University of Montenegro
Faculty of Electrical Engineering

Course: Industrial electronics

Vernier sensor for temperature mesaurement

Authors
Marko Marković
Stefan Šćepanović
Nina Blagojević
Milena Božović

Mentor
Prof. dr Radovan Stojanović

April 28, 2020

Contents

1 Summary

This is a laboratory project, made at the Faculty of Electrical Engineering, University of Montenegro, with mentoring of prof. dr Radovan Stojanović. The problem is presented in detail with our proposed solution. For realization of this project, we used an Arduino Uno (with its enviroment), a Vernier sensor, a cup, water, ice cubes and salt.

2 Problem Description

Our task was to demonstrate how Vernier temperature sensor works on one basic experiment from physics.

3 Solution

Firstly, we had to connect the Vernier sensor and the Arduino Uno with our computer (as seen on figure 1). Then we had to program our sensor. We set the desired time sampling interval and set up the Arduino to convert the value of the analog voltage from the sensor to temperature units [°C]. When we send an upper-case "A" over the serial monitor the sampling will start and the converted temperature values well be displayed on the Serial Monitor. To terminate this process, we send an upper case "B" The Arduino code is shown bellow.

```
1
 float dataRate = 2;           // set number of samples per second.
3 const char delimiter = '\t';  // delimitter character
  char c='a';
5 const int ThermistorPIN = A0;
  float Temp;
7 int rawAnalogReading;

9 unsigned long timeRef;     // reference for starting time

11 unsigned long timeInterval;
   unsigned long ndx;         // index for data counter
13 unsigned long thermistor;

15 void setup()
   {
17   Serial.begin(9600);
     Serial.println("Vernier Format 2");
19   Serial.println("Temperature Readings taken using Ardunio");
```

```
  Serial.println("Data Set");
  Serial.print("Time");
  Serial.print("\t"); //tab character
  Serial.println ("Temperature");
  Serial.print("seconds");
  Serial.print("\t"); // tab character
  Serial.println ("degrees C");
  timeInterval = 1000 / dataRate;
  timeRef = millis();
  cli();//stop interrupts
  //set timer1 interrupt at 1Hz
  TCCR1A = 0;// set entire TCCR1A register to 0
  TCCR1B = 0;// same for TCCR1B
  TCNT1  = 0;//initialize counter value to 0
  // set compare match register for 1hz increments
  OCR1A = (16000000) / (1 * 1024) - 1;
  // turn on CTC mode
  TCCR1B |= (1 << WGM12);
  // Set CS10 and CS12 bits for 1024 prescaler
  TCCR1B |= (1 << CS12) | (1 << CS10);
  // enable timer compare interrupt
  TIMSK1 |= (0 << OCIE1A);
  sei();//allow interrupts
}
ISR(TIMER1_COMPA_vect) {
  //the print below does the division first to avoid overflows
    Serial.print((float)(millis() - timeRef) / 1000, 2);
    rawAnalogReading = analogRead(ThermistorPIN);  // reads raw
    analog value from Arduino
    thermistor = resistance(rawAnalogReading);     // converts raw
     analog value to a resistance
    Temp = steinharthart(thermistor);              // Applies the
    Steinhart-hart equation

    Serial.print(delimiter); //tab character
    Serial.println(Temp,1);   // display temperature to one digit
}
void loop()
{
  if(Serial.available()){
    c = (char)Serial.read();
    Serial.print(c);
    if(c=='A'){
      Serial.print(c);
      TIMSK1 |= (1 << OCIE1A);
    }
    else if(c=='B'){
      Serial.print(c);
      TIMSK1 &= (0 << OCIE1A);
```

```
    }
  }


}
unsigned long resistance(unsigned long rawAnalogInput)

{
  unsigned long temp;  // temporary variable to store calculations
     in
  temp = (rawAnalogInput * 15000) / (1023 - rawAnalogInput);
  return temp; // returns the value calculated to the calling
    function.
}

float steinharthart(unsigned long resistance)
// function users steinhart-hart equation to return a temperature
    in degrees celsius.

{
  float temp; // temporary variable to store calculations in
  float logRes = log(resistance);
  // calculating logirithms is time consuming for a
    microcontroller - so we just
  // do this once and store it to a variable.
  float k0 = 0.00102119;
  float k1 = 0.000222468;
  float k2 = 0.000000133342;

  temp = 1 / (k0 + k1 * logRes + k2 * logRes*logRes*logRes);
  temp = temp - 273.15;  // convert from Kelvin to Celsius
  return temp;
}

```

After we ensured that this program works, we wanted to test this sensor. Therefore, we conducted a little experiment. We filled a cup with water and ice cubes. Then we added salt in the cup and wedged our sensor in it. We found that adding salt to ice water lower the temperature below 0°C. When salted ice melts, the water can't refreeze as readily because the salty water isn't pure water anymore and because the freezing point is lower. As more ice melts, more heat is absorbed, bringing the temperature down even lower. We observed our Serial Monitor and the temperature did start to decline to approximately -3°C. As our results were concordant with

theoretical expectations, we concluded that our experiment was successful. A graph showing decline of temperature was made of the results taken from our sensor (it can be seen on figure 2).

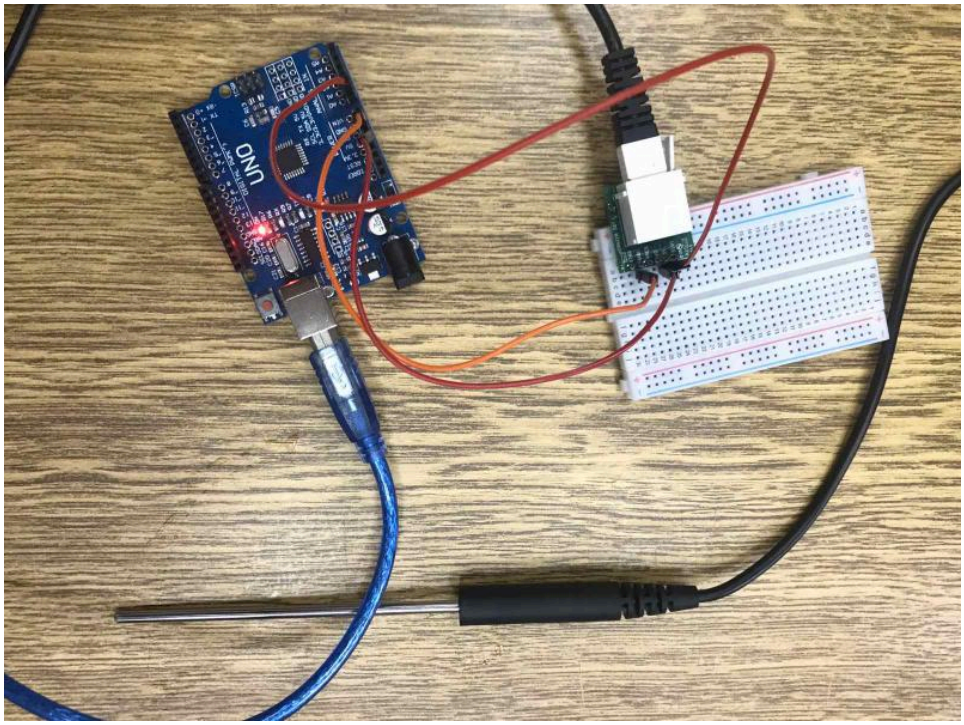

Figure 1:

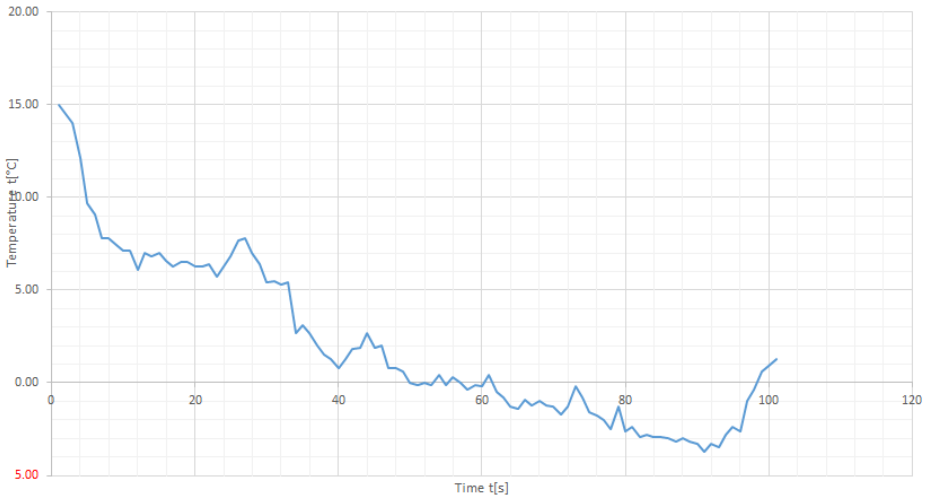

Figure 2:

4 Link for video

https://youtu.be/NCCueHk6aoQ

5 Literature

References

[1] *https://www.vernier.com/products/*

[2] *https://www.vernier.com/engineering/arduino/*

[3] *https://www.thoughtco.com/how-cold-does-ice-get-with-salt-4017627*