



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA
ELECTRÓNICO PARA EL MONITOREO, CONTROL DE LA
HELADA Y RIEGO, EN HUERTOS DE FRESAS POR MEDIO DE
REDES DE SENSORES INALÁMBRICOS CON HARDWARE
LIBRE.

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO ELECTRÓNICO EN CONTROL Y REDES
INDUSTRIALES

AUTOR: CARLOS DANIEL ARCOS GUAMÁN
TUTOR: ING. HENRY VALLEJO

Riobamba – Ecuador
2017

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Titulación certifica que: El trabajo de investigación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA ELECTRÓNICO PARA EL MONITOREO, CONTROL DE LA HELADA Y RIEGO, EN HUERTOS DE FRESAS POR MEDIO DE RED DE SENSORES INALÁMBRICOS CON HARDWARE LIBRE, de responsabilidad del señor Carlos Daniel Arcos Guamán, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Gilberto Luna E. DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Freddy Chávez V. DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES	_____	_____
Ing. Henry Vallejo V. DIRECTOR DEL TRABAJO DE TITULACION	_____	_____
Ing. José Guerra S. MIEMBRO DEL TRIBUNAL	_____	_____

Yo, Carlos Daniel Arcos Guamán, declaro ser responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Carlos Daniel Arcos Guamán

DEDICATORIA

El presente trabajo de titulación quiero dedicarle a mis queridos padres, Carlos y Martha, ya que me apoyaron en toda la carrera, trabajaron día tras día, para darme lo necesario y que mis sueños se cumplan. A mis hermanos Santiago y Mercy, que saben el esfuerzo que he puesto en los estudios, gracias por el cariño que me brindan. A Mamá Rosita y Papá Segundo, quiero agradecerles el ejemplo que me dan, ya que así estoy cumpliendo mis metas. Mi abuelito Carlos que desde el cielo de seguro me cuida. A toda mi familia, tíos y primos, que estuvieron al tanto de mí, apoyándome de alguna u otra forma.

Además, agradezco a todos mis amigos y compañeros, que durante esta carrera tuve la oportunidad de conocer, estuvieron en los días buenos y malos, al final logramos conseguir el objetivo trazado desde los primeros semestres.

Daniel

AGRADECIMIENTO

Agradezco primero a Dios que me ha bendecido todos los días y me ha permitido cumplir mi sueño, a mi familia y todos mis compañeros y amigos, por el cariño y apoyo que han tenido durante todo este periodo de mi carrera.

De la misma manera, agradecer a mi segundo hogar la Escuela Superior Politécnica de Chimborazo, así también como a la Escuela de Ingeniería Electrónica en Control y Redes Industriales, junto con sus autoridades, por darme la oportunidad de culminar mis estudios profesionales

A cada uno de mis maestros por la paciencia que han tenido en instruirme ya que han sido una guía para obtener la mayor parte de mis conocimientos alcanzados en la carrera. Y sobre todo a mi docente tutor, el Ing. Henry Ernesto Vallejo, quien supo guiarme para cumplir este objetivo en mi vida, su apoyo desinteresado ha ayudado a que este trabajo de titulación lo haya terminado con éxito.

Daniel

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE ABRIVIATURAS.....	xiii
RESUMEN.....	xiv
SUMMARY.....	xv
INTRODUCCIÓN.....	1

CAPÍTULO I

1. MARCO TEÓRICO.....	3
1.1. Agricultura de precisión.....	3
1.2. Tecnologías de recolección de datos.....	3
1.2.1 <i>Arduino</i>	4
1.2.2 <i>Waspnote</i>	5
1.2.3 <i>Raspberry Pi</i>	5
1.2.4 <i>Intel Galileo</i>	6
1.2.5 <i>Comparación de plataformas</i>	7
1.3. Sensores.....	7
1.3.1 <i>Tecnología de transmisión inalámbrica para sensores (WSN)</i>	8
1.3.1.1 <i>Módulos de transmisión inalámbrica</i>	9
1.3.2 <i>Topología para sensores inalámbricos</i>	10
1.3.2.1 <i>Topología Estrella</i>	10
1.3.2.2 <i>Topología en Malla</i>	11
1.3.2.3 <i>Topología Malla-Estrella</i>	11
1.4. Cultivo de Fresas.....	12
1.4.1 <i>Características de cultivo de fresas</i>	12
1.5. Fenómeno de la helada.....	13
1.5.1 <i>Incidencia de la topología de un terreno en la helada</i>	14
1.5.2 <i>Protección contra la helada</i>	15
1.6. Trabajos realizados en Ecuador sobre supervisión de huertos de fresas.....	16

CAPITULO II

2. DISEÑO DEL HARDWARE Y SOFTWARE DEL SISTEMA.....	18
2.1. Requerimientos del sistema.....	18
2.1.1 <i>Estudio para determinar la variación de humedad en las camas de fresas</i>	18
2.2. Concepción del sistema.....	20

2.3	Diseño de la arquitectura del sistema	21
2.3.1	Módulos de Supervisión	21
2.3.1.1	<i>Módulo de Supervisión Ambiente</i>	22
2.3.1.2	<i>Módulo de supervisión de Suelo</i>	22
2.3.1.3	<i>Elementos que conforman el módulo de Supervisión</i>	22
2.3.1.4	<i>Diagrama de conexión Módulo Supervisión Ambiente</i>	24
2.3.1.5	<i>Diagrama de conexión Módulo de Supervisión de Suelo</i>	25
2.3.2.	Módulo de recepción	26
2.3.2.1	<i>Elementos que conforman el módulo de recepción</i>	28
2.3.2.2	<i>Diagrama de conexiones módulo de recepción</i>	30
2.3.3	Módulo de Riego	34
2.3.3.1	<i>Selección de elementos para el módulo de Riego</i>	35
2.4	Calibración del sensor FC-28 mediante un tensiómetro de Suelo	36
2.5	Diseño del software del sistema	37
2.5.1	Requerimientos del sistema	37
2.5.2	Programación Módulo de supervisión ambiente	38
2.5.2.1	<i>Bibliotecas para la programación</i>	40
2.5.3	Diagrama de flujo módulo ambiente	40
2.5.3.1	<i>Bibliotecas para la programación</i>	43
2.5.4	Programación del módulo de recepción	44
2.5.4.1	<i>Bibliotecas para el módulo de recepción</i>	46
2.5.4.2	<i>Funciones para la programación módulo de recepción</i>	47
2.6	Cálculo de la antena para los módulos de transmisión	48
2.7	Diseños físicos del sistema	49
CAPÍTULO III		
3.	ANÁLISIS Y RESULTADOS	51
3.1	Análisis de las variables ambientales del terreno para la instalación del sistema	51
3.2	Diseño e instalación de los aspersores y actuadores	52
3.2.1	<i>Instalación de aspersores</i>	53
3.2.2	<i>Instalación de electroválvulas y bomba eléctrica</i>	54
3.3	Instalación de los módulos de supervisión	55
3.3.1	<i>Módulo de Suelo</i>	55
3.3.1.1	<i>Ubicación de los sensores de humedad</i>	56
3.3.2	<i>Módulo supervisión ambiente</i>	57
3.3.3	<i>Ubicación de los paneles solares</i>	58
3.4	Pruebas de transmisión	59
3.5	Pruebas de los sensores	60

3.5.1	<i>Comparación de mediciones sensor de ambiente</i>	60
3.5.1.1	<i>Sensor de temperatura</i>	60
3.5.1.2	<i>Sensor de humedad ambiente</i>	61
3.5.2	<i>Pruebas de estabilidad de los sensores</i>	62
3.5.2.1	<i>Repetitividad del módulo de supervisión ambiente</i>	62
3.5.2.2	<i>Pruebas sensores de suelo</i>	63
3.5.3	<i>Pruebas de comprobación de helado</i>	64
3.5.4	<i>Pruebas de humedad de suelo</i>	65
3.6	Pruebas de funcionamiento módulo de recepción	66
3.6.1	<i>Almacenamiento de datos</i>	66
3.7	Menú de usuario LCD	67
3.7.1	<i>Igualar flecha</i>	68
3.7.2	<i>Visualización de los datos</i>	68
3.7.3	<i>Riego manual</i>	69
3.8	Análisis de costos de la implementación de Sistema	69
CONCLUSIONES		71
RECOMENDACIONES		72
BIBLIOGRAFÍA		
ANEXOS		

INDICE DE TABLAS

Tabla 1-1 Características plataformas de Hardware Libre.....	7
Tabla 2-1 Módulos de transmisión inalámbrica.....	10
Tabla 1-2 Mediciones de Humedad en el suelo.....	19
Tabla 2-2 Arquitectura del sistema.....	21
Tabla 3-2 Elementos electrónicos de módulo de supervisión.....	23
Tabla 4-2 Consumo de corriente cada módulo de supervisión.....	23
Tabla 5-2 Elementos del módulo de recepción.....	29
Tabla 6-2 Pines de conexión módulo RT.....	30
Tabla 7-2 Pines de conexión módulo SD Card.....	31
Tabla 8-2 Pines de conexión pantalla LCD.....	31
Tabla 9-2 Pines de conexión teclado matricial.....	32
Tabla 10-2 Pines de conexión módulo relés.....	33
Tabla 11-2 Calibración para sensor de humedad FC-28.....	37
Tabla 1-3 Mediciones de temperatura ambiente en el huerto.....	52
Tabla 2-3 Distancia de transmisión módulos RF.....	59
Tabla 3-3 Mediciones error absoluto sensor de temperatura ambiente.....	61
Tabla 4-3 Mediciones error absoluto sensor de humedad ambiente.....	62
Tabla 5-3 Prueba repetitividad módulo Supervisión Ambiente.....	63
Tabla 6-3 Prueba repetitividad módulo Supervisión de Suelo.....	64
Tabla 7-3 Costos del sistema electrónico.....	70

INDICE DE FIGURAS.

Figura 1-1: Tipos de tarjetas arduino.....	4
Figura 2-1: Tarjeta Waspnote.....	5
Figura 3-1: Placa Raspberry Pi 10.....	6
Figura 4-1: Placa Intel Galileo.....	6
Figura 5-1: Sensores de agricultura.....	8
Figura 6-1: Red de sensores.....	9
Figura 7-1: Topología en estrella.....	10
Figura 8-1: Topología en malla.....	11
Figura 9-1: Topología malla Estrella.....	12
Figura 10-1: Camas de fresa.....	13
Figura 11-1: Plantas después de la helada.....	14
Figura 12-1: Esguerrimiento del aire frío a través de una ladera.....	15
Figura 13-1: Riego por aspersión.....	16
Figura 1-2: Higrómetro de suelo analógico.....	19
Figura 2-2: Arquitectura del sistema.....	20
Figura 3-2: Topología del sistema.....	20
Figura 4-2: Diagrama de bloques módulo supervisión.....	22
Figura 5-2: Cargador solar.....	24
Figura 6-2: Conexiones módulo supervisión ambiente.....	25
Figura 7-2: Conexiones módulo supervisión de sueocia.....	26
Figura 8-2: Diagrama de bloque módulo de recepción.....	27
Figura 9-2: Diagrama de bloques del bloque de potencia.....	27
Figura 10-2: Conexiones módulo de recepción.....	32
Figura 11-2: Conexiones bloque de potencia.....	35
Figura 12-2: Diagrama de bloques módulo de riego.....	35
Figura 13-2: Aspersores antihelada.....	35
Figura 14-2: Tensiómetro de suelo A.M.I GAUGE.....	36
Figura 15-2: Calibración de los valores de húmeda.....	36
Figura 16-2: Diagrama de flujo módulo de supervisión Ambiente.....	39
Figura 17-2: Trama de datos módulo de supervisión ambiente.....	40
Figura 18-2: Diagrama de flujo módulo de Supervisión de Suelo.....	42
Figura 19-2: Trama de datos módulo de Supervisión de Suelo.....	43
Figura 20-2: Diagrama de flujo bloque de almacenamiento y visualización.....	45
Figura 21-2: Diagrama de flujo bloque de Potencia.....	46
Figura 22-2: Longitud antena para módulos.....	49

Figura 23-2: Diseño físico módulos de Supervisión.....	49
Figura 24-2: Diseño físico Módulo de Recepción.....	50
Figura 1-3: Ubicación de termómetros en el terreno.....	51
Figura 2-3: Ubicación de aspersores.....	53
Figura 3-3: Instalación de aspersores.....	53
Figura 4-3: Instalación de electroválvulas.....	54
Figura 5-3: Bomba eléctrica de agua.....	54
Figura 6-3: Ubicación de los Módulos de Supervisión.....	55
Figura 7-3: Ubicación de los Módulos de Suelo.....	56
Figura 8-3: Ubicación sensores de humedad.....	56
Figura 9-2: Instalación de sensores en las camas de fresa.....	57
Figura 10-3: Instalación módulo de Supervisión Ambiente 1.....	57
Figura 11-3: Instalación módulo de Supervisión Ambiente 2.....	58
Figura 12-3: Ángulo de inclinación para paneles solares.....	58
Figura 13-3: Instalación paneles solares.....	59
Figura 14-3: Termómetro-Higrómetro.....	60
Figura 15-3: Activación de riego por aspersión.....	65
Figura 16-3: Activación de riego por goteo.....	65
Figura 17-3: Módulo de supervisión.....	66
Figura 18-3: Archivos de texto Módulos de Supervisión.....	66
Figura 19-3: Archivos de texto Módulos de Supervisión.....	67
Figura 20-3: Menú de usuario.....	68
Figura 21-3: Insertar fecha y hora.....	68
Figura 22-3: Visualización de datos.....	69
Figura 23-3: Control manual de riegos.....	69

ÍNDICE DE ABREVIATURAS

WSN	Wireless Sensor Network (Red Inalámbrica de Sensores)
RF	Radio Frecuencia
SD	Secure Digital (Seguro Digital)
GSM	sistema global para las comunicaciones móviles
E/S	Entradas y Salidas
RAM	<i>Random access memory</i>
AC	Alternating Current (Corriente Alterna)
DC	Direct Current (Corriente Directa) ´
EEPROM	Electrically Erasable Programmable Read-Only Memory (ROM Programable y Borrable Eléctricamente)
SRAM	Static Random Access Memory (Memoria Estática de Acceso Aleatorio)
I2C	Inter-Integrated Circuit (I-Cuadrado-C)
MISO	Master Input Slave Output (Salida de datos del Slave y entrada al Master)
MOSI	Master Output Slave Input (Salida de datos del Master y entrada de datos al Slave)
SPI	Serial Peripheral Interface (Interfaz Periférica Serial)
RTC	Real Time Clock (Reloj de Tiempo Real)
LCD	Liquid Crystal Display
SDA	Data Line
SCL	Clock line
TX	Transmisión
RX	Recepción

RESUMEN

En el presente trabajo de titulación, se diseñó e implementó un sistema electrónico para la supervisión, control de la helada y riego en huertos de fresas por medio de una red inalámbrica con sensores inteligentes. Cada uno de los elementos electrónicos se ha seleccionado detenidamente, con el fin de construir un sistema de bajo costo y consumo. Consta de un módulo de supervisión, uno de recepción, y un módulo de riego. El primer módulo se encarga de supervisar la temperatura, humedad relativa y las variaciones de la humedad del suelo, esos datos son enviados por medio de radio frecuencia hasta una distancia de 65m con una antena de 17cm de longitud, al módulo de recepción el cual contiene un bloque de interfaz de usuario, almacenamiento y potencia, este último bloque activa los actuadores: bomba de agua y electroválvulas, para el funcionamiento del módulo de riego, el cual se divide en un sistema de riego por goteo, que se activa de acuerdo a los rangos de humedad de suelo que requieren las plantas de fresa y un sistema de riego por aspersión, que actúa si el módulo de supervisión está detectando una helada en el huerto. En las pruebas realizadas los sensores inteligentes instalados constaron de coeficientes de variación menores al 1%, con errores absolutos de $\pm 4\%$ en humedad y $\pm 1.2^{\circ}\text{C}$, midiendo valores negativos con respecto a la temperatura, comprobando que cumplen con los requerimientos del sistema. Se recomienda que los datos de las variables obtenidas sean compartidos en la web con el fin que especialistas analicen el rendimiento de los cultivos de fresas según parámetros ambientales.

Palabras Claves: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <TECNOLOGÍA DEL CONTROL AUTOMÁTICO>, <AGRICULTURA DE PRECISIÓN>. <SUPERVISIÓN>, <RED DE SENSORES INALÁMBRICOS>, <HUMEDAD DE SUELO>, <VARIABLES AMBIENTALES>, <PUNTO DE ROCÍO>.

SUMMARY

In this titling work, an electronic system was designed and implement for supervision, frost control and irrigation in strawberry orchards by means of wireless network with intelligent sensors, each of the electronic elements has been carefully selected in order to build a system of low cost and consumption. It consists of a module of supervision, one of reception and modulo of risk. The first is responsible for monitoring the temperature, relative humidity and variations in soil moisture; These data are send by radio frequency up to a distance of 60m with an antenna of 17cm in length to the reception module containing a block of use interface , storage and power; the latter activates the actuators; water pump and electro-valves, for the irrigation module operation divided into a drip irrigation system activated according to the soil moisture ranges required by the strawberry plants and sprinkler irrigation system that acts if the supervision module is detecting a frost in the orchard. In the tests performed, the intelligent sensors consisted of coefficients of variation less than 1%, with absolute errors of $\pm 4\%$ in humidity and $\pm 1^\circ\text{C}$, measuring negative values with respect to temperature, proving that they comply with the system requirements. It is recommended that the data of the variables obtain be shared on the web in order for specialists to analyze the performance of strawberry crops according to environmental parameters.

Keywords: <TECHNOLOGY AND ENGINEERING SCIENCE>, <TECHNOLOGY OF THE AUTOMATIC CONTROL>, <PRECISION FARMING>. <SUPERVISION>, <WIRELESS SENSORS NETWORK>, <SOIL MOISTURE>, <ENVIROMENTAL VARIABLES>, <DEW POINTS>.

INTRODUCCIÓN

Desde el año de 1992, la Agricultura de Precisión ha llamado mucho la atención de la Producción a gran escala. Actualmente la inversión en la Agricultura de Precisión es prometedora especialmente en sectores de Norte América, para este campo el uso de tecnologías en economías en desarrollo es extensamente discutido y limitado. (García, 2012).

Los avances tecnológicos permiten automatizar los procesos agrícolas, debido a un sin número de opciones y servicios que se ofertan. En los países desarrollados está extendido el uso de soluciones tecnológicas para lograr una agricultura sostenible que lleve a la máxima eficiencia en el uso de recursos naturales y fertilizantes químicos. (Nates, 2013).

En el Ecuador la mayoría de huertos especialmente de fresas, son cultivados de forma artesanal, no cuentan con tecnologías que permitan supervisar variables ambientales y químicas de forma automatizada. Para el manejo del cultivo los agricultores lo realizan mediante hábitos que han adquirido durante los años, es por esto que les resulta difícil saber con exactitud cuándo se necesitan los recursos exactos para sus cultivos, antes que se produzca pérdidas económicas por mal manejo.

La necesidad de crear un sistema de supervisión y actuación sobre el control de riego y especialmente para el control de helada es de suma importancia, ya que, según datos en el año 2012, cerca del 40% de todo tipo de cultivos como: Papa, maíz, haba, mora, fresa en la Región Sierra sufrieron pérdidas de producción por estas variaciones del clima, que pasa de bajas temperaturas en las noches a altas durante el día. (Orozco, 2012).

Por tal razón el presente trabajo de investigación, se diseñó e implementó un sistema electrónico que permite controlar el riego por goteo del huerto y los daños producidos por la helada, para lograrlo se procedió a la investigación de: Rangos de humedad y temperatura que los huertos de fresa necesitan para su desarrollo, los diversos sensores para supervisar las variables ambientales, dispositivos de comunicación y tarjetas de desarrollo que trabajen para el correcto procesamiento de los datos a supervisar.

OBJETIVOS

- Investigar cuales son los rangos de temperatura y humedad en la que se produce la helada en los huertos de fresas
- Seleccionar los sensores y dispositivos electrónicos adecuados que se ajusten a los requerimientos del sistema.
- Determinar el tipo de tecnología inalámbrica adecuada que ofrezca las características necesarias para el sistema de supervisión.
- Diseñar la arquitectura del sistema apropiada para la supervisión con sensores inalámbricos.
- Diseñar los módulos electrónicos de monitoreo para el control de la helada y el riego por goteo en el huerto.
- Implementar los módulos y validar la fiabilidad de los datos obtenidos que formaran parte el sistema electrónico.

Para lograr con los objetivos planteados el documento consta de una metodología analítica con actividades como: Investigación Teórica, Diseño e Implementación del *Hardware y Software*, Análisis y resultados del sistema implementado. El proyecto se lo realizó con el fin de automatizar procesos de bajo costo en los huertos de fresa.

CAPÍTULO I

1. MARCO TEÓRICO

En este capítulo se analiza sobre las características de la agricultura de precisión, tecnologías existentes para la comunicación inalámbrica, y plataformas de desarrollo para el diseño del sistema electrónico. Así como conceptos necesarios sobre las fresas y la helada que afecta a estos cultivos.

1.1 La agricultura de precisión.

Con la aparición de nuevas tecnologías, como controladores lógicos, microprocesadores, la agricultura se maneja a una gran escala, creando sistemas de control que permitan ahorrar tiempo, reducir esfuerzo físico, tener un mejor uso de los recursos naturales y químicos que utilizan los agricultores. La construcción de maquinarias automatizadas, el diseño de sistemas de control en cultivos, permite al agricultor actualizarse con las tecnologías agrícolas, reduciendo las largas y agotadas jornadas laborales.

La agricultura de precisión en países desarrollados es una actividad de alta tecnología, sobre todo con avances importantes en la última década en el manejo agrícola automatizado, debido a esto se genera ventajas como: Aumento de la productividad, reducción de insumos, uso apropiado del agua, Simplicidad y confort. Una desventaja dentro de esta área es: El costo de la implementación de las tecnologías sobre todo en zonas de producción a baja escala. (Castellanos, 2013).

1.2 Tecnologías de recolección de datos

Las plataformas de *hardware* libre, son tarjetas electrónicas de desarrollo diseñadas con circuitos impresos que están compuestos por un microcontrolador y para su programación contienen un entorno de desarrollo. Al ser plataformas libres, las especificaciones y los diagramas que lo componen son de acceso público, de tal manera que se lo puede utilizar bajo algún tipo de pago como colaboración a la compañía que lo diseña, o también usarlo de forma gratuita.

Cada plataforma está diseñada de acuerdo a las necesidades que tenga el usuario, ofreciendo conexiones con varios dispositivos tales como, módulos electrónicos de diversas aplicaciones, sensores, tarjetas de almacenamiento, actuadores, interfaces de comunicación, entre otros, constando de E/S (entradas y salidas) tanto analógicas como digitales, de acuerdo al área donde se vaya a desarrollar.

Existen varias tarjetas de desarrollo en el mercado, en diferentes versiones que están a disposición del público, como: Arduino, Waspote, Galileo, Raspberry Pi, entre otros.

1.2.1 *Arduino*

Es una plataforma de hardware libre, basada en una placa con un microcontrolador con entorno de desarrollo IDE muy sencillo encargado de realizar los procesos lógicos y matemáticos dentro de la placa. El software se puede descargar gratuitamente, y los diseños de referencia del hardware están disponibles para los usuarios. (Garemar, 2016).

Presenta un gran número de modelos y versiones de sus placas, diferenciándolos en el número de Entradas y Salidas (E/S), tamaño, precio, memoria entre otras características, como se puede observar en la figura 1.1, las placas Arduino más reconocidas y utilizadas son: Uno, Mega, Pro-mini, Nano.



Figura 1-1: Tipos de tarjetas Arduino
Fuente: INGENIO TRIANA, 2016

1.2.2 *Waspote*

Es una plataforma modular opensource para construir redes de sensores inalámbricas de bajo consumo, fue creada por la empresa española Libelium. Esta placa consta de un microcontrolador, memoria, batería, acelerómetro y sockets para añadir módulos, como se aprecia en la figura 2-1. Usa el mismo entorno de desarrollo que Arduino, el mismo código es compatible en ambas plataformas, sólo ajustando pequeñas cosas como la asignación de pines y el esquema de E/S. (García, 2012). La plataforma es aceptada a nivel industrial, pero una de las desventajas es su costo elevado.



Figura 2-1 Tarjeta Waspote
Fuente: GARCÍA, Luis M. 2012

1.2.3 *Raspberry Pi*

Se trata de una placa base del tamaño de una tarjeta de crédito, desarrollada en Reino Unido con fines informáticos, en el que se aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad. Debido a su potencial, son muy apetecidas. Es una placa muy avanzada, que lo convierten en un ordenador, constan de entradas y salidas analógicas para utilizarlos con varios actuadores y módulos electrónicos. Tiene algunas versiones, en la figura 3-1 se observa las tarjetas disponibles de Raspberry. (Castro, 2014).

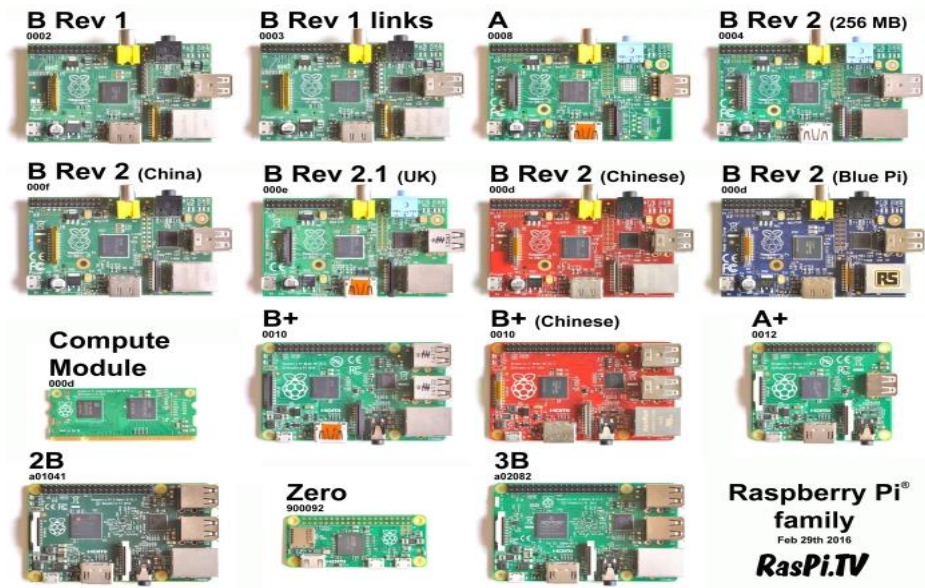


Figura 3-1: Placas Raspberry Pi
 Fuente: EAMES, Alex. 2016

1.2.4 Intel Galileo

Fue creada gracias a la colaboración de Intel y Arduino, con el objetivo de ayudar al desarrollo de sistemas o proyectos electrónicos, bajo la arquitectura Intel, es capaz de funcionar con sistema operativo Linux. Contiene un chip X86, a nivel *hardware* está basado en un sistema de bajo consumo. Se puede programar bajo diferentes plataformas como: Linux, Mac OS X y Windows, tiene las posibilidades de conectarse con internet. La placa se puede apreciar en la figura 4-1. Su precio no es muy accesible como para aplicaciones de bajo costo. (Velasco, 2013).



Figura 4-1: Placa Intel Galileo
 Fuente: VELASCO, J. 2013

1.2.5 Comparación de plataformas

En la siguiente tabla 1-1 se presenta la comparación entre las plataformas de *hardware* libre. Cada placa tiene características buenas de acuerdo al proyecto a implementar, pero Arduino es llamativo en el mercado por su precio y variedad de modelos.

Tabla 1-1: Características Plataformas de Hardware libre

Características	Arduino	Waspnote	Raspberry (Model B)	Intel Galileo
Procesador	<i>ATmega2560-168-328</i>	<i>ATMega1281</i>	<i>ARM11</i>	<i>SoC Quark X100</i>
Voltaje de entrada	<i>5-12 V</i>	<i>5 – 7 V</i>	<i>5 V</i>	<i>5 V</i>
Voltaje de Operación	<i>5V</i>	<i>0.5 - 3.8V</i>	<i>3.3 – 5V</i>	<i>3.3 V – 5 V</i>
Precio estimado (\$)	<i>20</i>	<i>180</i>	<i>70</i>	<i>90</i>
Ventajas	<i>Varios modelos a disposición. Bajo costo y fácil manejo</i>	<i>Cantidad de sensores a disposición para esta tarjeta</i>	<i>Muy avanzada. Rápida al realizar sus procesos</i>	<i>Compatible con Arduino en sus características</i>
Desventaja	<i>Su trabajo en área industrial es limitada</i>	<i>Precio elevado en el mercado</i>	<i>Usado para proyectos que requieran alto potencial</i>	<i>Alto costo. Por su similitud la mayoría prefiere Arduino</i>

Realizado por: ARCOS, Daniel. 2017

1.3 Sensores

Un sensor trata de un dispositivo que recibe señales físicas de un medio y lo convierte a señales eléctricas. Las señales que puede captar un sensor pueden ser: luz, calor, movimiento, humedad, temperatura, radiación, presión, peso, etc... La detección que realizan los sensores se da debido a las siguientes características:

- *Rango de medida:* Tipo de dominio en la magnitud que mida el sensor.
- *Precisión:* Exactitud de los datos recibidos en el sensor.
- *Resolución:* Variación de la medida con la que se reciben los datos.
- *Tiempo de Respuesta:* Varía dependiendo de la variación de magnitud a medir.

Estos elementos son usados en varias aplicaciones para supervisar las variables del área en donde se lo instalan. Con respecto a la agricultura existen un sin número de sensores para controlar diferentes aspectos como: Humedad (en ambiente, en suelo a distintas profundidades),

Temperatura, Luminosidad, radiación solar, Contaminación, Crecimiento de tallo, Diámetro de fruto, Diámetro de tronco, etc., En la figura 5-1 se muestra alguno de ellos. (Vermon).



Figura 5-1: Sensores para la agricultura
Realizado por: ARCOS, Daniel. 2017

1.3.1 Tecnología de transmisión inalámbrica para sensores (WSN)

WSN (Por sus siglas en inglés *Wireless Sensor Network*) consisten en una red de varios dispositivos distribuidos físicamente, denominados nodos de sensores, que se lo colocan en un área para que esta sea supervisada, con el fin de almacenar o estudiar los datos que se obtienen. (Fernández, 2009). La WSN se constituye de:

- *Evento:* Son las señales a medir por los sensores.
- *Sensores:* Dispositivo que capta la información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, para poder cuantificar y manipular.
- *Nodo del sensor:* Son los encargados reunir la información para enviar a la estación base, o comunicarse entre los otros nodos, según sea el tipo de red conectada.
- *Red Inalámbrica:* Comunicación entre dos o más terminales sin la necesidad de utilizar cables.
- *Gateway:* Elementos para la interconexión entre la red de sensores y una red TCP/IP.
- *Estación Base:* Recolector de datos.

En la figura 6-1 se puede observar las partes que compone la red de sensores

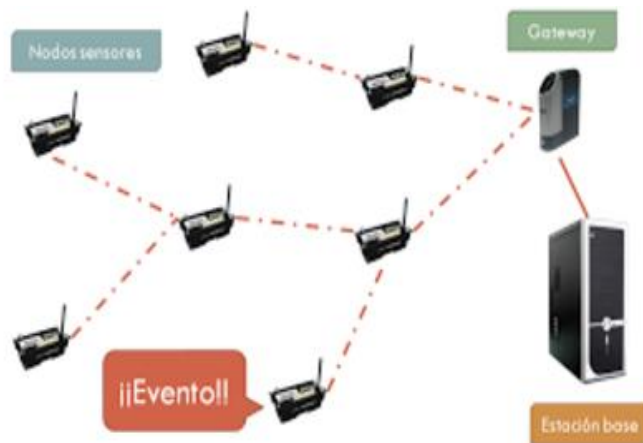


Figura 6-1: Red de sensores
Fuente: CIANCA, Kiara. 2012

1.3.1.1 Módulos de transmisión inalámbricas

En el mercado se encuentra una variedad de módulos de transmisión inalámbrica para formar una red de sensores, con diferentes características técnicas como alimentación, frecuencia de trabajo, distancia de transmisión, etc... En la tabla 2-1 se observa las tecnologías más comunes que se puede utilizar en la transmisión inalámbrica de datos

Tabla 2-1: Módulos de transmisión inalámbrica

TECNOLOGÍA	Rango	Velocidad de datos	Frecuencia de operación	Topología de Red	Precio (\$)
Zigbee	10-100m	20-250Kbps	868-902-928Mhz, 2.4Ghz	Árbol, Estrella, Malla	40-80
Radio Frecuencia	10-230m	10 Kbps	433.92Mhz	Simplex, Estrella	4-8
Wifi	30-100m	54Mbps	2.4 Ghz	Infraestructura, AD-HOC, Mess	15-30
GSM/GPRS	30 Km	114 Kbps	800-1800-1900 Mhz	Punto-Punto, Punto-Multipunto	40-50

Realizado por: ARCOS, Daniel. 2017

Al comparar las tecnologías de transmisión existen muchas diferencias con respecto a precios y a distancia de transmisión. Los módulos de radio frecuencia son los más económicos y la información se trasmite a grandes distancias, es una tecnología apropiada para sistemas de bajo costo.

1.3.2 Topología para sensores inalámbricos

Una topología de red se refiere a como se encuentran conectados los componentes *hardware*, para la transmisión de los datos. En una red inalámbrica para sensores existe varias arquitecturas: estrella, malla o estrella-malla, que contienen diferentes características, ventajas y desventajas. (Fernández, 2009).

1.3.2.1 Topología Estrella

Es un sistema en donde la información enviada por los nodos va directamente a la puerta de enlace o Gateway, ya que entre ellos no se pueden comunicar, en la imagen 7-1 se aprecia cómo se compone. Una de las principales ventajas de esta red es que si al dañarse algún nodo, esta va a seguir funcionando. Y la desventaja es que, si un nodo pierde la comunicación, no tiene otro camino para la transmisión de datos. (Fernández, 2009)

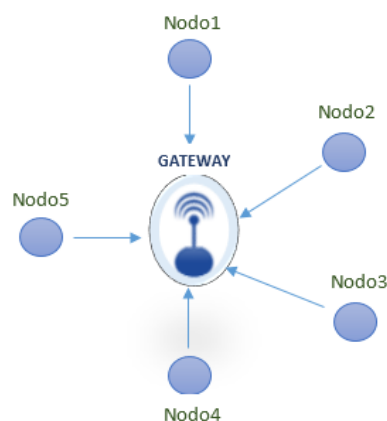


Figura 7-1: Topología en estrella
Realizado por: ARCOS, Daniel 2017

1.3.2.2 Topología en Malla

Es un sistema multipunto, donde todos los nodos son routers. Cada nodo puede enviar y recibir información de otro nodo y de la puerta de enlace, como se observa en la figura 8-1. A diferencia de la topología en estrella donde los nodos solo pueden comunicarse con la puerta de enlace, en esta los nodos pueden enviarse mensajes entre ellos.

La propagación de los datos a través de los nodos hacia la puerta de enlace hace posible crear una red con extensión posible ilimitada. Este tipo de red también es altamente tolerante a fallos ya que cada nodo tiene diferentes caminos para comunicarse con la puerta de enlace. (Fernández, 2009).



Figura 8-1: Topología malla
Realizado por: ARCOS, Daniel. 2017

1.3.2.3 Topología Malla – Estrella

Es una topología híbrida porque une dos topologías, lo que busca es cubrir una gran extensión de área y la información que se transmite en los nodos no se pierda, se convierte en una topología completa en todos los sentidos, la figura 9-1 muestra su estructura. (Fernández, 2009).

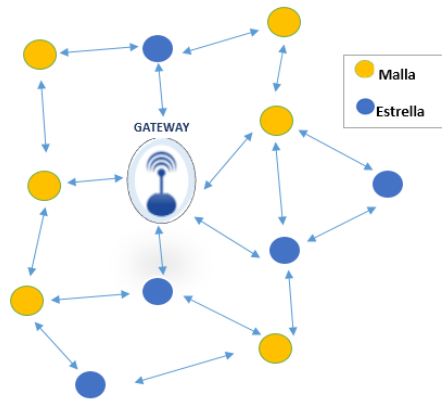


Figura 9-1: Topología malla Estrella
Realizado por: ARCOS, Daniel. 2017

1.4 Cultivo de Fresas

Estas frutas son apetecidas en el mundo, tiene un sin número de variedades, diferenciándose en la calidad, textura y cantidad de producción. Se caracterizan por adaptarse a cualquier clima, pero se desarrollan mejor en los climas fríos y templados. (Baudilio, 1977, p 24). Este cultivo en Ecuador ha crecido notablemente, pasando a ser una de las alternativas de cultivo para los agricultores, especialmente en la zona centro del país (El Agro, 2016).

1.4.1 Características de cultivo de fresas

Las plantas son sembradas en camas de tierra, por lo que facilita el escurrimiento del agua de las lluvias del centro hacia los costados. Estas camas son cubiertas por plásticos negros como se aprecia en la figura 10-1, que ayudan a generar un microclima en todo el huerto y con el riego a goteo se distribuye equitativamente el agua a cada una de las raíces de las plantas que crecen a un diámetro radicular de 30cm, de esta forma se disminuye el desperdicio del recurso hídrico. (Anexo 3)



Figura 10-1: Camas de fresa
Realizado por: ARCOS, Daniel. 2017

Rango de humedad

Las fresas son plantas que necesitan ser regadas abundantemente, por lo que se realiza riegos frecuentes. El rango de humedad recomendado esta entre los 25 y 35 centibares. (Morín, 1980, p 188).

1.5 Fenómeno de la helada.

Es uno de los fenómenos climáticos más peligrosos que afectan a la agricultura, ya que provoca cambios bruscos en el ambiente, generando al agricultor pérdidas económicas. Este fenómeno se produce durante la noche cuando: La temperatura ambiente llega a 0°C o menos, la humedad relativa desciende hasta que el punto de rocío está a menos de 0°C, esto causa que el vapor de agua tanto del aire y el de las plantas llegue al punto de congelación quemando los tejidos vegetales como se observa en la figura 11-1. (Snyder & Melo, 2010)



Figura 11-1: Plantas después de la helada
Fuente: Shisterova, Irina. 2016

Punto de Rocío

Es una variable ambiental que se calcula junto con la temperatura y humedad relativa. El punto de rocío se mide en Grados Celsius (°C), es la temperatura a la que empieza a condensarse el vapor de agua contenido en el aire.

Calculo del punto de rocío

Depende de las variables ambientales que son la humedad y temperatura.

$$Pr = \sqrt[8]{\frac{H}{100} \times [112 + (0.9 \times T)] + (0.1 \times T) - 112}$$

Donde:

Pr=Punto de Rocío

H=Humedad relativa

T=Temperatura

1.5.1 Incidencia de la topología de un terreno en una helada

Influye mucho en las bajas temperaturas, si los terrenos son planos, la temperatura es equitativa en toda el área, pero si existe suelos inclinados se intensifica la helada, esto se debe a que el aire frío es atrapado en las partes bajas de los valles, ya que al ser más denso que el aire tibio tiende a bajar, por lo tanto, el fenómeno es más intenso, como se observa en la figura 12-1. Además, hay

que tomar en cuenta si las plantaciones se encuentran en lugares recubiertos por árboles, estos sirven como cortinas para tapar el viento helado que daña las plantaciones (Rojas, 2005).

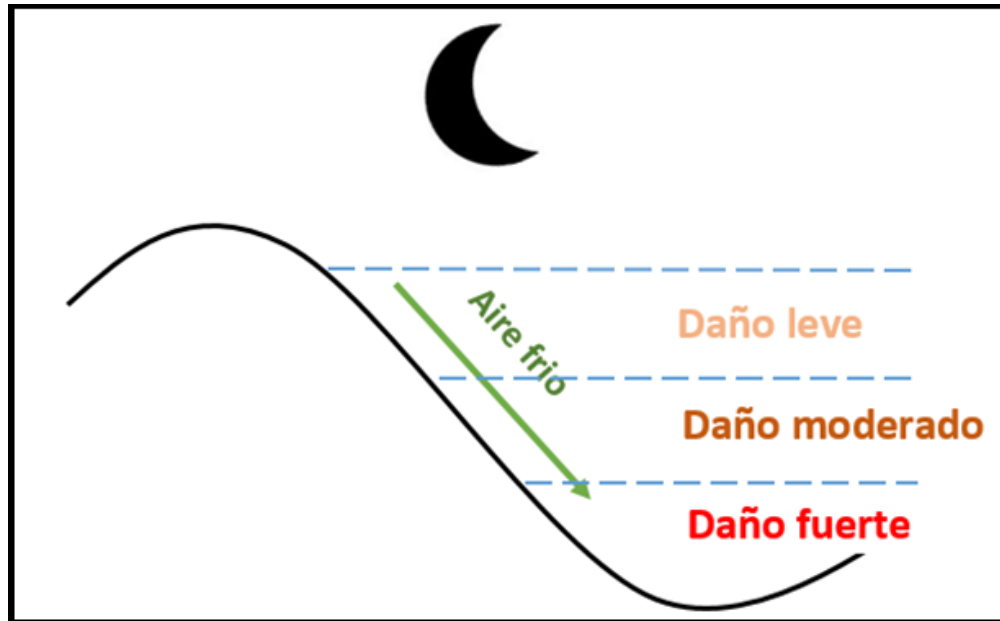


Figura 12-1: Escurrimiento del aire frío a través de una ladera
Fuente: ROJAS MONTECINO, Patricio. 2005

1.5.2 *Protección contra la helada.*

Es posible utilizar métodos para aislar a la planta durante las heladas, para esto existen varias formas para proteger a las plantaciones como la utilización de:

- Estufas
- Grandes ventiladores
- Los helicópteros
- Los aspersores
- El riego de superficie
- El aislamiento con espumas

Todos los métodos y combinaciones se implementan durante la noche de ocurrencia de heladas para mitigar los efectos de las temperaturas bajo cero.

La utilización de aspersores para la protección de helada tiene más ventaja ya que la aplicación de agua es en general menos cara. El consumo de energía es un 10% de la utilizada en la protección contra las heladas con estufas y por ello, los costes de funcionamiento son bajos si se compara con las estufas e incluso con los ventiladores. (Snyder & Melo, 2010, p.178). En la figura 13-1 se observa el sistema por aspersión en huertos de fresa.



Figura 13-1: Riego por aspersión
Realizado por: ARCOS, 2017

Para la activación de estos riegos se debe instalar termómetros o sensores en los huertos a una altura entre 1.25 y 2m desde el suelo en donde se encuentra la planta, para detectar si existe helada. (Snyder & Melo, 2010, p.144).

1.6 Trabajos realizados en Ecuador sobre supervisión de huertos de fresa.

En los últimos años los cultivos de fresas han aumentado considerablemente, es por eso la necesidad de desarrollar sistemas que permitan la automatización de diferentes procesos y actividades en estos huertos, pero pocos son los trabajos que se han realizado en la automatización y control para estas plantaciones.

Entre los trabajos realizados se puede mencionar: Un sistema automático de supervisión para riego por goteo para cultivo de fresas artesanales, en la provincia de Imbabura, para lo cual los desarrolladores utilizaron PLC's como el controlador principal, sistema que resultó costoso para realizarse. (Muñoz, 2014, p.15).

Otros de los trabajos referentes al proyecto es la supervisión de huertos de fresa para la protección contra heladas, en la Escuela Superior Politécnica del Ejército, se diseñó un sistema anti-helada mediante riego por aspersión, con microcontroladores, (Santos & Zarabia, 2008). Uno de los puntos que no se tomaron para este trabajo fue, la incidencia de la topología del terreno para el diseño del sistema anti-helada.

Es por esto que el presente trabajo se analizará estos dos puntos para tener un mejor sistema de supervisión.

CAPÍTULO II:

2. DISEÑO DEL *HARDWARE Y SOFTWARE* SISTEMA.

En este capítulo se muestra el diseño del hardware y el software para el sistema electrónico, la arquitectura utilizada, la selección de los elementos y como trabajan cada uno de los módulos de supervisión. Entre los diferentes elementos que consta el sistema están los sensores, los cuales supervisan las variables ambientales, la plataforma de hardware libre utilizado en el diseño, y la comunicación inalámbrica por el cual se envía los datos.

2.1 Requerimientos del sistema

Luego del diseño de la arquitectura, en esta sección se detalla los requerimientos que se necesita para cada bloque, y los módulos que lo conforman, eligiendo los elementos necesarios para el diseño del prototipo.

- Diseñar un módulo que supervise las variables ambientales.
- Diseñar de un módulo para la supervisión de la humedad del suelo.
- Crear un sistema electrónico de bajo costo.
- Utilizar sensor que mida valores negativos para captar la helada.
- Visualización de los valores medidos en tiempo real.
- Activación de actuadores, cuando las variables medidas estén fuera de los rangos permitidos,

2.1.1 *Estudio para determinar la variación de humedad en las camas de fresas.*

En el capítulo anterior se habla que la humedad del suelo no varía en las camas de fresa, ya que se genera un microclima en los acolchados. Para validar esta teoría se realizó las mediciones en el huerto de fresas ubicado en la Provincia de Tungurahua Catón Tisaleo, caserío San Francisco, en donde se realizará el sistema de prueba. El terreno consta de un área de 785 m² las mediciones se lo realizó en las 54 camas de fresa después del riego, con un tensiómetro de humedad de suelo como se puede apreciar en la figura 1-2.



Figura 1-2: Higrómetro de suelo analógico
 Realizado por: ARCOS, Daniel. 2017

Los valores obtenidos se muestran en la tabla 1-2, el máximo valor de humedad de suelo es de 87%, y el menor es 84%. De acuerdo a los datos obtenidos, existe una diferencia no muy relevante de 3% en humedad entre las camas, por lo que se puede ubicar un solo módulo de control de humedad que supervise todo el huerto para la activación del riego por goteo.

Tabla 1-2: Mediciones de Humedad en el suelo.

# CAMA	Humedad (%)	# CAMA	Humedad (%)	# CAMA	Humedad (%)
1	86	19	86	37	87
2	87	20	87	38	87
3	87	21	85	39	85
4	86	22	87	40	86
5	87	23	85	41	87
6	87	24	84	42	85
7	85	25	87	43	87
8	85	26	86	44	86
9	84	27	87	45	87
10	86	28	86	46	86
11	85	29	87	47	84
12	87	30	86	48	86
13	85	31	86	49	86
14	84	32	85	50	85
15	85	33	86	51	85
16	87	34	85	52	86
17	87	35	86	53	85
18	87	36	85	54	86

Realizado por: ARCOS, Daniel. 2017

2.2 Concepción del sistema

Para el diseño del sistema, se utiliza la siguiente arquitectura que se muestra en la figura 2-2, en donde se puede visualizar los módulos que se instalarán en el huerto, y los diferentes bloques que lo conforman, utilizando una comunicación inalámbrica y todas las tecnologías necesarias para el control de nuestro sistema electrónico.

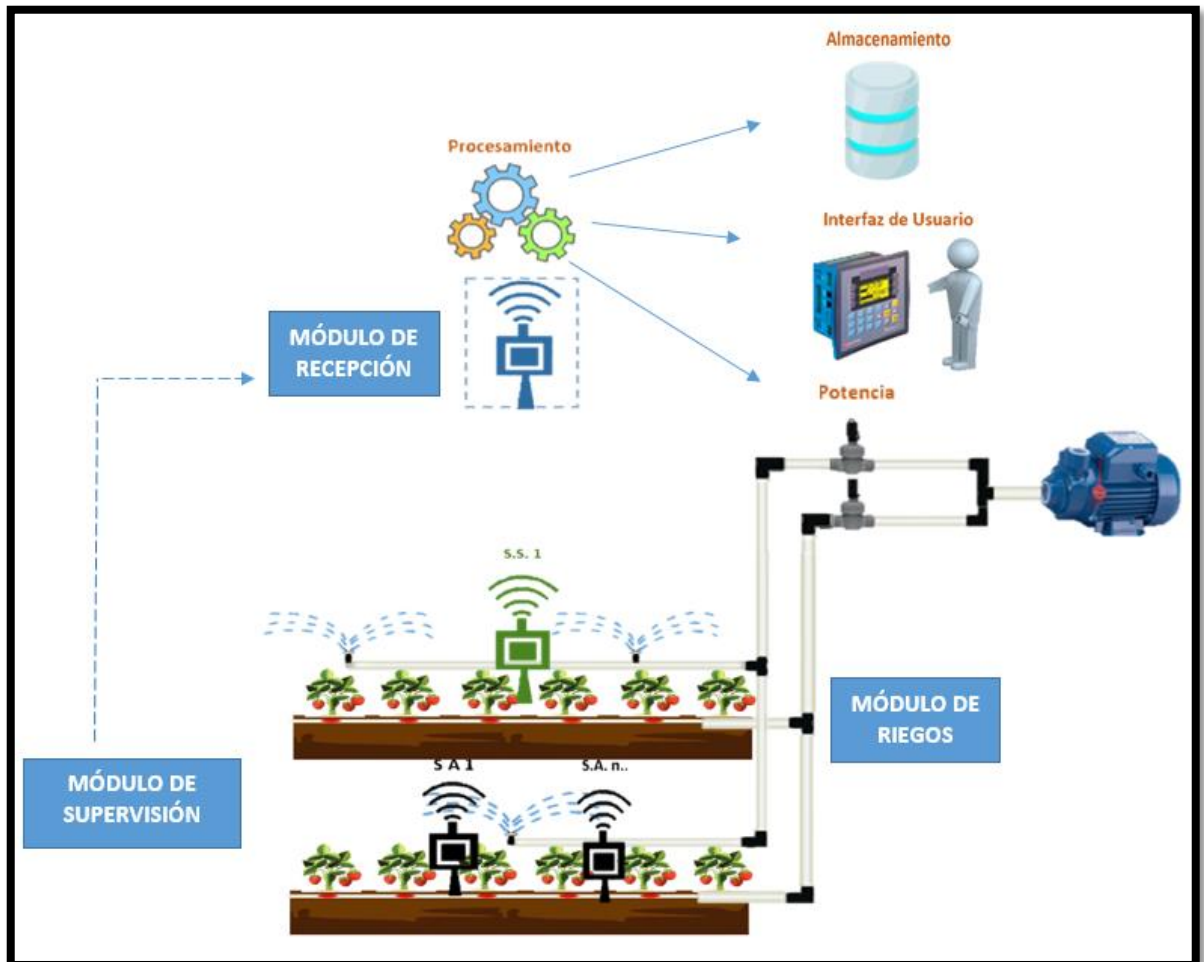


Figura 2-2: Arquitectura del sistema
Realizado por: ARCOS, Daniel. 2017

La comunicación entre los módulos de supervisión se lo realiza con una topología en estrella, por su facilidad de envío de datos, como se observa en la figura 3-2.

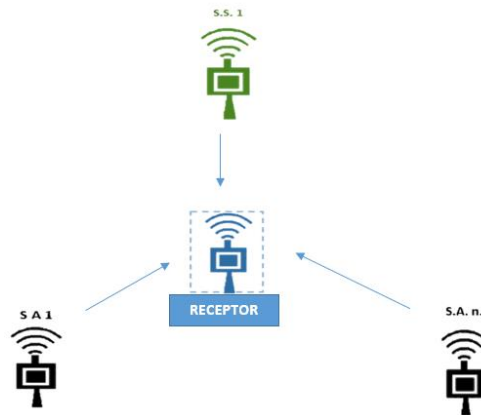


Figura 3-2: Topología del sistema.
Realizado por: ARCOS, Daniel. 2017

Tabla 2-2: Arquitectura del sistema

MÓDULO	DESCRIPCIÓN
SUPERVISIÓN	Es el primer módulo del sistema encargado de captar las variables de humedad, temperatura ambiente y la humedad del suelo. El número de módulos para el control de helada dependerá de la topología del terreno, en donde se instalará el sistema.
RECEPCIÓN	Recepta la información que envía el módulo de supervisión para procesarla, consta de bloques de: Procesamiento, almacenamiento, visualización de datos y un bloque de potencia, encargado de activar los actuadores: Bomba de agua y electroválvulas, para los sistemas de riego.
RIEGO	Consta de dos sistemas de riegos: A goteo y por aspersión que están instalados en el huerto de fresas.

Realizado por: ARCOS, Daniel 2017

2.3 Diseño de la arquitectura del sistema.

Con los módulos ya definidos se procede al diseño del hardware y selección de elementos necesarios para cada módulo.

2.3.1 Módulo de Supervisión.

Consta de un módulo de supervisión de Suelo, y varios módulos de Supervisión Ambiente dependiendo del terreno en donde será instalado el sistema.

2.3.1.1 Módulo de supervisión ambiente

Está encargado de la recolección de los datos ambientales en el huerto de fresa, utiliza sensores de humedad y temperatura que captan las variables, y son procesadas por la tarjeta de desarrollo. En la siguiente figura 4-2, se observa el diagrama de bloque de este módulo, en donde se aprecia la interconexión de cada uno de los bloques. El bloque de procesamiento, contiene un Arduino Pro-mini que recibe la información que envía el bloque de adquisición de datos conformado por el sensor DHT21, el cual capta la humedad y temperatura, por último se envía los datos mediante el módulo de transmisión de radio frecuencia el cual remite la información hacia el módulo de recepción.

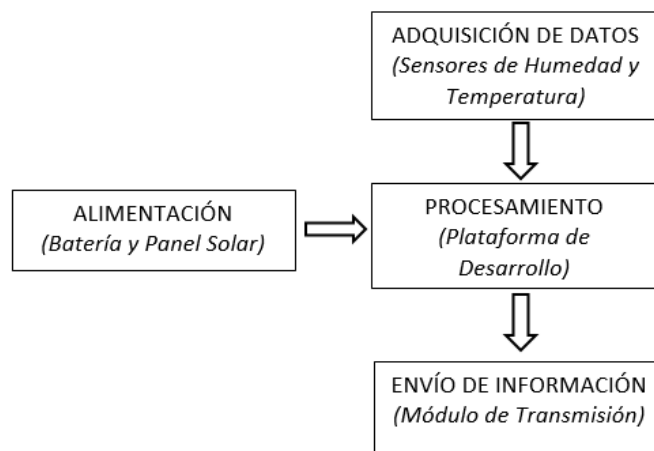


Figura 4-2: Diagrama de bloques módulo supervisión
Realizado por: ARCOS, Daniel. 2017




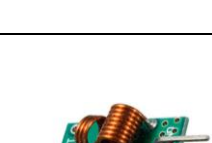
2.3.1.2 Módulo de supervisión de suelo.

Se encarga de recolectar los datos de humedad del suelo en las fresas, su diagrama de bloque lo se lo aprecia en la figura 2-4 anterior, en el bloque de Adquisición de datos contiene sensores de humedad el cual se encarga de captar la información.

2.3.1.3 Elementos que conforman el módulo de supervisión.

Los dispositivos electrónicos fueron elegidos de acuerdo a los requerimientos del sistema. En la tabla 3-2 se detalla los elementos de los módulos de supervisión con las características técnicas de cada uno de ellos.

Tabla 3-2 Elementos electrónicos de módulo de supervisión

Elemento	Imagen	Módulo al que conforma	Especificaciones técnicas
<i>Arduino Pro-mini</i>		Módulo de Ambiente y de Suelo	Alimentación 5V Corriente de consumo 40mA SRAM 2KB EEPROM 1KB
<i>Sensor de Humedad y Temperatura Ambiente DHT 21</i>		Modulo Ambiente	Alimentación 3.3-5V Corriente de consumo 8mA Rango de medición °T -40 a 80°C Rango de medición HR 0 a 100%
<i>Sensor de Humedad de Suelo FC-28</i>		Módulo de Suelo	Alimentación 3.3 – 5V Corriente consumo 35mA Salida valores analógicos Ajuste de sensibilidad
<i>Modulo RF 433 MHz emisor</i>		Módulo de Ambiente y de Suelo	Voltaje de operación: 3 – 12V Frecuencia de operación: 433.92 MHz Potencia: ~25mW @ 12V Error de frecuencia: +/- 150 KHz Máximo Velocidad de transmisión: <10 Kbps

Realizado por: ARCOS, Daniel. 2017

Cálculo consumo de corriente de los módulos.

Para seleccionar la batería adecuada de los módulos es necesario calcular las corrientes de todos los elementos electrónicos de cada módulo de supervisión. En la tabla 4-2 se observa los detalles de corriente de consumo

Tabla 4-2: Consumo de corriente cada módulo de supervisión

Módulo Ambiente		Módulo Suelo	
Elemento	Consumo	Elemento	Consumo
Arduino Pro-mini	15 mA	Arduino Pro-Mini	15 mA
Sensores	1,5 mA	Sensores	140 mA
Modulo RF	4 mA	Módulos RF	4 mA
TOTAL	20,5 mA	TOTAL	159 mA

Realizado por: ARCOS, Daniel. 2017

Batería de alimentación 5V y Panel solar.

Para la alimentación de los módulos se utilizará un cargador solar de la figura 5-2, el cual consta de una batería de 5V a 12000mAh y un panel solar de 200mA, el cual ayudara mantener una energía constante en los módulos de supervisión.



Figura 5-2: Cargador solar.
Realizado por: ARCOS, Daniel. 2017.

2.3.1.4 Diagrama de conexión módulo Supervisión Ambiente

Como controlador se encuentra el Arduino pro-mini, en donde se conecta el sensor DHT21 y el módulo de transmisión de radio frecuencia, todo el módulo esta alimentado por una batería de 5v, junto con un panel solar que entrega energía continua.

- Sensor DHT21 consta de tres pines, en donde se conecta a GND, VCC, conectados al Arduino y el tercer el pin de SEÑAL conectado a terminal dos del Arduino.
- Para la alimentación del módulo de radio frecuencia TX es conectado a 5V y al terminal GND. El envío de datos se lo hace a través del terminal de entrada DATA, conectado hacia el terminal siete del Arduino.

Las conexiones de los elementos se aprecian en la figura 6-2.

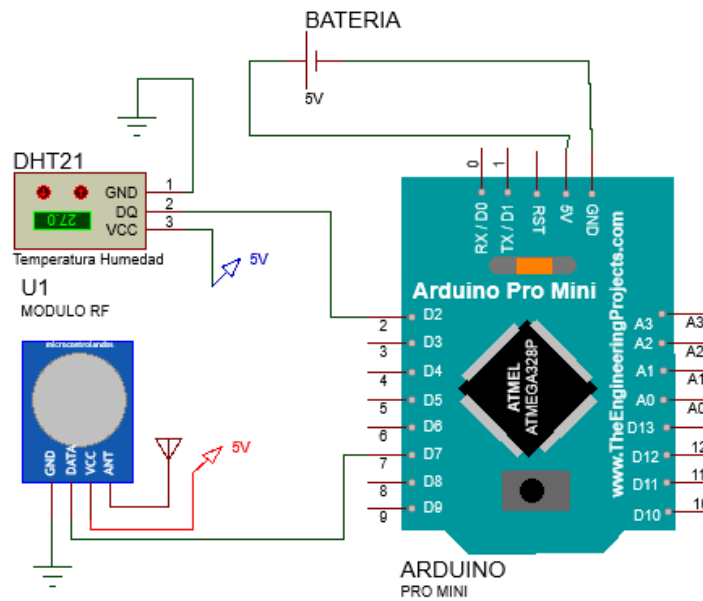


Figura 6-2: Conexión módulo de supervisión ambiente
Realizado por: ARCOS, Daniel. 2017

2.3.1.5 Diagrama de conexión módulo de Supervisión de Suelo

Este bloque es idéntico al de supervisión ambiente, al usar un Arduino pro-mini y alimentado con la batería y el panel solar de 5 V. Consta de cuatro sensores de humedad con el fin de supervisar de una manera más efectiva las camas de fresas, dos sensores en la parte superior y los otros en la parte inferior de la cama, cubriendo las raíces de las plantas. En la figura 7-2 se ve el diagrama de conexión

- En los terminales analógicos del 0 a 3 se encuentra conectados a la entrada de DATOS de los sensores de humedad FC-28
- El módulo de Radio Frecuencia TX, está conectado al terminal siete del Arduino para la transmisión de datos, y alimentado con 5v.

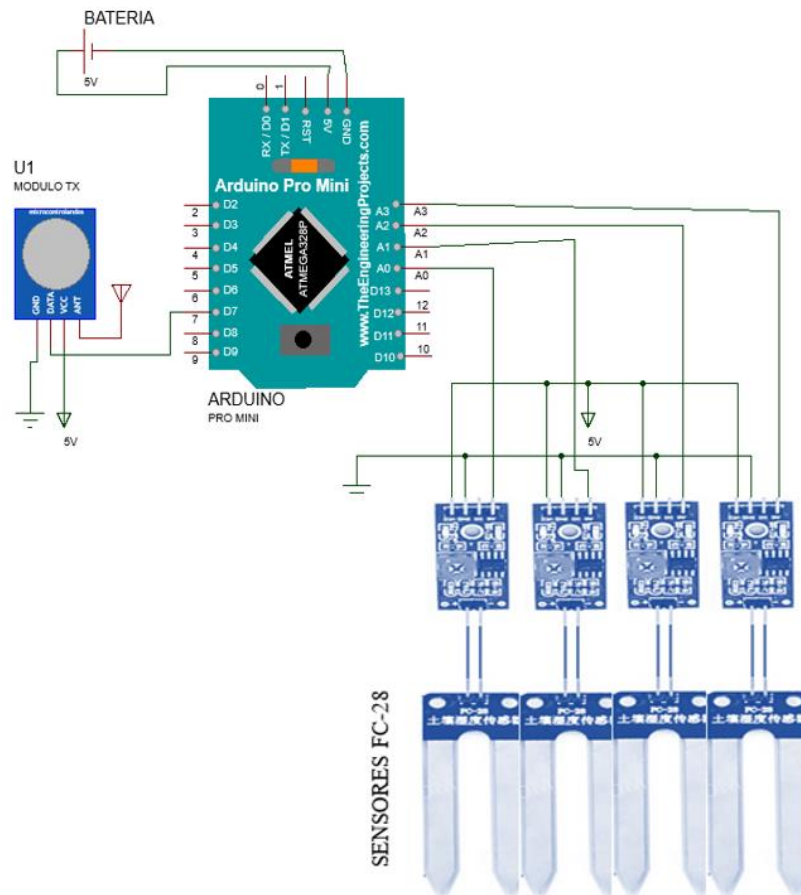


Figura 7-2: Conexión módulo supervisión de suelo.
Realizado por: ARCOS, Daniel. 2017.

2.3.2 Módulo de Recepción.

Aquí llegan todos los valores de los parámetros que envían los módulos de supervisión, al realizar el estudio en el primer capítulo de los rangos de humedad y temperatura, el sistema actuará de forma inmediata procesando los datos y activando el módulo de potencia para encender los elementos de actuación, de acuerdo a como se esté programado.

Bloque de visualización: Consta de una pantalla una *LCD* y un *teclado matricial*, mediante la ayuda de una interfaz de usuario, el interesado maneja con el teclado y observa los valores que se monitorean en el huerto. Así como activa y desactiva los riegos de forma manual.

Bloque Reloj Tiempo Real: Es necesario que nuestro sistema de Supervisión, conste la fecha exacta de supervisión, para lo cual un Módulo reloj RTC para Arduino logra generar.

Bloque de Almacenamiento: Con la ayuda de un módulo SD Card, y una tarjeta de almacenamiento, se guarda los datos que captan los módulos de Supervisión con la fecha y la hora.

Bloque de Potencia: Luego que los datos son procesados se envían las señales de activación para que el bloque de potencia ponga en marcha los actuadores para el riego respectivo. El bloque tiene una alimentación con corriente alterna.

En la figura 8-2 se aprecia el diagrama de bloques de cómo está constituido el módulo de recepción.

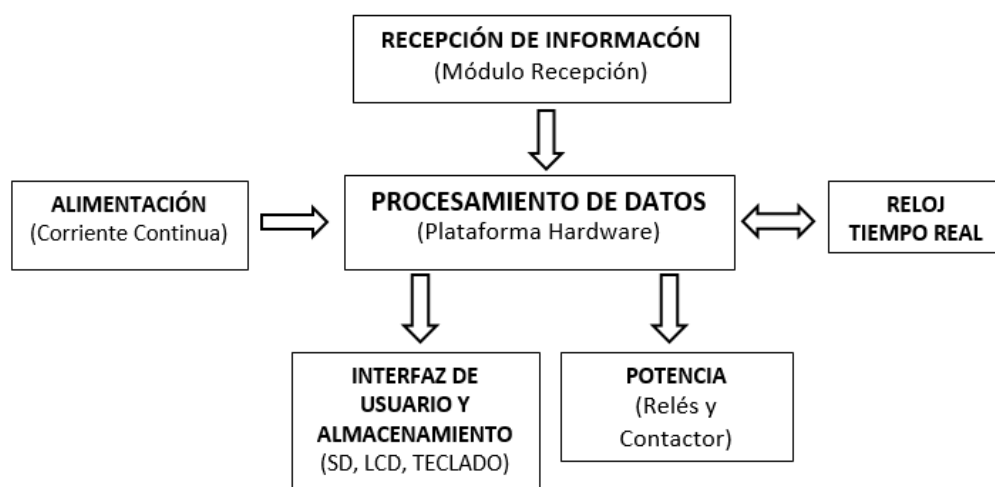


Figura 8-2: Diagrama de bloques módulo de recepción
Realizado por: ARCOS, Daniel. 2017

El bloque de potencia consta de un diagrama extenso en la figura 9-2, se puede observar cómo está compuesto su diagrama.

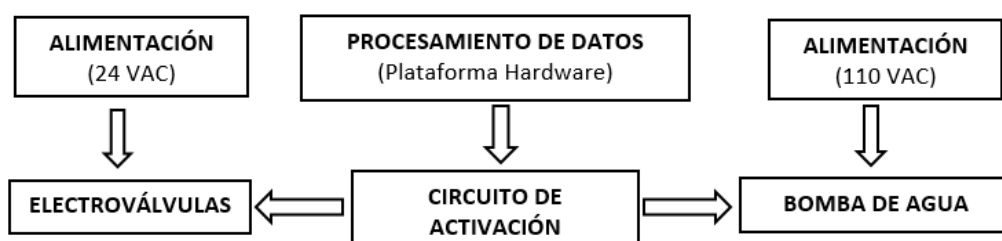


Figura 9-2: Diagrama de bloques de bloque de potencia
Realizado por: ARCOS, Daniel. 2017

2.3.2.1 Elementos que conforma el Módulo de Recepción.

De acuerdo al diagrama de bloques realizados en el diseño de la arquitectura, como controlador principal se utiliza un Arduino Mega, que brinda las características necesarias para el sistema como el número de entradas y salidas.

Para el bloque de potencia como protección al momento de la activación de la bomba eléctrica es necesario utilizar un contactor y para seleccionar este elemento se procede al cálculo de la corriente que consume la bomba. Los datos de la bomba eléctrica que trabaja con un voltaje de 110V la corriente son: 1HP a 110V, entonces:

$$P = I.V$$

$$I = P/V$$

Se sabe que 1 HP=746W. Entonces:

$$I = 746W/110V$$

$$I = 746W/110V$$

$$I = 6.78A$$

Entonces se procede a utilizar un contactor de 12 A. los elementos que conforman cada uno de los bloques del módulo de recepción se detallan en la tabla 5-2.

Tabla 5-2: Elementos del módulo de recepción

Elemento	Imagen	Observaciones
Arduino Mega		Microcontrolador ATmega 328 Voltaje de entrada 5-12V Voltaje funcionamiento 5V E/S Digitales 14 Pines PWM 6 Corriente 6mA SRAM 2KB EEPROM 1KB
Módulo receptor RF 433 MHz		Voltaje de operación: 4.5 – 5.5V Corriente de operación: 5.5 mA Modulación: ASK / OOK Frecuencia de operación: 433.92 MHz Velocidad de transmisión: <9.6 Kbps
Módulo Reloj RTC IC2		Chip DS1307 Batería de respaldo para el RTC incluye en el módulo Alimentación: 4.5 – 5.5 VDC Memoria EEPROM I2C: 24c32
Módulo Micro SD Card		Voltaje de funcionamiento: 3.3V y 5V Protocolo de comunicación: SPI Soporta tarjetas SD y Micro SD Testeada hasta con tarjetas de 4GB.
Pantalla LCD 20x4		20 caracteres x 4 líneas Backlight de LED color azul Caracteres color blanco Voltaje de alimentación: 5 V
Teclado Matricial 4x3		12 botones con organización matricial (4 filas x 3 columnas) Más resistencia al agua y al polvo. Máximo voltaje operativo: 24 V DC Máxima corriente operativa: 30 mA
BLOQUE DE POTENCIA		
<i>Bomba de agua</i>		Voltaje 110 o 220V Corriente 6 – 8A Potencia 1HP
<i>Electroválvulas</i>		Alimentación: 25VAC Corriente: 350mA – 60Hz 370mA – 50Hz

<i>Módulo Relés de 5V</i>		10A a 125VAC 10A a 30DC Activación a 5V
<i>Transformador de 24V</i>		Voltaje de Entrada: 110 – 220V Voltaje Salida: 12- 0 - 12 Corriente: 3A
<i>Interruptor o disyuntor</i>		Polos: 2 Corriente Nominal: 16A. Frecuencia: 50 / 60Hz. Temperatura ambiente: Limites: -20°C, +50°C
Contactor		Electricidad Tipo: AC Número de polos: 3 Circuito principal Tensión 110V 220V Corriente 12A

Realizado por: ARCOS, Daniel. 2017

2.3.2.2 Diagrama de conexiones módulo de recepción.

El Arduino Mega es el controlador principal del módulo que cumple la función de procesamiento, los dispositivos electrónicos que se utiliza para este diseño son los siguientes:

- El módulo de reloj RTC I2C, contiene terminales SDA Y SCL los cuales se conectan a los mismos de Arduino Mega, para la programación de la hora, en este caso son los terminales 20 y 21 respectivamente, y su alimentación es de 5V. En la tabla 6-2 se detallan las conexiones.

Tabla 6-2: Pines de conexión módulo RTC

Módulo Reloj RTC	Arduino MEGA Terminales
VCC	Terminal 5V
GND	GND
SDA	21
SCL	20

Realizado por: ARCOS, Daniel. 2017

- El módulo SD Card se encuentra alimentado de la misma forma con 5V, los terminales de conexión se especifican en la siguiente tabla 7-2.

Tabla 7-2: Pines de conexión módulo SD Card

Módulo SD Card	Arduino MEGA Terminales
MISO	50
MOSI	51
CS	10
CLK	52
GND	GND
VCC	Termina 5V

Realizado por: ARCOS, Daniel. 2017

- Como medio de visualización para la supervisión de los datos, se encuentra una LCD, conectados los terminales RS, E, D3, D4, D5, D7. En la tabla 8-2 se muestra los pines de conexión.

Tabla 8-2: Pines de conexión pantalla LCD

Pantalla LCD	Arduino MEGA Terminales
RS	40
E	41
D3	42
D4	43
D5	44
D7	45
K, VDD	GND
A, VCC	Terminal 5V

Realizado por: ARCOS, Daniel. 2017

- Para que el usuario pueda manejar el menú de usuario se utiliza en teclado matricial, la cual tiene los siguientes terminales de conexión como se observa en la tabla 9-2.

Tabla 9-2: Pines de conexión teclado matricial

Teclado Matricial	Arduino MEGA Terminales
A	32
B	28
C	34
D	24
1	30
2	26
3	22
VCC	Terminal 5V
GND	GND

Realizado por: ARCOS, Daniel. 2017

- El módulo de Radio Frecuencia RX, está conectado al terminal 46 del Arduino, y alimentado por 5V.

En la figura 10-2 se observar el diagrama de conexión de todo el módulo de Recepción.

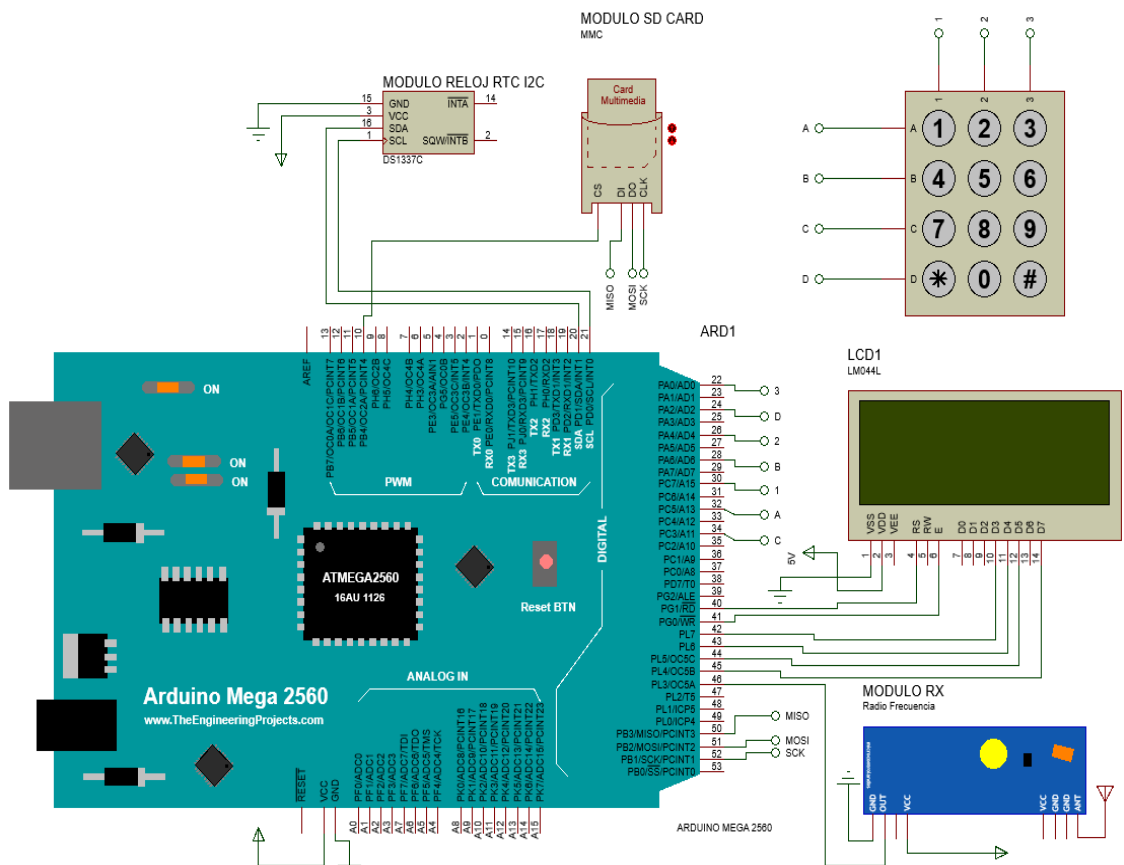


Figura 10-2: Diagrama de conexión módulo recepción
Realizado por: ARCOS, Daniel. 2017.

Diagrama de conexión bloque de potencia

Aquí se encuentra el panel de control de potencia el cual activa las electroválvulas y la bomba de agua eléctrica de acuerdo a la programación del bloque de procesamiento. La bomba de agua se alimenta de 110VAC, para el encendido es necesario que conste de un elemento de protección de acuerdo a su potencia. Las electroválvulas se alimentan de 24VAC. Todas las señales de activación que envía el bloque de procesamiento de datos, pasan por relés que permiten la activación de estos elementos, en la figura 10-2 muestra las conexiones de este bloque.

En la tabla 10-2 se especifica los terminales de conexión del módulo de relés con los terminales de Arduino y los actuadores a los que se encuentran conectados.

Tabla 10-2: Pines de conexión módulo relés

Módulo relés	Arduino Mega
VCC	5V
GND	GND
1	14 (Bomba)
2	15 (Electroválvula Helada1)
3	16 (Electroválvula Helada2)
4	17 (Electroválvula Goteo)

Realizado por: ARCOS, Daniel. 2017.

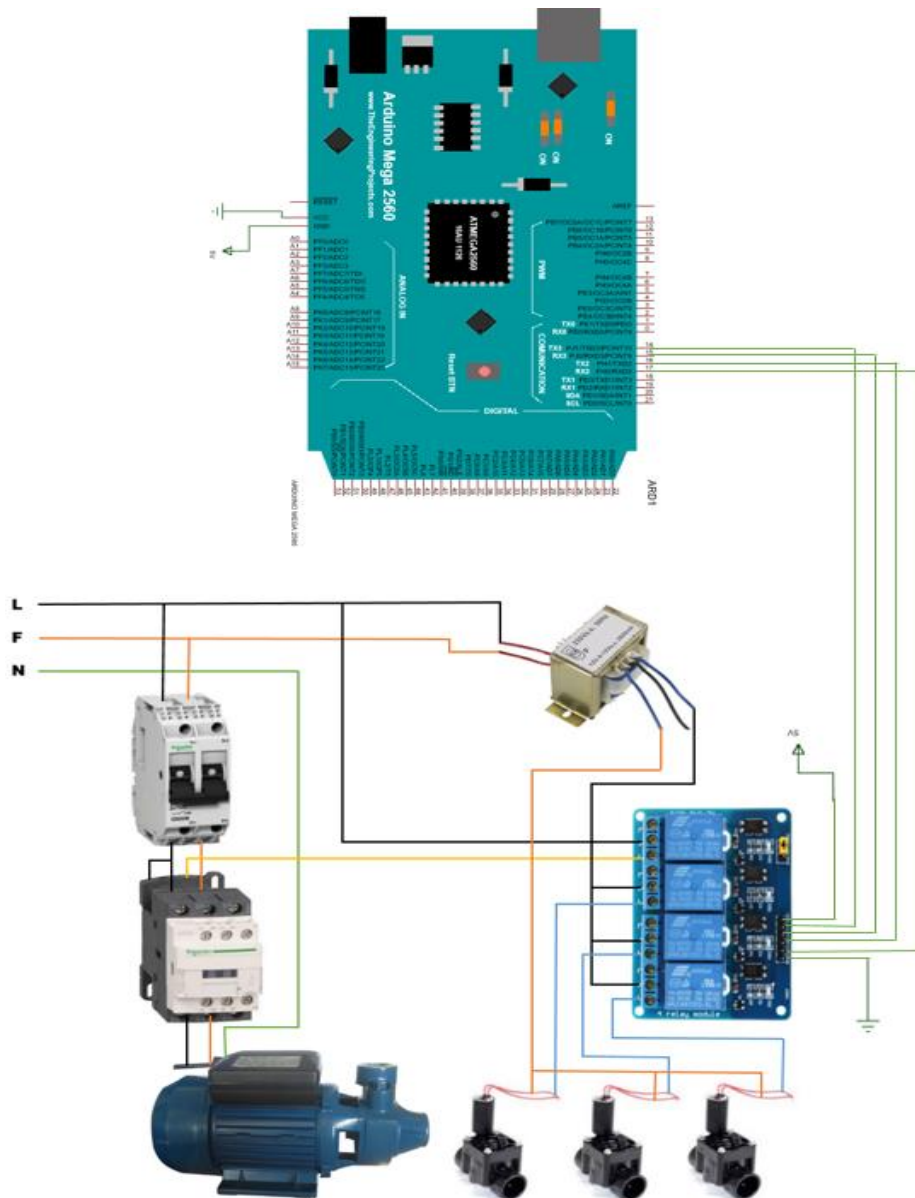


Figura 11-2: Conexiones bloque de potencia,
Realizado por: ARCOS, Daniel. 2017.

2.3.3 Módulo de Riego.

En la figura 12-2 se encuentra el diagrama de bloques de este módulo, que consta de una bomba de agua la cual se encarga de extraer el recurso hídrico desde un tanque reservorio, esta pasa por las electroválvulas que son las que se encargan de distribuir a los sistemas de riego.

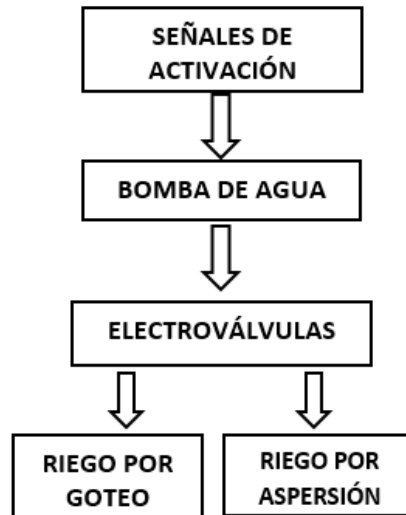


Figura 12-2: Diagrama de bloques módulo de riego
 Realizado por: ARCOS, Daniel. 2017

2.3.3.1 Selección de elementos para el módulo de Riego

Para que el sistema de riego por aspersión reduzca el daño en las plantas es necesario instalar aspersores de baja intensidad de precipitación, y para esto se recomienda utilizar los aspersores de 2.5 mm/h. En la figura 13-2 se aprecia este tipo de aspersor. (Martínez, 2007).



Figura 13-2: Aspersores anti-heladas
 Realizado por: MARTINEZ, Leoncio. 2007

2.4 Calibración del sensor FC-28 mediante un tensiómetro de suelo.

En el capítulo anterior se aprecia que el rango de valores de humedad de suelo para huertos de fresa debe estar entre 17–25 centibares, para medir en estas unidades se utilizó el tensiómetro de suelo A.M.I GAUGE, el que se aprecia en la figura 14-2.



Figura 14-2 Tensiómetro de suelo A.M.I GAUGE
Realizado por: ARCOS, Daniel. 2017

Para saber el rango de humedad en el sensor FC-28 para la supervisión de suelo, se realizó una relación entre los valores de los dos medidores. Para esto se ubica los dos sensores en la cama de fresa como se ve en la figura 15-2, durante una semana, y se empezó a tomar las medidas del rango de humedad que se necesita.



Figura 15-2: Calibración de los valores de humedad
Realizado por: ARCOS, Daniel. 2017

Luego de las muestras obtenidas durante esos días, se obtuvo los siguientes valores, como se observa a la tabla 11-2.

Tabla 11-2: Calibración para sensor de humedad FC-28

SENSOR	TENSÍOMETRO		SENSOR FC-28	
	Muestras	Estado Húmedo	Estado Seco	Estado Húmedo
1	17 cbares	25 cbares	91%	77%
2	17 cbares	25 cbares	92%	78%
3	17 cbares	25 cbares	91%	78%

Realizado por: ARCOS, Daniel. 2017

Gracias a los valores obtenidos en las mediciones junto con el tensiómetro y el sensor FC-28, se tomó como valor máximo de humedad 91%, mientras que el valor mínimo 79%, así se puede programar para la activación del riego por goteo.

2.5 Diseño del Software del Sistema

La programación realizada en el software Arduino IDE para cada bloque, se encuentra especificada en el *Anexo 1*. A continuación se muestra los diagramas de flujo, funciones y las librerías utilizadas en la programación.

2.5.1 *Requerimientos del sistema*

Las necesidades en cuanto al software del sistema de supervisión son los siguientes:

- Trabajar con señales analógicas para los rangos de humedad
- Utilizar librerías para la programación del sensor DHT21
- Enviar los datos de los módulos de supervisión al módulo de recepción.

Módulo de recepción

- Visualización de los datos, hora y fecha en la pantalla LCD
- Diseñar una interfaz de usuario para activar y desactivar los riegos manualmente
- Guardar los datos en una tarjeta micro SD con hora y fecha.

2.5.2 Programación Módulo Ambiente

En la figura 16-2 se muestra el diagrama de flujo técnico del módulo, el procedimiento para realizar:

- Inclusión de bibliotecas y declaración de las variables necesarias que se utiliza para la programación.
- Configuración de los terminales como las entradas y salidas, para los dispositivos electrónicos: Módulo de radiofrecuencia y sensor de humedad y temperatura.
- Lectura de los datos de humedad y temperatura por medio del sensor.
- Conversión de los datos de entero a string.
- Para posteriormente agregar un inicio a la trama.
- Envío de los datos por medio del módulo de radio frecuencia.

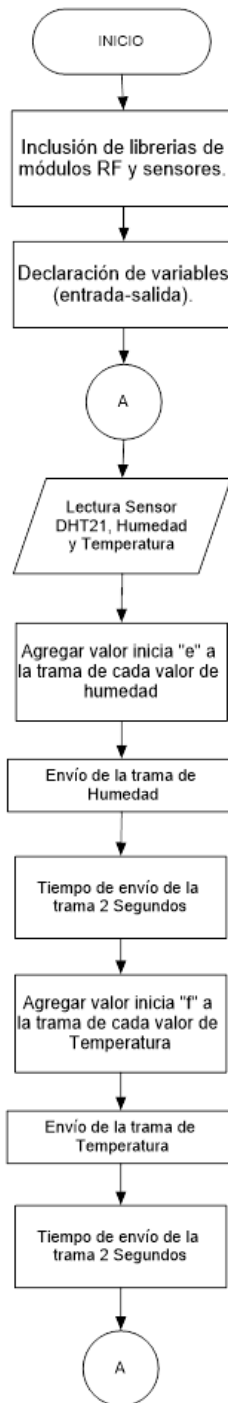


Figura 16-2: Diagrama de flujo módulo Supervisión Ambiente
 Realizado por: ARCOS, Daniel. 2017

En el diagrama de flujo se menciona el envío de datos la letra “f” como valor inicial para envío de temperatura y “e” para humedad, en la siguiente figura 17-2 se muestra la trama enviada.

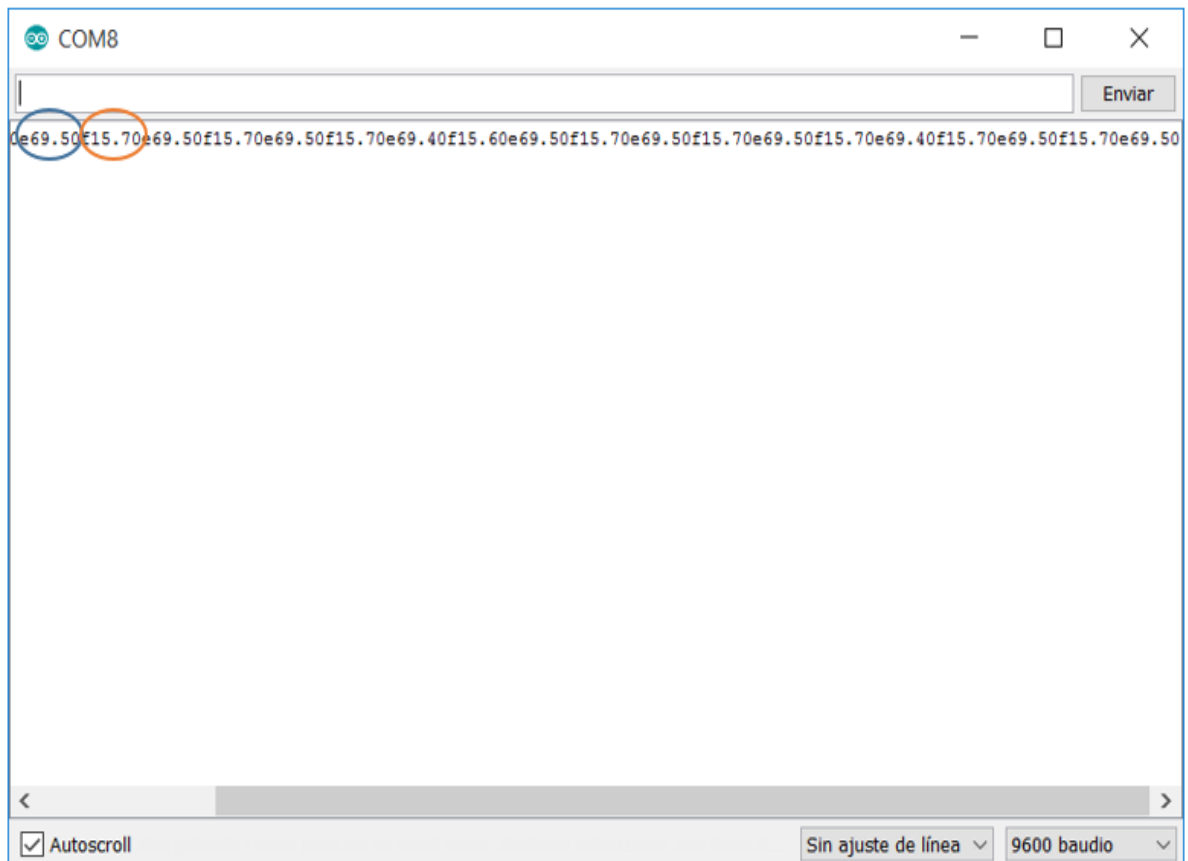


Figura 17-2: Trama de datos módulo Supervisión Ambiente
Realizado por: ARCOS, Daniel. 2017

2.5.2.1 Bibliotecas para la programación

VirtualWire.h: Es una biblioteca de comunicaciones que permite que Arduinos se comuniquen utilizando transmisores y receptores de RF.

DHT.h: Esta librería permite la comunicación con los sensores DHT11, DHT21, DHT22, para obtener los datos de humedad y temperatura

2.5.3 Diagrama de flujo del Módulo Ambiente

La programación del siguiente módulo es casi similar al módulo de ambiente, como se puede observar en el diagrama de flujo de la figura 18-2, el procedimiento es el siguiente.

- Colocación de bibliotecas acorde a la programación del módulo, y declaración de las variables a utilizar.

- Configuración de los terminales de entradas y salidas, para los dispositivos electrónicos: Módulo de radiofrecuencia y sensores de humedad y temperatura.
- Lectura de valores de los cuatro sensores de humedad.
- Conversión de los valores de humedad de entero a string, para posteriormente agregar un inicio a la trama.
- Envío de la trama de valores de temperatura del suelo por medio del módulo de radiofrecuencia TX.

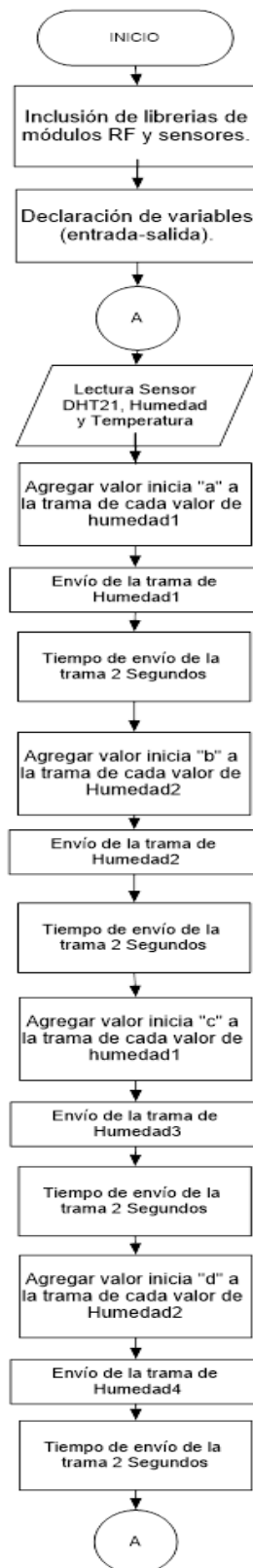


Figura 18-2: Diagrama de flujo módulo Supervisión de Suelo
 Realizado por: ARCOS, Daniel. 2017

La trama consta en el envío de los cuatro datos de humedad, cada uno con su valor inicial respectivo como se muestra en la figura en la figura 19-2.

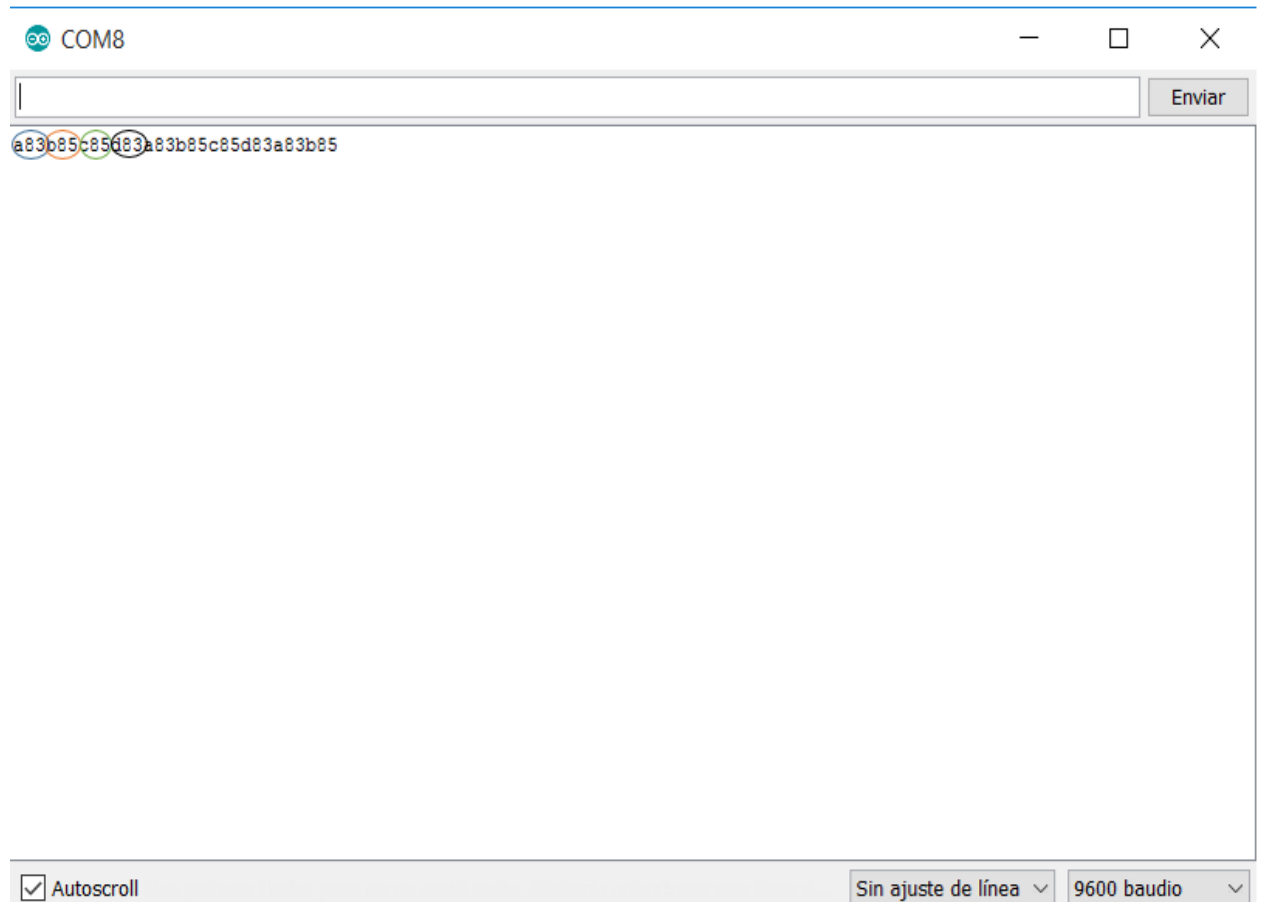


Figura 19-2: Trama enviada Módulo de suelo
Realizado por: ARCOS, Daniel. 2017

2.5.3.1 Bibliotecas para la programación

Para la comunicación del módulo de radiofrecuencia se utilizó la biblioteca **VirtualWire** ya mencionada anteriormente.

Cabe recalcar que para este módulo no se utilizó funciones para la programación.

2.5.4 Programación del módulo de Recepción.

La programación del módulo de recepción es más extensa que el módulo de supervisión como se observa en la figura 20-2, consta de varias bibliotecas, y funciones que integran el diseño del software, consta del siguiente procedimiento.

- Declaración de las variables a utilizar e inclusión de bibliotecas necesarias para la programación.
- Configuración de los terminales como las entradas y salidas, para los dispositivos electrónicos y módulos de: Relés, radiofrecuencia, SD card, Reloj RTC, teclado y pantalla LCD.
- Lectura de los datos por medio del módulo de radiofrecuencia RX, que envían el bloque de supervisión.
- Transformación de la trama de datos a valores enteros.
- Calculo del punto de rocío entre los valores de humedad y temperatura ambiente.
- Comparación de valores máximos y mínimos del ambiente, para el control de helada.
- Activación y desactivación de la bomba eléctrica y electroválvulas que controlar el riego por aspersión y goteo.
- Calculo de la fecha y hora actuales.
- Guardado en una tarjeta micro SD de los valores de humedad de suelo y ambiente en diferentes archivos de texto., especificando la hora y fecha.
- Visualización de los datos y control de los riegos por medio de en una pantalla LCD.

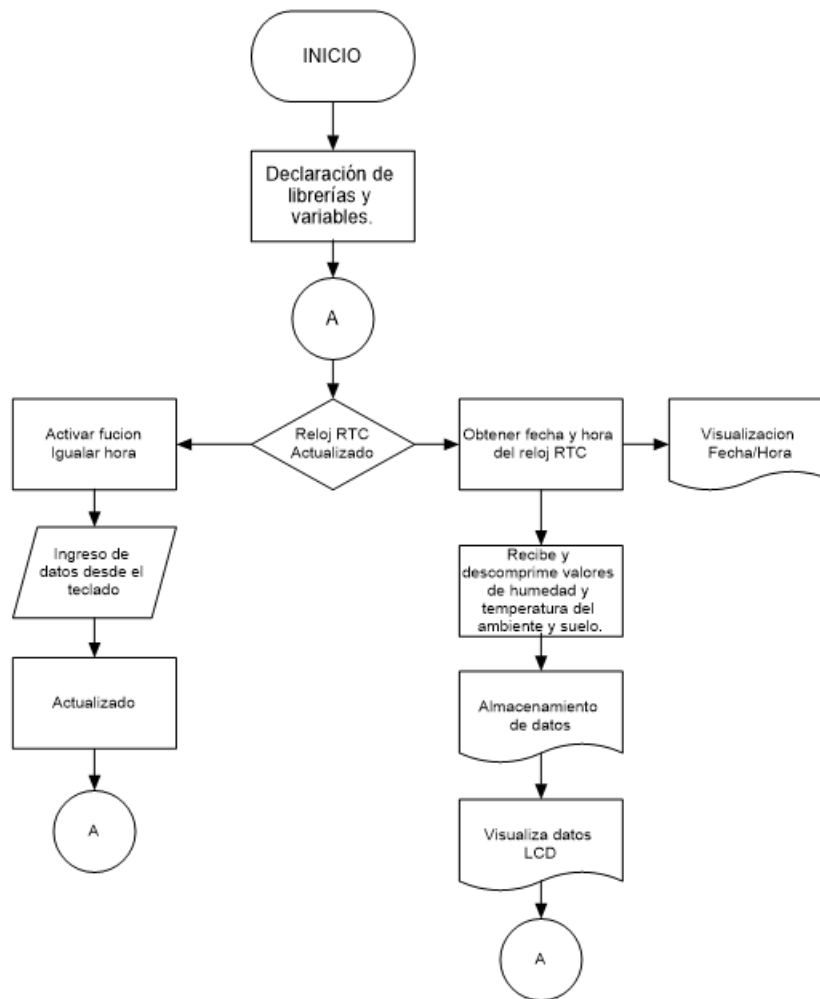


Figura 20-2: Diagrama de flujo bloque almacenamiento y visualización
 Realizado por: ARCOS, Daniel. 2017

El diagrama de flujo para la activación de los actuadores se encuentra detallada en la figura 21-2.

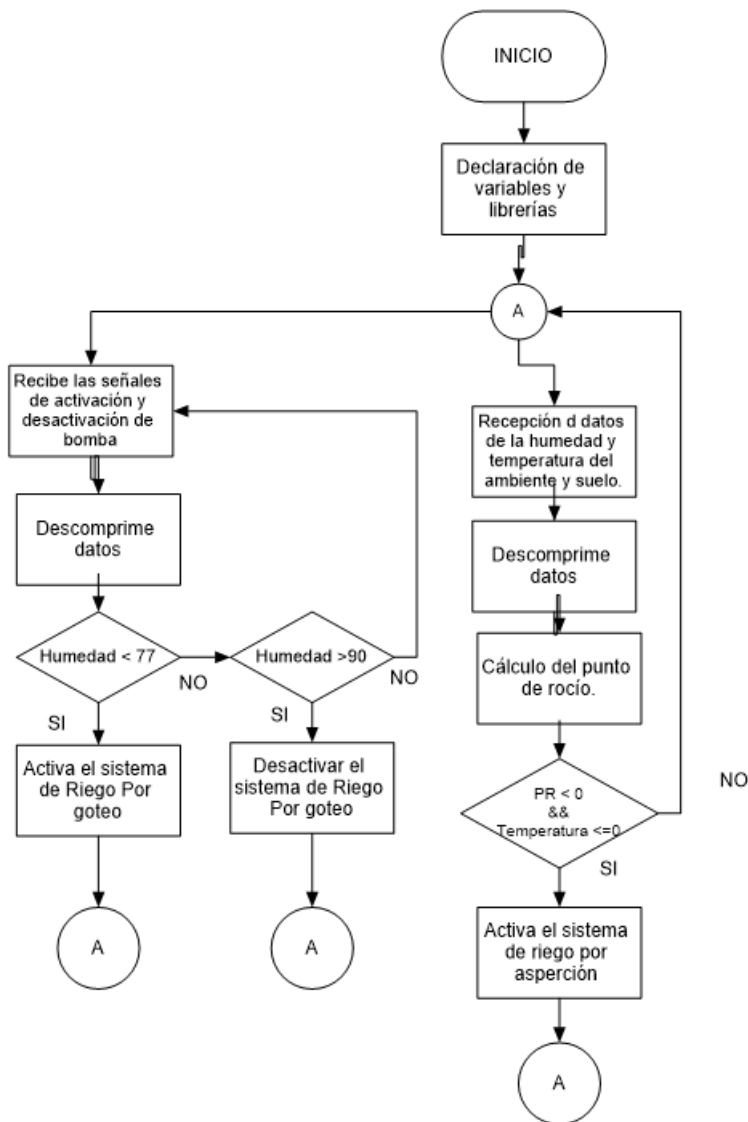


Figura 21-2: Diagrama de flujo bloque de potencia.
 Realizado por: ARCOS, Daniel. 2017

2.5.4.1 Bibliotecas para el módulo de recepción.

En esta programación se incluyó bibliotecas para cada uno de los módulos electrónicos incluidos en este bloque.

- **LiquidCrystal:** Esta biblioteca permite a que el Arduino pueda controlar la LCD, trabaja en el modo de 4 u 8 bits.
- **Wire:** Permite la comunicación I2C, utilizado para la configuración de fecha y hora del reloj RTC.

- **Keypad.h:** Usada para la matriz de teclado, manejar de mejor manera las pulsaciones, y ocultar las llamadas de pinMode.
- **SPI.h:** Es un protocolo utilizado para la comunicación con otro dispositivo externo, en este caso el módulo SD Card.
- **SD.h:** Permite la lectura y escritura de archivos de texto en la placa Arduino
- **DS1307RTC.h:** Biblioteca necesaria para acceder al reloj RTC1307
- **Time.h:** Añade funcionalidad a la hora en el Arduino. Permite el boceto para obtener la fecha y hora: año, mes, día, hora, minuto, segundo.

2.5.4.2 *Funciones para la programación módulo de recepción.*

- **rocio():** Creada para calcular el punto de rocío, utiliza los valores enteros de humedad y temperatura ambiente, y entrega de la misma forma otro valor entero.
- **menu():** Programada para visualizar el menú de usuario en la pantalla LCD.
- **guardado_Suelo():** Creada para almacenar los datos de humedad de suelo del huerto, cuando los datos son descomprimidos, se almacenaran con la hora y fecha que fueron monitoreados.
- **guardado_Helada():** Creadas independientemente para guardar los datos de Humedad y Temperatura ambiente, se crean diferentes archivos para cada módulo, y para mayor comodidad de supervisión del usuario.
- **Igualar_reloj():** Dentro de la función se encuentra la programación para igualar la fecha y hora, el usuario ingresa el año, mes, día, la hora y los minutos, estos se guardaran en la memoria RTC y se visualizara en al LCD.
- **control_manual():** En esta función se programa el encendido y apagado de los diferentes riegos del huerto, se lo realiza mediante el teclado numérico instalado en el módulo de recepción.
- **monitoreo_huertos():** Función encargada de la visualización en la pantalla LCD, de los datos de humedad y temperatura ambiente y suelo, para el monitoreo por parte del usuario

- **activacion_riego():** Dentro de esta función se encuentran las comparaciones de los datos máximos y mínimos de humedad y temperatura para la activación de las válvulas para los riegos.
- **activacion_riego_suelo():** Programado para la activación de la bomba y electroválvula que activa el riego a goteo del huerto.

2.6 Cálculo de la antena para los módulos de transmisión

La distancia de transmisión depende del diseño de la antena para los módulos. La antena que se ubica en los módulos es el transductor que convierte la señal eléctrica en ondas electromagnéticas de RF. La señal que llega a la antena debe tener la frecuencia apropiada, entonces se realizó los siguientes cálculos.

$$\lambda = \text{Longitud de Onda}$$

La longitud de una onda es el período espacial de la misma, es decir, la distancia a la que se repite la forma de la onda.

$$f = 433.92\text{Mhz}$$

La frecuencia de la portadora para nosotros es 433.92MHz que es la frecuencia a la que opera el módulo transmisor.

Las ondas electromagnéticas se propagan a una velocidad que en el vacío es: $C = 300.000 \text{ Km/seg}$. Se asume que esa también es la velocidad a la que se propagan en el aire. La fórmula que relaciona longitud de onda, frecuencia, y velocidad de propagación es: (Bolaños, 2011).

$$\lambda = C/f$$

$$\lambda = 300.000/433.92$$

$$\lambda = 0.69\text{m}$$

Para calcular la longitud de antena se necesita dividir para cuatro

$$\frac{\lambda}{4} = 17\text{cm}$$

Esta longitud es apropiada para el módulo emisor y receptor como se ve en la figura 22-2

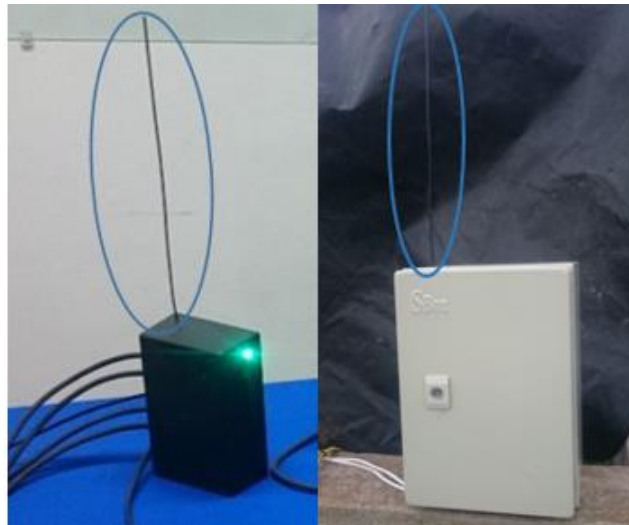


Figura 22-2: Longitud de Antenas de los módulos
Realizado por: ARCOS, Daniel. 2017

2.7 Diseños físicos del sistema.

Luego del diseño del software y del hardware del sistema, los módulos de supervisión se ubicaron en una caja plástica de 7*10*14, como se observa en la figura 23-2. El panel solar se ubicó en la parte exterior para la atracción de los rayos solares.

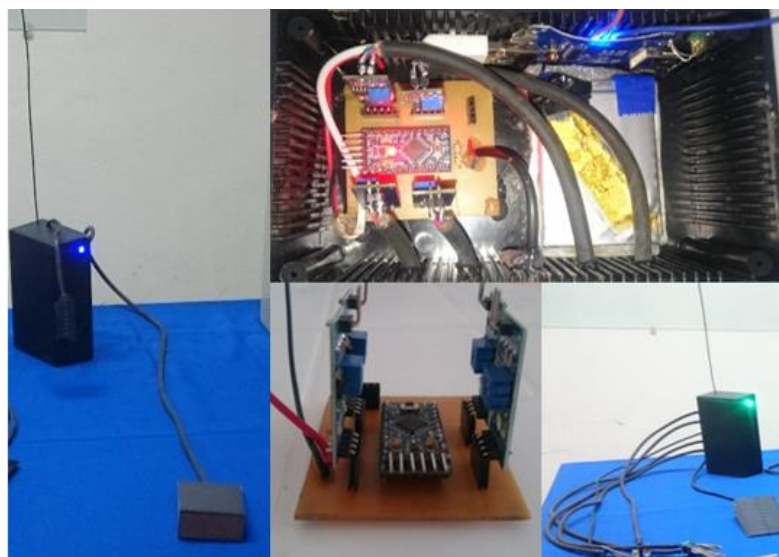


Figura 23-2: Módulos de supervisión
Realizado por:: ARCOS, Daniel. 2017

El módulo de recepción se instaló en una caja metálica como mayor protección, como se observa en la figura 24-2.



Figura 24-2: Módulos de recepción y potencia
Realizado por: ARCOS, Daniel. 2017

CAPITULO III

3 ANALISIS Y RESULTADOS

En el presente capitulo se habla de la instalación del sistema, las pruebas realizadas para cada uno de los módulos, que tan eficiente son en la transmisión de datos, comparando con equipos y herramientas de monitoreo similares. Y finalmente se presenta un análisis de costos del sistema comparando con el presupuesto de un sembrío de fresas.

3.1 Análisis de las variables ambientales del terreno para la instalación del sistema

En el capítulo uno se estudió la topología de los terrenos, y el efecto de la helada que produce en ellos, para comprobar esto se realizó mediciones en 3 puntos diferentes del huerto, (Parte superior, media e inferior), como se observa en la figura 1-3, se ubicaron termómetros y se procedió a supervisar por una semana en diferentes días y a distintos horarios. Los resultados de las mediciones se observan ver en la tabla 1-3.

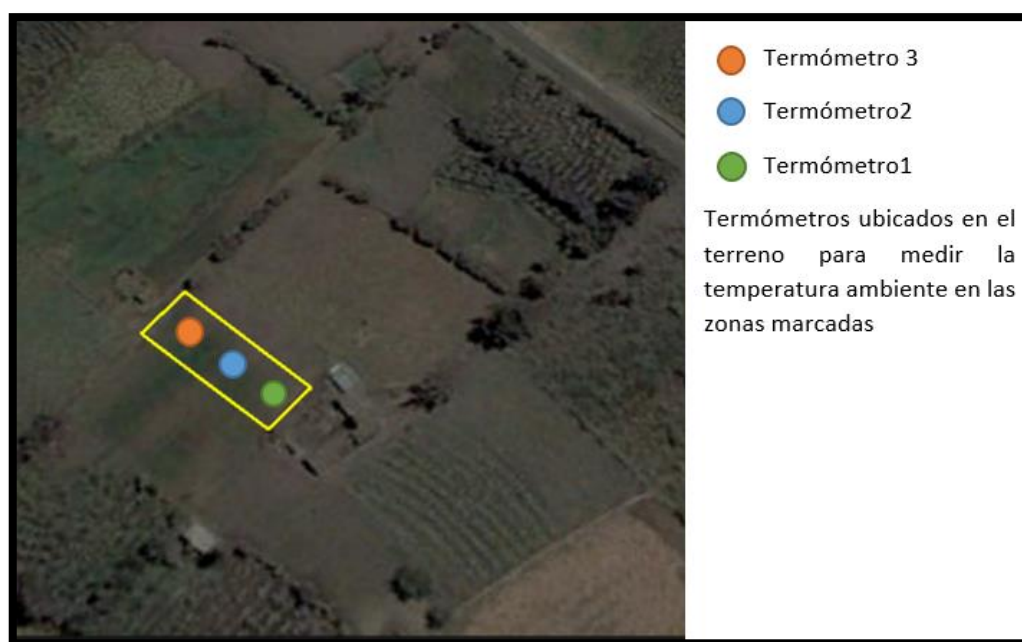


Figura 1-3: Ubicación de termómetros en el terreno
Realizado por: ARCOS, Daniel. 2017

Tabla 1-3: Mediciones de temperatura ambiente en el huerto

Día 1									
Hora	5:00	5:30	6:00	14:00	14:30	15:00	19:15	19:40	20:30
Termómetro 1 (°C)	8,9	9	30	30,5	31	9	8,2	7,9	7,5
Termómetro 2 (°C)	8,8	8,9	30	30,5	31,5	9,1	8,1	8	7,4
Termómetro 3 (°C)	8	8,1	30	30,5	31	8,8	7,2	6,8	6,2
Día 2									
Termómetro 1 (°C)	8,2	9	9	26	28	28,2	12	9,3	9,3
Termómetro 2 (°C)	8,2	9,2	9,1	25	27,5	28	12,1	9,1	9,4
Termómetro 3 (°C)	8	8,8	9	26	28	28	12,5	9	9
Día 3									
Termómetro 1 (°C)	10	9,8	10,1	25	25	24,5	11,5	11	10,8
Termómetro 2 (°C)	10	9,8	10	24,8	25,2	24	11,5	11,1	10,6
Termómetro 3 (°C)	9,8	8,9	10,1	24,6	24,9	24	11	10,1	10
Día 4									
Termómetro 1 (°C)	7,2	7,2	7,5	20,5	20	21,5	11	10,7	10,5
Termómetro 2 (°C)	7,1	7	7,3	20,5	20	21,4	11,1	10,5	10,5
Termómetro 3 (°C)	6,4	6,8	7,3	20,5	20	19,5	10,8	10,5	10

Realizado por: ARCOS, Daniel. 2017.

Se puede observar que las temperaturas obtenidas durante este periodo para las zonas uno y dos, la variación es mínima. Con respecto a la zona 3 la temperatura varía en un grado en relación a las 2 zonas. Concluyendo que en la zona baja el frío del viento es más fuerte que en la zona alta.

Por este motivo, se utilizará dos módulos que supervisen el ambiente. El primero cubrirá las 2 zonas y el segundo la parte inferior del huerto.

3.2 Diseño e instalación de los aspersores y actuadores.

Para la instalación del riego se utiliza aspersores que cubran un diámetro de 16 m. Ahora para cubrir el área de aspersión total del terreno de 785 m² se utilizó 4 aspersores. En la figura 2-3 se observa el diseño de la ubicación de los aspersores

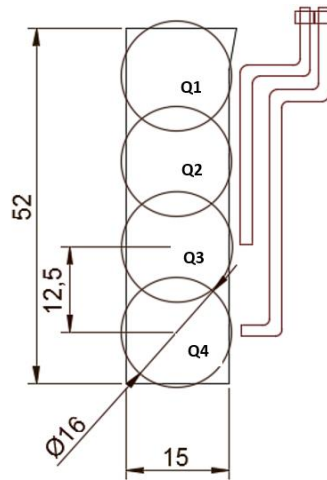


Figura 2-3: Ubicación de los aspersores.
Realizado por: ARCOS, Daniel. 2017

El área que no llega a cubrir los aspersores es aproximadamente 20 m², valor que no es representativo de acuerdo al área total del terreno

3.2.1 Instalación de aspersores

Luego del diseño de la ubicación de los aspersores se procedió a instalación de los mismos el cual se aprecia en la siguiente figura 3-3.



Figura 3-3: Instalación de aspersores
Realizado por: ARCOS, Daniel. 2017

3.2.2 Instalación de electroválvulas y bomba eléctrica

Para el riego de aspersión utilizado en las heladas y riego por goteo, se necesitan tres electroválvulas, las cuales se activarán de acuerdo a lo programado en el bloque de procesamiento. Dos electroválvulas son utilizadas para la activación de los aspersores, y una para el riego a goteo. En la figura 4-3 se puede apreciar la conexión de estas electroválvulas.



Figura 4-3: Instalación electroválvulas
Realizado por: ARCOS, Daniel. 2017

Es necesario la activación de una bomba de agua eléctrica que ayude a los riegos del huerto, en la figura 5-3 se ilustra las características de la bomba



Figura 5-3: Bomba de agua eléctrica
Realizado por: ARCOS, Daniel. 2017

3.3 Instalación de los Módulos de Supervisión.

Los módulos de Supervisión fueron instalados como se muestra en la siguiente figura 6-3, se especifica las distancias respectivas de cada módulo. Las iniciales MR son módulos de riego de suelo y MA módulos de supervisión Ambiente.

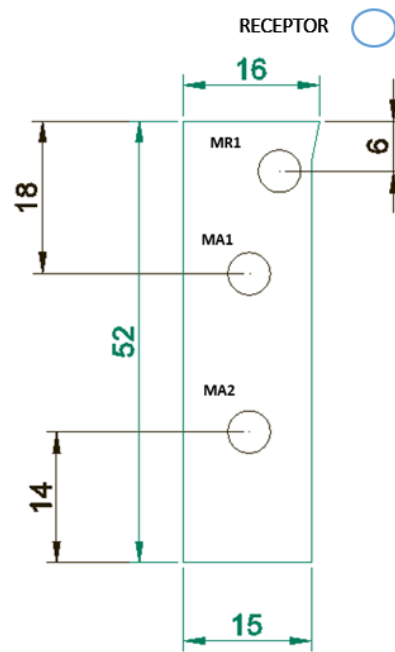


Figura 6-3: Ubicación de los módulos de Supervisión
Realizado por: ARCOS, Daniel. 2017

3.3.1 Módulo de Suelo

El módulo de supervisión de suelo se le ubicó los más cercano al módulo de recepción, ya que de acuerdo al estudio realizado en el capítulo uno, la humedad de las camas no tiene mucha variación de acuerdo a la distancia, en este caso se lo ubicó cerca del módulo de recepción, en la figura 7-3 se aprecia la instalación.



Figura 7-3: Ubicación Módulo Suelo
Realizado por: ARCOS, Daniel. 2017

3.3.1.1 Ubicación de los sensores de humedad

Los sensores fueron ubicados dos en la parte superior, dos a 30cm verticalmente en la parte inferior, ya que las raíces crecen a esa profundidad. En la siguiente figura 8-3, se aprecia la colocación de cada uno de ellos.

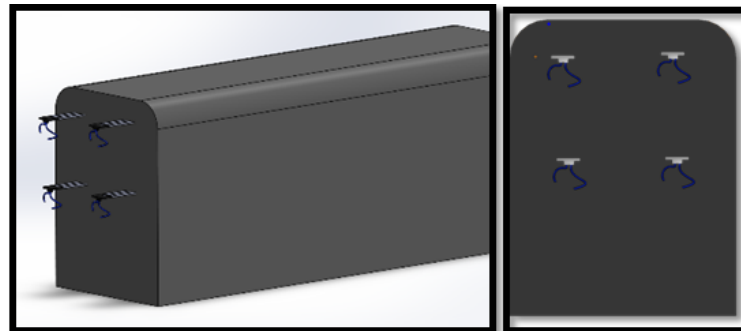


Figura 8-3: Ubicación de sensores de humedad
Realizado por: ARCOS, Daniel. 2017

De la misma forma que se observa en la figura anterior se procedió a la instalación en la cama de fresas, la figura 9-3 se muestra la ubicación de los sensores de humedad FC-28.



Figura 9-3: Instalación de sensores en la cama de fresa
Realizado por: ARCOS, Daniel. 2017

3.3.2 Módulo Supervisión Ambiente

En la figura 10-3 se observa la ubicación del primer módulo ambiente, se instaló para cubrir la parte superior del terreno, como se explica en la figura 3-6. El sensor de humedad y temperatura debe estar a una altura de 1.5m desde el suelo, para captar de mejor manera las variables ambientales.



Figura 10-3: Instalación módulo de
Supervisión Ambiente 1
Realizado por: ARCOS, Daniel. 2017

En la figura 11-3 se aprecia la ubicación del siguiente módulo cubre la zona inferior del terreno por la topología del mismo. Ubicado a 20m del primer módulo, y ubicado a una distancia de 52 m del receptor.



Figura 11-3: Ubicación módulo Supervisión Ambiente 2.

Realizado por: ARCOS, Daniel. 2017

3.3.3 Ubicación de los paneles solares.

Los paneles solares para que tenga una mejor captación de la energía, deben estar ubicados en un ángulo de 15° como se muestra en la figura 12-3, y es recomendable que tenga una orientación hacia el sur o norte. (Crespo & Quimbita. 2016).

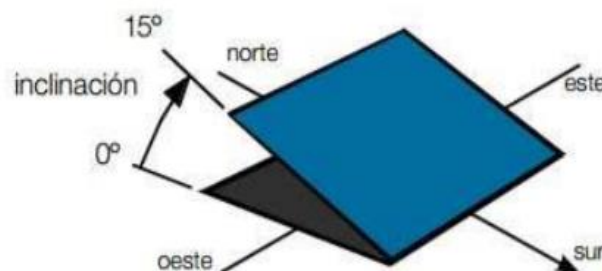


Figura 12-3: Angulo de inclinación para paneles solares

Realizado por: CRESPO, Marcos & QUIMBITA, Roberto. 2016

Estos se ubicaron en la cubierta que protege a los módulos, quedando como se muestra en la figura 13-3.



Figura 13-3 Instalación panel solar
Realizado por: ARCOS, Daniel. 2017

3.4 Pruebas de transmisión.

El sistema tiene la caracterización de su transmisión inalámbrica por medio de la comunicación de radiofrecuencia, el envío de datos sin pérdida de información es esencial para el correcto funcionamiento del sistema.

Las pruebas de transmisión se lo realizaron con el fin de determinar la distancia real de transmisión de cada uno de los módulos, con las antenas de 17cm para los módulos de supervisión y módulo de procesamiento. Después de varias pruebas, se determinó que la distancia máxima entre los emisores y el receptor es de 60m, alimentados con 5V. Para tener una idea de la distancia que llegaría si se variaría el voltaje en la tabla 2-3 veremos los resultados de distancia.

Tabla 2-3: Distancia de transmisión módulos RF.

Voltaje	Distancia Máxima
5V	60 m
7.3V	95m
9V	130m
12V	175m

Realizado por: ARCOS, Daniel. 2017

3.5 Pruebas de los sensores.

Para la comprobación del correcto funcionamiento de los módulos de supervisión, se realizaron pruebas de medición de todos los sensores del sistema, como alcance de transición, e integridad de los datos.

3.5.1 Comparación de mediciones sensor de ambiente

La comparación de los datos se lo realizó con un medidor de *temperatura e higrómetro analógico*, que se muestra en la figura 14-3, tiene un rango de temperatura de -20 a 50 °C, con un error de $\pm 2^{\circ}\text{C}$, con respecto a la humedad tiene un rango del 0-100% con un error de $\pm 3\%$.

Estas pruebas consisten en comparar las medidas entre el equipo de la figura y los sensores instalados en el sistema, así establecer el error lo que permitirá determinar si el equipo implementado tiene un comportamiento similar con el equipo comparado.

Las mediciones se lo realizaron durante un día y se tomaron 20 muestras para realizar las pruebas de los sensores.



Figura 14-3: Termómetro – Higrómetro
Realizado por: ARCOS, Daniel. 2017

3.5.1.1 Sensor de temperatura

El sensor DHT21 tiene un rango de -40 a 80 °C para temperatura, con una exactitud del $\pm 0.5^{\circ}\text{C}$. Esta prueba consiste en comprobar si el sensor de temperatura no incorpora más error del introducido por el sensor de temperatura que tiene un máximo de 2.5°C. En la tabla 3-3 se aprecia

que el valor absoluto máximo es de $\pm 1.2^{\circ}\text{C}$, valor que satisface los requerimientos planteados para el sistema.

Tabla 3-3: Mediciones error absoluto sensor de temperatura ambiente

Numero de Muestras	Termómetro (°C)	Módulo de supervisión (°C)	Error Absoluto (°C)
1	12.5	11.5	1
2	13.2	12.5	0.7
3	13.5	13	0.5
4	14.3	13.7	0.6
5	14	13.1	0.9
6	14.5	13.8	0.7
7	14.8	15.2	-0.4
8	14	14.8	-0.8
9	14.7	14	0.7
10	15.5	15	0.5
11	16.5	17	-0.5
12	16.6	16.4	0.2
13	16.9	17.5	-0.6
14	17.3	17	0.3
15	17.5	18	-0.5
16	18	19.2	-1.2
17	18.8	19.5	-0.7
18	19	20	-1
19	20.5	21.3	-0.8
20	22.6	23.8	-1.2

Realizado por: ARCOS, Daniel. 2017

3.5.1.2 Sensor de humedad relativa

De la misma forma el sensor DHT21 puede medir la humedad relativa, la cual tiene un rango de 0-100% y trabaja con una exactitud de 3%. El error máximo de humedad debería ser de 8%. En la tabla 4-3 se observa que tiene un error absoluto de $\pm 4\%$.

Tabla 4-3: Mediciones error absoluto sensor de humedad ambiente

Numero de Muestras	Higrómetro (%)	Módulo de supervisión (%)	Error Absoluto (%)
1	75	78	-3
2	74	78	-4
3	74	77	-4
4	72	75	-3
5	76	76	0
6	70	73	-3
7	72	76	-4
8	73	72	1
9	75	76	-1
10	77	75	2
11	73	76	-3
12	70	73	-3
13	68	70	-2
14	63	62	1
15	62	60	2
16	63	65	-2
17	60	63	-3
18	58	60	-2
19	56	58	-2
20	55	57	-2

Realizado por: ARCOS, Daniel. 2017

3.5.2 Pruebas de estabilidad de los sensores.

Esta prueba consiste en tomar muestras para determinar la desviación estándar y el coeficiente de variación de los datos obtenidos por los sensores.

3.5.2.1 Repetitividad del módulo de Supervisión Ambiente

Para estas pruebas se tomó muestras en un tiempo de 30 minutos, se lo guardó en una tarjeta de SD, en un archivo .txt. Con el registro de estos valores de los sensores de humedad y temperatura se calculó el coeficiente de desviación, la media y la desviación estándar, con el fin de probar la estabilidad del sistema. Los datos obtenidos se pueden ver en la tabla 5-3.

Tabla 5-3: Prueba repetitividad módulo de Supervisión Ambiente

Numero de Muestras	Hora	Valor Humedad (%)	Valor Temperatura (°C)
1	9:20:16	70	14
2	9:24:01	70	14
3	9:26:09	69	14
4	9:28:07	69	14
5	9:33:53	69	14
6	9:35:13	69	14
7	9:38:55	70	14
8	9:41:18	70	14
9	9:42:18	70	14
10	9:43:19	70	14
11	9:44:20	70	14
12	9:45:21	69	14
13	9:47:23	70	14
14	9:47:59	71	14
15	9:49:01	70	14
16	9:49:55	71	14
17	9:50:20	70	14
18	9:51:25	69	14
19	9:52:10	70	14
20	9:53:19	71	15
Media		69,69	14,05
Desviación Estándar		0,67	0,22
Coefficiente de Variación		0,96%	0%

Realizado por: ARCOS, Daniel. 2017

Los resultados muestran un coeficiente de variación menos del 1%, valor nada representativo con respecto al rango de medida, señalado así por (Marconi, 2009). Lo cual indica que el sensor es estable para su funcionamiento.

3.5.2.2 Pruebas sensores de Suelo.

De la misma forma se tomó datos de los cuatro sensores de humedad ubicados en el suelo para saber su estabilidad, la muestra se lo tomó durante 15 minutos, los resultados obtenidos se muestran en la tabla 6-3.

Tabla 6-3: Prueba repetitividad módulo de Supervisión de Suelo

Módulo Supervisión Suelo					
Numero de Muestras	Hora	SENSORES DE HUMEDAD			
		Inferior 1 (%)	Inferior 2 (%)	Superior 1 (%)	Superior 2 (%)
1	16:15:00	85	85	84	84
2	16:15:05	85	85	84	84
3	16:16:30	85	86	84	84
4	16:17:15	85	85	84	84
5	16:18:10	85	85	84	84
6	16:19:14	85	85	85	84
7	16:20:16	86	86	84	85
8	16:21:18	86	85	85	84
9	16:22:21	86	86	85	85
10	16:22:55	86	86	85	84
11	16:23:27	86	86	85	85
12	16:24:32	86	86	85	85
13	16:25:35	86	86	85	85
14	16:26:05	86	86	85	85
15	16:26:44	86	86	85	85
16	16:27:47	86	86	85	85
17	16:28:52	86	86	85	85
18	16:29:40	86	86	85	85
19	16:29:57	86	86	85	85
20	16:30:00	86	86	85	85
Media		85,63	85,63	84,63	84,50
Desviación Estándar		0,47	0,47	0,47	0,50
Coefficiente de Variación		0,55%	0,55%	0,56%	0,59%

Realizado por: ARCOS, Daniel. 2017

En la tabla se puede observar que se tiene un coeficiente de variación mínimo ya que los valores de la media son aproximadamente de 70, en el caso de la humedad están en perfectas condiciones de funcionamiento, ya que es muy estable.

3.5.3 Pruebas de comprobación de helada.

Para estas pruebas se simuló una helada, al bajar intencionalmente la temperatura a menos de 0 °C al sensor utilizando hielo seco, y así comprobar el funcionamiento de los módulos y la activación de los aspersores. Al bajar la helada menos de cero grados, los aspersores de acuerdo a la zona del módulo se activan exitosamente como se observa en la figura 15-3.



Figura 15-3: Activación de riego por aspersion
Realizado por: ARCOS, Daniel. 2017

3.5.4 Pruebas de humedad de suelo.

Al instalar el módulo de supervisión en la cama de fresas, se esperó un tiempo para observar los resultados, y ver que se active de forma correcta el riego por aspersion. Al tener una humedad menor al 77 % en el suelo, la electroválvula se activó correctamente y se procedió al riego que lo caracteriza, como se ve en la figura 16-3.



Figura 16-3: Activación de riego por goteo
Realizado por: ARCOS, Daniel. 2017

3.6 Pruebas de funcionamiento Módulo de Recepción

Este es el principal módulo del sistema electrónico, ya que es el cual llega la información de todos los módulos de supervisión, el funcionamiento consta del almacenamiento y visualización de los datos captados en el huerto de fresas, para posteriormente activar los diferentes riegos que consta el sistema, en la figura se observa este módulo.



Figura 17-3: Módulo de Recepción
Realizado por: ARCOS, Daniel. 2017

3.6.1 Almacenamiento de datos

En la siguiente figura 18-3 se puede ver los datos recibidos por los 3 módulos de supervisión, en el documento de texto, los cuales muestran los datos obtenidos en tiempo real, con la hora y fecha de monitoreo.

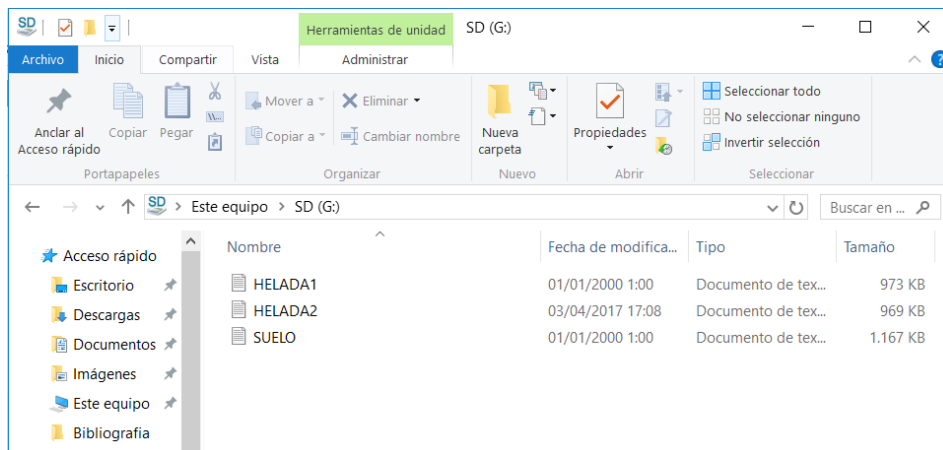


Figura 18-3: Archivos de texto de los módulos de supervisión
Realizado por: ARCOS, Daniel. 2017

Dentro del archivo los datos se registran de manera ordenada, con la fecha y hora receptada, y todos los datos que se monitorean en el huerto de fresas, en la figura 19-3 se detalla la información.

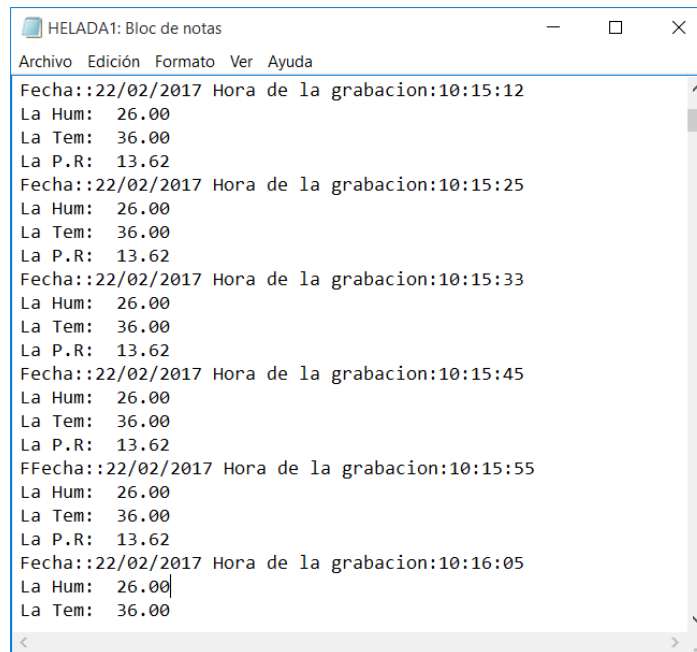


Figura 19-3: Archivo de texto con los datos de supervisión
Realizado por: ARCOS, Daniel. 2017

3.7 Menú de usuario LCD

La LCD está programado un menú en el cual el usuario puede interactuar de mejor manera con el sistema, figura 20-3, en el cual se puede igualar la hora, ver los datos de monitoreo de los módulos, y activación manual de los dos riegos instalados.



Figura 20-3: Menú de Usuario
Realizado por: ARCOS, Daniel. 2017

3.7.1 *Igualar fecha.*

En esta opción se iguala la fecha y hora, digitando cada uno de los valores desde el teclado matricial, figura 21-3.



Figura 21-3: Insertar fecha y hora
Realizado por: ARCOS, Daniel. 2017

3.7.2 *Visualización de los datos.*

Al entrar a la opción número dos del menú, se visualiza los datos de humedad, temperatura y punto de rocío de los módulos de supervisión ambiente, figura 22-3.



Figura 22-3: Visualización de datos
Realizado por: ARCOS, Daniel. 2017

3.7.3 *Riego Manual.*

En la siguiente opción, figura 23-3, el usuario puede activar y desactivar a su conveniencia los riegos por aspersión y por goteo.



Figura 23-3: Control Manual de Riegos
Realizado por: ARCOS, Daniel. 2017

3.8 Análisis de costos de la implementación del Sistema.

En la siguiente tabla 7-3 muestra los precios de los elementos utilizados para el diseño y la instalación del sistema.

Tabla 7-3: Costo del sistema electrónico

MÓDULO	DISPOSITIVOS	CANTIDAD	C/U (\$)	Costo Total
MÓDULOS SUPERVISIÓN AMBIENTE	Arduino Pro-Mini	2	6	12
	Sensor DHT21	2	12	24
	Batería y Panel solar	2	16	32
	Módulo RF emisor	2	3	6
	Elementos electrónicos y cables	2	5	10
	Instalación	2	10	20
	TOTAL PRECIO MÓDULO AMBIENTE			
MÓDULO DE SUELO	Arduino Pro-Mini	1	6	6
	Sensor FC-28	4	4,5	18
	Batería y Panel solar	1	16	16
	Módulo RF emisor	1	3	3
	Elementos electrónicos y cables	1	5	5
	Instalación	1	10	10
	TOTAL PRECIO MÓDULO SUELO			
MÓDULO DE RECEPCIÓN	Arduino Mega	1	20	20
	Módulo Reloj RTC	1	6,5	6,5
	Módulo SD Card	1	5	5
	Pantalla LCD 20X4	1	14	14
	Cables de conexión	1	5	5
	Módulo RF receptor	1	3,5	3,5
	Instalación	1	10	10
TOTAL PRECIO MÓDULO RECEPCIÓN				54
BLOQUE DE POTENCIA	Electroválvulas	3	31	93
	Aspersores	4	5	20
	Módulos de relés	1	12	12
	Transformador 24V	1	18	18
	Disyuntor doble	1	5	5
	Caja de protección	1	22	22
	Enchufe eléctrico	1	0,5	0,5
	Cables de conexión	1	15	15
	Instalación	1	10	10
TOTAL PRECIO M. POTENCIA				165,5
TOTAL				371,5

Realizado por: ARCOS, Daniel. 2017

En el terreno de prueba que consta de 785m² y según el propietario el coste de inversión en esta expansión fue de \$ 2500,00 (Anexo 3). Para la implementación del sistema electrónico se necesita \$ 371,50, que representa el 14.86% de la inversión inicial, valor que no es muy representativo para la instalación en los huertos de fresas.

CONCLUSIONES

- Se diseñó e implementó un sistema electrónico para la supervisión, control de riego y helada para huertos de fresa con red de sensores y hardware libre utilizando tecnología de radio frecuencia de 433Mhz para la trasmisión de datos
- Fue necesario la implementación de un solo módulo de supervisión de suelo para el riego del huerto, ya que las fresas están plantadas en camas con plásticos negros, en donde la humedad del suelo es equitativa en todas las camas.
- El tiempo de respuesta para la activación de los sistemas de riego es de 15 segundos.
- Es necesario un estudio del terreno en donde se instalará el sistema electrónico, ya que las topologías de estos, determinan cuantos sensores de supervisión ambiente deberían instalar en el huerto.
- Los sensores de humedad y temperatura utilizados cumplen con los criterios de funcionamiento adecuados ya que contienen errores absolutos de $\pm 1.2^{\circ}\text{C}$ en temperatura y 4% en humedad, que son bajos con respecto a sus valores máximos.
- De acuerdo a las pruebas realizadas de transmisión de datos con antenas de 17cm se obtuvo una distancia de 60m con una alimentación de 5V, sin pérdida de información.

RECOMENDACIONES

- Incrementar al sistema más sensores como: ph, radiación solar, temperatura etc. que permitirán un mejor control de las propiedades del suelo.
- Incorporar al sistema una plataforma WEB, en donde se almacene los datos con el fin de que especialistas analicen los datos que se supervisa en los huertos.
- Se recomienda incorporar un sistema de aviso mediante GSM al usuario, que envíe datos relevantes del huerto.
- Realizar estudios en otros tipos de cultivos para la posible implementación del sistema diseñado.

BIBLIOGRAFIA

1. **ARDUINO.** *Placas Arduino.* [Blog]. [Consulta: 25 de Agosto 2016]. Disponible en: <https://www.arduino.cc>
2. **AOSONG.** “*Temperature and humidity module AM2301 Product Manual*” Kropochev. [En Línea]. [Consulta: 05 de Diciembre 2016]. Disponible en: <https://kropochev.com/downloads/humidity/AM2301.pdf>
3. **BAUDILIO, J.** *Como cultivar Fresas, Fresones y Tomates.* 2ª ed. Barcelona-España: Aedos Barcelona, 1977, p 24.
4. **BOLAÑOS, D.** *Módulos de RF para Radio Control* [pdf], 2011. [Consulta: 20 de Febrero 2017]. Disponible en: http://www.bolanosdj.com.ar/TEORIA/MODULOS_DE_RF.pdf
5. **CASTELLANOS, Elvis.** *Beneficios de la Tecnología a la Agricultura.* [Blog]. 2013. [Consultado: 22 de Diciembre 2016]. Disponible en: <http://tecnologiaactualidadpositivoynegativo.blogspot.com/2013/01/beneficio-de-la-tecnologia-la.html>
6. **CRESPO PAREDES, Marcos Antonio; & QUIMBITA PACHUCO, Roberto Xavier.** *Implementación de un sistema híbrido de 1000 WATTS mediante el uso de energía eólica y solar para el electrificado de una vivienda de la hacienda Tambo.* [En línea]. (Tesis pregrado) Universidad Técnica de Cotopaxi. Unidad Académica de Ciencias de la Ingeniería y Aplicadas. Carrera de Ingeniería Electromecánica. Latacunga-Ecuador. 2016. P 11. Disponible en: <http://repositorio.utc.edu.ec/bitstream/27000/3664/1/T-UTC-000019.pdf>
7. **EAMES, Alex.** *Raspberry Pi Family Photo updated to include.* [Blog]. 2016. [Consulta: 20 de Diciembre 2016]. Disponible en: <http://raspi.tv/2016/raspberry-pi-family-photo-updated-to-include-pi3b-29-feb-2016>
8. **ECURED.** *Agricultura de precisión.* [En línea], Habana – Cuba. Mayte García Vega. 2012. [Consultado: 15 Agosto 2016]. Disponible en: https://www.ecured.cu/Agricultura_de_Precisi%C3%B3n

9. **ECURED.** *Riego por Aspersión.* [En línea]. Habana – Cuba. Minagri, Victor. 2013. [Consultado: 18 de Septiembre 2016]. Disponible en: https://www.ecured.cu/Riego_por_Aspersi%C3%B3n.
10. **FERNÁNDEZ MARTINEZ, Joaquín Ordieres.** *Redes inalámbricas de sensores: teoría y aplicación.* [pdf]. La Rioja – España: Universidad de La Rioja. 2012. [Consulta: 05 de Enero 2017]. Disponible en: <http://dialnet.unirioja.es/descarga/libro/377564.pdf>.
11. **FERNANDEZ, José Francisco.** *Uso de Sensores en agricultura.* [Blog]. 2016. [Consulta: 30 de Diciembre 2017]. Disponible en: <http://www.iagua.es/blogs/iriego/uso-sensores-agricultura>
12. **GASCÓN, David.** *Redes de Sensores Inalámbricos, la tecnología invisible.* [pdf], España, 2010. [Consulta: 15 de Octubre 2016]. Disponible en: <http://www.libelium.com/libelium-downloads/libelium-bit-coit.pdf>
13. **GARCÉS HERNANDEZ, David Rafael.** *Sistema de control y monitoreo del invernadero del centro experimental del riego de la Epoch mediante tecnología zigbee.* (Tesis de pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería En Telecomunicaciones y Redes. Riobamba-Ecuador. 2012. P 12.
14. **GAREMAR, S.** *Historia de Arduino.* [Blog]. 2015. [Consulta: 12 Diciembre 2016]. Disponible en: <http://industrino.es/historia-de-arduino/>
15. **GARCÍA GONZÁLEZ, Antony.** *Arduino.* [En línea] 2015. [Consulta: 24 de Diciembre de 2016.] Disponible en: <http://panamahitek.com/el-setup-y-el-loop-en-arduino/>
16. **GARCÍA, Luis Miguel.** *Que es Waspote.* [Blog]. 2012. [Consulta: 20 Diciembre 2016]. Disponible en: <https://unpocodejava.wordpress.com/2012/08/21/que-es-waspote/>
17. **INGENIO TRIANA.** *Arduino Industrial.* [Blog]. 2016. [Consulta: 20 Diciembre 2016] Disponible en: <http://ingenio-triana.blogspot.com/2016/07/arduino-industrial-el-open-source.html>

18. **LA NACION.** *Optimización en la aplicación de fertilizantes.* [blog]. 2006. [Consulta: 10 de Agosto 2016]. Disponible en: <http://www.lanacion.com.ar/846676-optimizacion-en-la-aplicacion-de-fertilizantes>
19. **MARCONI, Liliana; & D'AMELIO, Adriana.** *Medidas de Tendencia Central y de Variabilidad.* [pdf] 2009. [Citado el: 08 de Abril de 2016.] Disponible en: <http://www.deie.mendoza.gov.ar/aem/material/teoria/MEDIDAS%20DE%20TENDENCIA%20CENTRAL%20Y%20DE%20VARIABILIDAD.pdf>
20. **MARTINEZ, Leoncio.** *Control de heladas* [pdf], Chile, 2007. [Consulta: 25 de Febrero 2017]. Disponible en: <http://www2.inia.cl/medios/biblioteca/ta/NR34738.pdf>
21. **MORIN, Charles.** *Cultivo de Cítricos.* 2ª ed. Lima-Perú: Instituto Interamericano de Ciencias, 1980 p 188.
22. **MUÑOZ FUEREZ, Oscar Rene.** *Sistema automático de riego por goteo para cultivo de fresas artesanal.* [En línea]. (Tesis de pregrado). Universidad Técnica del Norte. Facultad de Ingeniería en Ciencias Aplicadas, Escuela de Ingeniería en Macatrónica. Ibarra – Ecuador. 2014. p. 1. [Consulta: 2016-09-01]. Disponible en: <http://repositorio.utn.edu.ec/handle/123456789/3744>
23. **OROZCO, Mónica.** “*Las heladas se mitigan de 4 maneras*” [En línea]. El Comercio, 2012. [Consulta: 22 de Abril 2017]. Disponible en: <http://www.elcomercio.com/actualidad/negocios/helada-se-mitiga-de-maneras.html>.
24. **REVISTA EL AGRO.** *Agricultores le apuestan al cultivo de fresas* [En línea]. El agro, 2016. [Consulta: 20 de Agosto 2016]. Disponible en: <http://www.revistaelagro.com/agricultores-le-apuestan-al-cultivo-de-fresas/>
25. **ROBINSON, J.** *Protege tus cultivos de las heladas.* [blog]. Estados Unidos, 2011. [Consultado: 10 de Febrero 2017]. Disponible en: <http://www.hortalizas.com/proteccion-de-cultivos/protege-tus-cultivos-de-las-heladas/>

26. **ROJAS MONTECINO, Patricio.** *La vitivinicultura y su proceso de expansión en ladera en el valle del Lontué, séptima región del Maule.* [En línea]. (Tesis de pregrado). Universidad de Chile. Facultad de Arquitectura y Urbanismo. Departamento de Geografía. (Santiago - Chile). 2005. p. 41. [Consultado: 09 de Febrero 2017]. Disponible en: http://repositorio.uchile.cl/tesis/uchile/2005/rojas_p/sources/rojas_p.pdf

27. **SANTIBAÑEZ, F.** *Estudio agroclimático proyecto Maipo.* Santiago de Chile – Chile. Laboratorio de Agroclimatología. Universidad de Chile, 1987.

28. **SNYDER L, Richard & MELO-ABREU J, Paulo.** “*Protección contra las heladas: fundamentos, práctica y economía*”. Italia-Roma: Fiat Pans, 2010, pp. 3-201.

29. **TEOREMA AMBIENTAL.** *Seis ventajas del uso de plásticos en agricultura* [En línea]. México: Miguel de Mendoza, 2002. [Consulta: 20 de Agosto 2016]. Disponible en: <http://www.teorema.com.mx/cienciaytecnologia/seis-ventajas-del-uso-de-plasticos-en-agricultura/>

30. **TECMIKRO.** “*Radiofrecuencia: Sistema TX/RX a 433MHz*”. [En línea]. Programación de microcontroladores, 2016. [Consulta: 25 de Agosto de 2016]. Disponible en: <http://programarpicenc.com/articulos/radiofrecuencia-sistema-tx-rx-a-433mhz/>

31. **UNDURANGA, Pablo; & VARGAS, Sigrid.** *Manual de frutilla.* [pdf]. Chillán-Chile, 2013. [Consulta: 15 de Diciembre 2016]. Disponible en: <http://biblioteca.inia.cl/medios/quilamapu/boletines/NR39084.pdf>

32. **VELASCO, J.** *Intel Galileo: kit de desarrollo de Intel compatible con Arduino y destinado a la educación.* [Blog]. 2013. [Consulta: 28 de Diciembre 2016]. Disponible en: <https://hipertextual.com/2013/10/intel-galileo-arduino-educacion>

33. **VERMON.** *Sensores en Agricultura y Ganadería.* [En línea]. Madrid – España. [Consulta: 01 de Enero 2016]. Disponible: <http://www.grupovermon.com/sensores-en-agricultura-y-ganaderia/>

ANEXOS.

ANEXO 1: Programación en Arduino del sistema.

PROGRAMACIÓN MODULO DE PROCESAMIENTO.

```

//      ESCUELA SUPERIOR POLITECNICA DE CHOMBORAZO
//      FACULTAD DE INFORATICA Y ELECTRONICA
// ESCUELA DE INGENIERIA ELECTRONICA EN CONTROL Y REDES INDUSTRIALES
//      //      TESIS DE GRADO
//      //      AUTOR: DANIEL ARCOS
//      //      TESIS

//-----CONDICIONES LCD-----
---

#include <LiquidCrystal.h>

//----- Inicializa o LCD-----
----
LiquidCrystal lcd(40, 41, 42, 43, 44, 45); // RS, E, D3, D4, D5, D7
#include <Wire.h> //librería para el manejo del protocolo

#include <VirtualWire.h> // LIBRERIA PARA LOS RF

//-----CONDICIONES TECLADO-----
-----

#include <Keypad.h>

// CONDICIONES RELOJ
#include "stdio.h"
#include "RTCLib.h" //libreria del reloj tiempo
#include "stdio.h"

RTC_DS1307 RTC; //función propia de la librería RTC para poder configurar los datos internos

-----// CONDICIONES SD
#include <SPI.h>
#include <SD.h>

-----// VARIABLES A UTILIZAR-----
const byte ROWS = 4; //four rows
const byte COLS = 3; //FOUR columns

int x =0;
int y =0;
int z =0;
int j=0;
int l=0;

char tecla;
```

```

char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}
};

byte rowPins[ROWS] = {28, 26, 24, 22};
byte colPins[COLS] = {34, 32, 30};

Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

//----- VARIABLES A UTILIZAR SENSORES-----
----

float t=0;          // TEMPERATURA AMBIENTE
float h=0;          // HUEMDAD AMBIENTE

float t2=0;        // TEMPERATURA AMBIENTE II
float h2=0;        // HUEMDAD AMBIENTE II

int hs1=0;         // HUMEDAD SENSOR SUPERIOR1
int hs2=0;         // HUMEDAD SENSOR SUPERIOR2
int hi1=0;         // HUMEDAD SENSOR INFERIOR1
int hi2=0;         // HUMEDAD SENSOR INFERIOR2

float pr;          // PUNTO DE ROCIO
float pr2;         // PUNTO DE ROCIO

int bomba=14;
int valvulaH=15;
int valvulaH2=16;
int valvulaR=17;

int humedadmin=50;
int humedadmax=90;
int temperaturamax=30;

int manual=0;      // VARIABLE PARA ACTIVAR RIEGO MANUAL

//-----VARIABLES PARA IGUALAR RELOJ-----
----

int cont=0;
int cont1=0.;
int cont2=0;
int cont3=0;
int cont4=0;
int cont5=0;

int aux=0;

```

```

int aux1=0;
char anos[4];
char mess[4];
char dias[4];
char horas[8];
char minutos[2];
char segundos [2];

int hora;
int minuto;
int segundo;
int ano;
int mes;
int dia;

String cadena="";

int ci=0;

const int chipSelect = 53;
int Guardado = 0;
int escritura = 0;

int Guardados = 0;
int escrituras = 0;
//*****

//-----CONDICIONES SENSORES-----
--

void setup(){

Serial.begin(9600);

lcd.begin(20, 4);

//----- INICIALIZAR RF-----
----
vw_setup(2000); // velocidad: Bits per segundo
vw_set_rx_pin(3); //Pin 3 como entrada del RF
vw_rx_start(); // Se inicia como receptor

//----- PINES DE ACTIVACION-----
----
pinMode(bomba, OUTPUT);
pinMode(valvulaH, OUTPUT);
pinMode(valvulaH2, OUTPUT);
pinMode(valvulaR, OUTPUT);

//----- CONDICIONES SD Y RELOJ-----
--

```

```

Wire.begin();
RTC.begin();

    pinMode(chipSelect, OUTPUT); // Definimos el módulo SD como salida

    // SD.begin(chipSelect); // Iniciamos el módulo SD

    if (!SD.begin(10)){
        //Serial.println("Error al leer la tarjeta.");
        return;
    }
    else{
        //Serial.println("Tarjeta OK..");
    }
}

//-----FUNCION PARA CALCULO DEL PUNTO DE ROCÍO-----
---
void rocio1()
{
float raiz;
float dh;
float hz;
hz=(float) h;

dh =(hz / 100);
raiz=pow((dh), (0.125));
pr=raiz*(112+(0.9*t))+(0.1*t)-112;
//Serial.print("PUNTO ROCIO");
//Serial.println(pr);
delay(10);
}

void rocio2()
{
float raiz2;
float dh2;
float hz2;
hz2=(float) h2;

dh2 =(hz2 / 100);
raiz2=pow((dh2), (0.125));
pr2=raiz2*(112+(0.9*t2))+(0.1*t2)-112;
//Serial.print("PUNTO ROCIO");
//Serial.println(pr2);
delay(10);
}
}

```

```
//-----ACTIVACION RIEGOS HELADA1-----  
----
```

```
void activacion_riego_z1(){  
    DateTime now = RTC.now();  
  
    rocio1();  
    if(((now.hour())>=20) || ((now.hour())<6))  
    {  
        if ((pr>0) && (t<=16))  
        {  
            digitalWrite(bomba, HIGH);  
            digitalWrite(valvulaH, HIGH);  
            delay(10);  
        }  
    }  
  
    if(((now.hour())>=6) || (t>16)){  
        digitalWrite(valvulaH, LOW);  
        delay(10);  
    }  
}
```

```
//-----ACTIVACION RIEGO HELADA2-----  
----
```

```
void activacion_riego_z2(){  
  
    DateTime now = RTC.now();  
  
    rocio2();  
    if(((now.hour())>=20) || ((now.hour())<6))  
    {  
        if ((pr2>0) && (t2<=16))  
        {  
            digitalWrite(bomba, HIGH);  
            digitalWrite(valvulaH2, HIGH);  
            delay(10);  
        }  
    }  
  
    if(((now.hour())>=6) || (t2>16)){  
        digitalWrite(valvulaH2, LOW);  
        delay(10);  
    }  
}
```

```
//-----ACTIVACIÓN DE RIEGO POR GOTEÓ-----  
----
```

```
void activacion_riego_suelo(){
```



```

if((hi1<77) && (hi2<77) && (hs1<77) && (hs2<77)){
digitalWrite(bomba, HIGH);
digitalWrite(valvulaR, HIGH);
delay(10);
}

if ((hi1>=90) && (hi2>=90) && (hs1>=90) && (hs2>=90)){
digitalWrite(valvulaR, LOW);
delay(10);
}
}

//-----VISUALIZACION PANTALLA LCD-----
----

void menu(){

lcd.setCursor(0,1);
lcd.print("1.Igualar hora");
lcd.setCursor(0,2);
lcd.print("2.Monitorio Huerto");
lcd.setCursor(0,3);
lcd.print("3.Riego Manual");

}

//-----CICLO DE REPETICIÓN-----
---

void loop(){

//-----RECEPCION DE DATOS POR RADIOFRECUENCIA-----
---
uint8_t buf[VW_MAX_MESSAGE_LEN];
uint8_t buflen = VW_MAX_MESSAGE_LEN;

if (vw_get_message((uint8_t *)buf,&buflen))
{
int i;
String DatoCadena="";
//Serial.println(buf);

//----- RECEPCIÓN DE HUMEDAD-----
-

//HUMEDAD DE SUELO INFERIOR 1

```

```

if((char)buf[0]=='a') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  hs1=DatoCadena.toInt();
  Serial.print("Humedad Suelo1: ");
  Serial.println(hs1);
}

delay (100);

// HUMEDAD DE SUELO INFERIOR 2
if((char)buf[0]=='b') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  hs2=DatoCadena.toInt();
  Serial.print("Humedad Suelo2: ");
  Serial.println(hs2);
}

// HUMEDAD DE SUELO SUPERIOR 1

if((char)buf[0]=='c') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }

  hi1=DatoCadena.toInt();
  Serial.print("Temperatura Suelo3: ");
  Serial.println(hs1);
}

delay (100);

// HUMEDAD DE SUELO SUPERIOR 2
if((char)buf[0]=='d') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  hi2=DatoCadena.toInt();
  Serial.print("Humedad Suelo4: ");
  Serial.println(hs2);
}

```

```

}

//----- RECEPCIÓN MODULO HUMEDAD TEMPERATURA AMBIENTE-----
-----

// HUEDAD AMBIENTE
if((char)buf[0]=='e') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  h=DatoCadena.toInt();
}

delay (10);

// TEMPERATURA AMBIENTE

if((char)buf[0]=='f') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  t=DatoCadena.toInt();
}

//-----RECEPCIÓN MODULO HUMEDAD TEMPERATURA AMBIENTE II-----
---

// HUEDAD AMBIENTE II
if((char)buf[0]=='g') //verificamos el inicio de trama
{
  for (i = 1; i < buflen; i++)
  {
    DatoCadena.concat((char)buf[i]);
  }
  h2=DatoCadena.toInt();
}

delay (10);

// TEMPERATURA AMBIENTE II

if((char)buf[0]=='h') //verificamos el inicio de trama
{

```

```

    for (i = 1; i < buflen; i++)
    {
    DatoCadena.concat((char)buf[i]);
    }

    t2=DatoCadena.toInt();

}
}

//-----IMPRESIÓN DE HORA Y FECHA-----

    lcd.clear();
    menu();

    DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

    lcd.setCursor(0,0);
    lcd.print(now.year(), DEC);
    lcd.print("/");
    //se Imprime el mes
    lcd.print(now.month(), DEC);
    lcd.print("/");
    //se Imprime el año
    // lcd.setCursor(0,4);
    lcd.print(now.day(), DEC);
    lcd.print(" ");

    lcd.setCursor(12,0);
    lcd.print(now.hour(), DEC);
    lcd.print(":");
    //se Imprime los minutos
    lcd.print(now.minute(), DEC);
    // lcd.print(":");
    //se Imprime los segundos
    //lcd.setCursor(16, 0);
    //lcd.print(now.second(), DEC);
    //lcd.print(" ");
    delay(200);
z=0;
y=0;
int ac=0;

//-----SELECCIÓN DE MENÚ DESDE TECLADO-----

//Tecla 1 Selección igualar reloj
tecla=customKeypad.getKey();

if(tecla) // Check for a valid key.
{
switch(tecla)
{
case '1': lcd.clear();
delay(10);

```

```

igualar_reloj());

break;

//Tecla 2 Monitoreo del huerto
case '2': lcd.clear();
lcd.clear();
lcd.setCursor (0, 0);
lcd.print("Seleccion Modulo");
delay(100);
lcd.setCursor (0, 1);
lcd.print("1.M. Helada Zona1");
lcd.setCursor (0, 2);
lcd.print("2.M. Helada Zona2");
lcd.setCursor (0, 3);
lcd.print("3.M. Riego");

//Tecla 3 Activación manual
monitoreo_huertos();
break;

case '3': // SELECCION PARA LEER SENSORE 1 Y 2
lcd.clear();
lcd.setCursor (1, 1);
lcd.print("***CONTROL MANUAL***");
delay(2000);
lcd.clear();
lcd.setCursor (0, 0);
lcd.print("Seleccione el Riego");
lcd.setCursor (5, 1);
lcd.print("a activar");
delay(100);
lcd.setCursor (0, 2);
lcd.print("1.Por aspercion");
lcd.setCursor (0, 3);
lcd.print("2.Por goteo");

Control_Manual();
break;

}
}

if(manual==0){
activacion_riego_z1();

activacion_riego_z2();

```

```

    activacion_riego_suelo();
}

//-----GUARDADO DE DATOS LLAMANDO A FUNCIONES-----

guardado_helada1();
delay(10);
guardado_helada2();
delay(10);
guardado_suelo();

}

//-----FUNCION GUARDADO DE SUELO-----

void guardado_suelo(){

    DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

    File dataFile = SD.open("Suelo.txt", FILE_WRITE); //Crea el archivo Datos Suelo o si ya existe
    escribe dentro de el (Podeis cambiarlo)

    if (dataFile)
    {
        //Lo que viene a continuación es lo que aparecerá dentro de la tarjeta SD.
        dataFile.print("Fecha:");dataFile.print(":"); dataFile.print(now.day(), DEC); dataFile.print("/");
        dataFile.print(now.month(), DEC); dataFile.print("/"); dataFile.print((now.year(), DEC));
        dataFile.print(" "); dataFile.print("Hora de la grabacion:"); dataFile.print(now.hour(), DEC);
dataFile.print(":");
        ;dataFile.print(now.minute(), DEC);dataFile.print(":"); dataFile.println(now.second(), DEC);
        dataFile.print("Humedad s1: "); dataFile.print(" "); dataFile.println(hs1);
        dataFile.print("Humedad s2: "); dataFile.print(" "); dataFile.println(hs2);
        dataFile.print("Humedad i1: "); dataFile.print(" "); dataFile.println(hi1);
        dataFile.print("Humedad i2: "); dataFile.print(" "); dataFile.println(hi2);

        //Serial.print("Grabacion automatica"); Serial.print(" ");Serial.print(" ");
        //Serial.println("efectuada correctamente"); Serial.println("MIRAR EN SD!!");
        dataFile.close();

        Guardados=0; //Ahora parte importante, ponemos la variable guardado a 0, para que cuando
volvamos a pulsar el botón, sea 1 y entre en este mismo if
        // de esta manera siempre va a entrar cada vez que pulsemos, porque siempre será 1.
        escrituras=0; // De esta manera evitaremos que se hagan más de un guardado en ese mismo
segund.
    }
}

```

```

    else{
        Serial.println("Error al escribir en valorpot.txt");
    }
    delay(100);
}

//-----FUNCIÓN GUARDADO DE DATOS DE HELADA M1-----
-

void guardado_helada1() {

    // RTC.adjust(DateTime(ano,mes,dia,hora,minuto,segundo));
    DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

    File dataFile = SD.open("Helada1.txt", FILE_WRITE); //Crea el archivo DHT11 o si ya existe
    escribe dentro de el (Podeis cambiarlo)

    if (dataFile)
    {
        //Lo que viene a continuación es lo que aparecerá dentro de la tarjeta SD.
        dataFile.print("Fecha:");dataFile.print(":"); dataFile.print(now.day(), DEC); dataFile.print("/");
        dataFile.print(now.month(), DEC); dataFile.print("/"); dataFile.print((now.year(), DEC));
        dataFile.print(" "); dataFile.print("Hora de la grabacion:"); dataFile.print(now.hour(), DEC);
dataFile.print(":");
        ;dataFile.print(now.minute(), DEC);dataFile.print(":"); dataFile.println(now.second(), DEC);
        dataFile.print("La Hum: "); dataFile.print(" "); dataFile.println(h);
        dataFile.print("La Tem: "); dataFile.print(" "); dataFile.println(t);
        dataFile.print("La P.R: "); dataFile.print(" "); dataFile.println(pr);

        dataFile.close();

        Guardado=0; //Ahora parte importante, ponemos la variable guardado a 0, para que cuando
volvamos a pulsar el botón, sea 1 y entre en este mismo if
        // de esta manera siempre va a entrar cada vez que pulsemos, porque siempre será 1.
        escritura=0; // De esta manera evitaremos que se hagan más de un guardado en ese mismo
segund.
    }

    else{
        //Serial.println("Error al escribir en valorpot.txt");
    }
    delay(100);
}

//-----GUARDADO DE DATOS DE HELADA M2-----

void guardado_helada2(){

```

```

// RTC.adjust(DateTime(ano,mes,dia,hora,minuto,segundo));
DateTime now = RTC.now(); // Obtiene la fecha y hora del RTC

// Serial.print(tm.Hour); Serial.print(":");Serial.print(tm.Minute);
// Serial.print(":"); Serial.println(tm.Second); delay(800); //Esto simplemente es para mostrar en el
video como van pasando los segundos.

File dataFile = SD.open("Helada2.txt", FILE_WRITE); //Crea el archivo o si ya existe escribe
dentro de el (Podeis cambiarlo)

if (dataFile)
{
//Lo que viene a continuación es lo que aparecerá dentro de la tarjeta SD.
dataFile.print("Fecha:");dataFile.print(":"); dataFile.print(now.day(), DEC); dataFile.print("/");
dataFile.print(now.month(), DEC); dataFile.print("/"); dataFile.print((now.year(), DEC));
dataFile.print(" "); dataFile.print("Hora de la grabacion:"); dataFile.print(now.hour(), DEC);
dataFile.print(":");
;dataFile.print(now.minute(), DEC);dataFile.print(":"); dataFile.println(now.second(), DEC);
dataFile.print("La Hum: "); dataFile.print(" "); dataFile.println(h2);
dataFile.print("La Tem: "); dataFile.print(" "); dataFile.println(t2);
dataFile.print("La P.R: "); dataFile.print(" "); dataFile.println(pr2);

dataFile.close();

Guardado=0; //Ahora parte importante, ponemos la variable guardado a 0, para que cuando
volvamos a pulsar el botón, sea 1 y entre en este mismo if
// de esta manera siempre va a entrar cada vez que pulsemos, porque siempre será 1.
escritura=0; // De esta manera evitaremos que se hagan más de un guardado en ese mismo
segund.
}

else{
// Serial.println("Error al escribir en valorpot.txt");
}
delay(100);
}

//-----FUNCIÓN PARA IGUALAR LA HORA MEDIANTE EL TECLADO-----
-----

void igualar_reloj ()
{
while(z<1)
{

```



```
RTC.adjust(DateTime(ano,mes,dia,hora,minuto,segundo));
```

```
    char customKey = customKeypad.getKey(); //se guarda en la variable customKey el caracter de la  
tecla presionada
```

```
    lcd.setCursor(0, 0);  
    lcd.print("Insertar:");  
    lcd.setCursor(0,1);  
    lcd.print("Año");  
    lcd.print(" ");  
    lcd.print("Mes");  
    lcd.print(" ");  
    lcd.print("Día");  
    lcd.print(" ");  
    lcd.print("Hor");  
    lcd.print(" ");  
    lcd.print("Min");
```

```
    lcd.setCursor(4,2);  
    lcd.print("/");  
    lcd.setCursor(7,2);  
    lcd.print("/");  
    lcd.setCursor(14,2);  
    lcd.print(":");
```

```
if (customKey)  
{  
    aux=aux+1;  
    if (aux<=4)  
    {  
        anos[cont]=customKey;    //se guarda caracter por caracter en el arreglo codigo[]  
        // Serial.println(anos[cont]); //se imprime en el puerto serie la tecla presionada  
        lcd.setCursor (cont, 2);  
        lcd.print(anos[cont]);  
        cont=cont+1;    //incrementamos la variable cont  
        if(cont==4)  
        {  
  
            lcd.setCursor (4, 2);  
            cont=0;  
  
            for (ci = 0; ci < 5; ci++)  
            {  
                cadena.concat((char)anos[ci]);  
            }  
  
            ano=cadena.toInt();  
        }  
    }  
}
```

```

if ((aux==6) || (aux==5))
{
    mess[cont]=customKey;    //se guarda caracter por caracter en el arreglo codigo[]
    //Serial.println(mess[cont]); //se imprime en el puerto serie la tecla presionada
    cont1=cont+5;
    lcd.setCursor (cont1, 2);
    lcd.print(mess[cont]);
    cont=cont+1;    //incrementamos la variable cont
    if(cont==2)
    {
        mes=mess[cont];
        lcd.setCursor (7, 2);
        cont=0;

        ci=0;
        String cadena1="";

        for (ci = 0; ci < 4; ci++)
        {
            cadena1.concat((char)mess[ci]);
        }

        mes=cadena1.toInt();

    }
}

if((aux==8) || (aux==7))
{
    dias[cont]=customKey;    //se guarda caracter por caracter en el arreglo codigo[]
    //Serial.println(mess[cont]); //se imprime en el puerto serie la tecla presionada
    cont2=cont+8;
    lcd.setCursor (cont2, 2);
    lcd.print(dias[cont]);
    cont=cont+1;    //incrementamos la variable cont
    if(cont==2)
    {
        dias[cont]=dia;
        lcd.setCursor (10, 2);
        cont=0;

        String cadena2="";
        for (ci = 0; ci < 4; ci++)
        {
            cadena2.concat((char)dias[ci]);
        }

        dia=cadena2.toInt();

    }
}

```

```

if((aux==9) || (aux==10))
{

horas[cont]=customKey;    //se guarda caracter por caracter en el arreglo codigo[]
//Serial.println(minutos[cont]); //se imprime en el puerto serie la tecla presionada
cont3=cont+12;
lcd.setCursor (cont3, 2);
lcd.print(horas[cont]);
cont=cont+1;    //incrementamos la variable cont
if(cont==2)
{

lcd.setCursor (14, 2);
cont=0;

String cadena3="";

for (ci = 0; ci < 4; ci++)
{
cadena3.concat((char)horas[ci]);
}

hora=cadena3.toInt();

}
}
if((aux==11) || (aux==12))
{
minutos[cont]=customKey;    //se guarda caracter por caracter en el arreglo codigo[]
//Serial.println(minutos[cont]); //se imprime en el puerto serie la tecla presionada
cont4=cont+15;
lcd.setCursor (cont4, 2);
lcd.print(minutos[cont]);
cont=cont+1;    //incrementamos la variable cont
if(cont==2)
{
minutos[cont]=minuto;
cont=0;

String cadena4="";

for (ci = 0; ci < 4; ci++)
{
cadena4.concat((char)minutos[ci]);
}

minuto=cadena4.toInt();
segundo=0;

}
}
if(aux==13){

```

```

if(customKey!='*')

z=1;

}
}
}
z=0;
}

//-----FUNCIÓN PARA EL CONTROL MANUAL-----

void Control_Manual(){

int ac;
while(z<1)
{
tecla=customKeypad.getKey();
delay(100);

if(tecla=='1'){
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("RIEGO ASPERCIÓN");
lcd.setCursor(0, 1);
lcd.print("1. Activar Riego");
lcd.setCursor(0, 2);
lcd.print("2. Desactivar Riego");
Activaciona();
int ac=0;
}

if(tecla=='2'){
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("**RIEGO GOTEO**");
lcd.setCursor(0, 1);
lcd.print("1.Activar Riego");
lcd.setCursor(0, 2);
lcd.print("2.Descativar Riego");
Activaciong();
int ac1=ac;
}

}
}

void Activaciona(){

```

```

while(y<1)
{

    tecla=customKeypad.getKey();
    if(tecla == '1'){
        lcd.clear();
        lcd.setCursor (1, 1);
        lcd.print("***Riego Activado***");
        act_esp1();
        act_esp2();

        delay(500);

        y=1;
        z=1;

        manual=1;
    }

    if(tecla == '2'){
        lcd.clear();
        lcd.setCursor (0, 1);
        lcd.print("***Riego Desactivado*");
        des_esp();
        lcd.setCursor (0, 1);
        delay(500);

        y=1;
        z=1;

        manual=0;
    }
}
}

```

```

void Activaciong() {

```

```

while(y<1)
{

    tecla=customKeypad.getKey();
    if(tecla == '1'){
        lcd.clear();
        lcd.setCursor (1, 1);
        lcd.print("***Riego Activado***");
        act_got();
        delay(500);

        y=1;
        z=1;

```

```

    manual=1;
}

    if(tecla == '2'){
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print("*Riego Desactivado*");
    des_got();
    delay(500);

    y=1;
    z=1;

    manual=0;
    }
    }
}

void act_esp1(){
    digitalWrite (bomba, HIGH);
    digitalWrite (valvulaH, HIGH);
}

void act_esp2(){
    digitalWrite (bomba, HIGH);
    digitalWrite (valvulaH2, HIGH);
}

void des_esp(){
    digitalWrite (bomba, LOW);
    digitalWrite (valvulaH, LOW);
    digitalWrite (valvulaH2, LOW);
}

void act_got(){
    digitalWrite (bomba, HIGH);
    digitalWrite (valvulaR, HIGH);
}

void des_got(){
    digitalWrite (bomba, LOW);
    digitalWrite (valvulaR, LOW);
}
//-----MONITOREO DE LOS DATOS DE LOS MÓDULOS-----
-----

void monitoreo_huertos()
{

```

```

while(y<1)
{
  tecla=customKeypad.getKey();

if(tecla=='1'){
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Modulo Ambiente 1");
  lcd.setCursor(1,1);
  lcd.print("Temperatura: ");
  lcd.setCursor(14,1);
  lcd.print(t);
  lcd.setCursor(1,2);
  lcd.print("Humedad: ");
  lcd.setCursor(14,2);
  lcd.print(h);
  lcd.setCursor(1,3);
  lcd.print("P. Rocio: ");
  lcd.setCursor(13,3);
  rocio1();
  lcd.print(pr);
  delay(100);

}

if(tecla =='2'){

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Modulo Ambiente 2");
  lcd.setCursor(1,1);
  lcd.print("Temperatura: ");
  lcd.setCursor(14,1);
  lcd.print(t2);
  lcd.setCursor(1,2);
  lcd.print("Humedad: ");
  lcd.setCursor(14,2);
  lcd.print(h2);
  lcd.setCursor(1,3);
  lcd.print("P. Rocio: ");
  lcd.setCursor(13,3);
  rocio2();
  lcd.print(pr2);
  delay(10);

}

if(tecla =='3'){

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("Monitoreo Suelo");
  lcd.setCursor(0,1);
  lcd.print("Hum.s1: ");

```

```

lcd.setCursor(7,1);
lcd.print(hs1);

lcd.setCursor(10,1);
lcd.print("Hum.s2: ");
lcd.setCursor(17,1);
lcd.print(hs2);

lcd.setCursor(0,2);
lcd.print("Hum.i1: ");
lcd.setCursor(7,2);
lcd.print(hi1);

lcd.setCursor(10,2);
lcd.print("Hum.i2: ");
lcd.setCursor(17,2);
lcd.print(hi2);

}

if(tecla == '*'){

lcd.clear();
delay(10);
y=1;
z=1;

}

}
}

```

PROGRAMACIÓN MÓDULOS DE MONITOREO

MONITOREO DE SUELO

```

#include <VirtualWire.h>
#include <OneWire.h> //Se importan las librerías
#include <DallasTemperature.h>

#define Pin 8 //Se declara el pin donde se conectará la DATA

OneWire ourWire(Pin); //Se establece el pin declarado como bus para la comunicación OneWire

DallasTemperature sensors(&ourWire); //Se instancia la librería DallasTemperature

// DECLARACION DE VARIABLES

int humedads1;
int humedads2;
int humedadi1;

```



```

int humedadi2;

char *mensaje;

void setup() {
  Serial.begin(9600);

  // INICIO RF
  //vw_set_ptt_inverted(true);
  vw_setup (2000);
  vw_set_tx_pin (7);
}

void loop() {

  // HUMEDAD 1

  int val1 = analogRead(A0);
  humedads1 = map(val1, 0, 1024, 100, 0);
  /* Serial.print("HUMEDAD1= ");
  Serial.print(humedads1);
  Serial.print(" ");*/
  delay(10);

  // HUMEDAD 2

  int val2 = analogRead(A1);
  humedads2 = map(val2, 0, 1024, 100, 0);
  /* Serial.print("HUMEDAD2= ");
  Serial.print(humedads2);
  Serial.print(" ");*/
  delay(10);

  // HUMEDAD 3

  int val3 = analogRead(A2);
  humedadi1 = map(val3, 0, 1024, 100, 0);
  /* Serial.print("HUMEDAD3= ");
  Serial.println(humedadi1);
  Serial.print(" ");*/
  delay(10);

  // HUMEDAD 4

  int val4 = analogRead(A3);
  humedadi2 = map(val4, 0, 1024, 100, 0);
  /* Serial.print("HUMEDAD4= ");
  Serial.println(humedadi2);*/
  delay(10);

  // ENVIO DE LA HUMEDAD 1

```

```

char buf[VW_MAX_MESSAGE_LEN]; // Cadena para enviar
String str="";

str = "e" + String(humedads1); //convertimos el entero a String y agramos un inicio de trama
str.toCharArray(buf, sizeof(buf)); //convertimos el String en un array
vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

Serial.print("Sensor1: ");
Serial.println(buf);
delay(2000);

str = "f" + String(humedads2); //convertimos el entero a String y agramos un inicio de trama
str.toCharArray(buf, sizeof(buf)); //convertimos el String en un array
vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

Serial.print("Sensor2: ");
Serial.println(buf);
delay(2000);

str = "g" + String(humedadi1); //convertimos el entero a String y agramos un inicio de trama
str.toCharArray(buf, sizeof(buf)); //convertimos el String en un array
vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

Serial.print(buf);
delay(2000);

str = "h" + String(humedadi2); //convertimos el entero a String y agramos un inicio de trama
str.toCharArray(buf, sizeof(buf)); //convertimos el String en un array
vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

Serial.println(buf);
delay(2000);
}

```

MONITOREO AMBIENTE 1

```

#include <VirtualWire.h>
#include <DHT.h>

```

```

#include "DHT.h" //cargamos la librería DHT
#define DHTPIN 2 //Seleccionamos el pin en el que se conectará el sensor
#define DHTTYPE DHT22 //Se selecciona

DHT dht(DHTPIN, DHTTYPE); //Se inicia una variable que será usada por Arduino para comunicarse
con el sensor

void setup() {
  Serial.begin(9600);

  // INICIO RF
  //vw_set_ptt_inverted(true);
  vw_setup(2000);
  vw_set_tx_pin(7);
}

void loop() {
  float h = dht.readHumidity(); //Se lee la humedad
  float t = dht.readTemperature(); //Se lee la temperatura
  //Se imprimen las variables
  //delay(2000); //Se provoca un lapso de 1 segundo antes de la próxima lectura

  //HUMEDAD
  char buf[VW_MAX_MESSAGE_LEN]; // Cadena para enviar
  String str="";

  str="a"+String(h); //convertimos el entero a String y agramos un inicio de trama
  str.toCharArray(buf,sizeof(buf)); //convertimos el String en un array
  vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
  vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

  Serial.print("Humedad: ");
  Serial.println(buf);
  delay(2000);

  // TEMPERATURA
  str="b"+String(t); //convertimos el entero a String y agramos un inicio de trama
  str.toCharArray(buf,sizeof(buf)); //convertimos el String en un array
  vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
  vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

  Serial.print("Temperatura: ");
  Serial.println(buf);
  delay(1000);
}

```

MONITOREO AMBIENTE2

```
#include <VirtualWire.h>
#include <DHT.h>

#include "DHT.h" //cargamos la librería DHT
#define DHTPIN 2 //Seleccionamos el pin en el que se conectará el sensor
#define DHTTYPE DHT22 //Se selecciona el DHT22(hay otros DHT)
DHT dht(DHTPIN, DHTTYPE); //Se inicia una variable que será usada por Arduino para comunicarse
con el sensor

void setup() {
  Serial.begin(9600);

  // INICIO RF
  //vw_set_ptt_inverted(true);
  vw_setup(2000);
  vw_set_tx_pin(7);
}

void loop() {
  float h = dht.readHumidity(); //Se lee la humedad
  float t = dht.readTemperature(); //Se lee la temperatura

  //HUMEDAD
  char buf[VW_MAX_MESSAGE_LEN]; // Cadena para enviar
  String str="";

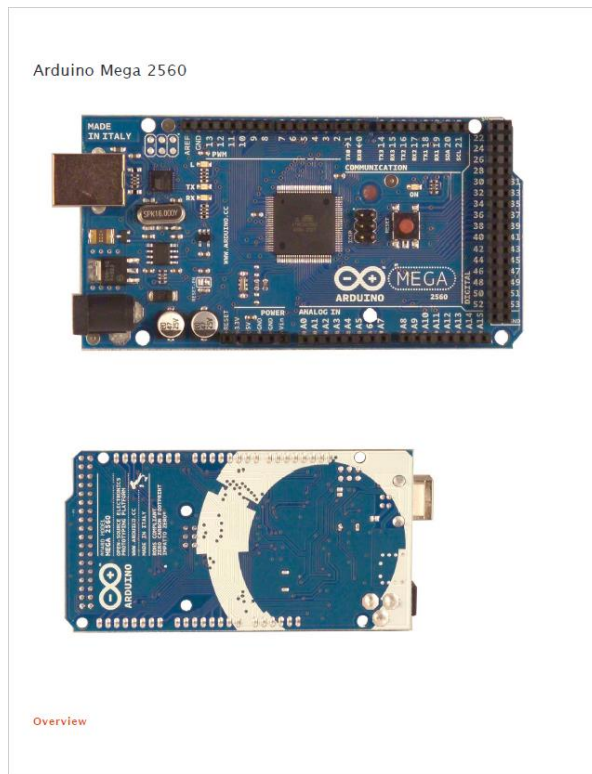
  str="d"+String(h); //convertimos el entero a String y agramos un inicio de trama
  str.toCharArray(buf,sizeof(buf)); //convertimos el String en un array
  vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
  vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

  Serial.print(buf);
  delay(2000);

  // TEMPERATURA
  str="c"+String(t); //convertimos el entero a String y agramos un inicio de trama
  str.toCharArray(buf,sizeof(buf)); //convertimos el String en un array
  vw_send((uint8_t *)buf, strlen(buf)); //Enviamos el array
  vw_wait_tx(); //Esperamos hasta que el mensaje se envíe

  Serial.print(buf);
  delay(2000);
}
```

ANEXO 2: Hojas técnicas elementos del sistema electrónico



Arduino Mega 2560

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Decimilla.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

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

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

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

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

There are a couple of other pins on the board:

+ AREF. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

+ Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

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

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

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 100 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Decimilla or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Decimilla. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Decimilla (analog inputs 4 and 5).*

- 4.2 Power supply pins (VDD GND)
AM2301 supply voltage range 3.3V – 5.2V, recommended supply voltage is 5V.
- 4.3 Serial data (SDA)
SDA pin is tri-state for reading, writing sensor data. Specific communication timing, see the detailed description of the communication protocol.

5. Sensor performance

5.1 Relative humidity

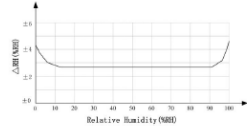
Table 2: AM2301 Relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution		0.1			%RH
Range	0		99.9		%RH
Accuracy ^[1]	25°C	±3			%RH
Repeatability		±1			%RH
Exchange		Completely interchangeable			
Response ^[2]	1/e(63%)	<6			S
Staggish		±0.3			%RH
Drift ^[3]	Typical	<0.5			%RH/yr

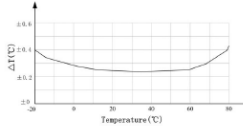
5.2 Temperature

Table 3: AM2301 Relative temperature performance

Parameter	Condition	min	typ	max	Unit
Resolution		0.1			°C
Accuracy	n		16		bit
Range		±0.3		±1	°C
Repeat		-40		80	°C
Exchange		Completely interchangeable			
Response ^[4]	1/e(63%)	<10			S
Drift		±0.3			°C/yr



Pic2: At 25°C The error of relative humidity



Pic3: The maximum temperature error

6. Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depending on the power supply. Table 4 details the electrical characteristics of the AM2301, if not identified, said supply voltage of 5V. To get the best results with the sensor, please design strictly in accordance with the conditions of design in Table 4.

Table 4: AM2301 DC Characteristics

Parameter	Condition	min	typ	max	Unit
Voltage		3.3	5	5.2	V
Power consumption ^[5]	Dormancy	10	15		µA
	Measuring		500		µA
	Average		300		µA
Low level output voltage	I_{OL} ^[6]	0		300	mV
High output voltage	$R_P < 25\text{ k}\Omega$	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High Voltage	Rise	70%		100%	VDD
R_{pu} ^[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	turn on		8		mA
	turn off		10	20	µA
Sampling period		2			S

[1] the accuracy of the factory inspection, the sensor 25 ° C and 5V, the accuracy specification of test conditions, it does not include hysteresis and nonlinearity, and is only suitable for non-condensing environment.
 [2] to achieve an order of 63% of the time required under the conditions of 25 ° C and 1m / s airflow.
 [3] in the volatile organic compounds, the value may be higher. See the manual application to store information.
 [4] this value at VDD = 5.0V when the temperature is 25 ° C, 25 / time, under the conditions of the average.
 [5] low output current.
 [6] that the pull-up resistor.

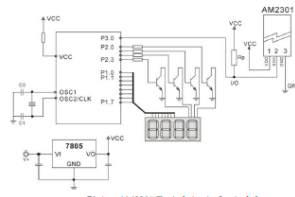
7. Single-bus communication (ONE-WIRE)

7.1 Typical circuits for single bus

Microprocessor and AM2301 connection typical application circuit is shown in Figure 4. Single bus communication mode, pull the SDA microprocessor I / O port is connected.

Special instructions of the single-bus communication:

1. Typical application circuit recommended in the short cable length of 30 meters on the 5.1K pull-up resistor pullup according to the actual situation of lower than 30 m.
2. With 3.3V supply voltage, cable length shall not be greater than 100cm. Otherwise, the line voltage drop will lead to the sensor power supply, resulting in measurement error.
3. Read the sensor minimum time interval for the 2S; read interval is less than 2S, may cause the temperature and humidity are not allowed or communication is unsuccessful, etc..
4. Temperature and humidity values are each read out the results of the last measurement For real-time data that need continuous read twice, we recommend repeatedly to read sensor, and each read sensor interval is greater than 2 seconds to obtain accurate the data.



Pic4: AM2301 Typical circuits for single bus

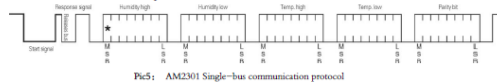
7.2. Single-bus communication protocol

Single-bus Description

AM2301 device uses a simplified single-bus communication. Single bus that only one data line, data exchange system, controlled by the data line to complete. Equipment (microprocessor) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external about 5.1kΩ pull-up resistor, so when the bus is idle, its status is high. Because they are the master-slave structure, only the host calls the sensor, the sensor will answer, so the hosts to access the sensor must strictly follow the sequence of single bus, if there is a sequence of confusion, the sensor will not respond to the host.

Single bus to send data definition

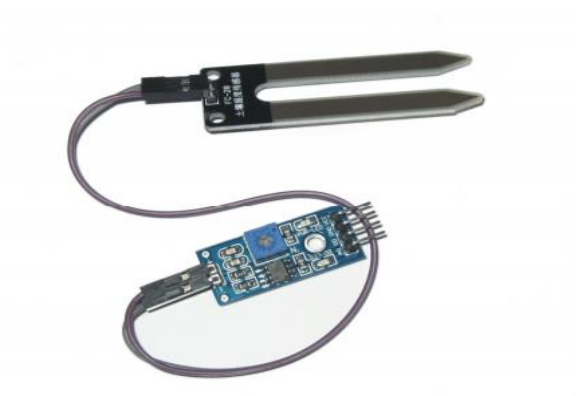
SDA For communication and synchronization between the microprocessor and the AM2301, single-bus data format, a transmission of 40 data, the high first-out. Specific communication timing shown in Figure 5, the communication format is depicted in Table 5.



Pic5: AM2301 Single-bus communication protocol

FC-28

Sensor de humedad del suelo



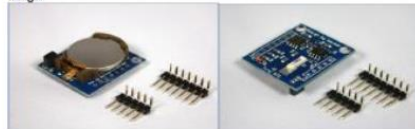
Voltaje de 3,3 a 5V cc
Salidas analógica y comparadora
Ajuste de sensibilidad
Dimensiones del sensor 6x-20mm contactos 45mm
Dimensión del comparador 30x14mm

Los cuatro conectores son

VCC positivo de 3,3 a 5V CC
GND masa
OD salida de señal digital (a una determinada humedad en el sensor se dispara) se puede ajustar con el potenciómetro el nivel a controlar
OA salida de señal analógica a cambia la tensión de salida en función de la cantidad de humedad que registra el sensor

Real Time Clock Module

Image:



Description:

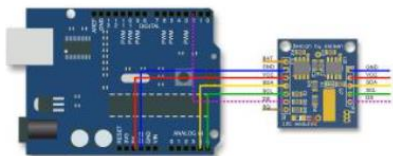
Arduino Tiny RTC I2C Real Time Clock

This Arduino Tiny RTC I2C module incorporates the DS1307 I2C real time clock IC and the 24C32 32K I2C EEPROM storage. What's more, it has a DS18B20 temperature sensor on board. All of this in a tiny package of 25mm x 28mm x 8.4mm. It comes with a LIR2303 rechargeable lithium battery, and a charging circuit is included in the module. When the temperature sensor is off, the RTC module can run for 1 year on a single charge.

Specs of DS18B20 Temperature Sensor

- 3.0-5.5V input voltage
- Waterproof
- -55°C to +125°C temperature range
- ±0.5°C accuracy from -10°C to +85°C
- 1 Wire interface

Wiring the RTC to Arduino



The most useful pins are duplicated from P1 to P2. If needed, the pin "BAT" can be fed into an ADC pin for monitoring the battery voltage. The pin-outs are explained below. Connecting "VCC" to 5V will trickle charge the onboard battery.

Arduino Tiny RTC I2C Real Time Clock Pinout

PIN	Description	Comment
BAT	Battery voltage	To monitor the battery voltage, or not connected
GND	Ground	Ground
VCC	5V supply	Power the module and charge the battery
SDA	I2C data	I2C data for the RTC
SCL	I2C clock	I2C clock for the RTC
DS	DS18B20 Temp. Sensor output	One wire interface
SQ	Square wave output	Normally not used

The I2C wires "SDA" and "SCL" are the data line and clock line, they should be connected to the corresponding pins depending on the Arduino board.

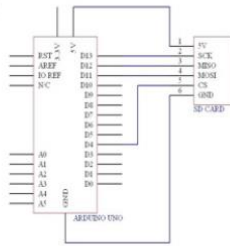
Board	I2C / TWI pins
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1



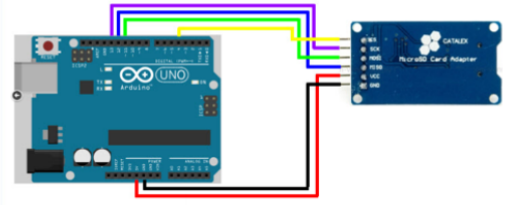
The Arduino SD Card Module is a simple solution for transferring data to and from a standard SD card. The pin out is directly compatible with Arduino, but can also be used with other microcontrollers. It allows you to add mass storage and data logging to your project.

Pin Definition

- CS-----Chip Select
- SCK-----Serial Clock
- MISO-----Serial data Out
- MOSI-----Serial data in
- VCC-----Voltage Supply (3.3 / 5 V)
- GND-----Ground



SD Card Arduino Tutorial



Arduino hardware Connection:

- | SD CARD | Arduino |
|------------------|----------------------------|
| CS----- | PIN 4 |
| SCK (CLOCK)----- | PIN 13 |
| MISO----- | PIN 12 |
| MOSI----- | PIN 11 |
| VCC----- | Voltage Supply (3.3 / 5 V) |
| GND----- | Ground |

Módulo de 4 relés para Arduino



Las salidas de las placas Arduino son perfectamente útiles para controlar cargas que no consuman demasiada corriente, como un Led, pero son insuficientes para cargas mayores. ¿Como hacemos para controlar por ejemplo una lámpara o un motor que se alimentan de 220 voltios? Una forma es emplear un módulo de relés como el que analizo en este artículo.

Antes de comenzar con el análisis: como sucede con muchos módulos para Arduino que se consiguen en el mercado, la placa que utilicé no tiene una marca que identifique a su fabricante ni información "oficial" sobre su funcionamiento. Seguramente proviene de alguna factoría china que produce estos productos que luego son vendidos por numerosas tiendas a través de la red y del cual pueden existir distintas variedades, con diferentes características. Antes de conectarlo según las indicaciones que se dan a continuación, por favor, asegúrense de que el módulo que tienen entre sus manos sea igual al que se describe aquí.

Hecha esta salvedad, comencemos con una descripción general de la placa. Se trata de un módulo de 4 relés (o relays) que funcionan a 5 Voltios, capaces de manejar cargas de hasta 10 Amperes en 250 Voltios, convenientemente aislados mediante optoacopladores de las entradas, las que cuentan con leds individuales que sirven como indicadores de estado.

Los distintos componentes del módulo pueden verse en la siguiente imagen:



Fig. 1. Partes del módulo

Como se puede apreciar, la placa tiene un conector de entradas (IN1 a IN4) y alimentación (GND es masa o negativo y Vcc es el positivo) [1], cuatro leds que indican el estado de las entradas [2], un jumper selector para la alimentación de los relés [3], cuatro optoacopladores del tipo FL17C [4], cuatro diodos de protección [5], cuatro relés marca SONGLE son bobinas de 5V y contactos capaces de controlar hasta 10 Amperes en una tensión de 250V [6] y cuatro bobinas, con tres contactos cada una (Común, Normal abierto y Normal cerrado), para las salidas de los relés [7].

Esquemático

En la imagen de mas abajo se puede apreciar el circuito esquemático de un canal, el resto de los canales repite la misma configuración.

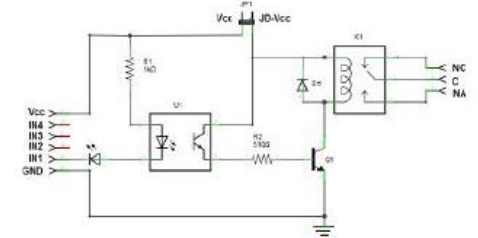


Fig. 2. Circuito esquemático

Funcionamiento

A partir del circuito de la Fig. 2 analicemos el funcionamiento del circuito: la entrada IN1 está conectada al cátodo del diodo del optoacoplador a través del led indicador. El ánodo del diodo del optoacoplador se conecta a Vcc (positivo) por intermedio de R1, una resistencia de 1000 ohms. Estos tres componentes, el diodo indicador, el diodo del resto y la R1 forman un circuito serie por el cual circula la corriente cuando la entrada está a un nivel BAJO (conectada a GND) y no circula si la entrada está a un nivel ALTO (conectada a Vcc).

El transistor de opto tiene su colector a JD-Vcc y su emisor conectado a Q1 a través de una resistencia de 510 ohms. Este es otro circuito serie por el cual circula corriente cuando el transistor del opto conduce al ser "iluminado" por su diodo, con lo que se introduce corriente en la base de Q1 a través de R2.

Finalmente, Q1 está conectado en una típica configuración emisor común, con su emisor a masa (GND) y la bobina del relé como carga en el colector. Cuando circula corriente por la base desde

el opto, Q1 se satura permitiendo el paso de la corriente a través de la bobina del relé, lo que produce que se cierren los contactos del mismo (común con normal abierto). El diodo D1 protege al transistor de la tensión que aparece en la bobina del relé cuando desea de circuitar corriente por la misma.

En síntesis, al pensarse la entrada a nivel BAJO se pone a la saturación el transistor Q1 a través de fotoacoplador con lo que se deriva el contacto normal abierto del relé.

Alimentación y consumo

La forma más sencilla de alimentar este módulo es desde Vcc y GND de la placa Arduino, manteniendo el jumper en su lugar, con lo que JD-Vcc = Vcc. Esta conexión tiene dos limitaciones importantes:

- Se pierde la aislación eléctrica que brindan los fotoacopladores, lo que aumenta la posibilidad de daño al Arduino si hay algún problema con las cargas de los relés.
- La corriente consumida por las bobinas de los relés debe ser provista por la placa Arduino. Cada bobina consume unos 90 mA y las cuatro juntas suman 360 mA. Si a eso le sumamos los consumos que pueden tener otras salidas, estamos muy cerca de los 500 mA que puede suministrar un puerto USB. En este caso se debería alimentar al Arduino con una fuente externa, lo que aumenta el límite de corriente a 1 A (en el caso de la Arduino UNO).



Fig. 3. Alimentación con una sola fuente

La forma más segura es remover el jumper y alimentar la placa de relés con dos fuentes: la de la placa Arduino conectada a Vcc y una segunda fuente, con el positivo a JD-Vcc y el negativo a GND, sin estar éste unido a la placa Arduino. Esta conexión tiene como ventajas:

- Hay completa aislación entre la carga y el Arduino.
- Todo el consumo de los relés es tomado de la segunda fuente y no del Arduino o el puerto USB.

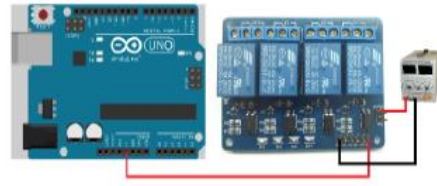


Fig. 4. Alimentación con dos fuentes

Entradas

Las entradas a la placa puede conectarse directamente a las salidas digitales de la placa Arduino. La única precaución a tener en cuenta es que cuando Arduino arranca a ser alimentado, los pines se configuran como entradas automáticamente y puede ocurrir que, por un brevísimo lapso de tiempo entre el arranque y la correcta configuración de estos pines como salidas, las entradas de control al módulo de relé queden en un estado indeterminado. Esto se puede evitar conectando en cada entrada un pull-up con una resistencia de 10K a Vcc, lo que asegura un estado ALTO durante el arranque.



RIEGO RESIDENCIAL Y COMERCIAL
Built on Innovation®

- 01 **Trabajando Duro:** Construida para durar y funcionar durante años
- 02 **Versátil:** La PGV está disponible en una gran variedad de configuraciones, adaptándose prácticamente a cualquier situación
- 03 **Práctica:** Las opciones de bajo caudal permiten el uso de los productos Hunter de riego localizado

PGV VÁLVULAS

Válvulas de Categoría Profesional Preparadas para Instalaciones de Cualquier Tamaño

Esta excelente y resistente válvula ofrece las mejores prestaciones de las electroválvulas Hunter. Creada con suficiente robustez para resistir los rigores de la mayoría de instalaciones residenciales y pequeñas áreas comerciales, la PGV está disponible en un amplio abanico de configuraciones. Para uso en jardines pequeños, conexión 1" BSP está disponible en cuatro configuraciones: Diseño de ángulo, en línea, macho - macho y macho-espiga. A su vez, cada modelo está disponible en la versión con control de caudal y sin control de caudal. Para aplicaciones en jardines más grandes hay conexiones disponibles de 1.5" y 2" BSP. La PGV viene en modelos en línea/ángulo con control de caudal. Todos los modelos cuentan con una construcción duradera de gran calidad y de un sólido diafragma con soporte para evitar deformaciones.

PGV VALVE CARACTERÍSTICAS Y ESPECIFICACIONES

Características

- Aplicación (PGV-100): Residencial
- Aplicación (PGV-101, 151, 201): Residencial/pequeñas áreas comerciales
- Conexión rosca: 1" (25 mm), 1.5" (40 mm) y 2" (50 mm)
- La purga manual interna y externa permite una activación rápida y sencilla a "nivel de la válvula"
- Diseño durable con tornillos para una máxima resistencia
- Diseño de junta de diafragma con doble reborde para un rendimiento superior sin pérdidas
- Los solenoides tipo "latch" CC admiten los programadores Hunter con pilas
- Los tornillos cautivos de la tapa permiten un mantenimiento de la válvula sin problemas
- La posibilidad de caudal bajo permite utilizar los productos de microrriego de Hunter
- Solenoide encapsulado de 24 V CA con émbolo cautivo para un mantenimiento sin problemas
- Temperatura máxima: 66 °C
- Período de garantía: 2 años

Especificaciones de funcionamiento

- Caudal:
 - PGV-100: de 0,05 a 9 m³/h de 0,7 a 150 l/min
 - PGV-101: de 0,05 a 9 m³/h de 0,7 a 150 l/min
 - PGV-151: de 5 a 27 m³/h de 75 a 450 l/min
 - PGV-201: de 5 a 34 m³/h de 75 a 570 l/min
- Intervalo de presión recomendado: de 1,5 a 10 bar; de 150 a 1.000 kPa

Especificaciones del Solenoide

- Solenoide de 24 VAC:
 - 350 mA en arranque, 190mA en mantenimiento, 60 HZ
 - 370 mA en arranque, 210mA en mantenimiento, 50 HZ

PGV VALVES

Model	Description
PGV-100G	Válvula en línea de plástico de 1" (25 mm), rosca hembra
PGV-101G	Válvula en línea de plástico de 1" (25 mm) con control de caudal, rosca hembra
PGV-100A	Válvula en ángulo de plástico de 1" (25 mm)
PGV-101A	Válvula en ángulo de plástico de 1" (25 mm) con control de caudal
PGV-100MB	Válvula en línea de plástico de 1" (25 mm), rosca macho x espiga T (25 mm)
PGV-100MB125	Válvula en línea de plástico de 1" (25 mm), rosca macho x espiga de 1/8" (20 mm)
PGV-100MB	Válvula en línea de plástico de 1" (25 mm) con control de caudal, rosca macho x espiga de 1" (25 mm)
PGV-100MB125	Válvula en línea de plástico de 1" (25 mm) con control de caudal, rosca macho x espiga de 1/8" (20 mm)
PGV-100MM	Válvula en línea de plástico de 1" (25 mm) con control de caudal, macho x rosca macho
PGV-100MM	Válvula en línea de plástico de 1" (25 mm) con control de caudal, macho x rosca macho
PCZ10125	Kit de control de zona de riego de 1" (25 mm), regulador 2,75 bares (175 kPa)
PCZ10140	Kit de control de zona de riego de 1.5" (40 mm), regulador 2,75 bares (175 kPa)
PGV-151	Válvula en línea/ángulo de plástico de 1.5" (40 mm) con control de caudal
PGV-201	Válvula en línea/ángulo de plástico de 2" (50 mm) con control de caudal

PGV, PÉRDIDA DE PRESIÓN EN BARES

m ³ /h	25 mm en línea	25 mm en ángulo	25 mm en línea	40 mm en línea	40 mm en ángulo	50 mm en línea	50 mm en ángulo
0,25	0,10	0,07					
1,00	0,10	0,07					
2,50	0,12	0,08					
3,50	0,16	0,09					
4,50	0,22	0,12	0,21	0,22	0,08	0,08	
7,00	0,44	0,22	0,22	0,21	0,08	0,08	
9,00			0,24	0,21	0,09	0,09	
11,00			0,26	0,22	0,11	0,09	
13,00			0,31	0,26	0,14	0,10	
16,00			0,44	0,37	0,21	0,14	
22,50			0,62	0,51	0,31	0,22	
27,00			0,84	0,75	0,44	0,33	
30,50					0,56	0,45	
34,00					0,70	0,59	



Website hunterindustries.com | Asistencia Técnica SoporteTecnico@hunterindustries.com

Lo que nos motiva es contribuir al éxito de nuestros clientes. Aunque nuestra pasión por la innovación y la ingeniería está presente en todo lo que hacemos, es el compromiso que hemos contraído de ofrecerle una asistencia excepcional lo que nos hace albergar la esperanza de que siga formando parte de la familia de clientes de Hunter en los próximos años.

G.R.H.A.
Gregory R. Hunter, President of Hunter Industries

© 2014 Hunter Industries Incorporated WNF-361, ES G 6/14

¡Aprenda más. Visite hunterindustries.com para más información.

MÓDULOS DE RF PARA RADIOCONTROL

Introducción

A la hora de pensar en diseñar sistemas radiocontrolados, surgen mas de una opción para llevar a cabo nuestro proyecto en lo que respecta al transmisor y receptor que emplearemos para el envío y recepción de las órdenes.

Hace un tiempo atrás la única posibilidad, económica, era la construcción y calibración del circuito transmisor y receptor, lo que implicaba dedicar un tiempo importante para conseguir este objetivo. Luego podíamos dedicarnos al resto de las partes del proyecto.

Hoy en día disponemos a un costo razonable módulos de RF, transmisor y receptor diseñados para poderlos a funcionar con pocos componentes adicionales.

NOTA: Los módulos que se presentarán aquí son solo para transmisión de datos digitales. Es decir estos **no sirven** si lo que deseamos es transmitir audio o video. Y dado que la tasa de transmisión de datos es muy baja, tampoco podremos transmitir audio y video digitalizados. El diseñador que quiera transmitir audio o video deberá elegir otro tipo de componentes.

Vemos a continuación un ejemplo:

MODULO EMISOR



MODULO RECEPTOR



Ejemplos de módulos emisor y receptor.

Podemos encontrar en el mercado opciones interesantes en cuanto a frecuencia, potencia y alcance de estos módulos, el diseñador del sistema radiocontrolado deberá optar por alguno de acuerdo a sus necesidades y procurarse la hoja de datos correspondiente.

Un fabricante es **Wenshing Electronics CO L.T.D** cuyos módulos **transmisores** presentan las siguientes características.

Model	Frequency (MHz)		Data Rate (Kbps)		Voltage (V)		Current (mA)	RF Power (dBm)
	Min	Max	Min	Max	Min	Max		
TWS-HSP	315	433.92	8	8	4.5	5.5	487	32
TWS-EIS	303.875	433.92	8	8	1.5	12	53	20
TWS-CS	315	433.92	8	8	2	12	0.5	0.0
TWS-MS	315	868.38	8	8	4.5	5.5	278	27
TWS-HFP	409	500	8	8	3	5.3	204	33
TWS-ES	303.875	916.5	8	8	3	12	8.4	14

Como se puede apreciar hay diferentes opciones para la selección, tensión de trabajo, corriente que consumen, frecuencia, potencia RF emitida.

Estos módulos transmisores tienen sus correspondientes módulos receptores, especialmente preparados para su funcionamiento conjunto.

Nivel dBm	Potencia	Aplicaciones
27 dBm	500 mw	Potencia típica de transmisión de un teléfono celular. Máxima salida de potencia para un teléfono celular UMTS/3G (Teléfono de potencia clase 2)
24 dBm	250 mw	Máxima salida de potencia para un teléfono celular UMTS/3G (Teléfono de potencia clase 3)
21 dBm	125 mw	Máxima salida de potencia para un teléfono celular UMTS/3G (Teléfono de potencia clase 4)
20 dBm	100 mw	Bluetooth estándar Clase 1. Cobertura 100 m. Potencia típica de un router inalámbrico WiFi.
15 dBm	32 mw	Potencia típica de transmisión WiFi en computadoras portátiles.
10 dBm	10 mw	
6 dBm	4 mw	
4 dBm	2.5 mw	Bluetooth estándar Clase 2. Cobertura 10 m.
3 dBm	2 mw	
2 dBm	1.6 mw	
1 dBm	1.3 mw	
0 dBm	1mw =1000 uW	Bluetooth estándar Clase 3. Cobertura 1 m.
-1 dBm	794 uW	
-10 dBm	100 uW	Potencia de señal típica de una red inalámbrica WiFi (-10 a -30 dBm).
-70 dBm	100 pW	Rango típico (-60 a -80 dBm) de potencia de señal inalámbrica (802.11x) recibida por una red.
-130 dBm	0.1 nW	Potencia típica recibida por un receptor GPS (en tierra)

Conclusión:

Esto nos dice que siempre desearemos que el transmisor emita con los mas dBm posibles (positivos) y que el receptor sea lo mas sensible posible, o sea mas dBm negativos.

Una de las especificaciones clave en los diseños de los receptores de RF de hoy en día es la sensibilidad. Esta indica la capacidad del receptor para capturar señales débiles y por tanto será una medida directa del alcance del sistema.

Criterios de calidad de un receptor

- Sensibilidad: Nivel mínimo de señal que es capaz de detectar con la calidad deseada.
- Sensibilidad: Nivel mínimo de señal que es capaz de detectar con la calidad deseada.
- Fidelidad: Capacidad de recibir y demodular la señal sin distorsión.

Tipo de modulación empleada por los módulos

Algunos conceptos previos:

Banda base

Se denomina banda base al conjunto de señales que no sufren ningún proceso de modulación a la salida de la fuente que las origina, es decir son señales que son transmitidas en su **frecuencia original**. Dichas señales se pueden codificar y ello da lugar a los códigos de banda base.

Las señales empleadas en banda base se pueden clasificar de la siguiente forma:

Unipolares

En este caso un 1 siempre toma una polaridad, positiva o negativa, mientras que un 0 vale siempre 0.



A continuación se muestra una tabla del mismo fabricante, pero de módulos receptores:

Model	Frequency (MHz)		Data Rate (Kbps)		Voltage (V)		Current (mA)	Sensitivity (dBm)
	Min	Max	Min	Max	Min	Max		
RWS-DXALLY	315	433.92	0.008	20	3.0	5.5	7	-118
RWS-C488	315	433.92	0.008	12	3.5	5.5	5.7	-117
RWS-375	315	433.92	4.8	4.8	4.9	5.1	4.8	-108
RWS-438	303.025	433.92	0.2	9.6	4.8	5.5	2.7	-114
RWS-438	315	433.92	4.8	4.8	4.5	5.5	7.6	-108
RWS-530	315	433.92	4.8	4.8	4.5	5.5	0.3	-112
RWS-274	303.375	433.92	4.8	4.8	4.9	5.1	4.8	-108
RWS-1230	123	123	4.8	4.8	*	*	*	-108
RWS-371	303.025	433.92	4.8	4.8	4.9	5.1	4.8	-108
RWS-A803	300	1000	3	160	2.7	5.5	12	-112
RWS-A816	300	1000	3	160	2.7	5.5	12	-112

Notemos que en cuanto a tensión de trabajo, este fabricante ofrece solo de 5volts.

Concepto de dBm (para comprender la tabla anterior)

El dBm es una unidad de medida utilizada principalmente en telecomunicación para expresar la potencia absoluta mediante una relación logarítmica.

El dBm se define como el nivel de potencia en decibelios en relación a un nivel de referencia de 1 mW.

El valor en dBm en un punto, donde tenemos una potencia P, viene dado por la fórmula siguiente:

$$dBm = 10 \times \log_{10} \frac{P}{1mW}$$

Al utilizarse un nivel de referencia determinado (1 mW) la medida en dBm constituye una verdadera medición de la potencia y no una simple relación de potencias como en el caso de la medida en decibelios. Así, una lectura de 20 dBm significa que la potencia medida es 100 veces mayor que 1mW y por tanto igual a 100 mW.

Puesto que se trata de una relación logarítmica, podemos cometer error al comparar por simple inspección potencias medidas en dBm. A continuación se agrega un cuadro que por medio de ejemplos nos muestra aplicaciones reales y las potencias involucradas. *recopilado de internet.

Nivel dBm	Potencia	Aplicaciones
80 dBm	100 kW	Potencia típica de transmisión de una estación de radio FM con un alcance de 50 Km, aproximadamente.
60 dBm	1 kW	Radiación típica combinada de RF de un horno de microondas.
40dBm	10 W	Potencia entregada a las antenas de telefonía móvil.
36 dBm	4 W	Salida típica para banda de radio ciudadana (27MHz).
33 dBm	2 W	Máxima salida de potencia para un teléfono celular UMTS / 3G. (Teléfono de potencia clase 1). Máxima salida de potencia para teléfono celular GSM 300/400.
30 dBm	1 W	Fuga de RF típica de un horno microondas- Máxima salida de potencia para teléfono celular GSM 1800/1900.

Polares

En este caso la señal tomara valores positivos para un 1 lógico y negativos para un 0 lógico pero nunca toma el valor 0.



Bipolares

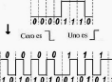
En este caso un dígito toma valor con polaridad alternada mientras que el otro permanece siempre en 0.



La transmisión en **banda base** es utilizada para cortas distancias debido a su bajo costo, por ejemplo en una red LAN cableada la información que viaja por el cable UTP lo hace en banda base.

Básicamente esto quiere decir que la información (digital) viaja tal como se genera, y eventualmente lo hace codificada. Notemos que si se tiene una larga secuencia de ceros o de unos, la señal permanece constante durante un tiempo bastante largo en la línea y el receptor no puede identificar el principio y fin de cada bit. Este inconveniente se resuelve justamente con la codificación. Por ejemplo en las redes LAN (Ethernet) cableadas se utiliza banda base pero con codificación Manchester.

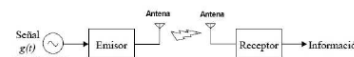
Codificación Manchester



Hasta ahora hemos mencionado la transmisión de señales digitales en banda base. En la práctica, ya sea por que se debe compartir el canal (por ejemplo el aire, el cable coaxial) o para poder usar antenas de dimensiones razonables (recordemos que las dimensiones de una antena dependen de la longitud de onda de la señal que desea transmitir), es necesario modular.

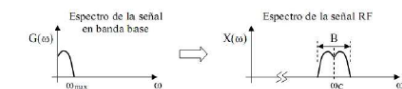
Modular significa modificar la **amplitud**, la **frecuencia** o la **fase** de una señal portadora (que puede ser una sinusoidal), en función del mensaje. Esa señal portadora siempre es una señal de frecuencia elevada, cuanto mas elevada es la frecuencia de la portadora, mas pequeña será la antena necesaria. Por supuesto que tal elección también dependerá de prestaciones y costos.

El esquema general de la radiodifusión es el que se muestra en la figura:



La **antena**, es el transductor que convierte la señal eléctrica en ondas electromagnéticas de RF. Pero la señal que llega a la antena tiene que tener la frecuencia apropiada.

La modulación consiste en trasladar el espectro (componentes de frecuencias) de la señal g(t) (que es la información que se desea transmitir) desde la componente continua, ω = 0, hasta ωc. Sucede además que el espectro se dobla alrededor de ωc, es decir, aparece una copia espejular del espectro además del inicial.



Operación de traslado en frecuencia que realiza el emisor

ANEXO 3: Entrevistas sobre el tema

ENTREVISTA SOBRE ASPECTOS REPRESENTATIVOS EN HUERTOS DE FRESA

Realizado por: Carlos Daniel Arcos Guamán

Realizada a: Ing. Agr. Jenny Acosta P.

¿Qué ventajas nos brinda al realizar camas con cubiertas plásticas para el cultivo de fresas?

El hecho de tener una cubierta plástica en los huertos de fresa nos brinda ventajas como:

- Conservación de la humedad y los nutrientes del suelo.
- Aumentar la Temperatura del suelo y en la raíz de la planta.
- Tener frutos limpios.

¿La humedad y temperatura son iguales en un número de camas?

Sí, porque al tener todas las camas de fresa la misma cubierta plástica se genera un microclima igual.

¿Cuál es la profundidad radicular de la fresa?

La profundidad radicular en las fresas llega a los 0.3m suelo.

¿Cuál es la humedad optima del suelo para los cultivos de fresa?

El suelo debe estar en "*capacidad de campo*", esto quiere decir que no debe faltar ni sobrar agua.

¿Cómo se puede identificar el cambio climático que genera las heladas?

Se puede identificar con aspectos como:

- Cielo despejado (sin nubes).
- Temperaturas bajas.
- Incremento de la intensidad y la velocidad del viento.

ENTREVISTA SOBRE DATOS DEL TERRENO DE PRUEBAS

Realizado por: Carlos Daniel Arcos

Realizada a: Sra. Martha Guamán

¿Cuántas plantas de fresas se encuentra sembrado en el huerto?

Aproximadamente 7500 plantas

¿Cuál fue la inversión inicial para las plantaciones?

Se gastó un total de \$ 2500,00

¿De qué forma ayuda el sistema de supervisión instalado en el huerto?

A reducir el gasto del agua y mediante el sistema de riego por aspersión que las plantas de fresas no se dañen en el momento de la helada

¿Cree representativo el costo de \$371,00 para la instalación del sistema de supervisión?

No tiene un costo elevando en comparación a la inversión inicial