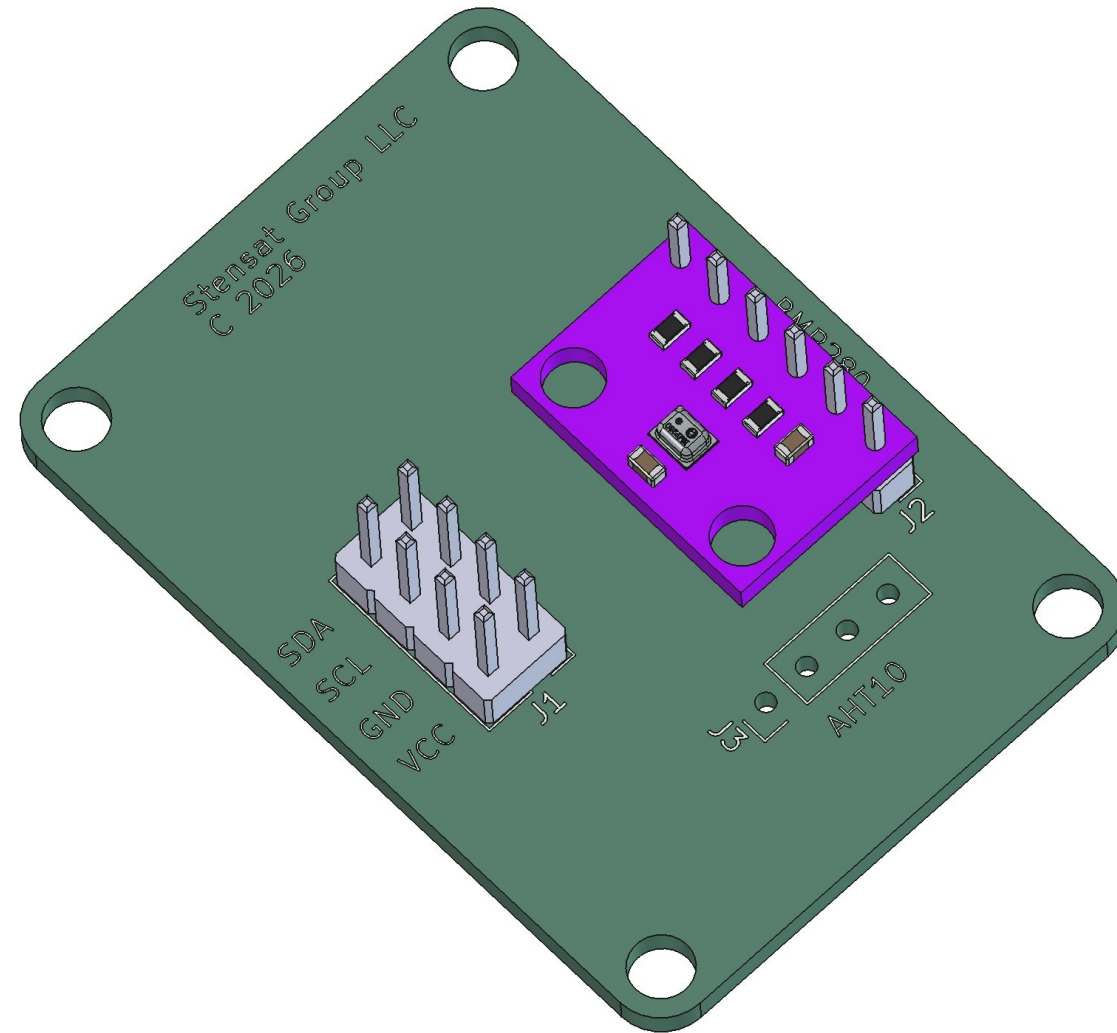
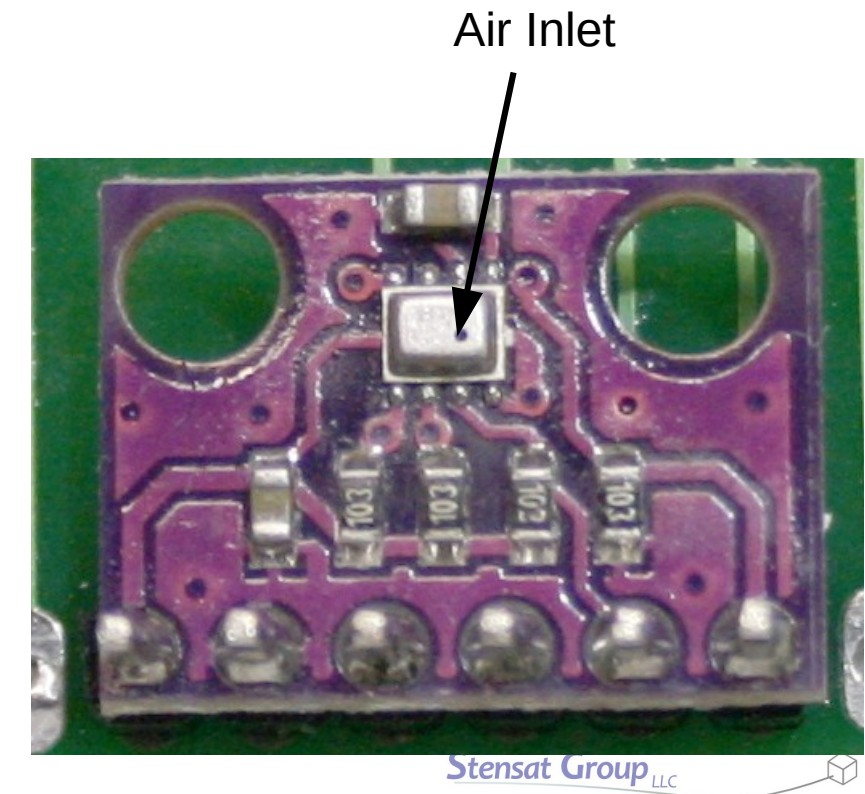


Air Pressure Sensor



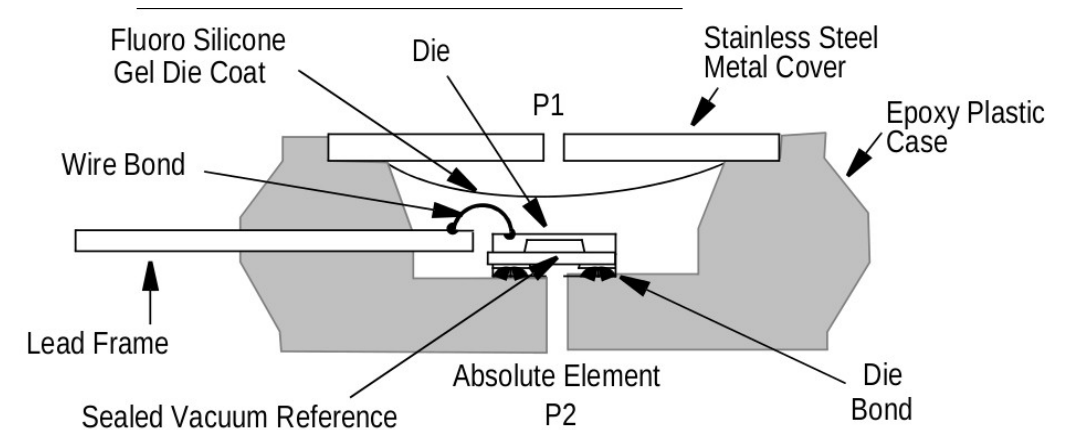
Pressure Sensor

- This is the pressure sensor board. The green board is the sensor carrier that is designed to support different sensors.
- The violet circuit board is the pressure sensor circuit.
- The small silver rectangle is the sensor itself. Notice the small hole. This hole allows air to enter so it can measure the air pressure.

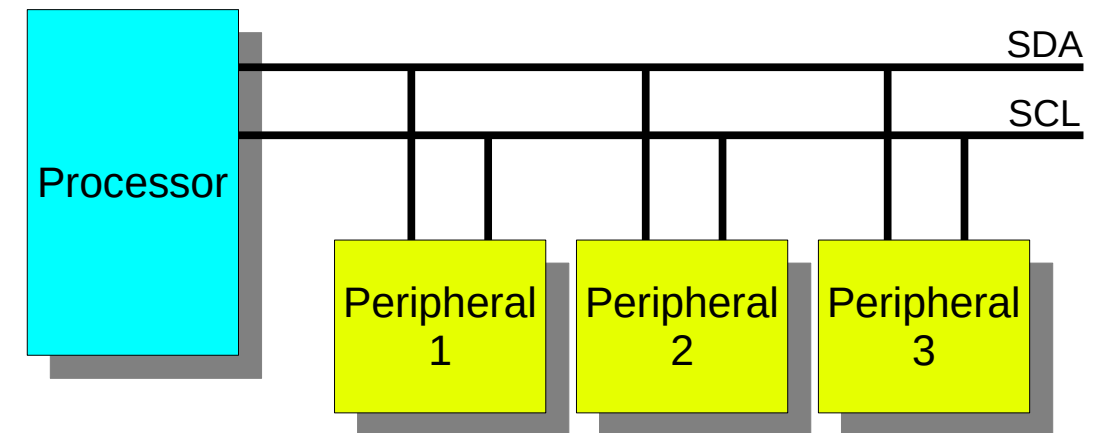


Pressure Sensor

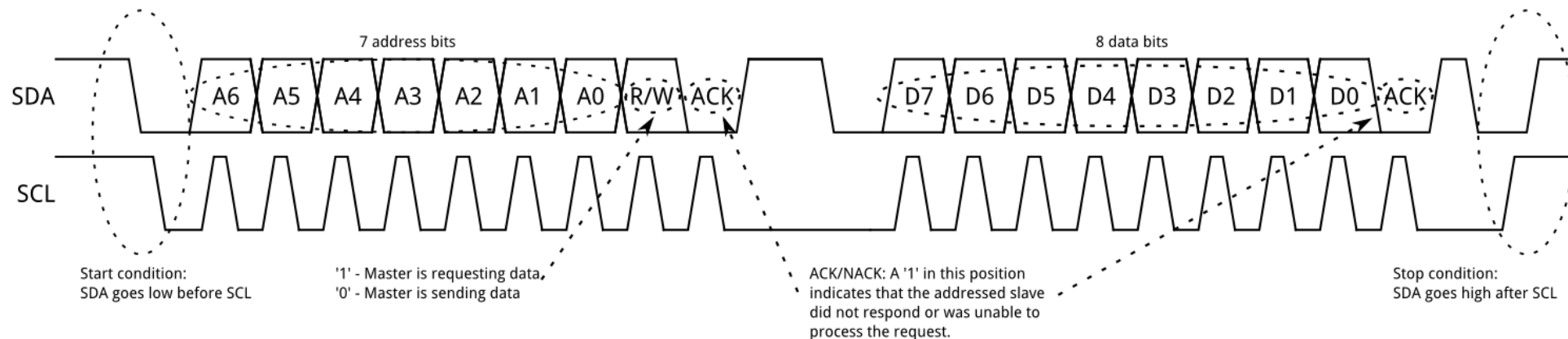
- The BMP280 is a piezo-resistive pressure sensor. The piezo device changes resistance when it bends.
- The piezo device is mounted over a cavity that has no air. The atmosphere presses against the piezo device bending it. As the air pressure drops, the piezo device bends less because there is less air pressing against it.



- I2C stands for Inter-Integrated Circuit. It is a serial type interface requiring only two signals, a clock signal and a data signal.
- The I2C bus is typically used to interface with sensors and peripheral devices not needing to communicate at high speeds. The standard data rate is 100 Kilobits per second.
- Multiple devices can be connected to a single I2C bus. The processor is the controller and all the connected devices are peripherals. The processor is also called the master and the peripherals are slaves. Each slave has a unique address.



- The clock signal is labeled SCL. This signal is used to control the flow of the data bits.
- The data signal is called SDA. This carries the data serially.
- The diagram below shows how a data transfer occurs.
- The data transfer protocol is for the master to first send out a device address. This is a 7 bit number followed by a bit indicating if the next byte is to be written to a slave or read from a slave.
- The SCL signal toggles for each bit sent.



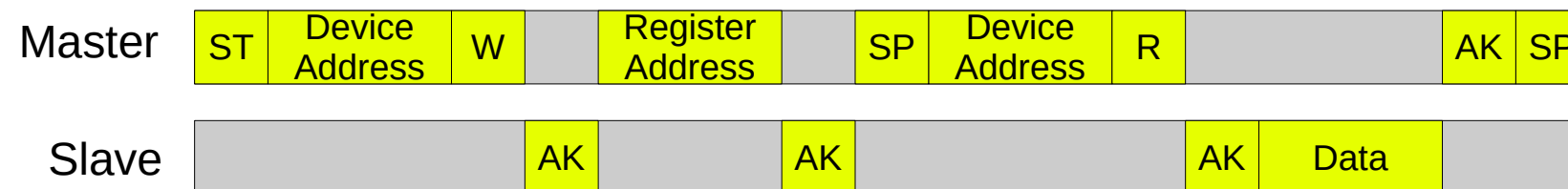
I2C Sequence

- Every device on the I2C bus has a unique 7-bit address. The pressure sensor address is 0x76.
- The I2C operation for writing to a register is:
 1. Send Start sequence by keeping SCL high and changing SDA from high to low (ST)
 2. Send the device address
 3. Send the register address
 4. Send the stop sequence by changing SDA from low to high while SCL is high first. (SP)
- All Bytes sent are acknowledged by the slave.



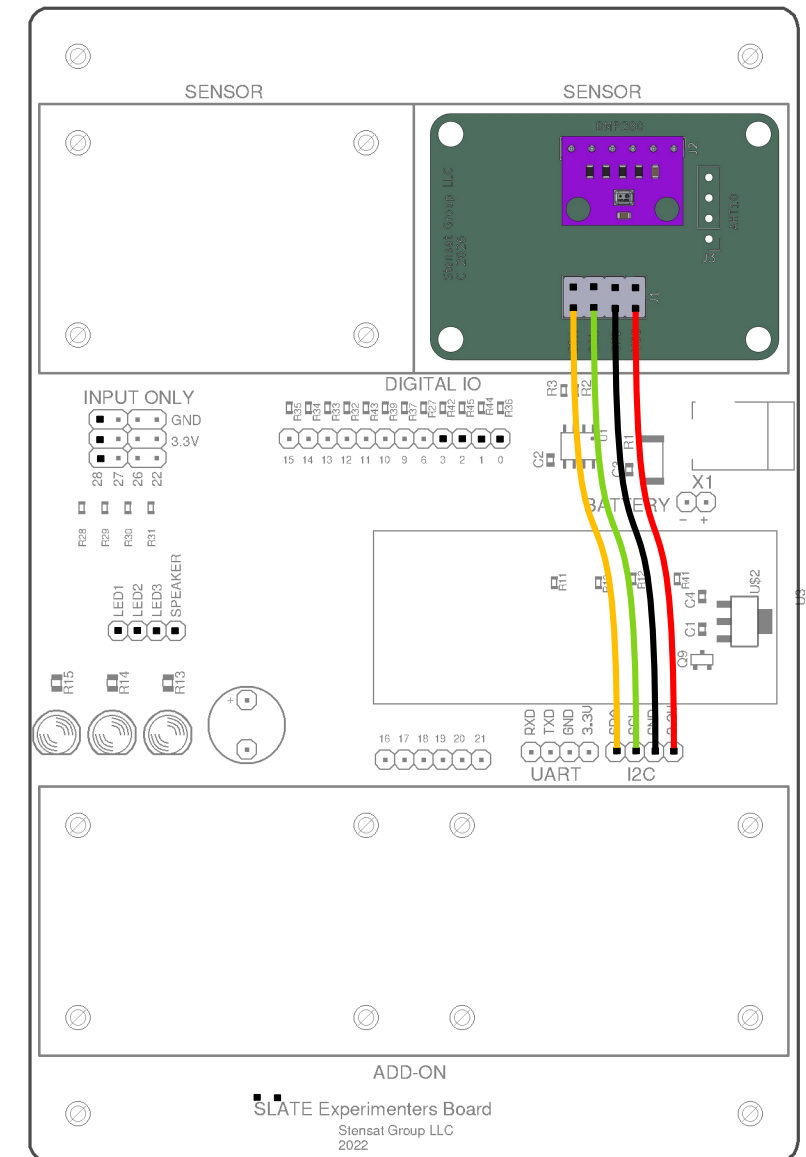
I2C Sequence

- The I2C operation for reading a register is:
 1. Send Start sequence by keeping SCL high and changing SDA from high to low (ST)
 2. Send the device address
 3. Send the register address
 4. Send a repeat start sequence with a read indication (SR)
 5. Read in 6 bytes
 6. Send a stop sequence (SP)



Connecting the Pressure Sensor

- The processor uses pins 4 and 5 for the I2C bus. Pin 4 is SCL and pin 5 is SDA. They are labeled SCL and SDA.
- Connect a jumper wire from 3.3V on the processor to +V on the sensor board.
- Connect the GND to GND.
- Connect SDA to SDA.
- Connect SCL to SCL.



- From Canvas, download the **bmp280.py** module.
- Using the File View in Thonny, locate the **bmp280.py** module.
- Make sure the SLATE board is plugged in and connected to Thonny.
- Find the file **bmp280.py** and right click on it and select **Upload**.
- The file should now appear in the SLATE file list.

Pressure Sensor Test Program

- Enter the program to the right. Make sure to use capital letters since it does matter.
- The first line imports the **I2C** bus interface functions and **Pin** functions.
- The second line imports the pressure sensor module functions.
- The third line imports the **time** functions.
- Next, the I2C bus is configured by defining the **SDA** and **SCL** pins and assigned to the object **bus**.
- Next, the pressure sensor is initialized and assigned to object **bmp**.
- In the while loop, the pressure and temperature are captured.
- The pressure function calls look like variables. This is an object oriented feature of python.
- Run the program and see the results.

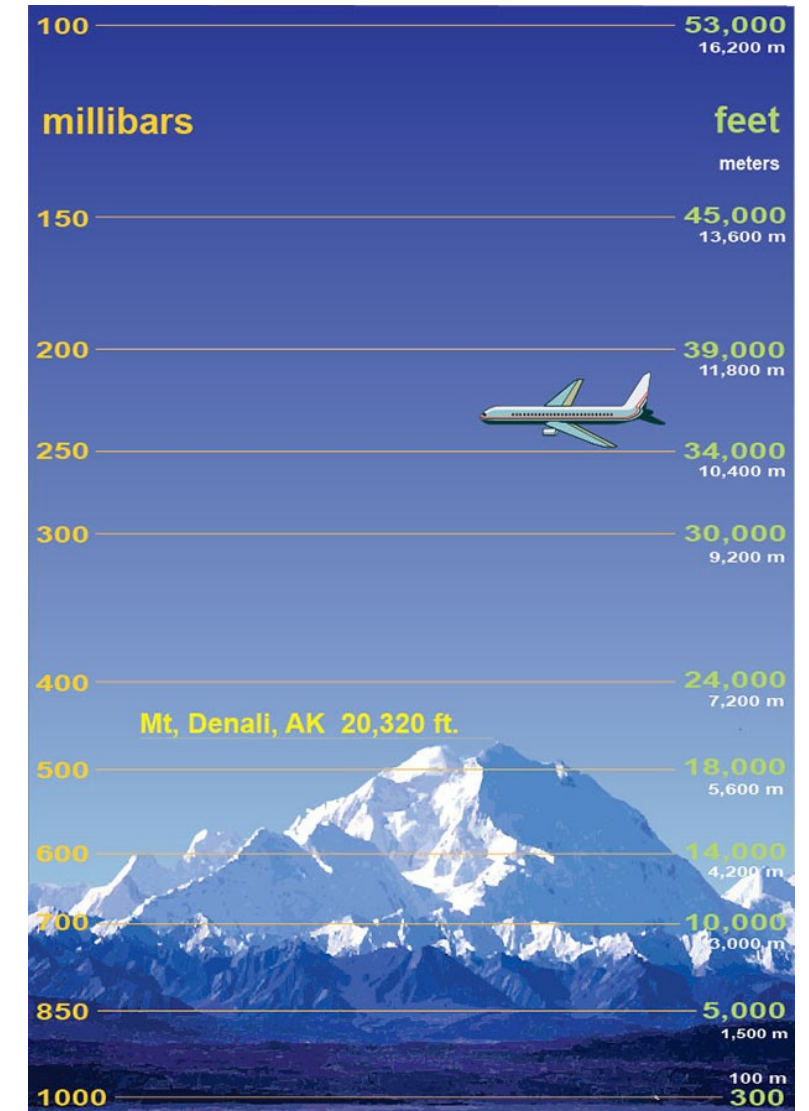
```
from machine import I2C,Pin
from bmp280 import BMP280
import time

bus = I2C(sda=Pin(4),scl=Pin(5))
bmp = BMP280(bus)

while 1:
    print(bmp.pressure,bmp.temperature)
    time.sleep(.1)
```

Pressure Sensor Test Program

- Air pressure is the weight of the atmosphere pressing down on the earth. The higher the altitude, the less air pressing down at that altitude. This allows the air pressure to be used for measuring altitude.
- At sea level, the air pressure is 101,000 Pascals with a temperature of 15 degrees C and 0% humidity. Humidity and temperature does affect the air pressure.
- Air pressure does change over time due to weather changes.



Altitude Calculation

- Altitude can be derived from air pressure. The function used assumes a temperature of 15C and references the air pressure at sea level.
- The highlighted code is the function to calculate altitude. The math module had to be added in order to do a power calculation.
- Add the print statement shown highlighted. This shows how to use the function to get the altitude.
- Run the program. The results should be the altitude referenced to sea level.
- Compare the result with your location using **Google Earth**. Enter your location and then click on your position and see what Google Earth indicates as a height. It can be off by 20 or so meters due to the weather.
- The weather can change the air pressure which will change the calculated height.

```
from machine import I2C,Pin
from bmp280 import BMP280
import time
import math

def calcAltitude(press):
    a = press / 101325.0
    b = 1/5.25588
    c = math.pow(a,b)
    c = 1.0 - c
    c = c / 0.0000225577
    return c

bus = I2C(sda=Pin(4),scl=Pin(5))
bmp = BMP280(bus)

while 1:
    print(bmp.pressure,bmp.temperature)
    print(calcAltitude(bmp.pressure))
    time.sleep(.1)
```

Pressure Sensor Test Program

- To properly measure altitude for a rocket flight, the ground altitude needs to be measured before launch. Call it the base altitude.
- As the rocket flies, the base altitude is subtracted from the measured altitude during flight. Over the relatively short flight of the rocket, the weather will not change enough to add errors to the measurements.
- The code to the right is a modification of the previous program. Variable **base** has been added to capture the base altitude.
- Variable **alt** has been added to get the current altitude relative to the ground by having base subtracted from the current calculated altitude.
- The print statement displays the altitude and the pressure.

```
from machine import I2C,Pin
from bmp280 import BMP280
import time
import math

def calcAltitude(press):
    a = press / 101325.0
    b = 1/5.25588
    c = math.pow(a,b)
    c = 1.0 - c
    c = c / 0.0000225577
    return c

bus = I2C(sda=Pin(4),scl=Pin(5))
bmp = BMP280(bus)

base = calcAltitude(bmp.pressure)

while 1:
    alt = calcAltitude(bmp.pressure) - base
    print(alt,bmp.pressure)
    time.sleep(.1)
```

