Portland State University

Maseeh College of Engineering

Team 31 Capstone Project

Open Bike Initiative: Air Quality Sensor Hub

System Test Plan

Team Members:
Ali Alavi
Robert Flory
Meng Lei
Pedro Munoz

Advisor:
Dr. Malgorzata Chrzanoska-Jeske
Sponsors:
Open Bike Inc.
Intel Corporation

Contents

1. <u>Introduction</u>

1.1 This Document

This document is intended to provide a comprehensive test plan for the evaluation of the Open Bike Initiative Air Quality Sensor Hub. It includes all the necessary background information for the system hardware and software. If you are familiar with the system, you may wish to skip to sections 4 and 5.

1.2 Conducting Tests

Each test must be completed in the order listed as the results of one test may affect the results of the next. If one test is not successful, then the results of the following tests will consequently be adversely affected. The procedures outlined will evaluate the following criteria:

- Performance
- Functionality
- Reliability
- Robustness
- Usability

1.3 Records and Results

Blank test record sheets can be found at the end of this document in the appendix or on the repository under the *tests* folder.

2. <u>System Reference</u>

2.1 System Requirements

Marketing Requirements	Engineering Requirements	Justification
1,5	**System will be capable of collecting levels of CO, and dust (particulate matter) in the environment.**	**Primary air pollutants defined by the EPA are ozone, nitrous oxides, sulfur dioxide, lead, particulate matter, and carbon monoxide. Sensors for detecting ozone, nitrous oxides, and lead are far too expensive (hundreds of dollars) to implement. Sensors for CO and particulate matter are affordable and thus implementable on many bikes.**
1,5	**System will be capable of collecting temperature, humidity levels, and GPS location.**	**Temperature and humidity are environmental factors that can affect pollutants and sensors and are therefore useful for data analysis. GPS location will be used to correlate the data with the position of the rider.**
2,3	**System will be housed in a small enclosure that allows copious amounts of airflow.**	**In order to effectively measure fluctuations in pollutant levels, the enclosure must allow for significant airflow.**
2,5	**System will be capable of storing large amounts of data internally. Data will be easy to access, analyze, and delete.**	**In order to accommodate for long bike rides and to keep the system memory from filling up too quickly, records should be stored on large memory units. These records should be written in an easy to view and**

		analyze format, and should be easily accessible by users.
1,5	**System will be capable of transmitting the sensor data to the main OBI unit.**	**In order to save on cost and to utilize the resources already available on the future OBI "smart-locks" (wireless data transmission), the unit will be capable of transmitting the sensor data to the main box.**
4	**System will be capable of passing power to the main OBI unit from a dynamo for charging of battery.**	**The OBI bicycles will have a hub dynamo to generate power. Our box will transmit the generated power to the main box in order to charge the batteries but will not make use of the power.**
1,4	**System will be powered via a regulated rechargeable battery which will turn on using a simple on/off switch.**	**The box will be completely self-contained and will require a power source. It will have a battery capable of lasting through a long bike ride which will be regulated to the appropriate voltage. In order to maximize ease of use, human interaction will be limited to a power switch will turn the system on and off.**
1,4	**System will connect with future OBI "smart-lock" unit through an RJ45 connection.**	**The future OBI box will have an RJ45 jack and wireless data to allow for communication with the outside world. Thus, in order to transmit the data and power to the main box in a cost effective manner, the systems will be connected via RJ45.**

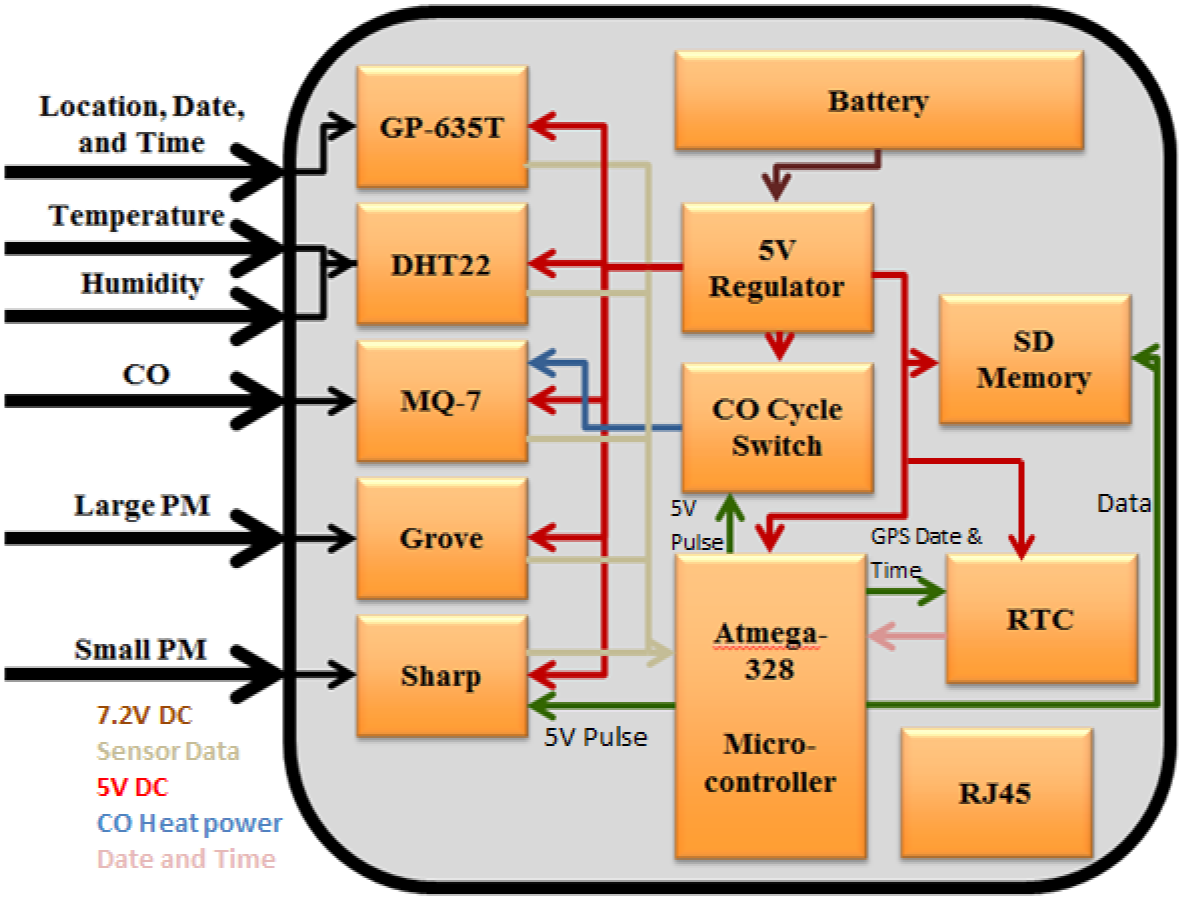
5	**System will be capable of time-stamping data.**	**In order to correlate the data with other sensors or to get an idea of the time of day for pollution information, each line of data will have a time stamp.**
1,4,5	**System must use an inexpensive microcontroller.**	**A microcontroller capable of communication and data collection from all sensors must be implemented.**
5	**System will collect sensor data every few seconds.**	**Since pollutant concentrations can fluctuate very frequently, the system must be capable of collecting data often.**
Marketing Requirements **1. The system should cost under $200 per unit to manufacture.** **2. The system should not interfere with bicycle rider.** **3. The system should be light and portable.** **4. The system should have somewhat low power consumption.** **5. The system should provide useful data.**		

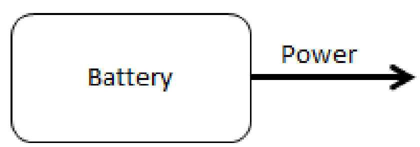
2.2 System Overview

The Air Quality Sensor Hub is meant to be installed on the front of the bicycle. The sensor hub consists of three primary air quality sensors capable of detecting small particulate matter, large particulate matter, and carbon monoxide. It also has sensors to detect temperature, humidity, and GPS. The system is self-contained and designed to write a new line of sensor data every 2 to 3 seconds, and store the data internally on an SD card in an easy to analyze format. The records are time-stamped using a real time clock and the SD card can easily be removed from the outside of the enclosure for data analysis. The system is powered with a rechargeable battery and all components are housed inside of a small enclosure with copious amounts of airflow. An on/off switch will start and stop data collection by supplying power to the system. The system is designed to be rugged enough for the average city bike ride and can be mounted to any bicycle with a rack.

2.3 Block Diagram and Design Documentation

The following pages contain an overview of the Sensor Hub system.

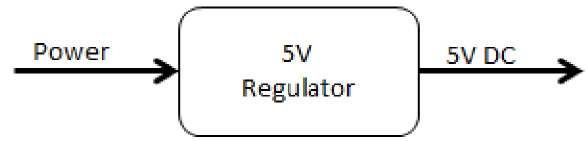


Module: Battery	

Inputs: None
Output: • **Power:** 7.2V DC
Functionality: • Supply power to the 5V linear regulator to power the system • A switch connects/disconnects the output signal to power the system on/off • Rechargeable 3300mAh Ni-MH battery

Module:

5V Regulator



Inputs:

- **Power:** 7.2V DC

Output:

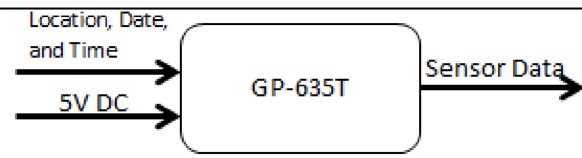
- **5V DC:** Regulated 5V DC signal

Functionality:

- The input signal enters a 7.2V to 5V linear DC to DC voltage regulator.
- The signal is smoothed using >= 10uF capacitors on the input and output.
- The signal is then fed to all powered components of the system.

Module:

GP-635T



Inputs:

- **Location, Date, and Time:** Satellite signal containing Location, Date, and Time.
- **5V DC:** Power for the sensor

Output:

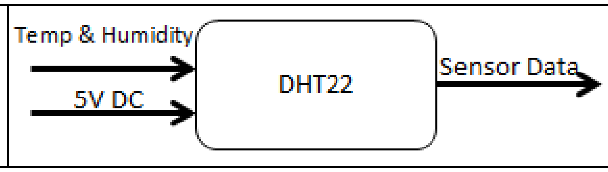
- **Sensor Data:** NEMA codes supplied by the GPS to indicate the Location, Date, and Time to the microcontroller.

Functionality:

- When connections to satellites are made, the GPS module receives the location, date, and time.
- The sensor translates the signals to NEMA codes which are translated to strings using the microcontroller.
- The module is powered with a 5V signal.

Module:

DHT22



Inputs:

- **Temp & Humidity:** The ambient temperature and ambient humidity in the air are detected by the sensor.
- **5V DC:** The sensor is powered with a 5V signal.

Output:

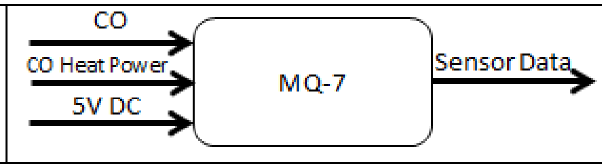
- **Sensor Data:** The sensor outputs the evaluated temperature and percent humidity to the microcontroller for processing.

Functionality:

- The sensor detects the ambient temperature and humidity in the surrounding air.
- The sensor sends the data to the microcontroller for processing and evaluation.
- The module is powered with a 5V signal.

Module:

MQ-7



Inputs:

- **CO:** The levels of CO in the surrounding air are detected by the sensor.
- **CO Heat Power:** 5V or 1.4V power for the heater coil for the sensor.
- **5V DC:** The sensor is powered with a 5V signal.

Output:

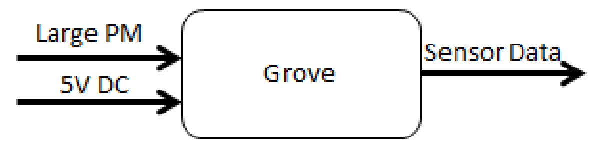
- **Sensor Data:** The sensor outputs an analog voltage to the microcontroller representing the relative amount of CO in the atmosphere.

Functionality:

- The sensor receives 5V as a reference for data communication with the microcontroller.
- The sensor has a coil which provides a resistance used to detect CO. This coil has two power cycles. As CO builds up on the coil, a 5V burn off (cleaning) phase is used at approximately 150mA for approximately 60s. After this phase a 1.4V phase is used for detection for approximately 90s. The cycles continually repeat.
- The sensors output is relative. In order to correlate the voltage with actual PPM values, certain tedious calibration techniques are necessary outlined in the datasheet.

Module:

Grove



Inputs:

- **Large PM:** Larger PM (>= 1um) is detected by the Grove sensor.
- **5V DC:** The sensor is powered with a 5V signal.

Output:

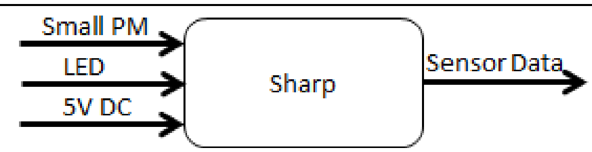
- **Sensor Data:** The concentration of PM in the surrounding air is detected by the sensor and transmitted to the microcontroller.

Functionality:

- The sensor outputs a digital signal proportional to the PM concentration (1um or larger) in the air.
- The sensor sends the data to the microcontroller for processing and evaluation.
- The module is powered with a 5V signal.

Module:

Sharp



Inputs:

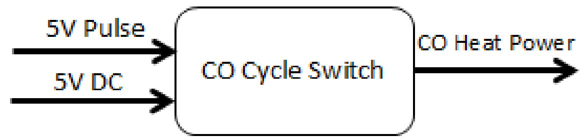
- **Small PM:** very fine PM is detected by the sharp sensor.
- **LED:** Signal from microcontroller to turn the LED in the sensor on and off.
- **5V DC:** The sensor is powered with a 5V signal.

Output:

- **Sensor Data:** The dust density in the air is detected by the sensor and transmitted to the microcontroller for processing.

Functionality:

- An infrared emitting diode and a phototransistor are used to detect the reflected light of very fine dust particles in the air.
- The sensor sends an analog voltage proportional to the very fine dust particles in the air to the microcontroller.
- To save power, an LED is turned on and off when detection takes place.
- The system is powered with a 5V signal.

Module:	
CO Cycle Switch	

Inputs:

- **5V Pulse:** a 5V pulse from the microcontroller that controls the CO sensor power cycle. It is high for 90s (1.4V detection phase) and low for 60s (5V burn-off phase).
- **5V DC:** The switch regulates the 5V signal to a 1.4V DC.

Output:

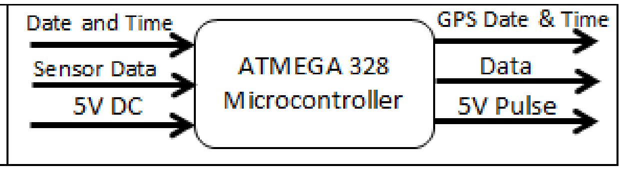
- **CO Heat Power:** The switch either passes the 5V signal to the MQ-7 directly, or passes the regulated 1.4V signal.

Functionality:

- The MQ-7 heat cycle is controlled using this module.
- The module regulates the 5V signal to 1.4V.
- A transistor network uses a 5V pulse that stays high for 90s and low for 60s.
- When the pulse is low, the 5V DC input is passed to the MQ-7 heater coil for the burn-off phase, when the pulse is low the regulated 1.4V signal is passed to the MQ-7 heater coil for the detection phase.

Module:

ATMEGA 328 Microcontroller



Inputs:

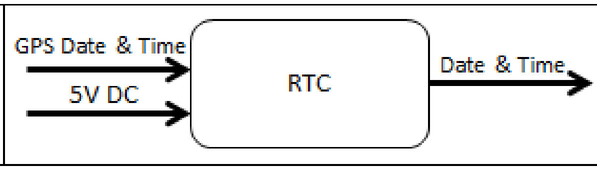
- **Date and Time:** The date and time is read from the RTC.
- **Sensor Data:** The CO, PM, temperature and humidity, and GPS data is read from the sensors.
- **5V DC:** The sensor is powered with a 5V signal.

Output:

- **GPS Date & Time:** The GPS date and time is used to set the RTC.
- **Data:** The processed sensor data is sent to the SD memory shield for storage.
- **5V Pulse:** 5V pulses are used to adjust turn on and off the LED for the sharp sensor, and to adjust the heating phase of the MQ-7.

Functionality:

- The microcontroller is powered using a 5V DC signal.
- The microcontroller is flashed with the Arduino boot loader and is programmed using the C based Arduino environment and language.
- The microcontroller is programmed to send a 5V pulse to set the CO sensor heat phase using a 5V pulse.
- The microcontroller is programmed to turn the sharp sensor LED on and off using a 5V pulse.
- The microcontroller is programmed to poll the sensor data approximately every 3 seconds, and organizes the data in an easy to analyze format.
- The microcontroller correlates the environmental data to GPS location as well as the date and time from the RTC.
- The microcontroller sends the data to the SD shield to store in an easy to analyze format in a .txt file.

Module: RTC (Real-Time Clock)	

Inputs:

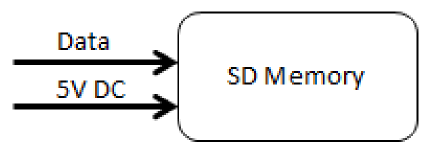
- **GPS Date & Time:** The microcontroller uses the obtained GPS date and time to set the RTC.
- **5V DC:** The RTC is powered using a 5V DC signal.

Output:

- **Date & Time:** The date and time are read by the microcontroller and written with each record.

Functionality:

- The module is independently powered using a CR2032 3.3V battery to maintain the date and time.
- The module uses I2C protocol to communicate with the microcontroller.
- The module has an independent 32.766 kHz crystal.
- The module is powered using a 5V DC signal.

Module: SD Memory	

Inputs:

- **Data:** The data is sent to the SD memory shield and written to an SD card in a .txt file.
- **5V DC:** The module is powered with a 5V DC signal.

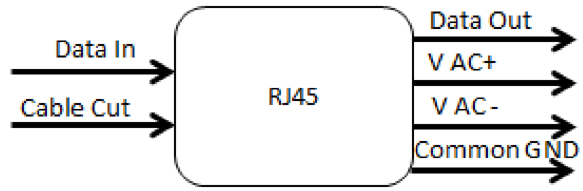
Output:

None

Functionality:

- The SD memory is used to store records.
- The records are written to a .txt file by the microcontroller.
- The SD card can easily be removed and inserted from the outside of the enclosure to the shield.
- The shield is powered using a 5V DC signal.

The RJ45 is not currently used by the system. In the future, it will be capable of connecting to the OBI Smart-locks and communicating via RS232 to transmit and receive data and pass dynamo power. The module will function as outlined below.

Module: RJ45	

Inputs:

- **Data In:** The RJ45 will be able to take data input from the OBI Smart Lock which will be sent to the microcontroller.
- **Cable Cut:** This line is grounded, and if the cable is cut it will result in an alarm and a siren sounding.

Output:

- **Data Out:** The microcontroller will send data output to the RJ45 for transmission to the OBI Smart Lock.
- **V AC+:** AC+ signal from the dynamo hub to be transferred to the OBI Smart Lock for charging the battery.
- **V AC-:** AC- signal from the dynamo hub to be transferred to the OBI Smart Lock for charging the battery.
- **Common GND:** Shared Common GND signal.

Functionality:

- The module will function as a connector between the air quality hub and the OBI Smart Lock.
- It will be capable of receiving data to be sent to the microcontroller.
- It will have a cable cut signal in case of theft.
- It will be able to send data to the OBI Smart Lock which can be transferred to the cloud.
- It will be capable of transmitting Dynamo power.

3. <u>Pretest Preparation</u>

3.1 Test Equipment
- Oscilloscope
- Multimeter
- Computer system with SD card reader and access to web.
- Assembled OBI front box
- Incense
- Thermometer
- Arduino uno for programming
- Box large enough to place over system and incense
- Spreadsheet software such as MS Excel.

3.2 Tester Qualification Level
- Tester is expected to have basic understanding of analog and digital circuit design, microcontroller (Arduino or Atmel) programming and safe operations of related lab equipment.

3.3 Test Setup

Test Case Number	Test ID	Test Case Name
1	OBI_UT_#01	Power Supply
2	OBI_UT_#02	SD Shield
2	OBI_UT_#03	Sharp Dust Sensor
3	OBI_UT_#04	Grove Dust Sensor
4	OBI_UT_#05	MQ7 CO Sensor
5	OBI_UT_#06	DHT22 Temperature & Humidity Sensor
6	OBI_UT_#07	GPS Unit
7	OBI_UT_#08	RTC Chip
8	OBI_IT_#01	All sensor reading
9	OBI_IT_#02	Robustness and Durability

4. System Tests (Step by Step)

4.1 Unit Test Cases

Case Study 1: Power Supply

- The supplied battery is a 7.2 V Ni-MH DC battery. The output is regulated through a linear voltage regulator
- The output of the power supply should be 4.8-5.1V DC.

<table>
<tr><td colspan="8">Test Writer: Meng Lei / Ali Alavi</td></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="2">Power Supply</td><td colspan="2">Test ID #:</td><td colspan="2">OBI_UT_#01</td></tr>
<tr><td colspan="2">Description:</td><td colspan="2">Determine if the power supply provides 4.8-5.1V DC</td><td colspan="2">Type:</td><td colspan="2">White box</td></tr>
<tr><td colspan="8">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td colspan="2"></td><td colspan="2">Date:</td><td colspan="2"></td></tr>
<tr><td colspan="2">Hardware Version:</td><td colspan="2"></td><td colspan="2">Time:</td><td colspan="2"></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="6">Connect the battery to the voltage regulator on daughter board using the power switch. Use output jumper to connect voltage regulator output to multimeter.</td></tr>
<tr><td>Step</td><td>Action</td><td colspan="2">Expected result</td><td>Pass</td><td>Fail</td><td>N/A</td><td>Comments</td></tr>
<tr><td>1</td><td>Connect battery</td><td colspan="2">Multimeter should read 5V</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Make sure battery is fully charged, connect 10Ω load</td><td colspan="2">The battery should still be working after 4 hours. Plan accordingly since the test takes some time.</td><td></td><td></td><td></td><td></td></tr>
</table>

Case Study 2: SD Shield

- The SD shield uses serial communication with the Microcontroller to store record data onto an SD card in a text file.
- The SD shield supports memory cards formatted in FAT16 or FAT32 filesystems.
- The text files written will be named according to the date the data was written (MM-DD-YY).

<table>
<tr><td colspan="7">Test Writer: Ali Alavi</td></tr>
<tr><td colspan="2">Test Case Name:</td><td>SD Shield Unit Test</td><td colspan="2">Test ID #:</td><td colspan="2">OBI_UT_#02</td></tr>
<tr><td colspan="2">Description:</td><td>Determine if record data is being written to the SD card every 2 to 3 seconds.</td><td colspan="2">Type:</td><td colspan="2">White box</td></tr>
<tr><td colspan="7">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td></td><td colspan="2">Date:</td><td colspan="2"></td></tr>
<tr><td colspan="2">Hardware Version:</td><td></td><td colspan="2">Time:</td><td colspan="2"></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="5">Format an SD card to FAT16 or FAT32 filesystems. Insert into the SD shield.</td></tr>
<tr><td>Step</td><td>Action</td><td>Expected result</td><td>Pass</td><td>Fail</td><td>N/A</td><td>Comments</td></tr>
<tr><td>1</td><td>Program chip, power on for 30 seconds, check SD card for data.</td><td>A red LED on the shield above the SD card should blink every 2 to 3 seconds when the system is powered on indicating data is being written.
When analyzing the card, there should be record lines written every 2 to 3 seconds to a text file.</td><td></td><td></td><td></td><td></td></tr>
</table>

Case Study 3: Sharp dust sensor

- The Sharp dust sensor outputs an analog voltage relative to PM with small diameter such as fine dust and smoke.
- In the program loop, Sharp dust sensor should be able to provide PM concentration every 2 to 3 seconds.

Test Writer: Meng Lei / Ali Alavi							
Test Case Name:		Sharp dust sensor test				**Test ID #:**	OBI_UT_#03
Description:		The sharp dust sensor should be able to provide an analog voltage relative to the PM concentration. The microcontroller will write the data to a text file stored on the SD card.				**Type:**	White box
Tester information							
	Name of tester:					**Date:**	
Hardware Version:						**Time:**	
Setup:		The Sharp dust sensor has 6 pin connections. Pin 1 is low-pass filtered 5V signal, pin 2 is GND, pin 3 is LED, pin 4 is GND, pin 5 is 5V DC, and pin 5 is the data pin. On fabricated PCB, data is connected to microcontroller analog pin 3, and LED is connected to digital pin 6. Ensure SD card is inserted.					
Step	**Action**	**Expected result**	**Pass**	**Fail**	**N/A**	**Comments**	
1	Program chip, power on, and check SD card data	A PM concentration value should be written to a text file stored on the SD memory card every 2 to 3 seconds.					
2	Attach TP_SHARP to multimeter	Multimeter should be able to read a voltage indicating the analog output of the sensor.					
3	Light up incense near sensor and check SD card data	The output written to text file should increase dramatically as the incense produces a large number of PM. As the sensor is exposed to more incense smoke, the multimeter should display a higher voltage proportionally.					

Case Study 4: Grove dust sensor

- The Grove dust sensor, compared to the Sharp, detects PM with relatively large diameters.
- In the program loop, Grove dust sensor should be able to provide PM concentration every 2 to 3 seconds.

<table>
<tr><td colspan="9">Test Writer: Meng Lei / Ali Alavi</td></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="3">Grove dust sensor test</td><td colspan="3">Test ID #:</td><td>OBI_UT_#04</td></tr>
<tr><td colspan="2">Description:</td><td colspan="3">The Grove dust sensor should be able to provide a digital output relative to the PM concentration. The microcontroller will write the data to a text file stored on the SD card.</td><td colspan="3">Type:</td><td>White box</td></tr>
<tr><td colspan="9">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td colspan="3"></td><td colspan="3">Date:</td><td></td></tr>
<tr><td colspan="2">Hardware Version:</td><td colspan="3"></td><td colspan="3">Time:</td><td></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="7">The Grove dust sensor has 5 total pins 3 of which are used for connections. Pin 1 is GND, pin 3 is 5V DC, and pin 4 is the output. On fabricated PCB, the data pin is connected to the microcontroller digital pin 3. Ensure SD card is inserted.</td></tr>
<tr><td>Step</td><td>Action</td><td>Expected result</td><td>Pass</td><td colspan="2">Fail</td><td>N/A</td><td colspan="2">Comments</td></tr>
<tr><td>1</td><td>Program chip, power on, and check SD card data</td><td>A PM concentration value should be written to a text file stored on the SD card every 2 to 3 seconds.</td><td></td><td colspan="2"></td><td></td><td colspan="2"></td></tr>
<tr><td>2</td><td>Attach TP_GROVE to oscilloscope</td><td>Bursts of 5V pulse should be visible on oscilloscope.</td><td></td><td colspan="2"></td><td></td><td colspan="2"></td></tr>
<tr><td>3</td><td>Light up incense near sensor for a minute and check SD card data</td><td>The output written to the text file should increase dramatically as the incense produces a large number of PM.</td><td></td><td colspan="2"></td><td></td><td colspan="2"></td></tr>
</table>

Case Study 5: MQ7 CO Sensor

- The MQ7 is a carbon monoxide sensor used for sensing CO levels in the air capable of sensing CO anywhere from 20-2000 ppm.
- The MQ7 should output an analog output representing the concentration of CO for 90 seconds, and will have a burn off or clean phase for 60 seconds.
- In the program loop, the output of the sensor is recorded every 2 or 3 seconds.

<table>
<tr><td colspan="9">Test Writer: Meng Lei / Ali Alavi</td></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="3">MQ7 CO Sensor</td><td colspan="3">Test ID #:</td><td>OBI_UT_#05</td></tr>
<tr><td colspan="2">Description:</td><td colspan="3">MQ7 has two phases of operation. The sensor will have a 5V heating phase for 60 seconds and a 1.4V detection phase for 90 seconds. During the heating phase the text file should display -- under CO read and an analog output under CO heat. During the read phase the text file should display -- under CO heat and an analog output under CO read relative to the CO concentration.</td><td colspan="3">Type:</td><td>White box</td></tr>
<tr><td colspan="9">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td colspan="3"></td><td colspan="3">Date:</td><td></td></tr>
<tr><td colspan="2">Hardware Version:</td><td colspan="3"></td><td colspan="3">Time:</td><td></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="7">With the board powered on, the voltage input to the MQ7 can be tested on the TP_CO_PWR test point using a digital multi-meter. The output of the CO sensor can be viewed on the test point labeled TP_CO using a digital multi-meter. Ensure SD card is inserted.</td></tr>
<tr><td>Step</td><td>Action</td><td>Expected result</td><td>Pass</td><td colspan="2">Fail</td><td>N/A</td><td colspan="2">Comments</td></tr>
<tr><td>1</td><td>Program chip, power on, check SD card data</td><td>With the multimeter connected to TP_CO_PWR it should read 4.8-5.1V for 60 seconds, and 1.3-1.5V for 90 seconds. The output values should be written to the SD card every 2 to 3 seconds.</td><td></td><td colspan="2"></td><td></td><td colspan="2"></td></tr>
</table>

2	Light up incense and hold above sensor for a minute and check SD card data	CO levels should rise and thus, with the multimeter connected to TP_CO the voltage should rise.				

Case Study 6: DHT22 Temperature and Humidity Sensor

- The DHT22 is a temperature and humidity sensor that detects the ambient temperature and humidity levels of its environment.
- The sensor outputs an analog signal corresponding to the temperature and the percent humidity detected.
- In the program loop, the temperature and humidity will be written to a text file stored on an the SD card every 2 to 3 seconds.

<table>
<tr><td colspan="7">Test Writer: Ali Alavi</td></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="2">DHT22 Temperature & Humidity Sensor Test</td><td colspan="2">Test ID #:</td><td>OBI_UT_#06</td></tr>
<tr><td colspan="2">Description:</td><td colspan="2">The DHT22 should be able to provide analog outputs relative to the temperature and humidity or the surrounding air.
The microcontroller will poll the data every 2 to 3 and write it to a text file stored on the SD card.</td><td colspan="2">Type:</td><td>White box</td></tr>
<tr><td colspan="7">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td colspan="2"></td><td colspan="2">Date:</td><td></td></tr>
<tr><td colspan="2">Hardware Version:</td><td colspan="2"></td><td colspan="2">Time:</td><td></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="5">The DHT22 has 4 connection pins. Pin 1 is VDD power (5V DC), pin 2 is the data signal out, pin 3 is not used, and pin 4 is GND. On the fabricated system, the signal out is connected to the microcontroller analog pin 1. Ensure SD card is inserted.</td></tr>
<tr><td>Step</td><td>Action</td><td>Expected result</td><td>Pass</td><td>Fail</td><td>N/A</td><td>Comments</td></tr>
<tr><td>1</td><td>Program chip, power on,</td><td>Temperature and humidity should be written to a text file on the SD card every 2 to 3 seconds.</td><td></td><td></td><td></td><td></td></tr>
</table>

	check SD card data					
2	Attach TP_TEMP to oscilloscope	Analog voltages should be visible relative to the temperature and humidity when they are polled from the sensor.				
3	Verify temperature by checking SD card data	Using a thermometer, check the ambient temperature and compare to the sensor output on text file.				
4	Verify humidity but checking SD card data.	Boil some water and place it under an enclosure with the sensor, the percent humidity written to the text file should rise.				

Case Study 7: GPS Unit

- The GPS unit is a GP-635T sensor communicating with the microcontroller via TTL.
- The GPS will only function outdoors or directly next to a window.
- The GPS unit is used to set the real time clock, and to write the GPS location to record line in a text box on the SD shield.

Test Writer: Ali Alavi						
Test Case Name:		GPS Unit Test		**Test ID #:**		OBI_UT_#07
Description:		Determine whether the GPS unit is functioning correctly, obtaining the time, date, and location.		**Type:**		White box
Tester information						
	Name of tester:			**Date:**		
Hardware Version:				**Time:**		
Setup:		GPS unit is connected to the TX and RX lines on the fabricated system. It uses 5V DC for power and is activated when the system is turned on. Ensure the system is outside or directly next to a window in order for the sensor to have successful satellite communication.				
Step	**Action**	**Expected result**	**Pass**	**Fail**	**N/A**	**Comments**

1	Program chip, power on, wait approximately 3 minutes, check SD card data for latitude and longitude with a mapping website (eg. google maps).	The data on the SD card for location should read 0 for both latitude and longitude when the system is first turned on. When the GPS successfully establishes a satellite link, the latitude and longitude will be displayed. Checking these coordinates should corresponding to the system's location. If this is the first time the system has been powered on and a satellite link has been established, the correct time should be displayed in UTC format as well.				

Case Study 8: RTC Chip

- The RTC chip is a DS1307 that operates via I2C protocol.
- When the system is off, the chip stays powered on with a 3.3V CR2032 battery to maintain the time and date.
- The system uses the information obtained by the GPS to set the chip and uses the chip to write the correct date and time to the SD memory card with each record.

Test Writer: Ali Alavi			
Test Case Name:	RTC Chip test	**Test ID #:**	OBI_UT_#08
Description:	The first time the system is powered on, the chip will be programmed with the correct date and time as soon as a GPS signal is found. It will keep its internal clock running even when the system is powered off. If for any reason it fails to keep the correct date and time, it will be programmed again the next time a GPS signal is found. The microcontroller will get the date and time from the RTC every 2 to 3 seconds.	**Type:**	White box
Tester information			
Name of tester:		**Date:**	
Hardware Version:		**Time:**	

<table>
<tr><th colspan="2">Setup:</th><td colspan="5">The RTC chip requires a 3.3V CR2032 battery for operation. Ensure that this battery is inserted into the battery holder on the fabricated system.</td></tr>
<tr><th>Step</th><th>Action</th><th>Expected result</th><th>Pass</th><th>Fail</th><th>N/A</th><th>Comments</th></tr>
<tr><td>1</td><td>program chip, power on, and wait 2-3 minutes near a window and check SD card data.</td><td>When the GPS gets a satellite signal, the chip will be programmed with the correct date and time. Every 2 to 3 seconds this information is written to the SD card. Ensure this information is correct.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>Turn system off for 2-3 minutes, power on again for 30 seconds away from window and check SD card data.</td><td>The RTC should have maintained the correct date and time despite having no GPS signal.</td><td></td><td></td><td></td><td></td></tr>
</table>

4.2 Integral Test Cases

Case Study 9: All Sensor Reading Test

Test Writer:			
Test Case Name:	All Sensor Reading Test	**Test ID #:**	OBI_IT_#01
Description:	This integral test is a combination of all previous tests and should ensure that all sensors are functioning correctly and the data is effectively written to the SD shield.	**Type:**	White box
Tester information			
Name of tester:		**Date:**	
Hardware Version:		**Time:**	
Setup:	Program chip, ensure the system is outside or directly next to a window.		

Step	**Action**	**Expected result**	**Pass**	**Fail**	**N/A**	**Comments**
1	Turn system on, wait 3 to 5 minutes.	The system should turn on and the red LED on the SD shield should begin blinking every 2 to 3 seconds.				
2	Light incense and place box over the system allowing smoke to build up. Wait 5 to 10 minutes, remove box, wait 2 to 3 minutes, turn system off, and check SD data.	The data on the SD card should have the correct date and time. After a satellite link was established by the GPS, it should also display the correct latitude and longitude (verifiable using mapping website). The values should have fluctuated according to the smoke buildup with PM and CO levels rising as more smoke built up.				
3	Import text file to spreadsheet program.	The spreadsheet should automatically be populated in the correct format. The top row should				

		be the headers, while each following row should be a line of sensor records.				
4	Create a graph for Sharp values.	The graph should show a rise in output values as the box filled with smoke and a drop as the box was removed.				
5	Create a graph for Grove values.	The graph should show a rise in output values as the box filled with smoke and a drop as the box was removed.				
6	Check CO levels by creating a graph for CO read values vs. CO heat values.	The graph should show the CO Heat values as consistently higher than the CO read values. CO read values should be low until the box was placed over the system and smoke built up. All values should rise as the smoke filled the box and drop as the box was removed.				
7	Check temperature with thermometer.	The temperature should be close to the thermometer reading.				
8	Check humidity	The percent humidity should stay fairly constant throughout the experiment.				

Case Study 10: Robustness and Durability Test
Test ID#: OBI_IT_#02

- The robustness and durability test can include several aspects. Bike riding, even on smooth tarmac can cause moderate vibrations and shocks. The OBI front box is designed to be functional under the stress of an average city bike ride. The testing procedure is to install the front box on a bike and take it to a test run. During the test run, make sure to stop every now and then to check if the battery is still powering the system, if there's still data being written to the SD card (the red LED on the SD card shield should be blinking every few seconds). After the test run, check if there are any irregularities in the record which would indicate one or more sensors have malfunctioned, or other components haven't been functioning very well.

5. Appendix

<table>
<tr><td colspan="7">Test Writer:</td></tr>
<tr><td colspan="2">Test Case Name:</td><td colspan="3"></td><td>Test ID #:</td><td></td></tr>
<tr><td colspan="2">Description:</td><td colspan="3"></td><td>Type:</td><td></td></tr>
<tr><td colspan="7">Tester information</td></tr>
<tr><td></td><td>Name of tester:</td><td colspan="3"></td><td>Date:</td><td></td></tr>
<tr><td colspan="2">Hardware Version:</td><td colspan="3"></td><td>Time:</td><td></td></tr>
<tr><td colspan="2">Setup:</td><td colspan="5"></td></tr>
<tr><td>Step</td><td>Action</td><td>Expected result</td><td>Pass</td><td>Fail</td><td>N/A</td><td>Comments</td></tr>
<tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>