

# EcoStruxure Machine Expert

Fonctions de modem

Guide de la bibliothèque Modem

06/2019

---

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

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

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

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

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

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

© 2019 Schneider Electric. Tous droits réservés.

---

# Table des matières

---



|                        |  |           |
|------------------------|--|-----------|
|                        | <b>Consignes de sécurité</b> .....                                   | <b>5</b>  |
|                        | <b>A propos de ce manuel.</b> .....                                  | <b>7</b>  |
| <b>Chapitre 1</b>      | <b>Généralités sur les modems</b> .....                              | <b>9</b>  |
|                        | Fonctions modem sur les automates .....                              | <b>10</b> |
|                        | Paramètres génériques .....  | <b>11</b> |
|                        | OperationErrorCodesExt: Codes d'erreur d'opération (Type ENUM) ..    | <b>13</b> |
| <b>Chapitre 2</b>      | <b>Configuration</b> .....   | <b>15</b> |
|                        | Ajout d'un modem à votre application .....                           | <b>15</b> |
| <b>Chapitre 3</b>      | <b>Description des blocs fonction</b> .....                          | <b>21</b> |
| 3.1                    | Ouverture et fermeture de communications transparentes .....         | <b>22</b> |
|                        | Dial : ouvrir des communications transparentes .....                 | <b>23</b> |
|                        | HangUp : Fermer les communications transparentes .....               | <b>25</b> |
| 3.2                    | Envoi et réception de SMS .....                                      | <b>26</b> |
|                        | SendSMS : envoyer un SMS .....                                       | <b>27</b> |
|                        | ReceiveSMS : recevoir un SMS .....                                   | <b>29</b> |
| 3.3                    | Services de carte SIM du modem GSM .....                             | <b>31</b> |
|                        | ConfigSim .....  | <b>31</b> |
| <b>Annexes</b> .....   |  | <b>35</b> |
| <b>Annexe A</b>        | <b>Représentation des fonctions et blocs fonction</b> .....          | <b>37</b> |
|                        | Différences entre une fonction et un bloc fonction .....             | <b>38</b> |
|                        | Utilisation d'une fonction ou d'un bloc fonction en langage IL ..... | <b>39</b> |
|                        | Utilisation d'une fonction ou d'un bloc fonction en langage ST ..... | <b>43</b> |
| <b>Glossaire</b> ..... |  | <b>47</b> |
| <b>Index</b> .....     |  | <b>49</b> |



# Consignes de sécurité



## Informations importantes

### AVIS

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



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



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

## DANGER

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

## AVERTISSEMENT

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

## ATTENTION

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

## AVIS

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

---

## REMARQUE IMPORTANTE

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

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

## 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 régulations 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.

---

# A propos de ce manuel

---



## Présentation

### Objectif du document

Ce document décrit la bibliothèque des fonctions modem utilisables par les contrôleurs EcoStruxure Machine Expert.

### Champ d'application

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

### Information spécifique au produit

## AVERTISSEMENT

### PERTE DE CONTROLE

- Le concepteur d'un système de commande doit envisager les modes de défaillance possibles des chemins de commande et, pour certaines fonctions de commande critiques, prévoir un moyen d'atteindre un état sécurisé en cas de défaillance d'un chemin, et après cette défaillance. Par exemple, l'arrêt d'urgence, l'arrêt en cas de surcourse, la coupure de courant et le redémarrage sont des fonctions de contrôle cruciales.
- Des canaux de commande séparés ou redondants doivent être prévus pour les fonctions de commande critique.
- Les chemins de commande système peuvent inclure les liaisons de communication. Soyez particulièrement attentif aux implications des retards de transmission imprévus ou des pannes de liaison.
- Respectez toutes les réglementations de prévention des accidents ainsi que les consignes de sécurité locales.<sup>1</sup>
- Chaque implémentation de cet équipement doit être testée individuellement et entièrement pour s'assurer du fonctionnement correct avant la mise en service.

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

---

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

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.
- 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 commande critiques (arrêt d'urgence, conditions de dépassement des valeurs limites, etc.) en fonction d'une analyse de la sécurité, des règles correspondantes et des réglementations.

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

---

# Chapitre 1

## Généralités sur les modems

---

### Introduction

Vous pouvez connecter un modem PSTN (Public Switched Telephone Network) ou GSM (Global System for Mobile Communications) à une ligne série pour communiquer de manière transparente avec un équipement distant selon les protocoles ASCII, Modbus ou Machine Expert. Vous pouvez également utiliser un modem GSM pour envoyer ou recevoir des SMS.

La bibliothèque modem propose un ensemble de blocs fonction permettant d'utiliser ces fonctions.

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

| Sujet  | Page |
|--|------|
| Fonctions modem sur les automates                              | 10   |
| Paramètres génériques  | 11   |
| OperationErrorCodesExt: Codes d'erreur d'opération (Type ENUM) | 13   |

## Fonctions modem sur les automates

### Introduction

Cette section décrit le fonctionnement et la gestion des fonctions de communication modem des automates.

**NOTE** : Les fonctions de communication sont traitées de manière asynchrone par rapport à la tâche applicative qui a appelé la fonction.

### Blocs fonction disponibles

Ce tableau décrit les blocs fonction modem pouvant être utilisés par les automates :

| Fonction                           | Description   |
|------------------------------------|---|
| Dial ( <i>voir page 23</i> )       | Dial établit des communications avec un équipement distant via un modem.  |
| HangUp ( <i>voir page 25</i> )     | HangUp ferme une connexion établie précédemment.  |
| SendSMS ( <i>voir page 27</i> )    | SendSMS envoie des SMS.   |
| ReceiveSMS ( <i>voir page 27</i> ) | ReceiveSMS permet à l'automate de recevoir des SMS.   |
| ConfigSim ( <i>voir page 31</i> )  | Utilisez ConfigSim si votre carte SIM nécessite un code PIN, un numéro de téléphone de centre SMS ou une commande d'initialisation. |

## Paramètres génériques

### Introduction

Cette section décrit le fonctionnement et la gestion des fonctions de communication modem, en prenant comme exemple le bloc fonction `Dial`.

### Représentation graphique

Les paramètres communs à tous les blocs fonction de la bibliothèque modem apparaissent en surbrillance dans le graphique ci-dessous :



### Paramètres communs

Ces paramètres sont partagés par plusieurs blocs fonction de la bibliothèque modem :

| Entrée  | Type | Commentaire  |
|---|------|--|
| <code>xExecute</code>   | BOOL | La fonction est exécutée sur le front montant de cette entrée.<br><b>REMARQUE</b> : lorsque l'entrée <code>xExecute</code> est réglée sur TRUE lors du premier cycle de tâche en mode RUN se produisant après un redémarrage à chaud ou à froid, le front montant n'est pas détecté. |
| <code>xAbort</code>   | BOOL | annule l'opération en cours au front montant   |
| <code>serialLineNb</code>   | BYTE | le nombre de lignes série (1 ou 2)   |
| <code>timeout</code>  | WORD | le timeout de l'échange est un multiple de 100 ms (0 pour l'infini)  |
| <b>REMARQUE</b> : une opération du bloc fonction peut nécessiter plusieurs échanges. Le timeout s'applique à chaque échange entre le contrôleur et le modem. Ainsi, la durée totale d'exécution du bloc fonction peut être supérieure au timeout. |      |  |

| Sortie             | Type | Commentaire  |
|--------------------|------|--|
| <code>xDone</code> | BOOL | La sortie <code>xDone</code> est réglée sur TRUE lorsque la fonction s'est exécutée avec succès. |
| <code>xBusy</code> | BOOL | La sortie <code>xBusy</code> est réglée sur TRUE lorsque la fonction est en cours d'exécution.   |

| Sortie   | Type                   | Commentaire  |
|--|------------------------|--|
| xAborted   | BOOL                   | La sortie xAborted est réglée sur TRUE dès lors que la fonction est annulée via l'entrée xAbort.   |
| xError   | BOOL                   | La sortie xError est réglée sur TRUE en cas d'interruption de la fonction du fait de la détection d'une erreur. Lorsqu'une erreur a été détectée, nCommError et nOperError permettent d'obtenir des informations sur celle-ci.   |
| nCommError   | SEN.Comm_ErrCodes      | La sortie nCommError contient les codes des erreurs de communication. Le type ENUM provient de la bibliothèque PLCCommunication associée. (Pour obtenir plus d'informations, reportez-vous au guide <i>Fonctions de communication : Bibliothèque PLCCommunication.</i> ) |
| nOperError   | OperationErrorCodesExt | La sortie nOperError contient les codes des erreurs d'opération ( <i>voir page 13</i> ).   |
| <p><b>REMARQUE :</b> dès que la sortie xBusy est réinitialisée sur FALSE, une de ces sorties (et une seule) est définie sur TRUE :</p> <ul style="list-style-type: none"> <li>● xDone</li> <li>● xError</li> <li>● xAborted</li> </ul> |                        |  |

Les blocs fonction nécessitent un front montant pour être initiés. Le bloc fonction doit tout d'abord voir l'entrée xExecute comme FALSE pour détecter un front montant ultérieur.

## AVERTISSEMENT

### FONCTIONNEMENT D'EQUIPEMENT NON INTENTIONNEL

Composez toujours le premier appel à un bloc fonction en réglant son entrée xExecute sur FALSE de sorte qu'il détecte un front montant ultérieur.

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

## OperationErrorCodesExt: Codes d'erreur d'opération (Type ENUM)

### Description du type énumération

Le type énumération `OperationErrorCodesExt` contient des codes qui correspondent aux erreurs détectées. La valeur du code d'erreur de communication renvoyée par la sortie `nCommError` affecte la signification du code d'erreur d'opération renvoyé par la sortie `nOperError`.

### nCommError = CommunicationOK

Lorsque le code d'erreur de communication est 00 hex (transaction correcte), le type énumération `OperationErrorCodesExt` peut renvoyer les valeurs suivantes :

| Enumérateur   | Valeur (hex.) | Description  |
|---|---------------|--|
| <code>OperationOK</code>  | 00            | L'échange est valide.  |
| <code>NotProcessed_or_TargetResourceMissing</code>  | 01            | La requête n'a pas été traitée ou la ressource du système cible est absente.   |
| <code>BadResponse</code>  | 02            | La réponse reçue est incorrecte.   |
| <code>ModemConfSLAsciiFailed</code>   | 100           | La configuration de la ligne série en ASCII a échoué. (Voir Remarque 1.)   |
| <code>ModemReconfSLFailed</code>  | 200           | Le repositionnement de la configuration de la ligne série sur la configuration utilisateur a échoué. (Voir Remarque 1.)  |
| <code>ModemBusy</code>  | 300           | Le modem répond "BUSY" à la commande <code>Dial</code> . (Voir Remarque 2.)  |
| <code>ModemNoDialtone</code>  | 400           | Le modem répond "NO DIALTONE" à la commande <code>Dial</code> . (Voir Remarque 2.)   |
| <code>ModemNoCarrier</code>   | 500           | Le signal porteuse du modem a été perdu ou déconnecté. Le modem répond "NO CARRIER" à la commande <code>Dial</code> . (Voir Remarque 2.)                                       |
| <code>ModemBadAnswer</code>   | 600           | La réponse du modem n'est pas valide.  |
| <code>SimConfigurationFailed</code>   | 1000          | La configuration de la carte SIM a échoué. (Par exemple, un code PUK est nécessaire ou <code>initSimString</code> n'est pas valide. <i>(voir page 33)</i> ) (Voir Remarque 3.) |
| <code>SimPinCodeInvalid</code>  | 2000          | Le code PIN n'est pas valide. (Voir Remarque 3.)   |
| <code>SimSmsCenterInvalid</code>  | 4000          | Le numéro de téléphone du centre SMS n'est pas valide. (Voir Remarque 3.)  |
| <b>REMARQUE 1 :</b> Ces codes énumération sont utilisés par les blocs fonction <code>Dial</code> et <code>HangUp</code> . |               |  |
| <b>REMARQUE 2 :</b> Ces codes énumération sont utilisés par le bloc fonction <code>Dial</code> .                          |               |  |
| <b>REMARQUE 3 :</b> Ces codes énumération sont utilisés par le bloc fonction <code>ConfigSim</code> .                     |               |  |

**nCommError = Refused**

Lorsque le code d'erreur de communication est FF hex (message refusé), le type énumération `OperationErrorCodesExt` peut renvoyer les valeurs suivantes :

| Enumérateur  | Valeur (hex.) | Description  |
|--|---------------|--|
| <code>NotProcessed_or_TargetResourceMissing</code> | 01            | La requête n'a pas été traitée ou la ressource du système cible est absente. |
| <code>BadLength</code>                             | 05            | La longueur est incorrecte.  |
| <code>CommChannelErr</code>                        | 06            | La voie de communication est associée à une erreur détectée.                 |
| <code>BadAddr</code>                               | 07            | L'adresse est incorrecte.  |
| <code>SystemResourceMissing</code>                 | 0B            | Une ressource du système fait défaut.  |
| <code>TargetCommInactive</code>                    | 0C            | Une fonction de communication cible n'est pas active.                        |
| <code>TargetMissing</code>                         | 0D            | La cible est absente.  |
| <code>ChannelNotConfigured</code>                  | 0F            | La voie n'est pas configurée.  |

---

# Chapitre 2

## Configuration

---

### Ajout d'un modem à votre application

#### Introduction

Lorsque vous ajoutez un modem à un gestionnaire de ligne série avec EcoStruxure Machine Expert, la bibliothèque modem est automatiquement incluse dans votre programme. Une fois la bibliothèque ajoutée, vous pouvez utiliser des blocs fonction spécifiques au modem dans les POU de votre application.

Le tableau ci-après répertorie les gestionnaires de ligne série et les fonctions compatibles :

| Gestionnaire de ligne série | Communication transparente  | SMS  |
|-----------------------------|---|--|
| ASCII                       | Modem PSTN ou GSM   | Modem GSM SR2MOD03 avec configuration ASCII spécifique |
| Modbus                      | Modem PSTN ou GSM   | non  |
| EcoStruxure Machine Expert  | Utiliser un modem capable d'ignorer les trames de protocole lorsqu'une connexion n'est pas établie. (Utiliser un modem PSTN TDW33.) | non  |

### Ajout d'un gestionnaire à votre ligne série

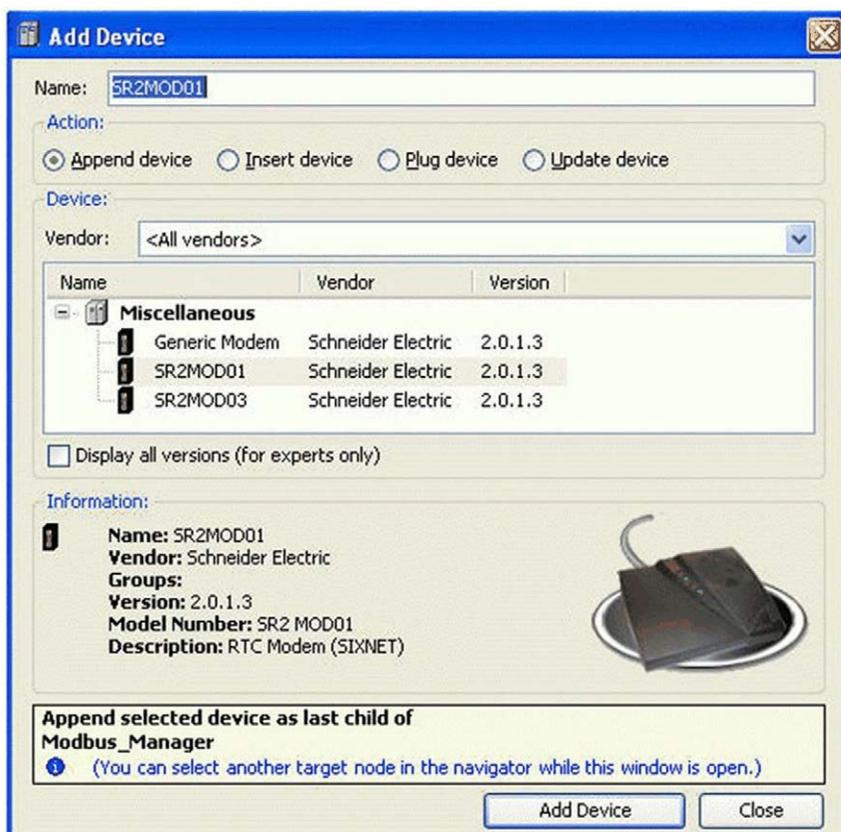
Pour envoyer ou recevoir des SMS via un modem GSM, procédez comme suit :

| Etape | Action   |
|-------|--|
| 1     | Ajoutez le gestionnaire approprié à la ligne série. (Reportez-vous au guide de programmation du contrôleur pour obtenir plus d'informations.)  |
| 2     | Configurez le gestionnaire pour toutes les communications transparentes nécessaires.   |
| 3     | L'utilisation de la fonctionnalité SMS nécessite un gestionnaire ASCII. Il est recommandé de définir une détection de fin de trame (aucun caractère de fin et aucune longueur de trame). (Reportez-vous à la figure ci-après.) |

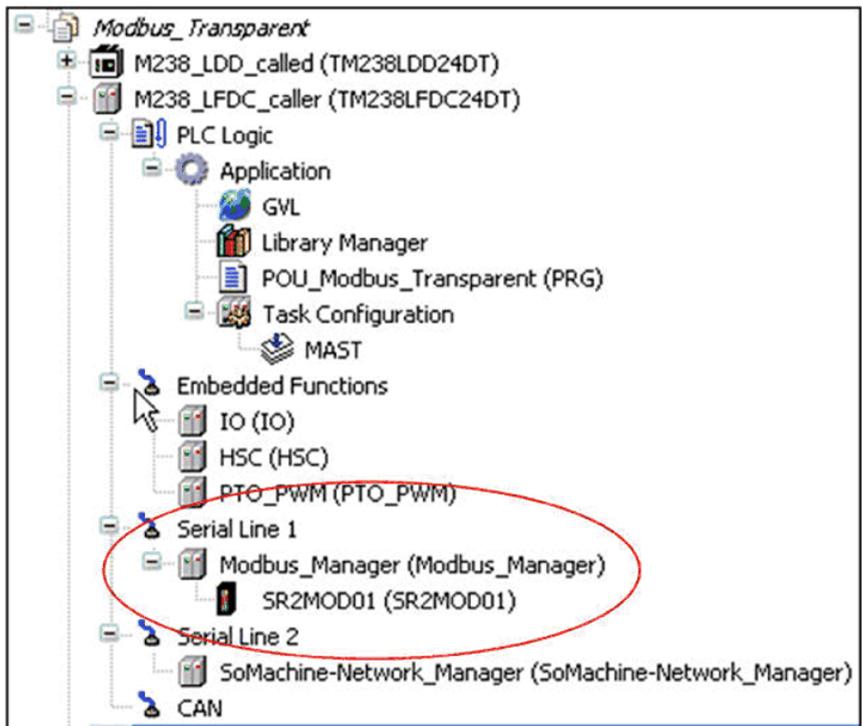
The screenshot shows the 'ASCII\_Manager' configuration window. It has three tabs: 'Configuration', 'Etat', and 'Information'. The 'Configuration' tab is active. Under the 'ASCII' section, there are four input fields: 'Caractère de début' (0), 'Longueur de trame reçue' (0), 'Premier caractère de fin' (0), and 'Timeout de trame reçue (ms)' (250). Below this is the 'Paramètres de ligne série' section, which includes: 'Débit en bauds' (19200), 'Parité' (Aucune), 'Bits de données' (8), 'Bits d'arrêt' (1), and 'Support physique' (RS232).

## Ajout d'un modem au gestionnaire

Ajoutez le modem sélectionné au gestionnaire de ligne série configuré dans la boîte de dialogue **Ajouter un équipement** :

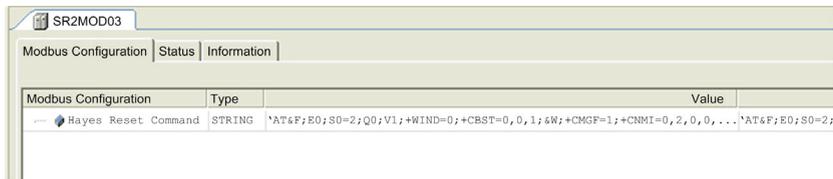


Dans l'arborescence des **équipements**, le modem apparaît en racine du gestionnaire de ligne série :



## Editeur de modem

Double-cliquez sur le modem pour ouvrir l'éditeur d'équipements :



Dans la vue de la configuration, la chaîne `Hayes Reset Command` est définie par défaut.

Pour les modems SR2MOD01 et TDW33 pris en charge par Schneider Electric, cette chaîne de commandes par défaut est à utiliser avec la configuration de ligne série suivante :

**Débit en bauds** 19 200

**Parité** Aucune

**Bits de données** 8

**Bits d'arrêt** 1

Si la configuration de ligne série est différente, la chaîne de commandes doit être modifiée en conséquence.

**NOTE** : La chaîne `Hayes Reset Command` est la chaîne d'initialisation du modem. Elle consiste en une série d'opérations appelées commandes Hayes ("AT"). Elle consiste en une série d'opérations appelées commandes Hayes ("AT"). Cette chaîne est envoyée sur la ligne série lors de la configuration de l'application (c'est-à-dire, après la mise sous tension du contrôleur, le téléchargement de l'application et les commandes de réinitialisation à chaud ou à froid). Si le modem renvoie **OK**, le modem connecté apparaît sans aucune erreur (vert) dans l'arborescence des équipements en mode connecté. Dans le cas contraire, un triangle rouge signale qu'une erreur a été détectée.

**NOTE** : La préparation du modem peut nécessiter quelques secondes.



---

# Chapitre 3

## Description des blocs fonction

---

### Introduction

Ce chapitre décrit les blocs fonction de la bibliothèque modem.

### Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

| Sous-chapitre | Sujet  | Page |
|---------------|--|------|
| 3.1           | Ouverture et fermeture de communications transparentes | 22   |
| 3.2           | Envoi et réception de SMS                              | 26   |
| 3.3           | Services de carte SIM du modem GSM                     | 31   |

## Sous-chapitre 3.1

### Ouverture et fermeture de communications transparentes

---

#### Introduction

Utilisez les blocs fonction `Dial` et `HangUp` pour ouvrir et fermer des communications transparentes entre un contrôleur et un équipement distant. Dans ce cas, un modem est nécessaire à chaque extrémité.

Vous pouvez ajouter trois types de gestionnaires à une ligne série pour échanger des requêtes :

- Gestionnaire Modbus
- Gestionnaire ASCII
- Gestionnaire de réseau Machine Expert

**NOTE** : L'un des exemples de EcoStruxure Machine Expert utilise les blocs fonction `Dial` et `HangUp` pour ouvrir une connexion Modbus entre deux contrôleurs. Le fichier (**Modem\_Modbus.project**) est accessible à partir du menu **Accueil**.

#### Contenu de ce sous-chapitre

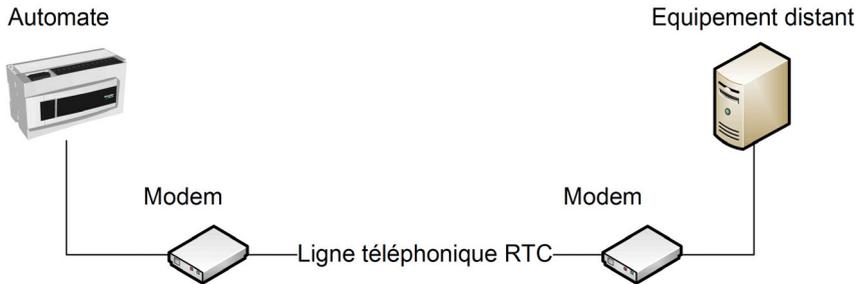
Ce sous-chapitre contient les sujets suivants :

| Sujet  | Page |
|--|------|
| Dial : ouvrir des communications transparentes   | 23   |
| HangUp : Fermer les communications transparentes | 25   |

## Dial : ouvrir des communications transparentes

### Introduction

Avec le bloc fonction `Dial`, le contrôleur peut établir des communications avec un équipement distant via un modem :



**NOTE :** Vous pouvez également établir des communications transparentes avec un modem GSM.

Le bloc fonction `Dial` exécute la commande `Dial` pour établir une communication transparente entre deux modems. Lorsque `xDone` a pour valeur `TRUE`, la communication peut démarrer avec le protocole configuré (Modbus, EcoStruxure Machine Expert ou ASCII).

### Représentation graphique



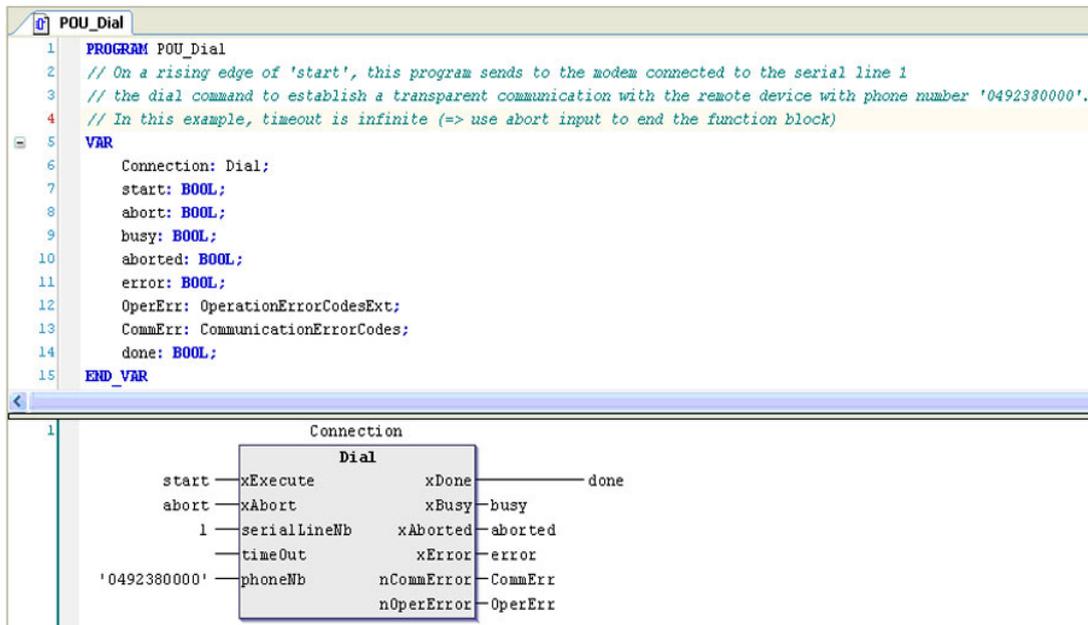
### Description des variables d'E/S

| Entrée  | Type   | Description   |
|---------|--------|---|
| phoneNb | STRING | L'entrée <code>phoneNb</code> contient le numéro de téléphone du modem à appeler. |

**NOTE :** Les paramètres d'entrée et de sortie communs à tous les blocs fonction de la bibliothèque modem sont décrits dans une autre section ([voir page 11](#)).

## Exemple

La figure ci-dessous illustre la déclaration et l'utilisation de la fonction Dial :



## HangUp : Fermer les communications transparentes

### Introduction

La fonction `HangUp` permet à l'automate de fermer une connexion précédemment ouverte.

### Représentation graphique



### Description des variables d'E/S

Les paramètres d'entrée et de sortie utilisés dans le bloc fonction `HangUp` sont ceux communs à tous les blocs fonctions de la bibliothèque modem. Ils sont présentés dans d'autres sections (*voir page 11*).

## Sous-chapitre 3.2

### Envoi et réception de SMS

---

#### Introduction

Les blocs fonction SMS (`SendSMS` et `ReceiveSMS`) permettent d'établir une connexion avec un modem GSM pour envoyer et recevoir des SMS.

**NOTE** : Un exemple de programme SMS utilisant les blocs fonction `SendSMS` et `ReceiveSMS` est inclus dans les exemples de EcoStruxure Machine Expert. Le fichier (**Modem\_SMS.project**) est accessible à partir du menu **Accueil**.

#### Contenu de ce sous-chapitre

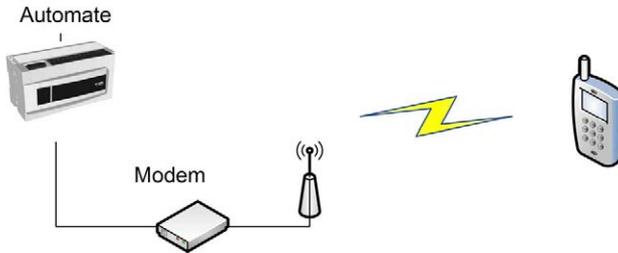
Ce sous-chapitre contient les sujets suivants :

| Sujet                        | Page |
|------------------------------|------|
| SendSMS : envoyer un SMS     | 27   |
| ReceiveSMS : recevoir un SMS | 29   |

## SendSMS : envoyer un SMS

### Introduction

Le bloc fonction SendSMS permet d'établir une connexion avec un modem GSM et d'envoyer un SMS à un récepteur donné. Par exemple, le contrôleur peut envoyer un SMS lorsqu'un déclencheur est actionné pour transmettre une alarme à un téléphone portable spécifié :



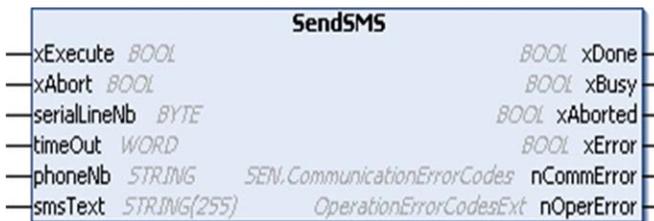
### NOTE :

Vérifiez que votre modem GSM est configuré correctement comme suit :

- Vérifiez que la carte SIM du modem est déverrouillée.
- Vérifiez que le numéro de téléphone du centre SMS est correct.

Vous pouvez utiliser le bloc fonction `ConfigSim` pour définir correctement ces paramètres à partir de votre application.

### Représentation graphique



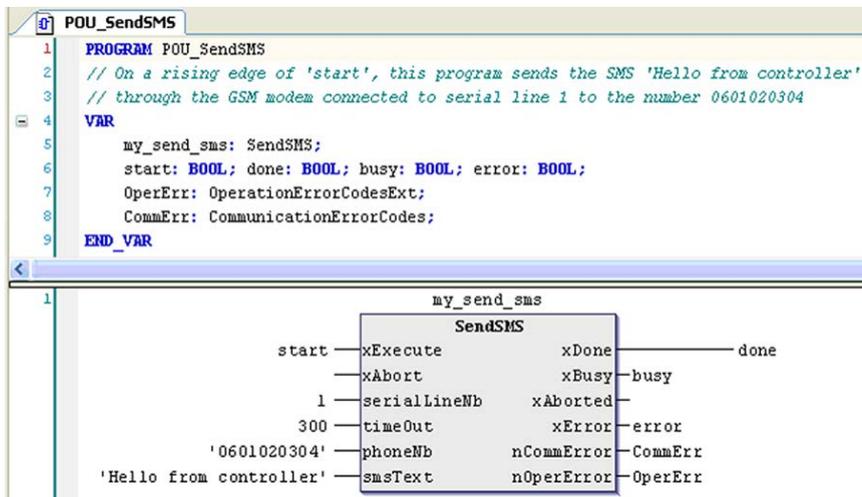
### Description des variables d'E/S

| Entrée               | Type                     | Description  |
|----------------------|--------------------------|--|
| <code>phoneNb</code> | <code>STRING</code>      | L'entrée <code>phoneNb</code> contient le numéro de téléphone du récepteur.      |
| <code>smsText</code> | <code>STRING(255)</code> | L'entrée <code>smsText</code> contient le corps du SMS (255 caractères maximum). |

Les paramètres d'entrée et de sortie communs à tous les blocs fonction de la bibliothèque modem sont décrits dans une autre section ([voir page 11](#)).

## Exemple

La figure ci-dessous illustre la déclaration et l'utilisation de la fonction ReceiveSMS :



## ReceiveSMS : recevoir un SMS

### Introduction

Le bloc fonction `ReceiveSMS` permet de capturer des SMS reçus par un modem GSM. Par exemple, le contrôleur peut traiter une commande envoyée par SMS depuis un téléphone portable donné.

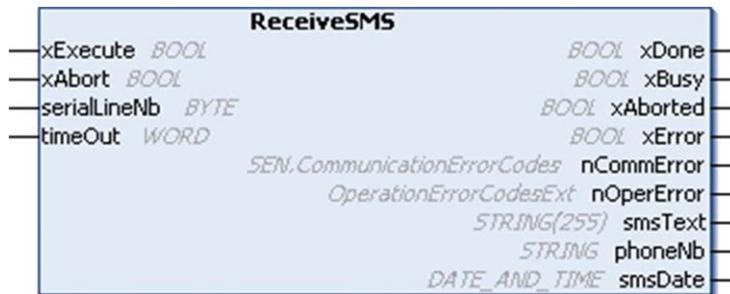
#### NOTE :

Vérifiez que votre modem GSM est configuré correctement comme suit :

- Vérifiez que la carte SIM du modem est déverrouillée.
- Vérifiez que le numéro de téléphone du centre SMS est correct.

Vous pouvez utiliser le bloc fonction `ConfigSim` pour définir correctement ces paramètres dans votre application.

### Représentation graphique



### Description des variables d'E/S

| Sortie               | Type                       | Description   |
|----------------------|----------------------------|---|
| <code>smsText</code> | <code>STRING(255)</code>   | La sortie <code>smsText</code> contient le corps du SMS.                            |
| <code>phoneNb</code> | <code>STRING</code>        | La sortie <code>phoneNb</code> contient le numéro du téléphone qui a envoyé le SMS. |
| <code>smsDate</code> | <code>DATE_AND_TIME</code> | La sortie <code>smsDate</code> contient la date de la communication.                |

Les paramètres d'entrée et de sortie communs à tous les blocs fonction de la bibliothèque modem sont décrits dans une autre section ([voir page 11](#)).

### Exemple

La figure ci-dessous illustre la déclaration et l'utilisation de la fonction ReceiveSMS.

The image shows a screenshot of a PLC programming environment. The top part displays the source code for a program named 'POU\_ReceiveSMS'. The code includes comments and variable declarations for the 'my\_recv\_sms' instance of the 'ReceiveSMS' function block.

```

1 PROGRAM POU_ReceiveSMS
2 // On a rising edge of 'start', this program waits for incoming SMS from the modem connected to the serial line 1
3 // In this example, timeout is infinite (=> use abort input to end the function block)
4
5 VAR
6   my_recv_sms: ReceiveSMS;
7   start: BOOL; abort: BOOL; done: BOOL; busy: BOOL; aborted: BOOL; error: BOOL;
8   messageReceived: STRING(255);
9   phoneOfCaller: STRING;
10  dateSMS: DT;
11  OperErr: OperationErrorCodesExt;
12  CommErr: CommunicationErrorCodes;
13 END_VAR
    
```

The bottom part of the screenshot shows a ladder logic diagram. A variable 'my\_recv\_sms' is used to instantiate the 'ReceiveSMS' function block. The connections between the variable and the function block are as follows:

| Variable Name | Function Block Input/Output | External Variable |
|---------------|-----------------------------|-------------------|
| start         | xExecute                    | done              |
| abort         | xAbort                      | busy              |
| l             | serialLineNb                | aborted           |
| timeOut       | xError                      | error             |
|               | nCommError                  | CommErr           |
|               | nOperError                  | OperErr           |
|               | smsText                     | messageReceived   |
|               | phoneNb                     | phoneOfCaller     |
|               | smsDate                     | dateSMS           |

## Sous-chapitre 3.3

### Services de carte SIM du modem GSM

---

#### ConfigSim

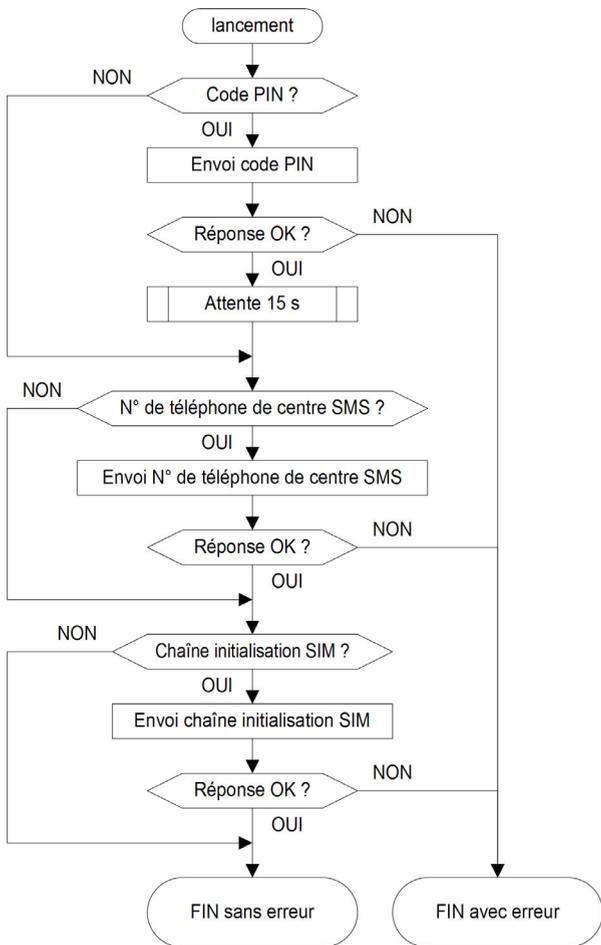
##### Introduction

Avant d'utiliser un autre bloc fonction dans la bibliothèque MODEM, utilisez le bloc fonction `ConfigSim` si la carte SIM du modem GSM nécessite une des opérations suivantes :

- Saisir le code PIN.
- Configurer le numéro de téléphone du centre SMS.
- Envoyer une commande d'initialisation.

Vous pouvez ensuite utiliser directement l'un des blocs fonction SMS dédiés.

Plusieurs commandes sont transmises au modem GSM conformément au diagramme ci-après :



## ⚠ AVERTISSEMENT

### FONCTIONNEMENT ACCIDENTEL DE L'EQUIPEMENT

Si vous utilisez un modem SR2MOD03 avec une carte SIM protégée par code PIN, la chaîne d'initialisation par défaut doit être modifiée dans l'éditeur de configuration du modem. Remplacez la valeur de la commande Hayes Reset par :

```
'AT&F;E0;S0=2;Q0;V1;+WIND=0;+CBST=0,0,1;&W'
```

Utilisez par ailleurs le bloc fonction `ConfigSim` pour transmettre une commande d'initialisation supplémentaire de la manière suivante :

```
entrée InitSimString = 'AT+CMGF=1;+CNMI=0,2,0,0,0;+CSAS'
```

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

### Représentation graphique



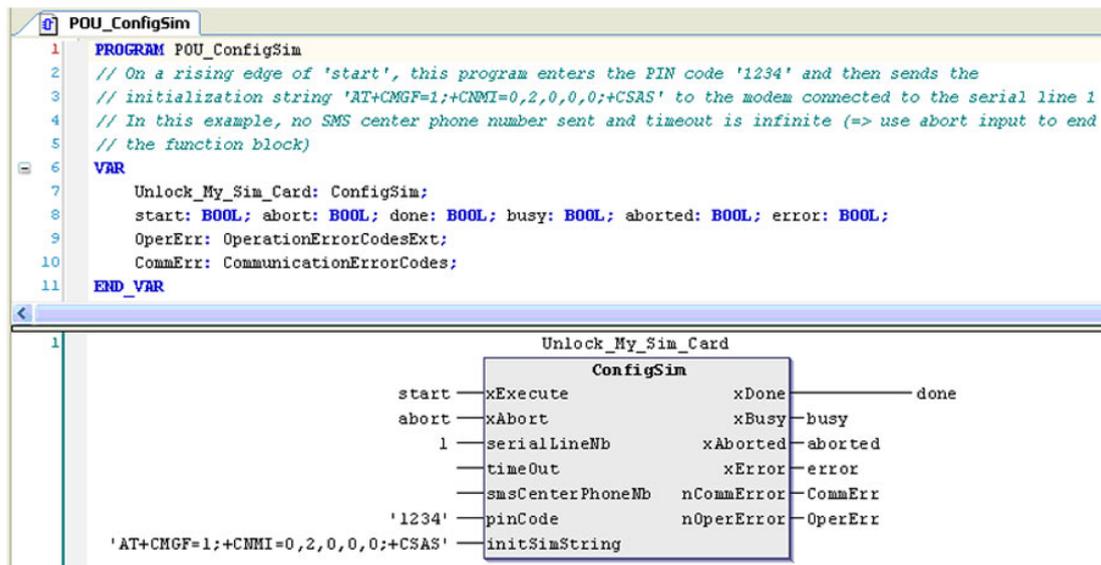
### Description des variables d'E/S

| Entrée                        | Type        | Description   |
|-------------------------------|-------------|---|
| <code>smsCenterPhoneNb</code> | STRING      | L'entrée <code>smsCenterPhoneNb</code> contient le numéro de téléphone du centre SMS à configurer dans la carte SIM. Si elle est vide, ce numéro n'est pas envoyé et le modem utilise le numéro par défaut.   |
| <code>pinCode</code>          | STRING(4)   | L'entrée <code>pinCode</code> indique le code PIN de la carte SIM à envoyer pour la débloquer. Si l'entrée <code>pinCode</code> est vide, aucun code PIN n'est envoyé.  |
| <code>initSimString</code>    | STRING(255) | L'entrée <code>initSimString</code> indique la chaîne d'initialisation de la carte SIM transmise après l'envoi du code PIN et du numéro de téléphone du centre de services.<br><b>REMARQUE</b> : pour le modem SR2MOD03, utilisez la commande suivante :<br>'AT+CMGF=1;+CNMI=0,2,0,0,0;+CSAS' |

Les paramètres d'entrée et de sortie communs à tous les blocs fonction de la bibliothèque modem sont décrits dans une autre section (*voir page 11*).

## Exemple

La figure ci-dessous illustre la déclaration et l'utilisation de la fonction ConfigSim :



---

# Annexes

---





---

# Annexe A

## Représentation des fonctions et blocs fonction

---

### Présentation

Chaque fonction peut être représentée dans les langages suivants :

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

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

### Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

| Sujet  | Page |
|--|------|
| Différences entre une fonction et un bloc fonction             | 38   |
| Utilisation d'une fonction ou d'un bloc fonction en langage IL | 39   |
| Utilisation d'une fonction ou d'un bloc fonction en langage ST | 43   |

## Différences entre une fonction et un bloc fonction

### Fonction

Une fonction :

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

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

### Bloc fonction

Un bloc fonction :

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

**Exemples** : temporisateurs, compteurs

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

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

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

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

### Informations générales

Cette partie explique comment mettre en œuvre une fonction et un bloc fonction en langage IL. Les fonctions `IsFirstMastCycle` et `SetRTCDrift`, ainsi que le bloc fonction `TON`, sont utilisés à titre d'exemple pour illustrer les mises en œuvre.

### Utilisation d'une fonction en langage IL

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

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

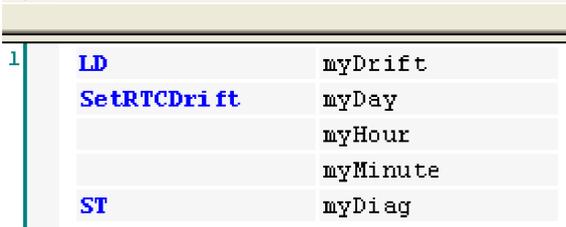
Pour illustrer la procédure, utilisons les fonctions `IsFirstMastCycle` (sans paramètre d'entrée) et `SetRTCDrift` (avec paramètres d'entrée) représentées graphiquement ci-après :

| Fonction   | Représentation graphique |
|--|--------------------------|
| sans paramètre d'entrée :<br><code>IsFirstMastCycle</code> |                          |

| Fonction                                  | Représentation graphique |
|---|--------------------------|
| avec paramètres d'entrée :<br>SetRTCDrift |                          |

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

| Fonction   | Représentation dans l'éditeur IL de POU  |
|--|--|
| Exemple en IL d'une fonction sans paramètre d'entrée :<br>IsFirstMastCycle | <pre> 1  PROGRAM MyProgram_IL 2  VAR 3      FirstCycle: BOOL; 4  END_VAR 5 </pre> <hr/> <pre> 1  IsFirstMastCycle    ST          FirstCycle </pre> |

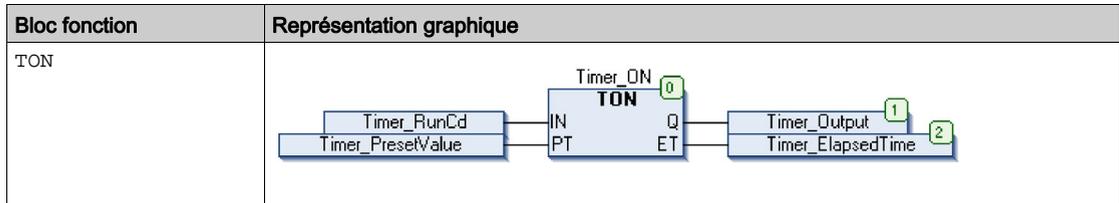
| Fonction   | Représentation dans l'éditeur IL de POU  |
|--|--|
| Exemple IL d'une fonction avec des paramètres d'entrée : SetRTCDrift | <pre> 1  PROGRAM MyProgram_IL 2  VAR 3      myDrift: SINT (-29..29) := 5; 4      myDay: DAY_OF_WEEK := SUNDAY; 5      myHour: HOUR := 12; 6      myMinute: MINUTE; 7      myDiag: RTCSETDRIFT_ERROR; 8  END_VAR 9 </pre>  |

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

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

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

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



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

| Bloc fonction | Représentation dans l'éditeur IL de POU   |
|---------------|---|
| TON           | <pre> 1  PROGRAM MyProgram_IL 2  VAR 3  Timer_ON: TON; // Function Block instance declaration 4  Timer_RunCd: BOOL; 5  Timer_PresetValue: TIME := T#5S; 6  Timer_Output: BOOL; 7  Timer_ElapsedTime: TIME; 8  END_VAR 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 </pre> |

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

### Informations générales

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

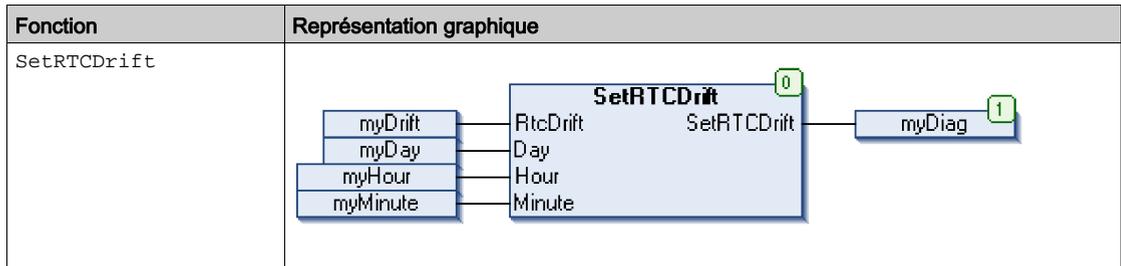
La fonction `SetRTCDrift` et le bloc fonction `TON` sont utilisés à titre d'exemple pour illustrer les mises en œuvre.

### Utilisation d'une fonction en langage ST

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

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

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



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

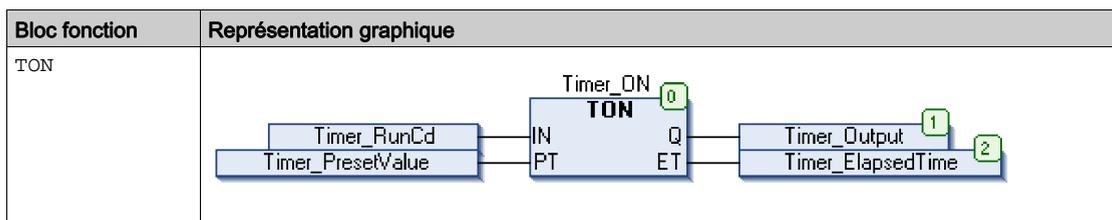
| Fonction    | Représentation dans l'éditeur ST de POU  |
|-------------|--|
| SetRTCDrift | <pre> PROGRAM MyProgram_ST VAR myDrift: SINT(-29..29) := 5; myDay: DAY_OF_WEEK := SUNDAY; myHour: HOUR := 12; myMinute: MINUTE; myRTCAjust: RTCDRIFT_ERROR; END_VAR myRTCAjust:= SetRTCDrift(myDrift, myDay, myHour,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

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

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

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

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



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

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





## B

### bloc fonction

Unité de programmation qui possède 1 ou plusieurs entrées et renvoie 1 ou plusieurs sorties. Les blocs fonction (FBs) sont appelés via une instance (copie du bloc fonction avec un nom et des variables dédiés) et chaque instance a un état persistant (sorties et variables internes) d'un appel au suivant.

Exemples : temporisateurs, compteurs

## C

### CFC

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

## F

### FB

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

## I

### IL

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

### INT

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

## L

### langage en blocs fonctionnels

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

### LD

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

## O

### octet

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

## P

### POU

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

## S

### SMS

Acronyme de *short message service*, service de messagerie texte. Service standard de communication pour les téléphones (et d'autres équipements), qui permet d'envoyer des messages alphanumériques de longueur limitée sur le système de communications mobiles.

### ST

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

## V

### variable

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



## B

bloc fonction  
  ConfigSim, *31*  
  Dial, *23*  
  HangUp, *25*  
  ReceiveSMS, *29*  
  SendSMS, *27*

## C

ConfigSim  
  bloc fonction, *31*

## D

Dial  
  bloc fonction, *23*

## F

fonctions  
  différences entre une fonction et un bloc  
  fonction, *38*  
  utilisation d'une fonction ou d'un bloc  
  fonction en langage IL, *39*  
  utilisation d'une fonction ou d'un bloc  
  fonction en langage ST, *43*

## H

HangUp  
  bloc fonction, *25*

## O

OperationErrorCodesExt  
  Type d'énumération, *13*

## R

ReceiveSMS  
  bloc fonction, *29*

## S

SendSMS  
  bloc fonction, *27*

## T

Type d'énumération  
  OperationErrorCodesExt, *13*