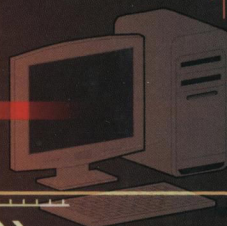
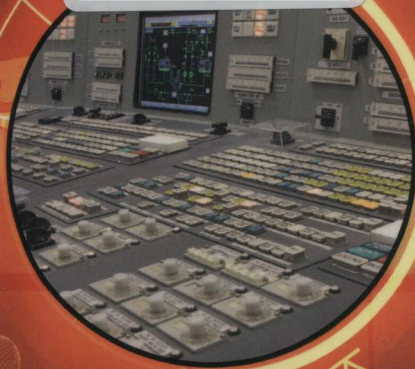


**ELSEVIER INSIGHTS**



# DESIGNING SCADA APPLICATION SOFTWARE

A PRACTICAL APPROACH

STUART G. McCRADY

# Designing SCADA Application Software

## A Practical Approach

**Stuart G. McCrady**

*Technical Instructor in SCADA Technology, ON, Canada*



AMSTERDAM • BOSTON • HEIDELBERG • LONDON • NEW YORK • OXFORD  
PARIS • SAN DIEGO • SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Elsevier

32 Jamestown Road, London NW1 7BY, UK  
225 Wyman Street, Waltham, MA 02451, USA

First edition 2013

Copyright © 2013 Elsevier Inc. All rights reserved

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangement with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: [www.elsevier.com/permissions](http://www.elsevier.com/permissions)

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

### Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

### British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

### Library of Congress Cataloging-in-Publication Data

A catalog record for this book is available from the Library of Congress

ISBN: 978-0-12-417000-1

For information on all Elsevier publications  
visit our website at [store.elsevier.com](http://store.elsevier.com)

This book has been manufactured using Print On Demand technology. Each copy is produced to order and is limited to black ink. The online version of this book will show color figures where appropriate.



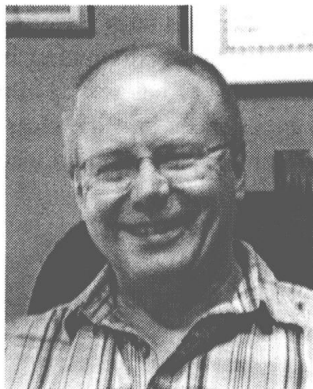
Working together  
to grow libraries in  
developing countries

[www.elsevier.com](http://www.elsevier.com) • [www.bookaid.org](http://www.bookaid.org)

# **Designing SCADA Application Software**



## About the Author



Stuart McCrady is a Certified Engineering Technologist in the field of Electronics and Physics Engineering, as well as a Certified Professional Educator in the field of technical training for adults. He spent the first two years of his career in electronics, installing and servicing both large mainframe computer systems and small minicomputers. He then shifted to software programming in automation systems. This field of minicomputer programming required developing application software in machine or assembly language, executing at the hardware level. Field devices such as limit switches, pushbuttons, and solenoid valves, were connected

to custom designed hardware interface boards installed inside the minicomputers. From the minicomputers of the 1970s to the PLCs and HMIs of today, Stuart has worked with a broad range of technologies using a variety of hardware and software platforms. He was involved in the design and implementation of more than 50 SCADA type projects. As his career progressed, Stuart acquired both more experience and more responsibility in the field of system integration and SCADA systems consulting. Stuart has served as programmer, project leader, project administrator, consultant, department manager, and SCADA system designer.

Throughout his career, Stuart strove to establish programming standards and design methodologies that could be applied to any SCADA application. In the mid-1970s, he developed a program design and documentation system which he called FLOCODE, which resembled high level languages such as C, but was written in plain English. The purpose of the system was to allow the programmer to design software using English-like statements using structured programming constructs such as: If-Then-Else and Do-While/Do-Until. He applied this method to his own programming at both the high level language and the machine level language; this design documentation then became the comments and program description once the program was completed.

Later, Stuart was involved in the establishment of a systematic tagging system for signal names which would work both for hardware signals as well as internal software points. In addition, a system of structured descriptions was developed for the PLCs which described the operations in simple English, but referenced key signals and operating parameters; this documentation served as the design document for the PLC programming. Stuart expanded this combination of systems into the complete design and documentation system which is the focus of his book: 'Designing SCADA Application Software: A Practical Approach'.

In addition to this book on Designing SCADA Application Software, Stuart has published articles in trade magazines, as well as presented a paper on the application of computer control systems in the water treatment plants at an American Water Works Association convention.

In 2006, Stuart made another shift in his career, becoming a full time instructor/trainer. Stuart taught courses at the community college level and at the trade school level; courses included: electronics, residential wiring, digital logic circuits, communication networks, electro-pneumatic control systems, electrical motors and motor control circuits, and PLC programming. Since 2011, Stuart has been traveling throughout Canada and the United States, teaching PLC and HMI programming in cities across both countries. His extensive experience in the industry has served him well in the classrooms, as he is able to bring real world experiences into the classroom such that the students not only understand the programming material, but also understand how the concepts are applied.



# Preface

Today's SCADA systems incorporate Programmable Logic Controllers (PLCs), Human-Machine Interface (HMI) workstations, and network communication systems into a complete integrated system. Each of the major components requires one or more form of programming from program logic to configuration to process graphic displays to communication configuration. While there are books on the market which teach PLC programming, there is a need for a comprehensive book that addresses the programming involved in all parts of a SCADA system.

Typically, the books available on PLC programming offer the reader detailed explanations of each of the instructions available for the particular brand of PLC. Short code sections are presented to illustrate how the instructions are used and how they function. Unfortunately, there is nothing in these books which explains to the reader how to organize and design the application software as a complete project, how to structure the database points, and how to document the program logic design. In short, there is a need for something that explains not only how SCADA application software should be developed, but also that explains how to create the type of software documentation required by today's customers.

Over a period of 35+ years, I have been involved in all aspects of SCADA systems: consulting, system design, programming, commissioning, training, and documentation. As a result of working on more than 50 such projects, a number of simple and efficient methods have evolved for designing and programming these systems. By following these methods, one can efficiently design, program, and implement the programming of all of the aspects of the SCADA system. A simple yet effective documentation system is used to ensure that not only is every part of the system designed properly, but also a thorough and complete documentation set is produced.

It is interesting that as I have been championing standards for software design and documentation, that more and more customers are establishing their own standards, all of which are based upon the ones that I have been using. It is rewarding to see many of the customers for whom I did SCADA programming have now created their own software standards, which are built upon those standards and concepts which I helped the customers to adopt. This is not to say that I single handedly caused these customers to follow my standards, but rather that I and others have established the need for good documentation, and that many of the standards which I used for these customers have now evolved into well-established standards. Anyone doing programming for these customers must adhere to their standards, and it is these standards that are presented in this book.



SCADA systems are global; every automated system in the world involves some form of process controller and user interface facility. These systems can range from a single controller that monitors and controls a small set of processes with a single workstation for user interfacing, to large geographical systems of controllers, user interface workstations, server computers, and both local and system wide communication networks. The user interface, in the form of animated process graphic displays, present current and historical operating information to the user, allowing users to observe system operations, as well as interact to effect control actions to the system. This user interface is functionally the same everywhere, regardless of the country; only the language would differ to suit the country.

The International Electrotechnical Commission, or IEC, has established a standard for programming of the field controllers, which has been adopted by most major controller manufacturers. This standard, IEC 61131-3, has defined five languages in which the controller programming can be done; a brief explanation of each language is provided in this book. In Canada and the United States, the controllers are typically programmed in 'Ladder Logic', which closely resembles electrical wiring diagrams. In Europe, a machine level language, 'Instruction List' or 'Machine Language', is the preferred language for field controllers.

Speaking personally, I have programmed with three of the five languages. The goal of this standard is to make the controller programming independent of the manufacturer; if one can program in Ladder Logic on one manufacturer's controller, then he or she can do so on another manufacturer's controller. When manufacturers adhere to these programming standards, then programmers can 'port' their software design across platforms; there will always be some changes and modifications required, but the essence of the program logic that was developed can be applied to any controller. One of the benefits of this standardization is that the system integrator does not have to learn a new language each time they switch to another manufacturer's product.

As an aside, the IEC is one of three key organizations which establish such standards; the other two are the International Telecommunication Union (ITU) and the International Organization for Standardization (ISO). An example of an ISO standards that is used quite commonly is the 7-layer model for communications; this standard has been adopted by most manufacturers for the interface driver communications. These three organizations work closely to ensure that any standard created works seamlessly across systems.

The methods and techniques presented in this book are based upon my own experience working in the Canadian market; but these methods apply to any system in any country. Every automation project needs software designed and developed for that particular application, and software documentation for this software is imperative. These methods of designing and developing application software are global, and can be applied to any project involving process controllers and user interface software systems, anywhere in the world. So regardless of location in the world, automation systems require proper software design with complete documentation.

This book is about how to design and develop application software for SCADA systems. Starting with the first chapter, the need for programming standards is

established by explaining the longevity of SCADA systems, and therefore the need to develop a design which can be carried through the changing technologies. The remaining chapters then address the elements of SCADA software from the perspective of what is being designed and how it should be designed and developed.

Following is a brief summary of each of the chapters of this book, so as to provide the reader with an overview of the material in this book, and serve as an introduction to the material presented.

### **Chapter 1: Introduction**

This chapter provides an explanation of SCADA software with respect to the intent of this book. This chapter highlights the importance of developing application software as part of the complete software development process. It also indicates that software documentation can be created 'after the fact' in a reverse engineering fashion, so that any system can be documented.

### **Chapter 2: The Elements of SCADA Software**

This chapter identifies the various aspects of application software found in today's SCADA systems. As will be seen, a SCADA system involves different programming facilities at all of the levels of the system. The software is not just the field controller program listing, but includes database documentation as well as user reference material and descriptions of the operations.

### **Chapter 3: Practical Procedures for SCADA Software Development**

This chapter offers a guide on how to design and develop the software for a typical SCADA project. From identifying the physical inputs and outputs of each controller, to the creation of the software documentation, to the final testing and checkout of the SCADA system, this chapter presents methodologies which have been used many times on many different projects. Each of the procedures is then covered in detail in following chapters.

### **Chapter 4: Documentation for SCADA Systems**

This chapter presents a detailed explanation of the documentation that every SCADA project should have. Beyond the controller program listings, this chapter explains the purpose and content of the documentation that is produced as a result of good development procedures, as outlined in the previous chapter. Each of the documentation sections is then addressed in detail in subsequent chapters.

### **Chapter 5: Tagnames and Signal Naming Conventions**

This chapter offers a system of naming database points which allows for a structured approach to tagnames. A tagging system is needed to establish a unified naming convention for both the field I/O signals and the internal program points used in the software. The method presented can be modified and tailored to the reader's specific application.

### **Chapter 6: Developing the Application Program Databases**

This chapter presents a methodology of using spreadsheets to organize and document all of the points in the databases for both the field controller and the user workstation software. By applying the unified tagging system presented in the previous chapter, a consistent database can be created for all parts of the application software. Once the project has been completed, these same spreadsheets serve as database documentation.

### **Chapter 7: Process Control Logic Descriptions**

This chapter provides a detailed explanation of every operation performed by every application program in every field controller. As this document, together with some of the initial database spreadsheets, form the design documents, a complete design

document set results that can be used by the application programmer to develop the actual programs for the field controllers. These documents also serve as 'Shop Drawings' in contracts, as they clearly explain the design intent of the software before the detailed programming begins.

#### **Chapter 8: User Operations Reference Manual**

The manual in this chapter is needed for the user of the system; that is, for the plant operators and supervisors. Sections in this document describe the process operations available to the user, including the control functions and setpoint parameters that can be entered. Historical trends and reports are explained including how to extract data from a trend into a spreadsheet file for transfer to other parties. Any operation that a user can perform and any alarm or abnormal condition that may arise are addressed in this manual.

#### **Chapter 9: Guidelines for Controller Application Programming**

This chapter presents structured approaches to organizing the program logic for the field I/O controllers. While the logic can be kept entirely in one program file, a modular approach using subroutines results in a better organized and easier to understand program. Controller software today allows for not only multiple routines within a program, but also multiple tasks, each with multiple programs and routines. This chapter presents an approach that can be applied to almost any SCADA application, making the most efficient use of the project structuring features of the software.

#### **Chapter 10: Guidelines for Workstation Application Programming**

This chapter is similar to the chapter on the controller, but addresses the design of the Human-Machine Interface and the organization of the process and historical databases, along with many other aspects that relate to a well-designed user interface. The overall structure of displays, colour conventions, alarm handling procedures and trend displays are addressed.

#### **Chapter 11: System Integration, Commissioning and Checkout**

This chapter offers procedures or methods to implement application software for a SCADA system in a stepwise manner. Once the software has been developed, it is ready to test. But often the actual field controller has not been fully wired and/or all of the process equipment has not been installed and checked. The application software is always the last part of a project to be implemented, since it requires all of the components of the system to be installed and tested. The approach in this chapter presents a method that allows for the incremental testing and checkout of the application software.

#### **Chapter 12: A Sample Project – Applying the Principles**

This chapter presents the development and design of a complete SCADA application, including the field I/O controller program and the SCADA workstation software. All of the concepts presented in the book are applied to show how all of the material presented can be applied to a typical programming project.

#### **Appendix A: Glossary of Technical Terms**

This glossary offers a summary of acronyms and terms used throughout this book. Most of these terms are common in the SCADA systems industry.

#### **Appendix B: TSNC Dictionaries**

The Tagname and Signal Naming Convention described in Chapter 5 is illustrated here with a sample set of tagname fragments from which tagnames can be constructed. The tagname system proposed uses a series of five fragments of phrases to describe each part of the tagname; example fragments are listed here for a water treatment plant, to serve as an illustration of the tagname dictionaries referred to in Chapter 5.

**Appendix C: Sample Process Control Logic Description**

A sample PCLD is provided here to illustrate the application of the concepts presented earlier in Chapter 7, showing a real-world example for a sample PPC. All of the sections of a PCLD are included here for this one PPC system.

**Appendix D: Program Listings for PPC Program**

The program listings for the sample project are included in this appendix; the project is described in Chapter 12 for a pumping station. All of the program logic for all routines are shown, to provide a real-world example of a complete PPC control program.

Now that the overall content has been described, please read on to learn how to design, document and program the various parts of today's SCADA systems.

**Stuart G. McCrady**



# Contents

<b>About the Author</b>	<b>xi</b>
<b>Preface</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 SCADA: Convergence of Evolving Technologies	1
1.1.1 Early Automation Systems	1
1.1.2 The Human Interface	2
1.1.3 Communications and Integration	2
1.2 Basics of SCADA Signal Processing	3
1.3 Defining the Scope of SCADA Software	4
1.4 Use of Generalized Terminology	5
1.5 The Need for Programming Standards	6
1.5.1 Need for Design Standards	6
1.5.2 Modify or Reprogram	7
1.6 The Importance of Software Documentation	7
1.6.1 Lifetime of Software Systems	8
1.6.2 Upgrades and Revisions	8
1.7 Purpose and Overview of This Book	9
1.7.1 Organized Design and Development	9
1.7.2 Complete Software Documentation	9
1.7.3 Adherence to Programming Standards	10
<b>2 The Elements of SCADA Software</b>	<b>11</b>
2.1 Understanding the Elements of SCADA Software	11
2.2 Typical SCADA System Architecture	12
2.2.1 Field Devices and Signals	13
2.2.2 Programmable Process Controller	14
2.2.3 SCADA Operations User Workstation	15
2.2.4 Communication Network	15
2.2.5 SCADA Server Computer	18
2.3 Sample Application: WTP SCADA System	19
2.3.1 Low Lift or Raw Water Station	19
2.3.2 Pretreatment	19
2.3.3 Filtration	20
2.3.4 High Lift or Treated Water Station	21
2.3.5 Chemical Systems	21
2.4 Getting the Most from Field Data	21

<b>3</b>	<b>Practical Procedures for SCADA Software Development</b>	<b>25</b>
3.1	Life Cycle of a SCADA Project	25
3.1.1	Initial SCADA Project Specifications	26
3.1.2	Development of SCADA Project	26
3.1.3	Factory Testing and Demonstration	27
3.1.4	Commissioning and Site Acceptance Testing	27
3.1.5	The Final Step – Documentation	28
3.2	Identifying Process Area Field Signals	29
3.2.1	Process Areas and PPCs	29
3.2.2	Identifying Input and Output Signals	29
3.2.3	Organizing Field I/O Signals	31
3.2.4	Allowing for Future Expansion	31
3.3	Creating and Documenting Application Databases	31
3.3.1	Developing a Standardized Tagname System	32
3.3.2	Creating Points Lists Using Spreadsheets	32
3.3.3	A Spreadsheet Illustration	33
3.4	Defining and Documenting the Process Controller Operations	33
3.4.1	The Initial Process Narrative	33
3.4.2	Developing the Detailed Process Logic Descriptions	35
3.4.3	Documenting System Operations	36
3.5	Designing and Developing the SCADA Application Software	36
3.5.1	Developing the Controller (PPC) Application Software	36
3.5.2	Developing the SCADA Workstation (SOW) Application Software	37
3.5.3	Networking and Interfacing	38
3.6	System Integration and Checkout	38
3.6.1	Controller I/O Signal Checkout	38
3.6.2	Testing and Commissioning Process Area Operations	38
3.6.3	System Acceptance Test Procedures	38
3.7	User Operations Reference Manual	39
3.7.1	Purpose of the User Operations Manual	39
3.7.2	Structure of the User Operations Manual	39
<b>4</b>	<b>Documentation for SCADA Systems</b>	<b>41</b>
4.1	SCADA Software Documentation	41
4.1.1	Reasons for Developing Documentation	42
4.1.2	Components of Software Documentation	42
4.2	Database Reference	42
4.2.1	Standardizing on Tagname Conventions	43
4.2.2	Programmable Controller Database	43
4.2.3	SCADA Workstation Database	44
4.3	Process Control Logic Descriptions	44
4.3.1	Control Strategy Overview	45
4.3.2	Equipment and Parameters	45



4.3.3	Control Logic Descriptions	45
4.3.4	Special Considerations	45
4.4	Controller Application Program Listings	46
4.5	SCADA Workstation Application Software Reference	46
4.6	SCADA User Operations Reference	47
4.6.1	General System Overview	48
4.6.2	System Graphic Displays	49
4.6.3	Process Graphic Displays	49
4.6.4	Historical Reports and Trend Displays	49
4.6.5	Special Operating Procedures	50
<b>5</b>	<b>Tagnames and Signal Naming Conventions</b>	<b>51</b>
5.1	Original Signal Tagnames	51
5.2	Purpose of Standardized Tagnames	52
5.3	Constructing Tagnames with Phrases	53
5.4	Tagname and Signal Naming Convention Structure	54
5.4.1	Structure of the TSNC Tagname	54
5.4.2	Location Fragment	55
5.4.3	Equipment Fragment	55
5.4.4	Component Fragment	55
5.4.5	Signal Type and Designation	56
5.5	Tagname Fragment Dictionaries	56
5.6	Illustration of Tagname Application	58
<b>6</b>	<b>Developing the Application Program Databases</b>	<b>59</b>
6.1	Review of Data Types and Databases	59
6.1.1	Raw Field Signal Data	59
6.1.2	PPC Data Types	60
6.1.3	SOW Program Data Types	61
6.2	Using Spreadsheets to Create Points Lists	61
6.2.1	Basic Spreadsheet Template	62
6.2.2	Populating Points List Spreadsheets	62
6.2.3	Creating Spreadsheet Workbook Files	64
6.3	Developing the Controller Database	64
6.3.1	Hardware Field I/O Signals	67
6.3.2	Software Program Points	70
6.3.3	Application of Program Points in Logic	71
6.3.4	Internal Program Support Points	72
6.3.5	Procedures for Creating Controller Spreadsheet Database	74
6.3.6	Sample Controller Database	79
6.4	Developing the SCADA Workstation Database	81
6.4.1	Controller Related Data Points	83
6.4.2	SCADA Workstation Database Components	84
6.4.3	Procedures for Creating Workstation Spreadsheet Database	85

<b>7</b>	<b>Process Control Logic Descriptions</b>	<b>87</b>
7.1	Purpose of PCLDs	87
7.2	Structure of PCLDs	88
7.3	System Control Strategy Overview	88
7.3.1	Overview of Process Areas	88
7.3.2	Summary of Operational Concepts	89
7.4	Facilities and Parameters	89
7.4.1	Summary of Equipment	89
7.4.2	Key Process Signals	90
7.4.3	Control Parameters and Setpoints	90
7.5	Control Logic Descriptions	90
7.5.1	General Requirements	90
7.5.2	Local Control Mode	91
7.5.3	Remote Manual Control Mode	91
7.5.4	Remote Automatic Control Mode	92
7.6	Special Considerations	92
7.6.1	Software Interlocks	92
7.6.2	Hardware Interlocks	93
7.6.3	Failures and Alarms	93
7.6.4	Software Interfacing	93
7.6.5	Additional Considerations	94
7.7	Sample PLCD for Pump Station	94
<b>8</b>	<b>User Operations Reference Manual</b>	<b>95</b>
8.1	Structure and Purpose of User Reference Manual	95
8.2	System Overview and Introduction	96
8.2.1	SCADA System Overview	96
8.2.2	Central Control Facilities	96
8.2.3	Display Conventions and Navigation	97
8.2.4	Security Levels and Passwords	98
8.3	System Graphic Displays	99
8.3.1	System Overview Displays	99
8.3.2	Alarms and Events Displays	100
8.4	Process Graphic Displays	100
8.5	Historical Reports and Trend Displays	103
8.5.1	Trend Display Operations	103
8.5.2	Historical Reports	103
8.5.3	Exporting Historical Data	104
8.6	Special Operating Procedures	104
8.6.1	Startup and Shutdown Procedures	104
8.6.2	System Backup Procedures	105
<b>9</b>	<b>Guidelines for Controller Application Programming</b>	<b>107</b>
9.1	Identifying the Controller Processes	107
9.1.1	Isolate and Define Processes	107
9.1.2	Organize Processes for Programming	108