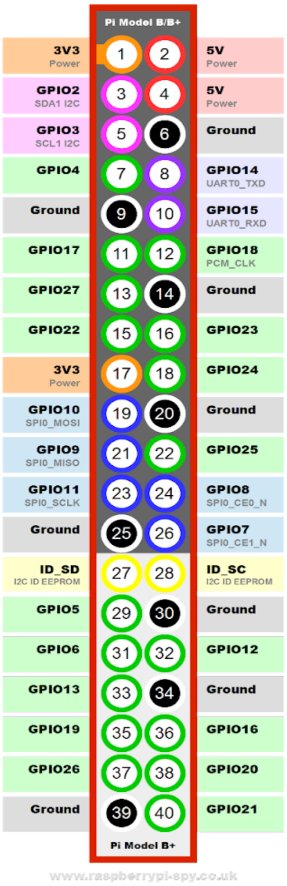
HAB.education Lab hand-out Name:__________________

Fill in the blank lines with your answers.

In this lab, you will program a temperature and humidity sensor for your high altitude balloon. The goal of this lab is to introduce you to assembling hardware, coding, and provide you with skills to explore more projects with inexpensive, easily available hardware such as the Raspberry Pi. You will learn terms useful for understanding how electronics work and how people use them. Finally, you will be able to then use your sensor to test its capabilities in preparation for its flight to nearspace.



Pi Model B/B+
3V3 Power 1 2 5V Power
GPIO2 SDA1 I2C 3 4 5V Power
GPIO3 SCL1 I2C 5 6 Ground
GPIO4 7 8 GPIO14 UART0_TXD
Ground 9 10 GPIO15 UART0_RXD
GPIO17 11 12 GPIO18 PCM_CLK
GPIO27 13 14 Ground
GPIO22 15 16 GPIO23
3V3 Power 17 18 GPIO24
GPIO10 SPI0_MOSI 19 20 Ground
GPIO9 SPI0_MISO 21 22 GPIO25
GPIO11 SPI0_SCLK 23 24 GPIO8 SPI0_CE0_N
Ground 25 26 GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM 27 28 ID_SC I2C ID EEPROM
GPIO5 29 30 Ground
GPIO6 31 32 GPIO12
GPIO13 33 34 Ground
GPIO19 35 36 GPIO16
GPIO26 37 38 GPIO20
Ground 39 40 GPIO21
Pi Model B+
www.raspberrypi-spy.co.uk

1. Identify sources of power. Read the pinouts diagram (topview) for your Raspberry Zero. The instructions below will help you figure out how to make your LED light up.

The LED has 2 legs. One is black, and one is yellow.
To light up the LED, you will need to complete the circuit which allows electrons to flow through the wires and LED.

- Connect the black wire to the **Ground** pin (indicated in black).
- You'll need the yellow wire to be connected to a pin that is a source of either **3.3V or 5V** of electricity.

What is the difference in light intensity you see when you complete the circuit with the **3.3V** pin compared to the **5V** pin?

____

- Try swapping the black and yellow wires pin connections. Does the LED still emit light? Yes or no?

____

- Swap it back. Try to connect the yellow pin to pin **11 (GPIO17)**. Does it still work? Yes or no?

____

2. Control the LED by using a software program instead of having to connect and de-connect the wires manually. Follow along more closely with your instructors' demonstration during this stage because there are many sequential steps that build upon one another.

- Make sure the LED black wire is connected to the Ground, and the yellow wire is connected to **pin 11 (GPIO 17)**.

- Get into the Node-red programming environment by clicking the browser icon, and type: **http://localhost:1880/** on your browser. You'll see the toolbox on the left, a work area in the middle, and a "debug" area (a space that reports steps as they are implemented when the program is running) on the right.

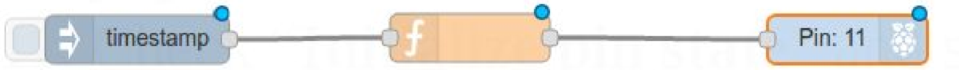
- Delete everything in the main work area. Click the **[+]** button on the top right corner to add a new blank work area.

- Drag "**=> inject ○**" from the toolbox to the work area. Double-click on it. Set "**payload**" to "**timestamp**," "**repeat**" to "**interval**," and "**every**" to **10** seconds. Check "**Inject once at start**." Click OK.

- Drag "**○ function ○**" to the work area, right next to the "**timestamp ○**" block you created. Double-click on it. Type in:
msg.payload=1;
on the line on top of **return msg;** Then click OK.

- Drag "**○ rpi-gpio [   ]**" right next to the "**○ function ○**" one you just created. Double-click on it. Set "**GPIO pin**" to **11**. Check "**Initialize pin state**" and set it to "**low**." Click OK.

- Now, use the mouse to draw the lines between the three blocks, so it looks like this:



- Click "Deploy" on top-right corner. What do you see?

____

- Remove the function (yellow) block from the work area by selecting it, then press Delete. We'll replace it with something that decides to turn the LED on or off by the temperature reported from the sensor.

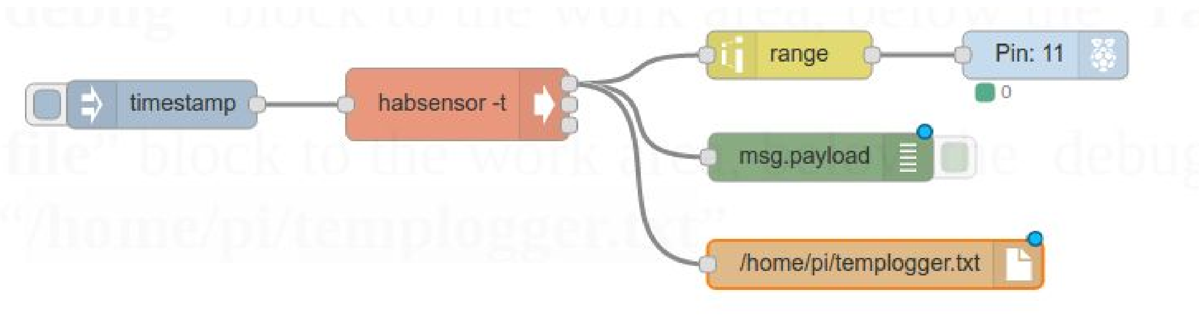
- Drag the "**○ exec ○**" block to the work area, right next to the **timestamp** block. Double-click on it. Type in in the "**Command**" area: **habsensor -t** (there is a space between the words). Click OK.

- Drag the "○ **range** ○" block to the work area, right next to the exec "**habsensor -t**" block. Set "**Action**" to "**Scale and limit...**" Then we want to "map input range" from 24 to 25 to "the result range" 0 to 1, and check "**round result to nearest integer**." Click OK.

- Drag the "○ **debug**" block to the work area, below the "**range**" block.

- Drag the "○ **file**" block to the work area, below the debug "○ **msg.payload"** block. In the "filename" field, set it to "**/home/pi/templogger.txt**" Click OK.

- Finally, we want to connect all of them together so it looks like this (notice how we connect the first output of "**habsensor -t**" to everything to its right):



- Click Deploy! Now, heat your sensor with your breath. Does it work?

---

3. Measure the pins' actual voltage levels with a voltmeter. Same as above, but you will have a black and a red wire, and you will have to work from the back of your board. Take extreme caution, make sure the tip of the voltmeter only touches ***one pin at a time***!

- Connect the black wire to the Ground pin.
- Connect the red wire to the 3.3V pin, what reading do you see?

____

- Connect the red wire to the 5V pin, what reading do you see?

____

Below, list three ways how you could test your sensor to determine if it will work in nearspace conditions?