Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2020

Earth Wise
Smart Green House

Duy Nguyen
Paras Shrestha
Andrea Rivera
Christopher Jones
Walter Mkpanam

Revision History

Revision	Date	Author(s)	Description
0.1	08.05.2020	AR	Document creation
0.2	08.06.2020	DN	System overview creation
1.0	08.20.2020	DN	System overview review

Contents

List of Figures

List of Tables

1 Introduction

This project aims to create a system for managing a small greenhouse through the use of a variety of sensors and devices to automate the process of managing a greenhouse. Ultimately the system hopes to provide a scalable way to promote plant growth as well as guides to help customers and users be successful in their plant growing endeavors. This document provides information about the goals for the core system functionality. This includes sub systems such as irrigation, cooling, heating, display information, and the control system. When combined, they form the greenhouse system that this projects aims to achieve. Once a system is proven to work, the project will be scaled up to accommodate larger greenhouses with the hopes to provide a similar or better experience to the users.

Each system has its own goals, functionality and requirements necessary to the success of the project. The irrigation subsystem provides humidity, moisture control, and the watering of plants. In addition the water levels of the reservoir art tracked to ensure the irrigation system operates optimally. The temperature subsystem monitors current atmospheric temperature and the greenhouse temperatures and makes adjustments the temperature by activating the cooling and heating systems as necessary. The subsystem processes information collected by the Arduino and stored in a Firebase database to display information relevant to the consumer of the greenhouse system such as the current temperature, humidity, or if an error has occurred. The light subsystem activates and deactivates plant growth promoting lights as the time of day changes. Finally we have the control and communication subsystem that manages control of all the subsystems from the Arduino systems attatched to the greenhouse.

2 System Overview

The Architectural Design Specification of the Smart Green House will have six layers: Control and Communication, Irrigation, Temperature, Light, Fire Extinguisher, and Display. The main processor of the Smart Green House will be Control and Communication layer. The purposes of other systems, except the display system, are to provide useful values such as soil moisture level, light duration, etc. and be controlled by the control and communication layer based on users' settings of an Android app. The Display layer has the responsibility to display the current status of the Smart Green House.

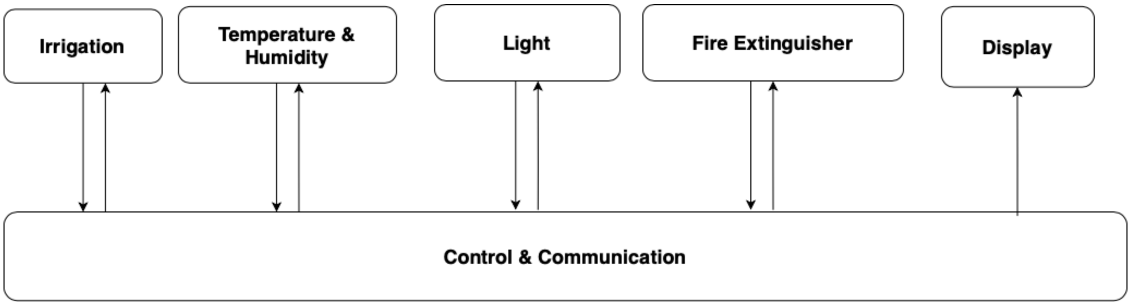

Figure 1: A simple architectural layer diagram

2.1 Control and Communication Layer Description

The Control and Communication layer has the responsibility to control all other systems: Irrigation, Temperature, Light, Fire Extinguisher, and Display. In another point of view, this layer acts as a database by storing the input values from an android app and adjust the output values of other systems by a micro-controller. To achieve this task, the Control and Communication layer needs to have the following subsystems: a micro-controller - Uno Arduino, a cloud database- Firebase, and an Android app. For example, to water the plants, the control and communication layer will receive the soil moisture level from the Irrigation system, then compare the moisture threshold value from Firebase that is updated from users' settings in an Android app. If the soil moisture is lower than the moisture threshold, Uno Arduino will turn on the Irrigation system.

2.2 Temperature and Humidity Layer Description

The temperature and humidity layer is responsible for regulating the temperature and humidity inside the greenhouse. The temperature and humidity layer has three subsystems; a temperature sensor, heating system and cooling/dehumidifying system. The sensor is responsible for accurately measuring the temperature and humidity inside the greenhouse and communicating with the control and communication layer so that the cooling and heating systems are turned on and off when appropriate. The cooling and humidity system is made up of fans and the water pump which turn on in unison and automatically turn off when the temperature or humidity is lowered to a suitable temperature for the plants. Similarly, the heating system turns on when the temperature goes under a given temperature and turns off when the temperature reaches the threshold again. The heating and cooling/dehumidifying systems can also be turned on and off manually by the users by using the application.

2.3 Light Layer Description

The Light layer is responsible for measuring light intensity and simulating growth for the plants. The light layer consists of two subsystems; the light sensor and the light bulb(s). The light sensor will measure the light intensity all plants are receiving and will communicate this data to the control and communication layer, specifically , the database hosted on Firebase would be modified and the Arduino

UNO would read the data from Firebase. Based on the settings made in the application, the control and communication layer will signal the two states of the light bulb (ON or OFF). Once intensity goes below the value specified, the bulb, which is a growth light will be turned on and will be turned off once the intensity goes above the value specified.

2.4 Fire Extinguisher Layer Description

The role of fire extinguisher layer is to detect any cases of fire and try to immediately extinguish the fire to minimize the possible damages. This layer will fulfill its responsibility with the help of two subsystems: Smoke sensor and Pump. The only job of the smoke sensor is to regularly check the level of smoke in surrounding air. The data from smoke sensor also needs to continuously fed into control and communication layer so that in any case of significant level of smoke in air, the control and communication layer can turn on the pump and activate the sprinkler system to put off the fire. The layer will also depend on temperature sensor from temperature layer to check if there is sudden spike in surrounding temperature. The readings from temperature sensor and smoke sensor are evaluated to properly detect case of fire.

2.5 Display Layer Description

The display layer is all functionality that notifies the user of the current status of the greenhouse. This layer is broken down into three categories. The first is the physical system that is attached to the greenhouse. This is a monitor that displays all information on the greenhouses current state such as the temperature, humidity, etc. This gets information directly collected from the Arduino displayed directly to the monitor. The next is another physical system attached to the greenhouse. This is an LED that notifies the user if there is a problem with the greenhouse. Lastly we have a digital system, an android app that functions similarly to the first category in that it displays all information about the state of the greenhouse. The difference is that the android app presents the information from the Firebase database instead of directly from the Arduino. All of these function to perform the main task of the display layer, that is to display any information a customer or user would need to monitor the greenhouse in an easily understood way.

2.6 Irrigation Layer Description

The irrigation layer is responsible to keep greenhouse irrigated without causing any over-watering issues. The irrigation layer is made up of three subsystems : Moisture sensor, Water level sensor and pump. The moisture sensor will be checking the current moisture level of soil in a continuous loop. The readings are transmitted to control and communication layer to decide if the moisture level is below desired percentage and turn the pump on as per necessity. As soon as the desired moisture is achieved, the pump will be again shut down until further required. The water sensor will be kept on water container and the job of this sensor is measure the volume of water in container. The readings are taken and transmitted to control layer in continuous manner and if the water level is low, the control and communication layer can alert the user regarding the issue.

3 Subsystem Definitions & Data Flow

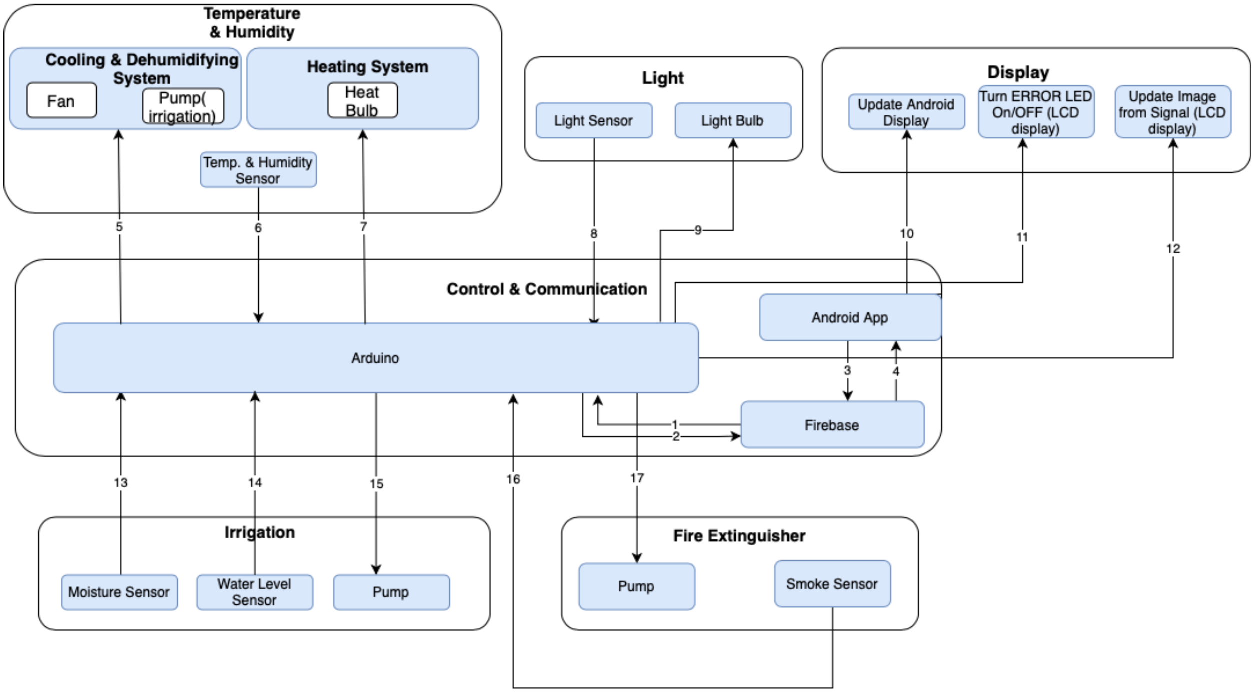

Figure 2: A simple data flow diagram

4 Control and Communication Layer Subsystems

4.1 Arduino

The Arduino is the main processor of the Smart Green House. It communicates with all subsystems of the project.

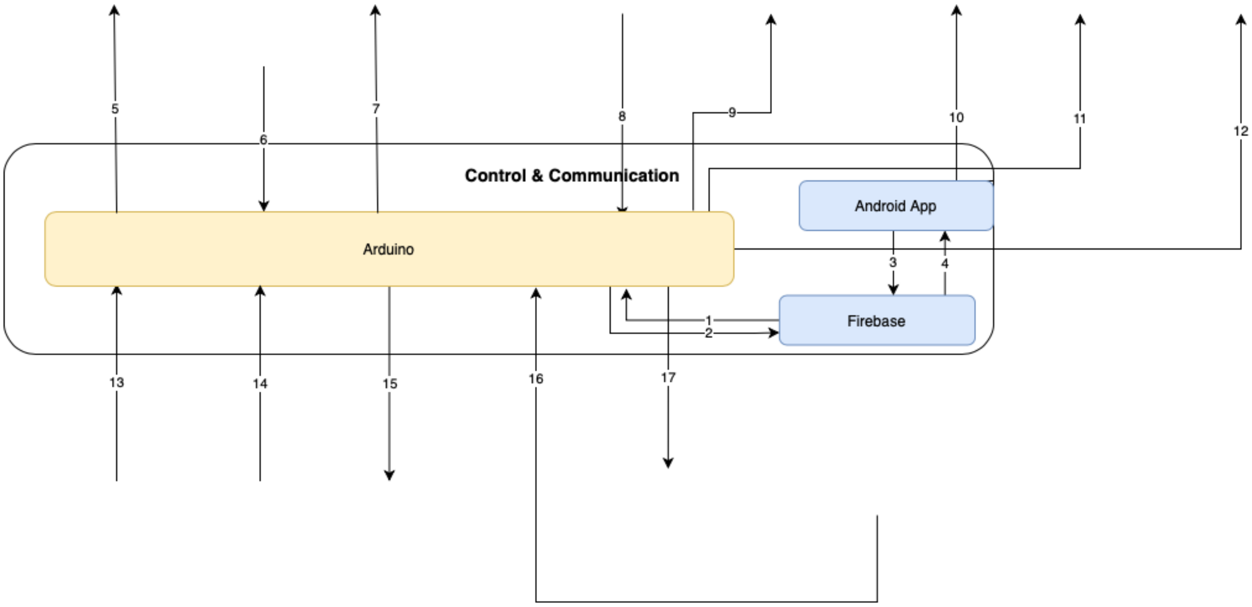

Figure 3: Arduino description diagram

4.1.1 Assumptions

The Arduino will :

- be connected to the outlet and have enough power to supply other subsystems
- always on, and connect to the internet during its operation.
- be in WiFi range area

4.1.2 Responsibilities

The Arduino subsystem is the main processor, controlling all input and output values, as well as upload necessary values to the Firebase subsystem. From the Irrigation system, this subsystem will monitor the moisture level and water the plants accordingly. From the Temperature system, the Arduino will monitor the temperature inside the greenhouse to control the following subsystems Fan, Pump, and Heater. From the Light system, this subsystem will monitor the exposed light duration of the plants and control the light based on the recorded value. From the Fire Extinguisher system, the Arduino will monitor the smoke level inside the greenhouse and turn on the water when the smoke level is higher than expected. From the Display system, the Arduino will display the current status value of the greenhouse, such as soil moisture level, light duration which the plants have exposed, inside temperature. to an LCD display.

4.1.3 Subsystem Interfaces

Each of the inputs and outputs for the subsystem are defined here. Create a table with an entry for each labelled interface that connects to this subsystem. For each entry, describe any incoming and outgoing data elements will pass through this interface.

Table 2: Arduino interfaces

ID	Description	Inputs	Outputs
#1	Update soil moisture, light duration, temperature thresholds to the Arduino	Soil Moisture Threshold Light Duration Threshold Temperature Threshold	N/A
#2	Send current soil moisture, light duration, temperature values of the greenhouse	N/A	Current Soil Moisture Current Light Duration Current temp. value
#5	Cool down the greenhouse environment	N/A	Voltage to Fan Voltage to Pump
#6	Monitor the inside temperature	Inside Temp. Value	N/A
#7	Warm up the greenhouse environment	N/A	Voltage to Heater
#8	Monitor the time that the plants have exposed to the sunlight	Light Sensor Value	N/A
#9	Turn on the artificial light in a period of time	N/A	Voltage to Light Bulb
#11	Indicate error of the product	N/A	Signals to LED
#12	Display current moisture value, light duration, temperature environment	N/A	Signals to LCD Display
#13	Monitor soil moisture level	Soil Moisture Value	N/A
#14	Monitor current water volume in the water container	Current Water Volume	N/A
#15	Water the plants	N/A	Voltage to Pump
#16	Detect the smoke level inside the greenhouse	Smoke Sensor Value	N/A
#17	Extinguish the fire	N/A	Voltage to Pump

4.2 Firebase

The Firebase is the database of the Smart Green House. This subsystems will store current values and threshold points of soil moisture, light duration, temperature, smoke, humidity, plant guides and users' credentials.

4.2.1 Assumptions

- all sensors works properly
- Firebase has enough space to store all values
- established connection between Arduino, Firebase, and Android app.

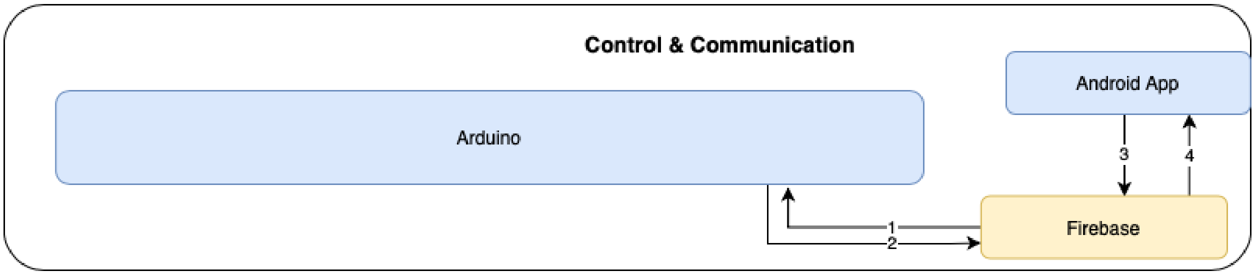

Figure 4: Firebase description diagram

4.2.2 Responsibilities

The Firebase subsystem acts as a middle-ware of Arduino and Android App for communication and storing necessary information. From Android App, users can set up threshold points of soil moisture, light duration, temperature, smoke, and humidity by uploading those values to Firebase. Then, this subsystem has the responsibility to notify what values have changed to Arduino, and vice versa. The Firebase also needs to notify the Android app what has changed after Arduino uploads necessary values to Firebase.

4.2.3 Subsystem Interfaces

Table 3: Firebase interfaces

ID	Description	Inputs	Outputs
#1	Notify the Arduino what have changed in the Firebase	N/A	Soil moisture threshold Light duration threshold Temperature threshold Smoke threshold Humidity threshold
#2	Update database such as current soil moisture, temperature, light duration, humdity to the Firebase from Arduino	Soil moisture value Light duration value Temperature value Smoke value Humidity value	N/A
#3	Receive updated values from the Android App	Soil moisture threshold Light duration threshold Temperature threshold Smoke threshold Humidity threshold	N/A
#4	Notify the Android App what have changed in the Firebase	N/A	Soil moisture value Light duration value Temperature value Smoke value Humidity value

4.3 Android App

The Android App is the main controller. It controls all components of the Smart Green House.

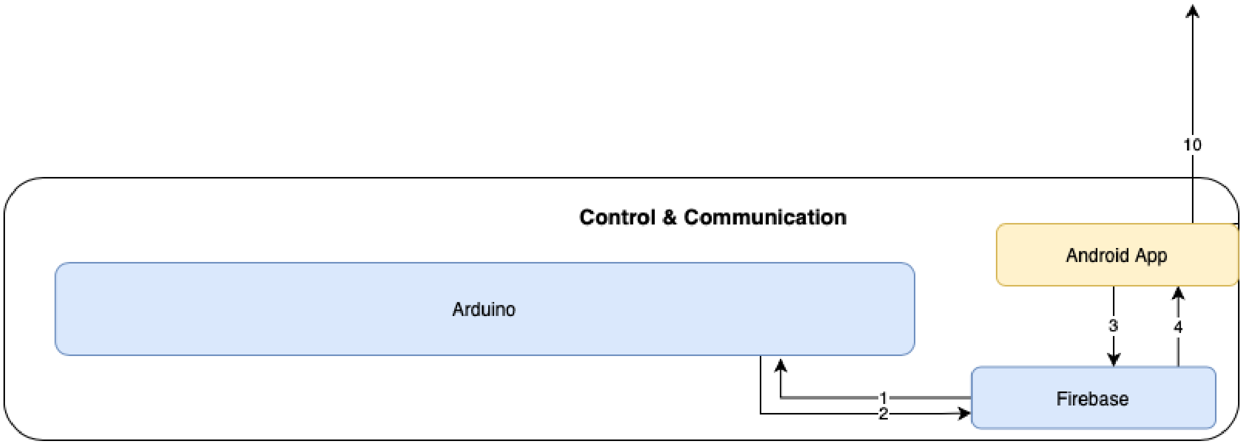

Figure 5: Android Application description diagram

4.3.1 Assumptions

- Android phone has enough space to install the app.
- Android phone can connect to the internet.
- other systems work properly.
- Firebase has enough space to store all values.
- Established connection between Arduino, Firebase, and Android app.

4.3.2 Responsibilities

The Android App has the responsibility to update the database in Fire base to control all other systems: temperature, light, display, irrigation systems. Its function is also to display to users what went wrong to the product. For example, to water the plants, a user needs to use an Android App to update the soil moisture threshold of the Firebase, then the Firebase will communicate with Arduino to turn on the water.

4.3.3 Subsystem Interfaces

Table 4: Android Application interfaces

ID	Description	Inputs	Outputs
#3	Send user’s settings data to Firebase	N/A	Soil moisture threshold Light duration threshold Temperature threshold Smoke threshold Humidity threshold
#4	Receive updated values from Firebase	Soil moisture value Light duration value Temperature value Smoke value Humidity value	N/A
#10	Process received data and display to the screen	N/A	Soil moisture value Light duration value Temperature value Smoke value Humidity value Other systems status

5 Temperature and Humidity Layer Subsystems

5.1 Temperature and Humidity Sensor

Temperature sensor measures temperature and humidity which are used as inputs for the control and communication subsystem.

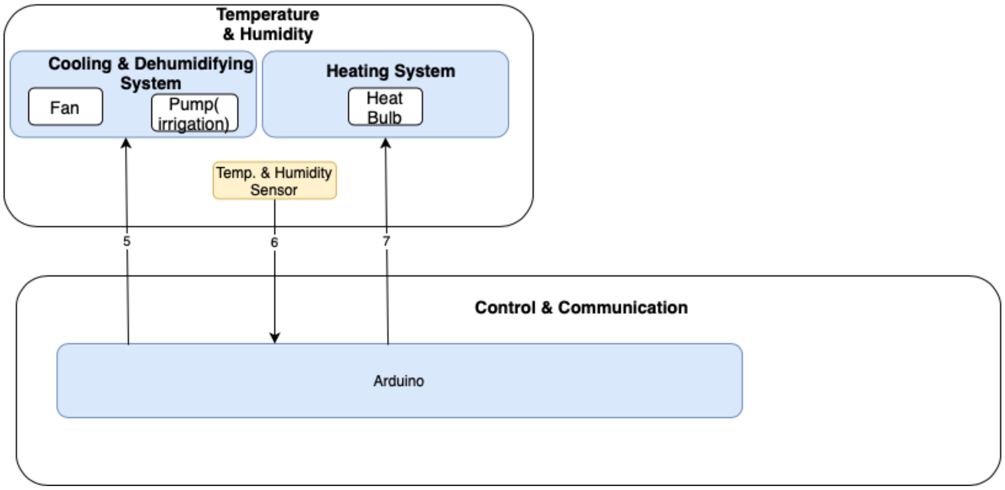

Figure 6: Temperature Sensor description diagram

5.1.1 Assumptions

- Sensor is properly connected to the Arduino.
- Control and communication layer is working correctly.
- Sensor is properly installed inside the greenhouse.
- Sensor is calibrated to ensure accurate temperature readings.

5.1.2 Responsibilities

The temperature and humidity sensor is responsible for continuously measuring temperature and humidity inside the greenhouse. The readings taken by the sensor are used by the control and communication subsystem as input to determine when to turn on and off the heating and cooling/humidifying systems.

5.1.3 Subsystem Interfaces

Table 5: Temperature sensor interfaces

ID	Description	Inputs	Outputs
#6	Measure temperature inside greenhouse	Temperature reading	Voltage to Arduino
#6	Measure humidity inside greenhouse	Humidity reading	Voltage to Arduino

5.2 Cooling and Dehumidifying System

Cooling and dehumidifying system utilizes fans and the water pump to lower the temperature and humidity inside the greenhouse.

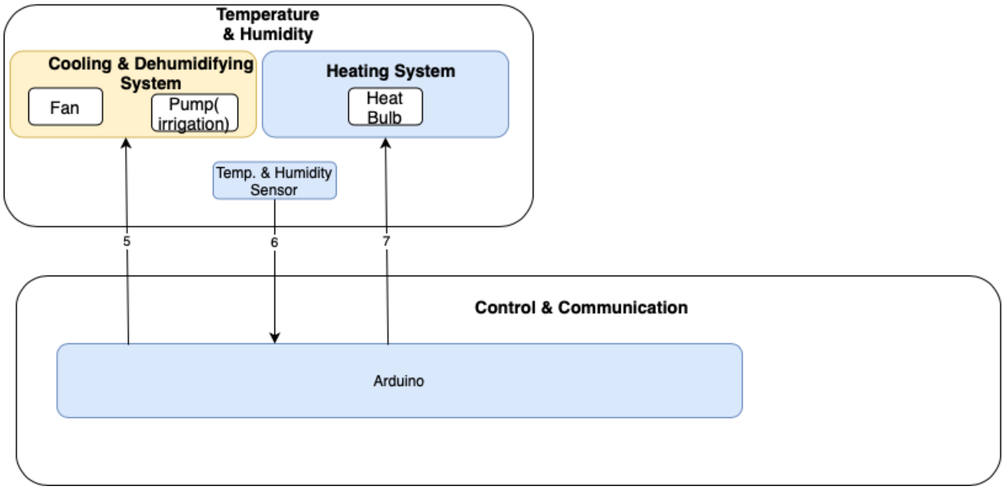

Figure 7: Cooling and Dehumidifier System description diagram

5.2.1 Assumptions

- Control and communication layer working properly.
- Temperature sensor is working properly.
- Fans and water pump properly connected to Arduino.
- Fans are working correctly.
- Water tank has sufficient amount of water.

5.2.2 Responsibilities

The cooling and dehumidifier system is responsible for lowering the temperature inside the greenhouse when prompted by the Arduino. The fans and irrigation system turn on only when the control and communication layer prompts it to do so. The system will turn off automatically when the greenhouse lowers to the desired temperature and humidity. Alternatively, the cooling system can be manually turned on and off by using the application.

5.2.3 Subsystem interfaces

Table 6: Cooling System interfaces

ID	Description	Inputs	Outputs
#5	Fans turn on/off.	Voltage from Arduino	N/A
#5	Pump turns on/off.	Voltage from Arduino	N/A

5.3 Heating System

Heating system raises temperature in the greenhouse when prompted by control system.

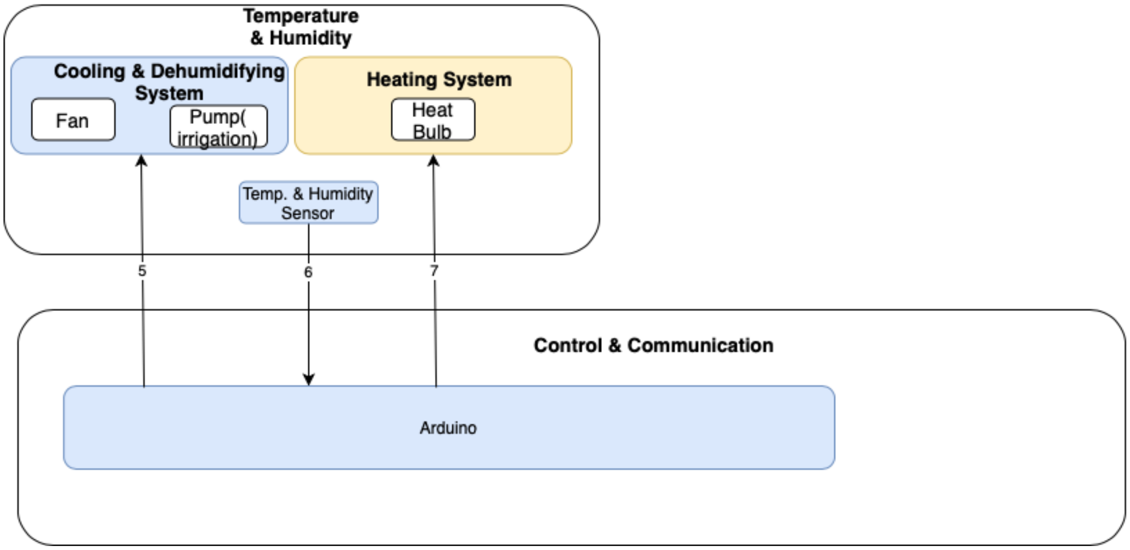

Figure 8: Heating system description diagram

5.3.1 Assumptions

- Control and communication layer works properly.
- Temperature sensor takes accurate readings.
- Heater works properly.
- Heater is properly connected to the Arduino.

5.3.2 Responsibilities

The heating system is responsible for raising the temperature in the greenhouse when prompted by the control and communication layer. The Arduino turns on the heater when the temperature falls below a certain threshold. The heater automatically turns off when the temperature in the greenhouse raises to the desired temperature. Users can also manually prompt the heater to turn on and off using the application.

5.3.3 Subsystem interfaces

Table 7: Heating System interfaces

ID	Description	Inputs	Outputs
#7	Heater turns on/off.	Voltage from Arduino	N/A

6 Light Layer Subsystems

6.1 Light sensor

The Light sensor measures light intensity which is used as input for the control and communication subsystem.

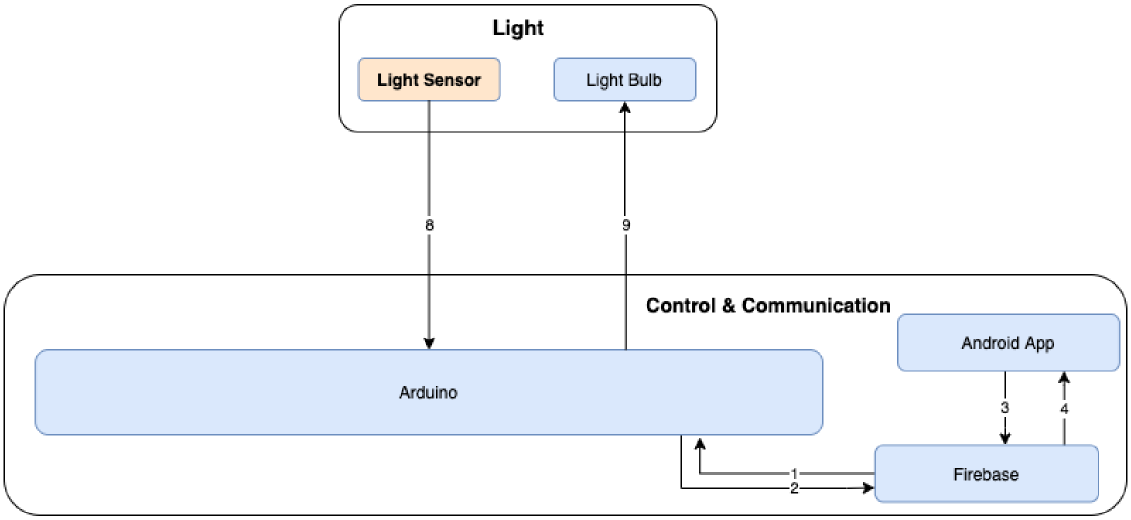

Figure 9: Light Sensor description diagram

6.1.1 Assumptions

- Light Sensor is properly connected to the Arduino.
- Control and communication layer is working correctly.
- Light Sensor is properly installed inside the greenhouse.
- Light Sensor is calibrated to ensure accurate light intensity readings.

6.1.2 Responsibilities

The light sensor is responsible for continuously measuring light intensity all plants are currently receiving inside the greenhouse. The readings taken by the sensor are used by the control and communication subsystem as input to show on the LED display and android application.

6.1.3 Subsystem Interfaces

Table 8: Light sensor interfaces

ID	Description	Inputs	Outputs
#8	Measure light intensity plants are receiving	Light intensity reading	Light intensity value

6.2 Light Bulb

The Light bulb simulates growth in the greenhouse when prompted by the user(through the application) or the control and communication system.

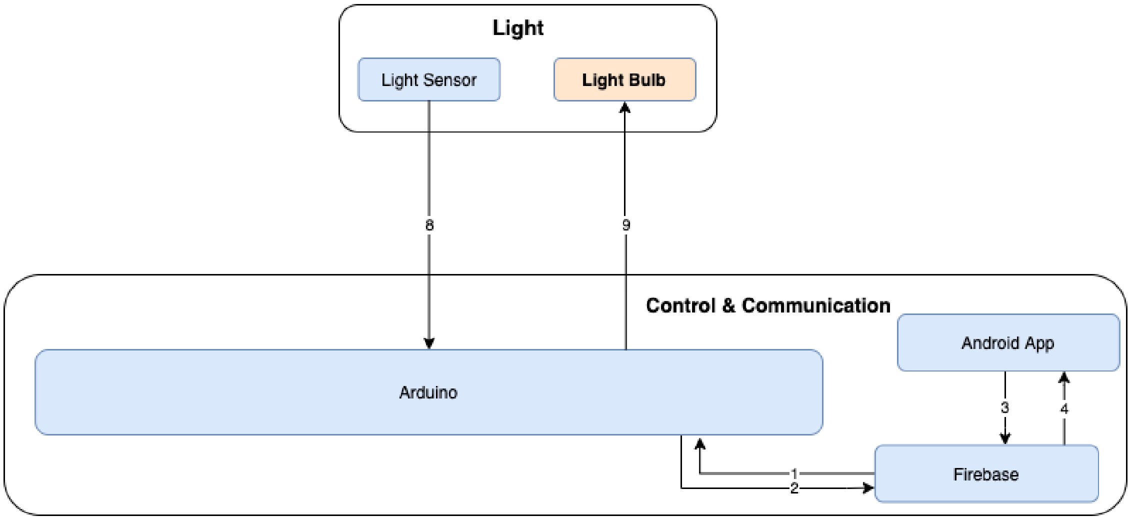

Figure 10: Light bulb description diagram

6.2.1 Assumptions

- Light bulbs are connected to Arduino properly.
- Control and communication layer is working correctly.
- Light sensor is properly installed and measuring accurate data
- Light bulbs function as specified.

6.2.2 Responsibilities

The light bulbs are responsible for simulating a consistent growth for the plants as weather conditions might not always be favorable. The state of the light bulbs (ON/OFF) will be prompted by the control and communication layer. The state can also be modified by a manual input from the android application.

6.2.3 Subsystem Interfaces

Table 9: Light bulb interfaces

ID	Description	Inputs	Outputs
#9	Light bulbs ON/OFF	Voltage from Arduino	N/A

7 Fire Extinguisher Layer Subsystems

7.1 Smoke Sensor

The smoke sensor will continuously check the smoke level in the surrounding air and will relay the reading to the control and communication layer. The Arduino in particular, will be in the receiving end in this case of data flow.

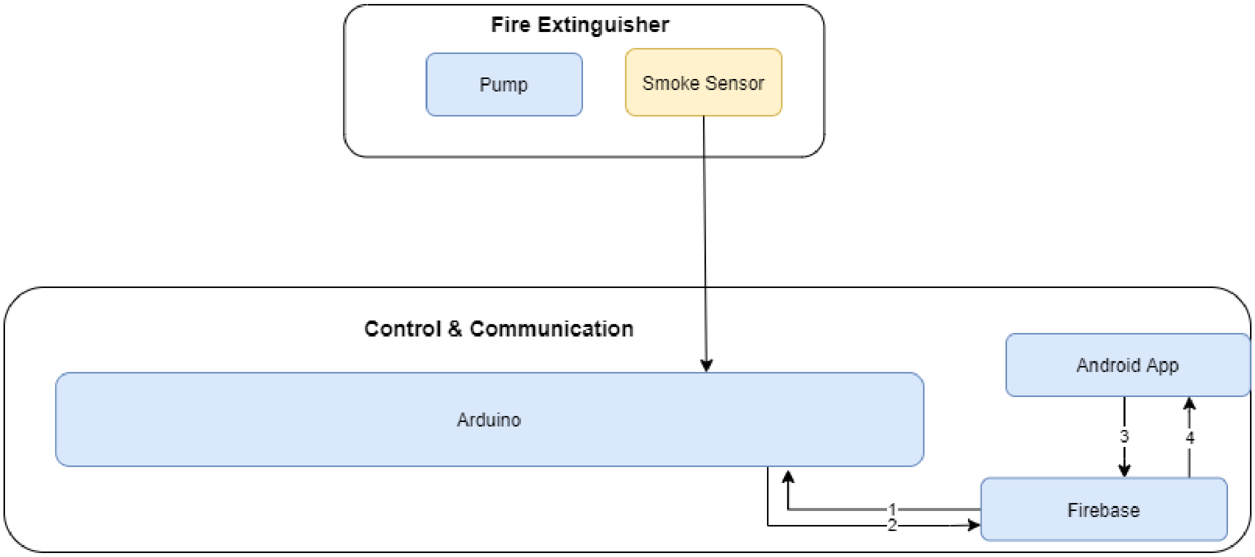

Figure 11: Smoke Sensor description diagram

7.1.1 Assumptions

- The sensor is properly calibrated to give correct readings.
- The Arduino is always ready to receive the data from sensor

7.1.2 Responsibilities

The only job of smoke sensor is take surrounding air sample and measure the level of smoke. The sensor has to then relay this information to arduino so that it check if there is too much smoke in air. There can be no exceptions and faults in smoke sensor as it is vital to detect any cases of fire. Once the information is received by arduino, it analyses the data and concludes if there is any risk present and take safety measures accordingly.

7.1.3 Subsystem Interfaces

Table 10: Smoke Sensor interfaces

ID	Description	Inputs	Outputs
#16	Measure the Smoke Level in air	Surrounding air Voltage from Arduino	Smoke reading

7.2 Pump

The pump will be fully controlled controlled by the Arduino. As per necessity, the Arduino will turn the pump on or off.

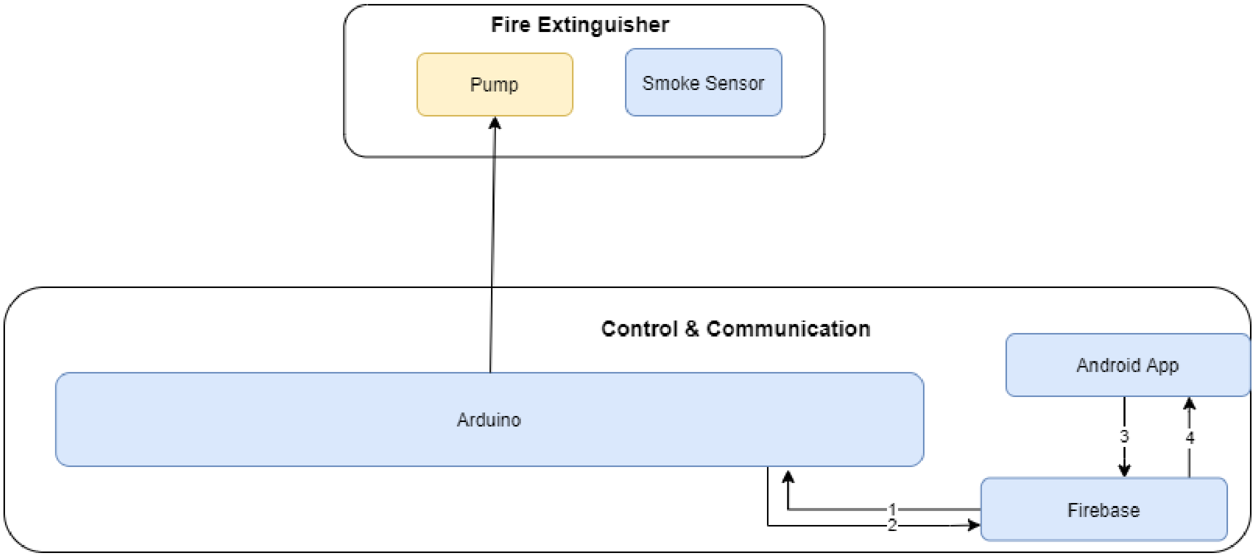

Figure 12: Pump description diagram

7.2.1 Assumptions

- The pump is properly powered and ready to work.
- The Arduino is programmed to turn pump on and off.
- Proper relay is being used to keep arduino away from high volt AC source.

7.2.2 Responsibilities

The Arduino will have full control over pump. It will decide when to turn the pump on and when to turn it off. If the readings from smoke sensor signals the case of fire, the pump will be turned on to activate sprinkler system. Since the pump is powered using AC source, Arduino can't directly control pump. So a relay is used which is controlled by Arduino which in turn controls the pump. Malfunction of pump can result in uncontrolled fire or flooding inside greenhouse. So the pump must be in proper state.

7.2.3 Subsystem Interfaces

Table 11: Pump Interfaces

ID	Description	Inputs	Outputs
#17	Pump ON/OFF	Voltage from Arduino	N/A

8 Irrigation Layer Subsystems

8.1 Moisture Sensor

The moisture sensor will take the moisture readings from the soil and pass that information to Control and communication layer for analysis purpose.

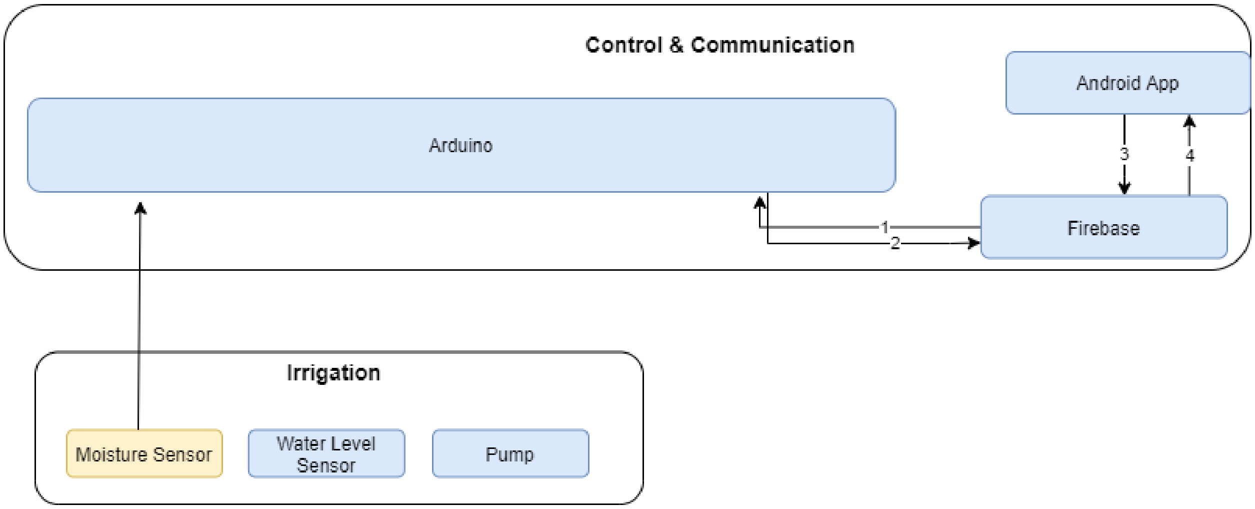

Figure 13: Moisture Sensor description diagram

8.1.1 Assumptions

- The Sensor is properly calibrated to give correct readings.
- The Arduino is always ready to receive the data from sensor
- The Sensor is properly put into the soil to right depth.

8.1.2 Responsibilities

The moisture is sensor is placed in the soil and it will take the moisture readings of soil regularly. These readings are passed on to Arduino to decide if moisture level is below the desired percentage. If so, the Arduino will start process to irrigate the soil. These sensors must be properly calibrated as wrong readings can cause greenhouse into desert or result in flooding.

8.1.3 Subsystem Interfaces

Table 12: Moisture Sensor interfaces

ID	Description	Inputs	Outputs
#13	Measure the soil moisture	Voltage from Arduino Water content from soil	Moisture reading

8.2 Water Level Sensor

This section should be a general description of a particular subsystem for the given layer. For most subsystems, an extract of the architectural block diagram with data flows is useful. This should consist of the subsystem being described and those subsystems with which it communicates.

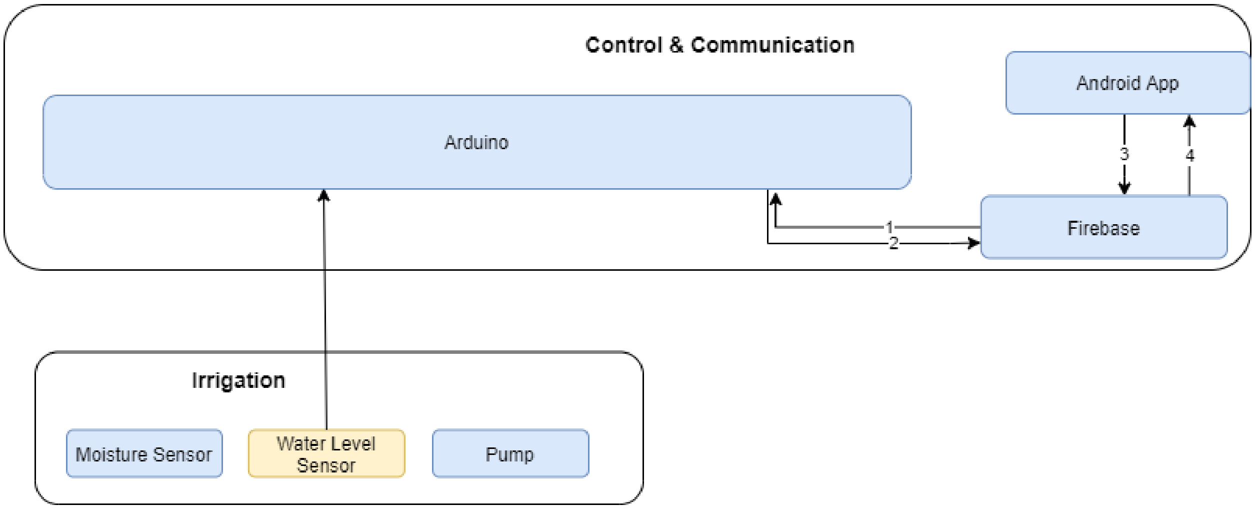

Figure 14: Water Level Sensor description diagram

8.2.1 Assumptions

- The Sensor is properly calibrated to give correct readings.
- The Arduino is always ready to receive the data from sensor

8.2.2 Responsibilities

The water level sensor is kept on the water tank and it is responsible to monitor the volume of water in the tank. The reading from the sensor is sent to the control and communication layer where the Arduino will decide if the level of water is below the minimum or not. If the sensor reading show the water is low, the Arduino will notify the user regarding the issue.

8.2.3 Subsystem Interfaces

Table 13: Water Level Sensor interfaces

ID	Description	Inputs	Outputs
#14	Measure water level in the tank	Voltage from Arduino Change in Capacitance	Water level reading

8.3 Pump

The pump will be fully controlled controlled by the Arduino. As per necessity, the Arduino will turn the pump on or off.

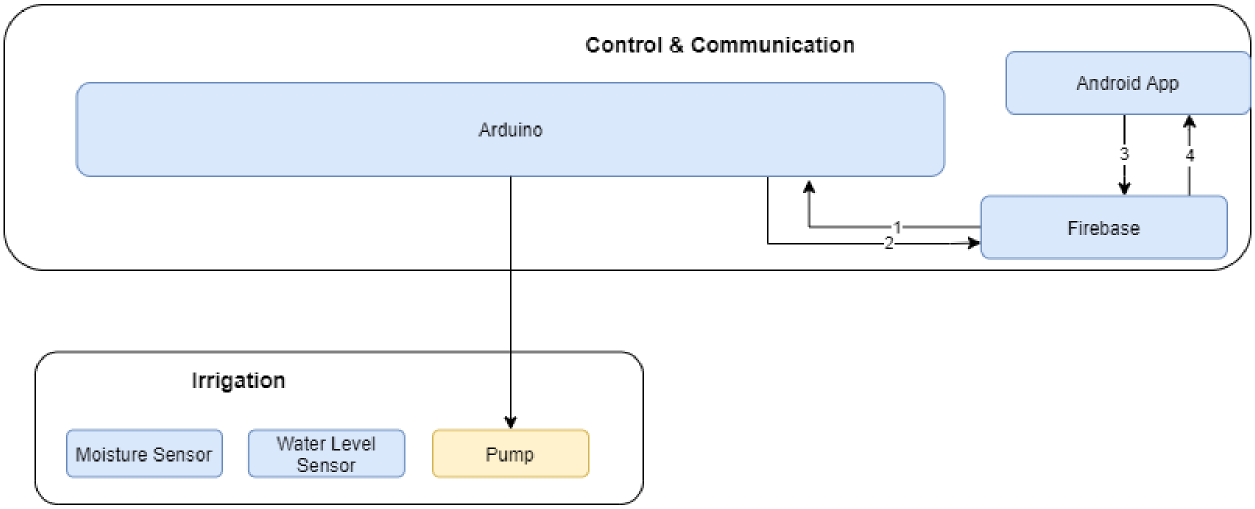

Figure 15: Pump description diagram

8.3.1 Assumptions

- The pump is properly powered and ready to work.
- The Arduino is programmed to turn pump on and off.
- Proper relay is being used to keep arduino away from high volt AC source.

8.3.2 Responsibilities

The Arduino will have full control over pump. It will decide when to turn the pump on and when to turn it off. If the readings from moisture sensor signals that soil is dry, the pump will be turned on to activate irrigation system. Whenever desired moisture is achieved, the pump is turned off again. Since the pump is powered using AC source, Arduino can't directly control pump. So a relay is used which is controlled by Arduino which in turn controls the pump. Arduino must make sure pump is properly handled or it can cause drought or flooding inside the greenhouse.

8.3.3 Subsystem Interfaces

Table 14: Pump interfaces

ID	Description	Inputs	Outputs
#15	Pump ON/OFF	Voltage from Arduino	N/A

9 Display Layer Subsystems

9.1 Android Display Updating

The display of the android application will update display data of the temperature, humidity, and other important information by pulling data from the Firebase server about the current state of the greenhouse.

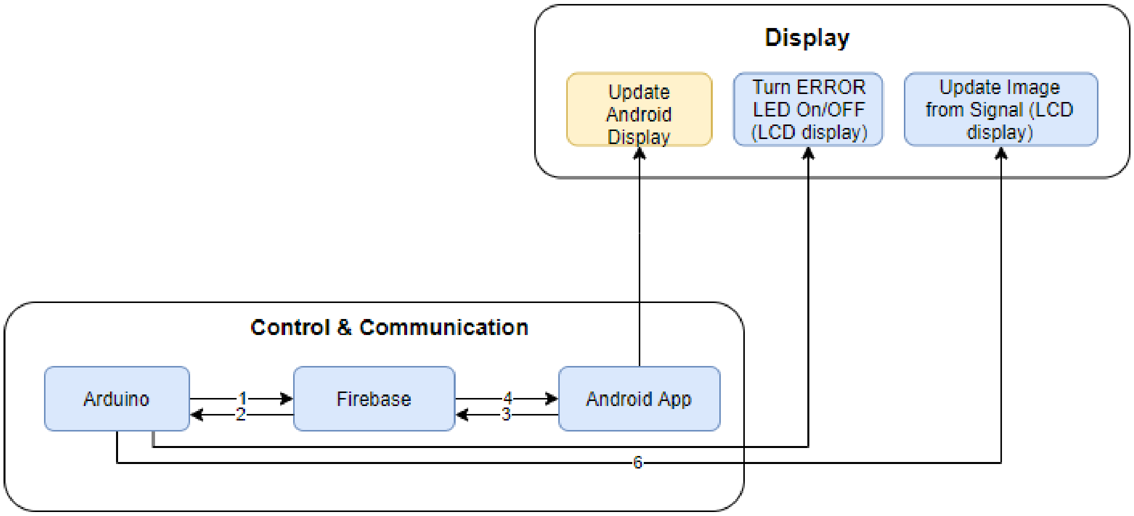

Figure 16: Update Android Display Diagram

9.1.1 Assumptions

- The Sensor is properly calibrated to give correct readings.
- The Arduino has updated Firebase with current greenhouse data such that it can be retrieved wirelessly.
- All sensors are functioning correctly.
- The android app will read the sensor data and make necessary calculations before displaying on the android app.

9.1.2 Responsibilities

The displayed information on the android application will run on the client side. Information to be displayed from greenhouse data will be pulled from Firebase data. The Arduino will update the Firebase periodically such that the application will need to periodically check Firebase for any changes. If there is a discrepancy in the data presented in Firebase and what is currently displayed on the android application, an update to the android display will be made to reflect these changes.

9.1.3 Subsystem Interfaces

Table 15: Android Display Interfaces

ID	Description	Inputs	Outputs
#10	Update Display	Signal from Firebase data	Update Display of Android Application

9.2 LED Error Signal

In the event of an error or malfunction of the greenhouse, this led acts as a physical notification that the greenhouse needs to be investigated or maintenance is required.

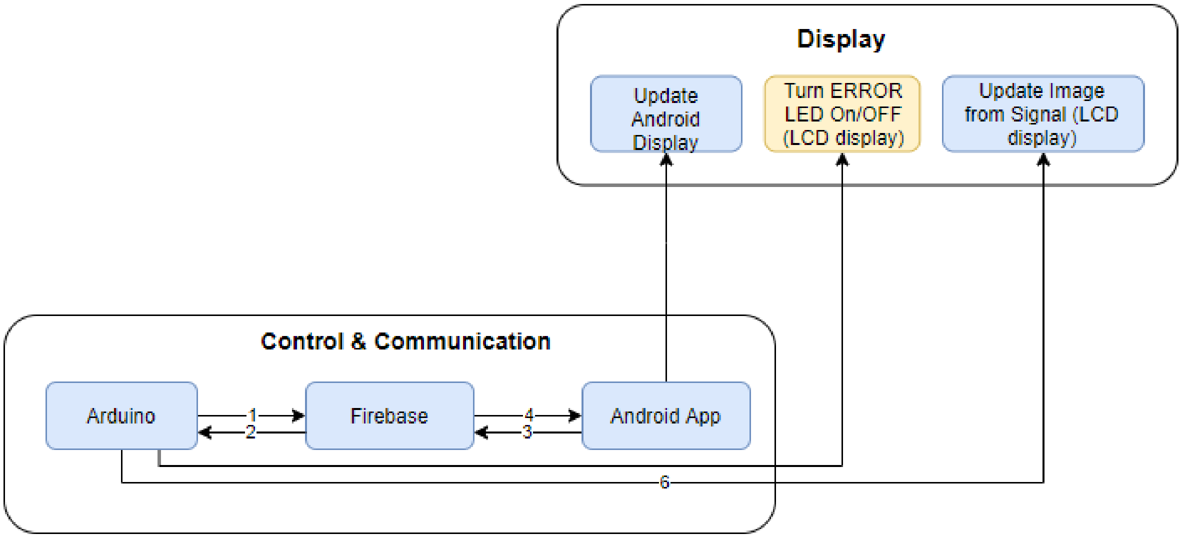

Figure 17: Update Android Display Diagram

9.2.1 Assumptions

- The LED is properly attatched to the greenhouse where it is visible.
- The Arduino detects anomaly in the greenhouse system.

9.2.2 Responsibilities

The LED's responsibility is to be a physical notification of a problem within the greenhouse system. The LED is mounted in a location on the greenhouse easily observable within 1 min of investigating the green house system. The LED will be red if there is an issue and green if no issue is found.

9.2.3 Subsystem Interfaces

Table 16: Error LED Interface

ID	Description	Inputs	Outputs
#11	Turn LED Green	Arduino Detects no Anomaly	LED is turned changed to a green color.
#11	Turn LED Red	Arduino Detects Anomaly	LED is turned changed to a red color.

9.3 Update Display Image

As the sensors data changes, the display of a monitor panel located near the greenhouse will be updated by the current readings of the Arduino. This provides a quicker response to the current state of the greenhose and allows the user to make necessary changes as they see fit from the provided data.

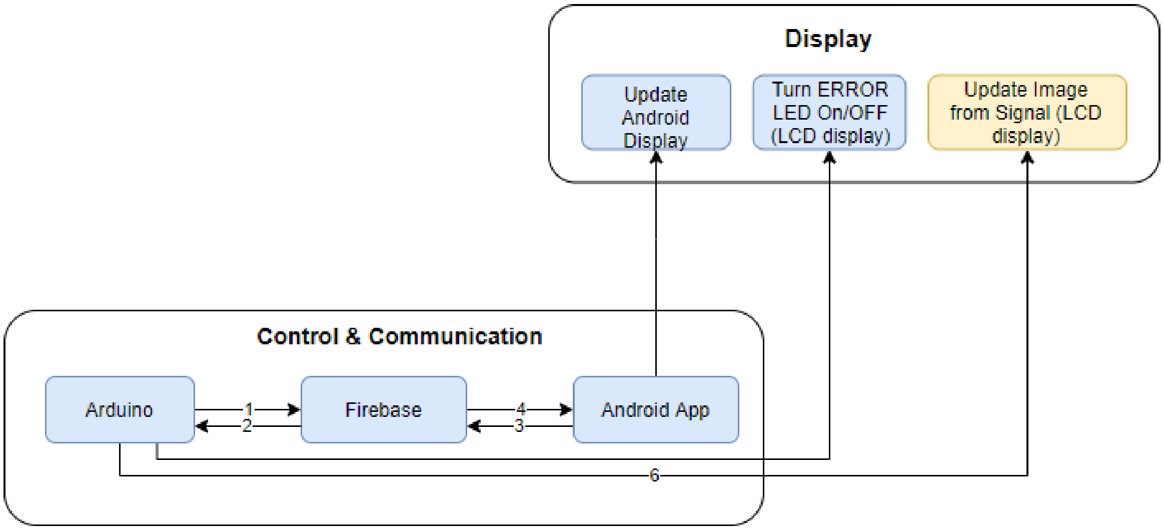

Figure 18: Update Android Display Diagram

9.3.1 Assumptions

- The display is attached or located nearby the greenhouse.
- The display gets display information from the Arduino managing the greenhouse system.

9.3.2 Responsibilities

The displays primary goal is to display accurate and current information about the state of the greenhouse. An Arduino collects and process the information, the appropriately updates the signal to the display such that the information is current.

9.3.3 Subsystem Interfaces

Table 17: Greenhouse Display Interface

ID	Description	Inputs	Outputs
#12	Update Greenhouse Display	Arduino Collects Information About the Greenhouse	Arduino update the signal to the display

REFERENCES