Table Of Contents

Introductory Concepts

Introduction

The following document outlines and defines all of the setup parameters that are available in the complete family of 87 series displays. It is a technical reference manual, and thus assumes familiarity with the basic operation of the models as outlined in their respective user manuals.

The 161 available setup parameters are used to create and program the complete "personality" of an 87 series display. The parameter settings are stored in the non-volatile setup memory of 87 series displays, and can also be represented by a special text file (called the parameter file). The parameter file serves as a bridge between the setup memory of an 87 series display and the Vorne Display-Pro® 4 (VDP4) software which is used to simplify setup for advanced 87 series applications. The bridge goes two ways - VDP4 can use a parameter file to program the setup memory of an 87 series display, and it can also create a parameter file by reading the setup memory of a display.

Modifying a parameter file can be complex, as there are interrelationships between many parameters that may not be immediately obvious. It is best to first use the "point and click" screens of the VDP4 software to define the basic operation of a unit, and then tweak the parameter file to satisfy any specialized aspects of the application. Keep in mind that the vast majority of applications can be satisfied without having to ever modify a parameter file. However, carefully reviewing this document may still be worthwhile, as it will give you a very thorough understanding of the full capabilities of the 87 series.

This document groups parameters together by type of operation, and where possible by model of unit. There are frequent references to 87 series model numbers in the following pages, so a summary table of the models is presented below. Note that the 87/100 modular display system is essentially an 87/232 serial input display with the hardware ability to drive extra display digits. It is therefore not treated separately in this document.

Model	Description
87/232	Serial Input Display
87/256	Count Display Plus
87/415	Rate Display Plus
87/705	Programmable Timer
87/708	Real Time Clock System
87/712	Parallel Input Display
87/719	Analog Input Display
87/805	Efficiency Display

Conventions Used In This Document

Although all 87 series units have seven segment displays optimized for displaying numbers, they are also capable of intelligibly displaying many characters. Thus, whenever this document refers to digits or numbers for display, keep in mind that in many cases characters and "strings" of characters can also be displayed (reference the Displayable Characters section).

Process Variables

When working with 87 series parameters, it is very important to understand how ***process variables*** (the items that are measured and displayed) are numerically measured, stored and displayed by 87 series displays. We will use an example of an 87/256 counter application to help explain some basic concepts.

We refer to the actual number of count impulses that the display receives as the ***measured process variable*** - the information we receive from the outside world. When the count impulses are accumulated and stored internally, they can go through a prescaler before being stored. For example, with a prescaler of 5, if the ***measured process variable*** is 10 counts, the ***stored process variable*** will actually be 50 (10 counts multiplied by the prescaler of 5). There are additional manipulations that can be done to a ***stored process variable*** before it is displayed (e.g. round, shift, add zeroes, etc.). Thus the ***displayed process variable*** while based on the ***stored process variable*** can be different due to manipulations that can be done to the numbers. If no numeric manipulations are set up in the parameter file then all three numbers will be the same (the ***measured process variable***, the ***stored process variable***, and the ***displayed process variable*** will all be equal to each other).

Many parameters in the 87 series (e.g. presets) must be set up to take into account the manipulations done to numbers, because they are compared to the ***stored process variable,*** not the ***displayed process variable***. Normally this complexity is hidden to the user by the VDP4 software, which automatically adjusts numbers entered by the user to take this into account.

The following stored process variables are used in the 87 series, and have the number formats as shown in the table below.

Average rate is calculated every 2.56 seconds.

Rate is calculated and stored to two decimal points.

Process Variable	Format	Models That Can Access Variable
Current Count	Signed Quad	87/256, 87/415
Current Rate	Unsigned Long	87/256, 87/415
Minimum Rate	Unsigned Long	87/256, 87/415
Maximum Rate	Unsigned Long	87/256, 87/415
Down Time	Signed Long	87/256, 87/415
Timer	Signed Quad	87/705
Timer Days	Signed Quad	87/705
Lap Time	Signed Quad	87/705
Clock Time	Signed Quad	87/708
Analog Value	Signed Long	87/719
Minimum Analog Value	Signed Long	87/719
Maximum Analog Value	Signed Long	87/719
Parallel Value	Signed Long	87/712
Production Efficiency	Unsigned Long	87/805
Production Count	Signed Long	87/805
Production Pace Count	Unsigned Long	87/805
Production Net Count	Signed Long	87/805
Production Average Rate	Unsigned Long	87/805
Serial Display Value	Unsigned String	87/232

Storing Process Variables On Power Down

Parameter Name	System_SaveAtPowerDown
Applies To	87/256, 87/415, 87/705, 87/805 units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter controls whether the process variable values (and thus the display screen value) will be saved when power is removed from the display, and restored when power is restored. **Yes** means that the process variable value will be saved when power is removed. **No** means that the process variable value will NOT be saved when power is removed. Timer/Down Time, Lap Time, Count, Pace Count, Input Count are saved. Min/Max values are not saved.

System And User Parameters

User Setup parameters are 87 series parameters that can be modified while the 87 series unit is operating in run mode. System parameters can only be modified when the 87 series unit is in program mode. The most frequently modified 87 series parameters are designated as User Setup parameters to allow easy modification in the field. The low and high presets are two examples of user setup parameters.

Number Formats Used For Parameters

All numeric values are stored in the 87 series in one of seven different formats, all of which are whole numbers. The seven formats are described below. Note that commas are optional when representing numbers in the parameter file (they are ignored, but can be used to make long numbers easier to read).

Format Name	Format Description
Unsigned Byte	This is an unsigned 8 bit binary number, ranging from 0 to 255.
Signed Byte	This is a signed 8 bit binary number, ranging from -128 to 127.
Unsigned Word	This is an unsigned 16 bit binary number, ranging from 0 to 65,535.
Signed Word	This is a signed 16 bit number, ranging from -32,768 to 32,767.
Unsigned Long	This is an unsigned 32 bit binary number, ranging from 0 to 4,294,967,295.
Signed Long	This is a signed 32 bit number, ranging from -2,147,483,648 to 2,147,483,647.
Signed Quad Word	This is a signed 64 bit number, ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807.

Signed numeric values are stored in a standard two's complement fashion. The reason this format is used is that allows representing negative numbers in a very natural and easy to work with fashion. Because of the way they are stored, overflow of signed numbers is always to the number of opposite sign (e.g. -128 to 127 and vice versa).

As an example, the Signed Byte stores number as follows:

-128	1000 0000	80h
-1	1111 1111	FFh
0	0000 0000	00h
1	0000 0001	01h
127	0111 1111	7Fh

Character Formats Used for Parameters

Many 87 series parameters require arguments of one or more ASCII characters. The parameter file allows these ASCII characters to be represented by a number of different methods.

All 256 ASCII codes can be represented by their decimal or hexadecimal equivalents.

Decimal equivalents are represented as <X> where X represents a decimal number. For example, the ASCII code for carriage return, which is decimal 13, would be represented by **<13>**.

Hexadecimal equivalents are represented as <YYh> where YY represents a two digit hexadecimal number. For example, the ASCII code for carriage return, which is hexadecimal 0D, would be represented by **<0Dh>**.

The first 32 ASCII characters (and character 127 - the delete character) are control codes. These characters can be represented in a parameter file by their decimal or hexadecimal equivalents, or also by a short mnemonic equivalent. The next 95 ASCII characters have a standard literal representation. These characters can be represented in a parameter file by their decimal or hexadecimal equivalents, or also by their literal representations. The tables below show the first 128 ASCII characters, with all three forms of equivalent representation.

The second 128 ASCII characters are represented in parameter files by their decimal or hexadecimal equivalents.

Decimal	<0>	<1>	<2>	<3>	<4>	<5>	<6>	<7>
Hexadecimal	<00h>	<01h>	<02h>	<03h>	<04h>	<05h>	<06h>	<07h>
Mnemonic	<NUL>	<SOH>	<STX>	<ETX>	<EOT>	<ENQ>	<ACK>	<BEL>

Decimal	<8>	<9>	<10>	<11>	<12>	<13>	<14>	<15>
Hexadecimal	<08h>	<09h>	<0Ah>	<0Bh>	<0Ch>	<0Dh>	<0Eh>	<0Fh>
Mnemonic	<BS>	<HT>	<LF>	<VT>	<FF>	<CR>	<SO>	<SI>

Decimal	<16>	<17>	<18>	<19>	<20>	<21>	<22>	<23>
Hexadecimal	<10h>	<11h>	<12h>	<13h>	<14h>	<15h>	<16h>	<17h>
Mnemonic	<DLE>	<DC1>	<DC2>	<DC3>	<DC4>	<NAK>	<SYN>	<ETB>

Decimal	<24>	<25>	<26>	<27>	<28>	<29>	<30>	<31>
Hexadecimal	<18h>	<19h>	<1Ah>	<1Bh>	<1Ch>	<1Dh>	<1Eh>	<1Fh>
Mnemonic	<CAN>	<EM>	<SUB>	<ESC>	<FS>	<GS>	<RS>	<US>

Decimal	<32>	<33>	<34>	<35>	<36>	<37>	<38>	<39>
Hexadecimal	<20h>	<21h>	<22h>	<23h>	<24h>	<25h>	<26h>	<27h>
Literal	space	!	"	#	$	%	&	'

Decimal	<40>	<41>	<42>	<43>	<44>	<45>	<46>	<47>
Hexadecimal	<28h>	<29h>	<2Ah>	<2Bh>	<2Ch>	<2Dh>	<2Eh>	<2Fh>
Literal	(	)	*	+	,	-	.	/

Decimal	<48>	<49>	<50>	<51>	<52>	<53>	<54>	<55>
Hexadecimal	<30h>	<31h>	<32h>	<33h>	<34h>	<35h>	<36h>	<37h>

Literal	0	1	2	3	4	5	6	7

Decimal	<56>	<57>	<58>	<59>	<60>	<61>	<62>	<63>
Hexadecimal	<38h>	<39h>	<3Ah>	<3Bh>	<3Ch>	<3Dh>	<3Eh>	<3Fh>
Literal	8	9	:	;	<	=	>	?

Decimal	<64>	<65>	<66>	<67>	<68>	<69>	<70>	<71>
Hexadecimal	<40h>	<41h>	<42h>	<43h>	<44h>	<45h>	<46h>	<47h>
Literal	@	A	B	C	D	E	F	G

Decimal	<72>	<73>	<74>	<75>	<76>	<77>	<78>	<79>
Hexadecimal	<48h>	<49h>	<4Ah>	<4Bh>	<4Ch>	<4Dh>	<4Eh>	<4Fh>
Literal	H	I	J	K	L	M	N	O

Decimal	<80>	<81>	<82>	<83>	<84>	<85>	<86>	<87>
Hexadecimal	<50h>	<51h>	<52h>	<53h>	<54h>	<55h>	<56h>	<57h>
Literal	P	Q	R	S	T	U	V	W

Decimal	<88>	<89>	<90>	<91>	<92>	<93>	<94>	<95>
Hexadecimal	<58h>	<59h>	<5Ah>	<5Bh>	<5Ch>	<5Dh>	<5Eh>	<5Fh>
Literal	X	Y	Z	[	\	]	^	_

Decimal	<96>	<97>	<98>	<99>	<100>	<101>	<102>	<103>
Hexadecimal	<60h>	<61h>	<62h>	<63h>	<64h>	<65h>	<66h>	<67h>
Literal	'	a	b	c	d	e	f	g

Decimal	<104>	<105>	<106>	<107>	<108>	<109>	<110>	<111>
Hexadecimal	<68h>	<69h>	<6Ah>	<6Bh>	<6Ch>	<6Dh>	<6Eh>	<6Fh>
Literal	h	i	j	k	l	m	n	o

Decimal	<112>	<113>	<114>	<115>	<116>	<117>	<118>	<119>
Hexadecimal	<70h>	<71h>	<72h>	<73h>	<74h>	<75h>	<76h>	<77h>
Literal	p	q	r	s	t	u	v	w

Decimal	<120>	<121>	<122>	<123>	<124>	<125>	<126>	<127>
Hexadecimal	<78h>	<79h>	<7Ah>	<7Bh>	<7Ch>	<7Dh>	<7Eh>	<7Fh>
Literal	x	y	z	{	|	}	~	<DEL>

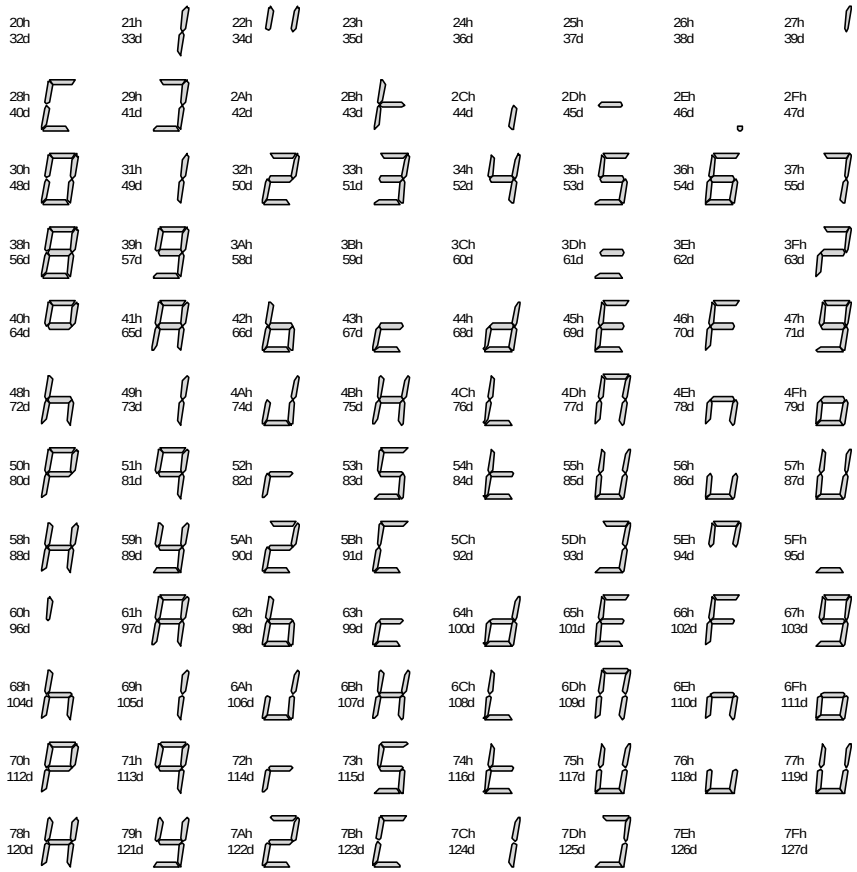
Display Screen Formatter Parameters

Introduction

In the previous section we discussed the differences between measured, stored and displayed process variables. This section outlines the parameters that control the interpretation of the stored process variables when they are displayed or transmitted serially. These parameters do not change the stored process variable value in any way, they only interpret it for display or serial transmission.

Displayable Characters

A subset of the ASCII characters set can be displayed on any 87 series display. The table below lists the available character set in a standard ASCII format. As a primarily numeric, seven segment display, many non-numeric characters can be displayed in only a limited fashion. The hexadecimal and decimal ASCII codes are also shown for each character. The upper and lower case alpha character sets are identical. They are designed to be as readable as possible, within the constraints of a seven segment display. If a character is not listed it will be displayed as blank (as will all characters 20h and below). The upper 128 ASCII codes (decimal 128 to 255) map to the lower 128 ASCII codes shown below.



Serial Data Screen Formatting

Parameter Name	System_RightJustify
Applies To	All units.
Format	Choice of Yes or No.
Default Setting	Yes
Description	This parameter allows a serial string to be placed on the display starting from either the right or left hand side of the display. If the length of the string is less than the size of the display, then the remaining characters will be filled with System_PadChar. **Yes** means that data will be right justified on the display screen. **No** means that data will be left justified on the display screen. When System_COM_MODE is set to V77Mode (providing compatibility with Vorne 77/232 units) the digits will be left justified regardless of the setting of this parameter.

Parameter Name	System_PadChar
Applies To	All units.
Format	Any single character (except the <NUL> character). Each character can also be represented by its hexadecimal, decimal or mnemonic equivalent.
Default Setting	**<20h>**
Description	This parameter controls what character will be used by the 87/232 unit to fill empty positions on the display screen. The default setting of <20h> is the space (sometimes referred to as the blank) character.
Examples	If System_RightJustify is set to Yes and System_PadChar is set to **0**, a six digit 87/232 unit receiving a process variable of 56 for display would show it as 000056.

Parameter Name	System_StripLeadingZeroes
Applies To	All units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter controls leading zero blanking. Leading zero blanking replaces any leading zero characters (ASCII <30h>) with spaces (ASCII <20h>). Spaces display as blank digit positions. **Yes** means that leading zeroes will be blanked. **No** means that leading zeroes will not be blanked.
Examples	If this parameter is set to **Yes** and the number for display is 0024, it will be displayed as 24.

Parameter Name	System_FixedDP
Applies To	All units.
Format	Number ranging from 0 to 8.
Default Setting	0

Description	This parameter allows a decimal point to be permanently turned on in a fixed position on the display screen. **0** means that there will be no fixed decimal point. **1** means that that the most significant (left most) digit will be followed by a fixed decimal point. **2** to **8** means that the referenced digit position (counting from the left most digit) will be followed by a fixed decimal point. When using Serial Virtual Displays (reference the Virtual Displays section) this parameter will affect all virtual display addresses. It is not possible to have a decimal point in front of the most significant (left most) digit, as decimal points are physically placed only on the right side of digits.
Examples	If an 87/232 unit receives a serial string of 2456 for display, and has a System_FixedDP setting of **2**, the display screen will show 24.56.

Process Variables Screen Formatting

These parameters do not apply to 87/232 units, as they have a different set of parameters to control display screen formatting (see the Serial Display Screen Formatting section). The 87 series models and the specific process variables they are normally associtated with are described below. Keep in mind that virtual displays (described in the Virtual Displays section) can access and display process variables from multiple 87 series models (for example showing *Current Count*, *Current Rate*, and *Down Time* all on one display).

Count process variables are referenced in this section by parameter names prefixed with **System_Counter**. These parameters apply to the *Current Count*, *Pace Count, and Production Count* process variables, and the 87/256, 87/415 and 87/805 units.

Rate process variables are referenced in this section by parameter names prefixed with **System_Rate**. These parameters apply to the *Current Rate*, *Minimum Rate*, *Maximum Rate* and *Average Rate* process variables, and the 87/256, 87/415 and 87/805 units.

Time and timer process variables are referenced in this section by parameter names prefixed with **System_Timer**. These parameters apply to the *Down Time*, *Timer*, *Lap Time*, and *Clock Time* process variables, and the 87/256, 87/415, 87/705 and 87/708 units. Time and timer process variables are treated differently than other process variables in this section, because they can have up to three timing stages defined (see the 87/705 Programmable Timer Parameters section), each with separate formatting. A one stage timer (e.g. only timing in hours) uses the timer parameters ending in 0. A two or three stage timer or clock (e.g. timing in hours and minutes) uses the timer parameter ending in 0 for the left most timer stage, and the timer parameter ending in 1 for the right most timer stage. A three stage timer or clock (timing in hours, minutes and seconds) always uses a fixed format for the middle stage (minutes) that is described with the timer format parameter below.

The 87/712 parallel data process variable is referenced in this section by parameter names prefixed with **System_Parallel**. These parameters apply to the *Parallel Value* process variable.

87/719 analog data process variables are referenced in this section by parameter names prefixed with **System_Analog**. These parameters apply to the *Analog Value*, *Minimum Analog Value* and *Maximum Analog Value* process variables.

The 87/805 production monitor process variable is referenced in this section by parameter names prefixed with **System_Prodmon**. These parameters apply to the *Production Monitor* process variable.

Parameter Name	System_CounterFormatCode System_RateFormatCode System_TimerFormatCode0 System_TimerFormatCode1 System_ParallelFormatCode System_AnalogFormatCode System_ProdmonFormatCode
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	List of six items, one from each line below. SpaceFill ZeroFill HidePlus ShowPlus Signed Unsigned Decimal Hexadecimal NoPrecision UsePrecision NoRoundOff UseRoundOff
Default Setting	*87/705, and 87/708 units default to:* ZeroFill + HidePlus + Unsigned + Decimal + NoPrecision + NoRoundOff *87/415 units default to:* SpaceFill + HidePlus + Unsigned + Decimal + NoPrecision + NoRoundOff *Other units default to:* SpaceFill + HidePlus + Signed + Decimal + NoPrecision + NoRoundOff
Description	This set of parameters is one of three that controls the interpretation of stored process variables when they are displayed. **SpaceFill** means any leading zeroes will be replaced by spaces. Spaces display as blank digit positions. **ZeroFill** means that leading zeroes will not be replaced, and thus will be displayed as zeroes. **HidePlus** means that positive numbers will be displayed without a plus sign. This choice is generally recommended for 87 series units, as the plus sign is rather unintelligible, on a seven segment display (refer to the Displayable Characters section to view the plus sign - ASCII <2Bh>). **ShowPlus** means that positive numbers will be preceded by a plus sign. **Signed** means that the stored process variable will be interpreted as a signed integer. It is important to note that this does not affect the way the process variable is stored, but only the interpretation when it is displayed. This choice should generally match the native format of the stored process variable, and in the case of the 87/712 unit should be set appropriately for the type of parallel data being loaded. **Unsigned** means that the stored process variable will be interpreted as an unsigned integer. **Decimal** means that the stored process variable will be interpreted as a base 10 (decimal) number. **Hexadecimal** means that the stored process variable will be interpreted as a base 16 (hexadecimal) number. **NoPrecision** means that the System_....Precision parameter will be ignored. This

	parameter is another of the three parameters that control how the stored process variable will be interpreted on the display screen. **UsePrecision** means that the System_....Precision parameter will be used. **NoRoundoff** means that the displayed process variable will not be rounded when right hand digits are truncated. Right hand digits can be truncated by the System_...Precision parameter. **UseRoundoff** means that 4/5 rounding will be applied to the displayed process variable when right hand digits are truncated. 87/705 units with a three stage timer (hours minutes and seconds) and 87/708 six digit clocks automatically format the minutes as: ZeroFill + HidePlus + Signed + Decimal + NoPrecision + NoRoundOff
Examples	See the examples at the end of this section.

Parameter Name	System_CounterPrecision System_TimerPrecision0 System_TimerPrecision1 System_RatePrecision System_ProdmonPrecision System_ParallelPrecision System_AnalogPrecision
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	List of two items, one from each line below. ZeroOut_0 ZeroOut_1 ZeroOut_2 …. ZeroOut_15 DropOff_0 DropOff_1 DropOff_2 ... DropOff_15
Default Setting	ZeroOut_0 + DropOff_0 For 87/705, 87/708, 87/712, 87/719, and 87/805 Rate and Counter have a drop off of 4 by default.

Description	This set of parameters is one of three that controls the interpretation of stored process variables when they are displayed. The Zero Out function is used in applications where it is desirable to create fixed zero positions in one or more of the less significant (right most) display screen digits. The most common use is for rate applications, where it may be desirable to give an appearance of extra rate stability, or make the rate easier to read by zeroing the least significant digits (see Examples below). This function takes the form of ZeroOut_X, where X is the number of right hand display screen digits to be fixed at zero (X should always be less than the number of display screen digits). **ZeroOut_0** means that this function will have no effect (no digits will be zeroed). **ZeroOut_1** means that only the least significant display screen digit will be zeroed. **ZeroOut_2** means that the two least significant display screen digits will be zeroed. The Drop Off function is used in applications where it is desirable to drop some of the least significant (right most) digits of the stored process variable when it is interpreted for display. The most common application for this feature is when setting up fractional scale factors (refer to the examples at the end of this section). This function takes the form of DropOff_X, where X is the number of right most process variable digits to drop off. **DropOff_0** means that this function will have no effect (no digits will be dropped off before display). **DropOff_1** means that the right most digit will be dropped off before display. **DropOff_2** means that two right most digits will be dropped off before display. The drop off function is performed first, followed by the zero out function.
Examples	**ZeroOut_3** in a rate application where the stored process variable fluctuates between 45678, 45987 and 46111 will be displayed as 45000, 45000 and 46000, giving an easier to read display. For additional examples, refer to the end of this section.

Parameter Name	System_CounterFieldSize System_TimerFieldSize0 System_TimerFieldSize1 System_RateFieldSize System_ProdmonFieldSize System_ParallelFieldSize System_AnalogFieldSize
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	List of two items, one from each line below. Field_0 Field_1 Field_2 ... Field_15 Decimal_0 Decimal_1 Decimal_2 ... Decimal_15
Default Setting	Field_0 + Decimal_0
Description	This set of parameters is one of three that controls the interpretation of stored process variables when they are displayed. The Field function can be used to set the size of the displayed process variable (including any displayed signs) to a fixed number of digits. When using the Decimal function (described below) to add a decimal point to the displayed process variable, the Field function must be increased by 1 to accommodate the decimal point position. The Field function takes the form of Field_X, where X is a number ranging from 0 to 15, representing the fixed size of the displayed process variable (with one added when a decimal point is used). **Field_0** is a special case as described below. If Field_X is set to lesser number of digits than the displayed process variable, then X right most digits will be displayed, and the remaining left most digits will be lost. **Field_0** means that this function will have no effect (the displayed process variable will not have a fixed size). If the displayed process variable is larger than the display screen capacity the left most digits will not be displayed. **Field_1** means that the displayed process variable will be fixed at 1 digit. **Field_2** means that the displayed process variable will be fixed at 2 digits (or 1 digit and 1 decimal point). **Field_3** means that the displayed process variable will be fixed at 3 digits (or 2 digit and 1 decimal point). The Decimal function can be used to insert a fixed decimal point in the displayed process variable. This function takes the form of Decimal_X, where X is the number of displayed digits to the right of the fixed decimal point. **Decimal_0** is a special case as described below. **Decimal_0** means that this function will have no effect (there will be no displayed decimal point). **Decimal_1** means that the decimal point will precede the least significant (right most) digit. **Decimal_2** means that the decimal point will precede the 2 right most digits. **Decimal_3** means that the decimal point will precede the 3 right most digits. Note that the Decimal function must be set to a smaller number than the Field function.
Examples	**Field_4 + Decimal_1** would result in a display of format XX.X. **Field_6 + Decimal_0** would result in a display of format XXXXXX. For additional examples, refer to the end of this section.

Parameter Name	System_CounterSerialFieldSize System_TimerSerialFieldSize0 System_TimerSerialFieldSize1 System_RateSerialFieldSize System_ProdmonSerialFieldSize System_ParallelSerialFieldSize System_AnalogSerialFieldSize
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	List of two items, one from each line below. Field_0 Field_1 Field_2 ... Field_15 Decimal_0 Decimal_1 Decimal_2 ... Decimal_15
Default Setting	Field_0 + Decimal_0
Description	This set of parameters controls the interpretation of stored process variables when they are transmitted by an 87 series unit through it's serial port. These parameters are only modified under special circumstances. The Field and Decimal functions operate exactly the same as the System_...FieldSize parameter functions described in the previous table, with one important exception. In this case, the functions are not applied to displayed process variables, but instead to serially transmitted process variables. This allows serially transmitted process variables to be larger than their displayed counterparts, overcoming any limitations caused by the number of available display digits. The System_...FormatCode and System_...Precision parameters still apply to process variables which are serially transmitted.
Examples	See the examples at the end of this section.

Parameter Name	System_DigitTranslate
Applies To	All units except 87/232.
Format	String of 16 characters. Each character may be represented by its hexadecimal, decimal, literal or mnemonic equivalent.
Default Setting	0123456789ABCDEF
Description	This parameter allows characters to be substituted for the most commonly displayed 87 series characters (the numbers 0 through 9 as well as the hexadecimal characters A through F). An 87/712 unit overrides the System_DigitTranslate value with 0123456789-EHLP<SP> if BCD mode is selected.
Examples	BCD units may use this function to define the display representation of the typically unused characters A though F. Some PLCs will set the I/O port to 0xFF when there is no output, in this case the F character could be redefined as a space to make the display blank.

Given the 1 foot to 1 inch scale factor of .08333 and the need to display a count in feet, the following set of parameters need to be modified.

UserSetup_CounterScale1 = 8333 We will imply a decimal point in the fifth digit.

Therefore to display the count in a 3 digit field with no decimal portion:

System_CounterFormatCode =
SpaceFill + Unused digits are blank.
HidePlus + Don't show a plus sign for positive numbers.
Signed + Count is defined as a Signed Value.

Decimal + Show the number in base 10.
UsePrecision + Apply the precision format code.
UseRoundOff Apply 4/5 rounding to the value after the precision is applied.

System_CounterPrecision =
DropOff_5 + Discard the entire decimal portion of 5 digits.
ZeroOut_0 No digits are forced to zero.

System_CounterFieldSize =
Field_3 + Fixed display size of three digits (in feet).
Decimal_0 No decimal portion.

Now, if we wanted to show the rate in XX.X feet per minute on a virtual display:

SystemRateFormatCode =
SpaceFill + Unused digits are blank.
HidePlus + Don't show a plus sign for positive numbers.
Unsigned + Rate is defined as a Unsigned Value.
Decimal + Show the number in base 10.
UsePrecision + Apply the precision format code.
UseRoundOff Apply 4/5 rounding to the value after the precision is applied.

System_RatePrecision =
DropOff_4 + Discard 4 decimal portion digits.
ZeroOut_0 No digits are forced to zero.

System_RateFieldSize =
Field_4 + Fixed display size of three digits plus a decimal point (in feet/min.).
Decimal_1 Put decimal point between first and second digits (XX.X).

Finally we are going to show down time on another virtual display, in the format MM:SS.S:

Note: Set SystemTimerType = TimerMinutesSeconds

Use the Timer0 codes for the seconds portion.

System_Timer0FormatCode =
ZeroFill + Fill unused seconds with zeroes because of colon.
HidePlus + Don't show a plus sign for positive numbers.
Unsigned + Timer is defined as a Unsigned Value.
Decimal + Show the number in base 10.
UsePrecision + Apply the precision format code.
NoRoundOff Never a good idea to round up time.

System_Timer0Precision =

DropOff_1 +	Time is stored as XX.XX seconds, we only want XX.X.
ZeroOut_0	No digits are forced to zero.

System_Timer0FieldSize =

Field_4 +	Fixed display size of three digits plus a decimal point for tenths of seconds.
Decimal_1	Put decimal point between first and second digits (XX.X).

System_Timer1FormatCode =

SpaceFill +	Allow unused minutes to be blanked.
HidePlus +	Don't show a plus sign for positive numbers.
Unsigned +	Timer is defined as a Unsigned Value.
Decimal +	Show the number in base 10.
NoPrecision +	Only apply precision to stage one of a timer.
NoRoundOff	Never a good idea to round up time.

System_Timer1Precision =

DropOff_0 +	No digits are dropped.
ZeroOut_0	No digits are forced to zero.

System_Timer1FieldSize =

Field_2 +	Minutes part is two digits long.
Decimal_0	No decimal point in minutes.

Tri-color Displays

The optional tri-color displays use a concept called color masks to control and change the color of the display. Color masks are a string of characters, each of which represents the color of one digit of the display.

The four color mask parameters listed below can be used to automatically trigger color changes in a display when it reaches a preprogrammed preset (reference the Preset Actions section), or as it sequentially displays more than one item on the same display (reference the Virtual Displays section). Four color masks are provided to allow a color change for each preset action, or for each of four sequential virtual displays.

Color masks can also be transmitted directly to a display using Vorne SPP (Simple Packet Protocol) serial commands. This method of controlling display color does not require the parameters outlined in the table below, since the color mask is embedded in the command structure.

Parameter Name	System_ColorMask0 System_ColorMask1 System_ColorMask2 System_ColorMask3
Applies To	All units.
Format	String of 2 to 8 characters. Valid characters are B, R, G, or Y. B = Blank R = Red G = Green Y = Yellow
Default Setting	RRRRRRRRRRRRRR
Description	Each of the color masks is a string of characters, with each character representing the color of one digit of the display. The left most character of the color mask represents the left most (most significant) digit of the display. If the color mask contains more characters than the display, extra (right most) characters will be ignored. The default setting is for an all red display.
Examples	**RRBBYY** would set a 6 digit display to 2 red digits, followed by 2 blank digits, followed by 2 yellow digits.

Virtual Displays

Virtual displays are an important concept, as they allow one 87 series display unit to write multiple process variables (e.g. rate, count and down time) to "virtual displays". There are three types of virtual displays - sequential, concurrent and serial.

Sequential virtual displays can show up to four process variables on one display by stepping through a display list in a round robin fashion. A display list is a specially formatted list (created with the System_DisplayList parameter) which specifies which process variables will be displayed. With the tri-color display option, sequential virtual displays can each be programmed to display in one of three available colors by using the color mask parameters (reference the Tri-color Displays section). System_ColorMask0 would be used to control the display color of the first process variable in the System_DisplayList, System_ColorMask1 for the second, and so on.

Concurrent virtual displays show up to four process variables on one display all at the same time by utilizing a large display field (8, 12, 16 or 20 characters) that is split into multiple virtual displays. The display list determines which process variables will be displayed, and the

System_VirtualDisplaySize (described below) determines the number of digits in each virtual display. With the tri-color display option, each virtual display can each be programmed to display in one of three available colors by using the color mask parameters (reference the Tri-color Displays section). Concurrent virtual displays can also be factory ordered with virtual display fields of fixed colors.

Serial virtual displays are commonly used in andon applications, as they allow up to 32 eight digit displays to be controlled from one serial port connection. Each virtual display is written to with a specifically addressed communication packet.

Parameter Name	System_VirtualDisplaySize
Applies To	All units.
Format	String of 1 to 32 numbers (no spaces between the numbers). Valid numbers are 1 to 9, A, B, C, D, E, and F. Where F = 15 digits.
Default Setting	8
Description	This parameter sets up the number of virtual displays (each number in the string represents one display), as well as how many digits there are in each virtual display (represented by the actual number in the string). Thus the default setting of 8 represents one display of 8 digits. All units can take in serial data in addition to their base type display, and show it as a virtual display.
Examples	**844** would represent three displays, an 8 digit, a 4 digit, and a 4 digit.

The actual display list variables are:

NotUsed			
Timer	LapTimer		
Counter			
Rate	RateMin	RateMax	
ProdMonRatio	ProdMonPaceCount	ProdMonRate	ProdMonDifference
Analog	AnalogMin	AnalogMax	
Parallel			
Clock	Days		

Parameter Name	System_DisplayList
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units
Format	List of one to four items from the fifteen choices described above.
Default Setting	NotUsed

Description	This list of parameters determines which items will be displayed when sequential and concurrent virtual displays are used. They are generally based on the types of values that would be viewed on the display, so for example, the count would include any scale factors, results of combining up and down counts, etc. **NotUsed** is the default setting, and enables the default display for all units. 87/256 and 87/415 units offer the following 5 choices: **Counter** will display the current count. **Rate** will display the current rate. **RateMin** will display the lowest rate value reached since the last reset. **RateMax** will display the highest rate value achieved since the last reset. 87/705 units offer the following 2 choices: **Timer** will display the main timer value. **LapTime** will display the lap time value. 87/719 units offer the following 3 choices: **Analog** will display the current analog display value. **AnalogMin** will display the lowest analog display value reached since the last reset. **AnalogMax** will display the highest analog display value reached since the last reset. 87/805 units offer the following 8 choices: **Count** will display the number of counts received since the last reset. **ProdMonDifference** will display the function Count – PaceCount. **ProdMonRatio** will display the function Count / PaceCount. **ProdMonPaceCount** will display the number of pace counts since the last reset. **ProdMonRate** will display the average rate of production since the last reset. **Rate** will display the current rate. **RateMin** will display the lowest rate value reached since the last reset. **RateMax** will display the highest rate value achieved since the last reset.
Examples	The following is an example that would set up a rate monitor application to be able to display current rate, down time (since last reset), and the maximum rate achieved (since last reset). **Rate + Timer + RateMax**

Parameter Name	System_ProcessVariableLocation
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	Number ranging from 0 to 3.
Default Setting	0
Description	Does not apply to the 87/232 because it has no process variables. This parameter selects which of the process variables in the Display List will be considered the primary process variable. The primary process variable is the one to which preset and range restriction parameters are applied. **0** means that the primary process variable is the first process variable listed in the System_DisplayList. **1** means that the primary process variable is the second process variable listed in the System_DisplayList. **2** means that the primary process variable is the third process variable listed in the System_DisplayList. **3** means that the primary process variable is the fourth process variable listed in the System_DisplayList.

Parameter Name	System_DisplayMethod
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.

Format	Number ranging from 0 to 2.
Default Setting	0
Description	This parameter controls how the System_DisplayList will interpreted. **0** means the System_DisplayList will be ignored (not used). **1** means each item listed in the System_DisplayList will be displayed in a sequential round robin fashion, with the timing controlled by the System_DisplayListTime parameter. **2** means each item listed in the System_DisplayList will be displayed concurrently on its own virtual display.

Parameter Name	System_DisplayListTime
Applies To	87/256, 87/415, 87/705, 87/708, 87/712, 87/719, 87/805 units.
Format	Unsigned Word (number ranging from 0 to 65535).
Default Setting	0
Description	This parameter controls the time interval for each step through the System_DisplayList. Each number represents 10 milliseconds (0.01 seconds), allowing a step time ranging from 10 milliseconds (setting of 1) to 655.35 seconds (setting of 65535). The default setting of 0 will cause the System_DisplayList setting to be ignored (not used).
Examples	**200** would represent a step time of 2 seconds for each item in the display list.

Leading And Trailing Display Characters

Displays can be programmed to have up to 3 fixed characters in the left and/or right most section of the display. A common application example would be to display a degree symbol and an F after a process variable to indicate a Fahrenheit temperature.

Parameter Name	System_LeadIn0 System_LeadIn1 System_LeadIn2 System_LeadIn3 System_Trailer0 System_Trailer1 System_Trailer2 System_Trailer3
Applies To	All units except 87/232
Format	String of 0 to 3 characters. Each character can also be represented by a hex code in the format of <XXh> where XX can be any hex number from 20 to 7F (refer to the earlier character chart).
Default Setting	The default setting is blank (no characters).
Description	The system lead in parameters define characters that will be displayed before the displayed process variable. The system trailer parameters define characters that will be displayed after the displayed process variable. The reason there are 4 lead in and 4 trailer parameters is to allow defining lead in and trailer characters for each of the 4 sequential or concurrent virtual displays (reference the Virtual Displays section). Characters are always displayed from left to right (most significant digit to least significant digit), thus if there are more characters than will fit on the display it is always the right most characters which will be lost.
Examples	System_Trailer0 is set to **<40h>F**, and the process variable is currently 127. The display will show 127°F.

Restricting The Range Of The Display

Most 87 series displays can be restricted to only show a programmed range of values, and to take special action if one of the display range limits is reached.

Parameter Name	System_RangeMin System_RangeMax
Applies To	All units except 87/232 and 87/708.
Format	Signed Quad Word
Default Setting	0
Description	These parameters are used to define the minimum (System_RangeMin) and maximum (System_RangeMax) process variable value to be allowed on the display. If the displayed process variable exceeds the range limits set here, the display will refer to the System_RangeRestriction parameter (described below) to determine what action to take.

Parameter Name	System_RangeRestriction
Applies To	All units except 87/232 and 87/708.
Format	List of two items, one each from the two groups shown below. LowNone LowStop LowFlash LowFlashString HighNone HighStop HighFlash HighFlashString
Default Setting	LowNone + HighNone

Description	This parameter controls what will happen if the System_RangeMin or System_RangeMax values are reached. *Low Group* A setting of **LowNone** means the low end range of the display will not be restricted by the System_RangeMin setting. A setting of **LowStop** instructs the unit to freeze the display when it reaches the System_RangeMin setting, and to not show numbers equal to or below the range limit. A setting of **LowFlash** instructs the unit to flash all values equal to or below the System_RangeMin setting. A setting of **LowFlashString** instructs the unit to display the string defined in System_UnderRange for values equal to or below the System_RangeMin setting. *High Group* A setting of **HighNone** means the high end range of the display will not be restricted by the System_RangeMax setting. A setting of **HighStop** instructs the unit to freeze the display when it reaches the System_RangeMax setting, and to not show numbers equal to or above the range limit. A setting of **HighFlash** instructs the unit to flash all values equal to or above the System_RangeMax setting. A setting of **HighFlashString** instructs the unit to display the string defined in System_OverRange for values equal to or above the System_RangeMax setting.
Examples	To flash the display when the stored process variable reaches lower or upper programmed limit points (System_RangeMin or System_RangeMax) set this parameter to **LowFlash + HighFlash**. To flash the display only if the stored process variable reaches the upper limit point (System_RangeMax) set this parameter to **LowNone + HighFlash**.

Parameter Name	System_UnderRange System_OverRange
Applies To	All units except 87/232 and 87/708.
Format	String of 0 to 8 characters. Each character can also be represented by a hex code in the format of <XXh> where XX can be any hex number from 20 to 7F (refer to the character chart in the earlier Displayable Characters section).
Default Setting	-9999999 99999999
Description	These are user programmable strings which will be displayed if System_RangeRestriction is set to **LowFlashString** and the stored process variable value becomes equal to or below the System_RangeMin setting (the System_UnderRange string will be displayed) or if System_RangeRestriction is set to **HighFlashString** and the stored process variable value becomes equal to or above the System_RangeMax setting (the System_OverRange string will be displayed).
Examples	If you would like the display to show "Error" if the System_RangeMax value is reached, set the System_OverRange parameter to **Error** or **<45h><52h><52h><4Fh><52H>**.

Serial Port Parameters

Introduction

Every 87 series display has a built in serial port that is capable of many functions. On the most basic level, this port is used to program the display with it's operating "personality". On a more advanced level this port can be used to transmit to slave displays, to connect a hand held remote terminal, and to access information from the display for data acquisition applications.

Basic Protocol Selections

The following parameters are used to set up basic communication protocols for the 87 series display.

Parameter Name	System_Baud
Applies To	All units.
Format	One item from the following nine choices: 300 600 1200 2400 4800 9600 19200 38400 76800
Default Setting	9600
Description	This parameter sets the baud rate that will be used for receiving serial communications whenever the display is not in program mode (in program mode the baud rate is automatically set to 9600 baud).
Examples	**19200** will set the display to communicate at 19,200 baud.

Parameter Name	System_TransmitBaud
Applies To	All units.
Format	One item from the following nine choices: 300 600 1200 2400 4800 9600 19200 38400 76800
Default Setting	9600
Description	This parameter sets the baud rate that will be used for transmitting serial communications whenever the display is not in program mode (in program mode the TransmitBaud rate is automatically set to 9600 baud). This allows the unit to receive and transmit serial data at different baud rates.
Examples	**19200** will set the display to communicate at 19,200 baud.

Parameter Name	System_Data
Applies To	All units.
Format	One item from the following two choices: 7 8
Default Setting	8
Description	This parameter sets the number of data bits that will be used for serial communications.
Examples	**7** will set the display to communicate using 7 data bits.

Parameter Name	System_COM_MODE
Applies To	All units.
Format	One item from the following four choices: TerminatorMode PacketMode UniversalMode V77Mode
Default Setting	PacketMode
Description	This parameter selects the type of communication packet that will be expected by the 87 series display whenever it is not in program mode (in program mode the communication mode is automatically set to a proprietary communication protocol optimized for programming the display). **TerminatorMode** This is for applications where you can control the data being transmitted to the display, and wish to use the very simplest form of communication possible. We define this simplest form of communication as sending the data to be displayed, followed by a carriage return or line feed character. **PacketMode** This sets the display to respond to the Simple Packet Protocol. This protocol is recommended for applications where you can control the data being transmitted to the display, but where you would like to access some more advanced features. These features include independently addressing multiple displays, sending an optional checksum to validate the data, receiving an optional acknowledgement from the display to verify the receipt of data, or controlling an optional relay in the display. **V77Mode** This is for applications where you want to connect 87 series displays to the same communication bus as 77/232 displays (an earlier Vorne product line), or where you wish to use existing software drivers written for 77/232 displays. **UniversalMode** This is for applications where you have limited or no control of the data being transmitted to the display, and would like the display to intelligently process incoming data and select a portion of it to be displayed. When in this mode, the 87 series display will process incoming serial data according to the contents of the System_TranslateString (see the Translator Strings section below).

Parameter Name	System_SPP_RESPONSE
Applies To	All units.
Format	Choice of Yes or No.
Default Setting	Yes
Description	This parameter is only used if SYSTEM_COM_MODE is set to PacketMode, and the Simple Packet Protocol is being used. When set to **No** the display will not transmit automatic acknowledgements to Simple Packet Protocol packets it receives. When set to **Yes** the display will automatically transmit acknowledgements to all Simple Packet Protocol packets it receives. These acknowledgements let the host computer verify that data was received by the display. The acknowledgement response packets are as follows: <ACK> This means the packet was received and processed by the display. <NAK> This means a valid packet was received, but the request cannot be carried out. <SOH>... This means that a valid packet was received, and the packet requires a response with data returned. The full format of the <SOH>... packet is: <SOH>Z:<#>data<Terminator>[checksum] # is a code representing the status of the received packet. The code can range from 0 to 4 with the following interpretation: 0 The received packet was fully processed. 1 The received packet was in an unacceptable format. 2 The data in the received packet contained a range error. 3 The received packet is not supported by this display. 4 The data in the received packet cannot be written. data is the actual data being returned. <Terminator> is the System_Terminator selected. [checksum] is the packet checksum (only included if System_SPP_CHECKSUM is set to Yes).

Parameter Name	System_SPP_CHECKSUM
Applies To	All units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter is only used if SYSTEM_COM_MODE is set to PacketMode, and the Simple Packet Protocol is being used. Checksums are an optional feature of the Simple Packet Protocol. Using checksums can improve the reliability of data transfer between the host device and the display, as they allow the display to check if data has become corrupted during transmission from the host device to the display (and vice versa). When checksums are used, they are calculated by the host computer or PLC for every transmission packet, and are appended to the end of the packet (after the terminator). The checksum itself consists of two ASCII characters. If the display receives a packet with an incorrect checksum, it will throw away the packet, and take no action on its contents. A checksum will not catch all possible instances of corrupted data, but it can flag most instances. Once activated, the checksum must be calculated and appended to each packet transmitted to the display. In short, the calculation consists of a 2's complement of the sum of all the bytes from **<SOH>** to the terminator character, inclusive, modulo 256. The result of this calculation is a one byte binary number, which is then converted to two hexadecimal digits. These two hexadecimal digits are appended to the end of the packet, and transmitted as two ASCII characters. When set to **No** the display will not expect to receive and will not automatically send checksums with Simple Packet Protocol packets. When set to **Yes** the display will expect to receive and will automatically send checksums with all Simple Packet Protocol packets.
Examples	Please refer to the 87 Series Advanced Application Guide for a complete checksum example.

Parameter Name	System_Terminator
Applies To	All units.
Format	Any single character (except the <NUL> character). Each character can also be represented by its hexadecimal, decimal or mnemonic equivalent.
Default Setting	<CR>
Description	This parameter is used by the display to mark the end of transmission packets in all four System_COM_MODE packet types.

Addressing Multiple Displays

Using an RS-422 or RS-485 network together with addressing allows a host computer or PLC to communicate with specific individual displays or groups of displays in a network. In order to use addressing, the System_COM_MODE must be set to either PacketMode or V77Mode. In PacketMode, each display in the network may be assigned a unit address *and* a group address. In V77Mode only a unit address is allowed. The format of the unit address is different in PacketMode and V77Mode.

Parameter Name	System_Unit
Applies To	All units.
Format	Unsigned Byte (number ranging from 0 to 255).
Default Setting	0
Description	This parameter is used to set a unit address for serial communication, and is only used if SYSTEM_COM_MODE is set to PacketMode or V77Mode. If a packet is directed to a specific unit address, only units set to that address will respond to the data. More than one display may use the same address. In PacketMode the full range of addresses from **0** to **255** is available, each representing an individual and unique unit address. In V77Mode a restricted range of addresses from **0** to **99** is available. A unit address of **0** means that addressing is not being used. A unit address of **1** to **98** represents an individual unit address, allowing up to 98 displays to be individually addressed in a network. A unit address of **99** means the display will show avery set of data it receives regardless of the transmitted address. Note that the 87 series display automatically converts single digit unit addresses in this parameter to the two digit format used by 77 series displays.

Parameter Name	System_Group
Applies To	All units.
Format	One to eight items from the following nine choices: GroupNone Group1 Group2 Group3 Group4 Group5 Group6 Group7 Group8
Default Setting	GroupNone
Description	This parameter is used to set up group addresses for displays communicating with SPP, thus it is only used if SYSTEM_COM_MODE is set to PacketMode. Up to eight display groups can be set up, and each display can belong to any group from no groups (**GroupNone**) to (**Group8**). When **GroupNone** is selected the display will not respond to packets with any group address with one exception (a transmitted group address of 0 is a broadcast to all units, regardless of what the internal address of each display is set to). Groups are useful when it is desirable to address both individual displays *and* groups of displays, or when it is desirable to address displays in various combinations of groups. A transmitted group address can refer to any combination of these groups. The way this is accomplished is that each group is represented by one bit in an eight bit binary number. Since an eight bit binary number can range in value from 0 to 255, it is this range of numbers which can be transmitted as the group address (in ASCII). To communicate with only group 4 you would transmit 8 as a group address. To communicate with only group 7, you would transmit 64 as a group address. To transmit to both groups 4 and 7, you would add together 8 + 64 and transmit 72 as the group address.
Examples	**Group4** will set the display to respond to packets with a group address value with the 3rd bit set (8 to 15, 24 to 31 … 248 to 255).

Translating And Interpreting Incoming Data

UniversalMode (one of the four communication mode choices that can be selected with the System_COM_MODE parameter) exists for its unique and very powerful feature of being able to match and synchronize to almost any repetitive pattern of data transmitted from a host device. The heart of UniversalMode is the translation string - a set of command codes that instruct the display on how to parse an incoming data stream, and select and process portions of the data for display.

Each new data packet (containing up to a maximum of 255 characters) is placed in the input buffer, where it is processed as directed by the command codes contained in the System_TranslateString. Most of the command codes are used to process characters from the input buffer and build a new display string in the display buffer.

Parameter Name	System_NonUSANumbers
Applies To	All units.
Format	List of two items, one from each of the two groups shown below. InputUSA InputNonUSA OutputUSA Output NonUSA
Default Setting	InputUSA + OutputUSA
Description	This parameter controls the interpretation of decimal points and commas received and transmitted through the 87 series serial port. Some devices from outside of the United States have alternate interpretations for commas and decimals in numbers (for example in the USA the number ten thousand may be represented as 10,000.00, while in Europe the same number may be represented as 10.000,00). **InputUSA** means that there will be no conversion of decimal points and commas received through the 87 series serial port. **InputNonUSA** means that decimal points received through the 87 series serial port will be converted to commas, and vice versa. **OutputUSA** means that there will be no conversion of decimal points and commas transmitted through the 87 series serial port. **OutputNonUSA** means that decimal points transmitted through the 87 series serial port will be converted to commas, and vice versa.

Parameter Name	System_TranslateString
Applies To	All units.
Format	String of 0 to 32 characters. Each character is a command code (or an argument to a command code), and can be represented by its hexadecimal, decimal literal, or mnemonic equivalent.
Default Setting	The default setting is blank (no characters).
Description	This parameter is only used when System_COM_MODE is set to UniversalMode. The complete set of command codes is shown below, represented by their hexadecimal equivalents. If less than 32 characters are used in the System_TranslateString, it must end with the <NUL> character (<00h>). The commands are organized below by their general type of function. The data matched or searched will be discarded in the default DISCARD MODE and sent to the display buffer if the mode is set to KEEP MODE. This mode may be changed as frequently as needed in the System_TranslateString. *Special Command Codes* **<00h>** Used to end a System_TranslateString of less than 32 characters. **<80h>** Select KEEP MODE. While in this mode, any further characters encountered will be moved to the display buffer for display. **<81h>** Select DISCARD MODE. While in this mode, any further characters encountered will be discarded.

<F6h> Move the remaining characters from the input buffer into the display buffer.

<8Dh> End the current translate operation, and display the display buffer.

<8Eh> End the current translate operation, and do not display the display buffer.

<F7h> Match the next characters to the unit address.

Looping Command Codes
These command codes allow repetitive execution of a section of the System_TranslateString. This is very useful when searching the input buffer for a particular set of characters that may not be in a fixed position in the input buffer.

<89h> Set this position as marker 1. Markers can be forward and backward referenced.

<8Bh> Jump to marker 1.

<87h>C1 If the next input buffer character does not match the user definable character C1 then jump to marker 1, without advancing the input buffer pointer. Otherwise execute the next command code.

<8Ah> Set this position as marker 2. Markers can be forward and backward referenced.

<8Ch> Jump to marker 2.

<88h>C1 If the next input buffer character does not match the user definable character C1 then jump to marker 2, without advancing the input buffer pointer. Otherwise execute the next command code.

<F8h> Rewind the input buffer pointer back to the beginning. This allows reparsing of the input buffer.

Conversion Command Codes
The following command codes are used to convert characters or case. When executing the following command codes, any characters that match will be moved to the display buffer when System_TranslateString has been set to KEEP MODE. If set to DISCARD MODE, matching characters will be discarded.

<82h>C1C2 This command must always be followed by two user definable characters (C1 and C2). If the next input buffer character matches C1, replace it with C2.

<83h> If the next character is lower case ASCII convert it to upper case.

<84h> If the next character is upper case ASCII convert it to lower case.

Skip And Move Commands
All command code ranges shown below use the second hexadecimal digit (the least significant digit) to represent a number from 1 to 16. By adding one to this digit, a number from 1 to 16 is obtained (for example <90h> represents 1, <91h> represents 2, and <9Fh> represents 16).

<90h> to **<9Fh>** Skip over the specified number of characters (from 1 to 16 characters) in the input buffer.

<8Fh>C1 Skip all instances of the user definable character C1 in the input buffer until a different character or command code is found.
<E0h> to **<EFh>** Skip over characters in the input buffer until the specified match string (from 1 to 16 characters) following this command code is found in the input buffer. This command code is immediately followed by a the match string of the specified number of characters. A <NUL> cannot be part of the match string.

<A0h> to **<AFh>** Move the display buffer pointer to a specified location (from 1 to 16 where 1 represents the most significant digit).

<B0h> to **<BFh>** Move a specified number of characters (from 1 to 16) from the input buffer to the display buffer.

<C0h> to **<CFh>** Move a specified number of characters (from 1 to 16) from the System_TranslateString to the display buffer.

Character Matching Commands
When executing the following command codes, any characters that match will be moved to the display buffer when System_TranslateString has been set to KEEP MODE. If set to DISCARD MODE, matching characters will be discarded.

<01h> to **<7Fh>** The next character in the input stream must match the specific ASCII character represented by this command code.

<D2h> The next character in the input buffer must be a lower ASCII page character (<00h> to <7Fh>).

<D0h> The next character in the input buffer must be alphanumeric (0-9, A-Z or a-z).

<D1h> The next character in the input buffer must be alphabetic (A-Z, or a-z).

<D6h> The next character in the input buffer must be lower case alphabetic (a-z).

<DAh> The next character in the input buffer must be upper case alphabetic (A-Z).

<DBh> The next character in the input buffer must be a hexadecimal number (0-9, A-F or a-f).

<D4h> The next character in the input buffer must be a decimal number (0-9).

<D8h> The next character in the input buffer must be an ASCII symbol. Symbols are defined as [! " # $ % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ ` { | } ~ or].

<D3h> The next character in the input buffer must be a control code (<00h> to <1Fh> or <7Fh>).

<D9h> The next character in the input buffer must be an ASCII cursor movement control code (<09h> to <0Dh> or <20h>).

<D7h> The next character in the input buffer must NOT be a control code (<00h> to <1Fh> or <7Fh>).

<D5h> The next character in the input buffer must NOT be a control code or space (<00h> to <20h> or <7Fh>).

	Virtual Display Command Codes **<F3h>** The next two input buffer characters are used to set the virtual display number for Serial virtual display applications. **<F4h>** This command code points to the display buffer corresponding to the next virtual display listed in the System_DisplayList. **<F5h>** This command code reads the next input buffer character and interprets it as an explicit pointer to a display buffer corresponding to one of the virtual displays listed in the System_DisplayList. *Miscellaneous Command Codes That Directly Control Functions Of the Display* The following commands are used to control selected aspects of the 87 series unit operation. **<85h>** This command is followed by either A, B, C, D, E, F or G, and must be the last command code in the System_TranslateString. Each letter represents a special instruction to the display. A - Simulate input 1 on, with a result identical to an actual hardware input. B - Simulate input 1 off, with a result identical to an actual hardware input. C - Simulate input 2 on, with a result identical to an actual hardware input. D - Simulate input 2 off, with a result identical to an actual hardware input. E - Simulate reset input, with a result identical to an actual hardware input. F - Turn the optional relay on. G - Turn the optional relay off. **<86h>C1C2** This command is only used with 87/708 clock units, and must always be followed by two user definable characters (C1 and C2). If the next input buffer character matches C1, set the 87/708 clock to 12 hour mode, or if the next input buffer character matches C2, set the 87/708 clock to 24 hour mode. **<F0h>** Flash the display. **<F1h>** Do not flash the display. **<F2h>** The next X input buffer characters will be interpreted as instructions to set the color of the display (where X is equal to the number of display digits as set by the first number in the System_VirtualDisplaySize parameter). This command only has any result with tri-color display models. The four available colors are red, yellow, green and blank (represented respectively by the characters R, Y, G & B).
Example 1	Search for address 34. If first character is a +, substitute a space and display the five remaining characters. <87h>3 If first character is not a '3' then jump to 1. <87h>4 If second character is not a '4' then jump to 1. <8Ch> Else if buffer started with '34' then jump to 2. <89h><E3h><CR><LF>34 1. Search for <CR><LF>'34'. <8Ah><82h>+<SP> 2. If the next character is '+' make it a space <A0h><B6h><8Dh> Copy the next six characters to the start of the display buffer and display.
Example 2	Ignore first five characters and display next six. <94h><B6h>

Example 3	Search for address determine command (D-display, F-Flash, R-relay) accept six digits and display only the last four. 00 Match address '00'. <87h>R<85h>-----10<8Eh> If first character is 'R' then do relay output. <89h><87h>F<F0h><91h><B3h><8Dh>If first is 'F' then display last four flashing. <89h><87h>D<F1h><91h><B3h><8Dh>If first is 'D' the display last four steady. <89h><8eh> Ignore if first is not 'R', 'F', or 'D'.
Example 4	Search for the word 'NET', skip five characters and display the next six characters. <81h><E2h>NET Search for and discard the word 'NET'. <94h><B5h> Skip five then display six characters.

Programmable Output Strings

All 87 series displays have the ability to automatically transmit a string of data on a periodic and repeating basis. Two common applications for this are to periodically send data to a slave display, and to automatically periodically prompt a host device to transmit data to the display.

Parameter Name	System_OutputPeriod
Applies To	All units.
Format	Unsigned Word (number ranging from 0 to 65,535).
Default Setting	0
Description	This parameter controls the time interval between automatic transmission of the System_OutputString (see below). Each number increment represents 10 milliseconds (0.01 seconds), allowing an interval between automatic string transmissions of 10 milliseconds (setting of **1**) to 655.35 seconds (setting of **65535**). The default setting of **0** will cause the System_OutputString to be disabled (not transmitted). Automatic string transmissions have lower priority than other serial communications, and only occur when the display is not in program mode. Generally, the System_OutputPeriod should never be set lower than one tenth of a second? (setting of **10**), as it is possible to overrun the serial port (attempting to transmit characters more frequently than the display can reliably handle).
Examples	**1000** would represent an automatic string transmit interval of 10 seconds. In other words, every 10 seconds the System_OutputString will automatically be transmitted.

Parameter Name	System_OutputString0 System_OutputString1 System_OutputString2 System_OutputString3
Applies To	All units.
Format	String of 0 to 32 characters. Each character can be represented by its hexadecimal, decimal, literal, or mnemonic equivalent. Characters are limited to the lower page ASCII characters (<00h> to <7Fh>), and three special codes (<80h>, <81h>, and <82h>).
Default Setting	The default setting is blank (no characters).

Description	These parameters determine what character string is transmitted out the serial port each time the System_OutputPeriod expires (only System_OutputString0), or when a preset action triggers an output string (any of the four Output_String... parameters). **<00h>** to **<7Fh>** These are specific characters which, if included in the System_OutputString..., will be transmitted. **<80h>** This is a special code which if included in System_OutputString... will cause the value of a virtual display to be transmitted. It is followed by the number of the virtual display to transmit. The number ranges from 1 to 32. This value must be sent as a hex value (i.e. <01h>). **<81h>** This is a special code which if included in the System_OutputString... will cause the Simple Packet Protocol checksum to be transmitted. **<82h>** This is a special code which if included in System_OutputString... will cause the value of a process variable to be transmitted. It is followed by the code for the display list selector for this process variable. This value must be sent as a hex value (i.e. <01h>).
Examples	**<SOH>s:D<80h><01h><CR><81h>** is a 8 character System_OutputString... that will broadcast the display contents to other 87 series displays regardless of their address. In other words it sends a packet suitable for driving 87 series slave displays. It is broken down as follows (note the mix of all four methods of representing characters): <SOH> character 1 a control code mnemonic that starts a new SPP packet. s character 2 a literal that signals data for a group address. : character 3 a literal that ends the address part of the packet, and since it is not preceded by a number implicitly sets an address of 0 (a broadcast to all displays). D character 4 a literal that signals data to be displayed. <80h> character 5 a hexadecimal code that causes the display contents to be transmitted. <01h> character 6 the display to send is virtual display number 1. <CR> character 7 a control code mnemonic that terminates the packet. <81h> character 8 a hexadecimal code that causes a checksum to be added to the packet.

Watch Dog Timer

In some applications it may be desirable for the display screen to indicate if a serial transmission has not been received by the unit within a fixed programmable time interval. This kind of function is commonly referred to as a watch dog, as the unit is "guarding" against a failure in serial transmissions from the host device. Each time a valid serial transmission is received, the watch dog timer is automatically reset and again begins timing out the watch dog time period. This parameter is generally only useful for 87/232 units, because any message displayed by the watch dog timer will be overwritten by other display events (i.e. a count, a rate update, a time update etc.).

Parameter Name	System_WatchDogTime
Applies To	All units.
Format	Unsigned Word (number ranging from 0 to 65,535).
Default Setting	0
Description	This parameter controls the amount of time the unit will wait to receive a valid serial transmission before it displays the System_WatchDogError string. A serial transmission in PacketMode is considered valid only if the complete packet is valid. In other modes receipt of the correct line terminator is considered a valid serial transmission. Each number increment represents 10 milliseconds (0.01 seconds) allowing a time range of 10 milliseconds (setting of **1**) to 655.35 seconds (setting of **65535**). The default setting of **0** disables the watch dog timer.
Examples	**6000** represents a watch dog time of 1 minute (60 seconds).

Parameter Name	System_WatchDogError
Applies To	All units.
Format	String of 0 to 8 characters. Each character can be represented by its hexadecimal, decimal, literal or mnemonic equivalent.
Default Setting	Error
Description	This parameter defines the message that will be displayed if the System_WatchDogTime expires. The message will be displayed until another valid serial packet is received, the unit is reset, or another display screen event occurs.

Programmable Input Parameters

Introduction

Active Levels

The following parameters define the active level and debounce times of the programmable inputs on 87 displays.

Parameter Name	System_InputActiveLevels
Applies To	All units except 87/232 and 87/708.
Format	List of three items, one each from each column below. Input1Low Input1High Input2Low Input2High Input3Low Input3High
Default Setting	Input1Low + Input2Low + Input3Low
Description	This parameter sets the active level of the discrete hardware inputs. The 87/256, 87/415, 87/705, and 87/805 units each have three discrete hardware inputs. Each of these inputs takes one argument, and can be separately set as active low or active high with this parameter. The 87/712 unit has sixteen parallel inputs plus a strobe input. The Input1 parameter sets the active level for the strobe input. All sixteen of the data inputs are set as active low (Input3Low), or active high (Input3High) with the third parameter argument. The other parameter argument, Input2, is disregarded. The 87/719 has one auxiliary input. This input is set as active low (Input1Low), or active high (Input1High). with the first parameter argument. The other two parameter arguments are disregarded.
Examples	**Input1Low + Input2Low + Input3Low** sets all of the discrete hardware inputs to be active low (the 87/712 and 87/719 units disregard the second and/or third parameter arguments).

Debounce Times

Parameter Name	System_DebounceTime1 System_DebounceTime2 System_DebounceTime3
Applies To	All units except 87/232 and 87/708.
Format	Unsigned Byte (number ranging from 0 to 255).
Default Setting	0 0 48
Description	This parameter controls the debounce time of the discrete hardware inputs. The debounce time is the length of time that the 87 series display will wait for an input to settle before recognizing a state change in the input. Each number increment represents 208 microseconds, allowing debounce times from no debounce (setting of **0**) to 53 milliseconds (setting of **255**). System_DebounceTime3 defaults to 10 milliseconds (setting of **48**) as it generally is used as a reset input. Reset inputs are frequently activated by contact closures, and eliminating contact bounce often require a debounce time of approximately 10 milliseconds. Parallel no strobe mode is special - 0.01 seconds per click. System_DebounceTime2 and System_DebounceTime3 are disregarded by the 87/712 and 87/719 units.

Preset And Relay Operation Parameters

Introduction

All units other than 87/232 and 87/708 units allow two presets (a low preset and a high preset) to be set.

The Preset Operation Mode defines the basic operation of these two presets.

Preset Actions allow presets to control a number of functions when presets are reached including operation of the optional relay, as well as a variety of display screen operations.

87 series units also allow access to the optional relay through serial port commands. The relay is provided for annunciation purposes only, and is not designed to be used for control applications. Relay Sequences allow programmable cyclical and timed relay operations.

Type Of Preset Operation

Parameter Name	System_PresetMode
Applies To	87/256, 87/415, 87/705, 87/712, 87/719, 87/805 units.
Format	One item from the following six choices: None OneUp OneDown TwoUp TwoDown InBand
Default Setting	None
Description	This parameter determines the operation of the two available presets. **None** means that there will be no preset operation. **OneUp** means that there will be one preset, and it will be triggered only by moving through it in an upward direction. **OneDown** means that there will be one preset, and it will be triggered only by moving through it in a downward direction. **TwoUp** means that there will be two presets, and they will each be triggered only by moving through them in an upward direction. **TwoDown** means that there will be two presets, and they will each be triggered only by moving through them in a downward direction. **InBand** means that there will be two presets, the area between which will be considered as an in band area. Moving beyond this in band area (moving through the low preset in a downward direction, or moving through the high preset in an upward direction) will trigger the respective preset actions. This is a very common choice when monitoring a process variable which should stay between specific defined limits. 87/712 input modes 0,2 and 3 use presets. Input mode 1 does not (there is no stable number).

Setting A Preset Value

When the low or high presets are modified in the parameter file, special care must be taken to format them to match the stored process variable format, not the displayed process variable format. The 87 Series Hand Held Terminal and VDP4 software automatically match the formats correctly, but this must be done manually when working directly with a parameter file. Please

refer to the chapter on Displayed Process Variable Number Formatting for additional background information on number formatting.

Parameter Name	UserSetup_LowPreset UserSetup_HighPreset
Applies To	87/256, 87/415, 87/705, 87/712, 87/719, 87/805 units.
Format	Signed Quad Word (Number ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	0
Description	These parameters set the value of the low preset and high preset. This value is compared to the stored process variable, not the displayed process variable. Therefore the format must match the format of the stored process variable.
Examples	In an example where 100 inputs equals 10 on the display, if a preset action is desired when the display screen reaches 500, the preset parameter must be set to **5000** (matching the stored process variable, not the displayed process variable).

Actions When Presets Are Reached

Parameter Name	System_LowPresetOn System_LowPresetOff System_HighPresetOn System_HighPresetOff
Applies To	87/256, 87/415, 87/705, 87/712, 87/719, 87/805 units.
Format	List of six items, one each from the six groups shown below.
Default Setting	RelayNone + FlashNone + ReLoadNo + StopNo + ColorMaskNone + XmitStringNone
Description	These four parameters determine the actions that will be taken when either of the two presets (the low preset or the high preset) are reached. In order for the preset actions defined by these four parameters to be triggered, System_PresetMode must be selected to something other than None. Independant actions can be defined for when a preset is triggered, and for when a preset is "untriggered" (by the process variable moving through the preset in the opposite direction). For example, if System_PresetMode is set to OneUp, the low preset will be triggered when the process variable moves through it in an upward direction. If triggered in this fashion, the preset actions defined by the System_LowPresetOn parameter will be executed. If the process variable later moves back below the low preset, the preset actions defined by the System_LowPresetOff parameter will be executed. The specific preset actions that are available are shown below, separated by the six groups. *Relay Action Group (only useful if the optional relay is installed)* **RelayNone** means that the current relay operation will not be changed. **RelayOff** means that the relay will be turned off. **RelaySequenceA** means that the System_Relay0Sequence will be executed (relay sequences can be used to latch the relay, time it, or cycle it). **RelaySequenceB** means that the System_Relay1Sequence will be executed (relay sequences can be used to latch the relay, time it, or cycle it). *Flash Display Screen Action Group* **FlashNone** means that the current display screen flash mode will not be changed. **FlashOff** means that if the display screen is currently flashing, this flashing will be turned off. **FlashSlow** means that the display screen will be flashed at the slower flash rate. **FlashFast** means that the display screen will be flashed at the faster flash rate.

	Reload Action Group **ReloadNo** means that the process variable value will not be changed. **ReloadYes** means that the process variable value will be reset to the Initial Value (set with the UserSetup_Reload parameter). This preset action can be used to provide an automatic reset of the process variable value when the preset is reached. *Stop Display Screen Action Group* **StopNo** means that the display screen will continue showing changes to the process variable value. **StopYes** means that the display screen will freeze the process variable value until the display is reset. *Color Mask Action Group (usually used only for tri-color displays)* **ColorMaskNone** means that the current display color will not be changed. **ColorMask0** means that the display color will be set according to the System_ColorMask0 parameter. **ColorMask1** means that the display color will be set according to the System_ColorMask1 parameter. **ColorMask2** means that the display color will be set according to the System_ColorMask2 parameter. **ColorMask3** means that the display color will be set according to the System_ColorMask3 parameter. *Transmit String Action Group* **XmitStringNone** means that a character string will not automatically be transmitted out of the serial port when the preset is reached. **XmitString0** means that System_OutputString0 will be transmitted. **XmitString1** means that System_OutputString1 will be transmitted. **XmitString2** means that System_OutputString2 will be transmitted. **XmitString3** means that System_OutputString3 will be transmitted.
Examples	The following is an example that would execute relay sequence A, stop and flash the display screen value, and set the display color according to the color mask 0 setting (if the preset action is triggered). **RelaySequenceA + FlashFast + ReloadNo + StopYes + ColorMask0 + XmitStringNone**

Setting An Intial Value

Parameter Name	UserSetup_Reload
Applies To	87/256, 87/415, 87/705, and 87/805 units.
Format	Signed Quad Word (Number ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	0
Description	This parameter sets the initial value of a stored process variable, and thus must be in the correct format of the stored process variable. The UserSetup_Reload parameter is used to set the initial value of the Current Count stored process variable (87/256 or 87/415 units) or of the Timer/Lap Time stored process variable (87/705 units). The initial value is what will be placed in the stored process variable on reset, or on power up (if System_SaveAtPowerDown is selected as No). A common application is for down counters or down timers, to set the initial value that will be counted down from each time the unit is reset.

Programmable Relay Operation (Relay Sequences)

Parameter Name	System_Relay0Sequence System_Relay1Sequence
Applies To	All units.
Format	List of four items, one from each line below. RelayNone RelayOn RelayOff RelayCycle OnTime_ 0 OnTime_1 ... OnTime_255 OffTime_0 OffTime_1 ... OffTime_255 Cycles_0 Cycles_1 ... Cycles_255
Default Setting	RelayNone + OnTime_0 + OffTime_0 + Cycles_0
Description	These two parameters control the operation of the optional relay when one of the two available relay sequences is triggered by reaching a preset, or by receiving a relay sequence command through the serial port. The first of four list items determines the basic operation of the relay sequence. **RelayNone** means that the current relay operation will not be changed. **RelayOn** means that the relay will turn on (latched or timed) after an optional delay time. **RelayCycle** means that the relay will cycle on and off for a programmable number of cycles, and with a programmable on time and off time. The second of four list items is utilized only if RelayOn or RelayCycle are selected, and sets the on time for the relay. Each number increment represents 100 milliseconds (one tenth of a second) allowing an on time ranging from 0.1 seconds (setting of **OnTime_1**) to 25.5 seconds (setting of **OnTime_255**). The default setting of 0 (setting of **OnTime_0**) is special, and represents latching the relay on. The third of four list items is utilized only if RelayOn or RelayCycle are selected, and sets the off time for the relay. If RelayOn is selected it is interpreted as a delay time before the relay is turned on. Each number increment represents 100 milliseconds (one tenth of a second) allowing an off/delay time ranging from 0.1 seconds (setting of **OffTime_1**) to 25.5 seconds (setting of **OffTime_255**). The default setting of 0 (setting of **OffTime_0**) is special, and represents no delay time. The default setting of 0 would only be used if RelayOn is selected. The fourth of four list items is utilized only if RelayCycle is selected, and selects how many times the relay will cycle on and off. A number of relay cycles ranging from 1 (setting of **Cycles_1**) to 255 (setting of **Cycles_255**) can be selected. The default setting of 0 (setting of **Cycles_0**) is special, and represents cycling the relay until the unit is reset, or another relay command is issued.
Examples	**RelayOn + OnTime_0 + OffTime_0 + Cycles_0** will latch the relay on with no delay time. **RelayOn + OnTime_0 + OffTime_50 + Cycles_0** will latch the relay on after a five second delay. **RelayCycle + OnTime_10 + OffTime_10 + Cycles_50** will cycle the relay on and off 50 times with a 2 second cycle time (1 second on and 1 second off).

87/708 Real Time Clock System Parameters

Basic Operation Parameters

The 87/708 Real Time Clock System is an 87/232 serial input display with an added real time clock. The four parameters described below are specific to the 87/708 clock operation (although several of the parameters are also used by the Programmable Timer used in 87/705 timer, 87/256 count and 87/415 rate).

Adding date as a process variable is under consideration.

Parameter Name	System_FlashRightDecimal
Applies To	87/256, 87/415, 87/705, and 87/708 units.
Format	Number ranging from 0 to 4.
Default Setting	No
Description	This parameter allows the display screen's right most decimal point to be used as a flashing seconds indicator. This can be useful in four digit clock applications (where only hours and minutes are displayed). **0 (No)** means that the right most decimal point will not be used to indicate seconds. **1 (Yes)** means that the right most decimal point will flash once a second in display 0 of the display list. **2** means that the right most decimal point will flash once a second in display 1 of the display list. **3** means that the right most decimal point will flash once a second in display 2 of the display list. **4** means that the right most decimal point will flash once a second in display 3 of the display list.

Parameter Name	System_ShowColonAsDP
Applies To	87/256, 87/415, 87/705, and 87/708 units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter allows a standard display (no colons) to use decimal points to simulate colons. It makes no sense to set this parameter if the display screen is already in a clock format (has built in colons). **Yes** means that a decimal point will be lit between the hours and minutes, and with six digit display screens also between the minutes and seconds. **No** means that this parameter will have no effect.
Examples	If System_ShowColonAsDP is set to **Yes** in an 87/708 unit with a six digit standard display, twelve noon would be displayed as 12.00.00.

Parameter Name	System_ClockDisplay12Hours
Applies To	87/708 units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter controls whether the 87/708 clock will display time in a 24 hour time format or a 12 hour time format. **Yes** means that the clock will display time in a 12 hour time format. **No** means that the clock will display time in a 24 hour time format.

Parameter Name	UserSetup_ClockOffset
Applies To	87/708 units.
Format	Number ranging from -12 to 12.
Default Setting	0
Description	This parameter allows an offset to be added to the 87/708 internally kept time before it is displayed. A typical application is for multiple time zone/multiple clock setups, where it is desired to synchronize all of the clocks to the same time, and to program an offset in each individual clock to allow it to display time of day for a different time zone. The range of **-12** to **12** allows time of day for any time zone to be displayed.
Examples	If this parameter is set to **4**, and the internal 87/708 clock time is 7:34, it will display the time as 11:34.

Parameter Name	UserSetup_ClockShowSeconds
Applies To	87/708 units.
Format	Number ranging from 0 to 1.
Default Setting	0
Description	This parameter allows the seconds of the time value to be disabled from being displayed or serially transmitted. **0** means that the seconds will not be displayed or transmitted. **1** means that the seconds will be displayed and transmitted.

Parameter Name	UserSetup_ClockStripLeadZero
Applies To	87/708 units.
Format	Number ranging from 0 to 1.
Default Setting	0
Description	This parameter allows the leading zeros of a time value to be replaced with space characters. **0** means that the leading zeros will not be stripped. **1** means that the leading zeros will be stripped.

Parameter Name	UserSetup_ClockTransmitTime
Applies To	87/708 units.
Format	Number ranging from 0 to 1.
Default Setting	0
Description	This parameter allows the time value to be transmitted serially to remote displays once per second. **0** means that the time value will not be transmitted. **1** means that the time value will be transmitted once per second.

Input Operation And Timing Modes

Parameter Name	System_InputMode
Applies To	87/705 units.
Format	Number ranging from 0 to 9.
Default Setting	0
Description	This parameter determines the operation of the three 87/705 inputs, and in doing so sets up some of the basic timer operation. There are 10 modes of input operation, of which one is selected by this parameter. For clearer understanding some terms are defined first. *Initial Value* is the initial number from which timing will start (in most applications this will be left at the default value of 0). The *Initial Value* is set with the UserSetup_Reload parameter. The *Initial Value* can be a negative number. *Timer* is the 87/705 internal timer. Unless stated otherwise, the *Timer* value is shown on the 87/705 display screen. *Start* means the *Timer* will begin timing (from the *Initial Value*). *Pause* means the *Timer* has paused, but can be restarted (*Continued*). *Continue* means the *Timer* will continue timing from where it last *Paused.* *Stop* means the *Timer* has stopped and cannot be restarted except by *Reset* or *Power Up*. *Reset* means the *Timer* will be set to the *Initial Value*. *Power Up* means that power has been previously removed and will now be reapplied to the 87/705 unit. If the parameter System_SaveAtPowerDown is set to Yes, when power is removed the *Timer* value will be saved, and on power up the *Timer* will be ready to *Continue*. If System_SaveAtPowerDown is set to No, when power is removed the *Timer* value will be lost, and on power up the *Timer* will be ready to *Start* (from the *Initial Value*). The 10 modes of input operation are defined below. **0** Input 1 An active going edge will *Start* or *Continue* the *Timer*. Input 2 An active going edge will *Pause* the *Timer*. Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive). **1** Input 1 While held active will *Start* or *Continue* the *Timer*, while held inactive will *Pause* the *Timer*. Input 2 Not used. Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive). **2** Input 1 While held active will *Start* or *Continue* the *Timer*, while held inactive will *Pause* the *Timer*. Input 2 An active going edge will freeze the display screen at the current *Timer* value, while the *Timer* continues running in the background.

When the input is released (becomes inactive) the display screen will no longer be frozen, and will again show current *Timer* values.

Input 3 An active going edge will *Reset* the timer, which will remain in reset until the input is released (becomes inactive).

3 On *Power Up* the *Timer* will automatically *Start* or *Continue* running in the background, while the display screen is frozen on zero.

Input 1 An active going edge will *Stop* the timer, and show the *Timer* value on the display screen.

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*. When the input is released (becomes inactive) the *Timer* will automatically *Start.*

4 On *Power Up* the *Timer* will automatically *Start* or *Continue* running.

Input 1 Not used.

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*. When the input is released (becomes inactive) the *Timer* will automatically *Start.*

5 Input 1 The first active going edge will *Start* or *Continue* the *Timer*. The second active going edge will *Pause* the *Timer*. This pattern repeats (the third active going edge will *Start* or *Continue* the *Timer*, the fourth active going edge will *Pause* the *Timer*, etc).

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive).

6 Input 1 The first active going edge will *Start* or *Continue* the *Timer*. The second active going edge will *Stop* the *Timer*. The third active going edge will *Reset* the *Timer*. This pattern repeats with every three active going edges.

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive).

7 Input 1 An active going edge will *Reset* and *Start* the *Timer.* The *Timer* will *Stop* when the input is released (becomes inactive).

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released.

8 Input 1 The first active going edge will *Reset* the *Timer*, freeze the display screen, and *Start* the *Timer*. The second and subsequent active going edges will *Stop* the *Timer*, freeze the display screen at the current *Timer* value (the elapsed time since the previous input), *Reset* the *Timer*, and *Start* the next timing cycle.

Input 2 Not used.

Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive).

9 Input 1 An active going edge will *Start* the *Timer*, with no effect on the display screen.

Input 2 An active going edge will *Stop* the *Timer* and freeze the *Timer* value on the display screen.

Input 3 An active going edge will *Reset* the *Timer*, which will remain in reset until the input is released (becomes inactive).

Basic Operation Parameters

Parameter Name	System_TimingDirection
Applies To	87/705 units.
Format	Choice of Up or Down.
Default Setting	Up
Description	This parameter is used to select the timing direction of the 87/705 unit. **Up** means the unit will operate as an up timer. **Down** means the unit will operate as a down timer.

Parameter Name	System_TimerType
Applies To	87/256, 87/415, 87/705 units.
Format	One item from the seven choices outlined below.
Default Setting	TimerSeconds
Description	This parameter is used to select the time units that will be shown on the display screen. The available timing resolution is also listed. The available timing resolution is the smallest time unit that can meaningfully be shown on the display screen. The actual display screen format is determined by the parameters described in the Displayed Process Variable Number Formatting section. **TimerSeconds** means the display screen will show only seconds, with an available resolution of 10 milliseconds (0.01 seconds). **TimerMinutes** means that the display screen will show only minutes, with an available resolution of 0.001 minutes. **TimerHours** means that the display screen will show only hours, with an available resolution of 0.0001 hours. **TimerDays** means that the display screen will show only days, with an available resolution of 0.0001 days. **TimerMinutesSeconds** means that the display screen will show minutes and seconds, with an available resolution of 10 milliseconds (0.01 seconds). **TimerHoursMinutes** means that the display screen will show hours and minutes, with an available resolution of 0.001 minutes. **TimerHoursMinutesSeconds** means that the display screen will show hours, minutes and seconds, with an available resolution of 10 milliseconds (0.01 seconds). **7** means that the display screen will show days, hours, minutes and seconds, with an available resolution of 10 milliseconds (0.01 seconds). DDD HH:MM:SS. This mode requires the System_VirtualDisplaySize to be set to 37. Also, the System_DisplayMethod must be set to 2 and the System_DisplayList must be set to 16 + Timer + NotUsed + NotUsed

Parameter Name	System_FlashRightDecimal
Applies To	87/256, 87/415, 87/705 units.
Format	Choice of Yes, No, 0, 1, 2, 3, or 4.
Default Setting	No
Description	This parameter allows the display screen's right most decimal point to be used as a flashing seconds indicator, whenever the timer is running. This can be a useful feature for 87/705 units with a display screen resolution of minutes or greater, by indicating passage of seconds whenever the timer is running. **0 (No)** means that the right most decimal point will not be used to indicate seconds. **1 (Yes)** means that the right most decimal point will flash once a second in display 0 of the display list. **2** means that the right most decimal point will flash once a second in display 1 of the display list. **3** means that the right most decimal point will flash once a second in display 2 of the display list. **4** means that the right most decimal point will flash once a second in display 3 of the display list.

Parameter Name	System_ShowColonAsDP
Applies To	87/256, 87/415, 87/705 units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter allows a standard display (no colons) to use decimal points to simulate colons. It makes no sense to set this parameter if the display screen is already in a clock format (has built in colons). **Yes** means that whenever the System_TimerType parameter includes two or more types of time units, a decimal point will be lit between the hours, minutes and/or seconds. **No** means that this parameter will have no effect.
Examples	If System_ShowColonAsDP is set to **Yes** in an 87/705 unit with a five digit standard display, an elapsed time of 3 hours 12 minutes and 4 seconds would be displayed as 3.12.04.

Parameter Name	System_ShowSeparator
Applies To	87/705 units.
Format	Number ranging from 0 to 1.
Default Setting	1
Description	This parameter allows the decimal points between digits to be used as separators between hours and minutes and between minutes and seconds when displaying time values. This would be used when a time value needs to be shown on a standard display. A typical application would be displaying down time on a rate monitor display. **0** means that the decimal points will not be used as separators. **1** means that the decimal points will be used as separators.

87/256 Count & 87/415 Rate Display Parameters

The 87/256 and 87/415 models are nearly identical in operation. By using virtual displays (refer to the Virtual Displays section) either model can display count, rate and/or down time. The biggest difference between the two models is that 87/256 units use Current Count as the primary process variable, while 87/415 units use Current Rate as the primary process variable.

Input Operation And Count/Rate Modes

Parameter Name	System_InputMode
Applies To	87/256, 87/415 units.
Format	Number ranging from 0 to 10.
Default Setting	0
Description	This parameter determines the operation of two of the three 87/256 and 87/415 inputs (the third input is always *Reset*). An active going edge on Input 3 will cause a reset. The unit will remain in reset until the input is released (becomes inactive). Rate always uses input 1. For clearer understanding some terms are defined first. *Initial Value* is the initial number from which counting will start (in most applications this will be left at the default value of 0). The *Initial Value* is set with the UserSetup_Reload parameter. The *Initial Value* can be a negative number. *Reset* will set the Current Count PV to the *Initial Value*, and will set the the Minimum Rate PV, Maximum Rate PV, and Down Time PV to zero. *Scale Factor* when referring to Input 1 is the value set in the parameter UserSetup_CounterScale1, and when referring to Input 2 is the value set in the parameter UserSetup_CounterScale2. The default value for both parameters is 1. *Count Up* means that the *Scale Factor* will be added to the Current Count PV. *Count Down* means that the *Scale Factor* will be subtracted from the Current Count PV. *Quadrature* is a special counting mode, normally used with shaft encoders. Two inputs are always used (referred to as A and B), with the count signals phase shifted by 90 degrees. By determining which signal is rising first the 87 series unit can tell which direction the shaft encoder is turning, and thus *Count Up* or *Count Down* in the appropriate direction. *x1* is the standard *Quadrature* mode, as described above. x2 is a special *Quadrature* mode, where both edges of input A are counted. This doubles the effective resolution. *x4* is a special *Quadrature* mode, where both edges of input A and Input B are counted. This quadruples the effective resolution. **0** Input 1 An active going edge will *Count Up.* Input 2 While held active the display screen will show Current Rate (87/256 units) or Current Count (87/415 units). While held inactive the display screen will show Current Count (87/256 units) or Current Rate

(87/415 units).

1 Input 1 An active going edge will *Count Up.*
Input 2 An active going edge will *Count Up.*

2 Input 1 An active going edge will *Count Down.*
Input 2 An active going edge will *Count Down.*

3 Input 1 An active going edge will *Count Up.*
Input 2 An active going edge will *Count Down.*

4 Input 1 An active going edge will *Count Up* or *Count Down* depending on the state of Input 2.
Input 2 While inactive Input 1 will *Count Up*. While active Input 1 will *Count Down*.

5 Input 1 An active going edge will *Count Up*, but only if counting is enabled by Input 2.
Input 2 While inactive counting is enabled. While active counting is disabled.

6 *Quadrature x1*
Input 1 Phase A
Input 2 Phase B

Count Up
Input 1 Inactive going edge – Input 2 Inactive
Count Down
Input 1 Active going edge – Input 2 Inactive

7 *Quadrature x2*
Input 1 Phase A
Input 2 Phase B

Count Up
Input 1 Active going edge – Input 2 Active
Input 1 Inactive going edge – Input 2 Inactive
Count Down
Input 1 Active going edge – Input 2 Inactive
Input 1 Inactive going edge – Input 2 Active

8 *Quadrature x4*
Input 1 Phase A
Input 2 Phase B

Count Up
Input 1 Inactive – Input 2 Active going edge
Input 1 Active going edge – Input 2 Active
Input 1 Active – Input 2 Inactive going edge
Input 1 Inactive going edge – Input 2 Inactive
Count Down
Input 1 Inactive – Input 2 Inactive going edge
Input 1 Active going edge – Input 2 Inactive
Input 1 Active – Input 2 Active going edge
Input 1 Inactive going edge – Input 2 Active

	9 Input 1 An active going edge will *Count Up*. Input 2 Not used. **10** Input 1 An active going edge will *Count Up* Input 2 Each active going edge will step the display screen to the next of three process variables (the Current Count PV, the Current Rate PV and the Down Time PV).

Down Timer Operation

Parameter Name	System_RateShowDownTime
Applies To	87/256, 87/415 units.
Format	Choice Of Yes or No.
Default Setting	No
Description	This parameter determines whether the unit will automatically switch the display screen to show the Down Time PV whenever the Current Rate PV is less than or equal to the System_RatePreset parameter setting. **Yes** means that the unit will automatically switch to the Down Time PV display screen. **No** means that the unit will not automatically switch to the Down Time PV display screen.

Parameter Name	System_RatePreset
Applies To	87/256, 87/415 units.
Format	Signed Quad Word (integer ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	0
Description	When the Current Rate PV is less than or equal to System_RatePreset, down time will automatically be acumulated. The System_RatePreset parameter is compared to the stored Current Rate PV, not the displayed Current Rate PV. Therefore the format must match the format of the stored process variable (refer to the chapter on Displayed Process Variable Number Formatting for additional background information on number formatting).

Scale Factors

Parameter Name	UserSetup_CounterScale1 UserSetup_CounterScale2
Applies To	87/256, 87/415 units.
Format	Unsigned Long (integer ranging from 0 to 4,292,967,295).
Default Setting	10000

Description	These parameters are used to set scalers for Input 1 (UserSetup_CounterScale1) and Input 2 (UserSetup_CounterScale2). For each up count the scaler will be added to the Current Count PV. For each down count the scaler will be subtracted from the Current Count PV. The calculated Current Rate PV will be multiplied by the scaler before it is stored. In order to create fractional count or rate scaling (e.g. converting from feet to inches by using a scale factor of 0.0833) it is necessary to shift the decimal location by using display number formatting parameters. For the foot to inch conversion example, the UserSetup_CounterScale parameter would be set to **833**, and the rest of the conversion would be accomplished by the display number formatting parameters. Refer to the Displayed Process Variable Number Formatting section for additional background information on display number formatting. Keep in mind that this set of parameters on its own can only create positive integer scaling.
Examples	**1** means that up counts will add 1 (and down counts will subtract 1) from the Current Count PV. The Current Rate PV will be multiplied by 1 (in other words it will not be modified). **2** means that up counts will add 2 (and down counts will subtract 2) from the Current Count PV. The Current Rate PV will be multiplied by 2 before it is stored (in other words it will be doubled). **25** means that up counts will add 25 (and down counts will subtract 25) from the Current Count PV. The Current Rate PV will be multiplied by 25 before it is stored.

Rate Basic Operation Parameters

Parameter Name	System_RateMinimumTime
Applies To	87/256, 87/415, 87/805 units.
Format	This is a signed 32 bit number, ranging from -2,147,483,648 to -1.
Default Setting	-4,800
Description	This parameter controls the time interval between updates of the Current Rate PV. Each number increment represents 1/4800th of a second, allowing an update interval ranging from .2 milliseconds (setting of -**1**) to 447392.4267 seconds (setting of **–2,147,483,648**). Generally speaking, the longer the update time interval the more stable (and the less responsive) the Current Rate PV will become. This setting is thus very application specific. The System_RateMinimumTime should not be set lower than 100 milliseconds (setting of -**480**) as settings lower than this will take up excessive amounts of processing time while calculating rate.
Examples	**-4,800** represents a time interval between rate updates of 1 second. **-288,000** represents a time interval between rate updates of one minute (60 seconds).

Parameter Name	System_RateMaximumTime
Applies To	87/256, 87/415 units, 87/805 units.
Format	This is a signed 32 bit number, ranging from -2,147,483,648 to -1.
Default Setting	-4,800

Description	This parameter sets the time to zero (the amount of time without inputs before the Current Rate PV will be set to zero). Each number increment represents 1/4800th of a second, allowing an update interval ranging from .2 milliseconds (setting of -**1**) to 447392.4267 seconds (setting of **–2,147,483,648**). Generally speaking, applications where inputs are received at slow speeds should use longer times to zero. This setting is therefore very application specific.
Examples	**-4,800** represents a time to zero of 1 second. **-28,800** represents a time to zero of 6 seconds.

Parameter Name	System_RateDisplayFactor
Applies To	87/256, 87/415, 87/805 units.
Format	One item from the three choices outlined below.
Default Setting	RatePerSecond
Description	This parameter is used to select the time units for the Current Rate PV. **RatePerSecond** means that the CurrentRate PV will be calculated in units of rate per second. **RatePerMinute** means that the Current Rate PV will be calculated in units of rate per minute. **RatePerHour** means that the Current Rate PV will be calculated in units of rate per hour. The Current Rate PV can be further scaled with the UserSetup_CounterScale1 parameter, together with the display number formatting parameters. Refer to the Displayed Process Variable Number Formatting section for additional background information on display number formatting.

87/805 Production Efficiency Display Parameters

Setting An Automatic Goal Counter

Parameter Name	UserSetup_ProductionReload
Applies To	87/805 units.
Format	Signed Quad Word (Number ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	100
Description	This parameter can be used to set a time interval for automatically incrementing or decrementing the Pace Count PV. Setting this time interval, together with UserSetup_CounterScale2 actually has the effect of setting a goal for the rate of production (for example Pieces/Hour), since the combination of the settings corresponds to a certain number of counts per hour. Each number increment in this parameter represents 10 milliseconds (0.01 seconds), allowing the Production Goal PV to be incremented from once every 10 milliseconds (setting of **1**) to once every hour (setting of **360,000**) or even longer. Each time click will increment the display by the value set in UserSetup_CounterScale2. The setting for this parameter should be calculated as follows. 1) Determine the production goal in terms of Pieces/Hour. 2) Divide 360,000 by the result of Step 1, and round to the nearest whole number. The result is the optimum setting for the UserSetup_ProductionReload parameter. 3) Check your result by converting the setting for UserSetup_ProductionReload to Pieces/Hour and comparing it to the Pieces/Hour determined in Step 1 (divide 360,000 by the calculated UserSetup_ProductionReload setting). The two Pieces/Hour should be very close to each other. A value of **0** disables automatic increments of the Pace Count PV.
Examples	The following are example calculations of this parameter using the procedure outlined above.

1) Pieces/Hour	17	107	589	943	4,750
2) Optimum Setting	**21,176**	**3,364**	**611**	**382**	76
3) Check Result	17.00	107.02	589.20	942.41	4736.84

Parameter Name	UserSetup_CounterScale1 UserSetup_CounterScale2
Applies To	87/256, 87/415, 87/805 units.
Format	Unsigned Long (integer ranging from 0 to 4,292,967,295).
Default Setting	10000

Description	The operation described here applies to 87/805 units. These parameters are used to set scalers for Input 1 (UserSetup_CounterScale1) and the goal counter (UserSetup_CounterScale2). For each up count the scaler will be added to the ActualCount PV. For each down count the scaler will be subtracted from the ActualCount PV. The calculated Current Rate PV will be multiplied by the scaler before it is stored. In order to create fractional count or rate scaling (e.g. converting from feet to inches by using a scale factor of 0.0833) it is necessary to shift the decimal location by using display number formatting parameters. For the foot to inch conversion example, the UserSetup_CounterScale parameter would be set to **833**, and the rest of the conversion would be accomplished by the display number formatting parameters. Refer to the Displayed Process Variable Number Formatting section for additional background information on display number formatting. Keep in mind that this set of parameters on its own can only create positive integer scaling. UserSetup_CounterScale2 determines the increment value of the goal count (automatic or manual).

Input Operation And Display Modes

Parameter Name	System_InputMode
Applies To	87/805 units.
Format	Number ranging from 0 to 8.
Default Setting	0
Description	This parameter determines the operation of inputs 1 and 2, which of three process variables the display screen will show, and whether the goal count will be manually or automatically incremented. Input 3 is always the reset input. An active going edge on this input will reset the unit. Reset will clear the Actual Count PV, and the Goal Count PV to zero. The unit will remain in reset until the input is released (becomes inactive). To help make the presentation of this parameter as clear and consistent as possible some terms are defined below. *Actual Count* is the current production count, stored in the Actual Count PV. *Pace Count* is the current production goal count, stored in the Pace Count PV. It can be automatically incremented using the UserSetup_ProductionReload parameter (*Time Based Pace Count*), or it can be manually incremented using input 2 (*Input Based Pace Count*). *Suspend Pace Count* is an operating mode meant for temporary production breaks where it is desirable to suspend increases in the goal count. When in this mode the *Pace Count* and the display screen will freeze at their current value. Note that inputs to *Actual Count* will still be recognized. *Difference* is a calculated value (*Actual Count - Pace Count*). Positive values indicate that production is ahead of the goal. Negative values indicate that production is not meeting the goal.

	Ratio is a calculated value (*Actual Count* * 100)/ *Pace Count* . Values over 100 indicate that production is ahead of the goal. Values under 100 indicate that production is not meeting the goal. *Input Modes That Display Pace Count Using A Time Based Pace Count* **0** Input 1 Not used. Input 2 Not used. **6** Input 1 Not used. Input 2 While held active will *Suspend Pace Count*. *Input Modes That Display Pace Count Using An Input Based Pace Count* **3** Input 1 Not used. Input 2 An active going edge will increment *Pace Count*. *Input Modes That Display Net Count Using A Time Based Pace Count* **1** Input 1 An active going edge will increment *Actual Count*. Input 2 An active going edge will decrement *Actual Count*. **7** Input 1 An active going edge will increment *Actual Count.* Input 2 While held active will *Suspend Pace Count*. *Input Modes That Display Net Count Using An Input Based Pace Count* **4** Input 1 An active going edge will increment *Actual Count*. Input 2 An active going edge will increment *Pace Count*. *Input Modes That Display Efficiency Using A Time Based Pace Count* **2** Input 1 An active going edge will increment *Actual Count*. Input 2 An active going edge will decrement *Actual Count*. **8** Input 1 An active going edge will increment *Actual Count*. Input 2 While held active will *Suspend Pace Count*. *Input Modes That Display Efficiency Using An Input Based Pace Count* **5** Input 1 An active going edge will increment *Actual Count*. Input 2 An active going edge will increment *Pace Count*.

87/719 Analog Input Display Parameters

Basic Analog Operation Parameters

The normal update time is very fast, faster than the eye could follow.

Parameter Name	System_AnalogRange
Applies To	87/719 units.
Format	One item from the seven choices outlined below.
Default Setting	1ma
Description	This parameter is not actually directly used by the 87/719 unit, but is a place for the VDP4 software to keep track of the analog input range selected by the user. **1ma** means that the user has selected a 1 milliamp analog input in VDP4. **20ma** means that the user has selected a 20 milliamp analog input in VDP4. **20mv** means that the user has selected a 20 millivolt analog input in VDP4. **1V** means that the user has selected a 1 Volt analog input in VDP4. **5V** means that the user has selected a 5 Volt analog input in VDP4. **10V** means that the user has selected a 10 Volt analog input in VDP4. **100V** means that the user has selected a 100 Volt analog input in VDP4.

Parameter Name	System_AnalogUniPolar
Applies To	87/719 units.
Format	One item from the two choices outlined below.
Default Setting	Unipolar
Description	This parameter configures the analog to digital converter in the 87/719 to read only positive voltages, or both positive and negative voltages. **Unipolar** means that the 87/719 analog to digital converter will read only positive voltages or currents. **Bipolar** means that the 87/719 analog to digital converter will read both positive and negative voltages or currents.

Parameter Name	System_AnalogGain
Applies To	87/719 units.
Format	One item from the four choices outlined below.
Default Setting	Gainx1
Description	This parameter is used to set the gain of the 87/719 analog to digital converter's internal amplifier. Higher gain settings allow smaller changes in voltages and currents to be recognized. However, higher gain settings also increase the internal noise and decrease the overall system resolution. **Gainx1** means the analog to digital converter will be set to a gain of 1. **Gainx2** means the analog to digital converter will be set to a gain of 2. **Gainx32** means the analog to digital converter will be set to a gain of 32. **Gainx128** means the analog to digital converter will be set to a gain of 128.

Parameter Name	System_InputMode
Applies To	87/719 units.
Format	Number ranging from 0 to 1
Default Setting	0
Description	This parameter determines the operation of input 1, the only discrete logic level input available in the 87/719 unit. **0** Input 1 An active going edge will reset the primary process variable, and the peak registers. **1** Input 1 An active going edge will show the peak. When the edge goes inactive the peak will be reset. On reset (same as rate) the current values are transferred to the peak valley registers.

Analog Sampling Parameters

Parameter Name	System_AnalogSampleFactor
Applies To	87/719 units.
Format	Number ranging from 0 to 7.
Default Setting	0
Description	This parameter determines the number of analog samples that will be taken and averaged as one 87/719 analog to digital converter reading. Combining this setting with the System_AnalogShiftValue allows various over / under sampling schemes for special applications. The A/D converter grabs a handful of readings and adds then together. Each doubling of samples has the effect of adding another bit to the resolution. **0** means 1 sample will be taken. **1** means 2 samples will be taken. **2** means 4 samples will be taken. **3** means 8 samples will be taken. **4** means 16 samples will be taken. **5** means 32 samples will be taken. **6** means 64 samples will be taken. **7** means 128 samples will be taken.
Examples	If this parameter is set to a higher value than the System_AnalogShiftValue, then the difference in the two values indicates the number of extra bits over sampled. Eg. System_AnalogSampleFactor = 4, System_AnalogShiftValue = 2 will produce an extra (4-2), or 2, bits in the resultant reading. In addition, 16 samples will be added together before the System_AnalogShiftValue is applied. A larger number of samples results in a more stable result from oversampling. It is also possible to undersample by make the System_AnalogShiftValue larger than the System_AnalogSampleFactor. However, undersampling is only used in very specialized applications.

Parameter Name	System_AnalogShiftValue
Applies To	87/719 units.
Format	Number ranging from 0 to 7.
Default Setting	0
Description	This parameter allows discarding a set number of low end bits from the 87/719 analog to digital converter reading. Combining this setting with the System_AnalogSampleFactor allows various oversampling schemes for special applications. **0** means 0 low end bit will be discarded. **1** means 1 low end bits will be discarded. **2** means 2 low end bits will be discarded. **3** means 3 low end bits will be discarded. **4** means 4 low end bits will be discarded. **5** means 5 low end bits will be discarded. **6** means 6 low end bits will be discarded. **7** means 7 low end bits will be discarded.
Examples	If this parameter is set to a higher value than the System_AnalogShiftValue, then the difference in the two values indicates the number of extra bits over sampled. Eg. System_AnalogSampleFactor = 4, System_AnalogShiftValue = 2 will produce an extra (4-2), or 2, bits in the resultant reading. In addition, 16 samples will be added together before the System_AnalogShiftValue is applied. A larger number of samples results in a more stable result from oversampling. It is also possible to undersample by make the System_AnalogShiftValue larger than the System_AnalogSampleFactor. However, undersampling is only used in very specialized applications.

Parameter Name	System_SmoothingFactor
Applies To	87/719 units.
Format	Number ranging from 0 to 128.
Default Setting	0
Description	This parameter allows the Analog Value PV to be smoothed by using a sliding window average to average multiple A to D converter readings before they are displayed. If over sampling or under sampling is used, it acts as a preprocessor to the smoothing (in other words the end result of the Sample Factor and Shift Value calculation is always one “sample”, as used by the Smoothing Factor). Generally speaking, the more readings that are averaged, the more stable (and less responsive to change) the Analog Value PV will become. The setting of **0** and the setting of **1** are equivalent, both resulting in no averaging. Other numbers result in the selected number of A to D converter readings to be averaged before display.

Analog Scaling Parameters

Parameter Name	System_AnalogTopDisplay System_AnalogBottomDisplay
Applies To	87/719 units.
Format	Number ranging from -9,999,999 to 99,999,999.
Default Setting	0
Description	These parameters are not directly used by the 87/719 unit, but instead are a place for the VDP4 software to keep track of the bottom of range and top of range scaling selections entered by the user.

Parameter Name	System_AnalogSlopeY System_AnalogSlopeX System_AnalogYIntercept
Applies To	87/719 units.
Format	*SlopeY* Signed Long (number ranging from -2,147,483,648 to 2,147,483,647). *SlopeX* Unsigned Long (number 0 to 4,294,967,295). *Intercept* Signed Long (number ranging from -2,147,483,648 to 2,147,483,647).
Default Setting	0
Description	These parameters describe a linear equation which is used to scale the Analog Value PV before it is displayed. The scaled display value is created from the standard line definition equation: y=mx+b y is the scaled display value. m is the result of dividing System_AnalogSlopeY by System_AnalogSlopeX. x is the raw A to D convertor reading b is the System_AnalogYIntercept.
Examples	For SlopeY = 25 SlopeY = 10 Intercept = 35 If the ADC reading is 4012, then the linearized value is: (25 * 4012) / 10 + 35 or 351,050

87/712 Parallel Input Display Parameters

Input Operation And Display Modes

Parameter Name	System_InputMode
Applies To	87/712 units.
Format	Number ranging from 0 to 3.
Default Setting	0
Description	This parameter determines how the 16 input parallel port will be read into the Parallel Value PV. Keep in mind that the Display Formatter can change the manner in which the Parallel Value PV is displayed (e.g. values can be interpreted as signed, as hexadecimal, etc. when they are displayed). If BCD mode is selected, the System_DigitTranslate value is set to 0123456789-EHLP<SP>. **0** means the parallel port will be read as a 16 bit value and stored as a binary number in the Parallel Value PV. The stored value can thus range from 0 to 63,535. The D0 input represents the least significant bit of the 16 bit binary number, while the D15 input represents the most significant bit. **1** means that the parallel port will be read as multiplexed data and stored as an up to eight digit value in the Parallel Value PV. Inputs D0 through D3 are interpreted as four bit data (D0 is the least significant bit). In most applications this four bit data represents a BCD number, but hexadecimal numbers can also be loaded. Inputs D4 through D7 are not used. Inputs D8 through D15 are interpreted as digit select lines (D8 selects the least significant digit). Whenever a set of multiplexed data (four bit data plus a digit select line) is read it is stored in the corresponding digit position of the Parallel Value PV. Thus, to load an eight digit number into the Parallel Value PV, it is necessary to load eight sets of multiplexed data to the unit. The advantage is that this number can be loaded with just 12 I/O lines. *Pilot Light* is a mode of operation where the parallel port is read as discrete inputs. Each input is associated with a default value (D0 is associated with 1, D1 is associated with 2...D15 is associated with 16). An offset can be added to this default value (refer to the System_ParallelPilotBase parameter). If one or more inputs are active, the value associated with one (and only one) of the inputs is stored in the Parallel Value PV. The manner in which multiple simultaneous active inputs is handled defines the difference between the two modes described below. If no inputs are active, a 0 is stored in the Parallel Value PV. **2** means that the parallel port will operate in *Priority Pilot Light* mode. In this mode, the value associated with the highest priority active input is stored in the Parallel Value PV. D15 represents the highest priority, D14 represents the second highest priority...D0 represents the lowest priority. **3** means that the parallel port will operate in *Round Robin Pilot Light* mode. In this mode, the inputs are continuously scanned, starting with D0 and walking up to D15. Whenever an active input is found, the value associated with that input is moved to the Parallel Port PV, and the scanning pauses for a fixed amount of time (set by the System_ParallelPilotDelay parameter). The unit then resumes scanning inputs starting with the next higher input. The net effect is that the display screen will continuously step through the active input values. If any inputs are active, the relay output will be activated.

Parameter Name	System_ParallelStrobeMode
Applies To	87/712 units.
Format	Choice of Yes or No.
Default Setting	No
Description	This parameter controls whether a Strobe input is required to validate data on the parallel port. **Yes** means that active inputs present on the parallel port (D0 through D15) must be validated with an active Strobe input. **No** means that the strobe input is not used.

Parameter Name	System_ParallelPilotDelay
Applies To	87/712 units.
Format	Number ranging from 1 to 65,535.
Default Setting	100
Description	This parameter controls the time interval for each display screen step when multiple input values are displayed with the Round Robin Pilot Light mode. Each number increment represents 10 milliseconds (0.01 seconds), allowing a step time ranging from 10 milliseconds (setting of **1**) to 655.35 seconds (setting of **65,535**).
Examples	**100** would represent a step time of 1 second for each display screen step.

Parameter Name	System_ParallelPilotBase
Applies To	87/712 units.
Format	Unsigned Word (number ranging from 0 to 65,535).
Default Setting	0
Description	This parameter allows an offset to be added to the default value in either of the two Pilot Light modes of operation. This offset is added to the default value before it is stored in the Parallel Value PV. A default value is associated with each parallel port input (D0 is associated with 1, D1 is associated with 2...D15 is associated with 16).
Examples	**100** means that a validated input on D0 would actually result in 101 being stored in the Parallel Value PV.

87 Terminal Parameters

87 Display Identification

Parameter Name	System_UnitId
Applies To	All units.
Format	String of 0 to 11 characters. Each character can be represented by its hexadecimal, decimal, literal, or mnemonic equivalent.
Default Setting	The default setting is blank (no characters).
Description	This parameter allows assigning a short name or other character based designation to each 87 series unit. The System_UnitId can be requested through the 87 serial port. The System_UnitId is also automatically displayed on the 87 Terminal display screen whenever the terminal has opened a communication channel to an 87 series unit. This feature is especially useful for applications which interface one 87 Terminal to multiple 87 series displays.
Examples	**Line1 Count** would be an appropriate setting for a counter monitoring production line 1.

Controlling Terminal Access To Displays And Values

Parameter Name	System_Password
Applies To	All units.
Format	String of 0 to 8 characters. Each character can be represented by its hexadecimal, decimal, literal, or mnemonic equivalent.
Default Setting	The default setting is blank (no characters).
Description	This parameter allows setting a password for each 87 series unit. This password will be required any time access is attempted from an 87 Terminal. The password will be required to read or write any of the *User Setup* parameters. This parameter only affects operation with the 87 Terminal. It is the terminal that prevents entry, not the unit. Although the password can be set to any characters, VDP4 restricts the entry to characters that can be generated by the terminal: 0 1 2 3 4 5 6 7 8 9 0 Space . : -

Parameter Name	System_EN_Display System_EN_Reload System_EN_LowPreset System_EN_HighPreset System_EN_ProductionReload System_EN_CounterScale1 System_EN_CounterScale2 System_EN_ClockOffset
Applies To	All units.
Format	Choice of Yes or No.
Default Setting	Yes
Description	These parameters control which 87 User Setup parameters can be written by the 87 Terminal. **Yes** means that the selected parameter can be written by the 87 Terminal. **No** means that the selected parameter can only be read by the 87 Terminal.

Restricting The Range Of User Setup Input Values

The following parameters are used to set low and high limit values for the corresponding User Setup parameter values. The System_LL... parameters set low limits, while the System_HL... parameters set high limits. These limits are only enforced when the parameters are written from an 87 Terminal.

Parameter Name	System_LL_Reload System_HL_Reload
Applies To	87/256, 87/415, 87/705, 87/805 units.
Format	Signed Quad Word (Number ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	-99,999,999 99,999,999
Description	These parameters set low and high limit values for the UserSetup_Reload parameter. The unit will not accept values outside the range set here from an 87 Terminal.

Parameter Name	System_LL_LowPreset System_LL_High Preset System_HL_LowPreset System_HL_High Preset
Applies To	All units.
Format	Signed Quad Word (Number ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	-99,999,999 99,999,999
Description	These parameters set low and high limit values for the UserSetup_LowPreset and UserSetup_HighPreset parameters. The unit will not accept values outside the range set here from an 87 Terminal.

Parameter Name	System_LL_ProductionReload System_HL_ProductionReload
Applies To	87/805 units.
Format	Signed Quad Word (integer ranging from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807).
Default Setting	1 99,999,999
Description	These parameters set low and high limit values for the UserSetup_ProductionReload parameter. The unit will not accept values outside the range set here from an 87 Terminal.

Parameter Name	System_LL_CounterScale1 System_LL_CounterScale2 System_HL_CounterScale1 System_HL_CounterScale2
Applies To	87/256, 87/415, 87/805 units.
Format	Unsigned Long (integer ranging from 0 to 4,292,967,295).
Default Setting	1 99,999,999
Description	These parameters set low and high limit values for the UserSetup_CounterScale1 and UserSetup_CounterScale2 parameters. The unit will not accept values outside the range set here from an 87 Terminal.

Parameter Name	System_LL_ClockOffset System_HL_ClockOffset
Applies To	87/708 units.
Format	Integer ranging from -12 to 12.
Default Setting	-12 12
Description	These parameters set low and high limit values for the UserSetup_ClockOffset parameter. The unit will not accept values outside the range set here from an 87 Terminal.

Setting Up User Function Keys

Parameter Name	System_UserDescription1 System_UserDescription2 System_UserDescription3 System_UserDescription4
Applies To	All units.
Format	String of 0 to 20 characters. Each character can be represented by its hexadecimal, decimal, literal, or mnemonic equivalent.
Default Setting	The default setting is blank (no characters).
Description	This parameter allows assigning a line of text which will be displayed on the 87 terminal when the selected function key is pressed. The line of text should be descriptive of the function that is accessed. This line of text is displayed on the third line of the terminal after one of the function buttons is pressed.
Examples	If System_UserFunction1 is set to **I1**, then a good choice for System_UserDescription1 is 'Jog Count'.

Parameter Name	System_UserFunction1 System_UserFunction2 System_UserFunction3 System_UserFunction4
Applies To	All units.
Format	String of 0 to 8 characters. Each character can be represented by its hexadecimal, decimal, literal, or mnemonic equivalent.
Default Setting	The default setting is blank (no characters).
Description	This parameter allows assigning a specific function to one of the four available user function keys available on the 87 Terminal. The function is assigned by listing a valid Simple Packet Protocol command that corresponds to the function.
Examples	**R1** means the relay would be turned on when the function key is pressed. **CRRRRRRRR** would set color mask 0 to all red. **I1** issues an input edge for input 1.

Reserved Parameters

Reserved parameters are special use parameters that should never be modified. They include error checking parameters, and unit identification parameters. Both types of parameters are necessary in order for the unit to function correctly.

Error Checking Parameters

Parameter Name	System_CRC UserSetup_CRC
Applies To	All units.
Format	Unsigned Word (integer ranging from 0 to 65,535).
Default Setting	There are no defaults for these parameters.
Description	DO NOT MODIFY THESE PARAMETERS! These parameters are used by the 87 unit to check the integrity of the User Setup (UserSetup_CRC) and System Setup (System_CRC) data.

Unit Identification Parameters

The following parameters are used to allow the unit to identify itself correctly to the programming system.

Parameter Name	System_Personality
Applies To	All units.
Format	One item from the eight choices listed below.
Default Setting	Serial
Description	DO NOT MODIFY THIS PARAMETER! This parameter selects the basic personality of the unit. Note that appropriate hardware is required for the unit to function correctly within it's personality. **Serial** means that this unit is an 87/232 Serial Input Display. **Clock** means that this unit is an 87/708 Real Time Clock System. **Timer** means that this unit is an 87/705 Programmable Timer. **Counter** means that this unit is an 87/256 Count Display Plus. **RateMonitor** means that this unit is an 87/415 Rate Display Plus. **ProductionMonitor** means that this unit is an 87/805 Production monitor Display. **Analog** means that this unit is an 87/719 Analog Input Display. **Parallel** means that this unit is an 87/712 Parallel Input Display. 87/232, 87/708, 87/712 and 87/719 units share a common hardware platform, based on the 87/232 unit hardware. Adding a real time clock chip creates the hardware necessary for the 87/708 unit. Adding a special daughter board creates the hardware necessary for the 87/712 unit (a parallel input daughter card) or the 87/719 unit (an analog input daughter card). 87/256, 87/415, 87/705 and 87/805 units all share a common and identical hardware platform. Therefore these units can be reprogrammed in the field to take on the listed alternate personalities.

Parameter Name	System_Family
Applies To	All units.
Format	Single character
Default Setting	B
Description	DO NOT MODIFY THIS PARAMETER! This parameter identifies the product family to a computer being used to program the unit. The entire 87 series product family has been assigned the character **B**.

Parameter Name	System_MAJ_VER System_MIN_REV System_REV_STG System_REV_REL
Applies To	All units.
Format	One of four characters (. α β δ) *System_REV_STG* Integer ranging from 0 to 255 *Others*
Default Setting	There are no defaults for these parameters.
Description	DO NOT MODIFY THESE PARAMETERS! These four parameters make up the firmware release version for this unit. This firmware release version will be used in the future to indicate changes which affect parameter operation. This information may be used by a computer to ensure proper programming of the unit