Test Automation for eConnects™

Lauren Syverson
Rockwell Collins, New Berlin, WI
Department of Physics, University of Wisconsin – River Falls
Department of Aerospace Engineering & Mechanics, University of Minnesota – Twin Cities

Abstract

Manual testing of systems controlled by a single eConnect™ requires weeks to complete. With the use of test automation, the time required for testing will be reduced to requiring only a few hours. Test automation is done with the use of Bash scripts. The Bash scripts that were created for test automation include moving map, cabin lights, and a system configuration editor. Some minor edits were made to the previously created Bash scripts such as DVT-Menu. The new Bash scripts are explained in high level detail. The results files from the tests are used to verify that the eConnect™ meet customer requirements. The results files are also used for traceability.

1. Introduction

In this modern age, both control and connectivity are in strong demand for business jets. An eConnect™ is an integrated unit, or an airplane computer, that controls cabin management, in-flight entertainment, and cabin wireless connectivity (B/E Aerospace, 2015). One version of an eConnect™ can be seen in Figure 1. Cabin management includes ambient temperature and lighting. In-flight entertainment includes features such as XM Satellite Radio and Blue ray DVD. Cabin wireless connectivity provides Wi-Fi so that passengers can go online. These features are controlled by using a GUI app on a personal device such as a smartphone or a tablet.

The main issue is that manual testing of the systems controlled by a single eConnect™ requires weeks to complete. The solution is to automate testing. This is done by making Bash scripts to completely test each system controlled by an eConnect™. With automatic testing, the time required drops down to only a few hours. This must be done to verify that the eConnects™ satisfy the customer requirements and for traceability.

Figure 1. This shows an eConnect™ (B/E Aerospace. (2017)

I want to thank Wisconsin Space Grant Consortium for financial support. I also want to thank Rockwell Collins for their assistance in this project.

Several Bash scripts were already created to test some systems on eConnect™ such as XM Satellite and Blue Ray DVD. A user can select a Design Verification Test, or a DVT, by running a Bash script called DVT-Menu. The purpose of this project was to accelerate automatic testing by creating additional DVT Bash scripts. The Bash scripts created for this project include moving map, cabin lights, and a system configuration editor.

2. Materials for Research

Before beginning test automation for an eConnect™, the necessary materials must be acquired. The following includes a list of devices needed for test automation. The eConnect™ is preinstalled with programs such as icdapp, GUI 2.0, and a MySQL database. The eConnect™ uses Gentoo Linux and a Bash shell compiler. A Windows 7 laptop with PuTTY and FileZilla installed was also used. PuTTY is a program on that uses the command line from Linux on the eConnect™. FileZilla is a program that transfer files between the eConnect™ and the laptop.

Note that no materials needed to be acquired for testing moving map or for the system configuration editor. Only the lights that will be tested are used for test automation. The lights used for testing include a Quasar II, a MultiWhite, and an RGB+A Flex. The Quasar II is a multicolored mood LED wash light. The MultiWhite is a white LED wash light. The RGB+A Flex light is a flexible multicolored mood light.

The following includes a list of cables needed for the project. An Ethernet cable is for communication between the laptop and the eConnect™. An RS-485 bus is for communication between the eConnect™ and the lights. A power supply is used for providing power to the lights and to the eConnect™. Eventually, an ARINC 429 interface will be used to provide realistic data to properly test the MMAP on the eConnect™.

3. DVT-Menu

DVT-Menu is a Bash script that generates a menu to allow a user to select a specific DVT such as ALTO-TESTS or AVDS-TESTS. This can be observed in Figure 2. After the user selects a test, the specified DVT will run the selected Bash test script. After conducting the selected Bash test script, the DVT-Menu will allow the user to select another DVT to conduct. Selecting "finished" will end the script.

In the DVT-Menu script, two new functions were created. One function asks the user for first and last name. The other function asks the user for the date and time the test is being conducted. The DVT-Menu script was also edited to include LIGHTING-TESTS and MMAP-TESTS. These test Bash scripts did not exist prior to the beginning of this project. In the test Bash scripts, minor edits were made to allow the new functions to be imported from DVT-Menu. The purpose of doing this was to maintain consistency in all results files.

```
Enter your first and last name
Lauren Syverson
The Date and Time is 2017-07-31 13:49:52. Is this correct?
y


Select DVT Test Group to execute:
1) ALTO-TESTS
2) AVDS-TESTS
3) IO-MODE-TESTS
4) ROSEN-BLURAY-TESTS
5) ROSEN-MONITOR-TESTS
6) XM-RADIO-TESTS
7) LIGHTING-TESTS
8) MMAP-TESTS
9) finished
Next DVT Test Group to execute?
```

Figure 2. This shows the DVT-Menu, which allows user to select which test to conduct. Notice how it asks for name, date, time at the beginning of the script.

4. Moving Map Testing

Moving Map, or MMAP, is a system on the eConnect™ that informs passengers on flight conditions in real time. To conduct a test for the MMAP on an eConnect™, a Bash script was created to simulate a simple flight. This script simulates flight by updating values of latitude and longitude, altitude, and the time remaining on the MySQL database (Emteq, 2016). The results of the data changes can be observed on the GUI 2.0 as observed in Figure 3.

Figure 3. This shows a MMAP on the GUI 2.0. On the bottom of MMAP is a row that shows Time Remaining, Altitude, and Latitude/Longitude. Notice the airplane icon on the map, which is located at Denver, CO.

The script generates a menu that allows the user to decide which MMAP test to conduct. Selecting "1" will make user input initial and final locations and amount of time for flight. Selecting "2" will use default locations and time for flight. Selecting either "1" or "2" will prompt the user to verify success or failure for the types of test conducted in the flight simulation.

Selecting “3” will use default information input info but will not prompt the user for success or failure of type of tests conducted in the flight simulation. Options 1-3 will go through an entire flight simulation. Selecting “4” will end the script. The menu can be observed in Figure 4. The flight simulation script is composed of three sub-functions. Once all three individual sub-functions performed as expected, then they were combined to create a single dynamic flight simulator.

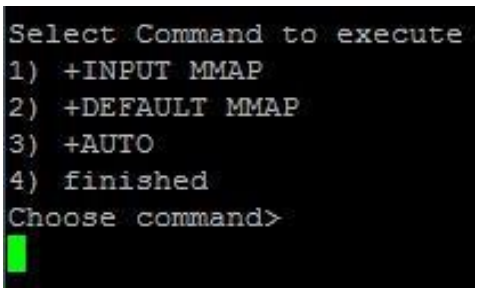

Figure 4. This shows MMAP menu for user selection. The user can select INPUT MMAP, DEFAULT MAP, AUTO, or finished.

The first function in the flight simulation automates changes in latitude and longitude. The user can decide to input initial and final locations of flight, or the user can use default locations of Milwaukee WI and Denver, CO. The input locations for latitude and longitude are in decimal degrees form (i.e. 38.95° instead of 38°57’33’’N). To determine the new values of latitude and longitude, the script uses the line equation from Eq. 1.

$$y=mx+b \quad (1)$$

The new latitude value is described with y, and the new longitude is described with x. The longitude value, x, depends on step size, which depends on distance between x_f and x_i. The slope of the line, which refers to “m” in Eq. 1, is described in Eq. 2.

$$m = \frac{y_f - y_i}{x_f - x_i} \quad (2)$$

The y intercept, which refers to "b” in Eq. 1, is y_i. Note that the “i" subscript refers to initial location, and the “f” subscript refers to final location. The user can see the aircraft icon moving on GUI 2.0 as the data in MySQL continuously updates.

The second function in the flight simulation automates changes in altitude. To simulate flight the values of the altitude are adjusted. The altitude of the aircraft is set at 1 ft. to simulate takeoff from initial location. The altitude is set at 35,000 ft. to simulate an aircraft at steady cruising flight. The altitude of the aircraft is set at 1 ft. to simulate landing at final location.

The third function in the flight simulation automates changes in time remaining for flight. If the user decides to input initial and final locations, then the script will ask the user for total time for flight in minutes. If the user decides to use default locations, then the total time for flight will be set to 180 minutes. The time remaining for flight depends on how much the aircraft travelled in the flight simulation.

All three sub-functions are the "TEST_TYPES" that are conducted for the flight simulator. These sub-functions are verified by use of visual confirmation of the MMAP as observed in Figure 3. "TEST_TYPE" of "latitude_longitude" is considered a success if the airplane icon on the MMAP correctly moved in the expected trajectory (i.e. latitude and longitude). "TEST_TYPE" of "time_remaining" is considered a success if the amount of time remaining for flight on bottom row reduces in value correctly. "TEST_TYPE" of "altitude" is considered a success if the altitude is 1 ft. at the beginning and end of the flight simulation, and is 35,000 ft. throughout the rest of the flight simulation.

After a flight simulation is completed, the test script asks the user if test performed as expected. The user then types in a "yes" for a success, or a "no" for a failure. The results of the test are stored onto a file in the eConnect™. The results file stores the first and last name of user, the date and time test was conducted, test group, command used, test type, and result of test. A sample results file for MMAP test can be observed in Figure 5.

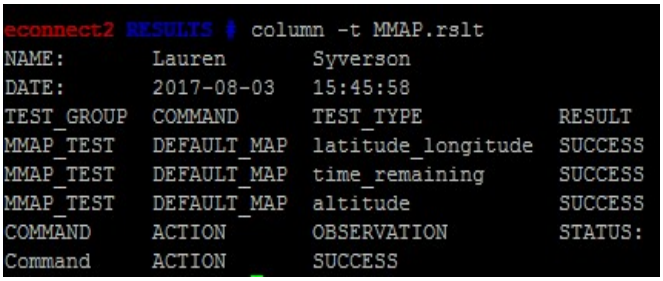

Figure 5. The figure shows a sample results file for the MMAP test. The TEST_GROUP is shown as MMAP_TEST. The COMMAND used in this example was DEFAULT_MAP. The individual component (TEST_TYPE) includes latitude and longitude, time remaining, and altitude. The result of each test types are success.

The Bash script created to make a simple flight simulation cannot be used to fully test MMAP on the eConnect™. To resolve this issue, an ARINC 429 interface will be used to create a more realistic flight simulation. The interface will feed dynamic data into the eConnect™. This will enable a more accurate way to gauge how an eConnect™ will perform when used on an aircraft.

5. Cabin Lights Testing

Cabin lights are used to illuminate the cabin interior, which is the area where passengers sit inside an aircraft. An eConnect™ allows control over the lights using the GUI 2.0. The lights used for testing include the Quasar II, the MultiWhite, and the RGB+A Flex, as seen in Figure 6. The Quasar II and the RGB+A Flex were both tested for color and intensity. The Multiwhite was tested for intensity as it only produces white light.

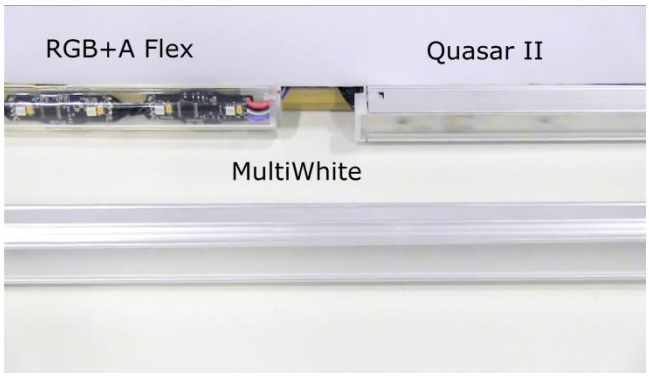

Figure 6. The three cabin lights are used for the test. The RGB+A Flex is on the top left, the Quasar II is on top right, and the Multiwhite is on the bottom.

A program installed on an eConnect™ is icdapp, which issues commands to all three lights. The codes required for colors and for intensity on icdapp were separate (Emteq, 2016) (Emteq, 2017). After both colors and intensity were confirmed to work separately, a Bash script was created to dynamically test the lights using icdapp commands. The colors used in the test include red, white, and blue. The intensity of the lights ranges from maximum to minimum brightness.

The user can select to test any of the lights, auto to test all lights, or "finished" to end the script. Selecting "1", "2", or "3" will test the respective lights and ask for confirmation of components. Selecting "4" will test 1-3 consecutively without confirmation. The menu for lighting test can be observed in Figure 7.

```
Select Command to execute
1) +CLNP ELW72 QUASAR II (BUS 1)
2) +CLNP ELW92 MULTIWHITE (BUS 2)
3) +CLNP RGB+A FLEX (BUS 3)
4) AUTO
5) finished
Choose command>
```

Figure 7. This shows light menu for user selection.

The tests conducted on the lights are determined to be successful or failures with the use of visual confirmation. "TEST_TYPE" of "intensity" is considered a success if the light starts at maximum brightness, then fades to off. "TEST_TYPE" of "color" is considered a success if the lights are in sequential order of red, then white, then blue. The "TEST_TYPE" of "color" does not apply to the MultiWhite light.

After the light tests are completed, the script asks the user if test performed as expected. The user then types in a "yes" for a success, or a "no" for a failure. The results of the test are stored onto a file in the eConnect™. The results file stores the first and last name of user, the date and time test was conducted, test group, command used, test type, and result of test. A sample results file can be observed from Figure 8.

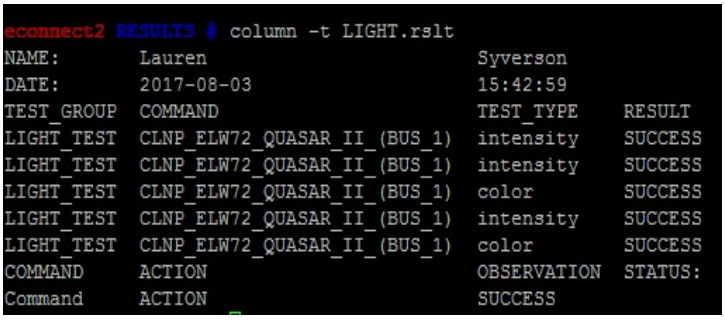

Figure 8. This shows sample results file for the light test.

6. System Configuration Editor

For troubleshooting the eConnect™ and to accelerate testing, a system configuration editor was created. This tool allows a user to quickly enable, disable, or enumerate systems that are controlled by an eConnect™. To create this editor, a Bash script was created. First, the script asks for first and last name of user, and then it asks for the current date and time. This is used for labeling the MySQL data dumps. Second, the script generates a menu to allow the user to select a field for editing. The menu can be observed in Figure 4. Then the user can select to enable, disable, or enumerate the chosen field. A MySQL data dump is performed before any changes are made. This is done if the user wishes to "undo" or "restore" changes.

The script then asks the user to select "Continue", "Update", or "Exit". Selecting "Continue" will allow the user to continue using the system configuration editor. Selecting "Update" will cause a MySQL data dump onto a file in the results directory. It will also save all changes permanently in the MySQL database. Selecting "Exit" will end the system configuration editor. The menu can be observed in Figure 9.

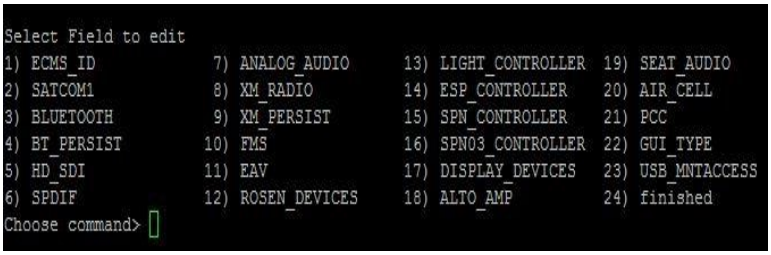

Figure 9. This shows menu generated for System Configuration Editor. These are all the fields that are available for making changes in eConnect.

7. Conclusion

Manual testing of the eConnect™ takes weeks to complete because it controls many systems. A user can do full automatic testing of eConnect, which will save time. Several Bash scripts were already created to test some features available on eConnect™. New Bash scripts were created to test additional features on an eConnect, including MMAP, cabin light, and a system configuration editor. After tests are done using any test scripts, the results are also stored onto a file in the eConnect™ in respective directories for traceability. Both the old and new DVT scripts are used for the entire testing process. Minor edits were made to DVT-MENU and main test menus to

streamline results files. The eConnect™ allows passengers to travel in comfort, be productive, and stay connected in this modern age.

References

B/E Aerospace. (2017, June 5). eConnect Part Numbers. New Berlin, Wisconsin, United States of America.

B/E Aerospace. (2015, February 2). Hardware Installation Manual eConnect 2.0 Product Line. New Berlin, Wisconsin, United States of America.

Emteq. (2016, October 6). eCMS Aftermarket App Database ICD PRAN. New Berlin, WI, United States of America.

Emteq. (2017, June 23). QII Protocol Cheat Sheet. New Berlin, Wisconsin, United States of America.