Sensors & Signals

Project Report on Sensor Measurement using Microcontroller

Anna Beatriz Yabe
Tamaki Kawamura

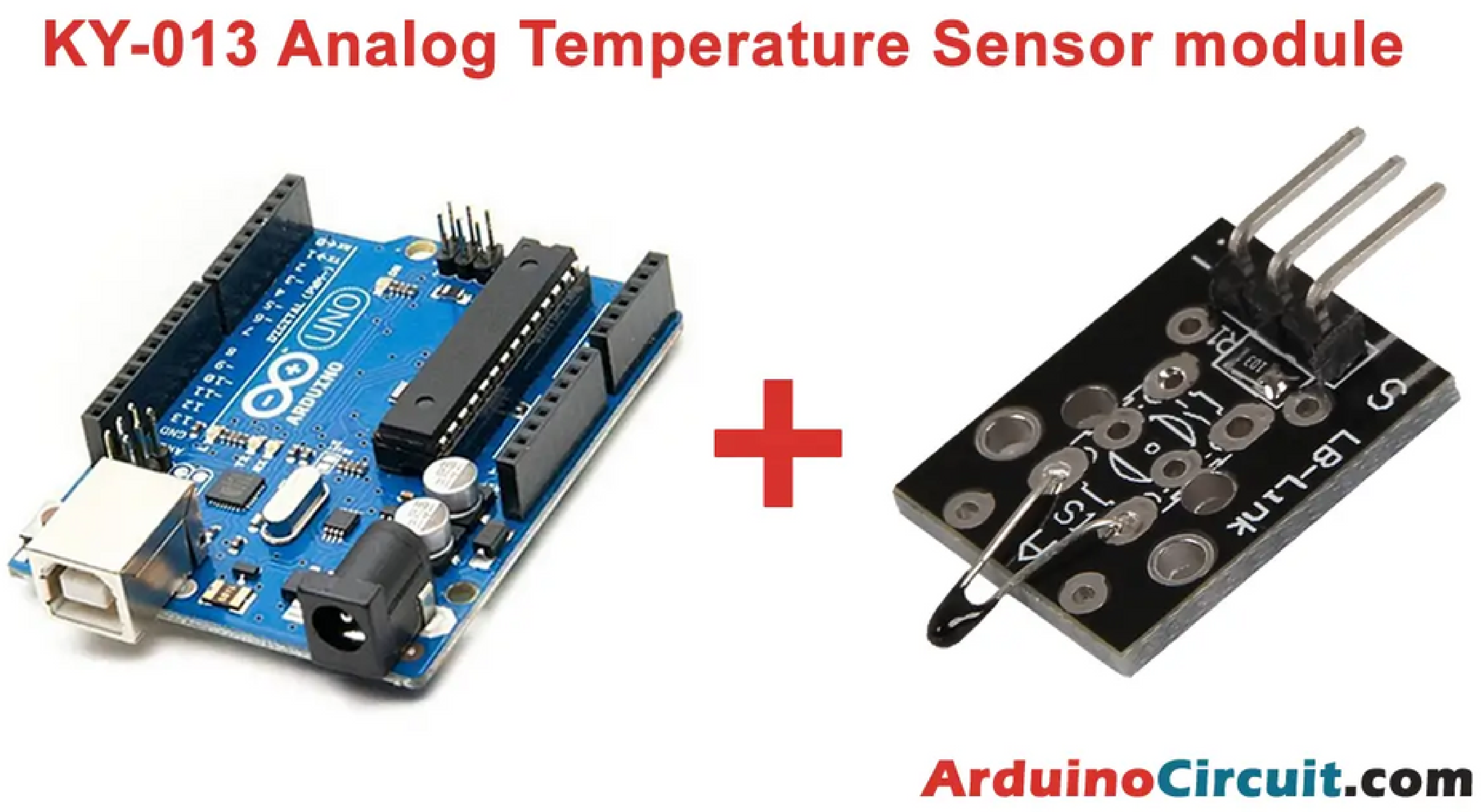
Turku University
of Applied Sciences

Summary

Introduction

This project is focused on analyzing how a microcontroller work with an analog sensor to take measurements. For this report the microcontroller used is SparkFun ReadBoard and the sensor is KY 013 Analog Temperature Sensor Module.

The assignment aimed to experiment and demonstrate sensor measurements using a microcontroller. It provided an opportunity to build a system where the measurements taken by the sensor and processed by the arduino are shown in the computer. Furthermore, we got the chance to analyze and study the uncertainties from the measurements and repeatability, leading to a deeper understanding of the sensors and signals subject. The system was the sum of a microcontroller SparkFun ReadBoard and KY 013 Analog Temperature Sensor Module. Where we developed a code in ArduinoIDE to read the temperatures from the system, and used an application called Teraterm to save the readings for a later analysis. After enough material, we studied the Type-A and Type-B uncertainties of the system where we could notice the errors in measurement, and the effects of jitter to the experiment.



Revised Learning Objectives

To fulfill the tasks for this project we had to work continuosly on understanding how measuremets in sensors using a microcontroller work. In the project's initial stages, we delved deeper into understanding the devices. The sensor we used is KY 013 Analog Temperature Sensor Module, the small electronic component is designed to measure temperature variations in its surroundings. Equipped with a thermistor, this module produces an analog voltage output proportional to the ambient temperature. And when paired with the microcontroller SparkFun ReadBoard and a properly code we can get measurements for the temperature.

To develop the code more research was needed, despite our previous experience with C code it was still a little complicated to do from scratch. The IDE used was Arduino IDE, which works basically with two main functions: setup and loop. In the setup function we configured our output to be in the Serial Port 9600. And in the loop function it starts by reading the analog voltage from the microcontroller. The reading has to be multiplied by the reference voltage of the microcontroller, in this case 5V, and then divide by 1024 to normalization purposes.

After that we can get the resistance and temperatures values by calling other functions. To get the resistance the function uses Ohm´s Law and Voltage Divider formulas. To achieve the temperature value in celsius the Steinhart-Hart equation is used. Which is the equation used to model the resistance-temperature relationship of certain types of temperature sensors. The equation has the parameters adjusted for the KY 013 Analog Temperature Sensor Module.

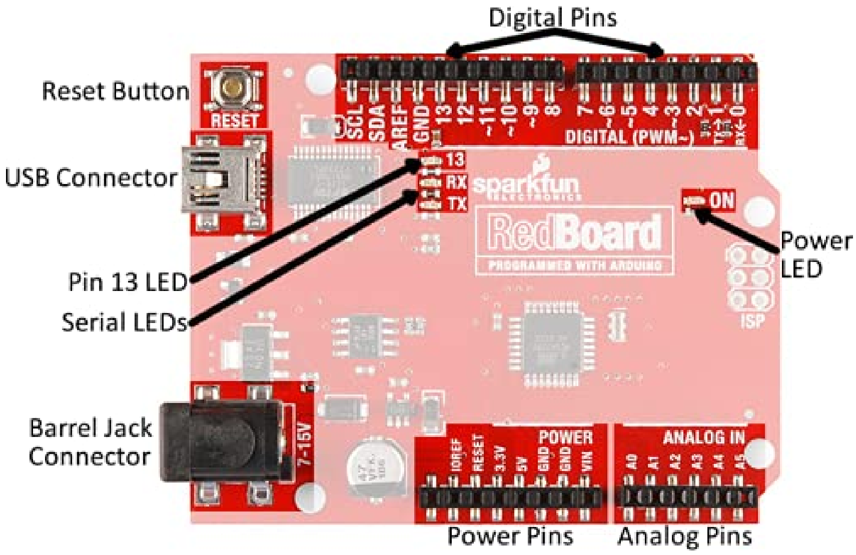
$$T = \frac{1}{A + B\ln(R) + C[\ln(R)]^3}$$

System Overview

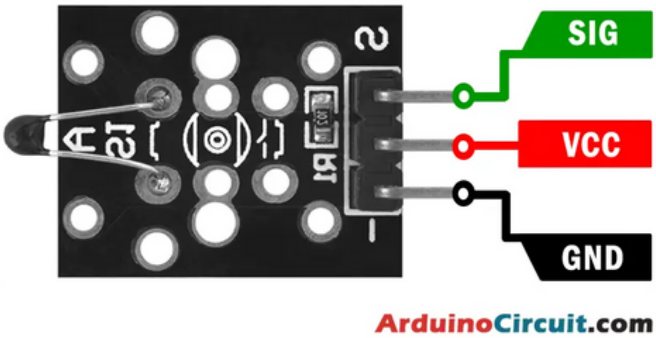
Components of the system:

- SparkFun ReadBoard
- KY 013 Analog Temperature Sensor Module
- Wires

SparkFun ReadBoard: is an Arduino-compatible development board that contains 14 Digital I/O pins with 6 PWM pins, 6 Analog Inputs, UART, SPI and external interrupts. Can be programmed over a USB Mini-B cable using the Arduino IDE.

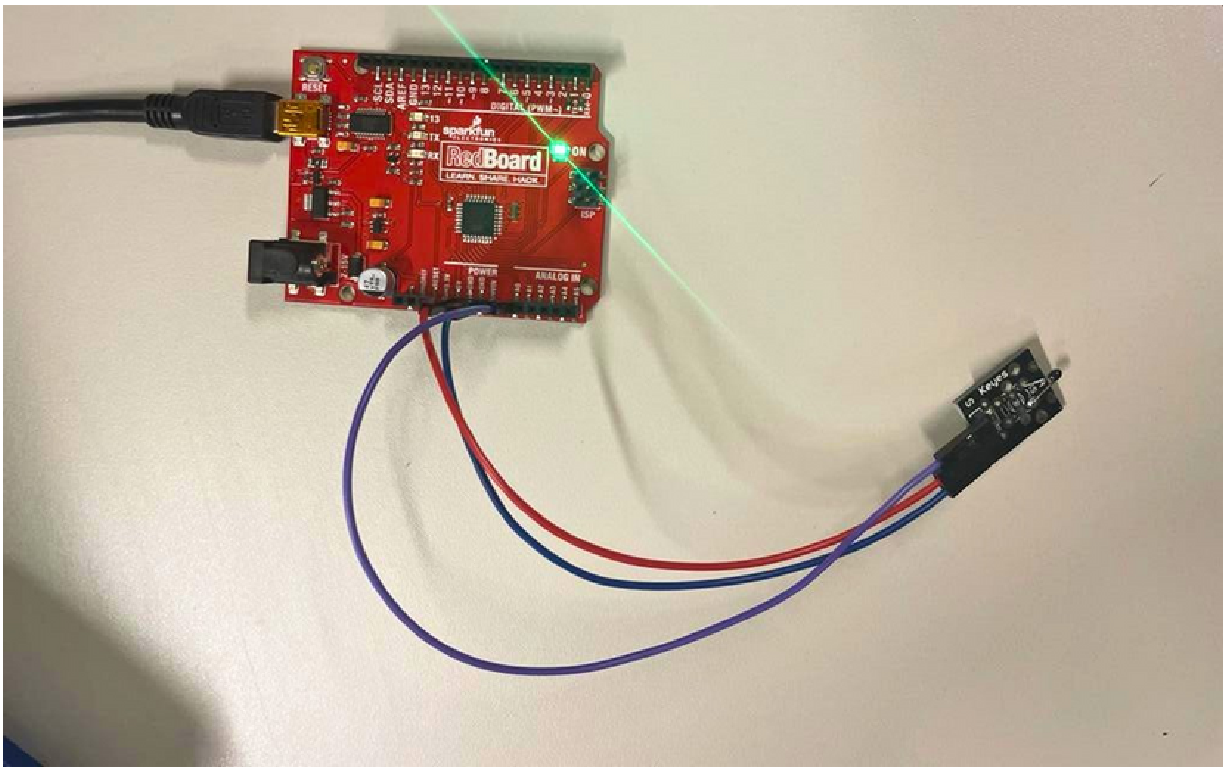


KY 013 Analog Temperature Sensor Module: is a sensor that can calculate the actual temperature by using the thermistor resistance that varies according to its surrounding temperature. Contains a NTC thermistor, a 10KΩ resistor, and 3 male header pins. The device is compatible with Arduino and ESP32.



Connection Details:

SparkFun ReadBoard	Sensor
GND	Ground (near -)
5V	Power Line
A0	Output (near S)

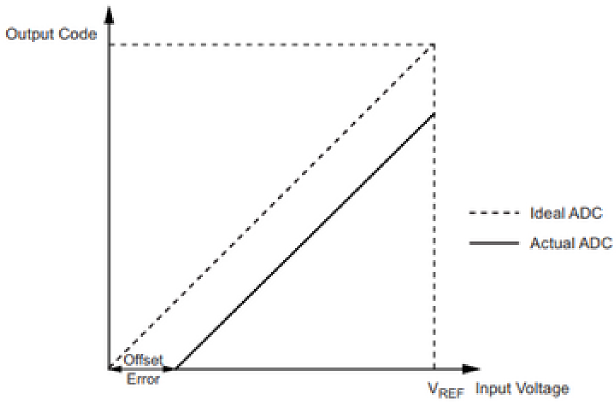


Type-B Uncertainty

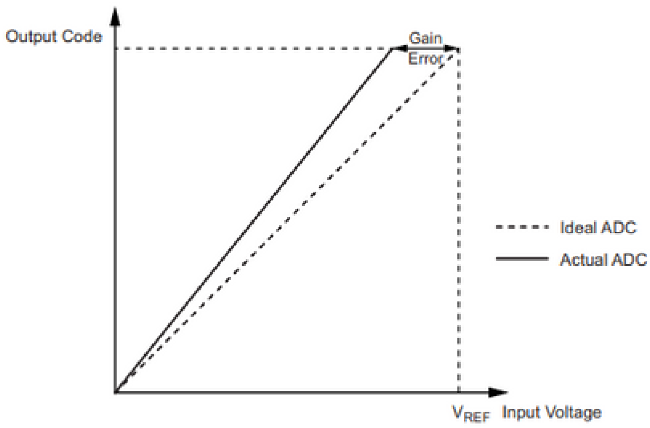
Type-B Uncertainty is defined as the inherent variability in measurements or observations caused by various sources like instrument, noise, environmental conditions, or flunctuations in the quantity being meaured. This uncertainty is evaluated from instrument specification, calibration reports, textbooks, and other data sources. And when the uncertainties are independent we can root-sum-square them to calculate the total uncertainty.

The SparkFun Redboard has an ATmega328 Microcontroller which we could find the datasheet. ATmega 328 has an ADC that cointains several parameters that leads to the deviation from the ideal behaviour, those are:

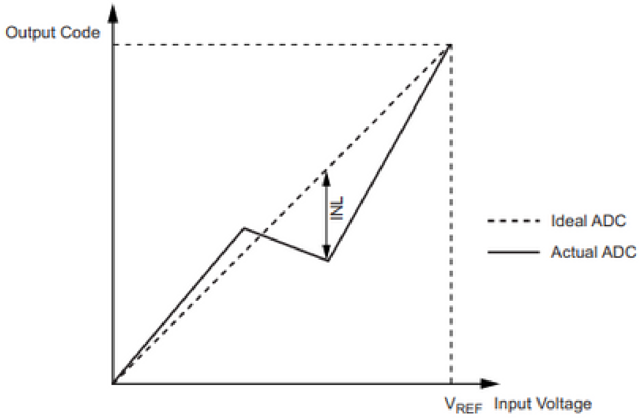
- **Offset:** deviation of the first transition compared to the ideal transition



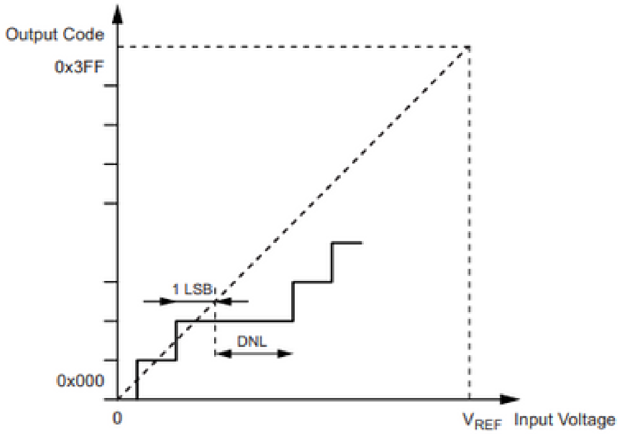
- **Gain error:** after the adjustment for the offset, the gain error is found as the deviation of the last transition compared to the ideal transition



- **Integral non-linearity:** after adjustments in the offset and gain error, the Integral non-linearity is the maximum deviation to the actual transition compared to an ideal transition for any code



- **Differential non-linearity:** maximum deviation of the actual core width from the ideal code width



- **Quantization error:** because of quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5 LSB.
- **Absolute accuracy:** is the maximum deviation of an actual (unadjustmed) transition compared to the ideal one. This is the compound effec of offset, gain error, differential error, non-linearity, and quantization error. Ideal value ±0.5 LSB

Furthermore, we searched in websites and in documentation about the KY 013 Temperature Sensor. The sensor can sense temperature measurements from -55°C to 125°C (-67°F to 257°F) with measurement accuracy of ±0.5°C. However, we cannot found the coverage factor for the module.

Type-A Uncertainty

Type-A Uncertainty is defined as random error from measurements outcomes. It is calculated from series of observations and is defined by standard deviation + standard error mean. This uncertainty is not caused by systematic error but due to the variability in measurement outcomes.

The standard deviation is calculated from the measurements divided for the number of measurements. To calculate it, we used MATLAB for reading the files and discovering the averages of the measurements as well as the Type-A Uncertainty. Above there is a essential part of our code.

```
% File
file = '../measurement_2.log';

% Reading from gile
f1 = fopen(file, 'r');
data = textscan(f1, 'Voltage (V): %f Resistance (Ohm): %f Temperature (-C): %f Temperature (-F): %f', 'delim
fclose(f1);

% Adding the data to vectors
voltage = data{1};
resistance = data{2};
temperatureC = data{3};
temperatureF = data{4};

% Type A Uncertainty
N = length(voltage);  % Number of measurements

s_temperatureC = std(temperatureC);  % Standard Deviation for Temperature in Celsius
s_temperatureF = std(temperatureF);  % Standard Deviation for Temperature in Fahrenheit

u_temperatureC = s_temperatureC / sqrt(N);  % Type A Uncertainty for Temperature in Celsius
u_temperatureF = s_temperatureF / sqrt(N);  % Type A Uncertainty for Temperature in Fahrenheit
```

For this assignment we took measurements from our lab, which is mostly controlled environment. The experiment took place in our university classroom, which has a HVAC that tries to maintain the temperature of the building. We took two meaurements on different days, and saved into files the information sensor we got.

From the first file, measuring the room temperature of our class on November 9th at 09:54AM, with desk to the right of the first row, and we got the following results:

```
>> typeA
Number of measurements:
    86

Voltage Average:
    2.7500

Resistance Average:
   1.2259e+04

Average Temperature in Celsius:
   19.8235

Average Temperature in Fahrenheit:
   67.6763

Type A Uncertainty for Temperature in Celsius:
    0.0020

Type A Uncertainty for Temperature in Fahrenheit:
    0.0036
```

The second file measured the room temperature of our class on November 16th at 08:23AM, with desk to the right of the first row, and we got the following results:

```
>> typeA
Number of measurements:
   486

Voltage Average:
    2.7645

Resistance Average:
   1.2382e+04

Average Temperature in Celsius:
   19.5710

Average Temperature in Fahrenheit:
   67.2287

Type A Uncertainty for Temperature in Celsius:
    0.0066

Type A Uncertainty for Temperature in Fahrenheit:
    0.0117
```

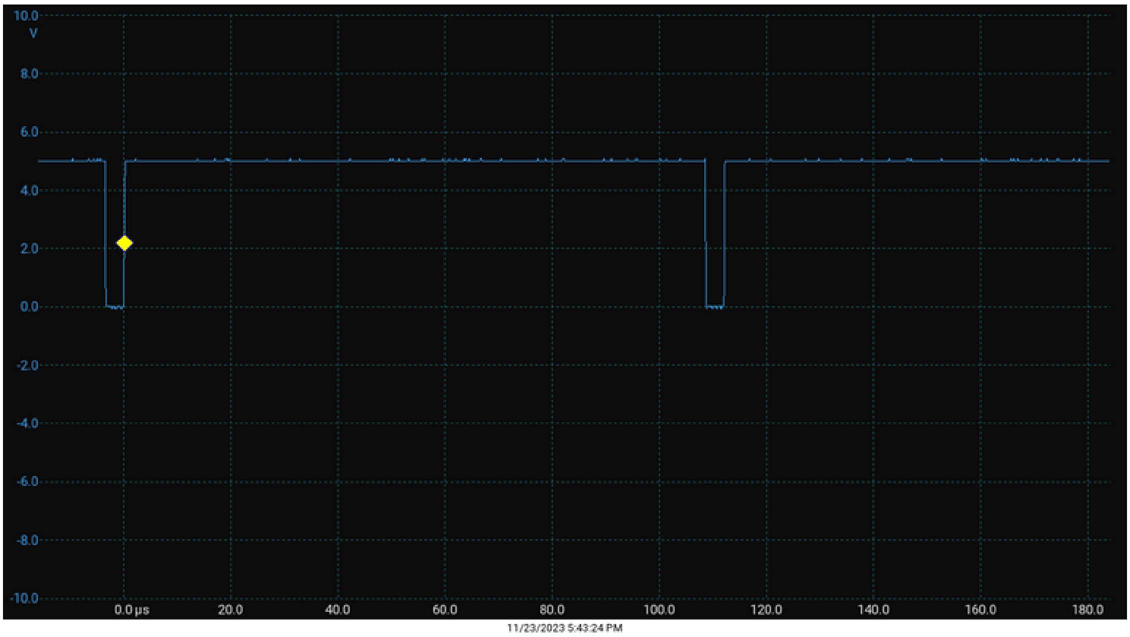
Exploring Constant Rate Sampling and Jitter Analysis

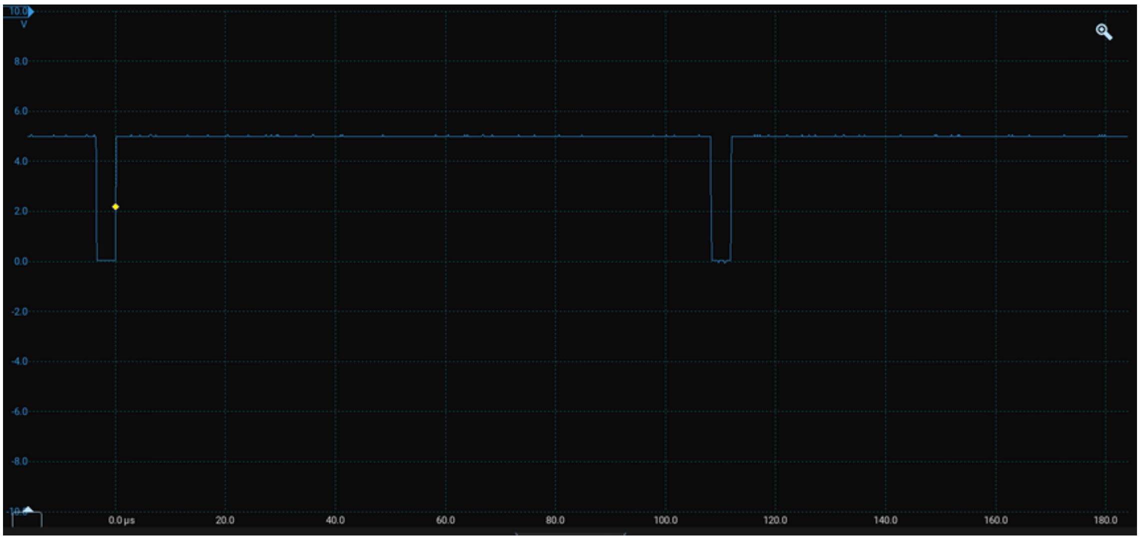
Jitter rerfers to the variation in timing of a signal's transitions or events, in other words is the instability in the timming of the event. This instability can be one of the sources of uncertainty of the measurement. Thus, it is important to consider the effects of jitter for measuring temperature.

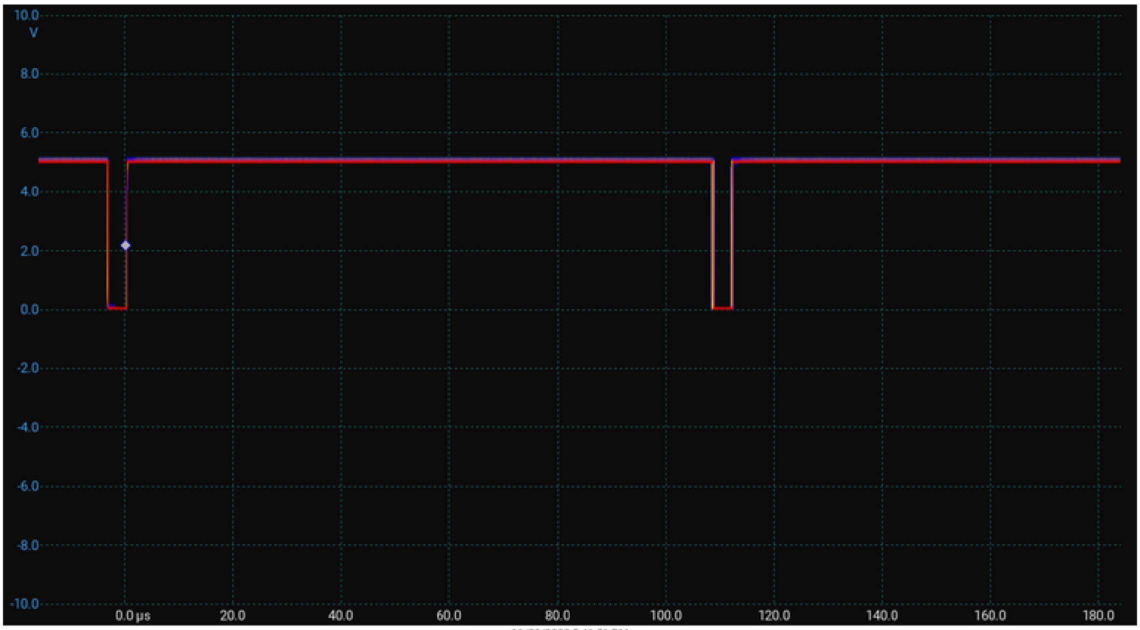
To measure jitter we used a breadboard, we created a line for ground where we connected the sensor ground, the redboard ground and the picoscope ground. Furthermore, there was a vcc line where the redboard and the sensor were connected, and lastly we connected picoscope to the A5 GPIO pin (that was not being used for the measurement) in the redboard.

The code was changed in Arduino IDE to trigger the rising and falling edge with a delay of 30ms. At first we were not able to see the jitter, so we deleted the delay as well with the print functions. Finally, we were able to use the picoscope to see how jitter affects the signal. The jitter is miniscule, less than milisseconds, which means that it does not affect that much the uncertainty.

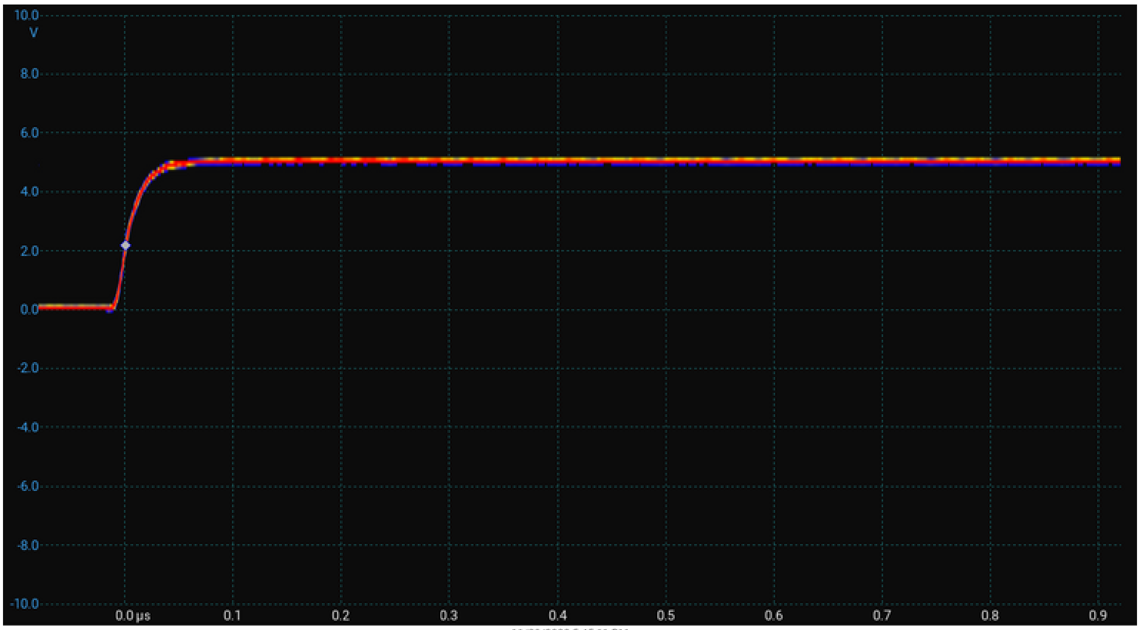
With the following pictures we can see the jitter from the fluctuation of the osciloscope measurement results:







11/23/2023 5:41:51 PM



11/23/2023 5:45:11 PM

Summary of Leaning Achievements

This project was developed to measure temperature by using an analog temperature sensor module and microcontroller. In the process we had to analyze the sensor specifications, calculate its uncertainties and inspect jitter effect.

For the first part we had to understand the components of our project. The microcontroller we used is SparkFun RedBoard, which can be programmed over a USB Mini-B cable using the Arduino IDE. The sensor we used was KY 013 Analog Temperature Sensor Module, which is compatible to the microcontroler. By using the thermistor resistance that varies according to its surrounding temperature we can calculate the temperature with the sensor. Ky 013 also contains a NTC thermistor, a 10KΩ resistor, and 3 male header pins.

Every sensors measurement has uncertainty, and throughout this project we could analyze both types of uncertainty found in our system. The Type-B Uncertainty in our system could be found from instrument specification, calibration reports, textbooks, and other data sources. For the microcontroller we discovered that it had a component where multiple factors such as offset, gain error, integral non-linearity, differential non-linearity, quantization error, and absolute accuracy affects the uncertainty of the measurements. And in the documentation of the sensor we discovered that its measurement have an accuracy of ±0.5°C. Type-A Uncertainty are considered the random errors we normally get from repeatability. In order to discover what was the Type-A Uncertainty we got our measurements and analyzed them with the help of MATLAB. For the first data set we got 0,0020 uncertainty for Temperatures in Celsius, and 0,0036 for Temperatures in Fahreinheit. The second data set we got 0,0066 uncertainty for Temperatures in Celsius, and 0,0117 for Temperatures in Fahreinheit.

Lastly, we inspected the jitter effects in our system. By using an osciloscope we were able to conclude that our jitter is extremelly small, less than milisseconds, which means that it does not affect that much the uncertainty.

The project was really important to apply our theorical knoledge for the subject in a real world example.