

Internet of Things Workshop

Lab 1- Wire sensor

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
| 11/7/2016 | Steve Busby | 1.0 | Initial draft |

Table of Contents

[Change Record 2](#_Toc466281458)

[Introduction 4](#_Toc466281459)

[1.) Environment and device setup 5](#_Toc466281460)

[2.) Connected to and powering your RPI 9](#_Toc466281461)

[3) setting up the RPI for development 10](#_Toc466281462)

[4.) Testing the DHT22 temperature sensor 11](#_Toc466281463)

Introduction

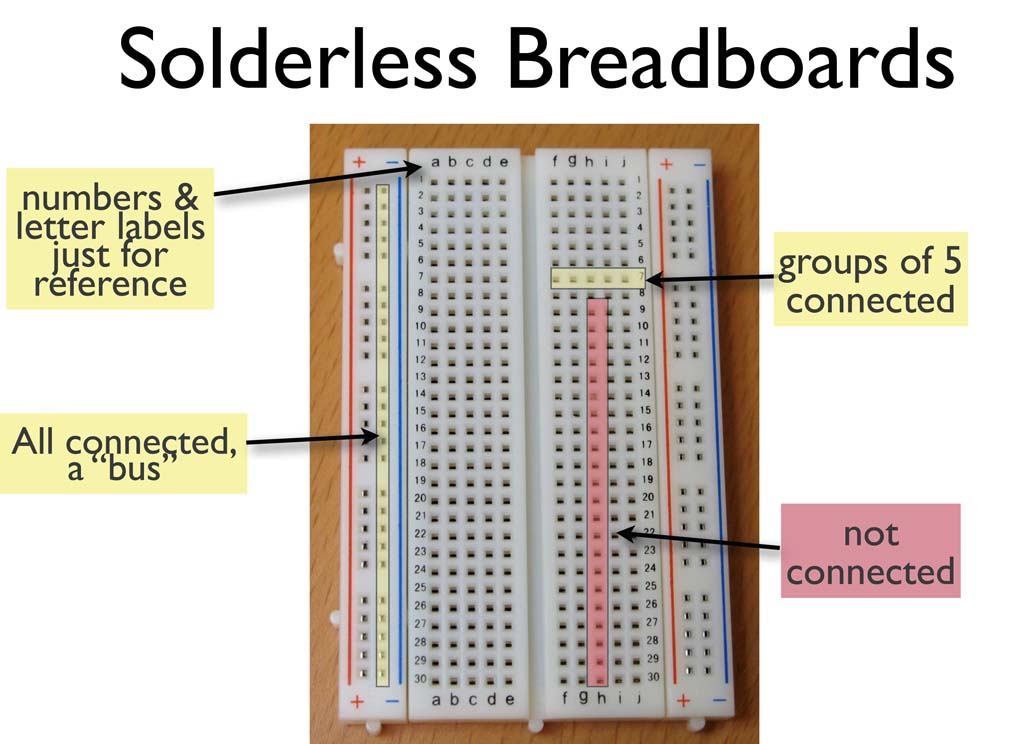
The purpose of this lab is to get the raspberry pi working with the DHT22 temperature sensor. We will leverage the Raspberry PI 3 and a DHT22 temperature and humidity sensor to represent our device.

1. Environment and device setup

In this section, we will make the physical connections between our Pi’s and the DHT22 temperature and humidity sensor. Once done, we will prepare our raspberry PIs for development and test our working sensor.

For the lab, we’ll first wire up the RPI before we power it on and connect to it. Like with most electronics, it’s always safer to do any wiring with the device powered off. If your RPI is powered on, remove the power at this point.

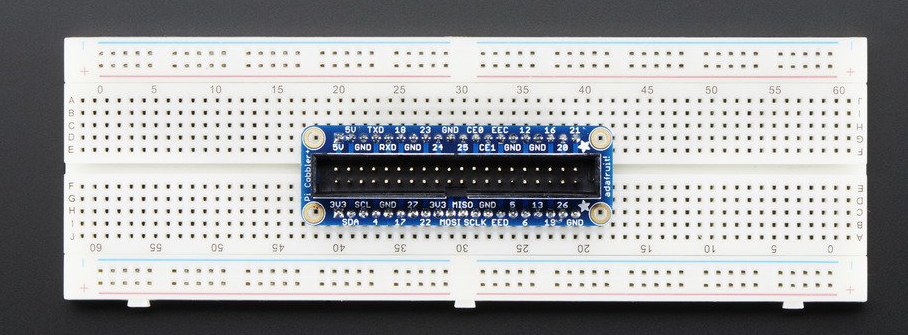
For our connections, we will be using a solderless breadboard for connecting the DHT22 temperature sensor to the RPI. If you aren’t familiar with a breadboard, they work like this:



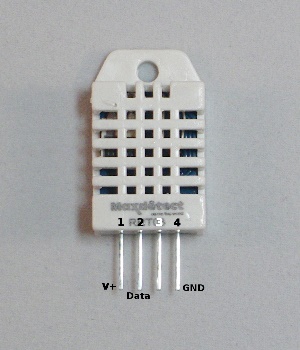
On a breadboard, the side rails (labeled + and -) are electrically “connected” all the way down the sides. They are generally used for power (+) and ground (-) or GND. In the middle section of the breadboard, each row of 5 connections are electrically connected together (but adjacent rows are not). So, for example, following the grid on the picture 1A, 1B, 1C, 1D, and 1E are connected together, but none of those pins are connected to either 2A, or 1F.

First let’s cover the electrical components involved.

Your kit comes with what’s called a “cobbler”. The purpose of the cobbler is to bridge the connections between your RPI and the breadboard. The cobbler plugs into the breadboard (as shown below), and then connects via the supplied ribbon cable to the ribbon connectors on the RPI



The DHT22 temperature and humidity sensor has pins that will plug into the breadboard directly.

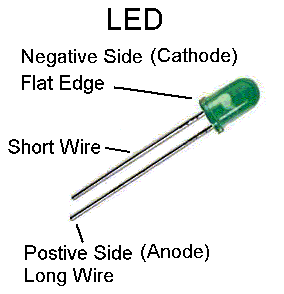


With the sensor facing you, like shown, the pins from left to right are:

1. V+/VCC – power. We will connect this to 5V power
2. DATA – this is the pin we will read the data from with the RPI
3. UNUSED – this pin is not used
4. GND – Ground. This pin will be connected to ground

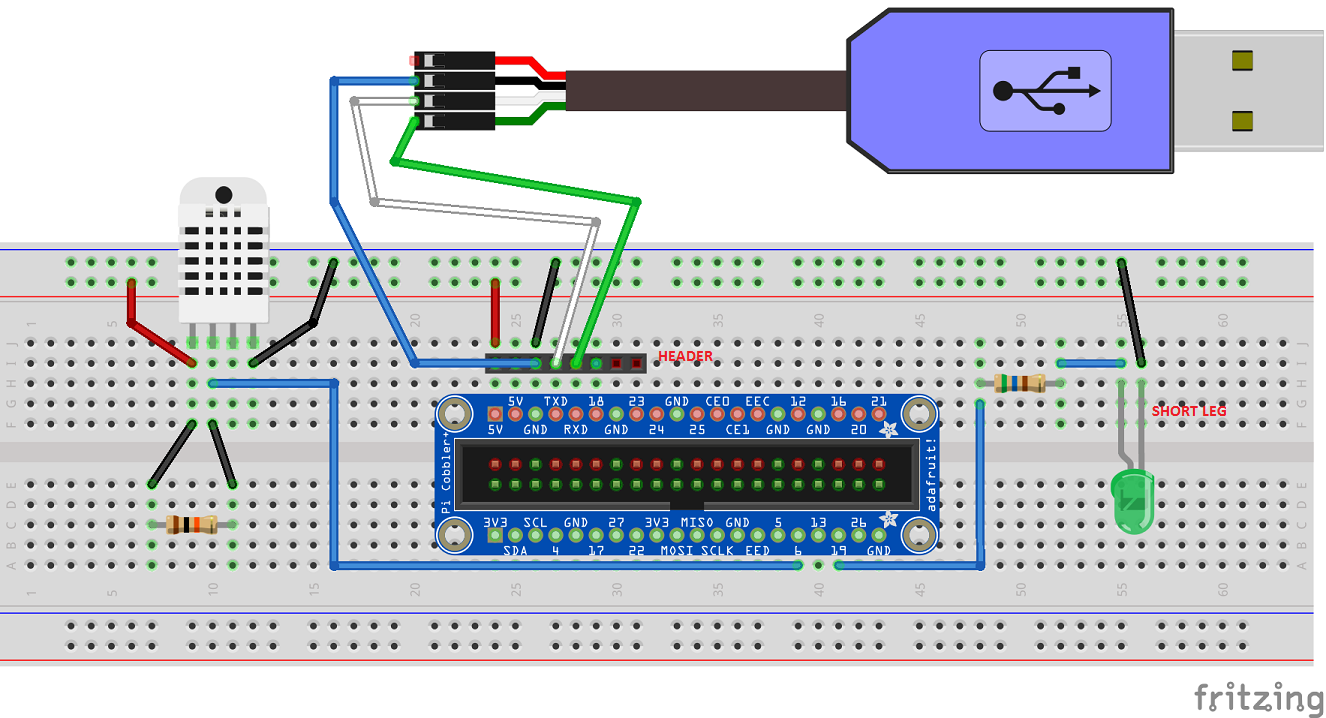
To keep the data pin from “floating” when it’s not being driven by either the RPI or the sensor, we will connect a 10k Ohm “pull up” resistor between the DATA pin and the power pin. This will ensure that, if the pin is in between data bits being transferred, it gets “pulled up” to 5V. This helps cut down on false or bad readings from the sensor.

And finally, we will use a light-emitting diode (LED) to represent our “high temperature alarm” indicator. In addition to lighting up, LEDs only allow power to flow in one direction, so we need to make sure they aren’t connected backwards. At these voltage levels, you won’t hurt the LED if you hook it backwards, it just won’t light.



As you can see from the picture, an LED has a positive side (anode) and a negative side (cathode). You can tell the difference because the cathode is the shorter of the two legs. The cathode must be connected to the “ground side” of the circuit with the anode connected to the “power side”. There may be other components of the circuit before or after the LED (in our case, a resistor), but the cathode in our case, will be connected to ground/GND. Finally, to keep the LED from drawing too much power from the RPI (the pins on an RPI can only ‘source’ a limited amount of power), we will connect a 560 Ohm resistor between the anode of the LED and the RPI pin that we use to drive it.

The picture below shows the connections that need to be made between your RPI cobbler and the different components, including the USB/TTL console cable (the USB cable with female wiring ends)



NOTE: this lab was written for a specific customer, where their network policy did not allow them to use SSH on their network. So they had to put the RPI on wireless, and use the console cable to connect from their laptop to their RPI. If you are on a network where you can SSH into the RPI directly, you can skip connecting the USB/TTL console cable.

A few notes about the connections

* The resistor that is connected to the DHT can be placed directly between pin 1 and pin 2 of the DHT sensor instead of having it in a separate part of the breadboard and wires running. My drawing tool just wouldn’t let me show it that way
* It’s hard to see, but the USB/TTL cable, because it has female ends, is connected to a header that is stuck in the breadboard (which sticks up with male ends). This is a picture:



* Also, hard to show in the diagram, but you don’t need to connect wires from the USB/TTL cable to the headers, you should just plug the cable ends directly into the header. NOTE – if you are not provided with a header, you’ll need to just plug a male to male wire into the breadboard and each lead from the USB/TTL cable

That’s it for the connections. If wired wrong, you can damage the DHT22 sensor, so at this point, before you fire up your RPI, feel free to call on the proctor(s) to double check your wiring. You don’t want to ‘let the magic smoke out of the box’ ☺

2.) Connected to and powering your RPI

NOTE: this lab was written for a specific customer, where their network policy did not allow them to use SSH on their network. So they had to put the RPI on wireless, and use the console cable to connect from their laptop to their RPI. If you are on a network where you can SSH into the RPI directly, you can skip this section.

At this point, we are almost ready to power up our RPI. Before we do, let’s connect our console cable so we can watch the PI boot.

While our RPI will connect over Wifi to the Internet (and Azure), and our laptops will also connect over Wifi to the Internet on the same network, corporate policy will not allow you to SSH from your laptop to the RPI directly. So, we must connect over serial console with the USB/TTL cable.

To use the cable, we must install a driver, and do some configuration (and install a terminal client, in the case of Windows).

To set up the connection, follow the instructions (with the caveats below) at <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-5-using-a-console-cable?view=all>

Caveats:

* The serial console has already been enabled on your RPIs, so you can skip down to the software installation section about mid-way down the page
* Make sure you get the right drivers for your machine (PC vs. Mac)
* For the actual connection of the console cable to your RPI, follow the picture in the previous section, not the instructions on the site
* Do NOT connect the red cable from the USB/TTL cable. We will be powering our RPI externally.

Once the PI is up and going. Log in with username ‘pi’ and password ‘raspberry’. To check network connectivity, try ‘sudo ping 8.8.8.8’ (CTRL-C to stop).

Congrats! You and up and running and ready to talk to Azure!

3) setting up the RPI for development

OPTIONAL -- check with instructor to see if you need to do this….. (may already be done for you…)

1. Install Python

sudo apt-get update

sudo apt-get install build-essential python-dev

1. Install drivers for the temperature sensor (DHT)

cd ~

git clone –recursive <https://github.com/adafruit/Adafruit_Python_DHT.git>

cd Adafruit\_Python\_DHT

sudo python setup.py install

1. Install Azure IoT Hub device SDK for Python

cd ~

git clone –recursive <https://github.com/Azure/azure-iot-sdks.git>

cd azure-iot-sdks/python/build\_all/linux

./setup.sh

1. Pull down the hands-on labs

cd ~

git clone –recursive <https://github.com/stevebus/UTCWorkshop>

4.) Testing the DHT22 temperature sensor

Once we have the DHT22 sensor hooked up properly, and the development environment prepped, we can test to make sure we are talking to the DHT22 sensor properly.

1. SSH into the raspberry PI
2. CD ~/Adafruit\_Python\_DHT/examples
3. Edit simpletest.py (‘nano simpletest.py’)
   1. Make sure line 31 is commented out
   2. Uncomment line 35 and set it to ‘pin=6’
   3. CTRL-O and then CTRL-X to save and close
4. Run the simpletest.py script (‘python simpletest.py’)
5. You should see a temperature and humidity reading

Congratulations – you’ve got the temperature sensor hooked up and working at this point, and are prepared to start sending data to IoTHub