**Ramolatela Lebaea**

Temperature Measuring System Using A 1N4001 Diode, The Temperature Should Be Calculated And Saved In The EEPROM To Keep It Safe Using Arduino.

The objectives of this project are to design a temperature measuring system using a 1N4001 diode, calculate and save the temperature in the EEPROM for safety, present the temperature values on an LCD along with a 2-minute timer, and implement a pushbutton to read and display the last 15 temperature values on the LCD screen. Additionally, the project aims to gain more knowledge about the EEPROM() library. EEPROM, which stands for Electrically Erasable Programmable Read-Only Memory, is a non-volatile memory used for data retention across power cycles. It offers byte-level access and limited write/erase cycles, and is commonly used for storing configuration data in electronic devices. Although it has been partially supplanted by faster and higher-capacity NAND and NOR flash memory in some applications, EEPROM remains a valuable component in modern embedded systems.

# **APPARATUS**

### 1× 1N4001 diode

### 1× Push button

### 1× Breadboard

### 1× Arduino Uno R3

### 1× LCD

### 1× Soldering iron

### Connecting wires

# **CIRCUIT DIAGRAM USING TINKER**

# **Setup & Instruction**

# Connect the 1N4001 diode to the analog pin (A0) of the Arduino. Connect the pushbutton to digital pin 2 and configure it with a pull-up resistor. Connect the LCD to the Arduino using the specified pins (9, 8, 7, 6, 5, 4).

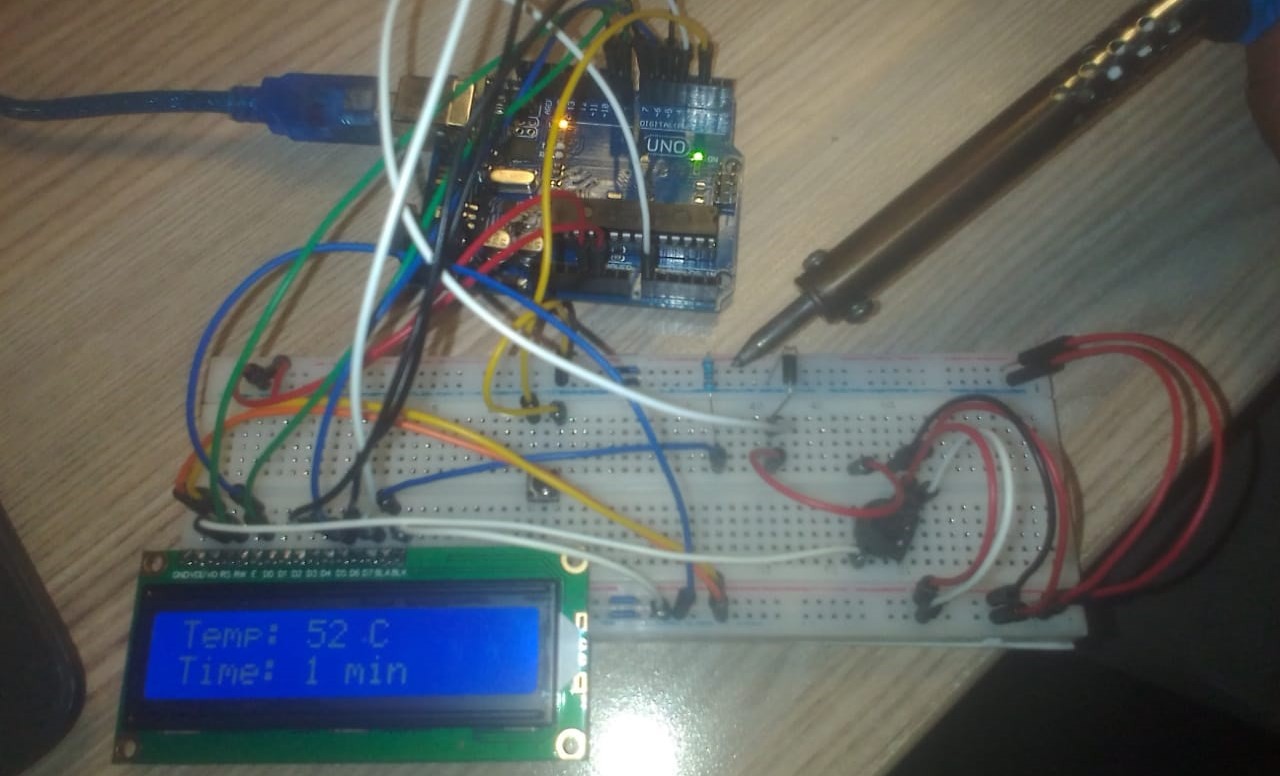
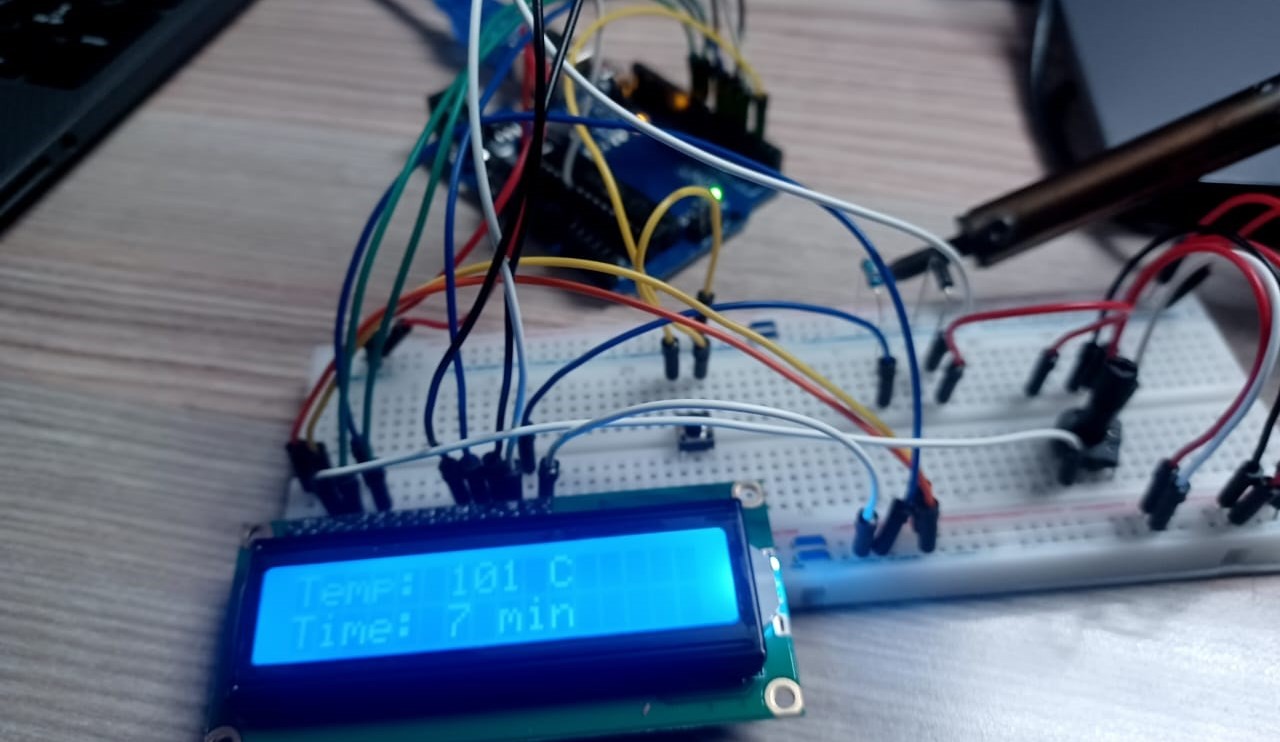
# In the setup function, initialize the serial communication at 9600 baud rate and set up the LCD with 16 columns and 2 rows. Configure the pushbutton pin as an input with a pull-up resistor.

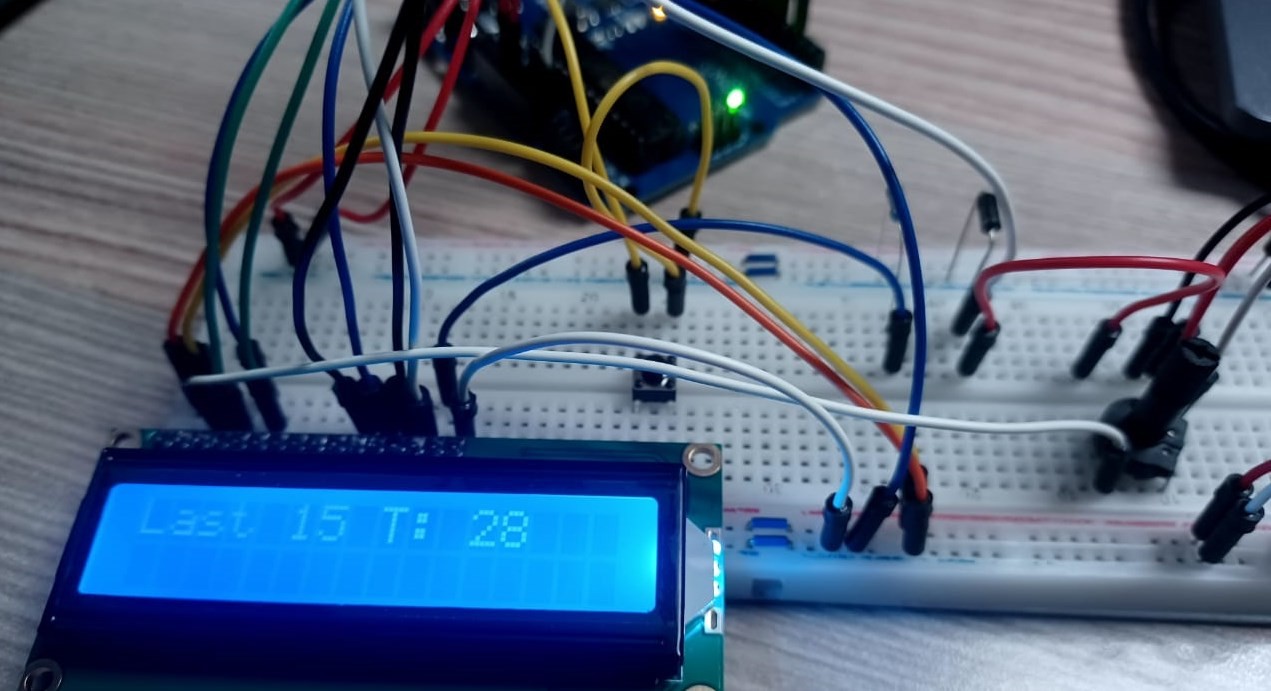
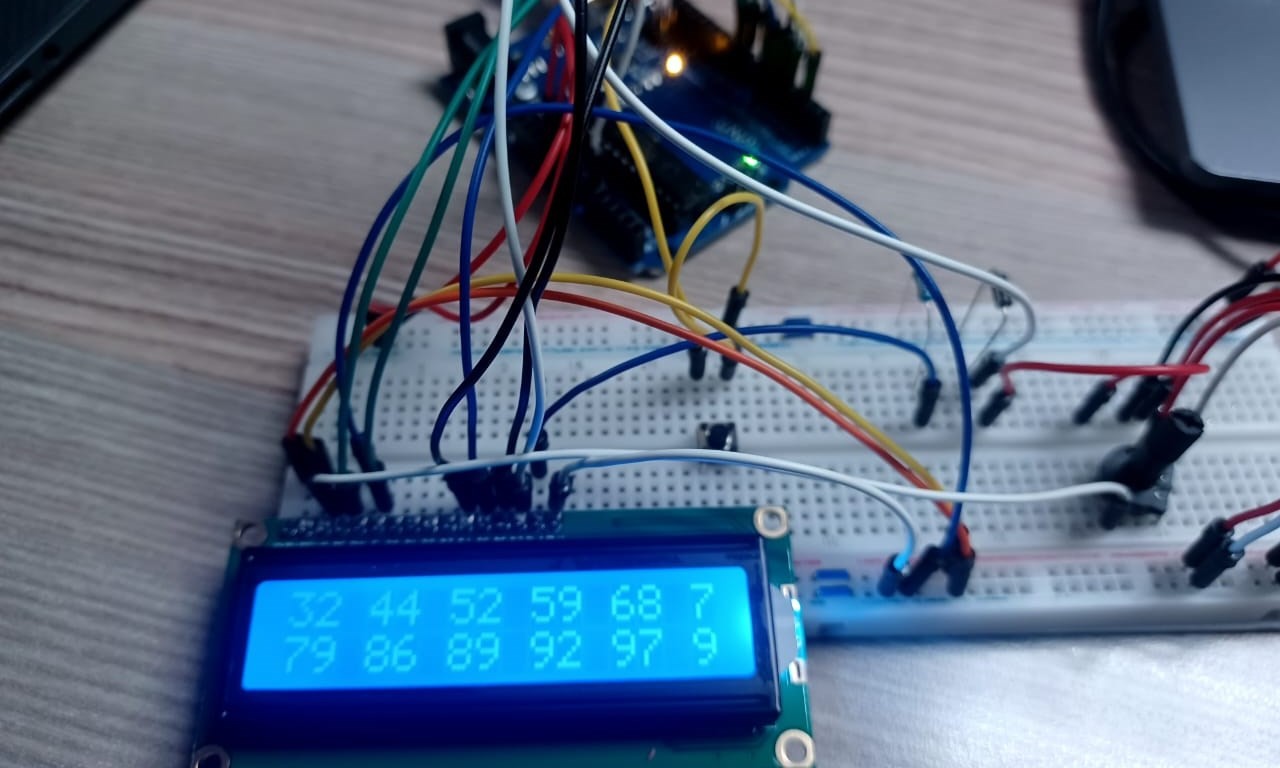
# In the loop function, read the analog value from the diode every minute. Calculate the temperature using the analog value and a coefficient specific to the diode's characteristics. Save the temperature value in the EEPROM and display it on the LCD along with the elapsed time in minutes.

# When the pushbutton is pressed, read the last 15 temperature values from the EEPROM and display them on the LCD for 3 seconds.

# Use a cyclic buffer to store the temperature values in the EEPROM. Once the buffer is full (after 15 readings), the oldest value is overwritten by the newest value, ensuring that the most recent data is always available.

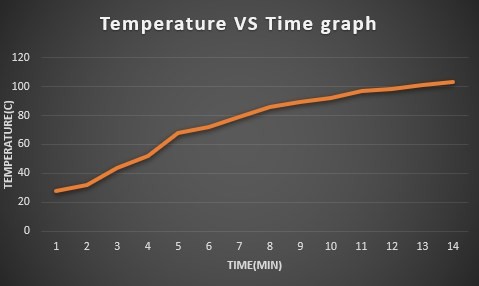
# A circuit board with wires and a soldering iron Description automatically generated**Hard Wired Circuit and Analysis of Results**





The analysis of the results from the temperature measuring system using a 1N4001 diode reveals several key findings. The system successfully measured and displayed temperature values on the LCD screen, with the data being accurately stored in the EEPROM for later retrieval. The pushbutton functionality allowed for the display of the last 15 temperature readings, demonstrating the system's ability to handle historical data effectively. The use of the EEPROM library proved to be efficient for data retention across power cycles, ensuring that the temperature values were preserved even after the system was powered off. The cyclic buffer implementation for storing temperature values worked as intended, with the oldest values being overwritten by the newest ones once the buffer was full. This approach ensured that the most recent data was always available for display. Overall, the system met its objectives by providing a reliable and user-friendly way to measure, store, and display temperature readings, while also offering insights into the practical applications of the EEPROM library in embedded systems.

Diode Output Graph



### The microcontroller assignment successfully achieved its objectives by implementing a temperature measuring system using a 1N4001 diode, storing temperature values in EEPROM, and displaying them on an LCD screen. The addition of a pushbutton to retrieve and display the last 15 temperature values on the LCD.

### I found this practical helpful because it helped me to gain more knowledge about EEPROM. I even gained a lot of understanding about using the EEPROM() library, like reading or putting data on it.