



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

AUTOMATIZACIÓN Y ECONOMIZACIÓN DE UN INVERNADERO

MEMORIA PRESENTADA POR:

José Vidal Torró

GRADO DE INGENIERIA ELECTRICA

Convocatoria de defensa: Septiembre 2016

ÍNDICE

1. INTRODUCCIÓN	Pág.03
2. JUSTIFICACIÓN DEL PROYECTO	Pág.03
3. OBJETIVOS	Pág.04
4. METODOLOGÍA	Pág.05
4.1 Requisitos del sistema y funcionamiento	Pág.05
Control de temperatura	Pág.05
Control de humedad en el suelo	Pág.06
Control de humedad en el aire	Pág.06
4.2 Elección del Hardware	Pág.07
Hardware de control:Arduino	Pág.07
Sensores	Pág.09
Actuadores	Pág.12
4.3 Elección del Software	Pág.12
Software empleado	Pág.12
4.4 Diseño	Pág.13
Estructura del sistema	Pág.14
4.5 Montaje	Pág.17
Sensor de temperatura y humedad relativa	Pág.17
Sensor de humedad en el suelo YL38 y YL69	Pág.18
Módulo de relés	Pág.19
Sensor de luminosidad	Pág.19
Entrada analógica	Pág.20
4.6 Programación	Pág.21
Librería y declaración de variables	Pág.22
Setup	Pág.23
Loop	Pág.24
Recepción de datos	Pág.24
Control de temperatura	Pág.24
Control de humedad	Pág.26
Aplicación Visual Basic	Pág.27
4.7 Pruebas, problemas y mejoras	Pág.28
5. PRESUPUESTO	Pág.30
6. ANEXOS	Pág.31
7. REFERENCIAS BIBLIOGRÁFICAS Y WEBGRAFÍA	Pág.32

1. INTRODUCCIÓN

En el presente trabajo abordaremos desde un punto de vista teórico-práctico el proceso de diseño y automatización de un invernadero cuyos objetivos principales serán: Por una parte, disminuir los costos de implementación del sistema, dado que actualmente el alto coste de dicha automatización, impide su comercialización a gran escala. Por otra parte, mejorar la calidad del trabajo del operador, dando lugar a su vez, a un aumento considerable de seguridad, confort y ahorro energético.

Para ello, se utilizará un micro controlador capaz de realizar todo el proceso de monitorización a través de sensores que medirán parámetros ambientales tales como temperatura, humedad (tanto relativa como en suelo) y parámetros lumínicos.

Con los parámetros obtenidos, actuaremos sobre diferentes salidas que nos proporcionaran el funcionamiento deseado del sistema. Todo esto, será transmitido por medio de comunicación serie para poder visualizar el estado del sistema en el monitor mediante una aplicación.

Además, cabe destacar la utilización de tecnologías de hardware libre, hecho que favorece la reducción de costes de una manera significativa, comparado con las opciones actuales del mercado.

Para concluir, el trabajo constará de un enfoque teórico-práctico, por lo que se diseñará el sistema de forma teórica y, a su vez, se realizará la programación y montaje de un prototipo para comprobar su funcionamiento.

2. JUSTIFICACION DEL PROYECYO

Este proyecto contempla la posibilidad de comercializar el sistema de control de un invernadero dado que en la zona geográfica en la que nos encontramos (Vall de Albaida) existe un escaso número de invernaderos automatizados.

Tras varias entrevistas a agricultores llegamos a la conclusión de que dos son los hechos que impiden la automatización de los invernaderos. Por una parte, el elevado coste que supone su instalación (aún a sabiendas de que es la mejor opción a largo plazo para un cultivo tanto eficaz como eficiente). Por otra parte, la no-confianza con el sistema.

Este proyecto plantea la automatización de un invernadero por las ventajas que ello puede suponer para el usuario. El cultivo bajo invernadero siempre ha permitido obtener producciones de calidad y mayores rendimientos, en cualquier momento del año, a la vez que también permiten alargar el ciclo de cultivo pudiendo producir en las épocas del año más difíciles y obteniendo mejores precios.

Por todo esto, consideramos importante demostrar y hacer saber cómo con un presupuesto reducido y un control absoluto se puede conseguir resultados óptimos sin renunciar a la calidad.

3. OBJETIVOS

- Objetivo general

El objetivo general de este proyecto es desarrollar un sistema autónomo que gestione los elementos presentes en un invernadero para conseguir así una mejora de la producción y de la calidad de los alimentos cultivados gracias a la monitorización de parámetros para ajustarla siempre a las necesidades del propietario.

- Objetivos específicos
 - Dotar al sistema de un conjunto de sensores capaces de realizar las mediciones de los parámetros (temperatura, parámetros lumínicos y humedad relativa y en suelo)
 - Desarrollar un código para el micro controlador capaz de tratar las señales adquiridas tanto analógicas como digitales
 - Diseñar y desarrollar un programa de control, el cual controle los parámetros ambientales en tiempo real.
 - Elaborar un presupuesto que permita implantar el proyecto.

4. METODOLOGIA.

Este apartado está dividido en varios sub apartados que tienen como fin explicar el desarrollo del sistema.

4.1 Requisitos del Sistema y funcionamiento

Para llevar a cabo la implementación del sistema lo principal y necesario es un micro controlador que sea el centro de todo, a partir del cual empezaremos a incluir sensores y actuadores hasta lograr que realice el control deseado.

La finalidad de este proyecto es que el invernadero sea totalmente autónomo. Para ello, a través de un micro controlador (Arduino) conseguiremos una lectura constante de los sensores e idearemos ciclos de trabajos: control de temperatura, control de humedad en el suelo y control de humedad en el aire

▪ Control de Temperatura

Para iniciar el ciclo del control de temperatura, determinaremos un valor para el sistema a través de un potenciómetro externo.

En primer lugar introduciremos un margen de tolerancia definido internamente en el programa. Dicha tolerancia nos otorgará un rango de temperaturas válidas y nos evitara la oscilación de conexión/desconexión de elementos asociados a este control una vez intentemos mantener la temperatura deseada.

Como algo evidente tendremos tres estados con referencia a la temperatura;

Temperatura elevada, temperatura baja o temperatura correcta. Si el valor de temperatura actual es inferior a la fijada, estaremos en una temperatura baja y nuestro sistema encenderá los calentadores hasta que lleguemos a la temperatura correcta. En el caso contrario que tengamos una temperatura elevada, el sistema actuara abriendo ventanas e iniciando la ventilación forzada hasta volver a la deseada.

- **Control de humedad en el suelo**

Con la monitorización de la humedad del suelo y teniendo una referencia obtenida gracias a los potenciómetros externos, actuaremos activando el riego siempre que no tengamos una humedad suficientemente alta.

En la práctica, este sistema puede provocar que el riego se active en momentos no óptimos como por ejemplo, mediodía. Con tal de evitar dicha situación, incorporamos una serie de sensores que nos informan de las condiciones lumínicas del invernadero.

Tomando esto como base, su funcionamiento será su activación siempre que haya una humedad menor a la deseada y la luz incidente sea “baja”, con baja quedan contempladas prácticamente todas las horas del día menos las centradas entre las 12 del mediodía y las 8 de la tarde.

Como medida adicional de error o alarma, si la humedad superara por alguna razón anómala en un 20% en porcentaje prefijado, se enviaría un mensaje de alerta al usuario a través de un elemento externo.

- **Control de Humedad en el aire**

Se obtendrá una lectura de la humedad en el aire que hay en el invernadero, esta es en la mayor de los casos dependiente de la temperatura y al tener un bloque entero solo para el control de esta, la humedad en el aire siempre quedará en unos valores seguros siendo innecesario otro bloque entero para esta.

Como medida adicional se ha optado por monitorizarla y, si en algún momento superara un valor prestablecido en el 85%, se activará una salida de ventilación de manera forzada.

Con todo esto, podemos concluir el siguiente análisis de necesidades:

- Medición de Temperatura
- Medición de humedad relativa en el aire
- Medición de humedad relativa en el suelo
- Encendido y apagado de circuitos 230
- Medición de luz incidente en el invernadero.

4.2 Elección del hardware

▪ Hardware de control: Arduino

Para la realización de este proyecto se podría haber utilizado cualquier tipo hardware de control, bien sea un PLC o algún módulo demótico modificado para trabajar en un invernadero existente en el mercado, pero uno de los objetivos del proyecto es desarrollar el sistema desde cero. Por ello se ha optado por una placa Arduino que es básicamente un micro controlador que cumple con todas las expectativas como más adelante veremos.

Las ventajas de trabajar con un sistema Arduino son:

- Precio: Existen multitud de modelos de placas de varios fabricantes con unos precios muy económicos, y uno de los pilares del proyecto es mantener los costes bajos
- Sistema abierto: Arduino es un sistema Open Souce (Codigo abierto) lo que significa que es desarrollado libremente
- Sistema didáctico: Debido a su configuración hardware, sencillez de lenguaje y a multitud de información existente es posible empezar pequeños proyectos de forma rápida y segura.
- Entradas y salidas digitales disponibles: Podemos elegir modelos en función de las salidas o entradas que necesitamos.
- Muy extendido: Existen infinidad de librerías de libre distribución para poder comunicarse con hardware y software de terceros.

Arduino es una plataforma de hardware libre que se basa en un micro controlador Atmel AVR y un entorno de desarrollo muy sencillo. Existen diferentes modelos como muestra la siguiente tabla, cabe destacar el elevado número de entradas y salidas que poseen incluso los modelos de gama más baja.

	Processor	Processor Voltage	Supply Voltage	Flash	SRAM	Digital I/O Pins	PWM Pins	Analog Inputs	Hardware Serial Ports	Dimensions	Shield Compatibility	Notes and Special Features
Uno	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Excellent (most will work)	
Uno Ethernet	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	2.1"x2.7" 53x75mm	Very Good (some pin conflicts)	Has Ethernet Port. Requires FTDI cable to program.
Mega	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	
Mega ADK	16MHz Atmega 2560	5v	7-12v	256Kb	8Kb	54	14	16	4	2.1"x4" 53x102mm	Good (some pinout differences)	Works with Android Development Kit.
Leonardo	16MHz Atmega 32U4	5v	7-12v	32Kb	2.5Kb	20*	7	12*	1	2.1"x2.7" 53x75mm	Fair (many Pinout Differences)	Native USB capabilities. USB Micro B programming port.
Due	84MHz ARM SAM3X8E	3.3v	7-12v	512Kb	96Kb	54	12	12	4	2.1"x4" 53x102mm	Poor (voltage and pinout differences)	Fastest processor. Most memory. 2-channel DAC. USB micro B programming port. Native micro AB port.
Micro	16MHz Atmega 32U4	5v	5v	32Kb	2.5Kb	20*	7	12*	1	0.7"x1.9" 18x49mm	N/A	Smallest board size. Native USB capabilities
Flora	8MHz Atmega 32U4	3.3v	3.5-16v	32Kb	2.5Kb	8*	4	4*	1	1.75" dia 44.5mm dia	N/A	Sewable Pads. Fabric-friendly design. Native USB Capabilities
DC Boarduino	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. Requires FTDI cable for programming
USB Boarduino	16MHz Atmega 328	5v (USB)	5v (USB)	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	N/A	Can build without headers or sockets for smaller size. USB Mini B programming port.
Menta	16MHz Atmega 328	5v	7-12v	32Kb	2Kb	14	6	6	1	0.8"x3" 20.5x76mm	Excellent (most will work)	Mint-Tin Size and Prototyping Area. Requires FTDI cable for programming.

Tabla (1): Características de las placas Arduino.

A continuación se resumen las principales características del hardware empleado en el proyecto.

Arduino Uno

La placa Arduino Uno probablemente sea la más conocida por ser la elegida para la iniciación del aprendizaje en este dispositivo. En nuestro caso utilizaremos la R3.

Cuenta con puerto de comunicaciones UART (Serial Hardware), USB(puerto virtual), comunicación mediante I2C (TWI) y SPI.

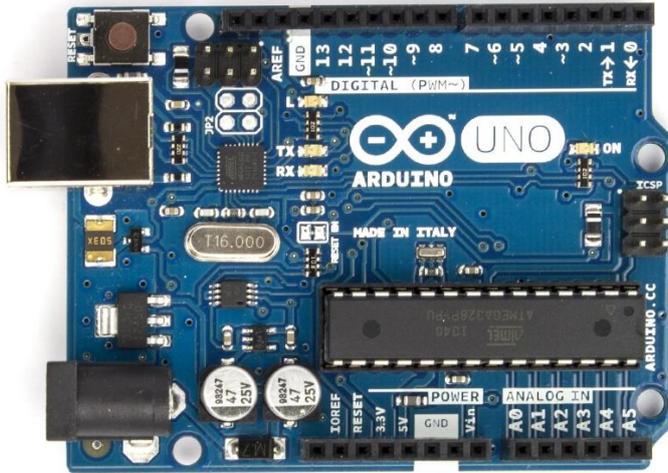


Figura (1): Vista frontal de Arduino UNO

Sus características son:

Microcontrolador	ATmega328P
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (limite)	6-20V
Pines digitales E/S	14(6 con salida PWM)
Pines de entrada analógica	6
Consumo por pin E/S	20mA
Consumo del pin 3.3	50mA
Memoria flash	32kb (0.5kb empleados por el bootloader)
SRAM	2kb
EEPROM	1kb
Frecuencia de reloj	16 Mhz
Longitud	68.6mm
Ancho	53.4mm
Peso	25g

Tabla (2): Características de Arduino UNO

- **Sensores**

Los sensores nos permite recibir información del entorno, para su elección se ha tenido en cuenta tanto el precio como la fiabilidad. Todos cuentan con características suficientes para llevar a cabo el proyecto.

Sensor de temperatura y humedad DHT11

El sensor DHT11 presenta un encapsulado de plástico, utiliza un elemento resistivo para la medición de la humedad y un termistor (NTC) para la medición de temperatura.

Devuelve una señal digital en la medición y permite una lectura cada segundo (No es rápido, pero para la aplicación destinada es más que suficiente).

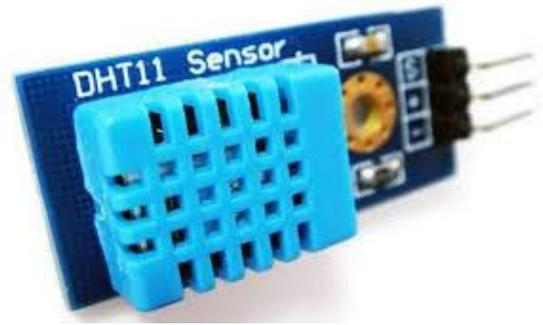


Figura (2): Sensor DHT11

Características:

- Rango de humedad: de 20% al 80% con 5% de precisión
- Rango de temperatura: de 0° a 50° con 5% de precisión (pero solo mide por grados, no fracciones)
- Voltaje de entrada : 3,3 - 5VDC

Sensor de humedad del suelo YL38 y YL69

Este sensor tiene la capacidad de medir la humedad del suelo. Aplicando una pequeña tensión entre los terminales del módulo YL-69 hace pasar una corriente que depende básicamente de la resistencia que se genera en el suelo.

El módulo YL-38 contiene un circuito comparador LM393 SMD (de soldado superficial) muy estable, un led de encendido y otro de activación de salida digital

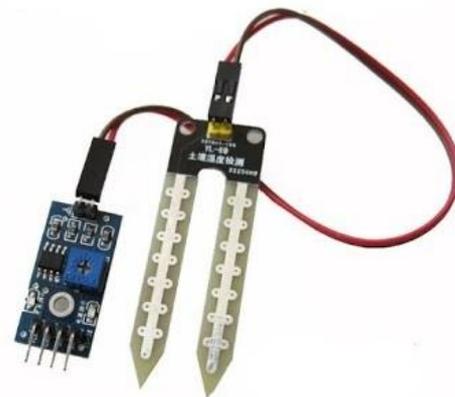


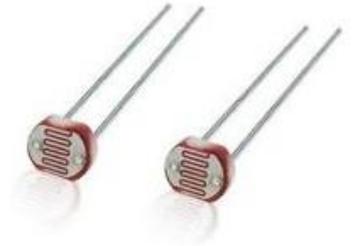
Figura (3): Sensor YL38 y YL69

Características:

- Voltaje de entrada: 3,3 - 5VDC
- Voltaje de salida: 0 – 4,2V
- Corriente: 35mA

Fotorresistencia GL-5549

Su funcionamiento consistente en una célula de Sulfuro de Cadmio, altamente estable, encapsulada con una resina epoxi transparente, resistente a la humedad. La respuesta espectral es similar a la del ojo humano. Su nivel de resistencia aumenta cuando el nivel de luz disminuye.



Características:

Figura (4):Fotorresistencias(LDR)

- Voltaje máximo: 150V
- Temperatura (°C): -30° + 70°
- Pico espectral (nm): 540
- Resistencia oscuridad (K Ω): 100-200
- Resistencia Luz Brillante(K Ω): 10000
- Gamma : 0.9
- Tiempo de respuesta(ms): 25

Potenciómetros lineales 10k

Es una resistencia variable (con cursor y tres terminales). Limita el paso de la corriente eléctrica y se puede variar gracias al cursor. Utilizaremos estos potenciómetros para determinar los valores deseados para el sistema, como la temperatura y la humedad en la tierra.



Características:

Figura(5): Potenciómetro

- Angulo de rotación 300°
- Resistencia 10k Ω
- Tolerancia 20%
- Voltaje máximo 150V AC
- Consumo 0.08W

▪ Actuadores

Los actuadores reciben sus órdenes directamente desde la salida del micro controlador. Existen multitud de actuadores que podemos controlar con Arduino como por ejemplo los Servo. Para este proyecto hemos usado un módulo de relés que permitan accionar circuitos a 220v

Modulo Relés

Utilizamos una placa de 8 relés para el accionamiento de circuitos 230vac. Las entradas son activas en modo LOW, un detalle a tener en cuenta en la programación del micro controlador.

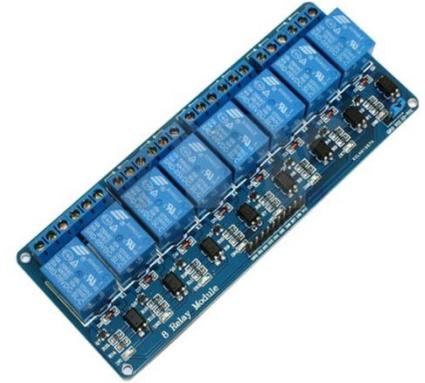


Figura (6): Modulo Relés

Características:

- Voltaje de entrada: +5V
- Voltaje de accionamiento: +5V
- Led de estado en cada canal
- Consumo por canal: 80mA
- Relés: AC250V 10 A; DC30V 10^a
- Dimensiones: 138 x 54 mm

4.3 Elección del Software

▪ Software empleado

Para la realización de este proyecto ha sido necesario la utilización de:

- IDE Arduino

Toda la programación del arduino ha sido realizada a través del software gratuito IDE que puede encontrarse en la página web oficial para descargar. Es posible programar el micro controlador mediante otros programas o incluso con el editor de textos pero el software propio de arduino cuenta con diferentes herramientas que agilizan y simplifican este proceso.

Podemos tener acceso rápido a las librerías, depuración de errores, selección de puerto y modelo a programar. Una de las herramientas más utilizadas es el Monitor Serial que nos permite comunicarnos mediante el USB de nuestro PC y ver los mensajes de ejecución que hayamos programado.

El micro controlador de la placa se programa mediante el lenguaje de programación de arduino y el entorno de desarrollo arduino. Acepta las estructuras de programación de C y casi todas las de C++.

El entorno de programación se puede extender con el uso de librerías, el IDE trae por defecto unas cuantas instaladas pero para nuestro proyecto hemos necesitado descargar e instalar las necesarias para nuestro hardware.

– Microsoft Visual Basic 2010 Express

Visual Basic 2010 Express es parte de la familia Visual Studio 2010 Express, un conjunto gratuito de utilidades que los desarrolladores Windows de cualquier nivel pueden usar para crear aplicaciones personalizadas usando ajustes básicos y expertos.

Es un software simple que te permite desarrollar y construir aplicaciones para la Web, tu smart phone, escritorio o la nube. Para la programación en esta plataforma se asignan acciones a los elementos disponibles en el programa, como puedan ser botones, ventanas de texto o gráficas.

4.4 Diseño

Dado que el proyecto puede ser ampliable, la estructura general del sistema se ha realizado de un modo abierto para poder realizar posibles modificaciones de manera externa a través de otros dispositivos.

Se contempló la posibilidad de añadir un módulo Ethernet shield para no tener que estar físicamente conectados con el nodo central pero se llegó a la conclusión que no era necesario dadas las especificaciones de nuestro proyecto y al no tener la necesidad real de estar conectados inalámbricamente debido a las cortas distancias y el gran aumento de costes que supondría.

En cualquier caso, a modo de ampliación y mejora como más tarde se detallara queda la posibilidad de implementarlo.

Para los datos obtenidos del invernadero como la temperatura y la humedad, hemos decidido, por la simplificación que esto nos proporciona y por la imposibilidad de simularlo de otra forma, reducir el número de sensores para la adquisición de los parámetros a uno.

Es evidente que en un entorno real de un invernadero sería poco preciso usar solamente un sensor de temperatura o humedad para determinar las lecturas de todo el lugar. En caso de una implementación real se distribuirían los sensores sobre la superficie a monitorizar y se calcularía una media para obtener un parámetro que representara todo el invernadero, o incluso si fuera lo suficientemente grande se obtendrían parámetros diferentes de varias zonas y se actuaría más centradamente en ellas para acondicionarlo a los parámetros deseados.

De igual forma, a modo de ampliación y mejora queda la posibilidad de implementación.

Estructura del sistema

Como se comentó con anterioridad el proyecto se compone de dos partes, Hardware y Software.

La parte Hardware será la formada por el nodo principal que incluirá el micro controlador Arduino, los sensores y los relés que actuaran sobre las salidas. Todo ello estará conectado via USB con una plataforma capaz de ejecutar nuestra aplicación software desarrollada en Visual Basic

A continuación se puede ver un esquema de lo dicho anteriormente

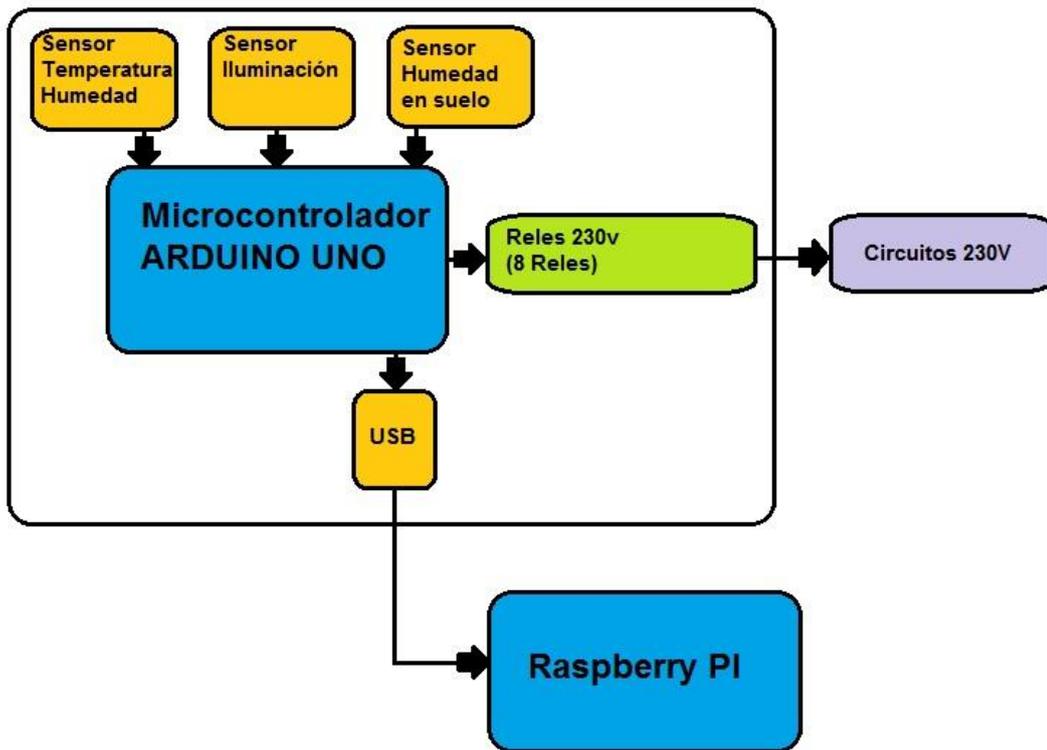


Figura (7): Estructura del Sistema

Nodo Principal.

Formado por la placa Arduino UNO y los sensores DHT11, YL38 e YL69. Este conjunto será el encargado de recoger los datos del sistema, procesarlos y actuar en consecuencia

Se instalará en el exterior del invernadero en una caja estanca preparada donde solo tendremos accesible los potenciómetros para el ajuste de la Temperatura y la Humedad

También controlará y monitorizará:

- Encendido y apagado de hasta 8 circuitos 220V
- Lectura de valores de luminosidad
- Lectura de Temperatura y Humedad relativa (DHT11)
- Lectura de Humedad en el suelo (YL38 – YL69)
- Recepción de consigna de Temperatura y Humedad objetivo
- Envío de datos a través del puerto serie

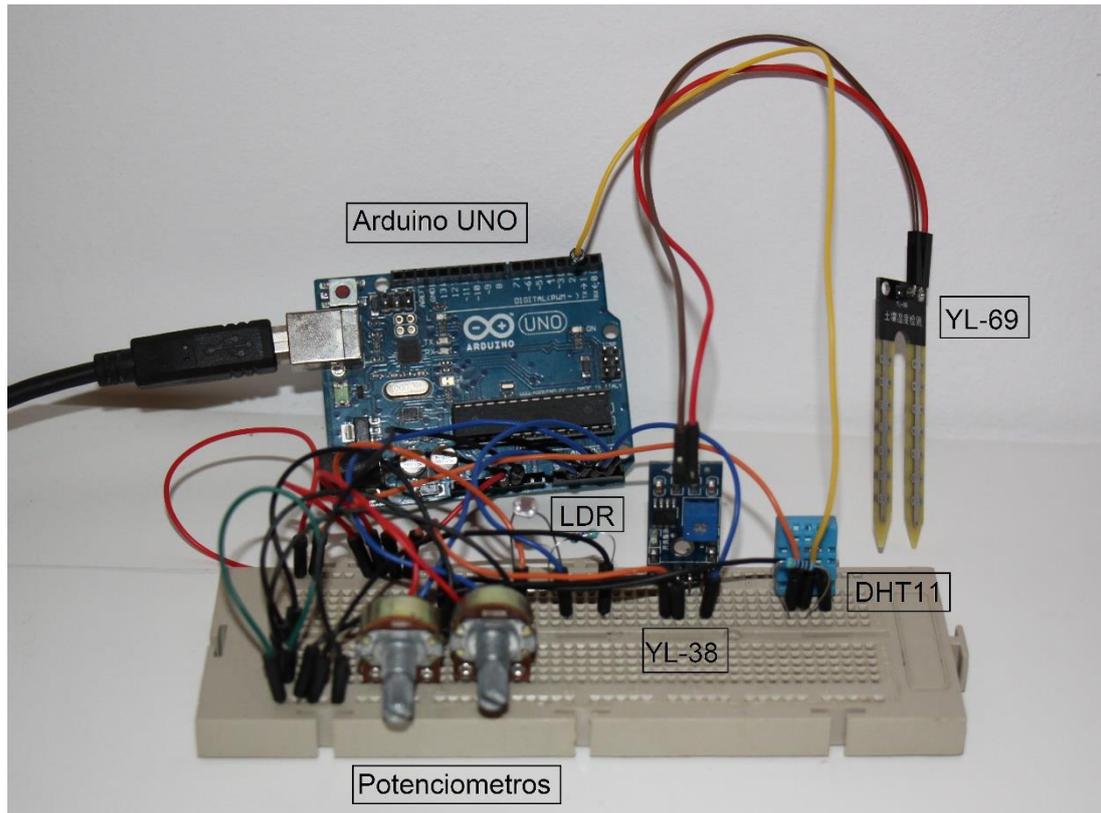


Figura (8): Montaje del nodo principal

Una vez montado se puede determinar la alimentación de la placa, se realizara mediante una fuente de alimentación que suministre el voltaje en continua y que cumpla con las especificaciones de esta, o puede estar alimentada directamente desde el puerto USB que la conectara con el PC

De cualquier forma ya que los elementos utilizados en el Nodo principal son pocos y no requieren de un alto consumo, se va a optar por alimentarlos a 5v a través de los pines que nos proporciona el propio Arduino.

Interfaz de control

Sera ejecutada desde el terminal, bien sea un PC o una Raspberry PI , en esta podremos ver un informe de estado del invernadero, los sensores y las salidas controladas por el Arduino. Contará con un botón de inicio para inicializarla, en ese momento la aplicación buscara los puertos COM disponibles en la unidad desde donde este siendo ejecutada y deberemos seleccionar el que esté conectado con el Arduino.

Tras ello con el botón Conexión iniciaremos la comunicación y empezaremos a ver en la pantalla los datos recibidos.

4.5 Montaje

En este apartado podremos observar cómo se han montado y conectado todos los dispositivos del sistema, como previamente se ha dicho este no será un montaje real pero se ha procedido a comprobar su correcto funcionamiento a una pequeña escala.

Cada sensor y actuador cuenta con unas características que deben tenerse en cuenta a la hora de cablearse y en el conexionado. Adjuntos en el Anexo se encuentran los datasheet de cada uno de los elementos aquí citados.

Sensor de Temperatura y Humedad Relativa (DHT11)

Se ha montado en la placa el sensor combinado de Temperatura y Humedad, concretamente el modelo DHT11. El pin de entrada para la lectura de datos es el D2, se conectan a una entrada digital pues estos sensores se comunican con un bus 1-wire.

Para la correcta lectura de los valores proporcionados se ha utilizado la librería específica del sensor.

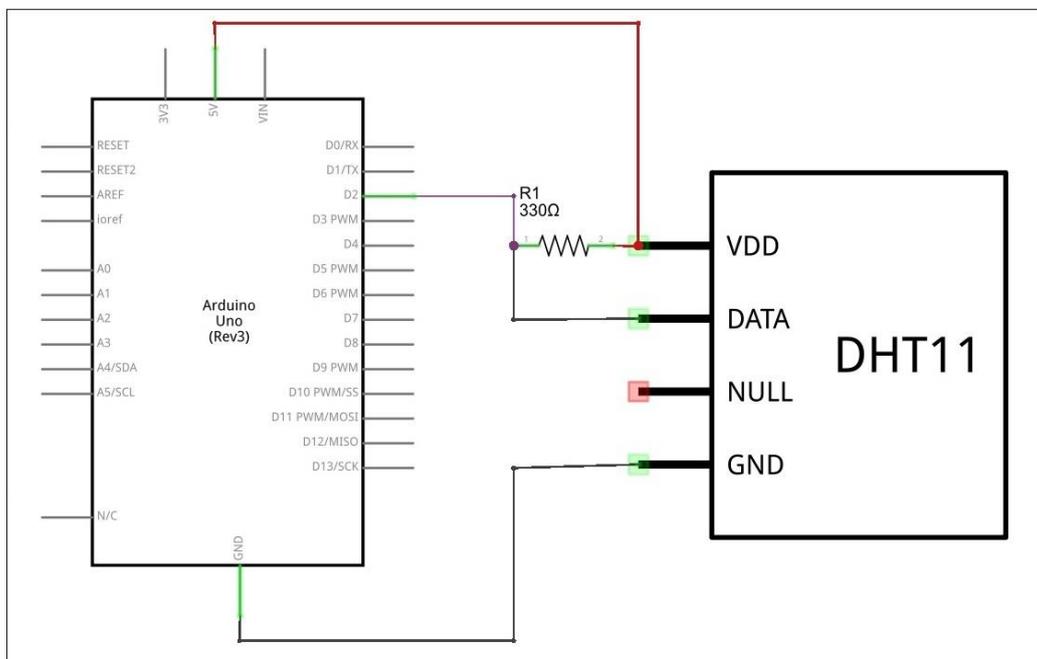


Figura (9): Conexión del sensor DHT11

Sensor Humedad en el suelo YL38 y YL69

Su función es la medición de la humedad en el suelo, para ello cuenta con un módulo principal, el YL-38, que contiene soldado en pequeña placa con un circuito comparador LM393 SMD, un led de encendido y otro de activación de salida digital. También cuenta con 2 pines para la conexión de la sonda YL-69, la cual, al hacer pasar una corriente que dependerá de la resistencia generada por el suelo nos da la capacidad de medir la humedad.

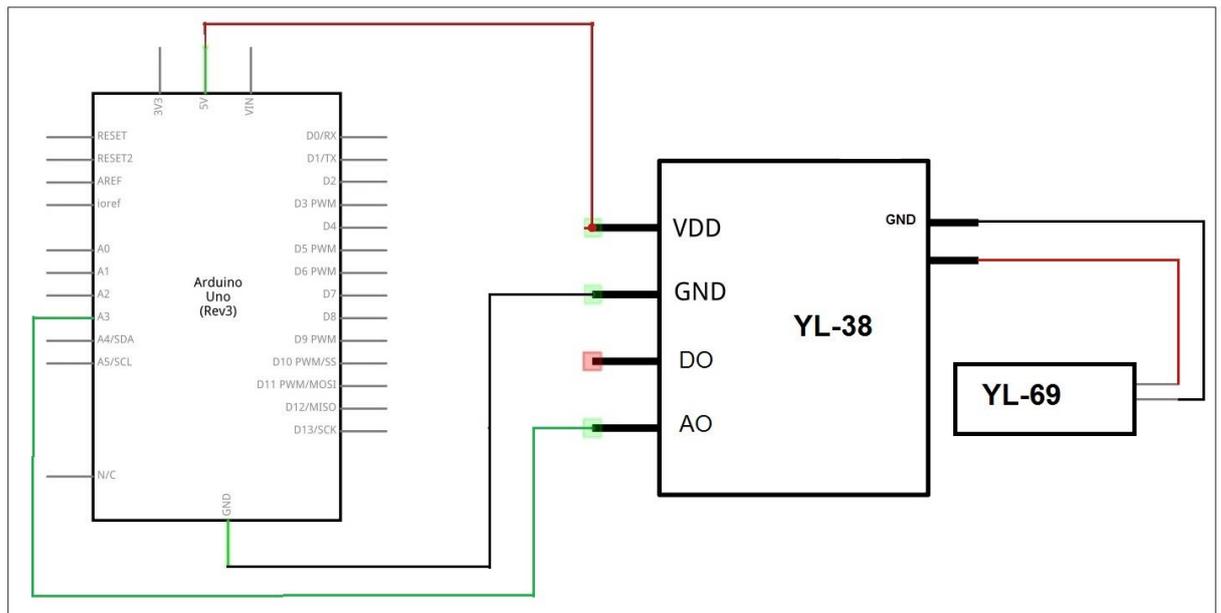


Figura (10): Conexión del sensor YL-38 y YL-69

Módulo de relés

Su función es el accionamiento de los circuitos de 230V. El modulo será accionado por salidas digitales.

La placa cuenta con opto-acopladores que aseguran una separación galvánica entre la parte de control que funciona a 5 V y la parte de potencia a 230V. En la siguiente imagen podemos ver la conexión eléctrica con la placa Arduino.

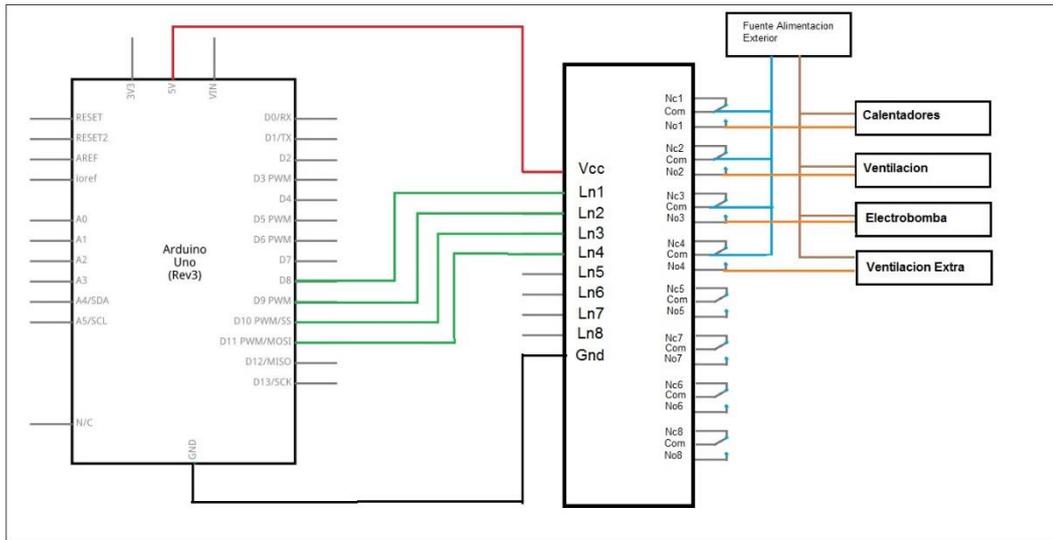


Figura (11): Conexión del módulo 8 Relés y Circuitos 230V

Sensor de luminosidad

Para las lecturas de luminosidad hemos acoplado fotorresistencias con su correspondiente divisor de tensión y llevado las lecturas a la entrada analógica A4.

Esta medición puede ser un poco imprecisa pero para el uso general que vamos a darle es más que suficiente. Si quisiéramos disponer de un control más exacto optaríamos por instalar luxómetros pero dispararía el presupuesto.

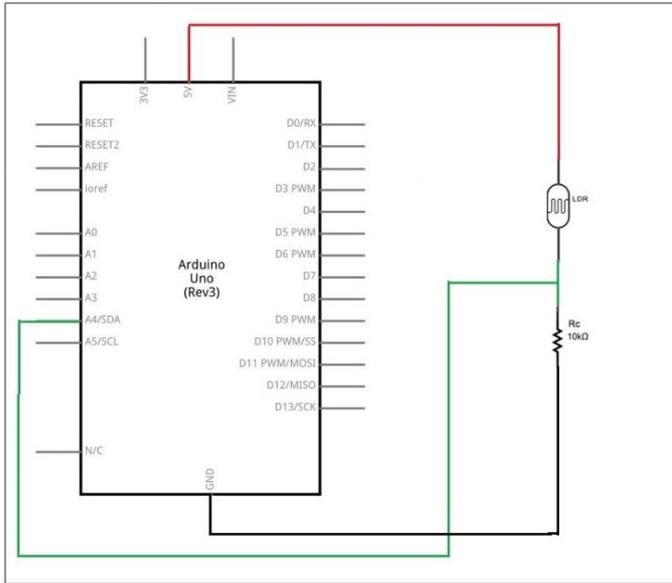


Figura (12): Conexión de LDR

Entradas Analógicas

Se utilizarán las entradas analógicas A1 y A2 para los valores objetivo de temperatura y humedad en el suelo. Para ello conectamos dos potenciómetros tal como se representa en la imagen.

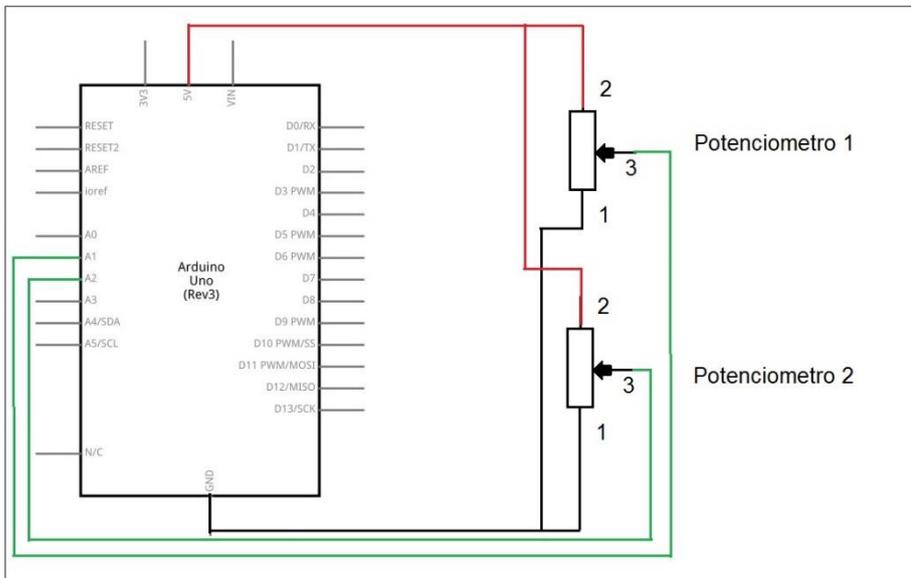


Figura (13): Conexión de potenciómetros a Entradas analógicas

Tabla resumen entradas / Salidas

Sensor/ Relé	Unidad de Control	Tipo de entrada/salida	Nº pin en Arduino	Función asociada
Relé 1	Arduino UNO	Salida Digital	D8	Calentadores
Relé 2	Arduino UNO	Salida Digital	D9	Ventilación
Relé 3	Arduino UNO	Salida Digital	D10	Electrobomba
Relé 4	Arduino UNO	Salida Digital	D11	Ventilación Extra
DHT11 (Tem./Hum)	Arduino UNO	Entrada Digital	D2	Sensor Temperatura Humedad
YL38-Y169	Arduino UNO	Entrada Analógica	A3	Sensor Humedad suelo
Fotorresistencias	Arduino UNO	Entrada Analógica	A4	Medición Luminosidad
Potenciómetro 1	Arduino UNO	Entrada Analógica	A1	Selector de Temperatura
Potenciómetro 2	Arduino UNO	Entrada Analógica	A2	Selector de Humedad

Tabla (3): Resumen entradas /Salidas del sistema

4.6 Programación

La programación en cuanto a Arduino se ha realizado dividiéndola en pequeños bloques para un mejor entendimiento y depuración.

En el código general del programa podemos ver partes comunes a cualquier programa en Arduino:

- Comentarios. No afectan al código ni se cargan en el micro controlador, por lo que no ocupan memoria, pero son recomendables para poder entender lo que se ha programado y añadir información útil como dónde se deberán conectar ciertos dispositivos.
- Llamada a librerías y declaración de variables.
- Setup, que se ejecuta una sola vez y es donde podemos inicializar variables, arrancar dispositivos y cualquier otra tarea que suponga una única configuración.
- Loop. En esta parte, el bucle, es donde se realiza la mayor parte de la programación pues es lo que realmente ejecutará el micro controlador, una y otra vez.

Aunque el loop es la parte donde se ejecutan las órdenes que definimos para el funcionamiento del programa es posible hacer uso de funciones. Esta se crea fuera del bucle loop pero al ser llamadas dentro de este se ejecutan de manera ininterrumpida.

Dado que el código para Arduino no es excesivamente extenso se procederá a mostrar y explicar con brevedad el funcionamiento de las partes más relevantes. En los ficheros Anexos se adjuntará el código completo con comentarios.

Librerías y declaración de variables

En esta sección definimos las variables que van a ser utilizadas durante la ejecución del programa y les asignamos una forma y nombre. También se incluyen librerías necesarias para el correcto funcionamiento del programa.

En nuestro programa solamente incluimos la librería del sensor combinado de Temperatura y Humedad como vemos en la primera línea, el fabricante nos recomienda usarla para no alterar el proceso de recepción de datos aunque siempre podríamos diseñar una librería nosotros mismos.

Por otra parte se definen como enteras todas las variables que utilizamos en el código con nombres representativos para una mejor comprensión.

```
#include <DHT11.h>
int DHT = 2;
int rele1 = 8;
int rele2 = 9;
int rele3 = 10;
int rele4 = 11;
DHT11 dht11(DHT);
int PTemperatura;
int Phumedad;

int Shumtierra;
int Stemp;
int Shum;
int Sluz;
```

Figura (14): Declaración de variables y librerías

Setup

La parte del setup se ejecuta una sola vez y es donde aprovechamos para definir entre otros el funcionamiento de los pines de la placa Arduino, ya que algunas pueden utilizarse tanto como entradas como salidas es necesario definir las.

El setup también es el lugar donde incluir la velocidad de transmisión para la comunicación serie para que quede definida al iniciar el programa

```
void setup()
{
  pinMode (rele2,OUTPUT);
  pinMode (rele1,OUTPUT);
  pinMode (rele3,OUTPUT);
  pinMode (rele4,OUTPUT);
  Serial.begin(9600);
  pinMode (A1, INPUT);
  pinMode (A2, INPUT);
  pinMode (A3, INPUT);
  pinMode (A4, INPUT);
}
```

Figura (15): Void Setup del código principal

Loop

En loop es la parte principal y la que continuamente se está ejecutando mientras el microcontrolador está en marcha. Se ha dividido esta parte en pequeños sub apartados para que sea más fácilmente entendible.

Recepción de datos

En el momento del inicio del bucle loop recogemos los datos introducidos al micro controlador y los escalamos utilizando la función map para adaptarlos a una escala en la que nos sea cómoda para trabajar con ellos . Las entradas digitales de la placa tienen un rango de entre 0 y 1023 valores, la función map los escala entre los valores que nosotros definimos.

```
int SensorValue1 = analogRead(A1);
int SensorValue2 = analogRead(A2);
PTemperatura = map(SensorValue1, 0, 1023, 10, 60);
Phumedad = map(SensorValue2, 0, 1023, 0, 100);

int SensorValue4 = analogRead(A3);
Shumtierra = map(SensorValue4, 0, 1024, 0, 100);

int SensorValue5 = analogRead(A4);
Sluz = map(SensorValue5, 0, 1023, 1, 120);
Sluz = Sluz * 100;
```

Figura (16): Recepción de datos del programa principal

La parte de recepción de los valores para el DHT 11 es algo más complicada y menos visual ya que trabaja con la librería propia. Esencialmente no devuelve los valores de la temperatura y la humedad relativa siempre y cuando no se detecte una señal de error por parte del sensor, para esto utilizamos un bucle IF.

Control de Temperatura

Para esta parte definimos una variable llamada tolerancia que estará preestablecida en 2 grados. Esto se hace ya que la temperatura al ser un valor que varía fácilmente en pequeños márgenes utilizar un valor fijo podría producir una conexión/desconexión continua si nuestra temperatura actual y la deseada están igualadas.

Con este valor tolerancia abrimos una pequeña ventana de valores validos sobre nuestra temperatura deseada sumando 2 grados por encima y restando 2 por debajo. Esta ventana puede reducirse o ampliarse en función de la estabilidad y precisión que deseemos.

En cuanto al funcionamiento del código, se han incluido bucles if que limitan las condiciones del sistema y actúan en consecuencia activados o desconectando salidas y dándonos información al puerto serie.

```
int Tdesmenos;
Tdesmas = PTemperatura + toltemp;
Tdesmenos = PTemperatura - toltemp;

if (Stemp <= Tdesmenos || Stemp >= Tdesmas)
{

    if (Stemp <= Tdesmenos)
    {
        digitalWrite(rele1,HIGH);
        digitalWrite(rele2,LOW);
        Serial.println();
        Serial.print("La temperatura es inferior a la deseada");
        Serial.println();
        Serial.print("Se ha activado el Calentador");
        Serial.println();

    }
    else if (Stemp >= Tdesmas)
    {
        digitalWrite(rele2,HIGH);
        digitalWrite(rele1,LOW);
        Serial.print("La temperatura es superior a la deseada");
        Serial.println();
        Serial.print("Se ha activado el Extractor ");
        Serial.println();

    }
}
else if (Stemp >= Tdesmenos || Stemp <= Tdesmas)
{

    Serial.print("Temperatura dentro de los limites establecidos");
    Serial.println();
    digitalWrite(rele1,LOW);
    digitalWrite(rele2,LOW);
    Serial.println();

}

}
```

¶

Figura (17): Control de temperatura del programa principal

Control de Humedad

En esta sección del programa controlamos la humedad del suelo, para esto tendremos en cuenta las condiciones de iluminación. En la sección de recepción de datos hemos escalado las lecturas de las fotorresistencias en valores entre 0 y 12000, esto es para tener una escala más precisa sobre la luz que incide en el invernadero donde 0 es oscuridad absoluta y 12000 totalmente iluminado. Se ha calculado que las horas de más sol a mediodía el valor de las fotorresistencias oscila entre 6000 y 7000, en la mañana y en la noche los valores fluctúan entre 3000 y 4000 así que hemos tomado como valor de referencia 3200 contando que cuando haya esa luminosidad debe ser por la mañana, por la tarde/noche.

Esta solución no es precisa y cuenta con muchísimo margen de mejora para afinarlo pero con esto solo se quería introducir una condición en la programación. En una situación real esta parte del código debería ser cambiada por completo.

Se ha programado una situación extraordinaria donde por alguna circunstancia la humedad sea excesivamente alta, se cuenta con un 20% más del valor deseado. Si esto ocurre se enviara una mensaje de alerta al monitor y se detendrá el riego si es que este estaba activo

```
if (Shumtierra <= Phumedad && Sluz <= 3200) /
{
  Serial.println();
  digitalWrite(rele3,HIGH);
  Serial.print("La humedad es menor que la deseada");
  Serial.println();
  Serial.print("El riego esta Activo");
  Serial.println();
}
else
  if(Shumtierra >= Hdesmas)
  {
    Serial.print("La humedad es muy elevada");
    Serial.println();
    digitalWrite(rele3,LOW);
  }
  else{
    Serial.print("Las condiciones ambientales no son optimas para el riego");
    Serial.println();
    digitalWrite(rele3,LOW);
    Serial.println();
  }
}
```

Figura (18): Control de humedad del programa principal

Por otra parte, se monitoriza en todo momento la humedad en el aire y si en algún momento sobrepasa un límite preestablecido en el 80% se activan ventilaciones extra para extraer la humedad restante.

Aplicación Visual Basic

En lo correspondiente a la aplicación de visual basic se ha realizado una programación para poder leer los datos obtenidos desde el puerto serie que queramos, se realizara una secuencia de arranque muy intuitiva para conectarnos con el Arduino y luego visualizaremos en la pantalla los datos que este nos envía.

Aquí se puede ver una captura de la aplicación en pleno funcionamiento bajo unas circunstancias anómalas.

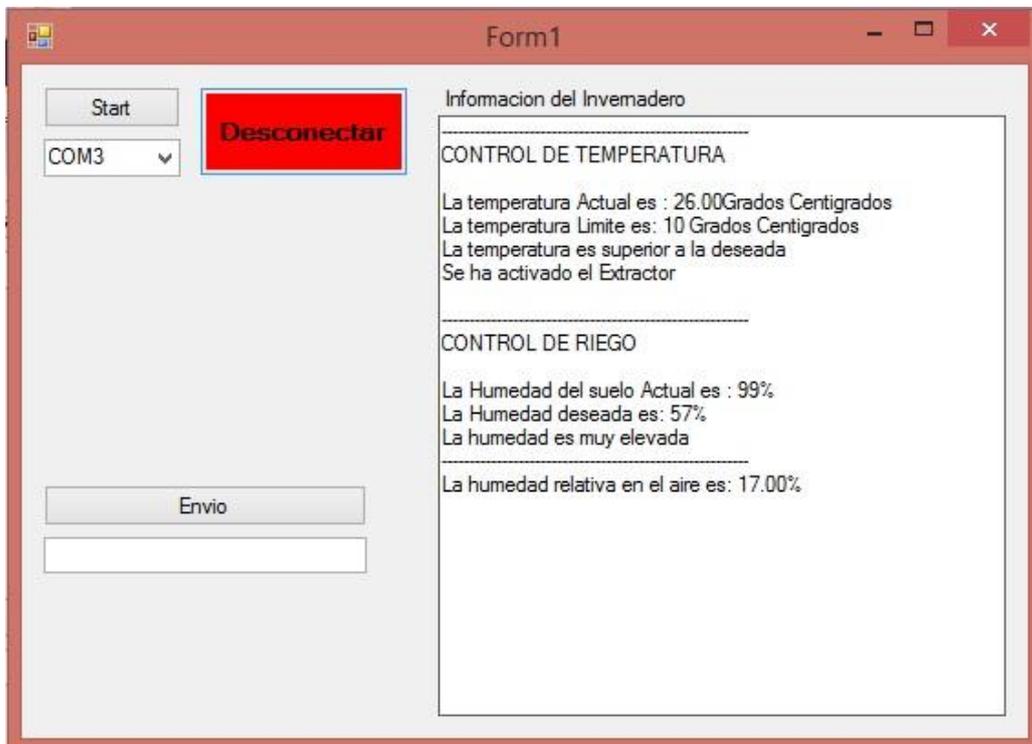


Figura (19): Vista de la aplicación VB en funcionamiento

Lo que se ha tenido en cuenta en esta aplicación a la hora de programar es tener una recepción de datos en el buffer a través de un timer que cada 2segundos nos actualice la información recibida. Se ha implementado un campo para el envío de datos hacia el Arduino que ahora mismo no tiene ninguna función pero que en futuras versiones nos permitirá mandar instrucciones a este.

A continuación un fragmento del código más impórtate de la aplicación, la parte del timer. El resto del código se puede encontrar en la carpeta de anexos.

```
Private Sub tmrtimer_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrtimer.Tick
    StrBufferIn = sppuertos.ReadExisting
    If StrBufferIn <> "" Then
        txtbufferin.Text = StrBufferIn
        StrBufferIn = ""
        sppuertos.DiscardInBuffer()
    End If
End Sub
```

Figura (20): Código del timer en Visual Basic

Consiste en que cada ciclo de reloj si hay datos en el buffer los recoge y los muestra en la pantalla, tras ello vacía el buffer y se prepara para el próximo ciclo.

4.7 Pruebas, problemas y mejoras.

En esta sección veremos los diversos pasos que han hecho falta para que nuestro sistema funcionara correctamente

▪ Pruebas

Para la realización del proyecto se fueron añadiendo poco a poco los elementos al programa principal probando previamente cada uno de ellos de forma individual.

Para esto se procedía con los siguientes pasos:

- Montaje del dispositivo a la placa Arduino
- Documentación sobre el mismo para entender su funcionamiento
- Creación de un código exclusivo para realizar una prueba
- Pruebas de funcionamiento analizando su comportamiento a través del serial monitor y unos mensajes estratégicamente ubicados para obtener información de las acciones del dispositivo
- Modificación del programa individual si se detectan errores de funcionamiento y vuelta al punto 4 hasta conseguir los resultados deseados
- Una vez se comporta de la manera deseada se incorpora el código individual al programa principal

- Pruebas de funcionamiento en el programa principal revisando que no hay solapamiento de variables ni nada extraño
- Revisión visual del nuevo elemento añadido al sistema y comprobación en el Serial Monitor de su correcto funcionamiento
- Se deja el programa en funcionamiento durante cierto tiempo por si aparecen fallos

En general tras seguir estos pasos la incorporación de elementos al programa principal ha resultado sencilla.

▪ **Problemas y soluciones**

A lo largo de la realización de este trabajo se han ido encontrando ciertos problemas de funcionamiento que en un principio deberían funcionar sin problemas.

Algunos de los bloques programados tenían un comportamiento demasiado tosco en las primeras pruebas debido a la complejidad del código. Tras la consulta a gente con más experiencia siempre se llegaba a la conclusión que, para el funcionamiento requerido por el sistema no era necesario complicar tanto el código, así que a través de la simplificación se han eliminado muchos errores.

▪ **Mejoras**

Durante la creación de este proyecto se han investigado diferentes dispositivos y aplicaciones que aunque no han sido incluidas en la memoria podría resultar útiles en vistas a una ampliación o mejora del proyecto, unas de esas mejoras son:

-La utilización de dispositivos Xbee para la transmisión de datos a la placa Arduino o, si habláramos de un proyecto más grande que fueran necesarias dos o más placas, la comunicación entre estas. La incorporación de estos dispositivos añadiría costes al presupuesto pero sería una alternativa interesante si no se puede realizar una instalación de cableado completa.

-Utilizar la Raspberry PI para crear una base de datos donde alojar toda la información del sistema. Tras esto a través de una aplicación web acceder a estos para realizar graficas o un seguimiento detallado.

-Montar la placa Arduino y la RaspberryPI conjuntamente para que, la placa Arduino se encargue solamente de la recogida de datos y sea la raspberry PI la encargada de la comunicación con el exterior

-Desarrollar la aplicación de Visual Basic para dotar al operario de un control total del sistema desde un terminal

5. PRESUPUESTO

En este apartado se detalla la lista de elementos necesarios para realizar el montaje de la simulación del proyecto.

En todos los precios está incluido el IVA.

Item.	Unidades	Precio/ud	Subtotal
Arduino UNO R3	1	20.00€	20.00€
Sensor YL38 – YL39	2	2.99€	5.98€
DHT11	2	2.45€	4.90€
Módulo 8 Relés	1	8.35€	8.35€
LDR	2	1.17€	2.34€
Pot. Lineal 10KΩ	2	2.2€	4.4€
Protoboard	1	13.3€	13.3€
Cable para conexiones	1	1.03€	1.03€
Raspberry PI mod B	1	25.25€	25.25€

Total: 85.55€

Tabla (4): Tabla presupuesto total del Proyecto

6. ANEXOS

Listado de Figuras

- Figura (1): Vista frontal de Arduino UNO
- Figura (2): Sensor DHT11
- Figura (3): Sensor YL38 y YL69
- Figura (4): Fotorresistencias (LDR)
- Figura (5): Potenciómetro
- Figura (6): Modulo Relés
- Figura (7): Estructura del Sistema
- Figura (8): Montaje del nodo principal
- Figura (9): Conexión del sensor DHT11
- Figura (10): Conexión del sensor YL-38 y YL-69
- Figura (11): Conexión del módulo 8 Relés y Circuitos 230V
- Figura (12): Conexión de LDR
- Figura (13): Conexión de potenciómetros a Entradas analógicas
- Figura (14): Declaración de variables y librerías
- Figura (15): Void Setup del código principal
- Figura (16): Recepción de datos del programa principal
- Figura (17): Control de temperatura del programa principal
- Figura (18): Control de humedad del programa principal
- Figura (19): Vista de la aplicación VB en funcionamiento
- Figura (20): Código del timer en Visual Basic

Listado de Tablas:

- Tabla (1): Características de las placas Arduino.
- Tabla (2): Características de Arduino UNO
- Tabla (3): Resumen entradas /Salidas del sistema
- Tabla (4): Tabla presupuesto total del Proyecto

7. REFERENCIAS BIBLIOGRAFICAS Y WEBGRAFIA

ARDUINO. (2016). Built-in examples. Recuperado de <http://goo.gl/nUvuPN>

MICROSOFT. (2016). Guia de programación en Visual Basic. Recuperado de <http://goo.gl/3h14nc>

MICROSOFT. (2016). Visual Basic. Recuperado de <http://goo.gl/pgcc3J>

PROMETEC. (2016). Curso Arduino. Recuperado de <http://goo.gl/IKYpYD>

PROMETEC. (2016). Primeros pasos con Arduino. Recuperado de <http://goo.gl/7Eftls>

RASPBERRY PI.(2016). Raspberry Pi 2 Model B. Recuperado de <http://goo.gl/C69EV6>