

***Automated Safety Warning Controller
System Development 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

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data herein. The contents do not necessarily reflect the official views or policies of the California Department of Transportation or the Federal Highway Administration.

Alternative accessible formats of this document will be provided upon request. Persons with disabilities who need an alternative accessible format of this information, or who require some other reasonable accommodation to participate, should contact Western Transportation Institute, Montana State University, PO Box 174250, Bozeman, MT 59717-4250, telephone number 406-994-6114.

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

1. System Development Summary.....	1
1.1. General system development environment	1
1.2. Review of Phase 1 Comments and Code	1
1.3. New Development.....	2
1.3.1. Front panel interface development.....	2
1.3.2. Protocol Development and Testing.....	2
1.3.3. SOCCS Automated Safety Warning System Controller.....	3
1.3.4. CMS module changes	3
1.4. Automate Installation	5
1.5. Implement Watchdog Timer	5
2. Items Not completed	6
2.1. Loop detector.....	6
2.2. Flashing Beacon, EMS	6
2.3. MVDS	6
2.4. HAR – text to voice, etc.	6
2.5. Email notification.....	6
3. References.....	7
4. Appendix: Controller Front Panel Interface	8
5. Appendix: SOCCS Automated Controller Applet User Guide.....	11

LIST OF FIGURES

Figure 1: CMS Logic	4
Figure 2: CMS Check Message Logic	5
Figure 3: Controller Front Panel Interface Menu Hierarchy	8
Figure 4: Controller Applet Home Screen	11
Figure 5: Controller Menu Options.....	12
Figure 6: Controller Settings Tab	13
Figure 7: Controller Authentication.....	13
Figure 8: Controller Admin Screen Showing ASWC in an Inactive State	14
Figure 9: Controller Admin Screen with Input Field Elements Tab Selected	15
Figure 10: Controller Admin Screen with Alert Scripts Tab Selected	16
Figure 11: Controller Admin Screen with Output Field Elements Tab Selected	17
Figure 12: Controller Sign Message Chooser	18

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” (ASWC). The ASWC will interface with roadside devices such as sensors and signs. The ASWC 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 the development conducted in this second phase of the project.

The testing and development lab is housed in the Systems Lab at the Western Transportation Institute at Montana State University, and was set up in collaboration with Caltrans to provide a realistic representation of a field element network, as deployed by Caltrans District 2. Caltrans District 2 staff constructed and configured the lab prior to delivery to WTI. WTI subsequently integrated various components into the lab, particularly the ASWC.

The ASWC interfaces with the various field elements in the testing and development lab, via code deployed on the ASWC. The ASWC is an embedded Linux system, and the ASWC code is primarily written using the open source Python programming language. The system has been developed in a modular fashion, allowing for future expansion.

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

1. SYSTEM DEVELOPMENT SUMMARY

1.1. General system development environment

The Automated Safety Warning Controller (ASWC) system was designed to be modular for flexibility. Modules can be added or removed to change the functionality of the system, and it requires no more than a software restart. In addition there can be multiple instances of each module, such as two or more CMSs connected to the ASWC, which are added via additional sections in the configuration file. Configuration files (ending in .ini) are the interface through which all the configurable parameters of a module are set.

The ASWC software was written in Python for portability. Python is a scripting language requiring no compilation or native code. Because of the inherent portability of Python all development could be done on commodity office PCs and the software would run without modification on the ASWC hardware, despite the differences between the x86 architecture of the office PCs and the ARM architecture of the embedded device. This portability also allows the ASWC software to be run on many different devices and architectures. The modularity combined with the portable code means that in the future the ASWC software could be adapted to purposes completely different from the original scenarios of monitoring weather to change the message on a CMS. The use of Python also greatly simplified development with its built in modules. TCP/UDP/IP communications and logging facilities are available through the default Python installation.

Hardware used for both development and deployment in the previous phase was the Moxa model UC-7420 standalone computer that runs on an Intel XScale ARM processor clocked at 266 MHz. For this phase the application was modified to also be capable of running on a Moxa model DA-661 rack mount computer that runs on an Intel XScale ARM processor clocked at 533 MHz. The rack mount design is consistent with Caltrans roadside network equipment mounting. The UC-7420 has an 8x16 character LCM display and 5 programmable push buttons while the DA-661 has a 2x16 display and 4 push buttons.

1.2. Review of Phase 1 Comments and Code

At the start of this phase, results and comments from Phase I were compiled and documented in the Phase 2 Review Summary and Recommendations (2). The start of Phase 2 development consisted of a full review of Phase 1 code addressing issues found during Phase 1 and where necessary refactoring code to improve efficiency and reliability. Attention was paid to reuse of code modules and eliminating any current or potential memory problems. Other items identified from Phase 1 that were addressed include:

- The code for field element modules was refactored to move repeated code to a common, generic module.
- Improvements were made to the CMS module's handling of retries. It was noticed in Ken Beal's lab setting that when the CMS module failed to connect it didn't appear to retry. A configurable retry number and retry interval were added.
- Improvements were made to the data module's auto archiving functionality. The system was only invoking the auto archiving when data other than the most recent reading was read from a dataset, for instance to get average wind speed. Since the pilot site is running an ice warning alert script that uses the most recent sensor values the auto archiving

wasn't invoked and the working file was allowed to grow very large. The system for auto archiving was changed to check file size when reading or writing data and auto archive when necessary.

- Improvements were made to the data storage. The ability to specify the order data fields are stored in the data files was added to improve the readability of the files.
- Enhancements were made to the user authentication system. The Phase 1 version contained an "enable" password to allow a user of the command line interface to go from operator to supervisor level of access. The authentication system was re-written to allow any number of username/passwords with each assigned either an operator or supervisor level of access. A simple Python script was written to allow administration of user authentications.

1.3. New Development

1.3.1. Front panel interface development

The FrontInterface module provides monitoring functionality through the LCD and push buttons on the front of the device. Its purpose is to provide quick status checks of the system without plugging a laptop into the device. The previous version of the ASWC front panel interface was designed for an 8 x 16 display and 5 front panel push buttons, for this phase the code would be revised to effectively use the 2 x 16 display and 4 front panel push buttons found on the Moxa DA-661. Among the design considerations taken into account is the need for the system to be intuitive for the occasional user such as a technician doing periodic maintenance on a site and, considering the roadside nature of access to the system, to allow a technician to access needed information as easily as possible. To accomplish this goal a simple hierarchical menu system was designed and implemented to allow the user to navigate through the options to retrieve module status and variable values or put a test message on the CMS. The default display is a quick status showing "OK" if all configured scripts finished their last run successfully and "ERROR" otherwise. Menu selections are available for displaying individual thread status, displaying variable values, and running a sign test script to put a test message on a CMS. Variables values or thread status are displayed in a name over value format. Up and down arrows are used to set through name/value pairs. Selecting the sign test option will execute an alert script that will display a test message on a selected CMS. After selecting the sign test option the display will acknowledge that it is putting a message on the sign and then display a countdown timer of the approximate amount of time the message will remain on the sign. For more information on the front panel interface see Appendix (4)

1.3.2. Protocol Development and Testing

To facilitate the communications with ASWC from external programs, such as the SOCCS Automated Safety Warning System interface described below, a well-defined communications protocol needed to be developed. The development of the protocol started with the decision to make the protocol based on a client-server model with the ASWC responding to requests from external clients. The next step was to develop a complete description of the protocol used by SOCCS to configure and manage the ASWC. The protocol was then implemented as a module on the ASWC. All command/reply pairs were then tested for correctness. For the complete definition of the protocol see Controller Message Protocol Definition (3).

1.3.3. SOCCS Automated Safety Warning Controller

A major part of the development effort for this phase was to develop the SOCCS ASWC interface to allow for monitoring and control of a remote ASWC. The SOCCS interface was developed to run on a browser on a PC capable of running Java applets. The SOCCS ASWC client software is located on a server in the TMC and when accessed by a client browser, reads a configuration file on the TMC server and uses this to configure an initial list screen. The list screen will connect to the configured ASWC(s) at a configured interval, get a quick status and the current CMS message(s), and disconnect. The status and messages are displayed for each configured ASWC. The status of each ASWC is displayed as text (OK, ERROR, UNKNOWN) as well as a colored border. This initial display gives the quick view of each ASWC and their associated CMS(s). For more detailed information an administrative screen was implemented. Accessing the administrative screen requires a username/password and initiates an authenticated connection with the ASWC. This connection stays open as long as the administrative window remains open. A configurable update interval can be set to maintain dialup connections. The administrative screen allows the user to view detailed information about the modules running and their associated variables and values. Using the administrative screen TMC personnel can add a message to a CMS. The Phase 1 specifications called for TMC messages overriding ASWC messages until explicitly cleared. During testing in Caltrans district 2 a decision was made to allow the ASWC to use one page of a two page sign and the TMC message the other. This was implemented within the SOCCS ASWC administrative screen and required changes were made to the ASWC CMS module as outlined below.

1.3.4. CMS module changes

The CMS module is slightly different from the other field elements in that it requires a degree of intelligence in its operation. While the other field elements consist mainly of retrieving data and storing it, the CMS module requires some decision making ability for selecting a message from the queue and choosing whether or not to place it on a sign. The other part of the CMS module handles what most of the other field element modules do: communicating with the field element. The logic portion was protocol independent enough that the protocol was moved to another module, which can be interchanged for a different model or version of sign or sign controller.

The CMS logic module handles the decision making concerning what message should be placed on a sign, if any, and when to clear the sign. During the testing of the SOCCS ASWC interface, changes to the way the ASWC handles messages put on the sign by the TMC were requested and implemented. The requirements from the previous phase specified that the ASWC would only change the message on a CMS if the sign was blank or if the message on the sign was placed there by the ASWC. If a message was placed on the sign by the TMC, using SOCCS CMS, the ASWC would not place a warning message on the CMS until the TMC blanked the CMS. Under the new requirements the ASWC scripts would be able to place one page messages on the CMS and the second page would be available for the TMC to place a one page message using the SOCCS ASWC interface. The CMS module contains the logic that is at the heart of the ASWC so a significant amount of development and testing was necessary to accomplish these changes. A complete flowchart of the new CMS decision making logic is included in Figure 1 and Figure 2.

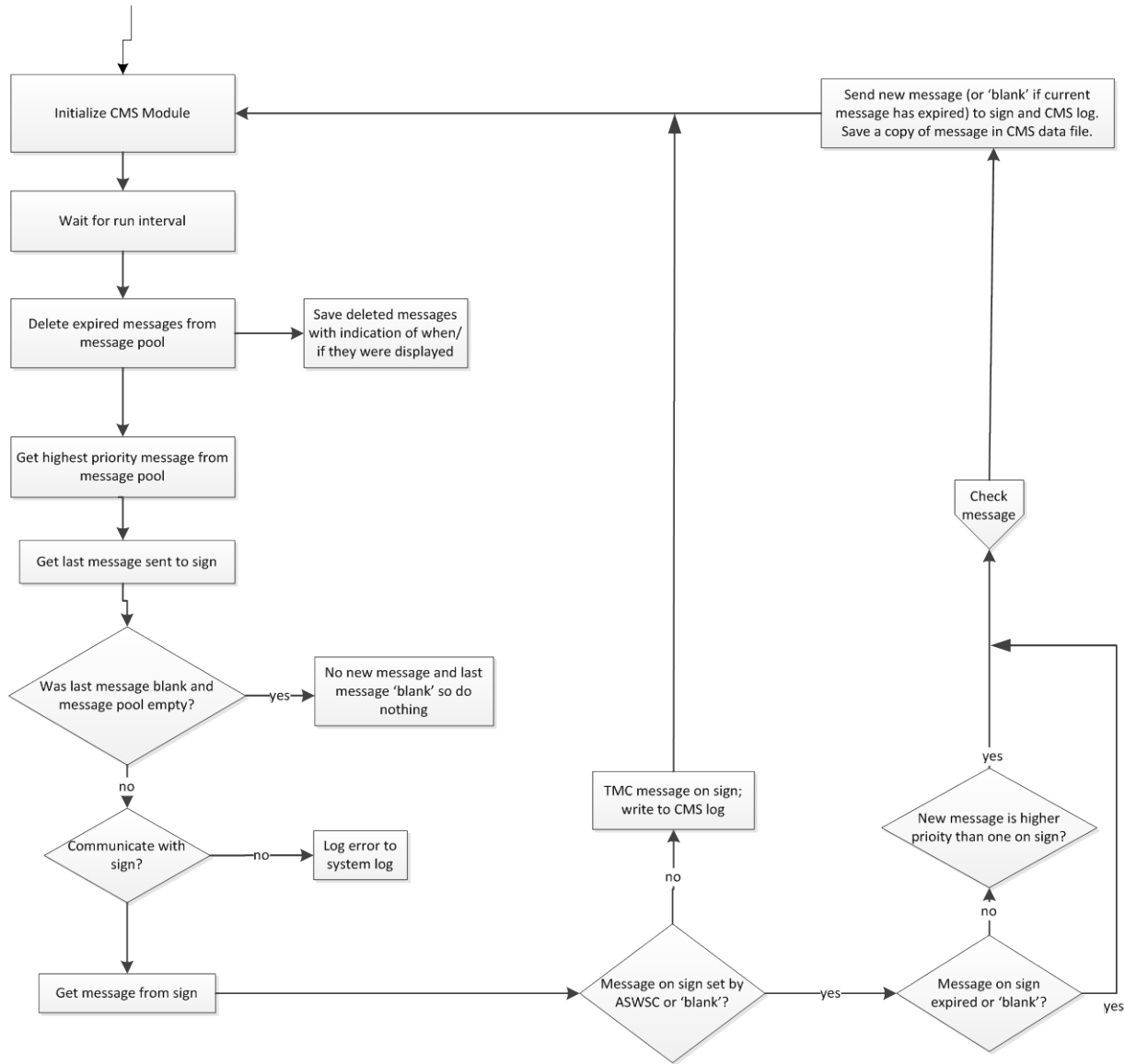


Figure 1: CMS Logic

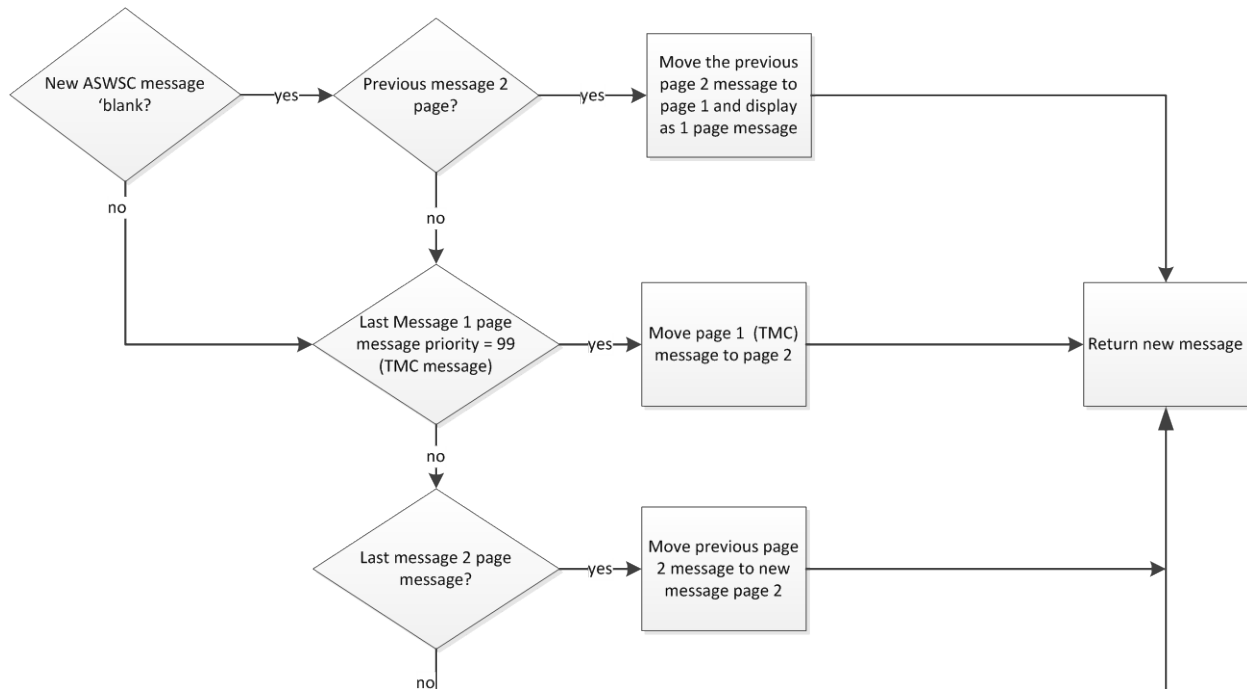


Figure 2: CMS Check Message Logic

1.4. Automate Installation

An installation script was created to streamline the installation of the ASWC on a new device. The single installation script will ask a series of questions then automate the creating of ASWC directories and moving the scripts onto the device.

1.5. Implement Watchdog Timer

A configuration option was added to implement the watchdog timer feature built in to the Moxa device. If the option is turned on the ASWC will send an “I’m alive” acknowledgement to the operating system every thirty seconds. If the “I’m alive” is not sent the Moxa will automatically reboot and restart the ASWC.

2. ITEMS NOT COMPLETED

The following items, deferred from Phase 1 were not completed as part of Phase 2.

2.1. Loop detector

A module for handling the Loop Detector protocol (on the 170 controller) was developed during Phase 1 and updated in Phase 2, but it was never fully tested, and associated, realistic alert scripts were not developed.

2.2. Flashing Beacon, Extinguishable Message Sign (EMS)

A protocol module was written to interact with a flashing beacon or EMS using a WebRelay™ device. A WebRelay™ device was not purchased under Phase 1 so this functionality was not fully tested. The hardware to fully test and implement the flashing Beacon and EMS modules was not purchased and tested in Phase 2.

2.3. Microwave Vehicle Detection System (MVDS)

Some preliminary work was done to handle the EIS X3 RTMS protocol, however full implementation was deferred until a suitable test site/application is found.

2.4. Highway Advisory Radio (HAR) – text to voice, etc...

This was deferred until a suitable test site and application is found.

2.5. Email notification

There was some discussion about implementing the ability of the ASWC to send an email message when an alert is generated or cleared. This functionality was not implemented during this phase. From an email from Ken Beals dated 23 February 2011:

An additional requirement which was not covered in this morning's meeting was the ability of the controller to send email messages when changing state. The connection to a simple smtp port 25 server would need to be configurable including server address, sender name, multiple recipients, etc. My initial take on this is that the "SendMail" function (or whatever it is named) would need to be a function usable in the alert script. The configuration info could be in an .ini file.

3. REFERENCES

1. Western Transportation Institute, September 2007, *Automated Safety Warning Controller System Concept and Requirements Specification*.
2. Western Transportation Institute, February 2011, *Automated Safety Warning Controller Phase 2 Review Summary and Recommendations*
3. Western Transportation Institute, November 2011, *Automated Safety Warning System Controller–Controller Message Protocol Definition Document*

4. APPENDIX: CONTROLLER FRONT PANEL INTERFACE

Menu Hierarchy:

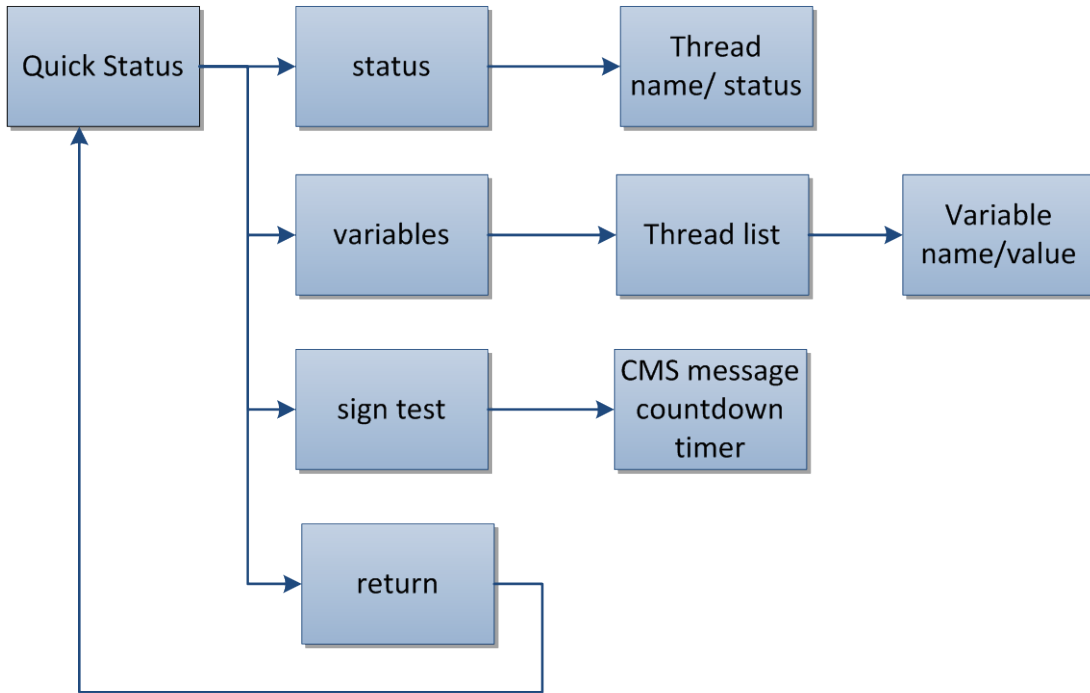


Figure 3: Controller Front Panel Interface Menu Hierarchy

Menu Detail

Quick Status

Quick Status is the default display. Controller starts up with Quick Status displayed and returns to it automatically after sitting idle for 5 minutes. Press **MENU** to escape to the main menu. Pressing any other button will manually update the display and turn the backlight on.

Example:

```
Controller OK
10 Dec 12 10:52
```

status

Used to see the status of the individual running threads to determine which one(s) is in error should the quick status show an error state. Use **↑** or **↓** to scroll through the list of threads.

Example:

```
RWIS      press ↓      Windwarning
OK                OK
```

variables

Use the variables command to view variables for each thread. The Variables command brings up a list of currently running threads, choosing a thread will bring up a list of variables used by that thread. Use **↑** or **↓** to select the desired thread. Then use **SEL** to view the variables and **↑** or **↓** to scroll through the configured variables.

Example:

```
>windwarning          highwindwarning
                      56
icewarning           press →      windwarning
                      41
```

sign test

Use to enter a high priority test message in the sign queue. Selecting this option executes an alert script used to test the Controller interaction with the sign. The display will show an approximate count down of how long the message will remain on the sign.

Example:

```
Test message on
sign for 15s
```

return

Use to manually return to the Quick Status display.

Navigating the Menu

The screen is 16 columns by 2 rows. There are three separate “Modes” for displaying information: Menu Mode, Info Mode, and Quick Status. In any mode the backlight turns off after 15 seconds, pressing any key will turn it back on.

Menu Mode

The screen has a cursor on the left side, any menu item that is longer than 15 characters will automatically scroll across the screen. The Up and Down buttons will move the cursor between items in the menu, press the SEL button to choose a menu item. In Menu Mode the Menu button returns you to the previous menu.

Info Mode

The screen enters Info Mode when displaying information at the end of a path through the menu. When showing the system status, or showing thread variables the screen is in Info Mode. There is no cursor in Info Mode and the only way to escape is to press the Menu button. The Up and Down buttons are used to scroll through the values pairs.

Quick Status

The Controller application begins in this mode. It is an idle state that updates every 1 minute. Pressing the Up and Down buttons will force a manual update.

5. APPENDIX: SOCCS AUTOMATED CONTROLLER APPLLET USER GUIDE

The Automated Safety Warning Controller (ASWC) is a roadside system that monitors road and weather conditions and updates driver warning systems accordingly. The ASWC system informs drivers through changeable message signs or flashing beacons. The Satellite Operations Center Command System (SOCCS) ASWC interface allows TMC operators to monitor and control basic functions of Controller devices in the field.

The SOCCS ASWC interface is a Java applet that is accessed via a browser. It gives the TMC operator basic control and monitoring capabilities of the ASWC and communicates with the remote ASWC using the controller message protocol. The screenshots below show the details of the interface.

Controller Applet Home Screen

The home screen shows a list of the configured controllers and gives a quick status and the current message on the configured CMS(s). The status of each controller can be OK, ERROR, or undefined (if communications can't be established). The display shows if the controller is in an active state and when this display was updated. Note that this display does not maintain a constant connection with the controller, rather it connects, retrieves the required data, then disconnects.

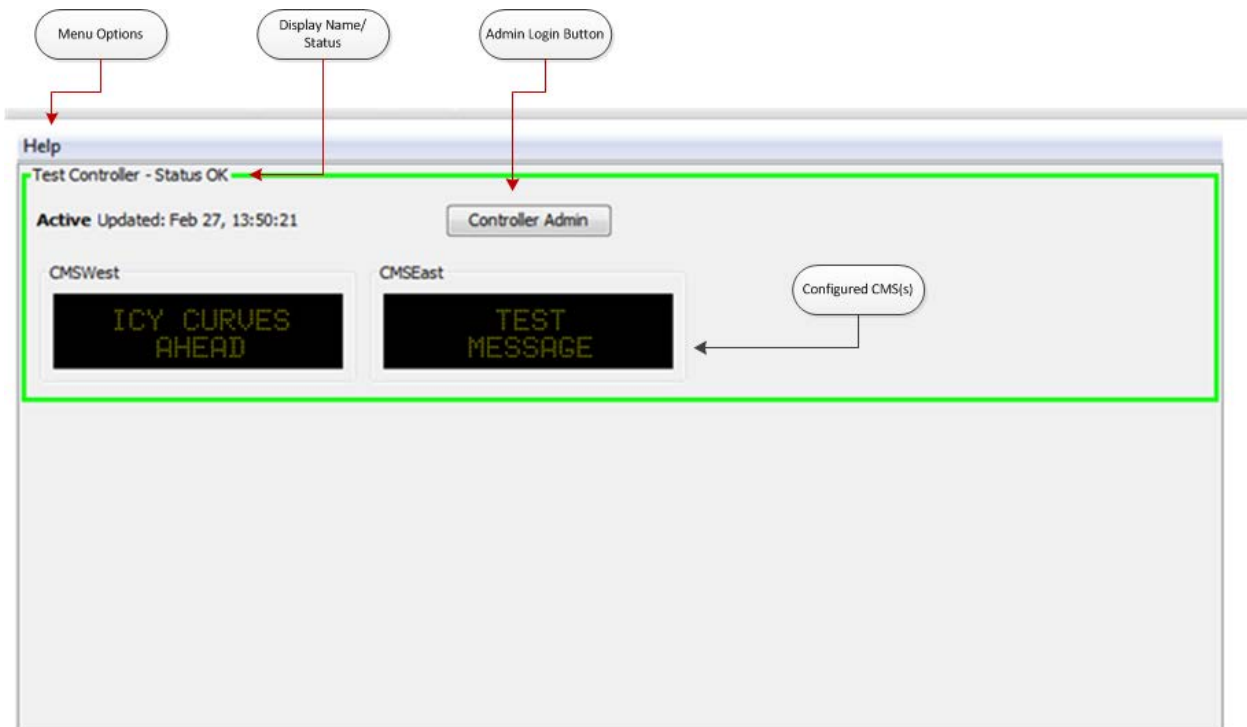


Figure 4: Controller Applet Home Screen

Menu Options

There are three help menu options available. The Settings tab is described in more detail below. Clicking the About tab will open up a web page providing background/contact information on the SOCCS ASWC. Clicking the User Guide tab would have brought you to this page.

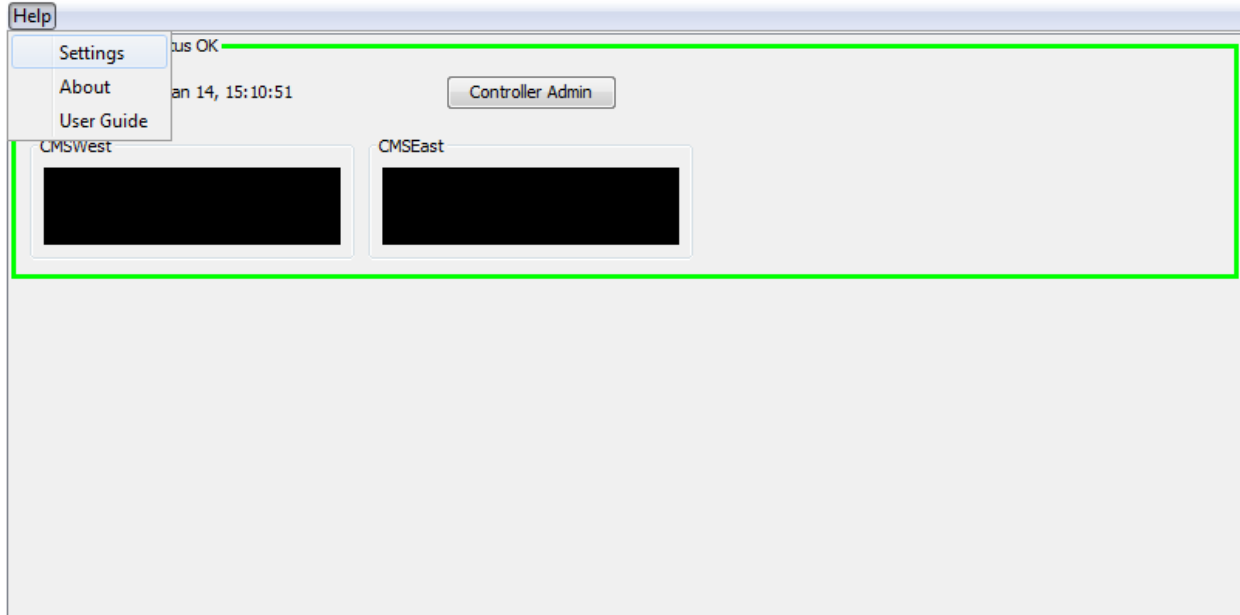


Figure 5: Controller Menu Options

Settings Tab

A window will display when the Settings tab is clicked. This window provides the current settings for all controller configuration entries including the Address, Display Name, Admin Update Interval (minutes), Timeout (minutes), and Update Interval (minutes). Note: a settings section will display for each active controller, separated by dashed lines.

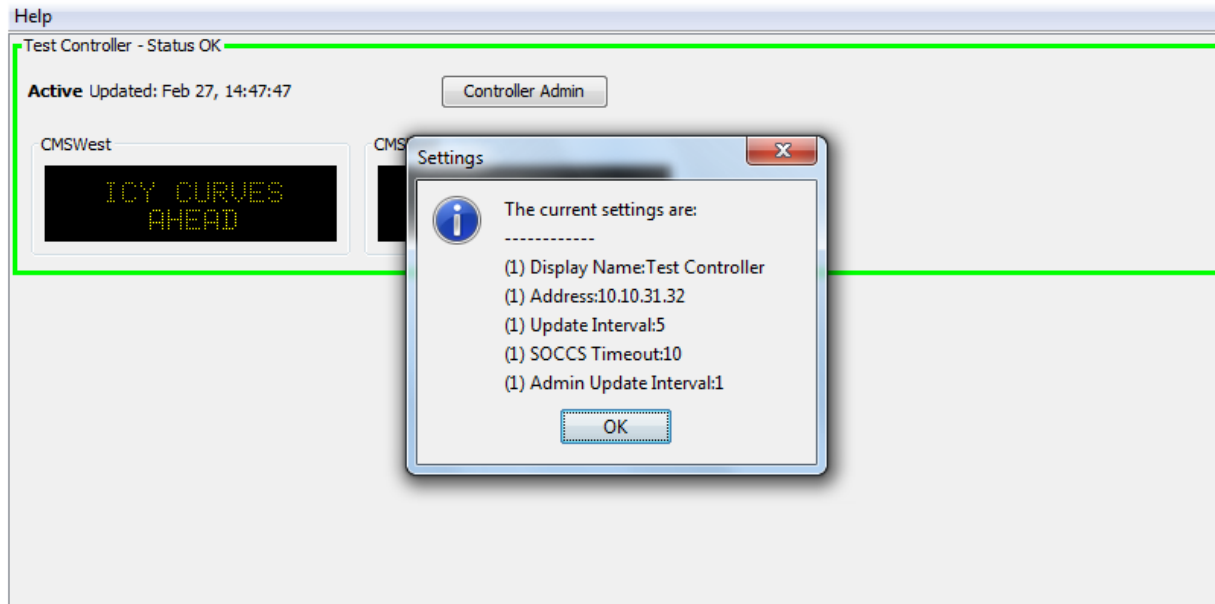


Figure 6: Controller Settings Tab

Controller Admin Button

Use the Controller Admin button to access administration screens giving more detailed information about the controller. An authentication window asking for the Username and Password will be displayed; you must a valid username/password combination to access the administration functions. Usernames and passwords are configured by the technician.

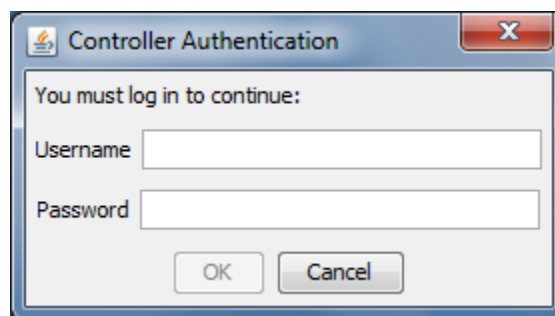


Figure 7: Controller Authentication

Admin Screen

After a successful login, a window will display allowing changes to be made to the controller. Selecting the “Disconnect and Close” button will terminate the connection with the ASWC and close the admin screen. Selecting the “Deactivate” button will place the ASWC in a state where it will continue to monitor conditions but will not put any messages on the CMS. The Controller will be shown as “Inactive” and there will be an “Activate” button that can be used to place the ASWC back in an active (normal) running state. Here is the SOCCS ASWC admin screen showing the ASWC in an inactive state.

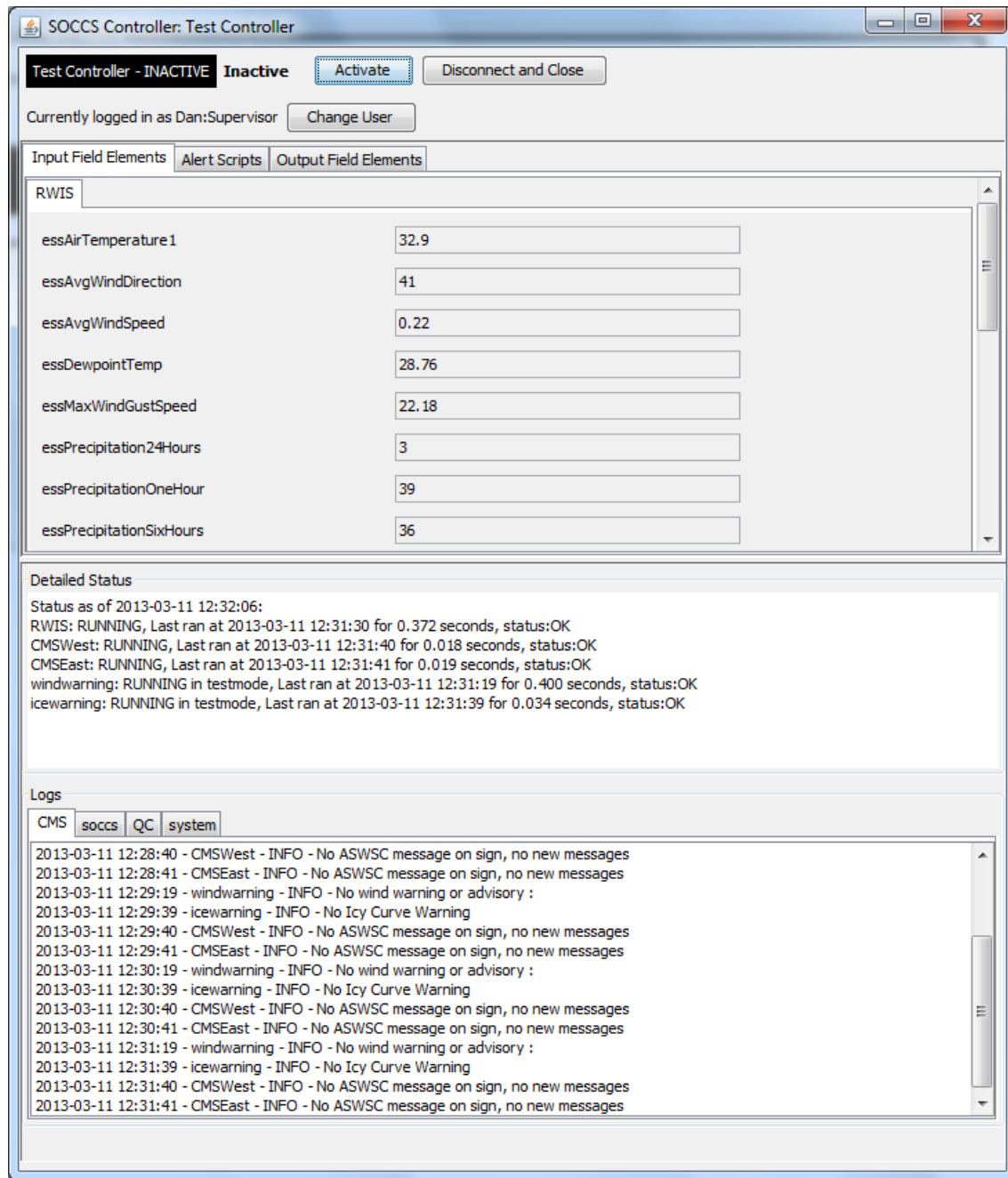


Figure 8: Controller Admin Screen Showing ASWC in an Inactive State

Here is the SOCCS ASWC admin screen with the Input Field Elements tab selected. Values of the input elements configured in the ASWC are shown in the top pane while the status of the modules and log messages are shown in the bottom two panes.

The screenshot shows the SOCCS Controller Admin Screen with the 'Input Field Elements' tab selected. The interface includes a status bar at the top indicating 'Test Controller - OK Active' with buttons for 'Deactivate' and 'Disconnect and Close'. Below this, it shows the user is logged in as 'Operator' with a 'Change User' button. The main content area is divided into three sections: 'Input Field Elements', 'Detailed Status', and 'Logs'.

Input Field Elements:

Element Name	Value
essAirTemperature1	32.9
essAvgWindDirection	41
essAvgWindSpeed	0.22
essDewpointTemp	28.76
essMaxWindGustSpeed	22.18
essPrecipitation24Hours	3
essPrecipitationOneHour	39
essPrecipitationSixHours	36

Detailed Status:

Status as of 2013-02-27 20:56:18:
 RWIS: RUNNING, Last ran at 2013-02-27 20:55:23 for 0.381 seconds, status:OK
 CMSWest: RUNNING, Last ran at 2013-02-27 20:55:20 for 0.018 seconds, status:OK
 CMSEast: RUNNING, Last ran at 2013-02-27 20:55:20 for 0.018 seconds, status:OK
 windwarning: RUNNING, Last ran at 2013-02-27 20:56:07 for 0.420 seconds, status:OK
 icewarning: RUNNING, Last ran at 2013-02-27 20:56:09 for 0.062 seconds, status:OK

Logs:

CMS | soccs | QC | system

```

2013-02-27 20:56:07 - windwarning - INFO - No wind warning or advisory :
2013-02-27 20:56:09 - icewarning - INFO - Icy Curve Warning. Values: 7, 3
2013-02-27 20:56:09 - icewarning - INFO - Icy Curve Warning. Values: 7, 3
  
```

Figure 9: Controller Admin Screen with Input Field Elements Tab Selected

Here is the SOCCS ASWC admin screen with the Alert Scripts tab selected:

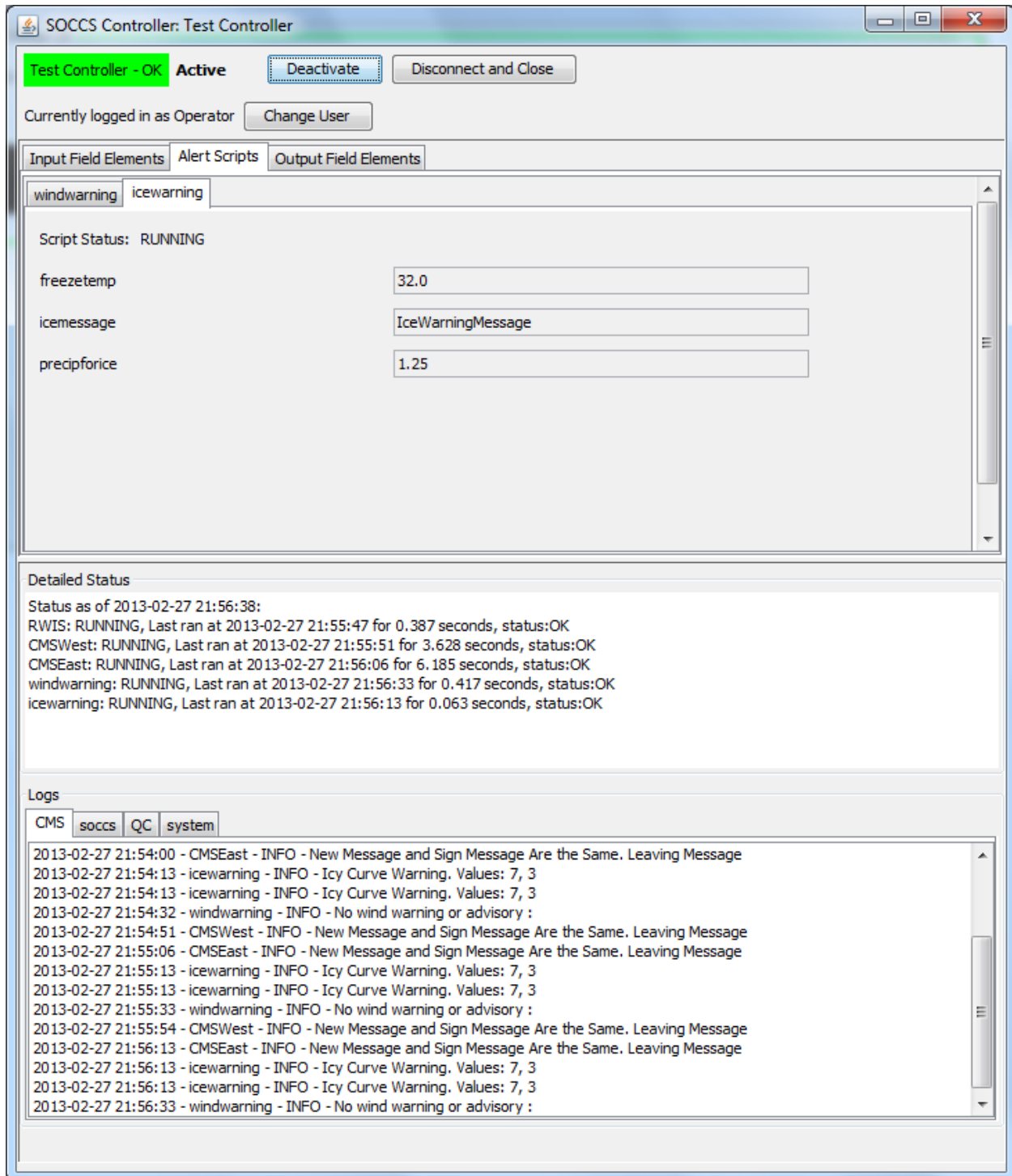


Figure 10: Controller Admin Screen with Alert Scripts Tab Selected

Here is the SOCCS ASWC admin screen with the Output Field Elements tab selected:

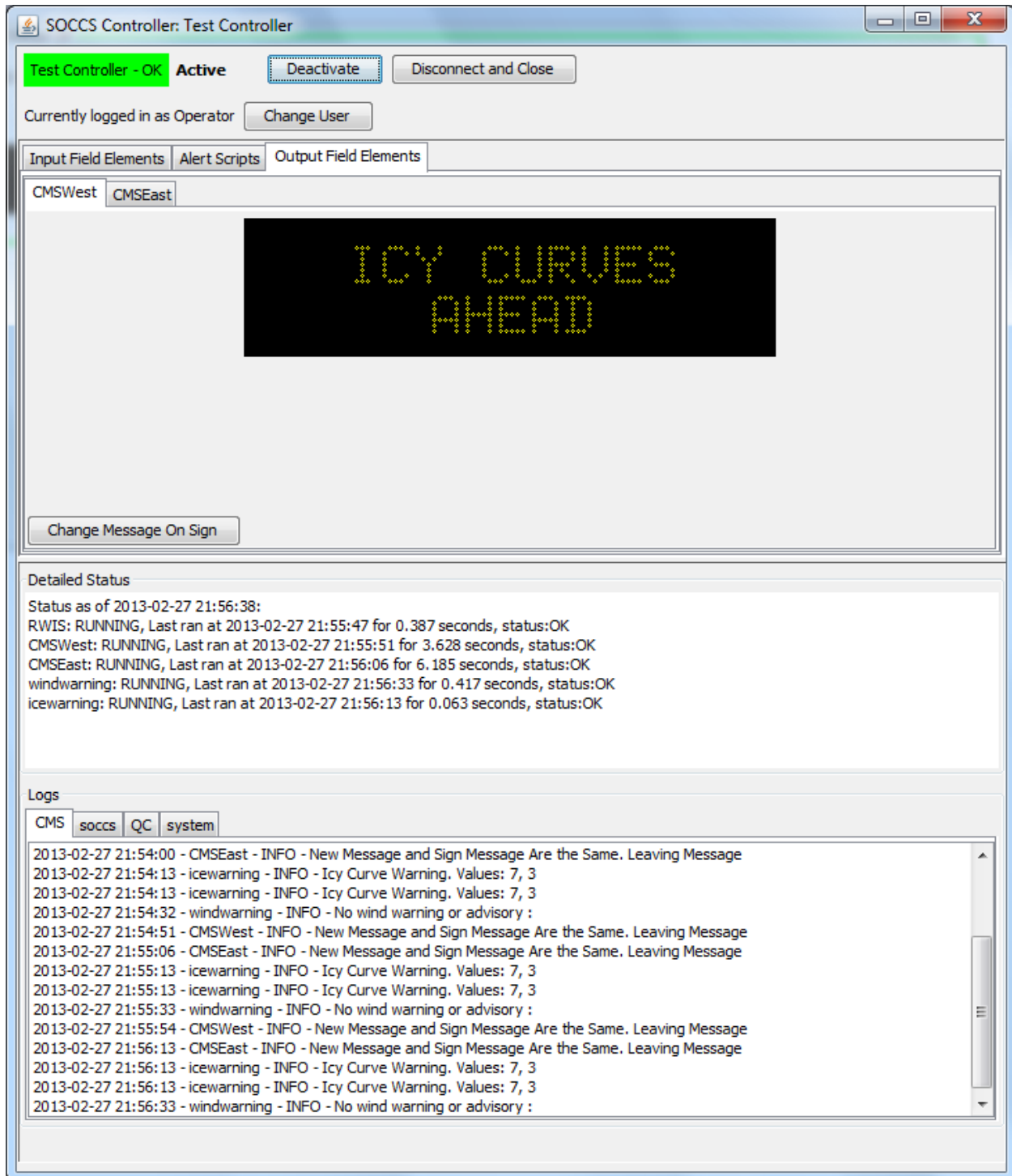
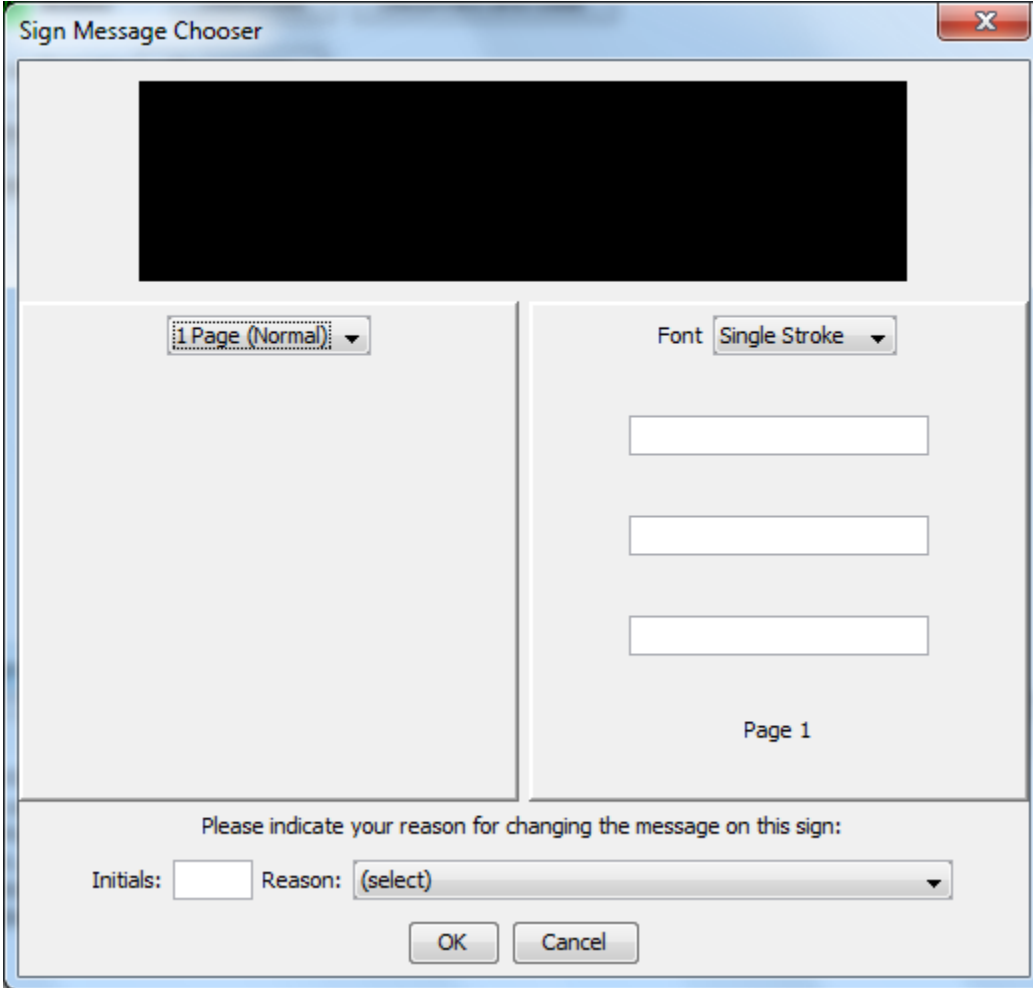


Figure 11: Controller Admin Screen with Output Field Elements Tab Selected

Selecting the Change Message On Sign button from the Output Field Elements tab brings up the Sign Message Chooser dialog box. This allows a one page message to be placed on the CMS. If the CMS is currently blank this message will be placed on page 1, if there is an ASWC message on the CMS this message will be placed on page two and will alternate with the ASWC message. Selecting “Blank” for the message type will blank out the CMS and allow the ASWC to put the appropriate message on the CMS.



Sign Message Chooser

1 Page (Normal)

Font Single Stroke

Page 1

Please indicate your reason for changing the message on this sign:

Initials: Reason: (select)

OK Cancel

Figure 12: Controller Sign Message Chooser

© Western Transportation Institute 2013