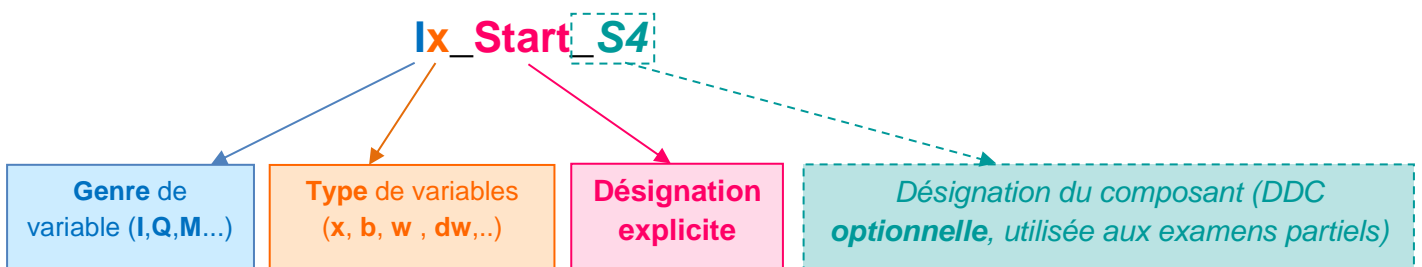


Notation symbolique

Dans les travaux de programmation, les variables sont écrites suivant une symbolique particulière. Cette démarche est faite dans un but de standardisation et de ligne de conduite. La symbolique nous apporte plusieurs informations, comme le type de variable, la désignation de la fonction, etc...

La programmation en langage ST est une suite de textes représentant des instructions, des variables, des déclarations, etc. Sa clarté de lecture peut en être affaiblie si la symbolique des variables est trop longue ou mal formulée. Il est donc important d'établir certaines règles.

Cette symbolique s'applique également à d'autres langages que le « texte structuré » (LD, FBD..)



Genre de variable

Le nom symbolique commence toujours par le **genre** de variable que l'on veut traiter (entrée **I**, sortie **Q**, memento **M**). Grâce aux interface de développement actuel (TIA Portal, TwinCat3...), cela permet une programmation rapide grâce à l'usage de l'**IntelliSense**.

Genre de variable	Type	Adressage absolu
I	Input, Entrée Physique	%I 10.0
Q	Output, Sortie physique	%Q 4.7
M	Mémento	%M 50.2
TON, TOF, TP	Temporisation	%DB 1
CTU, CTD, CTUD	Compteur	%DB2

Genre de variable	Type	Adressage absolu
I	Input, Entrée Physique	%I 10.0
Q	Output, Sortie physique	%Q 4.7
M	Mémento	%M 50.2
TON, TOF, TP	Temporisation	%DB 1
CTU, CTD, CTUD	Compteur	%DB2

Type de variables.

Le type de variables renseigne sur le format utilisé. Il est important de connaître le format des variables lorsque l'on fait des opérations mathématiques, de transfert, etc., car elles doivent être compatibles les unes aux autres. L'importance de cette information se fait sentir lors de l'utilisation d'un grand nombre de variables globales et locales et qu'il n'est pas toujours facile de retrouver l'endroit de leur déclaration. Le type s'écrit en minuscule. Voici une liste des différents types possibles :

Type symbolique	Type	Valeur min	Valeur max	Mémoire requise
x	BOOL	FALSE ou « 0 »	TRUE ou « 1 »	1 bit
b	BYTE	0	255	8 bits
w	WORD	0	65535	16 bits
dw	DWORD	0	4294967295	32 bits
si	SINT	-128	127	8 bits
us	USINT	0	255	8 bits
in	INT	-32768	32767	16 bits
ui	UINT	0	65535	16 bits
ud	UDINT	0	4294967295	32 bits
di	DINT	-2147483648	2147483647	32 bits
r	REAL	$\sim -3.40 \times 10^{38}$	$\sim 3.40 \times 10^{38}$	32 bits
lr	LREAL	$\sim -1.79 \times 10^{308}$	$\sim 1.79 \times 10^{308}$	64 bits
st	STRING		Max 80 caractères	80 + 1 bytes
da	DATE	D#1970-01-01	D#2106-02-06	32 bits
dt	DATE_AND_TIME	DT#1970-01-01-00 :00	DT#2106-02-06-06 :28 :15	32 bits
t	TIME	T#0ms	T#71582m47s295ms	32 bits
td	TIME_OF_DAY	TOD#00 :00	TOD#1193 :02 :47.295	32 bits
rt	Rising edge-Trigger			
ft	Falling edge-Trigger			

		Caractéristiques	Remarques	
su	STRUCT	Type de données utilisateur		
ar	ARRAYS	Type de données utilisateur	Jusqu'à 3 dimensions	
po	POINTER	Type de données utilisateur	Accès direct par adresse	
en	ENUM	Type de données utilisateur	Déclaration en tant que nom	
al	ALIAS	Type de données utilisateur	Déclaration en tant que	
ch	CHAR	-128	127	8 bits
fc	FUNCTION	Fonction		
fb	FUNCTION_BLOCK	Bloc fonctionnel		
ac	ACTION			
pr	PROGRAMME			

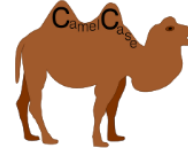
ut	UDT	Type de données utilisateur	<i>types ne concernant que les symboliques en programmation Step7</i>	
db	DBx	Bloc de données		
sd	SDBx	Bloc de données système		
ob	OBx	Bloc opérationnel		
an	ANY	Paramètre ANY		

Désignation

La désignation représente la définition de la variable. Il faut qu'elle soit la plus claire possible. La symbolique d'une variable ne doit pas atteindre une longueur trop importante, c'est pourquoi un nombre maximum de caractères doit être imposé. Ce choix se porte à **24 caractères au total**, y compris le type et le genre, (une tolérance de +2 est possible pour les cas ambigus).

Chaque mot commence par une **majuscule** de manière à éviter le « _ » qui tend à diminuer le nombre de caractères significatifs. On utilise la notation dite *Camel* <http://en.wikipedia.org/wiki/CamelCase> permettant une lecture efficace.

La désignation d'une variable se fait de la manière suivante: **MyCar**



Exemples de désignations : ..VitesseActuelleMot...
..TimerStartRequis
...BpStart...

Remarque :

Dans la plupart des entreprises, les commentaires de la programmation se font en anglais. Ceci en raison de l'exportation réalisée par l'industrie en Suisse.

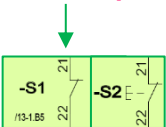
Exemples pour la notation symbolique

Ix_Start_S4 : Bouton poussoir Start S4 relié à l'adresse %I0.3 (I= entrée, x= BOOL, S4=DDC)

Ix_CubePrésent_B6 : Capteur B6 pour la détection d'un cube (I=entrée, x=BOOL, B6=DDC)

Ix_nArrêtUrgence_S1 : Coup de poing Arrêt d'urgence S1 relié à un **contact normalement fermé (n)**

Ix_nStop_S2 : Bouton poussoir S2: Stop de l'installation S2 relié à un **contact normalement fermé (n)**



Qx_SortirVérinM1_M1.2 : Electrovanne M1.2 commandant le vérin M1

Qx_VoyantLumineuxMagasinVide : Voyant lumineux P2

Mx_MagasinVide : Mémento magasin vide

Mx_MémentoDeCadence1Hz : Mémento de cadence de 1Hz %M100.5:

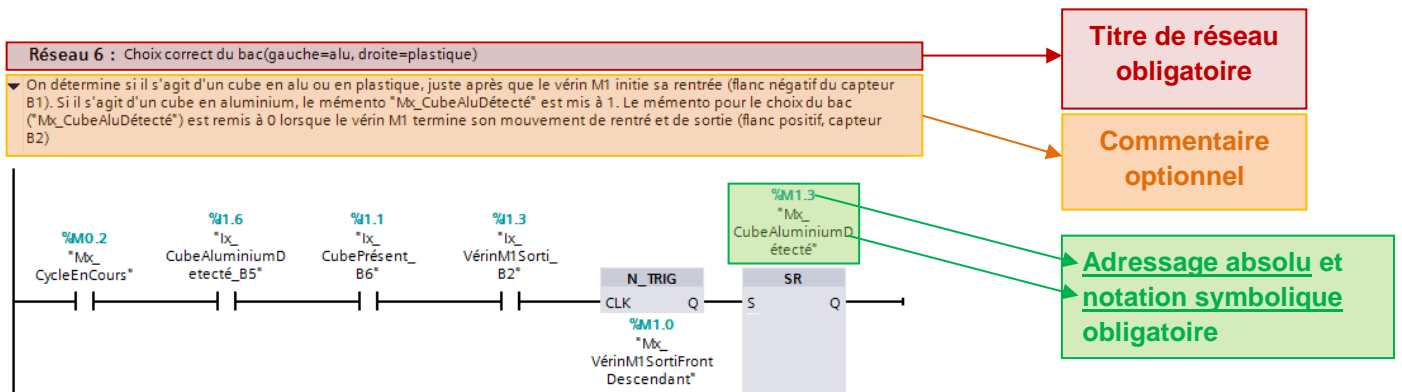
TON_MagasinVide : Temporisation à l'enclenchement magasin vide

TOF_SortirVérinM2 : Temporisation au déclenchement pour la sortie du vérin M2

CTU_NombreCubesTriés : Compteur CTU comptant le nombre de cubes triés:

Titres de réseaux et commentaires

En plus de la symbolique facilitant la lecture et la compréhension du programme, des titres de réseaux **claires et compréhensibles** sont **obligatoires** pour le rendre encore plus claire. Des commentaires **optionnels** peuvent dans certains cas être rajoutés afin d'expliquer plus en détail certaines fonctions.



Abréviations optionnelles

Il arrive fréquemment que des variables deviennent trop longues et sont scindées en plusieurs lignes dans les programmes. Afin de faciliter cette lecture, voici une liste d'abréviations pouvant être employées de manière facultative afin de raccourcir vos noms de variable.

Abréviation	Terme	Abréviation	Terme	Abréviation	Terme
Alm	Alarme (en)	Def	Défaut (en)	Po	Pompe
As	Alarme sonore	Di	Disjoncteur	Pt	Pressostat
Au	Arrêt d'urgence	DI	Détecteur de luminosité	Re	Régulateur
Bp	Bouton poussoir	Dp	Détecteur de présence	RI	Relais
Cf	Convertisseur de fréquence	Ed	Electro-distributeur	Sel	Sélecteur (x) positions
Co	Contacteur	Em	Electro-aimant	So	Sonde de mesure
Cp	Coup-de-poing	Gv	Générateur de vide	Th	Thermique
Cpc	Capteur de proximité capacitif	lb	Interrupteur basculant	Tt	Thermostat
Cpi	Capteur de proximité inductif	lp	Interrupteur de position	Va	Vanne
Cpl	Capteur de proximité laser	lr	Interrupteur rotatif	Ve	Vérin pneumatique ou hydraulique
Cpo	Capteur de proximité optique	Jo	Joystick	Vo	Voyant lumineux
Cr	Contact Reed	Mo	Moteur	Vt	Vacuostat

Exemples: Ix_BpStart, Ix_nCpAu, Ix_CpiCube

Correspondance des abréviations des langages utilisant la norme IEC 61131-3 et Siemens

Siemens	IEC 61131-3
CONT	LD (Ladder Diagram)
LOG	FDB (Function Block Diagram)
SCL	ST (Structured Text)
S7-GRAPH	SFC (Sequential Function Chart)
LIST	IL (Instruction List)

V3.3: Remarque: cette version 3.3 a été adaptée pour correspondre au cours CIE Swissmem AUBK3 ainsi qu'à l'examen partiel libéré de triage de cube (AUB351-3).