VIETNAM GENERAL CONFEDERATION OF LABOUR

TON DUC THANG UNIVERSITY

FACULTY OF ELECTRICAL ENGINEERING

LUU PHUOC SANG – 420H0317

Ultrasonic Distance Measurement Device

PROGRESS OF INDIVIDUAL PROJECT IMPLEMENTATION

ELECTRONICS AND TELECOMMUNICATIONS ENGINEERING

HO CHI MINH CITY, YEAR 2023

PROGRESS OF INDIVIDUAL PROJECT IMPLEMENTATION

LUU PHUOC SANG – 420H0317

Ultrasonic Distance Measurement Device

PROGRESS OF INDIVIDUAL PROJECT IMPLEMENTATION

ELECTRONICS & TELECOMMUNICATIONS ENGINEERING

Advised by

Dr. Do Vinh Quang

HO CHI MINH CITY, YEAR 2023

PROGRESS OF INDIVIDUAL PROJECT IMPLEMENTATION

ACKNOWLEDGMENT

Dear Professor Do Vinh Quang,

I would like to express my sincere gratitude for the guidance and support you have provided throughout my project. Your knowledge, expertise, and patience have been invaluable, and I could not have completed this project without your help.

Your feedback and constructive criticism have helped me to refine my ideas and improve my work. You have challenged me to think critically and creatively, and have inspired me to strive for excellence.

I have learned a great deal from this project, and I am confident that the skills and knowledge I have acquired will serve me well in my future endeavors. Your mentorship has been instrumental in my growth as a student and as a person, and I am truly grateful for the opportunity to work with you.

Besides, I would like to take a moment to express my heartfelt gratitude to my family and friends for their unwavering support and encouragement throughout my project. Their love and encouragement kept me motivated and inspired me to give my best.

Ho Chi Minh city, day month year

Author

Luu Phuoc Sang

TON DUC THANG UNIVERSITY

FACULTY OF ELECTRICAL AND ELECTRONICS ENGINEERING

PROGRESS OF INDIVIDUAL PROJECT IMPLEMENTATION

Student's name: Luu Phuoc Sang Student's ID: 420H0317

Class: 20H40201 Major: Electronics and Telecommunications Engineering

Advisor: Dr. Do Vinh Quang

Project title: Ultrasonic Distance Measurement Device

Week/Date	**The work has been finished (Please write details)**	**Confirmation of Advisor**
1 (18/09 – 24/9/2023)	- Receive the Project assignment sheet - Understand clearly the project content requirements and project implementation regulations	
2 (25/09 – 01/10/2023)	- Learn about Arduino Uno - Learn about Ultrasonic Module HC-SR04 - Learn about Module Bluetooth - Learn about Module DHT11	
3 (02/10 – 08/10/2023)	- Design the principle of operation, the function of blocks in the circuit - Write report	
4 (09/10 – 15/10/2023)	- Write the program to measurement the distance between the device and the object - Test the program with proteus or breadboard - Write report	

5 **(16/10 – 22/10/2023)**	**Advisor evaluates the level of completion……..%** □ Agree □ Disagree	
6 **(23/10 – 29/10/2023)**	- Write the program to send the data receive from device to laptop by module bluetooth - Test the program with proteus or breadboard - Write report	
7 **(30/10 – 05/11/2023)**	- Write the program to measurement the distance two objects by two Modules in two different sides - Test the program with proteus or breadboard - Write report	
8 **(06/11 – 12/11/2023)**	- Write the program to display the temperature and humidity by using DHT11 - Implement the hardware prototype - Write report	
9 **(13/11 – 19/11/2023)**	- Display all the data receive on LCD1602 with user-configurable display switching times. - Test and debug the prototype - Submit the final report	
10 **(20/11 – 26/11/2023)**	**Advisor has approved the project report. The project defense is:** □ Allowed □ Not allowed	

Table of Contents

LIST OF FIGURE

CHAPTER 1. INTRODUCTION

1.1 Research purpose

Currently, in the ever-evolving landscape of modern life heavily reliant on technology, the development of multifunctional distance measurement systems plays a crucial role in meeting the practical needs of today's society. Sensor technology is increasingly being applied in various fields, from industrial automation to smart applications, from environmental monitoring to healthcare and smart personal devices. The rapid development of IoT and interconnected smart devices necessitates intelligent and multifunctional sensor systems, which the project on a distance measurement system integrating temperature, humidity sensors, and wireless data transmission via Bluetooth aims to address.

To execute this project, specific components such as the HC-SR04 ultrasonic sensor for distance measurement, temperature and humidity sensors for environmental monitoring will be used. Integrating these sensors is essential to create a comprehensive measurement system. Coupled with the use of the LCD1602 display screen and HC-06 Bluetooth module for convenient data display and transmission, this will form an intelligent and flexible system.

The precise placement and calibration of sensors are crucial to ensure the accuracy of collected data. Subsequently, building and programming the control system to read data from sensors, displaying it on the LCD1602 screen, and transmitting it via Bluetooth will be a vital part of this project. Optimizing and integrating these functions cohesively and flexibly will ensure the stable and efficient operation of the system.

In summary, the combination of multifunctional sensors and wireless data transmission system will create an intelligent, versatile solution applicable across various fields. Careful research and precise implementation of these components are necessary to achieve optimal performance and wide practical applications.

1.2 Technical specification

The distance measuring circuit utilizing two HC-SR04 ultrasonic sensors allows measurement in two different directions with an accuracy of ±3mm and a measurement range from 2cm to 400cm. Additionally, the integration of temperature and humidity sensors enables the collection of environmental information with an accuracy of ±0.5°C for temperature and ±3% for humidity. The gathered data will be displayed directly on the LCD1602 screen, providing real-time information on distance, temperature, and humidity.

The system also incorporates the HC-06 Bluetooth module for wireless transmission of distance, temperature, and humidity data. Bluetooth 2.0 communication supports data transmission speeds of up to 2.1Mbps, ensuring flexibility in data communication.

Moreover, the circuit features an alert function through the HC-06 module when measured values exceed predefined thresholds. It also pairs with a control circuit to automatically disconnect from devices when necessary.

With reliable technical specifications and versatile application potential, this project can be used in various fields such as distance measurement, environmental monitoring, and remote control, particularly suitable for IoT (Internet of Things) applications and smart devices.

1.3 Implementation idea

To implement this project, I will begin by connecting two HC-SR04 ultrasonic sensors to the microcontroller to collect distance information in two directions. The temperature and humidity sensors will be integrated and connected to the microcontroller to gather environmental data. Then, I will set up the interface with the LCD1602 screen to display the collected information.

The integration of the HC-06 Bluetooth module will enable wireless transmission of distance, temperature, and humidity data to other devices. This will require programming the Bluetooth communication protocol and setting up control commands to accurately and continuously send data.

Additionally, I will develop an alert function through the Bluetooth module when the measured values exceed predefined thresholds. Alongside integrating the control circuit, I will program the system to automatically disconnect when necessary to ensure safety and energy efficiency.

The implementation process will involve software programming, circuit design, and hardware assembly. It will take time and patience to accurately integrate the components and ensure the stability of the system. This implementation plan will help create an intelligent, flexible, and practical distance measurement system for various real-world applications.

CHAPTER 2. HARDWARE DESIGN

2.1 Function block diagram

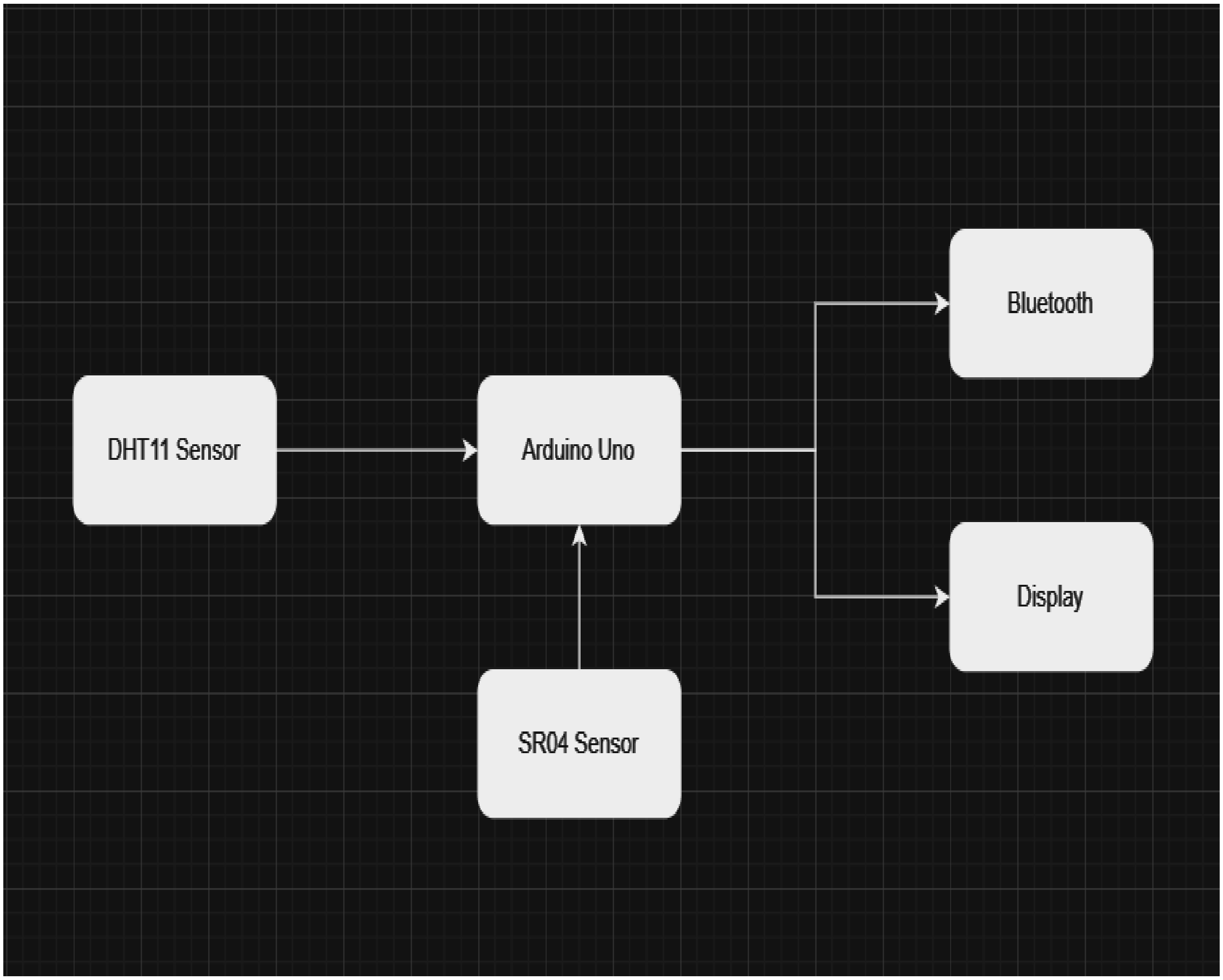

Figure 1: The block function diagram of the project

The ultrasonic distance measurement device is divided into 5 different blocks corresponding to different functions and tasks in the circuit to combine together to form a unified body. The 5 blocks include: Arduino uno block, HC-SR04 sensor block, Display block, Bluetooth Module block, DHT11 sensor block.

2.2 Arduino Uno

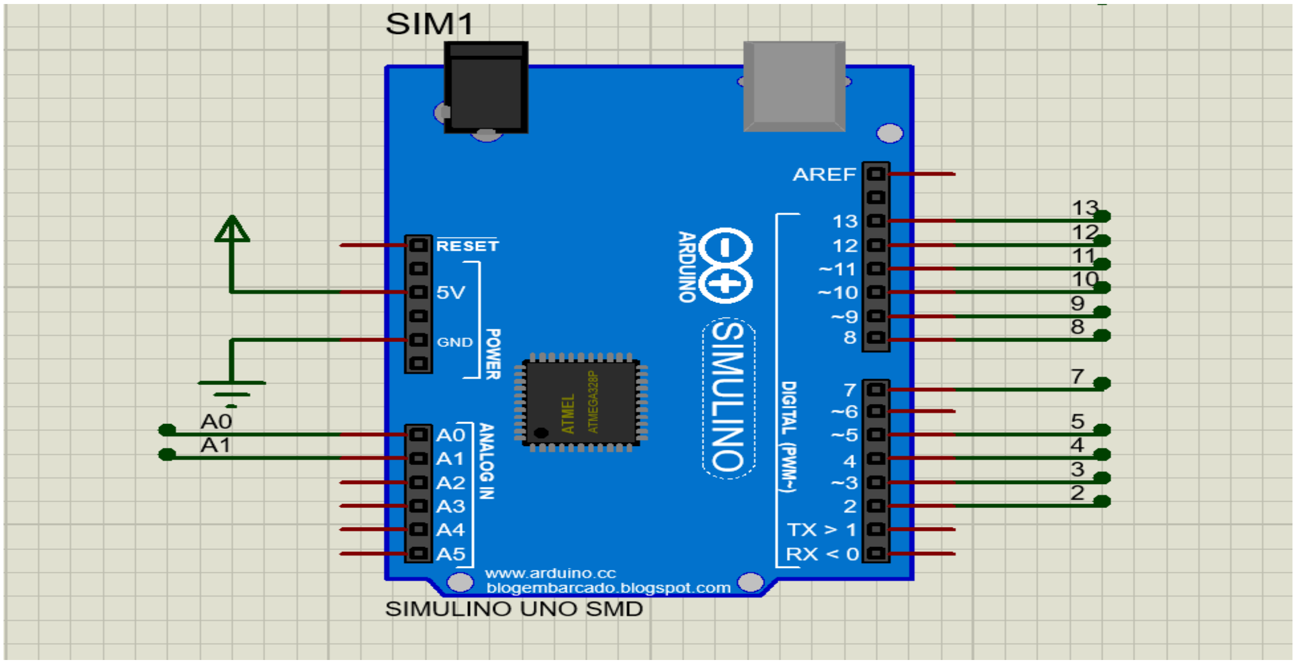

Figure 2: Schematic of Arduino uno block

Arduino Uno microcontroller board is tasked with receiving signals acquired from DHT11, HC-SR04 sensor modules, then analyzing the data for computation, and displaying specific numbers on the LCD screen. This is also the most important component that helps control the operation of the remaining components.

Arduino Uno can operate with external power source from 6 to 20V. If using lower voltage, typically 7V, the 5V pin may supply less than 5V, but it may cause unstable operation. When using power source above 12V, the voltage regulator will become hot and may get damaged, therefore it is recommended to use power source from 7 to 12V. When using an external power source, the supplied voltage should be within the specified range and not exceed the limits for the ATmega328P microcontroller on the board.

The energy pins are as follows:

• USB Cable: Arduino Uno R3 can be powered through the USB port by connecting it to a computer or any other USB power source.

• External DC Power: An external power source can be connected through the DC power jack on the Arduino Uno. The voltage of this source should be in the range of 7 to 12VDC.

You can use an AC-DC converter module or a battery pack.

• 5V Pin: You can also supply power directly to the 5V pin of Arduino Uno from a stable 5V source, such as a power module or a 5V USB source.

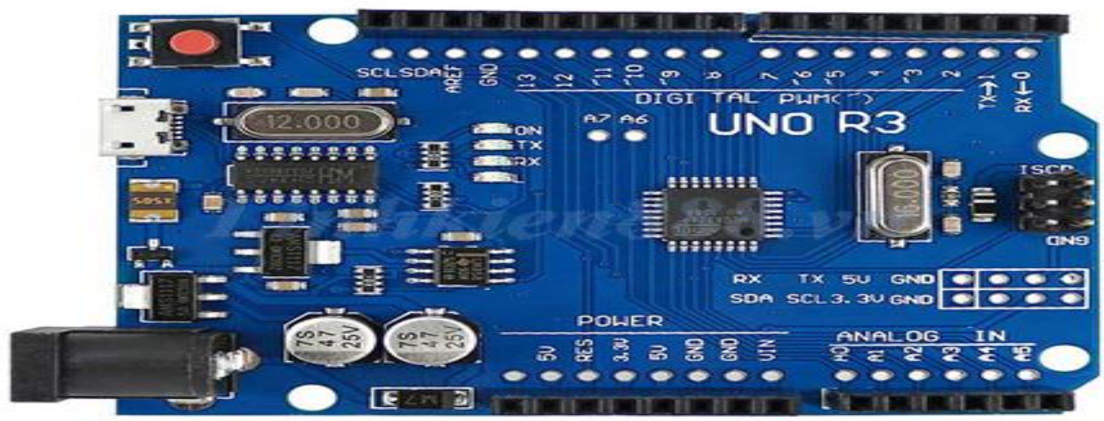

Figure 3: Arduino uno

Arduino Uno is one of the most popular boards in the Arduino product line. Launched in 2010, the Arduino Uno has become a versatile and powerful tool for beginners as well as professional electronics engineers. It's an open-source development board with a simple design, easy-to-use interface, and expandable capabilities. This board integrates an ATmega328P microcontroller and comes with a collection of ports and pins to connect sensors, displays, servo motors, and various other devices. Arduino Uno can be programmed through the Arduino Integrated Development Environment (IDE) using a programming language based on C/C++, allowing users to write code to control and interact with electronic components easily.

Table 1: Some key features of Arduino Uno include:

14 digital pins (including 6 PWM pins)
6 analog pins for reading values from analog sensors

USB connection for programming and communication with a computer
Power input and output pins
Reset button

Table 2: The basic technical specifications of the Arduino Uno board:

Microcontroller: ATmega328P
Operating Voltage: 5V
Recommended Input Voltage: 7-12V
Current Consumption through USB: 500mA
Maximum Output Current per I/O Pin: 20mA
Maximum Current through 3.3V Pin: 50mA
Flash Memory: 32 KB (ATmega328P, 0.5 KB used for bootloader)
SRAM: 2 KB (ATmega328P)
EEPROM: 1 KB (ATmega328P)
Clock Speed: 16 MHz
Number of Digital I/O Pins: 14 (including 6 PWM pins)
Number of Analog Input Pins: 6
Current through each 3.3V Pin: 50mA
Dimensions: Length 68.6 mm, Width 53.4 mm

Arduino's A0 and A1 pins are respectively connected to the TX and RX pins of the HC-06 Bluetooth module. The DATA pin of the DHT11 temperature and humidity sensor module will be connected to pin 13 of the Arduino. The trig and echo pins of the first HC-SR04 distance measuring module will be connected to pins 8 and 7 on the Arduino. The trig and echo pins of the second HC-SR04 distance measuring module will be connected to pins 10 and 9 on the Arduino. Pins 12 and 11 of the Arduino will be connected to the RS and E pins of the LCD1602 respectively. Pins 5, 4, 3, 2 on the Arduino will be connected to D4, D5, D6, D7 pins on the LCD1602. Finally, the VDD pin of the Arduino will be connected to a 220V power source via a 5V adapter, and the GND pin will be connected to ground.

2.3 HC-SR04 sensor

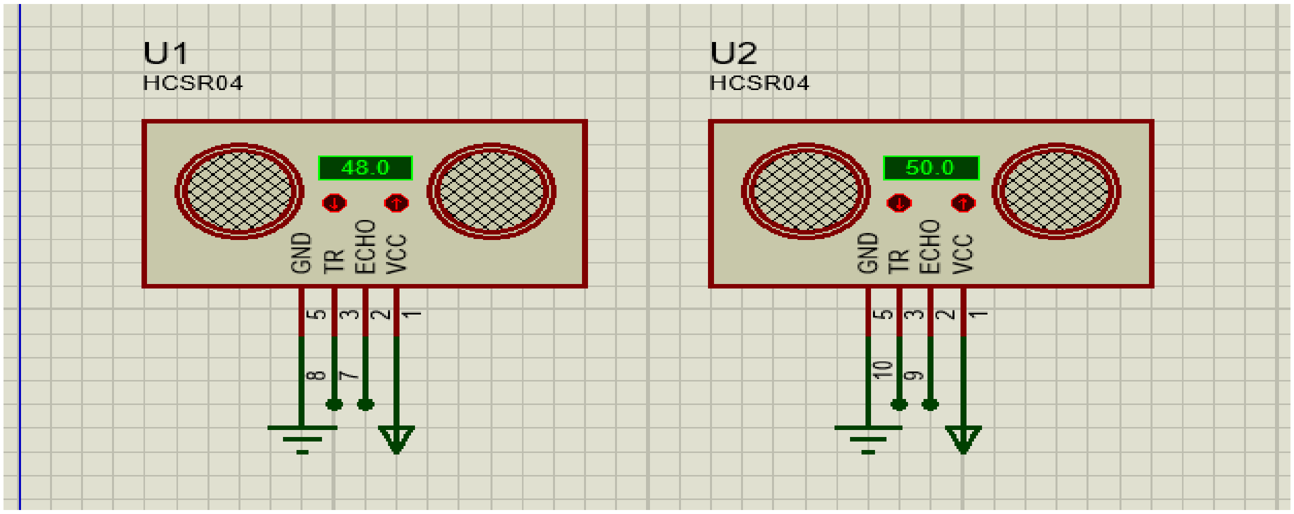

Figure 4: Schematic of HC-SR04

This project will use 2 ultrasonic distance measuring sensors HC-SR04, which are separately connected to each other and indirectly connected to the Arduino board via wires, to measure the distance between 2 objects in multiple directions without being limited to any specific direction. The modules will emit signal pulses from the trig pin and receive feedback signals from the echo pin, then transmit this information to the Arduino for processing.

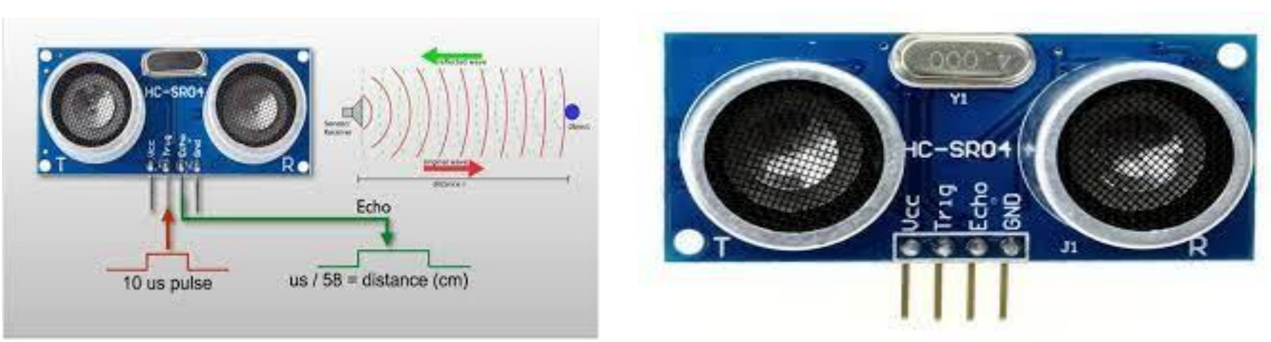

Figure 5: HC-SR04

HC-SR04 is a very popular ultrasonic distance sensor used in many electronic applications. This sensor uses ultrasonic waves to measure the distance from the sensor to the object it is pointing towards. The HC-SR04 consists of two main modules: an ultrasonic transmitter and an ultrasonic receiver. It operates by sending an ultrasonic pulse from the transmitter, then measuring the time for the ultrasonic wave to travel from the sensor to the object and back. Based on this time and the speed of sound in the air, the sensor can calculate the distance from itself to the target object. HC-SR04 is easy to use, cost-effective, and provides fairly accurate distance measurement results. This sensor is commonly used in robot projects, automatic distance measurement systems, or in IoT applications such as water level measurement, obstacle avoidance, or distance tracking.

Table 3: The ultrasonic distance sensor HC-SR04 has the following characteristics:

Operating principle: Uses ultrasonic waves to measure the distance from the sensor to the object. It consists of a transmitter and a receiver for ultrasonic waves.
Measurement range: It can typically measure distances from about 2cm to 4m, depending on environmental conditions.
Accuracy: Provides fairly accurate distance measurement results with low margin of error.
Simple and easy to use: It has a straightforward structure, easy to connect and use with microcontrollers such as Arduino or Raspberry Pi.
Communication: Uses ultrasonic signals to transmit and receive information through Trigger and Echo pins.
Fast response time: Allows for quick distance measurements, usually within a few milliseconds.
Wide applications: Commonly used in electronic projects like robots, automatic distance measurement systems, or in IoT applications such as water level measurement, obstacle avoidance, distance tracking, and various other applications.

Table 4: The technical specifications of the HC-SR04 ultrasonic distance sensor typically include:

Measurement Range: The minimum to maximum distance range that the sensor can measure, for example: from 2cm to 400cm.

Operating Voltage: The range of voltage within which the sensor can safely operate, usually around 5VDC.
Current Consumption: The current drawn by the sensor when in operation, typically a few milliamperes.
Communication: Connection pins like Trigger, Echo for transmitting and receiving ultrasonic signals.
Operating Frequency: The frequency at which the ultrasonic waves operate, typically in the tens of kHz.
Resolution: The precision of distance measurement, often dependent on the magnitude of the measuring unit.
Accuracy: The sensor's precision in measuring distances, usually determined by its margin of error.
Size and Weight: Physical dimensions and weight of the sensor.

The HC-SR04 provides fairly accurate distance measurements with low margin of error, aiding in determining distances reliably. It has a simple structure and is easily connectable to controllers such as Arduino, Raspberry Pi, ESP8266, etc. It can measure distances from a few centimeters to several meters, depending on environmental conditions, making it versatile for various applications. The sensor responds quickly, allowing for precise and rapid distance measurements. It's suitable for a wide range of applications, from robotics and automatic distance measuring systems to IoT and other creative electronic projects.

2.4 DHT11 Sensor

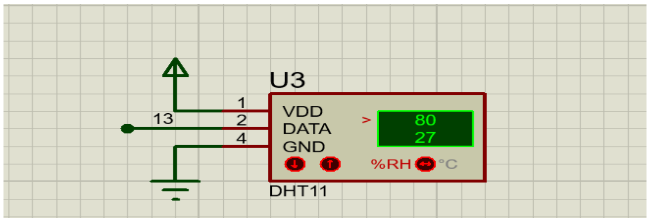

Figure 6: Schematic of DHT11

The DHT11 functions by receiving measured data from the DATA pin to gather temperature and humidity readings, then sending signals back to the main Arduino board to process tasks, calculate real-time data, and display them on an LCD.

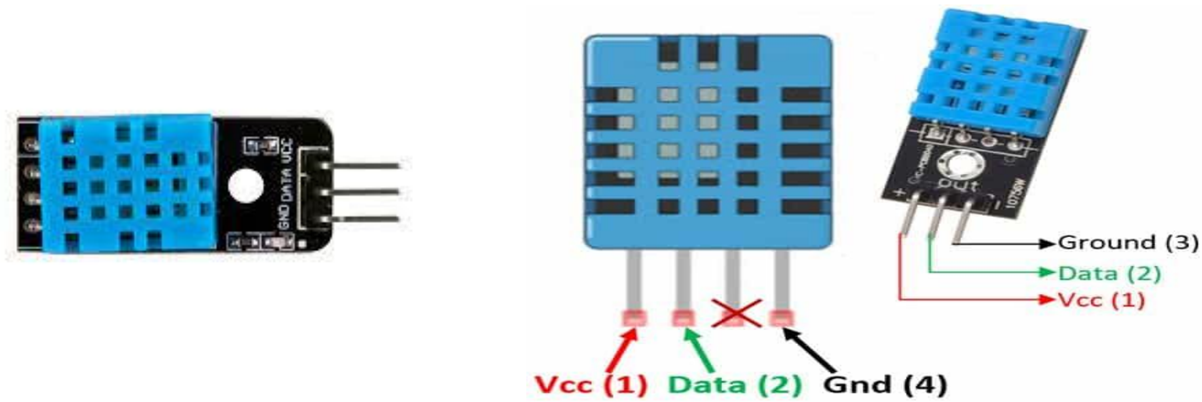

Figure 7: DHT11

The DHT11 is a popular digital temperature and humidity sensor widely used in various electronic applications. It has the capability to provide information about the temperature and humidity of the surrounding environment in an easy and efficient manner. The DHT11 sensor is compactly designed with three connecting pins: one for data transmission (DATA), one for positive power supply (VCC), and one for ground (GND). The measured data is transmitted as digital signals, making it convenient to read and process the information. With the ability to measure temperatures ranging from 0°C to 50°C and humidity from 20% to 90%, the DHT11 is commonly employed in IoT projects, weather sensing devices, smart home automation systems, or environmental monitoring applications. The DHT11 is cost-effective, user-friendly, and provides relatively accurate measurement results, making it a popular choice for simple to moderately complex electronic projects. However, it does have limitations regarding high humidity environments or requirements for high accuracy across a wider temperature range.

Table 5: The temperature and humidity sensor DHT11 possesses the following characteristics:

Function: Measures and provides information about the temperature and humidity of the surrounding environment.
Measurement Range: Temperature ranges from 0°C to 50°C and humidity from 20% to 90%, providing information within this range.
Communication: Transmits measured data as digital signals through the DATA connection pin.
Simple Connectivity: Consists of three basic connection pins: DATA, VCC (positive power), and GND (ground).
Versatile Applications: Commonly used in IoT projects, smart home automation systems, weather sensing devices, and environmental monitoring applications.
Low Cost and User-Friendly: Affordable and popular in simple to moderately complex electronic projects.
Relatively Accurate: Provides measurement information with relatively stable accuracy, suitable for many basic applications. However, it may have lower accuracy in high humidity environments or across wider temperature ranges.

Table 6: The technical specifications of the DHT11 temperature and humidity sensor typically include:

Temperature Range: From 0°C to 50°C with an accuracy of approximately ±2°C.
Humidity Range: From 20% to 90% with an accuracy of around ±5%.
Resolution: 8-bit for both temperature and humidity.
Power Supply: Usually 3.3V or 5VDC.
Current Consumption: Low, typically below 2.5mA during operation.
Communication: Digital data transmission through the DATA connection pin.
Measurement Frequency: Configurable to measure data every 1 or 2 seconds.
Physical Size: Compact dimensions, convenient for integration into projects.

This project utilizes the DHT11 because it provides an easy and accurate means to measure temperature and humidity levels in the environment. It delivers relatively stable and reasonably accurate information, suitable for various basic applications. With its simple structure, it seamlessly connects to microcontrollers such as Arduino, Raspberry Pi, ESP8266, and others. It is suitable for IoT projects, smart home automation systems, or weather sensing devices. Its affordability makes it fitting for projects with limited budgets. It can be employed in a wide range of applications, from environmental monitoring, habitat conditions surveillance, to control mechanisms in automated systems.

2.5 Display

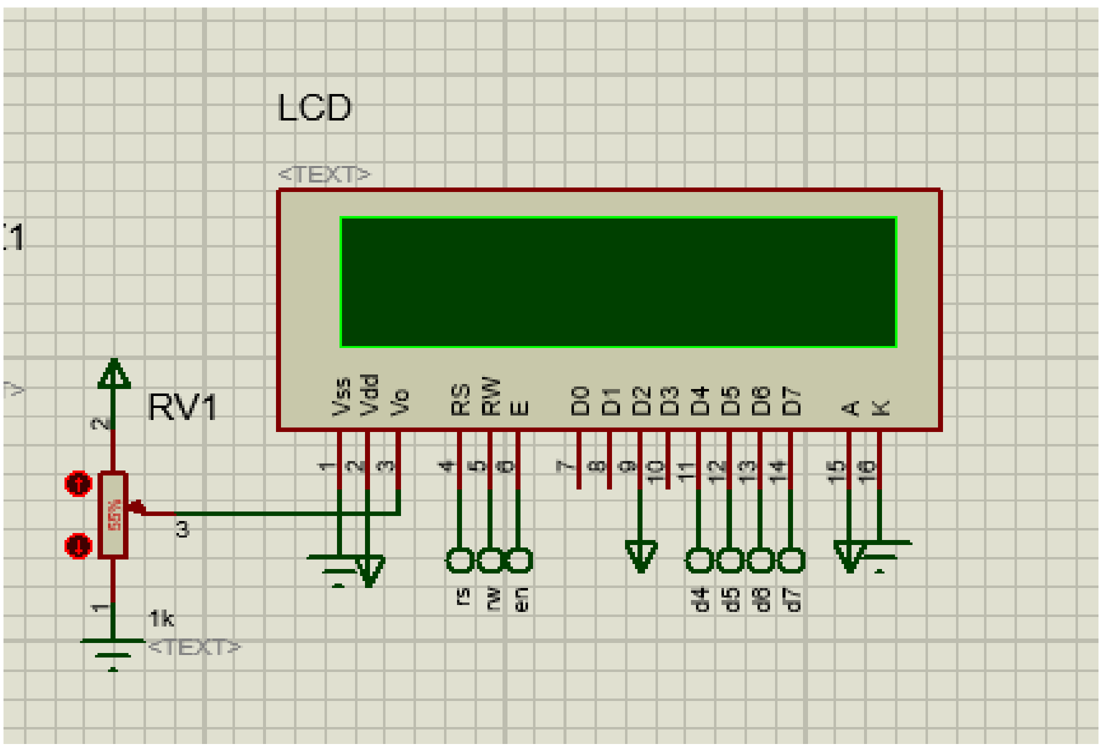

Figure 8: Schematic of Display

The display module uses the LCD1602 screen to showcase the data collected after being calculated by the controller unit, displaying the following parameters on the screen: the measured distance between 2 HC-SR04 modules, and the temperature and humidity from the DHT11 module.

Figure 9: LCD1602

LCD1602 is a type of LCD (Liquid Crystal Display) display with a size of 16x2 characters, meaning it can display 16 characters on 1 line and up to 2 lines. This display is commonly used in electronics projects such as Arduino, Raspberry Pi, and other microcontrollers to display information about variables, measurement results, notifications, and messages. The LCD1602 display uses LED backlight technology for illumination and uses a 16-pin connector to connect with other microcontrollers. It also has a built-in controller to display characters, commas, special characters, and numeric characters. The LCD1602 display can be adjusted for contrast and brightness, allowing users to fine-tune the display to meet their needs. The applications of the LCD1602 display are diverse, from displaying information in household electronic devices to displaying information in medical and industrial equipment. Additionally, the LCD1602 display is used in educational and research projects by students, teachers, and scientists.

Table 7:

Screen size: 16x2 characters
Operating voltage: 5V DC
Brightness: Adjustable
Contrast: Adjustable
Interface: 16-pin connector
Maximum power consumption: 1mA
Minimum input voltage: 4.7V DC

Maximum input resistance: 1MΩ
Operating temperature: -20 to 70 degrees Celsius
Operating humidity: 10% to 80% non-condensing.

Table 8:

LEG	CHARACTERISTIC	DESCRIBE	VALUE
1	VSS	GND	0V
2	VCC		5V
3	V0	Contrast	
4	RS	register selection	RS=0 (low) selects instruction register RS=1 (high) selects data register
5	R/W	Select data read/write register	R/W=0 write register R/W=1 read register
6	E	Enable	
7	DB0	Data pins	8 bit: DB0DB7
8	DB1		
9	DB2		
10	DB3		
11	DB4		
12	DB5		
13	DB6		
14	DB7		
15	A	LED background anode	0V to 5V
16	K	LED background cathode	0V

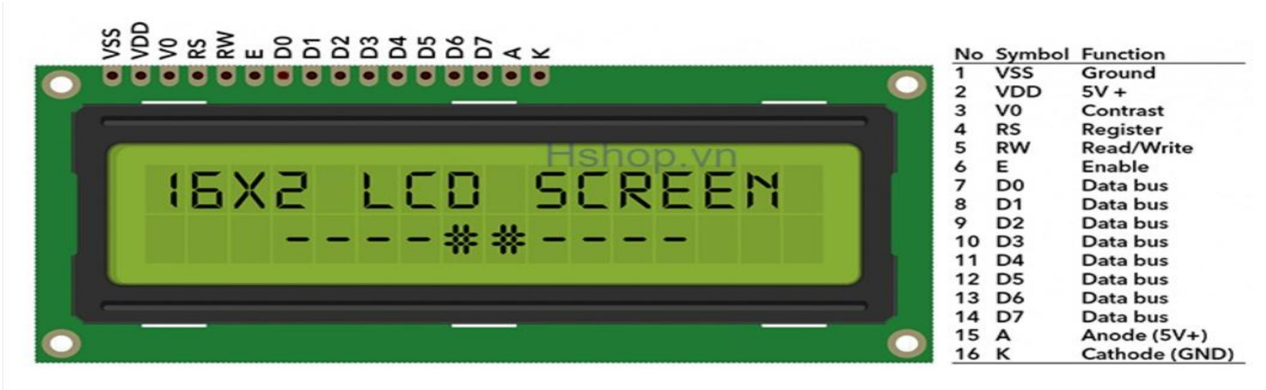

Figure 10: Schematic diagram of LCD1602

Figure 11: Variable resistor

Variable resistor is a device that has a pure resistance that can be adjusted as desired. They can be used in electronic circuits to adjust the operation of the circuit. Variable resistors are usually connected to other components in an electronic circuit and have three terminals: two terminals connected to the two ends of the resistor and the remaining terminal connected to the wiper (or rotary arm).

Pin number 1, 5 and 16 will be connected to ground. Pins number 2 and 15 will be connected to the positive power supply. Pin number 3 will be connected to a variable resistor to adjust the brightness of the display screen. Pins 4 and 6 will be connected to pins 12 and 11 of Arduino uno, respectively. Pins 11, 12, 13, and 14 (D4, D5, D6, D7) will be connected to pins 5, 4, 3, and 2 of Arduino no, respectively.

2.6 Blutooth HC-06

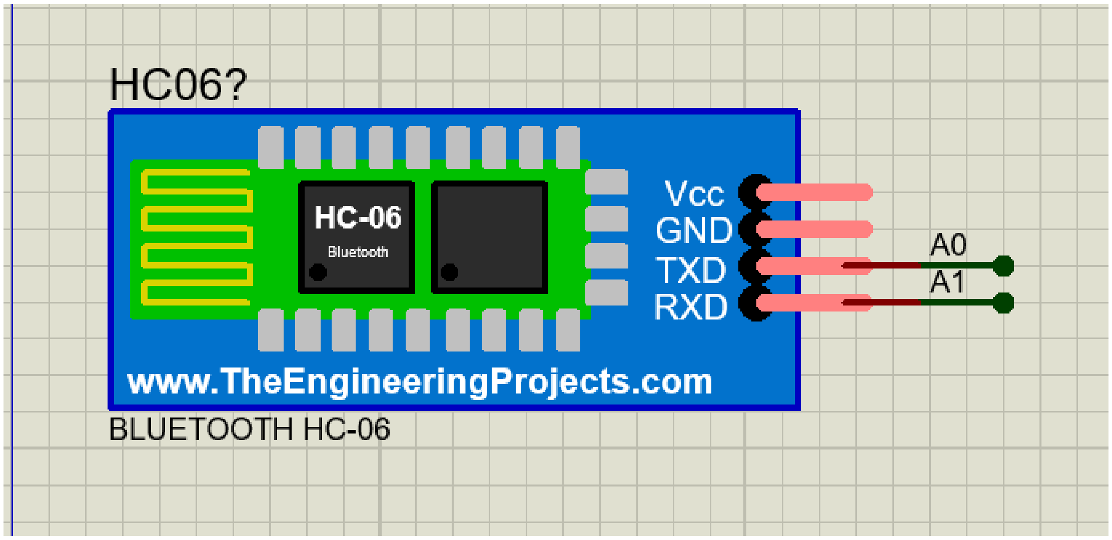

Figure 12: Schematic of HC-06

The HC-06 Bluetooth module will connect to other mobile devices via Bluetooth signals to transmit computed data from Arduino. This allows users to view the data online anytime, anywhere through a third-party application.

Figure 13: HC-06

HC-06 is one of the popular Bluetooth modules used in IoT applications, electronic projects, and wireless systems. It is designed to provide wireless connectivity via the Bluetooth protocol with other devices such as mobile phones, tablets, or other Bluetooth modules. Key features of the HC-06 include its ability to operate in Slave mode within the Bluetooth system, enabling easy connections with compatible Bluetooth devices. It commonly uses the Bluetooth 2.0 protocol and supports the UART (Universal Asynchronous Receiver-

Transmitter) connection standard for communication with microcontrollers like Arduino or Raspberry Pi. HC-06 is often used in projects requiring wireless connectivity and data transmission, such as remote control projects, transmitting data from sensors to mobile phones or computers, and in IoT network applications for data collection and transmission. It offers flexibility, stability, and ease of use, making it a popular choice in the electronics and programming community.

The main characteristics of the HC-06 Bluetooth module include:

- Operating Mode: Typically operates in Slave mode within the Bluetooth system, capable of connecting to Master devices.
- Bluetooth Version: Uses the Bluetooth 2.0 protocol, enabling connections with compatible Bluetooth devices.
- Communication: Supports UART (Universal Asynchronous Receiver-Transmitter) communication standard for interfacing with controllers like Arduino, Raspberry Pi, ESP8266, etc.
- Easy Connectivity: Allows wireless connection with devices such as mobile phones, tablets, or other Bluetooth modules.
- Common Applications: Often used in IoT projects, remote control applications, wireless data transmission, and various IoT network applications.
- Flexibility and Stability: Provides high flexibility, stable operation, and ease of use in designing wireless systems.

Table 9: The main technical specifications of the HC-06 Bluetooth module typically include:

Bluetooth Version: It uses the Bluetooth 2.0 protocol.
Operating Mode: Operates in Slave mode within the Bluetooth system.
Operating Range: Communication range usually spans from a few meters to several tens of meters, depending on the environment.
Operating Frequency: Typically operates at the 2.4GHz frequency.
Communication Interface: Supports UART (Universal Asynchronous Receiver-

Transmitter) communication for interfacing with controllers.
Operating Voltage: Usually 3.3V or 5VDC.
Current Consumption: Low power consumption during operation, typically below 40mA.
Compatibility: Capable of connecting and interfacing with other Bluetooth devices such as mobile phones, tablets, or other Bluetooth modules

.

The HC-06 provides wireless connectivity through the Bluetooth protocol, enabling data transmission between devices without the need for physical connection cables. It has a simple and user-friendly structure, utilizing the widely used UART communication, making it convenient to connect with controllers like Arduino, Raspberry Pi, ESP8266. It is widely used and compatible with many other Bluetooth devices, including mobile phones, tablets, and other Bluetooth modules. It efficiently communicates over distances ranging from a few meters to several tens of meters, depending on environmental conditions. Suitable for IoT projects, remote control, wireless data transmission, and various IoT network applications. With reasonable pricing, it is a popular choice for projects with limited budgets but requiring wireless connectivity.

CHAPTER 3. PROGRAMMING

3.1 Flowchart

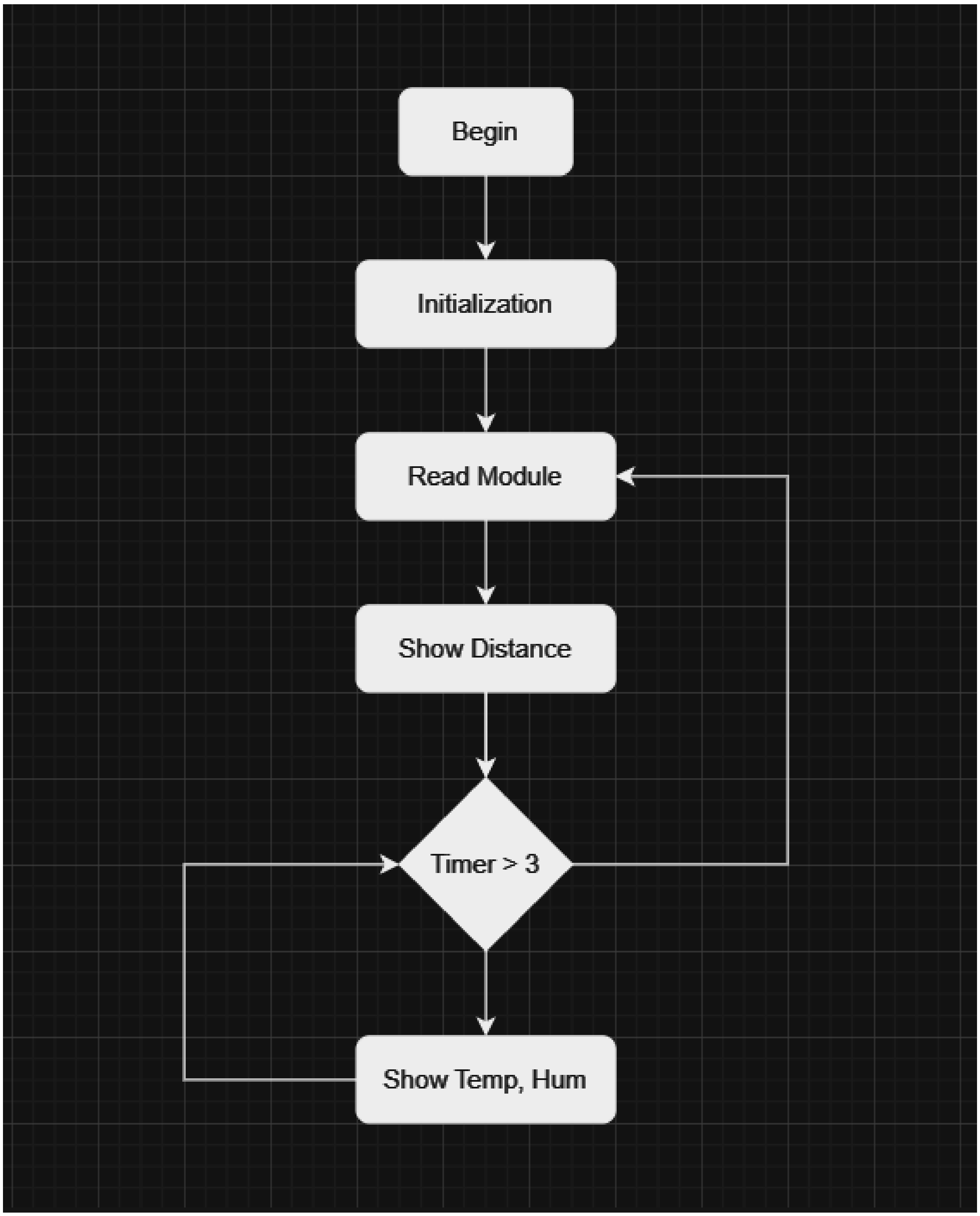

Figure 14: Flow chart

3.2 Explanation

First, the system will start initialization and initialize the LCD display task. Then, the system will read the sensor. Then, the distance data will be displayed on the LCD. Next, one of the following two cases will occur:

Case 1: If the display time for the two distances is less than 3 seconds, it will continue reading data for display until the countdown timer reaches the required time.

Case 2: If the display time for the two distances is sufficient or exceeds 3 seconds, it will switch to displaying the temperature and humidity. Then, it will continue checking if the timer has reached or exceeded 3 seconds; if not, it will revert back to displaying the distance data, and this process will repeat continuously.

3.3 Software / Library / Installation

3.3.1 Software

1. **Proteus 8 professional**

Proteus 8 Professional software is a professional electronic circuit simulation and design software. It allows users to simulate and design electronic circuits, verify the correctness of the circuit, and display the circuit graphics on the computer. Proteus software includes many components such as schematic diagrams, microcontroller simulation software, Analog and Digital circuit simulation software, PCB drawing software and many other tools to help users create accurate and effective electronic circuits.

Proteus can help design and simulate complex circuits including electronic components such as microcontrollers, drives, sensors, power supplies, and many other types of components. Proteus also allows users to simulate input, output and control signals to check the correctness of the circuit before actual production.

Proteus 8 Professional is widely used in fields such as electronic design, telecommunications, automation control, measurement, education, and scientific research.

With convenient and efficient features, Proteus 8 Professional is an indispensable software for experts and engineers in electronic circuit design.

2. **Arduino IDE**

The Arduino IDE (Integrated Development Environment) is an open-source software that provides an integrated development environment for writing and uploading code to Arduino boards and other development boards that use AVR or ARM microcontrollers.

Designed to simplify programming and uploading code onto boards, the Arduino IDE offers a user-friendly interface allowing users to write program code in a simple and understandable C/C++ programming language.

Key features of the Arduino IDE include a source code editor with syntax highlighting, built-in libraries to facilitate programming, and the ability to upload programs to Arduino boards through the USB port.

Widely used within the programming community and electronic projects, the Arduino IDE is an essential tool in developing IoT (Internet of Things) applications, device control, and various applications related to microcontroller-based systems.

3.3.2 Library

1. **Library of Proteus 8 professional**

The library of Proteus 8 Professional includes commonly used electronic components in electronic circuit design such as transistors, ICs, resistors, capacitors, inductors, and many other types of components. It also includes popular microcontrollers such as 8051, PIC, AVR, ARM, and many other types of microcontrollers. In addition, Proteus 8 Professional allows users to create their own libraries by adding electronic components, microcontrollers, and other modules to the library. This helps users to reuse components and modules created previously in new electronic circuit designs. All components and microcontrollers in the library of Proteus 8 Professional are accurately simulated and can be used to design and test complex electronic circuits before production.

2. Library of Arduino IDE

The Arduino IDE provides a range of pre-integrated libraries that aid programmers in executing specific functions without the need to write code from scratch. These libraries offer functions and modules to perform particular tasks such as sensor control, interfacing with peripherals, and executing complex operations.

Some common libraries within the Arduino IDE include:

- Wire Library: Used for I2C communication, allowing Arduino to connect and communicate with sensors, LCD screens, EEPROM, and other devices.
- Serial Library: Used for interaction with the Serial port, facilitating sending and receiving data via the Serial port on the Arduino board.
- Servo Library: Allows easy control of servo motors.
- Ethernet Library: Used for connecting Arduino to the Internet through an Ethernet module, supporting IoT application programming.
- SPI Library: Supports SPI communication between Arduino and devices like Flash memory, SD cards, or TFT screens.
- LiquidCrystal Library: Used to control and display data on LCD screens.

These libraries help streamline development time, reduce code complexity, and facilitate efficient utilization of specific functionalities in electronic projects.

3.3.3 Simulate on Proteus

Step 1: Press the button  on the screen to starting the circuit

Step 2: On the LCD screen, 'DISTANCE 1' will be displayed on line 1, and 'DISTANCE 2' on line 2 to represent the distances measured by the 2 HC-SR04 modules. The distance display on the LCD can be adjusted by adjusting the increase/decrease button on the 2 modules.

Step 3: After 3 seconds, the LCD screen will automatically switch to displaying 'Temperature' on line 1 and 'Humidity' on line 2 to represent the temperature and humidity data obtained from the DHT11 module. These parameters can also be adjusted using the increase/decrease buttons on the module.

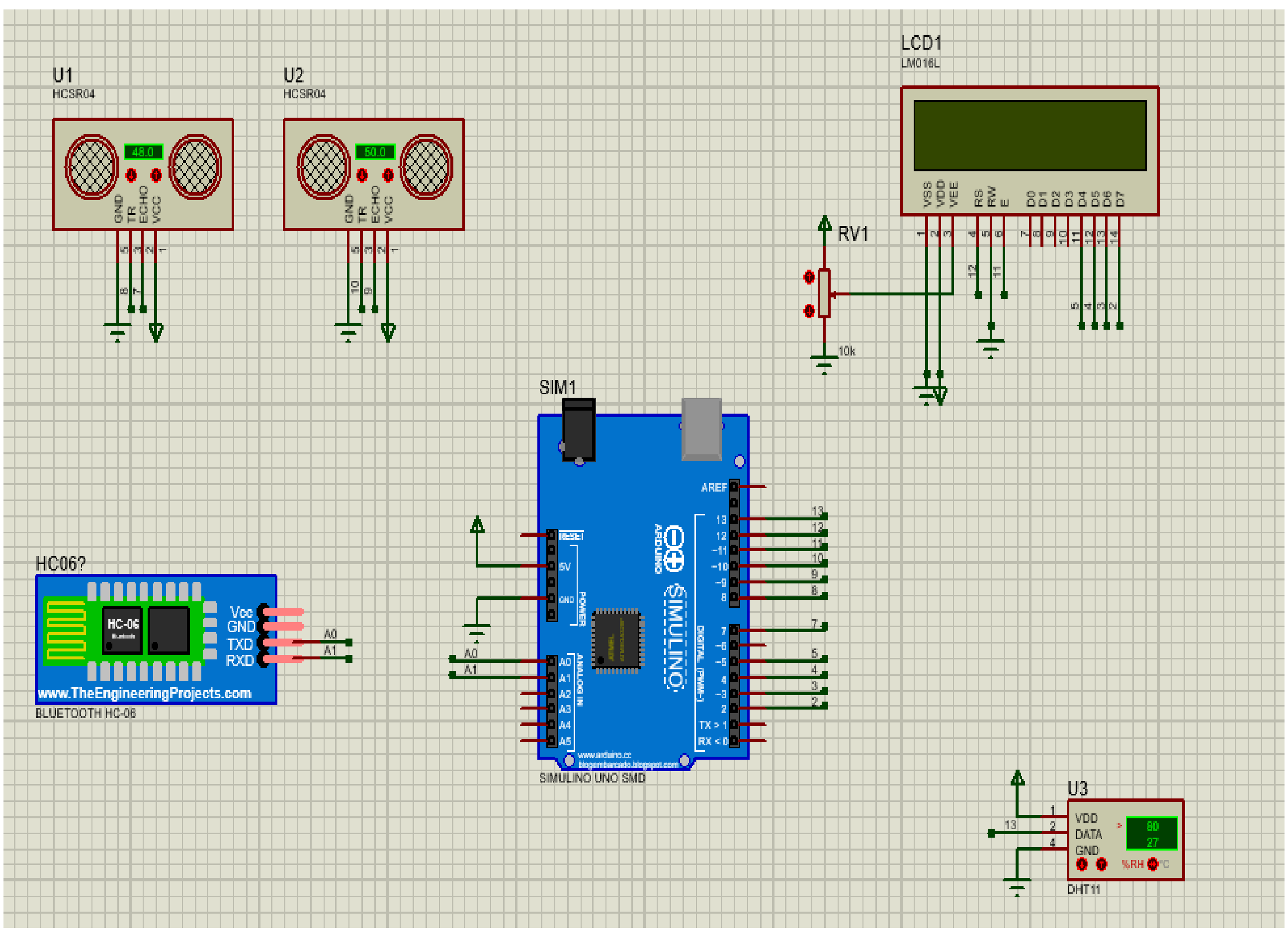

Figure 15: Before simulation

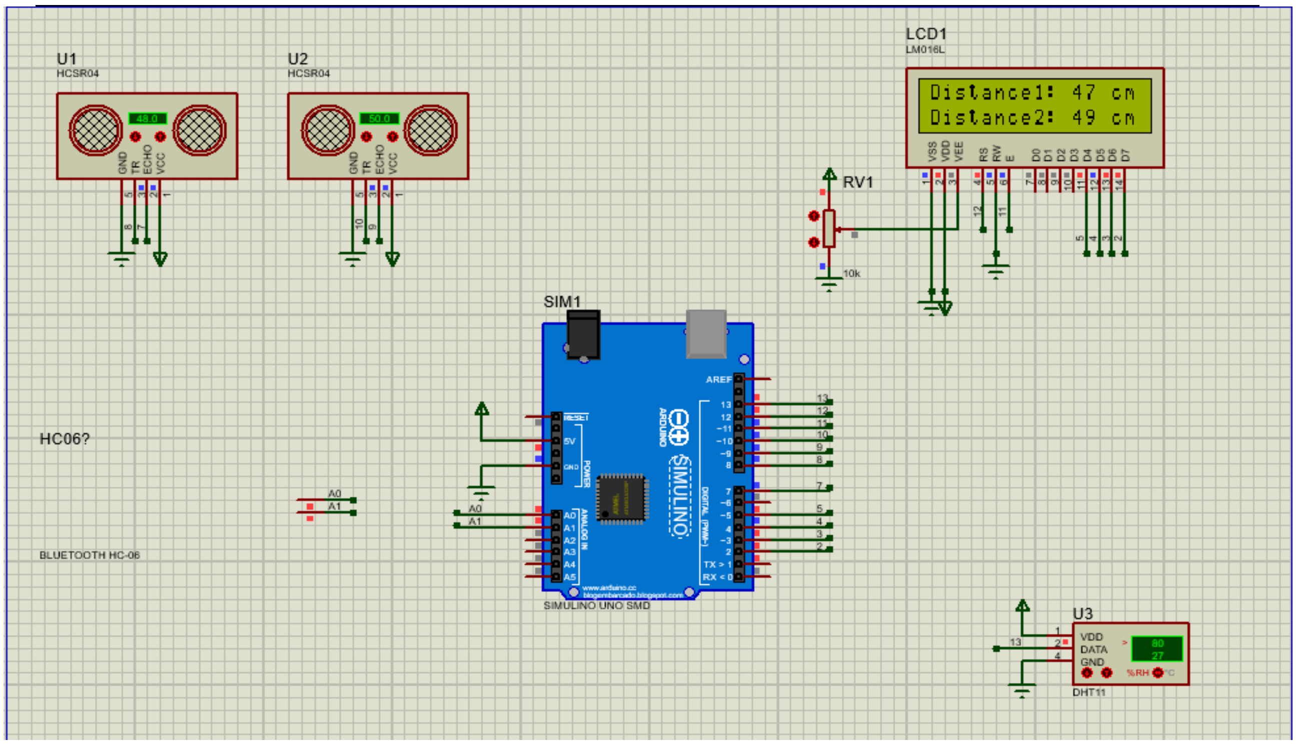

Figure 16: After simulation (1)

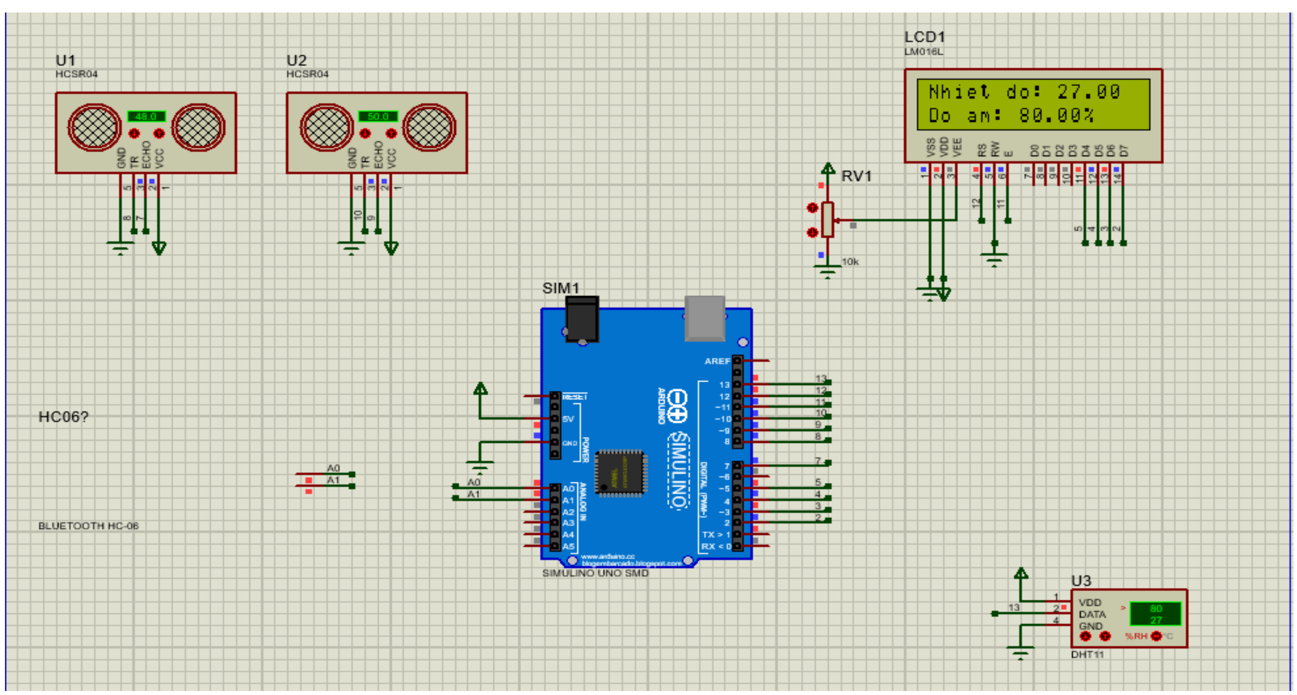

Figure 17: After simulation (2)

CHAPTER 4. IMPLEMENTAION

4.1 PCB Layout

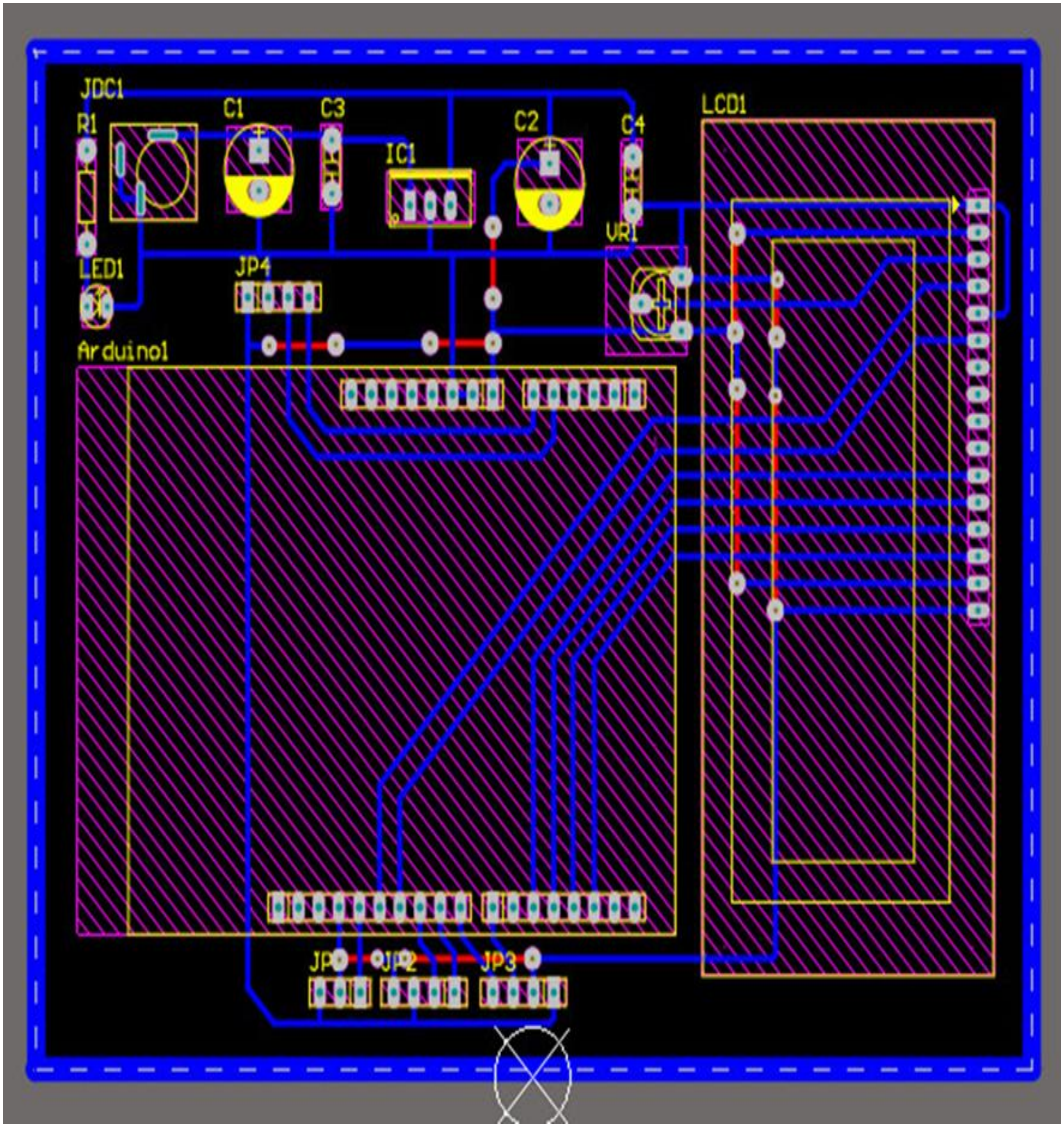

Figure 18: PCB Layout

First, open Proteus and create a new project. Select "Schematic Capture" from the toolbar to open the schematic editor. Design circuit by selecting components from the Proteus library and connecting them with wires. After completing the schematic, save and close the schematic editor. In the Proteus interface, choose "Layout Mode" from the toolbar to open the PCB editor. From the "Select Component" window, drag and drop the components from the schematic onto the PCB. Place the components in the desired positions on the PCB. Connect the pins of the components by drawing wires or tracks on the PCB. Use the "Route Mode" tool to draw the connection lines. If needed, add different layers for routing, power planes, and other purposes on the PCB. Check and edit PCB layout to ensure there are no technical errors such as short circuits or component conflicts. Save the PCB layout and select "Auto Router" if want Proteus to automatically route the connections. After completing the PCB layout design, can use the "3D Viewer" function to preview the final product. Finally, print or publish your PCB layout to prepare for the manufacturing process.

4.2 Prototype

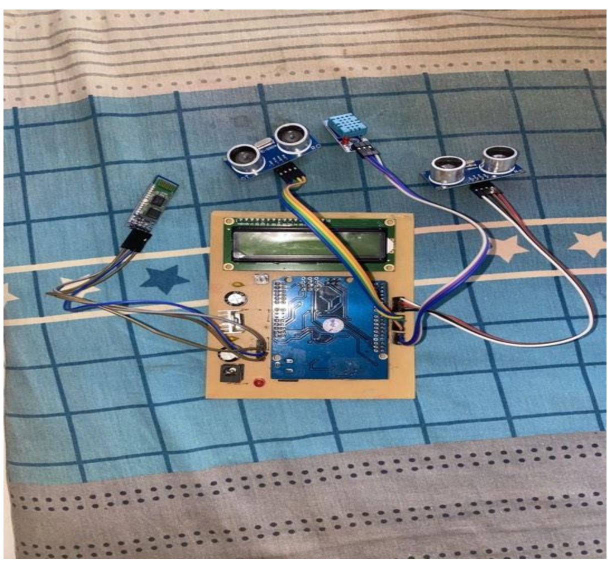

Figure 19: The front of circuit

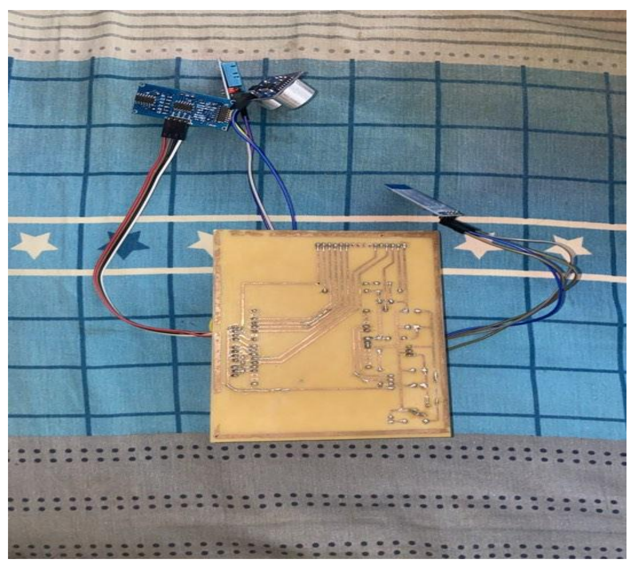

Figure 20: The back of circuit

4.3 Testing

4.3.1 Set up

First, we use a socket to connect to the 220V AC power supply through adapter 5V, and the system will start operating. The LCD screen will display "Distance 1" and "Distance 2", which means to represent the distances measured by the 2 HC-SR04 modules. The distance display on the LCD can be adjusted by adjusting the increase/decrease button on the 2 modules.

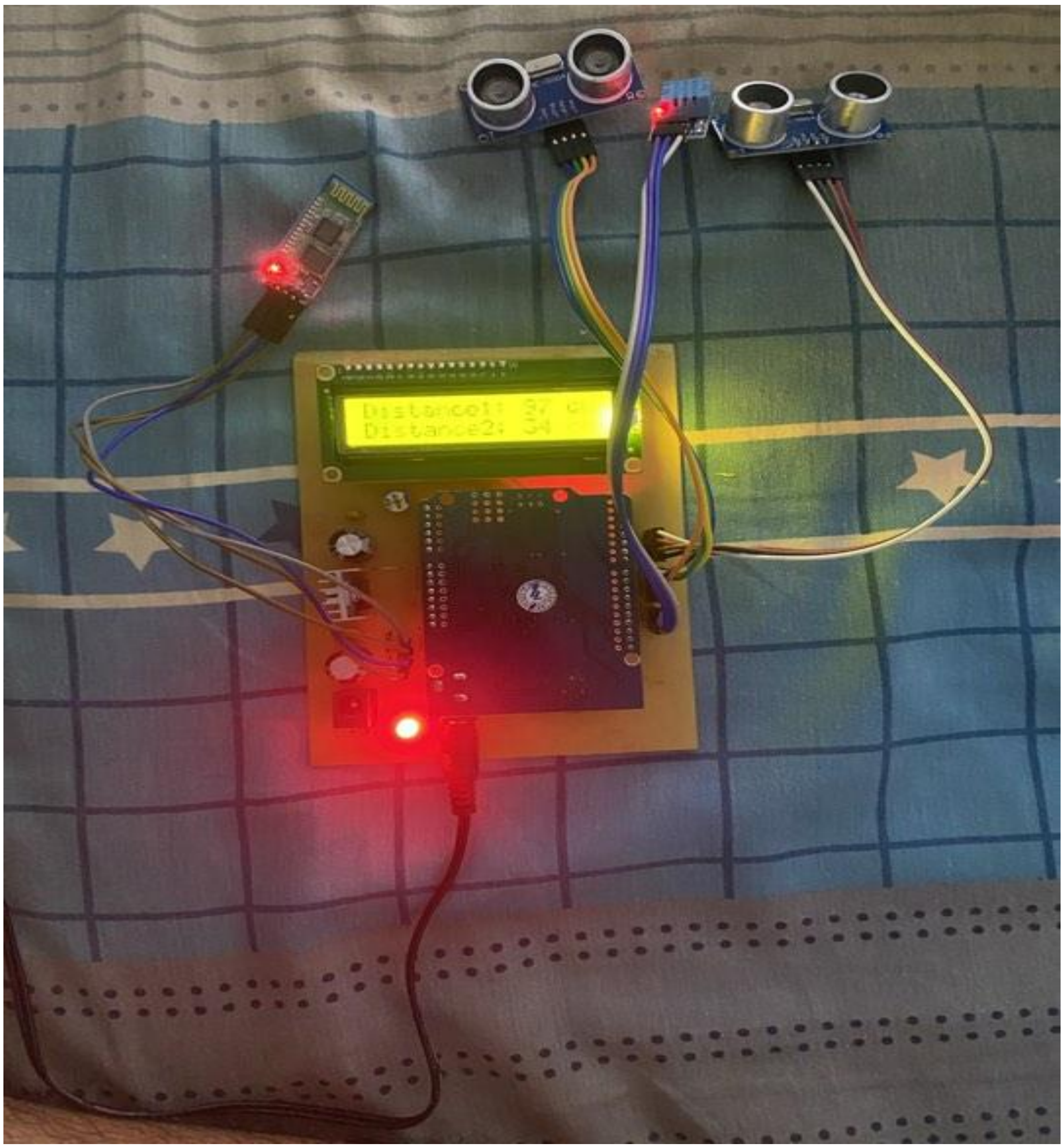

Figure 21: Starting the circuit

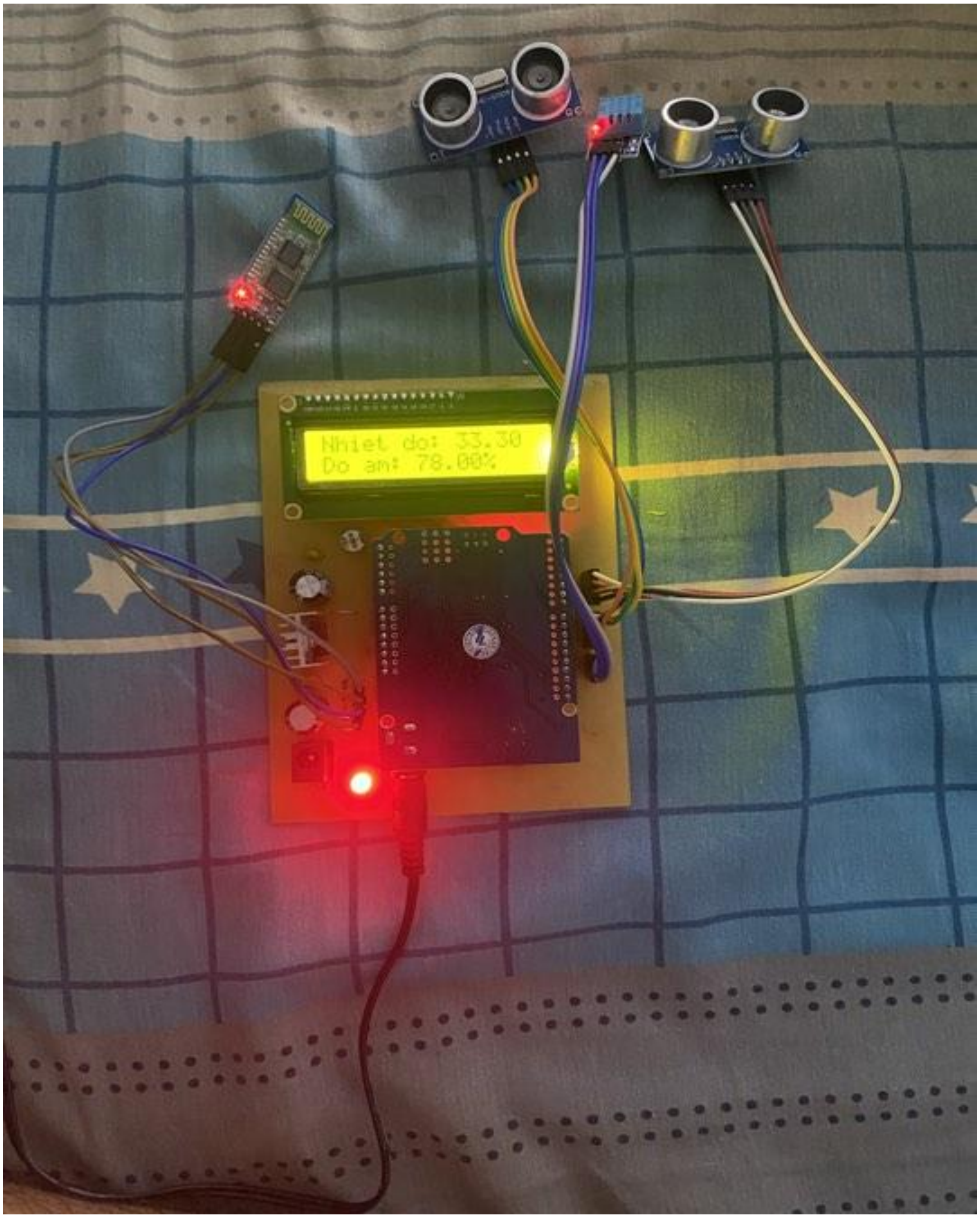

Figure 22: After 3 seconds

After 3 seconds, the LCD screen will automatically switch to displaying 'Temperature' on line 1 and 'Humidity' on line 2 to represent the temperature and humidity data obtained from the DHT11 module. These parameters can also be adjusted using the increase/decrease buttons on the module.

4.3.2 Debug / Disadvantages

Debug

To successfully complete this project, I had to address several related issues that affected the performance of this electrical circuit. One major issue I encountered was interference between the two ultrasonic HC-SR04 distance measurement modules when coding for this program. I resolved this problem by separating their functions and no longer merging their collected data. Another challenge arose while using the HC-06 Bluetooth module; it seemed incompatible with iOS-run phones. Therefore, I couldn't locate the module's address, potentially due to higher security on iOS preventing the detection of unknown connection addresses. To resolve this, I utilized a different device running on the Android operating system. These are all significant challenges I faced while working on this project.

Disadvantages

The limitation of this system is the inability to store the received data sent via Bluetooth signals from the HC-06 module onto an external drive or SD memory for users to view the history of computed results generated by this system during its operation. The main reason for this issue is due to my limited knowledge, constrained thinking, and my own level of expertise, thus preventing me from fully completing all the assigned tasks to design the system.

CHAPTER 5. CONCLUSION

5.1 Conclusion

Overall, in my own assessment, this system meets nearly all of the requirements set. The system operates quite stably with little to no significant deviation in calculations or minor errors during long-term operation. The functions within the electrical circuit all perform their tasks well, and the functionality of each block operates effectively. However, mistakes are inevitable. Through this project, I have gained additional experience, and from there, I can apply the knowledge gained to perform better in future projects.

5.2 Development

. The first thing I want to develop for this system is to create a larger measurement range for a larger environment. Currently, the system can only operate within a radius of a few meters. I want to extend the measurement range to the size of a classroom or even larger to ensure high accuracy and avoid discrepancies between the values displayed on the LCD screen and the Bluetooth data sent to the user. My goal is to minimize the delay when the system receives a signal so that the message can be immediately sent to the user, and the value must be close to the displayed value on the LCD screen to ensure user safety. Finally, an important feature that cannot be overlooked is the system's ability to record operational events for storage in external memory (SD) for the user.

REFERENCES

Vietnamese:

Trương, V. D. (2017). "Lập trình Arduino cơ bản." NXB Thông tin và Truyền thông.

Trần, M. H. (2018). "Arduino - Từ cơ bản đến nâng cao." Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh.

Lê, Q. H. (2016). "Sáng tạo cùng Arduino." NXB Lao Động Xã Hội.

English:

Banzi, M. (2011). "Getting Started with Arduino." Maker Media, Inc.

Monk, S. (2016). "Programming Arduino: Getting Started with Sketches." McGraw-Hill Education TAB.

Scherz, P., & Monk, S. (2013). "Practical Electronics for Inventors." McGraw-Hill Education.

APPENDIX

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <DHT.h>

SoftwareSerial BTSerial(A0, A1);

DHT dht(13, DHT11); // Khai báo cảm biến DHT11

const int trig1 = 8; // Chân trig của HC-SR04 #1
const int echo1 = 7; // Chân echo của HC-SR04 #1
const int trig2 = 10; // Chân trig của HC-SR04 #2
const int echo2 = 9; // Chân echo của HC-SR04 #2

LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Khai báo màn hình LCD với các chân kết nối
int currentDisplay = 0; // Biến chọn hiển thị
unsigned long displayTimer = 0; // Biến thời gian cho việc chuyển đổi hiển thị

void setup() {
  Serial.begin(9600); // Giao tiếp Serial với baudrate 9600
  BTSerial.begin(9600); // Giao tiếp qua module HC-06
  pinMode(trig1, OUTPUT); // Chân trig #1 sẽ phát tín hiệu
  pinMode(echo1, INPUT); // Chân echo #1 sẽ nhận tín hiệu
  pinMode(trig2, OUTPUT); // Chân trig #2 sẽ phát tín hiệu
  pinMode(echo2, INPUT); // Chân echo #2 sẽ nhận tín hiệu
  dht.begin(); // Khởi động cảm biến DHT11
  lcd.begin(16, 2); // Khởi động màn hình LCD 16x2
  lcd.setCursor(0, 0);
  lcd.print("Distance1: ");
  lcd.setCursor(0, 1);
  lcd.print("Distance2: ");
}


void loop() {
```

```
unsigned long duration1; // Biến đo thời gian từ HC-SR04 #1
int distance1; // Biến lưu khoảng cách từ HC-SR04 #1
unsigned long duration2; // Biến đo thời gian từ HC-SR04 #2
int distance2; // Biến lưu khoảng cách từ HC-SR04 #2
float humidity; // Biến lưu độ ẩm
float temperature; // Biến lưu nhiệt độ

/* Phát xung từ chân trig #1 */
digitalWrite(trig1, LOW); // Tắt chân trig #1
delayMicroseconds(2);
digitalWrite(trig1, HIGH); // Phát xung từ chân trig #1
delayMicroseconds(10); // Xung có độ dài 10 microSeconds
digitalWrite(trig1, LOW); // Tắt chân trig #1
/* Tính toán thời gian và khoảng cách từ HC-SR04 #1 */
duration1 = pulseIn(echo1, HIGH);
distance1 = duration1 / 58.2;


/* Phát xung từ chân trig #2 */
digitalWrite(trig2, LOW); // Tắt chân trig #2
delayMicroseconds(2);
digitalWrite(trig2, HIGH); // Phát xung từ chân trig #2
delayMicroseconds(10); // Xung có độ dài 10 microSeconds
digitalWrite(trig2, LOW); // Tắt chân trig #2
/* Tính toán thời gian và khoảng cách từ HC-SR04 #2 */
duration2 = pulseIn(echo2, HIGH);
distance2 = duration2 / 58.2;

/* Đọc dữ liệu từ cảm biến DHT11 */
humidity = dht.readHumidity(); // Độ ẩm
temperature = dht.readTemperature(); // Nhiệt độ

// Kiểm tra xem đã đến thời gian chuyển đổi hiển thị chưa (mỗi 3 giây)
if (millis() - displayTimer > 3000) {
  currentDisplay = (currentDisplay + 1) % 2; // Chuyển đổi hiển thị
  displayTimer = millis(); // Cập nhật thời gian
}
```

```
// Gửi dữ liệu qua Bluetooth (tùy thuộc vào lựa chọn hiển thị)
if (currentDisplay == 0) {
  BTSerial.print("Distance1: ");
  BTSerial.print(distance1);
  BTSerial.println(" cm");
  BTSerial.print("Distance2: ");
  BTSerial.print(distance2);
  BTSerial.println(" cm");
} else {
  BTSerial.print("Nhiet do: ");
  BTSerial.print(temperature);
  BTSerial.print(" *C - Do am: ");
  BTSerial.print(humidity);
  BTSerial.print("%");
}

// Hiển thị dữ liệu tương ứng với lựa chọn hiển thị
lcd.clear();
lcd.setCursor(0, 0);
if (currentDisplay == 0) {
  lcd.print("Distance1: ");
  lcd.print(distance1);
  lcd.print(" cm");
  lcd.setCursor(0, 1);
  lcd.print("Distance2: ");
  lcd.print(distance2);
  lcd.print(" cm");
} else {
  lcd.print("Nhiet do: ");
  lcd.print(temperature);
  lcd.print(" *C");
  lcd.setCursor(0, 1);
  lcd.print("Do am: ");
  lcd.print(humidity);
  lcd.print("%");
}
```

```
  Serial.print("Khoang cach 1: ");
  Serial.print(distance1);
  Serial.println(" cm");
  delay(100);
  Serial.print("Khoang cach 2: ");
  Serial.print(distance2);
  Serial.println(" cm");
  delay(300);
  Serial.print("Nhiet do: ");
  Serial.print(temperature);
  Serial.println(" *C");
  delay(500);
  Serial.print("Do am: ");
  Serial.print(humidity);
  Serial.println("%");
  delay(700);
}
```