

DISEÑO DE SISTEMA DE CONTROL DE OPTIMIZACIÓN AMBIENTAL PARA
CULTIVO HIDROPÓNICO

GIANCARLO DARAVIÑA PEÑA
RICHARD VALENCIA GUTIERREZ

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE TECNOLOGÍAS
INGENIERÍA EN MECATRÓNICA POR CICLOS
PEREIRA
2014

DISEÑO DE SISTEMA DE CONTROL DE OPTIMIZACIÓN AMBIENTAL PARA
CULTIVO HIDROPÓNICO

GIANCARLO DARAVIÑA PEÑA

RICHARD VALENCIA

Trabajo de grado para optar a título de Tecnólogo en Mecatrónica

Director

Carlos Andrés Rodríguez Pérez

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE TECNOLOGÍAS
INGENIERÍA EN MECATRÓNICA POR CICLOS
PEREIRA
2014

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

Pereira, enero de 2014

Dedicatoria

Dedicamos este proyecto a todos los que creyeron en nosotros, a toda la gente que nos apoyó, a nuestros amigos y familiares.

CONTENIDO

Pág.

Introducción	1
1. cultivo para la implementación del sistema de control	2
1.1. Lechugas hidropónicas	2
1.1.1. Lechuga	2
1.1.2. Método de cultivo	2
1.1.3. Requerimientos climáticos del cultivo de lechuga	3
2. Determinación de los sensores.	4
2.1. Sensor de temperatura	4
2.2. Sensor de humedad	6
2.2.1. Presión parcial de vapor	6
2.2.2. Presión de vapor de saturación	6
2.2.3. Humedad relativa	7
2.2.4. Sensores capacitivos	7
2.3. Control de circulación	9
3. Determinación del hardware y software para control de variables.	10
3.1. selección del hardware para la manipulación de variables	10
3.1.1. Arduino mega 2560	12
3.2. Determinación del software para programar el Arduino.	15
3.2.1. Software Arduino.	15
3.3. Lenguaje de programación Arduino	15
4. Sistema inalámbrico.	17
4.1. Modulo inalámbrico XBee serie 1.	17
4.1.1. Bluetooth	17
4.1.2. Radio Frecuencia.	17
4.1.3. XBee	18
4.1.4. El Coordinador.	19
4.1.5. Los Routers.	19
4.1.6. Aplicaciones	20
4.2. Modo de recibir/transmitir	20
4.3. Configuración del módulo Xbee	21
4.3.1. Conexión Punto a Punto	21
5. Software X-CTU	24
6. Selección de software para graficar las variables.	25
7. Construcción del invernadero	26
8. Conclusiones	29
9. Recomendaciones	30
10. Bibliografías	31
11. Anexos	33

LISTA DE FIGURAS

	Pág.
FIGURA 1. TUBERIA	9
FIGURA 2.TANQUE	9
FIGURA 3. ARDUINO UNO	11
FIGURA 4. ARDUINO MEGA	12
FIGURA 5. TRANSEIVER XBEE	18
FIGURA 6. COMUNICACIÓN INALAMBRICA	19
FIGURA 7. COORDINADOR PAN CON MÚLTIPLES NODOS.	20
FIGURA 8. COMUNICACIÓN PUNTO A PUNTO.	21
FIGURA 9. PC SETTINGS	24
FIGURA 10. MATLAB.	25
FIGURA 11. CONDUCTOS.	26
FIGURA 12. ESTRUCTURAS.	26
FIGURA 13. INVERNADERO.	27
FIGURA 14. SISTEMA DE CONTROL	27
FIGURA 15. TOMA DE DATOS	28

LISTA DE TABLAS

	Pág.
Tabla1: Requerimiento climáticos	3
Tabla 2: Comparativa de sensores más utilizados de temperatura.	4
Tabla 3. Comparativa de sensores más utilizados de Humedad.	8
Tabla 4: Comparativa De Arduino	13
Tabla 5: Comparativa De Atmega	14
Tabla 6: Comparativa Xbee	22

LISTA DE ECUACIONES

	Pág.
Ecuación 1. Humedad Relativa.	7
Ecuación 2. Capacitancia	7

INTRODUCCIÓN

La hidroponía es una técnica o método utilizado para cultivar plantas usando soluciones minerales sin suelo agrícola. Las raíces reciben una solución nutritiva equilibrada disuelta en agua con todos los elementos esenciales para el crecimiento de la planta, pueden crecer en una solución única y en medios inertes como arena lavada y perlita o grava.

En Colombia el control de variables ambientales para un cultivo suele ser complicado puesto que las condiciones ambientales varían mucho así que la mayoría de los cultivos suelen presentar pérdidas, esto pasa también con los nutrientes dado que en las tierras donde se cultivan son limitados y requieren aplicación de abonos para que los cultivos puedan avanzar en la etapa de desarrollo.

A través de este proyecto se pretende comprender lo relacionado con el diseño de un control ambiental, con la intención de poder ofrecer algunas soluciones adecuadas para las necesidades del sector agroindustrial del país, y así poder fomentar una mejora tecnológica significativa para el campo Colombiano.

En este proyecto se plantean algunas soluciones desde los conocimientos adquiridos durante el proceso de formación, aplicando protocolos de comunicación inalámbrica, que logran suplir las necesidades de control en el invernadero permitiendo la regulación de tres variables importantes del ambiente cerrado, como son temperatura, humedad y caudal de riego.

El objetivo general de este proyecto es diseñar un sistema de control para un cultivo hidropónico; para ejercer control en las variables críticas del ambiente, se establecerá un protocolo de comunicación inalámbrica para la transmisión de datos y la aplicación de una interfaz gráfica para la visualización de los mismos.

En el desarrollo del sistema de control se optó por utilizar un micro controlador de bajo costo pero eficiente, lo que permitió la adquisición y fácil acceso para la lectura de datos de los sensores aplicados para el proyecto, los cuales fueron calculados y procesados para así obtener una mejor respuesta.

Se busca además que el sistema sea competitivo en el mercado, para que los pequeños agricultores puedan acceder a él, y mejorar sus ingresos con el incremento en la producción de su cultivo

Se desarrolló una interfaz de trabajo que permite observar en el tiempo las variables sensadas permitiendo hacer seguimiento al comportamiento del sistema a lo largo del día.

1. CULTIVO PARA LA IMPLEMENTACIÓN DEL SISTEMA DE CONTROL

1.1 LECHUGAS HIDROPÓNICAS

En el mundo actual de la agroindustria la hidroponía es una técnica de cultivo que permite la implementación de ciertas hortalizas que se cultivan en una cama de agua permitiendo así la constante absorción de nutrientes de la planta fortaleciendo y mejorando sus características [1]. Por esta razón se optó por seleccionar un cultivo de lechugas para el diseño del sistema de control.

A continuación se muestran una serie de pasos sobre cómo se conforma un cultivo de lechugas hidropónicas.

1.1.1 Lechuga.

La lechuga es una hortaliza propia de las regiones semi-templadas que se cultivan con fines alimenticios. Las condiciones climáticas aptas para este tipo de hortaliza se dan en el piso térmico templado entre 1100 y 2100 m.s.n.m. [2]

1.1.2 Método de cultivo.

- a) **Siembra:** El método de siembra de esta hortaliza se realiza en charolas de plástico con 200 cavidades, se utiliza como sustrato principal la perlita, sembrando en cada cavidad 2 semillas con una profundidad de 5mm. [3]
- b) **Trasplante:** Una vez que transcurran de 30 a 40 días de la siembra, la lechuga se trasplanta, cuando tenga de 6 a 8 hojas y una altura aproximada de 8 centímetros desde el cuello del tallo hasta la punta de las hojas. [3]
- c) **Riego:** Se emplean 3 veces al día de 100 a 200 ml/planta, una opción para dar riego óptimo es el sistema de riego por goteo. Se aplica una solución nutritiva diariamente si el cultivo es un sustrato. [3]

1.1.3 Requerimientos climáticos del cultivo de lechuga.

Las condiciones climáticas para el desarrollo de la lechuga (ver Tabla 1) son de 15°C a 18°C, soportando una temperatura máxima de 24°C y una mínima de 7°C. [3]

Si la temperatura está por debajo o es igual a 7°C el desarrollo de la planta se detiene, mientras que si está por encima de 24°C el desarrollo de la hortaliza se acelera y por lo tanto la calidad de la planta se verá afectada debido a la acumulación de látex amargo en las venas. [3]

Tabla1: Requerimiento climático

Requerimiento	Rango
Temperatura	Germinación 19°C Desarrollo vegetativo: 16°C en el día y 6°C en la noche Formación del cogollo: 12°C en el día y 4°C en la noche
Humedad relativa	Entre 60 y 80%
pH	Entre 6.4 y 7.4
Riego	De 200 a 400 ml por planta

Fuente: www.hydroenv.com.mx
(Consultado: 5 de enero de 2014)

2. DETERMINACIÓN DE LOS SENSORES

2.1 SENSOR DE TEMPERATURA

Para lograr el control total en la industria es necesario tener sensores adecuados para cada variable que ofrezcan confiabilidad y eficiencia, para esto los expertos han desarrollado una variada gama de soluciones con muchos tipos de materiales, para satisfacer la demanda de los consumidores.

A continuación se muestra la clasificación de algunos sensores de temperatura para obtener una comparación entre ellos (ver Tabla 2) y así obtener las características que se acoplen a este proyecto.

Tabla 2. Comparativa de sensores más utilizados de temperatura.

Sensor	Vin Min	Vin Max	Tmin°C	Tmax°C	Tolerancia	Tipo de salida	Fabricante	Costo
Termocupla tipo J	-	-	-270	1200	1	-	Ajthermosensors	\$44.000
Termocupla tipo K	-	-	-200	1372	1	-	Ajthermosensors	\$12.000
RTD(PT100)	-	-	-50	250	Variable	-	LabFacility	\$18.000
LM35	4	30	-40	100	±1°C y ±1,5°C	Analógica	Texas Instruments	\$2.600

Fuente: Autores.

Al observar la Tabla 2 se muestran las características de los sensores más utilizados lo cual permite seleccionar el adecuado para el proyecto.

- **Termocupla Tipo J:** Se usa en gran cantidad de aplicaciones, tiene bajo costo y posee variedad de sondas. Tienen un rango de temperatura de -200 °C a +1372 °C y una sensibilidad de 41µV/°C aproximadamente. Posee buena resistencia a la oxidación. [4]

- **Termocupla Tipo K:** Se usa en el rango de $-270/+1200^{\circ}\text{C}$. Por sus características el fabricante recomienda su uso en atmósferas inertes, reductoras o en vacío, cuando se trabaja con ella continuamente a 800°C no presenta problemas, el mayor problema que tiene es la rápida oxidación que sufre el hierro por encima de 550°C y por debajo de 0°C es necesario tomar precauciones a causa de la condensación de vapor de agua sobre el hierro. [4]

La exactitud se ve limitada con temperaturas inferiores a 1°C ; su f.e.m de salida no es lineal así que el instrumento que mide su salida necesariamente debe incluir circuitos de linealización.

La aleación empleada en la construcción también influye en la exactitud del sensor.

Las termocuplas cumplen con el rango de temperatura para el proyecto pero la principal desventaja es que se necesita la construcción de circuitos de linealización lo cual puede aumentar los costos en la construcción del proyecto.

RTD (PT100): Detecta la temperatura por medio de una resistencia, basado en la variación de la resistencia de un conductor con la temperatura. [5]

Ventajas que proporciona una RTD:

- Margen de temperatura bastante amplio.
- Proporciona las medidas de temperatura con mayor exactitud y repetitividad.
- El valor de resistencia del RTD puede ser ajustado con gran exactitud por el fabricante (trimming), de manera que su tolerancia sea mínima. Además, éste será bastante estable con el tiempo.
- Los RTD son los sensores más estables con el tiempo, presentando variaciones en la medida del orden de 0.1°C/año . [5]
- La relación entre la temperatura y la resistencia es la más lineal.

Desventajas de la RTD:

- Los RTD se ven afectados por el auto calentamiento.
- Los RTD no son tan durables como los termopares ante vibraciones, golpe.

La RTD (PT100) es un sensor que garantiza un margen de temperatura amplio, además proporciona datos confiables que se ven afectados por el auto calentamiento debido al aumento de la corriente y son frágiles ante los golpes. [5]

Por último Se tomó el sensor LM35 por su resolución de 0,5 grados, es un dispositivo económico, además cumple con los rangos de medición de temperatura que se van a manejar en el cultivo hidropónico. Este dispositivo es diseñado por la empresa Texas Instruments, lo que garantiza que el producto es económico, disponible y se ajusta a las necesidades del proyecto. [6]

En cuanto a los sensores de temperatura se aclara que existen otras alternativas que por cuestiones de tiempo no se exploraron.

2.2 SENSOR DE HUMEDAD

Un sensor de humedad es un dispositivo que mide o detecta variables físicas que determinan el porcentaje de humedad.

2.2.1 Presión parcial de vapor (P_v).

Es definido como la parte de la presión atmosférica total ejercida por el vapor de agua contenido en la atmosfera.

2.2.2 Presión de vapor de saturación (P_{sat}).

Es presión establecida para una temperatura dada, donde la fase líquida y el vapor de agua se encuentra en un equilibrio dinámico, se define como la presión de la fase gaseosa para una temperatura determinada.

2.2.3 Humedad relativa (Hr).

La humedad relativa se define como la cantidad de vapor de agua presente en una masa de aire, en cuanto aumente la saturación este se aproximará al valor de la humedad relativa al 100% entonces estará más húmedo el ambiente.

Se Nota que si P_v es igual a P_{sat} , se tiene humedad relativa de 100%. (ver ecuación 1.)

$$Hr = \frac{P_v}{P_{sat}} \times 100\% \quad (1).$$

2.2.4 Sensores capacitivos.

Los sensores capacitivos son elementos electrónicos que detectan componentes metálicos y no metálicos que al aproximarse a la superficie activa del sensor sobrepasan una determinada capacitancia.

Este principio es aplicado al sensor de humedad relativa pero este funciona con respecto a un dieléctrico el cual detecta la presencia de agua suspendida en masas de aire y determinando el valor de la humedad del ambiente.

Si se utiliza en la mezcla gaseosa como dieléctrico entre las placas del condensador, el valor que se muestra en la siguiente ecuación (ver ecuación 2.) estará determinado por:

$$C = \epsilon \frac{A}{d} \quad (2).$$

Donde:

- C, es el valor de la capacidad.
- ϵ , es la permisividad del dieléctrico (constante dieléctrica).
- A, el área de las placas del condensador.
- D, la distancia entre las placas del condensador.

A continuación se mostrará una clasificación de los siguientes sensores para obtener una comparación entre ellos (ver Tabla 3), y así obtener las características principales que se adapten al proyecto dado.

Tabla 3. Comparativa de sensores más utilizados de Humedad.

Sensor	Vin Min	Vin Max	Tmin°C	Tmax°C	%RH Min	%RH Max	Precisión	Tiempo de Respuesta	Tipo de salida	Costo
SHT71	2.4	5.5	-40	123.8	0	100	+3.0 %RH +- 0.4 °C	-	Digital	115.000
HIH 4000	2.5	5.8	-	-	0	100	+3.5 %RH	-	Análoga	58.700
HIH 4030	2.5	5	-	-	0	100	+ 3.5 %RH	-	Análoga	43.500
DHT 11	2.5	5	0	50	20	90	+5% %RH +-2°C	2 seg	Digital	16.000

Fuente: Autores.

El SHT71 es un circuito integrado para la medición de temperatura y humedad relativa de precisión que entrega una salida digital. Con esto se obtienen mediciones rápidas, de calidad e inmunes a las perturbaciones externas. La desventaja de este sensor es su alto costo por lo que se descarta para el proyecto.

Las características del sensor de humedad HIH-4000 también fueron investigadas para este proyecto puesto que la salida de tensión lineal hacia posible conectarlo directamente a un controlador sin embargo por su alto costo no fue posible implementarlo.

El sensor de Humedad Relativa HIH - 4030 también fue investigado como opción posible puesto que es de precisión, entrega una salida de Tensión analógica que se puede conectar directamente a un micro controlador con convertidor analógico digital, y gracias a eso la tensión que entrega tiende a ser lineal y fácil de procesar, pero este dispositivo no fue incluido para el proyecto porque en el proyecto se buscaban sensores de bajo costo.

Finalmente se seleccionó el sensor de humedad relativa DHT11 porque sus características como bajo costo, disponibilidad del dispositivo y linealidad lo perfilaron como el que cubría las necesidades del invernadero que se plantearon, además tiene un bajo costo. [7]

2.3. CONTROL DE CIRCULACIÓN

Para lograr este control se hicieron pruebas del tiempo de llenado como se muestra en la Figura 1 y flujo de agua en las tuberías usadas de 1" (pulgada), se obtuvo un tiempo llenado de 42s con una bomba de 250 L/h, que son 2,915 L que se hacen fluir cada 5min, el tiempo se puede cambiar y con este la cantidad de agua, dependiendo del diámetro de las tuberías que se usen.

Figura 1. Tubería



En la figura 2 se observa un recipiente donde se almacena el agua para la circulación del agua, dentro del recipiente está sumergida la bomba que hace circular la solución nutritiva para alimentar las plantas.

Figura 2. Tanque



Fuente: Autores

3. DETERMINACIÓN DEL HARDWARE Y SOFTWARE PARA CONTROL DE VARIABLES

3.1 SELECCIÓN DEL HARDWARE PARA LA MANIPULACIÓN DE VARIABLES

Actualmente hay distintas plataformas electrónicas como el PLC y micro controladores PIC y arduinos, los cuales fueron sugeridos para este proyecto.

El PLC es un componente electrónico de control utilizado en la ingeniería y la automatización, se usa para diferentes procesos en grandes fábricas y montajes.

Para la utilización del PLC se necesita un software que permita programar y compilar el flujo del proceso, generalmente se necesitan licencias costosas por lo tanto se buscaron otras alternativas.

Los micro controladores PIC con componentes electrónicos creados por una de las empresas reconocidas Microchip, son componentes electrónicos capaces de hacer una serie de procesos para aplicaciones de robótica y en diferentes campos de la automatización, para los micro controladores PIC se requiere un software para desarrollar su código de programación, esto implica la obtención de una licencia para poder aplicar el diseño del código lo cual aumentaría el costo del proyecto y por lo tanto es descartado.

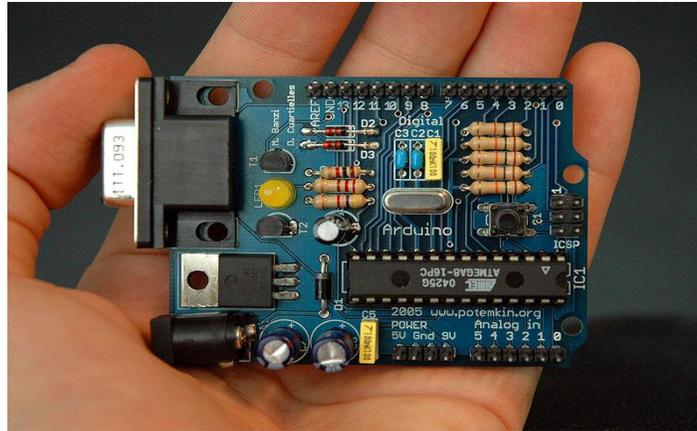
Anteriormente se trabajó con la plataforma de micro controladores PIC los cuales requirieren una serie de elementos adicionales para su funcionamiento, que son de bajo costo y eficientes. Presenta una serie de desventajas:

- **Requiere de un Quemador para poder descargar su código a la memoria interna del micro controlador.**
- **Requiere un diseño electrónico para el acoplamiento del micro controlador lo que aumentaría el costo del proyecto.**
- **Se requiere una licencia.**

Por otro lado Arduino es una plataforma de hardware libre, basada en una placa con un micro controlador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

En la Figura 3 se muestra la plataforma de Arduino que para este proyecto fue seleccionada por su fácil manejo, esta plataforma es de conexión directa a los sensores, bajo costo y eficiencia, es una base electrónica de código abierto lo que permite crear códigos de programación sin adquirir ninguna licencia en comparación a otras tarjetas de adquisición de datos, es una tarjeta que no requiere de componentes adicionales.

Figura 3. Arduino Uno



Fuente: www.wikipedia.org

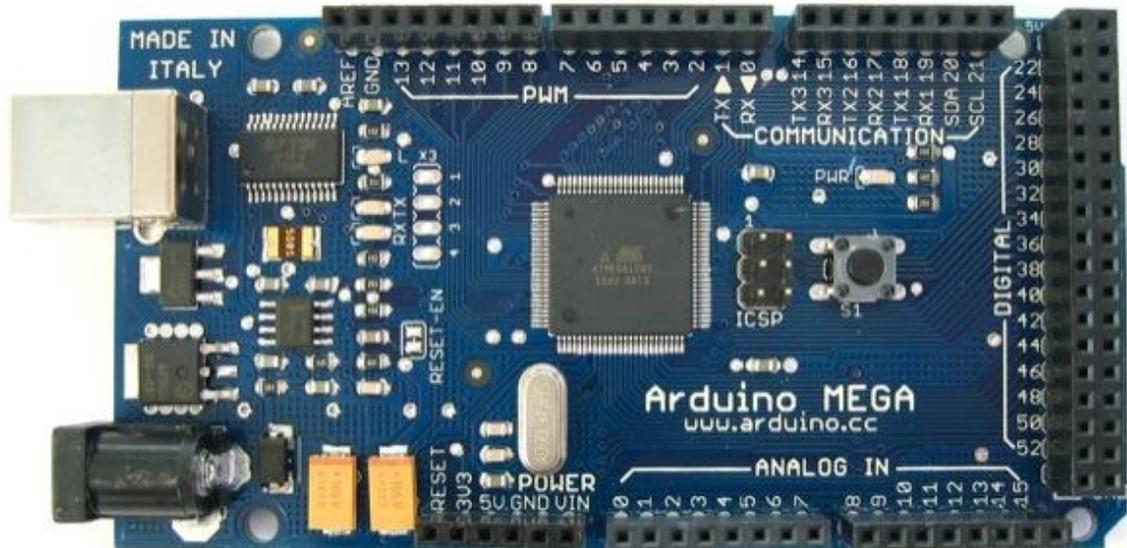
(Consultado: 5 de enero de 2014)

Al ser “*open-hardware*”, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto.

3.1.1 Arduino mega 2560.

El Arduino Mega en la Figura 4, es un controlador que poseen 54 puertos de entrada y salida tiene un procesador Atmega 1280, se seleccionó porque da la opción de conectar un número mayor de sensores al modulo permitiendo escalabilidad para el proyecto.

Figura 4. Arduino Mega



Fuente: www.arduino.cc
(Consultado: 5 de enero de 2014)

A continuación se muestra en la Tabla 4 una comparativa de los arduinos más utilizados actualmente.

Tabla 4: Comparativa De Arduinos

Características del Arduino	UNO	Mega 2560	Leonardo	DUE
Tipo De Micro controlador	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Velocidad De Reloj	16MHz	16MHz	16MHz	84MHz
Pines Digitales E/S	14	54	20	54
Entradas Analógicas	6	16	12	12
Salidas Analógicas	0	0	0	2(DAC)
Memoria De Programa (FLASH)	32Kb	256Kb	32Kb	512Kb
Memoria De Datos (SRAM)	2Kb	8Kb	2.5Kb	96Kb
Memoria Auxiliar (EEPROM)	1Kb	4Kb	1Kb	0Kb

Fuente: www.arduino.cc
(Consultado: 5 de enero de 2014)

Los micros controladores Arduino Diecimila, Arduino Duemilanove y Arduino Mega están basados en Atmega168, Atmega 328 y Atmega1280.

Al observar se seleccionó el Arduino mega 2560 por el alto número de puertos de comunicación, entradas y salidas lo que permite escalabilidad y garantiza el manejo de más cultivos. [8]

A continuación se observa (ver Tabla 5.) una tabla comparativa de los microcontroladores Atmega.

Tabla 5: Comparativa De Atmega

	Atmega168	Atmega328	Atmega1280
Voltaje operativo	5 V	5 V	5 V
Voltaje de entrada recomendado	7 - 12 V	7 - 12 V	7 - 12 V
Voltaje de entrada límite	6 - 20 V	6 - 20 V	6 - 20 V
Pines de entrada y salida digital	14 (6 proporcionan PWM)	14 (6 proporcionan PWM)	54 (14 proporcionan PWM)
Pines de entrada analógica	6	6	16
Intensidad de corriente	40 mA	40 mA	40 mA
Memoria Flash	16KB (2KB reservados para el bootloader)	32KB (2KB reservados para el bootloader)	128KB (4KB reservados para el bootloader)
SRAM	1 KB	2 KB	8 KB
EEPROM	512 bytes	1 KB	4 KB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz

Fuente: www.arduino.cc
(Consultado: 5 de enero de 2014)

3.2. DETERMINACIÓN DEL SOFTWARE PARA PROGRAMAR EL ARDUINO

3.2.1 Software Arduino.

En la actualidad se han desarrollado una serie de programas de licencia libre lo que permite la libertad de los usuarios para ejecutar, copiar, distribuir y estudiar el mismo e incluso modificar el software y distribuirlo modificado.

Para este proyecto se deseó aplicar un software que permita el desarrollo del código de programación para el funcionamiento del Arduino.

Se determinó la aplicación del software de Arduino, el cual puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto.

El entorno de Desarrollo Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.

3.3. LENGUAJE DE PROGRAMACIÓN ARDUINO

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el popular lenguaje de programación de alto nivel Processing. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino.

Algunos ejemplos son:

- Java
- Flash (mediante ActionScript)
- Processing
- Pure Data
- MaxMSP (entorno gráfico de programación para aplicaciones musicales, de audio y multimedia)
- VVVV (síntesis de vídeo en tiempo real)
- Adobe Director
- Python
- Ruby
- C
- C++ (mediante libSerial o en Windows)
- C#
- Cocoa/Objective-C (para Mac OS X)
- Linux TTY (terminales de Linux)
- 3DVIA Virtools (aplicaciones interactivas y de tiempo real)

- SuperCollider (síntesis de audio en tiempo real)
- InstantReality (X3D)
- Liberlab (software de medición y experimentación)
- BlitzMax (con acceso restringido)
- Squeak (implementación libre de Smalltalk)
- Mathematica
- Matlab
- Minibloq (Entorno gráfico de programación, corre también en OLPC)
- Isadora (Interactividad audiovisual en tiempo real)
- Perl
- PhysicalEtoys (Entorno gráfico de programación usado para proyectos de robótica educativa)
- Scratchfor Arduino (S4A) (entorno gráfico de programación, modificación del entorno para niños Scratch, del MIT)
- Visual Basic .NET
- Php [9]

Esto es posible debido a que Arduino se comunica mediante la transmisión de datos en formato serie que es algo que la mayoría de los lenguajes anteriormente citados soportan. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida. Es bastante interesante tener la posibilidad de interactuar Arduino mediante esta gran variedad de sistemas y lenguajes puesto que dependiendo de cuales sean las necesidades del problema que vamos a resolver podremos aprovecharnos de la gran compatibilidad de comunicación que ofrece. [9]

4. SISTEMA INALÁMBRICO

4.1. MODULO INALÁMBRICO XBEE SERIE 1

En el mundo actual existen una serie de módulos inalámbricos como bluetooth, los módulos RF y Xbee que permiten la transmisión de datos ahorrando así conexiones de cables que implican costos en un proyecto.

En el proyecto se presentaron una serie de soluciones para la transmisión de datos por vía inalámbrica los cual se presentarán a continuación:

4.1.1 Bluetooth.

Es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. [10]

Ventajas del bluetooth:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre éstos.

Desventajas:

- Limitado radio de acción entre los periféricos (30 pies entre ellos). Luego de esa distancia no hay garantías de transmisión adecuada de datos.

4.1.2 Radio Frecuencia.

Una red de área local por radio frecuencia o wlan (wireless lan) puede definirse como una red local que utiliza tecnología de radio frecuencia para enlazar los equipos conectados a la red en lugar de los medios utilizados en las LAN convencionales cableadas. [11]

Ventajas:

- Movilidad
- Simplicidad.
- Flexibilidad en la instalación
- Escalabilidad

Desventajas:

- Costo de inversión inicial alto.

4.1.3 Xbee.

Los módulos xbee (Figura 5) son componentes electrónicos que facilitan la comunicación inalámbrica, tienen un área de cobertura de más de 100mts de comunicación vía libre, se evita la utilización de cables y son totalmente compatibles con Arduino; además se pueden aplicar topologías para generar redes de comunicación entre ellos. [12]

Figura 5. Transeiver Xbee



Fuente: www.Dynamoelectronics.com
(Consultado: 10 de enero de 2014)

Una red Zigbee la pueden formar, teóricamente, con hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. [12]

Entre las necesidades que satisface el módulo se encuentran:

- Bajo costo.
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

El uso del protocolo Zigbee facilita el uso de protocolos de comunicación inalámbrica, logrando generar redes de comunicación entre los equipos conectados y entrega los datos para ser enviados a través de la red como se muestra en la Figura 6. [12]

Figura 6. Comunicación inalámbrica.



Fuente: www.olimex.com
(Consultado: 10 de enero de 2014)

Una red Zigbee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (endpoints).

4.1.4 El Coordinador.

Es el nodo de la red que tiene la única función de formar una red (ver figura 7).

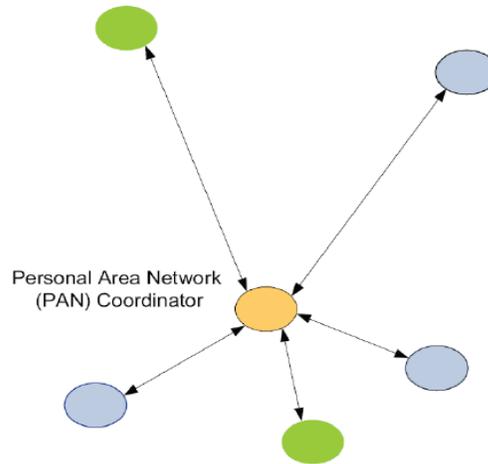
4.1.5 Los Routers.

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información.

4.1.6 Aplicaciones.

Estos módulos Xbee, pueden ser ajustados para usarse en redes de configuración punto a punto, punto a multipunto o peer-to-peer.

Figura 7. Coordinador PAN con múltiples nodos.



Fuente: www.olimex.com
(Consultado: 10 de enero de 2014)

4.2 MODO DE RECIBIR / TRASMITIR

Se encuentra en estos modos cuando al módulo le llega algún paquete RF a través de la antena (modo de recibido) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmitido).

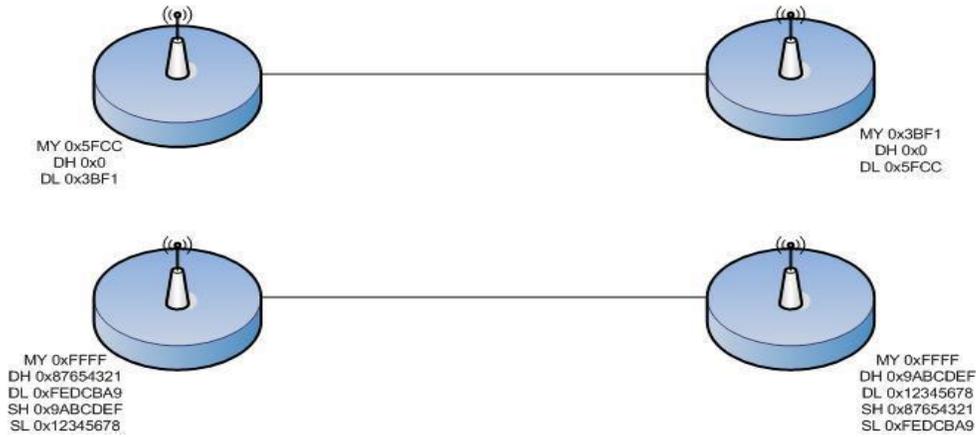
La información transmitida puede ser Directa o Indirecta. En el modo directo la información se envía inmediatamente a la dirección de destino. En el modo Indirecto la información es retenida durante un período de tiempo y es enviada sólo cuando la dirección de destino la solicita. [12]

4.3 CONFIGURACIÓN DEL MODULO XBEE

4.3.1 Conexión Punto a Punto.

Esta conexión se presta para sustituir la comunicación por un cable serial como se muestra en la figura (ver figura 8). [12]

Figura 8. Comunicación Punto a Punto.



Fuente: www.olimex.com
(Consultado: 10 de enero de 2014)

A continuación se observa una comparativa de los módulos inalámbricos Xbee serie 1 y serie 2 (ver tabla 6).

Tabla 6: Comparativa Xbee [13]

	Xbee 1mW 802.15.4 Chip Antenna Wire Antenna	Xbee PRO 60mW 802.15.4 Chip Antenna Wire Antenna	Xbee Series 2 1.25 mWZigBee Wire Antenna	XBee Series 2 PRO 63 mWZigBee Wire Antenna
General				
Compatibility	Series 1		Series 2	
Digi part #	XB24-ACI-001 XB24-AWI-001	XBP24-ACI-001 XBP24-AWI-001	XB24-Z7WIT - 004	XBP24-Z7WII - 004
Parallax part #	32404 32405	32406 32407	32408	32409
Antenna	Chip Wire	Chip Wire	Wire	Wire
Performance				
Rd data Range	250 kbps/up to 115 kbps	250 kbps/up to 115 kbps(1)	250 kbps/up to 1 115 kbps(1)	250 kbps/up to 1 115 kbps(1)
Indoor Range	100ft(30m)(2)	300ft(90m)(2)	133ft(40m)(2)	300ft(90m)(2)
Outdoor Range	300ft(100m)(2)	1 mile(1.6km)(2)	400ft(120m)(2)	2 mile(3.2km)(2)
Transmit Power	1mw(+0 dbm)	60 mw(+18 dbm)	1.25 mw(+1 dbm)	63mw(+18 dbm)
Receiver sensitivity	-92 dbm	-100 dbm	-98 dbm	-102 dbm
Features				
Serial data Interface	CMOS UART (no configuration needed)			
Configuration method	API or AT commands			
Frequency band	2.4 GHz			
ADC input	(7)10-bits ADC inputs			
Digital I/O	(8)Digital I/O		(10)Digital I/O	
Networking And Security				
Topology	Peer to peer, point to point, point to multipoint			
encryption	128 bits AES			

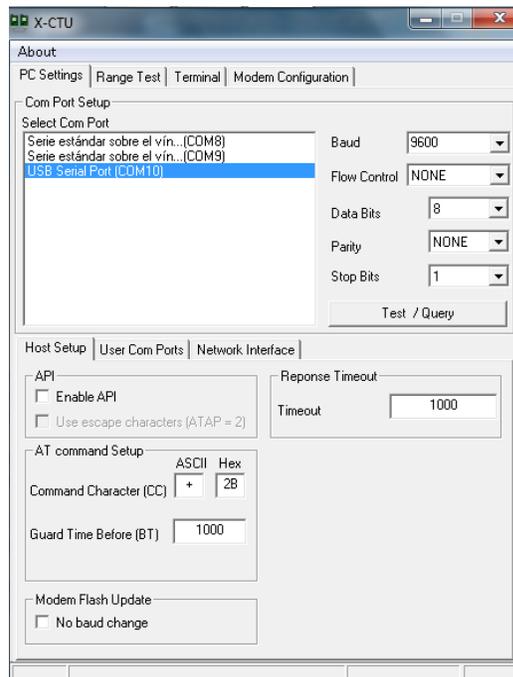
Reliable Packet Delivery	Retries/acknowledgments			
Channels	16 direct Sequence channels	12 direct Sequence channels	16 direct Sequence channels	12 direct Sequence channels
ID	PAN PID, 64 bits IEEE MAC			
Power Requirements				
Supply voltage	1.8 – 3.4 VDC (absolute)		2.1 - 3.6 VDC	2.7 - 3.6 VDC
Transmit current	45mA	215mA	35mA	205mA
Receive Current	50mA	55mA	38mA	48mA
Power-down current	<10 uA		<1uA	<3.5uA
Regulatory Approvals				
FCC(USA)	Our-Xbee	Our-XbeePro	Yes	
IC(Canadá)	4212A-Xbee	4212A-XbeePro	Yes	
ETSI(Europe)	Yes			
C-TICK(Australia)	Yes			
Telec (Japan)	Yes			

Fuente: www.olimex.com
(Consultado: 10 de enero de 2014)

5. SOFTWARE X-CTU

El software X-CTU es un programa que permite la configuración de los módulos inalámbricos Xbee (ver Figura 9).

Figura 9. PC Settings



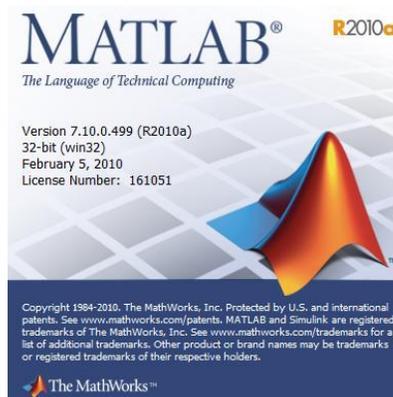
Fuente: www.olimex.com
(Consultado: 10 de enero de 2014)

Este software es en el que se puede direccionar cada modulo, configurar y establecer diferentes tipos de velocidades de transmisión de datos, en este software se pueden hacer pruebas con los xbee. [14]

6. SELECCIÓN DE SOFTWARE PARA GRAFICAR LA RESPUESTA DE LAS VARIABLES

Se escogió el programa Matlab (ver Figura 13.) puesto que es un software que se ha usado en proyectos anteriores y presenta una gran versatilidad a la hora de programar y administrar datos, facilitando así el diseño de la interfaz gráfica deseada para el proyecto, este software avanzado crea una interfaz denominada (GUI).[15]

Figura 10. Matlab.



Fuente: www.Mathworks.com
(Consultado: 10 de enero del 2014)

7. CONTRUCCION DEL INVERNADERO

Para el proyecto se construyó un invernadero a escala para poner en funcionamiento el sistema de control ya establecido.

En la figura 14. El sistema de circulación hecho en tubería de PVC con los orificios donde se introducirán las hortalizas.

Figura 11. Conductos.



Fuente: autores

En la figura 15. Se muestra la estructura hecha en madera del invernadero donde se pondrá el sistema de circulación.

Figura 12. Estructura.



Fuente: Autores

En la figura 16. Se observa de forma detallada, el invernadero construido con el sistema de circulación de agua y un telón de plástico que rodea la estructura para mantener las condiciones ambientales adecuadas para las hortalizas. Uno de los problemas que se presentaron fue que no se logro el resultado esperado porque la bomba que se utilizó no contaba con la presión suficiente para efectuar una aspersión adecuada.

Figura 13. Invernadero



Fuente: autores

En la figura 17. Se presenta la tarjeta de control con los valores de temperatura y humedad relativa ya establecidos para la activación de los actuadores.

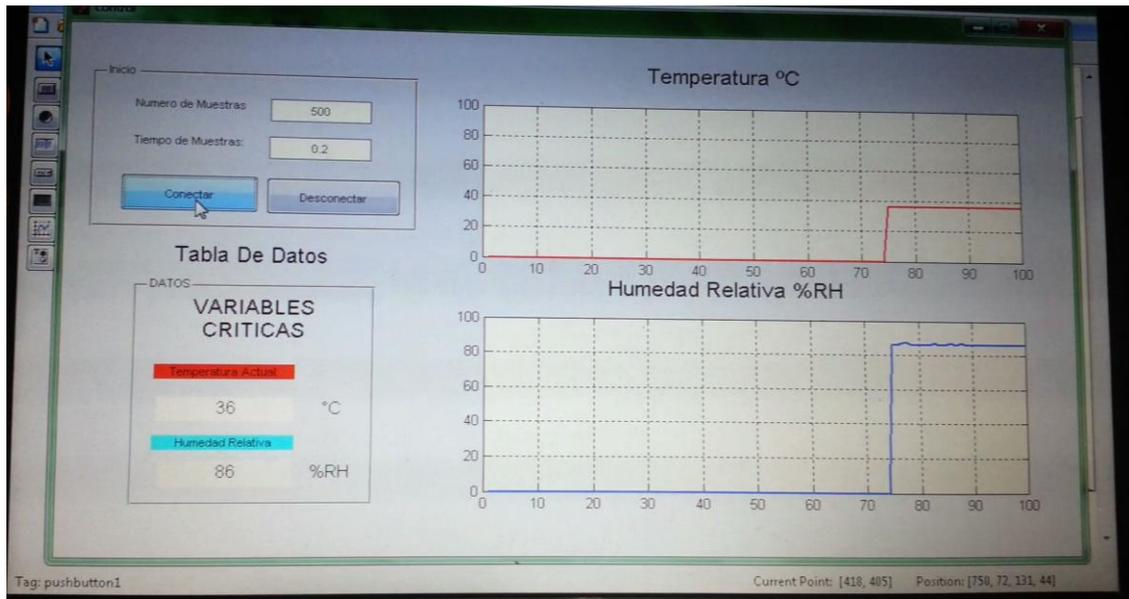
Figura 14. Sistema de control.



Fuente: Autores

Finalmente en la figura 18. Se ve como la interface gráfica interactúa con la tarjeta de control para la toma de datos y se generan las gráficas.

Figura 15. Toma de Datos.



Fuente: Autores.

8. CONCLUSIONES

1. En las pruebas realizadas con el sensor de temperatura LM35 y el sensor de Humedad relativa DHT11 se determinó que la respuesta es adecuada para el tipo de sistema que se está implementado debido a que en las pruebas realizadas las variaciones fueron mínimas y permite la toma de datos eficientemente para su medición.
2. Se concluye que la elección del Arduino mega 2560 fue acertada porque éste dispositivo cumplió con los resultados esperados, durante las pruebas no presentó fallas, ni recalentamiento, los datos fueron recibidos y procesados adecuadamente, proporcionando escalabilidad de tal forma que se pueden controlar más invernaderos con el mismo controlador.
3. Se encontró también que la programación y el acondicionamiento de señales que se presentan en este proyecto permitieron desarrollos a nivel casero para producción a baja escala y economía.
4. Se comprobó en las pruebas con los módulos inalámbricos XBee su excelente tiempo de respuesta, el rango de cobertura y su velocidad de transmisión de datos, estos lograron satisfacer las metas propuestas y funcionaron correctamente, no presentaron fallas por ruidos eléctricos que se presumía podría generar complicaciones.

9. RECOMENDACIONES

1. Para las personas que deseen realizar una investigación más exhaustiva sobre este tema, se recomienda hacer la búsqueda de las normas que rigen el rango de medición de Temperatura y de Humedad Relativa en cultivos hidropónicos en organizaciones internacionales, ya que en Colombia no existe este tipo de información de manera específica.
2. Implementar un software especializado que tome las variables del sistema y por medio de este programa las almacene en bases de datos en tiempo real para un mayor estudio y un control más eficiente.
3. Integrar los diferentes campos de la Ing. electrónica y la Ing. de sistemas, además integrar conocimientos de la industria agropecuaria con el fin de realizar estudios más confiables antes de la implementación, para este tipo de proyecto y así minimizar errores de construcción y sobrecostos.
4. Implementar las topologías que entregan los módulos Xbee para el manejo de varios invernaderos, lo cuales proporcionan escalabilidad en el proyecto.
5. Implementar un software de interfaz gráfica de código abierto como Scilab, para la reducción de costos en el proyecto.
6. Si se desea implementar otro cultivo se deben investigar sus condiciones óptimas de crecimiento para lograr que funcione correctamente el control.

10. BIBLIOGRAFÍAS

- [1] **Hidroponía**, Wikipedia®, enciclopedia virtual, Wikimedia Fundación (Agricultura Hidropónica), Tomada el 13 de Abril del 2012, internet: (<http://es.wikipedia.org/wiki/Hidropon%C3%ADa>)
- [2] **Condiciones Climáticas** (Pisos Térmicos) tomado el 5 de enero del 2014, Internet: (<http://colombiamitierrita.blogspot.com/2012/03/pisos-termicos.html>)
- [3] **Cultivo de la Lechuga Hidropónica** (Lactuca Sativa C.) tomado el 4 de enero del 2014, Internet: (http://www.hydroenv.com.mx/catalogo/index.php?main_page=page&id=52)
- [4] **Termocupla Tipo J y K** (Sensores De Temperatura) Tomado el 4 de enero del 2014, Internet: (<http://es.wikipedia.org/wiki/Termocupla>).
- [5] **RTD** (Sensor De Temperatura) Tomado el 4 de enero del 2014, Internet: (<http://es.wikipedia.org/wiki/RTD>)
- [6] **Sensor de Temperatura LM35** (Precision Centigrade Temperature Sensors), Tomado el 19 de Junio del 2013. Internet: (<http://www.ti.com/lit/ds/symlink/lm35.pdf>).
- [7] **Sensor de Temperatura y Humedad Relativa DHT11** (DHT11 Humidity & Temperature Sensor), Tomado el 19 de Junio del 2013. Internet: (<http://www.micro4you.com/files/sensor/DHT11.pdf>).
- [8] **Board Arduino Mega 2560** (Arduino) Tomado el 19 de Junio del 2013. Internet: (<http://arduino.cc/en/Main/arduinoBoardMega2560>).
- [9] **Lenguaje de Programación Arduino** (Arduino) Tomado el 4 de enero del 2014. Internet: (<http://es.wikipedia.org/wiki/Arduino>).
- [10] **Comunicación Bluetooth** (Comunicación Inalámbrica) Tomado el 4 de enero del 2014. Internet: (<http://es.wikipedia.org/wiki/Bluetooth>)
- [11] **Comunicación Radio Frecuencia** (Comunicación Inalámbrica) Tomado el 4 de enero del 2014. Internet: (<http://es.wikipedia.org/wiki/Radiofrecuencia>.)
- [12] **Andrés Oyarce, Xbee Serie 1 (Xbee)**, Tomado el 15 de agosto del 2013, Pág. 6 a 14.

[13] **Tabla comparativa Xbee (Módulos Inalámbrico)**, Tomado el 15 de agosto del 2013, Internet: (http://www.rambal.com/index.php?option=com_content&view=article&id=65).

[14] **Software X-CTU, Andrés Oyarce, Xbee Serie 1 (Guías de usuario)**, Tomado el 15 de agosto del 2013, Pág. 51.

[15] **Acercas de Interface Gráfica GUIDE (MATLAB)**, Tomado el 15 de agosto del 2013), Pág. 1 a 6.

11. ANEXOS

CONTENIDO

1. Ficha Técnica De Los Componentes Electrónicos.

- 1.1. Módulo Xbee Serie 1.
- 1.2. Arduino Mega 2560.
- 1.3. Sensor de Temperatura LM35.
- 1.4. Sensor de Humedad Relativa HIH 4030 SparkFun.

2. Diseño De Etapa De Potencia.

- 2.1. Materiales.
- 2.2. Diseño Eléctrico.
- 2.3. Construcción.

3. Interface Gráfica (MATLAB).

- 3.1. Interface de Diseño GUIDE MATLAB.
- 3.2. Código de Programación GUIDE MATLAB.
- 3.2. Visualización.

4. Código De Adquisición De Datos ARDUINO.

- 2.2 Código Arduino.

5. Diseño y Estructura del Invernadero.

- 5.1. Materiales.
- 5.2. Diseño.
- 5.3. Vistas Del Diseño.
- 5.4. Video.

1. FICHA TÉCNICA DE LOS COMPONENTES ELECTRÓNICOS.

1.1. MODULO XBEE SERIE 1.

Detalles	XBEE 802.15.4 (Series 1)
Tasa de datos	254 Kbps
Rango recinto interior	30 m
Rango línea de visión	100 m
Potencia de Transmisión	1 mW (+0 dBm)
Sensibilidad del receptor (1% PER7)	-92 dBm
Características	
Interfaz de Datos Serial	3.3V CMOS UART
Métodos de Configuración	Comandos AT o API, local o sobre el aire.
Banda de frecuencia	2.4 GHz
Inmunidad a la interferencia	DSSS (Direct Sequence Spread Spread Spectrum)
Tasa de Comunicación Serial	1200 bps a 250 Kbps
Conversores ADC	Conversores de 10 bit (hasta 7 disponibles).8
I/O Digitales	Hasta 8 disponibles.
PWM (Pulse Width Modulation)	Hasta 2 disponibles.
Opciones de Antena	Chip, Wire Chip, U.FL y RPSMA.
Redes y Seguridad.	
Encriptación	128 bits
Entrega confiable de paquetes.	Retries/Acknowledgments (Reintentos/aviso de recibo de paquetes)
ID's y canales.	PAN ID (personal Area Network), 64-bit IEEE MAC, 16 canales.
Requerimientos de poder	
Voltaje de Alimentación	2.8 a 3.4 Vdc
Corriente al transmitir.	45 mA @ 3.3 Vdc
Corriente al recibir.	50 mA @ 3.3 Vdc
Corriente de apagado.	<10uA @ 25 °C
Regulaciones aprobadas.	
FCC (USA)	OUR-XBEE
IC (Canadá)	4214 ^a -XBEE
ETSI (Europa)	Si
C-TICK Australia	Si
Telec (Japón)	Si

Fuente: www.olimex.com

2.2 ARDUINO MEGA 2560.

Microcontroller	ATmega2560
OperatingVoltage	5V
Input Voltage	7-12V
Input Voltage	6-20V
Digital I/O Pins	54 (of which 15 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
ClockSpeed	16 MHz

www.arduino.cc

a. SENSOR DE TEMPERATURA LM35.

Características
Sus características más relevantes son:
<ul style="list-style-type: none">• Está calibrado directamente en grados Celsius.• La tensión de salida es proporcional a la temperatura.• Tiene una precisión garantizada de 0.5°C a 25°C.• Opera entre 4 y 30 voltios de alimentación.• Baja impedancia de salida.• Baja corriente de alimentación (60uA).• Bajo costo.

Fuente: www.ti.com

b. SENSOR DE HUMEDAD RELATIVA HIH 4030 SPARKFUN.

Tabla 2. Especificaciones de Rendimiento.

Parámetros	mínimo	Típico	Máximo	Unidades	Nota específica
Intercambiabilidad	-	-	-	-	-
0% RH a 59% RH	-5	-	5	%RH	-
60% RH a 100% RH	-8	-	8	%RH	-
Precisión	-3.5	-	+3.5	%RH	1
Histéresis	-	3	-	%RH	-
Repetitibilidad	-	±0.5	-	%RH	-
Tiempo de estabilización.	-	-	70	Ms	-
Tiempo de Respuesta.	-	5	-	S	-
Estabilidad (al 50% de humedad relativa en un año)	-	±1.2	-	%RH	2
Estabilidad (al 50% de humedad relativa en un año)	-	±0.5	-	%RH	3
Alimentación de tensión	4	-	5.8	Vdc	4
Suministro de Corriente.	-	200	500	µA	-
Salida de Tensión.	$V_{OUT} = (V_{SUPPLY})(0.0062(\text{sensor RH}) + 0.16)$, typical at 25 °C				
Compensación de temperatura.	True RH = (Sensor RH)/(1.0546 – 0.00216T), T in °C				
Temp.Tensión de salida. Coeficiente al 50% de HR, 5 V	-	-4	-	mV/°C	-
Temperatura de Trabajo	-40[-40]	-	85[185]	°C[°F]	-
Temperatura de Almacenamiento.	-50[-58]	-	125[257]	°C[°F]	-

8. DISEÑO DE ETAPA DE POTENCIA.

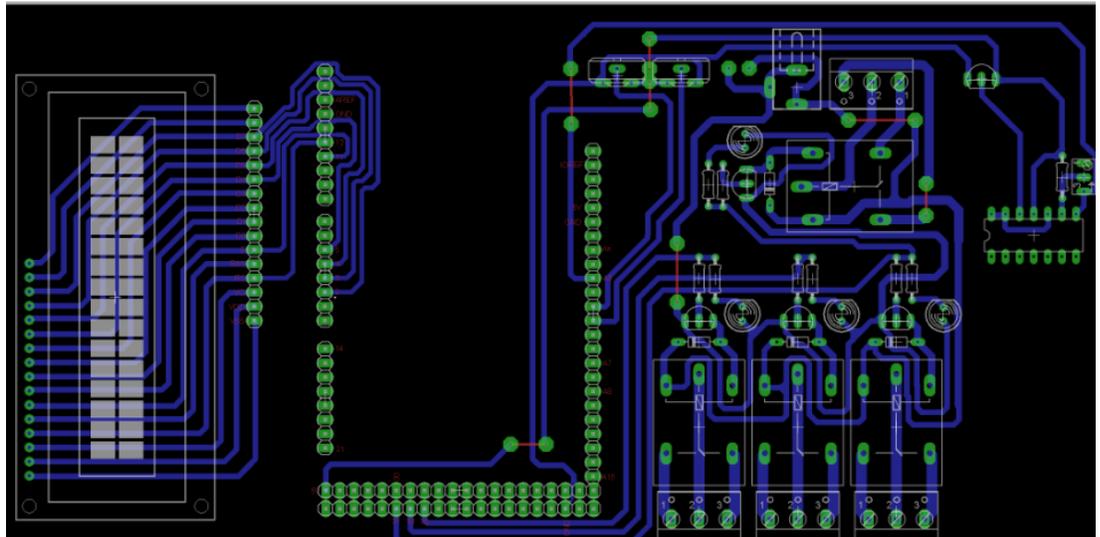
2.1 MATERIALES.

- 1 Arduino Mega 2560.
- 1 LCD 16x2 alfanumérica.
- 4 Relés de 12v.
- 1 LM324.
- 4 Borneras Triples.
- 4 transistores 2n2222.
- 1 Xbee Shields.
- 1 Xbee Serie 1.

2.2 DISEÑO ELÉCTRICO.

A continuación se mostrara el circuito propuesto Diseñado en Eagle.

Figura 1. Circuito Eléctrico.

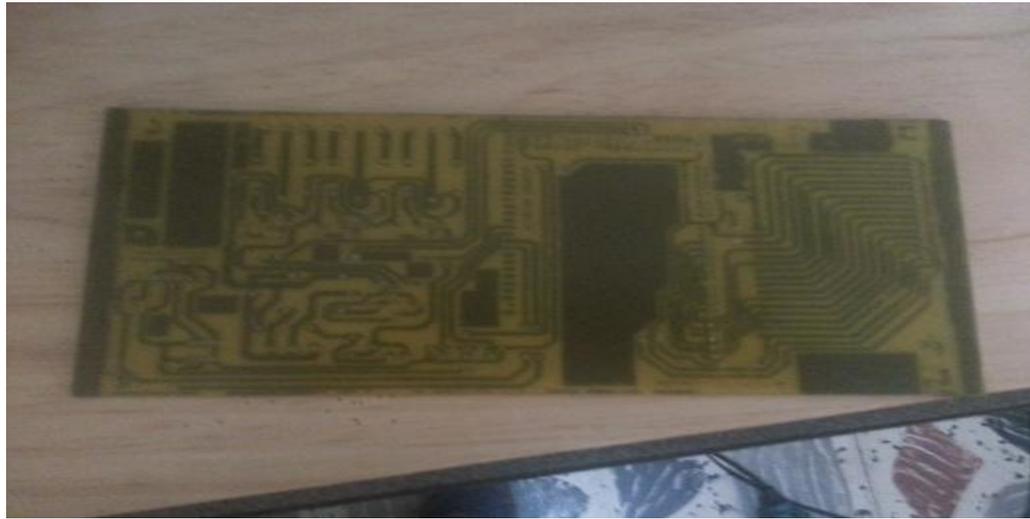


Fuente: Autores.

a. Construcción.

En la figura 2. Se observa la tarjeta con el diseño del circuito ya quemado.

Figura 2. Circuito quemado



Fuente: Autores

En la figura 3. Se ve la etapa de potencia ya implementada en el circuito de control con sus respectivos componentes.

Figura 3. Interfaz de potencia



Fuente: Autores

En la figura 4. Se implementa en la tarjeta de control el Arduino con el sistema de comunicación inalámbrica Xbee y la pantalla LCD para el control de variables.

Figura 4. Tarjeta de control



Fuente: Autores

Se muestra en la figura 5. El funcionamiento de la tarjeta de control ya terminado.

Figura 5. Tarjeta de control terminada



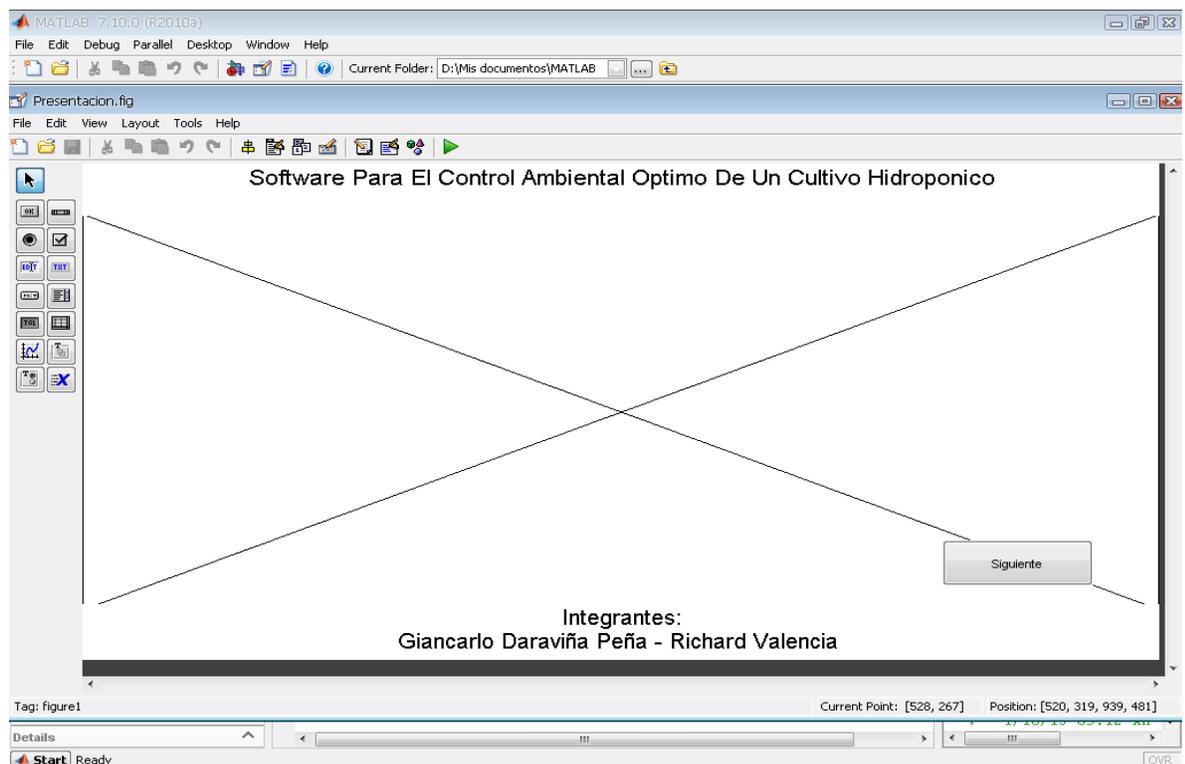
Fuente: Autores

9. INTERFACE GRAFICA (MATLAB).

3.1 INTERFACE DE DISEÑO GUIDE MATLAB.

A continuación se muestra la interface gráfica del software para el invernadero hidropónico donde se muestran las variables críticas a medir.

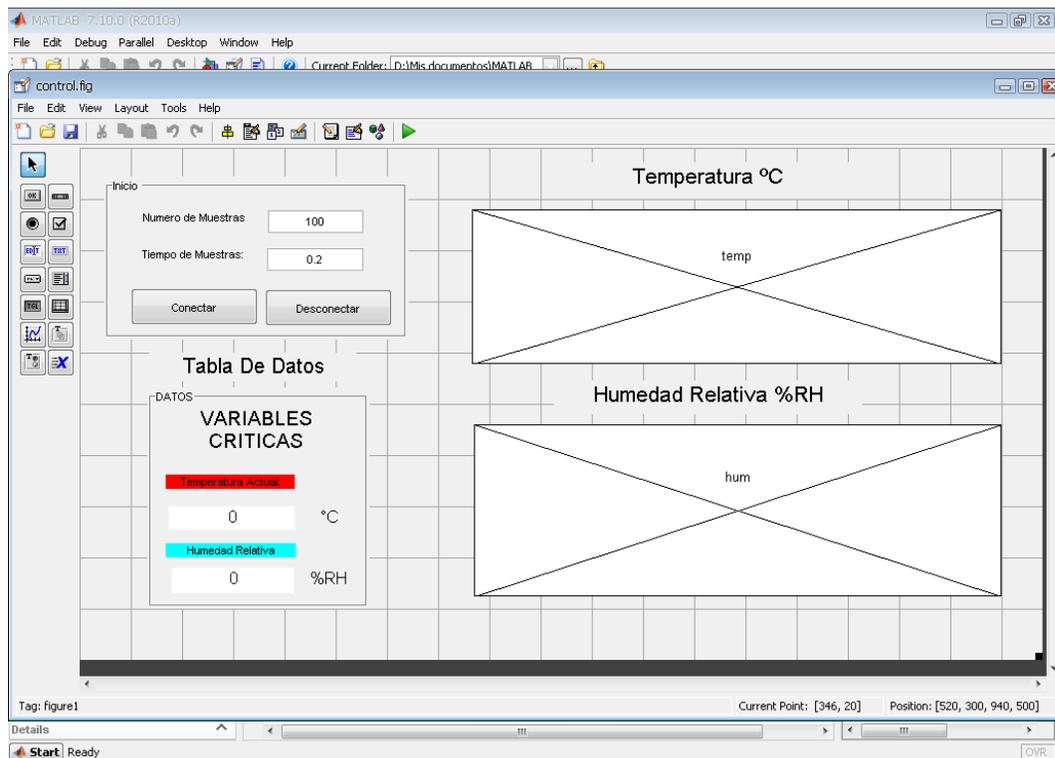
Figura 6. Presentación.



Fuente: Autores

En la figura 7 se muestra la interface de control hecho en Matlab, se observa una tabla de muestras en la cual se ingresan el tiempo de muestreo y las cantidades de muestras a medir, en la parte inferior izquierda se ve la tabla de datos donde se verán las variables críticas a medir y finalmente dos gráficos donde se verán los datos de las variables.

Figura 7. Control.



Fuente: Autores

3.2 CÓDIGO DE PROGRAMACIÓN GUIDE MATLAB.

CODIGO GUI MATLAB

```
function varargout = control (varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before control is made visible.
function control_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to control (see VARARGIN)

% Choose default command line output for control
handles.output = hObject;
warning off;
% handles.stop = 1;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes control wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
```

```

function varargout = control_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Desconectar1.
function Desconectar1_Callback(hObject, eventdata, handles)
% hObject handle to Desconectar1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

delete(instrfind({'Port'}, {'COM3'})); %desconecta el puerto
%limpia los AXES
cla(handles.temp, 'reset')
cla(handles.hum, 'reset')

%limpia los datos del Texto Estatico
clear_Dtemp=findobj(gcf, 'tag', 'Dtemp');
clear_DRH=findobj(gcf, 'tag', 'DRH');
limpiar='';
set(clear_Dtemp, 'string', limpiar);
set(clear_DRH, 'string', limpiar);
clc;
clear;

% --- Executes during object creation, after setting all properties.
function t_CreateFcn(hObject, eventdata, handles)
% hObject handle to t (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)
% hObject handle to start (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% delete(instrfind({'Port'}, {'COM4'}))

start = serial('COM3', 'BaudRate', 19200);
% handles.stop = 1;

```

```

fopen(start);

data = 1:100;
data1 = zeros(size(data)); %datos de almacenamiento (Trama 1)
data2 = zeros(size(data)); %datos de almacenamiento (Trama 2)
%Bucle while para que tome y dibuje las muestras que queremos
while contador_muestras<=numero_muestras

    Datos=fscanf(start, '%d,%d'); %lectura del datos

data1(1:end-1)=data1(2:end);
data1(end) = Datos(1); % lectura de la primera trama de datos
(vlaor_LM35)

% grafica Donde se muestran los datos de temperatura
axes(handles.temp)
plot(data,data1,'r','LineWidth',2);
ylim([0 100])
xlim([0 100])
gridon;

% grafica Donde se muestran los datos de Humedad
axes(handles.hum)
plot(data,data2,'b','LineWidth',2);
ylim([0 100])
xlim([0 100])
gridon;

% drawnow
contador_muestras = contador_muestras + 1;

%se muestra la trama de datos en un texto estatico

set(handles.Dtemp,'string',Datos(1));
set(handles.DRH,'string',Datos(2));
end

% guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function temp_CreateFcn(hObject, eventdata, handles)
% hObject handle to temp (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called

```

```

% Hint: place code in OpeningFcn to populate temp

% --- Executes during object creation, after setting all properties.
function hum_CreateFcn(hObject, eventdata, handles)
% hObject    handle to hum (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate hum

% --- Executes during object creation, after setting all properties.
function m_CreateFcn(hObject, eventdata, handles)
% hObject    handle to m (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
set(hObject, 'BackgroundColor', 'white');
end

% --- Executes during object creation, after setting all properties.
function start_CreateFcn(hObject, eventdata, handles)
% hObject    handle to start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function DRH_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DRH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% --- Executes during object creation, after setting all properties.
function Dtemp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Dtemp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

```

3.2 VISUALIZACIÓN.

Presentación Del Software

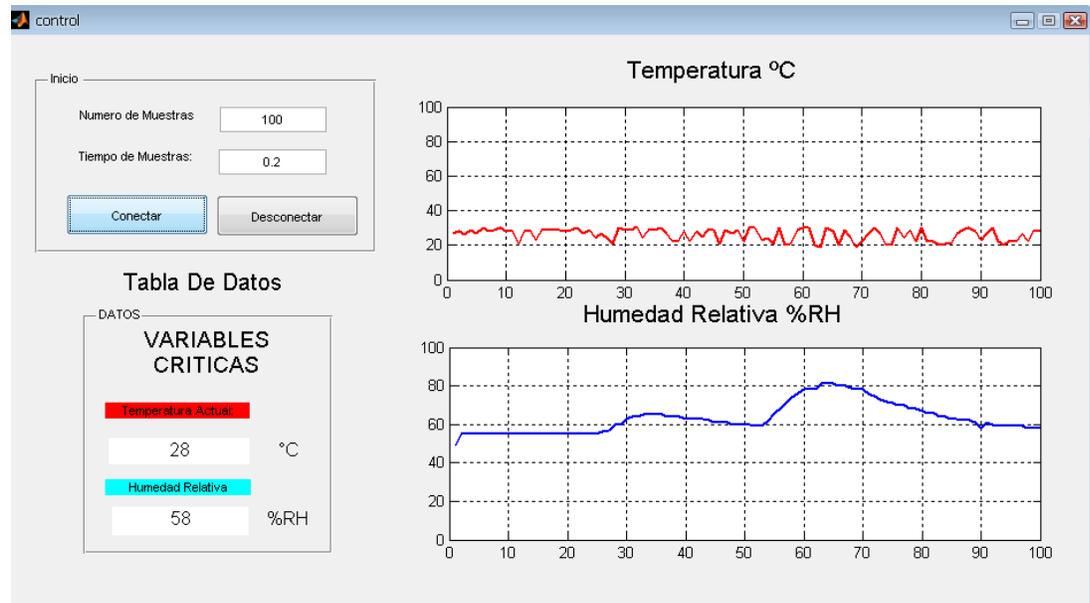
En la primera etapa del software se muestra una presentación de la interface gráfica, como se ve en la figura 8.

Figura 8. Presentación.



En la segunda etapa del programa se visualiza la interface donde se grafican los datos de las variables críticas y sus respectivos gráficos.

Figura 9. Interface Toma De Datos GUI.



Fuente: Autores

Este software posee las siguientes funciones donde el usuario puede acceder a ellas.

En la parte superior tiene un INICIO donde se encuentran Dos variables editables que son el Número De Muestras y El Tiempo De Muestras.

Al acceder a estas funciones el usuario puede editar el número de muestras que se deseen.

El Tiempo de Muestreo al acceder esta función se puede introducir cualquier valor numérico donde este dado en segundos y se puede introducir el valor que se desee.

También se muestran Dos pulsadores, Conectar Y Desconectar.

El Botón conectar inicia la comunicación con la tarjeta de adquisición de datos donde ya son procesados y tomados por el software.

El Botón desconectar Permite la Desconexión de la tarjeta y El reinicio de los datos y las gráficas.

10. CÓDIGO DE ADQUISICIÓN DE DATOS ARDUINO.

El siguiente código de programación Hecho en Arduino Toma, procesa y envía los datos en modo Serial Tomados de los sensores.

```
#include<LiquidCrystal.h>
////////////////////CONFIGURACION DE PINES PARA LA LECTURA DEL LOS
SENSORES////////////////////
int SensorLM35 = A0; // Lectura del Sensor de Temperatura LM35
inthumPin = A1; //Lectura del Sensor de Humedad HIH4030
intsensordist = A2; // lectura del Sensor de Distancia GP2Y0A02YK0F SHARP
int PotPinLM35 = A3; //Set-Point del LM35 para ajustar el Valor a medir
int potPinHIH4030 = A4; //Set-Point del HIH4030 para ajustar el Valor a medir

////////////////////SET-POINT////////////////////

int valor_LM35 = 0; // Variable donde se guardan Valores del lm35
intValor_dist = 0;
int potLM35 = 0; //Variable donde se almacena el Valor del setpoin del LM35
intpotHIH = 0; //Variable donde se almacena el valor del setpoint del HIH4030
intpotdist = 0;

////////////////////PINES PARA LA ACTIVACION DE
ACTUADORES////////////////////

int PinOut1=31;
int PinOut2=33;
int PinOut3=35;

LiquidCrystallcd(12, 11, 5, 4, 3, 2); // Configuración del La pantalla LCD

//INCIO DE PROGRAMA//
voidsetup()
{
Serial.begin(9600);
lcd.begin(16, 2);
lcd.print("SetP.HUM ");
lcd.setCursor(0, 1);
lcd.print("SetP.TEMP ");
```

```

lcd.setCursor(12, 0);
lcd.print(" %");
lcd.setCursor(12, 1);
lcd.print(" C");

pinMode(PinOut1,OUTPUT);
pinMode(PinOut2,OUTPUT);
pinMode(PinOut3,OUTPUT);
}

voidloop()
{
////////////////////TOMA DE DATOS DE LOS SENSORES////////////////////////////////////

  valor_LM35 = (5.0 * analogRead(SensorLM35)*100)/1024; // iteraciones entre
resolución

//float temperature = 25; //replace with a thermometer reading if you have it
floatrelativeHumidity =getHumidity(valor_LM35);

////////////////////Set-Point////////////////////////////////////

//// LM35////

intdatatemp = 0;
  potLM35 = analogRead(PotPinLM35);
  datatemp = map(potLM35, 0, 1023, 10, 50);
  lcd.setCursor(10,1);
  lcd.print(potLM35);

  if (valor_LM35>potLM35)
  digitalWrite(PinOut1, HIGH);
  else
  digitalWrite(PinOut1,LOW);

  /// HIH4030///

intdatahum = 0;
  potHIH = analogRead(potPinHIH4030);
  lcd.setCursor(10,0);
  lcd.print(potHIH);

  if (relativeHumidity>potHIH)
  digitalWrite(PinOut2, HIGH);

```

```

else
digitalWrite(PinOut2,LOW);

//////////Envío de Datos Por Serial//////////

Serial.print(valor_LM35);
Serial.print(",");
Serial.println(relativeHumidity);
delay(200);
}
floatgetHumidity(floatgradosCelsius)
{
//caculate relative humidity
floatVoltaje_Suministrado = 5.0;

// read the value from the sensor:

floatsensorRH = 161.0 * voltage_valor / Voltaje_Suministrado - 25.8;
floattrueRH = sensorRH / (1.0546 - 0.0026 * gradosCelsius);

returntrueRH;
}

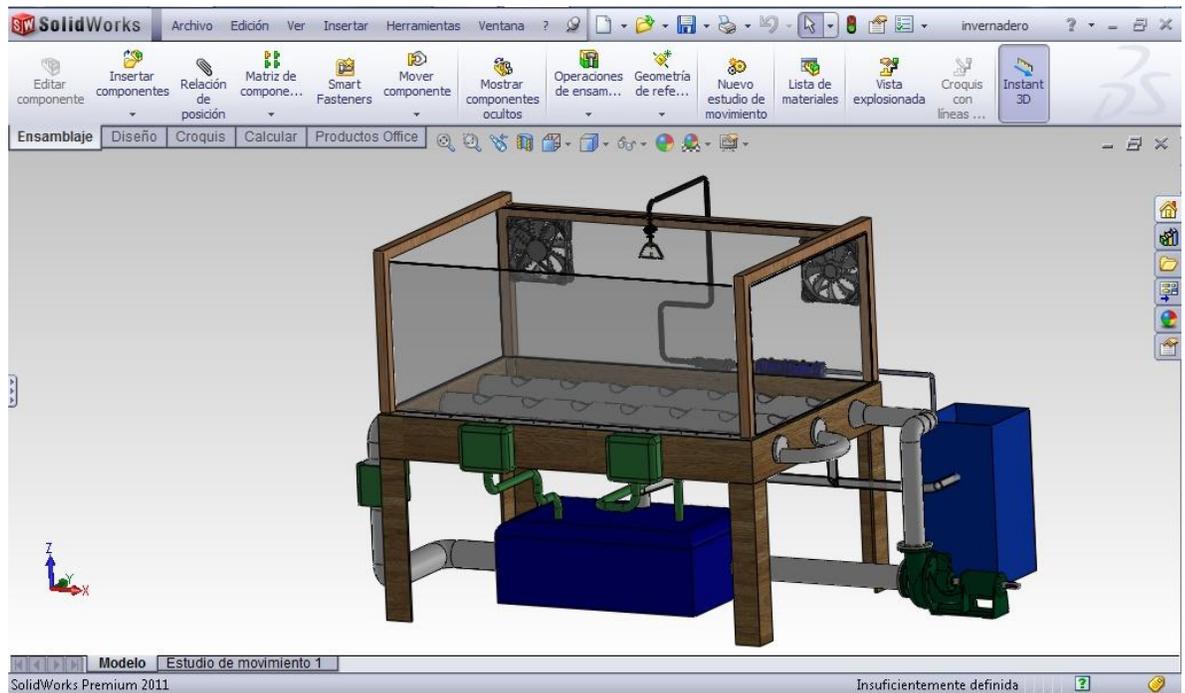
```

11. DISEÑO Y ESTRUCTURA DEL INVERNADERO.

5.1 DISEÑO

A continuación se muestra un diseño básico Creado En SolidWorks de un invernadero hidropónico lo cual se construirá como prototipo básico para la implementación del sistema de control.

Figura 10. Invernadero hidropónico.

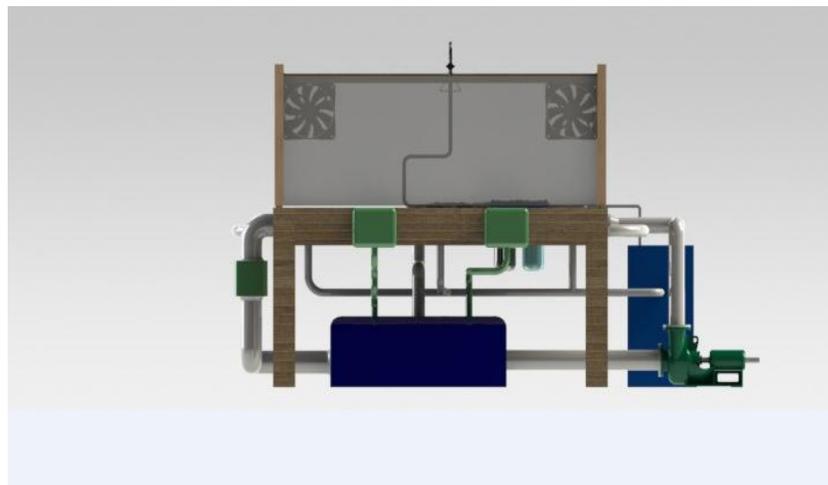


5.2. VISTAS DEL DISEÑO

Se mostrarán 6 vistas principales ya Renderizadas como Vista Frontal, Posterior, superior, inferior y sus laterales: Izquierda y Derecha.

En la figura 11. Se ve de forma detalla una vista frontal del invernadero donde se ven 2 ventiladores para la circulación de aire, en la parte inferior se ve el tanque donde se almacena el agua y en la parte inferior derecha se observa la bomba de circulación de agua.

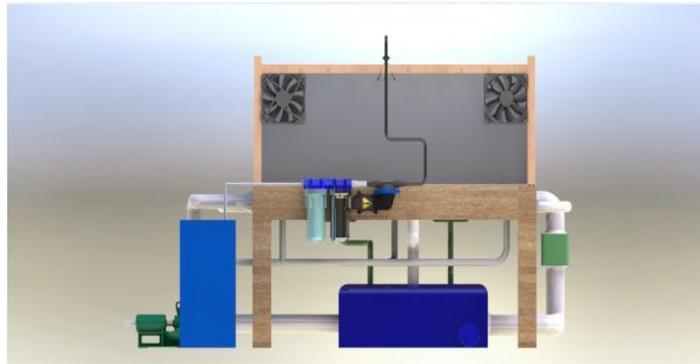
Figura 11. Vista Frontal.



Fuente: Autores

En la figura 12. Se observa de forma detallada una vista trasera una bomba de rocío con su filtro para el riego de las matas, en la parte izquierda inferior se ve un tanque donde se almacena agua para la bomba de rocío.

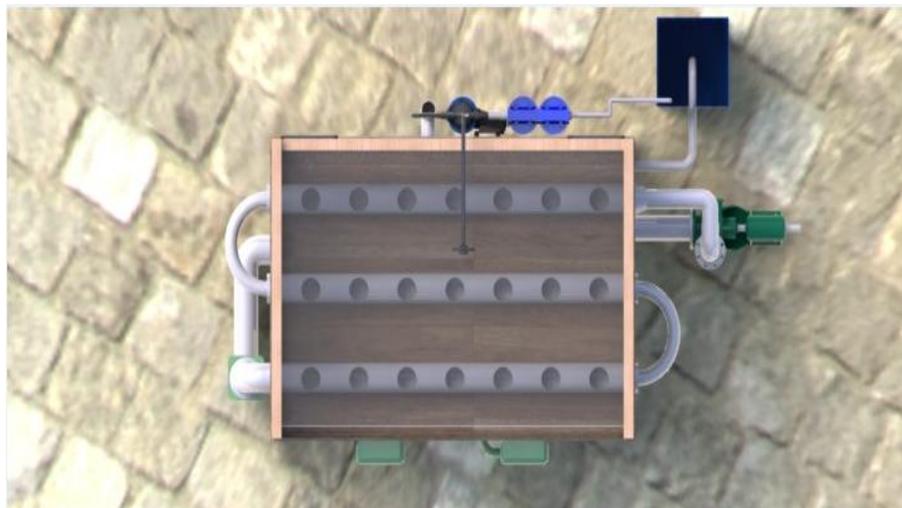
Figura 12. Vista Trasera.



Fuente: Autores

En la figura 13. Se ve de forma detallada una vista superior donde vemos el sistema de tubería donde se introducirán las hortalizas y la circulación de agua con retorno al tanque.

Figure 13. Vista Superior.



Fuentes: Autores