

Software Verification of Orion Cockpit Displays

M. A. Rafe Biswas, Matthew Prado, Samuel Garcia, Sadad
Hossain, Matthew Souris and Lee Morin

12th IEEE ICCSE Conference 2017

Orion and its Cockpit Displays

- NASA's Orion spacecraft is built to take humans to Mars effectively and safely
- To monitor spaceflight status and provide instructions to crew, cockpit has new software embedded digital displays to replace hundreds of pounds of paper documentations
- Digital displays have three glass screens to show graphical objects that are manipulated on screen using peripheral switches such as edge keys and knobs



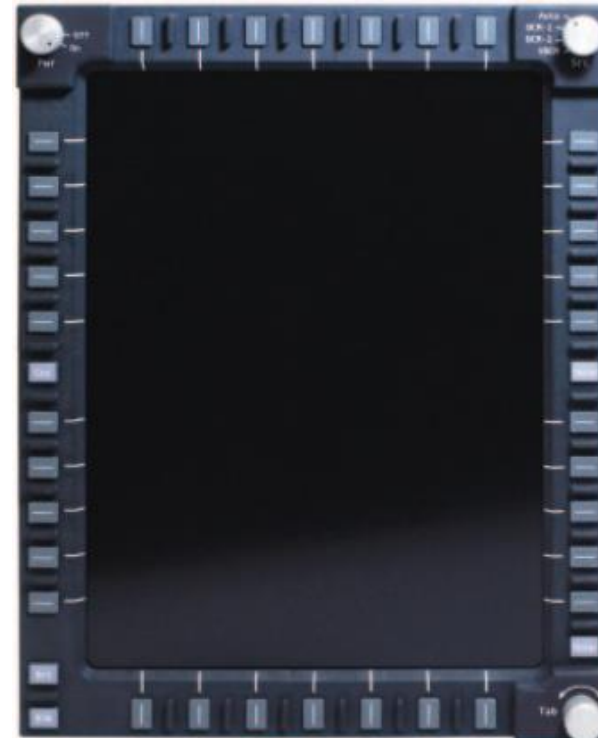
Motivation

- Many physical man-hours go into monotonous and manual software verification of human spacecraft displays
- Need a more automated, robust and thorough verification method for Orion's cockpit displays to reduce human error



Goal

- Verify the display software using an automation tool such as EggPlant Functional
- Develop a framework to allow for repeatability and consistency of verification of different display items

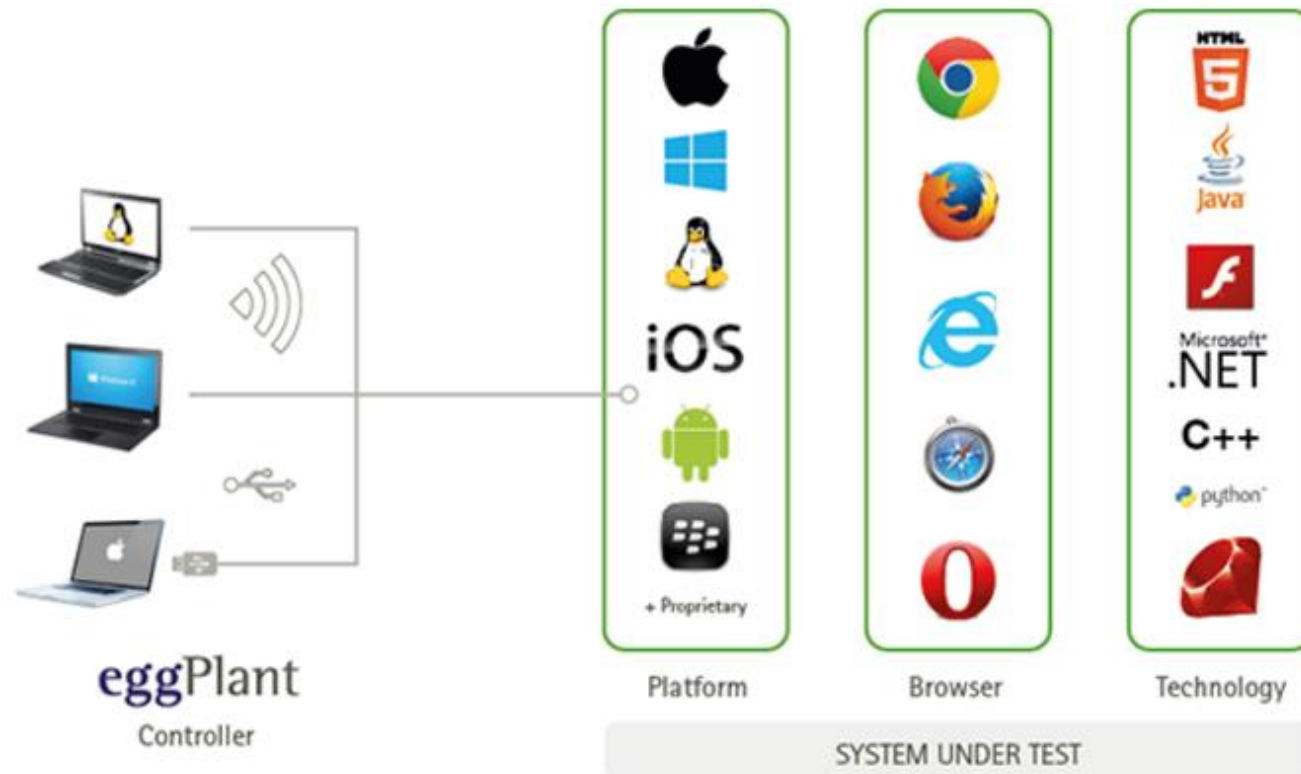


What is EggPlant?

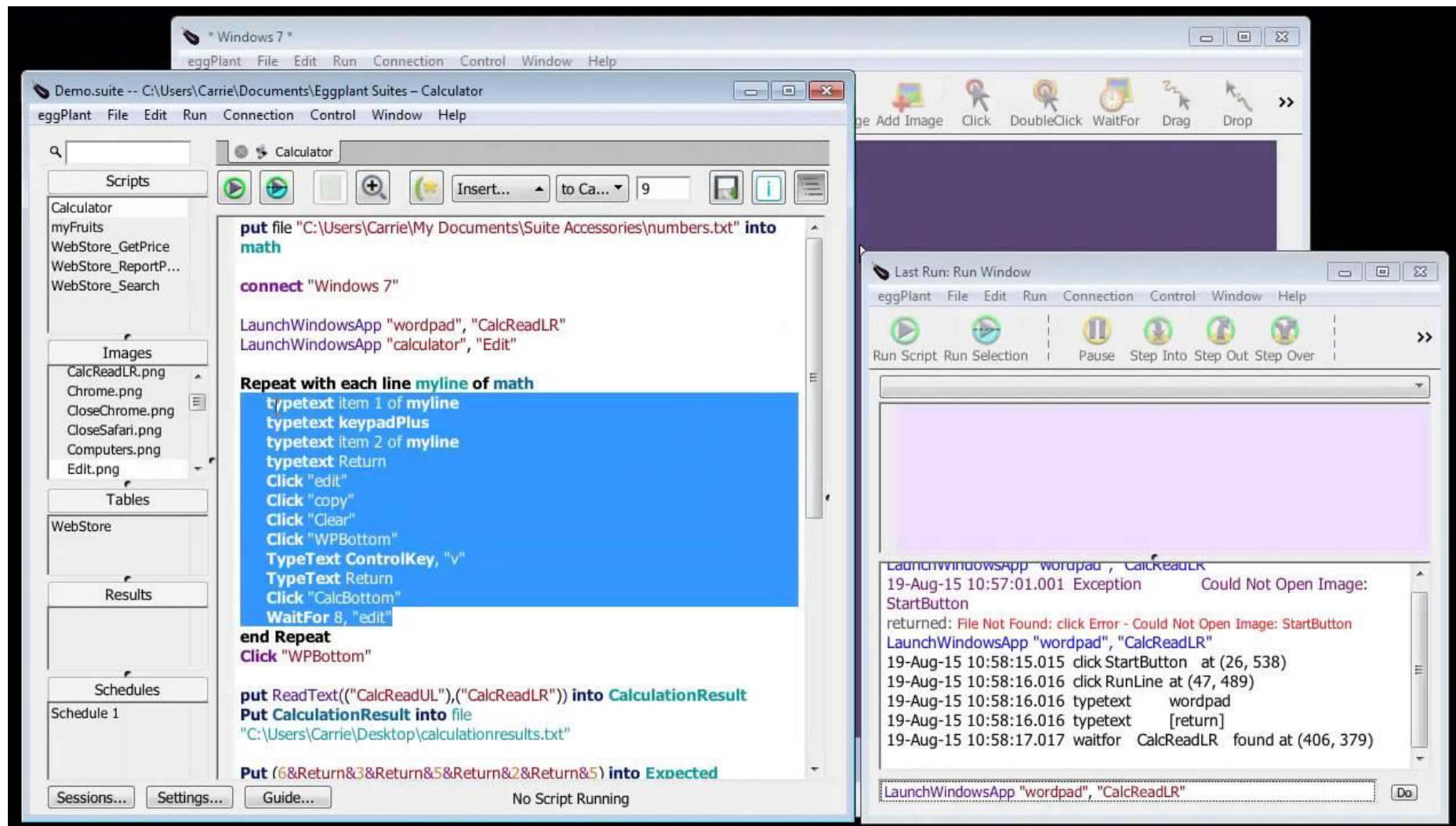
- Verification Software that utilizes image recognition
- Automation capabilities through simulation of user inputs
 - Clicks
 - Keyboard Keystrokes
- User-like testing method
 - Recognition of object on screen followed by a click or keystroke
- Thorough and long duration tests through automation
- Knowledge of background code or internal processes not needed

How Does EggPlant Work?

- EggPlant Functional works in a Controller – System Under Test Relationship



How Does EggPlant Work?



How Does EggPlant Work?

- Verifies nominal function by recognizing an image that appears at a specific instance during a test.
 - Images are captured prior to testing.
 - An example test on Windows OS would be recognizing an image of the Windows Start Menu after clicking on the Windows logo on the Windows Taskbar.
- Comes with a GUI to write test automation scripts for your System Under Test.
- Uses a language called SenseTalk for its scripts.
- Scripting is intuitive.
 - Code user interactions with simple EggPlant commands

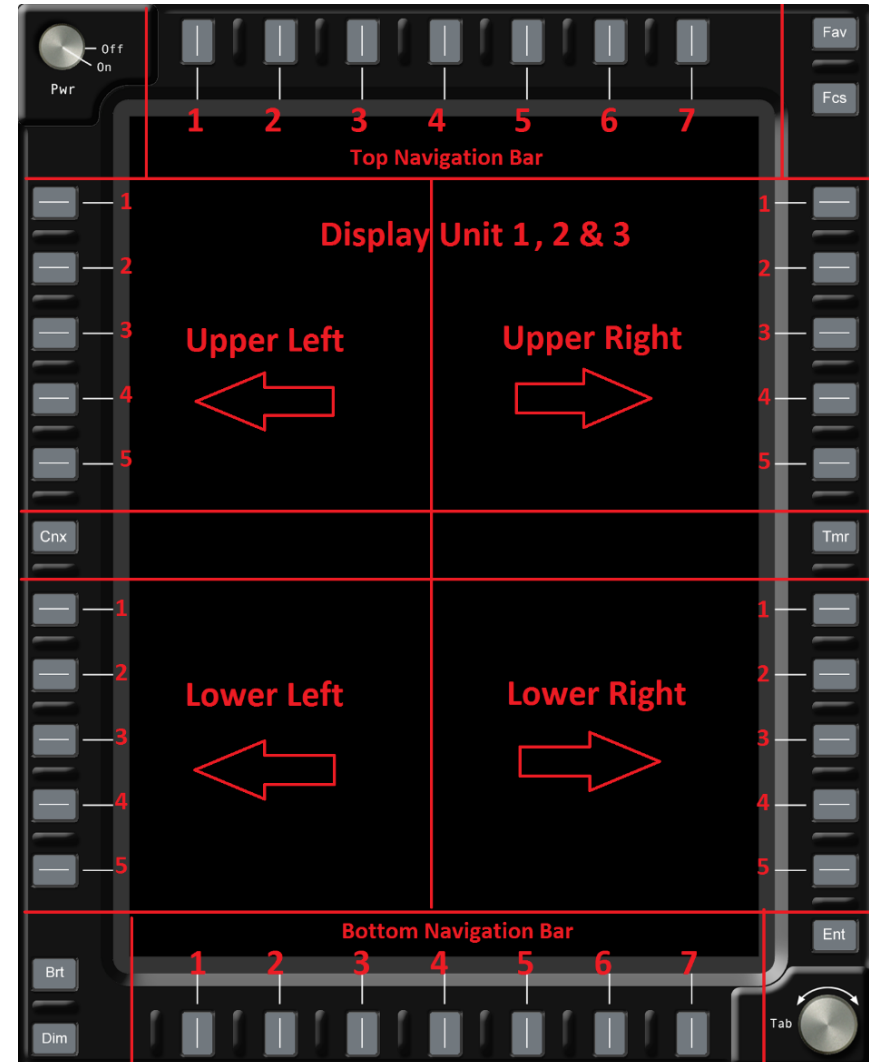
How Does EggPlant Work?

Example Code:

```
Click(Image:"sample_image", waitfor:5) // Click image  
Typetext "Sample text."                // Type "Sample text"  
  
ImageFound(Image:"sample_image") // Conditional Statement  
                                     // Returns True or False
```

Display Software Verification

- Display Unit are split into:
 - Upper Left Quadrant
 - Upper Right Quadrant
 - Lower Left Quadrant
 - Lower Right Quadrant
 - Top Navigation Quadrant
 - Bottom Navigation Quadrant



Framework Definitions

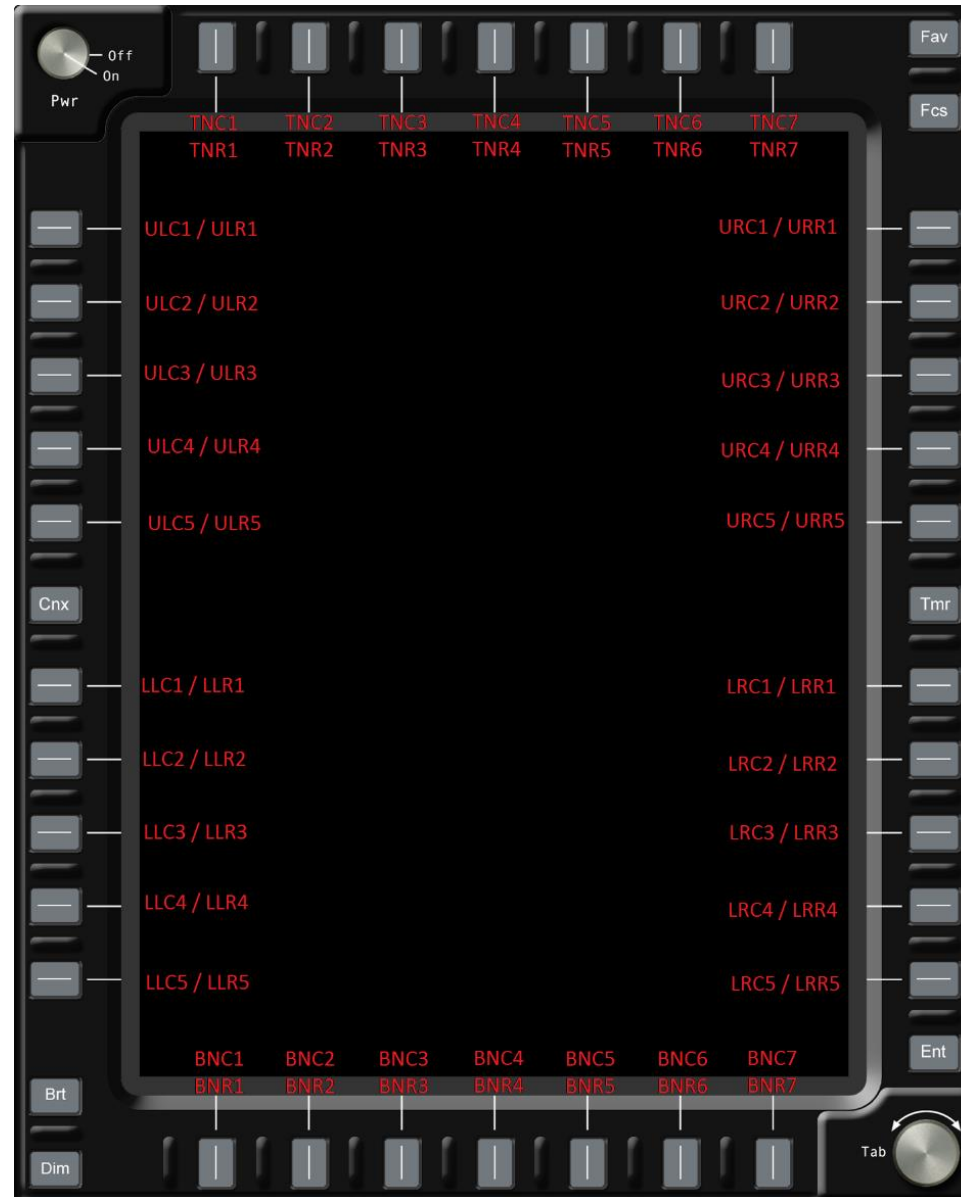
Framework Description	Framework Name
Display Unit 1	DU1
Display Unit 2	DU2
Display Unit 3	DU3
Cursor Control Device 1	CCD1
Cursor Control Device 2	CCD2
Display Calls	DC

Upper Left Quadrant

<u>Desired EdgeKey Function</u>	<u>Variable Name</u>	<u>DU1</u>	<u>DU2</u>	<u>DU3</u>
EK_Upper left click 1	ULC1	*30	,30	.30
EK_Upper left click 2	ULC2	*31	,31	.31
EK_Upper left click 3	ULC3	*32	,32	.32
EK_Upper left click 4	ULC4	*33	,33	.33
EK_Upper left click 5	ULC5	*34	,34	.34
EK_Upper left release 1	ULR1	'30	;30	\30
EK_Upper left release 2	ULR2	'31	;31	\31
EK_Upper left release 3	ULR3	'32	;32	\32
EK_Upper left release 4	ULR4	'33	;33	\33
EK_Upper left release 5	ULR5	'34	;34	\34

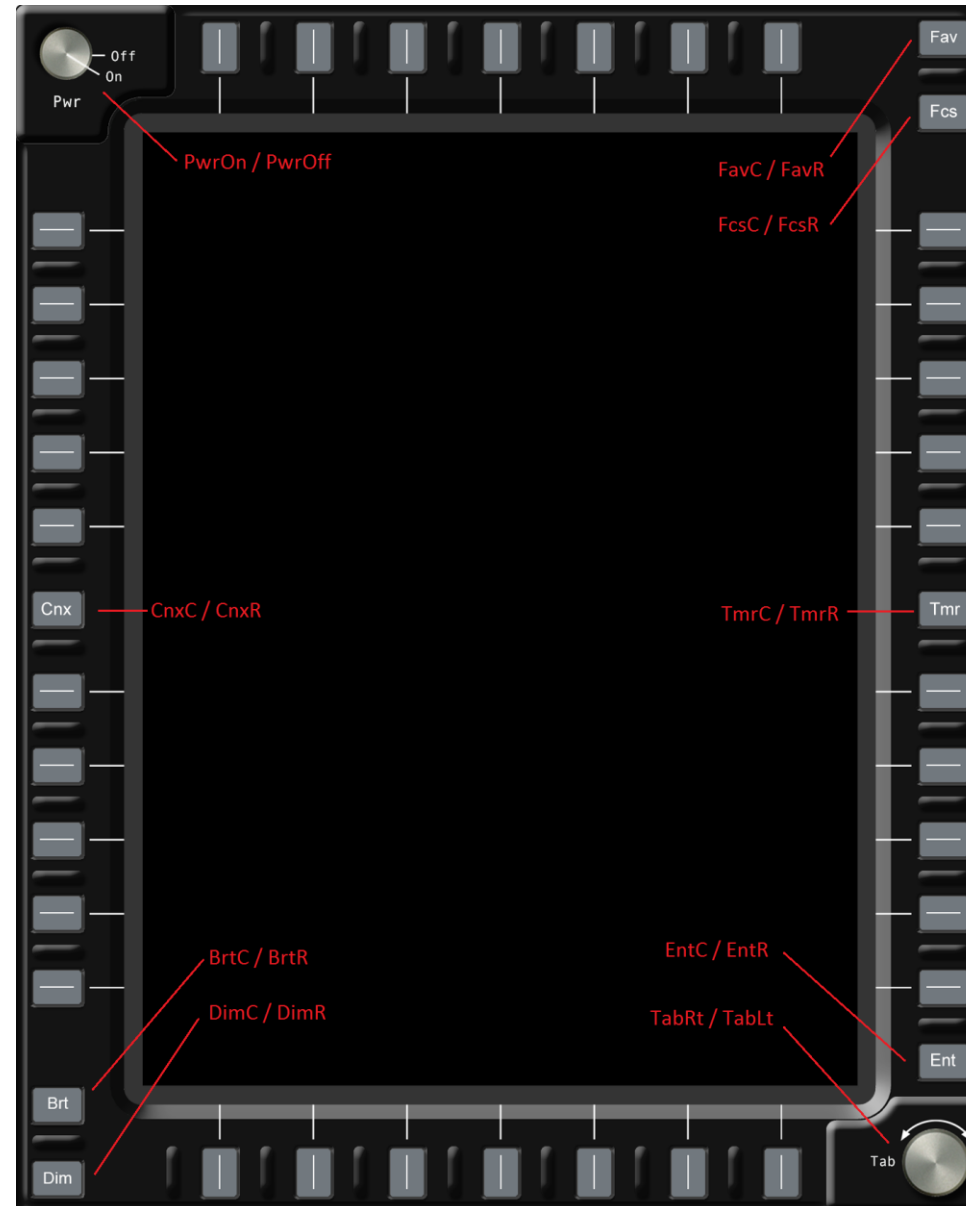
Display Software Verification

Schematic of the main Edge Keys with their assigned variable names.



Display Software Verification

Schematic of the miscellaneous Edge Keys with their assigned variable names.



Display Software Verification

Example Code:

```
put "C:\Users\user\Desktop\DisplayUnit_KeyStroke_Framework.suite\Scripts\DU1"  
into DU1
```

```
//Navigates to MPS Display
```

```
    put DU1.Flt_MPS
```

```
//Click and release 2 edge key buttons
```

```
    put DU1.URCR1
```

```
    put DU2.LRCR3
```

Conclusions and Future Work

- A display test script run multiple times on EggPlant Functional proved verification was carried out successfully
- EggPlant Functional and SenseTalk proved to be highly useful tools for automating the testing of the Orion cockpit display simulators.
- The Edge Key framework and Common EggPlant Function scripts enabled to write modular and re-usable code to apply to different displays and their items:
 - popup types
 - enumerations.
- Common Eggplant Functions can be used repeatedly in tests of other displays, and such iterative testing will help validate reliability and consistency of simulation of Orion cockpit display software.

Acknowledgments

- University of Texas at Tyler
Office of Research and
Technology Transfer
- NASA JSC's Rapid Prototype
Laboratory members including
Jeff Fox and Patrick Henry
- University of Texas at Tyler
engineering students



Thank You!



Keysight enables innovators to push the boundaries of engineering by quickly solving design, emulation, and test challenges to create the best product experiences. Start your innovation journey at www.keysight.com.

This information is subject to change without notice. © Keysight Technologies, 2017 – 2023, Published in USA, June 2, 2023, 3123-1301.EN