

# Modicon M262

## Logic/Motion Controller

### Encoder - Guide de la bibliothèque

05/2019



E1000000003676.00

[www.schneider-electric.com](http://www.schneider-electric.com)

**Schneider**  
Electric

---

Le présent document comprend des descriptions générales et/ou des caractéristiques techniques des produits mentionnés. Il ne peut pas être utilisé pour définir ou déterminer l'adéquation ou la fiabilité de ces produits pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur ou intégrateur de réaliser l'analyse de risques complète et appropriée, l'évaluation et le test des produits pour ce qui est de l'application à utiliser et de l'exécution de cette application. Ni la société Schneider Electric ni aucune de ses sociétés affiliées ou filiales ne peuvent être tenues pour responsables de la mauvaise utilisation des informations contenues dans le présent document. Si vous avez des suggestions, des améliorations ou des corrections à apporter à cette publication, veuillez nous en informer.

Vous acceptez de ne pas reproduire, excepté pour votre propre usage à titre non commercial, tout ou partie de ce document et sur quelque support que ce soit sans l'accord écrit de Schneider Electric. Vous acceptez également de ne pas créer de liens hypertextes vers ce document ou son contenu. Schneider Electric ne concède aucun droit ni licence pour l'utilisation personnelle et non commerciale du document ou de son contenu, sinon une licence non exclusive pour une consultation « en l'état », à vos propres risques. Tous les autres droits sont réservés.

Toutes les réglementations locales, régionales et nationales pertinentes doivent être respectées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et afin de garantir la conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque des équipements sont utilisés pour des applications présentant des exigences techniques de sécurité, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou d'un logiciel approuvé avec nos produits matériels peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cette consigne peut entraîner des lésions corporelles ou des dommages matériels.

© 2019 Schneider Electric. Tous droits réservés.

---

# Table des matières

---



	<b>Consignes de sécurité</b> .....	<b>5</b>
	<b>A propos de ce manuel</b> .....	<b>7</b>
<b>Chapitre 1</b>	<b>Principes des modes du codeur</b> .....	<b>11</b>
	Description du principe du mode incrémental .....	<b>12</b>
	Description du principe du mode SSI .....	<b>15</b>
<b>Chapitre 2</b>	<b>Blocs fonction M262 Logic/Motion Controller Encoder</b> ..	<b>17</b>
	FB_Encoder_M262 : activer et surveiller le codeur .....	<b>18</b>
	FB_EncoderPreset_M262 : prédéfinir le codeur .....	<b>21</b>
	FB_EncoderCapture_M262 : capturer la valeur du codeur .....	<b>23</b>
	FB_EncoderReadScalingParam_M262 : lire le paramètre de mise à l'échelle .....	<b>25</b>
<b>Chapitre 3</b>	<b>Types de données de la bibliothèque M262 Logic/Motion Controller</b> .....	<b>27</b>
	ET_ENC_CAP_EDGE_M262 : codes de capture du codeur .....	<b>28</b>
	ET_ENC_ERROR_M262 : codes d'erreur du codeur .....	<b>29</b>
	ET_ENC_INPUT_M262 : codes d'entrée du codeur .....	<b>30</b>
	ET_ENC_PRESET_MODE_M262 : codes de mode prédéfini du codeur .....	<b>31</b>
<b>Annexes</b>	.....	<b>33</b>
<b>Annexe A</b>	<b>Représentation des fonctions et blocs fonction</b> .....	<b>35</b>
	Différences entre une fonction et un bloc fonction .....	<b>36</b>
	Utilisation d'une fonction ou d'un bloc fonction en langage IL .....	<b>37</b>
	Utilisation d'une fonction ou d'un bloc fonction en langage ST .....	<b>42</b>
<b>Glossaire</b>	.....	<b>45</b>
<b>Index</b>	.....	<b>47</b>



# Consignes de sécurité



## Informations importantes

### AVIS

Lisez attentivement ces instructions et examinez le matériel pour vous familiariser avec l'appareil avant de tenter de l'installer, de le faire fonctionner, de le réparer ou d'assurer sa maintenance. Les messages spéciaux suivants que vous trouverez dans cette documentation ou sur l'appareil ont pour but de vous mettre en garde contre des risques potentiels ou d'attirer votre attention sur des informations qui clarifient ou simplifient une procédure.



La présence de ce symbole sur une étiquette "Danger" ou "Avertissement" signale un risque d'électrocution qui provoquera des blessures physiques en cas de non-respect des consignes de sécurité.



Ce symbole est le symbole d'alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

## DANGER

**DANGER** signale un risque qui, en cas de non-respect des consignes de sécurité, **provoque** la mort ou des blessures graves.

## AVERTISSEMENT

**AVERTISSEMENT** signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** la mort ou des blessures graves.

## ATTENTION

**ATTENTION** signale un risque qui, en cas de non-respect des consignes de sécurité, **peut provoquer** des blessures légères ou moyennement graves.

## AVIS

**AVIS** indique des pratiques n'entraînant pas de risques corporels.

---

## REMARQUE IMPORTANTE

L'installation, l'utilisation, la réparation et la maintenance des équipements électriques doivent être assurées par du personnel qualifié uniquement. Schneider Electric décline toute responsabilité quant aux conséquences de l'utilisation de ce matériel.

Une personne qualifiée est une personne disposant de compétences et de connaissances dans le domaine de la construction, du fonctionnement et de l'installation des équipements électriques, et ayant suivi une formation en sécurité leur permettant d'identifier et d'éviter les risques encourus.

# A propos de ce manuel



## Présentation

### Objectif du document

Ce document est destiné à vous familiariser avec les variables et les fonctions de codage proposées par M262 Logic/Motion Controller. La bibliothèque M262 Logic/Motion Controller Encoder contient des fonctions et des variables permettant d'obtenir des informations auprès du système codeur et d'envoyer des commandes vers celui-ci.

Ce document décrit les variables et les fonctions de type de données de la bibliothèque M262 Logic/Motion Controller Encoder.

Il requiert les connaissances préalables suivantes :

- Connaissances de base sur les fonctionnalités, la structure et la configuration du M262 Logic/Motion Controller.
- Programmation en langage FBD, LD, ST, IL ou CFC.
- Variables système (variables globales).

### Champ d'application

Ce document a été actualisé pour le lancement d'EcoStruxure™ Machine Expert V1.1.


### Document(s) à consulter

Titre de documentation	Référence
EcoStruxure Machine Expert - Guide de programmation	<a href="#">EIO0000002854 (ENG)</a> ; <a href="#">EIO0000002855 (FRE)</a> ; <a href="#">EIO0000002856 (GER)</a> ; <a href="#">EIO0000002858 (SPA)</a> ; <a href="#">EIO0000002857 (ITA)</a> ; <a href="#">EIO0000002859 (CHS)</a>
Modicon M262 Logic/Motion Controller - Guide de référence du matériel	<a href="#">EIO0000003659 (ENG)</a> ; <a href="#">EIO0000003660 (FRE)</a> ; <a href="#">EIO0000003661 (GER)</a> ; <a href="#">EIO0000003662 (SPA)</a> ; <a href="#">EIO0000003663 (ITA)</a> ; <a href="#">EIO0000003664 (CHS)</a> ; <a href="#">EIO0000003665 (POR)</a> ; <a href="#">EIO0000003666 (TUR)</a>

Titre de documentation	Référence
Modicon M262 Logic/Motion Controller - Guide de programmation	<a href="#">EIO0000003651 (ENG)</a> ; <a href="#">EIO0000003652 (FRE)</a> ; <a href="#">EIO0000003653 (GER)</a> ; <a href="#">EIO0000003654 (SPA)</a> ; <a href="#">EIO0000003655 (ITA)</a> ; <a href="#">EIO0000003656 (CHS)</a> ; <a href="#">EIO0000003657 (POR)</a> ; <a href="#">EIO0000003658 (TUR)</a>

Vous pouvez télécharger ces publications et autres informations techniques depuis notre site web à l'adresse : <https://www.schneider-electric.com/en/download>

### Information spécifique au produit

 <b>AVERTISSEMENT</b>
<p><b>PERTE DE CONTROLE</b></p> <ul style="list-style-type: none"> <li>● Le concepteur d'un système de commande doit envisager les modes de défaillance possibles des chemins de commande et, pour certaines fonctions de commande critiques, prévoir un moyen d'atteindre un état sécurisé en cas de défaillance d'un chemin, et après cette défaillance. Par exemple, l'arrêt d'urgence, l'arrêt en cas de surcourse, la coupure de courant et le redémarrage sont des fonctions de contrôle cruciales.</li> <li>● Des canaux de commande séparés ou redondants doivent être prévus pour les fonctions de commande critique.</li> <li>● Les chemins de commande système peuvent inclure les liaisons de communication. Soyez particulièrement attentif aux implications des retards de transmission imprévus ou des pannes de liaison.</li> <li>● Respectez toutes les réglementations de prévention des accidents ainsi que les consignes de sécurité locales.<sup>1</sup></li> <li>● Chaque implémentation de cet équipement doit être testée individuellement et entièrement pour s'assurer du fonctionnement correct avant la mise en service.</li> </ul> <p><b>Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.</b></p>

<sup>1</sup> Pour plus d'informations, consultez le document NEMA ICS 1.1 (dernière édition), « Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control » (Directives de sécurité pour l'application, l'installation et la maintenance de commande statique) et le document NEMA ICS 7.1 (dernière édition), « Safety Standards for Construction and Guide for Selection, Installation, and Operation of Adjustable-Speed Drive Systems » (Normes de sécurité relatives à la construction et manuel de sélection, installation et opération de variateurs de vitesse) ou son équivalent en vigueur dans votre pays.



## AVERTISSEMENT

### FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

- N'utiliser que le logiciel approuvé par Schneider Electric pour faire fonctionner cet équipement.
- Mettez à jour votre programme d'application chaque fois que vous modifiez la configuration matérielle physique.

**Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.**

### Terminologie utilisée dans les normes

Les termes techniques, la terminologie, les symboles et les descriptions correspondantes employés dans ce manuel ou figurant dans ou sur les produits proviennent généralement des normes internationales.

Dans les domaines des systèmes de sécurité fonctionnelle, des variateurs et de l'automatisme en général, les termes employés sont *sécurité, fonction de sécurité, état sécurisé, défaut, réinitialisation du défaut, dysfonctionnement, panne, erreur, message d'erreur, dangereux*, etc.

Entre autres, les normes concernées sont les suivantes :

Norme	Description
IEC 61131-2:2007	Automates programmables - Partie 2 : exigences et essais des équipements
ISO 13849-1:2015	Sécurité des machines : parties des systèmes de commande relatives à la sécurité. Principes généraux de conception
EN 61496-1:2013	Sécurité des machines : équipements de protection électro-sensibles. Partie 1 : Prescriptions générales et essais
ISO 12100:2010	Sécurité des machines - Principes généraux de conception - Appréciation du risque et réduction du risque
EN 60204-1:2006	Sécurité des machines - Équipement électrique des machines - Partie 1 : règles générales
ISO 14119:2013	Sécurité des machines - Dispositifs de verrouillage associés à des protecteurs - Principes de conception et de choix
ISO 13850:2015	Sécurité des machines - Fonction d'arrêt d'urgence - Principes de conception
IEC 62061:2015	Sécurité des machines - Sécurité fonctionnelle des systèmes de commande électrique, électronique et électronique programmable relatifs à la sécurité
IEC 61508-1:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : prescriptions générales.
IEC 61508-2:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : exigences pour les systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité.

Norme	Description
IEC 61508-3:2010	Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité : exigences concernant les logiciels.
IEC 61784-3:2016	Réseaux de communication industriels - Profils - Partie 3 : Bus de terrain de sécurité fonctionnelle - Règles générales et définitions de profils.
2006/42/EC	Directive Machines
2014/30/EU	Directive sur la compatibilité électromagnétique
2014/35/EU	Directive sur les basses tensions

De plus, des termes peuvent être utilisés dans le présent document car ils proviennent d'autres normes telles que :

Norme	Description
Série IEC 60034	Machines électriques rotatives
Série IEC 61800	Entraînements électriques de puissance à vitesse variable
Série IEC 61158	Communications numériques pour les systèmes de mesure et de commande – Bus de terrain utilisés dans les systèmes de commande industriels

Enfin, le terme *zone de fonctionnement* utilisé dans le contexte de la description de dangers spécifiques a la même signification que les termes *zone dangereuse* ou *zone de danger* employés dans la *directive Machines (2006/42/EC)* et la norme *ISO 12100:2010*.

**NOTE :** Les normes susmentionnées peuvent s'appliquer ou pas aux produits cités dans la présente documentation. Pour plus d'informations sur chacune des normes applicables aux produits décrits dans le présent document, consultez les tableaux de caractéristiques de ces références de produit.

---

# Chapitre 1

## Principes des modes du codeur

---

### Présentation

Ce chapitre explique comment utiliser un codeur en mode incrémental ou en mode SSI (Synchronous Serial Interface).

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Description du principe du mode incrémental	12
Description du principe du mode SSI	15

## Description du principe du mode incrémental

### Présentation

Cette section décrit l'utilisation du mode incrémental pour se connecter à des codeurs incrémentaux.

### Principe

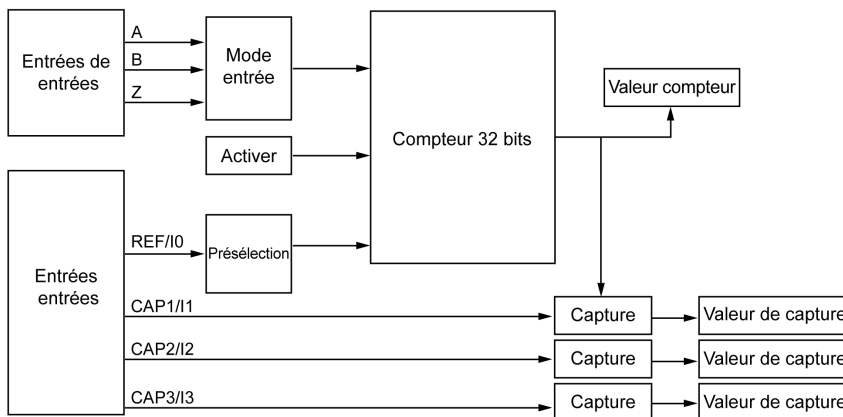
Le mode incrémental se comporte comme un compteur croissant/décroissant standard, utilisant des impulsions et comptant ces mêmes impulsions.

Les positions doivent être prédéfinies et le comptage initialisé pour implémenter et gérer le mode incrémental.

La valeur du compteur peut être stockée dans le registre de capture en configurant un événement externe.

### Schéma du principe

Le schéma suivant présente le codeur en mode incrémental :



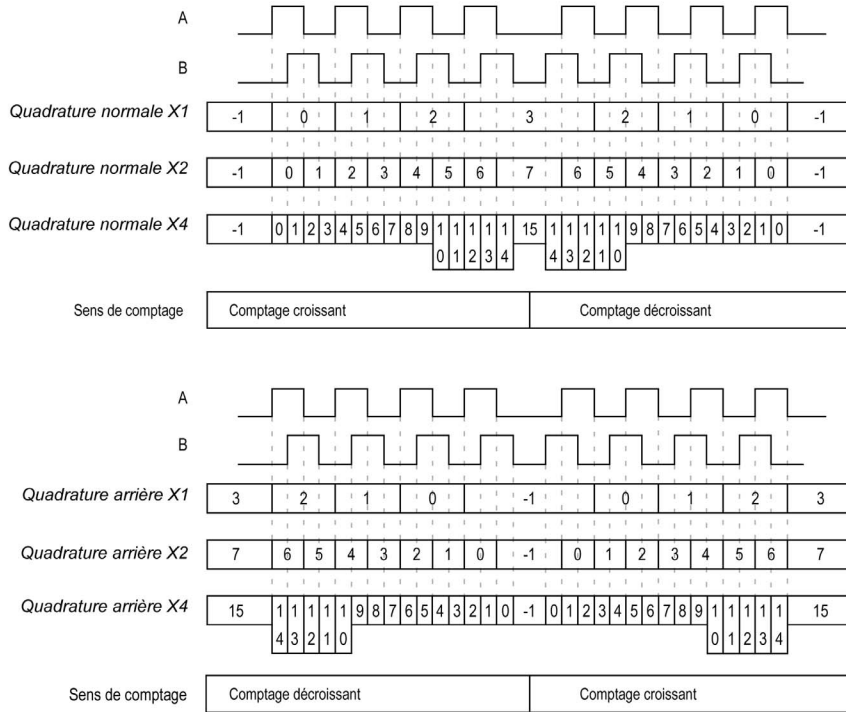
### Types d'axe

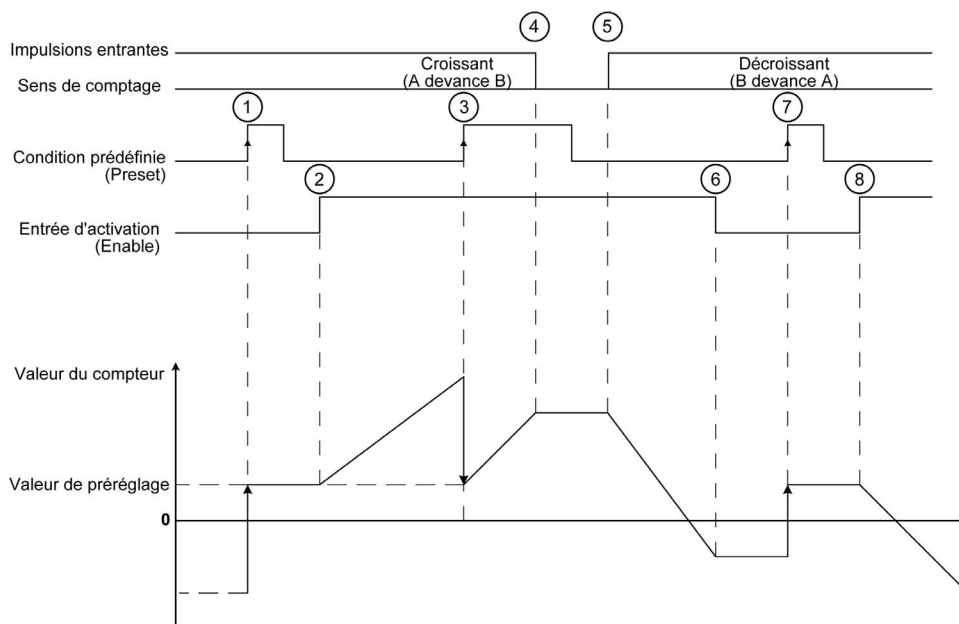
Le tableau suivant présente les deux types d'axe disponibles et les modes de comptage correspondants :

Type d'axe	Commentaire
Linéaire	Dans ce mode, le compteur se comporte comme un compteur fini.
Rotatif	Dans ce mode, le compteur se comporte comme un compteur infini.

## Schéma du principe

Le mode d'entrée en mode incrémental est toujours une quadrature :





Etape	Action
1	Sur le front montant de la condition Preset, la valeur du compteur est définie sur la valeur prédéfinie et le compteur est activé.
2	Lorsque la condition Enable a pour valeur 1, le compteur commence à s'incrémenter si le comptage est croissant.
3	Le front montant de la condition Preset charge la valeur <b>prédéfinie</b> .
4	Lorsque les impulsions entrantes cessent, le compteur conserve sa valeur.
5	Lorsque la condition Enable a pour valeur 1, le compteur commence à décrémenter si le comptage est décroissant.
6	Lorsque la condition Enable a pour valeur 0, le compteur ignore les impulsions appliquées aux entrées de comptage A et B.
7	Le front montant de la condition Preset charge la valeur prédéfinie.
8	Lorsque la condition Enable a pour valeur 1, le compteur commence à décrémenter si le comptage est décroissant.

**NOTE :** Les conditions Enable et Preset dépendent de la configuration. Elles sont décrites dans les fonctions Enable (*voir page 18*) et Preset (*voir page 21*).

## Description du principe du mode SSI

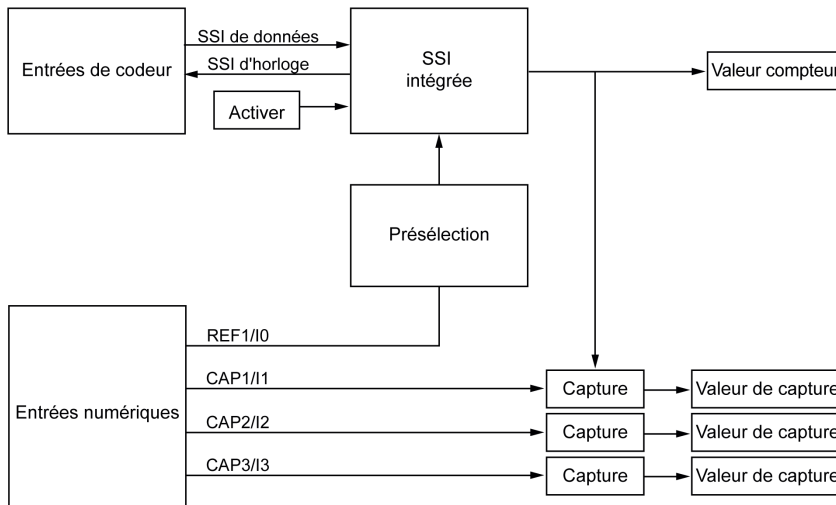
### Général

Le mode SSI (Synchronous Serial Interface) permet de connecter un codeur absolu.

La position du codeur absolu est lue via une liaison SSI.

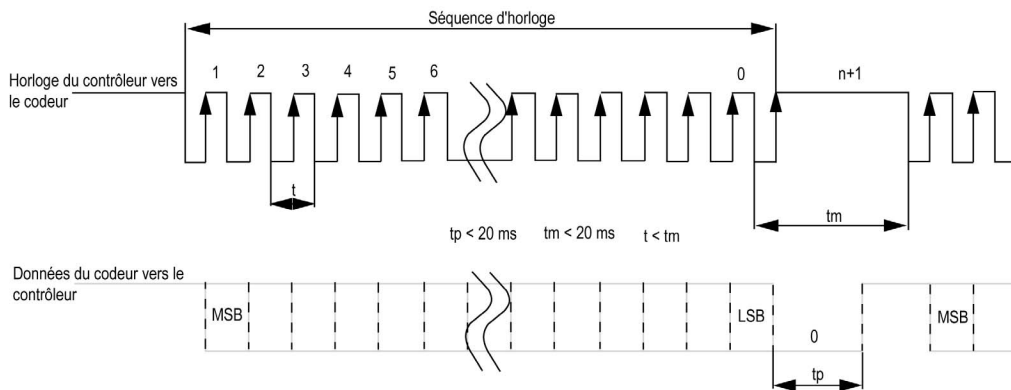
### Schéma du principe

Le schéma suivant présente le codeur en mode SSI :



## Schéma du principe

La figure suivante représente une trame SSI :



## Informations sur les données

Les données peuvent être configurées pour ajuster les informations du codeur absolu :

Paramètre	Plage	Commentaire
Vitesse de transmission	100 kHz ou 250 kHz ou 500 kHz	–
Nombre de bits par trame	8 à 64 bits	Longueur de trame = nombre implicite de bits d'en-tête (0 à 4) + nombre de bits de données (8 à 32) + nombre de bits d'état (0 to 4) + nombre de bit de parité (0 ou 1).
Nombre de bits de données	8 à 32 bits	Les bits les moins significatifs (8 à 32) indiquent la résolution par tour et les bits les plus significatifs (0 à 24) indiquent le nombre de tours.
Nombre de bits de données/tour	8 à 16 bits	–
Nombre de bits d'état	0 à 4 bits	–
Parité	Aucun Impaire Paire	–
Réduction de la résolution	0 à 17 bits	Ce paramètre permet de filtrer les données. Les bits les moins significatifs sont ignorés.
Codage binaire	Binaire Gris	Code binaire ou gris.



---

# Chapitre 2

## Blocs fonction M262 Logic/Motion Controller Encoder

---

### Présentation

Ce chapitre décrit les blocs fonction inclus dans la bibliothèque M262 Encoder. L'ajout d'un codeur a pour effet d'ajouter automatiquement la bibliothèque Encoder à votre contrôleur.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
FB_Encoder_M262 : activer et surveiller le codeur	18
FB_EncoderPreset_M262 : prédéfinir le codeur	21
FB_EncoderCapture_M262 : capturer la valeur du codeur	23
FB_EncoderReadScalingParam_M262 : lire le paramètre de mise à l'échelle	25

## FB\_Encoder\_M262 : activer et surveiller le codeur

### Description du bloc fonction

Ce bloc fonction permet d'activer et de surveiller le codeur, en mode incrémental ou SSI.  
 Vous ne pouvez utiliser qu'une seule instance de ce bloc fonction qui est appelé une fois.  
 Utilisez des appels cycliques pour actualiser les valeurs.

### Représentation graphique



### Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction (voir page 35)*.

## Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Par défaut	Commentaires
ENC_REF_M262	ENC_REF_M262	–	Référence de l'instance du codeur.
xEnable	BOOL	FALSE	<p>TRUE active le bloc fonction.</p> <p>Sur un front montant, les valeurs des paramètres de mise à l'échelle suivants sont prises en compte :</p> <ul style="list-style-type: none"> <li>• udiScaling_NbOfIncs</li> <li>• udiScaling_NbOfUnits</li> <li>• udiScaling_IncPerTurn</li> </ul> <p>Si vous modifiez ces valeurs, déclenchez un front montant sur xEnable pour les prendre en compte.</p>
udiScaling_NbOfIncs	UDINT	0	<p>0 indique que la mise à l'échelle est désactivée. La valeur en unités utilisateur diCurrentValue_Unit est égale à la valeur en impulsions diCurrentValue.</p> <p>&gt; 0 indique que la mise à l'échelle est activée. La valeur en unités utilisateur est calculée à partir de la valeur en impulsions diCurrentValue, comme ceci : <math>diCurrentValue\_Unit = diCurrentValue \times (udiScalingNbOfUnits / udiScalingNbOfIncs)</math>.</p>
udiScaling_NbOfUnits	UDINT	0	<p>0 indique que la mise à l'échelle est désactivée. La valeur en unités utilisateur diCurrentValue_Unit est égale à la valeur en impulsions diCurrentValue. En l'absence de mise à l'échelle, <math>udiScalingNbOfUnits = udiScalingNbOfIncs</math></p> <p>&gt; 0 indique que la mise à l'échelle est activée. La valeur en unités utilisateur est calculée à partir de la valeur en impulsions diCurrentValue, comme ceci : <math>diCurrentValue\_Unit = diCurrentValue \times (udiScalingNbOfUnits / udiScalingNbOfIncs)</math>.</p>
udiScaling_IncPerTurn	UDINT	0	<p>Quand cette valeur est égale à 0, le type d'axe est associé au mode compteur linéaire. La plage de comptage est la suivante : - 2 147 483 648 à 2 147 483 647.</p> <p>Si le nombre d'incrément est &gt; 0, le type d'axe est associé au mode compteur rotatif. La valeur udiSaling_IncPerTurn définit la valeur modulo, à laquelle le compteur bascule (la valeur modulo n'est jamais atteinte). La plage de comptage est la suivante : 0 à diScaling_IncPerTurn -1</p>

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Par défaut	Commentaire
xValid	BOOL	FALSE	TRUE indique que les valeurs de sortie du bloc fonction sont valides. Si le bloc fonction est désactivé, la sortie a la valeur FALSE .
xError	BOOL	FALSE	TRUE indique qu'une erreur est détectée.
etErrorId	ET_ENC_ERROR_M262	ENC_ERROR_NO	Indique le code de l'erreur détectée lorsque xError a la valeur TRUE.
diNbTurns	DINT	0	Indique la valeur modulo du codeur. En mode incrémental, elle est incrémentée lorsque le compteur dépasse sa limite supérieure. Elle est décrétementée lorsque le compteur dépasse sa limite inférieure. En mode SSI $diNbTurns = \text{valeur brute (valeur SSI - valeur prédéfinie) / udiScaling\_IncPerTurn}$ La valeur raw SSI est directement issue de SSI, sans aucune transformation.
diCurrentValue	DINT	0	En mode linéaire, indique la valeur de la position de l'équipement en impulsions. La plage de valeurs est : - 2 147 483 648 à 2 147 483 647. En mode rotatif, indique la valeur de la position en impulsions pour chaque tour de la mécanique. La plage de valeurs pour diCurrentValue est 0 à $diScaling\_IncPerTurn - 1$ .
lrCurrentValue_Unit	LREAL	0	Indique la valeur du codeur en unités de tours de la mécanique. $diCurrentValue\_Unit = diCurrentValue / udiScaling\_IncPerUnit$ lorsque $udiScaling\_IncPerUnit \geq 1$ .

## FB\_EncoderPreset\_M262 : prédéfinir le codeur

### Description du bloc fonction

Ce bloc fonction permet de prédéfinir le codeur, en mode incrémental ou SSI.

### Représentation graphique



### Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction (voir page 35)*.

### Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Par défaut	Commentaires
ENC_REF_M262	ENC_REF_M262	–	Référence de l'instance du codeur.
xEnable	BOOL	FALSE	TRUE active la fonction prédéfinie du codeur, via : <ul style="list-style-type: none"> <li>Le mode prédéfini en utilisant REF sur I0 et Z sur le codeur</li> <li>L'entrée xForce du bloc fonction</li> </ul>
xForce	BOOL	FALSE	Sur le front montant, prédéfini et démarre le compteur si xEnable a la valeur TRUE.
etREF_Input	ET_ENC_INPUT_M262	ENC_INPUT_REF_I0	Définit l'entrée REF. La seule valeur valide est I0 (voir page 30).

Entrée	Type	Par défaut	Commentaires
etMode	ET_ENC_PRESET_MODE_M262	ENC_PRESET_NO	Sélectionne les conditions pour prédéfinir la fonction de comptage avec les entrées <i>(voir page 31)</i> REF et Z.
diPresetValue	DINT	0	Définit la valeur chargée dans le codeur comme valeur réelle lors de l'événement prédéfini.

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Par défaut	Commentaire
xValid	BOOL	FALSE	TRUE indique que les valeurs de sortie du bloc fonction sont valides.
xError	BOOL	FALSE	TRUE indique qu'une erreur est détectée.
etErrorId	ET_ENC_ERROR_M262	ENC_ERROR_NO	Indique le code de l'erreur détectée lorsque xError a la valeur TRUE <i>(voir page 29)</i> .
xPresetFlag	BOOL	FALSE	Défini sur TRUE pour un cycle par la prédéfinition du codeur.

## FB\_EncoderCapture\_M262 : capturer la valeur du codeur

### Description du bloc fonction

Ce bloc fonction permet de capturer la valeur du codeur, en mode incrémental ou SSI.

Pour configurer plusieurs instances de ce bloc fonction, définissez une valeur `etCAP_Input` différente.

### Représentation graphique



### Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction* ([voir page 35](#)).

### Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Par défaut	Commentaire
ENC_REF_M262	ENC_REF_M262	–	Référence de l'instance du codeur.
xEnable	BOOL	FALSE	TRUE active la fonction de capture du codeur, via l'entrée de capture spécifiée par l'entrée <code>etCAP_Input</code> .
etCAP_Input	ET_ENC_INPUT_M262	ENC_INPUT_CAP_I1	Définit l'entrée utilisée pour la fonction de capture ( <a href="#">voir page 30</a> ).
etCAP_Edge	ET_ENC_CAP_EDGE_M262	ENC_CAP_EDGE_RISING	Indique la détection de front pour l'entrée de capture ( <a href="#">voir page 28</a> ).

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Par défaut	Commentaire
xValid	BOOL	FALSE	TRUE indique que les valeurs de sortie du bloc fonction sont valides.
xError	BOOL	FALSE	TRUE indique qu'une erreur est détectée.
etErrorId	ET_ENC_ERROR_M262	ENC_ERROR_NO	Indique le code de l'erreur détectée lorsque xError a la valeur TRUE (voir page 29).
xCaptureFlag	BOOL	FALSE	TRUE indique qu'un cycle est défini par l'événement de capture du codeur. xCaptureFlag n'a donc la valeur TRUE que pour un seul cycle.
diCapturedValue	DINT	0	Indique la valeur capturée en impulsions, valide au niveau du front montant xCaptureFlag. La valeur capturée est conservée jusqu'à la prochaine occurrence de xCaptureFlag. La valeur capturée est réinitialisée à 0 lorsque xEnable a la valeur FALSE.
lrCapturedValue_Units	LREAL	0.0	Indique la valeur capturée en unités, valide au niveau du front montant xCaptureFlag. La valeur capturée est conservée jusqu'à la prochaine occurrence de xCaptureFlag. La valeur capturée est réinitialisée à 0 lorsque xEnable a la valeur FALSE.



## FB\_EncoderReadScalingParam\_M262 : lire le paramètre de mise à l'échelle

### Description du bloc fonction

Ce bloc fonction permet de lire les valeurs actives du paramètre de mise à l'échelle servant à calculer la valeur unitaire, en mode incrémental ou SSI.

### Représentation graphique



### Représentation en langage IL et ST

Pour voir la représentation générale en langage IL ou ST, consultez le chapitre *Représentation des fonctions et blocs fonction* (voir page 35).

### Description des variables d'E/S

Le tableau suivant décrit les variables d'entrée :

Entrée	Type	Par défaut	Commentaire
ENC_REF_M262	ENC_REF_M262	-	Référence de l'instance du codeur.
xEnable	BOOL	FALSE	TRUE permet au bloc fonction du codeur de lire les valeurs actives du paramètres de mise à l'échelle servant à calculer lrCurrentValue_Unit FALSE désactive le bloc fonction.

Le tableau suivant décrit les variables de sortie :

Sortie	Type	Par défaut	Commentaire
xValid	BOOL	FALSE	TRUE indique que les valeurs de sortie du bloc fonction sont valides.
xError	BOOL	FALSE	TRUE indique qu'une erreur est détectée.
udiScalingNbOfIncs	UDINT	0	Indique la valeur active de udiScalingNbOfIncs pour calculer lrCurrentValue_Unit.
udiScalingNbOfUnits	UDINT	0	Indique la valeur active de udiScalingNbOfUnits pour calculer lrCurrentValue_Unit.
udiScaling_IncPerTurn	UDINT	0	Indique la valeur active de udiScaling_IncPerTurn pour calculer lrCurrentValue_Unit.

---

# Chapitre 3

## Types de données de la bibliothèque M262 Logic/Motion Controller

---

### Présentation

Ce chapitre décrit les types de données de la bibliothèque Encoder M262.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
ET_ENC_CAP_EDGE_M262 : codes de capture du codeur	28
ET_ENC_ERROR_M262 : codes d'erreur du codeur	29
ET_ENC_INPUT_M262 : codes d'entrée du codeur	30
ET_ENC_PRESET_MODE_M262 : codes de mode prédéfini du codeur	31

## ET\_ENC\_CAP\_EDGE\_M262 : codes de capture du codeur

### Description du type énuméré

Cette énumération décrit les types de fronts qui peuvent être utilisés à titre de référence et de capture sur un bloc fonction du codeur.

Le type de données d'énumération ET\_ENC\_CAP\_EDGE\_M262 contient les valeurs suivantes :

Nom du paramètre	Valeur	Description
ENC_CAP_EDGE_RISING	0	Capture sur le front montant en entrée.
ENC_CAP_EDGE_FALLING	1	Capture sur le front descendant en entrée.
ENC_CAP_EDGE_BOTH	2	Capture sur les deux fronts en entrée.

**ET\_ENC\_ERROR\_M262 : codes d'erreur du codeur****Description du type énuméré**

Cette énumération décrit les types d'erreurs qui peuvent se produire sur un bloc fonction du codeur.

Le type de données d'énumération ET\_ENC\_ERROR\_M262 contient les valeurs suivantes :

Nom du paramètre	Valeur	Description
ENC_ERROR_NO	0	Aucune erreur détectée.
ENC_ERROR_REF	1	La référence du codeur est incorrecte ou non configurée.
ENC_ERROR_PARAMETER_INVALID	3	La valeur d'un paramètre est incorrecte.
ENC_ERROR_COM	4	Une erreur de communication est détectée au niveau du codeur.
ENC_ERROR_SUPPLY	11	Alimentation du codeur non détectée.
ENC_ERROR_IO_EVT_CONFIGURED	12	IO est configuré en tant qu'événement et ne peut pas être utilisé pour la prédéfinition.
ENC_ERROR_RESERVED	13	Le bloc fonction FB_Encoder_M262 est réservé.

## ET\_ENC\_INPUT\_M262 : codes d'entrée du codeur

### Description du type énuméré

Cette énumération décrit les types d'entrées qui peuvent être utilisées à titre de référence et de capture sur un bloc fonction du codeur.

Le type de données d'énumération ET\_ENC\_INPUT\_M262 contient les valeurs suivantes :

Nom du paramètre	Valeur	Description
ENC_INPUT_REF_I0	0	Entrée REF sur I0 pour la prédéfinition.
ENC_INPUT_CAP_I1	1	Entrée de capture sur I1.
ENC_INPUT_CAP_I2	2	Entrée de capture sur I2.
ENC_INPUT_CAP_I3	3	Entrée de capture sur I3.

**ET\_ENC\_PRESET\_MODE\_M262 : codes de mode prédéfini du codeur****Description du type énuméré**

Cette énumération décrit les différents types de mode prédéfini qui peuvent être utilisés pour un bloc fonction du codeur.

Le type de données d'énumération ET\_ENC\_PRESET\_MODE\_M262 contient les valeurs suivantes :

Nom du paramètre	Valeur	Description
ENC_PRESET_NO	0	Aucune prédéfinition configurée.
ENC_PRESET_Z_EDGE_RISING	1	Prédéfinition sur le front montant Z (codeur incrémental uniquement).
ENC_PRESET_Z_EDGE_FALLING	2	Prédéfinition sur le front descendant Z (codeur incrémental uniquement).
ENC_PRESET_Z_EDGE_BOTH	3	Prédéfinition sur les deux fronts Z (codeur incrémental uniquement)
ENC_PRESET_REF_RISING	4	Prédéfinition sur le front montant REF.
ENC_PRESET_REF_FALLING	5	Prédéfinition sur le front descendant REF.
ENC_PRESET_REF_BOTH	6	Prédéfinition sur les deux fronts REF.
ENC_PRESET_Z_EDGE_RISING_AND_REF	7	Prédéfinition sur le front montant Z et REF (codeur incrémental uniquement).
ENC_PRESET_EDGE_RISING_Z_FIRST_AND_REF	8	Prédéfinition sur le premier front montant Z et REF (codeur incrémental uniquement).
ENC_PRESET_EDGE_RISING_Z_FIRST_AND_NO_REF	9	Prédéfinition sur le premier front montant Z et pas de REF (codeur incrémental uniquement).





---

# Annexes

---





---

# Annexe A

## Représentation des fonctions et blocs fonction

---

### Présentation

Chaque fonction peut être représentée dans les langages suivants :

- IL : (Instruction List) liste d'instructions
- ST : (Structured Text) littéral structuré
- LD : (Ladder Diagram) schéma à contacts
- FBD : Function Block Diagram (Langage à blocs fonction)
- CFC : Continuous Function Chart (Diagramme fonctionnel continu)

Ce chapitre fournit des exemples de représentations de fonctions et blocs fonction et explique comment les utiliser dans les langages IL et ST.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Différences entre une fonction et un bloc fonction	36
Utilisation d'une fonction ou d'un bloc fonction en langage IL	37
Utilisation d'une fonction ou d'un bloc fonction en langage ST	42

## Différences entre une fonction et un bloc fonction

### Fonction

Une fonction :

- est une POU (Program Organization Unit ou unité organisationnelle de programme) qui renvoie un résultat immédiat ;
- est directement appelée par son nom (et non par une instance) ;
- ne conserve pas son état entre deux appels ;
- peut être utilisée en tant qu'opérande dans des expressions.

**Exemples** : opérateurs booléens (AND), calculs, conversions (BYTE\_TO\_INT)

### Bloc fonction

Un bloc fonction :

- est une POU qui renvoie une ou plusieurs sorties ;
- doit être appelé par une instance (copie de bloc fonction avec nom et variables dédiées).
- Chaque instance conserve son état (sorties et variables internes) entre deux appels à partir d'un bloc fonction ou d'un programme.

**Exemples** : temporisateurs, compteurs

Dans l'exemple, `Timer_ON` est une instance du bloc fonction `TON` :

```
1  PROGRAM MyProgram_ST
2  VAR
3      Timer_ON: TON; // Function Block Instance
4      Timer_RunCd: BOOL;
5      Timer_PresetValue: TIME := T#5S;
6      Timer_Output: BOOL;
7      Timer_ElapsedTime: TIME;
8  END_VAR

1  Timer_ON(
2      IN:=Timer_RunCd,
3      PT:=Timer_PresetValue,
4      Q=>Timer_Output,
5      ET=>Timer_ElapsedTime);
```

## Utilisation d'une fonction ou d'un bloc fonction en langage IL

### Informations générales

Cette partie explique comment mettre en œuvre une fonction et un bloc fonction en langage IL.

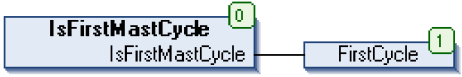

Les fonctions `IsFirstMastCycle` et `SetRTCDrift`, ainsi que le bloc fonction `TON`, sont utilisés à titre d'exemple pour illustrer les mises en œuvre.

### Utilisation d'une fonction en langage IL

La procédure suivante explique comment insérer une fonction en langage IL :

Étape	Action
1	Ouvrez ou créez une POU en langage IL (Instruction List, ou liste d'instructions). <b>NOTE</b> : La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, reportez-vous à la section Ajout et appel de POU.
2	Créez les variables nécessaires à la fonction.
3	Si la fonction possède une ou plusieurs entrées, chargez la première entrée en utilisant l'instruction LD.
4	Insérez une nouvelle ligne en dessous et : <ul style="list-style-type: none"> <li>● saisissez le nom de la fonction dans la colonne de l'opérateur (champ de gauche), ou</li> <li>● utilisez l'<b>Aide à la saisie</b> pour sélectionner la fonction (sélectionnez <b>Insérer l'appel de module</b> dans le menu contextuel).</li> </ul>
5	Si la fonction a plus d'une entrée et que l'assistant Aide à la saisie est utilisé, le nombre requis de lignes est automatiquement créé avec ??? dans les champs situés à droite. Remplacez les ??? par la valeur ou la variable appropriée compte tenu de l'ordre des entrées.
6	insérez une nouvelle ligne pour stocker le résultat de la fonction dans la variable appropriée : saisissez l'instruction ST dans la colonne de l'opérateur (champ de gauche) et le nom de la variable dans le champ de droite.

Pour illustrer la procédure, utilisons les fonctions `IsFirstMastCycle` (sans paramètre d'entrée) et `SetRTCDrift` (avec paramètres d'entrée) représentées graphiquement ci-après :

Fonction	Représentation graphique
sans paramètre d'entrée : <code>IsFirstMastCycle</code>	
avec paramètres d'entrée : <code>SetRTCDrift</code>	

En langage IL, le nom de la fonction est utilisé directement dans la colonne de l'opérateur :


Fonction	Représentation dans l'éditeur IL de POU
<p>Exemple en IL d'une fonction sans paramètre d'entrée : IsFirstMastCycle</p>	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      FirstCycle: BOOL; 4  END_VAR </pre> <hr/> <pre> 1  IsFirstMastCycle    ST          FirstCycle </pre>
<p>Exemple IL d'une fonction avec des paramètres d'entrée : SetRTCDrift</p>	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3      myDrift: SINT (-29..29) := 5; 4      myDay: DAY_OF_WEEK := SUNDAY; 5      myHour: HOUR := 12; 6      myMinute: MINUTE; 7      myDiag: RTCSETDRIFT_ERROR; 8  END_VAR </pre> <hr/> <pre> 1  LD          myDrift    SetRTCDrift myDay            myHour            myMinute    ST          myDiag </pre>

### Utilisation d'un bloc fonction en langage IL

La procédure suivante explique comment insérer un bloc fonction en langage IL :

Étape	Action
1	Ouvrez ou créez une POU en langage IL (Instruction List, ou liste d'instructions). <b>NOTE</b> : La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, reportez-vous à la section Ajout et appel de POU.
2	Créez les variables nécessaires au bloc fonction (y compris le nom de l'instance).
3	L'appel de blocs fonction nécessite l'utilisation d'une instruction <b>CAL</b> : <ul style="list-style-type: none"> <li>● Utilisez l'<b>Aide à la saisie</b> pour sélectionner le bloc fonction (cliquez avec le bouton droit et sélectionnez <b>Insérer l'appel de module</b> dans le menu contextuel).</li> <li>● L'instruction <b>CAL</b> et les E/S nécessaires sont automatiquement créées.</li> </ul> Chaque paramètre (E/S) est une instruction : <ul style="list-style-type: none"> <li>● Les valeurs des entrées sont définies à l'aide de « := ».</li> <li>● Les valeurs des sorties sont définies à l'aide de =&gt;.</li> </ul>
4	Dans le champ <b>CAL</b> de droite, remplacez les ??? par le nom de l'instance.
5	Remplacez les autres ??? par une variable ou une valeur immédiate appropriée.

Pour illustrer la procédure, utilisons le bloc fonction **TON** représenté graphiquement ci-après :

Bloc fonction	Représentation graphique
TON	



En langage IL, le nom du bloc fonction est utilisé directement dans la colonne de l'opérateur :

Bloc fonction	Représentation dans l'éditeur IL de POU
TON	<pre> 1  PROGRAM MyProgram_IL 2  VAR 3  Timer_ON: TON; // Function Block instance declaration 4  Timer_RunCd: BOOL; 5  Timer_PresetValue: TIME := T#5S; 6  Timer_Output: BOOL; 7  Timer_ElapsedTime: TIME; 8  END_VAR 9 </pre> <hr/> <pre> 1  CAL      Timer_ON(            IN:= Timer_RunCd,            PT:= Timer_PresetValue,            Q=&gt; Timer_Output,            ET=&gt; Timer_ElapsedTime) </pre>

## Utilisation d'une fonction ou d'un bloc fonction en langage ST

### Informations générales

Cette partie décrit comment mettre en œuvre une fonction ou un bloc fonction en langage ST.

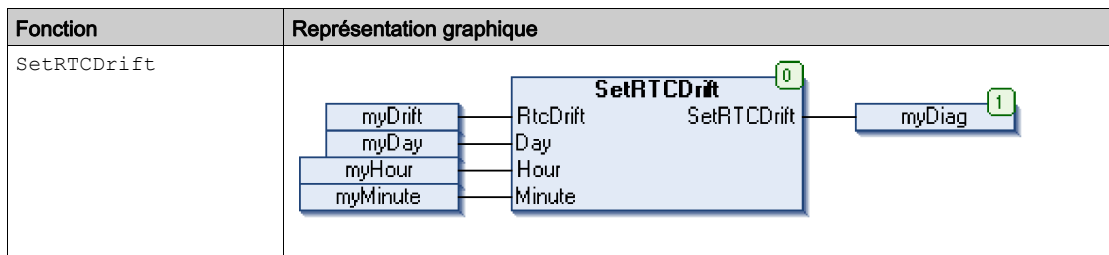
La fonction `SetRTCDrift` et le bloc fonction `TON` sont utilisés à titre d'exemple pour illustrer les mises en œuvre.

### Utilisation d'une fonction en langage ST

La procédure suivante explique comment insérer une fonction en langage ST :

Étape	Action
1	Ouvrez ou créez une POU en langage ST (Structured Text ou Littéral structuré). <b>NOTE</b> : La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations, reportez-vous à la section Ajout et appel de POU.
2	Créez les variables nécessaires à la fonction.
3	Utilisez la syntaxe générale dans l' <b>éditeur ST de POU</b> pour la représentation en langage ST d'une fonction. La syntaxe générale est la suivante : RésultatFonction:= NomFonction(VarEntrée1, VarEntrée2, ... VarEntréeX);

Pour illustrer la procédure, utilisons la fonction `SetRTCDrift` représentée graphiquement ci-après :



La représentation en langage ST de cette fonction est la suivante :

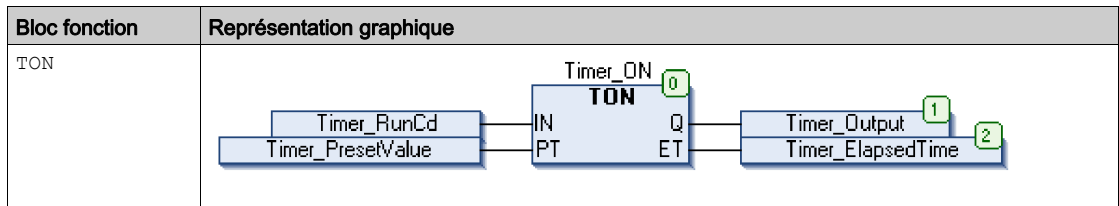
Fonction	Représentation dans l'éditeur ST de POU
SetRTCDrift	<pre>PROGRAM MyProgram_ST VAR myDrift: SINT(-29..29) := 5; myDay: DAY_OF_WEEK := SUNDAY; myHour: HOUR := 12; myMinute: MINUTE; myRTCAdjust: RTCDRIFT_ERROR; END_VAR myRTCAdjust:= SetRTCDrift(myDrift, myDay, myHour, myMinute);</pre>

## Utilisation d'un bloc fonction en langage ST

La procédure suivante explique comment insérer un bloc fonction en langage ST :

Étape	Action
1	Ouvrez ou créez une POU en langage ST (Structured Text ou Littéral structuré). <b>NOTE</b> : La procédure de création d'une POU n'est pas détaillée ici. Pour plus d'informations sur l'ajout, la déclaration et l'appel de POU, reportez-vous à la documentation associée.
2	Créez les variables d'entrée, les variables de sortie et l'instance requises pour le bloc fonction : <ul style="list-style-type: none"> <li>• Les variables d'entrée sont les paramètres d'entrée requis par le bloc fonction.</li> <li>• Les variables de sortie reçoivent la valeur renvoyée par le bloc fonction.</li> </ul>
3	Utilisez la syntaxe générale dans l' <b>éditeur ST de POU</b> pour la représentation en langage ST d'un bloc fonction. La syntaxe générale est la suivante : <pre>FunctionBlock_InstanceName (Input1:=VarInput1, Input2:=VarInput2, ... Ouput1=&gt;VarOutput1, Ouput2=&gt;VarOutput2, ...);</pre>

Pour illustrer la procédure, utilisons le bloc fonction TON représenté graphiquement ci-après :



Le tableau suivant montre plusieurs exemples d'appel de bloc fonction en langage ST :

Bloc fonction	Représentation dans l'éditeur ST de POU
TON	<pre>1 PROGRAM MyProgram_ST 2 VAR 3     Timer_ON: TON; // Function Block Instance 4     Timer_RunCd: BOOL; 5     Timer_PresetValue: TIME := T#5S; 6     Timer_Output: BOOL; 7     Timer_ElapsedTime: TIME; 8 END_VAR  1 Timer_ON( 2     IN:=Timer_RunCd, 3     PT:=Timer_PresetValue, 4     Q=&gt;Timer_Output, 5     ET=&gt;Timer_ElapsedTime);</pre>



## C

### CFC

Acronyme de *continuous function chart* (diagramme fonctionnel continu). Langage de programmation graphique (extension de la norme IEC 61131-3) basé sur le langage de diagramme à blocs fonction et qui fonctionne comme un diagramme de flux. Toutefois, il n'utilise pas de réseaux et le positionnement libre des éléments graphiques est possible, ce qui permet les boucles de retour. Pour chaque bloc, les entrées se situent à gauche et les sorties à droite. Vous pouvez lier les sorties de blocs aux entrées d'autres blocs pour créer des expressions complexes.

## F

### FB

Acronyme de *function block*, bloc fonction. Mécanisme de programmation commode qui consolide un groupe d'instructions de programmation visant à effectuer une action spécifique et normalisée telle que le contrôle de vitesse, le contrôle d'intervalle ou le comptage. Un bloc fonction peut comprendre des données de configuration, un ensemble de paramètres de fonctionnement interne ou externe et généralement une ou plusieurs entrées et sorties de données.

## I

### IL

Acronyme de *instruction list*, liste d'instructions. Un programme écrit en langage IL est composé d'instructions textuelles qui sont exécutées séquentiellement par le contrôleur. Chaque instruction comprend un numéro de ligne, un code d'instruction et un opérande (voir la norme IEC 61131-3).

### INT

Abréviation de *integer*, nombre entier codé sur 16 bits.

## L

### langage en blocs fonctionnels

Un des 5 langages de programmation de logique ou de commande pris en charge par la norme IEC 61131-3 pour les systèmes de commande. FBD est un langage de programmation orienté graphique. Il fonctionne avec une liste de réseaux où chaque réseau contient une structure graphique de zones et de lignes de connexion représentant une expression logique ou arithmétique, un appel de bloc fonction ou une instruction de retour.

## LD

Acronyme de *ladder diagram*, schéma à contacts. Représentation graphique des instructions d'un programme de contrôleur, avec des symboles pour les contacts, les bobines et les blocs dans une série de réseaux exécutés séquentiellement par un contrôleur (voir IEC 61131-3).

## O

### octet

Type codé sur 8 bits, de 00 à FF au format hexadécimal.

## P

### POU

Acronyme de *program organization unit*, unité organisationnelle de programme. Déclaration de variables dans le code source et jeu d'instructions correspondant. Les POU facilitent la réutilisation modulaire de programmes logiciels, de fonctions et de blocs fonction. Une fois déclarées, les POU sont réutilisables.

## S

### ST

Acronyme de *structured text*, texte structuré. Langage composé d'instructions complexes et d'instructions imbriquées (boucles d'itération, exécutions conditionnelles, fonctions). Le langage ST est conforme à la norme IEC 61131-3.

## V

### variable

Unité de mémoire qui est adressée et modifiée par un programme.



## A

absolu SSI  
modes du codeur, *15*

## B

blocs fonction  
FB\_Encoder\_M262, *18*  
FB\_EncoderCapture\_M262, *23*  
FB\_EncoderPreset\_M262, *21*  
FB\_EncoderReadScalingParam\_M262,  
*25*

## E

ET\_ENC\_CAP\_EDGE\_M262  
type de données, *28*  
ET\_ENC\_ERROR\_M262  
type de données, *29*  
ET\_ENC\_INPUT\_M262  
type de données, *30*  
ET\_ENC\_PRESET\_MODE\_M262  
type de données, *31*

## F

FB\_Encoder\_M262  
bloc fonction, *18*  
FB\_EncoderCapture\_M262  
bloc fonction, *23*  
FB\_EncoderPreset\_M262  
bloc fonction, *21*  
FB\_EncoderReadScalingParam\_M262  
bloc fonction, *25*  
fonctions  
différences entre une fonction et un bloc  
fonction, *36*  
utilisation d'une fonction ou d'un bloc  
fonction en langage IL, *37*  
utilisation d'une fonction ou d'un bloc

fonction en langage ST, *42*

## I

incrémental  
modes du codeur, *12*

## M

modes du codeur  
absolu SSI, *15*  
incrémental, *12*

## T

types d'entrée  
ET\_ENC\_PRESET\_MODE\_M262, *31*  
types de données  
ET\_ENC\_CAP\_EDGE\_M262, *28*  
ET\_ENC\_ERROR\_M262, *29*  
ET\_ENC\_INPUT\_M262, *30*

