



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**  
ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

“Trabajo de grado previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones”

**TRABAJO DE GRADUACIÓN**

Título del proyecto:

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO  
CON INTERFAZ DE RED INDUSTRIAL PARA INTEGRARSE A UN  
SISTEMA SCADA**

**AUTOR:**

WILLIAM HUGO HURTADO GUAPULEMA  
DAVID ARMANDO SIGCHO CUVI

**Director:**

ING. FABIÁN GUNSHA

Riobamba – Ecuador

AÑO 2016

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO CON INTERFAZ DE RED INDUSTRIAL PARA INTEGRARSE A UN SISTEMA SCADA**, presentado por: **William Hugo Hurtado Guapulema** y **David Armando Sigcho Cuvi** y dirigida por: **Ingeniero Fabián Gunsha**.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

**Ing. Paulina Vélez**  
**Presidente del Tribunal**



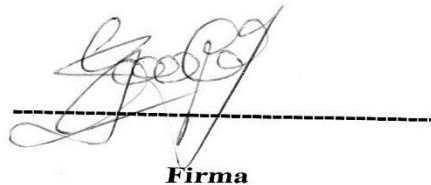
Firma

**Ing. Fabián Gunsha**  
**Director del Proyecto**



Firma

**Ing. Geovanny Cuzco**  
**Miembro del Tribunal**



Firma

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, corresponde exclusivamente a: **William Hugo Hurtado Guapulema, David Armando Sigcho Cuvi e Ingeniero Fabián Gunsha** y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



William Hurtado

CI: 060348797-6



David Sigcho

CI: 060419034-8

## INFORME DE TUTOR

Yo, Ing. FABIÁN CELSO GUNSHA MAJI, en mi calidad de tutor del trabajo investigativo con el título: DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO ELECTRÓNICO CON INTERFAZ DE RED INDUSTRIAL PARA SCADA, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuesto al público, luego de ser evaluado por el tribunal designado.

Atentamente.



Ing. Fabián Gunsha

TUTOR

## **AGRADECIMIENTO**

Agradezco a Dios por cada nuevo día de vida, a mis padres por todo su esfuerzo realizado para darme una buena educación, al Director y Asesor de tesis por las sugerencias efectuadas que permitieron el desarrollo exitoso del tema

A todos mil gracias.

William Hugo Hurtado Guapulema

## **AGRADECIMIENTO**

Agradezco a Dios por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

A mis padres, por ser mi ejemplo para seguir adelante en el convivir diario y por inculcarme valores que de una u otra forma me han servido en la vida, gracias por eso y por muchos más.

A mis hermanos por apoyarme en cada decisión que tomo, y por estar a mi lado en cada momento hoy, mañana y siempre.

Al Ing. Fabián Gunsha, director de tesis por su valiosa guía y asesoramiento a la realización de la misma.

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto.

David Armando Sigcho Cuvi

## **DEDICATORIA**

Dedico este trabajo a mis padres como muestra de agradecimiento por haber estado conmigo en el transcurso de mi desarrollo como profesional, a mis hermanos por ser la inspiración y la fuerza para salir adelante, por recibir su apoyo y cariño incondicional.

William Hugo Hurtado Guapulema

## **DEDICATORIA**

Al creador de todas las cosas, el que me ha dado fortaleza para continuar cuando a punto de caer he estado; por ello, con toda la humildad de mi corazón puede emanar, dedico primeramente mi trabajo a Dios.

De igual forma, dedico esta tesis a mis padres quienes con sus consejos me han sabido guiarme, han sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mis hermanos y abuelos que siempre han estado junto a mí y brindándome su apoyo.

David Armando Sigcho Cuvi



## ÍNDICE GENERAL

AUTORÍA DE LA INVESTIGACIÓN .....	III
AGRADECIMIENTO .....	V
DEDICATORIA .....	VII
ÍNDICE GENERAL.....	IX
ÍNDICE DE TABLAS .....	XIII
ÍNDICE DE FIGURAS.....	XV
RESUMEN.....	XVIII
SUMMARY .....	XIX
INTRODUCCIÓN .....	1
CAPÍTULO I.....	3
1. FUNDAMENTACIÓN TEÓRICA.....	3
1.1 Redes Industriales .....	3
1.2 Niveles en una red industrial.....	3
1.2.1 Nivel de entrada/salida.....	4
1.2.2 Nivel de control.....	4
1.2.3 Nivel de gestión.....	4
1.3 Protocolos industriales .....	5
1.3.1 Protocolos de Buses de Campo.....	5
1.4 Componentes de las redes industriales.....	10
1.5 Topología de Redes Industriales .....	11
1.6 Sistema SCADA .....	11
1.6.1 Componentes de hardware de SCADA.....	12

1.6.2	Software SCADA.....	14
1.7	LabVIEW .....	15
1.7.1	Distribución Debían .....	18
1.8	Tarjeta BeagleBone Black Industrial .....	20
1.8.1	Características técnicas de la Beaglebone Black Industrial .....	22
1.8.2	Lenguaje de programación IDE Cloud9 .....	23
1.8.2.1	Entorno.....	24
1.8.2.2	Editor .....	24
1.8.2.3	BoneScript .....	25
1.8.2.4	Debug .....	26
1.9	Arduino UNO.....	27
1.10	Ethernet Shield .....	27
1.11	Motor Eléctrico .....	28
1.11.1	Características del Motor Eléctrico Monofásico 1/2 Hp.....	29
1.12	Sensores .....	29
1.12.1	Sensores de Temperatura .....	30
1.12.2	Sensores de Humedad .....	30
1.13	Cable Ethernet Industrial.....	32
CAPÍTULO II .....		33
2.	METODOLOGÍA .....	33
2.1	TIPO DE ESTUDIO .....	33
2.2	POBLACIÓN Y MUESTRA .....	33
2.2.1	Población.....	33
2.2.2	Muestra.....	33
2.2.3	Hipótesis.....	34
2.3	OPERACIONALIZACIÓN DE VARIABLES .....	34
2.4	PROCEDIMIENTOS .....	35

2.5	PROCESAMIENTO Y ANÁLISIS .....	37
2.5.1	Diseño del controlador programable con Beaglebone Black Industrial..	37
2.5.1.1	Conexión de Beaglebone Black Industrial con el sensor DHT22 .....	37
2.5.1.2	Diseño electrónico del control ON/OFF .....	38
2.5.1.3	Diseño electrónico completo del controlador programable .....	39
2.5.1.4	Programación de la tarjeta Beaglebone Black Industrial.....	39
2.5.2	Conexión de Beaglebone Black Industrial con el Arduino UNO .....	42
2.5.3	Programación de la comunicación Ethernet del Arduino Uno.....	43
2.5.4	Interfaz desarrollada en Labview .....	46
2.5.4.1	Pantalla de presentación .....	48
2.5.4.2	Pantalla principal.....	48
2.5.4.3	Pantalla de Históricos .....	50
2.6	COMPROBACIÓN DE LA HIPÓTESIS .....	50
2.6.1	Planteamiento de la hipótesis estadística .....	50
2.6.2	Establecimiento del Nivel de significancia .....	51
2.6.3	Descripción de las poblaciones y suposiciones.....	51
2.6.4	Determinación del estadístico pertinente .....	51
2.6.5	Determinación del estadístico de prueba .....	51
CAPÍTULO III.....		55
3.	RESULTADOS.....	55
3.1	Pruebas del sistema .....	55
3.2	Resultados del sistema .....	58
3.3	Análisis financiero .....	61
CAPÍTULO IV .....		62
4.	DISCUSIÓN .....	62
CAPÍTULO V .....		64
5.	CONCLUSIONES Y RECOMENDACIONES.....	64
5.1	Conclusiones .....	64

5.2	Recomendaciones.....	64
CAPÍTULO VI.....		66
6.	PROPUESTA.....	66
6.1	Título de la Propuesta.....	66
6.2	Introducción .....	66
6.3	Objetivos .....	67
6.3.1	General .....	67
6.3.2	Específicos .....	67
6.4	Fundamentación científico –técnica.....	67
6.5	Descripción de la Propuesta .....	68
6.6	Diseño Organizacional .....	69
6.7	Monitoreo y Evaluación de la propuesta.....	69
CAPÍTULO VII .....		71
7.	BIBLIOGRAFÍA .....	71
CAPÍTULO VIII.....		73
8.	ANEXOS .....	73
8.1	Anexo 1 Programación de la BBBI.....	73
8.2	Anexo 2 Programación del Arduino UNO.....	75
8.3	Anexo 3 Hoja de datos Sensor DTH22 .....	79
8.4	Anexo 4 Hoja de datos del Arduino UNO .....	86
8.5	Anexo 5 Hoja de datos de Arduino Ethernet Shield .....	94

## ÍNDICE DE TABLAS

Tabla 1: Ejemplos de Software SCADA.....	15
Tabla 2: Características completas de la Beaglebone Black Industrial.....	23
Tabla 3: Sensores de temperatura en el mercado .....	30
Tabla 4: Sensores de Humedad en el mercado.....	31
Tabla 5: Sensores de Humedad y Temperatura.....	31
Tabla 6: Especificaciones del cable pre armado Rj45 .....	32
Tabla 7: Operacionalización de variables independiente y dependiente .....	35
Tabla 8: Funciones para crear cliente o servidor .....	40
Tabla 9: Configuración de Red Para el Arduino UNO .....	43
Tabla 10: Configuración cliente en el Arduino UNO .....	44
Tabla 11: Características técnicas del cliente (PC) .....	47
Tabla 12: Tiempos de respuesta con monitoreo.....	52
Tabla 13: Tiempos de respuesta sin monitoreo.....	53
Tabla 14: Media muestral .....	54
Tabla 15: Frecuencia esperada .....	54
Tabla 16: Calculo de Chi Cuadrado.....	54

Tabla 17: Promedio de muestras tomadas primer día .....	58
Tabla 18: Promedio de muestras tomadas segundo día .....	59
Tabla 19: Recursos Materiales .....	61

## ÍNDICE DE FIGURAS

Figura 1: Niveles de una red industrial. ....	4
Figura 2: Redes PROFIBUS .....	6
Figura 3: Conexión Hart en red Multipunto.....	6
Figura 4: Redes Foundation Fieldbus .....	7
Figura 5: Red Modbus.....	8
Figura 6: Red Ethernet/IP .....	9
Figura 7: Ethernet IP .....	10
Figura 8: Estructura básica de un Sistema Scada.....	12
Figura 9: Estructura básica de un sistema SCADA a nivel hardware.....	13
Figura 10: Panel Frontal de un VI.....	16
Figura 11: Paleta de controles .....	16
Figura 12: Diagrama de bloques de un VI .....	17
Figura 13: Paleta de funciones .....	18
Figura 14: Debian 8.2 (Jessie) con GNOME 3 .....	19
Figura 15: Diagrama de Bloques de la Tarjeta Beaglebone Black Industrial. ....	20
Figura 16: Beaglebone Black Industrial.....	21
Figura 17: Entorno de Cloud9.....	24

Figura 18: Vista dividida del editor Cloud9.....	25
Figura 19: Bonescript con un ejemplo de encendido de leds.....	25
Figura 20: Debug de Cloud9.....	26
Figura 21: Descripción del Arduino UNO .....	27
Figura 22: Arduino Ethernet placa vista frontal- vista trasera .....	28
Figura 23: Partes de un Motor.....	28
Figura 24: Motor Eléctrico WEG 1/2HP .....	29
Figura 25: Conectividad RJ45.....	32
Figura 26: Diagrama de bloques del sistema .....	36
Figura 27: Diagrama de conexión del sistema .....	36
Figura 28: Acondicionamiento de la señal que envía el sensor DHT22 .....	37
Figura 29: Conexión entre Beaglebone Black Industrial y sensor DHT22.....	37
Figura 30: Diagrama de flujo del circuito Control ON/OFF .....	38
Figura 31: Circuito para la activación y desactivación del motor.....	38
Figura 32: Diagrama completo del sistema.....	39
Figura 33: Código para la creación de un socket .....	39
Figura 34: Diagrama de Flujo cliente/servidor de la Beaglebone Black Industrial ....	41
Figura 35: Conexión de Beaglebone Black Industrial con el Arduino UNO.....	42
Figura 36: Diagrama de Flujo de la recepción y envío de datos en Arduino.....	46



Figura 37: Pantalla de presentación en Labview.....	48
Figura 38: Ingreso de contraseña en el programa .....	48
Figura 39: Pantalla principal en Labview .....	49
Figura 40: Pantalla de Históricos en Labview .....	50
Figura 41: Activación manual del motor .....	56
Figura 42: Motor Encendido .....	56
Figura 43: Monitoreo en Labview.....	57
Figura 44: Monitor serial Arduino .....	57
Figura 45: Gráfica del comportamiento de la Temperatura primer día. ....	59
Figura 46: Gráfica del comportamiento de la Humedad primer día. ....	60
Figura 47: Gráfica del comportamiento de la Temperatura segundo día.....	60
Figura 48: Gráfica del comportamiento de la Humedad segundo día.....	61
Figura 49: Diagrama del sistema Organizacional del Proyecto .....	69

## **RESUMEN**

El proyecto tiene como objetivo la implementación de un sistema de supervisión, control y adquisición de datos SCADA en Labview, para monitorear temperatura, humedad y la activación de un motor en tiempo real.

Para el diseño del prototipo se utilizó dos tarjetas de desarrollo: La Beaglebone Black Industrial que es un procesador de bajo costo basada en un procesador Sitara AM3358 ARM Cortex-A8, soporta una amplia gama de temperaturas entre -20 y 85 °C y un Arduino UNO acoplado a una Shield Ethernet que es el encargado de establecer la conexión con el Software Labview, por medio de una red industrial con el protocolo Ethernet/IP.

A medida que se añaden los módulos y sub-módulos a la VI en Labview, el sistema SCADA provoca un retardo en segundos en el envío de valores y las ordenes enviadas. Por tal razón el sistema resulta simple, permitiendo la comunicación entre el computador central y el sensor de campo.



UNIVERSIDAD NACIONAL DE CHIMBORAZO

CENTRO DE IDIOMAS INSTITUCIONAL

Lic. Lorena Gallegos

9 de agosto de 2016

### ABSTRACT

The objective of the project to implement a system of supervisory control and data acquisition SCADA in Labview to monitor the temperature, the humidity and the activation of an engine in real time.

Two cards of development were used in the design of the prototype: The Beaglebone Industrial Black which is a low cost processor based on a processor Sitara AM3358 ARM Cortex-A8. It supports a wide range of temperatures between -20 and 85°C and an Arduino one attached to a Shield Ethernet that is in charge of establishing the connection with the Labview Software. Through an industrial network with the Ethernet/IP protocol.

While you add the modules and sub-modules to the VI in Labview, the SCADA system causes a delay in seconds in the sending values and orders sent. For this reason the system is simple, allowing communication between the host computer and the sensor field.



## INTRODUCCIÓN

Las Industrias se han visto afectadas con el acelerado crecimiento de la tecnología en maquinaria de tipo industrial por lo que sus propietarios se han visto en la necesidad de equipar sus industrias con nuevos equipos, para poder competir con otras industrias, tanto a nivel nacional como con industrias de fuera del país.

Estos avances son positivos, pero también tiene lo negativo, que en este caso es el aspecto económico, puesto que la tecnología es cara y la mano de obra también lo es. Este problema ha ocasionado que no todos tengan acceso a estos avances, simplemente los únicos que pueden realizar estas inversiones son los grandes empresarios que poseen un capital suficiente.

Como solución al problema anteriormente presentado, el proyecto se centra en la implementación de un sistema de control y monitoreo de sensores de humedad y temperatura, implementando un protocolo de comunicación industrial comercial.

Entre los protocolos de comunicación industriales más utilizados se encuentra Ethernet/IP, basado en los protocolos estándar TCP/IP. Mediante su uso podremos acceder, configurar y controlar dispositivos de automatización industrial.

La particularidad que precisa a este sistema de monitoreo, es la activación y desactivación en tiempo real de un motor, también cuenta con un sensor de humedad y temperatura que son las variables para visualizar informes de lo que ocurre en una zona determinada, todo esto a través de un SCADA.

Al momento de realizar el diseño de un sistema SCADA hay que tomar en cuenta que software utilizar, debido a que existen una gran diversidad de Software dedicados a este fin, en este tema se preferido trabajar con el software Labview, que es un software líder en automatización y medición, de alto nivel, que permite gracias a su entorno de programación gráfico desarrollar proyectos de mayor eficiencia y calidad.

Los datos de temperatura, humedad y control ON-OFF del motor son procesados por la tarjeta de desarrollo Beaglebone Black Industrial, se enlaza con el Router TPLink que por medio de una interfaz Ethernet/IP se comunica al Shield Ethernet acoplado a Arduino Uno que es el encargado de establecer la conexión con el Software Labview.

La información que se puede visualizar en el SCADA son; valores actuales de las variables de humedad y temperatura y el desplazamiento en el tiempo del cambio de las mismas, la activación ON/OFF del motor, cambio de las condiciones de las variables de temperatura y humedad para activación o desactivación de alarmas.

# CAPÍTULO I

## 1. FUNDAMENTACIÓN TEÓRICA

### 1.1 Redes Industriales

En las comunicaciones, “Una arquitectura de red describe la forma en que se arregla o estructura una red de comunicación de datos y, en general, incluye el concepto de niveles o capas dentro de la arquitectura”. (Tomasi, 2003, p.605)

Por tanto, las Redes Industriales se puede definir como un conjunto de equipos de control que se enlazan para el intercambio de información, tales como:

- PC´s Industriales.
- Sensores.
- Microcontroladores.
- Controladores.
- Sistemas de Control Distribuido.
- Transductores y Actuadores.
- Módulos Inteligentes.

### 1.2 Niveles en una red industrial

En las plantas industriales modernas es muy común el empleo de computadoras y autómatas programables para un control efectivo y mejor nivel de tratamiento de la información, así también la implantación de redes de comunicación internas.

En una red industrial se pueden dividir elementalmente en tres niveles de automatización debido a la gran cantidad de equipos que coexisten, los cuales se muestran en la Figura 1.

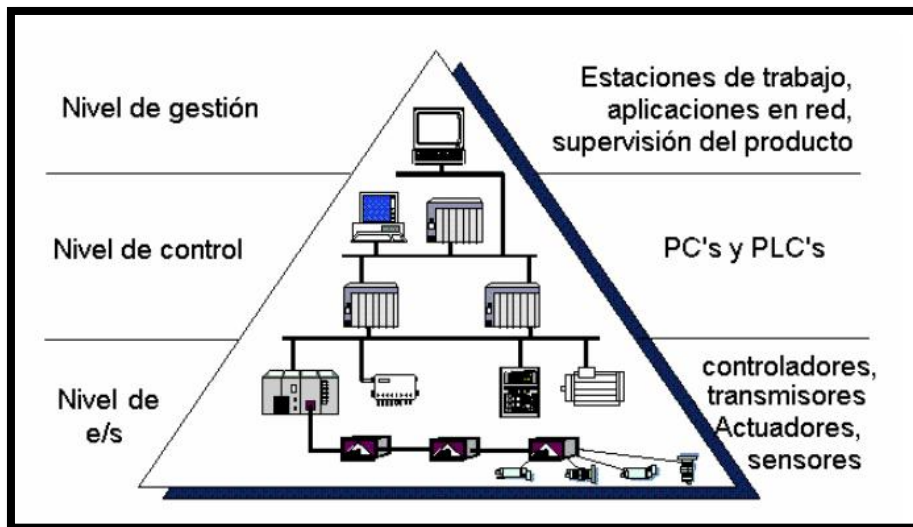


Figura 1: Niveles de una red industrial.

Fuente: <https://encrypted-tbn3.gstatic.com/images>

### 1.2.1 Nivel de entrada/salida

Este nivel es el más bajo de la red donde se conectan los dispositivos de adquisición de los datos en campo tales como: Sensores, Interfaces de operador, electroválvulas, controladores, etc. Son los elementos más directamente relacionados con el proceso productivo.

### 1.2.2 Nivel de control

Se localiza el nivel de los controladores, donde operan diversos sistemas y se pueden encontrar sistemas de control especializados, autómatas programables, buses de instrumentación sobre ordenadores, sistemas de supervisión y SCADA, entre otros.

### 1.2.3 Nivel de gestión

Este nivel se conoce como nivel de Información, en donde se emplean PC, estaciones de trabajo para realizar el control supervisorio del proceso. Este nivel maneja protocolos de comunicaciones estándares, tales como TCP/IP y Ethernet. Posibilita dotar de la información necesaria para que los operadores de una industria puedan tomar adecuadamente sus decisiones.

## 1.3 Protocolos industriales

### 1.3.1 Protocolos de Buses de Campo

Un bus de campo es un modo de transmisión de información (datos) por un sólo cable de comunicación que simplifica la instalación y operación de máquinas y equipos industriales utilizados en el proceso de producción. Son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como Sistema de Control Distribuido DCS, instrumentos de medida y transmisión, Controladores Lógicos Programables PLC y válvulas de control.

A continuación se indican los protocolos principales que funcionan a nivel de buses de campo y que nos permiten la integración de diferentes tecnologías independientemente del fabricante del dispositivo.

#### 1.3.1.1 Profibus

Es un estándar abierto e independiente que se acopla al modelo OSI y asegura una comunicación rápida y fácil entre dispositivos de diferentes fabricantes. Tal es así que Profibus tiene tres versiones, cada una de ellas especializada para un campo de comunicación como se observa en la Figura 2.

- **Profibus-DP** (Decentralized Periphery). Optimizado para aplicaciones de velocidad y bajo costo, orientado a sensores/actuadores enlazados a procesadores (PLC) o terminales.
- **Profibus-PA** (Process Automation). Utilizado para el control de procesos y cumple normas especiales de seguridad en ambientes peligrosos y con riesgo de explosión como la industria química (IEC 1 1158-2, seguridad intrínseca), su velocidad es de 31.5 Kbps y es aplicable a una distancia de 1,9 Km.
- **Profibus-FMS** (Fieldbus Message Specification). Es la solución para comunicación entre células de proceso o equipos de automatización, la evolución de Profibus hacia la utilización de protocolos TCP/IP para enlace al nivel de proceso hace que este perfil esté perdiendo importancia. Brinda



una alta velocidad de 9,6 Kbps a 1500 Kbps sobre distancias superiores a 100 Km.

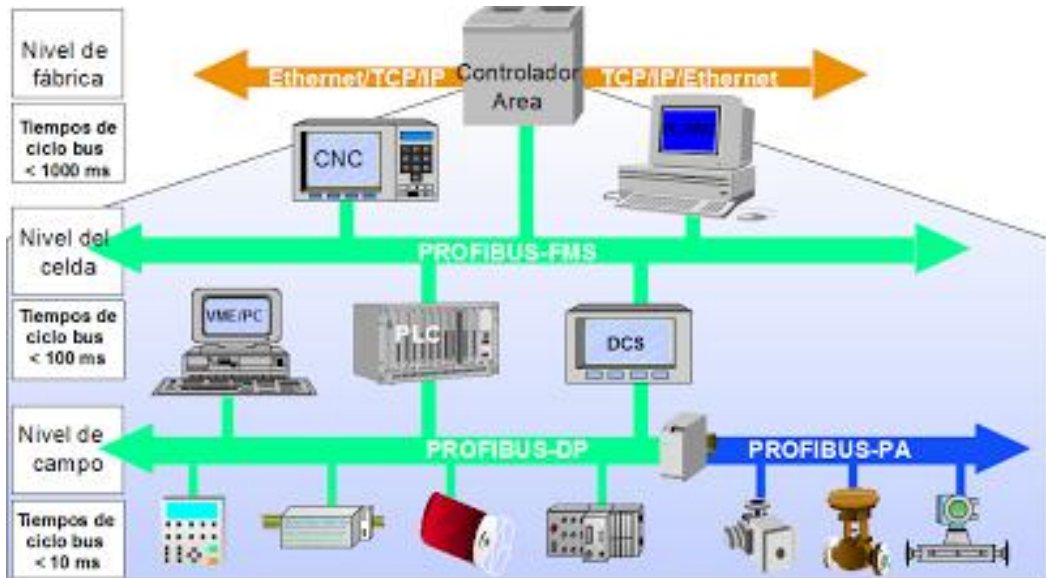


Figura 2: Redes PROFIBUS

Fuente: <http://comunicacionesindustrialesperez.blogspot.com/>

### 1.3.1.2 Hart

El protocolo HART (Highway Addressable Remote Transducer) agrupa la información digital sobre la señal analógica clásica de los sistemas de instrumentación, manteniendo estas en servicio. Ver Figura 3.

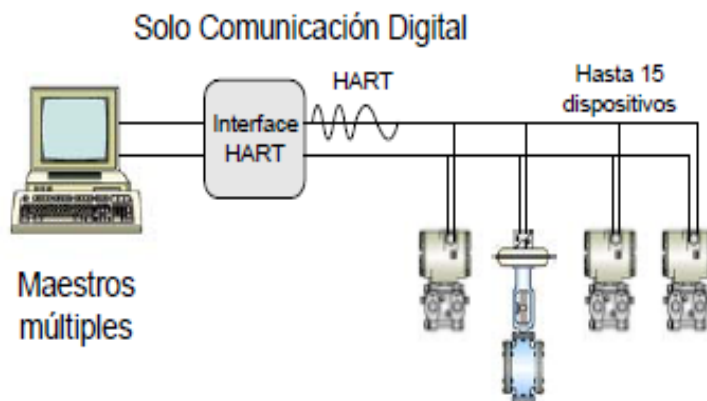


Figura 3: Conexión Hart en red Multipunto

Fuente: <http://docplayer.es/2252497-Escuela-politecnica-del-ejercito-extension-latacunga.html>

Utiliza el bus estándar 4 a 20 mA DC sobre el que transmite una señal digital que usa dos frecuencias individuales de 1200 y 2200 Hz, que representan los dígitos 1 y 0 respectivamente y que en conjunto forman una onda sinusoidal que se superpone al lazo de corriente de 4-20 mA.

### 1.3.1.3 Foundation fieldbus

El protocolo Foundation Fieldbus (FF) es un protocolo de comunicación digital para redes industriales, específicamente utilizado en aplicaciones de control distribuido. Puede comunicar grandes volúmenes de información, ideal para aplicaciones con varios lazos complejos de control de procesos y automatización. Está orientado principalmente a la interconexión de dispositivos en industrias de proceso continuo con una velocidad de transmisión generalmente de 38.4 Kbps. Ver Figura 4.

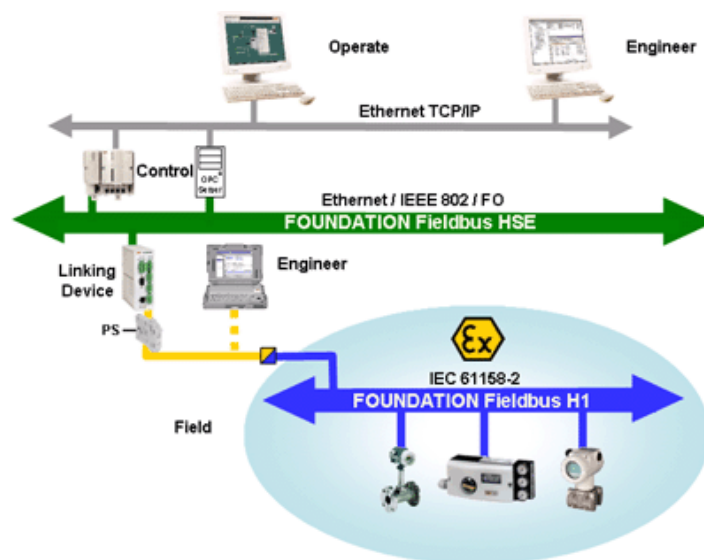


Figura 4: Redes Foundation Fieldbus

Fuente: <https://www02.abb.com/>

La integración de Foundation Fieldbus en la Industria aprovecha al máximo las características avanzadas que ofrece el protocolo. Estos incluyen la distribución de control en el campo, las estrategias redundantes y aplicaciones de optimización de activos, incluyendo el mantenimiento predictivo.

### 1.3.1.4 Modbus

Es un protocolo de comunicación estándar abierto desarrollado para sistemas de control de procesos, la velocidad de transmisión con este protocolo generalmente es de 38.4 Kbps, 9.6Kbps, 19.2 Kbps y soporta 32 nodos sin repetidores y 64 nodos con repetidores.

Este estándar solamente se refiere a la capa de red y de aplicación, por lo cual cualquier método físico de transporte puede ser utilizado. Es un estándar muy popular que opera en un ambiente esclavo-maestro con hasta 247 esclavos. Ver Figura 5.

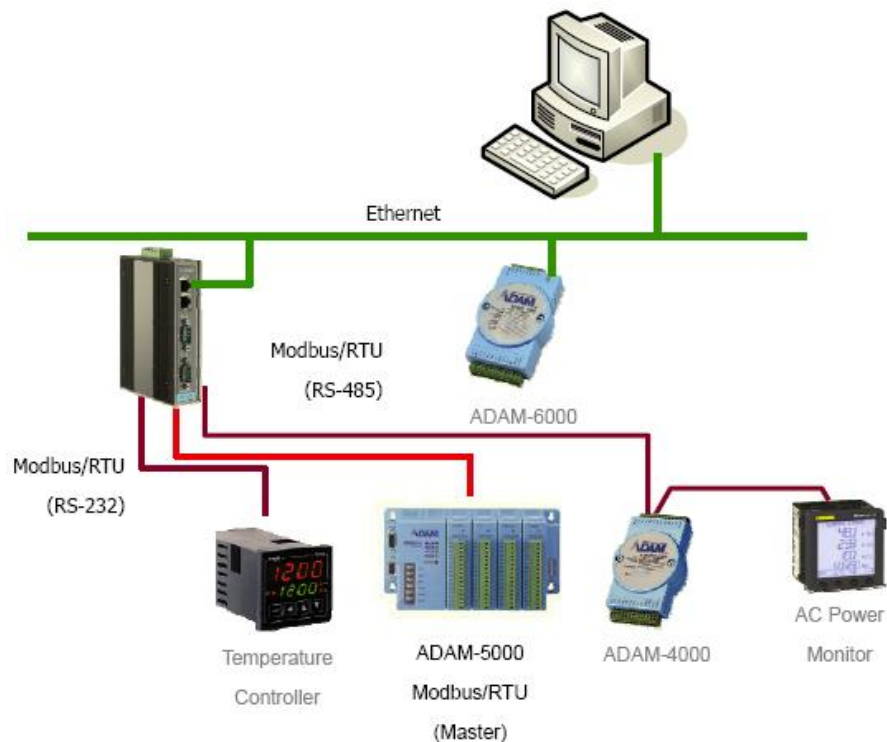


Figura 5: Red Modbus

Fuente: <http://www.suggest-keywords.com/bW9kYnVzIHJ0dQ/>

### 1.3.1.5 Ethernet industrial

El protocolo industrial Ethernet (EtherNet/IP) es un estándar abierto para la interconexión de redes industriales que aprovecha los medios físicos y los chips de comunicaciones Ethernet comerciales. Este protocolo está estandarizado en la

norma internacional IEEE 802.3, IEC 61158 y los dispositivos Ethernet/IP están certificados por ODVA para la interoperabilidad y conformidad (Ver Figura 6).

La utilización de esta tecnología en el nivel de planta permite la simplificación y agrupación de redes existentes, y reduce los costos de instalación mantenimiento y entrenamiento. Los equipos para Ethernet industrial están diseñados para operar en ambientes de planta cumpliendo o excediendo las comunicaciones de los equipos que interconectan tales como PLC, HMI y robots, entre otros.

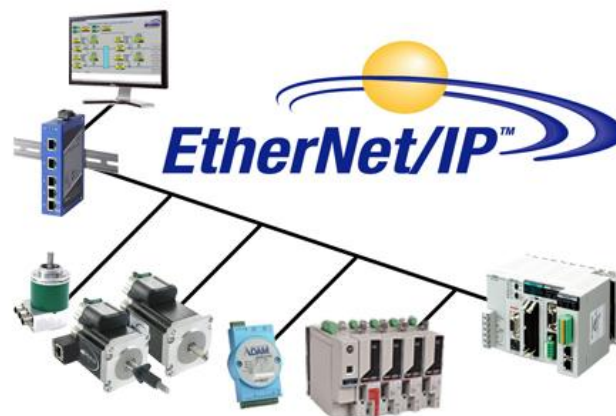


Figura 6: Red Ethernet/IP

Fuente: <http://industriyelethernetswitch.net/?m=201512>

Ethernet/IP permite el uso de aplicaciones de control con el protocolo industrial común (CIP), el mismo protocolo y modelo de objetos de capa superior que se encuentra en DeviceNet y ControlNet. CIP permite EtherNet/IP y los integradores de sistemas y usuarios DeviceNet para aplicar los mismos objetos y perfiles para plug-and-play de la interoperabilidad entre los dispositivos de diferentes fabricantes y en múltiples sub-redes. La combinación y complementación de las redes DeviceNet, ControlNet y Ethernet/IP proporciona un único sistema universal, además del apoyo de este grupo de fabricantes al software de la empresa.

Ethernet/IP utiliza todos los protocolos de transporte y de control de Ethernet estándar, incluyendo el Protocolo de Transporte de control (TCP), el usuario Protocolo de datagramas (UDP), el Protocolo de Internet (IP) y el acceso a los medios y tecnologías de señalización que se encuentran en off-the-shelf la

tecnología Ethernet. Sobre la base de estas tecnologías de comunicación estándar significa que Ethernet / IP funciona de forma transparente con todos los dispositivos Ethernet estándar que se encuentran en la plaza del mercado de hoy en día. También significa que Ethernet/IP se beneficia automáticamente de todas las mejoras tecnológicas adicionales, tales como Gigabit Ethernet y tecnologías inalámbricas (Ver Figura 7).

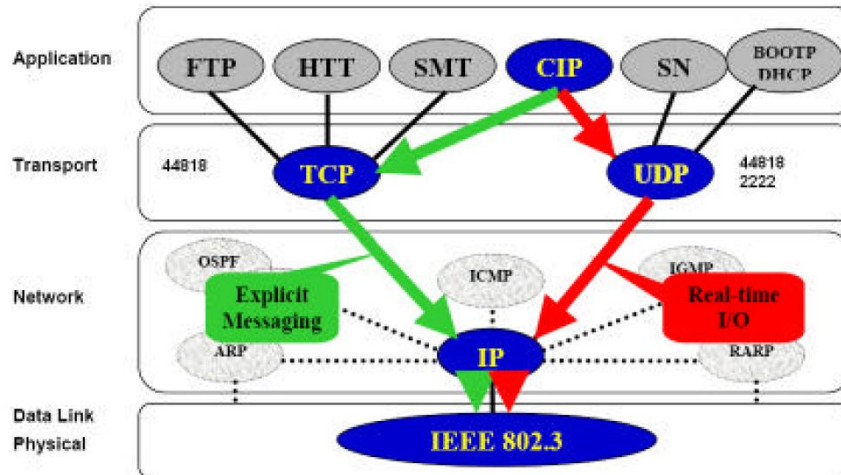


Figura 7: Ethernet IP

Fuente: [http://www.anybus.com/technologies/ethernetip\\_tech.shtml](http://www.anybus.com/technologies/ethernetip_tech.shtml)

#### 1.4 Componentes de las redes industriales

En grandes redes industriales, un simple cable no es suficiente para conectar el conjunto de todos los nodos de la red. Deben definirse topologías y diseños de redes para proveer un aislamiento y conocer los requerimientos de funcionamiento. (Sosa, 2011) resume lo siguiente:

- Bridge: Es un puente la conexión entre dos diferentes secciones de red, puede tener diferentes características eléctricas y protocolos; además puede enlazar dos redes diferentes.
- Repetidor: Es un dispositivo que intensifica las señales eléctricas para que puedan viajar grandes distancias entre nodos. Con este dispositivo se pueden conectar un gran número de nodos a la red; además, se pueden adaptar a diferentes medios físicos como cable coaxial o fibra óptica.

- Gateway: Es similar a un puente, ya que suministra interoperabilidad entre buses y diferentes tipos de protocolos; además, las aplicaciones pueden comunicarse a través de él.
- Enrutadores: Es un switch "enrutador" de paquetes de comunicación entre diferentes segmentos de red que definen la ruta hacia donde se transmite la información.

## 1.5 Topología de Redes Industriales

Los sistemas industriales usualmente consisten en dos o más dispositivos. Como un sistema industrial puede ser tremendamente grande, debe considerarse la topología de la red. Las topologías más comunes son:

- Red Bus: Se caracteriza por tener un único canal llamado bus troncal o backbone, se conecta en los diferentes dispositivos o llamados nodos.
- Red Estrella: Es la forma física en que todas las estaciones están conectadas a un solo nodo central.
- Red Anillo: Se compone de un solo anillo. En la que cada estación tiene una única conexión de entrada y otra de salida, cada entrada tiene un transmisor y un receptor.

## 1.6 Sistema SCADA

La expresión SCADA (Supervisory Control And Data Acquisition) que en español se traduce como Control Supervisor y Adquisición de Datos, usualmente se refiere a cualquier software que permita la adquisición de datos, supervisión y control de procesos industriales utilizando las herramientas de comunicación.

Tienen una estructura funcional Maestro-Esclavo como se puede observar en la Figura 8, donde cada dispositivo Remoto (Esclavo) responde cuando es interrogado desde una estación central (Maestro). Los mensajes son enviados desde el Maestro a intervalos regulares y son atendidos por todas los Esclavos, aunque solo responde aquella que reconoce su propio número de identificación.

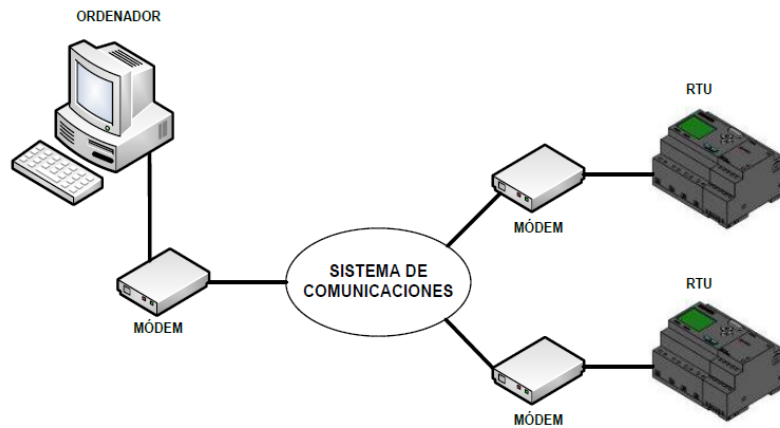


Figura 8: Estructura básica de un Sistema Scada.

Fuente: <http://www.uninotas.net/ventajas-y-desventajas-del-sistema-scada/>

### 1.6.1 Componentes de hardware de SCADA

Un sistema SCADA está formado por ciertos componentes de hardware en su sistema, para poder tratar y gestionar la información recibida. En la Figura 9 se muestra una estructura básica de estos sistemas.

Un sistema Scada está dividido en dos grandes grupos:

- a) **Captadores de datos:** Recopilan la información de los elementos de control del sistema (reguladores autómatas programables, registradores) y los procesan para su uso, es decir son los servidores del sistema.
- b) **Utilizadores de datos:** Utilizan la información recopilada por los anteriores, como pueden ser los operadores del sistema o las herramientas de análisis de datos, es decir son los usuarios del sistema.

El nivel de proceso e instrumentación conformado por los buses de campo, los controladores de proceso (sistemas de regulación o generalmente autómatas programables) envían la información a los servidores de datos (Data Servers) los cuales, a su vez, intercambian la información con el nivel de supervisión del sistema automatizado a través de redes de comunicaciones de área local.

Estos sistemas están formados por los siguientes elementos básicos:

- Unidad Central (MTU).
- Unidad Remota (RTU).
- Sistema de Comunicaciones.
- Instrumentos de Campo.

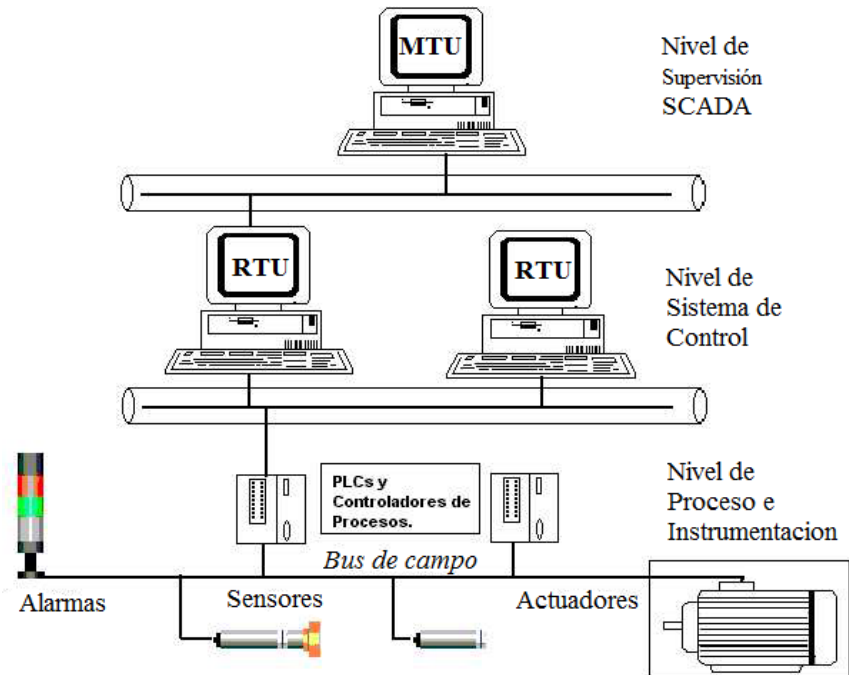


Figura 9: Estructura básica de un sistema SCADA a nivel hardware.

Fuente: Los Autores

### 1.6.1.1 Unidad central maestra (MTU)

La MTU es el equipo (ordenador o servidor) principal del sistema, el cual supervisa y recoge la información del resto de las unidades remotas; soporta una interfaz hombre-máquina. El sistema SCADA más sencillo es el compuesto por un único computador, el cual es la MTU que supervisa toda la estación desempeñando las siguientes funciones:

- Supervisión
- Gestionar las comunicaciones.
- Adquisición y almacenamiento de datos de las RTU.
- Envío de información.
- Comunicación con los operadores.
- Visualización de datos.



### **1.6.1.2 Unidad terminal remota (RTU)**

Una RTU es un equipo situado en un lugar remoto del sistema, está encargado de la adquisición de datos para luego ser transmitidos hacia la MTU. Estas unidades son sistemas basados en microcontroladores, de composición modular, provista de canales de entrada para detección o medición de las variables de un proceso y de canales de salida para control o activación de alarmas y un puerto de comunicaciones.

### **1.6.1.3 Sistema de comunicaciones**

Estos sistemas se encargan de la transferencia de información entre la unidad central y los clientes del sistema SCADA, como con las unidades remotas. Puede ser construida con cables o inalámbrica, utilizando cualquier protocolo industrial existente en el mercado, como por ejemplo; CANbus, Fieldbus, Modbus, etc.

### **1.6.1.4 Instrumentos de Campo**

Son todos aquellos dispositivos que permiten realizar la automatización o control del sistema (PLC's, controladores de procesos industriales, y actuadores en general), también se encargan de la captación de información de entornos físicos (sensores y transductores).

## **1.6.2 Software SCADA**

El software de un sistema SCADA es el encargado de administrar la adquisición y controlar el proceso de los datos a través de la MTU, a su vez, por medio de este, acceden los clientes SCADA a los datos del sistema (Bailey, 2003). Entre las importantes funciones que cumple el software se encuentran las siguientes:

- Manejo del soporte o canal de comunicación.
- Manejo de uno o varios protocolos de comunicación.
- Manejo y actualización de una base de datos.
- Administración de alarmas.
- Generación de archivos históricos.

- Interfaz con el operador.
- Capacidad de programación.
- Transferencia dinámica de datos.
- Conexión a redes de comunicación.

En el mercado podemos encontrar una gran variedad de software SCADA en sus dos plataformas de desarrollo, la plataforma propietaria son desarrolladas por los mismos fabricantes de equipos de automatización, mientras que la plataforma abierta son desarrolladas por empresas dedicadas a aplicaciones SCADA. Ejemplos de este tipo de software se presenta en la Tabla 1.

Tabla 1: Ejemplos de Software SCADA

SOFTWARE	FABRICANTE	TIPO DE PLATAFORMA
Ifix	General electric	Abierta
Intouch	Wonderware	Abierta
Lookout	National Instrument	Abierta
Rsvi32	Rockwell Automation	Propietaria
Wincc	Siemens	Propietaria

Fuente: Los Autores

## 1.7 LabVIEW

Labview (Laboratory Virtual Instrument Engineering Workbench) es un software diseñado para diversos sistemas operativos utilizados en el mundo como Linux y Windows entre otros. Según (Lajara y Pelegrí, 2011) LabVIEW es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control, por ello, a todos los módulos creados con Labview se les llama VI (Instrumento Virtual).

Un VI es un módulo de software que representa el panel frontal utilizando elementos de hardware accesibles por el ordenador (tarjetas de adquisición de datos (DAC), puerto serie RS-232, GPIB, VXI, USB, Ethernet), que al ser ejecutado se logra observar en la pantalla del computadora un panel cuya función es similar a la

de un instrumentó físico, preparando la visualización y el control del aparato. A partir de los datos mostrados en el panel frontal, el VI debe actuar generando o recogiendo señales, como lo haría su equivalente físico.

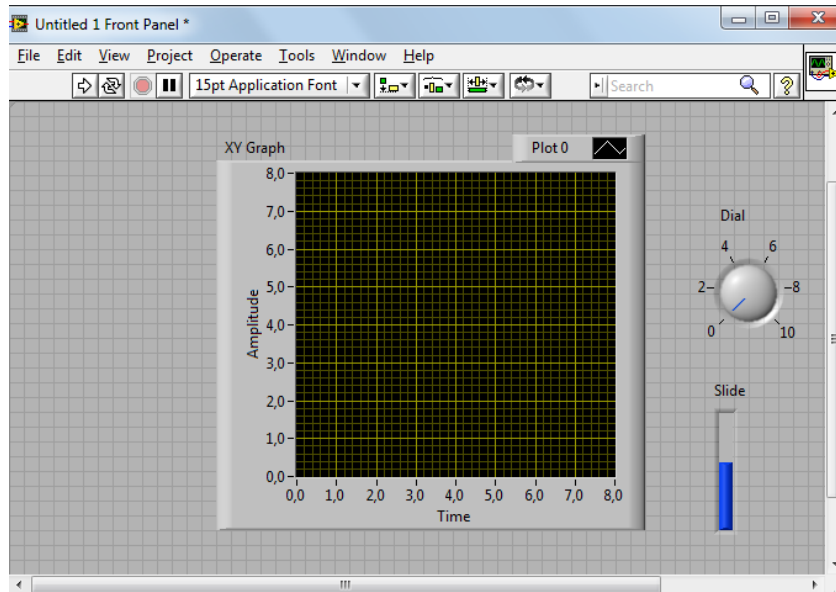


Figura 10: Panel Frontal de un VI

Fuente: Los Autores

Los VI se dividen en dos ventanas; la ventana del Panel Frontal y el Diagrama de Bloques. El panel frontal interactúa con el usuario cuando el programa se está ejecutando, permitiendo controlar el programa, cambiar las entradas, y ver los datos actualizados en tiempo real en la pantalla, como se observa en la Figura 10.

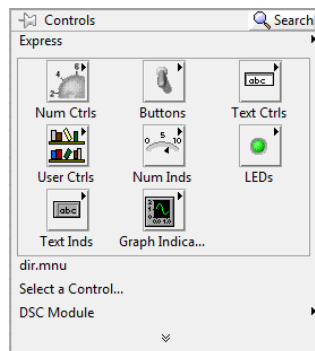


Figura 11: Paleta de controles

Fuente: Los Autores

En la Figura 11 se muestra la paleta de controles, el que contiene todos los controles e indicadores que se utilizan para crear el panel frontal. Para ingresar a este menú es cuestión de dar un clic derecho al mouse dentro del espacio en blanco de la ventana del panel frontal.

Los controles se utilizan para entregar datos al diagrama de bloques desde el panel de control por ejemplo ajustando un control deslizante para establecer un valor de alarma, encender un interruptor de encendido o apagado, o para detener un programa. Los indicadores se utilizan como salida de datos al panel frontal desde el diagrama de bloques como pueden ser termómetros, leds y otros indicadores que muestran valores de salida del programa. Estos pueden incluir datos, estados de programa y otra información (ver Figura 12).

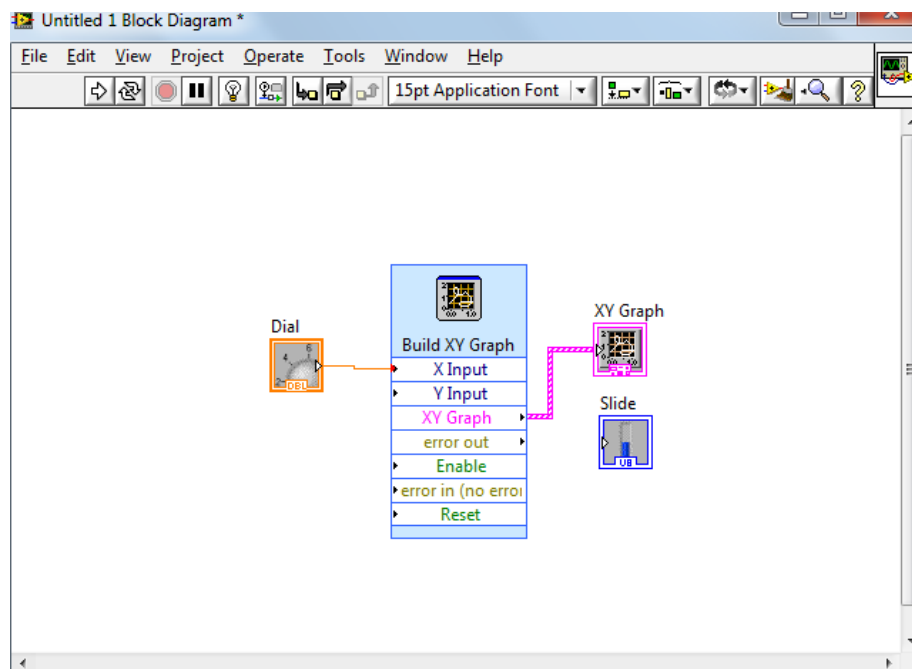


Figura 12: Diagrama de bloques de un VI

Fuente: Los Autores

En la Figura 13 se muestra la Paleta de funciones, el más operado al momento de programar; en él se encuentran todas las Vis, funciones y constantes para crear el diagrama de bloques y a través del mismo, es posible ingresar a este menú con solo dar un clic derecho al mouse dentro del diagrama de bloques.

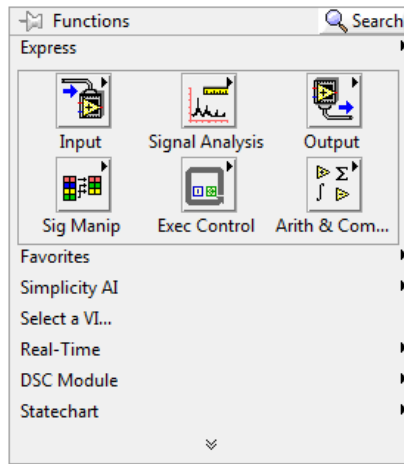


Figura 13: Paleta de funciones

Fuente: Los Autores

### 1.7.1 Distribución Debían

Debian es un sistema operativo completo, abarcando el software y los sistemas para su instalación y gestión, todo ello con base en el núcleo Linux y software libre. Debian es una distribución GNU/Linux. (Hertzog & Mas, 2015) afirman. “Una distribución GNU/Linux tiene dos objetivos principales: instalar un sistema operativo libre en un equipo (sea con o sin uno o más sistemas preexistentes) y proveer un rango de programas que cubran todas las necesidades del usuario” (p.24).

Esta distribución se compone de un gran número de paquetes de software. Cada paquete en la distribución contiene ejecutables, scripts, documentación, e información de configuración, y tiene un encargado que es el principal responsable de mantener el paquete al día, el seguimiento de los informes de fallo y comunicarse con los autores del software empaquetado. Nuestra gran base de usuarios, combinado con nuestro sistema de seguimiento de errores, asegura que los problemas se encuentren y resuelvan rápidamente (Ver Figura 14).

- **Multitarea:** Debian tiene la capacidad de manejar más de una trabajo a la vez. Esto se logró al dividir el tiempo de CPU de forma inteligente, donde algunas tareas puedan ejecutarse detrás sin impedir la ejecución de la tarea desplegada en la pantalla.

- **Multiusuario:** Esta es una capacidad del sistema operativo Debian, donde, las mismas aplicaciones pueden ser utilizadas por varios usuarios. Por supuesto, no en un solo terminal, se les da diferentes terminales para operar. Todos los terminales se conectan al servidor principal o Host Machine Linux, cuyos recursos y dispositivos periféricos conectados, se pueden utilizar.
- **Portabilidad:** Significa que el sistema operativo Debian y sus aplicaciones pueden trabajar en diferentes tipos de dispositivos de la misma manera. Kernel de Linux y los programas de aplicación compatibles con su instalación incluso en la configuración de hardware mínimo.
- **Comunicación:** Debian tiene una excelente característica para comunicarse con los otros usuarios, puede ser dentro de la red de un solo ordenador principal, o entre dos o más de tales redes de ordenadores. Los usuarios pueden intercambiar fácilmente el correo, datos, programa a través de dichas redes.

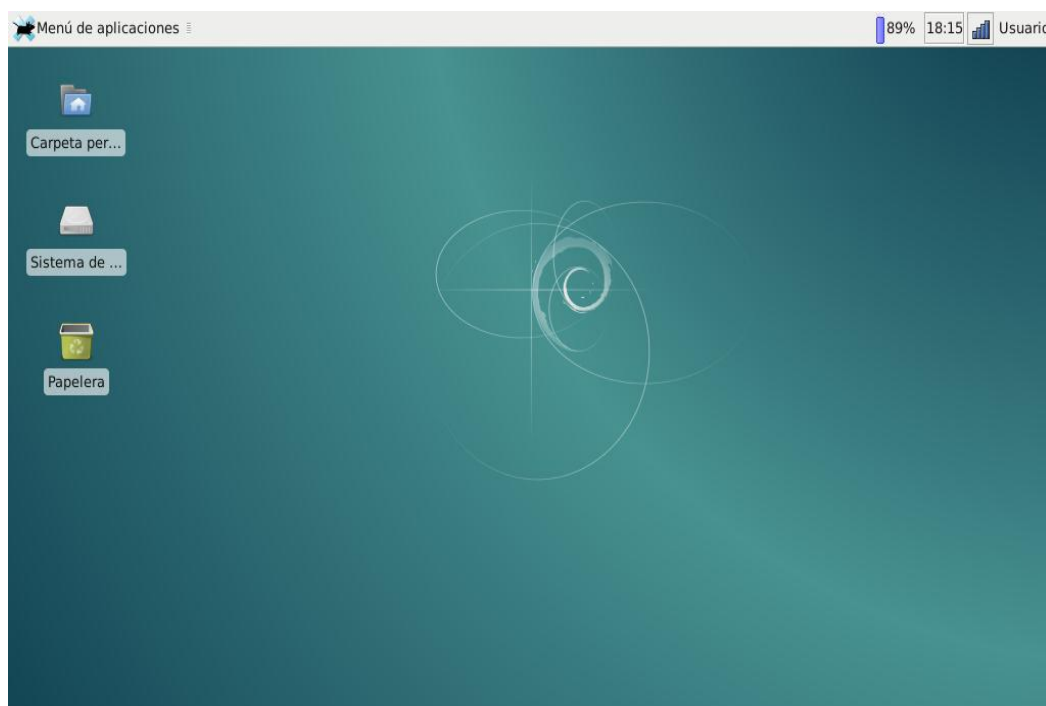


Figura 14: Debian 8.2 (Jessie) con GNOME 3

Fuente: Los Autores

## 1.8 Tarjeta BeagleBone Black Industrial

La tarjeta element14 BEAGLEBONE Negro Industrial (BBONE-NEGRO-IND-4G) es un procesador de bajo costo basada en un procesador Sitara AM3358 ARM Cortex-A8 de bajo coste grado de temperatura industrial de Texas Instruments BBONE-NEGRO-IND-4G viene con el sistema operativo Debian distribución de Linux preinstalado en el FLASH a bordo listo para iniciar la evaluación y el desarrollo.

Tal como se puede observar en el diagrama de bloques mostrado en la Figura 15, la tarjeta BBONE-NEGRO-IND-4G cuenta con un microprocesador, memoria RAM, almacenamiento, puertos periféricos y otros elementos que tienen gran semejanza con un computador.

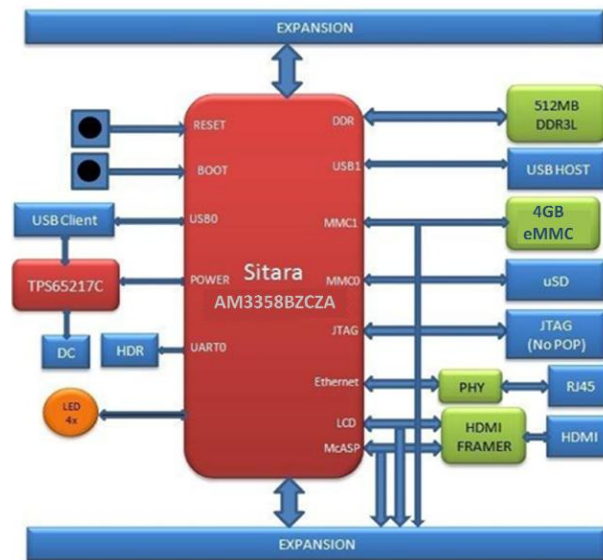


Figura 15: Diagrama de Bloques de la Tarjeta Beaglebone Black Industrial.

Fuente: Element14.

El microprocesador de la tarjeta BBONE-NEGRO-IND-4G es un Sitara AM3358BZCZA, basados en el procesador ARM Cortex-A8, de un solo núcleo con un bus de 32 bits, permite trabajar con memorias RAM actuales como DDR2 y DDR3, tiene puertos Ethernet y puertos USB. Además se han mejorado las

imágenes, procesamiento de gráficos, periféricos y opciones de interfaz industriales como EtherCAT, PROFIBUS y EtherNet/IP.

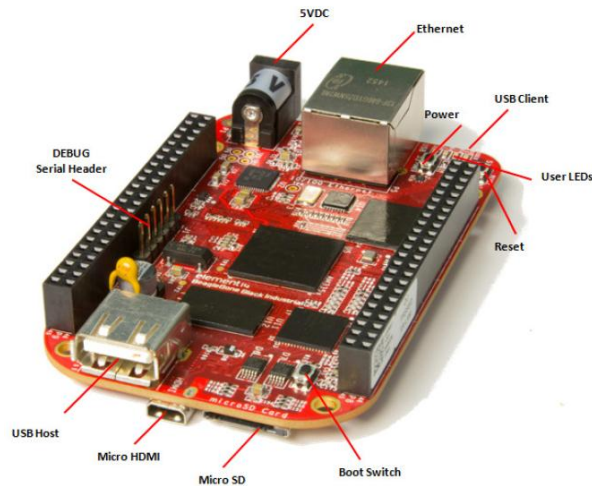


Figura 16: Beaglebone Black Industrial

Fuente: Element14.

El chip TPS65217C recibe el voltaje de alimentación de la tarjeta BBONE-NEGRO-IND-4G y lo acopla a sus demás partes como el microprocesador Sitara, los puertos y la memoria RAM. A continuación se detallan los componentes y la distribución de la BeagleBone Black Industrial y se aprecia en la Figura 16.

- **DC Power.**- Es la entrada principal de voltaje para alimentar a la placa, esta acepta 5V con una corriente de hasta 2A.
- **Puerto Ethernet.**- Permite realizar conexiones a redes LAN con velocidades de 10/100Mbps conexión RJ-45.
- **Serial Debug.**- Puerto que permite realizar comunicación serial.
- **USB Client.**- Es un conector mini-USB que permite conectar la tarjeta al computador y puede también servir como fuente de alimentación alternativa, brindando 5v y 1A de corriente.
- **BOOT switch.**- Este botón permite arrancar a la tarjeta en uno de los varios modos como por la micro-SD, serial o USB.



- **LEDS.-** Permiten visualizar el estado de funcionamiento de la tarjeta.
- **Reset Button.-** Permite inicializar al procesador.
- **Micro-SD slot.-** Ranura donde se coloca la micro-SD.
- **Micro-HDMI.-** Conector donde el monitor o display es conectado.
- **USB Host.-** Permite conectar diferentes dispositivos USB como teclado, Wi-fi, etc.

### 1.8.1 Características técnicas de la Beaglebone Black Industrial

Las características más importantes que incorpora la placa Beaglebone Black Industrial, son las enumeradas a continuación, que han sido extraídas de las especificaciones que se muestran en la tabla 2:

- Temperatura de funcionamiento: -20 a 85 °C
- Procesador: TI Sitara AM3358BZCZA100 1GHz ARM® Cortex-A8
  - 512MB de memoria RAM DDR3.
  - 4GB 8-bit eMMC de almacenamiento interno.
  - Acelerador gráfico 3D.
  - Acelerador de punto flotante NEON.
  - 2x PRU de 32-bit. (Unidades de procesado en tiempo real)
- Conectividad
  - Cliente para alimentación y comunicaciones USB.
  - USB host.
  - Conector Ethernet.
  - Micro-SD.
  - Puerto Micro HDMI.
  - 2 Conectores de 46 pines.
- Compatibilidad de software
  - Debian.
  - Android.
  - Cloud9 IDE en Node.js w / biblioteca BoneScript
  - Y muchos más.

Tabla 2: Características completas de la Beaglebone Black Industrial

	<b>Feature</b>	
<b>Processor</b>	Sitara AM3358BZCZ100	
<b>Graphics Engine</b>	1GHz, 2000 MIPS	
<b>SDRAM Memory</b>	SGX530 3D, 20M Polygons/S	
<b>Onboard Flash</b>	512MB DDR3L 800MHZ	
<b>PMIC</b>	4GB, 8bit Embedded MMC	
<b>Debug Support</b>	TPS65217C PMIC regulator and one additional LDO.	
<b>Power Source</b>	Optional Onboard 20-pin CTI JTAG, Serial Header	
<b>PCB</b>	miniUSB USB or DC Jack	5VDC External Via Expansion Header
<b>Indicators</b>	3.4" x 2.1"	6 layers
<b>HS USB 2.0 Client Port</b>	1-Power, 2-Ethernet, 4-User Controllable LEDs	
<b>HS USB 2.0 Host Port</b>	Access to USB0, Client mode via miniUSB	
<b>Serial Port</b>	Access to USB1, Type A Socket, 500mA LS/FS/HS	
<b>Ethernet</b>	UART0 access via 6 pin 3.3V TTL Header. Header is populated	
<b>SD/MMC Connector</b>	10/100, RJ45	
<b>User Input</b>	microSD , 3.3V	
<b>Video Out</b>	Reset Button Boot Button Power Button	
<b>Audio</b>	16b HDMI, 1280x1024 (MAX) 1024x768,1280x720,1440x900 ,1920x1080@24Hz w/EDID Support	
<b>Expansion Connectors</b>	Via HDMI Interface, Stereo	
<b>Weight</b>	Power 5V, 3.3V , VDD_ADC(1.8V) 3.3V I/O on all signals	
<b>Power</b>	McASP0, SPI1, I2C, GPIO(69 max), LCD, GPMC, MMC1, MMC2, 7 AIN(1.8V MAX), 4 Timers, 4 Serial Ports, CAN0, EHRPWM(0,2),XDMA Interrupt, Power button, Expansion Board ID (Up to 4 can be stacked)	
	1.4 oz (39.68 grams)	
	Refer to Section 6.1.7	

Fuente: Element14.

### 1.8.2 Lenguaje de programación IDE Cloud9

La Cloud9 IDE es una línea entorno de desarrollo integrado, publicado como software libre que combina un potente editor de código en línea con un espacio de trabajo completo de Ubuntu. Es compatible con cientos de lenguajes de programación, incluyendo PHP, Ruby, Perl, Python, JavaScript con Node.js y muchos más.

### 1.8.2.1 Entorno

El entorno de desarrollo de Cloud9 contiene un editor de texto para escribir los códigos, un área de mensajes, una barra de tareas con botones para las funciones más comunes y una serie de menús para interactuar con el usuario. Este se conecta al hardware del Beaglebone Black Industrial para comunicarse y cargar programas. (Ver Figura 17).

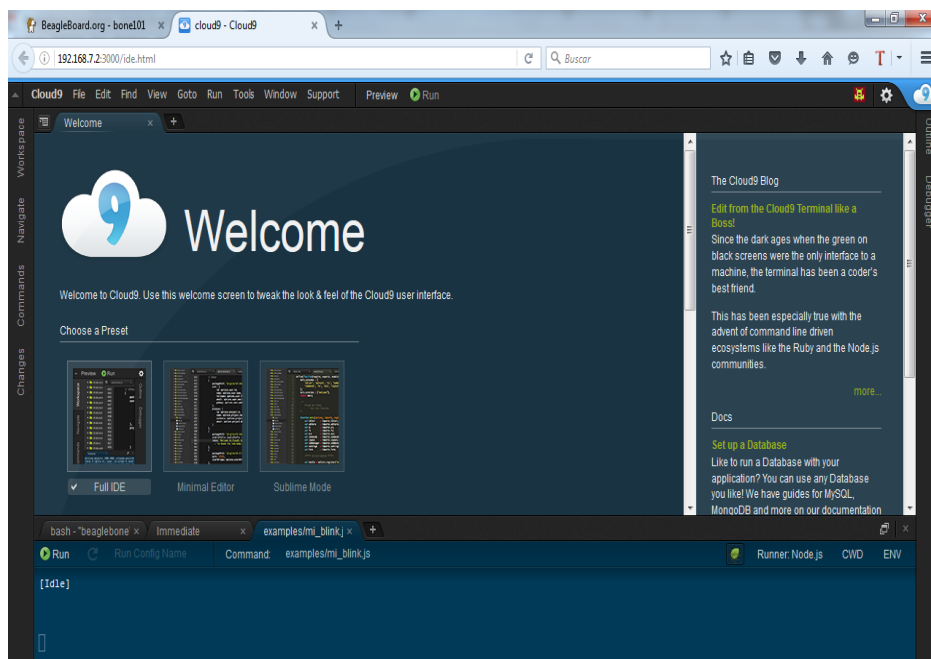


Figura 17: Entorno de Cloud9

Fuente: Los Autores

### 1.8.2.2 Editor

El editor es bastante potente. Configurable incluso en modo vim, emacs o modo sublime. Dispone de varios temas y esquemas de color que controlan el resultado de sintaxis y la interfaz de usuario. (Ver Figura 18).

Es posible dividir el editor en varias ventanas, basta con arrastrar cualquier archivo o pestaña terminal alrededor para crear tantas vistas divididas como desee en cualquier dirección. De esta forma se aprovecha más la pantalla y nos permite ver varios ficheros a la vez.

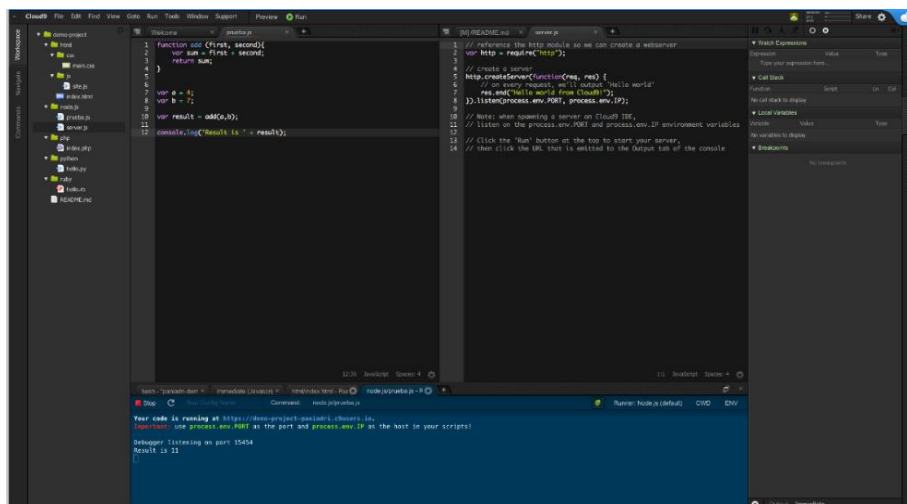


Figura 18: Vista dividida del editor Cloud9

Fuente: Los Autores

### 1.8.2.3 BoneScript

El software escrito para usar en Cloud9 se denomina BoneScript, es una biblioteca JavaScript para simplificar el aprendizaje de cómo realizar tareas de computación física utilizando el Linux embebido. Los archivos son guardados con la extensión `.js`. Esta extensión ofrece funciones como cortado/pegado y también para buscar y reemplazar texto. (Ver Figura 19).

```
1  var b = require('bonescript');
2
3  var state = b.LOW;
4
5  b.pinMode("USR0", b.OUTPUT);
6  b.pinMode("USR1", b.OUTPUT);
7  b.pinMode("USR2", b.OUTPUT);
8  b.pinMode("USR3", b.OUTPUT);
9  setInterval(toggle, 1000);
10
11 function toggle() {
12     if(state == b.LOW) state = b.HIGH;
13     else state = b.LOW;
14     b.digitalWrite("USR3", state);
15 }
```

Figura 19: Bonescript con un ejemplo de encendido de leds

Fuente: Los Autores

Sugerencias de código de finalización aparecen a medida que escribe para ayudarse con un código más rápido y evitar errores tipográficos. Utilice la vista de esquema para comprender mejor el código, y saltar a definición de variables para una fácil navegación.

### 1.8.2.4 Debug

El depurador permite establecer puntos de interrupción, sobre y a través de, e inspeccionar variables de cualquier aplicación JS / Node.js. O saltar a la derecha de la ventana Inmediata para ejecutar el código en una sesión de depuración.

En la parte superior del panel de debug encontramos las instrucciones propias para dirigir la ejecución según nos convenga: reanudar ejecución, pasar por encima, entrar en, etc. Además, podemos observar los puntos de interrupción, así como los valores de las variables locales en cada momento o de las expresiones que declaremos. (Ver Figura 20).

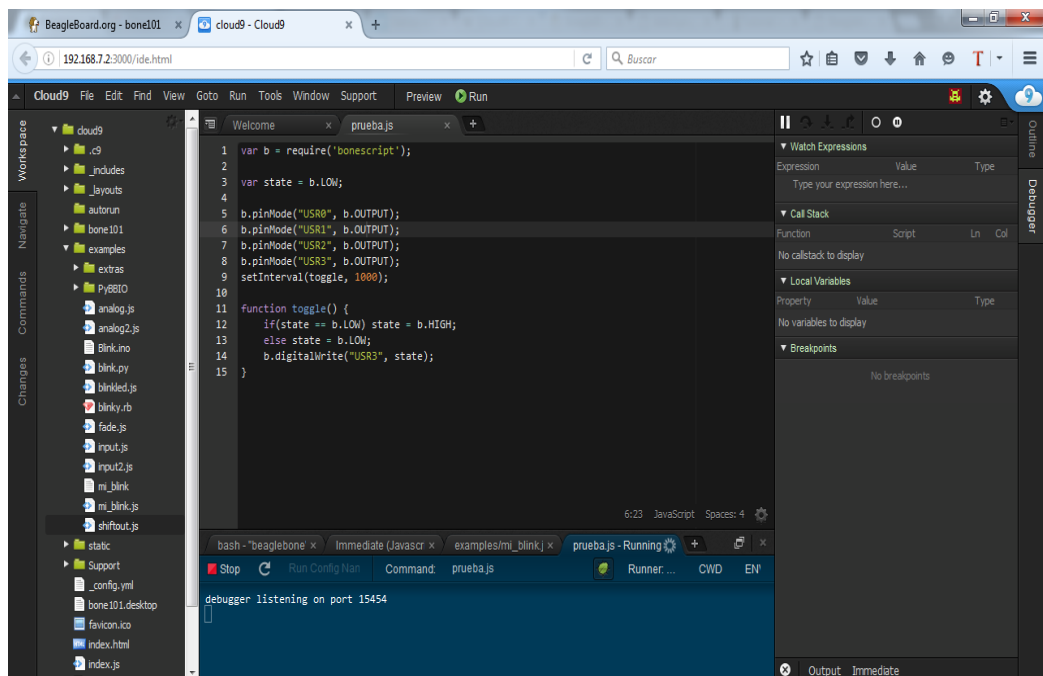


Figura 20: Debug de Cloud9

Fuente: Los Autores

## 1.9 Arduino UNO

El Arduino Uno es una placa electrónica basada en el Microcontrolador ATmega328P. Cuenta con 14 pines digitales de entrada/salida (de los cuales 6 se podrán utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador; basta con conectarlo a un ordenador con un cable USB o a la corriente con un adaptador de CA a CC o una batería. Ver Figura 21.

Arduino es una plataforma de código abierto muy popular entre la gente, basado en flexibilidad, hardware y software libre (Arduino ID) que se ejecuta en el ordenador para escribir y cargar el código de la computadora a la tarjeta física. La gran comunidad de usuarios han contribuido con códigos y han liberado instrucciones para una gran variedad de proyectos base en Arduino.

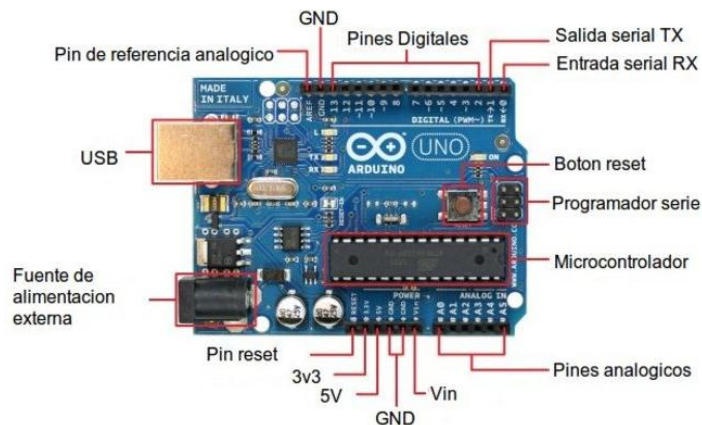


Figura 21: Descripción del Arduino UNO

Fuente: <http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>

## 1.10 Ethernet Shield

La Arduino Ethernet Shield es una placa electrónica que permite a una placa Arduino conectarse a internet (Ver Figura 22). Se basa en el chip de ethernet Wiznet W5100. El que ofrece una pila de red (IP) capaz de TCP y UDP. Es compatible con

hasta cuatro conexiones de socket simultáneas. Usa la librería Ethernet para escribir bocetos que se conectan a Internet a través de la pantalla.

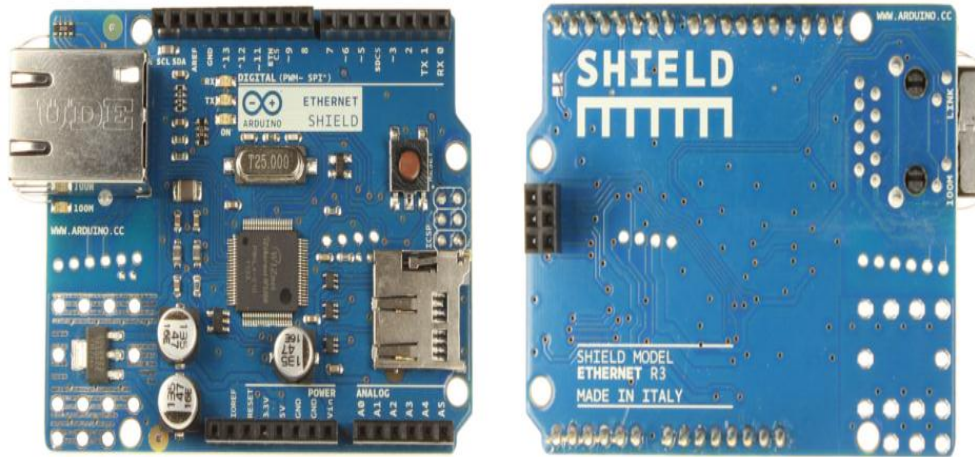


Figura 22: Arduino Ethernet placa vista frontal- vista trasera

Fuente: Ethernet Arduino

### 1.11 Motor Eléctrico

Un motor de corriente alterna se compone de dos partes básicas; un estator estacionario situado a fuera con las bobinas que se están suministrando corriente alterna con el fin de crear un campo magnético giratorio y el segundo componente siendo el rotor situado en el interior que está unido al eje de salida que produce un torque en virtud de campo magnético giratorio (Ver Figura 23).

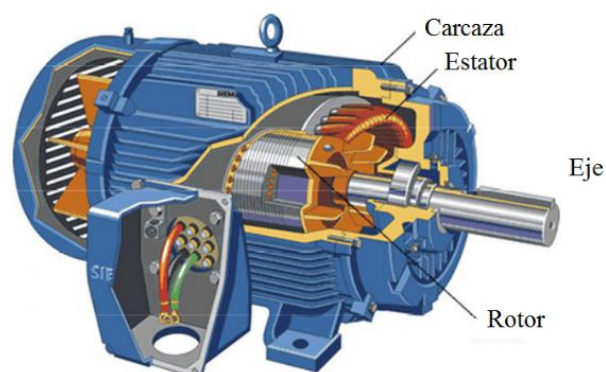


Figura 23: Partes de un Motor

Fuente: Los Autores



### 1.11.1 Características del Motor Eléctrico Monofásico 1/2 Hp

Es un Motor monofásico, carcasa de chapa, grado de protección IP21 y punta de eje con chaveta. Motor para aplicaciones en compresores, bombas, ventiladores y trituradores. Dimensiones según norma NEMA.

- 1/2 HP.
- Consumo energético 14 kWh/día
- Consumo energético máximo 18 kWh/día
- 10,8/4,6 A
- Tensiones: 127/220 V (Monofásico)
- Monofásicos, IV polos, 60Hz
- Capacitor de Arranque
- Rotor de Jaula de Ardilla
- Abierto a prueba de goteo, refrigerado por ventilador (ODP)



Figura 24: Motor Eléctrico WEG 1/2HP

Fuente: Los Autores

### 1.12 Sensores

Un sensor es un dispositivo que convierte la información que le llega del exterior en una señal eléctrica, normalmente digital (pasa o no pasa corriente), que puede ser analizado y procesado por la unidad de control del sistema.








### 1.12.1 Sensores de Temperatura

Los sensores de temperatura son dispositivos que se usan para medir sin problemas en cualquier punto, la temperatura del aire o la temperatura superficial de líquidos y sólidos.

Existen varias formas de medir la temperatura con todo tipo de sensores de diversas naturalezas como: Termopares, Resistivos y Semiconductores, que se utilizan según la aplicación específica que vaya a realizar. La tabla 3 muestra la gran variedad de dispositivos capaces de medir la temperatura.

Tabla 3: Sensores de temperatura en el mercado

NOMBRE	RANGO DE TEMPERATURA	PRECISIÓN	SALIDA	EMPAQUE
Termistor	-45 a 125 °C	±0.3 °C	Analógica	
Lm335az	-40 a 100 °C	10 mv/° K	Analógica	
Lm35	-55 a 150 °C	1 °C	Analógica, salida lineal	
DS18B20	-10 a 85 °C	±0.5°C 9 bits	Digital interface 1 - wire	
TMP100A	-55 a 125 °C	9 a 12 bits	Digital protocolo I2C	

Fuente: Los Autores

### 1.12.2 Sensores de Humedad


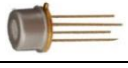


Un sensor de humedad es un dispositivo que mide y regularmente da información de la humedad relativa en un área dada. Pueden ser usados tanto para aplicaciones en interiores como en exteriores.

La Humedad Relativa se expresa en términos porcentuales como la relación entre la presión parcial del vapor de agua contenida en el aire y la presión de saturación del vapor, a una temperatura dada. Cuanto más caliente sea el aire, más humedad tendrá, por lo que la humedad relativa varía con los cambios de temperatura.

El sensor más empleado para medir la humedad es el capacitivo. Este sistema se basa en la capacidad eléctrica o la habilidad de dos conductores eléctricos cercanos para crear un campo magnético entre ellos.

En el mercado existen diferentes tipos de sensores de humedad, en la tabla 4 detallaremos cuatro de ellos.




Tabla 4: Sensores de Humedad en el mercado

NOMBRE	TEMPERATURA DE OPERACIÓN	RANGO DE HUMEDAD %	TIEMPO DE RESPUESTA	EMPAQUE
HS1101	-40 a 100 °C	0 – 100 RH	5s Salida Digital	
HYT 939	-40 a 125 °C	0 – 100 RH	10s Salida Digital	
HIH4000A	-40 a 85 °C	10 – 95 RH	15s Salida Analógica	
HC201	-40 a 110 °C	10 – 95 RH	15s Salida Analógica	

Fuente: Los Autores

Existen sensores que miden dos magnitudes a la vez como es el caso de los presentados en la Tabla 5, que miden la humedad relativa y temperatura.

Tabla 5: Sensores de Humedad y Temperatura

NOMBRE	RANGO DE TEMPERATURA	RANGO DE HUMEDAD	SALIDA	EMPAQUE
HMZ-433A1	0 a 60°C	0 – 95 RH	Lineal DC	
HHT02D	-40 a 123.8 °C	0 – 100 RH	Serial 2-wire	
SHT11	-40 a 123.8 °C	0 – 100 RH	Serial 2-wire	

Fuente: Los Autores

### 1.13 Cable Ethernet Industrial

El cable Ethernet está diseñado específicamente para su uso en entornos industriales exigentes, combinando un cable especialmente diseñado con un conector de construcción robusta para garantizar la fiabilidad y flexibilidad.

El cable es sin blindaje, disponible en TPE<sup>1</sup>, PVC<sup>2</sup> y Plenum<sup>3</sup> para diversas aplicaciones dando una solución de red duradera con conexión segura y protección confiable contra la fatiga mecánica, maximiza la flexibilidad y garantiza la retención confiable de carga (Ver Figura 25).



Figura 25: Conectividad RJ45

Fuente: <http://www.ab.com/en/epub/catalogs/6005557/6005561/10514505/10514628/print.html>

Las especificaciones del cable se pueden ver en la siguiente Tabla 6:

Tabla 6: Especificaciones del cable pre armado Rj45

Certificaciones	UL Listed
Conectores	RJ45 macho a RJ45 macho, Cat 5e
Tipo de cable	No blindado de 4 y 8 conductores, 24 AWG Cat5e Industrial Apantallados de 8 conductores 26 AWG Cat5e industrial
Tipo de envoltente	IP20
Diámetro exterior nominal	5,84 mm (0,23 pulg.)
Temperatura de funcionamiento [C (F)]	-20 ... + 60 ° (-4 ... + 140 °)
voltaje	300V

Fuente: Los Autores

<sup>1</sup> PVC: aplicaciones de uso general.

<sup>2</sup> TPE: aplicaciones de alta flexibilidad.

<sup>3</sup> Plenum: aplicaciones de distribución de aire.

## CAPÍTULO II

### 2. METODOLOGÍA

#### 2.1 TIPO DE ESTUDIO

**De Campo.-** La investigación se centra en realizar un estudio de diversas pruebas en el escenario a emplearse para conseguir información con datos reales.

**Aplicada.-** Busca la aplicación o utilización de los conocimientos aprendidos mediante la investigación, para aplicarlos en el desarrollo del proyecto a desarrollar.

El método a emplear en el proyecto de investigación es el método deductivo, que permite pasar de observaciones de fenómenos generales a hechos particulares contenidas en la situación general.

#### 2.2 POBLACIÓN Y MUESTRA

##### 2.2.1 Población

La población es representada por los reportes diarios que se generan con datos de las variables de Temperatura y Humedad.

##### 2.2.2 Muestra

La muestra está establecida por el cálculo de población infinita, es decir que es desconocida por razón que el número de pruebas puede ser infinito, por consiguiente la muestra es establecida de acuerdo al cálculo con la siguiente formula.

$$n = \frac{Z_{\alpha}^2 \cdot p \cdot q}{i^2}$$

Componentes de la ecuación:

$Z_{\alpha}$  Valor correspondiente a la distribución de gauss,  $Z_{\alpha=0.05} = 1,96$

$p$  Prevalencia esperada del parámetro a evaluar, de un 90%

$q = 1-p$

$i$  Es el error que se prevé cometer, de un 10%

$$n = \frac{(1,96)^2 \cdot (0,9) \cdot (0,1)}{(0,1)^2}$$

$$n = \frac{0,345744}{0,01}$$

$$n = 34,57$$

$$n \approx 35$$

### 2.2.3 Hipótesis

*”El diseño e implementación de un dispositivo electrónico con interfaz de red industrial para integrarse a un sistema SCADA permitiría el control y monitoreo de sensores de humedad y temperatura.”*

Al plantear la hipótesis del trabajo de graduación, está considerado que para mantener y controlar la temperatura y humedad en niveles aceptables se simulara un sistema de climatización mediante extractores de aire consiguiendo de esta manera controlar la temperatura en el interior de la planta.

### 2.3 OPERACIONALIZACIÓN DE VARIABLES

En la Tabla 7 se indica la Operacionalización de variables dependiente e independiente del proyecto. Está desarrollado de una variable independiente y una variable dependiente del tema expuesto, el concepto de cada uno, los indicadores y los ítems que se usó en este trabajo.

Tabla 7: Operacionalización de variables independiente y dependiente

VARIABLES	CONCEPTO	INDICADORES	ÍTEMS
<b>Independiente</b> Dispositivo electrónico para SCADA	Un conjunto de componentes que pueden regular su propia conducta o la de otro sistema con el fin de lograr un funcionamiento predeterminado, de modo que se reduzcan las probabilidades de fallos y se obtengan los resultados buscados.	SCADA  Señal de temperatura  Señal de humedad	Labview  DHT22.  DHT22.
<b>Dependiente</b> Interfaz de red industrial	Es la relación entre la información transmitida a través de una red de comunicaciones y el tiempo empleado para ello.	Tiempo de transmisión	Red Ethernet/IP

Fuente: Los Autores

## 2.4 PROCEDIMIENTOS

Aquí se podrán observar todas las partes involucradas en el desarrollo del dispositivo, al mismo tiempo se explicara de una manera clara y entendible paso a paso su implementación.

- La toma de datos se lo realizara con el sensor DTH22 que tiene la ventaja de ser de tamaño pequeño, bajo consumo de energía y transmisión de señales de larga distancia, hasta 20 metros.
- El procesamiento de las variables obtenidas del sensor en la tarjeta Beaglebone Black Industrial
- Transmisión de datos por Ethernet/IP
- Conversión y comunicación por puerto serie en la tarjeta Arduino UNO
- Monitoreo a través de un software SCADA con una interfaz intuitiva y simple de manejar, que se ejecutará desde un PC de propósito general.

En el siguiente diagrama de bloques (ver Figura 26) se muestra cada paso por el que se mueve la información desde que es tomada de los sensores, hasta que es presentada con la aplicación Labview.

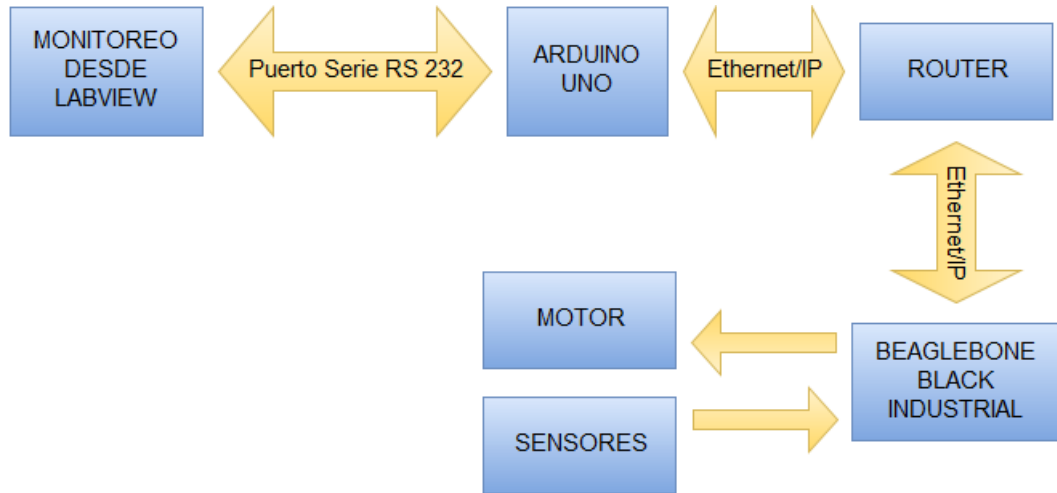


Figura 26: Diagrama de bloques del sistema

Fuente: Los Autores

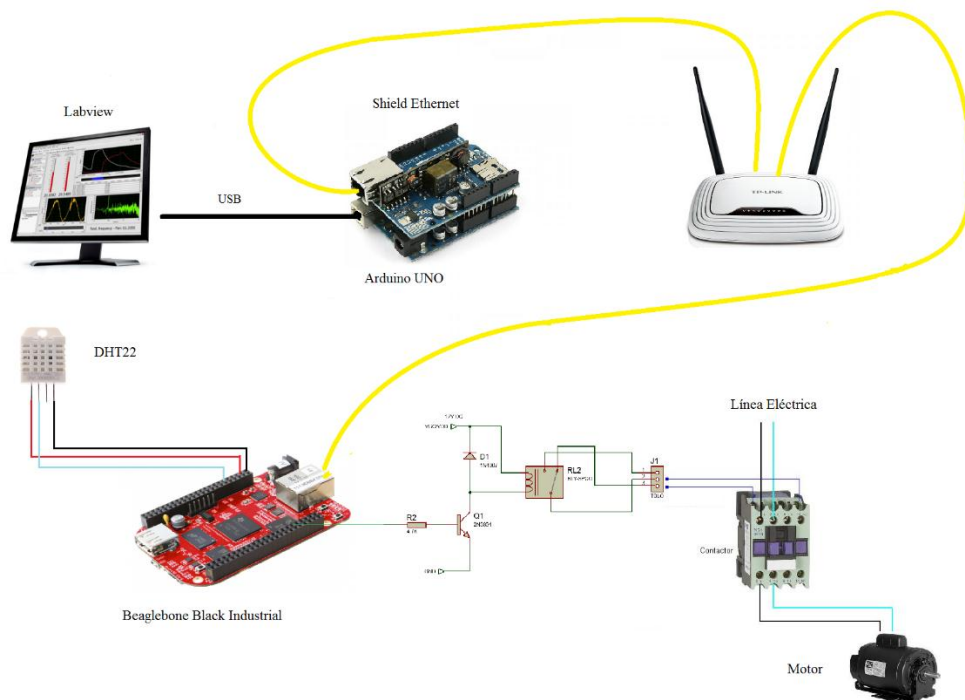


Figura 27: Diagrama de conexión del sistema

Fuente: Los Autores

## 2.5 PROCESAMIENTO Y ANÁLISIS

### 2.5.1 Diseño del controlador programable con Beaglebone Black Industrial

#### 2.5.1.1 Conexión de Beaglebone Black Industrial con el sensor DHT22

Debido a que el voltaje que devuelve el sensor DHT22 es diferente al que recibe la tarjeta Beaglebone Black Industrial, se ha hecho un circuito de acoplamiento que permita interactuar a estos elementos (Ver figura 28).

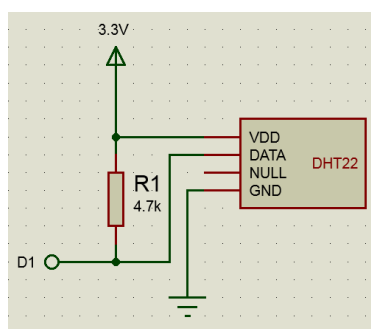


Figura 28: Acondicionamiento de la señal que envía el sensor DHT22

Fuente: Los Autores

El sensor DHT22 usa su propio protocolo para realizar una comunicación bidireccional por medio de un único hilo, el protocolo es simple de usar y se implementa usando un pin de datos, es decir no hay pines de entrada analógica.

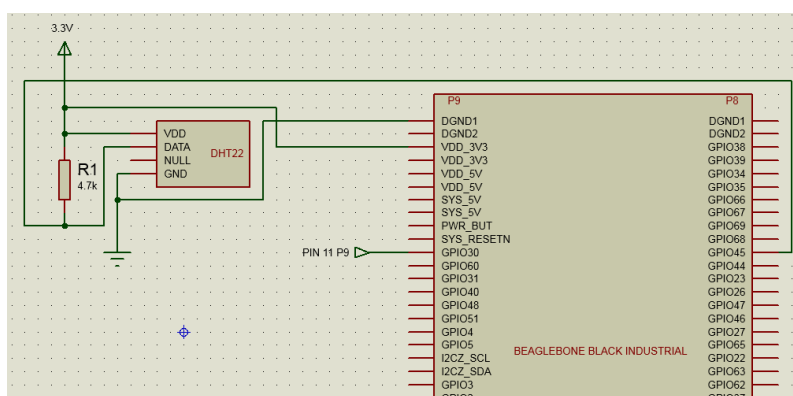


Figura 29: Conexión entre Beaglebone Black Industrial y sensor DHT22

Fuente: Los Autores



En la Figura 29 se muestra la conexión entre la placa Beaglebone Black Industrial y el sensor DHT22 que tiene 3 pines conectados de la siguiente forma: Vdd conectado al pin 3 del Periférico P9, Data conectado al pin 11 del Periférico P8, Gnd conectado al pin 1 del Periférico P9.

### 2.5.1.2 Diseño electrónico del control ON/OFF

El diseño constituye el control de un motor mediante un puerto de la Beaglebone Black Industrial. El sistema consta del SCADA para enviar los datos de activación y desactivación, además de un circuito de control para el accionamiento y manejo de un motor eléctrico, la Figura 30 muestra un diagrama de este proceso.

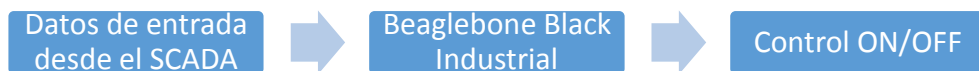


Figura 30: Diagrama de flujo del circuito Control ON/OFF

Fuente: Los Autores

Beaglebone Black Industrial activa y desactiva el motor con un pulso de salida a través del pin 14 del Periférico P9, de esta manera él pulso llega al transistor 2N3904 para conjuntamente con un relé, activar el pulso con el voltaje superior para la activación del motor. El sistema de alimentación del motor es independiente, funciona con un voltaje de alimentación de 12v DC, en la Figura 31 se muestra dicho circuito.

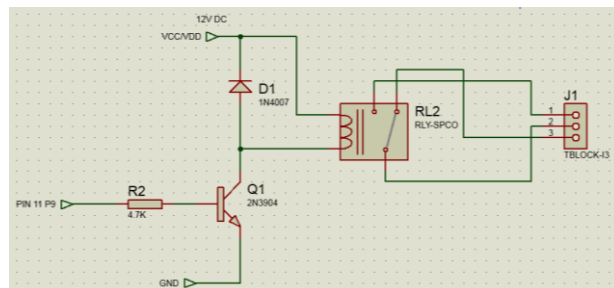


Figura 31: Circuito para la activación y desactivación del motor

Fuente: Los Autores

### 2.5.1.3 Diseño electrónico completo del controlador programable

En la Figura 32 se muestra el esquema completo de la parte actuadora del sistema, incluye la Beaglebone Black Industrial, un circuito ON/OFF del motor y el sensor de humedad y temperatura.

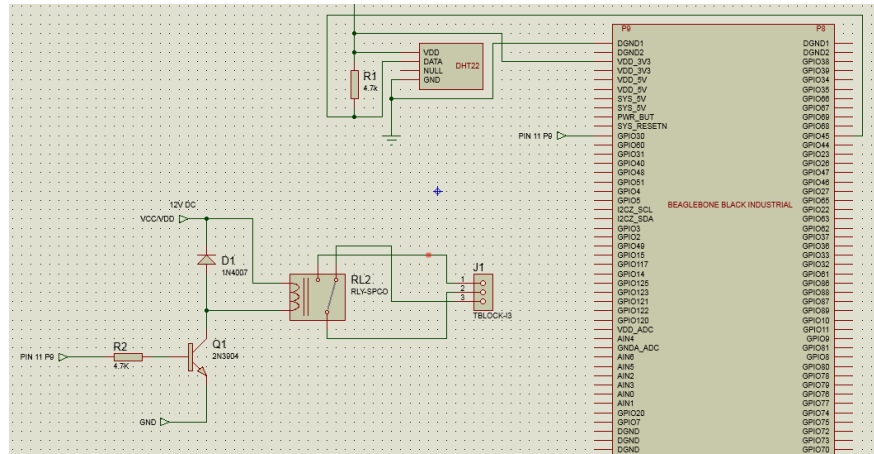


Figura 32: Diagrama completo del sistema

Fuente: Los Autores

### 2.5.1.4 Programación de la tarjeta Beaglebone Black Industrial

Cuando la señal enviada por el sensor ha sido acondicionada a los voltajes adecuados para la tarjeta Beaglebone Black Industrial. Se usará programación en Python para el procesamiento de las variables y luego hacer el envío por el protocolo de red Ethernet/IP, basado en los protocolos estándar TCP/IP.

```
import socket
s = socket.socket() #Instanciamiento para el objeto 's'
host = "            #Host local por defecto
port = 8006         #Puerto
s.bind((host,port)) #Asignación de dirección y puerto a la instancia
s.listen(5)         #Número máximo de conexiones entrantes
```

Figura 33: Código para la creación de un socket

Fuente: Los Autores

Python proporciona niveles de acceso a los servicios de red. Esto puede hacerse utilizando el módulo socket. El mismo que pueden ser configurados para actuar

como un servidor y escuchar los mensajes de entrada, o conectarse a otras aplicaciones como un cliente. Después de que ambos extremos de un socket TCP/IP están conectados, la comunicación es bidireccional. En la Figura 33 se puede observar cómo se crea un socket en Python.

A continuación en la Tabla 8 se presenta una lista de las funciones requeridas para crear un cliente o servidor.

Tabla 8: Funciones para crear cliente o servidor

FUNCIONES	DESCRIPCIÓN
s.bind ()	Este método enlaza la dirección al socket
s.listen ()	Este método establece y comienza a escuchar TCP
s.accept ()	Este aceptar pasivamente la conexión TCP client, esperando hasta que llegue la conexión (bloqueo).
s.connect ()	Este método inicia de forma activa la conexión del servidor TCP.
s.recv ()	Este método recibe el mensaje TCP

Fuente: Los Autores

Una vez que haya Socket, entonces se puede utilizar las funciones necesarias para crear un programa de cliente o servidor. Sabiendo esto, procedemos a escribir un programa cliente/servidor en la Beaglebone Black Industrial que espera la conexión de Arduino al puerto 8006. Una vez obtenida la conexión se procede a la lectura del sensor DHT22 a través del pin 11 digital (GPIO) del Periférico P8 con el comando *read\_retry* quién transmite los datos con una envoltura de TCP/IP por medio de la red. Mientras que la recepción de los niveles ON/OFF del motor se lo realiza con el comando *recv*, si el motor debe activarse, se debe tener un mensaje con la letra “A”, y si se desea apagar el motor se debe tener un mensaje con la letra “a”. En el diagrama de flujo de la Figura 34 se muestra la estructura del código y la programación se encuentra adjunta en el **Anexo 1**.

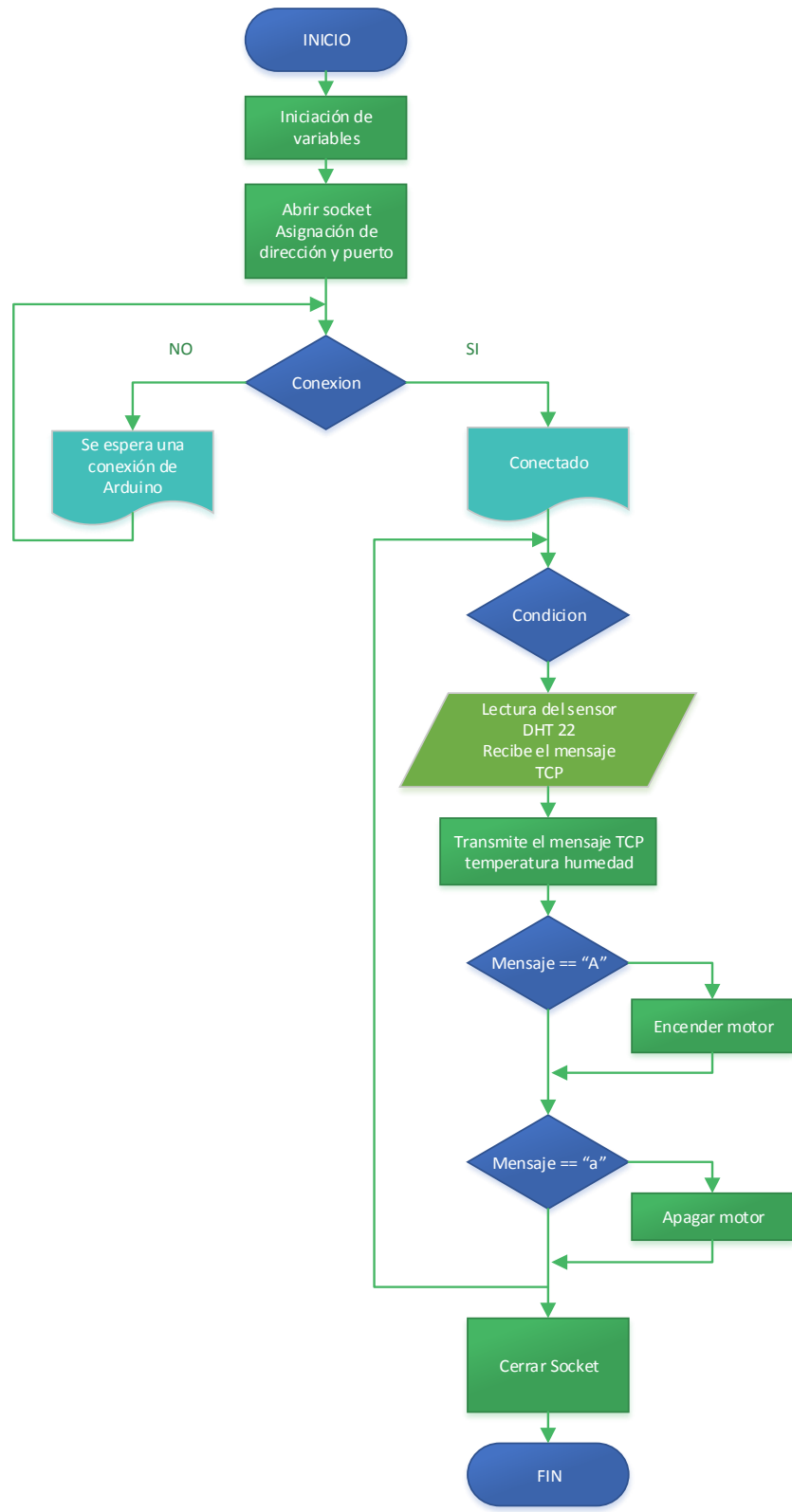


Figura 34: Diagrama de Flujo cliente/servidor de la Beaglebone Black Industrial

Fuente: Los Autores

### 2.5.2 Conexión de Beaglebone Black Industrial con el Arduino UNO

Para este esquema es necesario haber conectado el sensor DTH22 a la Beaglebone Black Industrial, para luego conectar el cable de Ethernet en el puerto Ethernet RJ45 de la misma y el otro extremo a un puerto LAN conectado a internet.

Por otro lado se pone el shield sobre el Arduino UNO y conectar el cable Ethernet al Router TP-LINK TL-WR841ND con Gateway definido. Para que la información pueda transmitirse correctamente será necesario configurar las interfaces de red de ambas placas para que se encuentren en la misma red. El Protocolo de control de transmisión TCP (Transmission Control Protocol) enlaza a los dos dispositivos dentro de la red, para poder enviar y recibir información.

Una vez realizadas las conexiones el resultado del sistema quedaría de la siguiente manera como se puede apreciar en la Figura 35.

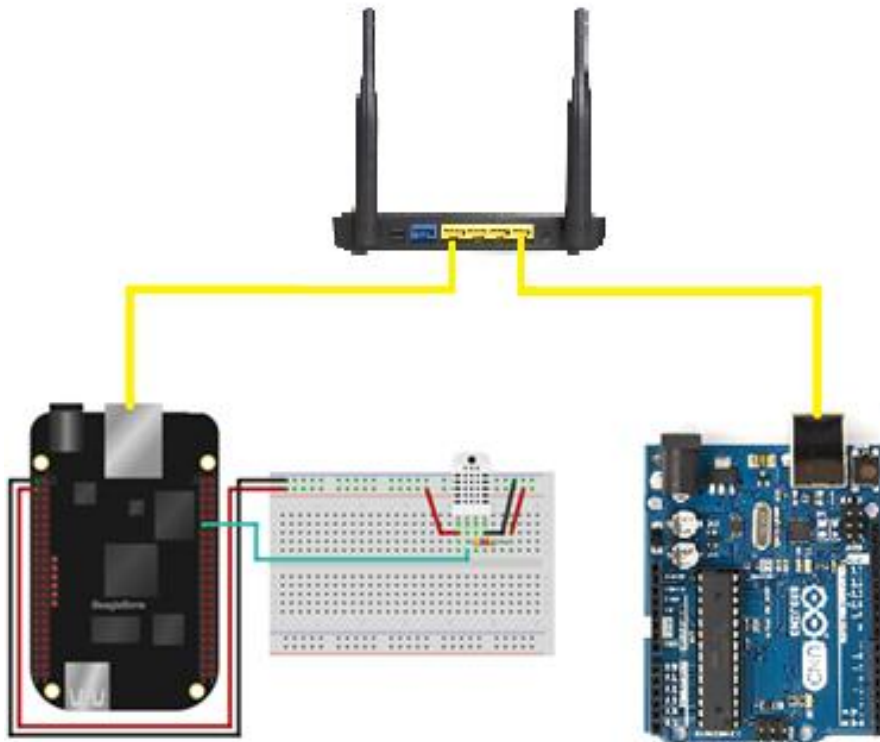


Figura 35: Conexión de Beaglebone Black Industrial con el Arduino UNO

Fuente: Los Autores

### 2.5.3 Programación de la comunicación Ethernet del Arduino Uno

Al momento de configurar definimos la dirección MAC (máscara de red) al Arduino UNO a través de una shield Ethernet. Se asigna la dirección 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED. La dirección MAC cuenta con esta estructura para evitar similitudes con otras direcciones dentro de la tabla de enrutamiento.

La dirección IP del Arduino es la 192.168.0.106 y de la Beaglebone Black Industrial es la 192, 168, 0, 104, estas direcciones figuran dentro del mismo grupo de la red creada por el Router TP-LINK TL-WR841ND con Gateway defino en la dirección IP 192.168.0.101. La información redactada anteriormente se muestra en la Tabla 9.

Tabla 9: Configuración de Red Para el Arduino UNO

<b>ARDUINO UNO</b>	<b>DESCRIPCIÓN</b>
Dirección IP	192.168.0.106
Mascara de Red	0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
Puerto de Comunicación	8006

Fuente: Los Autores

Para establecer una conexión Ethernet con Arduino se abre el entorno de programación y se procede a la importación de la librería Ethernet. Esta librería está diseñada para una Ethernet Shield que está basada en el integrado W5100. Puede servir como un servidor de aceptar conexiones entrantes o un cliente que realiza los salientes. La biblioteca es compatible con hasta cuatro conexiones simultáneas (entrante o saliente o una combinación). Arduino se comunica con la shield Ethernet mediante el bus SPI. Esto es en los pines digitales 11, 12, y 13.

Con el fin de iniciar una conexión como cliente se utiliza EthernetClient() que es el constructor de la clase Client origen de todas las operaciones Ethernet como cliente. Para poder enviar y recibir información es necesario establecer las siguientes configuraciones mostradas en la Tabla 10.

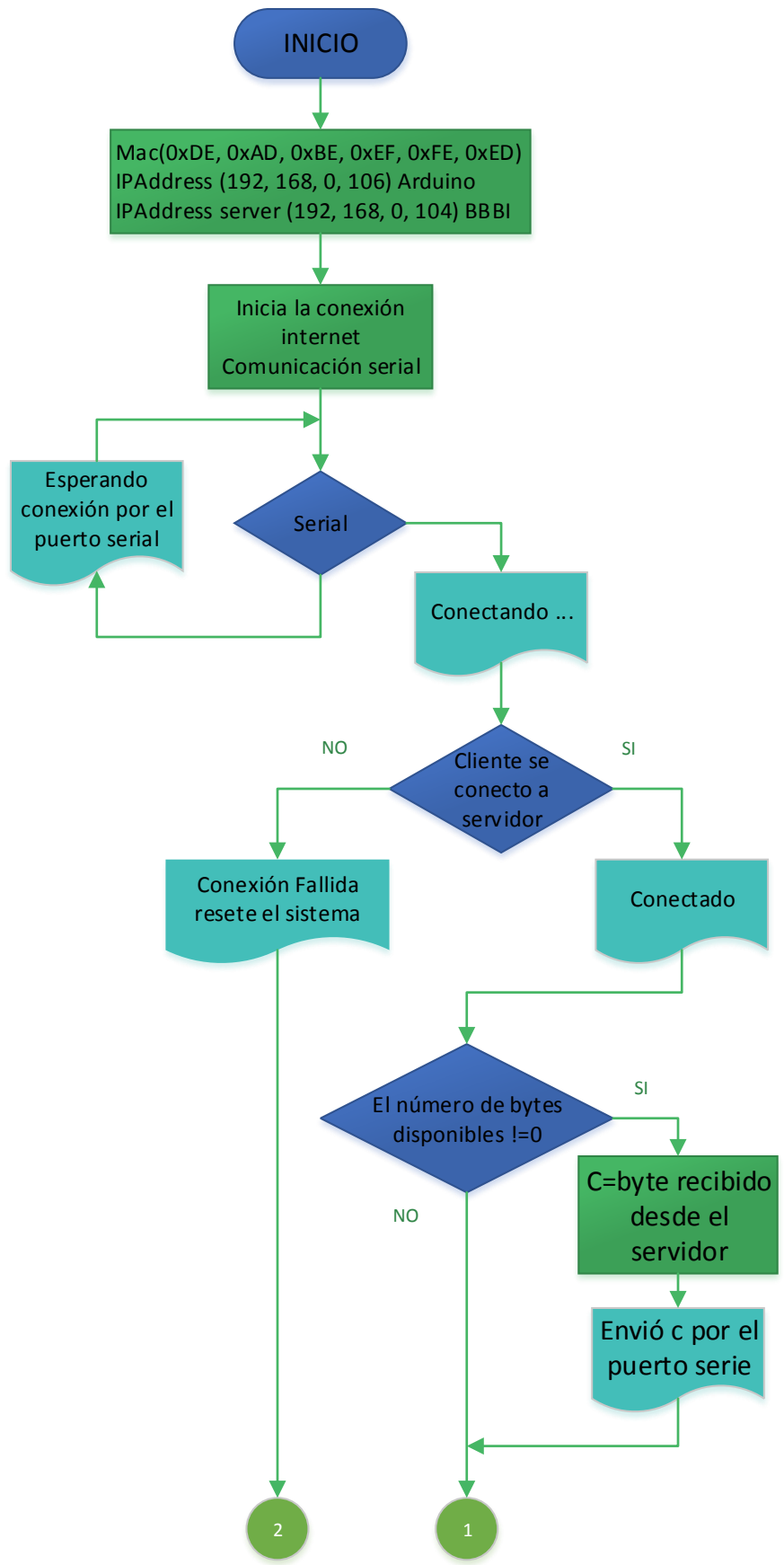
Tabla 10: Configuración cliente en el Arduino UNO

FUNCIONES	DESCRIPCIÓN
EthernetClient()	Crea un cliente que puede conectarse a una dirección IP específica de Internet y puerto
connected()	Informa si el cliente está conectado o no lo está.
connect()	Se conecta a una dirección IP y puerto. El valor de retorno indica el éxito o el fracaso
write()	Escribe datos en el servidor al que el cliente está conectado
print()	Imprime datos en el servidor que está conectado a un cliente.
println()	Imprime datos en el servidor con salto
available()	Devuelve el número de bytes disponibles a leer
flush()	Descarta cualquier byte que se haya escrito para el cliente, pero que aún no ha sido leído.
read()	Lee el siguiente byte recibido desde el servidor al que está conectado el cliente
stop()	Desconecta del servidor

Fuente: Los Autores

Para establecer la comunicación con un ordenador se abre el entorno de programación de Arduino y se procede a la importación de la librería SERIAL en la que se encuentra la función Serial.begin(rate) quien es la encargada de abrir el puerto serie, adicional se configuro la tasa de transmisión para la comunicación serial 9600 baudios para la transmisión de datos. También se usara la función Serial.Read() para leer desde el buffer serie el carácter enviado desde la aplicación de Labview.

El proceso de funcionamiento se ve plasmado en el flujograma de la Figura 36 y la programación se encuentra adjunta en el **Anexo 2**.





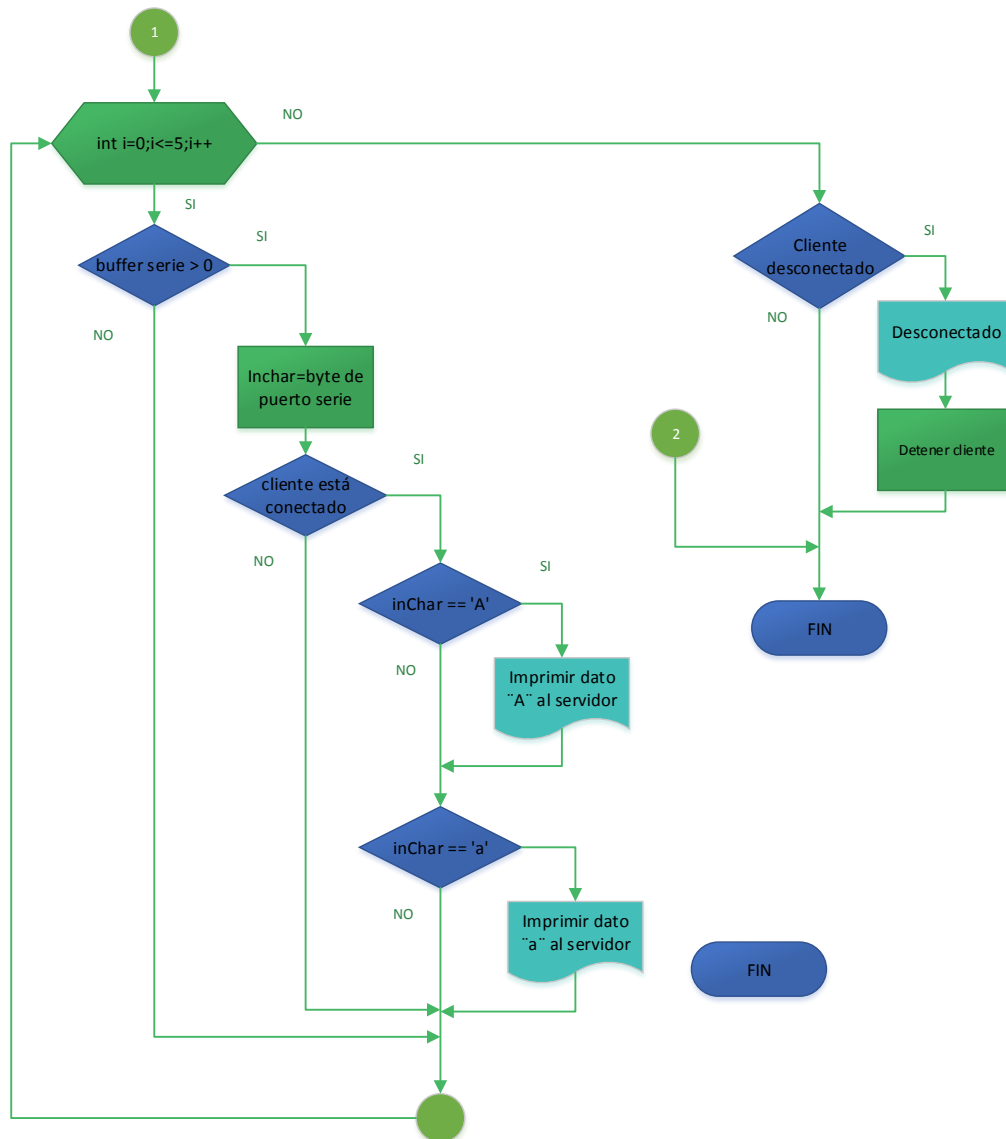


Figura 36: Diagrama de Flujo de la recepción y envío de datos en Arduino

Fuente: Los Autores

#### 2.5.4 Interfaz desarrollada en Labview

Una vez lista la comunicación entre el Arduino y la PC se diseñará el entorno en Labview, la visualización, los datos históricos, alarmas, entre otros, para esto se deberá crear una VI, aquí se realizara el monitoreo en tiempo real del proceso implementado de una manera fácil mediante la configuración de pantallas. Se debe tener en cuenta los siguientes datos para la elaboración de las pantallas:

- Resolución de la pantalla: 1024 \* 1280
- Usuarios: un Operador

En la tabla 11 se especifica las características técnicas del cliente (PC) en la cual se ha instalado el software necesario para el presente proyecto.

Tabla 11: Características técnicas del cliente (PC)

Nombre de equipo	Cliente
Aplicación	Labview Módulo LabVIEW Datalogging and Supervisory Control (DSC)
Sistema Operativo	Windows XP Professional – 32 bits SP3
Memoria RAM	2 GB
Disco Duro	350 GB
Tarjetas de red	Ethernet LAN 10/100 (01)
Ofimática	Microsoft Office 2010 SP2 (Word, Excel, Access)
Browser	Windows Internet Explorer 8

Fuente: Los Autores

Para visualizar los datos medidos y calculados en un solo programa fue necesario unir ambos programas en uno solo. Debido a que Arduino envía ambos datos, temperatura y humedad por una sola vía; Labview se encargaría de pedir en secuencia a Arduino cada uno de los datos para que en el VI sean procesados independientemente. Para lograr esto se utilizó el comando “Stacked Sequence Structure”, el cual permite realizar varias tareas en secuencia. En el estará contemplado todo lo pertinente con la lectura de la temperatura y humedad, además del encendido on/off del motor.

El sistema SCADA consta esencialmente de cuatro pantallas las mismas que surgen de acuerdo a los requerimientos del usuario y a las situaciones en que se encuentra la planta. Así que se tiene las siguientes pantallas:

### 2.5.4.1 Pantalla de presentación

Es la pantalla que aparece al inicio, aquí se puede observar la información relacionada con el proyecto como son: nombre de la institución, la carrera, los autores y el director del proyecto. De la misma se puede acceder a la pantalla principal presionando el botón “Ingresar al Sistema” y otro botón “Salir” con el cual se cerrara el programa de monitoreo y control. Como se muestra en la Figura 37.



Figura 37: Pantalla de presentación en Labview

Fuente: Los Autores

Después del cierre de la ventana de presentación aparece la pantalla de la Figura 38. En la cual se ingresa la contraseña y el usuario del administrador.



Figura 38: Ingreso de contraseña en el programa

Fuente: Los Autores

### 2.5.4.2 Pantalla principal

Es la pantalla de monitoreo en la que el operador puede cumplir actividades de visualización del estado de los sensores de temperatura y humedad, cambio de

parámetros, acceder a las ventanas de gráficos del proceso, visualizar alarmas con la ayuda de luces indicadoras.

Esta pantalla refleja la realidad de la planta industrial, representada con animaciones involucradas con gráficos ilustrativos, por ejemplo animaciones de motores y su encendido/apagado, como se observa en la figura 39.

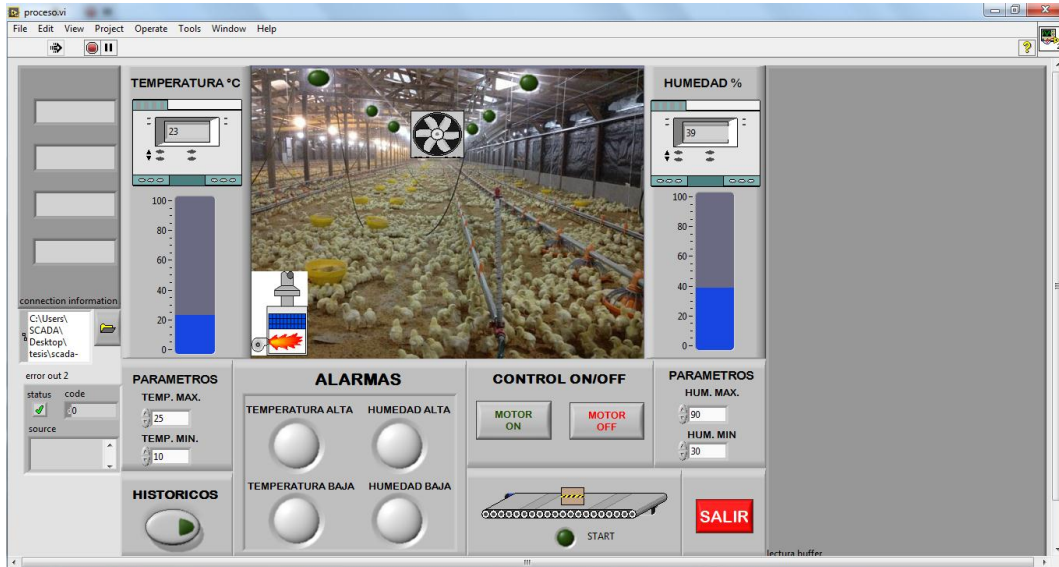


Figura 39: Pantalla principal en Labview

Fuente: Los Autores

#### 2.5.4.2.1 Visualización del estado de sensores

Los valores de temperatura y humedad se muestran de dos formas: en una barra y numéricamente.

#### 2.5.4.2.2 Cambio de parámetros

Se puede modificar los parámetros de temperatura y humedad de acuerdo a los requerimientos de la planta industrial.

#### 2.5.4.2.3 Panel de control

El operador puede activar o desactivar la planta presionando los botones de marcha, paro. En esta ventana también se observa dos botones de acceso, el primero

denominado “Históricos” despliega la pantalla de históricos, el segundo denominado “Salir” nos lleva a la página de inicio de nuestra aplicación.

### 2.5.4.3 Pantalla de Históricos

En esta pantalla se observa en forma gráfica el comportamiento histórico que han tenido las variables críticas del proceso: humedad y temperatura, este registro se hace cada 10 minutos en el día y se guardan como un vector en la base de datos del programa SCADA (Ver Figura 40).

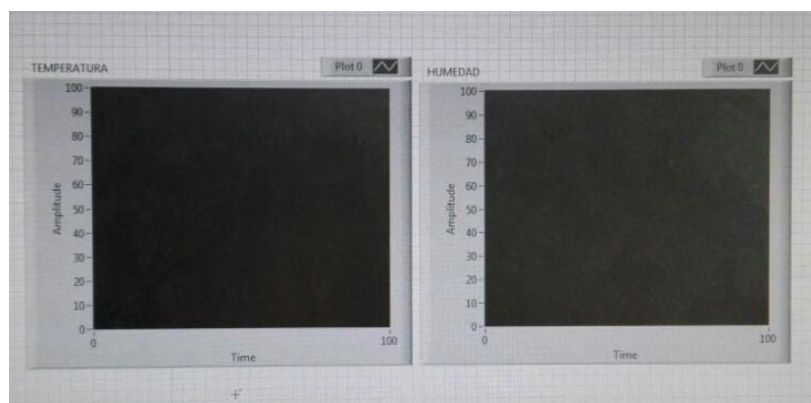


Figura 40: Pantalla de Históricos en Labview

Fuente: Los Autores

## 2.6 COMPROBACIÓN DE LA HIPÓTESIS

En la comprobación de la hipótesis se realizó el análisis mediante la distribución estándar normal de una población infinita o desconocida, y para esto se debe establecer los procedimientos seguidos paso a paso para realizar la comprobación de la hipótesis, mediante la prueba de aseveraciones acerca de una media poblacional o desconocida.

### 2.6.1 Planteamiento de la hipótesis estadística

**H<sub>0</sub>:** El diseño e implementación de un dispositivo electrónico con interfaz de red industrial para integrarse a un sistema SCADA, permitiría el control y monitoreo de sensores de humedad y temperatura.

**H1:** El diseño e implementación de un dispositivo electrónico con interfaz de red industrial para integrarse a un sistema SCADA, no permitiría el control y monitoreo de sensores de humedad y temperatura.

$$H_0: \mu_s \geq \mu_d$$

$$H_1: \mu_s < \mu_d$$

### **2.6.2 Establecimiento del Nivel de significancia**

La prueba se realizó con un 95 % de confiabilidad, es decir con un nivel de significancia de  $\alpha = 0,05$ .

### **2.6.3 Descripción de las poblaciones y suposiciones**

En el cálculo para una muestra de población desconocida, se asume que la población viene de muestras que están debidamente distribuidas y que sus varianzas no son conocidas. Las muestras son tomadas conforme a la formula anteriormente expuesta para conseguir buenos resultados.

### **2.6.4 Determinación del estadístico pertinente**

Si  $H_0$  es verdadera, la media de la distribución muestral es  $\mu_s = \mu_d \geq 0$  y varianza  $\frac{\sigma_s^2}{n_1} - \frac{\sigma_d^2}{n_2}$ , donde  $\sigma_s^2$  y  $\sigma_d^2$  son desconocidas, por lo tanto no se puede calcular la varianza verdadera de la muestra, en consecuencia, excluyo a  $z$  como estadístico de prueba.

### **2.6.5 Determinación del estadístico de prueba**

Se precisa un escenario de pruebas para la comprobación de la hipótesis, que se encuentra configurado por una tarjeta Beaglebone Black Industrial y el sistema de SCADA. En el escenario A se considera el sistema formado por el sensor de temperatura, de humedad relativa, Beaglebone Black industrial, con monitoreo como se muestra en la Tabla 12, para el cual se utilizó el comando ping, para realizar pruebas de latencia y retardo.

Tabla 12: Tiempos de respuesta con monitoreo

NÚMERO DE MUESTRAS	TIEMPO EN MILISEGUNDOS	
	Con Monitoreo	
	Sensor 1	Sensor 2
1	84	75
2	85	136
3	86	77
4	87	78
5	88	79
6	89	80
7	90	81
8	91	142
9	92	83
10	93	71
11	94	85
12	95	86
13	96	147
14	77	88
15	78	89
16	79	90
17	80	91
18	81	92
19	82	93
20	83	94
21	84	95
22	85	96
23	86	157
24	87	130
25	88	159
26	89	129
27	90	161
28	91	162
29	92	131
30	93	164
31	94	165
32	95	129
33	96	167
34	97	121
35	98	134

Fuente: Los Autores

En el escenario B se considera el sistema propuesto sin monitoreo de forma automática como indica la Tabla 13.

Tabla 13: Tiempos de respuesta sin monitoreo

NÚMERO DE MUESTRAS	TIEMPO EN MILISEGUNDOS	
	Sin Monitoreo	
	Sensor 1	Sensor 2
1	75	65
2	76	68
3	77	70
4	78	78
5	79	79
6	80	80
7	81	75
8	82	82
9	83	83
10	71	90
11	85	85
12	86	86
13	87	87
14	88	88
15	89	89
16	90	90
17	91	83
18	92	72
19	93	65
20	94	94
21	95	69
22	96	76
23	97	78
24	70	70
25	99	99
26	69	69
27	101	72
28	102	70
29	71	71
30	104	66
31	105	69
32	69	69
33	107	65
34	61	61
35	69	69

Fuente: Los Autores



Las pruebas tomadas para los dos escenarios fueron mediante la fórmula de muestreo de datos mencionado anteriormente. La siguiente Tabla 14, indica el resumen de las pruebas realizadas con los dos escenarios.

Tabla 14: Media muestral

	Sensor 1	Sensor 2	Proporción Muestral
<b>Ambiente A</b>	88,43	113,06	0,35
<b>Ambiente B</b>	85,49	76,63	0,27

Fuente: Los Autores

En la Tabla 15 se indica la frecuencia esperada.

Tabla 15: Frecuencia esperada

	Sensor 1	Sensor 2
<b>Ambiente A</b>	96,38	105,11
<b>Ambiente B</b>	77,54	84,58

Fuente: Los Autores

En la Tabla 16 se indica los valores calculados de chi-cuadrado.

Tabla 16: Calculo de Chi Cuadrado.

CHI CALCULADO	r	2
	k	2
Grados de Libertad	(r-1)(k-1)	1
Nivel de significación	A	0,05
Chi Cuadrado	x2	3,84
Probabilidad	P	0,093311065
X2Prueba		2,816303582

Fuente: Los Autores

Como  $2.81 < 3.84$  se puede decir que la hipótesis  $H_0$  es verdadera y rechazamos  $H_1$ . En la siguiente Figura se puede observar, el valor crítico de t cae en la zona de aceptación de la hipótesis  $H_0$ .

## CAPÍTULO III

### 3. RESULTADOS

En el presente capítulo se detalla las pruebas realizadas con el proyecto y los resultados derivados durante dichas pruebas.

#### 3.1 Pruebas del sistema

El acondicionamiento de la señal del sensor Dht22 y el circuito para la activación y desactivación del motor fue puesto en una estructura de material PVC, que permite aprovechar la eficacia del sensor y del motor al momento de conectar a la tarjeta Beaglebone Black industrial que tiene una arquitectura Cliente-Servidor.

El Shield de Ethernet apilada encima del Arduino Uno, se encuentra conectado al computador mediante el puerto USB ya que este equipo controlara y monitoreara la humedad y la temperatura.

Un router Tp-Link que se encarga de que las dos tarjetas desarrolladoras se encuentren en la misma red. Es decir, nuestra comunicación con el sistema de adquisición de datos, ha de pasar a través de nuestro router. En internet buscar la configuración del router para poder entrar en él y abrir el puerto para nuestro servidor, de esta forma será accesible desde cualquier sitio de la red.

Las primera prueba del monitoreo fue la activación y desactivación manual del motor, en las siguientes Figura 41 y Figura 42 se muestran la interfaz con la activación manual desde Labview y el motor encendido.

El encendido del motor se encuentra representado por una imagen de un motor y un texto de “Motor”, al presionar el botón ON se activa el motor, al igual, si se presiona el botón OFF el motor se apaga y la imagen cambia de verde a roja.

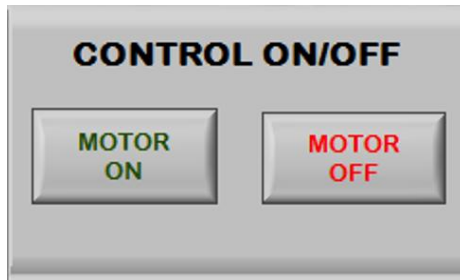


Figura 41: Activación manual del motor

Fuente: Los Autores



Figura 42: Motor Encendido

Fuente: Los Autores

En la siguiente prueba, la aplicación en Labview ha hecho el registro de las mediciones del sensor durante el día, en intervalos de una muestra por minuto, dando un total de 1420 muestras diarias. En la Figura 43 y Figura 44 se muestra los valores ingresados máximos y mínimos para la prueba y los valores reales de temperatura y humedad del medio ambiente, acompañados por una imagen que cambia al signo de exclamación como advertencia en esta prueba, como parte de

una prueba se muestra el tráfico de información del Arduino Uno a través del puerto Serial, en este despliegue de datos se visualizó la lectura de datos, para luego plasmarlos en la programación para su ejecución.



Figura 43: Monitoreo en Labview

Fuente: Los Autores

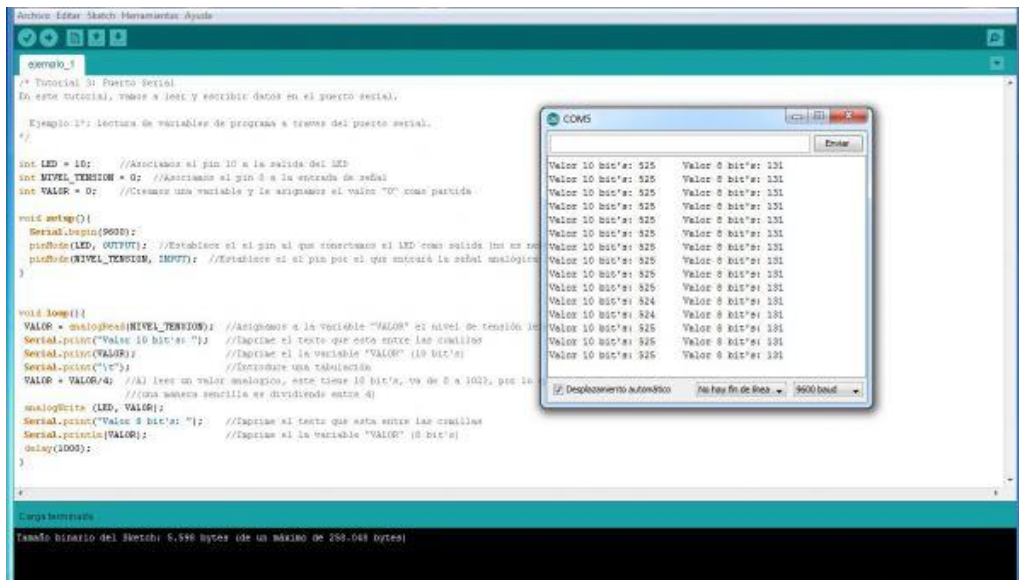


Figura 44: Monitor serial Arduino

Fuente: Los Autores

### 3.2 Resultados del sistema

Debido a la inmensa cantidad de datos que arroja el sistema, se ha calculado el promedio cada 30 valores para luego exportarlos al Excel en donde se tiene realizadas las tablas con fórmulas para que los resultados sean graficados automáticamente.

En la Tabla 17 y en la Tabla 18 se observa los valores y como han variado a lo largo del tiempo durante dos días consecutivos.

Tabla 17: Promedio de muestras tomadas primer día

Hora	Temperatura del ambiente promedio (°C)	Humedad del ambiente promedio (%)
7:00	15	80
7:30	15	76
8:00	16	76
8:30	17	73
9:00	17	71
9:30	18	69
10:00	19	69
10:30	19	69
11:00	18	69
11:30	19	68
12:00	20	66
12:30	20	66
13:00	20	66
13:30	20	66
14:00	20	65
14:30	21	65
15:00	21	65
15:30	21	66
16:00	20	66
16:30	20	67
17:00	20	69
17:30	20	71
18:00	19	72
18:30	19	73
19:00	19	75

Fuente: Los Autores

Tabla 18: Promedio de muestras tomadas segundo día

Hora	Temperatura del ambiente promedio (°C)	Humedad del ambiente promedio (%)
7:00	16	77
7:30	16	77
8:00	16	76
8:30	17	75
9:00	17	75
9:30	17	75
10:00	18	74
10:30	18	71
11:00	18	71
11:30	18	71
12:00	20	70
12:30	20	70
13:00	20	70
13:30	21	70
14:00	21	70
14:30	21	74
15:00	21	74
15:30	21	72
16:00	21	73
16:30	20	73
17:00	20	73
17:30	20	73
18:00	19	74
18:30	19	74
19:00	18	74

Fuente: Los Autores

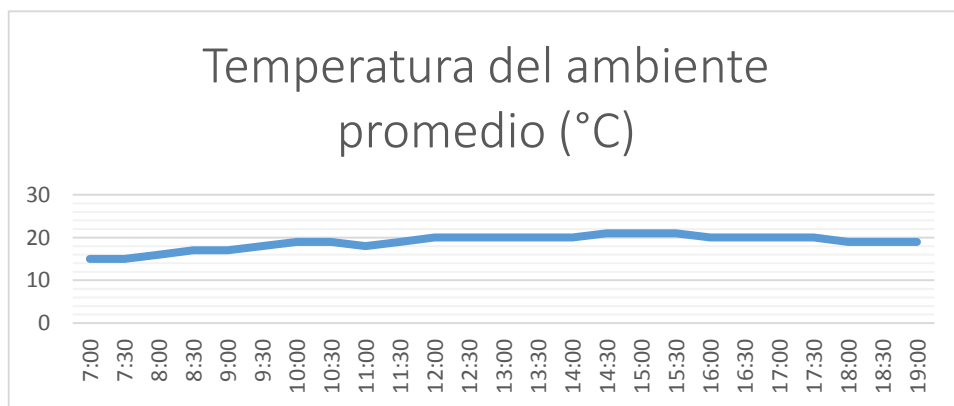


Figura 45: Gráfica del comportamiento de la Temperatura primer día.

Fuente: Los Autores

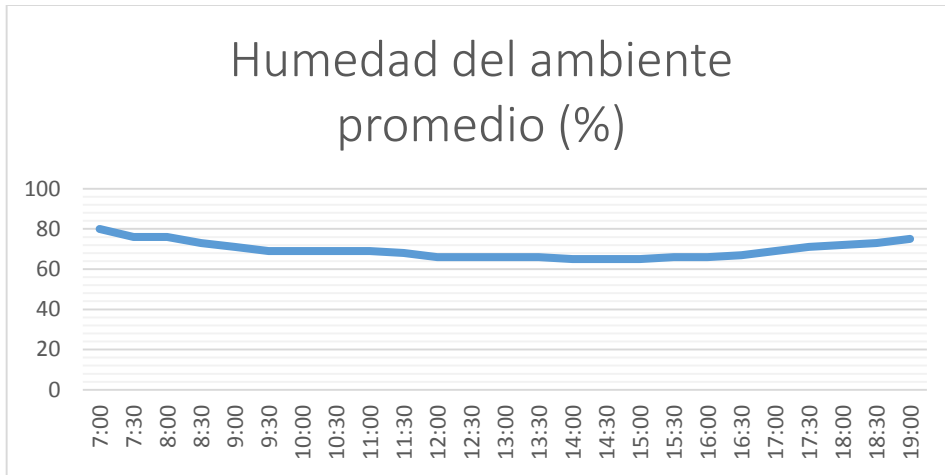


Figura 46: Gráfica del comportamiento de la Humedad primer día.

Fuente: Los Autores

La Figura 45 y Figura 46 muestran la variación de temperatura y humedad para el primer día de pruebas, mientras que en la Figura 47 y Figura 48 muestran los datos para el segundo día. Se puede notar los cambios en las condiciones climáticas al hacer una comparación con las gráficas mostradas para el primer día de pruebas.

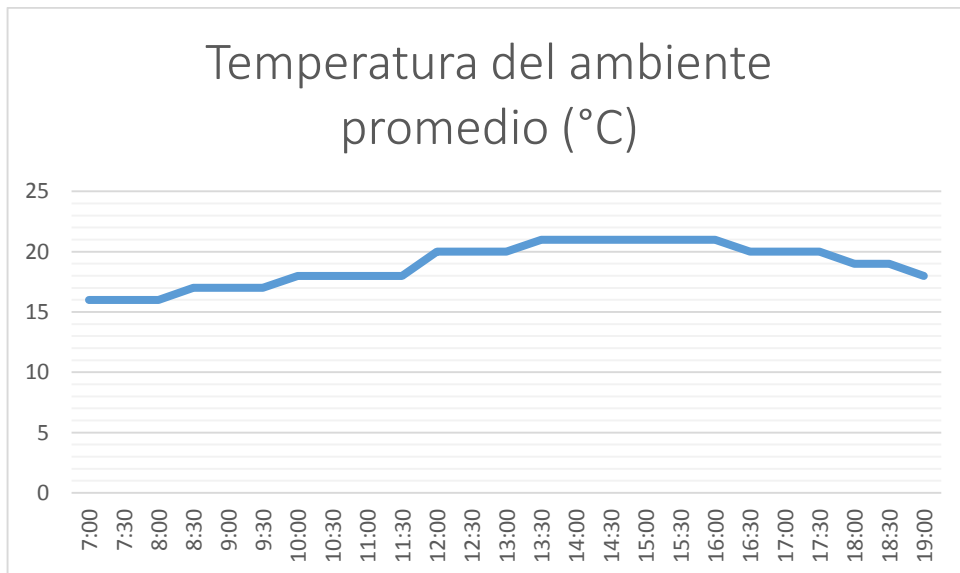


Figura 47: Gráfica del comportamiento de la Temperatura segundo día.

Fuente: Los Autores

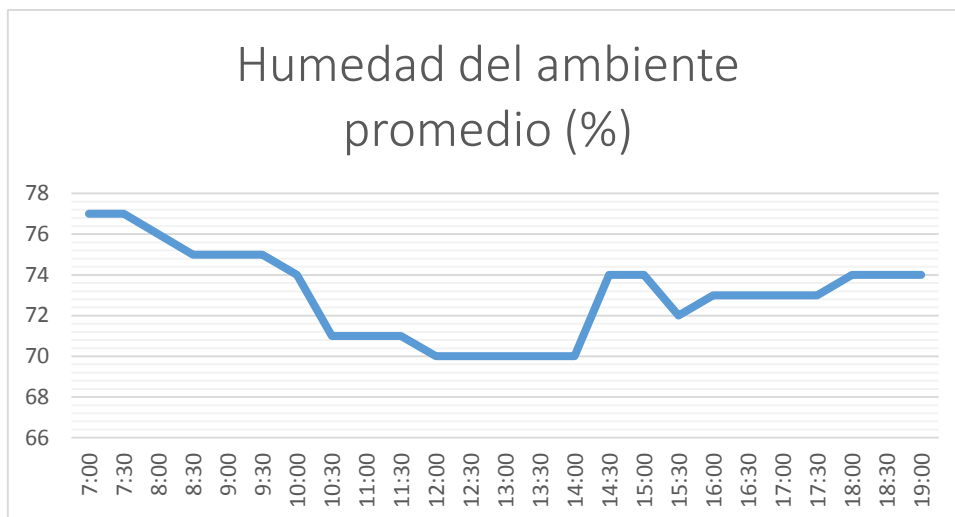


Figura 48: Gráfica del comportamiento de la Humedad segundo día.

Fuente: Los Autores

### 3.3 Análisis financiero

La Tabla 19, muestra el precio de los dispositivos que se usó en el proyecto de diseño, donde los recursos materiales justifican los beneficios obtenidos con la implementación.

Tabla 19: Recursos Materiales

CANT.	DETALLE	COSTO UNITARIO	COSTO TOTAL
1	Arduino Uno R3	\$15.00 USD	\$15.00 USD
1	Router DLINK DIR-600	\$35.00 USD	\$35.00 USD
1	Shield Ethernet Arduino	\$18.00 USD	\$18.00 USD
1	Sensores DHT22	\$20.00 USD	\$40.00 USD
1	Cargador AC a DC 5v	\$5.00 USD	\$5.00 USD
1	Placa recubierta de cobre	\$3.50 USD	\$3.50 USD
2	Resistencia	\$0.10 USD	\$0.20 USD
2	Cables Ethernet Industrial	\$30.00 USD	\$60.00 USD
2	Cajas ENCLOUSER	\$8.00 USD	\$16.00 USD
1	Beaglebone Black Industrial	\$120.00 USD	\$120.00 USD
1	Motor Eléctrico 1hp	\$150.00 USD	\$150.00 USD
12	Cables Dupont	\$0.08 USD	\$1.00 USD
<b>TOTAL</b>			<b>\$462.70 USD</b>

Fuente: Los Autores



## **CAPÍTULO IV**

### **4. DISCUSIÓN**

El desarrollo del prototipo, está basado en la tecnificación del control y monitoreo, a través de un profundo análisis e investigación en las tecnologías proyectadas para el control, se encontró módulos electrónicos adecuados que permiten el desarrollo de un mecanismo estable y un SCADA que permite al usuario un manejo sencillo en un entorno amigable.

Se puso mucho interés en el tipo de comunicación tomando como base de estudio la Red Ethernet/IP. Esta adapta el protocolo industrial común con el Ethernet estándar. Por tanto, se utiliza como una red de nivel de información, para transportar grandes volúmenes de información de forma rápida y segura.

Element14 ha diseñado Beaglebone Black 4G Industrial con las mismas especificaciones que Beaglebone Black Rev C, pero con un PCB de color rojo que soporta una amplia gama de temperaturas entre -20 y 85 °C, y protección del tablero con una película protectora.

La CIP utilizada para el control de los sensores de temperatura y humedad desempeña un esquema lógico, en el cual deberá efectuar cada una de sus etapas desde el reconocimiento de los sensores hasta la comunicación de los datos a través de la red, para así lograr que la información sea eficiente y eficaz, tomando en cuenta cada una de las necesidades del usuario.

La aplicación en Labview está basado en un lenguaje de programación gráfica nativo, el cual permite que el usuario interprete fácilmente el sistema por medio de cuadros de texto, botones, pulsadores, indicadores numéricos y mediante gráficas de historial a través del tiempo el funcionamiento del panel de control del SCADA

Después de concluir con la implementación de los equipos y el software, son capaces de realizar todos los requerimientos planteados anteriormente, proporcionando una comunicación estable, sin pérdida alguna de datos, velocidad de transmisión y con una ejecución exitosa de sus funciones de forma manual y automática.

## **CAPÍTULO V**

### **5. CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones**

- Se ha utilizado y probado la efectividad de la tarjeta Beaglebone Black Industrial como un controlador lógico programable que tiene una capacidad de integración y escalabilidad sobre entornos industriales.
- El sistema operativo Debian de tiempo real que se instaló en la tarjeta Beaglebone Black Industrial influyo de manera significativa en el desarrollo de la aplicación que depende de una respuesta rápida ante eventos externos.
- La implementación de una red industrial utilizando el protocolo Ethernet/IP tiene las características necesarias para monitorear y controlar sensores de temperatura y humedad, al tener todos los equipos de una red la posibilidad de ser clientes, servidores o servidores y clientes simultáneamente.
- El software Labview se muestra como una opción eficiente para desarrollar aplicaciones de supervisión y control de forma sencilla, proporcionando una infinidad de herramientas que pueden ser utilizadas para automatizar cualquier proceso industrial.
- Se puede hacer una red Ethernet/IP confiable a un precio realmente bajo, si se compara este último con el costo que tiene que pagar para ensamblar una red con un PLC idéntica a la realizada.

#### **5.2 Recomendaciones**

- Utilizar una micro-SD, de una capacidad mayor o igual a 8 GB de alta velocidad de transferencia de datos, para de este modo mejorar el rendimiento del sistema operativo que se desea implementar en la tarjeta Beaglebone Black Industrial.

- Manejar las protecciones debidas en los puertos de propósito general puesto que al momento de ejecutar la aplicación, las entradas solo soportan tensiones de 3.3 V. y al integrar un valor de 1 a 5 voltios se puede quemar los puertos y la tarjeta.
- Para mejorar el tiempo de respuesta de temperatura utilizar el sensor digital SHT15 que tiene un tiempo de respuesta muy rápido una muestra por segundo en comparación con el DHT22 un sensor lento que logra una muestra cada dos segundos.
- Para el voltaje de alimentación de la tarjeta Beaglebone Black Industrial, es recomendable que el voltaje que entre no supere los 5VDC a 2A, el daño no solo puede afectar la comunicación sino a toda la tarjeta.

## **CAPÍTULO VI**

### **6. PROPUESTA**

#### **6.1 Título de la Propuesta**

Diseño e implementación de un servidor web embebido para el control y monitoreo Industrial utilizando una tarjeta de desarrollo Beaglebone Black Industrial.

#### **6.2 Introducción**

Un sistema embebido es un sistema informático diseñado para funciones de control específicas dentro de un sistema más grande, a menudo con limitaciones de tiempo real .Cuando los dispositivos integrados cuentan con acceso a internet, es sin duda de que la demanda aumentará debido a la capacidad de acceso remoto de los dispositivos.

En este proyecto se combinan los sistemas integrados y la tecnología de Internet para formar una nueva tecnología – la Tecnología de Internet embebida, que se desarrolló con la popularización de la tecnología de redes informáticas en los últimos años. El corazón de la comunicación es el protocolo TCP / IP. La comunicación de red se realiza por la norma internacional IEEE 802.3. Es la más moderna tecnología de sistemas embebidos.

El servidor web embebido montado en Beaglebone Black tendrá una capacidad de ejecución rápida y un protocolo estándar Ethernet industrial que podrá proporcionar acceso a Internet a una velocidad razonable, este sistema es adecuado para mejorar

la seguridad en condiciones industriales mediante el control de forma remota de diversos procesos industriales.

### **6.3 Objetivos**

#### **6.3.1 General**

Diseño e implementación de un servidor web embebido para el control y monitoreo Industrial utilizando una tarjeta de desarrollo Beaglebone Black Industrial.

#### **6.3.2 Específicos**

- Desarrollar el programa de control en la tarjeta de desarrollo Beaglebone Black Industrial para la adquisición de datos.
- Desarrollar la interfaz de comunicación vía protocolo Ethernet para el sistema.
- Monitorear a través de una aplicación web los valores de las variables y en tiempo real que se guardan en la base de datos.
- Investigar y determinar los módulos de comunicación inalámbrica que permitan el envío de datos desde la tarjeta hasta la base de datos.

### **6.4 Fundamentación científico –técnica**

Para el desarrollo del servidor web embebido para el control y monitoreo Industrial, se considera distintos componentes que forman parte de la implementación y ejecución, los cuales serán especificados a continuación:

**Protocolo TCP / IP:** Describe la estructura en capas de servidor web embebido permitiendo la comunicación entre los equipos de diferente configuración. Así que con el uso del protocolo TCP / IP se puede acceder al servidor web en el PC remoto a través de Internet. El navegador y el servidor web se comunican a través del protocolo HTTP.

**El Beaglebone Black Industrial:** Es la última adición a la familia BeagleBoard.org y al igual que sus predecesores, es un procesador basado en ARM de bajo coste de la corteza A8. Se ha equipado con un conjunto mínimo de funciones para que el usuario pueda experimentar el poder del procesador y no pretende ser una plataforma de desarrollo completa ya que muchas de las características y las interfaces proporcionadas por el procesador no son accesibles desde el BEAGLEBONE sino a través del apoyo de algunas interfaces.

**El sensor digital DHT22:** Tiene la ventaja de ser de tamaño pequeño, bajo consumo de energía y transmisión de señales de larga distancia, hasta 20 metros. La temperatura es registrada en grados Celsius y la humedad en porcentaje de la relación entre el vapor de agua y el aire seco.

**El Sensor de gas -MQ2:** Es un sensor de monóxido de carbono (CO), apto para detectar concentraciones de monóxido de carbono (ppm) en el aire. El sensor MQ-2 puede medir concentraciones de CO que van del 20 al sensor de 2000ppm. Este tiene una alta sensibilidad y tiempo de respuesta rápido.

## 6.5 Descripción de la Propuesta

La aplicación de la tecnología de Internet embebido se consigue por medio del servidor Web incorporado. Se ejecuta en el sistema embebido con la limitación de los recursos de computación para mostrar documentos web, incluyendo información estática y dinámica sobre el sistema integrado en el navegador web. Podemos conectar cualquier dispositivo/equipo electrónico al servidor web y se puede obtener la información del estado en tiempo real y controlar el equipo remotamente sin restricción de tiempo y espacio a través de la página web lanzada por el servidor Web embebido.

La arquitectura de monitoreo industrial y sistema de control consta de dos módulos. En el primer módulo sensores, el nodo de dos sensores se desarrolla es decir, temperatura, humedad y gas, respectivamente, que medirá cantidades físicas de los dispositivos industriales. Los canales AD0, AD1, AD2 de la tarjeta Beaglebone

Black industrial se utilizan para conectar sensores MQ2, DHT22, MQ6 respectivamente. Después el módulo servidor web embebido es la PC del cliente con la interfaz gráfica de usuario GUI (Graphical User Interface) para el seguimiento de los parámetros industriales en la ubicación remota. La interfaz gráfica de usuario en el PC remoto consta de las siguientes fichas para el seguimiento y control:

- Relé 1 (ON / OFF)
- Relé 2 (ON / OFF)
- Valor de temperatura
- Valor del gas CO2
- Valor de gas metano

## 6.6 Diseño Organizacional

En la Figura 54. Se muestra, el diseño organizacional de los individuos que intervendrán en el desarrollo y la implementación de la propuesta.



Figura 49: Diagrama del sistema Organizacional del Proyecto

Fuente: Los Autores

## 6.7 Monitoreo y Evaluación de la propuesta

La implementación del servidor Web usando la tarjeta Beaglebone Black Industrial para la monitorización inteligente es un nuevo método para monitorear



un entorno industrial que se lo diseño para ser ejecutado en tiempo real. El sistema también se podrá comunicar con la PC a través del puerto serie RS-232.

Es compatible con la supervisión y el control en línea no sólo dentro de la red privada (LAN), sino también en la red pública (Internet). Todo el sistema tendrá buena apertura debido a su bajo costo, portabilidad y fácil de mantener y actualizar.

Es posible interconectar diferentes tipos de sensores con estos módulos y hacer diversas aplicaciones. Por lo tanto, puedo monitorear el estado de operación de mi sistema embebido a través de Internet, logrando el monitoreo de redes propuesto.

Este prototipo se puede ampliar con el uso de servidores integrados de gama alta, junto con las redes de sensores inalámbricos con un aumento en los parámetros y un aumento de los nodos de sensores.

## CAPÍTULO VII

### 7. BIBLIOGRAFÍA

Arduino, S. (26 de Julio de 2016). Servidor OPC libre para Arduino Obtenido de: <http://blog.bricogeek.com/noticias/arduino/servidor-opc-libre-para-arduino/#more>

Arduino, H. (22 de Marzo de 2016). Arduino UNO & Genuino UNO Obtenido de: <https://www.arduino.cc/en/Main/ArduinoBoardUno>

Arduino, H. (5 de Abril de 2016). Arduino Ethernet Shield Obtenido de: <https://www.arduino.cc/en/Main/ArduinoEthernetShield>

Debian (25 de abril de 2015). Debian.org Obtenido de: <https://www.debian.org/News/2015/20150426>

Upm (12 de agosto del 2013). Upm.es Obtenido de: [http://oa.upm.es/22322/1/PFC\\_ALBERTO\\_CASTRO\\_DOMINGUEZ.pdf](http://oa.upm.es/22322/1/PFC_ALBERTO_CASTRO_DOMINGUEZ.pdf)

Hertzog, R., Mas, R. (2015). The Debian Administrator's Handbook. Recuperado de: <https://debian-handbook.info/download/es-ES/stable/debian-handbook.pdf>

El equipo del instalador de Debian (2013). Guía de Instalación de Debian GNU/Linux, Recuperado de: <https://www.debian.org/releases/wheezy/i386/install.pdf.es>

Lajara, J., (2007), LabVIEW: Entorno gráfico de programación, México DF, México: Editorial Marcombo.

National Instruments Corporation (2000). LabVIEW™ Basics II Course Manual. Recuperado de: <http://stu.edu.vn/uploads/documents/030409-204759.pdf>

Carlos, C. (2009). Sistema SCADA, Recuperado de: <http://www.generatetecnologias.es/index.html.pdf>

Corrales, L., (2007), Interfaces de Comunicación Industrial, Dpto. De automatización Industrial,

Design, C., (5 de Octubre de 2015). Element14 BeagleBone Black Industrial 4G. Obtenido de: <https://www.element14.com/community/docs/DOC-78671/1/element14-beaglebone-black-industrial-4g>

Element14 (2016), Element14 BeagleBone Black Industrial System Reference Manual Recuperado de: [https://www.element14.com/community/servlet/JiveServlet/previewBody/80255-102-1-334379/BBBI\\_SRM\\_Rev%201.0%20VL.pdf](https://www.element14.com/community/servlet/JiveServlet/previewBody/80255-102-1-334379/BBBI_SRM_Rev%201.0%20VL.pdf)

Molloy, D., (2015), Exploring Beaglebone, Indianapolis, Estados Unidos de America: Editoreal Sons. Ins.

Lumme, J., (2013), Beaglebone Home Automation, Birmingham, Estados Unidos de America: Editoreal Packt Publishing

## CAPÍTULO VIII

### 8. ANEXOS

#### 8.1 Anexo 1 Programación de la BBBI

```
#!/usr/bin/python

import Adafruit_DHT
import Adafruit_BBIO.GPIO as GPIO
import socket
import time
var=1
sensor = Adafruit_DHT.DHT22
pin = 'P8_11'
led = "P9_14"

#initialize the pin as an OUTPUT
GPIO.setup(led, GPIO.OUT)

s = socket.socket()

host = ''
port = 8006

backlog = 5

s.bind((host,port))
s.listen(backlog)

print "Se espera una conexion de arduino"
s, (host,port) = s.accept()
#s, (host,port)
#s.accept()
print"conectado"
# Try to grab a sensor reading. Use the read_retry method which
will retry up
# to 15 times to get a sensor reading (waiting 2 seconds between
each retry).

#humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

# Note that sometimes you won't get a reading and
# the results will be null (because Linux can't
# guarantee the timing of calls to read the sensor).
# If this happens try again!

while True:
```

```

        humidity, temperature =
Adafruit_DHT.read_retry(sensor, pin)

        data = s.recv(2)
        print(data)
        #if not data: break
        #s.send("datos")

        #s.send( "H=")
        #s.send(str(humidity) )
        #s.send("    T=")
        #s.send(str(temperature) )

        s.send('Temp={0:0.1f}*C
Humidity={1:0.1f}%'.format(temperature, humidity))
        s.send('\n')

        #recibido = s.recv(1024)
        #print((recibido))

        print('Temp={0:0.1f}*C
Humidity={1:0.1f}%'.format(temperature, humidity))

        if data == 'A' :
                                GPIO.output(led, GPIO.HIGH)

        elif data == 'a':
                                GPIO.output(led, GPIO.LOW)
                                #time.sleep(1)

#s.send( "P=")
#s.send(str(humidity) )

#s.send("T=")
#s.send(str(temperature) )
s.close()

```

## 8.2 Anexo 2 Programación del Arduino UNO

```
#include <SPI.h>

#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.

// The IP address will be dependent on your local network:

byte mac[] = {

    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

IPAddress ip(192, 168, 0, 106);

// Enter the IP address of the server you're connecting to:

IPAddress server(192, 168, 0, 104); //beagle bone red

EthernetClient client;

void setup() {

    // start the Ethernet connection:

    Ethernet.begin(mac, ip);

    // Open serial communications and wait for port to open:

    Serial.begin(9600);

    while (!Serial) {

        ; // wait for serial port to connect. Needed for Leonardo only

    }

    // give the Ethernet shield a second to initialize:

    delay(1000);
```

```

Serial.println("Conectando...");

// if you get a connection, report back via serial:

if (client.connect(server, 8006)) {

    Serial.println("Conectado");

}

else {

    // if you didn't get a connection to the server:

    Serial.println("Conexion fallida... resetea el sistema");

}

}

void loop()

{

    // if there are incoming bytes available

    // from the server, read them and print them:

    if (client.available()) {

        char c = client.read();

        Serial.print(c);

        delay(5);

    } if(client.available()<=0){

        client.print(' ');

        delay(3000);

```

```

}

// as long as there are bytes in the serial queue,

// read them and send them out the socket if it's open:

for (int i=0;i<=5;i++){

if(Serial.available() > 0) {

char inChar = Serial.read();

if (client.connected()) {

if(inChar == 'A'){

client.print('A');

}

if(inChar == 'a'){

client.print('a');

}

}

}

//delay(10);

}

// if(Serial.available() <= 0){

// client.print(' ');

// delay(15);

//}

```



```
// if the server's disconnected, stop the client:  
  
if (!client.connected()) {  
  
    Serial.println();  
  
    Serial.println("disconnecting.");  
  
    client.stop();  
  
    // do nothing:  
  
    while (true);  
  
}  
  
//client.println('\n');  
  
//delay(10);  
  
}
```

### 8.3 Anexo 3 Hoja de datos Sensor DTH22

## Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

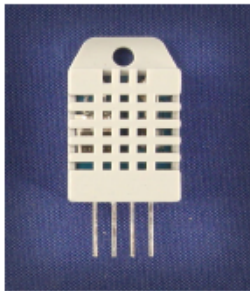
<http://www.aosong.com>

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

### Digital-output relative humidity & temperature sensor/module

#### AM2303



Capacitive-type humidity and temperature module/sensor

#### 1. Feature & Application:

- \* Full range temperature compensated
- \* Relative humidity and temperature measurement
- \* Calibrated digital signal
- \* Outstanding long-term stability
- \* Extra components not needed
- \* Long transmission distance
- \* Low power consumption
- \* 4 pins packaged and fully interchangeable

#### 2. Description:

AM2303 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable AM2303 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

#### 3. Technical Specification:

Model	AM2303
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer humidity capacitor & DS18B20 for detecting temperature
Measuring range	humidity 0-100%RH; temperature -40~125Celsius

# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

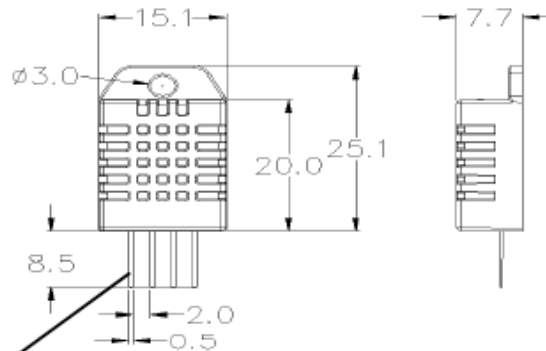
<http://www.aosong.com>

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

Accuracy	humidity $\pm 2\%RH$ (Max $+5\%RH$ );	temperature $\pm 0.2$ Celsius
Resolution or sensitivity	humidity $0.1\%RH$ ;	temperature $0.1$ Celsius
Repeatability	humidity $\pm 1\%RH$ ;	temperature $\pm 0.2$ Celsius
Humidity hysteresis	$\pm 0.3\%RH$	
Long-term Stability	$\pm 0.5\%RH/year$	
Sensing period	Average: 2s	
Interchangeability	fully interchangeable	

## 4. Dimensions: (unit---mm)



Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD—power supply
2	DATA—signal
3	NULL
4	GND

## 5. Operating specifications:

### (1) Power and Pins

Power's voltage should be 3.3-6V DC. When power is supplied to sensor, don't send any instruction to the sensor within one second to pass unstable status. One capacitor valued 100nF can be added between VDD and GND for wave filtering.

### (2) Communication and signal

Single-bus data is used for communication between MCU and AM2303, it costs 5mS for single time communication.

# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

http://www.aosong.com

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

Data is comprised of integral and decimal part, the following is the formula for data.

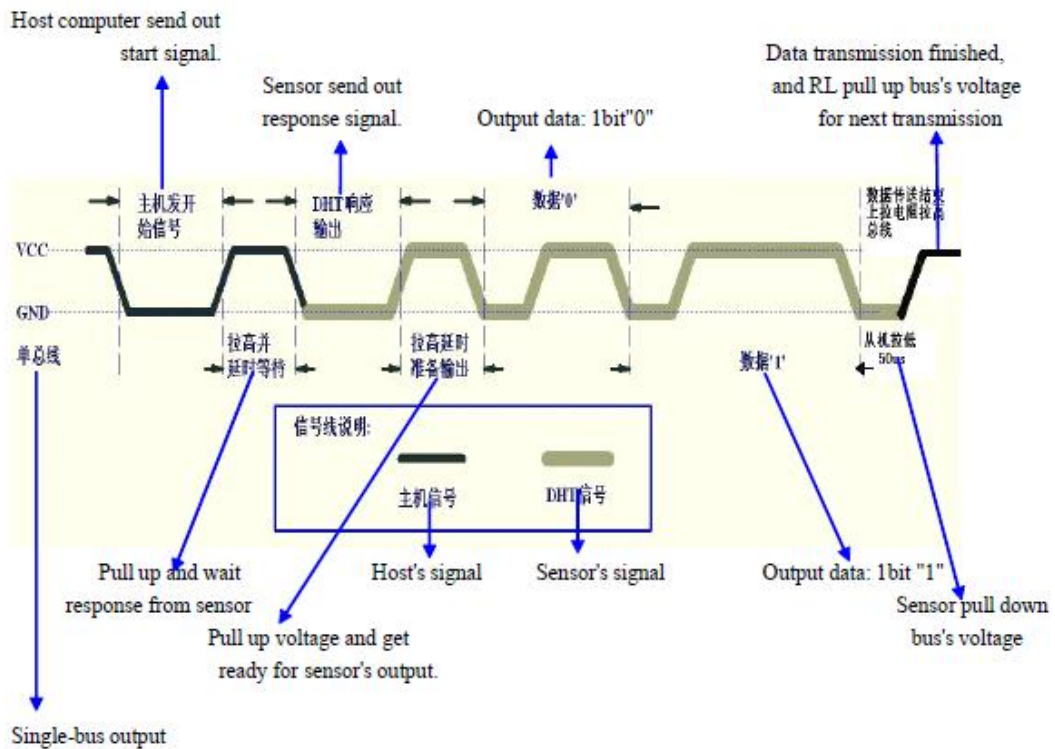
AM2303 send out higher data bit firstly!

DATA=8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data+8 bit check-sum

If the data transmission is right, check-sum should be the last 8 bit of "8 bit integral RH data+8 bit decimal RH data+8 bit integral T data+8 bit decimal T data".

When MCU send start signal, AM2303 change from low-power-consumption-mode to running-mode. When MCU finishes sending the start signal, AM2303 will send response signal of 40-bit data that reflect the relative humidity and temperature information to MCU. Without start signal from MCU, AM2303 will not give response signal to MCU. One start signal for one time's response data that reflect the relative humidity and temperature information from AM2303. AM2303 will change to low-power-consumption-mode when data collecting finish if it don't receive start signal from MCU again.

1) Check bellow picture for overall communication process:



# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

<http://www.aosong.com>

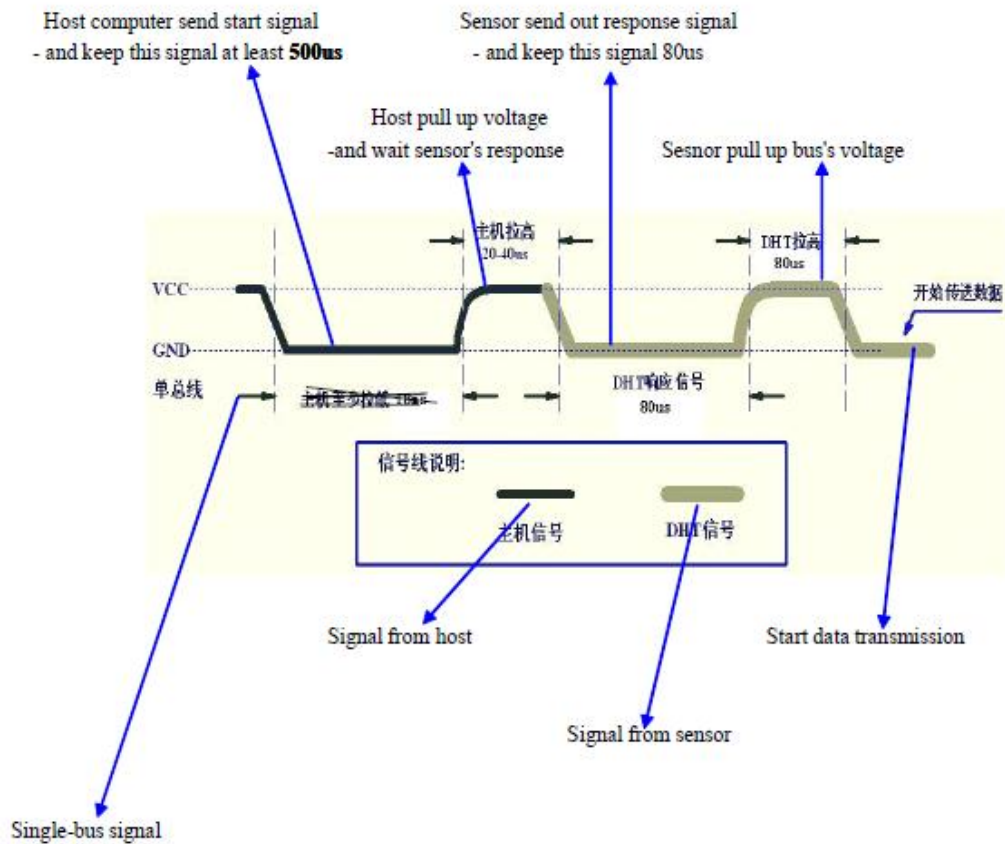
Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

## 2) Step 1: MCU send out start signal to AM2303

Data-bus's free status is high voltage level. When communication between MCU and AM2303 begin, program of MCU will transform data-bus's voltage level from high to low level and this process must beyond at least 18ms to ensure AM2303 could detect MCU's signal, then MCU will wait 20-40us for AM2303's response.

Check bellow picture for step 1:



## Step 2: AM2303 send response signal to MCU

When AM2303 detect the start signal, AM2303 will send out low-voltage-level signal and this signal last 80us as response signal, then program of AM2303 transform data-bus's voltage level from low to high level and last 80us for AM2303's preparation to send data.

Check bellow picture for step 2:

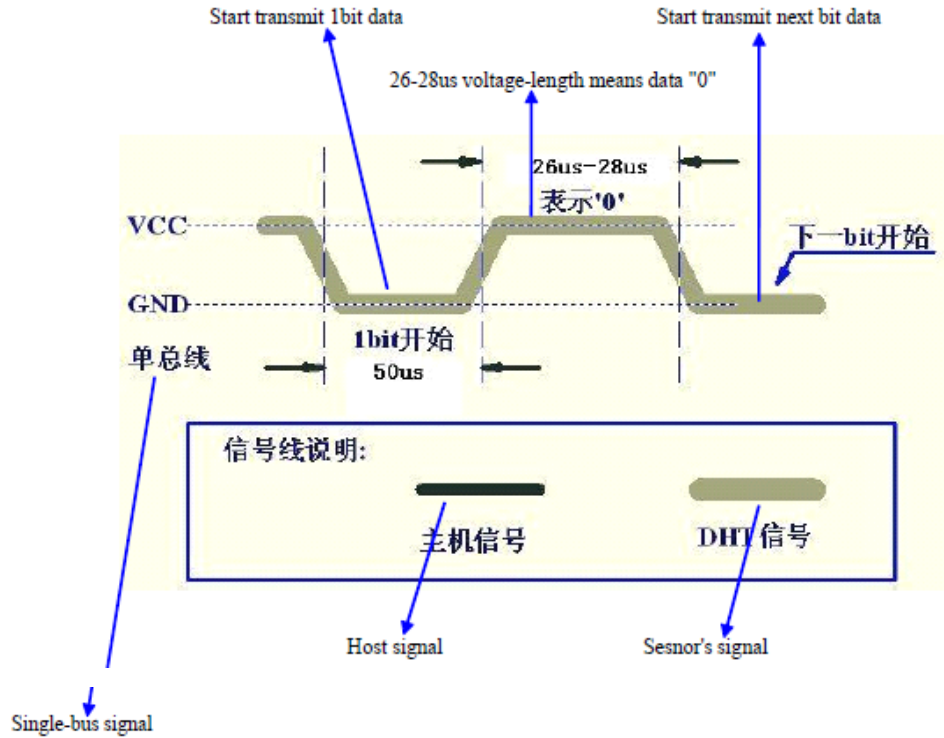
# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

<http://www.aosong.com>

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China



## Step 3: AM2303 send data to MCU

When AM2303 is sending data to MCU, every bit's transmission begin with low-voltage-level that last 50us, the following high-voltage-level signal's length decide the bit is "1" or "0".

Check bellow picture for step 3:



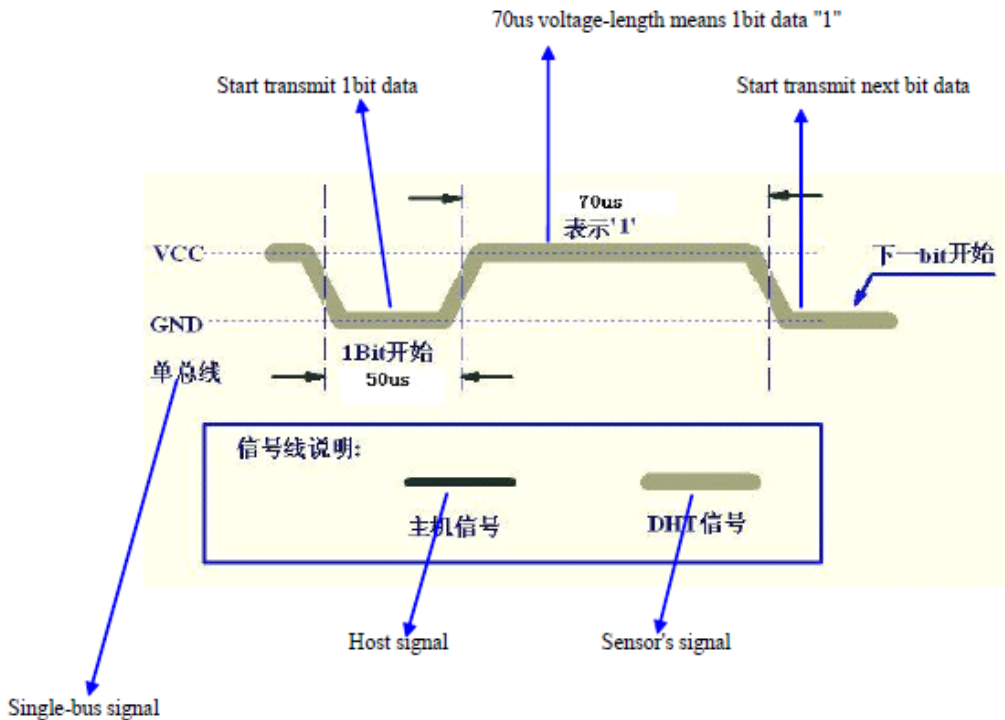
# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

<http://www.aosong.com>

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China



If signal from AM2303 is always high-voltage-level, it means AM2303 is not working properly, please check the electrical connection status.

## 6. Electrical Characteristics:

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3.3	5	5.5	V
Current supply	Measuring	1.3	1.5	2.1	mA
	Average	0.5	0.8	1.1	mA
Collecting period	Second	1.7		2	Second

\*Collecting period should be : >1.7 second.

## 7. Attentions of application:

(1) Operating and storage conditions

We don't recommend the applying RH-range beyond the range stated in this specification. The DHT11 sensor

# Aosong(Guangzhou) Electronics Co.,Ltd

Tell: +86-020-36380552, +86-020-36042809 Fax: +86-020-36380562

<http://www.aosong.com>

Email: [thomasliu198518@yahoo.com.cn](mailto:thomasliu198518@yahoo.com.cn) [sales@aosong.com](mailto:sales@aosong.com)

Address: No.56, Renhe Road, Renhe Town, Baiyun District, Guangzhou, China

can recover after working in non-normal operating condition to calibrated status, but will accelerate sensors' aging.

(2) Attentions to chemical materials

Vapor from chemical materials may interfere AM2303's sensitive-elements and debase AM2303's sensitivity.

(3) Disposal when (1) & (2) happens

Step one: Keep the AM2303 sensor at condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two: After step one, keep the AM2303 sensor at condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Attention to temperature's affection

Relative humidity strongly depend on temperature, that is why we use temperature compensation technology to ensure accurate measurement of RH. But it's still be much better to keep the sensor at same temperature when sensing.

AM2303 should be mounted at the place as far as possible from parts that may cause change to temperature.

(5) Attentions to light

Long time exposure to strong light and ultraviolet may debase AM2303's performance.

(6) Attentions to connection wires

The connection wires' quality will effect communication's quality and distance, high quality shielding-wire is recommended.

(7) Other attentions

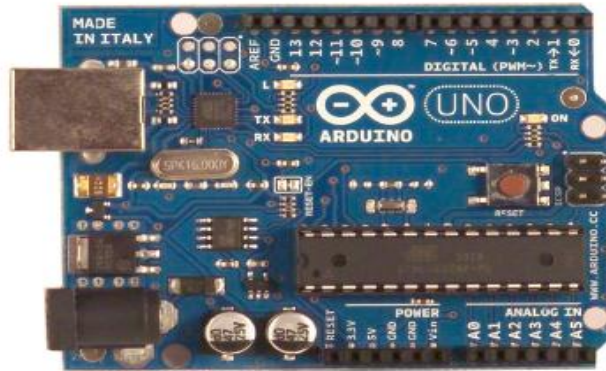
\* Welding temperature should be bellow 260Celsius.

\* Avoid using the sensor under dew condition.

\* Don't use this product in safety or emergency stop devices or any other occasion that failure of AM2303 may cause personal injury.



## 8.4 Anexo 4 Hoja de datos del Arduino UNO



### Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

### Index

Technical Specifications	Page 2
How to use Arduino Programming Enviroment, Basic Tutorials	Page 6
Terms & Conditions	Page 7
Enviromental Policies half sqm of green via Impatto Zero®	Page 7



*radiospares*

*RADIONICS*



# Technical Specification

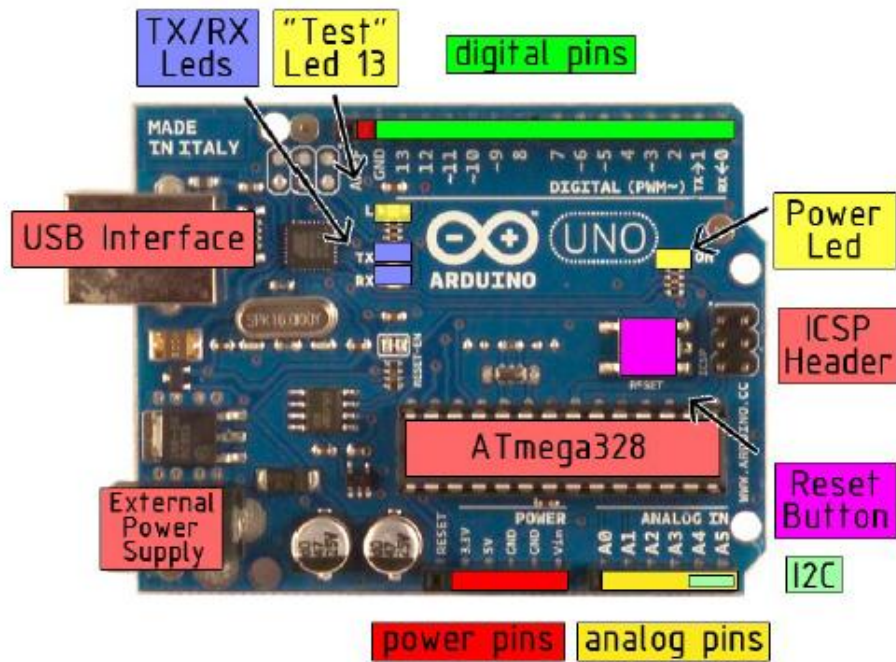


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

## Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

## the board



radiospares RADIONICS



## Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

## Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

## Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I<sup>2</sup>C: 4 (SDA) and 5 (SCL).** Support I<sup>2</sup>C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

## Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an \*.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

## Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



## Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

## USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

## Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



radiospares

RADIONICS



# How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

## Linux Install

## Windows Install

## Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

## Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>  
Arduino-0017>Examples>  
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000);                // wait for a second
  digitalWrite(ledPin, LOW);  // set the LED off
  delay(1000);                // wait for a second
}
```



Done compiling.

Press Compile button  
(to check for errors)



Upload

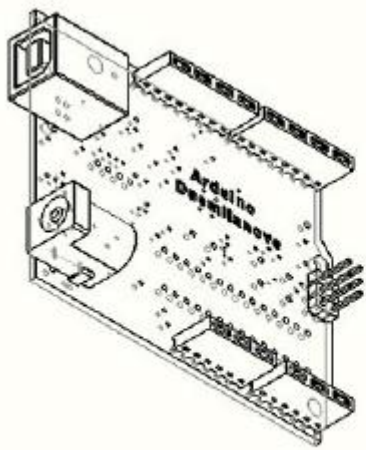
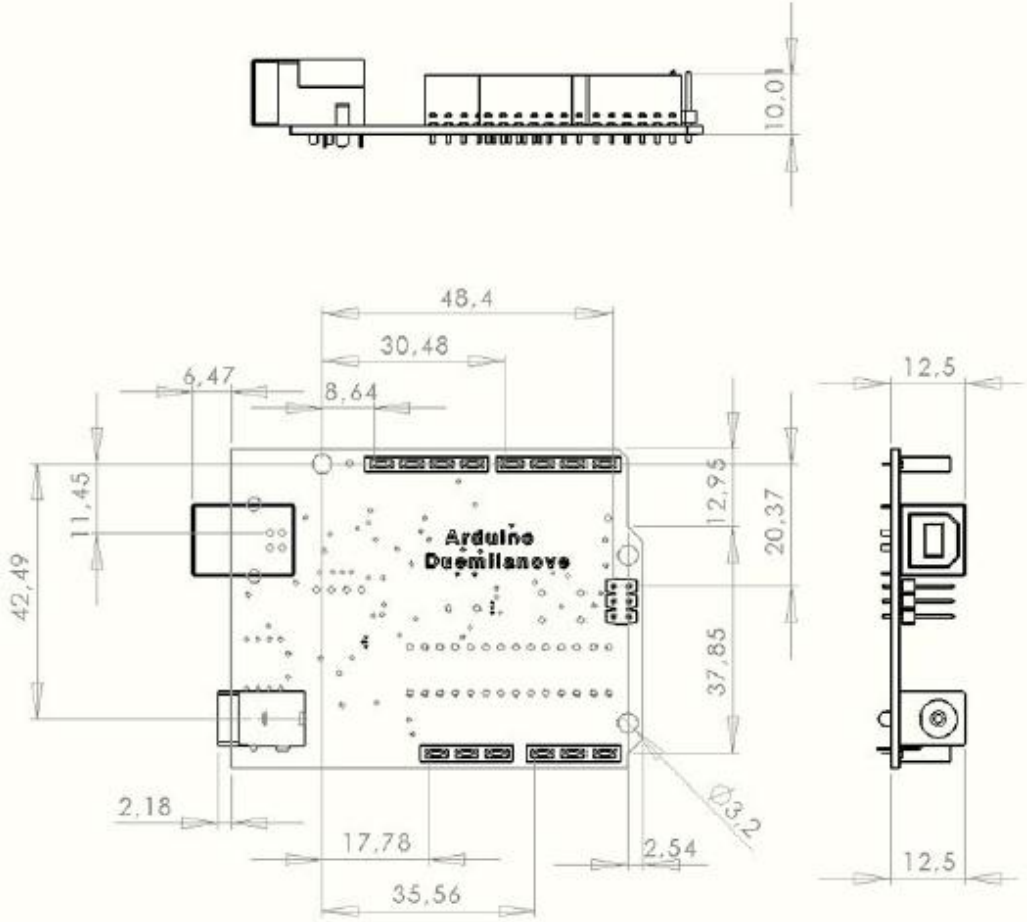


TX RX Flashing



Blinking Led!

Dimensioned Drawing



*radiospares* **RADIONICS**



# Terms & Conditions



## 1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

## 2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

## 3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

## 4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



## Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



*radiospares*

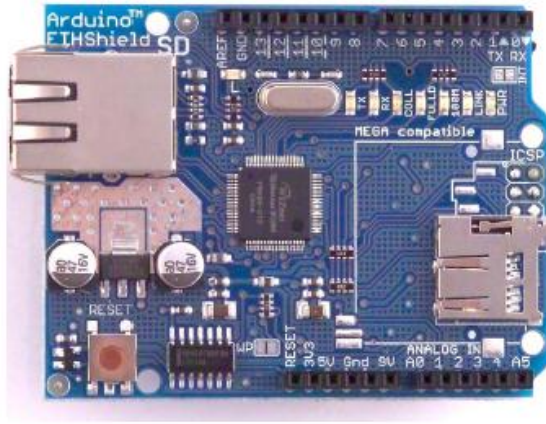
*RADIONICS*





## 8.5 Anexo 5 Hoja de datos de Arduino Ethernet Shield

### Arduino Ethernet Shield



**Download:** [arduino-ethernet-shield-05-schematic.pdf](#), [arduino-ethernet-shield-05-reference-design.zip](#)

**Download:** [arduino-ethernet-shield-schematic.pdf](#), [arduino-ethernet-shield-reference-design.zip](#)

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the [Wiznet W5100](#) ethernet chip ([datasheet](#)). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the [Ethernet library](#) to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the [sdfatlib](#) by Bill Greiman works well. See [this tutorial from Adafruit Industries](#) for instructions (thanks Limor!).

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general i/o. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 ethernet jack.

The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.

See also: [getting started with the ethernet shield](#) and [Ethernet library reference](#)

