor ancho de banda disponible (50 GHz), muy bajo ruido, pero no puede capturar eventos previos al trigger e incluso pasado un mínimo tiempo tras él y no puede medir transitorios. [39]

3.1.1.5 Submuestreo

Se conoce que el teorema de Nyquist dice que la frecuencia de muestreo tiene que ser mayor al doble del ancho de banda de la señal a muestrear. Sin embargo se suele confundir “ancho de banda” con “frecuencia más alta”.

Una señal contenida en los 100 KHz, puede tener un ancho de banda de 1 KHz. De este modo, según el teorema de Nyquist bien aplicado, se podría muestrear dicha señal a una frecuencia mayor a 2 KHz (la que es mucho menor a 200 KHz). Entonces, al de ancho de banda acotado, y centrado en una frecuencia alta, se pasa por filtros para estar seguros de que evitamos superposición y así, tenemos un resultado fiable de la señal muestreada.

Este método permite medir señales de alta frecuencia con conversores de baja frecuencia, sin embargo, es necesario un filtrado de la señal para evitar superposición, y no es apto para medición de grandes anchos de banda. [39]

3.2 Magnitudes físicas

3.2.1 Temperatura

La temperatura es la medida de la cantidad de energía térmica poseída por un objeto. Hay tres escalas usadas actualmente para medir la temperatura: la escala Fahrenheit ($^{\circ}\text{F}$), la escala Celsius ($^{\circ}\text{C}$), y la escala Kelvin (K). Cada una de estas usa una serie de divisiones basadas en diferentes puntos de referencia, siendo el Kelvin el más usado en la ciencia.

El Kelvin (antes llamado grado Kelvin), es la unidad de temperatura de la escala creada por el físico y matemático británico William Thomson Lord Kelvin, en 1848. El incremento en un grado Celsius coincide con el de un Kelvin, su importancia radica en el 0: la temperatura de 0 K es denominada “cero absoluto” y corresponde al punto en el que las moléculas y átomos de un sistema tienen la mínima energía térmica posible. A la temperatura medida en Kelvin se le llama “temperatura absoluta”. [40]

Conversión de escalas:

- De Fahrenheit a Kelvin: $K = \frac{5}{9}(\text{ }^{\circ}\text{F} - 32) + 273.15$
- De Kelvin a Fahrenheit: $\text{ }^{\circ}\text{F} = \frac{9}{5}(K - 273.15) + 32$
- De Celsius a Kelvin: $K = \text{ }^{\circ}\text{C} + 273.15$
- De Kelvin a Celsius: $\text{ }^{\circ}\text{C} = K - 273.15$

3.2.2 Presión

La presión (P) es una magnitud física que mide la fuerza en dirección perpendicular por unidad de superficie, y sirve para caracterizar como se aplica una determinada fuerza resultante sobre una superficie.

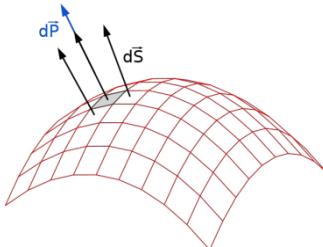


Fig. 22: Esquema de presión resultante a causa de una fuerza aplicada a una superficie. [4]

Cuando sobre una superficie plana de área A se aplica una fuerza normal (perpendicular) F de manera uniforme, la presión P viene dada por:

$$P = \frac{F}{A}$$

En un caso general donde la fuerza puede tener cualquier dirección y no estar distribuida uniformemente en cada punto, la presión se define como:

$$P = \frac{dF}{dA} n$$

Donde n es un vector unitario y normal a la superficie en el punto donde se pretende medir la presión.

En el Sistema Internacional la presión se mide en una unidad derivada que se denomina pascal (Pa), que es equivalente a una fuerza total de un Newton actuando uniformemente en un metro cuadrado de superficie $\left[\frac{N}{m^2}\right]$.

En determinadas aplicaciones la presión se mide no como la presión absoluta sino como la presión por encima de la presión atmosférica, denominándose presión relativa, presión normal, presión de gauge o presión manométrica. Consecuentemente, la presión absoluta es la presión atmosférica más la presión relativa. [4]

La presión atmosférica media es de 101,325 pascales, a nivel del mar, donde:

$$1 \text{ Atm} = 101,325 \text{ Pa} = 9.81 \text{ KPa.}$$

3.2.3 Humedad

La humedad es un fenómeno natural, que se presenta a nivel molecular y se encuentra relacionada con la cantidad de moléculas de agua presentes en una determinada sustancia, la cual puede estar en estado sólido o gaseoso. Si bien el grado de concentración de agua en el ambiente, no influye mayormente sobre la vida normal de un ser humano, resulta relevante para ciertos procesos, ya sean químicos, físicos o biológicos. Por ello, existe una necesidad de medirla, el dispositivo capaz de medir la humedad es llamado higrómetro o psicrómetro.^[51] La humedad puede medirse de tres formas:

- Humedad absoluta (H_{abs}): Es la razón entre la masa de agua presente en una mezcla, y el volumen de la misma.

$$H_{abs} = \frac{m_{H_2O}}{V_{mezcla}} \quad \left[\frac{Kg}{m^3} \right]$$

- Humedad específica (H): Es la razón entre la masa de agua y la masa de sustancia seca presente en la mezcla

$$H = \frac{m_{H_2O}}{m_{Sustancia\ seca}}$$

- Humedad relativa (H_r): Es la razón entre la presión parcial²⁸ de vapor actual, y la presión de vapor de saturación²⁹, expresada generalmente en porcentaje. O de otra manera, es la cantidad de agua que contiene un gas expresada en tanto por ciento de la cantidad que el gas tendría en estado de saturación, a la misma temperatura y presión absoluta.^[51]

$$H_r = \frac{P_v}{P_{Sat}} * 100\%$$

3.2.4 Densidad de flujo magnético

El efecto Hall, descubierto por Edwin Hall en 1879, consiste en la producción de una caída de voltaje a través de un conductor o semiconductor con corriente, bajo la influencia de un campo magnético externo. Para esto es necesario que la dirección del campo magnético sea perpendicular a la dirección de flujo de la corriente.

El campo magnético transversal (perpendicular) ejerce una fuerza desviadora (Fuerza de Lorentz) sobre el conductor o semiconductor. Esta fuerza causa la desviación de los portadores de carga que se mueven a través del material. Como resultado, aparece una diferencia de potencial V_{xy} (denominada

²⁸ Presión parcial de vapor: Es la fracción de presión ambiental debido a la presencia de vapor de agua en el aire, se mide en [Pascal] lo que equivale a $\left[\frac{N}{m^2} \right]$.

²⁹ Presión de vapor de saturación: Es la máxima presión de vapor que un gas puede soportar, a una cierta temperatura y presión atmosférica, a partir de la cual el vapor se comienza a condensar (cambia a estado líquido), y se mide en [Pa].

voltaje de Hall) entre los extremos del conductor. Este voltaje es proporcional a la intensidad del campo magnético aplicado y su polaridad depende del signo de los portadores de carga. El efecto Hall se presenta en conductores y en semiconductores. [41]

3.3 Sensores

3.3.1 Temperatura

3.3.1.1 Termopar

El termopar es el sensor de temperatura más usado, por su amplio intervalo de temperatura, su robustez, la relativa buena exactitud, rápida respuesta a cambios de temperatura, versatilidad de uso y bajo costo.

En 1831, Thomas Seebeck, descubrió que una corriente eléctrica fluía en un circuito cerrado de dos metales distintos, cuando una de las uniones era calentada como se muestra en la Fig. 23. La magnitud y dirección de la corriente son función de la diferencia de temperatura de las uniones y de las propiedades térmicas de los metales usados en el circuito. A este fenómeno se le conoce como efecto Seebeck.

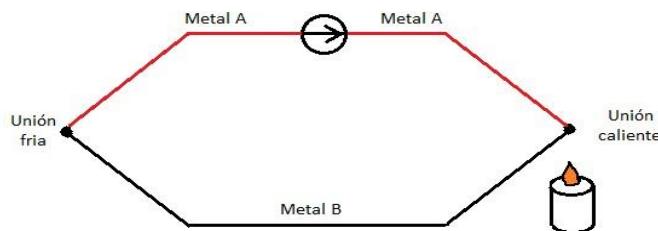


Fig. 23: Principio de Seebeck.

Si se abre este circuito (Fig. 24), se obtiene una diferencia de potencial pequeña, la cual es directamente proporcional a la temperatura de la unión y a la composición de los dos metales.

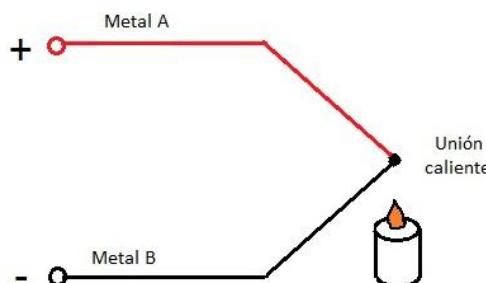


Fig. 24: Termopar.

Pero la diferencia de potencial no puede ser medida directamente con un voltímetro debido a que, la unión del termopar con el voltímetro crea un nuevo circuito termoeléctrico.

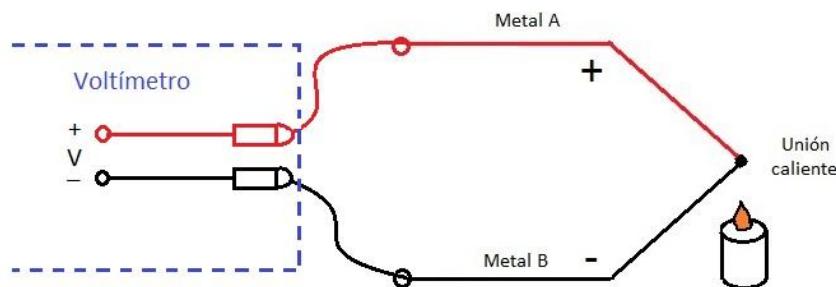


Fig. 25: Conexión directa de un voltímetro y un termopar.

Para poder realizar una buena medición se debe utilizar un bloque isotérmico³⁰ cuya temperatura sea medida con un termistor. [42]

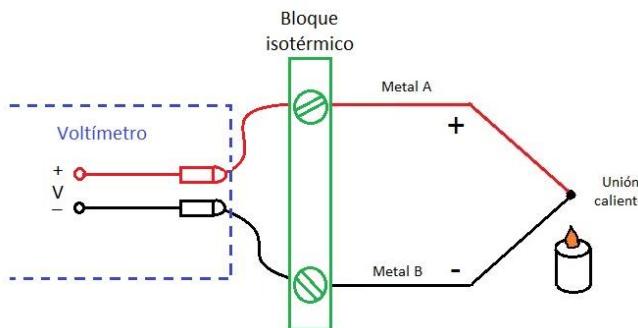


Fig. 26: Conexión adecuada de un termopar con un voltímetro.

3.3.1.2 Termistor

Los termistores, o resistores térmicos, son dispositivos semiconductores que se comportan como resistencias con un coeficiente de temperatura de resistencia alto y, generalmente negativo. En algunos casos, la resistencia de un termistor a temperatura ambiente puede disminuir hasta un 6% por cada 1°C que se eleve la temperatura. El uso de termistores está muy difundido en tales aplicaciones, en especial en el rango más bajo de temperatura de -100°C a 300°C.

Los termistores se componen de una mezcla sintética de óxidos de metales, como manganeso, níquel, cobalto, cobre, hierro y uranio. Su rango de resistencia va de 0.5Ω a 75Ω y están disponibles en una amplia gama de formas y tamaños. Los más pequeños son cuentas con un diámetro de 0.15 mm a 1.25mm.

³⁰ Isotérmico: De igual temperatura. [Diccionario Encyclopédico Salvat, 1985, Vol. 15, Varios Autores, pg. 2118]

Las cuentas se pueden colocar dentro de una barra de vidrio para formar sondas que son más fáciles de montar que las cuentas. Se hacen disco y arandelas³¹ presionando el material termistor en condiciones de alta presión en formas cilíndrica y plana con diámetros de 2.5 mm a 25 mm. Las arandelas se pueden apilar y conectar en serie o paralelo con el fin de incrementar la disipación de potencia.

Dos características importantes del termistor lo hacen extremadamente útil en aplicaciones de medición y control:

1. Resistencia-temperatura: un termistor tiene coeficiente de temperatura de resistencia muy elevado y negativo, lo cual lo convierte en un “transductor³² de temperatura ideal”.
2. Voltaje-corriente: la caída de voltaje a través de un termistor aumenta con el incremento de corriente hasta que alcanza un valor pico, más allá del cual la caída de voltaje decrece con el incremento de corriente. [43]

3.3.1.3 Termoresistencia

La termoresistencia trabaja en la medida que varía la temperatura, su resistencia se modifica, y la magnitud de esta modificación puede relacionarse con la variación de temperatura.

Las termoresistencias de uso más común se fabrican de alambres finos soportados por un material aislante y luego encapsulados. El elemento encapsulado se inserta dentro de una vaina o tubo metálico cerrado en un extremo, que se llena con un polvo aislante y se sella con cemento para impedir que absorba humedad. Los materiales utilizados para los arrollamientos de termoresistencias son fundamentalmente platino, níquel, níquel-hierro, cobre y tungsteno

El rango de temperaturas y precisión que se pueden alcanzar según los materiales más usados, se muestra en la tabla 1.

Material	Rango de Operación (°C)		Precisión
	de	a	
Platino	200	950	0.01
Níquel	150	300	0.50
Cobre	20	120	0.1

Tabla 1: Cuadro comparativo entre materiales, según su rango de operación y precisión. [43]

³¹ Arandela: Anillo metálico. [Diccionario Enciclopédico Salvat, 1985, Vol. 2, Varios Autores, pg. 262]

³² Transductor: Dispositivo que convierte la potencia eléctrica de una corriente en potencia acústica o mecánica, o viceversa. [Diccionario Enciclopédico Salvat, 1985, Vol. 25, Varios Autores, pg. 3572]

La relación fundamental para el funcionamiento es: $Rt = Ro(1 + \alpha t)$, donde:

- Ro : resistencia a 0 °C
- Rt : resistencia a t °C
- α : coeficiente de temperatura de la resistencia. [43]

3.3.2 Presión

Existen muchos tipos de sensores de presión (mecánicos, electromecánicos, y electrónicos) pero los que más sirven para el proyecto que se presenta, son los electrónicos. [44]

3.3.2.1 Sensor capacitivo

Tiene una variación de capacitancia que se produce en un condensador al desplazarse una de sus placas. Su rango de medición es de 0.5 a 600 bar. [44]

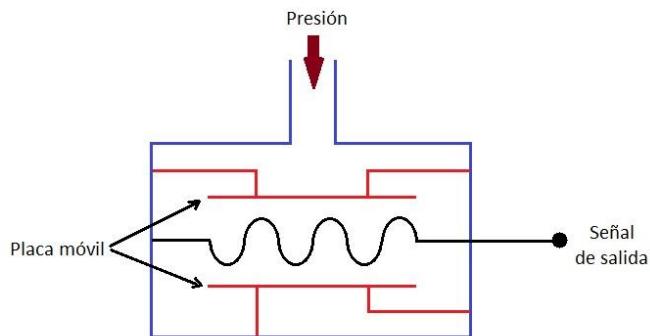


Fig. 27: Sensor de presión capacitivo.

3.3.2.2 Sensor piezoelectrónico

Son materiales cristalinos capaces de deformarse a causa de una presión, para generar una señal eléctrica. Sus mediciones están hechas con poca sensibilidad y su rango de medición va desde 0.1 hasta 600 bar. [44]

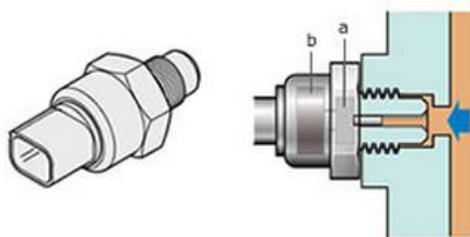


Fig. 28: Sensor de presión piezoelectrónico y funcionamiento.

[<http://www.aficionadosalamistica.net/sistema-esp.htm>]

3.3.2.3 Sensor resistivo

Consiste en un elemento elástico, que varía la resistencia de un potenciómetro en función de la presión. Este está conectado a un puente de Wheatstone. Estos sensores son de alta precisión y su rango de medición varía de 0.1 a 300 bar.

Los transductores resitivos son simples y su señal de salida es bastante potente como para proporcionar una corriente de salida suficiente para el funcionamiento de los instrumentos de indicación sin necesidad de amplificación. [44]

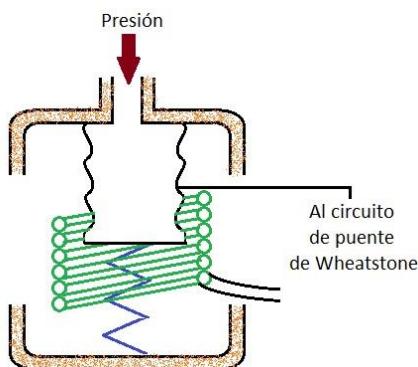


Fig. 29: Sensor de presión resistivo.

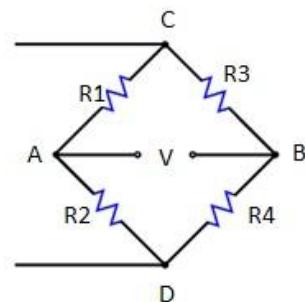


Fig. 30: Puente de Wheatstone.

3.3.2.4 Sensor magnético de inductancia variable

Su funcionamiento se basa en el desplazamiento de un núcleo móvil dentro de una bobina, aumenta la inductancia de ésta en forma casi proporcional a la porción metálica del núcleo, contenida dentro de la bobina. Esto hace aumentar la tensión inducida en el bobinado. Este sensor, tiene una respuesta lineal, son pequeños, de construcción robusta, su precisión es del orden del 1% y su rango de medición está entre 0.1 y 300 bar. [44]

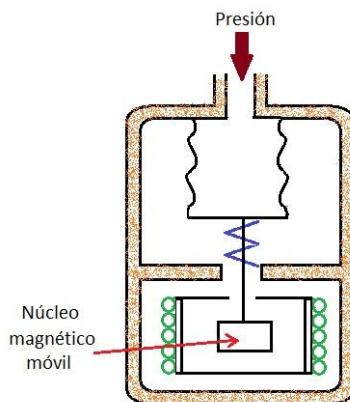


Fig. 31: Sensor de presión magnético de inductancia variable.

3.3.3 Humedad

3.3.3.1 Sensor de bulbos húmedo y seco

Este psicrómetro se basa fundamentalmente en la medición de temperatura, para a partir de ella deducir la cantidad de agua evaporada presente en una mezcla gaseosa. La idea consiste en disponer de dos termómetros lo más idéntico posibles, con uno de ellos se debe medir la temperatura de la mezcla (temperatura de bulbo seco), y con el otro, la temperatura en la superficie de una película de agua que se evapora (temperatura de bulbo húmedo), esto se logra envolviendo el bulbo de uno de los termómetros con un algodón humedecido con agua. Las moléculas de agua presentes en el algodón absorberán la energía necesaria para evaporarse del bulbo del termómetro, bajando la temperatura del mismo algunos grados por debajo comparado con la temperatura del termómetro seco.

Al conocerse el valor de ambas variables es posible determinar la humedad relativa, basta recurrir a las ecuaciones psicométricas. En un ambiente saturado, la cantidad de moléculas que se evaporan del bulbo húmedo son equivalentes a las que se condensan en él, por lo que ambos termómetros registran temperaturas idénticas. En la Fig. 34, se puede apreciar un diagrama explicativo de cómo se relacionan las variables que intervienen en la medición.

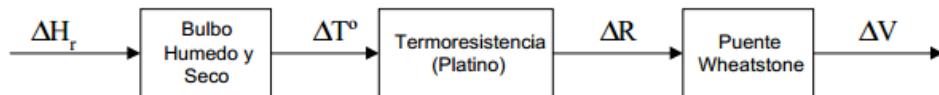


Fig. 32: Diagrama de relación de variables. [51]

Cabe señalar que en la actualidad ya no se emplean algodones ni termómetros convencionales, sino más bien mechazas que envuelven termocuplas o termoresistencias conectadas con algún recipiente de agua. También se emplean dispositivos cerámicos porosos que envuelven el bulbo y están en contacto con el agua.

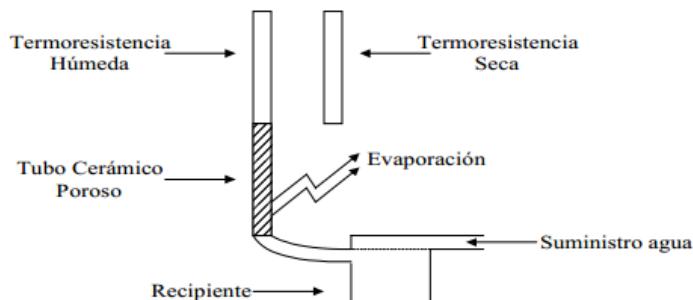


Fig. 33: Sensor de bulbo húmedo y seco. [51]

Este sensor requiere también una circulación de aire con una velocidad, para ello se pueden emplear ventiladores. Con este tipo de medición se logran valores de H_r de sólo 0.5% de error, sin embargo su flaqueza radica en que introduce vapor de agua al ambiente que se desea medir, lo cual en

algunos procesos puede ser un inconveniente. Los rangos de operación van de 0% a 100% de H_r , y de 0°C a 90°C. [51]

3.3.3.2 Sensor capacitivo

Son quizá, los más difundidos en la industria y meteorología, pues son de fácil producción, bajos costos, y alta fidelidad. El principio en el cual se basa este tipo de sensores, es en el cambio que sufre la capacidad (C en [Farad]) de un condensador, al variar la constante dieléctrica del mismo. Si se utiliza la mezcla gaseosa como dieléctrico entre las placas del condensador, el valor de este estará determinado por:

$$C = \left(\frac{\epsilon A}{d} \right)$$

Donde:

C : Valor de la capacidad.

ϵ : Permitividad del dieléctrico (constante dieléctrica).

A : Área de las placas del condensador.

d : Distancia entre las placas del condensador.

Dado a la naturaleza de composición del agua, en presencia de un campo eléctrico, se produce la alineación de la moléculas, siendo esta la causa del porque el agua presenta una constante dieléctrica mayor que el aire (80 y 1 respectivamente). Luego si se utiliza como dieléctrico, una mezcla gaseosa que contenga vapor de agua, el valor C del condensador va a variar dependiendo de la cantidad de moléculas de agua que estén presentes entre las placas. En consecuencia basta medir, o convertir el cambio de capacitancia, a otro tipo de variable eléctrica más fácil de manejar, lo anterior se puede lograr con un puente de Wheatstone (Fig. 30) de condensadores.

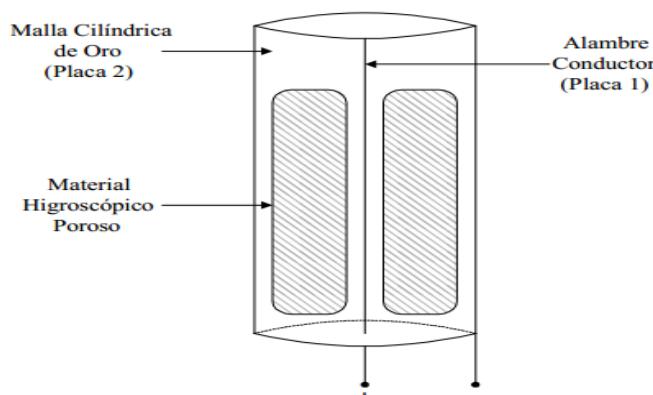


Fig. 34: Concentración de agua en las placas. [51]

Se utiliza como una de las placas un alambre conductor, mientras la otra es una malla fina de oro que permite el paso del gas, pero retiene impurezas. Como dieléctrico se utiliza un material higroscópico

poroso (cerámico) que rodea el alambre, el cual absorbe el agua de la muestra, aumentando aún más la constante dieléctrica del condensador, en proporción a la humedad relativa existente.

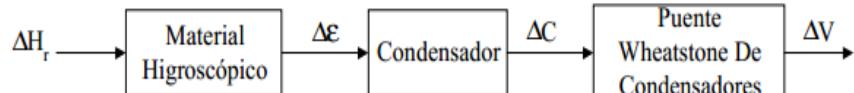


Fig. 35: Diagrama de relación de variables. [51]

Conviene limpiar el sensor periódicamente, con el fin de no obstruir la circulación de aire a través del manto, y por ende no alterar la medición. Son robustos y tienen excelente precisión, además de operar en rangos de temperaturas de -80°C a 60°C. [51]

3.3.3 Sensor piezoelectrónico

Los cristales poseen frecuencias de oscilación bastante estables, sin embargo, al cambiar la masa del cristal por deposiciones de materiales sobre él, éste experimenta una variación de 2000Hz en su frecuencia de oscilación, por cada microgramo de aumento de material sobre su superficie. Por ello son empleados en la medición de humedad, dado que basta cubrirlos con un material higroscópico, para que aumente la cantidad de agua sobre el cristal en forma proporcional a la humedad absoluta presente. A mayor masa, menor es la frecuencia de oscilación, con este sistema se pueden detectar hasta variaciones de $\pm 0.1\text{Hz}$, lo cual equivale a humedades de alrededor de 0.1 PPM (partes por millón). Son sensores bastante robustos, muy sensibles, y entregan una medida en forma de frecuencia, la cual puede ser utilizada para convertirla en voltaje. [51]

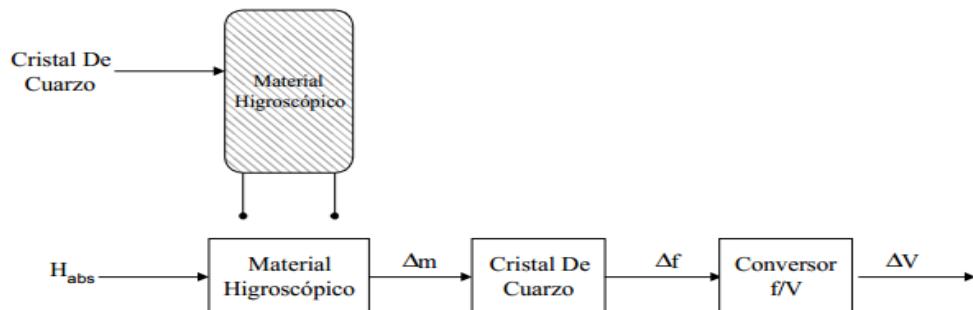


Fig. 36: Sensor piezoelectrónico y diagrama de relación de variables. [51]

3.3.4 Densidad de flujo magnético

3.3.4.1 Sensor de salida lineal

En la Fig. 39 se muestra la estructura simplificada de un sensor Hall de salida lineal. Debido a que la tensión Hall es muy pequeña, es necesaria una etapa de amplificación, esto se hace a

través de un amplificador diferencial. Posee además una compensación en temperatura y un regulador de voltaje. La salida posee un transistor, para realizar la interfaz con cualquier dispositivo.



Fig. 37: Sensor de efecto Hall de salida lineal. [40]

Según la polaridad del campo magnético que se le aplique al sensor, la tensión de salida será positiva o negativa. Pero el amplificador diferencial incorpora una tensión de polarización, de modo que con un campo magnético nulo tengamos en la salida una tensión positiva. [41]

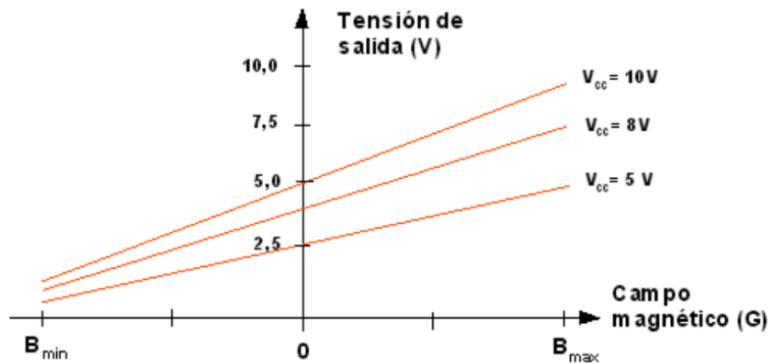


Fig. 38: Transferencia del sensor lineal. [41]

3.3.4.2 Sensor de salida digital

La estructura de este tipo de sensor es la misma que la del lineal, pero debemos agregar un comparador de disparador Schmitt.

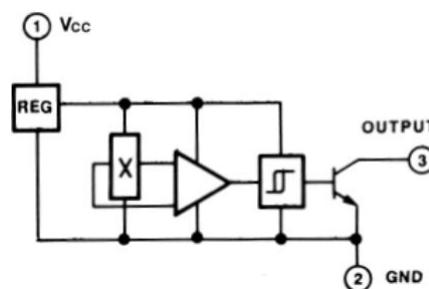


Fig. 39: Sensor de salida digital. [41]

Sin campo magnético aplicado al dispositivo, el transistor de salida permanece cortado (OFF). Al aplicar un campo magnético, perpendicular a la superficie del dispositivo, que está por encima de un valor umbral, el transistor de salida pasa a saturación (ON).

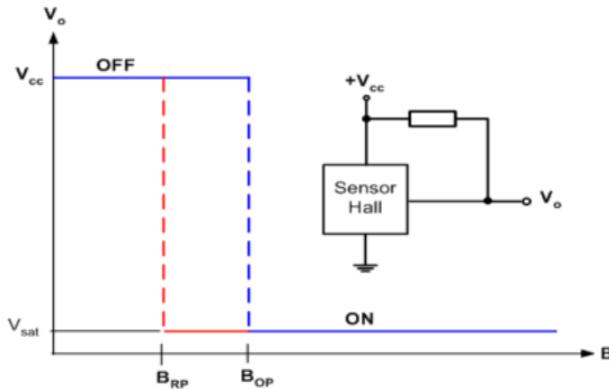


Fig. 40: Señal del sensor. [41]

3.4 Convertidor Analógico-Digital (ADC)

El ADC es un elemento muy indispensable en un sistema de adquisición de datos. Un ADC es un dispositivo que convierte una tensión eléctrica en un valor numérico (binario). Se usa este debido a que, de algún modo, cualquier otra señal puede llegar a manifestarse en tensión eléctrica, tal como funcionan, por ejemplo, los sensores de luz, de temperatura, etc. [45]

Un ADC toma valores analógicos de tensión y los convierte en códigos binarios. Los valores que definen los límites de las tensiones a medir se denominan voltajes de referencia y se representan por V_{ref-} (el mínimo) y V_{ref+} (el máximo).

La resolución del conversor queda determinada por la cantidad de bits que representan el resultado de la conversión. Un ADC de “ n ” bits puede representar hasta 2^n valores digitales, de modo que a la entrada analógica igual a V_{ref-} le asignará el 0 digital y la entrada igual a V_{ref+} le asignará el valor $2^n - 1$ digital. A los otros valores analógicos se les asignará los otros valores digitales distribuidos equidistantemente y definidos por la siguiente ecuación:

$$N_{ADC} = \frac{(V_{in} - V_{ref-})(2^n - 1)}{V_{ref+} - V_{ref-}}$$

De tal manera que si $V_{ref-} = Gnd$, es decir $V_{ref-} = 0v$, tenemos la siguiente ecuación:

$$N_{ADC} = \frac{V_{in}(2^n - 1)}{V_{ref+}}$$

La diferencia entre dos valores analógicos correspondientes a dos valores digitales consecutivos se define como resolución de voltaje de ADC y está dada por la siguiente ecuación:

$$Res = \frac{V_{ref+} - V_{ref-}}{2^n - 1}$$

De ahí, que el dato digital (Nivel de cuantización), también pueda verse como:

$$N_{ADC} = \frac{(V_{in} - V_{ref-})}{Res}$$

Y para cuando $V_{ref-} = Gnd = 0v$:

$$N_{ADC} = \frac{V_{in}}{Res}$$

Entre V_{ref-} y V_{ref+} se pueden concebir infinitos valores analógicos, pero con n bits solo se pueden formar 2^n valores discretos diferentes. Por lo tanto habrá valores analógicos que no podrán ser representados con exactitud, así, cada dato obtenido después de la conversión de un ADC, tendrá un error llamado "error de cuantización", el cual se define como:

$$E_C = \pm \frac{Res}{2}$$

A nivel de elemento de circuito, el ADC se caracteriza por una entrada analógica, una salida digital y varias señales de control y alimentación. [46]

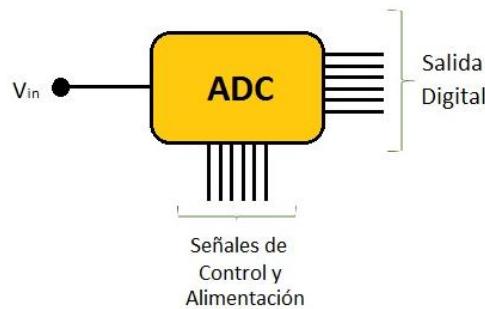


Fig. 41: Composición de un ADC.

Las señales de control más importantes y características son: SC (Start Conversion/Inicio de conversión) y EOC (End Of Conversion/Fín de conversión). La primera es una entrada que requiere el circuito para que comience la conversión, que durará un tiempo que a veces es conocido de antemano y otras veces no. La señal EOC es la que indica cuándo ha terminado la conversión, por tanto una señal de salida.

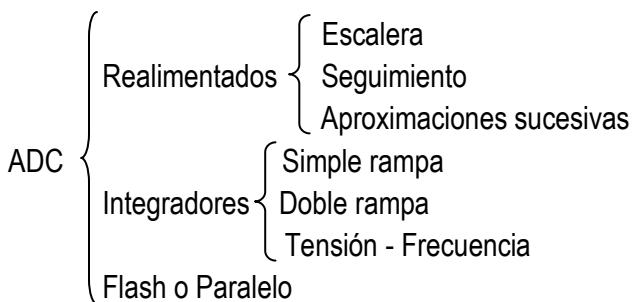
El elemento de salida del ADC es un latch o registro donde se almacena el dato. Este permanecerá almacenado o cambiará controlado por unas entradas de Enable y Chip Select del latch.

El funcionamiento de un ADC es muy simple: se inicia la conversión cuando la señal SC pasa de un estado lógico 0 a 1. El ADC comienza la conversión y avisa cuándo termina mediante una bajada a nivel lógico 0 del EOC. [46]

Entonces, las características principales que deben tomarse en cuenta al elegir un ADC, son: la exactitud, que depende de los errores que se produzcan y de la resolución (número de bits), y la velocidad o tiempo de conversión.

3.4.1 Tipos de ADC

Los ADC se pueden clasificar en los siguientes tipos:



Debido a que casi todos los módulos ADC de los microcontroladores son de aproximaciones sucesivas, en éste documento únicamente se explicará el método de dicho ADC y el del flash o paralelo puesto que es el más rápido, es decir, en el que el tiempo de conversión es más corto.

3.4.1.1 ADC de aproximaciones sucesivas

Funcionan con cuatro elementos básicos (Fig. 44): un comparador analógico, una lógica de control, un conversor digital analógico DAC y el reloj que guía los pasos de la conversión. Los DAC son mucho más simples que los ADC y entregan resultados casi de inmediato.

Cada número binario generado va siendo convertido en una tensión analógica V_{out} que luego se compara con la tensión de entrada que queremos medir V_{in} . Si son iguales (o los más cercanos posible), entonces es el número binario que corresponde a V_{in} .

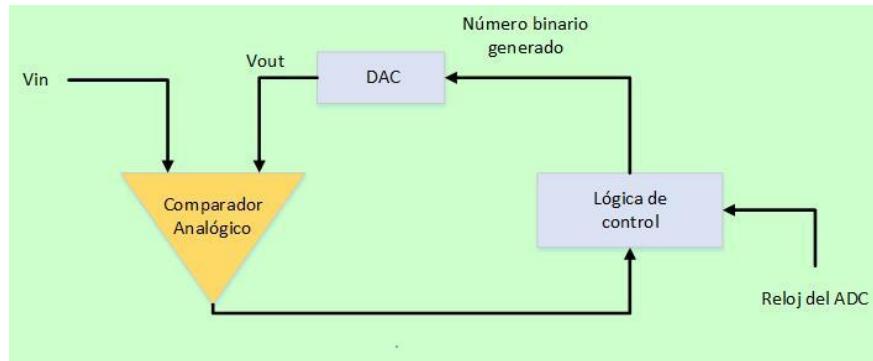


Fig. 42: Diagrama a bloques de un ADC de aproximaciones sucesivas.

Ahora bien, siendo el conversor de 10 bits y pudiéndose generar hasta 1024 números binarios distintos, no se realizan las 1024 comparaciones como pudiera pensarse, sino únicamente 10, para entender mejor cómo funciona éste ADC, supongamos que el ADC trabaja con tensiones de referencia de 0 V y 5 V y que queremos medir una señal V_{in} de 4 Volts. Como el ADC es de 10 bits, dará los 10 pasos mostrados en la siguiente tabla:

Paso #	Binario Generado										V_{out} DAC (V)	V_{in} ADC (V)	$V_{out} > V_{in}$
	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀			
1	1	0	0	0	0	0	0	0	0	0	2.50244	4.0	Sí, D ₉ = 1
2	1	1	0	0	0	0	0	0	0	0	3.75366	4.0	Sí, D ₈ = 1
3	1	1	1	0	0	0	0	0	0	0	4.37927	4.0	No, D ₇ = 0
4	1	1	0	1	0	0	0	0	0	0	4.06647	4.0	No, D ₆ = 0
5	1	1	0	0	1	0	0	0	0	0	3.91	4.0	Sí, D ₅ = 1
6	1	1	0	0	1	1	0	0	0	0	3.98826	4.0	Sí, D ₄ = 1
7	1	1	0	0	1	1	1	0	0	0	4.02737	4.0	No, D ₃ = 0
8	1	1	0	0	1	1	0	1	0	0	4.00782	4.0	No, D ₂ = 0
9	1	1	0	0	1	1	0	0	1	0	3.99804	4.0	Sí, D ₁ = 1
10	1	1	0	0	1	1	0	0	1	1	4.00293	4.0	No, D ₀ = 0

Tabla 2: Proceso de un ADC de 10 bits cuando $V_{in} = 4$ V. [47]

Valor final entregado por el ADC: 11 0011 0010, correspondiente a un voltaje de 3.99804 V \approx 4.0 V.

El primer número binario generado tiene el bit D₉ = 1. Este número se convierte en el valor analógico V_{out} , que después se compara con V_{in} . Como la comparación ($V_{out} < V_{in}$) es positiva nos quedamos con este bit. Después se prueba poniendo en alto el siguiente bit, D₈ = 0, y como ahora la evaluación ($V_{out} < V_{in}$) sigue siendo afirmativa también nos quedamos con este bit. En seguida prueba con el bit D₇ = 1, ahora la evaluación ($V_{out} < V_{in}$) es negativa y debemos limpiar D₇, es decir, D₇ = 0. Y se sigue con el resto de los bits de manera similar hasta alcanzar el bit D₀.

Al final, se obtiene el valor 11 0011 0010, que significa una tensión de 3.99804 V, y que comparado con los 4.0 V de entrada, tiene un error de 0.00196 V = 1.96 mV, o bien, de 0.049%. Con ello, podemos intuir que a mayor resolución en bits, habrá más aproximación. [47]

El tiempo en que se realiza la conversión está dado por:

$$t_c = (n + 1)T_{clk}$$

Donde:

T_{clk} : Período del reloj el ADC.

n: Número de bits del ADC.

3.4.1.2 ADC flash o paralelo

Se trata de un convertidor excepcionalmente rápido pero muy complejo desde el punto de vista del circuito. Su estructura tiene dos partes. En el primer nivel aparecen un conjunto de comparadores en donde, si el codificador tiene a la salida n niveles, necesitamos 2^n comparadores a la entrada (para 8 bits se necesitan 256 comparadores).

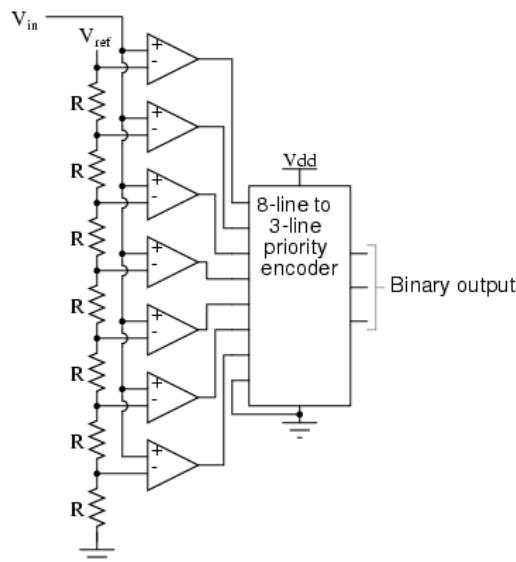


Fig. 43: Diagrama interno de un ADC flash. [48]

La diferencia de tensiones entre dos resistencias adyacentes es:

$$V_{LSB} = \frac{V_{ref+} - V_{ref-}}{2^n}$$

El funcionamiento de este comparador es simple: todos aquellos comparadores en los que V_{in} sea mayor que su tensión de referencia, estarán en saturación positiva entregando un 1 lógico, mientras que

los demás no, entregando un 0 lógico. El segundo nivel es un codificador que convierte las 2^n entradas en n salidas, las cuales se convierten en el resultado final del ADC. [48] La conversión es realizada casi instantáneamente así que el tiempo de conversión es de nanosegundos.

A continuación se muestra un cuadro comparativo (Tabla 3) de las características principales del ADC de aproximaciones sucesivas y el ADC flash.

ADC	
De Aproximaciones Sucesivas	Flash o Paralelo
Tiempos grandes de conversión.	Tiempos de conversión dados en nanosegundos.
Poca circuitería.	Se necesita un comparador por cada nivel de cuantización (mucha circuitería).
Es el más utilizado.	Poco utilizado en la industria.
Precio accesible.	Costo relativamente alto debido a su estructura interna.
Necesita un reloj interno para dar seguimiento a los pasos de la conversión.	No necesita reloj puesto que la conversión es casi instantánea.

Tabla 3: Cuadro comparativo entre ADC de aproximaciones sucesivas y ADC flash o paralelo.

3.5 Amplificadores Operacionales (AO)

Los AO son circuitos integrados con un nivel de componentes y estructura interna complicada, su símbolo se muestra en la Fig. 44. La alimentación del circuito se realiza por medio de dos fuentes (alimentación simétrica). Como se aprecia en la Fig. 45, la terminal de referencia (tierra) no está conectada directamente al amplificador operacional, debido a que ésta debe realizarse a través de elementos externos al operacional, tales como las resistencias.

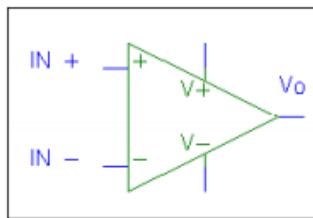


Fig. 44: Símbolo del Amplificador Operacional. [49]

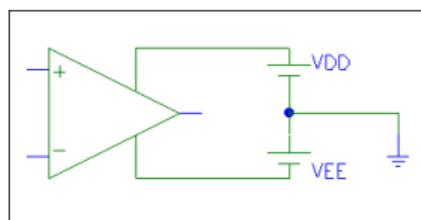


Fig. 45: Alimentación de un AO. [49]

Tienen dos entradas (“+” y “-”), la “-”, que se denomina “inversora” y la “+”, que se denomina “no inversora”, y una salida V_o . Se alimentan a través de dos terminales uno con tensión positiva $+V$ y otro con tensión negativa $-V$. Adicionalmente pueden tener otros terminales específicas para compensación de frecuencia, corrección de derivas de corriente continua etc.

3.5.1 Amplificador ideal

En el modelo de amplificador ideal, la salida del amplificador se obtiene a través de la expresión:

$$V_o = A(V_+ - V_-) = AV_d$$

Esta expresión nos dice que la salida del amplificador es directamente proporcional a la diferencia de potencial (V_d) en la entrada. “A” es una constante para cada amplificador y sus valores son muy altos (>200000 para amplificadores reales). Las propiedades del amplificador operacional ideal:

- La ganancia “A” es infinita.
- Las resistencias que se ven desde cada una de las terminales de entrada son infinitas, o lo que es lo mismo, las intensidades de entrada ($+I$ y $-I$) son nulas.
- La impedancia de carga de un circuito conectado en cascada con el AO no influye en la tensión de salida, esto es, $V \neq f(Z_L)$, lo que significa que, si tenemos el equivalente del amplificador de la forma:

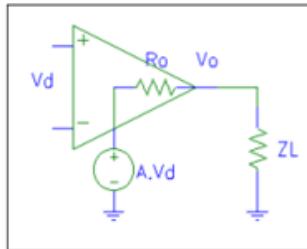


Fig. 46: Circuito conectado en cascada con un AO. [49]

$$\text{entonces: } V_o = \frac{Z_L}{R_0 + Z_L}$$

3.5.2 Amplificador Operacional como inversor

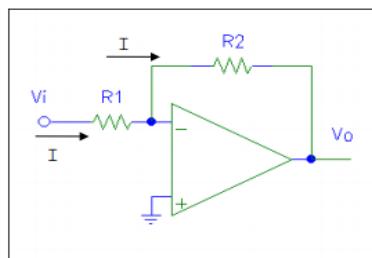


Fig. 47: Circuito esquemático de un AO como inversor. [49]

La ecuación característica de esta configuración es:

$$V_o = -\frac{R_2}{R_1} V_i$$

3.5.3 Amplificador Operacional en configuración sumador/restador no inversor

La salida de esta configuración de amplificador, es proporcional a la suma de las señales de entrada. Dado que $V_- = 0V$ por ser igual a V_+ , las intensidades que circulan por cada rama son independientes de las demás y no se produce redistribución de intensidad alguna. Con ello la intensidad total que atraviesa R_2 será la suma de las intensidades de cada una de las ramas de entrada, se puede tomar este circuito como restador ya que las entradas de voltaje pueden ser negativas.

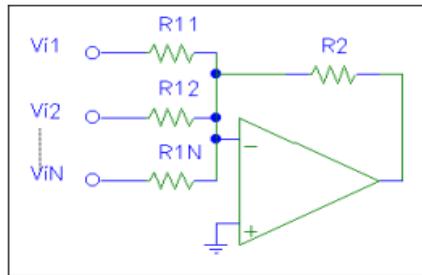


Fig. 48: Circuito esquemático de un AO en configuración sumador/restador no inversor. [49]

La ganancia en lazo cerrado (con retroalimentación) de esta configuración es:

$$V_o = -R_2 \left(\frac{V_1}{R_{11}} + \frac{V_{12}}{R_{11}} + \dots + \frac{V_{1N}}{R_{11}} \right)$$

3.5.4 Amplificador Operacional en configuración diferencial

Se trata de una configuración con dos entradas, en la que se amplifica la diferencia de potencial entre ambas, es la combinación de la configuración amplificador inversor y no inversor.

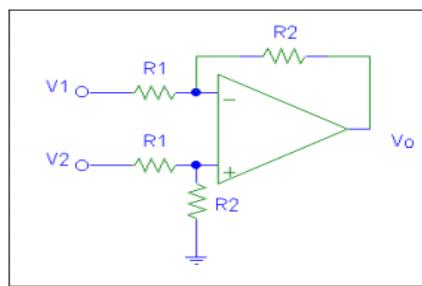


Fig. 49: Circuito esquemático de un AO en configuración diferencial. [49]

La ganancia en lazo cerrado de esta configuración es:

$$V_o = \frac{R_2}{R_1} (V_2 - V_1)$$

3.6 Bluetooth

Es una tecnología que provee comunicación a velocidades razonables, entre diferentes dispositivos, sin el uso de cables. Además, se busca facilitar la sincronización de datos de computadoras móviles, teléfonos celulares y manejadores de dispositivos. La tecnología bluetooth es de bajo costo y se caracteriza por usar enlaces de radio de corto alcance. Esta tecnología opera en la banda de 2.4 GHz, tiene la capacidad de atravesar paredes y ha sido diseñado para tener una comunicación continua, incluso en movimiento. [50]

3.6.1 Funcionamiento del estándar

Cada dispositivo Bluetooth, está equipado con un encapsulado, que transmite y recibe en la frecuencia 2.45 GHz, disponible en todo el mundo (con algunas variaciones de ancho de banda en diferentes países como: España, Francia y Japón), sin necesidad de licencia.

Cada dispositivo tiene una dirección única de 48 bits, basada en el estándar IEEE 802.11 para LAN inalámbrica que le permite formar, temporalmente, parte de una *piconet*³³. Las conexiones son uno a uno con una distancia máxima de 10m, entre sí, aunque utilizando amplificadores se puede llegar hasta los 100m, pero, con la desventaja de que se tendría interferencia en la comunicación (ruido en la señal).

Los datos se pueden intercambiar a velocidades de hasta 1 Mbit/s. La tecnología “frequency hop” (saltos de frecuencia), permite a los dispositivos comunicarse inclusive en áreas donde existe una gran interferencia electromagnética, además provee mecanismos de encriptación (con claves de hasta 64 bits) y autenticación, para controlar la conexión y evitar que un dispositivo no autorizado acceda a los datos.

Bluetooth se ha diseñado para operar en un ambiente multiusuario, hasta ocho usuarios o dispositivos pueden formar una piconet y hasta diez piconets pueden existir en la misma área de cobertura. Dado que cada enlace es codificado y protegido contra interferencia y pérdida de enlace, Bluetooth puede considerarse como una red inalámbrica de corto alcance muy segura.

En cuanto a interferencias con otros dispositivos, hay que tener cuidado con los que operan en la misma frecuencia. Cualquier otro dispositivo que incorpore un chip Bluetooth, podría interferir con los elementos de navegación. [50]

³³ Piconet: Red muy pequeña. Interconexión de máximo. [Definición propia]

Especificaciones:

- Banda de frecuencia de 2.4 GHz.
- Potencia del transmisor: 1mW para un alcance de 10m, 100 mW para un alcance de hasta 100m.
- Tecnología: Espectro expandido y saltos en frecuencia.
- Canales máximos de datos: 7 por piconet.
- Velocidad de datos: 771 Kbit/s por piconet.
- Rango esperado de comunicación: 10m.
- Número de dispositivos: 8 por piconet y hasta 10 piconets.
- Alimentación: 2.7 Volts.
- Consumo de potencia: desde de $30\mu\text{A}$ a 30 mA transmitiendo.
- Tamaño del módulo: 0.5 pulgadas cuadradas.
- Interferencia: Bluetooth minimiza la interferencia potencial al emplear saltos rápidos en frecuencia/1600 veces por segundo.

3.7 Universal Asynchronous Receiver/Transmitter (UART)

El receptor/transmisor asíncrono universal (Universal Asynchronous Receiver/Transmitter), es el dispositivo clave de un sistema de comunicaciones serie. Su función principal es convertir los datos serie a paralelos cuando se trata de datos recibidos (de entrada), y de convertir datos paralelos a serie para transmisión (de salida). En la Fig. 50, se muestra el esquema general con los bloques básicos de un UART. Aunque la transmisión es realmente síncrona, tradicionalmente se les ha llamado asíncrona para hacer referencia al hecho de que existe flexibilidad, para la transmisión de la palabra de datos siguiente. [52]

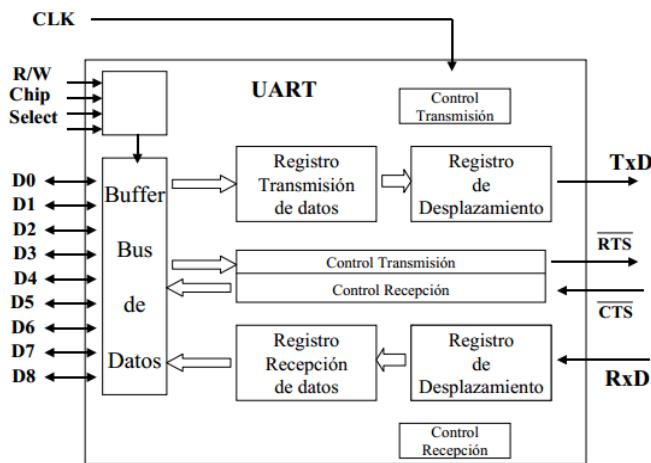


Fig. 50: Diagrama de bloques de un UART. [52]

Entre las aplicaciones básicas de la UART, se puede citar a las comunicaciones con la PC, transmisores de audio, video, juegos por infrarrojo o bluetooth, entre muchas otras.

Los parámetros que se han de definir para el correcto diseño de un interfaz de comunicaciones usando un UART son los siguientes:

- Sincronismo entre el receptor y el transmisor: Llevado a cabo, siguiendo el esquema de transmisión mostrado en la Fig. 51.
- Codificación de los datos: Cualquier código binario. El más utilizado es el código ASCII (American Standard Code for Information Interchange), que utiliza 7 bits para codificar 96 caracteres imprimibles y 32 caracteres de control.
- Prioridad en el envío de los bits: Se suelen llamar marcas (mark) al 1 lógico y espacios (space) al 0 lógico. Cuando no se realiza transmisión se mantiene el estado de marca. El comienzo de la transmisión se realiza con el estado de espacio.
- Tasa de envío de datos: Se mide en baudios o número de bits que se transmiten por segundo (bps). Puede ser cualquier valor, pero las tasas estándares son: 110, 150, 300, 900, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 31250, 38400, 57800 y 115200. [52]

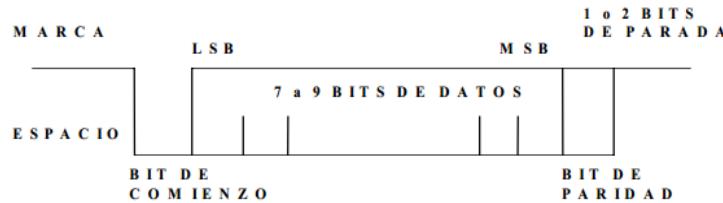


Fig. 51: Esquema de transmisión de un UART. [52]

En las Fig. 52 y 53, se muestran los diagramas de bloques del UART0 incluido en la tarjeta de desarrollo del FRDM-KL25Z, que es la utilizada en este proyecto.

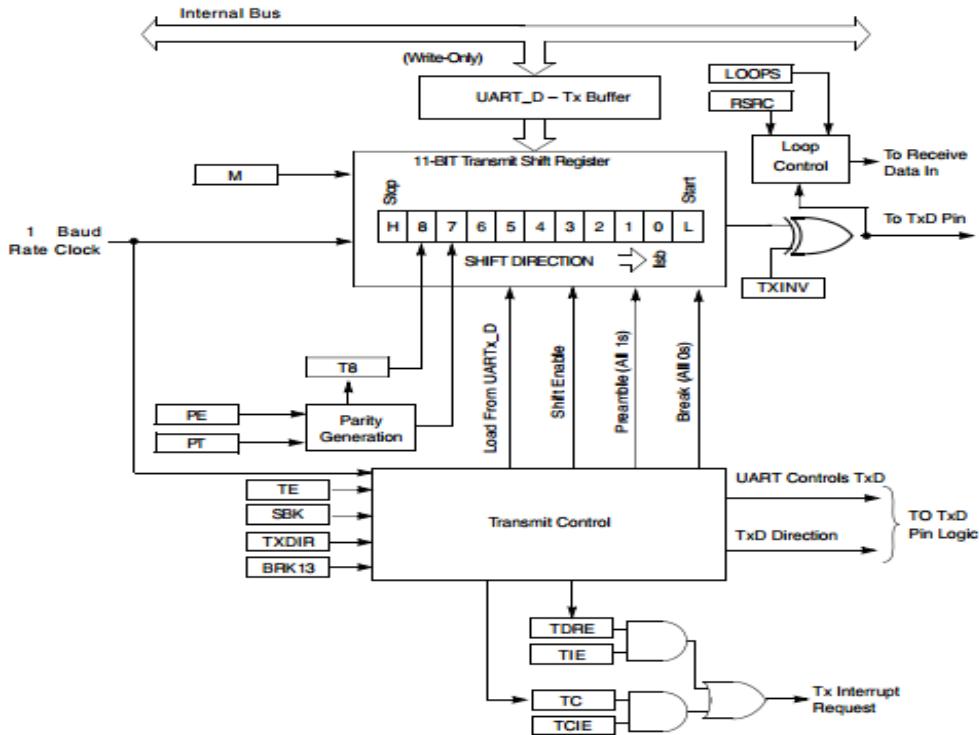


Fig. 52: Diagrama a bloques del transmisor del UART0 del microcontrolador MKL25Z128. [53]

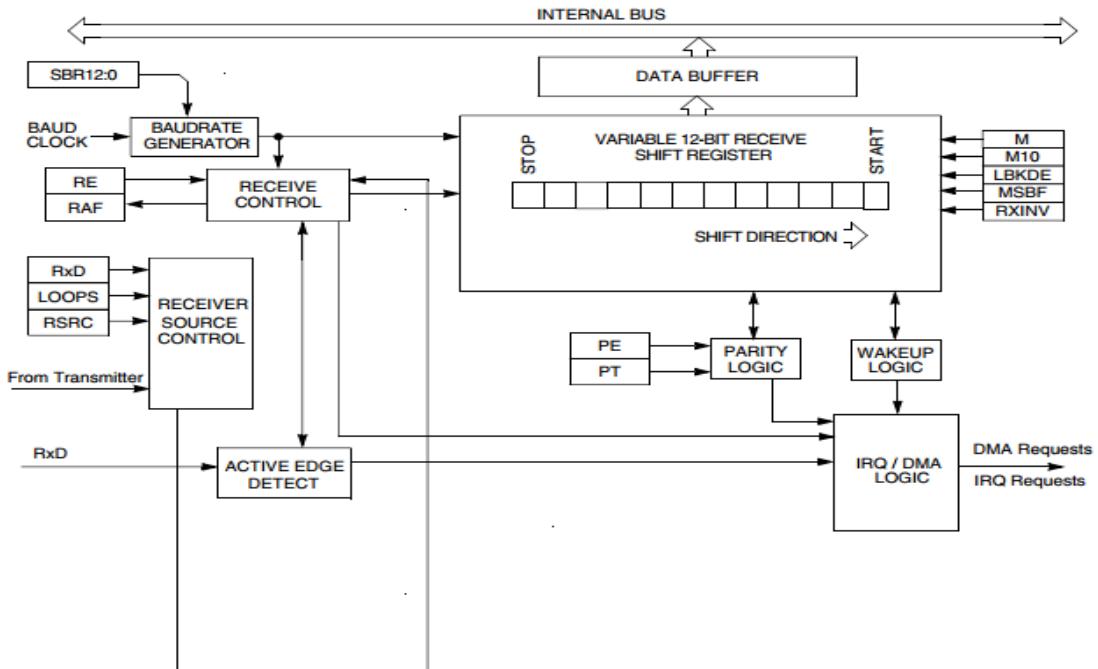


Fig. 53: Diagrama a bloques del receptor del UART0 del microcontrolador MKL25Z128. [53]

En el bloque de transmisor, se puede observar el bus interno del microcontrolador de sólo escritura hacia el bloque de transmisión y a su vez éste está interconectado a un registro de corrimiento de 11 bits, utilizado para el envío de los datos recibidos paralelamente por el microcontrolador en serie con cada ciclo que indica el control de lazo (Control Loop), a la frecuencia que indique el reloj de baudios, se pueden observar otros módulos, como un generador del bit de paridad y un bloque utilizado para controlar la transmisión de la señal, y saber si se va a recibir o a transmitir, así como un arreglo de compuertas lógicas que se utilizan en casos especiales como los servicios de interrupción.

En el bloque del receptor podemos ver un detector de flancos (active edge detect), utilizado para funciones adicionales como servicios de interrupción. El bloque de control de recepción (receive control) recibe las señales de los registros RE (Receive Enable), para habilitar la recepción, este bloque tiene conectado un generador de baudios, el cual debe ser sincronizado a la misma frecuencia que el dispositivo con el cual, se está efectuando la comunicación así como el transmisor. Después de este bloque se encuentra un registro de corrimiento variable de 12 bits, que tiene la tarea de convertir un dato serie en paralelo para ser enviado a través del bus interno.

3.8 Protocolo serial RS-232

Un uso común que se le da al módulo UART, es comunicar un microcontrolador con una computadora usando el puerto serie. A pesar de que muchos sistemas nuevos han abandonado el puerto serie y adoptado conexiones por USB, el puerto serie sigue siendo usado en la industria, debido a su sencillez, en comparación con protocolos seriales modernos como el USB. En el protocolo RS-232, el

dispositivo a conectar es llamado DCE (Data Communication Equipment) y la terminal es llamada DTE (Data Terminal Equipment).

El término "serial" viene del hecho de que el puerto serie "serializa" los datos. Esto quiere decir que toma un byte de datos y transmite un bit a la vez. La ventaja del puerto serie es que necesita únicamente un cable para transmitir los 8 bits (mientras que un puerto paralelo necesita 8). La desventaja es que dura 8 veces más para transmitir el dato que si tuviéramos 8 cables.

En el protocolo RS-232, antes de cada byte de información, se envía un bit de Start, el cual es un bit con valor de 0, después se envía el octeto y finalmente se envía un bit de stop, para indicar que el byte ha sido completado. Algunas veces también se manda un bit de paridad para verificación de errores. [54]

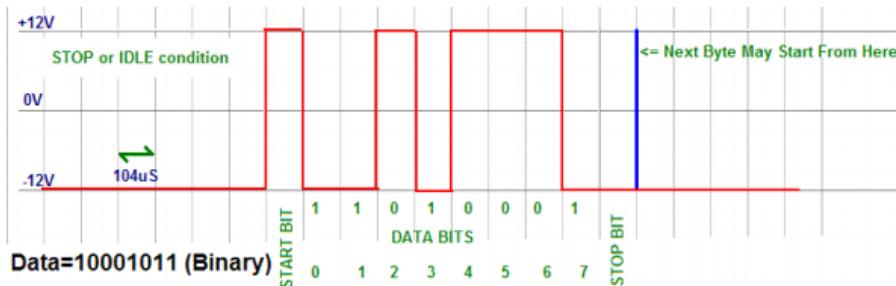


Fig. 54: Protocolo de comunicación RS232. [54]

Un aspecto importante de la comunicación serial es el concepto de control de flujo, que es la capacidad de sincronización entre transmisor y receptor con la finalidad de que el receptor no se sature. Existe el control de flujo por software y control de flujo por hardware.

Todos los pines del puerto serie de la computadora usa niveles RS-232, en los cuales un voltaje de +3V a +25V indican un “0” y de -3V a -25V indican un “1”. Cualquier voltaje entre -3V y 3V es un bit indefinido. Debido a esto es necesario un Convertidor de niveles RS-232.

3.8.1 Convertidor de niveles RS-232

Uno de los convertidores más comunes es el MAX3232, este convertidor convierte niveles de TTL a niveles RS-232 y viceversa. Incluye 2 receptores y 2 transmisores en el mismo encapsulado. Existen muchas variaciones de este convertidor, inclusive algunos con capacitores internos. En la Fig. 55 se muestra el diagrama de conexión eléctrica de un MAX3232 y una vista interna de los componentes que lo conforman, además de los componentes externos que debe tener.

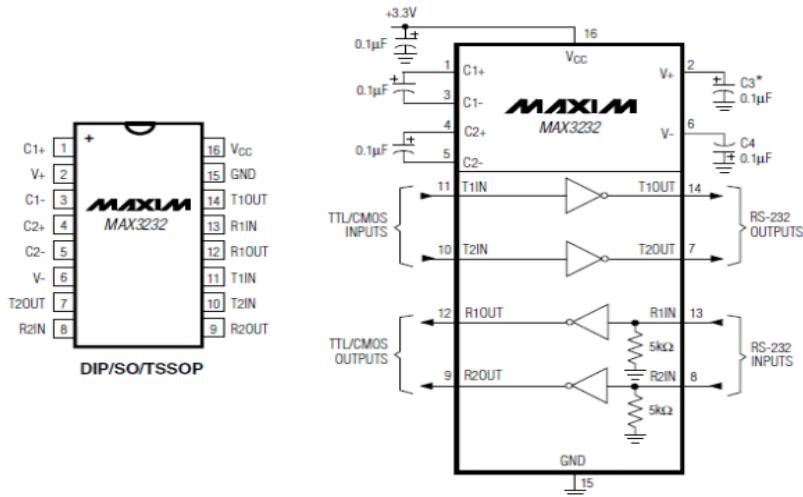


Fig. 55: Convertidor de niveles TTL a RS-232 utilizando el MAX3232. [54]

3.9 Interfaz Gráfica de Usuario (IGU)

La Interfaz Gráfica de Usuario (IGU), es una interfaz que permite al usuario interactuar de manera amigable con una aplicación de computadora, implementando ventanas, íconos y menús, usando los dispositivos de entrada conectados a la computadora, principalmente el teclado y el mouse.

Capítulo 4: Desarrollo del Hardware

4. Desarrollo del Hardware

En base al [objetivo general](#) planteado para el proyecto, y teniendo en cuenta que éste consta de dos partes fundamentales (hardware y software), en este capítulo, se explicará el proceso llevado a cabo, para el desarrollo del hardware.

Primero que nada, se creó el diagrama a bloques del funcionamiento de la parte física del sistema, el cual se muestra en la Fig. 56.

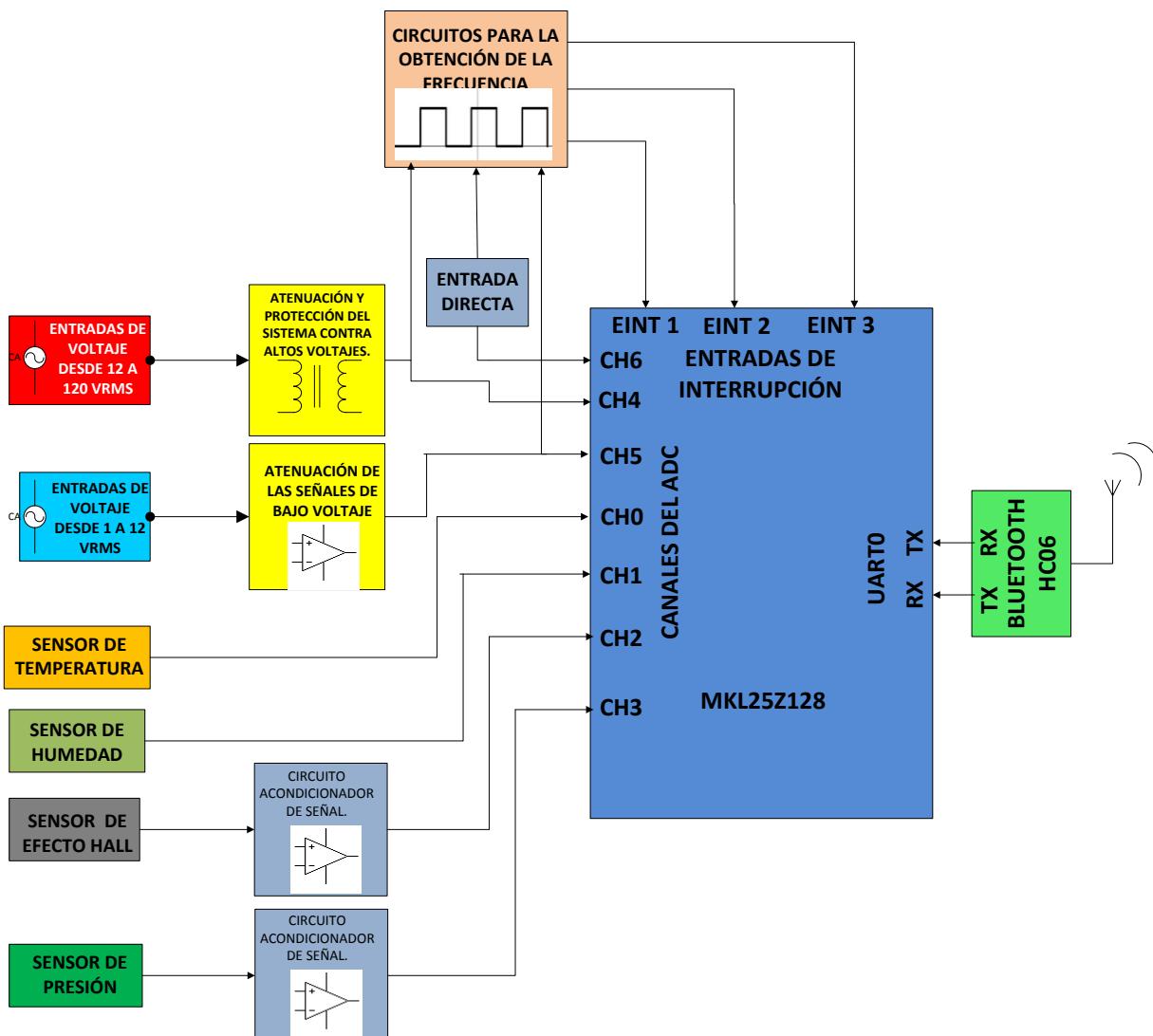


Fig. 56: Diagrama a bloques del hardware del proyecto.

Los pasos realizados, para el desarrollo de cada uno de los bloques que conforman dicho sistema, así como su diseño y funcionamiento, se explican a continuación.

4.1 Selección de dispositivos

4.1.1 Microcontrolador

Gracias al análisis y comparación de las principales características de diferentes microcontroladores ([Anexo A.2](#)), se tomó la decisión de que el microcontrolador más óptimo a usar para la adquisición de datos, es el microcontrolador MKL25Z128 de Freescale, contenido en la tarjeta de desarrollo FRDM-KL25Z, tomando en cuenta que posee las siguientes características:

- Tecnología ARM Cortex M0+.
- Tamaño de palabra de 32 bits.
- ADC de Aproximaciones Sucesivas (SAR: Successive Approximation Register).
- ADC de 16 bits (0 - 65535).
- Frecuencia de operación de 48 MHz.
- Puerto USB.
- Acceso directo a memoria (DMA).
- Comunicación serial.
- Memoria flash de 128 KB.
- Alimentación de 3.3V.
- DAC de 12 bits.

Sí bien, existen otros microcontroladores con características superiores que las que posee éste, al implementarlo en esta aplicación, se haría un desperdicio de recursos y con ello un gasto superior innecesario, puesto que el proceso a realizar por el microcontrolador, no requiere de grandes propiedades. En conclusión, el MKL25Z128, cuenta con lo necesario para la implementación del prototipo.

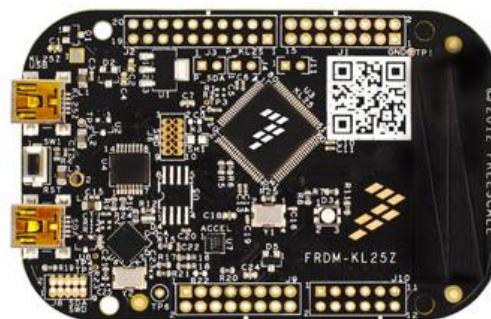


Fig. 57: Tarjeta de desarrollo FRDM-KL25Z. [53]

4.1.2 Sensores

Para seleccionar los distintos sensores a emplear en el prototipo, se elaboraron tablas de comparación ([Anexo A.1](#)), de las características más importantes de los sensores con mayor demanda en cada magnitud física a medir (temperatura, presión, humedad y densidad de flujo magnético), basándose también en su precio y proveedor.

Después del estudio de las propiedades electrónicas de cada sensor para las diferentes magnitudes, se llegó a la conclusión de que los más adecuados para ser utilizados en el prototipo, son los mostrados en la tabla 4.

Sensor	Matrícula	Distribuidor
Temperatura	LM61CIZ	AG
Presión	MPX10DP	AG
Humedad	HMZ-433A1	AG
Densidad de Flujo Magnético	UGN3503	AG

Tabla 4: Sensores seleccionados para el prototipo.

4.1.2.1 Sensor de temperatura LM61CIZ

Elegido por poseer las siguientes características:

- Es un sensor, con una función de transferencia lineal por lo que no requiere de un acondicionamiento previo de señal.
- Es muy preciso.
- Tiempo de respuesta de 20ms.
- No es necesario utilizar una fuente simétrica para alimentarlo, ya que el nivel de temperatura negativa se encuentra en un rango de valores de voltaje positivos.
- Rango de alimentación flexible.
- La impedancia de salida es baja por lo que puede comunicarse directamente con el microcontrolador.
- Alta disponibilidad en el mercado.
- Su precio es muy bajo para las propiedades que posee ([Anexo B.1](#)).
- Rango de medición: -30°C a 100°C
- Ecuación de la función de transferencia: $T = (V_o - 600) * 100$ [°C]



Fig. 58: Sensor de temperatura LM61CIZ.

[\[http://harald.studiokubota.com/wordpress/index.php/2011/07/16/arduino-and-temperature/\]](http://harald.studiokubota.com/wordpress/index.php/2011/07/16/arduino-and-temperature/)

4.1.2.2 Sensor de presión MPX10DP

Seleccionado debido a las siguientes características:

- El MPX10DP es un sensor lineal, sin embargo, al ser un sensor diferencial, requiere de un acondicionamiento de señal.
- Se puede usar para comparar dos magnitudes.
- Es muy preciso.
- Rango de alimentación flexible.
- La impedancia se encuentra en el rango necesario para alimentarlo directo con el microcontrolador $R < 2 \text{ K}\Omega$.
- Alta disponibilidad en el mercado.
- Muy usado en sistemas electrónicos de medición presión de gases en la industria.
- Costo relativamente bajo, en comparación con otros de su mismo tipo y para las características que posee ([Anexo B.2](#)).
- Rango de medición: 0Pa a 10KPa
- Ecuación de la función de transferencia: $P = \frac{V_o}{3.5} \quad [\text{KPa}]$



Fig. 59: Sensor de presión MPX10DP.

[<http://sigma.octopart.com/14173819/image/Freescale-Semiconductor-MPX10DP.jpg>]

4.1.2.3 Sensor de humedad HMZ-433A1

Apto para usarse en el prototipo, gracias a las siguientes propiedades:

- Es un sensor lineal.
- Es muy preciso.
- Alta disponibilidad en el mercado.
- Rango de medición: 20%RH a 90 %RH
- Costo relativamente bajo para contar con las características que posee ([Anexo B.3](#)).
- Ecuación de la función de transferencia: $R = V_o * 30.303 \quad [\%RH]$



Fig. 60: Sensor de humedad HMZ-433A1.

[http://img1.mlstatic.com/hmz-433a1-sensor-de-temperatura-y-humedad-hmz433-pic-avr_MLM-O-50132643_1943.jpg]

4.1.2.4 Sensor de efecto Hall (detector de densidad de flujo magnético) UGN3503

Se escogió este sensor por las siguientes características:

- El UGN3503 es un sensor lineal, sin embargo, requiere un previo acondicionamiento de señal ya que posee un nivel de CD.
- Muy baja impedancia de salida.
- Alto rango de medición.
- Bajo precio aún con las propiedades que posee ([Anexo B.4](#)).
- Alta disponibilidad en el mercado.
- Rango de medición: -900G a 900G
- Ecuación de la función de transferencia: $B = \frac{V_o}{1.3}$ [Gauss]



Fig. 61: Sensor de efecto Hall UGN3503.

[<http://www.masoportunidades.com.ar/aviso/7090313-a1301eu-a-t-reemplazo-ugn3503-sensor-hall-lineal-sip3-disponible-en-capital-federal/>]

4.2 Análisis e implementación de los sensores

4.2.1 LM61CIZ

Este sensor requiere una alimentación de entre 2.7V y 10V, por lo que es posible conectarlo directamente con pilas desde 3V hasta 9V. Tiene una respuesta lineal, por lo que no requiere de algún circuito para la mejora de la señal.

Para tener una señal que se adapte al rango máximo del ADC (3.3V), se puede implementar un circuito de acondicionamiento de señal, basado en amplificadores operacionales, sin embargo, esto aumentaría el costo del prototipo, por ello simplemente se utilizó el circuito de la Fig. 62, con las siguientes características:

- Se alimentó el sensor con una tensión de 5V.
- La salida de voltaje del sensor, se puede conectar directamente con la entrada correspondiente del ADC en el microcontrolador.

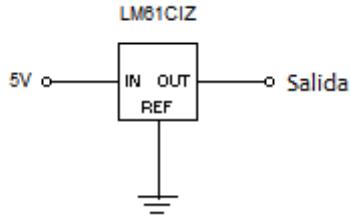


Fig. 62: Conexión del sensor de temperatura LM61CIZ.

4.2.2 MPX10DP

La respuesta de este sensor es lineal, cuenta con 2 salidas de voltaje, las cuales están referidas a la presión atmosférica y al vacío, y con un *voltaje de offset*³⁴, de entre 25mV y 30mV (dependiendo del fabricante), la sensitividad de este sensor es baja (3.5 mV/KPa), por lo que para tener una medición más confiable se puede amplificar.

Para la obtención de una variación confiable en el sensor se debe realizar una diferencia entre dichos voltajes, para lo cual se implementó un circuito basado en el amplificador de instrumentación (AD620), con una ganancia de 50. La ventaja de utilizar una configuración diferencial es la alta inmunidad al ruido debido a su CMRR (Common Mode Rejection Ratio → Modo de Rechazo Común).

Para eliminar el voltaje de offset de la señal, se empleó una configuración de sumador/restador en la segunda etapa de acondicionamiento.

Por otro lado, para tener una señal que se adapte al rango máximo del ADC, se puede implementar un circuito de acondicionamiento de señal adicional, sin embargo, esto aumentaría el costo del prototipo.

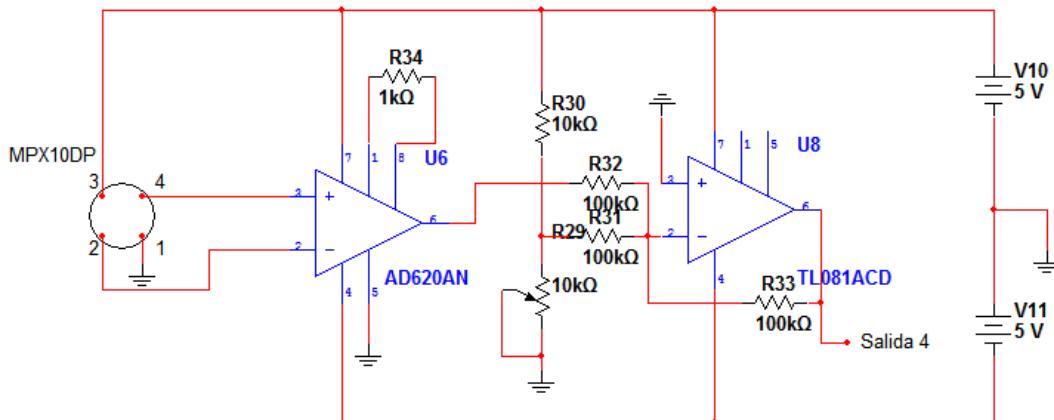


Fig. 63: Acondicionamiento de señal e implementación del sensor de presión MPX10DP.

³⁴ Voltaje de offset: voltaje diferencial que hay que aplicar entre las entradas de un circuito para que su salida valga justamente cero. [<http://www.ele.uva.es/~jesus/eanalogica/apuntes2.pdf>]

Explicación del circuito de la Fig. 63:

- El resistor R34 de 1KΩ ajusta la ganancia del amplificador de instrumentación “AD620” a 50, el cual obtiene a su salida la resta de las 2 salidas voltajes del sensor de presión, amplificada.
- Los resistores R31, R32, y R33 con el AO “TL081” (marcado como U1), en configuración de restador con ganancia unitaria.
- Los resistores R29 y R30 forman un divisor de voltaje, los cuales se conectan a una entrada del amplificador restador, por medio del resistor R31.

4.2.3 HMZ-433A1

La respuesta de este sensor es lineal independientemente de la temperatura en la que se halle, para tener una señal que se adapte al rango total del ADC, se puede implementar un circuito de acondicionamiento de señal, aumentando así el costo del prototipo.

Debido a que el dispositivo también posee un sensor de temperatura cuya salida varía su resistencia con la temperatura, se debe *pasivar*³⁵ la salida para evitar que se pueda causar una mala lectura de la señal del sensor, como se muestra en la Fig. 64.

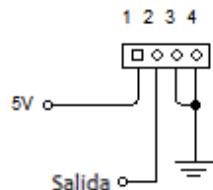


Fig. 64: Configuración del sensor de humedad HMZ-433A1.

4.2.4 UGN3503

La variación de salida de este sensor es lineal con respecto a la magnitud medida, cuenta con un voltaje de offset de entre 2.25V y 2.75V (dependiendo del fabricante), por lo que para eliminar el voltaje de offset de la señal, se empleó una configuración diferencial. En la entrada no inversora se conectó la salida del sensor y en la entrada inversora, se conectó un arreglo de resistores para realizar el ajuste a 0V (ajuste de offset), cuando el sensor se encuentra inactivo.

Nuevamente para tener una señal que se adapte al rango total del ADC, se puede implementar un circuito de acondicionamiento de señal adicional, aumentando el costo del dispositivo.

Explicación de conexión en el diagrama de la Fig. 65:

- Los resistores R1, R5, R6 y R9 junto con el AO “TL081”, forman un amplificador diferencial, utilizado para eliminar el voltaje de offset.

³⁵ Pasivar: Llevar la fuerza electromotriz a cero, o sea cortocircuitar, ya que para cualquier valor de la corriente no debe variar la tensión. [http://www.frba.utn.edu.ar/html/Electrica/archivos/Apuntes_El/Unidad_Tematica_2_Teoremas_de_circuitos.pdf]

- Los resistores R7 y R8 forman un divisor de voltaje, para generar un potencial en la primera entrada del AO a través del resistor R9 hacia la entrada no inversora, y así ajustar la salida a 0V.

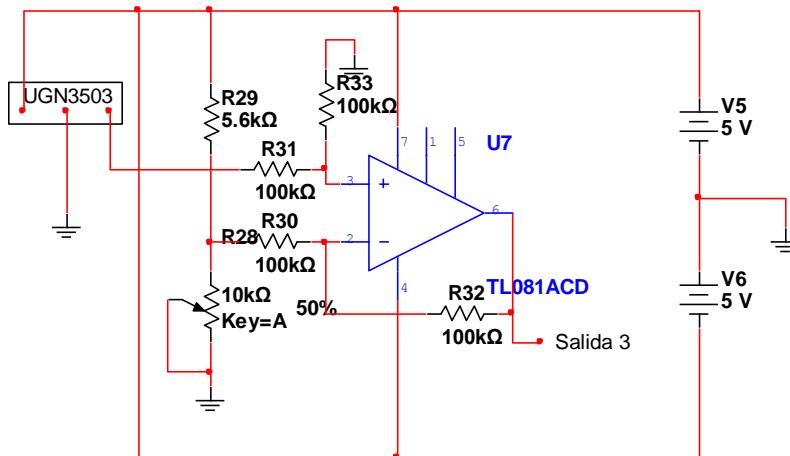


Fig. 65: Implementación del sensor de efecto Hall UGN3503.

4.3 Desarrollo del hardware para la entrada de una señal externa

Debido a que se quiere manejar un rango de voltajes desde $1V_{rms}$ a $120V_{rms}$ o menos, se emplean tres circuitos, puesto que, como se está atenuando la señal con un valor fijo, y sí solo se ocupa una razón de atenuación para todo el rango antes señalado, en voltajes pequeños ($<12V_{rms}$), la señal de salida, será muy pequeña (en el caso de tener $10V_{rms}$, $V_o = \frac{10V_{rms}}{120} = 0.0833V_{rms}$), lo cual no permitirá una buena y correcta lectura del ADC.

4.3.1 Diseño del circuito de obtención de frecuencia

Para obtener la frecuencia de la señal, la solución más simple fue la mostrada por la Fig. 66, cabe señalar que este circuito es aplicado por cada una de las entradas externas que se tienen, es decir, se tiene un circuito de obtención de frecuencia para voltajes de entre 12V y 120V, y otro idéntico para voltajes de entre 100mV a 12V.

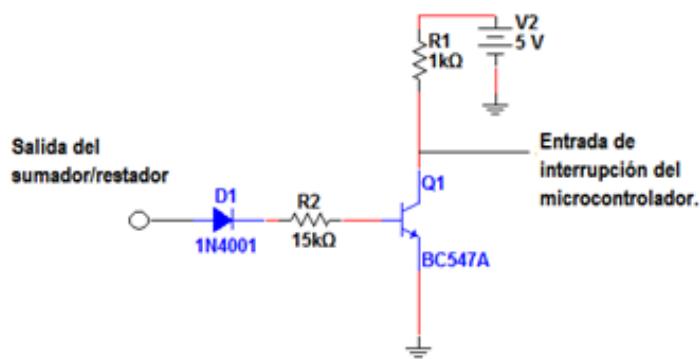


Fig. 66: Diagrama eléctrico para la obtención de la frecuencia de la señal.

El circuito funciona de la siguiente manera:

- El diodo (D1), rectifica la señal para eliminar los voltajes negativos, ya que el ADC del microcontrolador, sólo acepta voltajes positivos para la conversión.
- El resistor R2, sirve para polarizar al transistor y que circule una corriente en la base del mismo.
- El resistor R1, sirve para limitar la corriente máxima que el colector podrá manejar, dependiendo la corriente en la base antes de llegar a saturación.
- El transistor, se utiliza para obtener una señal cuadrada a la salida del colector, con la misma frecuencia de la señal de entrada. Cuando el potencial entre base y emisor rebasa el voltaje de umbral, la corriente aumenta paulatinamente, esto se ve reflejado como un voltaje en el colector, que alcanza un nivel máximo y se mantiene allí, sin importar que la corriente en la base siga aumentando, obteniendo a la salida se tiene un pulso cuadrado.
- Una vez obtenidos los pulsos cuadrados, se envía la señal hacia el microcontrolador, el cual contará dichos pulsos y obtendrá un aproximado de la frecuencia, mandándola como dato hasta la aplicación para ser mostrada.

4.3.2 Circuito para voltajes de entrada de 20 a 120 V_{rms}

El circuito permite recibir señales de entre 20 y 120 V_{rms} de C.A. Está constituido por una etapa de acoplamiento magnético, para la protección del microcontrolador, una segunda etapa de atenuación, utilizando un AO en configuración de sumador/restador, para agregar un voltaje de offset a la señal atenuada, ya que el microcontrolador debe recibir sólo señales de voltajes positivos.

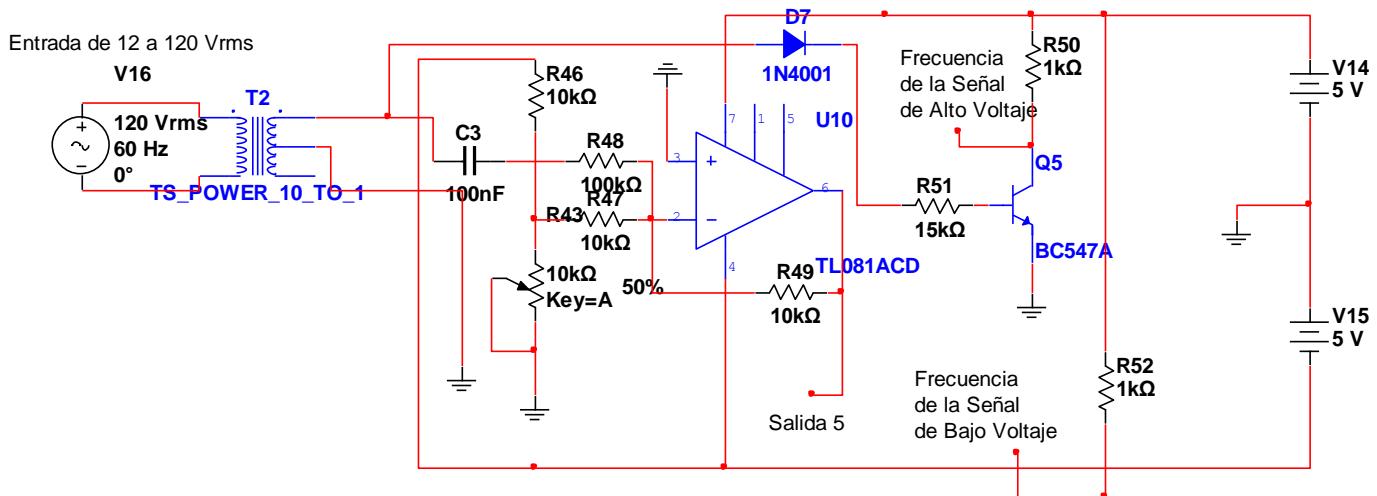


Fig. 67: Diagrama eléctrico, para entrada de señal externa entre 20 y 120 V_{rms}.

Explicación del circuito de la Fig. 67:

- El transformador (T1) tiene una relación 10:1 con respecto a la derivación central³⁶, para atenuar (disminuir) el valor máximo indicado a $20 V_{rms}$.
- El capacitor C1, se encarga de bloquear la señal de CD que pudiese contener la señal de entrada externa a medir, y dejar fuera de rango a la señal.
- Los resistores R2 y R3 son utilizados para atenuar la señal a una razón de 1:12 para adaptarse a la escala del ADC.
- Los resistores R4 y R5 son utilizados para generar el voltaje de CD, y así ajustar la señal desde 0V hasta el voltaje de referencia del ADC (3.3V), el resistor R5 es un potenciómetro de precisión (trimpot).
- El resistor R1 es del mismo valor que R3 para que no se atenue ni amplifique la señal de CD.

4.3.3 Circuito para voltajes de entrada de 3 a $20 V_{rms}$

Este módulo consta de un AO sumador/restador con una razón de atenuación de 1:12, y con un selector de voltaje de CD para ajustar la señal a las condiciones del microcontrolador (voltajes positivos).

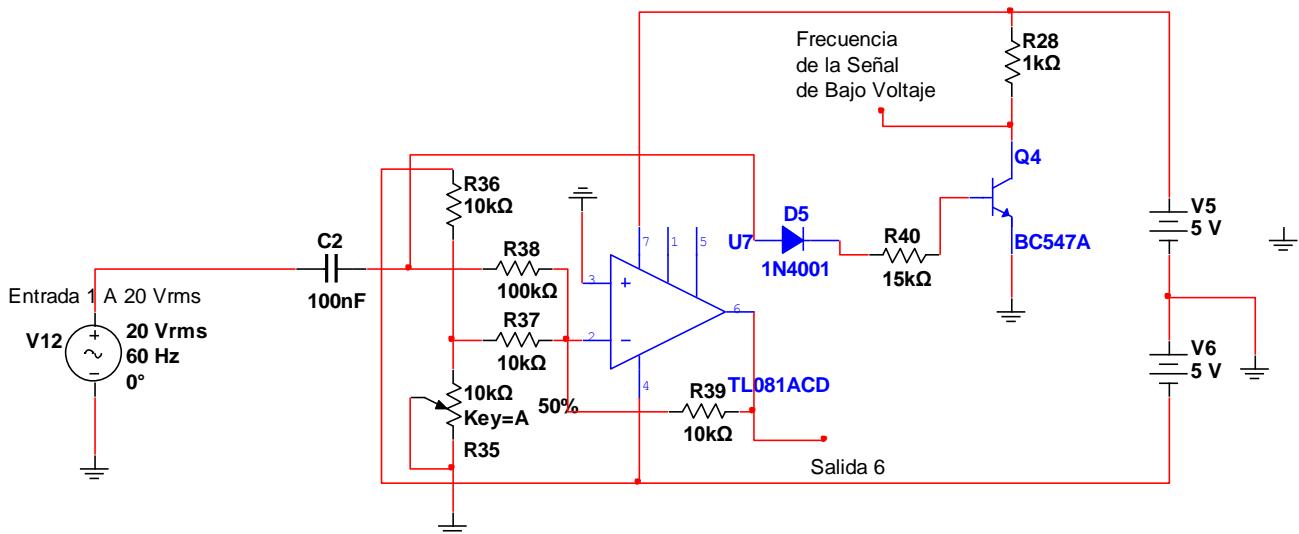


Fig. 68: Diagrama eléctrico, para entrada de señal externa entre 3 y $20 V_{rms}$.

Explicación del circuito mostrado en la Fig. 68:

- Los resistores R2 y R3 son utilizados para obtener la razón 1:12 de atenuación.
- Los resistores R4 y R5 sirven para generar el voltaje de CD, y el resistor R1 sirve para agregar la señal al AO sumador/restador.

³⁶ Derivación central: Es una conexión adicional en el bobinado secundario de un transformador, que divide la tensión (voltaje) en este bobinado en dos voltajes iguales. Esta conexión adicional se pone a tierra.

[http://www.unicrom.com/Tut_rectificador_onda_completa.asp]

4.3.4 Circuito para voltajes de entrada de 300 mV_{rms} a 3 V_{rms}

Ya que el usuario, puede requerir graficar señales directamente de un microcontrolador, que funcione con 3.3V, sin la necesidad de atenuarla, se añadió una entrada analógica más al sistema, así como el circuito de obtención de frecuencia, a través de una interrupción externa como se muestra en la Fig. 69.

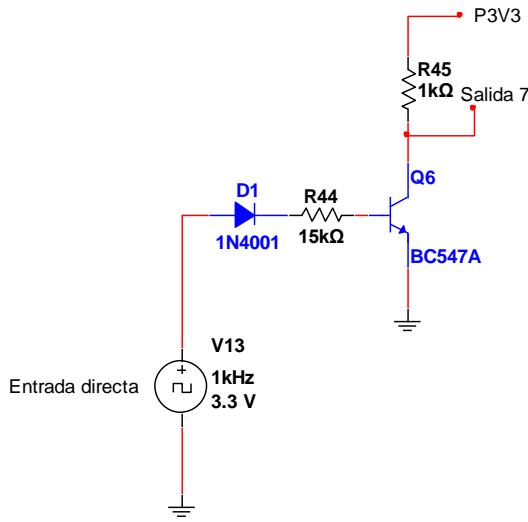


Fig. 69: Diagrama eléctrico, para entrada de señal externa entre 300 mV_{rms} y 3 V_{rms} .

4.4 Diseño del prototipo

4.4.1 Diagrama eléctrico de conexión

El diagrama de conexión final, se realizó integrando cada uno de los diagramas individuales mostrados anteriormente (conexión de cada sensor y del módulo para la entrada de señal externa), marcando y teniendo muy en claro las salidas de cada circuito individual. En la Fig. 70, se muestra el esquemático completo del prototipo.

4.4.2 Circuito impreso

Utilizando el software de desarrollo “Ares”, se realizó la placa del prototipo. En las Fig. 71 y 72, se muestran tanto el diseño de las pistas, como el diseño real de la placa basada en el diagrama de la Fig. 70.

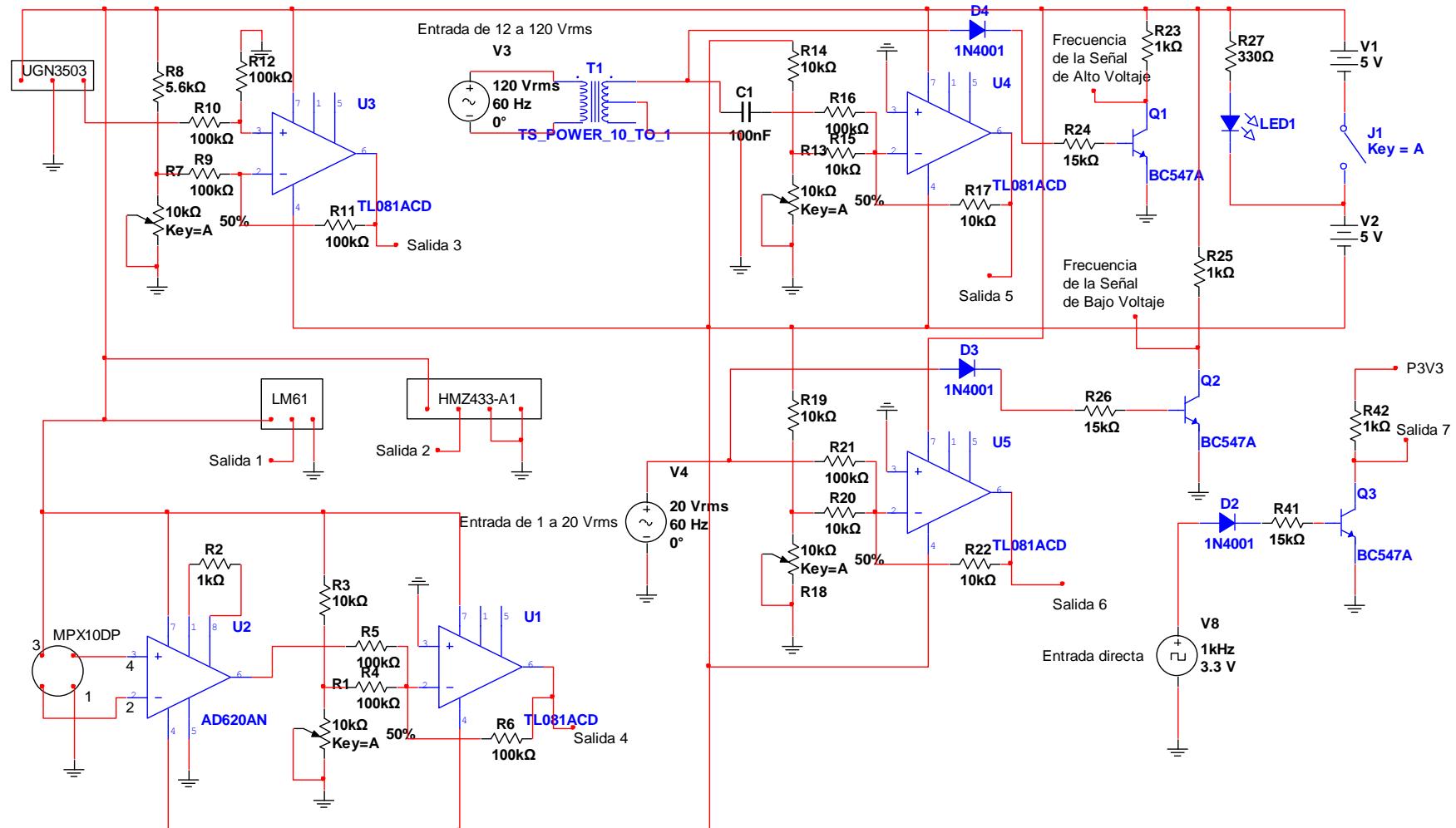


Fig. 70: Diagrama esquemático del prototipo.

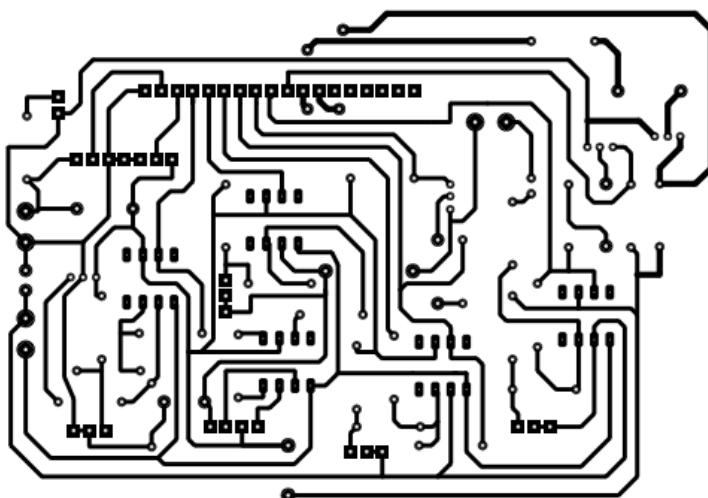


Fig. 71: Diseño de las pistas del PCB del prototipo.

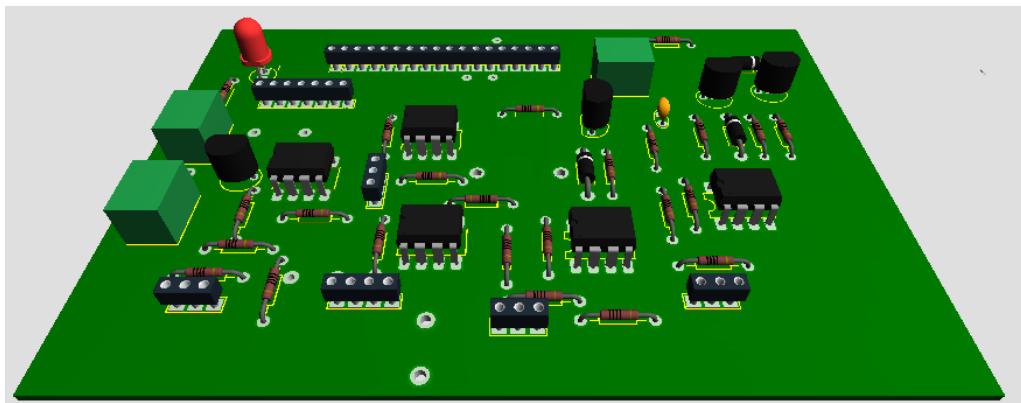


Fig. 72: Diseño de la placa del prototipo con los dispositivos correspondientes.

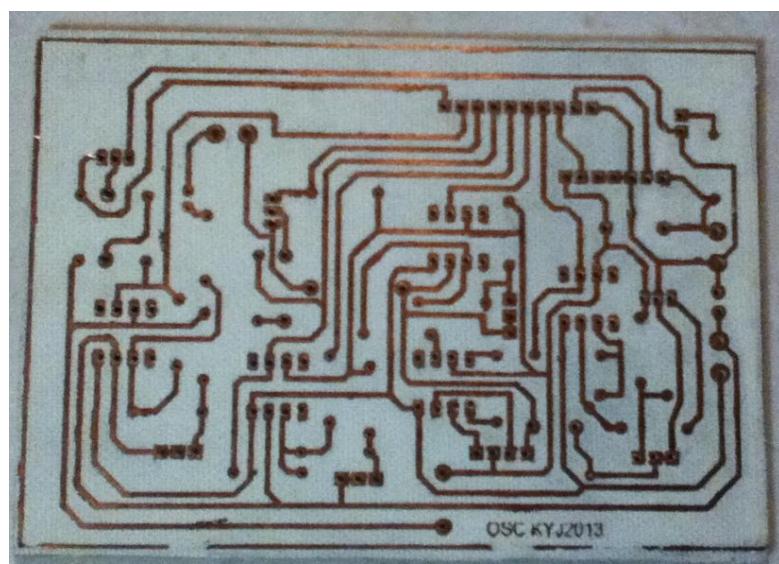


Fig. 73: Placa del prototipo.

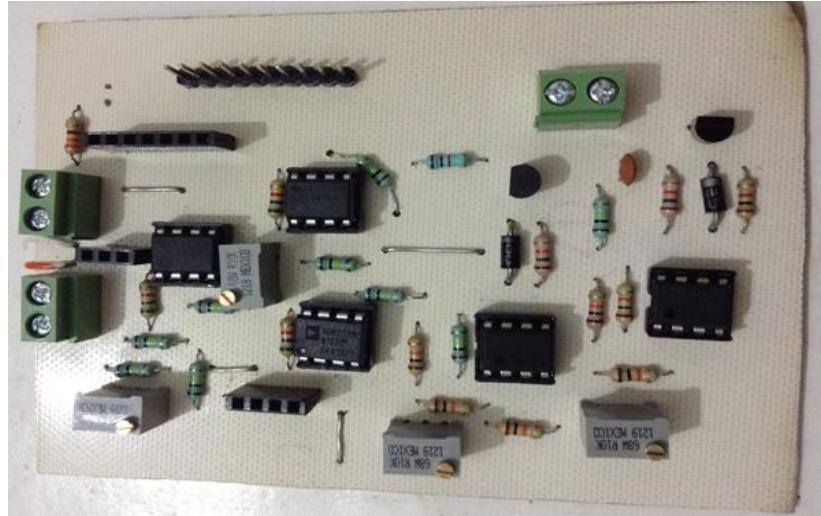


Fig. 74: Placa del prototipo con los dispositivos soldados.

4.5 Fuente simétrica 5V

Para la correcta alimentación del prototipo, se diseñó una fuente simétrica de ± 5 V, debido al requerimiento de los circuitos implementados con amplificadores operacionales, en un principio se planteó utilizar pilas para la alimentación, sin embargo, el diseño requirió de más corriente, por lo tanto, se diseñó el circuito siguiente:

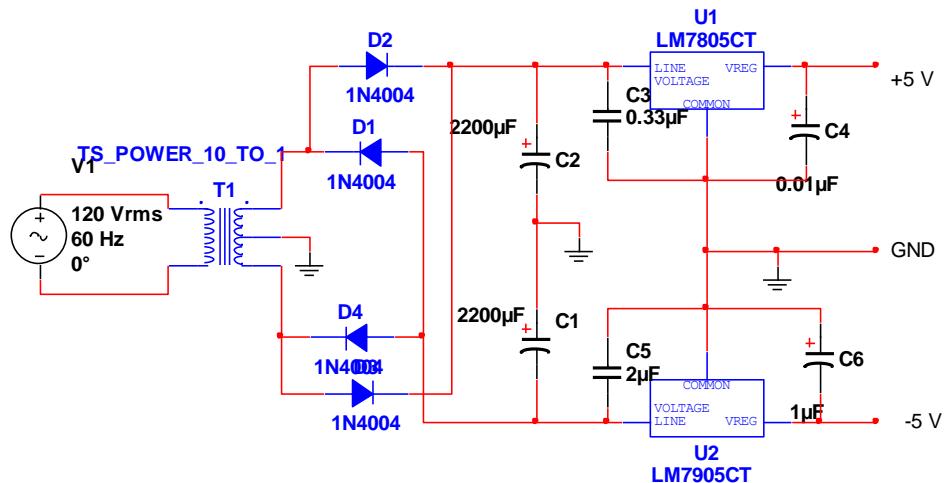


Fig. 75: Diagrama de conexión eléctrica de la fuente simétrica de 5V para alimentación de AO.

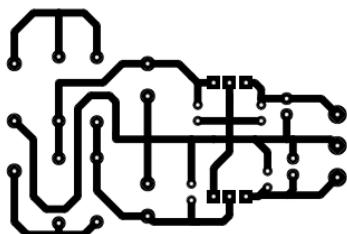


Fig. 76: Pistas para la placa de la fuente simétrica de 5V.



Fig. 77: Placa de la fuente simétrica de 5V, sin transformador.

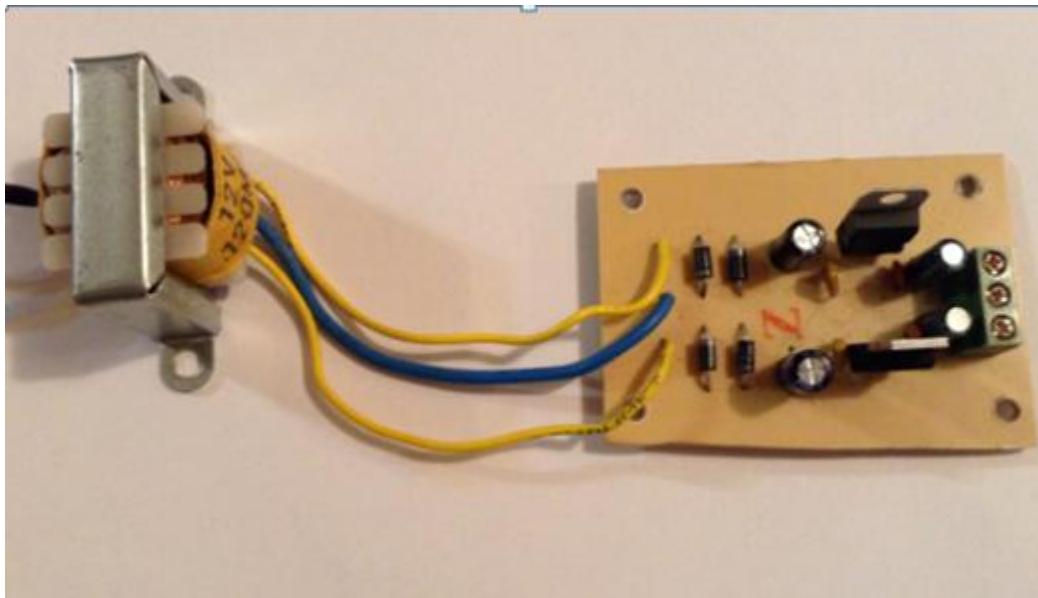


Fig. 78: Placa de la fuente simétrica de 5V, con el transformador conectado.

4.6 Módulo de comunicación serial inalámbrica (Bluetooth HC06)

Para la comunicación serial inalámbrica del sistema con la PC, se eligió el módulo Bluetooth HC06 debido a su bajo costo, y a su fácil conectividad con el microcontrolador. Simplemente se conectaron los pines del Bluetooth de la siguiente manera (Fig. 79):

- Los pines Vcc y Key, ambos al pin P3V3 del microcontrolador.
- El pin GND del Bluetooth (HC06) al pin GND del microcontrolador.
- El pin TX al pin RX del microcontrolador.
- El pin RX al pin TX del microcontrolador.

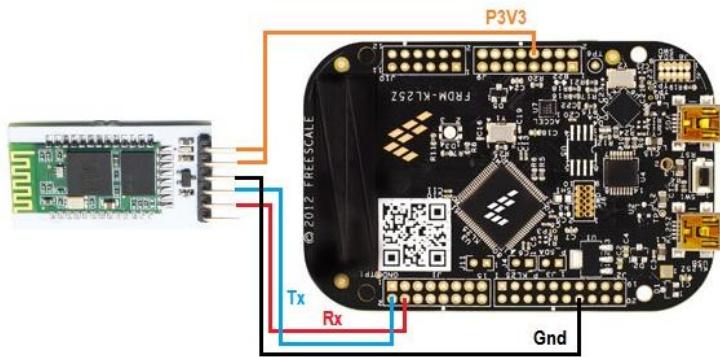


Fig. 79: Diagrama eléctrico de conexión del módulo Bluetooth.

Para poder adaptar el módulo directamente al microcontrolador, así como los sensores y salidas de las diversas señales analógicas, se diseñó en el programa ARES la siguiente placa:

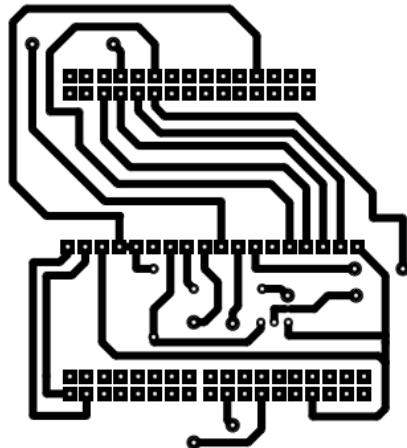


Fig. 80: Diseño de la placa del módulo Bluetooth HC06.

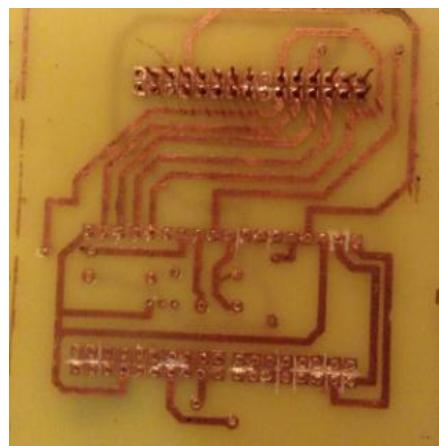


Fig. 81: Placa de adaptación, para módulo Bluetooth HC06.

Capítulo 5: Desarrollo del Software

5. Desarrollo del software

5.1 Desarrollo del software para el microcontrolador

5.1.1 Entorno de desarrollo CodeWarrior y herramienta ProcessorExpert

Para la programación del microcontrolador, se utilizó un software de desarrollo de la misma compañía Freescale, debido a su compatibilidad con el microcontrolador empleado, y su fácil manejo en cuestión de programación, tal software es llamado CodeWarrior.

CodeWarrior, nos permite utilizar una poderosa herramienta denominada “ProcessorExpert”, la cual maneja una estructura de programación por “componentes”. Dichos componentes, constan de los diversos subsistemas que contiene el microcontrolador, como:

- ADC.
- DAC.
- Temporizador.
- Control de PWM (Pulse Width Module → Modulación por Ancho de Pulso).
- Interfaces (SPI, Serial, USB, I2C).
- GPIO (General Propose Input-Output → Entrada-Salida de Propósito General).
- Interrupción externa.

Cada uno de los componentes de los subsistemas, cuenta con “métodos” para realizar una acción sin la necesidad de que el usuario cree un código complejo, por ejemplo, la lectura y escritura de un dato en un puerto, la lectura del ADC, comunicación de la interfaz I2C e incluso las interrupciones, para todo ello, no se requiere hacer un extenso estudio de la hoja de especificaciones, debido a que los métodos de cada subsistema contiene un menú de ayuda al usuario, y diversos ejemplos de cómo emplearlo.

Por otra parte, cabe mencionar que es necesario conocer los subsistemas del microcontrolador más a detalle, para tener mejor noción de como conectar y de las terminales a utilizar en cada proceso, y en cada módulo a controlar.

A continuación, se muestra una breve descripción de la interfaz, en el software de desarrollo CodeWarrior y ambiente de trabajo de la herramienta ProcessorExpert.

5.1.1.1 Ambiente del software CodeWarrior

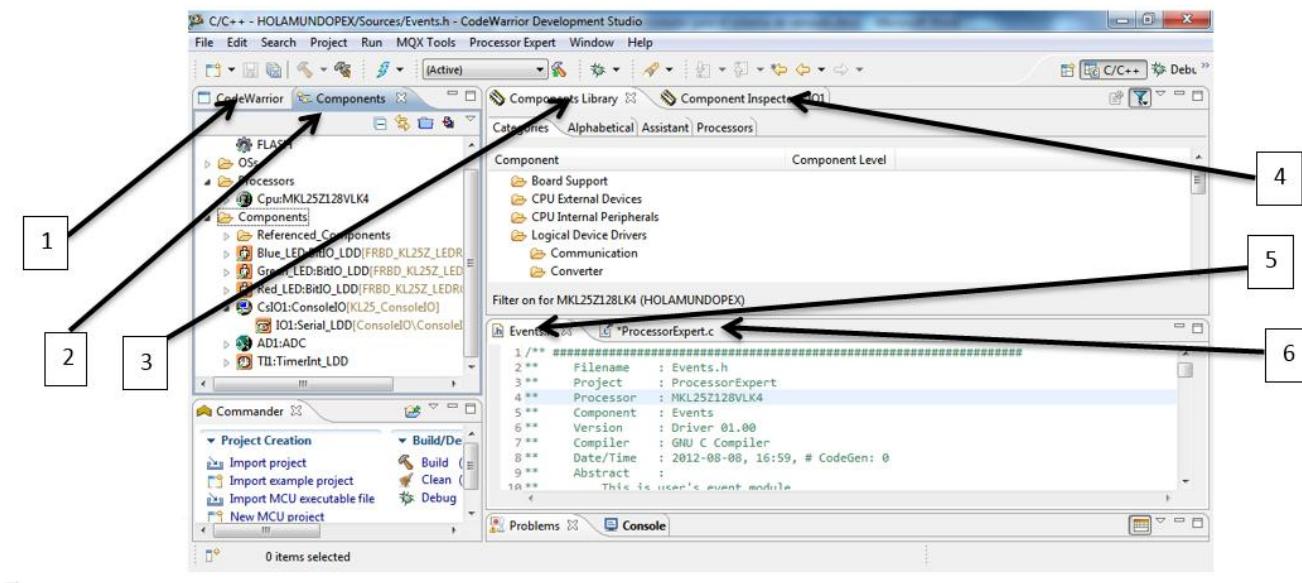


Fig. 82: Entorno CodeWarrior.

Los elementos más importantes y a identificar dentro del software de desarrollo CodeWarrior, son los marcados en la Fig. 82 y su explicación es la siguiente:

1. La pestaña de “CodeWarrior”, contiene todos los archivos fuente, cabeceras, códigos generados por el “ProcessorExpert” y todos los documentos relacionados con el proyecto creado, en tal software.
2. La pestaña de “Components”, contiene los componentes utilizados en el proyecto, y los diversos métodos correspondientes a cada uno de ellos.
3. La pestaña de “Components Library”, contiene los diversos componentes soportados por el microcontrolador, es decir, los posibles subsistemas o módulos a agregar y a manejar a través de la herramienta ProcessorExpert.
4. La pestaña de “Components Inspector”, es utilizada para la configuración, inicialización y habilitación de los componentes a emplear, una vez que estos ya han sido agregados al proyecto, es decir, ya se encuentran visibles en la ventana “Components”.
5. La pestaña de “Events.c”, es el código que genera el ProcessorExpert para los métodos de los diferentes subsistemas utilizados, y es posible abrirla siguiendo la ruta:
 - Pestaña CodeWarrior.
 - Click en el nombre del proyecto creado.
 - Carpeta “Sources”
 - Doble click en “Events.c”
6. La pestaña de “ProcessorExpert.c”, contiene el código del programa principal a ser desarrollado, y puede ser abierta siguiendo la ruta mostrada en el paso anterior, sólo como último paso, dar doble click en “ProcessorExpert.c”.

- Pestaña CodeWarrior.
- Click en el nombre del proyecto creado.
- Carpeta “Sources”
- Doble click en “ProcessorExpert.c”

5.1.2 Elección de los componentes usados en CodeWarrior, para el desarrollo del programa del microcontrolador

Para la elección de los componentes, primeramente se planteó un objetivo y la solución al mismo, sólo así se pudo iniciar el diseño del programa.

Objetivo: *Diseñar un programa que sea capaz de realizar la lectura de cuatro sensores y de dos señales externas analógicas y/o digitales, así como la obtención de las frecuencias de estas últimas, a partir de una instrucción (dato) recibida mediante la comunicación serial con la PC.*

Solución: *El programa requiere de una terminal de comunicación serial para poder recibir y enviar datos a la PC, debe contar con un módulo de conversión analógico-digital, con la posibilidad de medir siete canales (cuatro para sensores y tres para señales externas), tres entradas de interrupción externas, que utilicen, y un temporizador a un segundo, que registre el número de pulsos por segundo (frecuencia), de cada una de las señales externas.*

Para agregar los componentes, se seleccionan en la pestaña “Categories” de “Components Library”, los siguientes elementos (Fig. 83), con la ruta indicada:

- Un componente ConsoleIO, el cual permite la comunicación del microcontrolador con una terminal serial, como PuTTy, TeraTerm e Hyperterminal, o en este caso, la PC.
 - Carpeta “CPU External Devices”
 - Carpeta “Display”
 - Doble click en “ConsoleIO”
- Un componente ADC, que permite utilizar el ADC integrado en el microcontrolador.
 - Carpeta “CPU Internal Peripherals”
 - Carpeta “Converter”
 - Carpeta “ADC”
 - Doble click en elemento “ADC”
- Tres componentes ExtInt, que permite generar una interrupción externa, para el conteo de flancos.
 - Carpeta “Logical Device Drivers”
 - Carpeta “Interrupts”
 - Doble click en elemento “ExtInt_LDD”
- Un componente TimerInt, que agrega un temporizador, así como las interrupciones del mismo.
 - Carpeta “Logical Device Drivers”
 - Carpeta “Timer”

- Doble click en elemento “TimerInt_LDD”

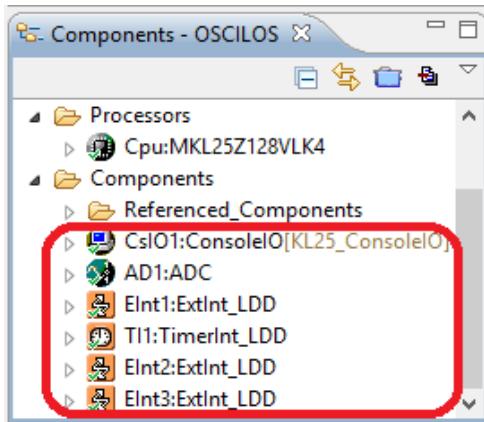


Fig. 83: Elementos agregados, utilizados en el software CodeWarrior.

5.1.3 Configuración de los componentes empleados

Una vez que los componentes se agregaron al proyecto, se configuró cada uno, para que éstos pudieran cumplir con su función. Para ello se selecciona el componente y se elige la pestaña de “Components Inspector” y “Properties”, en ésta aparecen diversas indicaciones para la configuración del componente. Las modificaciones de cada elemento, se muestran a continuación.

5.1.3.1 Componente ConsoleIO

Este componente tiene asociado un componente IO1:Serial_LDD, el cual es utilizado para generar la comunicación con la terminal de la PC, y cuenta con muchas opciones de configuración como lo muestra la Fig. 84, pero sólo fueron modificadas las opciones siguientes:

- Device: Esta opción permite seleccionar hasta 3 UART (Receptor-Transmisor Asíncrono Universal). La seleccionada fue la UART0, ya que ésta corresponde a la comunicación requerida con la PC.
- Data width: Esta opción por default se encuentra en 8 bits, que es la trama utilizada en las diversas comunicaciones seriales como la SPI e I2C, y ésta no fue modificada.
- Stop bits: Ésta permite seleccionar el número de bits de paro que se desea en la comunicación, en este caso 1 bit.
- Baud Rate: Permite seleccionar la frecuencia que se manejará en la comunicación, en este caso se utilizó una tasa de 115200 baudios.
- Idle line mode: Indica si la comunicación iniciará antes del bit de inicio o después del bit de paro. En este caso, se propuso que inicie después del bit de inicio.
- RxD: En esta opción se eligen la terminal del microcontrolador que fungirá como receptor, para la UART0. La terminal asociada es PTA1/UART0_RX.

- TxD: Se seleccionó el pin del microcontrolador que funge como transmisor, para la UART0 la terminal asociada es PTA2/UART0_TX.

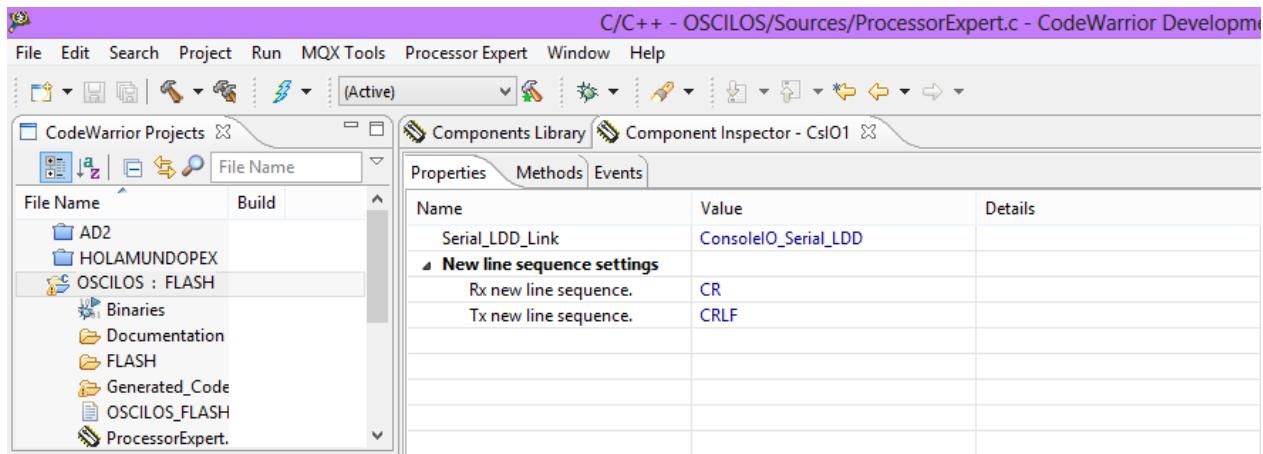


Fig. 84: Configuración del componente de serial.

5.1.3.2 Configuración del componente ADC

Éste se utiliza para la obtención de los datos analógicos de los cuatro sensores y las dos señales externas, las opciones utilizadas para la configuración fueron las siguientes:

- A/D channels: Esta opción permite elegir el número de canales analógicos que se desea, en este caso se eligió 7.
- Channel0 A/D channel: Aquí se eligió la terminal del microcontrolador, que se decidió emplear para el canal 0, siempre y cuando el software lo autorizara, ya que puede estar ocupado por algún otro dispositivo o subsistema. La terminal seleccionada fue PTE21.
- Channel1 A/D channel: Terminal seleccionada → PTE22
- Channel2 A/D channel: Terminal seleccionada → PTE23
- Channel3 A/D channel: Terminal seleccionada → PTB0
- Channel4 A/D channel: Terminal seleccionada → PTB2
- Channel5 A/D channel: Terminal seleccionada → PTE20
- Channel6 A/D channel: Terminal seleccionada → PTB3

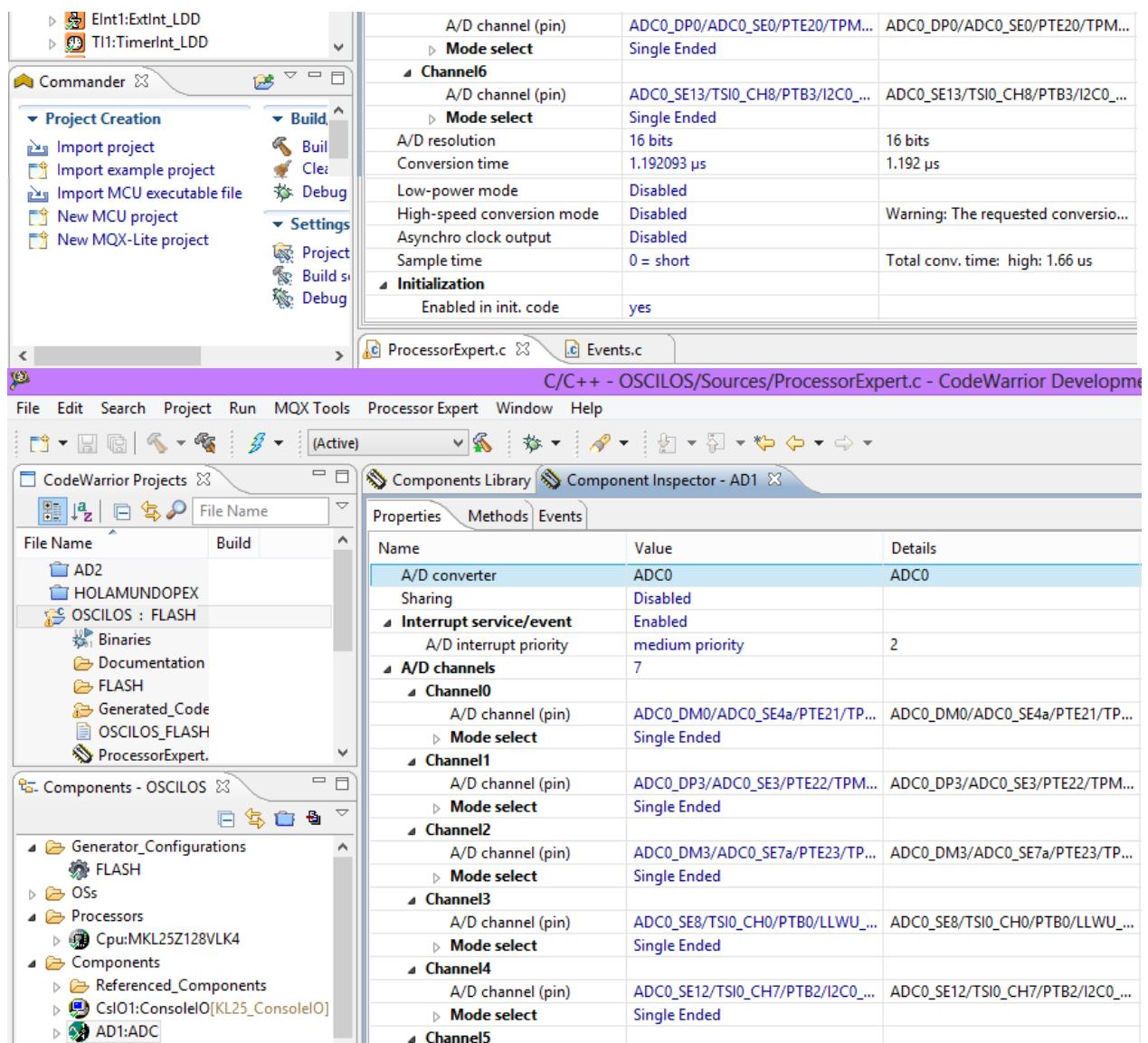


Fig. 85: Configuración del componente ADC.

5.1.3.3 Configuración de las interrupciones externas (ExtInt)

Aquí se configura una por una, las entradas de interrupción externa, para la obtención de la frecuencia, de cada una de las señales externas, las opciones de configuración que se emplearon son las siguientes:

- Pin: Aquí se eligió una terminal de la tarjeta de desarrollo, como entrada de interrupción externa. En este caso, debido a que son tres interrupciones, y que las todas se configuraron de la misma manera, las terminales correspondientes son: PTD2, PTD5 y PTD6.
- Generate interrupt on: Aquí se eligió el tipo de flanco que soporta la interrupción, en este caso se eligió flanco de subida (*rising edge*).

- Auto initialization: Se seleccionó “yes”, ya que se desea inicializar el componente desde el principio, es decir, que la cuenta comience desde que se alimenta el microcontrolador.

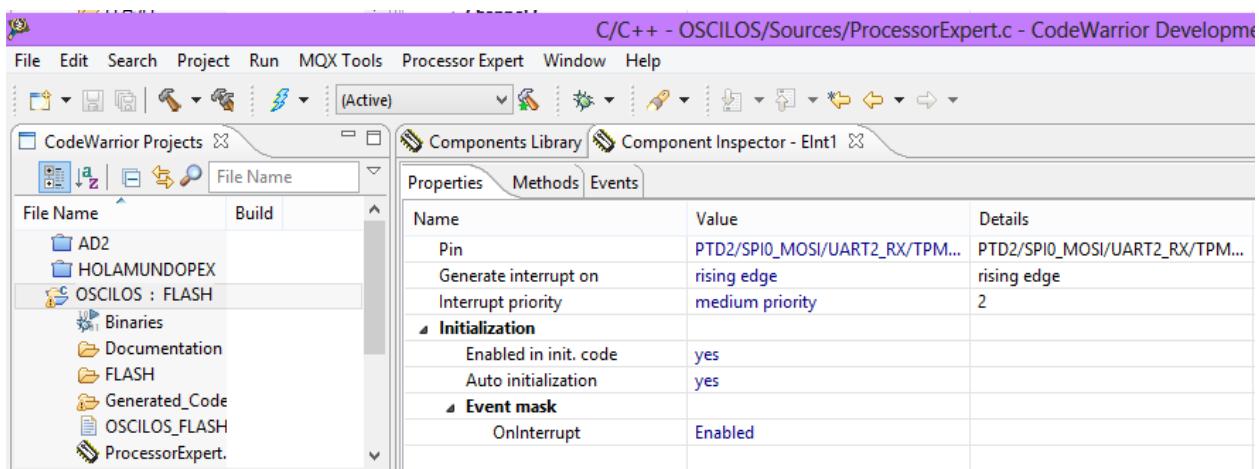


Fig. 86: Configuración del componente de interrupción externa ExtInt1.

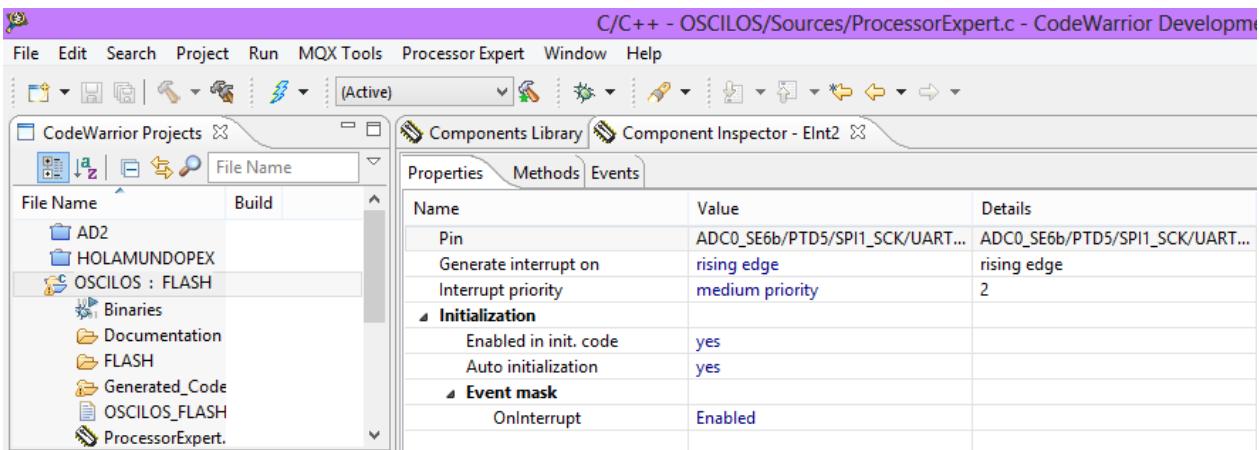


Fig. 87: Configuración del componente de interrupción externa ExtInt2.

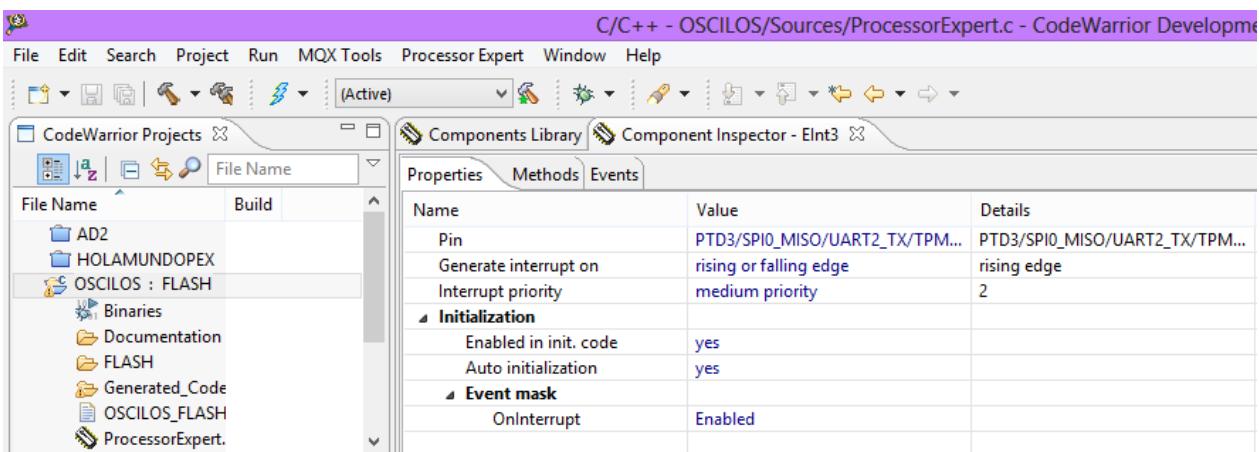


Fig. 88: Configuración del componente de interrupción externa ExtInt3.

5.1.3.4 Configuración del temporizador

Se configuró el temporizador de la siguiente manera (Fig. 89):

- Interrupt Period: Se seleccionó el valor hasta el que se desea que cuente el timer, en este caso, fue un segundo, ya que es el periodo en donde se cuentan los flancos para la obtención de la frecuencia.
- Auto initialization: Se seleccionó “yes”, ya que se desea inicializar el componente desde el principio, es decir, que la cuenta comience desde que se alimenta el microcontrolador.

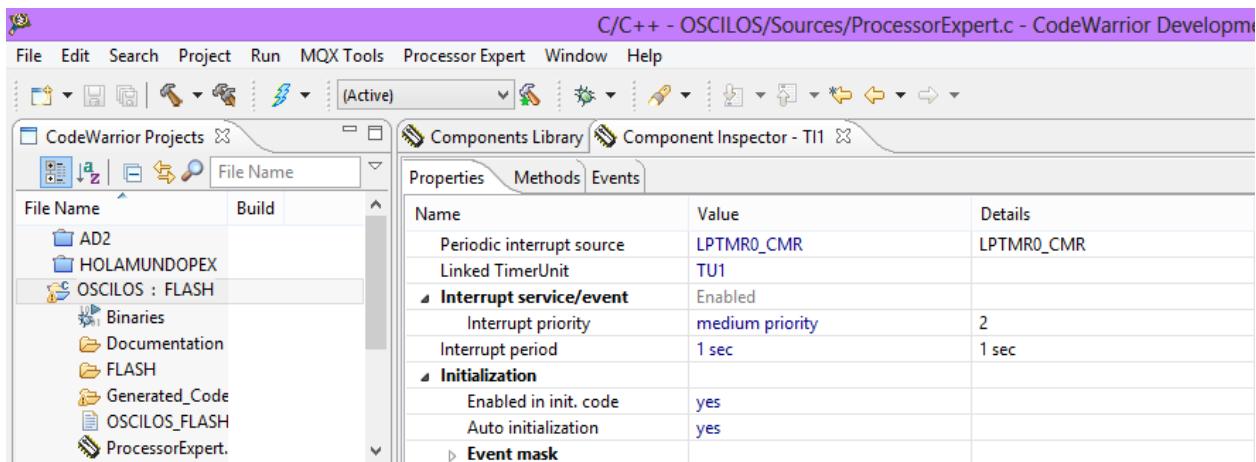


Fig. 89: Configuración del componente Timer.

5.1.4 Algoritmo de funcionamiento para el microcontrolador

El proceso llevado a cabo por el microcontrolador, es mostrado en el diagrama de flujo de la Fig. 89, y su explicación es la siguiente:

1. El microcontrolador espera hasta que se haya recibido algún carácter por el UART.
2. Una vez que se ha detectado un carácter, se verifica cuál es, y en base a este, se realiza un proceso de acuerdo a la Tabla 5.
3. Cuando dicho proceso se termina, el dato resultante de éste, es enviado por el UART, hasta la terminal de la PC, vía Bluetooth.

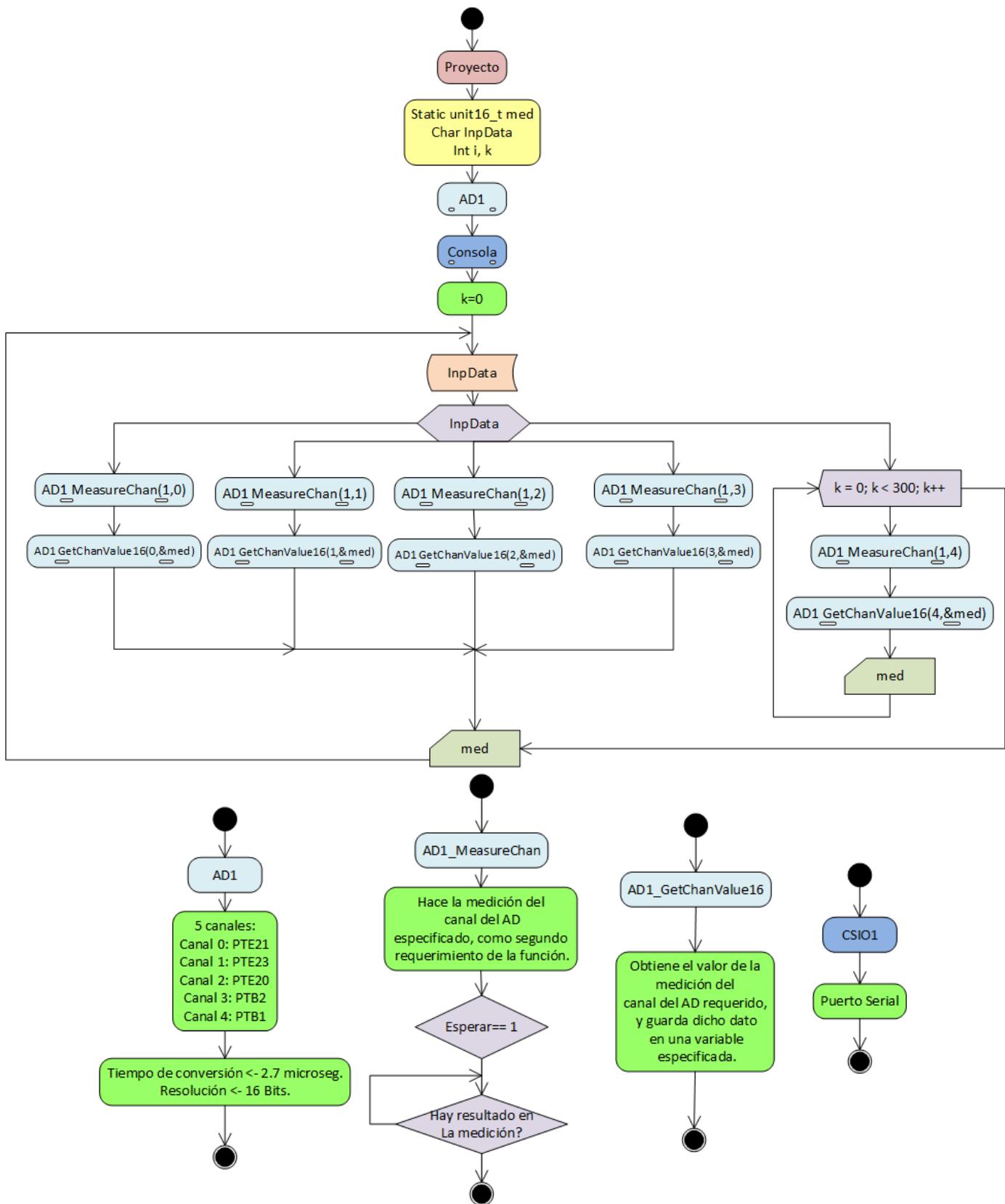


Fig. 90: Diagrama UML para el proceso a ejecutar por el microcontrolador.

Carácter detectado	Proceso	Carácter detectado	Proceso
'A'	Adquisición de frecuencia, tomado por la EInt1, correspondiente a señales de 20 – 120 V_{rms}	'H'	Toma de 50 muestras del sensor de humedad, y cálculo del promedio eliminando picos.
'B'	Adquisición de frecuencia, tomado por la EInt2, correspondiente a señales de 3 – 20 V_{rms}	'I'	Toma de 300 muestras de señales entre 3 – 20 V_{rms} a frecuencias mayores de 150KHz
'C'	Adquisición de frecuencia, tomado por la EInt3, correspondiente a señales de 300 mV_{rms} – 3 V_{rms}	'J'	Toma de 300 muestras de señales entre 300 mV_{rms} – 3 V_{rms} a frecuencias mayores de 150KHz
'D'	Toma de 300 muestras de señales entre 20 – 120 V_{rms} a frecuencias menores de 150KHz	'M'	Toma de 50 muestras del sensor de efecto Hall, y cálculo del promedio eliminando picos.
'E'	Toma de 300 muestras de señales entre 3 – 20 V_{rms} a frecuencias menores de 150KHz	'P'	Toma de 50 muestras del sensor de presión, y cálculo del promedio eliminando picos.
'F'	Toma de 300 muestras de señales entre 300 mV_{rms} – 3 V_{rms} a frecuencias menores de 150KHz	'T'	Toma de 50 muestras del sensor de temperatura, y cálculo del promedio eliminando picos.
'G'	Toma de 300 muestras de señales entre 20 – 120 V_{rms} a frecuencias mayores de 150KHz	---	---

Tabla 5: Correspondencias carácter-proceso.

5.1.5 Creación del código fuente, para el funcionamiento del microcontrolador

Para poder crear el código fuente, primero se eligieron los métodos a utilizar de cada componente. Los métodos de cada componente, se encuentran en la ventana “Components”, se despliega el contenido de la carpeta “Components”, y por último, se despliega el contenido del elemento, del cual se requiere algún método. Debido a que cada elemento tiene más de un método, a continuación, se explica lo correspondiente a los eventos requeridos para el proyecto:

1. Para el componente ADC, se requiere de:

- Un método que lea un canal seleccionado → AD1_MeasureChan
- Un proceso que obtenga el dato del canal medido → AD1_GetChanValue16

El método AD1_MeasureChan requiere como argumentos un dato tipo “bool”, que indique el fin de la conversión y un dato tipo “byte” que indique el canal a medir.

El método AD1_GetChanValue16 requiere como argumentos un dato tipo “byte”, que indique el canal cuyo dato leído se quiere obtener, y una variable tipo “word” para guardar dicho valor, la variable debe ser un entero de 16 bits para poder guardar el resultado del ADC.

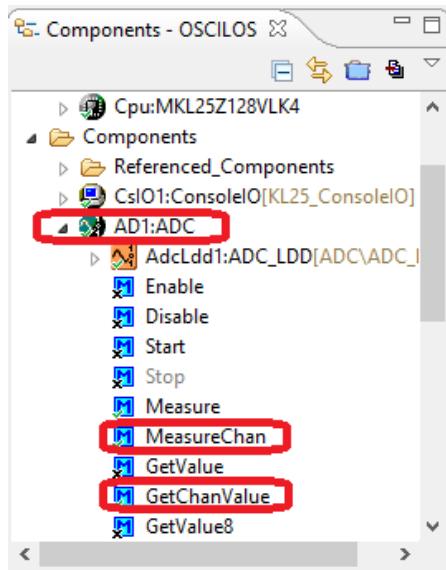


Fig. 91: Métodos usados, del componente ADC.

2. Para el componente de Interrupción externa, existen varios métodos, pero lo que se necesita en este caso, es detectar un flanco de subida en la terminal asignada para la interrupción (PTD2 para EInt1, PTD5 para EInt2 y PTD6 para EInt3), entonces, se utilizó para la entrada de interrupción externa 1, EInt1_OnInterrupt, como se puede observar en la Fig. 92, y de la misma manera, se hizo con EInt2 y EInt3.

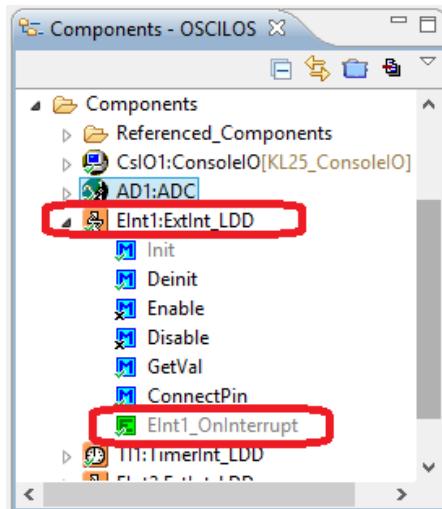


Fig. 92: Métodos del componente de interrupción externa.

3. Para el componente Timer, simplemente se seleccionó el método de interrupción (TI1_OnInterrupt), para cuando la cuenta programada (1s) haya concluido, obtener los pulsos contados por las interrupciones externas y obtener la frecuencia de las señales.

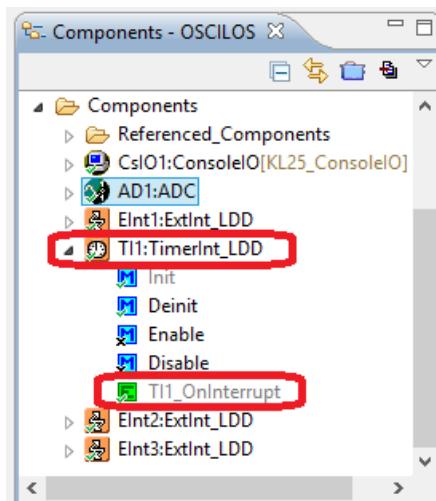


Fig. 93: Métodos del componente Timer.

Una vez elegidos los componentes se genera el código correspondiente a los mismos, por medio de la opción “Generate Processor Expert Code”, localizada en la parte superior derecha de la ventana “Components”, se dio click en el ícono mostrado en la Fig. 94.

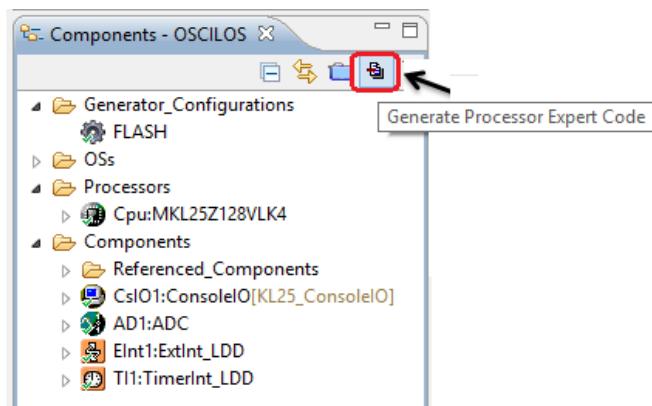


Fig. 94: Generación del código para cada uno de los componentes y métodos usados.

Se abrió la ventana de ProcessorExpert.c y se procedió a generar el código ([Anexo C.1](#)), según el objetivo planteado en la [página 103](#), y el diagrama UML (Unified Modeling Language → Lenguaje Unificado de Modelado), creado para el funcionamiento (Fig. 90), teniendo cuidado en las terminales asignadas, de acuerdo a las correspondencias mostradas en la tabla 6.

Terminal de la tarjeta FRDM-KL25Z	Subsistema al que pertenece:	Correspondiente a la salida del circuito de:
PTB0	ADC canal 3	Presión (Fig. 63)
PTB2	ADC canal 4	Señal externa de 20 a 120 V_{rms} (Fig. 67)
PTB3	ADC canal 6	Señal externa de 750 mV_{rms} a 3 V_{rms} (Fig. 69)
PTD2	EInt1	Obtención de Frecuencia señal externa de 20 a 120 V_{rms} (Fig. 66)
PTD5	EInt2	Obtención de Frecuencia señal externa de 3 a 20 V_{rms} (Fig. 66)
PTD6	EInt3	Obtención de Frecuencia señal externa de 750 mV_{rms} a 3 V_{rms} (Fig. 66)
PTE20	ADC canal 5	Señal externa de 3 a 20 V_{rms} (Fig. 68)
PTE21	ADC canal 0	Temperatura (Fig. 62)
PTE22	ADC canal 1	Humedad (Fig. 64)
PTE23	ADC canal 2	Efecto Hall (Fig. 65)

Tabla 6: Correspondencias terminal-subsistema-circuito.

Una vez creado el programa principal, se creó el código para cada uno de los métodos expuestos anteriormente ([Anexo C.2](#)), esto se llevó a cabo por medio de la ventana “Events.c”.

5.1.6 Grabado del microcontrolador

Se compila el programa, dando click en el ícono del martillo, que se encuentra en la parte superior izquierda de la ventana del programa CodeWarrior, tal como lo muestra la Fig. 95.

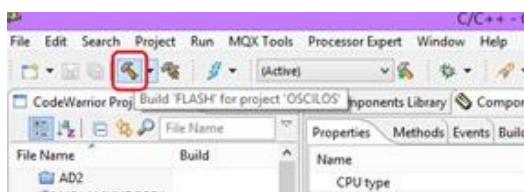


Fig. 95: Compilación del programa creado en el software CodeWarrior.

Se verificó que no hubiera errores en el código y posteriormente, se conectó la tarjeta FRDM-KL25Z por medio de un cable USB y una vez detectado por la computadora, se dio click en el ícono de la flecha hacia abajo, posicionada a un lado del ícono de un insecto en la parte superior de la ventana del software CodeWarrior (Fig. 96).

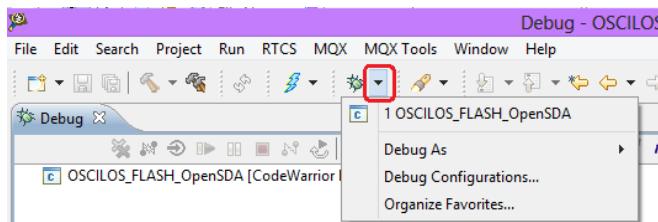


Fig. 96: Grabado del microcontrolador.

Se dio click en la opción “Debug Configurations...” y en la ventana que se abre, se seleccionó el nombre del proyecto con la opción de OpenSDA y click en el botón “Debug” (Fig. 97). Una vez hecho esto, se esperó a que el proceso de grabado terminara y en consecuencia, se abriera una nueva pestaña que contiene los datos de la depuración. Por último, se dio click en el botón “play” de la pestaña que se abrió, como se muestra en la Fig. 98, de esta manera, el programa se ejecuta en el microcontrolador.

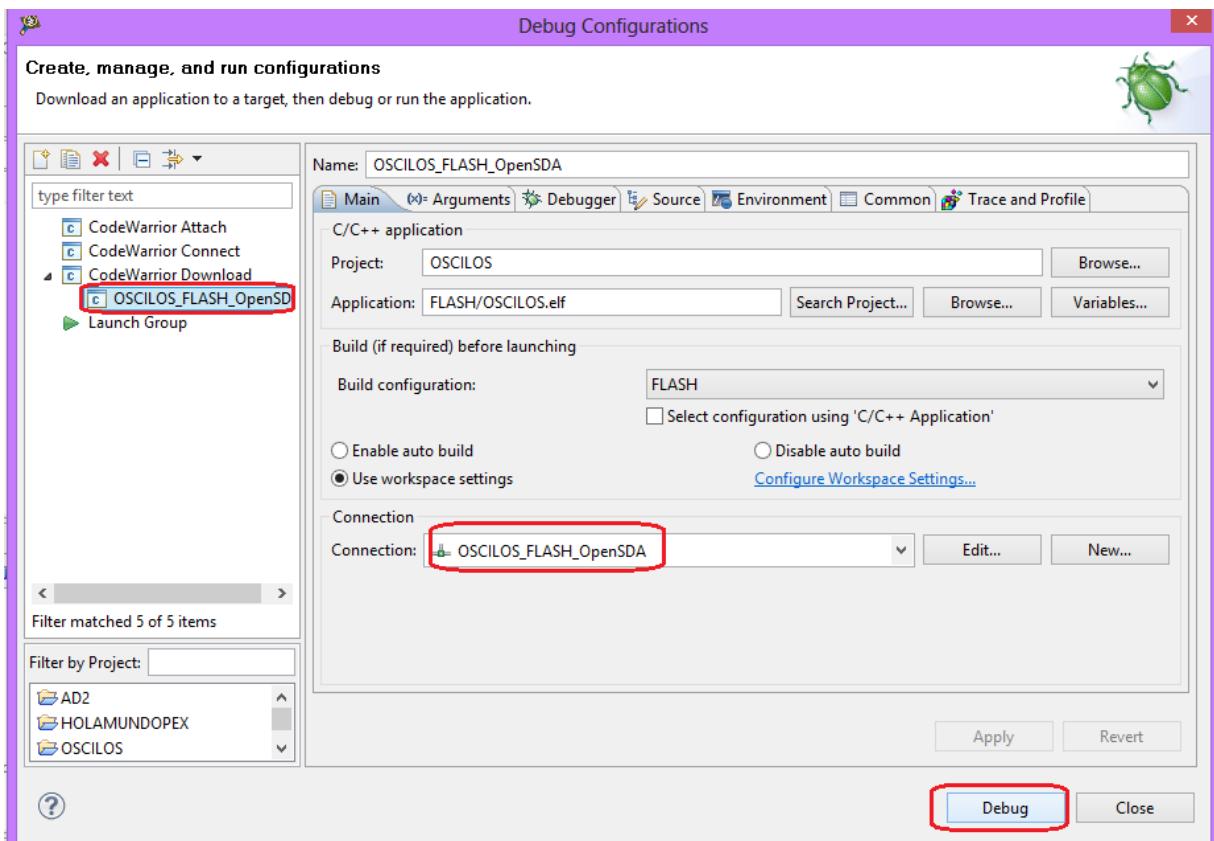


Fig. 97: Grabado proyecto CodeWarrior en el microcontrolador.

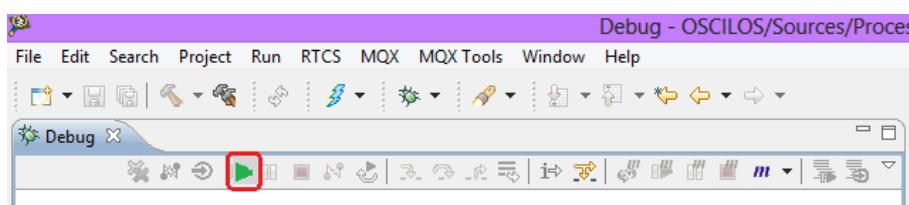


Fig. 98: Ejecución del programa en el microcontrolador.

5.2 Desarrollo del software, para la aplicación a PC

Se creó el diagrama a bloques del funcionamiento de la aplicación a PC (Fig. 99), es decir, de la interfaz gráfica de usuario (IGU).

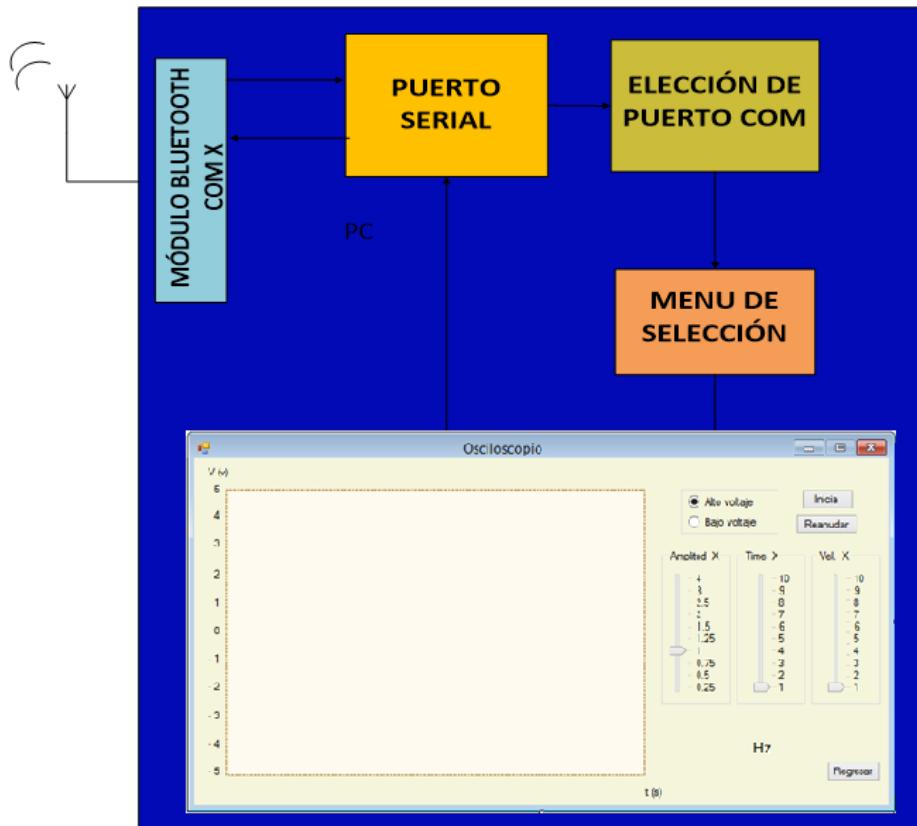


Fig. 99: Diagrama a bloques del software del proyecto.

5.2.1 Elección del entorno de programación

Se decidió trabajar con un entorno visual, ya que el objetivo principal del proyecto, es la graficación de las magnitudes de los distintos sensores, y de señales externas, provenientes de circuitos digitales o analógicos, es decir, la creación de un osciloscopio virtual, lo cual sería imposible lograr con un compilador antiguo.

Se eligió el entorno de programación Visual C#, debido a las herramientas gráficas que posee, la compatibilidad con el sistema operativo (SO) Windows 8, que es el que actualmente va a la alza, además de ser el SO más usado en el todo el mundo. Por otra parte, con Visual C#, se tiene la posibilidad de realizar comunicación con el puerto serial y por ende con el microcontrolador.

5.2.1.1 Ambiente de la plataforma de desarrollo Visual C#

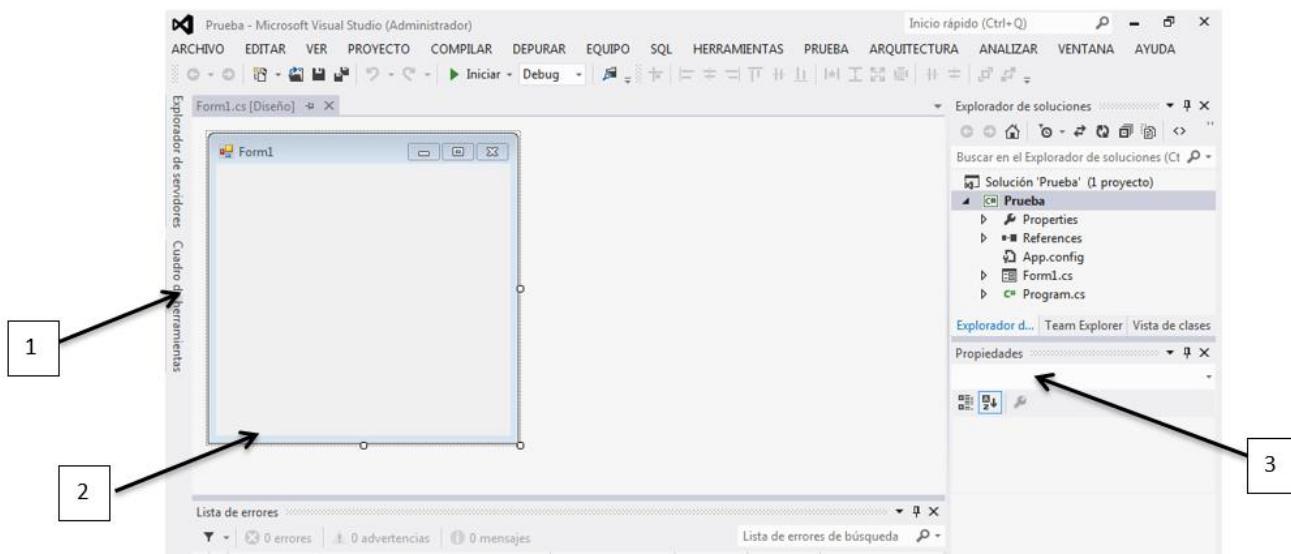


Fig. 100: Ejecución del programa en el microcontrolador.

En la plataforma de desarrollo de Visual Studio, dentro de la sección Visual C#, existen diversos elementos importantes, los cuales se tomaron en cuenta para el momento de crear la aplicación, éstos son los marcados en la Fig. 100 y su explicación es la siguiente:

1. Cuadro de Herramientas: Éste contiene los componentes que es posible utilizar para realizar la aplicación visual, como es el caso de: etiquetas, cajas de texto, botones, contenedores de imágenes, etc.
2. Formulario: Éste es el centro principal de cualquier aplicación a desarrollar, en él, se agregan los diferentes tipos de componentes para la creación de la interfaz gráfica (IGU) a realizar, y es con la que interactúa el usuario.
3. Propiedades: En esta sección se configuran las características de cada uno de los elementos que se van a utilizar en el formulario, incluyendo al mismo.

5.2.2 Elección de los componentes utilizados, para el diseño de la aplicación a PC

Para la elección de los componentes utilizados, para la creación de la aplicación, primeramente se planteó un objetivo de la IGU del proyecto y su solución.

Objetivo: Desarrollar una interfaz gráfica, que permita elegir el puerto COM virtual a utilizar para la comunicación con el microcontrolador, que al mismo tiempo se enlace con una interfaz gráfica principal, que permita al usuario elegir el tipo de magnitud que desee analizar, y a su vez enlazarlo con una ventana de graficación de la magnitud seleccionada.

Solución: La aplicación contará con tres ventanas, la primera de selección de puerto, la segunda de selección de magnitud a medir y la tercera de graficación de la magnitud seleccionada.

5.2.3 Diseño de la aplicación

5.2.3.1 Ventana de selección de puerto

Para la creación de la primera ventana, se realizó el algoritmo, de acuerdo al funcionamiento deseado, y en base a éste, se siguieron los pasos mostrados a continuación, en la plataforma de desarrollo de Visual C#:

- Se modificaron algunas propiedades del formulario:
 - Text: Puerto Serial → Nombre que tendrá la ventana para el usuario.
 - MaximizeBox: False → Para que el botón de maximizar se inactive.
 - MinimizeBox: False → Para que el botón de minimizar se inactive.
 - AutoSize: GrowAndShrink → Para evitar que el usuario modifique las dimensiones de la ventana.
- Se agregó un componente “Label” y se modificó la siguiente propiedad:
 - Text: Elige el puerto → Para así indicar al usuario lo que debe hacer.
- Se agregó un componente de “Combo Box” al formulario, el cual contiene las opciones de los puertos COM posibles a seleccionar.
- Se agregó un componente “Button” y en propiedades, se modificó:
 - Text: Continuar → Texto que contendrá el botón.
- Se agregó un componente “SerialPort”, no es necesario configurar ninguna propiedad, pues este elemento es únicamente para verificar que el COM seleccionado por el usuario, efectivamente existe
- Se ajustó el tamaño del formulario de acuerdo con los elementos antes agregados.

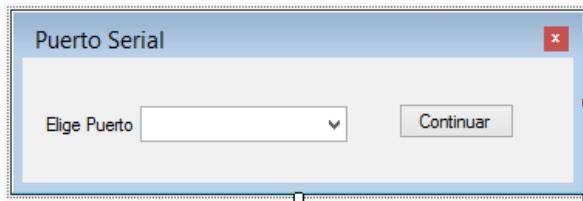


Fig. 101: Ventana para la selección de puerto COM.

Una vez diseñada la ventana, se generó el evento correspondiente al botón “Continuar” y se incluyó el uso de la librerías “System.IO.Ports” y “System.IO”, para mostrar al usuario el nombre de los puertos conectados a su computadora, y para guardar el número de puerto seleccionado por el usuario, respectivamente, lo anterior debido a que, al presionar el botón, esta ventana de selección de COM, se cerrará y se abrirá la siguiente ventana (de selección de magnitud a analizar). Entonces, se siguieron los pasos mostrados a continuación, para la programación de la ventana uno, de la aplicación ([Anexo C.3](#)):

- Doble click en el botón “Continuar”.
- En la pestaña que se abrió, se agregó el uso de la librería System.IO, justo en la primera línea.
- En el método de inicialización del formulario, se agregaron al componente “ComboBox” las opciones de puerto COM posibles de seleccionar, para que cuando el usuario despliegue el contenido, pueda dar click en alguna de ellas.
- En el método del botón, se le dio en la propiedad del nombre del SerialPort, el puerto seleccionado por el usuario antes de dar click en “Continuar”.
- Se programó que se probara si el puerto serial con el nombre seleccionado se podía abrir, de no ser así, mandar un mensaje al usuario (“El puerto no se puede abrir”), de poderse abrir, entonces la aplicación guardará el nombre del puerto en un archivo de datos.
- Se cierra esta ventana y se abre la ventana dos, de selección de magnitud a analizar.

obteniendo así, el diagrama de clases (Fig. 102), para esta ventana.

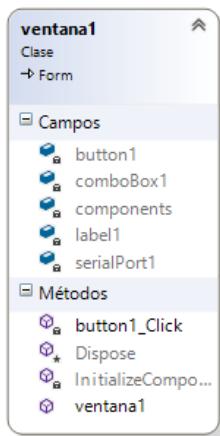


Fig. 102: Diagrama de clases para la ventana de selección de puerto COM.

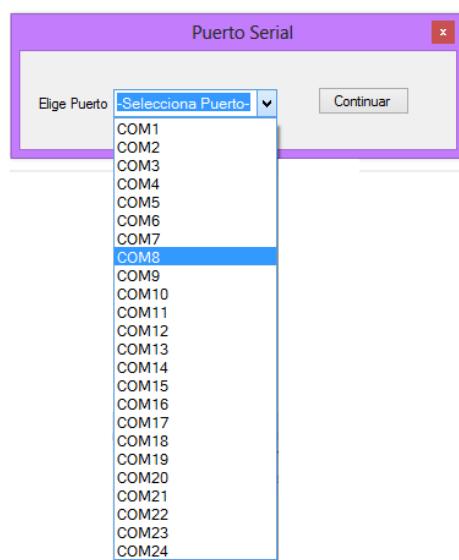


Fig. 103: Ejecución de la primera parte de la aplicación.

5.2.3.2 Ventana de selección de magnitud a medir

Para la creación de la aplicación selectora de magnitud a analizar (temperatura, presión, humedad, densidad de flujo magnético y voltaje), se realizó el diseño de la ventana, correspondiente al funcionamiento deseado, y en base a éste, se llevaron a cabo los pasos siguientes:

- Se generó un nuevo proyecto en Visual C#.
- Se modificaron algunas propiedades del formulario:
 - Text: Magnitudes a Analizar → Nombre que tendrá la ventana para el usuario.
 - AutoSizeMode: GrowAndShrink → Para que el usuario no pueda modificar las dimensiones de la ventana.
 - BackColor: LightSteelBlue → Para cambiar el color de fondo del formulario.
 - MaximizeBox: False → Inactivación del botón de maximizar.
- Se agregó un elemento ToolTip, para permitir que se generen leyendas cortas cuando el usuario pase el cursor por ciertas áreas.
- Se agregaron seis componentes PictureBox, y en cada uno se modificaron las siguientes propiedades:
 - BackgroundImage: _____ → Se dio una ruta correspondiente a una imagen referente a cada una de las magnitudes a medir.
 - BackgroundImageLayout: Stretch → Para que la imagen cargada se ajustará al tamaño propuesto del componente PictureBox.
 - Cursor: Hand → Para que cada vez que el usuario pase el cursor sobre la imagen, aparezca una manita, lo cual señala que si se da click, se abrirá otra ventana, es decir, al dar click en alguna imagen, el usuario estaría seleccionando la magnitud a medir.
 - ToolTip: _____ → Leyenda que se desea aparezca, cada que el usuario pase el cursor sobre la imagen (nombre de la magnitud representada).
- Se adicionaron seis elementos Label, para colocar arriba de cada imagen, el nombre de la magnitud que representan y que al usuario le quede más claro lo que se quiere dar a entender con la imagen.
 - Text: _____ → Nombre de la magnitud representada con cada imagen.

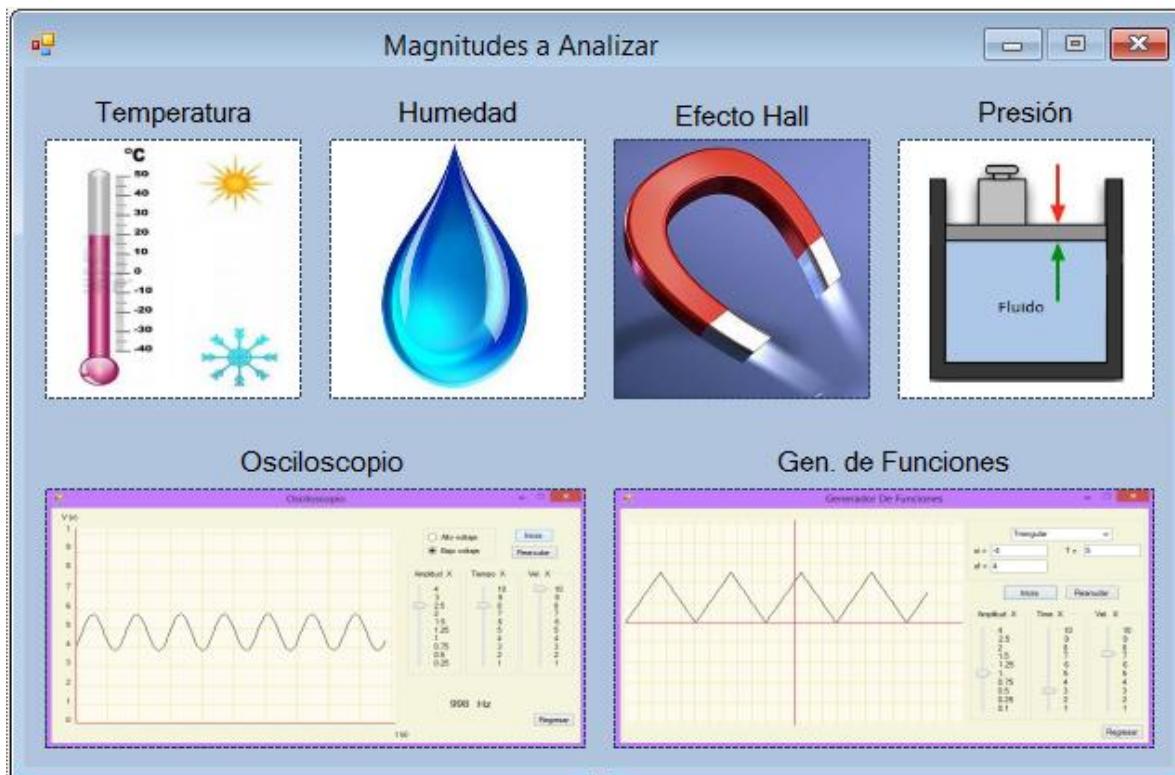


Fig. 104: Diseño de la ventana de selección de magnitud a analizar.

Una vez diseñada la ventana, se creó el código fuente, para el funcionamiento de esta sección de la aplicación ([Anexo C.4](#)), generando el evento correspondiente a cada PictureBox, dando lugar a que se abra una nueva y última ventana, la cual graficará las variaciones de la magnitud seleccionada. Para ello, se siguieron los pasos que se muestran a continuación:

- Doble click en todos y cada uno de los PictureBox agregados.
- En el método correspondiente a cada PictureBox, se abre la ventana de graficación de la magnitud seleccionada y se cierra esta ventana (de selección de magnitud).

obteniendo así, el diagrama de clases correspondiente a esta ventana.

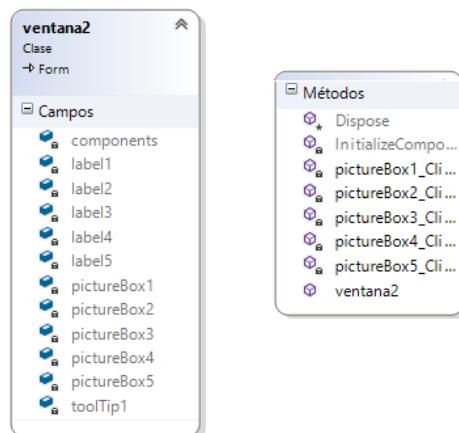


Fig. 105: Diagrama de clases para la ventana de selección de magnitud a graficar.

5.2.3.3 Ventana de graficación de magnitud seleccionada

5.2.3.3.1 Para los sensores

Para la creación de la ventana graficadora de la magnitud seleccionada, y si ésta depende del uso de alguno de los sensores de temperatura, presión, humedad o efecto Hall, se realizaron los siguientes pasos, para el diseño de la aplicación:

- Se creó un proyecto nuevo en Visual C# y en el formulario, se modificaron las siguientes propiedades:
 - Text: _____ → Nombre que tendrá la ventana para el usuario (nombre de la magnitud que se graficará).
 - AutoSizeMode: GrowAndShrink → Para que el usuario no pueda modificar las dimensiones de la ventana.
 - BackColor: Beige → Para cambiar el color de fondo del formulario.
 - MaximizeBox: False → Inactivación del botón de maximizar.
- Se agregó un componente PictureBox con las propiedades:
 - BackColor: FloralWhite → Para cambiar el color de fondo del control.
- Se agregaron diversos componentes Label para indicar la escala de la gráfica, el periodo a seleccionar para la gráfica, y algunos datos extras que el usuario debe saber.
- Se agregó un componente TextBox, para cambiar el valor de las muestras a graficar en el PictureBox.
- Se agregó un componente ListBox, para registrar las mediciones graficadas.
- Se adicionaron dos elementos TrackBar, para variar la amplitud en graficación de la señal y las muestras por PictureBox.
- Finalmente se agregaron tres componentes Button, el primero para iniciar y detener la graficación, el segundo para reanudar la graficación, y el tercero para regresar a la aplicación selectora de sensores.

Todos los pasos anteriores, fueron realizados para la creación de cada ventana según el sensor seleccionado, es decir, se tiene una ventana de graficación de temperatura, una para presión, una para humedad y una para efecto Hall.

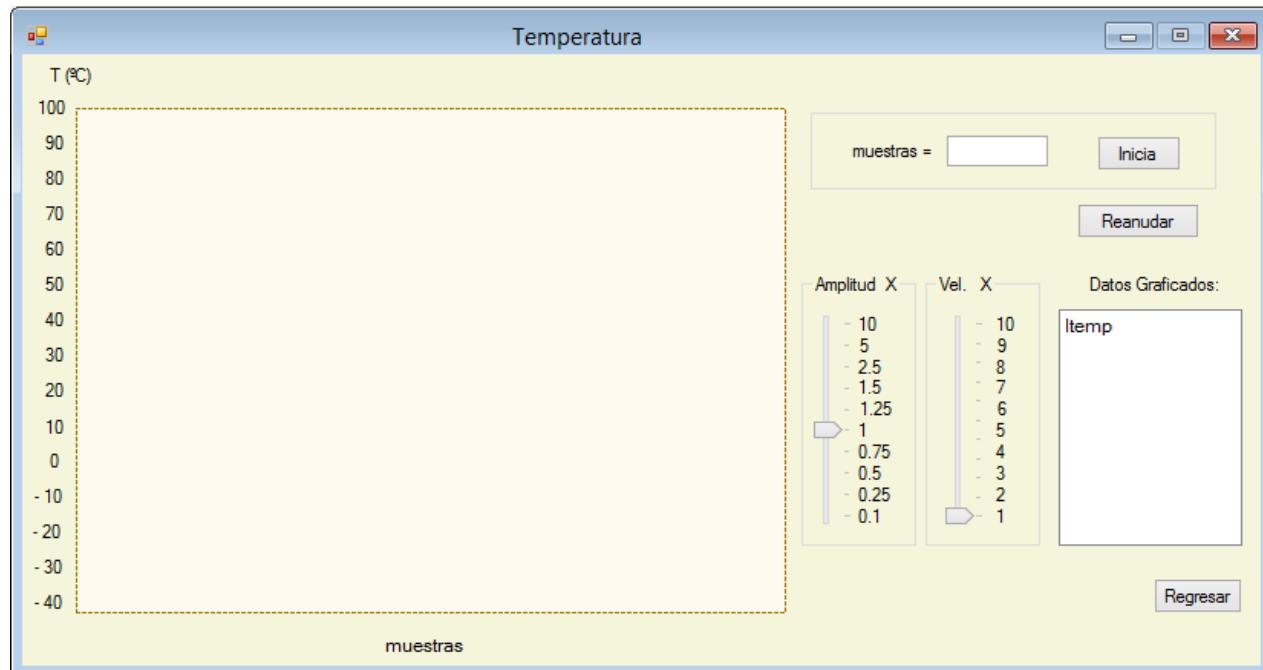


Fig. 106: Diseño de la ventana de graficación de temperatura.

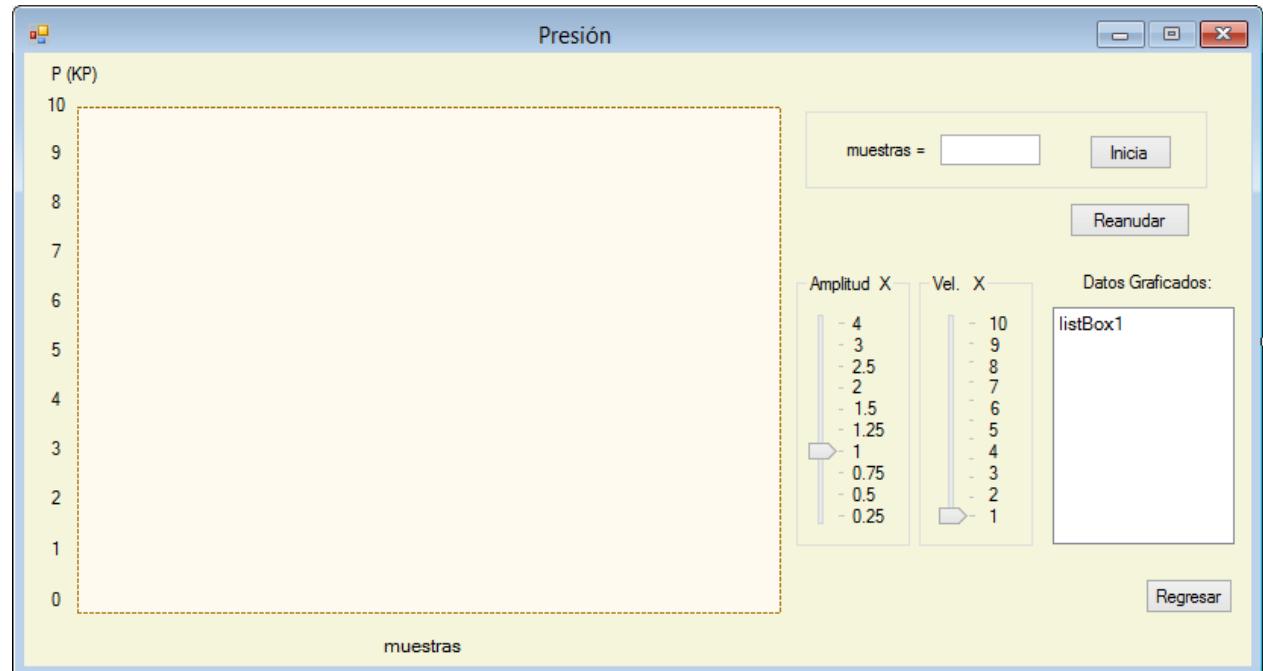


Fig. 107: Diseño de la ventana de graficación de presión.

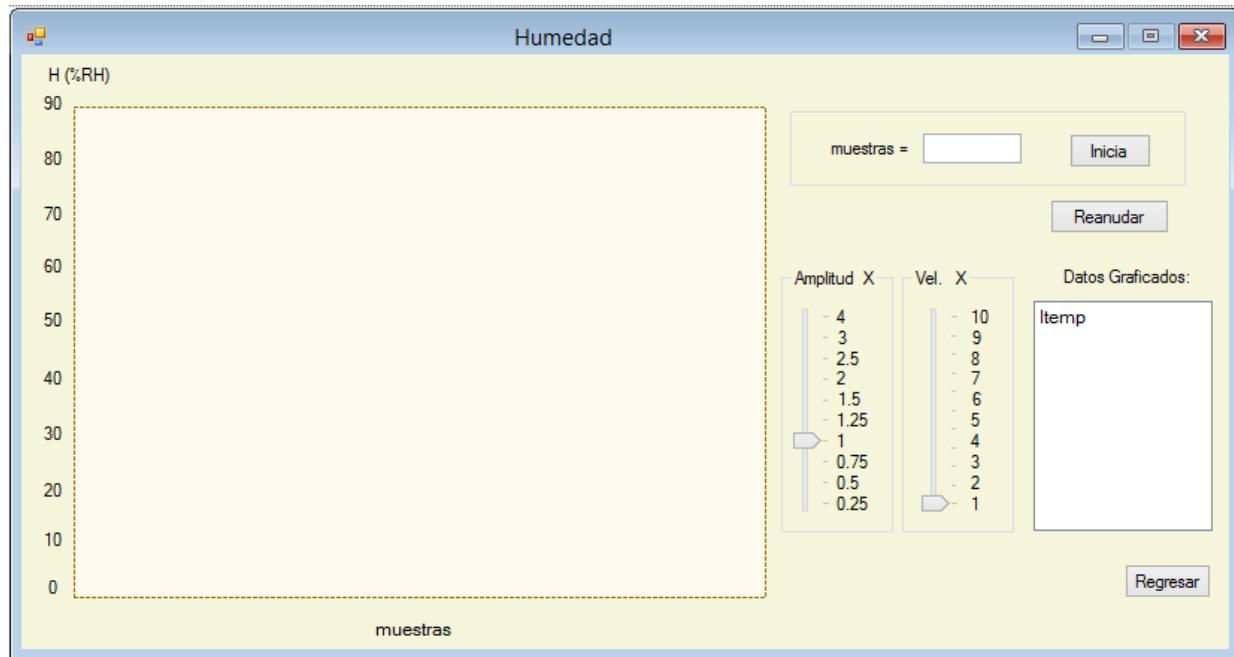


Fig. 108: Diseño de la ventana de graficación de humedad.

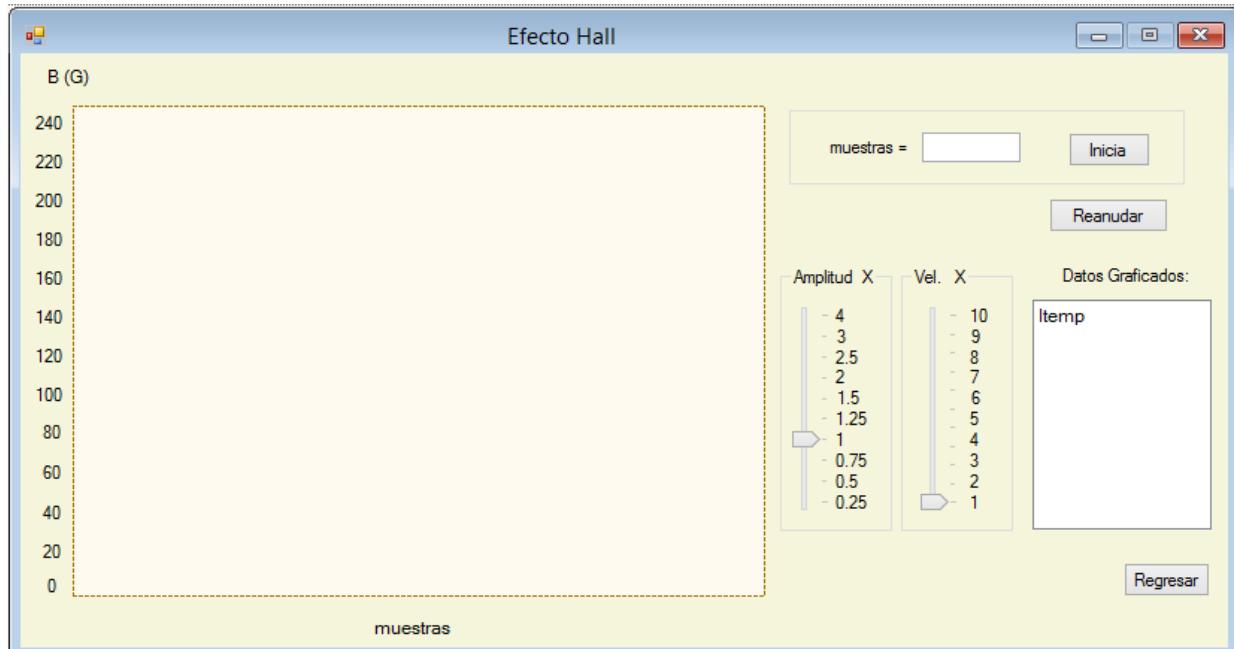


Fig. 109: Diseño de la ventana de graficación de efecto Hall.

Una vez diseñada la ventana, para la creación del código fuente, se incluyó el uso de la librería “System.IO”, para saber el número de puerto seleccionado por el usuario, guardado en la primera ventana (de selección de puerto COM) y con ello, poder hacer comunicación entre la PC y el microcontrolador, lo cual es necesario para la obtención de los datos arrojados por el sensor seleccionado

Antes de comenzar con la programación de esta parte de la aplicación, fue necesario, hacer una correspondencia entre las coordenadas de un plano cartesiano y las coordenadas de pantalla, y también, el ajuste o adecuación del dato proveniente del ADC al valor real de la magnitud medida.

Las diferencias y similitudes encontradas, entre el mapa de coordenadas físicas (plano cartesiano) y el mapa de coordenadas en pantalla, se muestran en la tabla 7.

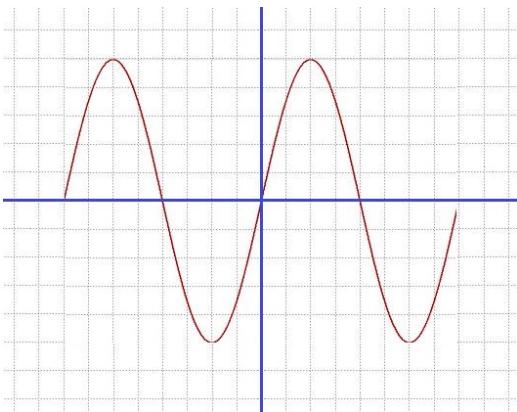


Fig. 110: Plano cartesiano.

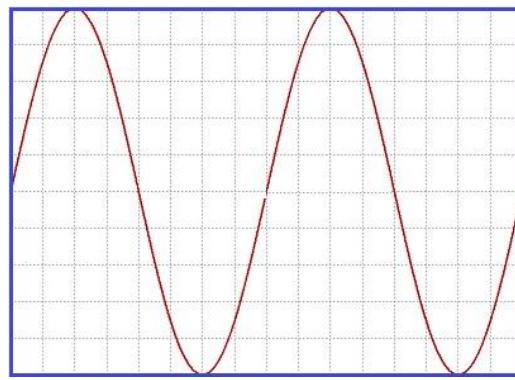


Fig. 111: Plano en pantalla.

Coordenadas físicas	Coordenadas en pantalla
Cuatro cuadrantes.	Un cuadrante.
Ejes x, y (ordenadas y abscisas) infinitos.	Ejes c, f (columnas y filas) finitos.
Origen en el punto de intersección de los cuatro cuadrantes.	Origen en la parte superior izquierda.
Gráfica formada por la unión de puntos reales.	Gráfica formada por píxeles (puntos de valor entero).
Infinidad de puntos.	Píxeles finitos.
El valor de las coordenadas, se encuentra en el conjunto de los números reales	El valor de las coordenadas, se encuentra en el conjunto de los números enteros computacionales (sin signo)

Tabla 7: Cuadro comparativo entre coordenadas físicas y coordenadas de pantalla.

Para crear un gráfico en el plano cartesiano, debido a que los puntos son infinitos, se hace una acotación de éstos, así, en el eje de las abscisas, se tiene un punto inicial (x_i) y otro final (x_f), para la graficación, con ello, se determina el número de puntos a colocar en el intervalo dado, entonces, cada

punto con respecto al anterior y al siguiente, posee la misma distancia (h), en conclusión, los puntos se determinan por:

$$\begin{aligned}x_1 &= x_i \\x_2 &= x_i + h \\x_3 &= x_i + 2h \\x_4 &= x_i + 3h \\\vdots &\end{aligned}$$

Y en cuanto al eje de las ordenadas, el punto inicial (y_i) y el final (y_f), están determinados por la función que se esté graficando $\rightarrow y = f(x)$

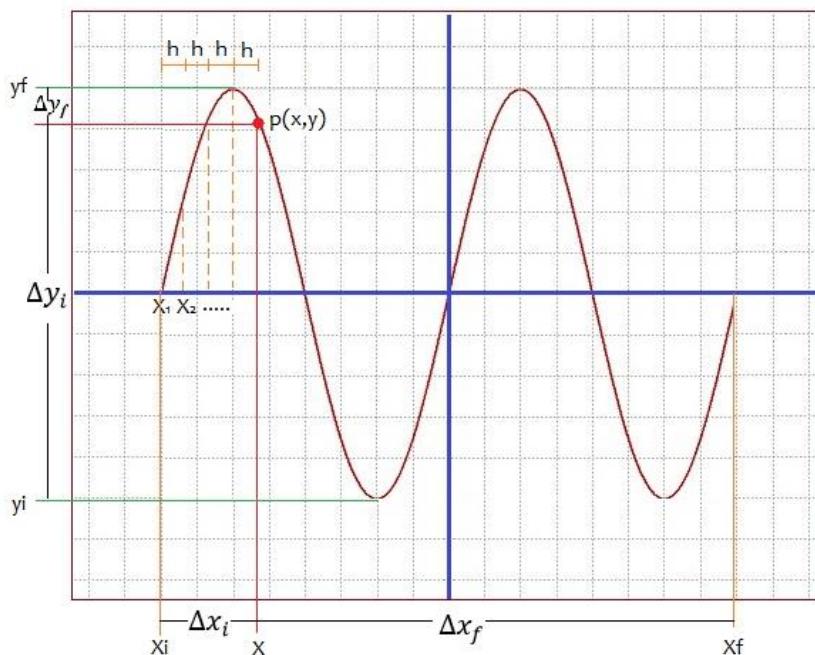


Fig. 112: Forma de graficación en el plano cartesiano.

Para encontrar un punto en pantalla, el razonamiento es similar, se tiene un número de columnas y filas finitos, entonces, un píxel, posee un número de fila y columna específicos $\rightarrow px(c, f)$.

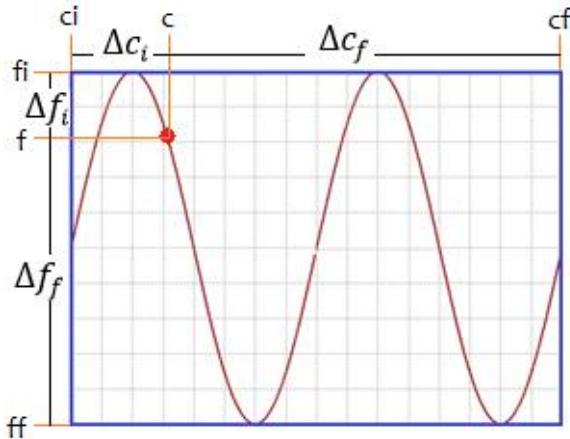


Fig. 113: Forma de graficación en el plano en pantalla.

Entonces, a cada diferencia (Δ), en el plano cartesiano, corresponde una diferencia en el plano en pantalla, es decir:

- A un Δx_i , corresponde un Δc_i .
- A un Δx_f , corresponde un Δc_f .
- A un Δy_i , corresponde un Δf_i .
- A un Δy_f , corresponde un Δf_f .

Por lo tanto, la relación entre ambos planos, en el eje horizontal, está dada por la siguiente ecuación:

$$\frac{\Delta x_i}{\Delta c_i} = \frac{\Delta x_f}{\Delta c_f} \rightarrow \frac{x - x_i}{c - c_i} = \frac{\Delta x_f - x}{c_f - c}$$

Despejando a “c”, se obtuvo la función, para el valor de la columna en el plano en pantalla, correspondiente al valor de la abscisa de un punto en el plano cartesiano.

$$c = c_f * \frac{x - x_i}{x_f - x_i} + c_i * \frac{x_f - x}{x_f - x_i}$$

Por otra parte, la relación entre ambos planos, en el eje vertical, tomando en cuenta que, el inicio de este eje en el plano cartesiano es en la parte inferior, y en el plano en pantalla, es en la parte superior, la ecuación es la siguiente:

$$\frac{\Delta y_i}{\Delta f_f} = \frac{\Delta y_f}{\Delta f_i} \rightarrow \frac{x - x_i}{c - c_i} = \frac{\Delta x_f - x}{c_f - c}$$

Despejando a “f”, se obtuvo la función, para el valor de la fila en el plano en pantalla, correspondiente al valor de la ordenada de un punto en el plano cartesiano.

$$f = f_f * \frac{y_f - y}{y_f - y_i} + f_i * \frac{x - y_i}{y_f - y_i}$$

Una vez realizado el análisis anterior, se procedió a obtener las funciones de ajuste al valor correspondiente de la magnitud medida. Debido a que el dato recibido por el puerto serial de la PC (dato a graficar), es un valor dado por la conversión ADC del microcontrolador, es decir, es un número entre 0 y 65535, éste se necesita procesar, para obtener el valor de la magnitud. Por lo que las ecuaciones de cada magnitud son las siguientes:

- Temperatura

$$T = SalDig_{ADC} * Res_{ADC} * FT_{LM61}$$

$$T = SalDig_{ADC} * \frac{3.3V}{65535} * \left(\frac{1^{\circ}C}{10mV} * \frac{1000mV}{1V} - 600mV * \frac{1V}{1000mV} \right)$$

$$T = \frac{3300 * SalDig_{ADC}}{(10)(65535)} - 60 [^{\circ}C]$$

- Presión

$$P = SalDig_{ADC} * Res_{ADC} * FT_{MPX10DP} = SalDig_{ADC} * \frac{3.3 V}{65535} * \frac{1 KPa}{3.5 mV} * \frac{1000 mV}{1 V}$$

$$P = \frac{3300 * SalDig_{ADC}}{(3.5)(65535)} [KPa]$$

- Humedad

$$HR = SalDig_{ADC} * Res_{ADC} * FT_{HMZ-433A1} = SalDig_{ADC} * \frac{3.3 V}{65535} * \frac{20 \%RH}{0.66 V}$$

$$HR = \frac{3.3 * 20 * (SalDig_{ADC})}{65535 * 0.66} [\%RH]$$

- Efecto Hall

$$B = SalDig_{ADC} * Res_{ADC} * FT_{UGN3503} = SalDig_{ADC} * \frac{3.3 V}{65535} * \frac{1 G}{1.3 mV} * \frac{1000 mV}{1 V}$$

$$B = \frac{3300 * SalDig_{ADC}}{(1.3)(65535)} [Gauss]$$

Como último paso, una vez diseñada la ventana, y teniendo en cuenta la equivalencia de una coordenada en plano cartesiano con una coordenada en pantalla, se creó el código de funcionamiento para la ventana de graficación de magnitud ([Anexo C.5](#)), generando el evento de cada control Button. Para ello, se siguieron los pasos que se muestran a continuación:

- Doble click en todos y cada uno de los Button agregados.
- En el método correspondiente a cada botón, realiza lo esperado (graficar, parar la gráfica o volver a la ventana de selección de magnitud)
- Doble click en cada elemento TrackBar
- En el método correspondiente a cada barra, se aumenta o disminuye la amplitud, o bien, la velocidad de graficación.
- Doble click en el control timer, para generar su evento
- En el evento del timer (para cada vez que acabe una cuenta), se envía el carácter asociado con la magnitud que se seleccionó para graficar, se procesa el dato obtenido por el ADC del microcontrolador y se grafica en el Picture Box.

De esta forma, se obtuvo el diagrama de clases, correspondiente a todas las ventanas de graficación de magnitud proveniente de los sensores, puesto que las propiedades y métodos, son los mismos para todos, y sólo se diferencian en la función de adecuación.

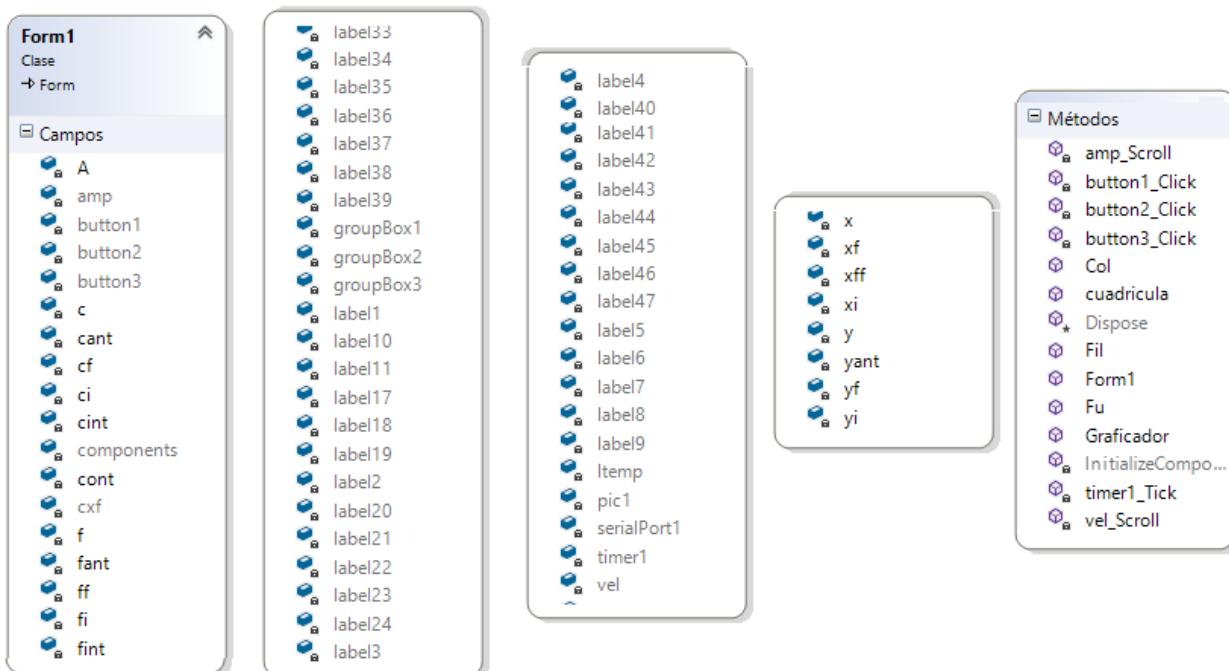


Fig. 114: Diagrama de clases, para la venta de graficación de magnitud proveniente de un sensor.

5.2.3.3.2 Para el osciloscopio

Se realizó el diseño de la ventana, correspondiente al funcionamiento deseado, y en base a esto, se llevaron a cabo los siguientes pasos:

- En un proyecto nuevo en Visual C#, en el formulario, se modificaron las siguientes propiedades:
 - Text: Osciloscopio → Nombre que tendrá la ventana para el usuario.
 - AutoSizeMode: GrowAndShrink → Para que el usuario no pueda modificar las dimensiones de la ventana.
 - BackColor: Beige → Para cambiar el color de fondo del formulario.
 - MaximizeBox: False → Inactivación del botón de maximizar.
- Se agregó un componente PictureBox con las propiedades:
 - BackColor: FloralWhite → Para cambiar el color de fondo del control.
- Se incluyó un elemento GrupBox y dentro de éste, dos controles RadioButton, en donde el usuario seleccionará el rango de voltaje de la señal a analizar, para saber si medir en el módulo de alto voltaje o en el de bajo voltaje.
- Se agregaron diversos componentes Label para indicar la escala de la gráfica, el periodo, y algunos datos extras que el usuario debe saber, como la frecuencia.
- Se adicionaron tres elementos TrackBar, para variar la amplitud en graficación de la señal, las muestras a graficar por PictureBox, y la velocidad de graficación.
- Finalmente se agregaron tres componentes Button, el primero para iniciar y detener la graficación, el segundo para reanudar la graficación, y el tercero para regresar a la aplicación selectora de sensores.

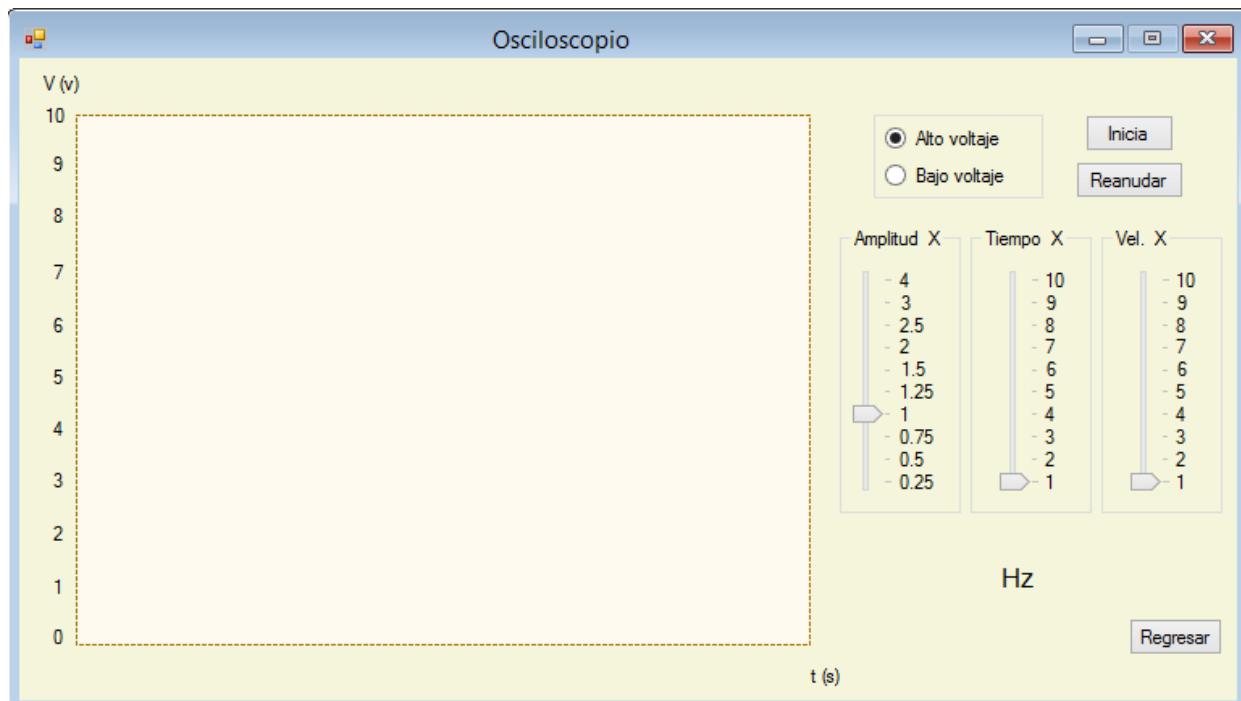


Fig. 115: Diseño de la ventana del osciloscopio.

Para la creación del código fuente, para esta ventana, se incluyó el uso de la librería “System.IO”, para saber el número de puerto seleccionado por el usuario, guardado en la primera ventana (de selección de puerto COM), se tomaron en cuenta las funciones de relación entre el plano cartesiano y el plano en pantalla, y la función de adecuación del dato recibido al voltaje medido, según la selección entre alto y bajo voltaje.

Una vez hecho lo anterior, se creó el código de funcionamiento para el osciloscopio ([Anexo C.6](#)), generando el evento correspondiente a cada botón. Para ello, se siguieron los pasos que se muestran a continuación:

- Doble click en todos los controles Button agregados.
- En el método correspondiente a cada botón, realiza lo esperado (graficar, parar la gráfica, reanudar la graficación, o volver a la ventana de selección de magnitud).
- Doble click en cada elemento TrackBar.
- En el método correspondiente a cada barra, se aumenta o disminuye la amplitud, la velocidad, o bien, el número de muestras a graficar por cada división del PictureBox.
- Doble click en el control timer, para generar su evento.
- En el evento del timer, se envía el carácter asociado con la selección de alto o bajo voltaje, se procesa el dato obtenido por el ADC del microcontrolador y se grafica en el Picture Box.

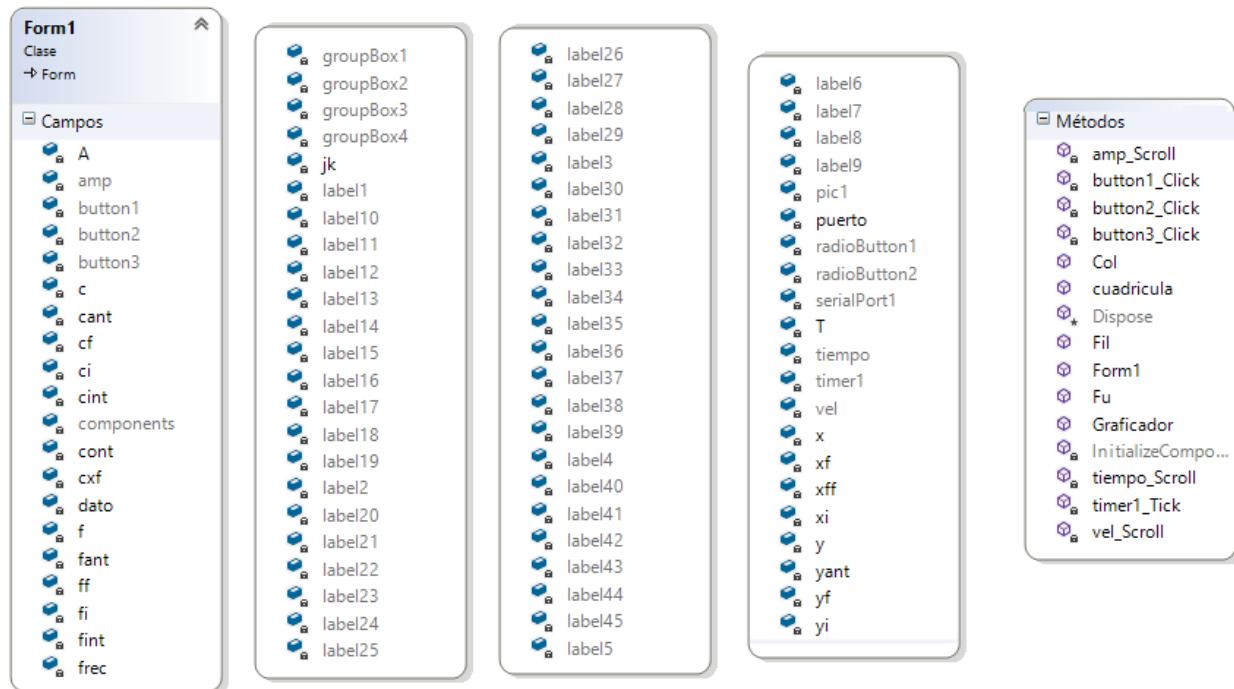


Fig. 116: Diagrama de clases, para la venta de graficación, del osciloscopio.

5.2.3.4 Ventana del generador de funciones

Se realizó el diseño de la ventana, correspondiente al funcionamiento deseado, llevando a cabo los siguientes pasos:

- En un proyecto nuevo en Visual C#, en el formulario, se modificaron las siguientes propiedades:
 - Text: Generador De Funciones → Nombre que tendrá la ventana para el usuario.
 - AutoSizeMode: GrowAndShrink → Para que el usuario no pueda modificar las dimensiones de la ventana.
 - BackColor: Beige → Para cambiar el color de fondo del formulario.
 - MaximizeBox: False → Inactivación del botón de maximizar.
- Se agregó un componente PictureBox con las propiedades:
 - BackColor: FloralWhite → Para cambiar el color de fondo del control.
- Se incluyó un elemento GrupBox y dentro de éste, una lista desplegable para indicarle al usuario las diversas funciones posibles a generar, de igual forma, se agregaron varios elementos TextBox, para que le usuario introduzca algunas características que deseé de la función (periodo, ciclo útil, etc.).
- Se agregaron diversos componentes Label para indicar la escala de la gráfica, el periodo, y algunos datos extras que el usuario debe saber, como la frecuencia.
- Se adicionaron tres elementos TrackBar, para variar la amplitud en graficación de la señal, las muestras a graficar por PictureBox, y la velocidad de graficación.
- Finalmente se agregaron tres componentes Button, el primero para iniciar y detener la graficación, el segundo para reanudar la graficación, y el tercero para regresar a la aplicación selectora de sensores.

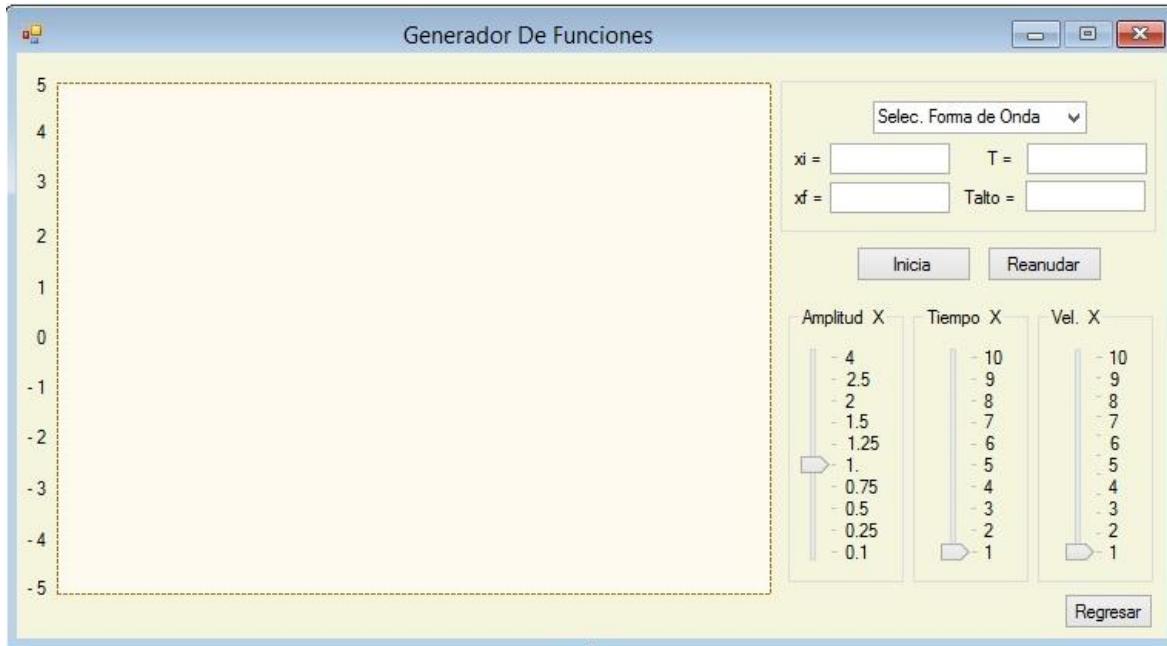


Fig. 117: Diseño de la ventana del generador de funciones.

Una vez diseñada la ventana del generador de funciones, se creó el código de funcionamiento para el osciloscopio ([Anexo C.7](#)), generando el evento correspondiente a cada botón. Para ello, se siguieron los pasos que se muestran a continuación:

- Doble click en todos los controles Button agregados.
- En el método correspondiente a cada botón, realiza lo esperado (graficar, parar la gráfica, reanudar la graficación, o volver a la ventana de selección de magnitud).
- Doble click en cada elemento TrackBar.
- En el método correspondiente a cada barra, se aumenta o disminuye la amplitud, la velocidad, o bien, el número de muestras a graficar por cada división del PictureBox.
- Doble click en el control timer, para generar su evento.
- En el evento del timer, se van graficando muestras de la función seleccionada por el usuario, utilizando las ecuaciones matemáticas, que se muestran a continuación:

➤ Función coseno: $y(x) = \cos(x)$

➤ Función seno: $y(x) = \sin(x)$

➤ Pulso: $y(x + T) = \begin{cases} 1 & 0 < x < D_c \\ 0 & D_c < x < T \end{cases}$

donde:

D_c : Ciclo útil de un periodo de la función (tiempo activo en un ciclo de la señal).

T : Periodo de un ciclo de la señal.

➤ Función rampa: $y(x + T) = x$

donde:

T : Periodo de un ciclo de la señal

➤ Función triangular: $y(x + T) = \begin{cases} x & 0 < x < \frac{T}{2} \\ -x + T & \frac{T}{2} < x < T \end{cases}$

donde:

T : Periodo de un ciclo de la señal

➤ Función rectangular: $y(x + T) = \begin{cases} 1 & 0 < x < \frac{T}{2} \\ -1 & \frac{T}{2} < x < T \end{cases}$

donde:

T : Periodo de un ciclo de la señal

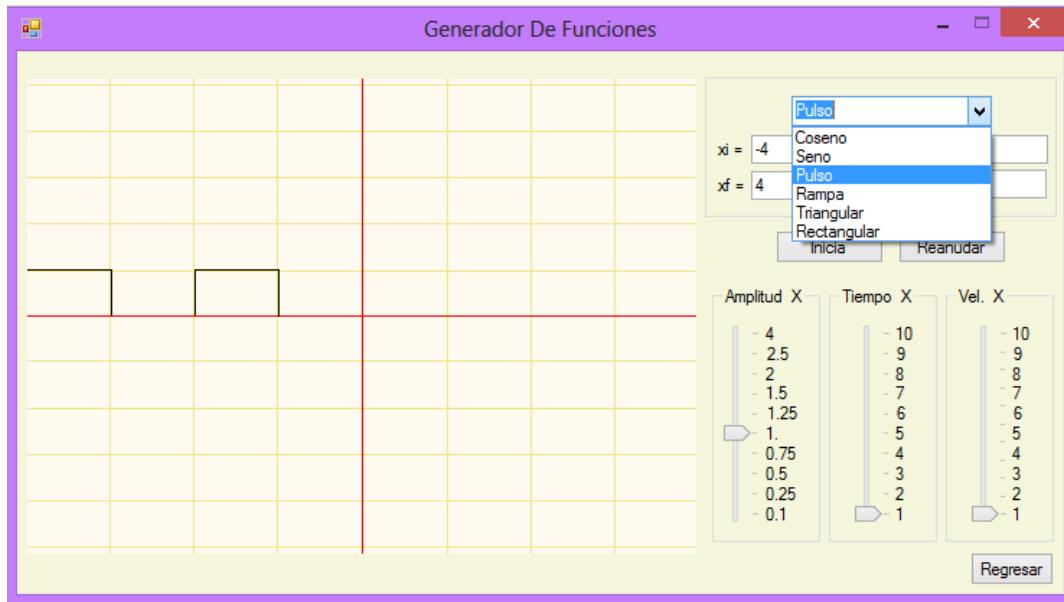


Fig. 118: Funcionamiento de la ventana del generador de funciones.

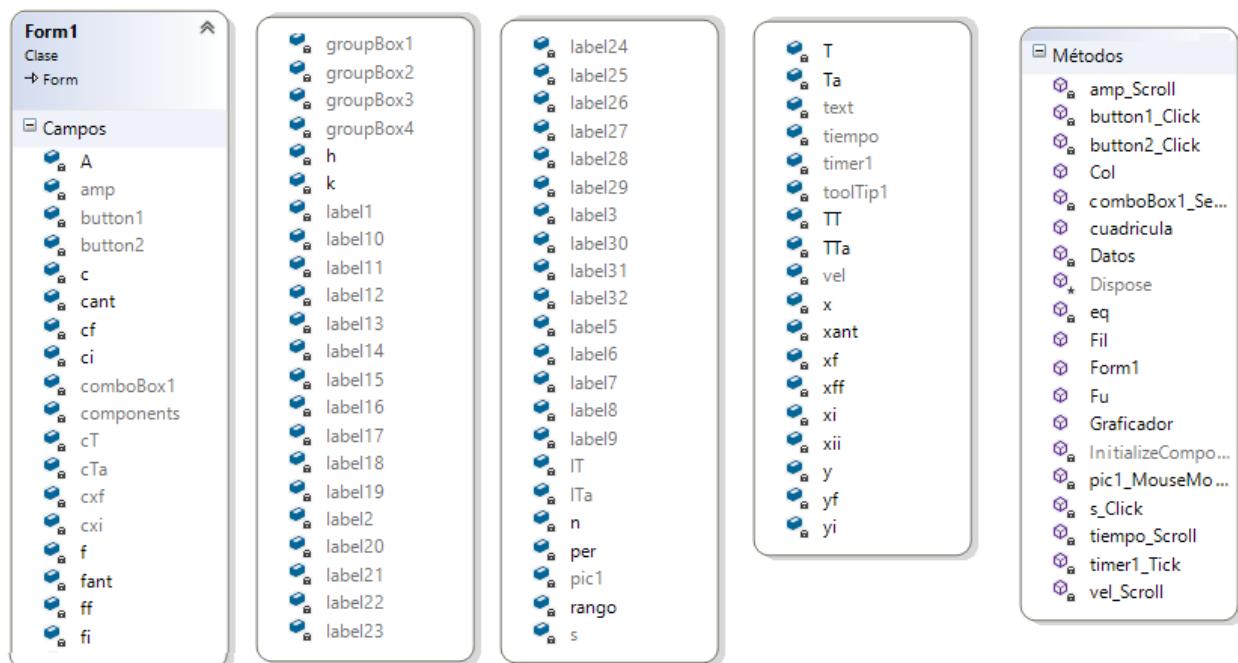


Fig. 119: Diagrama de clases del generador de funciones.

Capítulo 6:

Pruebas de

Funcionamiento

6. Pruebas de funcionamiento

Para comprobar el funcionamiento del prototipo, se realizaron diversas pruebas, utilizando para ello, un generador de funciones y un osciloscopio de banco, con el objetivo de comprobar el ancho de banda soportado por el osciloscopio desarrollado en visual C#, y las diferencias y similitudes entre éste y el osciloscopio de banco. Los resultados se muestran a continuación.

6.1 Para el medidor virtual

Para comprobar las mediciones de los sensores, se tomaron muestras de cada sensor, sin alterar las magnitudes, se midió la salida de voltaje de cada uno con la ayuda de un multímetro, y se realizó la conversión de voltaje a magnitud, posteriormente, se comprobó que los datos graficados, coincidieran con los valores calculados.

6.1.1 Medición de temperatura

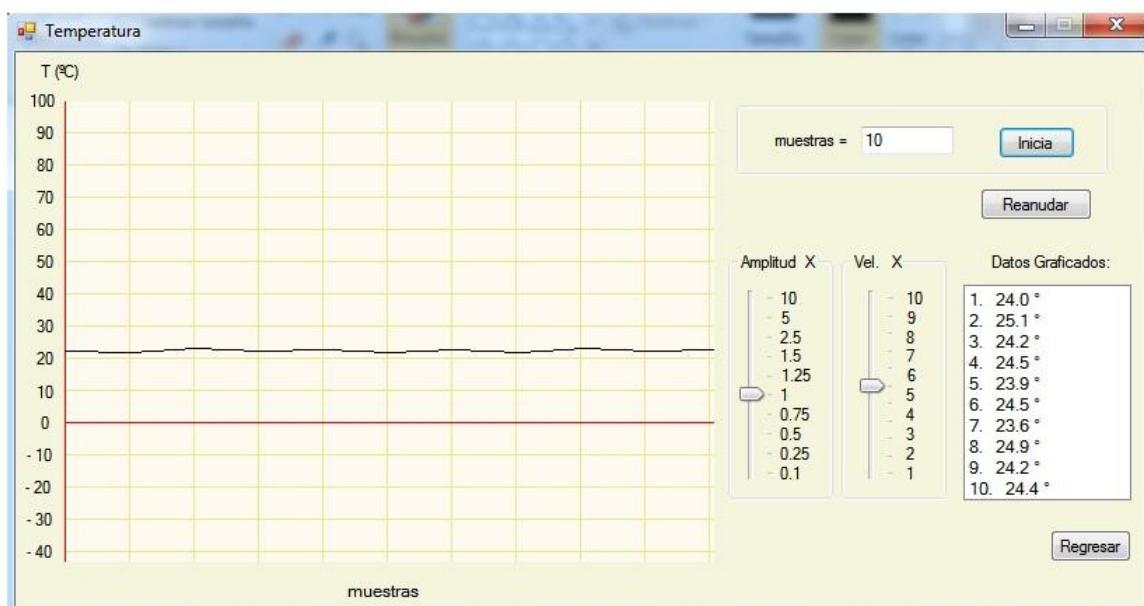


Fig. 120: Señal de 10 muestras del sensor de temperatura.

La ecuación de adecuación del sensor, para obtener el valor en grados, de la temperatura medida, se realiza por medio de la programación, de acuerdo al dato recibido por el microcontrolador (medición del ADC), como ya se había mostrado anteriormente, está dada por:

$$T = \frac{\left(\frac{Dato}{65535} * 3300 - 600 \right)}{10}$$

La función de transferencia del sensor, en función del voltaje de salida es:

$$T = (V_o - 600) * 100$$

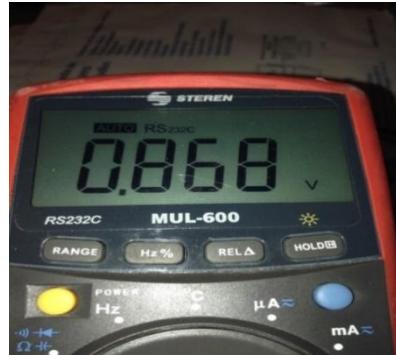


Fig. 121: Lectura de voltaje para el sensor de temperatura.

Sustituyendo el valor medido en la ecuación de la función de transferencia del sensor, se tiene que el valor de la temperatura medida, es:

$$T = (0.868 - 0.6) * 100 = 26.8^\circ$$

Teniendo los valores graficados por la aplicación, y los calculados a partir de la medición hecha por medio del multímetro, se hizo el cálculo del error entre el valor real detectado por el sensor, y el valor graficado más alejado al real.

$$E = \frac{|Val_{real} - Val_{graf}|}{Val_{real}} * 100\%$$

$$E_{Tem} = \frac{|26.8^\circ C - 23.6^\circ C|}{26.8^\circ C} * 100\% = 11.94\%$$

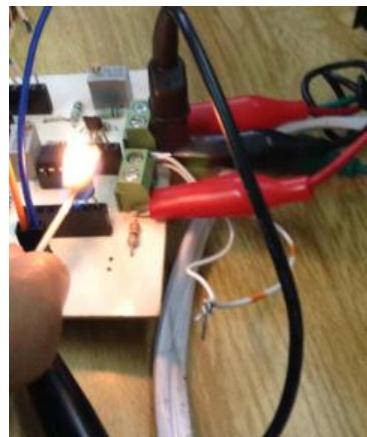


Fig. 122: Calentamiento del sensor de temperatura.

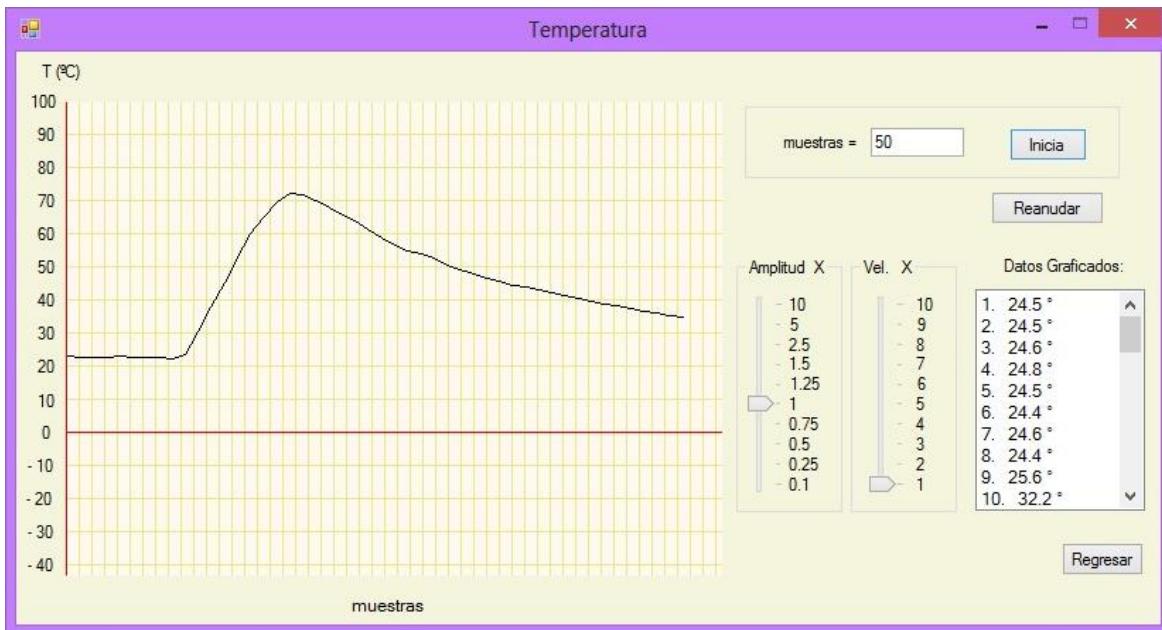


Fig. 123: Respuesta del sensor de temperatura al ser calentado, mostrado en la interfaz gráfica creada.

6.1.2 Medición de humedad

Haciendo uso de la función de transferencia del sensor, y la medida del multímetro para esta prueba, mostrada en la Fig.125, se obtiene:

$$RH = V_o * 30.303$$

$$RH = (2.055) * 30.303 = 62.27 \%RH$$

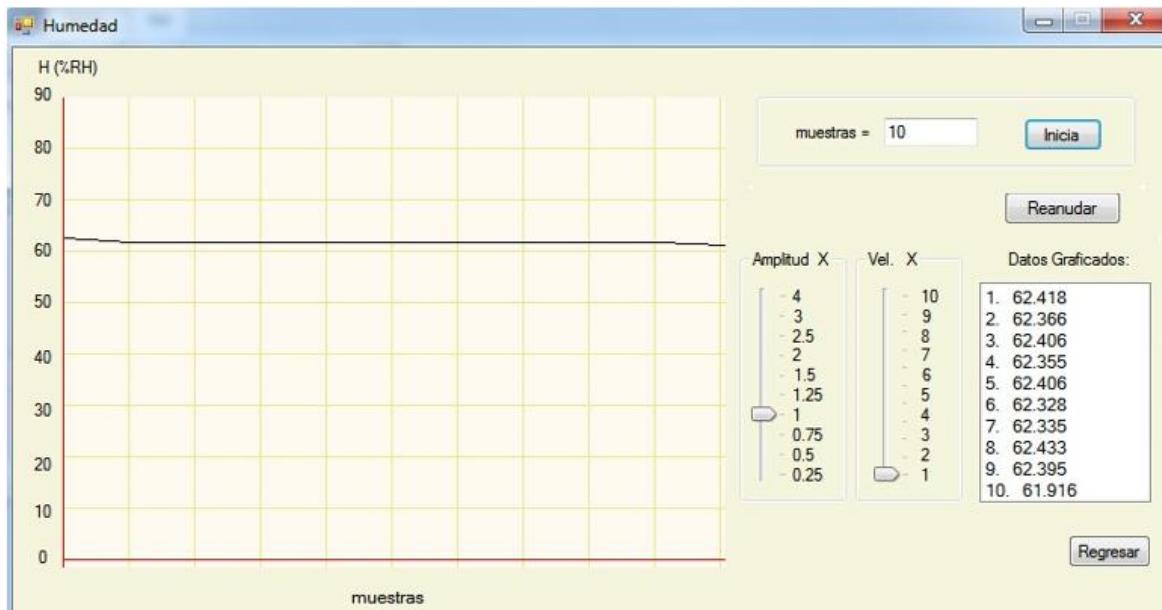


Fig. 124: Señal de 10 muestras del sensor de humedad.



Fig. 125: Lectura del voltaje para el HMZ4331.

Posteriormente, se hizo el cálculo del error entre el valor real detectado por el sensor, y el valor graficado más alejado al real.

$$E_{Hum} = \frac{|62.27\%RH - 61.916\%RH|}{62.27\%RH} * 100\% = 0.568\%$$

6.1.3 Medición de densidad de flujo magnético

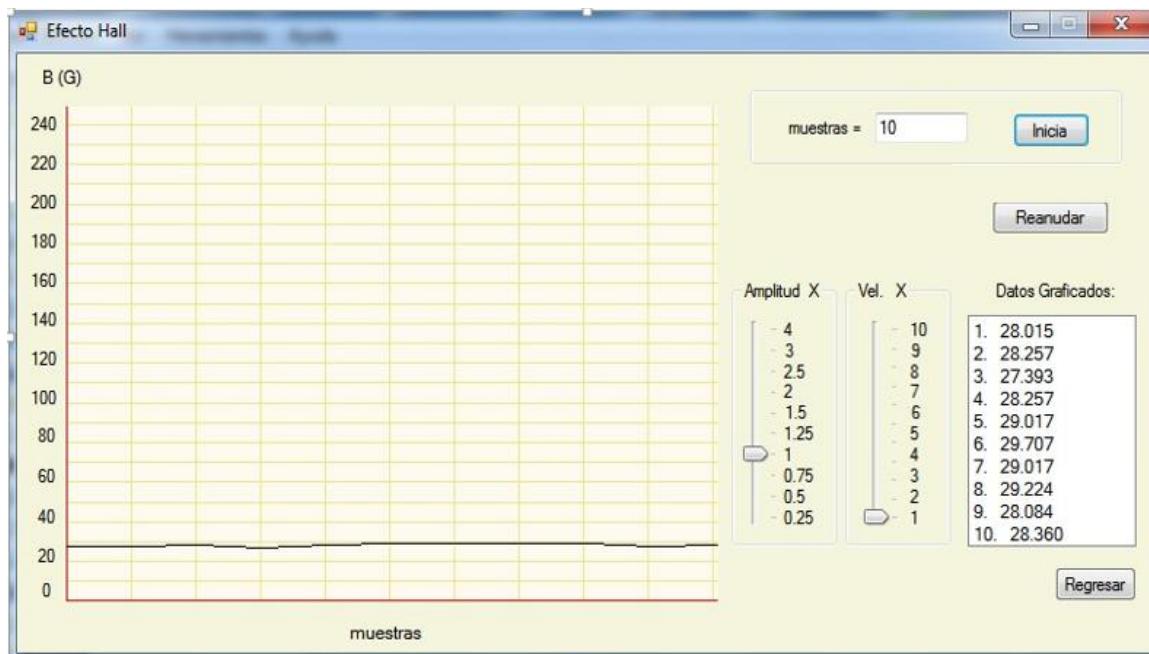


Fig. 126: Señal de 10 muestras del sensor de efecto Hall.

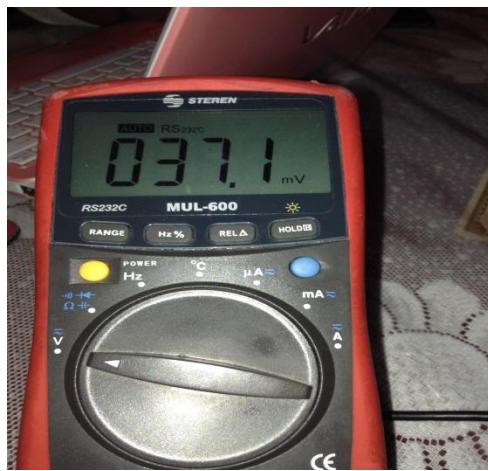


Fig. 127: Lectura del voltaje del sensor de efecto Hall.

Haciendo uso de la función de transferencia del sensor, y la medida del multímetro para esta prueba (Fig. 127), se obtuvo:

$$B = \frac{37.1 \text{ mV}}{1.3 \text{ mV}} = 28.54 \text{ G}$$

Error obtenido:

$$E_{E. \text{ Hall}} = \frac{|28.54G - 29.707G|}{28.54G} * 100\% = 4.088\%$$

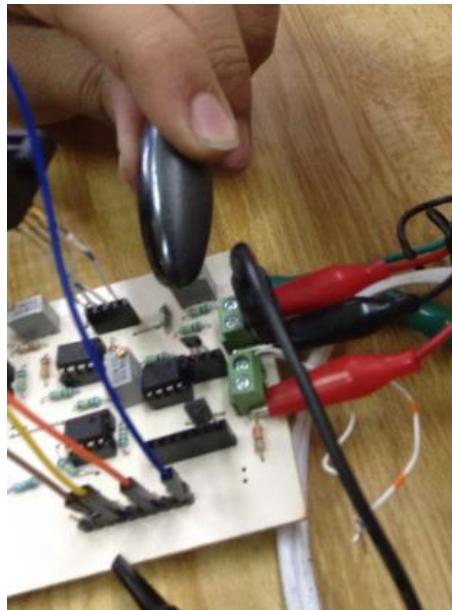


Fig. 128: Inducción de flujo magnético al sensor de efecto Hall.

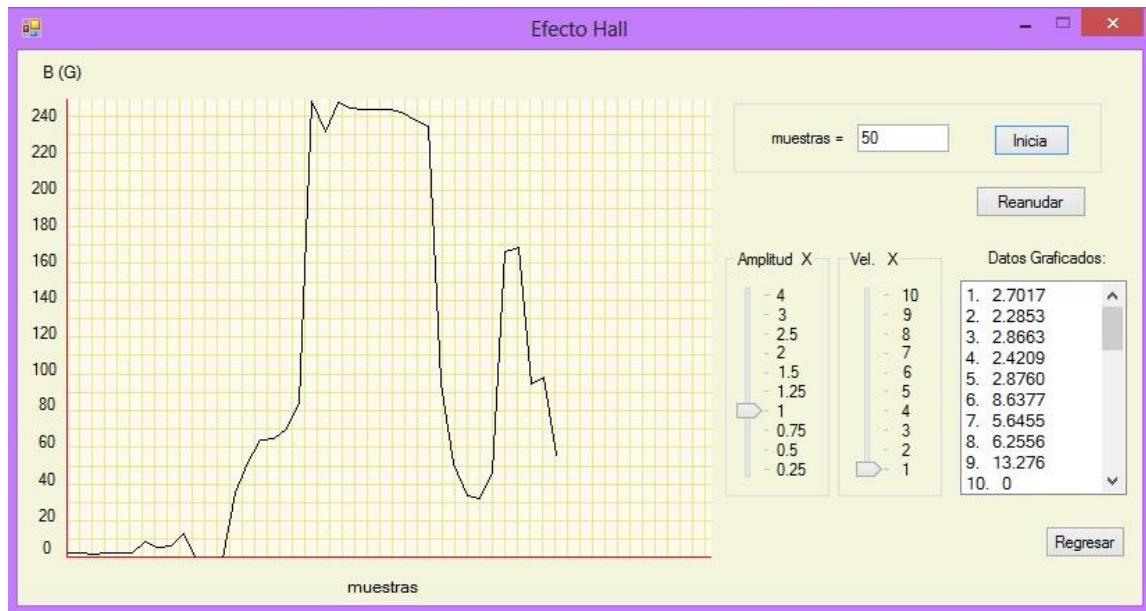


Fig. 129: Respuesta del sensor de efecto Hall, mostrado en la interfaz gráfica creada, al variar la densidad de flujo magnético.

6.1.4 Medición de presión



Fig. 130: Pruebas para el sensor de presión.

Para la prueba de presión, se colocó una manguera delgada de plastico llena de agua, y se conectó al sensor de MPX10DP, previamente llenado con agua para evitar burbujas de aire en su interior. Se alargo sobre una tabla para darle altura, la cual resultó medir 48.5 cm. Utilizando la equivalencia de columnas de agua contra kilopascales, se obtuvo la siguiente medición:

$$P_{KPa} = 48 \text{ cm}_{H_2O} * \frac{1 \text{ m}_{H_2O}}{100 \text{ cm}_{H_2O}} * \frac{1.422 \text{ PSI}}{1 \text{ m}_{H_2O}} * \frac{1 \text{ KPa}}{0.145 \text{ PSI}} = 4.757 \text{ KPa}$$

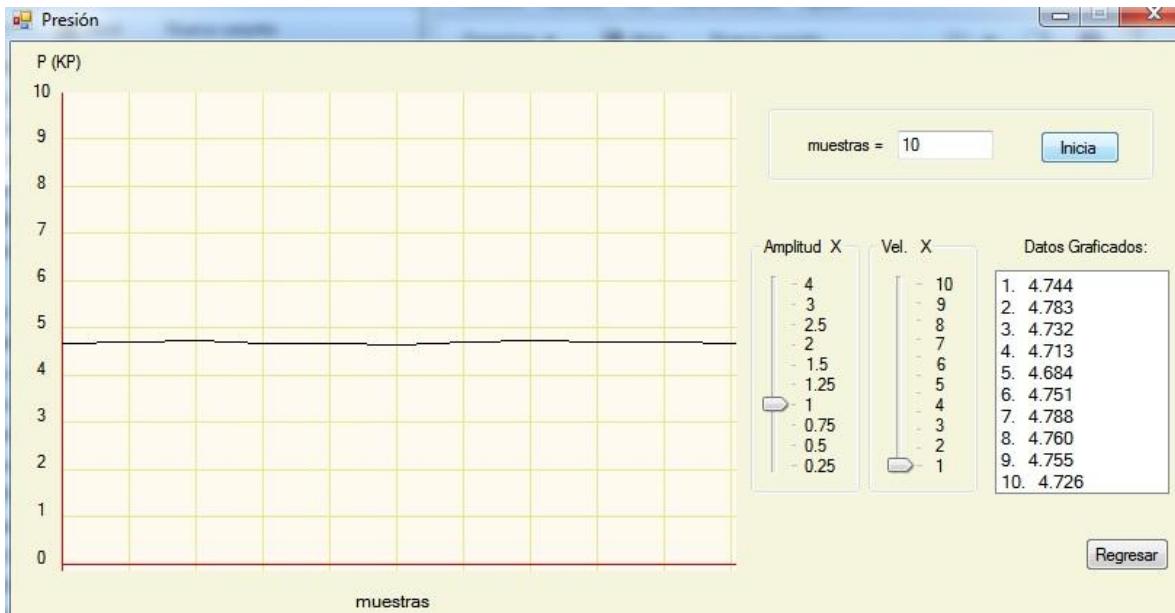


Fig. 131: Señal de 10 muestras, del sensor de presión diferencial.

Error obtenido:

$$E_{E. Hall} = \frac{|4.757 \text{ KPa} - 4.684 \text{ KPa}|}{4.757 \text{ KPa}} * 100\% = 1.534\%$$

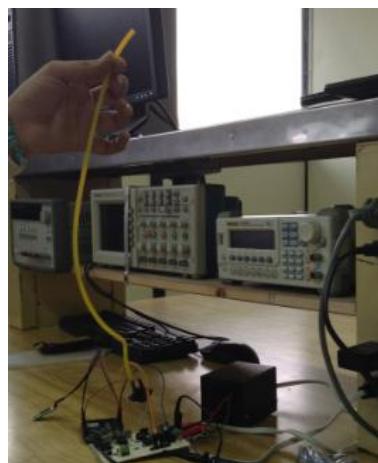


Fig. 132: Nivel de presión uno.

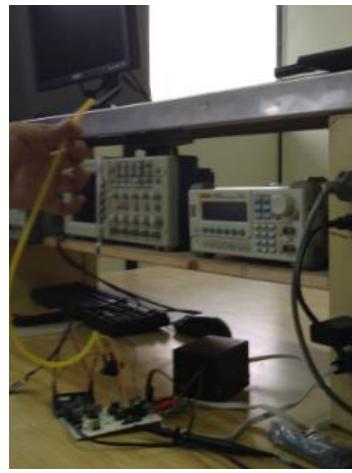


Fig. 133: Nivel de presión dos.



Fig. 134: Respuesta del sensor, al variar la presión, mostrado en la interfaz gráfica creada.

6.2 Para el osciloscopio

6.2.1 Señal rectangular de 30 Hz

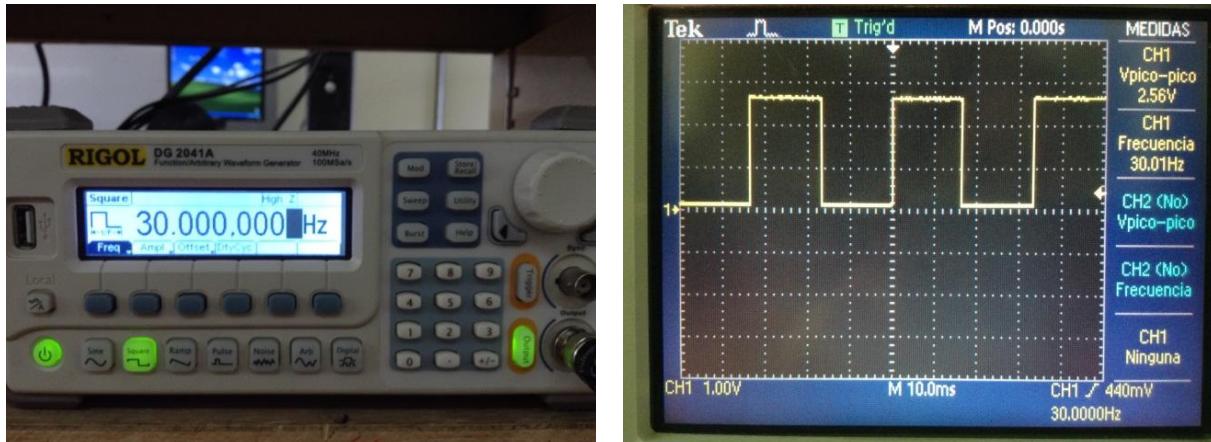


Fig. 135: Señal rectangular de 30 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.

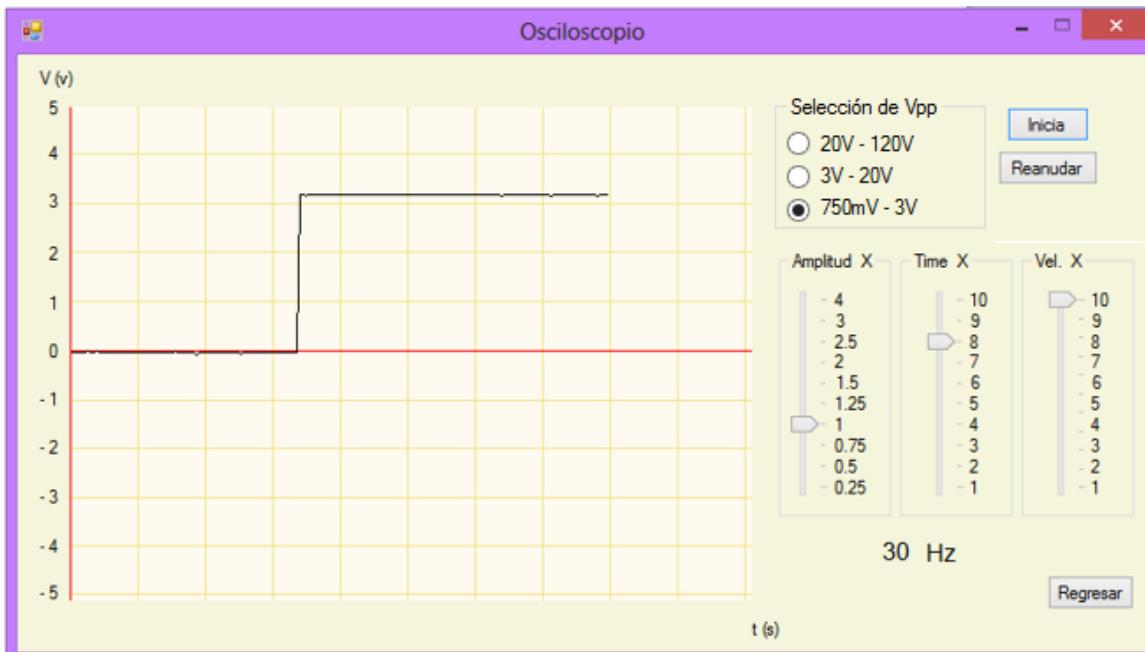


Fig. 136: Señal rectangular de 30 Hz, mostrada en el proyecto desarrollado.

6.2.2 Señal senoidal de 60 Hz

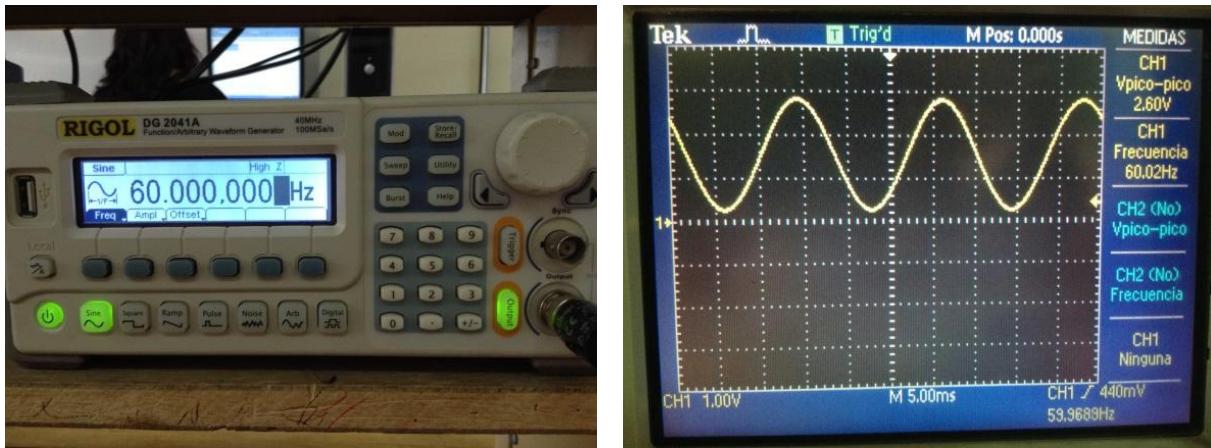


Fig. 137: Señal senoidal de 60 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.

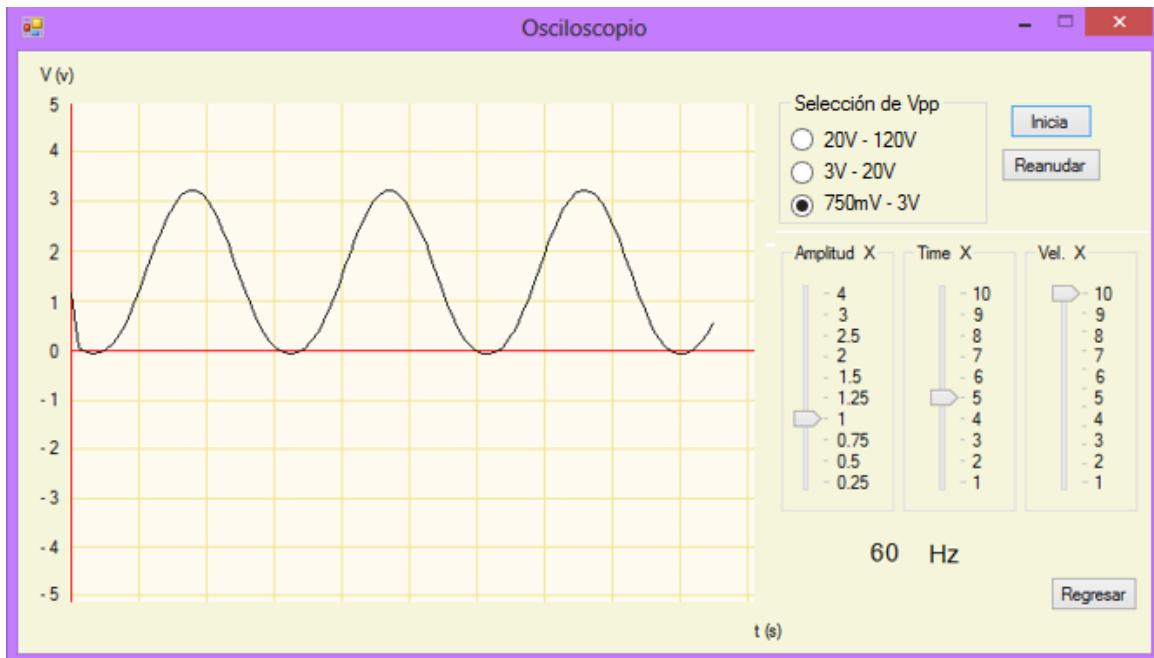


Fig. 138: Señal senoidal de 60 Hz, mostrada en el proyecto desarrollado.

6.2.3 Señal rampa de 120 Hz

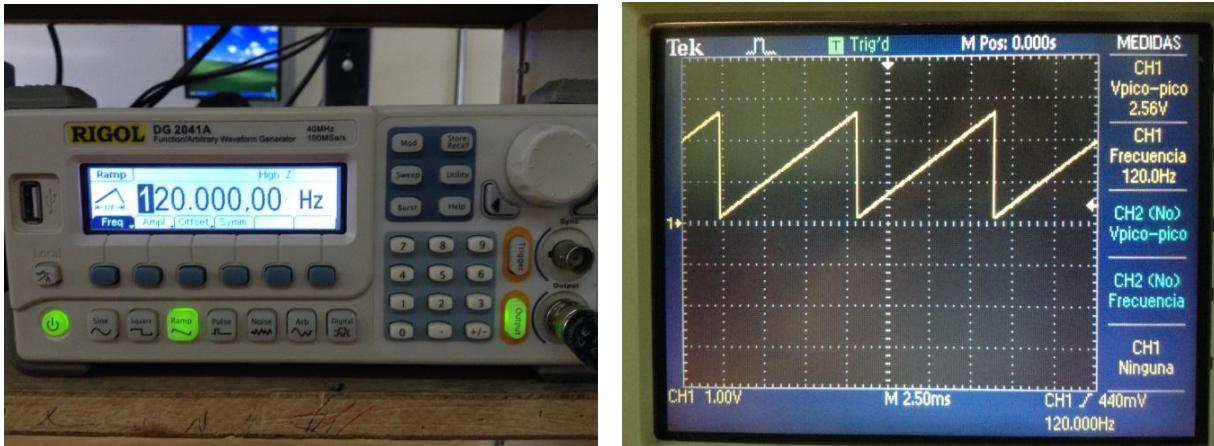


Fig. 139: Señal rampa de 120 Hz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.



Fig. 140: Señal rampa de 120 Hz, mostrada en el proyecto desarrollado.

6.2.4 Señal senoidal de 2.5 KHz

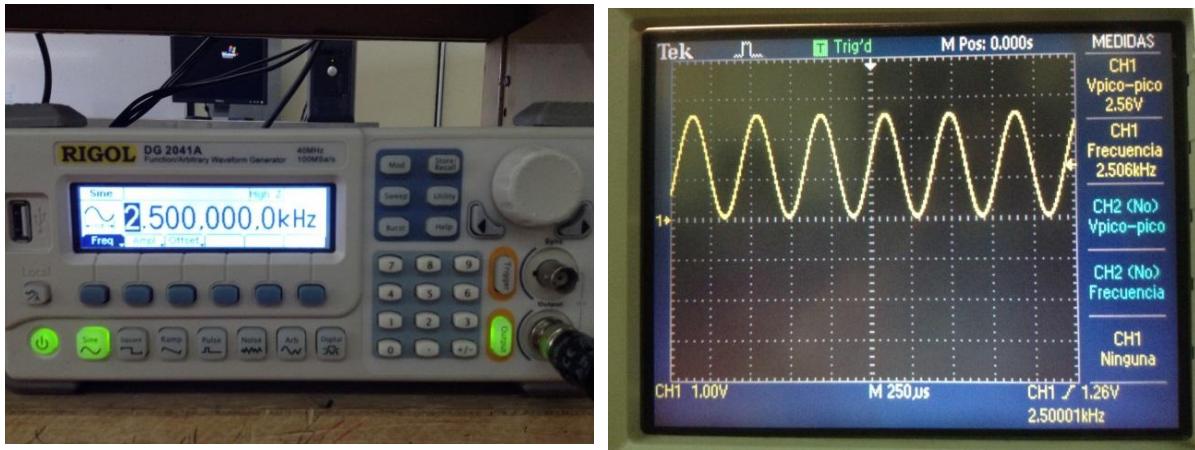


Fig. 141: Señal seno de 2.5 KHz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.

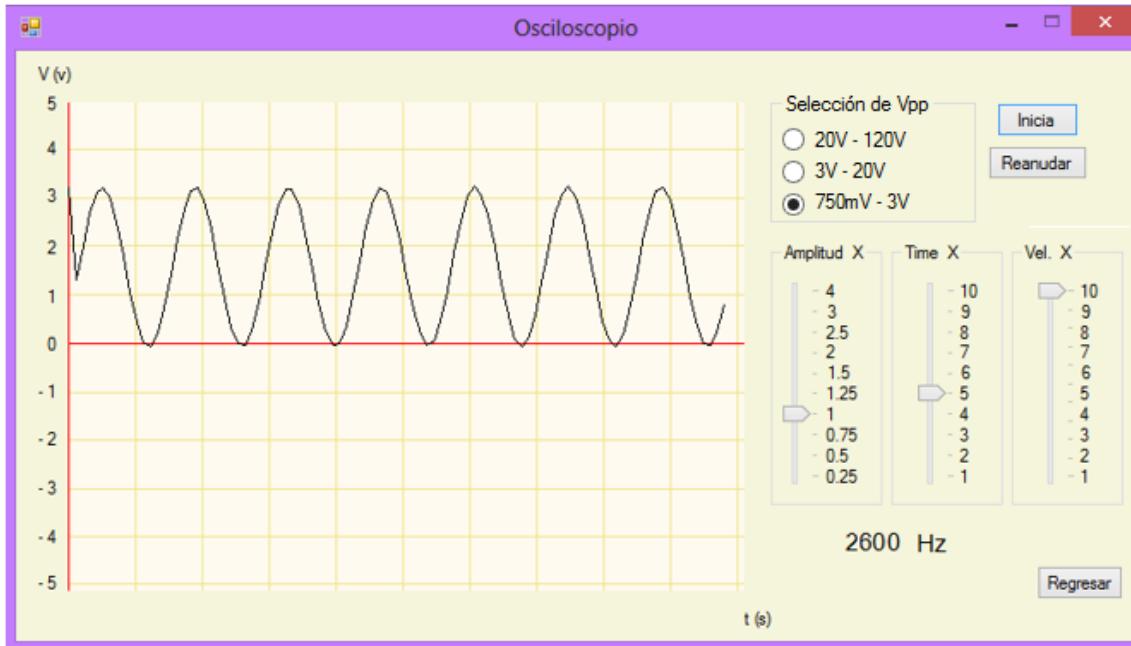


Fig. 142: Señal seno de 2.5 KHz, mostrada en el proyecto desarrollado.

6.2.5 Señal pulso cuadrado a 5 KHz con 30% de ciclo útil

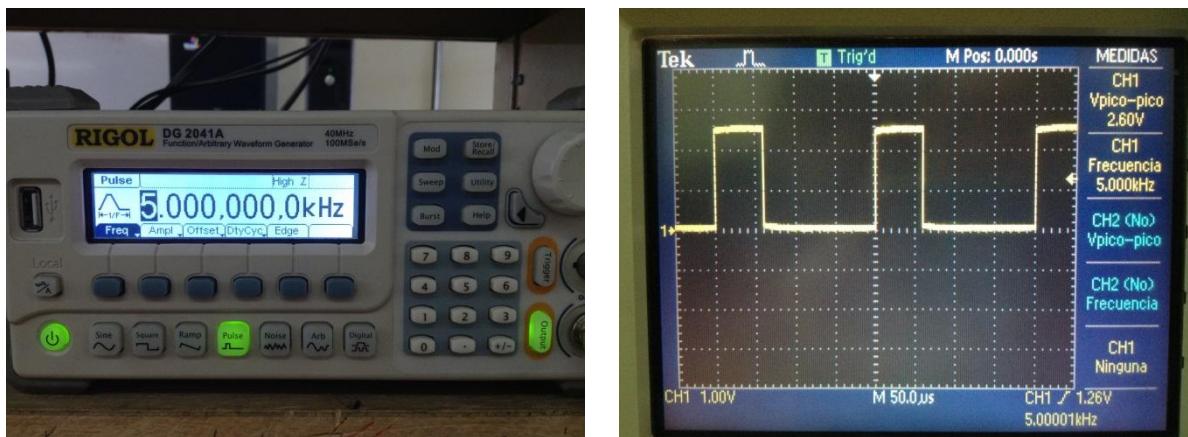


Fig. 143: Pulso cuadrado de 5 KHz, con 30% de ciclo útil, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.

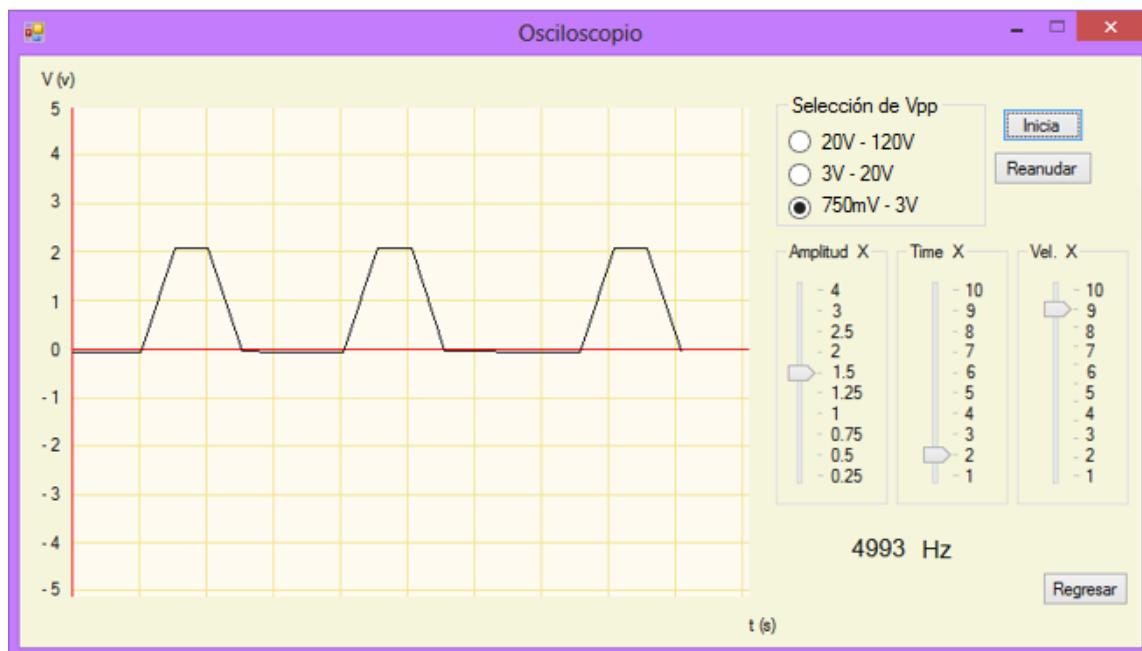


Fig. 144: Pulso cuadrado a 5KHz, con 30% de ciclo útil, mostrada en el proyecto desarrollado.

6.2.6 Señal triangular 4 KHz

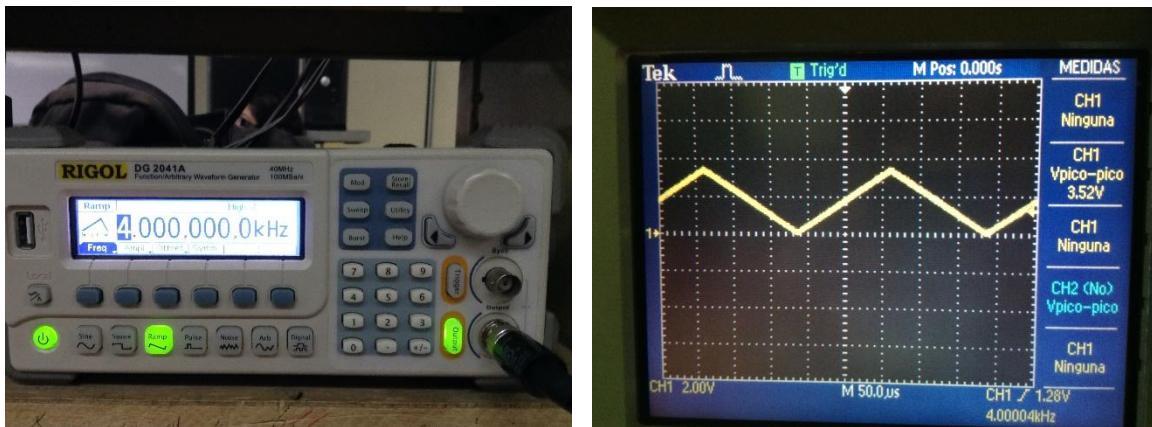


Fig. 145: Señal triangular de 4 KHz, proveniente de un generador de funciones digital y mostrada en un osciloscopio de banco, de la marca Tektronix.

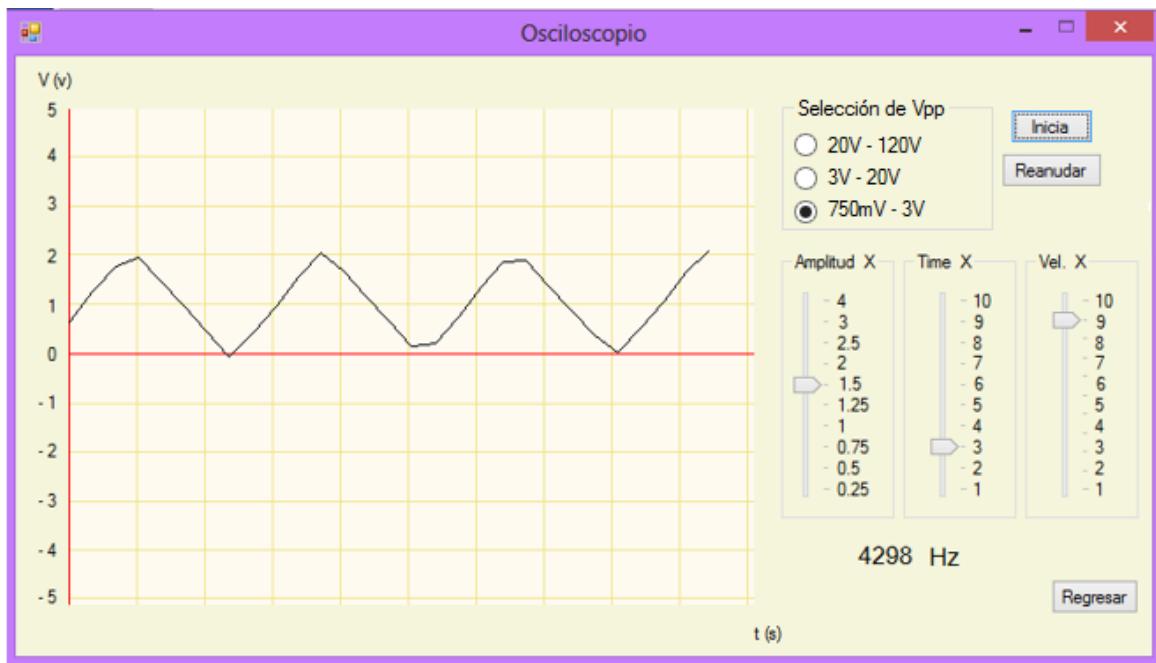


Fig. 146: Señal rampa a 4 KHz, mostrada en el proyecto desarrollado.

6.3 Para generador de funciones

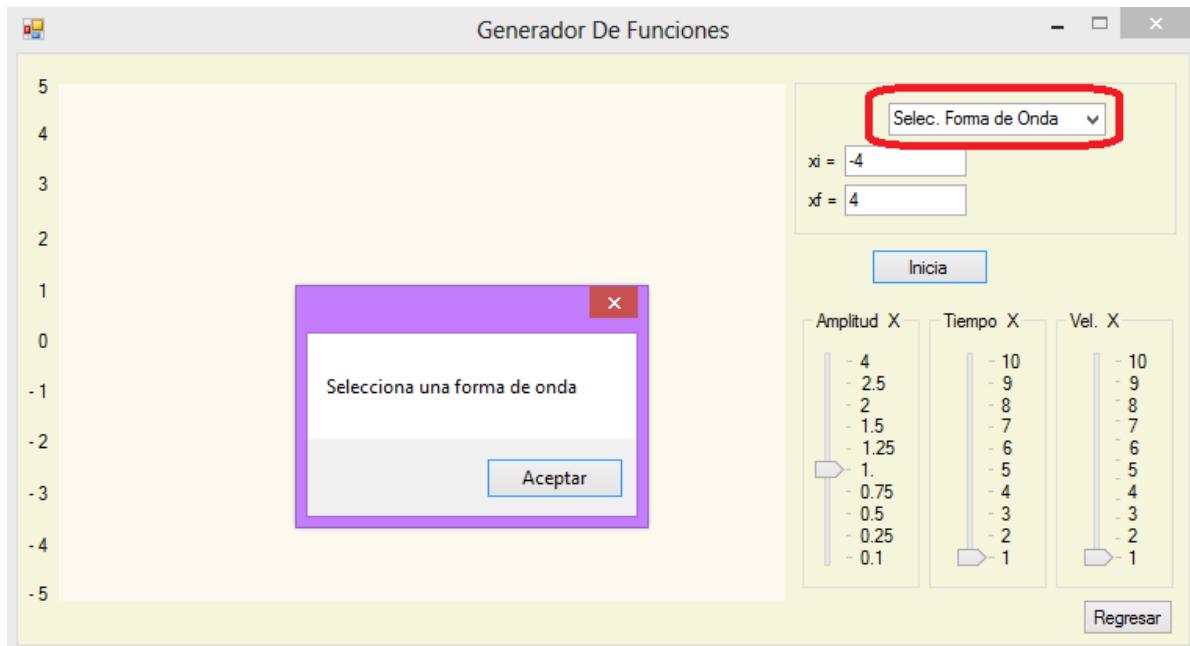


Fig. 147: Mensaje de error, al iniciar graficación y no haber seleccionado alguna función.

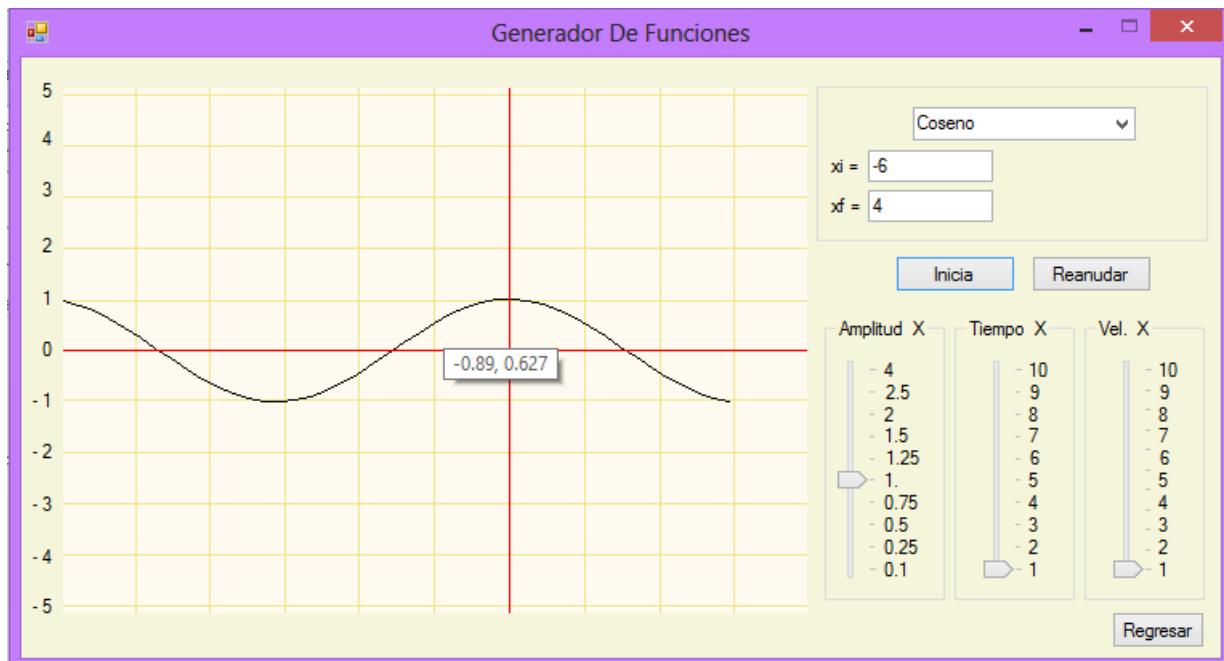


Fig. 148: Generación de función coseno, graficada en un rango de -6 a 4, sin variaciones de amplitud, tiempo o velocidad.

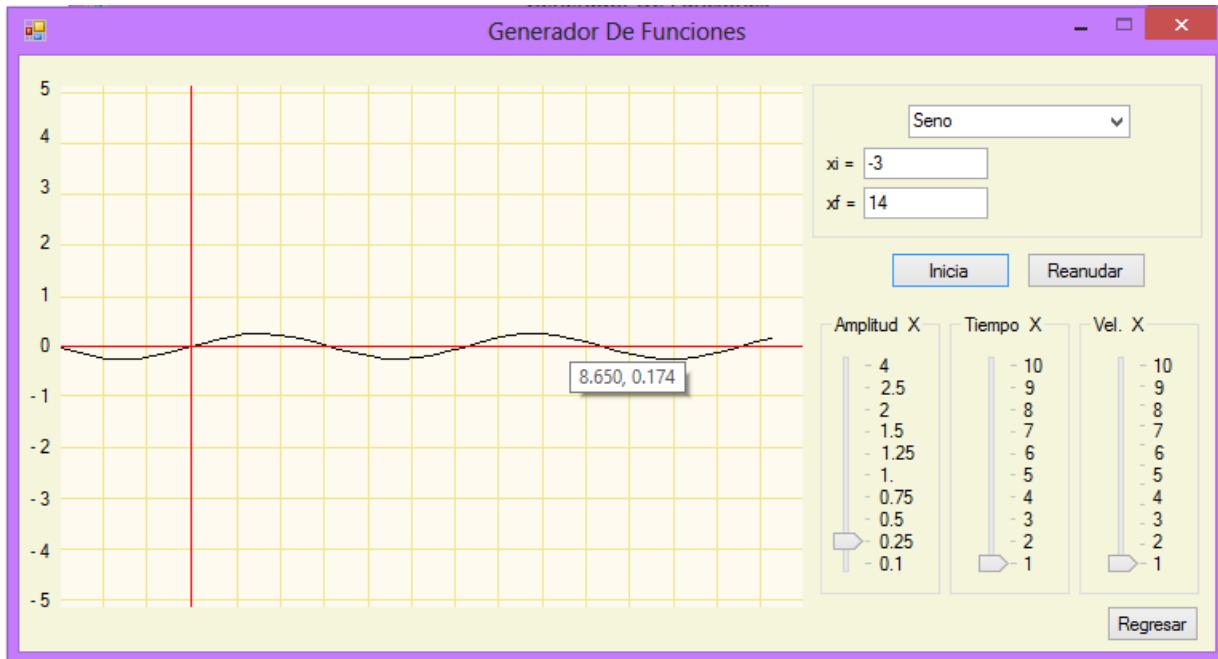


Fig. 149: Generación de función seno, graficada en un rango de -3 a 14, con variación de amplitud.

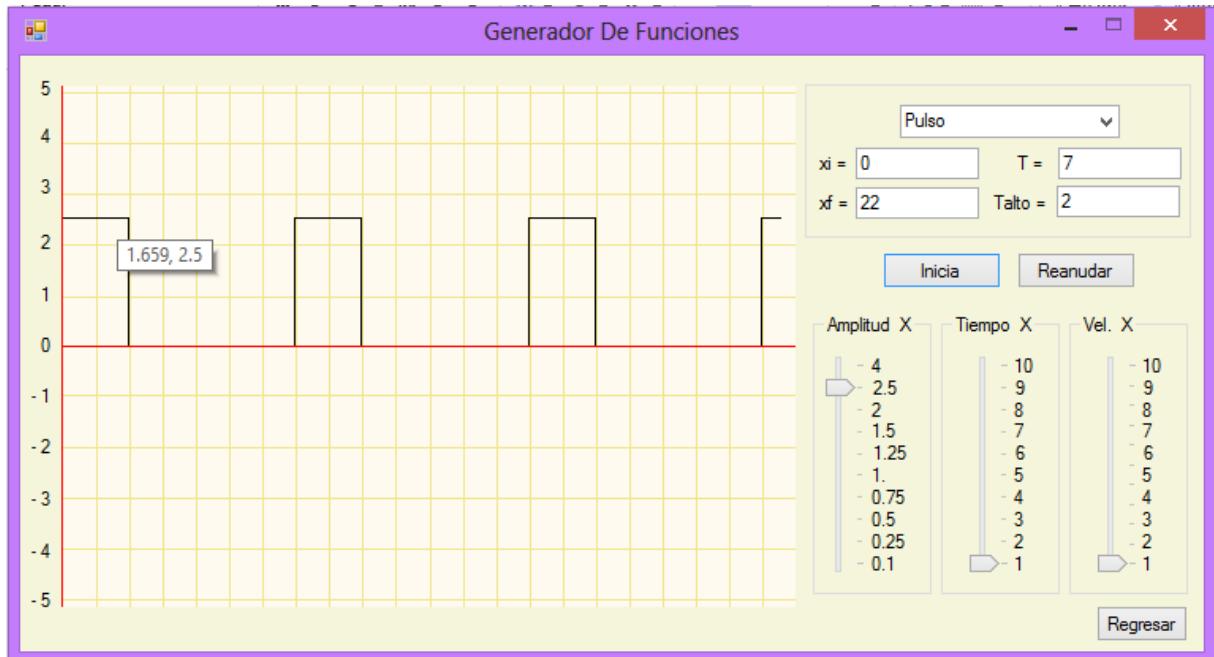


Fig. 150: Generación de una señal pulso, graficada en un rango de 0 a 22, con periodo de 7, ciclo útil de 2 y variación de amplitud.

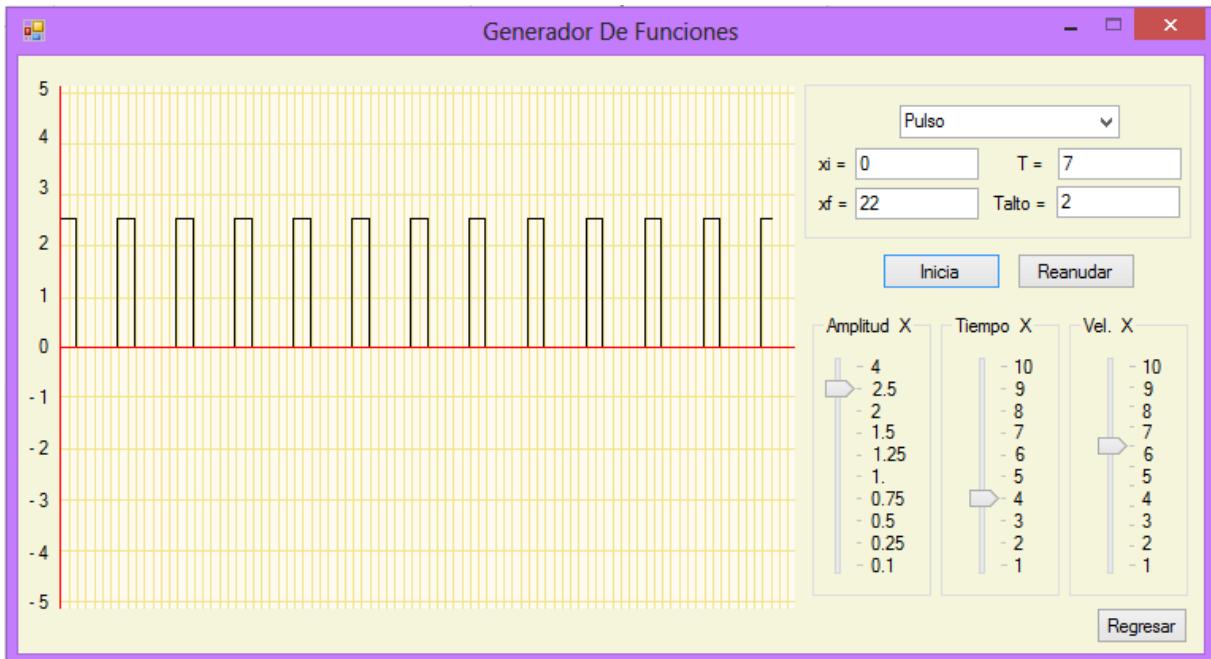


Fig. 151: Generación de una señal pulso, graficada en un rango de 0 a 22, con periodo de 7, ciclo útil de 2, y variación de amplitud, tiempo y velocidad.

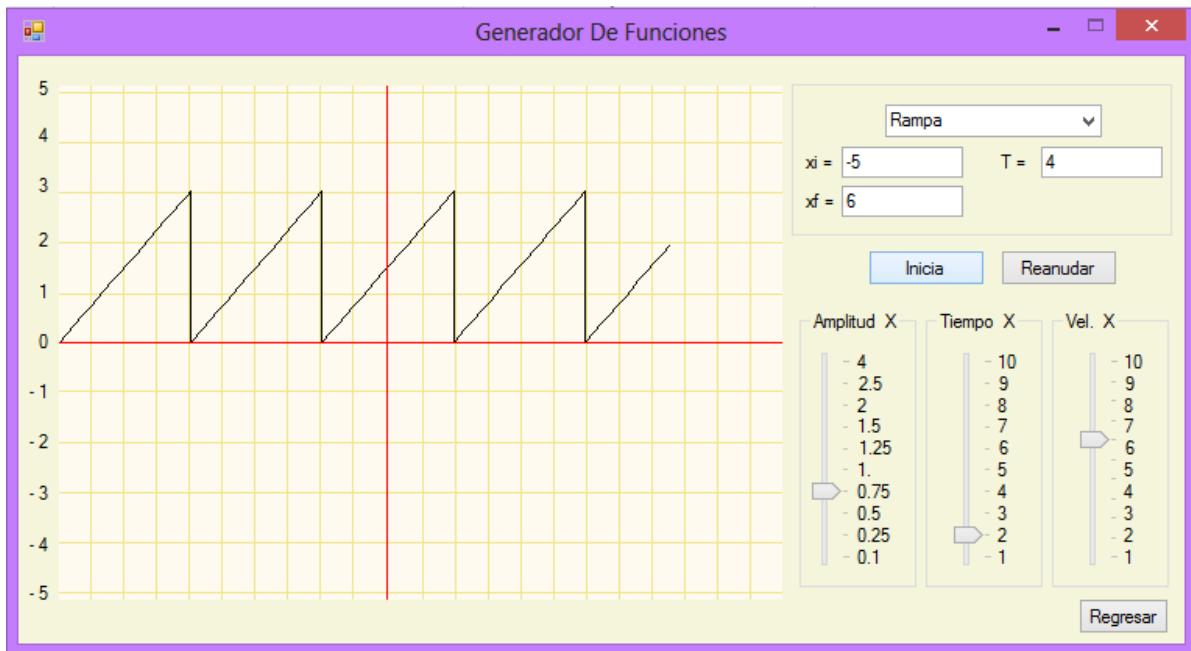


Fig. 152: Generación de la función rampa, graficada en un rango de -5 a 6, con periodo de 4, y variación de amplitud, tiempo y velocidad.

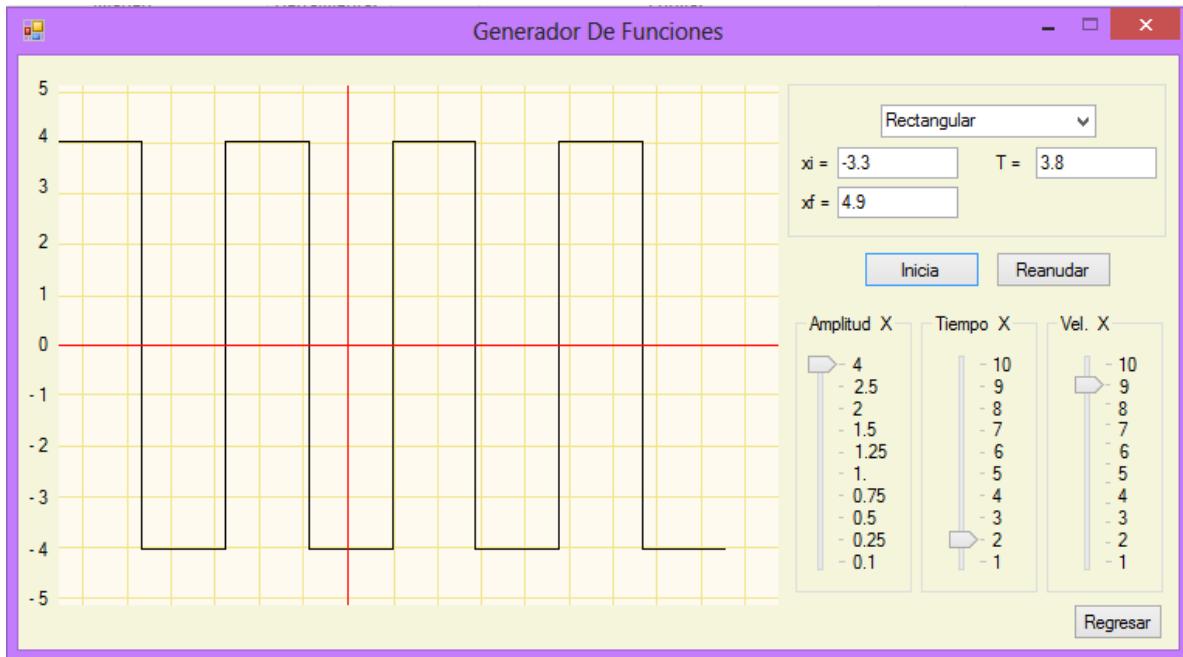


Fig. 153: Generación de la función rectangular, graficada en un rango de -3.3 a 4.9, con periodo de 3.8, y variación de amplitud, tiempo y velocidad.

Conclusiones

Se logró cumplir con los objetivos planteados al inicio de este proyecto, ya que se desarrolló un software y hardware capaz de graficar señales externas de un cierto rango de frecuencias, y monitorear las distintas señales de los sensores propuestos.

Al finalizar el proyecto y analizar los resultados obtenidos, a diferentes frecuencias de las siguientes características:

- El tipo de muestreo aunado a la frecuencia del microcontrolador.
- El tipo de conversión del ADC del microcontrolador (SAR).
- El envío y recepción de datos.
- La graficación de los mismos.
- El tipo de interfaz utilizada (Serial).

se pudo notar que no se obtuvo un ancho de banda muy grande (menor a 5KHz), y debido al tiempo de respuesta del ADC utilizado, no se pudieron graficar señales digitales de ciclo útil muy pequeño, esto es debido al tiempo de conversión del ADC, mientras mayor frecuencia tenga la señal, su periodo disminuirá, por ende el ciclo útil también lo hará, y así con un tiempo de conversión de sólo 1 μ s, los pulsos de un ciclo útil menor a éste, son imposibles de muestrear.

Sin embargo, para los rangos de frecuencia alcanzados, se pueden realizar prácticas en las materias de: mediciones electrónicas, dispositivos electrónicos, algunas de electrónica digital, electrónica lineal e incluso otras, de las diferentes especialidades impartidas (control, acústica, computación, electrónica y comunicaciones), sin la necesidad de utilizar un laboratorio y con la ventaja de tenerlo en casa a un precio accesible, comparándolo con un osciloscopio de banco, mostrados en el estado del arte de este mismo documento.

La portabilidad debido a la comunicación inalámbrica (Bluetooth), permite al usuario poder manejar su sistema sin necesidad de tenerlo cerca al ordenador, y mejor aún, sin alambres, los cuales pudiesen enredarse y causar accidentes.

Al ser compatible con la tecnología Bluetooth, el usuario puede desarrollar aplicaciones para los sistemas operativos Android e iOS, donde simplemente se necesita el uso de una terminal para enviar y recibir los datos del sistema.

Cabe mencionar, que los cuatro sensores agregados al prototipo, al hacer el análisis, su magnitud en la aplicación creada, podrían verse como instrumentos de medición incluidos, que pueden ser utilizados para muchas aplicaciones que el estudiante requiera a lo largo de su trayectoria académica.

Mejoras

Al ver los resultados arrojados por el proyecto y analizar el sistema utilizado, se proponen las siguientes alternativas, para disminuir las limitaciones de éste, así como mejoras en el diseño:

- Emplear un microcontrolador cuya frecuencia máxima de bus, sea mayor a los 24 MHz utilizados en el proyecto.
- Ocupar un ADC externo tipo flash, para mayores velocidades de conversión, debido a que la mayoría de los microcontroladores actuales tienen integrado un convertidor analógico de aproximaciones sucesivas.
- Implementar la tecnología DMA (Direct Memory Access → Acceso Directo a Memoria), para reducir aún más el tiempo de respuesta del sistema.
- Utilizar un PDS (Processor Digital of Signals), para procesar señales a frecuencias más altas y más canales a la vez.
- Implementar otro tipo de interfaz entre el sistema y el ordenador, por ejemplo USB 3.0, o bien, el puerto paralelo.
- Usar un amplificador de RF, para obtener distancias mayores a 10 m con el módulo Bluetooth.
- Utilizar un módulo RF o XBee, para abarcar distancias mayores a 100 m.
- Reducir el tamaño del dispositivo externo, utilizando componentes de montaje superficial, y placas a doble cara.
- Aumentar el número de canales en la aplicación, para poder medir ángulos de desfasamiento entre señales.
- Utilizar el DAC del microcontrolador, para obtener a la salida de éste, una señal analógica, con el uso de la aplicación del generador de funciones de la PC.
- Para estabilizar la alimentación de cada una de las etapas del sistema, es necesario corregir ciertos puntos del circuito final (Fig. 70), como lo son:
 - ✓ Agregar un filtro de línea, a la salida de cada sensor, para eliminar el ruido e interferencia en las señales.
 - ✓ Cada entrada de alimentación, en cada etapa, requiere un capacitor cerámico, denominado “de paso”, de 0.1uF, lo más cercano posible al sensor o amplificador operacional, el diseño omitió esos detalles, pero es importante para asegurar una señal más estable.
 - ✓ Para asegurar que se tenga una señal de alimentación estable, en la fuente se puede agregar un circuito, que permita una mejor regulación ante cambios bruscos en la línea, ya que no se sabe en qué lugar se energizará el sistema, en cualquier lado puede haber fluctuaciones y picos. Entonces, se propone un varistor para la entrada de la línea.
 - ✓ Se puede reemplazar la salida de la fuente, que consta de un simple capacitor, por un filtro π (pi), y calcular los elementos para tener un voltaje de rizo razonable.
 - ✓ Otra solución, es utilizar como fuente de alimentación baterías de 9V a 12V, y regularla a través de un integrado LM2575 (regulador de conmutación), ya que en esta configuración, la corriente de salida es mayor que a la entrada, lo que reduce el consumo de corriente, y

elimina en gran cantidad el ruido debido a la utilización de pilas, sin embargo, este sistema aumenta el costo.

- El circuito presentado en la Fig. 154, se propone como solución a la protección del circuito de atenuación de señales, tanto para la entrada de 1 a 20 V_{rms} , como de 20 a 120 V_{rms} . El circuito propuesto, realiza lo siguiente:
 - ✓ La señal de entrada, se conecta al circuito integrado CD4066, un interruptor analógico, el cual, es activado digitalmente.
 - ✓ La salida del interruptor analógico, es conectada a la etapa de atenuación planteada para el circuito, de igual forma, también es conectada a otra etapa de reducción de voltaje, formada por el integrado denominado U1, y los resistores marcados como R1 y R2, dicha etapa, sirve como circuito muestreador de voltaje.
 - ✓ El circuito conformado por los elementos R3, C1 y D1, se encarga de rectificar la señal de salida muestreada, y por medio del capacitor, obtener a la salida, una señal de CD. La eficiencia de conversión de la señal (de CA a CD), depende de la frecuencia y del circuito RC. Para una señal de baja frecuencia, el circuito entrega una señal con rizo, mientras que para frecuencias arriba de 1 KHz el rizo va disminuyendo.
 - ✓ El resultado de esta señal, será comparada con dos voltajes de referencia, el primero es el voltaje mínimo de rizo, obtenido para una señal de 1Hz, y el segundo, es el voltaje máximo obtenido para una señal de 5KHz, ya que no se conoce la forma de onda que se ingresará al sistema, los voltajes de referencia se proponen a la señal cuyo rizo será mayor, en este caso es en señal triangular.
 - ✓ La siguiente etapa, consta de dos circuitos comparadores y una compuerta OR, los cuales se encargan de abrir y cerrar los interruptores, si el voltaje excede el límite indicado en el sistema, lo que evita que la señal se conecte a la etapa de atenuación, utilizando para ello, los voltajes de referencia indicados anteriormente.
 - ✓ El buffer CD4050, se utiliza para poder activar los interruptores, ya que se requiere un circuito CMOS para esta tarea.

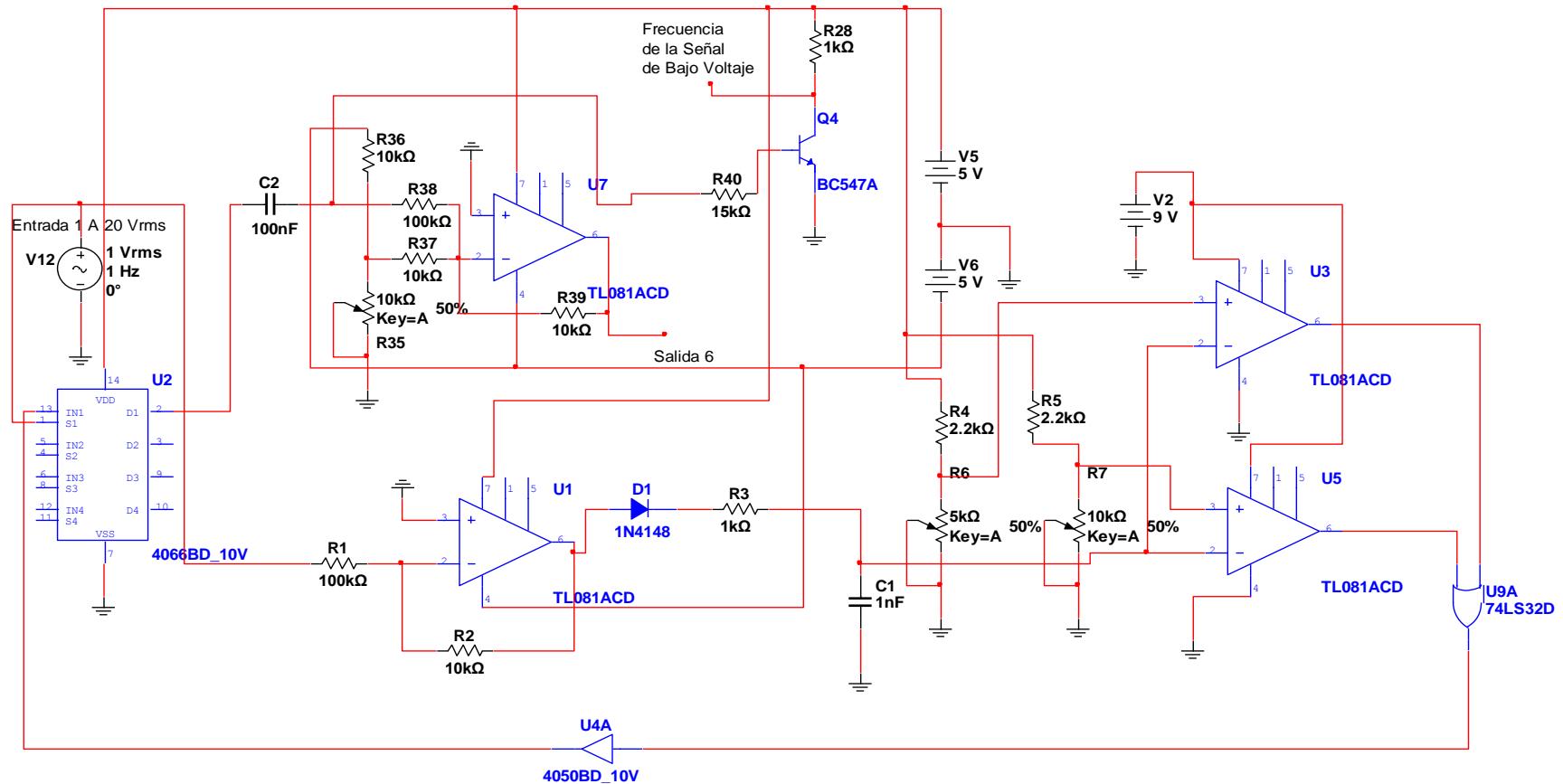


Fig. 154: Circuito de protección, para señales de entrada externa.

Anexo A

Anexo A.1: Tablas comparativas de sensores

Tablas comparativas de diferentes sensores de cada una de las magnitudes a medir, tomando en cuenta sus principales especificaciones, proveedor y precio.

Temperatura						
Sensor	Rango de alimentación	Factor de escala	Precisión	Rango de temperatura	Impedancia de salida	Corriente de salida
LM35DZ	4V - 30V	10mv/°C	±0.5% a 25°C	-55°C a 155°C Fuente simétrica	0.1Ω a 1mA	10mA
LM61CIZ	2.7V - 10V	10mV/°C	±3% (25°C – 85°C)	-30°C a 100°C	800 Ω (max)	10 mA
AD590JH	4V - 30V	1µA/K	±0.5°C	-55°C a 150°C	Alta	373µA max
AD592ANZ	4V - 30V	1µA/K	±0.5°C	-25°C a 105°C	Alta	373µA max
LM135H	2.5V - 10V	10mV/K	±0.5 K	-55°C a 150°C	0.5Ω	400µA a 5mA
LM335Z	3V - 14V	10mV/K	±0.5 K	-55°C a 150°C	0.6Ω	400µA a 5mA
LM34DZ	5V - 30V	10mV/°F	±1°F	-50°F a 300°F Fuente simétrica	0.4Ω/1mA	10 mA
TERMOPAR K	-	Función	±0.75°C	-50°C a 220°C	-	-

Tabla A.1.1: Comparación de características principales de diferentes sensores de temperatura.

Temperatura			
Matrícula	Distribuidor	Precio	Disponibilidad
LM35DZ	ROBODACTA	\$30.00	5
		\$ 25.22	6131
		\$12.10	380
		\$202.74	9
		\$75.64	51
		\$183.58	11
		\$10.08	24
		\$37.13	56
		\$790.00	1
AG			

Tabla A.1.2: Comparación de precios, proveedores y disponibilidad en sensores de temperatura.

Presión						
Sensor	Rango de alimentación	Factor de escala	Precisión	Rango de presión	Impedancia de salida	Corriente de salida
MPX10DP	3V - 6 V	3.5 mV/KPa	$\pm 0.1\%$	0Pa -10 KPa	750Ω - 1250Ω	-
MPX2010DP	10V - 16 V	2.5 mV/KPa	$\pm 0.1\%$	0Pa - 10 KPa	1400Ω - 3000Ω	-
MPX2200DP	10V - 16 V	0.2 mv/KPa	$\pm 0.1\%$	0Pa - 200KPa	1400Ω - 3000Ω	-
MPX5700DP	4.75V - 5.25V	6.4 mV/KPa	$\pm 2.5\%$	0Pa - 700 KPa	1400Ω - 3000Ω	-
MPL4115A2	1.95V - 3.6 V	20 bits, I2C	-	20Pa - 110 KPa	Baja	
PH1115	1.95V - 3.6 V	-	-	20Pa - 250 KPa	1000Ω	-

Tabla A.1.3: Comparación de características principales de diferentes sensores de presión.

Presión			
Matrícula	Distribuidor	Precio	Disponibilidad
MPX10DP	AG	\$122.05	34
MPX2010DP		\$150.29	30
MPX2200DP		\$192.65	30
MPX5700DP		\$221.90	18
MPL4115A2	ROBODACTA	\$240.00	5
PH1115	E.ESTUDIO	\$1187.55	EXISTENCIA
MICROCELDA DE CARGA		\$194.98	EXISTENCIA

Tabla A.1.4: Comparación de precios, proveedores y disponibilidad en sensores de presión.

Humedad					
Sensor	Rango de alimentación	Factor de escala	Rango de %RH	Precisión	Voltaje de salida
HMZ-433A1	5V	33 mV/%RH	20 a 90%	$\pm 5\%$ a 25°C	0V - 3.3V
HIH-4030	4V - 5.8V	30.680 mV/%RH	0 a 100%	$\pm 3.5\%$ RH	0.958V - 0.813 V

Tabla A.1.5: Comparación de características principales de diferentes sensores de humedad.

Humedad			
Matrícula	Distribuidor	Precio	Disponibilidad
HMZ-433A1	AG	\$129.18	479
HIH-4030	ROBODACTA	\$290.00	2

Tabla A.1.6: Comparación de precios, proveedores y disponibilidad en sensores de humedad.

Densidad de flujo magnético					
Sensor	Rango de alimentación	Factor de escala	Precisión	Rango de flujo magnético	Impedancia de salida
DN6851A	3.6 a 16 V	Funciona como un Switch			
UGN3503	4.5 a 6 V	1.3 mV/G	-	-900G a 900 G	50 Ω
US1881KUA	3.5 a 24 V	Funciona como un Latch			

Tabla A.1.7: Comparación de características principales de diferentes sensores de efecto Hall.

Densidad de flujo magnético			
Matrícula	Distribuidor	Precio	Disponibilidad
DN6851A	AG	\$24.20	1015
UGN3503		\$18.15	77
US1881KUA		\$8.02	273

Tabla A.1.8: Comparación de precios, proveedores y disponibilidad en sensores de efecto Hall.

 Sensores que debido a sus características, son aptos para el prototipo.

 Sensores elegidos por reunir las características necesarias para el prototipo y tener un precio accesible.

Consultas para las tablas del Anexo A.1:

¹ http://www.agelectronica.com/Virtual_Shop/index.asp → AG (Consulta 20/05/2013)

² http://www.robodacta.mx/index.php?dispatch=categories.view&category_id=243 → ROBODACTA (Consulta 20/05/2013)

³ <http://www.electronicaestudio.com/sensores.htm> → E. ESTUDIO (Consulta 20/05/2013)

Anexo A.2: Tabla comparativa de características de diferentes microcontroladores

Microcontrolador \ Característica	FRDMKL25Z 128 BOARD	Stellaris LM4F120 LaunchPad Evaluation Board (EK-LM4F120XL)	STM32VL Discovery BOARD	Arduino Due	LAUNCHXL-F28027 - C2000 Piccolo LaunchPad
Procesador	ARM Cortex M0+	ARM Cortex M4F	ARM Cortex M3	ARM Cortex M3	ARM Cortex M
Frecuencia de operación.	48 MHz	80 MHz	24 MHZ	84 MHz	60 MHz
Tamaño de palabra	32 Bits	32 Bits	32 Bits	32 Bits	32 bits
Tipo de ADC.	SAR	SAR	SAR	SAR	SAR
Resolución máxima del ADC	16 Bits	12 Bits	12 Bits	12 Bits	12 Bits
Número de canales analógicos.	10	12	21	12	12
Memoria flash	128KB	256 KB	256 KB	512 KB	64 KB
SRAM	16 KB	32KB	32 KB	96 KB	16 KB
Operación en bajo consumo.	Si	Si	Si	Si	Si
Precio	\$250.00	\$103.00	\$200.00	\$682.11	\$219.16

Tabla A.2.1: Comparación de características principales de diferentes microcontroladores.

 Microcontrolador elegido, por reunir las características necesarias para el prototipo y tener un precio accesible.

Consultas para las tablas del Anexo A.1:

¹ http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z → FRDMKL25ZUM Y FORM (Consulta 25/05/2013)

² <http://www.ti.com/lit/ds/sprs523j/sprs523j.pdf> → Piccolo (Consulta 25/05/2013)

³ http://www.st.com/st-web-ui/static/active/jp/resource/technical/document/user_manual/CD00267113.pdf → STM32 (Consulta 20/05/2013)

⁴ <http://www.ti.com/lit/ug/spmu289c/spmu289c.pdf> → Stellaris (Consulta 20/05/2013)

⁵ <http://arduino.cc/en/Main/arduinoBoardDue> → Arduino Due (Consulta 20/05/2013)

Anexo B

Anexo B.1: Hojas de especificaciones del sensor de temperatura LM61CIZ



LM61

www.ti.com

SNIS121I – JUNE 1999 – REVISED FEBRUARY 2013

LM61 2.7V, SOT-23 or TO-92 Temperature Sensor

Check for Samples: [LM61](#)**FEATURES**

- Calibrated Linear Scale Factor of +10 mV/°C
- Rated for Full -30° to +100°C Range
- Suitable for Remote Applications
- UL Recognized Component

APPLICATIONS

- Cellular Phones
- Computers
- Power Supply Modules
- Battery Management
- FAX Machines
- Printers
- HVAC
- Disk Drives
- Appliances

DESCRIPTION

The LM61 is a precision integrated-circuit temperature sensor that can sense a -30°C to +100°C temperature range while operating from a single +2.7V supply. The LM61's output voltage is linearly proportional to Celsius (Centigrade) temperature (+10 mV/°C) and has a DC offset of +600 mV. The offset allows reading negative temperatures without the need for a negative supply. The nominal output voltage of the LM61 ranges from +300 mV to +1600 mV for a -30°C to +100°C temperature range. The LM61 is calibrated to provide accuracies of $\pm 2.0^{\circ}\text{C}$ at room temperature and $\pm 3^{\circ}\text{C}$ over the full -25°C to +85°C temperature range.

The LM61's linear output, +600 mV offset, and factory calibration simplify external circuitry required in a single supply environment where reading negative temperatures is required. Because the LM61's quiescent current is less than 125 μA , self-heating is limited to a very low 0.2°C in still air. Shutdown capability for the LM61 is intrinsic because its inherent low power consumption allows it to be powered directly from the output of many logic gates.

Table 1. Key Specifications

	VALUE	UNIT
Accuracy at 25°C	± 2.0 or ± 3.0	°C (max)
Accuracy for -30°C to +100°C	± 4.0	°C (max)
Accuracy for -25°C to +85°C	± 3.0	°C (max)
Temperature Slope	+10	mV/°C
Power Supply Voltage Range	+2.7 to +10	V
Current Drain @ 25°C	125	μA (max)
Nonlinearity	± 0.8	°C (max)
Output Impedance	800	Ω (max)



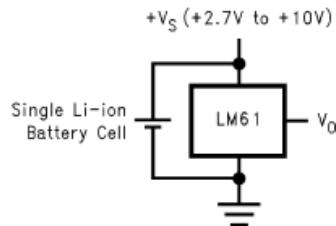
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

PRODUCTION DATA information is current as of publication date.
Products conform to specifications per the terms of the Texas
Instruments standard warranty. Production processing does not
necessarily include testing of all parameters.

Copyright © 1999–2013, Texas Instruments Incorporated

TYPICAL APPLICATION



$$A. \quad V_O = (+10 \text{ mV}/^\circ\text{C} \times T \text{ }^\circ\text{C}) + 600 \text{ mV}$$

Figure 1. Full-Range Centigrade Temperature Sensor (-30°C to $+100^\circ\text{C}$) Operating from a Single Li-Ion Battery Cell

Table 2. Temperature and Typical V_O Values of Figure 1

TEMPERATURE (T)	TYPICAL V_O
$+100^\circ\text{C}$	+1600 mV
$+85^\circ\text{C}$	+1450 mV
$+25^\circ\text{C}$	+850 mV
0°C	+600 mV
-25°C	+350 mV
-30°C	+300 mV

CONNECTION DIAGRAMS

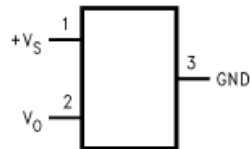


Figure 2. SOT-23 (Top View)
See Package Number DBZ0003A

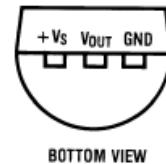


Figure 3. TO-92 (Bottom View)
See Package Number LP0003A



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

Absolute Maximum Ratings⁽¹⁾

Supply Voltage	+12V to -0.2V
Output Voltage	(+V _S + 0.6V) to -0.6V
Output Current	10 mA
Input Current at any pin ⁽²⁾	5 mA
Storage Temperature	-65°C to +150°C
Maximum Junction Temperature (T _{JMAX})	+125°C
ESD Susceptibility ⁽³⁾	
Human Body Model	2500V
Machine Model	250V

(1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the [Electrical Characteristics](#). The guaranteed specifications apply only for the test conditions listed. Some performance characteristics may degrade when the device is not operated under the listed test conditions.

(2) When the input voltage (V_I) at any pin exceeds power supplies (V_I < GND or V_I > +V_S), the current at that pin should be limited to 5 mA.
(3) The human body model is a 100 pF capacitor discharged through a 1.5 kΩ resistor into each pin. The machine model is a 200 pF capacitor discharged directly into each pin.

Operating Ratings⁽¹⁾

Specified Temperature Range:	T _{MIN} ≤ T _A ≤ T _{MAX}
LM61C	-30°C ≤ T _A ≤ +100°C
LM61B	-25°C ≤ T _A ≤ +85°C
Supply Voltage Range (+V _S)	+2.7V to +10V
Thermal Resistance, θ _{JA} ⁽²⁾ SOT-23 TO-92	450°C/W 180°C/W

Soldering process must comply with Reflow Temperature Profile specifications. Refer to <http://www.ti.com/packaging>.⁽³⁾

- (1) Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the [Electrical Characteristics](#). The guaranteed specifications apply only for the test conditions listed. Some performance characteristics may degrade when the device is not operated under the listed test conditions.
(2) The junction to ambient thermal resistance (θ_{JA}) is specified without a heat sink in still air.
(3) Reflow temperature profiles are different for lead-free and non-lead-free packages.

Electrical Characteristics

Unless otherwise noted, these specifications apply for $+V_S = +3.0 \text{ V}_{\text{DC}}$. **Boldface limits apply for $T_A = T_J = T_{\text{MIN}}$ to T_{MAX}** ; all other limits $T_A = T_J = 25^\circ\text{C}$.

PARAMETER	TEST CONDITIONS	TYPICAL ⁽¹⁾	LM61B	LM61C	UNITS (LIMIT)
			LIMITS ⁽²⁾	LIMITS ⁽²⁾	
Accuracy ⁽³⁾			±2.0	±3.0	${}^\circ\text{C}$ (max)
			±3.0	±4.0	${}^\circ\text{C}$ (max)
Output Voltage at 0°C		+600			mV
Nonlinearity ⁽⁴⁾			±0.6	±0.8	${}^\circ\text{C}$ (max)
Sensor Gain (Average Slope)		+10	+9.7	+9.6	$\text{mV}/{}^\circ\text{C}$ (min)
			+10.3	+10.4	$\text{mV}/{}^\circ\text{C}$ (max)
Output Impedance	$+3.0\text{V} \leq +V_S \leq +10\text{V}$ $-30^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$, $+V_S = +2.7\text{V}$ $+85^\circ\text{C} \leq T_A \leq +100^\circ\text{C}$, $+V_S = +2.7\text{V}$		0.8 2.3 5	0.8 2.3 5	k Ω (max) k Ω (max) k Ω (max)
Line Regulation ⁽⁵⁾	$+3.0\text{V} \leq +V_S \leq +10\text{V}$		±0.7	±0.7	mV/V (max)
	$+2.7\text{V} \leq +V_S \leq +3.3\text{V}$		±5.7	±5.7	mV (max)
Quiescent Current	$+2.7\text{V} \leq +V_S \leq +10\text{V}$	82	125	125	μA (max)
			155	155	μA (max)
Change of Quiescent Current	$+2.7\text{V} \leq +V_S \leq +10\text{V}$	±5			μA
Temperature Coefficient of Quiescent Current			0.2		$\mu\text{A}/{}^\circ\text{C}$
Long Term Stability ⁽⁶⁾	$T_J=T_{\text{MAX}}=+100^\circ\text{C}$, for 1000 hours	±0.2			${}^\circ\text{C}$

(1) Typicals are at $T_J = T_A = 25^\circ\text{C}$ and represent most likely parametric norm.

(2) Limits are guaranteed to TI's AOQL (Average Outgoing Quality Level).

(3) Accuracy is defined as the error between the output voltage and $+10 \text{ mV}/{}^\circ\text{C}$ times the device's case temperature plus 600 mV, at specified conditions of voltage, current, and temperature (expressed in ${}^\circ\text{C}$).

(4) Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

(5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

(6) For best long-term stability, any precision circuit will give best results if the unit is aged at a warm temperature, and/or temperature cycled for at least 46 hours before long-term life test begins. This is especially true when a small (Surface-Mount) part is wave-soldered; allow time for stress relaxation to occur. The majority of the drift will occur in the first 1000 hours at elevated temperatures. The drift after 1000 hours will not continue at the first 1000 hour rate.

Typical Performance Characteristics

The LM61 in the SOT-23 package mounted to a printed circuit board as shown in Figure 14 was used to generate the following thermal curves.

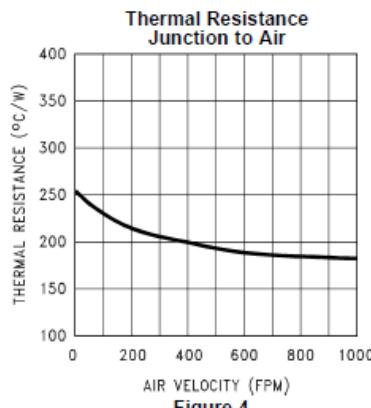


Figure 4.

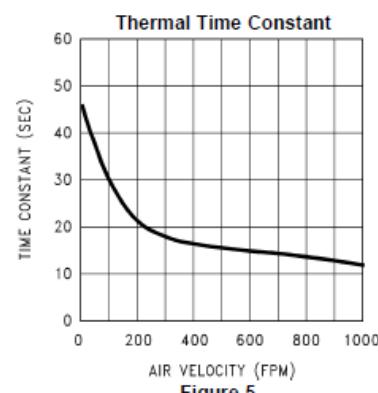


Figure 5.

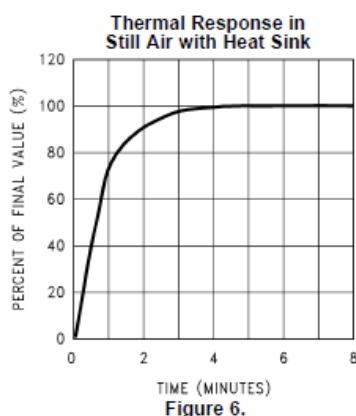


Figure 6.

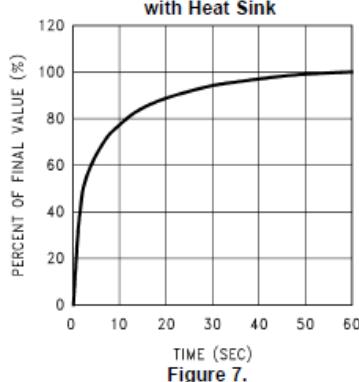


Figure 7.

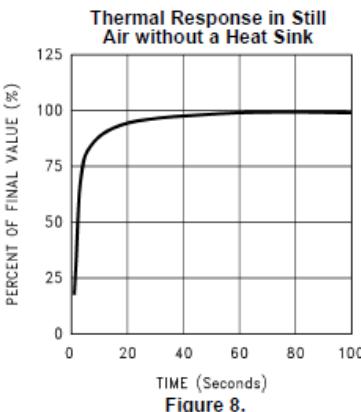


Figure 8.

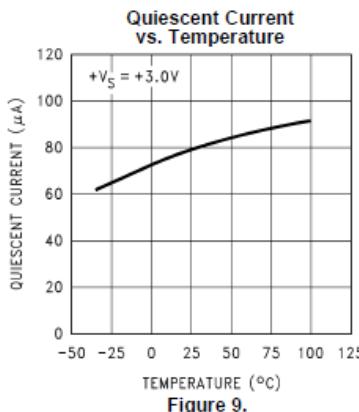


Figure 9.

Typical Performance Characteristics (continued)

The LM61 in the SOT-23 package mounted to a printed circuit board as shown in Figure 14 was used to generate the following thermal curves.

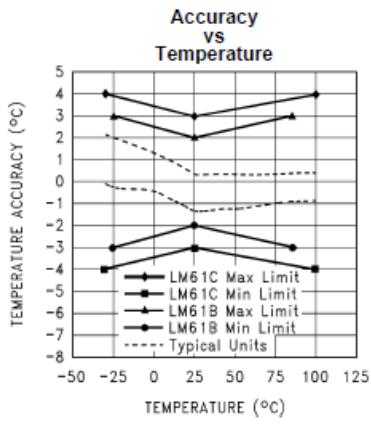


Figure 10.

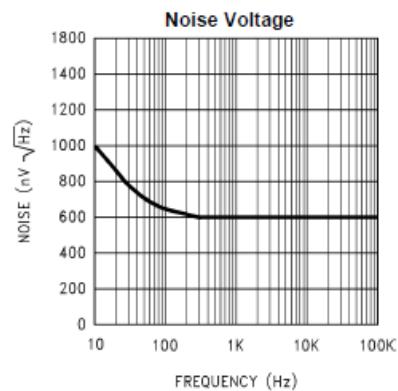


Figure 11.

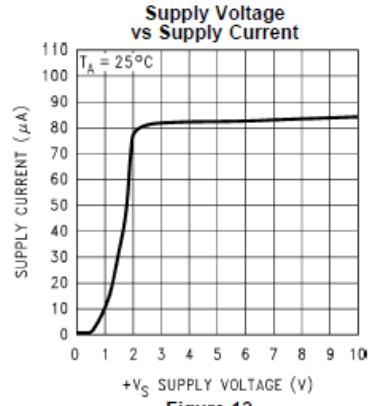


Figure 12.

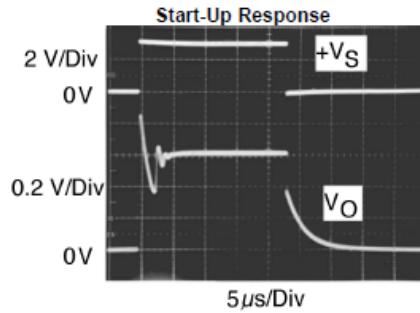
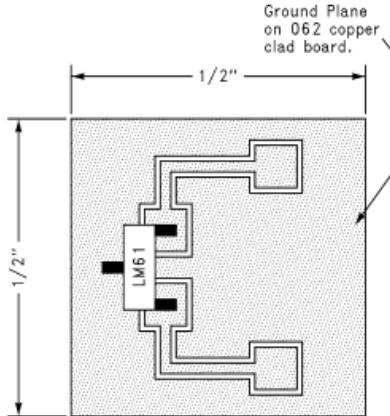


Figure 13.



A. $\frac{1}{2}"$ Square Printed Circuit Board with 2 oz. Copper Foil or Similar.

Figure 14. Printed Circuit Board Used for Heat Sink to Generate All Curves

APPLICATION INFORMATION

Mounting

The LM61 can be applied easily in the same way as other integrated-circuit temperature sensors. It can be glued or cemented to a surface. The temperature that the LM61 is sensing will be within about $+0.2^{\circ}\text{C}$ of the surface temperature that LM61's leads are attached to.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature measured would be at an intermediate temperature between the surface temperature and the air temperature.

To ensure good thermal conductivity the backside of the LM61 die is directly attached to the GND pin. The lands and traces to the LM61 will, of course, be part of the printed circuit board, which is the object whose temperature is being measured.

Alternatively, the LM61 can be mounted inside a sealed-end metal tube, and can then be dipped into a bath or screwed into a threaded hole in a tank. As with any IC, the LM61 and accompanying wiring and circuits must be kept insulated and dry, to avoid leakage and corrosion. This is especially true if the circuit may operate at cold temperatures where condensation can occur. Printed-circuit coatings and varnishes such as Humiseal and epoxy paints or dips are often used to ensure that moisture cannot corrode the LM61 or its connections.

The thermal resistance junction to ambient (θ_{JA}) is the parameter used to calculate the rise of a device junction temperature due to its power dissipation. For the LM61, [Equation 1](#) is used to calculate the rise in the die temperature.

$$T_J = T_A + \theta_{JA} [(+V_S I_Q) + (+V_S - V_O) I_L]$$

where

- I_Q is the quiescent current
 - I_L is the load current on the output
- (1)

Since the LM61's junction temperature is the actual temperature being measured care should be taken to minimize the load current that the LM61 is required to drive.

[Table 3](#) summarizes the rise in die temperature of the LM61 without any loading with a 3.3V supply, and the thermal resistance for different conditions.

Table 3. Temperature Rise of LM61 Due to Self-Heating and Thermal Resistance (θ_{JA})

	SOT-23 ⁽¹⁾		SOT-23 ⁽²⁾		TO-92 ⁽¹⁾		TO-92 ⁽³⁾	
	NO HEAT SINK		SMALL HEAT FIN		NO HEAT SINK		SMALL HEAT FIN	
	θ_{JA} (°C/W)	$T_J - T_A$ (°C)						
Still air	450	0.26	260	0.13	180	0.09	140	0.07
Moving air			180	0.09	90	0.05	70	0.03

(1) Part soldered to 30 gauge wire.

(2) Heat sink used is $\frac{1}{2}$ " square printed circuit board with 2 oz. foil with part attached as shown in [Figure 14](#).

(3) Part glued and leads soldered to 1" square of 1/16" printed circuit board with 2oz. foil or similar.

LM61



SNIS121I–JUNE 1999–REVISED FEBRUARY 2013

www.ti.com

Capacitive Loads

The LM61 handles capacitive loading well. Without any special precautions, the LM61 can drive any capacitive load as shown in [Figure 15](#). Over the specified temperature range the LM61 has a maximum output impedance of 5 kΩ. In an extremely noisy environment it may be necessary to add some filtering to minimize noise pickup. It is recommended that 0.1 μF be added from +V_S to GND to bypass the power supply voltage, as shown in [Figure 16](#). In a noisy environment it may be necessary to add a capacitor from the output to ground. A 1 μF output capacitor with the 5 kΩ maximum output impedance will form a 32 Hz lowpass filter. Since the thermal time constant of the LM61 is much slower than the 5 ms time constant formed by the RC, the overall response time of the LM61 will not be significantly affected. For much larger capacitors this additional time lag will increase the overall response time of the LM61.

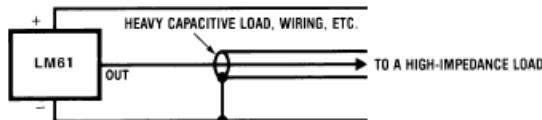


Figure 15. LM61 No Decoupling Required for Capacitive Load

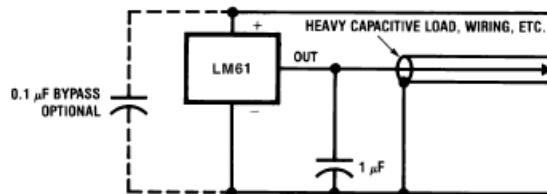


Figure 16. LM61 with Filter for Noisy Environment

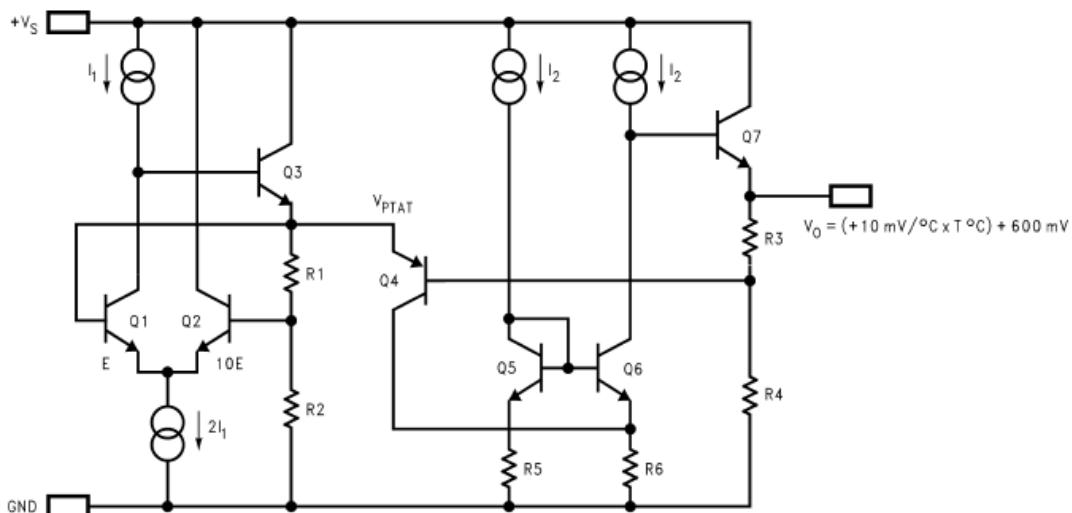


Figure 17. Simplified Schematic

Applications Circuits

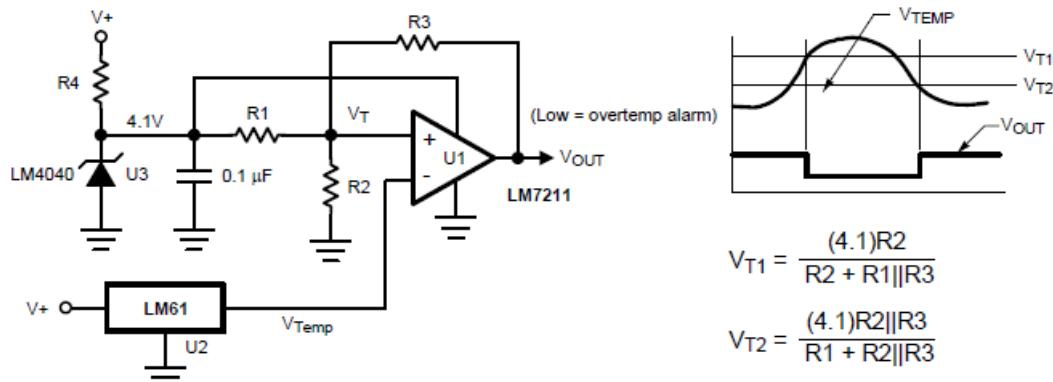


Figure 18. Centigrade Thermostat

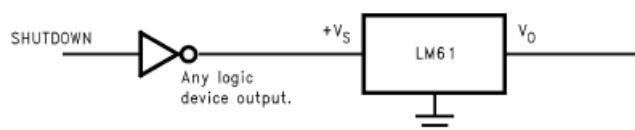
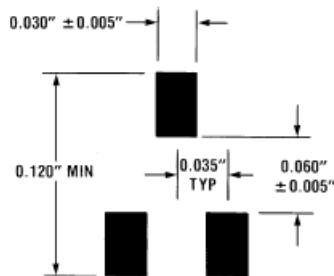


Figure 19. Conserving Power Dissipation with Shutdown

Recommended Solder Pads for SOT-23 Package



Anexo B.2: Hojas de especificaciones del sensor de presión MPX10DP

Pressure

Freescale Semiconductor

MPX10

Rev 14, 10/2008

**10 kPa Uncompensated
Silicon Pressure Sensors**

The MPX10 series silicon piezoresistive pressure sensors provide a very accurate and linear voltage output, directly proportional to the applied pressure. These standard, low cost, uncompensated sensors permit manufacturers to design and add their own external temperature compensation and signal conditioning networks. Compensation techniques are simplified because of the predictability of Freescale's single element strain gauge design.

Features

- Low Cost
- Patented Silicon Shear Stress Strain Gauge Design
- Ratiometric to Supply Voltage
- Differential and Gauge Options
- Durable Epoxy Unibody Element or Thermoplastic (PPS) Surface Mount Package

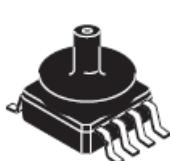
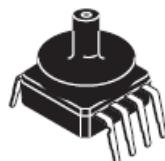
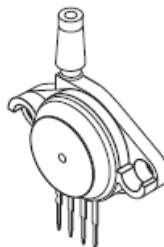
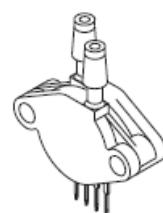
**MPX10
Series**

0 to 10 kPa (0 to 1.45 psi)
35 mV Full Scale Span
(Typical)

Application Examples

- Air Movement Control
- Environmental Control Systems
- Level Indicators
- Leak Detection
- Medical Instrumentation
- Industrial Controls
- Pneumatic Control Systems
- Robotics

ORDERING INFORMATION								
Device Name	Package Options	Case No.	# of Ports			Pressure Type		
			None	Single	Dual	Gauge	Differential	Absolute
Unibody Package (MPX10 Series)								
MPX10D	Tray	344	*				*	
MPX10DP	Tray	344C			*		*	
MPX10GP	Tray	344B		*		*		
Small Outline Package (MPXV10G Series)								
MPXV10GC6U	Rail	482A		*		*		
MPXV10GC7U	Rail	482C		*		*		

SMALL OUTLINE PACKAGEMPXV10GC6U
CASE 482A-01MPXV10GC7U
CASE 482C-03**UNIBODY PACKAGE**MPX10D
CASE 344-15MPX10GP
CASE 344B-01MPX10DP
CASE 344C-01

Pressure

Operating Characteristics

Table 1. Operating Characteristics ($V_S = 3.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P1 > P2$)

Characteristic	Symbol	Min	Typ	Max	Units
Differential Pressure Range ⁽¹⁾	P_{OP}	0	—	10	kPa
Supply Voltage ⁽²⁾	V_S	—	3.0	6.0	V _{dc}
Supply Current	I_O	—	6.0	—	mAdc
Full Scale Span ⁽³⁾	V_{FSS}	20	35	50	mV
Offset ⁽⁴⁾	V_{OFF}	0	20	35	mV
Sensitivity	$\Delta V/\Delta P$	—	3.5	—	mV/kPa
Linearity	—	-1.0	—	1.0	% V_{FSS}
Pressure Hysteresis (0 to 10 kPa)	—	—	±0.1	—	% V_{FSS}
Temperature Hysteresis	—	—	±0.5	—	% V_{FSS}
Temperature Coefficient of Full Scale Span	TCV_{FSS}	-0.22	—	-0.16	% $V_{FSS}/^\circ\text{C}$
Temperature Coefficient of Offset	TCV_{OFF}	—	±15	—	µV/°C
Temperature Coefficient of Resistance	TCR	0.21	—	0.27	% $Z_{IN}/^\circ\text{C}$
Input Impedance	Z_{IN}	400	—	550	Ω
Output Impedance	Z_{OUT}	750	—	1250	Ω
Response Time ⁽⁵⁾ (10% to 90%)	t_R	—	1.0	—	ms
Warm-Up Time ⁽⁶⁾	—	—	20	—	ms
Offset Stability ⁽⁷⁾	—	—	±0.5	—	% V_{FSS}

1. 1.0 kPa (kiloPascal) equals 0.145 psi.
2. Device is ratiometric within this specified excitation range. Operating the device above the specified excitation range may induce additional error due to device self-heating.
3. Full Scale Span (V_{FSS}) is defined as the algebraic difference between the output voltage at full rated pressure and the output voltage at the minimum rated pressure.
4. Offset (V_{OFF}) is defined as the output voltage at the minimum rated pressure.
5. Response Time is defined as the time for the incremental change in the output to go from 10% to 90% of its final value when subjected to a specified step change in pressure.
6. Warm-up Time is defined as the time required for the product to meet the specified output voltage after the pressure is stabilized.
7. Offset stability is the product's output deviation when subjected to 1000 hours of Pulsed Pressure, Temperature Cycling with Bias Test.

Maximum Ratings

Table 2. Maximum Ratings⁽¹⁾

Rating	Symbol	Value	Unit
Maximum Pressure ($P_1 > P_2$)	P_{MAX}	75	kPa
Burst Pressure ($P_1 > P_2$)	P_{BURST}	100	kPa
Storage Temperature	T_{STG}	-40 to +125	°C
Operating Temperature	T_A	-40 to +125	°C

1. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

Figure 1 shows a schematic of the internal circuitry on the stand-alone pressure sensor chip.

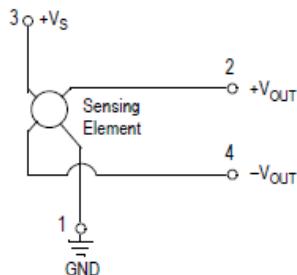


Figure 1. Uncompensated Pressure Sensor Schematic

Voltage Output versus Applied Differential Pressure

The output voltage of the differential or gauge sensor increases with increasing pressure applied to the pressure side (P_1) relative to the vacuum side (P_2). Similarly, output

voltage increases as increasing vacuum is applied to the vacuum side (P_2) relative to the pressure side (P_1).

MPX10

Pressure

Temperature Compensation

Figure 2 shows the typical output characteristics of the MPX10 series over temperature.

Because this strain gauge is an integral part of the silicon diaphragm, there are no temperature effects due to differences in the thermal expansion of the strain gauge and the diaphragm, as are often encountered in bonded strain gauge pressure sensors. However, the properties of the strain gauge itself are temperature dependent, requiring that the device be temperature compensated if it is to be used over an extensive temperature range.

Temperature compensation and offset calibration can be achieved rather simply with additional resistive components, or by designing your system using the MPX2010D series sensor.

LINEARITY

Linearity refers to how well a transducer's output follows the equation: $V_{out} = V_{off} + \text{sensitivity} \times P$ over the operating pressure range (Figure 3). There are two basic methods for calculating nonlinearity: 1) end point straight line fit or 2) a least squares best line fit. While a least squares fit gives the

"best case" linearity error (lower numerical value), the calculations required are burdensome.

Conversely, an end point fit will give the "worst case" error (often more desirable in error budget calculations) and the calculations are more straightforward for the user. Freescale's specified pressure sensor linearities are based on the end point straight line method measured at the midrange pressure.

Figure 4 illustrates the differential or gauge configuration in the basic chip carrier (Case 344). A silicone gel isolates the die surface and wire bonds from the environment, while allowing the pressure signal to be transmitted to the silicon diaphragm.

The MPX10 series pressure sensor operating characteristics and internal reliability and qualification tests are based on use of dry air as the pressure media. Media other than dry air may have adverse effects on sensor performance and long term reliability. Contact the factory for information regarding media compatibility in your application. Refer to application note AN3728, for more information regarding media compatibility.

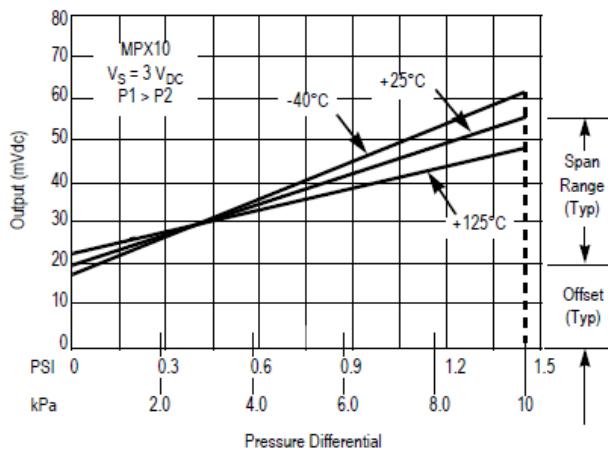


Figure 2. Output vs. Pressure Differential

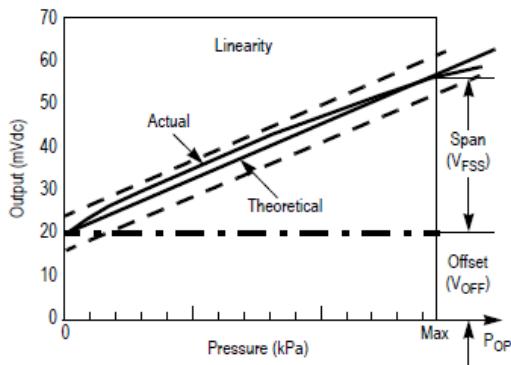


Figure 3. Linearity Specification Comparison

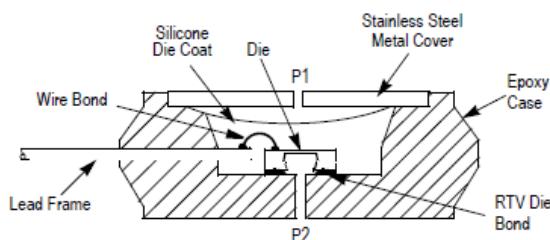


Figure 4. Unibody Package — Cross-Sectional Diagram
(Not to Scale)

Pressure**PRESSURE (P1)/VACUUM (P2) SIDE IDENTIFICATION TABLE**

Freescale designates the two sides of the pressure sensor as the Pressure (P1) side and the Vacuum (P2) side. The Pressure (P1) side is the side containing silicone gel which isolates the die from the environment. The pressure sensor is designed to operate with positive differential pressure applied, P1 > P2.

The Pressure (P1) side may be identified by using the following table.

Part Number	Case Type	Pressure (P1) Side Identifier
MPX10D	344	Stainless Steel Cap
MPX10DP	344C	Side with Part Marking
MPX10GP	344B	Side with Port Attached
MPXV10GC6U	482A	Side with Port Attached
MPXV10GC7U	482C	Side with Port Attached

MPX10

Anexo B.3: Hojas de especificaciones del sensor de humedad HMZ-433A1

PRODUCT SPECIFICATIONS

HMZ-433A1

HMZ-433A1 Humidity module specifications

1. Application range :

The HMZ-433A1 module consist of an HCZ sensor and integrated circuit to provide a linear DC Voltage for 0-100%RH to enable easy user application of the HCZ sensor. They are specifically designed for use appliances and controllers.

Feature	Application
Wide humidity operation range	Air condition ,humidifier, Dehumidifier.
Linear DC Output	Humidity controller, Humidity transmitter.
Easy operation	Hygrometer, Hygro-recorder.
Long-term stability	Copying machine.
Small and economical	Clock ,Weather-forecast barometer.

2. Electrical characteristics :

2.1 Sensing element (Humidity) : HMZ-433A1 Humidity sensor " GHITRON HCZ sensor"

2.2 Supply Voltage(Vin) : 5VDC±5%

2.3 Current Consumption : 5mA max:(2mA avg.)

2.4 Operating Range

Temperature : 0 to 60°C

Humidity : 95%RH or less

2.5 Storage

Temperature : -20 to 70°C

Humidity : 95%RH or less

2.6 Humidity transmitting

Range : 20 to 90%RH

2.7 Accuracy :

Temperature : Resistance(1%) : 50±0.5KΩ(at 25±0.2°C)

Humidity : ±5%RH (at 25°C, 50%RH, Vin=5.00VDC)

2.8 Humidity Output 0~3.3V

At 25° ;Vin=5.00VDC

Signal(Reference) : (Output Impedance approx:5KΩ)

Humidity(%RH)	20	30	40	50	60	70	80	90
Output Voltage(V)	0.66	0.99	1.32	1.65	1.98	2.31	2.64	2.97

2.9 Temperature Output

Using thermistor 503 R(25°C)=50KΩ±1%,

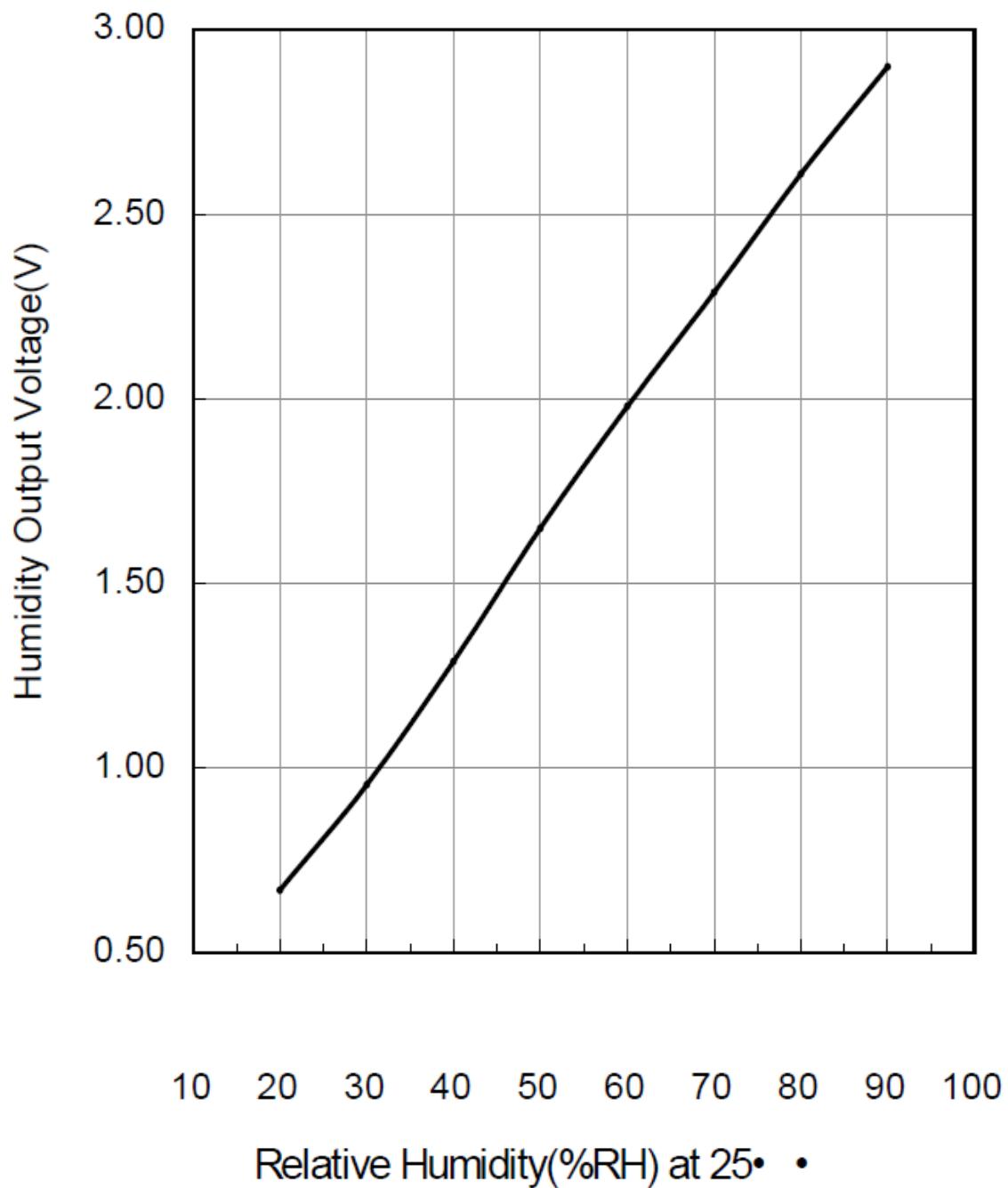
Signal(Reference) : B(25/85)=3950K±1%

PRODUCT SPECIFICATIONS

HMZ-433A1

Temperature(°C)	0	10	20	25	30	40	50	60
Resistance(KΩ)	160.56	98.714	62.328	50.0	40.356	26.756	18.138	12.554

2.10 Humidity Output 0~3.3V characteristics :



PRODUCT SPECIFICATIONS

HMZ-433A1

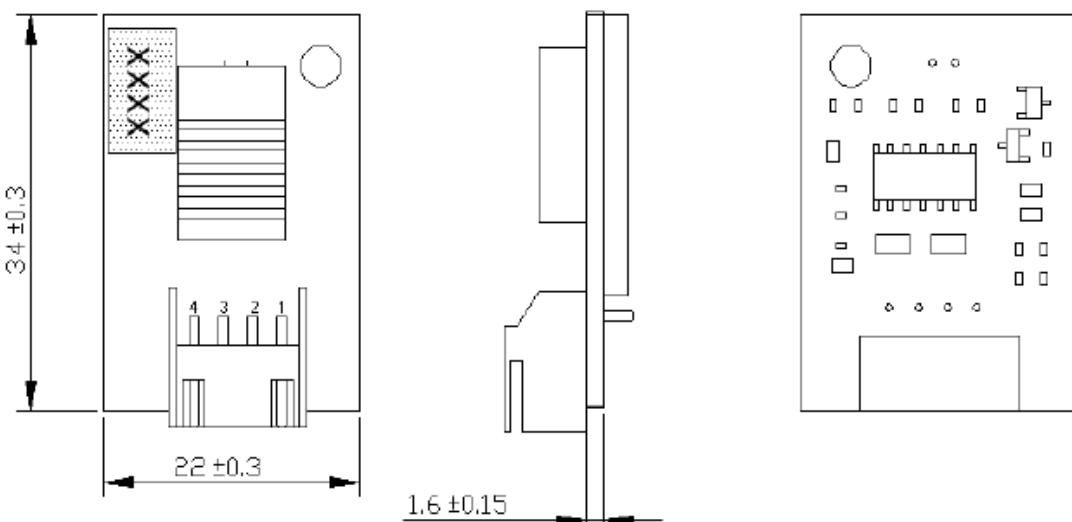
2.11 HMZ module Humidity Output 0~3.3V v.s Temperature characteristics :

	10° •	15° •	20° •	25° •	30° •	35° •	40° •
20%RH	0.75	0.72	0.69	0.66	0.62	0.59	0.55
30%RH	1.03	1.00	1.00	0.99	0.96	0.93	0.90
40%RH	1.32	1.30	1.31	1.32	1.30	1.28	1.25
50%RH	1.64	1.63	1.64	1.65	1.64	1.63	1.61
60%RH	1.97	1.97	1.98	1.98	1.98	1.98	1.96
70%RH	2.30	2.30	2.31	2.31	2.31	2.31	2.30
80%RH	2.64	2.64	2.63	2.64	2.63	2.63	2.61
90%RH	2.97	2.96	2.94	2.97	2.94	2.92	2.90

Remark• Accuracy• ±5%RH(at 25° • 60%RH• Vin=5.00VDC)

Output range• 1.815~2.145V(55%RH~65%RH Output)

3. Configuration & Dimensions (Units :mm) : Model No. HMZ-433A1 Drawing



3.1 Terminal Connection :

Terminal	Content. (4Pin Pitch : 2.5mm)
Pin1	Power Source 5V DC.
Pin2	Humidity Output
Pin3	GND
Pin4	Temp Output 50kΩ(at 25°C)

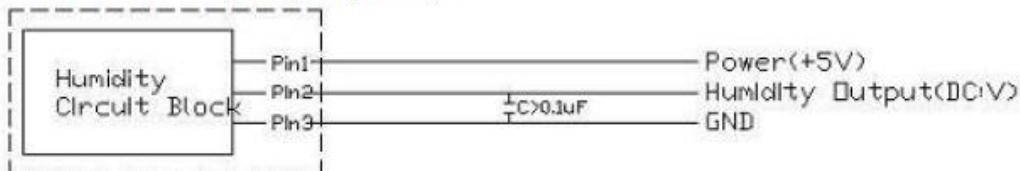
Remark : Please pay attention to the power source and GND polarity position.

PRODUCT SPECIFICATIONS

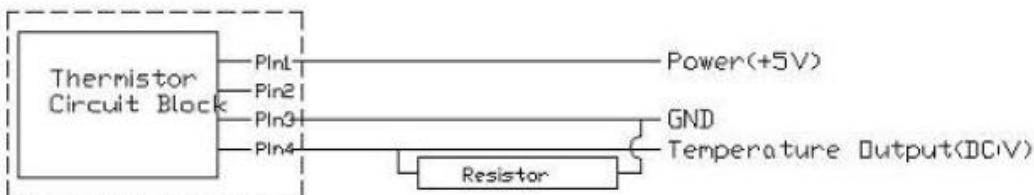
HMZ-433A1

4. Typical Applications :

4.1 Pin2 Humidity output (DC:V)



4.2 Pin4 Thermistor output (DC:V)



5. Order Form :

HMZ—433A1 is the module with 2317RJ-04 4 pins(pitch=2.5 mm) of terminal Connector of voltage output for humidity and temperature output 50KΩ(at 25°C) .

HMZ-433A1 Configure of definition.	
Input Voltage	5V
Humidity Output Voltage	0~3.3V
Temperature Output	50kΩ(at 25°C) . (*Note1*)
Terminal Connector	2317RJ-04 , 4pins , (Pitch=2.5 mm) .
Accuracy	±5%RH(at 25°C, 60%RH, Vin=5.00VDC)
Sensing element (Humidity)• •	Humidity sensor " GHITRON HCZ sensor"
Output Range• •	20~90%RH

(*Note1*) Optional type

Part_No.	NTC Configure of definition.
HMZ-433A1	50kΩ , SMD (Default type)
HMZ-433A1N	NA
HMZ-433A1D	50kΩ , DIP

6. Reliability test :

No.	ITEM	METHOD	REQUIREMENT
6.1	Impact test	To drop Module 3time at random on to a hard wooden plate from 1meter above high.	No breakage, nor cracks. Should be electrically normal
6.2	Vibration test	Vibration test in X-Y-Z axis for 30min.under 10-55Hz frequency, 1.5mm(10-55-10Hz) amplitude.	No breakage, nor cracks. Should be electrically normal

PRODUCT SPECIFICATIONS

HMZ-433A1

No.	ITEM	METHOD	REQUIREMENT
6.3	Humidity resistance	1000 hours@ 60° ; 90%RH .	△%RH < ±5%RH
6.4	Cool resistance	1000 hours@ -30° .	△%RH < ±5%RH
6-5	Temperature cycle test	Repeat 100 cycles , Each cycle: 30 minutes@-30° . 30 minutes@85° .	△%RH < ±5%RH
6.6	Loading test	Room Temperature / Humidity , Input +5V. for 1000hours.	△%RH < ±5%RH

Remarks :

- (1) All standard figures are based on humidity variation under 60%RH(25°C)
- (2) Upon completion of all tests, The module will be left over under nominal environment and humidity for 24hours.

7. Packing :

There are 50 pieces of modules to be packed in one tray.

8. Caution remarks on operation :

- 8.1 To avoid direct application of DC voltage on humidity sensor.
- 8.2 To protect sensor from dewfall and drenching.
- 8.3 To avoid and operation of humidity sensors in the following environmental ambient.

8.3 To avoid and operation of humidity sensors in the following environmental ambient.

8.3.1 Salt

8.3.2 Inorganic gas Sulfide dioxide, Chlorine, Ammonium, etc.

8.3.3 Organic gas Alcoholic, Glycols, Aldehydes, etc.

8.4 Recommendable storage condition :

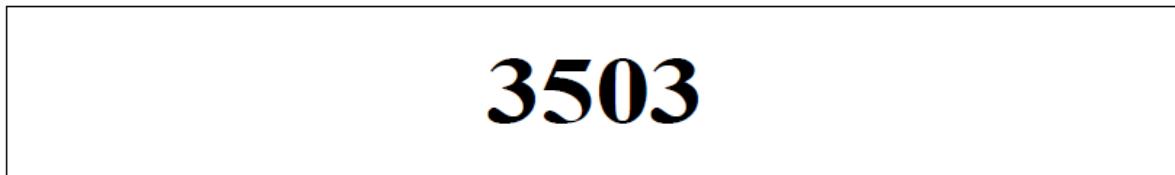
Temperature range 10~40° .

Humidity Range 90%RH or Less.

8.5 Do not store humidity sensors long period of time in an 60° ambient, due to some occasion of degradation on sensor housing case.

Remark : We have the right to revise specification and product configurations without notice.

Anexo B.4: Hojas de especificaciones del sensor de efecto Hall UGN3503



Data Sheet
27501B†

RATIOMETRIC, LINEAR HALL-EFFECT SENSORS

The diagram shows a block diagram of the UGN3503 internal circuit. It consists of a Hall sensing element (represented by an 'X') connected to a linear amplifier, which then drives an emitter-follower output stage. The output is shown as a square wave pulse. Below the block diagram, the package pinout is shown with three pins labeled 1, 2, and 3. Pin 1 is labeled 'SUPPLY' (V_{CC}), Pin 2 is labeled 'GROUND', and Pin 3 is labeled 'OUTPUT'. A reference drawing number 'Dwg. PH-006' is at the bottom.

Pinning is shown viewed from branded side.

ABSOLUTE MAXIMUM RATINGS

Supply Voltage, V _{CC}	8 V
Magnetic Flux Density, B	Unlimited
Operating Temperature Range, T _A	-20°C to +85°C
Storage Temperature Range, T _S	-65°C to +150°C

The UGN3503LT, UGN3503U, and UGN3503UA Hall-effect sensors accurately track extremely small changes in magnetic flux density—changes generally too small to operate Hall-effect switches.

As motion detectors, gear tooth sensors, and proximity detectors, they are magnetically driven mirrors of mechanical events. As sensitive monitors of electromagnets, they can effectively measure a system's performance with negligible system loading while providing isolation from contaminated and electrically noisy environments.

Each Hall-effect integrated circuit includes a Hall sensing element, linear amplifier, and emitter-follower output stage. Problems associated with handling tiny analog signals are minimized by having the Hall cell and amplifier on a single chip.

Three package styles provide a magnetically optimized package for most applications. Package suffix 'LT' is a miniature SOT-89/TO-243AA transistor package for surface-mount applications; suffix 'U' is a miniature three-lead plastic SIP, while 'UA' is a three-lead ultra-mini-SIP. All devices are rated for continuous operation over the temperature range of -20°C to +85°C.

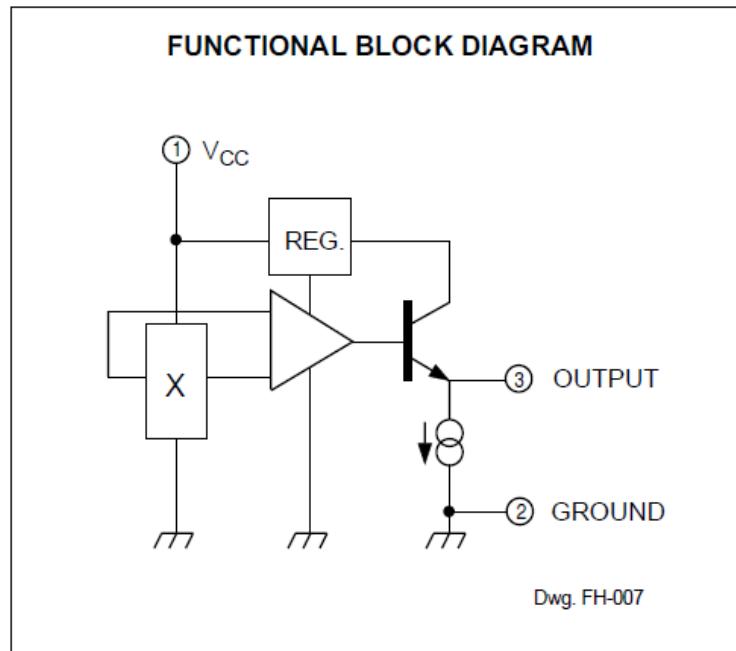
FEATURES

- Extremely Sensitive
- Flat Response to 23 kHz
- Low-Noise Output
- 4.5 V to 6 V Operation
- Magnetically Optimized Package

Always order by complete part number, e.g., **UGN3503UA**.



3503
RATIOMETRIC,
LINEAR
HALL-EFFECT SENSORS



ELECTRICAL CHARACTERISTICS at $T_A = +25^\circ\text{C}$, $V_{CC} = 5 \text{ V}$

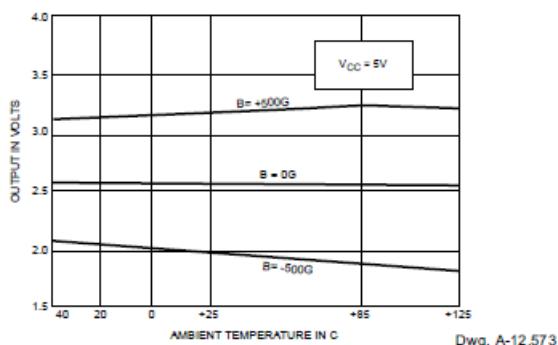
Characteristic	Symbol	Test Conditions	Limits			
			Min.	Typ.	Max.	Units
Operating Voltage	V_{CC}		4.5	—	6.0	V
Supply Current	I_{CC}		—	9.0	13	mA
Quiescent Output Voltage	V_{OUT}	$B = 0 \text{ G}$	2.25	2.50	2.75	V
Sensitivity	ΔV_{OUT}	$B = 0 \text{ G to } \pm 900 \text{ G}$	0.75	1.30	1.75	mV/G
Bandwidth (-3 dB)	BW		—	23	—	kHz
Broadband Output Noise	V_{out}	$BW = 10 \text{ Hz to } 10 \text{ kHz}$	—	90	—	μV
Output Resistance	R_{OUT}		—	50	220	Ω

All output-voltage measurements are made with a voltmeter having an input impedance of at least $10 \text{ k}\Omega$.

Magnetic flux density is measured at most sensitive area of device located $0.0165''$ (0.42 mm) below the branded face of the 'U' package; $0.0195''$ (0.50 mm) below the branded face of the 'UA' package; and $0.0305''$ (0.775 mm) below the branded face of the 'LT' package.

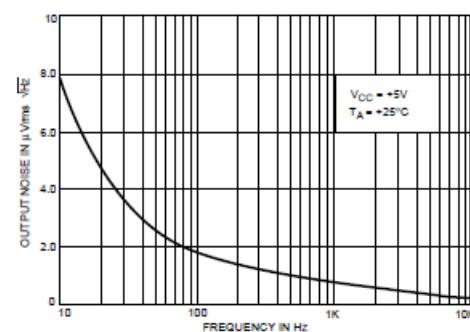
**3503
RATIOMETRIC,
LINEAR
HALL-EFFECT SENSORS**

**OUTPUT VOLTAGE AS A
FUNCTION OF TEMPERATURE**



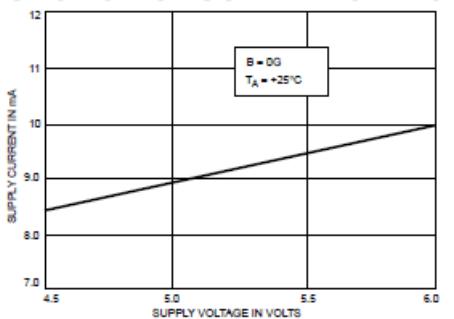
Dwg. A-12,573

**OUTPUT NOISE AS A
FUNCTION OF FREQUENCY**



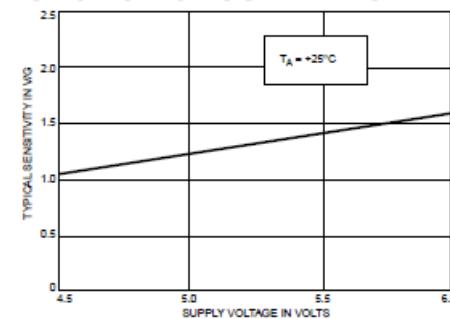
Dwg. A-12,505

**SUPPLY CURRENT AS A
FUNCTION OF SUPPLY VOLTAGE**



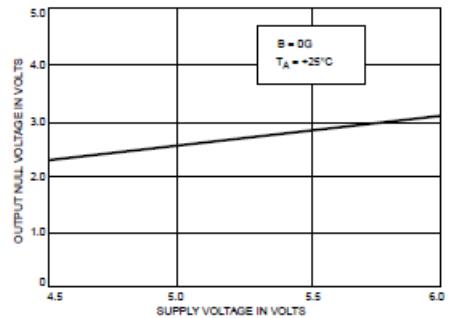
Dwg. A-12,506

**DEVICE SENSITIVITY AS A
FUNCTION OF SUPPLY VOLTAGE**



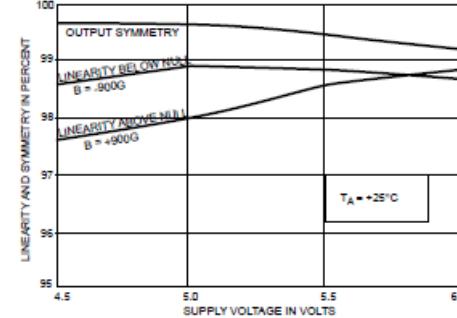
Dwg. A-12,507

**OUTPUT NULL VOLTAGE AS A
FUNCTION OF SUPPLY VOLTAGE**



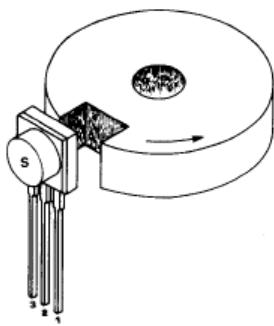
Dwg. A-12,508

**LINEARITY AND SYMMETRY AS A
FUNCTION OF SUPPLY VOLTAGE**

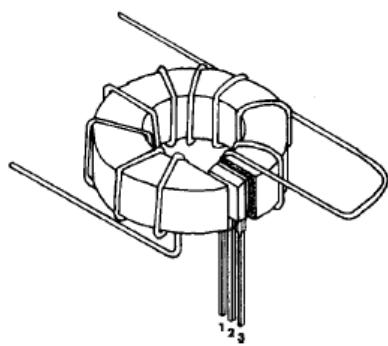


Dwg. A-12,509

3503
RATIOMETRIC,
LINEAR
HALL-EFFECT SENSORS

NOTCH SENSOR

Dwg. A-12,574



Dwg. A-12,513

OPERATION

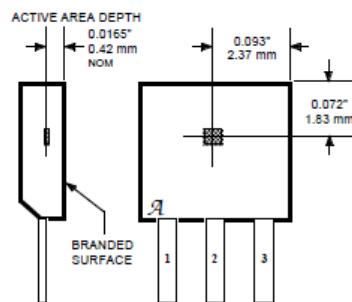
The output null voltage ($B = 0$ G) is nominally one-half the supply voltage. A south magnetic pole, presented to the branded face of the Hall-effect sensor will drive the output higher than the null voltage level. A north magnetic pole will drive the output below the null level.

In operation, instantaneous and proportional output-voltage levels are dependent on magnetic flux density at the most sensitive area of the device. Greatest sensitivity is obtained with a supply voltage of 6 V, but at the cost of increased supply current and a slight loss of output symmetry. The sensor's output is usually capacitively coupled to an amplifier that boosts the output above the millivolt level.

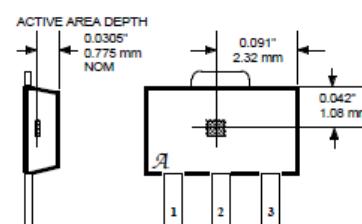
In two applications shown, a permanent bias magnet is attached with epoxy glue to the back of the epoxy package. The presence of ferrous material at the face of the package acts as a flux concentrator.

The south pole of a magnet is attached to the back of the package if the Hall-effect IC is to sense the presence of ferrous material. The north pole of a magnet is attached to the back surface if the integrated circuit is to sense the absence of ferrous material.

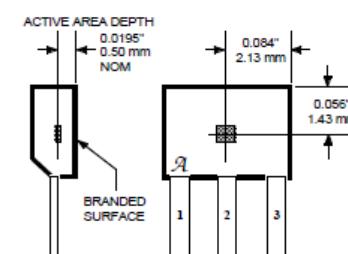
Calibrated linear Hall devices, which can be used to determine the actual flux density presented to the sensor in a particular application, are available.

SENSOR LOCATIONS**SUFFIX "U"**

Dwg. MH-002-SD

SUFFIX "LT"

Dwg. MH-008-5A

SUFFIX "UA"

Dwg. MH-011-3D



115 Northeast Cutoff, Box 15036
 Worcester, Massachusetts 01615-0036 (508) 853-5000

Anexo C

Anexo C.1: Código del programa principal del microcontrolador

```
int main(void)
{
/*Se declaran variables: La primera es un vector tipo estático entero de 16 bits "uin16_t", el cual se utiliza para guardar 300 datos de 16 bits del canal 4, 5, ó 6, que contendrán los valores de la señal de externa, para ser enviados hacia la PC, esto con el fin de reconstruir la señal en la aplicación de graficación. La segunda variable es de tipo char llamada "Dato", la cual obtendrá el carácter enviado por la PC, para iniciar la medición del canal deseado. La tercer variable es simplemente una auxiliar, y las últimas tres variables, contendrán el valor de la frecuencia según el rango de voltaje deseado. */
static uint16_t med[300];
char Dato;
int k=0;
uint16_t frecuencia, frecuencia2, frecuencia3;
/* Se inicia el ciclo infinito que requiere el microcontrolador para trabajar*/
while(1)
{
    /*Se asigna el carácter enviado por el puerto serial de la aplicación, la función uart_getchar(TERM_PORT_NUM) la crea por default el programa en el apartado de Events.c*/
    dato = uart_getchar(TERM_PORT_NUM);
    /* Se inicia una sentencia switch-case para la elección del canal a leer*/
    switch(dato)
    {
        /* El primer caso 'T' indica que la PC envió una letra T para poder recibir los datos del sensor de temperatura el cual se encuentra conectado en el canal 0 del ADC*/
        case 'T':
            /*Adquisición y ordenamiento de menor a mayor de 50 muestras del sensor de temperatura*/
            for(k=0;k<50;k++)
            {
                AD1_MeasureChan(1,0);
                AD1_GetChanValue(0,&med[k]);
                if(k==0)
                    sen[k]=med[k];
                else
                {
                    if(med[k]<=sen[0])
                    {
                        for(i=k;i>0;i--)
                            sen[i]=sen[i-1];
                        sen[0]=med[k];
                    }
                    else
                    {
                        if(med[k]>=sen[k-1])
                            sen[k]=med[k];
                        else
                        {
                            for(i=0;i<k;i++)
                            {
                                if(med[k]>=sen[i]&&med[k]<=sen[i+1])
                                {
```

```

        for(j=k;j>i+1;j--)
            sen[j]=sen[j-1];

            sen[i+1]=med[k];
            i=k;
        }
    }
}
}

/* Promedio de las muestras 10 mas cercanas a la media*/
prom=(sen[19]+sen[20]+sen[21]+sen[22]+sen[23]+sen[24]+sen[25]+sen[26]+sen[27]+sen[28])/10;
printf("%d\n",prom);
break;
case 'H':
for(k=0;k<50;k++)
{
    AD1_MeasureChan(1,1);
    AD1_GetChanValue(1,&med[k]);
    if(k==0)
        sen[k]=med[k];
    else
    {
        if(med[k]<=sen[0])
        {
            for(i=k;i>0;i--)
                sen[i]=sen[i-1];
            sen[0]=med[k];
        }
        else
        {
            if(med[k]>=sen[k-1])
                sen[k]=med[k];
            else
            {
                for(i=0;i<k;i++)
                {
                    if(med[k]>=sen[i]&&med[k]<=sen[i+1])
                    {
                        for(j=k;j>i+1;j--)
                            sen[j]=sen[j-1];
                        sen[i+1]=med[k];
                        i=k;
                    }
                }
            }
        }
    }
}
}

prom=(sen[19]+sen[20]+sen[21]+sen[22]+sen[23]+sen[24]+sen[25]+sen[26]+sen[27]+sen[28])/10;

```

```
        printf("%d\n",prom);
        break;
    case 'M':
        for(k=0;k<50;k++)
        {
            AD1_MeasureChan(1,2);
            AD1_GetChanValue(2,&med[k]);
            if(k==0)
                sen[k]=med[k];
            else
            {
                if(med[k]<=sen[0])
                {
                    for(i=k;i>0;i--)
                        sen[i]=sen[i-1];
                    sen[0]=med[k];
                }
                else
                {
                    if(med[k]>=sen[k-1])
                        sen[k]=med[k];
                    else
                    {
                        for(i=0;i<k;i++)
                        {
                            if(med[k]>=sen[i]&&med[k]<=sen[i+1])
                            {
                                for(j=k;j>i+1;j--)
                                    sen[j]=sen[j-1];
                                sen[i+1]=med[k];
                                i=k;
                            }
                        }
                    }
                }
            }
        }
        prom=(sen[19]+sen[20]+sen[21]+sen[22]+sen[23]+sen[24]+sen[25]+sen[26]+sen[27]+sen[28])/10;
        printf("%d\n",prom);
        break;
    case 'P':
        for(k=0;k<50;k++)
        {
            AD1_MeasureChan(1,3);
            AD1_GetChanValue(3,&med[k]);
            if(k==0)
                sen[k]=med[k];
            else
            {
                if(med[k]<=sen[0])
                {
```

```

        for(i=k;j>0;i--)
            sen[i]=sen[i-1];
        sen[0]=med[k];
    }
    else
    {
        if(med[k]>=sen[k-1])
            sen[k]=med[k];
        else
        {
            for(i=0;i<k;i++)
            {
                if(med[k]>=sen[i]&&med[k]<=sen[i+1])
                {
                    for(j=k;j>i+1;j--)
                        sen[j]=sen[j-1];
                    sen[i+1]=med[k];
                    i=k;
                }
            }
        }
    }
}
prom=(sen[19]+sen[20]+sen[21]+sen[22]+sen[23]+sen[24]+sen[25]+sen[26]+sen[27]+sen[28])/10;
printf("%d\n",prom);
break;
case 'A':
/* Obtención de frecuencia de la señal de voltajes altos*/
frecuencia=frec();
printf("%d\n",frecuencia);
break;
case 'B':
/* Obtención de frecuencia de la señal de voltajes medios*/
frecuencia2=frec2();
printf("%d\n",frecuencia2);
break;
case 'C':
/* Obtención de frecuencia de la señal de voltajes bajos*/
frecuencia3=frec3();
printf("%d\n",frecuencia3);
break;
case 'D':
/* Obtención de muestras para señal de voltajes altos a frecuencias menores de 150KHz */
for(k=0;k<300;k++)
{
    AD1_MeasureChan(1,4);
    AD1_GetChanValue(4,&med[k]);
    printf("%d\n",med[k]);
}
break;

```

```
        case 'E':
/* Obtención de muestras para señal de voltajes medios a frecuencias menores de 150KH*/
    for(k=0;k<300;k++)
    {
        AD1_MeasureChan(1,5);
        AD1_GetChanValue(5,&med[k]);
        printf("%d\n",med[k]);
    }
    break;
case 'F':
/* Obtención de muestras para señal de voltajes bajos a frecuencias menores de 150KHz */
/*
    for(k=0;k<300;k++)
    {
        AD1_MeasureChan(1,6);
        AD1_GetChanValue(6,&med[k]);
        printf("%d\n",med[k]);
    }
    break;
case 'G':
/* Obtención de muestras para señal de voltajes altos a frecuencias mayores de 150KHz */
    for(k=0;k<300;k++)
    {
        AD1_MeasureChan(1,4);
        AD1_GetChanValue(4,&med[k]);
    }
    for(k=0;k<300;k++)
        printf("%d\n",med[k]);
    break;
case 'I':
/* Obtención de muestras para señal de voltajes medios a frecuencias mayores de 150KHz */
    for(k=0;k<300;k++)
    {
        AD1_MeasureChan(1,5);
        AD1_GetChanValue(5,&med[k]);
    }
    for(k=0;k<300;k++)
        printf("%d\n",med[k]);
    break;
case 'J':
/* Obtención de muestras para señal de voltajes bajos a frecuencias mayores de 150KHz */
    for(k=0;k<300;k++)
    {
        AD1_MeasureChan(1,6);
        AD1_GetChanValue(6,&med[k]);
    }
    for(k=0;k<300;k++)
        printf("%d\n",med[k]);
    break;
}
}/*Fin del ciclo while*/
```

Anexo C.2: Código fuente de los métodos de los subsistemas del microcontrolador

```
/* Declaración de variables */
extern volatile bool DataReceivedFlg;
static uint16_t cont, cont2, cont3, fre, fre2, fre3;

/* Este método se manda a llamar cada que un bloque de información es recibido por la terminal, por el serial, pone a uno la bandera de dato recibido*/
void IO1_OnBlockReceived(LDD_TUserData *UserDataPtr)
{
    DataReceivedFlg = TRUE; /* Set DataReceivedFlg flag */
}

/*Esta función obtiene el carácter introducido por el usuario en la terminal) */
char uart_getchar (int uart_port_num)
{
    if (uart_port_num == 0)
    {
        while (!(UART0_S1 & UART_S1_RDRF_MASK));
        /* Se retorna el carácter recibido */
        return (uint8)UART0_D;
    }
    else if (uart_port_num == 1)
    {
        while (!(UART1_S1 & UART_S1_RDRF_MASK));
        /* Se retorna el carácter recibido */
        return (uint8)UART1_D;
    }
    else
    {
        while (!(UART2_S1 & UART_S1_RDRF_MASK));
        /* Se retorna el carácter recibido */
        return (uint8)UART2_D;
    }
}

// Esta función regresa el valor de la frecuencia de la señal de voltajes entre 20V – 120V.
int freq (void)
{
    return fre;
}

// Esta función regresa el valor de la frecuencia de la señal de voltajes entre 3V y 20V.
int freq2 (void)
{
    return fre2;
}
```

```
// Esta función regresa el valor de la frecuencia de la señal de voltajes entre 3V y 20V.  
int freq3 (void)  
{  
    return fre3;  
}  
  
/* Este método es mandado llamar cada vez que se detecta un flanco de subida de la señal 1*/  
void EInt1_OnInterrupt(LDD_TUserData *UserDataPtr)  
{  
    cont++;  
}  
  
/* Este método es mandado llamar cada vez que se detecta un flanco de subida de la señal 2.*/  
void EInt2_OnInterrupt(LDD_TUserData *UserDataPtr)  
{  
    cont2++;  
}  
  
/* Este método es mandado llamar cada vez que se detecta un flanco de subida de la señal 2.*/  
void EInt3_OnInterrupt(LDD_TUserData *UserDataPtr)  
{  
    cont3++;  
}  
  
/* Este método es llamado cada que la cuenta del temporizador completa el tiempo indicado, en este caso 1 seg. */  
void TI1_OnInterrupt(LDD_TUserData *UserDataPtr)  
{  
    // Se respalda el valor de la frecuencia antes de resetear los contadores.  
    fre=cont;  
    fre2=cont2;  
    fre3=cont3;  
    /* Se reinician los contadores que están leyendo el número de pulsos por segundo que ambas señales externas  
    generan. */  
    cont=0;  
    cont2=0;  
    cont3=0;  
}
```

Anexo C.3: Programa de la primera ventana de la aplicación a PC

```
//Esta librería se agrega para poder manejar un archivo de datos.  
using System.IO;  
//Esta librería se agrega para poder obtener el nombre de los puertos conectados a la PC.  
using System.IO.Ports;  
  
//Estas librerías las crea el programa por default.  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
//Clases creadas por default por el software  
namespace Proyecto  
{  
    public partial class ventana1 : Form  
    {  
        public ventana1()  
        {  
            //Inicialización de los componentes.  
            InitializeComponent();  
            //Agregado de las leyendas que tendrá el ComboBox cuando el usuario despliegue el contenido.  
            comboBox1.Text = "-Selecciona Puerto-";  
            //Busqueda e impresión en el ComboBox de los puertos posibles a seleccionar.  
            string[] ports = SerialPort.GetPortNames();  
            foreach (string s in ports)  
            {  
                comboBox1.Items.Add(s);  
            }  
        }  
  
        //Evento del botón con la leyenda “Continuar”, ejecutado cada que el usuario lo oprima.  
        private void button1_Click(object sender, EventArgs e)  
        {  
            //Dar nombre del puerto serial que el usuario haya seleccionado en el ComboBox al SerialPort.  
            serialPort1.PortName = comboBox1.Text;  
            //Probar que el puerto abra correctamente.  
            try  
            {  
                serialPort1.Open();  
                /*Si abrió, generar un objeto que no ayude a guardar como dato el nombre del puerto, que abra un  
                archivo de datos*/  
                StreamWriter com = new StreamWriter("COM.mat");  
                //Guardar el nombre del puerto.  
                com.WriteLine(serialPort1.PortName);  
            }  
        }  
}
```

```
    //Cerrar archivo de datos.  
    com.Close();  
    //Cerrar comunicación serial.  
    serialPort1.Close();  
    //Abrir ventana de selección de magnitud a analizar.  
    System.Diagnostics.Process.Start("pictures.exe");  
    //Cerrar ventana de elección de puerto serial COM.  
    Close();  
}  
//No fue posible abrir el puerto serial, mostrar un mensaje de error al usuario.  
catch  
{  
    MessageBox.Show("Selección incorrecta de puerto, verifica la conexión y vuelve a intentar");  
}  
}  
}  
}
```

Anexo C.4: Programa de la segunda ventana de la aplicación a PC

```
//Estas librerías las crea el programa por default.  
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
//Clase creada por el software por default.  
namespace pictures  
{  
    public partial class Form1 : Form  
    {  
        //Método creado por default para la inicialización de la ventana.  
        public Form1()  
        {  
            InitializeComponent();  
        }  
  
        //Método para la opresión de la imagen referente al análisis de la temperatura  
        private void pictureBox1_Click(object sender, EventArgs e)  
        {  
            //Se abre tercera ventana, de graficación de magnitud seleccionada  
            System.Diagnostics.Process.Start("grafica temperatura.exe");  
            /*Se cierra la segunda ventana, de selección de magnitud a analizar  
            Close();  
        }  
  
        //Método para la opresión de la imagen referente al análisis de la humedad  
        private void pictureBox2_Click(object sender, EventArgs e)  
        {  
            System.Diagnostics.Process.Start("grafica humedad.exe");  
            Close();  
        }  
  
        //Método para la opresión de la imagen referente al análisis de efecto Hall  
        private void pictureBox3_Click(object sender, EventArgs e)  
        {  
            System.Diagnostics.Process.Start("grafica Hall.exe");  
            Close();  
        }  
}
```

```
//Método para la opresión de la imagen referente al análisis de la presión
private void pictureBox4_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("grafica preión.exe");
    Close();
}

//Método para la opresión de la imagen referente al análisis del voltaje
private void pictureBox5_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Osciloscopio.exe");
    Close();
}

//Método para la opresión de la imagen referente al generador de funciones
private void pictureBox6_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("Gen de funciones.exe");
    Close();
}
```

Anexo C.5: Programa de ventana de graficación, para los sensores

```
//Librerías agregadas por default por el software
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
//Librerías extra agregada, para hacer uso del archivo de datos en donde se guarda el nombre del puerto serial a
//usar.
using System.IO;

//Clase creada por el software por default.
namespace grafica_temperatura
{
    public partial class Temp : Form
    {
        //Declaración de variables
        int ci = 1, fi = 1, c, f, cant, fant, cont, cint, fint, cf, ff;
        double x, y, xi, yi = -40, yf = 100, yant, xff, A = 1;
        double xf;

        //Inicialización de la ventana.
        public Temp()
        {
            InitializeComponent();
            StreamReader serial = new StreamReader("COM.mat");
            serialPort1.PortName = serial.ReadLine(); //lectura del archivo de datos, para saber que COM se eligió.
            serial.Close();
            serialPort1.Open(); //apertura del puerto serial para la comunicación con el microcontrolador.
            cxf.Text = "10"; //número de mediciones que mostrará por default la aplicación creada.
        }

        //método para el botón de inicio, paro de graficación.
        private void button1_Click(object sender, EventArgs e)
        {
            if (button1.Text=="Stop")
            {
                //Si el botón se presionó cuando la graficación estaba corriendo:
                button2.Visible = true; //Se pone visible el botón de regreso a la ventana de selección de magnitud
                button3.Visible = true; //Se pone visible el botón de reanudación de graficación
                button1.Text = "Inicia"; //El texto del botón cambia a "Inicia"
                timer1.Enabled = false; //Desactivación de timer
                cxf.Enabled = true;
            }
        }
}
```

```
else
{
    //Si el botón se presionó cuando la graficación estaba frenada:
    try
    {
        //Se pregunta si el número de muestras no rebasa el número de pixeles en el ancho del PictureBox
        if (double.Parse(cxf.Text) <= 0 | double.Parse(cxf.Text) >= 300)
            //Si sí se rebasa, enviar un mensaje de error
            MessageBox.Show("El número de muestras debe ser mayor a 0 y menor a 300");
        else
        {
            //No se rebasa, entonces, se inicializan las variables a usar en la graficación, se pone la leyenda
            "Stop" a este botón, se activa el timer, se ocultan los botones de reanudación y de regreso a la
            ventana anterior, y se limpia en PictureBox.
            xf = double.Parse(cxf.Text) / 10;
            cont = 0;
            xi = 0.0;
            xff = xf;
            cf = pic1.Width;
            ff = pic1.Height;
            button2.Visible = false;
            button1.Text = "Stop";
            button3.Visible = false;
            timer1.Enabled = true;
            cxf.Enabled = false;
            pic1.Refresh();
        }
    }
    catch (FormatException)
    {
        //Si el usuario introdujo letras o caracteres especiales, en el cuadro que pide el número de muestras por
        PictureBox, se manda un mensaje de verificación.
        MessageBox.Show("Se han introducido caracteres inválidos, verifica el número de muestras");
    }
}

//Evento del timer
private void timer1_Tick(object sender, EventArgs e)
{
    serialPort1.WriteLine("T"); //Envío del carácter correspondiente a la magnitud seleccionada
    Graphics g1 = pic1.CreateGraphics();
    Graficador(g1); //Proceso de graficación
}

//Método para dibujar una cuadricula en el PictureBox
public void cuadricula(Graphics g1)
{
    int k, r=0;
    fint = ff / 14;
```

```
cint = cf / (int)((xff) * 10);
for (k = 0; k <= cf; k = k + cint)
    g1.DrawLine(Pens.Khaki, k, 0, k, ff);
for (k = -1; k <= ff; k = k + fint)
{
    if(r == 10)
        g1.DrawLine(Pens.Red, 0, k, cf, k);
    else
        g1.DrawLine(Pens.Khaki, 0, k, cf, k);
    r = r + 1;
}
g1.DrawLine(Pens.Red, 0, 0, 0, ff);
}

//Proceso de graficación
public void Graficador(Graphics g1)
{
    if (x < xf - 0.09)
    {
        if (cont == 0)
        {
            cuadricula(g1);
            ltemp.Items.Clear();
            x = xi;
            y = Fu();
            cont = cont + 1;
            cant = Col();
            fant = Fil();
        }
        else
        {
            x = x + 0.1;
            y = Fu();
            cont = cont + 1;
            c = Col();
            f = Fil();
            g1.DrawLine(Pens.Black, cant, fant, c, f);
            cant = c;
            yant = y;
            fant = f;
        }
    }
    else
    {
        pic1.Refresh();
        cuadricula(g1);
        xi = xf;
        xf = xf + xff;
        x = xi + 0.1;
        ltemp.Items.Clear();
        cont = 1;
    }
}
```

```
y = Fu();
c = Col();
f = Fil();
g1.DrawLine(Pens.Black, 0, fant, c, f);
cant = c;
fant = f;
yant = y;
cont = cont + 1;
}
}

//función de conversión de abscisas a columnas
public int Col()
{
    int colu;
    colu = (int)((x - xi) / (xf - xi)) * cf + ((xf - x) / (xf - xi)) * ci);
    return (colu);
}

//función de conversión de ordenadas a filas
public int Fil()
{
    int filas;
    filas = (int)((yf - y) / (yf - yi)) * ff + ((y - yi) / (yf - yi)) * fi);
    return (filas);
}

//procesamiento del dato recibido por el puerto serial (dato proveniente del ADC)
public double Fu()
{
    string puerto;
    double tem = 0, dato=0;

    puerto = serialPort1.ReadLine();
    try
    {
        dato = double.Parse(puerto);
        if (dato <= 65535)
            tem = A * (((dato) *3300 / 65535) - 600) / 10.0); //Ecuación de adecuación del dato al valor de la magnitud.
        else
            tem = yant;
    }

    //Impresión en el control ListBox, de los valores graficados.
    if (cont != 0)
    {
        if (tem.ToString().Length >= 4)
        {
            if (tem > 100)
                ltemp.Items.Add(cont.ToString() + ". " + tem.ToString().Remove(5) + " °");
            else

```

```
        ltemp.Items.Add(cont.ToString() + ". " + tem.ToString().Remove(4) + " °");
    }
    else
        ltemp.Items.Add(cont.ToString() + ". " + tem.ToString() + " °");
}
return (tem);
}
catch (FormatException)
{
    tem = yant;
    return (tem);
}
}

//Evento del botón de regreso a la ventana de selección
private void button2_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("pictures.exe");
    Close();
}

//Evento de la barra de variación de amplitud de graficación.
private void amp_Scroll(object sender, EventArgs e)
{
    if (button3.Visible == true)
        button3.Visible = false;
    pic1.Refresh();
    cont = 0;
    switch (amp.Value)
    {
        case 0:
            A = 10;
            break;
        case 1:
            A = 4;
            break;
        case 2:
            A = 2;
            break;
        case 3:
            A = 4.0 / 3.0;
            break;
        case 4:
            A = 1;
            break;
        case 5:
            A = 0.8;
            break;
        case 6:
            A = 2.0 / 3.0;
            break;
    }
}
```

```
case 7:  
    A = 0.4;  
    break;  
case 8:  
    A = 0.2;  
    break;  
case 9:  
    A = 0.1;  
    break;  
}  
}  
  
//Evento de la barra de desplazamiento para la variación de la velocidad de graficación.  
private void vel_Scroll(object sender, EventArgs e)  
{  
    timer1.Interval = 100 - 10 * (vel.Value);  
}  
  
//Evento del botón de reanudación de graficación  
private void button3_Click(object sender, EventArgs e)  
{  
    timer1.Enabled = true;  
    button1.Text = "Stop";  
    button3.Visible = false;  
    button2.Visible = false;  
    cxf.Enabled = false;  
}  
}  
}
```

Anexo C.6: Programa de ventana de graficación, para el osciloscopio

```
//Librerías por default
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
//Librería para hacer uso de un documento de datos
using System.IO;

//clase por default
namespace Osciloscopio
{
    public partial class Form1 : Form
    {
        //Declaración de variables
        int ci = 1, fi = 1, c, f, cant, fant, cont, cint, fint, cf, ff, jk = 0;
        double x, y, xi, yi = 0, yf = 10, yant, xff, A = 1, cxf, T = 1, xf;
        double[] dato;
        string[] puerto;
        string freq;

        public Form1() //Inicialización de la ventana
        {
            InitializeComponent();
            StreamReader serial = new StreamReader("COM.mat");
            serialPort1.PortName = serial.ReadLine();           //Lectura del nombre del puerto COM
            serial.Close();
            serialPort1.Open();      //Apertura del puerto serial para la comunicación con el microcontrolador
            cxf = 10;             //Número de divisiones que se mostrarán en el PictureBox
        }

        //Evento del botón de Inicio y paro de graficación
        private void button1_Click(object sender, EventArgs e)
        {
            if (button1.Text == "Stop")
            {
                //Proceso para cuando se frena la señal
                timer1.Enabled = false;
                button2.Visible = true;
                button1.Text = "Inicia";
                button3.Visible = true;
                groupBox4.Enable = true;
            }
        }
```

```
else
{
    //Proceso para cuando se inicia la graficación
    cont = 0;
    xi = 0.0;
    xff = cxf / 10;    //número de medidas para llenar el pic
    xf = xff * T;
    cf = pic1.Width;
    ff = pic1.Height;
    pic1.Refresh();    //Limpieza del PictureBox

    //dimensión del vector que guarda los datos recibidos, para crear la señal y su frecuencia
    dato = new double[300];
    puerto = new string[300];

    //Envío del carácter correspondiente al nivel de voltaje seleccionado (alto o bajo)
    if (radioButton1.Checked)
        serialPort1.WriteLine("O");
    else
        serialPort1.WriteLine("B");

    freq = serialPort1.ReadLine();    //Recepción de frecuencia
    label44.Text = freq;
    //Recepcion de datos
    for (int j = 0; j < 300; j++)
    {
        puerto[j] = serialPort1.ReadLine();
        dato[j] = Convert.ToDouble(puerto[j]);
    }

    groupBox4.Enabled = false;
    label44.Visible = true;
    label45.Visible = true;
    timer1.Enabled = true;
    button2.Visible = false;
    button3.Visible = false;
    button1.Text = "Stop";
}

}

//Poroceso para dibujar una cuadricula en el PictureBox
public void cuadricula(Graphics g1)
{
    int k;
    fint = ff / 10;
    cint = cf / (int)((xff) * 10);

    for (k = 0; k <= cf; k = k + cint)
        g1.DrawLine(Pens.Khaki, k, 0, k, ff);
```

```
for (k = ff; k >= 0; k = k - fint)
{
    g1.DrawLine(Pens.Khaki, 0, k, cf, k);
}
g1.DrawLine(Pens.Red, 0, 0, 0, ff);
g1.DrawLine(Pens.Red, 0, ff-1, cf, ff-1);
}

//Proceso de graficación
public void Graficador(Graphics g1)
{
    if (x < xf)
    {
        if (cont == 0)
        {
            cuadricula(g1);
            x = xi;
            y = Fu();
            cont = cont + 1;
            cant = Col();
            fant = Fil();
        }
        else
        {
            x = x + 0.1;
            y = Fu();
            cont = cont + 1;
            c = Col();
            f = Fil();
            g1.DrawLine(Pens.Black, cant, fant, c, f); //Graficación de un punto intermedio en el PictureBox
            cant = c;
            fant = f;
        }
    }
    else
    {
        pic1.Refresh();
        cuadricula(g1);
        xi = xf;
        xf = xf + xff * T;
        x = xi + 0.1;
        cont = 1;
        y = Fu();
        c = Col();
        f = Fil();
        g1.DrawLine(Pens.Black, 0, fant, c, f); //Graficación del primer punto del PictureBox
        cant = c;
        fant = f;
        cont = cont + 1;
    }
}
```

```
//Conversión de abscisas a columnas
public int Col()
{
    int colu;
    colu = (int)((x - xi) / (xf - xi)) * cf + ((xf - x) / (xf - xi)) * ci;
    return (colu);
}

//Conversión de ordenadas a filas
public int Fil()
{
    int filas;
    filas = (int)((yf - y) / (yf - yi)) * ff + ((y - yi) / (yf - yi)) * fi;
    return (filas);
}

//Adecuación del dato recibido al voltaje medido
public double Fu()
{
    double v = 0;
    v = A * (dato[jk]) * 3.3 / 65535 + 5;
    return (v);
}

//Evento del botón de regreso a la ventana de selección de magnitud a medir
private void button2_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("pictures.exe");
    Close();
}

//Evento del timer, que se ejecuta cuando ha transcurrido un tiempo determinado, en el cual se grafica un punto
private void timer1_Tick(object sender, EventArgs e)
{
    Graphics g1 = pic1.CreateGraphics();
    if (jk < 300)
    {
        Graficador(g1);
        jk++;
    }
    else
    {
        if (altas.Checked)
            serialPort1.WriteLine("A");
        else
        {
            if (medias.Checked)
                serialPort1.WriteLine("B");
            else
                serialPort1.WriteLine("C");
        }
    }
}
```

```
frec = serialPort1.ReadLine();
label44.Text = frec;
fr = double.Parse(frec);

if (fr < 150)
{
    if (altas.Checked)
        serialPort1.WriteLine("D");
    else
    {
        if (medias.Checked)
            serialPort1.WriteLine("E");
        else
            serialPort1.WriteLine("F");
    }
}
else
{
    if (altas.Checked)
        serialPort1.WriteLine("G");
    else
    {
        if (medias.Checked)
            serialPort1.WriteLine("H");
        else
            serialPort1.WriteLine("I");
    }
}

for (int j = 0; j < 300; j++)
{
    puerto[j] = serialPort1.ReadLine();
    dato[j] = Convert.ToDouble(puerto[j]);
}

jk = 0;
cont = 0;
pic1.Refresh();
cuadricula(g1);
}

//Evento del botón de reanudación de graficación
private void button3_Click(object sender, EventArgs e)
{
    groupBox4.Enabled = false;
    timer1.Enabled = true;
    button3.Visible = false;
    button1.Text = "Stop";
}
```

```
//Evento de la barra de desplazamiento, para aumentar o disminuir la amplitud de la señal a graficar
private void amp_Scroll(object sender, EventArgs e)
{
    switch (amp.Value)
    {
        case 0:
            A = 4;
            break;
        case 1:
            A = 2;
            break;
        case 2:
            A = 1.333;
            break;
        case 3:
            A = 1;
            break;
        case 4:
            A = 0.8;
            break;
        case 5:
            A = 0.666;
            break;
        case 6:
            A = 0.5;
            break;
        case 7:
            A = 0.4;
            break;
        case 8:
            A = 0.333;
            break;
        case 9:
            A = 0.25;
            break;
    }
    pic1.Refresh();
    cont = 0;
}

//Evento del la barra de desplazamiento, para aumentr o disminuir la velocidad de graficación
private void vel_Scroll(object sender, EventArgs e)
{
    if (vel.Value == 9)
        timer1.Interval = 1;
    else
        timer1.Interval = 100 - 10 * (vel.Value);
}
```

```
//Evento de la barra de desplazamiento, para aumentar o disminuir el número de muestras a graficar en el  
PictureBox  
private void tiempo_Scroll(object sender, EventArgs e)  
{  
    switch (tiempo.Value)  
    {  
        case 0:  
            T = 1;  
            break;  
        case 1:  
            T = 2;  
            break;  
        case 2:  
            T = 3;  
            break;  
        case 3:  
            T = 5;  
            break;  
        case 4:  
            T = 10;  
            break;  
        case 5:  
            T = 15;  
            break;  
        case 6:  
            T = 20;  
            break;  
        case 7:  
            T = 25;  
            break;  
        case 8:  
            T = 35;  
            break;  
        case 9:  
            T = 50;  
            break;  
    }  
    xf = xff * T;  
    cont = 0;  
    pic1.Refresh();  
}  
}  
}
```

Anexo C.7: Programa de ventana del generador de funciones

```
//Librerias agregadas por default por el software
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Gen_de_funciones
{
    public partial class Form1 : Form
    {
        //declaración de variables
        int ci, fi, cf, f, n, cant=0, fant=0, k;
        double x, y, xi, yi, yf, h, T, Ta, TT, TTa, A = 1;
        double xf, xff, xii, rango, per,xant;

        public Form1()
        {
            //Inicialización de la ventana
            InitializeComponent();
            cxi.Text = "-4";
            cxf.Text = "4";
            cT.Text = "2";
            cTa.Text = "1";
            //Textos mostrados en un lista desplegable, para poder seleccionar función a generar
            comboBox1.Items.Add("Coseno");
            comboBox1.Items.Add("Seno");
            comboBox1.Items.Add("Pulso");
            comboBox1.Items.Add("Rampa");
            comboBox1.Items.Add("Triangular");
            comboBox1.Items.Add("Rectangular");
        }

        //Evento del botón de inicio y paro de graficación
        private void button1_Click(object sender, EventArgs e)
        {
            if (button1.Text == "Stop")
            {
                comboBox1.Enabled = true;
                button1.Text = "Inicia";
                timer1.Enabled = false;
                s.Visible = true;
                cxi.ReadOnly = false;
                cxf.ReadOnly = false;
```

```
if (comboBox1.SelectedIndex == 2)
{
    cT.ReadOnly = false;
    cTa.ReadOnly = false;
}
else
{
    if (comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 || comboBox1.SelectedIndex == 5)
        cT.ReadOnly = false;
}
else
{
    if (comboBox1.SelectedItem != null)
    {
        //Adquisición de datos para generar cualquier función
        cx1.ReadOnly = true;
        cxf.ReadOnly = true;
        if (comboBox1.SelectedIndex == 2)
        {
            //Adquisición de datos para generar un pulso
            cT.Visible = true;
            cTa.Visible = true;
            IT.Visible = true;
            ITa.Visible = true;
            TT = Double.Parse(cT.Text);
            TTa = Double.Parse(cTa.Text);
            T = TT;
            Ta = TTa;
            if (TT < 0 | TTa < 0)
                MessageBox.Show("Los Periodos no pueden ser negativos");
            else
            {
                if (TT == 0 | TTa == 0)
                    MessageBox.Show("Los Periodos no pueden ser cero");
                else
                {
                    if (TTa > TT)
                        MessageBox.Show("El tiempo alto debe ser menor al periodo");
                    else
                    {
                        cT.ReadOnly = true;
                        cTa.ReadOnly = true;
                        Datos();
                        per = xi;
                    }
                }
            }
        }
    }
}
else
{
    if (comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 || comboBox1.SelectedIndex == 5)
    {
```

```
//Adquisición de datos para generar las funciones rampa, triangular y rectangular
cT.Visible = true;
IT.Visible = true;
TT = Double.Parse(cT.Text);
T = TT;
cT.ReadOnly = true;
Datos();
per = xi;
}
else
    Datos();
}
else
    MessageBox.Show("Selecciona una forma de onda");
}

private void Datos()
{
    //Inicialización de variables
comboBox1.Enabled = false;
pic1.Refresh();
button1.Text = "Stop";
s.Visible = false;
xii = (double.Parse(cx1.Text));
xff = (double.Parse(cxf.Text));
if (xii >= xff)
{
    MessageBox.Show("xi debe ser menor que xf");
    comboBox1.Enabled = true;
    button1.Text = "Inicia";
    cx1.ReadOnly = false;
    cxf.ReadOnly = false;
    cT.ReadOnly = false;
    cTa.ReadOnly = false;
}
else
{
    xi = xii * (tiempo.Value + 1);
    xf = xff * (tiempo.Value + 1);
    cf = pic1.Width;
    ff = pic1.Height;
    k = 0;
    rango = xf - xi;
    ci = 0;
    fi = 0;
    n = cf - ci;
    h = ((xff - xii) / n) * 9.66;
    yi = -5.1;
    yf = 5.1;
```

```
        timer1.Enabled = true;
    }
}

//Evento del timer, para que cada determinado tiempo (establecido), se grafique un pedazo de la función
private void timer1_Tick(object sender, EventArgs e)
{
    k = k + 1;
    Graphics g1 = pic1.CreateGraphics();
    Graficador(g1);
}

//Proceso de graficación de la función seleccionada
public void Graficador(Graphics g1)
{
    if (xi * xf <= 0)
    {
        x = 0;
        c = Col();
        g1.DrawLine(Pens.Red, c, fi, c, ff);
    }
    if (yi * yf <= 0)
    {
        y = 0;
        f = Fil();
        g1.DrawLine(Pens.Red, ci, f, cf, f);
    }
    if (k == 1)
    {
        cuadricula(g1);
        x = xi;
        y = Fu(g1);
        xant = x;
        cant = Col();
        fant = Fil();
    }
    else
    {
        if (cant < cf)
        {
            x = xi + h * k;
            y = Fu(g1);
            xant = x;
            c = Col();
            f = Fil();
            g1.DrawLine(Pens.Black, cant, fant, c, f);
            cant = c;
            fant = f;
        }
        else
        {

```

```
k = 0;
pic1.Refresh();
xi = xf;
xf = xf + rango;
}
}

//Conversión de abscisas a columnas
public int Col()
{
    int colu;
    colu = (int)((x - xi) / (xf - xi)) * cf + ((xf - x) / (xf - xi)) * ci;
    return (colu);
}

//Conversión de ordenadas a filas
public int Fil()
{
    int filas;
    filas = (int)((yf - y) / (yf - yi)) * ff + ((y - yi) / (yf - yi)) * fi;
    return (filas);
}

//determinación de la función a generar, en base a lo seleccionado por el usuario
public double Fu(Graphics g1)
{
    double yy = 0;
    int op;
    op=comboBox1.SelectedIndex;
    switch (op)
    {
        case 0:
            yy = A * Math.Cos(x); //Generación de función coseno
            break;
        case 1:
            yy = A * Math.Sin(x); //Generación de función seno
            break;
        case 2:
            //Generación de una señal pulso
            Ta = per + TTa;
            if (xant < Ta && x > Ta)
            {
                x = Ta;
                y = A;
                c = Col();
                f = Fil();
                g1.DrawLine(Pens.Black, cant, fant, c, f);
                yy = 0;
                cant = c;
                fant = f;
            }
    }
}
```

```
        }
    else
    {
        if (x >= per + T)
        {
            per = per + T;
            x = per;
            y = 0;
            c = Col();
            f = Fil();
            g1.DrawLine(Pens.Black, cant, fant, c, f);
            yy = A;
            cant = c;
            fant = f;
        }
        else
        {
            if (x < Ta)
                yy = A;
            else
                yy = 0;
        }
    }
    break;
case 3:
//Generación de función rampa
if (x >= per + T)
{
    per = per + T;
    x = per;
    y = A * TT;
    c = Col();
    f = Fil();
    g1.DrawLine(Pens.Black, cant, fant, c, f);
    yy = 0;
    cant = c;
    fant = f;
}
else
    yy = A * (x - per);
break;
case 4:
//Generación de función triangular
Ta = per + T / 2;
if (xant < Ta && x > Ta)
{
    x = Ta;
    y = A * TT / 2;
    c = Col();
    f = Fil();
    g1.DrawLine(Pens.Black, cant, fant, c, f);
```

```
    yy = -A * (x - Ta) + A * (T / 2);
    cant = c;
    fant = f;
}
else
{
    if (x >= per + T)
    {
        per = per + T;
        x = per;
        yy = 0;
    }
    else
    {
        if (x <= Ta)
            yy = A * (x - per);
        else
            yy = -A * (x - Ta) + A * (T / 2);
    }
}
break;
case 5:
//Generación de función rectangular
Ta = per + T / 2;
if (xant < Ta && x > Ta)
{
    x = Ta;
    y = A;
    c = Col();
    f = Fil();
    g1.DrawLine(Pens.Black, cant, fant, c, f);
    yy = -A;
    cant = c;
    fant = f;
}
else
{
    if (x >= per + T)
    {
        per = per + T;
        x = per;
        y = -A;
        c = Col();
        f = Fil();
        g1.DrawLine(Pens.Black, cant, fant, c, f);
        yy = A;
        cant = c;
        fant = f;
    }
    else
    {
```

```
        if (x < Ta)
            yy = A;
        else
            yy = -A;
    }
}
break;
}
return (yy);
}

//Evento del botón de reanudación de graficación
private void s_Click(object sender, EventArgs e)
{
    button1.Text = "Stop";
    timer1.Enabled = true;
    cxi.Text = xii.ToString();
    cxf.Text = xff.ToString();
    cxi.ReadOnly = true;
    cxf.ReadOnly = true;
    s.Visible = false;
    comboBox1.Enabled = false;
    if (comboBox1.SelectedIndex == 2)
    {
        cT.Text = TT.ToString();
        cTa.Text = TTa.ToString();
        cT.ReadOnly = true;
        cTa.ReadOnly = true;
    }
    else
    {
        if (comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 || comboBox1.SelectedIndex == 5)
        {
            cT.Text = TT.ToString();
            cT.ReadOnly = true;
        }
    }
}

//Proceso para dibujar una cuadricula en el área de graficación (PictureBox)
public void cuadricula(Graphics g1)
{
    x = 0;
    int c1 = Col();
    y = 0;
    int f1 = Fil();
    for (int k = c1; k < cf; k++)
    {
        g1.DrawLine(Pens.Khaki, k, 0, k, ff);
        if (x <= xf)
        {
```

```
        x++;
        k = Col();
    }
}
x = 0;
for (int k = c1; k > ci; k--)
{
    g1.DrawLine(Pens.Khaki, k, 0, k, ff);
    if (x >= xi)
    {
        x--;
        k = Col();
    }
}
for (int k = f1; k <= ff; k--)
{
    g1.DrawLine(Pens.Khaki, 0, k, cf, k);
    if (y <= yf)
    {
        y--;
        k = Fil();
    }
}
y = 0;
for (int k = f1; k >= fi; k++)
{
    g1.DrawLine(Pens.Khaki, 0, k, cf, k);
    if (y >= yi)
    {
        y++;
        k = Fil();
    }
}

//Evento de la barra desplazable de modificación de velocidad de graficación
private void vel_Scroll(object sender, EventArgs e)
{
    if (vel.Value == 9)
        timer1.Interval = 1;
    else
        timer1.Interval = 100 - 10 * (vel.Value);
}

//Evento de la barra desplazable de modificación de amplitud de la señal a generar
private void amp_Scroll(object sender, EventArgs e)
{
    pic1.Refresh();
    k = 0;
    switch (amp.Value)
    {
```

```
case 0:  
    A = 0.1;  
    break;  
case 1:  
    A = 0.25;  
    break;  
case 2:  
    A = 0.5;  
    break;  
case 3:  
    A = 0.75;  
    break;  
case 4:  
    A = 1;  
    break;  
case 5:  
    A = 1.25;  
    break;  
case 6:  
    A = 1.5;  
    break;  
case 7:  
    A = 2;  
    break;  
case 8:  
    A = 2.5;  
    break;  
case 9:  
    A = 4;  
    break;  
}  
s.Visible = false;  
xi = xii * (tiempo.Value + 1);  
xf = xff * (tiempo.Value + 1);  
if (comboBox1.SelectedIndex == 2 || comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 ||  
comboBox1.SelectedIndex == 5)  
    per = xi;  
}  
  
//Evento para la lista desplegable de selección de función a generar  
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)  
{  
    s.Visible = false;  
    pic1.Refresh();  
    if (comboBox1.SelectedIndex == 2)  
    {  
        cT.Visible = true;  
        cTa.Visible = true;  
        IT.Visible = true;  
        ITa.Visible = true;  
    }  
}
```

```
        else
    {
        if (comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 || comboBox1.SelectedIndex == 5)
        {
            cT.Visible = true;
            IT.Visible = true;
            cTa.Visible = false;
            ITa.Visible = false;
        }
        else
        {
            cT.Visible = false;
            cTa.Visible = false;
            IT.Visible = false;
            ITa.Visible = false;
        }
    }
}

//Evento de la barra desplazable de modificación de muestras a graficar por PictureBox
private void tiempo_Scroll(object sender, EventArgs e)
{
    k = 0;
    s.Visible = false;
    pic1.Refresh();
    xi = xii * (tiempo.Value + 1);
    xf = xff * (tiempo.Value + 1);
    rango = xf - xi;
    if (comboBox1.SelectedIndex == 2 || comboBox1.SelectedIndex == 3 || comboBox1.SelectedIndex == 4 ||
    comboBox1.SelectedIndex == 5)
        per = xi;
}

//Evento del botón de regreso a la ventana de selección de magnitud a analizar
private void button2_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("pictures.exe");
    Close();
}

//Conversión de columnas a abscisas
private double eq(int columna)
{
    double eqq;
    eqq = ((columna * (xf - xi) + cf * xi + ci * xf) / (cf - ci));
    return eqq;
}

//Evento para mostrar el valor de la función, cuando ésta está frenada y se pasa el cursor sobre la señal
private void pic1_MouseMove(object sender, MouseEventArgs e)
{
```

```
int columna = 0, fila = 0, p = 0;
double ye, equis, xx, cont = 0;
Graphics g3 = pic1.CreateGraphics();
int opc;
columna = e.X;
fila = e.Y;
equis = eq(cant);
x = eq(columna);
opc = comboBox1.SelectedIndex;
switch (opc)
{
    case 0:
        y = A * Math.Cos(x);
        break;
    case 1:
        y = A * Math.Sin(x);
        break;
    case 2:
        xx = x;
        for (int j = 0; xx >= xii * (tiempo.Value + 1) + TT; j++)
            xx = xx - TT;

        if (xx <= (xii * (tiempo.Value + 1) + TTa))
            y = A;
        else
            y = 0;
        break;
    case 3:
        xx = x;
        cont = xii * (tiempo.Value + 1);
        for (int j = 0; xx >= xii * (tiempo.Value + 1) + TT; j++)
        {
            cont = cont + TT;
            xx = xx - TT;
        }
        y = A * x - A * cont;
        break;
    case 4:
        xx = x;
        cont = xii * (tiempo.Value + 1);
        for (int j = 0; xx >= xii * (tiempo.Value + 1) + TT; j++)
        {
            cont = cont + TT;
            xx = xx - TT;
            p++;
        }
        if (xx <= xii * (tiempo.Value + 1) + TT / 2)
            y = A * (x - cont);
        else
            y = A * (-x + cont + TT);
        break;
}
```

```
case 5:  
    xx = x;  
    for (int j = 0; xx >= xii * (tiempo.Value + 1) + TT; j++)  
        xx = xx - TT;  
    if (xx <= (xii * (tiempo.Value + 1) + TTa))  
        y = A;  
    else  
        y = -A;  
    break;  
}  
ye = Fil();  
if ((fila <= ye + 5) && (fila >= ye - 5))  
{  
    toolTip1.Active = true;  
    toolTip1.AutoPopDelay = 5000;  
    if (x <= xant)  
    {  
        toolTip1.Active = true;  
        toolTip1.AutoPopDelay = 5000;  
        if (s.Visible == true)  
        {  
            if (x.ToString().Length > 5)  
            {  
                if (y.ToString().Length > 5)  
                    toolTip1.SetToolTip(pic1, x.ToString().Remove(5) + "," + y.ToString().Remove(5));  
                else  
                    toolTip1.SetToolTip(pic1, x.ToString().Remove(5) + "," + y.ToString());  
            }  
            else  
            {  
                if (y.ToString().Length > 5)  
                    toolTip1.SetToolTip(pic1, x.ToString() + "," + y.ToString().Remove(5));  
                else  
                    toolTip1.SetToolTip(pic1, x.ToString() + "," + y.ToString());  
            }  
        }  
    }  
    else  
    {  
        toolTip1.Active = false;  
        toolTip1.AutoPopDelay = 500;  
    }  
}  
else  
{  
    toolTip1.Active = false;  
    toolTip1.AutoPopDelay = 500;  
}  
}  
}  
}
```

Anexo D

Anexo D.1: Evaluación económica

Caracterización:

El proyecto, es un producto que consta de dos partes, una tangible (circuitería) y otra intangible (software). La circuitería está constituida por un microcontrolador, sensores (uno de temperatura, uno de presión, uno de humedad y uno de densidad de flujo magnético) y arreglos para el acondicionamiento de señal. El software, es una aplicación a usar sobre el SO Windows, la cual grafica la señal detectada por algún sensor especificado por el usuario, o bien, la variación de voltaje proveniente de un circuito externo.

C clientes potenciales:

- Alumnos de nivel superior o media superior con estudios en electrónica, control y/o materias relacionadas con análisis de señales.
- Personas que necesiten analizar señales eléctricas y/o de control de diferentes magnitudes (temperatura, presión, humedad y densidad de flujo magnético).

Principal oferente: Osciloscopio USB con Interfaz Gráfica en LabView ([Página 43](#))

Similitudes del proyecto, con el Osciloscopio USB con Interfaz Gráfica en LabView.

- Ambos utilizan un puerto COM virtual, para comunicar el microcontrolador con la PC.
- Interfaz gráfica amigable.
- Controles para la adecuación del rango de la señal en la aplicación.

Diferencias del proyecto con el Osciloscopio USB con Interfaz Gráfica en LabView.

Osciloscopio y medidor virtual inalámbrico, implementado con tecnología .NET, utilizando sensores y microcontrolador ARM-M0+		Osciloscopio USB, con interfaz gráfica en LabView
Interfaz con Visual Studio		Interfaz con LabView
Mide voltaje, temperatura, presión, humedad y densidad de flujo magnético, y cuenta con un generador de funciones virtual		Solo mide voltajes
Programación en C#		Programación en C
Bajo consumo de energía		Alto consumo de energía
Dimensiones razonables para ser transportable (10 x 12 x 5 cm. aprox.)		Prototipo robusto
Usa barras deslizables para ajustar la señal		Utiliza botones y perillas difíciles de usar
No necesita instalaciones adicionales		Necesita instalación de programas y paquetería adicional.

Tabla D.1.1: Comparación de características entre el proyecto creado y el Osciloscopio USB con Interfaz Gráfica en LabView.

Costo del desarrollo del prototipo:

Material	Precio Unitario	Piezas	Precio Total
Tarjeta de desarrollo FRDM-KL25Z Freescale ³⁷	\$250.00	1	\$250.00
Cable MicroUSB ³⁸	\$35.00	1	\$35.00
Placa fenólica 10x15 ³⁹	\$7.00	1	\$7.00
Sensor LM61CIZ ³⁸	\$12.10	1	\$12.10
Sensor UGN3503 ³⁸	\$18.15	1	\$18.15
Sensor HMZ433A1 ³⁸	\$129.18	1	\$129.18
Sensor MPX10DP ³⁸	\$122.05	1	\$122.05
TL081 ³⁹	\$8.00	4	\$32.00
Resistores ³⁹	\$0.40	22	\$8.40
BC547 ³⁹	\$3.00	2	\$6.00
Conectores de 2 entradas ³⁹	\$5.00	5	\$15.00
Conectores de 3 entradas ³⁹	\$6.00	1	\$6.00
Trimpot 10K ³⁹	\$25.00	4	\$100.00
Headers tipo hembra 40 pines ³⁹	\$13.00	2	\$26.00
Header tipo macho 40 pines ³⁹	\$5.00	1	\$5.00
Base de 8 pines ³⁹	\$1.50	5	\$7.50
Flux ³⁹	\$35.00	1	\$35.00
Soldadura (1m) ³⁹	\$5.00	1	\$5.00
Cloruro férrico ³⁹	\$40.00	1	\$40.00
AD620 amplificador de intrumentación ³⁸	\$180.00	1	\$180.00
Diodo 1N4001 ³⁹	\$1.00	2	\$2.00
Alambre (1m) ³⁹	\$2.50	1	\$2.50
TOTAL	---	---	\$1,037.50

Tabla D.1.2: Desglose de costo del proyecto.

Posibilidades de producción:

Debido a que es necesario pagar regalías y licencias tanto a los proveedores de tecnología en el microcontrolador (freescale), como a las empresas distribuidoras de los softwares de desarrollo utilizados (Visual Studio 2012, CodeWarrior y Ares), no será posible la producción masiva del proyecto.

Una de las opciones que se tienen para el prototipo es que su venta, propaganda y distribución sea directamente con el cliente, por lo tanto, las piezas a producir son en base a los pedidos que se tengan.

³⁷ http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z (Consulta 26/04/2013)

³⁸ http://www.agelectronica.com/Virtual_Shop/index.asp (Consulta 10/05/2013)

³⁹ Tienda de material electrónico “SISCO” (Consulta 17/05/2013)

Insumos:

- Salario:
 - Salario de un ingeniero⁴⁰: 5 salarios mínimos mensuales
Salario mínimo⁴¹ = \$1,968.70
Salario para un ingeniero = \$9,843.50
Semanas de trabajo = 35 → días = 210 → horas x día = 4
Meses de trabajo = 9
→ Total: \$9,843.50 X 9 = \$88,591.5, tomando en cuenta que en el proyecto trabajaron 2 personas
→ Total salarios: \$177,183.00
- Instalaciones:
 - Renta de lugar de trabajo⁴²: \$2,300.00 x mes
Meses de trabajo: 9
→ Total de renta: \$20,700.00
 - Energía eléctrica⁴³: \$1.50 por hr.
Horas de trabajo = 840 horas
→ Total de gasto en energía eléctrica: \$1,260.00
 - Internet⁴⁴: \$349.00 por hr.
Meses = 9
→ Gasto en internet: \$3,141.00
- Licencias:
 - VS2012⁴⁵ = \$6,611.00
 - CodeWarrior: Licencia gratuita para programas de menos de 32 KB.
 - Microsoft Office Profesional 2013⁴⁶ = \$6,399.00
 - S.O. Microsoft Windows 8.1 Pro⁴⁷ = \$3,735.00
→ Total: \$16,745.00
- Herramientas:
 - Pinzas de corte: \$50.00
 - Pinzas de punta: \$50.00
 - Cautín: \$100.00
→ Total: \$200.00

⁴⁰ <http://www.laneros.com/temas/escala-salarios-de-la-aciem-para-ingenieros.111882/> (Consulta 20/10/2013)

⁴¹ [http://portal.infonavit.org.mx/wps/wcm/connect/Infonavit/Trabajadores/Convertidor+de+pesos+a+Veces+Salario+Minimo+\(VSM\)](http://portal.infonavit.org.mx/wps/wcm/connect/Infonavit/Trabajadores/Convertidor+de+pesos+a+Veces+Salario+Minimo+(VSM)) (Consulta 20/10/2013)

⁴² <http://www.olx.com.mx/q/mensuales/c-363> (Consulta 20/10/2013)

⁴³ <http://www.cfe.gob.mx> (Consulta 20/10/2013)

⁴⁴ <http://www.telmex.com/web/hogar/otros-planos-de-internet> (Consulta 20/10/2013)

⁴⁵ <http://www.visualstudio.com/es-es> (Consulta 20/10/2013)

⁴⁶ http://www.microsoftstore.com/store/msmx/es_MX/pdp/Office-Professional-2013/productID.260739700 (Consulta 20/10/2013)

⁴⁷ http://www.microsoftstore.com/store/msmx/es_MX/cat/Windows/categoryID.65865100 (Consulta 20/10/2013)

- Equipo:
 - Computadora: \$5,000.00, teniendo en cuenta que para el proyecto, se necesitó de dos desarrolladores:
→Total: \$10,000.00
- Extras:
 - Impresiones
Color → \$3.50 X 250 hojas = \$875.00
Blanco y negro → \$0.50 X 350 hojas = \$175.00
 - Transporte (para compra de material): \$540.00
→Total: \$1,590.00

Concepto	Gasto unitario	Cantidad	Total
Sueldos	\$9,843.50 / mes	9 X 2	\$177,183.00
Energía Eléctrica	\$1.50 / hora	840	\$1,260.00
Internet	\$8.00 / hora	500	\$3,141.00
Renta	\$1,500.00 / mes	9	\$20,700.00
Licencias	---	---	\$16,745.00
Herramientas	---	---	\$200.00
Equipo	\$5,000.00	2	\$10,000.00
Extras	---	---	\$ 1,590.00
Total	---	---	\$230,819.00

Tabla D.1.3: Desglose de gastos en la realización del proyecto.

Valor del proyecto = Costos + Gastos = \$1,037.50 + \$230,819.00

Valor del proyecto = \$231,856.50

Referencias

- [1] http://www.ecured.cu/index.php/Historia_de_la_medici%C3%B3n (Consulta 25/03/2013)
- [2] http://mimosa.pntic.mec.es/mlucas2/softEduca/umedida/la_medicion_en_la_historia.html (Consulta 25/03/2013)
- [3] <http://bogotasiempreazul.blogspot.mx/2010/08/osciloscopio-historia-el-primer-metodo.html> (Consulta 25/03/2013)
- [4] <http://todoingenieriaindustrial.wordpress.com/metrologia-y-normalizacion/2-3-sistemas-de-medicion-temperatura-presion-torsion-y-esfuerzos-mecanicos/> (Consulta 26/03/2013)
- [5] R. Zouckermann, Fundamenta Scientiae 2, 1981, pg.185. (Consulta 30/03/2013)
- [6] P.A. Tunbridge, Notes and Records of the Royal Society of London 26, 1971, pg. 15. (Consulta 30/03/2013)
- [7] L.B. Hunt, Journal of Scientific Instruments 21, 1944, pg. 37. (Consulta 30/03/2013)
- [8] W.E.K. Middleton, *The History of Barometer* (Johns Hopkins Press, Baltimore, 1964), pg. 373-375. (Consulta 30/03/2013)
- [9] <http://www.freunde-alter-wetterinstrumente.de/> (Consulta 30/03/2013)
- [10] https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&ved=0CFIQFjAC&url=http%3A%2F%2Fwww.testo.com%2Fonline%2Fembedded%2FSites%2FARG%2FSharedDocuments%2FDownloads%2FTecnologxa_de_medicixn_para_areas_limpias_y_sistemas_VAC.doc&ei=pBFYUrOnH8f8qwG8x4HoBQ&usg=AFQjCNG149iJQsO4wC5uh6n4tDC36hVfWg&sig2=i3xrbebOJHYwo0bF8PiZ3A&bvm=bv.53899372,d.aWM (Consulta 01/04/2013)
- [11] http://www.ecured.cu/index.php/Efecto_Hall (Consulta 30/03/2013)
- [12] <http://prezi.com/x1dwrlexxiad/plataformas-de-desarrollo-visual-e-interfaz-grafica-del-usuario-gui/> (Consulta 02/07/2013)
- [13] <http://www.ferret.com.au/c/Emona-Instruments/New-GW-VPO-Oscilloscopes-Set-to-Replace-DSO-Technology-n905339> (Consulta 02/04/2013)
- [14] <http://www.gigatronic.es/osciloskopios> (Consulta 02/04/2013)
- [15] <http://electronica.webcindario.com/news/n014.html> (Consulta 01/04/2013)
- [16] <http://www.mathworks.com/products/matlab/> (Consulta 02/04/2013)
- [17] <http://www.mathworks.com/products/simulink/> (Consulta 02/04/2013)
- [18] <http://www.ni.com/labview/esa/> (Consulta 02/04/2013)
- [19] [http://msdn.microsoft.com/es-mx/library/kx37x362\(v=vs.80\).aspx](http://msdn.microsoft.com/es-mx/library/kx37x362(v=vs.80).aspx) (Consulta 03/04/2013)
- [20] <http://www.mathworks.com/products/dsp-system/> (Consulta 02/04/2013)
- [21] http://wikitronica.labc.usb.ve/index.php/Freescale_Codewarrior_6.3 (Consulta 03/04/2013)
- [22] <http://www.iar.com/> (Consulta 03/04/2013)
- [23] <http://www.microchip.com/mplabx/> (Consulta 03/04/2013)
- [24] <http://www.gwinstek.com/en/product/productdetail.aspx?pid=3&mid=7&id=1290> (Consulta 03/04/2013)
- [25] <http://cp.literature.agilent.com/litweb/pdf/5991-1103EN.pdf> (Consulta 04/04/2013)
- [26] <http://www.gigatronic.es/osciloskopios/osciloskopios-lecroy-wavesurfer-serie-b-cuarta-generacion> (Consulta 04/04/2013)
- [27] <http://www.tme.eu/es/details/lc-hdo4104/osciloskopios-digitales/lecroy/hdo4104/#> (Consulta 09/04/2013)

Referencias

- [28] <http://cp.literature.agilent.com/litweb/pdf/5989-7819EN.pdf> (Consulta 13/04/2013)
- [29] http://cdn.teledynelecroy.com/files/pdf/lecroy_labmaster_10_zi_datasheet.pdf (Consulta 13/04/2013)
- [30] <http://superelectronicaccosciloscopio.blogspot.com/> (Consulta 26/03/2013)
- [31] <http://www.picotech.com/handheld-scope.html>, <http://www.logicbus.com.mx/picoescop2104-2105.php> (Consulta 26/03/2013)
- [32] http://www.hantek.com/english/produce_list.asp?unid=62 (Consulta 26/03/2013)
- [33] http://www.diotronic.com/instrumentacion/instrumentos-para-pc/instrumentos-para-pc/osciloscopio-usb-para-pc-2cana_r_846_14781.aspx (Consulta 26/03/2013)
- [34] http://www.hantek.com.cn/english/produce_list.asp?unid=94
http://www.hantek.com/english/produce_list.asp?unid=62 (Consulta 26/03/2013)
- [35] <http://www.fluke.com/fluke/usen/portable-oscilloscopes/fluke-190-series-ii-scopemeter.htm?PID=70366> (Consulta 26/03/2013)
- [36] Harper Enríquez, Fundamentos de Electricidad. Dispositivos y circuitos en corriente continua, Vol. 2, 1994, pg. 155 -170. (Consulta 30/03/2013)
- [37] TEKTRONIX, Operación Básica del Osciloscopio, "Electrónica fácil" Nº 10, Junio de 1990. (Consulta 30/03/2013)
- [38] <http://www.systems.caltech.edu/dsp/students/bojan/journ/Nyquist7decades.pdf> (Consulta 30/03/2013)
- [39] <http://www.edn.com/article/CA293235.html> (Consulta 26/03/2013)
- [40] Manual de física, Yavorsky, Detlaf, pg. 8 (Consulta 28/03/2013)
- [41] <http://www0.unsl.edu.ar/~interfases/labs/lab09.pdf> (Consulta 28/03/2013)
- [42] http://www.inymet.com.mx/web_inymet/ingenieria_y_metrologia/informacion/info/TERMOPARES.PDF (Consulta 29/03/2013)
- [43] <http://materias.fi.uba.ar/7609/material/S0307MedicionTemperatura2.pdf> (Consulta 29/03/2013)
- [44] <http://snsoresdepresion.blogspot.mx/> (Consulta 01/04/2013)
- [45] http://www.dte.us.es/ing_inf/ins_elec/temario/Tema%208.%20Convertidores%20A-D.pdf (Consulta 24/06/2013)
- [46] <http://www.cursomicros.com/avr/conversor-adc/adc-de-aproximaciones-sucesivas.html> (Consulta 24/06/2013)
- [47] <http://bibing.us.es/proyectos/abreproj/11237/fichero/PROYECTO+FIN+DE+CARRERA%252FCAPITULO02.pdf> (Consulta 26/06/2013)
- [48] http://webs.uvigo.es/ario/docencia/sad/SADGRADO_3.pdf (Consulta 26/06/2013)
- [49] http://www.dte.us.es/ing_inf/ins_elec/temario/Tema%201.%20Amplificadores%20Operacionales.pdf (Consulta 06/07/2013)
- [50] <http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/6883/1/ice%20%2020173a.pdf> (Consulta 03/08/2013)
- [51] http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/morales_h_oe/capitulo3.pdf (Consulta 20/04/2013)
- [52] http://www.el.uma.es/marin/Practica4_UART.pdf (Consulta 24/04/2013)
- [53] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z (Consulta 26/04/2013)
- [54] http://www.tervenet.com/itmaz/micros2/PIC32_11_UART.pdf (Consulta 03/05/2013)

