



Intégrer un automate Siemens
S7-1200 ou S7-1500 dans un système
(STEP7 - TIA Portal)

- 04 - Les objets de données

Les objets de données

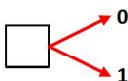
1. LES TYPES DE DONNEES	3
2. LES ENTREES	5
3. LES SORTIES	6
4. LES MEMENTOS	7
5. LES TEMPORISATIONS	8
6. LES COMPTEURS/DECOMPTEURS	9
7. LES BLOCS DE DONNEES (DB)	10
8. LE CODAGE DES DONNEES	11
9. LES DONNEES AU FORMAT BINAIRE DE TYPE ENTIER	15
10. LES DONNEES AU FORMAT REEL	16
11. LES DONNEES AU FORMAT CHAINE DE BITS	17
12. LES DONNEES AU FORMAT BINAIRE DE TYPE « S5TIME ou TIME »	18
13. LES DONNEES AU FORMAT BCD	19
14. LES DONNEES AU FORMAT CHAINE DE CARACTERES	21

Les objets de données

1. LES TYPES DE DONNEES

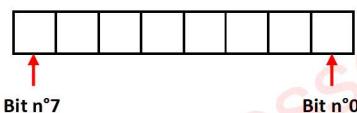
□ BIT

- ✓ information élémentaire « 0 » ou « 1 »



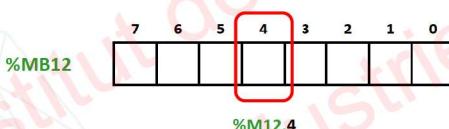
□ OCTET

- ✓ association de 8 BITS CONSECUTIFS



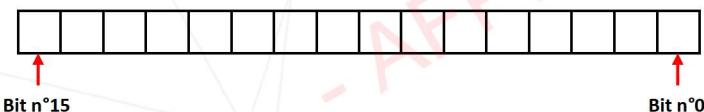
- ✓ Chez Siemens, un bit est extrait d'un octet.

➤ Exemple: le bit %M12.4 appartient à l'octet %MB12.



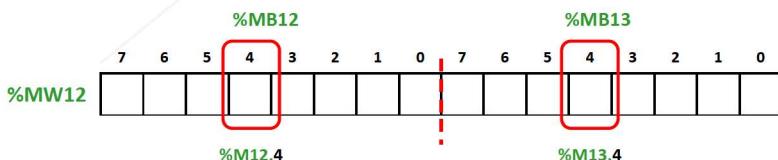
□ MOT

- ✓ association de 16 BITS CONSECUTIFS



- ✓ Chez Siemens, un mot est constitué de deux octets consécutifs

➤ Exemple: le mot %MW12 est constitué des octets %MB12 et %MB13.

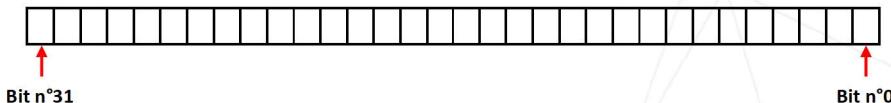


Les objets de données

1. LES TYPES DE DONNEES

DOUBLE MOT

- ✓ association de 32 BITS CONSECUTIFS

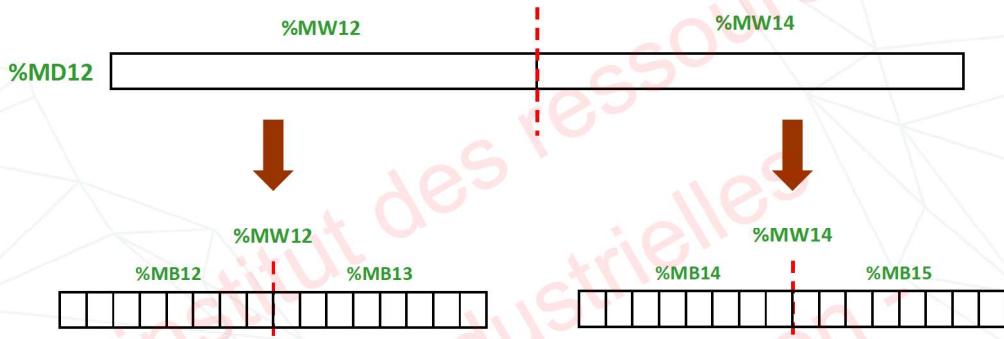


Bit n°31

Bit n°0

✓ Chez Siemens, un double mot est constitué de quatre octets consécutifs

➤ Exemple: le double mot %MD12 est constitué des octets %MB12, %MB13, %MB14 et %MB15. Le double mot suivant sera %MD16.



%MB12

%MB13

%MB14

%MB15

Les objets de données

2. LES ENTREES

□ LES BITS D'ENTRÉES

✓ Adressage SIEMENS :

E . y

- ❖ E : entrée
- ❖ x : adresse de l'octet d'entrées
- ❖ y : numéro du bit de l'entrée

✓ Adressage INTERNATIONAL:

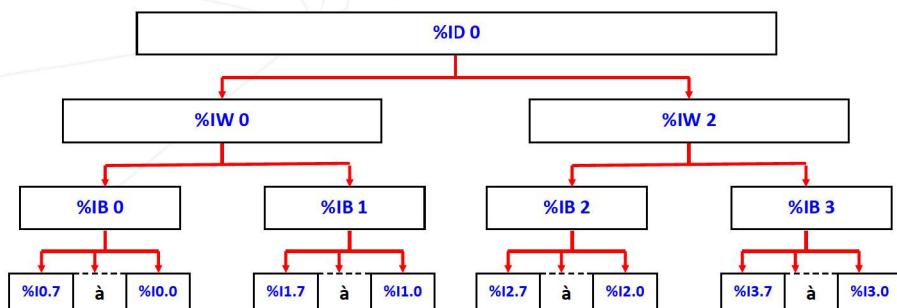
%I x . y

- ❖ %I : input
- ❖ x : adresse de l'octet d'entrées
- ❖ y : numéro du bit de l'entrée

□ LES ENTREES NUMERIQUES

Octet	EB ou %IB
Mot	EW ou %IW
Double mot	ED ou %ID

□ REGLES DE RECOUVREMENT



Les objets de données

3. LES SORTIES

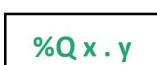
LES BITS DE SORTIES

✓ Adressage SIEMENS :



- ❖ **A** : sortie
- ❖ **x** : adresse de l'octet de sorties
- ❖ **y** : numéro du bit de la sortie

✓ Adressage INTERNATIONAL:

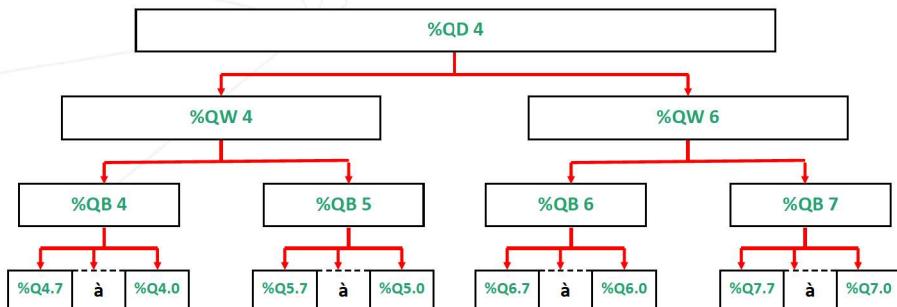


- ❖ **%Q** : sortie
- ❖ **x** : adresse de l'octet de sorties
- ❖ **y** : numéro du bit de la sortie

LES SORTIES NUMERIQUES

Octet	AB ou %QB
Mot	AW ou %QW
Double mot	AD ou %QD

REGLES DE RECOUVREMENT



Les objets de données

4. LES MEMENTOS

□ STRUCTURE

- ✓ Les bits de mémentos sont utilisés pour les résultats intermédiaires des opérations logiques réalisées dans le programme.
- ✓ Ils s'utilisent comme une sortie mais leur état logique ne commande pas d'interface.

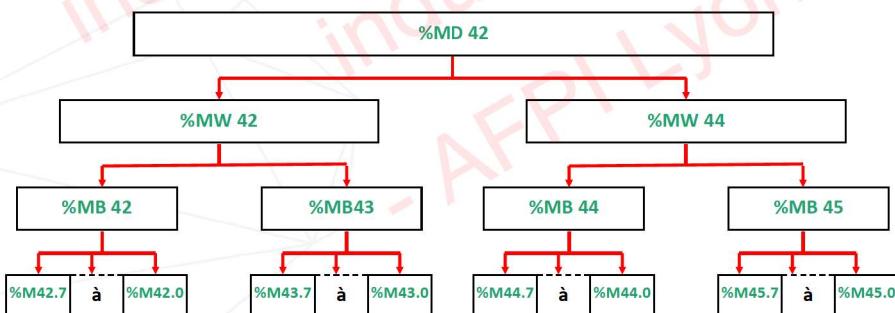
%M x . y

- ❖ %M : mémento
- ❖ x : adresse de l'octet
- ❖ y : numéro du bit

- ✓ Les mémentos s'utilisent également au format octet, mot ou double mot pour des opérations arithmétiques réalisées dans le programme.

Octet	%MB
Mot	%MW
Double mot	%MD

□ REGLES DE RECOUVREMENT



Les objets de données

5. LES TEMPORISATIONS

PRESENTATION

- ✓ Une temporisation permet de :
 - de gérer des temps d'attente,
 - de gérer des temps de contrôle,
 - de générer des impulsions,
 - de mesurer le temps.

REPRESENTATION

TEMPORISATION	REPRESENTATION TIA Portal
Impulsion	TP
Retard à la montée	TON
Retard à la retombée	TOF
Accumuler temporisation	TONR

- ✓ Il est possible de définir par configuration un nombre de temporisations sauvegardées.

Les objets de données

6. LES COMPTEURS/DECOMPTEURS

PRESENTATION

- ✓ Un compteur permet de réaliser des fonctions:
 - de comptage avec remise à zéro,
 - de décomptage avec une présélection,
 - de comptage et de décomptage.

REPRESENTATION

COMPTEUR	REPRESENTATION TIA Portal
Compteur incrémental	CTU
Compteur décrémental	CTD
Compteur incrémental / décrémental	CTUD

- ✓ Il est possible de définir par configuration un nombre de compteurs sauvegardés.

Les objets de données

7. LES BLOCS DE DONNEES (DB)

PRESENTATION

- ✓ Les blocs de données contiennent des informations pour le programme.
- ✓ Ils peuvent soit servir à tous les blocs de code (DB globaux), soit être associés à un bloc fonctionnel (FB) ou à un bloc fonctionnel système (SFB) spécifique (DB d'instance).

REPRESENTATION DES BLOCS DE DONNEES GLOBAUX

TYPE	ADRESSE	EXEMPLE	
Bit de données	DBX	DB1.DBX0.2	Bit 2 de l'octet 0 de DB 1
Octet de données	DBB	DB1.DBB4	octet 4 de DB 1
Mot de données	DBW	DB1.DBW8	Mot 8 de DB 1
Double mot de données	DBD	DB1.DBD10	Double Mot 10 de DB 1

REPRESENTATION DES BLOCS DE DONNEES D'INSTANCE

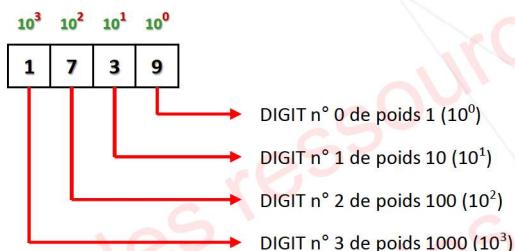
TYPE	ADRESSE	EXEMPLE	
Bit de données	DIX	DIX0.2	Bit 2 de l'octet 0 du DB d'instance ouvert
Octet de données	DIB	DIB4	octet 4 du DB d'instance ouvert
Mot de données	DIW	DBW8	Mot 8 du DB d'instance ouvert
Double mot de données	DID	DBD10	Double Mot du DB d'instance ouvert

Les objets de données

8. LE CODAGE DES DONNEES

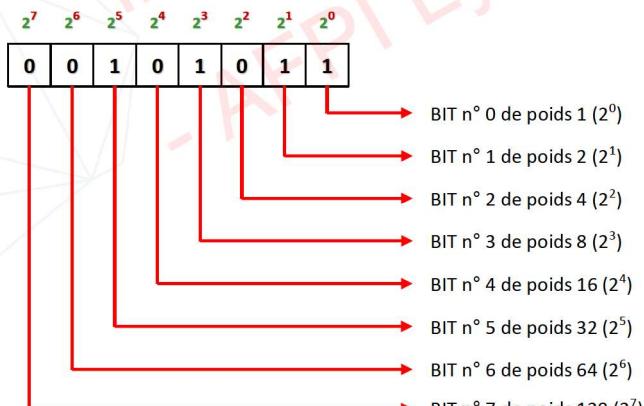
□ LE CODE DECIMAL

- ✓ Les caractères (DIGITS) qui composent le système décimal (**base 10**) sont compris entre 0 et 9.
- ✓ Un nombre, écrit sans précision de sa base, est de manière implicite décimal.
- ✓ La valeur est signée ou non signée.
- ✓ Une valeur signée est précédée par le symbole « - » pour les valeurs négatives et par le symbole « + » pour les valeurs positives (optionnel).
- ✓ **Exemple:** $N_{10} = 1739$



□ LE CODE BINAIRE PUR 2#

- ✓ Base utilisée en microélectronique.
- ✓ La base 2 est composé des DIGITS 0 et 1; le digit est alors appelé BIT.
- ✓ Un nombre, écrit en binaire, est précédé du préfixe 2#.
- ✓ **Exemple:** $N_2 = 2\#00101011$



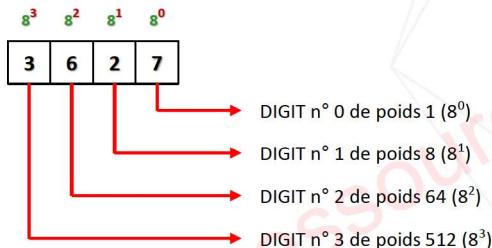
- ✓ Pour coder le nombre binaire en décimal, on ajoute les poids des bits à l'état 1
- ✓ $N_{10} = 32 + 8 + 2 + 1$
- ✓ $N_{10} = 43$

Les objets de données

8. LE CODAGE DES DONNEES

□ LE CODE OCTAL 8#

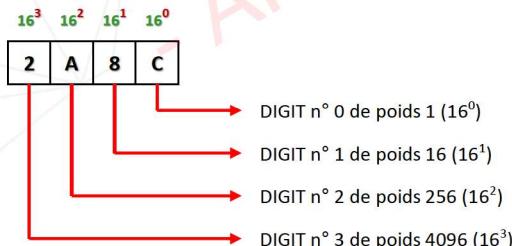
- ✓ Code utilisé en électronique numérique mais il est de moins en moins utilisé de nos jours.
- ✓ Les DIGITS qui composent la base octale (**base 8**) sont compris entre 0 et 7.
- ✓ Le nombre octal est précédé du préfixe 8#.
- ✓ **Exemple:** $N_8 = 3627$



- ✓ Pour coder le nombre octal en décimal, on fait la somme des digits multipliés par leur poids.
- ✓ $N_{10} = (3 \times 512) + (6 \times 64) + (2 \times 8) + (7 \times 1)$
- ✓ $N_{10} = 1536 + 384 + 16 + 7$
- ✓ $N_{10} = 1943$

□ LE CODE HEXADECIMAL 16#

- ✓ La base hexadécimale utilisent 16 digits: les chiffres 0 à 9 et les 6 premières lettres de l'alphabet A, B, C, D, E et F.
- ✓ Le nombre hexadécimal est précédé du préfixe 16#.
- ✓ **Exemple:** $N_{16} = 2A8C$



- ✓ Pour coder le nombre hexadécimal en décimal, on fait la somme des digits multipliés par leur poids.
- ✓ $N_{10} = (2 \times 4096) + (10 \times 256) + (8 \times 16) + (12 \times 1)$
- ✓ $N_{10} = 8192 + 2560 + 128 + 12$
- ✓ $N_{10} = 10892$

Les objets de données

8. LE CODAGE DES DONNEES

□ RECAPITULATIF

DECIMAL	BINAIRE PUR	OCTAL	HEXADECIMAL
00	2#0000 0000	8#00	16#00
01	2#0000 0001	8#01	16#01
02	2#0000 0010	8#02	16#02
03	2#0000 0011	8#03	16#03
04	2#0000 0100	8#04	16#04
05	2#0000 0101	8#05	16#05
06	2#0000 0110	8#06	16#06
07	2#0000 0111	8#07	16#07
08	2#0000 1000	8#10	16#08
09	2#0000 1001	8#11	16#09
10	2#0000 1010	8#12	16#0A
11	2#0000 1011	8#13	16#0B
12	2#0000 1100	8#14	16#0C
13	2#0000 1101	8#15	16#0D
14	2#0000 1110	8#16	16#0E
15	2#0000 1111	8#17	16#0F
16	2#0001 0000	8#20	16#10
17	2#0001 0001	8#21	16#11
18	2#0001 0010	8#22	16#12
19	2#0001 0011	8#23	16#13
20	2#0001 0100	8#24	16#14

Les objets de données

8. LE CODAGE DES DONNEES

□ LE CODE BCD

- ✓ Le code **BCD** (Binary Coded Decimal) ou DCB (Décimal Codé Binaire) permet de représenter les chiffres décimaux 0 à 9 à l'aide de 4 bits codés en binaire pur.

DECIMAL	BCD			
	2^3	2^2	2^1	2^0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

- ✓ **Exemple 1:** valeur décimale « 5207 » codée sur un format 16 bits

Valeur décimale	5	2	0	7
Valeur BCD	0101	0010	0000	0111

- ✓ **Exemple 2:** valeur décimale « 7816329 » codée sur un format 32 bits

Valeur décimale	0	7	8	1	6	3	2	9
Valeur BCD	0000	0111	1000	0001	0110	0011	0010	1001

Les objets de données

9. LES DONNEES AU FORMAT BINAIRE DE TYPE ENTIER

□ ENTIERS SIGNES AU FORMAT 16 BITS

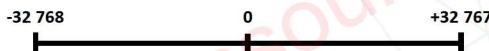
INT

✓ SIGNE

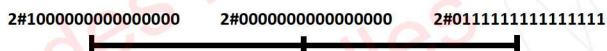
- Pour une donnée signée, le bit n°15 est le bit de signe;
- Par convention, il vaut:
 - ❖ 0 pour une donnée positive
 - ❖ 1 pour une donnée négative
- Sa valeur (poids) est égale à $-2^{15} = -32\ 768$

✓ VALEURS LIMITES

- Décimal:



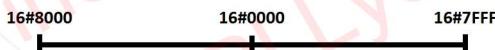
- Binaire Pur:



$$\text{Décimal: } -32768 = -1 \times 2^{15}$$

$$\text{Décimal: } 32767 = 1 \times 2^{14} + 1 \times 2^{13} + 1 \times 2^{12} + \dots + 1 \times 2^1 + 1 \times 2^0$$

- Hexadécimal:



$$\text{Décimal: } -32768 = -8 \times 16^3$$

$$\text{Décimal: } 32767 = 7 \times 16^3 + 15 \times 16^2 + 15 \times 16^1 + 15 \times 16^0$$

Les objets de données

9. LES DONNEES AU FORMAT BINAIRE DE TYPE ENTIER

ENTIERS SIGNES AU FORMAT 32 BITS

DINT

✓ SIGNE

- Pour une donnée signée, le bit n°31 est le bit de signe;
- Par convention, il vaut:
 - ❖ 0 pour une donnée positive
 - ❖ 1 pour une donnée négative
- Sa valeur (poids) est égale à $-2^{31} = -2\ 147\ 483\ 648$

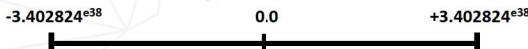
✓ VALEURS LIMITES

	MINIMUM	MAXIMUM
DECIMAL	L# - 2 147 483 648	L# + 2 147 483 647
BINAIRE PUR	$2^{31} 2^{30} 2^{29} \dots 2^1 2^0$	$2^{31} 2^{30} 2^{29} \dots 2^1 2^0$
HEXADECIMAL	$16^7 16^6 16^5 16^4 16^3 16^2 16^1 16^0$ 16# 8 0 0 0 0 0 0 0	$16^7 16^6 16^5 16^4 16^3 16^2 16^1 16^0$ 16# 7 F F F F F F F

10. LES DONNEES AU FORMAT REEL

PRESENTATION REAL

- ✓ Le format Real (virgule flottante) est codé sur 32 bits et correspond aux nombres à virgule flottante à décimale unique.
- ✓ Exemples: 10.0 56.5 -38.78
- ✓ Valeurs limites:



Les objets de données

11. LES DONNEES AU FORMAT CHAINE DE BITS

□ PRESENTATION

- ✓ L'ensemble des bits qui composent ce format ne représente pas une valeur numérique.
- ✓ Les données appartenant à ce format peuvent être représentées:
 - en hexadécimal (16#),
 - en octal (8#),
 - en binaire (2#).

□ LE TYPE

BYTE

- ✓ Il est codé sur un format de **8 bits**.

Base	Limite inférieure	Limite supérieure	Exemples
Hexadécimale	16#00	16#FF	16#A5
Octale	8#000	8#377	8#245
Binaire	2#0000 0000	2#1111 1111	2#1010 0101

□ LE TYPE

WORD

- ✓ Il est codé sur un format de **16 bits**.

Base	Limite inférieure	Limite supérieure	Exemples
Hexadécimale	16#0000	16#FFFF	16#2D3
Octale	8#000000	8#177777	8#001323
Binaire	2#0000 0000 0000 0000	2#1111 1111 1111 1111	2#0000 0010 1101 0011

□ LE TYPE

DWORD

- ✓ Il est codé sur un format de **32 bits**.

Base	Limite inférieure	Limite supérieure
Hexadécimale	16#0	16#FFFF FFFF
Octale	8#0	8#3777777777
Binaire	2#0	2#1111 1111 1111 1111 1111 1111 1111 1111

- ✓ Exemples: 16#8CA09E7F
- ✓ 8#21450117177
- ✓ 2#1000 1100 1010 0000 1001 1110 0111 1111

Les objets de données

12. LES DONNEES AU FORMAT BINAIRE DE TYPE « S5TIME ou TIME »

LE TYPE HEURE **S5TIME# ou TIME#**

- ✓ **S5TIME:** durée S7 au pas de 10ms.
- ✓ **TIME:** durée CEI au pas de 1ms.
- ✓ Les unités de temps autorisées pour la représentation des valeurs sont les suivantes:
 - **J:** jours
 - **H:** heures
 - **M:** minutes
 - **S:** secondes
 - **MS:** millisecondes
- ✓ **Exemple:** « 1 jour 15 heures 21 minutes 10 secondes 400 millisecondes »

REPRESENTATION	COMMENTAIRE
T#1J_15H_21M_10S_400MS	Valeur en jours\heures\minutes\secondes\millisecondes
T#39H_21M_10S_400MS	Valeur en heures\minutes\secondes\millisecondes
T#2361M_10S_400MS	Valeur en minutes\secondes\millisecondes
T#141670S_400MS	Valeur en secondes\millisecondes
T#141670400MS	Valeur en millisecondes

Les objets de données

13. LES DONNEES AU FORMAT BCD

LE TYPE « DATE » D#

- ✓ Le type Date, codé sur un format de **32 bits**, contient les informations suivantes:
 - Année: codée sur les 16 bits de poids forts (1990 à 2099)
 - Mois: codé sur 8 bits (01 à 12)
 - Jour: codé sur les 8 bits de poids faibles (01 à 31)
- ✓ Syntaxe: D#<Année>-<Mois>-<Jour>
- ✓ Le « 0 » de gauche du mois et du jour est toujours affiché mais peut être omis lors de la saisie.
- ✓ Exemple: « 6 Novembre 2012 »

D#2012-11-6

	ANNÉE				MOIS		JOUR	
Valeur décimale:	2	0	1	2	1	1	0	6
Valeur BCD:	0010	0000	0001	0010	0001	0001	0000	0110

LE TYPE « TIME OF DAY » TOD#

- ✓ Le type Time of Day, codé sur un format de **32 bits**, contient les informations suivantes:
 - Heures: codées sur les 8 bits de poids forts (00 à 23)
 - Minutes: codées sur 8 bits (00 à 59)
 - Secondes: codées sur 8 bits (00 à 59)
- ✓ Les 8 bits de poids faibles ne sont pas utilisés.
- ✓ Syntaxe: TOD#<Heures>:<Minutes>:<Secondes>
- ✓ Le « 0 » de gauche des heures, des minutes et des secondes est toujours affiché mais peut être omis lors de la saisie.
- ✓ Exemple: « 16 heures 8 minutes 52 secondes »

TOD#16:08:52

	HEURES		MINUTES		SECONDES		INUTILISES	
Valeur décimale:	1	6	0	8	5	2	0	0
Valeur BCD:	0001	0110	0000	1000	0101	0010	0000	0000

Les objets de données

13. LES DONNEES AU FORMAT BCD

LE TYPE « DATE AND TIME » DT#

- ✓ Le type Date and Time, codé sur un format de 64 bits, contient les informations suivantes:
 - **Année:** codée sur les 16 bits de poids forts (1990 à 2099)
 - **Mois:** codé sur 8 bits (01 à 12)
 - **Jour:** codé sur les 8 bits de poids faibles (01 à 31)
 - **Heures:** codées sur les 8 bits de poids forts (00 à 23)
 - **Minutes:** codées sur 8 bits (00 à 59)
 - **Secondes:** codées sur 8 bits (00 à 59)
- ✓ Les 8 bits de poids faibles ne sont pas utilisés.
- ✓ **Syntaxe:** DT#<Année>-<Mois>-<Jour>-<Heures>:<Minutes>:<Secondes>
- ✓ Le « 0 » de gauche du mois, du jour, des heures, des minutes et des secondes est toujours affiché mais peut être omis lors de la saisie.
- ✓ **Exemple:** « 6 Novembre 2012 à 16 heures 25 minutes 6 secondes »
DT#2012-11-6-16:25:6

	ANNÉE				MOIS		JOUR		HEURES		MINUTES		SECONDES		INUTILISES	
DECIMAL:	2	0	1	2	1	1	0	6	1	6	2	5	0	6	0	0
BCD:	0010	0000	0001	0010	0001	0001	0000	0110	0001	0110	0010	0101	0000	0110	0000	0000

Les objets de données

14. LES DONNEES AU FORMAT CHAINE DE CARACTERES

PRÉSENTATION STRING

- ✓ Le format chaîne de caractères permet de représenter des caractères ASCII.
- ✓ Chaque caractère (CHAR) est codé sur 8 bits ce qui permet de coder 256 caractères.
- ✓ Les caractères 0 à 127 sont communs à toutes les langues.

		MSB	0	1	2	3	4	5	6	7
		LSB	000	001	010	011	100	101	110	111
0	0000	NUL	DLE	SP	0	@	P	`	p	
1	0001	SOH	DC1	!	1	A	Q	a	q	
2	0010	STX	DC2	"	2	B	R	b	r	
3	0011	ETX	DC3	#	3	C	S	c	s	
4	0100	EOT	DC4	\$	4	D	T	d	t	
5	0101	ENQ	NAK	%	5	E	U	e	u	
6	0110	ACK	SYN	&	6	F	X	f	v	
7	0111	BEL	ETB	'	7	G	W	g	w	
8	1000	BS	CAN	(8	H	X	h	x	
9	1001	HT	EM)	9	I	Y	i	y	
A	1010	LF	SUB	*	:	J	Z	j	z	
B	1011	VT	ESC	+	;	K	[k	}	
C	1100	FF	FS	,	<	L	\	l		
D	1101	CR	GS	-	=	M]	m	{	
E	1110	SO	RS	.	>	N	^	n	~	
F	1111	SI	US	/	?	O	_	o	DEL	

« W »	MSB	LSB
BINAIRE PUR	2^2 1	2^1 0 2^0 1
HEXADECIMAL	5	7

- ✓ Les caractères 128 à 255 varient selon les langues.
- ✓ La taille maximale d'une chaîne de caractères est de 65 535 caractères.
- ✓ La saisie est précédée et terminée par le caractère apostrophe « ' » (code ASCII 16#27).

DECLARATION D'UNE VARIABLE ASCII

STRING[<n>]:

taille fixe de « n » caractères