# *Automated Safety Warning Controller System Testing Plan and Summary and System Evaluation Plan and Summary*


By

Daniel Richter, Research Associate
Kelvin Bateman, Research Associate
Douglas E. Galarus, Program Manager, Senior Research Associate

Western Transportation Institute
College of Engineering
Montana State University – Bozeman


For

The Caltrans Division of Research and Innovation

Attention: Sean Campbell
Attention: Ian Turnbull

March 22nd, 2013

# DISCLAIMER

# ACKNOWLEDGEMENTS

The authors would like to thank Sean Campbell, Chief, ITS Special Project Branch, California Department of Transportation, Division of Research and Innovation, Ian Turnbull, Chief, Office of ITS Engineering and Support, Caltrans District 2, Ken Beals, ITS Engineer, and the staff of Caltrans District 2 for their support and assistance on this project.

# TABLE OF CONTENTS

# EXECUTIVE SUMMARY

The California Department of Transportation (Caltrans) has contracted with the Western Transportation Institute (WTI) at Montana State University (MSU) to develop an "Automated Safety Warning Controller." The controller will interface with roadside devices such as sensors and signs. The controller will allow for automated data collection and application of best practice algorithms to analyze sensor data and to actuate related warning messages to motorists. For instance, wind warning messages might be actuated on a changeable message sign (CMS) when wind speed, as read from a sensor, exceeds a given threshold.

The controller allows for remote access and administration via standard IP-based connections established through POTS dialup or direct connection to wired access points, but is not dependent on those connections for operation. In other words, the controller will remain locally operable in the event that "outside" communication is unavailable for whatever reason. Even if a communication link to the outside is unavailable, the controller will continue to monitor and control its associated devices. This functionality is especially important in rural areas where weather conditions make communications services unreliable when the functionality is needed most. The standardized use of such a device would likely result in decreased maintenance costs, improved reliability, and greater flexibility in implementation when compared with "one-of-a-kind" deployments.

The purpose of this document is to present a summary of system testing and evaluation, including a summary of the original plans for system testing and evaluation. Testing and evaluation were limited in this Phase to in-lab testing at the Western Transportation Institute and in-lab testing and evaluation at Caltrans District 2. Testing was conducted at WTI throughout the project, and was integrated into the development process. Testing and evaluation was conducted at District 2 in the latter portion of 2011 and has continued through the end of the project. Testing of the Satellite Operations Center Command System (SOCCS) ASWC interface was conducted at Caltrans District 2 starting in mid-2012 and continued through the end of the project. The Phase 1 ASWC continues operating in its deployment in the field at Spring Garden at the time of this writing.

In general, the system has passed the prescribed in-lab tests.

For an introduction to the project see the companion document, Automated Safety Warning Controller System Concept and Requirements Specification (1).

# 1. TESTING

## 1.1. Testing Plan

Testing of the Automated Safety Warning Controller (ASWC) was done in four phases, continuous testing during development, in-lab testing of the deployment system at WTI, in-lab testing of the deployment system in the Caltrans District 2 lab, and deployment testing in the pilot location in California.

Development testing consisted mostly of unit tests and integration tests. With the refactoring of code modules from Phase 1 to Phase 2 it was necessary to make sure no inconsistencies were introduced. Field element modules were retested with all kinds of data (for the field elements that allow manipulation of data) to verify that they return and store the correct values. Integration tests were used to verify that each module communicated properly with the main system and with all other modules.

As modules completed unit testing they were incorporated into the in-lab deployment system testing. The deployment system has been running since Phase 1 and focuses on the operation of the system as a whole, verifying that the whole system works consistently and reliably. Testing involved end-to-end tests with a variety of simulated data, verifying that when a set of data should result in a warning message, it did, and when a set of data should not result in a warning message, it didn't. The SOCCS ASWC interface tested end to end functionality from client through the communications module to data sources and back. Communications errors were introduced to test error handling and recovery.

The long term reliability of the system continues to be tested. Devices are left running for long periods of time, with varying sets of data. System status and log files will be checked periodically for errors, and if errors occur it should be verified that the system can recover reliably from an error. Occasionally an error, such as loss of power or connectivity, may be deliberately generated to test the fault tolerance of the system. The system was also stress tested using more processes than are expected to be used in actual operation to assure adequate performance and reliability. Attention was paid to the storage and archiving of data and long term system memory usage.

Testing in the Caltrans District 2 lab is similar to the WTI lab testing; stress testing the system with data and scripts created by the District 2 ITS engineer(s). In addition the SOCCS ASWC interface was tested by TMC personnel.

Testing in the pilot location will consist primarily of long term reliability tests, with real data and possibly with real errors. This test will be similar to the in-lab testing of the deployment system in that it will be left for long periods with little interaction except to verify occasionally that the system is operating properly. This test is intended to show that the Controller system functions properly in real-world situations.

## 1.2. Development System Testing - In Lab w/ Simulated Systems

The Phase 1 pilot system has been running continuously since mid-August 2009 with few problems so the main focus of testing during Phase 2 was with modules that have changed during this phase of development and new modules such as the SOCCS interface and associated communications module. Each of these modules went through several tests to verify proper operation and predictable results based on its requirements and design. There were two classes of tests through which each field element went: unit tests to verify proper operation of the module as a standalone unit, and integration tests to verify that each module functioned properly as part of the system as a whole.

### 1.2.1. Unit Tests

Unit tests tested the basic operation of the module by itself. Input field elements such as RWIS were tested to verify the data received, while output field elements like CMS were tested to verify the data sent. The protocol handlers for the various input and output field elements were unchanged from Phase 1 and required no additional standalone testing. The CMS module was tested by inputting a specific message to the module, and using the SOCCS CMS software provided by Caltrans to verify the message that was actually put on the sign. The CMS logic for integrating TMC and ASWC messages on separate sign pages was tested.

The communications module developed to support the functions of the Satellite Operations Center Command System (SOCCS) ASWC interface were tested for correctness and completeness. The SOCCS ASWC interface was tested for usability and appearance.

Tests were run to ensure accurate saving and reading of data. The use of csv files facilitated this testing as known data can be input and the csv files manually inspected to verify the saved values. Phase 2 implemented the ability to specify the order of fields in the csv files and this functionality was tested. Attention was paid to the auto archiving functions as these were identified as problematic in Phase 1.

Unit testing also included evaluation of error handling in the module. The testing and development lab allowed for fabrication of error conditions, such as invalid data or loss of communication with the field element. Every field element module was tested for proper handling, recovery and logging of error conditions.

For the most part the unit tests entailed running a single test through the module and analyzing the results to assure the predicted outcome was achieved. Input data and starting conditions could be varied to test different scenarios. During these unit tests various bugs were found, fixed, and re-tested to verify proper operation. During these tests some of the issues found and fixed included the following.

- A check was added to verify that the order of fields in a data file hasn't changed; if it has then a new file is created.
- The necessity of having a sensorlist for every field element in FieldElement.ini was removed, it is only required for field elements that use that list.
- The CMS module was fixed to only open and close a connection to the CMS once per run interval, instead of two or three times.
- Many test cases were run through the CMS module to find logic bugs and assure proper message placement.

### 1.2.2. Integration Tests

The rest of the Controller system is designed to allow all the separate modules work together: passing data, providing support APIs, and providing a management interface for users. The integration tests are intended to verify the system works as a whole: that all the alert scripts, field element modules, and management interfaces get started and work together in the system.

In addition, some modules, such as the ThreadMonitor, FrontInterface, and the SOCCS server modules, are intended to work in conjunction with other modules and therefore couldn't be tested alone. Those modules could only be tested as part of the integration tests. The ThreadMonitor and FrontInterface modules provide the user interface portion of the system, ThreadMonitor provides the command line interface over SSH and FrontInterface provides the button and LCD interface on the front of the device. They were simple to test in that every command that could be issued was well defined; we simply had to execute each command in order.

The main test among the integration tests was the end-to-end data flow test. The fundamental operation of a Controller unit is that certain data values retrieved by a field element will result in a warning message being placed on the CMS. Numerous cycles of testing were performed by simulating a variety of data values on all simulated sensors, and making sure that a message is placed on the sign when the sensor values fall within a defined range.

### 1.2.3. Other Tests

The CMS module in particular needed a wider array of tests. The CMS module is the only one that could be thought of as "intelligent" in that it has specific logic that governs its operation. For a thorough description of the CMS decision making logic see the System Development Summary (2). The CMS module had some significant changes during this phase to allow a TMC message and an ASWC message to reside on a CMS at the same time. The SOCCS CMS software was used to verify the message on the sign. We used a variety of alert scripts combined with TMC messages added and cleared using the SOCCS ASWC interface to test the CMS module's handling of switching from a single page ASWC or TMC message to a two page message with both TMC and ASWC messages and back again. Conditions were created to simulate alert messages with various priorities; unknown or TMC messages on the sign; and various message expiration conditions to verify that all logical paths through the CMS module were tested and resulted in predictable outcomes.

Stress tests were conducted that involved running a large number of alert scripts and varying run intervals of various modules to test for potential conflicts and deadlock conditions.

Reliability tests involved leaving one device running for long periods of time with varied values for each sensor. Logs and status were checked regularly for error conditions. This method of testing was very close to the actual deployment scenario for the device, and the device proved reliable in this test.

In addition to the long term reliability tests, several artificial errors were simulated by removing connectivity to one or more field elements or injecting invalid data where it applied. The system was designed to acknowledge errors when they occur but to be relatively unaffected by them, and these tests verified that when an error occurred the system would keep running and would continue regular operations when the error condition stopped.

Tests were also run to ensure that in the event of an unexpected power cycle the system would boot up and return to a predictable and stable state.

Some examples of types of issues found during the integration testing are:

- Occasionally TMC messages placed on the CMS were clearing or getting garbled without reason.
- Efficiency improvements were made to the data module to keep data files in memory so it will not have to be read each time historical data was requested.
- Error handling routines were refactored to improve memory consumption.

## 1.3. Deployment System Testing - In Lab w/ Simulated Systems

A system was sent to Caltrans for testing in a deployment situation. Initially, one system was set up in a Caltrans lab with simulated field elements to test against. Ken Beals of Caltrans District 2 conducted a battery of tests on the deployment device before installing it at the pilot location. In addition, a deployment device was kept at WTI which was kept in parity with the devices that were sent to Caltrans, the WTI device also underwent continual testing on the WTI lab network.

Testing the documentation for completeness and clarity was an important part of this testing. Although the Moxa units sent to Caltrans were pre-configured for the test location, part of the in-lab testing conducted by Ken Beals was to go through the system in conjunction with the manual. Modifications were made and additional information added to the documentation as a result of this testing.

Some examples of types of issues found and subsequently fixed during the Caltrans lab testing are:

- The mishandling of empty fields in RWIS data file that would cause no data to be read.
- The CMS module had a problem where it would not correctly recognize a blank sign.
- The communications retries for the field element modules was happening to fast to be effective.
- A problem was found with the SOCCS ASWC client timing out before a dialup connection with the ASWC could be established.
- The initial problem with MOXA handling of compact flash cards was identified.
- TMC messages placed by the SOCCS ASWC interface were sometimes getting scrambled when viewed on CMS using SOCCS CMS.

## 1.4. Pilot testing at Caltrans site

Various delays resulted in the lab testing not completing soon enough prior to the bad weather season for the Phase 2 version of the ASWC to replace the Phase 1 system at Spring Garden. The pilot testing will be completed at a later date.

# 2. EVALUATION

## 2.1. Evaluation Plan

### 2.1.1. Technical Performance

The technical performance evaluation shows that a Controller system performs as specified in the requirements. It demonstrates that when a field element sensor reports a status that is deemed hazardous a warning message is placed on an appropriate sign to which the system is connected.

### 2.1.2. Reliability

Evaluation of reliability should show that the system will continue to operate unsupervised and without failure over long periods of time. The evaluation should show possible points of failure in the device operation and how the system will keep running in spite of those possible failures. Reliability evaluation should also involve a practical test of the system in the long term to show that the system is reliable in real situations.

### 2.1.3. Usability

Usability evaluation should show that the system provides the necessary commands and functions that a user needs to maintain a Controller system. The commands should also be easy to use and their use should be well defined and understandable. As the Controller system was designed to be used in various configurations as configured by field technicians, part of usability evaluation should consider the ease of setting up a Controller system for a particular field site. Usability evaluation will consist of gathering comments from qualified users of the system that did not play a part in the development of the system.

### 2.1.4. Maintainability

Maintainability evaluation will consist of an enumeration of the duties and activities that will be required to keep the system in a functional state over long term deployment. Also, the time and effort required to perform those duties and activities should be documented, assuming a well-documented step-by-step procedure.

### 2.1.5. Security

As with any complex system, a complete and exhaustive list of the possible security vulnerabilities in the Controller system will not likely be possible. It is assumed that the Controller will lie on a secured network, minimizing the possibility of a direct attack against the device; however, the designers of the system should still take every precaution they can make to guard against such attacks. The evaluation should show the total attack surface (virtual TCP and UDP ports open to the network and the attributes of those ports). All software running on the device that can receive network traffic should be documented with software version, protocol version (where applicable), and justification of why the software should be running on the device.

## 2.2. Technical Performance Evaluation

After development completion the system was thoroughly tested in-lab. Testing for evaluation purposes consisted of end-to-end testing, i.e. simulating sensor values in the field elements and verifying the system produces the expected result. Alert scripts based on best practice algorithms were written and installed on a Controller device, and the pertinent field elements were set to provide a range of data both inside and outside the alert thresholds. It was verified that the system did indeed produce a warning message on the CMS when the field element yielded a value inside the threshold, and no message was placed on the sign if the field element yielded a value outside the threshold. Independent testing by Caltrans in a District 2 lab yielded the same satisfactory results.

The Phase 1 system has been installed at a pilot location in Caltrans District 2: Spring Garden. The pilot testing consisted of an ice warning script that examined the values received from six surface status sensors attached to an RWIS. The system has been running unsupervised since mid-August 2009, and has shown success at deployment in the field with real data. Following are snippets from relevant data files from the Phase 1 system running at the Spring Garden pilot site followed by similar snippets from the Phase 2 system running in the Caltrans District 2 EE lab. Included is an explanatory timeline showing the success of the system at recognizing ice watch data and posting an icy curve warning to the CMS. This also demonstrates the ability to recreate events based on the data files.

### 2.2.1. Spring Garden data

#### 2.2.1.1. RWIS.csv

Note: due to the line length the field description line is left out. The data values, underlined below and pertinent to this discussion, are Timestamp( in UTC) and Surface status 1-6

2013-02-24 00:27:30,37.58,0.25,285,5.15,9.18,5.54,1000001,13,0,0,5,0,0,1,3,4,3,4,4,,31.82,36.68,33.8,33.08,32.0

2013-02-24 00:28:31,37.76,0.25,295,4.26,9.18,5.54,1000001,13,0,0,5,0,0,1,3,4,3,4,4,,31.82,36.68,33.8,33.08,32.0

2013-02-24 00:29:32,37.94,0.25,295,3.58,9.18,5.54,1000001,13,0,0,5,0,0,1,3,4,2,4,8,,31.82,36.68,,32.9,31.82

2013-02-24 00:30:33,38.12,0.25,270,2.69,9.18,5.54,1000001,13,0,0,5,0,0,1,3,4,2,3,8,,31.82,36.5,,32.9,31.82

…

2013-02-24 15:48:29,33.44,0.37,145,9.41,16.58,10.22,1000001,2,0,0,0,0,0,1,3,3,8,3,3,,28.22,30.38,29.3,29.3,29.84

2013-02-24 15:49:30,33.44,0.37,145,10.3,14.56,10.22,1000001,2,0,0,0,0,0,1,3,3,8,3,3,,28.4,30.38,29.3,29.48,30.02

2013-02-24 15:50:31,33.62,0.36,135,11.2,14.56,10.04,1000001,2,0,0,0,0,0,1,3,3,3,3,3,,28.58,30.74,29.66,29.66,30.2

2013-02-24 15:51:33,33.62,0.36,135,10.75,14.56,10.04,1000001,2,0,0,0,0,0,1,3,3,3,3,3,,28.76,30.92,30.02,29.84,30.38

#### 2.2.1.2. CMS.log

Note: timestamps in log files are shown in local time (Pacific time in this case).

2013-02-23 16:28:33 - icewarning - INFO - No Icy Road Warning

2013-02-23 16:28:51 - CMSWest - INFO - Sign is blank, no new messages

2013-02-23 16:29:20 - CMSEast - INFO - Sign is blank, no new messages

2013-02-23 16:29:33 - icewarning - INFO - Surface sensor #1 has invalid status or temp

2013-02-23 16:29:33 - icewarning - INFO - Surface sensor #4 has invalid status or temp

2013-02-23 16:29:33 - icewarning - INFO - Icy Road Warning: condition is Ice Watch

2013-02-23 16:29:33 - icewarning - INFO - Surface status: 1, 3, 4, 2, 4, 8

2013-02-23 16:29:33 - icewarning - INFO - Surface temps: 212.2, 31.8, 36.7, 212.2, 32.9, 31.8

2013-02-23 16:29:33 - icewarning - INFO - Message on East CMS: CAUTION ICY ROAD

2013-02-23 16:29:33 - icewarning - INFO - Message on West CMS: CAUTION ICY ROAD

2013-02-23 16:29:53 - CMSWest - INFO - Sign Message Blank. Placing New CMS Message: CAUTION ICY ROAD

2013-02-23 16:30:22 - CMSEast - INFO - Sign Message Blank. Placing New CMS Message: CAUTION ICY ROAD

…

2013-02-24 07:49:48 - icewarning - INFO - Icy Road Warning: condition is Ice Watch

2013-02-24 07:49:48 - icewarning - INFO - Surface status: 1, 3, 3, 8, 3, 3

2013-02-24 07:49:48 - icewarning - INFO - Surface temps: 212.2, 28.4, 30.4, 29.3, 29.5, 30.0

2013-02-24 07:49:48 - icewarning - INFO - Message on East CMS: CAUTION ICY ROAD

2013-02-24 07:49:48 - icewarning - INFO - Message on West CMS: CAUTION ICY ROAD

2013-02-24 07:50:14 - CMSEast - INFO - New Message and Sign Message Are the Same. Leaving Message

2013-02-24 07:50:46 - CMSWest - INFO - New Message and Sign Message Are the Same. Leaving Message

2013-02-24 07:50:49 - icewarning - INFO - Surface sensor #1 has invalid status or temp

2013-02-24 07:50:49 - icewarning - INFO - No Icy Road Warning

2013-02-24 07:51:16 - CMSEast - INFO - New Message and Sign Message Are the Same. Leaving Message

2013-02-24 07:51:48 - CMSWest - INFO - New Message and Sign Message Are the Same. Leaving Message

…

2013-02-24 08:10:53 - CMSEast - INFO - Put Blank Message on CMS

2013-02-24 08:10:53 - icewarning - INFO - Surface sensor #1 has invalid status or temp

2013-02-24 08:10:54 - icewarning - INFO - No Icy Road Warning

2013-02-24 08:11:25 - CMSWest - INFO - Put Blank Message on CMS

2013-02-24 08:11:53 - CMSEast - INFO - Sign is blank, no new messages

#### 2.2.1.3. CMSWest.csv

Note: due to the line length the field description line is left out. The data values, highlighted in red and pertinent to this discussion, are Time( in UTC) when the message was generated by the alert script and Time(in UTC) when the message was placed on the sign,

2013-02-23 21:06:47, , , , , , ,1,1,0,0,8,0,2013-02-23 21:06:47

2013-02-24 00:29:33,CAUTION,ICY ROAD,,,,,2,1,1200,10,1,10,2013-02-24 00:29:53

2013-02-24 00:30:34,CAUTION,ICY ROAD,,,,,2,1,1200,10,1,10,2013-02-24 00:50:29

…

2013-02-24 15:48:48,CAUTION,ICY ROAD,,,,,2,1,1200,10,1,10,2013-02-24 16:09:21

2013-02-24 15:49:48,CAUTION,ICY ROAD,,,,,2,1,1200,10,1,10,2013-02-24 16:10:23

2013-02-24 16:11:25, , , , , , ,1,1,0,0,8,0,2013-02-24 16:11:25

#### 2.2.1.4. Timeline

The following timeline attempts to explain the actions of Controller based on the data above. Note that for clarity times are shown as local time with the UTC in parenthesis:

- From RWIS.csv:

  16:28:31 (00:28:31 UTC): RWIS reads all surface status sensors as either 3 (dry) or 4 (trace)
- From CMS.log:

  16:28:51 (00:28:51 UTC): icewarning script reads RWIS data, reports No Icy Road Warning
- From RWIS.csv:

  16:29:32 (00:29:32 UTC): RWIS reads all surface status sensors, SurfaceStatus6 is 8 (icewatch)
- From CMS.log:

  16:29:33 (00:29:33 UTC): icewarning script reads RWIS data, generates Icy Road Warning
- From CMS.csv

  16:29:33 (00:29:33 UTC): CMSWest places Icy Curve message on sign

…

- From RWIS.csv:

  07:49:30 (15:49:30 UTC): RWIS reads all surface status sensors, SufaceStatus4 is 8 (icewatch)
- From CMS.log:

  07:49:48 (15:49:48 UTC): icewarning script reads RWIS data, generates Icy Road Warning
- From RWIS.csv:

  07:50:31 (15:50:31 UTC): RWIS reads all surface status sensors as either 3 (dry)
- From CMS.log:

  07:50:49 (15:50:49 UTC): icewarning script reads RWIS data, reports No Icy Road

…

- From CMS.csv

  08:10:23 (16:10:23 UTC): last Icy Road message is placed on sign

  08:11:25 (16:11:25 UTC): last Icy Curve message expires (20 minutes after it was created)

  08:11:25 (16:11:25 UTC): CMSEast places blank message on sign

This example demonstrates a few of the nuances of the system. One is the relationship between the alert script run interval and the alert message expiration. If the alert script run interval is short but the message expiration is long the message pool will grow fairly large. Ultimately this does not affect the end results as the CMS module recognizes if the new message generated is the same as the one already on the sign, and only updates the expiration time. The message will still expire 'expiration minutes' (defined in the alert script) after it was generated by the script regardless of how long it has actually been on the sign. This can be confusing when looking at the data files: in the second to last line of CMS.csv is a message that was generated at 15:49:48 (first timestamp) but not put on the sign until 16:10:23 (second timestamp). The message was then blanked out less than a minute later as it had expired (20 minutes after it was created). The net effect is that there was an “Icy Curves” message on the sign from the time the first one was created by the alert script until 20 minutes after the last one was created: as was desired at this location.

One other nuance: there is no configuration element for the minimum time that a message is on a sign. Rather, this is controlled by the run interval of CMS module. This can become an issue if a high priority message is generated while a low priority message is on the CMS. The CMS module run interval will define the minimum time the low priority message is on the CMS prior to being replaced. This will keep the sign from “blinking” one message up and then replacing it. Under normal operation this should be sufficient; however it does point out that on a Controller running multiple alert scripts, timing needs to be carefully considered. Note that if the low priority message’s expiration time has not passed when the high priority message has expired it will be placed back on the CMS until its expiration time has been reached. This is demonstrated in the next example.

### 2.2.2. ASWC Phase 2 running in District 2 lab

The following example utilizes data from the Phase 2 ASWC that is running in the Caltrans District 2 EE lab. This example uses data generated to test the system and its alert scripts and messages. In this example two alerts scripts are running, one running a wind warning and one running an ice warning. The ice warning has a higher priority and this example shows how the higher priority message takes precedence over the lower priority (wind warning) message.

#### 2.2.2.1. RWIS.csv

Note: due to the line length the field description line is left out. The data values, underlined below and pertinent to this discussion, are Timestamp( in UTC), Average Wind Gust, and Surface status 1-6, and Surface temperature

2013-02-27 23:21:10,73.4,0.5,361,0.0,0.0,53.6,24140,150,1006,503,1912,250,113,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:22:11,73.4,0.5,361,0.0,0.0,53.6,24140,150,1006,503,1912,250,113,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:23:11,73.4,0.5,361,31.36,33.6,53.6,24140,150,1004,503,1905,253,111,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:24:12,73.4,0.5,361,31.36,33.6,53.6,24140,150,1004,503,1905,253,111,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:25:12,73.4,0.5,361,31.36,33.6,53.6,24140,150,999,503,1900,253,106,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:26:13,73.4,0.5,361,31.36,33.6,53.6,24140,150,999,503,1900,253,106,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:27:13,73.4,0.5,361,31.36,33.6,53.6,24140,150,994,503,1900,253,101,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

2013-02-27 23:28:14,73.4,0.5,361,31.36,33.6,53.6,24140,150,994,503,1900,253,101,3,3,2,3,3,3,73.4,64.4,140.0,82.4,91.4,55.4

…

2013-02-27 23:51:48,73.4,0.5,361,31.36,33.6,53.6,24140,150,910,468,1902,235,10,4,4,2,4,4,4,64.4,55.4,140.0,73.4,82.4,46.4

2013-02-27 23:52:49,73.4,0.5,361,31.36,33.6,53.6,24140,150,910,468,1902,235,10,4,4,2,4,4,4,64.4,55.4,140.0,73.4,82.4,46.4

2013-02-27 23:53:49,73.4,0.5,361,31.36,33.6,53.6,24140,150,903,465,1905,232,7,5,5,2,5,5,5,50.0,41.0,140.0,59.0,68.0,32.0

2013-02-27 23:54:50,73.4,0.5,361,31.36,33.6,53.6,24140,150,903,465,1905,232,7,5,5,2,5,5,5,50.0,41.0,140.0,59.0,68.0,32.0

2013-02-27 23:55:50,73.4,0.5,361,31.36,33.6,53.6,24140,150,898,465,1907,232,12,5,5,2,5,5,5,50.0,41.0,140.0,59.0,68.0,32.0

...

2013-02-28 00:11:59,73.4,0.5,361,31.36,33.6,53.6,24140,150,938,432,1945,215,53,5,5,2,5,5,5,50.0,41.0,140.0,59.0,68.0,32.0

2013-02-28 00:13:00,73.4,0.5,361,31.36,33.6,53.6,24140,150,948,442,1955,215,63,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:14:00,73.4,0.5,361,31.36,33.6,53.6,24140,150,948,442,1955,215,63,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:15:01,73.4,0.5,361,31.36,33.6,53.6,24140,150,958,452,1965,215,73,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:16:01,73.4,0.5,361,31.36,33.6,53.6,24140,150,958,452,1965,215,73,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:17:02,73.4,0.5,361,31.36,33.6,53.6,24140,150,968,462,1975,215,83,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:18:03,73.4,0.5,361,31.36,33.6,53.6,24140,150,968,462,1975,215,83,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

2013-02-28 00:19:03,73.4,0.5,361,31.36,33.6,53.6,24140,150,979,473,1986,220,93,6,6,2,6,6,6,30.2,21.2,140.0,39.2,48.2,12.2

#### 2.2.2.2. CMS.log

Note: timestamps in log files are shown in local time (Pacific time in this case).

2013-02-27 15:23:29 - CMSWest - INFO - No ASWSC message on sign, no new messages

2013-02-27 15:23:29 - CMSEast - INFO - No ASWSC message on sign, no new messages

2013-02-27 15:24:02 - windwarning - INFO - Wind Advisory, values: 0.0 33.6

2013-02-27 15:24:02 - icewarning - INFO - No Icy Road Warning

2013-02-27 15:24:31 - CMSWest - INFO - Sign Message Blank. Placing New CMS Message: WIND ADVISORY GUSTS TO 33 MPH

2013-02-27 15:24:31 - CMSEast - INFO - Sign Message Blank. Placing New CMS Message: WIND ADVISORY GUSTS TO 33 MPH

2013-02-27 15:25:03 - windwarning - INFO - Wind Advisory, values: 0.0 33.6

...

2013-02-27 15:54:22 - CMSEast - INFO - New Message and Sign Message Are the Same. Leaving Message

2013-02-27 15:54:22 - CMSWest - INFO - New Message and Sign Message Are the Same. Leaving Message

2013-02-27 15:54:43 - windwarning - INFO - Wind Advisory, values: 31.36 33.6

2013-02-27 15:54:43 - icewarning - INFO - Icy Road Warning: condition is Snow

2013-02-27 15:54:43 - icewarning - INFO - Surface temps: 50.0, 41.0, 212.2, 59.0, 68.0, 32.0

2013-02-27 15:54:43 - icewarning - INFO - Surface status: 5, 5, 2, 5, 5, 5

2013-02-27 15:55:25 - CMSEast - INFO - Sign Message Expired. Placing New CMS Message: CAUTION ICY ROAD

2013-02-27 15:55:25 - CMSWest - INFO - Sign Message Expired. Placing New CMS Message: CAUTION ICY ROAD

...

2013-02-27 16:18:49 - windwarning - INFO - Wind Advisory, values: 31.36 33.6

2013-02-27 16:18:49 - icewarning - INFO - No Icy Road Warning

2013-02-27 16:19:08 - CMSEast - INFO - Sign Message Expired. Placing New CMS Message: WIND ADVISORY GUSTS TO 33 MPH

2013-02-27 16:19:08 - CMSWest - INFO - Sign Message Expired. Placing New CMS Message: WIND ADVISORY GUSTS TO 33 MPH

2013-02-27 16:19:49 - windwarning - INFO - Wind Advisory, values: 31.36 33.6

...

#### 2.2.2.3. CMSWest.csv

Note: due to the line length the field description line is left out. The data values, highlighted in red and pertinent to this discussion, are Time( in UTC) when the message was generated by the alert script and Time(in UTC) when the message was placed on the sign,

2013-02-27 22:59:29, , , , , , ,1,1,0,0,8,0,2013-02-27 22:59:29

2013-02-27 23:24:02,WIND ADVISORY,GUSTS TO 33 MPH,,,,,1,1,300,6,0,50,2013-02-27 23:24:31

2013-02-27 23:25:03,WIND ADVISORY,GUSTS TO 33 MPH,,,,,1,1,300,6,0,50,2013-02-27 23:29:40

...

2013-02-27 23:49:41,WIND ADVISORY,GUSTS TO 33 MPH,,,,,1,1,300,6,0,50,2013-02-27 23:54:22

2013-02-27 23:54:43,CAUTION,ICY ROAD,,,,,2,2,300,9,1,10,2013-02-27 23:55:25

2013-02-27 23:51:42,WIND ADVISORY,GUSTS TO 33 MPH,,,,,1,1,300,6,0,50,-1

...

2013-02-28 00:11:47,CAUTION,ICY ROAD,,,,,2,2,300,9,1,10,2013-02-28 00:18:04

2013-02-28 00:11:47,CAUTION,ICY ROAD,,,,,2,2,300,9,1,10,-1

2013-02-28 00:12:47,WIND ADVISORY,GUSTS TO 33 MPH,,,,,1,1,300,6,0,50,2013-02-28 00:19:08

...

#### 2.2.2.4. Timeline

The following timeline attempts to explain the actions of Controller based on the data above. Note that for clarity times are shown as local time with the UTC in parenthesis:

- From RWIS.csv:

15:22:11 (23:22:11 UTC): RWIS reads all surface status sensors as either 3 (dry) or 2 (error) and max wind gust as 0.0

- From CMS.log:

15:24:02 (23:24:02 UTC): icewarning script reads RWIS data, reports No Icy Road Warning

- From RWIS.csv:

15:23:11 (23:23:11 UTC): RWIS reads an max wind gust of 33.6 mph

- From CMS.log:

  15:24:02 (23:24:02 UTC): windwarning script reads RWIS data, generates Wind Advisory

- From CMSWest.csv

  15:24:31 (23:24:31 UTC): CMSWest places Wind Advisory message on sign

…

- From RWIS.csv:

  15:53:49 (23:53:49 UTC): RWIS reads all surface status sensors as 5 (wet) and surface temperature of 32.0, max wind gust is 33.6

- From CMS.log:

  15:54:43 (23:54:43 UTC): icewarning script reads RWIS data, generates Icy Road Warning

- From CMSWest.csv

  15:55:25 (23:55:25 UTC): CMSWest places Icy Road message on sign

- From RWIS.csv:

  16:13:00 (00:13:00 UTC): RWIS reads all surface status sensors as either 6 (chemically wet) or 2 (error) and max wind gust as 33.6

- From CMS.log:

  16:18:49 (00:18:49 UTC): windwarning script reads RWIS data, reports Wind Advisory

  16:18:49 (00:18:49 UTC): icewarning script reads RWIS data, reports No Icy Road

…

- From CMS.csv

  16:18:04 (00:18:04 UTC): last Icy Road message is placed on sign

  16:19:08 (00:19:08 UTC): last Icy Curve message expires and is replaced by Wind Advisory

While this example shows the same relationship between the log files as in the analysis of the system running in Spring Garden it also shows how multiple scripts interact. Initially there was a wind gust warning triggered by the 33.6 mph gust reading in from the RWIS ((23:23:11 UTC). This warning was replaced by an icy road warning that is triggered by the combination of wet surface condition and a surface temperature below 32.5F (23:53:49 UTC), note that the wind gust condition still exists (wind gust of 33.6). The ice warning script gives the ice warning a priority of 9 while the wind warning has a priority of 6.

## 2.3. Reliability Evaluation

Reliability testing is intended to show that the Controller system will run unsupervised for long periods of time without failure. The system was tested in-lab with simulated field elements both at WTI and at a Caltrans District 2 lab. Both in-lab tests utilized long term test runs of between a week and multiple months with the extent of supervision being occasionally checking the logs for errors. These tests have all proven satisfactory for the reliability of the system.

One unit was installed in Spring Garden connected to an RWIS and two CMSs in mid-August 2009. System log files recently sent by Jeff Worthington, who took over for Ken Beals in overseeing the pilot installation, didn't show a single error, network related or otherwise.

The system was designed to catch errors and simply wait until the error condition (such as loss of network connectivity) is resolved. The module then runs again as if nothing had happened. With this design the system will continue to run even in the event of normally fatal errors.

## 2.4. Usability Evaluation

The usability evaluation for this this phase of the Controller project primarily will only take into account the changes made during this phase. This phase was only running in a test environment in the Caltrans District 2 EE lab so evaluation is limited. The SOCCS ASWC interface developed during this phase was evaluated briefly by Joseph Baltazar of the Caltrans District 2 TMC and his comments are presented below.

Q. Is the front panel interface adequate and user friendly for checking the system status?

*A. From email from Jeff Worthington Caltrans District 2, 11 March 14, 2013:*

*A critique on the LCD interface: Looks good. I have spent some time perusing the menus. This behaves as expected to me. I've interfaced with several devices on an LCD panel and this seems just as easy. Wouldn't change a thing. I especially like the countdown timer for setting the warning signs for a duration. (This after spending a little time at Fredonyer and having to do this the manual way.)*

Q. Do you feel comfortable setting up the ASWC?

*A. From email from Jeff Worthington Caltrans District 2,March 15, 2013:*

*One primary concern I want to focus on is my ability to install this. This should be my next direct action related to the ASWSC, putting the current version, from the lab, live into the field for the first time at Spring Garden.*

Comments from Joseph Baltazar include:

- *Overall, The system is highly useful: THANK YOU for allowing the TMC to offer the 1 page Message versus none at all!*
- *RWIS Details? What about RWIS & CMS name, county, route, post mile. All important details if we have multiple sites.*
    - The location details are included in the ASWC initialization files but are not used anywhere. These should be passed up to the SOCCS ASWC interface.
- *When selecting "Change Message on Sign", Add CMS Name to "Sign Message Chooser" window.*
- *How is the operator notified of feedback when the CMS fails to connect.*

## 2.5. Maintainability Evaluation

The system is designed to be largely autonomous; however, it also logs every action it takes and all the data it collects, which can lead to very large amounts of data. The primary maintenance procedure the Controller system requires is removing old data to make space for new data. With a 4 GB CompactFlash card the device should be able to hold over a year's worth of data, depending on the number of sensors being recorded and the frequency at which messages are put on the sign.

All maintenance activities that don't require hardware repair can be done remotely, and generally consists of uploading or downloading files over FTP or changing configuration files through SSH accompanied by a restart of the software. This includes revisions and bug fixes in the Controller software itself, the changed files can simply be uploaded and the software restarted.

Aside from installing new versions of Controller software or site specific changes, the system should require regular maintenance no more than monthly, although system status checks are recommended at least once a week. The main maintenance function would be to download archived log files and data files.

## 2.6. Security Evaluation

The security of a computer system is largely dependent on the small size of its attack surface. For the purposes of this document a system's attack surface is defined as the total number of applications waiting for a remote computer to initiate a network connection. The Moxa device ships with more of these applications enabled than are actually necessary for a Controller system to operate. Following is a summary of running services, the port number they listen to, the software and version that implements those services, and the recommendation concerning those services on a Controller system:

- ftp (TCP 21) WU-ftpd 2.6.1
    - Recommend turning off and using SFTP (which goes over the ssh protocol on port 22)

- ssh (TCP 22) OpenSSH 3.9p1 (protocol v. 1.99)
    - Recommend keeping service active, it is necessary for user and administrator access to the Controller system

- http (TCP 80) Apache 2.2.2
    - Recommend turning off unless using the web management interface. (Security note: this version of Apache is configured to run as root)

- portmap/rpcbind (TCP 111,1024,2049)
    - Recommend turning off, this service is not necessary for Controller

The Controller software is written so as to use the authentication mechanism already provided by Linux. This is done so that the Controller system doesn't enlarge the attack surface of the device, and so that all user authentication is performed by tested and proven Linux login procedures.

# 3. REFERENCES

1. Western Transportation Institute, September 2007, *Automated Safety Warning Controller System Concept and Requirements Specification.*
2. Western Transportation Institute, March 2013. *Automated Safety Warning Controller System Development Summary*.