

OPALARM

Acknowledging Alarms in a Siemens Text OP

with a Subroutine Written for a Siemens S7-200 PLC

An Automation List PLC Archive Document

- **Author:** Michael Griffin
- **Document Date:** 23-JUL-2000
- **Date Originally Written:** 1999
- **Type:** PLC subroutine.
- **PLC Model:** Siemens S7-22x series (used with Siemens text OP).
- **Application History:** 3 systems in current use.
- **Known Problems:** None.
- **Current Revision:** Original version (adapted from S5 version).
- **Algorithm Source:** The algorithm was created for this application.

This document describes a simple method of acknowledging alarms in a Siemens text OP (operator panel with display and keypad). These OPs include the OP7, and OP17. OP "alarms" are an easy method of displaying and acknowledging faults. This version of the subroutine was translated from an older S5 function block which has been used in numerous applications.

Alarm messages are entered into the OP using the appropriate software (e.g. Protocol). When the PLC program sets a bit which has been defined as being part of an alarm word (which may be a flag word or data word), the OP displays the appropriate alarm message. When the operator presses the "Alarm Acknowledge" key on the OP, the OP sets an "alarm acknowledge bit" in the PLC which corresponds to the particular alarm which was acknowledged. At this point it is up to the PLC program to reset the original alarm bit.

Each alarm (fault) requires one alarm bit, and one alarm acknowledge bit. This means that one alarm word can be used to trigger up to 16 independent fault messages. The S7-200 PLC series also allows byte and double word addressing, but the 16 bit word size was found to be the most convenient for this application.

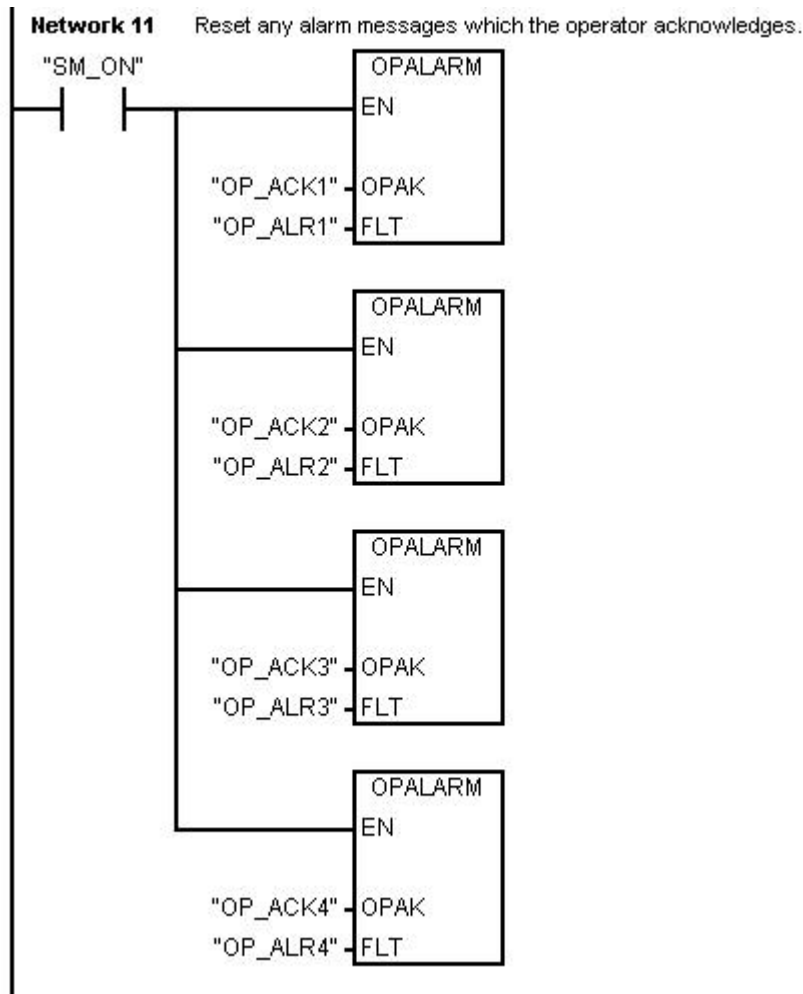
This subroutine makes this task easy by automatically resetting alarm bits when the corresponding alarm acknowledge bit is set. The algorithm used works on whole words at once, allowing up to 16 alarms to be individually handled by one subroutine call. This subroutine can be called as many times as necessary with different parameters to handle additional alarms.

Application Example

The following is an example showing the subroutine being called. Notice that it is being called unconditionally (the "SM_ON" contact is a system bit which is always true).

- **OP_ACK1** is the "Alarm Acknowledge word". This is the word the OP uses to signal when an alarm has been acknowledged.
- **OP_ALR1** is the "Alarm Word". This is the bit the PLC sets to trigger the display of an alarm on the OP.

This subroutine can be called as many times as necessary (as shown here), but of course different parameters must be used with each instance. I typically put these calls into a subroutine which contains other miscellaneous OP functions.



Algorithm

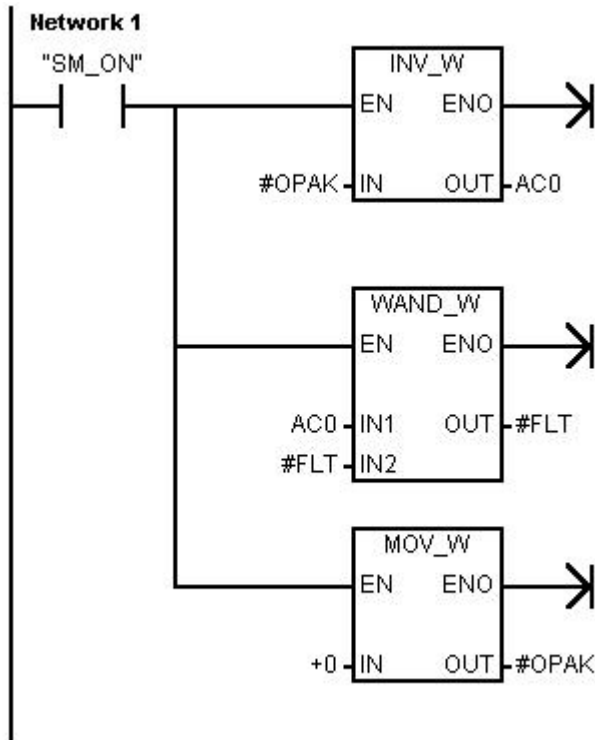
The following is the actual statement list source code for the subroutine. The algorithm consists of the following:

- 1) Invert (complement) the OP alarm acknowledge word.
- 2) AND it with the fault word.
- 3) Reset the alarm acknowledge word to zero.

Subroutine Source Code

OPALRM - SBR63. For use with Siemens OP displays. Reset any alarm messages which the operator acknowledges on the OP. When the operator presses the alarm acknowledge button, the OP sets a bit in the PLC which corresponds to the alarm which was acknowledged. The algorithm used is to complement (invert) the OP acknowledge word, and then AND it with the word which originally triggered the alarm. This block then resets the OP acknowledge word (by writing zero to it).

Addr	Name	Var Type	Data Type	Comment
	EN	IN	BOOL	
LW0	OPAK	IN_OUT	WORD	The word the OP uses to acknowledge alarms.
LW2	FLT	IN_OUT	WORD	The word which was used to trigger the alarms.



Note: *I have not experienced any problems with the program code in this document in my own applications so far. I have provided this example for illustrative purposes only. You are free to use the information in this document for your own purposes, but in doing so you must of course accept complete responsibility for any problems, bugs, unintended consequences, etc. you may encounter. In other words, use at your own risk.*