

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

60713

Première édition
First edition
1981-01

Sous-programmes CAMAC

Subroutines for CAMAC



Numéro de référence
Reference number
CEI/IEC 60713: 1981

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

Numéros des publications

Depuis le 1er janvier 1997, les publications de la CEI sont numérotées à partir de 60000.

Publications consolidées

Les versions consolidées de certaines publications de la CEI incorporant les amendements sont disponibles. Par exemple, les numéros d'édition 1.0, 1.1 et 1.2 indiquent respectivement la publication de base, la publication de base incorporant l'amendement 1, et la publication de base incorporant les amendements 1 et 2.

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 dans le Catalogue de la CEI.

Les renseignements relatifs à des questions à l'étude et des travaux en cours entrepris par le comité technique qui a établi cette publication, ainsi que la liste des publications établies, se trouvent dans les documents ci-dessous:

- **«Site web» de la CEI***
- **Catalogue des publications de la CEI**
Publié annuellement et mis à jour régulièrement (Catalogue en ligne)*
- **Bulletin de la CEI**
Disponible à la fois au «site web» de la CEI* et comme périodique imprimé

Terminologie, symboles graphiques et littéraux

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 60050: *Vocabulaire Electrotechnique International* (VEI).

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera la CEI 60027: *Symboles littéraux à utiliser en électrotechnique*, la CEI 60417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles*, et la CEI 60617: *Symboles graphiques pour schémas*.

* Voir adresse «site web» sur la page de titre.

Numbering

As from 1 January 1997 all IEC publications are issued with a designation in the 60000 series.

Consolidated publications

Consolidated versions of some IEC publications including amendments are available. For example, edition numbers 1.0, 1.1 and 1.2 refer, respectively, to the base publication, the base publication incorporating amendment 1 and the base publication incorporating amendments 1 and 2.

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 in the IEC catalogue.

Information on the subjects under consideration and work in progress undertaken by the technical committee which has prepared this publication, as well as the list of publications issued, is to be found at the following IEC sources:

- **IEC web site***
- **Catalogue of IEC publications**
Published yearly with regular updates (On-line catalogue)*
- **IEC Bulletin**
Available both at the IEC web site* and as a printed periodical

Terminology, graphical and letter symbols

For general terminology, readers are referred to IEC 60050: *International Electrotechnical Vocabulary* (IEV).

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications IEC 60027: *Letter symbols to be used in electrical technology*, IEC 60417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets* and IEC 60617: *Graphical symbols for diagrams*.

* See web site address on title page.

**NORME
INTERNATIONALE
INTERNATIONAL
STANDARD**

**CEI
IEC**

60713

Première édition
First edition
1981-01

Sous-programmes CAMAC

Subroutines for CAMAC

© IEC 1981 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.

International Electrotechnical Commission
Telefax: +41 22 919 0300

3, rue de Varembé Geneva, Switzerland
e-mail: inmail@iec.ch IEC web site <http://www.iec.ch>



Commission Electrotechnique Internationale
International Electrotechnical Commission
Международная Электротехническая Комиссия

CODE PRIX
PRICE CODE

U

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

SOMMAIRE

	Pages
PRÉAMBULE	6
PRÉFACE	6
Articles	
1. Domaine d'application	8
2. Objet	8
3. Généralités	8
4. Spécifications fonctionnelles	10
4.1 Sous-programmes de base	12
4.1.1 Déclaration d'un registre CAMAC	12
4.1.2 Exécution d'une action simple CAMAC	12
4.2 Sous-programmes d'action simple	12
4.2.1 Emission du signal d'initialisation sur l'Interconnexion	14
4.2.2 Emission de la remise à zéro du châssis	14
4.2.3 Etablissement ou suppression de l'inhibition sur l'Interconnexion	14
4.2.4 Contrôle de l'inhibition sur l'Interconnexion	14
4.2.5 Mise en service ou hors service de la demande dans un châssis	14
4.2.6 Contrôle de la mise en service de la demande dans un châssis	14
4.2.7 Contrôle de la présence de la demande dans un châssis	16
4.2.8 Déclaration de LAM	16
4.2.9 Mise en service ou hors service de LAM	16
4.2.10 Remise à zéro de LAM	16
4.2.11 Contrôle de LAM	16
4.2.12 Liaison du LAM à la procédure de service	18
4.3 Transferts de bloc, actions multiples et déclarations inverses	18
4.3.1 Action multiple générale	18
4.3.2 Scrutation d'adresses	20
4.3.3 Transfert de bloc synchronisé par le contrôleur	20
4.3.4 Transfert de bloc synchronisé par LAM	22
4.3.5 Transfert de bloc en mode répétitif	22
4.3.6 Analyse de l'identificateur de LAM	24
4.3.7 Analyse de l'identificateur de registre	24
5. Définitions de paramètres	24
5.1 <i>ext</i> (adresse externe)	24
5.2 <i>b</i> (numéro de branche)	26
5.3 <i>c</i> (numéro de châssis)	26
5.4 <i>n</i> (numéro de la station)	26
5.5 <i>a</i> (sous-adresse)	26
5.6 <i>f</i> (code de fonction)	26
5.7 <i>int</i> (mot de données CAMAC)	26
5.8 <i>q</i> (réponse Q)	26
5.9 <i>l</i> (valeur logique de vérité)	26
5.10 <i>lam</i> (identificateur de LAM)	26
5.11 <i>m</i> (identificateur de mode d'accès de LAM)	28
5.12 <i>inta</i> (tableau d'entiers)	28
5.13 <i>label</i> (identificateur du point d'entrée)	28
5.14 <i>fa</i> (codes de fonction)	28
5.15 <i>exta</i> (adresse externe CAMAC)	28
5.16 <i>intc</i> (tableau de données CAMAC)	30
5.17 <i>qa</i> (réponses Q)	30
5.18 <i>cb</i> (bloc de contrôle)	30
5.19 <i>extb</i> (adresses externes)	30
ANNEXE A – Sous-programmes dépendants du système	32
A1. Introduction	32
A2. Accès aux signaux spéciaux	32
A2.1 Initialisation de branche	32
A2.2 Contrôle de l'état de l'action précédente	32

CONTENTS

	Page
FOREWORD	7
PREFACE	7
Clause	
1. Scope	9
2. Object	9
3. General	9
4. Functional Specifications	11
4.1 Primary Subroutines	13
4.1.1 Declare CAMAC Register	13
4.1.2 Perform Single CAMAC Action	13
4.2 Single-Action Subroutines	13
4.2.1 Generate Dataway Initialize	15
4.2.2 Generate Crate Clear	15
4.2.3 Set or Clear Dataway Inhibit	15
4.2.4 Test Dataway Inhibit	15
4.2.5 Enable or Disable Crate Demand	15
4.2.6 Test Crate Demand Enabled	15
4.2.7 Test Crate Demand Present	17
4.2.8 Declare LAM	17
4.2.9 Enable or Disable LAM	17
4.2.10 Clear LAM	17
4.2.11 Test LAM	17
4.2.12 Link LAM to Service Procedure	19
4.3 Block Transfers, Multiple Actions and Inverse Declarations	19
4.3.1 General Multiple Action	19
4.3.2 Address Scan	21
4.3.3 Controller-Synchronized Block Transfer	21
4.3.4 LAM-Synchronized Block Transfer	23
4.3.5 Repeat Mode Block Transfer	23
4.3.6 Analyze LAM Identifier	25
4.3.7 Analyze Register Identifier	25
5. Definitions of Parameters	25
5.1 <i>ext</i> (external address)	25
5.2 <i>b</i> (branch number)	27
5.3 <i>c</i> (crate number)	27
5.4 <i>n</i> (station number)	27
5.5 <i>a</i> (subaddress)	27
5.6 <i>f</i> (function code)	27
5.7 <i>int</i> (CAMAC data word)	27
5.8 <i>q</i> (Q response)	27
5.9 <i>l</i> (logical truth value)	27
5.10 <i>lam</i> (LAM identifier)	27
5.11 <i>m</i> (LAM access specifier)	29
5.12 <i>inta</i> (integer array)	29
5.13 <i>label</i> (entry point identifier)	29
5.14 <i>fa</i> (function codes)	29
5.15 <i>exta</i> (CAMAC external address)	29
5.16 <i>intc</i> (CAMAC data array)	31
5.17 <i>qa</i> (Q responses)	31
5.18 <i>cb</i> (control block)	31
5.19 <i>extb</i> (external addresses)	31
APPENDIX A – System-Dependent Subroutines	33
A1. Introduction	33
A2. Access to Special Signals	33
A2.1 Branch Initialize	33
A2.2 Test Status of Preceding Action	33

Articles	Pages
A3. Identificateur de canal	34
A3.1 Déclaration de canal	34
A3.2 Analyse de la déclaration de canal	34
A4. Transferts de mot de données court	34
A4.1 Exécution d'une action simple CAMAC	34
A4.2 Action multiple générale	36
A4.3 Scrutation d'adresses	36
A4.4 Transfert de bloc synchronisé par le contrôleur	36
A4.5 Transfert de bloc synchronisé par LAM	36
A4.6 Transfert de bloc en mode répétitif	36
A5. Définition de l'identificateur de châssis	38
A6. Définitions de paramètres	38
A6.1 <i>k</i> (code d'état)	38
A6.2 <i>chan</i> (identificateur de canal)	38
A6.3 <i>ints</i> (mot de données CAMAC tronqué)	38
A6.4 <i>intt</i> (tableau de données CAMAC tronquées)	38
A6.5 <i>intb</i> (tableau d'entiers)	38
ANNEXE B – Mise en œuvre en FORTRAN	40
B1. Généralités	40
B2. Descriptions des sous-programmes	40
B2.1 Sous-programmes de base	40
B2.1.1 Déclaration d'un registre CAMAC	40
B2.1.2 Exécution d'une action simple CAMAC	40
B2.2 Sous-programmes d'action simple	42
B2.2.1 Emission du signal d'initialisation sur l'Interconnexion	42
B2.2.2 Emission de la remise à zéro du châssis	42
B2.2.3 Etablissement ou suppression de l'inhibition sur l'Interconnexion	42
B2.2.4 Contrôle de l'inhibition sur l'Interconnexion	44
B2.2.5 Mise en service ou hors service des demandes de châssis	44
B2.2.6 Contrôle de la mise en service de la demande de châssis	44
B2.2.7 Contrôle de la présence de demande	44
B2.2.8 Déclaration de LAM	44
B2.2.9 Mise en service ou hors service de LAM	44
B2.2.10 Remise à zéro de LAM	46
B2.2.11 Contrôle de LAM	46
B2.2.12 Liaison du LAM à la procédure de service	46
B2.3 Transferts de bloc, actions multiples et déclarations inverses	46
B2.3.1 Action multiple générale	46
B2.3.2 Scrutation d'adresses	48
B2.3.3 Transfert de bloc synchronisé par le contrôleur	50
B2.3.4 Transfert de bloc synchronisé par LAM	50
B2.3.5 Transfert de bloc en mode répétitif	50
B2.3.6 Analyse de l'identificateur de LAM	52
B2.3.7 Analyse de l'identificateur de registre	52
B3. Types de paramètres	52
B3.1 Entiers simples	52
B3.2 Valeurs logiques simples	52
B3.3 Tableaux d'entiers	52
B3.4 Tableau logique	52
B3.5 Mot de données CAMAC	54
B3.6 Tableau de données CAMAC	54
B3.7 Etiquette	54
ANNEXE C – Symboles mnémoniques du code de fonction	56

Clause	Page
A3. Channel Identifier	35
A3.1 Declare Channel	35
A3.2 Analyzer Channel Declaration	35
A4. Short Data-Word Transfers	35
A4.1 Perform Single CAMAC Action	35
A4.2 General Multiple Action	37
A4.3 Address Scan	37
A4.4 Controller-Synchronized Block Transfer	37
A4.5 LAM-Synchronized Block Transfer	37
A4.6 Repeat Mode Block Transfer	37
A5. Define Crate Identifier	39
A6. Definitions of Parameters	39
A6.1 <i>k</i> (status code)	39
A6.2 <i>chan</i> (channel identifier)	39
A6.3 <i>ints</i> (truncated CAMAC data word)	39
A6.4 <i>intt</i> (truncated CAMAC data array)	39
A6.5 <i>intb</i> (integer array)	39
APPENDIX B – FORTRAN Implementation	41
B1. General	41
B2. Description of Subroutines	41
B2.1 Primary Subroutines	41
B2.1.1 Declare CAMAC Register	41
B2.1.2 Perform Single CAMAC Action	41
B2.2 Single Action Subroutines	43
B2.2.1 Generate Dataway Initialize	43
B2.2.2 Generate Crate Clear	43
B2.2.3 Set or Clear Dataway Inhibit	43
B2.2.4 Test Dataway Inhibit	45
B2.2.5 Enable or Disable Crate Demands	45
B2.2.6 Test Crate Demand Enabled	45
B2.2.7 Test Demand Present	45
B2.2.8 Declare LAM	45
B2.2.9 Enable or Disable LAM	45
B2.2.10 Clear LAM	47
B2.2.11 Test LAM	47
B2.2.12 Link LAM to Service Procedure	47
B2.3 Block Transfers, Multiple Actions, and Inverse Declarations	47
B2.3.1 General Multiple Action	47
B2.3.2 Address-Scan	49
B2.3.3 Controller-Synchronized Block Transfer	51
B2.3.4 LAM-Synchronized Block Transfer	51
B2.3.5 Repeat-Mode Block Transfer	51
B2.3.6 Analyze LAM Identifier	53
B2.3.7 Analyze Register Identifier	53
B3. Parameter Types	53
B3.1 Single Integers	53
B3.2 Single Logical Values	53
B3.3 Integer Arrays	53
B3.4 Logical Array	53
B3.5 CAMAC Data Word	55
B3.6 CAMAC Data Array	55
B3.7 Label	55
APPENDIX C – Function-Code Mnemonic Symbols	57

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

SOUS-PROGRAMMES CAMAC

PRÉAMBULE

- 1) Les décisions ou accords officiels de la CEI en ce qui concerne les questions techniques, préparés par des Comités d'Etudes 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.
- 2) Ces décisions constituent des recommandations internationales et sont agréées comme telles par les Comités nationaux.
- 3) Dans le but d'encourager l'unification internationale, la CEI exprime le vœu que tous les Comités nationaux adoptent dans leurs règles nationales le texte de la recommandation de la CEI, dans la mesure où les conditions nationales le permettent. Toute divergence entre la recommandation de la CEI et la règle nationale correspondante doit, dans la mesure du possible, être indiquée en termes clairs dans cette dernière.

PRÉFACE

La présente norme a été établie par le Comité d'Etudes N° 45 de la CEI: Instrumentation nucléaire.

Un premier projet fut discuté lors de la réunion tenue à Stockholm en 1980. A la suite de cette réunion, un projet, document 45(Bureau Central)142, fut soumis à l'approbation des Comités nationaux suivant la Règle des Six Mois en août 1980.

Les Comités nationaux des pays ci-après se sont prononcés explicitement en faveur de la publication:

Afrique du Sud (République d')	Finlande
Allemagne	France
Australie	Italie
Autriche	Pays-Bas
Belgique	Pologne
Canada	République Démocratique Allemande
Egypte	Turquie
Espagne	Union des Républiques
Etats-Unis d'Amérique	Socialistes Soviétiques

Autres publications de la CEI citées dans la présente norme:

Publications n°s	516: Système modulaire d'instrumentation pour le traitement de l'information; système CAMAC.
	552: Système CAMAC - Organisation de systèmes multichâssis. Spécification de l'Interconnexion de branche et du contrôleur de châssis type A1.
	640: Système CAMAC - Interface pour Interconnexion de Branche Série.
	677: Transferts de bloc dans les systèmes CAMAC.

Autres publications: IML: CAMAC - Définition de l'IML, langage à utiliser dans les systèmes CAMAC, ESONE/IML/01, octobre 1974, Secrétariat ESONE, et TID-26615, janvier 1975, DOE, Washington, D.C., Etats-Unis d'Amérique.

INTERNATIONAL ELECTROTECHNICAL COMMISSION

SUBROUTINES FOR CAMAC

FOREWORD

- 1) 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.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules in so far as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rule should, as far as possible, be clearly indicated in the latter.

PREFACE

This standard has been prepared by IEC Technical Committee No. 45: Nuclear Instrumentation.

A first draft was discussed at the meeting held in Stockholm in 1980. As a result of this meeting, a draft, Document 45(Central Office)142, was submitted to the National Committees for approval under the Six Months' Rule in August 1980.

The National Committees of the following countries voted explicitly in favour of publication:

Australia	Italy
Austria	Netherlands
Belgium	Poland
Canada	South Africa (Republic of)
Egypt	Spain
Finland	Turkey
France	Union of Soviet
German Democratic Republic	Socialist Republics
Germany	United States of America

Other IEC publications quoted in this standard:

- Publications Nos. 516: A Modular Instrumentation System for Data Handling; CAMAC System.
 552: CAMAC - Organization of Multi-crate Systems. Specification of the Branch-highway and CAMAC Crate Controller Type A1.
 640: CAMAC - Serial Highway Interface System.
 677: Block Transfers in CAMAC Systems.

Other publications: IML: CAMAC - The Definition of IML, a Language for use in CAMAC Systems, ESONE/IML/01, October 1974, ESONE Secretariat, and TID-26615, January 1975, DOE, Washington, D.C., United States of America.

SOUS-PROGRAMMES CAMAC

1. Domaine d'application

La présente norme s'applique aux sous-programmes pour les systèmes CAMAC tels qu'ils sont définis dans la Publication 516 de la CEI: Système modulaire d'instrumentation pour le traitement de l'information; système CAMAC. Son application ne doit pas s'opposer aux dispositions obligatoires de la Publication 516 de la CEI ni provoquer d'opposition à leur rencontre.

2. Objet

La présente norme offre un ensemble de sous-programmes destinés à fournir des moyens généraux de communication avec les systèmes CAMAC définis dans la Publication 516 de la CEI.

Cette norme sera d'un intérêt primordial pour ceux qui souhaitent écrire leurs propres programmes de traitement de données dans un langage de haut niveau tel que le FORTRAN.

Le taux de transfert de données réalisable dépend naturellement d'un certain nombre de facteurs comme le langage utilisé, le système d'exploitation, le compilateur, les méthodes employées pour réaliser la fonction du sous-programme, le niveau d'exécution de celui-ci ainsi que des performances de l'ordinateur.

3. Généralités

Cette norme décrit un ensemble de sous-programmes normalisés permettant à différents langages de programmation de communiquer avec les systèmes CAMAC.

Intentionnellement, ces sous-programmes sont adaptés à l'utilisation du langage FORTRAN, bien qu'ils ne soient pas limités à ce seul langage. L'annexe B décrit une mise en œuvre recommandée de ces sous-programmes en FORTRAN.

La présente approche est largement fondée sur les publications de l'ESONE/IML/01 et TID-26615 sur IML: CAMAC - Définition de l'IML, langage à utiliser dans les systèmes CAMAC. Une distinction est faite entre les «déclarations», qui sont utilisées pour dénommer et spécifier les entités de l'ordinateur ou celles du CAMAC, et les «actions», qui sont employées pour réaliser les divers mouvements de données et effectuer les tests de conditions, le tout formant la portion du programme relative au CAMAC. Autant que possible, la nomenclature de l'IML-M1 a été suivie de façon à bénéficier de la routine actuelle de ce système et assurer une terminologie et un style aussi conformes que possible aux divers documents de logiciel CAMAC.

Du fait de l'emploi étendu du CAMAC avec des ordinateurs dont la longueur de mot est inférieure à 24 bits et parce qu'il existe de grandes différences entre les caractéristiques des ordinateurs et des systèmes d'exploitation, des caractéristiques spéciales dépendantes du système sont souvent exigées pour donner une plus grande efficacité ou pour utiliser au mieux les caractéristiques de ces systèmes particuliers. La partie principale de cette norme décrit des sous-programmes qui, au niveau de l'interface de l'utilisateur, dépendent seulement des caractéristiques CAMAC, et qui, par conséquent, lorsqu'ils sont écrits dans n'importe quel langage de procédure normalisé, devraient être indépendants de l'ordinateur. L'annexe A décrit des sous-programmes qui ne dépendent pas seulement du CAMAC, mais jusqu'à un

SUBROUTINES FOR CAMAC

1. Scope

This standard covers requirements for subroutines for CAMAC systems as defined in IEC Publication 516: A Modular Instrumentation System for Data Handling; CAMAC System. Its application shall not conflict or cause conflict with the mandatory requirements of IEC Publication 516.

2. Object

Recommendations are presented for a set of software subroutines to provide a general capability for communications with CAMAC systems as defined in IEC Publication 516.

They will be of primary interest to those who wish to write their own data-processing programs in a high level programming language, such as FORTRAN.

The achievable data transfer rate is of course dependent on a number of factors including the language used, the operating system, the compiler, the method and level of subroutine implementation and the computer.

3. General

This standard describes a set of standard subroutines to provide access to CAMAC facilities in a variety of computer-programming languages.

It is specifically intended that the subroutines be suitable for use with FORTRAN although they are not restricted to that language. Appendix B describes a recommended implementation of the subroutines explicitly for FORTRAN.

The present approach is based largely on ESONE/IML/01 and TID-26615 publications on IML: CAMAC - The Definition of IML, a Language for use in CAMAC Systems. A distinction is made between "declarations", which are used to name and specify computer and CAMAC entities, and "actions", which are used to implement the various data movements and condition tests which make up the CAMAC-related portion of a program. As far as possible, the nomenclature of IML-M1 has been followed in order to take advantage of existing familiarity with that system and to provide as uniform a terminology and style as possible among the various CAMAC software documents.

Because of the widespread use of CAMAC on computers with a word length of less than 24 bits and because of the great differences in computer and operating-system features, special, system-dependent features are often required to provide greater efficiency or to make appropriate use of the features of particular systems. The main body of this standard describes subroutines which, at the user interface, depend only on the features of CAMAC and therefore, when implemented in any standard procedural language, should be computer-independent. Appendix A describes subroutines which depend not only on CAMAC, but to some extent on individual computers. They cannot be made independent of the system on which they are implemented, and the user should take special precautions when it is necessary

certain point d'ordinateurs particuliers. Ils ne peuvent pas être rendus indépendants du système pour lequel ils ont été écrits, et l'utilisateur doit prendre des précautions spéciales quand il est nécessaire de les incorporer dans un programme. Un tel programme ne peut pas être transférable d'un ordinateur à un autre sans modification.

Les sous-programmes ont été groupés en trois catégories pour permettre des mises en œuvre à différents niveaux normalisés. Le niveau le plus bas requiert seulement deux sous-programmes mais donne cependant accès à la plupart des possibilités des systèmes CAMAC. Dans les niveaux plus élevés, on a ajouté des sous-programmes qui permettent des procédures d'écriture dans une terminologie plus mnémonique, offrent une meilleure manipulation des LAM, autorisent des procédures indépendantes du type d'Interconnexion de branche CAMAC utilisé et procurent des transferts de bloc efficaces.

4. Spécifications fonctionnelles

Cet article présente et décrit en détail tous les sous-programmes recommandés. Du fait que de nombreuses mises en œuvre n'ont pas besoin de l'ensemble des sous-programmes, ceux-ci ont été divisés en trois parties correspondant aux niveaux recommandés de réalisation. Le niveau A, le plus simple, fait appel aux sous-programmes du paragraphe 4.1; le niveau B, niveau intermédiaire, requiert les sous-programmes des paragraphes 4.1 et 4.2, et le niveau C qui est le plus élevé, nécessite l'emploi des sous-programmes des paragraphes 4.1, 4.2 et 4.3.

Deux dispositions CAMAC ne sont pas accessibles par ces sous-programmes: la réponse X à une action et la commande BZ. On peut néanmoins les obtenir en utilisant les sous-programmes dépendants du système décrits dans l'annexe A.

On a adopté, pour les dénominations, des conventions compatibles avec les langages de nombreux ordinateurs. Afin que l'utilisateur évite simplement les difficultés inhérentes aux dénominations, le nom de chaque sous-programme normalisé débute par la lettre «C». Le code de la seconde lettre précise sa fonction générale. Six lettres sont utilisées à cette fin:

- «C» indique que le sous-programme exécute une fonction de contrôle-commande;
- «D» indique que le sous-programme est une déclaration d'entité CAMAC;
- «F» indique que le sous-programme transfère des mots complets de données (24 bits);
- «G» indique que le sous-programme décompose une entité CAMAC désignée en ses composantes d'adresse;
- «S» indique que le sous-programme transfère des mots tronqués de données (moins de 24 bits);
- «T» indique que le sous-programme contrôle l'état d'un signal ou l'indication d'état.

Les lettres restantes de chaque nom (jusqu'à un maximum de six) des sous-programmes sont choisies pour leurs valeurs mnémoniques caractérisant la fonction que le sous-programme effectue.

Du fait qu'aucun langage particulier n'est pris en compte dans le corps de la norme, aucune syntaxe ne peut être définie pour un appel de sous-programme. Les sous-programmes sont décrits en fonction d'un nom de sous-programme et d'une séquence ordonnée de paramètres. Dans toute la mesure possible, chaque mise en œuvre doit conserver ses noms et l'ordre des paramètres associés. Pour la même raison, aucune forme spécifique ne peut-être définie pour les paramètres des sous-programmes. Les mots de données CAMAC sont des chaînes de 24 bits. Une mise en œuvre doit pouvoir représenter une telle chaîne. D'autres paramètres sont représentés soit en valeurs entières, soit en valeurs logiques de vérité «vrai» ou «faux». Dans

to incorporate them into a program. Such a program may not be transportable from one computer to another without modification.

The subroutines have been grouped into three subsets in order to provide different standard levels of implementation. The lowest level requires only two subroutines, but nevertheless gives access to most of the facilities which can be found in CAMAC systems. In higher levels of implementation, subroutines are added which permit procedures to be written in more mnemonic terminology, provide better handling of LAM's, permit procedures to be independent of the type of CAMAC highway used and provide efficient block-transfer capability.

4. Functional specifications

This clause introduces and describes in detail all the recommended subroutines. Since many implementations will not require the complete set, the subroutines are grouped into three subsets corresponding to the recommended implementation levels. Level A, the simplest, requires only the subroutines from Sub-clause 4.1. Level B, an intermediate level, requires the subroutines from Sub-clauses 4.1 and 4.2. Level C is the highest level and requires implementation of the subroutines from Sub-clauses 4.1, 4.2 and 4.3.

Two CAMAC facilities are not available through the use of these subroutines: the X response from an action and the BZ command. These facilities are available through the use of system-dependent subroutines described in Appendix A.

Naming conventions, compatible with the requirements of many computer languages, have been adopted. In order to make it simple for a user to avoid name conflicts, the name of each recommended subroutine begins with the letter "C". The second letter of the subroutine name is coded to indicate the general function of the subroutine. Six letters have been used for this purpose:

- "C" indicates that the subroutine performs a control function;
- "D" indicates that the subroutine is a declaration of a CAMAC entity;
- "F" indicates that the subroutine transfers full-length (24-bit) data words;
- "G" indicates that the subroutine analyzes a named CAMAC entity into its address components;
- "S" indicates that the subroutine transfers short (less than 24-bit) data words;
- "T" indicates that the subroutine tests the state of a signal or status indication.

The remaining letters of each subroutine name (to a maximum of six) are chosen for their mnemonic value in identifying which function the subroutine performs.

Since no particular language is assumed in the body of the standard, no syntax can be defined for a subroutine call. The subroutines are described in terms of a subroutine name and an ordered sequence of parameters. As far as possible every implementation should retain the designated name and the order of the parameters. For the same reason, no specific form can be defined for the subroutine parameters. CAMAC data words are bit strings with a length of 24 bits. An implementation must have the capacity to represent such strings. Other subroutine parameters are represented either as integer values or as the logical values "true" or "false". In this, the parameters of subroutines are described as variables or arrays of types CAMAC

cette norme, les paramètres des sous-programmes sont représentés comme des variables ou tableaux du type mot CAMAC, entier ou logique. Pour chaque réalisation, des unités de stockage appropriées et des formats de données doivent être choisis pour chacune de ces entités généralisées.

4.1 *Sous-programmes de base*

Ces deux sous-programmes, qui sont nécessaires dans toutes les mises en œuvre, forment le niveau A. Le premier fournit le moyen de définir l'adresse d'une entité CAMAC et d'y avoir accès. Le second est utilisé pour effectuer les opérations CAMAC sur les entités définies. En principe, toute entité CAMAC pour laquelle existe un mode d'accès normalisé défini est accessible par ces deux sous-programmes. En pratique, certains systèmes peuvent comporter des restrictions dans l'utilisation des contrôleurs de châssis ou de modules d'un autre système.

4.1.1 *Déclaration d'un registre CAMAC*

Nom: CDREG.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
b (numéro de branche, voir paragraphe 5.2);
c (numéro de châssis, voir paragraphe 5.3);
n (numéro de station, voir paragraphe 5.4);
a (sous-adresse, voir paragraphe 5.5).

Fonction: CDREG réunit le numéro de branche *b*, le numéro de châssis *c*, le numéro de station *n* et la sous-adresse *a* en une forme adéquate dépendante du système utilisé et stocke le résultat en *ext*. Du fait que la méthode de codage dépend de la réalisation, les contenus de *ext* ne devraient pas être modifiés par le programme. Quelques sous-programmes (voir paragraphe 4.2) n'exigent qu'une adresse de châssis; si les paramètres *n* et *a* ont tous deux la valeur zéro, CDREG code une adresse de châssis et la stocke en *ext*.

4.1.2 *Exécution d'une action simple CAMAC*

Nom: CFSA.

Paramètres: *f* (code de fonction, voir paragraphe 5.6);
ext (adresse externe, voir paragraphe 5.1);
int (mot de données CAMAC, voir paragraphe 5.7);
q (réponse Q, voir paragraphe 5.8).

Fonction: CFSA provoque une action CAMAC spécifiée par le code de fonction *f*, qui doit être exécutée à l'adresse CAMAC spécifiée par *ext*. Si *f* contient un code de lecture ou d'écriture, un transfert de données sur 24 bits est effectué entre le registre CAMAC adressé par *ext* et l'emplacement du stockage *int* de l'ordinateur. Sans quoi *int* est ignoré. L'état de Q résultant de l'opération est stocké en *q*, «vrai» si *Q* = 1, «faux» si *Q* = 0.

4.2 *Sous-programmes d'action simple*

Ces sous-programmes, en même temps que ceux qui sont décrits dans le paragraphe 4.1, constituent la mise en œuvre de niveau B, qui fournit un ensemble complet de spécifications des actions CAMAC simples dans une forme mnémonique, concise et indépendante du type d'Interconnexion de branche ou de contrôleur de châssis. Ils permettent de déclarer les appels (LAM) et d'exécuter des actions relatives à ceux-ci en utilisant des constructions indépendantes du mode d'accès au LAM (c'est-à-dire accès de la sous-adresse ou du registre).

word, integer, or logical. For each implementation, appropriate storage units and data formats must be chosen for each of these generalized entities.

4.1 Primary Subroutines

These two subroutines, which are required in all implementations, make up level A. The first provides the capability to define the address of a CAMAC entity and to access it. The second is used to perform CAMAC operations on the defined entities. In principle any CAMAC entity for which there is a defined standard mode of access can be accessed through the use of these two subroutines. In practice, some systems may contain restrictions on the use of crate controllers or other system modules.

4.1.1 Declare CAMAC Register

Name: CDREG.

Parameters: *ext* (external address, see Sub-clause 5.1);
b (branch number, see Sub-clause 5.2);
c (crate number, see Sub-clause 5.3);
n station number, see Sub-clause 5.4);
a (subaddress, see Sub-clause 5.5).

Function: CDREG combines the branch number *b*, the crate number *c*, the station number *n*, and the subaddress *a* into a convenient system-dependent form and stores the result in *ext*. Since the method of encoding depends on the implementation, the contents of *ext* should not be modified by the program. Some subroutines (see Sub-clause 4.2) require only a crate address; if the parameters *n* and *a* are both zero, CDREG encodes a crate address and stores it in *ext*.

4.1.2 Perform Single CAMAC Action

Name: CFSA.

Parameters: *f* (function code, see Sub-clause 5.6);
ext (external address, see Sub-clause 5.1);
int (CAMAC data word, see Sub-clause 5.7);
q (Q response, see Sub-clause 5.8).

Function: CFSA causes the CAMAC action specified by the function code of *f* to be performed at the CAMAC address specified by *ext*. If *f* contains a read or write code, a 24-bit data transfer occurs between the CAMAC register addressed by *ext* and the computer storage location *int*. Otherwise *int* is ignored. The state of Q resulting from the operation is stored in *q*, "true" if Q = 1, "false" if Q = 0.

4.2 Single-Action Subroutines

These subroutines, together with those described in Sub-clause 4.1, form the level B implementation, which provides a complete facility for specifying single CAMAC actions in a way which is mnemonic, compact, and independent of the type of highway or crate controller. Facilities are provided for declaring LAM's and performing LAM actions using constructions which are independent of the LAM access mode (i.e. subaddress or register access).

4.2.1 Emission du signal d'initialisation sur l'Interconnexion

Nom: CCCZ.

Paramètre: *ext* (adresse externe, voir paragraphe 5.1).

Fonction: CCCZ provoque l'émission du signal d'initialisation (Z) sur l'Interconnexion du châssis spécifié par *ext*.

4.2.2 Emission de la remise à zéro du châssis

Nom: CCCC.

Paramètre: *ext* (adresse externe, voir paragraphe 5.1).

Fonction: CCCC provoque l'émission du signal de remise à zéro (C) sur l'Interconnexion du châssis spécifié par *ext*.

4.2.3 Etablissement ou suppression de l'inhibition sur l'Interconnexion

Nom: CCCI.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CCCI provoque l'établissement de l'inhibition (I) sur l'Interconnexion d'un châssis spécifié par *ext* si la valeur de *l* est à «vrai» ou sa suppression si la valeur de *l* est à «faux».

4.2.4 Contrôle de l'inhibition sur l'Interconnexion

Nom: CTCI.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CTCI met la valeur de *l* à «vrai» si le signal d'inhibition sur l'Interconnexion dans le châssis spécifié par *ext* est présent et met la valeur de *l* à «faux» si ce signal est absent.

4.2.5 Mise en service ou hors service de la demande dans un châssis

Nom: CCCD.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CCCD provoque la mise en service de la demande du châssis spécifié par *ext* si la valeur de *l* est à «vrai» et sa mise hors service si la valeur de *l* est à «faux».

4.2.6 Contrôle de la mise en service de la demande dans un châssis

Nom: CTCD.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CTCD met la valeur de *l* à «vrai» si la demande du châssis spécifié par *ext* est en service et met la valeur de *l* à «faux» si la demande du châssis est hors service.

4.2.1 Generate Dataway Initialize

Name: CCCZ.

Parameter: *ext* (external address, see Sub-clause 5.1).

Function: CCCZ causes Dataway Initialize (Z) to be generated in the crate specified by *ext*.

4.2.2 Generate Crate Clear

Name: CCCC.

Parameter: *ext* (external address, see Sub-clause 5.1).

Function: CCCC causes Dataway Clear (C) to be generated in the crate specified by *ext*.

4.2.3 Set or Clear Dataway Inhibit

Name: CCCI.

Parameters: *ext* (external address, see Sub-clause 5.1);
l (logical truth value, see Sub-clause 5.9).

Function: CCCI causes Dataway Inhibit (I) to be set in the crate specified by *ext* if the value of *l* is "true" and to be reset if the value of *l* is "false".

4.2.4 Test Dataway Inhibit

Name: CTCI.

Parameters: *ext* (external address, see Sub-clause 5.1);
l (logical truth value, see Sub-clause 5.9).

Function: CTCI sets the value of *l* to "true" if Dataway Inhibit is set in the crate specified by *ext* and sets the value of *l* to "false" if Dataway Inhibit is not set.

4.2.5 Enable or Disable Crate Demand

Name: CCCD.

Parameters: *ext* (external address, see Sub-clause 5.1);
l (logical truth value, see Sub-clause 5.9).

Function: CCCD causes Crate Demand to be enabled in the crate specified by *ext* if the value of *l* is "true" and causes Crate Demand to be disabled if the value of *l* is "false".

4.2.6 Test Crate Demand Enabled

Name: CTCD.

Parameters: *ext* (external address, see Sub-clause 5.1);
l (logical truth value, see Sub-clause 5.9).

Function: CTCD sets the value of *l* to "true" if Crate Demand is enabled in the crate specified by *ext* and sets the value of *l* to "false" if Crate Demand is disabled.

4.2.7 Contrôle de la présence de la demande dans un châssis

Nom: CTGL.

Paramètres: *ext* (adresse externe, voir paragraphe 5.1);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CTGL met la valeur de *l* à «vrai» si n'importe quelle demande est présente dans le châssis spécifié par *ext* et met la valeur de *l* à «faux» si aucune demande n'est présente.

4.2.8 Déclaration de LAM

Nom: CDLAM.

Paramètres: *lam* (identificateur de LAM, voir paragraphe 5.10);
b (numéro de branche, voir paragraphe 5.2);
c (numéro de châssis, voir paragraphe 5.3);
n (numéro de station, voir paragraphe 5.4);
m (identificateur de mode d'accès du LAM, voir paragraphe 5.11);
inta (tableau d'entiers, voir paragraphe 5.12).

Fonction: CDLAM code le numéro de branche *b*, le numéro de châssis *c*, le numéro de station *n* et toutes les autres informations nécessaires concernant un LAM sous forme d'un entier dépendant du système et stocke le résultat dans *lam*. Le paramètre *m* est interprété comme une sous-adresse si sa valeur est supérieure ou égale à zéro. Il est interprété comme la valeur négative d'une position de bit si sa valeur est inférieure à zéro. Cette information est codée dans la valeur assignée à *lam* et utilisée par les autres sous-programmes pour déterminer si le LAM est accessible par des fonctions spéciales d'accès au LAM ou par lecture et écriture des registres du groupe 2. L'information contenue dans le tableau *inta* dépend complètement de la mise en œuvre. Elle contient tout renseignement pouvant être requis par le système de l'ordinateur ou l'interface CAMAC pour permettre au programme d'accéder au LAM. L'utilisation de *inta* par CDLAM n'est pas nécessaire, mais *inta* doit quand même figurer dans la chaîne de paramètres.

4.2.9 Mise en service ou hors service de LAM

Nom: CCLM.

Paramètres: *lam* (identificateur de LAM, voir paragraphe 5.10);
l (valeur logique de vérité, voir paragraphe 5.9).

Fonction: CCLM entraîne la mise en service du LAM désigné par *lam* si la valeur de *l* est à «vrai» et sa mise hors service si la valeur de *l* est à «faux».

4.2.10 Remise à zéro de LAM

Nom: CCLC.

Paramètre: *lam* (identificateur de LAM, voir paragraphe 5.10).

Fonction: CCLC entraîne la remise à zéro du LAM désigné par *lam*.

4.2.11 Contrôle de LAM

Nom: CTLM.

Paramètres: *lam* (identificateur de LAM, voir paragraphe 5.10);
l (valeur logique de vérité, voir paragraphe 5.9).

4.2.7 Test Crate Demand Present

Name: CTGL.

Parameters: *ext* (external address, see Sub-clause 5.1);
l (logical truth value, see Sub-clause 5.9).

Function: CTGL sets the value of *l* to “true” if any demand is present in the crate specified by *ext* and sets the value of *l* to “false” if no demand is present.

4.2.8 Declare LAM

Name: CDLAM.

Parameters: *lam* (LAM identifier, see Sub-clause 5.10);
b (branch number, see Sub-clause 5.2);
c (crate number, see Sub-clause 5.3);
n (station number, see Sub-clause 5.4);
m (LAM access specifier, see Sub-clause 5.11);
inta (integer array, see Sub-clause 5.12).

Function: CDLAM encodes the branch number *b*, the crate number *c*, the station number *n* and all other necessary information concerning a LAM into a system-dependent, integer form and stores the result in *lam*. The parameter *m* is interpreted as a subaddress if its value is greater than or equal to zero. It is interpreted as the negative of a bit position if its value is less than zero. This information is encoded in the value assigned to *lam* and used by other subroutines to determine whether the LAM is accessed via special functions for LAM access or via reading and writing group 2 registers. The information contained in the array *inta* is completely implementation-dependent. It contains any information which may be required by the computer system or the CAMAC interface to enable the program to access the LAM. The use of *inta* by CDLAM is not required, but *inta* must appear in the parameter string regardless.

4.2.9 Enable or Disable LAM

Name: CCLM.

Parameters: *lam* (LAM identifier, see Sub-clause 5.10);
l (logical truth value, see Sub-clause 5.9).

Function: CCLM causes the LAM specified by *lam* to be enabled if the value of *l* is “true” and causes it to be disabled if the value of *l* is “false”.

4.2.10 Clear LAM

Name: CCLC.

Parameter: *lam* (LAM identifier, see Sub-clause 5.10).

Function: CCLC causes the LAM specified by *lam* to be cleared.

4.2.11 Test LAM

Name: CTLM.

Parameters: *lam* (LAM identifier, see Sub-clause 5.10);
l (logical truth value, see Sub-clause 5.9).