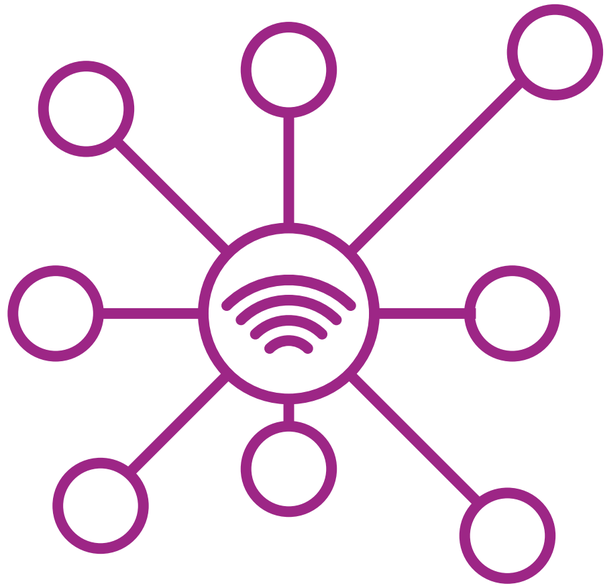
EMERTXE

IOT INTERNSHIP PROJECT

MOHAMAD ALZHORI

APRIL 2023



OUTLINE

I. Background
- About me
- Overview of IoT
- IOT Domains

II. Theory
- C Language in IOT
- C Concepts & C++

III. Practice
- Arduino
- Peripherals
- Project Requirements
- Software Used

IV. Implementation
- Putting Things Together
- Problems Faced

V. Project Overview and Future Scope
- Big Picture
- Project Usefulness
- Future Enhancements

VI. Acknowledgements

VII. References

VIII. Summary

Background

About me

Mohamad Alzhori

Electrical Engineering Student

Interests:

- IOT
- Embedded Systems
- Programming
- Project Management

Background

Overview of IoT

What's IOT ?

IoT is the network of physical objects connected to the internet that can exchange data. Devices collect data with sensors and software, then communicate via wireless or wired networking.

Importance

IoT has the potential to revolutionize a range of industries by enabling real-time monitoring and control, improving efficiency, and creating new opportunities for data-driven insights and services.

Examples of IOT

- **Smart Homes**
- **Wearable Devices**
- **Industrial IoT**

Background

IOT Domains

Healthcare

Remote patient monitoring allows healthcare providers to track vital signs and other health metrics in real-time, improving patient outcomes and reducing hospital readmissions.

Agriculture

Precision agriculture technologies can help farmers optimize water and fertilizer usage, reduce crop loss, and improve yields.

Energy

Smart grid technologies can help reduce energy waste, improve energy efficiency, and better integrate renewable energy sources.

Transportation

Connected vehicles and traffic sensors can help reduce congestion and improve safety.

Theory

C Language in IOT

Why C Language ?

- C language is a widely used programming language with a long history and strong community support.
- It is a low-level language that provides efficient memory management and direct hardware access, which is important for IoT devices with limited resources.
- Many IoT devices, such as sensors and microcontrollers, are programmed using C or a variant of C, such as C++.
- C language has a small runtime, which makes it suitable for embedded systems that have limited processing power and memory.

More Advantages of C

- C language is platform-independent, which means it can run on different hardware architectures, making it easier to develop and maintain code for different IoT devices.
- C language has a large number of libraries and tools available, which makes it easier to develop and debug IoT applications.
- C language provides the ability to write low-level, high-performance code, which is important for IoT applications that require real-time processing and fast response times.

Theory

C Concepts & C++

Introduction to C and Datatypes

- Basic structure of C programming
- Data types in C programming
- Variables and constants

Conditional Constructs and Operators

- If-else statements
- Switch statements
- Logical and bitwise operators

Operators and Array

- Arithmetic operators
- Relational operators
- Arrays

Pointers and Functions

- Pointers and memory management
- Functions and passing arguments
- Returning values from functions

Recursion, const with pointer and Strings

- Recursion and its uses
- Const pointer and pointer to const
- Strings and string manipulation

Preprocessing, storage classes, UDT- structure

- Preprocessor directives and macros
- Storage classes in C
- User-defined data types (structures)

Essentials of C++ - Object, class, constructor and destructor

- Introduction to C++
- Object-oriented programming concepts
- Classes, constructors, and destructors

Practice

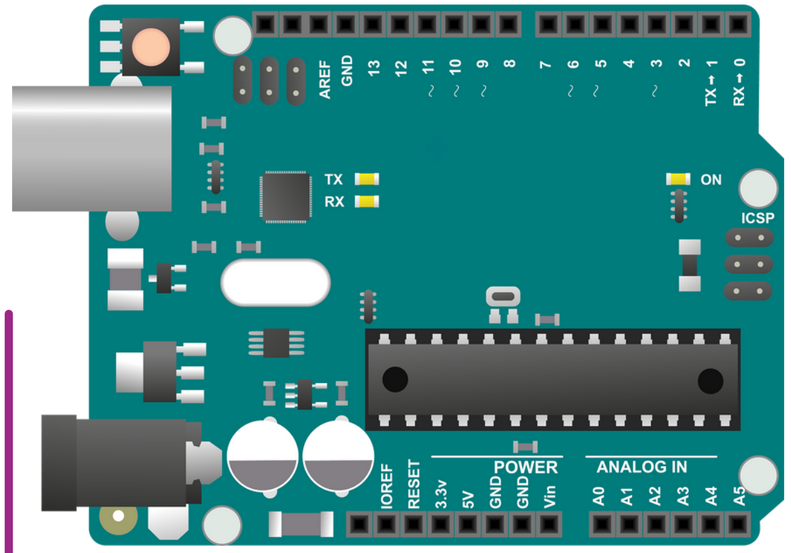
Arduino

What's Arduino?

Arduino is an open-source hardware and software platform that is designed for DIY electronics and prototyping projects. It consists of a microcontroller and a number of input/output pins that can be programmed to control various components such as sensors, motors, and lights. Arduino boards can be programmed using the Arduino IDE, which is a free software tool that makes it easy to write and upload code to the board. The low cost and flexibility of the Arduino board make it an ideal platform for experimenting with electronics and building custom projects.

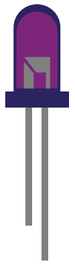
Why Arduino ?

Arduino is easy to use, with a simple programming language and a large community of developers and users. It is relatively low cost compared to other microcontroller boards, making it affordable for creating IoT prototypes and projects. Arduino is also open source, allowing anyone to use, modify, and distribute the hardware and software specifications. It is versatile and extensible, with a variety of board sizes and configurations, and the ability to easily connect to a wide range of sensors and actuators. Additionally, the platform can be expanded with add-on modules and shields, providing additional functionality for IoT projects.

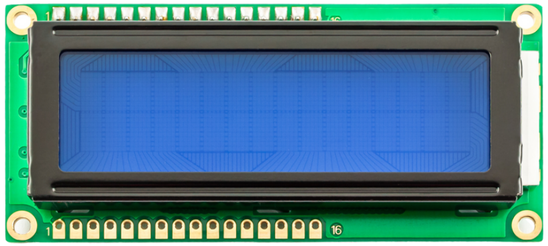


Practice

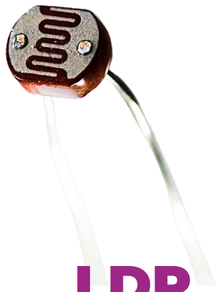
Peripherals



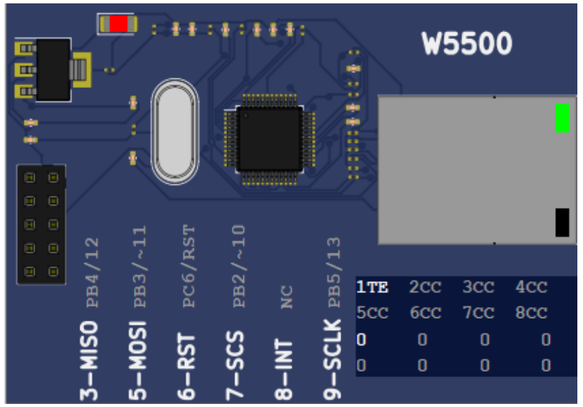
LED



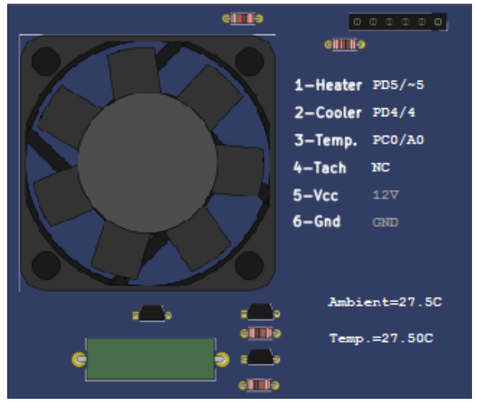
LCD



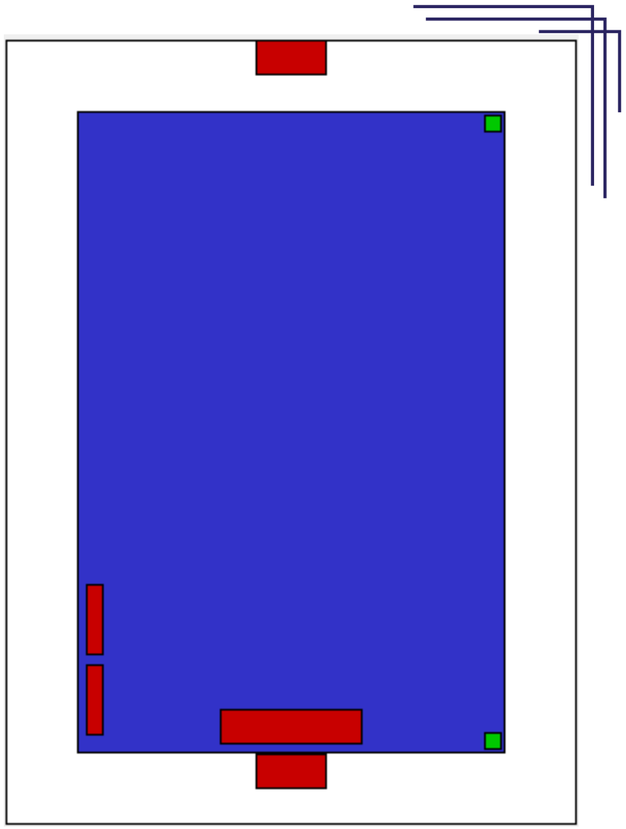
LDR



Ethernet Module



Temperature System (Heater/Cooler)



Serial Tank (Simulation)

Practice

Project Requirements

I. Garden Lights Control

- Reading the LDR sensor value
- Vary the brightness of the led based on the reading from LDR which resembles controlling garden lights based on the availability of sunlight

II. Temperature Control System

The temperature control system consists of a heating resistor, an LM35 temperature sensor, and a cooler. Which resembles the temperature control system at home.

- Read the temperature from LM35
- Display it on the CLCD
- Control the temperature of the system by turning ON/OFF the heater and cooler through the Blynk IOT mobile app .

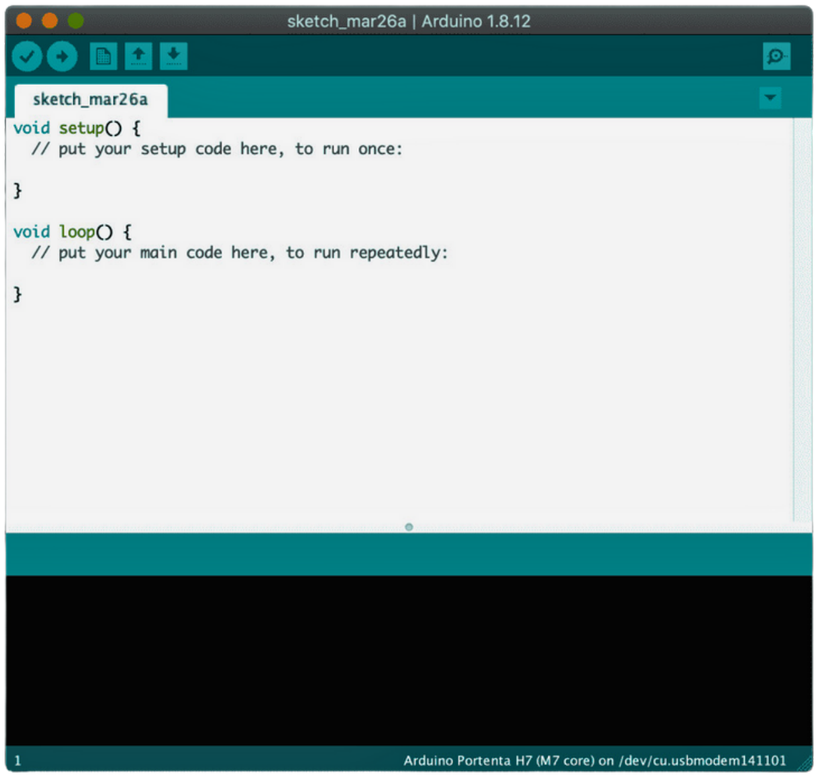
III. Water Tank Volume Control

- Read the volume of the water in the tank through Serial Communication
- Display it on the CLCD
- Control the volume of the water in the tank by controlling the inlet and outlet valve, by sending commands through serial communication

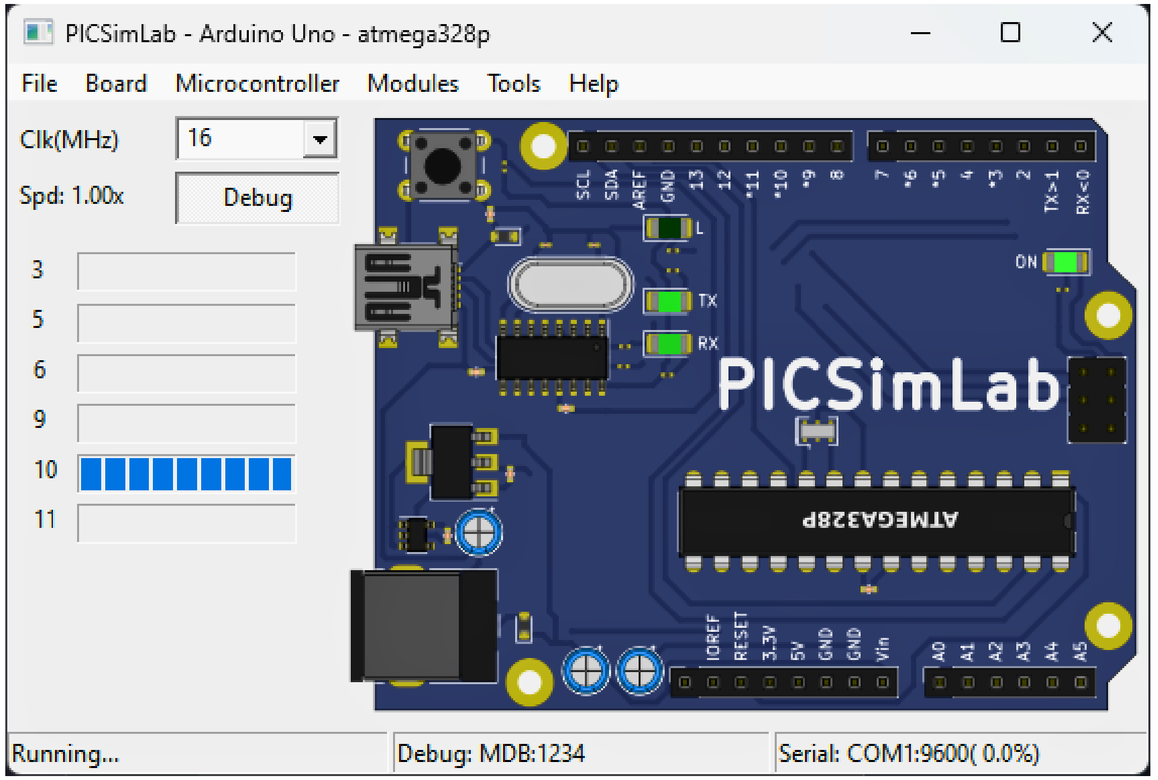
Practice

Software Used

Arduino IDE



PICSimLab



Practice

Software Used

Virtual Port

Virtual Serial Port Driver

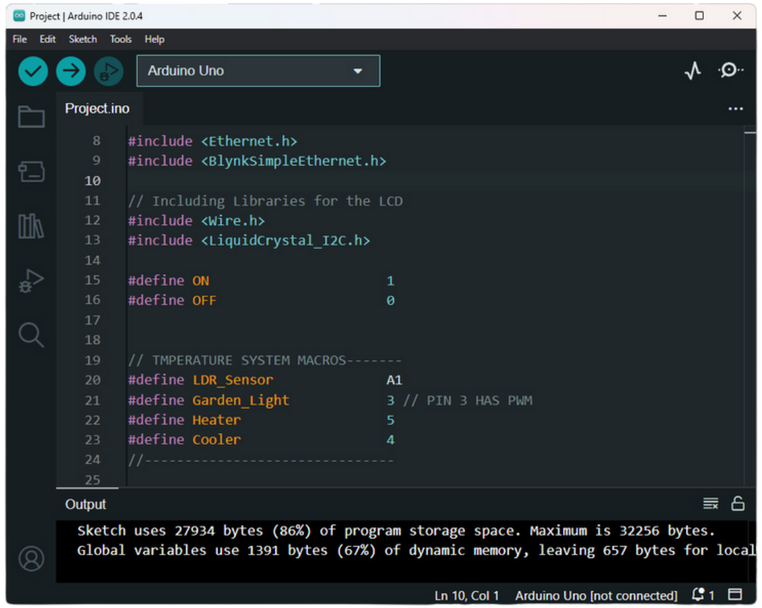
SCRCPY

Blynk

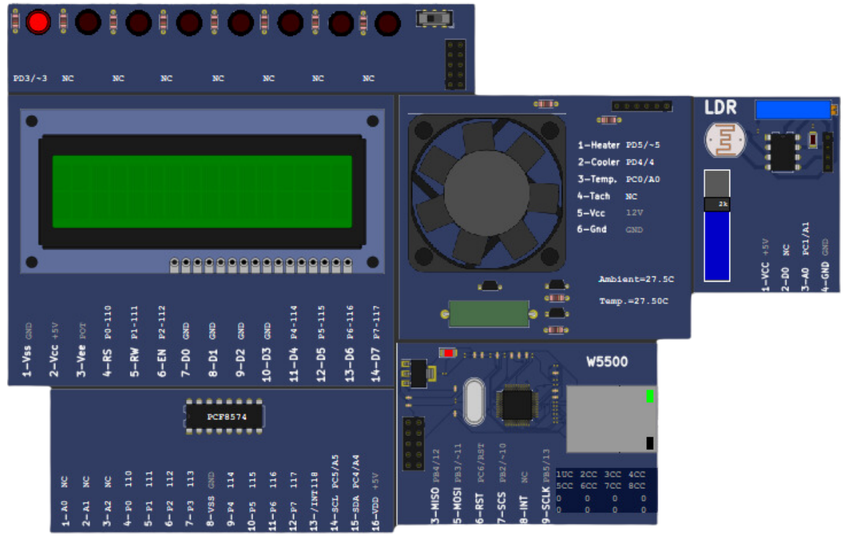
Implementation

Putting Things Together

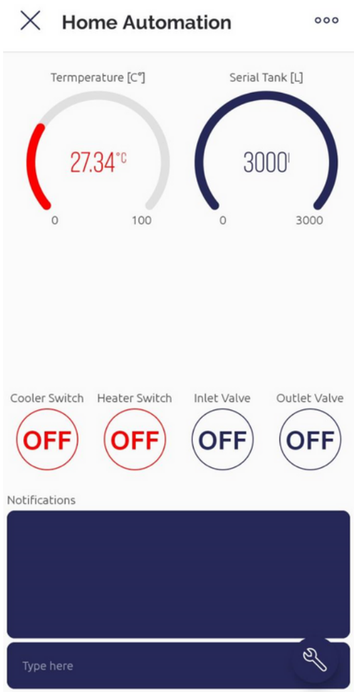
Writing Code



Simulating Hardware



Designing UI



Implementation

Problems Faced

Hardware Problems

One problem with using sensors like LDR or LM35 is that they do not provide ready-made values and may need to be converted into usable quantities. This can add complexity to the code and requires additional knowledge of the sensor's characteristics.

Cost

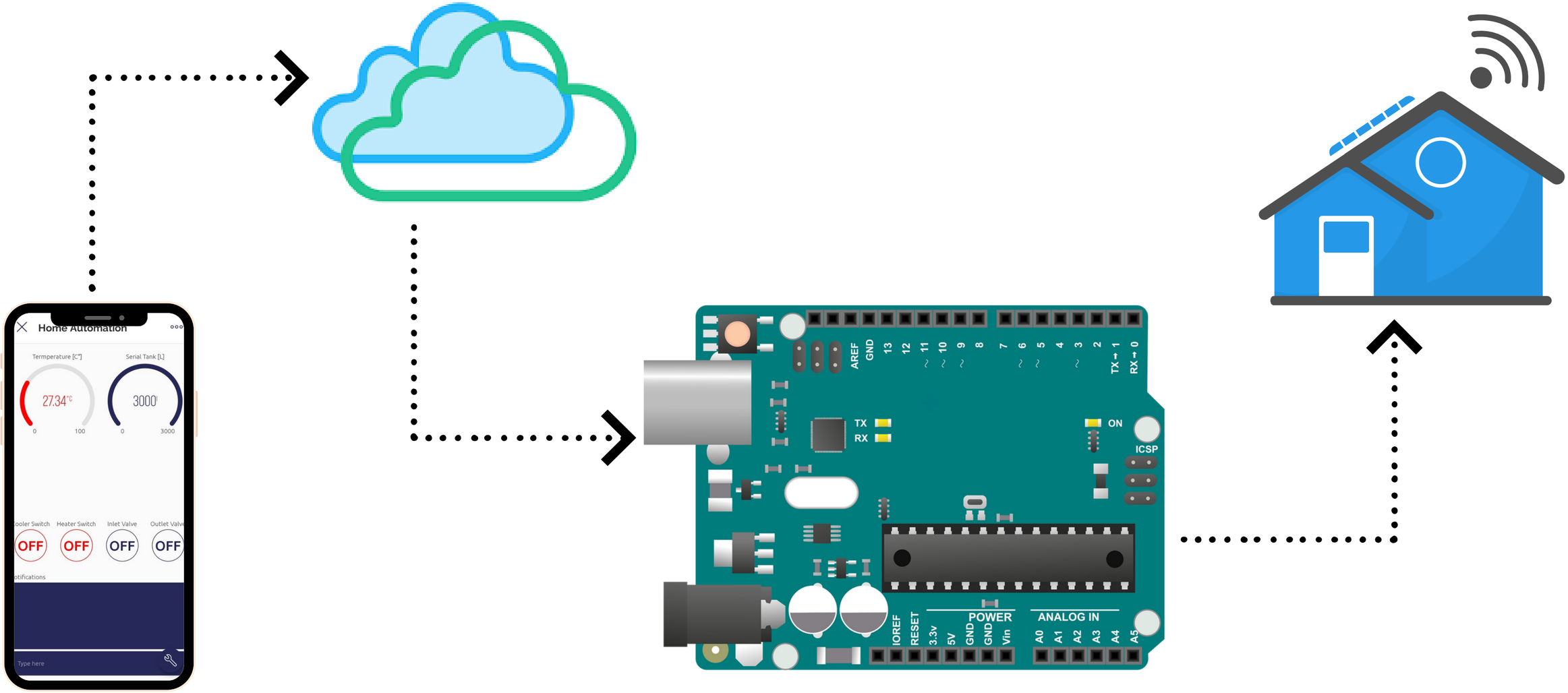
Embedded systems hobbyists often encounter the challenge of dealing with expensive hardware. To overcome this obstacle, this project utilized simulation tools to develop and test the system.

Connectivity Issues

The heavy reliance on Blynk Cloud in the project posed challenges for debugging and testing, particularly in times with poor internet connectivity.

Project Overview and Future Scope

Big Picture



Project Usefulness

Home automation

This project can be adapted to control and monitor various appliances in a house, such as thermostats, lights, and security systems. Users can control these appliances remotely through a mobile app or a web interface.

Agriculture

This project can be used to monitor and control different parameters in agricultural settings such as temperature, humidity, and water levels. This can help in improving crop yields and reducing water usage.

Industrial Automation

This project can be used in industrial settings to monitor and control different machines and equipment. This can help in optimizing the efficiency of industrial processes and reducing downtime.

Healthcare

This project can be adapted to monitor and control various parameters in healthcare settings such as temperature, humidity, and air quality. This can help in maintaining a safe and healthy environment for patients and healthcare workers.

Project Overview and Future Scope

Future Enhancements

Adding more sensors

Adding more sensors to monitor additional parameters such as humidity and air quality.

Upgrading the Microcontroller

Upgrading to a more powerful microcontroller like Arduino Mega or ESP32 to allow for more complex programming and handling of larger amounts of data.

Machine Learning

Incorporating machine learning algorithms to the system to enable it to learn and adapt to changing environmental conditions and optimize control parameters for efficiency and accuracy.

Better Integration

It may be useful to integrate the project with other smart home systems, such as Amazon Alexa or Google Home. This could allow for voice control and integration with other smart devices in the home.

VI. Acknowledgements

Mubeen J

Jayakumar balasubramanian

Special Thanks to Our Teachers

Saravanan S

Anusha M

Jayalaxmi N Dhanyal

References

Arduino Reference

https://www.arduino.cc/reference/en/

PICSimLab Referene

https://lcgamboa.github.io/picsimlab_docs/0.8.12/Introduction.html

Blynk Documentation

https://docs.blynk.io/en/

Summary

In short, we were able to design and implement an IoT-based temperature control system with the ability to remotely monitor and control the temperature, fluid levels, and lighting. The system was built using an Arduino Uno board, Blynk cloud platform, Ethernet adapter, LM35 temperature sensor, LDR sensor, and 16x2 LCD display. Through this project, we acquired skills in programming with Arduino IDE, working with sensors and actuators, using cloud-based IoT platforms, and designing and implementing an end-to-end IoT system.