**GREEN HOUSE MONITORING AND CONTROL**

**INTRODUCTION**

* The objective of this Case Study is to design a simple, easy to install, microcontroller-based circuit to monitor and record the values of temperature, humidity, soil moisture and sunlight of the natural environment that are continuously modified and controlled in order optimize them to achieve maximum plant growth and yield.
* It communicates with the various sensor modules in real-time in order to control the light, aeration and drainage process efficiently inside a greenhouse by actuating a cooler, fogger, dripper and lights respectively according to the necessary condition of the crops.

**Abstract**

Agriculture is a major part of our lives as human beings. A lot of research has been carried out in order to be able to develop a monitored and controlled greenhouse system/environment that will help in solving the main problems relating to agriculture which is to enable the increase in the crops being cultivated all year round in the comfort of a small space like the home, and also to reduce human interaction in a small-scale greenhouse environment. So accordingly, an automated greenhouse monitoring and control system was proposed for the sole purpose stated above. The methodology used in building the greenhouse monitoring and control system is a wired connection. The system was built using a number of connection wires, sensors, LCD, a cooling system, a power bank, LEDs, LDRs, Arduino board among a few other components. The result obtained was a fully functioning system that was set to monitor the greenhouse environment

**High Level Requirements**

* Identify measurable variables important to production.
  + It is very important to correctly identify the parameters that are going to be measured by the controller’s data acquisition interface, and how they are to be measured.
* Investigate the control strategies.
  + An important element in considering a control system is the control strategy that is to be followed. The simplest strategy is to use threshold sensors that directly affect actuation of devices.
* Identify the software and the hardware to be used.
  + Hardware must always follow the selection of software, with the hardware required being supported by the software selected.

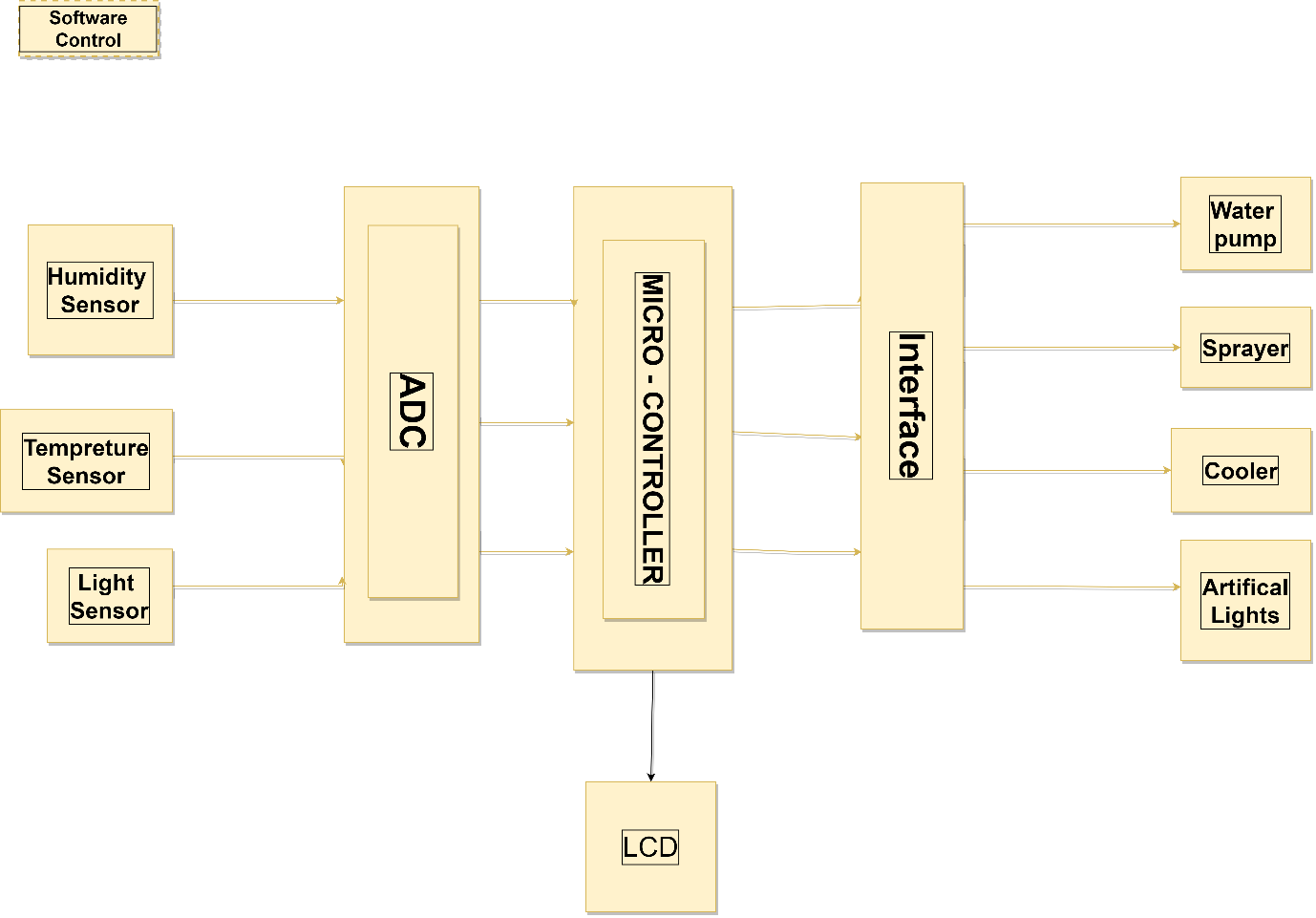
**Low Level Requirements**

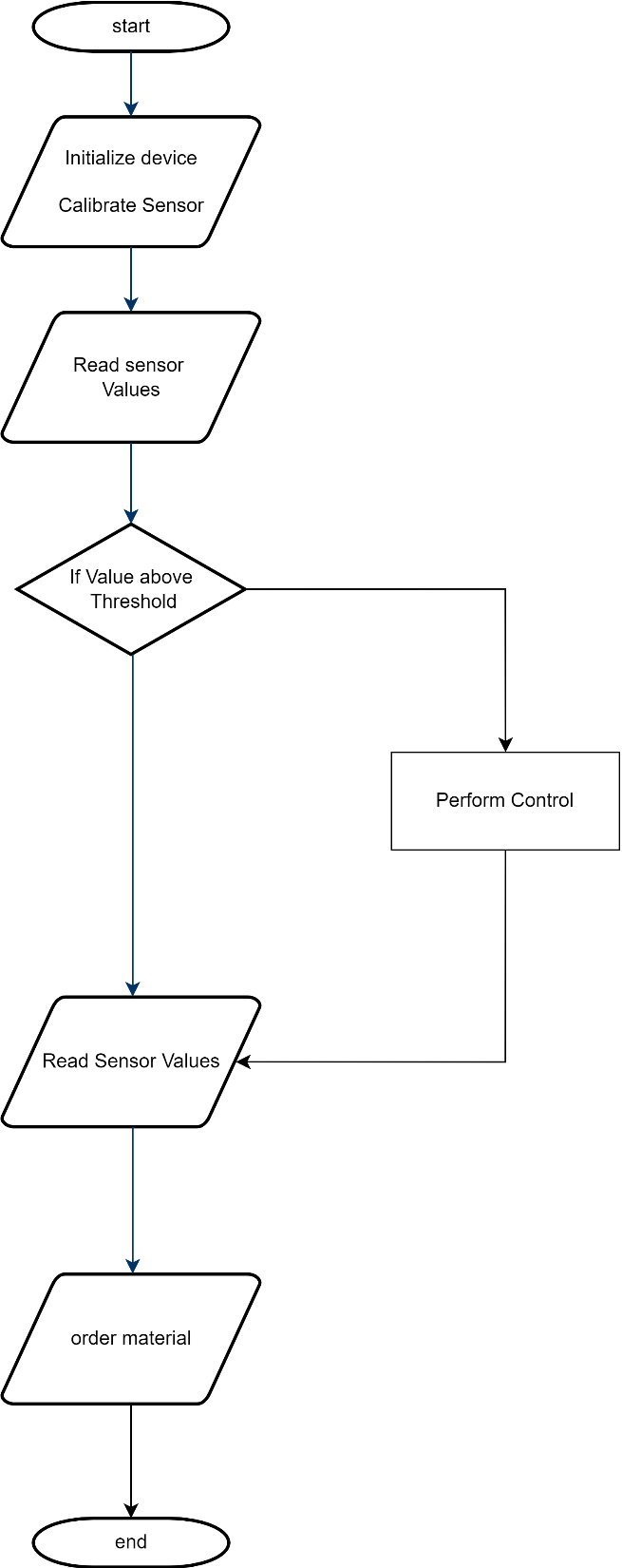
* Which type of strategy to follow
* The project shall give the output very fast
* The control hardware should include factors such as reliability, support, previous experiences with the equipment (successes and failures), and cost

**Advantages**

* Also, the use of easily available components reduces the manufacturing and maintenance costs.
* The design is quite flexible as the software can be changed any time.
* It can thus be tailor-made to the specific requirements of the user.
* This makes the proposed system to be an economical, portable and a low maintenance solution for greenhouse applications, especially in rural areas and for small scale agriculturists.

Block Diagram



Overflow Of Project

### **Parts of the System**

* Sensors (Data acquisition system)
  + Temperature sensor
  + Humidity sensor
  + Light sensor (LDR)
* Analog to Digital Converter (ADC)
* Microcontroller (AT89C51)
* Liquid Crystal Display
* Actuators – Relays (Interface)
* Devices controlled
  + Water Pump
  + Sprayer
  + Cooler
  + Artificial Lights

**Humidity & Temperature Sensor**

* humidity & temperature Sensor DHT11 module
  + Different crop species have different optimum growing temperatures and humidity.
  + Typical greenhouse temperatures vary between 10-30°C.
  + humidity varying between 20 and 70 %RH.

Flow Char of Humidity Sensor

Diagram

Description automatically generated

Flow Chart of Temperature Sensor

Diagram

Description automatically generated with low confidence

A picture containing text, electronics

Description automatically generated

**Light Sensor (LDR)**

A simple light intensity sensor can be constructed using light dependent resistance (LDR). By using light sensor, the illumination of light is detected for optimum or dim or dark or night.

Diagram, engineering drawing

Description automatically generated

Flow Char of Light Sensor

A picture containing polygon

Description automatically generated

**PIC18F2550**

* This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price
* with the addition of high endurance, Enhanced Flash program memory.
* In addition to these features, the PIC18F2455/2550/4455/4550 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

Diagram

Description automatically generated

Table

Description automatically generated

### **Analog to Digital Converter (ADC)**

### In physical world parameters such as temperature, pressure, humidity, and velocity are analog signals.

### A physical quantity is converted into electrical signals.

### We need an analog to digital converter (ADC), which is an electronic circuit that converts continuous signals into discrete form so that the microcontroller can read the data.

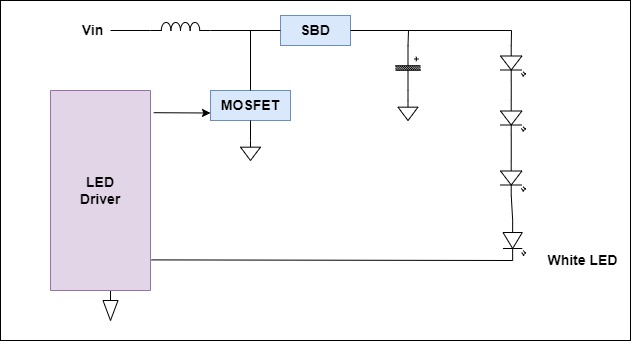
### Analog to digital converters are the most widely used devices for data acquisition.

Graphical user interface

Description automatically generated with medium confidence

### **Liquid Crystal Display**

* A liquid crystal display (LCD) is a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector.
* Each pixel consists of a column of liquid crystal molecules suspended between two transparent electrodes, and two polarizing filters, the axes of polarity of which are perpendicular to each other.

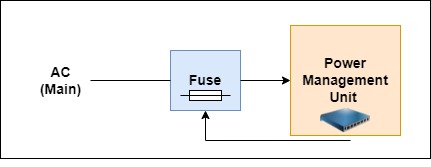


### **Relays**

* A relay is an electrical switch that opens and closes under the control of another electrical circuit.
* In the original form, the switch is operated by an electromagnet to open or close one or many sets of contacts.
* Because a relay is able to control an output circuit of higher power than the input circuit, it can be considered to be, in a broad sense, a form of an electrical amplifier.

### **Power Supply Connection**

The power supply section consists of step-down transformers of 230V primary to 9V and 12V secondary voltages for the +5V and +12V power supplies respectively.



## SUB – SYSTEM TESTING

| **Test\_ID** | **Sub-System Test Cases** |
| --- | --- |
| TC01 | Verify the functionality of the on/off button. |
| TC02 | Verify if it’s working in every temperature setting. |
| TC03 | Verify the minimum temperature that can be achieved by the AC. |
| TC04 | Verify the maximum temperature that can be achieved by the AC. |
| TC05 | Verify Humidity sensor in different climate change (Moisture, Dry). |
| TC06 | Verify light sensor (Bright, Dark). |
| TC07 | Verify Power Management unit status for high voltage. |
| TC08 | Verify Power Management unit status for low voltage. |
| TC09 | Verify the information Displayed in the display panel is correct or not. |
| TC10 | Verify if the display isn’t too bright or too dark. |

## SYSTEM TESTING

| **TEST\_ID** | **System Test cases** |
| --- | --- |
| TC01 | Verify the remote controls. |
| TC02 | Verify how accurately Microcontroller receives data. |
| TC03 | Verify all the relay connectivity. |
| TC04 | Verify the Output response. |
| TC05 | Display the data |

Algorithm for Greenhouse automation is as follows.

1. Start

2. Initialize ADC.

3. Initialize LCD.

4. Initialize UART

5. Get all set points from EEPROM

6. Get feedback from Temperature sensor

7. If temperature > Set temperature then turn ON the Cooler(FAN)

8. If temperature < Set temperature then turn ON the Heater

9. Get feedback from Humidity sensor

10. If humidity > Set humidity then turn ON the Humidifier

11. If humidity < Set humidity then turn ON the Dehumidifier

12. Get feedback from LDR

13. If light intensity < Set light intensity then turn ON the Bulb

14. If light intensity > Set light intensity then turn OFF the Bulb

15. If data received = „T‟ then update set temperature and and go to „6‟

16. If data received = „H‟ then update set humidity and go to „6‟

17. If data received = „L‟ then update set light intensity and go to „6‟

18. Go to „6‟

19. Stop.

**Source Code**

#include <LiquidCrystal.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <DHT\_U.h>

#define DHTPIN 2

#define DHTTYPE DHT11 // DHT 11

DHT\_Unified dht(DHTPIN, DHTTYPE);

uint32\_t delayMS;

LiquidCrystal lcd(12, 11, 5, 4, 3, 6);

int LDRpin = A0; // select the input pin for ldr

int LDRvalue = 0; // variable to store the value coming from the sensor

int moisturePin = A1;

int moistureValue = 0;

int threshold\_upper = 400;

int threshold\_lower = 250;

void setup()

{

Serial.begin(9600);

// Initialize device.

dht.begin();

lcd.setCursor(0, 0); // top left

lcd.begin(16, 2);

sensor\_t sensor;

dht.temperature().getSensor(&sensor);

dht.humidity().getSensor(&sensor);

// Set delay between sensor readings based on sensor details.

delayMS = sensor.min\_delay / 1000;

}

void loop()

{

// Delay between measurements.

delay(delayMS);

/\*\*\*\*\*\*\*\*\*\*\*\* DHT WORKING CODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

sensors\_event\_t event;

dht.temperature().getEvent(&event);

if (isnan(event.temperature))

{

Serial.println("Error reading temperature!");

}

else

{

Serial.print("Temperature: ");

Serial.print(event.temperature);

Serial.println(" \*C");

lcd.clear();

lcd.write("Temp: ");

lcd.print(event.temperature);

lcd.write(" \*C");

delay(1000);

}

// Get humidity event and print its value.

dht.humidity().getEvent(&event);

if (isnan(event.relative\_humidity))

{

Serial.println("Error reading humidity!");

}

else

{

Serial.print("Humidity: ");

Serial.print(event.relative\_humidity);

Serial.println("%");

lcd.clear();

lcd.write("Humidity: ");

lcd.print(event.relative\_humidity);

lcd.write("%");

delay(1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\* DHT CODE ENDS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\* LDR WORKING CODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDRvalue = analogRead(LDRpin); float Vout = (LDRvalue \* 0.0048828125);

float RLDR = (10000.0 \* (5 - Vout))/Vout;

float Lux = (500 / RLDR);

Serial.print("LDR VALUE: ");

Serial.println(LDRvalue); delay(100);

lcd.clear();

lcd.write("Light: ");

lcd.print(Lux);

lcd.write(" lux");

delay(1000);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* END OF LDR \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* MOISTURE SENSOR CODE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

moistureValue = analogRead(moisturePin);

Serial.print("MOISTURE VALUE: ");

Serial.println(moistureValue);

delay(100);

lcd.clear();

lcd.write("Moisture: ");

lcd.print(moistureValue);

lcd.write(" V");

delay(1000);

if(moistureValue <= threshold\_lower)

{lcd.clear();

lcd.write("Wet! Leave it.");

delay(1000);

}

if(moistureValue >= threshold\_upper)

{lcd.clear();

lcd.write("Dry! Water it.");

delay(1000);

}

}

**RESULT**

**HUMIDITY SENSOR**

Tolerance= ±0.1V

HUMIDITY SENSOR READINGS

|  |  |
| --- | --- |
| **Percentage RH (RELATIVE HUMIDITY)** | **Transducer Optimum Range** |
| 0% | 0-0.8V |
| 0% to 9.81% | 0.8-1.1V |
| 12.9% to 20.1% | 1.2-1.45V |
| 22.7% to 30.06% | 1.5-1.725V |
| 30.8% to 40.5% | 1.75-2.05V |
| 41.3%to50.3% | 2.075-2.35V |
| 51%to 60.02% | 2.375-2.65V |
| 61.6%to70.5% | 2.7-2.975V |
| 71%to80.2% | 3-3.275V |
| 81.1%to 90% | 3.3-3.6V |
| 91%to 100% | 3.6-3.9V |

**TEMPERATURE SENSOR**

TEMPERATURE SENSOR READINGS

|  |  |
| --- | --- |
| Temperature range in degree Celsius | Temperature sensor output  ( Vout ) |
| 10 0C | 0.5V |
| 15 0 to 20 0C | 0.75-1.0V |
| 20 0 to 25 0 C | 1.0-1.25V |
| 25 0 to 30 0C | 1.25-1.5V |
| 30 0 to 35 0C | 1.5-1.75V |
| 350 to 40 0C | 1.75-2.0V |
| 40 0 to 45 0C | 2.0-2.25V |
| 45 0 to 50 0 C | 2.25-2.5V |
| 50 0 to 55 0C | 2.5-2.75V |
| 55 0 to 60 0C | 2.75-3.0V |
| 60 0 to 65 0C | 3.0-3.25V |
| 65 0 to 70 0C | 3.25-3.5V |
| 70 0 to 75 0C | 3.5-3.75V |
| 75 0 to 80 0C | 3.75-4.0V |
| 80 0 to 85 0C | 4.0-4.25V |
| 85 0 to 90 0C | 4.25-4.5V |
| 90 0 to 95 0C | 4.5-4.75V |
| 95 0 to 100 0C | 4.75-5V |

**LIGHT SENSOR**

Tolerance = ±0.1V

LIGHT SENSOR READINGS

|  |  |
| --- | --- |
| **Illumination Status** | **Transducer Optimum**  Range |
| OPTIMUM  ILLUMINATION | 0V-0.69V |
| DIM LIGHT | 0.7V-2.5V |
| DARK | 2.5V- 3V |
| NIGHT | 3V-3.47V |

The unit will monitor the conditions of various parameter considerations and take appropriate action. Action taken is as follows:

* If temperature is lower than the set point: Heater relay will Turn ON and fan will turn OFF.
* If temperature is higher than the set point: then Fan will be Turned ON and heater will turn OFF.
* If Humidity is higher than the set point: then Fan / Humidifier will be Turned ON and heater / Dehumidifier will turn OFF.
* If Humidity is lower than the set point: then Fan / Humidifier will be Turned OFF and heater / Dehumidifier will turn ON.
* If temperature is higher than the set point: then Fan will be Turned ON and heater will turn OFF.
* If Light Intensity is higher than the set point: turn OFF the bulb.
* If Light Intensity is lower than the set point: turn ON the bulb.

**CONCLUSION**

The Project thereby undertaken has deepend the understanding of the associated course of Embedded Systems and PIC microcontroller. The project is a working prototype of a greenhouse monitoring system. The developed hardware can appropriately measure ambient temperature, humidity, light intensity the system is getting and the soil moisture content. All this information can ease the work of human taking caring of a greenhouse. This gives suitable environment to the plants to grow and a better yield.

This project has been simulated with the PIC18 family controller, but the hardware has been built with the Arduino Uno board. The LCD interfaced with the system shows the measured parameters.

The project has been successfully completed with good understanding of programming PIC microcontroller.