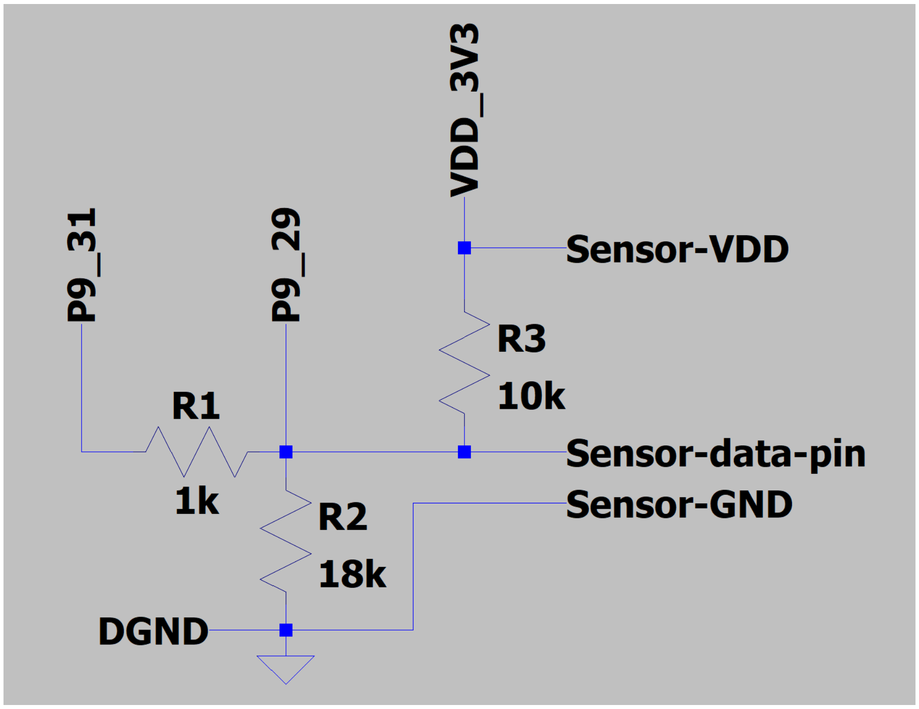
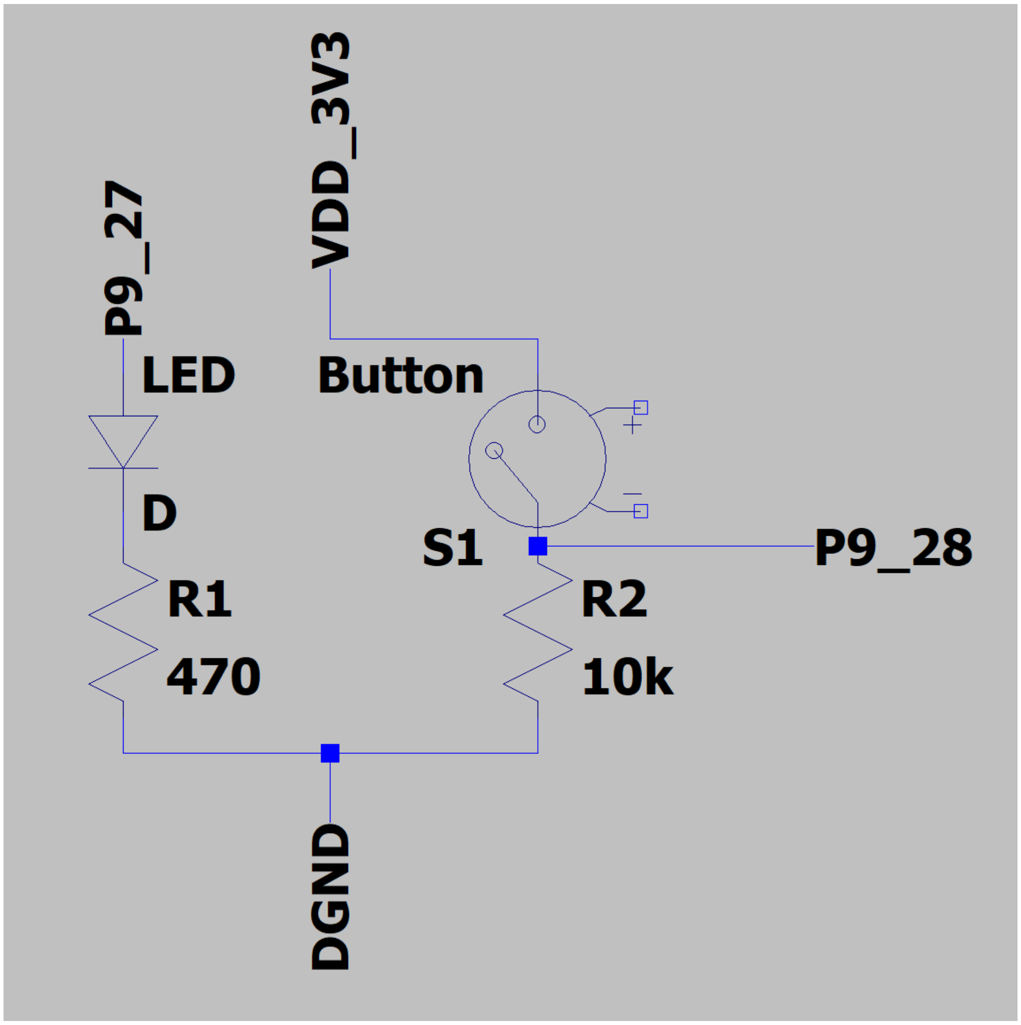
How To: Use the RHT03 Sensor
You will need:

- The RHT03 (**RHT**) temperature and humidity sensor
- Your Beaglebone Green (**BBG**) with all of the quick start guides completed
  - You must have superuser access to the BBG
- Resistors:
  - 1 × 1k (from ENSC lab or other provider)
  - 1 × 10k (from kit)
  - 1 × 18k (from ENSC lab or other provider)
  - 1 × 470 (optional) (from kit)
    - For debugging wiring setup with the Programmable Real-time Unit (**PRU**)
      - You will learn about the PRU later in the course; this guide will allow you to use it right away for easy access
  - Extra 1 × 10k (optional) (from kit)
    - For debugging your wiring setup with the PRU
- Button (optional)
  - For debugging your wiring setup with the PRU
- LED (optional)
  - For debugging your wiring setup with the PRU
- Wire

Wiring:
Schematic:

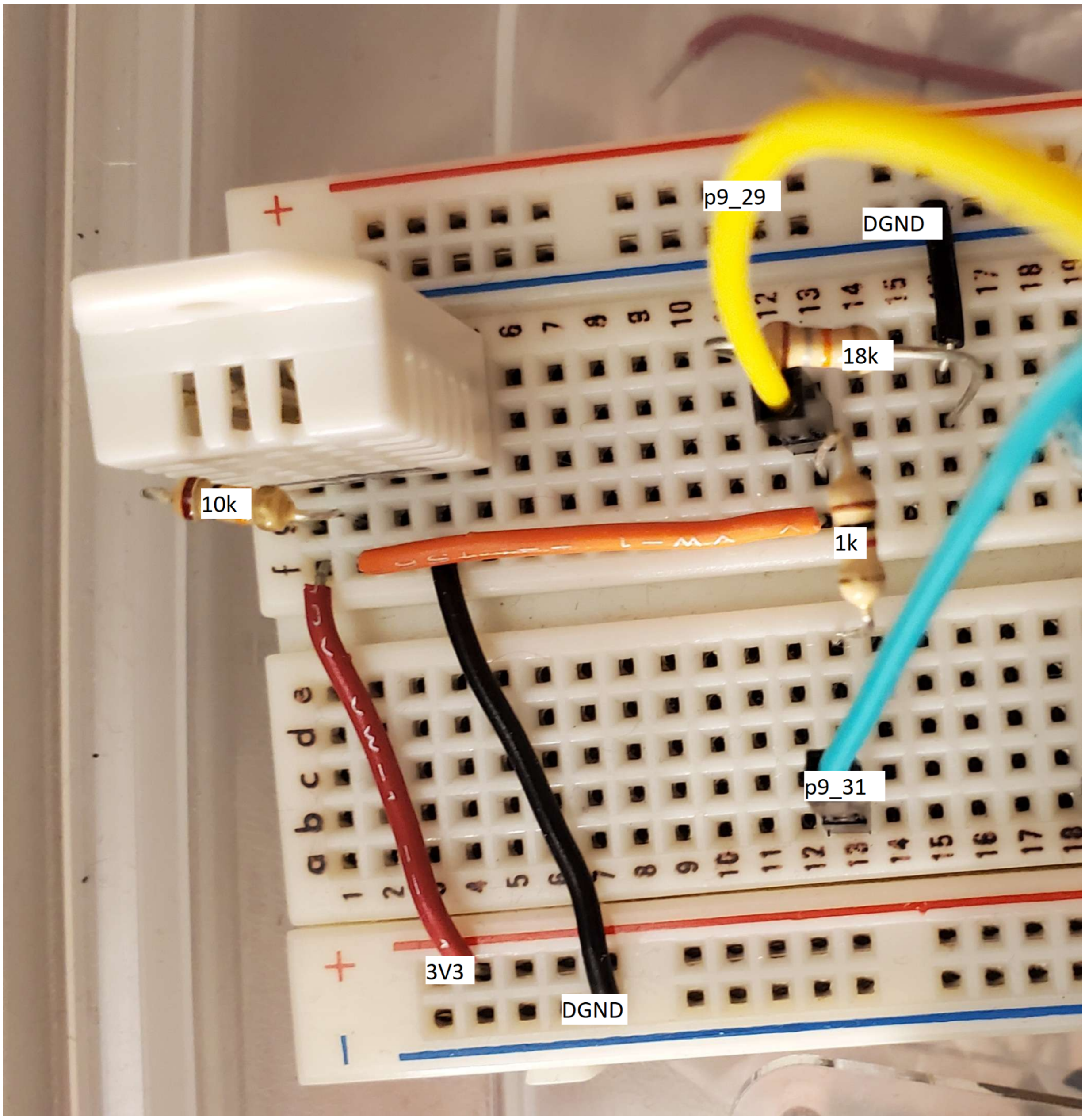


Wiring the sensor; node labels are BBG pins or sensor pins

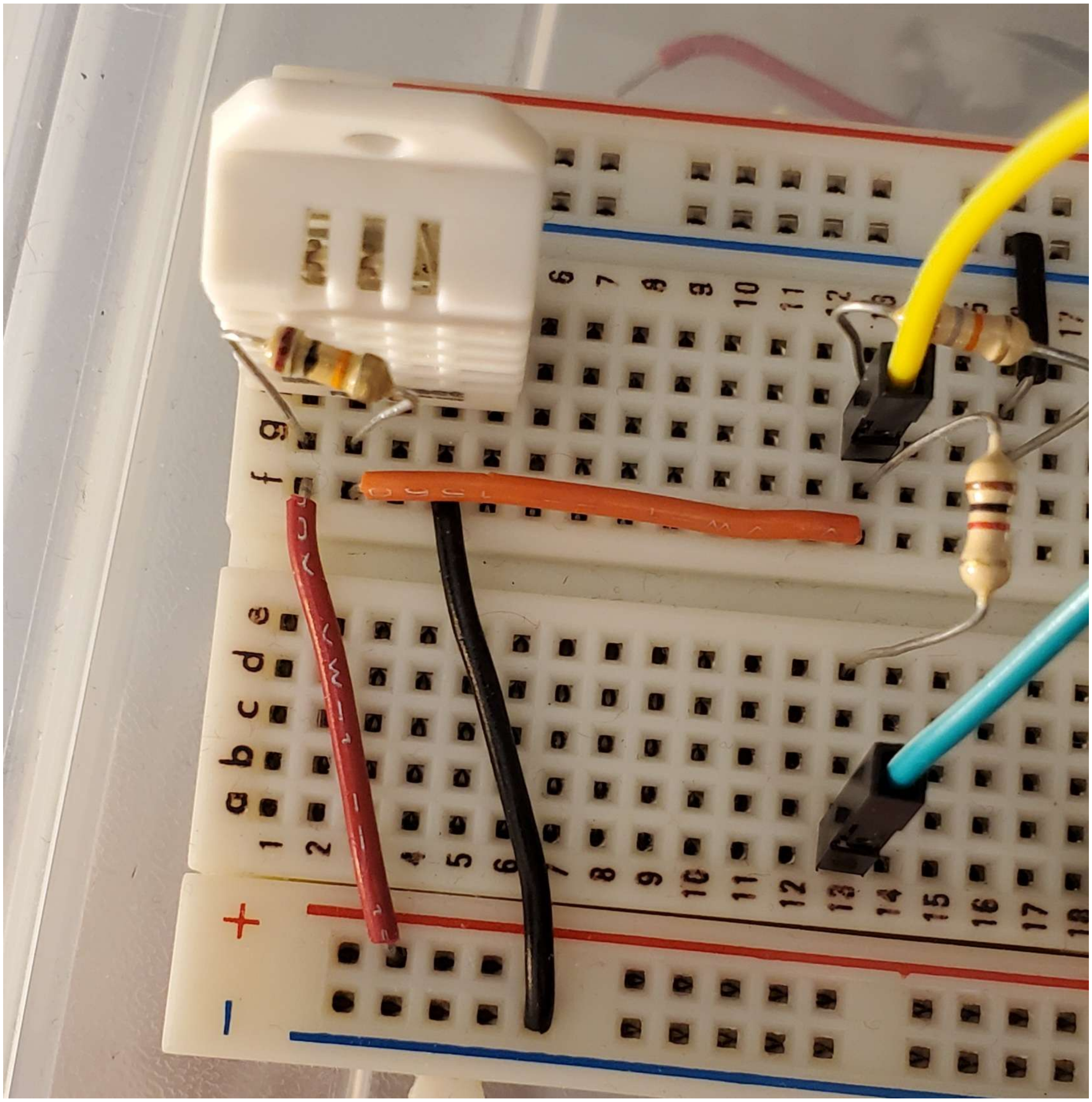


PRU debugging wiring setup; node labels are BBG pins or sensor pins

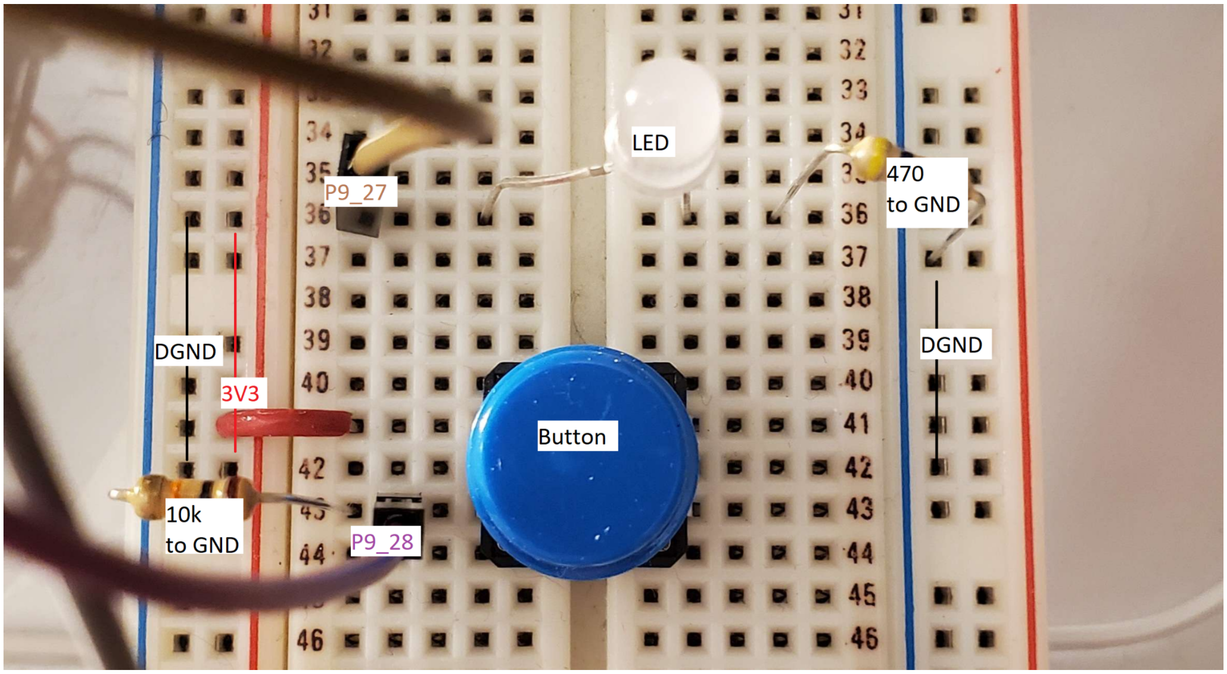
Example picture:



Wiring the sensor



Alternative angle



PRU debugging wiring setup

Running the sensor:

- Download the code provided from support files from the course website (home -> resources -> student how-to guides)
- Extract the .zip to your VM (**host**)
- To read data from the sensor, perform the following steps:
  (host) indicates a step to be run on the host; (bbg) indicates a step to be run on the BBG
  - (host)$ make
    - Make this directory on the host; ensure you started in the directory where you extracted the .zip to
    - Note that this makefile will assume the target directory for NFS is $(HOME)/cmpt433/public/myApps
      - If this is not so for you, either create this directory and configure it for NFS, or rename your current directory to this
  - (bbg)$ cd
    - This resets you to the home directory; if you are already here, you may skip this step
    - While you are here, make sure you initialize NFS if you are just logging in freshly
  - (bbg)$ cd /sys/class/remoteproc/remoteproc1
  - (bbg)$ echo 'start' | sudo tee ./state
    - Enter the superuser password; defaults to 'temppwd'
  - (bbg)$ cd
  - (bbg)$ cd /mnt/remote/pru/project-pru

  - (bbg)$ make
  - (bbg)$ make install_PRU0
    - Enter the superuser password
  - (bbg)$ cd ../../myApps
  - (bbg)$ sudo ./rht-sensor-demo
    - Enter the superuser password

Troubleshooting:

- If nothing works:
  - Verify the BBG is powered on
  - Verify the BBG's USB connection is fully inserted on both ends
  - Verify the BBG has been connected to your VM (if using a VM)
  - Verify the BBG has been set up
    - Internet to target is set up
    - NFS is set up
  - Try waiting a few seconds; after connecting the BBG, there is a ~10-30 second delay before it is ready to receive inputs
- If you get a "permission denied" error
  - Try running the command with "sudo" at the start
- If you receive an error message that the PRU timed out:
  - Double check your wiring setup
  - Double check that the PRU is running
    - (bbg)$ cd /sys/class/remoteproc/remoteproc1
    - (bbg)$ cat state
      - Should show "running" without quotes
  - Run the following to confirm pin statuses:
    - (bbg)$ config-pin -q p9_27
      - Should show pruout
    - (bbg)$ config-pin -q p9_28
      - Should show pruin
    - (bbg)$ config-pin -q p9_31
      - Should show pruout
    - (bbg)$ config-pin -q p9_29
      - Should show pruin
  - If all the above are correct, set DEBUG_PRU to 1 in both tempSensor.c and project-pru/pruMain.c
    - From there, find debugPru(void) and uncomment step by step to verify that each subsystem works
    - Make sure to set DEBUG_PRU to 0 again in both files once correct functionality has been confirmed
- If you receive an error that the checksum does not match the elements:
  - Double check your wiring setup
  - In tempSensor.c, set DEBUG to 1 to see what each element is in hex
  - Verify whether the sum of the non-checksum elements equals the checksum

    - If there would be overflow in your manual sum, truncate it off
  - In tempSensor.c, set DEBUG to greater than 1 to see exactly what elements were read by the sensor
  - In the ENSC lab, use an oscilloscope to verify behaviour; more tips for the oscilloscope are included below
    - Connect 1 probe to the junction between the sensor data pin, P9_31, and P9_29
    - Connect the other between P9_27 and the debugging LED
    - Wait until the trigger is activated
    - Zoom in and verify that the waveforms largely match

Oscilloscope tips:

- Reset and calibrate the scope
- Center both waveforms at 0V
- Connect a probe to the junction between the sensor data pin, P9_31, and P9_29
  - You should find that the shown waveform hovers around 3.3V
- Adjust the trigger to slightly below this waveform
  - Approximately 50-75% of the way from GND to the waveform is usually sufficient
  - Set the edge trigger to falling
- Adjust time/div so that ~200ms is shown
  - This provides sufficient reaction time for you to manually stop the scope
- Follow the steps to run the sensor listed above
- When the TRIGD light lights up, immediately press the STOP button
  - In the lab oscilloscopes used at time of writing, this is in the bottom right corner on the menu panel
- Find where in the data a large amount of oscillation occurs and focus in on it
  - Center that section of the data and adjust time/div until the waveform is clearly visible as highs and lows
- Following the sensor datasheet, you can manually read in a series of 1s and 0s to verify sensor behaviour
  - Typical lab conditions are ~23degC, ~15% humidity
- Compare what you manually read in, to the hex display in the terminal, by setting DEBUG to a value greater than 1

FAQ about this how-to guide:

Q: Why don't we use the provided readFromFile method on Linux?
A: The sensor operates on the microsecond scale; it needs to be read extremely quickly and continuously. Linux's readFromFile behaviour is too slow to read the sensor properly. Thus, this guide provides code so that you may use the sensor without having to wait until late in the term, after learning about the PRU, to use the sensor.

Q: How does the Linux code communicate with the sensor?
A: It uses a shared data struct with signaling flags. Once the PRU receives the "start" flag, it signals the RHT, reads, and signals Linux that it has finished.

Q: Where can I find the sensor datasheet?
A: You can either Google “RHT03 sensor datasheet”, or use the one included with the student guide. The datasheet is NOT written by the student making this guide; no credit or responsibility is taken in regards to datasheet contents.

Q: I ran outside with the sensor in the freezing cold, but it’s still showing the temperature from the lab where I was a moment ago!
A: The sensor takes time to update its temperature and humidity readings. In particular, the temperature readings seem to update very slowly; let it sit for about 5-10 minutes and it should show the right value.