GoPiGo3 DI Sensors

Contents

Dexter Sensors Documentation

DI sensor documentation: https://di-sensors.readthedocs.io/en/master/

Dexter Temperature, Humidity, and Pressure Sensor Tutorial

A tutorial for how to use the Dexter Temperature, Humidity Sensor (BME280).

Barometric pressure compensation for altitude:
https://www.engineeringtoolbox.com/barometers-elevation-compensation-d_1812.html

1. Shutdown the GoPiGo3. (Do not connect sensors when the GoPiGo3 has power.)
2. Plug the BME280 sensor into an I2C port.
3. Mount the sensor on a sensor mount.

bme280_sensor.py

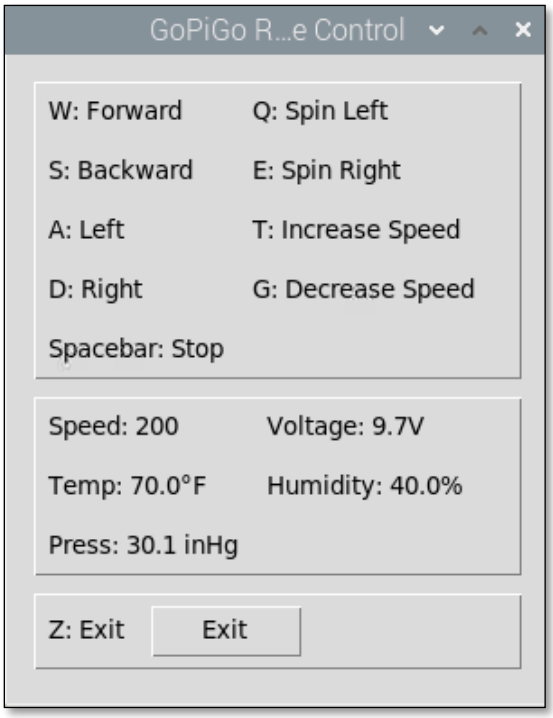
This program will read the Dexter Temperature, Humidity Sensor (BME280) every 5 seconds and display to the console.

```
#!/usr/bin/env python3
# Name: thp_sensor.py
# Purpose: Read temperature, humidity and barometric pressure
# ------------------------------------------------
# History
# ------------------------------------------------
# Author    Date        Comments
# Loring    10/24/21    Changed to fahrenheit, convert pressure to inHg,
#                       compensate for altitude
# Barometric pressure compensation for altitude:
# https://www.engineeringtoolbox.com/barometers-elevation-compensation-d_1812.html
#
# DI sensor documentation: https://di-sensors.readthedocs.io/en/master/
#
# Python example program for the Dexter Industries
# BME280 Temperature Humidity Pressure Sensor

```

```
from time import sleep
from di_sensors.easy_temp_hum_press import EasyTHPSensor

print("Example program for reading Dexter Industries")
print("Temperature Humidity Pressure Sensor on an I2C port.")

my_thp = EasyTHPSensor()

while True:
    # Read the temperature
    # temp = my_thp.safe_celsius()
    temp = my_thp.safe_fahrenheit()

    # Read the relative humidity
    hum = my_thp.safe_humidity()

    # Read the pressure in pascals
    press = my_thp.safe_pressure()

    # Convert pascals to inHg, compensate for 4000' altitude
    press = (press / 3386.38867) + 4.08

    # Print the values to the console
    print("Temperature: {:5.1f}F  Humidity: {:5.1f}%  Pressure: {:5.2f}".format(
        temp, hum, press))

    # Pause between readings
    sleep(5)
```

The next step would be to send the data every 15 seconds or more to ThingSpeak. All sensors can be setup to upload data to ThingSpeak.

The data can also be displayed in a GUI program. This is an example of a Tkinter remote control program that also displays real time data from a bme280 sensor.



Dexter Grove Buzzer

AD1 or AD2.

buzzer.py

Dexter Light and Color Sensor

I2C port

sensor_light_color.py

Dexter Inertial Measurement Unit (IMU)

I2C port.

sensor_imu.py

Dexter IR Sensor and Remote

AD1 port.

sensor_ir.py