SmartVibes

Progress Report Deux

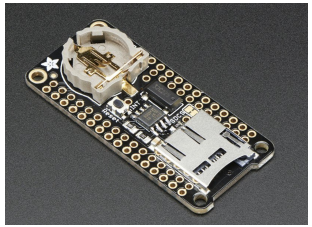
What we set out to create...

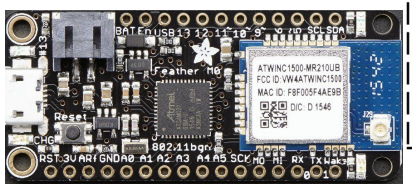
Smart Structures asked us to design a robust and portable vibration monitoring solution.

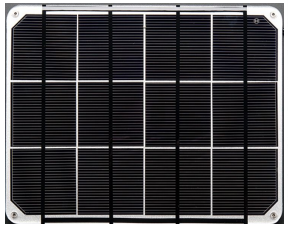
The solution would be used to monitor ground vibrations at a construction site.

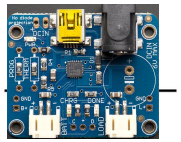
We decided it needed to meet the following requirements:

- Use an accelerometer and geophone
- Monitor minimum frequency of 8hz
- Store data locally
- Provide immediate emergency triggers
- Provide real time monitoring
- Have WiFi connectivity
- Cloud database, remote access
- Four-week power time
- Weather resistant









LOCAL DATA STORAGE

Coin Cell Battery

3V Input

SD Card

SPI

SD Card Breakout

I2C

MICROCONTROLLER & WIFI

802.11 b/g/n

Built-in Wi-Fi Chip & uFL Connector for Antenna

SPI

Adafruit Feather Microcontroller

- 8 PWM Pins
- 10 Analog Inputs
- 1 Analog Output
- Hardware Serial, I2C, and SPI Support

Cloud Server

Alarm/Data Access over Internet

Personnel Device

CLOUD DATA STORAGE

ALARM SYSTEM

Analog Signal

Speaker

3.7V Input

Analog Signal

I2C

Lithium-ion Battery

JST 2-pin Cable

Battery Charger

POWER

Solar Panel

DC Jack Cable

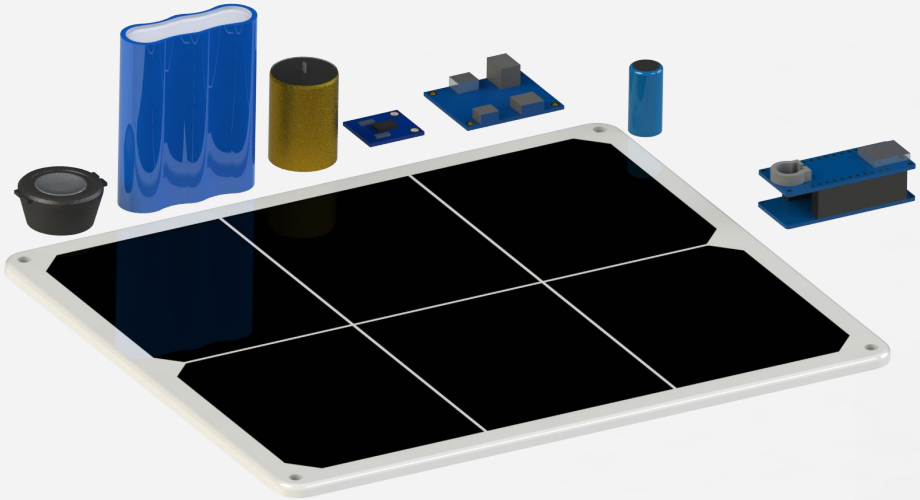
SENSORS

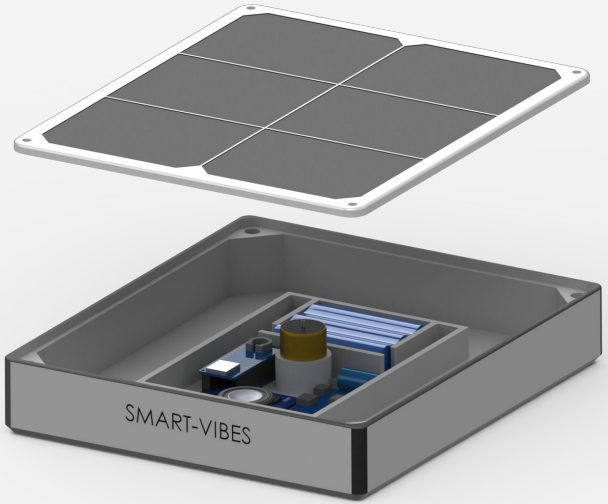
Geophone

Accelerometer

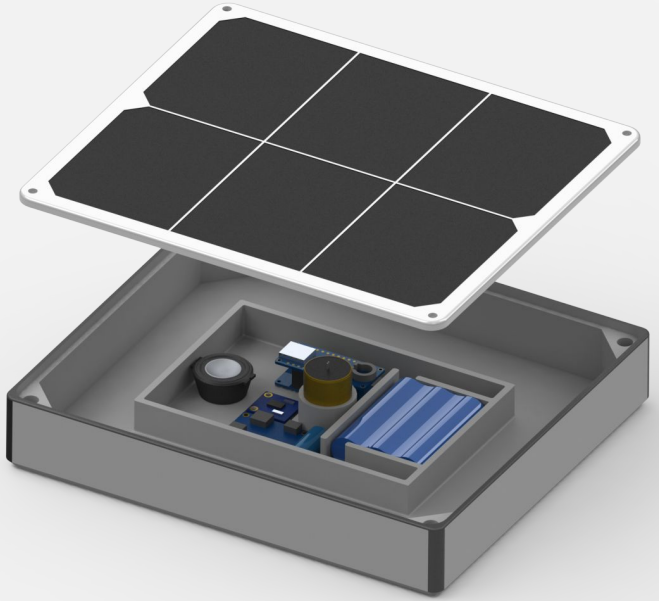
Functional Diagram

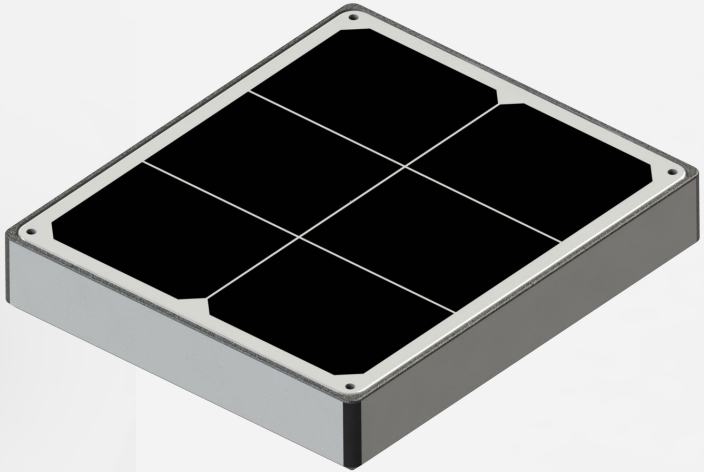
State of Enclosure





State of Enclosure

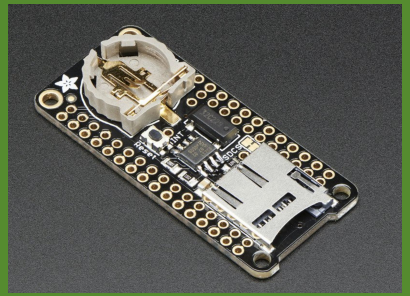




State of Sensors

Progress

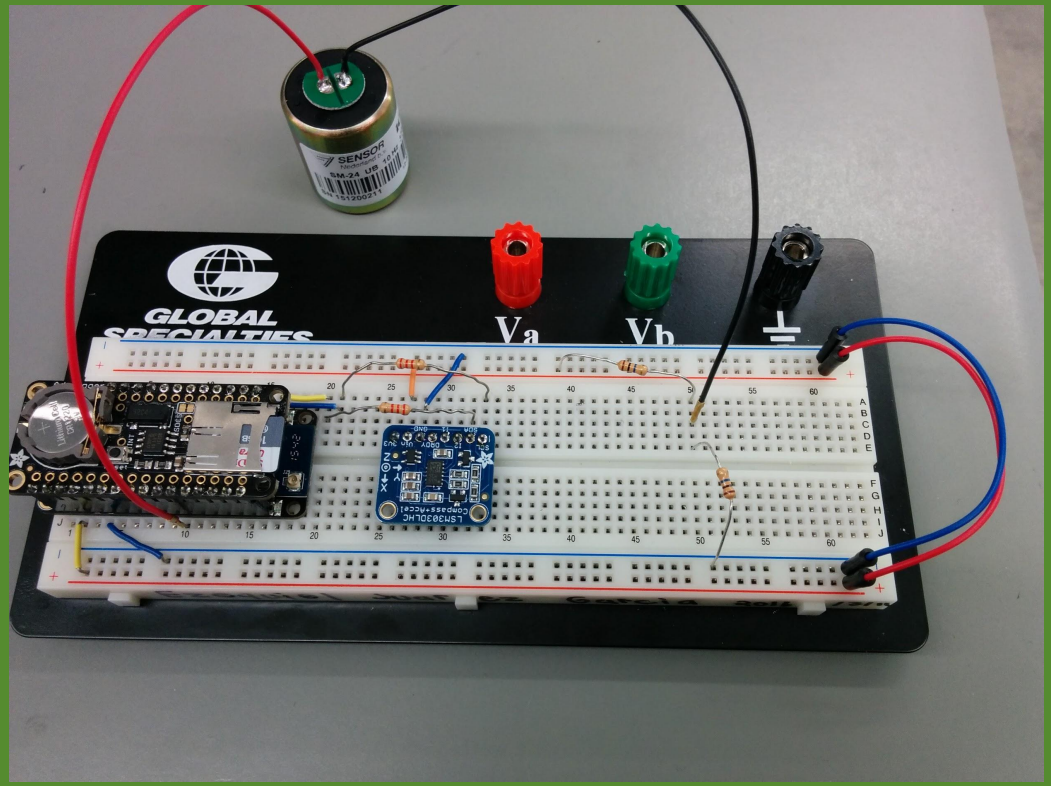
- Accelerometer data is being gathered over I2C.
- RTC clock has been set. Data can now be timestamped correctly.
- Analog data from geophone is being collected.



Problems

- Interrupts are have been difficult to set up.
- Lack of accessible FFT library for the Feather makes it difficult to calculate the frequency of vibrations.

State of Sensors

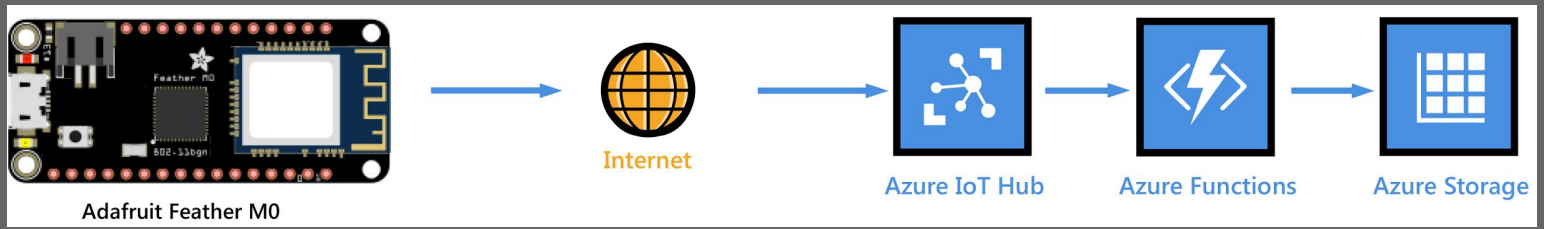


State of Data-linking

- Now using Microsoft Azure IoT Hub platform
- This allows us to send messages to and from the Feather via wireless module
- It uses gulp.js, a toolkit that allows us to automate tasks

Next Steps

- Create Functions that make these messages meaningful
- Data Storage/ Management
- Data interrupts and signal issues
- Alarm system



State of Data-linking

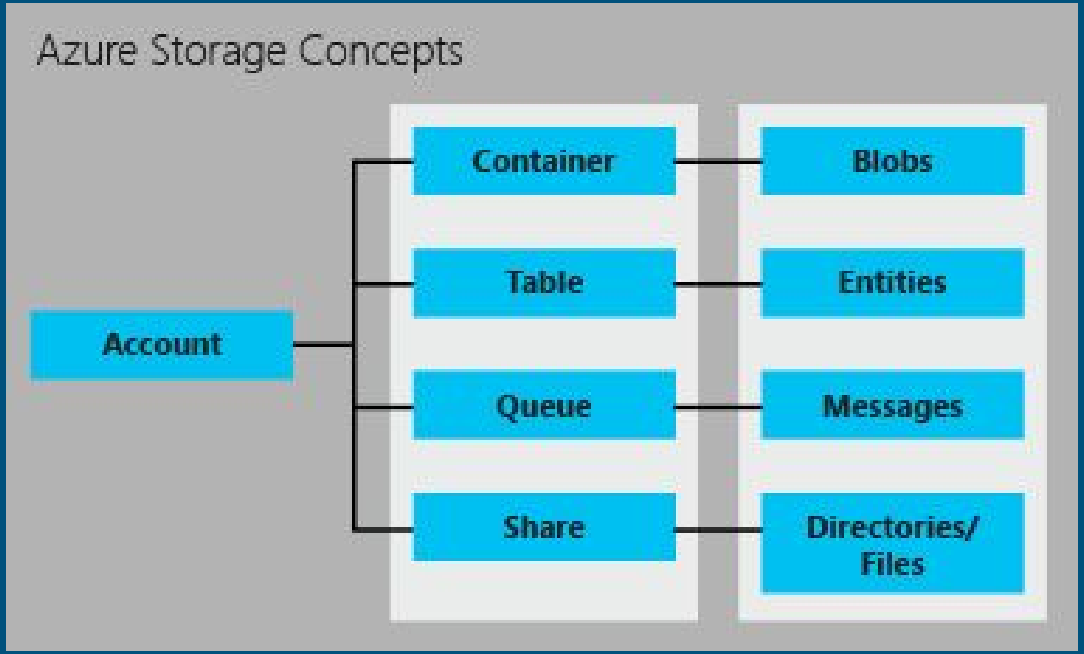
Using gulp file to send messages reporting to the serial monitor and sending them to the database

```
vsc-imac:Lesson3 mac$ gulp run --listen
[12:03:40] Using gulpfile ~/iot-hub-c-feather-m0-getting-started/Lesson3/gulpfile.js
[12:03:40] Starting 'deploy'...
[12:03:40] Starting 'deploy-internal'...
[12:03:40] Starting 'query-iot-hub-messages'...
[12:03:40] Finished 'query-iot-hub-messages' after 20 ms
[12:03:48] Finished 'deploy-internal' after 8.02 s
[12:03:48] Starting 'listen'...
[12:03:48] Finished 'listen' after 11 ms
[12:03:48] Finished 'deploy' after 8.04 s
[12:03:48] Starting 'run'...
[12:03:48] Finished 'run' after 1.83 µs
Serial port /dev/cu.usbmodem1441 opened for monitoring
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
^[[D[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
[Serial Monitor - /dev/cu.usbmodem1441] blinking...
```

State of Database

- We will be using Microsoft Azure IoT's built in storage known as Microsoft Azure Storage.
- The specific database we will use is Table Storage.
- Table storage is a NoSQL key-attribute data store, which allows for rapid development and fast access to large quantities of data.

State of Database cont'd



What's Next?

- Collect both geophone and accelerometer data and store on SD card.
- Program alarm.
- Look into the FFT problem.
- Collectively look into the battery and solar power aspect.
- 3-D print the housing.

Division of Work

Josh - Enclosure

Zeq- Sensor array

Jordan- Data linking

Marc- Database