From: Ide Bradley

Subject :Status Report :

Date: 08/09/15

Brief Project Overview

This is a sensor project, uploading data from a magnetometer sensor and pre-processing it on a PC. The vector data will then be transformed into direction and field strength which is normally how magnetic fields are compared. The results are uploaded to a SQL Azure cloud database. This data will then be plotted and visualised in 3-D. The intent is to create an app which highlights when the earth's magnetic field is starting to respond to a solar storm and provide an early warning for an potential aurora sighting. The system is being set up for multiple sensors of different types and multiple users. However, it is initially planned to use one sensor type and one set of baseline data.

Current Status

The initial work is being done using a much simpler chip BMP180 Barometric Pressure/Temperature/Altitude Sensor- 5V ready . I intend to develop the system on this using pressure and temperature as sample data. It's much easier to debug the data flow with a simpler sensor where the data needs no interpretation and later add in the more complicated data from the magnetometer sensor.

Phase 1 status : Initial temp/pressure sensor connected with a wire to a PC

Project Plan for weather stations	status	comment
set up board with temp and pressure chip	DONE	
Download Arduino sketch and write program to output data	DONE	
Create C# program that asks Arduino for data at timed intervals	DONE	
C# program that builds files from Arduino data that witll work with SQL Azure	DONE	
C# program that reads directory files into datatable and uploads datatable to SQl Azure	DONE	
C# create a regularly scheduled job to load to database	DONE	
SQL Azure database accessible with data	DONE	accessed with SSIS
Reliability test	DONE	Fail. PC reboots too often. (std windows updates) Data set patchy. Looking for 99% uptime.

Completed successfully , but discovered a major design flaw, using the PC to control the dataflow from the Arduino/sensor combination leads to poor performance due to PC interrupts for multiple reasons(windows updates, internet outages, power etc.) If the PC is down the data doesn't get collected and the Arduino does not have enough on -board memory to store more than one or two data points. Trending becomes an issue – since physical intervention is required to restart. A simpler solution is required.

Phase2 status: Add WiFI module to allow the board to communicate directly

wireless temp logging	*research alternatives to wired connections*	*DONE*	*Add ESP266 module to board. http://www.adafruit.com/datasheets/ESP8266_Specifications_English.pdf*
wireless temp logging	*identify app for troubleshooting*	*DONE*	*Thingspeak. Others available but this seemed to be most user friendly and easiest to get going quickly*
wireless temp logging	*set up board with temp and pressure chip AND wireless module*	*DONE*	*Swapped Arduino UNO for MEGA (two serial outputs - handy for debugging)*
wireless temp logging	*Use Thingspeak channels to do wireless readout and verify setup works*	*DONE*	*Step could be omitted but allows for verification that module work .Code modified to recheck wifi connected. Modify code to take out network access when uploading to github*
wireless temp logging	Drop Thingspeak, replace by own code to do wireless readout and serve out to SQL Azure	INP	RaspPi may be more stable. Mobile Service set up on Azure with database. Modifying Arduino code to create JSON
wireless temp logging	C# create a regularly scheduled job to load to database		
wireless temp logging	Reliability test	INP	Thingspeak display has gaps - but downloaded data does not. Modified code to recheck connection. So far data looks consistent with no dropped information
wireless temp logging	Security test		
wireless temp logging	UnitLevel Tests		

Wireless monitoring is working successfully with the ESP8266 module attached. ThingSpeak describes itself as an open source application platform for the Internet of Things. Using ThingSpeak you can build an application around data collected by sensors. I'm planning on using this as an interim step to test the set up and then move to using ESP2866 directly. Data available at https://thingspeak.com/channels/53977 (temporarily a public channel) . No dropped data.

The next step is to migrate to Azure mobile services using SQL Azure. The mobile service has been set up and a new Arduino code is being modified to load directly to the database.

Next steps

Complete database load code and do reliability testing.

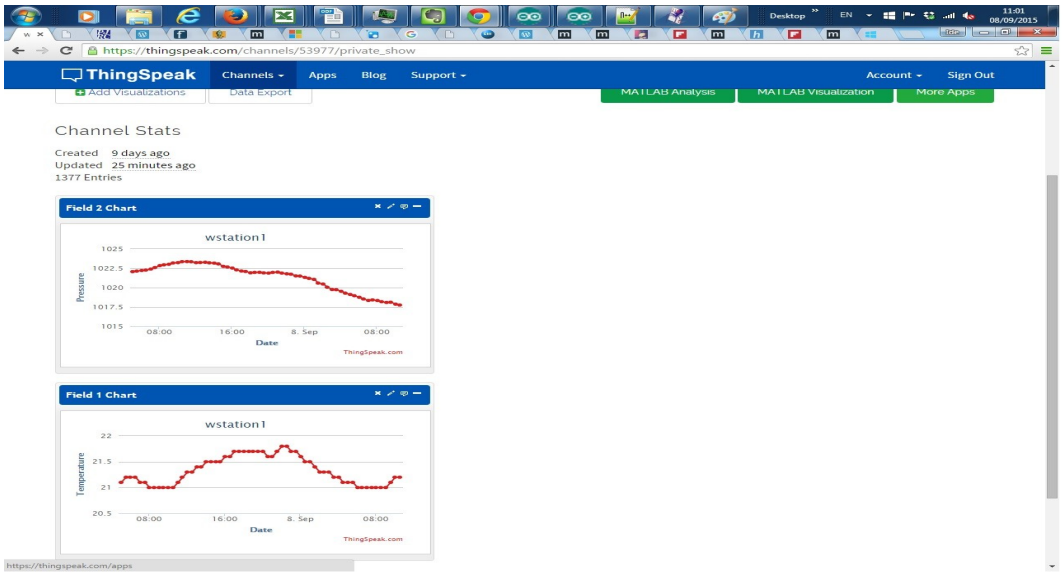
Use the temperature and pressure data to start the data visualisation portion of the project. Build a Arduino/Magnetometer/ESP8266 powered combination and start to load that to it's own database.

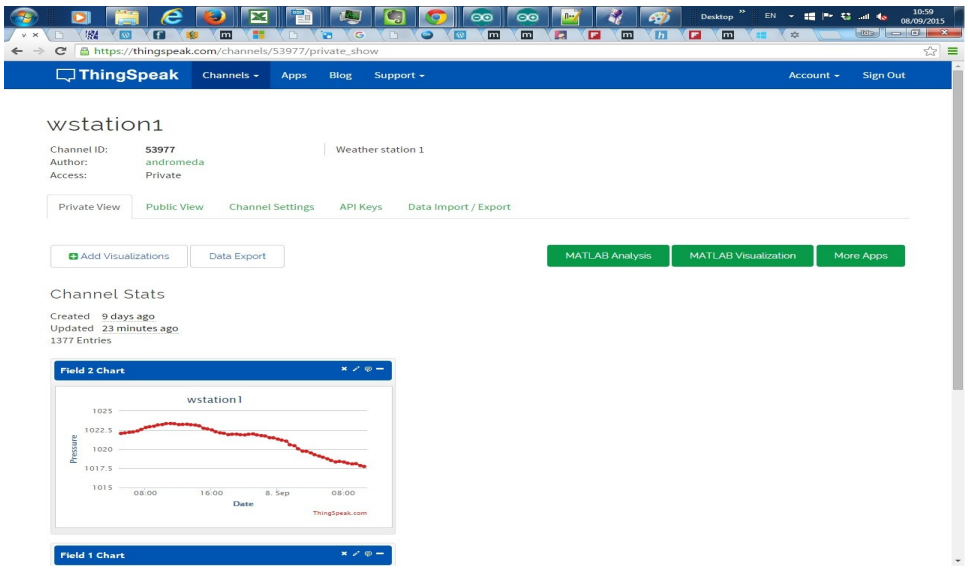
Reference

1. BMP180 : http://www.adafruit.com/products/1603
2. ESP2866: http://www.adafruit.com/products/2282
3. Thingspeak : https://thingspeak.com/

Appendix 1

Thingspeak screencaps of pressure and temperature - readings taken hourly


https://thingspeak.com/channels/53977/private_show
ThingSpeak
Channels
Apps
Blog
Support
Account
Sign Out
Add Visualizations
Data Export
MATLAB Analysis
MATLAB Visualization
More Apps
Channel Stats
Created 9 days ago
Updated 25 minutes ago
1377 Entries
Field 2 Chart
wstation1
Pressure
1025
1022.5
1020
1017.5
1015
08:00
16:00
8. Sep
08:00
Date
ThingSpeak.com
Field 1 Chart
wstation1
Temperature
22
21.5
21
20.5
08:00
16:00
8. Sep
08:00
Date
ThingSpeak.com
https://thingspeak.com/apps


https://thingspeak.com/channels/53977/private_show
ThingSpeak
Channels
Apps
Blog
Support
Account
Sign Out
wstation1
Channel ID: 53977
Author: andromeda
Access: Private
Weather station 1
Private View
Public View
Channel Settings
API Keys
Data Import / Export
Add Visualizations
Data Export
MATLAB Analysis
MATLAB Visualization
More Apps
Channel Stats
Created 9 days ago
Updated 23 minutes ago
1377 Entries
Field 2 Chart
wstation1
Pressure
1025
1022.5
1020
1017.5
1015
08:00
16:00
8. Sep
08:00
Date
ThingSpeak.com
Field 1 Chart

subgroup	Project Plan for weather stations	status	comment
wired temp logging	*set up board with temp and pressure chip*	*DONE*	
wired temp logging	*Download Arduino sketch and write program to output data*	*DONE*	
wired temp logging	*Create C# program that asks Arduino for data at timed intervals*	*DONE*	
wired temp logging	*C# program that builds files from Arduino data that witll work with SQL Azure*	*DONE*	
wired temp logging	*C# program that reads directory files into datatable and uploads datatable to SQl Azure*	*DONE*	
wired temp logging	*C# create a regularly scheduled job to load to database*	*DONE*	
wired temp logging	*SQL Azure database accessible with data*	*DONE*	*accessed with SSIS*
wired temp logging	*Reliability test*	*DONE*	*Fail. PC reboots too often. (std windows updates) Data set patchy. Looking for 99% uptime.*
wireless temp logging	*research alternatives to wired connections*	*DONE*	*Add ESP266 module to board. http://www.adafruit.com/datasheets/ESP8266_Specifications_English.pdf*
wireless temp logging	*identify app for troubleshooting*	*DONE*	*Thingspeak. Others available but this seemed to be most user friendly and easiest to get going quickly*
wireless temp logging	*set up board with temp and pressure chip AND wireless module*	*DONE*	*Swapped Arduino UNO for MEGA (two serial outputs - handy for debugging)*
wireless temp logging	*Use Thingspeak channels to do wireless readout and verify setup works*	*DONE*	*Step could be omitted but allows for verification that module work .Code modified to recheck wifi connected. Modify code to take out network access when uploading to github*
wireless temp logging	Drop Thingspeak, replace by own code to do wireless readout and serve out to SQL Azure	INP	RaspPi may be more stable. Mobile Service set up on Azure with database. Modifying Arduino code to create JSON
wireless temp logging	C# create a regularly scheduled job to load to database		
wireless temp logging	Reliability test	INP	Thingspeak display has gaps - but downloaded data does not. Modified code to recheck connection. So far data looks consistent with no dropped information
wireless temp logging	Security test		
wireless temp logging	UnitLevel Tests		
N wireless temp logging	Set up N wireless boards with access to power		Provided original device passes reliability (uptime), security, unit level testing
N wireless temp logging	Replicate original programming with unique ID for each board		
N wireless temp logging	Create regularly scheduled job to load to database		
Data visualisation	client side REST to display SQL Azure data so far	INP	
Data visualisation	Research display alternatives	INP	
Magnetometer loggers	Set up N wireless boards with access to power		
Magnetometer loggers	Replicate original programming with unique ID for each board		
N wireless temp logging	Create regularly scheduled job to load to database		
Magnetometer loggers	Create SQL database		
Magnetometer loggers	Add global dataset		
Final Package	Reliability test		
Final Package	Security test		
Final Package	UnitLevel Tests		
Tidying up	Move Arduino Code into Visual Studio C?		all code in same environment?
Improvements	Use Raspberry Pi as local server and interface to upload to the cloud and then to SQL Azure		