SLOVENSKI STANDARD

SIST HD 445 S1:2003

april 2003

Subroutines for CAMAC

ICS 27.120.99; 35.240.99

Referenčna številka
SIST HD 445 S1:2003(en)

CENELEC
GENERAL SECRETARIAT
Rue Bréderode 2, Bte 5 - 1000 BRUXELLES
Tel. 02 - 511 79 32 - Telex 26257 Cenlec b

HD 445

ENGLISH VERSION

UDC: 621.039-791.2 621.317.39::621.039 CAMAC.004.24/.26(083.73)
Key words: Nuclear instrumentation - electrical - measurement of quantities of nuclear technology - computer automated measurement and control - working instructions - precautions

SUBROUTINES FOR CAMAC

Sous-programmes CAMAC **CAMAC-Unterprogramme**

BODY OF HD

The Harmonization Document consists of:

- **IEC 713 (1981) edition 1; IEC/TC 45,** not appended

This Harmonization Document was approved by CENELEC on **7 September 1983**

The English and French versions of this HD are provided by the text of the IEC publication and the German version is the official translation of the IEC text which is not yet available.

According to the CENELEC Internal Regulations the CENELEC member National Committees are bound:

to announce the existence of this Harmonization Document at national level

by or before **1984-03-01**

to publish their new harmonized national standard

by or before **1985-03-01**

to withdraw all conflicting national standards

by or before **1985-03-01.**

Harmonized national standards are listed on the HD information sheet, which is available from the CENELEC National Committees or from the CENELEC General Secretariat.

The CENELEC National Committees are the national electrotechnical committees of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Luxemburg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, United Kingdom.

NORME INTERNATIONALE

INTERNATIONAL STANDARD

CEI
IEC
60713

Première édition
First edition
1981-01

Sous-programmes CAMAC

Subroutines for CAMAC

International Electrotechnical Commission 3, rue de Varembé Geneva, Switzerland
Telefax: +41 22 919 0300 e-mail: inmail@iec.ch IEC web site http: //www.iec.ch

IEC 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

CONTENTS

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 Socialist Republics
German Democratic Republic	United States of America
Germany	

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.

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

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

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 *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.