Software and Human-Machine Interface Development for Environmental Controls Subsystem Support

Matthew Dobson
Kennedy Space Center
Major: Electrical Engineering
NIFS 2018 Spring Session
Date: 05/04/2018

Software and Human-Machine Interface Development for Environmental Controls Subsystem Support

Matthew R. Dobson
Colorado School of Mines, Golden CO, 80401

Nomenclature

ECS	=	Environmental Control Subsystem
EM-1	=	Exploration Mission 1
HMI	=	Human-Machine Interface
KSC	=	Kennedy Space Center
LCC	=	Launch Control Center
ML	=	Mobile Launcher
NASA	=	National Aeronautics and Space Administration
PLC	=	Programmable Logic Controller
PID	=	Proportional-Integral-Derivative
SLS	=	Space Launch System
V&V	=	Validation and Verification
VAB	=	Vehicle Assembly Building

Abstract

The Space Launch System (SLS) is the next premier launch vehicle for NASA. It is the next stage of manned space exploration from American soil, and will be the platform in which we push further beyond Earth orbit. In preparation of the SLS maiden voyage on Exploration Mission 1 (EM-1), the existing ground support architecture at Kennedy Space Center required significant overhaul and updating. A comprehensive upgrade of controls systems was necessary, including programmable logic controller software, as well as Launch Control Center (LCC) firing room and local launch pad displays for technician use. Environmental control acts as an integral component in these systems, being the foremost system for conditioning the pad and extremely sensitive launch vehicle until T-0. The Environmental Controls Subsystem (ECS) required testing and modification to meet the requirements of the designed system, as well as the human factors requirements of NASA software for Validation and Verification (V&V). This term saw significant strides in the progress and functionality of the human-machine interfaces used at the launch pad, and improved integration with the controller code.

I. Introduction

After the cancellation of the Constellation in 2009 and Shuttle program in 2011, the future of manned spaceflight missions from Kennedy Space Center was uncertain. Routine spaceflight was shifted towards the commercial sector, and as aerospace companies developed their own launch vehicles to support the agency, a new vehicle was devised internally at NASA. Unveiled as the Space Launch System, this modular rocket would rival and eventually surpass the Saturn V.

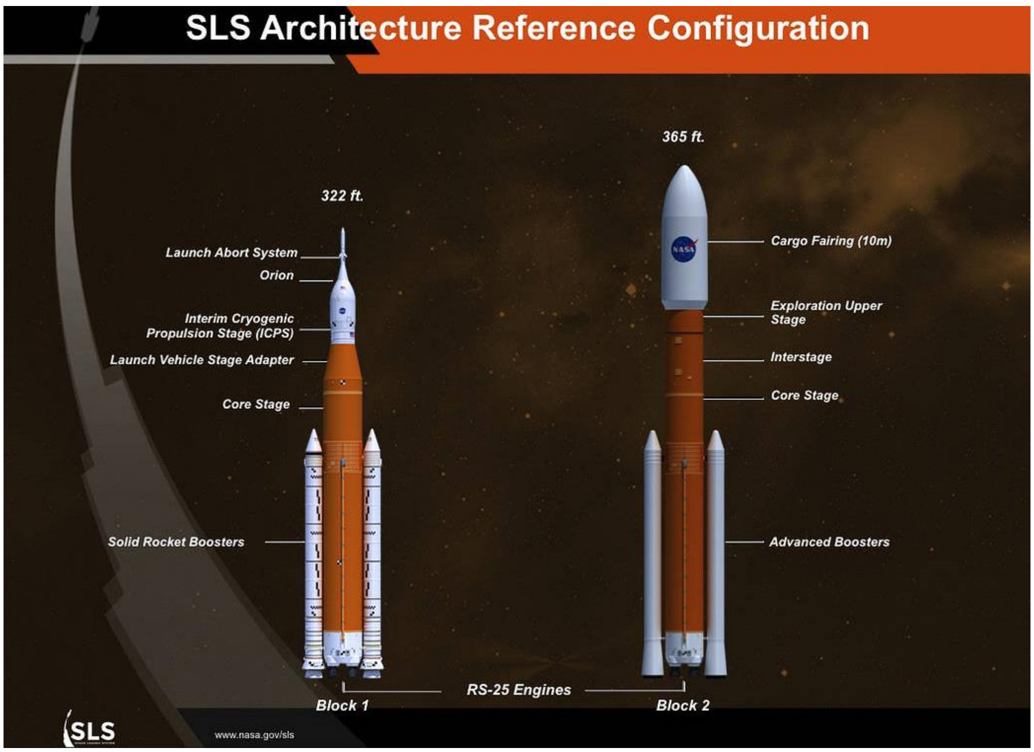

Figure 1: Outline of SLS modularity[1]

[1] NASA, "Space Launch System," NASA, [Online]. Available: https://www.nasa.gov/exploration/systems/sls/multimedia/gallery/SLS_Concepts.html

The design focus was on modularity and upgradability across its lifespan to adapt for an ambitions future of space exploration, taking large payloads and spacecraft well beyond Earth orbit. To support this endeavor, new launch support hardware was designed and installed for the Vehicle Assembly Building (VAB), launch pad 39B, and a new mobile launcher (ML) platform. Additional electrical systems, software and instrumentation upgrades were also implemented in the infrastructure updates.

Figure 2: Mobile launch platform on Pad 39B, the future launch site of the SLS program[2].

KSCs ground support programs include management and development of industrial controls systems to maintain operation of the space port. The ECS component of these controls systems includes environmental conditioning and power for the pad and vehicle, as well as sensor data logging to support operation. My role and project involved working with the ECS group as controls system architecture was upgraded to support the SLS.

Industrial controls systems are a hallmark of modern automated process control, and are governed by the use of programmable logic controllers (PLCs). These controllers act as the processing core of the system, and are specialized computer hardware designed to continuously operate in stressful and extreme conditions. Human operation of the system is performed through a human-machine interface (HMI), a set of specialized computer displays that show control functionality and indicate system data.

These HMI displays serve as the interface between technician and the control system itself. The purpose is to enable control of the system in an intuitive fashion, as well as provide on-site information instantaneously. Important display elements include sensor feedback data, fault and alarm monitors, as well as command keys controlling the operation of the system. As a controls software developer, the primary distribution of my work involved modification and design of HMI displays. Reliable and continuous automation is the baseline requirement, excellence in design and functionality is a NASA standard, and the user experience of technicians operating the system must also be a prime consideration.

[2] NASA, "KSC-2011-8047," NASA Images and Video Library, [Online]. Available: https://images.nasa.gov/details-KSC-2011-8047.html

Data and control elements are referenced in the form of tags, which are a unique identifier from the PLC code. Tags can carry various data types, such as digital boolean, or analog inputs, and form the structure of the code. The programming format is referred to as “ladder logic”, named for its sequence of progressive “rungs” that perform operations. The bulk of ladder logic involves toggle components that act like normally open or normally closed switches, which sequentially enable or disable indicators or functions. Throughout this term, I developed familiarity with the design process through troubleshooting, modifying and writing code for my assigned systems.

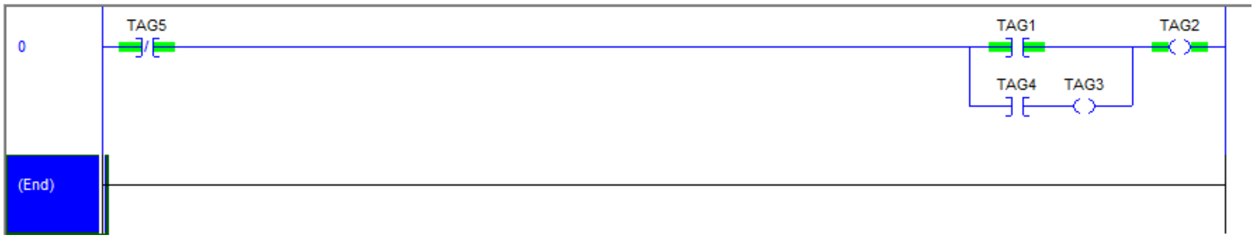

Figure 3: Generic example of ladder logic.

More complex PLC programs utilize object-oriented programming principles, giving instructions that provide more involved functionality, such as calculating data values or introducing time delay systems. This allows for considerable customization and modularity in the program. Additionally, every form of modern control system will generally make use of proportion-integral-derivative (PID) controllers, which I worked with and modified to fit the needs of the system code.

PID controllers manage the stability and response time of an automated system in an efficient manner, pulling feedback sensor data and running a series of calculations to determine ideal operation parameters. Automation processes are fully dependent on the principles of control theory, which sees application in the form of these PID-capable controller devices. They allow for smooth temperature transitions when enabling heating or cooling appliances, seeing wide use in environmental conditioning, and are capable of controlling motion response rates and positioning for actuation in robotics systems and autonomous aircraft avionics.

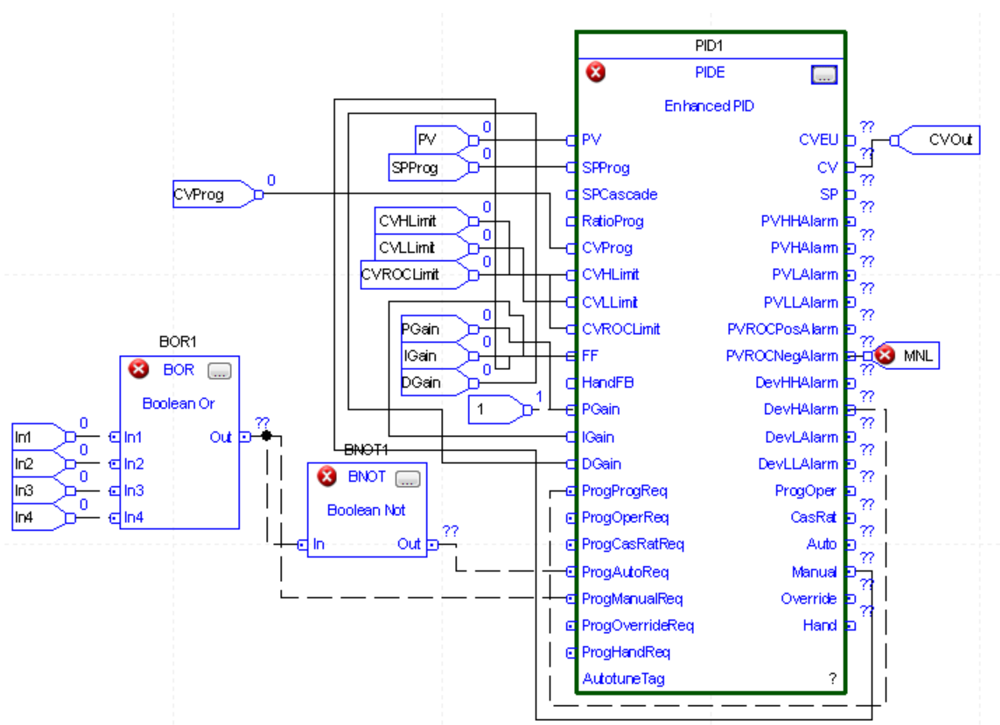

Figure 4: Generic example of a PID controller in the PLC program.

II. Project and Deliverables

A. Initial Testing

Learning the system began with the process of running test procedures against existing PLC code and HMIs. For this complex system, testing entailed thousands of steps across over a hundred screens. Each screen was tested against a specified procedure document, where each tag present on a display was required to have a procedure associated with it. Tests involved analyzing specific devices or elements present on a screen, and toggling the state of tags associated with the aforementioned elements. Tags in the PLC are distributed in a list that outlines their properties such as data type and description, and can be manipulated numerically in this list, or graphically in the ladder logic.

Name	Value	Style	Data Type	Description
TAG1	1	Decimal	BOOL	PLC tag 1
TAG2	1	Decimal	BOOL	PLC tag 2
TAG3	0	Decimal	BOOL	PLC tag 3
TAG4	1	Decimal	BOOL	PLC tag 4
TAG5	0	Decimal	BOOL	PLC tag 5
TAG6	33.5	Float	REAL	PLC tag 6
TAG7	732.1	Float	REAL	PLC tag 7

Figure 5: Tag management in the PLC program.

Successful passing elements were to fulfill a set of truth table cases, while any discrepancies or lack of functionality resulted in a test failure with detailed observations noted. These test procedures required meticulous attention to detail to ensure reliability of the software. Highly detailed documentation of outcomes and observations also gave a reference point for addressing problem areas. As the process progressed it was necessary to update any outdated steps, as this set of procedures would become our final testing documents for V&V and hardware functional testing. Final verification of the completeness of these procedures would come later, after a final and complete set of project PLC software was delivered and the HMI displays were ready to operate on hardware.

Step	Description	Acceptance Criteria								Pass/Fail	Comments
		Tag	Tag 1	Tag 2	Tag 3	Tag 4	Tag 5	Indicator Color	Indicator Value		
1	Device 1	Tag 1:{TAG_NAME1} Tag 2:{TAG_NAME2} Tag 3:{TAG_NAME3}	1	x	x			green	Condition 1	□PASS *FAIL	Device does not correctly transition into state 2 based on TAG_NAME2
			0	1	x			red	Condition 2		
			0	0	1			red	Condition 3		
			0	0	0			yellow	Condition 4		
2	Device 2	Tag 1:{TAG_NAME4}	0					yellow	Condition 1	*PASS □FAIL	
			1					green	Condition 2		

Figure 6: A generic example of the test procedures.

This initial round of testing identified key areas where modification was necessary. Due to the project deadline requirements, the test procedures at this point were a working draft and would need updating as well. The entire process of performing test steps involved identifying problem areas in the PLC code, the HMI and in the test procedure. As a result, this was an effective method of learning the intricacies of the system and intended functionality of the ECS process.

B. Display Work

Functionality problems compiled into the first round fix list gave us a starting point for areas that required work. Of the unique display files, each had an individual set of modifications that needed to be addressed. Due to limited documentation supporting the HMI development software, many fixes were produced through experimentation and creative design. Often the design was dictated by the limitations of the development software feature set, where the innovative use of the included tools was a strong skill to employ. Overview displays served to provide the operator with a broad perspective of the system as it ran, featuring control and data indicators throughout. Process control was operated through dedicated controls screens, including command buttons and set-point inputs. Each variety of commanding and set-point also required a form of two-step confirmation to avoid button press mistakes in field operation.

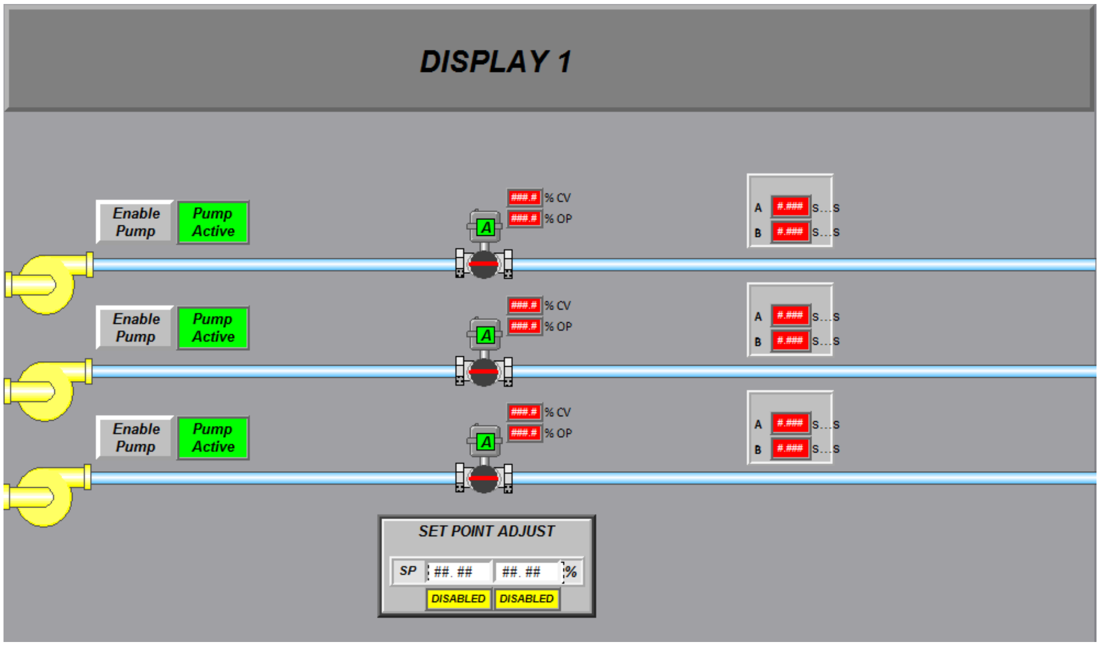

Figure 7: Generic example of an HMI display.

Displays for ECS were built using a particular design philosophy to expedite the development process. Extensive use of shared objects and displays were employed and configured to pull data from a set of parameters unique to each specific screen. This allowed a developer to implement changes to a single element on the screen, and that change would be reflected across all screens that utilized that same element. Additional complexity was added using this approach, but it drastically reduced the time when addressing repeat problems. A single valve object containing a specific set of parameters could be created once and shared in multiple different displays. The primary benefit from this process was seen when any modifications or functionality improvements were necessary, as all changes to that single object would propagate to all instances of it.

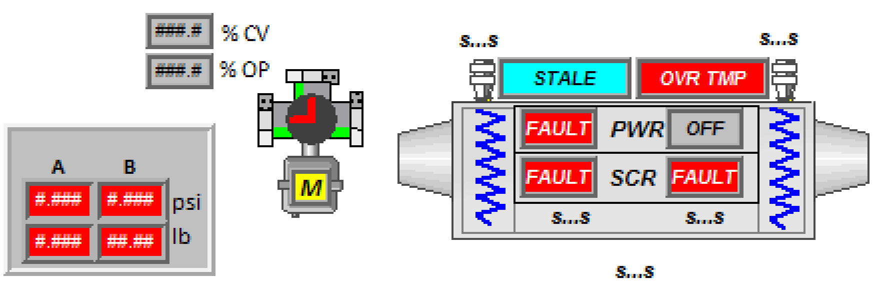

Figure 8: Generic examples of global objects.

Due to the complexity of the fixes and modifications, it was necessary for me to document a step-by-step guide to the more involved problem solutions to assist future personnel on the project. The utilization of parameter files and global objects, as well as shared template screens throughout the project introduced a significant learning curve to understanding the HMI software. Once properly trained with guidelines, team members were able to modify and create global objects that fit within the design philosophy. Despite the difficulty, the use of global objects and template screens consistently proved to be the most reliable and efficient method of development, simply requiring a small amount of procedural preparation for newcomers to the project.

To supplement the modification to displays, it was necessary to modify and complete subsets of the PLC code to ensure consistent functionality. I was given the task of building routines in a component of our system using electrical design documents. Developing PLC code intuitively branches off of technical design drawings for mechanical and electrical systems. This often involved using proprietary software tools used in creating code rungs and troubleshooting code issues.

C. Team Building

Throughout the course of the project, additional personnel were assigned to support the ECS launch pad software development to meet the increasingly pressing deadlines. Due to my experience working hands-on with the HMI and software development process over the course of the term, I was assigned the responsibility of training new members and coordinating work in the controls development lab. Team members were introduced in batch waves and trained initially through running test procedures. Through the testing process, it was possible to provide a new user and designer an in-depth glance at the system functionality as a whole, assisting them with their understanding of both software and hardware, and how each of the components were integrated together. When members completed test procedures and displayed aptitude towards the design process, they were reassigned to assist with the development and modification stage.

As the project grew, it became necessary to distribute members into smaller teams, each responsible for a particular subset of the system. Small teams of 2-3 members would test and modify specific displays, and update the testing procedures. Final completion of the project relied on the completion of several areas of work, which included the local and remote displays themselves, the PLC code, comprehensive test procedures and software requirements documentation. Personnel were organized based on skillset and background to address the needs of the project. Those with work experience in PLC and HMI design led the smaller teams, guiding the progression of fixes, testing and updates. Each sub team reported to a support team group that provided in-depth assistance with more complicated functionality modifications and insight on design philosophy and system operation.

To improve workflow, the project team made use of new and innovative processes to allow for automated data parsing, as well as more effective methods of version control in the design lab. Extensive scripting use was employed to perform deep dive analyses when assessing the completeness of test procedures as compared to the HMI displays and PLC code. By using automated scripts for pulling and analyzing tag data from displays and code, the team could reallocate time resources towards fixing displays and test procedures, rather than parsing manually. Additionally, through improvements in the file sharing and collaborative development software system, exchange of design files was streamlined. Project files were separated into individual archives under a project directory, allowing for individual checkout and documentation for modification. With this system in place, loss of work was prevented and documentation of revisions was improved significantly. This organizational structure provided small design teams the freedom and flexibility to work independently on their assigned component of the system, while still allowing for consistency through design as directed by supporting developers.

III. Conclusion

Noteworthy deliverables in this term were extensive in the scope of the project. A surge in personnel to address the growing list of modifications brought a significant increase in productivity in the development lab, where I acted as a team lead to organize workflow and train new members. The large volume of error and bug fixing performed by this team has improved the status of the displays to meet functional testing deadlines, and will result in a stronger group of HMI developers for ECS moving into the future. To meet future demands of HMI development, I have created documentation with the purpose of assisting NASA and contract personnel with troubleshooting problems with the displays and development software, in addition to providing starting points for implementation of complex components. Included in the aforementioned document are guidelines for effective global object use to optimize development time.

Knowledge and tools I have obtained in of this term have been passed to the team as they continue moving the project forward in preparation for a successful launch. We have built a team that will provide varied skills and expertise to KSC ground controls support structure, an integral component to the framework of successful missions to come with the SLS program. The development period of a project can be an arduous undertaking, and often the emphasis of testing, bug fixing and verification is underappreciated for its difficulty and rigor. With this team, I was given an opportunity to take a leadership role, which challenged me to be adaptable and creative in addressing problems as they arose.

Acknowledgments

I would like to provide my heartfelt thanks and appreciation to my mentor for the term, Terri White, who showed continued confidence in my ability to work independently, and challenged me to grow into a more effective team member and leader. As a project lead and mentor, she remained focused and dedicated to the progress of the project, and was an inspiration throughout with her uncompromised work and expertise. I would also like to extend my thanks to John Wilkas and Christian Eslin for their invaluable guidance and contributions to the project. Additionally, I wish the best to my development team members and appreciate their diligence, eagerness and openness. Finally, I thank the KSC Education Office for this incredible opportunity, which has helped me grow professionally in profound ways.