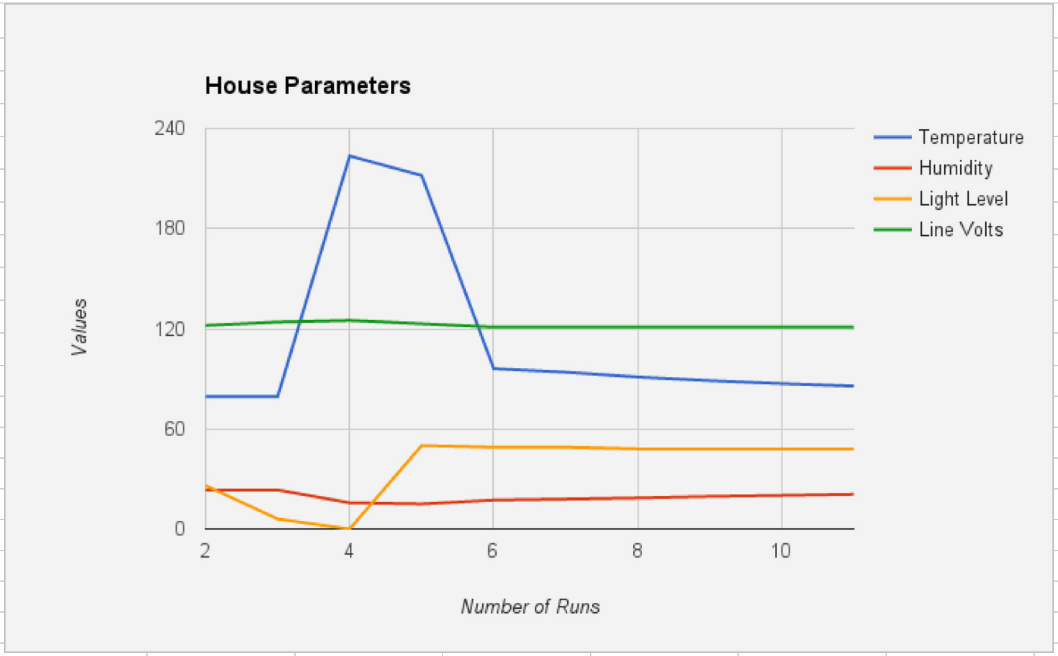
Number	Temperature	Humidity	Light Level	Line Volts
2	79.34	23.4	26	122
3	79.34	23.4	6	124
4	223.52	15.7	0	125
5	211.82	15	50	123
6	96.08	17.3	49	121
7	93.92	17.9	49	121
8	91.04	18.6	48	121
9	88.88	19.5	48	121
10	87.08	20.1	48	121
11	85.64	20.8	48	121



```
/*
************************************
*  Temboo Cloud Monitoring Station *
* ================================ *
*   By Roy H Guerra Jr.            *
*   2/21/16                        *
************************************
This code uses a Dagino Wi-Fi Board in "bridge mode" connected to an Arduino Uno to
monitor the light intensity, line voltage, temperature and humidity.
The results are transferred to a cloud server at the temboo Web Site (with previously
stored profiles)
and is based on Google API Drive, and places the measurements in a Google data
spreadsheet (seperatly created) that was contructed with a graphing function. In
addition, anther
choreo profile was created to send a Text to my phone when the hose Temperature > 90 Deg.
F

Hardware Required:
------------------
- Arduino Uno R3
- Dragino Yun Wi-Fi Shield
- DHT22 temperature sensor
- Photocell
- Misc (project case, power supply, etc.)

 The circuit has the following features:
```

```
 1) Connects and Transmits / Recieves through the house wireless router
 2) Connects to a Google cloud server webpage that could be pulled up anywhere
 3) Sends a Text to your phone if the temperature is above 90 degrees F
 4) Uses an unregulated DC input through a volatge divider that is calibrated and scaled
to read VAC rms

 Note(s)-
 * See my seperate instructions on the Dragino setup and Arduino Yun library installation
 * Set up a Google Temboo account, create a Google spreadsheet,  and Twillo account (for
texting)
 * Photocell tied to (+) and a series 10K resistor with one end tied to Gnd, other end to
"A0"
 * Line volts (DC) tied to "A1" (if this function is used, but also must have a battery
backup and voltage divider)
 * Set up a Google Temboo account, and log in the developers console and choose & set up
a Google Drive "API"
 * Run the Choreo's on the Google website after you have created to ensure Cloud site
will function and save the "profiles".
 * Once website and program are verified to run correctly, write and compile the program.
 * Activate the "console window within Arduino to debug as required.

 DHT 22 Sensor Wiring
 * Pin 1 (VDD)  = + 5v
 * Pin 2 = N/C
 * Pin 3 = Digital pin #7
 * Pin 4 = Gnd
*/
```

```
// Declare Libraries
// -----------------
#include <Bridge.h>
#include <Console.h>
#include <DHT.h>
#include <Process.h>
#include <Temboo.h>
#include "TembooAccount.h" // contains Temboo account information, as described below

// DHT sensor Characteristics
//---------------------------
#define DHTPIN 7     // Pin we are connected to
#define DHTTYPE DHT22   // DHT 22
DHT dht(DHTPIN, DHTTYPE);

// Light Level Sensor Characteristics
//-----------------------------------
#define LIGHT_SENSOR_PIN A0 // Pin we are connected to

// Voltage Input
//-----------------------------------
#define Voltage_PIN A1 // Pin we are connected to, Analog Input #1

// Declare Variables
// =================
int LightLevel; // Light Intensity storage Variable
```

```
int LightLevelSc; // Light Intensity storage Variable after scaling
int VoltLevel; // Light Intensity storage Variable
int VoltLevelSc; // Light Intensity storage Variable after scaling
float Humidity; // Humidity storage Variable
float TemperatureC;  // Temperature storage variable for Deg C
float TemperatureF;  // Temperature storage variable for Deg F
int numRuns = 1;   // Execution count, so this doesn't run forever
int maxRuns = 10;   // Maximum number of times the Choreo's should be executed

void setup() {
  pinMode(4, OUTPUT); // Set Device 1 feed as digital output (if used)
  pinMode(5, OUTPUT); // Set Device 2 feed as digital output (if used)
  digitalWrite(4,HIGH); // Turn Device 1 "Off"
  digitalWrite(5,HIGH); // Turn Device 2 "Off"
  dht.begin(); // Initialize DHT sensor
  Bridge.begin();  // Start Bridge
  Console.begin(); // Start Console
  Console.println("Setup complete. Waiting for sensor input");
  delay(4000); // 4 second delay
}

void loop() {
  if (numRuns <= maxRuns) {
    SensorData(); // go to Sensor Function
    Console.println("Running Program - Run #" + String(numRuns++));
    if (TemperatureF > 90){  // See if house is above 90 degrees F
       sendtext(); // Goto Text Sub-Function
```

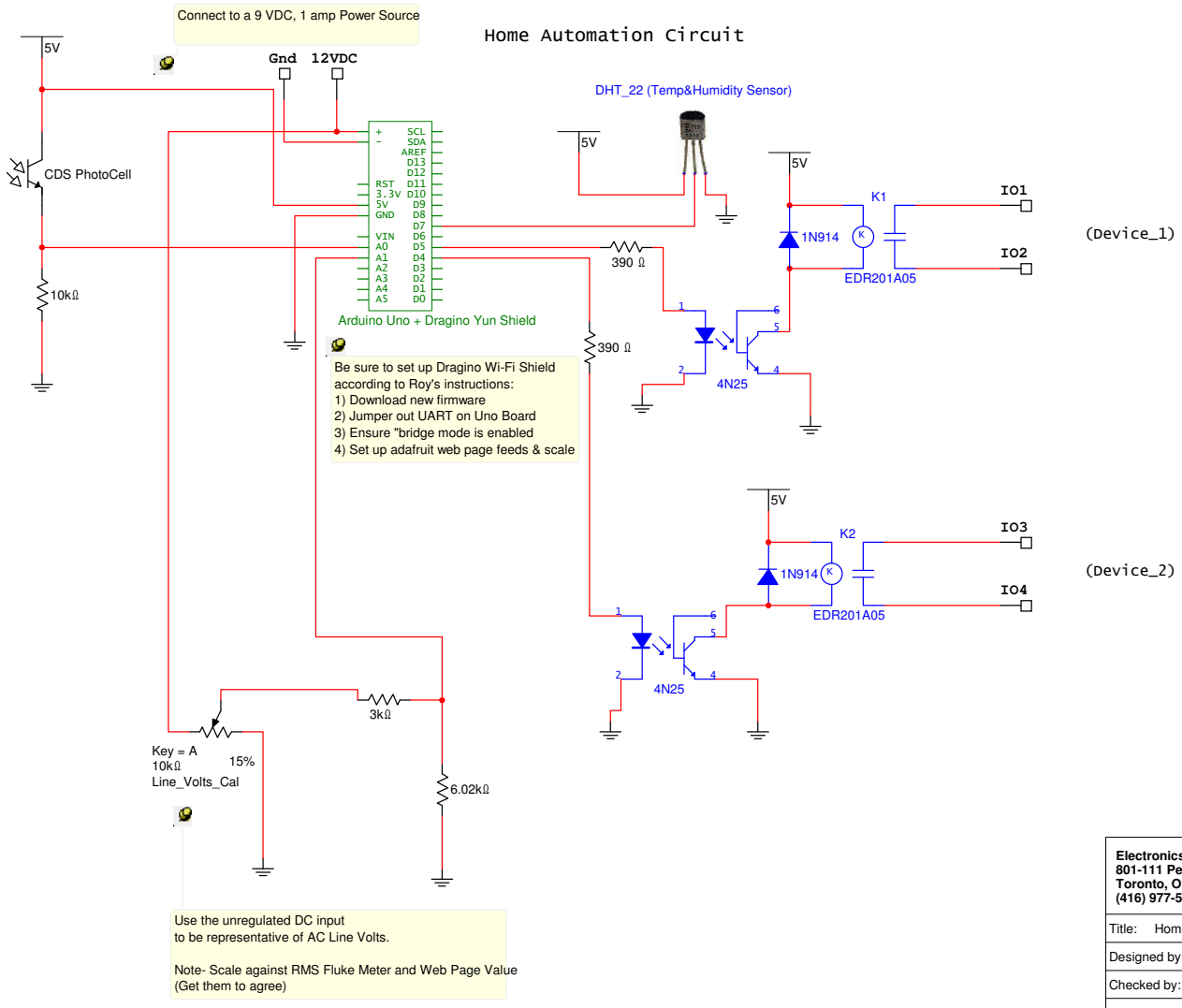
```
    }
    runAppendRow(); // Goto Function to Append data to Google Docs sheet
    delay(300000); // Repeat every 5 minutes
  }
}

void SensorData() {  //Read Sensor Function
  Humidity = dht.readHumidity();  // Measure the humidity
  TemperatureC = dht.readTemperature(); // Measure the temperature
  TemperatureF = ((TemperatureC * 9/5) + 32); // Convert temperature to degrees
Fahrenheit
  LightLevel = analogRead(LIGHT_SENSOR_PIN); // Measure light level
  LightLevelSc = map(LightLevel, 0, 1023, 0, 100);  // Scale light level (0-100%)
  VoltLevel = analogRead(Voltage_PIN); // Measure voltage level
  VoltLevelSc = map(VoltLevel, 0, 1023, 0, 150);  // Scale voltage level (0-150)
  // Print the measurements on console for debugging
    Console.println("Humidity: ");
    Console.println(Humidity);
    Console.println("Light level: ");
    Console.println(LightLevelSc);
    Console.println("Temperature: ");
    Console.println(TemperatureF);
    Console.println("AC Volts: ");
    Console.println(VoltLevelSc);
    Console.println("");
   // Compare temperature & humidity events and perform a check sum.
   if (isnan(TemperatureC) || isnan(Humidity)){
```

```
    Console.println("Bad Check Sum Value");
    }
}

void runAppendRow() { // Appendrow Sub-Function
 Console.println("Running AppendRow");

    // Run Append row Choreo
     TembooChoreo AppendRowChoreo;

    // Invoke the Temboo client
    AppendRowChoreo.begin();

    // Set Temboo account credentials
   AppendRowChoreo.setAccountName(TEMBOO_ACCOUNT);
   AppendRowChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
   AppendRowChoreo.setAppKey(TEMBOO_APP_KEY);

    // Set profile to use for execution ???
   AppendRowChoreo.setProfile("GoogAppendrow");

    // Format data
     String data;
     data += String(numRuns);
     data += ",";
     data += String(TemperatureF);
     data += ",";
```

```
   data += String(Humidity);
   data += ",";
   data += String(LightLevelSc);
   data += ",";
   data += String(VoltLevelSc);

  // Set Choreo inputs
 AppendRowChoreo.addInput("RowData", data);

  // Identify the Choreo to run
AppendRowChoreo.setChoreo("/Library/Google/Spreadsheets/AppendRow");

  // Run the Choreo; when results are available, print them
  unsigned int returnCode = AppendRowChoreo.run();

  // A return code of zero means everything worked
  if (returnCode == 0) {
    Console.println("Completed execution of the AppendRow Choreo.\n");
  }
 else {
  // A non-zero return code means there was an error
  // Read and print the error message
  while (AppendRowChoreo.available()) {
    char c = AppendRowChoreo.read();
    Console.print(c);
  }
}
```

```
 AppendRowChoreo.close();
}

void sendtext() { // Text Sub-Function
   Console.println("Running Text Choreo");

   TembooChoreo SendSMSChoreo;

   // Invoke the Temboo client
   SendSMSChoreo.begin();

   // Set Temboo account credentials
  SendSMSChoreo.setAccountName(TEMBOO_ACCOUNT);
  SendSMSChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
  SendSMSChoreo.setAppKey(TEMBOO_APP_KEY);

   // Set profile to use for execution
  SendSMSChoreo.setProfile("roytwillo");

   // Identify the Choreo to run
 SendSMSChoreo.setChoreo("/Library/Twilio/SMSMessages/SendSMS");

   // Run the Choreo; when results are available, print them to serial
   SendSMSChoreo.run();

   while(SendSMSChoreo.available()) {
     char c = SendSMSChoreo.read();
```

```
      Console.print(c);
    }
   SendSMSChoreo.close();
}
```



Electronics Workbench
801-111 Peter Street
Toronto, ON M5V 2H1
(416) 977-5550

Title: Home Automation	Desc.: Controls "2" devices and monitors house parameters	
Designed by: RHG	Document No: 0001	Revision: 1.0
Checked by: RHG	Date: 2016-02-25	Size: B
Approved by: RHG	Sheet 1 of 1	