



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

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

Título del proyecto:

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
MONITOREO Y CONTROL DE HUMEDAD Y
TEMPERATURA PARA INVERNADEROS CON
ADMINISTRACIÓN SNMP”**

AUTORA:

Jenny del Rocío Patín Chimbo

TUTORA:

Ing. Deysi Inca Balseca

Riobamba -Ecuador

AÑO 2016

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE HUMEDAD Y TEMPERATURA PARA INVERNADEROS CON ADMINISTRACIÓN SNMP** presentado por: Jenny del Rocío Patín Chimbo dirigida por: Ingeniera Deysi Inca Balseca.

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. Deysi Inca
Directora del Proyecto



Firma


Ing. Juan Carlos Cepeda
Miembro del Tribunal



Firma

AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del Proyecto de Graduación, titulado “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE HUMEDAD Y TEMPERATURA PARA INVERNADEROS CON ADMINISTRACIÓN SNMP**”, es absolutamente original, fidedigno y personal, en tal virtud, el contenido del mismo es exclusivamente responsabilidad de: Jenny del Rocío Patín Chimbo e Ingeniera Deysi Inca Balseca. En tal virtud los resultados obtenidos son de exclusividad del autor, del director de tesis y el patrimonio intelectual de la Universidad Nacional de Chimborazo.



Jenny Patín Chimbo

CI: 0201904008-3

Tesista

AGRADECIMIENTO

Agradezco, a Dios por darme la fuerza y sabiduría a lo largo de todos estos años, a mis padres, hermana y hermano por creer en mí y motivarme a seguir en cada momento, a mi hija por ser mi compañera y amiga durante este proceso, a la Facultad por brindarme los recursos necesarios para realizarme como profesional y como persona, a mi Tutora Ing. Deysi Inca Balseca por su apoyo, guía y paciencia en el desarrollo de este proyecto, finalmente gracias a los docentes y amigos que contribuyeron con su asesoría durante el desarrollo del presente proyecto.

Jenny del Rocío Patín Chimbo

DEDICATORIA

Dedico este proyecto de titulación a mis padres Segundo Antonio Patín Pasto y María Rosa Chimbo Cando quienes me han brindado su amor, comprensión y apoyo en todo instante a lo largo de mi vida para seguir cumpliendo con mis objetivos y ver el presente con nuevas perspectivas para un futuro mejor, a mi hija Nayha Suyay Cotacachi Patín por ser mi deseo de superación constante.

Jenny del Rocío Patín Chimbo

INDICE

AUTORÍA DE LA INVESTIGACIÓN	III
AGRADECIMIENTO.....	IV
DEDICATORIA.....	V
INDICE	VI
INDICE DE TABLAS	IX
INDICE DE ILUSTRACIONES.....	X
RESUMEN.....	XIX
SUMARY	XX
INTRODUCCIÓN	XXI
CAPÍTULO I.....	1
1. FUNDAMENTACIÓN TEÓRICA.....	1
1.1. DISEÑO	1
1.2. IMPLEMENTACIÓN.....	1
1.3. PROTOTIPO.....	1
1.4. ADMINISTRACIÓN Y GESTIÓN DE RED	2
1.4.1. SISTEMA DE GESTIÓN DE RED	2
1.4.2. MODELOS DE GESTIÓN DE RED	2
1.4.3. MODELO DE GESTIÓN DE INTERNET.....	2
1.4.4. BASE DE INFORMACIÓN DE GESTIÓN (MIB).....	7
1.4.5. PROTOCOLO SNMP	9
1.4.6. RFCS Y VERSIONES DEL PROTOCOLO SNMP.....	10
1.4.7. SNMP VERSIÓN 2C	11
1.4.8. OPERACIONES Y MENSAJES SNMP	12
1.5. SISTEMA OPERATIVO.....	14
1.6. MONITOREO.....	16
1.6.1. TIPOS DE MONITOREO.....	16
1.6.2. SOFTWARE DE MONITOREO NAGIOS	17
1.7. COMUNICACIÓN LOCAL INALAMBRICA.....	18
1.7.2. MÓDULO XBEE	22
1.7.3. CIRCUITO BÁSICO XBEE	22
1.8. MICRO CONTROLADORES.....	23

1.9. ROUTER.....	29
1.10. SENSORES	30
1.11. INVERNADERO	34
1.12. SISTEMA DE ACTUADORES.....	34
CAPÍTULO II	35
2. METODOLOGÍA.....	35
2.1. TIPO DE ESTUDIO.....	35
2.2. MÉTODOS, TÉCNICAS E INSTRUMENTOS.....	35
2.2.1. MÉTODOS	35
2.2.2. TÉCNICA	35
2.3. POBLACIÓN Y MUESTRA.....	36
2.3.1. POBLACIÓN	36
2.3.2. MUESTRA.....	36
2.3.3 HIPÓTESIS	37
2.4. OPERACIONALIZACIÓN DE VARIABLES	37
2.5. PROCEDIMIENTOS	38
2.6. PROCESAMIENTO Y ANÁLISIS.....	40
2.6.1. IDENTIFICACIÓN Y ESTUDIO DEL PROTOTIPO.....	40
2.6.2. SELECCIÓN Y ESTUDIO DE LOS EQUIPOS ELECTRÓNICOS Y SENSORES.....	41
2.6.3. TEST DE COMUNICACIÓN ENTRE EL ARDUINO Y LOS SENSORES	54
2.6.4. DISEÑO DE ETAPAS.....	79
2.6.5. PROGRAMACIÓN EN EL ENTORNO DE ARDUINO	80
2.6.6. ENSAMBLAJE DE LAS ETAPAS	82
2.6.7. RECOLECCIÓN DE DATOS MEDIANTE UN SNIFFER	83
2.6.8. COMPROBACIÓN DE LA HIPÓTESIS	83
CAPÍTULO III	91
3. RESULTADOS	91
3.1. PROCESAMIENTO DE LA INFORMACIÓN	91
3.2. PRUEBAS DE COMUNICACIÓN ENTRE DISPOSITIVOS	92
3.2.1. VELOCIDAD DE TRANSMISIÓN	92
3.2.2. INTERCAMBIO DE PAQUETES	93
3.2.3. TIEMPO DE TRANSMISIÓN	94

3.3. ANÁLISIS FINANCIERO	94
CAPÍTULO IV	97
4. DISCUSIÓN	97
CAPÍTULO V	99
5. CONCLUSIONES Y RECOMENDACIONES	99
5.1. CONCLUSIONES	99
5.2. RECOMENDACIONES	100
CAPÍTULO VI.....	101
6. PROPUESTA	101
6.1. TÍTULO DE LA PROPUESTA.....	101
6.2. INTRODUCCIÓN	101
6.3. OBJETIVOS	102
6.3.1. OBJETIVO GENERAL	102
6.3.2. OBJETIVOS ESPECIFICOS	102
6.4. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA	102
6.5. DESCRIPCIÓN DE LA PROPUESTA	103
6.6. DISEÑO ORGANIZACIONAL.....	104
6.7. MONITOREO Y EVALUACIÓN DE LA PROPUESTA	105
7. BIBLIOGRAFÍA	106
8. APÉNDICES Y ANEXOS.....	108
8.1. ANEXO 1.- CONFIGURACIÓN DEL MODEM DOMICILIARIO PARA LA COMUNICACIÓN INALÁMBRICA LOCAL Y REMOTA.....	108
8.2. ANEXO 6.- PRUEBAS DE COMUNICACIÓN Y MONITOREO.....	110
8.4. ANEXO 7.- DISEÑO DE LOS CIRCUITOS.....	112
8.5. ANEXO 10.- HOJA DE DATOS SENSOR DTH22.....	114
8.6. ANEXO 11.- HOJA DE DATOS ARDUINO MEGA 2560	121
8.7. ANEXO 12.- HOJA DE DATOS ARDUINO ETHERNET SHIELD	124
8.8. ANEXO 12.- HOJA DE DATOS XBEE	125
8.9. ANEXO 13.- HOJA DE DATOS VENTILADOR.....	128
8.10. ANEXO 14.- HOJA DE DATOS BOMBA	129
8.11. ANEXO 15.- HOJA DE DATOS OLED	130
8.12. ANEXO 16.- HOJA DE DATOS SENSOR SOIL.....	131

INDICE DE TABLAS

Tabla 1: Resumen de las versiones SNMP	10
Tabla 2: Operaciones SNMP	12
Tabla 3: Operacionalización de variables independiente y dependiente	37
Tabla 4: Parámetros de temperatura y humedad del tomate riñón.....	39
Tabla 5: Equipos electrónicos y sensores	41
Tabla 6: Características de la tarjeta Arduino MEGA	43
Tabla 7: Características Shield relé de 4 canales 5V	45
Tabla 8: Características de la tarjeta Arduino Ethernet	46
Tabla 9: Características tarjeta Xbee.	48
Tabla 10: Características de Sensor DHT22	50
Tabla 11: Características Sensor Soil	52
Tabla 12: Características Oled 128x64 Arduino.....	54
Tabla 13: Tiempos de respuesta sin monitoreo.....	86
Tabla 14: Tiempos de respuesta con monitoreo	87
Tabla 15: Media muestral	89
Tabla 16: Frecuencia esperada.....	89
Tabla 17: Tabla de valores estadísticos	89
Tabla 18: Velocidad de Transmisión	92
Tabla 19: Paquetes Capturados durante la Transmisión.....	93
Tabla 20: Tiempo de Transmisión	94
Tabla 21: Recursos Materiales.....	95
Tabla 22: Recursos Varios	95
Tabla 23: Recursos Financieros	96

INDICE DE ILUSTRACIONES

Ilustración 1: Arquitectura de un Sistema de Administración	3
Ilustración 2: Parte de árbol de objetos según SMIV1	5
Ilustración 3: Parte de árbol de objetos según SMIV2	6
Ilustración 4: Subárbol MIB II.....	9
Ilustración 5: Arquitectura TCP/IP y SNMP	10
Ilustración 6: Estructura PDU	13
Ilustración 7: Sistema Operativo.....	15
Ilustración 8: Ejemplos de sistemas operativos.	15
Ilustración 9: Formas de medir el tráfico en la red.	16
Ilustración 10: Ejemplo de Interfaz de Nagios	18
Ilustración 11: Conexión de red inalámbrica	19
Ilustración 12: Usos de ZigBee.....	20
Ilustración 13: Tipos de Dispositivos ZigBee.....	21
Ilustración 14: Módulo XBee.....	22
Ilustración 15: Circuito Básico XBee	23
Ilustración 16: Partes que integran un microcontrolador	23
Ilustración 17: Entorno Arduino para Microsoft Windows	25
Ilustración 18: Boceto con un ejemplo de lectura análoga serial.....	26
Ilustración 19: Menú para acceder al sketchbook.....	27
Ilustración 20: Menú de librerías disponibles en el software Arduino	28
Ilustración 21: Botón de acceso al monitor serial	28
Ilustración 22: Router	30
Ilustración 23: Estructura básica de un sensor	33
Ilustración 24: Invernadero	34

Ilustración 25: Diagrama de procedimiento.....	39
Ilustración 26: Arduino Mega 2560 R3.	42
Ilustración 27: Arduino Mega 2560 R3	44
Ilustración 28: Shield relé de 4 canales 5V.....	44
Ilustración 29: Diagrama de conexión tarjeta Shield y los sensores.....	45
Ilustración 30: Arduino Ethernet placa vista frontal- vista trasera	46
Ilustración 31: Tarjeta Xbee.....	48
Ilustración 32: Diagrama de la Tarjeta Xbee.	49
Ilustración 33: Sensor DHT22.	49
Ilustración 34: Pines de conexión	51
Ilustración 35: Encapsulado del Sensor Soil.....	52
Ilustración 36: Pines de conexión.	53
Ilustración 37: Pantalla Oled 128x64 Arduino	53
Ilustración 38: Conexión entre la tarjeta Arduino y los sensores	56
Ilustración 39: Programación del Sensor DHT22.....	57
Ilustración 40: Conexión entre la tarjeta Arduino y sensor Soil.....	58
Ilustración 41: Programación del Sensor Soil.....	58
Ilustración 42: Shield Xbee V03.....	59
Ilustración 43: Modulo Xbee PRO con antena integrada.	59
Ilustración 44: Integración del módulo Xbee+Shield Xbee 03+Arduino.	60
Ilustración 45: Xbee+Shield Xbee 03+Arduino.	60
Ilustración 46: Código usado para la transmisión de módulo XBee.....	61
Ilustración 47: Código usado para la recepción de módulo XBee.....	62
Ilustración 48: Router y Shield Ethernet (PC).	63
Ilustración 49: Agentuino sitio web del autor.....	63

Ilustración 50: Uso del SNMPGET de Agentuino.....	64
Ilustración 51: Uso del SNMPSET de Agentuino.	64
Ilustración 52: Pruebas usando SNMPSET manualmente.....	65
Ilustración 53: Instalación de Nagios por consola.....	65
Ilustración 54: Usuario para Nagios	66
Ilustración 55: Password para Nagios.....	66
Ilustración 56: Grupo Nagcmd.....	66
Ilustración 57: Página de Nagios	67
Ilustración 58: Comandos de Nagios	67
Ilustración 59: Plugins descargados.....	67
Ilustración 60: Directorio de Nagios.....	68
Ilustración 61: Comandos terminal Ubuntu.....	68
Ilustración 62: Interface web de Nacgios.....	68
Ilustración 63: Comando nagios.conf	69
Ilustración 64: Respuesta de Nagios	69
Ilustración 65: Cuenta Nagios admin.....	69
Ilustración 66: Servicio Apache.....	70
Ilustración 67: Plugins de Nagios con servicio Apache.....	70
Ilustración 68: Edición del archivo contacts.org.....	71
Ilustración 69: Configuración de datos personales	71
Ilustración 70: Permiso de acceso administrativo.....	72
Ilustración 71: Reiniciar Apache	72
Ilustración 72: Instalación de Nagios 4.1.1 en Ubuntu 15.10.....	73
Ilustración 73: Servicio NRPE.....	74
Ilustración 74: Archivo nrpe.cfg.....	74

Ilustración 75: Servidor a monitorear	74
Ilustración 76: Servicio de NRPE	75
Ilustración 77: Configuración de nagios.cfg.....	75
Ilustración 78: Códigos de líneas	75
Ilustración 79: Directorio servers.....	76
Ilustración 80: Configuración para monitoreo	76
Ilustración 81: Líneas de estructura	76
Ilustración 82: Creación de nuevo host.....	77
Ilustración 83: Creación de un nuevo host en Nagios 4.1.1.....	77
Ilustración 84: Monitoreo de la Humedad	78
Ilustración 85: Monitoreo de la Temperatura	78
Ilustración 86: Pruebas Wireshark	79
Ilustración 87: Tarjeta Arduino MEGA con la placa shield y el módulo XBee	79
Ilustración 88: Ubicación de los sensores dentro del invernadero con la tarjeta Shiel y módulo Xbee.....	80
Ilustración 89: Librerías en el entorno de Arduino	81
Ilustración 90: Programación del proyecto	81
Ilustración 91: Ensamblaje de las etapas´	82
Ilustración 92: Recolección de datos	83
Ilustración 93: Diagrama esquemático del proyecto.....	85
Ilustración 94: Campana de Gauss para decisión	90
Ilustración 95: Velocidad de Transmisión	92
Ilustración 96: Valores estadísticos	93
Ilustración 97: Tiempo de Transmisión en Segundos.....	94
Ilustración 98: Diagrama del sistema organizacional del proyecto	104

Ilustración 99: Ingreso interfaz de configuración del modem	108
Ilustración 100: Interfaz del modem - ip fija asignada	108
Ilustración 101: Configuración para habilitación de puertos	109
Ilustración 102: Interfaz de Portforward.....	109
Ilustración 103: Comprobación del puerto habilitado	110
Ilustración 104: Comprobación de comunicación mediante el CMD.....	110
Ilustración 105: Comprobación de monitoreo mediante el Sniffer.....	111
Ilustración 106: Diagrama de flujo	111
Ilustración 107: Circuito Esquemático Receptor	112
Ilustración 108: Circuito Esquemático Transmisor	113

RESUMEN

Este proyecto tiene como objetivo la implementación de un sistema de monitoreo y control con sensores, que hace posible que el invernadero se automatice, mediante la administración SNMP. La idea principal de este proyecto es presentar una opción de control y monitoreo de datos mediante una red inalámbricamente y escoger el software de monitoreo adecuado el que enviara una alerta mediante un e-mail al usuario si existen cambios bruscos de climatización.

El Sistema de Control se realiza mediante la utilización del Shield relé de 4 canales de 5V y sensores de temperatura, humedad relativa y humedad del suelo los que están encargados de recopilar toda la información de los cambios ambientales que suscitan en el lapso de un tiempo determinado y permite que se activen manual y automático los elementos auxiliares que se encuentran en el invernadero (calefactor, ventilador y bomba). Se realizó un estudio básico del protocolo de comunicación inalámbrica ZigBee, capaz de proveer las mediciones, a un módulo central, el cual toma decisiones basadas en parámetros establecidos por el usuario y realiza el control de los actuadores, para ello se acondicionaron las señales de los sensores para que puedan acoplarse al módulo inalámbrico a fin de que se puedan transmitir de forma adecuada los datos.

Para la administración SNMP con la integración de Arduino con el protocolo SNMP se lo realiza mediante la utilización de la librería AGENTUINO, para el monitoreo del sistema se utiliza el software libre Nagios por la facilidad de uso. El control de parámetros ambientales de temperatura y humedad, es un servicio que contribuye a la optimización de tiempo y recursos en la producción del invernadero, de acuerdo a las necesidades del usuario.

Las pruebas se realizan mediante la manipulación de los sensores de forma manual o automática, adicional se usara una maqueta exclusivamente para este proyecto.



UNIVERSIDAD NACIONAL DE CHIMBORAZO



FACULTAD DE INGENIERÍA

CENTRO DE IDIOMAS

Lic. Geovanny Armas

1 de abril de 2016

SUMMARY

This project aims to implement a monitoring and control system with sensors; they make the automation of a greenhouse possible through the SNMP management. The main idea of this project is to present an option for the control and monitoring of data over a wireless network and pick the appropriate monitoring software which will send the user an alert by e-mail in case there are sudden changes in climate.

The control system is performed by using the Shield 4-channel 5V relay and temperature sensors, relative humidity and soil moisture which are responsible for gathering all the information about environmental changes that arise in a determined period of time, it allows the manual and automatic activation of the auxiliary devices inside the greenhouse (heater, blower and pump). A basic study of the ZigBee wireless communication protocol was carried out, it is capable of providing measurements towards a central module which makes decisions based on parameters set by the user and performs the control of the actuators, for this the signals of the sensors were conditioned so that they can be coupled to the wireless module in order to transmit data properly.

The SNMP management with the integration of Arduino with the SNMP protocol is carried out by using the AGENTUINO library. For monitoring the system, the Nagios open source software is used. The control of environmental parameters of temperature and humidity is a service that contributes to the optimization of time and resources in the production of the greenhouse, according to the needs of the user.

The tests are performed by manipulating sensors manually or automatically, additionally a dummy model will be exclusively used for this project.



INTRODUCCIÓN

Desde que se creó el sistema de internet y redes de comunicación, en la actualidad han evolucionado y se ha vuelto más complejo, por esta razón es importante mantener una correcta administración que no solo consista en un simple monitoreo de red, si no que se involucre mucho más, realizando una gestión con herramientas que proporcionen una información más completa, amplia y eficiente que ayude el correcto control, uso y desempeño de los equipos, para que de manera sencilla que hagan un reconocimiento de comportamiento de todo lo utilizado en la red y el estado de los mensajes.

Por los condiciones anteriormente presentados el proyecto se centra en la implementación de un sistema de control y monitoreo climático en el interior de un invernadero, de esta forma se podrá ayudar a los usuarios a facilitar la manipulación del ambiente de un invernadero mediante un sistema de control de la temperatura y humedad, con el propósito o finalidad hallar un modelo climático adecuado que se ajuste a las condiciones del invernadero para el cultivo de tomate riñón. Con la administración SNMP de los sensores de temperatura, humedad relativa y humedad del suelo; se ayuda al usuario el monitoreo y enviar una alerta mediante la Web, el cual permite activar los actuadores (el calefactor, ventilador y bomba); de forma manual o automática para poder obtener una buena condición climática para el cultivo, la comunicación inalámbrica se ha utiliza los módulos XBee los cuales se encargan transmisión/recepción de datos, basado en tecnología ZigBee.

Este documento se realizó en varios capítulos y cuenta con anexos en los que incluye los respaldos de cada uno de los módulos desarrollados.

En el Primer Capítulo se representa una descripción de la fundamentación teórica de SNMP, Sistema Operativos, Sistema de Monitoreo, Software de Monitoreo,

Microcontroladores, Arduino, Shields, otros dispositivos que permiten la construcción del prototipo, de comunicación y administración SNMP.

En el Segundo Capítulo contiene la metodología que se va usar, la población y muestra en donde se va realizar el estudio, la operacionalización de variables y los procedimientos que se siguieron para la construcción de los dispositivos de control, de comunicación y administración SNMP.

En el tercer Capítulo contiene la descripción de los resultados, las pruebas de los equipos de forma manual o automática y el análisis financiero.

En el Cuarto Capítulo se muestra la discusión del proyecto.

En el Quinto Capítulo demuestro las pruebas de análisis de cada uno de los módulos diseñados en el sistema, las conclusiones y recomendaciones a la que llegué después de haber concluido la investigación.

En el Sexto Capítulo esta la propuesta, introducción los objetivos del proyecto a investigar, Fundamentación Científico –Técnica, Descripción de la propuesta y el diseño organizacional.

En el Séptimo Capítulo demuestro los folletos, libros, páginas Web que se usaron para realizar la investigación.

En el Octavo Capítulo demuestro los anexos.

El dispositivo responde favorablemente en las fases en las que fue diseñado, por lo que el equipo es considerado práctico para su uso.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

En este proyecto de investigación se indica el desarrollo de un prototipo de control y monitoreo, con administración SNMP para invernaderos.

1.1. DISEÑO

Es el proceso que antepone a una configuración, “pre-configuración”, en la investigación de solución en cualquier campo. Utilizándolo regularmente en el contexto de la industria, ingeniería, arquitectura, comunicación y otras disciplinas creativas siendo el paso previo a la búsqueda de una solución o un conjunto de soluciones. (Diccionario, 2014)

1.2. IMPLEMENTACIÓN

Es la construcción de una aplicación, instalación o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política (ibídem).

1.3. PROTOTIPO

Se puede decir que ya existen invernaderos automatizados pero aún no se han incorporado la administración de SNMP, facilita el monitoreo mediante un software capaz de enviar una alerta al usuario si existe un cambio climático brusco de forma más eficiente.

Se puede referir a cualquier tipo de máquina en pruebas, o un objeto diseñado para una demostración de cualquier tipo (Ibídem)

1.4. ADMINISTRACIÓN Y GESTIÓN DE RED

La administración y gestión de red comprende todo un conjunto de actividades que se encargan de la vigilancia y control de los recursos de una red, para garantizar su eficiencia y rendimiento con un mínimo costo. Toda pequeña red posee una gestión de red.

1.4.1. Sistema de Gestión de Red

Es un conjunto de elementos que permiten realizar el proceso de administración de la red, a través de una interfaz física donde se envían instrucciones y muestra información para monitorear y gestionar cualquier elemento de la red.

1.4.2. Modelos de Gestión de Red

El modelo de gestión implementado se fundamentará en los siguientes modelos:

- Modelo de gestión de Internet.
- Modelo de gestión ISO/OSI.
- Modelo de gestión TMN.

1.4.3. Modelo de Gestión de Internet

Este modelo gestiona la arquitectura TCP/IP. Los componentes que trabajan en forma conjunta en la gestión de red en Internet son los siguientes:

- Estructura de la información de gestión (SMI).
- Base de información de gestión (MIB).
- Protocolo simple de gestión de red (SNMP)

1.4.3.1. Base de Información de Gestión MIB

La base de información de gestión (MIB)¹ está compuesta por un conjunto de objetos administrados con sus atributos, esta información se encuentra organizada jerárquicamente.

La (**Ilustración 1**) muestra los elementos que conforman la arquitectura de un sistema de administración.

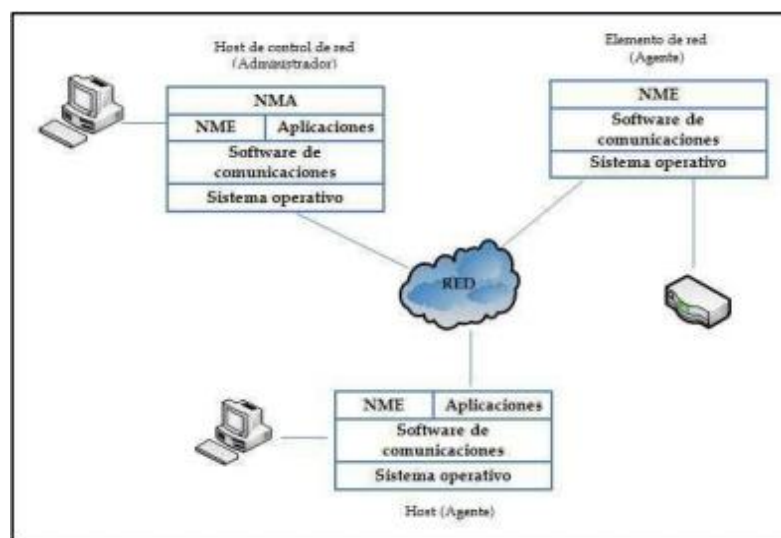


Ilustración 1: Arquitectura de un Sistema de Administración

Fuente: Fundamentos de administración de red

Administrador: es un equipo (servicio) que envía consultas a un Agente (equipo administrado). En algunos casos el administrador SNMP envía peticiones al Agente para que realice cambios de configuración.

Agente: Es un equipo o dispositivo que responde peticiones de los Managers SNMP. El Agente envía eventos específicos como un reinicio o acceso ilegal sin que el Administrador lo solicite.

1.4.3.2. Estructura de la Información de gestión

¹ Base de Información de Gestión

La SMI², es un lenguaje de definición de datos que proporciona una manera de definir objetos administrados y su comportamiento, identifica los tipos de datos que pueden utilizarse en la MIB y especifica cómo se debe representar y nombrar los objetos dentro de la MIB, elimina la ambigüedad en la sintaxis y semántica de los datos.

La SMI está definida en el RFC 1155 para la versión 1 (SMIv1) y en el RFC 2578 para la versión 2 (SMIv2) que tiene mejoras para SNMPv2. El objetivo principal de la SMI es mantener a la MIB simple y extensible, por lo cual solo se pueden utilizar los tipos de datos simples como escalares y arreglos bidimensionales de escalares.

La descripción de los objetos gestionados se lo hace mediante el ASN.1³. La SMI proporciona formas para definir:

- La estructura de la MIB.
- Sintaxis y tipos de valores para objetos individuales.
- Codificación de los valores de los objetos.

1.4.3.3. Estructura de la MIB

La MIB es una base de datos que contiene información jerárquica, estructurada en forma de árbol, donde sus ramificaciones son los objetos a ser gestionados en una red de comunicaciones.

Asociado con cada objeto en una MIB existe un identificador único conocido como OID⁴; este identificador de objeto es una secuencia de números enteros positivos, separados por un punto.

Existe otra forma para identificar a los objetos gestionados y se lo hace mediante una secuencia de nombres que representan los números, lo que facilita al

² Estructura de la Información de Gestión

³ Abstract Syntax Notation - Notación Sintáctica Abstracta

⁴ Identificador de Objeto

usuario al momento de ubicar cualquier objeto dentro del árbol.

En la **(Ilustración 2)** se muestra los niveles superiores del árbol de objetos organizados según SMIV1. En este árbol de objetos tengo en la parte superior el nodo denominado como Raíz (Root Node) el cual no tiene ningún número de identificación, solo se lo identifica con un punto (.) o con su nombre; se denomina nodos hojas a los objetos que no poseen ninguna ramificación y subárbol a los objetos que poseen ramificaciones por ejemplo el nodo iso (1). Cabe mencionar que los nodos ccitt y joint no se relacionan con SNMP

Para el modelo de gestión de Internet el subárbol iso.org.dod.internet que se representa como 1.3.6.1 es el más importante, a partir de este existen más ramificaciones o estructuras que se derivan del subárbol mencionado.

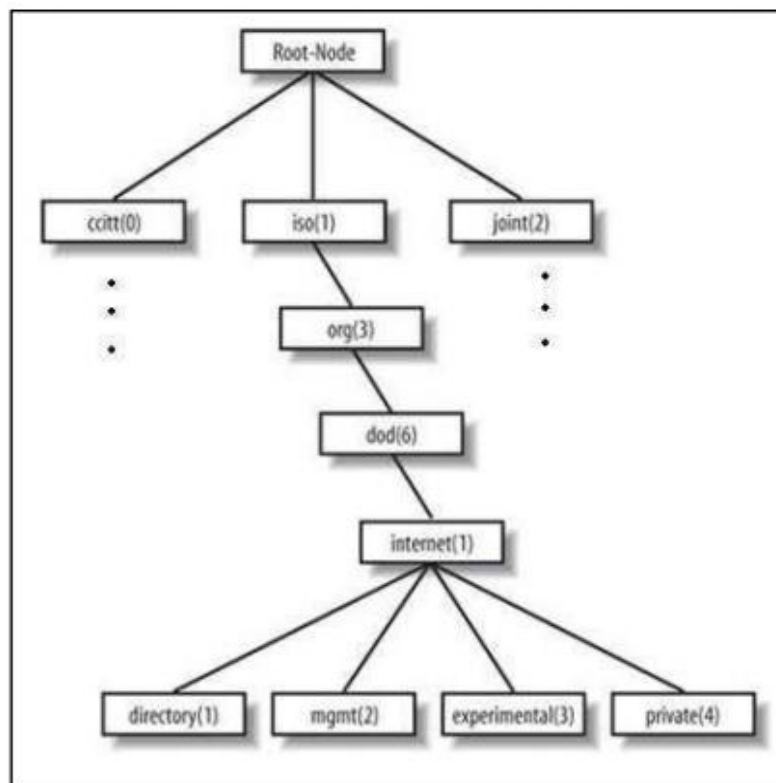


Ilustración 2: Parte de árbol de objetos según SMIV1

Fuente: Utilidades de la gestión de Redes

SMIV2 extiende el árbol de objetos SMI mediante la adición de la rama SNMPv2 al subárbol Internet, además de aumentar nuevos tipos de datos e incluir

una serie de cambios.

En la **(Ilustración 3)** se muestra parte del árbol de objetos MIB según SMIV2. El OID de esta nueva rama es 1.3.6.1.6 o iso.org.dod.internet.snmpV2.

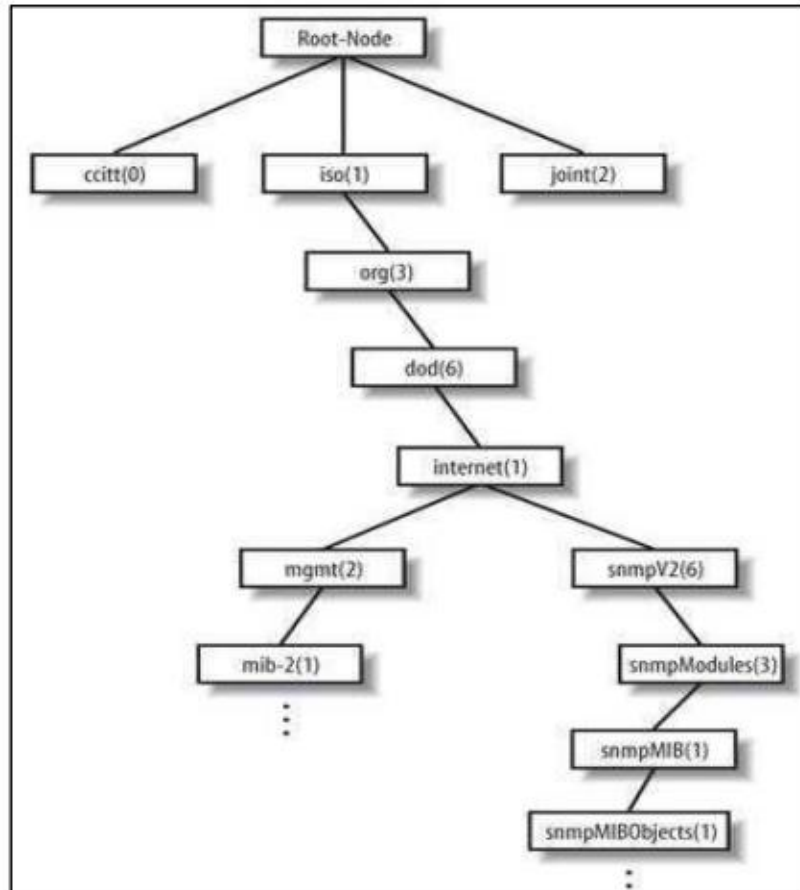


Ilustración 3: Parte de árbol de objetos según SMIV2

Fuente: Utilidades de la gestión de Redes

1.4.3.4. Sintaxis de objetos

Los objetos dentro de la MIB se definen de manera específica mediante la sintaxis ASN.1 (Abstract Syntax Notation One). La definición de un objeto MIB comprende el tipo de dato, los estados permitidos, rango de valores y la relación con otros objetos dentro de la MIB.

1.4.3.5. Tipos de datos

SMI define los diversos tipos de datos que son permitidos, utilizando un subconjunto de elementos y características de ASN.1.

Los tipos de datos se los clasifica en los siguientes grupos:

- Primitivos o simples
- Estructurados o compuestos
- Definidos o etiquetados

1.4.4. Base de Información de Gestión (MIB)

La MIB como ya se indicó en la sección 1.4.3.3. , es una colección de datos que almacena valores de los objetos gestionados y se organiza de manera jerárquica. Existen dos tipos de objetos MIB:

- Escalares: Objetos que tienen una sola instancia de la variable que almacenan.
- Tabulares: Objetos que especifican varias instancias relacionadas a objetos MIB.

Dentro de la MIB existen varios grupos definidos por organismos de estandarización o empresas privadas, sin embargo el grupo MIB II debe ser implementado en cualquier elemento de red sin importar su fabricante.

1.4.4.1. MIB II

La MIB II es un grupo importante dentro de la administración de red ya que todo dispositivo que soporte SNMP debe tener este grupo. La MIB II es parte del grupo management identificándose a este grupo de la siguiente manera root.iso.org.dod.internet.management.mibII o .1.3.6.1.2.1.

La MIB II se compone de los siguientes grupos:

System: Los objetos de este grupo proporcionan información del sistema gestionado, como el nombre, descripción, contacto del sistema, disponibilidad.

OID: .1.3.6.1.2.1.1: Interfaces: Registra información de las interfaces de red presentes en el sistema y de los eventos ocurridos en las mismas.

OID: .1.3.6.1.2.1.2: At: Grupo de traducción de direcciones (address translation), relacionado con las traducciones de direcciones de red a direcciones físicas. Obsoleto pero se mantiene por compatibilidad con MIB-I.

OID: .1.3.6.1.2.1.3: Ip: Almacena información correspondiente al protocolo IP, tanto de configuración como de estadísticas.

OID: .1.3.6.1.2.1.4: Icmp: Recopila información sobre el protocolo ICMP, como paquetes perdidos, descartados o con errores.

OID: .1.3.6.1.2.1.5: Tcp: Almacena información relacionada al protocolo TCP, como el estado de una conexión TCP.

OID: .1.3.6.1.2.1.6: Udp: Mantiene información correspondiente al protocolo UDP.

OID: .1.3.6.1.2.1.7: Egp: Recopila información relativa al protocolo EGP.

OID: .1.3.6.1.2.1.8: Transmission: Guarda información sobre los medios de transmisión.

OID: .1.3.6.1.2.1.10: Snmp: Contiene información para implementación y operación del protocolo SNMP.

OID: .1.3.6.1.2.1.11 [25]: Se muestra en la **(Ilustración 4)** los grupos que componen la MIB II.

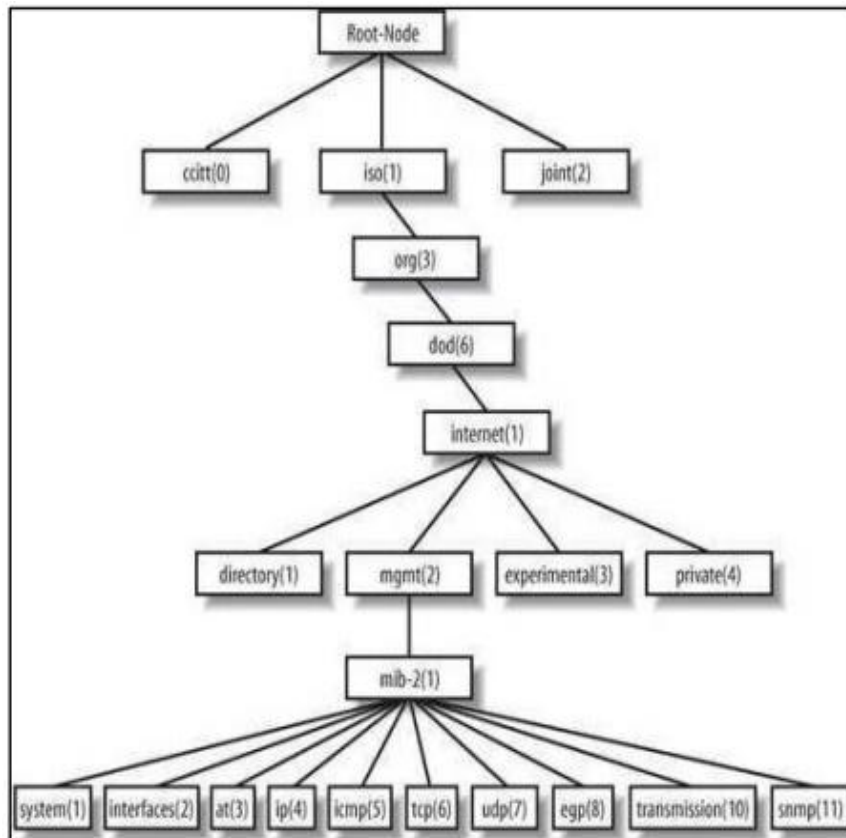


Ilustración 4: Subárbol MIB II

Fuente: Utilidades de la gestión de Redes.

1.4.5. Protocolo SNMP

El protocolo SNMP⁵ basado en la arquitectura TCP/IP, trabaja con una estructura cliente-servidor y usa servicios no orientados a conexión a través del protocolo de transporte UDP⁶.

Los agentes SNMP utilizan el puerto UDP 161 para escuchar peticiones por parte del gestor SNMP. El gestor a través del puerto UDP 162 recibe notificaciones que generan el agente y donde debe existir un proceso gestor de interrupciones que las procesa.

⁵ Protocolo Simple de Administración de Red.

⁶ Protocolo de Datagrama de Usuario.

La **(Ilustración 5)** muestra la arquitectura TCP/IP y SNMP.

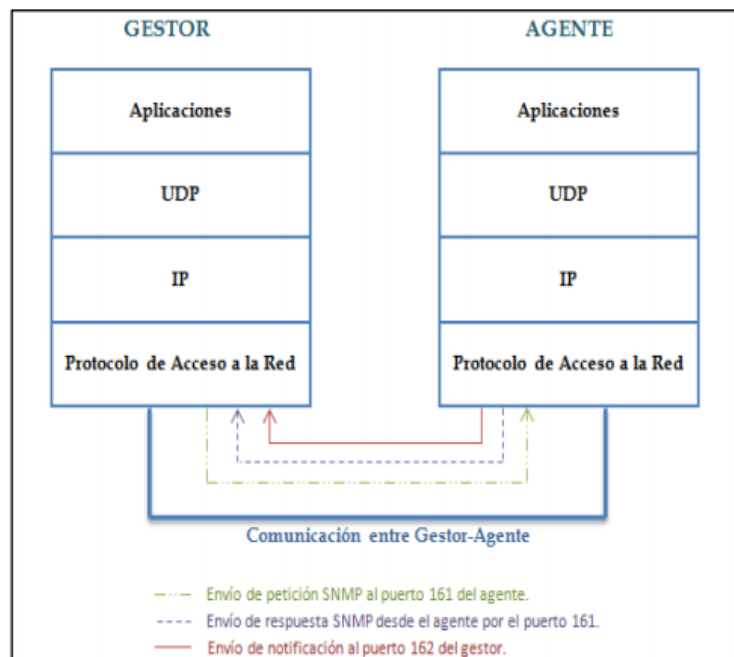


Ilustración 5: Arquitectura TCP/IP y SNMP

Fuente: Essential SNMP

1.4.6. RFCs y Versiones del protocolo SNMP

El protocolo SNMP está definido en varios RFC⁷ propuestos por el IETF⁸, existen tres versiones: SNMPv1, SNMPv2c y SNMPv3. Se puede ver en la siguiente (**Tabla 1**).

Tabla 1: Resumen de las versiones SNMP

Versión	Descripción
SNMPv1	Se desarrolló debido a la creciente necesidad de un mecanismo simple y estandarizado para la gestión y monitoreo de los equipos de la red y que no supusiese cambios en el rendimiento de las redes.

⁷ Request For Comments - Petición de Comentarios

⁸ Internet Engineering Task Force - Fuerza de Tareas de Ingeniería de Internet

SNMPv2, SNMPv2c	<p>Se desarrolló con la idea de mejorar la primera versión y solventar problemas de seguridad y sobrecarga en las transferencias de datos. Sin embargo solo se optimizó la denomina SNMPv2c, donde la “c” indica que se mantiene el esquema de transferencia de datos, por ello también de seguridad basado en comunidades. Actualmente es el más empleado debido, sobre todo, a su facilidad de implantación y mantenimiento.</p> <p>Se agrega soporte para la administración en redes distribuidas y centralizas.</p> <p>Mejoras en el SMI y en las operaciones. Se implementan 2 nuevas PDUs: GetBulkRequest, InformRequest.</p>
SNMPv3	<p>Surge con el propósito de proveer un sistema de seguridad y administración más robusto y flexible. Emplea un nuevo modelo de seguridad que asegura autenticación y encriptación. No supone cambios determinantes en cuanto a la operatividad que ofrece SNMPv2, aparte de los cambios en cuanto a seguridad.</p>

Autor.- Jenny Patín Chimbo

1.4.7. SNMP Versión 2c

SNMPv2c como tal no incluye mecanismos de seguridad, pero si presenta mejoras en el intercambio de información de gestión como lo es la eficiencia de operación, la funcionalidad y el rendimiento.

SNMPv2c además permite la comunicación entre NMA (gestor-gestor), aumenta el número de operaciones mediante el manejo de nuevas PDU, permite la lectura de tablas completas en una sola operación, agrega características a la SMI y desarrolla una MIB para esta versión de SNMP.

1.4.8. Operaciones y Mensajes SNMP

El corazón de SNMP es una serie simple de operaciones que les da a los administradores la capacidad de analizar los distintos dispositivos administrados e interactuar con estos.

Las operaciones de SNMP se puede ver en la siguiente **Tabla 2:**

Tabla 2: Operaciones SNMP

SNMP v1	
Get	Se utiliza por el NMS para recuperar el valor de una o más instancias de objetos de un agente.
get-next	Es utilizada por el NMS para recuperar el valor de la siguiente instancia de objeto de una tabla o lista dentro de un agente
Set	Es utilizada por NMS para configurar valores de instancias de objetos dentro de un agente.
Trap	Es utilizada por los agentes para informar de forma asíncrona el SMN de un hecho relevante.
SNMP v2	
GetBulk	Para que el gestor recupere de una forma eficiente grandes bloques de datos, tales como columnas de una tabla.

Inform	Para que un agente envíe información espontánea al gestor y reciba confirmación
Report	Para que el agente envíe de forma espontáneas excepciones y errores de protocolo.

Autor.- Jenny Patín Chimbo

Mensajes utilizados por SNMP poseen el siguiente formato:

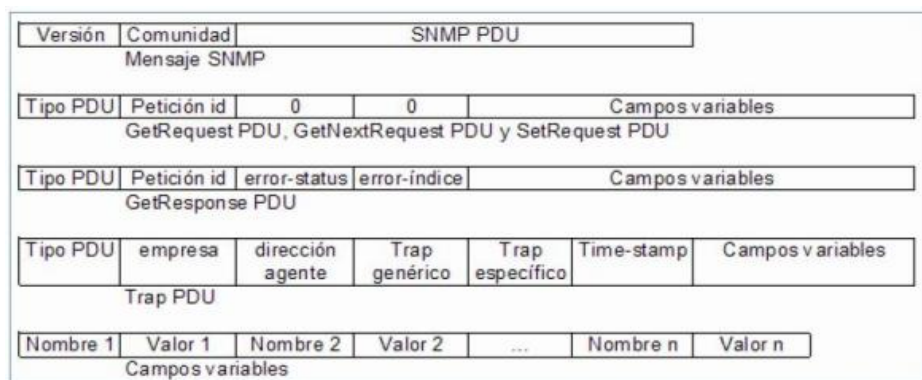


Ilustración 6: Estructura PDU

Fuente: Utilidades de la gestión de Redes

- **Versión:** Número de versión de protocolo que se está utilizando (por ejemplo 1 para SNMPv1)
- **Comunidad:** Nombre o palabra clave que se usa para la autenticación.

SNMP PDU: indica el contenido de la PDU, el que depende de la operación que se ejecute, que puede ser algún tipo de Request (como GetRequest, GetNextRequest y SetRequest), un GetResponse o un Trap.

- **Petición ID:** usado para distinguir una solicitud con una identificación única, entre las demás solicitudes.
- **Error-status:** usado para indicar que ha habido una excepción mientras se procesaba una solicitud.

- **Error-índice:** cuando el error-status es diferente de cero (no hubo error) puede proporcionar información adicional indicando que variable causó la excepción.
- **Campos variables:** una lista de nombre de variables con sus correspondientes valores. Normalmente contiene los datos solicitados por una operación Get o Trap.

Como se aprecia en la (**Ilustración 6**) un mensaje de tipo Trap tiene una estructura diferente:

- **Empresa (Enterprise):** indica el tipo de objeto que genera el Trap.
- **Dirección agente:** indica la dirección IP del Agente que emite el Trap.
- **Trap genérico:** tipo de Trap que informa sobre un estado, válido para cualquier dispositivo.
- **Trap específico:** utilizado para Traps privados (de fabricantes), así como para precisar la información de un determinado Trap genérico.
- **Time-stamp:** tiempo transcurrido entre la última vez que se reinició el dispositivo de red y la generación del Trap.

1.5. SISTEMA OPERATIVO

Los sistemas operativos son el software básico de toda computadora que provee una interfaz, entre el resto de programas del ordenador, los dispositivos hardware y el usuario. Ver (**Ilustración 7**).

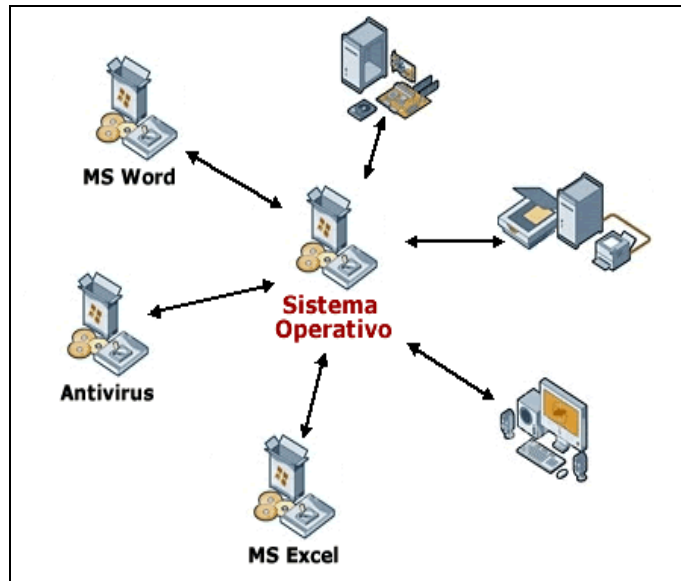


Ilustración 7: Sistema Operativo

Fuente: <https://www.masadelante.com/faqs/sistema-operativo>

Los sistemas operativos usados en el proyecto son Windows y Linux como todo el desarrollo se realizara en un solo computador se hará uso de máquinas virtuales las mismas que simularan a los dos servidores y les permitirán ejecutar programas.

En la (**Ilustración 8**), detallo algunos ejemplos de sistemas operativos:

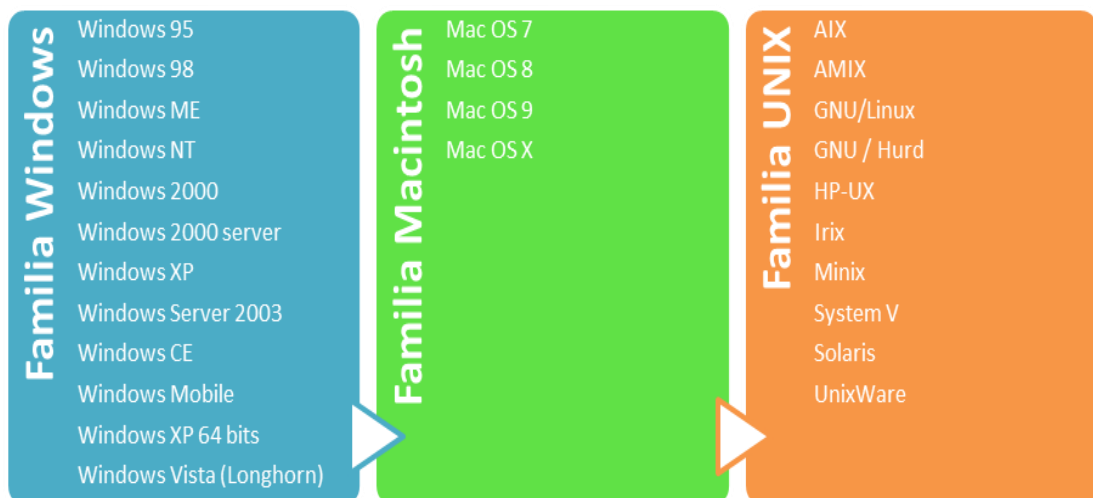


Ilustración 8: Ejemplos de sistemas operativos.

Fuente: Jenny Patín Chimbo.

1.6. MONITOREO

El término Monitoreo de Sistemas de Telecomunicaciones describe el uso de un medio que verifica el estado de los diversos módulos que conforman dicho sistema, en busca de componentes defectuosos o con mal funcionamiento, para luego notificar dicha situación a los administradores de la red, mediante diversas modalidades. Los equipos que se incluyen en redes de comunicaciones en la actualidad son muy diversos, es común ver equipos muy amigables al usuario y de fácil gestión, intuitivos y que tienen la facilidad de ser monitoreados de manera nativa, como Routers, Switchs, o servidores entre otros. (Ibídem)

1.6.1. Tipos de monitoreo

Existe dos tipos de monitoreo, basadas en la influencia en el tráfico de la red a causa de las mediciones y la información que buscan obtener de ella, bajo estos criterios se puede clasificar las acciones de monitoreo como activo o pasivo.

1.6.1.1. Monitoreo activo

Este tipo de monitoreo se realiza introduciendo paquetes de pruebas en la red, o enviando paquetes a determinadas aplicaciones y midiendo sus tiempos de respuesta. Este enfoque tiene la característica de agregar tráfico en la red y es empleado para medir el rendimiento de la misma. Ver **Ilustración 9**.

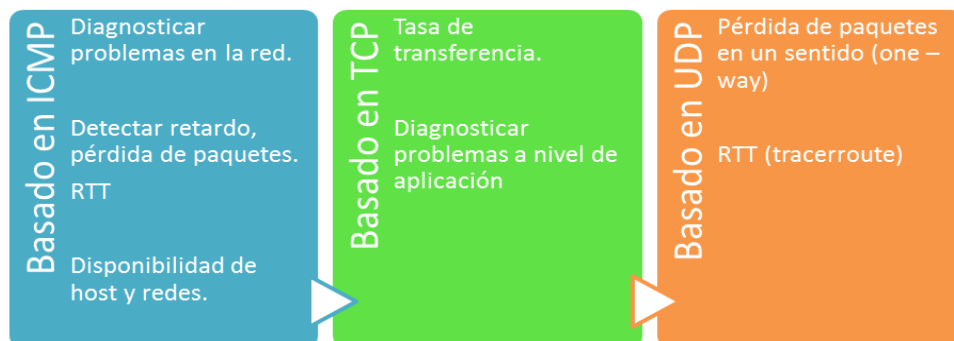


Ilustración 9: Formas de medir el tráfico en la red.

Fuente: Jenny Patín Chimbo.

1.6.1.2. Monitoreo pasivo

Este enfoque se basa en la obtención de datos a partir de recolectar y analizar el tráfico que circula por la red. Se emplean diversos dispositivos como sniffers, routers, computadoras con software de análisis de tráfico y en general dispositivos con soporte para SNMP, RMON y Netflow. Este enfoque no agrega tráfico a la red como lo hace el activo y es utilizado para caracterizar el tráfico en la red y para contabilizar su uso, algunos ejemplos de este tipo son:

Mediante SNMP:

Esta técnica es utilizada para obtener estadísticas sobre la utilización de ancho de banda en los dispositivos de red, estado de los dispositivos o comportamiento dentro de la red, para ello se requiere tener acceso a dichos dispositivos. Al mismo tiempo, este protocolo genera paquetes llamados traps que indican que un evento inusual se ha producido. Esta técnica será retomada a mayor detalle más adelante.

1.6.2. Software de monitoreo NAGIOS

Nagios es un sistema de monitorización de redes ampliamente utilizado, de código abierto, que monitorea los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos no sea el deseado. Entre sus características principales la monitorización de servicios de red (SMTP, POP3⁹, HTTP¹⁰, SNMP, etc.), la monitorización de los recursos de sistemas hardware, independencia de sistemas operativos, posibilidad de monitorización remota mediante túneles SSL¹¹ cifrados o SSH¹², y la posibilidad de programar plugins específicos para nuevos sistemas.

⁹ Protocolo de Oficina Postal

¹⁰ Protocolo de Transferencia de Hipertextos

¹¹ Secure Sockets Layer

¹² Interprete de ordenes seguro

Se trata de un software que proporciona una gran versatilidad para consultar prácticamente cualquier parámetro de interés de un sistema, y genera alertas, que pueden ser recibidas por los responsables correspondientes mediante (entre otros medios) correo electrónico y mensajes SMS¹³, cuando estos parámetros exceden de los márgenes definidos por el administrador de red. Nagios fue originalmente diseñado para ser ejecutado en GNU/Linux, pero también se ejecuta bien en variantes de Unix. Ver (**Ilustración 10**).

Interfaz:

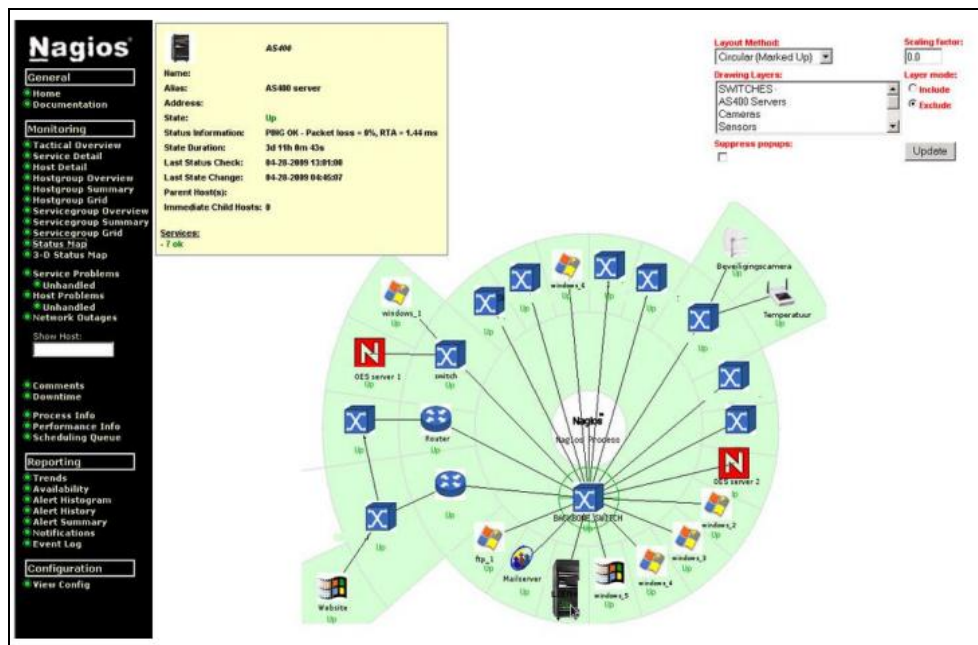


Ilustración 10: Ejemplo de Interfaz de Nagios

Fuente: <https://www.nagios.org/>

1.7. COMUNICACIÓN LOCAL INALÁMBRICA

La tecnología inalámbrica, utiliza ondas de radiofrecuencia de baja potencia, y una banda específica (espectro) de uso libre o privado para transmitir entre dispositivos. Las redes inalámbricas se clasifican de diversas maneras pero al tomarse en cuenta el alcance y la distancia para el desarrollo del control local se

¹³ Servicio de mensajes simples

toma en cuenta la red WLAN¹⁴ o estándar IEEE¹⁵ 802.11, también llamada wi-fi, que es una de las tecnologías de comunicación inalámbrica mediante ondas, ofrece desde 11 Mbit/s hasta 54 Mbit/s. (**Ilustración 11**). (Viana Patron, Nazar Torres, & May Poot, 2012).

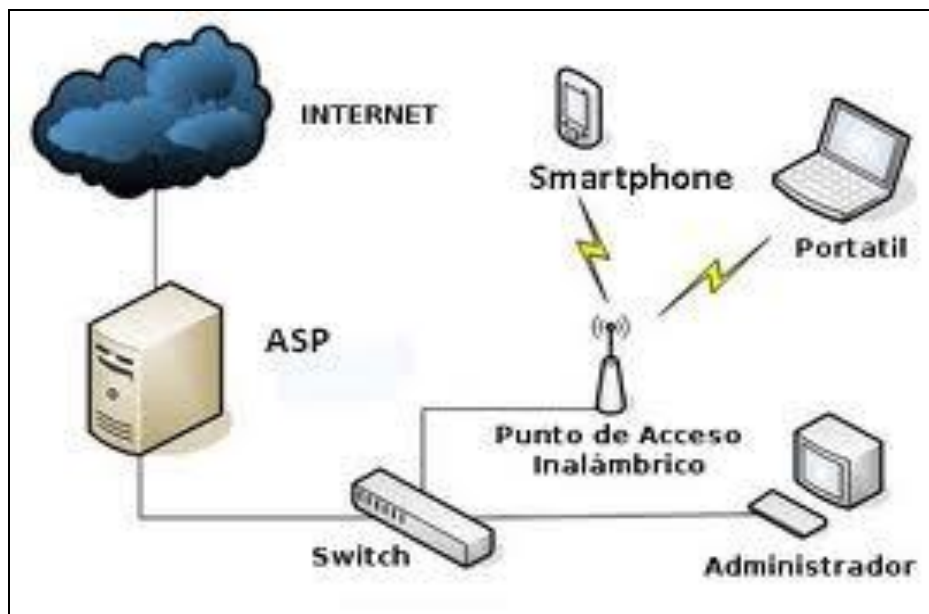


Ilustración 11: Conexión de red inalámbrica

Fuente: www.portal2.conlinux.net/Hot Spot

1.7.1. Zigbee.

Es un protocolo de comunicaciones inalámbricas basado en el estándar IEEE 802.15.4 y su función es la de solucionar los problemas de interoperabilidad, duración de la batería y costos de los protocolos propietarios en las aplicaciones de domótica (home automation).

Al igual que WiMax y WiFi, ZigBee posee una alianza de empresas, ZigBee

¹⁴Wireless LAN, red inalámbrica

¹⁵ Instituto de ingenieros eléctricos y electrónicos

Alliance.

ZigBee posee una arquitectura basada en el modelo OSI. El IEEE 802.15.4 define las dos capas más bajas: la Capa Física y la Subcapa de Control de Acceso al Medio de la Capa de Enlace de Datos, la cual se encarga de aislar los detalles de las tecnologías físicas a la capa de control de acceso al medio. Estas capas son utilizadas por ZigBee para crear un marco de trabajo para las aplicaciones.

La Capa Física puede trabajar en uno de tres rangos de frecuencias: 868 MHz, 915 MHz o 2.4 GHz con velocidades de hasta 20 Kbps, 40 Kbps y 250 Kbps y cuya utilización de bandas de frecuencias es Europa, JUL y el resto del mundo, respectivamente. Permite alcanzar distancias de hasta 100 metros con muy baja energía, lo que la hace práctica en la mayoría de las aplicaciones domóticas.

Algunos usos que se dan a ZigBee en la (Ilustración 12):



Ilustración 12: Usos de ZigBee

Fuente: www.zigbee.org

1.7.1.1. Tipos de dispositivos

Una red Zigbee está formada básicamente por 3 tipos de elementos.

- a) **El coordinador:** Es el nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red.
- b) **Los routers:** Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información.
- c) **Enddevice:** Los dispositivos finales no tienen capacidad de enrutar paquetes, deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro enddevice. Ver (**Ilustración 13**).
- d)

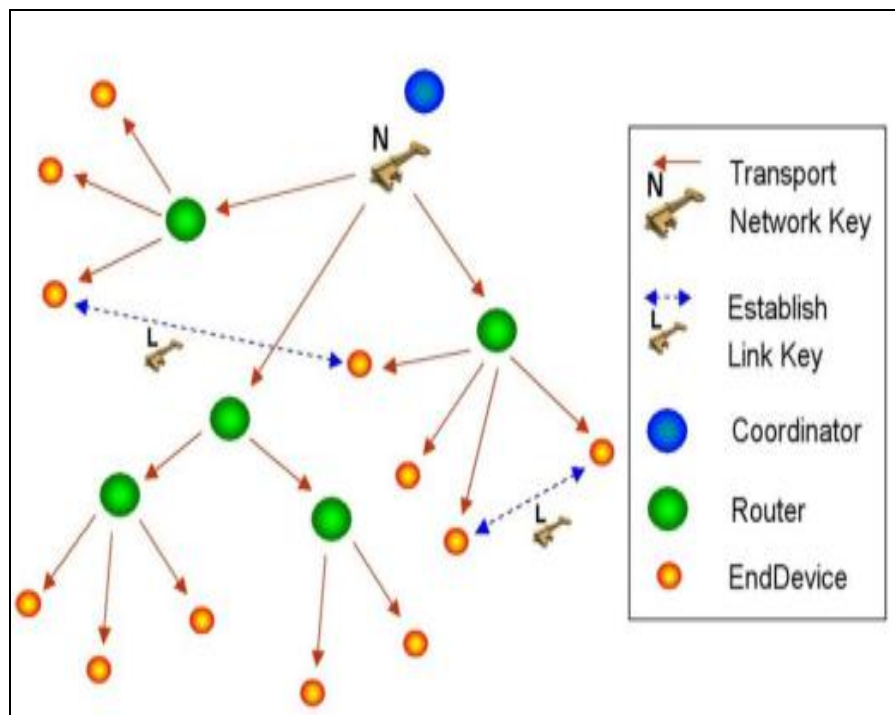


Ilustración 13: Tipos de Dispositivos ZigBee

Fuente: www.zigbee.org

1.7.1.2. Topología de red

ZigBee permite tres topologías de red:

- **Topología en estrella.** - el coordinador se sitúa en el centro.
- **Topología en árbol.** - el coordinador será la raíz del árbol.
- **Topología de malla.** - al menos uno de los nodos tendrá más de dos conexiones.

1.7.2. Módulo XBee

Módulo de Transmisión/Recepción de datos, basado en tecnología ZigBee.



Ilustración 14: Módulo XBee

Fuente: Jenny Patín Chimbo

1.7.3. Circuito básico Xbee

La (**Ilustración 15**) muestra las conexiones mínimas que necesita el módulo Xbee para poder ser utilizado, luego de esto, se debe con observar según el modo de operación que sea para la aplicación requerida por el usuario.

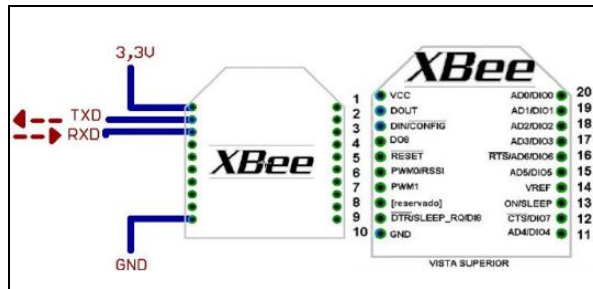


Ilustración 15: Circuito Básico XBee

Fuente: Datasheet

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un micro controlador, o directamente a un puerto serial utilizando algún convertor adecuado para los niveles de voltaje.

1.8. MICRO CONTROLADORES

Un microcontrolador es un circuito integrado programable, que puede ejecutar las órdenes cargadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento (CPU¹⁶), memoria y periféricos de entrada/salida. Ver (**Ilustración 16**). (Reyes, 2008).

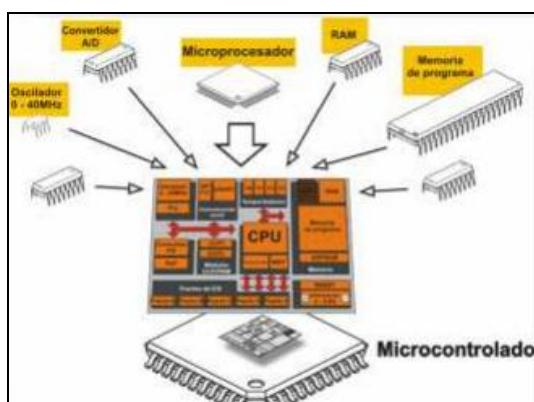


Ilustración 16: Partes que integran un microcontrolador

¹⁶ Unidad Central de Proceso

Fuente: mikroe.com/

Al momento de ser fabricado, el microcontrolador no contiene datos en la memoria ROM¹⁷. Si se desea realizar o controlar algún proceso es necesario generar o crear y luego cargar en la EEPROM¹⁸ o equivalente, algún programa, el cual puede ser escrito en lenguaje ensamblador; sin embargo para que el programa pueda ser grabado en la memoria, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

1.8.1. Tarjeta Arduino

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa de circuito impreso que contiene un microcontroladores de la marca “ATMEL” que cuenta con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación *processing*. El dispositivo conecta el mundo físico con el mundo virtual, o el analógico con el digital controlando, sensores, alarmas, sistemas de luces, motores, sistemas de comunicaciones y actuadores físicos. (Banzi, 2012)

1.8.1.1. Lenguaje de programación

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, debido a que usa la transmisión serial de datos soportada por la mayoría de los lenguajes.

Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para

¹⁷ Memoria de Solo Lectura

¹⁸ ROM Programable y Borrable

permitir una comunicación fluida. Los principales lenguajes de programación compatibles con Arduino son:

- C
- C++ (mediante libSerial o en Windows)
- C#
- Java
- Matlab
- Python
- Visual Basic .NET

1.8.1.2. Entorno

El entorno de desarrollo de Arduino contiene un editor de texto para escribir los códigos, un área de mensajes, una consola de texto para el puerto serial, 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 Arduino para comunicarse y cargar programas. (Ibídem).



Ilustración 17: Entorno Arduino para Microsoft Windows

Fuente: Jenny Patín Chimbo

1.8.1.3. Bocetos

El software escrito para usar en Arduino se denomina Bocetos (Sketches en inglés). Estos se escriben en el editor del texto del entorno Arduino. Los sketches son guardados con la extensión de archivo *.ino*. Esta extensión ofrece funciones como cortado/pegado y también para buscar y reemplazar texto.

El área de mensajes ofrece ayuda mientras guarda, exporta y también muestra errores. La consola muestra la salida de texto por medio del entorno del Arduino, incluyendo mensajes de error completos e información adicional. La esquina inferior derecha de la pantalla muestra el puerto usado y la tarjeta conectada. Los botones de la barra de tarea permiten verificar y cargar programas, crear, abrir y guardar sketches, y abrir el monitor serial.



```

AnalogReadSerial5
/**AnalogReadSerial
 * Lee una entrada analógica en el pin 0, imprime el resultado al
 * monitor serial
 * Conecte el pino de entrada de un potenciómetro al pin A0,
 * y los pines exteriores a +5V y tierra
 * This example code is in the public domain.
 */
// esta rutina se inicia una vez presionado el botón reset:
void setup() {
  // se inicializa la comunicación serial a 9600 bits por segundo:
  Serial.begin(9600);
}
// esta rutina loop corre una y otra vez :
void loop() {
  // se lee la entrada analógica del pin 0:
  int sensorValue = analogRead(A0);
  // se muestra el valor leído:
  Serial.println(sensorValue);
  delay(1); // retraso entre los valores para estabilidad
}

```

Ilustración 18: Boceto con un ejemplo de lectura analógica serial

Fuente: Jenny Patín Chimbo

1.8.1.4. Libro de bocetos

El entorno Arduino usa el concepto de *Libro de Bocetos* (*Sketchbook en inglés*). Es un lugar estándar en donde se guardan los programas o bocetos. Los bocetos dentro de un sketchbook pueden ser abiertos mediante el menú *Archivo Sketchbook* o por medio del botón *Abrir* en la barra de tareas.

La primera vez que se inicie el software del entorno Arduino, este creará automáticamente un directorio para el sketchbook. Se puede ver o cambiar la localización del sketchbook desde el cuadro de diálogo de *Preferencias* en el menú *Archivo*.

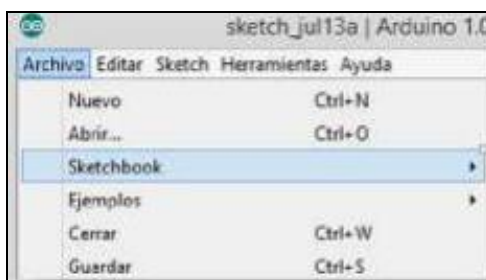


Ilustración 19: Menú para acceder al sketchbook

Fuente: Jenny Patín Chimbo

1.8.1.5. Librerías

Las librerías dentro de entorno Arduino proveen de una funcionalidad extra para usar en los bocetos, como por ejemplo, trabajar con hardware adicional o manipular datos externos. Para usar una librería en un boceto, se debe seleccionarla desde el menú *Sketch>Importar Librería*. Esto insertará una o más declaraciones del tipo *#include* al inicio del boceto y compilará la librería con el boceto.

A causa de que las librerías son cargadas a la tarjeta conjuntamente con el boceto, estas incrementan la cantidad de espacio necesario que requiere el

programa. Si un boceto no necesita de una librería, simplemente se debe borrar la declaración *#include* en el inicio del código.



Ilustración 20: Menú de librerías disponibles en el software Arduino

Fuente: Jenny Patín Chimbo

1.8.1.6. Monitor Serial

El monitor serial es el encargado de mostrar los datos que están siendo enviados desde la tarjeta Arduino (sea mediante conexión USB o Serial). Se puede acceder al monitor serial mediante el botón en forma de lupa que se encuentra en la parte superior derecha de la ventana del programa. Para enviar datos a la tarjeta, se debe ingresar el texto deseado y luego presionar el botón *Send* o simplemente dar *Enter*.

Se debe escoger la tasa de baudios desde el menú inferior, y esta debe coincidir con la tasa designada en la declaración *Serial.begin* del boceto.

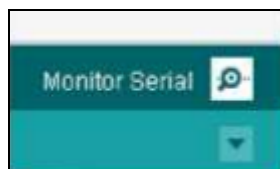


Ilustración 21: Botón de acceso al monitor serial

Fuente: Jenny Patín Chimbo

1.8.1.7. Hardware externo para terceros

El soporte para hardware adicional de terceros, puede ser añadido al directorio hardware del directorio del sketchbook. Las plataformas que allí se instalen pueden incluir definiciones de tarjetas (las cuales aparecerán en el directorio de tarjetas), librerías integrales, iniciadores de carga y definiciones de programación.

Para instalar una de estas opciones, se debe crear e directorio hardware, luego descomprimir la plataforma de terceros dentro de su propio sub-directorio.

Se debe tener la precaución de no usar "arduino" como nombre de subdirectorio, puesto que se sobrescribirá el directorio integrado de la plataforma Arduino. Como ejemplo de hardware de terceros, se tiene los sensores.

1.8.1.8. Tarjeta Shield Ethernet

Las "Shields" son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Las diferentes "shields" siguen la misma filosofía que el conjunto original: son fáciles de montar y su costo de producción es bajo. Las shields suelen ser diseños bastante simples y en general de código abierto, el diseño suelen ser publicados libremente.

1.9. ROUTER

Un Router es un dispositivo que proporciona conectividad a nivel de red o nivel tres en modelos OSI. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra. (CCNA).



Ilustración 22: Router

Fuente: www.portal2.conlinux.net/Hot Spot

1.10. SENSORES

Un sensor es un dispositivo que puede convertir magnitudes físicas o químicas, llamadas variables de instrumentación, en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc. (es.wikipedia.org/wiki/Sensor, 2014). Al estar conectado siempre a la variable de instrumentación, se dice que aprovecha una de sus propiedades, con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo.

Como ejemplo se puede citar un fototransistor, que determina la activación o desactivación de una bombilla dependiendo de la luz ambiental presente en ese momento. En otras palabras, un sensor puede convertir un tipo de energía a un impulso eléctrico.

Los sensores se clasifican dependiendo de la señal o magnitud de entrada al dispositivo, la señal de salida, y la naturaleza de la señal generada. (Ortega, 2014).

1.10.1. Características del Sensor

Dentro de las características principales de sensor se tiene:

- **Rango de medida:** dominio en la magnitud medida en el que puede aplicarse el sensor.
- **Precisión:** es el error de medida esperado.
- **Offset o desviación de cero:** valor de la variable de salida cuando la variable de entrada es nula.
- **Linealidad o correlación lineal.**
- **Sensibilidad:** es la relación que existe entre la variación de la magnitud de entrada y la variación de la magnitud de salida
- **Resolución:** mínima variación de la magnitud de entrada que puede apreciarse a la salida.
- **Rapidez de respuesta:** Es la rapidez con la que el sensor entrega los datos al dispositivo al cual está conectado. Depende de la magnitud que se desea medir y también de la capacidad del sistema al que está conectado para seguir las variaciones de dicha magnitud
- **Repetitividad:** Es el error esperado al repetir varias veces la misma medida.

Depende del número de repeticiones y de la precisión del sensor

1.10.2. Según el tipo de señal de entrada:

- **Mecánica:** Son aquellos sensores que miden una propiedad extrínseca de

la materia, o la interacción entre dos fuerzas de distinto tipo. Como ejemplo la longitud, masa, velocidad, aceleración, fuerza, torque, presión, intensidad acústica, longitud de onda.

- **Química:** Son los sensores que pueden identificar las propiedades intrínsecas de la materia, como por ejemplo concentración, composición, potencial de oxidación/reducción, pH.
- **Eléctrica:** Son los sensores que son capaces de captar la interacción de los electrones en una corriente, como por ejemplo está el voltaje, corriente eléctrica, constante dieléctrica, polarización, campo eléctrico, frecuencia dipolar
- **Magnética:** Son los sensores que miden cambios en la intensidad de un campo magnético, como también su densidad y permeabilidad.
- **Térmica:** En este grupo están los sensores que detectan la variación de la temperatura en el ambiente o en un determinado material, así como también el flujo de calor que se puede transmitir de un cuerpo a otro.

1.10.3. Según el tipo de señal de salida

- **Sensores Analógicos:** Componen la mayor parte de sensores disponibles en el mercado, y son aquellos que entregan su señal continua en el tiempo, como por ejemplo los sensores generadores de señal.
- **Sensores Digitales:** Son aquellos sensores que poseen una salida de carácter discreto.

Como ejemplo los sensores de posición, sensores codificadores incrementales, sensores auto resonantes, entre otros.

1.10.4. Según la naturaleza de la señal generada

- **Sensores pasivos:** Son aquellos que utilizan una señal externa o auxiliar para poder realizar la medición de la magnitud. Ejemplos de estos sensores son los sensores de luz, de temperatura, de parámetros variables.
- **Sensores activos:** Se los conoce también como sensores generadores de señal y son aquellos que no requieren de una señal externa para poder realizar la medición de una magnitud, ya que son capaces de emitir una señal propia.

1.10.5. Estructura básica de un sensor (García Rodríguez)

Generalmente, los sensores se componen de los siguientes elementos:

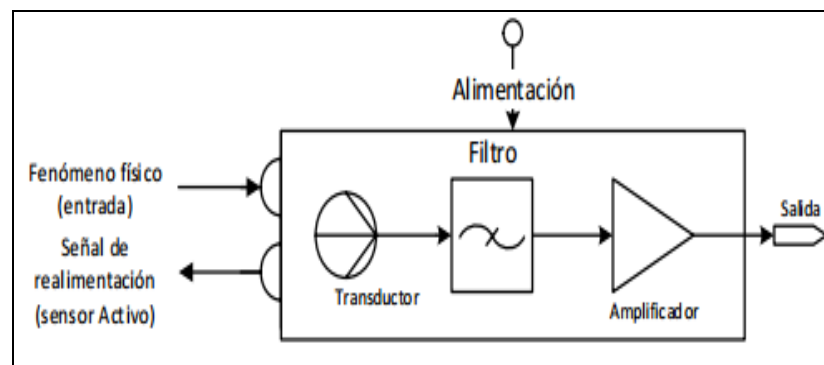


Ilustración 23: Estructura básica de un sensor

Fuente: Alex Molina Espinoza

- **Transductor:** Es el que se encarga de convertir la variación de la magnitud recibidas del fenómeno físico en forma de energía, en variaciones de una magnitud eléctrica, es decir las transforma en una señal eléctrica.
- **Filtro:** Luego de recibida la señal eléctrica, el filtro se encarga de diferenciar entre la frecuencia recibida, y que será usada en el proceso de censado, del ruido o cualquier otra señal que no sea la que se está

midiendo.

- **Amplificador:** Dado que la señal convertida a magnitud eléctrica es de muy baja energía, la mayoría de las veces se requiere alterar esta señal para que pueda obtenerse una adecuada lectura de los datos, ya sea necesaria una amplificación, una modulación, etc.

1.11. INVERNADERO

Un invernadero (o invernáculo) es un lugar cerrado, estático y accesible a pie, que se destina a la horticultura, dotado habitualmente de una cubierta exterior translúcida de vidrio o plástico, que permite el control de la temperatura, la humedad y otros factores ambientales para favorecer el desarrollo de las plantas. Ver (**Ilustración 24**). (Ávalos G. Luisa).



Ilustración 24: Invernadero

Fuente: www.invernaderos-agricolas.com/2010/10/agricultura-presenta-software-gestion.html

1.12. SISTEMA DE ACTUADORES

Un actuador es un dispositivo capaz de transformar energía hidráulica, neumática o eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado. Como ejemplos se tiene: el ventilador, la bomba y el calefactor.

CAPÍTULO II

2. METODOLOGÍA.

2.1. TIPO DE ESTUDIO.

Descriptivo-Explicativo.- Con esta investigación se describió el problema de la mayoría de los invernaderos, y a la vez se intentó explicar y comprobar los resultados una vez que el prototipo fue implementado.

2.2. MÉTODOS, TÉCNICAS E INSTRUMENTOS

2.2.1. Métodos

2.2.1.1 Deductivo.

Se usó este tipo de metodología ya que es necesaria la observación de fenómenos generales con el propósito de señalar las verdades particulares contenidas explícitamente en la situación general.

2.2.1.2 Analítico.

Se usó este tipo de metodología ya que es necesaria para entender en forma particular el funcionamiento de cada uno de los componentes necesarios para el desarrollo del prototipo, además del software que manejará dicho sistema, y la forma de interactuar de cada uno de los componentes entre sí para que el dispositivo desempeñe su función.

2.2.2. Técnica

2.2.2.1. Observación

Esta técnica me permite recoger la mayor cantidad de información que me

ayude a resolver el problema, y me dé la solución para desarrollar e implementar el sistema.

2.2.2.2. Instrumentos

Se utilizó libros, folletos, archivos, páginas web, datasheets, blogs, entre otros para la selección de la información que se refiere al diseño e implementación del mismo.

2.3. POBLACIÓN Y MUESTRA

2.3.1. Población

La población, es decretada por los datos obtenidos en las pruebas aleatorias de acuerdo a distintas formas de administración entre el usuario y el dispositivo, estableciendo de esta manera la efectividad de la comunicación.

2.3.2. Muestra

La población carece de registro definido, es decir que es desconocida ya que el número de pruebas puede ser infinito, por lo tanto la muestra es establecida de acuerdo al cálculo con la siguiente fórmula:

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

Significado de cada parámetro:

Z_{α}	Distribución de Gauss donde $z_{\alpha=0.005}=2.575$
p	Prevalencia esperada de parámetro a evaluar, el valor tomado es un porcentaje alto de prevalencia $p=0.9$ es decir el 90 %.
q	$1-p$
i	error considerado que se prevé cometer es del $i=0.01$

$$n = \frac{1.96^2 * 0.9 * (1 - 0.9)}{0.1^2}$$

$$n = \frac{3,8416 * 0.9 * 0.1}{0.01}$$

$$n = 34,47 \approx 34$$

2.3.3 Hipótesis

“El diseño e implementación de un sistema de monitoreo y control de humedad y temperatura para invernaderos con administración SNMP permitirá mejorar el control local de los sensores”.

2.4. OPERACIONALIZACIÓN DE VARIABLES

En la **Tabla 3** se indica la operacionalización de variables dependiente e independiente del proyecto. Está formado de una variable dependiente y una variable independiente, el concepto de cada uno, los indicadores y las técnicas que se usó para la realización de dicha tabla.

Tabla 3: Operacionalización de variables independiente y dependiente

Variable	Concepto	Indicadores	Técnicas e Instrumentos
Independiente El diseño e implementación de un sistema de monitoreo y control de humedad y	Diseño e implementación de un sistema de monitoreo y control por medio de un conjunto de dispositivos bajo ciertas condiciones alineadas a un mismo objetivo.	Conexión con dispositivos. Niveles de los	Datasheets (Características técnicas de los sensores) Programación en Arduino

temperatura para invernaderos con administración SNMP.		sensores.	MIB Comandos
Dependiente Permitirá el control local de los sensores.	Mejorar el monitoreo y administración de tal manera que los tiempos de respuesta de los actuadores sea mínimo. (Calefactor, bomba y ventilador).	Tiempo de respuesta de la red WiFi	Red WiFi Sensor DHT22 Sensor Soil

Fuente.- Jenny Patín Chimbo

2.5. PROCEDIMIENTOS

Para el desarrollo de este dispositivo, se solicitó seguir una serie de sendas y períodos, que en conjunto, cumplirían con el objetivo planteado. Cada etapa desempeña una función específica dentro del esquema general, y también sirve de apoyo para la siguiente etapa, haciendo del dispositivo implementado un sistema funcional.

La primera etapa, se escogió el cultivo para el que se desarrolla el dispositivo, por varias razones positivas se escoge el cultivo del tomate riñón, ya que en la actualidad, el tomate riñón es la hortaliza más cultivada en el mundo, por su contenido nutricional y su demanda en la dieta diaria. El tomate riñón es uno de los cultivos más importantes en invernadero, por ser una hortaliza de

consumo masivo, su popularidad aumenta debido a su alta producción y rentabilidad. Sin duda, esta alternativa es un rubro significativo en la economía actual de las familias campesinas. Los parámetros declarados para el prototipo se indican en la siguiente **Tabla 4**.

Tabla 4: Parámetros de temperatura y humedad del tomate riñón

Temperatura Optima	21-27 C
Temperatura día	23-26 C
Temperatura Noche	14-17 C
Humedad Ralativa	65-75%
Goteo	1.5 a 2 lt./día

Fuente.- Jenny Patín Chimbo

Se eligió utilizar un sensor DHT22, puesto que es un dispositivo que permite un fácil ensamblaje en la maqueta, además de tener un costo accesible para la adquisición de varios en caso de accidentes propios de las pruebas. Se eligió también el sensor soil por su facilidad de uso características que satisficían al manejo del sistema.



Ilustración 25: Diagrama de procedimiento.

Fuente: Jenny Patín Chimbo.

En lo que respecta a la comunicación se usó los módulos Xbee, para la transmisión y recepción de datos; el protocolo de comunicación que usa es Zigbee, por tener un mejor manejo de protocolo de errores, la distancia de comunicación es de 16 Km y la facilidad de encontrar en el mercado nacional.

El montaje de los componentes se dividió en dos etapas:

La primera etapa es la transmisora (Tx) está formado por un Shield Xbee, tarjeta Arduino 1, un módulo Xbee Pro, dos sensores DHT22 y un sensor de humedad de suelos y un Shield relé de 4 canales.

Para alimentar de sistema de energía se usó un adaptador de 5 VDC-2 A. Para comunicar la red se usó un router Starbrige 1532a

La segunda etapa es la receptora (Rx) está formado por un Arduino Mega, un Shield Ethernet W5100, un Shield Xbee y un Xbee PRO.

Para la parte del monitoreo del sistema se eligió un software libre Nagios que esta está programado bajo sistema operativo Linux, donde se reflejaran tres diagramas de bloque que indicaran la temperatura, humedad relativa y humedad del suelo, los cuales tienen parámetros los cuales indican que el sistema está funcionando correctamente o a su vez envíe una alerta mediante una alerta al usuario para que pueda activar de forma manual o automática los actuadores.

2.6. PROCESAMIENTO Y ANÁLISIS.

2.6.1. Identificación y estudio del prototipo

Para diseñar el prototipo, se realizó observaciones del sistema de monitoreo encontrándose con una desventaja que no todos los usuarios contaban con una red interna adecuada. Debido a esto, el diseño a implementar se enfocó en atender

este principal inconveniente, ya que resulta beneficioso para el usuario disponer con una gestión de redes adecuada para el uso del mismo en todo momento.

De esta manera, el mejor diseño que se encontró para poder cubrir estas necesidades; se administró la red mediante SNMP, el cual es capaz de enviar una alerta mediante un e-mail al usuario he indicar que su invernadero está sufriendo cambios de temperatura mediante los sensores DHT22 y sensor Soil, los sensores tienen parámetros establecidos para el cultivo de tomate riñón.

Los sensores se ubicaron dentro del invernadero, tomando en cuenta los parámetros establecidos para el cultivo del tomate riñón donde se activan de forma manual y automática los actuadores que se encuentran dentro del invernadero.

Los módulos XBee se encargan del envío y transmisión de la información mediante la comunicación inalámbrica. La parte del monitoreo del sistema se realiza mediante un software libre Nagios está configurando bajo Linux, por su facilidad de manejo.

2.6.2. Selección y estudio de los equipos electrónicos y sensores.

Los equipos electrónicos y sensores a usarse en la construcción del prototipo se detallan en la **Tabla 5**.

Tabla 5: Equipos electrónicos y sensores

CANTIDAD	DISPOSITIVO/SENSOR
2	Arduino MEGA R3
1	Shield Ethernet
1	Shield Rele 4Ch 5V

2	Módulos XBEE Pro
2	Sensores DHT22
1	Sensor de humedad de suelos SOIL
2	Adaptadores de 5V-2 a
1	Router
1	Micro Ventilador 5V
1	Router
1	Mini Bomba de 12V

Fuente.- Jenny Patín Chimbo

Los dispositivos y sensores fueron escogidos muy cuidadosamente debido a que los equipos deben ser administrados por medio de una red.

Los sensores están encargados de recopilar todos los cambios ambientales dentro del invernadero, a su vez enviar la información a la tarjeta Arduino, los actuadores deben dar una respuesta inmediata cuando haya un cambio de clima

2.6.2.1. Arduino MEGA 2560 R3

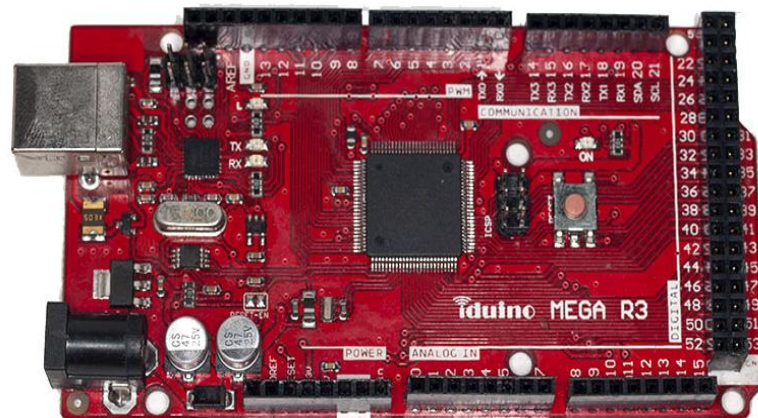


Ilustración 26: Arduino Mega 2560 R3.

Fuente: Jenny Patín Chimbo

El Arduino Mega 2560 es una placa electrónica basada en el chipset Atmega 2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertas seriales),

un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB, adaptador de CA o la batería a CC para empezar. A continuación se muestra las características de la tarjeta Arduino Mega. Ver (**Tabla 6**).

Tabla 6: Características de la tarjeta Arduino MEGA

Microcontrolador	Atmega 2560
Voltaje de funcionamiento	5 Vcd
Voltaje de entrada (recomendado)	7-12 Vcd
Pines E/S digitales	54 (14 proporcionan salida PWM)
Pines de entrada analógica	16
Intensidad por pin	40 mA
Intensidad en pin 3.3Vcd	50 mA
Memoria Flash	256 kB de los cuales 8 kB son usadas por el bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Fuente.- Jenny Patín Chimbo

El software de programación consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing /Wiring y el cargador de arranque que es ejecutado en la placa.

El entorno de programación no es distinto de los lenguajes ensambladores utilizados habitualmente, por lo que comparten la mayoría de comandos, cambiando solamente la sintaxis de las instrucciones.

Características eléctricas.

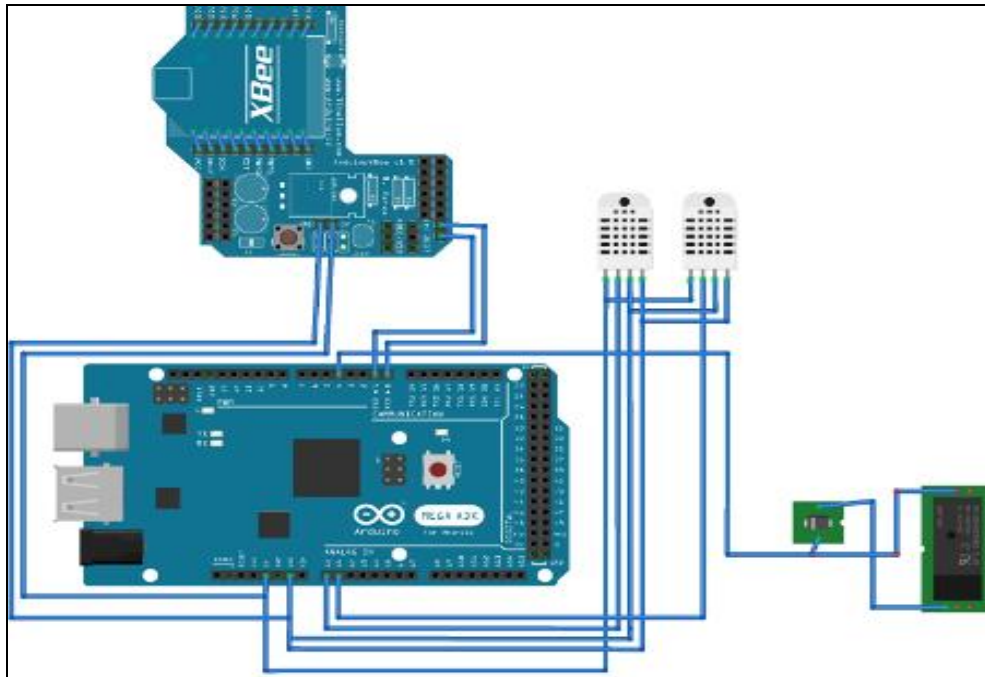


Ilustración 27: Arduino Mega 2560 R3

Fuente: Jenny Patín Chimbo

La tarjeta Arduino es considerada como el cerebro del sistema, dado que se encarga de recibir e interpretar los datos que vienen de los sensores.

2.6.2.2. Tarjeta Shield

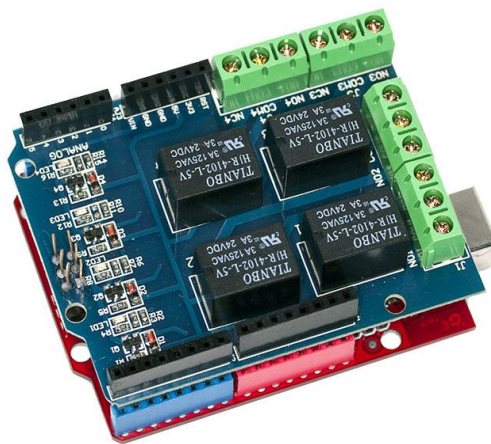


Ilustración 28: Shield relé de 4 canales 5V.

Fuente: Jenny Patín Chimbo

Para la parte del circuito de control de temperatura y humedad se optó por la utilización de un shield relé de 4 canales de 5v para el control de la ventilación, humedad y calor hacia el invernadero. Ver **Ilustración 28**.

Las características técnicas se determinan en la siguiente **Tabla 7**.

Tabla 7: Características Shield relé de 4 canales 5V

Voltaje de la energía:	4.75 ~ 5V
Consumo de energía (a 5 V)	180 mA
Apertura y cierre por minuto	30 veces, cada canal máximo
Convierte voltaje:	125V AC / DC 24V
Cada canal máximo convierte la corriente:	3 A

Fuente: Jenny Patín Chimbo

Características eléctricas.

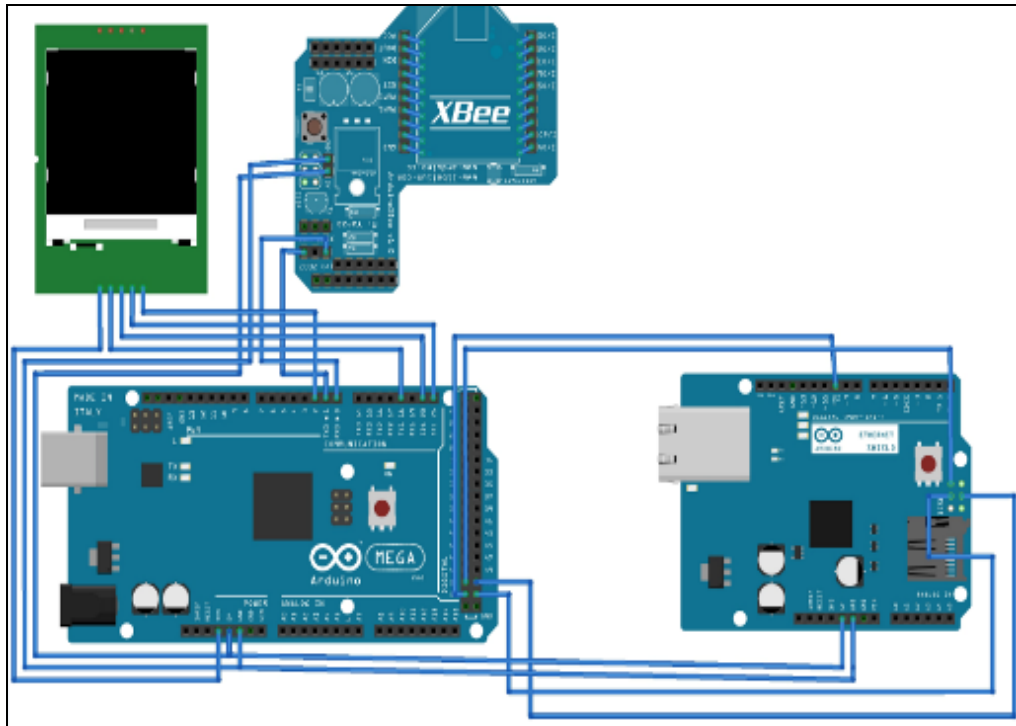


Ilustración 29: Diagrama de conexión tarjeta Shield y los sensores.

Fuente: Jenny Patín Chimbo

2.6.2.3. Modulo Ethernet



Ilustración 30: Arduino Ethernet placa vista frontal- vista trasera

Fuente: Arduino Ethernet

Arduino Ethernet es una placa electrónica basada en el chipset ATmega328. Cuenta con 14 pines digitales de entrada / salida, 6 entradas analógicas, un oscilador de cristal de 16 MHz, un RJ45 de conexión, un conector de alimentación, una cabecera ICSP, y un botón de reinicio.

Las características técnicas se determinan en la siguiente **Tabla 8**.

Tabla 8: Características de la tarjeta Arduino Ethernet

Microcontrolador	ATmega328
Voltaje de funcionamiento	5 V
Plug Voltaje de entrada (recomendado)	7-12 V
Tensión enchufe de entrada (limites)	6-20 V
Voltaje de entrada PoE	36-57 V

Digital pines E/S digitales	14 (de las cuales 4 proporcionan salida PWM)
Arduino prendedores reservados	
	10 a 13, utilizando para SPI
	4 utilizado para la tarjeta SD
	2 W5100 de interrupción (cuando puente)
Pines de entrada analógica	6
Corriente DC por pin I/O	40 mA
Corriente DC de 3.3 V pin	50 mA
Memoria Flas	32 KB (ATM mega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1KB (ATmega328)
Velocidad del reloj	16 MHz
Ethernet Controller Embedded W5100 TCP/IP	
Power Over Ethernet listo mangnética jack	
Tarjeta Micro SD, con traductores de voltaje activos	

Fuente.- Jenny Patín Chimbo

Este equipo se usó para la administración SNMP, estará conectada al router y a la PC del usuario el cual permitirá enrutar la red mediante la IP 192.168.1.6 con mascara 255.255.255.0.

2.6.2.4. Xbee.



Ilustración 31: Tarjeta Xbee.

Fuente: Jenny Patín Chimbo

Los módulos Xbee son dispositivos que integran un transmisor - receptor de ZigBee y un procesador en un mismo módulo, lo que le permite a los usuarios desarrollar aplicaciones de manera rápida y sencilla.

A continuación se ve en la (**Tabla 9**) las características del módulo Xbee.

Tabla 9: Características tarjeta Xbee.

Alimentación	3.3V
Velocidad de transferencia	250kbps Max
Potencia de salida	1mW o 60mW (+18dBm)
Alcance	90metros o 1500 metros aprox.
Certificado	FCC
pinos ADC de 10-bit	6
pinos digitales IO	8
Encriptación	128-bit
Configuración	Local o de forma inalámbrica
Comandos	AT o API

Fuente.- Jenny Patín Chimbo

Funcionan con la pila 802.15.4 (la base de ZigBee) y funcionan mediante un simple protocolo serie TTL. Permiten una comunicación bidireccional entre microcontroladores, ordenadores o prácticamente cualquier cosa que disponga de un puerto serie o Serie/USB a nivel TTL.

Soporta funciones de comunicación punto a punto o en red con varios módulos XBee, también llamado *Mesh Network*

Características eléctricas.

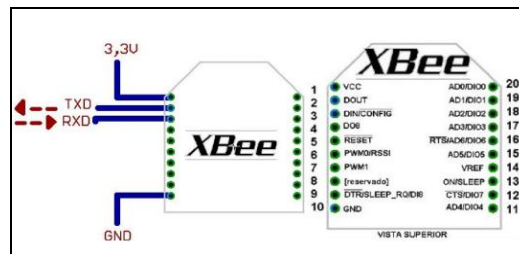


Ilustración 32: Diagrama de la Tarjeta Xbee.

Fuente: Jenny Patín Chimbo

2.6.2.5. Sensor DHT22

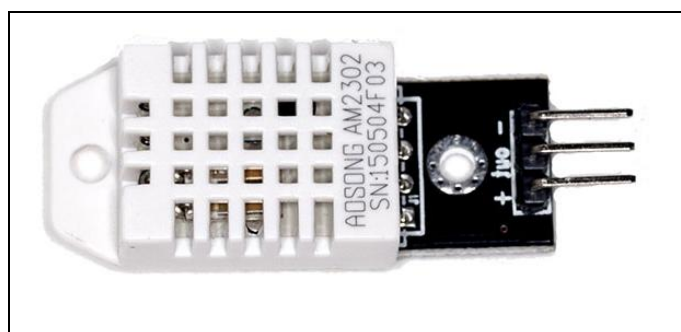


Ilustración 33: Sensor DHT22.

Fuente: Jenny Patín Chimbo

El sensor DHT22, utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, pero

requiere sincronización cuidadosa para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimas cada 2 segundos. (Ilustración 33).

A continuación presento las características del sensor.

Tabla 10: Características de Sensor DHT22

Alimentación	3.3v – 5.5v, tomando como valor recomendado 5v.
Resolución decimal	Es decir, los valores tanto para humedad como para temperatura serán números con una cifra decimal.
Tiempo de muestreo:	2 segundos, es decir, sólo puede ofrecer datos cada 2 segundos.
Rango de valores de temperatura	Desde -40°C hasta 80°C de temperatura.
Precisión de la temperatura	$\pm 0.5^{\circ}\text{C}$, $\pm 1^{\circ}\text{C}$ como máximo en condiciones adversas.
Tiempo de respuesta de temperatura	<10 segundos, es decir, de media, tarda menos de 10 segundos en reflejar un cambio de temperatura real en el entorno.
Rango de valores de humedad relativa	Desde 0% hasta 99.9% de Humedad Relativa.
Precisión de la humedad relativa	: $\pm 2\% \text{RH}$, a una temperatura de 25°C.

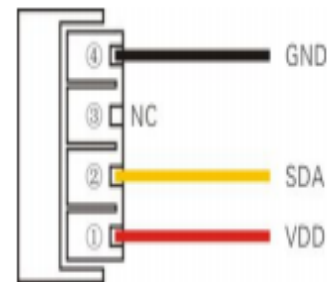
Tiempo de respuesta de humedad relativa	<5 segundos, es decir, de media, tarda menos de 5 segundos en reflejar un cambio de humedad relativa real en el entorno. Además, para darse esta afirmación, los tests indicaron que la velocidad del aire debe ser de 1 m/s.
---	---

Fuente: Jenny Patín Chimbo

Características eléctricas.

Para la conexión eléctrica del sensor se debe tomar en cuenta los pines de conexión.

Pin	Name	Description
①	VDD	Power (3.3V-5.5V)
②	SDA	Serial data, bidirectional port
③	NC	Empty
④	GND	Ground



PIC1: AM2302 Pin Assignment

Ilustración 34: Pines de conexión

Fuente: Datasheet

En el prototipo se usa 2 sensores DHT22 de temperatura y humedad que se ubicaron en el invernadero.

2.6.2.6. Sensor de humedad del suelo soil

El sensor Soil-Clik utiliza tecnología probada para medir la humedad en la zona radicular.

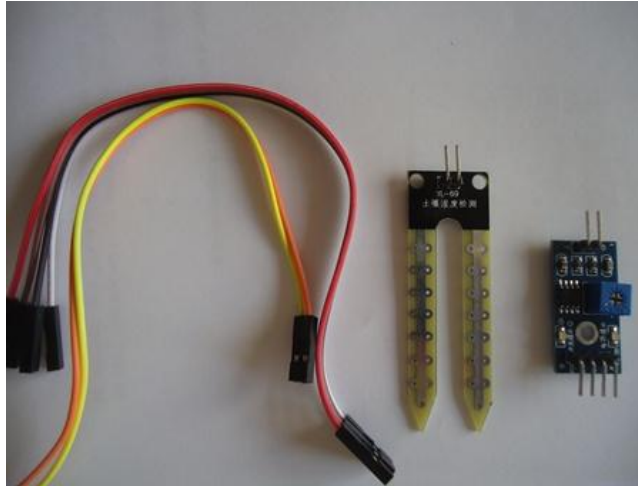


Ilustración 35: Encapsulado del Sensor Soil.

Fuente: Jenny Patín Chimbo

Soil-Clik simplifica la detección de humedad del suelo. Cuando la sonda detecta que el suelo ha alcanzado el nivel de humedad deseado, se cerrará el riego, evitando malgastar agua.

El mismo que está formado por una salida digital y una salida analógica.

A continuación presento las características del sensor. Ver (**Tabla 11**).

Tabla 11: Características Sensor Soil

Alimentación	3,3 V o 5 V
Voltaje de salida	0 ~ 4.2v
Corriente	35 mA
Salida analógica	(cable azul)
GND	(alambre Negro)
Alimentación	(cable rojo)
Tamaño	60x20x5mm
Final superficial:	Oro de la inmersión

Fuente.- Jenny Patín Chimbo

Características eléctricas.

Para la conexión eléctrica del sensor se debe tomar en cuenta los pines de conexión.

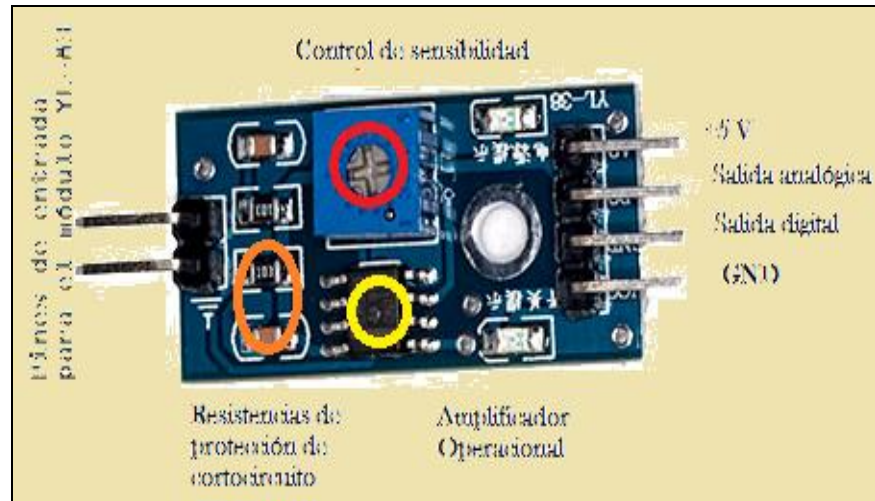


Ilustración 36: Pines de conexión.

Fuente: Editado por Jenny Patín Chimbo

Para el prototipo se usó 1 sensor de humedad del suelo, por su facilidad de uso que se puede encontrar en el mercado. Su precio es accesible y es muy de fácil uso.

2.6.2.7. Pantalla Oled 128x64 Arduino

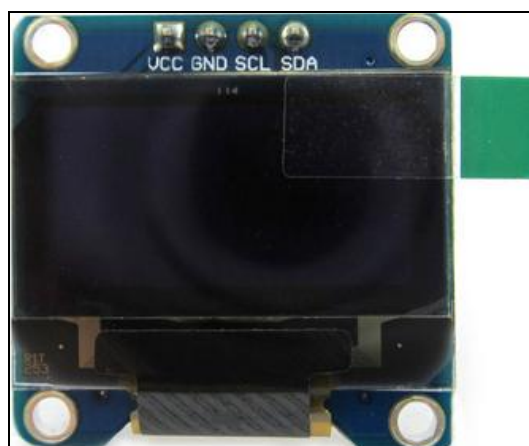


Ilustración 37: Pantalla Oled 128x64 Arduino

Fuente: Jenny Patín Chimbo

Estas pantallas son pequeñas, sólo alrededor de 1 "de diámetro, pero muy fácil de leer debido al alto contraste de una pantalla OLED.

A continuación presento las características de la pantalla Oled. Ver (**Tabla 12**).

Tabla 12: Características Oled 128x64 Arduino

PCB	38 mm x 29 mm (1,5 "x 1")
Pantalla	25mm x 14mm
Espesor	4 mm
Peso	8,5 g

Fuente.- Jenny Patín Chimbo

El consumo de corriente es completamente dependiente de su uso: cada OLED consume corriente cuando en más píxeles haya iluminado, se utiliza la más actual. Ellos tienden a atraer ~ 20 mA o menos en la práctica, pero para el número exacto debe medir la corriente en el circuito de uso. Esta placa / chip utiliza I2C dirección de 7 bits entre 0x3C-0x3D, seleccionable mediante puentes.

Se usó para la visualización de datos en el invernadero y otra en las placas. Se encuentra instalado en la transmisora del prototipo para la muestra de los datos del sistema.

2.6.3. Test de comunicación entre el Arduino y los sensores

2.6.3.1. Requisitos de software para las pruebas

Para la comunicación entre la tarjeta Arduino y los dispositivos a usarse, se necesitó la instalación del entorno y las librerías que deben estar en la computadora para las pruebas de comunicación y el desarrollo del programa a realizarse.

Los programas y archivos necesarios son:

- Entorno Arduino 1.6.3
- Librería Dht.h
- Librería Streaming.h
- Librería MemoryFree.h
- Librería Agentuino.h
- Librería Flash.h
- Librería Idoc.h
- Para el monitoreo se utiliza un software libre Linux y Nagios.
- Comunicación Serial sensor del suelo
- Comunicación Serial de los Módulos XBee para la comunicación inalámbrica.
- Para los actuadores se usa un pulso es 0 o 1.

Se debe ingresar correctamente en el directorio del entorno Arduino las librerías necesarias, para que al armar el código, el programa pueda ser compilado y no presente ningún error de librerías no declaradas.

2.6.3.2. Conexión de Arduino con el Sensor DHT22

1. Se cargó la librería correspondiente al sensor DHT22 en el directorio del entorno de Arduino.
2. Se inició el entorno Arduino y se verificó que reconozca la librería, mediante el menú *Sketch>Importar librería*.

3. Se implementó el código de uso del sensor, para obtener una prueba de comunicación. El código tuvo la siguiente estructura.
4. Se compiló el código para probar que no haya errores en el programa.
5. Se realizó la conexión de los sensores DHT22 y la tarjeta Arduino.

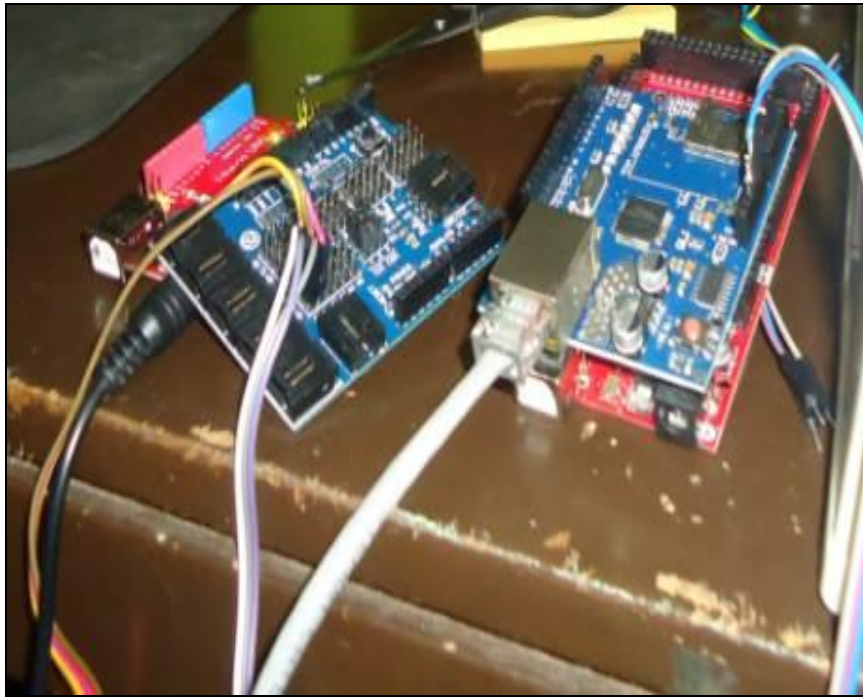
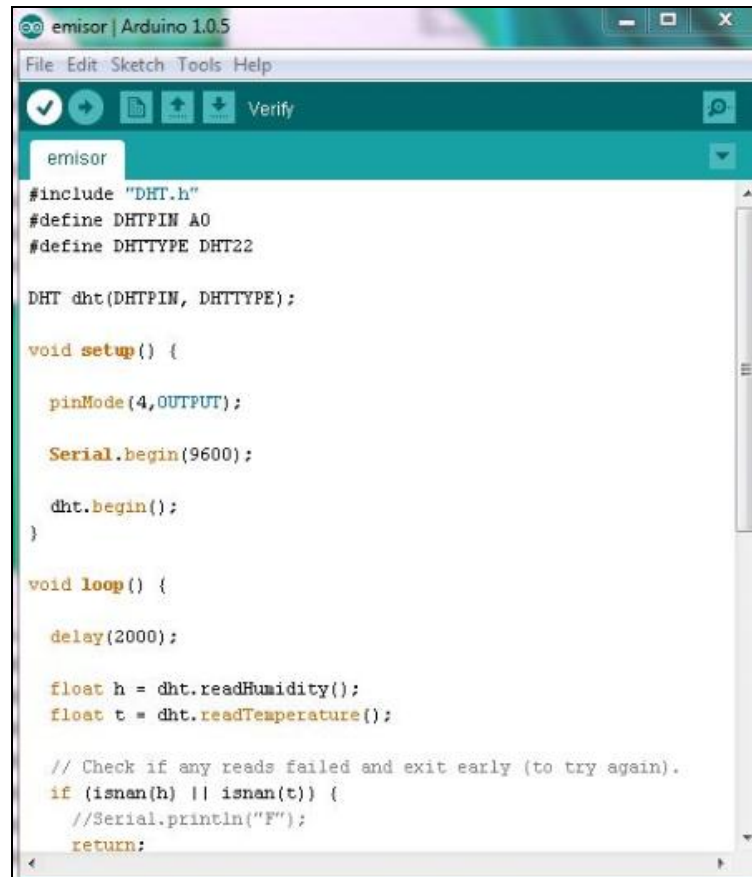


Ilustración 38: Conexión entre la tarjeta Arduino y los sensores

Fuente.- Jenny Patín Chimbo

6. Se conectó la tarjeta Arduino al PC mediante un cable USB y verificar que el entorno reconozca al puerto COM, reseteó el Arduino y cargó nuevamente el programa.
7. Se confirmó la comunicación entre el Arduino y el sensor DHT22. El sensor se encarga de la toma de los datos de la temperatura y la humedad relativa. El programa tuvo la siguiente estructura:



```
emisor | Arduino 1.0.5
File Edit Sketch Tools Help
Verify
emisor
#include "DHT.h"
#define DHTPIN A0
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup() {
  pinMode(4, OUTPUT);
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  delay(2000);

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    //Serial.println("F");
    return;
  }
}
```

Ilustración 39: Programación del Sensor DHT22

Fuente.- Jenny Patín Chimbo

2.6.3.3. Conexión de Arduino con el sensor Soil.

Este sensor tiene la capacidad de medir la humedad del suelo. Aplicando una pequeña tensión entre los terminales del módulo, hace pasar una corriente que depende básicamente de la resistencia que se genera en el suelo y ésta depende mucho de la humedad. Por lo tanto al aumentar la humedad la corriente crece y al bajar la corriente disminuye. Posee dos tipos de salidas, una analógica y una digital. La salida digital entregará un pulso bajo cuando haya conductividad suficiente entre cada una de las puntas. El umbral de disparo se puede establecer moviendo el potenciómetro del circuito de control. En la salida analógica el nivel de voltaje dependerá directamente de cuanta humedad haya en el suelo. Es decir,

dependiendo de cuanta conductividad (producto del agua en el suelo) haya entre las puntas del módulo, así variará el valor entregado por Arduino (entre 0 y 1023).

Se conectó el sensor de la siguiente forma:

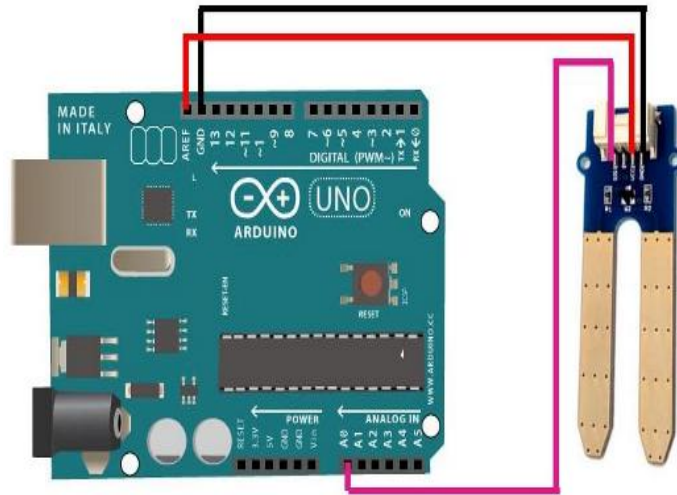


Ilustración 40: Conexión entre la tarjeta Arduino y sensor Soil

Fuente.- Jenny Patín Chimbo

Se confirmó la comunicación entre el Arduino y el sensor Soil. El programa tuvo la siguiente estructura:

```
sketch_mar04a | Arduino 1.0.5
File Edit Sketch Tools Help
sketch_mar04a $
int sensorPin = A4;
int sensorValue = 0;
String humedad;
int A;
char c;
void setup() {
  Serial.begin(9600);
}
void loop() {
  sensorValue = analogRead(sensorPin);
  delay(1000);
  Serial.print("sensor = ");
  Serial.println(sensorValue);
  if(sensorValue >= 266) {
    digitalWrite(4, HIGH);
  } else {
    digitalWrite(4, LOW);
  }
}
```

Ilustración 41: Programación del Sensor Soil

Fuente.- Jenny Patín Chimbo

2.6.3.4. Conexión de los Módulos XBee (Comunicación inalámbrica).

De la misma manera la comunicación del dispositivo de humedad y control lo realizó mediante módulos XBee, el cual está conectado al Arduino por medio del shield de integración Xbee.



Ilustración 42: Shield Xbee V03

Fuente: Jenny Patín Chimbo

Para la comunicación inalámbrica utilizo un par de Xbee Pro, una para la base del monitoreo SNMP y otro para el control y toma de datos de los sensores DHT22 y Soil.



Ilustración 43: Modulo Xbee PRO con antena integrada.

Fuente: Jenny Patín Chimbo

La facilidad de usar Shield XBee es simplemente conectar el módulo XBee sobre el Shield y el Shield sobre la placa Arduino y de esta manera todo queda integrado en un solo paso sin necesidad de cables ni conexiones extras, como se puede apreciar en la siguiente (**Ilustración 44**).

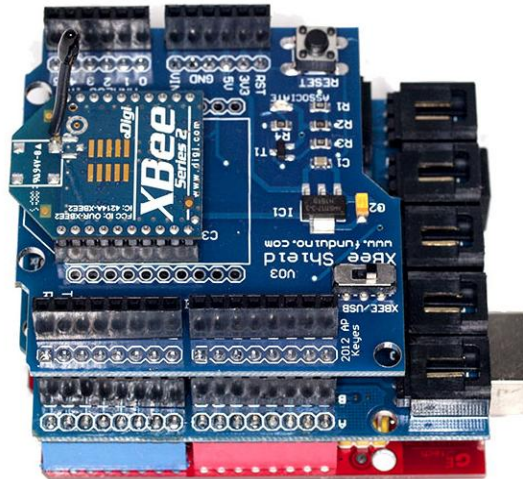


Ilustración 44: Integración del módulo Xbee+Shield Xbee 03+Arduino.

Fuente: Jenny Patín Chimbo

De igual manera ocurre con el diseño del circuito de comunicación de temperatura y humedad ya que no hay necesidad de realizar conexiones extras pues el Shield tiene esa característica de no usar cables adicionales.

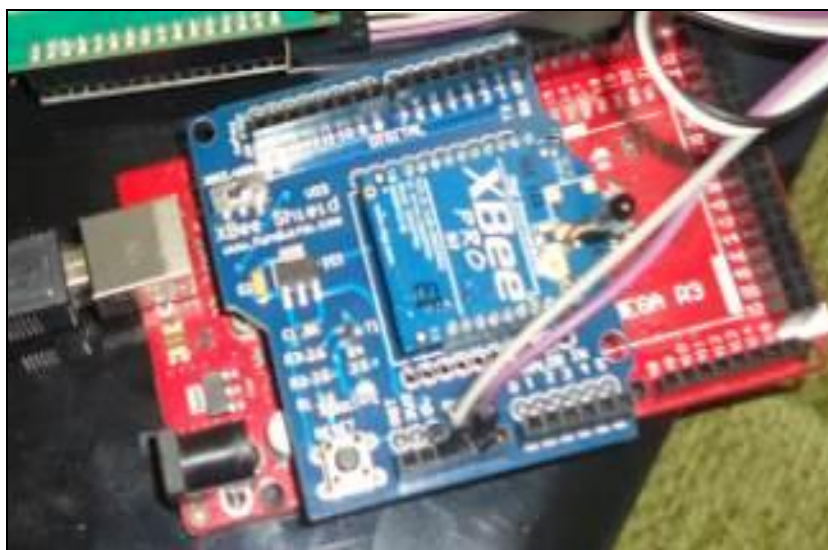
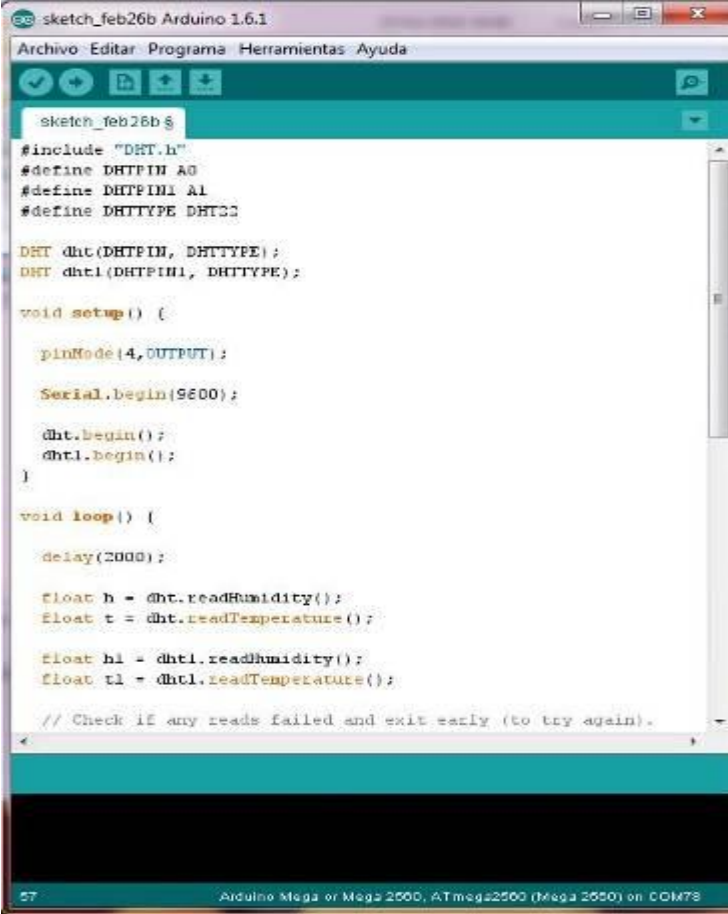


Ilustración 45: Xbee+Shield Xbee 03+Arduino.

Fuente: Jenny Patín Chimbo

Para la configuración de los módulos XBee se usó el programa XCTU, para cargar el Firmware y dar la dirección MAC a los dispositivos de Tx y Rx.

Se abre el entorno de programación de Arduino y se procede a la importación de librerías *DHT* y *SERIAL* para el procesamiento de datos provenientes del sensor DHT22, adicional se configuro la tasa de transmision para la comunicación serial 9600, 8, None,1 (velocidad de tx-bits-paridad-bit de parada). En la (**Ilustración 46**) se muestra el código usado para la transmisión de las variables de temperatura y humedad provenientes de los sensores. Los modulos XBee Pro tienen un alcance de 1600 metros de distancia con una frecuencia de transmisión de 2.4 GHz.



```
sketch_feb26b Arduino 1.6.1
Archivo Editar Programa Herramientas Ayuda
sketch_feb26b
#include "DHT.h"
#define DHTPIN A0
#define DHTPIN1 A1
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
DHT dht1(DHTPIN1, DHTTYPE);

void setup() {
  pinMode(4, OUTPUT);
  Serial.begin(9600);
  dht.begin();
  dht1.begin();
}

void loop() {
  delay(2000);

  float h = dht.readHumidity();
  float t = dht.readTemperature();

  float h1 = dht1.readHumidity();
  float t1 = dht1.readTemperature();

  // Check if any reads failed and exit early (to try again).
```

Ilustración 46: Código usado para la transmisión de módulo XBee

Fuente: Jenny Patín Chimbo

Se abre el entorno de programación de Arduino y se procede a la importación de librerías *Wire.h*, *LiquidCrystal_I2C.h* para la recepción de datos provenientes del sensor DHT22.

En la (Ilustración 47) se muestra el código usado para la recepción de datos temperatura y humedad provenientes de los sensores.



```
sketch_feb26a $
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27

float vt1, vh1, vt2, vh2;
String temperatur1, temperatura2;
String humedad1, humedad2;
int ft1=0, fh1=0, ft2=0, fh2=0;
char c;

void actualizar(){

    temperatur1.replace('B',' ');
    temperatur1.replace('C',' ');
    Serial.println("temp= "+temperatur1);

    humedad1.replace('A',' ');
    humedad1.replace('E',' ');
    Serial.println("hum= "+humedad1);
}

Global variables use 539 bytes (6%) of dynamic memory, leaving
7.653 bytes for local variables. Maximum is 8.192 bytes.

159 Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM78
```

Ilustración 47: Código usado para la recepción de módulo XBee

Fuente: Jenny Patín Chimbo

2.6.3.5. Conexión de Arduino con el router

Utilice el Shield Ethernet con IP 192.168.1.6 con máscara 255.255.255.0 y un router con IP 192.168.20.1 con máscara 255.255.255.0, con Gateway 192.168.1.1 con máscara 255.255.255.0 para enrutar la red. Ver en la (Ilustración 48).

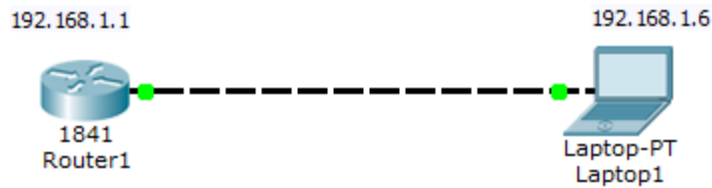


Ilustración 48: Router y Shield Ethernet (PC).

Fuente: Jenny Patín Chimbo

2.6.3.6. Conexión de Administración SNMP

La integración del Arduino con el protocolo SNMP se lo realiza mediante la utilización de una librería diseñada específicamente para el uso de dicho protocolo, dicha librería es de código abierto y puede ser obtenido de la página web del autor <https://code.google.com/p/agentuino/>, la librería en cuestión es la AGENTUINO. Ver la siguiente (Ilustración 49)

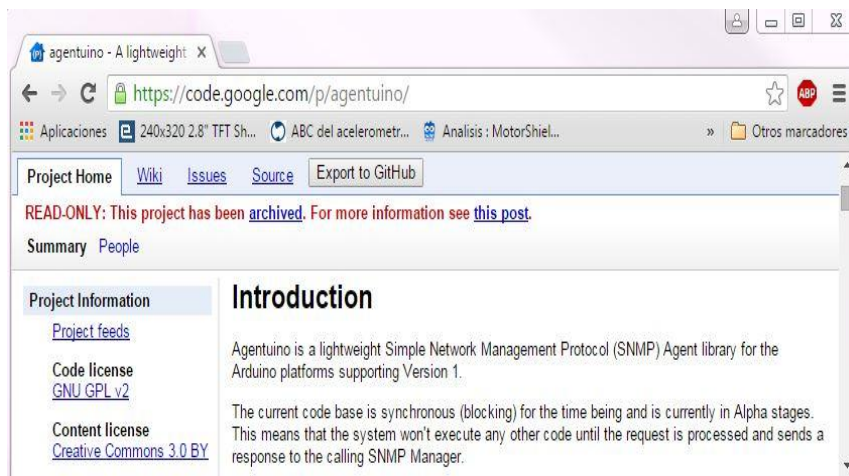


Ilustración 49: Agentuino sitio web del autor.

Fuente: Jenny Patín Chimbo

El uso de los comandos para el envío y recolección de datos sigue los protocolos SNMP establecidos, como se muestran en la siguientes **(Ilustración 50)**.

Net-Snmp Get-Request

Open a command prompt or console and type the following (edit the host address, e.g. Agentuino IP Address, as needed);

```
snmpget -v 1 -r 1 -c public 192.168.2.64 sysUpTime.0
```

The command should respond with the following output;

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (1471800) 4:05:18.00
```

Ilustración 50: Uso del SNMPGET de Agentuino.

Fuente: Jenny Patín Chimbo

De la misma manera ocurre con el ingreso de datos en formato SNMP como se puede apreciar en la siguiente **(Ilustración 51)**.

Net-Snmp Set-Request

Open a command prompt or console and type the following (edit the host address, e.g. Agentuino IP Address, as needed);

```
snmpset -v 1 -r 1 -c public 192.168.2.64 sysName.0 s NewName
```

The command should respond with the following output;

```
SNMPv2-MIB::sysName.0 = STRING: NewName
```

Let's test the updated sysName with a Get-Request. Open a command prompt or console and type the following (edit the host address, e.g. Agentuino IP Address, as needed);

```
snmpget -v 1 -r 1 -c public 192.168.2.64 sysName.0
```

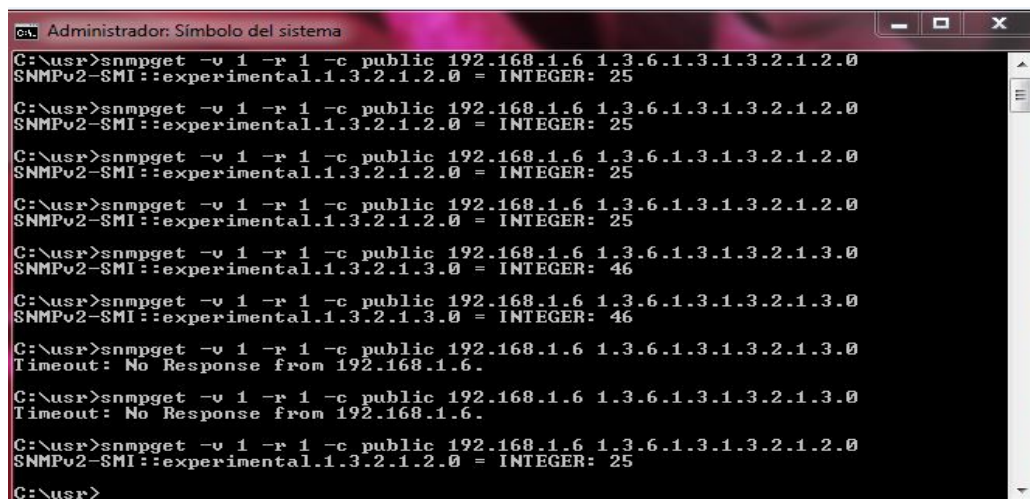
The command should respond with the following output;

```
SNMPv2-MIB::sysName.0 = STRING: NewName
```

Ilustración 51: Uso del SNMPSET de Agentuino.

Fuente: Jenny Patín Chimbo

Los sensores se encuentran en una misma red, lo que cambia es su OID, de esa manera se pueden administrar con SNMP. Ver en la **(Ilustración 52)**.



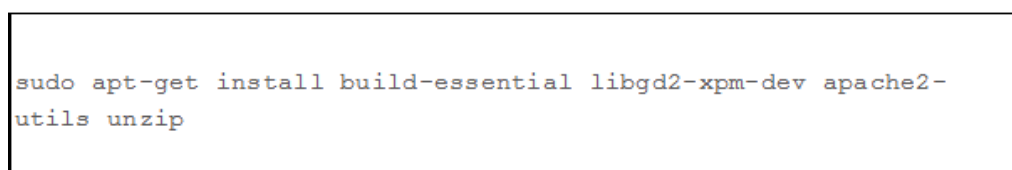
```
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
SNMPv2-SMI::experimental.1.3.2.1.3.0 = INTEGER: 46
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
SNMPv2-SMI::experimental.1.3.2.1.3.0 = INTEGER: 46
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
Timeout: No Response from 192.168.1.6.
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
Timeout: No Response from 192.168.1.6.
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25
C:\usr>
```

Ilustración 52: Pruebas usando SNMPSET manualmente

Fuente: Jenny Patín Chimbo

2.6.3.7. Conexión de Nagios en Ubuntu

El monitoreo de los parámetros de temperatura y humedad lo realizó mediante NAGIOS sobre Linux específicamente sobre Ubuntu 15.10, para lo cual debó, instalar Ubuntu y luego proceder a la instalación del NAGIOS, dicha instalación se la realiza mediante consola como es lo común en Linux para lo que debó seguir los siguientes pasos y digitar para lo cual abro una consola o terminal presionando Crtl+Alt+T y acto seguido escribo ver (**Ilustración 53**).



```
sudo apt-get install build-essential libgd2-xpm-dev apache2-
utils unzip
```

Ilustración 53: Instalación de Nagios por consola

Fuente: Jenny Patín Chimbo

Se crea una nueva cuenta de usuario para Nagios digitando. Ver (**Ilustración 54**).

```
sudo useradd -m nagios
```

Ilustración 54: Usuario para Nagios

Fuente: Jenny Patín Chimbo

Y de la misma manera se crea un password para dicho usuario se digita. Ver **(Ilustración 55)**.

```
sudo password nagios
```

Ilustración 55: Password para Nagios

Fuente: Jenny Patín Chimbo

Se crea un nuevo grupo nagcmd para permitir comandos externos que se presentará a través de la interfaz web. Añadir tanto el usuario nagios y el usuario apache al grupo, se digita. . Ver **(Ilustración 56)**.

```
sudo groupadd nagcmd  
  
sudo usermod -a -G nagcmd nagios  
  
sudo usermod -a -G nagcmd www-data
```

Ilustración 56: Grupo Nagcmd

Fuente: Jenny Patín Chimbo

Se va a la página de Nagios a descargar y obtener la última versión. Al realizar esta investigación, la versión más reciente era 4.1.1, para lo cual se digita. Ver **(Ilustración 57)**.

```
wget
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
```

Ilustración 57: Página de Nagios

Fuente: Jenny Patín Chimbo

De igual manera se descargó los plugins de Nagios que permiten monitorear el hosts, dispositivos, protocolos y aplicaciones digitando los siguientes comandos. . Ver **(Ilustración 58)**.

```
wget http://www.nagios-plugins.org/download/nagios-plugins-2.1.1.tar.gz
```

Ilustración 58: Comandos de Nagios

Fuente: Jenny Patín Chimbo

Con Nagios y los plugins descargados procede a descomprimir el archivo digito. Ver **(Ilustración 59)**.

```
tar xzf nagios-4.1.1.tar.gz
```

Ilustración 59: Plugins descargados

Fuente: Jenny Patín Chimbo

Se cambió al directorio de Nagios y se digita. Ver **(Ilustración 60)**.

```
cd nagios-4.1.1/
```

Ilustración 60: Directorio de Nagios

Fuente: Jenny Patín Chimbo

Se ejecutó los siguientes comandos uno por uno en la terminal de Ubuntu para compilar e instalar Nagios. Ver **(Ilustración 61)**.

```
sudo ./configure --with-command-group=nagcmd  
  
sudo make all  
  
sudo make install  
  
sudo make install-init  
  
sudo make install-config  
  
sudo make install-commandmode
```

Ilustración 61: Comandos terminal Ubuntu

Fuente: Jenny Patín Chimbo

Seguidamente se instaló la interface web de Nagios digitando los siguientes comandos. Ver **(Ilustración 62)**.

```
sudo make install-webconf
```

Ilustración 62: Interface web de Nagios

Fuente: Jenny Patín Chimbo

Se verifica `nagios.conf` está colocado en el siguiente directorio `/etc/apache2/sites-enabled/`. Ver (**Ilustración 63**).

```
sudo ls -l /etc/apache2/sites-enabled/
```

Ilustración 63: Comando `nagios.conf`

Fuente: Jenny Patín Chimbo

Y como respuesta se obtiene. Ver (**Ilustración 64**).

```
total 4

lrwxrwxrwx 1 root root 35 Nov 28 16:49 000-default.conf ->
../sites-available/000-default.conf

-rw-r--r-- 1 root root 1679 Nov 28 17:02 nagios.conf
```

Ilustración 64: Respuesta de Nagios

Fuente: Jenny Patín Chimbo

Se crea la cuenta *nagios admin* para poder comunicar en la interface web, se debe asignar un password para necesariamente poder ingresar al interfaz web digitando. Ver (**Ilustración 65**).

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

Ilustración 65: Cuenta Nagios admin

Fuente: Jenny Patín Chimbo

Se procede a reiniciar el servicio de Apache para que todos los cambios tomen efecto para lo que se digita el siguiente comando por medio del cual se observa los cambios que han sucedido. Ver **(Ilustración 66)**.

```
sudo systemctl restart apache2
```

Ilustración 66: Servicio Apache

Fuente: Jenny Patín Chimbo

De igual manera se instala los plugins de Nagios se digita los siguiente comandos los mismos que ayudara a Nagios a proceder con un correcto funcionamiento.. Ver **(Ilustración 67)**.

```
tar xzf nagios-plugins-2.1.1.tar.gz

cd nagios-plugins-2.1.1/

sudo ./configure --with-nagios-user=nagios --with-nagios-
group=nagios

sudo make

sudo make install
```

Ilustración 67: Plugins de Nagios con servicio Apache

Fuente: Jenny Patín Chimbo

Para configurar Nagios se debe editar el archivo contacts.org que es el que contiene la información personal y que se encuentra ubicado en el siguiente directorio /usr/local/nagios/etc/objects/contacts.cfg. Ver **(Ilustración 68)**.


```
sudo nano /usr/local/nagios/etc/objects/contacts.cfg
```

Ilustración 68: Edición del archivo contacts.org

Fuente: Jenny Patín Chimbo

Se ingresa mi e-mail para la configuración de los datos personales como se indica a continuación. Ver **(Ilustración 69)**.

```
[...]  
  
define contact{  
  
    contact_name          nagiosadmin          ;  
    Short name of user  
  
    use                   generic-contact      ;  
    Inherit default values from generic-contact template (defined  
    above)  
  
    alias                 Nagios Admin        ; Full  
    name of user  
  
    email                 jenrocio_4787@hotmail.com;  
  
}
```

Ilustración 69: Configuración de datos personales

Fuente: Jenny Patín Chimbo

Se guarda y se cierra el archivo para ahora editar el nagios.cfg para poder modificar los valores de las IP. Ver **(Ilustración 70)**.

Aquí, se da a Nagios permiso de acceso administrativo a solamente a la serie de IP 192.168.20.0/1. Por lo tanto, he editado este archivo como se muestra a continuación.

```
[...]  
  
## Comment the following lines ##  
  
# Order allow,deny  
  
# Allow from all  
  
## Uncomment and Change lines as shown below ##  
  
Order deny,allow  
  
Deny from all  
  
Allow from 127.0.0.1 192.168.20.0/10  
  
[...]
```

Ilustración 70: Permiso de acceso administrativo

Fuente: Jenny Patín Chimbo

Con esto tan solo queda reiniciar y rescribir los servicios de Apache se digita. Ver (**Ilustración 71**).

```
sudo a2enmod rewrite  
  
sudo a2enmod cgi  
  
sudo systemctl restart apache2  
  
sudo service nagios start
```

Ilustración 71: Reiniciar Apache

Fuente: Jenny Patín Chimbo

Con todo esto por fin habré logrado la instalación de Nagios en Ubuntu como se puede apreciar. Ver (**Ilustración 72**).



Ilustración 72: Instalación de Nagios 4.1.1 en Ubuntu 15.10

Fuente: Jenny Patín Chimbo

2.6.3.8. Configuración de Nagios para el monitoreo de los sensores de humedad y temperatura.

Una vez instalado Nagios en Ubuntu, procedo a la creación de un Host y la configuración de los comandos y servicios para el monitoreo de los sensores de humedad y temperatura mediante OID y utilizando el comando `snmp_check` de Nagios para el monitoreo de los datos de los sensores.

Se instaló el servicio NRPE y se actualiza los siguientes comandos. Ver **(Ilustración 73)**.

```
sudo apt-get update

sudo apt-get install nagios-nrpe-server nagios-plugins
```

Ilustración 73: Servicio NRPE

Fuente: Jenny Patín Chimbo

Se editó el archivo nrpe.cfg para el cual se digita. Ver (**Ilustración 74**).

```
sudo nano /etc/nagios/nrpe.cfg
```

Ilustración 74: Archivo nrpe.cfg

Fuente: Jenny Patín Chimbo

Ahora se debe añadir la IP al servidor a monitorear en mi caso 192.168.20.6. Ver (**Ilustración 75**).

```
[...]

## Find the following line and add the Nagios server IP ##

allowed_hosts=127.0.0.1 192.168.20.6

[...]
```

Ilustración 75: Servidor a monitorear

Fuente: Jenny Patín Chimbo

Se inicializó los servicios de nrpe se digita. Ver **(Ilustración 76)**.

```
sudo /etc/init.d/nagios-nrpe-server restart
```

Ilustración 76: Servicio de NRPE

Fuente: Jenny Patín Chimbo

Ahora se debe editar en nagios.cfg para añadir un nuevo servidor para lo cual se digita. Ver **(Ilustración 77)**.

```
sudo nano /usr/local/nagios/etc/nagios.cfg
```

Ilustración 77: Configuración de nagios.cfg

Fuente: Jenny Patín Chimbo

Ahora se debe buscar las siguientes líneas de código y desconectar. Ver **(Ilustración 78)**.

```
## Find and uncomment the following line ##  
  
cfg_dir=/usr/local/nagios/etc/servers
```

Ilustración 78: Códigos de líneas

Fuente: Jenny Patín Chimbo

Se crea un directorio llamado “servers” bajo la siguiente dirección “/usr/local/nagios/etc/”. Ver **(Ilustración 79)**.

```
sudo mkdir /usr/local/nagios/etc/servers
```

Ilustración 79: Directorio servers

Fuente: Jenny Patín Chimbo

Se crea el archivo de configuración para monitorear el objetivo se digita. Ver (**Ilustración 80**).

```
sudo nano /usr/local/nagios/etc/servers/clients.cfg
```

Ilustración 80: Configuración para monitoreo

Fuente: Jenny Patín Chimbo

Se agregó las siguientes líneas manteniendo esta estructura. Ver (**Ilustración 81**).

```
define host{  
    use                linux-server  
    host_name          JENNY_TESIS  
    alias              server  
    address            192.168.20.6  
    max_check_attempts 5  
    check_period       24x7  
    notification_interval 30  
    notification_period 24x7  
}
```

Ilustración 81: Líneas de estructura

Fuente: Jenny Patín Chimbo

Queda solamente reiniciar el servicio de Nagios y se crea un nuevo host a ser monitoreado para lo cual se digita. Ver (**Ilustración 82**).

```
sudo /etc/init.d/nagios restart
```

Ilustración 82: Creación de nuevo host

Fuente: Jenny Patín Chimbo

Se abre nuevamente el navegador e ingresó a “/localhost/nagios/” digitó mi usuario y contraseña y doy clic para ingresar a Nagios, en la pestaña del lado izquierdo verifico la opción HOST y doy clic se desplegará una nueva ventana donde muestra la información del nuevo host creado para monitorear tal y como se muestra en el siguiente (**Ilustración 83**).

The screenshot shows the Nagios Core web interface in a Mozilla Firefox browser. The page displays the current network status, host status totals, and service status totals. The host status details section shows two hosts: JENNY_TESIS and localhost, both with a status of UP. The last check for JENNY_TESIS was on 01-13-2016 at 10:23:02, and for localhost it was on 01-13-2016 at 10:20:08. The status information for both hosts is PING OK - Packet loss = 0%, RTA = 0.77 ms and PING OK - Packet loss = 0%, RTA = 0.08 ms respectively.

Host	Status	Last Check	Duration	Status Information
JENNY_TESIS	UP	01-13-2016 10:23:02	0d 0h 1m 32s	PING OK - Packet loss = 0%, RTA = 0.77 ms
localhost	UP	01-13-2016 10:20:08	8d 22h 45m 56s	PING OK - Packet loss = 0%, RTA = 0.08 ms

Ilustración 83: Creación de un nuevo host en Nagios 4.1.1.

Fuente: Jenny Patín Chimbo

En la (**Ilustración 84**) se muestra el monitoreo de la humedad mediante

Nagios.

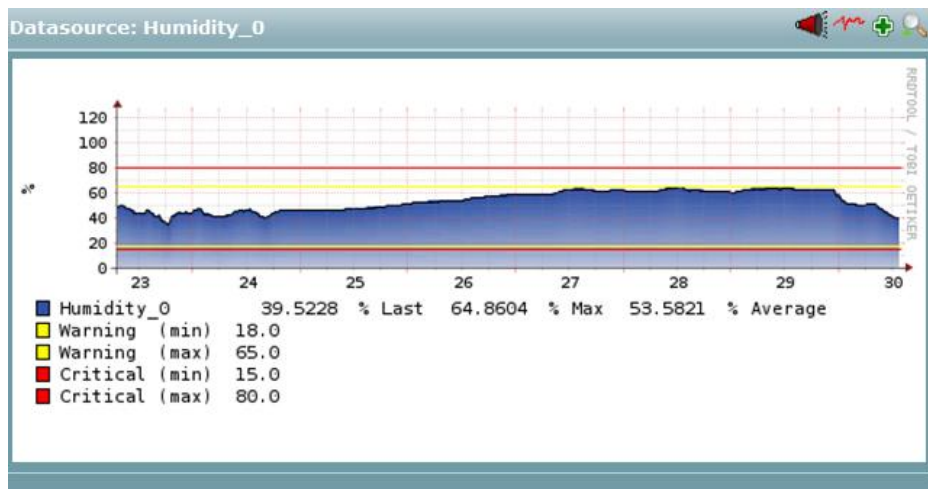


Ilustración 84: Monitoreo de la Humedad

Fuente.- Jenny Patín Chimbo

En la **(Ilustración 85)** se muestra el monitoreo de la Temperatura mediante Nagios.

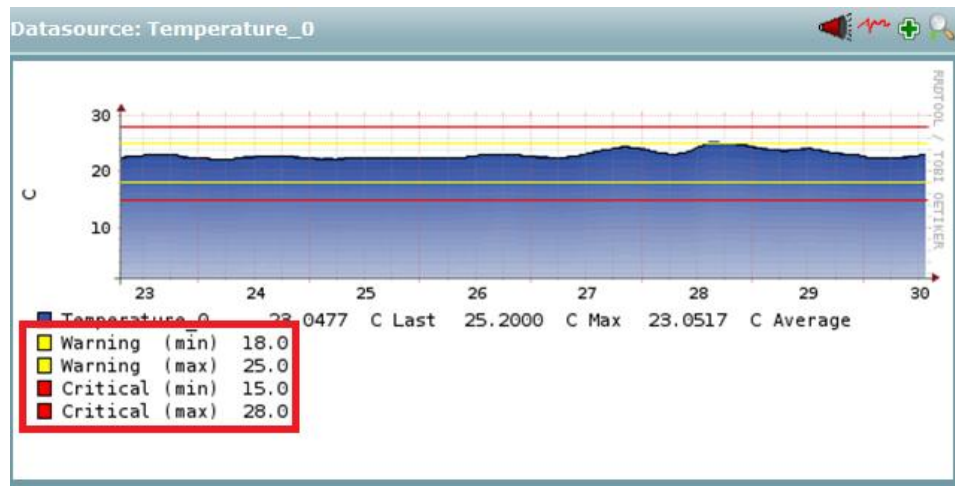


Ilustración 85: Monitoreo de la Temperatura

Fuente.- Jenny Patín Chimbo

En la **(Ilustración 86)** se muestra el sniffer que se usó para verificar que el prototipo esta administrado mediante SNMP.

No.	Time	Source	Destination	Protocol	Length	Info
288	628.056220	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
289	629.026453	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
290	631.077290	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
291	631.077345	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
292	631.537695	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
293	632.540538	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
294	633.127675	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
295	634.452434	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
296	635.178643	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
297	635.178700	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
298	635.462373	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
299	637.229009	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
300	637.229065	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
301	638.026419	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
302	639.040455	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
303	639.279713	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
304	639.279756	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
305	641.330093	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
306	641.330152	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
308	643.380688	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
309	643.380743	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
315	648.221349	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
316	649.227835	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
318	649.527677	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.2.0

Frame 11: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
 Ethernet II, Src: IntelCor_ac:ef:6f (00:19:d1:ac:ef:6f), Dst: de:ad:be:ef:fe:ed (de:ad:be:ef:fe:ed)
 Internet Protocol Version 4, Src: 192.168.1.2, Dst: 192.168.1.6

Ilustración 86: Pruebas Wireshark

Fuente.- Jenny Patín Chimbo

2.6.4. Diseño de etapas.

El dispositivo se diseñó para equiparse en dos etapas: la primera consta de la tarjeta Arduino MEGA, placa Shield, el módulo XBee, todos estos elementos se integraron en una caja de plástico fabricada a la medida. Ver (**Ilustración 87**).



Ilustración 87: Tarjeta Arduino MEGA con la placa shield y el módulo XBee

Fuente.- Jenny Patín Chimbo

Fuera de esta caja se colocó un router inalámbrico que se encarga de que todos los equipos se encuentren en la misma red.

La segunda etapa consta de los sensores de humedad relativa, temperatura y humedad del suelo, que se ubicaron dentro del invernadero en la parte superior del invernadero, uno en el suelo, uno a lado izquierdo del invernadero. Dentro del invernadero se encuentran ubicados dos micro ventiladores, una mini bomba y un calefactor que funciona de acuerdo a los cambios climáticos que sucedan dentro del mismo, también se usa placa Shield, Modulo XBee que estarán dentro de una caja. Para conectarlas a la tarjeta Arduino usó cable UTP. Ver (**Ilustración 88**).



Ilustración 88: Ubicación de los sensores dentro del invernadero con la tarjeta Shiel y módulo Xbee

Fuente.- Jenny Patín Chimbo

2.6.5. Programación en el entorno de Arduino

Luego de realizar las pruebas de comunicación entre el Arduino, Módulos XBee y los sensores se procedió a escribir el código para controlar el dispositivo. Se incluyen primero las librerías de los sensores. Ver (**Ilustración 89**).

```

sketch_feb26a $
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27

float vt1, vh1, vt2, vh2;
String temperatur1, temperatura2;
String humedad1, humedad2;
int ft1=0,fh1=0,ft2=0,fh2=0;
char c;

```

Ilustración 89: Librerías en el entorno de Arduino

Fuente.- Jenny Patín Chimbo

El cual permite que el programa se cargue a la memoria de la tarjeta Arduino los comandos determinados, tanto para los sensores, los módulos XBee y los actuadores. Se declaró las variables de entrada y salida de datos de los sensores usando los pines bidireccionales. Ver (**Ilustración 90**).

```

Tesis_1no
//librerías para publicación SNI
#include <String.h> //
#include <Ethernet.h> //librería que controla el manejo del Shield
#include <SPI.h> //librería que maneja el protocolo SPI
#include <MemoryFree.h>
#include <Agentuino.h>
#include <Pins.h>
#include <"DHT.h"> // Librería del dht

//definiciones de analog read
int sensorValue = 0;
int sensorValue2 = 0;
int sensorValue3 = 0;
int sensorValue4 = 0;
const int DHTpin1 = A2; // DHT22
const int DHTpin2 = A3; // DHT22

//..... Configurando para la correcta comunicación en la red.
static byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
static byte ip[] = {192,168,0,81};
static byte gateway[] = {192,168,0,1};
static byte subnet[] = { 255, 255, 255, 0 };

//-----Se crea una cadena de caracteres que obtienen los OIP
static char sysDescr[] PROGMEM = "1.3.6.1.2.1.1.1.0"; // read-only
// .iso.org.cod.internet.mgmt.mib-2.system.sysObjectID {1.3.6.1.2
static char sysUpTime[] PROGMEM = "1.3.6.1.2.1.1.3.0"; // read-only
// .iso.org.cod.internet.mgmt.mib-2.system.sysContact {1.3.6.1.2
static char sysContact[] PROGMEM = "1.3.6.1.2.1.1.4.0"; // read-wr
// .iso.org.cod.internet.mgmt.mib-2.system.sysName {1.3.6.1.2.1.1
static char sysName[] PROGMEM = "1.3.6.1.2.1.1.5.0"; // read-write
// .iso.org.cod.internet.mgmt.mib-2.system.sysLocation {1.3.6.1.2
static char sysLocation[] PROGMEM = "1.3.6.1.2.1.1.6.0"; // read-r
// .iso.org.cod.internet.mgmt.mib-2.system.sysServices {1.3.6.1.2
static char sysServices[] PROGMEM = "1.3.6.1.2.1.1.7.0"; // read-r

//-----Creando los OID para la variable a medir en este caso
// SysDht22 para de Temperatura
static char SysDht22[] PROGMEM = "1.3.6.1.3.1.3.2.1.3.0"; // read-
static char SysDht22E[] PROGMEM = "1.3.6.1.3.1.3.2.1.3.0"; // read
static char SysDht22I[] PROGMEM = "1.3.6.1.3.1.3.2.1.4.0"; // read
static char SysDht22S[] PROGMEM = "1.3.6.1.3.1.3.2.1.5.0"; // read

// RFC1213 local values
static char locDescr[] = "Agentuino, un agente SNMP"; // read-only
static uint32_t locUpTime = 0; // read-only (static)
static char locContact[20] = "JENNY JENNY"; // should be stored/ce
static char locName[20] = "Agentuino"; // should be stored/read 2x

```

Ilustración 90: Programación del proyecto

Fuente.- Jenny Patín Chimbo

2.6.6. Ensamblaje de las etapas

Una vez que la programación fue cargada al Arduino y además probada con todos sus componentes individuales, se procedió a armar las dos etapas:

La primera consta de la tarjeta Arduino MEGA, placa Shield, el módulo XBee, todos estos elementos se integraron en una caja de plástico fabricada a la medida.

Para la segunda etapa se necesitó la caja plástica, en la que se colocó la tarjeta Arduino, el shield con el módulo Xbee. Además se incluyó un interruptor para encender y apagar el sistema de manera sencilla.

Se hicieron orificios en la caja de 15,7m de largo y 4cm de ancho, para las siguientes conexiones: entrada del cable de red, conector de alimentación de 5V cables para la conexión con los sensores ubicados en la maqueta. Ver **(Ilustración 91)**.

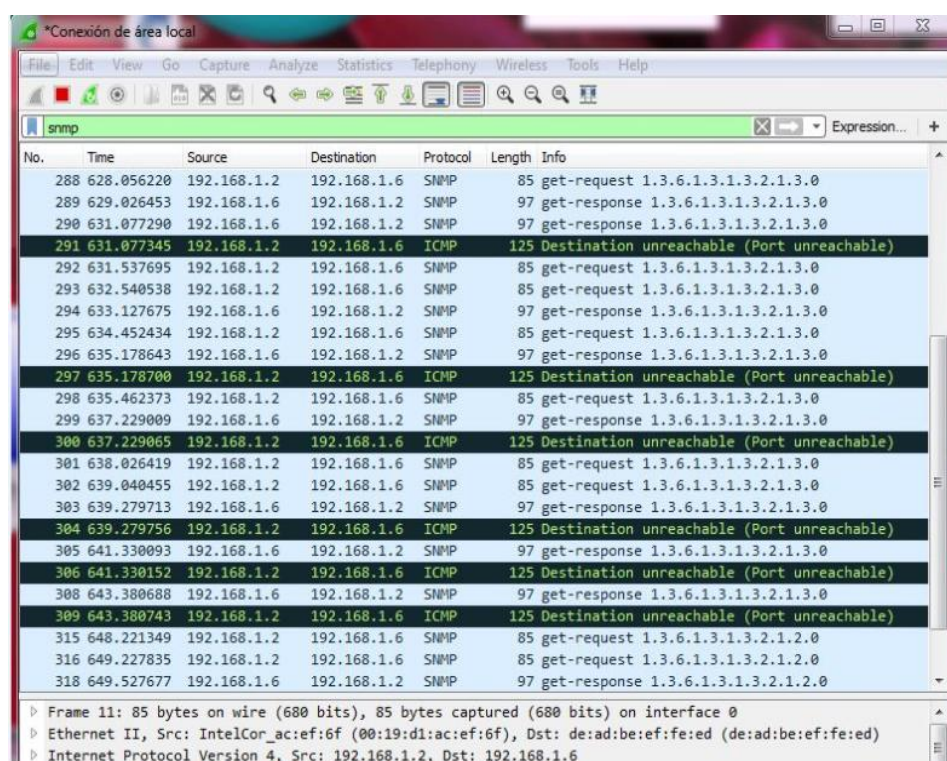


Ilustración 91: Ensamblaje de las etapas´

Fuente.- Jenny Patín Chimbo

2.6.7. Recolección de datos mediante un Sniffer

Para la recolección de datos se usó un sniffer en donde se verifica la correcta administración de los sensores mediante SNMP. Estos datos pueden ser usados como control para comprobar el correcto funcionamiento de los sensores, además de establecer otros rangos para la medición y la emisión de las alertas.



The screenshot shows a Wireshark interface with a capture filter set to 'snmp'. The packet list pane displays a series of network packets. The table below represents the data visible in the packet list pane.

No.	Time	Source	Destination	Protocol	Length	Info
288	628.056220	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
289	629.026453	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
290	631.077290	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
291	631.077345	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
292	631.537695	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
293	632.540538	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
294	633.127675	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
295	634.452434	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
296	635.178643	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
297	635.178700	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
298	635.462373	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
299	637.229009	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
300	637.229065	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
301	638.026419	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
302	639.040455	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
303	639.279713	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
304	639.279756	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
305	641.330093	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
306	641.330152	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
308	643.380688	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
309	643.380743	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
315	648.221349	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
316	649.227835	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
318	649.527677	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.2.0

Ilustración 92: Recolección de datos

Fuente.- Jenny Patín Chimbo

2.6.8. COMPROBACIÓN DE LA HIPÓTESIS

El paso de comprobación de la hipótesis en la investigación permite garantizar los hechos obtenidos que coincidan con la hipótesis expuesta. Las hipótesis se verifican con la realidad y pueden ser aprobadas o rechazadas, y para

esto se usa modelos de simulación, diseños experimentales y procedimientos estadísticos.

2.6.8.1. Planteamiento de la hipótesis estadística

H₀: El diseño e implementación de un sistema de monitoreo y control de humedad y temperatura para invernaderos con administración SNMP, permitirá mejorar el control local de los sensores.

H₁: El diseño e implementación de un sistema de monitoreo y control de humedad y temperatura para invernaderos con administración SNMP, no permitirá mejorar el control local de los sensores.

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

$$H_1: \mu_s < \mu_d$$

2.6.8.2. Establecimiento del Nivel de significancia

La prueba se realizó con un 95.5 % de confiabilidad, es decir con un nivel de significancia de $\alpha = 0.005$.

2.6.8.3. Descripción de las poblaciones y suposiciones

Se asume que las poblaciones de las que provienen las muestras están normalmente distribuidas y que sus varianzas no son conocidas. Las muestras son tomadas de acuerdo a la formula anteriormente expuesta para poder obtener buenos resultados. La población se toma en cuenta que es desconocida.

2.6.8.4. Determinación del estadístico pertinente

Si H_0 es verdadera, la media de la distribución muestral es $u_s = u_d \geq 0$ su varianza es $\frac{\sigma_s^2}{n_1} - \frac{\sigma_d^2}{n_2}$. Como σ_s^2 y σ_d^2 son desconocidas, no se puede calcular la varianza verdadera, en consecuencia, excluyo a z como estadístico de prueba.

2.6.8.5. Determinación del estadístico de prueba.

Se define un escenario de pruebas para la comprobación de la hipótesis, que se encuentra configurado por un Access Point a 10 metros y el sistema de monitoreo. Ver (Ilustración 93).

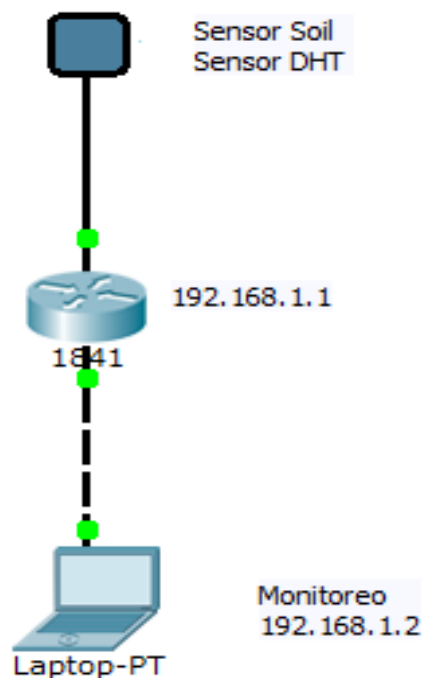


Ilustración 93: Diagrama esquemático del proyecto

Fuente.- Jenny Patín Chimbo

En el escenario A se considera el sistema de sensores formado por el sensor de temperatura, de humedad relativa y humedad del suelo los datos son tomados de manera manual, el tiempo en milisegundos, sin monitoreo como se muestra en

la (Tabla 13), para el cual se utilizó el comando ping, para realizar pruebas de latencia y retardo.

Tabla 13: Tiempos de respuesta sin monitoreo

NÚMERO DE MUESTRAS	TIEMPO EN MILISEGUNDOS		
	Sin Monitoreo		
	Sensor 1	Sensor 2	Sensor 3
1	84	75	135
2	85	136	136
3	86	77	137
4	87	78	138
5	88	79	139
6	89	80	140
7	90	81	141
8	91	142	142
9	92	83	143
10	93	71	131
11	94	85	145
12	95	86	146
13	96	147	147
14	77	88	148
15	78	89	149
16	79	90	150
17	80	91	151
18	81	92	152
19	82	93	153
20	83	94	154
21	84	95	155
22	85	96	156
23	86	157	157
24	87	130	130

25	88	159	159
26	89	129	129
27	90	161	161
28	91	162	162
29	92	131	131
30	93	164	164
31	94	165	165
32	95	129	129
33	96	167	167
34	97	121	121

Fuente.- Jenny Patín Chimbo

En el escenario B se considera el sistema propuesto que se encuentra implementado los datos se toma de forma automática. El tiempo en milisegundos con monitoreo como indica la (Tabla 14).

Tabla 14: Tiempos de respuesta con monitoreo

NÚMERO DE MUESTRAS	TIEMPO EN MILISEGUNDOS		
	Con Monitoreo		
	Sensor 1	Sensor 2	Sensor 3
1	75	65	62
2	76	68	68
3	77	70	68
4	78	78	80
5	79	79	90
6	80	80	87
7	81	75	64
8	82	82	82

9	83	83	67
10	71	90	90
11	85	85	69
12	86	86	86
13	87	87	72
14	88	88	88
15	89	89	70
16	90	90	90
17	91	83	63
18	92	72	72
19	93	65	65
20	94	94	94
21	95	69	69
22	96	76	76
23	97	78	78
24	70	70	62
25	99	99	99
26	69	69	69
27	101	72	72
28	102	70	69
29	71	71	71
30	104	66	65
31	105	69	69
32	69	69	69
33	107	65	65
34	61	61	61

Fuente.- Jenny Patín Chimbo

Las pruebas tomadas para los dos escenarios fueron mediante la fórmula de muestreo de datos mencionado anteriormente.

La siguiente (**Tabla 15**), indica el resumen de las pruebas realizadas con los dos escenarios.

Tabla 15: Media muestral

	Sensor 1	Sensor 2	Sensor 3	Proporción Muestral
Ambiente A	88,14	112,44	145,97	0,39
Ambiente B	85,97	76,85	74,14	0,27

Fuente.- Jenny Patín Chimbo

En la (Tabla 16) indica la frecuencia esperada.

Tabla 16: Frecuencia esperada

	Sensor 1	Sensor 2	Sensor 3
Ambiente A	107,52	113,47	125,56
Ambiente B	73,52	77,59	85,85

Fuente.- Jenny Patín Chimbo

En la (Tabla 17) indica los valores estadísticos.

Tabla 17: Tabla de valores estadísticos

	R	3
	K	3
Grados de Libertad	$(r-1)(k-1)$	4
Nivel de significación	A	0,005
Chi Cuadrado	χ^2	14,860259
Probabilidad	P	0,019092482
X2Prueba		11,77656481

Fuente.- Jenny Patín Chimbo

2.6.8.6. Establecer la regla de decisión

Como $11.77 < 14.86$, se puede decir que la hipótesis es verdadera. Como se puede observar en la siguiente (Ilustración 94), el valor crítico de t cae en la zona

de aceptación de la hipótesis H_0 .

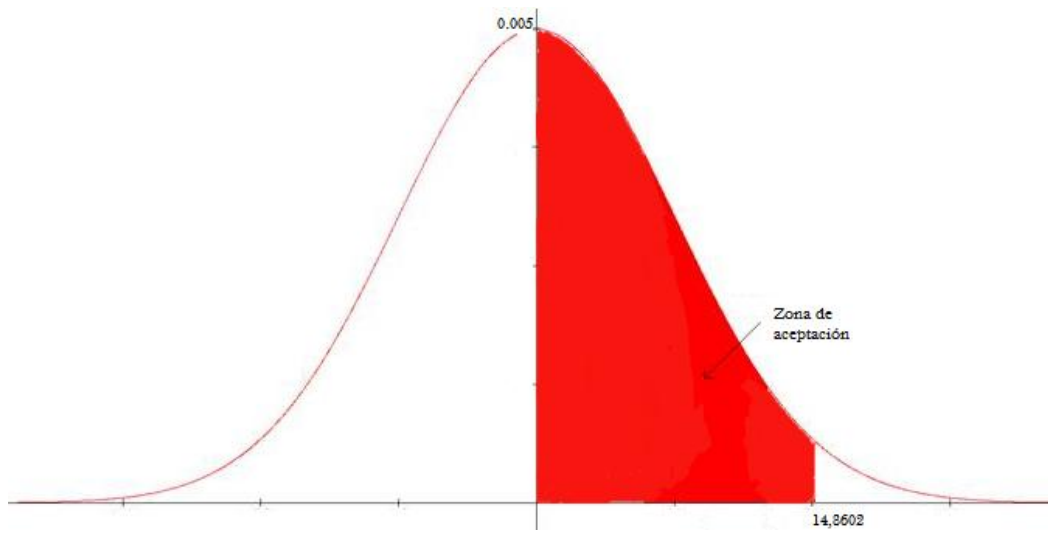


Ilustración 94: Campana de Gauss para decisión

Fuente.- Jenny Patín Chimbo

CAPÍTULO III

3. RESULTADOS

Este capítulo, detalla las pruebas realizadas entre los dispositivos de control y el software de monitoreo, con comunicación inalámbrica.

3.1. PROCESAMIENTO DE LA INFORMACIÓN

Para realizar el procesamiento de la información, se utiliza un analizador de redes que captura el tráfico de la red inalámbrica. Se analizó el rendimiento de la red considerando los siguientes es parámetros:

- Velocidad de transmisión
- Paquetes enviados, transmitidos, perdidos.
- Tiempo de transmisión

Se procede a definir los escenarios.

- En el escenario A se considera el sistema de sensores formado por el sensor de temperatura, de humedad relativa y humedad del suelo los datos son tomados de manera manual, el tiempo en milisegundos, sin monitoreo
- En el escenario B se considera el sistema propuesto que se encuentra implementado los datos se toma de forma automática, el tiempo en milisegundos, con monitoreo.

3.2. PRUEBAS DE COMUNICACIÓN

Las pruebas se realizan tomando en cuenta que el sistema este con monitoreo y sin monitoreo del sistema propuesto.

3.2.1. Velocidad de transmisión

En la (Ilustración 95), se muestra las velocidades transmitidas.

Tabla 18: Velocidad de Transmisión

	Rata(Kbps)	%
Ambiente A	3305	57,68
Ambiente B	9600	100,00

Fuente.- Jenny Patín Chimbo

En la (Tabla 18) muestra la velocidad de transmisión en paquetes por segundo y rate en porcentaje, considerando a la más alta velocidad con el peso de 100%, ya que se supone que es el nivel óptimo o máximo que va alcanzar el sistema.

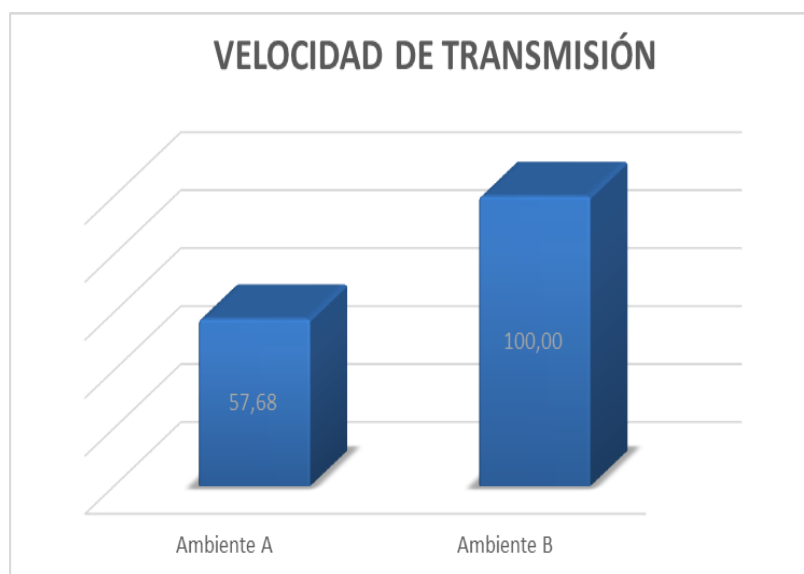


Ilustración 95: Velocidad de Transmisión

Fuente.- Jenny Patín Chimbo

3.2.2. Intercambio de paquetes

El sniffer utilizado muestra el tráfico total generado. Se obtienen los paquetes perdidos del tráfico capturado TCP.

Tabla 19: Paquetes Capturados durante la Transmisión

	Paq. Tx	Paq. Rx	Paq. Loss
Ambiente A	477	900	423
Ambiente B	677	743	66

Fuente.- Jenny Patín Chimbo

En la (Tabla 19) y (Ilustración 96), se observa un resumen de los paquetes el momento de la transmisión del archivo.

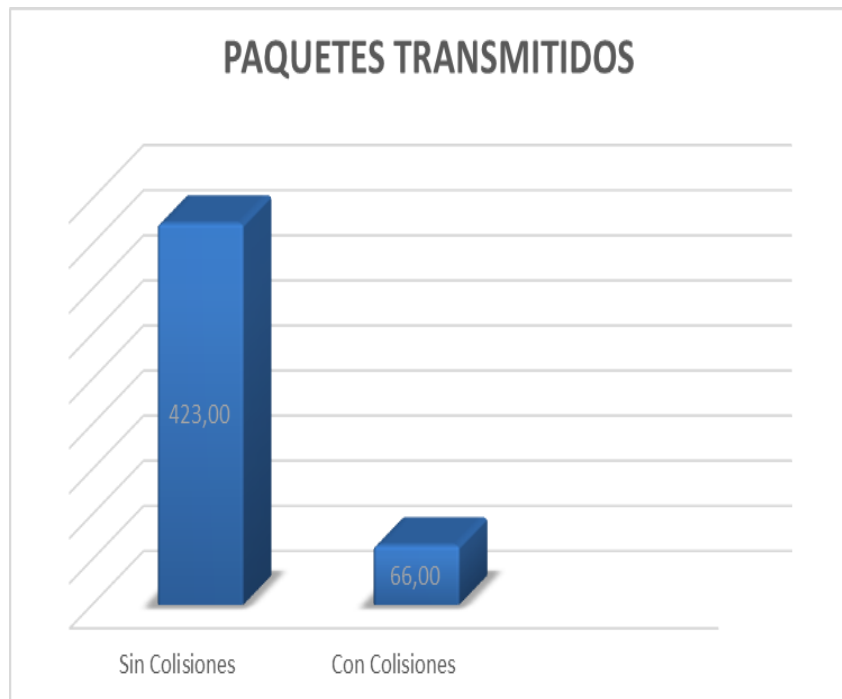


Ilustración 96: Paquetes Perdidos en la Transmisión

Fuente.- Jenny Patín Chimbo

3.2.3. Tiempo de transmisión

Otro factor importante es el tiempo que se demora en ver la administración SNMP. En la (Tabla 20) y (Ilustración 97), se muestra los tiempos en segundos.

Tabla 20: Tiempo de Transmisión

	Duración	S	%
Ambiente A	2:50:41	124	56,452
Ambiente B	1:04:01	70	100,000

Fuente: Jenny Patín Chimbo

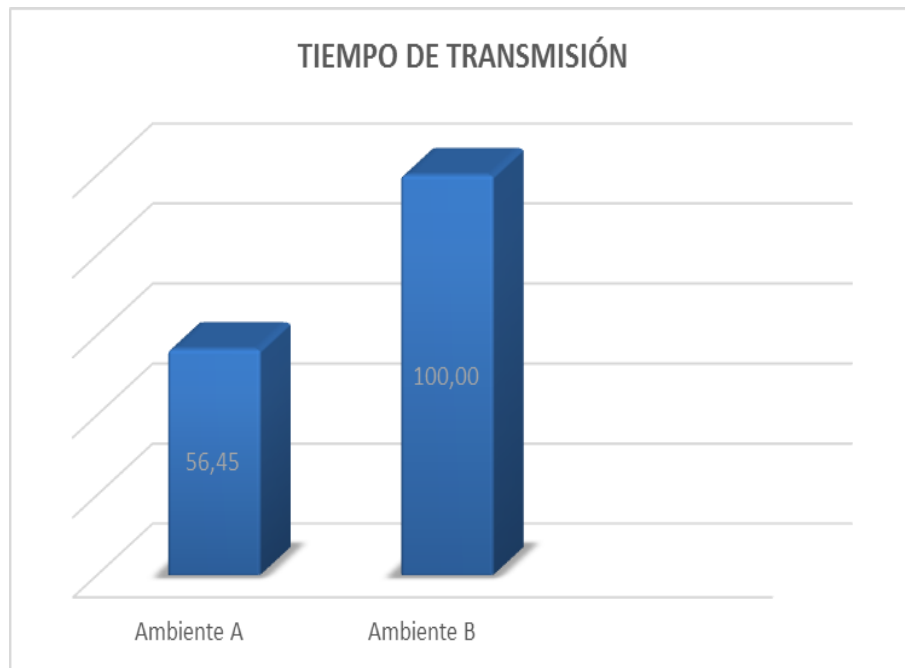


Ilustración 97: Tiempo de Transmisión en Segundos

Fuente.- Jenny Patín Chimbo

3.3. ANÁLISIS FINANCIERO

La (Tabla 21), muestra el costo de los recursos materiales que se usó en el proyecto de diseño e implementación, donde los recursos materiales justifica los beneficios ofrecen los dispositivos.

Tabla 21: Recursos Materiales

CANT.	DETALLE	COSTO UNITARIO	COSTO TOTAL
2	Arduino MEGA R3	\$45.00USD	\$90.00USD
1	Router DLINK DIR-600	\$35.00USD	\$35.00USD
2	Modulos XBEE S2	\$45.00USD	\$90.00USD
2	Shield Xbee 03	\$18.00USD	\$36.00USD
1	Shield Rele 4Ch 5V	\$20.00USD	\$20.00USD
2	Sensores DHT22	\$20.00USD	\$40.00USD
2	Sensor de humedad de suelos SOIL	\$12.00USD	\$12.00USD
2	Cajas ENCLOUSER	\$8.00USD	\$16.00USD
1	Micro Ventilador 5V	\$8.00USD	\$8.00USD
1	Mini Bomba de 12V	\$15.00USD	\$15.00USD
1	Maqueta	\$45.00USD	\$45.00USD
40	Cables Dupon	\$0.08USD	\$3,50 USD
TOTAL			\$410.50USD

Fuente.- Jenny Patín Chimbo

La (Tabla 22) se muestra los recursos varios que se usó en el proyecto.

Tabla 22: Recursos Varios

CANT.	DETALLE	COSTO UNITARIO	COSTO TOTAL
2	Lápiz Rotring	\$3.00USD	\$6.00USD
3	Hojas de impresión	\$5.00USD	\$15.00USD
2	Libretas de apuntes	\$3.00USD	\$6.00USD
1	Cartucho de tinta	\$15.00USD	\$15.00USD
2	USB	\$8.00USD	\$16.00USD
TOTAL			\$58.00USD

Fuente.- Jenny Patín Chimbo

La (Tabla 23) muestra los recursos financieros y gastos totales del proyecto.

Tabla 23: Recursos Financieros

Rubro	Recursos de los participantes
Recursos Materiales	\$410.50USD
Recursos Varios	\$58.00USD
TOTAL	\$468.50USD

Fuente.- Jenny Patín Chimbo

CAPÍTULO IV

4. DISCUSIÓN

El desarrollo del prototipo, están basado en la automatización, a través de una exhaustiva investigación y recolección de información de las nuevas técnicas proyectadas para el control, se logró encontrar los módulos adecuados que permiten el desarrollo de un mecanismo estable y una administración de red SNMP que brinda a los usuarios la interfaz adecuada para la conexión.

Se puso mucho interés en el tipo de comunicación tomando como base de estudio la red ZigBee mediante el uso de los módulos XBee, el uso frecuente de tecnología permite realizar redes de fácil uso, ya que estos módulos transmiten y reciben la información inalámbricamente, cada vez estos puntos se siguen extendiendo y multiplicando que ahora permiten a los usuarios alcanzar largas distancias de transmisión y recepción de datos, conexión local.

Para la administración SNMP se utiliza el Modem D Link el cual ofrece acceso a internet, es ilimitado y de gran velocidad, al tomarlo como punto de estudio en el proyecto, se ayuda aprovechar al beneficiario los múltiples servicios que puede conseguir con el internet.

La MIB utilizada para el control de los sensores de temperatura cumple con un esquema lógico, donde debe cumplir cada una de sus etapas desde el reconocimiento de los sensores hasta la transmisión de los datos a través de la red inalámbrica, para así lograr que la comunicación sea eficiente y eficaz, además se debe tomar en cuenta cada una de las necesidades del usuario desde un fácil uso hasta una interfaz amigable.

El software que se utilizó para el desarrollo de la aplicación, está basado en un estudio de las nuevos sistemas operativos utilizados por los usuarios en las PCs como el sistema Nagios, que se han considerado como necesario para el monitoreo de actividades, control de dispositivos, etc., por su fácil manejo, rapidez y precisión.

Así pues culminado los equipos, son capaces de cumplir con todos los requerimientos planteados, de una forma eficiente, sin pérdida mayor de datos, velocidad de transmisión donde los actuadores funciones de forma manual y automática.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- La elaboración de un circuito y programación del mismo para que cumpla con los estándares de un protocolo determinado requieren de un nivel muy elevado de conocimientos y de investigación para la elaboración del mismo, ya que dicho sistema debe poder ser analizado, testeado y ser compatible con los demás equipos o sistemas bajo el mismo protocolo.
- Al utilizar el protocolo SNMP para el monitoreo de un invernadero se facilita enormemente ya que se puede conectar un sinnúmero de sensores a una sola dirección IP mediante la cual se podrá monitorear remotamente cada uno de dichos sensores sin la necesidad de agregar más equipos costosos a más del emisor (TX) del invernadero, de esta manera se puede bajar costos y mejorar la productividad del invernadero al ser posible el monitoreo del mismo.
- Después de haber realizado las pruebas respectivas se verifica que se cumplen con los objetivos planteados, es decir, se realiza el control y monitoreo de la temperatura ambiental para que no supere su valor establecido. La importancia de la implementación de mediciones temperatura, humedad relativa y humedad del suelo, radica en lograr mantener un servicio constante y adecuado mediante la prevención de eventos fuera de la cotidianidad de la operación de los equipos de telecomunicaciones, como fallas de suministro eléctrico o fenómenos atmosféricos que afecten su funcionamiento cotidiano
- El monitoreo en Nagios representa la mejor opción para el monitoreo del

sistema de control y los actuadores asociados la posibilidad de utilizar un asistente para la configuración de equipos como Routers o Switchs facilita su monitoreo, además de ser amigable con MIB generadas a partir de interfaces adaptativas creadas por tercero y que contienen información que no es nativa del protocolo SNMP.

5.2. RECOMENDACIONES

- Es importante garantizar el funcionamiento adecuado de los elementos que componen los sistemas, como la temperatura ambiente, la humedad del suelo y la humedad relativa para el funcionamiento de los actuadores dentro del invernadero.
- Los equipos que permitan este tipo de monitoreo, resultan en un aumento significativo del costo de los sistemas y en casos de sistemas ya implementados a menudo se cuenta con equipos que no cuentan con la capacidad de implementación de monitoreo SNMP.
- Usar una placa Arduino Mega 2560 en lugar de una Arduino Uno R3 para la parte receptora (RX) ya que al cargar el programa en la memoria no queda mucha memoria RAM libre para el procesamiento de datos, haciendo inestable el sistema de SNMP.
- Usar un router con acceso a la configuración SNMP como los CISCO, HUAWEY o HP, los router comerciales como DLINK no sirven para poder implementar la red con conexión al shield ethernet W5100 y acceso a las OID del sistema SNMP.

CAPÍTULO VI

6. PROPUESTA

6.1. TÍTULO DE LA PROPUESTA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE HUMEDAD Y TEMPERATURA PARA INVERNADEROS CON ADMINISTRACIÓN SNMP.

6.2. INTRODUCCIÓN

En la actualidad se cuenta con invernaderos no automatizados y se ha buscado implementar uno incorporando a su vez una gestión de redes mediante el protocolo SNMP, las variables que se requieren controlar son la temperatura ambiental del invernadero, la humedad del suelo y la humedad relativa para el cual se utilizaron sensores de temperatura y humedad. Los sensores envían la información por medio de los Xbee (comunicación inalámbrica), que a su vez transmiten inalámbricamente a una tarjeta Arduino, encargado de realizar el control de la temperatura y humedad, la información se recopila en una base de datos, se procede a utilizar software Nagios para visualizar las información y enviar una alerta mediante un correo electrónico al usuario, los mismos que deben encontrarse en una red LAN con la IP: 192.168.1.1 con la mascara por defecto 255.255.255.0 los cuales se encargan de erutar la red y que el sistema este siendo administrada por medios SNMP (Protocolo simple de Gestión de red).

Para el control de la temperatura se utiliza un ventilador el cual se activa o se apaga dependiendo de los parámetros del cultivo. Para la humedad se controla una bomba que se activa o se apaga dependiendo de la humedad los mismos que estarán siendo controlados manualmente mediante un monitoreo por CMD y automáticamente mediante un Sniffer.

6.3. OBJETIVOS

6.3.1. OBJETIVO GENERAL

- Diseñar e implementar de un sistema de monitoreo y control de humedad y temperatura para invernaderos con administración SNMP.

6.3.2. OBJETIVOS ESPECIFICOS

- Analizar un software de gestión de redes de bajo costo, que se pueda utilizar para el monitoreo de sistemas de control climático (Temperatura y Humedad).
- Diseñar un dispositivo de control que permitan controlar la temperatura y humedad en forma manual y automática, integrar al sistema de monitoreo los equipos auxiliares que no dispongan de capacidad SNMP.
- Proponer alternativas para incorporar al sistema de monitoreo, facilidades de gestión o ejecución de software de terceros con administración SNMP.

6.4. FUNDAMENTACIÓN CIENTÍFICO –TÉCNICA

Para la implementación de los dispositivos de temperatura y humedad, se considera diversos elementos que forman parte del proceso, los cuales cumplen un papel importante dentro del sistema de monitoreo.

EL dispositivo está formado por un sensor DHT22 y el sensor Soil, el sensor DHT22 está en el rango de valores desde -40°C hasta 80°C de temperatura. Con una precisión: $\pm 0.5^{\circ}\text{C}$, $\pm 1^{\circ}\text{C}$ como máximo en condiciones adversas. El tiempo de

respuesta: <10 segundos, es decir, de media, tarda menos de 10 segundos en reflejar un cambio de temperatura real en el entorno. La Humedad relativa está en el rango de valores desde 0% hasta 99.9% de Humedad Relativa. Con una precisión: $\pm 2\%RH$, a una temperatura de 25°C. El tiempo de respuesta: <5 segundos, es decir, de media, tarda menos de 5 segundos en reflejar un cambio de humedad relativa real en el entorno. Además, para darse esta afirmación, los tests indicaron que la velocidad del aire debe ser de 1 m/s. El protocolo de comunicación es a través de un único hilo (protocolo 1 wire), por lo tanto hace que la integración de este sensor sea rápida y sencilla.

El sensor Soil-Clik simplifica la detección de humedad del suelo. El sensor Soil-Clik utiliza tecnología probada para medir la humedad en la zona radicular. Cuando la sonda detecta que el suelo ha alcanzado el nivel de humedad deseado, se cerrará el riego, evitando malgastar agua.

Como parte primordial se tiene el Módulo Mega 2560 R3 que posee un microcontrolador ATmega, esta se encarga de analizar los datos recibidos de los sensores el cual es enviado mediante el módulo XBee para luego poder ser visualizado en el software de monitoreo Nagios, conjuntamente con el software de monitoreo y el protocolo de administración de redes SNMP envía una alerta al usuario mediante un e-mail de alerta.

6.5. DESCRIPCIÓN DE LA PROPUESTA

El proyecto parte del uso de sensores de temperatura y humedad, para poder determinar las condiciones ambientales adecuadas para el cultivo del tomate riñón. Entre los sensores de temperatura y humedad están el DHT22 que es capaz de enviar la temperatura exacta y la humedad relativa exacta contando con la información se recopila cas 2s. Por lo tanto es necesario el desarrollo de un sistema de control automático para invernadero, que sea flexible y pueda adaptarse a diferentes circunstancias de uso, que pueda ser entendido y

manipulado fácilmente por el usuario y que además sea eficientes en el uso de agua y energía.

La propuesta se fundamenta en la elección de un software de monitoreo y gestión de red y analizar su viabilidad de aplicación en un prototipo de sistema de comunicación; el cual deberá incluir una red inalámbrica y actuadores. Además, el sistema a proponer debe permitir ejecutar software de gestión de terceros, y facilidades para incorporar nuevos elementos conforme sea necesario. Los parámetros a monitorear, son la temperatura, humedad relativa y humedad del suelo en un invernadero con administración SNMP.

6.6. DISEÑO ORGANIZACIONAL.

A continuación se presenta la estructura funcional de la unidad administrativa con la que se ejecutó el proyecto (**Ilustración 98**).

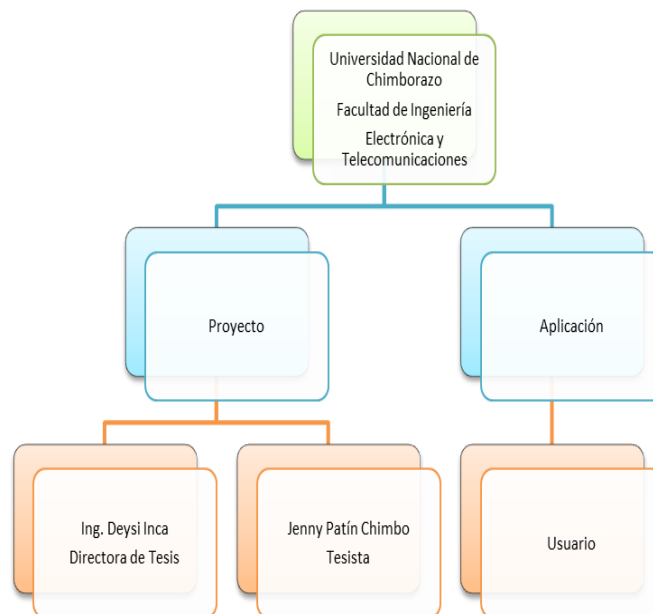


Ilustración 98: Diagrama del sistema organizacional del proyecto

Fuente: Jenny Patín Chimbo

6.7. MONITOREO Y EVALUACIÓN DE LA PROPUESTA

Para el monitoreo y evaluación de la propuesta, se realizara pruebas de comunicación entre los dispositivos y la administración SNMP, así como un seguimiento de la disponibilidad de la red para el control local, confiabilidad de transmisión y recepción de datos desde el invernadero, para poder evaluar al equipo de una manera eficiente, además se documentará cada avance que se vaya generando durante cada etapa del desarrollo del proyecto, para perfeccionar los equipos y establecer las limitaciones que presenta.

7. BIBLIOGRAFÍA

1. H. U. Villamil, “Microprocesadores,” en *Modulo de curso Microprocesadores y Microcontroladores*, 2013 ed. Chiquinquirá, Colombia,
2. Aguirre Solvez Lorena, Estudio de una red de sensores sin hilos basados en la tecnología Arduino bajo protocolos de comunicaciones Zigbee, Editorial Politécnica, Catalunya – España.
3. Arlanzón David, Introducción a la plataforma Open Source Arduino para aplicaciones domóticas y automatización, Catalunya – España, 2012
4. Banzi, M. (2012). Arduino. Obtenido de What is arduino: <http://www.arduino.cc/>
5. Diccionario. (2014). Diccionario de la Real Academia de la Lengua.
6. es.wikipedia.org/wiki/Sensor. (2014)
7. Rodríguez, Rafael, Tabares, José, Medina, José, Cultivo moderno del tomate, Madrid, Ediciones Mundi-Prensa,1997, pág. 255.
8. Torrente Artero Oscar, Arduino: Curso práctico de formación, Madrid-España.-Grupo RC. 2013
9. Ernesto Ariganello y Enrique Barrientos Sevilla, Redes Cisco CCNP a Fondo. Guía de estudio para profesionales, Madrid, España
10. Reyes, C. A. (2008). Microcontroladores PIC y programación en Basic. Quito
11. Ávalos G Luisa, “Invernadero”, Perú, febrero 2004
12. Ortega, E. (junio de 2014). Slideshare. Obtenido de es.slideshare.net/clasificacion-de-sensores-36167798

Request for Comments

RFC 1157 (SNMP, 1990)

RFC 1155 (SNMP, 1990)

RFC 3410 (SNMPv3, 2002)

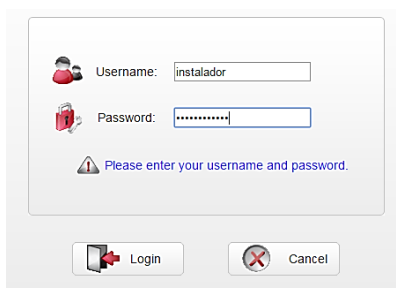
Webgrafía

- http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-series-switches/product_data_sheet0900aecd80322c0c.pdf
- <https://www.nagios.com/products/nagios-xi/>
<https://code.google.com/p/agentuino/>
- <http://es.arduino.cc>
- <http://www.datasheet.com>

8. APÉNDICES Y ANEXOS

8.1. ANEXO 1.- CONFIGURACIÓN DEL MODEM DOMICILIARIO PARA LA COMUNICACIÓN INALÁMBRICA LOCAL Y REMOTA

- Abrir un navegador web e ingresar la siguiente dirección 192.168.1.1 donde se digita: (**Ilustración 99**)
 - Usuario: admin
 - Password: .admin



Username: instalador

Password:


Please enter your username and password.

Login Cancel

Ilustración 99: Ingreso interfaz de configuración del modem

Fuente.- Jenny Patín Chimbo

- Automáticamente aparece el interfaz donde se encuentran las configuraciones del modem, para verificar la IP fija contratada dar un click en Status/WAN/Network en el apartado IP information (**Ilustración 100**).



Connection Name	Status	Online Duration
INTERNET_TROAD_R_0_35	Connected	00:01:08
INTERNET_E_0_36	Connected	00:00:00
INTERNET_E_0_37	Connected	00:00:00
INTERNET_E_0_39	Connected	00:00:00
INTERNET_E_0_40	Connected	00:00:00
INTERNET_E_0_41	Connected	00:00:00
EU080	Disconnected	00:00:00

Connection Name	IP Address	Subnet Mask
INTERNET_TROAD_R_0_35	190.152.117.80	255.255.255.255
INTERNET_E_0_36

Ilustración 100: Interfaz del modem - ip fija asignada

Fuente.- Jenny Patín Chimbo

- Para abrir los puertos en ADVANCED/NAT/Port Mapping y click en la pestaña New. En los Settings se completan los campos del siguiente modo:

- Protocolo: selecciona entre UDP o TCP.
 - Puerto externo inicial: introducir el número de puerto que utiliza la aplicación.
 - Puerto Externo final: introducir el número de puerto que utiliza la aplicación.
 - Equipo interno: la IP para el que se quieren abrir los puertos.
 - Puerto Interno: introducir el número del puerto que se quiere habilitar.
 - Nombre Asignado: escribir el nombre de la aplicación.
- Como se muestra en la **Ilustración 101**, se habilita la IP 192.168.1.6.

Advanced > NAT > Port Mapping

ALG DMZ **Port Mapping** Port Triggering

Mapping Name	Interface	Protocol	Remote Host	External Start Port	External End Port	Internal Port	Internal Host	Enable
WIFLY	INTERNET_TR069_R_0_35	TCP/UDP		2000	2000	2000	192.168.1.7	Enable
MODE-WIFLY	INTERNET_TR069_R_0_35	TCP/UDP		25	25	25	190.152.117.80	Enable

Settings

Type: Customization Application Choose...

Interface: INTERNET_TR069_R_0_35

Protocol: TCP/UDP

Remote host:

External start port: 2000

External end port: 2000

Internal host: 192.168.1.7

Internal port: 2000

Mapping name: WIFLY

Submit

Ilustración 101: Configuración para habilitación de puertos

Fuente.- Jenny Patín Chimbo

- Con la ayuda del software PORTFORWARD se comprueba si el puerto de la tarjeta está habilitada con la opción PORTCHECKER. (**Ilustración 102**)

Port Number To Check: 2000

Protocol: TCP UDP

Check Me

Notes
TCP testing can detect if a TCP port is forwarded to either this computer or another device on the network like a DVR or an IP camera.
UDP testing only works if a UDP port is forwarded to this computer. You can not test UDP forwards to other devices.

External IP Address: **Your external IP address is: 190.152.117.80**

Port Check Result: **Your port is OPEN on another device!**

Ilustración 102: Interfaz de Portforward

Fuente.- Jenny Patín Chimbo

- En la pestaña Status, se podrá observar la información que rastrea las IP como se observa en **Ilustración 103**.



Ilustración 103: Comprobación del puerto habilitado

Fuente.- Jenny Patín Chimbo

8.2. ANEXO 2.- PRUEBAS DE COMUNICACIÓN Y MONITOREO.

```

GA. Administrador: Símbolo del sistema
C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
SNMPv2-SMI::experimental.1.3.2.1.3.0 = INTEGER: 46

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
SNMPv2-SMI::experimental.1.3.2.1.3.0 = INTEGER: 46

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
Timeout: No Response from 192.168.1.6.

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.3.0
Timeout: No Response from 192.168.1.6.

C:\usr>snmpget -v 1 -r 1 -c public 192.168.1.6 1.3.6.1.3.1.3.2.1.2.0
SNMPv2-SMI::experimental.1.3.2.1.2.0 = INTEGER: 25

C:\usr>

```

Ilustración 104: Comprobación de comunicación mediante el CMD

Fuente.- Jenny Patín Chimbo

No.	Time	Source	Destination	Protocol	Length	Info
288	628.056220	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
289	629.026453	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
290	631.077290	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
291	631.077345	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
292	631.537695	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
293	632.540538	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
294	633.127675	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
295	634.452434	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
296	635.178643	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
297	635.178700	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
298	635.462373	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
299	637.229009	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
300	637.229065	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
301	638.026419	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
302	639.040455	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.3.0
303	639.279713	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
304	639.279756	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
305	641.330093	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
306	641.330152	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
308	643.380688	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.3.0
309	643.380743	192.168.1.2	192.168.1.6	ICMP	125	Destination unreachable (Port unreachable)
315	648.221349	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
316	649.227835	192.168.1.2	192.168.1.6	SNMP	85	get-request 1.3.6.1.3.1.3.2.1.2.0
318	649.527677	192.168.1.6	192.168.1.2	SNMP	97	get-response 1.3.6.1.3.1.3.2.1.2.0

Ilustración 105: Comprobación de monitoreo mediante el Sniffer

Fuente.- Jenny Patín Chimbo

8.3. ANEXO 3.- DIAGRAMA DE FLUJO DEL PROYECTO.

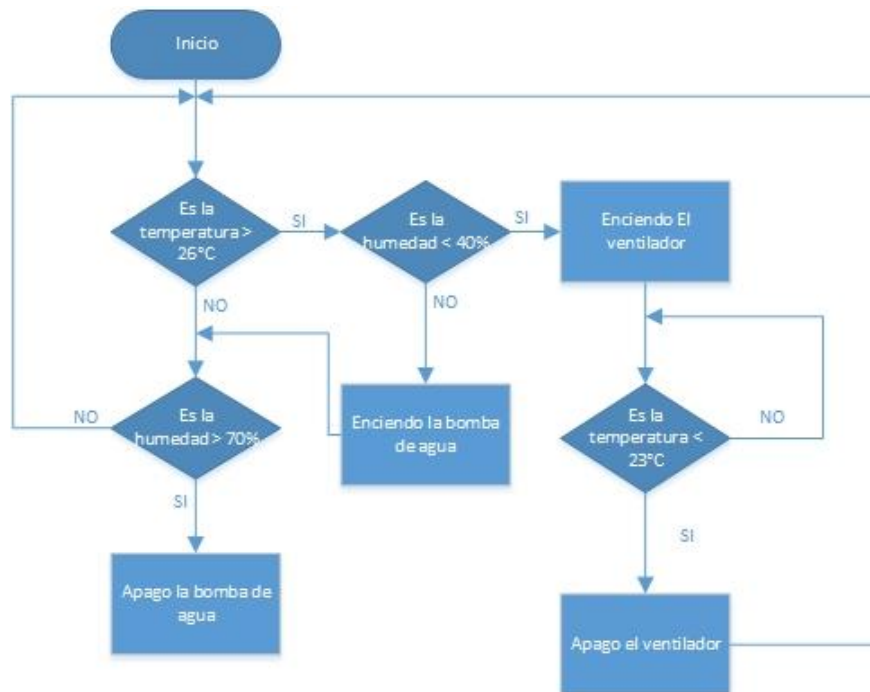


Ilustración 106: Diagrama de flujo

Fuente.- Jenny Patín Chimbo

8.4. ANEXO 4.- DISEÑO DE LOS CIRCUITOS

En la **(Ilustración 107)** se puede ver el diagrama esquemático del circuito receptor.

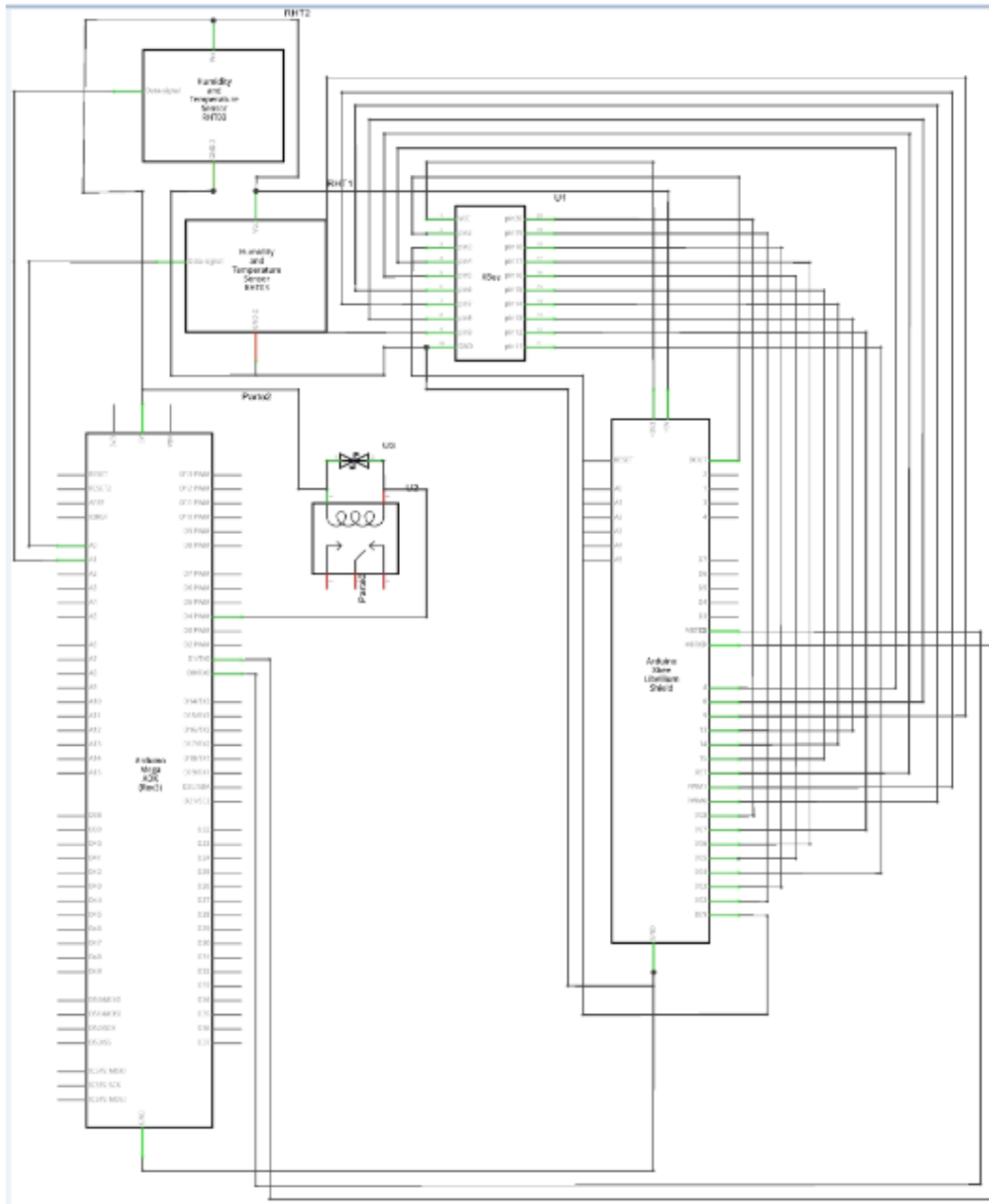


Ilustración 107: Circuito Esquemático Receptor

Fuente.- Jenny Patín Chimbo

8.5. ANEXO 5.- 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 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

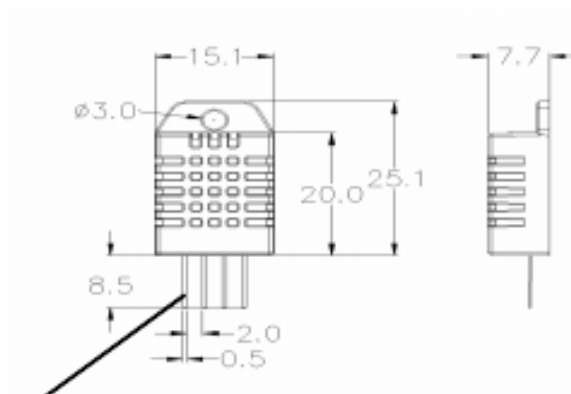
http://www.aosong.com

Email: thomasliu198518@yahoo.com.cn, sales@aosong.com

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

Accuracy	humidity $\pm 2\%RH$ (Max $\pm 5\%RH$);	temperature ± 0.2 Celsius
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity $\pm 1\%RH$;	temperature ± 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, 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.

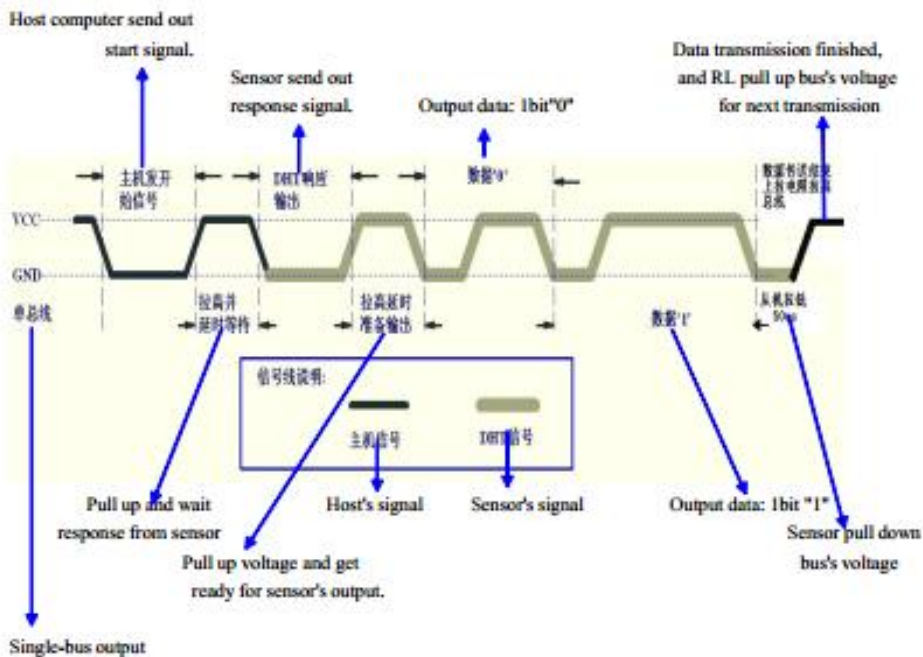
AM2303 send out higher data bit firstly!

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>

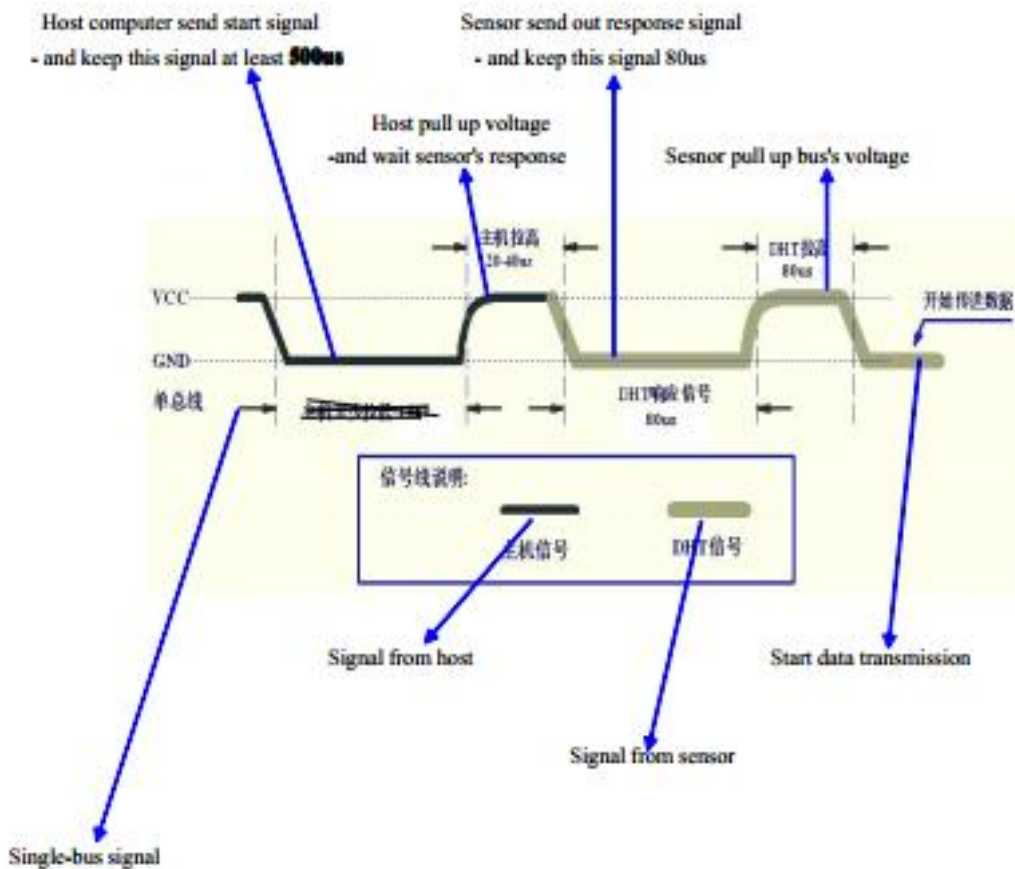
Email: thomasliu198518@yahoo.com.cn sales@aosong.com

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:



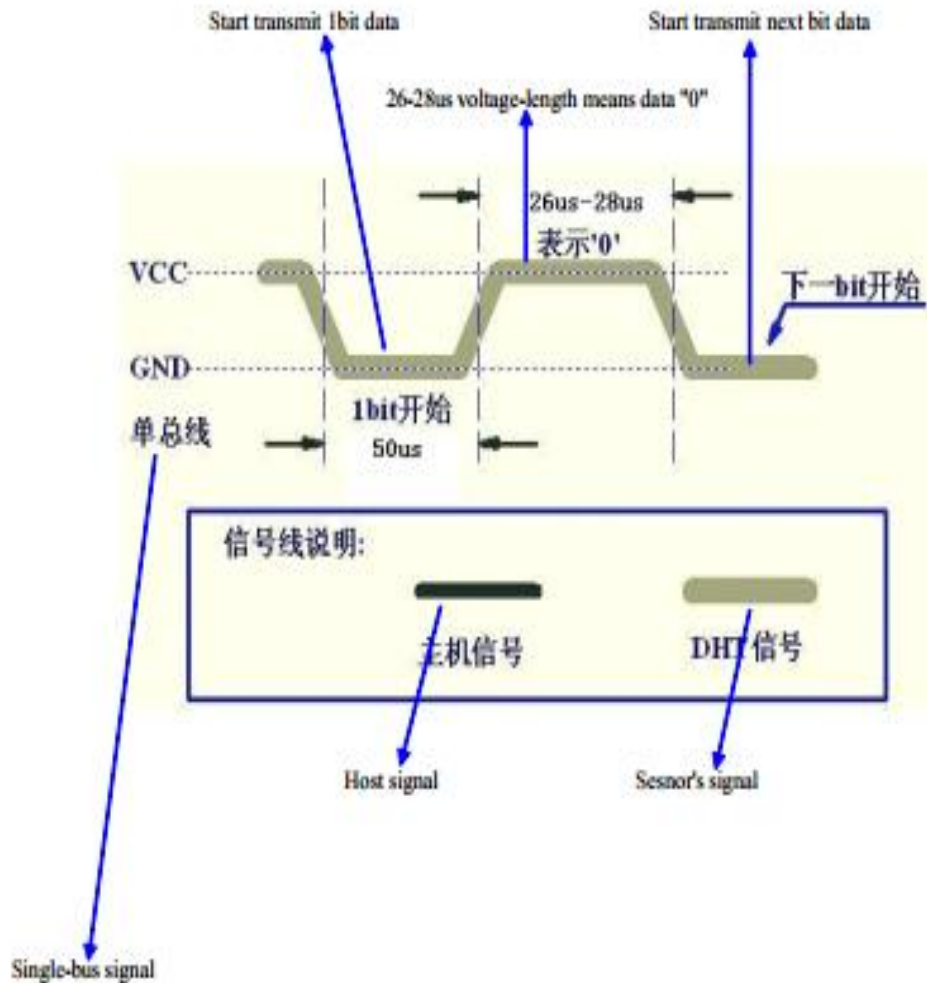
Aosong(Guangzhou) Electronics Co.,Ltd

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

<http://www.aosong.com>

Email: thomasliu198518@yahoo.com.cn sales@aosong.com

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



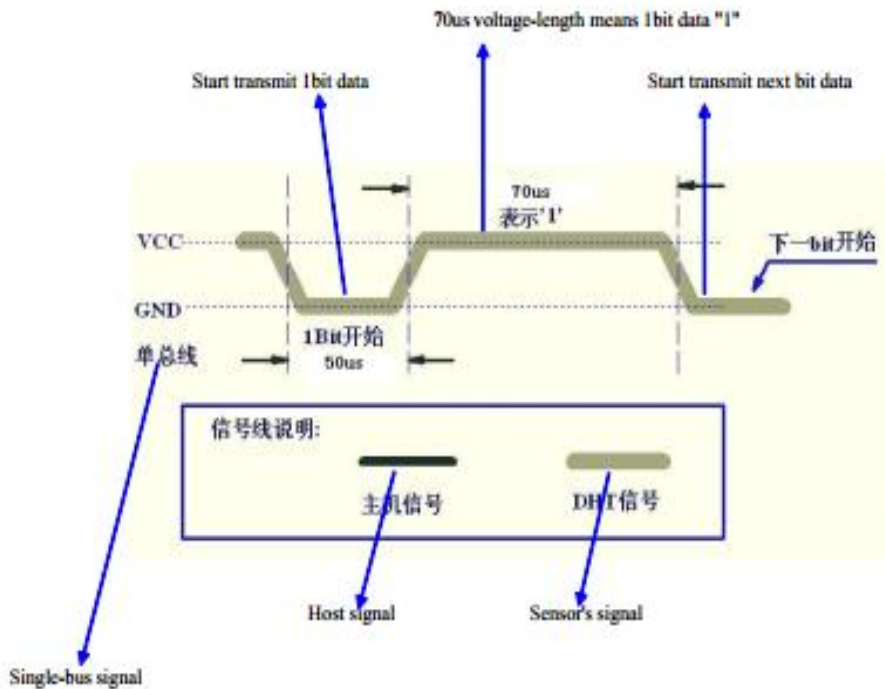
Aosong(Guangzhou) Electronics Co.,Ltd

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

<http://www.aosong.com>

Email: thomasliu198518@yahoo.com.cn 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, 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.

8.6. ANEXO 6.- HOJA DE DATOS ARDUINO MEGA 2560



Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V

8-bit Atmel Microcontroller with 16/32/64KB In-System Programmable Flash

DATASHEET

Features

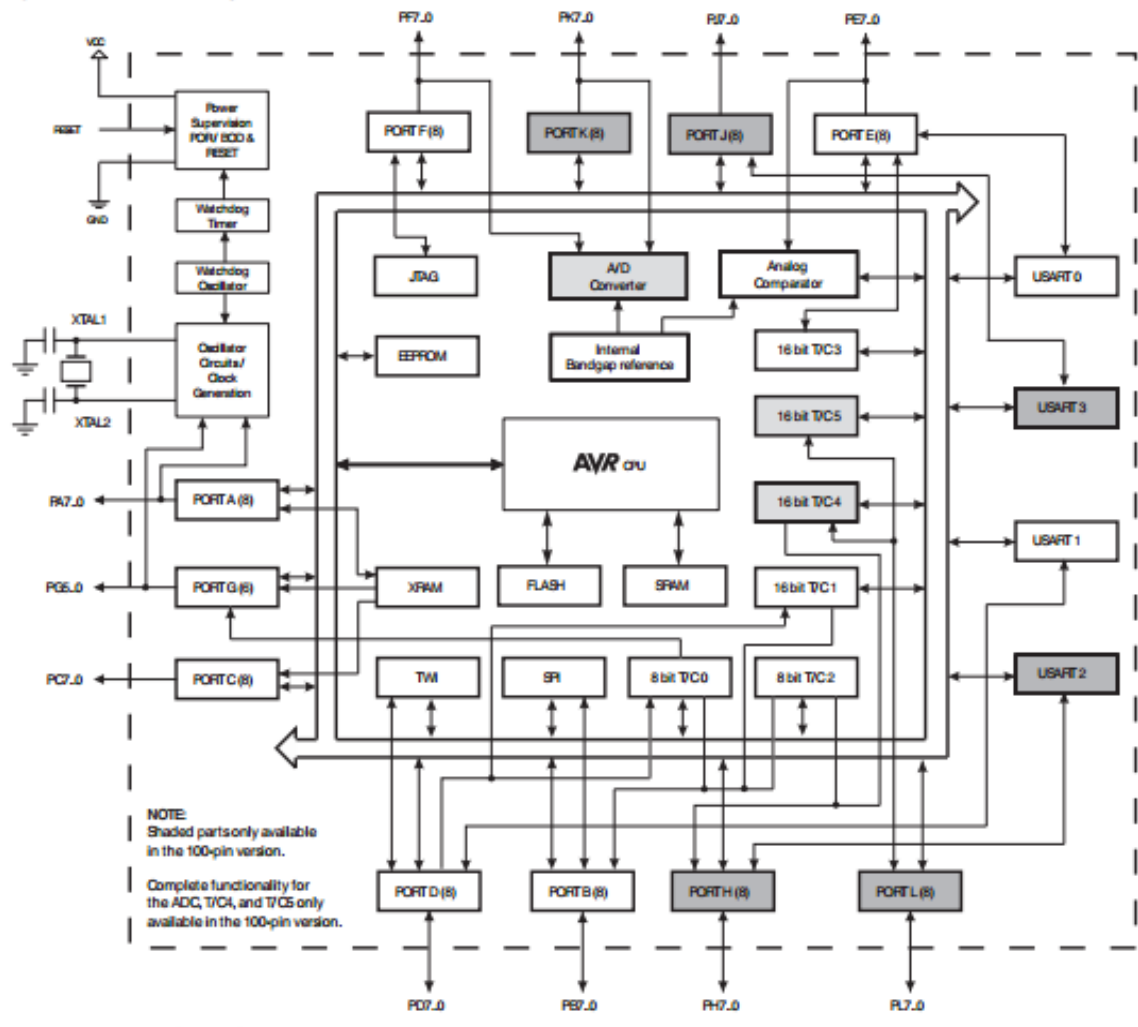
- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles:10,000/Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C
 - Optional Boot Code Section with Independent Lock Bits
 - * In-System Programming by On-chip Boot Program
 - * True Read-While-Write Operation
 - Programming Lock for Software Security
 - * Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 54/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
 - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
 - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
 - RoHS/Fully Green
- Temperature Range:
 - -40°C to 85°C Industrial
- Ultra-Low Power Consumption
 - Active Mode: 1MHz, 1.8V: 500µA
 - Power-down Mode: 0.1µA at 1.8V
- Speed Grade:
 - ATmega640V/ATmega1280V/ATmega1281V:
 - * 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega2560V/ATmega2561V:
 - * 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega640/ATmega1280/ATmega1281:
 - * 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
 - ATmega2560/ATmega2561:
 - * 0 - 16MHz @ 4.5V - 5.5V

2. Overview

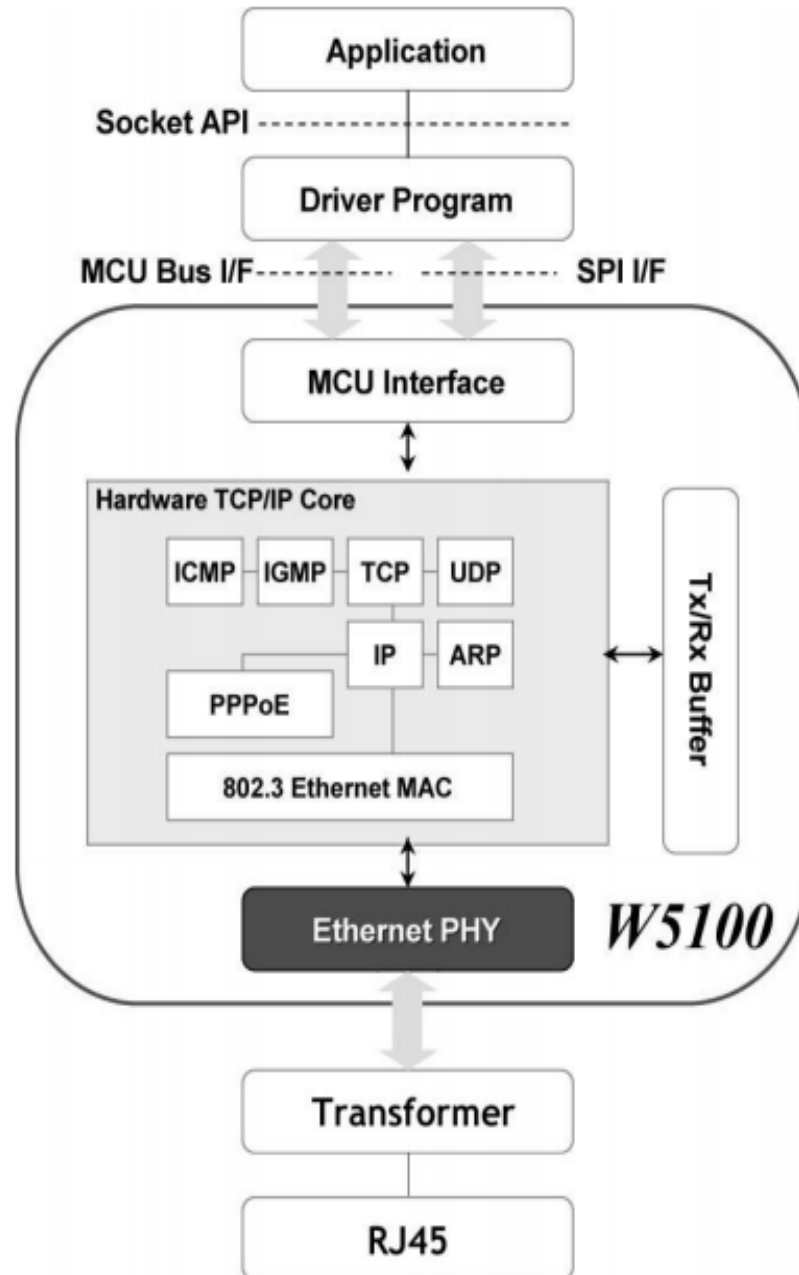
The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



8.7. ANEXO 7.- HOJA DE DATOS ARDUINO ETHERNET SHIELD Block Diagram



8.8. ANEXO 8.- HOJA DE DATOS XBEE



EMBEDDED RF
MODULES FOR OEMS



XBEE® 802.15.4 RF MODULES

Low-cost, easy-to-deploy modules provide critical end-point connectivity to devices and sensors

XBee 802.15.4 RF modules provide OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/Mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the XBee can substitute one XBee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-to-market.

XBee 802.15.4 RF modules are ideal for applications requiring low latency and predictable communication timing. Providing quick, robust communication in point-to-point, peer-to-peer, and multipoint/star configurations, XBee 802.15.4 products

enable robust end-point connectivity with ease. Whether deployed as a pure cable replacement for simple serial communication, or as part of a more complex hub-and-spoke network of sensors, XBee 802.15.4 RF modules maximize performance and ease of development.

XBee 802.15.4 modules seamlessly interface with compatible gateways, device adapters and range extenders, providing developers with true beyond-the-horizon connectivity.

BENEFITS

- Point-to-multipoint network topology
- 2.4 GHz for worldwide deployment
- 900 MHz for long-range deployment
- Fully interoperable with other other Digi networking products, including gateways, device adapters and range extenders
- Common XBee footprint for a variety of RF modules
- Low-power sleep modes
- Multiple antenna options
- Industrial temperature rating (-40° C to 85° C)

RELATED PRODUCTS



ConnectPort®
X4/X4H Gateways



XBee®
Adapters



XCTU

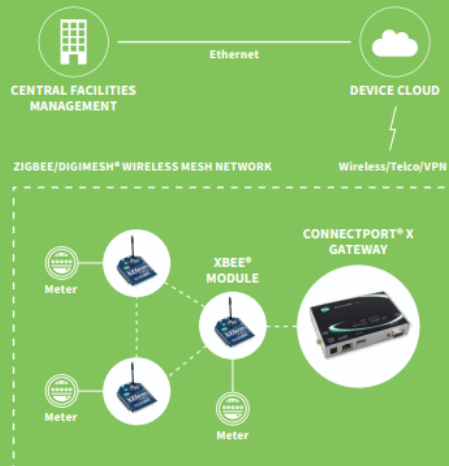


Digi Device
Cloud™

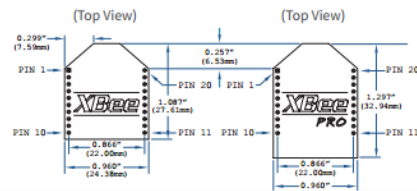
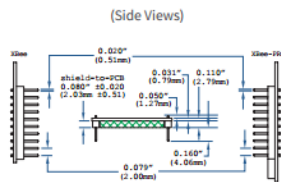
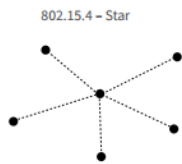


Development
Kits

APPLICATION EXAMPLE

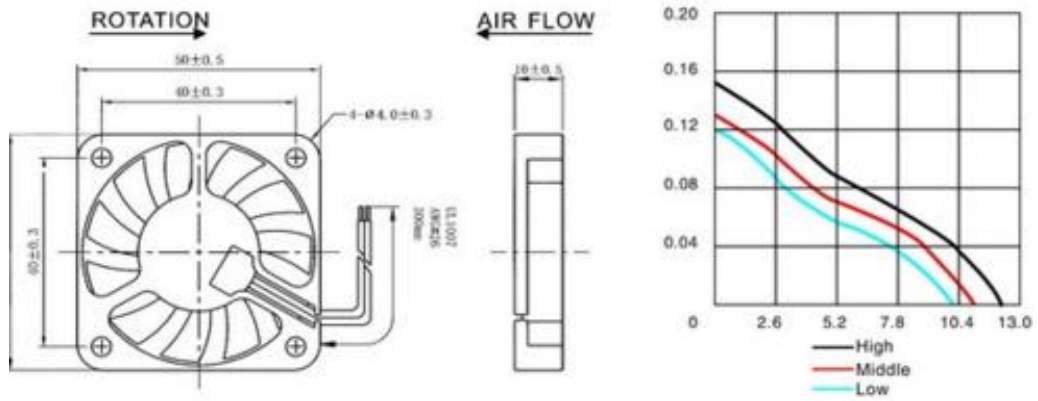


SPECIFICATIONS	XBee® 802.15.4	XBee-PRO® 802.15.4
PERFORMANCE		
RF DATA RATE	250 kbps	250 kbps
INDOR/URBAN RANGE	100 ft (30 m)	300 ft (100 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	300 ft (100 m)	1 mi (1.6 km)
TRANSMIT POWER	1 mW (+0 dBm)	60 mW (+18 dBm)*
RECEIVER SENSITIVITY (1% PER)	-92 dBm	-100 dBm
FEATURES		
SERIAL DATA INTERFACE	3.3V CMOS UART	
CONFIGURATION METHOD	API or AT Commands, local or over-the-air	
FREQUENCY BAND	2.4 GHz	
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)	
SERIAL DATA RATE	1200 bps - 250 kbps	
ADC INPUTS	(6) 10-bit ADC inputs	
DIGITAL I/O	8	
ANTENNA OPTIONS	Chip, Wire Whip, U.F.L, & RPSMA	
NETWORKING & SECURITY		
ENCRYPTION	128-bit AES	
RELIABLE PACKET DELIVERY	Retries/Acknowledgments	
IDS AND CHANNELS	PAN ID, 64-bit IEEE MAC, 16 Channels	
POWER REQUIREMENTS		
SUPPLY VOLTAGE	2.8 - 3.4VDC	2.8 - 3.4VDC
TRANSMIT CURRENT	45 mA @ 3.3VDC	215 mA @ 3.3VDC
RECEIVE CURRENT	50 mA @ 3.3VDC	55 mA @ 3.3VDC
POWER-DOWN CURRENT	<10 uA @ 25° C	
REGULATORY APPROVALS		
FCC (USA)	OUR-XBEE	OUR-XBEEPRO
IC (CANADA)	4214A-XBEE	4214A-XBEEPRO
ETSI (EUROPE)	Yes	Yes - Max TX 10 mW
C-TICK AUSTRALIA	Yes	
TELEC (JAPAN)	Yes	



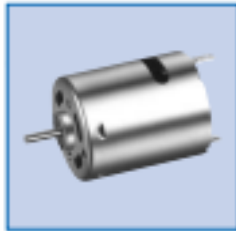
PART NUMBERS	DESCRIPTION
XKB2-AT-WWC	Wireless Connectivity Kit w/ XBee 802.15.4
XKB2-AT-WWG	XBee / Arduino Compatible Coding Platform
XB24-AWI-001	XBee 802.15.4 low-power module w/ wire antenna
XB24-API-001	XBee 802.15.4 low-power module w/ PCB antenna
XB24-AUI-001	XBee 802.15.4 low-power module w/ U.fl connector
XB24-ASI-001	XBee 802.15.4 low-power module w/ RPSMA connector
XBP24-AWI-001	XBee-PRO 802.15.4 extended-range module w/ wire antenna
XBP24-AWI-001J	XBee-PRO 802.15.4 extended-range module w/ wire antenna (International)
XBP24-AUI-001	XBee-PRO 802.15.4 extended-range module w/ U.fl connector
XBP24-AUI-001J	XBee-PRO 802.15.4 extended-range module w/ U.fl connector (International)
XBP24-ASI-001	XBee-PRO 802.15.4 extended-range module w/ RPSMA connector
XBP24-ASI-001J	XBee-PRO 802.15.4 extended-range module w/ RPSMA connector (International)
XBP24-API-001	XBee-PRO 802.15.4 extended-range module w/ PCB antenna
XBP24-API-001J	XBee-PRO 802.15.4 extended-range module w/ PCB antenna (International)

8.9. ANEXO 9.- HOJA DE DATOS VENTILADOR



Model (型号)	Rated Voltage (VDC) (额定电压)	Startup Voltage (VDC) (启动电压)	Current Amp (A) (电流)	Speed (rpm) (转速)	Air Flow (CFM) (风量)	Static Pressure (Inch-H2o) (风压)	Noise Level (dB-A) (噪音)	Weight (g) (重量)
YM0505PFS(B)1	5.00	3.00	0.21	4500	11.20	0.13	26	20.00
YM0505PFS(B)2	5.00	3.00	0.16	4000	10.08	0.12	23	20.00
YM0505PFS(B)3	5.00	3.00	0.12	3500	8.96	0.11	21	20.00
YM1205PFS(B)1	12.00	5.00	0.10	4500	11.20	0.13	26	20.00
YM1205PFS(B)2	12.00	5.00	0.08	4000	10.08	0.12	23	20.00
YM1205PFS(B)3	12.00	5.00	0.07	3500	8.96	0.11	21	20.00
YM2405PFS(B)1	24.00	10.00	0.08	4500	11.20	0.13	26	20.00
YM2405PFS(B)2	24.00	10.00	0.05	4000	10.08	0.12	23	20.00

8.10. ANEXO 10.- HOJA DE DATOS BOMBA



RS-360SH

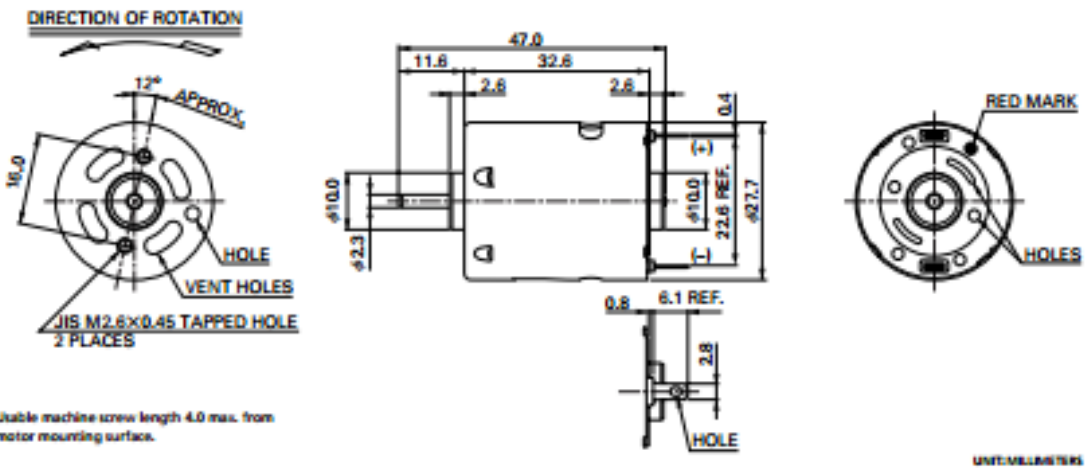
MABUCHI MOTOR
Carbon-brush motors

OUTPUT : 0.7W~40W (APPROX)

WEIGHT : 55g (APPROX)

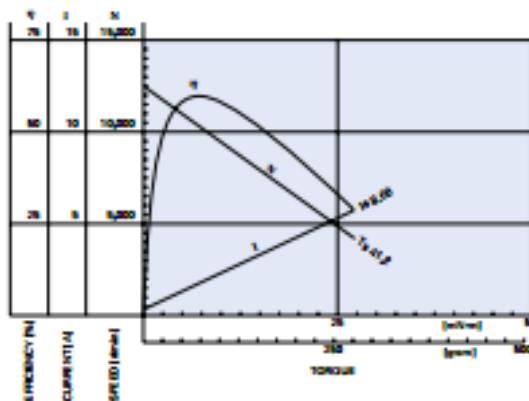
Typical Applications Home Appliances : Massager / Vibrator
Office Automation Equipment : Copy Machine / Laser Printer
Precision Instruments : Vending Machine

MODEL	VOLTAGE		NO LOAD		AT NOMINAL EFFICIENCY		AT MAXIMUM EFFICIENCY		STALL			
	OPERATING RANGE	NOMINAL	SPEED	CURRENT	SPEED	CURRENT	TORQUE	OUTPUT	TORQUE	CURRENT		
			rpm	A	rpm	A	kg-cm	W	kg-cm	kg-cm	A	
RS-360SH-2885	3-9	7.2V CONSTANT	12500	0.36	10380	1.76	7.00	71.3	7.50	41.2	420	8.60
RS-360SH-10500	12-25	12V CONSTANT	3500	0.050	2590	0.14	2.74	28.0	0.74	10.6	108	0.41



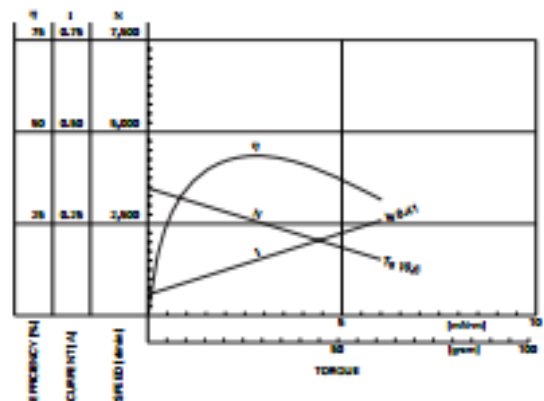
RS-360SH-2885

7.2V



RS-360SH-10500

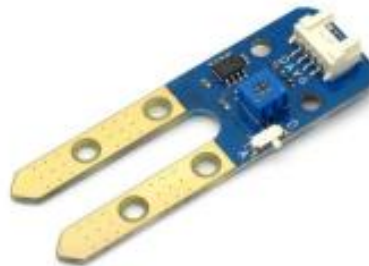
12.0V



8.12. ANEXO 12.- HOJA DE DATOS SENSOR SOIL

Soil Moisture Sensor

Overview



What is an electronic brick? An electronic brick is an electronic module which can be assembled like Lego bricks simply by plugging in and pulling out. Compared to traditional universal boards and circuit modules assembled with various electronic components, electronic brick has standardized interfaces, plug and play, simplifying construction of prototype circuit on one's own. There are many types of electronic bricks, and we provide more than twenty types with different functions including buttons, sensors, Bluetooth modules, etc, whose functions cover from sensor to motor drive, from Ethernet to wireless communication via Bluetooth, and so on. We will continue to add more types to meet the various needs of different projects.

Electronic brick of soil moisture sensor is mainly used to detect the moisture content in the soil. The control board can get the moisture value or threshold in the soil via analog or digital pins.

Specifications

PCB size	71.65mm X 24.00mm X 1.6mm
Working voltage	3.3or 5V DC
Operating voltage	3.3 or 5V DC
Compatible interfaces	2.54 3-pin interface and 4-pin Grove interface ^{[1][2]}

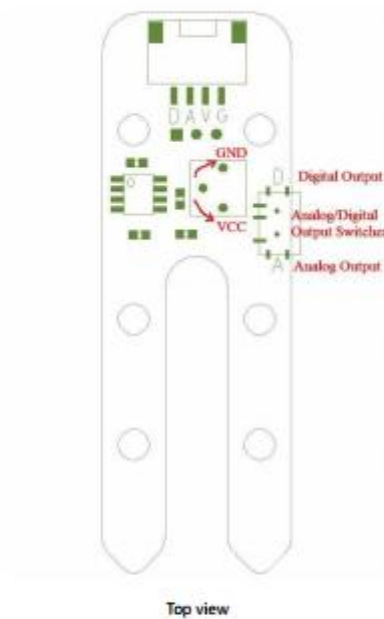
Note 1 : D for digital output port, A for analog output port, S for analog/digital output port (defined according to the switch), V and G for voltage at the common collector and ground respectively

Note 2 : When setting as analog output, output range is 0-3.3V or 0-5V according to the working voltage; when setting as digital output, output is 0/3.3V or 0/5V according to the working voltage.

Electrical characteristics

Parameter	Min.	Typical	Max.	Unit
Working voltage	2.1	5	5.5	VDC
Analog output voltage (VCC=5V)	0	Vout	5	V
Digital output voltage (VCC=5V)	0	-	5	V
Working current (VCC=5V)	-	5	-	mA
Threshold hysteresis ΔU_{th}	-	VCC*0.09	-	V

Hardware



Switch and indicator

1. Regulating of threshold voltage:

The threshold voltage is a voltage for comparison. When the soil moisture value read by the sensor is above the threshold value, a low level (0V) will be digitally output; when the soil moisture value read by the sensor is below the threshold value, a high level (3.3V or 5V) will be digitally output. In this way, the digital pin can be used directly to read the current soil moisture value to see if it is above the threshold or not. The threshold voltage can be regulated by simply twisting the potentiometer which is shown in Figure 1., and it increases by rotating to left side and decreases by rotating to right side.