



eScada

v24.2.0  
Drivers

eScada.Drivers.Applicom

## eScada.Drivers.Applicom

( Direct-Link PC Network Interfaces )

### OS availability

Windows x86

### Atomic data type

Bit, Byte or 16 bit Word oriented protocol.

### Hardware and documentation reference

www.molex.com

### Parameters available in every section

Channel:	none	
Device:	Applicom version	Applicom driver version. < 3.0 or 3.0 and higher
	Driver mode	Wait mode
	Addressing type	Standard Applicom
	Channel number	Channel number, from Applicom configuration
	Equipment number	Equipment number, from Applicom configuration
	Reconnect timeout [ms]	Waiting time before a re-connection after COMM break-down
Group:	none	
Tag:	none	

### Remarks for devices

The following attributes can be expressed for each device.

Bytes order actions	None, Swap bytes order, Swap bytes order in DWords, Swap words order, Swap bytes order in DWords then words order
String actions	None, Swap bytes in words

### Implemented memory area for standard Applicom addressing

Internal memory bits	<b>Bx</b>	Read/Write	READPACKBITS, WRITEPACKBITS
Input bits	<b>Bix</b>	Read only	READPACKIBITS, WRITEPACKIBITS
Output bits	<b>BOx</b>	Read/Write	READPACKQBITS, WRITEPACKQBITS
Internal memory bytes	<b>Ox</b>	Read/Write	READBYTE, WRITEBYTE
Input bytes	<b>Oix</b>	Read only	READIBYTE, WRITEIBYTE
Output bytes	<b>OOx</b>	Read/Write	READQBYTE, WRITEQBYTE
Internal memory words	<b>Wx</b>	Read/Write	READWORD, WRITEWORD
Input words	<b>Wix</b>	Read only	READIWORD, WRITEIWORD
Output words	<b>WOx</b>	Read/Write	READQWORD, WRITEQWORD

**Remarks:** timers and counters are not implemented.

**x** = Standard Applicom address.

Expressed using base 10 numbers.

Please refer to your specific protocol document to understand how to address variables to your device.

See addressing examples section.

**Standard addressing**

See addressing examples section for more information.

Variable type	Type	Address type	Items
<b>Boolean</b> In case of word addressing, the number of items used declaring TAGs, must be a multiples of 16			
Single bit	Bit	B, BI, BO, W, WI, WO	(C)
<b>Byte</b> In case of word addressing, the number of items used declaring TAGs, must be a multiples of 2			
Unsigned 8 bit	UInt8	O, OI, OO, W, WI, WO	(C)
Signed 8 bit	Int8		
<b>16 bit</b>			
Unsigned integer 16 bit	UInt16	W, WI, WO	(C)
Signed integer 16 bit	Int16		
<b>32 bit</b>			
Unsigned integer 32 bit	UInt32	W, WI, WO	(C)
Signed integer 32 bit	Int32		
Single precision 32 bit - ( IEEE 754 )	Float		
<b>64 bit</b>			
Unsigned integer 64 bit	UInt64	W, WI, WO	(C)
Signed integer 64 bit	Int64		
Double precision 64 bit - ( IEEE 754 )	Double		
<b>Strings</b> In case of word addressing, the string length used declaring TAGs, must be a multiple of 2 String bytes can be interpreted as ASCII, UTF-7, UTF-8, UTF-16 or UTF-32 encoding			
Array of bytes	String	O, W	(A, C)
Array of bytes. (Siemens S7) Array of bytes. (AllenBradley style)	S7String ABString	O, W	(B, C)
(A) It depends on the string's length: e.g. if you want to read strings with a length of 10 chars each string, you can set a number of items of $74 / 10 = 7$ consecutive items.			
(B) It depends on the string's length: e.g. if you want to read strings with a length of 10 chars each string, you can set a number of items of $74 / (10+2) = 6$ consecutive items.			
(C) It depends on PLC model. The best way is to try with the maximum items you need. Please refer to your PLC model and protocol documentation			

**S7 strings format**

They have got two bytes at the beginning.

The first byte is for max allowed string length, the second one is for the real string length.

These types of strings can be declared with a length of 255 bytes max.

**AB Strings format**

They have got one word (16 bit) at the beginning, it contains the string length.

**Consecutive items**

The number of consecutive read/write items, depends on the protocol type and PLC model.

**Applicom addressing examples**

The following examples have been taken from their protocol manual delivered with Applicom product.

Please ask your supplier for a copy, based on the protocol you are using. They show how to calculate a standard Applicom address.

- Addressing mode for GE-SRTP protocol

PLC variable DDE-OPC item	Use by the applicom® library/DLL			
	Addressing		Exchange type	
	Decimal	Hexadecimal	Function name	
%Mz	z-1	z-1	Read bit	READPACKBIT
%Tz	131072 + z-1	0002 0000 + z-1	Write bit	WRITEPACKBIT
%Gz	262144 + z-1	0004 0000 + z-1		
%Sz	393216 + z-1	0006 0000 + z-1		
%SAz	524288 + z-1	0008 0000 + z-1		
%SBz	1048576 + z-1	0010 0000 + z-1		
%SCz	1179648 + z-1	0012 0000 + z-1		
%Iz	z-1	z-1	Read input bit	READPACKIBIT
%Qz	z-1	z-1	Read output bit Write output bit	READPACKQBIT WRITEPACKQBIT
%MBy	y-1	y-1	Read byte	READPACKBYTE, READBYTE
%TBy	131072 + y-1	0002 0000 + y-1	Write byte	WRITEPACKBYTE, WRITEBYTE
%GBy	262144 + y-1	0004 0000 + y-1		
%IBy	y-1	y-1	Read input byte	READPACKIBYTE, READIBYTE
%QBy	y-1	y-1	Read output byte Write output byte	READPACKQBYTE, READQBYTE WRITEPACKQBYTE, WRITEQBYTE
%Rx	x-1	x-1	Read word, BCD word	READWORD READWORDBCD
%Px *	1310720 + x-1	0014 0000 + x-1	Write word, BCD word	WRITEWORD WRITEWORDBCD
SPRGNAM%Lx *	1441792 + n* 16777216 + x-1	mm16 0000 + x-1		
%AIx	x-1	x-1	Read input word	READIWORD
%AQx	x-1	x-1	Read output word Write output word	READQWORD WRITEQWORD
%RDx	x-1	x-1	Read double word Write double word	READDWORD WRITEDWORD
%RFx	x-1	x-1	Read floating word Write floating word	READFWORD WRITEFWORD

**x**: Word number - **y** : Byte number - **z** : Bit number.

- Addressing mode for AllenBradley protocol DF1

Allen-Bradley PLC Variable	applicom® Addressing (Addr)	Exchange type (cyclic mode)	Corresponding applicom® Function (library/DLL access)
Bx/w or Bx:y/z	x * 16384 + w x * 16384 + y*16 + z	Read bits Write bits	READPACKBIT, READDIFBIT WRITEPACKBIT, WRITEDIFPACKBIT
Nx:y	x * 16384 + y	Read words Write words	READWORD, READDIFWORD WRITEWORD, WRITEDIFWORD
Nx:y	x * 16384 + y	Read double words * Write double words *	READDWORD, READDIFDWORD WRITEDWORD, WRITEDIFDWORD
Fx:y	x * 16384 + y	Read floating words Write floating words	READFWORD, READDIFFWORD WRITEFWORD, WRITEDIFFWORD
Tx:y/PRE	16384*x + y		READTIMER, WRITETIMER
Cx:y/PRE	16384*x + y		READCOUNTER, WRITECOUNTER

**w**: N° of bits (0 to 15999) - **x**: N° of file ( 0 to 999 )

**y**: N° of element in the file ( 0 to 999 ) - **z**: N° of sub-element ( 0 to 15 )

- Addressing mode for SIEMENS S7-Protocol

S7-300 S7-400 variable	<b>applicom</b> ® addressing (addr)	Type of exchange	<b>applicom</b> ® function (library/DLL access)
My.z DBx.DBXy.z	y*8+z x*524288+y*8+z	Read bits Write bits	READPACKBIT WRITEPACKBIT
Ey.z	y*8+z	Read input bits	READPACKIBIT
Ay.z	y*8+z	Read output bits Write output bits	READPACKQBIT WRITEPACKQBIT
MBy DBx.DBBy	y x*65536+y	Read bytes Write bytes	READPACKBYTE,READBYTE WRITEPACKBYTE,WRITEBYTE
EBy	y	Read input bytes	READPACKIBYTE,READIBYTE
ABy	y	Read output bytes Write output bytes	READPACKQBYTE,READQBYTE WRITEPACKQBYTE,WRITEQBYTE
MWy DBx.DBWy	y x*65536+y	Read words Write words	READWORD WRITEWORD
EWy	y	Read input words	READIWORD
AWy	y	Read output words Write output words	READQWORD WRITEQWORD
MDy DBx.DBBy	y x*65536+y	Read double words Write double words	READDWORD WRITEDWORD
MDyF DBx.DBByF	y x*65536+y	Read floating words Write floating words	READFWORD WRITEFWORD
T n	n	Read timers Write timers	READTIMER WRITETIMER
Z n	n	Read counters Write counters	READCOUNTER WRITECOUNTER

**x**: DB number - **y**: Byte number - **z**: Number of the bit in the byte (0 to 7)