Honeywell

TPS R500—R530
US Implementation

Build Custom Alarm Schematic

L53435T
LCN

TotalPlant®

Copyright, Notices, and Trademarks

Revision 08 – February 11, 1998

Honeywell IAC courseware is subject to change without notice.

Honeywell
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85023

1-800-852-3211

Table of Contents

Figures and Tables

References

Publication Title	Publication Number	Binder Title	Binder Number
Picture Editor Reference Manual	SW09-550	Implementation/Engineering Operations-2	TPS 3032-2
Actors Manual	SW09-555	Implementation/Engineering Operations-2	TPS 3032-2

Introduction

Module Overview

Introduction

This course module describes how to use the new Picture Editor enhancements provided in R500 through R530 to build a custom schematic display containing area, unit, PRIMMOD group, annunciator group, and/or point alarm status information.

Objectives

Given a description of requirements for a custom alarm schematic, use the Picture Editor to apply the following enhancements to meet the display requirements:

- **$AL_ENTY DDB variable** to show data for an operator-selected point,
- **alarm status collector** to view a composite alarm status (the highest level alarm) for an area, unit, PRIMMOD group, annunciator group, or individual point,
- **alarm count collector** to view the total number of alarms for a given alarm status category, such as unacknowledged emergency alarms, and
- Universal Station **Processor Status Data Point (PSDP) parameters** to view alarm status information for a unit.

Sample test items

This course module's Criterion Test asks you to demonstrate successful completion of the lab exercise.

Build Custom Alarm Schematic

Composite Alarm Status Collector

Description

New schematic collectors are provided that return the composite alarm status (the highest level alarm) of any of the following:

- the local Area ($AREASTS)
- a unit ($UNITSTS)
- a PRIMMOD group ($PRIMSTS and $PDDBSTS)
- an annunciator group ($ANNSTS and $ADDBSTS)
- a point ($PNTSTS)

The collectors allow you to provide alarm status information on customized alarm displays.

Composite alarm status

Each composite alarm status collector returns a state of the $ALRMSTS enumeration set. The states of the set are

UNACKEM - highest alarm condition is unacknowledged emergency
UNACKHI - highest alarm condition is unacknowledged high
UNACKLO - highest alarm condition is unacknowledged low
ACKEM - highest alarm condition is acknowledged emergency
ACKHI - highest alarm condition is acknowledged high
ACKLO - highest alarm condition is acknowledged low
NOALARM - no alarms exist

Collectors

The formats of the alarm status collectors are shown below.

$AREASTS

$UNITSTS(“Unit ID”)

$PRIMSTS(“PRIMMOD Name”)

$PDDBSTS(Name of variable or DDB String containing a PRIMMOD Name)

$ANNSTS(“Annunciator Title”)

$ADDBSTS(Name of DDB Variable or String containing Annunciator Title)

$PNTSTS(Entity ID)

Examples

The collectors can be added as values or used in variants and conditions.

Value `$PRIMSTS(“DIST841”)`

Variant or Condition `IF $PRIMSTS(“DIST841”)=UNACKEM THEN...`

Alarm Count Collector

Description

New Schematic Collectors are provided to return an integer showing the alarm count of a specified alarm status category for any of the following:

- the local Area ($AREACNT)
- a unit ($UNITCNT)
- a PRIMMOD group ($PRIMCNT and $PDDBCNT)
- an annunciator group ($ANNCNT and $ADDBCNT)
- a point ($PNTCNT)

The collectors allow you to provide alarm count information on customized alarm displays.

Collector format

The formats of the alarm count collectors are shown below (where nnnnnnnn is the alarm status category).

$AREACNT($ALRMSTS:nnnnnnnn)

$UNITCNT("Unit ID",$ALRMSTS:nnnnnnnn)

$PRIMCNT("PRIMMOD Name",$ALRMSTS:nnnnnnnn)

$PDDBCNT(Name of DDB String or Variable containing PRIMMOD name, name of DDB Enumeration variable containing $ALRMSTS:nnnnnnnn)

$ANNCNT("Annunciator Title",$ALRMSTS:nnnnnnnn)

$ADDBCNT(Name of DDB String variable containing Annunciator Title, Name of DDB Enumeration variable containing $ALRMSTS:nnnnnnnn)

$PNTCNT(Entity ID,$ALRMSTS:nnnnnnnn)

Alarm Status Categories

You must use the $ALRMSTS enumeration to specify an alarm status category in the collector, as shown in the example below.

```
$ALRMSTS:UNACKEM
```

The alarm status categories are

UNACKEM	ACKEM
UNACKHI	ACKHI
UNACKLO	ACKLO

Examples

The alarm count collectors can be added as values or used in variants and conditions.

Value `$AREACNT($ALRMSTS:UNACKEM)`

Variant or Condition `IF $AREACNT($ALRMSTS:UNACKEM)<>0 THEN...`

New US PSDP Parameters

Description

In R500, new Universal Station PSDP parameters are available for read-only access to the following information for a specified unit in the local station's Area.

- Unit ID
- Unit Composite Alarm Status
- Unit Alarm Count
- Unit Assignment State
- Unit System Alarm State
- Unit Console Alarm State
- Unit Lost Event Recovery State

Using these parameters, the information that is shown on the unit annunciator boxes of the Alarm Summary and Alarm Annunciator displays can be shown on custom schematics. In addition, a count of alarms at each priority level on each unit can be shown on custom schematics.

As described earlier, there are Picture Editor collectors also available to return the unit alarm count and the unit composite alarm status. The collectors use a user-configured Unit ID. The PSDP parameters require the area-relative unit number and the local node number; however, if you want to provide a complete alarm overview schematic, the PSDP parameters provide access to additional information (assignment state, system alarm state, console alarm state, and lost event recovery state).

Parameters

Each of the new PSDP parameters is an array of 36 values, as indexed by the Area relative unit number. Table 1 defines each parameter (where ii = the Area relative unit index number, and nn = the local US node number).

When adding these values to a custom schematic, the system display database parameter $MY_PNA can be used to access the local station's node number.

Continued on next page

New US PSDP Parameters, Continued

Table 1 New US PSDP Parameters

Parameter	Definition
$PRSTSnn.UNIT_ID(ii)	Unit ID—This parameter of type string returns the 2-character user-configured Unit ID, as defined in the NCF Unit Names. If the Unit is not configured, “garbage” is returned. This will be fixed in a maintenance release to return blanks.
$PRSTSnn.UNACKEM(ii) $PRSTSnn.UNACKHI(ii) $PRSTSnn.UNACKLO(ii) $PRSTSnn.ACKEM(ii) $PRSTSnn.ACKHI(ii) $PRSTSnn.ACKLO(ii)	Unit Alarm Counter—These parameters of type integer return the number of alarms (0-600) for a specified alarm status category in the specified unit. If the Unit is not configured in the area, the counters are zero. The unit alarm count is also available using the following Picture Editor collector: $UNITCNT(Unit ID, $ALRMSTS:nnnnnnnn)
$PRSTSnn.UNITSTAT(ii)	Unit Composite Alarm Status—This parameter returns a state of the enumeration $ALRMSTS indicating the current composite alarm status of the specified unit. The states are listed below in order of precedence: UNACKEM UNACKHI UNACKLO ACKEM ACKHI ACKLO If all the counters are zero, the status is NOALARM. If the Unit is not configured in the area, the status in the area is NOTCONFG The composite Unit alarm status is also available using the following Picture Editor collector: $UNITSTS (“Unit ID”)
$PRSTSnn.UNITSYS(ii)	Unit System Alarm Status—This parameter returns a state of the enumeration ALENBST, indicating if the process alarms for the unit will be annunciated system-wide: ENABLE (alarms are annunciated, historized, and journaled.) DISABLE (alarms are not annunciated. The process alarm event history collection and RTJ are not affected.) INHIBIT (alarms are not annunciated, and the process alarm event history collection and RTJ are stopped.) If the Unit is not configured in the area, the status is DISABLE.
ii= Area relative unit number (index number of unit) nn = US node number	

Continued on next page

Table 1 New US PSDP Parameters, *continued*

Parameter	Definition
$PRSTSnn.UNITCONS(ii)	Unit Console Alarm Status - This parameter returns a state of the enumeration ALENBST, indicating if the process alarms for the unit will be annunciated on the stations in the local console (Console Alarm Status): ENABLE (alarms are annunciated) DISABLE (alarms are not annunciated) If the Unit is not configured in the area, the status is DISABLE.
$PRSTSnn.UNITASSG(ii)	Unit Assignment Status—This parameter returns a state of the enumeration UNITASGN, indicating if the unit is under the control of the local area: NO (unit is not assigned) YES (unit is assigned) If the Unit is not configured in the area, the status is @@@.
$PRSTSnn.UNITRECV(ii)	Unit Lost Event Recovery Status—This parameter returns a state of the enumeration $UNITREC, indicating the current unit alarm recovery state: NOLOST (no lost events) RECVREQD (event recovery has been requested) LOSTEVNT (lost events exist, but no recovery has been requested) RECINPRG (event recovery is in progress) If the Unit is not configured in the area, the status is NOLOST.
ii = Area relative unit number (index number of unit) nn = US node number	

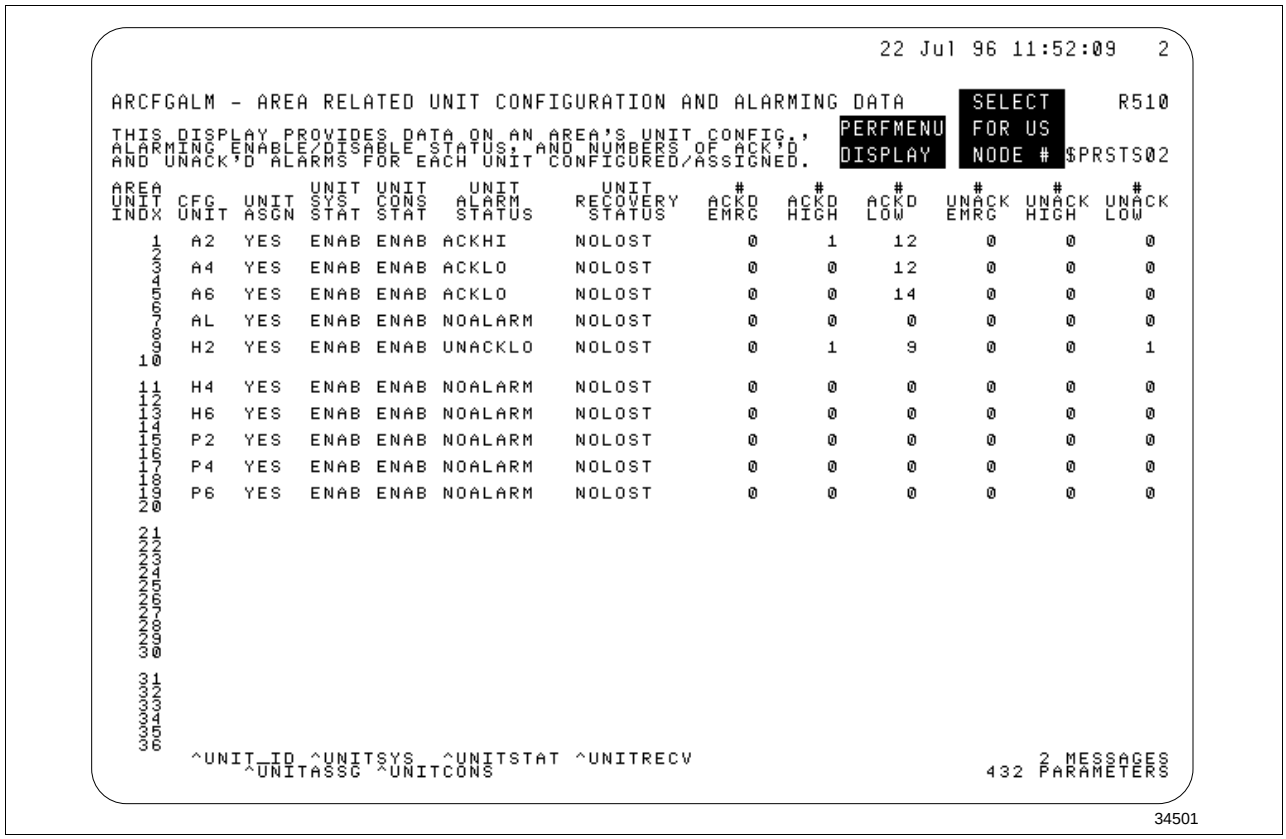
Continued on next page

R500 Universal Station PSDP Parameters, Continued

Area configuration and alarm display

An example of a graphic using the Universal Station PSDP parameters is in the R510 Area Configuration and Alarm display (ARCFGALM) which is accessible from the PERFMENU. This display shows the value of the alarm overview PSDP parameters for each unit. The Area relative index number for each unit is shown here also.

Figure 1 Area Configuration and Alarm Display



New Actor—Convert Unit ID to Unit Number

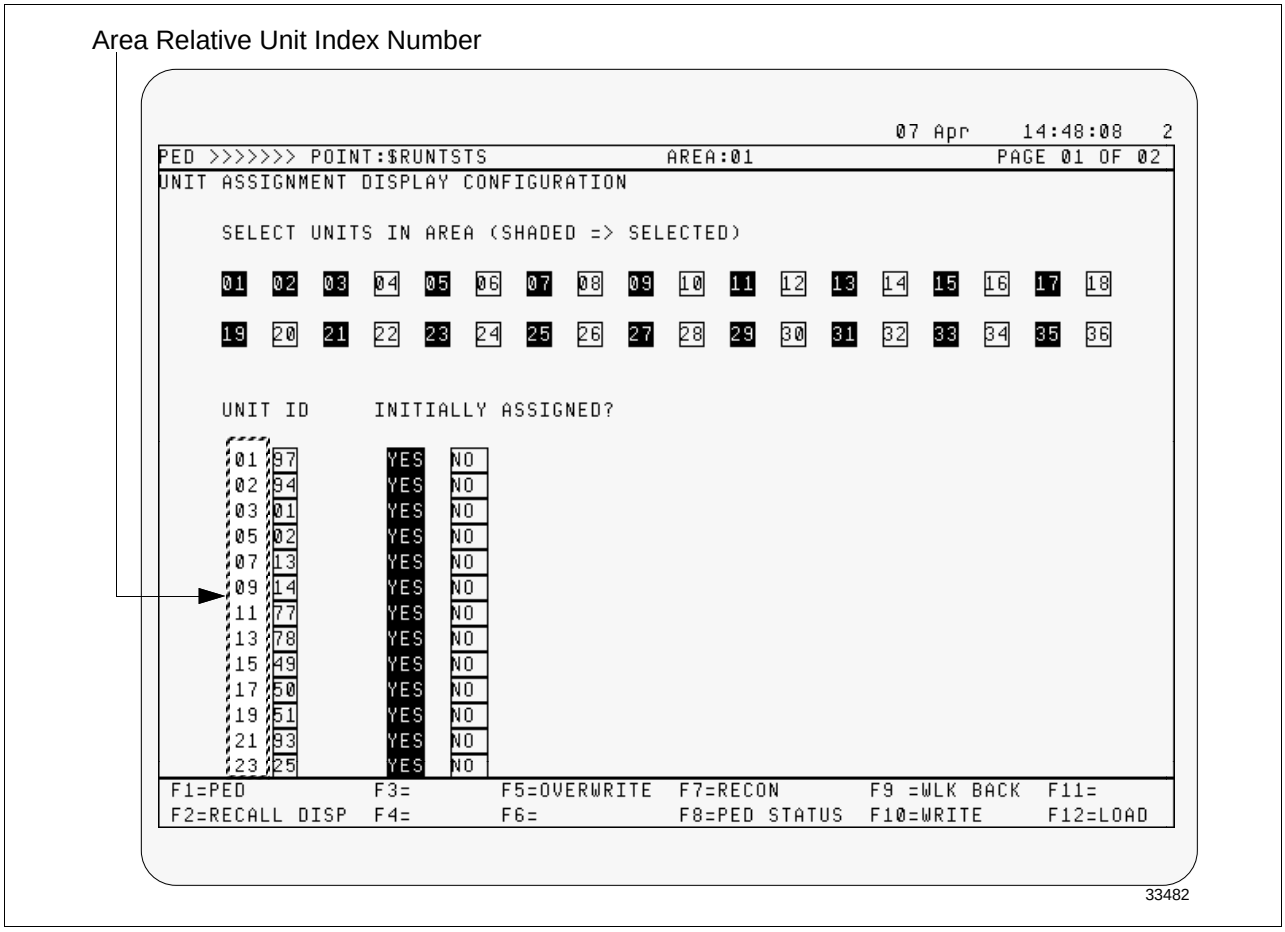
Description

The R500 Picture Editor has a new actor used to convert a user-defined Unit ID to the Area-relative unit index number (1-36).

The actor is intended to be used in conjunction with the new Universal Station PSDP parameters that require the Area-relative unit index number.

Figure 1 shows the Area-relative unit index numbers as they appear in the Area Unit Assignment configuration display.

Figure 2 Area Relative Unit Numbers



Continued on next page

New Actor—Convert Unit ID to Unit Index Number, Continued

Format

The format of the new actor is

```
CNV_UID(“string”)
```

Example

In the following example, the actor is used to convert the Unit ID “FE” to the local Area-relative unit index number. The converted value is then stored to a local DDB integer variable.

```
S_INT(INT01,CNV_UID(“FE”))
```

The converted unit index number can then be used as a parameter index to fetch one of the new PSDP parameters:

```
$PRSTSnn.ACKEM(INT01)
```

`$PRSTSnn.ACKEM(ii)` returns the number of acknowledged emergency alarm in the unit.

where:

`nn`=local US node number
`ii` = unit index number

Application

You can build a target to allow the operator to enter the user’s Unit ID as shown in the following example. The CNV_UID actor converts the operator-entered Unit ID to the index number required by the PSDP parameter:

```
S_INT(INT01,CNV_UID(R_STR(15,1,2,”ENTER UNIT
ID”,T,1)))
```

New Actors—PRIMMOD Status and Count

Description

You can display the composite alarm status of points in a primmod group and the current count of alarms in the primmod group, by using actors PRIMSTS and PRIMCNT. These new actors are the equivalent of primmod collectors $PRIMSTS and $PRIMCNT. The actors can be assigned to a keyboard button or to individual targets on a schematic. PRIMSTS returns the composite alarm status of the points in a primmod group. PRIMCNT returns the current count of how many alarms for a specified alarm priority are in the primmod group.

When assigned to individual targets on a schematic, the PRIMSTS actor returns the composite alarm status of the primmod group. The PRIMCNT actor returns the count of the number of points in the primmod group that are currently in an alarm condition.

PRIMSTS Format

The format of the PRIMSTS actor is as follows:

PRIMSTS (PRIMMOD)

PRIMSTS Example

In the following example, the string representing an alarm status on primmod point AM102 is returned and stored in STRING01.

S_STR(STRING01,"**PRIMSTS** (AM102)")

The string value returned is one of the following values

ACKEM	UNACKEM
ACKHI	UNACKHI
ACKLO	UNACKLO
NOALARM	

Continued on next page

New Actors—PRIMMOD Status and Count, Continued

PRIMCNT Format

The format of the PRIMCNT actor is as follows:

PRIMCNT ("PRIMMOD", ALARM_ENM)

PRIMCNT Examples

In this example, an integer representing the number of unacknowledged emergency alarms on primmod point AM102 is returned and stored in INT01.

S_INT(INT01,**PRIMCNT** ("AM102",UNACKEM))

Another example of the use of the PRIMCNT actor is as follows:

S_INT(INT01,**PRIMCNT** ("AM102",$ALRMSTS:ALARM_ENM))

This example is similar to the previous example except that $ALRMSTS automatically specifies the highest unacknowledged alarm type on primmod point AM102.

New Actors—Configurable Button PRIMMOD Assignment

Description

With R520, you can have a schematic change and retrieve the primmod assignment of any of the 40 configurable keyboard buttons that have LEDs.

The new picture-editor actors are G_PRIM and S_PRIM.

S_PRM Format

S_PRIM allows a primmod to be assigned to the lamp-specific data of a configurable LED button on the keyboard. The format for this actor is

S_PRIM (<button>, <primmod>)

where:**<button>** is a symbolic button name that is a string of 1-8 characters, and resides in the button-name file.

<primmod> is the name of the primmod.

S_PRM Examples

S_PRIM("AN_CNF7","PRIM_AM_GRP101") assigns primmod PRIM_AM_GRP101 to symbolic button AN_CNF7.

The same actor can be used to clear the button assignment by using 1 to 16 null characters instead of the primmod names as follows:

S_PRIM("AN_CNF7"," ")

G_PRM Format

G_PRIM allows the name of the primmod assigned to the lamp-specific data of a configurable LED button on the keyboard to be retrieved. The format for this actor is

G_PRIM (<button>)

where: **<button>** is a symbolic button name that is a string of 1-8 characters, and resides in the button-name file.

Continued on next page

New Actors—Configurable Button PRIMMOD Assignment, Continued

G_PRIM Example

For example: G_PRIM("AN_CNF7") retrieves the name of the primmod that is currently assigned to symbolic button AN_CNF7.

Button-Name File

The button-name file is a user-created text file that contains button names for use in schematic actors and LC SEND statements. This allows button reassignment to be performed without having to modify and recompile schematics and CL programs.

A sample button-name file is as follows:

AN_CNF7 BTN_TK1THIS IS A COMMENT
AN_CNF8 BTN_TK2THIS IS A COMMENT
AN_CF12 BTN_TK3 THIS IS A COMMENT

where: AN_CNF7, AN_CNF8, and ANCF_12 are symbolic names of keyboard buttons 7, 8, and 12.

BTN_TK1, BTN_TK2, and BTN_TK3 are user-defined button names that can be up to eight characters in length.

Each button-name entry allows the user to enter a comment line that relates to the button.

The file name of the user-created button-name text file must have the same file name and residency as the configured button file, except that the user-created text file has an extension of ".KN". The button-name text file is read into the Universal Station during initial startup of the Universal Station and during an area change at the Universal Station.

New DDB Variable for Selected Point—$AL_ENTY

Definition

In R500, a new system Display Database (DDB) variable named $AL_ENTY can be used in schematics to return the name of a point that is selected on the Alarm Summary display, Unit Alarm Summary, Area Annunciator display, or one of the Organizational Summary displays. Whenever the operator selects a point from one of these displays, the name of the point is stored into the global DDB variable. $AL_ENTY can then be used in a schematic that is accessed following the point selection. The information in the schematic could change, depending on the point selected.

The DDB variable can be accessed by standard PE commands or actors.

- To retrieve and store the PV value of the selected point

  `S_REAL(REAL01,GS_REAL($AL_ENTY.PV)`

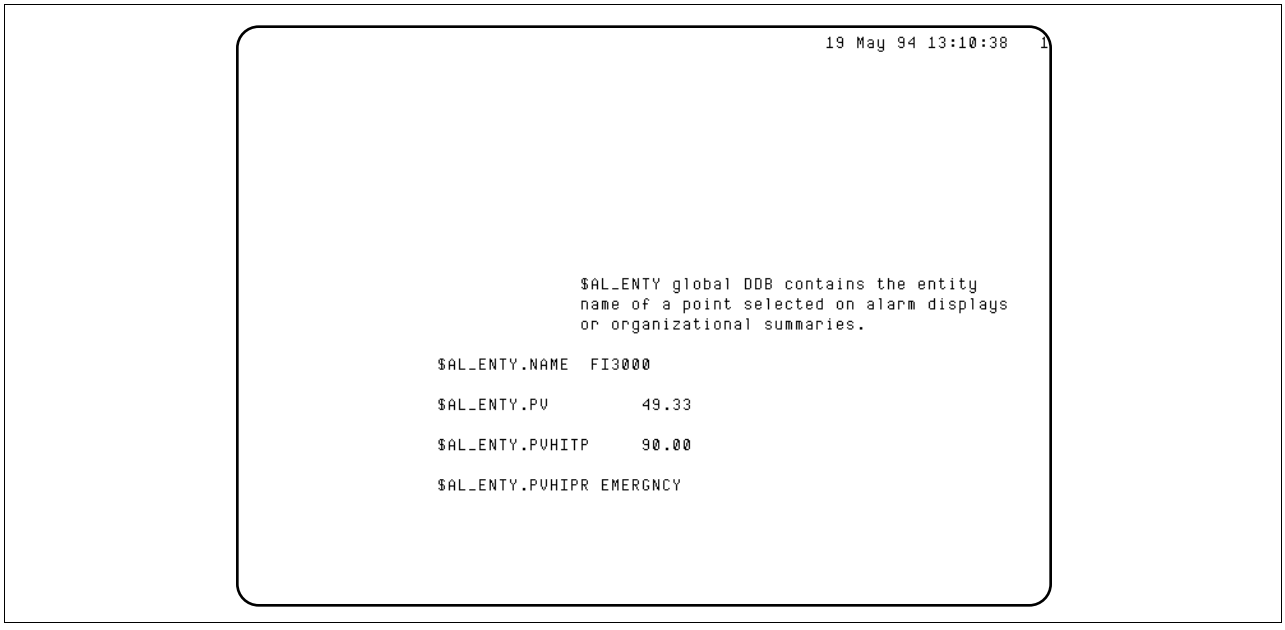
- To add the selected point name to a schematic

  `ADD VAL` of `$AL_ENTY.NAME` with a type of String

Application example 1

Figure 2 is an example of a schematic using $AL_ENTY. The value $AL_ENTY.NAME was added to the schematic. When the operator selects a point and then calls up the schematic, the value returns the name of the selected point. The other values shown in Figure 2 (.PV, .PVHITP, and .PVHIPR) also fill-in with data for the selected point, providing quick access to point information.

Figure 3 Customized Detail Display Using $AL_ENTY



Continued on next page

New DDB Variable for Selected Point—$AL_ENTY, Continued

Digital points

If you select a digital point, then call up a schematic that has $AL_ENTY.PV defined as type real, asterisks (***) are returned. To return the state text on a customized detail display for digital points, specify the PV variable type as sd_enum.

New DDB Variables to Indicate Priority Colors

Description

In R500 the colors to indicate alarm priorities are user selectable. Three DDB variables have been provided to indicate the current color configuration on schematics. The variables allow you to build generic subpictures and schematics to use on different systems and reflect each system's color configuration.

DDB variables

The new DDB variables of type integer are listed below:

- $ALMCOLR—alarm priority color option selected in NCF

  0 = 2 color (red, yellow)

  1 = 3 color (user selects colors)

- $EALMCLR—emergency priority alarm color selected in NCF
- $HALMCLR—high priority alarm color selected in NCF
- $LALMCLR—low priority alarm color selected in NCF

  1 = Red

  2 = Green

  3 = Yellow

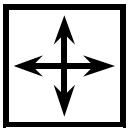
  4 = Blue

  5 = Magenta

  6 = Cyan

  7 = White

Directions



DIRECTIONS—This is the end of the study material for this course module.

At this time, do the lab exercise named “Build Custom Alarm Schematic” (document number L53435L), located immediately following this course module. Discuss questions concerning the study material or lab exercise with a colleague or a course manager.

After completing the lab exercise, if you are satisfied that you have achieved the objective of this course module, continue with the Student Proficiency Evaluation.

Last Page