

EcoStruxure Machine Expert

Toolbox

Guide de la bibliothèque

EIO0000000097.09

12/2023

Mentions légales

La marque Schneider Electric et toutes les marques de commerce de Schneider Electric SE et de ses filiales mentionnées dans ce guide sont la propriété de Schneider Electric SE ou de ses filiales. Toutes les autres marques peuvent être des marques de commerce de leurs propriétaires respectifs. Ce guide et son contenu sont protégés par les lois sur la propriété intellectuelle applicables et sont fournis à titre d'information uniquement. Aucune partie de ce guide ne peut être reproduite ou transmise sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, photocopie, enregistrement ou autre), à quelque fin que ce soit, sans l'autorisation écrite préalable de Schneider Electric.

Schneider Electric n'accorde aucun droit ni aucune licence d'utilisation commerciale de ce guide ou de son contenu, sauf dans le cadre d'une licence non exclusive et personnelle, pour le consulter tel quel.

Les produits et équipements Schneider Electric doivent être installés, utilisés et entretenus uniquement par le personnel qualifié.

Les normes, spécifications et conceptions sont susceptibles d'être modifiées à tout moment. Les informations contenues dans ce guide peuvent faire l'objet de modifications sans préavis.

Dans la mesure permise par la loi applicable, Schneider Electric et ses filiales déclinent toute responsabilité en cas d'erreurs ou d'omissions dans le contenu informatif du présent document ou pour toute conséquence résultant de l'utilisation des informations qu'il contient.

En tant que membre d'un groupe d'entreprises responsables et inclusives, nous actualisons nos communications qui contiennent une terminologie non inclusive. Cependant, tant que nous n'aurons pas terminé ce processus, notre contenu pourra toujours contenir des termes standardisés du secteur qui pourraient être jugés inappropriés par nos clients.

© 2023 – Schneider Electric. Tous droits réservés.

Table des matières

Consignes de sécurité	9
QUALIFICATION DU PERSONNEL	9
INSTRUCTIONS D'UTILISATION	10
Avant de commencer	10
Démarrage et test	11
Fonctionnement et réglages	12
A propos de ce manuel	13
Exigences des fonctions	19
Exigences des fonctions.....	20
Configuration requise pour la bibliothèque de fonctions Toolbox	20
Fonctions de bits	21
Organisation de bits.....	22
Organisation des bits pour le type DWORD.....	22
SetBitTo: Activation/désactivation d'un bit.....	24
Fonction SetBitTo.....	24
TestBit: Test d'un bit.....	26
Fonction TestBit.....	26
Fonctions en boucle fermée	28
FB_2points : Elément de transfert en boucle fermée à 2 points	29
Bloc fonction FB_2points.....	29
Modes de fonctionnement	29
Description des broches d'entrée	31
Structure utilisée.....	31
Description des broches de sortie	32
FB_3points : élément de transfert de 3 points en boucle fermée	33
Bloc fonction FB_3points.....	33
Modes de fonctionnement	33
Description des broches d'entrée	35
Structure utilisée.....	35
Description des broches de sortie	36
FB_3points_Ext : Extension de boucle fermée pour l'élément de transfert en 3 points	37
Bloc fonction FB_3points_Ext.....	37
Modes de fonctionnement	37
Description des broches d'entrée	39
Structure utilisée.....	40
Description des broches de sortie	40
FB_P : Boucle de commande avec algorithme proportionnel seulement	42
Bloc fonction FB_P.....	42
Modes de fonctionnement	42
Description des broches d'entrée	44
Description des broches de sortie	44
FB_PI: Commande de processus proportionnelle et intégrale	46
Bloc fonction FB_PI.....	46
Description des broches d'entrée	47
Structure utilisée.....	49
Description des broches de sortie	51

FB_PID : Commande de processus de fonctionnement en mode manuel	52
Bloc fonction <code>FB_PID</code>	52
Description des broches d'entrée	55
Structure utilisée	57
Description des broches de sortie	58
Instanciation et exemple d'utilisation.....	61
FB_PI_PID : Boucle de commande en cascade PI_PID	64
Bloc fonction <code>FB_PI_PID</code>	64
Modes de fonctionnement	64
Description des broches d'entrée	66
Structures utilisées.....	68
Description des broches de sortie	68
Fonctions de commande d'équipement	71
FB_Cyclic_Monitoring : Surveillance cyclique	72
Bloc fonction <code>FB_Cyclic_Monitoring</code>	72
Description des broches d'entrée	73
Description des broches de sortie	74
Instanciation et exemple d'utilisation.....	74
FB_DeadBand: Suppression d'oscillations d'amplitude	76
Bloc fonction <code>FB_DeadBand</code>	76
Description des broches d'entrée	78
Description des broches de sortie	78
FB_Limiter : Limitation des signaux d'entrée	79
Bloc fonction <code>FB_Limiter</code>	79
Description des broches d'entrée	81
Description des broches de sortie	81
FB_PWM : Fourniture d'une sortie PWM	82
Bloc fonction <code>FB_PWM</code>	82
Description des broches d'entrée	86
Structure utilisée	87
Description des broches de sortie	87
FB_Redundant_Sensor_Monitoring : Surveillance de capteur redondant	88
Bloc fonction <code>FB_Redundant_Sensor_Monitoring</code>	88
Description des broches d'entrée	89
Description des broches de sortie	92
FB_Scaling : Mise à l'échelle des signaux d'entrée	93
Bloc fonction <code>FB_Scaling</code>	93
Description des broches d'entrée	95
Description des broches de sortie	96
FB_Sensor_Monitoring : Surveillance de capteur	97
Bloc fonction <code>FB_Sensor_Monitoring</code>	97
Description des broches d'entrée	98
Description des broches de sortie	99
Instanciation et exemple d'utilisation.....	99
Fonctions de filtrage	102
Liste des paramètres globaux.....	103
Liste des paramètres globaux (GPL)	103
Filter_AnalogInput : Vérification de la variabilité d'entrée analogique	104

Bloc fonction <code>Filter_AnalogInput</code>	104
Description des broches d'entrée	105
Description des broches de sortie	106
Filter_Arithmetic: Fourniture de moyenne arithmétique	107
Bloc fonction <code>Filter_Arithmetic</code>	107
Description des broches d'entrée	109
Description des broches de sortie	109
Filter_MovingAverage : Fourniture d'une valeur de moyenne mobile	110
Bloc fonction <code>Filter_MovingAverage</code>	110
Description des broches d'entrée	112
Description des broches de sortie	112
Filter_PT1 : Fourniture d'une fonction de transfert PT1	114
Bloc fonction <code>Filter_PT1</code>	114
Description des broches d'entrée	116
Description des broches de sortie	116
Instanciation et exemple d'utilisation.....	117
Retournements	120
JK_FlipFlop : Initialisation/réinitialisation de l'entrée vers sortie de retournement	121
Bloc fonction <code>JK_FlipFlop</code>	121
JK_FlipFlop_MasterSlave: Initialisation/réinitialisation de l'entrée de la sortie de retournement.....	123
Bloc fonction <code>JK_FlipFlop_MasterSlave</code>	123
RS_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement	126
Bloc fonction <code>RS_FlipFlop</code>	126
SR_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement	128
Bloc fonction <code>SR_FlipFlop</code>	128
Toggle_FlipFlop: Basculement de l'entrée/sortie de retournement	130
Bloc fonction <code>Toggle_FlipFlop</code>	130
Fonctions mathématiques	132
Analysis: Calcul de valeurs intégrales et dérivées	133
Bloc fonction <code>Analysis</code>	133
Frequency_Multiplier: Mise en œuvre de 32 clignoteurs.....	135
Bloc fonction <code>Frequency_Multiplier</code>	135
Sans maintien Description	136
Fonctionnalité avec description de condition	137
Frequency_Output: Mise en œuvre d'une fréquence	139
Bloc fonction <code>Frequency_Output</code>	139
Normalizer_With_Limiter: Mise à l'échelle de l'entrée Minimale et Maximale	144
Bloc fonction <code>Normalizer_With_Limiter</code>	144
ONE_SEC_PULSE: Fourniture d'impulsions à chaque seconde.....	147
Bloc fonction <code>ONE_SEC_PULSE</code>	147
Quantizer: Numérisation de la valeur d'entrée pour l'intervalle.....	148
Bloc fonction <code>Quantizer</code>	148
Signal_Saturation: Limitation aux limites supérieure et inférieure de saturation.....	150

Bloc fonction <code>Signal_Saturation</code>	150
Signal_Statistics: Calcul de Maximum, de Minimum, de Moyenne et de Variance	154
Bloc fonction <code>Signal_Statistics</code>	154
Check_Divisor: Vérification de condition de division par zéro	157
Bloc fonction <code>Check_Divisor</code>	157
Fonctions de conversion numérique	158
ArrayOfByte_TO_String: Conversion d'un tableau en format octet au format chaîne	159
Fonction <code>ArrayOfByte_TO_String</code>	159
DT_AS_WORD: Conversion de Date et Heure sous forme d'un mot.....	163
Bloc fonction <code>DT_AS_WORD</code>	163
DWORD_AS_WORD: Division d'un mot double en deux mots	165
Bloc fonction <code>DWORD_AS_DWORD</code>	165
String_TO_ArrayOfByte: Tableau de sortie et valeur ASCII de la chaîne d'entrée	166
Fonction <code>String_TO_ArrayOfByte</code>	166
WORD_AS_DWORD: Décalage du mot de poids fort et ajout du mot de poids faible.....	170
Bloc fonction <code>WORD_AS_DWORD</code>	170
Conversion physique	171
Celsius_TO_Fahrenheit: Conversion de Celsius en Fahrenheit.....	172
Bloc fonction <code>Celsius_TO_Fahrenheit</code>	172
Celsius_TO_Kelvin: Conversion de Celsius en Kelvin.....	173
Bloc fonction <code>Celsius_TO_Celsius</code>	173
Fahrenheit_TO_Celsius: Conversion de Fahrenheit en Celsius	175
Bloc fonction <code>Fahrenheit_TO_Celsius</code>	175
Frequency_TO_Period: Calcul de la période temporelle d'une fréquence donnée	176
Fonction <code>Frequency_TO_Period</code>	176
Kelvin_TO_Celsius: Conversion de Kelvin en Celsius.....	178
Bloc fonction <code>Kelvin_TO_Celsius</code>	178
Period_TO_Frequency: Calcul de la fréquence du temps donné	180
Fonction <code>Period_TO_Frequency</code>	180
Utilitaires	182
Hour_Meter : Accumulation d'heures de fonctionnement	183
Bloc fonction <code>Hour_Meter</code>	183
Description des broches d'entrée	185
Description des broches de sortie	185
Broche d'entrée – sortie.....	186
Structures utilisées.....	186
Description des bits du mot de commande	187
Mot d'état	187
Instanciation et exemple d'utilisation.....	188
Operation_Mode : Sélection du mode de fonctionnement	191
Bloc fonction <code>Operation_Mode</code>	191
Description des broches d'entrée	192
Description des broches de sortie	193
Description des bits du mot de commande	193

Mot d'état	194
Commande de vanne	195
Bistable_Valve: Commandes vannes bistables	196
Bloc fonction <code>Bistable_Valve</code>	196
Description des broches d'entrée	198
Description des broches de sortie	199
Structure utilisée	200
Description des bits du mot de commande	200
Mot d'état	200
Instanciation et exemple d'utilisation	201
Monostable_Valve: Commande de vannes monostables	202
Bloc fonction <code>Monostable_Valve</code>	202
Description des broches d'entrée	204
Description des broches de sortie	205
Structure utilisée	205
Description des bits du mot de commande	206
Mot d'état	206
Proportional_Valve: Commande de vannes proportionnelles	207
Bloc fonction <code>Proportional_Valve</code>	207
Description des broches d'entrée	209
Description des broches de sortie	210
Broche d'entrée/sortie	211
Structures utilisées	211
Description des bits du mot de commande	212
Mot d'état	212
Instanciation et exemple d'utilisation	213
Glossaire	215
Index	219

Consignes de sécurité

Informations importantes

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.

QUALIFICATION DU PERSONNEL

Une personne qualifiée est une personne qui a les qualifications suivantes :

- Compétences et connaissances liées à la construction et à l'exploitation d'équipements électriques et à l'installation.
- Connaissance de l'exploitation des fonctionnalités de la machine dans l'implémentation du logiciel.

- A reçu une formation en sécurité permettant de reconnaître et d'éviter les dangers potentiels.

La personne qualifiée doit être capable de détecter d'éventuels dangers qui pourraient découler du paramétrage, de modifications des valeurs de paramétrage et plus généralement des équipements mécaniques, électriques ou électroniques. La personne qualifiée doit connaître les normes, dispositions et réglementations liées à la prévention des accidents de travail, et doit les observer lors de la conception et de l'implémentation du système.

INSTRUCTIONS D'UTILISATION

Ce produit est une bibliothèque à utiliser avec un système de commande et des servo-amplificateurs. La bibliothèque n'est conçue qu'en vue d'une utilisation telle que décrite dans la présente documentation, appliquée au secteur industriel.

Observez en permanence les instructions applicables liées à la sécurité, les conditions spécifiques et les données techniques.

Réalisez une analyse des risques en rapport avec l'utilisation spécifique avant d'utiliser ce produit. Prenez les mesures de sécurité qui découlent des résultats.

Étant donné que ce produit est utilisé au sein d'un système qui l'englobe, vous devez assurer la sécurité du personnel par la conception même du système global (la conception de la machine, par exemple).

Aucune autre utilisation n'est prévue. Toute autre utilisation pourrait être dangereuse.

Avant de commencer

N'utilisez pas ce produit sur les machines non pourvues de protection efficace du point de fonctionnement. L'absence de ce type de protection sur une machine présente un risque de blessures graves pour l'opérateur.

▲ AVERTISSEMENT

EQUIPEMENT NON PROTEGE

- N'utilisez pas ce logiciel ni les automatismes associés sur des appareils non équipés de protection du point de fonctionnement.
- N'accédez pas aux machines pendant leur fonctionnement.

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

Cet automatisme et le logiciel associé permettent de commander des processus industriels divers. Le type ou le modèle d'automatisme approprié pour chaque application dépendra de facteurs tels que la fonction de commande requise, le degré de protection exigé, les méthodes de production, des conditions inhabituelles, la législation, etc. Dans certaines applications, plusieurs processeurs seront nécessaires, notamment lorsque la redondance de sauvegarde est requise.

Vous seul, en tant que constructeur de machine ou intégrateur de système, pouvez connaître toutes les conditions et facteurs présents lors de la configuration, de l'exploitation et de la maintenance de la machine, et êtes donc en mesure de déterminer les équipements automatisés, ainsi que les sécurités et verrouillages associés qui peuvent être utilisés correctement. Lors du choix de l'automatisme et du système de commande, ainsi que du logiciel associé pour une application particulière, vous devez respecter les normes et réglementations locales et nationales en vigueur. Le document National Safety Council's Accident Prevention Manual (reconnu aux Etats-Unis) fournit également de nombreuses informations utiles.

Dans certaines applications, telles que les machines d'emballage, une protection supplémentaire, comme celle du point de fonctionnement, doit être fournie pour l'opérateur. Elle est nécessaire si les mains ou d'autres parties du corps de l'opérateur peuvent entrer dans la zone de point de pincement ou d'autres zones dangereuses, risquant ainsi de provoquer des blessures graves. Les produits logiciels seuls, ne peuvent en aucun cas protéger les opérateurs contre d'éventuelles blessures. C'est pourquoi le logiciel ne doit pas remplacer la protection de point de fonctionnement ou s'y substituer.

Avant de mettre l'équipement en service, assurez-vous que les dispositifs de sécurité et de verrouillage mécaniques et/ou électriques appropriés liés à la protection du point de fonctionnement ont été installés et sont opérationnels. Tous les dispositifs de sécurité et de verrouillage liés à la protection du point de fonctionnement doivent être coordonnés avec la programmation des équipements et logiciels d'automatisation associés.

NOTE: La coordination des dispositifs de sécurité et de verrouillage mécaniques/électriques du point de fonctionnement n'entre pas dans le cadre de cette bibliothèque de blocs fonction, du Guide utilisateur système ou de toute autre mise en œuvre référencée dans la documentation.

Démarrage et test

Avant toute utilisation de l'équipement de commande électrique et des automatismes en vue d'un fonctionnement normal après installation, un technicien qualifié doit procéder à un test de démarrage afin de vérifier que l'équipement fonctionne correctement. Il est essentiel de planifier une telle vérification et d'accorder suffisamment de temps pour la réalisation de ce test dans sa totalité.

▲ AVERTISSEMENT

RISQUES INHERENTS AU FONCTIONNEMENT DE L'EQUIPEMENT

- Assurez-vous que toutes les procédures d'installation et de configuration ont été respectées.
- Avant de réaliser les tests de fonctionnement, retirez tous les blocs ou autres cales temporaires utilisés pour le transport de tous les dispositifs composant le système.
- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.

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

Effectuez tous les tests de démarrage recommandés dans la documentation de l'équipement. Conservez toute la documentation de l'équipement pour référence ultérieure.

Les tests logiciels doivent être réalisés à la fois en environnement simulé et réel

Vérifiez que le système entier est exempt de tout court-circuit et mise à la terre temporaire non installée conformément aux réglementations locales (conformément au National Electrical Code des Etats-Unis, par exemple). Si des tests diélectriques sont nécessaires, suivez les recommandations figurant dans la documentation de l'équipement afin d'éviter de l'endommager accidentellement.

Avant de mettre l'équipement sous tension :

- Enlevez les outils, les instruments de mesure et les débris éventuels présents sur l'équipement.
- Fermez le capot du boîtier de l'équipement.
- Retirez toutes les mises à la terre temporaires des câbles d'alimentation entrants.
- Effectuez tous les tests de démarrage recommandés par le fabricant.

Fonctionnement et réglages

Les précautions suivantes sont extraites du document NEMA Standards
Publication ICS 7.1-1995 :

(En cas de divergence ou de contradiction entre une traduction et l'original
anglais, le texte original en anglais prévaudra.)

- Malgré le soin apporté à la conception et à la fabrication de l'équipement ou au choix et à l'évaluation des composants, des risques subsistent en cas d'utilisation inappropriée de l'équipement.
- Il arrive parfois que l'équipement soit dérégulé accidentellement, entraînant ainsi un fonctionnement non satisfaisant ou non sécurisé. Respectez toujours les instructions du fabricant pour effectuer les réglages fonctionnels. Les personnes ayant accès à ces réglages doivent connaître les instructions du fabricant de l'équipement et les machines utilisées avec l'équipement électrique.
- L'opérateur ne doit avoir accès qu'aux réglages fonctionnels dont il a besoin. L'accès aux autres commandes doit être limité afin d'empêcher les changements non autorisés des caractéristiques de fonctionnement.

A propos de ce manuel

Objectif du document

Ce document décrit les fonctions de la bibliothèque Toolbox EcoStruxure Machine Expert.

Champ d'application

Ce document a été actualisé pour le lancement de EcoStruxure™ Machine Expert V2.2.

Les caractéristiques décrites dans le présent document, ainsi que celles décrites dans les documents mentionnés dans la section Documents associés ci-dessous, sont consultables en ligne. Pour accéder aux informations en ligne, allez sur la page d'accueil de Schneider Electric www.se.com/ww/fr/download/.

Les caractéristiques décrites dans le présent document doivent être identiques à celles fournies en ligne. Toutefois, en application de notre politique d'amélioration continue, nous pouvons être amenés à réviser le contenu du document afin de le rendre plus clair et plus précis. Si vous constatez une différence entre le document et les informations fournies en ligne, utilisez ces dernières en priorité.

Document(s) à consulter

Titre du document	Référence
Meilleures pratiques en matière de cybersécurité	CS-Best-Practices-2019-340
Recommandations de cybersécurité pour les solutions d'automatisme EcoStruxure Machine Expert, Modicon et PacDrive et les équipements associés	EIO0000004242
EcoStruxure Machine Expert - Fonctions et bibliothèques - Guide de l'utilisateur	EIO0000002829 (ENG); EIO0000002830 (FRE); EIO0000002831 (GER); EIO0000002832 (ITA); EIO0000002833 (SPA); EIO0000002834 (CHS)
EcoStruxure Machine Expert - Guide de programmation	EIO0000002854 (ENG); EIO0000002855 (FRE); EIO0000002856 (GER); EIO0000002857 (ITA); EIO0000002858 (SPA); EIO0000002859 (CHS)

Informations produit

NOTE: Schneider Electric respecte les bonnes pratiques de l'industrie, en vigueur dans le développement et la mise en œuvre des systèmes de contrôle. Cette approche, dite de « défense en profondeur », permet de sécuriser les systèmes de contrôle industriels. Elle place les contrôleurs derrière des pare-feu pour restreindre leur accès aux seuls personnels et protocoles autorisés.

▲ AVERTISSEMENT

ACCES NON AUTHENTIFIE ET EXPLOITATION PAR CONSEQUENT NON AUTORISEE DES MACHINES

- Estimez si votre environnement ou vos machines sont connecté(e)s à votre infrastructure vitale et, le cas échéant, prenez les mesures nécessaires de prévention, basées sur le principe de défense en profondeur, avant de connecter le système d'automatisme à un réseau quelconque.
- Limitez au strict nécessaire le nombre d'équipements connectés à un réseau.
- Isolez votre réseau industriel des autres réseaux au sein de votre société.
- Protégez chaque réseau contre les accès non autorisés à l'aide d'un pare-feu, d'un VPN ou d'autres mesures de sécurité éprouvées.
- Surveillez les activités au sein de votre système.
- Empêchez tout accès direct ou liaison directe aux équipements sensibles par des utilisateurs non autorisés ou des actions non authentifiées.
- Préparez un plan de récupération intégrant la sauvegarde des informations de votre système et de votre processus.

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

Pour plus d'informations sur les mesures organisationnelles et les règles d'accès aux infrastructures, reportez-vous aux normes suivantes : famille de normes ISO/IEC 27000, Critères Communs pour l'évaluation de la sécurité des Technologies de l'Information, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, Cybersecurity Framework (Cadre de cybersécurité) du NIST, Standard of Good Practice for Information Security (Bonne pratique de sécurité de l'information) de l'Information Security Forum. Consultez également le document *Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment*.

⚠ AVERTISSEMENT

PERTE DE CONTROLE

- Réalisez une analyse des modes de défaillance et de leurs effets (FMEA) ou une analyse de risques équivalente sur l'application et appliquez les contrôles de prévention et de détection appropriés avant la mise en œuvre.
- Prévoyez un état de repli pour les événements ou séquences de commande indésirables.
- Le cas échéant, prévoyez des chemins de commande séparés et redondants.
- Définissez les paramètres appropriés, notamment pour les limites.
- Examinez les conséquences des retards de transmission et prenez les mesures correctives nécessaires.
- Examinez les conséquences des interruptions de la liaison de communication et prenez des mesures correctives nécessaires.
- Prévoyez des chemins indépendants pour les fonctions de commande critiques (arrêt d'urgence, dépassement de limites, conditions d'erreur, etc.) en fonction de votre évaluation des risques ainsi que des réglementations et consignes applicables.
- Appliquez les réglementations et les consignes locales de sécurité et de prévention des accidents.¹
- Testez chaque mise en œuvre d'un système pour vérifier son bon fonctionnement avant de le mettre en service.

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

¹ 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, d'installation et d'exploitation de variateurs de vitesse) ou leur équivalent en vigueur dans votre pays.

Avant de tenter de fournir une solution (machine ou processus) pour une application spécifique en utilisant les POU trouvés dans la bibliothèque, vous devez tenir compte de la réalisation et de l'exécution des bonnes pratiques. La liste non exhaustive de ces pratiques liées à cette bibliothèque inclut l'analyse des risques, la sécurité fonctionnelle, la compatibilité des composants, les tests et la validation du système.

▲ AVERTISSEMENT

UTILISATION INCORRECTE DES UNITES ORGANISATIONNELLES DU PROGRAMME

- Effectuez une analyse de la sécurité de l'application et des équipements installés.
- Vérifiez que les POU sont compatibles avec les équipements du système et n'ont pas d'effets inattendus sur le bon fonctionnement du système.
- Assurez-vous que l'axe est en position d'origine et que le référencement est valide avant d'utiliser des mouvements absolus ou des POU utilisant des mouvements absolus.
- Utilisez les paramètres appropriés, notamment les valeurs limites, et observez l'usure de la machine et son fonctionnement à l'arrêt.
- Vérifiez que les capteurs et déclencheurs sont compatibles avec les POU sélectionnés.
- Testez de manière approfondie toutes les fonctions durant la vérification et la mise en service dans tous les modes de fonctionnement.
- Indiquez des méthodes indépendantes pour les fonctions de contrôle critiques (arrêt d'urgence, conditions des valeurs limites dépassées, etc.) conformément à une analyse de sécurité, aux règles en vigueur et aux réglementations applicables.

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

▲ AVERTISSEMENT

FONCTIONNEMENT IMPREVU DE L'EQUIPEMENT

- N'utilisez 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.
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.

Exigences des fonctions

Contenu de cette partie

Exigences des fonctions	20
-------------------------------	----

Vue d'ensemble

La partie décrit les exigences des fonctions.

Exigences des fonctions

Contenu de ce chapitre

Configuration requise pour la bibliothèque de fonctions Toolbox..... 20

Vue d'ensemble

Ce chapitre décrit les exigences des fonctions.

Configuration requise pour la bibliothèque de fonctions Toolbox

Configuration matérielle requise

La bibliothèque de fonctions Toolbox s'utilise avec tous les contrôleurs Schneider Electric gérés par le logiciel EcoStruxure Machine Expert.

Utilisation de la bibliothèque

⚠ AVERTISSEMENT

FONCTIONNEMENT IMPRÉVU DE L'ÉQUIPEMENT

- Vérifiez que les bibliothèques EcoStruxure Machine Expert incluses dans votre programme ont la version correcte après mise à jour du logiciel EcoStruxure Machine Expert.
- Vérifiez que les versions des bibliothèques mises à jour correspondent aux spécifications de l'application.

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

Pour obtenir des informations détaillées, consultez Bibliothèques de Schneider Electric (voir EcoStruxure Machine Expert, Présentation des bibliothèques).

Fonctions de bits

Contenu de cette partie

Organisation de bits.....	22
SetBitTo: Activation/désactivation d'un bit.....	24
TestBit: Test d'un bit.....	26

Vue d'ensemble

Cette partie décrit la famille de Fonctions de bits.

Organisation de bits

Contenu de ce chapitre

Organisation des bits pour le type DWORD 22

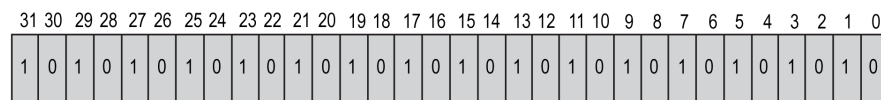
Vue d'ensemble

Ce chapitre décrit l'Organisation de bits.

Organisation des bits pour le type DWORD

Organisation des bits pour le type DWORD

Cette figure présente la règle de codage des positions de bit dans un DWORD. Un exemple est donné pour 16#AAAAAAAA correspondant à la valeur de 2863311530.



Description des types de données entiers

Ce tableau présente les types de données entiers. Chacun des différents types de nombre correspond à une gamme de types différente.

Type	Limite inférieure	Limite supérieure	Espace mémoire
BYTE	0	255	8 bits
WORD	0	65 535	16 bits
DWORD	0	4 294 967 295	32 bits
LWORD	0	2 ⁶⁴ -1	64 bits
SINT	-128	127	8 bits
USINT	0	255	8 bits
INT	-32 768	32 767	16 bits
UINT	0	65 535	16 bits
DINT	-2147483648	214 7483 647	32 bits
UDINT	0	4 294 967 295	32 bits
LINT	-2 ⁶³	2 ⁶³ -1	64 bits
ULINT	0	2 ⁶⁴ -1	64 bits

Description des types de données REAL/LREAL

Ce tableau présente les types de données REAL/LREAL. Les types REAL et LREAL sont appelés types à virgule flottante. Ils sont indispensables pour représenter des nombres rationnels.

Type	Plage	Résolution	Espace mémoire
REAL utilise 4 octets	-3,402e+38 à 3,402e+38 (-2 ¹²⁸ ...2 ¹²⁸)	1,175e-38 (2 ⁻¹²⁶)	32 bits
LREAL utilise 8 octets	-1,797e+308 à 1,797e+308 (-2 ¹⁰²⁴ ...2 ¹⁰²⁴)	2,225e-308 (2 ⁻¹⁰²²)	64 bits

NOTE: La prise en charge du type de données LREAL dépend de l'équipement cible. Consultez la documentation correspondante pour savoir si le type LREAL 64 bits est converti en REAL lors de la compilation (éventuellement avec perte d'informations) ou s'il est conservé en l'état.

SetBitTo: Activation/désactivation d'un bit

Contenu de ce chapitre

Fonction SetBitTo 24

Vue d'ensemble

Ce chapitre décrit la fonction SetBitTo.

Fonction SetBitTo

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction SetBitTo :



Description fonctionnelle

La fonction SetBitTo règle ou réinitialise (selon l'entrée) un bit spécifié par le bit dans l'entrée *DWORD* donnée. Les bits sont rangés du poids faible au poids fort en commençant à 0.

La plage valide est comprise entre 0 et 31.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée de la fonction SetBitTo :

Entrée	Type de données	Description
i_dwInput	DWORD	Valeur d'entrée Plage : 0 à 4 294 967 295
i_iPos	INT	Position du bit Plage : 0 à 31
i_xSet	BOOL	TRUE: régler FALSE : réinitialiser.

Description des broches de sortie

Ce tableau décrit les broches de sortie de la fonction SetBitTo :

Sortie	Type de données	Description
SetBitTo	DWORD	Valeur de sortie Plage : 0 à 4 294 967 295

Limitations

Si l'entrée i_iPos n'est pas dans la plage valable, l'entrée est interprétée en mode modulo.

TestBit: Test d'un bit

Contenu de ce chapitre

Fonction `TestBit`26

Vue d'ensemble

Ce chapitre décrit la fonction `TestBit`.

Fonction `TestBit`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `TestBit` :



Description fonctionnelle

La fonction `TestBit` teste un bit spécifié par le bit dans l'entrée `DWORD`. Les bits sont rangés du poids faible au poids fort en commençant à 0.

La sortie affiche l'état de présence du bit dans la position spécifiée. La plage valide est comprise entre 0 et 31.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée de la fonction `TestBit` :

Entrée	Type de données	Description
<code>i_dwInput</code>	DWORD	Valeur d'entrée Plage : 0...4294967295
<code>i_iPos</code>	INT	Rang du bit Plage : 0...31

Description des broches de sortie

Ce tableau décrit les broches de sortie de la fonction `TestBit` :

Sortie	Type de données	Description
<code>TestBit</code>	BOOL	Le résultat est True ou False.

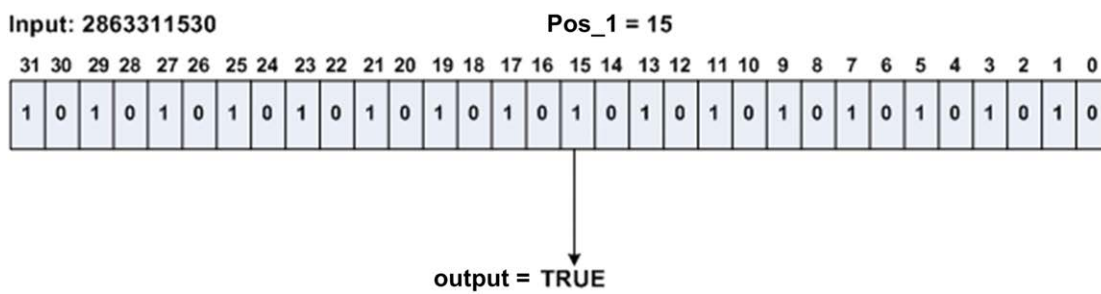
Limitations

Si l'entrée i_iPos n'est pas dans la plage valable, l'entrée est interprétée en mode modulo.

Exemple d'utilisation



Cette figure constitue un exemple de la fonction `TestBit` :



Fonctions en boucle fermée

Contenu de cette partie

FB_2points : Elément de transfert en boucle fermée à 2 points	29
FB_3points : élément de transfert de 3 points en boucle fermée.....	33
FB_3points_Ext : Extension de boucle fermée pour l'élément de transfert en 3 points	37
FB_P : Boucle de commande avec algorithme proportionnel seulement	42
FB_PI: Commande de processus proportionnelle et intégrale	46
FB_PID : Commande de processus de fonctionnement en mode manuel	52
FB_PI_PID : Boucle de commande en cascade PI_PID	64

Présentation

Cette partie décrit la famille de fonctions en boucle fermée.

FB_2points : Elément de transfert en boucle fermée à 2 points

Contenu de ce chapitre

Bloc fonction FB_2points	29
Modes de fonctionnement.....	29
Description des broches d'entrée.....	31
Structure utilisée	31
Description des broches de sortie	32

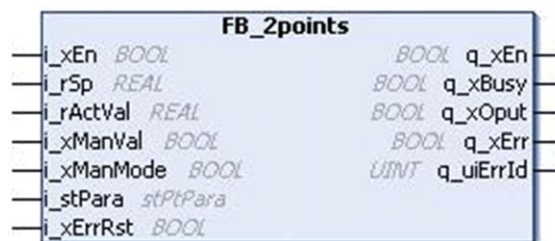
Vue d'ensemble

Ce chapitre décrit le bloc fonction d'élément de transfert FB_2points.

Bloc fonction FB_2points

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_2points :



Description fonctionnelle

Le bloc fonction FB_2points fournit une fonction de transfert en 2 points basé sur une entrée d'hystérésis.

Modes de fonctionnement

Mode automatique

En mode automatique (*i_xEn* a la valeur TRUE et *i_xManMode* a la valeur FALSE), si l'erreur de processus calculée est supérieure à 50 % de l'hystérésis dans le sens positif, alors *q_xOput* a la valeur TRUE.

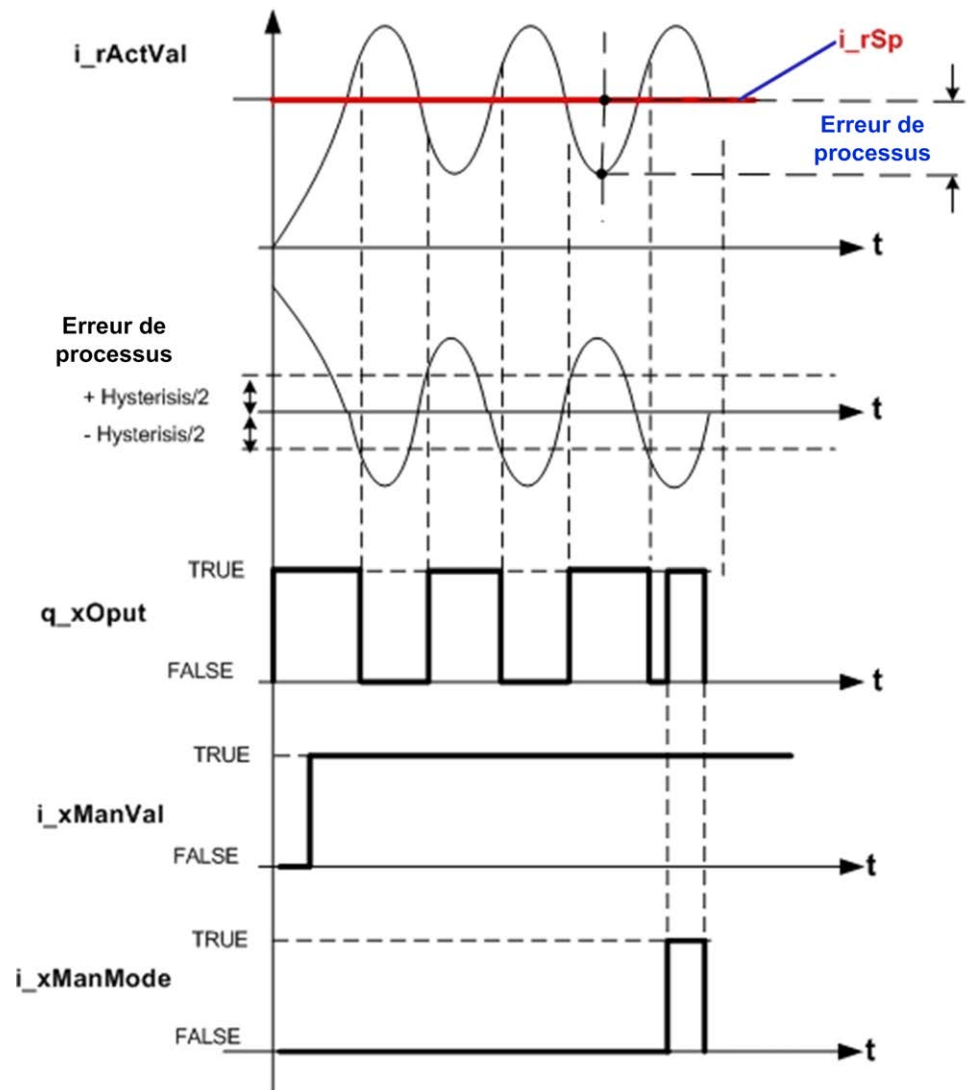
q_xOput n'est réinitialisé que si l'erreur de processus calculée descend en dessous de 50 % de l'hystérésis dans le sens négatif.

Erreur de processus = Valeur du point de consigne - Valeur réelle

Mode manuel

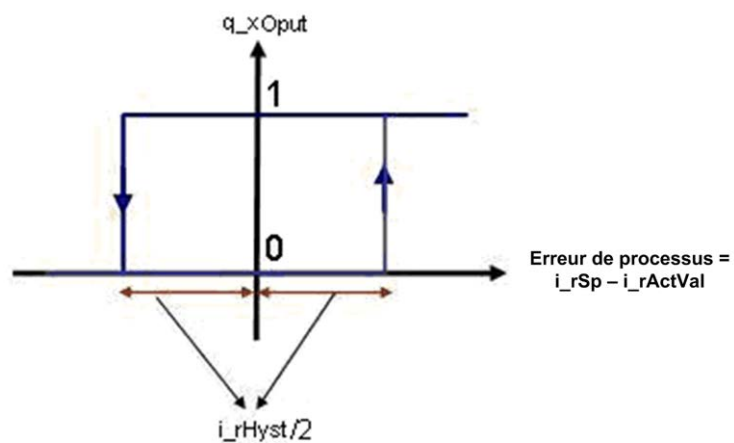
Si $i_xManMode$ a la valeur TRUE, quelles que soient les entrées en cours, la sortie est mise à l'état de $i_xManVal$.

Cette figure présente le chronogramme du bloc fonction FB_2points :



Chronogramme du mode

Cette figure présente la fonction de transfert du bloc fonction FB_2points.



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, la valeur de sortie est mise à zéro. L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée `i_xErrRst`.

La sortie `q_xBusy` a la valeur TRUE, chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `FB_2points` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : Active le bloc fonction FALSE : Désactive le bloc fonction
<code>i_rSp</code>	REAL	Valeur de point de consigne Plage : $\pm 3,4e^{+38}$
<code>i_rActVal</code>	REAL	Valeur réelle Plage : $\pm 3,4e^{+38}$
<code>i_xManVal</code>	BOOL	Valeur manuelle (optionnel)
<code>i_xManMode</code>	BOOL	Mode Manuel (optionnel)
<code>i_stPara</code>	STRUCT <code>stPtPara</code>	Paramètre de structure (Consultez le tableau ci-dessous).
<code>i_xErrRst</code>	BOOL	Réinitialiser l'erreur détectée (Le front montant réinitialise l'erreur détectée). (optionnel)

Structure utilisée

`stPtPara`

Élément de structure	Type	Description
<code>tCyclTime</code>	TIME	Durée de cycle de tâche Plage : $1...1e^{32}$ ms
<code>rHyst</code>	REAL	La plage de l'hystérésis doit être supérieure à 0,0. Plage : $\pm 3,4e^{+38}$

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_2points :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : FB activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Bloc fonction actif et pas d'erreur détectée. FALSE : Bloc fonction désactivé ou le bloc fonction a détecté une erreur
q_xOput	BOOL	TRUE : Si l'erreur de processus est supérieure à 50% de l'hystérésis FALSE : Si q_xOput a la valeur TRUE et que l'erreur détectée descend en dessous de 50% de l'hystérésis dans le sens négatif.
q_xErr	BOOL	TRUE : Pendant une erreur détectée FALSE : Si aucune erreur n'est détectée
q_uiErrId	UNITE	0 = Aucune erreur détectée 1 = Paramètre non valable Temps de cycle de tâche = 0 2 = Paramètre non valable i_rHyst <= 0.

FB_3points : élément de transfert de 3 points en boucle fermée

Contenu de ce chapitre

Bloc fonction FB_3points	33
Modes de fonctionnement.....	33
Description des broches d'entrée.....	35
Structure utilisée	35
Description des broches de sortie	36

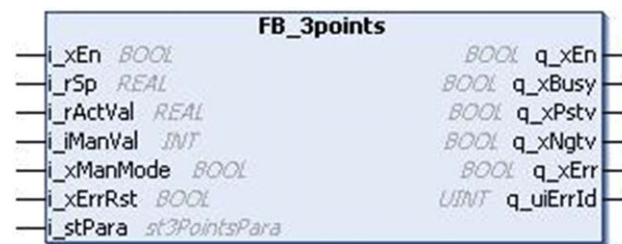
Présentation

Ce chapitre décrit le bloc fonction d'élément de transfert FB_3points.

Bloc fonction FB_3points

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_3points :



Description fonctionnelle

Le bloc fonction FB_3points fournit un élément de transfert en 3 points dans le logigramme fonctionnel.

Modes de fonctionnement

Mode automatique

Ce bloc fonction vérifie la valeur de l'erreur de processus (point de consigne - valeur réelle).

Si l'erreur de processus est positive et supérieure à la valeur de seuil supérieur pour le côté positif rPstvOutOn, elle active le signal de sortie q_xPstv.

La sortie q_xPstv est réinitialisée si l'erreur de processus descend en dessous de la valeur de seuil inférieure pour le côté positif rPstvOutOff.

Si l'erreur de processus est négative, le signal de sortie q_xNgtv est activé lors du dépassement de cible de la valeur seuil supérieure pour le côté négatif rNgtvOutOn.

q_xNgtv est réinitialisé lors de la descente en dessous de la valeur de seuil inférieure pour le côté négatif rNgtvOutOff.

Mode Manuel

La sortie du bloc fonction est définie manuellement selon la valeur de la broche d'entrée $i_rManVal$.

If

$i_iManVal \geq 1$ then $q_xPstv = TRUE$, $q_xNgtv = FALSE$.

$i_iManVal \leq -1$ then $q_xPstv = FALSE$, $q_xNgtv = TRUE$.

Else

$q_xPstv = FALSE$, $q_xNgtv = FALSE$

Cette figure présente la fonction de transfert pour le bloc fonction $FB_3points$:

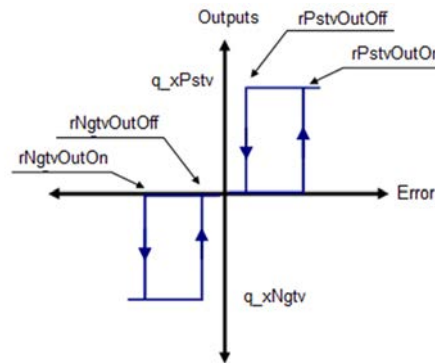
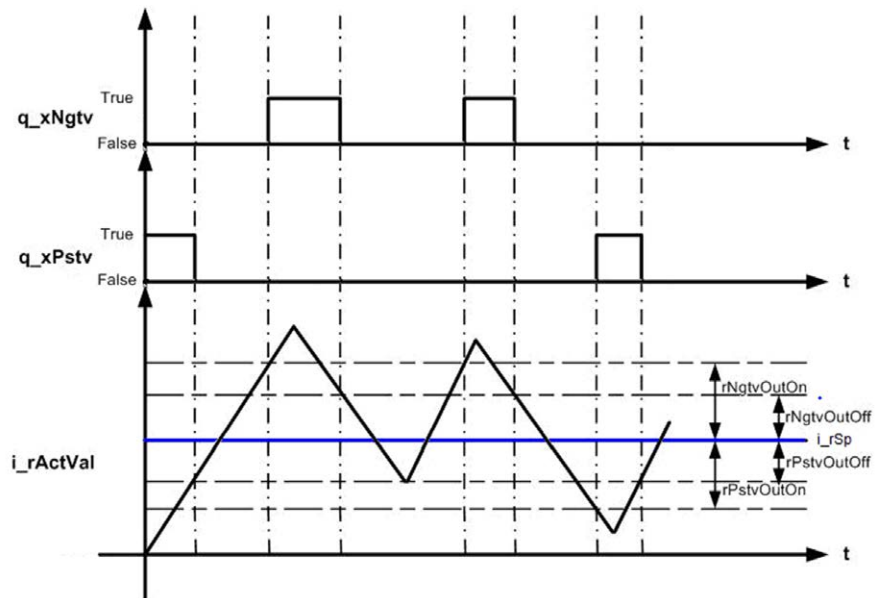


Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction $FB_3points$:



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une d'erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, les valeurs de sortie sont mises à zéro. L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$.

La sortie `q_xBusy` a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `FB_3points` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : active le bloc fonction. FALSE : bloc fonction désactivé.
<code>i_rSp</code>	REAL	Valeur du point de consigne du processus Plage : $\pm 3,4e^{+38}$
<code>i_rActVal</code>	REAL	Valeur réelle du processus Plage : $\pm 3,4e^{+38}$
<code>i_iManVal</code>	INT	Valeur d'entrée pour le mode manuel. Plage : -32 768 à 32 767
<code>i_xManMode</code>	BOOL	TRUE : bloc fonction en mode manuel. FALSE : bloc fonction en mode automatique. (Facultatif)
<code>i_xErrRst</code>	BOOL	Réinitialise l'erreur détectée (Déclenché par un front montant). (Facultatif)
<code>i_stPara</code>	STRUCT <code>st3PointsPara</code>	Variable de structure du bloc fonction

Structure utilisée

st3PointsPara

Élément de structure	Type	Description
<code>rPstvOutOn</code>	REAL	Valeur de seuil supérieure pour le côté positif de l'erreur de processus Plage : $0 \dots 1e^{38}$
<code>rPstvOutOff</code>	REAL	Valeur de seuil inférieure pour le côté positif de l'erreur de processus Plage : $0 \dots 1e^{38}$
<code>rNgtvOutOn</code>	REAL	Valeur de seuil supérieure pour le côté négatif de l'erreur de processus Plage : $0 \dots 1e^{38}$
<code>rNgtvOutOff</code>	REAL	Valeur de seuil inférieure pour le côté négatif de l'erreur de processus Plage : $0 \dots 1e^{38}$

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_3points :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : Bloc fonction activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Bloc fonction actif et pas d'erreur détectée. FALSE : Bloc fonction désactivé ou erreur détectée
q_xPstv	BOOL	Cette sortie de l'élément 3 points a la valeur TRUE si la branche supérieure de la courbe d'hystérésis est active.
q_xNgtv	BOOL	Cette sortie de l'élément 3 points a la valeur TRUE si la branche inférieure de la courbe d'hystérésis est active.
q_xErr	BOOL	Signal d'erreur détectée
q_uiErrId	UINT	Fournit l'ID d'erreur détectée quand la sortie q_xErr est active. Plage : 0...3

q_uiErrId

Cette valeur entière unique indique une erreur détectée spécifique :

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	Valeurs de paramètre non valables (si $rPstvOutOn < 0$ ou $rPstvOutOff < 0$ ou $rNgtvOutOff < 0$ ou $rNgtvOutOn < 0$)
2	Valeurs de paramètre non valables ($rPstvOutOn < rPstvOutOff$ ou $rNgtvOutOn < rNgtvOutOff$)
3	Erreur interne détectée (bloc fonction en état inconnu)

FB_3points_Ext : Extension de boucle fermée pour l'élément de transfert en 3 points

Contenu de ce chapitre

Bloc fonction FB_3points_Ext	37
Modes de fonctionnement	37
Description des broches d'entrée	39
Structure utilisée	40
Description des broches de sortie	40

Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_3points_Ext.

Bloc fonction FB_3points_Ext

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_3points_Ext :



Description fonctionnelle

Le bloc fonction FB_3points_Ext fournit un élément de transfert en 3 points dans le logigramme fonctionnel.

Ce bloc fonction est une extension du bloc fonction FB_3points. Il produit une sortie de commande q_rOput sous forme de boucle d'hystérésis à 3 points. La sortie dépend de l'erreur de processus, des valeurs de gain et de décalage données par l'utilisateur.

Modes de fonctionnement

Mode automatique

Ce bloc fonction vérifie la valeur de l'erreur de processus (différence entre le point de consigne et la valeur réelle). Si l'erreur de processus est positive et supérieure à la valeur supérieure de seuil rOutOn, elle calcule la sortie de commande comme suit.

$$q_rOput = \text{Erreur de processus} \times rGain + rOfst$$

Si l'erreur de processus descend en dessous de la valeur inférieure de seuil rOutOff, elle réinitialise la sortie de commande à zéro.

De même si l'erreur de processus est négative et si sa valeur absolue est supérieure à la valeur supérieure de seuil r_{OutOn} , elle calcule la sortie de commande comme suit,

$$q_{rOput} = \text{Erreur de processus} \times r_{Gain} - r_{Ofst}$$

q_{rOput} est remis à zéro si la valeur absolue de l'erreur de processus devient inférieure à la valeur de seuil inférieure r_{OutOff} .

Mode Manuel

La sortie du bloc fonction est définie manuellement selon la valeur de la broche d'entrée i_{rManVa} .

IF	AND IF	THEN
$Abs(i_{rManVal}) < 1$	-	$q_{rOput} = 0,0$
$Abs(i_{rManVal}) \geq 1$	$rE > 0$	$q_{rOput} = rE \times r_{Gain} + r_{Ofst}$
$Abs(i_{rManVal}) \geq 1$	$rE < 0$	$q_{rOput} = rE \times r_{Gain} - r_{Ofst}$
$Abs(i_{rManVal}) \geq 1$	$rE = 0$	$q_{rOput} = 0,0$

$rE = i_{rSp} - i_{rActVal}$
Abs() Fonction valeur absolue.

Cette figure présente la fonction de transfert pour le bloc fonction FB_3points_Ext :

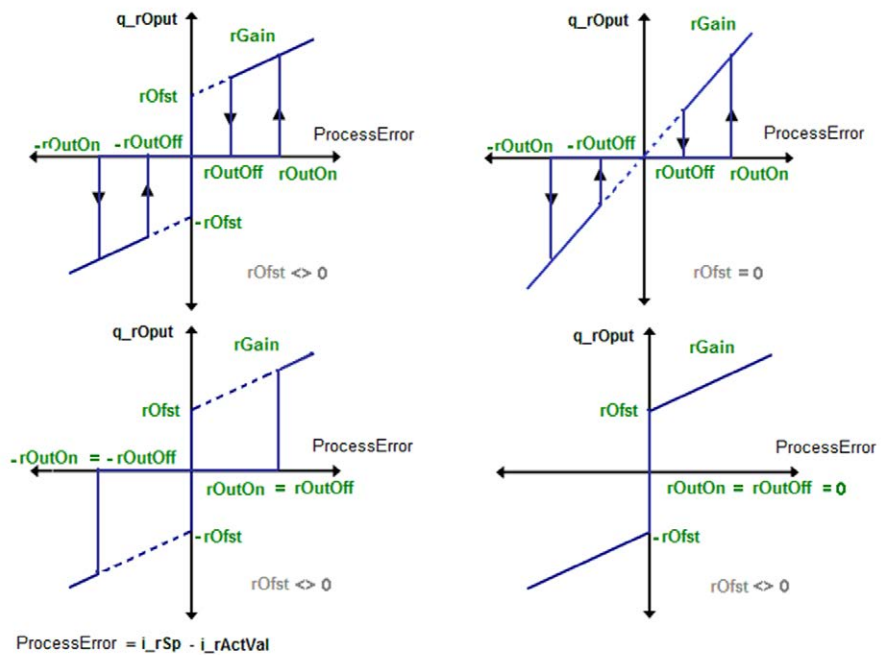
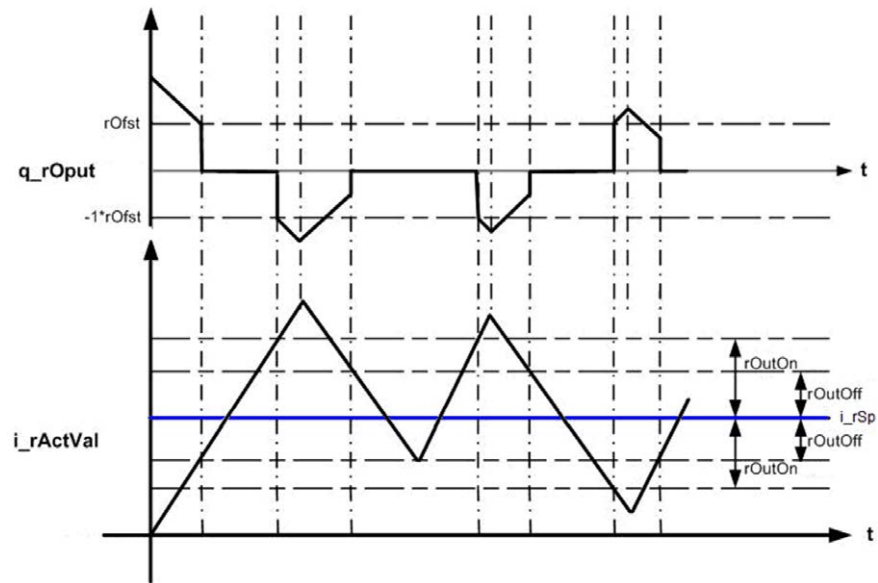


Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction FB_3points_Ext :



Etat d'erreur détectée

Un paramètre non valable à l'entrée du bloc fonction conduit à un état d'erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, les valeurs de sortie sont mises à zéro. L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$.

La sortie q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_3points_Ext :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction FALSE : Bloc fonction désactivé
i_rSp	REAL	Valeur de point de consigne du processus Plage : $\pm 3,4e^{+38}$
$i_rActVal$	REAL	Valeur réelle du processus Plage : $\pm 3,4e^{+38}$
$i_rManVal$	REAL	Valeur d'entrée pour le mode manuel. Plage : $\pm 3,4e^{+38}$ (optionnel)
$i_xManMode$	BOOL	TRUE : Bloc fonction en mode manuel

Entrée	Type de données	Description
		FALSE : Bloc fonction en mode automatique (optionnel)
i_xErrRst	BOOL	Réinitialiser l'erreur détectée (Déclenché par un front montant) (optionnel)
i_stPara	STRUCTst3PtsExtendedPara	Variable structure de bloc fonction

Structure utilisée

st3PtsExtendedPara

Élément de structure	Type	Description
rOutOn	REAL	Valeur du seuil supérieur Plage : 0...1e38
rOutOff	REAL	Valeur du seuil inférieur Plage : 0...1e38
rGain	REAL	Gain Plage : 0...1e38
rOfst	REAL	Décalage Plage : 0...1e38

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_3points_Ext :

Sortie	DataType	Description
q_xEn	BOOL	TRUE : FB activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Bloc fonction actif et pas d'erreur détectée. FALSE : Bloc fonction désactivé ou erreur détectée
q_rOput	REAL	Sortie du bloc fonction. Plage : $\pm 3,4e^{+38}$
q_xErr	BOOL	Signal d'erreur détectée

q_uiErrId

Cette valeur entière unique indique une erreur détectée spécifique :

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	Valeurs de paramètre non valables ($r_{OutOn} < 0$ OU $r_{OutOff} < 0$)
2	Valeurs de paramètre non valables ($r_{OutOn} < r_{OutOff}$)
3	Valeurs de paramètre non valables ($r_{Gain} < 0$)
4	Valeurs de paramètre non valables ($r_{Ofst} < 0$)
5	Erreur interne détectée (bloc fonction en état inconnu)

FB_P : Boucle de commande avec algorithme proportionnel seulement

Contenu de ce chapitre

Bloc fonction FB_P	42
Modes de fonctionnement.....	42
Description des broches d'entrée.....	44
Description des broches de sortie.....	44

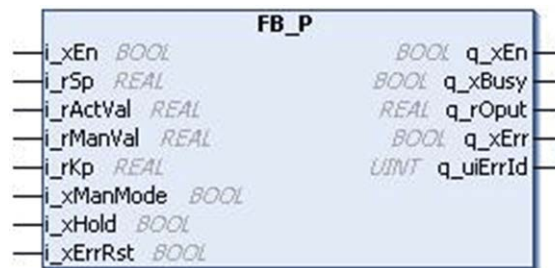
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_P.

Bloc fonction FB_P

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_P :



Description fonctionnelle

Ce bloc fonction FB_P est conçu pour fournir une boucle de commande avec algorithme proportionnel seulement.

Modes de fonctionnement

Mode automatique

Ce bloc fonction donne une réponse proportionnelle, c'est-à-dire que la sortie est l'erreur de processus multipliée par le gain.

$$G(s) = Kp$$

Cette équation présente la fonction de transfert :

Avec :

Kp	= Gain proportionnel
q_rOput	= G(s) * Erreur processus

Mode Manuel

La sortie du bloc fonction q_rOput est définie égale à $i_rManVal$.

Cette figure présente le diagramme fonctionnel du bloc fonction FB_P :

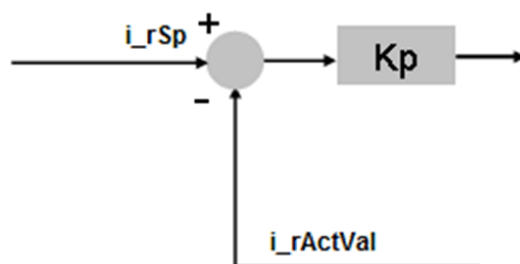
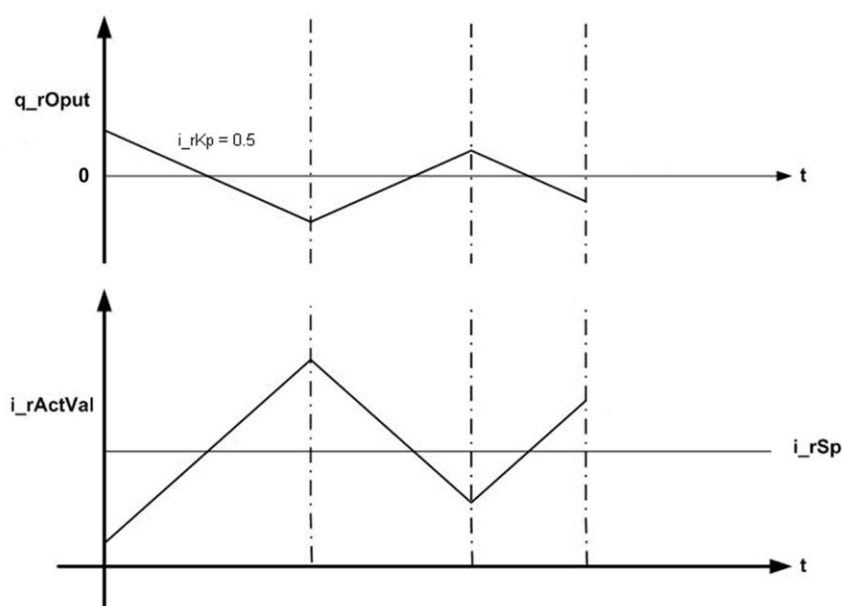


Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction FB_P :



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, les valeurs de sortie sont mises à zéro. L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$.

La sortie q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_P :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction FALSE : Bloc fonction désactivé
i_rSp	REAL	Valeur de point de consigne du processus Plage : $\pm 3,4e^{+38}$
i_rActVal	REAL	Valeur réelle du processus Plage : $\pm 3,4e^{+38}$
i_rManVal	REAL	Valeur d'entrée pour le mode manuel. Plage : $\pm 3,4e^{+38}$ (optionnel)
i_rKp	REAL	Gain proportionnel Plage : $0 \dots 3,4e^{+38}$
i_xManMode	BOOL	TRUE : Bloc fonction en mode manuel FALSE : Bloc fonction en mode automatique (optionnel)
i_xHold	BOOL	TRUE : Maintenir l'état interne et la sortie du bloc fonction constante à sa valeur en cours. FALSE : Désactivé (optionnel)
i_xErrRst	BOOL	Réinitialisation d'erreur détectée (déclenchée par un front montant). (optionnel)

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_P :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : Bloc fonction activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Bloc fonction actif et pas d'erreur détectée. FALSE : Bloc fonction désactivé ou erreur détectée
q_rOput	REAL	Sortie du bloc fonction Plage : $\pm 3,4e^{+38}$

Sortie	Type de données	Description
q_xErr	BOOL	Signal d'erreur détectée
q_uiErrId	UINT	Fournit l'ID d'erreur détectée quand la sortie q_xErr est active. 0: Aucune erreur détectée 1: Valeurs de paramètre non valables (si $i_{rKp} < 0$) 2: Erreur interne détectée (bloc fonction en état inconnu)

q_uiErrId

Cette valeur entière unique indique une erreur détectée spécifique

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	Valeurs de paramètre non valables ($i_{rKp} < 0$)
2	Erreur interne détectée (bloc fonction en état inconnu)

FB_PI: Commande de processus proportionnelle et intégrale

Contenu de ce chapitre

Bloc fonction <code>FB_PI</code>	46
Description des broches d'entrée	47
Structure utilisée	49
Description des broches de sortie	51

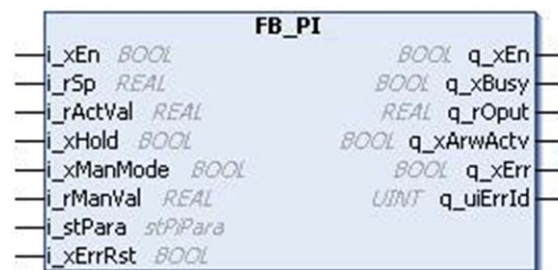
Vue d'ensemble

Ce chapitre décrit le bloc fonction de commande de processus en boucle fermée `FB_PI`.

Bloc fonction `FB_PI`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `FB_PI` :



Description fonctionnelle

Le bloc fonction `FB_PI` est un bloc fonction PI standard avec optimisation manuelle, fonction de maintien et saturation antiréinitialisation.

Le contrôleur PI génère une sortie de commande basée sur l'erreur de processus dans le système (Erreur de processus = Point de consigne – Valeur réelle). A l'aide des paramètres du bloc fonction, la sortie de commande peut être optimisée pour réduire l'erreur de processus.

Les valeurs proportionnelles et intégrales du processus sont calculées en continu à partir de la valeur réelle, du point de consigne et des paramètres d'entrée. Le bloc fonction limite aussi la sortie de commande en fonction des paramètres de limite.

Fonction de transfert

L'équation ci-dessous est la fonction de transfert du bloc fonction `FB_PI` :

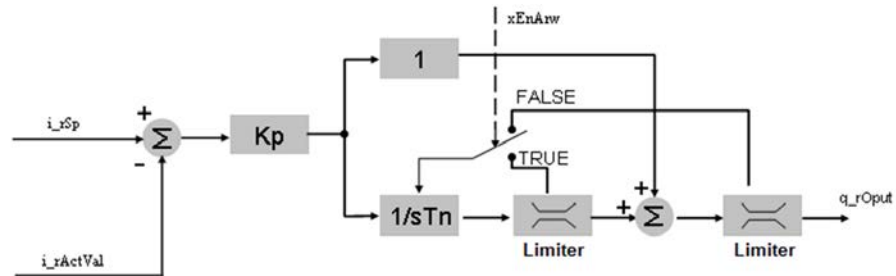
$$G(s) = K_p \left(1 + \frac{1}{sT_n} \right)$$

Où :

K_p	= Gain proportionnel.
sT_n	= Temps intégral

Schéma fonctionnel

Cette figure présente le schéma fonctionnel du bloc fonction FB_PI :



Ce bloc fonction permet de commander les processus en boucle fermée avec les variables d'entrée et de sortie continues.

Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction génère une erreur détectée et un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, la valeur de sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$. La sortie q_xBusy a la valeur TRUE, chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_PI :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : active le bloc fonction. FALSE : désactive le bloc fonction.
i_rSp	REAL	Valeur du point de consigne Plage : $\pm 3,4e^{+38}$
$i_rActVal$	REAL	Valeur réelle Plage : $\pm 3,4e^{+38}$
$i_rManVal$	REAL	Valeur manuelle Plage : $\pm 3,4e^{+38}$ (Facultatif)
$i_xManMode$	BOOL	Valeur manuelle (Facultatif)
i_xHold	BOOL	Maintien

Entrée	Type de données	Description
		(Facultatif)
i_xErrRst	BOOL	Réinitialise l'erreur détectée (le front montant réinitialise l'erreur détectée) (Facultatif)
i_stPara	STRUCT stPiPara	Paramètre de structure (Consultez la description de stPiPara, page 49.)

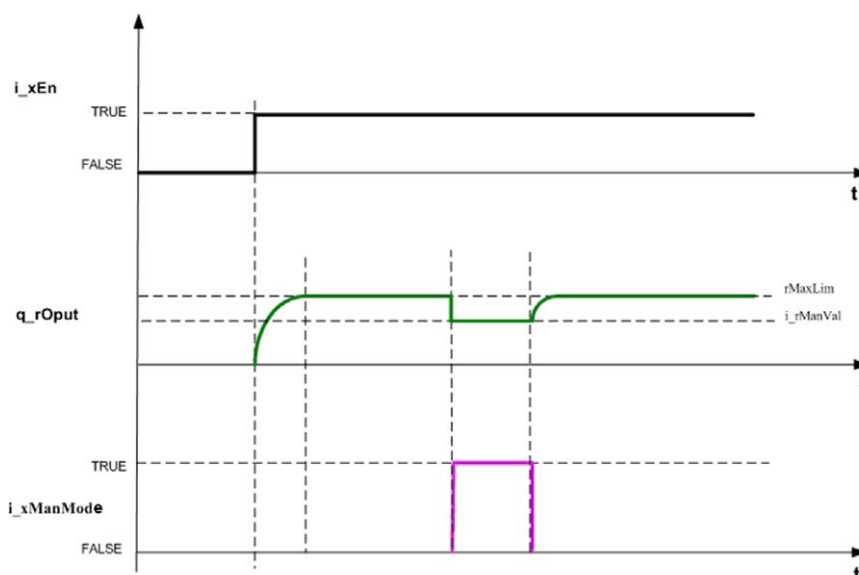
i_xManMode

i_xManMode détermine le mode manuel du bloc fonction FB_PI.

Si le bloc fonction est activé et que manuel a la valeur TRUE, le bloc fonction définit la valeur manuelle (i_rManVal) comme la sortie PI et arrête l'algorithme PI comme indiqué dans le schéma fonctionnel du bloc fonction en mode manuel.

Si le mode auto est activé, l'algorithme PI s'exécute en continu.

La figure ci-dessous présente le chronogramme du bloc fonction en mode manuel :



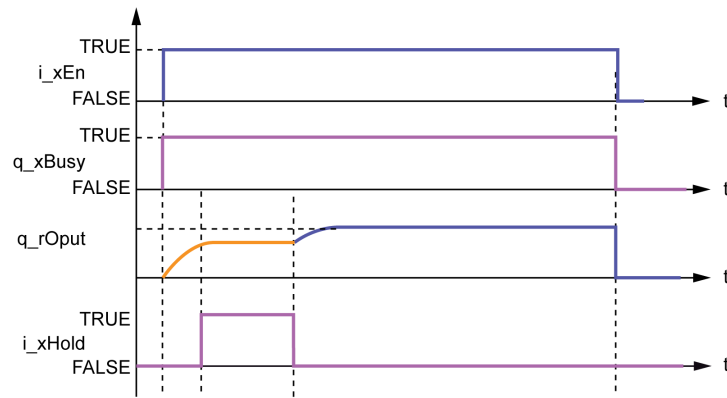
i_xHold

i_xHold maintient la sortie PI au niveau en cours.

Si cette entrée a la valeur TRUE, la sortie PI est maintenue à la dernière valeur et les calculs internes de l'algorithme PI sont arrêtés comme indiqué dans le schéma fonctionnel du bloc fonction en maintien.

Si cette entrée a la valeur FALSE, l'algorithme PI est exécuté cycliquement. La nouvelle sortie PI est calculée à partir de la dernière valeur.

Chronogramme du bloc fonction en pause :



Structure utilisée

stPiPara

Élément de structure	Type	Description
<i>tCyclTime</i>	TIME	Durée de cycle de tâche Plage : 10 ms...60 s
<i>xEnArw</i>	BOOL	Activer la saturation antiréinitialisation
<i>tTn</i>	TIME	Temps d'action intégrale Plage : 1...1e32 ms
<i>rKp</i>	REAL	Valeur de gain proportionnel Plage : ±3,4e+38
<i>rMaxLim</i>	REAL	Limite de sortie maximale Plage : ±3,4e+38
<i>rMinLim</i>	REAL	Limite de sortie minimale Plage : ±3,4e+38

tCyclTime

tCyclTime est le temps entre deux exécutions du bloc fonction. Si la tâche est définie comme cyclique, il est égal au temps de cycle de tâche de la tâche cyclique.

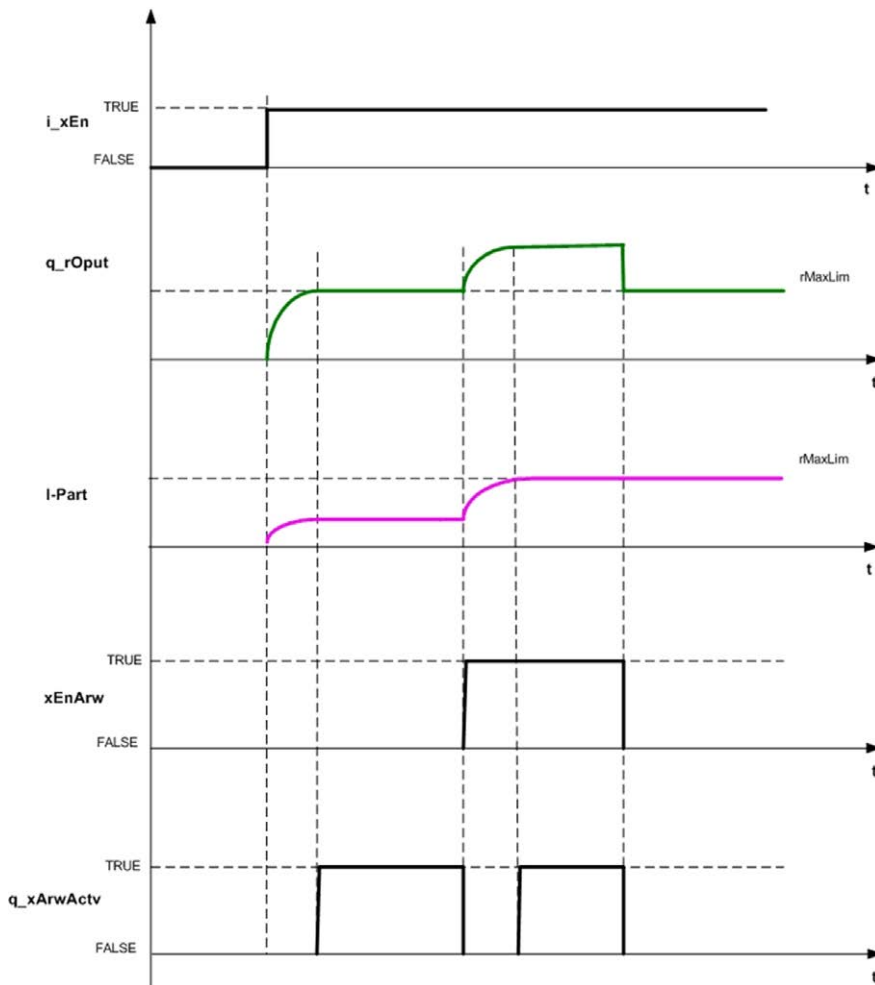
xEnArw

xEnArw active une opération de saturation antiréinitialisation (ARW).

Si la valeur est FALSE, maintient la partie intégrale si la sortie de commande complète atteint une limite.

Si la valeur est TRUE, le bloc fonction ne maintient que la partie intégrale si celle-ci atteint une limite. La sortie est égale à la somme de la valeur limite et de la partie proportionnelle, si la partie intégrale atteint une limite comme indiqué dans le schéma fonctionnel du bloc fonction en cas d'activation de ARW.

Cette figure présente le bloc fonction en cas d'activation du mode ARW :



tTn

Temps intégral de la boucle PI

rKp

Gain proportionnel de la boucle PI

rMaxLim

Une sortie supérieure à cette limite est limitée à la valeur rMaxLim.

rMinLim

Une sortie inférieure à cette limite est limitée à la valeur rMinLim.

NOTE: Si la valeur rMinLim est supérieure à 0, le fonctionnement de l'algorithme PI commence à la valeur rMinLim.

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_PI :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : bloc fonction activé
q_xBusy	BOOL	TRUE : bloc fonction actif et aucune erreur détectée. FALSE : bloc fonction désactivé ou ayant détecté une erreur
q_rOput	REAL	Sortie PI calculée Plage : $\pm 3,4e^{+38}$
q_xArwActv	BOOL	TRUE : sortie limitée, saturation antiréinitialisation active.
q_xErr	BOOL	Signal d'erreur détectée
q_uiErrId	UNIT	ID de l'erreur détectée Plage : 0 à 5

q_uiErrId

Cette valeur entière unique identifie l'erreur détectée.

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	Temps de cycle de tâche non valable = zéro
2	Paramètre non valable $i_rOputMaxLim < i_rOputMinLim$
3	Paramètre non valable $rKp < \text{zéro}$
4	Paramètre non valable $tTn = \text{zéro}$
5	Erreur interne détectée (bloc fonction en état inconnu)

FB_PID : Commande de processus de fonctionnement en mode manuel

Contenu de ce chapitre

Bloc fonction FB_PID	52
Description des broches d'entrée	55
Structure utilisée	57
Description des broches de sortie	58
Instanciation et exemple d'utilisation	61

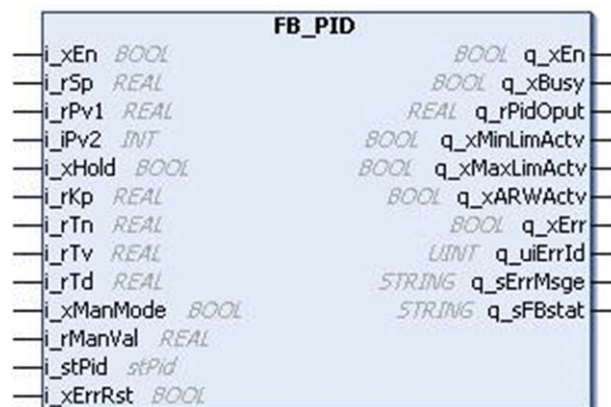
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_PID.

Bloc fonction FB_PID

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_PID :



Description fonctionnelle

Le bloc fonction FB_PID est un bloc fonction PID standard avec optimisation manuelle, fonction de maintien, transfert sans à-coup et temps d'amortissement pour l'action dérivée.

Ce bloc fonction dispose des fonctionnalités suivantes :

- Différents modes : P, PI, PD et PID.
- Fonctionnement en mode manuel pour commander la sortie PID en mode manuel.
- Saturation antiréinitialisation pour éviter la saturation dans l'action intégrale : si la variable de commande atteint la limite de l'actionneur, l'erreur du processus continue à intégrer un terme intégral très important (appelé saturation).
- Temps d'amortissement (Td) pour filtrer le dépassement de cible dû à l'action dérivée.

- Le transfert sans à-coup est activé quand le mode passe de manuel à auto.
Le transfert sans à-coup évite des variations brutales de la sortie PID aux changements de mode.
- L'état d'erreur détectée est généré par un bloc fonction pour afficher les erreurs détectées.
- Les fonctions de fenêtre intérieure et extérieure sont utilisées dans les calculs d'intégrale.

Si la valeur absolue de l'erreur de processus est inférieure à la fenêtre intérieure, la partie intégrale est dimensionnée selon le facteur $[\text{ABS}(\text{err}) / \text{Fenêtre intérieure}]$.

Ceci permet de réduire le dépassement de cible dans la sortie PID.

Si la valeur absolue de l'erreur de processus est supérieure à la fenêtre intérieure, mais inférieure à la fenêtre supérieure, les calculs d'intégrale normaux sont effectués.

Si la valeur absolue de l'erreur de processus est supérieure à la fenêtre extérieure, la saturation antiréinitialisation est active et la sortie intégrale maintient la dernière valeur.

Sortie PID

L'équation ci-dessous présente la sortie PID :

$$y(t) = K_p \left(e(t) + \frac{1}{T_n} \int e(t) dt + \frac{T_v}{1 + T_d} \frac{de(t)}{dt} \right)$$

Où :

y (t)	= Sortie PID
Kp	= Gain proportionnel
Tn	= Temps d'intégrale
Tv	= Temps de dérivée
Td	= Temps de filtre de la dérivée
e (t)	= Erreur de processus entre le point de consigne et la valeur de rétroaction.

Schéma de principe

Cette figure présente le schéma fonctionnel du bloc fonction FB_PID :

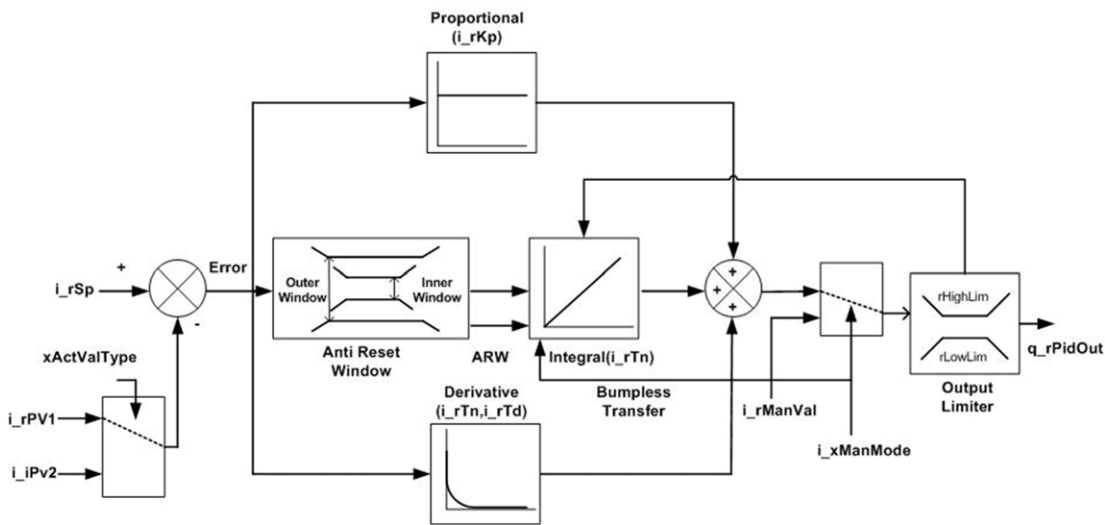


Schéma de comportement normal

La figure présente le schéma de comportement normal du bloc fonction FB_PID :

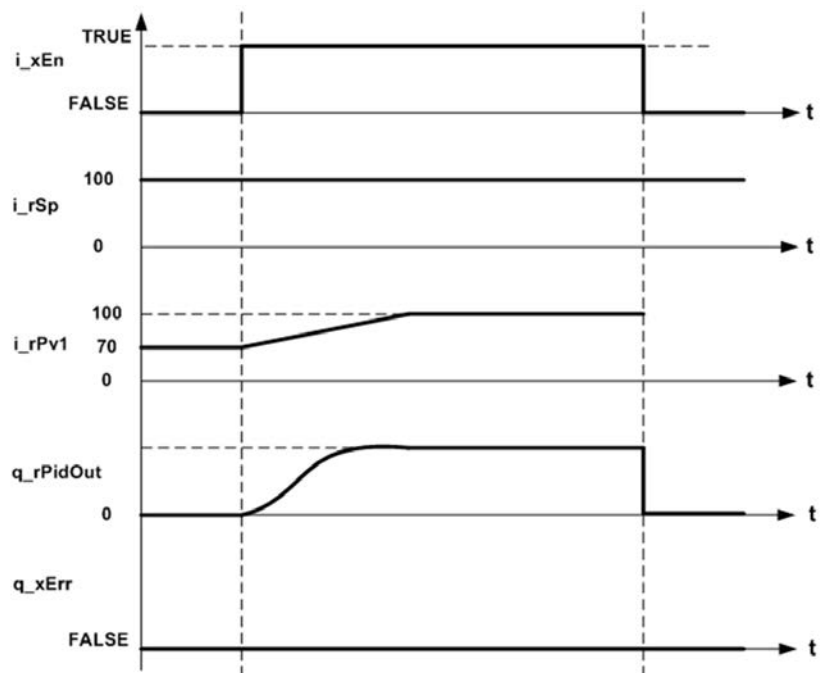
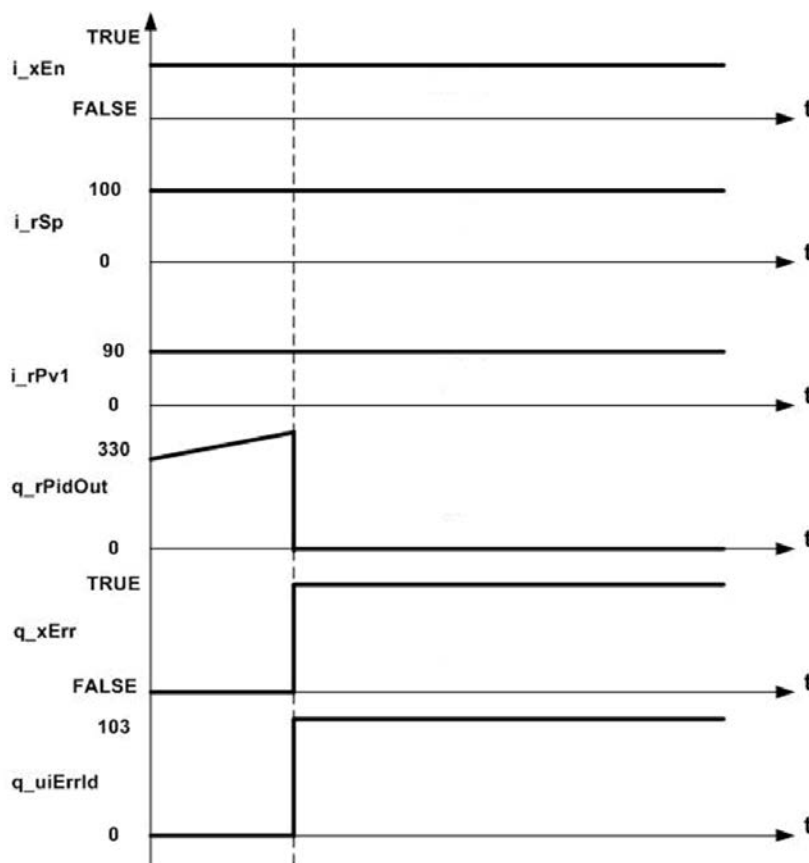


Schéma d'erreur détectée

Cette figure présente le schéma de bloc fonction FB_PID avec erreur détectée :



Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_PID :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : active le bloc fonction FB_PID. FALSE : désactive le bloc fonction FB_PID.
i_rSp	REAL	Point de consigne Plage : $\pm 3,4e^{+38}$
i_rPv1	REAL	Valeur de rétroaction du processus. Plage : $\pm 3,4e^{+38}$
i_iPv2	INT	Valeur de rétroaction du processus. Plage : -32768 à 32767
i_xHold	BOOL	TRUE : maintient la sortie PID au niveau en cours et arrête les calculs du PID. FALSE : fonctionnement normal du mode PID. (Facultatif)
i_rKp	REAL	Gain proportionnel du PID Plage : 0,0 à $3,4e^{+38}$

Entrée	Type de données	Description
i_rTn	REAL	Temps intégral du PID Plage : 0,0 à 60 000 millisecondes
i_rTv	REAL	Temps de dérivation de la commande PID Plage : 0,0 à 60 000 millisecondes
i_rTd	REAL	Temps d'amortissement de l'action de dérivation Plage : $60\,000,0 > i_rTd > i_stPid.rTargCyclTime$
i_xManMode	BOOL	TRUE : mode manuel FALSE : mode auto (réglage d'usine) (Facultatif)
i_rManVal	REAL	Valeur de sortie du PID manuel quand i_xManMode a la valeur TRUE. Plage : $\pm 3,4e^{+38}$ (Facultatif)
i_stPid	STRUCT st_Pid	Structure de paramètre (Consultez la description de , page 57 i_stPid.)
i_xErrRst	BOOL	TRUE : réinitialiser l'erreur détectée (Facultatif) (Le front montant réinitialise l'erreur détectée)

i_xEn

La valeur TRUE active le bloc fonction.

Si la valeur est FALSE, la sortie du PID est mise à zéro, l'état d'erreur détectée (q_xErr) est effacé et l'ID d'erreur détectée (q_uiErrId) est mis à zéro.

i_xHold

Si la valeur est TRUE, la sortie du PID est maintenue à la dernière valeur et les calculs internes de l'algorithme PID sont arrêtés.

Si la valeur est FALSE, l'algorithme du PID est exécuté cycliquement.

La nouvelle sortie PID est calculée à partir de la dernière valeur.

i_xErrRst

Ceci réinitialise l'erreur détectée en cours.

L'erreur détectée est réinitialisée sur le front montant de cette entrée.

La sortie PID est mise à zéro.

i_xManMode

Si la valeur est TRUE, la sortie PID est égale à i_rManVal.

Si la valeur est FALSE, selon le transfert sans à-coups, la sortie PID commence à $i_rManVal$ et l'algorithme PID commence à s'exécuter.

Structure utilisée

stPid

Élément de structure	Type	Description
<i>xActValType</i>	BOOL	TRUE : i_rPv1 est la valeur de processus. FALSE : i_rPv2 est la valeur de processus.
<i>rDbnd</i>	REAL	Valeur de plage morte pour le calcul d'erreur de processus interne. Plage : 0,0...100,0 (Facultatif)
<i>rTargCyclTime</i>	REAL	Temps de cycle de la cible Plage : 60000,0 > $rTargCyclTime$ > 0,0 Unité : ms
<i>rLowLim</i>	REAL	Limite inférieure de la sortie PID Plage : $\pm 3,4e^{+38}$
<i>rHighLim</i>	REAL	Limite supérieure de la sortie PID Plage : $\pm 3,4e^{+38}$
<i>rInerWndo</i>	REAL	Fenêtre intérieure pour la partie intégrale réduite. Plage : 0,0... $3,4e^{+38}$
<i>rOterWndo</i>	REAL	Fenêtre extérieure pour la désactivation de la partie intégrale. Plage : 0,0... $3,4e^{+38}$

NOTE: *xActValType*, *rLowLim*, *rHighLim*, *rInerWndo*, *rOterWndo* acceptent le nouvel état ou le changement de valeur sur le front montant de i_xEn .

xActValType

Sélectionne la valeur de processus.

rDbnd

La valeur de plage morte est utilisée pour le calcul d'erreur de processus.

Si la valeur absolue de l'erreur de processus est supérieure à la plage morte, la valeur de l'erreur de processus est calculée comme suit : Erreur de processus = Point de consigne - Valeur de rétroaction réelle.

Si la valeur absolue de l'erreur de processus est inférieure à la plage morte, l'erreur de processus a la valeur 0,0.

rTargCyclTime

Temps de cycle de la cible du PID

Le temps de cycle peut être mesuré et entré sous la forme d'une valeur fixe ou le temps de cycle courant peut être mesuré à chaque scrutation du contrôleur en tant que temps de cycle cible.

La valeur 0 du paramètre *stPid.rTargCyclTime* indique une erreur détectée avec ID20 : Temps de cycle non valide

Une valeur du paramètre *stPid.rTargCyclTime* supérieure au temps de vidage dérivé (Td) conduit à une erreur détectée avec ID201 : Paramètre Td incorrect

rLowLim

Limite inférieure de la sortie PID.

Si la sortie PID interne est inférieure à *rLowLim*, la sortie PID est fixée à *rLowLim*.

rHighLim

Limite supérieure de la sortie PID

Si la sortie PID interne est supérieure à *rHighLim*, la sortie PID est fixée à *rHighLim*.

rInerWndo

Si la valeur absolue de l'erreur de processus est inférieure à *rInerWndo*, les calculs d'intégrale sont mis à l'échelle selon le facteur [ABS (erreur détectée) / Fenêtre intérieure].

L'erreur de processus est la différence entre le point de consigne et la valeur réelle du processus.

Si l'erreur de processus est supérieure à *rInerWndo* et inférieure à *rOterWndo*, les calculs d'intégrale normaux sont effectués.

Cette fenêtre permet de réduire le dépassement de cible de la sortie PID.

Erreur de processus = Point de consigne - Valeur de processus

rOterWndo

Si la valeur absolue de l'erreur de processus est supérieure à *rOterWndo*, les calculs d'intégrale sont arrêtés et la sortie intégrale est maintenue à la dernière valeur.

Dans la sortie PID, la contribution maximale est due à l'action intégrale.

Si l'erreur de processus ne se réduit pas malgré une variation de la sortie PID, la sortie PID se sature suite à la saturation de l'action intégrale.

Pour éviter cet effet, le paramètre *rOterWndo* est indispensable.

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_PID :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : bloc fonction activé. FALSE : bloc fonction désactivé.
q_xBusy	BOOL	TRUE : le PID est actif et aucune erreur interne n'est détectée. FALSE : le PID n'est pas actif ou une erreur est détectée.
q_rPidOput	REAL	Sortie du contrôleur PID. Plage : i_stPid.rLowLim à i_stPid.rHighLim
q_xMinLimActv	BOOL	TRUE : si la sortie du PID est inférieure ou égale à i_stPid.rLowLim (limite minimale). FALSE : si la sortie du PID est supérieure à i_stPid.rLowLim (limite minimale).
q_xMaxLimActv	BOOL	TRUE : si la sortie du PID est supérieure ou égale à i_stPid.rHighLim (limite maximale). FALSE : si la sortie du PID est inférieure à i_stPid.rHighLim (limite maximale).
q_xARWActv	BOOL	TRUE : la saturation antiréinitialisation est active. FALSE : la saturation antiréinitialisation n'est pas active.
q_xErr	BOOL	TRUE : le bloc fonction a détecté une erreur FALSE : aucune erreur détectée.
q_uiErrId	UNIT	Indique le numéro d'erreur détectée quand la sortie d'erreur détectée est active. Plage : 0, 100, 103, 104, 105, 106, 107
q_sErrMsge	STRING	Message d'erreur détectée
q_sFBstat	STRING	Etat du bloc fonction

q_xEn

True pour indiquer l'état d'activation du bloc fonction.

q_xBusy

TRUE si le bloc fonction est démarré sans erreur détectée.

Si une erreur détectée survient, la sortie occupée passe à l'état bas.

q_rPidOput

Le bloc fonction FB_PID calcule la sortie à chaque cycle s'il n'y a pas d'erreur détectée.

q_xARWActv

Indique l'état de la saturation antiréinitialisation.

Si le temps d'intégrale est supérieur à zéro, l'action intégrale est active.

S'il est TRUE, l'action intégrale est l'arrêt et la sortie intégrale est le maintien à la dernière valeur.

TRUE :

- Cas 1 : temps intégral > 0,0 et ((Sortie PID >= limite maximale) et (Erreur processus > 0,0))
- Cas 2 : temps intégral > 0,0 et ((Sortie PID <= limite minimale) et (Erreur processus < 0,0))

q_xErr

TRUE indique l'erreur détectée et la sortie du PID est mise à zéro.

q_uiErrId et q_sErrMsg

Ceci donne le numéro d'erreur détectée et le message d'erreur détectée quand q_xErr a la valeur TRUE.

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	Erreur interne détectée
20	Temps de cycle non valable
114	Paramètre de limite non valable
115	Limite de bande morte non valable
200	Paramètre PID incorrect
201	Paramètre Td incorrect
202	Fenêtre I incorrecte

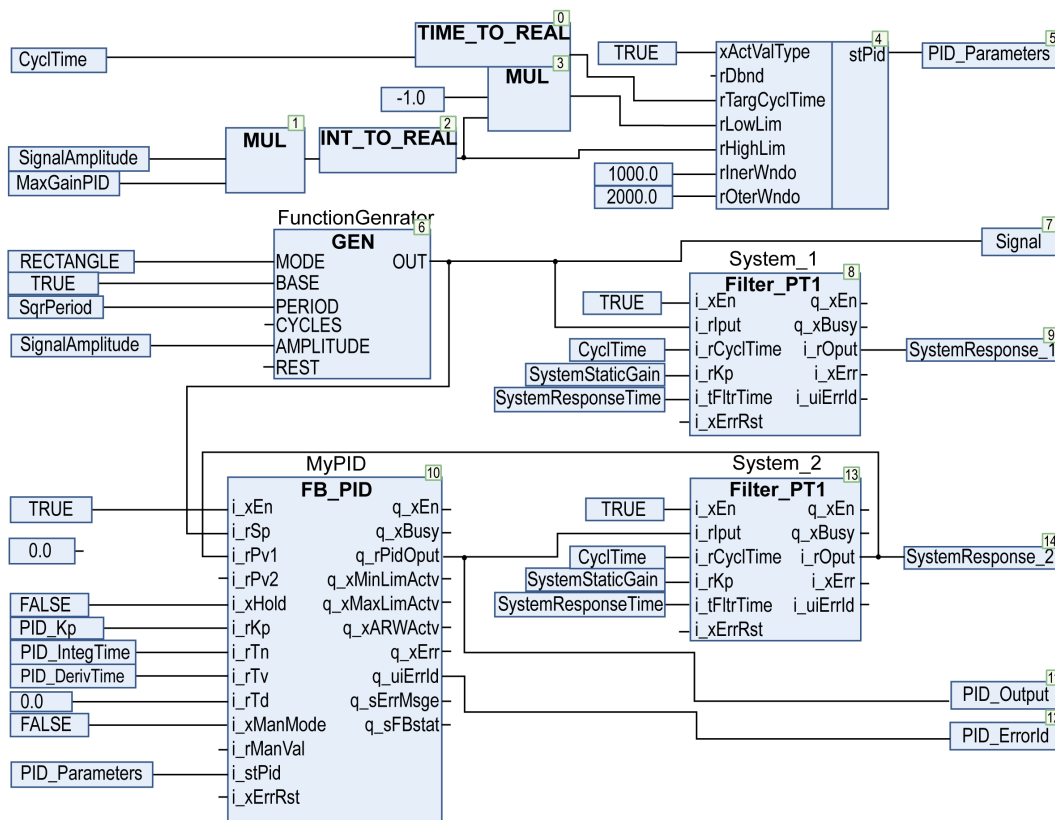
q_sFBstat

Bloc fonction actif :	Le bloc fonction est actif et s'exécute sans erreur détectée.
Erreur détectée dans le bloc fonction :	Le bloc fonction est actif et une erreur est détectée.
Bloc fonction désactivé :	Le bloc fonction est désactivé.

Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction FB_PID :



- Un signal carré est généré à l'aide du GEN, les paramètres essentiels sont SqrPeriod et SignalAmplitude.
- Le système à commander est un filtre simple de premier ordre, les paramètres essentiels sont SystemResponseTime et SystemStaticGain.
- Un tracé est effectué en boucle ouverte avec SystemResponse_1 et en boucle fermée à l'aide du bloc fonction FB_PID.

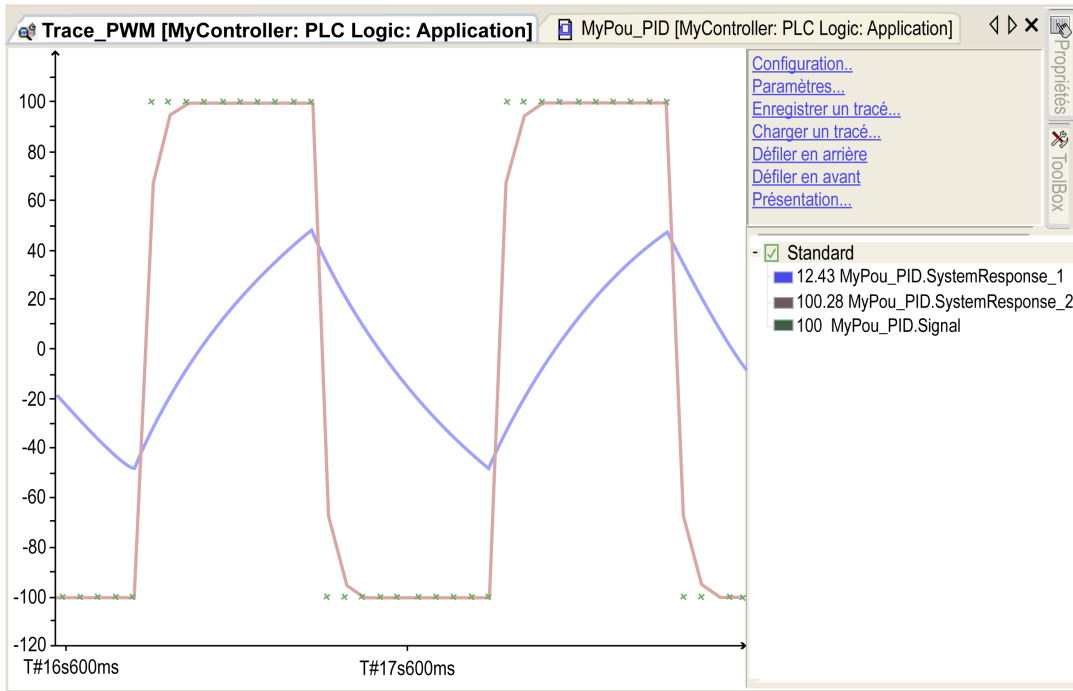
Les données de cet exemple sont :

```

Trace_PWM [MyController: PLC Logic: Application] | MyPou_PID [MyController: PLC Logic: Application]
1 PROGRAM MyPou_PID
2 VAR
3   FlagDemarrer      : BOOL;
4   FunctionGenerator : GEN;
5   SqrPeriod         : TIME := T#1000MS;
6   SignalAmplitude  : INT := 100;
7   Signal           : INT;
8   System_1        : Filter_PT1;
9   System_2        : Filter_PT1;
10  CyclTime        : TIME := T#50MS; (* Devrait avoir la même valeur que la périodicité du POU dans le MAST*)
11  SystemStaticGain : REAL := 1.0; TIME:=T#5000MS;
12  SystemResponseTime : TIME := T#500MS;
13  SystemResponse_1 : REAL;
14  SystemResponse_2 : REAL;
15
16  MyPID           : FB_PID;
17  PID_Parameters  : stPid;
18  PID_Kp         : REAL := 7.5;
19  PID_IntegTime  : REAL := 44.0; TIME:=T#5000MS;
20  PID_DerivTime  : REAL := 0.0;
21  PID_ErrorId    : UINT;
22  PID_Output     : REAL;
23  MaxGainPID     : INT := 20;
24 END_VAR

```

Avec le paramètre précédent, la réponse de point de consigne/boucle ouverte/boucle fermée est :



L'entrée `i_tCyclTime` des filtres de premier ordre `System_1` et `System_2` (`dataCyclTime`) doivent avoir exactement la même valeur que la période du POU dans la tâche MAST, ici 50 millisecondes.

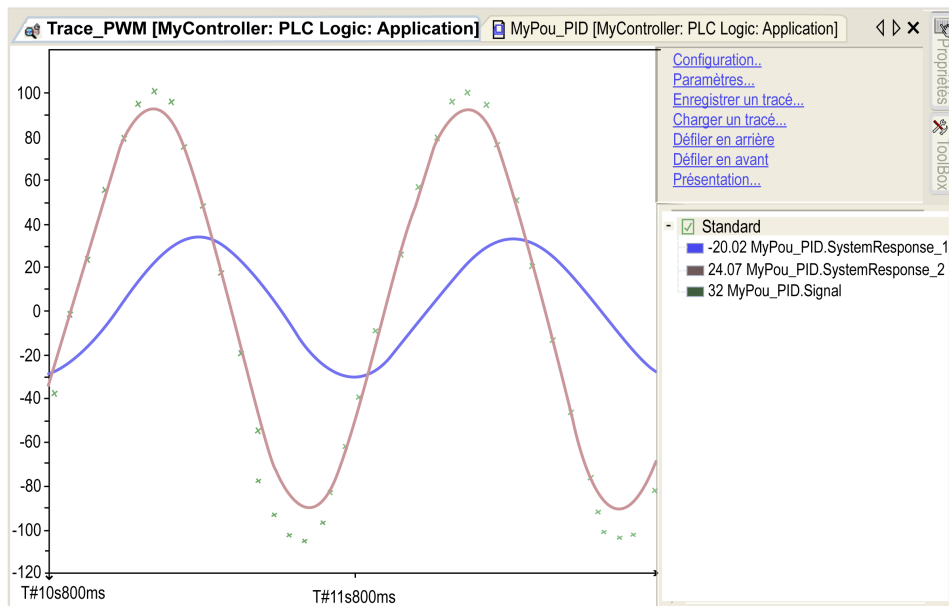
Priorité (0...31) : 15

Type: Cyclique, Intervalle (p.ex. t#200ms) : 50 ms

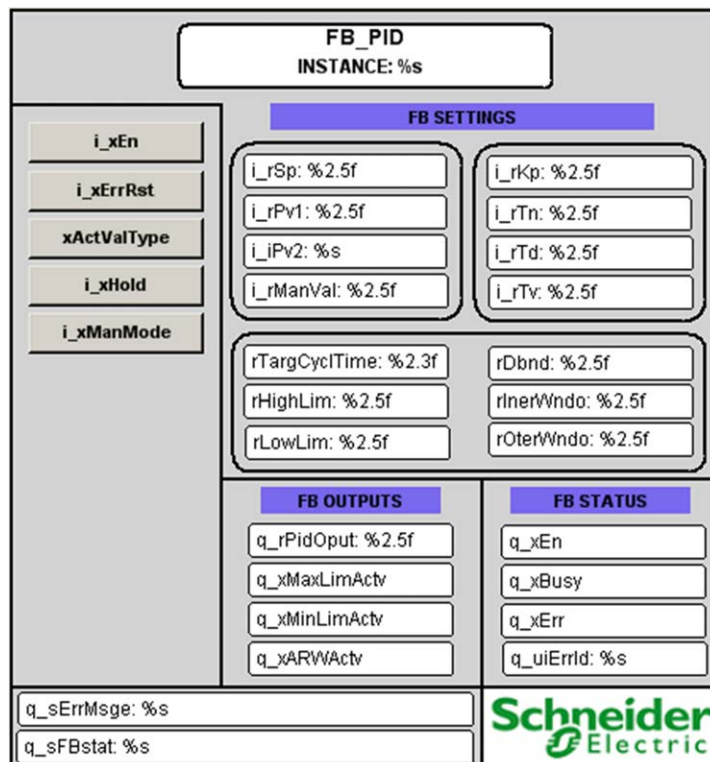
Chien de garde: Activer, temps (p.ex. t#200ms) : 1000 ms, Sensibilité : 1

POU	Commentaire
MyPou_PID	

Quand GEN MODE passe de RECTANGLE à SINUS avec les autres paramètres identiques, la réponse de Sinus est :



Cette figure présente la visualisation du bloc fonction FB_PID :



Etat d'erreur détectée

Ce tableau décrit certaines erreurs détectées générales :

Problème	Cause	Solution
Etat d'erreur détectée	Paramètre d'entrée non valable	Entrez un paramètre valable, puis réinitialisez l'erreur détectée

NOTE: Si le bloc fonction est désactivé, les sorties sont mises à zéro.

FB_PI_PID : Boucle de commande en cascade PI_PID

Contenu de ce chapitre

Bloc fonction FB_PI_PID	64
Modes de fonctionnement.....	64
Description des broches d'entrée.....	66
Structures utilisées.....	68
Description des broches de sortie.....	68

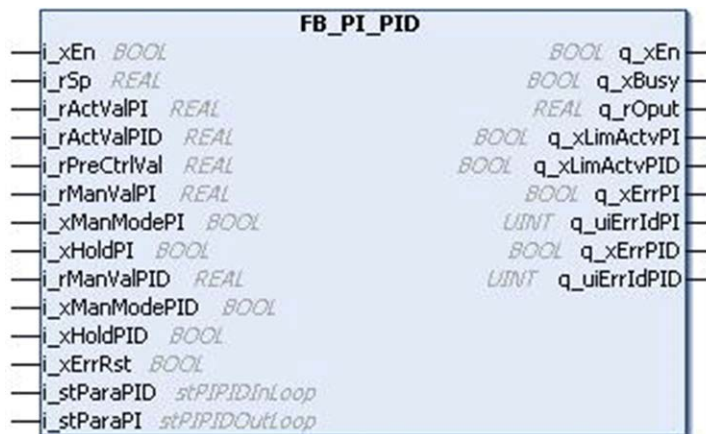
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_PI_PID.

Bloc fonction FB_PI_PID

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_PI_PID :



Description fonctionnelle

Le bloc fonction FB_PI_PID donne un fonctionnement en cascade de FB_PI, FB_Limiter et FB_PID.

Ce bloc fonction est constitué d'un élément PI, d'un limiteur de contrôle et d'un élément PID.

Modes de fonctionnement

Mode automatique

Le bloc fonction calcule la réponse PI pour le point de consigne i_rSp et la valeur réelle de la boucle extérieure i_rActValPI. Cette réponse PI est ajoutée à la valeur de précommande i_rPreCtrlVal et limitée par les entrées de seuil maximale et minimale, et sert de point de consigne pour la boucle PID interne.

La boucle interne calcule une réponse PID avec *i_rActValPID* comme valeur réelle de boucle interne.

Cette équation présente la fonction de transfert de l'élément PI :

$$G(s) = Kp(1 + \frac{1}{sT_n})$$

Avec :

K_p	= Gain proportionnel
T_n	= Temps d'intégrale

Cette équation présente la fonction de transfert de l'élément PID :

$$G(s) = Kp(1 + \frac{1}{sT_n} + \frac{sT_v}{1 + sT_d})$$

Avec

K_p	= Gain proportionnel
T_n	= Temps d'intégrale

Mode automatique :

T_d	= Temps de dérivée
T_v	= Temps de filtrage

Mode Manuel

La boucle PI et la boucle PID peuvent être configurées individuellement en mode manuel par les broches d'entrée *i_xManModePI* et *i_xManModePID* respectivement. En mode manuel, la sortie de la boucle PI et les sorties de boucle PID sont remplacées par des valeurs sur les broches d'entrée *i_rManValPI* et *i_rManValPID* respectivement.

Cette figure présente la fonction de transfert pour le bloc fonction FB_PI_PID :

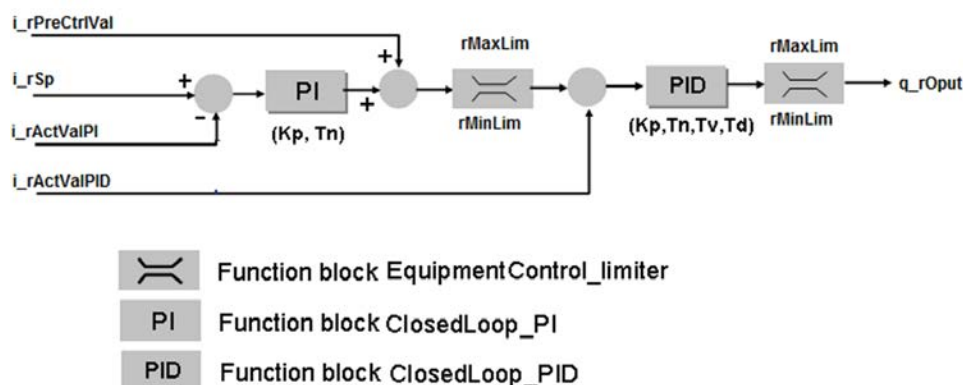
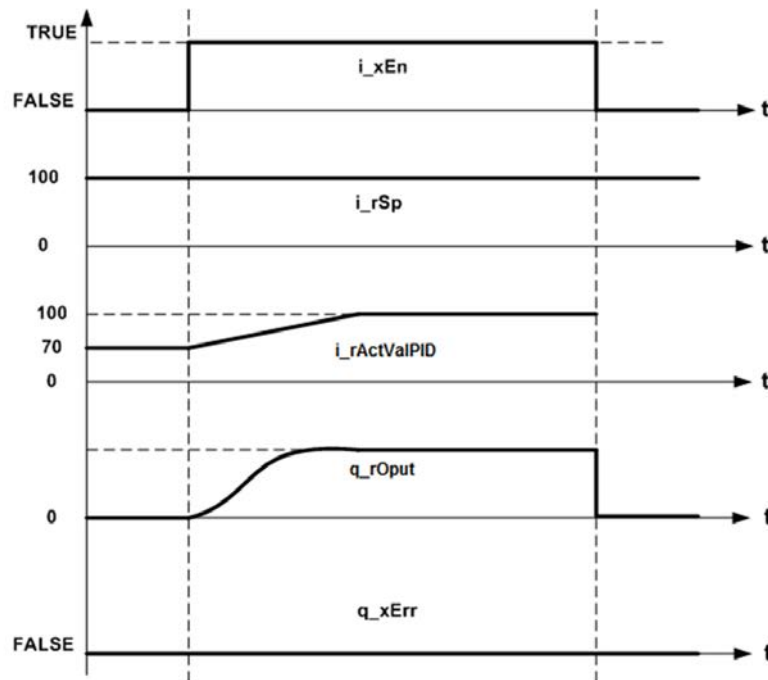


Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction FB_PI_PID :



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une d'erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, les valeurs de sortie sont mises à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée i_xErrRst.

La sortie q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Tableau des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_PI_PID :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction FALSE : Désactive le bloc fonction
i_rSp	REAL	Valeur de point de consigne du processus Plage : $\pm 3,4e^{+38}$
i_rActValPI	REAL	Valeur réelle du processus vers la boucle extérieure PI Plage : $\pm 3,4e^{+38}$
i_rActValPID	REAL	Valeur réelle du processus vers la boucle intérieure PID

Entrée	Type de données	Description
		Plage : $\pm 3,4e^{+38}$
i_rPreCtrlVal	REAL	Valeur de précontrôle ajoutée à la sortie de la boucle extérieure PI Plage : $\pm 3,4e^{+38}$
i_rManValPI	REAL	Entrée manuelle pour la boucle extérieure PI Plage : $\pm 3,4e^{+38}$ (optionnel)
i_xManModePI	BOOL	TRUE : Fonctionnement de la boucle extérieure PI en mode manuel. FALSE : Fonctionnement de la boucle extérieure PI en mode auto (optionnel)
i_xHoldPI	BOOL	TRUE : Maintenir la sortie de la boucle extérieure PI et l'état interne constant FALSE : Désactivé (optionnel)
i_rManValPID	REAL	Entrée manuelle pour la boucle intérieure PID Plage : $\pm 3,4e^{+38}$ (optionnel)
i_xManModePID	BOOL	TRUE : Fonctionnement de la boucle intérieure PID en mode manuel. FALSE : Fonctionnement de la boucle intérieure PID en mode auto (optionnel)
i_xHoldPID	BOOL	TRUE : Maintenir la sortie de la boucle intérieure PID et l'état interne constant FALSE : Désactivé (optionnel)
i_xErrRst	BOOL	Réinitialiser l'erreur détectée (Le front montant réinitialise l'erreur détectée). (optionnel)
i_stParaPID	STRUCT stPIPIDOutLoop	Paramètres de commande de la boucle intérieure PID
i_stParaPI	STRUCT stPIPIDInLoop	Réinitialiser l'erreur détectée

Structures utilisées

stPIPIDOutLoop

Élément de structure	Type	Description
<i>rKp</i>	REAL	Gain proportionnel Plage : 0,0...1e ³⁸
<i>tTn</i>	TIME	Temps d'intégrale Plage : 0...60000 ms
<i>tTv</i>	TIME	Temps de dérivée Plage : 0...60000 ms
<i>tTd</i>	TIME	Temps de filtrage Plage : 0...60000 ms
<i>rMaxLim</i>	REAL	Limite de sortie maximale pour la boucle PID intérieure Plage : ±3,4e ³⁸
<i>rMinLim</i>	REAL	Limite de sortie maximale pour la boucle PID intérieure Plage : ±3,4e ³⁸
<i>rInerWndo</i>	REAL	Fenêtre intérieure pour la partie I réduite. Plage : ±3,4e ³⁸
<i>rOterWndo</i>	REAL	Fenêtre extérieure pour la partie I désactivée. Plage : ±3,4e ³⁸

stPIPIDInLoop

Élément de structure	Type	Description
<i>tCyclTime</i>	TIME	Durée de cycle de tâche. Plage : 1...60000 ms
<i>rKp</i>	REAL	Gain proportionnel Plage : 0...1e ³⁸
<i>tTn</i>	TIME	Temps d'intégrale Plage : 0...60000 ms
<i>rMaxLim</i>	REAL	Limite de sortie maximale pour la boucle PI extérieure Plage : ±3,4e ³⁸
<i>rMinLim</i>	REAL	Limite de sortie minimale pour la boucle PI extérieure Plage : ±3,4e ³⁸

Description des broches de sortie

Tableau des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_PI_PID :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : Le bloc fonction est activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Le bloc fonction est actif et aucune erreur n'est détectée. FALSE : Le bloc fonction est désactivé ou une erreur est détectée
q_rOput	REAL	Sortie de la boucle PID en cascade PI. Plage : $\pm 3,4e^{+38}$
q_xLimActvPI	BOOL	TRUE : La sortie de la boucle extérieure PI est limitée FALSE : La sortie de la boucle extérieure PI n'est pas limitée
q_xLimActvPID	BOOL	TRUE : La sortie de la boucle intérieure PID est limitée FALSE : La sortie de la boucle intérieure PID n'est pas limitée
q_xErrPI	BOOL	Erreur détectée dans la boucle PI
q_uiErrIdPI	UNITE	Affiche l'ID de l'erreur détectée pour la boucle PI quand q_xErrPI prend la valeur TRUE Plage : 0...4
q_xErrPID	BOOL	Erreur détectée dans la boucle PID
q_uiErrIdPID	UNITE	Affiche l'ID de l'erreur détectée pour la boucle PID quand q_xErrPID prend la valeur TRUE Plage : 0...4

q_uiErrIdPI

Cette valeur entière unique indique une erreur détectée spécifique :

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	i_stParaPI.tCyclTime hors plage
2	i_stParaPI.rMaxLim < i_stParaPI.rMinLim
3	i_stParaPI.rKp inférieur à zéro
4	i_stParaPI.tTn hors plage

q_uiErrIdPID

Cette valeur entière unique indique une erreur détectée spécifique :

ID d'erreur détectée	Description
0	Aucune erreur détectée
1	i_stParaPID.tTv hors plage ou i_stParaPID.tTn hors plage ou i_stParaPID.tTd hors plage ou i_stParaPID.rKp inférieur à zéro.
2	i_stParaPID.tTd < (i_stParaPI.tCyclTime / 2)

ID d'erreur détectée	Description
3	i_stParaPID.rMaxLim < i_stParaPID.rMinLim
4	i_stParaPID.rOterWndo < i_stParaPID.rInerWndo ou i_stParaPID.rOterWndo < 0 ou i_stParaPID.rInerWndo < 0

Fonctions de commande d'équipement

Contenu de cette partie

FB_Cyclic_Monitoring : Surveillance cyclique	72
FB_DeadBand: Suppression d'oscillations d'amplitude	76
FB_Limiter : Limitation des signaux d'entrée.....	79
FB_PWM : Fourniture d'une sortie PWM	82
FB_Redundant_Sensor_Monitoring : Surveillance de capteur redondant.....	88
FB_Scaling : Mise à l'échelle des signaux d'entrée.....	93
FB_Sensor_Monitoring : Surveillance de capteur.....	97

Présentation

Cette partie décrit les fonctionnalités et la mise en œuvre des blocs fonction de commande d'équipement.

FB_Cyclic_Monitoring : Surveillance cyclique

Contenu de ce chapitre

Bloc fonction <code>FB_Cyclic_Monitoring</code>	72
Description des broches d'entrée	73
Description des broches de sortie	74
Instanciation et exemple d'utilisation	74

Vue d'ensemble

Ce chapitre explique le bloc fonction `FB_Cyclic_Monitoring`.

Bloc fonction `FB_Cyclic_Monitoring`

Schéma d'affectation des broches

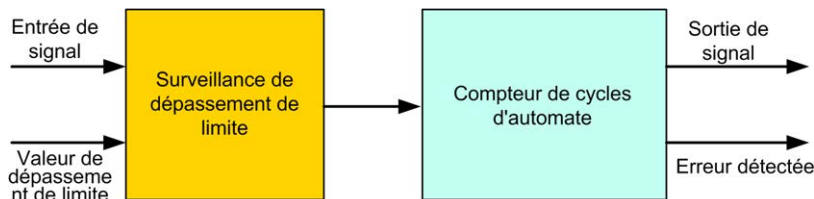
La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `FB_Cyclic_Monitoring`:



Description fonctionnelle

Le bloc fonction `FB_Cyclic_Monitoring` surveille le signal d'entrée pour y détecter une valeur maximale (pourcentage de la valeur maximale absolue), sur un nombre prédéfini de cycles d'automate avant détection d'une limite de dépassement empêchant le fonctionnement.

Ce bloc fonction permet de surveiller un signal d'entrée réel et de transférer le signal d'entrée à la sortie seulement si l'entrée est dans les limites. Il fait rester la valeur d'entrée au-dessus d'une valeur de limitation prédéfinie pendant plus d'un nombre prédéfini de cycles consécutifs.

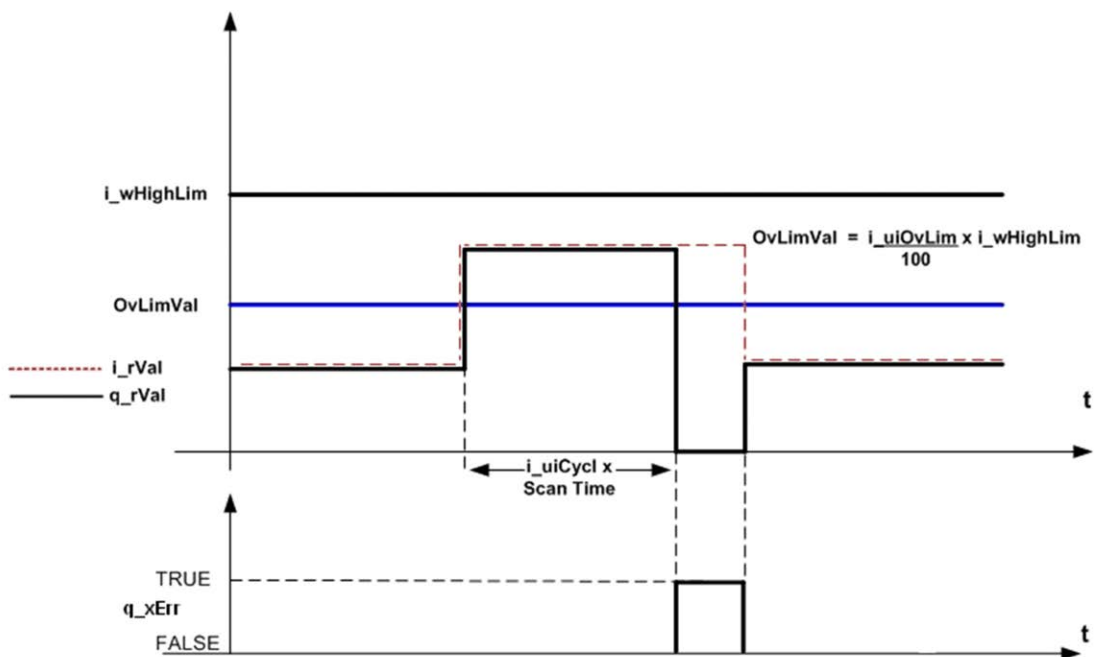


En fonctionnement normal, le signal d'entrée est transféré à la sortie en fonction des conditions suivantes

- Si le signal d'entrée est inférieur ou égal à la limite haute de limite de dépassement (%).
- Si le signal d'entrée dépasse la limite haute pendant un nombre "n" de cycles consécutifs inférieur à l'entrée de cycle.

Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction FB_Cyclic_Monitoring :



Etat d'erreur détectée

Si le signal d'entrée dépasse la limite de dépassement (%) de la limite haute pendant un nombre n de cycles supérieur ou égal à l'entrée de cycle, la sortie est mise à zéro et la sortie d'erreur détectée mise à TRUE. L'erreur détectée est réinitialisée automatiquement si le signal d'entrée revient dans les limites.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_Cyclic_Monitoring :

Entrée	Type de données	Description
i_wHighLim	WORD	Limite haute du signal Plage : 0...65535
i_uiOvlim	UINT	% de i_wHighLim définissant la plage de dépassement de limite. Plage : 0...100 La valeur d'entrée dépassant 100% est limitée à 100%.

Entrée	Type de données	Description
i_rVal	REAL	Signal d'entrée à surveiller Plage : $\pm 3,4e^{+38}$
i_uiCycl	UINT	Nombre de cycles d'automate que le bloc fonction autorise d'entrée en sortie quand l'entrée dépasse la plage de dépassement de limite. Plage : 0...65535

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_Cyclic_Monitoring :

Sortie	Type de données	Description
q_rVal	REAL	Valeur de sortie surveillée. Plage : $\pm 3,4e^{+38}$
q_xErr	BOOL	Bit d'erreur détectée
q_rPerc	REAL	Valeur d'entrée surveillée (i_rVal) en % de i_wHighLim. Plage : $\pm 3,4e^{+38}$

Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction FB_Cyclic_Monitoring :



Exemple

Ce tableau présente un exemple du fonctionnement du bloc fonction FB_Cyclic_Monitoring :

Exemple	Entrées	Sorties	Remarques
1	i_wHighLim = 200, i_uiOvLim = 50, i_rVal = 40, i_uiCycle = 10	q_rVal = 40, q_rPerc = 20, q_xErr = FALSE	Valeur d'entrée (i_rVal) inférieure ou égale à la valeur limite de dépassement calculé (c'est-à-dire $[(i_uiOvLim/100) \times i_wHighLim]$). Sortie : L'entrée i_rVal est affectée à la sortie q_rVal et la sortie q_xErr a la valeur FALSE.
2	i_wHighLim = 200, i_uiOvLim = 50, i_rVal = 110, i_uiCycle = 10	q_rVal = 110 q_rPerc = 55 q_xErr = FALSE	La valeur d'entrée (i_rVal) est supérieure à la valeur limite de dépassement calculé (c'est-à-dire $[(i_uiOvLim/100) \times i_wHighLim]$) et le nombre de cycles de scrutation terminés est inférieur au nombre de cycles définis. Sortie : L'entrée i_rVal est affectée à la sortie q_rVal et la sortie q_xErr a la valeur FALSE.
3	i_wHighLim = 200, i_uiOvLim = 50, i_rVal = 110, i_uiCycle = 10	q_rVal = 0 q_rPerc = 55 q_xErr = TRUE	La valeur d'entrée (i_rVal) est supérieure à la valeur limite de dépassement calculé (c'est-à-dire $[(i_uiOvLim/100) \times i_wHighLim]$) et le nombre de cycles de scrutation terminés est égal ou supérieur au nombre de cycles définis. Sortie : La sortie q_rVal est égale à zéro et la sortie q_xErr a la valeur TRUE.

Etat d'erreur détectée

Ce tableau présente quelques problème généraux avec leur solution :

Problème	Cause	Résolution
Etat d'erreur détectée	La valeur d'entrée (i_rVal) n'est pas dans la limite de dépassement (%) de i_wHighLim pendant le nombre de cycles n	Une valeur d'entrée (i_rVal) inférieure à la valeur limite de dépassement réinitialise automatiquement l'erreur détectée.

FB_DeadBand: Suppression d'oscillations d'amplitude

Contenu de ce chapitre

Bloc fonction FB_DeadBand	76
Description des broches d'entrée	78
Description des broches de sortie	78

Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_DeadBand.

Bloc fonction FB_DeadBand

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_DeadBand :



Description fonctionnelle

Le bloc fonction FB_DeadBand est un bloc fonction de plage morte permettant de faire passer l'entrée à la sortie seulement si elle est supérieure à la limite de la plage morte.

Ce bloc fonction supprime les oscillations de faible amplitude causées par du bruit, la quantification ou le calcul de paramètre. Il supprime le signal d'entrée s'il se trouve à l'intérieur du seuil comme indiqué sur la figure ci-dessous illustrant la fonction de transfert.

Sur le diagramme de temps :

- Si i_rInput est inférieur à la plage de seuil définie, q_rOput est mis à zéro et $q_xSigUndThsh$ a la valeur TRUE.
- Si la valeur d'entrée (i_rInput) est supérieure ou égale à la plage de seuil, q_rOput est égal à la valeur i_rInput .

q_xEn a la valeur TRUE tant que i_xEn a la valeur TRUE indépendamment d'une erreur détectée.

Cette figure présente la fonction de transfert pour le bloc fonction FB_DeadBand :

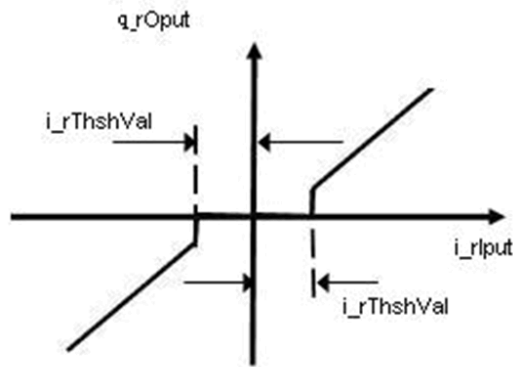
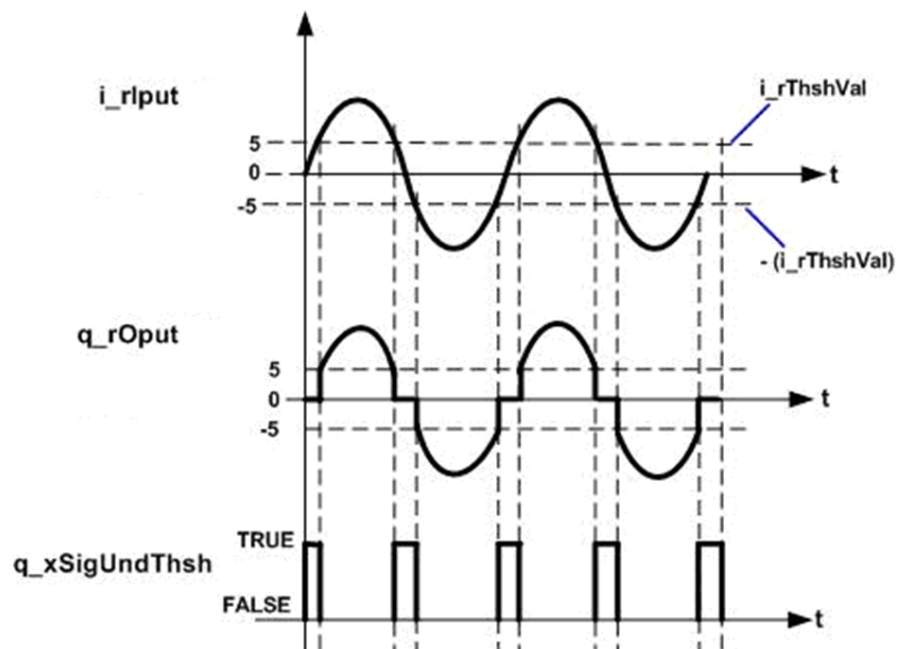


Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction FB_DeadBand :



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une d'erreur détectée et à la génération d'un ID d'erreur détectée correspondant. Pendant l'état d'erreur détectée, la valeur de sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée `i_xErrRst`.

`q_xBusy` a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_DeadBand :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction. FALSE : Désactive le bloc fonction.
i_rIput	REAL	Valeur d'entrée Plage : $\pm 3,4e^{+38}$
i_rThshVal	REAL	Valeur du seuil Plage : $0.0...3,4e^{+38}$
i_xErrRst	BOOL	Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction FB_DeadBand :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE si le bloc fonction est activé.
q_xBusy	BOOL	TRUE si le bloc fonction est activé et sans erreur détectée.
q_rOput	REAL	Sortie de l'entrée donnée. Plage : $\pm 3,4e^{+38}$
q_xSigUndThsh	BOOL	TRUE si l'entrée est en dessous de la limite du seuil.
q_xErr	BOOL	Erreur détectée
q_uiErrId	UINT	0 = Aucune erreur détectée 1 = Paramètre non valable $i_rThsh < 0$. Plage : 0...1

FB_Limiter : Limitation des signaux d'entrée

Contenu de ce chapitre

Bloc fonction FB_Limiter	79
Description des broches d'entrée	81
Description des broches de sortie	81

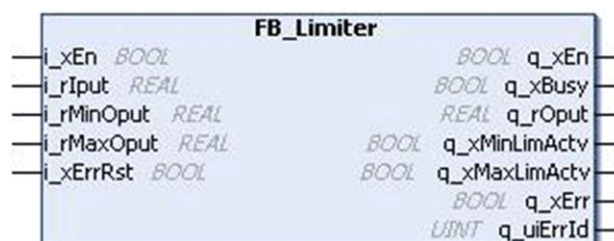
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_Limiter.

Bloc fonction FB_Limiter

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_Limiter :



Description fonctionnelle

Le bloc fonction FB_Limiter est un bloc fonction limiteur permettant de limiter un signal d'entrée dans une plage définie.

Le signal d'entrée est limité à une plage définie par les valeurs i_rMaxOput et i_rMinOput dont la fonction de transfert est présentée sur la figure ci-dessus.

Si l'entrée dépasse la limite supérieure ou inférieure, la sortie est limitée aux valeurs maximale ou minimale, respectivement.

Sur le diagramme de temps ci-dessous :

- Si l'entrée est dans la plage définie, la sortie est égale à la valeur d'entrée.
- Si la valeur d'entrée dépasse la limite maximale, la sortie est limitée à la valeur de sortie maximale.
- De même; si l'entrée descend en dessous de la valeur de sortie minimale, la sortie est limitée à la valeur de sortie minimale.
- Si le bloc fonction limite la sortie, q_xMinLimActv ou q_xMaxLimActv a la valeur TRUE, selon le type de limite franchie.

q_xEn a la valeur TRUE tant que i_xEn a la valeur TRUE, indépendamment d'une erreur détectée.

Cette figure présente la fonction de transfert du bloc fonction FB_Limiter :

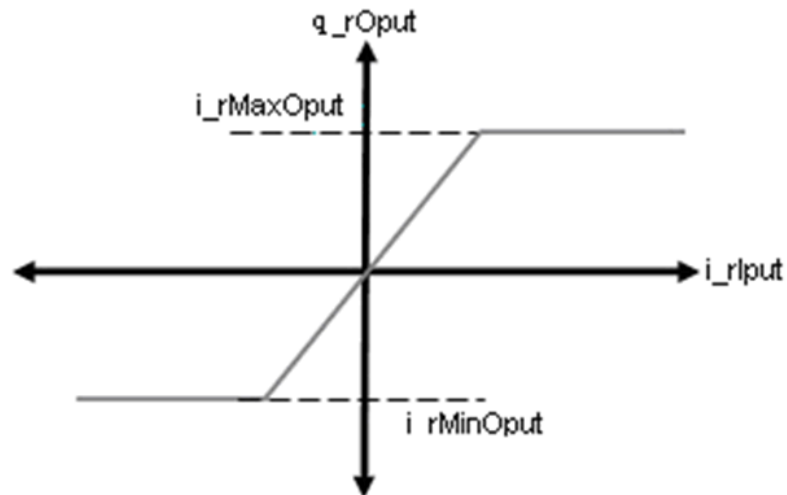
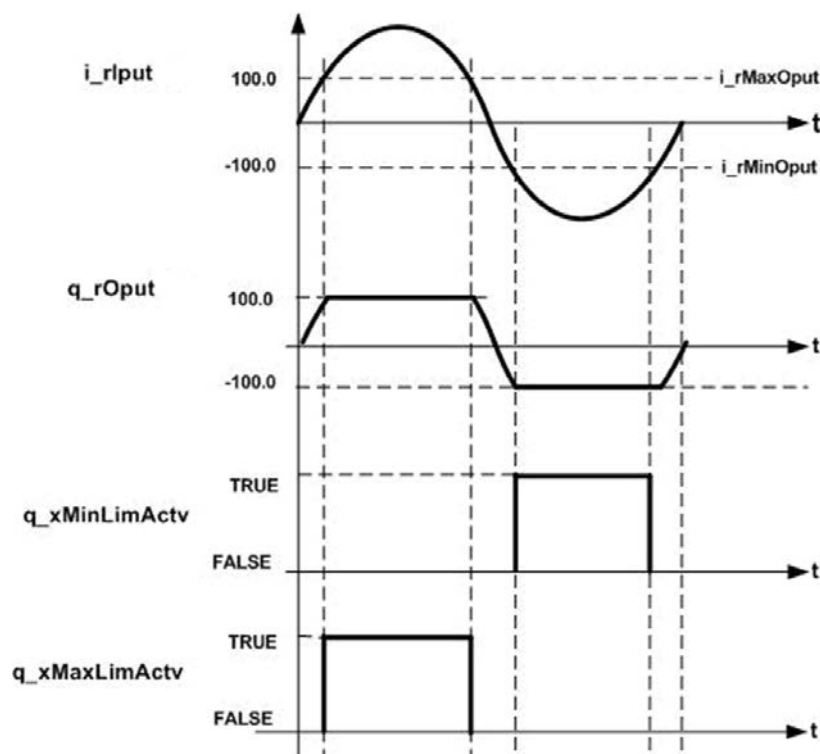


Diagramme de temps

Cette figure présente le diagramme de temps du bloc fonction FB_Limiter :



Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une d'erreur détecté et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$. La sortie q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_Limiter :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction. FALSE : Désactive le bloc fonction.
i_rIput	REAL	Valeur d'entrée devant être limitée. Plage : $\pm 3,4e^{+38}$
i_rMinOput	REAL	Valeur de sortie minimale. Plage : $\pm 3,4e^{+38}$
i_rMaxOput	REAL	Valeur de sortie maximale. Plage : $\pm 3,4e^{+38}$
i_xErrRst	BOOL	Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches d'entrée du bloc fonction FB_Limiter :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE si le bloc fonction est activé.
q_xBusy	BOOL	TRUE si le bloc fonction est activé et sans erreur détectée.
q_rOput	REAL	Sortie de l'entrée donnée. Plage : $\pm 3,4e^{+38}$
q_xMinLimActv	BOOL	TRUE si l'entrée est égale ou inférieure à la valeur de sortie minimale.
q_xMaxLimActv	BOOL	TRUE si l'entrée est égale ou supérieure à la valeur de sortie maximale.
q_xErr	BOOL	Erreur détectée
q_uiErrId	UINT	0 = Aucune erreur détectée 1 = Paramètre non valable $i_rMaxOput < i_rMinOput$ Plage : 0 ... 1

FB_PWM : Fourniture d'une sortie PWM

Contenu de ce chapitre

Bloc fonction FB_PWM	82
Description des broches d'entrée	86
Structure utilisée	87
Description des broches de sortie	87

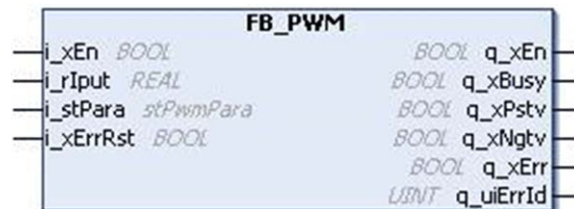
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_PWM.

Bloc fonction FB_PWM

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_PWM :



Description fonctionnelle

Le bloc fonction FB_PWM est conçu pour fournir une sortie PWM (modulation de largeur d'impulsion) en fonction du paramètre d'entrée.

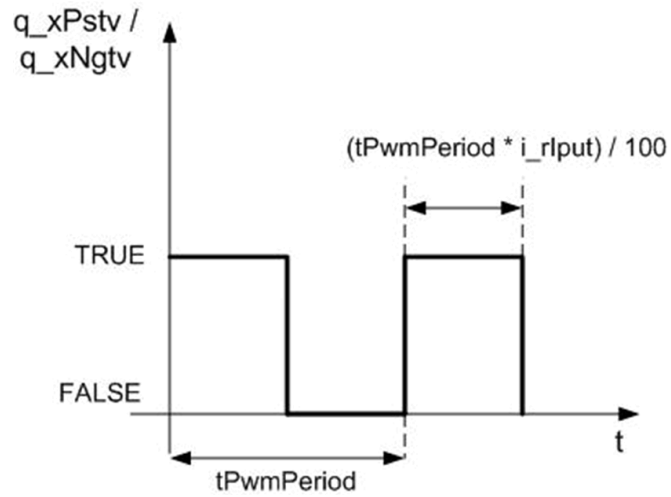
La sortie PWM est générée avec un temps d'activation ON et un temps de désactivation OFF conformes à l'entrée indiquée dans le premier chronogramme ci-dessous.

Par rapport au deuxième chronogramme :

- Si i_rIput est une valeur positive, la sortie PWM est disponible dans q_xPstv . L'entrée i_rIput doit être comprise entre -100 et 100. Le temps d'activation du bloc fonction PWM est calculé comme suit : Temps d'activation de PWM = $(i_rIput \times tPwmPeriod) / 100$.
- Si i_rIput a une valeur négative, la sortie PWM est disponible dans q_xNgtv .
- Si i_rIput est supérieur à 100, la valeur est limitée à 100. Si i_rIput est inférieur à -100, la valeur est limitée à -100.
- Si $i_xPwmInstUpdt$ a la valeur TRUE, la variation du paramètre d'entrée est mise à jour dans le cycle PWM en cours, comme indiqué sur le chronogramme.
- Si $i_xPwmInstUpdt$ a la valeur FALSE, la variation d'entrée n'est mise à jour qu'au démarrage d'un nouveau cycle PWM.

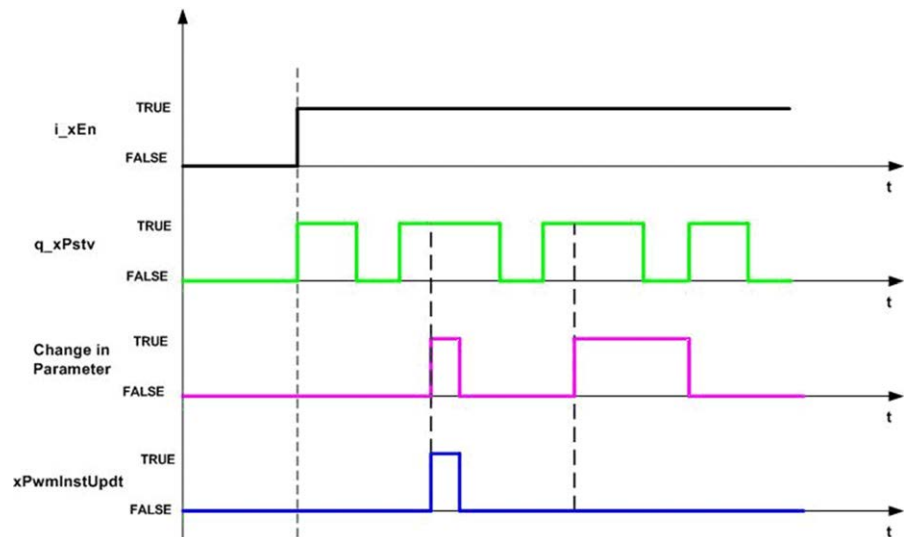
q_xEn a la valeur TRUE tant que l'entrée i_xEn a la valeur TRUE, quelle que soit l'erreur détectée.

Cette figure présente le chronogramme du calcul de FB_PWM :



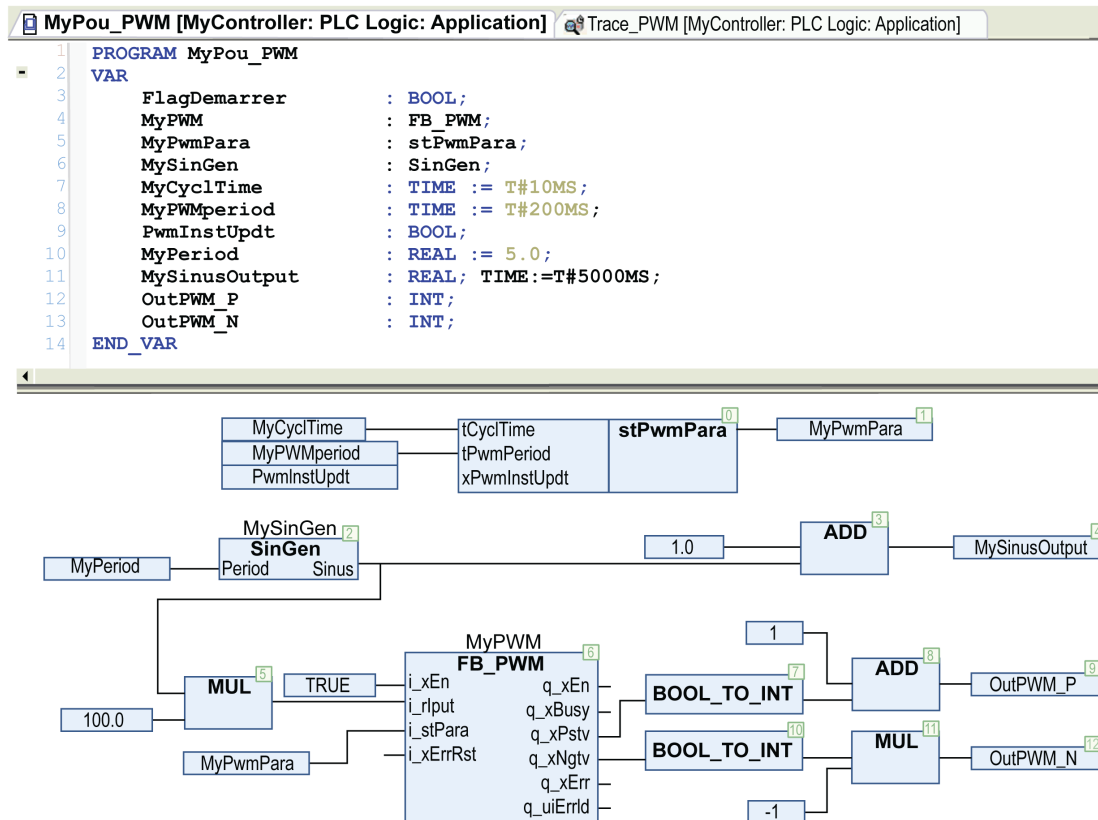
Chronogramme

Cette figure présente le chronogramme du bloc fonction FB_PWM :

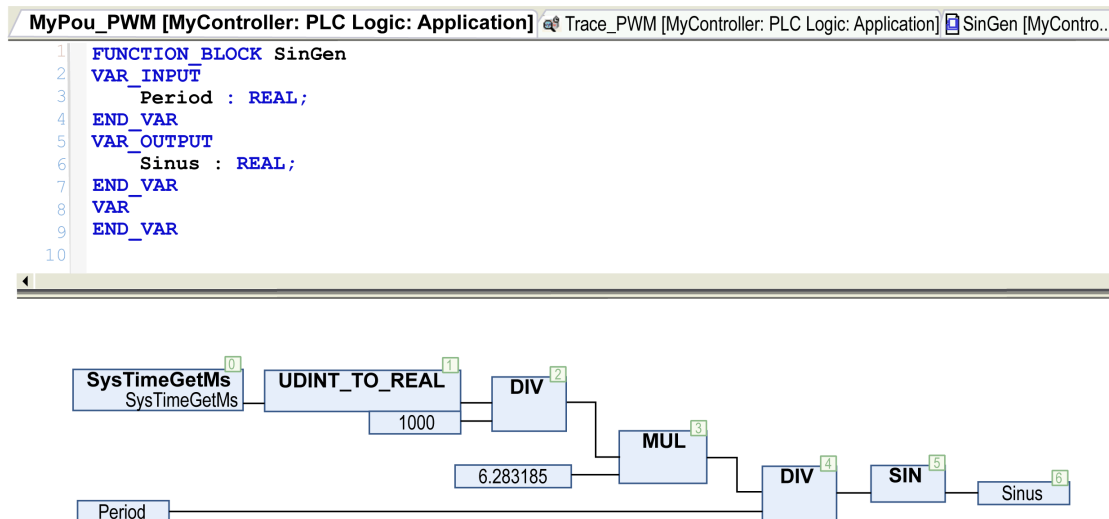


Exemple avec un signal de fréquence

Le programme crée un signal sinusoïdal avec une période définie (5 secondes/0,2 Hz). Ce signal sinusoïdal est l'entrée de FB_PWM.



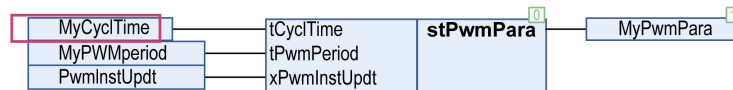
Définition du bloc fonction SinGen :



L'entrée `stPwmPara.tCyclTime` du bloc fonction `FB_PWM` doit avoir exactement la même valeur que la période du POU dans la tâche `MAST`, ici 10 millisecondes (voir la zone encadrée en rouge).

```

1 PROGRAM MyPou_PWM
2 VAR
3     FlagDemarrer      : BOOL;
4     MyPWM              : FB_PWM;
5     MyPwmPara         : stPwmPara;
6     MySinGen          : SinGen;
7     MyCyclTime        : TIME := T#10MS;
8     MyPWMperiod       : TIME := T#200MS;
9     PwmInstUpdt      : BOOL;
10    MyPeriod          : REAL := 5.0;
11    MySinusOutput     : REAL; TIME := T#5000MS;
12    OutPWM_P         : INT;
13    OutPWM_N         : INT;
14 END_VAR
    
```



Configuration

Priorité (0...31) : 15

Type
 Cyclique Intervalle (p.ex. t#200ms) : 10 ms

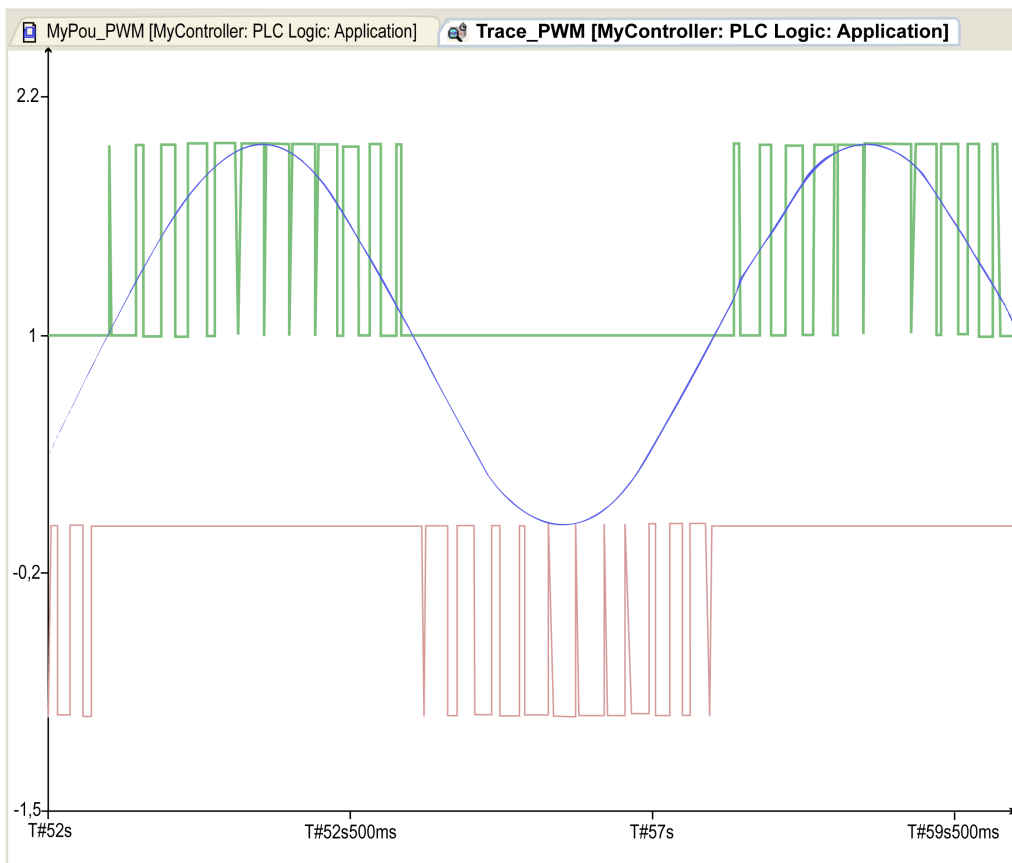
Chien de garde
 Activer
 temps (p.ex. t#200ms) : 100 ms
 Sensibilité : 1

POU

POU	Commentaire
MyPou_PWM	

[Ajouter POU](#)
[Supprimer POU](#)
[Ouvrir le POU](#)
[Aide à la saisie...](#)
 Monter
 Descendre

Résultat du POU précédent :



Bleu Signal sinusoïdal de `i_rInput` à 0,2 Hz (bloc fonction `My_Filter_PT1_1`).

Vert `q_xPstv` (un décalage est ajouté pour le tracé).

Rouge `q_xNgtv` (le signal est inversé pour le tracé).

Etat d'erreur détectée

Un paramètre non valable aux entrées du bloc fonction conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée `i_xErrRst`. La sortie `q_xBusy` a la valeur TRUE, chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau présente les broches d'entrée du bloc fonction `FB_PWM` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : Active le bloc fonction. FALSE : Désactive le bloc fonction.
<code>i_rInput</code>	REAL	Valeur d'entrée PWM Plage : -100...100

Entrée	Type de données	Description
<code>i_stPara</code>	STRUCT <code>stPwmPara</code>	Paramètre de structure
<code>i_xErrRst</code>	BOOL	Réinitialise l'erreur détectée (sur front montant) (optionnel)

Structure utilisée

stPwmPara

Élément de structure	Type	Description
<code>tCyclTime</code>	TIME	Durée de cycle de tâche Plage : 1...1e32 ms
<code>tPwmPeriod</code>	TIME	Période de temps PWM Plage : 1...1e32 ms
<code>xPwmInstUpdt</code>	BOOL	TRUE : une nouvelle valeur d'entrée est adoptée immédiatement, même à l'intérieur du cycle PWM en cours. (Facultatif)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches d'entrée du bloc fonction FB_PWM :

Sortie	Type de données	Description
<code>q_xEn</code>	BOOL	TRUE si le bloc fonction est activé.
<code>q_xBusy</code>	BOOL	TRUE si le bloc fonction est activé et sans erreur détectée.
<code>q_xPstv</code>	BOOL	La sortie PWM est positive si l'entrée PWM >0.
<code>q_xNgtv</code>	BOOL	La sortie PWM est négative si l'entrée PWM <0.
<code>q_xErr</code>	BOOL	Erreur détectée
<code>q_uiErrId</code>	UINT	0 = Aucune erreur détectée 1 = Paramètre non valable <code>i_tCyclTime = 0</code> . 2 = Paramètre non valable <code>i_tPwmPeriod <= i_tCyclTime</code> . Plage : 0...2

FB_Redundant_Sensor_Monitoring : Surveillance de capteur redondant

Contenu de ce chapitre

Bloc fonction FB_Redundant_Sensor_Monitoring	88
Description des broches d'entrée	89
Description des broches de sortie	92

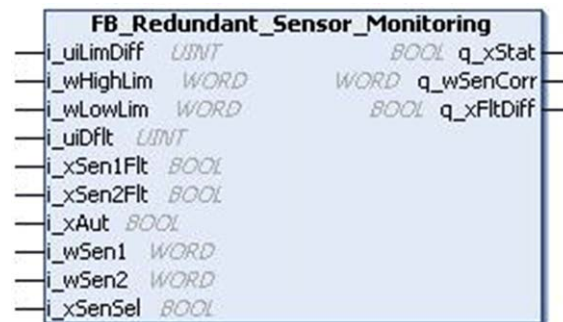
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_Redundant_Sensor_Monitoring.

Bloc fonction FB_Redundant_Sensor_Monitoring

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_Redundant_Sensor_Monitoring :



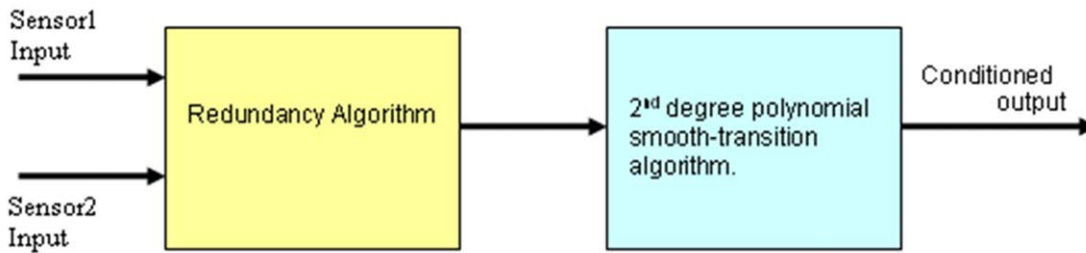
Description fonctionnelle

Le bloc fonction FB_Redundant_Sensor_Monitoring surveille les signaux provenant de deux sources redondantes de signaux analogiques ou capteurs de terrain de même plage et caractéristiques.

Le bloc fonction effectue les fonctions suivantes :

- Gère la sortie en fonction de la ou les valeurs lues sur les capteurs (valeur moyenne).
- Surveille si les 2 valeurs lues sur les capteurs sont dans la limite de différence spécifiée, sinon prend une action corrective.
- Effectue l'action prédéfinie au cas où un capteur est signalé comme non opérationnel.
- Sélection automatique du capteur opérationnel pour la sortie au cas où un autre capteur n'est pas opérationnel.
- Possibilité de sélection manuelle de capteur.
- Un algorithme de "transition douce" polynomial du second degré élimine toute variation brutale en créneaux dans la sortie du bloc.
- La sortie pour une différence détectée entre les deux capteurs est mise à TRUE quand la différence entre les deux valeurs de capteur n'est pas dans les limites. La différence entre les valeurs d'entrée de capteur est réinitialisée juste après que la différence entre les entrées de capteur est revenue dans les limites.

Cette figure présente le schéma fonctionnel du bloc fonction FB_Redundant_Sensor_Monitoring :



Ce tableau contient les conditions de l'algorithme polynomial du second degré de transition douce :

Exemple	Condition	Sortie du bloc fonction
1	La différence absolue entre la sortie calculée et la sortie précédente du bloc fonction est supérieure ou égale à 15, la sortie calculée est supérieure à la sortie précédente du bloc fonction.	Sortie précédente du bloc fonction - ((0.07 * (Sortie calculée - Sortie précédente du bloc fonction)) - 23)
2	La différence absolue entre la sortie calculée et la sortie précédente du bloc fonction est supérieure ou égale à 15, la sortie calculée est inférieure ou égale à la sortie précédente du bloc fonction.	Sortie précédente du bloc fonction - ((0.07 * (Sortie calculée - Sortie précédente du bloc fonction)) + 23)
3	La différence absolue entre la sortie calculée et la sortie précédente du bloc fonction est inférieure à 15.	Sortie calculée

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_Redundant_Sensor_Monitoring :

Sortie	Type de données	Description
i_uiLimDiff	UINT	La différence entre les valeurs sensor-1 et sensor-2 en % de la plage de capteur dans laquelle les capteurs redondants doivent travailler. Plage : 0...100 La valeur dépassant 100% est limitée à 100%. 0%: Il ne doit y avoir aucune différence entre les capteurs. 100%: Accepter toute valeur qui se présente ou toujours faire la moyenne entre Sen1_ip et Sen2_ip.
i_wHighLim	WORD	Limite haute des entrées de capteur Plage : 0...65535
i_wLowLim	WORD	Limite basse des entrées de capteur Plage : 0...65535
i_uiDflt	UINT	Valeur par défaut en % de la plage d'entrée du capteur Plage : 0...100
i_xSen1Flt	BOOL	TRUE : Le capteur a une erreur détectée FALSE : Le capteur est opérationnel.

Sortie	Type de données	Description
		(optionnel)
i_xSen2Flt	BOOL	TRUE : Le capteur a une erreur détectée. FALSE : Le capteur est opérationnel. (optionnel)
i_xAut	BOOL	TRUE : Mode automatique FALSE : Mode Manuel.
i_wSen1	WORD	Valeur brute Sensor-1 Plage : 0...65535 (On suppose que sensor-1 et 2 ont la même plage et les mêmes caractéristiques).
i_wSen2	WORD	Valeur brute Sensor-2 Plage : 0...65535 (On suppose que sensor-1 et 2 ont la même plage et les mêmes caractéristiques).
i_xSenSel	BOOL	TRUE : Sensor 2 sélectionné FALSE : Sensor 1 sélectionné Applicable seulement pour le mode manuel.

i_uiLimDiff

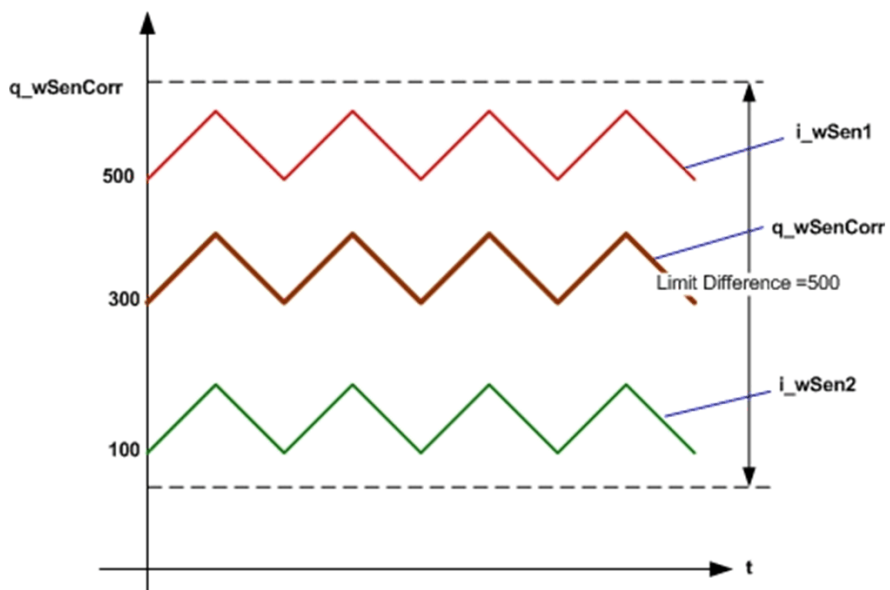
Différence limite en %. La sortie moyenne (q_wSenCorr) n'est générée que si la différence entre les deux entrées de capteur est inférieure à la différence limite.

- La différence limite se calcule par l'équation ci-dessous

$$\text{Différence limite} = (i_uiLimDiff \times (i_wHighLim - i_wLowLim)) / 100$$

- Si la différence entre l'entrée Sensor 1 et l'entrée Sensor 2 est inférieure à la différence limite, la sortie est la moyenne des deux capteurs comme indiqué sur la figure ci-dessous.

Cette figure présente la fonction de calcul de moyenne dans le bloc fonction FB_Redundant_Sensor_Monitoring :



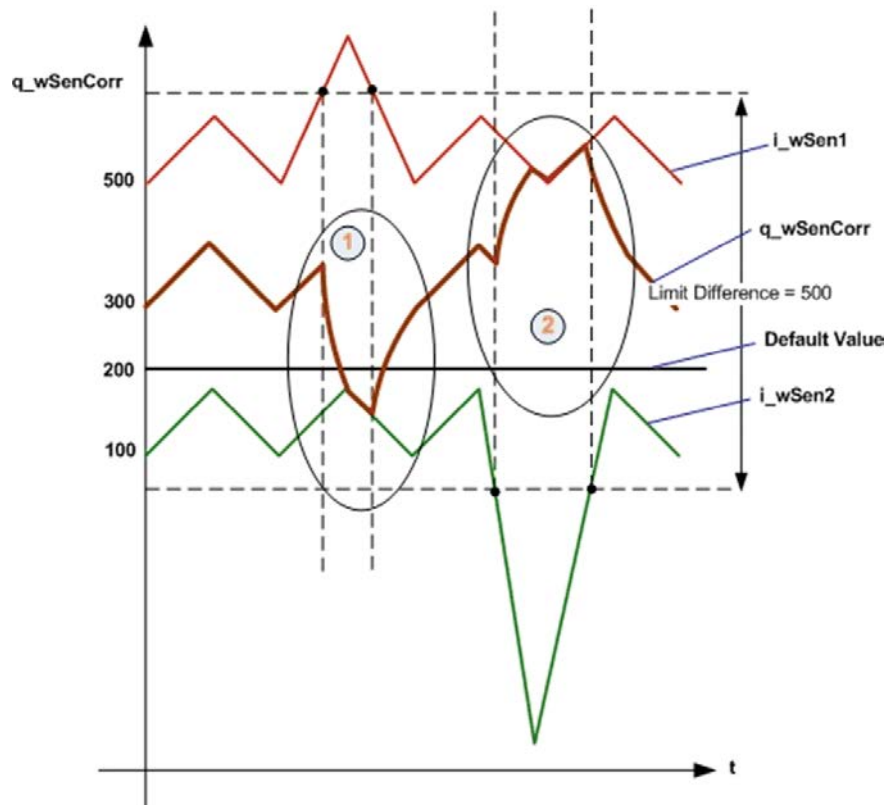
i_uiDflt

Cette entrée est la valeur par défaut en %, utilisée pour générer la sortie la plus appropriée si la différence n'est pas dans la limite et que les deux capteurs sont opérationnels.

- La valeur par défaut est calculée à partir de l'équation ci-dessous :

$$\text{Valeur par défaut} = (i_uiDflt \times (i_wHighLim - i_wLowLim)) / 100$$
- Si la différence entre l'entrée des 2 capteurs est hors limite, le bloc fonction donne en sortie la valeur du capteur le plus proche de la valeur par défaut comme indiqué sur la figure ci-dessous.

Cette figure présente la fonction de valeur par défaut dans le bloc fonction FB_Redundant_Sensor_Monitoring :



- 1 Difference between the sensors is not within the limit & Sensor 2 input is closer to default value, so Output follows the Sensor 2 regardless of sensor1 This transition is gradual due to Smooth transition algorithm.
- 2 Difference between the sensors is not within the limit & Sensor 1 input is closer to default value, so Output follows the Sensor 1 regardless of sensor 2. This transition is gradual due to Smooth transition algorithm.

i_xSen1Flt et i_xSen2Flt

Ces entrées permettent de détecter si les deux capteurs sont opérationnels ou non.

- Si les deux capteurs ne sont pas opérationnels, la sortie ($q_wSenCorr$) est mise à zéro.
- Si le capteur sensor 1 n'est pas opérationnel, la sortie est définie à l'entrée de sensor 2. De même si le capteur sensor 2 n'est pas opérationnel, la sortie est définie à l'entrée de sensor 1.

i_xAut

Cette entrée permet de sélectionner le mode auto ou manuel.

- Si l'entrée a la valeur TRUE, le bloc fonction fonctionne en mode auto. En mode auto, selon les entrées de capteur et la différences entre ces entrées, le bloc fonction génère une sortie appropriée.
- Si l'entrée a la valeur FALSE, le bloc fonction est en mode manuel. En mode manuel, selon l'entrée `i_xSenSel`, la sortie est forcée à la valeur de sensor 1 ou sensor 2.
- Si `i_xSenSel` a la valeur FALSE, c'est l'entrée sensor 1 qui est sélectionnée. De même si `i_xSenSel` a la valeur TRUE, c'est l'entrée sensor 2 qui est sélectionnée.

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `FB_Redundant_Sensor_Monitoring` :

Sortie	Type de données	Description
<code>q_xStat</code>	BOOL	TRUE : Mode automatique FALSE : Mode Manuel
<code>q_wSenCorr</code>	WORD	Valeur traitée de capteur redondant Plage : 0...65535
<code>q_xFltDiff</code>	BOOL	TRUE : Erreur détectée FALSE : Normal Cette sortie est à l'état TRUE quand la différence entre les capteurs n'est pas dans la limite pendant plus de 3 cycles consécutifs d'automate en mode auto.

NOTE: La sortie du bloc fonction `q_wSenCorr` ne change pas brutalement en fonction de l'entrée. Une fonction polynomiale du second degré est appliquée pour éviter les variations en créneaux brutaux dans la sortie du bloc.

FB_Scaling : Mise à l'échelle des signaux d'entrée

Contenu de ce chapitre

Bloc fonction <i>FB_Scaling</i>	93
Description des broches d'entrée	95
Description des broches de sortie	96

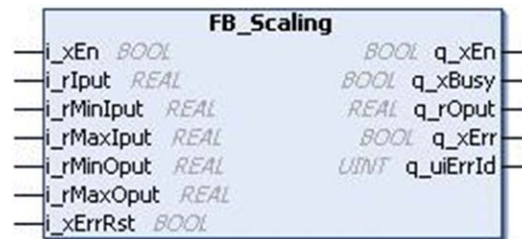
Vue d'ensemble

Ce chapitre décrit le bloc fonction *FB_Scaling*.

Bloc fonction *FB_Scaling*

Schéma d'affectation des broches

La figure suivante présente le schéma d'affectation des broches du bloc fonction *FB_Scaling* :



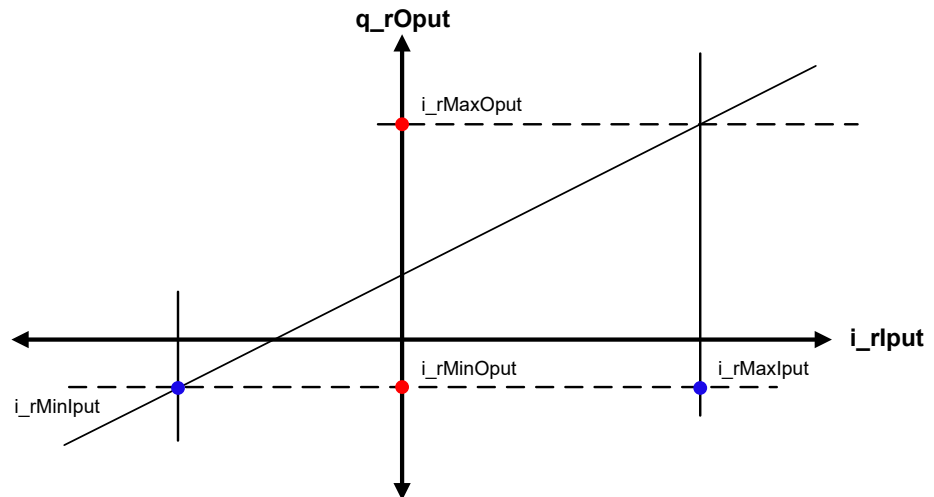
Description fonctionnelle

Le bloc fonction *FB_Scaling* est conçu pour convertir une valeur d'entrée linéairement dans une plage de sortie spécifiée et une erreur est détectée en cas de paramètre non valide.

Ce bloc fonction met à l'échelle un signal d'entrée en une sortie linéaire par rapport à une plage définie par un maximum et un minimum. Les valeurs minimales et maximales utilisées pour la mise à l'échelle ne limitent pas la valeur de sortie.

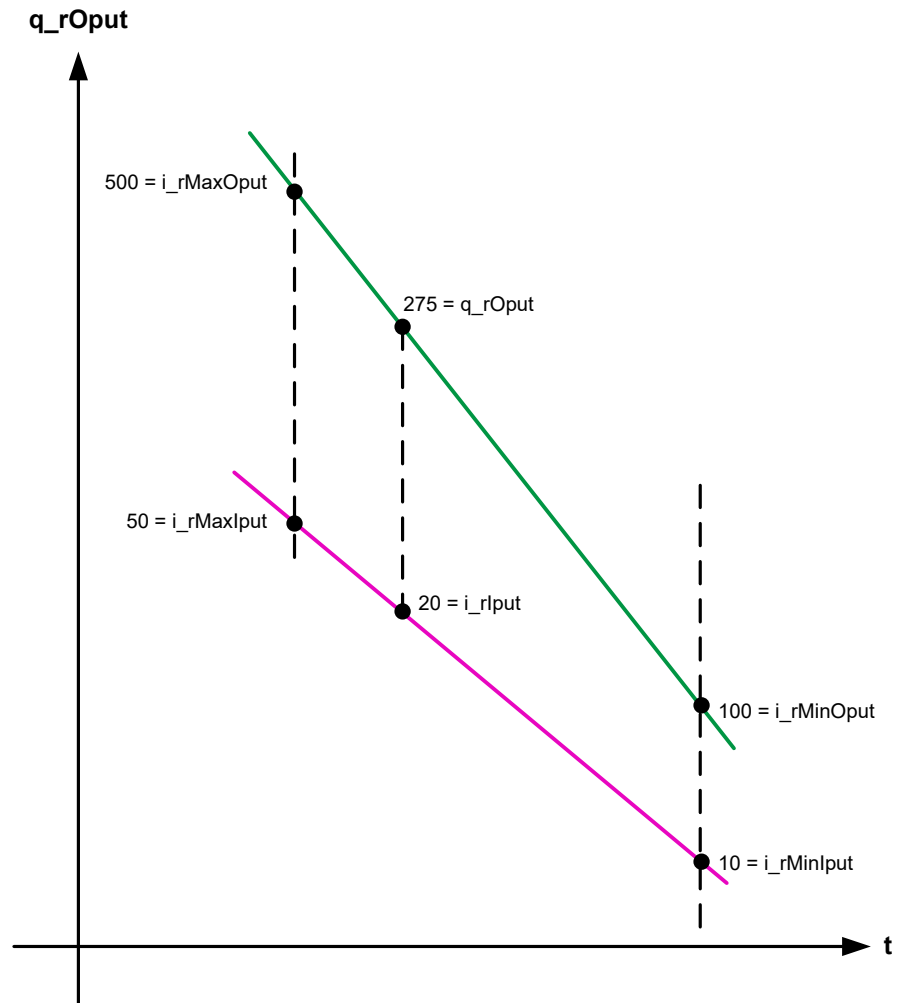
NOTE: Pour limiter la valeur mise à l'échelle fournie par le bloc fonction, utilisez *FB_Limiter*.

Le signal d'entrée est mis à l'échelle de manière linéaire par rapport à deux plages de valeurs comme indiqué sur la figure ci-dessous :



La sortie est modifiée dynamiquement en fonction de la variation de l'entrée :

- $Pente = (i_{rMaxOutput} - i_{rMinOutput}) / (i_{rMaxInput} - i_{rMinInput})$
- $Décalage = i_{rOutMax} - (Pente * i_{rMaxInput})$
- $q_{rOutput} = (Pente * i_{rInput}) + Décalage$



q_xEn a la valeur TRUE tant que l'entrée i_xEn a la valeur TRUE, quelle que soit l'erreur détectée.

Etat d'erreur détectée

Un paramètre non valide aux entrées du bloc fonction conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant. La sortie est mise à zéro pendant une erreur détectée. L'erreur détectée ne peut être réinitialisée que par un front montant de $i_xErrRst$. L'entrée q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `FB_Scaling` :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Active le bloc fonction. FALSE : Désactive le bloc fonction.
i_rIput	REAL	Valeur d'entrée à mettre à l'échelle. Plage : $\pm 3,4e^{+38}$

Entrée	Type de données	Description
i_rMinInput	REAL	Valeur d'entrée minimale Plage : $\pm 3,4e^{+38}$
i_rMaxInput	REAL	Valeur d'entrée maximale Plage : $\pm 3,4e^{+38}$
i_rMinOput	REAL	Valeur de sortie minimale Plage : $\pm 3,4e^{+38}$
i_rMaxOput	REAL	Valeur de sortie maximale Plage : $\pm 3,4e^{+38}$
i_xErrRst	BOOL	Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Le tableau ci-dessous inclut les différentes sorties du bloc fonction avec la description des identificateurs ou commandes.

Sortie	Type de données	Description
q_xEn	BOOL	TRUE si le bloc fonction est activé.
q_xBusy	BOOL	TRUE si le bloc fonction est activé et sans erreur détectée.
q_rOput	REAL	Sortie mise à l'échelle de l'entrée donnée. Plage : $\pm 3,4e^{+38}$
q_xErr	BOOL	Erreur détectée
q_uiErrId	UINT	0 = Aucune erreur détectée 1 = Paramètre non valable $i_rMinInput = i_rMaxInput$ Plage : 0...1

FB_Sensor_Monitoring : Surveillance de capteur

Contenu de ce chapitre

Bloc fonction FB_Sensor_Monitoring	97
Description des broches d'entrée	98
Description des broches de sortie	99
Instanciation et exemple d'utilisation	99

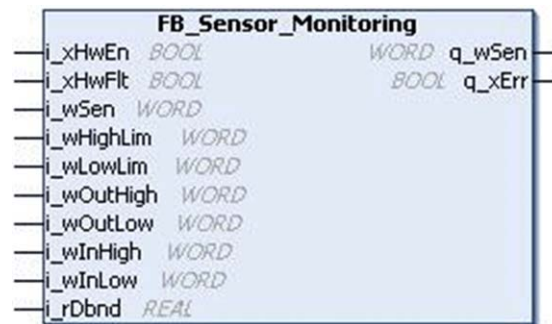
Vue d'ensemble

Ce chapitre décrit le bloc fonction FB_Sensor_Monitoring.

Bloc fonction FB_Sensor_Monitoring

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction FB_Sensor_Monitoring :



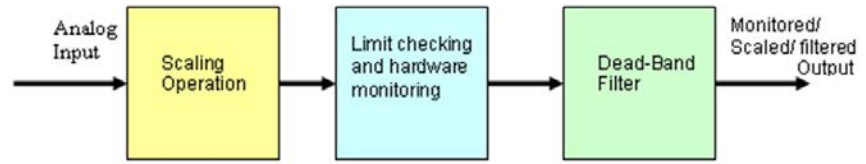
Description fonctionnelle

Le bloc fonction FB_Sensor_Monitoring surveille et/ou met à l'échelle et/ou filtre la plage morte d'un signal analogique d'entrée.

Ce bloc fonction effectue les opérations ci-dessous sur un signal d'entrée analogique :

- Vérifie si le relevé du capteur est dans la plage spécifiée par l'opérateur, sinon une sortie d'erreur est détectée.
- Surveille le matériel d'E/S et génère une alarme en cas de détection d'une erreur.
- Permet d'activer/désactiver la fonction de surveillance du matériel d'E/S dans le bloc.
- Met à l'échelle la valeur d'entrée dans la plage de sortie voulue.
- Passe la sortie finale par un filtre de plage morte. La plage morte supprime l'oscillation relative entre l'entrée précédente et l'entrée présente, puis génère une sortie.

Cette figure présente le schéma fonctionnel du bloc fonction FB_Sensor_Monitoring :



Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction FB_Sensor_Monitoring :

Entrée	Type de données	Description
i_xHwEn	BOOL	TRUE : Surveillance matérielle activée FALSE : Surveillance matérielle désactivée
i_xHwFlt	BOOL	TRUE : Erreur matérielle détectée FALSE : Aucune erreur matérielle détectée (optionnel)
i_wSen	WORD	Valeur de capteur Plage : 0...65535
i_wHighLim	WORD	Limite haute d'entrée de capteur. Plage : 0...65535 i_wHighLim devrait être supérieur à i_wLowLim
i_wLowLim	WORD	Limite basse d'entrée de capteur Plage : 0...65535
i_wOutHigh	WORD	Limite haute de plage de sortie pour la fonction de mise à l'échelle. Plage : 0..65535 Pour la fonction de mise à l'échelle, i_wOutHigh, i_wOutLow, i_wInHigh et i_wInLow sont utilisés
i_wOutLow	WORD	Limite basse de plage de sortie pour la fonction de mise à l'échelle. Plage : 0...65535
i_wInHigh	WORD	Limite haute de plage d'entrée pour la fonction de mise à l'échelle. Plage : 0...65535

Entrée	Type de données	Description
i_wInLow	WORD	Limite basse de plage d'entrée pour la fonction de mise à l'échelle. Plage : 0...65535
i_rDbnd	REAL	Définit la largeur de la "plage morte" dans le filtre de plage morte, en % de la plage. Plage : 0.0...100.0 0.0: Pas de filtrage de plage morte. 100.0: Bloque tous les signaux. Dans l'idéal la valeur est donc inférieure à 100.0 (optionnel)

Description des broches de sortie

Description des broches de sortie

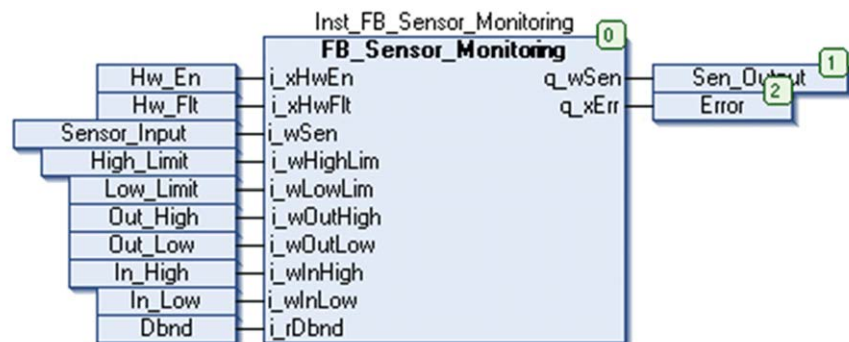
Ce tableau décrit les broches de sortie du bloc fonction FB_Sensor_Monitoring :

Sortie	Type de données	Description
q_wSen	WORD	Sortie de capteur valable Plage : 0...65535
q_xErr	BOOL	TRUE : En cas d'erreur matérielle détectée ou si la sortie mise à l'échelle n'est pas dans les limites de capteur pendant plus de 3 cycles consécutifs de scrutation de l'automate. FALSE : Aucune erreur détectée

Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Cette figure présente une instance du schéma d'affectation des broches du bloc fonction FB_Sensor_Monitoring :



Exemple

Cet exemple illustre diverses fonctionnalités du bloc fonction FB_Sensor_Monitoring :

Exemple	étapes,	Entrées	Sorties
1	<p>Mise à l'échelle :</p> <p>A partir des paramètres d'entrée de mise à l'échelle, l'entrée de capteur <code>i_wSen</code> est mise à l'échelle linéairement et la valeur de sortie mise à l'échelle passée pour vérification de limite.</p>	<code>i_wSen = 1000, i_wOutHigh = 2000, i_wOutLow = 100, i_wInHigh = 1000, i_wInLow = 10.</code>	<p>Sortie mise à l'échelle interne calculée = 2000</p>
2	<p>Vérification de limite :</p> <ul style="list-style-type: none"> • Si la sortie mise à l'échelle calculée est supérieure à l'entrée de limitation maximale, <code>i_wHighLim</code>, la sortie est limitée à <code>i_wHighLim</code> • Si la sortie mise à l'échelle calculée est inférieure à l'entrée de limitation minimale, <code>i_wLowLim</code>, la sortie est limitée à <code>i_wLowLim</code> • Si la sortie mise à l'échelle calculée est dans les limites, elle est passée pour traitement ultérieur aux fonctions de plage morte. <p><code>q_xErr</code> a la valeur TRUE si la sortie mise à l'échelle calculée est en dehors de la plage pendant plus de 3 cycles consécutifs de scrutation de l'automate.</p>	<code>i_wSen = 1000, i_wHighLim = 2000, i_wLowLim = 4000, i_wOutHigh = 2000, i_wOutLow = 100, i_wInHigh = 1000, i_wInLow = 10, i_rDbnd = 10.</code>	<p>Sortie interne après vérification de limite = 4000</p> <p><code>q_xErr = FALSE/TRUE.</code></p>

Exemple	étapes,	Entrées	Sorties
3	<p>Surveillance d'erreur matérielle détectée : La sortie <code>q_xErr</code> a la valeur TRUE et <code>q_wSen</code> maintient sa dernière valeur quand la surveillance d'erreur matérielle détectée est activée et que l'entrée d'erreur matérielle détectée (<code>i_xHwFlt</code>) a la valeur TRUE.</p>	<p><code>i_xHwEn=1, i_xHwFlt=1, i_wSen=1000, i_wOutHigh=2000, i_wOutLow=100, i_wInHigh=1000, i_wInLow=10, i_rDbnd=10.</code></p>	<p><code>q_wSen = 4000, q_xErr = TRUE.</code></p>
4	<p>Filtrage de plage morte :</p> <ul style="list-style-type: none"> • Si la différence entre la sortie mise à l'échelle calculée et la sortie précédente du bloc fonction est inférieure ou égale à la valeur de différence de plage morte calculée (c'est-à-dire $[(i_rDbnd/100) \times (i_wHighLim - i_wLowLim)]$), la sortie finale du bloc fonction est égale à la sortie précédente de ce bloc fonction. • Si la différence entre la sortie mise à l'échelle calculée et la sortie précédente du bloc fonction est supérieure à la valeur de différence de plage morte calculée (c'est-à-dire $[(i_rDbnd/100) \times (i_wHighLim - i_wLowLim)]$), la sortie finale du bloc fonction est égale à la sortie mise à l'échelle calculée. <p>La sortie d'état <code>q_xErr</code> peut être égale à 0 ou 1 selon les fonctionnalités de vérification de limite et de surveillance d'erreur matérielle détectée détaillées ci-dessus.</p> <p>Remarque : <code>i_rDbnd = 0</code>: Pas de filtrage de plage morte. <code>i_rDbnd >= 100</code>: Bloque tous les signaux.</p>	<p><code>i_wSen = 1000, i_wHighLim = 20000, i_wLowLim = 4000, i_wOutHigh = 2000, i_wOutLow = 100, i_wInHigh = 1000, i_wInLow = 10, i_rDbnd = 10.</code></p>	<p><code>q_wSen = 4000, q_xErr = False.</code></p>

Fonctions de filtrage

Contenu de cette partie

Liste des paramètres globaux.....	103
Filter_AnalogInput : Vérification de la variabilité d'entrée analogique	104
Filter_Arithmetic: Fourniture de moyenne arithmétique.....	107
Filter_MovingAverage : Fourniture d'une valeur de moyenne mobile	110
Filter_PT1 : Fourniture d'une fonction de transfert PT1	114

Vue d'ensemble

Cette partie décrit les fonctionnalités et la mise en œuvre des blocs fonction de filtrage.

Liste des paramètres globaux

Contenu de ce chapitre

Liste des paramètres globaux (GPL) 103

Liste des paramètres globaux (GPL)

Présentation

Type :	Paramètres globaux
Disponible à partir de :	3.0.1.0

Description

La liste des paramètres globaux (GPL) contient les constantes globales utilisées par les blocs de fonction de filtrage de la bibliothèque. Il est possible de modifier un à un les paramètres pour chaque application dans laquelle la bibliothèque est utilisée. Les modifications doivent être apportées dans le **gestionnaire de bibliothèques** du projet référençant la bibliothèque.

Paramètres globaux

Variable	Type de données	Valeur par défaut	Plage	Description
<i>Gc_uiMaxAvgeSmpl</i>	UINT	100	100 à 1000	Le nombre maximal d'échantillons comptés par l'entrée, page 112 <i>i_uiSmplCnt</i> .

Filter_AnalogInput : Vérification de la variabilité d'entrée analogique

Contenu de ce chapitre

Bloc fonction <code>Filter_AnalogInput</code>	104
Description des broches d'entrée	105
Description des broches de sortie	106

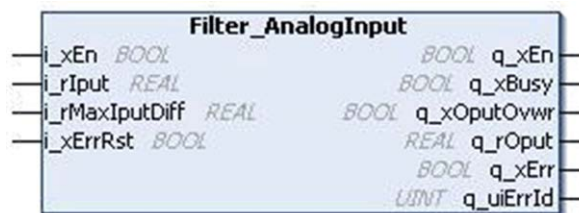
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Filter_AnalogInput`.

Bloc fonction `Filter_AnalogInput`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Filter_AnalogInput` :



Description fonctionnelle

Le bloc fonction `Filter_AnalogInput` vérifie la plausibilité d'une entrée analogique mesurée.

En état normal de fonctionnement, si la différence entre la valeur d'entrée présente et la valeur précédente :

- est inférieure ou égale à la valeur spécifiée `i_rMaxIputDiff`, la sortie suit la valeur d'entrée.
- est supérieure à la valeur spécifiée `i_rMaxIputDiff`, la sortie est remplacée par la valeur de sortie précédente pendant au maximum trois cycles de scrutation de l'automate. Le bit d'état de remplacement de sortie `q_xOputOvwr` est à TRUE dans ce cas.
- dépasse la valeur spécifiée `i_rMaxIputDiff` pendant plus de trois cycles consécutifs de scrutation de l'automate, la sortie suit à nouveau la valeur d'entrée.

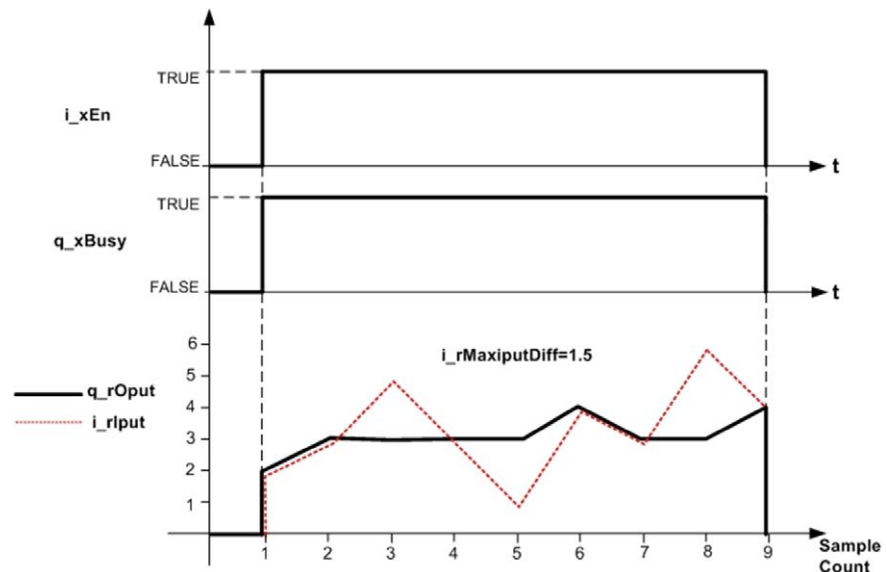
NOTE: A l'activation du bloc fonction, pendant le premier cycle de scrutation l'entrée est affectée à la sortie.

Exemple

Différence maximale entre les entrées présente et précédente (`i_rMaxIputDiff`) = 1.5 :

Cycle de scrutation	Valeur d'entrée (i_rIput)	Valeur de sortie (q_rOput)	Bit de remplacement de sortie (q_xOputOvvr)
Premier	2.0	2.0	FALSE
Deuxième	3.0	3.0	FALSE
Troisième	5.0	3.0	TRUE
Quatrième	3.0	3.0	TRUE
Cinquième	1.0	3.0	TRUE
Sixième	4.0	4.0	FALSE

La figure ci-dessous présente le comportement normal du bloc fonction `Filter_AnalogInput` :



Etat d'erreur détectée

Un paramètre non valable tel que `i_rMaxIputDiff < 0` conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant. Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée `i_xErrRst`.

Comme indiqué dans le comportement de la sortie figure ci-dessus, `q_xBusy` a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Filter_AnalogInput` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : Activé FALSE : Désactivé
<code>i_rIput</code>	REAL	Variable d'entrée analogique

Entrée	Type de données	Description
		Plage : $1,17e^{-38} \dots 3,4e^{+38}$
i_rMaxIputDiff	REAL	Différence maximale entre les entrées en cours et précédente pour vérification de plausibilité Plage : $0 \dots 3,4e^{+38}$ $i_rMaxIputDiff < 0$ génère une erreur détectée
i_xErrRst	BOOL	TRUE : Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Filter_AnalogInput :

Sortie	Type de données	Description
q_xEn	BOOL	TRUE : activé FALSE : désactivé
q_xBusy	BOOL	TRUE : actif et aucune erreur détectée. FALSE : désactivé ou erreur détectée.
q_xOputOvwr	BOOL	TRUE : si la sortie présente est remplacée par la sortie précédente. FALSE : si la sortie suit l'entrée présente.
q_rOput	REAL	Valeur de sortie analogique de Plage : $\pm 3,4e^{+38}$.
q_xErr	BOOL	TRUE : erreur détectée. FALSE : aucune erreur détectée.
q_uiErrId	UINT	Indique le numéro d'erreur détectée quand la sortie d'erreur détectée est active. 0 : aucune erreur détectée. 1 : paramètre non valable $i_rMaxIputDiff < 0$.

Filter_Arithmetic: Fourniture de moyenne arithmétique

Contenu de ce chapitre

Bloc fonction <code>Filter_Arithmetic</code>	107
Description des broches d'entrée	109
Description des broches de sortie	109

Vue d'ensemble

Ce chapitre décrit le bloc fonction `Filter_Arithmetic`.

Bloc fonction `Filter_Arithmetic`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Filter_Arithmetic`:



Description fonctionnelle

Le bloc fonction `Filter_Arithmetic` calcule la valeur moyenne arithmétique du nombre d'échantillons d'entrée défini par l'utilisateur.

Le calcul de la sortie commence à l'activation du bloc fonction.

Quand le nombre d'échantillons enregistrés est égal à la valeur spécifiée `i_uiSmplCnt`, le bloc fonction donne en sortie la moyenne calculée et le bit de validité de sortie `q_xOputVld` passe à TRUE.

La sortie du bloc fonction maintient cette valeur jusqu'à la désactivation du bloc fonction ou le passage dans l'état d'erreur détectée.

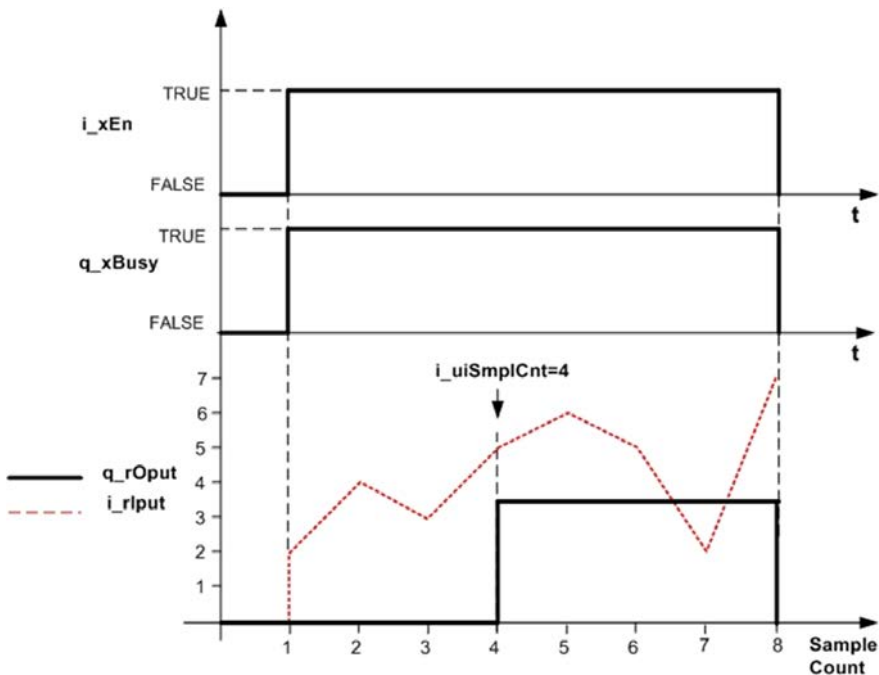
Exemple

Nombre d'échantillons dont la moyenne doit être calculée (`i_uiSmplCnt`) = 4:

Cycle de scrutation	Valeur d'entrée (<code>i_rInput</code>)	Valeur de sortie (<code>q_rOput</code>)	Bit de validité de sortie (<code>q_xOputVld</code>)
Premier	2.0	0	FALSE
Deuxième	4.0	0	FALSE
Troisième	3.0	0	FALSE
Quatrième	5.0	3.5	TRUE

Cycle de scrutation	Valeur d'entrée (i_rInput)	Valeur de sortie (q_rOutput)	Bit de validité de sortie (q_rOutput)
Cinquième	6.0	3.5	TRUE
Sixième	5.0	3.5	TRUE

Cette figure présente un comportement de sortie normal :



Complément mathématique

Cette équation présente la valeur moyenne arithmétique généralisée :

$$\bar{x} = \frac{1}{n} \left(\sum_n x_n \right)$$

Avec :

n = Nombre d'échantillons entré par l'utilisateur pour le calcul de valeur moyenne,

Xn = Echantillons d'entrée,

\bar{x} = Sortie calculée.

Etat d'erreur détectée

Un paramètre non valable tel que $i_uiSmplCnt = 0$ conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant. Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$.

Comme indiqué dans la figure de comportement de sortie ci-dessus, q_xBusy a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Filter_Arithmetic` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : Activé FALSE : Désactivé
<code>i_rInput</code>	REAL	Valeur d'entrée Plage : $\pm 3,4e^{+38}$
<code>i_uiSmplCnt</code>	UINT	Nombre d'échantillons (valeurs d'entrée) dont la moyenne doit être calculée Plage : 1...65535 <code>i_uiSmplCnt = 0</code> génère une erreur détectée.
<code>i_xErrRst</code>	BOOL	TRUE : Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Filter_Arithmetic` :

Sortie	Type de données	Description
<code>q_xEn</code>	BOOL	TRUE : Activé FALSE : Désactivé
<code>q_xBusy</code>	BOOL	TRUE : Actif et pas d'erreur détectée. FALSE : Désactivé ou erreur détectée.
<code>q_xOputVId</code>	BOOL	TRUE : Si le nombre d'échantillons enregistrés est supérieur ou égal à l'entrée <code>i_uiSmplCnt</code> .
<code>q_rOput</code>	REAL	Valeur de sortie de $\pm 3,4e^{+38}$
<code>q_xErr</code>	BOOL	TRUE : Erreur détectée. FALSE : Aucune erreur détectée.
<code>q_uiErrId</code>	UINT	Numéro d'erreur détectée quand la sortie d'erreur détectée est active. 0: Aucune erreur détectée. 1: Paramètre non valable <code>i_uiSmplCnt = 0</code> .

Filter_MovingAverage : Fourniture d'une valeur de moyenne mobile

Contenu de ce chapitre

Bloc fonction <code>Filter_MovingAverage</code>	110
Description des broches d'entrée	112
Description des broches de sortie	112

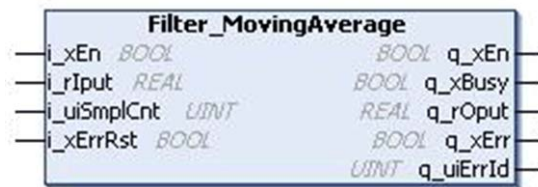
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Filter_MovingAverage`.

Bloc fonction `Filter_MovingAverage`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Filter_MovingAverage` :



Description fonctionnelle

Le bloc fonction `Filter_MovingAverage` calcule la valeur de la moyenne mobile sur le nombre d'échantillons d'entrée défini par l'utilisateur.

Quand le nombre d'échantillons enregistrés est :

- Inférieur à la valeur spécifiée `i_uiSmplCnt`, le bloc fonction calcule la valeur moyenne avec le nombre d'entrées disponibles et donne la sortie correspondante.
- Egal ou supérieur à la valeur spécifiée `i_uiSmplCnt`, le bloc fonction calcule la valeur moyenne avec le nombre d'entrées `i_uiSmplCnt` et fournit la sortie correspondante. Il fonctionne comme filtre de moyenne mobile.
- Pour `i_uiSmplCnt = 0`, la valeur d'entrée est affectée à la sortie.

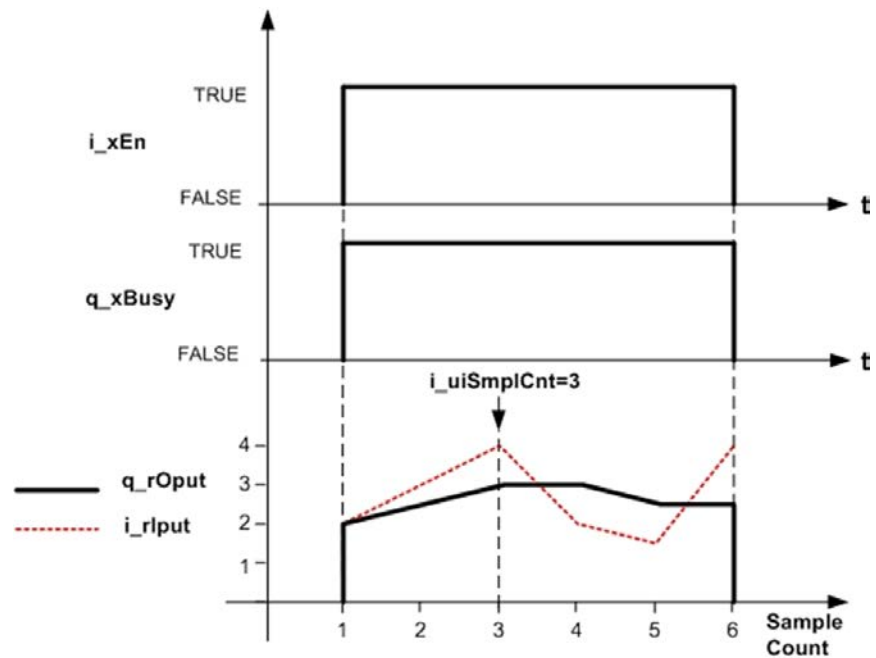
Exemple

Nombre d'échantillons dont la moyenne doit être calculée (`i_uiSmplCnt`) = 3 :

Cycle de scrutation	Valeur d'entrée (<code>i_rInput</code>)	Valeur de sortie (<code>q_rOput</code>)
1	2,0	2,0
2	3,0	2,5
3	4,0	3,0
4	2,0	3,0

Cycle de scrutation	Valeur d'entrée (i_rInput)	Valeur de sortie (q_rOutput)
5	1,5	2,5
6	4.0	2,5

Cette figure présente un comportement normal :



Complément mathématique

Cette équation présente l'équation généralisée de la fonction `Filter_MovingAverage` :

$$\bar{x}_k^n = \frac{1}{n} \left(\sum_{i=k-n}^k x_i \right)$$

n = Nombre d'échantillons,

x_i = Echantillons d'entrée,

k = `GPL.Gc_uiMaxAvgeSmpl`, constante interne,

\bar{x}_k^n = Sortie calculée.

Remarque

En cas de diminution du nombre d'échantillons (`i_uiSmplCnt`), la sortie (`q_rOutput`) dans les scrutations ultérieures est calculée par réduction du nombre d'échantillons d'une unité à chaque scrutation consécutive.

Etat d'erreur détectée

Un paramètre non valable tel que `i_uiSmplCnt > GPL.Gc_uiMaxAvgeSmpl` entraîne une erreur détectée et la génération d'un ID d'erreur détectée correspondant.

Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée `i_xErrRst`.

Comme indiqué dans le comportement de la sortie figure ci-dessus, `q_xBusy` a la valeur TRUE chaque fois que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Filter_MovingAverage` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE: activé FALSE : désactivé
<code>i_rInput</code>	REAL	Valeur d'entrée Plage : $\pm 3,4e^{+38}$
<code>i_uiSmplCnt</code>	UINT	Nombre d'échantillons (valeurs d'entrée) dont la moyenne doit être calculée Plage : 0 à <code>GPL.Gc_uiMaxAvgeSmpl</code> <code>i_uiSmplCnt > GPL.Gc_uiMaxAvgeSmpl</code> génère une erreur détectée.
<code>i_xErrRst</code>	BOOL	Réinitialise l'erreur détectée (sur front montant) (Facultatif)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Filter_MovingAverage` :

Sortie	Type de données	Description
<code>q_xEn</code>	BOOL	TRUE: activé FALSE : désactivé
<code>q_xBusy</code>	BOOL	TRUE : Actif et aucune erreur détectée. FALSE : désactivé ou erreur détectée.
<code>q_rOput</code>	REAL	Valeur de la sortie Plage : $\pm 3,4e^{+38}$

Sortie	Type de données	Description
q_xErr	BOOL	TRUE : erreur détectée. FALSE : aucune erreur détectée.
q_uiErrId	UINT	Numéro d'erreur détectée quand la sortie d'erreur détectée est active. 0 : aucune erreur détectée. 1 : Paramètre non valide $i_uiSmplCnt > GPL.Gc_uiMaxAvgeSmpl$.

Filter_PT1 : Fourniture d'une fonction de transfert PT1

Contenu de ce chapitre

Bloc fonction <code>Filter_PT1</code>	114
Description des broches d'entrée	116
Description des broches de sortie	116
Instanciation et exemple d'utilisation	117

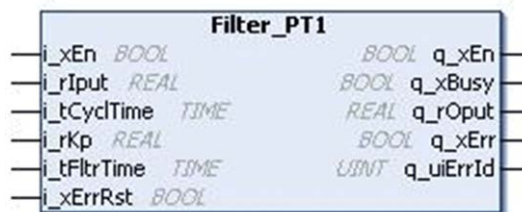
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Filter_PT1`.

Bloc fonction `Filter_PT1`

Schéma d'affectation des broches

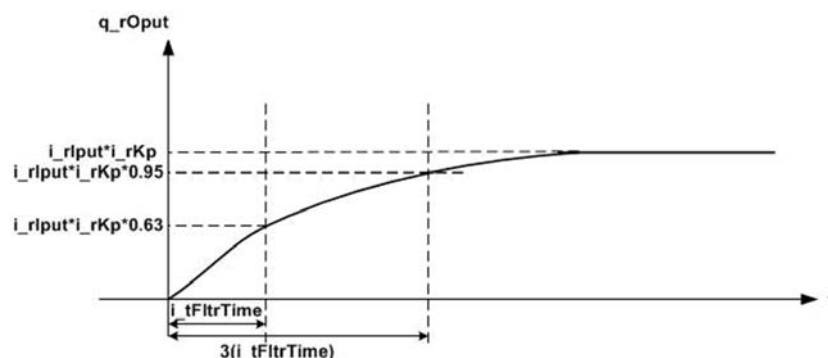
La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Filter_PT1` :



Description fonctionnelle

Le bloc fonction `Filter_PT1` fournit une fonction de transfert PT1. La valeur de sortie augmente à 63% de la valeur d'entrée dans un temps égal à la constante de temps du filtre. La valeur de sortie atteint 95% de la valeur d'entrée après une durée égale à 3 * Constante de temps du filtre puis monte progressivement à 100% de la valeur d'entrée.

Cette figure présente la fonctionnalité de profil de sortie du bloc fonction `Filter_PT1` :



Quand la période est égale à :

- La constante de temps du filtre, la valeur de sortie augmente à 63% de la valeur d'entrée.

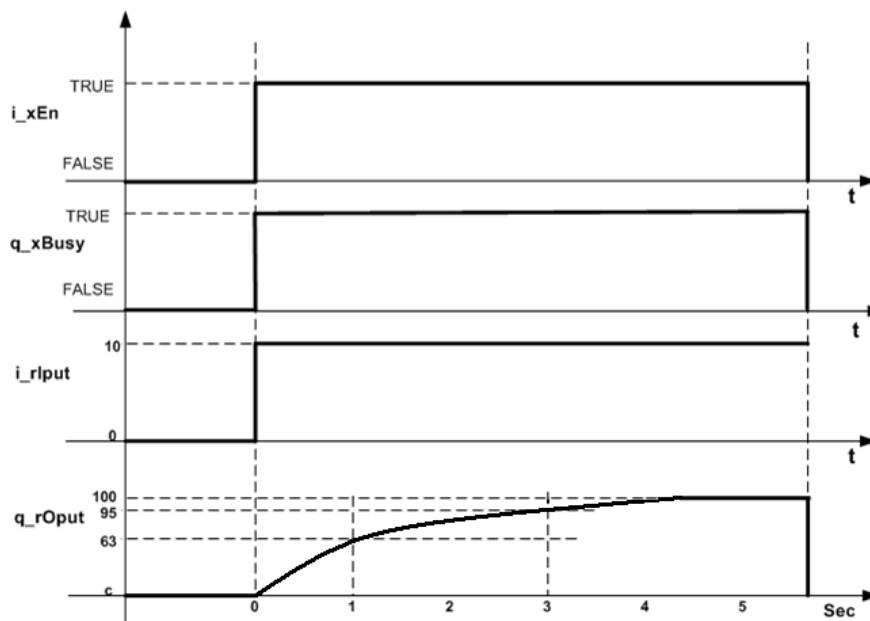
- Trois fois la constante de temps du filtre, la valeur de sortie augmente à 95% de la valeur d'entrée puis monte progressivement à 100% de l'entrée.

Exemple

Si la valeur d'entrée (i_rInput) est égale à 10 et la constante de temps du filtre ($i_tFltrTime$) est une seconde pour un gain de filtrage de 10, la valeur de sortie (q_rOput) est égale à 63 après une durée de une seconde.

La valeur de sortie est égale à 95 après une durée de trois secondes (trois fois la constante de temps du filtre), puis la sortie monte progressivement à 100.

Cette figure présente un comportement normal :



Complément mathématique

Cette équation présente la fonction de transfert :

$$G(s) = K_p \frac{1}{1 + T_s s}$$

Avec :

K_p	= Gain ou amplification de la fonction PT1
T_s	= Constante de temps du filtre de la fonction PT1
$G(s)$	= Fonction de transfert

L'équation présentée ci-dessus est en notation de Laplace pour le filtre passe-bas de premier ordre.

Dans les systèmes temporels numériques, cette fonction est souvent appelée pulse-transfer (fonction PT1).

Etat d'erreur détectée

Un paramètre non valable tel que $i_tCyclTime = 0$ ou $i_tFltrTime < i_tCyclTime$ conduit à une erreur détectée et à la génération d'un ID d'erreur détectée correspondant. Pendant l'état d'erreur détectée, la sortie est mise à zéro.

L'erreur détectée ne peut être réinitialisée que par un front montant de l'entrée $i_xErrRst$.

Comme indiqué dans la figure de sortie du bloc fonction, q_xBusy a la valeur TRUE tant que le bloc fonction est activé et qu'il n'y a pas d'erreur détectée.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches de sortie du bloc fonction Filter_PT1 :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Activé FALSE : Désactivé
i_rIput	REAL	Valeur d'entrée pour traitement Plage : $\pm 3,4e^{+38}$
$i_tCyclTime$	TIME	Durée de cycle de tâche Plage : 0...4294967295 ms $i_tCyclTime < 0$ génère une erreur détectée.
i_rKp	REAL	Gain/amplification de la fonction PT1 Plage : $\pm 3,4e^{+38}$
$i_tFltrTime$	TIME	Constante de temps du filtre Plage : 0...4294967295 ms Réglage d'usine = t#0ms, $i_tFltrTime < i_tCyclTime$ génère une erreur détectée.
$i_xErrRst$	BOOL	TRUE : Réinitialise l'erreur détectée (sur front montant) (optionnel)

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Filter_PT1 :

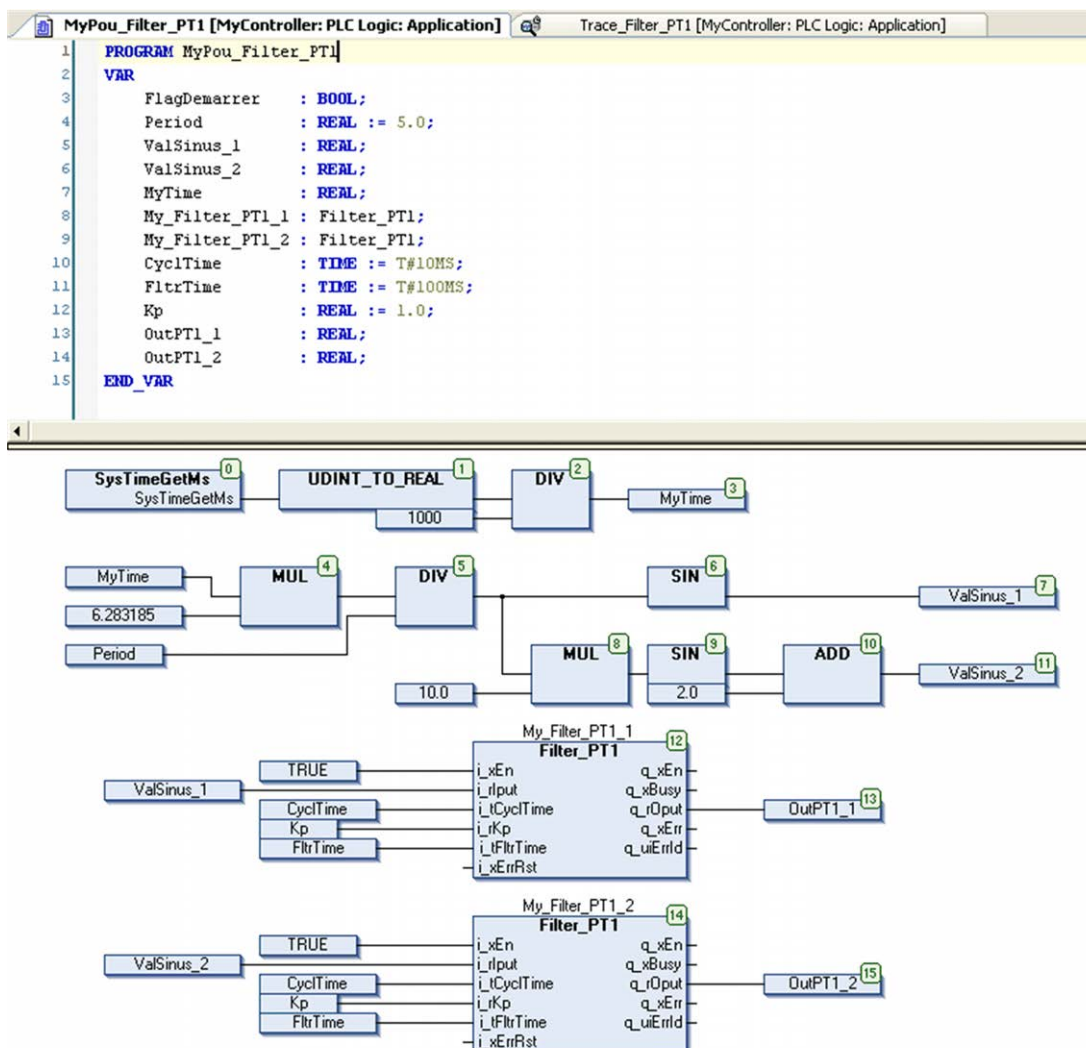
Sortie	Type de données	Description
q_xEn	BOOL	TRUE : Activé FALSE : Désactivé
q_xBusy	BOOL	TRUE : Actif et pas d'erreur détectée. FALSE : Désactivé ou erreur détectée.
q_rOput	REAL	Valeur de sortie

Sortie	Type de données	Description
		Plage : $\pm 3,4e^{+38}$
q_xErr	BOOL	TRUE : Erreur détectée. FALSE : Aucune erreur détectée.
q_uiErrId	UINT	Numéro d'erreur détectée quand la sortie d'erreur détectée est active. 0: Aucune erreur détectée. 1: Paramètre non valable $i_tCyclTime < 0$. 2: Paramètre non valable $i_tFltrTime < i_tCyclTime$.

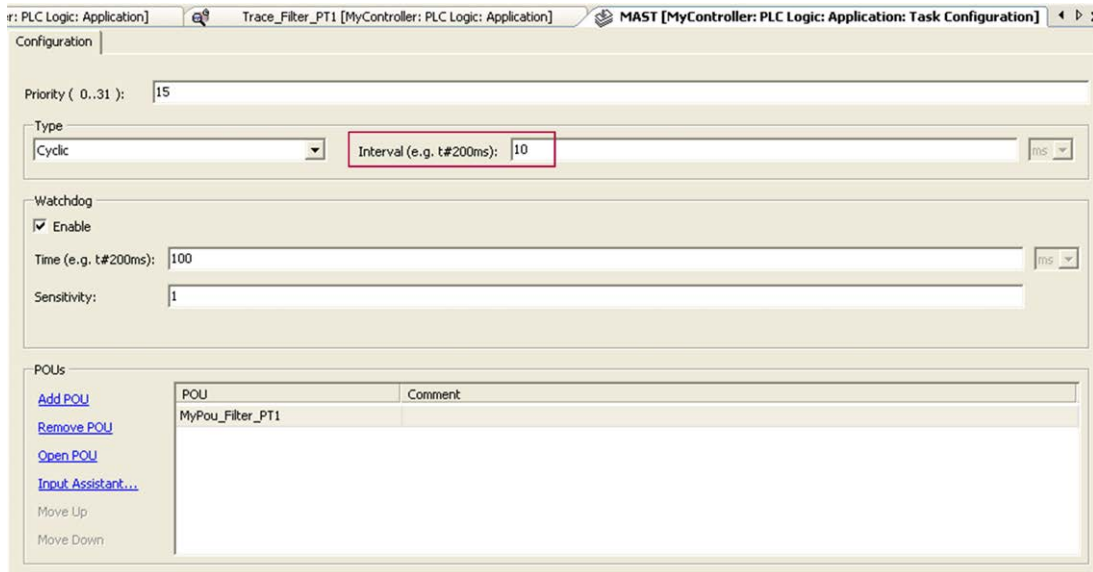
Instanciation et exemple d'utilisation

Exemple avec un signal de fréquence

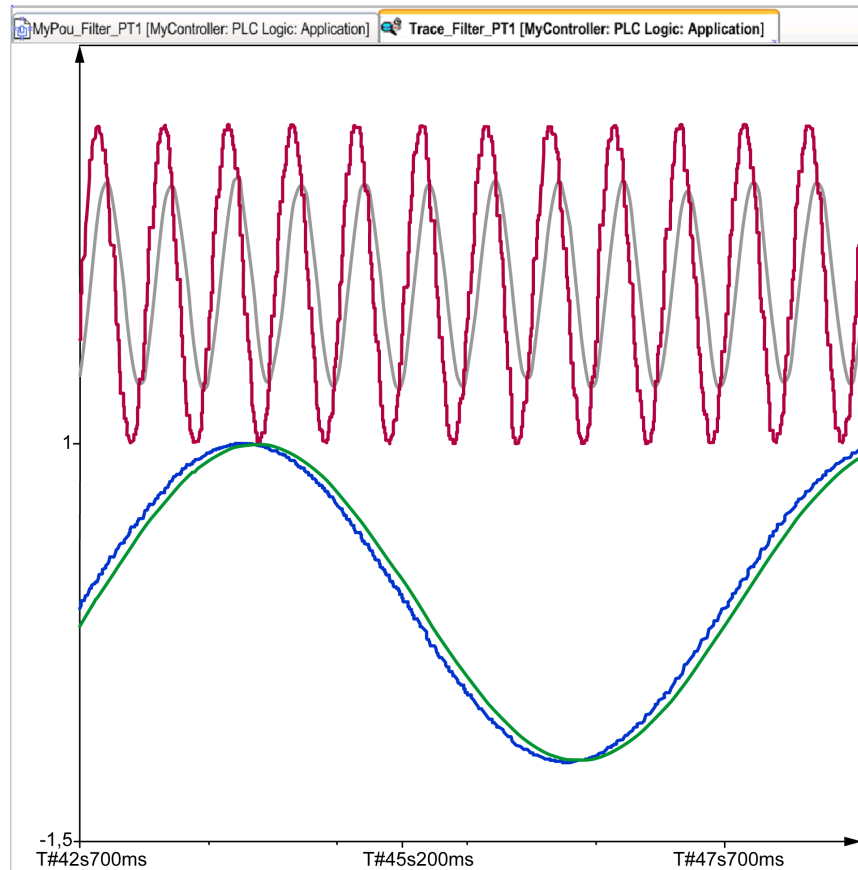
Le programme crée un signal sinusoïdal avec une certaine période (5 secondes/0,2 Hz) et un signal sinusoïdal plus élevé d'une décade (0,5 secondes/2 Hz).



L'entrée `i_tCyclTime` du bloc fonction `Filter_PT1` doit avoir exactement la même valeur que la période du POU dans le MAST, ici 10 millisecondes (voir zone encadrée en rouge).

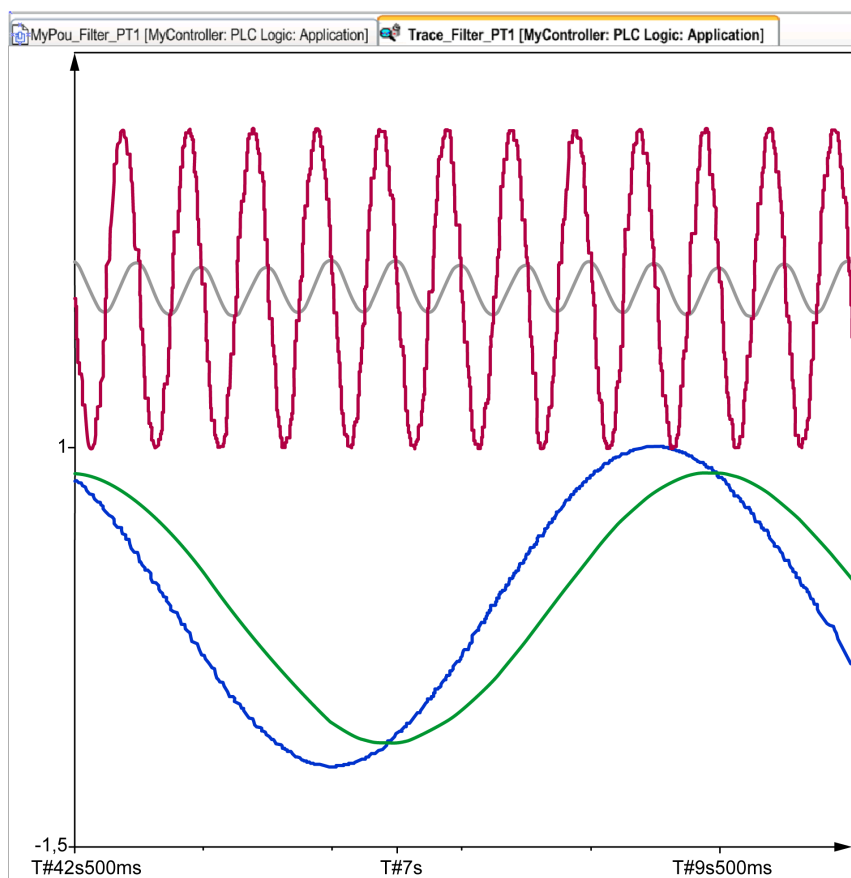


Résultat du POU précédent quand l'entrée `i_tFltrTime` est égale à 100 ms :



- Bleu** `i_rInput` signal sinusoïdal à 0,5 Hz (bloc fonction `My_Filter_PT1_1`).
- Vert** `q_rOutput` signal filtré (bloc fonction `My_Filter_PT1_1`)
- Rouge** `i_rInput` signal sinusoïdal à 5 Hz (bloc fonction `My_Filter_PT1_2`).
- Gris** `q_rOutput` signal filtré (bloc fonction `My_Filter_PT1_2`)

Résultat du POU précédent quand l'entrée $i_tFiltrTime$ est égale à 500 ms :



Bleu i_rInput signal sinusoïdal à 0,5 Hz (bloc fonction `My_Filter_PT1_1`).

Vert q_rOput signal filtré (bloc fonction `My_Filter_PT1_1`)

Rouge i_rInput signal sinusoïdal à 5 Hz (bloc fonction `My_Filter_PT1_2`).

Gris q_rOput signal filtré (bloc fonction `My_Filter_PT1_2`)

Retournements

Contenu de cette partie

JK_FlipFlop : Initialisation/réinitialisation de l'entrée vers sortie de retournement	121
JK_FlipFlop_MasterSlave: Initialisation/réinitialisation de l'entrée de la sortie de retournement.....	123
RS_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement	126
SR_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement	128
Toggle_FlipFlop: Basculement de l'entrée/sortie de retournement.....	130

Vue d'ensemble

Cette partie décrit la famille de Retournements.

JK_FlipFlop : Initialisation/réinitialisation de l'entrée vers sortie de retournement

Contenu de ce chapitre

Bloc fonction JK_FlipFlop 121

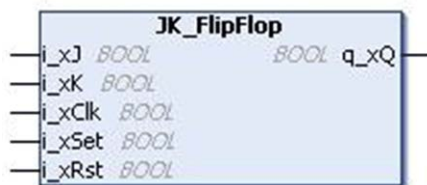
Vue d'ensemble

Ce chapitre décrit le bloc fonction JK_FlipFlop.

Bloc fonction JK_FlipFlop

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction JK_FlipFlop :



Description fonctionnelle

Le bloc fonction JK_FlipFlop assure la mise en œuvre de la table de vérité du retournement JK.

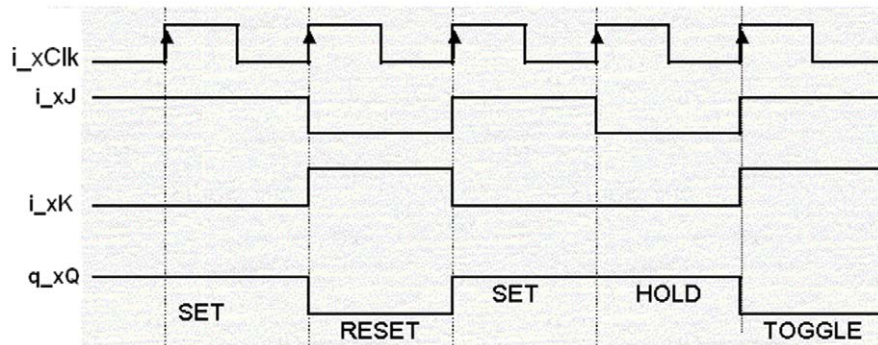
Ce bloc fonction fait référence à un retournement obéissant à la table de vérité ci-dessous :

i_xClk	i_xJ	i_xK	q_xQ (n)	q_xQ (n+1)	Fonctionnement
0	X	X	X	Q(n)	Maintien
RE	0	0	0	0	Maintien
RE	0	0	1	1	Maintien
RE	0	1	0	0	Reset
RE	0	1	1	0	Reset
RE	1	0	0	1	Activer
RE	1	0	1	1	Activer
RE	1	1	0	1	Toggle
RE	1	1	1	0	Toggle

r "n" est l'état présent et (n+1) l'état suivant.
RE Front montant

L'entrée Reset (i_xRst) réinitialise la sortie du retournement q_xQ, alors que l'entrée Set (i_xSet) active la sortie du retournement q_xQ.

Table de vérité sous forme de diagramme temporel :



Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction JK_FlipFlop :

Entrée	Type de données	Description
i_xJ	BOOL	TRUE : Entrée i_xJ active. FALSE : Désactivé (réglage usine)
i_xK	BOOL	TRUE : Entrée i_xK active. FALSE : Désactivé (réglage usine)
i_xClk	BOOL	TRUE : Signal d'horloge actif. FALSE : Désactivé (réglage usine)
i_xSet	BOOL	TRUE : Active la sortie de retournement. FALSE : Désactivé (réglage usine)
i_xRst	BOOL	TRUE : Réinitialise la sortie de retournement. FALSE : Désactivé (réglage usine)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction JK_FlipFlop :

Sortie	Type de données	Description
q_xQ	BOOL	Sortie de retournement

Limitations

Dans le retournement JK, les entrées i_xSet et i_xRst ont une priorité supérieure aux entrées i_xJ et i_xK. Quand les deux entrées i_xSet et i_xRst sont à l'état soit FALSE /TRUE, la sortie du bloc fonction q_xQ dépend des entrées i_xJ, i_xK et i_xClk.

JK_FlipFlop_MasterSlave: Initialisation/réinitialisation de l'entrée de la sortie de retournement

Contenu de ce chapitre

Bloc fonction JK_FlipFlop_MasterSlave 123

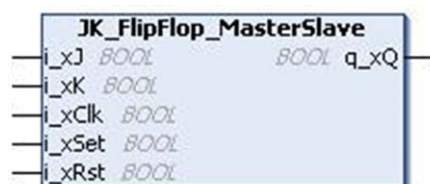
Vue d'ensemble

Ce chapitre décrit le bloc fonction JK_FlipFlop_MasterSlave.

Bloc fonction JK_FlipFlop_MasterSlave

Schéma d'affectation des broches

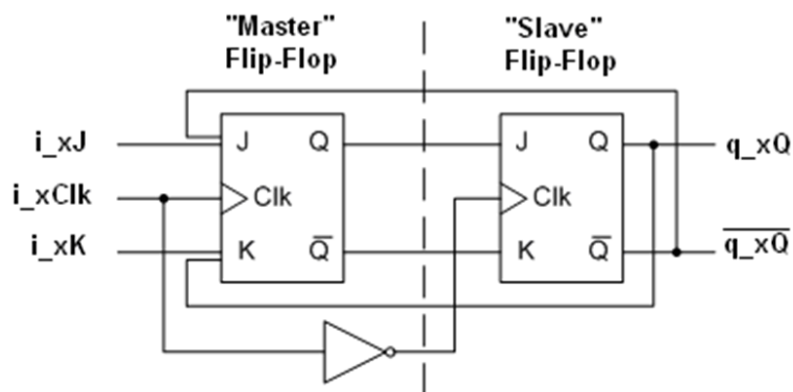
La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction JK_FlipFlop_MasterSlave :



Description fonctionnelle

Le bloc fonction JK_FlipFlop_MasterSlave assure la mise en œuvre de la table de vérité du retournement maître/esclave JK. La sortie du maître est capturée sur le front montant du signal d'horloge et la sortie de l'esclave est mise à jour sur le front descendant du signal d'horloge.

Ce schéma représente l'architecture interne du bloc fonction JK_FlipFlop_MasterSlave :



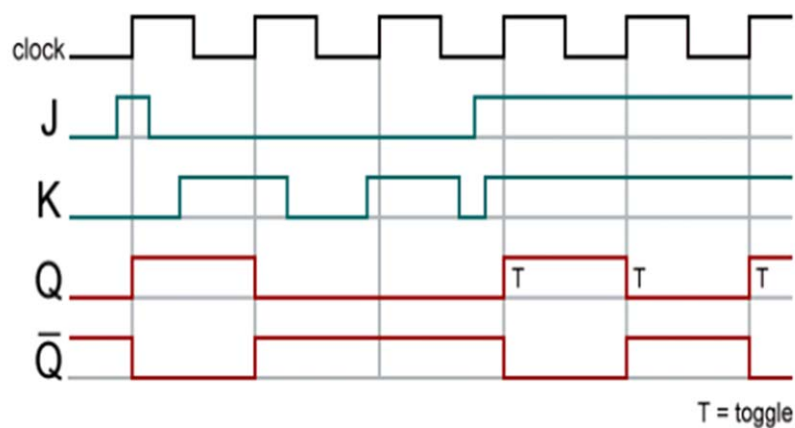
NOTE: La sortie complémentaire $\overline{q_xQ}$ n'est pas une sortie du bloc fonction.

Le bloc fonction JK_FlipFlop_MasterSlave fait référence à un retournement obéissant à la table de vérité suivante :

Activer	Reset	CLK	J	K	Q
1	0	X	X	X	1
0	1	X	X	X	0
1	1	X	X	X	1'
0	0	↑	0	0	Inv.
0	0	↑	1	0	1
0	0	↑	0	1	0
0	0	↑	1	1	Toggle
0	0	0	X	X	Inv.

L'entrée Reset (i_xRst) réinitialise la sortie du retournement q_xQ , alors que l'entrée Set (i_xSet) active la sortie du retournement q_xQ .

Table de vérité sous forme de diagramme temporel :



Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction JK_FlipFlop_MasterSlave :

Entrée	Type de données	Description
i_xJ	BOOL	TRUE : Entrée i_xJ active. FALSE : Désactivé (réglage usine)
i_xK	BOOL	TRUE : Entrée i_xK active. FALSE : Désactivé (réglage usine)
i_xClk	BOOL	TRUE : Signal d'horloge actif. FALSE : Désactivé (réglage usine)
i_xSet	BOOL	TRUE : Active la sortie de retournement. FALSE : Désactivé (réglage usine)
i_xRst	BOOL	TRUE : Réinitialise la sortie de retournement. FALSE : Désactivé (réglage usine)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction JK_FlipFlop_MasterSlave :

Sortie	Type de données	Description
q_xQ	BOOL	Sortie du retournement (True /False)

Limitations

Dans le retournement maître/esclave JK, les entrées `i_xSet` et `i_xRst` ont une priorité supérieure aux entrées `i_xJ` et `i_xK`. Quand les deux entrées `i_xSet` et `i_xRst` sont à l'état soit FALSE /TRUE, la sortie du bloc fonction `q_xQ` dépend des entrées `i_xJ`, `i_xK` et `i_xClk`.

RS_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement

Contenu de ce chapitre

Bloc fonction RS_FlipFlop 126

Vue d'ensemble

Ce chapitre décrit le bloc fonction RS_FlipFlop.

Bloc fonction RS_FlipFlop

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction RS_FlipFlop :



Description fonctionnelle

Le bloc fonction RS_FlipFlop met en œuvre la table de vérité pour le retournement RS avec priorité à la réinitialisation.

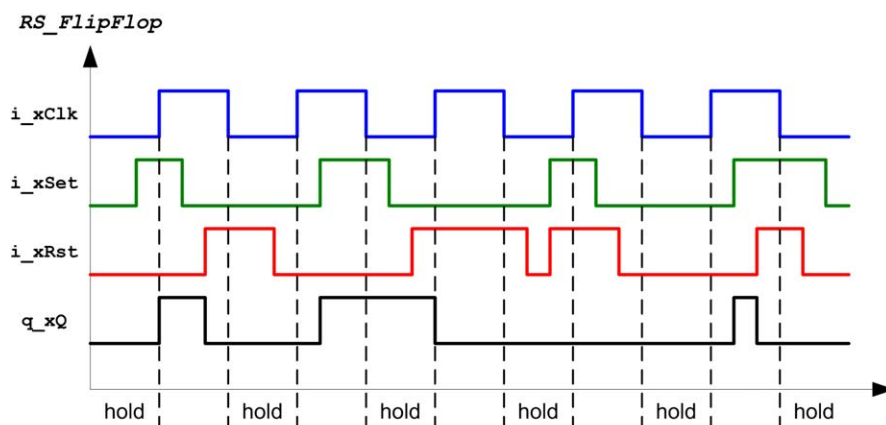
Le bloc fonction RS_FlipFlop fait référence à un retournement obéissant à la table de vérité suivante :

i_xClk	i_xSet	i_xRst	q_xQ (n+1)
0	X	X	Q(n)
1	0	0	Q(n)
1	0	1	0
1	1	0	1
1	1	1	0

r "n" est l'état présent et (n+1) l'état suivant.

Il a deux entrées, une entrée d'activation ou i_xSet et une entrée de réinitialisation, ou i_xRst. Il a aussi une sortie q_xQ. Quand les deux entrées d'activation et de réinitialisation sont à l'état haut, la priorité est donnée à l'entrée de réinitialisation (i_xSet=1 et i_xRst=1).

Table de vérité sous forme de diagramme temporel :



Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction *RS_FlipFlop* :

Entrée	Type de données	Description
i_xClk	BOOL	TRUE : Signal d'horloge actif. FALSE : Désactivé (réglage usine)
i_xSet	BOOL	TRUE : Active la sortie de retournement. FALSE : Désactivé (réglage usine)
i_xRst	BOOL	TRUE : Réinitialise la sortie de retournement. FALSE : Désactivé (réglage usine)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction *RS_FlipFlop* :

Sortie	Type de données	Description
q_xQ	BOOL	Sortie de retournement

SR_FlipFlop: Activation/réinitialisation de l'entrée/sortie de retournement

Contenu de ce chapitre

Bloc fonction SR_FlipFlop 128

Vue d'ensemble

Ce chapitre décrit le bloc fonction SR_FlipFlop.

Bloc fonction SR_FlipFlop

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction SR_FlipFlop :



Description fonctionnelle

Le bloc fonction SR_FlipFlop met en œuvre la table de vérité pour le retournement SR avec priorité à l'activation.

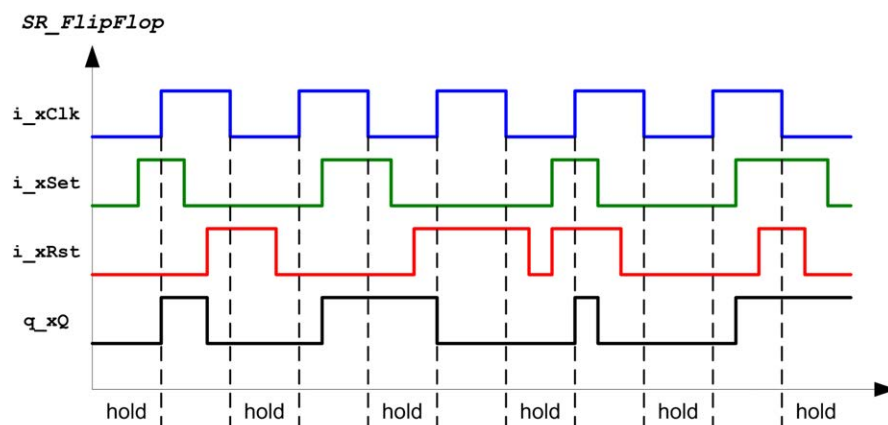
Le bloc fonction SR_FlipFlop fait référence à un retournement obéissant à la table de vérité suivante :

i_xClk	i_xSet	i_xRst	q_xQ (n+1)
0	X	X	Q(n)
1	0	0	Q(n)
1	0	1	0
1	1	0	1
1	1	1	1

r "n" est l'état présent et (n+1) l'état suivant.

Il a deux entrées, une entrée d'activation ou i_xSet et une entrée de réinitialisation, ou i_xRst. Il a aussi une sortie q_xQ. Quand les deux entrées d'activation et de réinitialisation sont à l'état haut, la priorité est donnée à l'entrée d'activation (i_xSet=1 et i_xRst=1).

Table de vérité sous forme de diagramme temporel :



Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction *SR_FlipFlop* :

Entrée	Type de données	Description
i_xClk	BOOL	TRUE : Signal d'horloge actif. FALSE : Désactivé (réglage usine)
i_xSet	BOOL	TRUE : Active la sortie de retournement. FALSE : Désactivé (réglage usine)
i_xRst	BOOL	TRUE : Réinitialise la sortie de retournement. FALSE : Désactivé (réglage usine)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction *SR_FlipFlop* :

Sortie	Type de données	Description
q_xQ	BOOL	Sortie de retournement

Toggle_FlipFlop: Basculement de l'entrée/sortie de retournement

Contenu de ce chapitre

Bloc fonction Toggle_FlipFlop 130

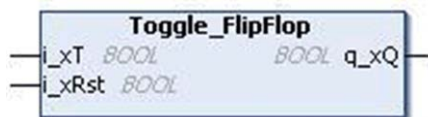
Vue d'ensemble

Ce chapitre décrit le bloc fonction Toggle_FlipFlop.

Bloc fonction Toggle_FlipFlop

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Toggle_FlipFlop :



Description fonctionnelle

Le bloc fonction Toggle_FlipFlop met en œuvre la table de vérité pour le retournement T(Toggle) avec priorité à l'activation.

Le bloc fonction Toggle_FlipFlop est un type de retournement obéissant à la table de vérité suivante :

i_xRst	i_xT _(n-1)	i_xT _(n)	q_xQ _(n)	q_xQ _(n+1)
0	0	0	X	Q _(n)
0	0	1	0	1
0	0	1	1	0
0	1	x	x	Q _(n)
1	x	x	x	0

r "n" est l'état présent et (n+1) l'état suivant.

Il a deux entrées, soit l'entrée i_xT et une entrée de réinitialisation, ou i_xRst. Il a aussi une sortie q_xQ.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Toggle_FlipFlop :

Entrée	Type de données	Description
i_xT	BOOL	Le front montant 0...1 bascule le retournement. Réglage usine : FALSE
i_xRst	BOOL	TRUE : Réinitialise la sortie de retournement. FALSE : Désactivé (réglage usine)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Toggle_FlipFlop :

Sortie	Type de données	Description
q_xQ	BOOL	Sortie de retournement

Fonctions mathématiques

Contenu de cette partie

Analysis: Calcul de valeurs intégrales et dérivées	133
Frequency_Multiplier: Mise en œuvre de 32 clignoteurs	135
Frequency_Output: Mise en œuvre d'une fréquence	139
Normalizer_With_Limiter: Mise à l'échelle de l'entrée Minimale et Maximale	144
ONE_SEC_PULSE: Fourniture d'impulsions à chaque seconde	147
Quantizer: Numérisation de la valeur d'entrée pour l'intervalle	148
Signal_Saturation: Limitation aux limites supérieure et inférieure de saturation	150
Signal_Statistics: Calcul de Maximum, de Minimum, de Moyenne et de Variance	154
Check_Divisor: Vérification de condition de division par zéro	157

Vue d'ensemble

Cette partie décrit la famille de Fonctions mathématiques.

Analysis: Calcul de valeurs intégrales et dérivées

Contenu de ce chapitre

Bloc fonction `Analysis` 133

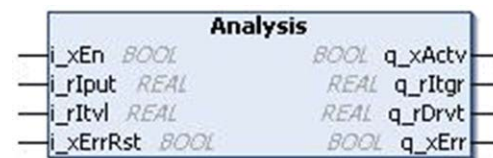
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Analysis`.

Bloc fonction `Analysis`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Analysis` :



Description fonctionnelle

Le bloc fonction `Analysis` calcule les valeurs intégrale et dérivée d'une série d'entrées. La sortie commence à zéro au front montant de `i_xEn`. La valeur intégrale augmente par multiples de l'entrée d'intervalle.

A chaque scrutation la sortie intégrale et la sortie dérivée sont mises à jour à partir de la valeur d'intervalle.

Une erreur est détectée si la valeur d'intervalle est égale ou inférieure à zéro ou si l'entrée est hors de la plage ou encore si la sortie intégrale ou dérivée dépasse $3,4e^{+38}$.

Intégrale = Intégrale + (Entrée en cours + Entrée précédente) / 2 * Intervalle.

Dérivée = (Entrée en cours + Entrée précédente) / Intervalle.

Exemple

Entrée = 10 (Entrée précédente : 0), Intervalle = 10, la sortie après le premier cycle d'exécution est comme suit :

- Intégrale = $0 + (10+0) / 2 * 10 = 50$
- Dérivée = $(10-0) / 10 = 1$

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Analysis` :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Bloc fonction activé FALSE : Bloc fonction désactivé
i_rIput	REAL	Valeur d'entrée Plage : $1,17e^{-38}$... $3,4e^{38}$
i_rItvl	REAL	Valeur d'entrée Plage : $1,17e^{-38}$... $3,4e^{38}$
i_xErrRst	BOOL	TRUE : Réinitialiser l'erreur détectée. (Sur le front montant) (optionnel)

Description des broches de sortie

Ce tableau décrit les broches de sortie :

sortie	Type de données	Description
q_xActv	BOOL	Sortie d'état du bloc fonction
q_rItgr	REAL	Valeur intégrale Plage : $1,17e^{-38}$... $3,4e^{38}$
q_rDrvt	REAL	Valeur dérivée Plage : $1,17e^{-38}$... $3,4e^{38}$
q_xErr	BOOL	TRUE : Entrée $i_rItvl \leq 0$ ou $i_rIput < 1,17e^{-38}$ ou $i_rIput > 3,4e^{38}$ ou $q_rItgr > 3,4e^{38}$ ou $q_rDrvt > 3,4e^{38}$ FALSE : Aucune erreur détectée

Frequency_Multiplier: Mise en œuvre de 32 clignoteurs

Contenu de ce chapitre

Bloc fonction <code>Frequency_Multiplier</code>	135
Sans maintien Description	136
Fonctionnalité avec description de condition.....	137

Vue d'ensemble

Ce chapitre décrit le bloc fonction `Frequency_Multiplier`.

Bloc fonction `Frequency_Multiplier`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Frequency_Multiplier` :



Description fonctionnelle

Le bloc fonction `Frequency_Multiplier` met en œuvre 32 clignoteurs représentés par les bits de sortie.

Sur chaque front montant du signal d'activation, la sortie du clignoteur commence à zéro. Le bit de poids le plus faible change d'état après une durée égale à la base de temps. Le deuxième bit clignote à une fréquence moitié de la fréquence initiale. Le troisième bit clignote à la moitié de la fréquence du deuxième et ainsi de suite, jusqu'à la réinitialisation du signal d'activation. Si l'entrée `i_xHold` est activée, l'état en cours des clignoteurs est Hold (maintien). Si des clignoteurs de type `BOOL` sont nécessaires, il est possible d'utiliser le bloc fonction `DWORD_AS_BIT` (bibliothèque Util).

La sortie est réinitialisée sur le front montant de l'entrée Enable.

Exemple

Fréquences (Enable = TRUE, Timebase = t#100ms, Hold = FALSE)

`DWORD_AS_BIT` (Entrée = Sortie de fréquence)

`DWORD_AS_BIT.B00` clignote toutes les 100 ms

`DWORD_AS_BIT.B01` clignote toutes les 200 ms

`DWORD_AS_BIT.B02` clignote toutes les 400 ms

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Frequency_Multiplier` :

Entrée	Type de données	Description
<code>i_xEn</code>	BOOL	TRUE : FB activé FALSE : Désactivé
<code>i_tBase</code>	TIME	Période de temps Plage : 1...4294967295 ms (≥ temps de cycle de l'automate)
<code>i_xHold</code>	BOOL	TRUE : Active FALSE : Désactivé

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Frequency_Multiplier` :

Sortie	Type de données	Description
<code>q_xActiv</code>	BOOL	TRUE : Bloc fonction activé FALSE : Désactivé
<code>q_dwOuput</code>	DWORD	Etat de sortie Plage : 0...4294967295

Sans maintien Description

Sans maintien

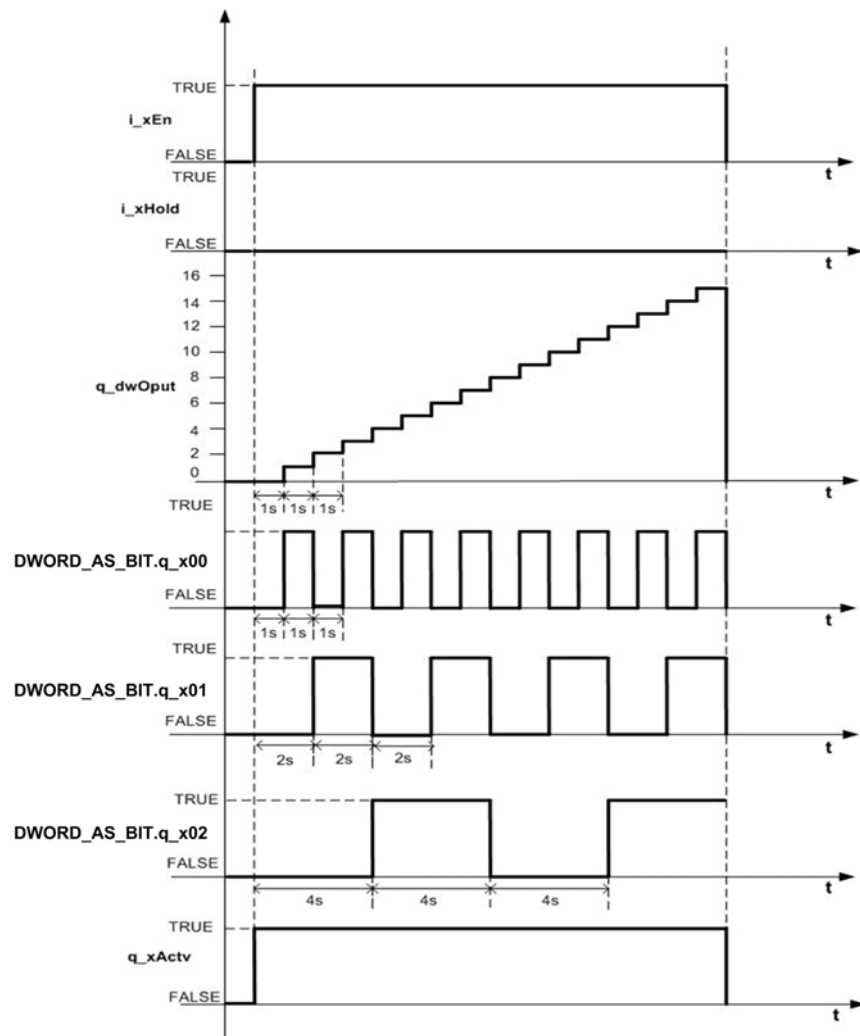
Si l'entrée de la broche `i_xEn` est à l'état haut, le temps `i_tBase` est mis à 1 s et l'entrée `i_xHold` a la valeur `FALSE`, `q_dwOuput` est augmenté d'une unité, après l'achèvement de la période `i_tBase`.

`DWORD_AS_BIT (Input: = Frequency_Multiplier.q_dwOuput):`

- `DWORD_AS_BIT.q_x00` est ON après 1 s d'activation du bloc fonction pour la durée de la base de temps Time base
- `DWORD_AS_BIT.q_x01` est ON après 2 s d'activation du bloc fonction pour 2 * durée de la base de temps Time base
- `DWORD_AS_BIT.q_x02` est ON après 4 s d'activation du bloc fonction pour 4 * durée de la base de temps Time base

Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction `Frequency_Multiplier` sans maintien d'entrée :



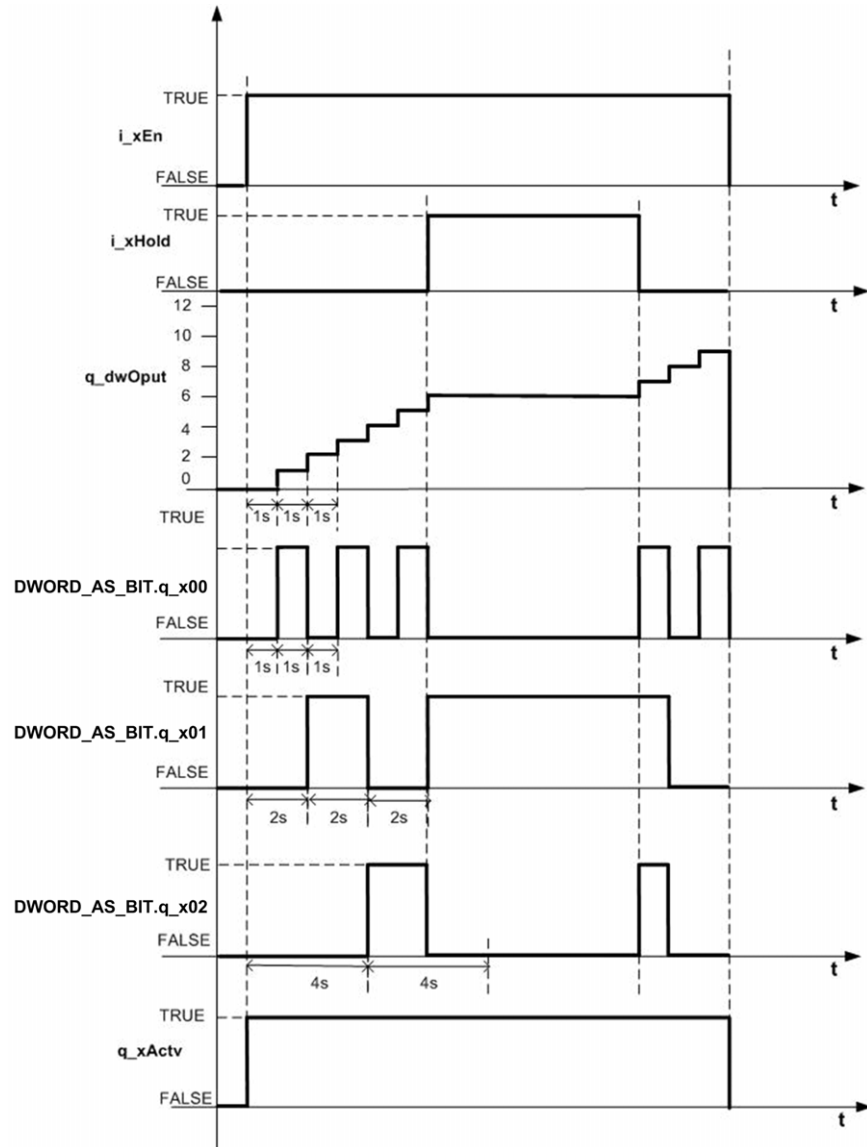
Fonctionnalité avec description de condition

Description de la fonctionnalité

Si l'entrée de la broche `i_xEn` est à l'état haut, le temps `i_tBase` est réglé à 1s et l'entrée `i_xHold` a la valeur TRUE, alors `q_dwOput` maintient l'état précédent des clignoteurs. FB reprend son fonctionnement normal quand l'entrée `i_xHold` passe à FALSE.

Diagramme de temps

Cette figure présente le diagramme de temps pour le bloc fonction `Frequency_Multiplier` avec maintien d'entrée :



Frequency_Output: Mise en œuvre d'une fréquence

Contenu de ce chapitre

Bloc fonction Frequency_Output 139

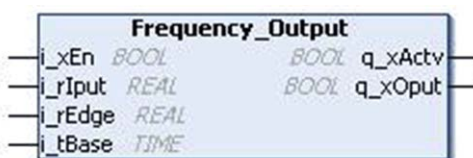
Vue d'ensemble

Ce chapitre décrit le bloc fonction Frequency_Output.

Bloc fonction Frequency_Output

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Frequency_Output :



Description fonctionnelle

Le bloc fonction Frequency_Output assure la mise en œuvre des fréquences.

Un signal positif sur i_xEn active le bloc. Le signal de sortie commence à TRUE et maintient ce signal pendant une durée égale à $(TimeBase * i_rIput)$ puis passe à FALSE pendant une durée égale à $TimeBase * (1-i_rIput)$.

Si le signal d'entrée entre dans la plage frontière 0 à i_rEdge, le signal de sortie n'alterne plus et reste continuellement à l'état FALSE. S'il entre dans la plage 1-i_rEdge à 1, le signal de sortie a en permanence la valeur TRUE.

Si le signal i_xEn est supprimé, le signal de sortie reste à sa valeur en cours jusqu'au redémarrage du bloc fonction par un signal positif sur i_xEn.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Frequency_Output :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : FB activé FALSE : Désactivé
i_rIput	REAL	Valeur d'entrée Plage : 0.0...1.0
i_rEdge	REAL	Valeur d'entrée Plage : 0.0...1.0
i_tBase	TIME	Période de temps Plage : 0...4294967295 ms

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Frequency_Output` :

sortie	Type de données	Description
q_xActiv	BOOL	Etat actif du bloc fonction
q_xOput	BOOL	Valeur de sortie

Instanciation et exemple d'utilisation

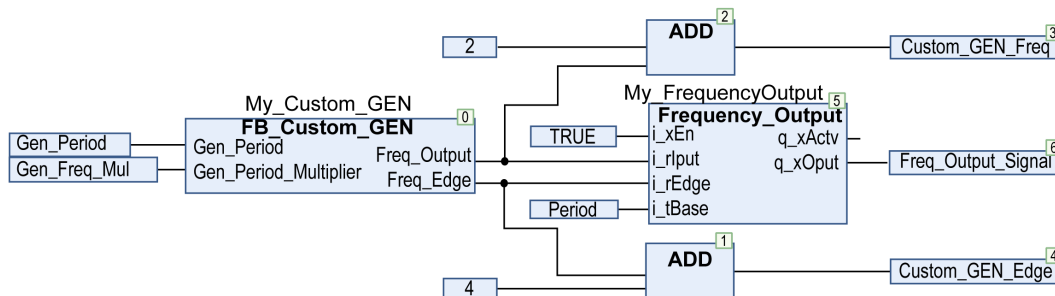
L'exemple utilise les générateurs de signaux personnalisés suivants qui créent 2 signaux :

- Un signal de période 5 secondes qui sera utilisé comme entrée du bloc fonction `Frequency_Output`.
- Un signal de période 50 secondes qui sera utilisé comme front pour le bloc fonction `Frequency_Output`.

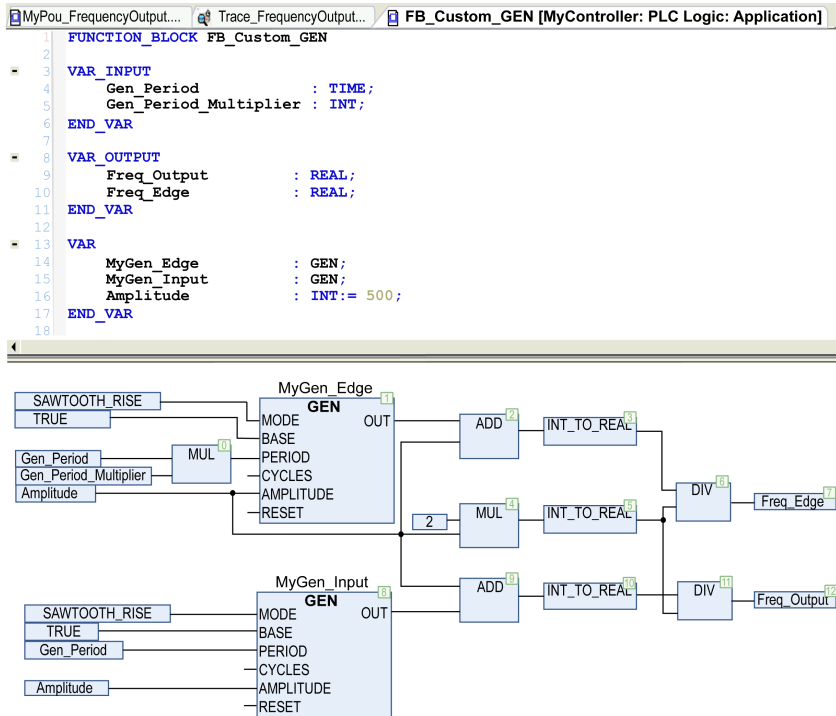
Ce POU a une période de 10 ms dans le MAST.

```

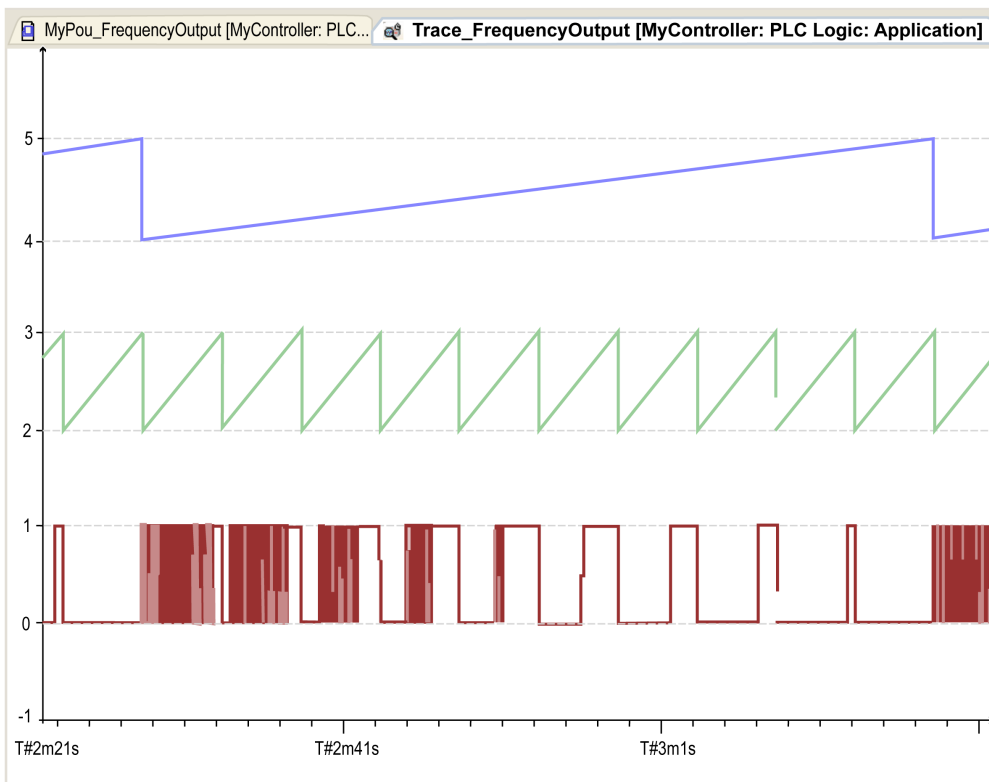
1 PROGRAM MyPou_FrequencyOutput
2 VAR
3   FlagDemarrer      : BOOL;
4
5   My_Custom_GEN      : FB_Custom_GEN;
6   Gen_Period         : TIME := T#5000MS;
7   Gen_Freq_Mul       : INT := 10;
8
9   My_FrequencyOutput : Frequency_Output;
10  Period             : TIME := T#100MS;
11  Freq_Output_Signal : BOOL; TIME:=T#5000MS;
12  Custom_GEN_Freq    : REAL;
13  Custom_GEN_Edge    : REAL;
14  Grid_Draw_I       : REAL;
15 END_VAR
    
```



Cette figure présente le générateur de signal personnalisé :



Cet exemple donne les résultats suivants :



Bleu Signal i_rEdge , période de 50 secondes.

Vert Signal i_rInput , période de 5 secondes.

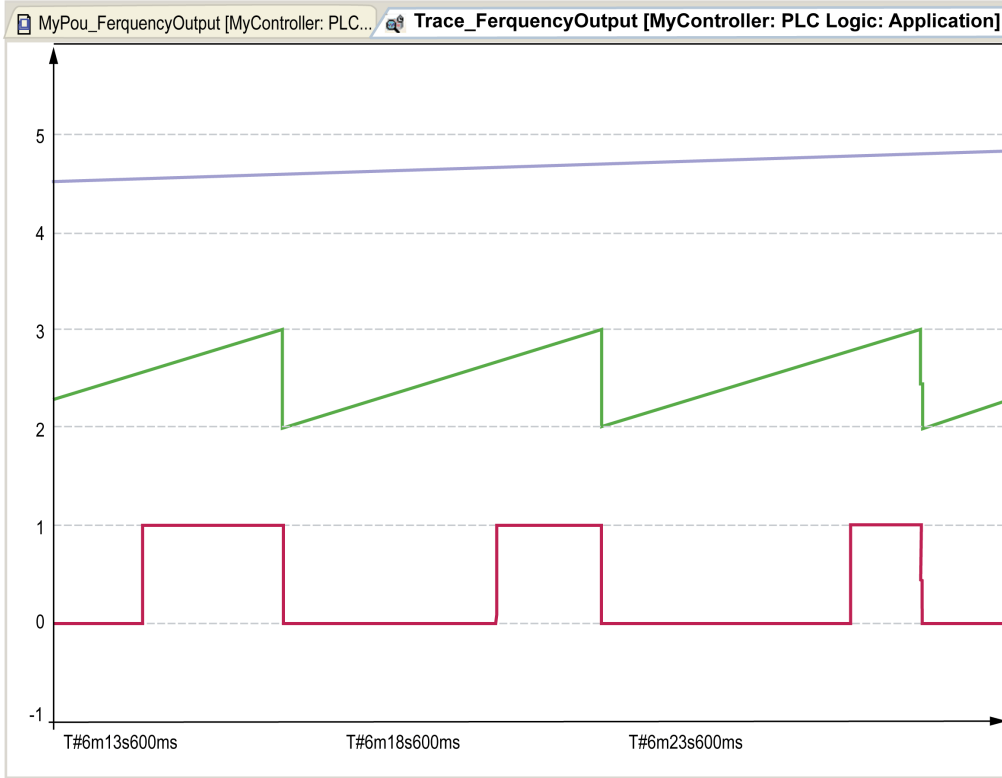
Rouge $q_xOutput$, sortie du bloc fonction `Frequency_Output`.

Ce tableau présente la table de vérité :

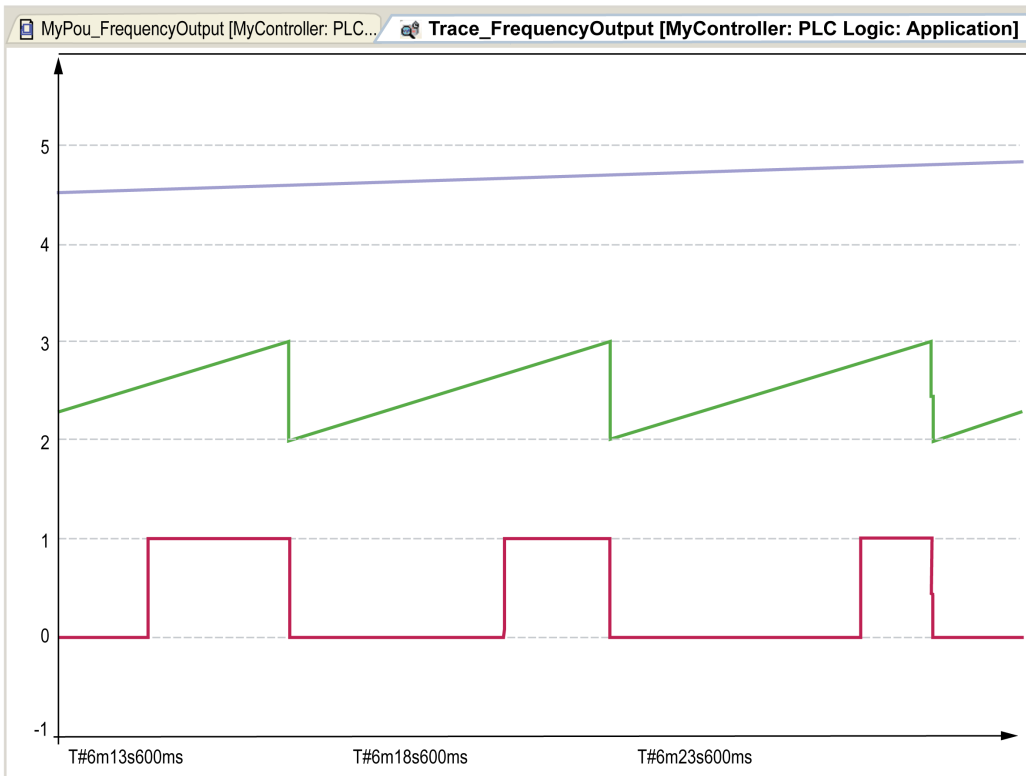
Niveau de front	Niveau d'entrée	Sortie
$i_rEdge < 0,5$	$i_rInput < i_rEdge$	FALSE

Niveau de front	Niveau d'entrée	Sortie
	$i_rEdge < i_rInput < (1-i_rEdge)$	PWM; rapport cyclique = i_rInput
	$(1 - i_rEdge) < i_rInput$	TRUE
$i_rEdge \geq 0,5$	$i_rInput < i_rEdge$	FALSE
	$i_rEdge \leq i_rInput$	TRUE

Exemple avec $i_rEdge < 0,5$:

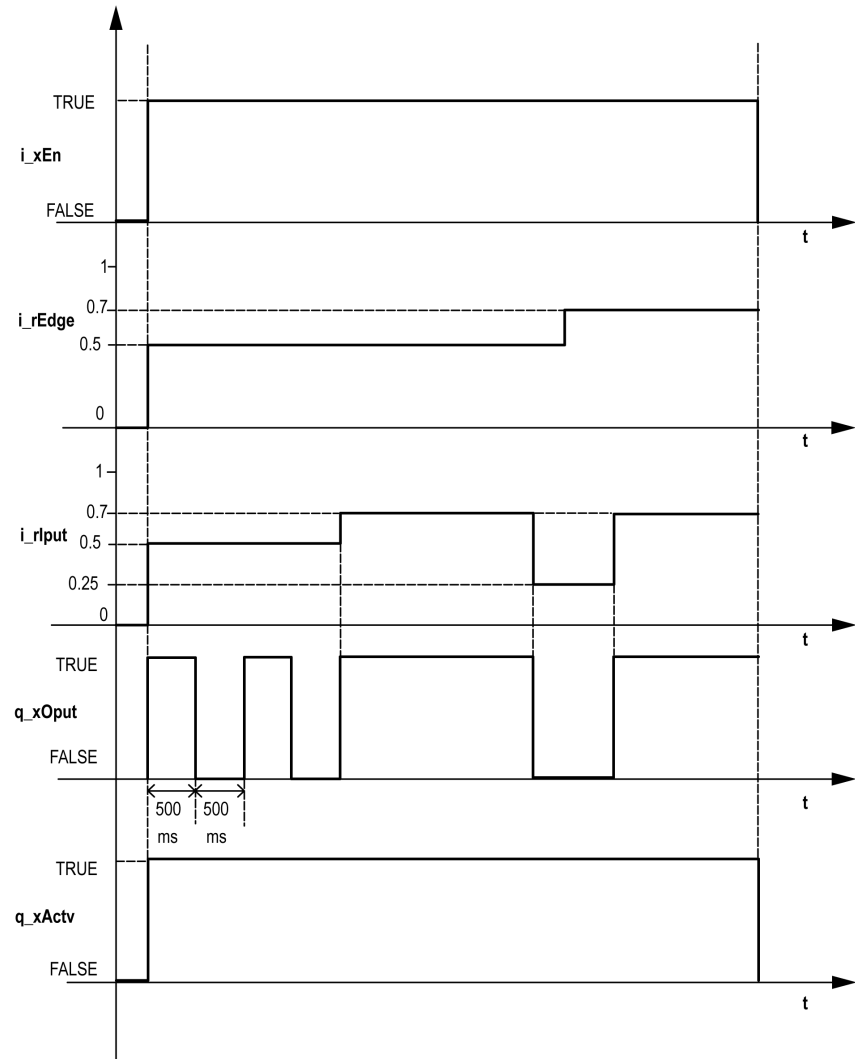


Exemple avec $i_rEdge \geq 0,5$:



Exemple : Si l'entrée à la broche i_xEn est à l'état haut, l'entrée i_rIput et au départ l'entrée i_rEdge sont à 0,5 et l'entrée i_tBase est définie à 1 sec, alors q_xOput est défini à 500 ms et réinitialisée pour une durée de 500 ms. Après quelque temps l'entrée i_rEdge passe à 0,7.

Cette figure présente le diagramme de temps de l'exemple ci-dessus :



Normalizer_With_Limiter: Mise à l'échelle de l'entrée Minimale et Maximale

Contenu de ce chapitre

Bloc fonction `Normalizer_With_Limiter` 144

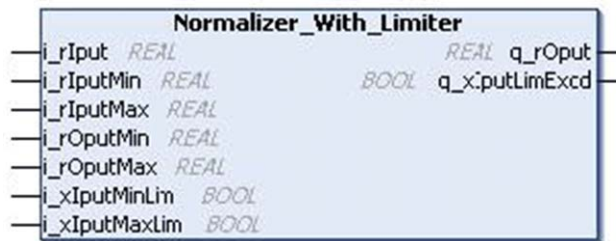
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Normalizer_With_Limiter`.

Bloc fonction `Normalizer_With_Limiter`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Normalizer_With_Limiter` :



Description fonctionnelle

Le bloc fonction `Normalizer_With_Limiter` met à l'échelle la valeur d'entrée maximale et minimale selon la plage de sortie minimale et maximale. La valeur d'entrée peut être limitée aux valeurs de sortie minimale et maximale.

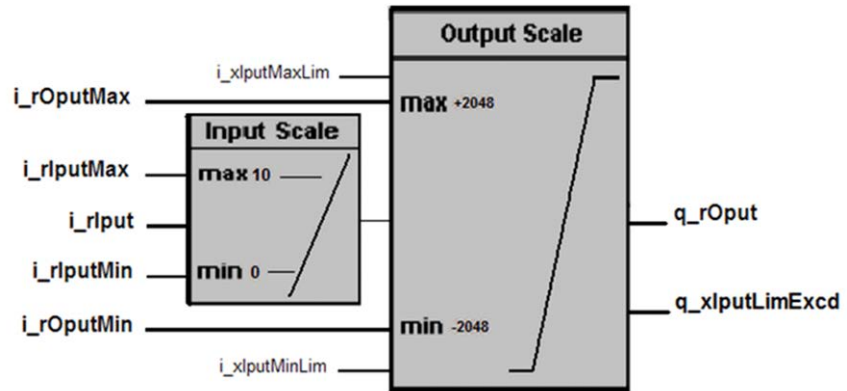
La valeur de sortie peut être limitée aux valeurs de sortie minimale et maximale par `i_xIputMinLim` et `i_xIputMaxLim`.

Quand l'entrée dépasse la valeur limite d'entrée maximale/minimale, `q_xIputLimExcd` a la valeur TRUE.

Complément mathématique

$$q_rOput = \frac{(i_rIput - i_rIputMin) \times (i_rOputMax - i_rOputMin)}{(i_rIputMax - i_rIputMin)} + i_rOputMin$$

Exemple



i_xIputMaxLim	i_xIputMinLim	i_rIput	q_rOput	q_xIputLimExcd
FALSE	FALSE	12.5	3072	TRUE
TRUE	FALSE	12.5	2048	TRUE
TRUE	FALSE	10	2048	FALSE
TRUE	FALSE	7.5	1024	FALSE
TRUE	TRUE	5	0	FALSE
FALSE	TRUE	2.5	-1024	FALSE
FALSE	TRUE	0	-2048	FALSE
FALSE	TRUE	-2.5	-2048	TRUE
FALSE	FALSE	-2.5	-3072	TRUE

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Normalizer_With_Limiter :

Entrée	Type de données	Description
i_rIput	REAL	Valeur d'entrée Plage : $\pm 3,4e^{+38}$
i_rIputMin	REAL	Entrée minimale Plage : $\pm 3,4e^{+38}$
i_rIputMax	REAL	Valeur d'entrée maximale Plage : $\pm 3,4e^{+38}$
i_rOputMin	REAL	Valeur de sortie minimale Plage : $\pm 3,4e^{+38}$
i_rOputMax	REAL	Valeur de sortie maximale Plage : $\pm 3,4e^{+38}$
i_xIputMinLim	BOOL	TRUE : Limite activée pour la valeur d'entrée minimale FALSE : Limite désactivée
i_xIputMaxLim	BOOL	TRUE : Limite activée pour la valeur d'entrée maximale FALSE : Limite désactivée

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Normalizer_With_Limiter` :

Sortie	Type de données	Description
<code>q_rOput</code>	REAL	Mise à l'échelle de la valeur de sortie Plage : $\pm 3,4e^{+38}$
<code>q_xIputLimExcd</code>	BOOL	La valeur d'entrée dépasse la valeur limite

ONE_SEC_PULSE: Fourniture d'impulsions à chaque seconde

Contenu de ce chapitre

Bloc fonction ONE_SEC_PULSE 147

Vue d'ensemble

Ce chapitre décrit le bloc fonction ONE_SEC_PULSE.

Bloc fonction ONE_SEC_PULSE

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction ONE_SEC_PULSE :



Description fonctionnelle

Le bloc fonction ONE_SEC_PULSE génère des impulsions d'une durée de une seconde sur la sortie q_xOput.

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction ONE_SEC_PULSE :

sortie	Type de données	Description
q_xOput	BOOL	Impulsion de sortie Etat de sortie = TRUE pendant 1 cycle de tâche Fréquence = 1 Hz

Instanciation et exemple d'utilisation

Cette figure ci-dessus présente une instance du bloc fonction ONE_SEC_PULSE :



Quantizer: Numérisation de la valeur d'entrée pour l'intervalle

Contenu de ce chapitre

Bloc fonction `Quantizer` 148

Vue d'ensemble

Ce chapitre décrit le bloc fonction `Quantizer`.

Bloc fonction `Quantizer`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Quantizer` :



Description fonctionnelle

Le bloc fonction `Quantizer` discrétise la valeur d'entrée (-32768 à 32767) pour un intervalle donné.

Si l'entrée est supérieure à la plage d'entrée, la sortie d'erreur détectée à la valeur TRUE et la sortie affiche une valeur zéro.

Complément mathématique

$$q_rOput = i_rInput \times \text{ROUND} \left(\frac{i_rInput}{i_rItvl}; 0 \right)$$

Avec `Itvl` = intervalle.

Exemple

Entrée	Intervalle	Sortie
32766.7	1	32767 Erreur détectée = FALSE
32768	15	0 Erreur détectée = TRUE
-32768	20	-32760 Erreur détectée = FALSE
36.89	15	30

Entrée	Intervalle	Sortie
		Erreur détectée = FALSE
-47.98	-10	-50 Erreur détectée = FALSE
-42.14	-10	-40 Erreur détectée = FALSE
3456.78	80	3440 Erreur détectée = FALSE
De -4,99 à 4,99	10	0
De 5 à 14,99	10	10

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Quantizer` :

Entrée	Type de données	Description
<code>i_rInput</code>	REAL	Valeur d'entrée Plage : -32768...32767
<code>i_rItvl</code>	REAL	Valeur d'entrée d'intervalle de quantification Plage : $\pm 3,4e^{+38}$
<code>i_xErrRst</code>	BOOL	Réinitialiser l'erreur détectée. (Sur le front montant) (optionnel)

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Quantizer` :

sortie	Type de données	Description
<code>q_rOput</code>	REAL	Valeur de sortie Plage : -32768...32767
<code>q_xErr</code>	BOOL	TRUE : Limite d'entrée dépassée FALSE : Aucune erreur détectée

Signal_Saturation: Limitation aux limites supérieure et inférieure de saturation

Contenu de ce chapitre

Bloc fonction Signal_Saturation 150

Vue d'ensemble

Ce chapitre décrit le bloc fonction Signal_Saturation.

Bloc fonction Signal_Saturation

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Signal_Saturation :



Description fonctionnelle

Le bloc fonction Signal_Saturation limite le signal d'entrée à la limite de saturation supérieure et inférieure.

Quand la valeur d'entrée basse est supérieure à la valeur haute, la sortie d'erreur détectée a la valeur TRUE et la sortie affiche zéro.

Quand l'entrée dépasse la valeur limite d'entrée haute/basse, q_xIputLimExcd a la valeur TRUE.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Signal_Saturation :

Entrée	Type de données	Description
i_rInput	REAL	Valeur d'entrée Plage : ±3,4e ⁺³⁸
i_rLow	REAL	Valeur d'entrée inférieure Plage : ±3,4e ⁺³⁸
i_rHigh	REAL	Valeur d'entrée supérieure Plage : ±3,4e ⁺³⁸
i_xErrRst	BOOL	Réinitialiser l'erreur détectée. (Sur le front montant) (Facultatif)

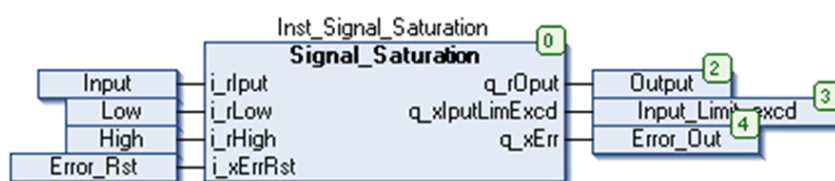
Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Signal_Saturation` :

Sortie	Type de données	Description
<code>q_rOput</code>	REAL	Valeur de sortie : $\pm 3,4e^{+38}$
<code>q_xIputLimExcd</code>	BOOL	TRUE : la valeur d'entrée dépasse la valeur limite.
<code>q_xErr</code>	BOOL	TRUE : limite d'entrée incorrecte FALSE : aucune erreur détectée.

Instanciation et exemple d'utilisation

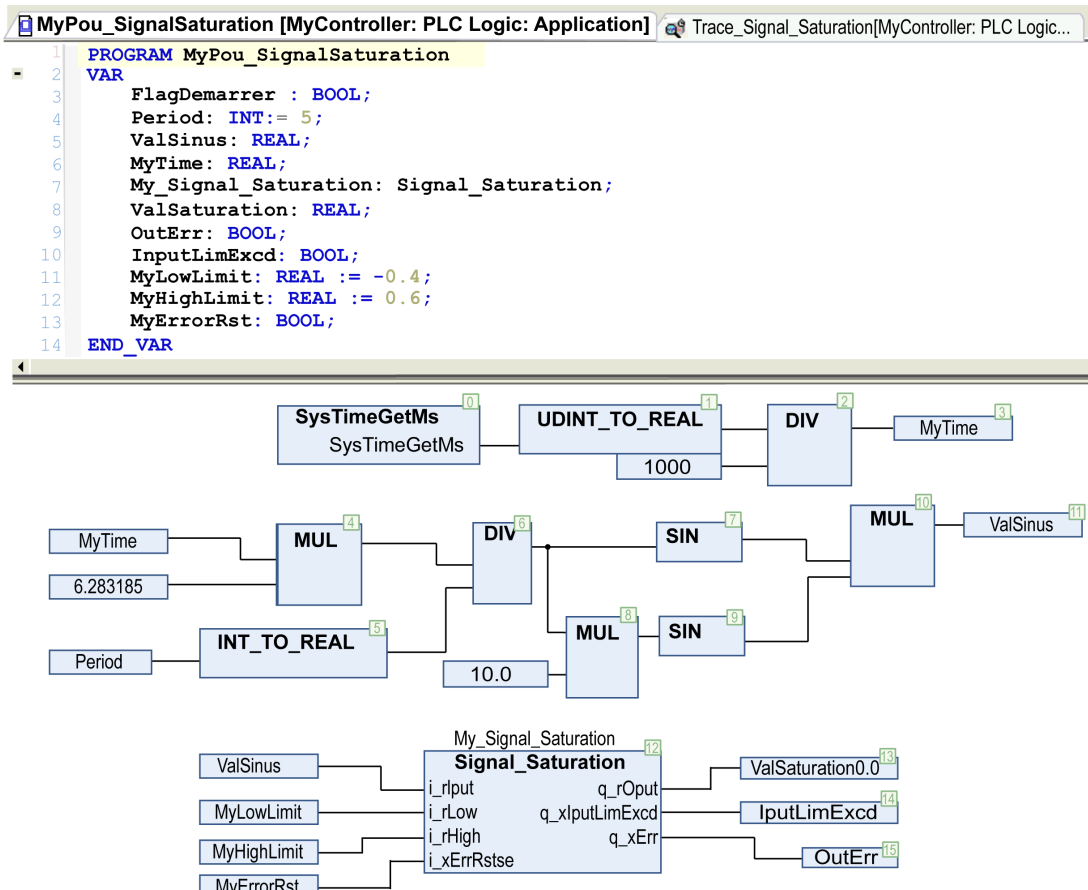
Cette figure présente une instance du bloc fonction `Signal_Saturation` :



Si l'entrée `i_rIput` est réglée sur 4, `i_rLow` sur 5 et `i_rHigh` sur 10, la valeur de sortie de saturation `i_rLow` prend la valeur 5 et `q_xIputLimExcd` la valeur TRUE.

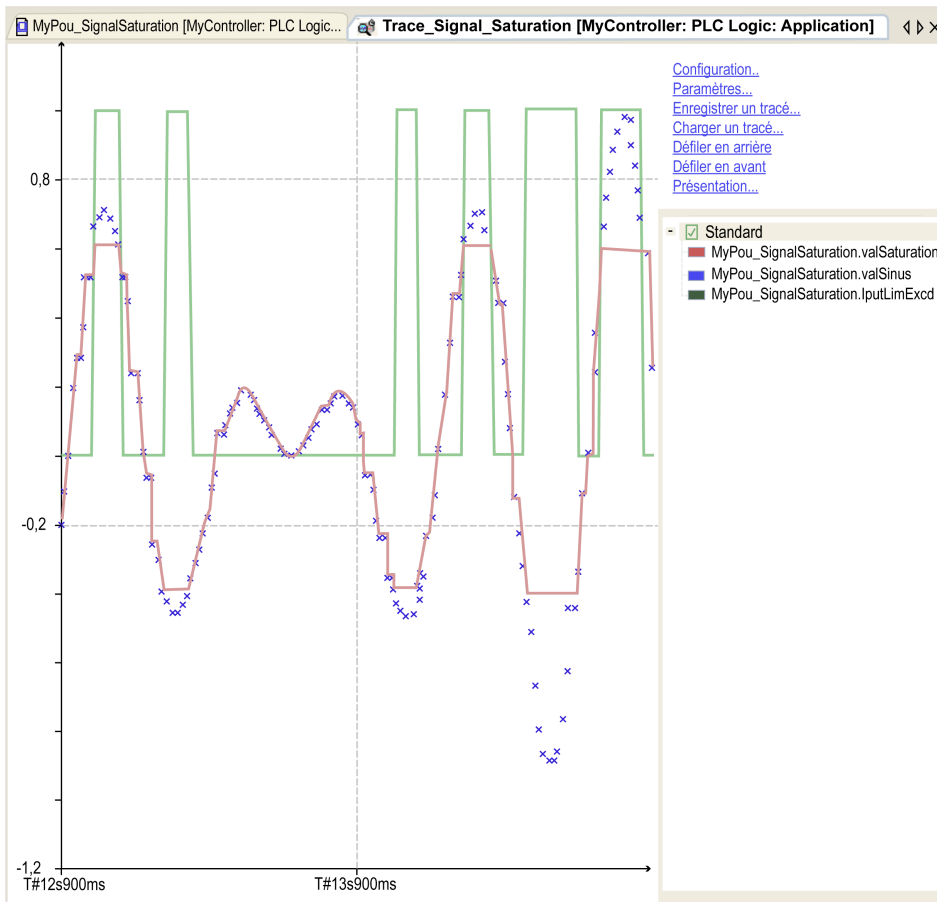
Exemple CFC

Cette figure présente l'exemple CFC d'une mise en œuvre de `Signal_Saturation` :



Chronogramme

Cette figure présente le chronogramme du bloc fonction Signal_Saturation :

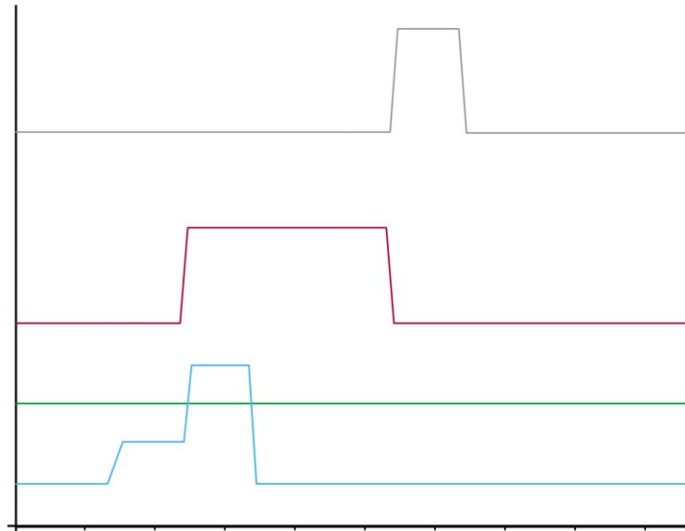


Bleu Signal d'entrée

Rouge Signal de sortie, limité dans la plage [Bas ; Haut]/[-0,4 ; +0,6].

Vert lputLimExcd, TRUE quand le signal d'entrée est hors plage.

Cette figure présente le chronogramme de la sortie q_xErr :



Bleu : Signal d'entrée i_rLow

Vert : Signal d'entrée i_rHigh

Rouge : Signal de sortie q_xErr , TRUE dès que i_rLow est supérieur à i_rHigh .

Gris : signal d'entrée $i_xErrRst$, réinitialise le signal de sortie q_xErr sur le front montant tant que i_rLow est inférieur à i_rHigh .

Signal_Statistics: Calcul de Maximum, de Minimum, de Moyenne et de Variance

Contenu de ce chapitre

Bloc fonction `Signal_Statistics` 154

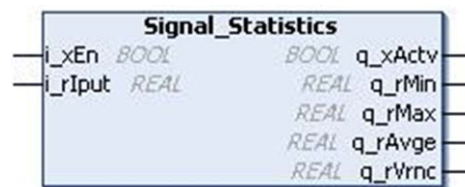
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Signal_Statistics`.

Bloc fonction `Signal_Statistics`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Signal_Statistics` :



Description fonctionnelle

Ce bloc fonction calcule le maximum, le minimum, la moyenne et la variance d'une série de valeurs d'entrée.

Ce bloc fonction prend en compte la valeur d'entrée dans chaque cycle de scrutation d'automate comme un échantillon.

Valeur minimum

La valeur de sortie minimum est la valeur constituant le minimum de tous les échantillons enregistrés.

Valeur maximum

La valeur de sortie maximum est la valeur constituant le maximum de tous les échantillons enregistrés.

Valeur moyenne

La valeur moyenne est égale à la somme des observations (échantillons) divisée par le nombre d'observations (échantillons).

$$\bar{x} = \frac{1}{n} \left(\sum_n x_n \right)$$

Avec :

- n = Nombre d'échantillons enregistrés
- Xn = Echantillons d'entrée
- \bar{x} = Sortie calculée

Valeur de variance

La variance est égale à la moyenne des carrés des échantillons moins le carré de la moyenne (valeur de sortie moyenne).

$$\bar{x} = \frac{1}{n} \left(\sum_1^n (x_n)^2 \right) - \left(\frac{1}{n} \sum_1^n x_n \right)^2$$

Avec :

- n = Nombre d'échantillons enregistrés
- Xn = Echantillons d'entrée
- \bar{x} = Sortie calculée

Exemple

- Statistiques (Enable: = TRUE, Input: = 1, 2)
- Sortie minimum = 1
- Sortie maximum = 2
- Moyenne = $(1 + 2) / 2 = 1,5$
- Variance = $((1 * 1 + 2 * 2) / 2) - (1,5 * 1,5) = 2,5 - 2,25 = 0,25$

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Signal_Statistics :

Entrée	Type de données	Description
i_xEn	BOOL	TRUE : Bloc fonction activé FALSE : Bloc fonction désactivé
i_rInput	REAL	Rang du bit Plage : $\pm 3,4e^{+38}$

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Signal_Statistics :

sortie	Type de données	Description
q_xActv	BOOL	Sortie d'état du bloc fonction
q_rMin	REAL	Valeur minimum Plage : $\pm 3,4e^{+38}$
q_rMax	REAL	Valeur maximum Plage : $\pm 3,4e^{+38}$
q_rAvge	REAL	Valeur moyenne Plage : $\pm 3,4e^{+38}$
q_rVrnc	REAL	Valeur de variance Plage : $\pm 3,4e^{+38}$

Check_Divisor: Vérification de condition de division par zéro

Contenu de ce chapitre

Bloc fonction Check_Divisor 157

Vue d'ensemble

Ce chapitre décrit le bloc fonction Check_Divisor.

Bloc fonction Check_Divisor

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Check_Divisor :



Description fonctionnelle

Le bloc fonction Check_Divisor vérifie un état de division par zéro.

Si le diviseur i_rDvsr est égal à zéro, la sortie $q_rChkDiv$ est égale à 1, sinon si le diviseur n'est pas égal à 0, la sortie est égale au diviseur.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Check_Divisor :

Entrée	Type de données	Description
i_rDvsr	REAL	Sortie de diviseur Plage : $\pm 3,4e^{+38}$

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Check_Divisor :

sortie	Type de données	Description
$q_rChkDiv$	REAL	Vérification de sortie de division Plage : $\pm 3,4e^{+38}$ NOTE: Cette sortie n'est pas valable pour la valeur 0.

Fonctions de conversion numérique

Contenu de cette partie

ArrayOfByte_TO_String: Conversion d'un tableau en format octet au format chaîne	159
DT_AS_WORD: Conversion de Date et Heure sous forme d'un mot.....	163
DWORD_AS_WORD: Division d'un mot double en deux mots	165
String_TO_ArrayOfByte: Tableau de sortie et valeur ASCII de la chaîne d'entrée	166
WORD_AS_DWORD: Décalage du mot de poids fort et ajout du mot de poids faible	170

Vue d'ensemble

Cette partie décrit la famille de Conversion numérique.

ArrayOfByte_TO_String: Conversion d'un tableau en format octet au format chaîne

Contenu de ce chapitre

Fonction ArrayOfByte_TO_String 159

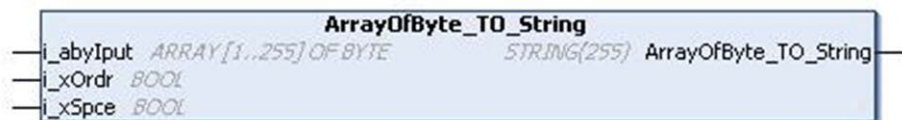
Vue d'ensemble

Ce chapitre décrit le bloc fonction ArrayOfByte_TO_String.

Fonction ArrayOfByte_TO_String

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches de la fonction ArrayOfByte_TO_String :



Description fonctionnelle

La chaîne de sortie [255] est l'ensemble des caractères de chaîne, correspondant à la valeur ASCII du tableau d'entrées donné au format Byte.

Si l'entrée ORDER a la valeur TRUE, l'ordre des caractères dans la chaîne de sortie correspond à l'ordre des octets dans le tableau d'entrée. Il existe donc une correspondance biunivoque entre l'ordre des octets d'entrée et l'ordre des caractères de chaîne renvoyés en sortie comme expliqué à l'exemple 1.

Si l'entrée ORDER a la valeur FALSE, l'ordre des caractères de la chaîne de sortie est tel que le caractère de chaîne correspondant à la valeur ASCII de input [1] est affiché en sortie en position[2] de output[1..255]. Le caractère de chaîne correspondant à la valeur ASCII de input[2] est affiché en sortie en position[1] de output[1..255].

De même le caractère de chaîne correspondant à la valeur ASCII de input[3] est affiché en sortie en position[4] de output[1..255]. Le caractère de chaîne correspondant à la valeur ASCII de input[4] est affiché en sortie en position[3] de output[1..255] comme expliqué dans l'exemple 2.

Si et uniquement si l'entrée ORDER a la valeur FALSE et si l'entrée SPACE a la valeur TRUE avec un nombre d'octets d'entrée impair, un caractère d'espace est ajouté avant le dernier caractère de la chaîne de sortie comme indiqué dans les exemples 3 et 4.

Mais si l'entrée ORDER a la valeur TRUE, l'entrée d'espace n'a aucune influence sur la sortie comme l'indique l'exemple 6.

Exemple 1

Entrée : ARRAY [1..255] OF BYTE = 72, 69, 76, 76, 79;

Order : TRUE

Space : FALSE

Chaîne : 'HELLO'

Comme indiqué ci-dessus, l'ordre des caractères dans la chaîne de sortie correspond à l'ordre des octets d'entrée, c'est-à-dire que la valeur d'octet en première position de Array[1..255] est 72, ce qui correspond à la valeur de chaîne en première position de sortie, soit H. La valeur d'octet en deuxième position de Array[1..255] est 69, ce qui correspond à la valeur chaîne en deuxième position de la sortie, soit E, etc.

Exemple 2

Entrée : ARRAY [1..255] OF BYTE = 65, 66, 67, 68, 69, 70, 71;

ByteOrder : FALSE

InsertSpace : FALSE

Chaîne : 'BADCFEG'

Comme indiqué ci-dessus, l'ordre des caractères dans la chaîne de sortie est modifié, c'est-à-dire que la valeur d'octet en première position dans Array [1 à 255] est 65, ce qui correspond à la valeur chaîne en deuxième position de la sortie, soit A. La valeur d'octet en deuxième position dans Array[1 à 255] est 66, ce qui correspond à la valeur chaîne en première position de la sortie, soit B. De même, la valeur d'octet en troisième position dans Array[1 à 255] est 67, ce qui correspond à la valeur de chaîne en quatrième position de la sortie, soit C. La valeur d'octet en quatrième position dans Array[1 à 255] est 68, ce qui correspond à la valeur de chaîne en troisième position de la sortie, soit D, etc.

Exemple 3

Entrée : ARRAY [1..255] OF BYTE = 72, 69, 76, 76, 79;

Order : FALSE

Space : TRUE

Chaîne : 'EHLL O'

Exemple 4

Entrée : ARRAY [1..255] OF BYTE = 65, 66, 67, 68, 69, 70, 71;

Order : FALSE

Space : TRUE

Chaîne : 'BADCFE G'

Comme indiqué ci-dessus dans les exemples 3 et 4, le nombre d'entrées est de 5 dans l'exemple 3 et de 7 dans l'exemple 4. Comme 5 et 7 sont impairs, l'entrée ORDER a la valeur FALSE et l'entrée SPACE a la valeur TRUE. Donc, les sorties de chaîne sont "EHLL O" et "BADCFE G" respectivement.

NOTE: Mais si le nombre d'octets en entrée est 255, l'entrée ORDER a la valeur FALSE et l'entrée SPACE a la valeur TRUE. L'entrée SPACE devient non significative, comme expliqué dans l'exemple 5 ci-dessus.

Exemple 5

Entrée : ARRAY [1..250] OF BYTE = 65 et ARRAY [251..255] OF BYTE = 66, 67, 68, 69, 70;

Order : TRUE

Space : TRUE/FALSE

Chaîne[1 à 250] : 'A' et chaîne[251 à 255] = 'CBEDF'

Comme indiqué dans l'exemple ci-dessus, si le nombre d'octets en entrée est 255, la chaîne de sortie n'est pas affectée par l'entrée SPACE.

Exemple 6

Entrée : ARRAY [1..255] OF BYTE = 65, 66, 67, 68, 69, 70, 71;

Order : TRUE

Space : TRUE

Chaîne : 'ABCDEFGF'

Comme indiqué ci-dessus, si l'entrée ORDER a la valeur TRUE, l'entrée SPACE devient non significative.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction ArrayOfByte_TO_String :

Entrée	Type de données	Description
i_abyInput	ARRAY [1..255] OF BYTE	Valeur d'entrée Plage : 1 à 255
i_xOrdr	BOOL	TRUE : dans l'ordre de l'entrée FALSE : intervertit les octets de poids fort et de poids faible.
i_xSpce	BOOL	TRUE : insère un espace quand i_xOrdr est à l'état bas. FALSE : aucun espace n'est inséré.

NOTE: I_xSpce insère un caractère d'espace juste avant le dernier caractère de la chaîne de sortie, quand l'entrée ORDER a la valeur FALSE et que le nombre d'octets en entrée est impair.

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction ArrayOfByte_TO_String :

Sortie	Type de données	Description
ArrayOfByte_TO_String	STRING (255)	Sortie de caractères de chaîne

NOTE: L'utilisateur doit obligatoirement définir une taille de [255] pour la chaîne de sortie, sinon la taille choisie par défaut est 80.

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction `ArrayOfByte_TO_String` :



Avec l'entrée ORDER et sans l'entrée SPACE

Si l'entrée est :

- `i_abyIput` [255]
 - Entrée [1] = 65
 - Entrée [2] = 66
 - Entrée [3] = 67
 - Entrée [4] = 68
 - Entrée [5] = 69
- `i_xOrdr` : TRUE
- `i_xSpce` : FALSE

La fonction `ArrayOfByte_TO_String` affiche "ABCDE".

Avec l'entrée ORDER et l'entrée SPACE

Si l'entrée est :

- `i_abyIput` [255]
 - Entrée [1] = 65
 - Entrée [2] = 66
 - Entrée [3] = 67
 - Entrée [4] = 68
 - Entrée [5] = 69
- `i_xOrdr` : FALSE
- `i_xSpce` : TRUE

La fonction `ArrayOfByte_TO_String` affiche "BADC E".

DT_AS_WORD: Conversion de Date et Heure sous forme d'un mot

Contenu de ce chapitre

Bloc fonction DT_AS_WORD 163

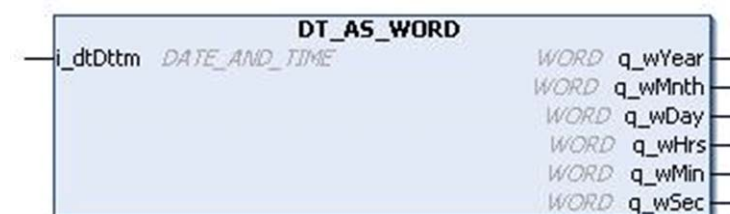
Vue d'ensemble

Ce chapitre décrit le bloc fonction DT_AS_WORD.

Bloc fonction DT_AS_WORD

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction DT_AS_WORD :



Description fonctionnelle

Le bloc fonction DT_AS_WORD extrait les données d'une date et les convertit en mots équivalents.

L'entrée DATE_AND_TIME est convertie en sortie sous forme d'un WORD séparant année, mois, date, heure, minute et seconde.

Exemple

Avec l'entrée DT#2008-08-15-11:05:30, les sorties sont :

- Sortie année : 2008
- Sortie mois : 8
- Sortie jour : 15
- Sortie heures : 11
- Sortie minutes : 5
- Sortie secondes : 30

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction DT_AS_WORD :

Entrée	Type de données	Description
i_dtDttm	DT	Entrée date et heure Plage : 1970-01-01-00:00:00... 2106-02-07-06:28:15

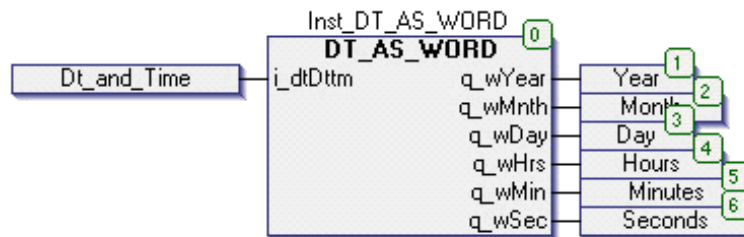
Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction DT_AS_WORD :

Sortie	Type de données	Description
q_wYear	WORD	Sortie année Plage : 1970...2106
q_wMnth	WORD	Sortie mois Plage : 1...12
q_wDay	WORD	Sortie date Plage : 1...31
q_wHrs	WORD	Sortie heures Plage : 1...23
q_wMin	WORD	Sortie minutes Plage : 1...59
q_wSec	WORD	Sortie secondes Plage : 1...59

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction DT_AS_WORD :



L'utilisation du bloc fonction est donnée dans l'exemple ci-dessous :

- i_dtDttm: DT#2008-08-15-11:05:30
- q_wYear: 2008
- q_wMnth: 8
- q_wDay: 15
- q_wHrs: 11
- q_wMin: 5
- q_wSec: 30

DWORD_AS_WORD: Division d'un mot double en deux mots

Contenu de ce chapitre

Bloc fonction `DWORD_AS_WORD` 165

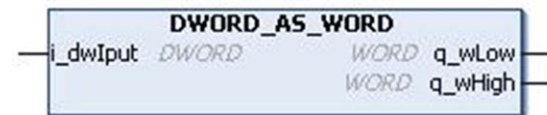
Vue d'ensemble

Ce chapitre décrit le bloc fonction `DWORD_AS_WORD`.

Bloc fonction `DWORD_AS_WORD`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `DWORD_AS_WORD` :



Description fonctionnelle

Le bloc fonction `DWORD_AS_WORD` convertit une valeur d'entrée de type de données `DWORD` en sortie de poids faible et de poids fort de type `WORD`.

Le mot double d'entrée `i_dwInput` est divisé en deux mots, de poids fort `q_wHigh` et de poids faible `q_wLow`.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `DWORD_AS_WORD` :

Entrée	Type de données	Description
<code>i_dwInput</code>	<code>DWORD</code>	Valeur d'entrée Plage : 0...4294967295

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `DWORD_AS_WORD` :

Sortie	Type de données	Description
<code>q_wLow</code>	<code>WORD</code>	Valeur de sortie mot de poids faible Plage : 0...65535
<code>q_wHigh</code>	<code>WORD</code>	Valeur de sortie mot de poids fort Plage : 0...65535

String_TO_ArrayOfByte: Tableau de sortie et valeur ASCII de la chaîne d'entrée

Contenu de ce chapitre

Fonction `String_TO_ArrayOfByte` 166

Vue d'ensemble

Ce chapitre décrit le bloc fonction `String_TO_ArrayOfByte`.

Fonction `String_TO_ArrayOfByte`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches de la fonction `String_TO_ArrayOfByte` :



Description fonctionnelle

La fonction `String_TO_ArrayOfByte` est une sortie Array [255] d'octets représentant la valeur ASCII de la chaîne d'entrée.

Si l'entrée ORDER a la valeur TRUE, l'ordre des valeurs de sortie correspond à l'ordre des caractères de la chaîne en entrée. Il existe donc une correspondance biunivoque 1:1 dans l'ordre des entrées et l'ordre des valeurs ASCII renvoyées en sortie comme expliqué à l'exemple 1.

Si l'entrée ORDER a la valeur FALSE, la sortie est telle que la valeur ASCII du caractère chaîne en input[1] de l'entrée est array[1..255] affichée en position[2] de la sortie. La valeur ASCII du caractère chaîne en input[2] de l'entrée array[1..255] est affichée en position 1 de la sortie. De même la valeur ASCII du caractère chaîne en input[3] de l'entrée array[1..255] est affichée en position 4 de la sortie et la valeur ASCII du caractère chaîne en input[4] de l'entrée array[1..255] est affichée en position 3 de la sortie comme expliqué dans l'exemple 2.

Exemple 1

Si l'entrée ORDER a la valeur TRUE, seul le tableau de sortie est affiché dans l'ordre de l'entrée chaîne comme indiqué ci-dessous :

```
i_sIput='ABCDE'
```

```
i_xOrdr= TRUE
```

Alors la sortie de la conversion de chaîne en tableau d'octets est :

- output [1] = 65
- output [2] = 66
- output [3] = 67
- output [4] = 68

- output [5] = 69
- output [6] = 0

Comme indiqué dans l'exemple ci-dessus, Input [1] = A, son code ASCII correspondant est 65, affiché en position output [1].

De même input [2] = B, son code ASCII correspondant est 66, affiché en position output [2], etc.

Exemple 2

```
i_sInput='ABCDE'
```

```
i_xOrder= FALSE
```

Alors la sortie de la conversion de chaîne en tableau d'octets est :

- output [1] = 66
- output [2] = 65
- output [3] = 68
- output [4] = 67
- output [6] = 0
- output [5] = 69

Comme indiqué dans l'exemple ci-dessus,

Input [1] = A, son code ASCII correspondant est 65, affiché en position output [2].

Input [2] = B, son code ASCII correspondant est 66, affiché en position output [1].

De même Input [3] = C, son code ASCII correspondant est 67, affiché en position output [4].

Input [4] = D, son code ASCII correspondant est 68, affiché en position output [3].

De même Input [5] = E, son code ASCII correspondant est 69, affiché en position output [6].

Input [6] = (espace), son code ASCII correspondant est " " (c'est-à-dire un espace), il est affiché en position output [5].

NOTE: Mais si le nombre d'octets en entrée est 255, l'entrée ORDER a la valeur FALSE. La valeur ASCII reste alors à la même position (voir exemple 3 ci-dessous).

Input:

- i_sInput [1...250]='A'
- i_sInput [251...255]='BCDEF'

Order: FALSE

Sortie

- Output [1...250]='65'
- Output [251...255]='CBEDF'

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction String_TO_ArrayOfByte :

Entrée	Type de données	Description
i_sIput	STRING [1...255]	Valeur chaîne d'entrée (1...255)
i_xOrdr	BOOL	TRUE : Sortie dans l'ordre de l'entrée FALSE : La sortie échange les octets de poids fort et de poids faible.

NOTE: L'utilisateur doit obligatoirement définir la taille de la chaîne input[255], sinon la chaîne est prise comme 80 par défaut.

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction String_TO_ArrayOfByte :

Sortie	Type de données	Description
String_TO_ArrayOfByte	ARRAY [0...255] OF BYTE	Tableau de valeurs ASCII Plage : 0...255

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction String_TO_ArrayOfByte :



Avec l'entrée Order

i_sIput [255]:

- Input [1] = A
- Input [2] = B
- Input [3] = C
- Input [4] = D
- Input [5] = E

i_xOrdr: TRUE

La fonction String_TO_ArrayOfByte affiche "65, 66, 67, 68, 69"

Sans l'entrée Order

i_sIput [255]:

- Input [1] = A
- Input [2] = B
- Input [3] = C
- Input [4] = D
- Input [5] = E

i_xOrdr: FALSE

La fonction `String_TO_ArrayOfByte` affiche "66, 65, 68, 67, 69"

WORD_AS_DWORD: Décalage du mot de poids fort et ajout du mot de poids faible

Contenu de ce chapitre

Bloc fonction WORD_AS_DWORD 170

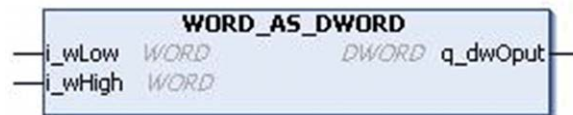
Vue d'ensemble

Ce chapitre décrit le bloc fonction WORD_AS_DWORD.

Bloc fonction WORD_AS_DWORD

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction WORD_AS_DWORD :



Description fonctionnelle

Le bloc fonction WORD_AS_DWORD fusionne deux valeurs d'entrée de type WORD en une seule sortie de type DWORD.

Le mot d'entrée de poids fort *i_wHigh* est décalé vers la gauche de 4 demi-octets et s'ajoute au mot de poids faible d'entrée *i_wLow* pour obtenir une sortie Dword *q_dwOput*.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction WORD_AS_DWORD :

Entrée	Type de données	Description
<i>i_wLow</i>	WORD	Valeur d'entrée du mot de poids faible Plage : 0 à 65 535
<i>i_wHigh</i>	WORD	Valeur d'entrée du mot de poids fort Plage : 0 à 65 535

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction WORD_AS_DWORD :

Sortie	Type de données	Description
<i>q_dwOput</i>	DWORD	Valeur de sortie Dword Plage : 0 à 4 294 967 295

Conversion physique

Contenu de cette partie

Celsius_TO_Fahrenheit: Conversion de Celsius en Fahrenheit.....	172
Celsius_TO_Kelvin: Conversion de Celsius en Kelvin	173
Fahrenheit_TO_Celsius: Conversion de Fahrenheit en Celsius.....	175
Frequency_TO_Period: Calcul de la période temporelle d'une fréquence donnée.....	176
Kelvin_TO_Celsius: Conversion de Kelvin en Celsius	178
Period_TO_Frequency: Calcul de la fréquence du temps donné	180

Vue d'ensemble

Cette partie décrit la famille de Conversion physique.

Celsius_TO_Fahrenheit: Conversion de Celsius en Fahrenheit

Contenu de ce chapitre

Bloc fonction Celsius_TO_Fahrenheit 172

Vue d'ensemble

Ce chapitre décrit le bloc fonction Celsius_TO_Fahrenheit.

Bloc fonction Celsius_TO_Fahrenheit

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Celsius_TO_Fahrenheit :



Description fonctionnelle

La fonction Celsius_TO_Fahrenheit convertit une température de Celsius en Fahrenheit.

Utilisez Fahrenheit_TO_Celsius pour la procédure inverse.

Formule : $T_Fahrenheit = [(T_Celsius * 1,8) + 32]$

Description des broches d'entrée

Ce tableau décrit les broches d'entrée de la fonction Celsius_TO_Fahrenheit :

Entrée	Type de données	Description
i_rInput	REAL	Valeur d'entrée en Celsius Plage : ±1,89e38 (les valeurs supérieures donnent un résultat infini en sortie).

Description des broches de sortie

Ce tableau décrit les broches de sortie de la fonction Celsius_TO_Fahrenheit :

Sortie	Type de données	Description
Celsius_TO_Fahrenheit	REAL	Valeur de sortie en Fahrenheit

Celsius_TO_Kelvin: Conversion de Celsius en Kelvin

Contenu de ce chapitre

Bloc fonction Celsius_TO_Kelvin 173

Vue d'ensemble

Ce chapitre décrit le bloc fonction Celsius_TO_Kelvin.

Bloc fonction Celsius_TO_Kelvin

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Celsius_TO_Kelvin :



Description fonctionnelle

Le bloc fonction Celsius_TO_Kelvin convertit la valeur en unité Celsius de type REAL en unité Kelvin. Le résultat est un nombre de type REAL.

La broche i_rInput sert à l'entrée de la valeur en Celsius.

La broche q_rOput renvoie la valeur équivalente en Kelvin sous forme d'un type de données REAL.

Formule : Kelvin = Celsius + 273,15

Erreur d'entrée détectée

La broche q_xErr prend l'état TRUE si une valeur non valable en Celsius est entrée sur la broche i_rInput et la broche q_rOput renvoie 0 car la température en unité Kelvin ne peut pas être inférieure à 0.

Cette broche d'erreur détectée est réinitialisée sur une valeur d'entrée valable.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Celsius_TO_Kelvin :

Entrée	Type de données	Description
i_rInput	REAL	Valeur d'entrée en Celsius Plage : -273,15...3,4e+38

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Celsius_TO_Kelvin :

Sortie	Type de données	Description
q_xErr	BOOL	TRUE : Entrée non valable FALSE : Entrée valable
q_rOput	REAL	Valeur de sortie en Kelvin Plage : 0...3,4e+38

L'entrée `i_rIput` ne peut pas être définie à moins de -273.15 parce que la valeur équivalente en Kelvin est inférieure à 0, ce qui est théoriquement impossible.

Fahrenheit_TO_Celsius: Conversion de Fahrenheit en Celsius

Contenu de ce chapitre

Bloc fonction Fahrenheit_TO_Celsius 175

Vue d'ensemble

Ce chapitre décrit le bloc fonction Fahrenheit_TO_Celsius.

Bloc fonction Fahrenheit_TO_Celsius

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Fahrenheit_TO_Celsius :



Description fonctionnelle

La fonction Fahrenheit_TO_Celsius convertit une température de Fahrenheit en Celsius.

Utilisez Celsius_TO_Fahrenheit pour la procédure inverse.

Formule : $T_Celsius = [(T_Fahrenheit - 32) / 1,8]$

Description des broches d'entrée

Ce tableau décrit les broches d'entrée de la fonction Fahrenheit_TO_Celsius :

Entrée	Type de données	Description
i_rInput	REAL	Valeur d'entrée en Fahrenheit Plage : $\pm 3,4e^{+38}$

Description des broches de sortie

Ce tableau décrit la sortie de la fonction Fahrenheit_TO_Celsius :

Sortie	Type de données	Description
Fahrenheit_TO_Celsius	REAL	Valeur de sortie en Celsius Plage : $\pm 1,89e^{+38}$

Frequency_TO_Period: Calcul de la période temporelle d'une fréquence donnée

Contenu de ce chapitre

Fonction `Frequency_TO_Period` 176

Vue d'ensemble

Ce chapitre décrit la fonction `Frequency_TO_Period`.

Fonction `Frequency_TO_Period`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches de la fonction `Frequency_TO_Period`:



Description fonctionnelle

La fonction `Frequency_TO_Period` convertit une valeur de fréquence (en Hertz) de type `REAL` en temps. Ce résultat est du type `TIME`.

La période de temps est calculée à partir de la fréquence donnée. La fréquence est définie sur la broche `i_rInput` au format de données `REAL`. La valeur de temps équivalente est renvoyée sur la broche `Frequency_TO_Period` au format de données `TIME`.

$$\text{Période} = 1 / \text{Fréquence}$$

NOTE: Si l'entrée n'est pas dans la plage précédente, la sortie est égale à zéro.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Frequency_TO_Period`:

Entrée	Type de données	Description
<code>i_rInput</code>	<code>REAL</code>	Fréquence d'entrée 0,0...1000,0 Hz

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Frequency_TO_Period`:

Sortie	Type de données	Description
Frequency_TO_Period	TIME	Période de temps de l'entrée en fréquence. 0...4294967295 ms

Kelvin_TO_Celsius: Conversion de Kelvin en Celsius

Contenu de ce chapitre

Bloc fonction Kelvin_TO_Celsius 178

Vue d'ensemble

Ce chapitre décrit le bloc fonction Kelvin_TO_Celsius.

Bloc fonction Kelvin_TO_Celsius

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Kelvin_TO_Celsius :



Description fonctionnelle

Le bloc fonction Kelvin_TO_Celsius convertit la valeur en unité Kelvin de type REAL en unité Celsius. Le résultat est un nombre de type REAL.

La broche i_rInput sert à l'entrée de la valeur en Kelvin.

La broche q_rOput renvoie la valeur équivalente en Celsius sous forme d'un type de données REAL.

Formule : Celsius = Kelvin – 273,15

Erreur d'entrée détectée

La broche q_xErr de type BOOL, prend l'état TRUE si une valeur en Kelvin non valable (c'est-à-dire < 0) est entrée par la broche i_rInput et la broche q_rOput renvoie –273.15, parce que la valeur équivalente en Celsius de la valeur minimale en Kelvin est –273.15.

Cette broche d'erreur détectée est réinitialisée sur une valeur d'entrée valable.

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Kelvin_TO_Celsius :

Entrée	Type de données	Description
i_rInput	REAL	Valeur d'entrée en Kelvin Plage : 0...3,4e+38

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Kelvin_TO_Celsius` :

Sortie	Type de données	Description
<code>q_xErr</code>	BOOL	TRUE : Entrée non valable FALSE : Entrée valable.
<code>q_rOput</code>	REAL	Valeur de sortie en Celsius Plage : -273,15...3,4e ⁺³⁸

Limitations

L'entrée `i_rIput` ne peut pas être définie à moins de 0 parce que en théorie la valeur en Kelvin ne peut pas être inférieure à 0.

Period_TO_Frequency: Calcul de la fréquence du temps donné

Contenu de ce chapitre

Fonction `Period_TO_Frequency` 180

Vue d'ensemble

Ce chapitre décrit la fonction `Period_TO_Frequency`.

Fonction `Period_TO_Frequency`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches de la fonction `Period_TO_Frequency` :



Description fonctionnelle

La fonction `Period_TO_Frequency` convertit un temps de type `TIME` en fréquence (Hertz). Ce résultat est un nombre de type `REAL`.

Cette fonction calcule la fréquence d'une période sous forme d'un temps donné. Le temps est défini par la broche `i_rInput` au format de données `TIME`. La valeur de fréquence équivalente est renvoyée sur la broche `Period_TO_Frequency` au format de données `REAL`.

$$\text{Fréquence} = 1 / \text{Période}$$

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Period_TO_Frequency` :

Entrée	Type de données	Description
<code>i_tInput</code>	<code>TIME</code>	Valeur d'entrée de temps Plage : 0...4294967295 ms

NOTE: Si l'entrée n'est pas dans la plage précédente, la sortie est mise à zéro.

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction `Period_TO_Frequency` :

Sortie	Type de données	Description
Period_TO_Frequency	REAL	Fréquence équivalente à l'entrée de temps Plage : 0 à 1000 Hz

Utilitaires

Contenu de cette partie

Hour_Meter : Accumulation d'heures de fonctionnement	183
Operation_Mode : Sélection du mode de fonctionnement	191

Vue d'ensemble

Cette partie décrit la famille de blocs fonction utilitaires.

Hour_Meter : Accumulation d'heures de fonctionnement

Contenu de ce chapitre

Bloc fonction Hour_Meter	183
Description des broches d'entrée	185
Description des broches de sortie	185
Broche d'entrée – sortie	186
Structures utilisées	186
Description des bits du mot de commande	187
Mot d'état	187
Instanciation et exemple d'utilisation	188

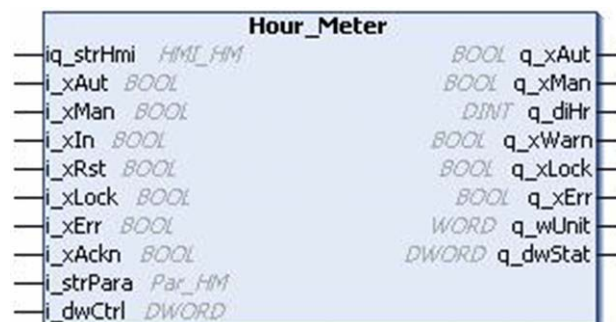
Vue d'ensemble

Ce chapitre décrit le bloc fonction Hour_Meter.

Bloc fonction Hour_Meter

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction Hour_Meter :



Description fonctionnelle

Le bloc fonction Hour_Meter permet de cumuler les heures de fonctionnement de différents équipements.

Limitations

- Si une erreur est détectée, le temps n'est pas calculé tant que l'erreur détectée n'a pas été acquittée. Ensuite le calcul de temps reprend à partir de la valeur au moment où l'erreur a été détectée. Donc, si l'équipement dont le temps de disponibilité est calculé est actif pendant l'état d'erreur détectée, le temps cumulé est inférieur au temps réel.
- Pour une valeur erronée d'unité de temps spécifiée dans la variable `i_strPara.wTypeTime`, le bloc fonction affiche la valeur précédente du temps calculé au moment où l'entrée d'unité de temps était correcte. Mais le bloc fonction n'informe pas l'utilisateur de cette erreur.
- Même si le bloc est verrouillé par une entrée de verrouillage externe, le bloc fonction peut recevoir, générer et afficher une erreur détectée.

- En mode verrouillé, le bloc fonction peut réinitialiser l'erreur détectée en recevant l'entrée d'acquiescement.
- La valeur de temps calculée en sortie n'est qu'en secondes, minutes ou heures. L'utilisateur doit le convertir en un format tel que HH:MM:SS si nécessaire.
- La variable de structure `i_strPara` contient la valeur de temps d'avertissement et l'unité de temps. L'utilisateur doit prendre des précautions spécifiques pour ne pas modifier accidentellement l'unité de temps, car ceci pourrait générer des alarmes intempestives.

Modes de fonctionnement

Le cumul peut s'effectuer en mode manuel ou en mode automatique :

- **Mode automatique** : le mode automatique est sélectionné par la broche d'entrée `i_xAut`. Quand l'entrée `i_xIn` est réglée sur TRUE, le bloc cumule le temps et s'arrête quand `i_xIn` prend la valeur FALSE. Le temps cumulé est disponible sur la broche de sortie `q_diHr`.
- **Mode manuel** : le mode manuel est activé par la broche `i_xMan`. Quand l'entrée `i_xIn` est réglée sur TRUE, le bloc cumule le temps et s'arrête quand `i_xIn` prend la valeur FALSE. Le temps cumulé est disponible sur la broche de sortie `q_diHr`. Le cumul peut être inhibé par le bit de commande `i_dwCtrl`.

Le bloc est désactivé au démarrage du contrôleur et reste dans le mode spécifié jusqu'à la sélection d'un autre mode. Si les deux entrées sont mises à 1, le mode de fonctionnement n'est pas valable.

Réinitialisation de valeur

La réinitialisation de la sortie `q_diHr` est effectuée par un front montant sur l'entrée `i_xRst` en mode automatique ou par un bit de commande en mode manuel.

La valeur de réinitialisation de la sortie `q_diHr` est définie à la valeur `i_strPara.diSp` (point de consigne). De plus, le signal détecté `q_xWarn` est activé quand la sortie `q_diHr` dépasse la limite d'erreur détectée spécifiée par le paramètre `i_strPara.diWaitTime`.

Définition du type de la valeur de sortie

Le paramètre `i_strPara.wTypeTime` définit l'unité de la valeur de sortie. Les choix possibles sont secondes, minutes et heures. La fonction de cumul ne dépend pas de cette valeur, mais elle s'effectue toujours en secondes.

Conditions de fonctionnement

Le comptage ne s'effectue que si l'entrée de verrouillage `i_xLock` a la valeur FALSE. Un signal de verrouillage actif inhibe le fonctionnement du compteur d'heures. Un verrouillage actif est indiqué sur la sortie `q_xLock`.

Le bloc fonction active le signal d'erreur détectée si l'entrée d'erreur détectée `i_xErr` est TRUE (erreur externe détectée) ou si le mode de fonctionnement n'est pas valable (erreurs internes détectées). Les erreurs détectées sont indiquées dans l'IHM. Pour réinitialiser la sortie d'erreur détectée, l'erreur détectée doit être acquiescée par un front montant sur l'entrée `i_xAckn` ou par utilisation du bit 16 du signal `i_dwCtrl`.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction Hour_Meter :

Entrée	Type de données	Description
i_xAut	BOOL	TRUE : Mode auto activé FALSE : Désactivé (réglage usine)
i_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé (réglage usine)
i_xIn	BOOL	TRUE : Le bloc cumule le temps FALSE : Désactivé (réglage usine)
i_xRst	BOOL	TRUE : Ramène la sortie à la valeur prédéfinie entrée à l'entrée i_strPara. FALSE : Désactivé (réglage usine) (optionnel)
i_xLock	BOOL	Entrée de verrouillage pour fonctionnement TRUE : Le verrouillage est actif FALSE : Pas de verrouillage. (Réglage usine) (optionnel)
i_xErr	BOOL	TRUE : L'erreur externe détectée est active. FALSE : Aucune erreur externe détectée (réglage d'usine).
i_xAckn	BOOL	Acquittement par un front montant. L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie q_xErr.
i_strPara	STRUCT Par_HM	Structure contenant les paramètres de ce bloc. Consultez la description de , page 186Par_HM.
i_dwCtrl	DWORD	Bits de commande pour interaction depuis l'IHM. Plage : 0...4294967295 Consultez la description des bits des mots de commande, page 187.

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Hour_Meter :

Sortie	Type de données	Description
q_xAut	BOOL	TRUE : Mode automatique activé FALSE : Désactivé
q_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé
q_diHr	DINT	Temps cumulé de fonctionnement en secondes, minutes ou heures.

Sortie	Type de données	Description
		Plage : -2147483648...2147483647
q_xWarn	BOOL	TRUE : Signal indiquant que le temps à l'heure a dépassé une valeur d'avertissement FALSE : Désactivé.
q_xLock	BOOL	TRUE : Le verrouillage est actif. FALSE : Pas de verrouillage Indique que l'opération est bloquée par un verrouillage (entrée i_xLock)
q_xErr	BOOL	TRUE : Erreur détectée active FALSE : Aucune erreur détectée
q_wUnit	WORD	Indique l'unité du temps de fonctionnement affiché à l'heure de sortie : <ul style="list-style-type: none"> 16#01: Secondes 16#02: Minutes 16#04: Heures
q_dwStat	DWORD	Bits d'état à afficher dans l'IHM Plage : 0...4294967295

Broche d'entrée – sortie

Entrée - sortie de l'IHM

Ce tableau décrit la broche d'entrée – sortie du bloc fonction Hour_Meter :

Entrée - sortie	Type de données	Description
iq_strHmi	STRUCT HMI_HM	Structure d'interface avec l'IHM Consultez les structures utilisées, page 186.

Structures utilisées

Par_HM

Élément de structure	Type	Description
diSp	DINT	Valeur prédéfinie pour la sortie de temps de compteur d'heures utilisée à la réinitialisation
diWaitTime	DINT	Valeur de temps pour l'activation du signal d'erreur détectée
wTypeTime	WORD	Définit l'unité du temps cumulé en sortie en secondes, minutes ou heures : <ul style="list-style-type: none"> 16#01 : Secondes 16#02 : Minutes 16#04 : Heures

HMI_HM

Élément de structure	Type	Description
<i>diVal</i>	DINT	Temps cumulé de fonctionnement en secondes, minutes ou heures.
<i>diSp</i>	DINT	Valeur prédéfinie pour la sortie de temps de compteur d'heures utilisée à la réinitialisation
<i>diWaitTime</i>	DINT	Valeur de temps pour l'activation du signal d'erreur détectée
<i>wTypeTime</i>	WORD	Indication de l'unité du temps de fonctionnement affiché à la sortie Heure

Description des bits du mot de commande

Fonctionnalité

Ce tableau décrit les bits du mot de commande :

Rang du bit	Description
0...3	Inutilisé
4	Réinitialise la sortie <i>q_diHr</i> à la valeur de <i>i_strPara.diSp</i>
5	Comptage de temps de pause
6...15	Inutilisé
16	Un changement de ce bit acquitte l'erreur détectée.
17...31	Inutilisé

Mot d'état

Fonctionnalité

Le tableau décrit les bits de mot d'état :

Rang du bit	Description
0	Le mode auto est actif
1	Le mode manuel est actif
2	Aucun mode n'est sélectionné.
3	Le bloc fonction est verrouillé par l'entrée de verrouillage <i>i_xLock</i>
4	Inutilisé
5	Le comptage de temps est en pause
6	Le temps de fonctionnement en <i>i_strPara.diWaitTime</i> est dépassé.
7...15	Inutilisé
16	L'erreur détectée est réinitialisée
17	L'erreur détectée est présente
18...23	Inutilisé
24	L'erreur interne détectée est présente
25	L'erreur externe détectée est présente
26...31	Inutilisé

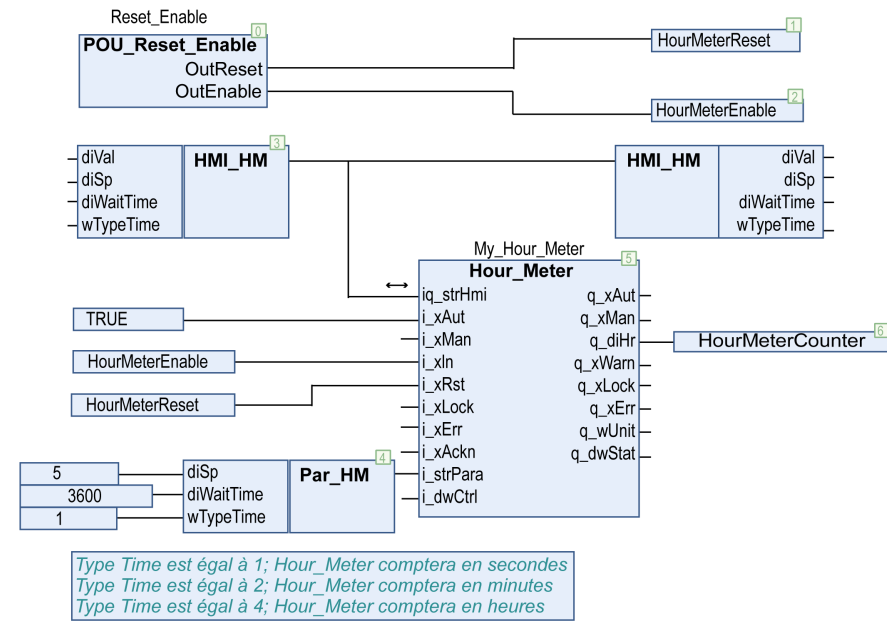
Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Le programme suivant réinitialise et active périodiquement l'entrée d'un bloc fonction Hour_Meter :

```

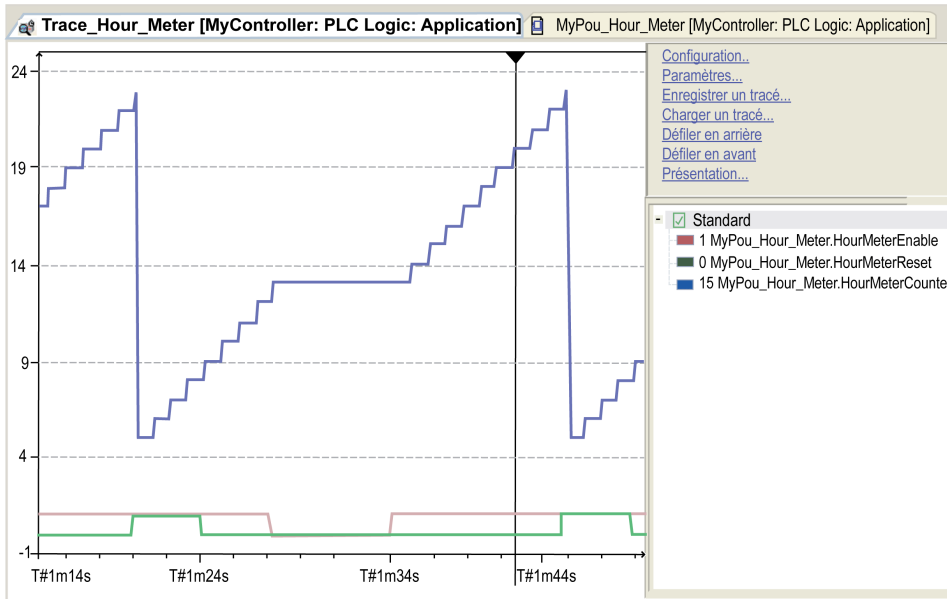
1 PROGRAM MyPou_Hour_Meter
2 VAR
3   FlagStart           : BOOL;
4   Reset_Enable       : POU_Reset_Enable;
5   My_Hour_Meter      : Hour_Meter;
6   HourMeterEnable    : BOOL;
7   HourMeterReset     : BOOL;
8   HourMeterCounter   : DINT;
9 END_VAR
    
```



HourMeterReset et HourMeterEnable sont gérés pour le POU suivant :

```
Controller: PLC Logic: Application] MyPou_Hour_Meter [MyController: PLC Logic:
1  FUNCTION_BLOCK POU_Reset_Enable
2  VAR_INPUT
3  END_VAR
4  VAR_OUTPUT
5      OutReset      : BOOL := FALSE;
6      OutEnable     : BOOL := FALSE;
7  END_VAR
8  VAR
9      MyTimer       : UDINT;
10     CptSeconds    : UDINT;
11     CptSeconds2   : UDINT;
12
13 END_VAR
14
1
2  MyTimer      := SysTimeGetMs();
3  CptSeconds   := ( MyTimer / 1000 );
4  CptSeconds2 := CptSeconds MOD 25;
5
6  CASE CptSeconds2 OF
7
8  0 :
9      OutReset := TRUE;
10     OutEnable := TRUE;
11
12 4 :
13     OutReset := FALSE;
14
15 8 :
16     OutEnable := FALSE;
17
18 15 :
19     OutEnable := TRUE;
20
21 END_CASE
```

Toutes les 25 secondes, les données de HourMeterCounter sont réinitialisées avec une valeur initiale de 5. Quand HourMeterReset a la valeur FALSE, le compteur maintient sa valeur en cours.



Bleu HourMeterCounter

Vert HourMeterReset

Rouge HourMeterEnable

Dans cet exemple, le temps de cycle du POU dans le MAST n'a aucune influence. Pour cet exemple, la périodicité est de 100 millisecondes.

Operation_Mode : Sélection du mode de fonctionnement

Contenu de ce chapitre

Bloc fonction <code>Operation_Mode</code>	191
Description des broches d'entrée	192
Description des broches de sortie	193
Description des bits du mot de commande	193
Mot d'état	194

Vue d'ensemble

Ce chapitre décrit le bloc fonction `Operation_Mode`.

Bloc fonction `Operation_Mode`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Operation_Mode` :



Description fonctionnelle

Le bloc fonction `Operation_Mode` permet de sélectionner entre les modes de fonctionnement auto et manuel depuis deux sources différentes :

- Interrupteurs/programme d'automate
- IHM

Limitations

- Si aucun mode de fonctionnement n'est actuellement sélectionné, le mode précédemment actif est maintenu. Si par exemple le mode auto avec commande locale était activé précédemment, à la réinitialisation l'entrée auto est maintenue jusqu'à la définition d'un autre mode.
- Le mode local a une priorité supérieure à la commande de l'IHM. Un interrupteur en mode de fonctionnement n'est pas pris en compte automatiquement quand le mode local est réinitialisé et que la commande d'IHM est activée précédemment en même temps que le mode local.

Mode local

Le mode local peut être activé avec le mode auto ou manuel. Le mode local s'active à partir de l'entrée `i_xLoc` et interdit l'interaction manuelle depuis l'IHM par l'entrée du mot de commande `i_dwCtrl`.

Quand le mode local est actif, le mode de fonctionnement peut être activé à l'aide des entrées `i_xAut` et `i_xMan`:

- Si `i_xAut` est activé, le mode automatique est activé et indiqué à la sortie `q_xAut`.
- Si `i_xMan` est activé, le mode manuel est activé et indiqué à la sortie `q_xMan`.

Si le mode local n'est pas actif, l'activation des bits de `i_dwCtrl` permet aussi d'activer les modes de fonctionnement depuis l'IHM.

NOTE: Quand `q_xHmiCtrl` est activé, les entrées `i_xAut` et `i_xMan` sont ignorées

Priorité

`i_xLoc` a une priorité supérieure au mot de commande `i_dwCtrl`. Ainsi dès que `i_xLoc` est activé, le mode de fonctionnement est à nouveau activé par les entrées `i_xAut` et `i_xMan`.

Réinitialisation d'une erreur détectée

Le bloc génère un mode de fonctionnement non valable si les deux modes auto et manuel sont sélectionnés (erreur interne détectée) et s'affiche sur la sortie `q_xErr`. Ceci active aussi le signal d'erreur détectée si l'entrée d'erreur détectée `i_xErr` est mise à 1 (erreur externe détectée). Les erreurs détectées sont indiquées dans les bits du mot d'état `q_dwStat`. Pour réinitialiser la sortie d'erreur détectée, l'erreur détectée doit être acquittée par un front montant sur l'entrée **input Ack** ou à l'aide du bit d'acquiescement de l'entrée `i_dwCtrl`.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Operation_Mode` :

Entrée	Type de données	Description
<code>i_xAut</code>	BOOL	TRUE : Mode automatique activé FALSE : Désactivé (réglage usine)
<code>i_xMan</code>	BOOL	TRUE : Mode manuel activé. FALSE : Désactivé. (Réglage usine)
<code>i_xLoc</code>	BOOL	TRUE : Mode local activé FALSE : Désactivé. (Réglage usine)
<code>i_xErr</code>	BOOL	TRUE : Erreur externe détectée active. FALSE : Aucune erreur externe détectée. (Réglage usine)

Entrée	Type de données	Description
i_xAckn	BOOL	Acquittement par un front montant. L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie q_xErr.
i_dwCtrl	DWORD	Bits de commande pour interaction depuis l'IHM. Plage : 0...4294967295 Consultez la description des mots d'état, page 193

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Operation_Mode :

Sortie	Type de données	Description
q_xAut	BOOL	TRUE : Mode automatique activé FALSE : Désactivé
q_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé
q_xLoc	BOOL	TRUE : Mode local activé Le mode de fonctionnement ne peut pas être modifié par l'IHM et les blocs fonction ne peuvent pas être actionnés depuis l'IHM. FALSE : Désactivé
q_xHmiCtrl	BOOL	TRUE : Le mode de fonctionnement est donné par l'IHM. q_xHmiCtrl est remplacé par le mode local. FALSE : Désactivé
q_xErr	BOOL	TRUE : Erreur détectée active. FALSE : Aucune erreur détectée
q_dwStat	DWORD	Bits d'état à afficher dans l'IHM Plage : 0...4294967295 Consultez la description des mots d'état, page 194.

Description des bits du mot de commande

Fonctionnalité

Ce tableau décrit les bits du mot de commande :

Rang du bit	Description
0	Active le mode automatique
1	Active le mode manuel
2, 3	Inutilisé
4	Activé : commande IHM active. Ce n'est qu'après l'activation de ce bit que les sélections par le bit 0 et le bit 1 s'effectuent

Rang du bit	Description
5...15	Inutilisé
16	Le front montant de ce bit vaut acquittement de l'erreur détectée.
17...31	Inutilisé

Ce tableau présente la table de vérité :

i_xAut	i_xMan	i_xLoc	i_dwCtrl			q_xAut	q_xMan	q_xLoc	q_xHmi-Ctrl	q_xErr
			bit0 Aut	bit1 Man	bit4 IHM					
0	0	0	0	0	0	0	0	0	0	0
0	0	1	X	X	X	0	0	1	0	0
1	1	1	X	X	X	0	0	1	0	1
1	0	1	X	X	X	1	0	1	0	0
0	1	1	X	X	X	0	1	1	0	0
X	X	1 → 0	X	X	0	PS	PS	0	0	0
X	X	0	0	0	1	0	0	0	1	0
X	X	0	1	0	1	1	0	0	1	0
X	X	0	0	1	1	0	1	0	1	0
X	X	0	X	X	1 → 0	PS	PS	0	0	0
X	X	0	1	1	1	0	0	0	1	1

PS Etat précédent

Mot d'état

Fonctionnalité

Le tableau décrit les bits de mot d'état :

Rang du bit	Description
0	Le mode auto est actif
1	Le mode manuel est actif
2	Le mode local est actif
3	Inutilisé
4	Le mode de commande IHM est actif
5...15	Inutilisé
16	Indique la réinitialisation de l'erreur détectée
17	L'erreur détectée est présente
18...23	Inutilisé
24	L'erreur interne détectée est présente
25	L'erreur externe détectée est présente
26...31	Inutilisé

Commande de vanne

Contenu de cette partie

Bistable_Valve: Commandes vannes bistables	196
Monostable_Valve: Commande de vannes monostables	202
Proportional_Valve: Commande de vannes proportionnelles	207

Vue d'ensemble

Cette part décrit la bibliothèque de fonctions de commande de vanne.

Bistable_Valve: Commandes vannes bistables

Contenu de ce chapitre

Bloc fonction <code>Bistable_Valve</code>	196
Description des broches d'entrée	198
Description des broches de sortie	199
Structure utilisée	200
Description des bits du mot de commande	200
Mot d'état.....	200
Instanciation et exemple d'utilisation	201

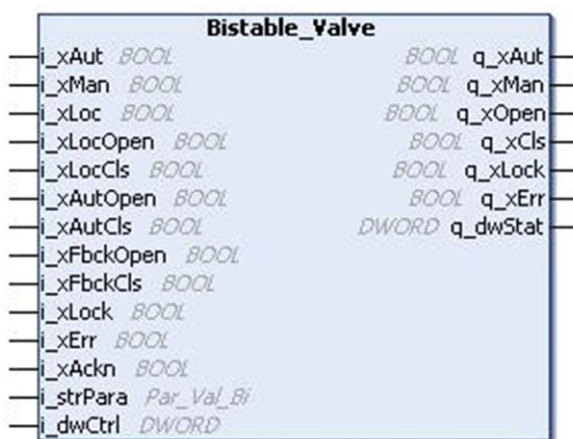
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Bistable_Valve`.

Bloc fonction `Bistable_Valve`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Bistable_Valve` :



Description fonctionnelle

Le bloc fonction `Bistable_Valve` permet de commander la vanne bistable.

Modes de fonctionnement

Le bloc fonction `Bistable_Valve` autorise trois modes de fonctionnement :

- Mode automatique** : Le mode automatique est activé par la broche d'entrée `i_xAut`. Dans ce mode, la vanne est ouverte et fermée par les entrées `i_xAutOpen` et `i_xAutCls` respectivement, que le mode local soit activé ou non.

- **Mode Manuel** : Le mode manuel est activé par la broche `i_xMan`.
Cas 1 : Le mode local n'est pas actif. La vanne est ouverte ou fermée par les commandes de bit du signal `i_dwCtrl`.
Cas 2 : Le mode local est actif. La vanne est ouverte ou fermée par les signaux d'entrée `i_xLocOpen` et `i_xLocCls` respectivement.
- **Mode local** : Le mode local est activé par une broche d'entrée `i_xLoc` et défini en plus du mode automatique ou manuel. Le mode local n'influence pas le mode automatique, mais modifie la source du fonctionnement manuel.

Comportement de sortie

La sortie `q_xOpen` reste active tant que le signal de rétroaction `i_xFbckOpen` reste à l'état bas. La sortie `q_xCls` reste aussi active tant que le signal de rétroaction `i_xFbckCls` reste à l'état bas. Ce comportement de sortie est valable pour les modes manuel et local.

Comportement au démarrage de l'automate

Le bloc est désactivé au démarrage de l'automate et reste dans le même mode de fonctionnement, à moins qu'un nouveau mode soit sélectionné. Si les deux modes automatique et manuel sont sélectionnés simultanément (entrées `i_xAut` et `i_xMan` mises à 1), le mode de fonctionnement n'est pas valable, cet état est indiqué sur la sortie `q_xErr`.

Supervision de la vanne

La position de la vanne est supervisée par les signaux de rétroaction `i_xFbckOpen` et `i_xFbckCls`. Au démarrage du fonctionnement, les entrées de rétroaction doivent signaler la position correcte de la vanne dans un temps défini. Si ce temps est dépassé, le bloc indique une erreur détectée. Ce temps peut être défini par l'élément de structure `iFbckDly` à l'entrée `i_strPara`.

Quand les deux signaux de rétroaction d'ouverture et de fermeture sont manquants (`i_xFbckOpen` et `i_xFbckCls` mis à 0), et si la position de la vanne est inconnue (`QOpen_bi` et `QClose_bi` mis à 0), une erreur de position inconnue est détectée.

Quand l'entrée `xFbckEn` a la valeur `FALSE`, la supervision de temps n'est PAS activée. Consultez la section [Comportement de sortie](#), page 197.

Fonctionnement de la vanne

La vanne ne peut être actionnée que si l'entrée `i_xLock` est mise à 0. Un signal de verrouillage actif inhibe le fonctionnement de la vanne. Un verrouillage actif est indiqué par la sortie `q_xLock`.

La vanne ne peut être actionnée que si la sortie `q_xErr` est mise à 0. Un signal d'erreur détectée actif inhibe le fonctionnement de la vanne.

Gestion d'erreurs détectées

La sortie `q_xErr` est à l'état haut si une erreur est détectée. L'erreur détectée peut être :

- Erreur interne détectée (mode de fonctionnement non valable, signal de rétroaction manquant ou position inconnue).

- Erreur externe détectée

Les erreurs détectées sont indiquées comme alarmes dans l'IHM. Si un verrouillage ou une erreur est détecté pendant le fonctionnement de la vanne, le comportement du bloc fonction dépend de l'élément de structure `i_strPara.xFrceEn` sur l'entrée `i_strPara`. Si cet élément est mis à 1, le bloc active le déplacement de la vanne à sa position par défaut, et la sortie correspondante est à l'état haut (`q_xOpen` ou `q_xCls`) pendant un temps de `i_strPara.iFbckDly` secondes. Sinon le fonctionnement est arrêté et doit être redémarré après la disparition du verrouillage.

Pour réinitialiser `q_xErr`, l'erreur détectée doit être acquittée par un front montant sur l'entrée `i_xAckn` ou par utilisation du bit 16 du signal `i_dwCtrl`.

Définition de la position par défaut

La position par défaut de la vanne peut être définie par `i_strPara.xPosDflt`. Cette description suppose une fermeture par défaut. Si `i_strPara.xPosDflt` est mis à 1, c'est l'ouverture qui est la position par défaut.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Bistable_Valve` :

Entrée	Type de données	Description	Remarques
<code>i_xAut</code>	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	
<code>i_xMan</code>	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	
<code>i_xLoc</code>	BOOL	TRUE : Mode local activé FALSE : Désactivé	
<code>i_xLocOpen</code>	BOOL	Le front montant de 0 à 1 ouvre manuellement la vanne en mode local.	Les modes manuel et local doivent être simultanément mis à 1.
<code>i_xLocCls</code>	BOOL	Le front montant de 0 à 1 ferme manuellement la vanne en mode local.	Les modes manuel et local doivent être simultanément mis à 1.
<code>i_xAutOpen</code>	BOOL	Le front montant de 0 à 1 ouvre la vanne en mode automatique.	
<code>i_xAutCls</code>	BOOL	Le front montant de 0 à 1 ferme la vanne en mode automatique.	
<code>i_xFbckOpen</code>	BOOL	TRUE : Le signal de réaction d'ouverture est actif. FALSE : Pas de réaction d'ouverture	
<code>i_xFbckCls</code>	BOOL	TRUE : Le signal de réaction de fermeture est actif. FALSE : Pas de réaction de fermeture	
<code>i_xLock</code>	BOOL	TRUE : Le verrouillage est actif	

Entrée	Type de données	Description	Remarques
		FALSE : Pas de verrouillage (optionnel)	
i_xErr	BOOL	TRUE : Erreur externe détectée active. FALSE : Aucune erreur externe détectée	
i_xAckn	BOOL	Acquittement par un front montant.	L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie q_xErr.
i_strPara	STRUCT Par_Val_Bi	Structure contenant les paramètres de ce bloc.	Consultez la description des structures utilisées, page 200.
i_dwCtrl	DWORD	Bits de commande pour interaction depuis l'IHM. Plage : 0...4294967295	Consultez la description des mots de commande, page 200.

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Bistable_Valve :

Sortie	Type de données	Description	Remarques
q_xAut	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	-
q_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	-
q_xOpen	BOOL	TRUE : Commande d'ouverture active FALSE : Désactivé	-
q_xCls	BOOL	TRUE : Commande de fermeture active FALSE : Désactivé	-
q_xLock	BOOL	TRUE : Le verrouillage est actif FALSE : Pas de verrouillage	Indique que l'opération est bloquée par un verrouillage (entrée i_xLock)
q_xErr	BOOL	TRUE : Erreur détectée active FALSE : Aucune erreur détectée	-
q_dwStat	DWORD	Bits d'état à afficher dans l'IHM Plage : 0...4294967295	Consultez la description des mots d'état, page 200.

Structure utilisée

Par_Val_Bi

Élément de structure	Type	Description
<i>xFbckEn</i>	BOOL	Active la supervision du signal de rétroaction
<i>iFbckDly</i>	INT	Temporisation en secondes pour obtenir le signal de rétroaction de la vanne.
<i>iRevDly</i>	INT	Temporisation en secondes d'actionnement de la vanne en sens opposé.
<i>xPosDflt</i>	BOOL	Position par défaut de la vanne (FALSE : fermée, TRUE : ouverte).
<i>xFrceEn</i>	BOOL	Active l'application de la position par défaut en cas de verrouillage ou d'erreur détectée.

Description des bits du mot de commande

Fonctionnalité

Ce tableau décrit les bits du mot de commande :

Rang du bit	Description
0...7	Inutilisé
8	Le front montant de 0 à 1 ouvre manuellement la vanne (la sortie <i>q_xOpen</i> a la valeur TRUE) en mode manuel.
9	Le front montant de 0 à 1 ferme manuellement la vanne (la sortie <i>q_xCls</i> a la valeur TRUE) en mode manuel.
10...15	Inutilisé
16	Acquittements d'erreurs internes et externes détectée est indiquée à la sortie <i>q_xErr</i> . Le déclenchement s'effectue sur front montant.
17...31	Inutilisé

Mot d'état

Fonctionnalité

Le tableau décrit le mot d'état :

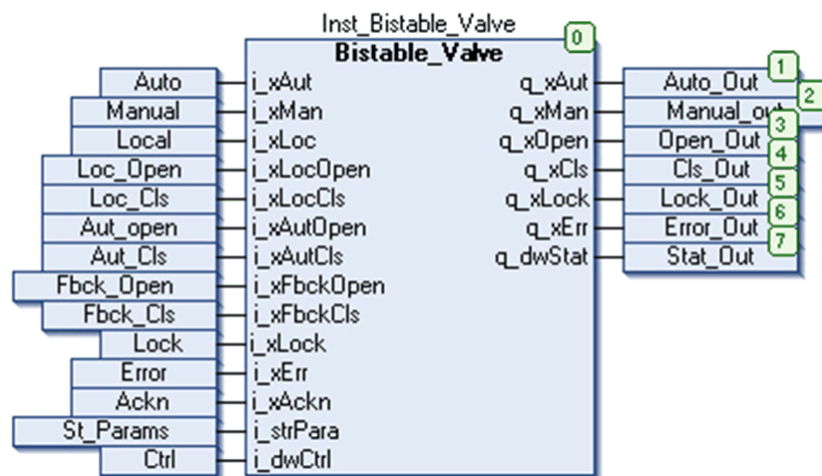
Rang du bit	Description
0	Le mode auto est actif.
1	Le mode manuel est actif.
2	Le mode local est sélectionné.
3	Indique que l'opération est bloquée par un verrouillage (entrée <i>i_xLock</i>).
4...7	Inutilisé
8	Signal pour ouvrir la vanne.
9	Signal pour fermer la vanne.
10...13	Inutilisé

Rang du bit	Description
14	Signal de rétroaction de vanne ouverte.
15	Signal de rétroaction de vanne fermée.
16	Réinitialise l'erreur détectée (q_xErr).
17	Indique que l'opération est bloquée par une erreur détectée interne ou externe (entrée i_xErr), qui n'est pas acquittée.
18	Inutilisé
19	Active la supervision du signal de rétroaction.
20	Position par défaut de la vanne (FALSE : fermée, TRUE : ouverte)
21...23	Inutilisé
24	Erreur détectée mode de fonctionnement non valable.
25	Erreur externe détectée.
26	Erreur détectée rétroaction manquante.
27	Erreur détectée position inconnue.
28...31	Inutilisé

Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction Bistable_Valve :



Limitations

Quand l'entrée d'activation forcée est active (xFrceEn), la vanne est forcée en position par défaut (xPosDflt) seulement pendant iFbckDly secondes. Elle est réinitialisée si un signal de rétroaction approprié est activé, ou si un signal de verrouillage est supprimé, ou si la sortie q_xErr est acquittée.

Assurez-vous que le signal de rétroaction est activé avant d'exécuter le bloc ; sinon une erreur de position inconnue est détectée après une temporisation de iFbckDly secondes.

Monostable_Valve: Commande de vannes monostables

Contenu de ce chapitre

Bloc fonction <code>Monostable_Valve</code>	202
Description des broches d'entrée	204
Description des broches de sortie	205
Structure utilisée	205
Description des bits du mot de commande	206
Mot d'état	206

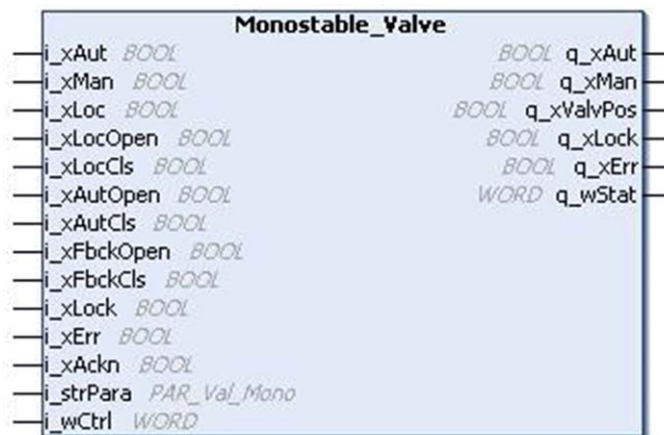
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Monostable_Valve`.

Bloc fonction `Monostable_Valve`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Monostable_Valve` :



Description fonctionnelle

Le bloc fonction `Monostable_Valve` permet de commander la vanne monostable.

Modes de fonctionnement

Le bloc fonction `Monostable_Valve` autorise trois modes de fonctionnement :

- Mode automatique** : Le mode automatique est activé par la broche d'entrée `i_xAut`. Dans ce mode, la vanne est ouverte et fermée par les entrées `i_xAutOpen` (la position par défaut est fermée) et `i_xAutCls` (la position par défaut est ouverte) respectivement, que le mode local soit activé ou non. La sortie `q_xValvPos` reste active tant que les entrées `i_xAutOpen`/`i_xAutCls` restent actives.

- **Mode Manuel** : Le mode manuel est activé par la broche `i_xMan`.
Cas 1 : Le mode local n'est pas actif. La vanne est ouverte ou fermée par les commandes de bit du signal `i_dwCtrl`.
Cas 2 : Le mode local est actif. La vanne est ouverte ou fermée par les signaux d'entrée `i_xLocOpen` et `i_xLocCls` respectivement.
- **Mode local** : Le mode local est activé par une broche d'entrée `i_xLoc` et défini en plus du mode automatique ou manuel. Le mode local n'influence pas le mode automatique, mais modifie la source du fonctionnement manuel.
NOTE: Si les deux modes automatique et manuel sont sélectionnés simultanément (entrées `i_xAut` et `i_xMan` mises à 1), le mode de fonctionnement n'est pas valable, cet état est indiqué sur la sortie `q_xErr`.

Comportement au démarrage de l'automate

Le bloc est désactivé au démarrage de l'automate et reste dans le même mode de fonctionnement, à moins qu'un nouveau mode soit sélectionné.

Supervision de la vanne

La position de la vanne est supervisée par les signaux de rétroaction `i_xFbckOpen` et `i_xFbckCls`. Au démarrage du fonctionnement, les entrées de rétroaction doivent signaler la position correcte de la vanne dans un temps défini. Si ce temps est dépassé, le bloc indique une erreur détectée (erreur détectée rétroaction manquante). Ce temps peut être défini par l'élément de structure `iFbckDly` sur l'entrée `i_strPara`. La supervision peut être désactivée par l'élément de structure `xFbckEn` à l'entrée `i_strPara`.

Fonctionnement de la vanne

La vanne peut être actionnée que si l'entrée `i_xLock` est mise à 0. Un signal de verrouillage actif inhibe le fonctionnement de la vanne et est indiqué par la broche de sortie `q_xLock`.

La vanne ne peut être actionnée que si la sortie `q_xErr` est mise à 0. Un signal d'erreur détectée actif inhibe le fonctionnement de la vanne.

Gestion d'erreurs détectées

La sortie `q_xErr` est à l'état haut si une erreur est détectée. L'erreur détectée peut être :

- Erreur interne détectée (mode de fonctionnement non valable, signal de rétroaction manquant ou position inconnue).
- Erreur externe détectée

Les erreurs détectées sont indiquées comme alarmes dans l'IHM. Si un verrouillage ou une erreur est détecté pendant le fonctionnement de la vanne, le comportement du bloc fonction dépend de l'élément de structure `i_xFrceEn` sur l'entrée `i_strPara`. Si cet élément est mis à 1, le bloc active le déplacement de la vanne à sa position par défaut, et la sortie correspondante est à l'état haut (`q_xOpen` ou `q_xCls`) pendant un temps de `i_strPara.iFbckDly` secondes. Sinon le fonctionnement est arrêté et doit être redémarré après la disparition du verrouillage.

Pour réinitialiser `q_xErr`, l'erreur détectée doit être acquittée par un front montant sur l'entrée `i_xAckn` ou par utilisation du bit 16 du signal `i_dwCtrl`.

Définition de la position par défaut

La position par défaut de la vanne peut être définie par `i_strPara.xPosDflt`. Cette description suppose une fermeture par défaut. Si `i_strPara.xPosDflt` est mis à 1, c'est l'ouverture qui est la position par défaut.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Monostable_Valve` :

Entrée	Type de données	Description	Remarques
<code>i_xAut</code>	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	-
<code>i_xMan</code>	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	-
<code>i_xLoc</code>	BOOL	TRUE : Mode local activé FALSE : Désactivé	-
<code>i_xLocOpen</code>	BOOL	Le front montant de 0 à 1 ouvre manuellement la vanne en mode local.	Les modes manuel et local doivent être simultanément mis à 1.
<code>i_xLocCls</code>	BOOL	Le front montant de 0 à 1 ferme manuellement la vanne en mode local.	Les modes manuel et local doivent être simultanément mis à 1.
<code>i_xAutOpen</code>	BOOL	TRUE : La commande d'ouverture de vanne est activée. FALSE : Désactivé	-
<code>i_xAutCls</code>	BOOL	TRUE : La commande de fermeture de vanne est activée. FALSE : Désactivé	-
<code>i_xFbckOpen</code>	BOOL	TRUE : Le signal de réaction d'ouverture est actif. FALSE : Pas de réaction d'ouverture	-
<code>i_xFbckCls</code>	BOOL	TRUE : Le signal de réaction de fermeture est actif. FALSE : Pas de réaction de fermeture	-
<code>i_xLock</code>	BOOL	TRUE : Le verrouillage est actif FALSE : Pas de verrouillage	Entrée de verrouillage pour le fonctionnement de la vanne. Le fonctionnement de la vanne est inhibé, quand l'entrée est mise à 1.
<code>i_xErr</code>	BOOL	TRUE : Erreur externe détectée active. FALSE : Aucune erreur externe détectée	-

Entrée	Type de données	Description	Remarques
i_xAckn	BOOL	Acquittement par un front montant.	L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie q_xErr.
i_strPara	STRUCT PAR_Val_Mono	Structure contenant les paramètres de ce bloc.	Consultez la description des structures utilisées, page 205.
i_wCtrl	WORD	Bits de commande pour interaction depuis l'IHM. Plage : 0...65535	Consultez la description des mots de commande, page 206.

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Monostable_Valve :

Identificateurs	Type de sortie	Description	Remarques
q_xAut	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	-
q_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	-
q_xValvPos	BOOL	TRUE : Commande d'ouverture/fermeture active FALSE : Désactivé	La commande d'ouverture est active si la position par défaut est fermée/la commande de fermeture est active si la position par défaut est ouverte.
q_xLock	BOOL	TRUE : Le verrouillage est actif. FALSE : Pas de verrouillage	Indique que l'opération est bloquée par un verrouillage (entrée i_xLock)
q_xErr	BOOL	TRUE : Erreur détectée active. FALSE : Aucune erreur détectée	-
q_wStat	WORD	Bits d'état à afficher dans l'IHM Plage : 0...65535	Consultez la description des mots d'état, page 206.

Structure utilisée

PAR_Val_Mono

Élément de structure	Type	Description
xFbckEn	BOOL	Active la supervision du signal de rétroaction
iFbckDly	INT	Temporisation en secondes pour obtenir le signal de rétroaction de la vanne
xPosDflt	BOOL	Position par défaut de la vanne (0 : fermée, 1 : ouverte)

Description des bits du mot de commande

Fonctionnalité

Ce tableau décrit les bits du mot de commande :

Rang du bit	Description
0	Le front montant de 0 à 1 ouvre manuellement la vanne en mode manuel.
1	Le front montant de 0 à 1 ferme manuellement la vanne en mode manuel.
2	L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie <code>q_xErr</code> . Acquittement par un front montant
3...15	Inutilisé

Mot d'état

Fonctionnalité

Le tableau décrit le mot d'état :

Rang du bit	Description
0	Le mode auto est actif.
1	Le mode manuel est actif.
2	Le mode local est actif.
3	La vanne est ouverte/fermée en fonction de la position par défaut.
4	Position par défaut de la vanne.
5	Le signal de verrouillage est actif.
6	Signal de rétroaction de vanne ouverte.
7	Signal de rétroaction de vanne fermée.
8	Erreur détectée active.
9	Erreur détectée mode de fonctionnement non valable.
10	Erreur externe détectée.
11	Erreur détectée rétroaction manquante.
12	Active la supervision de rétroaction.
13-15	Inutilisé.

Proportional_Valve: Commande de vannes proportionnelles

Contenu de ce chapitre

Bloc fonction <code>Proportional_Valve</code>	207
Description des broches d'entrée	209
Description des broches de sortie	210
Broche d'entrée/sortie	211
Structures utilisées	211
Description des bits du mot de commande	212
Mot d'état	212
Instanciation et exemple d'utilisation	213

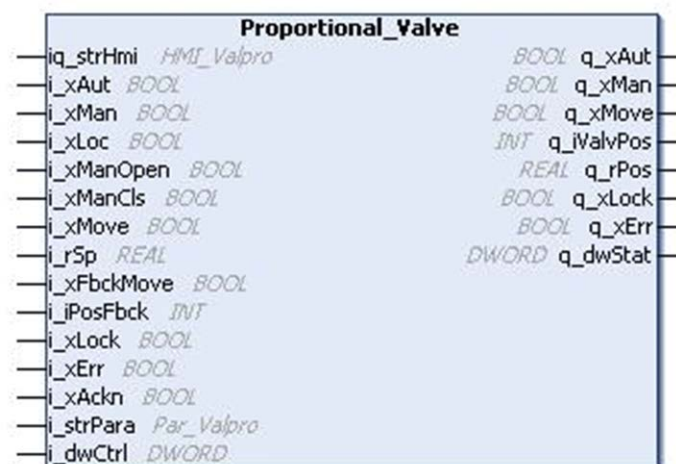
Vue d'ensemble

Ce chapitre décrit le bloc fonction `Proportional_Valve`.

Bloc fonction `Proportional_Valve`

Schéma d'affectation des broches

La figure ci-dessous présente le schéma d'affectation des broches du bloc fonction `Proportional_Valve` :



Description fonctionnelle

Le bloc fonction `Proportional_Valve` s'utilise pour la commande de la vanne proportionnelle.

Modes de fonctionnement

Le bloc fonction `Proportional_Valve` propose trois modes de fonctionnement :

- **Mode automatique** : Le mode automatique est activé par la broche d'entrée `i_xAut`. Dans ce mode, la vanne est ouverte et fermée par l'entrée `i_xMove`, que le mode local soit activé ou non. Le nouveau point de consigné est donné par l'entrée `i_rSp`.

- Mode Manuel** : Le mode manuel est activé par la broche `i_xMan`.
Cas 1 : Le mode local n'est pas actif. La vanne est ouverte et fermée par une commande de bit dans la variable `i_dwCtrl` et la valeur du point de consigne est donnée par `rSP` sur l'entrée-sortie `iq_strHmi`.
Cas 2 : Le mode local est actif. La vanne est actionnée par les entrées `i_xManOpen` et `i_xManCls`.
- Mode local** : Le mode local est activé par une broche d'entrée `i_xLoc` et défini en plus du mode automatique ou manuel. Le mode local n'influence pas le mode automatique, mais modifie la source du fonctionnement manuel. La sortie `q_iValvPos` est définie automatiquement à `i_strPara.rMinSp` en cas d'utilisation de `i_xManCls` ou `i_strPara.rMaxSp` en cas d'utilisation de `i_xManOpen`.
NOTE: Si le mode de fonctionnement passe de manuel ou IHM en mode automatique, tout mouvement de la vanne est arrêté. Toute autre modification du mode de fonctionnement n'influence pas le mouvement de la vanne, mais le point de consigne est ajusté à la valeur du mode de fonctionnement en cours.

Comportement de sortie

La sortie `q_xMove` reste active tant que la nouvelle position donnée par le point de consigne n'est pas atteinte.

Définition d'une plage morte

Il est possible de définir une plage morte par `i_strPara.rBnd`, de façon que `q_xMove` soit désactivé quand l'écart entre la position réelle et le point de consigne est inférieur à la plage morte.

En cas d'utilisation des entrées `i_xManOpen` et `i_xManCls` en mode local, la sortie `q_xMove` est active tant que les entrées sont actives ou si la position maximale ou minimale est atteinte (en prenant en compte la plage morte).

Supervision de la vanne

La position de la vanne est supervisée par les signaux de rétroaction `i_xFbckOpen` et `i_xFbckCls`. Au démarrage du fonctionnement, les entrées de rétroaction doivent signaler la position correcte de la vanne dans un temps défini. Si ce temps est dépassé, le bloc indique une erreur détectée. Ce temps peut être défini par l'élément de structure `iFbckDly` à l'entrée `i_strPara`. La supervision peut être désactivée par l'élément de structure `xFbckEn` à l'entrée `i_strPara`.

Fonctionnement de la vanne

La vanne peut être actionnée si l'entrée `i_xLock` est mise à 0. Un signal de verrouillage actif inhibe le fonctionnement de la vanne. Un verrouillage actif est indiqué par la sortie `q_xLock`.

La vanne ne peut être actionnée que si la sortie `q_xErr` est mise à 0. Un signal d'erreur détectée actif inhibe le fonctionnement de la vanne.

Gestion d'erreurs détectées

La sortie `q_xErr` est à l'état haut si une erreur est détectée. L'erreur détectée peut être :

- Erreur interne détectée (mode de fonctionnement non valable, signal de rétroaction manquant ou position inconnue).
- Erreur externe détectée

Les erreurs détectées sont indiquées comme alarmes dans l'IHM. Si un verrouillage ou une erreur est détecté pendant le fonctionnement de la vanne, le comportement du bloc fonction dépend de l'élément de structure `i_strPara.xFrceEn` sur l'entrée `i_strPara`. Si cet élément est mis à 1, le bloc active le déplacement de la vanne à sa position par défaut, et la sortie est à l'état haut (`q_xMove`) pendant un temps de `i_strPara.iFbckDly` secondes. Sinon le fonctionnement est arrêté et doit être redémarré après la disparition du verrouillage.

Pour réinitialiser `q_xErr`, l'erreur détectée doit être acquittée par un front montant sur l'entrée `i_xAckn` ou par utilisation du bit 16 du signal `i_dwCtrl`.

Définition de la position par défaut

La position par défaut de la vanne peut être définie par `i_strPara.xPosDfltSet`. Cette description suppose une fermeture par défaut. Si `i_strPara.xPosDflt` est mis à 1, c'est l'ouverture qui est la position par défaut.

Description des broches d'entrée

Description des broches d'entrée

Ce tableau décrit les broches d'entrée du bloc fonction `Proportional_Valve` :

Entrée	Type de données	Description	Remarques
<code>i_xAut</code>	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	-
<code>i_xMan</code>	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	-
<code>i_xLoc</code>	BOOL	TRUE : Mode local activé FALSE : Désactivé	-
<code>i_xManOpen</code>	BOOL	TRUE : Ouvre manuellement la vanne (sortie <code>q_xMove</code>) en mode local tant que le signal est activé. FALSE : Désactivé	-
<code>i_xManCls</code>	BOOL	TRUE : Ferme manuellement la vanne (sortie <code>q_xMove</code>) en mode local tant que le signal est activé. FALSE : Désactivé	-
<code>i_xMove</code>	BOOL	TRUE : Déplace la vanne à une nouvelle position en mode automatique FALSE : Désactivé	-
<code>i_rSp</code>	REAL	Nouvelle position pour le déplacement de la vanne en mode automatique. Plage : $1,17e^{-38} \dots 3,4e^{+38}$	-
<code>i_xFbckMove</code>	BOOL	TRUE : Rétroaction pour le déplacement de la vanne FALSE : Désactivé	-
<code>i_iPosFbck</code>	INT	Rétroaction de la position de la vanne.	-

Entrée	Type de données	Description	Remarques
		Plage : 0...31	
i_xLock	BOOL	TRUE : Verrouillage activé. FALSE : Désactivé	Entrée de verrouillage pour le fonctionnement de la vanne. Le fonctionnement de la vanne est inhibé, quand l'entrée est mise à 1.
i_xErr	BOOL	TRUE : Erreur externe détectée active. FALSE : Aucune erreur externe détectée	-
i_xAckn	BOOL	Acquittement par un front montant.	L'entrée pour acquittement d'erreurs internes et externes détectées est indiquée à la sortie q_xErr.
i_strPara	STRUCT Par_Valpro	Structure contenant les paramètres de ce bloc.	Consultez les descriptions des structures utilisées, page 211.
i_dwCtrl	DWORD	Bits de commande pour interaction depuis l'IHM. Plage : 0...4294967295	Consultez les descriptions des mots de commande, page 212.

Description des broches de sortie

Description des broches de sortie

Ce tableau décrit les broches de sortie du bloc fonction Proportional_Valve :

Identificateurs	Type de sortie	Description	Remarques
q_xAut	BOOL	TRUE : Mode automatique activé FALSE : Désactivé	-
q_xMan	BOOL	TRUE : Mode manuel activé FALSE : Désactivé	-
q_xMove	BOOL	TRUE : Commande d'ouverture active FALSE : Désactivé	-
q_iValvPos	INT	Signal indiquant la nouvelle position de la vanne. La valeur est donnée par le point de consigne en mode automatique ou manuel. En cas d'actionnement manuel de la vanne en mode local, la valeur est définie en position ouverture à fond ou fermée à fond. Plage : 0...31	Si l'application de la position par défaut est active, le signal est défini à la position par défaut de la vanne en cas de verrouillage ou d'erreur détectée.
q_rPos	REAL	Position réelle de la vanne. Plage : $\pm 3,4e^{+38}$	-

Identificateurs	Type de sortie	Description	Remarques
q_xLock	BOOL	TRUE : Le verrouillage est actif. FALSE : Pas de verrouillage	Indique que l'opération est bloquée par un verrouillage (entrée i_xLock)
q_xErr	BOOL	TRUE : Erreur détectée active. FALSE : Aucune erreur détectée	-
q_dwStat	DWORD	Bits d'état à afficher dans l'IHM Plage : 0...4294967295	Consultez la description des mots d'état, page 212.

Broche d'entrée/sortie

Description

Ce tableau décrit l'entrée/sortie `iq_strHmi` :

Identificateur	Type	Description
iq_strHmi	STRUCT HMI_Valpro	Structure d'interface avec l'IHM Consultez les structures utilisées, page 211.

Structures utilisées

Par_Valpro

Élément de structure	Type	Description
<i>xFbEn</i>	BOOL	Active la supervision du signal de rétroaction
<i>iFbTime</i>	INT	Temporisation en secondes pour obtenir le signal de rétroaction et que la vanne atteigne le point de consigne
<i>xPosDflt</i>	BOOL	Position par défaut de la vanne (0 : fermée, 1 : ouverte)
<i>xPosDfltSet</i>	BOOL	Active l'application de la position par défaut en cas de verrouillage ou d'erreur détectée
<i>rMinSp</i>	REAL	Valeur minimale de point de consigne : position fermée
<i>rMaxSp</i>	REAL	Valeur maximale de point de consigne : position ouverte
<i>rCnvrFact</i>	REAL	Facteur de conversion pour la sortie de position et l'entrée de position de rétroaction
<i>rBnd</i>	REAL	Ecart autorisé entre le point de consigne et la position

HMI_Valpro

Élément de structure	Type	Description
<i>rVal</i>	REAL	Position réelle de la vanne
<i>rSp</i>	REAL	Point de consigne pour le mode manuel
<i>rHighLim</i>	REAL	Limite basse pour la génération d'une erreur détectée et d'une alarme
<i>rLowLim</i>	REAL	Limite haute pour la génération d'une erreur détectée et d'une alarme

Description des bits du mot de commande

Fonctionnalité

Ce tableau décrit les bits du mot de commande :

Rang du bit	Description
0...9	Inutilisé
10	Déplace la vanne vers le point de consigne. Le bit est déclenché par le front
11...15	Inutilisé
16	Le front montant de ce bit vaut acquittement de l'erreur détectée.
17...31	Inutilisé

NOTE: A réception du déclencheur de front (1->0 ou 0->1) sur ce bit (bit 10 du mot de commande), la commande de déplacement de la vanne est active et mise à TRUE tant que la vanne n'atteint pas le point de consigne défini. La commande de déplacement devient inactive et réinitialisée à FALSE quand la vanne atteint la valeur de point de consigne surveillée par l'entrée `i_iPosFbck` OU `siq_xErr` a la valeur TRUE.

Mot d'état

Fonctionnalité

Le tableau décrit le mot d'état :

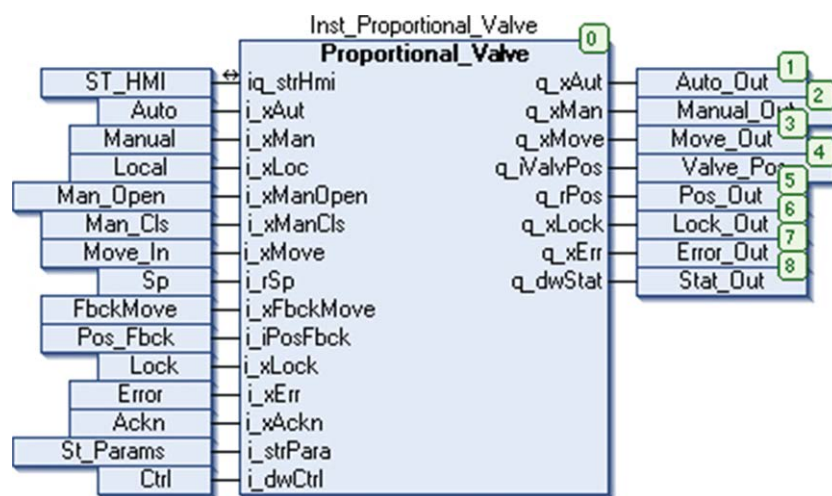
Rang du bit	Description
0	Le mode auto est actif.
1	Le mode manuel est actif.
2	Le mode local est actif.
3	Le bloc fonction est verrouillé par l'entrée de verrouillage <code>i_xLock</code> .
4...7	Inutilisé
8	Point de consigne maximum atteint par la vanne (lu à l'entrée <code>i_iPosFbck</code>).
9	Point de consigne minimum atteint par la vanne (lu à l'entrée <code>i_iPosFbck</code>).
10	Indique un déplacement de la vanne.
11...13	Inutilisé
14	Ce bit est mis à 1, si <code>i_strPara.xFbEn</code> a la valeur TRUE et qu'il y a une réaction de déplacement de vanne ou si <code>i_strPara.xFbEn</code> a la valeur FALSE et qu'il y a une commande de déplacement de vanne
15	Inutilisé
16	Indique la réinitialisation de l'erreur détectée.
17	L'erreur détectée est présente.
18	Inutilisé
19	Équivalent à <code>i_strPara.xFbEn</code> (activation de réaction).
20	Équivalent à <code>i_strPara.xPosDflt</code> (position par défaut).
21...23	Inutilisé

Rang du bit	Description
24	Mode de fonctionnement non valable.
25	L'erreur externe détectée est présente.
26	Rétroaction manquante.
27	La vanne a atteint la limite basse spécifiée par <code>i_strPara.rMinSp</code> .
28	La vanne a atteint la limite haute spécifiée par <code>i_strPara.rMaxSp</code> .
29...31	Inutilisé

Instanciation et exemple d'utilisation

Instanciation et exemple d'utilisation

Cette figure présente une instance du bloc fonction `Proportional_Valve` :



Limitations

Si la rétroaction est activée en mettant `i_strPara.xFbEn = 1`, la rétroaction de déplacement doit non seulement parvenir à l'entrée `i_xFbckMove` mais la vanne doit aussi atteindre le point de consigne dans le temps spécifié `i_strPara.iFbTime`.

Si `i_strPara.rMinSp = i_strPara.rMaxSp = i_rSP` et que le bloc est mis en mode auto, les deux bits du mot d'état `q_dwStat` d'ouverture et de fermeture de la vanne sont mis à 1.

En fonctionnement avec rétroaction activée, même si aucune rétroaction de déplacement n'est reçue par le bloc mais que la vanne atteint le point de consigne dans le temps spécifié, aucune erreur n'est détectée.

Limitations de gestion de priorité

En fonctionnement en mode manuel avec commande locale, les entrées `i_xManOpen` et `i_xManCls` activées ensemble ne produisent aucun résultat jusqu'au retrait de l'une d'entre elles. Mais dans ce cas, la valeur de sortie `q_iValvPos` est imprévisible tant que les deux entrées sont à l'état haut.

Glossaire

A

ARRAY:

Agencement systématique d'objets de données d'un même type sous la forme d'un tableau défini dans la mémoire d'un Logic Controller. La syntaxe est la suivante : `ARRAY [<dimension>] OF <Type>`

Exemple 1 : `ARRAY [1..2] OF BOOL` est un tableau à 1 dimension comportant 2 éléments de type `BOOL`.

Exemple 2 : `ARRAY [1..10, 1..20] OF INT` est un tableau à 2 dimensions comportant 10 x 20 éléments de type `INT`.

ARW:

Acronyme de *anti-reset windup* (saturation antiréinitialisation). Fonction qui arrête l'action de réinitialisation lorsque la mesure est en dehors de la bande proportionnelle afin d'éviter la surcharge du circuit de réinitialisation. L'action de réinitialisation recommence lorsque la mesure revient dans la bande proportionnelle. La saturation antiréinitialisation est une fonction standard dans la plupart des contrôleurs PID de qualité.

ASCII:

Acronyme de *American Standard Code for Information Interchange*. Protocole utilisé pour représenter les caractères alphanumériques (lettres, chiffres, ainsi que certains caractères graphiques et de contrôle).

B

boucle fermée:

Un contrôle en boucle fermée est un système de contrôle de mouvement qui utilise à la fois les informations de retour de positionnement et de vitesse pour générer un signal de correction. Pour cela, il compare sa position et sa vitesse aux valeurs de paramètres spécifiés. Les équipements qui fournissent les informations de retour sont généralement des encodeurs, des résolveurs, des LVTD et des tachymètres.

Voir aussi : *boucle ouverte*

boucle ouverte:

Un système de contrôle de mouvement en boucle ouverte n'utilise pas de capteurs externes pour fournir les signaux de correction de position ou de vitesse.

Voir aussi : *boucle fermée*

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.

chaîne:

Variable composée d'une série de caractères ASCII.

consigne:

Dans un contrôleur PID, valeur cible définie par l'utilisateur. Le principal objectif du contrôleur PID est de garantir que la valeur de processus (PV) atteigne la valeur de consigne (SV).

Voir aussi *valeur PV*

D**DWORD:**

Abréviation de *double word*, mot double. Codé au format 32 bits.

E**élément:**

Raccourci pour l'élément d'un ARRAY.

entrée analogique:

Convertit les niveaux de tension ou de courant reçus en valeurs numériques. Vous pouvez stocker et traiter ces valeurs au sein du Logic Controller.

entrée/sortie:

Index d'un ARRAY.

équipement:

Partie d'une machine comprenant des sous-ensembles tels que des transporteurs, des plaques tournantes, etc.

E/S:

Entrée/sortie

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.

fonction:

Unité de programmation possédant 1 entrée et renvoyant 1 résultat immédiat. Contrairement aux blocs fonction (FBs), une fonction est appelée directement par son nom (et non via une instance), elle n'a pas d'état persistant d'un appel au suivant et elle peut être utilisée comme opérande dans d'autres expressions de programmation.

Exemples : opérateurs booléens (AND), calculs, conversion (BYTE_TO_INT).

H**HMI:**

Acronyme de *human machine interface*, interface homme-machine (IHM). Interface opérateur (généralement graphique) permettant le contrôle d'équipements industriels par l'homme.

I**ID:**

(*identificateur/identification*)

M**MAST:**

Tâche de processeur exécutée par le biais de son logiciel de programmation. La tâche MAST comprend deux parties :

- **IN** : les entrées sont copiées dans la section IN avant l'exécution de la tâche MAST.
- **OUT** : les sorties sont copiées dans la section OUT après l'exécution de la tâche MAST.

NOTE:**ms:**

Abréviation de *milliseconde*

O**octet:**

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

P**PID:**

Acronyme de *proportional, integral, derivative*, proportionnel-intégral-dérivé. Mécanisme de retour de boucle de contrôle générique (contrôleur) largement utilisé dans les systèmes de contrôle industriels.

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.

programme:

Composant d'une application constitué de code source compilé qu'il est possible d'installer dans la mémoire d'un Logic Controller.

PWM:

Acronyme de *pulse width modulation*, modulation de largeur d'impulsion. Sortie rapide qui oscille entre OFF et ON au cours d'un cycle de service réglable, ce qui produit une forme d'onde rectangulaire (ou carrée selon le réglage).

Q**quartet:**

Demi-octet (soit 4 bits)

R

réseau de commande:

Réseau incluant des contrôleurs logiques, des systèmes SCADA, des PC, des IHM, des commutateurs, etc.

Deux types de topologies sont pris en charge :

- à plat : tous les modules et équipements du réseau appartiennent au même sous-réseau.
- à 2 niveaux : le réseau est divisé en un réseau d'exploitation et un réseau intercontrôleurs.

Ces deux réseaux peuvent être indépendants physiquement, mais ils sont généralement liés par un équipement de routage.

S

scrutation:

Fonction comprenant les actions suivantes :

- lecture des entrées et insertion des valeurs en mémoire
- exécution du programme d'application instruction par instruction et stockage des résultats en mémoire
- utilisation des résultats pour mettre à jour les sorties

%:

Selon la norme IEC, % est un préfixe qui identifie les adresses mémoire internes des contrôleurs logiques pour stocker la valeur de variables de programme, de constantes, d'E/S, etc.

sortie analogique:

Convertit des valeurs numériques stockées dans le Logic Controller et envoie des niveaux de tension ou de courant proportionnels.

STOP:

Commande ordonnant au contrôleur de cesser d'exécuter un programme d'application.

T

tâche cyclique:

Le temps de scrutation cyclique a une durée fixe (intervalle) spécifiée par l'utilisateur. Si le temps de scrutation réel est plus court que le temps de scrutation cyclique, le contrôleur attend que le temps de scrutation cyclique soit écoulé avant de commencer une nouvelle scrutation.

NOTE:

tâche:

Ensemble de sections et de sous-programmes, exécutés de façon cyclique ou périodique pour la tâche MAST, ou périodique pour la tâche FAST.

Une tâche présente un niveau de priorité et des entrées et sorties du contrôleur lui sont associées. Ces E/S sont actualisées par rapport à la tâche.

Un contrôleur peut comporter plusieurs tâches.

NOTE:

V

variable:

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

Index

A

Analysis.....	133
ArrayOfByte_TO_String.....	159

B

bibliothèque Toolbox	
GPL.....	103
Bistable_Valve.....	196

C

Celsius_TO_Fahrenheit.....	172
Celsius_TO_Kelvin.....	173
Check_Divisor.....	157
configuration système requise.....	20

D

DT_AS_WORD.....	163
DWORD_AS_WORD.....	165

F

Fahrenheit_TO_Celsius.....	175
FB_2points.....	29
FB_3points.....	33
FB_3points_Ext.....	37
FB_Cyclic_Monitoring.....	72
FB_DeadBand.....	76
FB_Limiter.....	79
FB_P.....	42
FB_PI.....	46
FB_PI_PID.....	64
FB_PID.....	52
FB_PWM.....	82
FB_Redundant_Sensor_Monitoring.....	88
FB_Scaling.....	93
FB_Sensor_Monitoring.....	97
Filter_AnalogInput.....	104
Filter_Arithmetic.....	107
Filter_MovingAverage.....	110
Filter_PT1.....	114
Frequency_Multiplier.....	135
Frequency_Output.....	139
Frequency_TO_Period.....	176

G

GPL	
bibliothèque Toolbox.....	103

H

HMI_HM.....	187
HMI_Valpro.....	211
Hour_Meter.....	183

J

JK_FlipFlop.....	121
------------------	-----

JK_FlipFlop_MasterSlave.....	123
------------------------------	-----

K

Kelvin_TO_Celsius.....	178
------------------------	-----

M

Monostable_Valve.....	202
-----------------------	-----

N

Normalizer_With_Limiter.....	144
------------------------------	-----

O

ONE_SEC_PULSE.....	147
Operation_Mode.....	191
organisation des bits pour le type DWORD.....	22

P

Par_HM.....	186
Par_Val_Bi.....	200
PAR_Val_Mono.....	205
Par_Valpro.....	211
Period_TO_Frequency.....	180
PI.....	28
PID.....	28
Proportional_Valve.....	207

Q

Quantizer.....	148
----------------	-----

R

RS_FlipFlop.....	126
------------------	-----

S

SetBitTo.....	24
Signal_Saturation.....	150
Signal_Statistics.....	154
SR_FlipFlop.....	128
st3PointsPara.....	35
st3PtsExtendedPara.....	40
stPid.....	57
stPiPara.....	49
stPIPIDInLoop.....	68
stPIPIDOutLoop.....	68
stPtPara.....	31
stPwmPara.....	87
String_TO_ArrayOfByte.....	166

T

TestBit.....	26
Toggle_FlipFlop.....	130
Toolbox	
Analysis.....	133
ArrayOfByte_TO_String.....	159
Bistable_Valve.....	196
Celsius_TO_Fahrenheit.....	172

Celsius_TO_Kelvin	173
Check_Divisor	157
configuration système requise	20
DT_AS_WORD	163
DWORD_AS_WORD	165
Fahrenheit_TO_Celsius	175
FB_2points	29
FB_3points	33
FB_3points_Ext	37
FB_Cyclic_Monitoring	72
FB_DeadBand	76
FB_Limiter	79
FB_P	42
FB_PI	46
FB_PI_PID	64
FB_PID	52
FB_PWM	82
FB_Redundant_Sensor_Monitoring	88
FB_Scaling	93
FB_Sensor_Monitoring	97
Filter_AnalogInput	104
Filter_Arithmetic	107
Filter_MovingAverage	110
Filter_PT1	114
Frequency_Multiplier	135
Frequency_Output	139
Frequency_TO_Period	176
Hour_Meter	183
JK_FlipFlop	121
JK_FlipFlop_MasterSlave	123
Kelvin_TO_Celsius	178
Monostable_Valve	202
Normalizer_With_Limiter	144
ONE_SEC_PULSE	147
Operation_Mode	191
organisation des bits pour le type DWORD	22
Period_TO_Frequency	180
Proportional_Valve	207
Quantizer	148
RS_FlipFlop	126
SetBitTo	24
Signal_Saturation	150
Signal_Statistics	154
SR_FlipFlop	128
String_TO_ArrayOfByte	166
TestBit	26
Toggle_FlipFlop	130
WORD_AS_DWORD	170

W

WORD_AS_DWORD	170
---------------------	-----

Schneider Electric
35 rue Joseph Monier
92500 Reuil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

Les normes, spécifications et conceptions pouvant changer de temps à autre, veuillez demander la confirmation des informations figurant dans cette publication.

© 2023 Schneider Electric. Tous droits réservés.

EIO000000097.09