



IOT BASED HOME AUTOMATION SOLUTION

PROJECT GUIDE : JAYALAXMI N DHANYAL

SUBMITTED BY : ANIMALLA ALEKHYA
ELECTRONICS AND COMMUNICATION ENGINEERING
G. NARAYANAMMA INSTITUTE OF TECHNOLOGY AND SCIENCE



CONTENTS

- ❑ C/C++ LANGUAGES
- ❑ OOPS CONCEPT
- ❑ WHAT IS IoT?
- ❑ WHAT IS EMBEDDED SYSTEM?
- ❑ ABOUT THE PROJECT
- ❑ REQUIREMENTS
- ❑ ARCHITECTURE
- ❑ IMPLEMENTATION
- ❑ RESULTS
- ❑ CONCLUSION

TOPICS I LEARNT IN THIS INTERNSHIP

➤ **C Programming**

Keywords, Basic Datatypes, Operators, Conditional constructs and Looping concepts, Overflow and Underflow, Arrays and Pointers, Functions, Storage Classes, Pre-Processor Directives and Header files.

➤ **C++ Programming**

Oops Concepts – Encapsulation, Inheritance, Data Abstraction, Polymorphism.

➤ **Internet of Things (IoT)**

Definition of IoT and Architecture of IoT.

➤ **Embedded Systems**

Definition of Embedded Systems, Categories of Embedded Systems and Components of Embedded Systems.

➤ **Peripherals used in Project**

LED's, CLCD using I2C communication, LM35 (temperature sensor), LDR sensor, Ethernet shield, Temperature System and Serial Tank (Serial Communication).



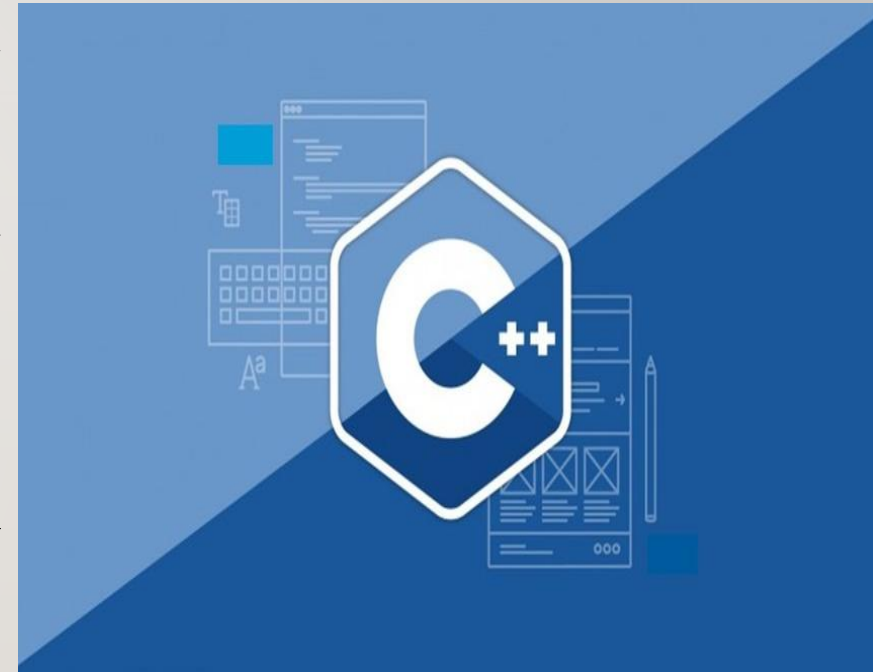
C PROGRAMMING

- C programming is a powerful and versatile language commonly used for system and application development.
- It offers low-level memory access, making it ideal for developing operating systems, embedded systems, and high-performance applications.
- C provides a rich set of operators and built-in functions, allowing efficient code execution and manipulation of data.
- Its simplicity, portability, and widespread use have made C a foundational language in computer science and software development.



C++ PROGRAMMING

- C++ is an extension of the C programming language that adds object-oriented features, making it well-suited for developing complex software systems.
- It supports classes, inheritance, and polymorphism, enabling code reuse and modularity.
- C++ also provides low-level memory manipulation like C, allowing developers to optimize performance in resource-constrained environments.
- Its versatility and extensive standard library make C++ a popular choice for game development, real-time systems, and high-performance applications.



OOPS CONCEPT

Object-Oriented Programming (OOP) is a programming paradigm centered around the concept of objects, which are instances of classes that encapsulate data and behavior.

The four key principles of Oops are:

1. **Encapsulation**
2. **Inheritance**
3. **Polymorphism**
4. **Abstraction**



WHAT IS IoT?

IoT stands for Internet of Things. IoT is the network of physical objects that contain embedded technology to communicate and interact with their internal states/ external environments.

Applications of IoT:

- Architecture
- Factory
- Automobiles
- Transport Utilities
- Insurance
- Consumers, etc



WHAT IS EMBEDDED SYSTEM?

Embedded System is any combination of hardware and software which is intended to do specific task.

Three main components of Embedded systems are:

1. Hardware
2. Software
3. Firmware

Categories of Embedded System :

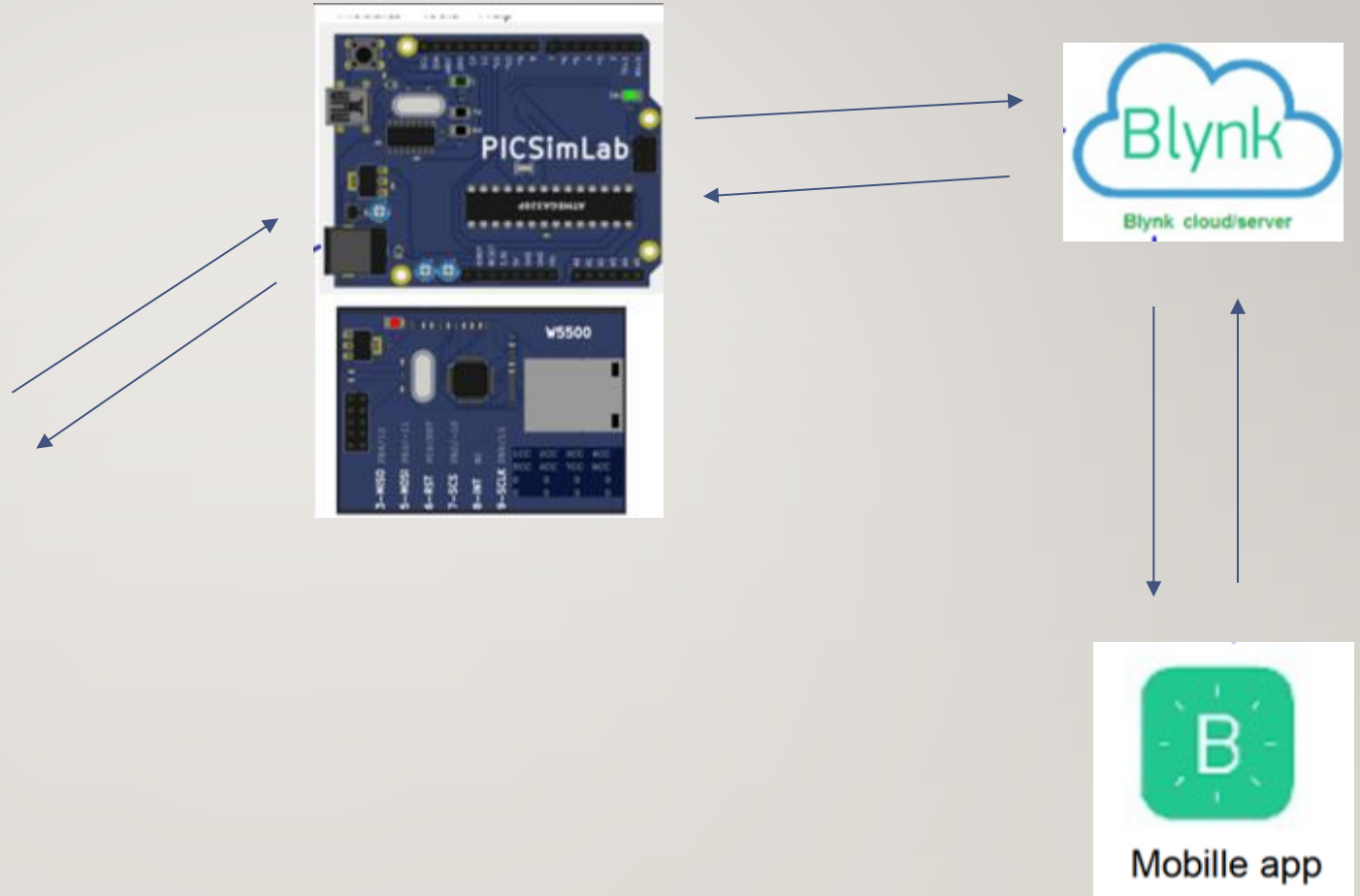
1. Stand Alone Embedded System
2. Real time Embedded System
3. Networked Embedded System
4. Mobile Embedded System



ABOUT THE PROJECT – HOME AUTOMATION

Main Components:

- ❑ Arduino Uno
- ❑ PicSimLab
- ❑ Blynk App



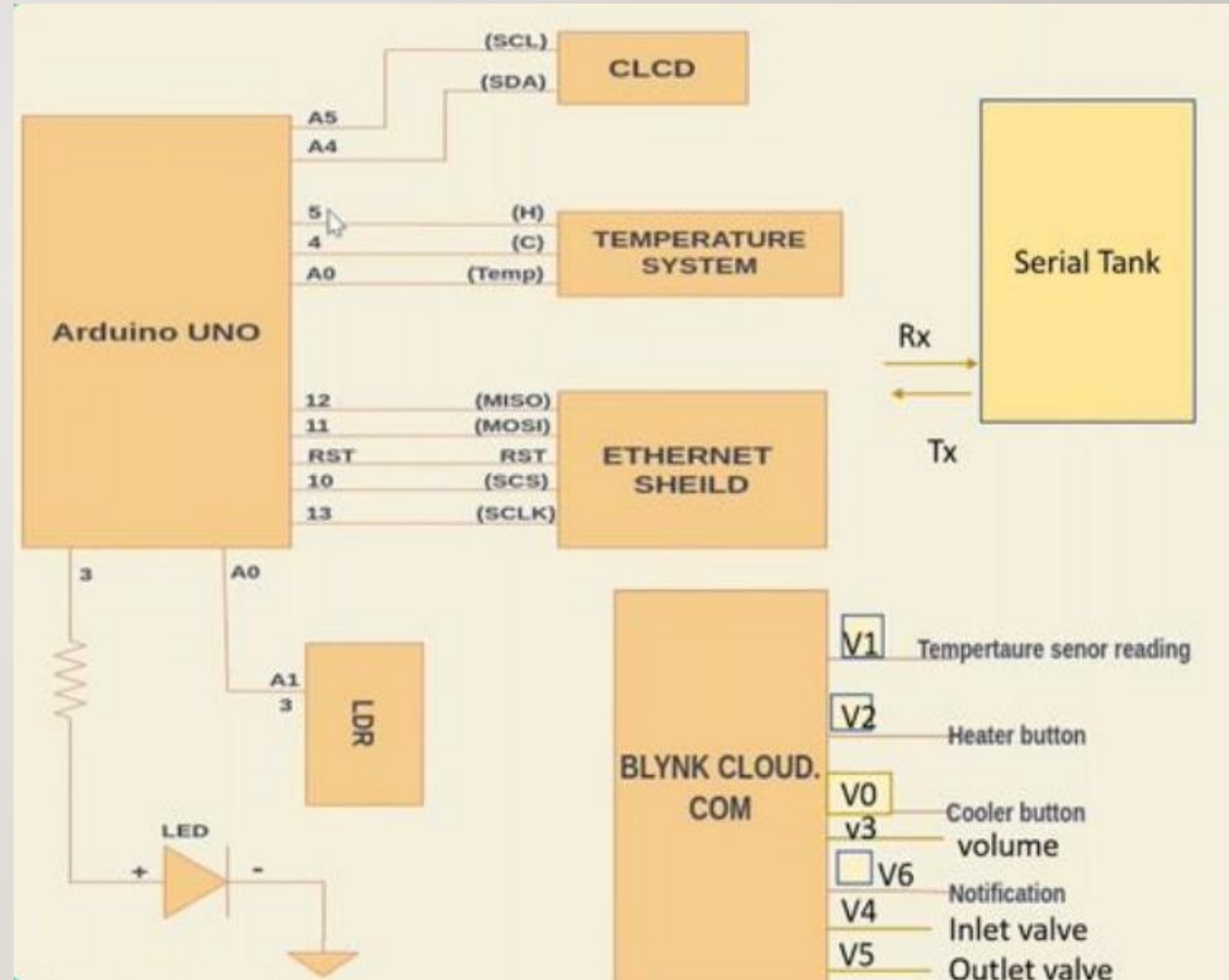
REQUIREMENTS

- ✓ Garden Light Control
- ✓ Temperature Control System
 - ☐ Cooler Control System
 - ☐ Heater Control System
 - ☐ Threshold Temperature Control
- ✓ Water tank inlet and outlet valve control
 - ☐ Inlet Valve Control
 - ☐ Outlet Valve Control
 - ☐ Control the volume of water in the tank

ARCHITECHTURE

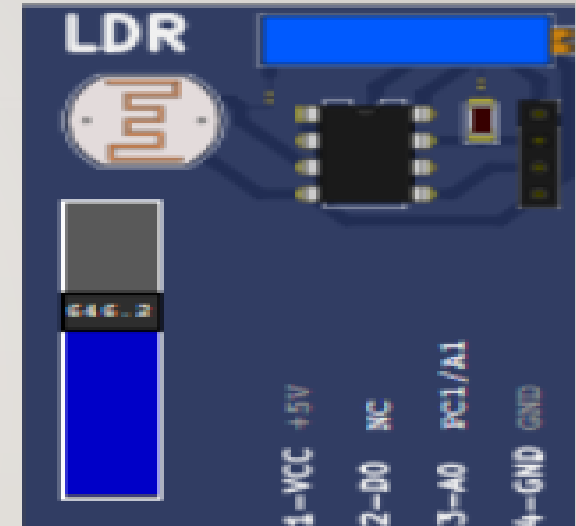
Peripherals

- Arduino UNO
- LED's
- LDR sensor
- CLCD using I2C communication
- Ethernet shield
- Temperature System
- Serial Tank (Serial Communication)
- Blynk Cloud.com



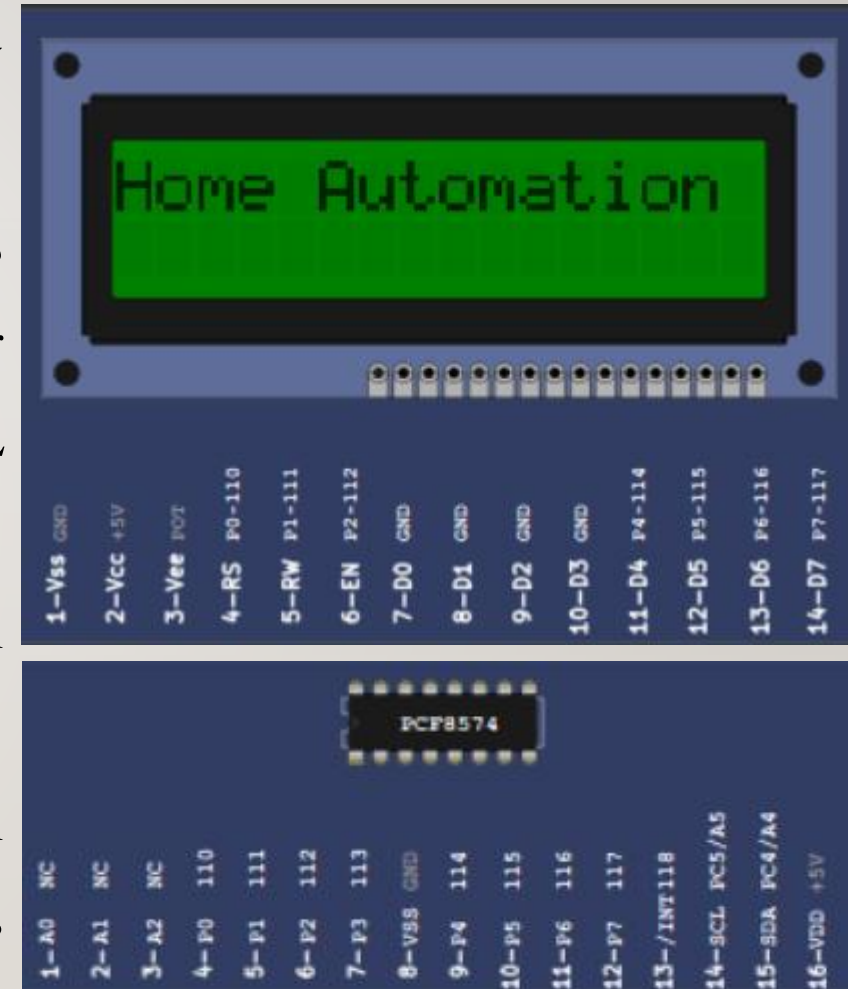
Peripherals – LDR Sensor and LEDs

- LDR (Light dependent resistor) SENSOR
- An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.
- The resistance of a Photo resistor decreases with increasing incident light intensity. In other words, it exhibits photoconductivity.
- LED (Light Emitting Diode) is a semiconductor device that emits light when an electric current passes through it
- For an LED, a logic high typically turns it on, and a logic low turns it off, depending on the circuit configuration.



Peripherals – CLCD using I2C communication

- A Character LCD (CLCD) using I2C communication is a type of display that interfaces with a microcontroller via the I2C bus.
- The PCF8574 is an I2C I/O expander integrated circuit. It provides 8 digital input/output pins and communicates with microcontrollers over the I2C bus using only two wires: SDA (Serial Data Line) and SCL (Serial Clock Line).
- This reduces the number of pins needed compared to traditional parallel LCD connections.
- The I2C module on the CLCD typically includes an I2C-to-parallel converter, allowing for easy communication and control of the LCD's display functions like writing characters and setting the cursor position.



Peripherals – Ethernet shield (ETH W5500)

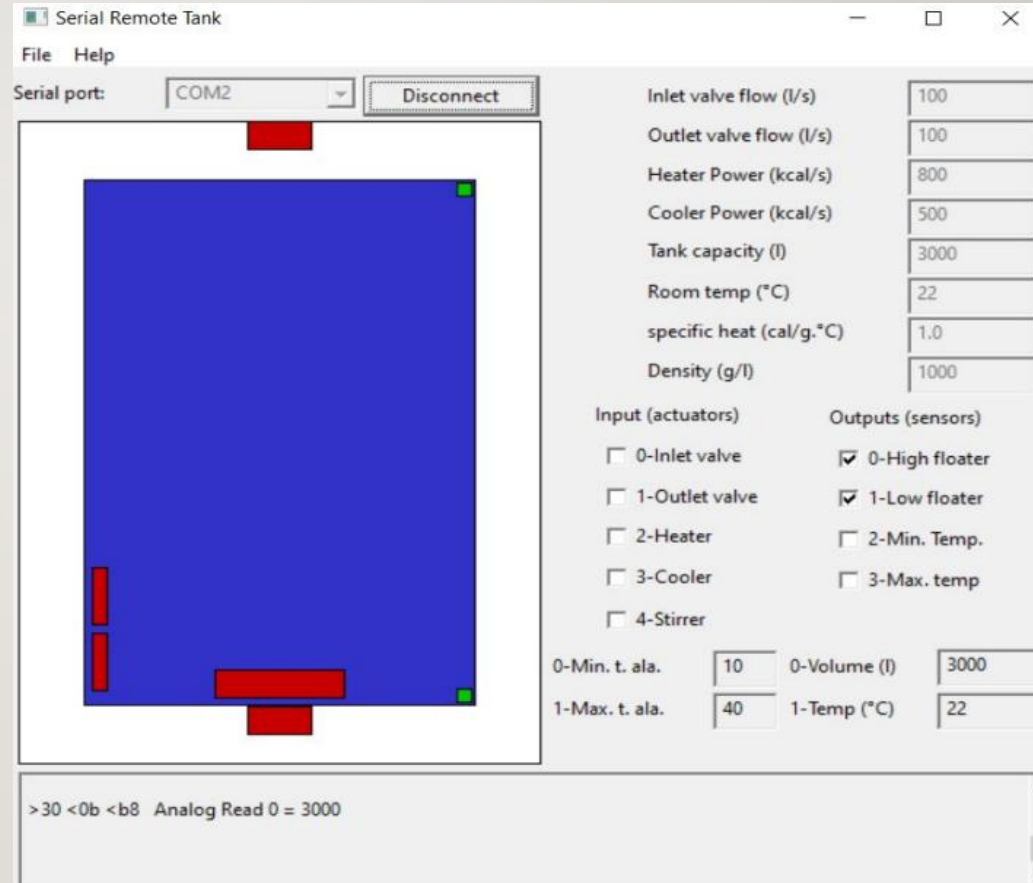
- The W5500 chip is a Hardwired TCP/IP embedded Ethernet controller that provides easier Internet connection to the embedded system.
- W5500 enables users to have the Internet connectivity in their applications just by using the single chip in which TCP/IP stack, 10/100 Ethernet MAC and PHY embedded.
- Supports Wake on LAN over UDP. Supports High Speed Serial Peripheral Interface(SPI MODE 0,3).



Peripherals – Temperature System and Serial Tank



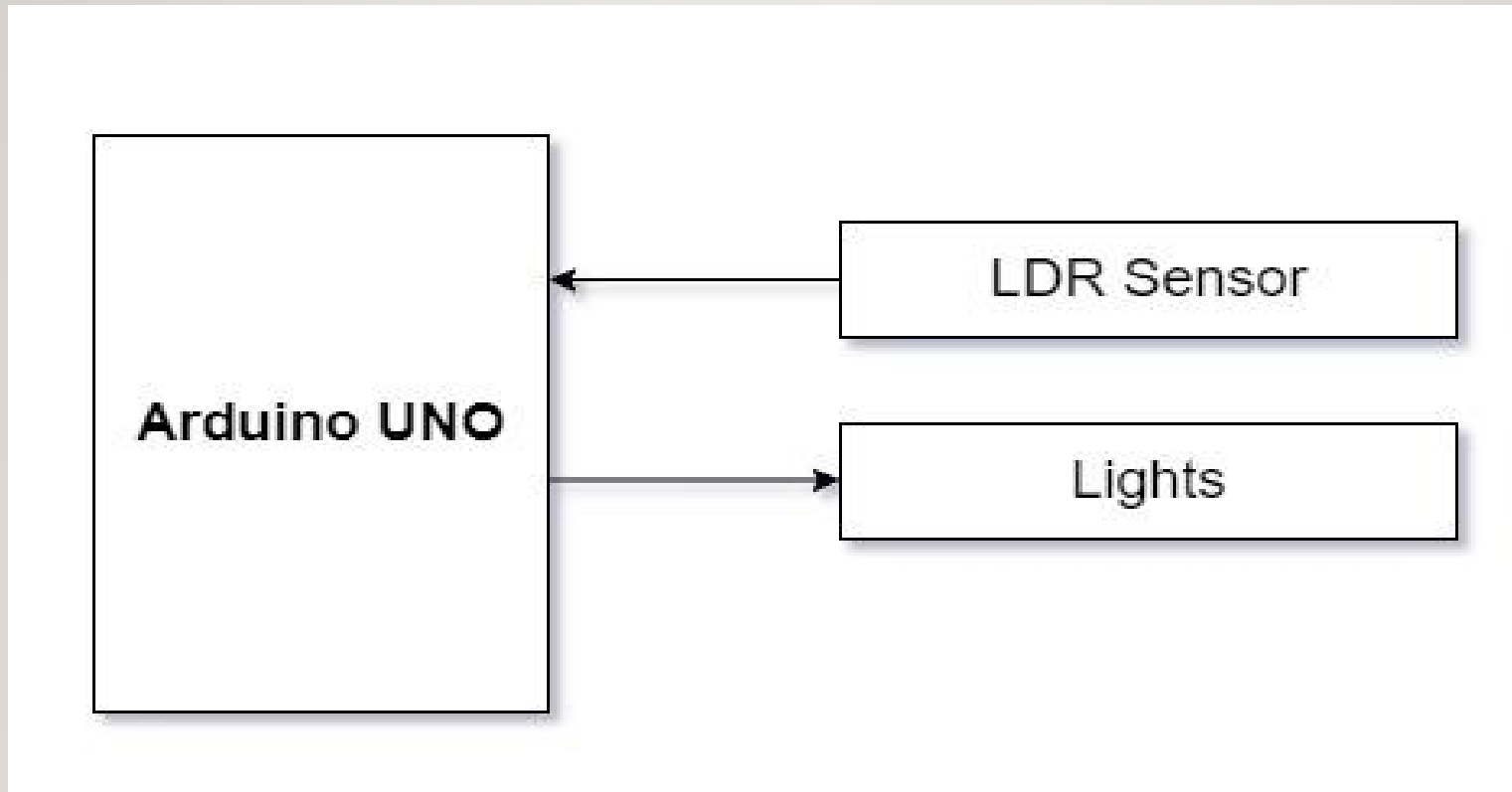
Temperature System



Serial Tank

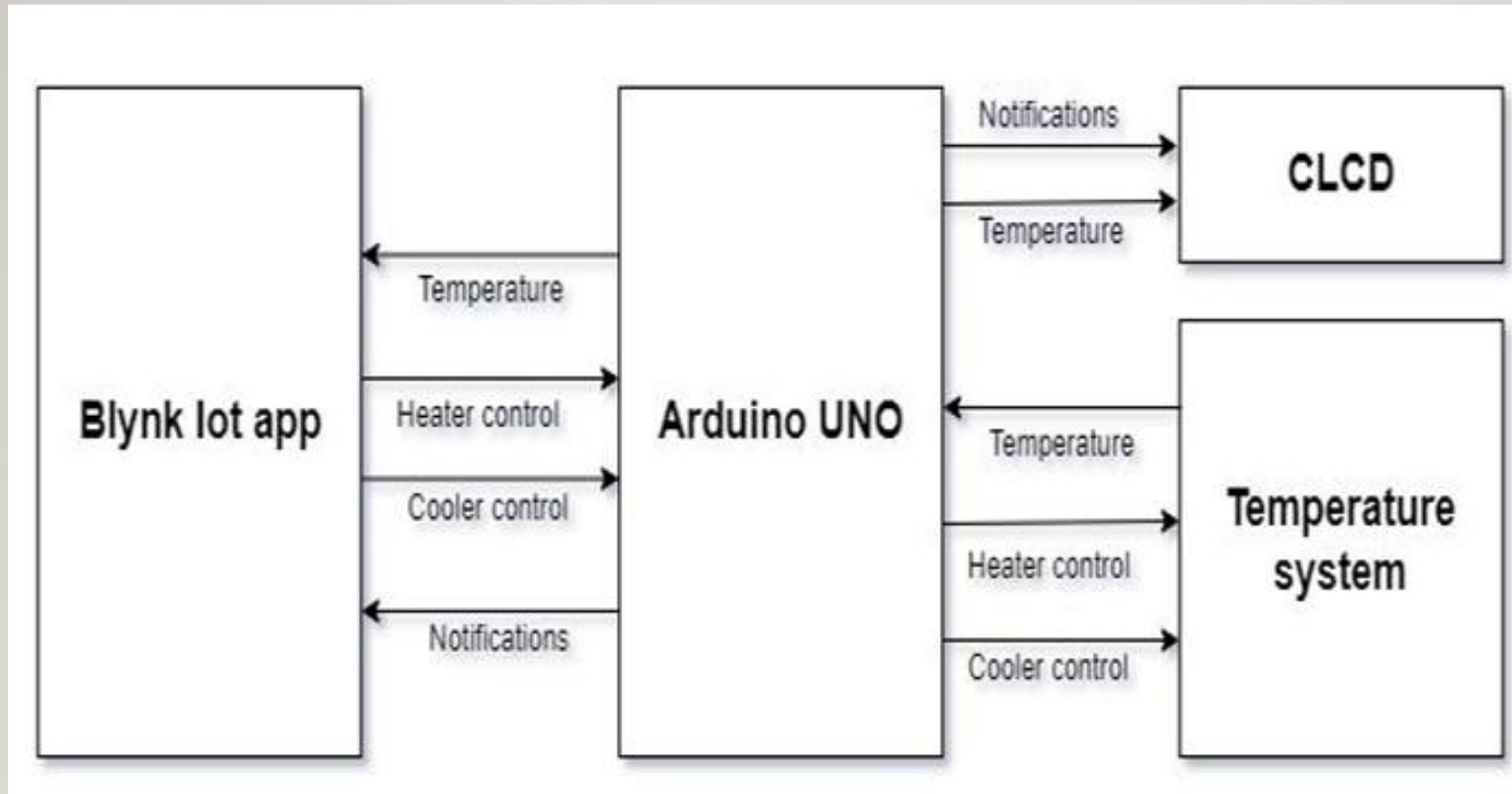
IMPLEMENTATION

1. Garden lights control:



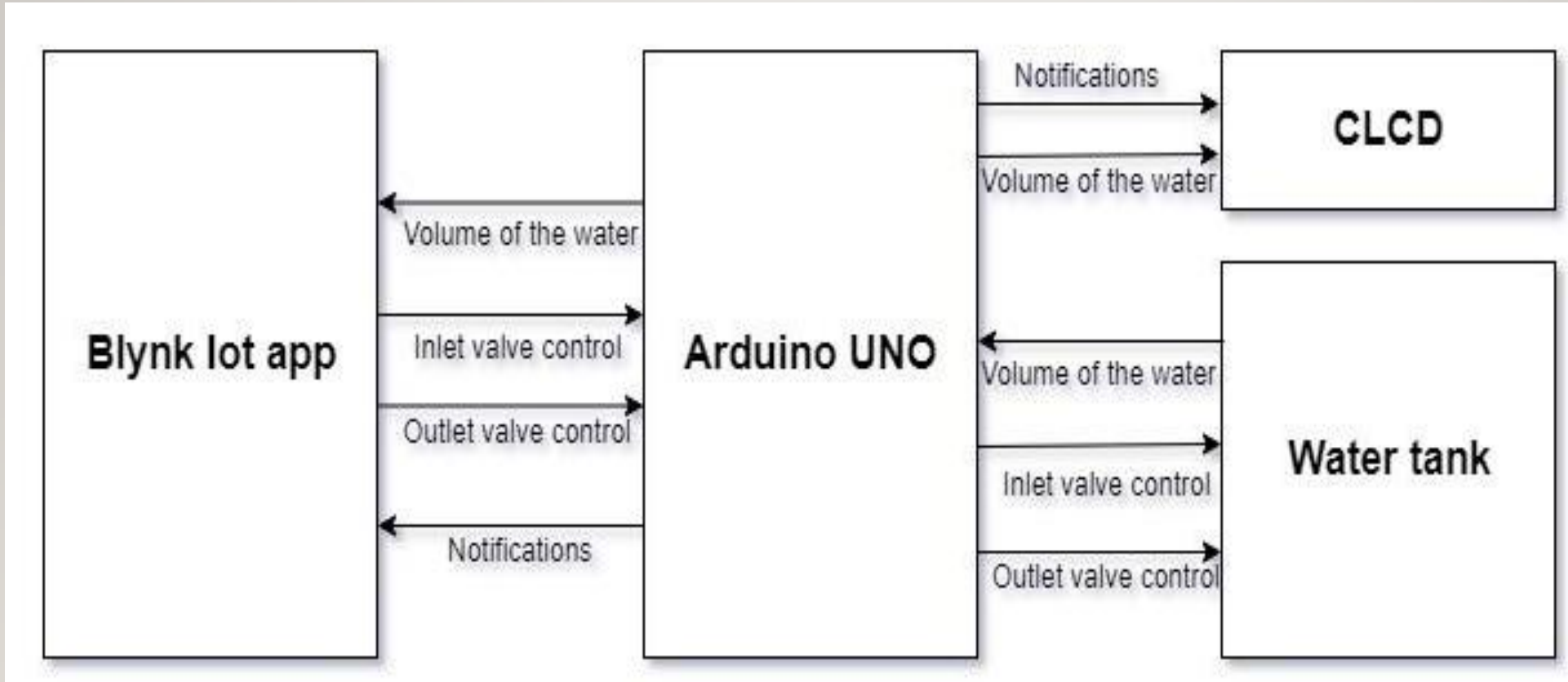
IMPLEMENTATION (Cont.)

2. Temperature Control System:



IMPLEMENTATION (Cont.)

3. Water tank inlet and outlet valve control:



IMPLEMENTATION (Cont.)

- A folder is created in the name of the project and we will include all the files like `home_automation_blynk_controlled.ino`, `ldr.cpp`, `ldr.h`, `serial_tank.cpp`, `serial_tank.h`, `temperature_system.cpp`, `temperature_system.h`, `main.h`.
- After coding in these files in Arduino IDE, we will export compiled binary file of the `.ino` file.
- In PICSimlab we will arrange all the spare parts required like LED, LDR, LCD, Temperature system, Ethernet.
- Then we will load the hex file for Arduino UNO board selected in PICSimlab to view the output.
- Also we will use Blynk app in mobile to visualize and control the same.



SIMULATION TOOLS USED IN PROJECT

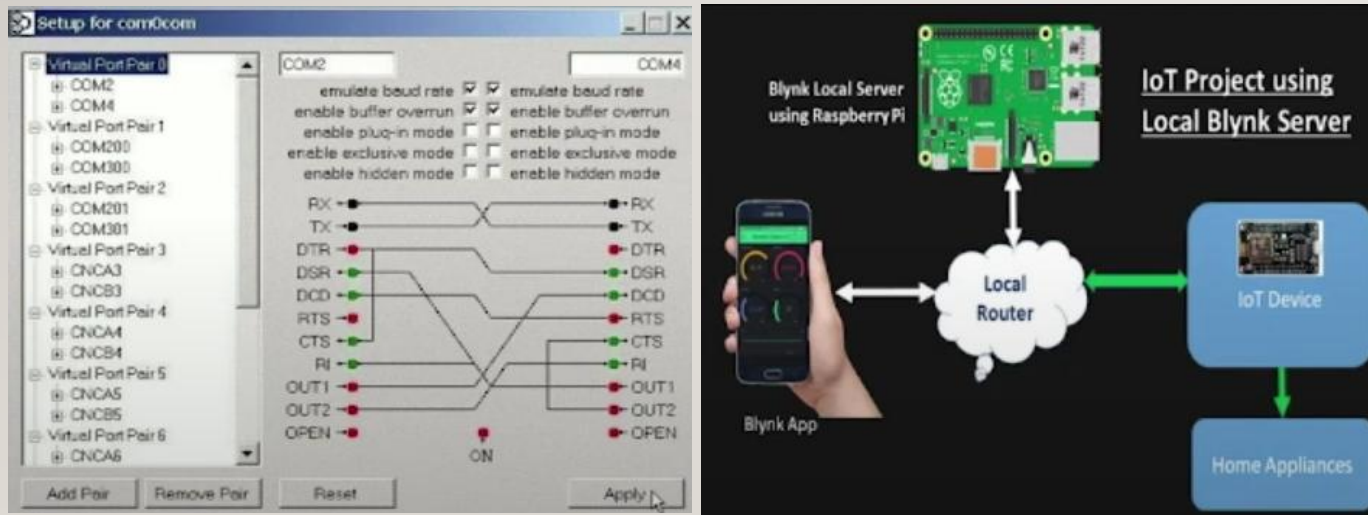


➤ Arduino IDE

IDE for Arduino programming and generating hex files

➤ PicSimLab

Used for simulating and running hex files



➤ Null Emulator

Used for serial communication between Arduino IDE, PicSimLab and Serial Tank

➤ Blynk Cloud and Blynk IoT Mobile Application

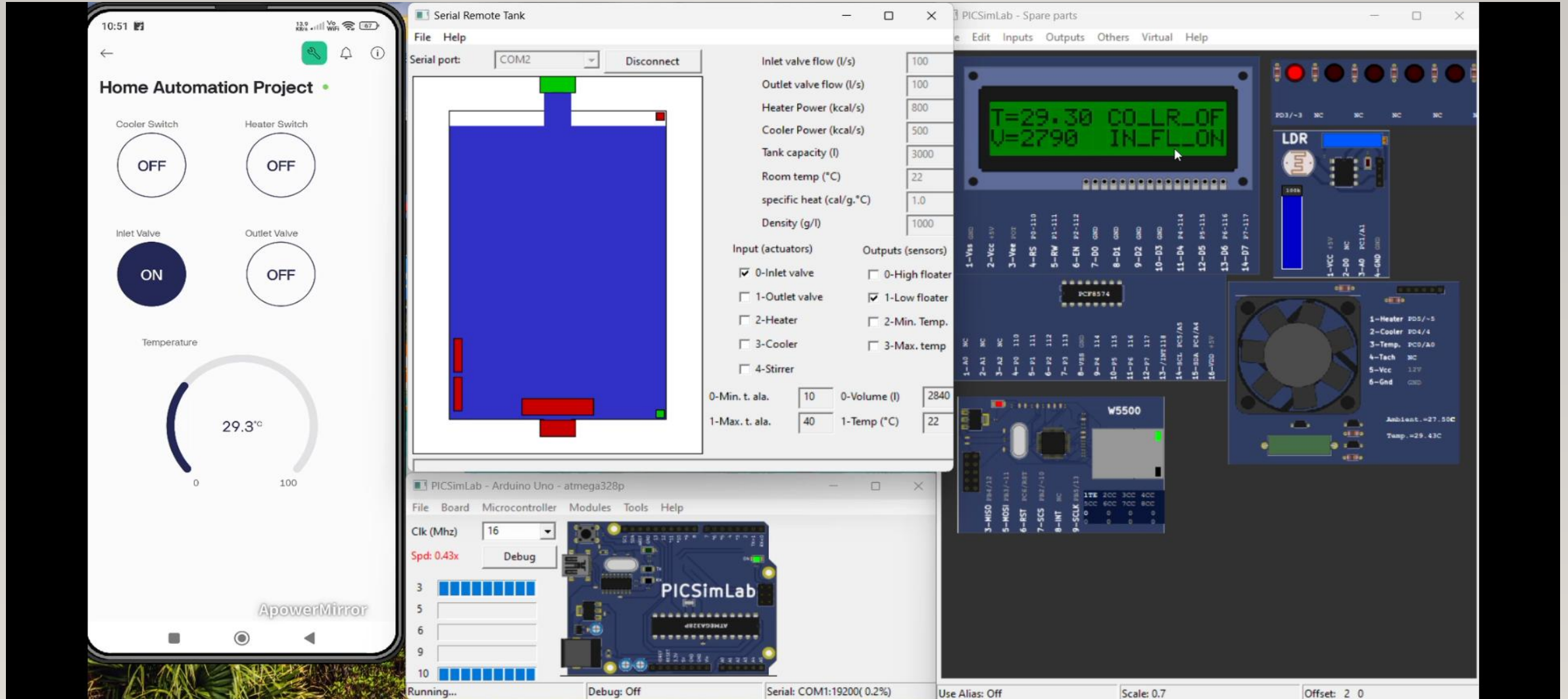
Used to control temperature system and serial tank in the project

Blynk Application

- Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, visualize it and do many other cool things.
- Create widgets on Mobile blynk application like-
 - **Button widgets** to control heater, cooler, inlet valve , outlet value.
 - **Gauge widgets** to display temperature and volume of the water in the tank on the mobile application.



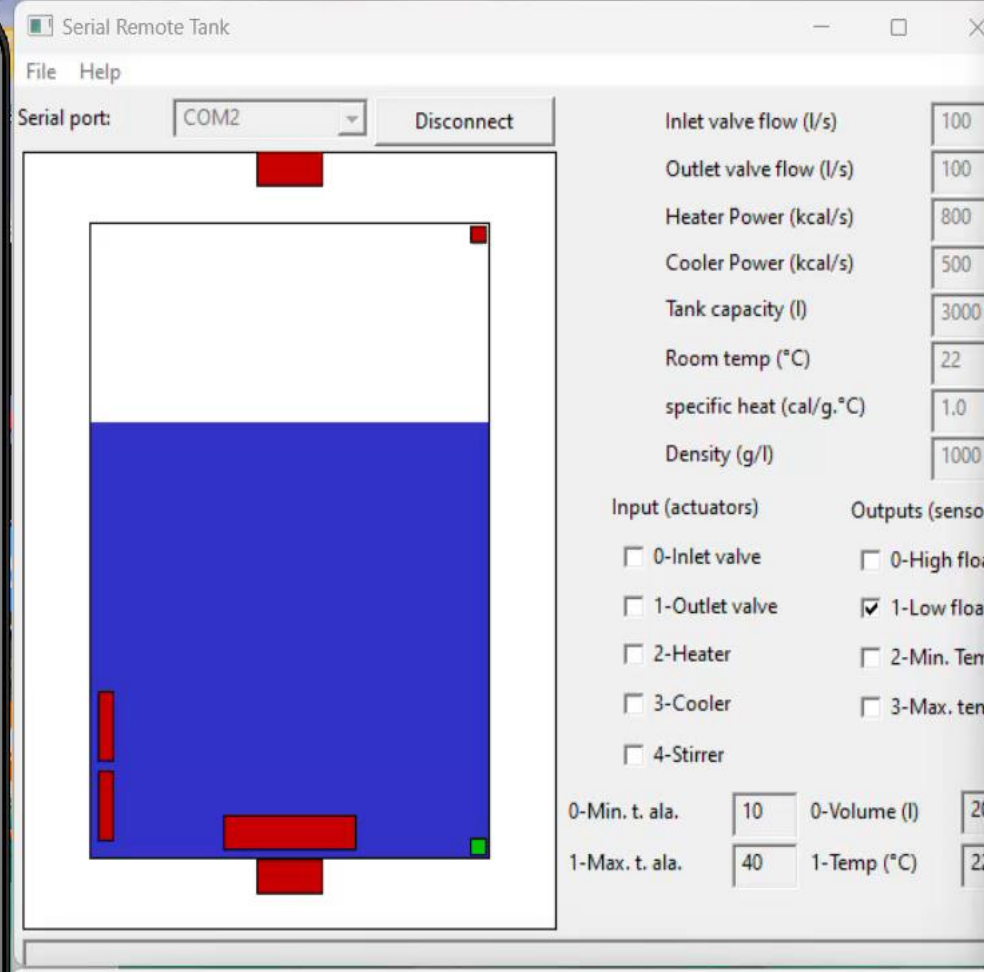
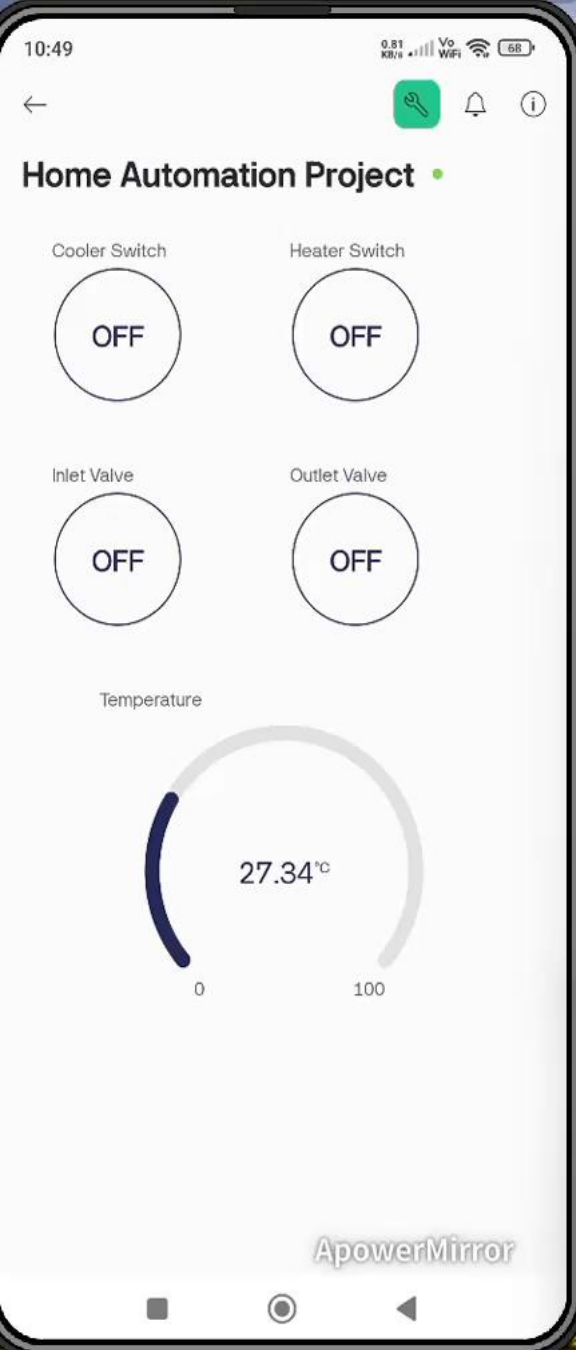
RESULTS



```
21 #include <LiquidCrystal_I2C.h>
22
23 #include "main.h"
24 #include "temperature_system.h"
25 #include "ldr.h"
26 #include "serial_tank.h"
27
28 char auth[] = BLYNK_AUTH_TOKEN;
29 bool heater_sw,inlet_sw,outlet_sw,cooler_sw;
30 unsigned int tank_volume;
31
32 BlynkTimer timer;
33
34 LiquidCrystal_I2C lcd(0x27, 16, 2); // set the LCD address to 0x27 for a 16 chars and 2 line display
35
36 // This function is called every time the Virtual Pin 0 state changes
37 /*To turn ON and OFF cooler based virtual PIN value*/
38
39 BLYNK_WRITE(COOLER_V_PIN)
40 {
41     cooler_sw=param.asInt();
42     if(cooler_sw)
43     {
44         cooler_control(ON);
```

Output

```
Sketch uses 27578 bytes (85%) of program storage space. Maximum is 32256 bytes.
Global variables use 1225 bytes (59%) of dynamic memory, leaving 823 bytes for local variables. Maximum is 2048 bytes.
```

CONCLUSION

- ✓ Using BLYNK Iot application and Picsimlab simulator, simulated home automation, where LED, temperature system, Serial tank resembles Light, Heater, Cooler and Water tank in real time.
- ✓ CLCD acts like a dash board used for displaying the events, Widgets from Blynk Iot app like button widgets are used to control heater, cooler and inlet valve, outlet valve.
- ✓ Gauge widgets to display the temperature and volume of the water

