



TIPOS DE DATOS

13

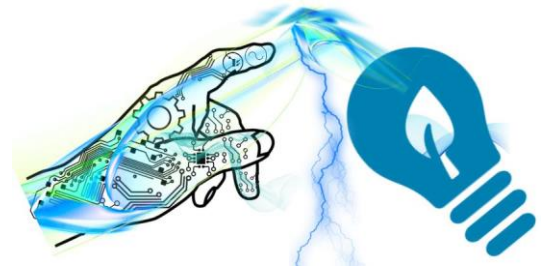
TIA V14

PLC SIEMENS S7 1200 (TIA PORTAL V14)

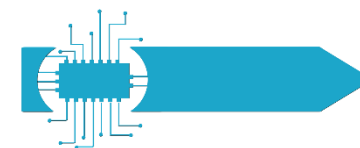
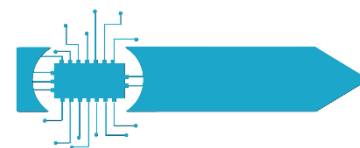
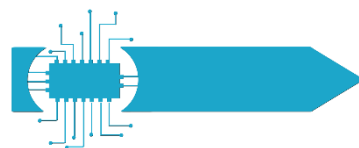
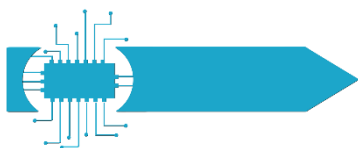
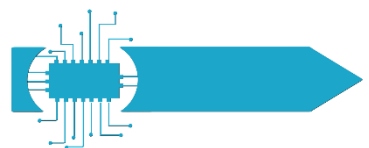


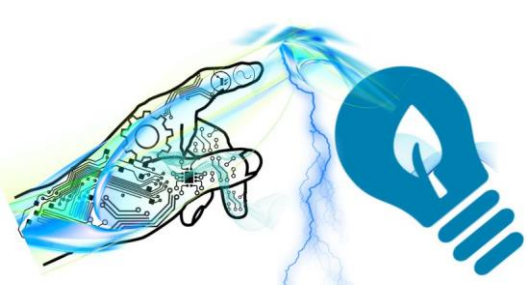


TIPOS DE DATOS

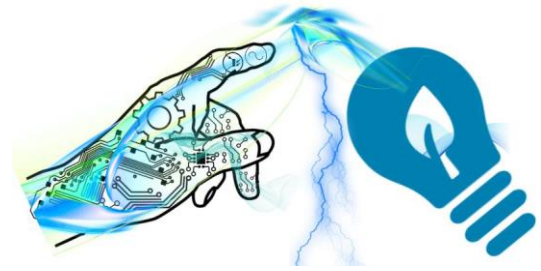


1. Números binarios
2. Enteros
3. Números en coma flotante
4. Temporizadores
5. Fecha y hora
6. Cadenas de caracteres
7. Tipos de datos PLC (UDT)

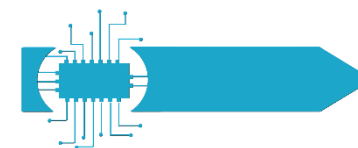
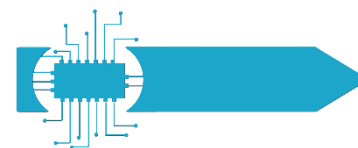
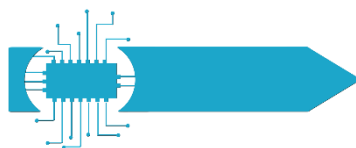
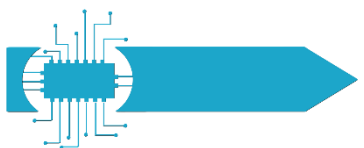


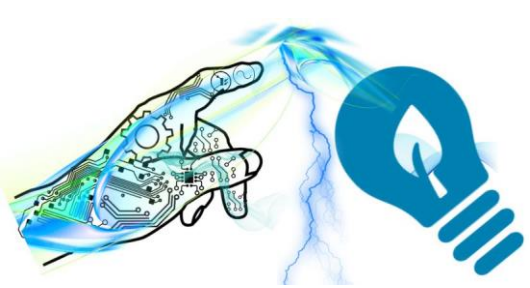


TIPOS DE DATOS

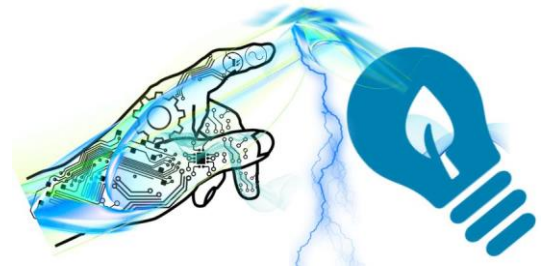


8. Estructura de datos STRUCT (estructuras anónimas)
9. ARRAY
10. Puntero
11. Tipos de parámetros
12. Tipos de datos de sistema
13. Tipos de datos de hardware
14. Conversión de los tipos de datos en el S7-1500 (S7-1500)
15. Conversión de los tipos de datos en el S7-1200 (S7-1200)
16. Conversión de tipos de datos en el S7-300/S7-400 (S7-300, S7-400)





01 NUMEROS BINARIOS



☞ **BOOL (BIT) booleano: == → dirección (M0.1....)**

💡 Un operando del tipo de datos BOOL representa un valor de bit y contiene uno de los siguientes valores:

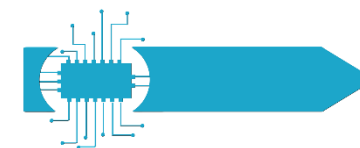
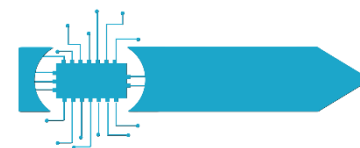
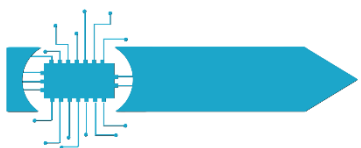
- True (1)
- False (0)

☞ **BYTE: ==== → dirección (MB10.....)**

💡 Un operando de tipo de datos BYTE es un secuencia de 8 bits

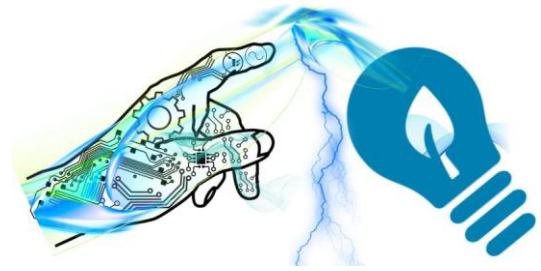
💡 Enteros (sistema decimal):

- Enteros con signo: de -128 a +127
- Enteros sin signo : de 0 a 255





01 NUMEROS BINARIOS



👉 **WORD : ==→ dirección (MW10....)**

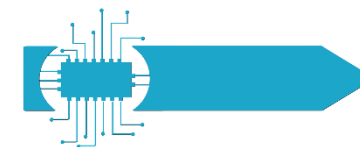
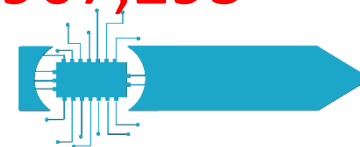
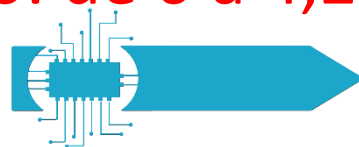
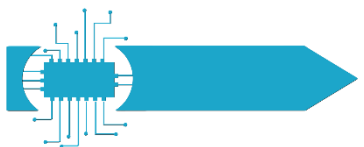
💡 Un operando de tipo de datos WORD es una secuencia de 16 bits.

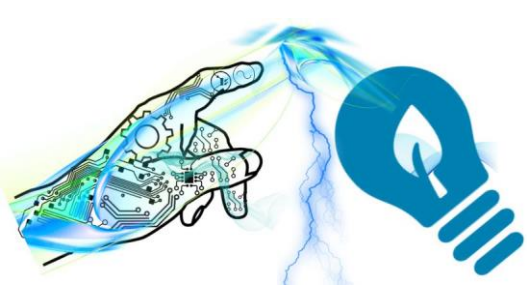
- Enteros con signo: de -32,768 a +32,767
- Enteros sin signo: de 0 a 65,535

👉 **DWORD : ==→ dirección (MD10....)**

💡 Un operando de tipo de datos DWORD es una secuencia de 32 bits.

- Enteros con signo: de -2,147,483,648 a +2,147,483,648
- Enteros sin signo: de 0 a 4,294,967,295





02 ENTEROS



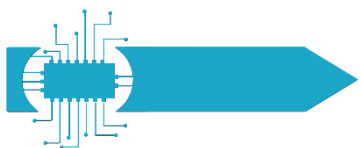
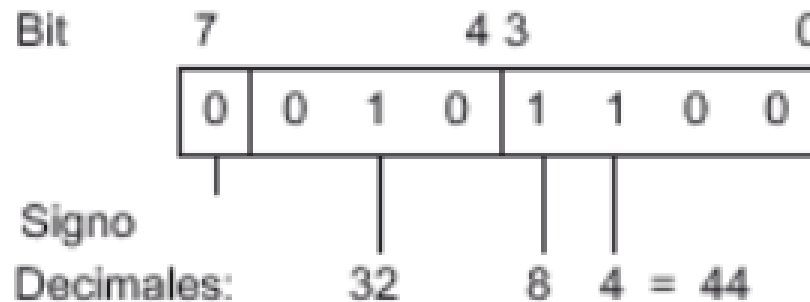
👉 SINT (enteros de 8 bits):

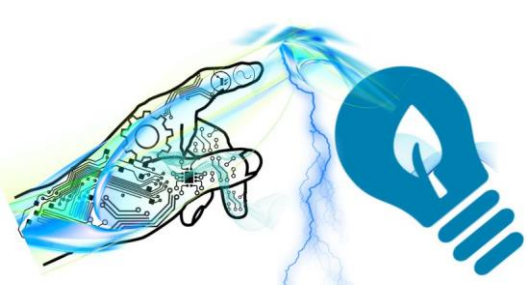
💡 Un operando del tipo de datos SINT (Short INT) tiene una longitud de 8 bits y consta de dos componentes: Un signo y un valor numérico en complemento a dos. Los estados lógicos de los bits 0 a 6 representan el valor del número. El estado lógico del bit 7 representa el signo. El signo puede adoptar el estado lógico "0" para positivo o "1" para negativo.

- Enteros con signo (sistema decimal) de -128 a +127

Ejemplo

La figura siguiente muestra el entero +44 como número binario:





02 ENTEROS



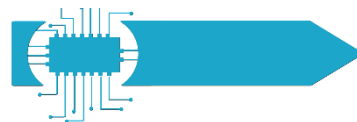
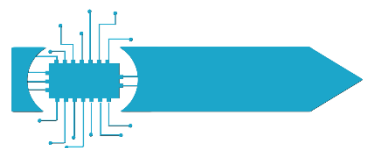
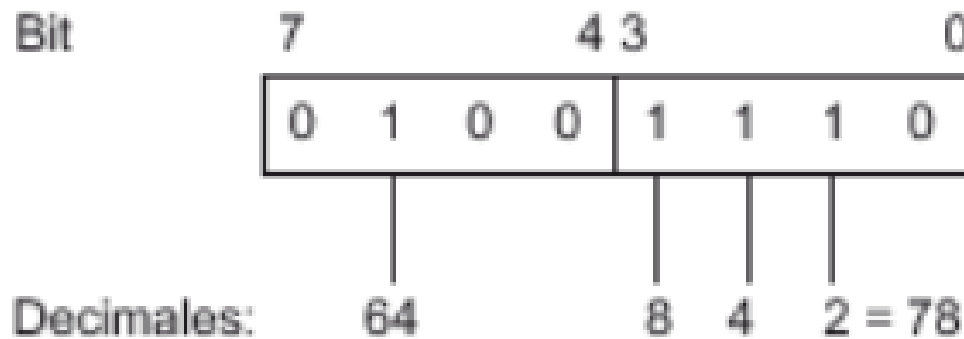
👉 USINT (enteros de 8 bits):

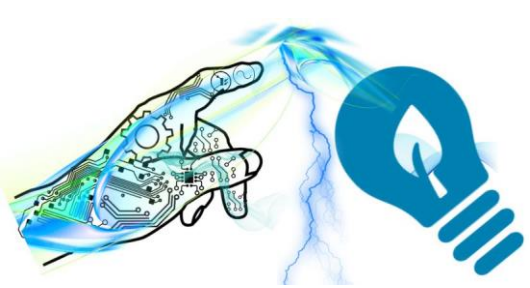
💡 Un operando del tipo de datos USINT (Unsigned Short INT) tiene una longitud de 8 bits y contiene valores numéricos sin signo.

- Entero sin signo: de 0 a 255

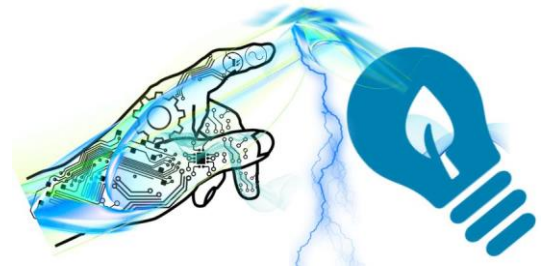
Ejemplo

La figura siguiente muestra el entero 78 como número binario:





02 ENTEROS



👉 INT (enteros de 16 bits) CON SIGNO → de -32,768 a +32,767

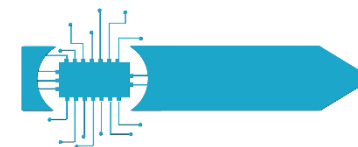
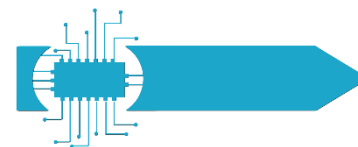
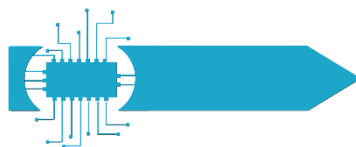
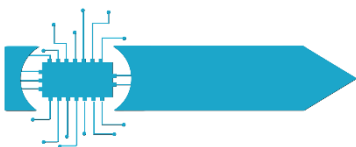
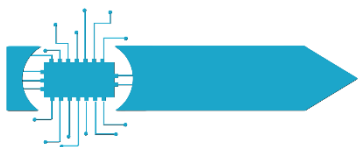
👉 UINT (enteros de 16 bits) SIN SIGNO → de 0 a 65,535

👉 DINT (enteros de 32 bits) CON SIGNO → de -2,147,483,648 a +2,147,483,648

👉 UDINT (enteros de 32 bits) SIN SIGNO → de 0 a 4,294,967,295

👉 LINT (enteros de 64 bits) (S7-1500)

👉 ULINT (enteros de 64 bits) (S7-1500)





03 NUMEROS EN COMA FLOTANTE

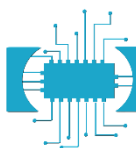
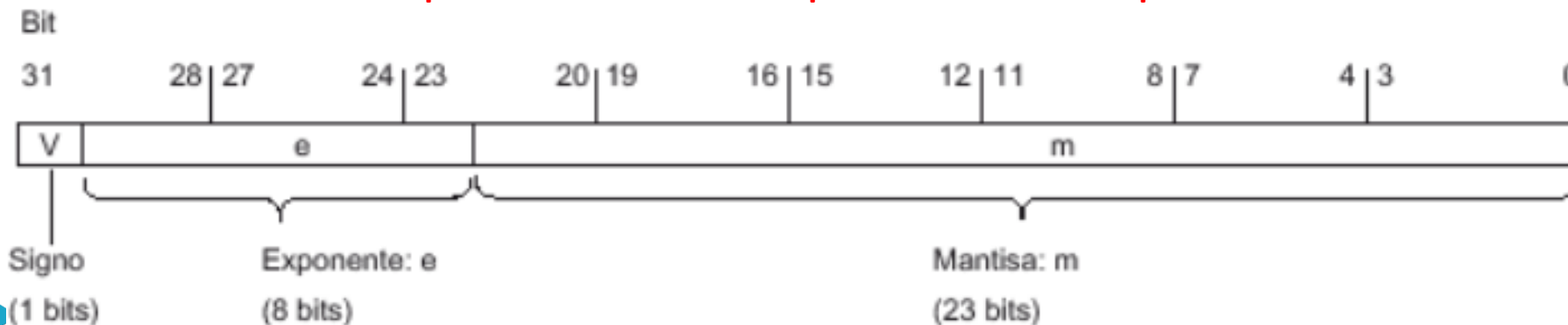


REAL:

Los operandos del tipo de datos REAL tienen una longitud de 32 bits y se utilizan para representar números en coma flotante. Un operando del tipo de datos REAL consta de los tres componentes siguientes:

- Signo: el estado lógico del bit 31 determina el signo. El bit 31 puede adoptar los valores "0" (positivo) o "1" (negativo).
- Exponentes de 8 bits en base 2: El exponente se incrementa en una constante (base, +127), de modo que presenta un rango de valores de entre 0 y 255.

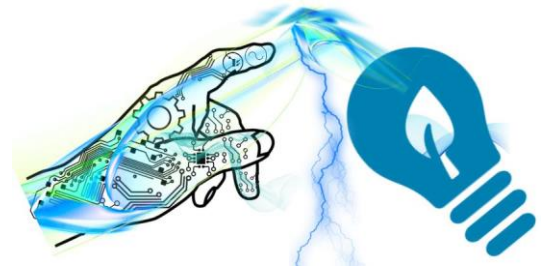
El tipo de datos REAL se procesa con una precisión de 6 posiciones tras la coma.



ELECTROALL

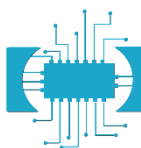
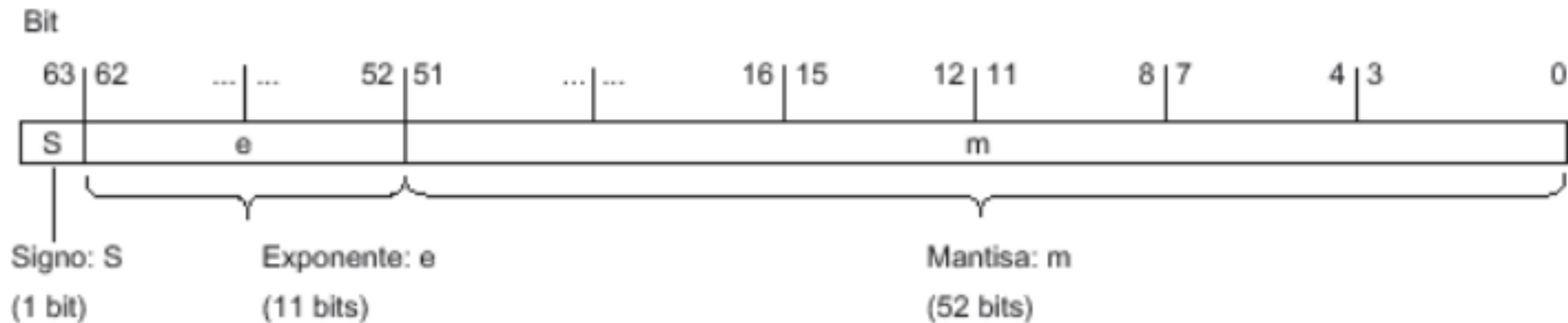


03 NUMEROS EN COMA FLOTANTE



Los operandos del tipo de datos LREAL tienen una longitud de 64 bits y se utilizan para representar valores en coma flotante. Un operando del tipo de datos LREAL consta de los tres componentes siguientes:

- Signo: el estado lógico del bit 63 determina el signo. El bit 63 puede adoptar los valores "0" (positivo) o "1" (negativo).
- Exponentes de 11 bits en base 2: El exponente se incrementa en una constante (base, +1023), de modo que presenta un rango de valores de entre 0 y 2047.





04 TEMPORIZADORES



S5TIME (tiempo) (S7-300, S7-400) 2H 46MIN 30S

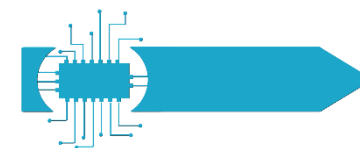
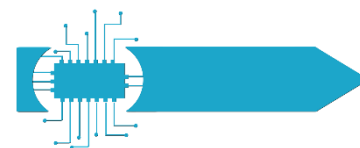
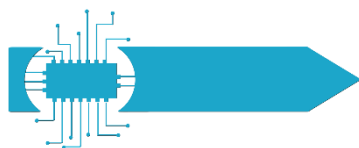
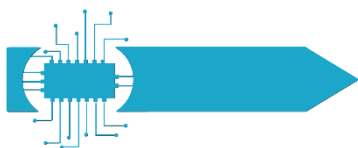
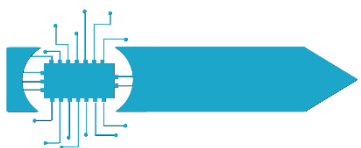
LTIME (temporizador CEI) (S7-1500) 106751 días, 23 horas, 59 minutos, 59 segundos, 999 milisegundos, 999 microsegundos o 999 nanosegundos.

TIME (tiempo IEC):

👉 El contenido de un operando del tipo TIME se interpreta como milisegundos. La representación contiene especificaciones de días (d), horas (h), minutos (m), segundos (s) y milisegundos (ms).

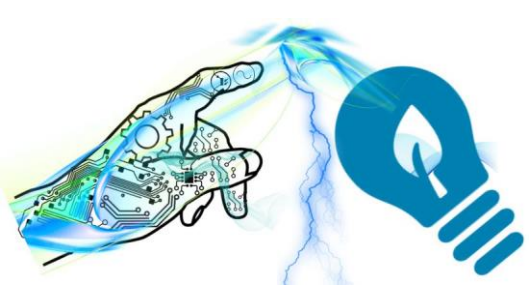
💡 Longitud (bits) 32

👉 No es necesario indicar todas las unidades de tiempo. Por ejemplo, T#5h10s es válido. Si se indica solo una unidad, el valor absoluto de días, horas y minutos no podrá exceder los límites superiores ni inferiores. Si se indica más de una unidad de tiempo, el valor correspondiente no podrá exceder 24 días, 23 horas, 59 minutos, 59 segundos o 999 milisegundos.

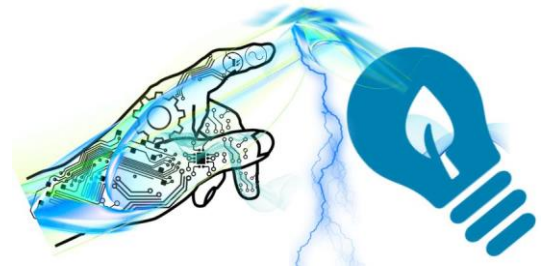


ELECTROALL





05 FECHA Y HORA



👉 5.1 DATE

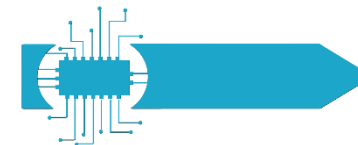
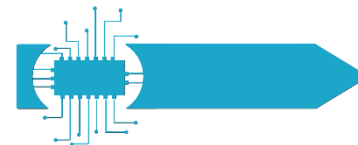
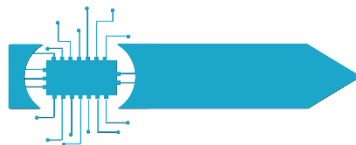
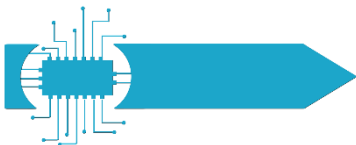
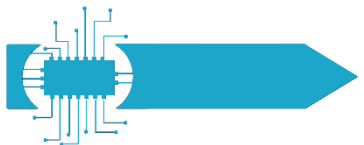
👉 5.2 TOD (TIME OF DAY)

👉 5.3 LTOD (LTIME OF DAY) (S7-1500)

👉 5.4 DT (DATE AND TIME)

👉 5.5 LDT (DATE AND LTIME) (S7-1500)

👉 5.6 DTL (S7-1200, S7-1500)



ELECTROALL





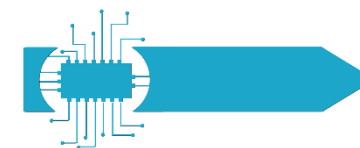
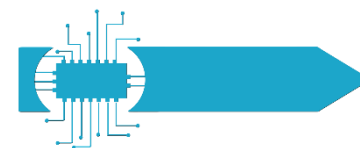
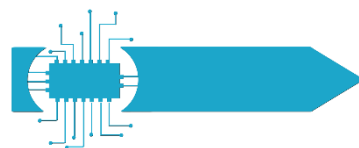
05 FECHA Y HORA

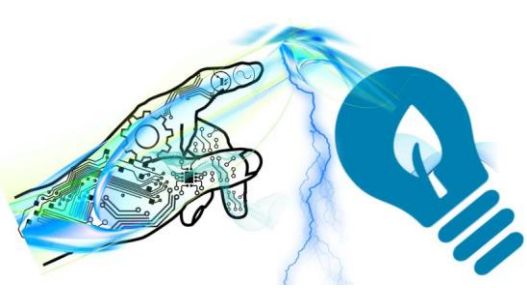


5.1 DATE

- ➡ El tipo de datos DATE guarda una fecha como numero entero sin signo. La representación contiene el año, mes y día
- ➡ El contenido de un operando del tipo de datos DATE corresponde al numero de días desde 01/01/1990, expresado en formato exadecimal

Longitud (bytes)	Formato	Rango de valores	Ejemplos de entrada de valores
2	Fecha IEC (año-mes-día)	De D#1990-01-01 a D#2169-06-06	D#2009-12-31, DATE#2009-12-31
	Números hexadecimales	De 16#0000 a 16#FFFF	16#00F2





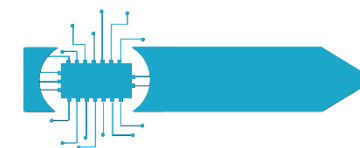
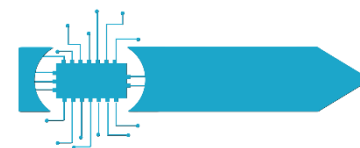
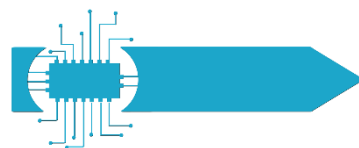
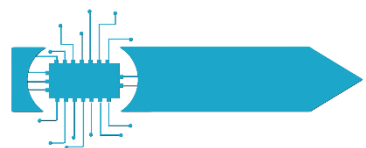
05 FECHA Y HORA

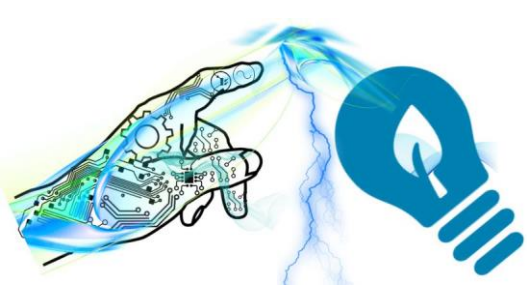


5.2) TOD (TIME_OF_DAY)

👉 El tipo de datos TOD (TIME OF DAY) ocupa una palabra doble (MD) y guarda el número de milisegundos desde el inicio del día, como un número entero sin signo.

Longitud (bytes)	Formato	Rango de valores	Ejemplos de entrada de valores
4	Hora (horas:minutos:segundos.milisegundos)	TOD#00:00:00.000 a TOD#23:59:59.999	TOD#10:20:30.400, TIME_OF_DAY#10:20:30.400





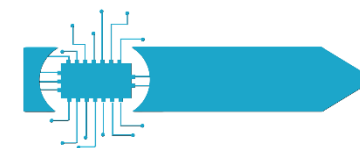
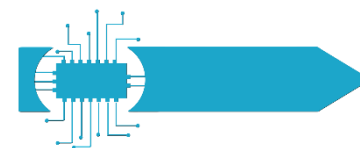
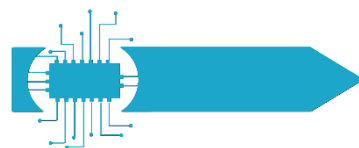
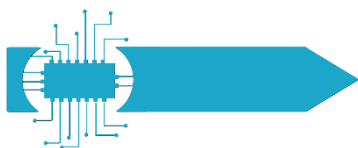
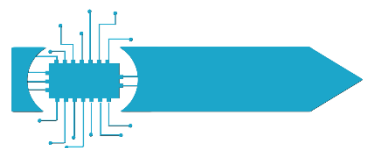
05 FECHA Y HORA



5.3) LTOD (LTIME_OF_DAY) (S7-1500)

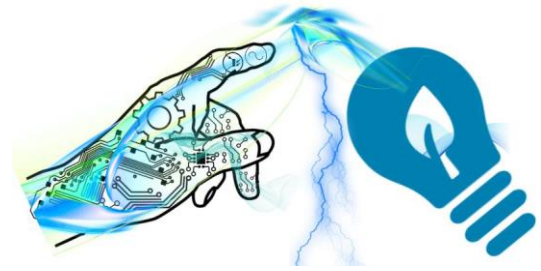
👉 El tipo de datos LTOD (TIME OF DAY) ocupa dos palabras dobles y guarda el numero de nanosegundos desde el comienzo del día, como un numero entero sin signo.

Longitud (bytes)	Formato	Rango de valores	Ejemplos de entrada de valores
8	Hora (horas:minutos:segundos.milisegundos)	de LTOD#00:00:00.000000000 a LTOD#23:59:59.999999999	LTOD#10:20:30.400_365_215, LTIME_OF_DAY#10:20:30.400_365_215





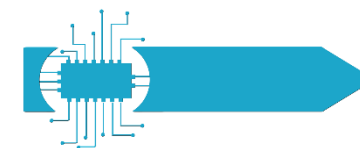
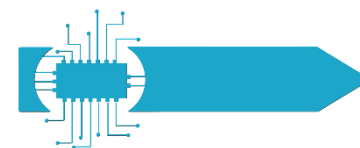
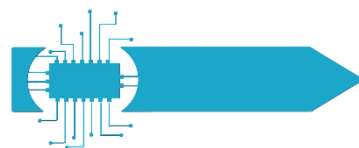
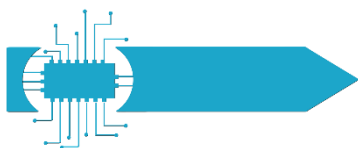
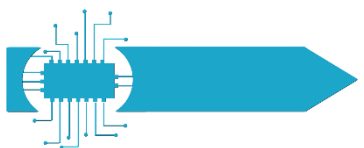
05 FECHA Y HORA



5.4) DT (DATE AND TIME)

👉 El tipo de datos DT (DATE AND TIME) guarda la información en formato BCD

Longitud (bytes)	Formato	Rango de valores	Ejemplo de entrada de valores
8	Fecha y hora (año-mes-día- hora:minuto:segundo:milisegundo ³⁾)	Mín.: DT#1990-01-01- 00:00:00.000 Máx.: DT#2089-12- 31-23:59:59.999	DT#2008-10-25-08:12:34.567, DATE_AND_TIME#2008-10-25- 08:12:34.567

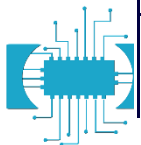


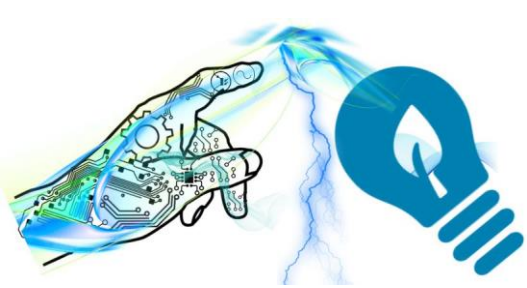


05 FECHA Y HORA

5.4) DT (DATE AND TIME)

Byte	Contenido	Rango de valores
0	1990 Año 90 1001 0000	de 0 a 99 (años 1990 a 2089) BCD#90 = 1990 ... BCD#0 = 2000 ... BCD#89 = 2089
1	Mes	BCD#1 a BCD#12
2	Día	BCD#1 a BCD# 31
3	Hora	BCD#0 a BCD#23
4	Minuto	BCD#0 a BCD#59





05 FECHA Y HORA



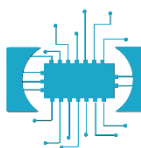
5.4 DT (DATE AND TIME)

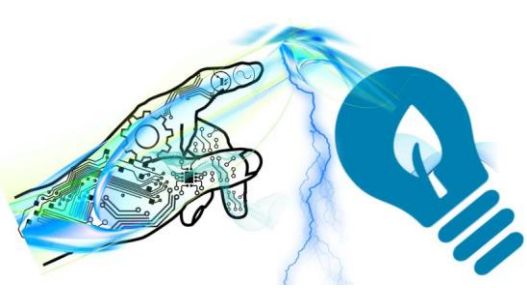
5	Segundo	BCD#0 a BCD#59
6	Los dos valores más significativos de MSEC	BCD#0 a BCD#999
7 (4MSB) ¹⁾	El valor menos significativo de MSEC	BCD#0 a BCD#9
7 (4LSB) ²⁾	Día de la semana	BCD#1 a BCD#7 BCD#1 = domingo ... BCD#7 = sábado

¹⁾ MSB: Most Significant Bit

²⁾ LSB: Least Significant Bit

³⁾ Número en coma fija





05 FECHA Y HORA



5.5 LDT (DATE_AND_LTIME) (S7-1500)

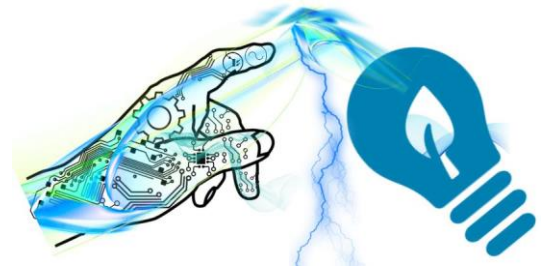
- El tipo de datos LDT (DATE_AND_LTIME) guarda información de fecha y hora en nanosegundos desde el 01.01.1970 0:0.

Longitud (bytes)	Formato	Rango de valores	Ejemplo de entrada de valores
8	Fecha y hora (año-mes-día- hora:minuto:segundo.nanosegundo)	Mín.: LDT#1970-01-01- 00:00:00.000000000 Máx.: LDT#2262-04-11- 23:47:16.854775807	LDT#2008-10-25-08:12:34.567
	Números hexadecimales	de 16#0 a 16#7FFF_FFFF_FFFF_FFFF	16#7FFF





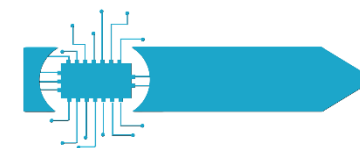
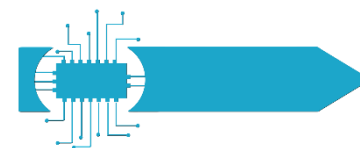
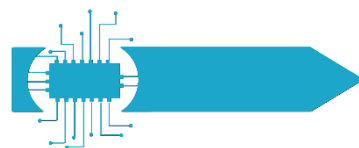
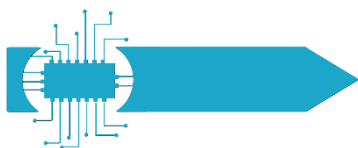
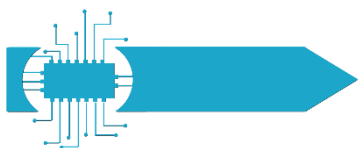
05 FECHA Y HORA



5.6 DTL (S7-1200, S7-1500)

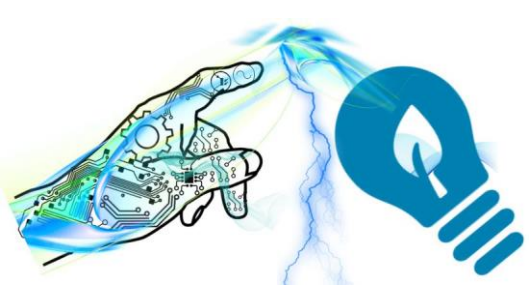
👉 Un operando de tipo de datos DTL tiene una longitud de 12 bytes

Longitud (bytes)	Formato	Rango de valores	Ejemplo de entrada de valores
12	Fecha y hora (año-mes-día- hora:minuto:segundo.nanosegundo)	Mín.: DTL#1970-01-01-00:00:00.0 Máx.: DTL#2262-04-11- 23:47:16.854775807	DTL#2008-12-16- 20:30:20.250



05 FECHA Y HORA

Byte	Componente	Tipo de datos	Rango de valores
0	Año	UINT	de 1970 a 2262
1			
2	Mes	USINT	de 1 a 12
3	Día	USINT	1 hasta 31
4	Día de la semana	USINT	de 1 (domingo) a 7 (sábado) El día de la semana no se tiene en cuenta al introducir los valores.
5	Hora	USINT	0 hasta 23
6	Minuto	USINT	0 hasta 59
7	Segundo	USINT	0 hasta 59
8	Nanosegundo	UDINT	de 0 a 999999999
9			
10			
11			



06 CADENAS DE CARACTERES



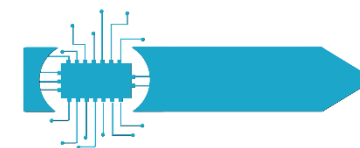
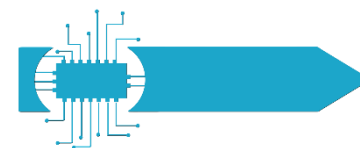
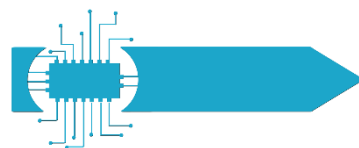
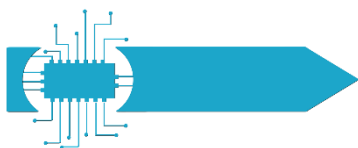
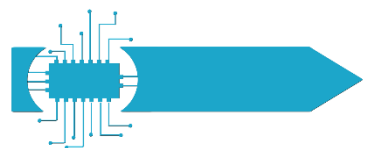
6.1) Carácter

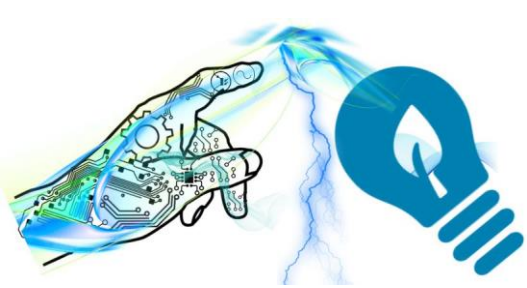
💡 6.1.1 CHAR

💡 6.1.2 WCHAR (S7 1200 – S7 1500)

6.2) Cadenas de caracteres

💡 6.2.1 String





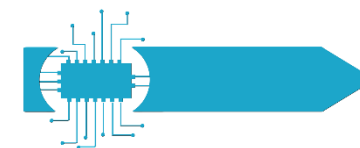
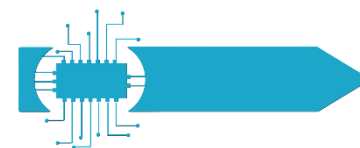
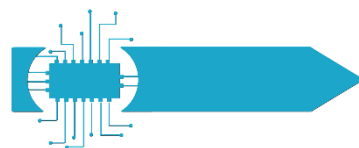
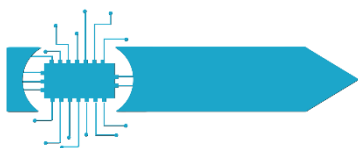
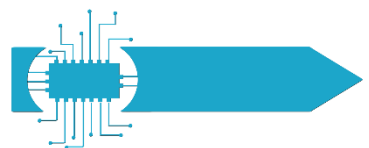
6.1 CHARACTER

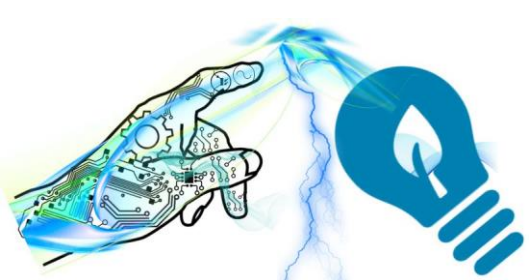
👉 6.1.1 CHAR

- 💡 Una variable de tipo de datos char (character) tiene una longitud de 8 bits.
- 💡 El tipo de datos CHAR almacena solo un carácter depositado en codificación

LA SIGUIENTE TABLA MUESTRA LOS SIGUIENTES VALORES DATOS CHAR

Longitud (bits)	Formato	Rango de valores	Ejemplo de entradas de valores
8	Caracteres ASCII	Juego de caracteres ASCII	'A', CHAR#'A'



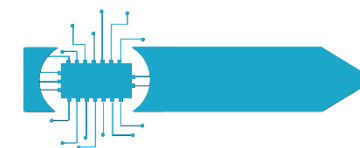
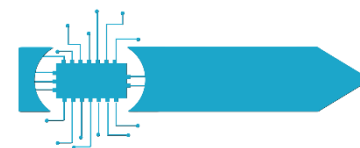
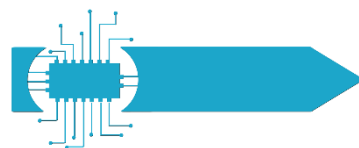
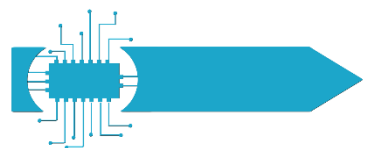


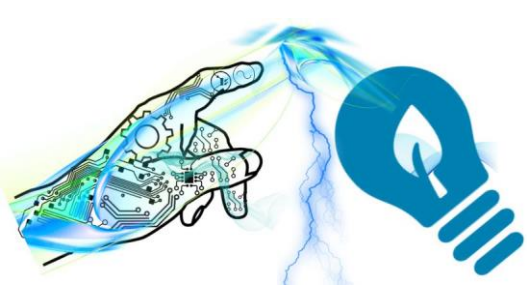
6.1 CHARACTER

6.1.2 WCHAR

- Una variable de tipo de datos WCHAR (wide characters) tiene una longitud de 16 bits
- El tipo de datos WCHAR almacena un carácter individual de un juego de caracteres ampliado con codificación UTF- 16, no obstante, solo se cubre una parte de todo el rango Unicode. Los caracteres no representables se hacen representables con ayuda de un carácter de escape.

Longitud (bits)	Formato	Rango de valores	Ejemplo de entrada de valores
16	Unicode	\$0000 – \$D7FF	WCHAR#'a'



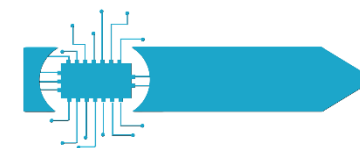
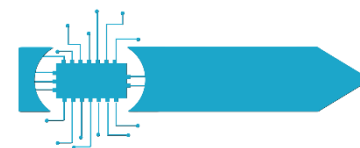
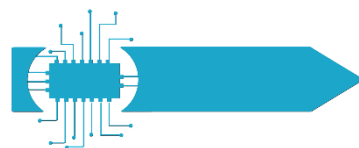
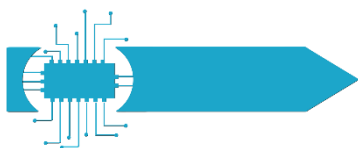
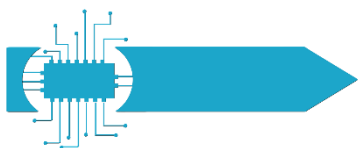


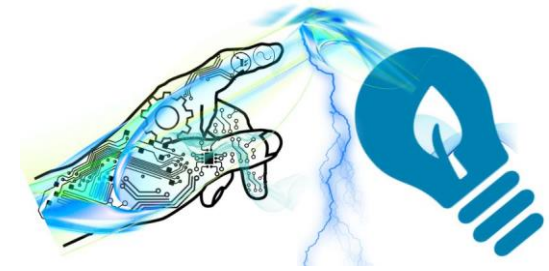
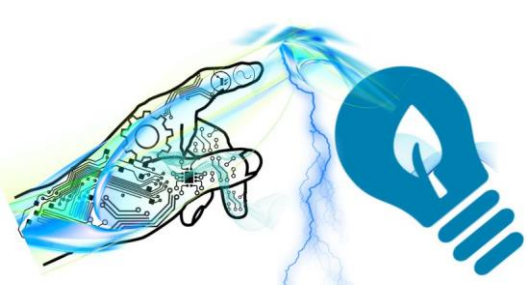
6.2 CADENAS DE CARACTERES



STRING

- 💡 Un operando de tipo de datos STRING guarda guarda varios caracteres en una cadena formada por un máximo de 254 caracteres. En las cadenas de caracteres se admiten todo los caracteres de la pagina de codigos ajustados en el sistema de creación, los caracteres se introducen entre comillas sencillas
- 💡 Una cadena de caractes puede contener también caracteres especiales. Para especificar caracteres de control, caracteres d dólar y comillas simples, se utiliza el carácter de escape \$.

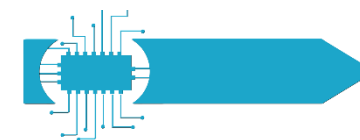
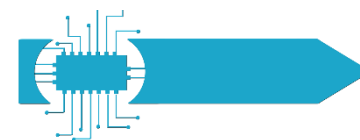
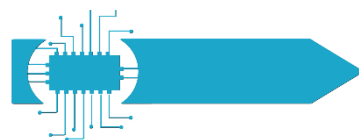
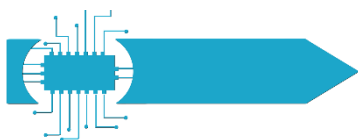
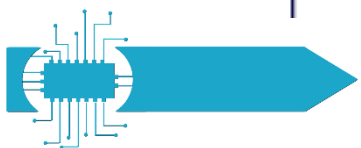




6.2 CADENAS DE CARACTERES

👉 6.2.1 STRING

Longitud (bytes)	Formato	Rango de valores	Ejemplo de entrada de valores
$n + 2$ ¹⁾	Cadena de caracteres ASCII, incluidos los caracteres especiales	0 a 254 caracteres	<ul style="list-style-type: none">• 'Nombre'• STRING#'NAME'• STRING#'Na... (La longitud actual de la cadena es mayor que el espacio disponible en la pantalla).• STRING#' (la cadena está vacía.)
¹⁾ Un operando del tipo de datos STRING ocupa en la memoria dos bytes más que la longitud máxima especificada.			





07 Tipos de datos PLC (UDT)

- ➡ Un tipo de datos PLC (UDT) es un tipo de datos compuesto personalizado que puede emplearse para la declaración de una variable. Representa una estructura de datos de varios componentes de diferentes tipos de datos. Los componentes también pueden derivarse como estructura de otro tipo de datos PLC, de un ARRAY o directamente mediante la palabra clave STRUCT. La profundidad de anidamiento está limitada a 8 niveles.
- ➡ Es posible modificar centralmente un tipo de datos PLC (UDT) y emplearlo varias veces en el código del programa. Todas las ubicaciones (lugares de uso) se actualizan automáticamente.

