

**NORME  
INTERNATIONALE  
INTERNATIONAL  
STANDARD**

**CEI  
IEC  
870-5-4**

Première édition  
First edition  
1993-08

---

---

**Matériels et systèmes de téléconduite**

**Partie 5:**

Protocoles de transmission

Section 4: Définition et codages des éléments  
d'information d'application

**Telecontrol equipment and systems**

**Part 5:**

Transmission protocols

Section 4: Definition and coding of  
application information elements



Numéro de référence  
Reference number  
CEI/IEC 870-5-4: 1993

## Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles auprès du Bureau Central de la CEI.

Les renseignements relatifs à ces révisions, à l'établissement des éditions révisées et aux amendements peuvent être obtenus auprès des Comités nationaux de la CEI et dans les documents ci-dessous:

- **Bulletin de la CEI**
- **Annuaire de la CEI**  
Publié annuellement
- **Catalogue des publications de la CEI**  
Publié annuellement et mis à jour régulièrement

## Terminologie

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 50: *Vocabulaire Electrotechnique International* (VEI), qui se présente sous forme de chapitres séparés traitant chacun d'un sujet défini. Des détails complets sur le VEI peuvent être obtenus sur demande. Voir également le dictionnaire multilingue de la CEI.

Les termes et définitions figurant dans la présente publication ont été soit tirés du VEI, soit spécifiquement approuvés aux fins de cette publication.

## Symboles graphiques et littéraux

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera:

- la CEI 27: *Symboles littéraux à utiliser en électro-technique;*
- la CEI 417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles;*
- la CEI 617: *Symboles graphiques pour schémas;*

et pour les appareils électromédicaux,

- la CEI 878: *Symboles graphiques pour équipements électriques en pratique médicale.*

Les symboles et signes contenus dans la présente publication ont été soit tirés de la CEI 27, de la CEI 417, de la CEI 617 et/ou de la CEI 878, soit spécifiquement approuvés aux fins de cette publication.

## Publications de la CEI établies par le même comité d'études

L'attention du lecteur est attirée sur les listes figurant à la fin de cette publication, qui énumèrent les publications de la CEI préparées par le comité d'études qui a établi la présente publication.

## Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available from the IEC Central Office.

Information on the revision work, the issue of revised editions and amendments may be obtained from IEC National Committees and from the following IEC sources:

- **IEC Bulletin**
- **IEC Yearbook**  
Published yearly
- **Catalogue of IEC publications**  
Published yearly with regular updates

## Terminology

For general terminology, readers are referred to IEC 50: *International Electrotechnical Vocabulary* (IEV), which is issued in the form of separate chapters each dealing with a specific field. Full details of the IEV will be supplied on request. See also the IEC Multilingual Dictionary.

The terms and definitions contained in the present publication have either been taken from the IEV or have been specifically approved for the purpose of this publication.

## Graphical and letter symbols

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications:

- IEC 27: *Letter symbols to be used in electrical technology;*
- IEC 417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets;*
- IEC 617: *Graphical symbols for diagrams;*

and for medical electrical equipment,

- IEC 878: *Graphical symbols for electromedical equipment in medical practice.*

The symbols and signs contained in the present publication have either been taken from IEC 27, IEC 417, IEC 617 and/or IEC 878, or have been specifically approved for the purpose of this publication.

## IEC publications prepared by the same technical committee

The attention of readers is drawn to the end pages of this publication which list the IEC publications issued by the technical committee which has prepared the present publication.

**NORME  
INTERNATIONALE  
INTERNATIONAL  
STANDARD**

**CEI  
IEC  
870-5-4**

Première édition  
First edition  
1993-08

---

---

**Matériels et systèmes de téléconduite**

**Partie 5:**  
Protocoles de transmission  
Section 4: Définition et codages des éléments  
d'information d'application

**Telecontrol equipment and systems**

**Part 5:**  
Transmission protocols  
Section 4: Definition and coding of  
application information elements

© CEI 1993 Droits de reproduction réservés — Copyright — all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembe Genève, Suisse

---

---



Commission Electrotechnique Internationale  
International Electrotechnical Commission  
Международная Электротехническая Комиссия

CODE PRIX  
PRICE CODE

**S**

Pour prix, voir catalogue en vigueur  
For price, see current catalogue

## SOMMAIRE

	Pages
AVANT-PROPOS .....	4
INTRODUCTION .....	6
Articles	
1    Domaine d'application et objet .....	6
2    Références normatives .....	8
3    Définitions .....	8
4    Règles de déclaration pour les éléments d'information .....	10
4.1    Types de données .....	10
4.2    Taille des données .....	10
4.3    Position des bits .....	10
4.4    Valeurs et codes .....	12
4.5    Affectation des symboles de fonction et des fonctions .....	14
4.6    Identificateurs de séquences de champs de données .....	14
4.7    Taille de champ variable .....	14
4.8    Champs de données répétitifs .....	16
4.9    Combinaisons logiques de champs de données .....	16
4.10    Empaquetage et ordre de transport des champs de données .....	16
5    Eléments d'information normalisés .....	18
5.1    Type 1: ENTIER SANS SIGNE (UI) .....	18
5.2    Type 2: ENTIER (I) .....	18
5.3    Type 3 : NOMBRE À VIRGULE FIXE SANS SIGNE (UF) .....	18
5.4    Type 4: NOMBRE À VIRGULE FIXE (F) .....	20
5.5    Type 5: RÉEL (R) .....	20
5.6    Type 6: CHAÎNE BINAIRE (BS) .....	20
5.7    Type 7: CHAÎNE D'OCTETS (OS) .....	20
6    Ensemble d'éléments d'information .....	22
6.1    Type 1: ENTIER SANS SIGNE (UI) .....	22
6.2    Type 2: ENTIER (I) .....	24
6.3    Type 3: NOMBRE À VIRGULE FIXE SANS SIGNE (UF) .....	26
6.4    Type 4: NOMBRE À VIRGULE FIXE (F) .....	26
6.5    Type 5: RÉEL (R) .....	28
6.6    Type 6: CHAÎNE BINAIRE (BS) .....	30
6.7    Type 7: CHAÎNE D'OCTETS (OS) .....	30
6.8    ELÉMENTS D'INFORMATION COMPOSITE (CP) .....	32

## CONTENTS

	Page
FOREWORD .....	5
INTRODUCTION .....	7
Clause	
1 Scope and object .....	7
2 Normative references .....	9
3 Definitions .....	9
4 Declaration rules for information elements .....	11
4.1 Data types .....	11
4.2 Data size .....	11
4.3 Bit position .....	11
4.4 Values and codes .....	13
4.5 Assignment of function symbols and functions .....	15
4.6 Identifiers of data field sequences .....	15
4.7 Variable field size .....	15
4.8 Repetitive data fields .....	17
4.9 Logical combinations of data fields .....	17
4.10 Packing and order of transport of data field .....	17
5 Standard information elements .....	19
5.1 Type 1: UNSIGNED INTEGER (UI) .....	19
5.2 Type 2: INTEGER (I) .....	19
5.3 Type 3: UNSIGNED FIXED POINT NUMBER (UF) .....	19
5.4 Type 4: FIXED POINT NUMBER (F) .....	21
5.5 Type 5: REAL (R) .....	21
5.6 Type 6: BITSTRING (BS) .....	21
5.7 Type 7: OCTETSTRING (OS) .....	21
6 Set of information elements .....	23
6.1 Type 1: UNSIGNED INTEGER (UI) .....	23
6.2 Type 2: INTEGER (I) .....	25
6.3 Type 3: UNSIGNED FIXED POINT NUMBER (UF) .....	27
6.4 Type 4: FIXED POINT NUMBER (F) .....	27
6.5 Type 5: REAL (R) .....	29
6.6 Type 6: BITSTRING (BS) .....	31
6.7 Type 7: OCTETSTRING (OS) .....	31
6.8 COMPOUND INFORMATION ELEMENTS (CP) .....	33

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE**

**Partie 5: Protocoles de transmission**

**Section 4: Définition et codages  
des éléments d'information d'application**

AVANT-PROPOS

- 1) La CEI (Commission Electrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI, entre autres activités, publie des Normes internationales. Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par les comités d'études où sont représentés tous les Comités nationaux s'intéressant à ces questions, expriment dans la plus grande mesure possible un accord international sur les sujets examinés.
- 3) Ces décisions constituent des recommandations internationales publiées sous forme de normes, de rapports techniques ou de guides et agréées comme telles par les Comités nationaux.
- 4) Dans le but d'encourager l'unification internationale, les Comités nationaux de la CEI s'engagent à appliquer de façon transparente, dans toute la mesure possible, les Normes internationales de la CEI dans leurs normes nationales et régionales. Toute divergence entre la norme de la CEI et la norme nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.

La norme internationale CEI 870-5-4 a été établie par le comité d'études 57 de la CEI: Téléconduite, téléprotection et télécommunications connexes pour systèmes électriques de puissance.

Le texte de cette norme est issu des documents suivants:

DIS	Rapport de vote
57(BC)62	57(BC)67

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

## TELECONTROL EQUIPMENT AND SYSTEMS

## Part 5: Transmission protocols

Section 4: Definition and coding  
of application information elements

## FOREWORD

- 1) The IEC (International Electrotechnical Commission) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international cooperation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 3) They have the form of recommendations for international use published in the form of standards, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.

International Standard IEC 870-5-4 has been prepared by IEC technical committee 57: Telecontrol, teleprotection and associated telecommunications for electric power systems.

The text of this standard is based on the following documents:

DIS	Report on voting
57(CO)62	57(CO)67

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

## **MATÉRIELS ET SYSTÈMES DE TÉLÉCONDUITE**

### **Partie 5: Protocoles de transmission**

#### **Section 4: Définition et codages des éléments d'information d'application**

##### **INTRODUCTION**

La présente section de la CEI 870-5 définit les règles de déclaration normalisées pour les données d'application dans les systèmes de téléconduite.

##### **1 Domaine d'application et objet**

La présente section de la CEI 870-5 s'applique aux matériels et aux systèmes de téléconduite, à transmission en série de données binaires, destinés à la surveillance et à la conduite de processus géographiquement dispersés.

Cette section de la CEI 870-5 donne les règles pour la définition des éléments d'information et présente un ensemble d'éléments d'information, correspondant en particulier aux variables numériques et analogiques de processus fréquemment utilisées dans les applications de téléconduite.

L'article 4 présente les règles de syntaxe pour la définition des éléments d'information spécifiques de l'application. Ces règles comprennent les méthodes pour les déclarations sémantiques, c'est à dire l'affectation de l'interprétation fonctionnelle des éléments d'information définis.

L'article 5 présente les principaux types d'éléments d'information normalisés et introduit des sous-types particuliers d'éléments d'information normalisés en utilisant la méthode de déclaration définie à l'article 4.

L'article 6 présente un ensemble d'éléments d'information fréquemment utilisés dans les applications de téléconduite. Ces éléments et les applications mentionnées et recommandées pour ces éléments ne sont que des recommandations. Les déclarations définitives des éléments d'information doivent être fixées dans les profils d'application.

Les éléments d'information simples, les séquences ou combinaisons d'éléments d'information peuvent former un objet d'information qui est identifié par une adresse d'objet et par une spécification de structure d'objet décrite dans la CEI 870-5-3.



## TELECONTROL EQUIPMENT AND SYSTEMS

### Part 5: Transmission protocols

#### Section 4: Definition and coding of application information elements

#### INTRODUCTION

This section of IEC 870-5 defines standard declaration rules for application data in telecontrol systems.

#### 1 Scope and object

This section of IEC 870-5 applies to telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes.

This section of IEC 870-5 gives rules for defining information elements and presents a set of information elements, in particular of digital and analog process variables, that are frequently used in telecontrol applications.

Clause 4 presents syntactic rules for defining application specific information elements. These rules comprise methods for semantic declarations, that is assignments of the functional interpretation of the defined information fields.

Clause 5 applies the declaration method to the basic data types defined in Clause 4 and introduces particular subtypes of data.

Clause 6 presents a set of information elements that are frequently used in telecontrol applications. These elements and the mentioned recommended applications for these elements are recommendations only. Definitive declarations of information elements have to be fixed in application profiles.

Single information elements, sequences or combinations of information elements may form an information object that is identified by an object address and by an object structures specification described in IEC 870-5-3.

## 2 Références normatives

Les documents normatifs suivants contiennent des dispositions qui, par suite de la référence qui y est faite, constituent de dispositions valables pour la présente section de la CEI 870-5. Au moment de la publication, les éditions indiquées étaient en vigueur. Tout document normatif est sujet à révision et les parties prenantes aux accords fondés sur la présente section de la CEI 870-5 sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des documents normatifs indiqués ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur.

CEI 50(371): 1984, *Vocabulaire Electrotechnique International (VEI) - Chapitre 371: Téléconduite.*

CEI 870-1-1: 1988, *Matériels et systèmes de téléconduite - Première partie: Considérations générales - Section un: Principes généraux.*

CEI 870-1-3: 1990, *Matériels et systèmes de téléconduite - Première partie: Considérations générales - Section trois: Glossaire.*

CEI 870-5-3: 1992, *Matériels et systèmes de téléconduite - Cinquième partie: Protocoles de transmission - Section trois: Structure générale des données d'application.*

ISO/IEC 10000-1: 1990, *Systèmes de traitement de l'information - Cadre et taxonomie des profils internationaux normalisés - Partie 1: Cadre (publiée actuellement en anglais seulement).*

Norme IEEE 754, 1985.

Recommandation V.3 du CCITT: 1972, *Alphabet International n° 5.*

## 3 Définitions

Pour les besoins de la présente section de la CEI 870-5, les définitions suivantes s'appliquent:

**3.1 type de données:** Méthode définie de présentation des données.

*Par exemple: type de données ENTIER pour des nombres entiers ou type de données CHAÎNE D'OCTETS pour un assemblage d'octets.*

**3.2 taille des données:** Longueur en bits d'un champ d'un type de données spécifié.

**3.3 élément d'information:** Quantité (d'information) variable indivisible bien définie.

*Par exemple: valeur mesurée ou information de signalisation double.*

**3.4 profil:** Norme ou ensemble de normes, accompagnée, lorsque applicable, de l'identification des classes choisies, des sous-ensembles, des options et des paramètres de ces normes de base nécessaires pour accomplir une fonction particulière (voir 3.1.2 de l'ISO/IEC 10000-1).\*

---

\* Disponible en anglais seulement.

## 2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this section of IEC 870-5. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this section of IEC 870-5 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 50(371): 1984, *International Electrotechnical Vocabulary (IEV) - Chapter 371: Telecontrol*.

IEC 870-1-1: 1988, *Telecontrol equipment and systems - Part 1: General considerations - Section One: General principles*.

IEC 870-1-3: 1990, *Telecontrol equipment and systems - Part 1: General considerations - Section Three: Glossary*.

IEC 870-5-3: 1992, *Telecontrol equipment and systems - Part 5: Transmission protocols - Section Three: General structure of application data*.

ISO/IEC 10000-1: 1990, *Information technology - Framework and taxonomy of international standardized profiles - Part 1: Framework*.

IEEE Standard 754, 1985.

CCITT Recommendation V.3: 1972, *International Alphabet No. 5*.

## 3 Definitions

For the purpose of this section of IEC 870-5, the following definitions apply:

**3.1 data type:** Defined method of data presentation.

*For example: data type INTEGER for whole numbers or data type OCTETSTRING for an assembly of octets*

**3.2 data size:** Field length of a specified data type in bits.

**3.3 information element:** A well defined variable quantity which is indivisible.

*For example: measured value or double-point information.*

**3.4 profile:** A set of one or more base standards, and, where applicable, the identification of chosen classes, subsets, options and parameters of those base standards, necessary for accomplishing a particular function (see 3.1.2 of ISO/IEC 10000-1).

#### 4 Règles de déclaration pour les éléments d'information

Les éléments d'information sont définis par la méthode de déclaration syntactique et sémantique suivante:

	Type de données	Taille des données	Position des bits	Valeurs et code	Fonction
Nom de l'élément d'information :=	TYPE	i	[ p <sub>1</sub> .. p <sub>1</sub> +i-1]	<v <sub>1</sub> ... v <sub>n</sub> code>	:= Fonction

Le symbole d'affectation " := " est utilisé pour associer des déclarations de champs d'information aux noms des éléments d'information et pour associer des fonctions ou symboles de fonction (sigles) aux déclarations de champs.

##### 4.1 Types de données

Tableau 1 - Types de données

Numéro du type	Type de données	Symbole	Signification
1	ENTIER SANS SIGNE	UI	Nombre entier positif
2	ENTIER	I	Nombre entier positif ou négatif
3	NOMBRE À VIRGULE FIXE SANS SIGNE	UF	Nombre à virgule fixe positif
4	NOMBRE À VIRGULE FIXE	F	Nombre à virgule fixe positif ou négatif
5	RÉEL	R	Nombre à virgule flottante positif ou négatif
6	CHAÎNE BINAIRE	BS	Assemblage de bits indépendants <sup>1)</sup>
7	CHAÎNE D'OCTETS	OS	Assemblage d'octets

<sup>1)</sup> Le type de données BOOLÉEN est une CHAÎNE BINAIRE de taille 1.

##### 4.2 Taille des données

La *taille des données* *i*, notée directement après le symbole de la notation du *type de données*, est un nombre cardinal qui spécifie la longueur en bits d'un champ de données.

##### 4.3 Position des bits

La position des bits d'un champ spécifié de *taille des données* *i* est notée entre crochets [p<sub>1</sub>..p<sub>n</sub>], où p<sub>1</sub> et p<sub>n</sub> indiquent le premier et le dernier bits du champ. L'ordre des bits est donné par le schéma suivant:

#### 4 Declaration rules for information elements

Information elements are defined by the following syntactic and semantic declaration method:

	Data type	Data size	Bit position	Values and code	Function
Name of information element :=	TYPE	i	[ p <sub>1</sub> ..p <sub>1</sub> + i-1]	<v <sub>1</sub> ... v <sub>n</sub> code>	:= Function

The assignment symbol "==" is used to associate information field declarations to the name of information elements and to associate functions to field declarations.

#### 4.1 Data types

Table 1 - Data types

Type number	Data types	Symbol	Meaning
1	UNSIGNED INTEGER	UI	Positive whole number
2	INTEGER	I	Positive or negative whole number
3	UNSIGNED FIXED POINT	UF	Positive fixed point number
4	FIXED POINT	F	Positive or negative fixed point number
5	REAL	R	Positive or negative floating-point number
6	BITSTRING	BS	Assembly of independent bits <sup>1)</sup>
7	OCTETSTRING	OS	Assembly of octets

<sup>1)</sup> The data type BOOLEAN is a BITSTRING of size 1.

#### 4.2 Data size

The *data size* *i*, noted directly after the symbol of the *data type* notation, is a cardinal number that specifies the length of a data field in bits.

#### 4.3 Bit position

The bit position of a specified field of *data size* *i* is denoted in square brackets [p<sub>1</sub>..p<sub>n</sub>], where p<sub>1</sub> and p<sub>n</sub> denote the first and the last bits of the field. The order of bits is shown in the following scheme:

Tableau 2 - Position des bits

Octets	Bits							
	8	7	6	5	4	3	2	1
1	8	7	6	5	4	3	2	1
2	16	15	14	13	12	11	10	9
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
j	8j	8j-1	8j-2	8j-3	8j-4	8j-5	8j-6	8j-7

Par exemple, si une CHAÎNE BINAIRE de taille 6 occupe le champ encadré de filet gras dans le tableau 2, elle est alors notée *BS6[7..12]*. Le dernier bit d'un champ de taille *i* qui commence en position de bit  $p_1$  est donné par  $p_n = p_1 + i - 1$ . Si la taille du champ est de 1, la position est alors indiquée par une notation unique [ $p_1$ ] entre crochets.

#### 4.4 Valeurs et codes

Lorsque cela est applicable, une gamme de valeurs choisie et un code choisi pour les valeurs du champ de données déclaré sont indiqués entre signes inférieur et supérieur:  $\langle v_1..v_n \text{ code} \rangle$ . En général, la déclaration comprend l'étendue des valeurs admises et un terme qui désigne le code utilisé. Les termes qui désignent les codes sont: code binaire (BIN), décimal codé binaire (DCB), code Gray, code k-parmi-n, code ASCII, etc. La déclaration du code par défaut est le code binaire, c'est-à-dire que si aucun terme n'est employé pour caractériser le code, les valeurs données du champ de données sont alors interprétées comme des valeurs binaires.

Si le *type de données* est un nombre (ENTIER, NOMBRE À VIRGULE FIXE ou RÉEL), le code est alors défini sans ambiguïté par la notation de l'étendue de nombres et par le code utilisé:

$\langle \text{valeur}_1 \text{ (limite inférieure) .. valeur}_n \text{ (limite supérieure) code} \rangle$ .

Par exemple, un compteur qui énumère les jours de la semaine utilise une déclaration d'ENTIER SANS SIGNE de taille 3 (UI3) suivie de la déclaration du code et de l'étendue des valeurs:  $\langle 1..7 \text{ BIN} \rangle$ , ou plus brièvement:  $\langle 1..7 \rangle$ .

Les valeurs individuelles sont déclarées sous la forme:  $\langle \text{valeur code} \rangle$  (par exemple,  $\langle 3 \text{ BIN} \rangle$  ou  $\langle 3 \rangle$  pour mercredi dans le cas présent).

Dans le cas du *type de données* CHAÎNE D'OCTETS, un code de caractères à 8 bits défini, comme par exemple l'ensemble de caractères ASCII à 8 bits, est choisi par la déclaration du code:  $\langle \text{ensemble de caractères ASCII à 8 bits} \rangle$ . Dans ce cas, l'ensemble des valeurs est défini par l'Alphabet International n° 5 de la Recommandation V.3 du CCITT, de manière que cette déclaration de code suffise.

Table 2 - Bit positions

Octets	Bits							
1	8	7	6	5	4	3	2	1
2	16	15	14	13	12	11	10	9
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
j	8j	8j-1	8j-2	8j-3	8j-4	8j-5	8j-6	8j-7

For example, if a BITSTRING of size 6 occupies the field surrounded with thick lines in table 2, then it is denoted as  $BS6[7..12]$ . The last bit of a field of size  $i$  that begins in bit position  $p_1$  is given by  $p_n = p_1 + i - 1$ . If the field size is 1, then the position is denoted by a single notation  $[p_1]$  in the brackets.

#### 4.4 Values and codes

If applicable, a selected range and a selected code of values of the declared data field is denoted within angle brackets:  $\langle v_1..v_n \text{ code} \rangle$ . In general this is declared by the range of admitted values and by a term that identifies the used code. Terms that identify codes are: binary code (BIN), binary coded decimal (BCD), Gray-code, k-of-n code, ASCII-code, etc. The default code declaration is the binary code, that is, if no terms are used to characterize the code, then given values of the data field are interpreted as binary values.

If the *data type* is a number (INTEGER, FIXED POINT or REAL), then the code is unambiguously defined by the notation of the range of numbers and by the used code :

$\langle value_1 \text{ (lower limit)}..value_n \text{ (upper limit) code} \rangle$ .

For example, a counter that enumerates the days of the week uses an UNSIGNED INTEGER declaration with size 3 (UI3) followed by the declaration of code and range of values:  $\langle 1..7 \text{ BIN} \rangle$ , or shorter:  $\langle 1..7 \rangle$ .

Individual values are declared as  $\langle value \text{ code} \rangle$  (e.g.  $\langle 3 \text{ BIN} \rangle$  or  $\langle 3 \rangle$  for wednesday in this example).

In case of the *data type* OCTETSTRING, a defined 8 bit character code such as e.g. the 8 bit ASCII character set is selected by the code declaration:  $\langle 8 \text{ bit ASCII character set} \rangle$ . In this case the set of values is defined by the International Alphabet No. 5 of CCITT recommendation V.3, so that this code declaration suffices.

#### 4.5 Affectation des symboles de fonction et des fonctions

Le but fonctionnel d'un champ de données spécifié est déclaré par l'utilisation du symbole d'affectation " := "

*type de données*  $i [p_1..p_i]$  <code> := fonction

Les sigles doivent être introduits en les définissant en toutes lettres et en utilisant le symbole d'équivalence " = ".

*fonction* = FCT

Par exemple: *erreur* = ER := BSI[8] signifie qu'un bit d'erreur dont le sigle est ER est situé dans une CHAÎNE BINAIRE de taille 1 en position (de bit) 8 d'un champ de données.

#### 4.6 Identificateurs de séquences de champs de données

Si un élément d'information est composé de différents champs de données, il est alors déclaré comme un champ de données COMPOSITE ou SÉQUENTIEL.

Séquence de champs	Symbole	Signification
COMPOSITE	CP	Séquence de champs de données avec attributions de bits consécutives
SÉQUENTIEL	SQ	Séquence de champs de données avec attributions de bits commençant par 1 pour chaque champ de données

NOTE - Dans les champs de données les séquences sont utilisées avec des éléments d'information répétés.

Les champs de données COMPOSITE sont déclarés en inscrivant les champs de données individuels séparés par des virgules ou énumérés dans une colonne, entre parenthèses. La liste suivante énonce les types de données, les tailles, les attributions de bits et les buts fonctionnels des champs de données individuels. Le premier champ de données déclaré commence avec une position de bit 1, les autres champs utilisant des attributions de bits consécutives:

*Elément d'information* := CPi { *champ de données 1, champ de données 2, ...* }  
*champ de données 1* := *type de données 1* taille  $i_1 [1..i_1]$  := *fonction 1*  
*champ de données 2* := *type de données 2* taille  $i_2 [i_1+1..i_1+i_2]$  := *fonction 2*  
 etc.

Les champs de données SÉQUENTIELS sont déclarés comme les champs de données composites, mais chaque champ de données commence avec une attribution de bits commençant par 1:

*Elément d'information* := SQi { *champ de données 1, champ de données 2, ...* }  
*champ de données 1* := *type de données 1* taille  $i_1 [1..i_1]$  := *fonction 1*  
*champ de données 2* := *type de données 2* taille  $i_2 [1..i_2]$  := *fonction 2*  
 etc.

#### 4.7 Taille de champ variable

Les éléments d'information définis d'une manière générale pour les tailles variables  $i$  ( $i = \text{entier} > 0$ ) sont déclarés en notant les positions de bits et les spécifications de code en fonction de la taille  $i$ :

*type de données*  $i [1..i]$  < valeur (=  $f(i)$ ) code >, voir définitions des éléments d'information normalisés à l'article 5.



#### 4.5 Assignment of function symbols and functions

The functional purpose of a specified data field is declared by using the assignment symbol " := "

$$\text{data type } i [p_1..p_i] <\text{code}> := \text{function}$$

Acronyms have to be introduced by full text descriptions and by using the equivalence symbol " = ".

$$\text{function} = \text{FCT}$$

For example:  $\text{error} = \text{ER} := \text{BS1}[8]$  means that an error bit with the acronym ER is located in a BITSTRING of size 1 in bit position 8 of a data field.

#### 4.6 Identifiers of data field sequences

If an information element is composed of different data fields, then it is either declared as a COMPOUND or as a SEQUENCE data field.

Field sequence	Symbol	Meaning
COMPOUND	CP	Sequence of data fields with successive bit allocations
SEQUENCE	SQ	Sequence of data fields with bit allocations beginning with 1 per data field

NOTE – Sequences are used in data fields with repeated information elements.

COMPOUND data fields are declared by listing individual data fields separated by commas or listed in a column, within curly brackets. A following list declares the data types, the sizes, the bit allocations and the functional purposes of the individual data fields. The first declared data field begins with bit position 1, the other fields use successive bit allocations:

*Information element := CPi* {data field 1, data field 2, ... }  
 $\text{data field 1} := \text{data type 1 size } i_1 [1..i_1] \quad := \text{function 1}$   
 $\text{data field 2} := \text{data type 2 size } i_2 [i_1+1..i_1+i_2] \quad := \text{function 2}$   
 etc.

SEQUENCE data fields are declared as compound data fields, however each data field begins with bit allocation 1:

*Information element := SQi* {data field 1, data field 2, ... }  
 $\text{data field 1} := \text{data type 1 size } i_1 [1..i_1] \quad := \text{function 1}$   
 $\text{data field 2} := \text{data type 2 size } i_2 [1..i_2] \quad := \text{function 2}$   
 etc.

#### 4.7 Variable field size

Information elements defined in a general way for variable sizes  $i$  ( $i = \text{integer} > 0$ ) are declared by noting bit positions and code specifications as a function of size  $i$ :

$\text{data type } i [1..i] <\text{value} (= f(i)) \text{ code}>$ , see definitions of standard information elements in Clause 5.

#### 4.8 Champs de données répétitifs

Si un champ de données est composé d'un sous-champ de taille  $i$  qui est répété en  $n$  boucles, il peut alors être déclaré par deux méthodes alternatives, à savoir:

- a) Déclaration du sous-champ  $i$  répété en  $n$  boucles:

*Élément d'information de taille  $ni$  :=  $n$  type de données  $i$  [1.. $i$ ] < valeurs et code du sous-champ  $i$  >*

- b) Déclaration du champ entier  $ni$ :

*Élément d'information de taille  $ni$  := type de données  $ni$  [1.. $ni$ ] < valeurs et code du champ  $ni$  >*

Les deux méthodes de déclaration dans le cas d'entiers codés DCB sont indiquées en 5.1.2.

Chaque fois qu'un nombre (ou un élément réservant la place d'un nombre  $n$ ) précède la déclaration du type de données, la déclaration des positions de bits commence alors à 1 et les valeurs et codes déclarés se réfèrent au sous-champ répété, voir par exemple la définition de l'élément de champ "Etat à 8 bits + Détection de transitoire" en 6.6.

#### 4.9 Combinaisons logiques de champs de données

Dans certaines applications, les fonctions résultent de combinaisons logiques entre champs. L'exemple typique est le type de données RÉEL (voir 6.5), où les valeurs résultantes sont définies comme des combinaisons logiques "et" entre les valeurs des champs de données "Fraction" et "Exposant", ou certains champs de commande dans lesquels la fonction d'un champ dépend des valeurs d'un autre champ. Dans ces applications, les termes "et" et "ou" combinés avec des valeurs spécifiques ou des intervalles de valeurs des champs de données impliqués sont utilisées pour exprimer les fonctions résultantes:

*type de données 1 < intervalle de valeur 1 > et (ou: ou) type de données 2 < intervalle de valeur 2 > := fonction*

#### 4.10 Empaquetage et ordre de transport des champs de données

La manière de présenter les éléments de données suit en général les exigences des applications fonctionnelles voulues pour les contenus des données. Ainsi, les nombres sont en général présentés comme il est d'usage de les écrire et de les lire, à savoir de gauche à droite, dans l'ordre des puissances décroissantes de leurs bases.

Les tailles des éléments d'information qui ne sont pas multiples de 8 bits permettent un empaquetage condensé des séquences d'éléments d'information pour obtenir des structures en octets du champ d'information.

Pour les éléments d'information plus longs qu'un octet, l'octet noté en haut est livré en premier pour un transport en mode 1 ou en dernier pour un transport en mode 2. Le choix des modes est spécifié dans le profil d'application utilisé.

#### 4.8 Repetitive data fields

If a data field is composed of a subfield of size  $i$  that is repeated in  $n$  loops, then it can be declared by two alternative methods, namely:

- a) Declaration of subfield  $i$  that is repeated in  $n$  loops:

*Information element of size  $n_i := n$  data type  $i$  [1.. $i$ ] < values and code of subfield  $i$  >*

- b) Declaration of the whole field  $n_i$ :

*Information element of size  $n_i :=$  data type  $n_i$  [1.. $n_i$ ] < values and code of field  $n_i$  >*

The two declaration methods in case of BCD integers are given in 5.1.2.

Whenever a number (or a place holder for a number  $n$ ) precedes the declaration of the data type, then the declaration of bit positions begins with 1 and the declared values and codes refer to the repeated subfield, see for example definition of the field element "8 bit Status+Transient Detection" in 6.6.

#### 4.9 Logical combinations of data fields

In some applications functions result from logical combinations between fields. Typical examples are the data type REAL (see clause 6.5), where resulting values are defined as logical "and" combinations between values of the data fields "Fraction" and "Exponent", or certain Control fields in which the function of one field depends on values of another field. In these Applications the terms "and" and "or" in combination with specified values or ranges of values of the involved data fields are used to express the resulting functions:

*data type 1 < value range 1 > and (or: or) data type 2 < value range 2 > := function*

#### 4.10 Packing and order of transport of data field

The way of presenting data elements usually follows the requirements of intended functional applications of the data contents. Thus numbers are usually presented as we are used to write and read them, namely from left to right with decreasing powers of their bases.

Sizes of Information elements that are not multiples of 8 bits allow condensed packing of sequences of information elements to achieve octet structures of the information field.

For information elements longer than one octet, the octet noted at the top is delivered first for transportation in mode 1 or is delivered last in mode 2. The choice of modes is specified in the application profile used.

## 5 Éléments d'information normalisés

### 5.1 Type 1: ENTIER SANS SIGNE (UI) (nombre entier positif)

#### 5.1.1 Type 1.1: Entier binaire sans signe = UIi

UIi := UIi [1..i] < 0..+2<sup>i-1</sup> >

i	i-1	-	-	3	2	1
2 <sup>i-1</sup>	2 <sup>i-2</sup>	-	-	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
MSB		Intervalle: 0 .. + 2 <sup>i-1</sup> - 1				LSB

#### 5.1.2 Type 1.2: Entier sans signe codé DCB = nUI4BCD n = nombre de chiffres décimaux

nUI4BCD := nUI4[1..4] < 0..9 BCD >  
 := UI4n[1..4n] < 0..10<sup>n</sup> - 1 BCD >  
 Chiffre 1 := UI4[1..4] < 0..9 BCD >  
 Chiffre 10 := UI4[5..8] < 0..9 BCD >  
 Chiffre 100 := UI4[9..12] < 0..9 BCD >  
 :  
 Chiffre 10<sup>n-1</sup> := UI4[4n-3..4n] < 0..9 BCD >

8	7	6	5	4	3	2	1
Chiffre 10				Chiffre 1			
8	4	2	1	8	4	2	1
-				Chiffre 100			
-				8	4	2	1
Chiffre 10 <sup>n-1</sup>				-			
8	4	2	1	-	-	-	-

Intervalle: 0 .. 10<sup>n</sup> - 1

### 5.2 Type 2: ENTIER (I) (nombre entier positif ou négatif)

#### 5.2.1 Type 2.1: Entier binaire avec signe = Ii

Les nombres négatifs sont représentés en complément à deux

Ii := CPi { UIi-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Signe } := Ii[1..i] < -2<sup>i-1</sup> .. +2<sup>i-1</sup> - 1 >  
 Signe = S := BS1[i] S < 0 > := positif, S < 1 > := négatif

S < 1 > et UIi-1 < 0 > := -2<sup>i-1</sup>  
 S < 1 > et UIi-1 < 1.2<sup>i-1</sup>-1 > := -2<sup>i-1</sup>+1 .. -1  
 S < 0 > et UIi-1 < 0.2<sup>i-1</sup>-1 > := 0 .. 2<sup>i-1</sup>-1

i	i-1	i-2	-	-	2	1
S	2 <sup>i-2</sup>	2 <sup>i-3</sup>	-	-	2 <sup>1</sup>	2 <sup>0</sup>

Intervalle: -2<sup>i-1</sup> .. +2<sup>i-1</sup> - 1

#### 5.2.2 Type 2.2: Entier avec signe codé DCB = I4n+1BCD n = nombre de chiffres décimaux

I4n+1BCD := I4n+1[1..4n+1] < 1-10<sup>n</sup> .. 10<sup>n</sup>-1 BCD >  
 := CP4n+1 { nBCDdigit, Signe }  
 BCDdigit := UI4[1..4] < 0..9 BCD >  
 Signe = S := BS1 [4n+1]  
 S < 0 > := positif  
 S < 1 > := négatif

8	7	6	5	4	3	2	1
Chiffre 10				Chiffre 1			
8	4	2	1	8	4	2	1
-				Chiffre 100			
-				8	4	2	1
S				Chiffre 10 <sup>n-1</sup>			
-				8	4	2	1

Intervalle: 1 - 10<sup>n</sup> .. 10<sup>n</sup> - 1

### 5.3 Type 3: NOMBRE À VIRGULE FIXE SANS SIGNE (UF) (nombre à virgule fixe positif)

#### 5.3.1 Type 3.1: Nombre normalisé à virgule fixe sans signe = UFi

UFi := UFi [1..i] < 0..1-2<sup>-i</sup> >

i	i-1	-	-	2	1
2 <sup>-1</sup>	2 <sup>-2</sup>	-	-	2 <sup>-i+1</sup>	2 <sup>-i</sup>

Intervalle: 0 .. +1 - 2<sup>-i</sup>

#### 5.3.2 Type 3.2: Nombre à virgule fixe sans signe normalisé à +2<sup>j</sup> × 100% = UFi,j

UFi,j := UFi,j [1..i] < 0..+2<sup>j</sup>-2<sup>j-i</sup> >

i	i-1	-	-	2	1
2 <sup>j-1</sup>	2 <sup>j-2</sup>	-	-	2 <sup>j-i+1</sup>	2 <sup>j-i</sup>

Intervalle: 0 .. +2<sup>j</sup>-2<sup>j-i</sup>

**5 Standard information elements**

**5.1 Type 1: UNSIGNED INTEGER (UI) (positive whole number)**

**5.1.1 Type 1.1: Unsigned binary = UIi**

UIi := UIi [1..i] < 0..2<sup>i-1</sup> >

i	i-1	-	-	3	2	1
2 <sup>i-1</sup>	2 <sup>i-2</sup>	-	-	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
MSB		Range: 0 .. + 2 <sup>i-1</sup> - 1				LSB

**5.1.2 Type 1.2: Unsigned binary coded decimal integer = nUI4BCD n = number of decimal digits**

nUI4BCD := nUI4[1..4] < 0..9 BCD >  
 := UI4n[1..4n] < 0..10<sup>n</sup> - 1 BCD >  
 Digit 1 := UI4[1..4] < 0..9 BCD >  
 Digit 10 := UI4[5..8] < 0..9 BCD >  
 Digit 100 := UI4[9..12] < 0..9 BCD >

8	7	6	5	4	3	2	1
Digit 10				Digit 1			
8	4	2	1	8	4	2	1
-				Digit 100			
-				8 4 2 1			
Digit 10 <sup>n-1</sup>				-			
8 4 2 1				-			

Range: 0 .. 10<sup>n</sup> - 1

**5.2 Type 2: INTEGER (I) (positive or negative whole number)**

**5.2.1 Type 2.1: Signed binary integer = Ii**

negative numbers are presented in two's complement

Ii := CPi { UIi-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Sign } := Ii[1..i] < -2<sup>i-1</sup> .. +2<sup>i-1</sup> - 1 >  
 Sign = S := BS1[i] S < 0 > := positive, S < 1 > := negative

S < 1 > and UIi-1 < 0 > := -2<sup>i-1</sup>  
 S < 1 > and UIi-1 < 1..2<sup>i-1</sup>-1 > := -2<sup>i-1</sup>+1 .. -1  
 S < 0 > and UIi-1 < 0..2<sup>i-1</sup>-1 > := 0 .. 2<sup>i-1</sup>-1

i	i-1	i-2	-	-	2	1
S	2 <sup>i-2</sup>	2 <sup>i-3</sup>	-	-	2 <sup>1</sup>	2 <sup>0</sup>
Range: -2 <sup>i-1</sup> .. +2 <sup>i-1</sup> - 1						

**5.2.2 Type 2.2: Signed binary coded decimal integer = I4n+1BCD n = number of decimal digits**

I4n+1BCD := I4n+1[1..4n+1] < 1-10<sup>n</sup> .. 10<sup>n</sup>-1 BCD >  
 := CP4n+1 { nBCDdigit, Sign }  
 BCDdigit := UI4[1..4] < 0..9 BCD >  
 Sign = S := BS1 [4n+1]  
 S < 0 > := positive  
 S < 1 > := negative

8	7	6	5	4	3	2	1
Digit 10				Digit 1			
8	4	2	1	8	4	2	1
-				Digit 100			
-				8 4 2 1			
-				Digit 10 <sup>n-1</sup>			
-				8 4 2 1			
S				-			

Range: 1 - 10<sup>n</sup> .. 10<sup>n</sup> - 1

**5.3 Type 3: UNSIGNED FIXED POINT NUMBER (UF) (positive fixed point number)**

**5.3.1 Type 3.1: Normalized unsigned fixed point number = UFi**

UFi := UFi [1..i] < 0..1-2<sup>-i</sup> >

i	i-1	-	-	2	1
2 <sup>-1</sup>	2 <sup>-2</sup>	-	-	2 <sup>-i+1</sup>	2 <sup>-i</sup>

Range: 0 .. +1 - 2<sup>-i</sup>

**5.3.2 Type 3.2: Unsigned fixed point number normalized to +2<sup>j</sup> × 100% = UFi,j**

UFi,j := UFi,j [1..i] < 0..+2<sup>j</sup>-2<sup>j-i</sup> >

i	i-1	-	-	2	1
2 <sup>j-1</sup>	2 <sup>j-2</sup>	-	-	2 <sup>j-i+1</sup>	2 <sup>j-i</sup>

Range: 0 .. +2<sup>j</sup>-2<sup>j-i</sup>

5.4 Type 4: NOMBRE À VIRGULE FIXE (F) (nombre à virgule fixe négatif ou positif)

Les nombres négatifs sont représentés en complément à deux

5.4.1 Type 4.1: Nombre normalisé à virgule fixe avec signe = Fi

Fi := CPi { Uli-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Signe } := Fi[1..i] < -1..+1-2<sup>1-i</sup> >  
 Signe = S := BS1[i]  
 S < 0 > := positif, S < 1 > := négatif  
 S < 1 > et Uli-1 < 0 > := -1  
 S < 1 > et Uli-1 < 1..2<sup>i-1</sup>-1 > := -1+2<sup>1-i</sup> .. -2<sup>1-i</sup>  
 S < 0 > et Uli-1 < 0..2<sup>i-1</sup>-1 > := 0 .. 1-2<sup>1-i</sup>

	i	i-1	i-2	-	-	2	1
S	2 <sup>-1</sup>	2 <sup>-2</sup>	-	-	2 <sup>-i+2</sup>	2 <sup>-i+1</sup>	

Intervalle: -1 .. +1 -2<sup>1-i</sup>

5.4.2 Type 4.2: Nombre à virgule fixe avec signe normalisé à ±2<sup>j</sup> x100% = Fi.j

Fi.j := CPi { Uli-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Signe } := Fi.j[1..i] < 2<sup>j</sup>(-1..+1-2<sup>1-i</sup>) >  
 Signe = S := BS1[i]  
 S < 0 > := positif, S < 1 > := négatif  
 S < 1 > et Uli-1 < 0 > := -2<sup>j</sup>  
 S < 1 > et Uli-1 < 1..2<sup>i-1</sup>-1 > := -2<sup>j</sup>(1-2<sup>1-i</sup> .. -2<sup>1-i</sup>)  
 S < 0 > et Uli-1 < 0..2<sup>i-1</sup>-1 > := 2<sup>j</sup>(0 .. +1-2<sup>1-i</sup>)

	i	i-1	i-2	-	-	2	1
S	2 <sup>j-1</sup>	2 <sup>j-2</sup>	-	-	2 <sup>j-i+2</sup>	2 <sup>j-i+1</sup>	

Intervalle: -2<sup>j</sup> .. +2<sup>j</sup> -2<sup>j+1-i</sup>

5.5 Type 5: RÉEL (R) (nombre à virgule flottante)

Nombre à virgule flottante := Ri.j { Fraction, Exposant, Signe } i = taille totale du nombre à virgule flottante  
 Fraction = F := Uij[1..j] < 0..1-2<sup>-j</sup> > j = taille de la fraction  
 Exposant = E := Uli-j-1[j+1..i-1] < 0..2<sup>1-j-1</sup> > i-j-1 = taille de l'exposant  
 Signe = S := BS1[i] S < 0 > := positif  
 S < 1 > := négatif

	i	i-1	i-2	-	-	j+2	j+1		j	j-1	-	-	2	1
Signe	E x p o s a n t						F r a c t i o n							
S	2 <sup>i-j-2</sup>	2 <sup>i-j-3</sup>	-	-	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	-	-	2 <sup>-j-1</sup>	2 <sup>-j</sup>		

5.6 Type 6: CHAÎNE BINAIRE (BS) (assemblage de bits indépendants)

Chaîne binaire := BSi[1..i] = registre d'état de taille i  
 BSi[n] := Sn < 0..1 > = bit d'état en position n

	i	i-1	-	n	-	2	1
Si	Si-1	-	Sn	-	S2	S1	

Sn < 0 > := FAUX  
 Sn < 1 > := VRAI

5.7 Type 7: CHAÎNE D'OCTETS (OS) (assemblage d'octets)

Chaîne d'octets := OS8i[1..8i] < code >

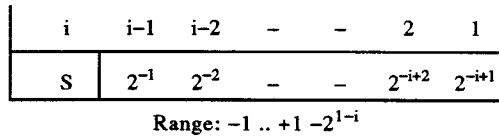
Octets	Bits	8	7	6	5	4	3	2	1
1		Octet 1							
2		Octet 2							
.		.							
.		.							
i		Octet i							

5.4 Type 4: FIXED POINT NUMBER (F) (positive or negative fixed point number)

Negative numbers are presented in two's complement

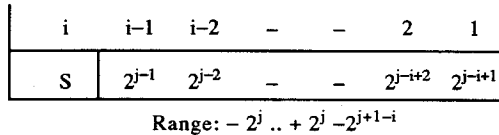
5.4.1 Type 4.1: Normalized signed fixed point number = Fi

Fi := CPi { Uli-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Sign } := Fi[1..i] < -1..+1-2<sup>1-i</sup> >  
 Sign = S := BS1[i]  
 S < 0 > := positive, S < 1 > := negative  
 S < 1 > and Uli-1 < 0 > := -1  
 S < 1 > and Uli-1 < 1..2<sup>i-1</sup>-1 > := -1+2<sup>1-i</sup> .. -2<sup>1-i</sup>  
 S < 0 > and Uli-1 < 0..2<sup>i-1</sup>-1 > := 0 .. 1-2<sup>1-i</sup>



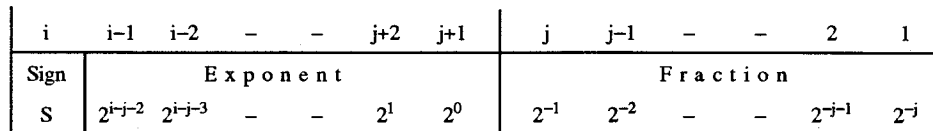
5.4.2 Type 4.2: Signed fixed point number normalized to ±2<sup>j</sup> × 100% = Fi,j

Fi,j := CPi { Uli-1[1..i-1] < 0..2<sup>i-1</sup>-1 >, Sign } := Fi,j[1..i] < 2<sup>j</sup>(-1..+1-2<sup>1-i</sup>) >  
 Sign = S := BS1[i]  
 S < 0 > := positive, S < 1 > := negative  
 S < 1 > and Uli-1 < 0 > := -2<sup>j</sup>  
 S < 1 > and Uli-1 < 1..2<sup>i-1</sup>-1 > := -2<sup>j</sup>(1-2<sup>1-i</sup> .. -2<sup>1-i</sup>)  
 S < 0 > and Uli-1 < 0..2<sup>i-1</sup>-1 > := 2<sup>j</sup>(0 .. +1-2<sup>1-i</sup>)



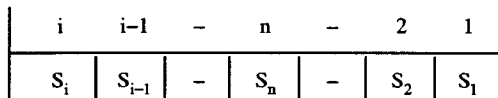
5.5 Type 5: REAL (R) (floating point number)

Floating point number := Ri,j { Fraction, Exponent, Sign }      *i* = total size of floating point number  
 Fraction = F := Uli[1..j] < 0..1-2<sup>-j</sup> >                              *j* = size of fraction  
 Exponent = E := Uli-j-1[j+1..i-1] < 0..2<sup>i-j-1</sup> >              *i*-*j*-1 = size of exponent  
 Sign = S := BS1[i]      S < 0 > := positive  
                                  S < 1 > := negative



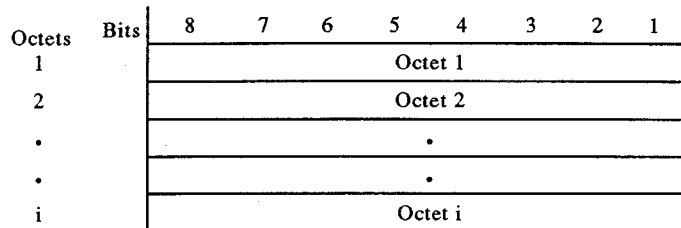
5.6 Type 6: BITSTRING (BS) (assembly independent bits)

Bitstring := BS1[1..i] = status register of size *i*  
 BS1[n] := S<sub>n</sub> < 0..1 > = status bit in position *n*  
  
 S<sub>n</sub> < 0 > := FALSE  
 S<sub>n</sub> < 1 > := TRUE



5.7 Type 7: OCTETSTRING (OS) (assembly of octets)

Octetstring := OS8i[1..8i] < code >



## 6 Ensemble d'éléments d'information

Quelques recommandations sont présentées ci-dessous.

### 6.1 Type 1: ENTIER SANS SIGNE (UI) (nombre entier positif)

#### 6.1.1 Type 1.1: Entier binaire sans signe

- Elément d'information "Commande double" (VEI 371-03-03) = UI2DoubleCommand

UI2DoubleCommand := UI2[1..2] < 0..3 >  
 < 0 > := non autorisé  
 < 1 > := OFF  
 < 2 > := ON  
 < 3 > := non autorisé

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Elément d'information "Commande continue de régulation" (VEI 371-03-12) = UI2PersistRegCommand

UI2PersistRegCommand := UI2[1..2] < 0..3 >  
 < 0 > := non autorisé  
 < 1 > := INFÉRIEUR  
 < 2 > := SUPÉRIEUR  
 < 3 > := non autorisé

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Elément d'information "Commande de régulation par échelons" (VEI 371-03-13) = UI2RegStepCommand

UI2RegStepCommand := UI2[1..2] < 0..3 >  
 < 0 > := non autorisé  
 < 1 > := échelon INFÉRIEUR suivant  
 < 2 > := échelon SUPÉRIEUR suivant  
 < 3 > := non autorisé

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Elément d'information "Information de signalisation double" (VEI 371-02-08) = UI2DoublePointInf

UI2DoublePointInf := UI2[1..2] < 0..3 >  
 < 0 > := état indéterminé  
 < 1 > := état déterminé OFF  
 < 2 > := état déterminé ON  
 < 3 > := état indéterminé

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Elément d'information "Code 1-parmi-8" = UI8\_1of8Code (Un bit sur 8 est égal à 1, les 7 autres bits sont nuls).

UI8\_1of8Code := UI8[1..8] < 1..8 code 1-of-8 >

**Application recommandée:** information sur les positions des échelons.

- Elément d'information "Entier sans signe à 8 bits" = UI8

UI8 := UI8[1..8] < 0..2<sup>8</sup>-1 >

8	7	6	5	4	3	2	1	
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
MSB				Intervalle: 0..255				LSB

- Elément d'information "Valeur dans l'intervalle de 0 à 250" = UI8Range250

UI8Range250 := UI8[1..8] < 0..250 >  
 < 251..255 > := réservé à un usage spécial



## 6 Set of information elements

Some recommendations are given below.

### 6.1 Type 1: UNSIGNED INTEGER (UI) (positive whole number)

#### 6.1.1 Type 1.1: Unsigned binary integer

- Information element "Double Command" (IEV 371-03-03) = UI2DoubleCommand

UI2DoubleCommand := UI2[1..2] < 0..3 >  
 < 0 > := not permitted  
 < 1 > := OFF  
 < 2 > := ON  
 < 3 > := not permitted

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Information element "Persistent Regulating Command" (IEV 371-03-12) = UI2PersistRegCommand

UI2PersistRegCommand := UI2[1..2] < 0..3 >  
 < 0 > := not permitted  
 < 1 > := LOWER  
 < 2 > := HIGHER  
 < 3 > := not permitted

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Information element "Regulating Step Command" (IEV 371-03-13) = UI2RegStepCommand

UI2RegStepCommand := UI2[1..2] < 0..3 >  
 < 0 > := not permitted  
 < 1 > := next step LOWER  
 < 2 > := next step HIGHER  
 < 3 > := not permitted

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Information element "Double-Point Information" (IEV 371-02-08) = UI2DoublePointInf

UI2DoublePointInf := UI2[1..2] < 0..3 >  
 < 0 > := indeterminate state  
 < 1 > := determined state OFF  
 < 2 > := determined state ON  
 < 3 > := indeterminate state

8	7	6	5	4	3	2	1
-	-	-	-	-	-	2 <sup>1</sup>	2 <sup>0</sup>

- Information element "1-of-8 Code" = UI8\_1of8Code (One of 8 bits is 1, the other 7 bits are 0).

UI8\_1of8Code := UI8[1..8] < 1..8 code 1-of-8 >

**Recommended application:** Information on step positions.

- Information element "8 bit Unsigned Integer" = UI8

UI8 := UI8[1..8] < 0..2<sup>8</sup> - 1 >

8	7	6	5	4	3	2	1	
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
MSB				Range: 0..255				LSB

- Information element "Value with Range 0 to 250" = UI8Range250

UI8Range250 := UI8[1..8] < 0..250 >  
 < 251..255 > := reserved for special use

6.1.2 Type 1.2: Entier sans signe codé DCB

- Élément d'information "Entier sans signe codé DCB à 6 chiffres" = 6UI4BCD

6UI4BCD := 6UI4[1..4] < 0..9 BCD >  
 := UI24[1..24] < 0..999 999 BCD >

8	7	6	5	4	3	2	1
Chiffre 10				Chiffre 1			
8	4	2	1	8	4	2	1
Chiffre 1 000				Chiffre 100			
8	4	2	1	8	4	2	1
Chiffre 100 000				Chiffre 10 000			
8	4	2	1	8	4	2	1

Intervalle: 0 .. 999 999

6.2 Type 2: ENTIER (I) (nombre entier positif ou négatif)

6.2.1 Type 2.1: Entier binaire avec signe

- Élément d'information "Entier à 8 bits" = I8

I8 := I8[1..8] < -2<sup>7</sup>..+2<sup>7</sup>-1 >

8	7	6	5	4	3	2	1	
S	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
MSB				Intervalle: -128 .. +127				LSB

- Élément d'information "Entier à 12 bits aligné à droite" = I12Right

I12Right := I12[1..12] < -2<sup>11</sup>..+2<sup>11</sup>-1 >

8	7	6	5	4	3	2	1
2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
-	-	-	-	S	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>

- Élément d'information "Entier à 12 bits aligné à gauche" = I12Left

I12Left := I12[5..16] < -2<sup>11</sup>..+2<sup>11</sup>-1 >

8	7	6	5	4	3	2	1
2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	-	-	-	-
S	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>

Intervalle: -2048 .. +2047

Applications recommandées: valeurs mesurées bipolaires | commandes de valeur de consigne bipolaire.

6.2.2 Type 2.2: Entier avec signe codé DCB

- Élément d'information "Entier codé DCB à 5 chiffres" = I21BCD

I21BCD := I21[1..21] < -99 999..+99 999 BCD >  
 := CP { 5UI4BCD, Signe }  
 UI4BCD := UI4[1..4] < 0..9 BCD >  
 Signe = S := BS1[21]  
 S < 0 > := positif  
 S < 1 > := négatif

8	7	6	5	4	3	2	1	
Chiffre 10				Chiffre 1				
8	4	2	1	8	4	2	1	
Chiffre 1 000				Chiffre 100				
8	4	2	1	8	4	2	1	
				S	Chiffre 10 000			
					8	4	2	1

Intervalle: -99 999 .. +99 999

6.1.2 Type 1.2: Unsigned binary coded decimal integer

- Information element "6 digit Unsigned BCD Integer" = 6UI4BCD

6UI4BCD := 6UI4[1..4] < 0.9 BCD >  
 := UI24[1..24] < 0.999 999 BCD >

8	7	6	5	4	3	2	1
Digit 10				Digit 1			
8	4	2	1	8	4	2	1
Digit 1 000				Digit 100			
8	4	2	1	8	4	2	1
Digit 100 000				Digit 10 000			
8	4	2	1	8	4	2	1

Range: 0 .. 999 999

6.2 Type 2: INTEGER (I) (negative or positive whole number)

6.2.1 Type 2.1: Signed binary integer

- Information element "8 bit Integer" = I8

I8 := I8[1..8] <  $-2^7 \dots +2^7 - 1$  >

8	7	6	5	4	3	2	1	
S	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
MSB				Range: -128 .. +127				LSB

- Information element "12 bit Integer Rightjustified" = I12Right

I12Right := I12[1..12] <  $-2^{11} \dots +2^{11} - 1$  >

8	7	6	5	4	3	2	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
-	-	-	-	S	$2^{10}$	$2^9$	$2^8$

- Information element "12 bit Integer Leftjustified" = I12Left

I12Left := I12[5..16] <  $-2^{11} \dots +2^{11} - 1$  >

8	7	6	5	4	3	2	1
$2^3$	$2^2$	$2^1$	$2^0$	-	-	-	-
S	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$

Range: -2048 .. +2047

Recommended applications: bipolar measured values | bipolar set point commands.

6.2.2 Type 2.2: Signed binary coded decimal integer

- Information element "5 digit BCD Integer" = I21BCD

I21BCD := I21[1..21] < -99 999 .. +99 999 BCD >  
 := CP { 5UI4BCD, Sign }  
 UI4BCD := UI4[1..4] < 0.9 BCD >  
 Sign = S := BS1[21]  
 S < 0 > := positive  
 S < 1 > := negative

8	7	6	5	4	3	2	1	
Digit 10				Digit 1				
8	4	2	1	8	4	2	1	
Digit 1 000				Digit 100				
8	4	2	1	8	4	2	1	
				Digit 10 000				
				S	8	4	2	1

Range: -99 999 .. +99 999

6.3 Type 3: NOMBRE À VIRGULE FIXE SANS SIGNE (UF) (nombre à virgule fixe positif)

6.3.1 Type 3.1: Nombre normalisé à virgule fixe sans signe

- Élément d'information "Nombre normalisé à virgule fixe sans signe à 8 bits" = UF8

UF8 := UF8[1..8] < 0..1-2<sup>-8</sup> >

8	7	6	5	4	3	2	1
2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	2 <sup>-8</sup>
MSB		Intervalle: 0 .. +1 - 1/256				LSB	

6.3.2 Type 3.2: Nombre à virgule fixe sans signe normalisé à +2<sup>j</sup> × 100%

- Élément d'information "Nombre à virgule fixe sans signe à 8 bits normalisé à 200%" = UF8.1

UF8.1 := UF8[1..8] < 0..+2-2<sup>-7</sup> >

8	7	6	5	4	3	2	1
2 <sup>-0</sup>	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>
MSB		Intervalle: 0 .. +2 - 1/128				LSB	

6.4 Type 4: NOMBRE À VIRGULE FIXE (F) (nombre à virgule fixe positif ou négatif)

6.4.1 Type 4.1: Nombre normalisé à virgule fixe avec signe

- Élément d'information "Nombre normalisé à virgule fixe avec signe à 16 bits" = F16

F16 := F16[1..16] < -1..+1-2<sup>-15</sup> >

								LSB
8	7	6	5	4	3	2	1	
2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>	2 <sup>-12</sup>	2 <sup>-13</sup>	2 <sup>-14</sup>	2 <sup>-15</sup>	
S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	
MSB		Intervalle: -1 .. +1 - 2 <sup>-15</sup>						

- Élément d'information "Nombre normalisé à virgule fixe avec signe à 12 bits aligné à droite" = F12Right

F12Right := F12[1..12] < -1..+1-2<sup>-11</sup> >

8	7	6	5	4	3	2	1
2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>
-	-	-	-	S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>

- Élément d'information "Nombre normalisé à virgule fixe avec signe à 12 bits aligné à gauche" = F12Left

F12Left := F12[5..16] < -1..+1-2<sup>-11</sup> >

8	7	6	5	4	3	2	1
2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>	-	-	-	-
S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>

Applications Recommandées: valeurs mesurées<sup>1)</sup> commandes<sup>1)</sup> de valeur de consigne (VEI 371-03-11).

1) Valeur nominale mesurée recommandée N = E/P avec E = limite de l'intervalle et P = 1,2 (par défaut); autres spécifications de P selon accord.

### 6.3 Type 3: UNSIGNED FIXED POINT NUMBER (UF) (positive fixed point number)

#### 6.3.1 Type 3.1: Normalized unsigned fixed point number

- Information element "8 bit Normalized Unsigned Fixed Point Number" = UF8

$$UF8 := UF8[1..8] < 0..1-2^{-8} >$$

8	7	6	5	4	3	2	1
$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$
MSB		Range: 0 .. +1 - 1/256				LSB	

#### 6.3.2 Type 3.2: Unsigned fixed point number normalized to $+2^j \times 100\%$

- Information element "Unsigned 8 bit Fixed Point Number Normalized to 200%" = UF8.1

$$UF8.1 := UF8[1..8] < 0..+2-2^{-7} >$$

8	7	6	5	4	3	2	1
$2^{-0}$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$
MSB		Range: 0 .. +2 - 1/128				LSB	

### 6.4 Type 4: FIXED POINT NUMBER (F) (positive or negative fixed point number)

#### 6.4.1 Type 4.1: Normalized signed fixed point number

- Information element "16 bit Normalized Signed Fixed Point Number" = F16

$$F16 := F16[1..16] < -1..+1-2^{-15} >$$

8	7	6	5	4	3	2	1	LSB 
$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$	$2^{-12}$	$2^{-13}$	$2^{-14}$	$2^{-15}$	
S	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	
MSB		Range: -1 .. +1 - 2 <sup>-15</sup>						

- Information element "12 bit Normalized Signed Fixed Point Number Rightjustified" = F12Right

$$F12Right := F12[1..12] < -1..+1-2^{-11} >$$

8	7	6	5	4	3	2	1
$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$	$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$
-	-	-	-	S	$2^{-1}$	$2^{-2}$	$2^{-3}$

- Information element "12 bit Normalized Signed Fixed Point Number Leftjustified" = F12Left

$$F12Left := F12[5..16] < -1..+1-2^{-11} >$$

8	7	6	5	4	3	2	1
$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$	-	-	-	-
S	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$

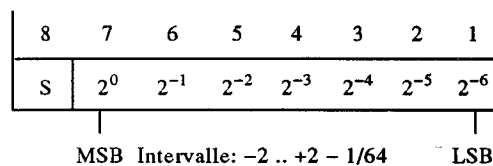
**Recommended applications:** measured values<sup>1)</sup> set point commands<sup>1)</sup> (IEV 371-03-11).

<sup>1)</sup> Recommended nominal measured value  $N = E/P$ , with  $E$  = limit of range and  $P = 1,2$  (default); other specifications of  $P$  by agreement.

6.4.2 Type 4.2: nombre à virgule fixe avec signe normalisé à  $\pm 2^j \times 100\%$

- Élément d'information "Nombre à virgule fixe avec signe à 8 bits normalisé à  $\pm 200\%$ " = F8.1

F8.1 := F8[1..8] < 2(-1..+1-2<sup>-7</sup>) >



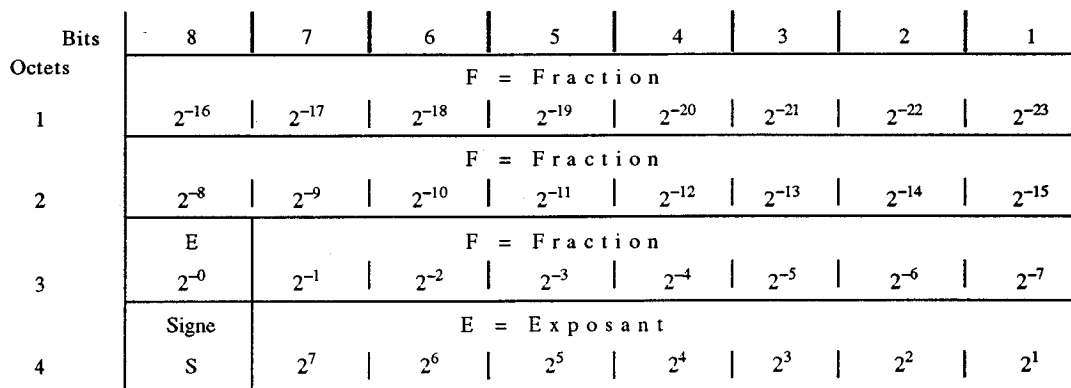
6.5 Type 5: RÉEL (R) (nombre à virgule flottante)

- Élément d'information "Nombre court à virgule flottante IEEE STD 754" = R32IEEESTD 754

R32IEEESTD754:= R32.23 { Fraction, Exposant, Signe }

Fraction = F := UI23[1..23] < 0..1-2<sup>-23</sup> >  
 Exposant = E := UI8[24..31] < 0..255 >  
 Signe = S := BS1[32] S < 0 >:= positif  
 S < 1 >:= négatif

F < 0 > et E < 0 > := (-1)<sup>S</sup> x 0 = ± zéro  
 F < ≠ 0 > et E < 0 > := (-1)<sup>S</sup> x 2<sup>E-126</sup>(0,F) = nombres dénormalisés  
 E < 1..254 > := (-1)<sup>S</sup> x 2<sup>E-127</sup>(1,F) = nombres normalisés  
 F < 0 > et E < 255 > := (-1)<sup>S</sup> x ∞ = ± infini  
 F < ≠ 0 > et E < 255 > := NaN = non-nombre, quel que soit S



Ce paragraphe est en accord avec la Norme IEEE 754 et spécifie les gammes suivantes pour l'arithmétique à virgule flottante:

Intervalle: -2<sup>128</sup> + 2<sup>104</sup> ... + 2<sup>128</sup> - 2<sup>104</sup>, c'est-à-dire: -3,4 x 10<sup>38</sup> ... +3,4 x 10<sup>38</sup>

Nombre négatif le plus faible: -2<sup>-149</sup>, c'est-à-dire: -1,4 x 10<sup>-45</sup>

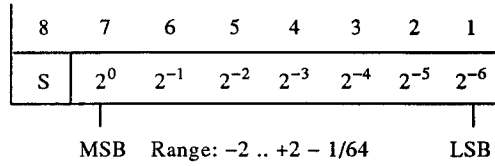
Nombre positif le plus faible: +2<sup>-149</sup>, c'est-à-dire: +1,4 x 10<sup>-45</sup>

Application recommandée: nombre à virgule flottante.

6.4.2 Type 4.2: Signed fixed point number normalized to  $\pm 2^j \times 100\%$

- Information element "8 bit Signed Fixed Point Number Normalized to  $\pm 200\%$ " = F8.1

F8.1 := F8[1..8] <  $2(-1..+1-2^{-7})$  >



6.5 Type 5: REAL (R) (floating point number)

- Information element "Short Floating Point Number IEEE STD 754" = R32IEEESTD 754

R32IEEESTD754:= R32.23 { Fraction, Exponent, Sign }

Fraction = F:= UI23[1..23] <  $0..1-2^{-23}$  >  
 Exponent = E:= UI8[24..31] <  $0..255$  >  
 Sign = S:= BS1[32] S < 0 >:= positive  
 S < 1 >:= negative

F < 0 > and E < 0 > :=  $(-1)^S \times 0$  =  $\pm$  zero  
 F <  $\neq 0$  > and E < 0 > :=  $(-1)^S \times 2^{E-126}(0,F)$  = denormalized numbers  
 E <  $1..254$  > :=  $(-1)^S \times 2^{E-127}(1,F)$  = normalized numbers  
 F < 0 > and E <  $255$  > :=  $(-1)^S \times \infty$  =  $\pm$  infinite  
 F <  $\neq 0$  > and E <  $255$  > := NaN = not a number, regardless of S

	8	7	6	5	4	3	2	1
Bits								
Octets	F = Fraction							
1	$2^{-16}$	$2^{-17}$	$2^{-18}$	$2^{-19}$	$2^{-20}$	$2^{-21}$	$2^{-22}$	$2^{-23}$
2	F = Fraction							
	$2^{-8}$	$2^{-9}$	$2^{-10}$	$2^{-11}$	$2^{-12}$	$2^{-13}$	$2^{-14}$	$2^{-15}$
3	E	F = Fraction						
	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$	$2^{-5}$	$2^{-6}$	$2^{-7}$
4	Sign	E = Exponent						
	S	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$

This subclause is in line with IEEE Standard 754 and specifies the following ranges for floating point arithmetics:

Range:  $-2^{128} + 2^{104} \dots + 2^{128} - 2^{104}$ , that is:  $-3,4 \times 10^{38} \dots +3,4 \times 10^{38}$   
 Smallest negative number:  $-2^{-149}$ , that is:  $-1,4 \times 10^{-45}$   
 Smallest positive number:  $+2^{-149}$ , that is:  $+1,4 \times 10^{-45}$

Recommended application: floating point number.

### 6.6 Type 6: CHAÎNE BINAIRE (BS) (assemblage de bits indépendants)

- Élément d'information "Commande simple" (VEI 371-03-02) = BS1Command

BS1Command =	SC	:= BS1[1]	8	7	6	5	4	3	2	1
	SC < 0 >	:= OFF	-	-	-	-	-	-	-	SC
	SC < 1 >	:= ON								

- Élément d'information "Information de signalisation simple" (VEI 371-02-07) = BS1Info

BS1Info =	SPI	:= BS1[1]	8	7	6	5	4	3	2	1
	SPI < 0 >	:= OFF	-	-	-	-	-	-	-	SPI
	SPI < 1 >	:= ON								

- Élément d'information "Registre d'état à 8 bits" = BS8Status

BS8Status :=	BS8[1..8]		8	7	6	5	4	3	2	1
	BS8[n] :=	Sn	S8	S7	S6	S5	S4	S3	S2	S1
	1 ≤ n ≤ 8	Sn < 0 > := OFF								
		Sn < 1 > := ON								

- Élément d'information "Etat à 8 bits + Détection de transitoire" = BS8Stat+Trans

4 paires de bits d'état et de bits de détection de transitoire

BS8Stat+Trans :=	4BS2[1..2]		8	7	6	5	4	3	2	1
	BS2[1] :=	ST = état	TR4	ST4	TR3	ST3	TR2	ST2	TR1	ST1
		ST < 0 > := OFF								
		ST < 1 > := ON								
	BS2[2] :=	TR = détection de transitoire								
		TR < 0 > := aucun transitoire n'a été détecté depuis la dernière transmission								
		TR < 1 > := deux transitions ou plus ont été détectées depuis la dernière transmission								

**Application recommandée:** Information de signalisation simple avec détection de transitoire

- Élément d'information "Etat à 16 bits + Détection de changement d'état" = BS16Stat+Change

8 bits d'état avec 8 bits de détection de changement d'état

BS16Stat+Change :=	BS16[1..16]		8	7	6	5	4	3	2	1
	BS16[n] :=	STn = bit d'état en position de bit n	ST8	ST7	ST6	ST5	ST4	ST3	ST2	ST1
	1 ≤ n ≤ 8	STn < 0 > := OFF	CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1
		STn < 1 > := ON								
	BS16[n+8] :=	CDn = bit de détection de changement d'état en position de bit n+8								
		CDn < 0 > := aucun changement d'état n'a été détecté depuis la dernière transmission								
		CDn < 1 > := au moins un changement d'état a été détecté depuis la dernière transmission								

**Application recommandée:** 8 informations de signalisation simples avec détection de changement d'état.

### 6.7 Type 7: CHAÎNE D'OCTETS (OS) (assemblage d'octets)

- Élément d'information "Chaîne de caractères ASCII" = OS8iASCII  
chaîne de caractères à 8 bits à partir de l'ensemble étendu de caractères ASCII

OS8iASCII := OS8i[1..8i] < ASCII 8 bit code >

**Application recommandée:** information textuelle.



6.6 Type 6: *BITSTRING (BS)* (assembly of independent bits)

- Information element "Single Command" (IEV 371-03-02) = BS1Command

BS1Command = SC := BS1[1]  
 SC < 0 > := OFF  
 SC < 1 > := ON

8	7	6	5	4	3	2	1
-	-	-	-	-	-	-	SC

- Information element "Single-Point Information" (IEV 371-02-07) = BS1Info

BS1Info = SPI := BS1[1]  
 SPI < 0 > := OFF  
 SPI < 1 > := ON

8	7	6	5	4	3	2	1
-	-	-	-	-	-	-	SPI

- Information element "8 bit Status Register" = BS8Status

BS8Status := BS8[1..8]  
 BS8[n] := Sn Sn < 0 > := OFF  
 1 ≤ n ≤ 8 Sn < 1 > := ON

8	7	6	5	4	3	2	1
S8	S7	S6	S5	S4	S3	S2	S1

- Information element "8 bit Status+Transient Detection" = BS8Stat+Trans  
 4 pairs of status- and transient detection bits

BS8Stat+Trans := 4BS2[1..2]  
 BS2[1] := ST = status  
 ST < 0 > := OFF  
 ST < 1 > := ON  
 BS2[2] := TR = transient detection  
 TR < 0 > := no transient was detected since last transmission.  
 TR < 1 > := two or more transitions were detected since last transmission.

8	7	6	5	4	3	2	1
TR4	ST4	TR3	ST3	TR2	ST2	TR1	ST1

**Recommended application:** single-point information with transient detection.

- Information element "16 bit Status + Status Change Detection" = BS16Stat+Change  
 8 status bits with 8 status change detection bits

BS16Stat+Change := BS16[1..16]  
 BS16[n] := STn = status bit in bit position n  
 1 ≤ n ≤ 8 STn < 0 > := OFF  
 STn < 1 > := ON  
 BS16[n+8] := CDn = status change detection bit in bit position n+8  
 CDn < 0 > := no status change was detected since last transmission  
 CDn < 1 > := at least one status change was detected since last transmission

8	7	6	5	4	3	2	1
ST8	ST7	ST6	ST5	ST4	ST3	ST2	ST1
CD8	CD7	CD6	CD5	CD4	CD3	CD2	CD1

**Recommended application:** 8 single-point information with status change detection.

6.7 Type 7: *OCTETSTRING (OS)* (assembly of octets)

- Information element "ASCII Character String" = OS8iASCII  
 string of 8 bit characters from the extended ASCII character set

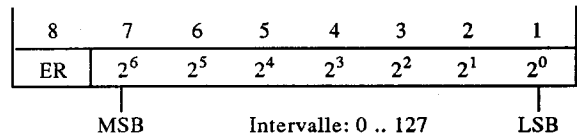
OS8iASCII := OS8i[1..8i] < ASCII 8 bit code >

**Recommended application:** text information.

### 6.8 ELÉMENTS D'INFORMATION COMPOSITE (CP) (Séquences de champs de données)

- Elément d'information "Valeur avec indication d'erreur" = CP8Valeur+ Erreur

CP8Valeur+Erreur:= CP8 { Valeur, Erreur }  
 Valeur:= UI7[1..7] < 0..127 >  
 Erreur = ER:= BS1[8] ER < 0 > := pas d'erreur  
 ER < 1 > := erreur

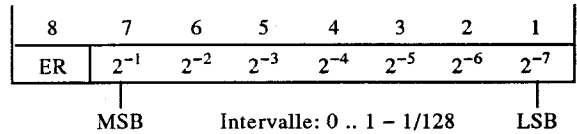


- Elément d'information "Valeur dans l'intervalle de 0 à 120 et indication d'erreur" = CP8ValRange120+Er

CP8ValRange120+Er:= CP8 { Valeur dans l'intervalle de 0 à 120, Erreur }  
 Valeur dans l'intervalle de 0 .. 120:= UI7[1..7] < 0..120 >  
 <121..127 > := réservé à un usage spécial  
 Erreur = ER:= BS1[8] ER < 0 > := pas d'erreur  
 ER < 1 > := erreur

- Elément d'information "Valeur normalisée avec indication d'erreur" = CP8NormVal+Er

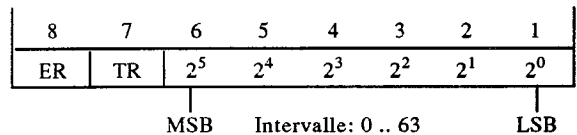
CP8NormVal+Er := CP8 { Valeur normalisée, Erreur }  
 Valeur normalisée := UF7 [1..7] < 0..1-2<sup>-7</sup> >  
 Erreur = ER:= BS1[8] ER < 0 > := pas d'erreur  
 ER < 1 > := erreur



**Applications recommandées:** valeurs mesurées | commandes de valeur de consigne (VEI 371-03-11)

- Elément d'information "Valeur avec indication de transitoire et d'erreur" = CP8NormVal+Tr +Er

CP8NormVal+Tr+Er := CP8 { Valeur, Transitoire, Erreur }  
 Valeur := UI6[1..6] < 0..63 >  
 Transitoire = TR := BS2[7]  
 TR < 0 > := le matériel n'est pas en état transitoire  
 TR < 1 > := le matériel est en état transitoire  
 Erreur = ER := BS2[8]  
 ER < 0 > := pas d'erreur  
 ER < 1 > := erreur

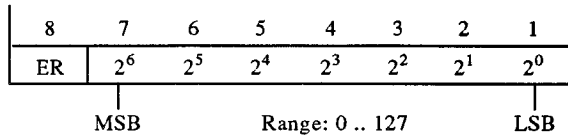


**Applications recommandées:** information de position par échelons | information de position de prise de transformateurs.

6.8. COMPOUND INFORMATION ELEMENTS (CP) (Sequences of data fields)

- Information element "Value with Error Indication" = CP8Value+ Error

CP8Value+Error:= CP8 { Value, Error }  
 Value := UI7[1..7] < 0..127 >  
 Error = ER:= BS1[8] ER < 0 > := no error  
 ER < 1 > := error

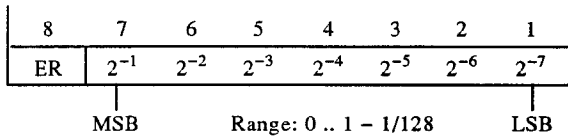


- Information element "Value with Range 120 and Error Indication" = CP8ValRange120+Er

CP8ValRange120+Er:= CP8 { Value with range 120, Error }  
 Value with range 0 .. 120:= UI7[1..7] < 0..120 >  
 <121..127 > := reserved for special use  
 Error = ER:= BS1[8] ER < 0 > := no error  
 ER < 1 > := error

- Information element "Normalised Value with Error Indication" = CP8NormVal+Er

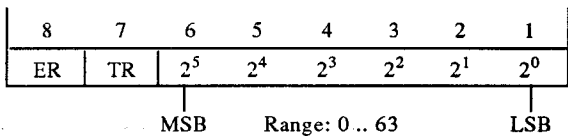
CP8NormVal+Er := CP8 { Normalized value, Error }  
 Normalized value := UF7[1..7] < 0..1-2<sup>-7</sup> >  
 Error = ER:= BS1[8] ER < 0 > := no error  
 ER < 1 > := error



**Recommended applications:** measured values | set point commands (IEV 371-03-11)

- Information element " Value with Transient and Error Indication" = CP8NormVal+Tr +Er

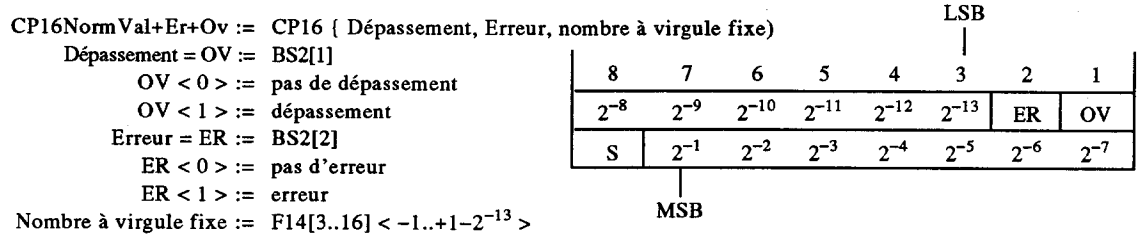
CP8NormVal+Tr+Er := CP8 { Value, Transient, Error }  
 Value := UI6[1..6] < 0..63 >  
 Transient = TR := BS2[7]  
 TR < 0 > := equipment is not in transient state  
 TR < 1 > := equipment is in transient state  
 Error = ER := BS2[8]  
 ER < 0 > := no error  
 ER < 1 > := error



**Recommended applications:** step position information | tap position information of transformers.

- Elément d'information "valeur normalisée à virgule fixe à 14 bits avec indication de dépassement et d'erreur" = CP16NormVal+Er+Ov

Valeur avec signe dans une notation normalisée à virgule fixe de taille 14 combinée avec indication de dépassement et d'erreur:



Unité de LSB = 2<sup>-13</sup> (par défaut); autres spécifications selon accord.

NOTE - Si la résolution de la valeur mesurée est plus approximative que (inférieure à) l'unité de poids la plus faible (LSB), les bits les moins significatifs sont alors mis à zéro.

Intervalle de valeurs si l'unité du LSB = 2<sup>-13</sup>

S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>	2 <sup>-12</sup>	2 <sup>-13</sup>	ER	OV	:= valeur
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	:= >1-2 <sup>-13</sup>
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	:= 1-2 <sup>-13</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	:= 2x2 <sup>-13</sup>
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	:= 1x2 <sup>-13</sup>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	:= 0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	:= -1x2 <sup>-13</sup>
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	:= -2x2 <sup>-13</sup>
1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	:= -3x2 <sup>-13</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	:= -1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	:= <-1

Application recommandée: valeurs mesurées bipolaires | commandes de valeur de consigne bipolaires.

Valeur nominale mesurée recommandée N = E/P avec E = limite de l'intervalle et P = 1,2 (par défaut); autres spécifications de P selon accord.

- Information element "14 bit normalized Fixed Point Value with Error and Overflow Indication" = CP16NormVal+Er+Ov

Signed value in normalized fixed point notation of size 14 combined with error and overflow indication:

CP16NormVal+Er+Ov := CP16 { Overflow, Error, fixed point number }

Overflow = OV := BS2[1]  
 OV < 0 > := no overflow  
 OV < 1 > := overflow

Error = ER := BS2[2]  
 ER < 0 > := no error  
 ER < 1 > := error

Fixed point number := F14[3..16] < -1..+1-2<sup>-13</sup> >

Unit of LSB = 2<sup>-13</sup> (default); other specifications by agreement.

								LSB									
8	7	6	5	4	3	2	1										
2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>	2 <sup>-12</sup>	2 <sup>-13</sup>	ER	OV										
S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>										
								MSB									

NOTE - If the resolution of the measured value is coarser (lower) than the unit of the LSB, then the least significant bits are set to zero.

Range of values if unit of LSB = 2<sup>-13</sup>

S	2 <sup>-1</sup>	2 <sup>-2</sup>	2 <sup>-3</sup>	2 <sup>-4</sup>	2 <sup>-5</sup>	2 <sup>-6</sup>	2 <sup>-7</sup>	2 <sup>-8</sup>	2 <sup>-9</sup>	2 <sup>-10</sup>	2 <sup>-11</sup>	2 <sup>-12</sup>	2 <sup>-13</sup>	ER	OV	:= value
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	:= >1-2 <sup>-13</sup>
0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	:= 1-2 <sup>-13</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	:= 2x2 <sup>-13</sup>
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	:= 1x2 <sup>-13</sup>
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	:= 0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	:= -1x2 <sup>-13</sup>
1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	:= -2x2 <sup>-13</sup>
1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0	:= -3x2 <sup>-13</sup>
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	:= -1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	:= <-1

Recommended application: bipolar measured values | bipolar set point commands.

Recommended nominal measured value N = E/P with E = limit of range and P = 1,2 (default); other specifications of P by agreements.

- Elément d'information "Valeur avec octet de signe" = CP8(n+1)Val+Sign jusqu'à 8 octets d'ENTIERS SANS SIGNE, plus un octet de signe

$CP8(n+1)Val+Signs := CP8(n+1) \{ Valeurs, Octet\ de\ signe \}$   
 $1 \leq n \leq 8,$  Valeurs :=  $nUI8[1..8] < 0..255 >$   
Octet de signe :=  $BS8[8n+1..8n+n]$   
 $BS8[8n+n] := S_n$   
 $S_n < 0 > := positif$   
 $S_n < 1 > := négatif$

Octets	Bits	8	7	6	5	4	3	2	1
		Valeur 1							
1		$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
2		Valeur 2							
3		Valeur 3							
.		.							
.		.							
8		Valeur 8							
9		S8	S7	S6	S5	S4	S3	S2	S1

Application recommandée: fichiers comportant jusqu'à huit valeurs mesurées, plus un octet de signe.

- Elément d'information "Objet d'information avec descripteur de qualité" = CP8(i+1)Info+Qual

$CP8(i+1)Info+Qual := CP8(i+1) \{ Objet\ d'information, Descripteur\ de\ qualité \}$   
Objet d'information :=  $BS8i[1..8i]$   
Descripteur de qualité :=  $BS8[8i+1..8i+8]$   
 $BS8[8i+1] := Dépassement = OV$   $OV < 0 > := pas\ de\ dépassement$   
 $OV < 1 > := dépassement$

Dépassement: La valeur de l'objet d'information est en dehors du domaine de variation prédéfini (applicable surtout aux grandeurs analogiques).

$BS8[8i+2 .. 8i+4] < 0 > := Réserve = RES$   
 $BS8[8i+5] := Bloqué = BL$   $BL < 0 > := non\ bloqué$   
 $BL < 1 > := bloqué$

Bloqué: La valeur de l'objet d'information est bloquée pour la transmission. La valeur conserve son état précédant le blocage. Blocage et déblocage peuvent être déclenchés par un verrou local ou automatiquement.

$BS8[8i+6] := Substitué = SB$   $SB < 0 > := non\ substitué$   
 $SB < 1 > := substitué$

Substitué: La valeur de l'objet d'information provient d'une rentrée manuelle de l'opérateur (dispatcher) ou d'une source automatique.

$BS8[8i+7] := Non\ actuel = NT$   $NT < 0 > := actuel$   
 $NT < 1 > := non\ actuel$

Non actuel: La valeur de l'objet d'information est actuelle si sa mise à jour a eu lieu récemment. La valeur n'est pas actuelle si sa mise à jour n'a pas eu lieu pendant une durée prédéfinie ou si cette valeur n'est pas disponible.

$BS8[8i+8] := Non\ valide = IV$   $IV < 0 > := valide$   
 $IV < 1 > := non\ valide$

Non valide: Une valeur est valide si elle est acquise normalement. Lorsque la fonction d'acquisition reconnaît des conditions anormales de la source d'information (manquante ou non opérationnelle), la valeur est marquée non valide. La valeur de l'objet d'information n'est pas définie dans ce cas. La marque non valide est utilisée pour indiquer au destinataire que la valeur est probablement incorrecte et qu'elle ne doit pas être utilisée.

Objet d'information					
IV	NT	SB	BL	RES	OV

Descripteur de qualité

Application recommandée: objet d'information avec description de qualité.



- Elément d'information "**Lecture de compteur binaire 1**" = CP8(n+1)CountRead1  
Lecture de compteur (ENTIER SANS SIGNE jusqu'à 4 octets) avec trois indications d'état et un nombre séquentiel à 5 bits.

CP8(n+1)CountRead1 := CP8(n+1) { Lecture de compteur, Notation séquentielle }  
1 ≤ n ≤ 4

Lecture de compteur := UI8n[1..8n] < 0..2<sup>8n</sup>-1 >

Notation séquentielle := CP8{ Nombre séquentiel, Report, Réglage compteur, Non valable }

Nombre séquentiel = SQ := UI5[8n+1..8n+5] < 0..31 >

Report = CY := BS3[8n+6] CY < 0 > := aucun dépassement entier depuis la dernière lecture

CY < 1 > := dépassement entier depuis la dernière lecture

Réglage compteur = CA := BS3 [8n+7] CA < 0 > := compteur non réglé depuis la dernière lecture

CA < 1 > := compteur réglé depuis la dernière lecture

Non valable = IV := BS3 [8n+8] IV < 0 > := lecture du compteur valable

IV < 1 > := lecture du compteur non valable

Lecture de compteur							
IV	CA	CY	Nombre séquentiel				
			2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
							Notation séquentielle

- Elément d'information "**Lecture de compteur binaire bipolaire 1**" = CP8(n+1)BipolCountRead1  
voir Elément d'information "**Lecture de compteur binaire 1**", à ceci près que Lecture de compteur est un ENTIER AVEC SIGNE:

CP8(n+1)BipolCountRead1:= CP8(n+1) { Lecture de compteur bipolaire, Notation séquentielle }  
1 ≤ n ≤ 4

Lecture de compteur bipolaire:= I8n[1..8n] < -2<sup>8n-1</sup> ..+2<sup>8n-1</sup> -1 >

- Elément d'information "**Lecture de compteur codé DCB 1**" = CP8(n+1)BCDCCountRead1  
voir élément d'information "**Lecture de compteur binaire 1**", à ceci près que Lecture de compteur utilise une notation codée en DCB:

CP8(n+1)BCDCCountRead1:= CP8(n+1) { Lecture de compteur DCB, Notation séquentielle }  
1 ≤ n ≤ 4

Lecture de compteur DCB := 2nUI4[1..4] < 0..9 BCD >

Taille de Lecture de compteur n[octets]	Intervalle des valeurs de lecture de compteur		
	Lecture de compteur binaire bipolaire 1 -2 <sup>8n-1</sup> ... +2 <sup>8n-1</sup> - 1	Lecture de compteur binaire 1 0 ... 2 <sup>8n</sup> - 1	Lecture de compteur codé DCB 1 0 ... 10 <sup>2n</sup> - 1
4	-2 147 483 648 ... +2 147 483 647	0 ... 4 294 967 295	0 ... 99 999 999
3	-8 388 608 ... +8 388 607	0 ... 16 777 215	0 ... 999 999
2	-32 768 ... +32 767	0 ... 65 535	0 ... 9 999
1	-128 ... +127	0 ... 255	0 ... 99

**Application recommandée:** télécomptage, transmission de valeurs intégrées (VEI 371-01-05).



- Information element "**Binary Counter Reading 1**" = CP8(n+1)CountRead1  
Counter reading (UNSIGNED INTEGER up to 4 octets) with three status indications and a 5 bit sequence number.

CP8(n+1)CountRead1 := CP8(n+1) { Counter reading, Sequence notation }  
1 ≤ n ≤ 4

Counter reading := UI8n[1..8n] < 0..2<sup>8n</sup>-1 >

Sequence notation := CP8{ Sequence number, Carry, Counter was adjusted, Invalid }

Sequence number = SQ := UI5[8n+1..8n+5] < 0..31 >

Carry = CY := BS3[8n+6] CY < 0 > := no integer overflow occurred since last reading

CY < 1 > := integer overflow occurred since last reading

Counter was adjusted = CA := BS3 [8n+7] CA < 0 > := counter was not adjusted since last reading

CA < 1 > := counter was adjusted since last reading

Invalid = IV := BS3 [8n+8] IV < 0 > := counter reading is valid

IV < 1 > := counter reading is invalid

Counter reading						
IV	CA	CY	Sequence number			
			2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>

Sequence notation

- Information element "**Bipolar Binary Counter Reading 1**" = CP8(n+1)BipolCountRead1  
see information element "**Binary Counter Reading 1**", however Counter reading is a SIGNED INTEGER:

CP8(n+1)BipolCountRead1 := CP8(n+1) { Bipolar counter reading, Sequence notation }  
1 ≤ n ≤ 4

Bipolar counter reading := I8n[1..8n] < -2<sup>8n-1</sup> .. +2<sup>8n-1</sup> - 1 >

- Information element "**BCD Counter Reading 1**" = CP8(n+1)BCDCountRead1  
see information element "**Binary Counter Reading 1**", however Counter reading uses binary coded decimal (BCD) notation:

CP8(n+1)BCDCountRead1 := CP8(n+1) { BCD counter reading, Sequence notation }  
1 ≤ n ≤ 4

BCD Counter reading := 2nUI4[1..4] < 0..9 BCD >

Size of Counter reading n[octets]	Range of counter reading		
	Bipolar binary Counter reading 1 -2 <sup>8n-1</sup> ... +2 <sup>8n-1</sup> - 1	Binary Counter reading 1 0 ... 2 <sup>8n</sup> - 1	BCD Counter reading 1 0 ... 10 <sup>2n</sup> - 1
4	-2 147 483 648 ... +2 147 483 647	0 ... 4 294 967 295	0 ... 99 999 999
3	-8 388 608 ... +8 388 607	0 ... 16 777 215	0 ... 999 999
2	-32 768 ... +32 767	0 ... 65 535	0 ... 9 999
1	-128 ... +127	0 ... 255	0 ... 99

**Recommended application:** telecounting, transmission of integrated totals (IEV 371-01-05)

- Élément d'information "Temps codé DCB 1" = CP48Time1BCD

CP48Time1BCD := CP48 { Millisecondes, Secondes, 10 Secondes, Minutes, 10 Minutes, Heures, 10 Heures, Jours, 10 Jours, 100 Jours }  
 Millisecondes := UI10[1..10] < 0..999 >  
 Secondes := UI4[13..16] < 0..9 BCD >  
 10 Secondes := UI3[17..19] < 0..5 >  
 Minutes := UI4[21..24] < 0..9 BCD >  
 10 Minutes := UI3[25..27] < 0..5 >  
 Heures := UI4[29..32] < 0..9 BCD >  
 10 Heures := UI2[33..34] < 0..2 >  
 Jours := UI4[37..40] < 0..9 BCD >  
 10 Jours := UI4[41..44] < 0..9 BCD >  
 100 Jours := UI4[45..48] < 0..9 BCD >

Octets	Bits	8	7	6	5	4	3	2	1
1		Millisecondes							
		2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2		Secondes				Millisecondes			
		8	4	2	1	-	-	2 <sup>9</sup>	2 <sup>8</sup>
3		Minutes				10 secondes			
		8	4	2	1	-	40	20	10
4		Heures				10 Minutes			
		8	4	2	1	-	40	20	10
5		Jours				10 Heures			
		8	4	2	1	-	-	20	10
6		100 Jours				10 Jours			
		800	400	200	100	80	40	20	10

**Application recommandée:** information temporelle incrémentale | information temporelle absolue.  
 Date de référence selon accord.

- Élément d'information "Temps binaire 1" = CP40Time1

Information temporelle en notation binaire avec une indication d'heure d'été\* et 1 bit réservé.

CP40Time1 := CP40 { millisecondes, réserve, horaire d'été }  
 Millisecondes := UI38[1..38] < 0..2<sup>38</sup> - 1 > , c'est-à-dire 0 ms .. 8,7 années  
 Réserve = RES := BS2[39] < 0..1 >  
 Horaire d'été\* = SU := BS2[40] < 0..1 >  
 SU < 0 > := temps normalisé  
 SU < 1 > := horaire d'été.

Octets	Bits	8	7	6	5	4	3	2	1
1		2 <sup>7</sup>							
		2 <sup>0</sup>							
2		2 <sup>15</sup>							
		2 <sup>8</sup>							
3		2 <sup>23</sup>							
		2 <sup>16</sup>							
4		2 <sup>31</sup>							
		2 <sup>24</sup>							
5		SU	RES	2 <sup>37</sup>				2 <sup>32</sup>	
		MSB							

Unité de plus faible poids (LSB) = 1 ms (par défaut); autres spécifications selon accord .

**Application recommandée:** information temporelle absolue.

Date de référence selon accord.

\* Le terme anglais "summer time" est également appelé "daylight saving time" dans certains pays.

- Information element "BCD Time 1" = CP48Time1BCD

```

CP48Time1BCD:= CP48 { Milliseconds, Seconds, 10 Seconds, Minutes, 10 Minutes, Hours,
10 Hours, Days, 10 Days, 100 Days }
  Milliseconds := UI10[1..10] < 0..999 >
    Seconds := UI4[13..16] < 0..9 BCD >
    10 Seconds := UI3[17..19] < 0..5 >
    Minutes := UI4[21..24] < 0..9 BCD >
    10 Minutes := UI3[25..27] < 0..5 >
    Hours := UI4[29..32] < 0..9 BCD >
    10 Hours := UI2[33..34] < 0..2 >
    Days := UI4[37..40] < 0..9 BCD >
    10 Days := UI4[41..44] < 0..9 BCD >
    100 Days := UI4[45..48] < 0..9 BCD >
    
```

Octets	Bits	8	7	6	5	4	3	2	1
		Milliseconds							
1		2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
2		Seconds				Milliseconds			
		8	4	2	1	-	-	2 <sup>9</sup>	2 <sup>8</sup>
3		Minutes				10 Seconds			
		8	4	2	1	-	40	20	10
4		Hours				10 Minutes			
		8	4	2	1	-	40	20	10
5		Days				10 Hours			
		8	4	2	1	-	-	20	10
6		100 Days				10 Days			
		800	400	200	100	80	40	20	10

**Recommended application:** incremental time information | absolute time information.  
**Reference date by agreement.**

- Information element "Binary Time 1" = CP40Time1

Time information in binary notation with summer time\* indication and with 1 bit reserve.

```

CP40Time1:= CP40 { Milliseconds, Reserve, Summer time }
  Milliseconds := UI38[1..38] < 0..238 - 1 > , that is 0 ms .. 8,7 years
  Reserve = RES := BS2[39] < 0..1 >
  Summer time* = SU := BS2[40] < 0..1 >
  SU < 0 > := standard time
  SU < 1 > := summer time*
    
```

Octets	Bits	8	7	6	5	4	3	2	1
1		LSB							
		↓							
1		2 <sup>0</sup>							
2		2 <sup>8</sup>							
3		2 <sup>16</sup>							
4		2 <sup>24</sup>							
5		SU	RES	2 <sup>37</sup>				2 <sup>32</sup>	
		↑							
		MSB							

Unit of LSB = 1 ms (default); other specifications by agreement.

**Recommended application:** absolute time information.

**Reference date by agreement.**

\* "summer time" is also called "daylight saving time" in some countries.



- Information element "Binary Time 2a" = CP56Time2a

CP56Time2a := CP56 { Milliseconds, Minutes, Reserve1, Invalid, Hours, Reserve2, Summer time  
 Day of month, Day of week, Months, Reserve3, Years, Reserve4 }  
 Milliseconds := UI16[1..16] < 0..59 999 >  
 Minutes := UI6[17..22] < 0..59 >  
 Reserve1 = RES1 := BS1[23]  
 Invalid = IV := BS1[24] < 0..1 > IV < 0 > := valid  
 IV < 1 > := invalid  
 Hours := UI5[25..29] < 0..23 >  
 Reserve2 = RES2 := BS2[30..31]  
 Summer time = SU := BS1[32] < 0..1 > SU < 0 > := standard time  
 SU < 1 > := summer time  
 Day of month := UI5[33..37] < 1..31 >  
 Day of week := UI3[38..40] < 1..7 >  
 Months := UI4[41..44] < 1..12 >  
 Reserve3 = RES3 := BS4[45..48]  
 Years := UI7[49..55] < 0..99 >  
 Reserve4 = RES4 := BS1[56]

Octets	Bits	8	7	6	5	4	3	2	1	
1	Milliseconds									0 .. 59 999 ms
2	Milliseconds									
3	Minutes									0 .. 59 min
4	IV	RES1	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>		
5	Hours									0 .. 23 h
6	SU	RES2		2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>		
7	Day of month			Day of week						1 .. 31 days of month
8	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	1 .. 7 days of week	
9	RES3					Months				1 .. 12 months
10	Years									0 .. 99 years
11	RES4	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>		

NOTE - Shortened or incremental time information is transmitted by discarding high order octets or low order octet as appropriate.

- Information element "Binary Time 2b" = CP56Time2b, see Information Element "Binary Time 2a", however, the count of months of year is replaced by the count of weeks of year.

CP56Time2b := CP56 { Milliseconds, Minutes, Invalid, Hours, Summer time, Day of month,  
 Day of week, Weeks, Years }  
 Weeks := UI6[41..46] < 1..53 >  
 Reserve3 = RES3 := BS2[47..48]  
 Years := UI7[49..55] < 0..99 >

- Information element "Binary Time 2c" = CP56Time2c, see Information Element "Binary Time 2a", however, the unstructured count of milliseconds and seconds is replaced by a structured count.

CP56Time2c := CP56 { Milliseconds, Seconds, Minutes, Invalid, Hours, Summer time,  
 Day of month, Day of week, Months, Years }  
 Milliseconds := UI10[1..10] < 0..999 >  
 Seconds := UI6[11..16] < 0..59 >

Recommended application: incremental time information | absolute time information.  
 Reference date by agreement.

**Index des ensembles d'éléments d'information**

	Pages	
Type 1	ENTIER SANS SIGNE	22
Type 1.1	Entier binaire sans signe	22
	Commande double	22
	Commande continue de régulation	22
	Commande de régulation par échelons	22
	Information de signalisation double	22
	Code 1-parmi-8	22
	Entier sans signe à 8 bits	22
	Valeur dans l'intervalle de 0 à 250	22
Type 1.2	Entier sans signe codé DCB	24
	Entier sans signe codé DCB à 6 chiffres	24
Type 2	ENTIER	24
Type 2.1	Entier binaire avec signe	24
	Entier à 8 bits	24
	Entier à 12 bits aligné à droite	24
	Entier à 12 bits aligné à gauche	24
Type 2.2	Entier avec signe codé DCB	24
	Entier codé DCB à 5 chiffres	24
Type 3	NOMBRE À VIRGULE FIXE SANS SIGNE	26
Type 3.1	Nombre normalisé à virgule fixe sans signe	26
	Nombre normalisé à virgule fixe sans signe à 8 bits	26
Type 3.2	Nombre à virgule fixe sans signe normalisé à $\pm 2^J \times 100\%$	26
	Nombre à virgule fixe sans signe à 8 bits normalisé à 200%	26
Type 4	NOMBRE À VIRGULE FIXE	26
Type 4.1	Nombre normalisé à virgule fixe avec signe	26
	Nombre normalisé à virgule fixe avec signe à 16 bits	26
	Nombre normalisé à virgule fixe avec signe à 12 bits aligné à droite	26
	Nombre normalisé à virgule fixe avec signe à 12 bits aligné à gauche	26
Type 4.2	Nombre à virgule fixe avec signe normalisé à $\pm 2^J \times 100\%$	28
	Nombre à virgule fixe avec signe normalisé à $\pm 200\%$	28
Type 5	RÉEL	28
	Nombre court en virgule flottante IEEE STD 754	28
Type 6	CHAÎNE BINAIRE	30
	Commande simple	30
	Information de signalisation simple	30
	Registre d'état à 8 bits	30
	Etat à 8 bits+détection de transitoire	30
	Etat à 16 bits+détection de changement d'état	30
Type 7	CHAÎNE D'OCTETS	30
	Chaîne de caractères ASCII	30
	ÉLÉMENTS D'INFORMATION COMPOSITE	32
	Valeur avec indication d'erreur	32
	Valeur dans l'intervalle de 0 à 120 et indication d'erreur	32
	Valeur normalisée avec indication d'erreur	32
	Valeur avec indication de transitoire et d'erreur	32
	Valeur normalisée avec indication de dépassement et d'erreur	34
	Valeurs avec octet de signe	36
	Objet d'information avec descripteur de qualité	36
	Lecture de compteur binaire 1	38
	Lecture de compteur binaire bipolaire 1	38
	Lecture de compteur codé DCB 1	38
	Temps codé DCB 1	40
	Temps binaire 1	40
	Temps binaire 2a	42
	Temps binaire 2b	42
	Temps binaire 2c	42

## Index of set of information elements

	Page	
Type 1	UNSIGNED INTEGER	23
Type 1.1	Unsigned binary integer	23
	Double Command	23
	Persistent Regulating Command	23
	Regulating Step Command	23
	Double-Point Information	23
	1-of-8 Code	23
	8 bit Unsigned Integer	23
	Value with Range 0 to 250	23
Type 1.2	Unsigned binary coded decimal integer	25
	6 digit Unsigned BCD Integer	25
Type 2	INTEGER	25
Type 2.1	Signed binary integer	25
	8 bit Integer	25
	12 bit Integer Rightjustified	25
	12 bit Integer Leftjustified	25
Type 2.2	Signed binary coded decimal integer	25
	5 digit BCD Integer	25
Type 3	UNSIGNED FIXED POINT NUMBER	27
Type 3.1	Normalized unsigned fixed point number	27
	8 bit Normalized Unsigned Fixed Point Number	27
Type 3.2	Unsigned Fixed Point Number Normalized to $\pm 2^j \times 100\%$	27
	Unsigned 8 bit Fixed Point Number Normalized to 200%	27
Type 4	FIXED POINT NUMBER	27
Type 4.1	Normalized signed fixed point number	27
	16 bit Normalized Signed Fixed Point Number	27
	12 bit Normalized Signed Fixed Point Number Rightjustified	27
	12 bit Normalized Signed Fixed Point Number Leftjustified	27
Type 4.2	Signed fixed point number normalized to $\pm 2^j \times 100\%$	29
	Signed Fixed Point Number Normalized to $\pm 200\%$	29
Type 5	REAL	29
	Short Floating Point Number IEEE STD 754	29
Type 6	BITSTRING	31
	Single Command	31
	Single-Point Information	31
	8 bit Status Register	31
	8 bit Status+Transient Detection	31
	16 bit Status+Status Change Detection	31
Type 7	OCTETSTRING	31
	ASCII Character String	31
	COMPOUND INFORMATION ELEMENTS	33
	Value with Error Indication	33
	Value with Range 120 and Error Indication	33
	Normalized Value with Error Indication	33
	Value with Transient and Error Indication	33
	Normalized Value with Error and Overflow Indication	35
	Values with Sign Octet	37
	Information Object with Quality Descriptor	37
	Binary Counter Reading 1	39
	Bipolar Binary Counter Reading 1	39
	BCD Counter Reading 1	39
	BCD Time 1	41
	Binary Time 1	41
	Binary Time 2a	43
	Binary Time 2b	43
	Binary Time 2c	43

LICENSED TO MECON Limited. - RANCHI/BANGALORE  
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.



LICENSED TO MECON Limited, - RANCHI/BANGALORE  
FOR INTERNAL USE AT THIS LOCATION ONLY, SUPPLIED BY BOOK SUPPLY BUREAU.

---

**ICS 33.200**

---