



A05.01 LaACES First Semester Capstone

Datalogger and Calibration Report



Introduction

In this activity, you will combine your Arduino, GPS-SD Shield, and Diode temperature sensor into a simple temperature data logger that records calibrated temperature data, timestamps that data using the GPS, and saves that to the SD Card.

Be sure you read the report requirements first, so you do not forget to record required information in your lab notebook.

Materials List

1. Arduino Mega
2. Adafruit GPS Logger Shield
3. SD Card
4. Diode Temperature PCB
5. Programming Laptop

Adjusting and Calibrating the Temperature Sensor

1. You must adjust your temperature sensor circuit so that when read out by the Arduino ADC, it meets the following minimum requirements. This will require you to adjust the gain and/or offset of the amplifying circuit on your PCB.
 - 1.1. Maximum readout temperature of **+50 °C or greater**
 - 1.2. Minimum readout temperature of **-50 °C or lower**
 - 1.3. Temperature resolution of **1 °C/ADC or better (a smaller resolution is better)**
 - 1.4. Note these are minimum requirements; your circuit does not have to exactly hit any of these targets, but all three must be met or exceeded. For example, A maximum of 75 °C(at 0 ADC), a minimum of -100 °C(at 1023 ADC), and a resolution of 0.25 °C/ADC is acceptable.
2. Once your circuit is set, perform a final calibration to determine the conversion from ADC units to °C.
 - 2.1. Your calibration data should include measurements at **8 different temperatures**
 - 2.2. 1 Measurement **below 10 °C**
 - 2.3. 1 Measurement **above 50 °C**



A05.01 LaACES First Semester Capstone

Datalogger and Calibration Report



Writing your Datalogger Software

1. Once you have completed your calibration, write an Arduino sketch that writes data files to the SD card with the following contents (You may order the data as you choose)
 - 1.1. Temperature sensor reading, in calibrated °C.
 - 1.2. GPS Time (Hours, minutes, and seconds)
 - 1.3. GPS Date (Year, Month, and Day)
 - 1.4. GPS Altitude (an active fix is not required, but your program should record 0 or null if no altitude is available)
 - 1.5. Fix Status as 0 / 1 or True / False
 - 1.6. Arduino timestamp to milliseconds (ie, millis() or similar)
2. In addition to the above data, your files must use the following format
 - 2.1. CSV Format with each data element from 1 in a separate column
 - 2.2. Each data file should include column labels
 - 2.3. Data must be recorded in multiple, ordered files (numbered or time-named files)
 - 2.4. Each data record (row) must include sufficient information so that the data points could be organized in order if the rows were randomly shuffled
3. Your program must meet these minimum performance requirements
 - 3.1. Start and run automatically when powered on
 - 3.2. Sample data at least 1/minute
 - 3.3. Operate continuously for 30 minutes or more without user input

Report Requirements

Upon completion of the calibration

- 1) Cover Page
 - a) Title
 - b) Name(s)
 - c) Date Submitted
- 2) Electrical Design
 - a) Sensor Interface – This should include the following elements
 - i) A brief description (1 to 2 sentences) of the temperature sensor circuit.
 - ii) A schematic or block diagram showing how you connected the temperature sensor to the Arduino (and any external electronics like power supplies if used)
 - iii) How you calculated the resistor values for your circuit, and the resulting gain and offset – **This should include the actual values**



A05.01 LaACES First Semester Capstone



Datalogger and Calibration Report

- b) Calibration Procedure – a brief description (**1 page max**) with photos of the procedure used to calibrate the temperature sensor.
- c) Calibration Results
 - i) The resulting calibration data, including a plot with ADC on the X axis and temperature on the Y axis. You must also include the calibration equation (**with units**) for converting from Arduino ADC to °C.
- 3) Software Design
 - a) Data Format
 - i) Describe your data format, including what description of the data, order, format of actual recorded values, and sample rate. **Include a screenshot sample of actual data.**
 - ii) Calculate the size (in bytes) of a single data record in your above format and then use that to calculate the total storage needed for four hours of data.
 - b) Software Operation
 - i) Using a written description or flowchart, describe how your software operates from startup (**1 page max, a screenshot or copy of Arduino code is not acceptable**)
- 4) Proof of Performance
 - a) Describe any tests you performed to verify that your completed datalogger functioned, including the results. This should include:
 - i) A sample data file
 - ii) A time(x-axis) plot of temperature(y-axis), GPS time is not required; you may choose any time data available to you
 - iii) A time plot of any other recorded quantity from your data format