Test Plan for the

Real-time Air Monitoring System

Project Team: Marvin Lee
Sam To
Tessa Ryan
Rouzbeh Roshanravan

Contact Person: Rouzbeh Roshanravan
rra19@sfu.ca

Submitted to: Dr. Andrew Rawicz – ENSC 440
Steve Whitmore – ENSC 305

School of Engineering Science
Simon Fraser University

Issued date: March 14, 2013

Revision: 1.3

Test Plan

Testing begins with examining the compliance of each part of the system through individual testing. These parts include sensors, the GSM module, the fan (ventilation system), the SD module, and the Mamba shield which provides the main communication link between the system parts. After each individual part is fully tested and is confirmed that each part is functional Airtack will test the device in action under extreme conditions.

1.1 Unit Testing

To verify that each part is working properly Airtack will test each individual part as follows:

1.1.1 GSM Module:

1. Monitor the Arduino from the serial port after inserting the SIM card.
2. If the status of the device is set to active, then both the SIM Card and the antenna are functional.
3. Program the Arduino to send a simple SMS text message to a phone number.
4. Send an SMS text message to the SIM Card and monitor the status through the serial port.

1.1.2 Fan:

1. Airtack make a simple circuit with the fan.
2. Airtack will provide an input to the fan and will monitor the results.
3. If the fan starts to spin upon receiving the input, the fan is functional.

1.2 Sensor Testing

Our system has five different sensors. Each sensor will be tested and monitored separately to ensure that each sensor is functional. The purpose of this testing is only and only to prove that each sensor works properly and calibration of the sensor is not performed by the following measures.

Testing of all each sensor will take place in an enclosed transparent plastic box.

1.2.1 $CO_{(g)}$ Sensor:

1. A simple circuit is created on a bread board.
2. Sensor is placed on the circuit.
3. The output is monitored through serial port on our computer.
4. Sensor is placed under the transparent box.
5. An option to test the sensor at this point is to spray some $CO_{(g)}$ from a Carbon Monoxide Test Kit.
6. If the sensor's output showed the changes, then the sensor is functional.

Calibration of the device involves repeating the above steps. Different known concentrations of Carbon Monoxide should produce different responses, which can be monitored from the computer through USB to serial. The response of the RTAMS must satisfy the requirements stipulated in the functional specification documentation, particularly CSA Standards. [4] [5] [6]

1.2.2 Natural Gas Sensor:

1. A simple circuit is created on a bread board.
2. Sensor is placed on the circuit.
3. The output is monitored through serial port on our computer.
4. Sensor and calibrated natural as monitor is placed under the glass box.
5. A small amount of methane of known concentration is sprayed near the sensor.
6. If the sensor's output showed the changes, then the sensor is functional.

Calibration of the device involves repeating the above steps. Different known concentrations of Natural Gas should produce different responses, which can be monitored from the computer through USB to serial. The response of the RTAMS must satisfy the requirements stipulated in the functional specification documentation, particularly CSA Standards. [4] [5] [7]

1.2.3 Smoke Sensor:

1. A simple circuit is created on a bread board.
2. Sensor is placed on the circuit.
3. The output is monitored through serial port on our computer.
4. Sensor is placed under the glass box.
5. Airtack will use a fog machine.
6. Airtack will turn on the fog machine and monitor the results through the serial port.
7. Airtack should see an increase in the data produced by our sensor.
8. Once the glass box and the fog machine are removed the monitored level should go back up to its initial value.

Calibration of the smoke sensor device was done at the time of production. Information can be found on the device data sheet. [8]

1.2.4 Humidity Sensor:

1. A simple circuit is created on a bread board.
2. Sensor is placed on the circuit.
3. The output is monitored through serial port on our computer.
4. Sensor is placed under the glass box.
5. Airtack will use a small portable air humidifier.
6. The air humidifier is placed under the glass box along with the sensor.
7. Airtack will turn on the air humidifier and monitor the results through the serial port.
8. Airtack should see an increase in the data produced by our sensor.

9. Once the glass box and the humidifier are removed the monitored level should go back up to its initial value.

1.2.5 Temperature Sensor:

1. A simple circuit is created on a bread board.
2. Sensor is placed on the circuit.
3. The output is monitored through serial port on our computer.
4. Sensor is placed under the glass box.
5. Airtack will use an air can.
6. A small amount of air will be directly sprayed on the sensor.
7. Since the air coming out of the can has a lower temperature a change(decrease) in data should be observed on the monitored data coming from the sensor.
8. Once the can is not sprayed anymore the monitored level should go back up to its initial value.

1.3 Communication Testing

Communication testing will begin once unit testing and sensor testing has been completed. Several different setup scenarios will be analyzed.

1.3.1 Two sensor modules:

1. Two sensor modules will be installed in separate rooms
2. Wait for both modules to complete initialization
3. One of the sensors will be stimulated with smoke from a fog machine
4. If the alarm sound on both modules, then Airtack know that the communication has been set up properly

1.3.2 Two sensor modules and a server:

1. A server module is installed
2. Wait for the server to complete initialization
3. Install the two server modules
4. Wait for both sensor modules to complete initialization
5. A computer is connected to the server
6. The data is read from the server
7. If the data from the sensor modules is correct, the server communication functions are working.

1.3.3 Sensor, Server, and Active:

1. The server module is installed and wait for initialization
2. The sensor module is installed and wait for initialization
3. The active module is installed and wait for initialization
4. Configure the server to turn on a fan when smoke is detected

5. Stimulate the sensor module with smoke from a fog machine
6. If the fan connected to the active module turns on, the server module is correct configured to control the active module.

1.4 Software/Data Analysis Testing and Corner Cases

The main stand-alone software component of this device is for data analysis. The central server of the device is connected to a PC through serial and data is transferred from the central server's SD card to the computer for analysis. Three types of graphs are available- short term data trends, long term data trends and correlation value bar graphs. Data analysis is performed with scripts written in and for **R**, an open source statistical computing application. Software testing has been ongoing throughout its development to ensure predictable and accurate behavior even before integration with system hardware. During the integration process, further tests will be executed as per below.

1.4.1 CSV Input Testing

This test must ensure CSV data input has the expected number of variables. CSV data should be input into the software in the following format:

second, minute, hour, day, month, year, sensor_address, data1, data2, data3, data4, data5

The software shall first check that CSV data is received from serial. If this is successful, it should check the number of parameters (columns) in the CSV input and compare it to the expected number of parameters.

Failure Conditions: CSV data not received, **OR** CSV data received **AND** (number of parameters) != (expected number of parameters). Software will throw an error to indicate data not received or an error to indicate missing data due to a missing sensor not connected to the system hardware.

Success Conditions: CSV data is received **AND** (number of parameters) = (expected number of parameters). Software will initiate normal data analysis procedures.

1.4.2 PDF Output Generation

If an output PDF already exists and is still open when a new report is generated as part of the real-time update, the software should create a new report by the same name with an appended number following the previous filename. This will address the error R throws when a file meant to be edited is still open.

Failure Condition: An error is still thrown by R: *Error in pdf("DataAnalysis.pdf") : cannot open file ' DataAnalysis.pdf'*.

Success Condition: A new PDF is successfully created with an appended number.

1.4.3 Matching Sensor Samples for Analysis

For correlation calculations and computation of bar graphs to illustrate correlation coefficients, data from two sensors at a time must be paired and analyzed together. This process requires both sets of data to be the same size. A corner case has been programmed into the code to check if each individual data vector for analysis has an identical number of samples so pairing will be successful. If there is a problem with pairing data, an error will be thrown and hardware operation/communication modules should be checked for correct operation.

Failure Condition: An error is thrown by R:

Error in cor(exampletest, datamatrix$data5, use = "all.obs", method = "kendall") :

incompatible dimensions

Success Condition: All data vectors of interest are of identical lengths. Software will initiate normal data analysis procedures.

1.4.4 Dangerous Environmental Factor Levels Indication

Action alerts for dangerous levels are sent by the system's central server when it has determined that levels of an environmental factor have gone outside of acceptable ranges. It is not crucial that the data analysis software output, which just generates trending and correlation information also alert the user to problems, however this feature has been included to enhance the quality of information the outputted graphs provide. This corner case ensures that when a sample in the graph's range has stepped outside of acceptable threshold levels, a red warning message will be included on the graph.

1.5 System Testing

1.5.1 Normal Case 1: No Sensors Activated

User Input: User plugs in the fan, sensors, and the server to the outlet.

Conditions: There are no significant substances in the air to trigger any of the sensors.

Expected Observations: Each individual part is connected to the server. None of the sensors are triggered. The data monitored from all sensors shows a moderate normal status. The fan is off. No text message is sent.

1.5.2 Normal Case 1: No Sensors Activated

User Input: User plugs in the fan, sensors, and the server to the outlet. One sensor is activated.

Conditions: One of smoke, $CO_{(g)}$ or natural gas, temperature, humidity levels are higher than the normal level.

Expected Observations: Each individual part is connected to the server. Data monitored shows which sensor is triggered. For any sensor activated a text message is sent to specified number indicating the level of the gas, level of danger, and the buzzer is triggered. If either of $CO_{(g)}$ or natural gas sensors are activated the fan will also start to run. Once the level of substance detected drops to normal level a text is sent again to inform user that everything is back to normal, the buzzer and fan will stop as well.

1.5.3 Extreme Case2: All Sensors Activated at Once

User Input: User plugs in the fan, sensors, and the server to the outlet. All sensors are activated simultaneously.

Conditions: All devices are communicating with the main server. All of the smoke, $CO_{(g)}$, natural gas, temperature, and humidity levels are higher than the normal level.

Expected Observations: Each individual part is connected to the server. Data monitored shows that all of the sensors are triggered. For all sensors a text message is sent to specified number indicating each level of the gases, level of danger, and the buzzer is triggered. Since, smoke sensor is also activated, the fan will NOT start. The buzzer will be triggered. Once the level of substance detected drops to normal level a text is sent again to inform user that everything is back to normal, the buzzer will stop as well.