Using the SD library to log data

This example shows how to use the SD card Library to log data from three analog sensors to a SD card. Please click here for more information on the SD library.

https://www.arduino.cc/en/Tutorial/Datalogger

Hardware Required

- Arduino or Genuino board
- Ethernet Shield (or other board with an SD slot)
- 3 10k ohm potentiometers
- hook-up wires
- breadboard
- Formatted SD card

Circuit

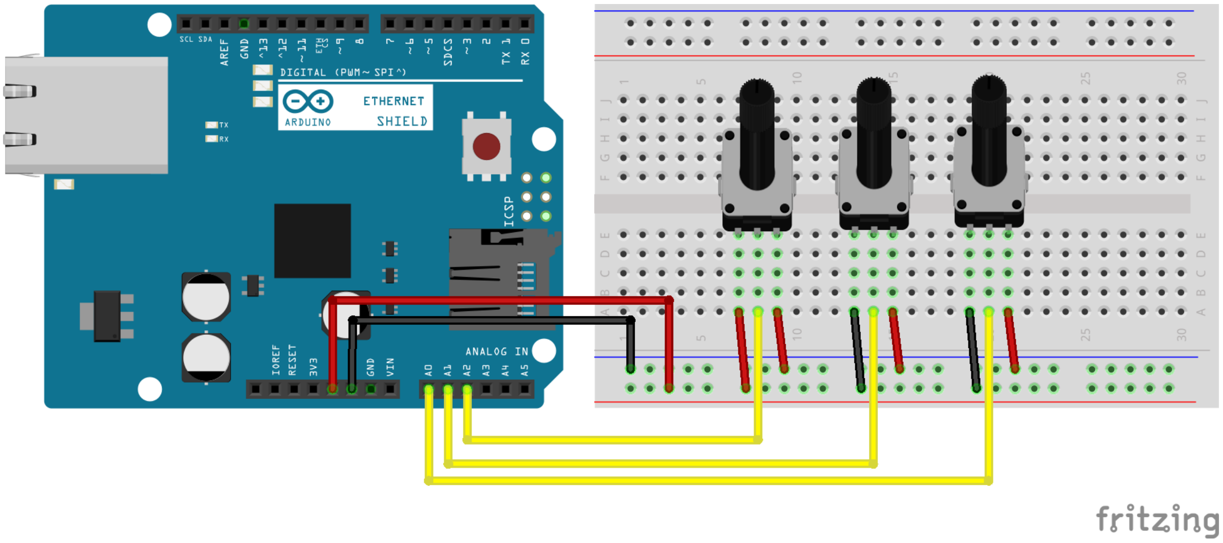

image developed using Fritzing. For more circuit examples, see the Fritzing project page

The Arduino or Genuino board has to be connected to the Ethernet Shield. The three potentiometers may be substituted with other analog sensors, to be wired according to their interfacing techniques.

Schematics

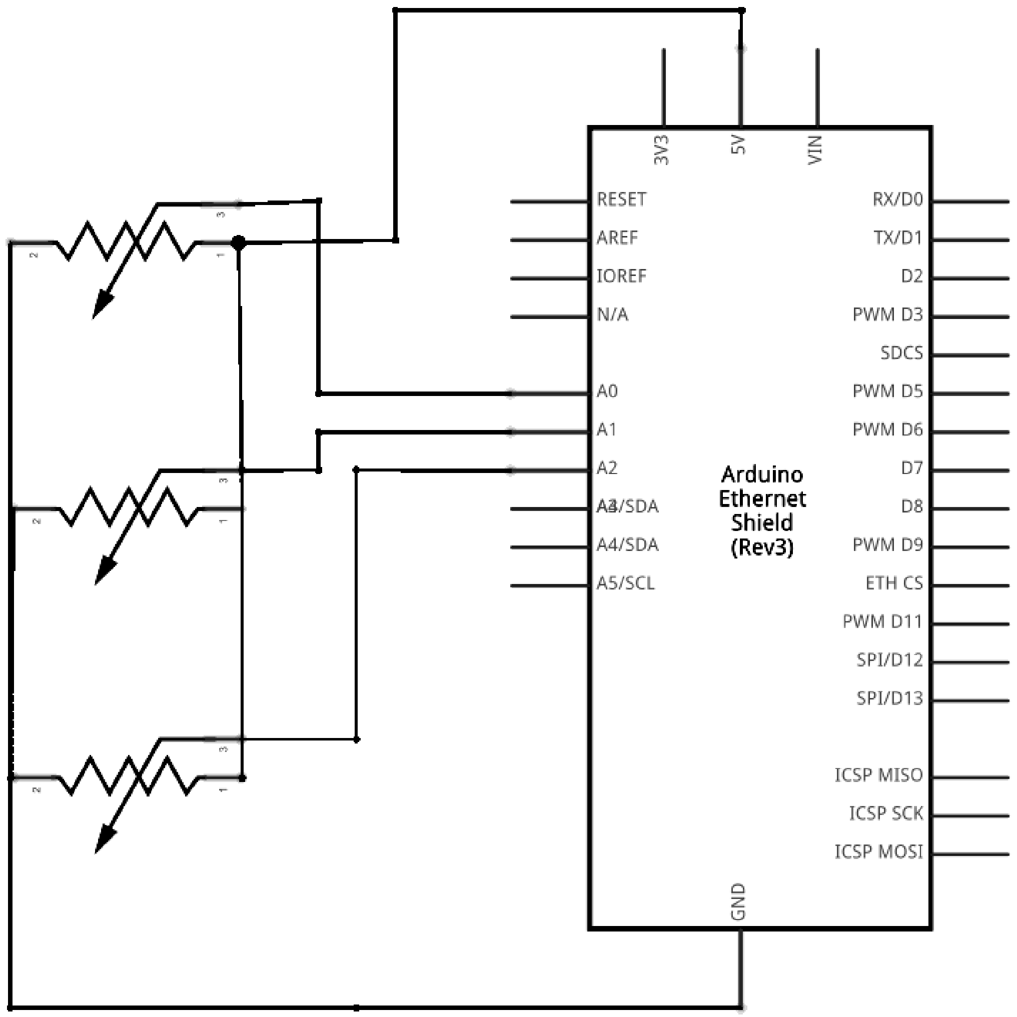

image developed using Fritzing. For more circuit examples, see the Fritzing project page

Code

The code below is configured for use with an Ethernet shield, which has an onboard SD slot. In the `setup()`, call `SD.begin()`, naming pin 4 as the CS pin. This pin varies depending on the make of shield or board.

In the `loop()`, a String is created to hold the information from three analog sensors. The code iterates through the sensors, adding their data to the string.

Next, the file on the SD card is opened by calling `SD.open()`. Once available, the data is written to the card when `dataFile.println()` is used. The file must be closed with `dataFile.close()` to save the information.

```
/*
  SD card datalogger

 This example shows how to log data from three analog sensors
 to an SD card using the SD library.

 The circuit:
 * analog sensors on analog ins 0, 1, and 2
 * SD card attached to SPI bus as follows:
 ** MOSI - pin 11
 ** MISO - pin 12
 ** CLK - pin 13
 ** CS - pin 4 (for MKRZero SD: SDCARD_SS_PIN)

 created  24 Nov 2010
 modified 9 Apr 2012
 by Tom Igoe

 This example code is in the public domain.

 */

#include <SPI.h>
#include <SD.h>

const int chipSelect = 4;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }


  Serial.print("Initializing SD card...");

  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    while (1);
  }
  Serial.println("card initialized.");
}

void loop() {
  // make a string for assembling the data to log:
  String dataString = "";

  // read three sensors and append to the string:
  for (int analogPin = 0; analogPin < 3; analogPin++) {
    int sensor = analogRead(analogPin);
    dataString += String(sensor);
    if (analogPin < 2) {
      dataString += ",";
    }
  }

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  File dataFile = SD.open("datalog.txt", FILE_WRITE);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }
}
```

[Get Code]

See Also

- SD library - Your reference for the SD Card library.
- Notes on using SD cards – What you need to know when you use SD cards

- CardInfo - Get info about your SD card.
- DumpFile - How to read a file from the SD card.
- Files - How to create and destroy an SD card file.
- Listfiles - How print out the files in a directory on a SD card.
- ReadWrite - How to read and write data to and from an SD card.

Last revision 2015/08/17 by SM