Wind Speed, Pressure, & Humidity Sensors

ECE 476 Advanced Embedded Systems

Jake Glower - Lecture #23

Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

Introduction:

Three more things that can be useful to know

Wind Speed

- Important for airplane flight controls
- Evaluating a site for wind turbines,
- Evaluating if conditions are ripe for fungus growth

Air pressure is needed for

- Aircraft flight controls,
- Predicting weather, and
- Measuring wind speed with a pitot tube (coming shortly)

Humidity tells you

- How comfortable a given temperature is
- If the room conditions are OK for some plants and equipment

In this lecture, we'll look at

- Measuring wind speed with a
  - Fan,
  - Hot-wire anemometer, and
  - Pitot tube.
- Measuring air pressure using a
  - BMP280 digital sensor, and
- Measuring humidity using a
  - BMP280 digital sensor,
  - DHT11 digital sensor, and
  - DHT22 digital sensors.

There are many other ways to measure each of these - but this gives you an idea of some of the methods available to you.

Wind Speed: Fan

One way to measure wind speed.

- The faster you spin a fan, the faster with breeze you produce,
- The faster the breeze driving a fan, the faster the fan will spin

Essentially, remove the motor from a fan and you have a wind speed sensor.

Sensors of this type can be

- Non-directional
  - the direction of the wind doesn't matter
- Directional
  - the sensor must be perpendicular to the wind

In either case, the result is the same:

- If you can measure the speed of a fan, you can measure the wind's speed.

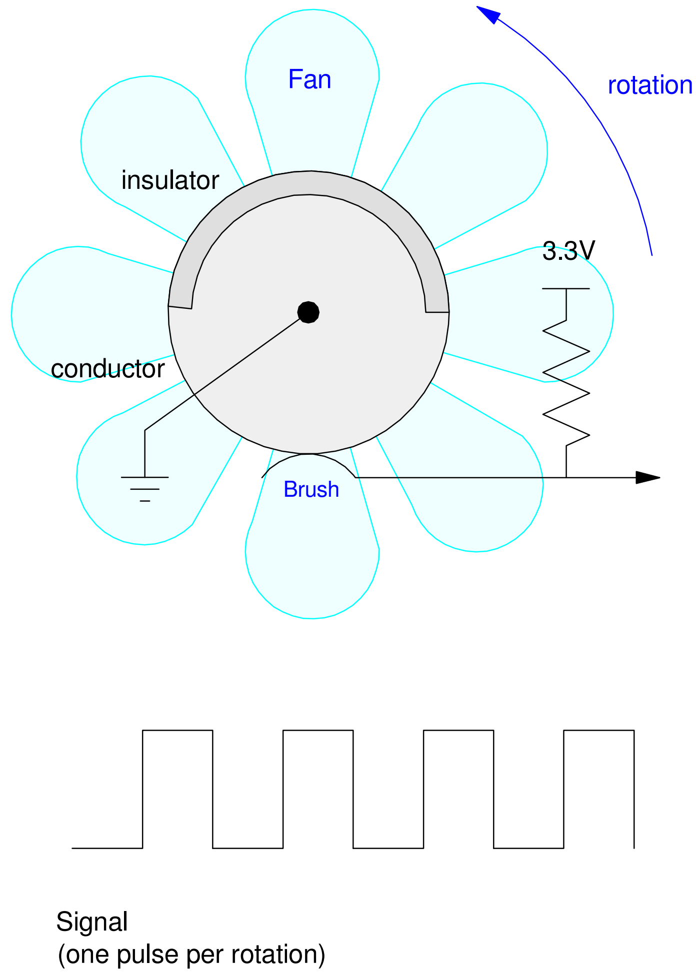
Fan Speed via Encoder

One way to measure fan speed it to output a square wave with one or more pulses per rotation.

By measuring the frequency (or period), you have a measurement of fan speed (and hence wind speed).

sidelight: computer fans usually include a sensor like this

- *If the fan is spinning (a square wave is being output), the fan is OK*
- *If the fain stops (no square wave is detected), shut off the computer to prevent overheating*



In code, you can use methods from lecture #1 to measure

- The time of the positive pulse on pin GP0 (x):
- The time of the negative pulse on GP0 (y), or
- The period (z)

```
from machine import Pin, time_pulse_us

Fan = Pin(0, Pin.IN, Pin.PULL_UP)

while(1):
   x = time_pulse_us(0, 1, 100_000)
   y = time_pulse_us(0, 0, 100_000)
   z = x + y
   print(x, y, x+y)
```

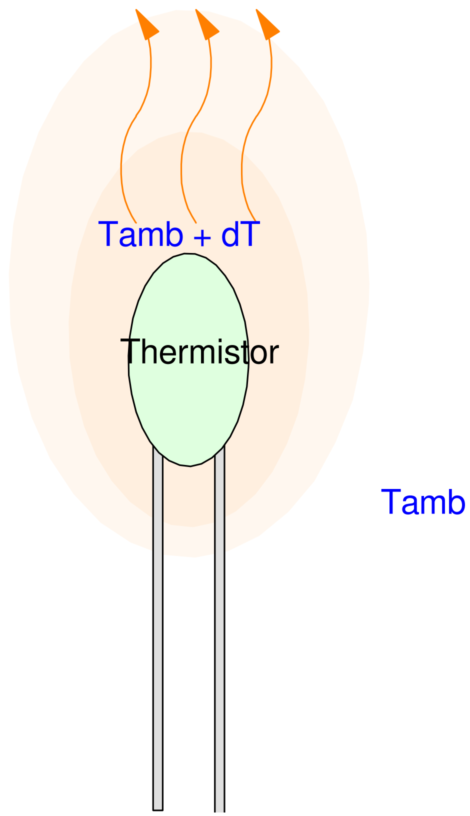
```
10183 10517 20700
10382 10466 20848
10186 10492 20678
:
```

Wind Speed: Hot-Wire Anemometer

With a thermistor, you can measure air speed. The idea is as follows...

Assume you have a thermistor measuring air temperature.

- The measured reading is actually the temperature of the thermistor, not the air
- When you apply current to a thermisor, it warm up due to I^2R heating
- The amount it warms up is related to the dissipation factor: 3.5mW/K in this case



Calculations:

For the previous example, the self heating at 25C will be:

$$I = \frac{3.3V}{R+R_0} = \frac{3.3V}{2000\Omega} = 1.65mA$$

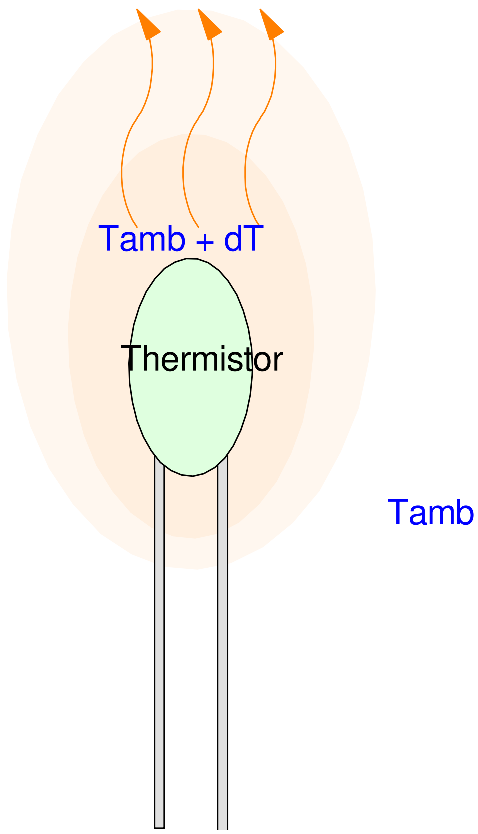
$$P = I^2R = (1.65mA)^2 \cdot 1000\Omega$$

$$P = 2.7225mW$$

The self-heating will then be

$$dT = \frac{2.7225mW}{3.5mW/K} = 0.778K$$

i.e. the reading you get will be high by 0.778 degrees C (or K)



This is when there is no breeze

- The stronger the wind, the smaller dT will be

Hot-Wire Anemometer

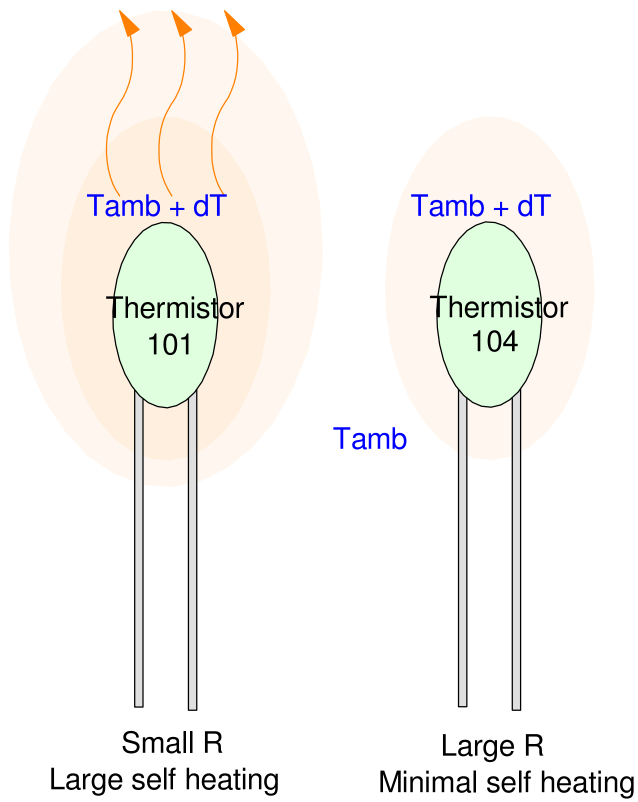
In order to measure the self heating use two thermistors

- One has a low R
  - R(25C) = 100 Ohms
  - Large self heating
- One has a large R
  - R(25C) = 100k Ohms
  - Minimal self heating

Assume each has a B-value of 3930:

$$R_0 = 100 \cdot \exp\left(\frac{3930}{T+273} - \frac{3930}{298}\right)\Omega$$

$$R_1 = 100k \cdot \exp\left(\frac{3930}{T+273} - \frac{3930}{298}\right)\Omega$$



Assume two voltage dividers

- Matching 100 and 100k resistors

With self-heating (no wind)

$$P_0 = \frac{V^2}{R} = \frac{(1.65V)^2}{100} = 27.225mW$$

$$P_1 = \frac{V^2}{R} = \frac{(1.65V)^2}{100k} = 27.225\mu W$$

The temperature difference will be

$$\delta T_0 = \frac{27.225mW}{3.5mW/K} = 7.778K$$

$$\delta T_1 = \frac{27.225\mu W}{3.5mW/K} = 0.00778K$$

The 100 Ohm resistor is 7.77 degrees warmer than the 100k resistor

The temperature difference causes a difference in resistance

$$R_0 = 71.501\Omega \qquad \textit{(100 Ohms nominal)}$$

$$R_1 = 99.965k\Omega \qquad \textit{(100k Ohms nominal)}$$

and a difference in voltage:

$$V_0 = 1.3758V$$

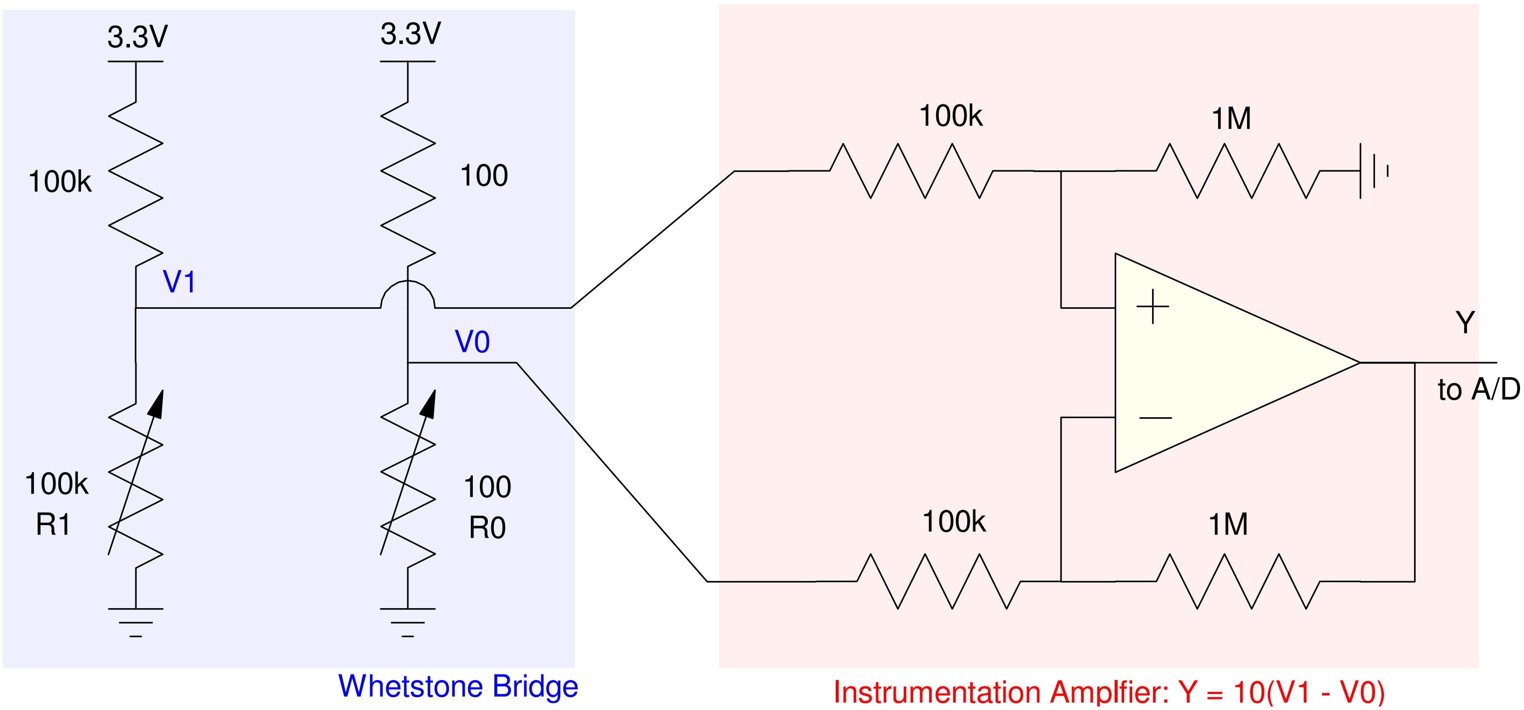
$$V_1 = 1.6497V$$

$$dV = V_1 - V_0 = 0.2739V$$

The voltage difference is a measure of air speed

- As the air speed increases
  - you get more cooling, meaning
  - the temperature (and voltage) difference decreases.

To improve the sensitivity, add a differential amplifier



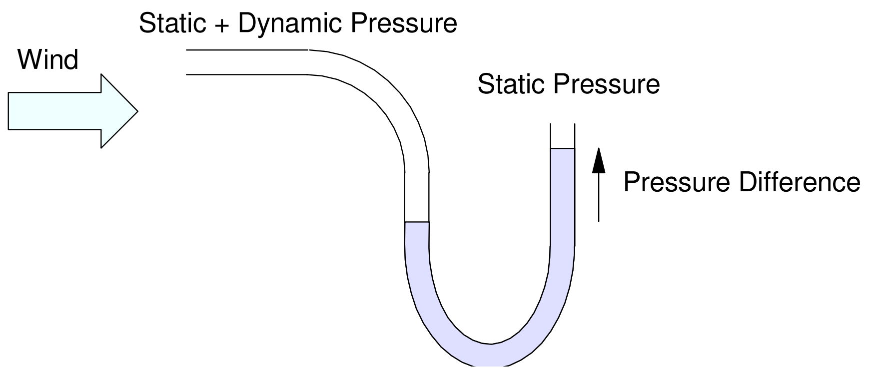
Hot-Wire Anemometer: A Whetstone bridge outputs a voltage difference.
An instrumentation amplifier takes the difference and amplifies it

Wind Speed: Pitot Tube

A third way to measure wind speed is to measure

- The pressure of a tube going into the wind, and
- The pressure of a type perpendicular to the wind

This is called a pitot tube and is a common way for aircraft to measure air speed.



Pitot Tube: The pressure difference is a measure of air speed

From Wikipedia, the total pressure (static plus dynamic) is

$$P_t = P_s + \left(\frac{\rho v^2}{2}\right)$$

where ρ is the fluid density. Solving for velocity in terms of the pressure difference:

$$v = \sqrt{\frac{2(P_t - P_s)}{\rho}}$$

or

$$v = \left(\sqrt{\frac{2}{\rho}}\right)(\Delta P)^{1/2}$$

By measuring the pressure difference, you have a measurement of speed.

Example: Measure 0 - 30m/s

- 0 - 67 mph

The density of air at 20C at 1 atmosphere is 1.204 kg / m2 (Wikipedia)

The pressure difference at 30 m/s is

$$\Delta P = \frac{\rho v^2}{2} = \frac{1}{2}\left(1.204\frac{kg}{m^3}\right)(30\tfrac{m}{s})^2$$

$$\Delta P = 541.8\frac{kg}{m \cdot s^2}$$

Adjusting the units:

$$\Delta P = 541.8\left(\frac{kg \cdot m}{s^2}\right)\left(\frac{1}{m^2}\right)$$

$$\Delta P = 541.8\left(\frac{N}{m^2}\right) = 541.8\ Pa$$

or

$$\Delta P = 5.418\ hPa$$

note: One atmosphere is 1013.25 hPa

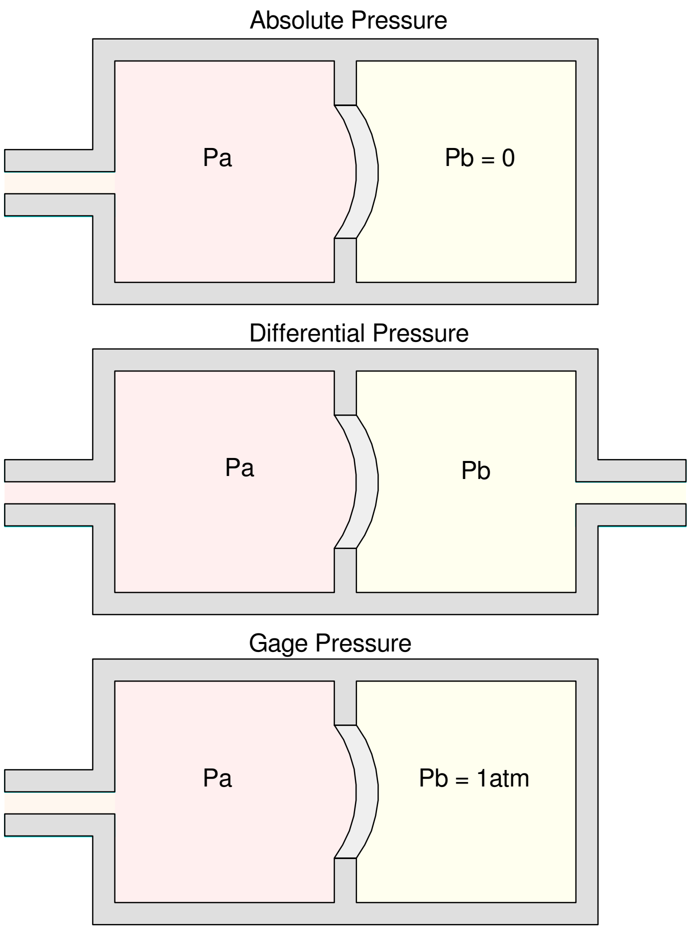
- This method works better at higher speeds (i.e. airccraft)

Digital Pressure Sensor (BME280, BMP280)

https://randomnerdtutorials.com/raspberry-pu-pico-bme280-micropython/

There are several types of pressure sensors

- Absolute pressure
  - Pressure relative to a vaccuum
- Differential pressure
  - Pressure of A vs. B
- Gage pressure
  - Pressure relative to 1 atmosphere



You can also get

- Analog pressure sensors
  - output is a voltage
  - Digital pressure sensors
  - output is read via SCI or I2C

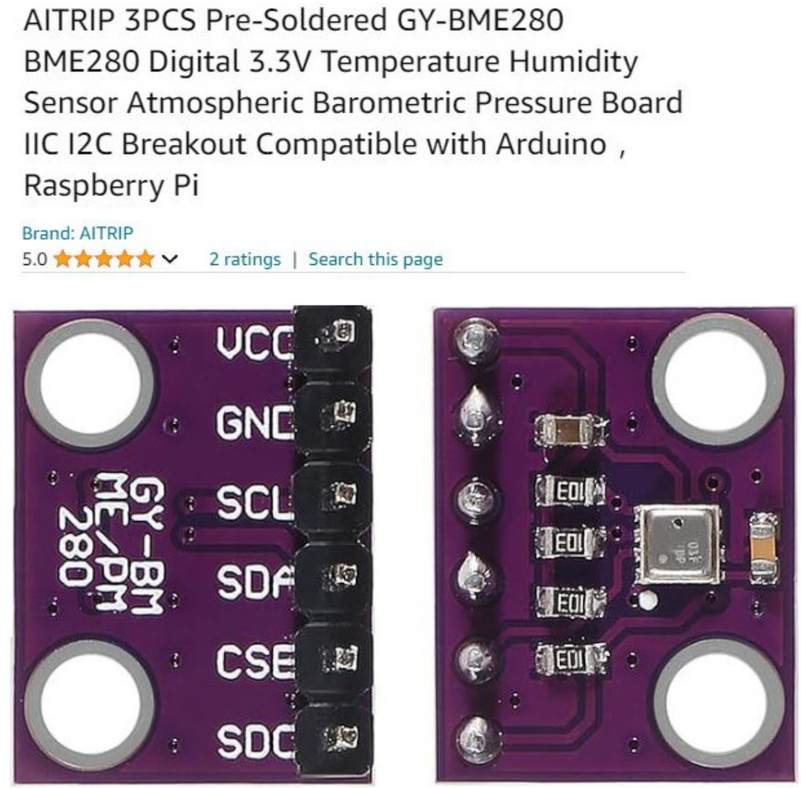
The BME280 or BMP280.

These sensors measure

- Temperature
  - resolution 0.01C
- Absolute Pressure
  - resolution 0.01 hPa
  - 1atm = 1013.25 hPa
- Relative humidity
  - resolution 0.01%
  - BME280 only

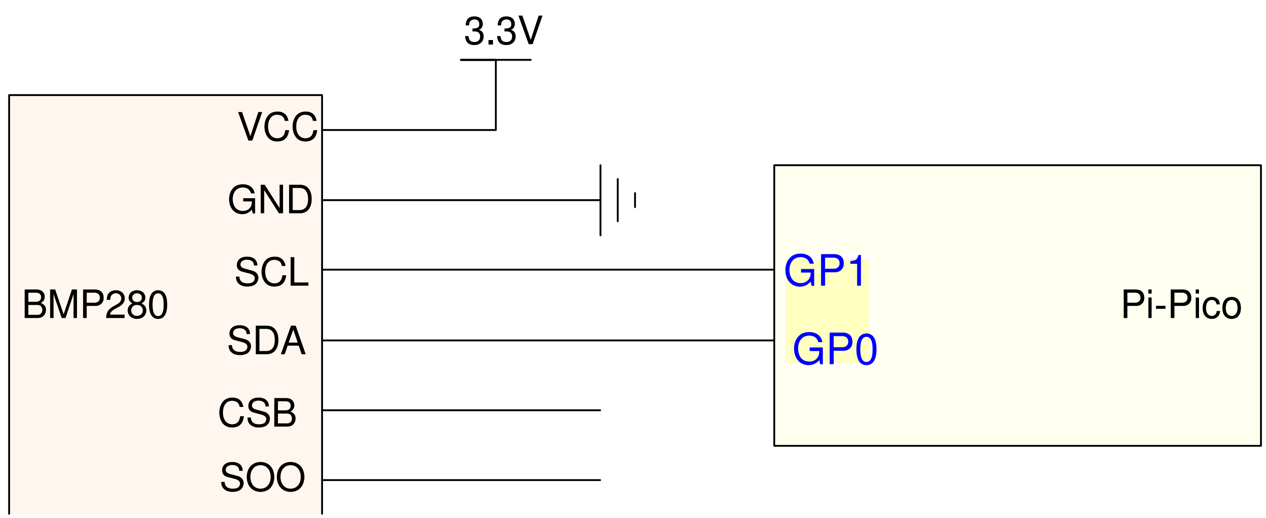
These sensors also support

- I2C communications
- SPI communications



Hardware: I2C Communications

. Digital sensors are pretty easy to interface, so let's go with these.



Software: I2C Communications

- Option 1: Write your own code
- Option 2: Find code someone else has written

Going with option #2 for I2C communications

- *https://randomnerdtutorials.com/raspberry-pu-pico-bme280-micropython/*
- Download the BME280 library
- Download the BME280 driver code

Following the schematics for I2C communications

- And you're reading temperature, humidity, and pressure
- BMP280 has temperature and pressure (humidity reads as 0.00%)
- BME280 has temperature, humidity, and pressure

```
# Raspberry Pi Pico: BME280 Code
# Complete project details at
# https://RandomNerdTutorials.com/raspberry-pi-pico-bme280-micropython/

from machine import Pin, I2C
from time import sleep
import BME280

# Initialize I2C communication
i2c = I2C(id=0, scl=Pin(1), sda=Pin(0), freq=10000)

while True:
    try:
        # Initialize BME280 sensor
        bme = BME280.BME280(i2c=i2c)

        # Read sensor data
        tempC = bme.temperature
        hum = bme.humidity
        pres = bme.pressure

        print('Temperature: ', tempC)
        print('Humidity: ', hum)
        print('Pressure: ', pres)
```

shell

```
Temperature: 21.64C
Humidity:  0.00%
Pressure:  986.70hPa
```

Humidity Sensor

- DHT11 Temperature and Humidity
- DHT22 Temperature and Humidity

These are older sensors

- Digikey lists then as being obsolete
  - Replaced by the BME280 (?)
- But, drivers are included in MicroPython

Specifications

- 3.3V to 6.0V
- Relative Humidity (RH): 0% to 100%
  - 2% accuracy
- Temperature: -40C to +125C
  - 0.2C accuracy
- Resolution
  - 1C and 1% RH (DHT11)
  - 0.1C and 0.1% RH (DHT12)

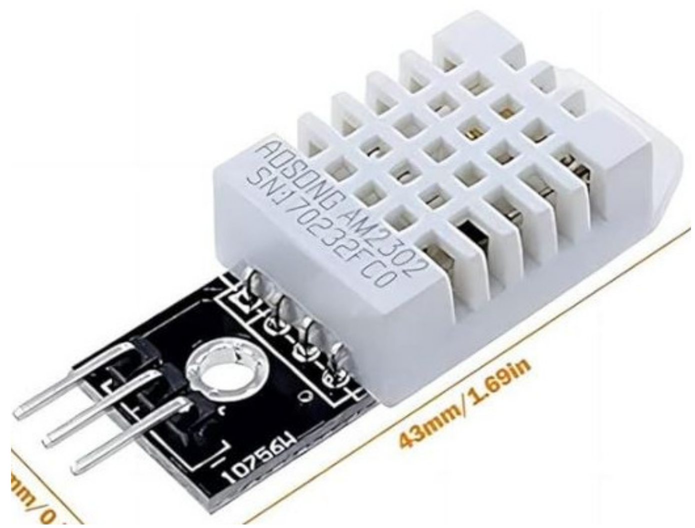
3Pcs DHT22/AM2302 Digital Temperature and Humidity Sensor Module Temperature Measure-40-80℃ (±0.5℃) Humidity Measure 0%-100% RH (±2%) for Arduino DIY

Brand: WWZMDiB

4.2 ★★★★☆ 12 ratings | Search this page

50+ bought in past month

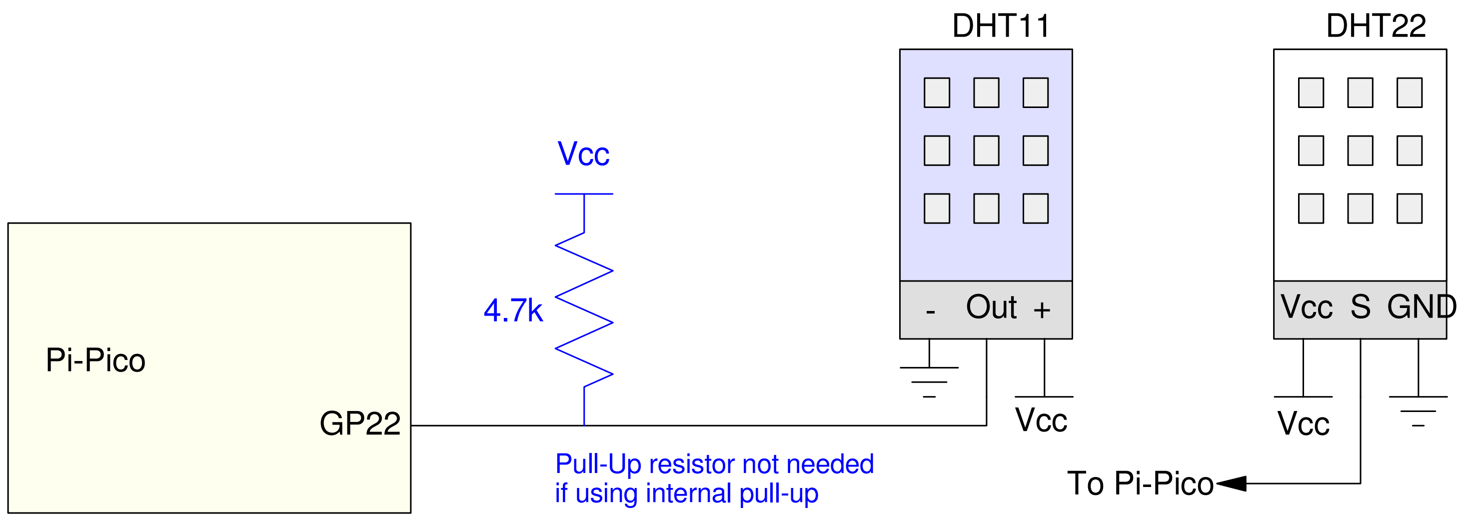
$9.99



DTH11 & DTH22 Wiring

The wiring is similar

- Each needs power and ground
- Each uses a 4.7k pull-up resistor in the signal line
  - The internal pull-up of the Pi-Pico could be used



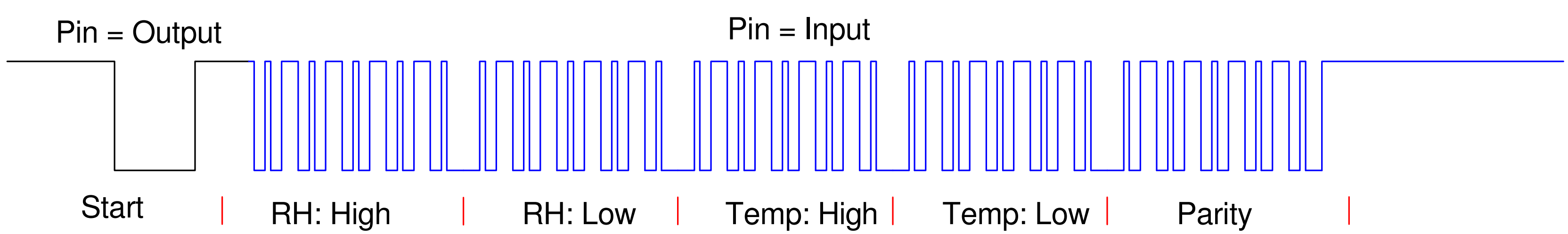
1-Wire Communications

First, the Pi-Pico

- Output a 500us low pulse followed by a 20-40us high pulse.
- This tells the DHT sensor to start a data conversion
- Also tells it to return the previous reading
  - Do a double-read to get current data

Next, set the IO pin to input

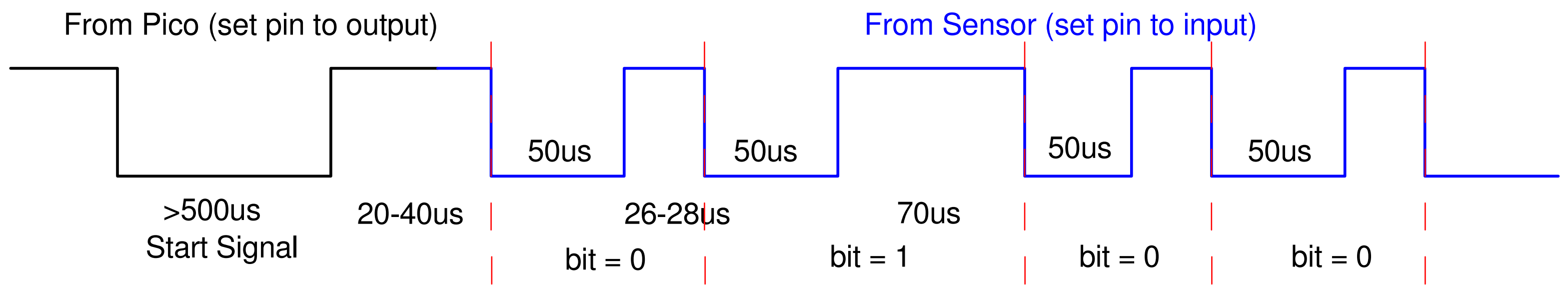
- The DTHXX sensor takes over
- 5 bytes are then returned



Logic 1 and 0:

1's and 0's are encoded by the duration of the high pulse:

- Logic 0: 26-28us high pulse
- Logic 1: 70us high pulse



Software:

The driver routines are part of the standard library set

```
>>> import dht
>>> help(dht)

object <module 'dht' from 'dht.py'> is of type module
  DHTBase -- <class 'DHTBase'>
  __version__ -- 0.1.0
  DHT11 -- <class 'DHT11'>
  DHT22 -- <class 'DHT22'>

>> help(dht.DHT11)

object <class DHT11'> is of type type
  __module__ -- dbt
  humidity -- <function humidity at 0x20013c00>
  temperature -- <function temperature at 0x200137c0>
  __qualname__ -- DHT11
```

To read a DHT11 sensor, the following Python code works. Note that the results are integers.

```
# DHT11 (blue)
import dht, machine, time

d11 = dht.DHT11(machine.Pin(22))

for i in range(0,3):
    d11.measure()
    Temp = d11.temperature()
    RH = d11.humidity()
    print(i, 'RH = ', RH, '%  Temp = ', Temp,'C')
    time.sleep(0.5)
```

shell

```
0 RH = 56 %  Temp = 25 C
1 RH = 58 %  Temp = 21 C
2 RH = 58 %  Temp = 21 C
```

To read a DHT22 sensor, a similar program works (note with the DHT22, the resolution is 0.1)

```
# DHT22 (white)
import dht, machine, time

d22 = dht.DHT22(machine.Pin(22))

for i in range(0,3):
    d22.measure()
    Temp = d11.temperature()
    RH = d22.humidity()
    print(i, 'RH = ', RH, '%  Temp = ', Temp,'C')
    time.sleep(0.5)
```

shell

```
0 RH = 57.7 %  Temp = 21.2 C
1 RH = 61.7 %  Temp = 21.5 C
2 RH = 62.9 %  Temp = 21.6 C
```

Summary

Measuring wind speed, air pressure, or humidity is also fairly easy with a Pi-Pico. Especially if someone else has already written the drivers.

References

Pi-Pico and MicroPython

- https://github.com/geeekpi/pico_breakboard_kit
- https://micropython.org/download/RPI_PICO/
- https://learn.pimoroni.com/article/getting-started-with-pico
- https://www.w3schools.com/python/default.asp
- https://docs.micropython.org/en/latest/pyboard/tutorial/index.html
- https://docs.micropython.org/en/latest/library/index.html
- https://www.fredscave.com/02-about.html

Pi-Pico Breadboard Kit

- https://wiki.52pi.com/index.php?title=EP-0172

Other

- https://docs.sunfounder.com/projects/sensorkit-v2-pi/en/latest/
- https://electrocredible.com/raspberry-pi-pico-external-interrupts-button-micropython/
- https://peppe8o.com/adding-external-modules-to-micropython-with-raspberry-pi-pico/
- https://randomnerdtutorials.com/projects-raspberry-pi-pico/
- https://randomnerdtutorials.com/projects-esp32-esp8266-micropython/