Energy Facility Contractors Group (EFCOG)
Software Quality Assurance Task Group

White Paper

Firmware Software Quality Assurance Considerations

WP-2019-SAF-QA-SQA-001

Revision 0

6/12/2019

Prepared By: _____ June 12. 2019
Sidney Ailes (Atkins Global) Date
Task Team Lead

Approved By: _____ June 12, 2019
Vicki L. Pope, EFCOG SQA Chairman Date

REVISION HISTORY

Document Version	Revision Date	Originator	Revision Description
Rev 0	3/19/2019	Sid Ailes	Initial draft
Rev 0	6/12/2019	Sid Ailes	Inclusion of comments

TABLE OF CONTENTS

1. INTRODUCTION

Many facilities across the complex have embedded software (firmware) in use as part of important to safety systems, structures, components (SSC) and Measuring and Test Equipment (M&TE). In many cases, the firmware is not accessible by the user but is delivered as part of SSC or M&TE. Traditionally, this firmware was considered part of the delivered system and the quality assurance requirements applied were limited to calibration or testing of the system. However, as more and more systems are delivered with middleware or configurable software where firmware once was installed, the need for software controls has increased. For example, Middleware now allows the user to modify pressure, flow and level solutions for Magnetic Flowmeters by modifying the system configuration baseline. The firmware or embedded software itself is not modified, but the user is able to select from pre-set features and adjust functionality by modifying configuration parameters. These user-configured parameters and data are then saved to ".bat" files or other memory that the firmware uses to perform the desired safety function. This paper will help the user to determine when software quality assurance controls are needed for firmware and middleware.

2. DEFINITIONS

Firmware: The combination of a hardware device and computer instructions and data that reside as **READ-ONLY** software on that device. Firmware refers to code that resides in non-volatile memory. Non-Modifiable Non-Configurable Firmware is delivered as an integral part of items, where the computer instructions and data can only be modified by replacement of the hardware device Firmware that can change features based on changes to a configuration baseline is considered configurable software. Non-Modifiable Configurable Firmware is delivered as an integral part of items, with a limited ability to adjust functionality by modifying configuration parameters, via a set-up process. Modifiable - Configurable Firmware is delivered as an integral part of items, where the computer instructions and data can be modified, including at run time.

Configurable Software: Software that is commercially available that allows the user to modify the functioning of the software in a limited way to suit user needs within clearly defined limits. Configurable software is an out-of-the-box solution that allows the owner to personalize certain aspects of the software themselves, without the help of experienced software developers. Configurable software is flexible, scalable and can be continually shaped through a human-machine interface to meet an organization's industry-specific and organization-specific needs.

3. SCOPE/BACKGROUND

The scope of this white paper is limited to software embedded in Systems, Structures and Components or Equipment that are cited in a Documented Safety Analysis (DSA) or safety analysis report. Computer Software which is developed or otherwise acquired for design and/or safety analysis of Department of Energy (DOE) Nuclear facilities is outside the scope of this white paper because the application of the software work activities identified in DOE Order 414.1D are clearly applicable. However, the application of the software work activities identified in the DOE Order to embedded software is not consistently applied across the DOE

Nuclear sites. For example, the embedded software in Fire Alarm Panels at some DOE sites are not considered safety software, while at other sites they are.

The very nature of the different applications of embedded software can result in an unnecessary burden if sites attempt to apply the software work activities from DOE O 414.1D to the management and control of embedded software, especially in SSC such as emergency generators, ovens, or stand-alone measurement and test equipment. Making the application of software work activities more difficult is the expanding use of embedded software in "expert systems." New instruments are being developed by suppliers that provide "middleware" that will allow users to "morph" safety systems and instruments into ever expanding applications. For example, modifiable-configurable firmware associated with the Rosemount 3095 MultiVariable Mass Flow Transmitter allows users to configure the detector to identify plugged impulse lines using statistical process monitoring technology and calculate a fully compensated Mass Flow. In these complex applications, the firmware associated with the transmitter must be controlled as a computer program. However, for the firmware embedded in the Fluke multimeter, where the range of use is selected by the user from pre-set features, the firmware is controlled as part of a calibrated system and no special controls are applied other than calibration of the M&TE.

As a result of the wide range of complexity, configurability and modifiability issues, determining the level of control that should be applied to firmware can be a daunting task. On one-extreme, the firmware that can be modified by the user to essentially reprogram a device function must be controlled as software. On the other extreme, applying software work activities, such as configuration control and software verification and validation to firmware installed on Read Only Memory (ROM) inside a multimeter would void the warranty on the instrument. This paper will provide guidance on appropriate levels of control for firmware.

4. APPLICATION OF STANDARD

ASME NQA-1-2017 allows three possible approaches for firmware:

- If the computer program can be changed after it is embedded, including at run time, all applicable controls of Subpart 2.7 should be applied
- If the computer program cannot be changed after it is embedded and testing of the completed device is not adequate for full acceptance, Subpart 2.7 software development controls should be applied
- If the embedded computer program functions can be adequately verified by testing the completed unit and the computer program cannot be changed, including at run time, without repeating this verification, controls beyond those used for hardware may not be necessary.

While these approaches seem straightforward, the multi-function of today's instruments make the application of the requirement challenging. For example, the M-Series Gas Flow Meters by Alicat provides pressure, temperature, volumetric flow and mass flow parameter measurements as shown in Figure 1. Pressing the button next to each parameter will put that parameter in the primary display window. Pressing the button, a second time, will change the engineering units and/or data feed (device units). Several preset menus are provided for User selection as shown in Figure 2 as well as a RS-232 interface. Because the firmware features and functions are pre-

set and can only be selected by the user, even with a computer connected, this firmware is non-modifiable and non-configurable, and the NQA-1 Subpart 2.7 requirements DO NOT apply. Software work activities would only apply to the external computer that collected or manipulated the data received. If the RS-232 interface allowed the user to set a control point (e.g., 120psi), then the firmware would be non-modifiable, configurable, and configuration controls would apply to document the parameter change, but only at the "system" level. If the RS-232 interface allowed the user to modify the program by downloading a new code to be executed by the device, then NQA-1 Subpart 2.7 requirements would apply.

A more challenging aspect of applying NQA-1 requirements to firmware are Fire Alarm Control Panels. Modern fire alarm systems are capable of detecting smoke and heat from a small flame, water flow in a sprinkler system or an activated pull station and reporting this information to on-site and off-site personnel. However, control panels vary from stand-alone panels with conventional annunciator connectors to addressable Data Panels with Digital Alarm Communicator/Transmitters (DACT) connecting multiple panels in different buildings and zones to a central panel. The DACT transmits system status (alarms, troubles, AC loss, etc.) to the central station via the public switched telephone networks and allows remote programming or interrogation of the control panel using utility software. In the former case, software controls would not be applicable because annunciators are hard-wired, and the alarm functionality is readily verified by fire system tests. However, the latter presents a more complex problem because remote programming from a laptop, without proper security protocols, could inadvertently change the annunciator settings such that an alarm no longer works

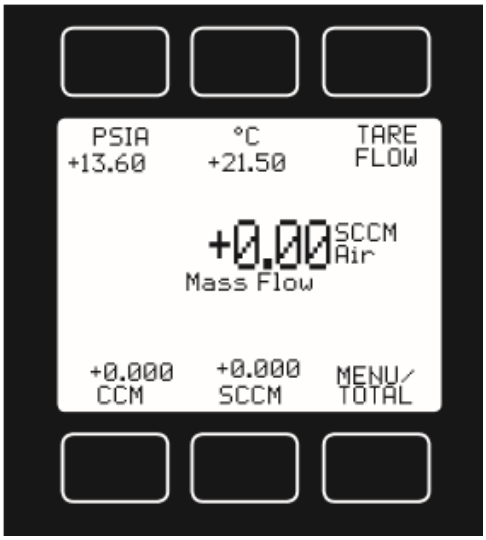

Figure 1. Flow Meter Data Screen

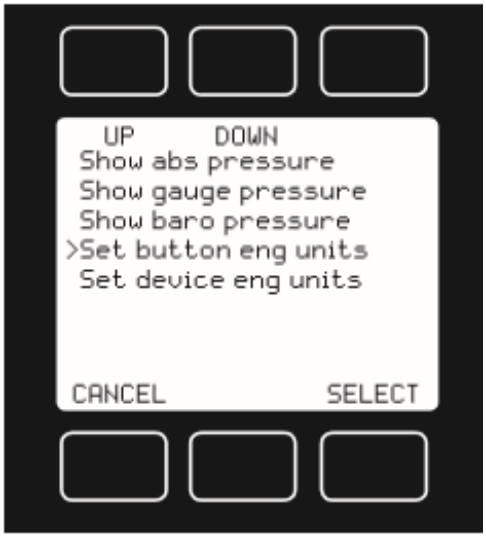

Figure 2. Flow Meter Function Selection Screen

The programming for most Fire Control Panels is limited to configuration of input zones, zone types, Notification Appliance Circuit (NACS), Relays, option modules (e.g., ANN-BUS) and system setup. All of these inputs can be accomplished by keypad entry through the front panel (see below). But the initial setup could involve hundreds of individual keypad touches [e.g., Mode, 2 (Programming Mode), Enter, 1 (Input Zone), Enter, 1 (Zone 1), Enter, 1, (Zone Type), Enter, 1 (PullStation), Enter, etc.]. The control panel system setup and device input selection

(e.g., annunciator, NACS, Relays, etc.) can be greatly simplified with less input error by entering data on a computer spreadsheet and then downloading to the panel using utility software provided by the manufacturer.

The problem is that the term "programming" as used in Fire Alarm Control Panel manuals is not coding in the traditional sense. The DACS "programming" does not involve modifying the computer program itself but configuring the panel within manufacturer specified settings to communicate with each device, annunciator, NACs and Relay in the fire control system. Once the panel is configured, system tests are performed by the Fire Marshal in accordance with NFPA-72. No additional controls are required because the acceptance of the fire alarm system, including the panel, is covered by NFPA-72. If the firmware or the DACS were to be removed or updated, the configuration of the system would require re-testing, but this would be under the auspices of the Fire Marshal, and not a software work activity.

The following sections will look at each software work activity defined in the DOE Order and provide examples of recommended strategies for firmware control for a variety of situations. For the purposes of the paper, all firmware is installed in systems or devices that perform a safety function.

5. SOFTWARE PROJECT MANAGEMENT AND QUALITY PLANNING

This work activity primarily applies to the procurement and development of software. Firmware in most applications is installed by the manufacturer with the delivered system or device. Although general project planning certainly applies to the procurement of the system or device, no firmware specific Software QA Plan is warranted unless the firmware contains volatile program memory that can be modified by the user. A software QA plan should be developed for modifiable firmware to ensure that required software work activities are planned and responsibilities and interfaces defined. Examples of each firmware type are indicated below:

- Non-configurable, non-modifiable firmware – Oven controller, diesel generator, pump

- Configurable, non-modifiable firmware – Fire Alarm Control Panel, Flow Meter, Spectrometer,
- Modifiable firmware – PLC, PAL, PLA, CPLD, FPGA,

6. SOFTWARE RISK MANAGEMENT

This work activity applies to firmware. The user must continuously assess what can go wrong, determine what risks are important to address, and implement actions to address those risks. Even though non-configurable, non-modifiable firmware has no user alterable inputs, the user must determine the impact if the device the firmware is installed in fails to operate correctly. For example, if the firmware in a new oven temperature controller fails to shut off at the selected temperature, and the resulting run-away condition could result in a danger to the public, the worker or the environment, then risk management should be implemented (e.g., Pu239 is pyrophoric on exposure to air and moisture).

7. SOFTWARE CONFIGURATION MANAGEMENT

7.1. Configuration Identification

The identification of configuration items pertaining to firmware is daunting for many systems today. Consider the firmware configuration items in a fire control system. Each alarm, sensor, annunciator, controller, transmitter and network bus contains firmware. For a large commercial building this can result in over a hundred individual configuration items associated with one control panel. In addition, the identification of the firmware as a configuration item is often problematic, because the system or device in which the firmware is installed may not display or otherwise identify the firmware version. Opening the device containing the firmware to verify physical identification will void the warranty on the device. In this situation, the firmware should be accounted for at the component level (e.g., annunciator, relay, NAC, etc.). Programming utility software provided by the control panel manufacturer can provide an efficient and effective method of capturing and documenting each component that makes up the fire control system.

7.2. Configuration Control

The traditional definition states that firmware is “permanent software programmed into a read-only memory.” This would indicate that configuration control does not apply to firmware. However, for most modern devices, the firmware includes electrically erasable programmable read-only memory (EEPROM) circuitry which allows for reprogramming. In fact, most manufacturers recommend “updating” firmware regularly to enhance functionality and security. The main difference between configurable and modifiable firmware is although the user may install the update, the program is not altered by the user, only downloaded from the manufacturer of the device and stored in flash memory. From the standpoint of Subpart 2.7, the modified software is still firmware if the update is from the manufacturer of the device. But if the firmware is reprogrammed (not configured), then the firmware would be considered software under the requirements of Subpart 2.7.

Change control for firmware starts with access control. Physical security as well as password security is important to ensure unauthorized changes are not made. For configurable firmware, parameter changes should be logged, and the system retested to ensure the device works as intended. For complex systems such as a Fire Alarm Control Panel (FACP), NFPA 72 requirements require a certified technician be at the control panel whenever the programming utility is used to download any configuration information to the FACP. Changes to the firmware must be tested to ensure changes do not adversely affect other COMPONENTS IN THE SYSTEM.

7.3. Configuration Status Accounting

Configuration Status Accounting for non-configurable, non-modifiable firmware is limited to identifying systems or devices that include firmware, if known. For configurable, non-modifiable firmware, the operations log can be used to record parameter changes. For configurable, modifiable firmware, subpart 2.7 requirements would apply and controls similar to an Engineering Change Notice would be required to ensure the firmware change does not adversely affect the other components in the system.

7.4. Configuration Audits and Reviews

Configuration audits and reviews for non-configurable, configurable, non-modifiable firmware is not applicable because such firmware is not accessible except by the manufacturer. For modifiable firmware, subpart 2.7 requirements would apply and controls similar to code walkthroughs would be required to ensure the firmware change meets the specified requirements.

8. PROCUREMENT AND SUPPLIER MANAGEMENT

In many cases when devices or instruments containing firmware are purchased, the "requisitioner" may not even be aware that firmware is included. Even when the purchaser is aware that firmware is involved, technical and quality requirements are limited to the system or device being manufactured. If custom firmware such as PLC or FPGA is being developed as a service under a purchase order to meet the purchaser specified requirements, then the requirements of subpart 2.7 would apply and controls similar to the procurement of custom software would be required, including performing an audit of the supplier.

9. SOFTWARE REQUIREMENTS IDENTIFICATION AND MANAGEMENT

As indicated in section 8.0 above, firmware is typically incorporated by the manufacture of systems or device based on system requirements. The manufacturer would normally develop system level and component requirements that result in a catalog item description or operations manual that is published. The requirements for the individual components that make up the device, including firmware, would normally be considered proprietary. Most devices, such as an auxiliary diesel generator or a programmable flow meter will not have requirement specifications for the firmware. Instead the manufacturer will provide general specifications for the device. As a result, system level requirements should address functional, performance, security, interface

and safety requirements that the system must meet. However, if site- or project-specific performance specifications are provided with the purchase order, the manufacturer will charge more for a custom product! For modifiable firmware, subpart 2.7 would apply and the developer would need to develop requirement specifications to ensure the firmware performs as intended. For Programmable Logic Controllers, the Software Requirements Specification may be met through the Process and Instrumentation Documentation (P&ID) and Mechanical Sequence Diagrams (MSD) or similar documentation. For Fire Alarm Control Panels, the software requirements will include a written narrative providing system description and floor plan layout showing locations of all devices, control equipment, and supervising station and shared communications equipment.

10.SOFTWARE DESIGN AND IMPLEMENTATION

As indicated in section 9.0 above, design requirements for non-configurable, configurable, and non-modifiable firmware contained in a commercial product (e.g., diesel generator, oven controller) would normally be considered proprietary. For modifiable firmware, subpart 2.7 would apply and the developer would need to develop design documentation, implement the design to established coding standards, and perform unit testing and stress testing to ensure design integrity. For Programmable Logic Controllers, the Software Design Specification requirement may be met through Input/Output lists and control logic diagrams which are the basis for PLC programming. For Fire Alarm Control Panels, the design documentation will include a description of the sequence of operation as well as the manufacturer's published instructions.

11.SOFTWARE SAFETY ANALYSIS AND SAFETY DESIGN METHODS

Software safety analysis would apply to firmware used to control a safety system. If the manufacturer provides a system controlled by firmware that performs a safety function, such as a pump or valve, then methods to mitigate the consequences of system failure. Potential failure modes need to be identified and evaluated for their consequences of failure and probability of occurrence. Because non-configurable, configurable and non-modifiable firmware provided with commercial devices may not be tested separately from the device, consequence analysis should identify test problems that will evaluate the performance and error tolerance of the system. For modifiable firmware, the software design should be evaluated for simplicity, decoupling and isolation to eliminate or mitigate hazards.

12.VERIFICATION AND VALIDATION

Verification and Validation of firmware is heavily dependent upon the system or device that the firmware communicates with. Although reviews and inspections of software documentation for modifiable firmware will help to ensure design integrity, ultimately the critical characteristic is the performance of the system. For non-configurable and non-modifiable firmware verification and validation activities are limited to testing of the complete system because nothing is known about the design of the firmware. In this case, testing is similar to black box testing for software, except the output is the performance of the system. For configurable firmware, verification

activities are limited to reviews or inspections of the final product and associated parameter listings. For modifiable firmware, subpart 2.7 testing activities would apply, although unit testing may not be possible as the output of the firmware is dependent on the system or device performance. For example, verification and validation of PLC software results must be performed in a mock-up of the real system using the same hardware as will be used in the operational environment. Acceptance testing work activities should be formally planned and documented, acceptance test cases and procedures, including expected results should be created, test results should be documented and test results independently verified against specified acceptance criteria.

13.PROBLEM REPORTING AND CORRECTIVE ACTION

Because of the difficulty in identifying firmware configuration items, especially for non-configurable and non-modifiable firmware, the problem reporting and corrective action process should focus on nonconformance control for the overall system. For configurable and modifiable firmware, the problem reporting and corrective action process should address the appropriate requirements of the site corrective action system. The corrective action system will cover 1) methods for documenting, evaluating and correcting firmware problems; 2) an evaluation process for determining the impact of the problem; and 3) the roles and responsibilities for disposition of the nonconformance reports.

14.TRAINING OF PERSONNEL

The breadth of firmware use by manufactures of systems and devices with widely different applications that are not transferable outside of the specific manufacturer's product makes defining training requirements difficult. For example, only fire safety technicians certified in accordance with NFPA requirements should configure and test Fire Alarm Control Panels. However, ASME NQA-1 requirements for establishing a formal training program for personnel configuring, testing, updating, modifying and operating safety systems should be implemented. Such training should be sufficient to adapt to changes in technology, methods or job responsibilities. Where formal on-line or classroom training is not available, on-the-job training requirements will be defined so that personnel receive the experience needed to achieve and maintain proficiency in the assigned task.

A1. TEAM MEMBER BIOGRAPHIES

Sidney Ailes (Atkins Global), Task Team Lead

Lance Abbott (SRS)

Carl Wharton (SNL)