



Chapitre 4

**Traitement des grandeurs
analogiques par un API**

Traitement des grandeurs analogiques par un API

4.1 Introduction

Toutes les grandeurs physiques (tension, température, pression, positionnement, vitesse, etc.) sont des grandeurs analogiques. Afin de pouvoir traiter ces grandeurs dans un API, il est nécessaire de :

- ✦ Convertir la grandeur physique en une grandeur analogique électrique ;
- ✦ Convertir la grandeur analogique électrique en une grandeur numérique.

L'objectif du traitement des grandeurs analogiques par un API est, à partir des principes de base de la technique analogique, des caractéristiques des modules d'entrées et de sorties analogiques.

4.2 Les signaux analogiques et binaires

Les signaux binaires ne peuvent prendre que deux états logiques : l'état logique 1 (tension présente) et l'état logique 0 (aucune tension n'est présente). Dans la technique d'automatisation, des signaux non seulement binaires mais aussi analogiques doivent souvent être lus, traités et transmis. Contrairement aux signaux binaires, les signaux analogiques peuvent prendre toutes les valeurs contenues dans une plage définie.

4.3 Les entrées analogiques

Par opposition aux entrées digitales ou logiques, une entrée analogique permet de recueillir un signal électrique variable pouvant prendre une infinité de valeurs s'étendant sur une plage définie. La valeur d'une entrée analogique change continuellement dans le temps en fonction de la grandeur physique mesurée par le capteur.

- **Les signaux de type tension** : Les signaux de type tension sont générés par certains types de capteurs analogiques. Les plages d'opération les plus courantes sont de 0 à 5 V/DC et de 0 à 10 V/DC.

- **Les signaux de type courant** : Le signal 4÷20 mA est devenu le signal de courant standard dans l'industrie, des fois on peut aussi rencontrer des capteurs de type 0÷20 mA.

- **Les signaux de type résistance** : Les signaux de type résistance sont dans la plupart du temps utilisés avec les capteurs de température comme les thermocouples.

Pour réaliser la conversion de la grandeur physique vers le signal électrique, il suffit d'effectuer une règle de trois :

$$\text{Signal}_{\text{électrique}} = \frac{\text{signal}_{\text{électrique max}} \times \text{grandeur}_{\text{physique}}}{\text{grandeur}_{\text{physique max}}}$$

Exemple N°1 : Capteur de température pouvant mesurer des températures entre 0 et 500°C.

Température	Signal analogique électrique équivalent		
	0..10 VDC	0..20 mA	4..20 mA
0°C	0 VDC	0 mA	4 mA
250°C	5 VDC	10 mA	12 mA
500°C	10 VDC	20 mA	20 mA
300°C	6 VDC	12 mA	13,6 mA

Exemple N°2 : Moteur triphasé dont la vitesse varie de 0 à 4000 tr/min.

Vitesse	Signal analogique électrique équivalent		
	0..10 VDC	0..20 mA	4..20 mA
0 tr/min.	0 VDC	0 mA	4 mA
2000 tr/min	5 VDC	10 mA	12 mA
4000 tr/min	10 VDC	20 mA	20 mA
2500 tr/min	6,25 VDC	12,5 mA	14 mA

4.4 Les sorties analogiques

A la différence des entrées analogiques, les sorties analogiques sont des signaux variables.

Ces signaux sont générés par l'automate (module de sortie analogique) agissent sur des pré-actionneurs analogiques. Par exemple, une sortie analogique peut permettre de contrôler un variateur de vitesse ou encore une électrovanne proportionnelle. Comme pour les entrées analogiques, les types de signaux les plus fréquents sont les des signaux de type tension, courant et résistance.

4.5 Adressage des variables analogiques

Les valeurs analogiques sont lues ou écrites par l'API comme informations de mots. L'accès à ces mots s'effectue par exemple avec les opérandes :

- %IW 64 : mot d'entrée analogique 64 ;
- %QW 64 : mot de sortie analogique 64.

L'adressage des valeurs d'entrée et des valeurs de sortie dépend de l'adressage dans la vue des appareils. La Figure 4.1 montre la configuration matérielle d'un automate programmable Simatic S7-1500 comportant des modules d'entrées/sorties analogiques.

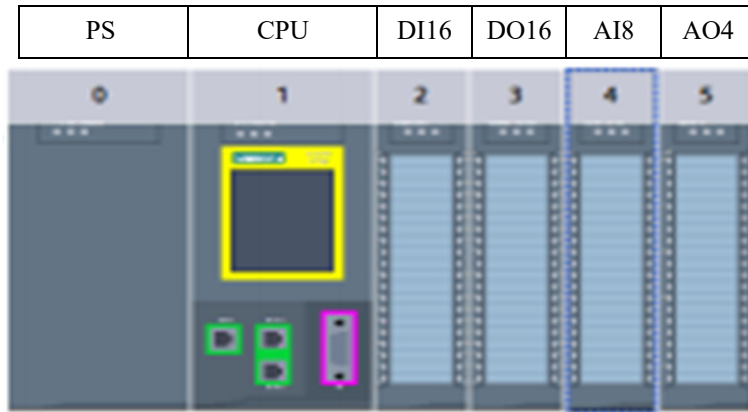


Figure 4.1 - Exemple d'adressage

Cet automate programmable comporte huit (8) entrées analogiques, l'adresse de la première entrée analogique serait dans ce cas %IW64 et celle de la huitième entrée analogique est %IW78. Comme elle comporte quatre (4) sorties analogiques, l'adresse de la première sortie analogique serait dans ce cas est %QW64 et celle de la quatrième sortie analogique est %QW70.

4.6 Normalisation des valeurs analogiques

Si une valeur d'entrée analogique est présente en tant que valeur numérisée dans une plage bipolaire 27648 (voit Tableau 4.1 et Figure 4.2), elle doit habituellement être normalisée pour que les valeurs numériques correspondent à des valeurs physiques du processus. La sortie analogique est également réalisée par la transmission d'une valeur normée qui doit être ensuite mise à l'échelle à la valeur de la sortie bipolaire 27648 comme est montré au Tableau 4.1.

Tableau 4.1 – Traitement des grandeurs analogiques par un API

Les entrées	
Entrée tension	$Reg_{10} = \frac{27648}{10} * V_{A(V)}$
Entrée courant	$Reg_{10} = \frac{27648}{20} * I_{A(mA)}$
Les sorties	
Sortie tension	$V_A = \frac{10}{27648} * Reg_{10}$
Sortie courant	$I_{A(mA)} = \frac{20}{27648} * Reg_{10}$

Pour adapter le signal analogique on utilise la conversion normalisation et mise à l'échelle, deux fonctions prédéfinies par le logiciel Step7.

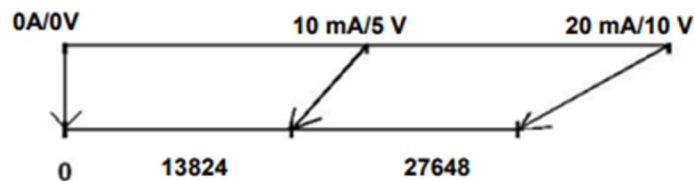


Figure 4.2 – Plage nominale de valeurs analogiques/valeurs numérisées

Le bloc fonctionnel «NORM_X» permet de normaliser la valeur de la variable à l'entrée VALUE en la reproduisant sur une échelle linéaire. On définit, avec les paramètres MIN et MAX, les limites d'une plage de valeurs qui est reflétée sur l'échelle. Le résultat est calculé en fonction de la position de la valeur à normaliser dans cette plage et il est stocké sous forme de nombre réel à virgule flottante à la sortie OUT (voir Figure 4.3).

- Quand la valeur à normaliser est égale à la valeur à l'entrée MIN, la sortie OUT fournit la valeur 0.0.
- Quand la valeur à normaliser est égale à la valeur à l'entrée MAX, la sortie OUT fournit la valeur 1.0.

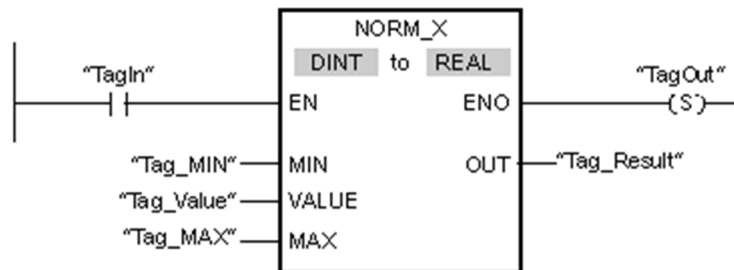


Figure 4.3 – Bloc fonctionnel NORM_X

L'instruction de normalisation utilise l'équation suivante :

$$\text{OUT} = \frac{(\text{VALUE} - \text{MIN})}{(\text{MAX} - \text{MIN})}$$

La sortie de validation ENO fournit l'état logique 0 quand l'une des conditions suivantes est vraie :

- L'entrée de validation EN fournit l'état logique 0 ;
- La valeur à l'entrée MIN est supérieure ou égale à la valeur à l'entrée MAX ;
- La valeur d'un nombre à virgule flottante indiqué se trouve en dehors de la plage des nombres normalisés selon IEEE-754 ;
- La valeur à l'entrée VALUE est le résultat d'une opération de calcul non valide (NaN : not a number).

4.7 Mise à l'échelle du résultat d'une conversion

L'instruction de mise à l'échelle permet de mettre à l'échelle la valeur résultante d'une opération de conversion en la reproduisant sur une plage de valeurs spécifiée. Lors de l'exécution de l'instruction «SCALE_X» prédéfinies par le logiciel Step7, le nombre à virgule flottante à l'entrée VALUE est mis à l'échelle à la plage des valeurs qui ont été définie par les paramètres MIN et MAX. Le résultat de la mise à l'échelle est un nombre entier qui est stocké à la sortie OUT comme est montré à la Figure 4.4.

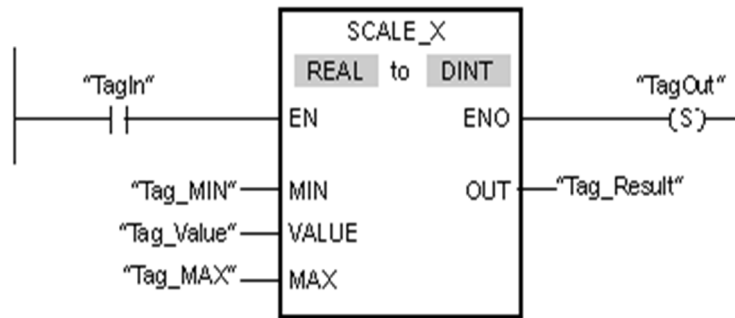


Figure 4.4 – Bloc fonctionnel SCALE_X

Le bloc fonctionnel «SCALE_X» utilise l'équation suivante :

$$OUT = [VALUE * (MAX - MIN)] + MIN$$

La sortie de validation ENO fournit l'état logique 0 quand l'une des conditions suivantes est vraie :

- L'entrée de validation EN fournit l'état logique 0 ;
- La valeur à l'entrée MIN est supérieure ou égale à la valeur à l'entrée MAX ;
- La valeur d'un nombre à virgule flottante indiqué se trouve en dehors de la plage des nombres normalisés selon IEEE-754 ;
- Un débordement se produit ;
- La valeur à l'entrée VALUE est le résultat d'une opération de calcul non valide (NaN : not a number).

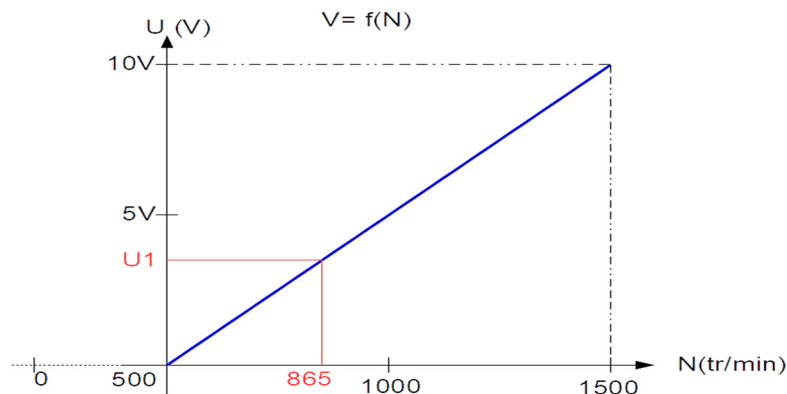
Application 1 : Mesure de vitesse

La vitesse d'une machine tournante doit être enregistré dans une plage par le billet d'un transducteur de mesure permettant le passage de l'intervalle [500÷1500] tr/min à une plage de tension de [0÷10] V.

Pour une vitesse mesurée de 865 tr/min, quelle est la valeur de la tension transmise par le transducteur de mesure. En déduire le contenu numérique dans le registre Reg10.

Solution de l'application 1 :

La caractéristique de la réponse du transducteur est présentée par la figure suivante :



Cette caractéristique est décrite par l'équation suivante :

$$U = K.(N - N_0)$$

On a : $U = 0V$ lorsque $N = 500$ tr/min et $U = 10V$ lorsque $N = 1500$ tr/min, d'où :

$$K = \frac{10}{1500 - 500} = 10^{-2}$$

- Pour : $N = 500$ tr/min $\Rightarrow U_0 = 10^{-2} * (500 - 500) = 0V$;
- Pour : $N = 1500$ tr/min $\Rightarrow U_{max} = 10^{-2} * (1500 - 500) = 10V$.

D'où alors : $N = 865$ tr/min $\Rightarrow U = 10^{-2} * (865 - 500) = 3.65V$.

Cette grandeur analogique sera traitée par l'entrée analogique de l'API pour qu'elle soit lue puis convertie en une information numérique (Conversion analogique numérique) cela signifie que la valeur de la tension et de 3.65V sera stockée en tant qu'information numérique dans le registre d'entrée Reg₁₀.

$$Reg_{10} = \frac{27648}{10} * U = \frac{27648}{10} * 3.65 = 10091$$

Application 2 : contrôle de niveau de citerne

Le niveau d'un liquide dans une citerne est converti en une tension allant de [0÷+10V] par un capteur au quel est associé un transducteur de mesure.

Le transducteur est connecté à la première entrée analogique du Simatic S7-1200 au niveau du quel le signal analogique doit être converti, normalisé et mis à l'échelle.

Proposer un programme Step7 qui répond aux conditions suivantes :

- 0V correspond à un niveau de 100 litres et +10V correspond à un niveau de 1000 litres ;
- Le niveau maximum permissible est 990 litres et le niveau minimum permissible est 110 litres.

Solution de l'application 2 :

Les variables locales nécessaires pour cette application sont :

Adresses	Variables	Types	Commentaires
%IW64	niveau_citerne_EA	Entier	Niveau de la citerne
%Q0.0	citerne_max	Booléen	Signalisation niveau > 990 litres
%Q0.1	citerne_min	Booléen	Signalisation niveau < 110 litres

Il est particulièrement important que les types de données correctes soient utilisés, puisque dans la suite du programme, ils doivent être compatibles avec les fonctions de conversion utilisées. Après avoir déclaré les variables locales on écrit le programme en l'adressage symbolique des variables.

