Lab 5

Explore the Adafruit Feather Bluefruit Sense

Part 1 - Running the Feather Sense Demo program

Plug your Feather into the computer and go to file explorer. We need to check to make sure the necessary CP libraries are in the \lib folder on the Feather.

List the folders and file names found in your \lib folder on the Feather Sense:

- **adafruit_apds9960**
- **adafruit_bmp280.mpy**
- **adafruit_bus_device**
- **adafruit_lis3mdl.mpy**
- **adafruit_lsm6ds**
- **adafruit_register**
- **adafruit_sht31d.mpy**
- **neopixel.mpy**

Highlight the library we already imported and used in Lab 4.

Open the Mu editor and create a new file in CircuitPython mode. Go here: https://learn.adafruit.com/adafruit-feather-sense/circuitpython-sense-demo . Copy and paste the “Feather Sense Sensor Demo” file into your Mu editor. Save the file to the computer as “FeatherSenseDemo.py”. Then save the file to your Feather board as “code.py”.

The file should immediately start running and displaying data from the variety of sensors found on the Feather Sense board. Change the argument for the sleep method from 0.3 to 5.0 and save the code again. ***Show your code working on the Feather to your instructor.*** ☐

List the data values being printed and study the code to figure out how CP retrieves the data values from each of the physical sensors on the Feather Sense. Add the retrieval method to the list below. The first output has been completed as an example for you.

1. Proximity data	apds9960.proximity
2. Color data	apds9960.color_data
3. Temperature data	bmp280.temperature
4. Barometric Pressure data	bmp280.pressure
5. Altitude data	bmp280.altitude
6. Magnetic field data	lis3mdl.magnetic
7. Acceleration data	lsm6ds33.acceleration
8. Gyroscope data	lsm6ds33.gyro
9. Humidity data	sht31d.relative_humidity
10. Sound data	microphone.record (gets samples and then use normalize_rms function to calculate a sound level)

Explore the sensor outputs. Move the Feather Sense and see how some of the values change. As you move the Feather Sense around the magnetic field values change, the acceleration values change and the Gyro values change. What happens when you put your hand over the feather? Proximity value changes, light values change. What happens when you talk to the feather? Sound level changes, humidity level changes, temperature changes. What happens when you blow on the feather? Humidity level changes, temperature changes. Change the value of bmp280.sea_level_pressure (approximately line 36) to the current barometric pressure reading from your sensor. What changes? Setting the bmp280.sea_level_pressure to the current bmp280.pressure effectively zeroes the altitude reading.

Spend some time studying the Feather Sense Demo file. Add header information at the top and add many inline comments before every line in the file. Make sure to study the creation of the many objects that CP is making. These abstractions are key to allowing us to easily access sensor data with CP with methods provided by the many imported modules. Save the fully documented file as lastname_lab5_part1.py.

Possible solution in file lab5_part1.py

Part 2 - Data logging Feather Sense data to hard disk

While instantaneous data can be useful, often a user will want to watch data for trends or study data points that are separated by longer periods of time, i.e. data recorded every hour, every 12 hours, or even once a day.

For this part of the lab, we are going to work on two separate python files. One file will be a CP file running on the Feather Sense, sending data points. The other file will be a Python 3 file which will listen to the serial port (USB) connection and will store data it receives into a data file on the hard disk.

CP File

Modify your file from part 1 to print temperature (in degrees F), barometric pressure and relative humidity data on the same line with a 10 second delay between each data transmission. Print the data with two decimal place precision.

E.g.
78.65 998.04 46.94
78.57 997.98 45.13

You should strip everything out of your working file that you do not need and then document fully. Save the file to the hard drive as “lastname_lab5_part2_cp.py”. ***Show your code working on the Feather to your instructor.***

Possible solution in file lab5_part2_cp.py

Python 3 file

We are going to import a new module, the `serial` module, which will allow us to grab data from the serial port/USB port of the computer. Switch Mu to Python 3 mode and create a new file “lastname_lab5_part2.py”

```
#serial data logger
import serial

# mac  - command to find usb port --->ls /dev/tty* | grep usb
# ser = serial.Serial('/dev/tty.usbmodem11401') #create mac serial object

# windows  - use device manager to help find the right COM port
ser = serial.Serial('COM4')              #create windows serial object

#clear the serial port
ser.flushInput()

fh = open("data_log.txt","w")
fh.close()


try:
    while True:
        ser_bytes = ser.readline()
        decoded_bytes = ser_bytes[0:len(ser_bytes)-2].decode("utf-8")
        print("Hit CTRL-C to exit data logging...")
        print(decoded_bytes)
        fh = open("data_log.txt","a")
        fh.write(decoded_bytes+"\n")
        fh.close()
except KeyboardInterrupt:
    print("keyboard interrupt")
    fh.close()
```

With your Feather Sense plugged into a USB port, run serial_data_logger.py. You should see the data being sent from the feather. Is data being saved? Check for the data_log.txt file. Does the file contain the correct values? ***Show your working data logger to your instructor.***

You are now successfully storing sensor data from the Feather Sense! Add inline comments to the lastname_lab5_part2.py file describing what each line is doing.

Possible solution in file lab5_part2.py

Part 3 - Data logging Feather Sense data to hard disk with time stamp

In Part 2 you stored Feather Sense data to the hard drive. To make the data even more useful, in Part 3 you will add a timestamp to every data point saved.

We can once again use the time module to help. Study and test the following code in the python REPL:

```
import time

time_tuple = time.localtime() # get struct_time
time_string = time.strftime("%m/%d/%Y - %H:%M:%S", time_tuple)

print(time_string)
```

Save a copy of your “lastname_lab5_part2.py” as “lastname_lab5_part3.py” and using the example code given, add a timestamp to every data set you save to the data file “timed_data_log.txt”.

E.g. saved data
04/11/2021 - 14:35:28 79.69 41.67 997.15
04/11/2021 - 14:35:38 79.69 41.69 997.13

You could format the saving of data and the time stamp any way you want really. This is just one example.

Show your working time stamped data logger code to your instructor.

Possible solution in file lab5_part3.py

Part 4 - Data logging some new Feather Sense data to hard disk with time stamp

In Part 4 you will modify the code running on your Feather Sense to send some different data to be stored to a file. You should not have to modify the lastname_lab5_part3.py code from part 3.

As we saw in part 1, the Feather Sense has lots of sensors available. Go back to the Feather Sense Demo File and pick 4 different sensor values and send them to the serial stream. Save the CP file with a new file name “lastname_lab5_part4_cp.py” and make sure to update all the file’s inline documentation. Save the file to your Feather Sense as code.py to make sure it works properly.

Run your Python code from part 3 and confirm the new sensor values are being properly recorded in the data file.

Show your running lastname_lab5_part4_cp.py code to your instructor.

Students will be choosing their own sensor values to send so many possible solutions here…