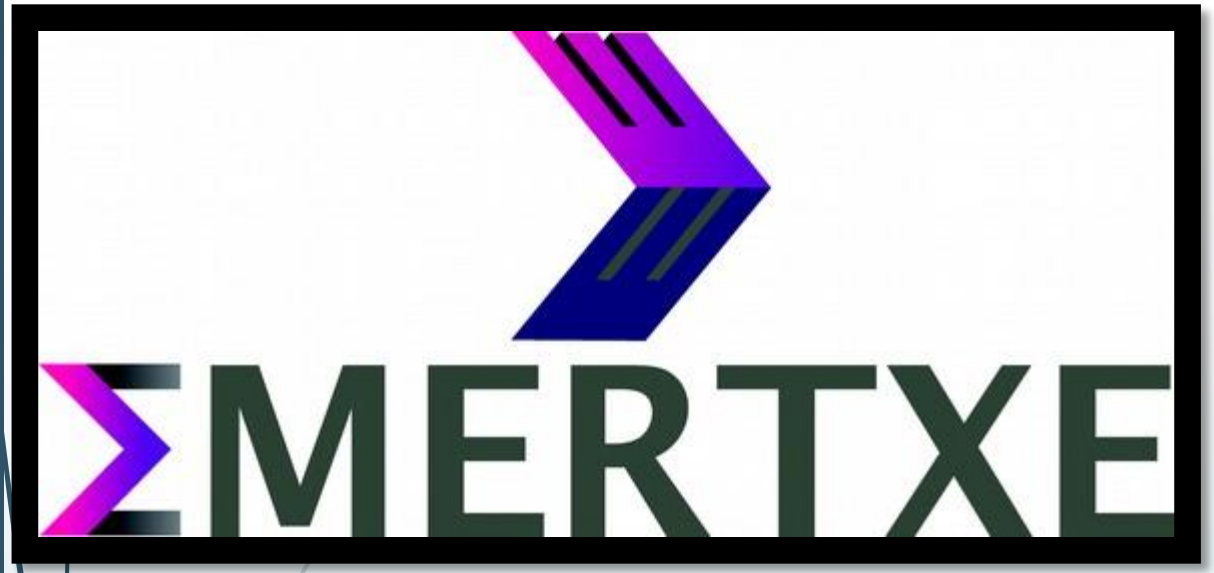


IOT BASED HOME AUTOMATION SYSTEM

Project Guide :- JAYALAXMI .N. DHANYAL

Submitted By : - SIDDINENI TARANYA

Designation :- Final Year

College :-Seshadri Rao Gudlavalleru Engineering College

ABOUT INTERNSHIP

This internship focused on **IoT Home Automation using the Blynk IoT Cloud**. The main objective was to provide students with exposure to the concepts of the **Internet of Things (IoT)** and its various real-world applications.

The internship began with an **introduction to C and C++ programming**, followed by a **brief overview of embedded systems**, and then progressed to **IoT concepts**. Throughout the program, we implemented several small basic projects to build our understanding, leading up to the **final project**.

The **final project** was developed using the **Arduino IDE**, **PICSIMLab simulator**, and the **Blynk IoT mobile application**, all of which are key tools in IoT-based development.

Timeline of Internship

Day 1 :C language, Data types, modifiers, Number systems, data representation

- **C Language:** Developed by *Dennis Ritchie* (1972, Bell Labs); a general-purpose, procedural language and the base for many modern languages.
- **Number Systems:** Binary, Octal, Decimal, Hexadecimal.
- **Data Types:** Primary (int, float, char), Derived (array, pointer), and User-defined (enum, struct).
- **Data Representation:** Bits, bytes, and memory storage of numbers.
- **Modifiers:** short, long, signed, unsigned.

Timeline of Internship

Day 2 :Sign modifiers , Decision-making statements , Looping structures

- **Sign Modifiers:** Use of *signed* and *unsigned* variables.
- **Decision Making:** *if*, *if-else*, *else-if*, *nested if*, and *switch* statements with syntax and flow diagrams.
- **Loops:** *for*, *while*, and *do-while* loops; use of loop control variables and handling infinite loops.

Timeline of Internship

Day 3 – Operators, Arrays & Jumping Statements

- **Jumping Statements:** *break*, *continue*, *goto*, *return*.
- **Operators:** Arithmetic, Logical, Relational, Assignment, Ternary, and Bitwise; with concepts of *overflow* and *underflow*.
- **Arrays:** One-dimensional and two-dimensional arrays.

Timeline of Internship

Day 4 – Functions and Pointers

- **Functions:**
  - Definition, declaration, and calling
  - Function arguments and return types
  - Call by value vs Call by reference

- **Pointers:**
  - Definition: variable storing address of another variable
  - Syntax: int *ptr;
  - Dereferencing operator * and address operator &

Timeline of Internship

Day 5 : Advanced Pointers

- **Endianness:** Difference between *Little Endian* and *Big Endian* byte order.
- **Pass by Reference:** Using pointers to modify actual function arguments.
- **Passing Arrays to Functions:** Example – calculating the average of array elements.
- **Returning Arrays:** Using *static/local arrays* or *dynamic memory allocation*.

Timeline of Internship

Day 6 – Pointer Types , Strings in C , Memory Segments

- **Pointer Types:** *Null*, *Wild*, and *Constant* pointers with examples.
- **Strings in C:** Declaration, input/output using *gets()* and *puts()*, and use of common *string.h* functions.
- **Memory Segments:** Overview of *Code*, *Stack*, *Heap*, *Data (static/global)*, and *Text* segments.

Timeline of Internship

Day 7 : Storage Classes & Preprocessor Directives

- **Storage Classes:** *auto*, *register*, *static*, and *extern* – their scope, lifetime, and usage.
- **Preprocessor Directives:** Use of *#include*, *#define*, *#ifndef*, and *#ifdef* for macros and conditional compilation.
- **Macros:** Creating reusable code segments using *#define* and managing compilation conditions.

Timeline of Internship

Day 8 : Introduction to C++

- **Introduction:** Developed by *Bjarne Stroustrup*; C++ is a *multi-paradigm* language combining procedural and object-oriented features.
- **POP vs OOP:**
  - *POP* – function-centric, less secure data handling.
  - *OOP* – object-centric, with data encapsulation and abstraction.
- **C vs C++:** Key differences in approach, data handling, and structure.
- **Classes & Objects:** Syntax for defining classes, creating objects, and accessing members.
- **Access Specifiers:** *public*, *private*, and *protected*.
- **Constructors:** Special functions for object initialization – *default*, *parameterized*, and *copy constructors*.

Timeline of Internship

Day 9 – OOP Concepts

Destructors

Pillars of Object-Oriented Programming

- **Destructor:** Automatically called when an object is destroyed to release resources.
- **Pillars of OOP:**
  - **Encapsulation:** Hiding data using classes.
  - **Abstraction:** Displaying only essential features.
  - **Inheritance:** Promotes code reuse — *Single, Multiple, Multilevel, Hierarchical,* and *Hybrid* types.
  - **Polymorphism:** Representing objects in multiple forms — *Compile-time* (function overloading) and *Run-time* (function overriding using virtual functions).

Timeline of Internship (Project part)

Arduino / IoT Recordings

Day 1 – Project Requirements
•Identify project goals (e.g., smart home, sensor automation)
•Required components: Arduino board, sensors, jumper wires, power source
•Circuit planning and block diagram

Day 2 – Introduction to IoT
•What is IoT?
•Components: sensors, actuators, connectivity, data processing
•Real-life examples: smart home, smart irrigation

Day 3 – Introduction to Embedded Systems (ES)
•Definition: combination of hardware and software for specific functions
•Microcontroller vs Microprocessor
•Arduino architecture overview

Day 4 – LED Interface
•Circuit diagram of LED with Arduino
•Code example: Blink LED
•Using digital pins and delay() function

Timeline of Internship (Project part)

Day 5 – CLCD Interface
•16x2 LCD working principle
•Pin configuration (RS, RW, EN, D0–D7)
•LCD initialization and data display code

Day 6 – Temperature Sensor
•Introduction to LM35 or DHT11
•Reading analog/digital data
•Displaying sensor output on Serial Monitor or LCD

Day 7 – Analog Write and PWM
•Pulse Width Modulation (PWM) concept
•Using analogWrite(pin, value)
•Applications: motor speed control, LED brightness

Timeline of Internship (Project part)

Sept 29 & Oct 6 – Serial Tank

•Using serial communication (Serial.begin, Serial.print)
•Water level sensor integration
•Displaying level data over serial monitor

Oct 7 – Blynk App Integration

•Overview of Blynk IoT platform
•Steps:
- Install Blynk app
- Create project, get auth token
- Code integration with Arduino IDE

•Controlling devices via mobile

TOPICS COVERED IN C PROGRAMMING

- Fundamental use of c and its applications
- Number systems
- Data types
- Conditional Statements
- Operators
- Arrays
- Pointers
- Functions
- Storage classes
- Various macros, Header files and pre-processors

Internet Of Things(IOT)

The Internet of Things (IoT) is a network of physical objects embedded with sensors, software, and other technologies that allow them to connect and exchange data over the internet

- **Applications of IoT:**

- **Smart Home Automation** – Lights, fans, ACs controlled via phone.
- **Smart Agriculture** – Monitoring soil moisture and automating irrigation.
- **Healthcare** – Patient vital monitoring systems.
- **Industrial IoT** – Predictive maintenance of machines.
- **Smart Cities** – Smart traffic and waste management.

Embedded System

An *embedded system* is a microcontroller or microprocessor-based system designed to perform a specific, dedicated function.

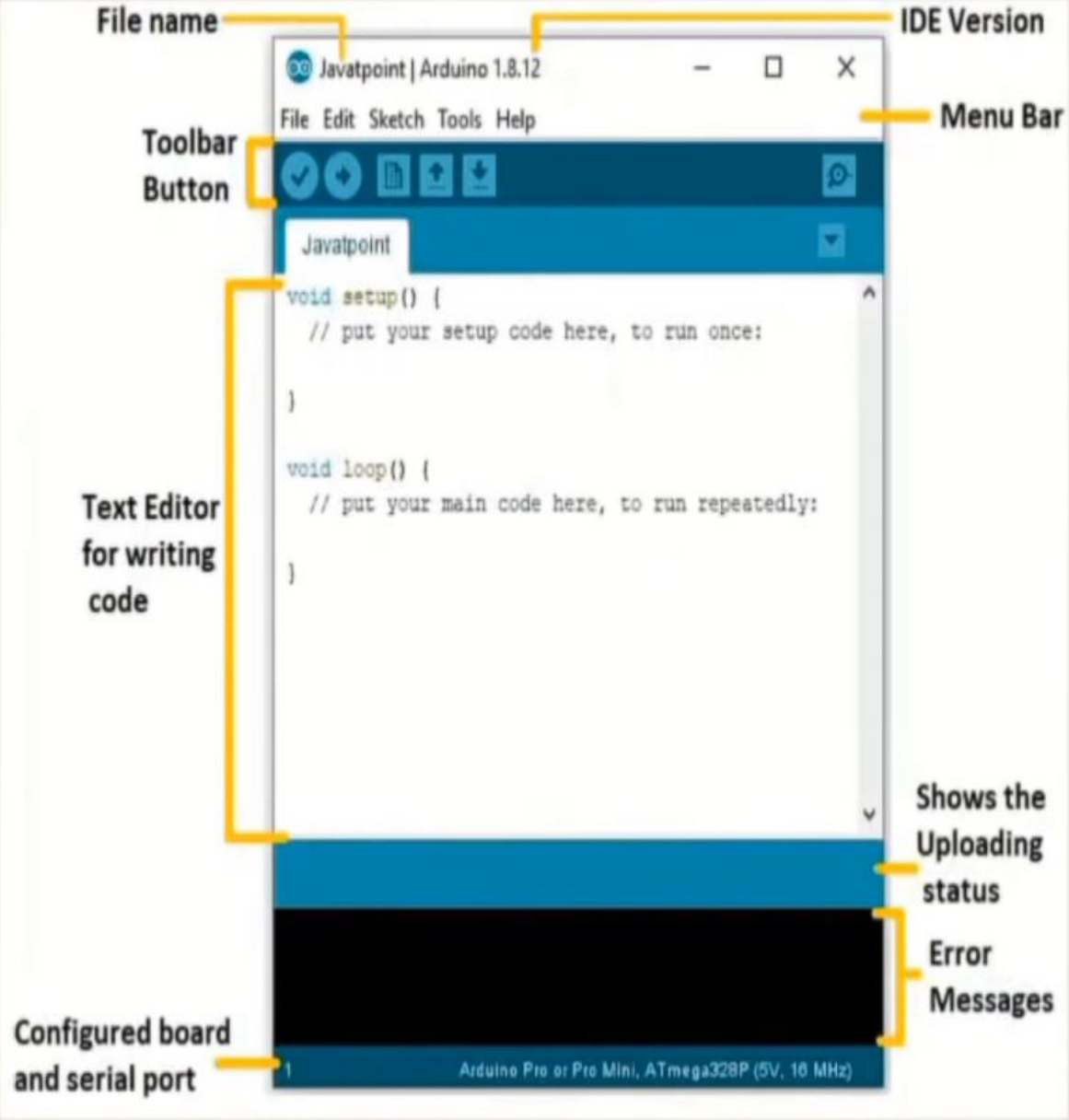
Examples : Washing machine, ATM, Smartwatch, Airbag system, Router.

- **Components:**
- **Hardware:** Microcontroller, Memory, Sensors, Actuators, I/O Ports
- **Software:** Embedded firmware, RTOS, Application code

Applications:

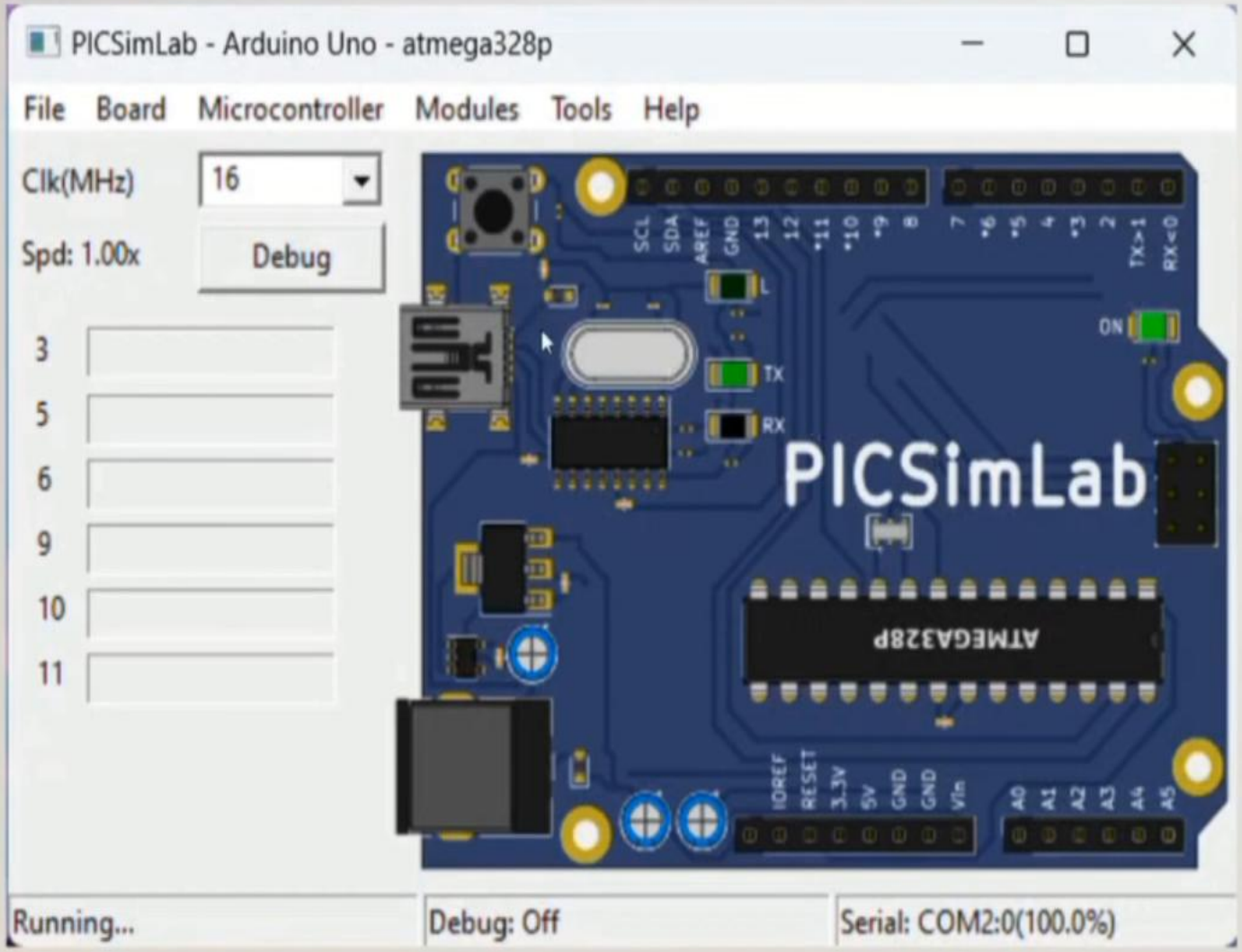
- **Automobile:** Engine control, Airbags
- **Medical:** Pacemakers, Monitoring systems
- **Industrial:** Robotics, Automation
- **Consumer:** Smart TVs, Cameras

ARDUINO IDE:



- The Arduino Integrated Development Environment **(Arduino IDE)** contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and series of menus.
- It connects to the Arduino hardware to upload programs and communicate with them.
- Version Used – **Arduino 1.8.12**

PICSIM LAB:

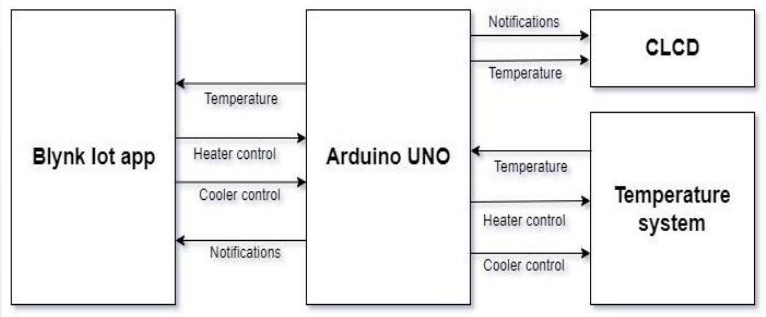


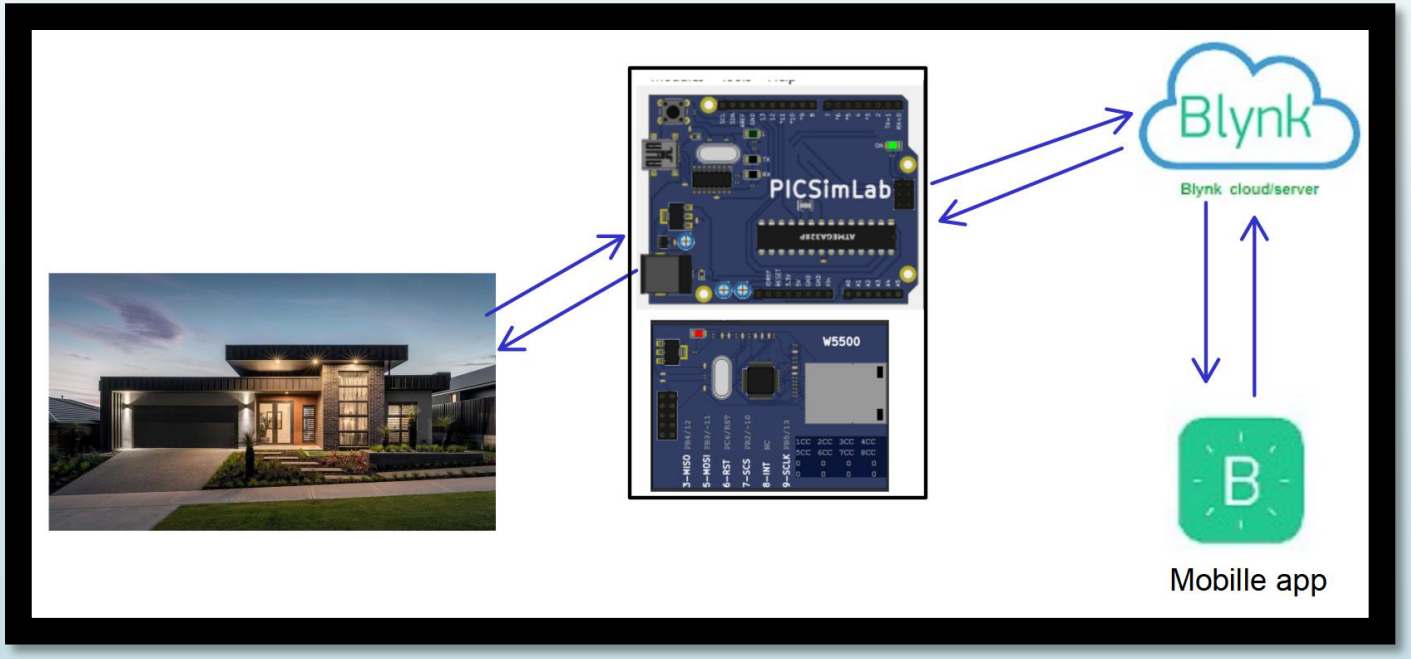
- ✓ PICSimLab is a real-time emulator of development boards with integrated MPLABX/avr-gbd debugger.
- ✓ The simulator which we used to implement the programs in our project is PICSimLab.
- ✓ With atmega328p Arduino Uno board.
- ✓ We upload our program in the format of .hex file in the PICSimLab simulator and we run it.
- ✓ And the output is observed according to the parts used.

Topics covered related to Iot

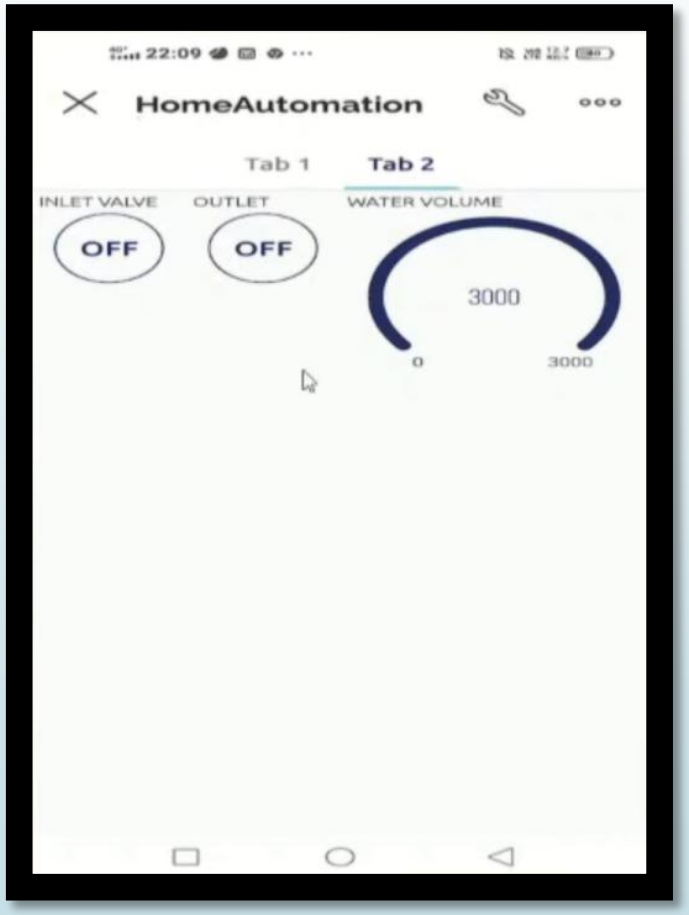
- Introduction to IoT
- IoT architecture and its applications
- Embedded Systems
- Arduino Ide and PICSIMLAB simulator
- Working of LED and LDR
- Basic program of controlling LED using LDR
- About CLCD and its working and a program to interface CLCD
- Basic program related to temperature control
- Introduction to blynk iot application
- Basic program using blynk iot application
- Basic program related to serial remote tank
- Final project implementation

IOT BASED HOME AUTOMATION SYSTEM

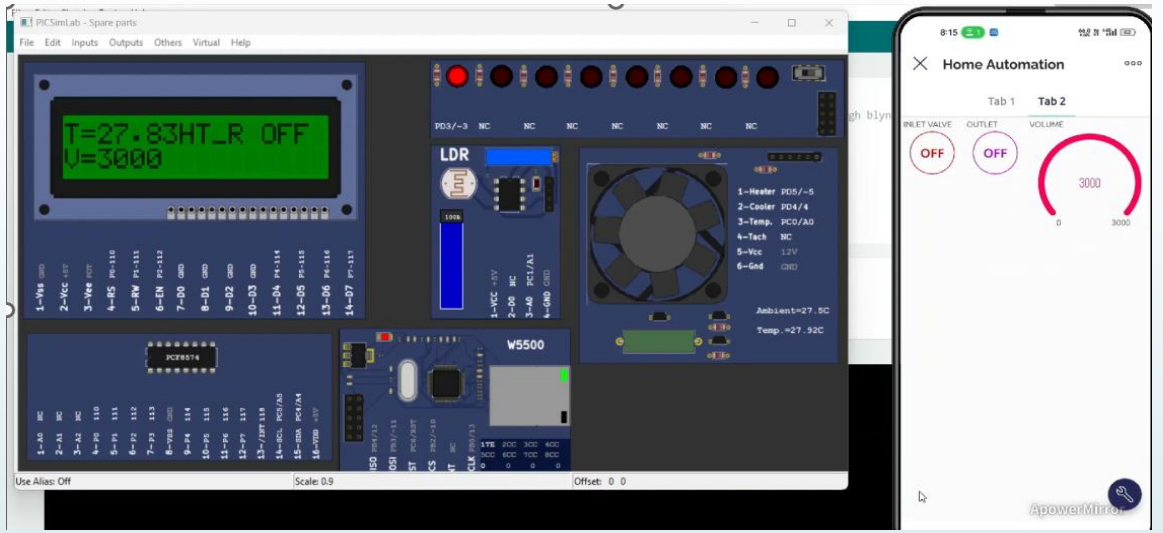


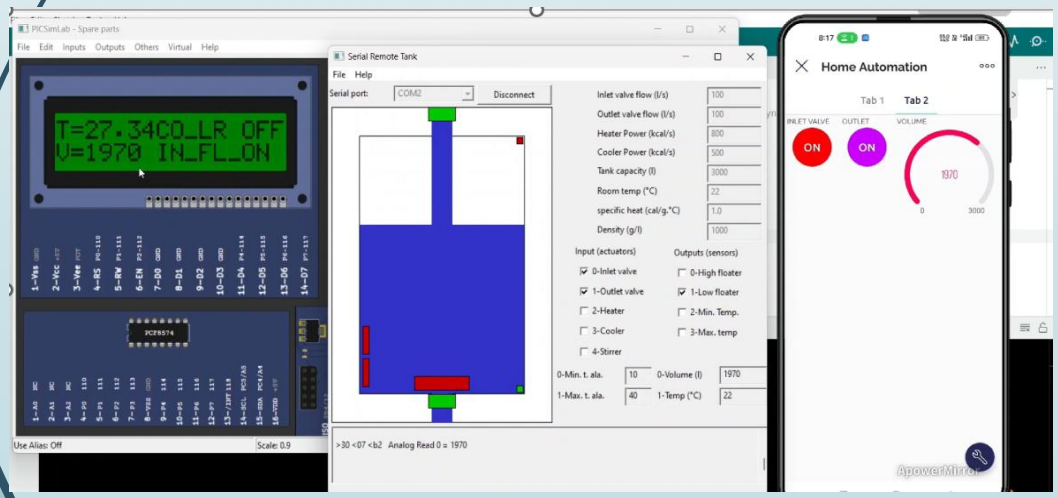


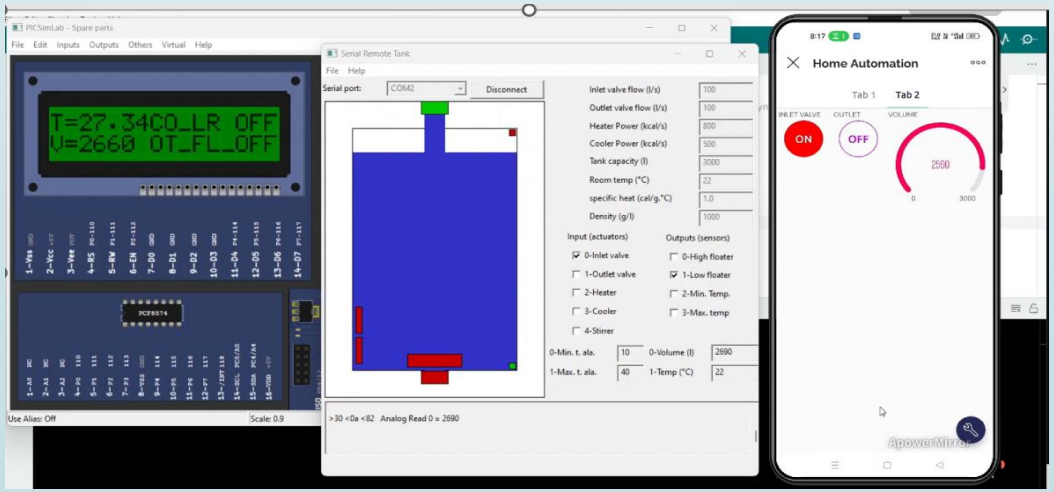
BLYNK IOT MOBILE APP



Project Outputs:







Conclusion

This internship provided a comprehensive learning experience in **C/C++ programming, embedded systems, and IoT development**. Through hands-on projects with Arduino and the Blynk IoT platform, I not only strengthened my technical skills but also gained valuable insights into **real-world application development**. Additionally, the program emphasized the importance of **teamwork, documentation, and systematic project planning**, fostering a mindset of **innovation, problem-solving, and continuous learning.**

THANK YOU !!