



# **UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES  
DE COMUNICACIÓN**

**TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO EN ELECTRÓNICA Y REDES DE COMUNICACIÓN**

**TEMA:**

**“MÓDULO DIDÁCTICO DE HARDWARE LIBRE PARA ANÁLISIS Y  
VERIFICACIÓN DE SISTEMAS ANÁLOGOS DIGITALES.”**

**AUTOR: Vicente Javier Ruiz Nogales**

**DIRECTOR: Ing. Milton Bertulfo López Guerra.**

**Ibarra – Ecuador**

**2016**



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

La **UNIVERSIDAD TÉCNICA DEL NORTE** dentro del proyecto **Repositorio Digital Institucional** determinó la necesidad de disponer textos completos en forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad. Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual se pone a disposición la siguiente información.

<b>DATOS DEL CONTACTO</b>	
Cédula de identidad	1003200696
Apellidos y nombres	Ruiz Nogales Vicente Javier
Dirección	Av. Juan De Albarracín y Amaguaña
E-mail	vicente_90ruiz@hotmail.com
Teléfono fijo	062922379
Teléfono móvil	0998921532
<b>DATOS DE LA OBRA</b>	
Título	MÓDULO DIDÁCTICO DE HARDWARE LIBRE PARA ANÁLISIS Y VERIFICACIÓN DE SISTEMAS ANÁLOGOS DIGITALES
Autor	Vicente Javier Ruiz Nogales
Fecha	2016/01/28
Programa	Pregrado
Título por el que opta	Ingeniero en ELECTRÓNICA Y REDES DE COMUNICACIÓN
Director	Ing. Milton López.

#### 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Vicente Javier Ruiz Nogales con cédula de identidad N° 1003200696, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar en forma digital y autorizo a la Universidad Técnica Del Norte, la publicación de la obra en Repositorio Digital Institucional y uso del archivo digital en la biblioteca de la Universidad con fines académicos para ampliar la disponibilidad de material y como apoyo a la educación,

investigación y extensión, en concordancia con la ley de Educación Superior Artículo 144.

### 3. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo sin violar derechos de autor de terceros, por lo tanto la obra es original, y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 28 días del mes enero de 2016

EL AUTOR:



-----  
VICENTE JAVIER RUIZ NOGALES



## UNIVERSIDAD TÉCNICA DEL NORTE

# CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, Vicente Javier Ruiz Nogales, con Cédula de identidad Nro. 1003200696, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado: MÓDULO DIDÁCTICO DE HARDWARE LIBRE PARA ANÁLISIS Y VERIFICACIÓN DE SISTEMAS ANÁLOGOS DIGITALES, que ha sido desarrollada para optar por el título de: Ingeniero en ELECTRÓNICA Y REDES DE COMUNICACIÓN en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte.

Ibarra, a los 28 días del mes enero de 2016

EL AUTOR:

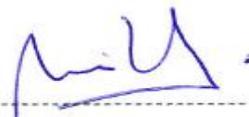
-----  
VICENTE JAVIER RUIZ NOGALES

1003200696

## CERTIFICACIÓN

Certifico que la Tesis previa a la obtención del título de Ingeniero en ELECTRÓNICA Y REDES DE COMUNICACIÓN con el tema: **MÓDULO DIDÁCTICO DE HARDWARE LIBRE PARA ANÁLISIS Y VERIFICACIÓN DE SISTEMAS ANÁLOGOS DIGITALES**, ha sido desarrollada y terminada en su totalidad por el Sr. Vicente Javier Ruiz Nogales, con cédula de identidad: 1003200696-6, bajo mi supervisión para lo cual firmo en constancia.

Atentamente,



-----  
Ing. Milton López.

DIRECTOR DE PROYECTO

## **AGRADECIMIENTO**

A DIOS que me ha concedido culminar esta meta, a mi Madre y Hermanos fuente de respaldo firme e incondicional en toda mi vida y más aún en mis años de carrera de pregrado, ya que sin su asistencia hubiera sido imposible culminar mi profesión, además que con su generosidad y simpatía me han tutelado por el camino apropiado.

## **DEDICATORIA**

A mi madre Mónica, ya que me ha respaldado completamente impulsándome a crecer en todos los aspectos, por sus lecciones, sus valores, por su motivación inquebrantable que me ha permitido conseguir la primera de muchas metas, pero más que nada, por su comprensión, devoción y amor.

A mis Hermanas y Hermanos, que persistentemente me han apoyado y guiado, en cada una de las etapas en el lapso de mi vida universitaria.

# ÍNDICE GENERAL

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....	iv
CERTIFICACIÓN.....	v
AGRADECIMIENTO .....	vi
DEDICATORIA .....	vii
ÍNDICE GENERAL .....	viii
ÍNDICE DE FIGURAS.....	xviii
ÍNDICE DE TABLAS.....	xxiv
GLOSARIO DE TÉRMINOS .....	xxvii
RESUMEN.....	xxviii
ABSTRACT.....	xxix
PRESENTACIÓN .....	xxx
1 Antecedentes .....	1
1.1 Tema .....	1
1.2 Problema.....	1
1.3 Objetivos .....	2
1.3.1 Objetivo General .....	2
1.3.2 Objetivos Específicos .....	2
1.4 Alcance.....	2
1.5 Justificación.....	4
1.6 Hardware Libre .....	5
1.6.1 Conceptualización .....	5
1.6.2 Clases .....	6

1.6.2.1	De acuerdo a su naturaleza.....	7
1.6.2.2	De acuerdo a su filosofía.....	7
1.6.3	Libertades.....	8
1.6.3.1	Plano Esquemático.....	8
1.6.3.2	Circuito Impreso (PCB).....	8
1.6.3.3	Fichero de fabricación .....	8
1.6.3.4	Restricciones y Lineamientos .....	9
1.6.4	Dispositivos basados en hardware libre .....	10
1.6.4.1	Proyecto RepRap .....	10
1.6.4.2	Raspberry Pi.....	11
1.6.4.3	Hardware libre en base a Arduino .....	11
1.6.4.4	CubieBoard.....	13
1.6.5	Aplicaciones con hardware libre .....	14
1.6.5.1	Monitoreo de sensores y control de actuadores .....	14
1.6.5.2	Cámaras reconfigurables de red .....	14
1.7	Conversión Análoga Digital .....	15
1.7.1	Muestreo .....	15
1.7.2	Cuantificación.....	16
1.7.3	Codificación.....	17
1.8	Sensores.....	18
1.8.1	Sensores de posición .....	18
1.8.1.1	Los captadores fotoeléctricos .....	18
1.8.2	Sensores de contacto.....	19
1.8.3	Captadores de circuitos oscilantes.....	19
1.8.4	Sensores de movimiento.....	20
1.8.4.1	Sensores de deslizamiento.....	20
1.8.4.2	Sensores de velocidad .....	20
1.9	Sensores y hardware libre.....	21
1.9.1	Sensores analógicos .....	21
1.9.1.1	Sensor de temperatura con elemento semiconductor LM35	22

1.9.1.2	Sensor de distancia sharp GP2Y02YK0F.....	22
1.9.1.2.1	Linealización de la señal de salida.....	23
1.9.1.3	Sensor de posición potenciómetro .....	24
1.9.1.4	Sensor de velocidad.....	25
2	Dimensionamiento Electrónico.....	27
2.1	Diagrama de bloques del módulo.....	27
2.2	Selección de los módulos de hardware libre .....	28
2.2.1	Selección del módulo de conversión de las señales analógicas en digitales.....	28
2.2.2	Selección del módulo para conversión de señales digitales en analógicas.....	31
2.3	Dimensionamiento de los sensores.....	32
2.3.1	Sensor de distancia.....	32
2.3.1.1	Recomendaciones de la óptica del sensor.....	35
2.3.1.2	Recomendaciones de las características .....	35
2.3.2	Sensor de temperatura.....	36
2.3.3	Sensor de posición angular.....	38
2.3.4	Sensor de velocidad angular.....	39
2.3.4.1	Ventilador PWM.....	39
2.3.4.1.1	Ejemplo de un ventilador de 4 hilos .....	40
2.3.4.1.2	Ventilador de 4 pines PWM .....	41
2.4	Selección del lenguaje de programación.....	48
2.4.1	Lenguaje c.....	48
2.4.2	Lenguaje c++.....	49
2.4.3	Lenguaje java .....	49
2.4.4	Lenguaje python.....	50
2.4.5	Lenguaje de programación seleccionado. Python.....	50
2.4.5.1	Reglas sintácticas básicas del lenguaje Python y una comparación simplificada con C y lenguajes Java.....	52
2.4.5.1.1	Sangría .....	53

2.4.5.2	El código mínimo Python .....	53
2.4.5.3	Variables en Python .....	54
2.4.5.4	Cadenas de caracteres, funciones, clases y bucles .....	55
2.4.5.5	Uso de módulos de Python.....	56
2.4.5.6	Ejemplo de un código Python tipo estructurado .....	57
2.5	Programa para desarrollar la interfaz gráfica de usuario.....	57
2.5.1	Qt .....	57
2.5.1.1	Qt Designer .....	58
2.5.1.1.1	Descubriendo la interfaz Qt Designer .....	58
2.5.1.1.2	Barra de menús .....	59
2.5.1.1.3	Barra de herramientas .....	59
2.5.1.1.4	Barra de Widgets .....	60
2.5.1.1.5	Inspector de objetos .....	60
2.5.1.1.6	Editor de propiedades.....	60
2.5.1.1.7	Editor de señales .....	61
2.5.1.2	PyQt .....	62
2.5.1.3	Pyduino.....	62
3	Implementación del Módulo Didáctico .....	66
3.1	Pcduino primeros pasos .....	66
3.1.1	Arduino IDE en el Pcduino .....	67
3.1.2	Verificación de los módulos de Pcduino .....	68
3.1.3	Verificación y puesta en marcha de Python en el Pcduino .....	69
3.1.4	Instalación de Pyduino en nuestro Pcduino.....	70
3.1.5	Instalación de interfaz gráfica de usuario, GUI (graphical user interface).....	71
3.1.6	Instalaciones de paquetes complementarios.....	73
3.1.7	Resumen de los programas más significativos a utilizar .....	74
3.2	Aplicaciones de interfaz de usuario en QT-Designer .....	75
3.2.1	Generación del archivo Python de la interfaz gráfica de usuario (GUI) .....	77

3.3	Control centralizado para el sistema en tiempo real de adquisición de datos de las magnitudes físicas temperatura, distancia, posición, velocidad angular. ....	77
3.3.1	Formas de adquisición y verificación de los estados de los pines analógicos y de E/S .....	78
3.3.1.1	Medición de las entradas analógicas del pcDuino.....	78
3.3.1.2	Acceso a los pines de E/S.....	80
3.4	Sistema de adquisición de datos de los sensores de magnitudes físicas: temperatura, distancia, posición, velocidad angular.....	83
3.4.1	Adquisición de los valores de los sensores analógicos .....	84
3.4.2	Diagrama de flujo de la realización de las gráficas de los valores obtenidos de las entradas analógicas.....	86
3.5	Procedimientos a seguir para crear los programas destinados al análisis de las magnitudes físicas: temperatura, distancia, posición, velocidad angular. ....	87
3.5.1	Procedimientos para la creación de interfaces graficas de usuarios para las magnitudes físicas: temperatura, distancia, posición, velocidad angular.....	87
3.5.1.1	Descripción general de las Interfaces gráficas de usuario ..	89
3.5.2	Proceso de adquisición de la señal proveniente de las entradas analógicas. ....	92
3.5.2.1	El proceso de graficación de la señal proveniente de los sensores analógicos. ....	93
3.5.3	Muestreo de la señal proveniente de las entradas analógicas (A [2, 3, 4, 5]). ....	94
3.5.3.1	Visualización del muestreo de las señales provenientes de las entradas analógicas. ....	96
3.5.4	Cuantificación de las señales provenientes de las entradas analógicas. ....	97
3.5.4.1	Visualización de la cuantificación de las señales provenientes de las entradas analógicas. ....	98

3.5.5	Codificación de las señales provenientes de los sensores analógicos. ....	99
3.5.6	Circuito de activación desactivación de los módulos de sensores. ....	100
3.6	Creación del programa destinado al análisis de la magnitud temperatura.....	101
3.6.1	Interfaz gráfica de usuario .....	101
3.6.2	Diagramas de conexión del sensor de temperatura Lm35 ....	102
3.6.3	Descripción de funcionamiento del módulo de temperatura..	104
3.6.4	Proceso de adquisición de la señal proveniente del sensor lm35 .....	104
3.6.5	Muestreo de la señal proveniente del sensor de temperatura.....	104
3.6.6	Cuantificación de la señal proveniente del sensor de temperatura. ....	106
3.6.7	Codificación de la señal proveniente del sensor de temperatura. ....	106
3.7	Creación del programa destinado al análisis de la magnitud distancia. ....	107
3.7.1	Interfaz gráfica de usuario de la magnitud distancia. ....	107
3.7.2	Diagrama de conexión del sensor de distancia. ....	108
3.7.3	Descripción de funcionamiento del módulo de distancia.....	109
3.7.4	Proceso de adquisición de la señal proveniente del sensor Sharp .....	110
3.7.5	Codificación de la señal proveniente del sensor de distancia. ....	110
3.8	Creación del programa destinado al análisis de la magnitud posición. ....	111
3.8.1	Interfaz gráfica de usuario de la magnitud posición. ....	112

3.8.2	Diagrama de conexión del sensor de posición.....	112
3.8.3	Descripción de funcionamiento del módulo de posición.....	114
3.8.4	Codificación de la señal proveniente del sensor de posición.	114
3.9	Creación del programa destinado al análisis de la magnitud velocidad angular.....	115
3.9.1	Interfaz gráfica de usuario de la magnitud velocidad angular.	115
3.9.2	Diagrama de conexión del sensor de velocidad angular.....	116
3.9.3	Descripción de funcionamiento del Módulo de velocidad.....	118
3.9.4	Codificación de la señal proveniente del sensor de velocidad angular.....	119
3.10	Conversión digital a analógica.....	119
3.10.1	Red R2R.....	120
3.10.1.1	Interfaz gráfica del módulo R2R.....	121
3.10.2	Pwm.....	123
3.10.2.1	Interfaz gráfica del módulo de PWM.....	124
3.10.3	SPI.....	126
3.10.3.1	Interfaz gráfica de matriz de leds.....	127
3.11	Proceso de ensamblaje del módulo didáctico.....	128
3.11.1	Dimensiones del módulo didáctico.....	128
3.11.2	Posicionamiento de los módulos de hardware libre.....	129
3.11.3	Posicionamiento de los sensores en el módulo.....	130
3.11.4	Fuente de Alimentación.....	132
3.11.5	Pantalla para visualización de la interfaz gráfica.....	133
3.11.6	Ensamblaje total del módulo didáctico.....	134
3.11.6.1	Dispositivos y materiales necesarios.....	134
3.12	Análisis económico del módulo didáctico.....	135
3.12.1	Materiales para cada uno de los módulos de sensores.....	136
3.12.1.1	Costo módulo sensor de temperatura.....	136

3.12.1.2	Costo módulo sensor de posición.....	137
3.12.1.3	Costo módulo sensor de distancia.....	137
3.12.1.4	Costo módulo sensor de velocidad.....	138
3.12.2	Costo del software utilizado en el módulo didáctico .....	138
3.12.3	Costo del desarrollo (Investigación, diseño, programación y pruebas) del módulo didáctico .....	139
3.12.4	Costo total del módulo didáctico.....	139
3.12.5	Análisis beneficios .....	140
4	Desarrollo de Guía de Prácticas. ....	141
4.1	Esquematización de prácticas.....	141
4.2	Práctica # 1: Verificación de la señal proveniente del sensor de la magnitud temperatura. ....	149
4.2.1	Objetivo .....	149
4.2.2	Equipo Requerido.....	149
4.2.3	Prerrequisitos de la práctica .....	150
4.2.4	Procedimiento .....	150
4.2.5	Cuestionario .....	154
4.2.6	Funcionamiento y Conclusiones.....	155
4.2.7	Actividades Propuestas .....	155
4.3	Práctica # 2: Verificación de la señal proveniente del sensor de la magnitud distancia. ....	157
4.3.1	Objetivo .....	157
4.3.2	Equipo Requerido.....	157
4.3.3	Prerrequisitos de la práctica .....	158
4.3.4	Procedimiento .....	158
4.3.5	Cuestionario .....	162
4.3.6	Funcionamiento y Conclusiones.....	163
4.3.7	Actividades Propuestas .....	163

4.4	Práctica # 3: Verificación de la señal proveniente del sensor de la magnitud posición. ....	165
4.4.1	Objetivo .....	165
4.4.2	Equipo Requerido.....	165
4.4.3	Prerrequisitos de la práctica .....	166
4.4.4	Procedimiento .....	166
4.4.5	Cuestionario .....	170
4.4.6	Funcionamiento y Conclusiones.....	170
4.4.7	Actividades Propuestas .....	170
4.5	Práctica # 4: Verificación de la señal proveniente del sensor de la magnitud velocidad angular. ....	172
4.5.1	Objetivo .....	172
4.5.2	Equipo Requerido.....	172
4.5.3	Prerrequisitos de la práctica .....	173
4.5.4	Procedimiento .....	173
4.5.5	Cuestionario .....	176
4.5.6	Funcionamiento y Conclusiones.....	177
4.5.7	Actividades Propuestas .....	177
4.6	Práctica # 5: Verificación del funcionamiento de conversión digital analógica.....	179
4.6.1	Objetivo .....	179
4.6.2	Equipo Requerido.....	179
4.6.3	Prerrequisitos de la práctica .....	180
4.6.4	Procedimiento .....	180
4.6.5	Cuestionario .....	186
4.6.6	Funcionamiento y Conclusiones.....	186
4.6.7	Actividades Propuestas .....	187

4.7 Práctica # 6: Verificación del funcionamiento del módulo didáctico en su totalidad.....	188
4.7.1 Objetivo .....	188
4.7.2 Equipo Requerido.....	188
4.7.3 Prerrequisitos de la práctica .....	189
4.7.4 Procedimiento .....	190
4.7.5 Cuestionario .....	193
4.7.6 Funcionamiento y Conclusiones.....	193
4.7.7 Actividades Propuestas .....	194
5 Conclusiones y Recomendaciones .....	195
Bibliografía.....	199
ANEXOS.....	203

## ÍNDICE DE FIGURAS

Figura 1 Placa de hardware libre pcduino versión 3. ....	6
Figura 2 Módulo ensamblado en el proyecto RepRap. ....	10
Figura 3 Microcomputador distribuido por la Fundación Raspberry Pi. ....	11
Figura 4 Placa PcDuino (Mini Pc y Arduino). ....	12
Figura 5 Mini Ordenador distribuido por Cubietech Limited. ....	13
Figura 6 Proceso de conversión de señales analógicas en digitales. ....	15
Figura 7 Muestras obtenidas a partir de una señal original. ....	16
Figura 8 Cuantificación de una señal muestreada. ....	17
Figura 9 Codificación de una señal cuantificada. ....	18
Figura 10 Sensor con interfaz gráfica en PC-duino. ....	21
Figura 11 Sensor Sharp de la familia GP2Y. ....	23
Figura 12 Linealización de una señal. ....	23
Figura 13 Potenciómetro de precisión. ....	24
Figura 14 Diagrama de bloques del módulo didáctico de hardware libre. .	28
Figura 15 Pc-duino versión 3 distribuido por SparkFun. ....	30
Figura 16 Placa de hardware libre Cubie-truck distribuido por Cubie-tech.	32
Figura 17 Sensor infrarrojo de distancia Sharp GP2Y02YK0F. ....	33
Figura 18 Diagrama de Bloques del sensor Sharp GP2Y02YK0F. ....	34
Figura 19 Curva característica del sensor Sharp seleccionado. ....	34
Figura 20 Sensor de temperatura lm-35. ....	37
Figura 21 Curva característica de temperatura LM35. ....	37
Figura 22 Potenciómetro de precisión. ....	39
Figura 23 Ventilador NIDEC F09A-12B3S1. ....	40
Figura 24 Circuitería Ventilador NIDEC F09A-12B3S1. ....	40
Figura 25 Control PWM ventilador de 4 Hilos. ....	42
Figura 26 Diagrama en bloques y Distribución de pines del LM2917. ....	44
Figura 27 Diagrama en bloques LM2917. ....	45
Figura 28 Ejemplo de conexión del LM2917. ....	47
Figura 29 Logotipo del lenguaje de programación Python. ....	51
Figura 30 Ejemplo de aplicación de código con Python. ....	57
Figura 31 Zonas funcionales de Qt-Designer. ....	61
Figura 32 Consola de pcduino con el comando sudo board-config.sh. ....	66

Figura 33 Consola de Pduino con las opciones de configuración del dispositivo.....	66
Figura 34 Consola de Pduino con las actualizaciones de los paquetes del dispositivo.....	67
Figura 35 Ejemplo de ejecución de Arduino IDE en el Pduino.....	68
Figura 36 Formas de verificar el funcionamiento de los módulos del Pduino. ....	69
Figura 37 Consola de Pduino ejecutando Python.....	69
Figura 38 Archivos que contiene Python-pduino.....	70
Figura 39 Ventana Principal del IDE Pyduino.....	71
Figura 40 Consola con comando para instalar e inicializar Qt-Designer....	72
Figura 41 Ventana principal de Qt-Designer.....	72
Figura 42 Consola con los comandos de instalación de paquetes complementarios. ....	73
Figura 43 Consola con el comando de instalar pyqtgraph.....	73
Figura 44 Diagrama de flujo del proceso de creación de interfaces graficas de usuario.....	75
Figura 45 Menú principal de la aplicación del trabajo de grado.....	76
Figura 46 Control centralizado de adquisición de datos en tiempo real.....	78
Figura 47 Forma simplificada y forma cruda de obtención de los valores de pines de las entradas analógicas.....	79
Figura 48 Obtención de los valores de las entradas analógicas del pduino. ....	80
Figura 49 Consola con el directorio de los pines E/S. ....	81
Figura 50 Versión simplificada y versión cruda de accesos a los valores de los pines de E/S.....	82
Figura 51 Sistema de adquisición de datos de sensores.....	83
Figura 52 Diagrama esquemático de los sensores analógicos.....	85
Figura 53 Diagrama de flujo para graficación de las entradas analógicas.	86
Figura 54 Flujograma de creación de las interfaces graficas de usuario. ..	88
Figura 55 Diagrama generalizado de las interfaces graficas de usuario. ..	91
Figura 56 Interfaz gráfica de usuario de las magnitudes físicas (Posición). ....	91

Figura 57 Formación de la señal muestreada a partir de la señal de los sensores. ....	94
Figura 58 Señal muestreada.....	96
Figura 59 Proceso de cuantificación.....	98
Figura 60 Circuito de activación desactivación de los módulos de sensores. ....	100
Figura 61 Interfaz gráfica de usuario de la magnitud temperatura.....	102
Figura 62 Conexión del sensor de temperatura Lm35 en la entrada analógica A5 del pduino. ....	102
Figura 63 Diagrama pictórico del sensor de temperatura. ....	103
Figura 64 Diagrama del circuito impreso del sensor de temperatura.....	103
Figura 65 Interfaz gráfica de la magnitud distancia. ....	107
Figura 66 Diagrama esquemático del sensor de distancia. ....	108
Figura 67 Diagrama pictórico del sensor de distancia. ....	108
Figura 68 Diagrama del circuito impreso del sensor de distancia.....	109
Figura 69 Valores de voltaje en relación a la distancia sensor Sharp.....	111
Figura 70 Interfaz gráfica de la magnitud posición. ....	112
Figura 71 Diagrama esquemático del sensor de posición. ....	112
Figura 72 Diagrama pictórico del sensor de posición. ....	113
Figura 73 Diagrama circuito impreso del sensor de posición.....	113
Figura 74 Interfaz gráfica de la magnitud velocidad angular.....	116
Figura 75 Diagrama esquemático del sensor de velocidad. ....	117
Figura 76 Diagrama pictórico del sensor de velocidad. ....	117
Figura 77 Diagrama circuito impreso del sensor de velocidad.....	118
Figura 78 Valores de voltaje en relación a rpm.....	119
Figura 79 Circuito R2R conversión digital analógico. ....	121
Figura 80 Interfaz gráfica de R2R.....	122
Figura 81 Graficación del valor de R2R.....	122
Figura 82 Ilustración de onda PWM.....	123
Figura 83 Ciclo de trabajo señal PWM.....	123
Figura 84 Interfaz gráfica de PWM. ....	124
Figura 85 Graficación de la señal PWM.....	124
Figura 86 Configuración del bus SPI. ....	126

Figura 87 Interfaz gráfica de matriz de Leds.....	127
Figura 88 Matriz de Leds. ....	128
Figura 89 Estructura del módulo didáctico.....	129
Figura 90 Montaje y Cableado de las entradas analógicas del Pcdiuno..	129
Figura 91 Montaje del cubietruck.....	130
Figura 92 Posicionamiento de los módulos de los sensores. ....	130
Figura 93 Cableado de los sensores del Pcdiuno.....	131
Figura 94 Ensamblaje total del módulo didáctico.....	135
Figura 95 Distribución de los elementos de sensores. ....	143
Figura 96 Distribución general de los elementos del módulo didáctico. ...	143
Figura 97 Etiquetado general de los elementos del módulo didáctico. ....	144
Figura 98 Interfaz gráfica de usuario general para las magnitudes físicas. .....	145
Figura 99 Botón OFF/ON Módulo. ....	145
Figura 100 Sección botones de inicio. ....	145
Figura 101 Botón parar graficación.....	146
Figura 102 Botón borrar graficación. ....	146
Figura 103 Sección de botones ADC.....	146
Figura 104 Sección valores digitales. ....	147
Figura 105 Sección de señales.....	147
Figura 106 Sección zoon señales.....	148
Figura 107 Sección opciones.....	148
Figura 108 Diagrama para verificación de conexión modulo temperatura. .....	151
Figura 109 Botón OFF/ON módulo de temperatura.....	151
Figura 110 Botón muestreo del módulo de temperatura.....	151
Figura 111 Señal muestreada, módulo de temperatura.....	152
Figura 112 Sección de valores digitales, módulo de temperatura. ....	152
Figura 113 Sección de alarma temperatura.....	153
Figura 114 Interfaz gráfica de usuario del módulo de temperatura, en ejecución.....	153
Figura 115 Interfaz gráfica de la práctica #1.....	154
Figura 116 Diagrama para verificación de conexión modulo distancia. ...	159

Figura 117 Interfaz gráfica de usuario de distancia. ....	159
Figura 118 Botón OFF/ON, módulo de distancia. ....	160
Figura 119 Rango de medición de módulo de distancia. ....	160
Figura 120 Proceso de cuantificación, módulo de distancia. ....	161
Figura 121 Práctica #2, módulo de distancia. ....	161
Figura 122 Interfaz gráfica de distancia en ejecución. ....	162
Figura 123 Diagrama para verificación de conexión modulo posición. ....	167
Figura 124 Interfaz gráfica de usuario de posición. ....	167
Figura 125 Botón OFF/ON, módulo de posición. ....	168
Figura 126 Proceso de codificación, módulo de posición. ....	168
Figura 127 Práctica #3, módulo de posición. ....	169
Figura 128 Interfaz gráfica de posición en ejecución. ....	169
Figura 129 Diagrama para verificación de conexión modulo velocidad. ...	174
Figura 130 Interfaz gráfica de usuario de velocidad. ....	174
Figura 131 Botón OFF/ON, módulo de velocidad. ....	175
Figura 132 Práctica #4, módulo de velocidad. ....	175
Figura 133 Interfaz gráfica de velocidad en ejecución. ....	176
Figura 134 Diagrama para verificación de conexión módulo R2R. ....	181
Figura 135 Interfaz gráfica de usuario de R2R. ....	181
Figura 136 Interfaz gráfica de usuario de R2R en funcionamiento. ....	182
Figura 137 Diagrama para verificación de conexión módulo ventilador...	182
Figura 138 Interfaz gráfica de usuario de PWM. ....	183
Figura 139 Interfaz gráfica de usuario de PWM en funcionamiento. ....	183
Figura 140 Diagrama para verificación de conexión módulo Matriz de Leds. .....	184
Figura 141 Interfaz gráfica de usuario de Matriz de Leds. ....	184
Figura 142 Interfaz gráfica de usuario de Matriz de Leds en funcionamiento. .....	185
Figura 143 Interfaz gráfica de usuario de práctica # 5. ....	185
Figura 144 Interfaz gráfica de la práctica general. ....	190
Figura 145 Botón OFF/ON, módulos. ....	191
Figura 146 Botones de acceso a graficación de la práctica general. ....	191
Figura 147 Interfaz gráfica de la práctica general en funcionamiento. ....	192

Figura 148 Práctica #6, practica general del módulo didáctico..... 192

## ÍNDICE DE TABLAS

Tabla 1 Comparativo entre Pc-duino 3 y raspberrypi.....	29
Tabla 2 Características de dispositivo CubieTruck. ....	31
Tabla 3 Características del sensor infrarrojo de distancia Sharp GP2Y02YK0F. ....	33
Tabla 4 Características del sensor lm-35.....	36
Tabla 5 Características del dispositivo para medición de posición angular	38
Tabla 6 PinOut Ventilador.....	39
Tabla 7 Características del ventilador.....	41
Tabla 8 Tabla comparativa de sintaxis.....	53
Tabla 9 Código elemental de ejecución en phyton. ....	54
Tabla 10 Variables utilizadas en lenguaje Python. ....	54
Tabla 11 Cadenas de caracteres, funciones, clases y condiciones en Python. .....	55
Tabla 12 Manera de importa los módulos en el lenguaje de programación Python.....	56
Tabla 13 Tabla de funciones más comunes utilizadas de Pyduino.....	64
Tabla 14 Comandos para descargar IDE Pyduino.....	71
Tabla 15 Comandos de descarga e instalación de Pyqtgraph.....	74
Tabla 16 Resumen de software a utilizar.....	74
Tabla 17 Variables de almacenamiento de los valores de las entradas analógicas.....	92
Tabla 18 Array de dos dimensiones. ....	92
Tabla 19 Método para que los valores se actualicen de forma automática. .....	92
Tabla 20 Código para Inicializar el grafico de la señal de las entradas analógicas.....	93
Tabla 21 Código para asignar valores predeterminados para la visualización en los ejes x, y. ....	93
Tabla 22 Código para inicializar la graficación de los valores.....	94
Tabla 23 Código para graficar el muestreo.....	97
Tabla 24 Código para graficar el cuantificación. ....	99

Tabla 25 Graficación del muestreo de la señal proveniente del sensor de temperatura. ....	105
Tabla 26 Graficación de cuantificación de la señal proveniente del sensor de temperatura. ....	106
Tabla 27 Codificación de la señal proveniente del sensor de temperatura. ....	107
Tabla 28 Valores obtenidos del sensor del Sharp GP2Y02YK0F. ....	110
Tabla 29 Codificación de la señal proveniente del sensor de posición....	114
Tabla 30 Directorios de pines GPIO Cubietruck. ....	122
Tabla 31 Ficheros de PWM Cubietruck. ....	125
Tabla 32 Código del Método de cambiar el valor de ciclo útil de PWM. ...	125
Tabla 33 Código de graficación de la señal PWM. ....	125
Tabla 34 Código del Botón OK módulo matriz de leds. ....	127
Tabla 35 Tabla de dimensiones del módulo didáctico. ....	128
Tabla 36 Tabla de colores de cables utilizados para los sensores. ....	131
Tabla 37 Cables utilizados para las GPIO del Pcdduino y entradas analógicas asignadas para los sensores. ....	131
Tabla 38 Especificaciones fuente de alimentación. ....	132
Tabla 39 Estimación de consumo de corriente de los módulos de sensores. ....	132
Tabla 40 Características de la pantalla.....	133
Tabla 41 Dispositivos y materiales del módulo didáctico. ....	134
Tabla 42 Costo de dispositivos y materiales del módulo didáctico. ....	135
Tabla 43 Costo módulo sensor de temperatura.....	136
Tabla 44 Costo módulo sensor de posición.....	137
Tabla 45 Costo módulo sensor de distancia.....	137
Tabla 46 Costo módulo sensor de velocidad.....	138
Tabla 47 Costo del software utilizado en el módulo didáctico.....	138
Tabla 48 Costo del desarrollo. ....	139
Tabla 49 Costo total del módulo didáctico. ....	139

## ÍNDICE DE ECUACIONES

Ecuación 1 Ecuación de salida de voltaje del LM2917. ....	45
Ecuación 2 Ecuación de cálculo de R1 del LM2917. ....	46
Ecuación 3 Ecuación de cálculo de C1 del LM2917. ....	46
Ecuación 4 Ecuación de cálculo de I2 del LM2917. ....	46
Ecuación 5 Ecuación de cálculo de C2 del LM2917. ....	47
Ecuación 6 Función delta de Dirac. ....	94
Ecuación 7 Primer instante de muestreo. ....	95
Ecuación 8 Instantes de muestreo k+1. ....	95
Ecuación 9 Función de salida del muestreador. ....	95
Ecuación 10 Sumatoria de secuencia de salida del muestreador. ....	96
Ecuación 11 Señal antes del mantenedor de orden cero. ....	98
Ecuación 12 Señales después del mantenedor de orden cero. ....	98
Ecuación 13 Función potencial del sensor Sharp. ....	111
Ecuación 14 Función lineal de Voltaje - RPM. ....	119
Ecuación 15 Ecuación de R2R. ....	121

## GLOSARIO DE TÉRMINOS

PC-DUINO.- Mini Ordenador, mini Pc y Arduino.

CUBIETRUCK.- Mini Ordenador, distribuido por Cubietech Limited.

ARDUINO.- Plataforma de prototipos electrónicos de código abierto basada en hardware y software flexibles.

PCB. - Printed Circuit Board. Circuitos impresos.

GPL.- General Public License. Licencia Pública General.

OSI. - Open System Interconnection. Modelo de referencia de Interconexión de Sistemas Abiertos.

GNU. - GNU's Not Unix. GNU no es UNIX.

RAM. - Random Access Memory. Memoria de acceso aleatorio.

NTC.- Negative Temperature Coefficient. Resistencia con coeficiente de temperatura negativo.

LDR.- Light Dependent Resistor. Resistor Dependiente de la Luz.

A/D.- Analog-to-Digital. Conversor Analógico Digital.

D/A.- Digital-to- Analog. Conversor Digital Analógico.

HMTL.- HyperText Markup Language. Lenguaje de Marcado de Hipertexto.

GPIO. - General Purpose Input/Output. Entrada/Salida de Propósito General.

IDE.- Integrated development environment. Entorno de desarrollo integrado.

ARM. - Advanced RISC Machines. Microcontroladores ARM.

RPMS.- Revoluciones Por Minuto: Es la cantidad de vueltas que un cuerpo giratorio completa alrededor de su eje cada sesenta segundos.

PYTHON.- Lenguaje de programación de alto nivel, interpretado y multipropósito.

## RESUMEN

Se ha implementado un módulo didáctico de hardware libre para análisis y verificación de sistemas análogos digitales con la finalidad de suministrar un instrumento de aprendizaje práctico a los estudiantes que cursan la carrera de ingeniería en electrónica y redes de comunicación. El equipo didáctico desarrollado consta de tres sistemas primordiales: Sistema de adquisición de datos, compuesto por cuatro sensores que determinan las magnitudes físicas de temperatura, distancia, posición y velocidad angular; y dos placas de hardware libre como elementos principales de este sistema, los cuales en conjunto establecen la lectura de dichas magnitudes para procesarlas posteriormente. Sistema de conversión Análoga a digital, el cual se encarga de transformar todas las señales analógicas en digitales evidenciando todas y cada una de las etapas que este proceso demanda, para lo cual utiliza Python como plataforma para lograr estas acciones. La interfaz de usuario, está compuesto por una serie de pantallas interactivas que permiten acceder de manera gráfica a cada una de las etapas que intervienen del proceso de conversión de señales digitales en analógicas, tales como muestreo cuantificación, codificación y presentación visual de cada una de las magnitudes antes descritas. El módulo realizado contiene todos los dispositivos que se requieren para la realización de prácticas referentes a las etapas comprendidas en conversión análoga a digital, de tal manera que el estudiante pueda ir abordándolas etapa por etapa, hasta alcanzar las habilidades necesarias para manipular el módulo en su totalidad.

## **ABSTRACT**

I implemented a training module for free hardware analysis and verification of digital modeling systems in order to provide a practical learning tool for students who are studying engineering in electronics and communication networks. The training module has developed three major systems: Data acquisition, comprising four sensors that determine the physical quantities of temperature, distance, position and angular velocity; and two sheets of free hardware main elements of this system, which together establish the reading of these magnitudes for processing later. System Analog to digital conversion, which is responsible for transforming all analogue signals to digital evidencing each and every one of the steps necessary for this process, for which it uses as Python platform to achieve these actions. Graphical user interface: it consists of interactive series displays which allow access graphically at each of the stages involved in the process of converting digital signals to analog, such as sampling quantization, coding and display of each of the magnitudes described above. The module contains all devices made required for the experiments concerning the steps involved in analog to digital conversion, so that the student can go step by step approaching them, to reach the necessary skills to handle the module in whole.

# PRESENTACIÓN

Con este proyecto se brinda una herramienta versátil para la realización de prácticas referentes a conversión de señales analógicas en digitales, consolidando los conocimientos teóricos con la experiencia práctica.

En el **primer capítulo** se especifica todos los conceptos, compendios y fundamentos requeridos para la realización de todo el proyecto, abordando desde una descripción general de los aspectos mas distinguidos, hasta llegar a los elementos constitutivos que se demandan para conformar todo el sistema.

En el **segundo capítulo** se detalla todo lo referente a hardware libre, así como el dimensionamiento de los módulos físicos con los cuales se implementa el módulo. También describe el dimensionamiento de los dispositivos electrónicos complementarios, que conjuntamente con los módulos de hardware libre, determinan las prestaciones académicas y versatilidad de este proyecto.

En el **tercer capítulo** se trata sobre los procedimientos a seguir para la implementación del módulo en su totalidad, así como la estructuración del sistema de comunicación entre los diferentes módulos de hardware libre.

En el **cuarto capítulo** se describe las prácticas que se constituyen por diferentes ejercicios de montaje en hardware que incrementan su nivel de complejidad a medida que se avanza, las cuales se desarrollan a través de actividades manuales de montaje y manipulación de hardware: sensores, componentes activos/pasivos, Matriz de Leds, entre otros.

En el **quinto capítulo** se describe las conclusiones y recomendaciones respecto a implementación de todo el proyecto.

# 1 ANTECEDENTES

En este capítulo se detalla todo lo referente a los precedentes necesarios para la realización del presente trabajo de grado, abordándolos primeramente desde una visión global, hasta su detalle en cada uno de los temas necesarios intervenidos en este proyecto.

## 1.1 TEMA

“Módulo didáctico de hardware libre para análisis y verificación de sistemas análogos digitales.”

## 1.2 PROBLEMA

Para la educación y formación de una persona se hace indispensable el manejo de métodos que faciliten la comprensión y análisis de tipo teórico-práctico; circunstancia que ha tenido muy en cuenta la carrera de ingeniería en Electrónica y Redes de Comunicación durante todos los años de existencia.

Al mejorar cada día la calidad de educación, la carrera de ingeniería en Electrónica y Redes de Comunicación cuenta con muchas herramientas que facilitan el aprendizaje de los alumnos y la enseñanza de los docentes, sin embargo, existen áreas en las cuales se hace indispensable el manejo de equipos didácticos que simulen los procesos de Sistemas Analógicos Digitales y Sistemas Digitales Análogos.

Por lo que es necesario un módulo didáctico de Hardware Libre para la FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS que facilite el desarrollo de las prácticas de laboratorio de forma integrada, para lograr un claro entendimiento de varios de los conceptos aplicados en la asignatura de Sistemas Analógicos Digitales, y poder relacionarlos con procesos de

digitalización de señales, tales como el muestreo, cuantificación y codificación de datos.

## **1.3 OBJETIVOS**

### **1.3.1 OBJETIVO GENERAL**

Implementar un módulo didáctico de hardware libre que permita profundizar los conceptos desarrollados en la asignatura Sistemas Analógicos Digitales; contribuyendo a fortalecer el aprendizaje práctico del estudiante en laboratorio.

### **1.3.2 OBJETIVOS ESPECÍFICOS**

- Establecer los principios y fundamentos teóricos necesarios para la realización del módulo didáctico.
- Seleccionar los módulos principales de hardware libre a utilizarse.
- Dimensionar los elementos Eléctricos y Electrónicos los cuales van a ser controlados por los módulos de hardware libre.
- Desarrollar una guía de prácticas para facilitar la utilización del módulo que se presenta.

## **1.4 ALCANCE**

El módulo que se detalla constara de un grupo de sensores de medición de magnitudes físicas tales como temperatura, distancia, posición y velocidad, los cuales presentaran en su salida señales analógicas las cuales al ingresar al módulo de hardware libre serán procesadas hasta convertirlas en señales digitales.

El análisis de la temperatura será posible al monitorizar esta magnitud mediante un sensor de tipo semiconductor el cual entregara una señal de voltaje

proporcional a la temperatura medida, este voltaje que es analógico puede ser transferido a señales digitales pasando por todos los procesos que se requieren para este fin.

Para el caso de la medición y codificación digital de distancia a objetos se dispondrá de un sensor de proximidad infrarrojo, el cual proporcionará una señal analógica que a su vez deberá ser primeramente linealizada para luego ser convertida en digital, mostrándose todo el proceso de las señales de forma gráfica en la pantalla del módulo.

El proceso de especificación de posición será posible gracias a la utilización de un transductor de posición angular, de tipo absoluto, variación lineal y con salida de tipo analógico, para provocar la demostración de conversión A/D de este tipo de señales que denotaran posiciones angulares por variación de resistencia eléctrica.

Los procesos secuenciales de conversión de señales para obtener la velocidad angular de un objeto será posible mediante la utilización del principio de medición analógica de la corriente producida por una bobina y que es proporcional a la velocidad del movimiento. Todo esto representado gráficamente y verificado mediante la aplicación de los conceptos concernientes a la asignatura de sistemas análogos digitales.

Las placas de hardware libre se encargaran tanto de los procesos de conversión analógica digital, mediante programación de los convertidores analógico-digitales, entradas de captura y cuenta de pulsos; así como también del proceso de conversión digital análogo, utilizando las líneas de entradas y salidas digitales que pueden recibir señales que entregan los acondicionares ya sean estos por puente de resistencias, transistores o amplificadores operacionales.

El proceso de digitalización de las señales se podrá monitorizar a través de una pantalla, mostrándose en esta las señales antes de ser convertidas y también una vez que se hayan digitalizado, además de los procesos tales como: muestreo, cuantificación y codificación que demandan la conversión de señales analógicas en digitales y digitales en analógicas.

Se realizara un manual de prácticas que el alumno ira complementando de manera secuencial, hasta obtener la destreza para manipular el módulo en su totalidad.

Una vez concluida la implementación del proyecto se procederá a la donación del mismo al laboratorio de Electrónica y Redes de Comunicación de la FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS de la UNIVERSIDAD TÉCNICA DEL NORTE.

## **1.5 JUSTIFICACIÓN**

La formación en Electrónica y Redes de Comunicación debe abarcar una fuerte componente práctica en el estudio de la asignatura Sistemas Analógicos Digitales, para así permitir al estudiante desarrollar una visión global de la tecnología de electrónica análoga/digital básica en todo proceso que actualmente se está utilizando.

El conocimiento práctico de estas tecnologías es de vital importancia para el Ingeniero Electrónico y Redes de Comunicación, ya que estas son parte primordial de una gran mayoría de procesos de comunicación existentes. Los elementos y dispositivos utilizados en módulos de aprendizaje didáctico (sensores, módulos de hardware libre, actuadores, LCD, etc.) son similares a los que utilizan las empresas que prestan servicios de comunicaciones, lo cual permite al alumno trabajar con equipos reales, resistentes y similares a los que encontrará en su quehacer profesional.

Es necesario que el laboratorio de la FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS facilite y capacite en este tipo de tecnologías, y más específicamente sobre simulaciones de procesos que demanden la utilización de hardware libre, lográndose con esto desarrollar destrezas a nivel práctico y obtener una mejor formación profesional lo que conlleva a la generación de un contingente humano con capacidad de desenvolverse, proyectarse y marchar acorde a los nuevos retos tecnológicos.

Con la realización de este módulo didáctico se busca complementar el laboratorio de Electrónica y Redes de Comunicación de la FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS facilitando la implementación de prácticas que ayuden a los estudiantes a visualizar y comprender de mejor manera los factores y dispositivos involucrados en procesos de conversión análogo/digital y digital/análogo que demanden la utilización de hardware libre.

## **1.6 HARDWARE LIBRE**

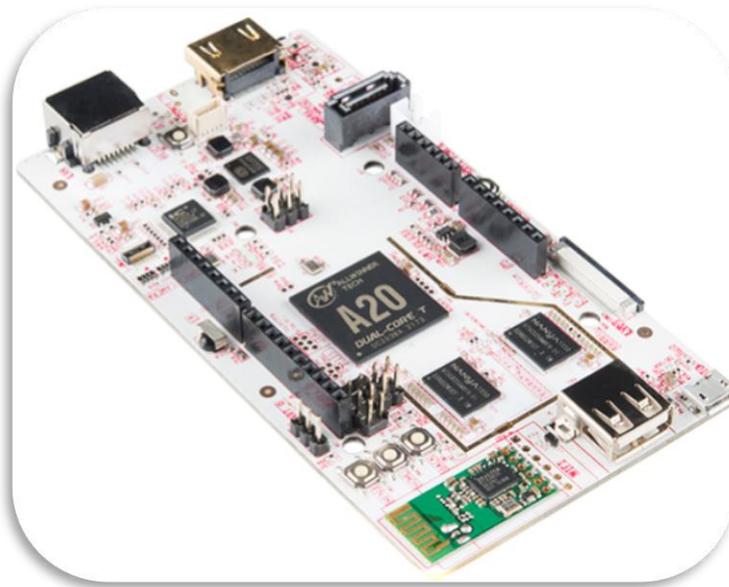
Como precedente a la generación de la conceptualización de plataformas de hardware libre, en las instituciones de educación superior se instruía la programación de micro controladores como requerimiento para fundamentar o interactuar con dispositivos a través de montajes electrónicos sencillos, pero las herramientas disponibles estaban orientadas a profesionales, y muy pocas eran amigables para enseñar la programación de estos dispositivos a estudiantes de niveles intermedios.

### **1.6.1 CONCEPTUALIZACIÓN**

El hardware libre es aquel cuyos diagramas esquemáticos, PCB y diseños lógicos están disponibles de forma gratuita y pública para que cualquier persona pueda reproducir los diseños. No obstante la gran mayoría de proyectos mediante la utilización de hardware libre se empezaron a desarrollar en la última década.

Torres, Lazalde y Vila-Viñas (2015) expresan que los conceptos que inducen el hardware libre tampoco son nuevos ya que la fabricación local de dispositivos autómatas, la legislación de la soberanía e independencia tecnológica, mediante las cuales cualquier persona con los suficientes conocimientos podrá emplearlos para su propio beneficio.

Al relacionarse con el hardware libre se hace mención a aquel hardware que es diseñado e implementado, los mismos que comparten sus diagramas esquemáticos así como también las descripciones del mismo; de esta manera se faculta que el desarrollo sea universalizado para todas las personas que tienen acceso a estas tecnologías. En cuanto al software libre se hace referencia a aquel código fuente de aplicaciones el mismo que es de libre acceso para cualquier persona interesada, el mismo que por sus libertades es modificable, copiable y redistribuible.



*Figura 1 Placa de hardware libre pduino versión 3.*

Fuente: El Autor.

### **1.6.2 CLASES**

Existen diversas clasificaciones de hardware libre, ya que es un término muy amplio al momento de implementar sus aplicaciones.

Torres et al. (2015) concluyen que el hardware libre se debe especificar de qué tipo de hardware se está hablando ya que pueden ser de diferente naturaleza.

### **1.6.2.1 De acuerdo a su naturaleza**

Hardware libre reconfigurable.- Se trata de aquel hardware que puede expresarse con la utilización del lenguaje de descripción del mismo. Cabe mencionar que los diseños se tratan de ficheros de texto, en los cuales se encuentra el código fuente. Se puede utilizar concisamente una licencia libre, similar a GPL. Las dificultades no emergen por el concepto de qué es libre o que es necesario para serlo, sino que surgen con las instrumentales de desarrollo que resultan ineludibles. Para lograr que el hardware reconfigurable coexista como libre, es necesario la aplicación de la licencia GPL a su código. Se desenvuelve de una forma muy análoga a como se realiza con el software libre.

Hardware libre estático.- Es una agrupación de materiales que pertenecen a los sistemas electrónicos.

### **1.6.2.2 De acuerdo a su filosofía.**

La argumentación en lo referente al hardware libre se establece de las personas que platican en las comunidades de software y hardware. Las palabras "hardware" y "diseño de hardware" nombran a dos cosas distintas.

Hardware libre "design".- Representa a un diseño de hardware que consigue ser copiado, distribuido, reformado y elaborado libremente. Se puede pensar que el diseño también no tiene costo, o que cualquier puesta en práctica de hardware libre design estará libre de costo alguno.

Hardware "Open source".- La información del diseño se pone en consideración y uso del público en general.

Open Hardware.- Es una marca patentada del Open Hardware Specification Program. Es una representación específica de open source hardware, para la cual el requerimiento es que: Una documentación suficiente del elemento debe estar disponible para que un programador pueda escribir un controlador del dispositivo. La documentación debe contener todas las

particularidades de la interfaz del dispositivo - controlador que se dispone a que cualquier usuario lo utilice. (Medrano & Chourio, 2015).

### **1.6.3 LIBERTADES**

El hardware libre plantea las cuatro libertades que se establecen en el software libre, pero referentes a los planos del hardware, mediante los cuales conseguimos ensamblar el hardware. Los planos son utilizables por cualquiera para que los pueda utilizar, modificar y distribuir. (González, González y Gómez-Arribas, 2003).

#### **1.6.3.1 Plano Esquemático**

Suministra una lista de los elementos lógicos y las señales que se enlazan entre ellos, pero no proporciona información del aspecto físico de la placa.

#### **1.6.3.2 Circuito Impreso (PCB)**

Facilita la ubicación en donde situar los componentes, las dimensiones, encapsulados y que vías siguen las pistas para unir los pines. Adicionalmente describe con superfluidad el aspecto físico de la placa y sus dimensiones.

#### **1.6.3.3 Fichero de fabricación**

Establece toda la información requerida para que se puedan fabricar los PCBs. Para elaborar la placa los diagramas esquemáticos y el PCB. El fichero GERBER es el que conseguimos como resultado y nos permite efectuar una fabricación industrial de nuestro PCB. Este fichero puede pero no siempre es

generado por el diseñador. A veces es el propio fabricante lo genera, utilizando la información que hay en el PCB. (González et al. 2003)

#### **1.6.3.4 Restricciones y Lineamientos**

Se establecen los siguientes lineamientos y restricciones de acuerdo a la documentación existente. (Medrano & Chourio, 2015).

Documentación.- La información completa del hardware debe estar disponible de manera libre y deberá estar sujeta a cualquier modificación.

Software Necesario.- En el caso que el diseño demande de licencia de software, este deberá establecer parámetros de documentación suficiente y se publicará bajo una licencia admitida por OSI.

Obras Derivadas.- Las modificaciones de licencia deben ser permitidas y también la realización de trabajos derivados y la fabricación, comercialización y utilización de productos creados a partir de los archivos de diseño.

Redistribución libre.- La licencia no establece restricciones a un tercero el comercializar o conceder la documentación del proyecto. No es posible ejecutar ningún derecho sobre obras derivadas.

Atribución de licencia.- Puede solicitar documentos derivados y advertencias de copyright asociados a los elementos. De igual forma se debe hacer referencia al diseñador.

Distribución de la licencia.- Se entiende distribuida sin solicitar permisos añadidos. La licencia no debe establecer restricciones a un diferente hardware o software.

La neutralidad tecnológica de la licencia es indispensable.- Ninguna disposición debe fundamentarse en una tecnología específica, componente, material o interfaz para su utilización. (González et al. 2003)

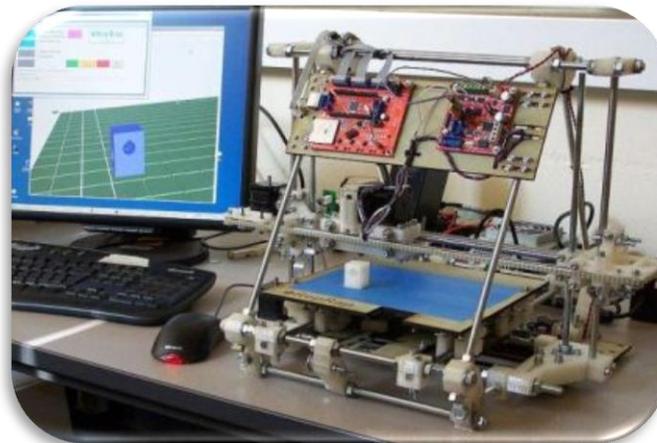
## 1.6.4 DISPOSITIVOS BASADOS EN HARDWARE LIBRE

Se describen los proyectos más interesantes en lo referente a este concepto y que ocasionan desde la elaboración de ordenadores portátiles y teléfonos modulares hasta la producción de automóviles, cruzando por el lanzamiento de la robótica en el aula. (Medrano & Chourio, 2015).

### 1.6.4.1 Proyecto RepRap

En 2004 Adrian Bowyer con la ayuda de sus compañeros de la Universidad de Bath inician este proyecto en Reino Unido, y como aplicación implementan una impresora 3D la cual es preparada para replicarse a sí misma con la impresión de gran parte de sus elementos.

El dispositivo se encuentra bajo licencia GNU GPL, que admite la reproducción por parte de cualquier persona, la misma que puede estudiar, distribuir y optimizar sus diseños y código fuente.



*Figura 2 Módulo ensamblado en el proyecto RepRap.*

Fuente: RepRap.org.

#### 1.6.4.2 Raspberry Pi

Se trata de un ordenador de dimensiones reducidas que consta de una película base sobre la cual se acopla un procesador, un chip gráfico y un bloque memoria RAM.

El principal objetivo para su lanzamiento fue la estimulación en la enseñanza de informática en establecimientos educativos de todo el mundo. Este proyecto actualmente reúne a un notable grupo de desarrolladores preparados para compartir detalladamente las características que van sumándose a la placa. (Torres et al. 2015).



*Figura 3 Microcomputador distribuido por la Fundación Raspberry Pi.*

Fuente: <http://opensource.com/life/14/3/favorite-raspberry-pi-projects>.

#### 1.6.4.3 Hardware libre en base a Arduino

Se refiere a un dispositivo que adiciona un sencillo microcontrolador y un entorno de desarrollo para diseñar aplicaciones muy variadas, desde robots hasta procedimientos de riego automático.

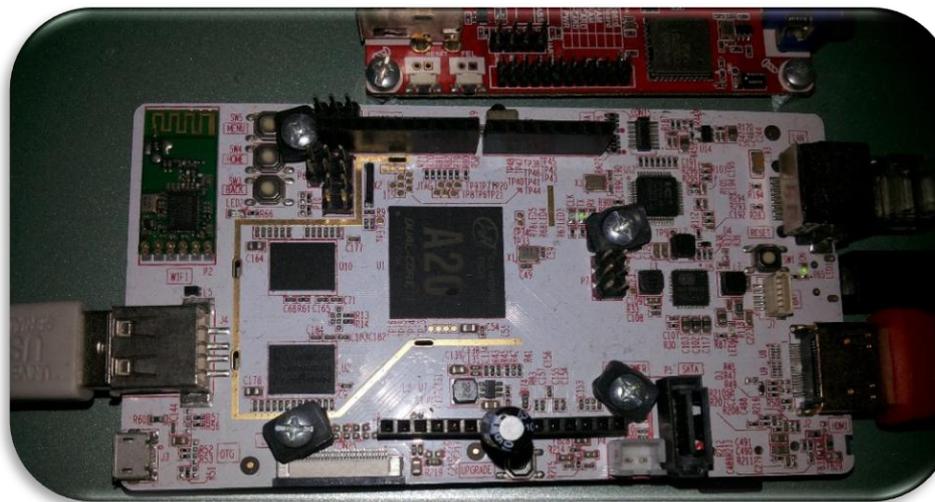
Arduino se considera como una plataforma de hardware libre, básicamente es un módulo de computación física de código abierto, fácil de usar y constituido en una placa electrónica simple. Está encaminado a diseñadores,

estudiantes y cualquier persona que guste del desarrollo de herramientas o aplicaciones interactivas. (Torres et al. 2015)

Con la utilización adecuada se puede medir las variables físicas del medio ambiente, mediante la captación de señales de una variedad de sensores o interruptores y a su vez efectuar control a varios dispositivos tales como luces, motores, y otros actuadores. El microcontrolador en el módulo se programa utilizando lenguaje Arduino y el entorno de desarrollo Arduino. Los proyectos realizados pueden ser autónomos o se pueden conectar con algún software ejecutado en un ordenador indistinto.

Las placas se las elabora a mano o se puede conseguirlas pre ensambladas; el software se descarga de forma gratuita. Los diseños relacionados al hardware como los archivos CAD, están utilizables bajo una licencia de código abierto, los mismos que se los puede adaptar de acuerdo con la necesidad. (Medrano & Chourio, 2015).

El software está disponible para Windows, Mac OS y Linux. El lenguaje de programación está basado en un entorno de programación multimedia de procesamiento.



*Figura 4 Placa Pcduino (Mini Pc y Arduino).*

Fuente: El Autor.

#### 1.6.4.4 CubieBoard

El módulo de desarrollo Cubieboard pertenece al conjunto de dispositivos que aparecieron con el lanzamiento de Raspberry Pi. El diseño, fabricación y mercantilización de esta placa es realizada por la organización Cubietech Limited. Entre los dispositivos destacados se establecen Cubieboard, Cubieboard2 y Cubietruck, siendo este el minicomputador más actual.

Varios desarrolladores adheridos a una comunidad que se especializa en esta placa han realizado algunas adaptaciones de los sistemas operativos basados en Linux más populares, los cuales son:

- ✓ Arch Linux, Berryboot, Debian, usando el entorno de escritorio LXDE.
- ✓ Fedora, Kali, Mer, Tiny Core, Lubuntu
- ✓ Android. La versión varía de acuerdo a la placa elegida.

Éstos se proporcionan para libre descarga desde la web oficial de Cubietech Limited.



*Figura 5 Mini Ordenador distribuido por Cubietech Limited.*

Fuente: El Autor.

## **1.6.5 APLICACIONES CON HARDWARE LIBRE**

Actualmente debido a la versatilidad que poseen los mini computadores cuya arquitectura corresponde a hardware libre se establecen muchas aplicaciones en las cuales se facilitan los procedimientos necesarios. (Medrano & Chourio, 2015).

### **1.6.5.1 Monitoreo de sensores y control de actuadores**

Una de las aplicaciones más comunes que se suscitan a menudo son las que se realizan mediante la utilización de minicomputadores para el monitoreo de varios tipos de sensores, ya sean estos de magnitudes físicas tales como temperatura, distancia, humedad, luminosidad, entre otros, los cuales captan los valores de cada magnitud, estos son recolectados por el dispositivo y controlados mediante una interfaz gráfica de usuario, a su vez son enviados a guardarse en su propia memoria o de forma externa, donde una aplicación está diseñada para generar estadísticas y controlar en tiempo real el estado de actuadores tales como ventiladores, servomotores etc.

### **1.6.5.2 Cámaras reconfigurables de red**

Actualmente se desarrolla procedimientos de imagen con software y hardware libre, así como también se fabrican cámaras reconfigurables de red que utilizan GNU/Linux y FPGA reconfigurable. El código es libre y todas las disposiciones PCB, están utilizables bajo la licencia de GNU/GPL. (González et al. 2003)

## 1.7 CONVERSIÓN ANÁLOGA DIGITAL

La conversión análogo-digital demanda la transformación de señales analógicas en digitales pasando por una serie de procedimientos.

Un convertidor de señales analógicas en digitales se refiere a un dispositivo capaz de exhibir en su salida una señal digital desde una señal analógica de entrada. El proceso de conversión necesariamente debe abordar tres partes importantes: el muestreo de la señal analógica, cuantificación de dichas muestras y la codificación de las mismas. (Maloney, 2006)

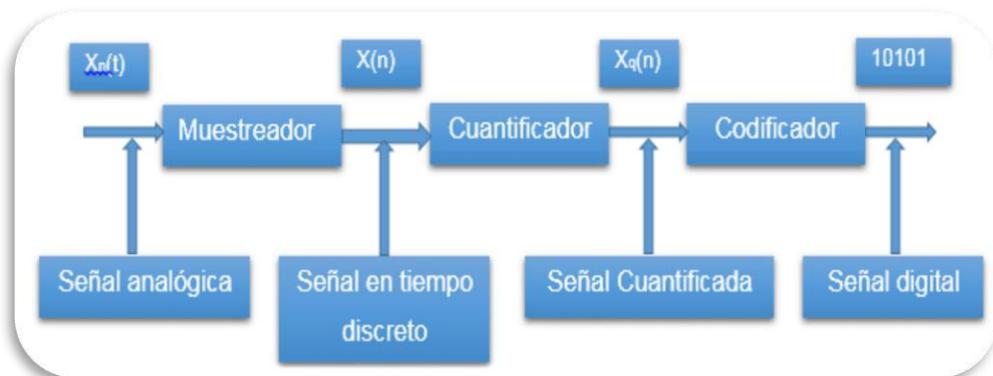


Figura 6 Proceso de conversión de señales analógicas en digitales.

Fuente: El Autor.

### 1.7.1 MUESTREO

Se trata del procedimiento de medir el valor instantáneo de la señal analógica a ser digitalizada, estableciendo la frecuencia de medición, la cual se instaura conforme al teorema de Nyquist. Este establece que la frecuencia mínima para muestrear una señal ( $f_s$ ) debe ser al menos dos veces el valor de la frecuencia máxima de la señal analógica de entrada ( $f_a$ ). Si  $f_a$  es mayor que  $2f_s$  existirá distorsión en la señal regenerada. (Balcells, 2006). Entonces tenemos que:

$$f_s \geq 2f_a$$

Donde:

$f_s$  = frecuencia mínima de muestreo de Nyquist (Hertz).

$f_a$  = máxima frecuencia que se debe muestrear (Hertz).

El periodo de muestreo se determina por el tiempo transcurrido entre cada proceso de medición de la señal y está dado por:

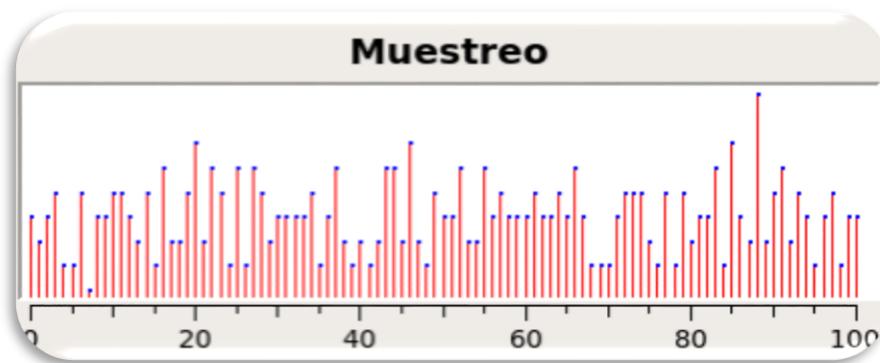
$$T_s = 1/f_s$$

Donde:

$T_s$  = periodo de muestreo

$f_s$  = frecuencia mínima de muestreo de Nyquist (Hertz)

Generalmente como parte del proceso de muestreo se encuentra la etapa de retención. Esta etapa es usada para conservar el valor de la muestra el tiempo requerido para que sea cuantizada. En la figura (1-7) se identifica una señal sinusoidal a la cual se le designa un procedimiento de muestreo y retención. (Balcells, 2006).



*Figura 7 Muestras obtenidas a partir de una señal original.*

Fuente: El Autor.

## 1.7.2 CUANTIFICACIÓN

Este proceso se determina por la limitación o conversión del rango continuo de valores que las muestras logran tener a un número establecido y limitado de valores discretos. El número de valores tolerados depende

concisamente del número de salidas o de bits con los que se van a simbolizar las muestras. (Balcells, 2006).

Entonces se establece que el número de valores tolerados está dado por:

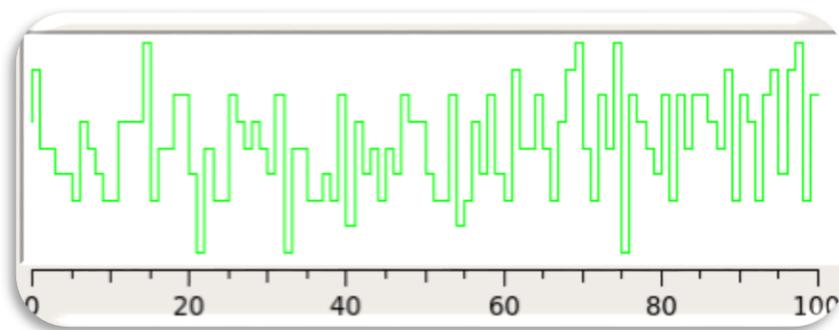
$$N = 2^n$$

Donde:

N= número de valores discretos permitidos

n = número de salidas o de bits para representar las muestras

La señal cuantizada solamente obtiene valores en los niveles permitidos, esto ocasiona un error conocido como ruido de cuantización y es posible reducirlo incrementando el número de niveles admitidos para la señal digital.

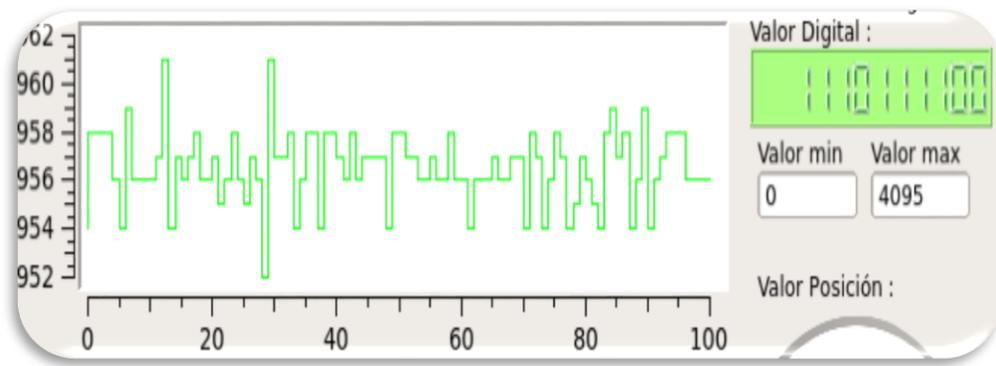


*Figura 8 Cuantificación de una señal muestreada.*

Fuente: El Autor.

### 1.7.3 CODIFICACIÓN

Representa la asignación de un código o número binario a todas y cada una de las muestras cuantificadas. En la figura (9) se muestra la asignación del código a cada uno de los niveles de cuantificación utilizando 12 bits.



*Figura 9 Codificación de una señal cuantificada.*

Fuente: El Autor.

## **1.8 SENSORES.**

Un sensor se trata de un dispositivo diseñado para tomar información de una magnitud del exterior y convertirla en otra magnitud, regularmente eléctrica, que sea capaz de cuantificarse y manipularse.

Habitualmente estos elementos se fabrican mediante el uso de componentes pasivos (resistencias variables, NTC, LDR, etc... todos los dispositivos que modifican su magnitud en función de una determinada variable), y la utilización de componentes activos. (Ruiz, 2010).

### **1.8.1 SENSORES DE POSICIÓN**

Su función primordial es medir o captar la posición de un objeto específico en el espacio y se puede encontrar de varios tipos.

#### **1.8.1.1 Los captadores fotoeléctricos**

La fabricación de esta clase de sensores, se basa en el empleo de una fuente de señal luminosa y una célula captadora de dicha señal. El principio de

funcionamiento se basa en la emisión de luz, y en la detección realizada por los fotodetectores. De acuerdo con la forma en que se realice esta emisión y detección de luz, este tipo de captadores se dividen en: captadores por barrera y captadores por reflexión. (Balcells, 2006).

- ✓ Captadores por barrera. Estos dispositivos detectan la presencia de un objeto, debido a la interferencia en la recepción de la señal luminosa.
- ✓ Captadores por reflexión; La señal es reflejada por el objeto de características especiales de reflexión, y esta luz reflejada es absorbida por el captador fotoeléctrico, lo que indica la presencia de un objeto. (Ruiz, 2010).

### **1.8.2 SENSORES DE CONTACTO.**

Estos elementos, son los más sencillos, ya que solamente se trata de interruptores que se activan o desactivan al contacto con un objeto, de esta forma se inspecciona la presencia de un objeto en un lugar específico. Su facilidad de construcción y su robustez, los hacen muy importantes en el desarrollo de robots. (Pérez, Acevedo, & Silva, 2009).

### **1.8.3 CAPTADORES DE CIRCUITOS OSCILANTES.**

Se basan en la presencia de un circuito en el mismo que crea una oscilación a una frecuencia determinada, cuando no existe presencia de un objeto en el campo de detección del sensor, el circuito conserva fijamente su oscilación, pero cuando un objeto está en la zona de detección, la oscilación deja de generarse, consecuentemente el objeto es detecta. Este tipo de sensores en la mayoría de casos se usan como detectores de presencia, ya que al no poseer elementos mecánicos, su robustez y su vida útil son elevadas.

#### **1.8.4 SENSORES DE MOVIMIENTO**

Es uno de los más trascendentales en robótica, ya que proporciona información sobre las maniobras de las partes que constituyen el robot, y de esta forma se controla con un alto grado de precisión los movimientos del robot en su ambiente de trabajo. (Pérez, Acevedo, & Silva, 2009).

##### **1.8.4.1 Sensores de deslizamiento.**

Se utilizan singularmente en robótica para determinar con que fuerza ha de sujetar un objeto para que no se rompa por concepto de una fuerza excesiva, o a su vez que no se caiga de las pinzas del robot por una sujeción inadecuada.

Su funcionamiento es simplificado, ya que se encuentran instalados en el elemento final (pinzas), cuando el robot ejecuta la acción de apresar el objeto, las pinzas lo sujetan con una fuerza terminante y lo intentan levantar, si se genera un ligero desplazamiento del objeto, inminentemente es incrementada la presión generada en las pinzas sobre el objeto, y esta acción se repite hasta que el movimiento del objeto se ha eliminado.

##### **1.8.4.2 Sensores de velocidad**

Se trata de sensores que pueden detectar la velocidad de un objeto ya sea linealmente o como movimiento angular, la aplicación más distinguida es la medición de velocidad angular de motores que generan movimiento a los diferentes elementos del robot. La forma más conocida de medir la velocidad del giro de un motor, es ocupar un dinamo taco métrico articulado al eje del motor que deseamos conocer su velocidad, este mecanismo genera un nivel fijo de tensión continua de acuerdo con la velocidad de giro del eje, y al conocer a qué valor de tensión pertenece una velocidad específica, se puede establecer a qué velocidad gira un motor. (Ramos, 2002).

## 1.9 SENSORES Y HARDWARE LIBRE.

Para conseguir las características de un sistema integrado evidenciando la versatilidad de las plataformas de hardware de libre es necesario tener a disposición una cantidad considerable de sensores capaz de medir varias magnitudes físicas, las cuales al ser captadas y procesadas, posteriormente sean visualizadas mediante la utilización de una interfaz gráfica, donde se pueda observar de forma detallada los procesos por los cuales las señales de estos sensores deben atravesar hasta ser convertidas en señales de tipo digital.



Figura 10 Sensor con interfaz gráfica en PC-duino.

Fuente: El Autor.

### 1.9.1 SENSORES ANALÓGICOS

Las señales resultantes de la medición de las magnitudes físicas a través de la utilización de sensores son de tipo analógico, las cuales deben ser procesadas con el uso de conversores de analógico al digital, para su posterior

establecimiento de la equivalencia entre la señal medida y la magnitud pasando por todas las etapas que este proceso demanda.

Una vez que las señales son tratadas y convertidas a valores digitales se puede proceder a interpretarlas de acuerdo a la cantidad de la dimensión que estas representan. (Ramos, 2002).

#### **1.9.1.1 Sensor de temperatura con elemento semiconductor LM35**

Se trata de un sensor de precisión en circuito integrado, el cual entrega a su salida una señal de tensión lineal y proporcional a la temperatura en grados Celsius. No es necesaria una calibración externa ya que proporciona una precisión de  $\pm 1/4^{\circ}\text{C}$  en un entorno cerrado y  $\pm 3/4^{\circ}\text{C}$  en el rango establecido de  $-55$  a  $+150^{\circ}\text{C}$ .

Este tipo de sensor se halla calibrado específicamente en la escala Celsius, su señal de salida es lineal y contiene un factor de escala de  $10.0\text{ mV}/^{\circ}\text{C}$  y una precisión asegurada de  $0.5^{\circ}\text{C}$ . Su rango de funcionamiento resulta muy útil para proyectos que demanden aplicaciones remotas. Funciona con voltajes de alimentación de 4 a 30 voltios y a una corriente nominal de  $60\mu\text{A}$ . Es capaz de mantener una baja impedancia en su salida y contiene tres terminales, dos de alimentación y uno desde donde se toma la señal de salida.

#### **1.9.1.2 Sensor de distancia Sharp GP2Y02YK0F**

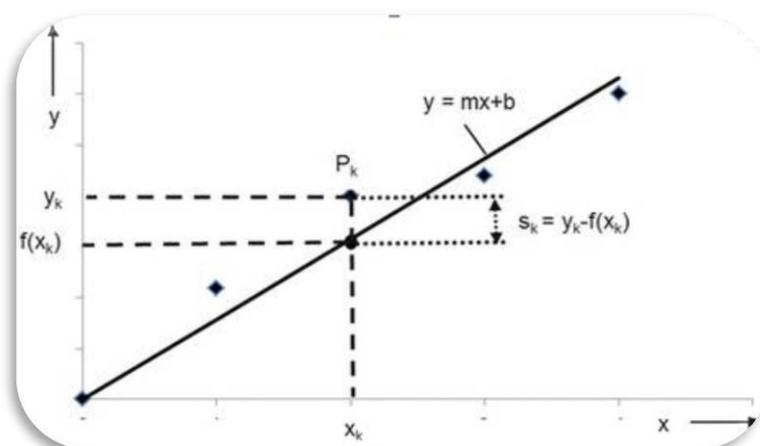
GP2Y0A02YK0F es un elemento de medición para censar distancia, el cual está compuesto por un detector sensitivo de posición, un emisor de infrarrojos y un circuito de proceso de señales. Este dispositivo suministra en su salida una tensión proporcional a la distancia de detección. El rango de medición es de alrededor de 20 cm a 150 cm. La variación de la reflectividad, la temperatura y el periodo de funcionamiento no representan interferencia significativa en la detección de distancia debido a la aplicación del procedimiento de triangulación.



*Figura 11 Sensor Sharp de la familia GP2Y.*  
Fuente: <https://www.sparkfun.com/products/12728>.

#### 1.9.1.2.1 Linealización de la señal de salida.

La relación entre la tensión de salida del sensor y la inversa de la distancia medida es aproximadamente lineal sobre el rango utilizable del sensor. La hoja de datos GP2Y0A02YK contiene un gráfico de tensión de salida analógica como una función inversa de la distancia a un objeto reflectante. Se puede utilizar este argumento para convertir el voltaje de salida del sensor a una distancia aproximada mediante la construcción de una línea de mejor ajuste que relaciona la inversa de la tensión de salida (V) a la distancia (cm).



*Figura 12 Linealización de una señal.*

Fuente: <http://www.hbm.com/es/menu>.

### 1.9.1.3 Sensor de posición potenciómetro

Es un elemento electromecánico que se constituye por una resistencia de valor fijo sobre la cual se traslada un contacto corredizo denominado cursor y que la divide eléctricamente.

La aplicación más frecuente de estos dispositivos en instrumentación es como sensor de desplazamiento del tipo resistivo. El movimiento ya sea lineal o de giro del cursor produce una variación en la resistencia, el cual puede ocuparse para determinar desplazamientos lineales o angulares de elementos acoplados al cursor. (Balcells, 2006).



*Figura 13 Potenciómetro de precisión.*

Fuente: El Autor.

Los potenciómetros pueden utilizarse para medir numerosas magnitudes físicas siempre y cuando se puedan transformar en desplazamiento.

Los componentes resistivos utilizados son de diferentes tipos. Se utilizan primordialmente el bobinado que resulta muy lineal y además el plástico conductor que aparte de garantizar la linealidad ofrece una vida útil muy prolongada. El elemento más común es el carbón, su principal característica es el precio pero como desventajas se presentan los cambios de temperatura y su ciclo útil. El cermet es un material que se compone de una combinación de material cerámico y metal que perfecciona las características del carbón. Luego se encuentra el bobinado, que sus primordiales ventajas son el reducido

coeficiente de temperatura, su vida mecánica, menor ruido, alta disipación, y permanencia con el tiempo. (Franco, 2005).

#### **1.9.1.4 Sensor de velocidad.**

Existen diversos sensores dependiendo de la magnitud física a evaluar. En este apartado el trabajo a efectuar, es medir la velocidad, el dispositivo utilizado para tal fin es conocido como tacómetro.

Un tacómetro es un dispositivo que mide la velocidad angular de un eje giratorio. Las unidades más comunes para expresar la velocidad angular son revoluciones por minuto (rpm) y radianes por segundo. (Maloney, 2006).

La característica principal para establecer una clasificación de estos dispositivos es la manera en que censan la velocidad, en donde destacan cuatros tipos de sensores para tacómetros:

- Tacogenerador: Básicamente consiste de un generador de c-d, en el cual el campo se implanta por medio de un imán permanente acoplado en el estator o por un electroimán encendido separadamente en el estator. El voltaje de salida es creado en un devanado de armadura de c-d, el cual es proporcional linealmente a la velocidad del inducido.
- Sensor pickup magnético: Distinguido además como tacómetro de rotor dentado, y está compuesto de varios dientes ferromagnéticos en su rotor, los cuales generan una señal de salida que puede ser utilizada para especificar la velocidad.
- Sensor de efecto Hall: Este fenómeno ocurre cuando por una placa metálica circula una corriente eléctrica y a su vez se encuentra ubicada en un campo magnético perpendicular a la dirección de la corriente, se genera en dicha placa un campo eléctrico transversal. Este campo, llamado campo de Hall, es la consecuencia de fuerzas ejercidas por el campo magnético sobre las partículas de la corriente eléctrica, sean estas positivas o

negativas. Este efecto es muy utilizado para la fabricación de varios sensores, particularmente en aplicaciones de posición y movimiento, lineal o rotatorio.

- Encoders: Estos dispositivos consisten de un elemento electro-opto-mecánico que se encuentra acoplado a un eje, y proporciona una señal digital de la posición angular. Está constituido por una fuente de luz, un diodo fotoreceptor y un disco óptico con un conjunto de ranuras o sellos de tal forma que, cuando se produce la rotación, éstas generan una sucesión de pulsos de luz que son detectados por el fotorreceptor, y estableciendo la cantidad de pulsos generados es posible determinar el número de revoluciones por unidad de tiempo.

Entre las clases de sensores citadas, para el proceso de medición de velocidad, se manejará el sensor de efecto Hall que viene montado en un ventilador de 4 pines comúnmente utilizados para control de temperatura en procesadores de computaras de escritorio; ya que disponen de un circuito que compone el sensor, el acondicionamiento de la señal y el control de velocidad por medio de PWM. Las características de estos elementos se describen en apartados posteriores. (Maloney, 2006).

Una vez obtenida la señal de pulsos desde el sensor de efecto Hall del ventilador, esta señal es usada en la entrada del convertidor de frecuencia a voltaje LM2917 con el objetivo de conseguir un voltaje de corriente directa a partir de la frecuencia captada por los pulsos.

## **2 DIMENSIONAMIENTO ELECTRÓNICO**

Este capítulo presenta todo lo concerniente a los procedimientos de dimensionamiento de todos los elementos electrónicos necesarios para la implementación del módulo, los mismos que cumplen los requerimientos para realizar el sistema de hardware libre de análisis y verificación de sistemas análogo-digitales.

### **2.1 DIAGRAMA DE BLOQUES DEL MÓDULO**

El proyecto en su totalidad se lo divide en tres subsistemas esenciales, los cuales demandan la utilización de los siguientes elementos:

- Sistema de Conversión de Análogo a digital
  - Módulo de hardware libre
  - Sensor de temperatura
  - Sensor de distancia
  - Sensor de posición
  - Sensor de velocidad angular
- Sistema de Conversión de digital a análogo
  - Módulo de hardware libre
  - Módulo R2R
  - Ventilador
  - Matriz de leds
- Interfaz gráfica de usuario
  - Lenguaje de Programación
  - Interfaz Gráfica

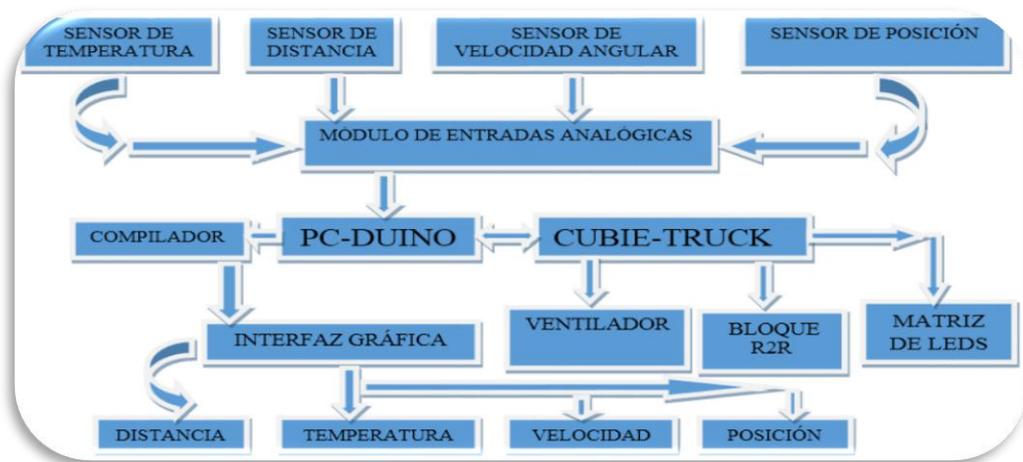


Figura 14 Diagrama de bloques del módulo didáctico de hardware libre.

Fuente: El Autor.

## 2.2 SELECCIÓN DE LOS MÓDULOS DE HARDWARE LIBRE

De acuerdo con los objetivos planteados en el presente proyecto es necesario dimensionar y seleccionar los módulos de hardware libre, los cuales son los encargados de gestionar, modificar y administrar toda la información requerida para el correcto funcionamiento de las etapas necesarias que constituyen este trabajo investigativo.

Uno de los aspectos más significativos que se toma como base para la selección es la versatilidad que estos pueden brindar al momento de abordar cada una de las etapas requeridas de acuerdo con la jerarquía que estas demanden en lo que se refiere a conversión de señales analógicas en digitales y viceversa.

### 2.2.1 SELECCIÓN DEL MÓDULO DE CONVERSIÓN DE LAS SEÑALES ANALÓGICAS EN DIGITALES.

Debido a que se requiere convertir señales analógicas en digitales abordando cada una de las fases que demanda la conversión análoga-digital,

es de vital importancia considerar el número de dispositivos que serán conectados tanto a la entrada como a salida de los módulos principales.

De acuerdo a la planificación del trabajo de grado que se detalla, el número de sensores necesarios se especifica en cuatro de tipo analógico, los cuales entregan una señal correspondiente y proporcional a la magnitud física que estos estén midiendo, que concierne concretamente a temperatura, posición angular, distancia y velocidad angular. Entonces será necesario que los módulos que se proceda a seleccionar tengan la capacidad de leer, almacenar y procesar señales de tipo analógico mediante el uso de la respectiva entrada analógica con una adecuada resolución traducida en forma de bits.

También es necesario establecer lineamientos de selección a través de una adecuada comparación entre dispositivos existentes en el mercado como por ejemplo la capacidad de procesamiento de los mismos, memoria interna, memoria expandible y programas existentes que faciliten la representación de la interfaz gráfica de las etapas que debe atravesar una señal hasta convertirse en digital.

Las especificaciones de los equipos más destacados en cuanto a hardware libre se refieren y de los cuales se puede disponer se encuentran detallados en la tabla (1).

*Tabla 1 Comparativo entre Pc-duino 3 y raspberrypi.*

ITEMS	PC-DUINO 3	RASPBERRYPI
<b>SISTEMA</b>		
<b>CPU</b>	ARMv7 Cortex A8 (Allwinner A10) @ 1Ghz	ARMv6@700MHz - Broadcom 2835
<b>RAM</b>	DRAM 1GB	256Mo (version A) / 512 Mo (version B)
<b>Almacenamiento</b>	microSD standard (<=32GB)	4GB flash, microSD standard (<=32GB)
<b>GPU</b>	OpenGL ES 2.0,	OpenGL ES2.0, OpenVG 1.1 Mali 400 Dual core
<b>ESPECIFICACIONES DE SOFTWARE</b>		
<b>OS</b>	Lubuntu 12.04; Android 4.2	Raspbian (ARM de Debian Whee)
<b>Lenguaje de programación</b>	C, C++, Java, Android SDK, Python	C, C++, Java, Python
<b>MULTIMEDIA</b>		
<b>Audio</b>	HDMI, out : jack 3,5mm, in: --	HDMI, out : jack 3,5mm, in : --
<b>Video</b>	HDMI, RCA	HDMI
<b>INTERFAZ</b>		

<b>Red</b>	Ethernet 10/100 Mbps; Wifi Built-in	Ethernet 10/100/1000 Mbps
<b>USB</b>	1 puertos USB + 1 puerto micro-USB OTG	2 puertos
<b>ALIMENTACIÓN ELÉCTRICA</b>		
<b>Power</b>	microUSB de 5V@700mA (5V@2A)	microUSB de 5V@400mA (5V@1A)
<b>BLOQUES DE TERMINALES DE E/S</b>		
<b>GPIO</b>	18 E/S en 3.3V compatible con Arduino-like	8 E/S en 3.3V compatible con Arduino
<b>Entradas Analógicas</b>	2 x 6 bits en 0-2V y 4 x 12 bits en 0-3.3V	No
<b>Salidas Analógicas</b>	6 x PWM hardware (2-pines "fast" en 520Hz - 8 bits y 4-pin "slow" en 5 Hz - 20 niveles)	1 x PWM hardware
<b>Comunicaciones Seriales</b>	I2C, SPI, UART	I2C, SPI, UART
<b>PRUEBA DE USO</b>		
<b>Captura de video</b>	Fps 320x240 en VNC : 29 fps	Fps 320x240 : 10 fps
<b>Freq. max. ON/OFF pines E/S</b>	5555 Hz	100 Hz

Fuente: El Autor.

De acuerdo con los requerimientos del sistema general en cuanto a disponibilidad de entradas analógicas de alta y baja resolución para la lectura de los sensores, capacidad de procesamiento versatilidad de memoria, adaptabilidad de señales y existencia de programas que permitan la representación gráfica de las mencionas señales, se establece que la adquisición de una placa de hardware libre denominada PC-Duino y representada en la figura (15) es la mejor opción para continuar con la elaboración de los procesos de conversión análogo a digital.



*Figura 15 Pc-duino versión 3 distribuido por SparkFun.*

Fuente: El Autor.

## 2.2.2 SELECCIÓN DEL MÓDULO PARA CONVERSIÓN DE SEÑALES DIGITALES EN ANALÓGICAS.

El aspecto más representativo para proceder a seleccionar este equipo se encuentra en la peculiaridad que este debe cumplir en cuanto al número de salidas digitales que son necesarias para representar una señal o un conjunto de señales capaz de demostrar las fases que demanda un proceso de conversión de digital a analógico.

Cabe destacar que es necesario también la utilización de puertos de comunicación que faciliten el traspaso de información entre los módulos, así es indispensable que este módulo contenga los mismos puertos de comunicación que la placa utilizada para convertir señales analógicas en digitales.

Con todo lo antes expuesto la mejor opción para cumplir los requerimientos que son necesarios en este proceso se trata de una placa de hardware libre denominada CubieTruck, la cual se encarga de procesar las señales digitales y presentarlas en sus puertos de salida para su traspaso a señales analógicas a través de dispositivos externos.

*Tabla 2 Características de dispositivo CubieTruck.*

<b>PROCESADOR:</b>	AllWinnerTech SOC A20, ARM® Cortex™-A7 Dual-Core, ARM® Mali400 MP2 Complies with OpenGL ES 2.0/1.1
<b>MEMORIA RAM:</b>	2GB DDR3@480MHz
<b>SALIDA DE VIDEO:</b>	HDMI&VGA 1080P display output on-board
<b>TARJETA DE RED:</b>	10M/100M/1G Ethernet
<b>WIRELESS:</b>	Wifi+BT wireless connection with antenna on-board
<b>INTERFACES HDD:</b>	SATA 2.0 soporte interfaz para 2.5' HDD, (para3.5' HDD, necesita otra entrada de 12V)
<b>ALMACENAMIENTO:</b>	8GB NAND flash, combinación posible con uSD (sin tarjeta incluida) y/o HDD SATA
<b>PUERTOS E/S:</b>	2 x USB HOST, 1 x OTG, 1 x SPDIF (fibra óptica), 1 x IR, 4 x LEDs, 1 Auriculares
<b>ALIMENTACIÓN:</b>	DC5V @ 2.5 <sup>a</sup> with HDD, soporte para entrada de batería de litio & RTC

---

**EXTENDED PINS:** 54 extended pins including I2S, I2C, SPI, CVBS, LRADC x2,UART, PS2, PWMx2, TS/CSI, IRDA, LINEIN&FMIN&MICIN, TVINx4 with 2.0 pitch connectors

---

**TAMAÑO:** 11cm x8cmx1.4mm

---

Fuente: <http://cubieboard.org/2013/10/30/cubieboard3-cubietruck-is-all-ready/>.



*Figura 16 Placa de hardware libre Cubie-truck distribuido por Cubie-tech.*

Fuente: El Autor.

## **2.3 DIMENSIONAMIENTO DE LOS SENSORES.**

Para medir las magnitudes físicas de temperatura, posición, distancia y velocidad es necesario la utilización de sensores que entreguen una señal analógica de acuerdo y a la proporcionalidad de cada una de estas, por tal motivo se vuelve primordial la selección de los dispositivos que mejor se adapten a los requerimientos de todo el sistema y a su vez cumplan con satisfacción el acoplamiento de señales hacia los módulos de hardware libre.

### **2.3.1 SENSOR DE DISTANCIA**

Una de las magnitudes físicas que el modulo es capaz de controlar y representar gráficamente es la distancia, la cual es medida a través de un sensor

infrarrojo que entrega una señal analógica en voltaje de acuerdo con la separación que existe entre el sensor y la barrera de medición.

Existen varios modelos y marcas de sensores que cumplen con este propósito, pero debido a dimensiones del proyecto y requerimientos didácticos del sistema se establece que un dispositivo de dimensiones reducidas con carcasa de montaje mediante tornillos de la marca y modelo Sharp GP2Y02YK0F es suficiente para censar la variable distancia y consta de las características y modelo detallados en la tabla (3).

*Tabla 3 Características del sensor infrarrojo de distancia Sharp GP2Y02YK0F.*

<b>Tensión de funcionamiento</b>	4,5 V a 5,5 V
<b>Consumo de corriente medio</b>	33 mA
<b>Rango de medición Distancia</b>	20 cm a 150 cm (8 "a 60")
<b>Tipo de salida</b>	voltaje analógico
<b>Diferencial de salida</b>	Tensión por encima del rango distancia: 2,05 V (típico)
<b>Período de actualización</b>	38 ± 10 ms
<b>Tamaño</b>	44,5 mm x 18,9 mm x 21,6 mm (1,75 "x 0,75" x 0,85 ")
<b>Peso</b>	5 g (0,18 oz)

Fuente: Datasheet Sharp.



*Figura 17 Sensor infrarrojo de distancia Sharp GP2Y02YK0F.*

Fuente: El Autor.

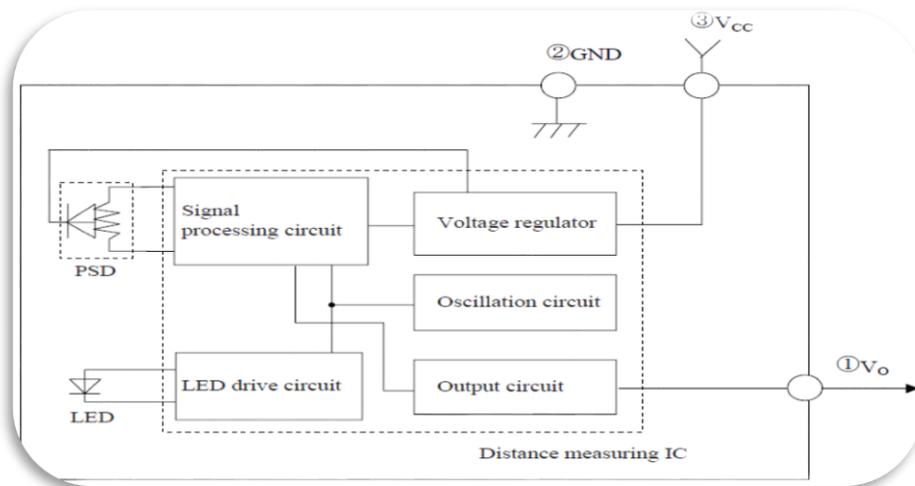


Figura 18 Diagrama de Bloques del sensor Sharp GP2Y02YK0F.

Fuente: Datasheet Sharp.

Uno de los aspectos más puntuales a los que se debe hacer referencia es que el sensor Sharp entrega una señal analógica en forma de voltaje no lineal determinada por la curva representada en la figura (19), por lo que es necesario linealizar esta señal para obtener una proporcionalidad directa entre la distancia medida y su valor en voltaje. Para obtener una adecuada linealización se procede a utilizar un algoritmo sostenido mediante software posterior a la conversión análogo a digital y se detalla más adelante. (Ramos, 2002).

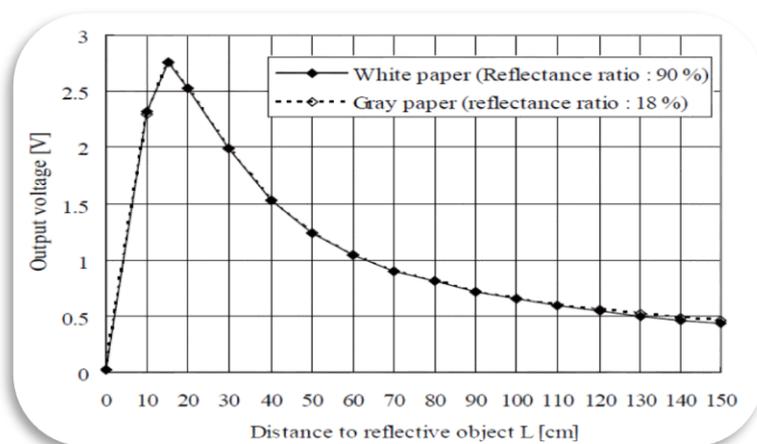


Figura 19 Curva característica del sensor Sharp seleccionado.

Fuente: Datasheet Sharp.

### **2.3.1.1 Recomendaciones de la óptica del sensor**

El objetivo de este dispositivo debe mantenerse limpio. Hay casos en los que el polvo, agua o aceite deterioran las características de este dispositivo. Es necesario tener en cuenta el comportamiento en una aplicación real. Este dispositivo no debe tener contacto con el agua. El agua puede deteriorar las características del sistema óptico.

### **2.3.1.2 Recomendaciones de las características**

Ambas caras del filtro deben ser espejo pulido. También, ya que hay casos que las características no pueden ser satisfechas de acuerdo a la distancia entre la cubierta de protección y este producto o el espesor de la cubierta de protección, por favor utilice este producto después de confirmar la operación con la suficiente aplicación real. En caso de que exista un objeto cerca del emisor a un lado del sensor entre el sensor y el objeto a detectar, por favor utilice este dispositivo después de confirmar que las características de este sensor no cambian por el objeto. Cuando el detector está expuesto a la luz directa del sol, lámpara de tungsteno y así sucesivamente, hay casos en que no se puede medir la distancia exactamente. (Maloney, 2006).

La distancia a un reflector de espejo no puede ser medida a veces exactamente, en este caso se recomienda cambiar el ángulo de montaje de este sensor, ya que se puede mejorar la medición de la distancia.

EL sensor de distancia de este apartado, está diseñado para su uso en equipos electrónicos en general, tales como:

- ✓ Las computadoras personales, equipos de automatización de oficinas, equipamiento de telecomunicaciones prueba, medición de equipos, control Industrial, equipo audio visual, y en general en la electrónica de consumo.

Se debe tomar medidas de seguridad con un diseño redundante, para garantizar fiabilidad y seguridad cuando se utilice dispositivos SHARP en conexiones con equipos que requiere una mayor confiabilidad, tales como:

- ✓ Control de Transporte y el equipo de seguridad es decir: aviones, trenes, automóviles, señales de tráfico, sensores interruptores de fugas de gas, alarmas, varios dispositivos de seguridad, etc.

### 2.3.2 SENSOR DE TEMPERATURA

Para el requerimiento de medición de la magnitud física temperatura es primordial seleccionar un sensor que se comporte de manera que entregue una señal en voltaje, proporcional a la transición por cada grado centígrado que varíe esta magnitud, esto se logra valiéndose de un elemento dotado de semiconductores que al incidir sobre el un cambio de temperatura cambie su voltaje de salida linealmente.

Por todo lo expuesto y debido a exigencias del sistema se establece que el sensor LM-35, dotado de encapsulado de montaje es la mejor alternativa para desempeñar la función de medición de este tipo de magnitud, el cual presenta las características que se expresan en la tabla (4).

*Tabla 4 Características del sensor lm-35.*

Calibrado directamente en la escala Celsius
Lineal con un factor de escala de 10.0 mV/°C
Precisión garantizada de 0.5°C a una temperatura de 25°
Rango de trabajo de -55° a +150°C
Apropiado para aplicaciones remotas
Bajo costo debido al nivel clásico
Opera con voltajes de alimentación en el rango de 4 a 30 voltios
Baja corriente de alimentación (60uA)
Bajo auto calentamiento, 0.08°C en aire estacionario
Baja impedancia de salida, 0.1W por 1mA
Posee tres terminales, dos de alimentación y uno para el voltaje de salida

Fuente: El Autor.

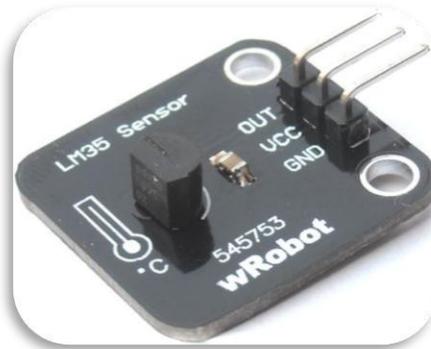


Figura 20 Sensor de temperatura Lm-35

Fuente: <http://www.wRobot.com/sensor/temperatura>

Con respecto a la fiabilidad y repetitividad de este sensor es muy necesario tomar en cuenta las condiciones tanto eléctricas como ambientales en las cuales este dispositivo va a desempeñarse y se detallan en la figura (21) obtenida del datasheet proporcionado por el fabricante del este elemento.

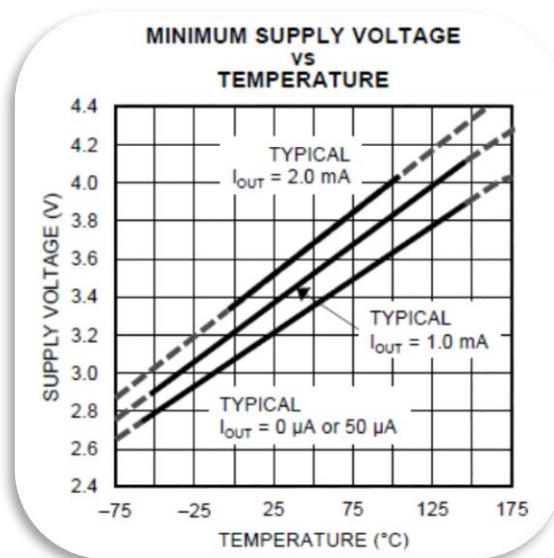


Figura 21 Curva característica de temperatura LM35

Fuente: Datasheet Lm35.

Debido a que este sensor proporciona una señal de salida lineal y proporcionalmente directa de acuerdo con la temperatura, es necesario solamente establecer una igualdad mediante software para utilizarla en procesos posteriores a la lectura de mediciones de la magnitud en cuestión.

### 2.3.3 SENSOR DE POSICIÓN ANGULAR

El proceso de especificación de posición es posible gracias a la utilización de un transductor de posición angular, de tipo absoluto, variación lineal y con salida de tipo analógico, para provocar la demostración de conversión A/D de este tipo de señales que denotaran posiciones angulares por variación de resistencia eléctrica.

La medición de posición angular con el uso de elementos variantes linealmente es una de las situaciones más desarrolladas y utilizadas actualmente para conocer el estado de giro de elementos que requieren realizar este tipo de trabajo. Para este efecto se recurre a la utilización del elemento más común de medición como es el denominado potenciómetro que varía su resistencia de acuerdo con el movimiento que se ejerza en su tornillo cuando se trata de un dispositivo con carcasa y montaje por sujeción a tornillo. (Ramos, 2002).

Una vez que está conectado a una red eléctrica básica, este es capaz de entregar en uno de sus pines un voltaje proporcional a la resistencia que se genera al girar la manija del mismo, dicha señal basta con establecer mediante software una igualdad sencilla para conocer la posición angular de acuerdo con la variación de resistencia.

*Tabla 5 Características del dispositivo para medición de posición angular*

<b>ELEMENTO DE MEDICIÓN</b>	
<b>Resistencia</b>	502 – 5000 ohm
<b>Tamaño de pista</b>	36,5 mm
<b>Eje de salida</b>	6 mm
<b>Linealidad independiente</b>	Mas menos 0,1 %
<b>Señal de salida</b>	Divisor de voltaje
<b>Repetitividad</b>	0,007 %
<b>Coefficiente de temperatura</b>	9ppm/°C
<b>Temperatura de operación</b>	-40 °C + 100 °C
<b>Vida útil (movimientos)</b>	100x10 <sup>6</sup>

Fuente: El Autor



*Figura 22 Potenciómetro de precisión.*  
Fuente: El Autor.

### 2.3.4 SENSOR DE VELOCIDAD ANGULAR.

Los procesos secuenciales de conversión de señales para obtener la velocidad angular de un objeto es posible mediante la utilización del principio de medición analógica de la corriente producida por una bobina y que es proporcional a la velocidad del movimiento. Para cumplir con lo antes descrito se procede a dimensionar y seleccionar el dispositivo que cumple con estos requerimientos y se acopla al sistema que se detalla.

#### 2.3.4.1 Ventilador PWM

Existen ventiladores de conexión a 4 hilos, el objetivo principal de esto es la posibilidad de medición de la velocidad del ventilador y realizar un control de precisión de la revoluciones en todo el rango de velocidad.

*Tabla 6 PinOut Ventilador.*

Conectores pinout del Ventilador		
Pin	Función	Color Cable
1	GND	Black
2	12V	Yellow
3	Sense (tach.)	Green
4	Control(PWM)	Blue

Fuente: El Autor.

La descripción de cada uno de los pines es: Señal GND es tierra y 12V es la alimentación de tensión para el ventilador; Señal Sense o simplemente tacómetro, proporciona dos pulsos por revolución del ventilador. Salida de

colector abierto y la placa principal debe tener una resistencia pull-up a 12V; Señal de control (PWM) es la entrada para pulsos PWM. La frecuencia base es 25kHz y es admisible desde 21kHz a 28kHz. Entrada tiene un nivel TTL e incluye resistencia de pull-up a 5V o 3.3V en nuevas construcciones. La señal no se invierte y el 100% PWM significa revoluciones máximas de ventilador. (Balcells, 2006).

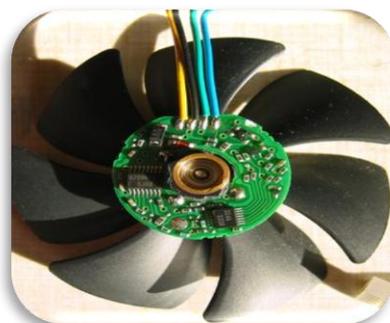
#### 2.3.4.1.1 Ejemplo de un ventilador de 4 hilos

NIDEC F09A-12B3S1, que se suministra con los procesadores Intel. El ventilador es de alta calidad, contiene una hélice equilibrada. Se trató de desmontarle sin destruirle, pero no se ha logrado, porque prácticamente no es posible.



*Figura 23 Ventilador NIDEC F09A-12B3S1.*

Fuente: El Autor.



*Figura 24 Circuitería Ventilador NIDEC F09A-12B3S1.*

Fuente: [http://www.pavouk.org/hw/fan/en\\_fan4wire.html](http://www.pavouk.org/hw/fan/en_fan4wire.html)

*Tabla 7 Características del ventilador.*

<b>Corriente</b>	No será superior a 1,5 A.
<b>Voltaje de entrada del ventilador</b>	Dentro del rango de $12\text{ V} \pm 5\% \text{ V}$ .
<b>Tacómetro: Señal de salida</b>	El ventilador proporcionará una señal de salida del tacómetro con las siguientes características: <ul style="list-style-type: none"><li>• Dos pulsos por revolución.</li><li>• Colector abierto o drenaje abierto, tipo de salida</li></ul>
<b>Control PWM de señal de entrada</b>	Los siguientes requisitos se valoran por el pin PWM (control) del conector del cable del ventilador: <ul style="list-style-type: none"><li>• Frecuencia PWM: Frecuencia central 25 kHz, rango de funcionamiento aceptable 21 kHz a 28 kHz.</li><li>• Tensión máxima para la lógica baja: <math>V_{IL} = 0,8\text{ V}</math>.</li><li>• Máximo absoluto actual proviene: <math>I_{max} = 5\text{ mA}</math> (corriente de cortocircuito).</li><li>• El nivel de tensión máxima absoluta: <math>V_{Max} = 5,25\text{ V}</math> (tensión de circuito abierto).</li></ul>

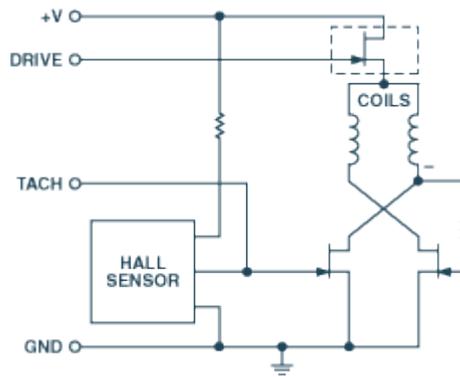
Fuente: El Autor.

#### *2.3.4.1.2 Ventilador de 4 pines PWM*

Este ventilador incluye también el sensor Hall, pero además tiene dos entradas diferentes para la alimentación de los circuitos y el control PWM.

El control de velocidad permite conectar una señal PWM de alta frecuencia para el control del electroimán, además tienen la opción de ser independiente la señal de control (PWM) de la señal de alimentación (+12V).

Al lado opuesto del electroimán se encuentra la circuitería de control. Se puede ver que en ambos hay dos elementos importantes: el electroimán (derecha, "coils") y un sensor Hall (izquierda, "Hall Sensor"). El electroimán es el motor en sí, el sensor Hall es un circuito que permite detectar la velocidad de giro del ventilador. (Balcells, 2006).



*Figura 25 Control PWM ventilador de 4 Hilos.*

Fuente: [http://www.analog.com/library/analogdialogue/archives/38-02/fan\\_speed.html](http://www.analog.com/library/analogdialogue/archives/38-02/fan_speed.html)

En el esquema, +V y Drive se unen mediante un transistor Jfet de canal N, que funciona en condición de interruptor: Cuando drive está a nivel alto (12V) impide el paso de +V, y cuando drive está a nivel bajo (0V) entonces permite el paso de +V. Es decir: es la entrada Drive la que controla exactamente cuándo está conectado y cuando no +V al electroimán. (Ruiz, 2010).

Se consigue regular la velocidad de este ventilador conectando al electroimán un voltaje a pulsos en lugar de un voltaje continuo. Los pulsos de voltaje se convierten en impulsos al electroimán, y al reducir el tiempo que se está aplicando potencia sobre el electroimán, se reduce la velocidad del mismo. Estas señales a pulsos se conocen como señales PWM (“Pulse Width Modulation”). Las mismas que tiene dos características transcendentales:

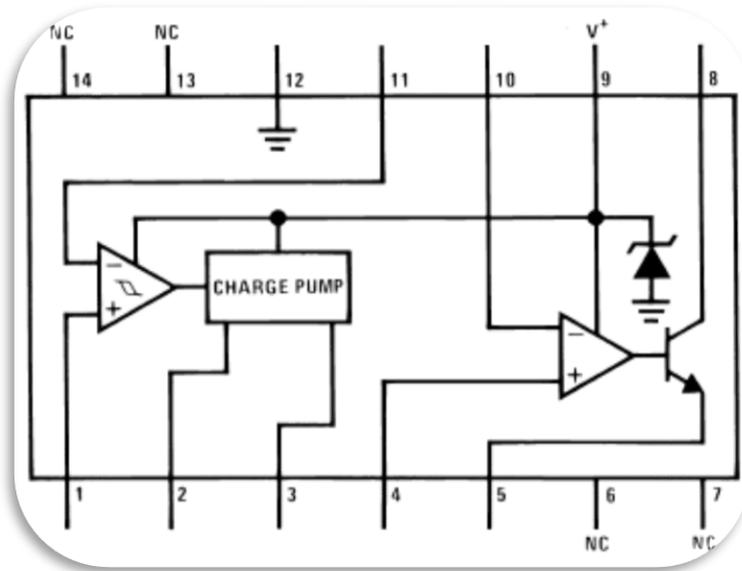
- ✓ Frecuencia: Las señal PWM que se utiliza para regular el ventilador son normalmente ondas cuadradas periódicas. El tiempo de cada repetición (nivel alto + nivel bajo de señal) corresponde al periodo de la señal. El inverso de este tiempo es la frecuencia.
- ✓ Ciclo de trabajo (“duty cycle”): La proporción del tiempo que permanece la señal a nivel alto con respecto al tiempo que está a nivel bajo en cada periodo es lo que se conoce como ciclo de trabajo. Esto es lo que realmente afecta a la velocidad del ventilador. El regulador PWM de velocidad del ventilador lo que hace realmente para variar la velocidad es variar el ciclo de trabajo.

La señal del tacómetro, cuando es impulsado por una tensión de corriente continua, tiene una salida de onda cuadrada muy similar a los tacómetros ideales. Con PWM de baja frecuencia, la señal de velocidad sólo es válida cuando se aplica energía al ventilador, es decir, durante la fase del pulso. Cuando la unidad PWM se conmuta a la fase de apagado, la circuitería de generación de señal del tacómetro interno del ventilador es también apagada. (Horenstein, 1997).

Una vez conseguida la señal de pulsos del sensor de efecto Hall, esta es llevada al convertidor de frecuencia a voltaje (LM2917N) con el fin de lograr un voltaje de corriente directa a partir de la frecuencia captada por los pulsos de la señal de entrada. El LM2917 es un circuito integrado elaborado por la empresa National Semiconductor utilizado como un convertidor de frecuencia a voltaje, la aplicación más relevante es la fabricación de tacómetros, debido a que suministra un voltaje de corriente directa, el mismo que es proporcional a la frecuencia de la señal de entrada. Las características del LM2917 más significativas son:

- ✓ Entrada con amplificador operacional en modo comparador.
- ✓ Técnica de Carga Bombeada "Change Pump".
- ✓ Linealidad del 0,3%.
- ✓ Amplificador Operacional y transistor flotante de salida.
- ✓ Zener de alimentación.

Generalmente los circuitos convertidores de frecuencia a voltaje demandan de un gran número de componentes adicionales para poder ser utilizados en la construcción de circuitos. Al contrario, el circuito integrado LM2917, contiene los elementos necesarios para su funcionamiento, y asimismo, solo requiere un número mínimo de componentes externos. En la figura se muestra el diagrama en bloques de dicho circuito integrado.



*Figura 26 Diagrama en bloques y Distribución de pines del LM2917*

Fuente: Datasheet Lm2917.

Este circuito integrado cuenta con tres elementos básicos: un amplificador operacional de entrada, un convertidor de frecuencia a voltaje denominado “bomba de carga”, y un amplificador operacional comparador con un transistor de salida. Conjuntamente, cuenta con un diodo zener que efectúa la función de regulación del voltaje de salida. El amplificador de entrada tiene una histéresis a  $\pm 15$  mV, la misma que provee un switcheo nítido donde si se demuestra ruido en la señal de entrada, nos permite un rechazo total del ruido debajo de la amplitud donde no hay señal de entrada. La bomba de carga convierte la frecuencia de entrada en voltaje de corriente directa. Para ello se quiere un capacitor, una resistencia de salida y un capacitor integrativo. (Lago, 2012).

Puesto que se resaltó, el voltaje corriente directa de salida del LM2917 es proporcional a la frecuencia de la señal de entrada, de modo que se obtiene una gráfica donde se señala la velocidad del ventilador en r.p.m. frente a el voltaje de referencia adquirido de la salida del tacómetro.

El voltaje de salida del tacómetro es proporcional a las r.p.m. del ventilador, desde 1800 hasta 3000 r.p.m. nominales del ventilador. Donde se señala que cuando la velocidad alcanzada en el ventilador sea de 3000 r.p.m.; el sensor mide las r.p.m. en el inducido y el circuito del LM2917 acondiciona la

señal, suministrando un voltaje de referencia en sus terminales alrededor de 1.82 volts de corriente directa.

El dispositivo presentado en este apartado, aclara que se puede medir la velocidad del ventilador de corriente directa de 4 pines e inspeccionar el proceso por medio de la interfaz gráfica de usuario correspondiente a la magnitud velocidad. Así, la salida de la señal de referencia obtenida del tacómetro es llevada a la entrada analógica del Pc-Duino, para verificarse con la velocidad del ventilador.

La hoja de datos del LM2917 enuncia una serie de ejemplos de utilización del mismo, una parte importante es conocer los dispositivos que harán que funcione de manera adecuada, a continuación se presenta un esquema que representa en diagrama de bloques sus componentes:

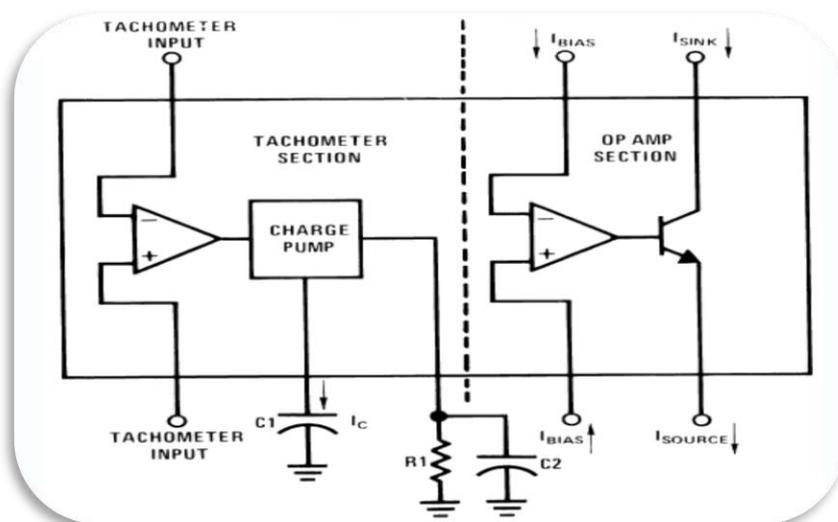


Figura 27 Diagrama en bloques LM2917.

Fuente: Datasheet Lm2917.

Ejemplo de diseño:

$$V_{out} = f_{IN} * V_{cc} * R1 * C1 \dots (1)$$

Ecuación 1 Ecuación de salida de voltaje del LM2917.

Fuente: Datasheet Lm2917.

Para obtener un voltaje que se acople perfectamente a una entrada analógica del Pc\_duino, el valor del mismo debe ser como máximo de 3,3v, el circuito integrado está conectado a una fuente de alimentación de 12v, la

frecuencia máxima proporcionada por el ventilador estará por el orden de los 100 Hz, la frecuencia mínima del ventilador esta por el orden de 50 Hz, por la hoja de datos del circuito integrado (datasheet Lm2917) se tiene que el voltaje de rizado es de 0,05 Volt.

Calculo de R1:

$$R1 > \frac{V_{3max}}{I_{3min}} \dots (2)$$

*Ecuación 2 Ecuación de cálculo de R1 del LM2917.*

Fuente: Datasheet Lm2917.

Donde:

$I_{3min} = 150\mu A$  por datasheet y  $V_{3max} = 3,3$  Volt.

$$R1 > \frac{3,3 V}{0,000150 A}$$

$$R1 > 22000 \Omega$$

Calculo de C1:

$$C1 = \frac{V_{3max}}{(R1 * V_{cc} * f_{max})} \dots (3)$$

*Ecuación 3 Ecuación de cálculo de C1 del LM2917.*

Fuente: Datasheet Lm2917.

$$C1 = \frac{3,3v}{(22k * 12 * 100)}$$

$$C1 = 0,000000125 F$$

$$C1 = 125 nF$$

Calculo de I<sub>2</sub>:

$$I_2 = f_{IN} * C1 * V_{cc} \dots (4)$$

*Ecuación 4 Ecuación de cálculo de I2 del LM2917.*

Fuente: Datasheet Lm2917.

$$I_2 = 100 * 125nf * 12$$

$$I_2 = 0,00015 A$$

$$I_2 = 0,15 mA$$

Calculo de C2:

$$C_2 = \frac{V_{cc}}{2} * \frac{C_1}{V_{rizado}} * \left(1 - \frac{V_{cc} * f_{min} * C_1}{I_2}\right) \dots (5)$$

*Ecuación 5 Ecuación de cálculo de C2 del LM2917.*

Fuente: Datasheet Lm2917.

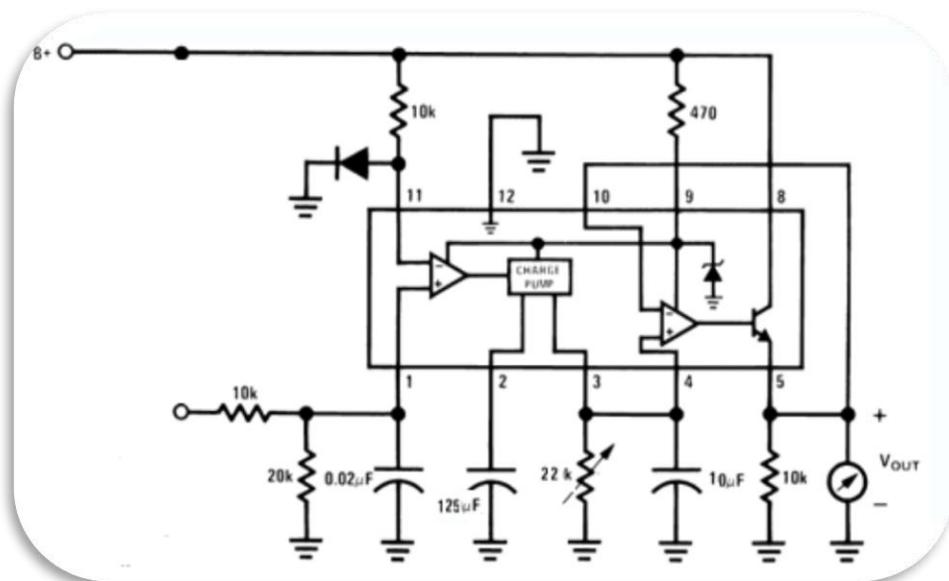
$$C_2 = \frac{12v}{2} * \frac{125nf}{0,05v} * \left(1 - \frac{12 * 50 * 125nf}{0,00015}\right)$$

$$C_2 = \frac{0,0000015}{0,1} * \left(1 - \frac{0,000075}{0,00015}\right)$$

$$C_2 = 0,0000075 F$$

$$C_2 = 7,5 uf$$

Con los valores obtenidos se procede a realizar un diagrama de conexión del LM2917, a continuación se puede apreciar dicho ejemplo.



*Figura 28 Ejemplo de conexión del LM2917.*

Fuente: El Autor.

## **2.4 SELECCIÓN DEL LENGUAJE DE PROGRAMACIÓN**

El lenguaje de programación consiste en un conjunto de órdenes o comandos que describen el proceso deseado de ingresar las señales analógicas provenientes de los sensores de temperatura, posición, velocidad y distancia, convertirlas en señales digitales y presentarlas de manera gráfica cuantitativamente de acuerdo con cada una de las magnitudes. Este lenguaje posee instrucciones y articulados característicos, que se disponen para establecer los programas de cómputo demandados. Cabe destacar que el lenguaje de programación que se utiliza no es una aplicación, sino una herramienta que permiten construir y adecuar las distintas etapas que demanda la conversión de analógico en digital.

Se trata de un lenguaje de alto nivel en que se escribe un programa que resulta de la agrupación de varias instrucciones, éste es traducido a lenguaje máquina antes de que pueda ser ejecutado. Esta conversión de instrucciones de alto-nivel a, instrucciones a nivel de máquina se realiza por el programa de software del sistema, denominado compilador e intérprete. Estos programas especiales son los denominados traductores. (Guzdial & Barbara, 2013).

Para una adecuada determinación de que lenguaje es la mejor opción para desarrollar el proyecto es necesario la descripción de los lenguajes más utilizados en la actualidad y así establecer las prestaciones de cada uno de estos como antelación a la descripción de los criterios de selección del software adecuado.

### **2.4.1 LENGUAJE C**

Se trata de un lenguaje de propósito general que permite controlar con gran precisión los factores que influyen en la eficiencia de los programas. Pero esta capacidad de control fino que ofrece este lenguaje tiene un precio: la escritura de programas puede ser mucho más costosa, pues hemos de estar pendientes de numerosos detalles. Tanto así que muchos programadores

afirman que no es un lenguaje de alto nivel, sino de nivel intermedio. (Kernighan & Ritchie, 1991).

### **2.4.2 LENGUAJE C++**

Se refiere a un lenguaje de programación, creado a mediados de 1980 por Bjarne Stroustrup, como extensión del lenguaje C. Este lenguaje abarca tres paradigmas de la programación:

- ✓ Programación Estructurada
- ✓ Programación Genérica
- ✓ Programación Orientada a Objetos

En la actualidad, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones, ya sea en Windows o GNU Linux. (Joyanes, Rodríguez, & Fernández, 1996).

### **2.4.3 LENGUAJE JAVA**

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como pueden ser pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos. Al ser un lenguaje de tipo script significa que no es un lenguaje compilado, es decir, tal cual se va leyendo se ejecuta por el cliente. Al estar basado en objetos, no vamos a tener toda la potencia que estos nos dan en Java, sólo algunas de sus características. El programa que va a interpretar los JavaScript es el propio navegador, lo que significa que si el nuestro no soporta JavaScript, no podremos ejecutar las funciones que programemos. (Kernighan & Ritchie, 1991)

#### **2.4.4 LENGUAJE PYTHON**

Python es un potente lenguaje de programación, muy fácil de aprender. Cuenta con estructuras de datos de alto nivel eficientes y un enfoque simple pero efectivo de programación orientada a objetos. Python contiene una elegante sintaxis y tipado dinámico, junto con su naturaleza interpretada, lo convierten en un lenguaje ideal para secuencias de comandos y el desarrollo rápido de aplicaciones en muchas áreas en la mayoría de las plataformas. Python es un lenguaje de programación que nos permitirá trabajar con mayor rapidez e integrar nuestros sistemas con mayor eficacia. Se puede aprender a utilizar Python y ver un incremento casi inmediato en la productividad y menores costos de mantenimiento. (Guzdial & Barbara, 2013).

Es necesario escoger un Lenguaje de Programación adecuado para elaborar la parte lógica del módulo didáctico. Éste deberá tener la capacidad de acceder a los registros GPIO de los pines de los módulos de hardware libre, brindar una estructura amigable y de fácil entendimiento para los usuarios finales, así como la facilidad de procesar interacciones graficas de estos con las magnitudes físicas que se requiere convertir y graficar dinámicamente.

#### **2.4.5 LENGUAJE DE PROGRAMACIÓN SELECCIONADO. PYTHON**

“Es un lenguaje inventado por Guido van Rossum. Él nombró este lenguaje en honor al famoso grupo humorístico inglés Monty Python. Este lenguaje se ha utilizado durante años por personas sin una capacitación formal en ciencias computacionales; está orientado a ser fácil de usar.” (Guzdial & Barbara, 2013, p.40).



*Figura 29 Logotipo del lenguaje de programación Python*

Fuente: <http://www.eweb.unex.es/eweb/sage/sagepython2014/>

Python es un lenguaje interpretado, lo que significa que las instrucciones se ejecutan sobre la marcha del código esto quiere decir que no es necesario compilar el código antes de ejecutarlo y la ejecución se puede hacer directamente desde el código crudo. Esto es particularmente interesante para los desarrolladores inexpertos. Podemos escribir el código, empezar a correr, modificar o corregir y volver a ejecutar. La forma de procesamiento del lenguaje Python es en modo consola ejecutable, resultado directo del lenguaje interpretado. Esto hace que sea muy fácil escribir sus primeras líneas de código a tan solo pocos minutos después de la instalación. Muy útil también para utilizar la consola de Python en cualquier momento para poner a prueba una nueva instrucción que uno aun no reconoce o el resultado de una línea de código. El lenguaje Python es un lenguaje llamado de alto nivel que es mucho más flexible que un lenguaje como C / C ++ o Java Management, debido a los tipos de parámetros que se pasan a las funciones ya que tiene varias características nativas de gran alcance incluidas las listas equivalentes a tablas de tamaño variable o el bucle que resulta de sus capacidades extendidas. (Guzdial & Barbara, 2013)

“Un lenguaje interpretado o de script es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados).” (González R., 2009, p. 120).

La sintaxis básica del lenguaje Python se simplifica evitando el uso de {}. La estructuración del código se basa sólo en sangría, es decir, el uso de tabulaciones simples, un poco confuso al principio, pero con la utilización de esta sintaxis se convierte con facilidad a códigos estructurados visualmente. El espacio de desarrollo mínimo para la codificación de inicio es fácil de instalar, de hecho una sintaxis sencilla en el editor de texto es suficiente. También concurren plugins IDE completas para plataformas de programación. Esta fase es muy importante, especialmente en la etapa de descubrimiento y pruebas. Es posible ir hasta una instrucción en concreto y no perderse en los sistemas y procedimientos complejos que no son familiares. Nada que ver por ejemplo con la instalación necesaria para escribir una primera aplicación en código Java, C, C++. (Guzdial & Barbara, 2013).

Python utiliza pocos recursos del sistema, lo que representa gran versatilidad, sobre todo en los pequeños módulos de hardware libre como Pduino y cubietruck. Tiene muchos plugins para administrar el sistema, red, puerto serie, sonido, cálculos matemáticos y la facilidad para exportar muchas bibliotecas esenciales y útiles como por ejemplo Python-Qt para la implementación de interfaces gráficas, OpenCV para el procesamiento de imágenes y visión por computador, y OpenGL para representaciones 3D.

Se debe tener en cuenta el uso de Python en el caso de un sistema GNU/Linux instalado en el Pduino o cubietruck es particularmente simple, lo que genera una reputación insuperable y es ampliamente utilizado en la ciencia o por organizaciones de renombre tales como la NASA o ILM (Lucasfilm).

#### **2.4.5.1 Reglas sintácticas básicas del lenguaje Python y una comparación simplificada con C y lenguajes Java**

La forma más fácil de entender la sintaxis del lenguaje Python es comparar a la de un lenguaje conocido como C o Java. Describir los elementos específicos del lenguaje Python. (González R. , 2009)

### 2.4.5.1.1 Sangría

Tabla 8 Tabla comparativa de sintaxis.

	<b>Python</b>	<b>C / C++ / Arduino</b>	<b>Java</b>
<i>Sangría</i>	Las secciones de código se definen encerradas por una incisión o etiquetas	Las secciones de código se definen encerradas por {}: cualquier aparato {} debe ser un cerrado correctamente.	Las secciones de código se definen encerradas por {}: cualquier aparato {} debe ser un cerrado correctamente.
	Nivel 1	{	{
	Nivel 2	Nivel 1{	Nivel 1{
	Nivel 2	Nivel 2	Nivel 2
	Nivel 3	Nivel 2	Nivel 2
	Nivel 3	Nivel 2 { Nivel 3	Nivel 2 { Nivel 3
	Nivel 2	}	}
	Nivel 1	}	}
	Nivel 2	}	}
	Nivel 3	}	}
		}	}
Comentario	# Comentario línea	// Comentario línea	// Comentario línea
	""" comentario multilínea """	/* comentario multilínea */	/* comentario multilínea */
Fin de la línea	No se utiliza coma, sólo el retorno a la etiqueta al final de la línea	Punto y coma al final de la línea	Punto y coma al final de la línea
	Instrucción1 Instrucción2	Instrucción1; Instrucción2;	Instrucción1; Instrucción2;

Fuente: El Autor.

### 2.4.5.2 El código mínimo Python

El código Python mínimo ejecutable simplificado de las principales formas sintácticas del lenguaje se detalla en la tabla 9, en comparación con el C / C ++ y Java. El objetivo aquí es mostrar la correspondencia entre los lenguajes más conocidos y Python.

Tabla 9 Código elemental de ejecución en python.

Código	Descripción
<code>#!/usr/bin/python</code>	Indicar la ruta de
<code># -*-coding:UTF-8 -*</code>	acceso al intérprete de Python y la codificación a utilizar.
<code>#-- código de las clases se debe escribir aquí</code> <code># ...</code>	Aquí el código principal: funciones, clases, etc.
<code>--- requerido para hacer que el código sea ejecutable ---</code> <code>if __name__ == "__main__": # si esta condición se cumple</code> el archivo ejecuta el programa <code># Código a ejecutarse al inicio del programa, se escribe aquí.</code> <code>exit; # fin del programa</code>	Se requiere de la condición final para hacer que el código sea ejecutable.

Fuente: El Autor.

### 2.4.5.3 Variables en Python

Una variable es un espacio de memoria reservado en el ordenador para contener valores que pueden cambiar durante la ejecución de un programa. Los tipos que se le asignen a estas determinan como se manipulara la información contenida en ellas. Cada variable necesita un identificador que la distingue de las demás. Un identificador válido es una secuencia de una o más letras, dígitos o guiones bajos, recordando que no deben coincidir con palabras reservadas del lenguaje, deben comenzar por una letra. Una de las características más potentes de los lenguajes de programación es la capacidad de manipular variables. En definitiva una variable es un nombre que hace referencia a un valor. (Guzdial & Barbara, 2013).

Tabla 10 Variables utilizadas en lenguaje Python.

Python	C / C++ / Arduino	Java
<code>a=2 # int</code>	<code>int a=2 ; // int</code>	<code>int a=2 ; // int</code>
<code>b=2.0 # float</code>	<code>float b=2.0 ; // float</code>	<code>float b=2.0 ; // float</code>

Fuente: El Autor.

#### 2.4.5.4 Cadenas de caracteres, funciones, clases y bucles

Las cadenas no son más que texto encerrado entre comillas simples ('cadena') o dobles ("cadena"). Dentro de las comillas se pueden añadir caracteres especiales escapándolos con \, como \n, el carácter de nueva línea, o \t, el de tabulación.

“Una cadena puede estar precedida por el carácter u o el carácter r, los cuales indican, respectivamente, que se trata de una cadena que utiliza codificación Unicode y una cadena raw. Las cadenas raw se distinguen de las normales en que los caracteres escapados mediante la barra invertida (\) no se sustituyen por sus contrapartidas.” (González R., 2009, p. 128).

*Tabla 11 Cadenas de caracteres, funciones, clases y condiciones en Python.*

	Python	C / C++ / Arduino	Java
<b>Cadenas</b>	myStr = "hola"	String myStr="hola";	String myStr="hola";
<b>Funciones</b>	def nombreFunción (self, param1, param2): código ... ..	nombreFunción(type1 param1, type2 param2) { código ... .. }	nombreFunción(type1 param1, type2 param2) { código ... .. }
<b>Clases</b>	class Persona(objeto): # crea la clase def __init__ (self, count): #Constructor código .. .. def funciónClass (self, param1, param2): código ..		public class Chain{// creación de clases // Las variables de clase private Person first = null; public Chain(type param)// constructor { código ..... } // Fin constructor public funciónClass (tipo param) { código ..... } // Función de clase End } // Fin de la clase
<b>Condiciones</b>	if condición: código ... .. else: código	if (condición) { código ... .. else {	if (condición) { código ... .. else {

	...	código	código
	...	...	...
	...	...}	...}
<b>Bucles</b>	for counter in range(1, 6):  código  ... ...	for (int i = 0; i < 100; i++) {  código  ... ...}	for (int i = 0; i < 100; i++) {  código  ... ...}
<b>listas</b>	xx = array ([val1, val2, val3]) a = [5, 0, 3, 9, 4, 3, 7] a.append(15) ## a = [5, 0, 3, 9, 4, 3, 7, 15]  a.index(3) # =>2	int array[10];	int [] array; // Declaración de una tabla array = new int [10]; // Inicializar tabla

Fuente: El Autor.

### 2.4.5.5 Uso de módulos de Python

Para facilitar el mantenimiento y la lectura los programas demasiados largos pueden dividirse en módulos, agrupando elementos relacionados. Los módulos son entidades que permiten una organización y división lógica de nuestro código. Los ficheros son su contrapartida física: cada archivo Python almacenado en disco equivale a un módulo.

*Tabla 12 Manera de importa los módulos en el lenguaje de programación Python.*

<b>Python</b>	<b>C / C++ / Arduino</b>	<b>Java</b>
import Módulo # importaciones de módulos y de todas las clases disponibles. from Módulo import *# importar todas las clases del Módulo - accesible en la clase de forma directa. from Módulo import a,b,c# reciben clases a, b, c del Módulo - accesible en la clase de forma directa.	#include < Módulo >;	import Módulo;

Fuente: El Autor.

### 2.4.5.6 Ejemplo de un código Python tipo estructurado

```
#!/usr/bin/python
# -*-coding:UTF-8 -*

# Módulos para importar
# import Módulo

# Clase (s) del programa
class myApp:
    #--Funciones de Clases
    def ma_función(self,a,b):
        return(a*b)
#Funciones globales
def ma_función (a,b):
    return(a*b)

# clase principal
class main():
    print("prog mini Ok")
    #Llamada a la función global
    a=input("a ?")
    b=input("b ? ")
    c = ma_función(a,b) #llama a la función global
    print="",c

    #Llamada a la función Clase
    classe=myApp() # declarar instancia de clase
    a=input("a ?")
    b=input("b ? ")
    d=classe.ma_función(a,b) # llama a la función de clase
    print="",d

#-- Para que sea el código ejecutable --
if __name__ == "__main__": # esto es cierto si el archivo es el programa en ejecución
    main() #llamada a la clase main()
```

Figura 30 Ejemplo de aplicación de código con Python.

Fuente: El Autor.

## 2.5 PROGRAMA PARA DESARROLLAR LA INTERFAZ GRÁFICA DE USUARIO.

Una de las herramientas más indispensables para cubrir los requerimientos de este proyecto es el programa de desarrollo de la interfaz gráfica, el cual es el encargado de convertir toda la información digital en graficas que pueden ser relacionadas por el usuario con las magnitudes físicas medidas por los sensores.

### 2.5.1 QT

Se trata de una agrupación de herramientas de interfaz gráfica de usuario que contiene embebecimientos de conectores de red, hilos, expresiones regulares, bases de datos SQL, SVG, OpenGL, XML, un navegador web

completamente funcional, un sistema de ayuda, un marco multimedia, así como una variada recopilación de widgets GUI. (Crespo, 2015).

Las clases de Qt emplean un mecanismo signal/slot para la comunicación entre objetos que es de tipo seguro pero menos vinculados por lo que es fácil crear componentes de software reutilizables.

Qt también incluye Qt Designer, un diseñador de interfaz gráfica de usuario. También es posible añadir nuevos controles GUI escrito en Python para Qt Designer, como por ejemplo utilizar PyQt para generar código de interfaz gráfica de usuario en Python desde una aplicación creada en Qt Designer.

### **2.5.1.1 Qt Designer**

Qt Designer es una herramienta de Qt para el diseño y la creación de interfaces gráficas de usuario (GUI), se puede preparar y personalizar los widgets o diálogos, también se puede probarlos utilizando diferentes estilos y resoluciones.

Los Widgets y las formas creadas con Qt Designer se integran a la perfección con el código dispuesto para nuestra aplicación, con señales y slots de Qt, que además permite asignar fácilmente el comportamiento de los elementos gráficos. Todas las propiedades establecidas en Qt Designer se pueden cambiar de forma dinámica dentro del código. Además, algunas características extras como la organización de widget y plugins personalizados permiten utilizar nuestros propios componentes con Qt Designer. (Monk, 2013).

#### *2.5.1.1.1 Descubriendo la interfaz Qt Designer*

La interfaz Qt Designer puede parecer compleja por primera vez, pero en realidad, contiene una interfaz bastante simple y lógica. Distinguiendo: En la parte superior: Una barra de menú clásico y una barra de herramientas A la izquierda de la "Widget Box" que contiene todos los gráficos que se pueden utilizar para las interfaces.

En la derecha, tenemos: Una ventana "Inspector de objeto" que contiene la lista de elementos que utiliza el diseño de la interfaz gráfica de usuario, el nombre y su clase. Una ventana "Editor de propiedades" que proporciona acceso a las propiedades del objeto actual. Una ventana con pestañas que es el Editor de señal, un Editor de acciones y explorador de recursos. Se debe tener en cuenta que el menú Display permite definir los diferentes elementos visibles o no. (PyQt4, 2015)

Por último, pero no menos importante, la zona central en la que el diseño se llevará a cabo, en realidad es forma como nosotros vamos a ver a la interfaz gráfica de usuario. La gran fuerza de Qt Designer se encuentra en esta zona: puesto que se va a añadir los widgets en su interfaz, guardar, y obtener el código en algunas líneas.

#### *2.5.1.1.2 Barra de menús*

En esta barra se encuentran siete pestañas, cada una serie de iconos de diversas funcionalidades. No se detallara todas las opciones de este menú, ya que se tendrá la oportunidad de descubrir a medida que se avanza con la investigación. De hecho, en la práctica, el menú se lo utiliza muy poco, principalmente se lo utiliza para guardar el archivo resultante. Otra característica útil es el menú Ver, que le permite configurar los elementos visibles se recomienda dejar el valor predeterminado.

#### *2.5.1.1.3 Barra de herramientas*

En la barra de herramientas, se puede señalar esencialmente los iconos de selección del modo activo de la interfaz de Qt-Designer. De forma predeterminada, se encuentra activa la función de diseño de gráficos. Para editar las señales, se lo realiza activando el icono de edición de señales.

#### *2.5.1.1.4 Barra de Widgets*

Esta ventana contiene todos los gráficos disponibles para su edición. La elección es importante de acuerdo a la aplicación que se requiera realizar. Se tiene una clasificación específica de Widgets, en realidad sólo unos pocos de widgets básicos son muy útiles, como por ejemplo Identificar un Label que es una simple etiqueta de texto, el LineEdit que es un campo de texto, el deslizador que es un ajuste horizontal o vertical entre muchos más.

Para poder utilizarlos, simplemente haga clic en el widget que desee y mantenga pulsado el botón izquierdo del ratón, mientras mantiene pulsado el botón izquierdo del ratón, mueva el cursor hasta el widget a la ventana básica presente en la zona de edición y suelte el botón del ratón, el widget se ha añadido a la interfaz. Finalmente queda hacer los ajustes de tamaño, definir propiedades y cambiar el texto del widget, se lo puede realizar haciendo doble clic en él para que sea editable modificar y validar dependiendo de los requerimientos. (Crespo, 2015).

#### *2.5.1.1.5 Inspector de objetos*

El Inspector de Objetos contiene la lista de los gráficos de la interfaz diseñada. Para cada widget, se encuentra: el nombre del objeto, el nombre de clase del objeto. El widget activo está resaltado.

A los objetos se les asignan automáticamente un nombre. Basta con utilizar el nombre predeterminado. Pero sería conveniente renombrar los elementos, se lo realiza haciendo doble clic sobre el nombre del objeto, se hace editable con lo que se procede a cambiar y validar.

#### *2.5.1.1.6 Editor de propiedades*

Esta ventana es una de las más grandes en el diseño de una interfaz: permite configurar gráficamente las propiedades de cada objeto. Se debe tener en cuenta que cualquier objeto, además de sus propias propiedades, también

hereda propiedades de las clases Generales. Esto da como resultado el hecho de que puede ser complicado de visualizar entre las propiedades de un objeto que sean las suyas, o las que ha heredado.

Esto tiene dos consecuencias muy prácticas:

- ✓ Se puede visualizar fácilmente cuando se utiliza el legado ascendente del objeto, y por lo tanto se puede reconocer las clases generales que hereda y que facilita la comprensión de la lógica del código, sobre todo la relación entre clases de objetos.
- ✓ Se distingue muy bien la jerarquía de las propiedades de un objeto, de un solo vistazo.

#### 2.5.1.1.7 Editor de señales

Se puede decir que esta ventana permite ver las relaciones que se han creado entre los objetos gráficamente.

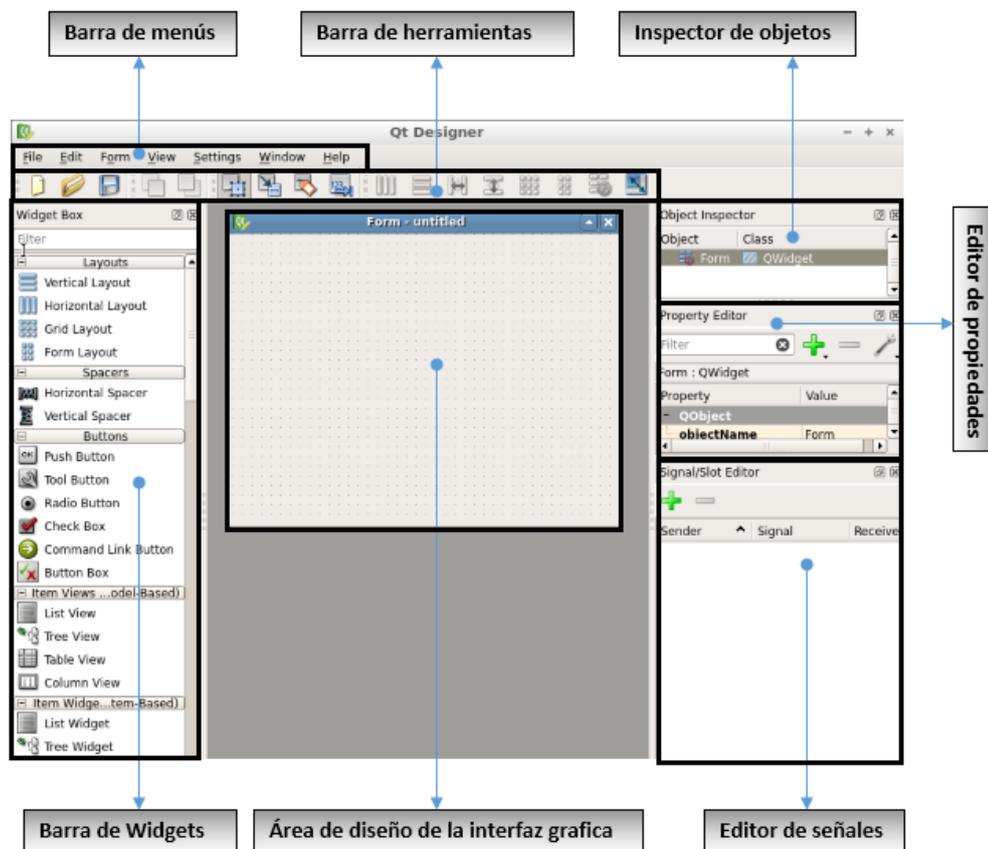


Figura 31 Zonas funcionales de Qt-Designer.

Fuente: El Autor.

### 2.5.1.2 PyQt

PyQt es una agrupación de enlaces (bindings) en Python para aplicaciones de Qt, funciona en todas las plataformas soportadas por Qt incluyendo Windows, MacOS / X y Linux. Los enlaces se efectúan como una agrupación de bloques de Python y se puede diferenciar más de 620 clases. Tiene licencia dual, no contiene una copia de Qt. Reúne un marco de aplicación multiplataforma Qt C ++ y lenguaje Python que es multiplataforma e interpretado. (PyQt4,8, 2015)

PyQt es capaz de generar código Python desde un archivo de Qt Designer. También es posible añadir nuevos controles GUI escritos en Python para Qt Designer. Adopta todas las superioridades de Qt y Python, de esta manera el usuario tiene todas las prestaciones de Qt, pero es capaz de utilizar con la sencillez de Python. También contiene una serie de programas de utilidad tales como:

- ✓ pyuic4: Convierte interfaces gráficas de usuario creadas con Qt Designer en código Python.
- ✓ pyrcc4: Incrusta recursos arbitrarios como Iconos, imágenes y archivos de traducción, descritos por un archivo de colección de recursos en un módulo de Python.

### 2.5.1.3 Pyduino

El objetivo de este apartado es dar una introducción rápida a la biblioteca Pyduino para conocer los alcances que dicha biblioteca puede tener en el desarrollo de la tesis.

Pyduino básicamente es una librería escrita en Python, es ligero de instalar, tiene bajo consumo de recursos y de gran alcance. Implementa la mayor parte de las funciones básicas de Arduino, así como las funciones que permiten el uso de funciones del sistema operativo.

Así que con pyduino es posible:

- ✓ Escribir un código de Arduino en un miniPC.
- ✓ Se puede tener acceso simultáneo a los pines de E / S y las características del sistema o multimedia en un mismo código.
- ✓ El uso de las funciones de red, archivos, son directamente accesibles.

Se elige utilizar pyduino puesto que cada mini-PC tiene sus propias características físicas de acceso a las E / S y los bloques de terminales son más o menos difíciles de utilizar en la práctica. No es fácil cuando se escribe un código, integrar simultáneamente el uso de las funciones del sistema y pines E/S. Y durante el uso, una solución viable en una plataforma con recursos limitados como en un mini-PC, es la simplicidad del sistema que lo podemos lograr gracias a la utilización de pyduino.

Ahora bien si nos preguntamos ¿Para qué utilizar Pyduino con un mini PC?, se podría decir que para todos los proyectos en los que las prestaciones se están empezando a acumularse, ya que podríamos unificarlas en un solo conjunto de instrucciones, además si se desea utilizar registros de los datos de los pines de E/S fácilmente se adjuntarían las instrucciones.

La base para utilizar un mini-PC con Pyduino, es solamente necesario, una tarjeta mini-PC con un sistema operativo GNU / Linux instalado; se recomienda normalmente GUI LXDE.

La biblioteca Pyduino incluye funciones de Arduino más nativos sin cambios en su sintaxis, añadiendo nuevas características útiles y su integración en un espacio de desarrollo más potente, el lenguaje Python. Las funciones Pyduino, tienen instrucciones Arduino basadas en el lenguaje Python y por lo tanto son compatibles con todas las construcciones estándar del mismo lenguaje. Python es un lenguaje interpretado que se basa en el uso de un intérprete de Python que ejecuta instrucciones de una en una. Esto permite en particular que una instrucción de prueba o un pedazo de código ejecutarse en el intérprete de Python directamente. Python, es la base de la biblioteca Pyduino, es también un lenguaje muy poderoso en sus posibilidades, más fácil de utilizar en la gestión de los tipos de variables, lo que le permite interactuar con el sistema. Por último, la biblioteca Pyduino permite en el mismo código de acceso

a los pines de E/S de las mini-PCs, las funciones del sistema de archivos o multimedia. (LinkSprite, 2015).

*Tabla 13 Tabla de funciones más comunes utilizadas de Pyduino.*

	<b>Tipo</b>	<b>Descripción</b>
<b>Estructura</b>	def :	Declaración de una función
	setup()	La función setup () se llama cuando se inicia el programa.
	loop()	Ejecuta en un bucle sin fin.
<b>Variables, constantes, objetos</b>	HIGH, LOW	Significado diferente dependiendo de si el pin es configurado como entrada o salida.
	INPUT	Los pines de una tarjeta mini-PC se configuran en el modo INPUT (= entrada) utilizando pinMode () son llamados en un estado de alta impedancia.
	OUTPUT	Los pines configurados en modo OUTPUT (= salida) con pinMode () están en un estado llamado baja impedancia.
<b>Funciones de pines E/S</b>	pinMode()	Establece el pin especificándose como debe comportarse ya sea de entrada o salida.
	digitalWrite()	Si el pin se ha configurado con salida con pinMode (), su voltaje se establece en el valor correspondiente: 3.3V normalmente en tarjetas miniPC de HIGH, 0V (tierra) para LOW.
	digitalRead()	Lee el nivel lógico de un pin de entrada digital, y devuelve el valor HIGH o LOW.
	analogRead()	Lee el valor de la tensión en el pin especificado. Esta función devuelve un valor "en bruto", que es una función de la resolución del pin utilizado.
	analogWrite()	Genera una pulso/ período anchura deseado en un pin de la placa, onda PWM.
	analogWritePercent()	Función variante de analogWrite que recibe un valor de porcentaje para expresar el ancho de pulso de una onda PWM.

Fuente: El Autor.

Para convertir el código de Arduino en un código de Pyduino, existen cinco reglas sencillas para hacerlo, suficientemente simplificado:

- ✓ Void se convierte en def
- ✓ Inicio {llave de inicio, y llave de cierre} desaparecen.
- ✓ Las líneas dentro de un bloque de instrucción deben estar precedidas por un espacio equivalente a un Tab.
- ✓ ; la instrucción End desaparece, solamente el código debe estar alineado correctamente.
- ✓ // para comentarios se lo realiza con #.

### 3 IMPLEMENTACIÓN DEL MÓDULO DIDÁCTICO

#### 3.1 PCDUINO PRIMEROS PASOS

Pcduino es una potente plataforma que combina las ventajas de un mini-PC basado en ARM y el ecosistema Arduino, representa una pc + Arduino. Pcduino contiene un grupo de pines compatible con Arduino por lo que se puede utilizar el IDE de Arduino entre otras librerías que se pueden instalar. Al ser compatible con el IDE de Arduino, es una plataforma que une el poder de software abierto Linux y la potencia del hardware abierto.

Para iniciar a utilizar el pcdduino de forma correcta se debe actualizar los paquetes:

- ✓ Se introduce el comando `sudo board-config.sh` en consola.

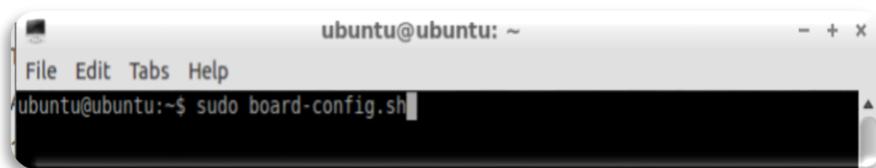


Figura 32 Consola de pcdduino con el comando `sudo board-config.sh`.

Fuente: El Autor.

- ✓ Con las teclas para arriba o para abajo, nos posicionamos en `update`.

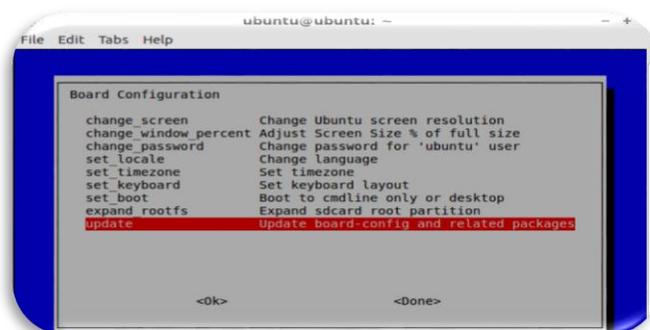


Figura 33 Consola de Pcdduino con las opciones de configuración del dispositivo.

Fuente: El Autor.



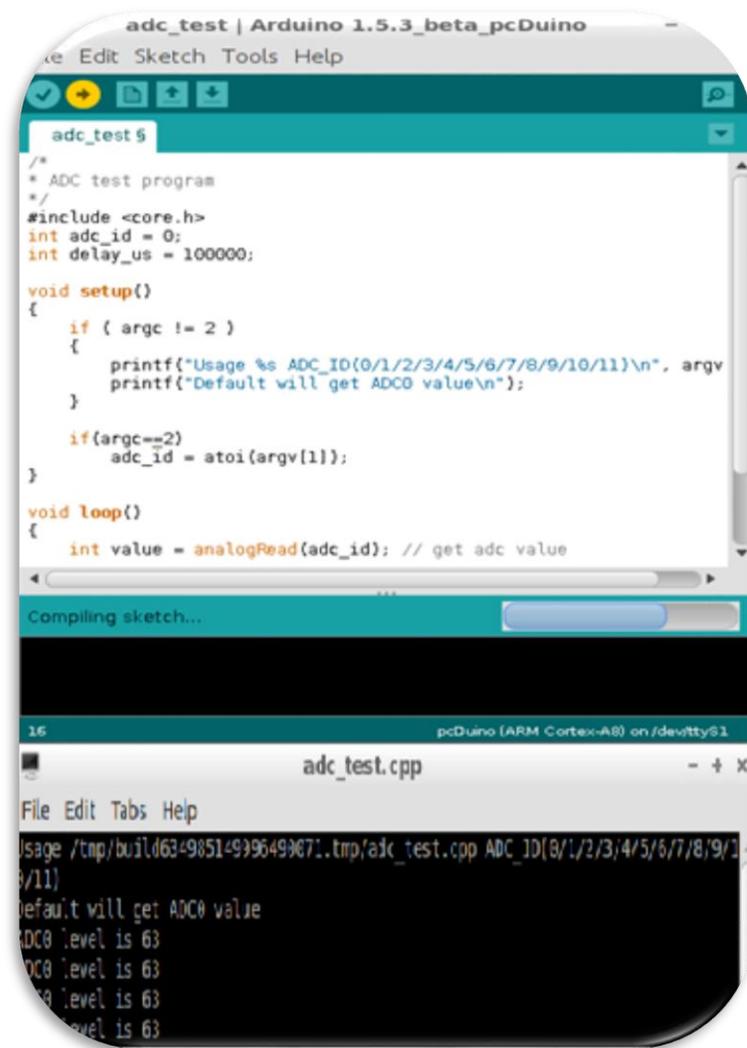


Figura 35 Ejemplo de ejecución de Arduino IDE en el Pcdduino.

Fuente: El Autor.

### 3.1.2 VERIFICACIÓN DE LOS MÓDULOS DE PCDUINO

Una forma de verificar que los módulos de nuestro pcdduino estén en correcto funcionamiento es ejecutando en consola: `sudo modprobe hardwarelib` o verificar el archivo donde se encuentran los módulos con el comando en consola: `nano /etc/modules`.

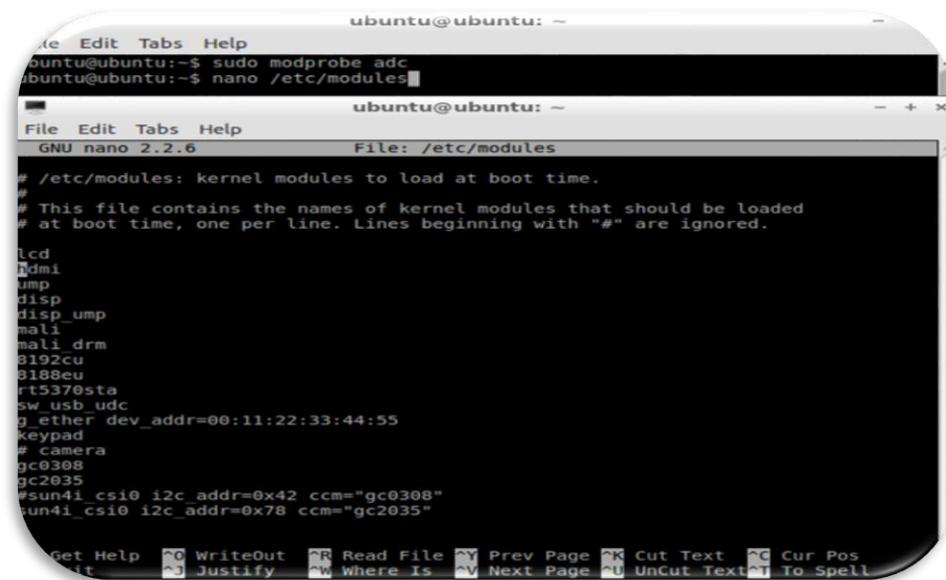


Figura 36 Formas de verificar el funcionamiento de los módulos del Pcdduino.

Fuente: El Autor.

### 3.1.3 VERIFICACIÓN Y PUESTA EN MARCHA DE PYTHON EN EL PCDUINO

Ahora bien el lenguaje de programación para realizar el trabajo de grado es python como anteriormente se describió, viene preinstalado en nuestra distribución de Lubuntu, para verificar su versión y correcto funcionamiento solamente basta con introducir en consola el comando: Python.

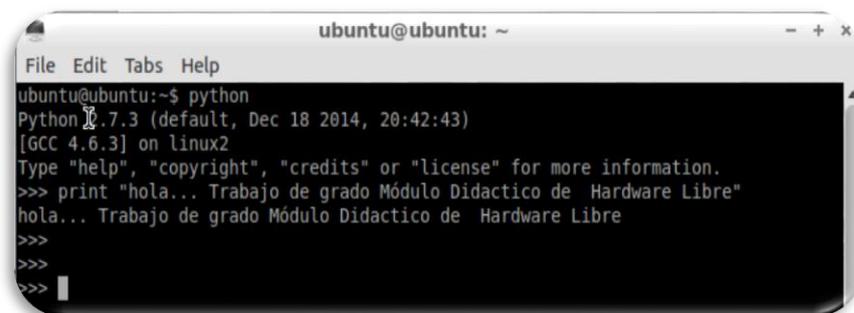
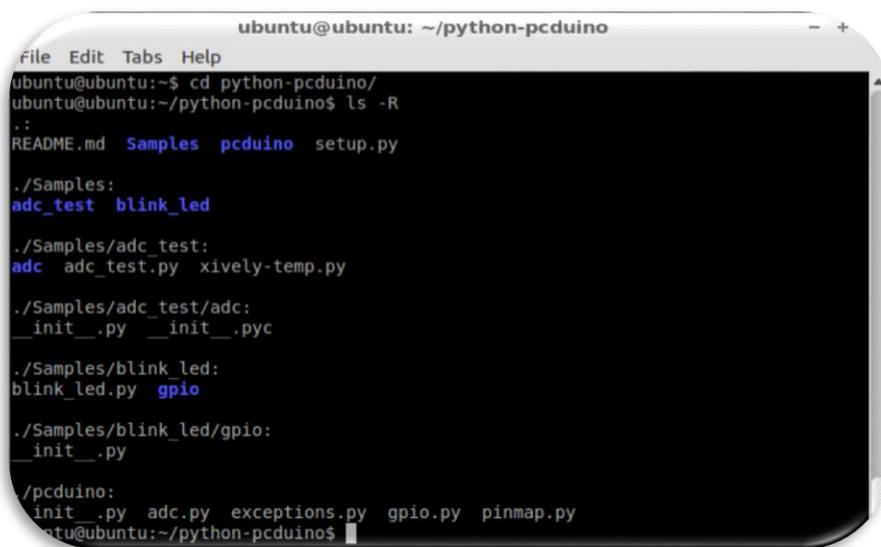


Figura 37 Consola de Pcdduino ejecutando Python.

Fuente: El Autor.

En nuestro pcdduino la utilización de Python-pcdduino se lo realiza clonando del repositorio github de pcdduino la carpeta python-pcdduino de la siguiente manera: `sudo apt-get install git; sudo git clone https://github.com/pcduino/python-pcdduino`; ingresamos a la carpeta clonada con el comando: `cd python-pcdduino`; una vez dentro de esta carpeta ejecutamos el comando `ls -R` para observar los subdirectorios existentes; finalmente para poder ejecutar un ejemplo debemos anteponer python al nombre del archivo a ejecutarse. Para mayor exactitud en el código python-pcdduino a utilizar diríjase a <https://github.com/pcduino/python-pcdduino>.



```
ubuntu@ubuntu: ~/python-pcdduino
File Edit Tabs Help
ubuntu@ubuntu:~$ cd python-pcdduino/
ubuntu@ubuntu:~/python-pcdduino$ ls -R
.:
README.md Samples pcdduino setup.py

./Samples:
adc_test blink_led

./Samples/adc_test:
adc adc_test.py xively-temp.py

./Samples/adc_test/adc:
__init__.py __init__.pyc

./Samples/blink_led:
blink_led.py gpio

./Samples/blink_led/gpio:
__init__.py

./pcduino:
__init__.py adc.py exceptions.py gpio.py pinmap.py
ubuntu@ubuntu:~/python-pcdduino$
```

*Figura 38 Archivos que contiene Python-pcdduino.*

Fuente: El Autor.

### 3.1.4 INSTALACIÓN DE PYDUINO EN NUESTRO PCDUINO

Vamos a instalar primero un editor de texto que servirá muy a menudo, llamado Geany con el comando: `sudo apt-get install geany –yes`.

Geany es un editor de texto útil para modificar fácilmente los archivos de configuración, también para escribir programas de control simple para los pines de E/S del pcDuino utilizando un GUIs en Python. La gran ventaja de Geany en comparación con otras soluciones de edición de texto, es que es muy ligero y

por lo tanto no utiliza muchos recursos, tiene un arranque instantáneo sin ninguna configuración especial.

Se requiere instalar gdebi para simplificar la instalación de paquetes .deb, el comando siguiente para la instalación gdebi y validar: `sudo apt-get install gdebi -yes`; A continuación, se debe descargar el archivo .deb del IDE Pyduino, actualmente se tiene la versión 0.4.

*Tabla 14 Comandos para descargar IDE Pyduino*

---

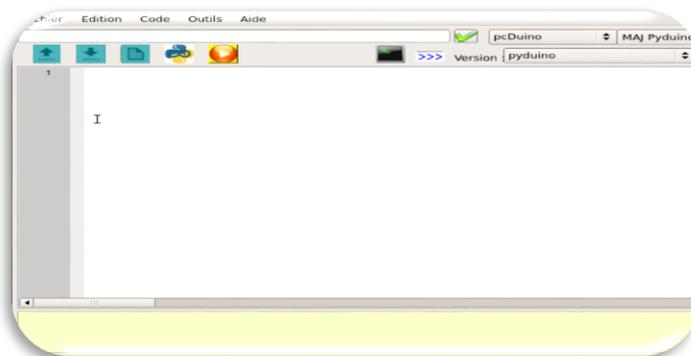
<code>cd Downloads</code>
<code>wget -4 -N http://www.cloud-mon-club</code>
<code>elec.fr/public.php?service=files\&amp;t=80e7cb812fbf2171356b8721c3bb7b27\&amp;download</code>

---

Fuente: El Autor

Nos dirigimos al directorio deseado en este caso a descargas y abrimos el paquete .deb con gdebi y procedemos a instalarlo.

Ahora, se puede abrir el IDE Pyduino, directamente desde: menú >Programming> IDE Pyduino. Se abrirá el IDE Pyduino, y se puede empezar a utilizarlo.



*Figura 39 Ventana Principal del IDE Pyduino.*

Fuente: El Autor.

### **3.1.5 INSTALACIÓN DE INTERFAZ GRÁFICA DE USUARIO, GUI (GRAPHICAL USER INTERFACE)**

Para realizar una interfaz gráfica de usuario muy necesaria para la realización de la tesis, se puede empezar su desarrollo con Python y Qt. Lo

primero que se debe hacer es instalar todo lo necesario para iniciar a desarrollar con Python y Qt en pcduino.

El diseño de la interfaz gráfica se lo realiza en Qt qt4-designer: Para crear y diseñar la interfaz gráfica de usuario del programa, vamos a utilizar una herramienta extremadamente cómodo y eficaz suministrada por Qt: llamada Qt Designer. La instalación se realiza con la siguiente línea de comandos: sudo apt-get install qt4-designer.

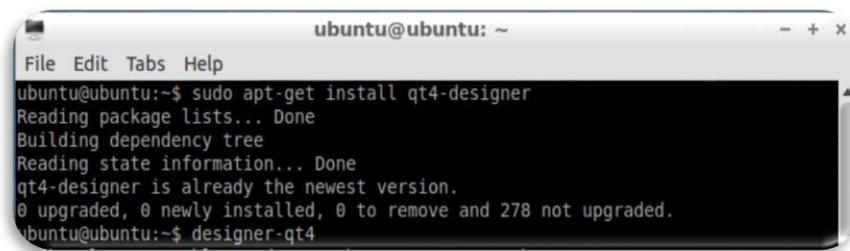


Figura 40 Consola con comando para instalar e inicializar Qt-Designer.

Fuente: El Autor.

Para iniciar Qt Designer, desde la línea de comandos con el comando: \$ designer-qt4; o ya sea desde el menú gráfico de la distribución: Aplicaciones> Programming / > Qt 4 Designer.

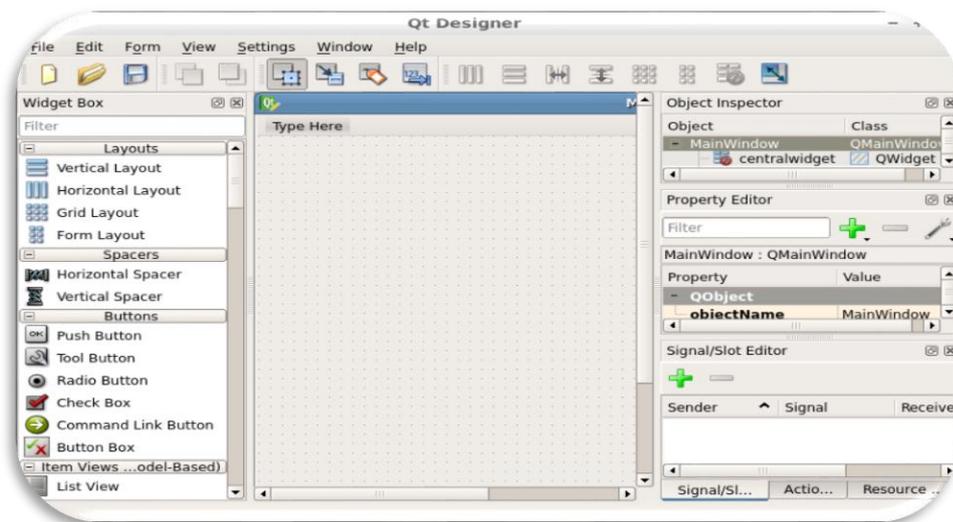


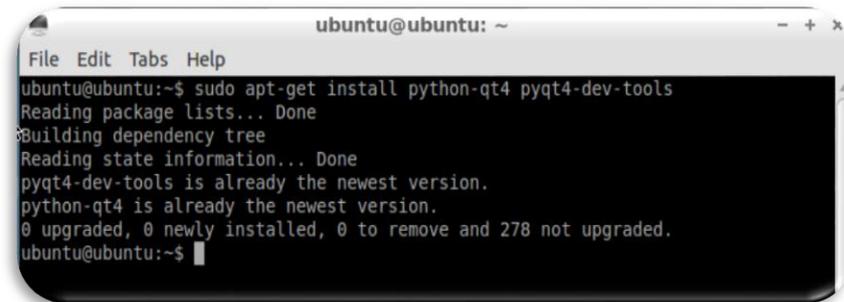
Figura 41 Ventana principal de Qt-Designer.

Fuente: El Autor.

### 3.1.6 INSTALACIONES DE PAQUETES COMPLEMENTARIOS

Los siguientes paquetes de Python también son muy útiles, se instalarán en el sistema: `sudo apt-get install python-qt4 pyqt4-dev-tools`

- ✓ python-qt4 (PyQt)
- ✓ PyQt4-dev-herramientas (varias paquetes de interés, incluyendo pyuic4)

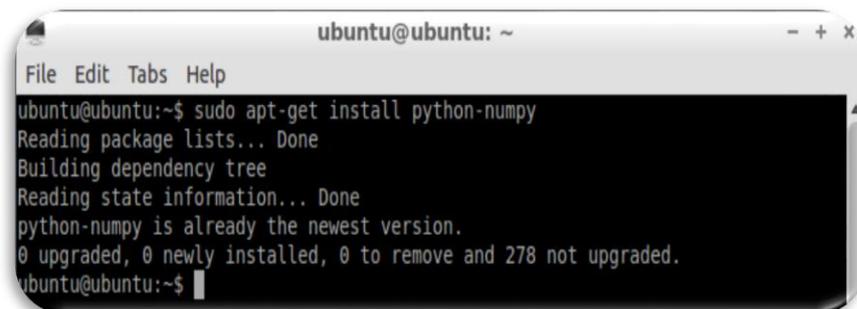


```
ubuntu@ubuntu: ~  
File Edit Tabs Help  
ubuntu@ubuntu:~$ sudo apt-get install python-qt4 pyqt4-dev-tools  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
pyqt4-dev-tools is already the newest version.  
python-qt4 is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 278 not upgraded.  
ubuntu@ubuntu:~$
```

*Figura 42 Consola con los comandos de instalación de paquetes complementarios.*

Fuente: El Autor.

Pyqtgraph es una biblioteca de Python basado en el lenguaje de Python Qt (PyQt) y permite realizar gráficas de tipo científico muy ligeras. También existe procesamiento de imágenes. Como requisitos de la biblioteca pyqtgraph, es necesario la instalación de los siguientes paquetes: `sudo apt-get install python-numpy Python-scipy`.



```
ubuntu@ubuntu: ~  
File Edit Tabs Help  
ubuntu@ubuntu:~$ sudo apt-get install python-numpy  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python-numpy is already the newest version.  
0 upgraded, 0 newly installed, 0 to remove and 278 not upgraded.  
ubuntu@ubuntu:~$
```

*Figura 43 Consola con el comando de instalar pyqtgraph.*

Fuente: EL Autor.

Es necesario descargar el código fuente e instalación de Pyqtgraph, Cuando se haya terminado, se puede probar fácilmente la biblioteca con los ejemplos: `python -m pyqtgraph.examples`.

*Tabla 15 Comandos de descarga e instalación de Pyqtgraph.*

cd Downloads
wget http://www.pyqtgraph.org/downloads/pyqtgraph-0.9.7.tar.gz
tar xvfz pyqtgraph-0.9.7.tar.gz
cd pyqtgraph-0.9.7
sudo python setup.py install

Fuente: El Autor.

### 3.1.7 RESUMEN DE LOS PROGRAMAS MÁS SIGNIFICATIVOS A UTILIZAR

En este apartado se ha realizado una tabla resumen de los programas más significativos a utilizar en la realización de este trabajo de grado.

*Tabla 16 Resumen de software a utilizar.*

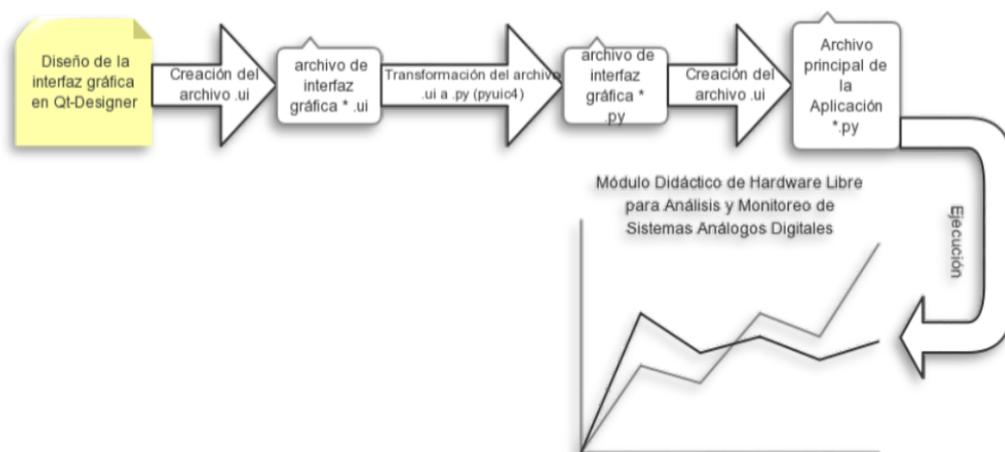
	<p>Python es un lenguaje de programación que utilizamos: es un lenguaje de alto nivel, de gran alcance, interpretado, multiplataforma. Este lenguaje puede ser utilizado de forma independiente, sin una interfaz gráfica de usuario si lo desea. Pero ese no es nuestro objetivo en este proyecto.</p>
<p>Fuente logo Python: <a href="https://www.python.org/static/community_logos/python-logo-master-v3-TM.png">https://www.python.org/static/community_logos/python-logo-master-v3-TM.png</a></p>	
	<p>Qt es una biblioteca escrita en C ++. Es multiplataforma. Hay otras librerías gráficas posibles pero Qt es el que parece tener la mejor portabilidad, ya que el entorno de escritorio KDE se basa en Qt.</p>
<p>Fuente logo qt: <a href="http://www.qt.io/application-development/">http://www.qt.io/application-development/</a></p>	
	<p>PyQt Python es un binding de la biblioteca Qt: se encontrará, algunas sintaxis acerca de adaptaciones de todas las clases de bibliotecas Qt en Python. Así que para defender la oportunidad y la máxima simplicidad, se optó por PyQt.</p>
<p>Fuente logo PyQt: <a href="http://pyqt.sourceforge.net/Docs/PyQt4/">http://pyqt.sourceforge.net/Docs/PyQt4/</a></p>	
	<p>Qt Designer es el software de diseño de interfaz gráfica de usuario suministrada con el lenguaje Qt: Este software permite crear de forma sencilla y rápida interfaces gráficas de usuario. Este software genera un archivo que describe el tipo de interfaz en formato XML *.ui Este archivo se puede utilizar tanto con el programa Qt C ++ o un programa con PyQt.</p>
<p>Fuente logo Qt-designer: <a href="http://doc.qt.io/qt-5/qt designer-manual.html">http://doc.qt.io/qt-5/qt designer-manual.html</a></p>	

Fuente: El Autor.

## 3.2 APLICACIONES DE INTERFAZ DE USUARIO EN QT-DESIGNER

El diseño de una interfaz gráfica de usuario en Python con Qt Designer de Qt responderá a las siguientes etapas:

- ✓ Diseño de la interfaz gráfica en Qt Designer.
- ✓ Archivar la interfaz bien diseñada en el archivo de descripción de interfaz \* .ui equivalente en XML.
- ✓ Conversión de la descripción del archivo \* .ui, a un archivo de Python que genere la misma interfaz.
- ✓ Creación de un archivo principal de Python en el que se incluya el archivo creado anteriormente y editar el código activo deseado en el archivo Python.
- ✓ Ejecución del archivo principal de Python, ya sea desde el editor, desde la línea de comandos o haciendo doble clic sobre el mismo.



*Figura 44 Diagrama de flujo del proceso de creación de interfaces gráficas de usuario.*

Fuente: El Autor.

A primera vista, esta secuencia puede parecer compleja, pero en realidad no lo es, porque el primer paso es gráfico, los pasos 2 y 3 automáticos, y sólo el paso 4 es activo en términos de estructuración del proyecto.

Por último, tenemos un directorio de aplicación que contiene 3 archivos:

- ✓ \* archivo .ui (XML) de la interfaz obtenida con Qt-Designer.
- ✓ .py (Python) de la interfaz obtenida con utilidad de conversión pyuic4.
- ✓ Main.py el archivo (Python) que contiene el código activo de estructuración del proyecto en sí mismo: es el archivo que se edita y se ejecuta.

De hecho, el archivo \*.ui, no es obligatorio para la ejecución pero muchos lo dejan en el directorio desde su creación. Y los otros 2 archivos son archivos de texto simple.

El uso de la aplicación creada será muy flexible, de hecho es posible copiar y pegar en el directorio deseado y ejecutar haciendo doble clic en el archivo principal con la extensión en Python. De la misma manera, podemos crear una nueva aplicación desde una existente por un simple copiar / pegar en el directorio y será suficiente cambiar los nombres de archivos antes de realizar la adaptación necesaria.

En nuestro caso, la interfaz a la que hemos añadido contiene los siguientes elementos:

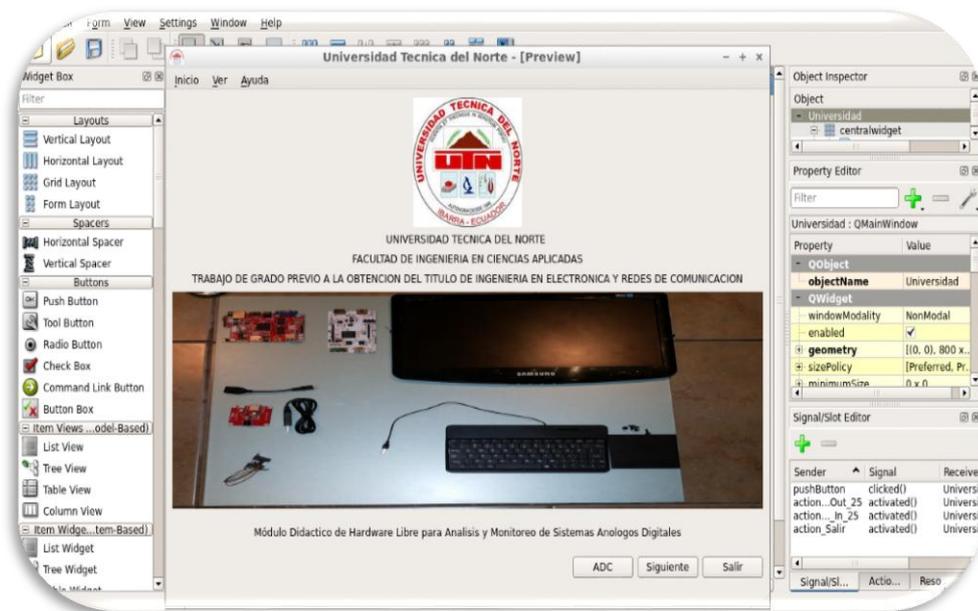


Figura 45 Menú principal de la aplicación del trabajo de grado.

Fuente: El Autor.

Se tiene que tener en cuenta que en cualquier momento, se puede obtener una vista previa de la interfaz usando el atajo de teclado CONTROL + R. Sólo se tiene que cerrar la ventana de visualización para volver a la interfaz de edición.

### **3.2.1 GENERACIÓN DEL ARCHIVO PYTHON DE LA INTERFAZ GRÁFICA DE USUARIO (GUI)**

Una vez que la configuración gráfica se ha completado, se podrá guardar el interfaz. El archivo se guarda en \*.ui, obtenido un archivo XML que se guarda en el directorio deseado. Este archivo describe la interfaz y todos los parámetros utilizados para crear la misma. Este archivo es un archivo de texto simple en formato XML.

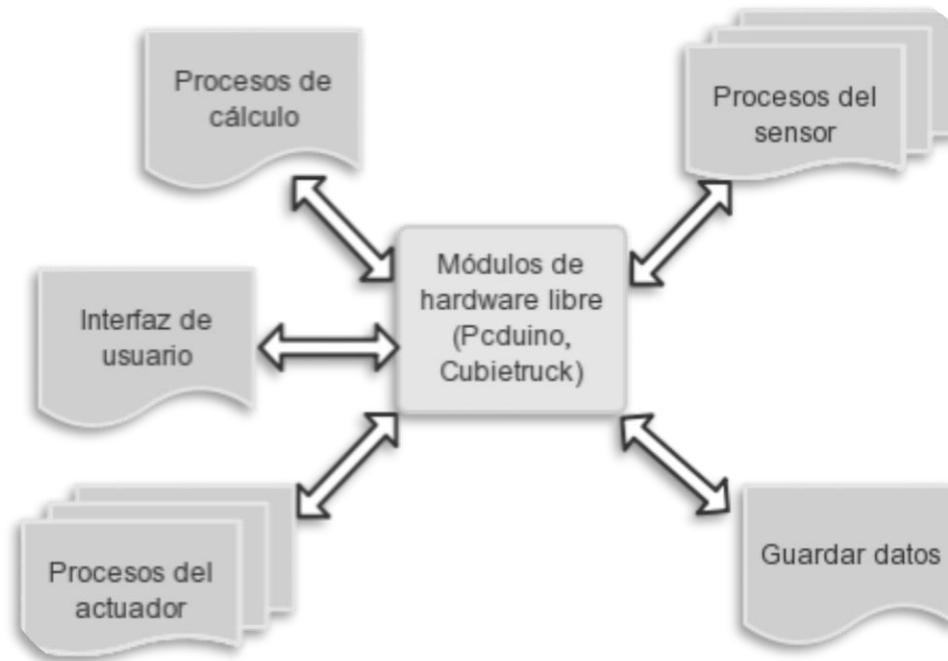
Para convertir el archivo XML \* ui \* en un archivo .py se lo realiza con el siguiente comando, se debe ubicarse primero en el directorio que almacena el archivo .ui con el comando cd y la ruta del mismo: pyuic4 -o nombreArchivo.py -x nombreArchivo.ui

En este comando -o indica el archivo en python de salida \* .py; -x para crear el archivo de destino con la función \_main\_, que hace que el código Python sea ejecutable; el archivo \* .ui es el archivo de la aplicación grafica creada en Qt-Designer.

### **3.3 CONTROL CENTRALIZADO PARA EL SISTEMA EN TIEMPO REAL DE ADQUISICIÓN DE DATOS DE LAS MAGNITUDES FÍSICAS TEMPERATURA, DISTANCIA, POSICIÓN, VELOCIDAD ANGULAR.**

La etapa de controlador del sistema resuelve cuándo inicializar o finalizar los procesos en función de los estados de los módulos, se realiza ciclos en forma continua para verificar los cambios de estados de los sensores. El sistema

verifica si nuevos procesos han originado información para ser procesada o para remitir información para su respectivo procesamiento.



*Figura 46 Control centralizado de adquisición de datos en tiempo real.*

Fuente: EL Autor.

### **3.3.1 FORMAS DE ADQUISICIÓN Y VERIFICACIÓN DE LOS ESTADOS DE LOS PINES ANALÓGICOS Y DE E/S**

#### **3.3.1.1 Medición de las entradas analógicas del pcDuino**

En los diagramas de flujos de la adquisición de los datos de las entradas analógicas, se pueden evidenciar los pasos a seguir para obtener los datos de los sensores analógicos de nuestro modulo didáctico; tanto de una forma cruda a través de la lectura de los scripts de los pines o de una forma simplificada a través de la utilización de la instrucción `analogRead()` asignándole a una variable.

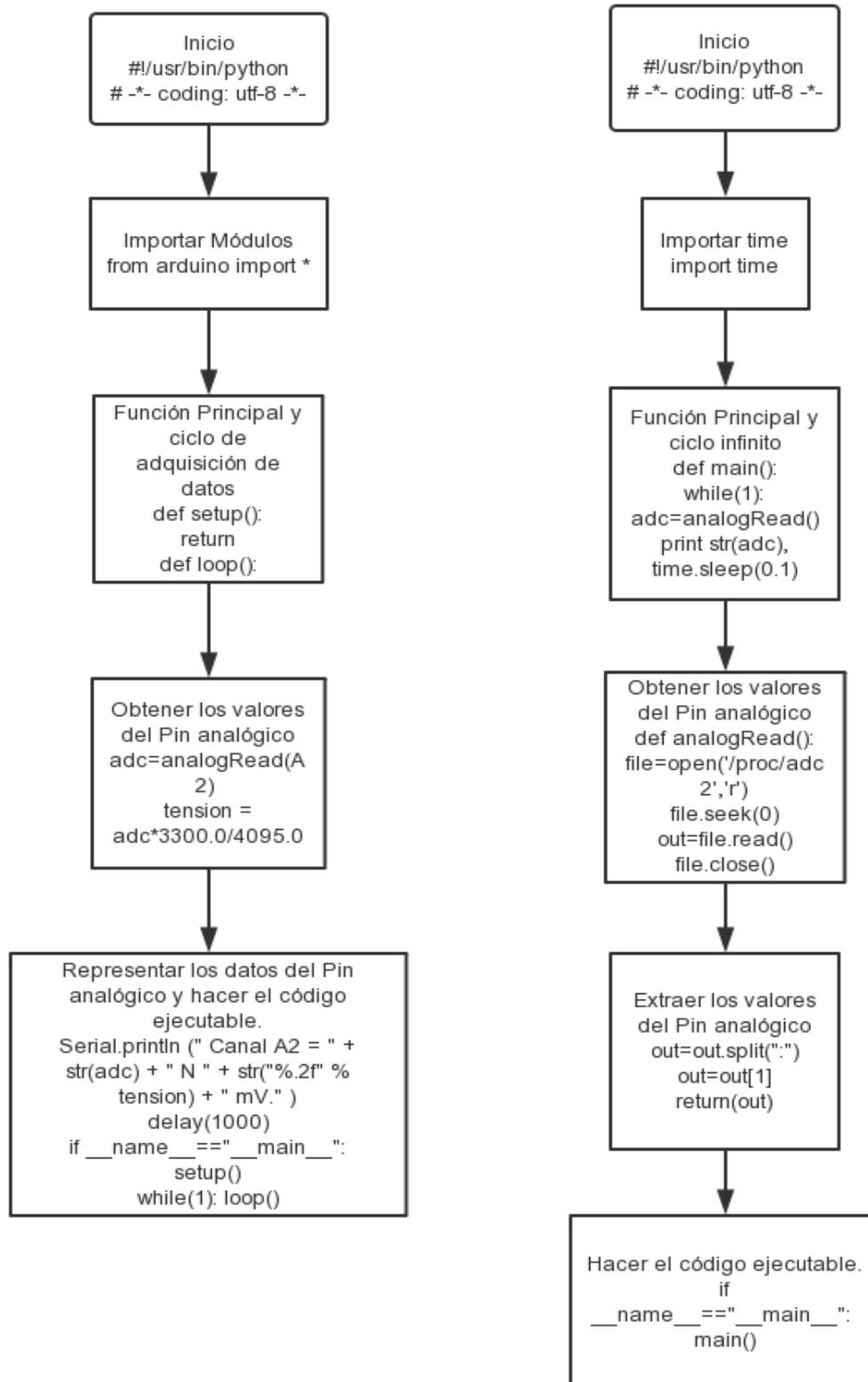


Figura 47 Forma simplificada y forma cruda de obtención de los valores de pines de las entradas analógicas.

Fuente: El Autor.

Dentro del bucle while (1) crear una función infinita nombrada analogRead(), se lee el archivo y se extraer el valor del pin de entrada analógica.

<pre>#!/usr/bin/python #-*- coding: utf-8 -*- import time  # -- Clase principal-- def main():     while(1):         adc=analogRead()         print str(adc),         time.sleep(0.1)  def analogRead(): # lectura del fichero     file=open('/proc/adc0','r')     file.seek(0)     out=file.read()     file.close() # extracción del valor     out=out.split(":")     out=out[1]     return(out)  if __name__=="__main__":     main()</pre>	<pre>#!/usr/bin/python #-*- coding: utf-8 -*- from pyduino_pduino import * #--- setup ---  def setup():     return #     # -- fin setup --  # -- loop -- def loop(): #voie A2     adc=analogRead(A2)     tension = acd*3300.0/4095.0 #      Serial.println (" Canal A2 = " + str(adc) + " valor " + str("%.2f" % tension) + " mV." )     delay(1000)# # -- fin loop --  if __name__=="__main__":     setup()     while(1): loop()</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*Figura 48 Obtención de los valores de las entradas analógicas del pduino.*

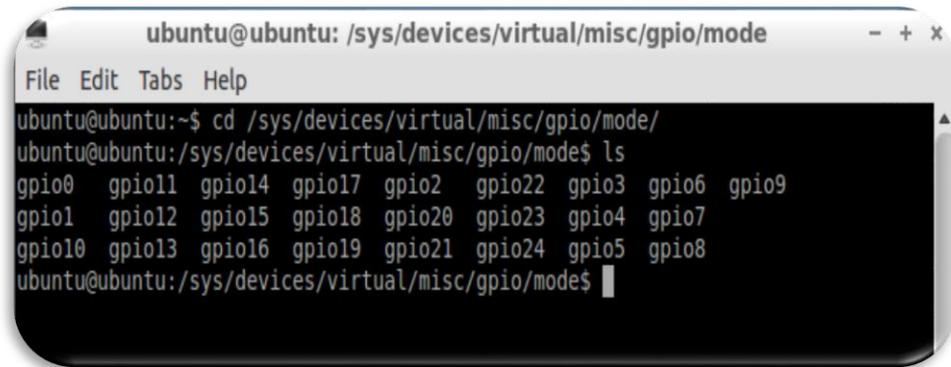
Fuente: El Autor.

Se puede destacar en el código que la única diferencia notable con el código de Arduino es el formato nativo Serial.println() la misma que nos ayuda como función del lenguaje Python. Se evidencia también cómo las declaraciones de variables se las realiza sin tener que especificar el tipo, he aquí la flexibilidad y simplicidad del lenguaje Python.

### 3.3.1.2 Acceso a los pines de E/S

Como ya se ha mencionado anteriormente, los pines de E/S se controlan mediante archivos simples. El registro de cada pin es la de la forma gpioX donde X es el número de los pines.

Los archivos en el modo de entrada o salida de los pines de E/S se colocan en el directorio: `/sys/devices/virtual/misc/gpio/mode/`



```
ubuntu@ubuntu: /sys/devices/virtual/misc/gpio/mode
File Edit Tabs Help
ubuntu@ubuntu:~$ cd /sys/devices/virtual/misc/gpio/mode/
ubuntu@ubuntu:/sys/devices/virtual/misc/gpio/mode$ ls
gpio0  gpio11  gpio14  gpio17  gpio2  gpio22  gpio3  gpio6  gpio9
gpio1  gpio12  gpio15  gpio18  gpio20  gpio23  gpio4  gpio7
gpio10  gpio13  gpio16  gpio19  gpio21  gpio24  gpio5  gpio8
ubuntu@ubuntu:/sys/devices/virtual/misc/gpio/mode$
```

*Figura 49 Consola con el directorio de los pines E/S.*

Fuente: El Autor.

Los archivos que contienen los scripts de los pines E/S, se pueden definir el estado de las mismas y se las coloca en el directorio: `/sys/devices/virtual/misc/gpio/pin/`

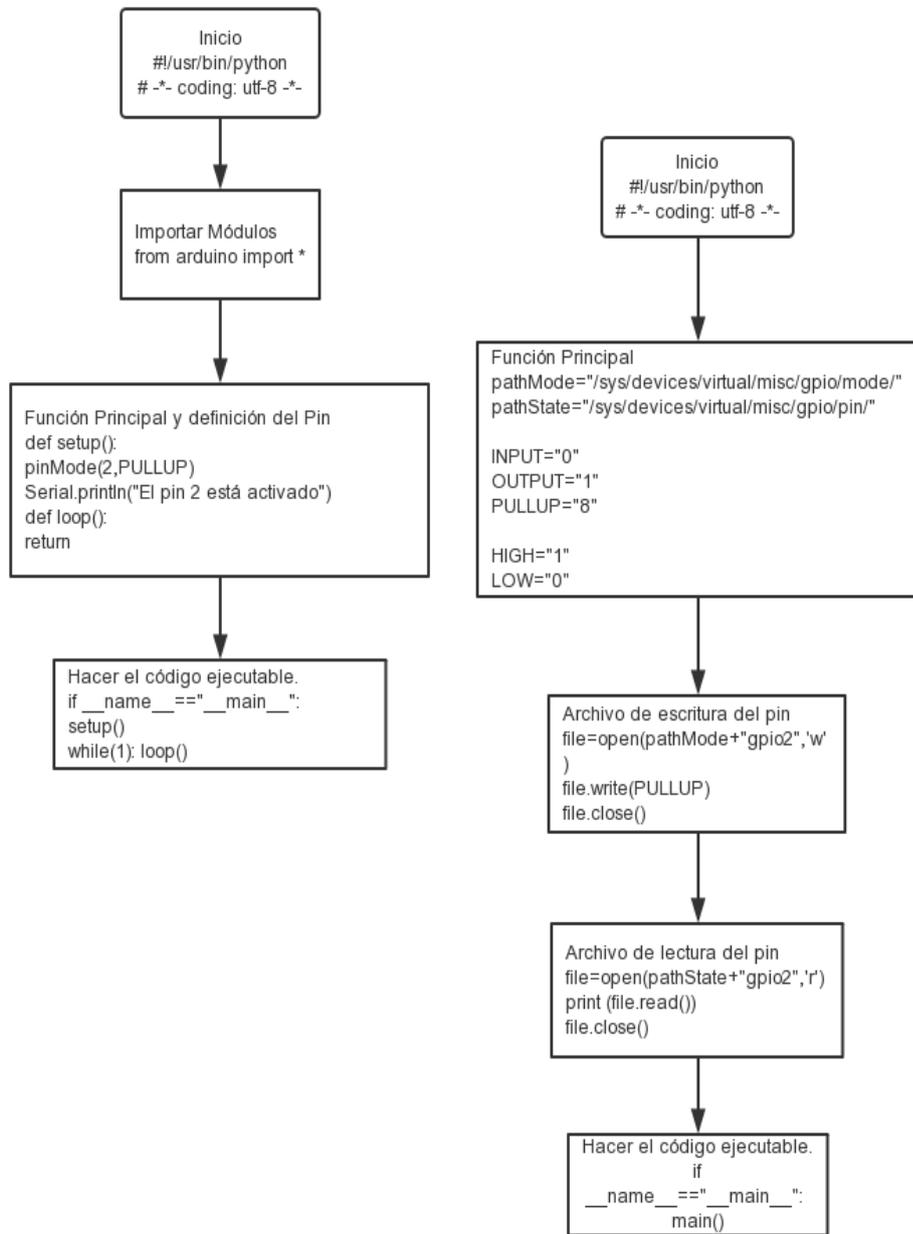
Se debe comenzar definiendo la expresión útil en una variable con el texto:

```
pathMode= "/sys/devices/virtual/misc/gpio/mode/"
```

Para fijar el significado de un pin de E/S, se lo realiza de manera muy sencilla: sólo se tiene que abrir el archivo y escribir el valor deseado:

- ✓ '0' para asignar como entrada al pin: INPUT= "0"
- ✓ '1' para asignar como salida al pin: OUTPUT= "1"
- ✓ '8' para asignar al pin de entrada con PULLUP= "8" para pulsadores, pull-up está disponible en todos los pines inmediatamente.

En los diagramas esquemáticos del acceso a los ficheros de los pines de E/S se da a conocer de forma detallada como se debe acceder a dichos ficheros o si bien si se utiliza la forma simplificada el procedimiento de cómo asignar los valores deseados a los pines de E/S.



*Figura 50 Versión simplificada y versión cruda de accesos a los valores de los pines de E/S.*

Fuente: El Autor.

### 3.4 SISTEMA DE ADQUISICIÓN DE DATOS DE LOS SENSORES DE MAGNITUDES FÍSICAS: TEMPERATURA, DISTANCIA, POSICIÓN, VELOCIDAD ANGULAR.

Los sistemas de adquisición de datos recogen datos de sensores para su posterior procesamiento y análisis. Este sistema se maneja en situaciones en las que los sensores han levantado grandes conjuntos de datos del entorno del sistema y no es preciso procesar los datos recopilados en tiempo real. En los sistemas de adquisición de datos, los sensores pueden estar generando datos velozmente, y el inconveniente principal es aseverar que una lectura del sensor es acumulada antes de que cambie el valor del sensor. Todo esto proporciona una arquitectura genérica tal y como se muestra en la figura siguiente, la característica esencial de la arquitectura de los sistemas de adquisición de datos es que cada grupo de sensores tiene tres métodos incorporados: el proceso del sensor que interactúa con el sensor y convierte valores analógicos a valores digitales si es necesario, un proceso búfer que almacena los datos, y un proceso que consume los datos y realiza un procesamiento adicional. En general los sistemas de adquisición de datos contienen métodos de visualización e informes que congregan los datos acumulados y realizan un procesamiento adicional.

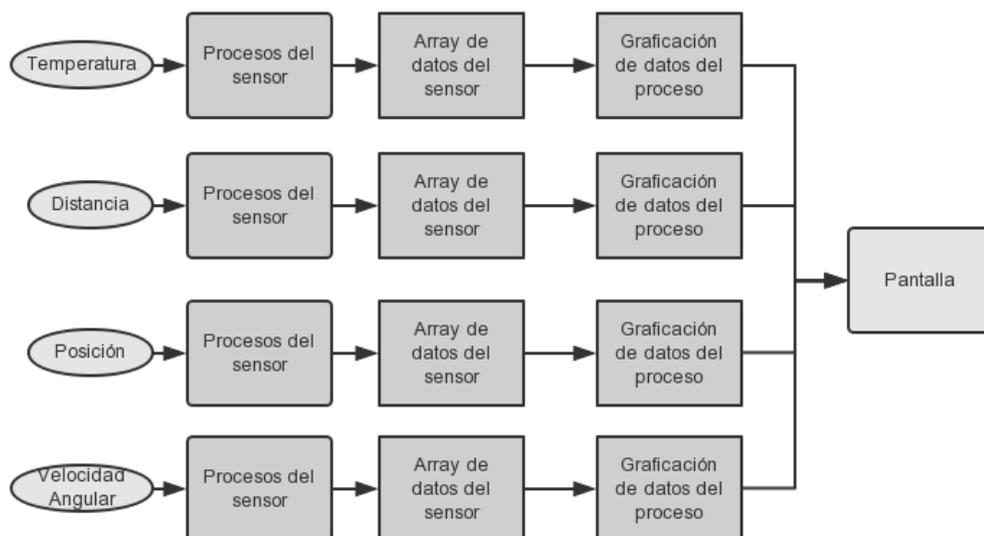


Figura 51 Sistema de adquisición de datos de sensores.

Fuente: El Autor.

Cada sensor posee un proceso incorporado que convierte la señal entrada analógica en una señal digital. Este proceso envía dicho flujo de datos, con el identificador del sensor, al búfer de datos del sensor. El proceso responsable del procesamiento de los datos toma los datos de este búfer, los procesa y los envía a un proceso de visualización para mostrarlos en una consola o pantalla del operador.

En sistemas de tiempo real que involucran la adquisición y el procesamiento de datos, las velocidades y periodos del proceso de adquisición y el proceso de procesamiento pueden no estar sincronizados cuando se requieren un procesamiento de los datos. Si simplemente es preciso realizar cálculos sencillos, el procesamiento puede ser más rápido que la adquisición de los datos. Para suavizar estas diferencias de velocidad, los sistemas de adquisición de datos almacenan los datos de entrada utilizando un bufer circular.

### **3.4.1 ADQUISICIÓN DE LOS VALORES DE LOS SENSORES ANALÓGICOS**

Los sensores de magnitudes físicas tales como la temperatura, posición, distancia y velocidad angular van a tener una señal de salida analógica; la obtención de los valores de dichas señales será posible mediante la utilización de las entradas analógicas del pduino. Como ya se ha evidenciado la obtención de los valores se las va a realizar por medio del manejo de la sintaxis de instrucciones tanto de Pyduino como la sintaxis propia de Python.

Se puede observar el diagrama esquemático de la conexión de los sensores analógicos, ciertamente es una abstracción simplificada de la conexión; más adelante se detallaran cada una de las conexiones y funcionalidades de los sensores analógicos.

Diagrama esquemático de los sensores analógicos.

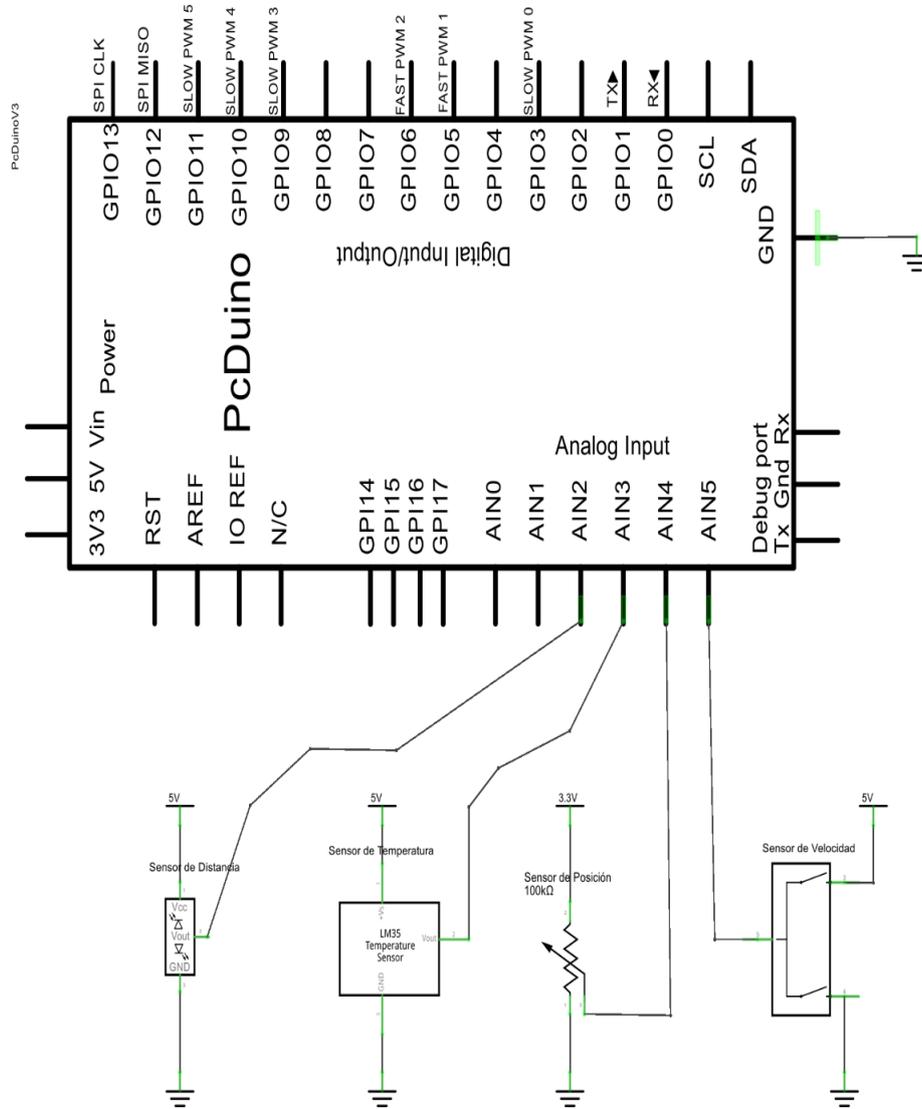


Figura 52 Diagrama esquemático de los sensores analógicos.

Fuente: El Autor.

### 3.4.2 DIAGRAMA DE FLUJO DE LA REALIZACIÓN DE LAS GRÁFICAS DE LOS VALORES OBTENIDOS DE LAS ENTRADAS ANALÓGICAS.

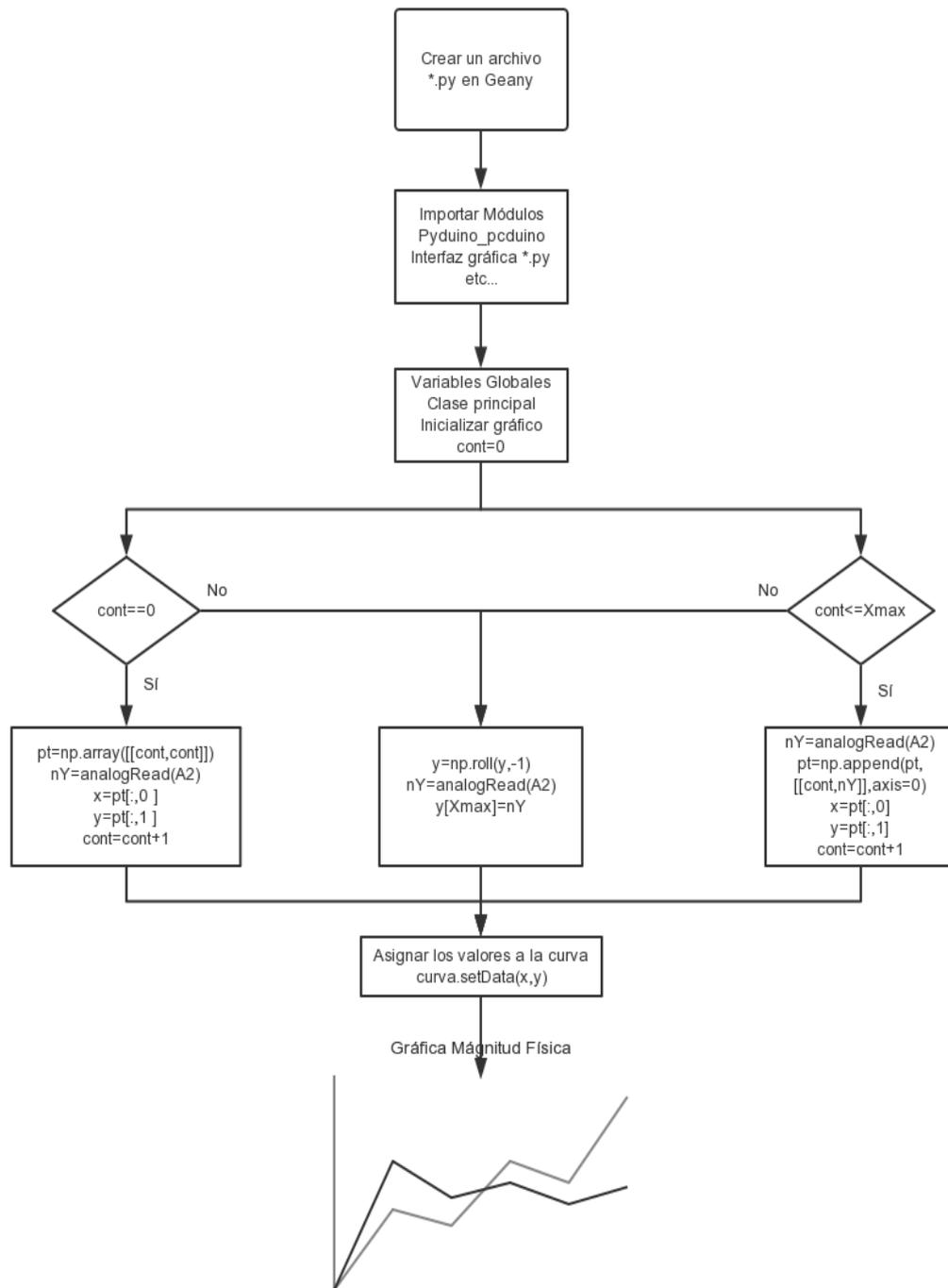


Figura 53 Diagrama de flujo para graficación de las entradas analógicas.

Fuente: El Autor.

### **3.5 PROCEDIMIENTOS A SEGUIR PARA CREAR LOS PROGRAMAS DESTINADOS AL ANÁLISIS DE LAS MAGNITUDES FÍSICAS: TEMPERATURA, DISTANCIA, POSICIÓN, VELOCIDAD ANGULAR.**

En este apartado se detalla los procedimientos a seguir para obtener un programa general del análisis de las magnitudes físicas: temperatura, distancia, posición, velocidad angular.

#### **3.5.1 PROCEDIMIENTOS PARA LA CREACIÓN DE INTERFACES GRAFICAS DE USUARIOS PARA LAS MAGNITUDES FÍSICAS: TEMPERATURA, DISTANCIA, POSICIÓN, VELOCIDAD ANGULAR.**

El proceso frecuente de análisis y diseño de la interfaz de usuario se aborda con la creación de varios modelos del funcionamiento en forma general del módulo didáctico. Se inicia proyectando las tareas encaminadas al usuario y al módulo en su totalidad es decir los lineamientos que se requieren a fin de obtener el funcionamiento del módulo, para prontamente considerar los aspectos que se aplican a todos los diseños de los interfaces de cada una de magnitudes físicas que se consideran en este módulo didáctico. Se debe explotar las herramientas de Qt-designer para hacer prototipos e implementar el modelo del diseño final de las interfaces de usuarios.

La interfaz se centra en el perfil de los usuarios que interactuarán con el módulo didáctico. En esencia, se vela para hacer entender la percepción del módulo didáctico a los usuarios. Una vez definidos los requerimientos generales, se lleva a cabo un detallado análisis de la magnitud física. Asimismo, se identifican, describen y elaboran prácticas de laboratorio que el estudiante va a realizar para alcanzar los fines del módulo didáctico de hardware libre para análisis y verificación de sistemas análogos digitales. En el capítulo 4 de prácticas, se detalla el análisis de cada una de las prácticas de laboratorio y la forma de desarrollarlos por medio de una interfaz gráfica de usuario.

La meta del diseño de la interfaz de usuario es precisar un conjunto de objetos, acciones y sus representaciones en la pantalla; que permitan al estudiante desarrollar todas las prácticas de laboratorio definidas en forma tal que se cumpla cada meta de la usabilidad definida para el módulo didáctico.

Para realizar la interfaz gráfica de usuario básicamente se debe ser ordenado en la forma de representar los valores obtenidos de los sensores. Se presenta un diagrama de flujo que describe los pasos a seguir.

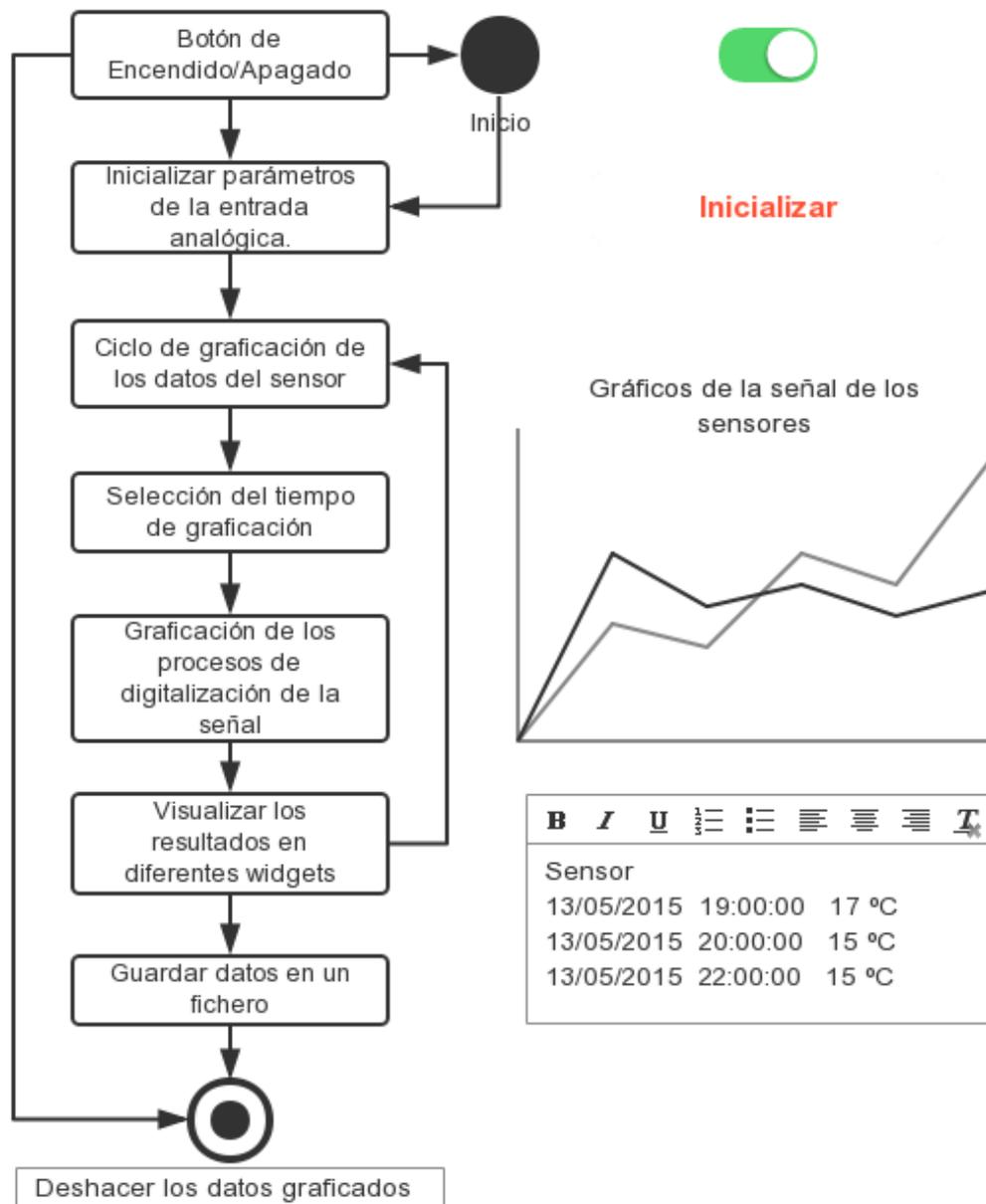


Figura 54 Flujograma de creación de las interfaces graficas de usuario.

Fuente: EL Autor.

### 3.5.1.1 Descripción general de las Interfaces gráficas de usuario

Para realizar la interfaz gráfica de usuario destinada al análisis de cada una de las magnitudes, procedemos a abrir Qt-designer y creamos una forma es decir un Widget; en general los campos principales que van tener dichas interfaces gráficas serán: Los datos informativos del proyecto; un PlotWidget para graficar de los valores obtenidos de las entradas analógicas A[2, 3, 4, 5]; un qwtplot para representar los tres pasos para la conversión análoga digital; varios QlcdNumber para visualizar los valores de los diferentes procesos de la señal; un qwtWidget para representar gráficamente los valores de las magnitudes. En general, finalmente luego de varias pruebas de interfaces gráficas de usuario, se puede apreciar:

- Región de activación/desactivación de las lecturas de las entradas analógicas: contendrá un led indicador de encendido o apagado de los módulos de sensores y un botón que cumplirá la función de activar o desactivar el módulo de los sensores indistintamente de la magnitud a analizar.
- Datos generales del trabajo de grado y de la magnitud a analizar.
- Región de gráfica de los valores de los sensores (PlotWidget).
- Región de botones de inicio de la gráfica de los valores de los sensores: se podrá modificar el tiempo de graficación de la señal mediante un Dial que contendrá los valores desde un 1ms a 256 ms; se podrá borrar los datos graficados a través de un botón Borrar; se podrá parar la graficación de los datos mediante un botón Parar, que detendrá el timer de graficación.
- Región de gráfica de los procesos de conversión análoga digital de los sensores (Qwtplot).

- Región de botones de selección de gráfica de los procesos de conversión analógica digital de los sensores:
  - ✓ Contendrá un botón muestreo.- Este botón abarca el método de muestrear la señal proveniente de la entrada analógica del pduino, indistintamente de que magnitud se esté monitorizando o de la entrada analógica que este activa. El proceso de graficación de la señal muestreada se podrá observar en el qwtplot.
  - ✓ Contendrá un botón cuantificación.- Este botón abarca el método de cuantificación de los datos obtenidos del proceso de muestreo, indistintamente de que magnitud se esté monitorizando o de la entrada analógica que este activa. El proceso de graficación de la señal cuantificada se podrá observar en el qwtplot.
  - ✓ Contendrá un botón codificación.- Este botón abarca el método de codificar la señal cuantificada, de acuerdo a cada magnitud que se esté monitorizando y de la entrada analógica que este activa. El proceso de graficación de la señal muestreada se podrá observar tanto en el qwtplot como en la sección de valores digitales en la interfaz gráfica de usuario.
  
- Región de valores digitales: contendrá un LcdNumber con el valor de las magnitudes en binario; dos LineEdit en los cuales se podrá modificar los valores mínimos y máximos que se podrá mostrar en el LcdNumber.
  
- Región de visualización de datos de las magnitudes físicas: constara de un grupo de widgets en los cuales se podrá evidenciar los valores de las magnitudes físicas, dependiendo de la magnitud variara la distribución de los mismos.

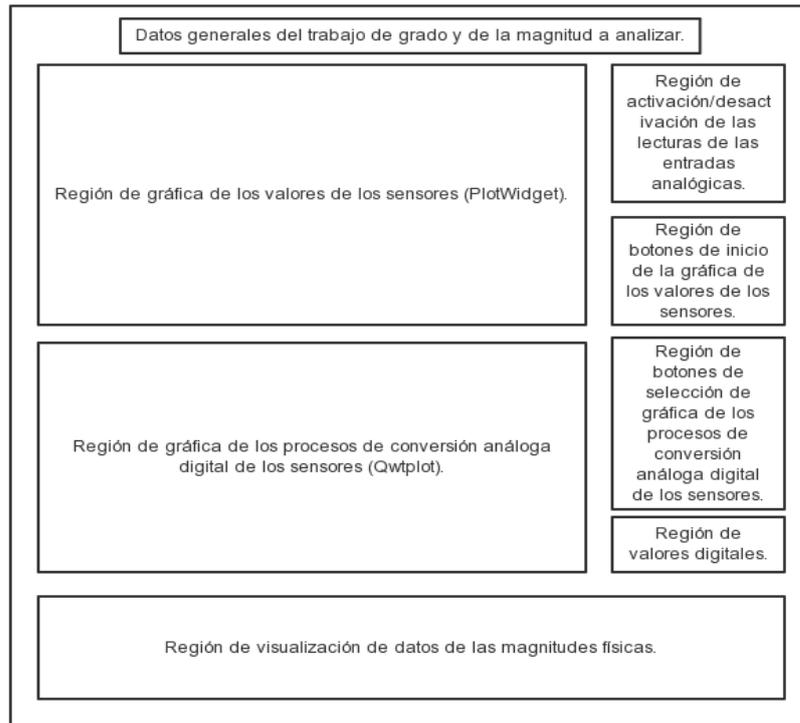


Figura 55 Diagrama generalizado de las interfaces gráficas de usuario.

Fuente: EL Autor.

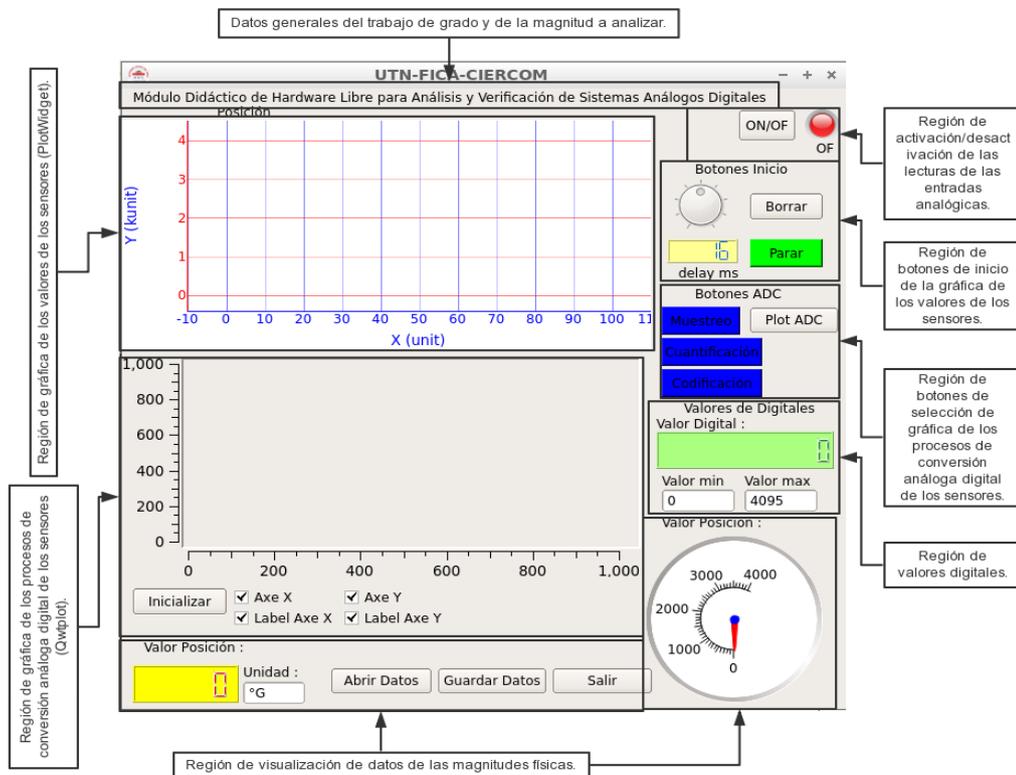


Figura 56 Interfaz gráfica de usuario de las magnitudes físicas (Posición).

Fuente: El Autor.

### 3.5.2 PROCESO DE ADQUISICIÓN DE LA SEÑAL PROVENIENTE DE LAS ENTRADAS ANALÓGICAS.

El proceso de adquisición de la señal proveniente de los sensores Im35, Sharp, potenciómetro de precisión, tacómetro; se lo realiza en las entradas analógicas desde A2 hasta A5 respectivamente, las mismas que serán asignadas a variables nYT, nYD, nYP, nYV ; con la expresión: nY[T, D, P, V]= analogRead(A[2, 3, 4, 5]), los valores obtenidos se deben ir almacenando en un búfer (array o vector) para poder realizar los siguientes procesos de la conversión análoga digital.

*Tabla 17 Variables de almacenamiento de los valores de las entradas analógicas.*

```
adc[T, D, P, V]=analogRead(A2-A5)
adc[T, D, P, V]= adc[T, D, P, V]*(3300.0)/4095
self.cont=0
```

Fuente: El Autor.

Creación del array de dos dimensiones, en referencia a la variable de conteo self.cont=0.

*Tabla 18 Array de dos dimensiones.*

```
self.points= np.array([[self.cont,adc[T, D, P, V]])
```

Fuente: El Autor.

Para que los valores se actualicen de forma automática se realiza una comparación entre la variable contador (cont), los tres posibles casos a comparar son:

*Tabla 19 Método para que los valores se actualicen de forma automática.*

```
if self.cont==0: # primera comparación
    self.points= np.array([[self.compt,adc[T, D, P, V]])
    self.x=self.points[:,0]
    self.y=self.points[:,1]
    self.cont=self.cont+1
elif self.cont<=self.nombreValores:
    nY[T, D, P, V]=adc[T, D, P, V]
    self.points=np.append(self.points,[[self.cont,nY[T, D, P,
V]]],axis=0)
    self.x=self.points[:,0]
```

```

self.y=self.points[:,1]
self.cont=self.cont+1
else
self.y=np.roll(self.y,-1)
self.y[self.nombreValores]=adc[T, D, P, V]

```

Fuente: El Autor.

### 3.5.2.1 El proceso de graficación de la señal proveniente de los sensores analógicos.

Se lo realiza utilizando un PlotWidget que hace que sea muy fácil el trazado y visualización de datos. En primer lugar debemos inicializar el grafico con los valores necesarios:

*Tabla 20 Código para Inicializar el grafico de la señal de las entradas analógicas.*

```

self.graph.setBackgroundBrush(QBrush(QColor(Qt.white)))
self.graph.showGrid(x=True, y=True)
self.graph.getAxis('boton').setPen(pg.mkPen(0,0,255))
self.graph.getAxis('left').setPen(pg.mkPen(255,0,0))
labelStyle = {'color': '#00F', 'font-size': '10pt'}
self.graph.getAxis('boton').setLabel('Medida', units='unit', **labelStyle)
self.graph.getAxis('left').setLabel('Valor', units='unit', **labelStyle)
self.graph.enableAutoRange(axis=pg.ViewBox.YAxis, enable=False)

```

Fuente: El Autor.

Asignamos los valores predeterminados para la visualización en los ejes x, y:

*Tabla 21 Código para asignar valores predeterminados para la visualización en los ejes x, y.*

```

self.minY=0
self.maxY=3300
self.graph.setYRange(self.minY,self.maxY)
self.minX=0
self.maxX=self.nombreValores
self.graph.setXRange(self.minX,self.maxX)

```

Fuente: El Autor.

Por último se inicializa la curva con los valores asignados a x,y:

Tabla 22 Código para inicializar la graficación de los valores.

---

```
self.curva.setData(self.x,self.y) # inicialización de los valores de la curva.
```

---

Fuente: El Autor.

### 3.5.3 MUESTREO DE LA SEÑAL PROVENIENTE DE LAS ENTRADAS ANALÓGICAS (A [2, 3, 4, 5]).

Como se ha descrito en apartados anteriores el proceso de muestrear una señal, radica en medir periódicamente el valor de la misma y establecer un conjunto de valores. Este conjunto de valores en nuestro caso es de 0 a 4095, debido a que tenemos entradas analógicas de 12 bits.

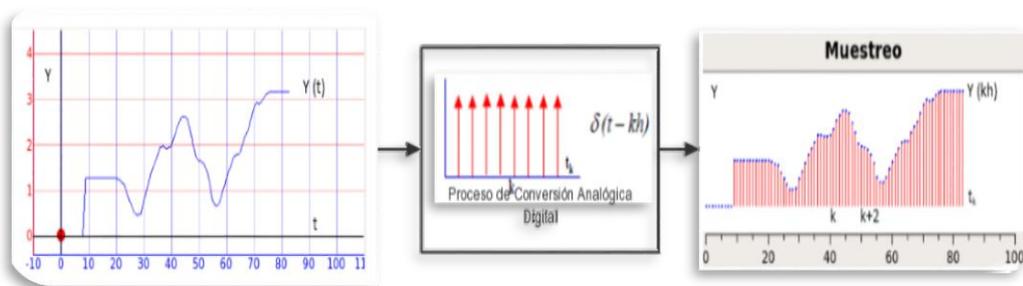


Figura 57 Formación de la señal muestreada a partir de la señal de los sensores.

Fuente: El Autor.

A partir del punto de vista matemático, muestrear una señal analógica se obtiene multiplicando la señal analógica por la función delta de Dirac. La función delta de Dirac está dada por:

$$\delta(t - kh) \dots (6)$$

Ecuación 6 Función delta de Dirac.

Fuente: (Neamen, 2010).

Donde:

Sólo tiene valores en los instantes en que t es igual a kh, donde k es un entero 0, 1, 2, etc., que representa el instante de muestreo. Dato que para los

demás valores la función vale cero, la señal resultante sólo presentará valores en los instantes de muestreo.

En un instante de muestreo cualquiera,  $k$ , el tiempo transcurrido,  $t_k$ , será:

$$t_k = k * h \dots (7)$$

Ecuación 7 Primer instante de muestreo.

Fuente: (Neamen, 2010).

El valor de variable será:  $y_k$

En el siguiente instante los valores serán:

$$t_{k+1} = (k+1) * h \quad y \quad y_{k+1} \dots (8)$$

Ecuación 8 Instantes de muestreo  $k+1$ .

Fuente: (Neamen, 2010).

Fuera de los momentos de muestreo, el valor de la señal de salida del muestreador es 0 y en el instante  $k$  el valor será:

$$y(kh) = y(t_k) * \delta(t - kh) \dots (9)$$

Ecuación 9 Función de salida del muestreador.

Fuente: (Neamen, 2010).

Donde:

$y(kh)$  → Señal muestreada. Valores únicamente para  $k = 0, 1, 2, \dots$

$y(t_k)$  → Valor de la señal continua en el instante de muestreo  $k$ , es decir, cuando ha transcurrido un tiempo igual a ecuación (2) ( $t_k = k * h$ ).

La secuencia de salida del muestreador puede expresarse por la siguiente ecuación en el dominio del tiempo.

$$y(t_0) * \delta(t) + y(t_1) * \delta(t - h) + y(t_2) * \delta(t - 2h) + y(t_k) * \delta(t - kh) + \dots$$

$$= \sum_{k=0}^{\infty} y(t_k) * \delta(t - kh) \quad \dots (10)$$

Ecuación 10 Sumatoria de secuencia de salida del muestreador.

Fuente: (Neamen, 2010).

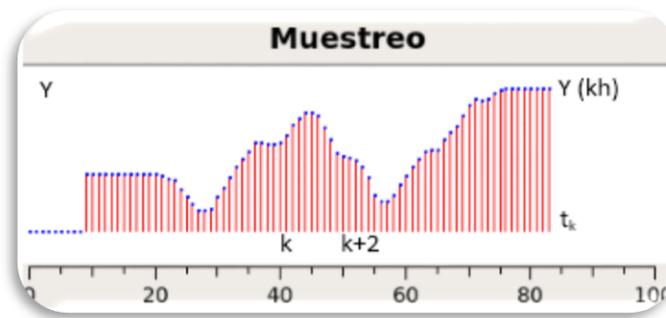


Figura 58 Señal muestreada.

Fuente: El Autor.

Tal como se ha mencionado, la señal de salida, es una señal que sólo tiene valores en los momentos de muestreo y su valor es igual al de la señal de entrada al muestreador para el tiempo  $t_k$ .

### 3.5.3.1 Visualización del muestreo de las señales provenientes de las entradas analógicas.

Este apartado de la visualización del muestreo de la señal se hace necesaria la utilización de Qwt o Qt Widgets for Technical Applications que es un conjunto de widgets para Qt, componentes GUI y clases técnicas que se utilizan principalmente en programas para mostrar gráficos.

La representación gráfica del muestreo de una señal se la puede realizar a través de la función Stem, la cual nos permite realizar representaciones discretas sin mayor complicación; es decir nos grafica líneas rectas que parten desde la abscisa 't' hasta el valor que le corresponde en el eje 'y' el mismo que representa el valor de los datos obtenidos en la entrada analógica. En Qwt existe

un variante a este comando denominada sticks que realiza la misma función en un qwtplot.

En definitiva se utiliza qwtplot y la variante sticks para trazar la secuencia de datos Y como impulsos, los mismos que se extienden a partir de valores igualmente espaciados y generadas automáticamente a lo largo del eje x, del array creado anteriormente.

Tabla 23 Código para graficar el muestreo.

```
def plotMuestreo():  
    self.curvaR.setData(self.x, self.y)  
    self.qwtPlot.replot()  
    self.qwtPlot.setCanvasBackground(Qt.Qt.white)  
    self.qwtPlot.setTitle("Muestreo")  
    self.qwtPlot.insertLegend(Qwt.QwtLegend(), Qwt.QwtPlot.BottomLegend);  
    self.curvaR = Qwt.QwtPlotCurve("Muestreo")  
    self.curvaR.attach(self.qwtPlot)  
    self.curvaR.setSymbol(Qwt.QwtSymbol(Qwt.QwtSymbol.Ellipse,  
                                       Qt.QBrush(),  
                                       Qt.QPen(Qt.Qt.blue),  
                                       Qt.QSize(2, 2)))  
  
    self.curvaR.setStyle(Qwt.QwtPlotCurve.Sticks)  
    self.curvaR.setPen(Qt.QPen(Qt.Qt.red))  
    self.qwtPlot.connect(self.timer, SIGNAL('timeout()'), plotMuestreo)
```

Fuente: El Autor.

#### **3.5.4 CUANTIFICACIÓN DE LAS SEÑALES PROVENIENTES DE LAS ENTRADAS ANALÓGICAS.**

Consecutivamente es necesario conseguir la señal continua será necesario reconstruirla, este proceso se conoce con nombre de cuantificación o mantenimiento de la señal. Se puede realizar matemáticamente mediante retención o mantenedores. El mantenedor más simple es un mantenedor de orden cero, que consiste en mantener el valor de la variable hasta que llegue el siguiente valor.

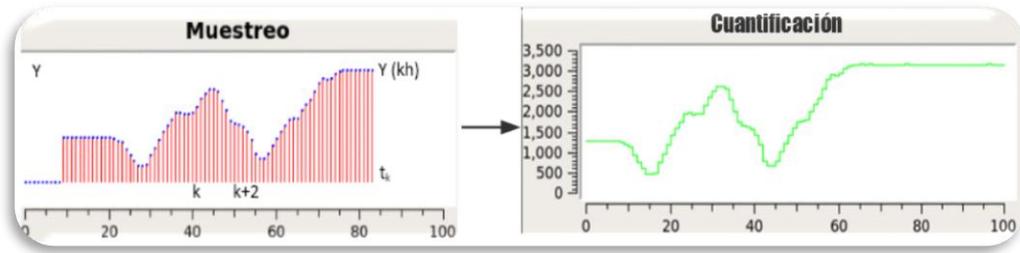


Figura 59 Proceso de cuantificación.

Fuente: El Autor.

Señal antes del mantenedor de orden cero:

$$y(0h) * \delta(t) + y(1h) * \delta(t - h) + y(2h) * \delta(t - 2h) + \dots + y(kh) * \delta(t - kh) + \dots \dots (11)$$

Ecuación 11 Señal antes del mantenedor de orden cero.

Fuente: (Neamen, 2010).

Señal después del mantenedor:

$$\text{Para } 0 \leq t < h \rightarrow y(t) = y(0 * h)$$

$$\text{Para } h \leq t < 2h \rightarrow y(t) = y(1 * h)$$

$$\text{Para } 2h \leq t < 3h \rightarrow y(t) = y(2 * h)$$

$$\text{Para } kh \leq t < (k + 1)h \rightarrow y(t) = y(k * h) \dots (12)$$

Ecuación 12 Señales después del mantenedor de orden cero.

Fuente: (Neamen, 2010).

### 3.5.4.1 Visualización de la cuantificación de las señales provenientes de las entradas analógicas.

Del mismo modo que en el muestreo se utiliza un qwtPlot, además utilizamos la función escalón (step). Dicha función step hace una graficación de los valores mediante la utilización de líneas hasta que se presente el siguiente

valor a graficar, es decir, la entrada va a hacer graficada hasta que el valor de la entrada cambie y se grafique los nuevos valores nuevamente.

*Tabla 24 Código para graficar el cuantificación.*

```
def plotCuantificación():  
    self.curvaL.setData(self.x, self.y)  
    self.qwtPlot.replot()  
    self.qwtPlot.setCanvasBackground(Qt.Qt.white)  
    self.qwtPlot.setTitle("Cuantificación ")  
    self.qwtPlot.insertLegend(Qwt.QwtLegend(), Qwt.QwtPlot.BottomLegend);  
    self.curvaL = Qwt.QwtPlotCurve("Cuantificación")  
    self.curvaL.attach(self.qwtPlot)  
    self.curvaL.setStyle(Qwt.QwtPlotCurve.Steps)  
    self.curvaL.setPen(Qt.QPen(Qt.Qt.green))  
    self.qwtPlot.connect(self.timer, SIGNAL('timeout()'), plotCuantificación)  
    picker = Qwt.QwtPlotPicker(Qwt.QwtPlot.xBottom,  
        Qwt.QwtPlot.yLeft,  
        Qwt.QwtPicker.PointSelection,  
        Qwt.QwtPlotPicker.CrossRubberBand,  
        Qwt.QwtPicker.ActiveOnly,  
        self.qwtPlot.canvas())  
    picker.setTrackerPen(Qt.QPen(Qt.Qt.black)) # color de texto  
    picker.setRubberBandPen(Qt.QPen(Qt.Qt.magenta))
```

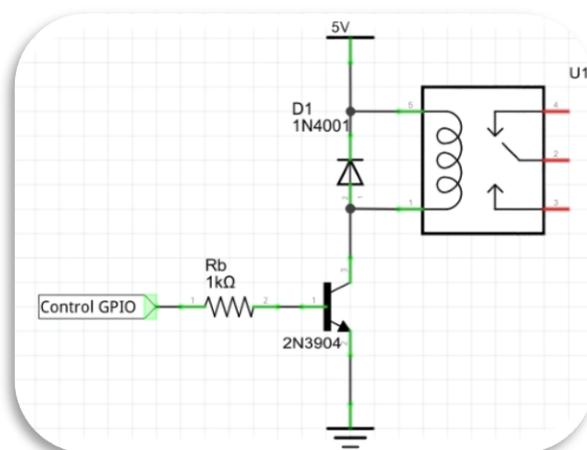
Fuente: El Autor.

### **3.5.5 CODIFICACIÓN DE LAS SEÑALES PROVENIENTES DE LOS SENSORES ANALÓGICOS.**

La codificación de los valores obtenidos en las entradas analógicas (A [2, 3, 4, 5]), se lo puede realizar al expresar los valores en forma binaria en un QlcdNumber o realizando una aproximación de los valores obtenidos en voltios o milivoltios de la misma entrada creando un ciclo para estabilizar y realizar una comparación del comportamiento lineal o exponencial de los sensores en función de las variaciones de las magnitudes medidas, para luego expresarles en un QlcdNumber y también en un indicador gráfico por ejemplo un QwtThermo. En los apartados subsecuentes se podrá evidenciar de forma específica la codificación de las señales de los sensores.

### 3.5.6 CIRCUITO DE ACTIVACIÓN DESACTIVACIÓN DE LOS MÓDULOS DE SENSORES.

En este apartado se detalla como acoplar un relé a un circuito electrónico digital a través de un transistor, el mismo que nos será de gran utilidad ya que activara o desactivara cada uno de los módulos de sensores.



*Figura 60 Circuito de activación desactivación de los módulos de sensores.*

Fuente: El Autor.

La manera más sencilla para activar un relé con un circuito electrónico de control es mediante un transistor NPN acoplado (la base a una resistencia, el emisor a tierra y el colector a la bobina del relé), el mismo que cierra el circuito colocando a tierra (gnd) el terminal de la bobina mientras que el otro terminal se encuentra conectado a positivo.

La singularidad de conexión de unión a tierra posee la ventaja de consentir el uso de relés con tensiones de alimentación de la bobina diferentes respecto a la tensión de trabajo del circuito de control, ya que el circuito de control es una GPIO del Pc-Duino configurada como salida y que su activación (3,3v) enciende el circuito de los módulos de sensores también su desactivación (0v) apaga los módulos de sensores.

Dicho de otra manera el circuito trabaja de esta forma:

- Cuando la salida de la GPIO es baja (0V) la base del transistor también lo será y consiguientemente este no dejará pasar corriente entre emisor y colector para activar la bobina del relé.
- Cuando la salida del GPIO es alta (3,3V), se supera la tensión de umbral de la base del transistor (0,6V) y consecuentemente, promueve una circulación de corriente entre base y tierra. Esta corriente produce que el transistor ingrese en estado de conducción entre colector y emisor, cerrando el circuito de la bobina del relé, por lo tanto activándolo.

Al desactivar el relé a través del transistor, por un instante, se produce un pico de tensión elevada de polaridad opuesta en sus terminales. Para solucionar este inconveniente, lo más simple es conectar en paralelo con la bobina un diodo rectificador inversamente polarizado de manera tal que este absorba estos picos de tensión de polaridad opuesta. Se hace uso de un diodo rectificador común el 1N4007.

Los leds indicadores, se los adhiere porque nos permiten confirmar el funcionamiento de los módulos de sensores.

### **3.6 CREACIÓN DEL PROGRAMA DESTINADO AL ANÁLISIS DE LA MAGNITUD TEMPERATURA.**

El proceso de adquisición de la señal proveniente del sensor Im35 se lo realiza en la entrada analógica A5, la misma que será asignada a una variable, los valores obtenidos se deben ir almacenando en un array (vector) para poder realizar los siguientes procesos de la conversión análoga digital.

#### **3.6.1 INTERFAZ GRÁFICA DE USUARIO**

Para realizar la interfaz gráfica de usuario destinada al análisis de la magnitud temperatura, procedemos a abrir Qt-designer y creamos una forma es

decir un Witget; arrastramos varios texEdit, los editamos con los datos informativos del proyecto; un PlotWidget para graficar de los valores obtenidos de la entrada analógica A5; un qwtplot para representar los tres pasos para la conversión análoga digital; varios QlcdNumber para visualizar los valores de los diferentes procesos de la señal; un qwtTherno para representar la temperatura gráficamente. La forma final de interfaz gráfica de usuario se puede apreciar:

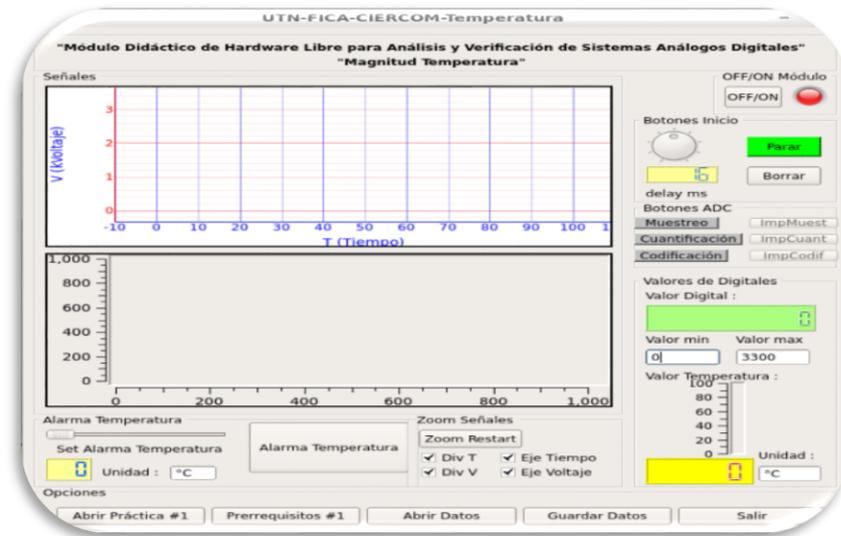


Figura 61 Interfaz gráfica de usuario de la magnitud temperatura.

Fuente: El Autor.

### 3.6.2 DIAGRAMAS DE CONEXIÓN DEL SENSOR DE TEMPERATURA LM35

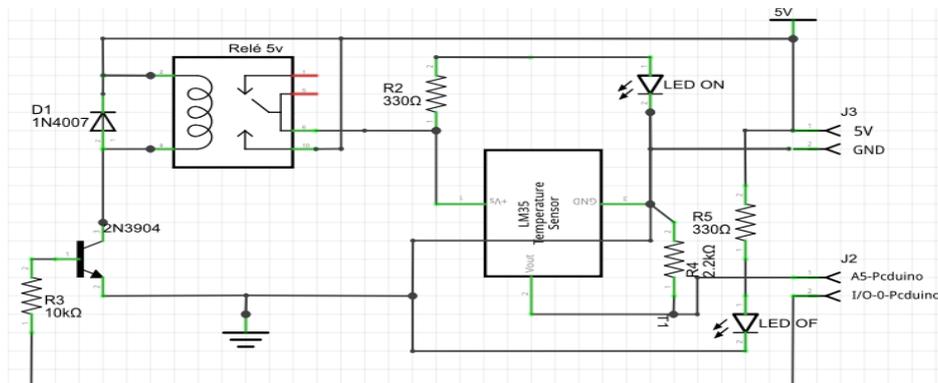


Figura 62 Conexión del sensor de temperatura Lm35 en la entrada analógica A5 del pcdiuno.

Fuente: El Autor.



Figura 63 Diagrama pictórico del sensor de temperatura.

Fuente: El Autor.

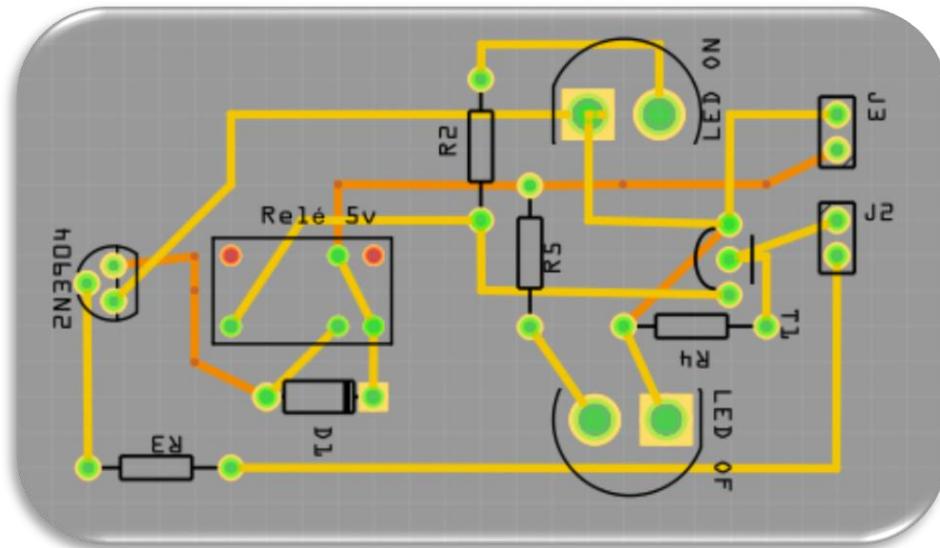


Figura 64 Diagrama del circuito impreso del sensor de temperatura.

Fuente: El Autor.

### **3.6.3 DESCRIPCIÓN DE FUNCIONAMIENTO DEL MÓDULO DE TEMPERATURA.**

Se inicia por la activación del módulo a través de la GPIO-0 del Pc-Duino, la misma que al estar en 3,3v activa la bobina del relé permitiendo que el contacto normalmente abierto del mismo se accione, de esta manera se energizara el sensor Lm35 y se encenderá el led indicador verde. El sensor de temperatura Lm35 solamente necesita ser energizado para funcionar y entregar en su pin Out el voltaje en mili voltio proporcional a la temperatura del medio.

Haciendo un breve análisis de consumo de corriente del circuito del módulo de temperatura, se tiene que el consumo de corriente de cada uno de los elementos inmiscuidos en el mismo es: Lm35 consume un promedio de 0,05mA, relé consume un promedio de 50mA, transistor consume un promedio 0,5mA, leds consumen un promedio 10mA. En general el consumo total del módulo de temperatura esta alrededor de 70mA, lo cual es un consumo de corriente pequeño.

### **3.6.4 PROCESO DE ADQUISICIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR LM35**

Como se ha mencionado anteriormente el procedimiento de adquisición de la señal procedente del sensor lm35 se lo realiza en la entrada analógica A5, la misma que será asignada a una variable adcT; en general se utiliza el proceso de adquisición de la señal proveniente de las entradas analógicas para este fin.

### **3.6.5 MUESTREO DE LA SEÑAL PROVENIENTE DEL SENSOR DE TEMPERATURA.**

En este apartado de la visualización del muestreo de la señal se hace necesaria la utilización de Qwt o Qt Widgets for Technical Applications que es un

conjunto de widgets para Qt, componentes GUI y clases técnicas que se utilizan principalmente en programas para mostrar gráficos.

La representación gráfica del muestreo de una señal se la puede realizar a través de la función Stem, la cual nos permite realizar representaciones discretas sin mayor complicación; es decir nos grafica líneas rectas que parten desde la abscisa 't' hasta el valor que le corresponde en el eje 'y' el mismo que representa el valor de los datos obtenidos en la entrada analógica. En Qwt existe un variante a este comando denominada sticks que realiza la misma función en un qwtplot.

En definitiva se utiliza qwtplot y la variante sticks para trazar la secuencia de datos Y como impulsos, los mismos que se extienden a partir de valores igualmente espaciados y generadas automáticamente a lo largo del eje x, del array creado anteriormente.

*Tabla 25 Graficación del muestreo de la señal proveniente del sensor de temperatura.*

```
def plotMuestreo():  
    self.curvaR.setData(self.x, self.y)  
    self.qwtPlot.replot()  
    self.qwtPlot.setCanvasBackground(Qt.Qt.white)  
    self.qwtPlot.setTitle("Muestreo")  
    self.qwtPlot.insertLegend(Qwt.QwtLegend(), Qwt.QwtPlot.BottomLegend);  
    self.curvaR = Qwt.QwtPlotCurve("Muestreo")  
    self.curvaR.attach(self.qwtPlot)  
    self.curvaR.setSymbol(Qwt.QwtSymbol(Qwt.QwtSymbol.Ellipse,  
                                       Qt.QBrush(),  
                                       Qt.QPen(Qt.Qt.blue),  
                                       Qt.QSize(2, 2)))  
  
    self.curvaR.setStyle(Qwt.QwtPlotCurve.Sticks)  
    self.curvaR.setPen(Qt.QPen(Qt.Qt.red))  
    self.qwtPlot.connect(self.timer, SIGNAL('timeout()'), plotMuestreo)
```

Fuente: El Autor.

### 3.6.6 CUANTIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE TEMPERATURA.

Del mismo modo que en el muestreo se utiliza un `qwtPlot`, además utilizamos la función escalón (`step`). Dicha función `step` hace un escalón unitario  $1(t)$  es decir, la entrada va desde cero a uno en el tiempo  $t = 0$ .

*Tabla 26 Graficación de cuantificación de la señal proveniente del sensor de temperatura.*

```
def plotCuantificación():
    self.curvaL.setData(self.x, self.y)
    self.qwtPlot.replot()
self.qwtPlot.setCanvasBackground(Qt.Qt.white)
self.qwtPlot.setTitle("Cuantificación ")
self.qwtPlot.insertLegend(Qwt.QwtLegend(), Qwt.QwtPlot.BottomLegend);
self.curvaL = Qwt.QwtPlotCurve("Cuantificación")
self.curvaL.attach(self.qwtPlot)
self.curvaL.setStyle(Qwt.QwtPlotCurve.Steps)
self.curvaL.setPen(Qt.QPen(Qt.Qt.green))
self.qwtPlot.connect(self.timer, SIGNAL('timeout()'), plotCuantificación)
picker = Qwt.QwtPlotPicker(Qwt.QwtPlot.xBottom,
    Qwt.QwtPlot.yLeft,
    Qwt.QwtPicker.PointSelection,
    Qwt.QwtPlotPicker.CrossRubberBand,
    Qwt.QwtPicker.ActiveOnly,
    self.qwtPlot.canvas())
picker.setTrackerPen(Qt.QPen(Qt.Qt.black)) # color de texto
picker.setRubberBandPen(Qt.QPen(Qt.Qt.magenta))
```

Fuente: El Autor.

### 3.6.7 CODIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE TEMPERATURA.

La codificación de los valores obtenidos en la entrada analógica A5, se lo puede realizar de manera instantánea al expresar los valores en forma binaria en un `QlcdNumber` o realizando una aproximación de los valores obtenidos en milivoltios de la misma entrada creando un ciclo para estabilizar y realizar una comparación del comportamiento lineal del sensor `Im35` en función de los grados

centígrados medidos, para luego expresarle en un QlcdNumber y también en un QwtThermo.

Tabla 27 Codificación de la señal proveniente del sensor de temperatura.

```
lm=0
for i in range(20):
    lm=lm+analogReadmV(A5)
valorCalc=lm/20
self.Thermo.setValue(valorCalc/10.0)
```

Fuente: El Autor.

### 3.7 CREACIÓN DEL PROGRAMA DESTINADO AL ANÁLISIS DE LA MAGNITUD DISTANCIA.

En este apartado solamente nos vamos a enfocar en el desarrollo de la interfaz gráfica y la codificación de la magnitud distancia.

#### 3.7.1 INTERFAZ GRÁFICA DE USUARIO DE LA MAGNITUD DISTANCIA.

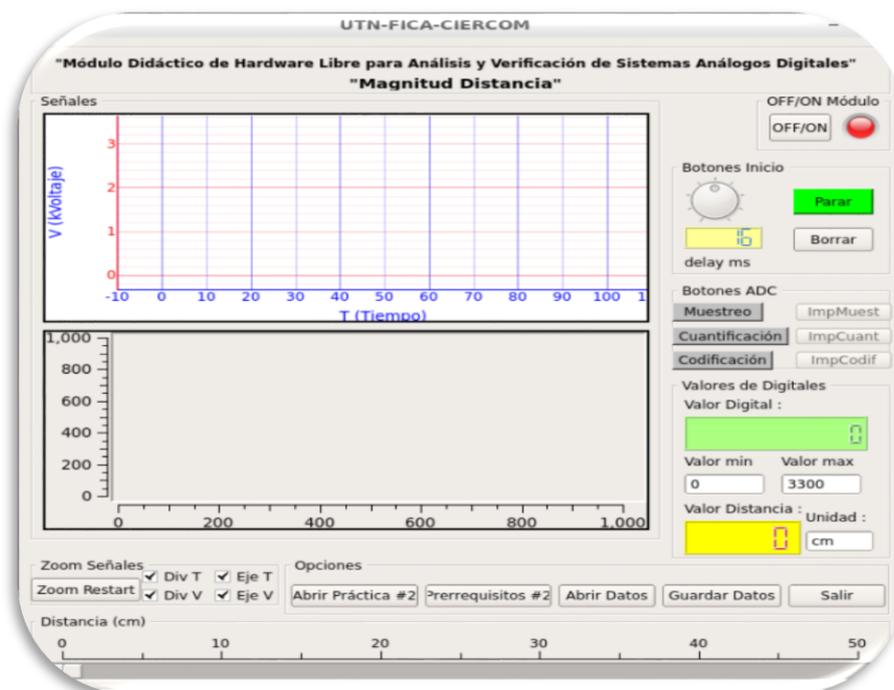


Figura 65 Interfaz gráfica de la magnitud distancia.

Fuente: El Autor.

### 3.7.2 DIAGRAMA DE CONEXIÓN DEL SENSOR DE DISTANCIA.

Se presenta el diagrama esquemático, pictórico y circuito impreso del sensor de distancia.

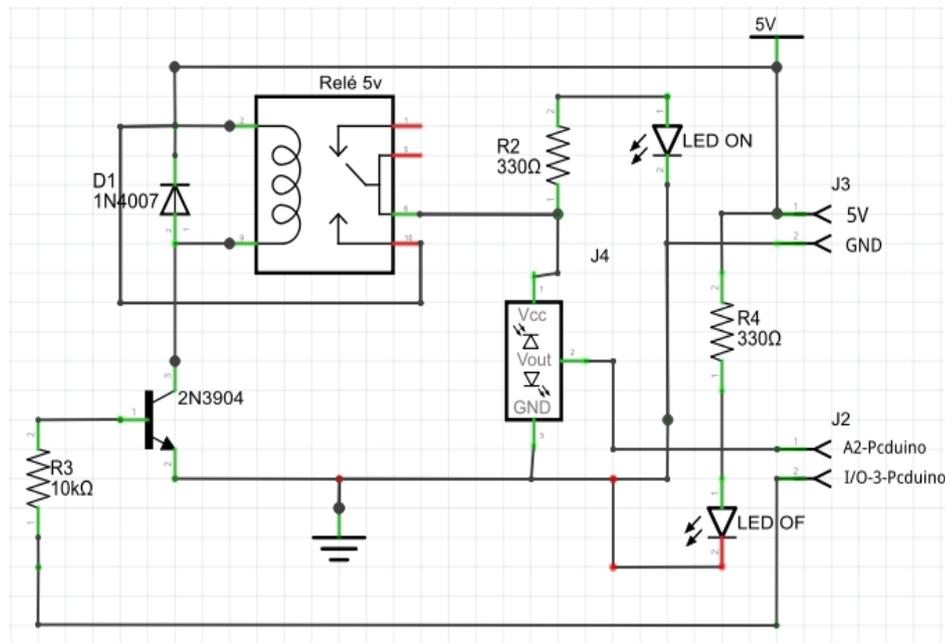


Figura 66 Diagrama esquemático del sensor de distancia.

Fuente: El Autor.

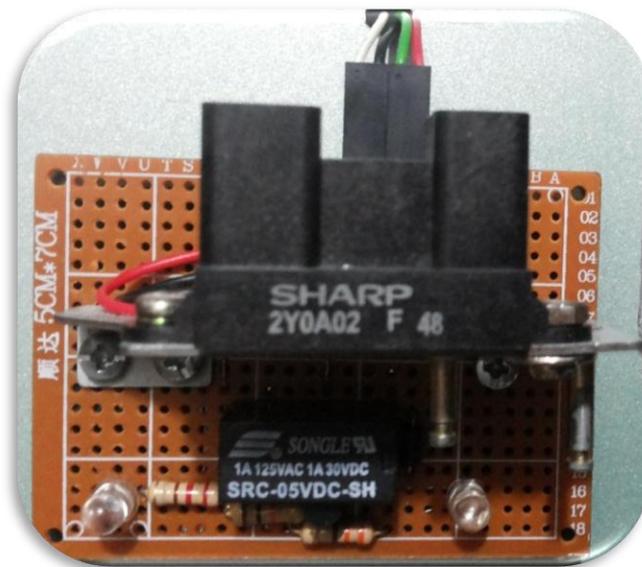


Figura 67 Diagrama pictórico del sensor de distancia.

Fuente: El Autor.

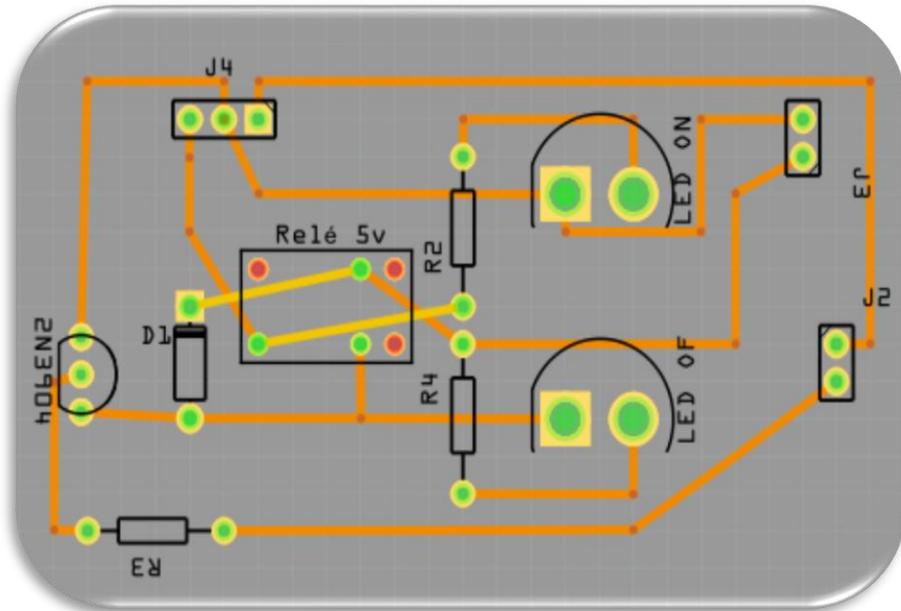


Figura 68 Diagrama del circuito impreso del sensor de distancia.

Fuente: El Autor.

### 3.7.3 DESCRIPCIÓN DE FUNCIONAMIENTO DEL MÓDULO DE DISTANCIA.

Se inicia por la activación del módulo a través de la GPIO-2 del Pc-Duino, la misma que al estar en 3,3v activa la bobina del relé permitiendo que el contacto normalmente abierto del mismo se accione, de esta manera se energizara el sensor Sharp gp2y0a02yk0f y se encenderá el led indicador verde. El sensor Sharp gp2y0a02yk0f por sus características de encapsulado, solamente necesita ser energizado para funcionar y entregar en su pin Out el voltaje proporcional a la distancia entre el obstáculo y el sensor.

Haciendo un breve análisis de consumo de corriente del circuito del módulo de distancia, se tiene que el consumo de corriente de cada uno de los elementos inmiscuidos en el mismo es: Sharp gp2y0a02yk0f consume un promedio de 33mA, relé consume un promedio de 50mA, transistor consume un promedio 0,5mA, leds consumen un promedio 10mA. En general el consumo total del módulo de temperatura esta alrededor de 100mA, lo cual es un consumo de corriente pequeño.

### 3.7.4 PROCESO DE ADQUISICIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR SHARP

Será el mismo método descrito en el Proceso de adquisición de la señal proveniente del sensor LM35, solamente cambiando la entrada analógica a A2, puesto que son señales de voltaje indistintamente de que magnitud se esté midiendo el proceso de adquisición de datos en las entradas analógicas van a utilizar el mismo procedimiento.

### 3.7.5 CODIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE DISTANCIA.

Para obtener el valor de la distancia se utiliza en primer instancia la medición de los valores de voltaje que el sensor Sharp proporciona por medio de una tabla de datos en donde se describe sus valores y continuación haciendo uso de la herramienta Excel transformamos los valores en una ecuación aproximada como se puede apreciar en las siguientes graficas:

*Tabla 28 Valores obtenidos del sensor del Sharp GP2Y02YK0F.*

<b>Distancia</b>	<b>Voltaje</b>
15	2,80
20	2,61
25	2,34
30	2,03
35	1,77
40	1,56
45	1,36
50	1,22
55	1,11
60	1,01
65	0,93
70	0,87
75	0,81
80	0,76
85	0,74
90	0,70
95	0,66
100	0,64

<b>105</b>	0,60
<b>110</b>	0,56
<b>115</b>	0,55
<b>120</b>	0,54
<b>125</b>	0,53
<b>130</b>	0,52
<b>140</b>	0,50
<b>150</b>	0,45

Fuente: El Autor.

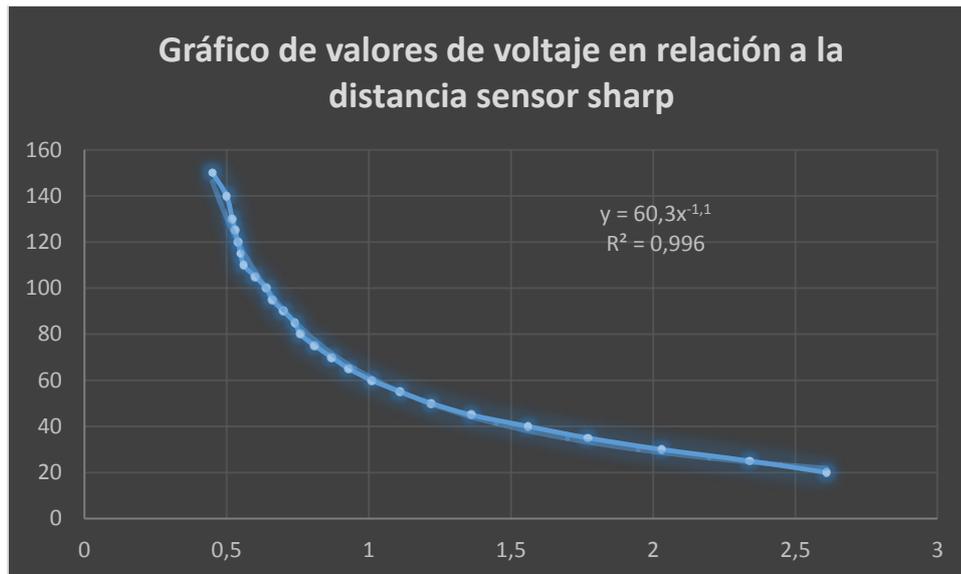


Figura 69 Valores de voltaje en relación a la distancia sensor Sharp.

Fuente: El Autor.

La función potencial,  $y = 60,3 x^{(-1,1)}$  en color azul en la gráfica es la que más se ajusta a nuestra curva de datos.

$$y = 60,3x^{-1,1} \quad \dots (13)$$

Ecuación 13 Función potencial del sensor Sharp.

### 3.8 CREACIÓN DEL PROGRAMA DESTINADO AL ANÁLISIS DE LA MAGNITUD POSICIÓN.

En cuanto a la magnitud posición se refiere el análisis se lo realiza midiendo la señal que entrega el sensor de posición, haciendo un reescalamiento entre el nivel de voltaje y posición del potenciómetro. Se hace

hincapié en la posición en la cual se encuentra el potenciómetro, realizando una aproximación simple, los procesos se podrán evidenciar en la interfaz gráfica de usuario correspondiente a la magnitud posición.

### 3.8.1 INTERFAZ GRÁFICA DE USUARIO DE LA MAGNITUD POSICIÓN.

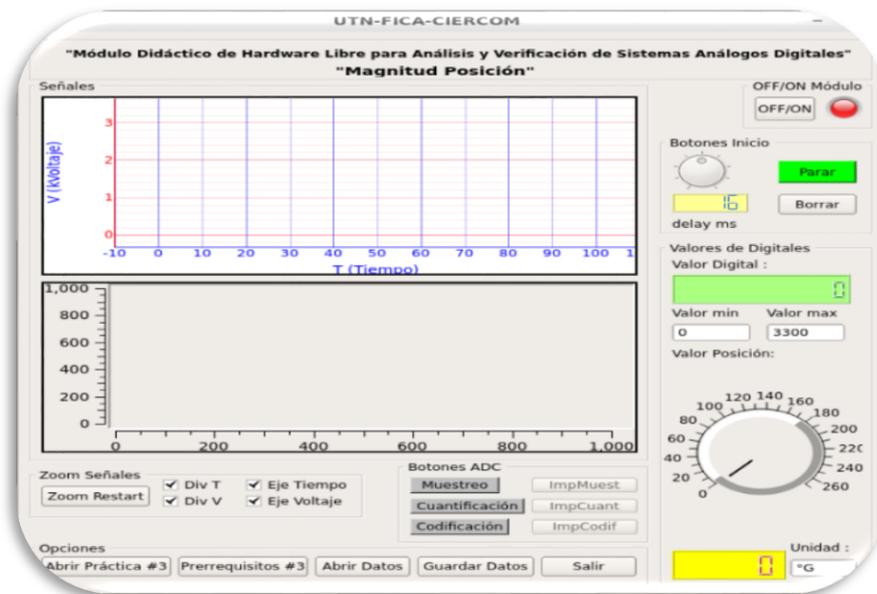


Figura 70 Interfaz gráfica de la magnitud posición.

Fuente: EL Autor.

### 3.8.2 DIAGRAMA DE CONEXIÓN DEL SENSOR DE POSICIÓN.

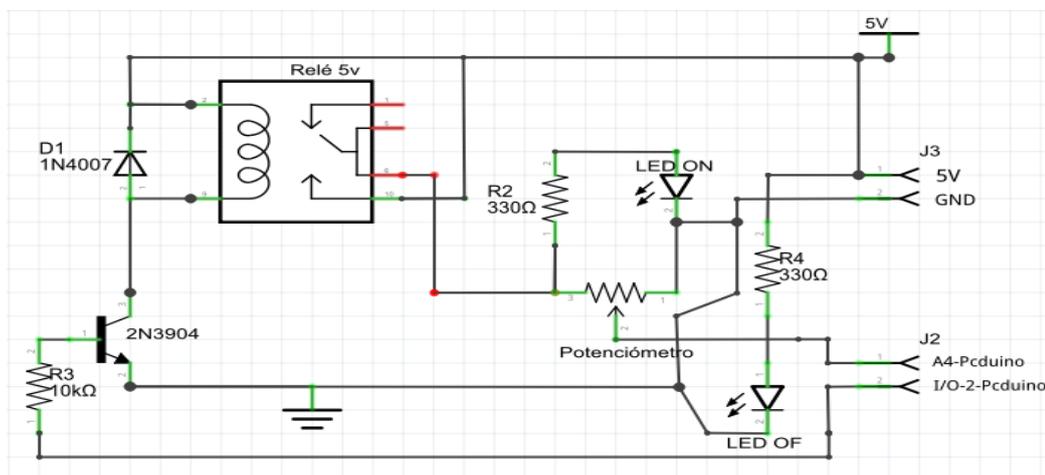


Figura 71 Diagrama esquemático del sensor de posición.

Fuente: EL Autor.



Figura 72 Diagrama pictórico del sensor de posición.

Fuente: EL Autor.

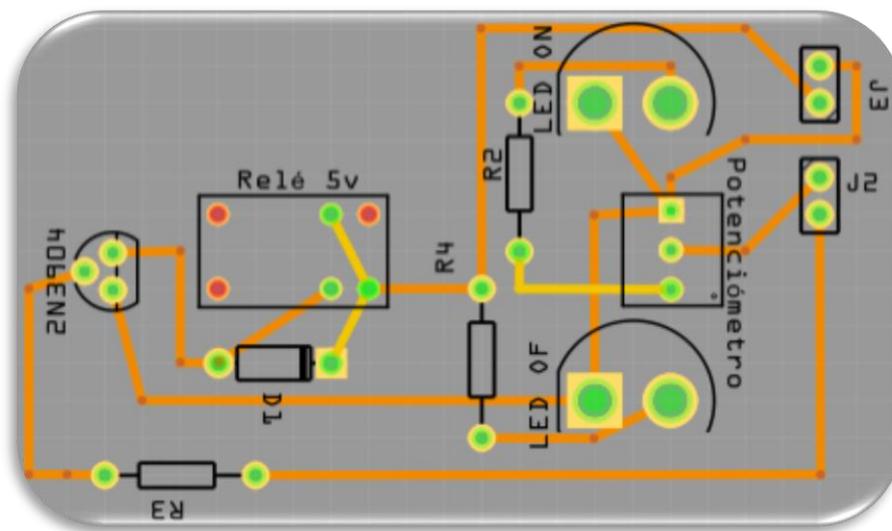


Figura 73 Diagrama circuito impreso del sensor de posición.

Fuente: EL Autor.

### 3.8.3 DESCRIPCIÓN DE FUNCIONAMIENTO DEL MÓDULO DE POSICIÓN.

Se inicia por la activación del módulo a través de la GPIO-3 del Pc-Duino, la misma que al estar en 3,3v activa la bobina del relé permitiendo que el contacto normalmente abierto del mismo se accione, de esta manera se energizara el potenciómetro y se encenderá el led indicador verde. El potenciómetro al ser energizado funciona como divisor de voltaje y entrega en su pin Out el voltaje proporcional a la posición de la perrilla del mismo.

Haciendo un breve análisis de consumo de corriente del circuito del módulo de posición, se tiene que el consumo de corriente de cada uno de los elementos inmiscuidos en el mismo es: potenciómetro consume un promedio de 3,3mA, relé consume un promedio de 50mA, transistor consume un promedio 0,5mA, leds consumen un promedio 10mA. En general el consumo total del módulo de temperatura esta alrededor de 70mA, lo cual es un consumo de corriente pequeño.

### 3.8.4 CODIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE POSICIÓN.

Se utiliza para este fin la función interpolación de python; la misma que realiza una operación para calcular los valores intermedios a lo largo de una curva, dada un conjunto de puntos de la curva.

*Tabla 29 Codificación de la señal proveniente del sensor de posición.*

---

```
self.valorCalc=np.interp(nYP,  
[self.Ymin,self.Ymax],[float(self.lineEditValorMin.text()),float(self.lineEditValor  
Max.text())])  
self.lcdNumberValorCalc.display(self.valorCalc)  
self.Dial.setValue(self.valorCalc)
```

---

Fuente: EL Autor.

### **3.9 CREACIÓN DEL PROGRAMA DESTINADO AL ANÁLISIS DE LA MAGNITUD VELOCIDAD ANGULAR.**

En este apartado se describe la metodología utilizada para procesar la magnitud velocidad, además el proceso de codificación de la misma haciendo uso de varias secuencias.

#### **3.9.1 INTERFAZ GRÁFICA DE USUARIO DE LA MAGNITUD VELOCIDAD ANGULAR.**

La interfaz gráfica de usuario se compone de una serie de botones que permiten acceder a las diferentes etapas de la conversión de analógica digital de la magnitud velocidad.

La interfaz permite la visualización simultánea de la señal analógica original y las etapas de muestreo, cuantificación, codificación y representación digital del valor digital que toma dicha señal en un determinado instante de tiempo.

También es posible establecer la impresión de una pantalla específica a cada etapa según el requerimiento en un determinado instante de tiempo presionando el botón adecuado para este fin.

Se dispone de botones específicos para generar y guardar datos de mediciones de todas las magnitudes y a su vez abrir un historial de acuerdo a la fecha y hora requeridas.

En la figura (74) se observa la interfaz gráfica de usuario con todas las prestaciones disponibles.

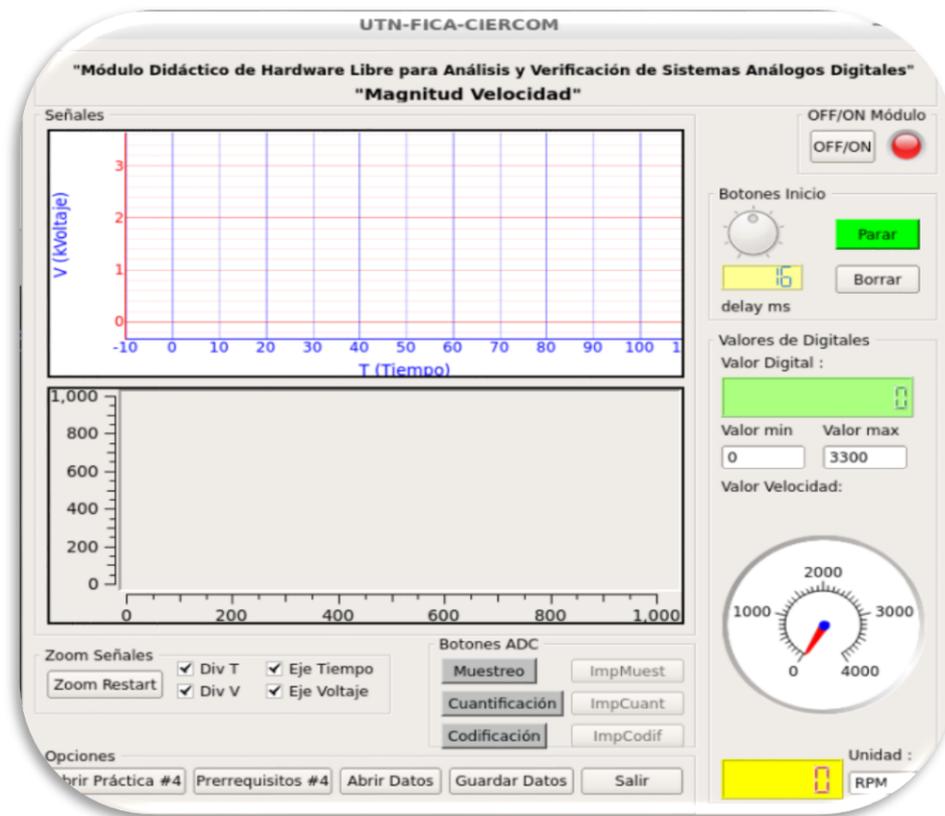


Figura 74 Interfaz gráfica de la magnitud velocidad angular.

Fuente: EL Autor.

### 3.9.2 DIAGRAMA DE CONEXIÓN DEL SENSOR DE VELOCIDAD ANGULAR.

Para el efecto es necesario utilizar una entrada analógica específica en el módulo de hardware libre previa a la energización adecuada del sensor de acuerdo con los niveles de voltaje requeridos.

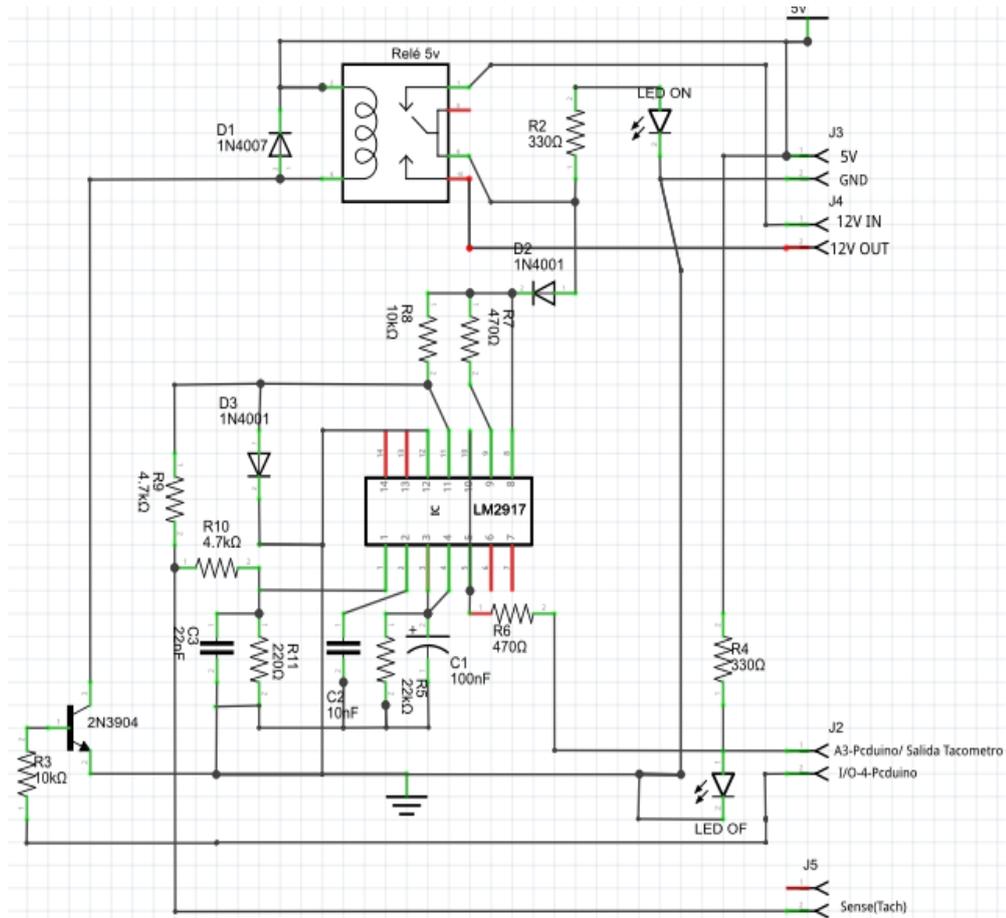


Figura 75 Diagrama esquemático del sensor de velocidad.

Fuente: EL Autor.

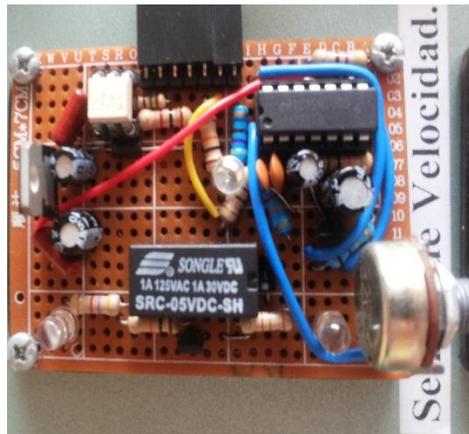


Figura 76 Diagrama pictórico del sensor de velocidad.

Fuente: EL Autor.

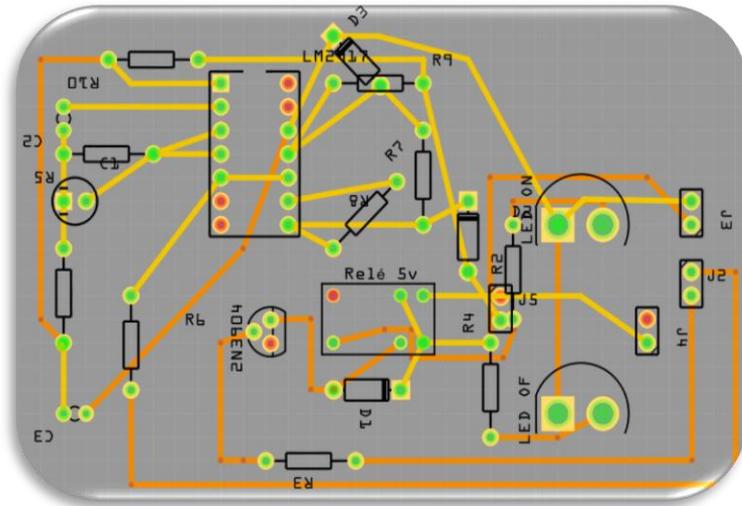


Figura 77 Diagrama circuito impreso del sensor de velocidad.

Fuente: EL Autor.

### 3.9.3 DESCRIPCIÓN DE FUNCIONAMIENTO DEL MÓDULO DE VELOCIDAD.

Se inicia por la activación del módulo a través de la GPIO-4 del Pc-Duino, la misma que al estar en 3,3v activa la bobina del relé permitiendo que el contacto normalmente abierto del mismo se accione, de esta manera se energizara el Lm2917 y se encenderá el led indicador verde. El Lm2917 al ser energizado funciona como conversor de frecuencia a voltaje, y entrega en su pin Out el voltaje proporcional a la frecuencia de entrada del mismo.

Haciendo un breve análisis de consumo de corriente del circuito del módulo de velocidad, se tiene que el consumo de corriente de cada uno de los elementos inmiscuidos en el mismo es: Lm2917 consume un promedio de 50mA, relé consume un promedio de 50mA, transistor consume un promedio 0,5mA, leds consumen un promedio 10mA. En general el consumo total del módulo de temperatura esta alrededor de 125mA, lo cual es un consumo de corriente pequeño.

### 3.9.4 CODIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE VELOCIDAD ANGULAR.

En referencia a codificar la señal proveniente del sensor de velocidad angular, el voltaje suministrado del LM2917 es lineal, al realizar una función directa obtenemos la velocidad en Rpm.

Valores obtenidos del sensor:

Voltaje	RPM
0	0
1	1800
1,83	3000

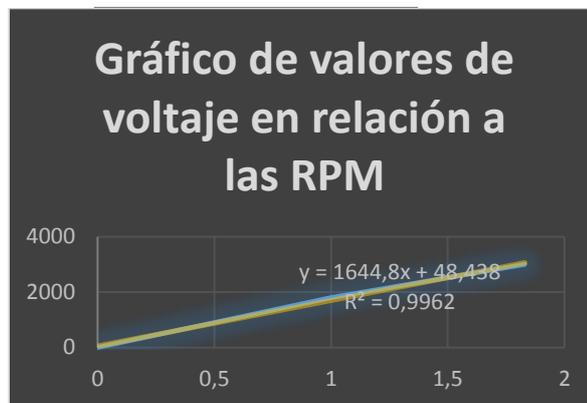


Figura 78 Valores de voltaje en relación a rpm.

Fuente: El Autor.

La función lineal,  $y = 1644,8x + 48,438$  en color azul en la gráfica es la función que va a gobernar el cambio de valores en la medición de rpm.

$$y = 1644,8x + 48,438 \dots (14)$$

Ecuación 14 Función lineal de Voltaje - RPM.

### 3.10 CONVERSIÓN DIGITAL A ANALÓGICA

La conversión digital a analógica es la contrapuesta de la conversión analógica digital. En la conversión analógica digital, iniciábamos con una tensión continua en la entrada del conversor. El conversor analógico digital asignaba un

valor al paso de tensión más cercano al real y enviaba un número binario indicando el paso medido.

La conversión digital a analógica inicia en la entrada con un número en forma binaria y se obtiene una salida con variación de tensión. La señal de salida no es un valor analógico efectivo que varía consecutivamente en el tiempo, sin embargo es una tensión discreta que varía a pasos.

El número de niveles de tensión que un conversor digital analógico consigue originar, está explícita por el número de bits binarios que logra manipular, esto indica la resolución. Podemos usar nuevamente la ecuación de combinaciones para obtener la cantidad.

El conversor digital analógico que se utiliza como modo de ejemplo tiene una resolución de 4-bits, así que el número de niveles de tensión en la salida del mismo es:

$$\text{combinaciones} = 2^{\text{bits}} = 2^4 = 16$$

### **3.10.1 RED R2R**

Montamos un conversor digital analógico empleando resistencias eléctricas, a este circuito se lo denomina R2R (red resistiva en escalera), la misma que permite agregar o quitar resistencias para alterar la resolución. Con una R2R en escalera, si inicia con un conversor de 4-bits y se requiere incrementar la resolución en 1-bit, simplemente debe adicionar dos resistencias a la misma.

La entrada del número binario la realiza en paralelo como 4-bits por las GPIO PG1, PG3, PG5, PG7 del Cubietruck. Mientras que se le asigne un valor a los cuatro pines GPIO antes mencionados, la salida del conversor digital analógico (output DAC) poseerá un valor discreto de tensión.

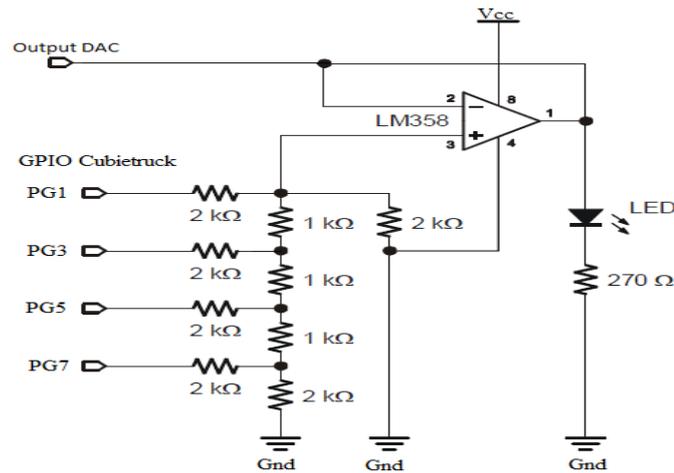


Figura 79 Circuito R2R conversión digital analógico.

Fuente: El Autor.

El voltaje de salida del conversor digital analógico sin un LM358 (buffer) puede alcanzar un valor máximo por debajo de los 3 Voltios que es capaz de entregar el conversor digital analógico expuesto. Sin embargo, el voltaje de salida con un LM358 (buffer) alcanza los 3 Voltios sin inconvenientes.

La ecuación que gobierna la red R2R, se expresa en función del voltaje suministrado por las salidas digitales del Cubietruck de 3,3v, donde PG1-PG7 tomaran un valor de uno o cero:

$$Output\ DAC = \frac{3,3v}{2^4} * (PG1 * 2^3 + PG3 * 2^2 + PG5 * 2^1 + PG7 * 2^0) \dots (15)$$

*Ecuación 15 Ecuación de R2R.*

Fuente: (Tocci & Welmer, 2003).

### 3.10.1.1 Interfaz gráfica del módulo R2R

La interfaz gráfica del módulo R2R, se lo realiza de la misma manera que en los apartados anteriores.

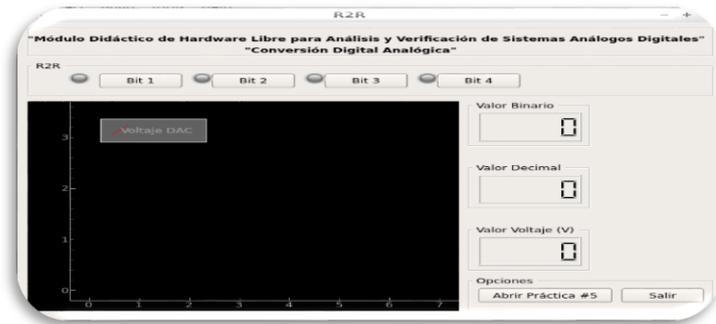


Figura 80 Interfaz gráfica de R2R.

Fuente: El Autor.

Contiene información del trabajo de grado, una sección de R2R de activación de los bits del Cubietruck los mismos que acceden a los registros (archivos) de los pines a activar y se cambia los valores. A continuación se presenta una lista de los registros utilizados para activar o desactivar el bit para la red R2R. El valor a adherirse a los registros es 1 para que el pin este 3,3v o 0 para que el pin este a 0v. El bit más significativo es el Bit 4 mientras que el bit menos significativo es el bit 1 para la red R2R.

Tabla 30 Directorios de pines GPIO Cubietruck.

<code>gpio="/sys/class/gpio/gpio3_pg1/value"</code>
<code>gpio5="/sys/class/gpio/gpio5_pg3/value"</code>
<code>gpio7="/sys/class/gpio/gpio7_pg5/value"</code>
<code>gpio9="/sys/class/gpio/gpio9_pg7/value"</code>

Fuente: El Autor.

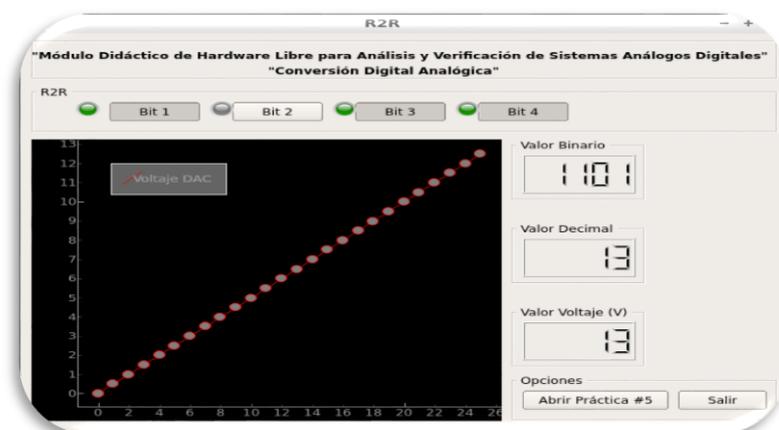


Figura 81 Graficación del valor de R2R.

Fuente: El Autor.



Entre las principales aplicaciones en las cuales se inmiscuye PWM están, Servo-motor, y control de velocidad de giro de los motores eléctricos

### 3.10.2.1 Interfaz gráfica del módulo de PWM.

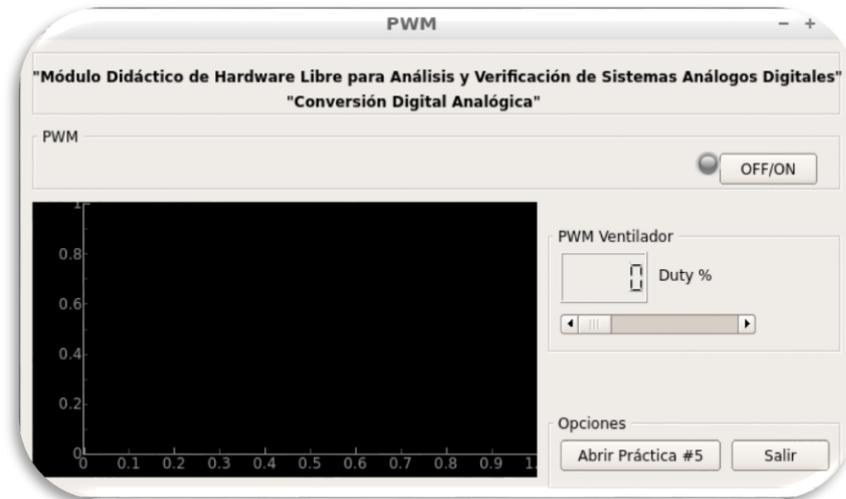


Figura 84 Interfaz gráfica de PWM.

Fuente: El Autor.

El botón OFF/ON activa groupBox-Pwm Ventilador el mismo que contiene un QscrollBar - DutyPWM que es el encargado de cambiar el porcentaje del ciclo útil de pwm, el cambio de los valores se observaran en el LCDNumber-PWM y también en el gráfico de anexo.



Figura 85 Graficación de la señal PWM.

Fuente: El Autor.

Los principales archivos de PWM en la placa Cubietruck son los siguientes: `"/sys/class/pwm-sunxi/pwm0/duty_percent"`, `"/sys/class/pwm-`

sunxi/pwm0/period", "/sys/class/pwm-sunxi/pwm0/run"; la manera de generar una onda PWM es variar los valores en dichos archivos.

*Tabla 31 Archivos de PWM Cubietruck.*

```
duty="/sys/class/pwm-sunxi/pwm0/duty_percent"
periodo="/sys/class/pwm-sunxi/pwm0/period" # echo 150khz > /sys/class/pwm-sunxi/pwm0/period
run="/sys/class/pwm-sunxi/pwm0/run"
```

Fuente: El Autor.

Para acceder a variar el valor del ciclo útil de PWM en la placa Cubietruck se deben adherirse los valores del QscrollBar – DutyPWM al archivo "/sys/class/pwm-sunxi/pwm0/duty\_percent", a continuación se presenta el método completo de variar el valor del ciclo útil.

*Tabla 32 Código del Método de cambiar el valor de ciclo útil de PWM.*

```
def timerClockEvent(self):
    y=self.lcdPWMDUTY.value()
    i=int(y)
    x=str(i)
    file=open(duty,'w')
    file.write(x)
    time.sleep(0.02) # Retardo por 0.2 segundos
```

Fuente: El Autor.

El gráfico demuestra la variación del ciclo útil de PWM, se emplea la función signal.square de la librería scipy (herramientas numéricas para Python), que devuelve una forma de onda periódica de una onda cuadrada. La función utilizada en scipy.signal.square(t, duty=0.5), a continuación se presenta el método completo de graficación de la señal PWM.

*Tabla 33 Código de graficación de la señal PWM.*

```
def setup(self):
    global b
    b=self.DutyPWM.value()*0.01
    t = np.linspace(0, 1, 500, endpoint=False)
    pwm = signal.square(1 * np.pi * 10 * t, duty=b)
    self.graficoDAC.addLegend()
    self.graficoDAC.plot(pwm, pen='r', name='PWM')
    self.connect(self.DutyPWM, SIGNAL("sliderMoved(int)"), self.setup1)
```

Fuente: El Autor.

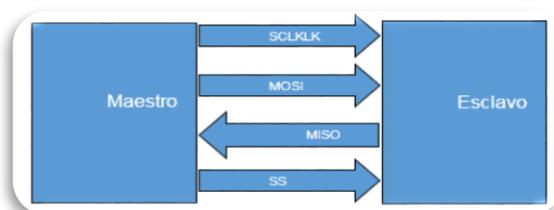
### 3.10.3 SPI

Abreviatura de "Interfaz Periférico Serial", es un bus de cuatro líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas cuatro líneas porta la información entre los diferentes dispositivos conectados al bus, ya sea esto en una comunicación simple de un maestro y un esclavo, como en la de un maestro y varios esclavos.

Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es llamado también full dúplex.

Entre la aplicaciones en las cuales se puede encontrar, más adecuado para "flujos de datos", es decir convertidores ADC, capacidad duplex, es decir, la comunicación entre un codec y procesador de señal digital.

Las características del Protocolo Serial Peripheral Interface (SPI) son entre otras, soporta sólo un maestro; puede soportar más de un esclavo; a corta distancia entre dispositivos, por ejemplo, en un placas de circuito impreso (PCB); Debe ser observado que la polaridad y la fase de la señal de reloj especial atención; el maestro envía datos sobre uno de los bordes de reloj y lee los datos en el otro borde. Por lo tanto, se puede enviar / recibir al mismo tiempo.



*Figura 86 Configuración del bus SPI.*

Fuente: El Autor.

La configuración de bus SPI, enlace de datos en serie síncrono que funciona en una relación full dúplex maestro/esclavo; 2 señales de datos: MOSI - Salida de datos maestros, la entrada de datos de esclavos; MISO - Entrada de datos maestros, la salida de datos de esclavos 2 señales de control: SCLK – Reloj/SS - seleccionar al esclavo.

Dentro de la estructura de SPI se observa como el registro transmite el byte para el esclavo en la línea de señal de MOSI, el esclavo transfiere los contenidos de su registro de desplazamiento de nuevo a la principal en la línea de señal MISO, intercambiar los contenidos de los dos registros de desplazamiento.

### 3.10.3.1 Interfaz gráfica de matriz de leds.



*Figura 87 Interfaz gráfica de matriz de Leds.*

Fuente: El Autor.

El botón OFF/ON activa los campos del groupBox-Matriz de Leds el mismo que contiene: Texto a Visualizar.- Es un LineEdit en el cual se introducirá el texto que en la matriz de leds se visualizara esto es posible mediante la utilización de la librería de max7219; el QDial- DelayMatriz proporcionara el tiempo de retardo que el texto se visualizara en la matriz de leds, el valor se puede observar en el LDCNumber-IcdDelayMatriz; el botón de Reset borra el texto en LineEdit; el botón OK agrega el texto y el retardo a la matriz de leds a continuación de presenta una tabla que contiene el código de este botón.

*Tabla 34 Código del Botón OK módulo matriz de leds.*

```
def botonMatrizOK(self):
    device = led.matrix()
    device.orientation(180)
    s = self.lineEdit.text()
    maxt = str(s)
    a=self.DelayMatriz.value()*0.01
    device.show_message(maxt, font=proportional(CP437_FONT),delay=a)
```

Fuente: El Autor.

En el groupBox-Opciones, se tiene el botón-Abrir Práctica #5 el mismo que al ser presionado abre la interfaz gráfica de la Práctica #5, el botón salir cierra la interfaz gráfica.



*Figura 88 Matriz de Leds.*

Fuente: El Autor.

La finalidad del módulo de Matriz de Leds es evidenciar el proceso de envío de datos digitales los mismos que serán representados mediante la activación y desactivación de los 64 leds que componen la matriz, demostrando la conversión digital analógica.

### **3.11 PROCESO DE ENSAMBLAJE DEL MÓDULO DIDÁCTICO**

En este apartado se detalla el proceso de ensamblaje del módulo didáctico, abordándolo desde una manera específica de cada una de las etapas que componen al mismo.

#### **3.11.1 DIMENSIONES DEL MÓDULO DIDÁCTICO**

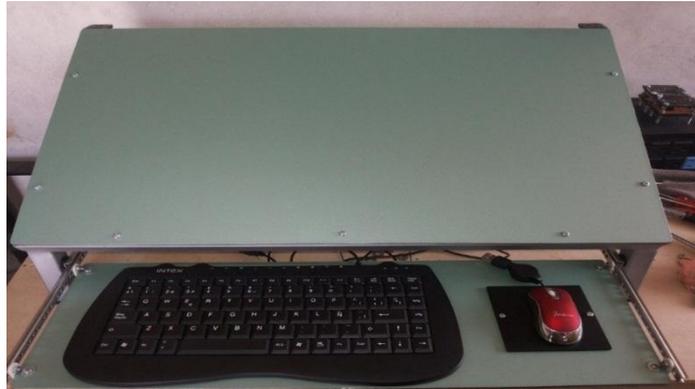
El modulo didáctico contiene una infraestructura metálica de tubo cuadrado, con dimensiones:

*Tabla 35 Tabla de dimensiones del módulo didáctico.*

<b>Altura: parte frontal 12 cm, parte posterior 27cm</b>
<b>Ancho 62 cm</b>
<b>Largo 32 cm</b>

Fuente: El Autor.

Contiene también una placa frontal de ALUCOBOND, en donde se alojan las diferentes etapas.

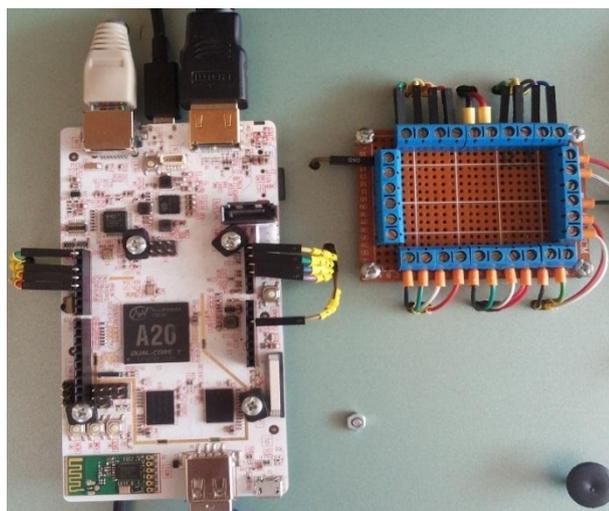


*Figura 89 Estructura del módulo didáctico.*

Fuente: El Autor.

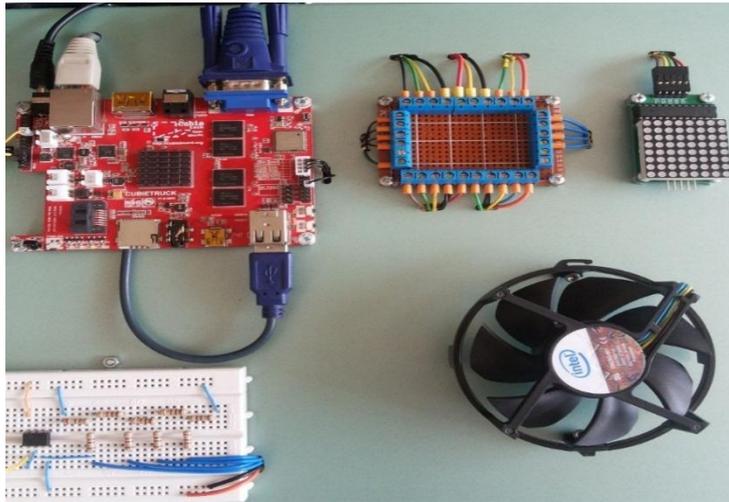
### **3.11.2 POSICIONAMIENTO DE LOS MÓDULOS DE HARDWARE LIBRE.**

Estos dispositivos son integrados al módulo mediante la utilización de pernos de anclaje introducidos y sujetos, utilizando los orificios que estos contienen para este propósito. Para facilitar la manipulación de las entradas y salidas digitales y analógicas se dispone de un conjunto de borneras que se destinan una por cada señal que se requiera procesar.



*Figura 90 Montaje y Cableado de las entradas analógicas del Pcdiuno.*

Fuente: El Autor.

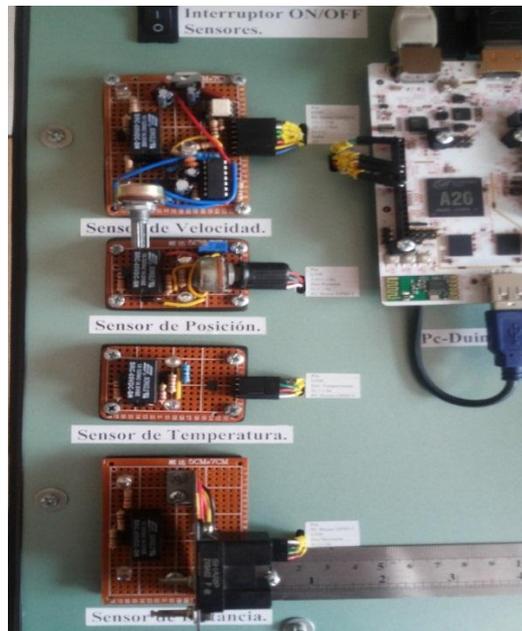


*Figura 91 Montaje del cubietruck.*

Fuente: El Autor.

### 3.11.3 POSICIONAMIENTO DE LOS SENSORES EN EL MÓDULO.

Cada uno de los sensores implementados posee cuatro orificios en los cuales ingresan los pernos que sujetan cada bloque a la placa frontal del módulo. Las señales de alimentación y magnitud que cada sensor necesita se toman del bloque del conjunto de borneras destinado para cada módulo de hardware libre.



*Figura 92 Posicionamiento de los módulos de los sensores.*

Fuente: El Autor.

Para facilitar la identificación del cableado utilizado se destina un color específico para cada requerimiento.

*Tabla 36 Tabla de colores de cables utilizados para los sensores.*

Color de Cable	Conector del sensor
Blanco	GPIO Pcdduino
Negro	Tierra (Gnd)
Verde	Señal de salida del sensor
Rojo	Alimentación de voltaje (Vcc)

Fuente: El Autor.

Una vez establecido el código de colores se procede a cablear las GPIO del Pcdduino con los sensores y las entradas analógicas asignadas a los mismos mediante la utilización del conjunto de borneras.

*Tabla 37 Cables utilizados para las GPIO del Pcdduino y entradas analógicas asignadas para los sensores.*

GPIO Pcdduino	Colores de los cables utilizados	Entradas analógicas del Pcdduino	Sensor
0	Verde	A5	Temperatura
2	Amarillo	A4	Posición
3	Tomate	A2	Distancia
4	Rojo	A3	Velocidad

Fuente: El Autor.

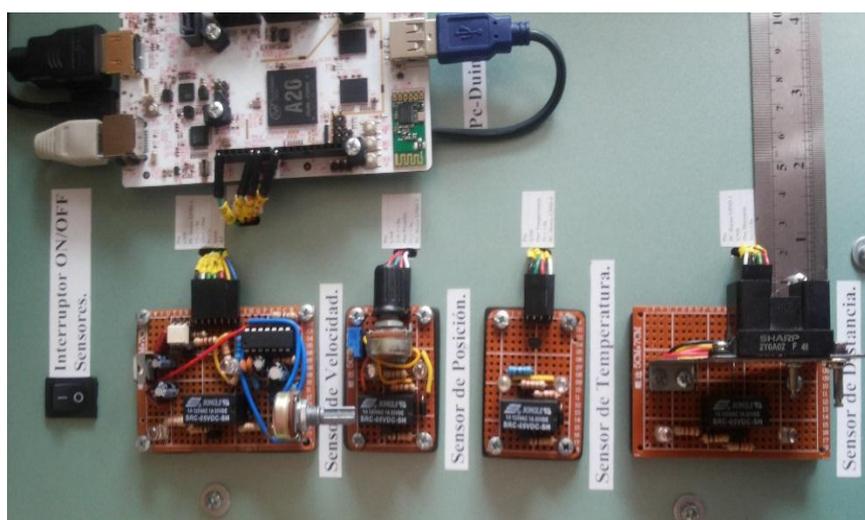


Figura 93 Cableado de los sensores del Pcdduino.

Fuente: El Autor.

### 3.11.4 FUENTE DE ALIMENTACIÓN

Se trata de una fuente de suministro que posee diferentes niveles de voltaje de acuerdo con los requerimientos del módulo en su totalidad, además posee la característica de protección contra cortocircuitos para precautelar el funcionamiento de todos los dispositivos y evitar posibles averías en los mismos. Se trata de una fuente RexPower ATX PM-350 que tiene las siguientes especificaciones.

*Tabla 38 Especificaciones fuente de alimentación.*

AC	Output DC	Corriente Max	Corriente Min
115V – 60Hz	3.3V	20A	0A
	5V	40A	0A
	12V	16A	0A

Fuente: El Autor.

Se exhibe una tabla con la estimación de consumo de cada uno de los módulos de sensores:

*Tabla 39 Estimación de consumo de corriente de los módulos de sensores.*

Módulos	Corriente (mA)
Módulo de temperatura	70
Módulo de distancia	100
Módulo de posición	70
Módulo de velocidad	125
Matriz de Leds	20
Ventilador	140
R2R	15
<b>Total</b>	<b>540</b>

Fuente: El Autor.

La sumatoria de cada uno de los módulos de sensores (125mA + 70mA + 100mA + 70mA + 125mA + 20mA + 140mA + 15mA) no supera 600mA, es decir son módulos de bajo consumo de corriente. La fuente que se ha seleccionado

supera enormemente la capacidad de corriente de consumo de los elementos del módulo didáctico.

La fuente de poder es un dispositivo esencial en el módulo didáctico, ya que provee la energía eléctrica en distintos niveles ya sean estos: de 5v para los módulos de sensores, de 12v para el módulo de velocidad incluido el ventilador y de 3,3v para el módulo R2R. Se ha elegido una fuente de PC para suministrar la energía puesto que la misma cuenta con los niveles voltajes suficientes para el módulo didáctico, también que la fuente conserva una protección contra sobrecargas y cortocircuitos que son características de gran funcionalidad para el módulo didáctico. Si fuese el caso de que algún estudiante al realizar las practicas genera un corto circuito la fuente se debe desconectar y para retornar a ponerla en ejecución es ineludible resetearla.

### 3.11.5 PANTALLA PARA VISUALIZACIÓN DE LA INTERFAZ GRÁFICA.

La interfaz gráfica del módulo que se detalla requiere de la utilización de una pantalla que permita visualizar todas sus funciones, en consecuencia se procede a utilizar un monitor de uso común de la casa comercial Samsung y que presenta las siguientes características.

Tabla 40 Características de la pantalla.

<b>Monitor</b>	<b>Samsung SyncMaster 632NW</b>
<b>Display</b>	15.6"
<b>Max Resolución</b>	1360 x 768
<b>Señal entrada</b>	VGA
<b>Voltaje Requerido</b>	AC 120V – 60Hz

Fuente: El Autor.

### 3.11.6 ENSAMBLAJE TOTAL DEL MÓDULO DIDÁCTICO.

#### 3.11.6.1 Dispositivos y materiales necesarios.

Para garantizar la funcionalidad total del módulo que se presenta es necesaria la utilización de una serie de dispositivos, elementos y materiales agrupados de tal manera que faciliten el desarrollo de todas las prácticas con sus respectivas prestaciones y funcionalidades y a modo de resumen de todo lo anteriormente descrito en lo referente a equipos utilizados se detalla la tabla siguiente.

*Tabla 41 Dispositivos y materiales del módulo didáctico.*

<b>Pantalla</b>	<b>Samsung SyncMaster 632NW</b>
<b>Fuente de alimentación</b>	RexPower ATX PM-350
<b>Teclado</b>	Intex
<b>Mouse</b>	Genius
<b>KVMSwitch</b>	KVM41UA – 4 Port
<b>Módulo de hardware libre 1</b>	Pcduino 3
<b>Módulo de hardware libre 2</b>	Cubietruck
<b>Alimentación DC Pcduino</b>	EasyAcc 11UNMIC5P-2S5FTUS-(5V-2000mA)
<b>Alimentación DC Cubietruck</b>	SunFounder HZPW- 5V2.5A
<b>Cables de KVM</b>	
<b>Convertor HDMI-VGA</b>	
<b>Protoboard</b>	Wishboard 1-Regleta
<b>Juegos de Borneras</b>	
<b>Módulo Sensor de Velocidad</b>	
<b>Módulo Sensor de Posición</b>	
<b>Módulo Sensor de Temperatura</b>	
<b>Módulo Sensor de Distancia</b>	
<b>Juegos de cables</b>	

Fuente: El Autor.

Se puede observar en la figura la implementación total del módulo didáctico, se observa las regiones de sensores, las posiciones de los módulos de hardware libre, los módulos de conectores 1 y 2, y la región del protoboard.

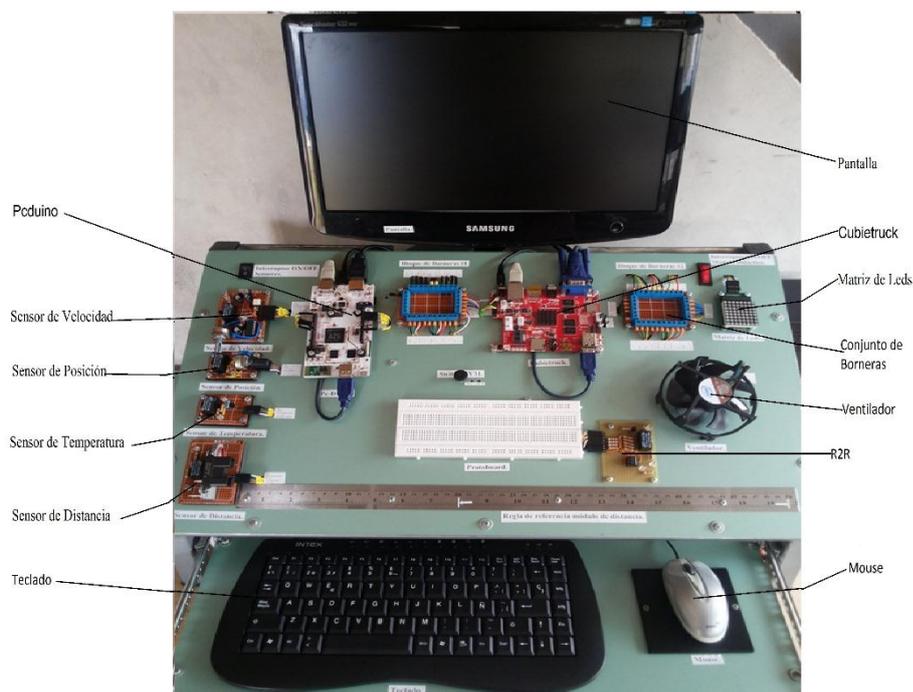


Figura 94 Ensamblaje total del módulo didáctico.

Fuente: El Autor

### 3.12 ANÁLISIS ECONÓMICO DEL MÓDULO DIDÁCTICO

En este apartado se detallan los dispositivos y materiales del módulo didáctico, enfocándose en el costo que representa cada uno de los mismos, para hacer posible la construcción en su totalidad del módulo didáctico.

Tabla 42 Costo de dispositivos y materiales del módulo didáctico.

Cantidad	Dispositivos	Valor u/c (\$)	Subtotal (\$)
1	Pantalla	94	94
1	Fuente de alimentación	25	25
1	Teclado	9	9
1	Mouse	5	5
1	KVMSwitch	26	26
1	Módulo de hardware libre 1	150	150
1	Módulo de hardware libre 2	190	190
1	Alimentación DC Pduino	15	15
1	Alimentación DC Cubietruck	15	15
2	Cables de KVM	7	14
1	Conversor HDMI-VGA	22	22
1	Protoboard	12	12

<b>1</b>	Juegos de Borneras	10	<b>10</b>
<b>1</b>	Módulo Sensor de Velocidad	31,15	<b>31,15</b>
<b>1</b>	Módulo Sensor de Posición	7,1	<b>7,1</b>
<b>1</b>	Módulo Sensor de Temperatura	8,6	<b>8,6</b>
<b>1</b>	Módulo Sensor de Distancia	30,85	<b>30,85</b>
<b>1</b>	Juegos de cables	10	<b>10</b>
Dispositivos y materiales del módulo didáctico			<b>674,7</b>

Fuente: El Autor.

### 3.12.1 MATERIALES PARA CADA UNO DE LOS MÓDULOS DE SENSORES

Se detalla el costo de cada uno de los elementos de los módulos de sensores, tratándose de elementos fáciles de conseguir, el costo no es elevado.

#### 3.12.1.1 Costo módulo sensor de temperatura

Para construir el modulo del sensor de temperatura se utiliza los materiales electrónicos básicos tales como:

*Tabla 43 Costo módulo sensor de temperatura.*

<b>Cantidad</b>	<b>Material</b>	<b>Valor u/c (\$)</b>	<b>Subtotal (\$)</b>
<b>1</b>	Relé 5v	1,75	1,75
<b>1</b>	Transistor 2N3904	0,4	0,4
<b>4</b>	Resistencias	0,1	0,4
<b>1</b>	Diodo 1N4007	0,3	0,3
<b>2</b>	Leds	0,25	0,5
<b>1</b>	Sensor LM35	3,25	3,25
<b>1</b>	Otros	2	2
Valor del Módulo Sensor de Temperatura			<b>8,6</b>

Fuente: El Autor.

### 3.12.1.2 Costo módulo sensor de posición

Para construir el modulo del sensor de Posición se utiliza los materiales electrónicos básicos tales como:

*Tabla 44 Costo módulo sensor de posición.*

<b>Cantidad</b>	<b>Material</b>	<b>Valor u/c (\$)</b>	<b>Subtotal (\$)</b>
1	Relé 5v	1,75	1,75
1	Transistor 2N3904	0,4	0,4
4	Resistencias	0,1	0,4
1	Diodo 1N4007	0,3	0,3
2	Leds	0,25	0,5
1	Sensor Posición	1,75	1,75
1	Otros	2	2
Valor del Módulo Sensor de Posición			7,1

Fuente: El Autor.

### 3.12.1.3 Costo módulo sensor de distancia

Para construir el modulo del sensor de Distancia se utiliza los materiales electrónicos básicos tales como:

*Tabla 45 Costo módulo sensor de distancia.*

<b>Cantidad</b>	<b>Material</b>	<b>Valor u/c (\$)</b>	<b>Subtotal (\$)</b>
1	Relé 5v	1,75	1,75
1	Transistor 2N3904	0,4	0,4
4	Resistencias	0,1	0,4
1	Diodo 1N4007	0,3	0,3
2	Leds	0,25	0,5
1	Sensor GP2Y0A02YK0F	25,5	25,5
1	Otros	2	2
Valor del Módulo Sensor de Distancia			30,85

Fuente: El Autor.

### 3.12.1.4 Costo módulo sensor de velocidad

Para construir el modulo del sensor de Velocidad se utiliza los materiales electrónicos básicos tales como:

*Tabla 46 Costo módulo sensor de velocidad.*

Cantidad	Material	Valor u/c (\$)	Subtotal (\$)
1	Relé 5v	1,75	1,75
1	Transistor 2N3904	0,4	0,4
4	Resistencias	0,1	0,4
1	Diodo 1N4007	0,3	0,3
2	Leds	0,25	0,5
1	IC LM2917	17,3	17,3
1	Ventilador 12v - 4 pines	8,5	8,5
1	Otros	2	2
Valor del Módulo Sensor de Velocidad			31,15

Fuente: El Autor.

### 3.12.2 COSTO DEL SOFTWARE UTILIZADO EN EL MÓDULO DIDÁCTICO

El software utilizado en general tiene la licencia GPL la misma que avala a los usuarios compartir, modificar, estudiarlos y usarlos según los propios requerimientos de los mismos.

*Tabla 47 Costo del software utilizado en el módulo didáctico.*

Software	Costo (\$)
Lubuntu Pcdino3	0
Lubuntu Cubietruck	0
Python - PyQt	0
Qt Designer Open Source	0
Valor del Software	0

Fuente: El Autor.

### 3.12.3 COSTO DEL DESARROLLO (INVESTIGACIÓN, DISEÑO, PROGRAMACIÓN Y PRUEBAS) DEL MÓDULO DIDÁCTICO

Se describe las horas utilizadas para desarrollar las diferentes actividades, haciendo hincapié al costo por hora, la misma que se ha tomado como referencia el salario básico unificado vigente.

*Tabla 48 Costo del desarrollo.*

<b>Horas</b>	<b>Desarrollo</b>	<b>Costo/Hora (\$)</b>	<b>Subtotal (\$)</b>
<b>250</b>	Investigación	2,18	545
<b>50</b>	Diseño	2,18	109
<b>150</b>	Programación	2,18	327
<b>50</b>	Pruebas	2,18	109
Valor del Desarrollo			1090

Fuente: El Autor.

### 3.12.4 COSTO TOTAL DEL MÓDULO DIDÁCTICO

Se realiza una adición de todos los costos realizados para obtener un valor simbólico referencial del módulo didáctico.

*Tabla 49 Costo total del módulo didáctico.*

	<b>Subtotal (\$)</b>
Dispositivos y materiales del módulo didáctico	674,7
Software	0
Desarrollo	1090
Costo total del módulo didáctico	1764,7

Fuente: El Autor.

### 3.12.5 ANÁLISIS BENEFICIOS

Con la implementación del módulo didáctico, se tiene beneficios en varios aspectos, a continuación se señalan:

- Disponibilidad de un instrumento de aprendizaje práctico a los estudiantes de la CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN (CIERCOM).
- Los estudiantes conseguirán una mejor formación profesional lo que conlleva a la concepción de un contingente humano con capacidad de desenvolverse, proyectarse y marchar acorde a los nuevos retos tecnológicos.
- Se complementa el laboratorio de ELECTRÓNICA Y REDES DE COMUNICACIÓN de la FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS, facilitando la implementación de prácticas que ayuden a los estudiantes a comprender y visualizar los métodos involucrados en procesos de conversión analógica digital.
- Se contribuye en “Asegurar en los programas educativos la inclusión de contenidos y actividades didácticas e informativas que motiven el interés por las ciencias, las tecnologías y la investigación, para la construcción de una sociedad socialista del conocimiento.” (Plan nacional del buen vivir objetivo 4. Fortalecer las capacidades y potencialidades de la ciudadanía parte 4.4 literal i, j.).

## 4 DESARROLLO DE GUÍA DE PRÁCTICAS.

Este capítulo presenta un manual de prácticas, pensado como una guía para que el alumno pueda realizar experimentos prácticos de adquisición de datos de los sensores de magnitudes físicas tales como la temperatura, distancia, posición, velocidad angular y verificación de los valores obtenidos, abordando cada una de las etapas que el módulo didáctico que se presenta posee, hasta llegar a realizar prácticas que demanden la utilización del sistema en su totalidad.

### 4.1 ESQUEMATIZACIÓN DE PRÁCTICAS

Para la realización del esquema de las prácticas se toma en cuenta la misma denominación de los componentes que el módulo tiene, los cuales fueron descritos en los capítulos anteriores. Debido a que el módulo ya se encuentra cableado y todos sus componentes están debidamente conectados hacia el Pcdduino y Cubietruck respectivamente, para establecer los requerimientos de cada una de las prácticas se debe referirse a la tabla 4-1, la cual describe cada una de las entradas analógicas y salidas GPIO, su función en cuanto se refiere a que elemento es comandado con cada una de estas. Por ejemplo si la práctica demanda la utilización del sensor de temperatura, bastará con utilizar la salida GPIO 0 del Pcdduino para activar o desactivar el módulo del sensor de temperatura, y así para todos los elementos que intervienen en el módulo.

*Tabla 4-1. Asignación de Entradas Analógicas y Pines GPIO hacia Sensores.*

<b>Nomenclatura Módulo de Velocidad</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>SV1</b>	GND	Negro
<b>SV2</b>	PC-Duino GPIO-4	Blanco
<b>SV3</b>	12v (+) In	Verde/Blanco
<b>SV4</b>	12v (+) Out	Verde/Negro
<b>SV5</b>	5v (+)	Rojo
<b>Nomenclatura Módulo de Posición</b>		
<b>Pin</b>	Valor Pin	Color de cable

<b>SP1</b>	GND	Negro
<b>SP2</b>	3.3v (+) In	Rojo
<b>SP3</b>	Out Posición	Verde
<b>SP4</b>	5v (+) In	Rojo
<b>SP5</b>	PC-Duino GPIO-2	Blanco
<b>Nomenclatura Módulo de Temperatura</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>ST1</b>	GND	Negro
<b>ST2</b>	Out Temperatura	Verde
<b>ST3</b>	5v (+) In	Rojo
<b>ST4</b>	PC-Duino GPIO-0	Blanco
<b>Nomenclatura Módulo de Distancia</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>SD1</b>	PC-Duino GPIO-3	Blanco
<b>SD2</b>	GND	Negro
<b>SD3</b>	Out Temperatura	Verde
<b>SD4</b>	5v (+) In	Rojo
<b>Nomenclatura Pc-Duino</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>GPIO0</b>	PC-Duino GPIO-0	Verde
<b>GPIO2</b>	PC-Duino GPIO-2	Amarillo
<b>GPIO3</b>	PC-Duino GPIO-3	Tomate
<b>GPIO4</b>	PC-Duino GPIO-4	Rojo
<b>A5</b>	PC-Duino A-5	Amarillo
<b>A4</b>	PC-Duino A-4	Verde
<b>A3</b>	PC-Duino A-3	Azul
<b>A2</b>	PC-Duino A-2	Morado
<b>GND</b>	PC-Duino GND	Café
<b>Nomenclatura Cubietruck</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>PC19</b>	Cubietruck PC19	Amarillo
<b>PC20</b>	Cubietruck PC20	Negro
<b>PC21</b>	Cubietruck PC21	Verde
<b>PC22</b>	Cubietruck PC22	Rojo
<b>PG1</b>	Cubietruck PG1	
<b>PG3</b>	Cubietruck PG3	
<b>PG5</b>	Cubietruck PG5	
<b>PG7</b>	Cubietruck PG7	
<b>PB0</b>	Cubietruck PB2 (PWM)	Amarillo
<b>GND</b>		
<b>Nomenclatura Módulo de Matriz de Leds</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>L1</b>	CLK	Café
<b>L2</b>	CS	Amarrillo
<b>L3</b>	DIN	Verde
<b>L4</b>	GND	Negro
<b>L5</b>	5(+) Vcc	Rojo
<b>Nomenclatura Módulo de R2R</b>		
<b>Pin</b>	Valor Pin	Color de cable
<b>G0</b>	Cubietruck PG0	Bit menos significativo (LSB)
<b>G1</b>	Cubietruck PG1	
<b>G2</b>	Cubietruck PG2	
<b>G3</b>	Cubietruck PG3	Bit más significativo (MSB)
<b>Nomenclatura Ventilador</b>		
<b>Pin</b>	Función	Color del Cable

F1	GND	Negro
F2	12V	Amarrillo
F3	Sense (tach.)	Verde
F4	Control (PWM)	Azul

Fuente: EL Autor.



Figura 95 Distribución de los elementos de sensores.

Fuente: EL Autor.



Figura 96 Distribución general de los elementos del módulo didáctico.

Fuente: EL Autor.



Figura 97 Etiquetado general de los elementos del módulo didáctico.

Fuente: EL Autor.

Resumen de las funcionalidades de las interfaces gráficas de usuario de cada una de las magnitudes físicas. Para más información dirigirse al capítulo 3 del trabajo de grado.

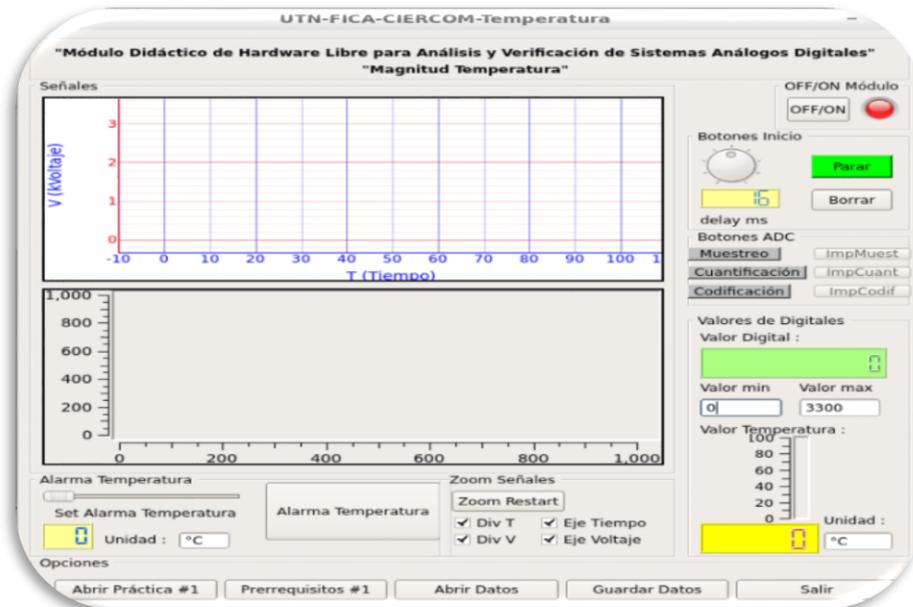


Figura 98 Interfaz gráfica de usuario general para las magnitudes físicas.

Fuente: EL Autor.

Al accionar el pulsador de INICIO en la interfaz gráfica de la magnitud física se activara la graficación de los datos adquiridos por sensor de la misma.



Figura 99 Botón OFF/ON Módulo.

Fuente: EL Autor.

Se puede hacer que la graficación se realice con más rapidez realizando un ajuste en la sección de botones de inicio en el DialDelay.



Figura 100 Sección botones de inicio.

Fuente: EL Autor.

En la sección de botones de inicio también se tiene el botón Parar que congela los datos graficados para realizar una verificación del valor. El botón se torna rojo hasta que se lo presione otra vez para que continúe con la graficación de los datos.



*Figura 101 Botón parar graficación.*

Fuente: EL Autor.

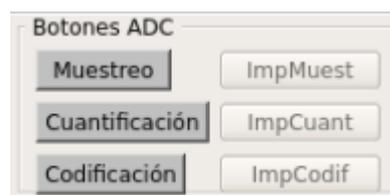
En la sección de botones de inicio también se tiene un botón Borrar el mismo que realiza una limpieza de los datos e inicializa la graficación de los datos obtenidos.



*Figura 102 Botón borrar graficación.*

Fuente: EL Autor.

En la sección de botones ADC se encuentran los botones de los procesos que intervienen en la conversión analógica digital. También se encuentran tres botones que imprimen la imagen del proceso de conversión analógica digital que este activo, para posteriormente anexarle al reporte de la práctica.



*Figura 103 Sección de botones ADC.*

Fuente: EL Autor.

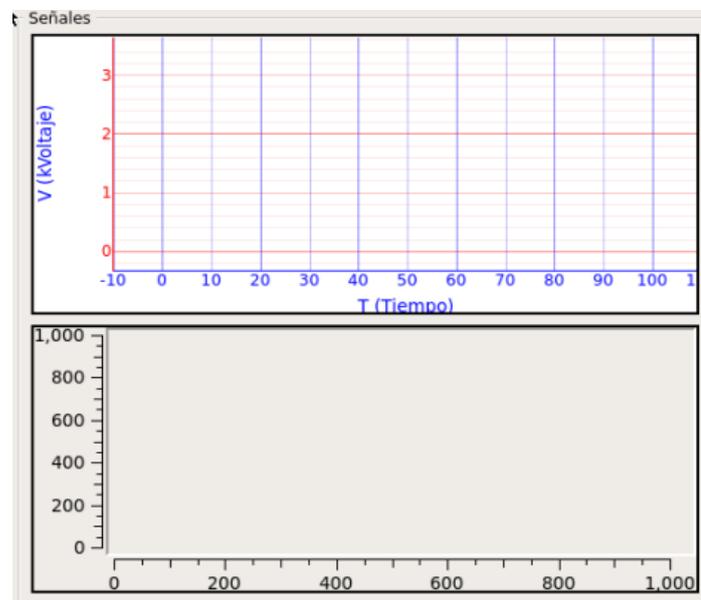
El botón muestreo realiza el primer proceso del ADC, el botón cuantificación realiza el segundo proceso del ADC. El botón codificación realiza el tercer proceso del ADC, en la gráfica se puede observar dicho procedimiento. También se puede apreciar los valores proceso de codificación en la sección de Valores Digitales.



*Figura 104 Sección valores digitales.*

Fuente: EL Autor.

En la sección de señales son representados gráficamente los valores obtenidos y los procesos de la conversión analógica digital.



*Figura 105 Sección de señales.*

Fuente: EL Autor.

En la sección de Zoon Señales se encarga de retornar a la representación de la señal a la escala asignada, también se puede ocultar los nombres de los ejes de la misma.



*Figura 106 Sección zoon señales.*

Fuente: EL Autor.

En la sección opciones se tiene varios botones cada uno de los cuales contiene una función asignada de la siguiente manera:



*Figura 107 Sección opciones.*

Fuente: EL Autor.

Abrir practica: al presionar este botón se redirige a la interfaz gráfica de la práctica correspondiente a cada magnitud física.

Prerrequisitos: al presionar este botón se redirige al archivo PDF de prerrequisitos de cada una de las practicas.

Guardar datos: guarda los datos de la magnitud física que se está censando, en un archivo de texto.

Abrir datos: con esta opción se puede visualizar los datos guardados de las diferentes magnitudes físicas.

## 4.2 PRÁCTICA # 1: VERIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE LA MAGNITUD TEMPERATURA.

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. Electrónica y Redes de Comunicación			

PRÁCTICA No.	LABORATORIO DE:	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
1	Electrónica	Verificación de la señal proveniente del sensor de la magnitud temperatura.	1

### 4.2.1 OBJETIVO

El alumno aprenderá a utilizar el módulo de temperatura como aplicación específica de adquisición de magnitud física temperatura, a través de un accionamiento directo por medio de la una GPIO 0 cuyo valor lógico activa o desactiva dicho módulo. Además el estudiante al culminar esta práctica será capaz de describir el proceso de muestreo que demanda la conversión analógica digital.

### 4.2.2 EQUIPO REQUERIDO

- 1 Pcdduino
- 1 Módulo de sensor de temperatura
- 1 Juego de cables
- Bloque de borneras # 1

### **4.2.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe revisar el manual del módulo didáctico.

El estudiante antes de realizar la práctica debe tener en claro los términos de: Pc-Duino, entrada analógica, GPIO, LM35, muestreo y responder las interrogantes que a continuación se listan:

¿Qué es un Pc-Duino?

¿Qué es una entrada analógica?

¿Qué es el LM35?

¿Qué es el muestreo?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Pc-Duino, módulo de temperatura, LM35, muestreo.

### **4.2.4 PROCEDIMIENTO**

Para identificar las partes de la interfaz gráfica de usuario diríjase a la sección 3.5.1.1 del trabajo de grado.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de temperatura, para el accionamiento del mismo conecte el pin de T-IN a la GPIO-0 del pcdduino. La salida del módulo de temperatura debe conectarse a la entrada analógica A5 del pcdduino. Verificar las conexiones del módulo de temperatura con el bloque de borneras con la función continuidad del multímetro. A continuación se presenta el diagrama de conexión del módulo de temperatura.

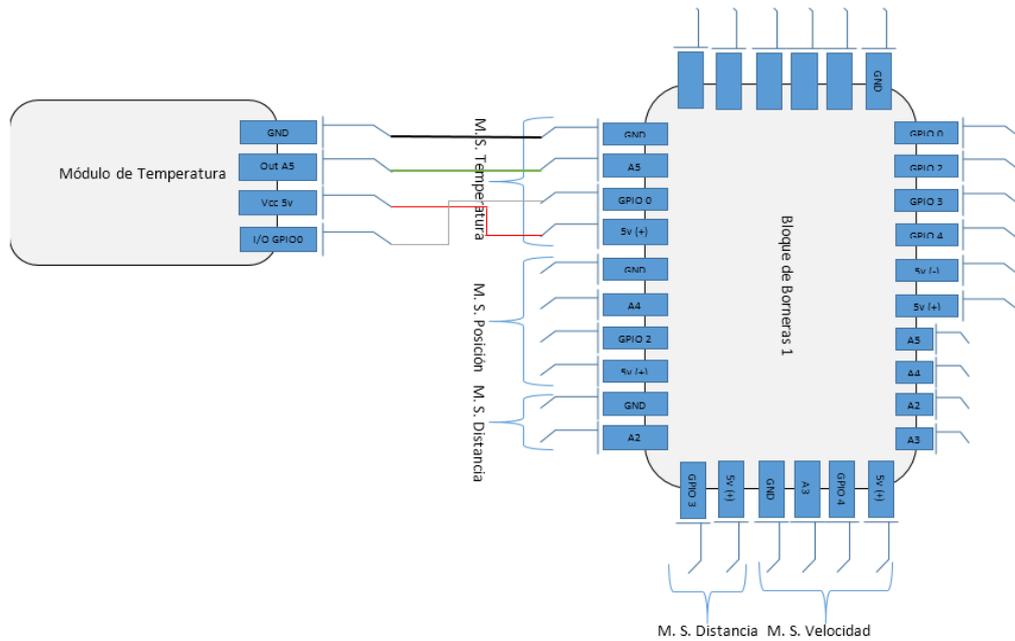


Figura 108 Diagrama para verificación de conexión modulo temperatura.

Fuente: EL Autor.

Al accionar el pulsador de INICIO en la interfaz gráfica de la magnitud temperatura se activara la graficación de los datos adquiridos por sensor de temperatura.



Figura 109 Botón OFF/ON módulo de temperatura.

Fuente: El Autor.

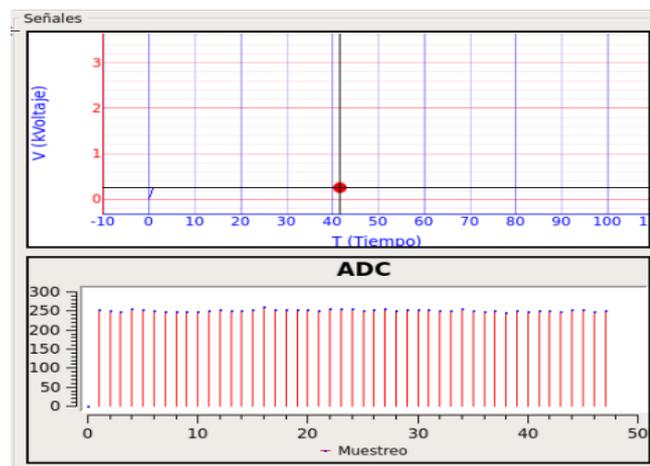
En la sección de botones ADC se encuentran el botón del proceso de muestreo, al ser activado también se activa el botón ImpMuest (imprimir muestreo) que al ser presionado realizara una captura del grafico del proceso del muestreo.



Figura 110 Botón muestreo del módulo de temperatura.

Fuente: El Autor.

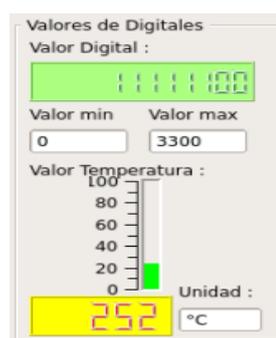
Presionando el botón del proceso de ADC correspondiente a muestreo se podrá evidenciar el método de muestrear la señal proveniente de la entrada analógica A5 del pduino, para más información de este proceso diríjase a la sección 3.5.3. del trabajo de grado. El proceso de graficación de la señal muestreada se podrá observar en el qwtplot, en la gráfica se puede observar dicho procedimiento.



*Figura 111 Señal muestreada, módulo de temperatura.*

Fuente: El Autor.

Presionando cada uno de los botones del proceso de ADC se podrá evidenciar los métodos por los cuales la señal debe pasar para convertirse en digital y representar el valor de la temperatura del medio, en el indicador digital.



*Figura 112 Sección de valores digitales, módulo de temperatura.*

Fuente: El Autor.

La sección de alarma de temperatura nos ofrece un regulador del valor de la alarma de temperatura (Set Alarma Temperatura), si el valor de la temperatura esta cinco unidades por debajo del Set Alarma el botón Alarma Temperatura

empezará a cambiar de color, dando como resultado una alerta visual que la temperatura está llegando al valor del Set de Temperatura.



Figura 113 Sección de alarma temperatura.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a la temperatura.

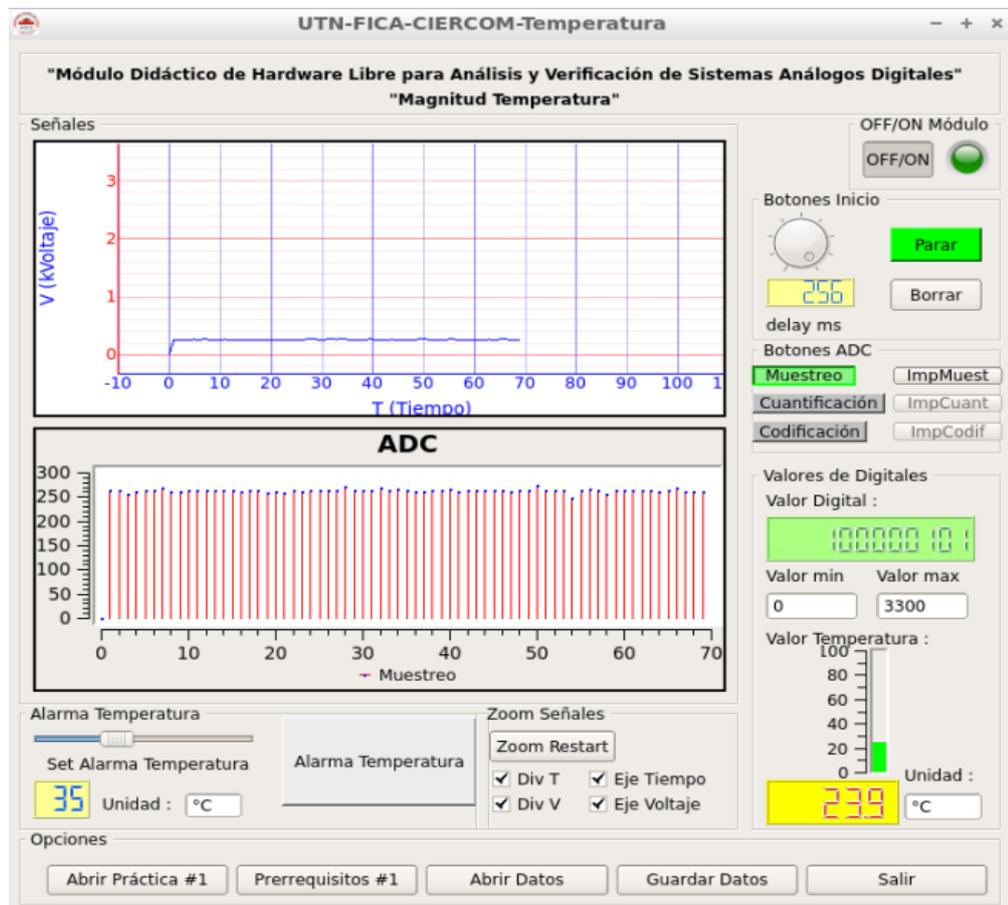
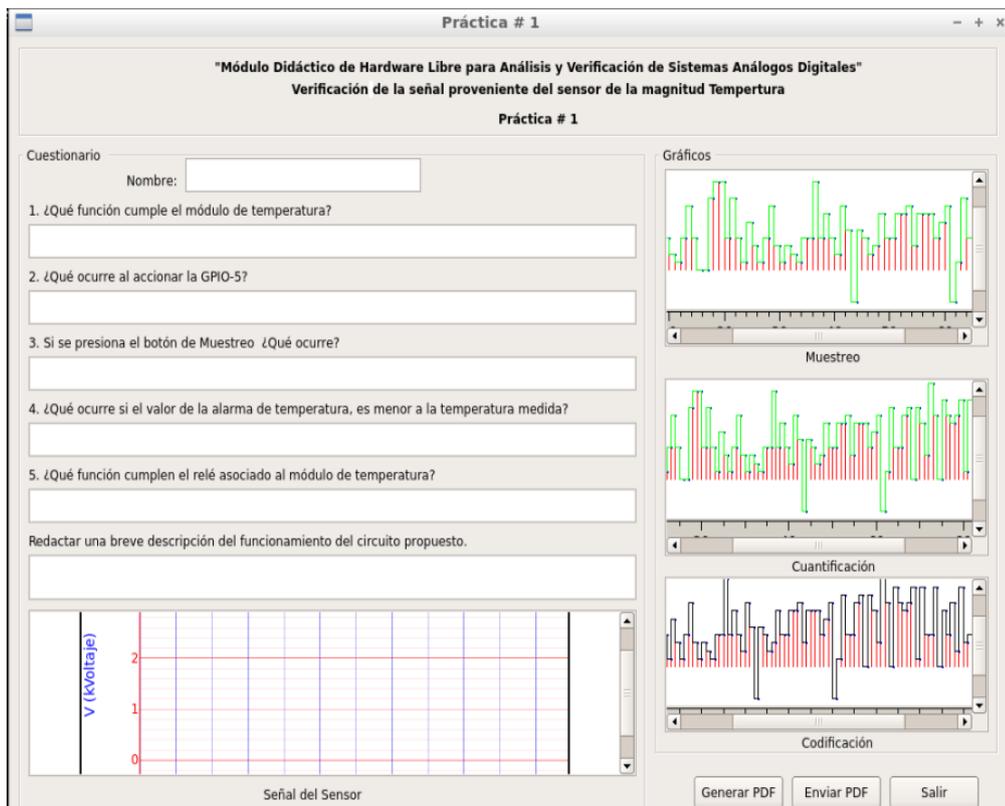


Figura 114 Interfaz gráfica de usuario del módulo de temperatura, en ejecución.

Fuente: El Autor.

Para conocer las funciones complementarias de la interfaz gráfica de usuario dirijase a la sección 3.6 del trabajo de grado y al manual de usuario del mismo.

Al presionar el botón Abrir Practica # 1, se direcciona a la interfaz gráfica de la misma; el estudiante debe escribir en todos los campos respondiendo las interrogantes, al presionar el botón generar pdf se crea un archivo de la practica realizada y si se presiona el botón enviar pdf el archivo de la practica será enviado al correo electrónico del docente encargado de la asignatura.



*Figura 115 Interfaz gráfica de la práctica #1.*

Fuente: El Autor.

#### 4.2.5 CUESTIONARIO

- a. ¿Qué función cumple el módulo de temperatura?

- b. ¿Qué ocurre al accionar la GPIO-0?
  
- c. Si se presiona el botón de Muestreo ¿Qué ocurre?
  
- d. ¿Qué ocurre si el valor de la alarma de temperatura, es menor a la temperatura medida?
  
- e. ¿Qué función cumplen el relé asociado al módulo de temperatura?

#### **4.2.6 FUNCIONAMIENTO Y CONCLUSIONES**

Redactar una breve descripción del funcionamiento del circuito propuesto.

.....

.....

.....

.....

.....

.....

.....

#### **4.2.7 ACTIVIDADES PROPUESTAS**

- Realizar la simulación del circuito del módulo de temperatura mediante la utilización de algún software (Proteus, Fridzing), y verificar el correcto funcionamiento del mismo.
- Implementar el circuito del módulo de temperatura en un protoboard, conectarlo a la entrada A5 y comprobar el funcionamiento.

- Realice una investigación de los diferentes sensores de temperatura existentes en el mercado y exponga a sus compañeros en un tiempo máximo de diez minutos.
- Realice una tabla con los valores medidos de temperatura en distintos intervalos de tiempo, se recomienda utilizar una fuente de calor y una fuente de enfriamiento.
- Cree un scripts que almacene los valores del sensor de temperatura cada cinco minutos.
- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.

**PRECAUCIÓN:**

**Revise las conexiones entre el módulo de temperatura antes de conectarlo al pduino, recuerde que el voltaje máximo de una entrada analógica de 12 Bits del pduino es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

### 4.3 PRÁCTICA # 2: VERIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE LA MAGNITUD DISTANCIA.

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. Electrónica y Redes de Comunicación			

PRÁCTICA No.	LABORATORIO DE:	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
2	Electrónica	Verificación de la señal proveniente del sensor de la magnitud distancia.	1

#### 4.3.1 OBJETIVO

El alumno aprenderá a utilizar el módulo de distancia como aplicación específica de adquisición de magnitud física distancia, a través de un accionamiento directo por medio de la una GPIO 3 cuyo valor lógico activa o desactiva dicho módulo. Además el estudiante al culminar esta práctica será capaz de describir el proceso de cuantificación que demanda la conversión analógica digital.

#### 4.3.2 EQUIPO REQUERIDO

- 1 Pcdduino
- 1 Módulo de sensor de distancia
- 1 Juego de cables
- Bloque de borneras # 1

### **4.3.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe tener en claro los términos de: Pc-Duino, sensor de distancia, Sharp GP2Y0A02YK0F, cuantificación y responder las interrogantes que a continuación se listan:

¿Qué es un Pc-Duino?

¿Qué es una GPIO?

¿Qué es el Sharp GP2Y0A02YK0F?

¿Qué es la cuantificación?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Pc-Duino, módulo de distancia, Sharp GP2Y0A02YK0F, cuantificación.

### **4.3.4 PROCEDIMIENTO**

Para identificar las partes de la interfaz gráfica de usuario diríjase a la sección 3.5.1.1 del trabajo de grado.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de distancia, para el accionamiento del mismo conecte el pin de D-IN a la GPIO-3 del pduino. La salida del módulo de distancia debe conectarse a la entrada analógica A2 del pduino. Verificar las conexiones del módulo de distancia con el bloque de borneras # 1 con la función continuidad del multímetro.

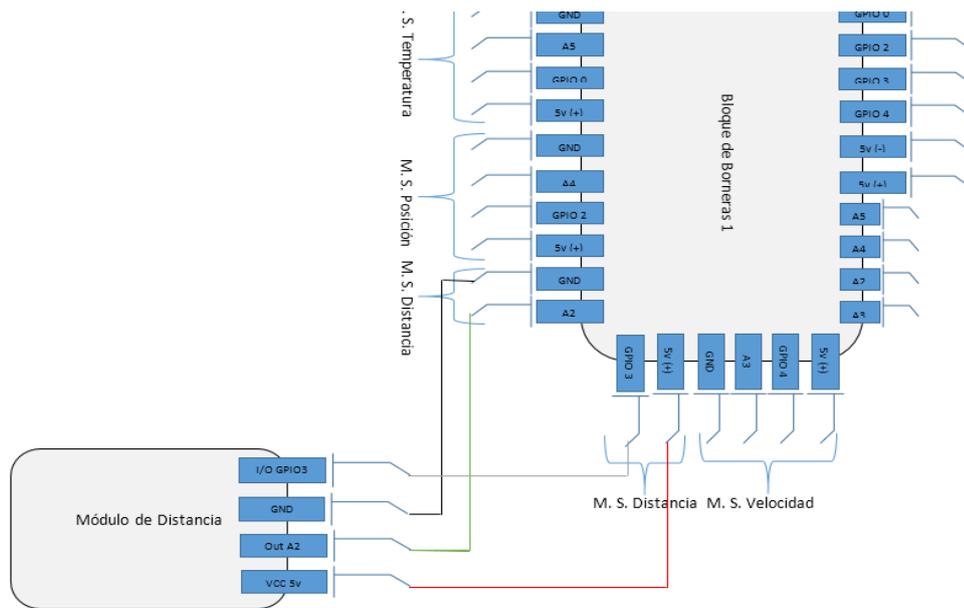


Figura 116 Diagrama para verificación de conexión modulo distancia.

Fuente: EL Autor.

La interfaz gráfica de usuario de distancia es la siguiente:

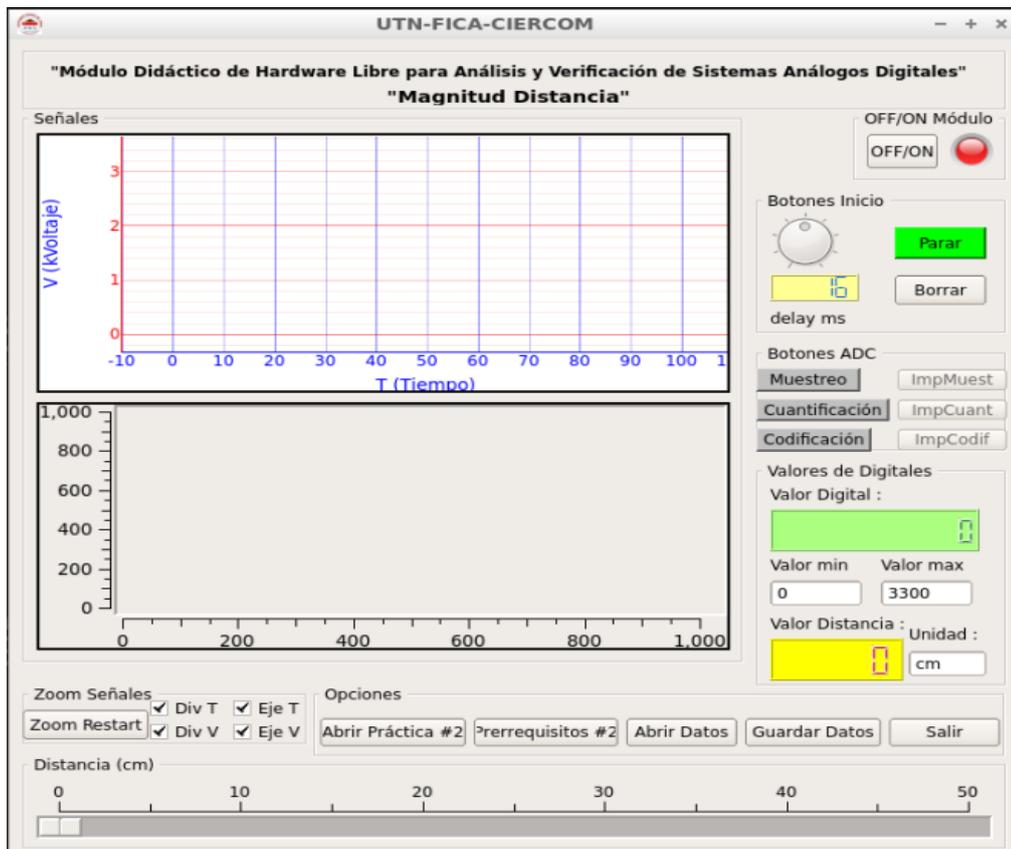


Figura 117 Interfaz gráfica de usuario de distancia.

Fuente: El Autor.

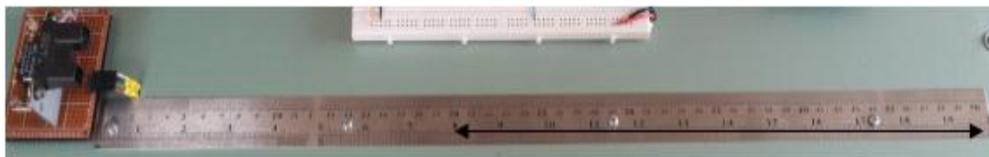
Al accionar el pulsador de INICIO en la interfaz gráfica de la magnitud distancia se activara la graficación de los datos adquiridos por sensor de distancia.



*Figura 118 Botón OFF/ON, módulo de distancia.*

Fuente: El Autor.

Cave recalcar que el rango de operación del sensor de distancia está limitado de 20 cm a 150 cm, pero en el módulo didáctico para una demostración práctica se ha colocado una regleta y de igual manera el rango está limitado de 20 cm a 50cm.



*Figura 119 Rango de medición de módulo de distancia.*

Fuente: El Autor.

Como se puede observar la cota de medición, se la utiliza para dar una referencia de la aproximación que el sensor está realizando y por ende la conversión analógica digital de dicha señal.

Presionando cada uno de los botones del proceso de ADC se podrá evidenciar los procesos por los cuales la señal debe pasar para convertirse en digital y representar el valor de la distancia del medio en el indicador digital.

Presionando el botón del proceso de ADC correspondiente a cuantificación se podrá evidenciar el método de cuantificar la señal proveniente de la entrada analógica A2 del pduino, para más información de este proceso diríjase a la sección 3.5.4. del trabajo de grado. El proceso de graficación de la señal cuantificada se podrá observar en el qwtplot, en la gráfica se puede observar dicho procedimiento.

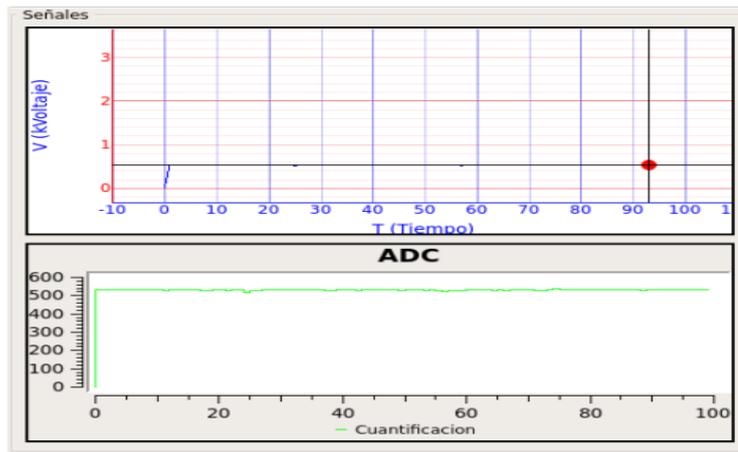


Figura 120 Proceso de cuantificación, módulo de distancia.

Fuente: El Autor.

Para verificar que los valores obtenidos en las distintas gráficas y LCDWidget son correctas, se recomienda que el estudiante realice mediciones en intervalos de 5 minutos y compare estos valores con los valores que el datasheet proporciona.

En la sección opciones, si se presiona abrir practica #2 nos redirigimos a la interfaz gráfica, la misma que el estudiante deberá completar, generar el pdf y enviar al correo electrónico del docente encargado de la asignatura.

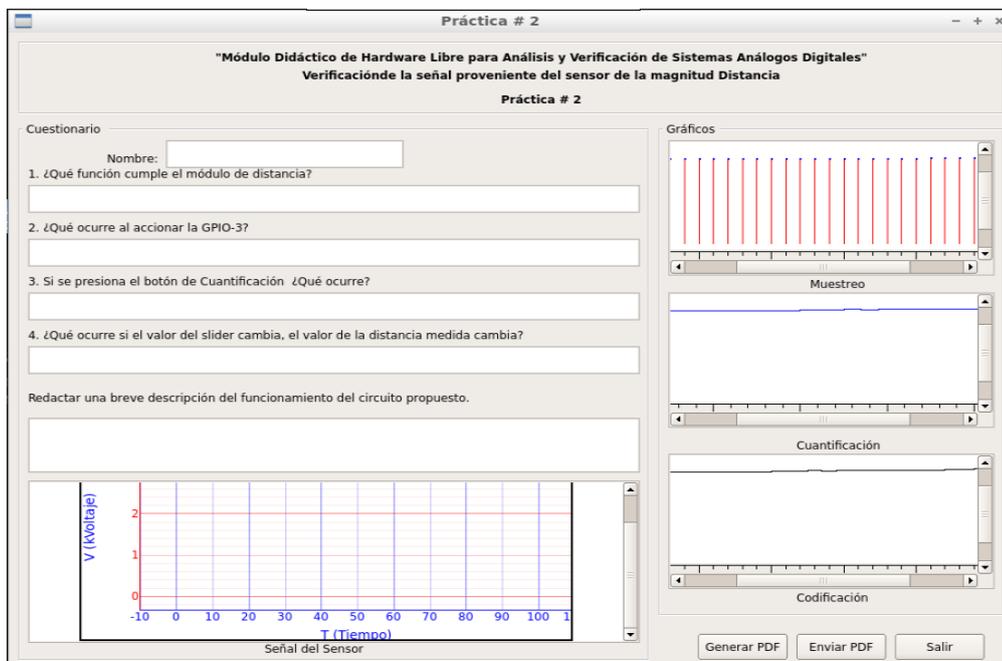


Figura 121 Práctica #2, módulo de distancia.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a la distancia:

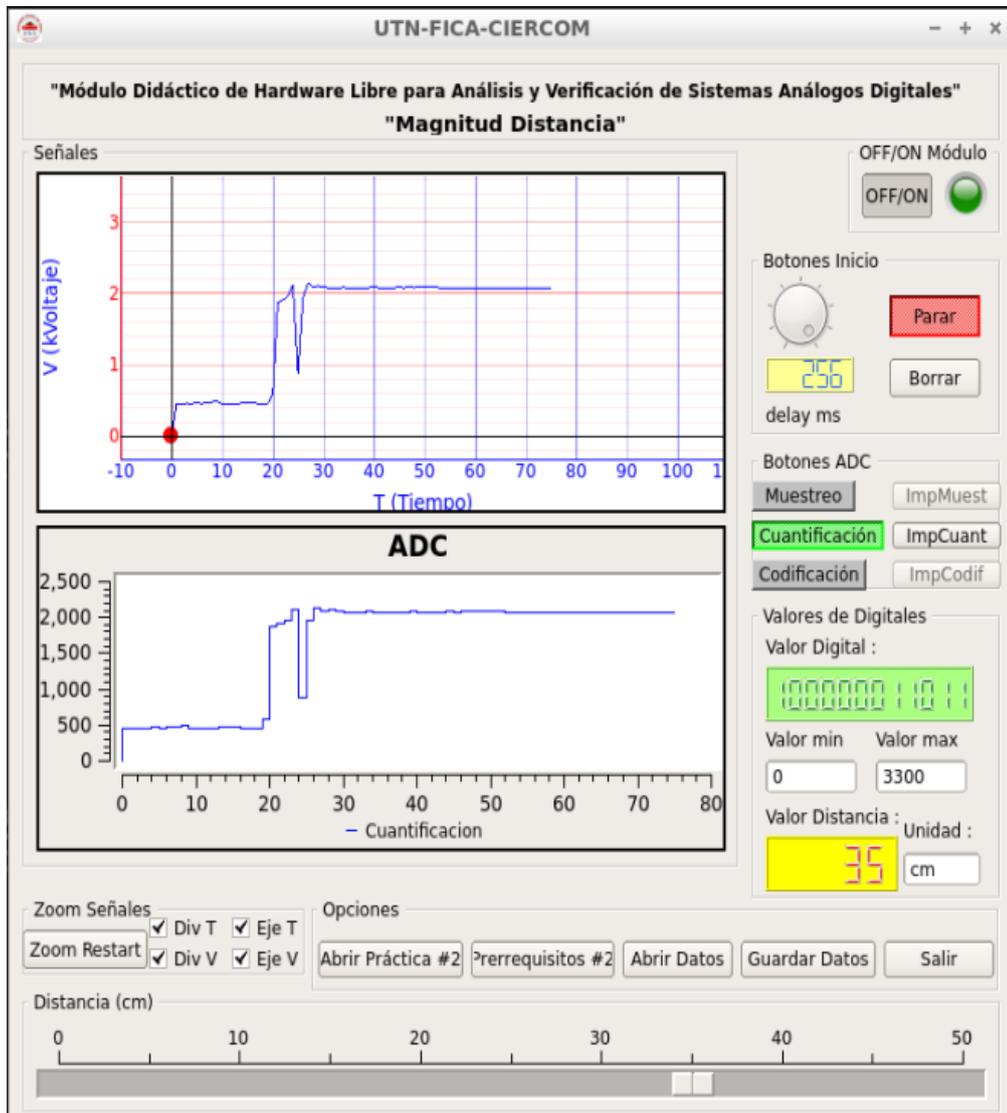


Figura 122 Interfaz gráfica de distancia en ejecución.

Fuente: El Autor.

#### 4.3.5 CUESTIONARIO

- ¿Qué función cumple el módulo de distancia?
- ¿Qué ocurre al accionar la GPIO-3?

c. Si se presiona el botón de Cuantificación ¿Qué ocurre?

d. ¿Qué ocurre si el valor del slider, si el valor de la distancia cambia?

#### 4.3.6 FUNCIONAMIENTO Y CONCLUSIONES

Redactar una breve descripción del funcionamiento del circuito propuesto.

.....  
.....  
.....  
.....  
.....  
.....  
.....

#### 4.3.7 ACTIVIDADES PROPUESTAS

- Realizar la simulación del circuito del módulo de distancia mediante la utilización de algún software (Proteus, Fridzing), y verificar el correcto funcionamiento del mismo.
- Implementar el circuito del módulo de temperatura en un protoboard, conectarlo a la entrada A2 y comprobar el funcionamiento.
- Investigue una función matemática diferente a la propuesta, que exprese la relación existente entre la señal censada y la distancia medida.
- Investigue otros sensores de distancia y comparta con sus compañeros sus experiencias.
- Cree un scripts que cense la distancia y que cada quince centímetros medidos se active una alarma.

- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.

**PRECAUCIÓN:**

**Revise las conexiones entre el módulo de distancia antes de conectarlo al pduino, recuerde que el voltaje máximo de una entrada analógica de 12 Bits del pduino es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

#### 4.4 PRÁCTICA # 3: VERIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE LA MAGNITUD POSICIÓN.

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. Electrónica y Redes de Comunicación			

PRÁCTICA No.	LABORATORIO DE:	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
3	Electrónica	Verificación de la señal proveniente del sensor de la magnitud posición.	1

##### 4.4.1 OBJETIVO

El alumno aprenderá a utilizar el módulo de posición como aplicación específica de adquisición de magnitud física posición, a través de un accionamiento directo por medio de la una GPIO 2 cuyo valor lógico activa o desactiva dicho módulo. Además el estudiante al culminar esta práctica será capaz de describir el proceso de codificación que demanda la conversión analógica digital.

##### 4.4.2 EQUIPO REQUERIDO

- 1 Pcdduino
- 1 Módulo de sensor de magnitud posición
- 1 Juego de cables
- Bloque de borneras #1

### **4.4.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe tener en claro los términos de: Pc-Duino, sensor de posición, potenciómetro, codificación y responder las interrogantes que a continuación se listan:

¿Qué es un Pc-Duino?

¿Qué es un potenciómetro?

¿Qué es la codificación?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Pc-Duino, módulo de posición, potenciómetro, codificación.

### **4.4.4 PROCEDIMIENTO**

Para identificar las partes de la interfaz gráfica de usuario diríjase a la sección 3.5.1.1 del trabajo de grado.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de distancia, para el accionamiento del mismo conecte el pin de D-IN a la GPIO-2 del pcdiino. La salida del módulo de distancia debe conectarse a la entrada analógica A4 del pcdiino. Verificar las conexiones del módulo de distancia con el bloque de borneras # 1 con la función continuidad del multímetro.

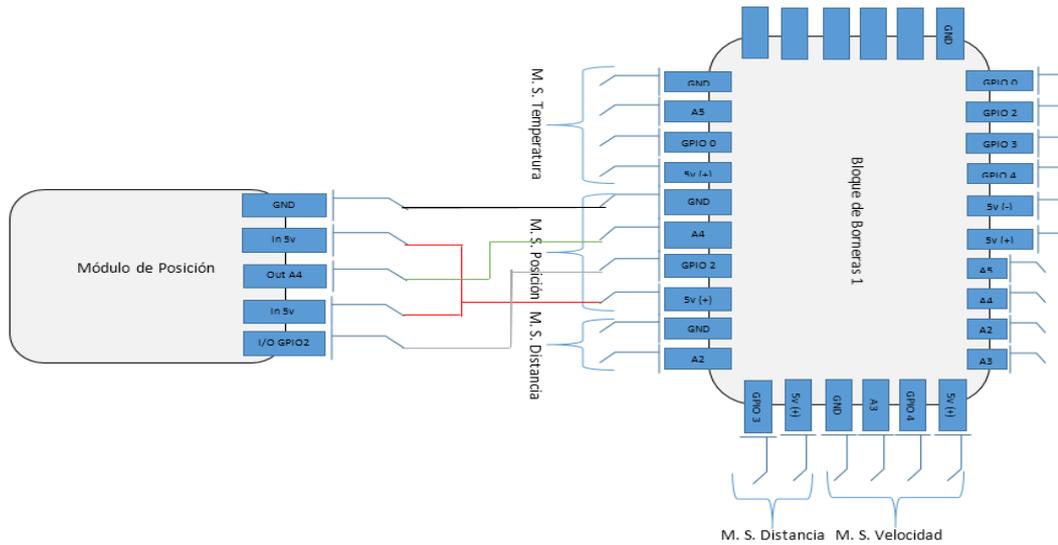


Figura 123 Diagrama para verificación de conexión modulo posición.

Fuente: EL Autor.

La interfaz gráfica de usuario de posición es la siguiente:

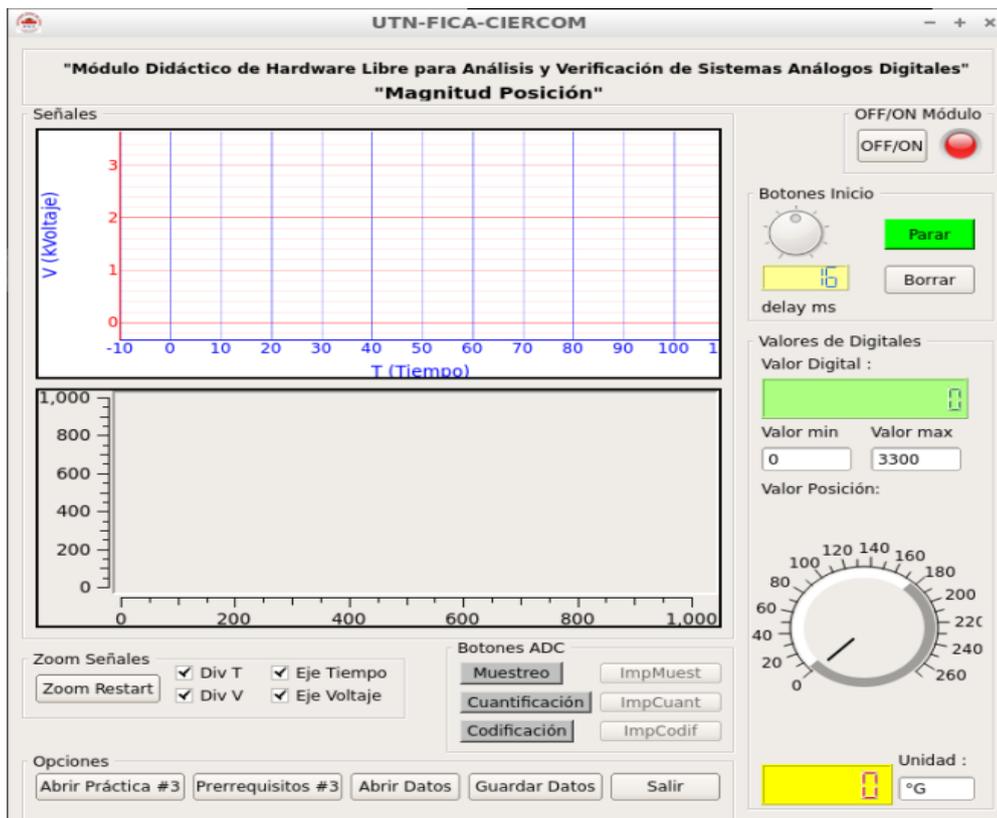


Figura 124 Interfaz gráfica de usuario de posición.

Fuente: El Autor.

Al accionar el pulsador de INICIO en la interfaz gráfica de la magnitud posición se activara la graficación de los datos adquiridos por sensor de posición.



Figura 125 Botón OFF/ON, módulo de posición.

Fuente: El Autor.

Presionando cada uno de los botones del proceso de ADC se podrá evidenciar los procesos por los cuales la señal debe pasar para convertirse en digital y representar el valor de la posición del medio en el indicador digital.

Presionando el botón del proceso de ADC correspondiente a codificación se podrá evidenciar el método de codificar de la señal proveniente de la entrada analógica A4 del pduino, para más información de este proceso diríjase a la sección 3.5.5. del trabajo de grado. El proceso de graficación de la señal codificada se podrá observar en el qwtplot haciendo referencia a los valores de los ejes, en la gráfica se puede observar dicho procedimiento, también se puede evidenciar en la sección de valores digitales.

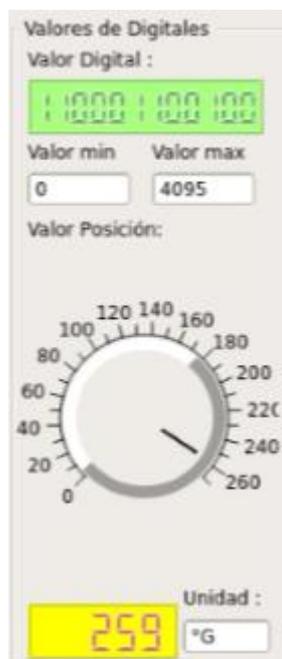


Figura 126 Proceso de codificación, módulo de posición.

Fuente: El Autor.

En la sección opciones, si se presiona abrir practica #3 nos redirigimos a la interfaz gráfica, la misma que el estudiante deberá completar, generar el pdf y enviar al correo electrónico del docente encargado de la asignatura.

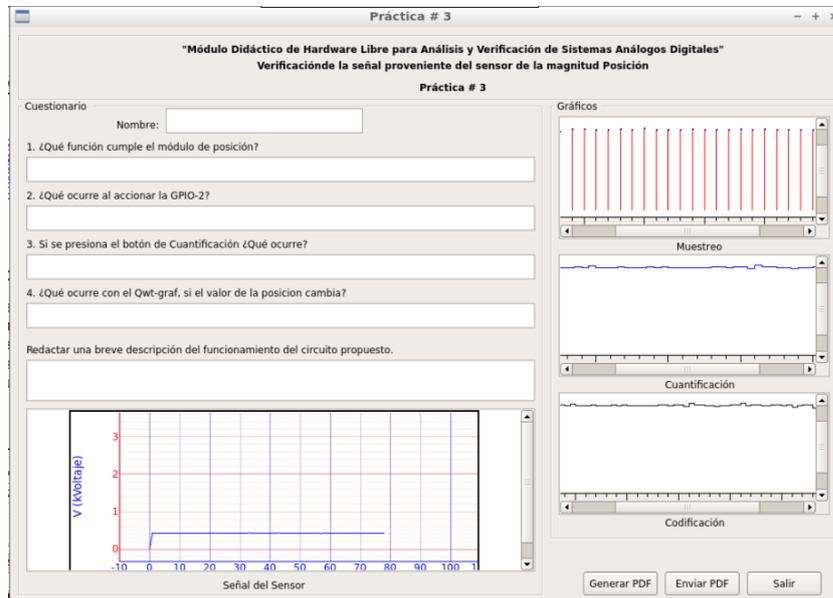


Figura 127 Práctica #3, módulo de posición.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a la posición:

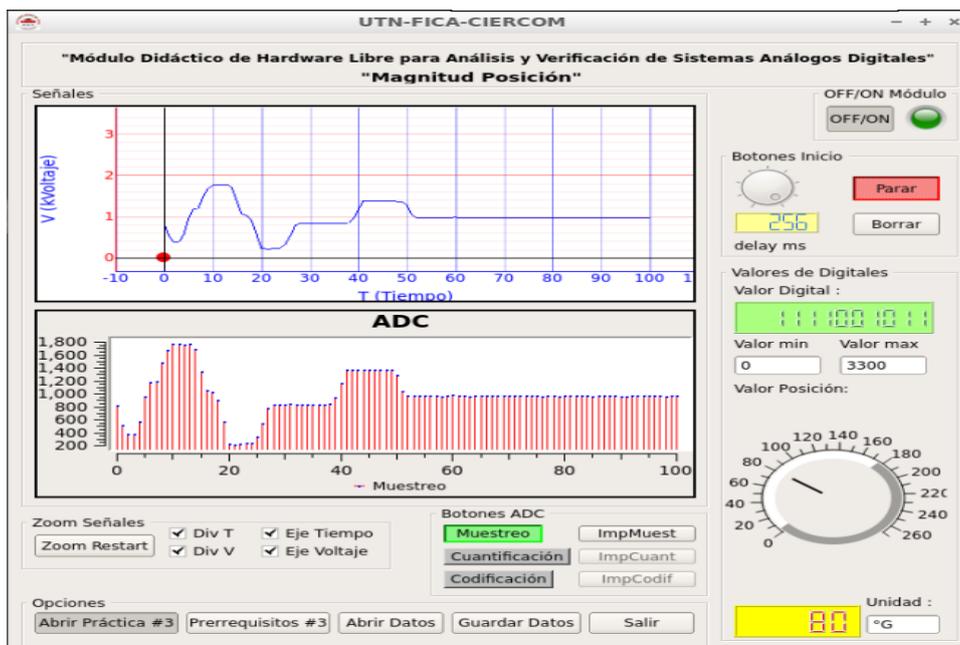


Figura 128 Interfaz gráfica de posición en ejecución.

Fuente: El Autor.

#### 4.4.5 CUESTIONARIO

- a. ¿Qué función cumple el módulo de posición?
  
- b. ¿Qué ocurre al accionar la GPIO-2?
  
- c. Si se presiona el botón de Codificación ¿Qué ocurre?
  
- d. ¿Qué ocurre si con el Qwt-graf, si la medida de la posición cambia?

#### 4.4.6 FUNCIONAMIENTO Y CONCLUSIONES

Redactar una breve descripción del funcionamiento del circuito propuesto.

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

#### 4.4.7 ACTIVIDADES PROPUESTAS

- Realizar la simulación del circuito del módulo de posición mediante la utilización de algún software (Proteus, Fridzing), y verificar el correcto funcionamiento del mismo.

- Implementar el circuito del módulo de posición en un protoboard, conectarlo a la entrada A4 y comprobar el funcionamiento.
- Cree un scripts que cense la posición y cada treinta grados en el indicador cambie de color.
- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.

**PRECAUCIÓN:**

**Revise las conexiones entre el módulo de posición antes de conectarlo al pduino, recuerde que el voltaje máximo de una entrada analógica de 12 Bits del pduino es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

## **4.5 PRÁCTICA # 4: VERIFICACIÓN DE LA SEÑAL PROVENIENTE DEL SENSOR DE LA MAGNITUD VELOCIDAD ANGULAR.**

<b>CARRERA</b>	<b>PLAN DE ESTUDIO</b>	<b>CLAVE ASIGNATURA</b>	<b>NOMBRE DE LA ASIGNATURA</b>
Ing. Electrónica y Redes de Comunicación			

<b>PRÁCTICA No.</b>	<b>LABORATORIO DE:</b>	<b>NOMBRE DE LA PRÁCTICA</b>	<b>DURACIÓN (HORAS)</b>
<b>4</b>	Electrónica	Verificación de la señal proveniente del sensor de la magnitud velocidad angular.	1

### **4.5.1 OBJETIVO**

El alumno aprenderá a utilizar el módulo de velocidad como aplicación específica de adquisición de magnitud física velocidad, a través de un accionamiento directo por medio de la una GPIO-4 cuyo valor lógico activa o desactiva dicho módulo. Además el estudiante al culminar esta práctica será capaz de describir el proceso de muestreo, cuantificación y codificación que demanda la conversión analógica digital.

### **4.5.2 EQUIPO REQUERIDO**

- 1 Pcdduino
- 1 Módulo de sensor de velocidad angular
- 1 Juego de cables
- Bloque de borneras #1

### **4.5.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe tener en claro los términos de: Pc-Duino, sensor de velocidad, ventilador, muestreo, cuantificación, codificación, tacómetro y responder las interrogantes que a continuación se listan:

¿Qué es un Pc-Duino?

¿Qué es un ventilador?

Describa las partes de un ventilador de cuatros pines.

¿Qué es un tacómetro?

¿Cómo funciona un sensor de efecto hall, haciendo referencia al sensor existente en el ventilador?

¿Qué es el muestreo, cuantificación y codificación?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Pc-Duino, módulo de velocidad, ventilador, muestreo, cuantificación, codificación, tacómetro.

### **4.5.4 PROCEDIMIENTO**

Para identificar las partes de la interfaz gráfica de usuario diríjase a la sección 3.5.1.1 del trabajo de grado.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de distancia, para el accionamiento del mismo conecte el pin de D-IN a la GPIO-4 del pcdduino. La salida del módulo de distancia debe conectarse a la entrada analógica A3 del pcdduino. Verificar las conexiones del módulo de distancia con el bloque de borneras # 1 con la función continuidad del multímetro.

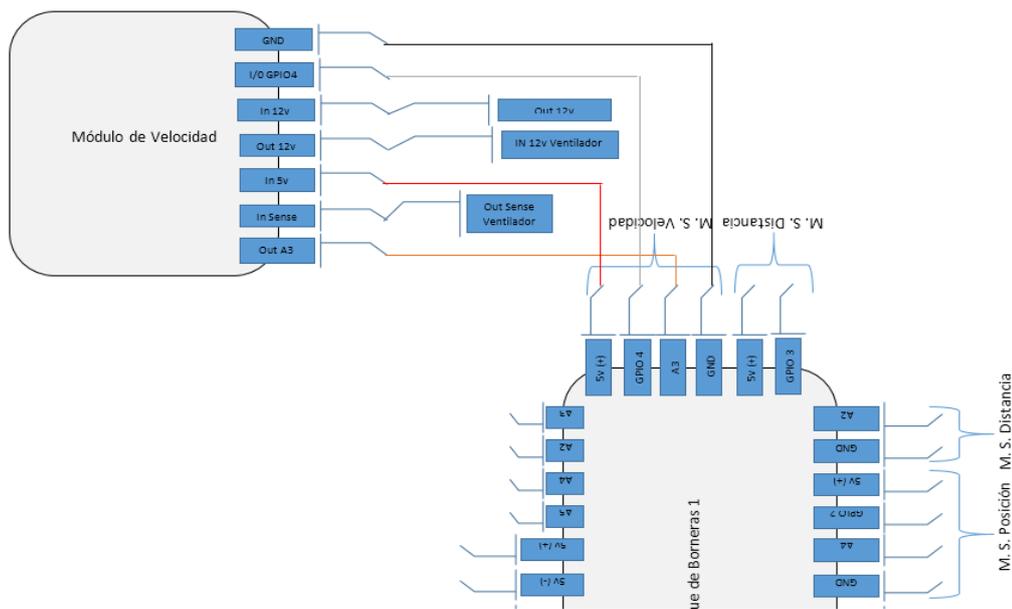


Figura 129 Diagrama para verificación de conexión modulo velocidad.

Fuente: EL Autor.

La interfaz gráfica de usuario de posición es la siguiente:

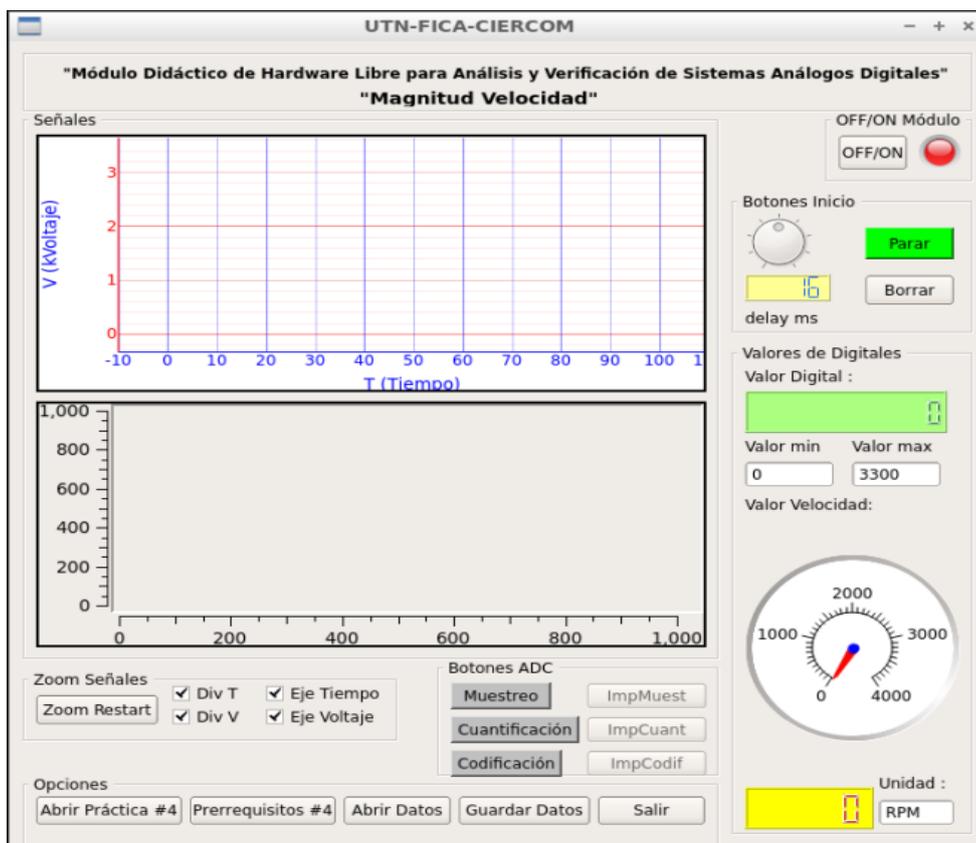


Figura 130 Interfaz gráfica de usuario de velocidad.

Fuente: El Autor.

Al accionar el pulsador de INICIO en la interfaz gráfica de la magnitud velocidad se activara la graficación de los datos adquiridos por sensor de velocidad.



Figura 131 Botón OFF/ON, módulo de velocidad.

Fuente: El Autor.

Presionando cada uno de los botones del proceso de ADC se podrá evidenciar los procesos por los cuales la señal debe pasar para convertirse en digital y representar el valor de la velocidad del medio en el indicador digital.

Presionando los botones del proceso de ADC, se podrá evidenciar los métodos de ADC de la señal proveniente de la entrada analógica A3 del pcdduino, para más información de estos procesos diríjase a la sección 3.5.(3-5). del trabajo de grado.

En la sección opciones, si se presiona abrir practica #4 nos redirigimos a la interfaz gráfica, la misma que el estudiante deberá completar, generar el pdf y enviar al correo electrónico del docente encargado de la asignatura.

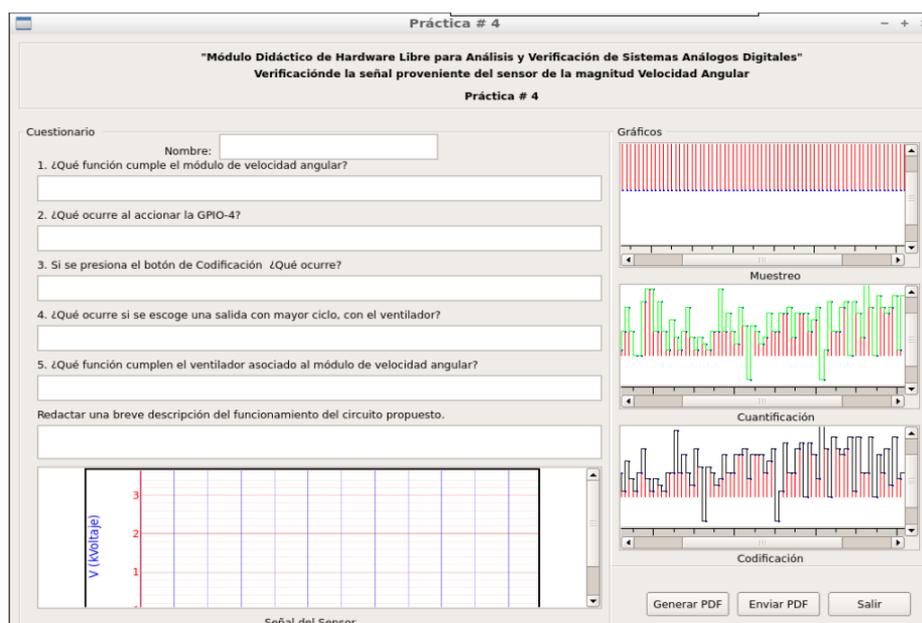


Figura 132 Práctica #4, módulo de velocidad.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a la posición:

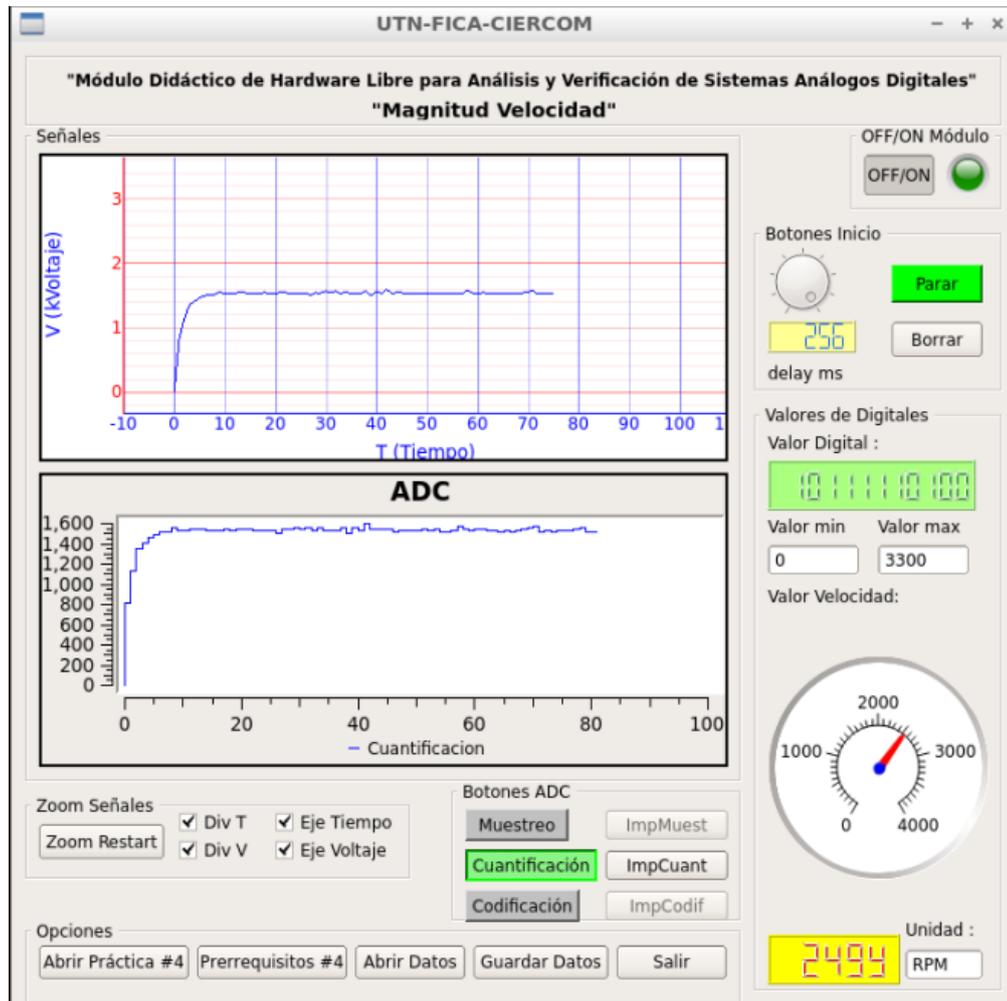


Figura 133 Interfaz gráfica de velocidad en ejecución.

Fuente: El Autor.

#### 4.5.5 CUESTIONARIO

- ¿Qué función cumple el módulo de velocidad angular?
- ¿Qué ocurre al accionar la GPIO-4?

- c. Si se presiona el botón de Codificación ¿Qué ocurre?
  
- d. ¿Qué ocurre si se escoge una salida con mayor ciclo, con el ventilador?
  
- e. ¿Qué función cumplen el ventilador asociado al módulo de velocidad angular?

#### 4.5.6 FUNCIONAMIENTO Y CONCLUSIONES

Redactar una breve descripción del funcionamiento del circuito propuesto.

.....

.....

.....

.....

.....

.....

.....

.....

#### 4.5.7 ACTIVIDADES PROPUESTAS

- Realizar la simulación del circuito del módulo de velocidad angular mediante la utilización de algún software (Proteus, Fridzing), y verificar el correcto funcionamiento del mismo.
- Implementar el circuito del módulo de velocidad angular en un protoboard, conectarlo a la entrada A3 y comprobar el funcionamiento.
- Cree un scripts que cense la velocidad angular y cada vez que cambie se encienda un led que empiece a parpadear según la velocidad angular.

- Consiga un ventilador de mayor RPM y cree un scripts que cense dicha velocidad angular.
- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.
- Redacte una breve descripción sobre el funcionamiento del sensor de efecto hall que se encuentra en el ventilador.

**PRECAUCIÓN:**

**Revise las conexiones entre el módulo de velocidad antes de conectarlo al pduino, recuerde que el voltaje máximo de una entrada analógica de 12 Bits del pduino es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

## 4.6 PRÁCTICA # 5: VERIFICACIÓN DEL FUNCIONAMIENTO DE CONVERSIÓN DIGITAL ANALÓGICA.

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. Electrónica y Redes de Comunicación			

PRÁCTICA No.	LABORATORIO DE:	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
5	Electrónica	Verificación del funcionamiento de conversión digital analógica.	1

### 4.6.1 OBJETIVO

El alumno aprenderá a utilizar los módulos R2R, PWM y Matriz de Leds como aplicaciones específicas de conversión digital a analógica, a través de un accionamiento directo de las GPIO-(PG1-7) del cubietruck cuyo valor lógico activa o desactiva los bits de R2R. Además el estudiante al culminar esta práctica será capaz de describir el proceso de conversión digital analógica, e interactuara con aplicaciones de DAC como lo son PWM y Matriz de Leds.

### 4.6.2 EQUIPO REQUERIDO

- 1 Cubietruck
- 1 Módulo de sensor de velocidad angular
- 1 Módulo R2R

- 1 Módulo de PWM
- 1 Juego de cables
- Bloque de borneras #2
- Matriz de Leds

### **4.6.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe tener en claro los términos de: Cubietruck, sensor de velocidad, ventilador, R2R, PWM, Matriz de Leds y responder las interrogantes que a continuación se listan:

¿Qué es un Cubietruck?

¿Qué es un ventilador?

¿Qué es R2R, PWM, Matriz de Leds?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Cubietruck, módulo de velocidad, ventilador, R2R, PWM, Matriz de Leds, PG-(1-7).

### **4.6.4 PROCEDIMIENTO**

Para identificar las partes de la interfaz gráfica de usuario diríjase a la sección 3.5.1.1 del trabajo de grado.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de R2R, para el accionamiento del módulo R2R conecte los pin de GP(1-7) del cubietruck a los pines del bloque de borneras #2 correspondientes a PG(1-7). Verificar las conexiones del módulo R2R con el bloque de borneras # 2 con la función continuidad del multímetro.

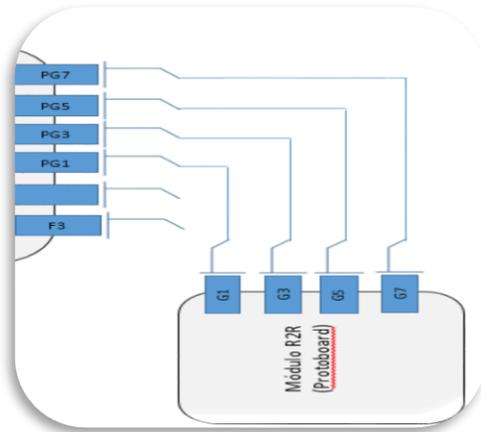


Figura 134 Diagrama para verificación de conexión módulo R2R.

Fuente: EL Autor.

La interfaz gráfica de usuario de R2R es la siguiente:



Figura 135 Interfaz gráfica de usuario de R2R.

Fuente: El Autor.

Al accionar cada uno de los pulsadores de los bits en la interfaz gráfica de R2R se activará la graficación de la combinación de datos para la conversión digital analógica, se representa también el valor binario, decimal y el valor del voltaje de la combinación de bits en indicadores digitales. Para más información de estos procesos diríjase a la sección 3.10. del trabajo de grado.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a R2R:

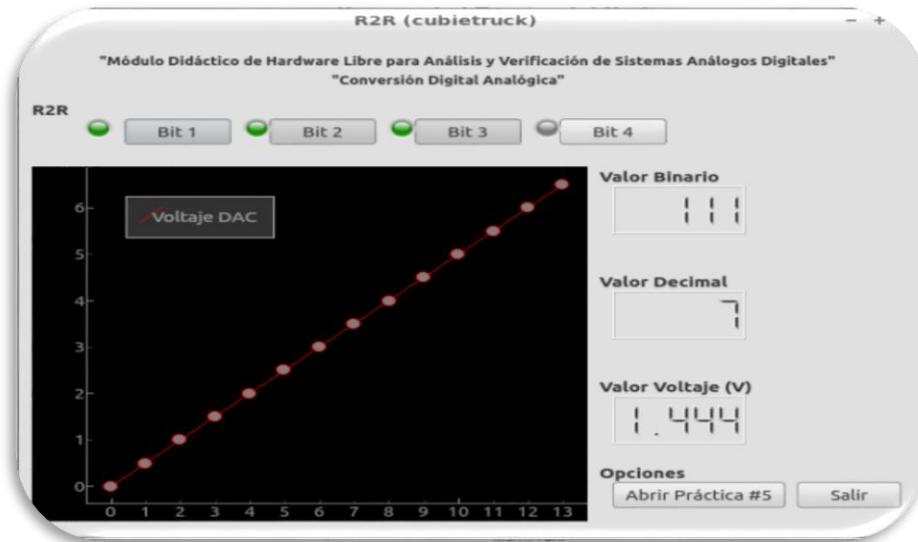


Figura 136 Interfaz gráfica de usuario de R2R en funcionamiento.

Fuente: El Autor.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de PWM, para el accionamiento del módulo PWM conecte el pin de PB(1) del cubietruck al pin del bloque de borneras #2 correspondientes a PB(1). Verificar las conexiones del módulo PWM con el bloque de borneras # 2 con la función continuidad del multímetro.

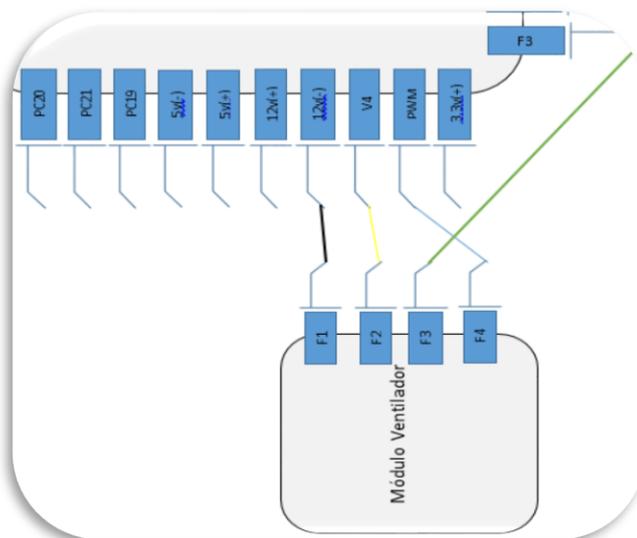


Figura 137 Diagrama para verificación de conexión módulo ventilador.

Fuente: EL Autor.

La interfaz gráfica de usuario de PWM es la siguiente:

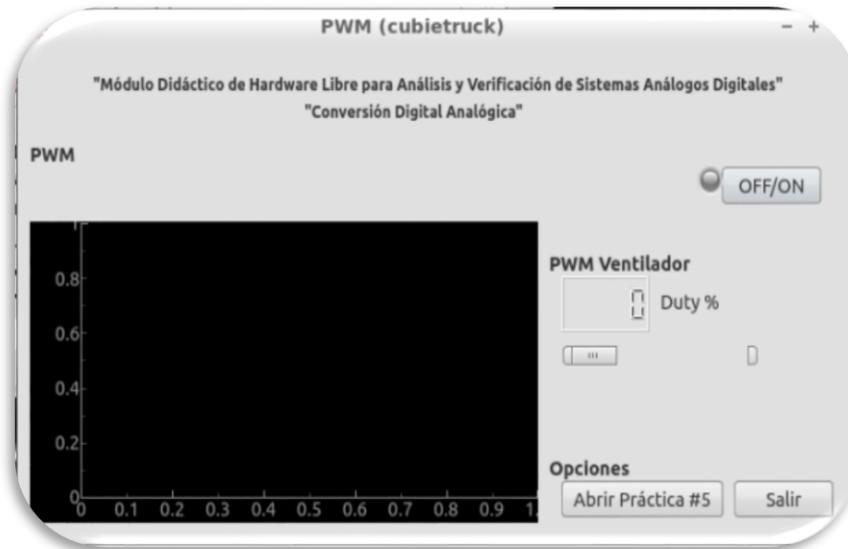


Figura 138 Interfaz gráfica de usuario de PWM.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a PWM:

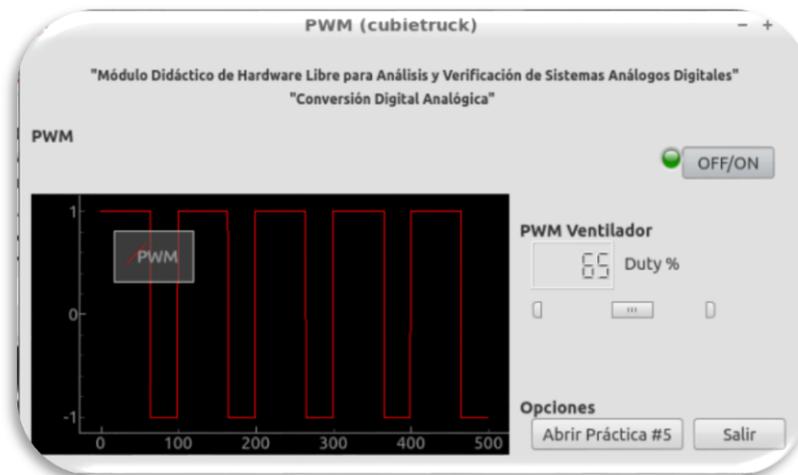


Figura 139 Interfaz gráfica de usuario de PWM en funcionamiento.

Fuente: El Autor.

La figura 4-(1-2) representan el esquema de distribución de elementos del módulo de Matriz de Leds, para el accionamiento del módulo Matriz de Leds conecte los pines de CLK, CS, DIN, GND, 5(+) Vcc del cubietruck a los pines del bloque de borneras #2 correspondientes a L(1-5). Verificar las conexiones del

módulo Matriz de Leds con el bloque de borneras # 2 con la función continuidad del multímetro.

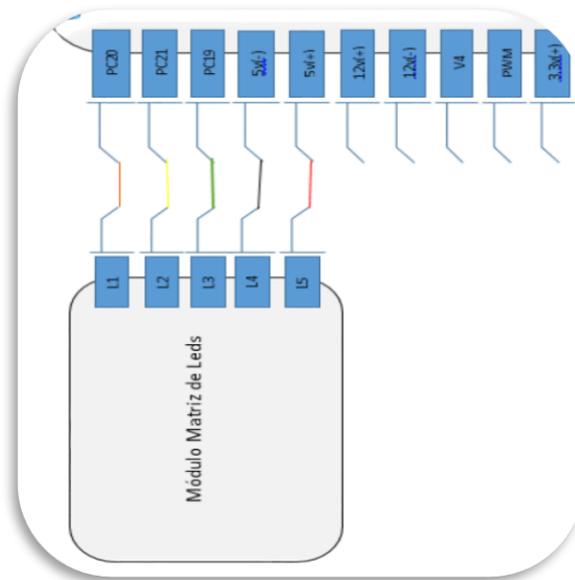


Figura 140 Diagrama para verificación de conexión módulo Matriz de Leds.

Fuente: EL Autor.

La interfaz gráfica de usuario de Matriz de Leds es la siguiente:



Figura 141 Interfaz gráfica de usuario de Matriz de Leds.

Fuente: El Autor.

En la siguiente figura se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a Matriz de Leds:



Figura 142 Interfaz gráfica de usuario de Matriz de Leds en funcionamiento.

Fuente: El Autor.

En la sección opciones de cada una de las interfaces graficas antes descritas, si se presiona abrir practica #5 nos redirigimos a la interfaz gráfica, la misma que el estudiante deberá completar, generar el pdf y enviar al correo electrónico del docente encargado de la asignatura.

Figura 143 Interfaz gráfica de usuario de práctica # 5.

Fuente: El Autor.

#### 4.6.5 CUESTIONARIO

- a. ¿Qué función cumple el módulo R2R??
  
- b. ¿Qué ocurre al accionar la PG-(1-7) del Cubietruck?
  
- c. Si se cambia el valor del ciclo útil (PWM) ¿Qué ocurre?
  
- d. ¿Qué ocurre si en R2R se aumenta dos bits, es mayor la resolución del DAC?
  
- e. ¿Qué función cumplen PWM asociado al módulo de velocidad?

#### 4.6.6 FUNCIONAMIENTO Y CONCLUSIONES

Redactar una breve descripción del funcionamiento de R2R, PWM y Matriz de Leds.

.....  
.....  
.....  
.....  
.....  
.....  
.....

#### 4.6.7 ACTIVIDADES PROPUESTAS

- Realizar la simulación del circuito del módulo de R2R mediante la utilización de algún software (Proteus, Fridzing), y verificar el correcto funcionamiento del mismo.
- Implementar el circuito del módulo de R2R de 5 bits en un protoboard, conectarlo y comprobar el funcionamiento.
- Consulte configuraciones diferentes de DAC, realice una breve descripción de los mismos e impleméntelos en el protoboard del módulo didáctico.
- Consiga un ventilador de mayor RPM y cree un scripts que varíe con PWM dicha velocidad angular.
- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.

**PRECAUCIÓN:**

**Revise las conexiones entre el módulo de R2R, PWM y Matriz de Leds antes de conectarlo al Cubietruck, recuerde que el voltaje máximo de una GPIO del Cubietruck es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

## 4.7 PRÁCTICA # 6: VERIFICACIÓN DEL FUNCIONAMIENTO DEL MÓDULO DIDÁCTICO EN SU TOTALIDAD.

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
Ing. Electrónica y Redes de Comunicación			

PRÁCTICA No.	LABORATORIO DE:	NOMBRE DE LA PRÁCTICA	DURACIÓN (HORAS)
6	Electrónica	Verificación del funcionamiento del módulo didáctico en su totalidad.	1

### 4.7.1 OBJETIVO

El alumno verificara los módulos de temperatura, distancia, posición, velocidad angular, R2R, PWM y Matriz de Leds como aplicaciones específicas de conversión análoga digital y digital a analógica, a través de los accionamientos directos de las GPIO-(0-4-Pcduino)-(PG1-7-cubietruck) cuyo valor lógico activa o desactiva los módulos correspondientes a cada uno de los sensores y aplicaciones DAC. Además el estudiante al culminar esta práctica será capaz de describir el proceso de conversión análoga digital y digital analógica, e interactuara con todas las aplicaciones del módulo didáctico.

### 4.7.2 EQUIPO REQUERIDO

- 1 Pduino
- 1 Módulo de sensor de temperatura

- 1 Módulo de sensor de distancia
- 1 Módulo de sensor de posición
- 1 Módulo de sensor de velocidad angular
- 1 Juego de cables
- Bloque de borneras #1
- 1 Cubietruck
- 1 Módulo de sensor de velocidad angular
- 1 Módulo R2R
- 1 Módulo de PWM
- 1 Juego de cables
- Bloque de borneras #2
- Matriz de Leds

#### **4.7.3 PRERREQUISITOS DE LA PRÁCTICA**

El estudiante antes de realizar la práctica debe tener en claro los términos de: PcDuino, sensor de temperatura, sensor de distancia, sensor de posición, sensor de velocidad, Cubietruck, ventilador, R2R, PWM, Matriz de Leds y responder las interrogantes que a continuación se listan:

¿Qué es un Pc-Duino?

¿Qué es una entrada analógica?

¿Qué es el LM35?

¿Qué es una GPIO?

¿Qué es el Sharp GP2Y0A02YK0F?

¿Qué es un potenciómetro?

¿Qué es el muestreo, cuantificación y codificación?

¿Qué es un Cubietruck?

¿Qué es un ventilador?

¿Qué es un tacómetro?

¿Qué es R2R, PWM, Matriz de Leds?

Todos estos términos y definiciones se encuentran detallados en los capítulos del trabajo de grado, el estudiante también puede utilizar diferentes referencias bibliográficas para consultar los mismos.

Palabras clave: Pc-Duino, Cubietruck, módulo de didáctico, sensor de temperatura, sensor de distancia, sensor de posición, sensor de velocidad, ventilador, R2R, PWM, Matriz de Leds.

#### 4.7.4 PROCEDIMIENTO

Para identificar las partes de la interfaz gráfica de usuario diríjase al manual de usuario del trabajo de grado. Se representa el esquema de distribución de los elementos del módulo didáctico, para los accionamientos de los distintos módulos conecte los pin de GPIO correspondientes a cada uno de los mismos a los pines del bloque de borneras #1 y #2 correspondientes a GPIO. Verificar todas las conexiones de los módulos con los bloques de borneras #1 y #2 con la función continuidad del multímetro.

La interfaz gráfica de usuario de la práctica general se puede evidenciar en la figura 144:

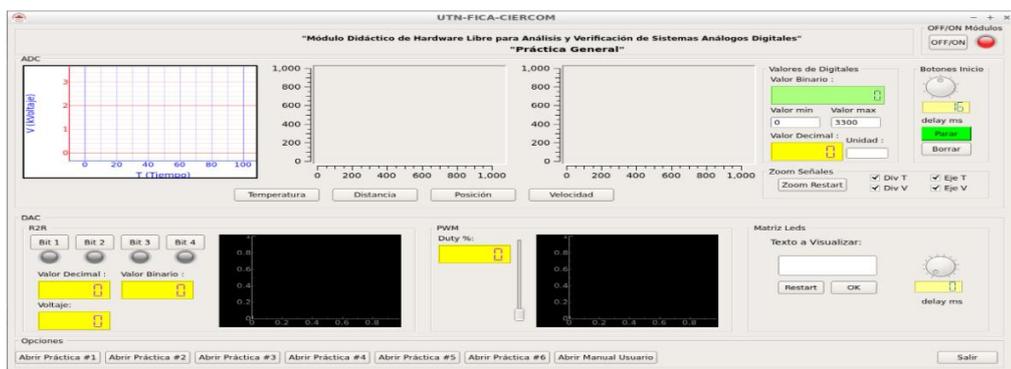


Figura 144 Interfaz gráfica de la práctica general.

Fuente: El Autor.

Al accionar el pulsador de INICIO en la interfaz gráfica de la práctica general se activaran todos los módulos de sensores, la graficación de los datos adquiridos por cada uno de los sensores.



Figura 145 Botón OFF/ON, módulos.

Fuente: El Autor.

Al accionar cada uno de los botones de los sensores y módulos de DAC en la interfaz gráfica de se activara la graficación de la combinación de datos para la conversión análoga digital y digital analógica, se representa también el valor decimal y el valor del voltaje de la señal en cada uno de los canales analógicos del pcdduino y la combinación de bits en indicadores digitales para R2R. Para más información de estos procesos dirijase a la sección manual de usuario del trabajo de grado.

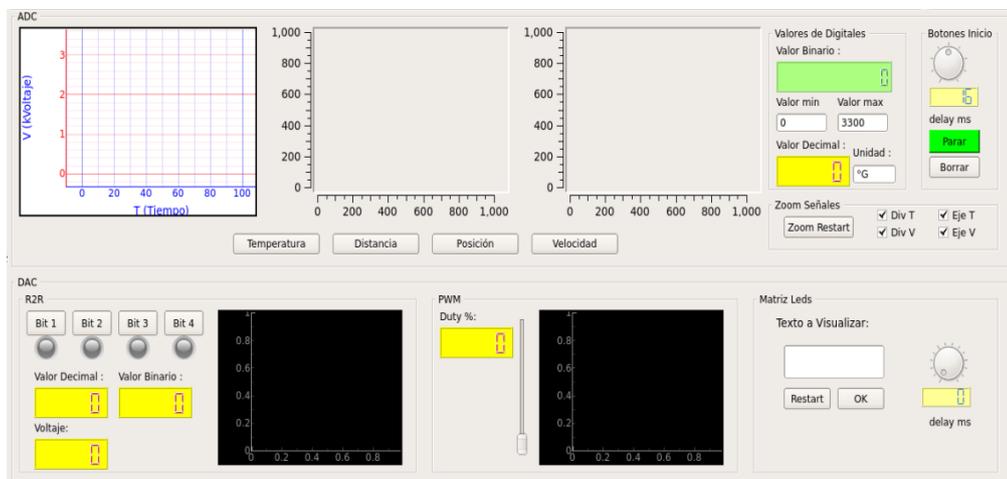


Figura 146 Botones de acceso a graficación de la práctica general.

Fuente: El Autor.

En la figura (tal) se muestra en funcionamiento de la interfaz gráfica de usuario correspondiente a la práctica general:

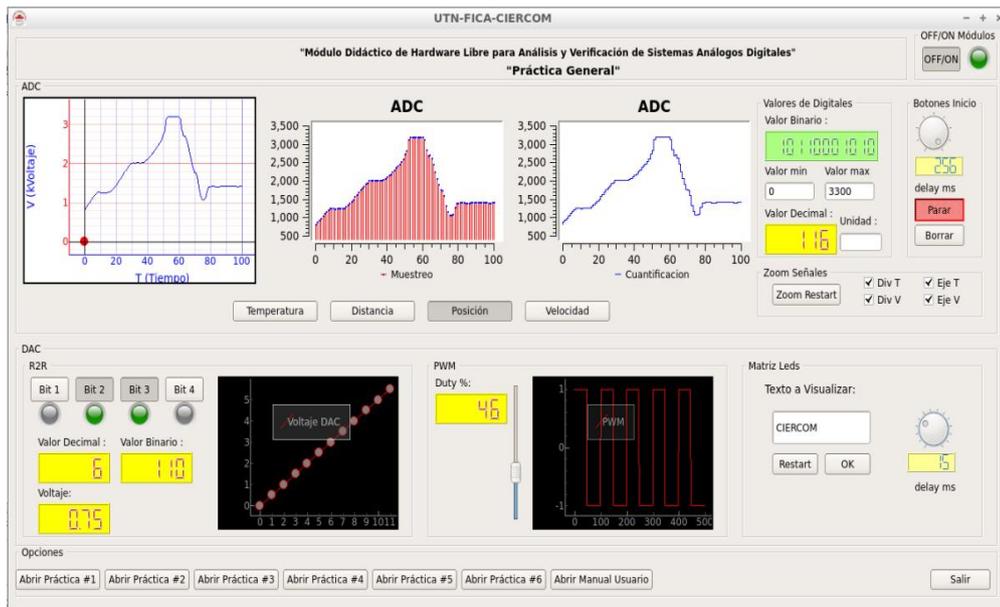


Figura 147 Interfaz gráfica de la práctica general en funcionamiento.

Fuente: El Autor.

En la sección opciones de la interfaz gráfica antes descrita, si se presiona abrir practica #6 nos redirige a la interfaz gráfica de la práctica, la misma que el estudiante deberá completar, generar el pdf y enviar al correo electrónico del docente encargado de la asignatura.

Figura 148 Práctica #6, practica general del módulo didáctico.

Fuente: El Autor.

#### 4.7.5 CUESTIONARIO

- a. ¿Qué función cumple el módulo Pc-Duino?
  
- b. ¿Qué función cumple el módulo Cubietruck?
  
- c. ¿Qué es el muestreo, cuantificación y codificación?
  
- d. ¿Qué son los módulos: temperatura, distancia, posición y velocidad?
  
- e. ¿Qué son los módulos: R2R, PWM, Matriz de Leds?

#### 4.7.6 FUNCIONAMIENTO Y CONCLUSIONES

Redactar una breve descripción del funcionamiento del módulo didáctico.

.....

.....

.....

.....

.....

.....

.....

.....

#### 4.7.7 ACTIVIDADES PROPUESTAS

- Realizar un resumen del funcionamiento del módulo didáctico y exponer en no más de diez minutos a sus compañeros.
- Realizar una exposición con las conclusiones e impactos que se han generado al realizar las prácticas en el módulo didáctico.
- Consulte sobre las aplicaciones de los módulos de hardware libre y exponga a sus compañeros.
- Genere un archivo PDF con las preguntas del cuestionario de esta práctica y envíe el documento al correo electrónico del docente encargado.

**PRECAUCIÓN:**

**Revise las conexiones del módulo didáctico antes de conectarlo al Pc-Duino y Cubietruck, recuerde que el voltaje máximo de una GPIO de los módulos de hardware libre es de 3,3V. Evite realizar actividades que no correspondan.**

Elaborado por :	Revisado por:
Docente:	Coordinador:

## 5 CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- En la implementación del módulo didáctico de hardware libre para análisis y verificación de sistemas análogos digitales, se puntualiza el funcionamiento de un proceso de conversión analógica digital a escala, en este argumento el muestreo, cuantificación y codificación son procesos que se grafican mediante el uso de interfaces graficas de usuario fáciles de entender, para una óptima comprensión de los estudiantes.
- Una vez fundados los principios y fundamentos teóricos se establecen las guías necesarias para avalar un recomendable dimensionamiento de los elementos constitutivos del módulo didáctico de hardware libre para análisis y verificación de sistemas análogos digitales, así, empleando todos estos principios se concluye que el funcionamiento de este instrumento didáctico desempeña con satisfacción los requerimientos necesarios para los procesos de conversión analógica digital.
- A modo de consecuencia del dimensionamiento de los elementos electrónicos, y módulos de hardware libre (pcduino3 y cubietruck) del módulo didáctico que se detalla, se concluye que las señales producidas tanto de los sensores como de los módulos de hardware libre son relativamente bajas (3,3 voltios como máximo) debido a que solamente está enfocado a un uso didáctico. No obstante, efectúan convenientemente con las imposiciones para simular de forma práctica la conversión analógica digital.
- A modo consecuencia de la construcción de la infraestructura que contiene incorporados todos los elementos se concluye que la distribución y funcionamiento de los elementos están muy relacionadas ya que no se debe interferir directamente para no causar variaciones en las medidas de

las señales o causar un inadecuado funcionamiento de los módulos de hardware libre (pcduino3 y cubietruck).

- Una vez implementado el sistema electrónico a través de la utilización de los módulos de hardware libre (pcduino3 y cubietruck) se establece que los procesos de encender, adquirir señales de los sensores, presentar los procesos de digitalización de dichas señales y realizar las prácticas, se inspeccionan mediante la utilización del pcduino3 y el cubietruck, simplificándose con esto el uso de varios elementos electrónicos. La implementación de las diferentes interfaces graficas de usuario en el módulo ayuda a los estudiantes a visualizar parámetros trascendentales del funcionamiento del mismo, obteniendo con esto mejor entendimiento para los estudiantes que realicen las prácticas de laboratorio propuestas.
- Con la utilización del módulo didáctico se podrá profundizar los conceptos desarrollados en la asignatura de sistemas analógicos digitales en las siguientes competencias:

Competencias de la asignatura sistemas analógicos digitales.

<b>COMPETENCIAS</b>	<b>SISTESIS DE LA ASIGNATURA</b>	<b>% de Avance Materia</b>
Capacidad de conocer los diferentes dispositivos de entrada de datos y dar soluciones prácticas a problemas de ingeniería.	CIRCUITOS PARA ENTRADA Y SALIDA DE DATOS	30 – 55 %
Capacidad para convertir cualquier señal analógica a digital.	ADQUISICION DE DATOS	55 – 75 %
Capacidad para realizar el diseño e implementación de circuitos de control usando dispositivos digitales.	LAZO ASEGURADO DE FASE (PLL)	75 – 100 %

- Con lo antes descrito se puede expresar que con la utilización del módulo didáctico se aborda un porcentaje alto del contenido de la asignatura, se puede extender aún más el porcentaje si el estudiante realiza las actividades sugeridas en cada una de las prácticas del capítulo cuatro.
- El modulo didáctico implementado se podría utilizar en asignaturas del mismo semestre que se articulan con los temas en desarrollo: Sistemas de comunicación analógica; Sistemas Microprocesados.

## RECOMENDACIONES

- Para un apropiado uso del módulo que se exhibe se recomienda registrarse al manual de prácticas descrito en el capítulo 4.
- Es necesaria la desconexión de todas las fuentes de energía eléctrica, para proceder a realizar operaciones de ajuste, montaje o desmontaje de elementos constitutivos del módulo didáctico.
- Para una adecuada medición de señales de los sensores se debe limpiar periódicamente los mismos, para liberar a estos dispositivos de impurezas que puedan interferir en su funcionamiento normal.
- Una vez inicializado los procesos, si se desea detener la secuencia normal de los mismos se los ejecuta exclusivamente a través de los botones de las interfaces gráficas de usuario destinadas para el caso.
- Cerciorarse que los dispositivos estén correctamente conectados antes de iniciar la operación del módulo didáctico.
- Considere en el diseño que el detector de distancia no este expuesto a la luz directa del sol, lámpara de tungsteno, etc; ya que tales fuentes de luz pueden influir en los valores de la medición de distancias. Para el caso de elegir un objeto reflectante que va a estar en la línea de influencia del sensor debe ser de un material o un color reflectante en general se debe elegir un color blanco, con el fin de disminuir la desviación de medición de distancia; además se recomienda ajustar el sensor en la dirección y posición de la base del módulo alineando en paralelo al centro emisor y al centro detector.
- Impedir rigurosamente actividades que no correspondan a la realización de las prácticas para evitar posibles accidentes.

- Si se desea utilizar otro lenguaje de programación para realizar las practicas o para realizar los diferentes procedimientos de conversión análoga digital se recomienda para la elección de esta herramienta tomar en cuenta los siguientes criterios:
  - Especificación de entrada/salida: ¿El lenguaje respalda varios dispositivos? ¿Qué tan fácil es ejecutar un volumen elevado de procesamientos con elementos de entrada /salida?
  - Capacidades de manejo de datos: ¿Qué tipo de datos pueden ser procesados?
  - Capacidades de almacenamiento de datos: ¿Qué clases de organización de archivos se pueden utilizar? ¿Es fácil que los conjuntos de datos almacenados puedan ser creados?
  - Ayudas para la programación ¿Qué tan fácil es su uso? ¿Cuáles son sus capacidades de documentación? ¿Cuáles son sus ayudas para la depuración o eliminación de los errores?
  - Eficiencia: ¿Es eficiente el proceso de compilación? ¿Es eficaz el código generado durante la compilación? ¿Existe apoyo o soporte para la modificación de las características del lenguaje?

# BIBLIOGRAFÍA

## Libros

- Balcells, J. (2006). *Autómatas Programables*. Barcelona: Marcombo.
- Bolton, W. (2001). *Mecatrónica*. Barcelona-España: Alfaomega.
- Cenatic. (2013). *Open Smart Cities*. Obtenido de <http://www.cenatic.es/>
- Couch, W. L. (2008). *Sistemas de comunicación digitales y analógicos*. México: PEARSON EDUCACIÓN.
- Espí, J. (2009). *Electrónica analógica*. Valencia: Pearson Prentice Hall.
- Espí, J., & Muñoz, J. (2006). *Fundamentos de electrónica analógica*. Valencia: Servei de Publicacions.
- Franco, S. (2005). *Analógicos, Diseño con Amplificadores Operacionales y Circuitos Integrados*. México: McGraw Hil.
- Garza, J. Á. (2006). *Sistemas Digitales y Electrónica Digital, prácticas de laboratorio*. México: PEARSON EDUCACIÓN.
- González, I., González, J., & Gómez-Arribas, F. (2003). España: Universidad Autónoma de Madrid.
- González, R. (2009). *Python para todos*. España: Creative Commons.
- Guzdial, M., & Barbara, E. (2013). *Introducción a la computación y Programación con Python*. México: PEARSON EDUCACIÓN.
- Hayt, W., Kemmerly, J., & Durbin, S. ( 2012). *Análisis de Circuitos en Ingeniería* . México: McGraw Hill .
- Hertzog, R., & Mas, R. (2013). *El Libro del Administrador de Debian* . Copyright .
- Horenstein, M. N. (1997). *Microelectrónica: Circuitos y Dispositivos* . Prentice Hall .
- Joyanes, L., Rodríguez, L., & Fernández, M. (1996). *Fundamentos de programación*. México: McGraw-Hill.

- Kernighan, B., & Ritchie, D. (1991). *El Lenguaje de programación C*. México: Pearson Educación.
- Lago, A. (2012). *Dispositivos y circuitos electrónicos analógicos. Aplicación práctica en el laboratorio*. Madrid: Andavira Editora.
- Lindsay, A. (2009). *Analógico y Digital Básicos*. México: Parallax.
- Maloney, T. J. (2006). *Electrónica industrial moderna. 5a. edición*. México: PEARSON EDUCACIÓN.
- Mano, M. (2003). *Diseño Digital*. México: Pearson Educación .
- Medrano, A., & Chourio, L. (2015). *HARDWARE LIBRE Y ABIERTO, MODELOS DE NEGOCIOS PARA AMÉRICA*. Venezuela: Fundación CENDITEL.
- Monk, S. (2013). *Raspberry Pi Cookbook*. CA: O'Reilly Media.
- Muhammad, R. (2002). *Circuitos Microelectrónicas*. Thompson.
- Neamen, D. A. (2010). *Microelectronics : Circuit Analysis and Design* . New York: McGraw-Hill.
- Pérez, E. M., Acevedo, J. M., & Silva, C. F. (2009). *Autómatas programables y sistemas de automatización*. Marcombo.
- Proakis, J. G., & Manolakis, D. G. (1999). *Digital Signal Processing: Principles, Algorithms, and Applications*. New Jersey: Prentice-Hall.
- Ramos, G. (2002). *Electrónica Industrial y Automatización*. Colombia: Cedit.
- Robert Boylestad, L. N. (2009). *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*. México: Pearson Educación.
- Ruiz, A. S. (2010). *Guía Práctica de Sensores*. Madrid: Creaciones Copyright, S.L.
- Tocci, R. J., & Welmer, N. S. (2003). *Sistemas Digitales: Principios y Aplicaciones*. México: Pearson Educación.
- Tomasi, W. (2003). *Sistemas de Comunicaciones Electrónicas*. México : Prentice Hall.

Torres, F. (2002). *Automática y Robótica*. México: Prentice Hall.

Torres, J., Lazalde, A., & Vila-Viñas, D. (2015). *Hardware libre (v.1.2)*. Quito: IAEN: Buen Conocer - FLOK Society documento de política pública 4.1.

Zabala, G. (2007). *Robótica*. Banfield: Users Express.

## URLS

Arduino. (13 de febrero de 2015). *Reading Fan RPM*. Obtenido de <http://playground.arduino.cc/Main/ReadingRPM>

Braga, N. d. (22 de febrero de 2015). *Como funcionan os sensores de Efeito Hall (ART1050)*. Obtenido de <http://www.newtonbraga.com.br/index.php/como-funciona/6640-como-funcionam-os-sensores-de-efeito-hall-art1050>

Crespo, E. (6 de marzo de 2015). *Tutorial de PyQt*. Obtenido de <https://ernestocrespo13.wordpress.com/category/pyqt/>

Cubietech. (5 de febrero de 2015). *A20-cubietruck GPIO Pin*. Obtenido de [http://docs.cubieboard.org/a20-cubietruck\\_gpio\\_pin](http://docs.cubieboard.org/a20-cubietruck_gpio_pin)

Cubietech. (4 de febrero de 2015). *Cubietech Cubietruck*. Obtenido de <http://linux-sunxi.org/Cubietruck>

Fritzing. (12 de febrero de 2015). *Breadboard View Graphics*. Obtenido de <http://fritzing.org/fritzings-graphic-standards/#download> Fritzing's

GitHub. (7 de febrero de 2015). *WiringCB for Python and c*. Obtenido de <https://github.com/gootoomoon/WiringCB-python>

LinkSprite. (7 de febrero de 2015). *Image for pcDuino3*. Obtenido de [http://www.linksprite.com/?page\\_id=855](http://www.linksprite.com/?page_id=855)

Linksprite. (25 de febrero de 2015). *How to use PhoenixSuit to Flash New Image to pcDuino3*. Obtenido de <http://learn.linksprite.com/pcduino/quick-start/pcduino3/how-to-use-phoenixsuit-to-flash-new-image-to-pcduino3/>

PyQt4. (7 de marzo de 2015). *Python Programming/PyQt4*. Obtenido de [https://en.wikibooks.org/wiki/Python\\_Programming/PyQt4](https://en.wikibooks.org/wiki/Python_Programming/PyQt4)

PyQt4\_8. (8 de febrero de 2015). *PyQt4 Reference Guide*. Obtenido de <http://pyqt.sourceforge.net/Docs/PyQt4/>

Sysmagazine. (3 de marzo de 2015). *Cubietruck. Practical applications*. Obtenido de <http://sysmagazine.com/posts/216211/>

# **ANEXOS**

*ANEXO 1 Fotografías del Módulo Didáctico Completo.*

*ANEXO 2 Manual del Módulo Didáctico.*

## *ANEXO 3 Datasheets*

*ANEXO 4 Hojas guías de prácticas.*