
Chapter 12

REMOTE_VARS

Edition 3

OVERVIEW

RMT_BOOL	12-1
Functional Description	12-1
Function Block Attributes	12-1
Parameter Descriptions	12-2
Parameter Attributes	12-4
RMT_REAL	12-5
Functional Description	12-5
Function Block Attributes	12-5
Parameter Descriptions	12-6
Parameter Attributes	12-8
RMT_INT	12-9
Functional Description	12-9
Function Block Attributes	12-9
Parameter Descriptions	12-10
Parameter Attributes	12-13
RMT_STR	12-14
Functional Description	12-14
Function Block Attributes	12-14
Parameter Descriptions	12-15
Parameter Attributes	12-17

Contents (continued)

RMT_SW	12-18
Functional Description	12-19
Function Block Attributes	12-19
Parameter Descriptions	12-19
Parameter Attributes	12-22
RMT_****_8	12-23
Functional Description	12-23
RMT_****_64.....	12-25
Functional Description	12-25

OVERVIEW

REMOTE parameters are used in association with a suitable MASTER communications driver. They are used when the PC3000 is the MASTER device. The blocks are so named because data belonging to the communicating slave device appears to the PC3000 as remote. The blocks may be viewed as fixed addresses or 'letterboxes' into which data may be written or read by the PC3000. The blocks provide diagnostics, read/write control, time stamping, etc.

The REMOTE multi-segment parameters allow more efficient memory usage since one block can be used to control the transactions associated with several different values. Additionally, there are benefits in speed of initialisation since only one address has to be set to its' cold start value compared with several when using individual parameters.

Boolean, Real, Integer, Time , String and Status_Word

Remote Parameter
Rmt_Bool
Rmt_Real
Rmt_Dint
Rmt_Time
Rmt_Str
Rmt_Bool_8
Rmt_Real_8
Rmt_Dint_64
Rmt_Bool_64
Rmt_Real_64
Rmt_Dint_64
Rmt_SW

This class is used in conjunction with the COMMS Function Block class.

For further information refer to the PC3000 Communications Overview in chapter 3.

Note 1: Prior to the release of version 2.27 system firmware these blocks were called Remote_****. Version 2.27 added a number of new multi-element remote parameters and all blocks in this class were renamed, All blocks are now identified as Rmt_**** where **** can be BOOL, REAL, DINT or STR.

This has no effect on programs generated using earlier versions.

Note 2: Not all protocols support the multi-element remote parameters . For version 2.27 usage is:

Driver	Multi-element parameter support
El_Bisync_M	3
Raw_Comms	N/A
JBus_M	8
Siemens_M_S	8
Toshiba_M	8

RMT_BOOL FUNCTION BLOCK

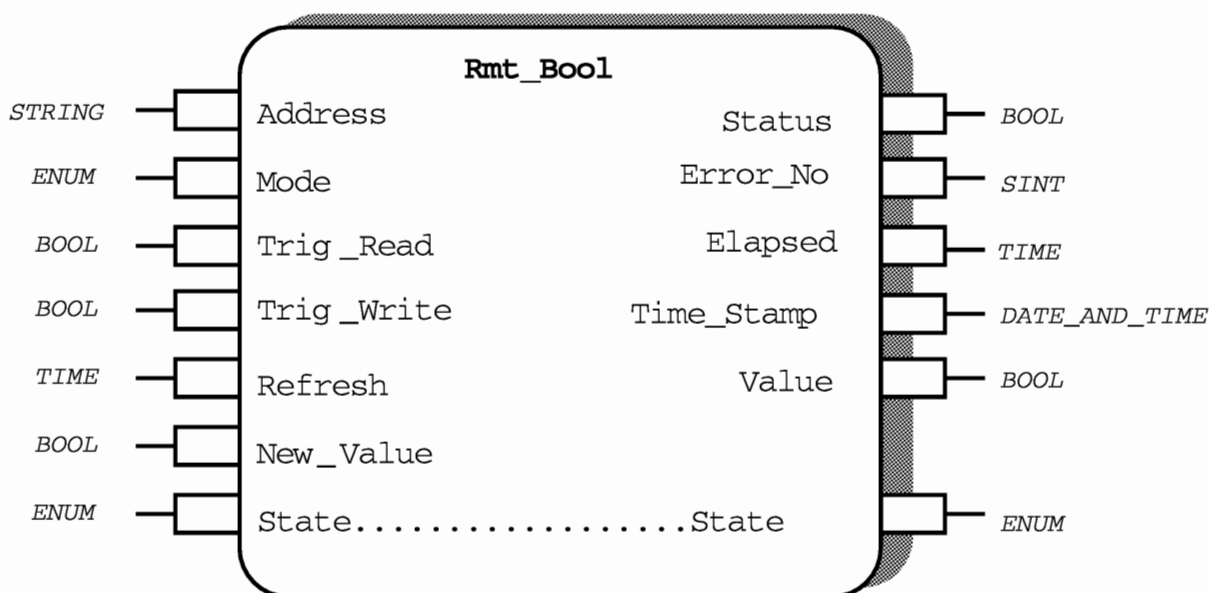


Figure 12-1 Rmt_Bool Variable Function Block

Functional Description

The Rmt_Bool Variable function block allows the reading and / or writing of the value of a boolean parameter via a communications port. The function block is not communications protocol specific, but can only be used with ports which support protocols which function in the Master mode.

For a full description of the use of Rmt_Vars function blocks, see the PC3000 Communications Overview in chapter 3.

Function Block Attributes

Type:..... 3A 10
 Class:REMOTE_VARS
 Default_Task: Task_2
 Short List: Value, New_Value, Mode, State
 Memory Requirements: 88 Bytes
 Execution Time: 128 μ Secs

Parameter Descriptions

Address (A)

This parameter is a string which is used to associate the function block with a parameter in the remote device to which the PC3000 is connected. The first character of Address defines the PC3000 module on which the communications port is situated. The second character defines the Port on the module. The rest of the address is protocol specific and is used to identify the instrument and parameter that the function block is communicating with. Refer to the associated Master communications function block for details of address format.

Mode (M)

The Mode specifies mode of operation of the function block. It can have four possible values:

- Demand (0): In Demand mode, a single remote read or write can be triggered by changing the State parameter from OK(0) to Read(4) or Write(3).
- R_Cont (1): In R_Cont mode, the remote variable will be continuously polled at regular intervals defined by Refresh.
- W_Cont (2): In W_Cont mode, the remote variable will be continuously written to at regular intervals defined by Refresh.
- Change (3): In Change mode, the remote variable is written to when the value of the New_Value input does not match that of Value. The minimum period between changes being written to the remote device is defined by Refresh.

Trig_Read (TR)

When the function block is operating in Demand mode, changing Trig_Read from Off (0) to On (1) will cause a remote variable read to take place. This input is provided to permit a Read to be triggered by soft-wiring.

Trig_Write (TW)

When the function block is operating in Demand mode, changing Trig_Write from Off (0) to On (1) will cause a remote variable write to take place. This input is provided to permit a Write to be triggered by soft-wiring.

Refresh (R)

Refresh defines the period between update of the parameter when the function block is in R_Cont or W_Cont modes.

New_Value (NV)

New_Value is the value to be written to the remote variable, when the function block is in a write mode.

State (S)

The parameter State indicates the current state of the function block. It can have one of five values:

- Ok (0): No communications actions are currently taking place.
- Pending (1): A transaction has started and has not yet been completed.
- Error (2): A communications transaction has failed. The parameter Error_No provides a diagnostic indicator to the type of error.
- Write (3): Setting State to Write (3) triggers a single write of New_Value to the remote device.
- Read (4): Setting State to Read (4) triggers a single read of the remote parameter.

When State is set to Write(3) or Read(4) it will automatically revert to OK(0) if the transmission was successful or remain in the Error(2) state.

Status (ST)

If Status is set to Go (1), the previous communications request has been completed successfully. If Status is set to NOGO (0), an error has occurred.

Error_No (ERR)

Error_No is a diagnostic parameter which provides an indication to the type of communications error which has occurred. Reference should be made to appropriate Master communications driver in Chapter 3 for error codes.

Elapsed (E)

Elapsed indicates the amount of time that has passed since the last read or write transaction. Elapsed is only valid when the function block is operating in R_Cont or W_Cont modes.

Time_Stamp (TS)

Time_Stamp defines the time and date at which the last read or write transaction took place successfully. It is set coincident with the State parameter changing to Ok (0).

Value (VAL)

The Value parameter contains the last value which was successfully written to or read from the remote variable.

Parameter Attributes

Name	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Address	STRING		Oper	Config		
Elapsed	TIME	0	Oper			
Error_No	SINT	0	Oper		High Limit Low Limit	255 0
Mode	ENUM	Demand (0)	Oper	Super	Senses	Demand (0) R_Cont (1) W_Cont (2) Change (3)
New_Value	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Refresh	TIME	10s	Oper	Super	High Limit Low Limit	19d_59m_59s 100ms
State	ENUM	Ok (0)	Oper	Oper	Senses	Ok (0) Pending (1) Error (2) Write (3) Read (4)
Status	BOOL	Go (1)	Oper		Senses	NOGO (0) Go (1)
Time_Stamp	DATE AND TIME	0	Oper			
Trig_Read	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Trig_Write	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Value	BOOL	Off (0)	Oper		Senses	Off (0) On (1)

Table 12-1 Remote_Boot Parameter Attributes

RMT_REAL FUNCTION BLOCK

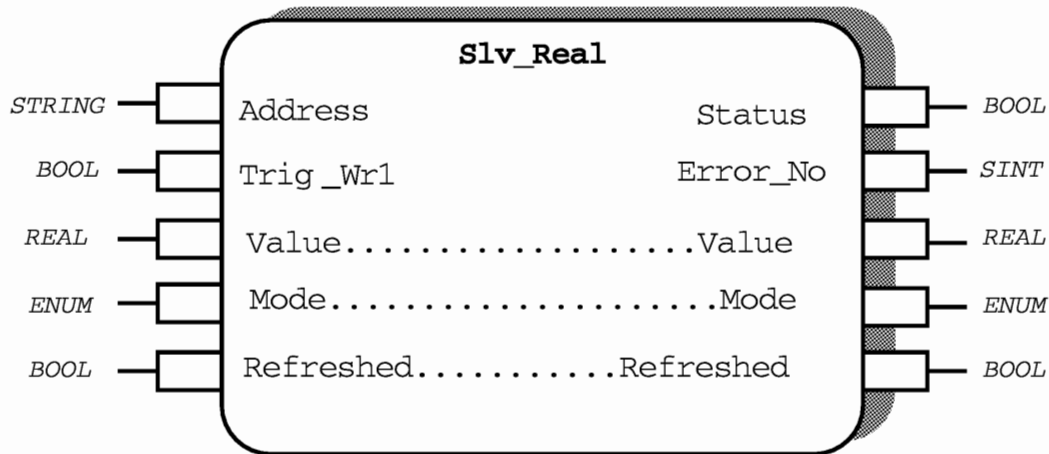


Figure 12-2 Rmt_Real Variable Function Block

Functional Description

The Rmt_Real Variable function block allows the reading and / or writing of the value of a real parameter via a communications port. The function block is not communications protocol specific, but can only be used with ports which support protocols which function in the Master mode.

For a full description of the use of Rmt_Vars function blocks, see the PC3000 Communications Overview n chapter 3.

Function Block Attributes

Type: 3A 20
 Class: REMOTE_VARS
 Default_Task: Task_2
 Short List: Value, New_Value, Mode, State
 Memory Requirements: 98 Bytes
 Execution Time: 129 μ Secs

Parameter Descriptions

Address (A)

This parameter is a string which is used to associate the function block with a parameter in the remote device to which the PC3000 is connected. The first character of Address defines the PC3000 module on which the communications port is situated. The second character defines the Port on the module. The rest of the address is protocol specific and is used to identify the instrument and parameter that the function block is communicating with.

Mode (M)

The Mode specifies mode of operation of the function block. It can have four possible values:

- Demand (0): In Demand mode, a single remote read or write can be triggered by changing the State parameter from OK(0) to Read(4) or Write(3).
- R_Cont (1): In R_Cont mode, the remote variable will be continuously polled at regular intervals defined by Refresh.
- W_Cont (2): In W_Cont mode, the remote variable will be continuously written to at regular intervals defined by Refresh.
- Change (3): In Change mode, the remote variable is written to when the value of the New_Value input does not match that of Value. the minimum period between changes being written to the remote device is defined by Refresh.

Trig_Read (TR)

When the function block is operating in Demand mode, changing Trig_Read from Off (0) to On (1) will cause a remote variable read to take place. This input is provided to permit a Read to be triggered by soft-wiring.

Trig_Write (TW)

When the function block is operating in Demand mode, changing Trig_Write from Off (0) to On (1) will cause a remote variable write to take place. This input is provided to permit a Write to be triggered by soft-wiring.

Refresh (R)

Refresh defines the period between update of the parameter when the function block is in R_Cont or W_Cont modes.

New_Value (NV)

New_Value is the value to be written to the remote variable, when the function block is in a write mode.

State (S)

The parameter State indicates the current state of the function block. It can have one of five values:

- Ok (0): No communications actions are currently taking place.
- Pending (1): A transaction has started and has not yet been completed.
- Error (2): A communications transaction has failed. The parameter Error_No provides a diagnostic indicator to the type of error.
- Write (3): Setting State to Write (3) triggers a single write of New_Value to the remote device.
- Read (4): Setting State to Read (4) triggers a single read of the remote parameter.

When State is set to Write(3) or Read(4) it will automatically revert to OK(0) if the transmission was successful or remain in the Error(2) state.

Status (ST)

If Status is set to Go (1), the previous communications request has been completed successfully. If Status is set to NOGO (0), an error has occurred.

Error_No (ERR)

Error_No is a diagnostic parameter which provides an indication to the type of communications error which has occurred. Reference should be made to appropriate Master communications driver in Chapter 3 for error codes.

Elapsed (E)

Indicates the amount of time that has passed since the last read or write transaction. Elapsed is only valid when the function block is operating in R_Cont or W_Cont modes.

Time_Stamp (TS)

Time_Stamp defines the time and date at which the last read or write transaction took place successfully. It is set coincident with the State parameter changing to Ok (0).

Value (VAL)

This parameter contains the last value which was successfully written to or read from the remote variable.

Parameter Attributes

Name	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Address	STRING		Oper	Config		
Elapsed	TIME	0	Oper			
Error_No	SINT	0	Oper		High Limit Low Limit	255 0
Mode	ENUM	Demand (0)	Oper	Super	Senses	Demand (0) R_Cont (1) W_Cont (2) Change (3)
New_Value	REAL	0	Oper	Oper	High Limit Low Limit	10,000,000 -10,000,000
Refresh	TIME	10s	Oper	Super	High Limit Low Limit	19d_59m_59s 100ms
State	ENUM	Ok (0)	Oper	Oper	Senses	Ok (0) Pending (1) Error (2) Write (3) Read (4)
Status	BOOL	Go (1)	Oper		Senses	NOGO (0) Go (1)
Time_Stamp	DATE AND TIME	0	Oper			
Trig_Read	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Trig_Write	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Value	REAL	0	Oper		High Limit Low Limit	10,000,000 -10,000,000

Table 12-2 Remote_Real Variable Parameter Attributes

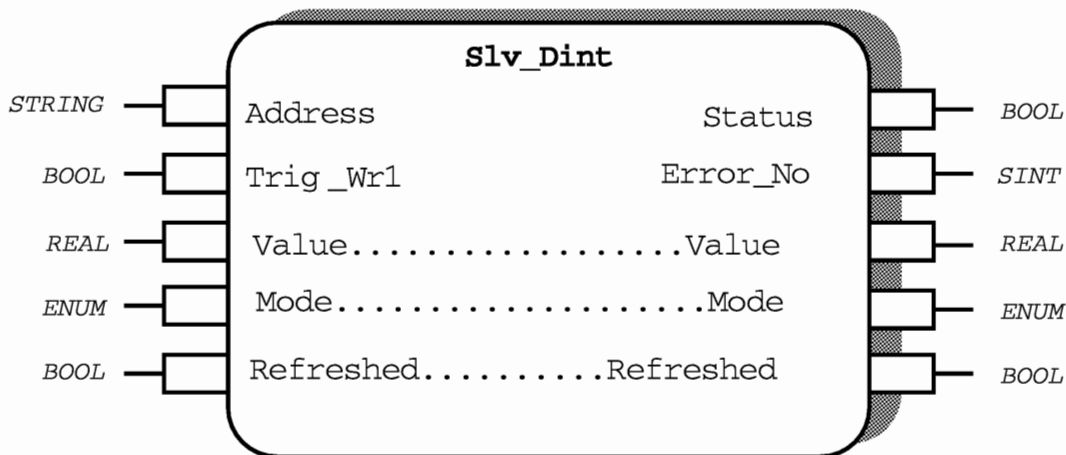
RMT_INT FUNCTION BLOCK:

Figure 12-3 Rmt_Int Variable Function Block

Functional Description

The Rmt_Int Variable function block allows the reading and / or writing of the value of a 32-bit integer variable via a communications port. The function block is not communications protocol specific, but can only be used with ports which support protocols which function in the Master mode.

For a full description of the use of Rmt_Vars function blocks, see the PC3000 Communications Overview in chapter 3.

Function Block Attributes

Type: 3A 30
 Class: REMOTE_VARS
 Default_Task: Task_2
 Short List: Value, New_Value, Mode, State
 Memory Requirements: 98 Bytes
 Execution Time: 124 μ Secs

Parameter Descriptions

Address (A)

This parameter is a string which is used to associate the function block with a parameter in the remote device to which the PC3000 is connected. The first character of Address defines the PC3000 module on which the communications port is situated. The second character defines the Port on the module. The rest of the address is protocol specific and is used to identify the instrument and parameter that the function block is communicating with.

Mode (M)

The Mode specifies mode of operation of the function block. It can have four possible values:

- Demand (0): In Demand mode, a single remote read or write can be triggered by changing the State parameter from OK(0) to Read(4) or Write(3).
- R_Cont (1): In R_Cont mode, the remote variable will be continuously polled at regular intervals defined by Refresh.
- W_Cont (2): In W_Cont mode, the remote variable will be continuously written to at regular intervals defined by Refresh.
- Change (3): In Change mode, the remote variable is written to when the value of the New_Value input does not match that of Value. The minimum period between changes being written to the remote device is defined by Refresh.

Trig_Read (TR)

When the function block is operating in Demand mode, changing Trig_Read from Off (0) to On (1) will cause a remote variable read to take place. This input is provided to permit a Read to be triggered by soft-wiring.

Trig_Write (TW)

When the function block is operating in Demand mode, changing Trig_Write from Off (0) to On (1) will cause a remote variable write to take place. This input is provided to permit a Write to be triggered by soft-wiring.

Refresh (R)

Refresh defines the period between update of the parameter when the function block is in R_Cont or W_Cont modes.

New_Value (NV)

This is the value to be written to the remote variable, when the function block is in a write mode.

State (S)

The parameter State indicates the current state of the function block. It can have one of five values:

- Ok (0): No communications actions are currently taking place.
- Pending (1): A transaction has started and has not yet been completed.
- Error (2): A communications transaction has failed. The parameter Error_No provides a diagnostic indicator to the type of error.
- Write (3): Setting State to Write (3) triggers a single write of New_Value to the remote device.
- Read (4): Setting State to Read (4) triggers a single read of the remote parameter.

When State is set to Write(3) or Read(4) it will automatically revert to OK(0) if the transmission was successful or remain in the Error(2) state.

Status (ST)

If Status is set to Go (1), the previous communications request has been completed successfully. If Status is set to NOGO (0), an error has occurred.

Error_No (ERR)

Error_No is a diagnostic parameter which provides an indication to the type of communications error which has occurred. Reference should be made to appropriate Master communications driver in Chapter 3 for error codes.

Elapsed (E)

Elapsed indicates the amount of time that has passed since the last read or write transaction. Elapsed is only valid when the function block is operating in R_Cont or W_Cont modes.

Time_Stamp (TS)

Defines the time and date at which the last read or write transaction took place successfully. It is set coincident with the State parameter changing to Ok (0).

Value (VAL)

The Value parameter contains the last value which was successfully written to or read from the remote variable.

Parameter Attributes

Name	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Address	STRING		Oper	Config		
Elapsed	TIME	0	Oper			
Error_No	SINT	0	Oper		High Limit Low Limit	255 0
Mode	ENUM	Demand (0)	Oper	Super	Senses	Demand (0) R_Cont (1) W_Cont (2) Change (3)
New_Value	DINT	0	Oper	Oper	High Limit Low Limit	10,000,000 -10,000,000
Refresh	TIME	10s	Oper	Super	High Limit Low Limit	19d_59m_59s 100ms
State	ENUM	Ok (0)	Oper	Oper	Senses	Ok (0) Pending (1) Error (2) Write (3) Read (4)
Status	BOOL	Go (1)	Oper		Senses	NOGO (0) Go (1)
Time_Stamp	DATE AND TIME	0	Oper			
Trig_Read	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Trig_Write	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Value	DINT	0	Oper		High Limit Low Limit	10,000,000 -10,000,000

Table 12-3 Remote_Int Variable Parameter Attributes

RMT_STR FUNCTION BLOCK:

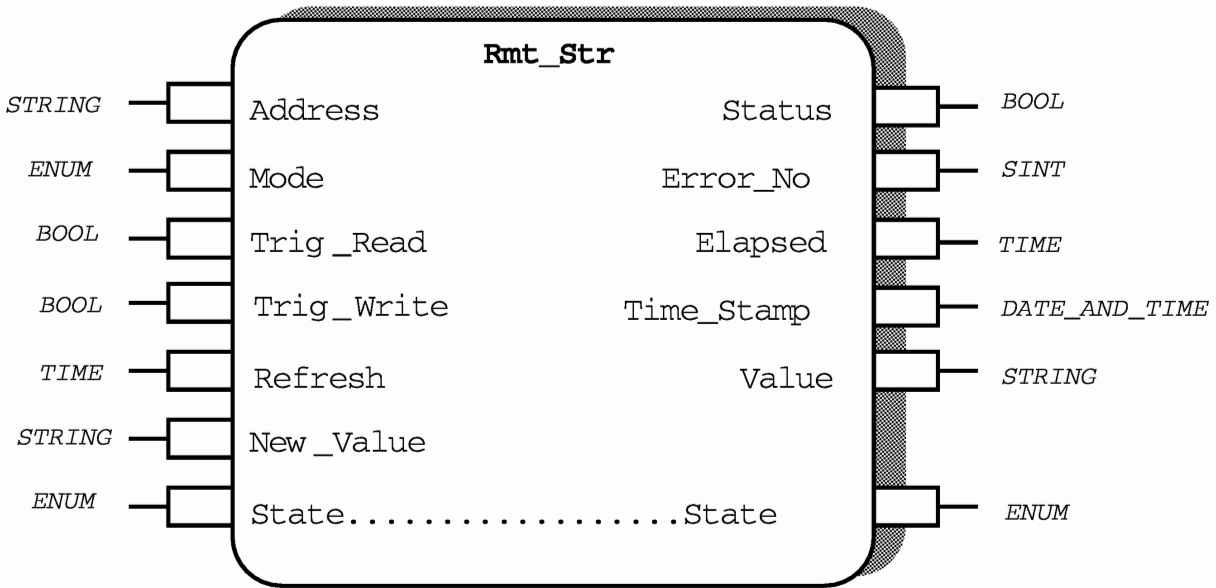


Figure 12-4 Rmt_Str Variable Function Block

Functional Description

The Rmt_Str Variable function block allows the reading and / or writing of the value of a string variable via a communications port. The function block is not communications protocol specific, but can only be used with ports which support protocols which function in the Master mode.

For a full description of the use of Remote_Vars function blocks, see the PC3000 Communications Overview in chapter 3.

Function Block Attributes

- Type:3A 80
- Class:.....REMOTE_VARS
- Default_Task:Task_2
- Short List:Value, New_Value, Mode, State
- Memory Requirements:474 Bytes
- Execution Time:316 μ Secs

Parameter Descriptions

Address (A)

This parameter is a string which is used to associate the function block with a parameter in the remote device to which the PC3000 is connected. The first character of Address defines the PC3000 module on which the communications port is situated. The second character defines the Port on the module. The rest of the address is protocol specific and is used to identify the instrument and parameter that the function block is communicating with.

Mode (M)

The Mode specifies mode of operation of the function block. It can have four possible values:

- Demand (0): In Demand mode, a single remote read or write can be triggered by changing the State parameter from OK(0) to Read(4) or Write(3).
- R_Cont (1): In R_Cont mode, the remote variable will be continuously polled at regular intervals defined by Refresh.
- W_Cont (2): In W_Cont mode, the remote variable will be continuously written to at regular intervals defined by Refresh.
- Change (3): In Change mode, the remote variable is written to when the value of the New_Value input does not match that of Value. The minimum period between changes being written to the remote device is defined by Refresh.

Trig_Read (TR)

When the function block is operating in Demand mode, changing Trig_Read from Off (0) to On (1) will cause a remote variable read to take place. This input is provided to permit a Read to be triggered by soft-wiring.

Trig_Write (TW)

When the function block is operating in Demand mode, changing Trig_Write from Off (0) to On (1) will cause a remote variable write to take place. This input is provided to permit a Write to be triggered by soft-wiring.

Refresh (R)

Refresh defines the period between update of the variable when the function block is in R_Cont or W_Cont modes.

New_Value (NV)

New_Value is the value to be written to the remote variable, when the function block is in a write mode.

State (S)

The parameter State indicates the current state of the function block. It can have one of five values:

- Ok (0): No communications actions are currently taking place.
- Pending (1): A transaction has started and has not yet been completed.
- Error (2): A communications transaction has failed. The parameter Error_No provides a diagnostic indicator to the type of error.
- Write (3): Setting State to Write (3) triggers a single write of New_Value to the remote device.
- Read (4): Setting State to Read (4) triggers a single read of the remote parameter.

When State is set to Write(3) or Read(4) it will automatically revert to OK(0) if the transmission was successful or remain in the Error(2) state.

Status (ST)

If Status is set to Go (1), the previous communications request has been completed successfully. If Status is set to NOGO (0), an error has occurred.

Error_No (ERR)

Error_No is a diagnostic parameter which provides an indication to the type of communications error which has occurred. Reference should be made to appropriate Master communications driver in Chapter 3 for error codes.

Elapsed (E)

Elapsed indicates the amount of time that has passed since the last read or write transaction. Elapsed is only valid when the function block is operating in R_Cont or W_Cont modes.

Time_Stamp (TS)

Time_Stamp defines the time and date at which the last read or write transaction took place successfully. It is set coincident with the State parameter changing to Ok (0).

Value (VAL)

The Value parameter contains the last value which was successfully written to or read from the remote variable.

Parameter Attributes

Name	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Address	STRING		Oper	Config		
Elapsed	TIME	0	Oper			
Error_No	SINT	0	Oper		High Limit Low Limit	255 0
Mode	ENUM	Demand (0)	Oper	Super	Senses	Demand (0) R_Cont (1) W_Cont (2) Change (3)
New_Value	STRING		Oper	Oper		
Refresh	TIME	10s	Oper	Super	High Limit Low Limit	19d_59m_59s 100ms
State	ENUM	Ok (0)	Oper	Oper	Senses	Ok (0) Pending (1) Error (2) Write (3) Read (4)
Status	BOOL	Go (1)	Oper		Senses	NOGO (0) Go (1)
Time_Stamp	DATE AND TIME	0	Oper			
Trig_Read	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Trig_Write	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Value	STRING		Oper			

Table 12-4 Rmt_Str Variable Parameter Attributes

RMT_SW FUNCTION BLOCK:

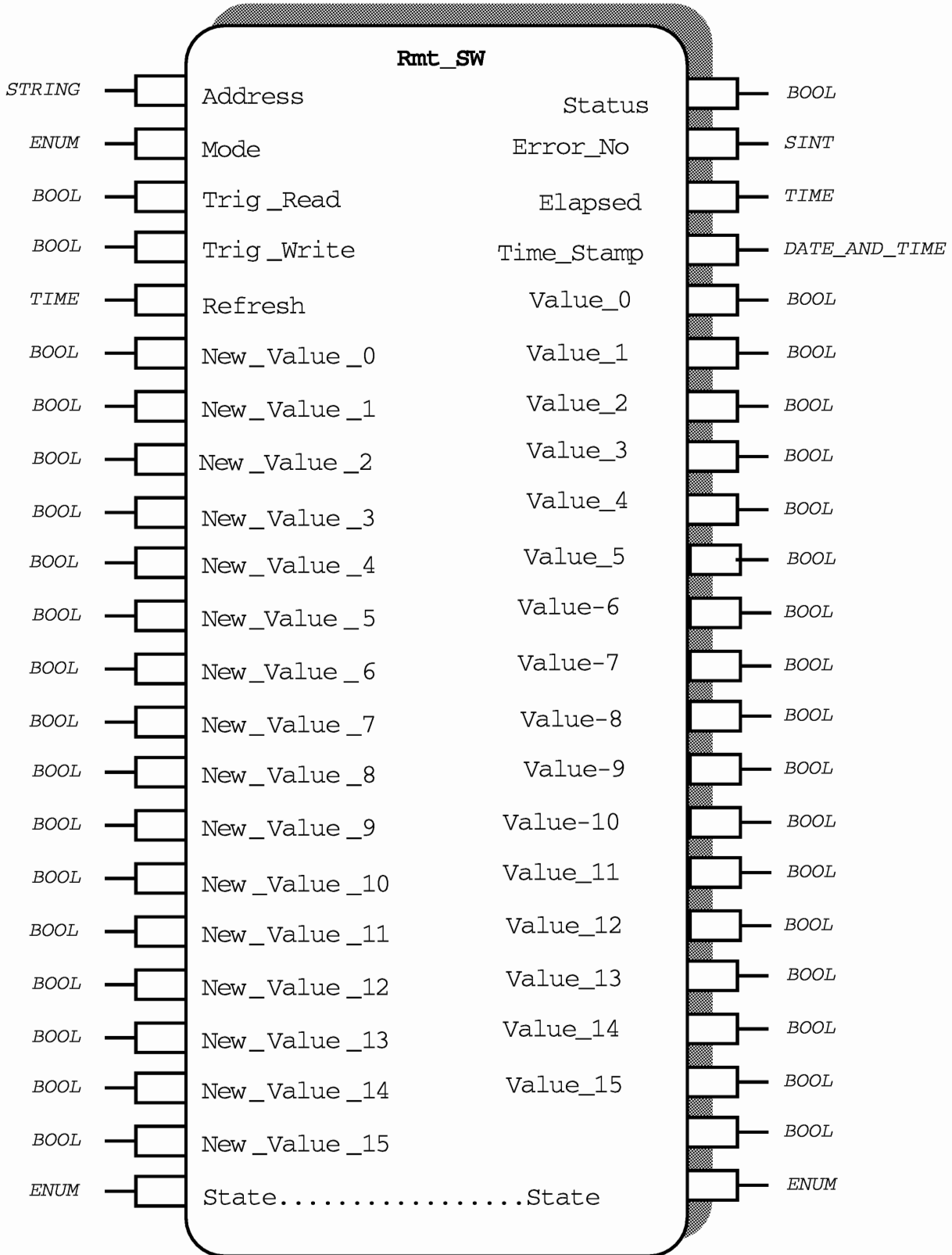


Figure 12-5 Rmt_SW Variable Function Block

Functional Description

The Rmt_SW Variablefunction block allows the reading and / or writing of the value of a 16-bit variable via a communications port. The 16 bits are individually addressable as boolean variables. The function block is not communications protocol specific, but can only be used with ports which support protocols which function in the Master mode.

For a full description of the use of Rmt_Vars function blocks, see the PC3000 Communications Overview in chapter 3.

Function Block Attributes

Type:..... 3A FO
 Class:REMOTE_VARS
 Default_Task: Task_2
 Short List: Mode, State
 Memory Requirements: 122 Bytes
 Execution Time; 132 μ Secs

Parameter Descriptions

Address (A)

This parameter is a string which is used to associate the function block with a parameter in the remote device to which the PC3000 is connected. The first character of Address defines the PC3000 module on which the communications port is situated. The second character defines the Port on the module. The rest of the address is protocol specific and is used to identify the instrument and parameter that the function block is communicating with.

Mode (M)

The Mode specifies mode of operation of the function block. It can have four possible values:

Demand (0): In Demand mode, a single remote read or write can be triggered by changing the State parameter from OK(0) to Read(4) or Write(3).

- R_Cont (1): In R_Cont mode, the remote variable will be continuously polled at regular intervals defined by Refresh.
- W_Cont (2): In W_Cont mode, the remote variable will be continuously written to at regular intervals defined by Refresh.
- Change (3): In Change mode, the remote variable is written to when the value of the New_Value input does not match that of Value. The minimum period between changes being written to the remote device is defined by Refresh.

Trig_Read (TR)

When the function block is operating in Demand mode, changing Trig_Read from Off (0) to On (1) will cause a remote variable read to take place. This input is provided to permit a Read to be triggered by soft-wiring.

Trig_Write (TW)

When the function block is operating in Demand mode, changing Trig_Write from Off (0) to On (1) will cause a remote variable write to take place. This input is provided to permit a Write to be triggered by soft-wiring.

Refresh (R)

Refresh defines the period between update of the variable when the function block is in R_Cont or W_Cont modes.

New_Value_0 to New_Value_15 (N0 to N15)

New_Value_0 to New_Value_15 are the values to be written to the remote variable bits 0 to 15 respectively, when the function block is in a write mode.

State (S)

The parameter State indicates the current state of the function block. It can have one of five values:

- Ok (0): No communications actions are currently taking place.
- Pending (1): A transaction has started and has not yet been completed.
- Error (2): A communications transaction has failed. The parameter Error_No provides a diagnostic indicator to the type of error.
- Write (3): Setting State to Write (3) triggers a single write of New_Value to the remote device.
- Read (4): Setting State to Read (4) triggers a single read of the remote variable.

When State is set to Write(3) or Read(4) it will automatically revert to OK(0) if the transmission was successful or remain in the Error(2) state.

Status (ST)

If Status is set to Go (1), the previous communications request has been completed successfully. If Status is set to NOGO (0), an error has occurred.

Error_No (ERR)

Error_No is a diagnostic parameter which provides an indication to the type of communications error which has occurred. Reference should be made to appropriate Master communications driver in Chapter 3 for error codes.

Elapsed (E)

Elapsed indicates the amount of time that has passed since the last read or write transaction. Elapsed is only valid when the function block is operating in R_Cont or W_Cont modes.

Time_Stamp (TS)

Time_Stamp defines the time and date at which the last read or write transaction took place successfully. It is set coincident with the State parameter changing to Ok (0).

Value_0 to Value_15 (V0 to V15)

The Value_0 to Value_15 parameters contain the last values which were successfully written to or read from the remote variable bits 0 to 15 respectively.

Parameter Attributes

Name	Type	Cold Start	Read Access	Write Access	Type Specific Information	
Address	STRING		Oper	Config		
Elapsed	TIME	0	Oper			
Error_No	SINT	0	Oper		High Limit Low Limit	255 0
Mode	ENUM	Demand (0)	Oper	Super	Senses	Demand (0) R_Cont (1) W_Cont (2) Change (3)
New_Value_0 to New_Value_15	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Refresh	TIME	10s	Oper	Super	High Limit Low Limit	19d_59m_59s 100ms
State	ENUM	Ok (0)	Oper	Oper	Senses	Ok (0) Pending (1) Error (2) Write (3) Read (4)
Status	BOOL	Go (1)	Oper		Senses	NOGO (0) Go (1)
Time_Stamp	DATE AND TIME	0	Oper			
Trig_Read	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Trig_Write	BOOL	Off (0)	Oper	Oper	Senses	Off (0) On (1)
Value_0 to Value_15	BOOL	Off (0)	Oper		Senses	Off (0) On (1)

Table 12-5 Rmt_SW Variable Parameter Attributes

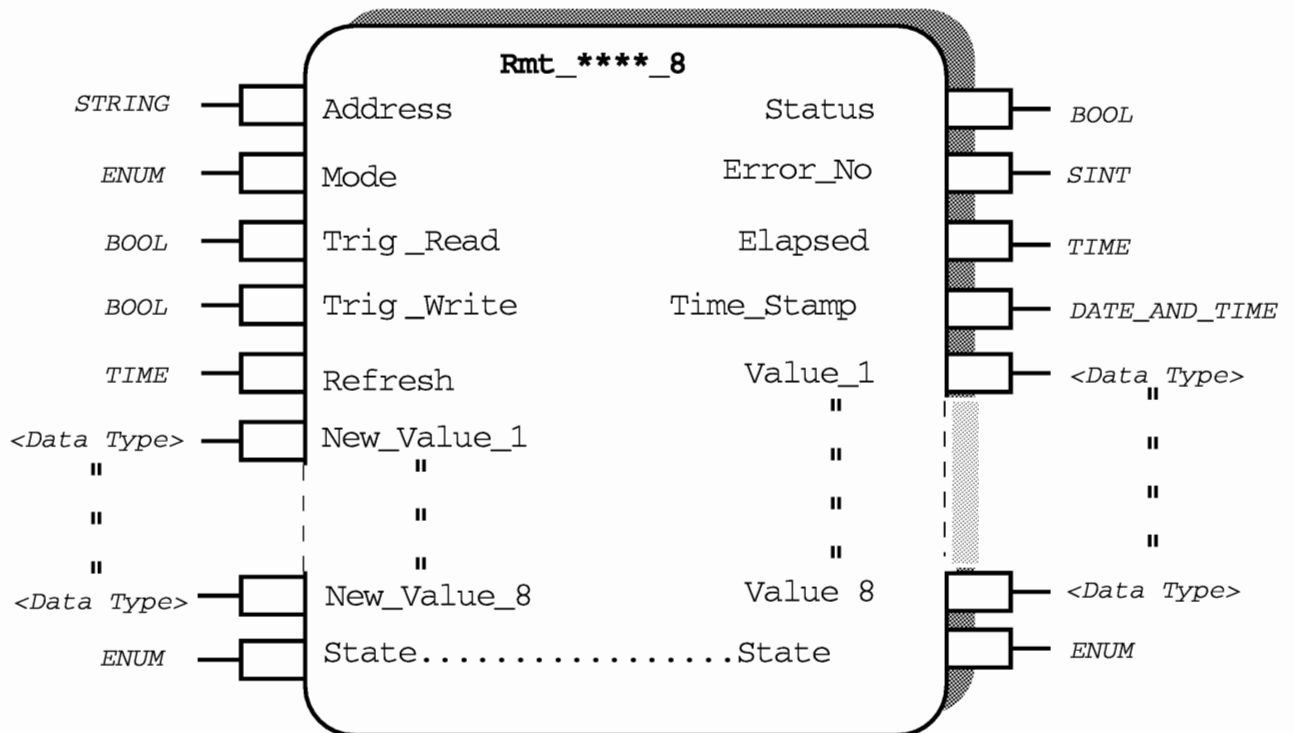
RMT_**_8 FUNCTION BLOCK**

Figure 12-6 Rmt_****_8 Function Block

Functional Description

All blocks in this class are described as Rmt_****_8 where **** is BOOL, REAL, DINT or STR.

Usage and facilities provided are identical to the Rmt_*** function blocks described earlier in this chapter. The primary difference is the provision of multiple New_Value inputs (New_Value_1 through New_Value_8 [Mnemonics N1 - N8]) and multiple Value outputs (Value_1 through Value_8 [Mnemonics V1 - V8]).

These function blocks provide an efficient method for accessing multiple parameters using a single read or write transaction. The blocks can access 8 contiguous locations. The data to be read or written is accessed using a single composite parameter. The order of data within the composite parameter is the same as the displayed order i.e. Value_1 to Value_8.

The blocks allow more efficient use of memory, since only one address is defined and there is only one set of control and status parameters for 8 data points.

These blocks may be used as an alternative to packing and unpacking data into strings using the COMPACT functions provided in the Structured Text language.

Note: This type of block is only suitable for use with the EI_Bisync_M function block at the present time. Other protocols may be supported in the future.

Example of addressing:

For Eurotherm Bisync,
Address = 0C 7 0 0 1

would result in the 8 values stored at addresses 001 through 008 being read/written in a single transaction.

RMT_****_64 FUNCTION BLOCK

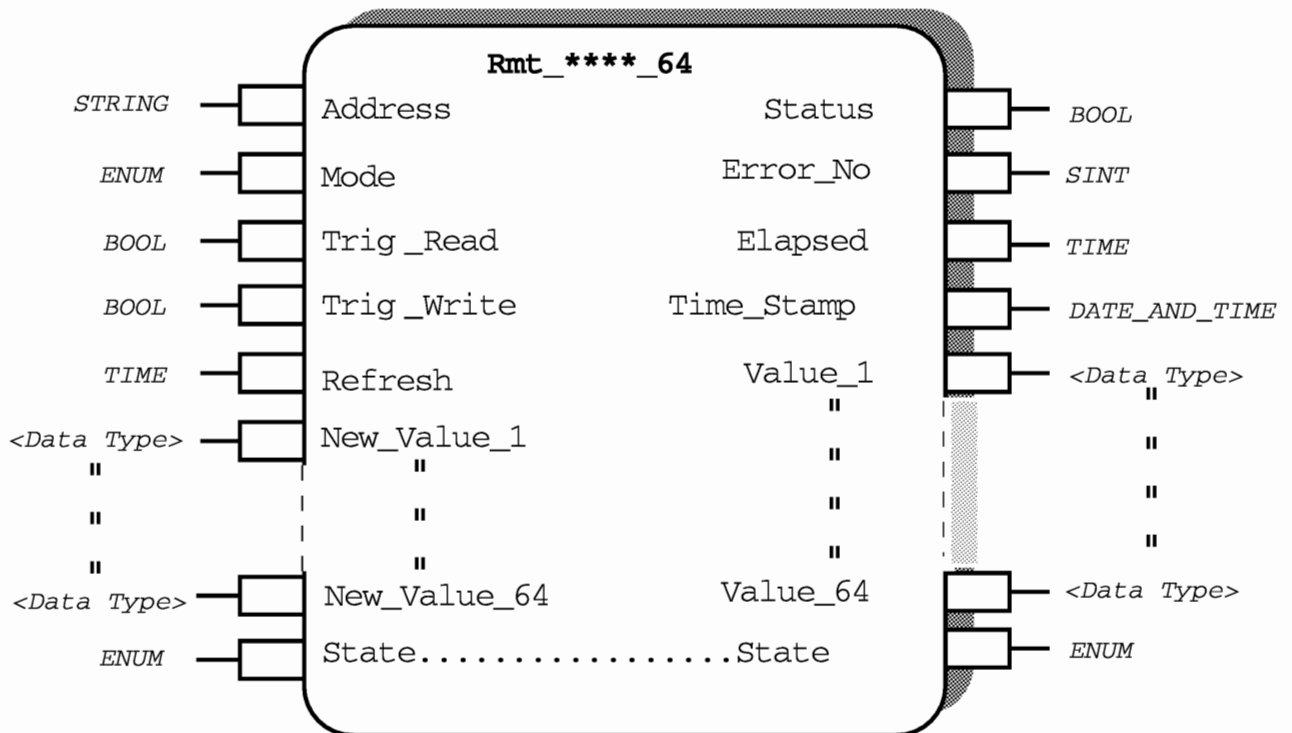


Figure 12-7 Rmt_****_64 Function Block

Functional Description

All blocks in this class are described as Rmt_****_64 where **** is BOOL, REAL, DINT or STR.

Usage and facilities provided are identical to the Rmt_*** function blocks described earlier in this chapter. The primary difference is the provision of multiple New_Value inputs (New_Value_1 through New_Value_64 [Mnemonics N1 - N64]) and multiple Value outputs (Value_1 through Value_64 [Mnemonics V1 - V64]).

These function blocks provide an efficient method for accessing multiple parameters using a single read or write transaction. The blocks can access 64 contiguous locations. The data to be read or written is accessed using a single composite parameter. The order of data within the composite parameter is the same as the displayed order i.e. Value_1 to Value_64.

The blocks allow more efficient use of memory, since only one address is defined and there is only one set of control and status parameters for 64 data points.

These blocks may be used as an alternative to packing and unpacking data into strings using the COMPACT functions provided in the Structured Text language.

Note: This type of block is only suitable for use with the EI_Bisync_M function block at the present time. Other protocols may be supported in the future.

Example of addressing:

For Eurotherm Bisync,
Address = 0C 7 0 0 1

would result in the 64 values stored at addresses 001 through 064 being read/written in a single transaction.