8.1 Sending Data to an External Program

This section further demonstrates the capabilities of the Arduino by utilizing its ability to communicate over a serial cable with other programs besides the Arduino IDE. The external program we will use is called 'Processing' and the steps to install it are defined below. There are other programs (such as Python or LabVIEW) that could be used to communicate with the Arduino but 'Processing' is uniquely designed to work with the Arduino and its code is very similar to the Arduino's code.

“Processing is an open source language/ development tool for writing programs in other computers. Useful when you want those other computers to "talk" with an Arduino, for instance to display or save some data collected by the Arduino.” - Arduino Playground

Steps to get 'Processing':

1. The 'Processing' Libraries are located at the Arduino 'Playground' website here: http://playground.arduino.cc/Interfacing/Processing#.UyIPEtt38Yw
2. Go to the link above and follow the instructions on the website for installing the 'Processing' libraries on your computer.
3. The actual program is located here: http://processing.org/
4. Again, follow the instructions on the website to install 'Processing.'
5. Run processing in order to confirm it installed correctly.

Steps to set up the 'Graph' program:

1. Run the Arduino IDE.
2. Open 'Graph' located in the dropdown menu, 'Open -> 04.Communication -> Graph' or 'File -> Open -> examples -> 04.Communication -> Graph -> Graph.ino'.
3. About halfway down the program there is a line that looks like this: /* Processing code for this example
4. Copy all the code below this line and above the block of gibberish and past it into 'Processing.'
5. Uncomment the code by deleting all the '/*' and the '*/' at the start and end of the copied section of the code. (Everything between these is commented out)
6. Once those are deleted, the code should change color to confirm it is no longer commented.
7. Compile the code now, if it compiles correctly, 'Processing' should be able to run the 'Graph' program.

Steps for running the program:

1. Ensure 'Serial Monitor' is not running; processing cannot read the serial port if another program is reading from it.
2. Start the 'Graph' program in 'Processing'; you should see a window with a black background and a yellow output signal. This signal is being read from the potentiometer and its amplitude will vary as the potentiometer is adjusted.
3. If you do not see this graph, double check the serial port settings in 'Processing,' it will be a little different than the Arduino IDE and may take a little bit more effort to get working.
4. The serial port on 'Processing' is set to '0' by default, try incrementing the value by '1' and try again, repeat as necessary.

5. Note: You can stop 'Processing' and start 'Serial Monitor' from IDE in order to get a numerical representation of what signal is getting sent from the potentiometer.

The following diagram shows the flow of data in this setup. The Arduino IDE program transmits the code you used in section 7 to the Arduino over the serial port. The Arduino saves the code and starts running the program by sending inputs through the circuit and looking at the output signals it receives. The Arduino then sends the output signal to the computer across the serial cable and any program on the computer can now look at that information but only one-at-a-time can look at it. If the Arduino IDE is running the 'Serial Monitor' program, it will block any other program from looking at it. Same goes for the 'Processing' program. Basically its a first-come-first-serve basis to see which program gets to look at the serial port.
References and Contributors:

Written by Paul Ashby

Original contributors are Bret Comnes, Dan Lankow, and Andres La Rosa.

Full document is “Introduction to Micro-controllers” used for a lab in PH 315.