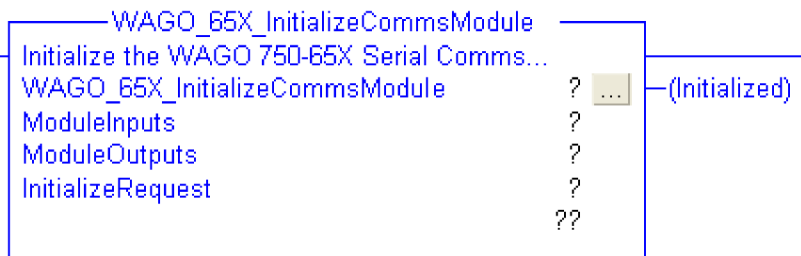


WAGO_65X_InitializeCommsModule v1.2
 WAGO

Initialize the WAGO 750-65X Serial Comms Module, halting transmit/receive functions and clearing all buffers.

Available Languages

Relay Ladder



Function Block



Structured Text

WAGO_65X_InitializeCommsModule(ModuleInputs, ModuleOutputs, InitializeRequest);

Parameters

| Required | Name | Data Type | Usage | Description |
|----------|--------------------------------|--------------------------------|--------|---|
| X | WAGO_65X_InitializeCommsModule | WAGO_65X_InitializeCommsModule | InOut | Initialize the WAGO 750-65X Serial Comms Module, halting transmit/receive functions and clearing all buffers. |
| | EnableIn | BOOL | Input | |
| | EnableOut | BOOL | Output | |
| X | ModuleInputs | SINT | InOut | |
| X | ModuleOutputs | SINT | InOut | |
| X | InitializeRequest | BOOL | Input | Set Initialize Request TRUE to begin module initialization. |
| | Initialized | BOOL | Output | Indicates a successful initialization when set TRUE. |

Extended Description

The WAGO_65X_InitializeCommsModule instruction initializes the WAGO Serial Communications Module (750-650 or 750-653), halting transmit/receive functions of the module, and clearing all buffers.

This instruction will only work with a 750-650 or 750-653 module configured for 3-bytes of data transfer. This is the default configuration of the 750-65X module.

As many as 64 750-65X serial communications modules can reside in one WAGO I/O node, and can be controlled by separate instances of this instruction.

The 'ModuleInputs' and 'ModuleOutputs' arrays must be mapped to the 4-byte (SINT) input and output process images of the respective WAGO 750-65X Serial Communications Module.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's input process image (see 'Controller Tags') to a SINT[4] array for passing data to the 'ModuleInputs' InOut array used by each serial module Add-On Instruction. Perform this operation before calling the 750-65X Add-On Instructions.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's output process image (see 'Controller Tags') to a SINT[4] array for passing data from the 'ModuleOutputs' InOut array used by each serial module Add-On Instruction. Perform this operation after calling the 750-65X Add-On Instructions.

Consult the respective product manual for further information about the 750-65X Serial Communications Module.

Execution

| Condition | Description |
|-----------|-------------|
|-----------|-------------|

| | |
|------------------|--|
| EnableIn is true | |
|------------------|--|

Revision v1.2 Notes

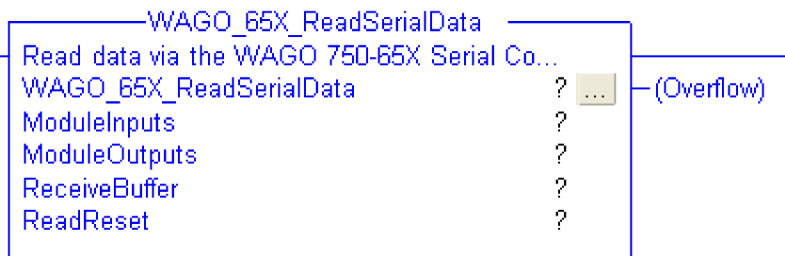
Changed name of instruction and parameters for consistency with other WAGO Add-On Instructions. No change in functionality.

WAGO_65X_ReadSerialData v1.2
 WAGO

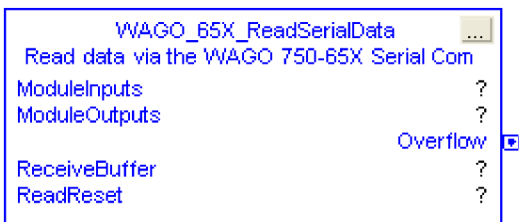
Read data via the WAGO 750-65X Serial Comms Module.

Available Languages

Relay Ladder



Function Block



Structured Text

WAGO_65X_ReadSerialData(ModuleInputs, ModuleOutputs, ReceiveBuffer, ReadReset);

Parameters

| Required | Name | Data Type | Usage | Description |
|----------|-------------------------|-------------------------|--------|--|
| X | WAGO_65X_ReadSerialData | WAGO_65X_ReadSerialData | InOut | Read data via the WAGO 750-65X Serial Comms Module. |
| | EnableIn | BOOL | Input | |
| | EnableOut | BOOL | Output | |
| X | ModuleInputs | SINT | InOut | |
| X | ModuleOutputs | SINT | InOut | |
| | Overflow | BOOL | Output | ReceiveBuffer has reached max limit of 80 chars. Reading is halted until ReadReset is performed. |
| X | ReceiveBuffer | STRING | InOut | String buffer where incoming serial data will be stored. |
| X | ReadReset | BOOL | InOut | Set ReadReset TRUE to initialize the ReceiveBuffer index to 0. |

Extended Description

The WAGO_65X_ReadSerialData instruction processes data received by a WAGO Serial Communications Module (750-650 or 750-653). □

This instruction will only work with a 750-650 or 750-653 module configured for 3-bytes of data transfer. This is the default configuration of the 750-65X module.

As many as 64 750-65X serial communications modules can reside in one WAGO I/O node, and can be controlled by separate instances of this instruction.

The 'ModuleInputs' and 'ModuleOutputs' arrays must be mapped to the 4-byte (SINT) input and output process images of the respective WAGO 750-65X Serial Communications Module.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's input process image (see 'Controller Tags') to a SINT[4] array for passing data to the 'ModuleInputs' InOut array used by each serial module Add-On Instruction. Perform this operation before calling the 750-65X Add-On Instructions.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's output process image (see 'Controller Tags') to a SINT[4] array for passing data from the 'ModuleOutputs' InOut array used by each serial module Add-On Instruction. Perform this operation after calling the 750-65X Add-On Instructions.

Consult the respective product manual for further information about the 750-65X Serial Communications Module.

Execution

| Condition | Description |
|-----------|-------------|
|-----------|-------------|

| | |
|------------------|--|
| EnableIn is true | |
|------------------|--|

Revision v1.2 Notes

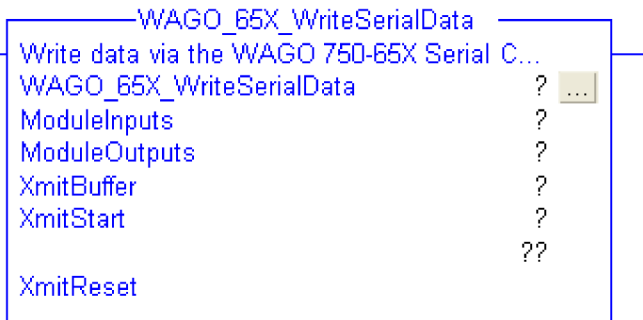
Changed name of instruction and parameters for consistency with other WAGO Add-On Instructions. No change in functionality.

WAGO_65X_WriteSerialData v1.1
 WAGO

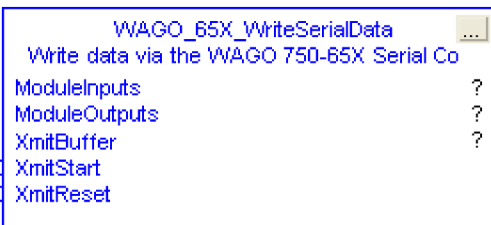
Write data via the WAGO 750-65X Serial Comms Module.

Available Languages

Relay Ladder



Function Block



Structured Text

WAGO_65X_WriteSerialData(ModuleInputs, ModuleOutputs, XmitBuffer, XmitStart);

Parameters

| Required | Name | Data Type | Usage | Description |
|----------|--------------------------|--------------------------|--------|---|
| X | WAGO_65X_WriteSerialData | WAGO_65X_WriteSerialData | InOut | Write data via the WAGO 750-65X Serial Comms Module. |
| | EnableIn | BOOL | Input | |
| | EnableOut | BOOL | Output | |
| X | ModuleInputs | SINT | InOut | |
| X | ModuleOutputs | SINT | InOut | |
| X | XmitBuffer | STRING | InOut | String of data to transmit serially. |
| X | XmitStart | BOOL | Input | Set XmitStart TRUE to begin character transmission. |
| | XmitReset | BOOL | Input | Set XmitReset TRUE to reset the flags and indexes within the WriteSerialData routine. Reset to re-enable comms. |

Extended Description

The WAGO_65X_WriteSerialData instruction transmits a string of data via a WAGO Serial Communications Module (750-650 or 750-653). □

This instruction will only work with a 750-650 or 750-653 module configured for 3-bytes of data transfer, with the 'Continuous Send' setting turned off. This is the default configuration of the 750-65X module.

As many as 64 750-65X serial communications modules can reside in one WAGO I/O node, and can be controlled by separate instances of this instruction.

The 'ModuleInputs' and 'ModuleOutputs' arrays must be mapped to the 4-byte (SINT) input and output process images of the respective WAGO 750-65X Serial Communications Module.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's input process image (see 'Controller Tags') to a SINT[4] array for passing data to the 'ModuleInputs' InOut array used by each serial module Add-On Instruction. Perform this operation before calling the 750-65X Add-On Instructions.

Use the Copy (COP) instruction to copy the 750-65X (750-650 or 750-653) serial module's output process image (see 'Controller Tags') to a SINT[4] array for passing data from the 'ModuleOutputs' InOut array used by each serial module Add-On Instruction. Perform this operation after calling the 750-65X Add-On Instructions.

Consult the respective product manual for further information about the 750-65X Serial Communications Module.

Execution

| Condition | Description |
|------------------|-------------|
| EnableIn is true | |

Revision v1.1 Notes

Changed name of instruction and parameters for consistency with other WAGO Add-On Instructions. No change in functionality.