Mathematics of Temperature Sensing and Control

Introduction:

In this project, we explore the principles of temperature measurement and control using an Arduino Nano microcontroller along with a potentiometer. The goal is to convert analog temperature readings into digital values, compute temperatures in Celsius and Fahrenheit, and activate an alarm (buzzer) and fan when a predefined temperature threshold is reached.

Components Used:

- Arduino Nano
- Potentiometer to prototype temperature rise and drop.
- Breadboard for Circuit Assembly
- Buzzer and Fan for Temperature Control
- Serial Analyzer for Data Visualization

Mathematical Concepts:

1. Analog-to-Digital Conversion (ADC):

- The potentiometer outputs an analog voltage proportional to the temperature.
- Arduino's ADC converts this analog voltage to a digital value (adcVal) ranging from 0 to 1023 (10-bit resolution).

2. Voltage Calculation:

- The ADC value (adcVal) is converted to voltage (milliVolt) using: milliVolt = adcVal * (5000 / 1024) Here, 5000 millivolts is the reference voltage (ADC_VREF_mV) and 1024 is the ADC resolution (ADC_RESOLUTION).

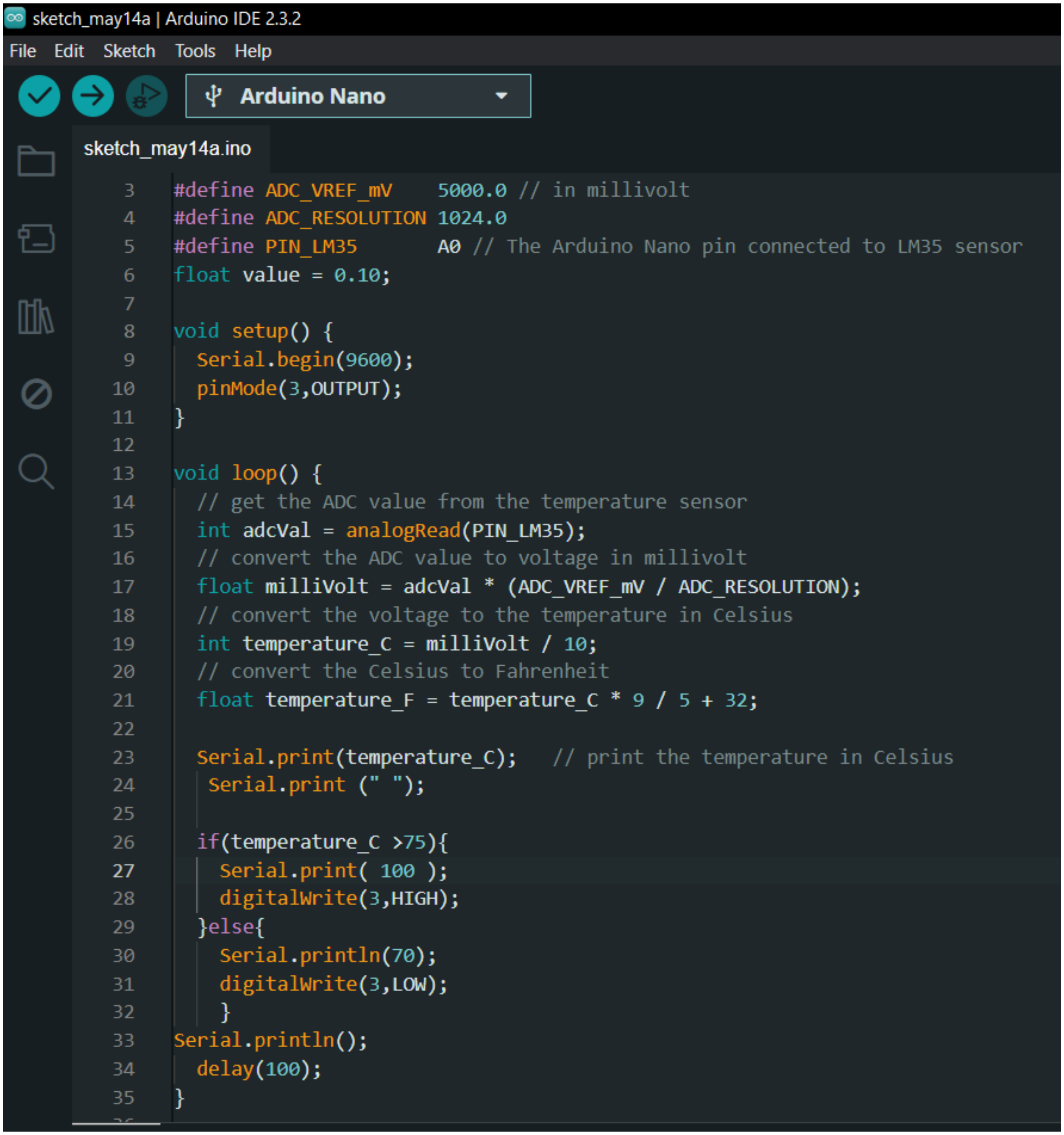
3. Temperature Calculation:

- Temperature in Celsius (temperature_C) is calculated from milliVolt: temperature_C = milliVolt / 10
- Temperature in Fahrenheit (temperature_F) is derived from Celsius: temperature_F = temperature_C * 9/5 + 32

4. Temperature Threshold and Control:

- In this project, we set a temperature threshold of 70°C . When the temperature exceeds this threshold, we activate the buzzer and a fan to cool down the system.

- The buzzer is connected to pin 3 of the Arduino Nano and is controlled using `digitalWrite()` based on the measured temperature.


sketch_may14a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Nano
sketch_may14a.ino

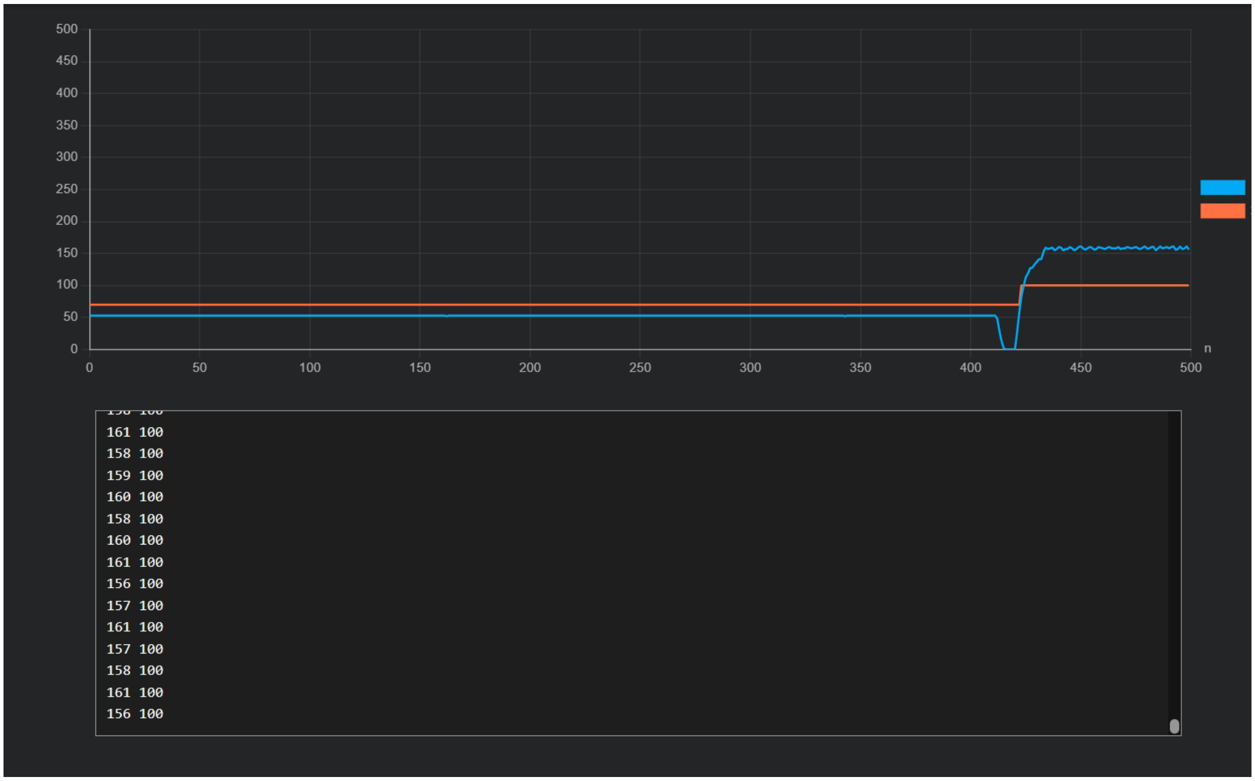
```
#define ADC_VREF_mV    5000.0 // in millivolt
#define ADC_RESOLUTION 1024.0
#define PIN_LM35       A0 // The Arduino Nano pin connected to LM35 sensor
float value = 0.10;

void setup() {
  Serial.begin(9600);
  pinMode(3,OUTPUT);
}

void loop() {
  // get the ADC value from the temperature sensor
  int adcVal = analogRead(PIN_LM35);
  // convert the ADC value to voltage in millivolt
  float milliVolt = adcVal * (ADC_VREF_mV / ADC_RESOLUTION);
  // convert the voltage to the temperature in Celsius
  int temperature_C = milliVolt / 10;
  // convert the Celsius to Fahrenheit
  float temperature_F = temperature_C * 9 / 5 + 32;

  Serial.print(temperature_C);   // print the temperature in Celsius
   Serial.print (" ");

  if(temperature_C >75){
    Serial.print( 100 );
    digitalWrite(3,HIGH);
  }else{
    Serial.println(70);
    digitalWrite(3,LOW);
    }
Serial.println();
  delay(100);
}
```

Project Execution and Results:

- The potentiometer give a prototype temperature readings due to its linear output.

- The Arduino Nano reads these temperature values, computes them in both Celsius and Fahrenheit, and then triggers the buzzer and fan when the temperature exceeds a predefined limit.

- The use of a serial analyzer helps in plotting and visualizing the digital output (temperature values) and analog input (voltage readings) over time, aiding in data analysis and system monitoring.



```
158 100
161 100
158 100
159 100
160 100
158 100
160 100
161 100
156 100
157 100
161 100
157 100
158 100
161 100
156 100
```

Conclusion:

This project demonstrates the application of mathematical concepts in real-world scenarios, such as temperature sensing and control. By leveraging ADC conversions and simple arithmetic operations, we can build effective temperature monitoring systems using microcontrollers like Arduino. The integration of components like LM35 sensors, buzzers, and fans showcases the interdisciplinary nature of STEM projects, combining electronics, mathematics, and programming for practical applications.