

Object Types

Introduction

The language object types for the M221 Logic Controller are described in the following table:

Object Type	Object	Object Function	Description
Memory objects	%M	Memory bits	Stores memory bit.
	%MW	Memory words	Stores 16-bit memory word.
	%MD	Memory double words	Stores 32-bit memory word.
	%MF	Memory floating point	Stores memory floating point in a mathematical argument which has a decimal in its expression.
	%KW	Constant words	Stores 16-bit constant word.
	%KD	Constant double words	Stores 32-bit constant word.
	%KF	Constant floating points	Stores constant floating point in a mathematical argument which has a decimal in its expression.
System objects	%S	System bits, page 167	Stores system bit.
	%SW	System words, page 174	Stores system word.
	%IWS	Input channel status word, page 189	Contains diagnostic information concerning analog input channels.
	%QWS	Output channel status word, page 191	Contains diagnostic information concerning analog output channels.
I/O objects	%I	Input bits, page 153	Stores value of the digital input.
	%Q	Output bits, page 154	Stores value of the digital output.
	%IW	Input words, page 154	Stores value of the analog input.
	%QW	Output words, page 156	Stores value of the analog output.
	%FC	Fast counters	Execute fast counts of pulses from sensors, switches, and so on.
	%HSC	High speed counters	Execute fast counts of pulses from sensors, switches, and so on, that are connected to the fast inputs.
	%PLS	Pulse	Generates a square wave pulse signal on dedicated output channels.
	%PWM	Pulse width modulation	Generates a modulated wave signal on dedicated output channels with a variable duty cycle.
	%PTO	Pulse train output	Generates a pulse train output to control a linear single-axis stepper or servo drive in open loop mode.
	%FREQGEN	Frequency Generator	Generates a square wave signal on a dedicated output channel with programmable frequency and duty cycle of 50%.

Object Type	Object	Object Function	Description
Network objects	%QWE	Input assembly (EtherNet/IP), page 157	The values of EtherNet/IP Input assembly frames sent by the logic controller. NOTE: For more details about directionality, refer to Configuring EtherNet/IP, page 110.
	%IWE	Output assembly (EtherNet/IP), page 158	The values of EtherNet/IP Output assembly frames received by the logic controller. NOTE: For more details about directionality, refer to Configuring EtherNet/IP, page 110.
	%QWM	Input registers (Modbus TCP), page 159	The values of Modbus mapping table Input registers sent by the logic controller.
	%IWM	Output registers (Modbus TCP), page 160	The values of Modbus mapping table Output registers received by the logic controller.
	%IN	Digital inputs (IOScanner), page 161	The values of Modbus Serial or TCP IOScanner digital input bits.
	%QN	Digital outputs (IOScanner), page 162	The values of Modbus Serial or TCP IOScanner digital output bits.
	%IWN	Input registers (IOScanner), page 163	The values of Modbus Serial or TCP IOScanner digital input words.
	%QWN	Output registers (IOScanner), page 164	The values of Modbus Serial or TCP IOScanner digital output words.
	%IWNS	IOScanner network diagnostic codes, page 166	The values of Modbus Serial or TCP IOScanner network diagnostic bits.
Software objects	%TM	Timers	Specifies a time before triggering an action.
	%C	Counters	Provides up and down counting of actions.
	%MSG	Messages	Stores the status message at the communication port.
	%R	LIFO/FIFO registers	Stores memory up to 16 words of 16 bits each in 2 different ways, queue, and stacks.
	%DR	Drums	Operates on a principle similar to an electromechanical drum controller which changes step according to external events.
	%SBR	Shift bit registers	Provides a left or right shift of binary data bits (0 or 1).
	%SC	Step counters	Provides a series of steps to which actions can be assigned.
	SCH	Schedule blocks	Controls actions at a predefined month, day, and time.
	%RTC	RTC	Allows reading or writing the value of the Real Time Clock (RTC) on the logic controller.
	PID	PID	Provides a generic control loop feedback in which output is proportional, integral, and derivative of the input.
	%X	Grafset steps	Bit objects associated with individual Grafset (SFC) steps. Object is set to 1 when the corresponding step is active, and set to 0 when the step is deactivated.
Object Type	Object	Object Function	Description
PTO objects	Refer to Pulse Train Output.		
Drive objects	Refer to Drive Objects.		
Communication objects	%READ_VAR	Read Var	The %READ_VAR function block is used to read data from a remote device on Modbus SL or Modbus TCP.
	%WRITE_VAR	Write Var	The %WRITE_VAR function block is used to write data to an external device using the Modbus SL or Modbus TCP protocol.
	%WRITE_READ_VAR	Write Read Var	The %WRITE_READ_VAR function block is used to read and write data stored in internal memory words to an external device using the Modbus SL or Modbus TCP protocol.
	%SEND_RECV_MSG	Send Receive Message	The %SEND_RECV_MSG function block is used to send or receive data on a serial line configured for the ASCII protocol.
	%SEND_RECV_SMS	Send Receive SMS	The %SEND_RECV_SMS function block is used to send or receive SMS messages through a GSM modem connected to a serial line.

Object Type	Object	Object Function	Description
User-defined function and user-defined function block objects	%RET0	Return value	The return value of a user-defined function.
	%PARAM	Parameter	Parameters of a user-defined function or user-defined function block. The parameters are different for each object type.
	%VAR	Local variable	Local variables of a user-defined function or user-defined function block. The local variables are different for each object type.

Memory objects and software objects are generic objects used in EcoStruxure Machine Expert - Basic, whereas system objects and I/O objects are controller-specific. All controller-specific objects are discussed in the [Programming, page 152](#) section.

For programming details of memory objects, software objects, and communication objects, refer to the EcoStruxure Machine Expert - Basic Generic Functions Library Guide.

For programming details of PID, Drive, and PTO objects, refer to the Advanced Functions Library Guide.

For more information on user-defined functions and user-defined function blocks, refer to EcoStruxure Machine Expert - Basic Operating Guide (see EcoStruxure Machine Expert - Basic, Operating Guide).

Addressing I/O Objects

Addressing Examples

This table presents addressing examples for various object types:

Object Type	Syntax	Example	Description
Memory objects			
Memory bits	%Mi	%M25	Internal memory bit 25.
Memory words	%MWi	%MW15	Internal memory word 15.
Memory double words	%MDi	%MD16	Internal memory double word 16.
Memory floating points	%MFi	%MF17	Internal memory floating point 17.
Constant words	%KWi	%KW26	Constant word 26.
Constant double words	%KDi	%KD27	Internal constant double word 27.
Constant floating points	%KFi	%KF28	Internal constant floating point 28.
System objects			
System bits	%Si	%S8	System bit 8.
System words	%SWi	%SW30	System word 30.
I/O objects			
Digital inputs	%Iy.z	%I0.5	Digital input 5 on the controller (embedded I/O).
Digital outputs	%Qy.z	%Q3.4	Digital output 4 on the expansion module at address 3 (expansion module I/O).
Analog inputs	%IWy.z	%IW0.1	Analog input 1 on the controller (embedded I/O).
Analog outputs	%QW0.mOn	%QW0.100	Analog output 0 on the cartridge 1.
Fast counters	%FCi	%FC2	Fast counter 2 on the controller.
High speed counters	%HSCi	%HSC1	High speed counter 1 on the controller.
Pulse	%PLSi	%PLS0	Pulse output 0 on the controller.
Pulse width modulation	%PWMi	%PWM1	Pulse width modulation output 1 on the controller.
Pulse train output	%PTOi	%PTO1	Pulse train output 1 on the controller.
Frequency generator	%FREQGENi	%FREQGEN1	Frequency generator 1 on the controller.
Network objects			
Input assembly (EtherNet/IP)	%QWEi	%QWE8	Input assembly instance 8.
Output assembly (EtherNet/IP)	%IWEi	%IWE6	Output assembly instance 6.
Input registers (Modbus TCP)	%QWMi	%QWM1	Input register instance 1.
Output registers (Modbus TCP)	%IWMi	%IWM0	Output register instance 0.
Digital inputs (IOScanner)	%INa.b.c	%IN300.2.1	Modbus TCP IOScanner slave device 0 on ETH1, channel 2, digital input 1.
Digital outputs (IOScanner)	%QNa.b.c	%QN101.1.0	Modbus Serial IOScanner slave device 1 on SL1, channel 1, digital output 0.
Input registers (IOScanner)	%IWNa.b.c	%IWN302.3.0	Modbus TCP IOScanner slave device 2 on ETH1, channel 3, input register 0.
Output registers (IOScanner)	%QWNa.b.c	%QWN205.0.4	Modbus Serial IOScanner slave device 5 on SL2, channel 0, output register 4.

Object Type	Syntax	Example	Description
IOScanner network diagnostic codes	%IWNSa	%IWNS302	Status of Modbus TCP IOScanner slave device 2 on ETH1.
	%IWNSa.b	%IWNS205.3	Status of channel 3 of Modbus Serial IOScanner slave device 5 on serial line SL2
Software objects			
Timers	%TMi	%TM5	Timer instance 5.
Counters	%Ci	%C2	Counter instance 2.
Message	%MSGi	%MSG1	Program compilation status message 1.
LIFO/FIFO registers	%Ri	%R3	FIFO/LIFO registers instance 3.
Drums	%DRi	%DR6	Drum register 6 on the controller.
Shift bit registers	%SBRi	%SBR5	Shift bit register 5 on the controller.
Step counters	%SCi	%SC5	Step counter 5 on the controller.
Schedule blocks	SCH i	SCH 3	Schedule block 3 on the controller.
RTC	RTCi	RTC 1	Real-time clock (RTC) instance 1.
PID	PIDi	PID 7	PID feedback object 7 on the controller.
Grafcet Steps	Xi	X1	Grafcet step 1.
PTO objects			
MC_Power_PTO (motion function block)	%MC_POWER_PTOi	%MC_POWER_PTO1	MC_POWER_PTO function block instance 1.
MC_Reset_PTO (administrative function block)	%MC_RESET_PTOi	%MC_RESET_PTO0	MC_RESET_PTO function block instance 0.
Communication objects			
Read Var	%READ_VARi	%READ_VAR2	READ_VAR function block instance 2.
Write Var	%WRITE_VARi	%WRITE_VAR4	WRITE_VAR function block instance 4.
Write Read Var	%WRITE_READ_VARi	%WRITE_READ_VAR0	WRITE_READ_VAR function block instance 0.
Send Receive Message	%SEND_RECV_MSGi	%SEND_RECV_MSG6	SEND_RECV_MSG function block instance 6.
Send Receive SMS	%SEND_RECV_SMSi	%SEND_RECV_SMS0	SEND_RECV_SMS function block instance 0.
User-defined function and user-defined function block objects			
Return value	%RETi	%RET0	Return value of a user-defined function.
Parameters	%PARAMi	%PARAM0	Parameter of a user-defined function.
Local variables	%VARi	%VAR0	Local variables of a user-defined function.
a: 100 + device number on SL1, 200 + device number on SL2, 300 + device number on ETH1. b: Channel number of the Modbus Serial IOScanner or Modbus TCP IOScanner device. c: Object instance identifier in the channel. i: Object instance identifier that indicates the instance of the object on the controller. m: Cartridge number on the controller. n: Channel number on the cartridge. y: Indicates the I/O type. It is 0 for the controller and 1, 2, and so on, for the expansion modules. z: Channel number on the controller or expansion module.			

Maximum Number of Objects

Maximum Number of Objects Description

This table provides information about the maximum number of objects supported by the M221 Logic Controller:

Objects	M221 Logic Controller References			
	Modular References		Compact References	
	TM221M16R• TM221ME16R•	TM221M16T• TM221ME16T• TM221M32TK TM221ME32TK	TM221C••R TM221CE••R	TM221C••T TM221CE••T TM221C••U TM221CE••U
Memory objects				
%M ⁽¹⁾	512 1024	512 1024	512 1024	512 1024
%MW	8000	8000	8000	8000
%MD	7999	7999	7999	7999
%MF				
%KW	512	512	512	512
%KD	511	511	511	511
%KF				
System objects				
%S	160	160	160	160
%SW	234	234	234	234
%IWS	1 created automatically for each analog input			
%QWS	1 created automatically for each analog output			
I/O objects				
%I	8	8 (for TM221M16T• and TM221ME16T•)	9 (for TM221C16• and TM221CE16•)	9 (for TM221C16• and TM221CE16•)
		16 (for TM221M32TK and TM221ME32TK)	14 (for TM221C24• and TM221CE24•)	14 (for TM221C24• and TM221CE24•)
			24 (for TM221C40• and TM221CE40•)	24 (for TM221C40• and TM221CE40•)
%Q	8	8 (for TM221M16T• and TM221ME16T•)	7 (for TM221C16• and TM221CE16•)	7 (for TM221C16• and TM221CE16•)
		16 (for TM221M32TK and TM221ME32TK)	10 (for TM221C24• and TM221CE24•)	10 (for TM221C24• and TM221CE24•)
			16 (for TM221C40• and TM221CE40•)	16 (for TM221C40• and TM221CE40•)
%IW	2	2	2	2

Objects	M221 Logic Controller References			
	Modular References		Compact References	
	TM221M16R• TM221ME16R•	TM221M16T• TM221ME16T• TM221M32TK TM221ME32TK	TM221C••R TM221CE••R	TM221C••T TM221CE••T TM221C••U TM221CE••U
%QW	0	0	NOTE: Analog outputs are not built in with the controller. Use cartridges TMC2AQ2V and/or TMC2AQ2C to add analog outputs to your controller configuration.	
			2 (if 1 cartridge is used) 4 (if 2 cartridges are used with TM221C40R or TM221CE40R)	2 (if 1 cartridge is used) 4 (if 2 cartridges are used with TM221C40T or TM221CE40T or TM221C••U or TM221CE••U)
%FC	4	4	4	4
%HSC	Up to 4	Up to 4	Up to 4	Up to 4
%PLS	0	2	0	2
%PWM				
%PTO				
%FREQGEN				
Network objects				
%QWE	20 (for TM221ME16R•)	20 (for TM221ME16T• and TM221ME32TK)	20 (for TM221CE16•)	20 (for TM221CE16•)
%IWE	20 (for TM221ME16R•)	20 (for TM221ME16T• and TM221ME32TK)	20 (for TM221CE16•)	20 (for TM221CE16•)
%QWM	20 (for TM221ME16R•)	20 (for TM221ME16T• and TM221ME32TK)	20 (for TM221CE16•)	20 (for TM221CE16•)
%IWM	20 (for TM221ME16R•)	20 (for TM221ME16T• and TM221ME32TK)	20 (for TM221CE16•)	20 (for TM221CE16•)
%IN	128	128	128	128
%QN	128	128	128	128
%IWN	128 ⁽²⁾	128 ⁽²⁾	128 ⁽²⁾	128 ⁽²⁾
%QWN	128 ⁽²⁾	128 ⁽²⁾	128 ⁽²⁾	128 ⁽²⁾
%IWNS	1 for each configured Modbus Serial IOScanner or Modbus TCP IOScanner device, plus 1 for each channel			
%QWNS	1 for each configured Modbus Serial IOScanner or Modbus TCP IOScanner device, plus 1 for each channel			
Software objects				
%TM	255	255	255	255
%C	255	255	255	255
%MSG	2	2	1 (for TM221C••R)	1 (for TM221C••T and TM221C••U)
			2 (for TM221CE••R)	2 (for TM221CE••T and TM221CE••U)

Objects	M221 Logic Controller References			
	Modular References		Compact References	
	TM221M16R• TM221ME16R•	TM221M16T• TM221ME16T• TM221M32TK TM221ME32TK	TM221C••R TM221CE••R	TM221C••T TM221CE••T TM221C••U TM221CE••U
%R	4	4	4	4
%DR	8	8	8	8
%SBR	8	8	8	8
%SC	8	8	8	8
%SCH	16	16	16	16
%RTC	2	2	2	2
PID	14	14	14	14
Drive objects				
%DRV	16	16	16	16
Communication objects				
%READ_VAR	32 (if functional level ≥ 10.1) or 16 (if functional level < 10.1).			
%WRITE_VAR	32 (if functional level ≥ 10.1) or 16 (if functional level < 10.1).			
%WRITE_READ_VAR	32 (if functional level ≥ 10.1) or 16 (if functional level < 10.1).			
%SEND_RECV_MSG	16	16	16	16
%SEND_RECV_SMS	1	1	1	1
User-defined function objects				
%RET0	1 per user-defined function			
%PARAM	5 per user-defined function			
%VAR	48 (including any existing %PARAM)			
User-defined function block objects				
%Q_	32 (if functional level ≥ 10.0) or 8 (if functional level < 10.0).			
%I_	32 (if functional level ≥ 10.0) or 8 (if functional level < 10.0).			
%PARAM	48 (including any existing %VAR)			
%VAR	48 (including any existing %PARAM)			
(1) The value 512 is for software version < 1.3.				
(2) If functional level < 6.0. If functional level ≥ 6.0, the maximum number of objects is 512.				

Maximum Number of PTO Objects Description

This table provides information about the maximum number of PTO objects supported by the M221 Logic Controller:

Categories/Objects	M221 Logic Controller References		
	TM221M16R• TM221ME16R• TM221C••R TM221CE••R	TM221M16T• TM221ME16T• TM221M32TK TM221ME32TK TM221C••T TM221CE••T TM221C16U TM221CE16U TM221C24U TM221CE24U	TM221C40U TM221CE40U
Motion/Single-axis			
%MC_POWER_PTO	0	86	
%MC_MOVEVEL_PTO			
%MC_MOVEREL_PTO			
%MC_MOVEABS_PTO			
%MC_HOME_PTO			
%MC_SETPOS_PTO			
%MC_STOP_PTO			
%MC_HALT_PTO			
Motion/Motion Task			
%MC_MotionTask_PTO	0	2	4
Administrative			
%MC_READACTVEL_PTO	0	40	
%MC_READACTPOS_PTO			
%MC_READSTS_PTO			
%MC_READMOTIONSTATE_PTO			
%MC_READAXISERROR_PTO			
%MC_RESET_PTO			
%MC_TOUCHPROBE_PTO			
%MC_ABORTTRIGGER_PTO			
%MC_READPAR_PTO			
%MC_WRITEPAR_PTO			