

GEEN 1200 - Engineering Discovery 1
Electrical Engineering and Computer Engineering (EECE) Module
Laboratory 2 - Data Acquisition using the Arduino and DHT22 Sensor
Cristinel Ababei and Susan C. Schneider
Dept. of Electrical and Computer Engineering, Marquette University
Fall 2017

1. Objectives

Students will build a simple data acquisition system using a DHT22 humidity and temperature sensor and an Arduino Uno board connected to a host PC.

2. Introduction

In this project we connect the DHT22 sensor to the Arduino board, which reads the humidity and temperature data from the sensor, and then passes them to the host PC via the Serial port. The schematic diagram of this set-up is shown in Fig. 1

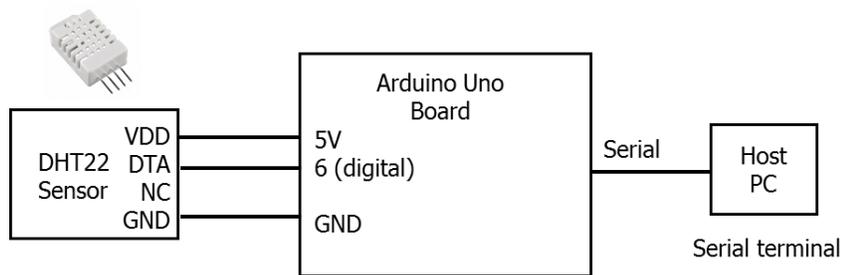


Figure 1: Schematic diagram.

The DHT22 sensor - also known as the RHT03 and shown in Fig.2, is a humidity and temperature sensor. The sensor is already calibrated and does not require extra components. Hence, we can use it directly to measure relative humidity and temperature.

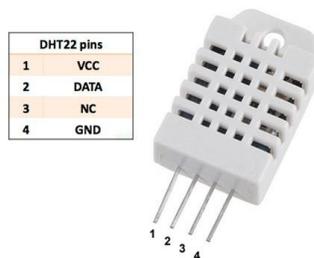


Figure 2: DHT22 humidity and temperature sensor.

The sensor has four terminals (VDD, DTA, NC, and GND). The communication between the sensor and the microcontroller is done through a single wire using a communication protocol described in the datasheet [1]. The actual sensor that we use in this project was purchased from SparkFun [2]. To communicate with the sensor, we will use the open source “code library” that was developed by Rob Tillaart. By using functions defined in the library, we will be able to easily retrieve the data from the sensor [3].

Materials Needed

<p>Hardware – DHT22 temperature/humidity sensor Arduino Uno, USB mini to USB B cable Experimenter’s wireless breadboard, connection wires</p>	<p>Software – Arduino IDE DHTLib – Arduino library to accompany the DHTxx sensor</p>
---	---

3. Preparation for Laboratory 2

To be completed PRIOR TO your lab session on 11/9/2017.

- Create the DHTLib library in the appropriate directory on your computer.
 - Basically, you need to create a new folder called **DHTLib** in the **libraries/** folder of your own Arduino installation directory. Two files, **dht.h** and **dht.cpp**, are placed inside the DHTLib folder. You can download these files directly from: <http://dejazzzer.com/geen1200/resources/DHTLib.zip>
 - Be able to answer (simple) questions on the library, like “what is the path for YOUR installation of the DHTLib library”, etc.
- Get datasheet for the DHT22.
 - Be able to find relevant information on this datasheet for things like pinout, power supply values, accuracy of humidity/temperature, number of bits of serial signal. Here is a copy of the datasheet: http://dejazzzer.com/geen1200/resources/RHT03_T_sensor_datasheet.pdf
- Type up the sketch using the code in Appendix A. Edit the comments appropriately. Store on your computer in the appropriate location.
 - Be able to answer questions on the code.
- Think about which set of extensions you want to work on. Start doing some preparation work.

4. In the Laboratory

Complete the following experiments. As each experiment is completed, show the working system to a teaching assistant for credit.

1) Experiment 1: Build and test the basic datalogger system

a) Hardware setup

Use the given components to assemble the entire system. Connect the DHT22 to the Arduino board via the breadboard using the connections shown in Fig.1. Your final hardware setup will look similar to that shown in Fig.3.

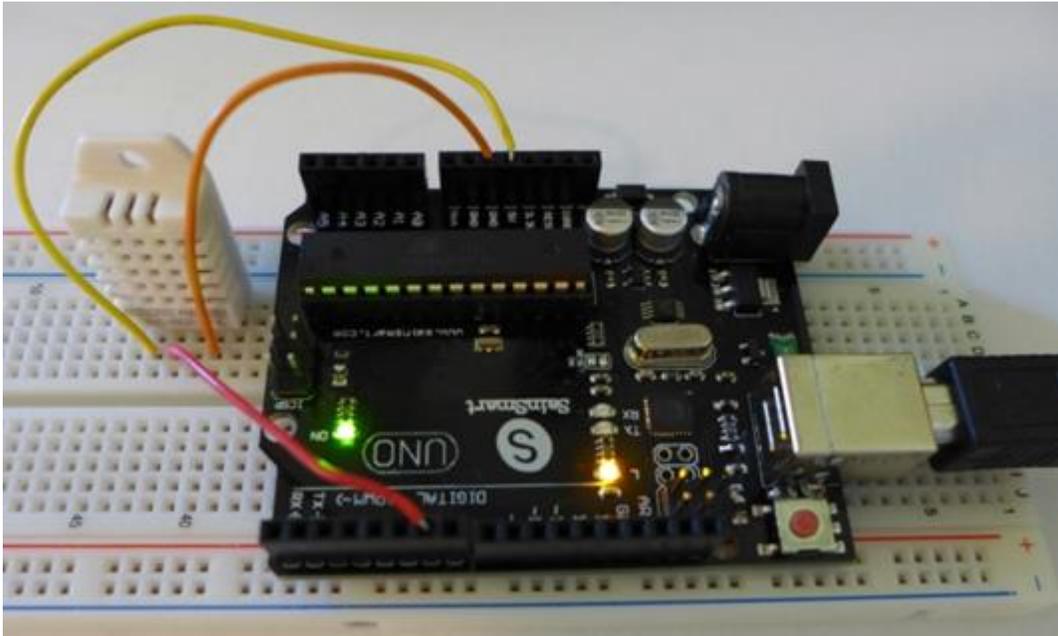


Figure 3: Protoboard layout.

b) Software (i.e., sketch)

Create the sketch **project_ece_freshmen.ino** using the listing in Appendix A. Save this *.ino file in a new folder with the same name as the file (project_ece_freshmen) that you create in your (Arduino) working directory.

c) **Verify and compile.** If your sketch has errors, debug and fix them.

d) **Upload compiled sketch to Arduino's MCU**

e) **Observe operation and plot data**

In the Arduino IDE, open the Serial Monitor tool. Check that a baudrate of 9600 is set to match the baudrate in the sketch. Restart the program by pressing the Reset button of the Arduino board. Observe the operation. You should see something similar to that shown in Fig. 4.

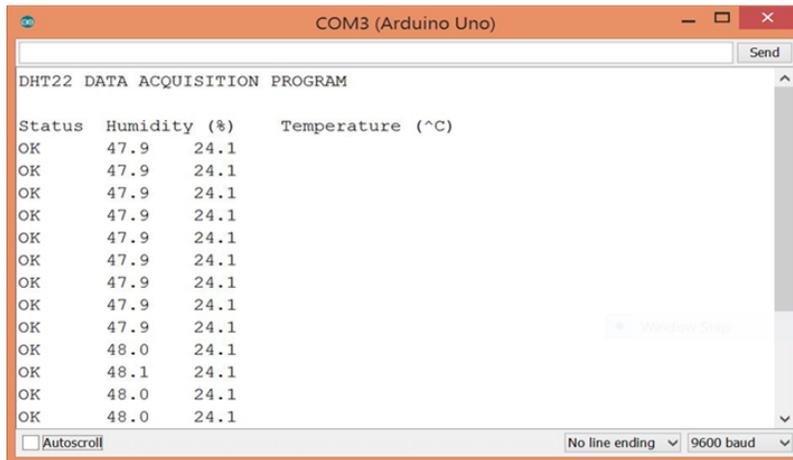


Figure 4: Snapshot of (Arduino IDE) Serial Monitor tool showing the humidity and temperature data.

2) Experiment 2: TWO Extensions

You get to choose which SET of two extensions you will complete to further demonstrate your mastery of introductory microcontroller data acquisition and analysis. As you complete each extension in your chosen set, demonstrate your working system to a teaching assistant for credit.

EXTENSION SET ONE

- Take temperature and humidity data for at least 5 minutes using a sample rate of approximately 1 sample every 10 seconds. During the 5 minute “run”, blow on the sensor several times to change the temperature/humidity.
- Copy the data from the serial monitor to an excel spreadsheet and produce a plot of temperature versus time and a second plot which shows humidity as a function of time.

EXTENSION SET TWO

- Modify the sketch to
 - display a sample number in the first column
 - display the temperature in Fahrenheit (°F) as well as in Celsius.
 - align the data with the column headings
- Modify the sketch to calculate and display “heat index” using the temperature and humidity data. *Hint: use the internet to find a formula for heat index.*

References

- Datasheet of DHT22 Sensor; <http://cdn.sparkfun.com/datasheets/Sensors/Weather/RHT03.pdf>
- DHT22 Humidity and Temperature Sensor; <https://www.sparkfun.com/products/10167>
- Rob Tillaart’s DHTLib; <http://playground.arduino.cc/Main/DHTLib>

Appendix A – Listing of project_ece_freshmen.ino sketch

```
// Sketch: project_ece_freshmen.ino
// copyleft :-)
//
// cristinel ababei; july.9.2015; rev.0;
// susan c. schneider; aug.20.2015; rev.1; oct.24.2016; rev.2
//
// Description:
// this is a simple Arduino project which uses a sensor to read
// temperature and humidity. These data are sent to the host PC
// via the serial connection and displayed in a hyperterminal
// such as the Arduino Serial Monitor;
//
// Credits:
// this program is adapted from the example that comes with the Arduino
// library developed by Rob Tillaart;
// the library can be found at http://arduino.cc/playground/Main/DHTLib
//
// Notes:
// remember, before verify/compile and upload, you must
// 1) select your actual Arduino board: Tools->Board->Make-Your-Selection
// 2) check-mark the right COM port: Tools->Port->Make-Your-Selection
//
//
// HARDWARE
// digital pin 6 of the Arduino board is connected to pin DTA of the
// DHT22 sensor, as shown in this diagram:
//
// | _____ |
// | DHT22 Sensor | Front view (grid side; see thru) of DHT22 sensor
// | 1 2 3 4 | Pin number (left to right)
// | VDD DTA NC GND | Pin name
// -----
// | | | |
// | | | |
//      ^is connected to digital pin 6 of Arduino board
//      ^is connected to 5V of Arduino
//      ^is connected to a GND pin of Arduino
//      ^is left disconnected (i.e., Not-Connected)

#include <dht.h>
dht DHT;
#define DHT22_PIN 6

void setup()
{
  // create serial connection;
  // CHECK Serial Monitor tool to confirm 9600 baudrate!
  // print Title and Column Headings
  Serial.begin(9600);
  Serial.println("DHT22 DATA ACQUISITION PROGRAM");
  Serial.println();
  Serial.println("Status \tHumidity (%) \tTemperature (^C)");
}

void loop()
```

```

{
  // (1) read data from the DHT22 sensor;
  // check that all is ok and print status;

  int chk = DHT.read22( DHT22_PIN);
  switch ( chk) {
    case DHTLIB_OK:
      Serial.print("OK \t");
      break;
    case DHTLIB_ERROR_CHECKSUM:
      Serial.print("Checksum error,\t");
      break;
    case DHTLIB_ERROR_TIMEOUT:
      Serial.print("Time out error \t");
      break;
    default:
      Serial.print("Unknown error \t");
      break;
  }

  // (2) send the humidity and temperature data as serial data;
  // these data can be seen using a serial monitor program
  Serial.print(DHT.humidity, 1);
  Serial.print(" \t");
  Serial.println(DHT.temperature, 1);

  // (3) wait for about 2 seconds;
  delay(2000);
}

```