```
/*
  SD card datalogger

 This example shows how to log data from three analog sensors
 to an SD card mounted on the YunShield/YΓΊn using the Bridge library.

 The circuit:
 * analog sensors on analog pins 0, 1 and 2
 * SD card attached to SD card slot of the YunShield/YΓΊn

 Prepare your SD card creating an empty folder in the SD root
 named "arduino". This will ensure that the YΓΊn will create a link
 to the SD to the "/mnt/sd" path.

 You can remove the SD card while the Linux and the
 sketch are running but be careful not to remove it while
 the system is writing to it.

 created  24 Nov 2010
 modified 9 Apr 2012
 by Tom Igoe
 adapted to the YΓΊn Bridge library 20 Jun 2013
 by Federico Vanzati
 modified  21 Jun 2013
 by Tom Igoe

 This example code is in the public domain.

 http://www.arduino.cc/en/Tutorial/YunDatalogger

 */

#include <FileIO.h>
#include <DHT.h>
#include <Process.h>
//Constants
#define DHTPIN A0     // what pin we're connected to
#define DHTTYPE DHT11   // DHT 22  (AM2302)
DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz Arduino


//Variables
int chk;
float hum;  //Stores humidity value
float temp; //Stores temperature value

void setup() {
  dht.begin();
  // Initialize the Bridge and the Console
  Bridge.begin();
  Console.begin();
  FileSystem.begin();

  while (!Console); // wait for Console port to connect.
  Console.println("Filesystem datalogger\n");
}


void loop() {
 Process p;
  // This command line runs the WifiStatus script, (/usr/bin/pretty-wifi-
info.lua), then
  // sends the result to the grep command to look for a line containing the word
  // "Signal:"  the result is passed to this sketch:
```

```
  // make a string that start with a timestamp for assembling the data to log:
  String dataString;
  dataString += getTimeStamp();
  dataString += " = ";
  delay(2000);
    //Read data and store it to variables hum and temp
    hum = dht.readHumidity();
    temp= dht.readTemperature();
    //Print temp and humidity values to Console monitor
    Console.print("Humidity: ");
    Console.print(hum);
    Console.print(" %, Temp: ");
    Console.print(temp);
    Console.println(" Celsius");
    delay(10000); //Delay 2 sec.
  // read three sensors and append to the string:
//  for (int analogPin = 0; analogPin < 1; analogPin++) {
  //  int sensor = analogRead(analogPin);
  //  dataString += String(sensor);
    // if (analogPin < 2) {
    //  dataString += ",";  // separate the values with a comma
  //  }
  // }
dataString +="Humidity: "+String(hum)+" %, Temp: "+String(temp);


  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  // The FileSystem card is mounted at the following "/mnt/FileSystema1"
  File dataFile = FileSystem.open("/mnt/sd/datalog1.txt", FILE_APPEND);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    dataFile.close();
 //   p.runShellCommand(". /mnt/sd/put_file")
 p.runShellCommand("date > file2");
     // do nothing until the process finishes, so you get the whole output:
while (p.running());

    // print to the Console port too:
    Console.println(dataString);

  }
  // if the file isn't open, pop up an error:
  else {
    Console.println("error opening datalog.txt");
  }

  delay(15000);

}

// This function return a string with the time stamp
String getTimeStamp() {
  String result;
  Process time;
  // date is a command line utility to get the date and the time
  // in different formats depending on the additional parameter
  time.begin("date");
  time.addParameter("+%D-%T");  // parameters: D for the complete date mm/dd/yy
  //             T for the time hh:mm:ss
  time.run();  // run the command
  // read the output of the command
  while (time.available() > 0) {
    char c = time.read();
    if (c != '\n') {
      result += c;
    }
  }

  return result;
}
```