**Data Logger Project**

Overview

This lab exercise will introduce the student to an Arduino Uno configured as a data logging device. It uses an Adafruit Data Logger shield equipped with a real-time clock (RTC) and a Secure Digital (SD) memory card. This Arduino data logger configuration is then connected to a temperature sensor using the Arduino’s Inter-Integrated Circuit (I2C) bus.

Notes:

Make sure that power has been removed from the Arduino before inserting or removing the SD memory card.

Your data logger board header pins are already soldered in place

Your SD memory card is already pre-formatted

Your battery has already been inserted into the battery holder

For info on the Data Logger Shield, see: <http://learn.adafruit.com/adafruit-data-logger-shield>

This sketch requires three libraries: Wire, SD, and RTC. Two of the three libraries, Wire and SD, come preinstalled with the Arduino IDE. You will have to manually load the RTC library as part of the lab procedure below.

The Microchip brand TC74 temperature sensor uses the I2C bus for communication. Since the RTC on the Data Logger board also uses the I2C bus for its interface, the I2C bus pins are available and labeled explicitly on the Data Logger board as you will see in the lab procedure. A good reference for the TC74 is located at <http://www.electroschematics.com/9798/reading-temperatures-i2c-arduino/>. [Note that for this lab you do not have to provide separate pull-up resistors for the I2C bus lines as shown on this website because the Data Logger circuit board provides them.]

Procedure:

1. If your RTC library is not already installed: copy the RTC library onto your PC:
   1. Create a folder in your Arduino library space titled “RTClibmaster”. It will look like this in your directory path when complete: <Your Directory>/Arduino/libraries/RTClibmaster
   2. Using the master CD, copy the RTC library files to the new folder. It will look like this when complete:

<Your Directory> /Arduino/libraries/RTClibmaster /<various library files>

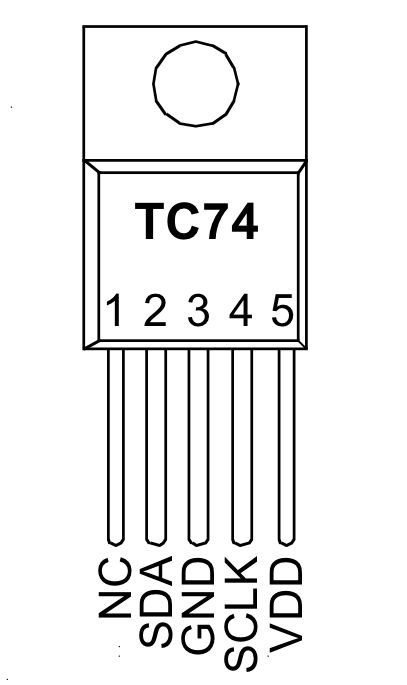
**Note: We are sharing a few master CDs; if one isn’t available just yet, skip steps 1 & 2 and return to them before compiling the data logger program.**

1. If your Data Logger program is not already installed: copy the Data Logger program onto your PC:
   1. Using the master CD, copy the Data\_Logger\_code folder (includes the .ino source code) to your Arduino folder. It will look like this in your directory path when complete:

<Your Directory> /Arduino/Data\_Logger\_code/Data\_Logger\_code.ino

**Note: We are sharing a few master CDs; if one isn’t available just yet, skip steps 1 & 2 and return to them before compiling the data logger program.**

1. If you haven’t already, unplug the USB cable from the Arduino to remove power.
2. Plug the Data Logger shield (board) into the Arduino board
   1. Carefully hold the data logger shield above the Arduino board; align the long pins of the data logger shield with the pin sockets on the Arduino board. (Note that the pins only align one way)
   2. Gently press the data logger shield pins into the Arduino sockets until they are fully seated. (Note: alternate pressure on various areas of the pin/socket connections to maintain roughly even seating of all of the pins so as not to tilt the data logger board too severely and to prevent bending of the data logger pins.)
3. Connect the TC74 temperature sensor to the Arduino using the prototyping board and long jumper wires from the Sunfounder kit:
   1. Insert the TC74 temperature sensor device into five consecutive rows on the proto board (example: pin 1 of the sensor into row 14 of the proto board, pin 2 into row 15, etc). It doesn’t matter which column (a-e) within the row that you use although I prefer column e so that I have room to insert the jumper wires in front of the temperature sensor device. In Fig 1, the front of the temperature device (with lettering that you can’t see in this picture) is facing you. Here is a pinout for the TC74 (note that pin 1 is not used):



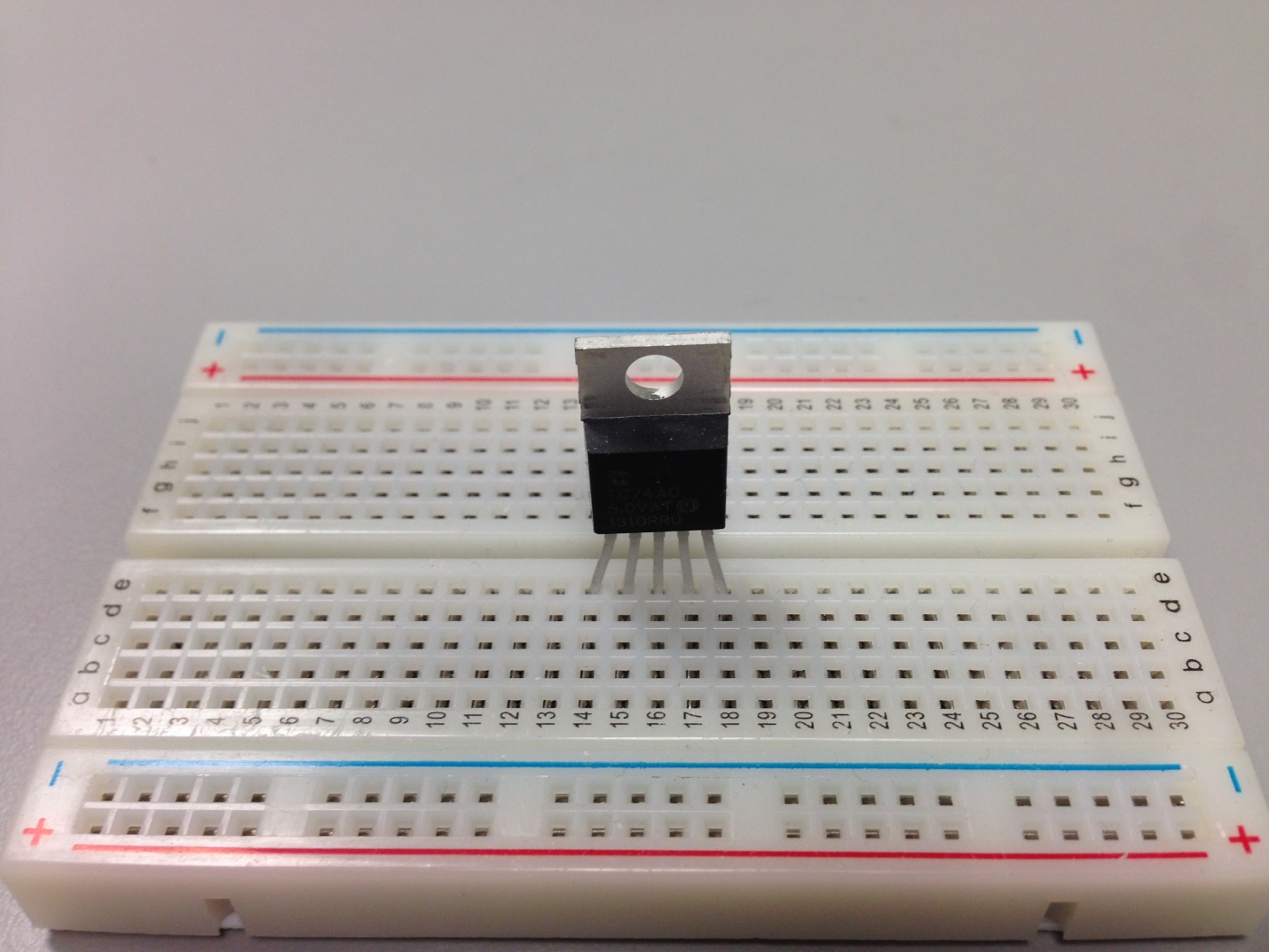


Figure - Temperature sensor inserted into proto board

* 1. Connect the wires between the Arduino data logger board and the temperature sensor per the data logger schematic (or as follows):
     1. Run a long orange wire from pin 2 of the TC74 to the SDA socket on the data logger board (near the reset switch)
     2. Run a long green wire from pin 3 of the TC74 to the Gnd socket on the data logger board
     3. Run a long blue wire from pin 4 of the TC74 to the SCL socket on the data logger board (near the reset switch)
     4. Run a long red wire from pin 5 of the TC74 to the 5V socket on the data logger board
     5. Your circuit should look like Figs 2-4 below

1. Ensure that the write-protect switch on the 4GB SD memory card is set to unlocked and insert the memory card into the card slot as shown in Fig 2. (Make sure power is off to the Arduino.)

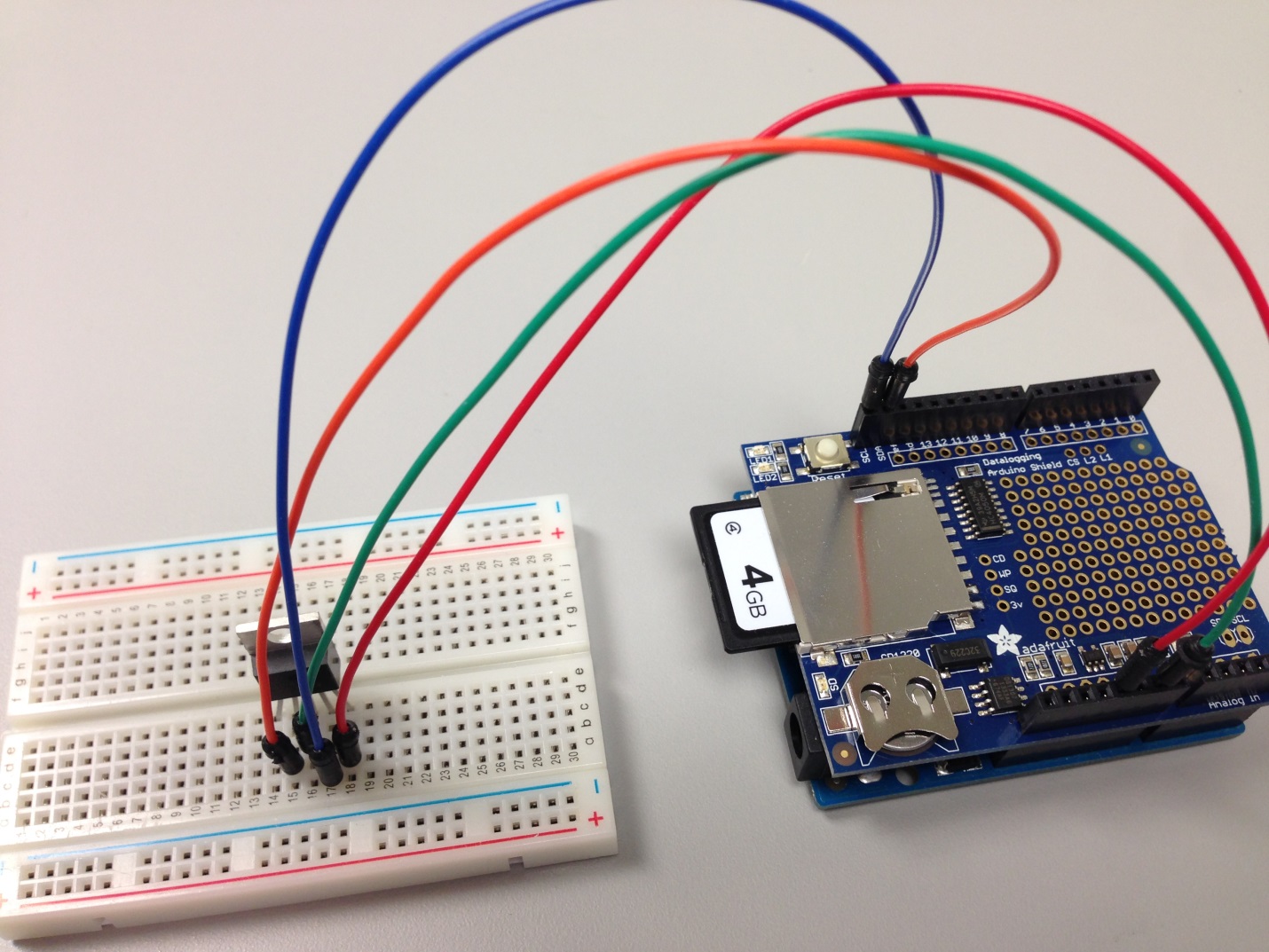


Figure - Temperature sensor wired to the Arduino data logger board

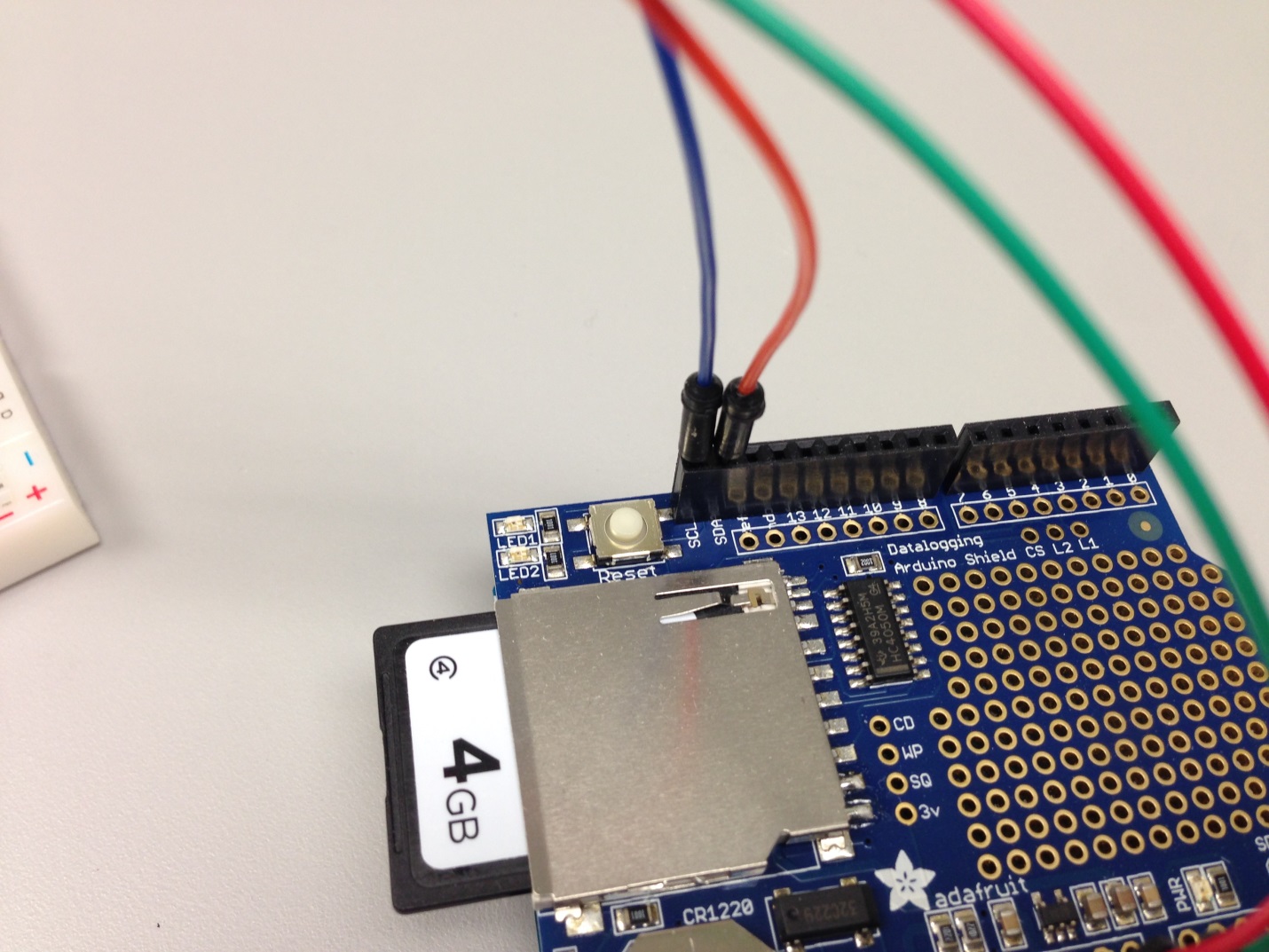


Figure - SDA & SCL connection detail

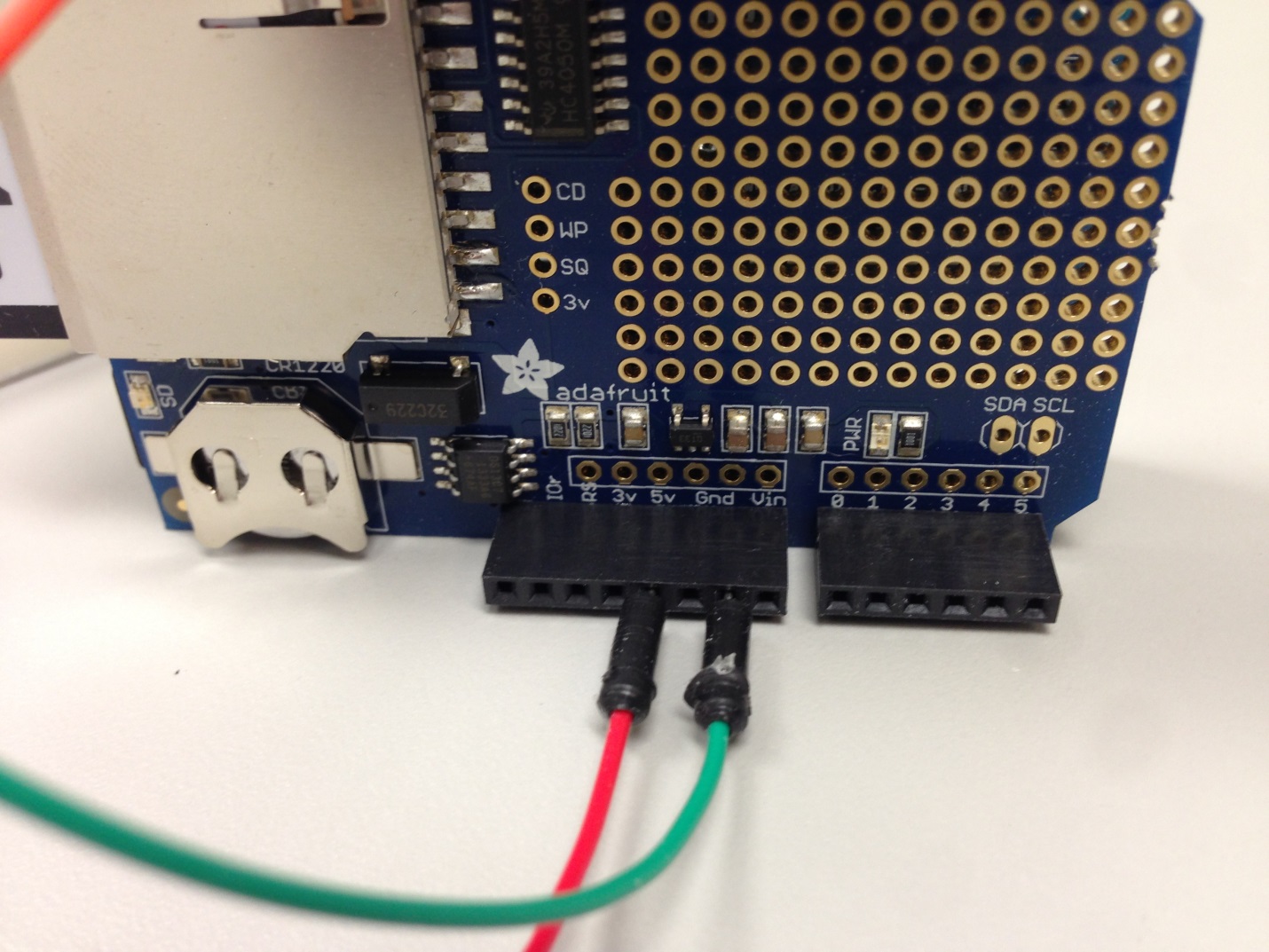


Figure - 5V & Gnd connection detail

1. Doublecheck your wiring. A mistake here could damage the Arduino, data logger board, or the temperature sensor.
2. Your TC74 is now ready to communicate with the Arduino.
3. Reconnect the USB cable to the Arduino, being careful not to dislodge any of your jumper wires to the proto board.
4. Confirm the basic operation of the SD memory card by compiling and running the CardInfo program located in the IDE under File/Examples/SD/CardInfo. Note that you must change the chipSelect value from 4 to 10 on line 34 prior to compiling your program. Open a terminal window to observe the output. You should see textual output indicating that the card is present, size information, etc. Close the terminal window when you are finished.
5. Confirm basic operation of the RTC by compiling and running the ds1307 program located in the IDE under File/Examples/RTClibmaster/ds1307. Note that you must change the serial port baud rate from 57600 to 9600 on line 10 prior to compiling your program. Open a terminal window to observe the output. You should see textual output of the current date and time every three seconds. Close the terminal window when you are finished. [Note: you can use other baud rates for the terminal window; you just need to be consistent between the code and the terminal emulator on the computer.]
6. Confirm basic operation of the overall data logger board and temperature sensor.
   1. A simple program, titled Data\_Logger\_code, has been developed for your use in this lab. It is a compilation of three individual programs available to you in the Arduino development environment as noted in the Data\_Logger\_code program. It is located in the IDE under File/Sketchbook/Data\_Logger\_code. Note: ensure that your temperature sensor device address on line 96 of Data\_Logger\_code matches your device ID as follows:
      1. if you have a temp sensor part number that includes “A0”, then line 96 is already correct and does not need to be changed
      2. if you have a temp sensor part number that includes “A2”, then the variable on line 96 must be changed from ‘address1’ to ‘address3’ to switch the value (see lines 30-35 in the source code for further info)
   2. Compile and run the program.
   3. Open a terminal window to observe the output. You should see a textual display of the current date, time and temperature along with the logfile entry of textual data that will be written to the SD card every two seconds. The data log entries each contain: 1) time stamp (Unix time (seconds since Jan 1st 1970)), and 2) temperature (degrees C). Note that a text file “datalog.txt” is created by the program to record these logfile entries.
7. While the data logger is running, observe a rise in ambient temperature by gently holding the silver metal heatsink on the temperature sensor with your fingers for several seconds. You should observe a rise in the temperature being displayed. Release the sensor once you have observed this behavior. You should now see the temperature slowly returning to room temperature. (note: as a data point, 25C = 77F). Close the terminal window when you are finished.
8. Confirm the collection of your temperature data on the SD card by compiling and running the DumpFile program located in the IDE under File/Examples/SD/DumpFile. Note that you must change the chipSelect value from 4 to 10 on line 29 prior to compiling the program. Open a terminal window to observe the output. You should see textual output with one row of data for each log file entry that was made. Close the terminal window when you are finished.
9. This concludes the main portion of the Data Logger lab. If there is time left, you can explore whatever interests you. Some possibilities:
   1. Examine your data in Microsoft Excel.
      1. Unplug your Arduino and remove your SD memory card. Enable the write-protect switch on the card and insert it into an SD card reader on your PC. Copy the datalog.txt file from the SD card onto your PC. Open Microsoft Excel, import and plot your data.
   2. Add a light sensor to your data logger.
      1. Using the photocell from your Sunfounder kit, add a light sensor to your data logger. You should use one of the unused analog input lines to monitor the amount of light reaching the sensor. Don’t forget a pullup resistor to create a voltage divider for sensing the variable voltage drop across the photocell. (Let me know if you need help on this)
   3. Add a manual control for data logging events.
      1. Use the pushbutton switch in the Sunfounder kit to signal when you want to take a data sample. Modify your main loop rountine to monitor for this switch press instead of the two-second delay. Use an unused digital input line and enable the internal pullup resistor for that input on the Arduino. Tie the other end of the switch to ground.

**APPENDIX A – Data Logger Project: Common Gotchas**

1. Correct naming of RTC Library
   1. No special characters
2. RTC Library installed in correct directory level
   1. /Your\_Directory/Arduino/libraries/RTClibmaster
3. Correct (& consistent) baud rate on the serial terminal (9600) (such as: Serial.begin(9600);)
4. SD card chip select line must be set to 10 in the software (currently set to 4)

**APPENDIX B - SD Card File Integrity**

Over the many times I have used an SD card with an Arduino data logger (cycling power, pulling cards, etc), I have only experienced a potential file corruption issue once. I’m not sure what caused it, but to resolve the issue quickly, I simply reformatted the SD card (and thus losing the data that had already been written.)

The corruption issue does exist. One precaution is to remove power from the Arduino before attempting to install or remove the SD card.

The only measure taken in this data logging program is to minimize the amount of time that the file is actually open. The code opens the file, writes the data, and then closes the file. I use the same approach with my home data logger system and it has run without problems for three years. Below is one link I found that discusses the problem. Among other tips, it mentions that the Sandisk SD cads seem to be particularly resistant to file corruption. (YMMV)

<http://www.raspberrypi.org/forums/viewtopic.php?f=28&t=36533>