DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON

DETAILED DESIGN SPECIFICATION
CSE 4317: SENIOR DESIGN II
FALL 2020

EARTHWISE
SMART GREEN HOUSE

DUY NGUYEN
PARAS SHRESTHA
ANDREA RIVERA
CHRISTOPHER JONES
WALTER MKPANAM

Revision History

Revision	Date	Author(s)	Description
0.1	9.01.2020	AR	Document Creation
0.2	9.18.2020	WM	Light Layer Subsystem Complete
0.3	9.22.2020	CJ	Display Layer Subsystem Complete
0.4	9.22.2020	DN	Control Layer Subsystem Complete
0.5	9.24.2020	PS	Irrigation Layer Subsystem Complete
0.6	9.24.2020	PS	Fire Extinguisher Layer Subsystem Complete

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

Earthwise is a smart greenhouse which automates the process of growing plants and makes gardening simple and effortless for inexperienced gardeners and busy customers interested in growing their own produce. System requirements have been put in place to ensure that the smart greenhouse will take care of most aspects of gardening with minimum intervention from the user. Some of the most important requirements are that the greenhouse be self-watering, monitors and self-regulates temperature, measures and displays environment conditions inside the greenhouse and can be controlled remotely through a mobile application. Other system requirements such as packaging, safety and performance requirements are outlined and can be read in detail in the requirement specification document. The core functionality of the greenhouse can be divided into six layers including; Control and Communication, Irrigation, Temperature, Light, Fire Extinguisher, and Display layers. These layers can be further divided into subsystems including the Android application, heating and cooling and dehumidifying subsystems, among others which can be found and read in detail in the architectural design document. This document expands on the architectural design document and gives detailed implementation information about each layer and subsystem in the greenhouse. Each layer and subsystem has its own hardware, software, programming languages, operating systems, data structures and data processing that they utilize in order to successfully accomplish their individual tasks.

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, and other necessary data, and be controlled by the control and communication layer based on usersâ settings of an Android app. The Display layer has the responsibility to display the current status of the Smart Green House.

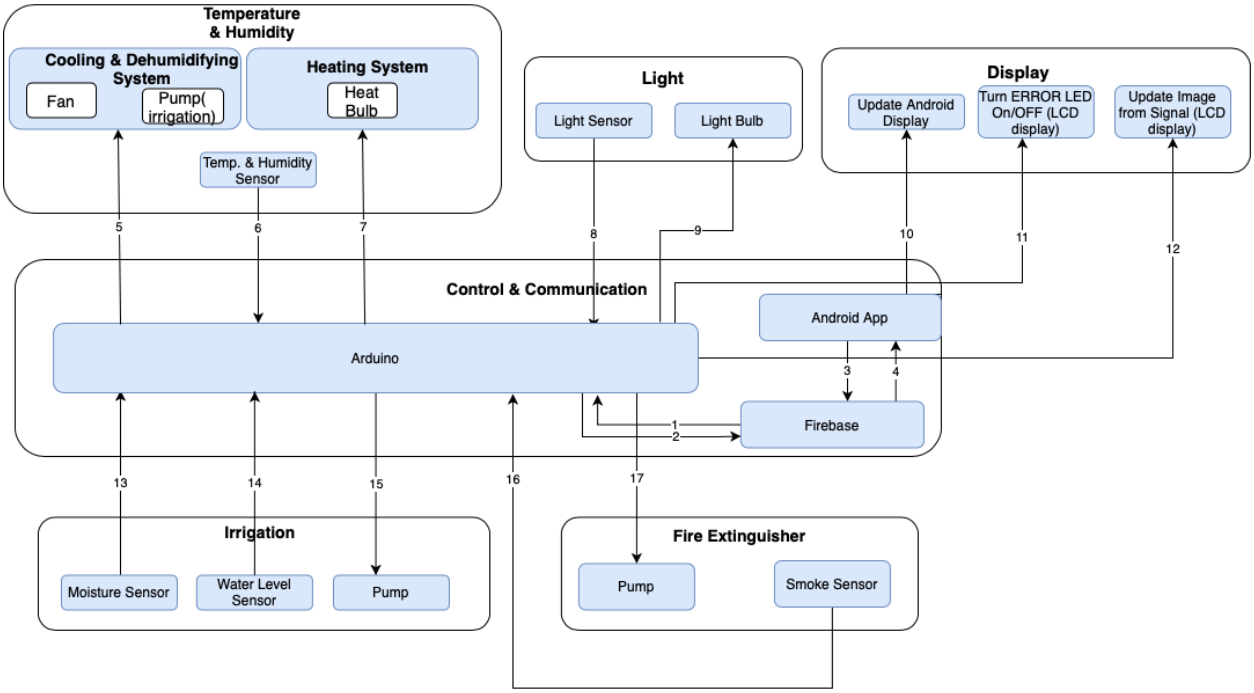

Figure 1: System architecture

3 Control and Communication Layer Subsystems

The control layer subsystem has a responsibility to control all other following systems: temperature and humidity, light, display, irrigation, and fire extinguisher systems. 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 microcontroller. 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 usersâ settings in an Android app. If the soil moisture is lower than the moisture threshold, Arduino Uno Rev 2 will turn on the Irrigation system.

3.1 Layer Hardware

The main hardware of the greenhouse product is Arduino Uno Rev 2.0. Users can control the greenhouse produce by an Android phone to edit their desired watering times, light level, humidity level, specific plants to grow, etc. To accomplish that feature, Arduino will connect to wifi, so Arduino will listen to the Firebase database update and control other systems accordingly.
The other hardware is a phone capable of running the Android operating system 4.4 KitKat or above. An Android app will help users to interact with the product.

3.2 Layer Operating System

This layer requires the Android operating system with the Kitkat version 4.4 and above, so an Android app can run smoothly on the phone. To develop Arduino algorithms and Android app, we need to use Arduino IDE software and Android Studio which is compatible with Windows 10, Mac OS X, and Linux.

3.3 Layer Software Dependencies

Arduino IDE v1.8.12, Android Studio v3.3.2 will be used to program Arduino and Android App. Firebase will be used as a cloud database for this project.

3.4 ARDUINO

Arduino is the main processor of the product. It gets data from temperature and humidity, light, display, irrigation, and fire extinguisher systems; then, control subsystems of those systems accordingly.

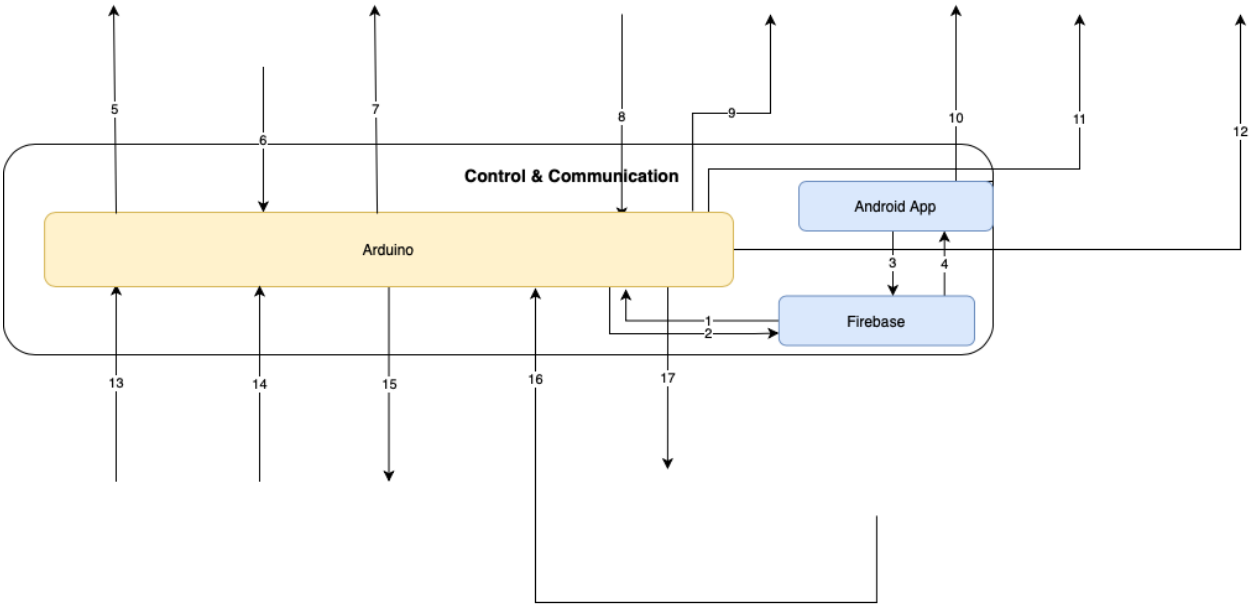

Figure 2: Arduino subsystem description diagram

3.4.1 SUBSYSTEM HARDWARE

Arduino Uno Wifi Rev2

3.4.2 SUBSYSTEM OPERATING SYSTEM

To download the necessary programs to control this subsystem, either the Windows 10, Mac OS, or Linux is required.

3.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

The Arduino IDE is necessary to program the Arduino Uno Rev2.0

3.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

To program using the Arduino IDE, both C and C++ programming languages are required.

3.4.5 SUBSYSTEM DATA STRUCTURES

Data Structure will be made up of JSON objects and JSON arrays.

3.4.6 SUBSYSTEM DATA PROCESSING

Data from other systems will be maintained by Arduino.

3.5 Android App

The Android App is the remote controller. It controls temperature and humidity, light, display, irrigation, and fire extinguisher systems of the Smart Green House by communicating with Firebase which is connected to Arduino.

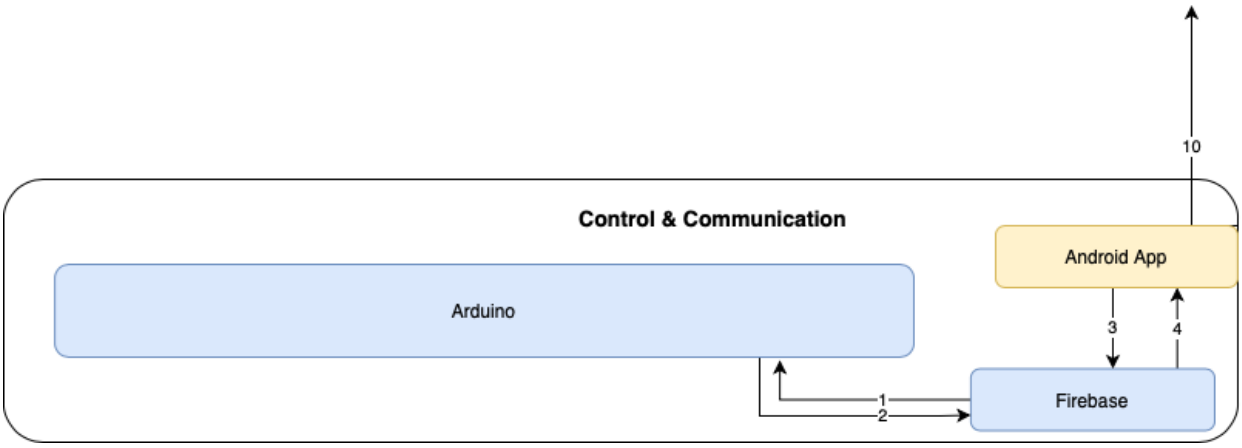

Figure 3: Android App subsystem description diagram

3.5.1 Subsystem Hardware

an Android Phone which is capable of running the android operating system 4.4 kitkat

3.5.2 Subsystem Operating System

To download the necessary programs to control this subsystem, either the Windows 10, Mac OS, or Linux is required.

3.5.3 Subsystem Software Dependencies

Android Studio v3.3.2

3.5.4 Subsystem Programming Languages

Java

3.5.5 Subsystem Data Structures

JSON objects and JSON arrays

3.5.6 Subsystem Data Processing

The android application will modify data in the Firebase database.

3.6 FIREBASE

This subsystem is the cloud database. It acts as a memory, stores moisture thresh-hold level, light level, desired temperature humidity, and other necessary data of the product.

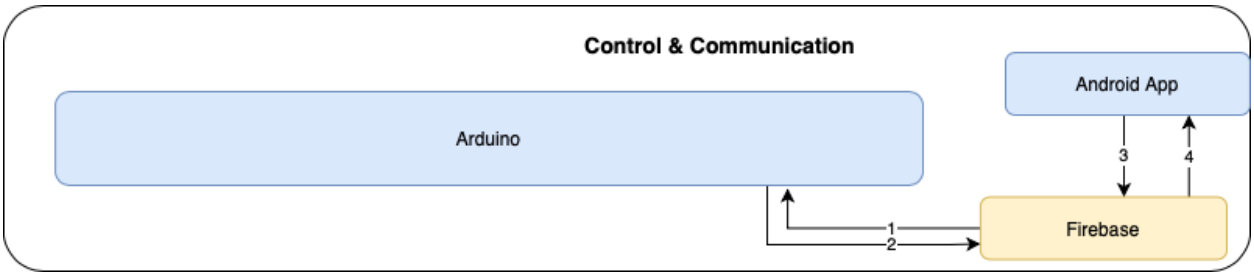

Figure 4: Firebase subsystem description diagram

3.6.1 SUBSYSTEM HARDWARE

Any hardware that can browse the internet can have an access to Firebase.

3.6.2 SUBSYSTEM OPERATING SYSTEM

Any OS

3.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

Firebase requires its URL and secret key to access the database.

3.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

N/A

3.6.5 SUBSYSTEM DATA STRUCTURES

JSON objects and JSON arrays

3.6.6 SUBSYSTEM DATA PROCESSING

Firebase database will be updated or replaced by Arduino and Android App signals.

4 Light Layer Subsystems

The light layer is responsible for receiving light intensity readings from the light sensor subsystem, sending that information to the control and communication layer, and receiving information from the control and communication layer to operate the light bulb subsystem.

4.1 Layer Hardware

The main hardware components for this layer consists of the light bulb and sensor. The sensor will pass light intensity readings to the Arudino Uno Rev 2.0 micro controller which would update a value in our database. Based on that light intensity value and other values received from other sensors, the micro controller will signal the light bulb to turn on or off.

4.2 Layer Operating System

The light bulb and sensor will be operated using the Arduino Uno Rev 2.0 micro controller. The micro controller will run on the Arduino operating system.

4.3 Layer Software Dependencies

Arduino software platform

4.4 Light Bulb

The Light bulb subsystem will receive operation instructions (ON/OFF) from the control and communication layer in order to provide the necessary heating for the green house.

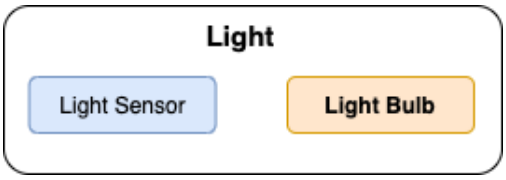

Figure 5: Light Bulb subsystem description diagram

4.4.1 Subsystem Hardware

The hardware component of this subsystem is primary the light heat bulb which will be used to provide the necessary heating requirements for the green house. The other component would be the micro controller which would given commands to turn on or turn off the bulb.

4.4.2 Subsystem Operating System

The necessary IDE (Integrated development environment) and programs required to operate this subsystem can be downloaded on a Windows, Mac OS or Linux system.

4.4.3 Subsystem Software Dependencies

The Arduino IDE must be used to program the micro controller.

4.4.4 Subsystem Programming Languages

To program using the Arduino IDE, C or C++ programming languages will be used.

4.4.5 Subsystem Data Structures

Boolean value to signal ON/OFF.

4.4.6 Subsystem Data Processing

Little data will be processed as such an algorithm will not be implemented.

4.5 Light Sensor

The light sensor subsystem is responsible for reading light intensity data present in the green house and will send the data to arduino micro controller for processing. The data from this sensors and others will be used to control the other component of the light layer; the light bulb.

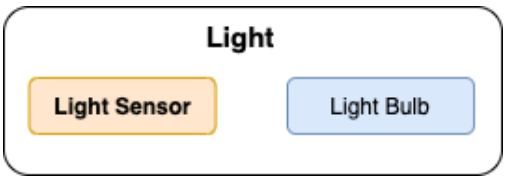

Figure 6: Light Sensor subsystem description diagram

4.5.1 Subsystem Hardware

The hardware component of this subsystem is primary the light sensor which will be used to provide light intensity readings observed in the green house environment. The other component would be the micro controller which would process the intensity data and make necessary database and hardware operations.

4.5.2 Subsystem Operating System

The necessary IDE (Integrated development environment) and programs required to operate this subsystem can be downloaded on a Windows, Mac OS or Linux system.

4.5.3 Subsystem Software Dependencies

The Arduino IDE must be used to program the micro controller.

4.5.4 Subsystem Programming Languages

To program using the Arduino IDE, C or C++ programming languages will be used.

4.5.5 Subsystem Data Structures

Light intensity data as an object.

4.5.6 Subsystem Data Processing

Little data will be processed as such an algorithm will not be implemented.

5 Temperature and Humidity Layer Subsystems

The temperature and humidity layer is responsible for monitoring the temperature and humidity levels inside and outside the greenhouse. The temperature readings taken by the sensors in this layer are used by the control and communication layer as input to determine when the cooling and heating systems should be turned on or off. Additionally, the cooling and heating systems in this layer can be manually turned on or off using the smart greenhouse application.

5.1 Layer Hardware

The hardware in the temperature and humidity layer consists of the temperature and humidity sensor, a heat lamp, fans and the pump. The sensor sends temperature readings to the Arduino Uno Wifi Rev 2 which sends the recorded data to a database on Firebase. Based on the values taken by the temperature sensor, the Arduino prompts the heating and cooling systems to turn on or off.

5.2 Layer Operating System

This layer utilizes the Android operating system Kitkat version 4.4 and above as the operating system for the application. Additionally, either Windows 10, Mac OS or Linux are required for the Arduino Uno Wifi Rev 2.

5.3 Layer Software Dependencies

The temperature and humidity layer is controlled by the Arduino Uno Wifi Rev 2 and uses the Arduino IDE. Android Studio is utilized for the purpose of developing an application which runs on Android OS 4.4 Kitkat and above.

5.4 Cooling and Dehumidifying System

The cooling and dehumidifying system is prompted to turn on by the control and communication layer when the temperature and humidity reading goes above the threshold set by the user. The system is made up of hardware components which work to lower the temperature and humidity levels inside the greenhouse.

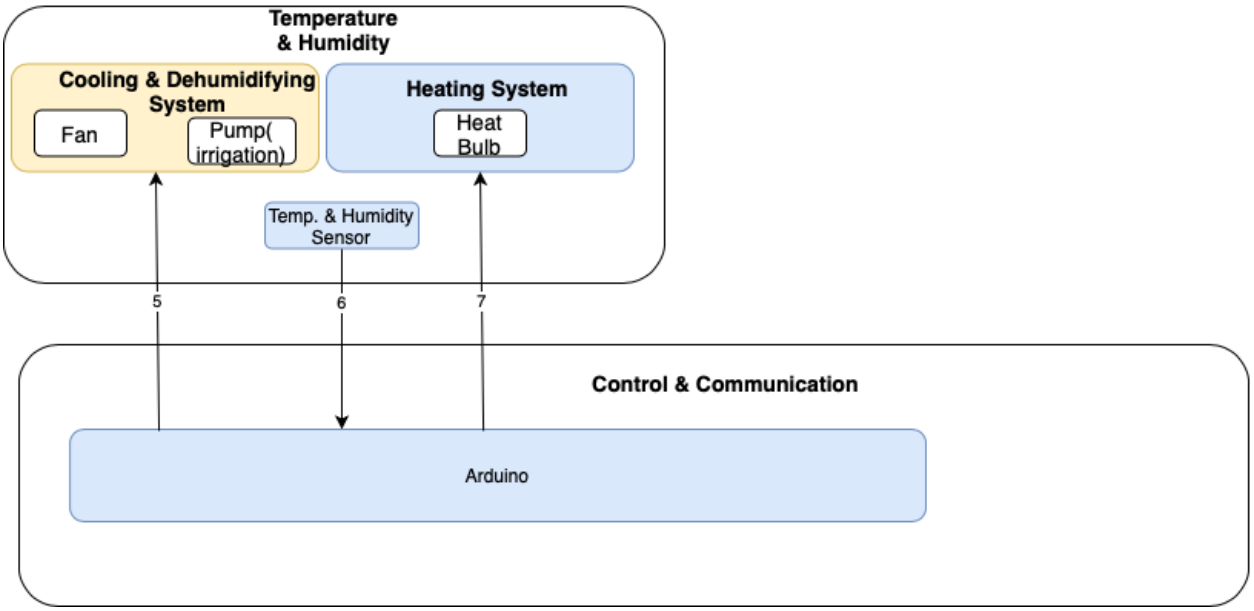

Figure 7: Cooling and Dehumidifying subsystem description diagram

5.4.1 Subsystem Hardware

The cooling and dehumidifying system is made up of a set of 120mm axial fans and a 12 V submersible water pump, which is also used by the irrigation layer system. The Arduino Uno Wifi Rev 2, sends the signal to turn on and off the fans and the water pump.

5.4.2 Subsystem Operating System

Windows 10, Mac OS, or Linux are required for controlling Arduino Uno Wifi Rev 2 and the Android OS Kitkat 4.0 is required for the mobile application.

5.4.3 Subsystem Software Dependencies

This subsystem requires Arduino IDE and Android Studio for the purpose of programming the automation of the cooling and dehumidifying system as well as the application to control the subsystem remotely.

5.4.4 Subsystem Programming Languages

Programming in Arduino IDE uses C and C++ programming languages. Programming in Android Studio uses Java.

5.4.5 Subsystem Data Structures

Boolean value that signals ON/OFF.

5.4.6 Subsystem Data Processing

Data will be processed by the Arduino and stored in a database in Firebase.

5.5 Heating System

The heating system is made up of hardware components responsible for warming up the greenhouse when prompted by the control and communication layer due to the temperature inside the greenhouse falling below the temperature threshold set by the user.

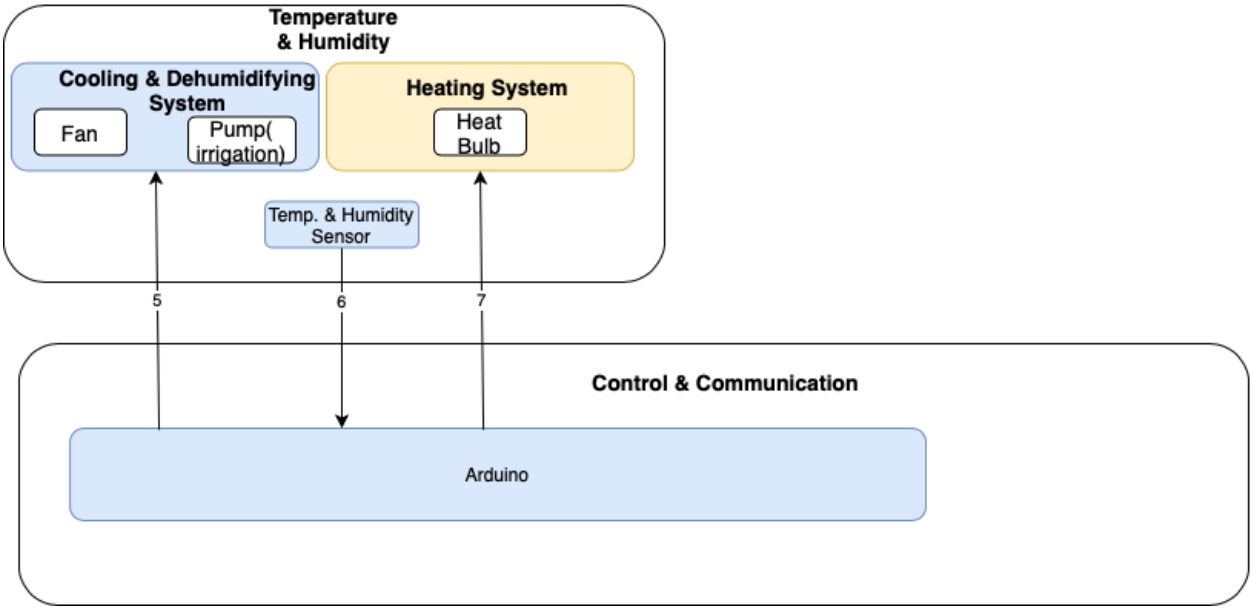

Figure 8: Heating System subsystem description diagram

5.5.1 Subsystem Hardware

Heating system is made up of a single 120 V incandescent red heat lamp and Arduino Uno Wifi Rev 2 which prompts the lamp to turn on and off.

5.5.2 Subsystem Operating System

Windows 10, Mac OS, or Linux are required for controlling Arduino Uno Wifi Rev 2 and the Android OS Kitkat 4.0 is required for the mobile application.

5.5.3 Subsystem Software Dependencies

This subsystem requires Arduino IDE and Android Studio for the purpose of programming the automation of the heating system as well as the application to control the subsystem remotely.

5.5.4 Subsystem Programming Languages

Programming in Arduino IDE uses C and C++ programming languages. Programming in Android Studio uses Java.

5.5.5 Subsystem Data Structures

Boolean value that signals ON/OFF.

5.5.6 Subsystem Data Processing

Data will be processed by the Arduino and stored in a database in Firebase.

5.6 Temperature and Humidity Sensor

The temperature and humidity sensor has the responsibility of continuously monitoring the temperature and humidity levels inside and outside the greenhouse. Based on the temperature and humidity levels read by the sensor, the control and communication layer prompts heating, cooling and dehumidifying subsystems to turn on and off.

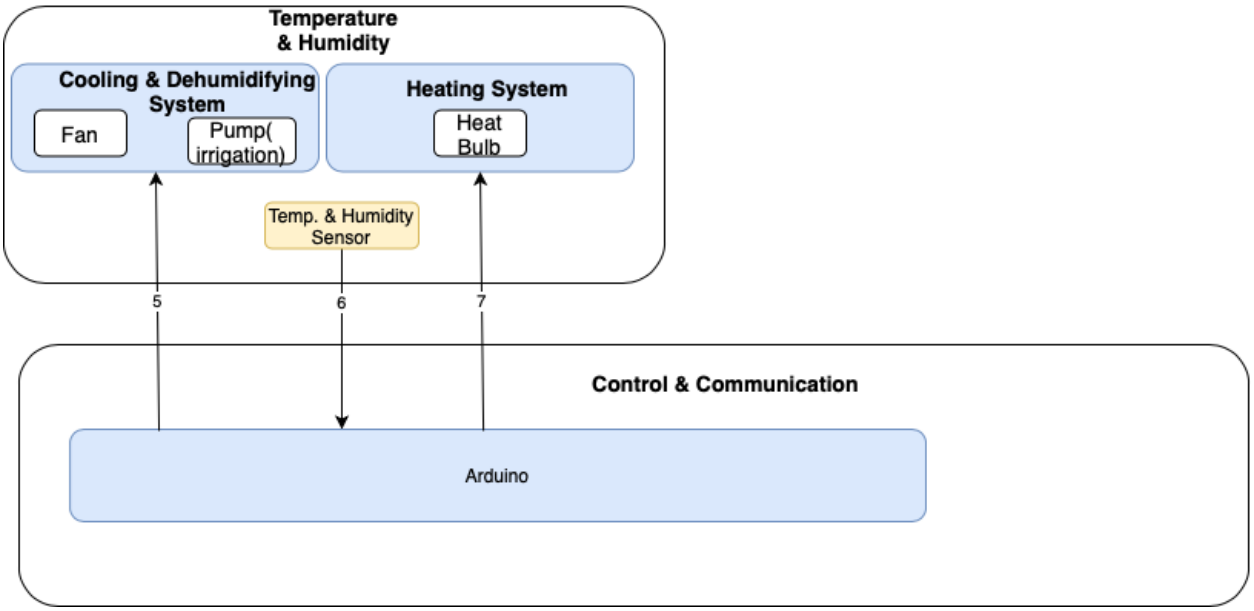

Figure 9: Temperature and Humidity Sensor subsystem description diagram

5.6.1 Subsystem Hardware

This subsystem is made up of a couple DHT22 temperature and humidity sensors and Arduino Uno Wifi Rev 2 which processes the data and prompts other subsystems based on the sensor input.

5.6.2 Subsystem Operating System

Windows 10, Mac OS, or Linux is required for controlling Arduino Uno Wifi Rev 2.

5.6.3 Subsystem Software Dependencies

Arduino IDE is needed for the purpose of reading and processing temperature and humidity data read by the sensor.

5.6.4 Subsystem Programming Languages

Arduino IDE uses C and C++ programming languages.

5.6.5 Subsystem Data Structures

Temperature and humidity data as an object.

5.6.6 Subsystem Data Processing

Data will be processed by the Arduino and stored in a database in Firebase.

6 Display Layer Subsystems

The display layer subsystem is responsible for outputting information about the state of the greenhouse to the user. Ideally this information is to be presented in an way such that the user can find the information they are looking for within 10 seconds of reviewing any subsystem in the display layer. There are three different subsystems that make up the display layer subsystem. The first two are the LCD display and LED error signal which are physical systems attached to the greenhouse, and the third is the Earthwise android application which is a software based system.

6.1 Layer Hardware

The main hardware components of the system are the LCD Display and the LED Error Signal. Both of the are controlled by and Arduino Uno Rev2.0 and get updated based on data collected from the other sensors connected to the greenhouse. The Arduino will periodically read measurements for the other systems and update the LCD display such that the current readings are displayed to the user directly. The main purpose of the LCD display is to only show the information about the greenhouses temperature, humidity, moisture levels, and the type of plant being grown.
The LED error signal is a small hardware component that also controlled by the Arduino Uno Rev2.0. This component alerts the user that there is a problem with the greenhouse that needs to be addressed by being turning the LED on such that the user can see there is an issue. The LED signal is dependent on the Arduino's analysis of the sensors attached to the greenhouse.
The last hardware this project is a phone capable of running the android operating system 4.4 kitkat or above. An application installed on the phone will allow the user to interact with the greenhouse and view current readings of the greenhouse. Essentially the phone is a wireless equivalent of the LCD display with some extra features.

6.2 Layer Operating System

The Android operating system Kitkat version 4.4 and above, will be the main operating system used to control the system. To develop on for this system both windows 10 and Mac OS will be utilized.

6.3 Layer Software Dependencies

The greenhouse system is primarily controlled by the Arduino Uno Rev2.0 that is connected to the greenhouse and all the sensors. This makes it necessary to utilize the Arduino IDE to program the Arduino to manage these systems. The Arduino code created in the IDE will communicate with all the sensors on the greenhouse as well as our Firebase database.
Using android studio an application is developed that can run on Android OS 4.4 Kitkat and above. This application will use the information from the Firebase database to display relevant information on the greenhouse. It will also allow the user to manually change the data of the greenhouse.

6.4 LCD Display

The LCD display receives a signal from the Arduino Uno Rev2.0 to display information about the current state of the greenhouse.

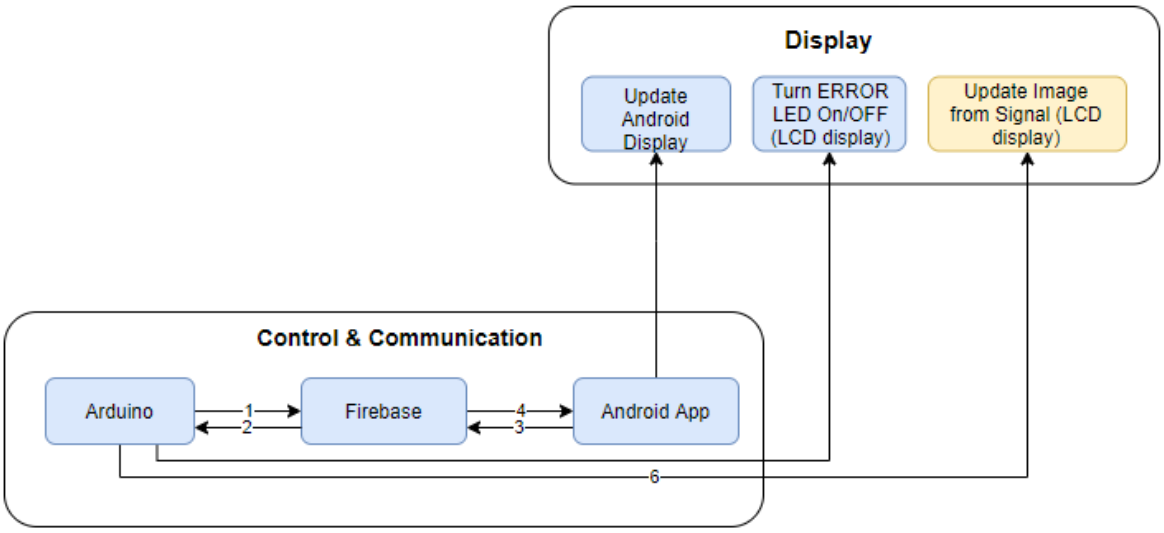

Figure 10: LCD Display subsystem description diagram

6.4.1 Subsystem Hardware

The LCD display has who Hardware subsystems. The first is LCD display mounted to the greenhouse that can display the greenhouse information. The second is the Arduino Uno Rev2.0 which sends the signal to the LCD display such that the LCD displays the correct information.

6.4.2 Subsystem Operating System

To download the necessary programs to control this subsystem, ether the Windows 10, Mac OS, or Linux is required.

6.4.3 Subsystem Software Dependencies

The Arduino IDE is necessary to program the Arduino Uno Rev2.0

6.4.4 Subsystem Programming Languages

To program using the Arduino IDE, both C and C++ programming languages are required.

6.4.5 Subsystem Data Structures

Data Structure will be made up of objects.

6.4.6 Subsystem Data Processing

A small amount of data will be processed by the Arduino.

6.5 LED Error Signal

This is simply a Red LED that is mounted to the greenhouse. In the event of an error, the LED will be activated to alert the user that there is an issue with the greenhouse.

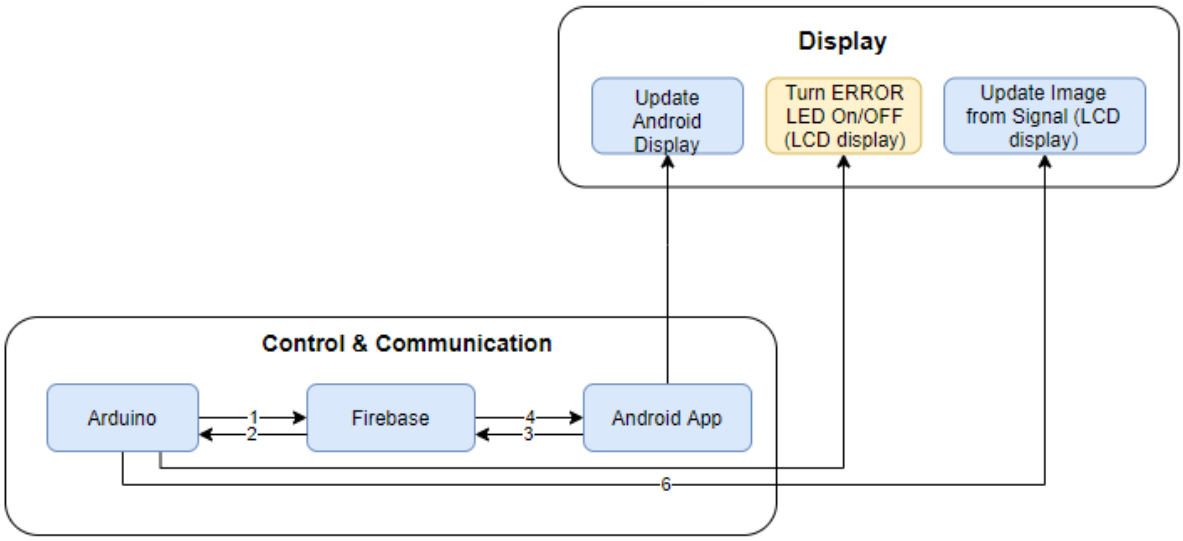

Figure 11: Led Error subsystem description diagram

6.5.1 Subsystem Hardware

There are two hardware components to allow this to work, the LED attached to the greenhouse and the Arduino Uno Rev2.0 to update the signal of an issue.

6.5.2 Subsystem Operating System

To download the necessary programs to control this subsystem, ether the Windows 10, Mac OS, or Linux is required.

6.5.3 Subsystem Software Dependencies

The Arduino IDE is necessary to program the Arduino Uno Rev2.0

6.5.4 Subsystem Programming Languages

To program using the Arduino IDE, both C and C++ programming languages are required.

6.5.5 Subsystem Data Structures

The LED Signal will be an Boolean object.

6.5.6 Subsystem Data Processing

This subsystem does not do any data processing.

6.6 Earthwise Android Application

This subsystem is a remote display of the greenhouses state through the use of an android application developed for android 4.4 Kitkat and above. This subsystem allows the user to view the greenhouse data remotely as well as update any settings as necessary.

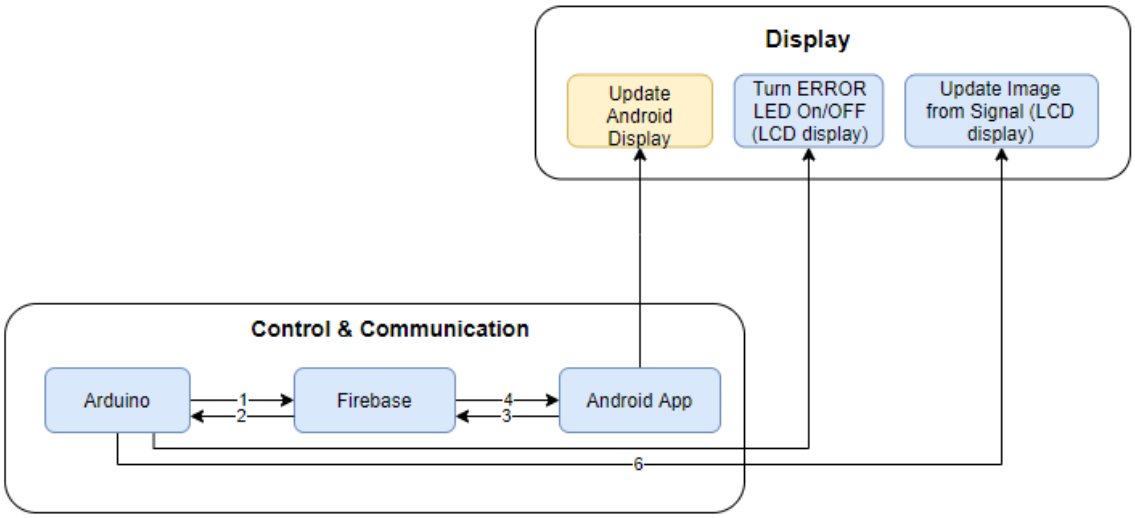

Figure 12: Earthwise Android description diagram

6.6.1 Subsystem Hardware

A phone capable of running the android application.

6.6.2 Subsystem Operating System

Any Android operating system that is Android 4.4 Kitkat or above.

6.6.3 Subsystem Software Dependencies

The application development is dependent on the use of Android Studio to compile and build the application. To communcate between the Arduino Uno Rev2.0 and the application, Google's Firebase database software will be used.

6.6.4 Subsystem Programming Languages

Android studio utilizes Java, Kotlin, and XML to create the functionality of the application.

6.6.5 Subsystem Data Structures

The data structure will most likely be a graph data structure and the array data structure.

6.6.6 Subsystem Data Processing

The android application will modify data in the Firebase database.

7 Irrigation Layer Subsystems

The Irrigation Subsystem is responsible to water the plants whenever the soil moisture goes down. This system is basically composed of multiple hardware components like pump, moisture sensors, water level sensors etc. Some of them provide data to arduino while some are controlled by arduino based on the data it receives.

7.1 Layer Hardware

The main hardware components for this layer are pump, moisture sensor and water level sensor. The moisture sensor and water level sensor will take readings and pass those readings to arduino. These readings are taken periodically i.e like every 5 minutes and the micro-controller will control the pump based on these readings. If the moisture reading is low, the arduino will turn the pump on and once enough moisture level is achieved, the pump is turned off again. Similarly, low reading from water level sensor means arduino will not turn the pump on but rather notify user through communication layer.

7.2 Layer Operating System

The layer is operated by Arduino Uno REV2 WIFI. Arduino uses Arduino programming language and Arduino Software IDE.

7.3 Layer Software Dependencies

The required libraries and frameworks are automatically imported by the Arduino IDE.

7.4 Moisture Sensor

Moisture sensor is a hardware component that measures the water content of soil.The Soil Moisture Sensor uses capacitance to measure dielectric permittivity of the surrounding medium. In soil, dielectric permittivity is a function of the water content. The sensor creates a voltage proportional to the dielectric permittivity, and therefore the water content of the soil.

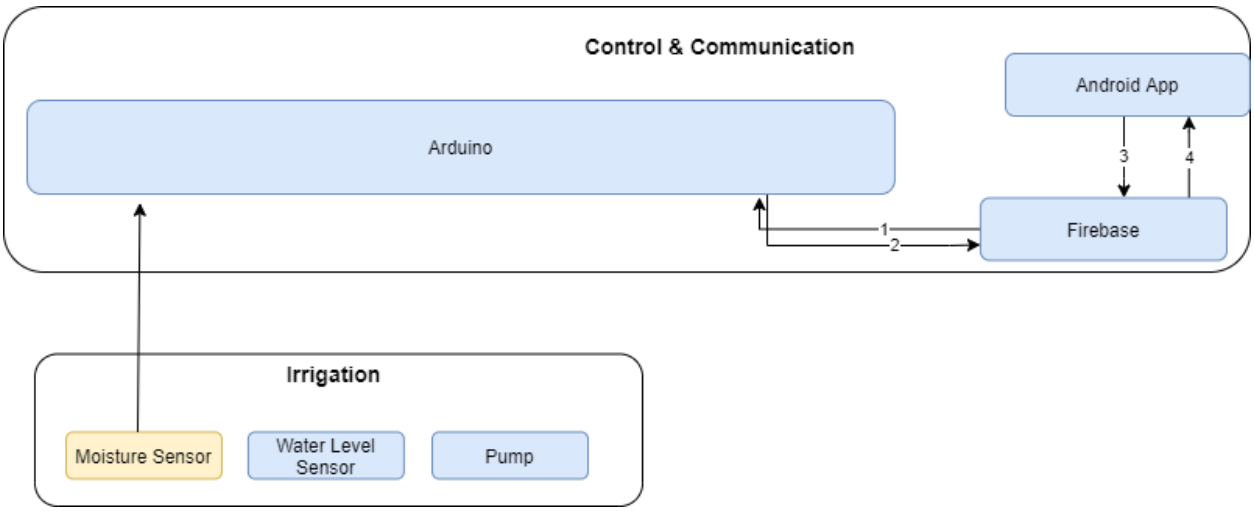

Figure 13: Moisture Sensor subsystem description diagram

7.4.1 Subsystem Hardware

The hardware involved in this system are moisture sensor itself and Arduino UNO REV 2 WIFI. The moisture sensor will send the readings to arduino and the arduino will process those readings.

7.4.2 Subsystem Operating System

The subsystem is operated by Arduino UNO REV 2 WIFI which is programmed using Arduino IDE that can be installed in WINDOWS, LINUX or MAC OS.

7.4.3 Subsystem Software Dependencies

All the libraries required are automatically imported during the installation of Arduino IDE.

7.4.4 Subsystem Programming Languages

The Arduino UNO REV 2 is programmed using Arduino programming language.

7.4.5 Subsystem Data Structures

The readings from the sensor are passed to arduino in the form of electric pulse and these readings are sent from arduino as a packets of serial data.

7.4.6 Subsystem Data Processing

The readings from sensor are processed by Arduino UNO REV 2 WIFI.

7.5 Water Level Sensor

The water level sensor is basically a complex circuit that measures the volume of the water in the tank. The circuit consists of copper wires and LEDs that enables us to determine volume of water in the tank.

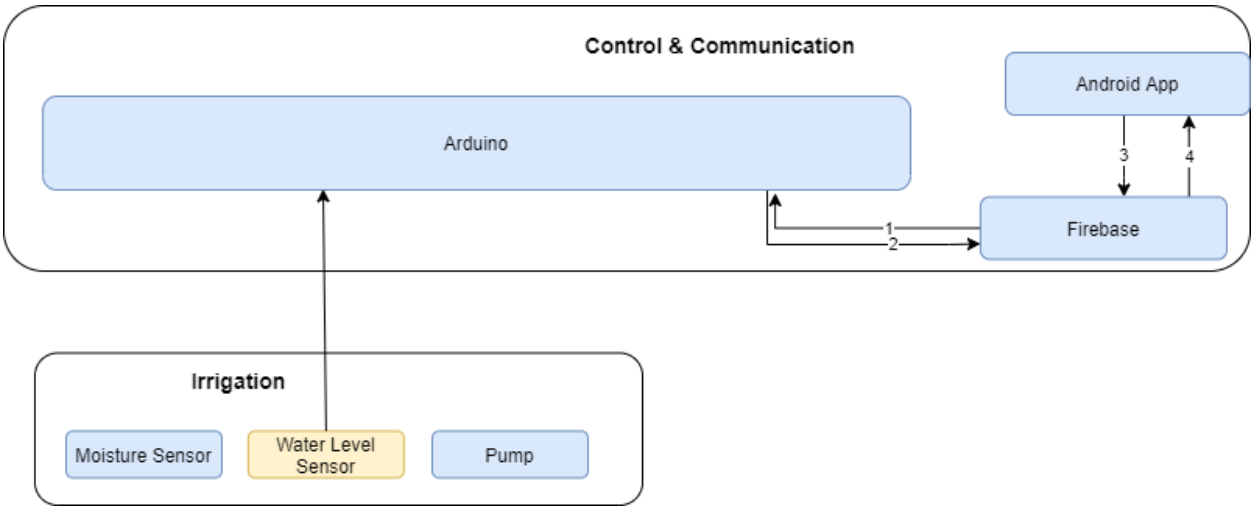

Figure 14: Water Level Sensor subsystem description diagram

7.5.1 Subsystem Hardware

In this hardware setup, copper wires are placed at heights with know quantity of water. The output transistor collector side is input to arduino. Three arduino pins are declared as inputs. Now when charged water touches the copper wire the particular led glows up and arduino reads it as a high input. Once arduino finds the particular pin status as high it prints the quantity of water on its serial monitor.

7.5.2 Subsystem Operating System

The subsystem is operated by Arduino UNO REV 2 WIFI which is programmed using Arduino IDE that can be installed in WINDOWS, LINUX or MAC OS.

7.5.3 Subsystem Software Dependencies

All the libraries required are automatically imported during the installation of Arduino IDE.

7.5.4 Subsystem Programming Languages

The Arduino UNO REV 2 is programmed using Arduino programming language.

7.5.5 Subsystem Data Structures

The readings from the water level sensor are passed to arduino in the form of electric voltage that reads high or low and these readings are sent from arduino as a packets of serial data.

7.5.6 Subsystem Data Processing

The readings from sensor are processed by Arduino UNO REV 2 WIFI.

7.6 PUMP

The subsystem 'Pump' is a hardware component that is responsible for irrigating the plants. The pump is connected to other physical components like vinyl tube that goes to water tank and sprinkler system and also arduino that controls the pump.

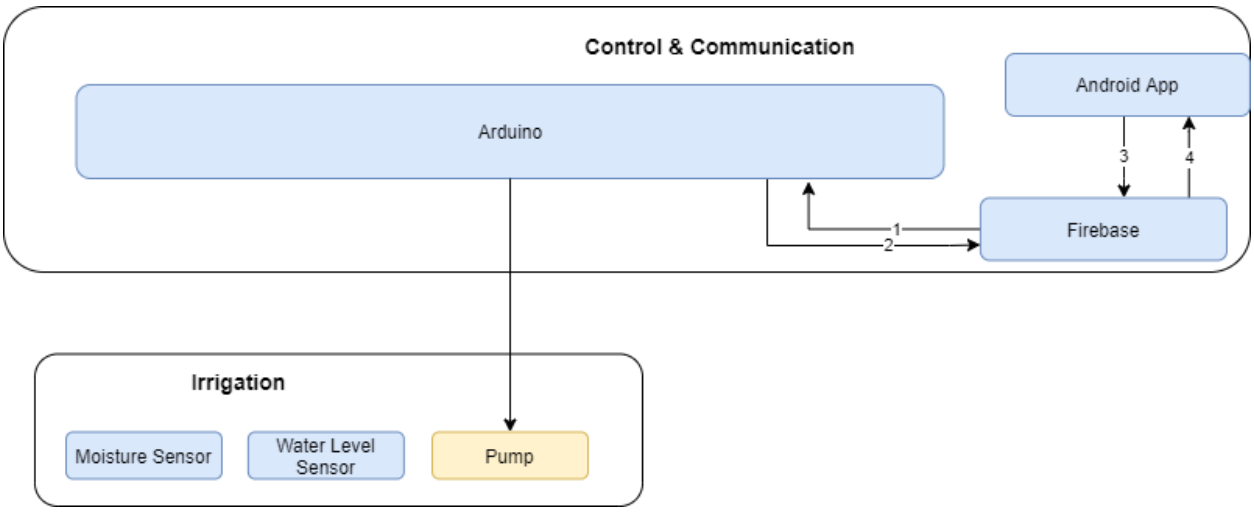

Figure 15: Pump subsystem description diagram

7.6.1 SUBSYSTEM HARDWARE

The main hardware component of the layer is pump. The pump is powered using 12 V adapter using relay. Two tubes are connected to pump. One of them is connected to water tank and other one is connected to sprinkler system. The Arduino controls the switch to the pump and turns it on or off as per necessity.

7.6.2 SUBSYSTEM OPERATING SYSTEM

The subsystem is operated by Arduino UNO REV 2 WIFI which is programmed using Arduino IDE that can be installed in WINDOWS, LINUX or MAC OS.

7.6.3 SUBSYSTEM SOFTWARE DEPENDENCIES

All the libraries required are automatically imported during the installation of Arduino IDE.

7.6.4 SUBSYSTEM PROGRAMMING LANGUAGES

The Arduino UNO REV 2 is programmed using Arduino programming language.

7.6.5 SUBSYSTEM DATA STRUCTURES

Arduino will control the pump using Boolean object. The pump is either on or off at all times.

7.6.6 SUBSYSTEM DATA PROCESSING

There is no data processing in this layer.

8 Fire Extinguisher Layer Subsystems

This subsystem is responsible for turning on the sprinkler system if the case of fire is detected. The subsystem is composed of some hardware components and arduino that will provide working mechanism to these components.

8.1 Layer Hardware

The main hardware components of this layer are pump and smoke sensor. The smoke sensor provides readings to arduino every minute. Based on the readings, the arduino is calibrated to determine if there is case of fire or not. If risk of fire is determined, the arduino turns the pump on.

8.2 Layer Operating System

The layer is operated by Arduino Uno REV2 WIFI. Arduino uses Arduino programming language and Arduino Software IDE

8.3 Layer Software Dependencies

The required libraries and frameworks are automatically imported by the Arduino IDE.

8.4 PUMP

The subsystem 'Pump' is a hardware component that is responsible for eliminating fire hazard. The pump is connected to other physical components like vinyl tube that goes to water tank and sprinkler system and also the arduino that controls the pump.

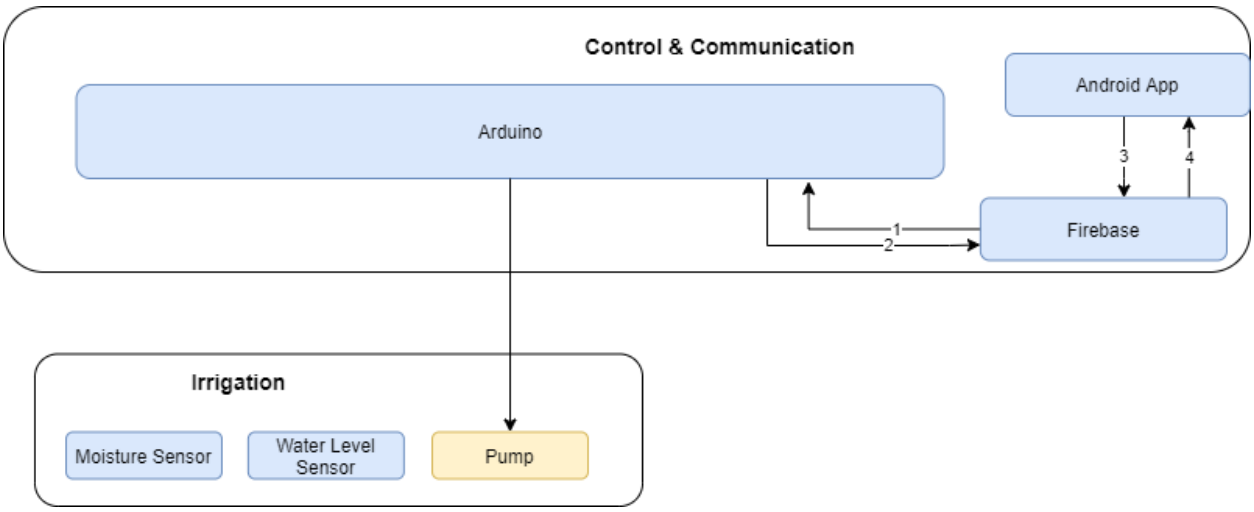

Figure 16: Pump subsystem description diagram

8.4.1 SUBSYSTEM HARDWARE

The main hardware component of the layer is pump. The pump is powered using 12 V adapter using relay. Two tubes are connected to pump. One of them is connected to water tank and other one is connected to sprinkler system. The Arduino controls the switch to the pump and turns it on or off as per necessity.

8.4.2 SUBSYSTEM OPERATING SYSTEM

The subsystem is operated by Arduino UNO REV 2 WIFI which is programmed using Arduino IDE that can be installed in WINDOWS, LINUX or MAC OS.

8.4.3 SUBSYSTEM SOFTWARE DEPENDENCIES

All the libraries required are automatically imported during the installation of Arduino IDE.

8.4.4 SUBSYSTEM PROGRAMMING LANGUAGES

The Arduino UNO REV 2 is programmed using Arduino programming language.

8.4.5 SUBSYSTEM DATA STRUCTURES

Arduino will control the pump using Boolean object. The pump is either on or off at all times.

8.4.6 SUBSYSTEM DATA PROCESSING

There is no data processing in this layer.

8.5 Smoke Sensor

The smoke sensor is Metal Oxide Semiconductor (MOS) type Gas Sensor also known as Chemiresistors as the detection is based upon change of resistance of the sensing material when the Gas comes in contact with the material. Using a simple voltage divider network, concentrations of gas can be detected.

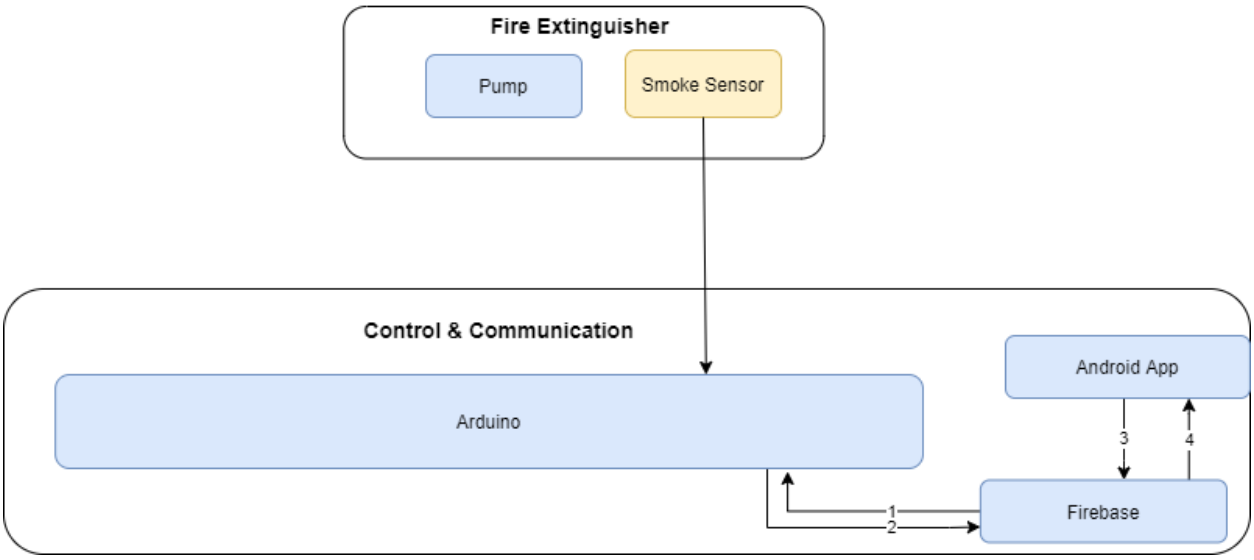

Figure 17: Smoke Sensor subsystem description diagram

8.5.1 Subsystem Hardware

The sensor is actually enclosed in two layers of fine stainless steel mesh called Anti-explosion network. It ensures that heater element inside the sensor will not cause an explosion, as we are sensing flammable gases. When tin dioxide (semiconductor particles) is heated in air at high temperature, oxygen is adsorbed on the surface. In clean air, donor electrons in tin dioxide are attracted toward oxygen which is adsorbed on the surface of the sensing material. This prevents electric current flow.

In the presence of reducing gases, the surface density of adsorbed oxygen decreases as it reacts with the reducing gases. Electrons are then released into the tin dioxide, allowing current to flow freely through the sensor.

8.5.2 Subsystem Operating System

The subsystem is operated by Arduino UNO REV 2 WIFI which is programmed using Arduino IDE that can be installed in WINDOWS, LINUX or MAC OS.

8.5.3 Subsystem Software Dependencies

All the libraries required are automatically imported during the installation of Arduino IDE.

8.5.4 Subsystem Programming Languages

The Arduino UNO REV 2 is programmed using Arduino programming language.

8.5.5 Subsystem Data Structures

The readings from the smoke sensor are passed to arduino in the form of electric pulses and these readings are sent from arduino as a packets of serial data.

8.5.6 Subsystem Data Processing

The readings from sensor are processed by Arduino UNO REV 2 WIFI.

9 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES