DRAFT
DOCUMENT

TC 5/SC 2
(DE + BIML)

SUBMITTED FOR CIML PRELIMINARY ONLINE BALLOT

New OIML Document

General requirements for software controlled measuring instruments

Draft submitted for CIML Preliminary online ballot on 2008.06.26. Voting closes on 2008.09.15.

The final Draft will be submitted for approval to the CIML at its 43rd Meeting

New OIML Document

ORGANISATION INTERNATIONALE
DE MÉTROLOGIE LÉGALE

INTERNATIONAL ORGANIZATION
OF LEGAL METROLOGY

INTERNATIONAL
DOCUMENT

OIML D XX

Edition 200X (E)

General Requirements for Software Controlled Measuring Instruments.

Exigences générales pour les instruments de mesure contrôlés par logiciel

OIML D XX Edition 200X (E)

ORGANISATION INTERNATIONALE
DE METROLOGIE LEGALE

INTERNATIONAL ORGANIZATION
OF LEGAL METROLOGY

Explanatory Note

Reasons for edition of this document

- Overall need for preparation of a document concerning software of measuring instruments, declared as high priority OIML project;
- Existing OIML R-documents have been checked. It turned out there were few concerning software explicitly and quite some requirements concerning software implicitly but they were unbalanced (between different OIML Recommendations);
- There are lot of international standards concerning IT, but they do not fit to legal metrology issues completely;
- Referring to standards (quotation of standards: i.e. loading and updating of software);
- Process related approach is perhaps not usual for other metrological fields.

History of development, changes in the course of the development:

There were three major inputs during preparation of the first pre-draft:

- Answers of the SC2 members to the questionnaire;
  - After analysis of the results of the Query the importance of the issues was ranked (see annual report 2003);
  - table with ranking;
  - note: no member opposed to this approach;
- Analysis of software related requirement in OIML R documents;
- Present experience of member countries;
  - List of important ideas of member countries introduced into this document;
  - Table with an evaluation of the member's comments;
  - Existing drafts of regional software requirements for measuring instruments;

According to time schedule proposed during the first meeting of the TC 5/SC 2 held in Berlin, December 2007, it was accepted that this Second Committee would be the last and would be submitted for acceptation as a Draft Document. This 2 CD take into accounts the comments received on the 1CD and the conclusion of the first meeting. Furthermore, the layout of this 2CD was modified to comply with the standard OIML layout. The main modification cause by this "new" layout is the deletion of heading "requirement" and "purpose" and a deep modification of the numbering of the document. The layout is not "change tracked" for facilitating the reading.

Contents

Foreword

The International Organization of Legal Metrology (OIML) is a worldwide, intergovernmental organization whose primary aim is to harmonize the regulations and metrological controls applied by the national metrological services, or related organizations, of its Member States. The main categories of OIML publications are:

- **International Recommendations (OIML R),** which are model regulations that establish the metrological characteristics required of certain measuring instruments and which specify methods and equipment for checking their conformity. OIML Member States shall implement these Recommendations to the greatest possible extent;

- **International Documents (OIML D),** which are informative in nature and which are intended to harmonize and improve work in the field of legal metrology;

- **International Guides (OIML G),** which are also informative in nature and which are intended to give guidelines for the application of certain requirements to legal metrology; and

- **International Basic Publications (OIML B),** which define the operating rules of the various OIML structures and systems.

OIML Draft Recommendations, Documents and Guides are developed by Technical Committees or Subcommittees which comprise representatives from the Member States. Certain international and regional institutions also participate on a consultation basis. Cooperative agreements have been established between the OIML and certain institutions, such as ISO and the IEC, with the objective of avoiding contradictory requirements. Consequently, manufacturers and users of measuring instruments, test laboratories, etc. may simultaneously apply OIML publications and those of other institutions.

International Recommendations, Documents, Guides and Basic Publications are published in English (E) and translated into French (F) and are subject to periodic revision.

Additionally, the OIML publishes or participates in the publication of **Vocabularies (OIML V)** and periodically commissions legal metrology experts to write **Expert Reports (OIML E)**. Expert Reports are intended to provide information and advice, and are written solely from the viewpoint of their author, without the involvement of a Technical Committee or Subcommittee, nor that of the CIML. Thus, they do not necessarily represent the views of the OIML.

This publication - reference OIML D XX, edition 200X (E) - was developed by the OIML Technical Subcommittee TC 5/SC 2 *Software*. It was approved for final publication by the International ~~Conference~~ Committee of Legal Metrology in 2008.

OIML Publications may be downloaded from the OIML web site in the form of PDF files. Additional information on OIML Publications may be obtained from the Organization's headquarters:

Bureau International de Métrologie Légale
11, rue Turgot - 75009 Paris - France
Telephone: 33 (0)1 48 78 12 82
Fax: 33 (0)1 42 82 17 27
E-mail: biml@oiml.org
Internet: www.oiml.org

General Requirements
for Software Controlled Measuring Instruments

1 INTRODUCTION

The primary aim of this International Document is to provide the OIML technical committees and subcommittees with guidance for establishing appropriate requirements for software-related functionality in measuring instruments covered by OIML Recommendations.

Furthermore, this International Document can provide guidance to OIML Member States in the implementation of OIML Recommendations in their national laws.

2 SCOPE AND FIELD OF APPLICATION

2.1 This International Document specifies the general requirements applicable to software related functionality in measuring instruments and gives guidance for verifying the compliance of an instrument with these requirements.

2.2 This Document shall be taken into consideration by the OIML technical committees and subcommittees as a basis for establishing particular software requirements and procedures to be specified in OIML Recommendations applicable to particular categories of measuring instruments (hereafter in brief: relevant OIML Recommendation).

2.3 The instructions given in this document apply only to software-controlled measuring instruments or electronic devices.

Notes:

(1) This Document does not cover all of the technical requirements specific for that kind of measuring instrument; these requirements are to be given in the relevant OIML Recommendation, e.g. for weighing instruments, water meters;

(2) This Document addresses some aspects concerning data security. In addition, national regulations for this area have to be considered;

(3) As software controlled devices are always electronic, it is necessary to consider OIML D11 (“General requirements for electronic measuring instruments”) as well.

3 TERMINOLOGY

Some of the definitions used in this International Document are in conformity with the International vocabulary of basic and general terms in metrology (VIM) [1], with the OIML International Document General requirements for electronic measuring instruments (OIML D 11:2004) and several ISO/IEC International Standards. For the purpose of this International Document, the following definitions and abbreviations apply.

3.1 General terminology

3.1.1 Acceptable solution

A design or a principle of a software module or hardware unit, or of a feature that is considered to comply with a particular requirement. An acceptable solution provides an example of how a particular requirement may be met. It does not prejudice any other solution that also meets the requirement.

3.1.2 Audit trail

A continuous data file containing an information record of events like e.g. the changes to the values of the parameters of a device, of updates of the software or other activities that are legally relevant and may influence metrological characteristics.

Every information record has a time stamp.

3.1.3 Authentication

Checking of the declared or alleged identity of a user, process, or device (e.g. checking that a downloaded software originates from the owner of the type approval certificate).

3.1.4 Authenticity

Result of the process of authentication (passed or failed).

3.1.5 Checking facility [D 11, 3.18]

A facility that is incorporated in a measuring instrument and which enables significant faults to be detected and acted upon.

Note:

> "Acted upon" refers to any adequate response by the measuring instrument (luminous signal, acoustic signal, prevention of the measurement process, etc.).

3.1.6 Closed network

A network of a fixed number of participants with a known identity, functionality and location (see also *Open network*).

3.1.7 Commands

Commands may be a sequence of electrical (optical, electromagnetic, etc.) signals on input interfaces or codes in data transmission protocols. They can be generated by the software of the measuring instrument / electronic device / sub-assembly (software commands) or generated by the user through the user interface of the measuring instrument (user commands).

3.1.8 Communication

Exchange of information between two or more units (as software modules, electronic devices, sub-assemblies…) according to specific rules.

3.1.9 Communication interface

An electronic, optical, radio or other technical interface that enables information to be passed between components of measuring instruments (e.g. electronic devices) or sub-assemblies.

3.1.10 Cryptographic certificate

A data set containing the public key belonging to a measuring instrument or a person plus a unique identification of the subject like e.g. serial number of the measuring instrument or name or Personal Identification Number (PIN) of the person. The data set is signed by a trustworthy institution with an electronic signature. The assignment of a public key to a subject can be verified by using the public key of the trustworthy institution and decrypting the signature of the certificate.

3.1.XX Cryptographic means

a) *Encryption* of data by the sender (storing or transmitting program) and *decryption* by the receiver (reading program) with the purpose of hiding information from unauthorised persons.

b) *Electronic signing* of data with the purpose to enable the receiver or user of the data to verify the origin of the data i.e. to prove their authenticity.

Note:

> For electronic signing a public key system is used in general, i.e. the algorithm needs a pair of keys where only one has to be kept secret; the other may be public.
>
> The sender (the sending or storing program) is generating a hash code (see 3.1.23) of the data and encrypts it with his *secret key*. The result is the signature. The receiver (the receiving or reading program) decrypts the signature with the *public key* of the sender and compares the result with the actual hash code of the data. In case of equality, the data are authentic.
>
> The receiver may require a cryptographic certificate of the sender (see 3.1.10) to be sure about the authenticity of the public key.

3.1.11 Data domain

~~It represents parameters, variables, stacks or registers, which are used by programs to keep values of data.~~ Location in memory that each program needs for processing data. Depending on the kind of the programming language used, this location is defined by hardware addresses or by symbolic names (variable names). The size of the smallest addressable domain is typically one byte, but the size is nearly not limited: it ranges from 1 bit (e.g. a flag of a register) to arbitrary data structures just as big as the needs of the programmer are.

Data domains may belong to one *software module* only or to several. For high level languages (like JAVA, C/C++ etc.) it is easy to separate the data domain of one software module against access by any other by means of the language.

3.1.12 Device-specific parameter

Legally relevant parameter with a value that depends on the individual instrument. Device-specific parameters comprise adjustment parameters (e.g. span adjustment or other adjustments or corrections) and configuration parameters (e.g. maximum value, minimum value, units of measurement, etc).

3.1.13 Durability [D 11, 3.17]

The ability of the measuring instrument to maintain its performance characteristics over a period of use.

3.1.14 Electronic measuring instrument [D 11, 3.1]

A measuring instrument intended to measure an electrical or non-electrical quantity using electronic means and/or equipped with electronic devices.

Note:

For the purpose of this Document, ancillary equipment, as far as subject to metrological control, is considered to be part of the measuring instrument.

3.1.15 Electronic device [D 11, 3.2]

A device employing sub-assemblies and performing a specific function. Electronic devices are usually manufactured as a separate unit and are capable of being tested independently.

Notes:

(1) An electronic device may be a complete measuring instrument (for example: counter scale, electricity meter) or a part of a measuring instrument (for example: printer, indicator);

(2) An electronic device can be a module in the sense this term is used in the OIML Publication B3 “The OIML Certificate system for Measuring Instruments” [2].

3.1.16 Error (of indication) [VIM 5.20, D11, 3.5]

Indication of a measuring instrument minus a true value of the corresponding input quantity [1]).

3.1.17 Evaluation [VIM]

{Definition to be added according to that in the latest VIM.}

3.1.18 Event

~~-~~Occurrence of unwanted incidence, e.g.fault~~Unwanted incidence of e.g. a fault~~ or~~ an occurrence of an~~ operation with ~~that has an~~ influence on ~~a~~the measuring instrument e.g. a modification of ~~a preset value of the measuring instrument's~~ parameter~~s~~ or updat~~eing~~ of ~~its~~ software module~~s~~.

3.1.19 Event counter

A non resetable counter that increments each time an event occurs.

[1]) VIM is being revised. In the present draft, “*error of indication*” is defined in A6

3.1.20 Executable code

Executable code is a file installed on the computer system of the measuring instrument, electronic device, or sub-assembly (EPROM, hard disk,). This code is interpreted by the microprocessor and transposed into certain logical, arithmetical, decoding, or data transporting operations.

3.1.21 Fault [D 11, 3.9]

A defect or an undesired change of data that has an impact on the properties or functions of the measuring instrument or causes an error of indication greater than the MPE.

3.1.22 Fixed legally relevant software part

Part of the legally relevant software that is and remains identical in the executable code to that of the approved type [2]).

3.1.23 Hash function [ISO/IEC 9594-8: 2001]

A (mathematical) function which maps values from a large (possibly very large) domain into a smaller range. A “good” hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range.

3.1.24 Integrity of programs, data, or parameters

Assurance that the programs, data, or parameters have not been subjected to any unauthorised or unintended changes while in use, transfer, storage, repair or maintenance.

3.1.25 Interface [ISO 2382-9]

A shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges, and other characteristics of the units, as appropriate.

3.1.26 Intrinsic error [VIM 5.24, D11, 3.7]

The error of a measuring instrument, determined under reference conditions[3]).

3.1.27 Legally relevant

Software/hardware/data or part of the software/hardware/data of a measuring instrument which interferes with properties regulated by legal metrology, e.g. the accuracy of the measurement ~~regulated by legal metrology~~ or ~~with~~ the correct functioning of the measuring instrument.

~~Note:~~

> ~~It is advisable that TC’s and SC’s specify in the relevant Recommendation software / hardware / data or part of the software/hardware/data that are legally relevant.~~
>
> ~~National regulation may prescribe that a specific software/hardware/data or part of the software/hardware/data is legally relevant.~~

[2]) This part is responsible for monitoring the software update (loading software, authentication, integrity checking, installation and activation).

[3]) VIM is being revised. In the present draft, “*intrinsic error*” is defined in A13.

3.1.28 Legally relevant parameter

Parameter of a measuring instrument, electronic device, or a sub-assembly subject to legal control. The following types of legally relevant parameters can be distinguished: *type-specific parameters* and *device-specific parameters*.

3.1.29 Legally relevant software part

The part of all *software modules* of a measuring instrument, electronic device, or sub-assembly that is legally relevant.

3.1.30 Maximum permissible error (of a measuring instrument) [VIM 5.21, D11, 3.6]

Extreme values of an error permitted by specifications, regulations, etc. for a given measuring instrument.

3.1.31 Measuring instrument [VIM, 4.1]

Device intended to be used to make measurements, alone or in conjunction with supplementary device(s).

3.1.32 Non-interruptible / interruptible measurement

A non-interruptible measurement is a cumulative continuous measuring process with no definite end. The measuring process cannot be stopped and continued again by a user or operator without inadmissibly disturbing the measurement or the supply with goods or energy.

If the cumulative measurement of a quantity of a substance can be stopped easily and rapidly during normal operation – not only in case of emergency – without falsifying the measurement result, it is called interruptible.

3.1.33 Open network

A network of arbitrary participants (electronic devices with arbitrary functions). The number, identity and location of a participant can be dynamic and unknown to the other participants (see also *Closed network*).

3.1.34 Performance [D 11, 3.16]

The ability of the measuring instrument to accomplish its intended functions.

3.1.35 Program code

Source code or *executable code*.

3.1.36 Sealing

~~To set a special protection to serve as an indicator for the case of unauthorised access to the device's hardware or software part. It can be achieved by hardware, software or a combination of both.~~Means intended to protect the measuring instrument against any unauthorized modification, readjustment, removal of parts, software, etc It can be achieved by hardware, software or a combination of both.

~~3.1.37 Special protection~~

~~Hardware of software protection against intentional, unintentional and accidental changes to the measuring instrument's metrological properties.~~

3.1.38 Securing

To prevent unauthorised access to the device's hardware or software part.

3.1.39 Software

Generic term comprising program code, data, and parameters.

3.1.40 Software examination

Technical operation that consists of determination of one or more characteristics of a software according to the specific procedure (e.g. analysis of technical documentation or running the program under controlled conditions).

3.1.41 Software identification

A sequence of readable characters (e.g. version number, checksum) that is inextricably linked to the software or *software module* under consideration. It can be checked at an instrument in use.

3.1.42 Software interface

It consists of program code and a dedicated data domain. It receives, filters, or transmits data between *software modules (not necessarily legally relevant)*.

3.1.43 Software module [similar IEC 61508-4, 3.3.7]

Logic entities like programs, subroutine, libraries, objects including their *data domains* that may be in relationship with other entities. The software of measuring instruments, electronic devices or sub-assemblies consists of one or more software modules.

3.1.44 Software protection

Securing of measuring instrument software or data domain by hardware or software implemented seal. The seal has to be removed, damaged or broken to get access to change software.

3.1.45 Software separation

Software in measuring instrument/electronic devices/sub-assembly can be divided into a *legally relevant part* and a legally non-relevant part. These parts communicate via a *software interface*.

3.1.46 Source code

Computer program written in a form (programming language) which is legible and editable. Source code is compiled or interpreted into *executable code*.

3.1.47 Storage device

Storage used for keeping measurement data ready after completion of the measurement for later legally relevant purposes (e.g. the conclusion of a commercial transaction).

3.1.48 Sub-assembly [D 11, 3.3]

A part of an electronic device employing electronic components and having a recognisable function of its own.

Examples:

> Amplifiers, comparators, power converters...

3.1.49 Test [D 11, 3.20]

A series of operations intended to verify the compliance of the equipment under test (EUT) with specified requirements.

3.1.50 Time stamp

Unique monotonically increasing time value e.g. in seconds or a date and time string labelling an event. Makes it possible to distinguish an event from others and determine the order of occurrence.

3.1.51 Transmission of measurement data

Transmission of measurement data via communication networks or other means to a distant electronic device where they are further processed and/or used for legally regulated purposes.

3.1.52 Type-specific parameter

Legally relevant parameter with a value that depends on the type of instrument only. Type-specific parameters are part of the legally relevant software.

Example:

> Considering a measuring system of liquids other than water, the range of cinematic viscosity of a turbine is a type specific parameter fixed by the type approval of the turbine. All the manufactured turbines of the same type have the same range of viscosity.

3.1.53 Universal computer

Computer that is not constructed for a specific purpose but can be adapted to the metrological task by software. In general this software is founded on an operating system that permits loading and execution of software for specific purposes.

3.1.54 User interface

An interface that enables information to be interchanged between a human user and the measuring instrument or its hardware or software components, as e.g. switches, keyboard, mouse, display, monitor, printer, touch-screen, a software window on a screen including the software that generates it.

3.1.55 Validation [derived from ISO/IEC 14598 and IEC 61508-4]

Confirmation by examination and provision of objective evidence (i.e. information that can be proved true, based on facts obtained from observations, measurement, test, etc.) that the particular requirements for the specific intended use are fulfilled. In the present case the related requirements are those of this Document.

3.1.56 Verification [VIML, 2.13]

Procedure (other than type approval) which includes the examination and marking and/or issuing of a verification certificate that ascertains and confirms that the measuring instrument complies with the statutory requirements [4]).

3.2 Abbreviations

EUT	Equipment Under Test
IEC	International Electrotechnical Commission
I/O	Input / Output (refers to ports)
ISO	International Organisation for Standardisation
IT	Information Technology
MPE	Maximum Permissible Error
N/A	Not applicable
OIML	International Organisation of Legal Metrology
PCB	Printed Circuit Board
PIN	Personal identification number

4 INSTRUCTIONS FOR USE OF THIS DOCUMENT IN DRAFTING OIML RECOMMENDATIONS

4.1 Provisions of this document apply only to new OIML Recommendations and OIML Documents under revision. The TC's and SC's should use this guidance document to establish software related requirements in addition to the other technical and metrological requirements of the relevant OIML Recommendation.

4.2 All normative documents are subject to revision, and the users of this Document are encouraged to investigate the possibility of applying the most recent editions of the normative documents.

4.3 It is the objective of this Document to provide the TCs or SCs responsible for elaboration of OIML Recommendations with a set of requirements – partly with different levels – that are suitable to cover the demands of all kinds of measuring instruments and all areas of application. The TC or SC shall determine which level for protection or conformity issues or validation intensity is suitable and how to incorporate the relevant portions of this document into their OIML Recommendation. In Chapter 8 some aid is given for performing this task.

[4]) Note: Differing definition from other standards like e.g. ISO/IEC 14598, clause 4.23 or IEC 61508-4, clause 3.8.1.

5 REQUIREMENTS FOR MEASURING INSTRUMENTS WITH RESPECT TO THE APPLICATION OF SOFTWARE

5.1 General requirements

At the time of publication of this Document the general requirements represent the state of the art in information technology (IT). They are in principle applicable to all kinds of software controlled measuring instruments, electronic devices and sub-assemblies and should be considered in all OIML Recommendations. In contrast to these ~~elementary~~ general requirements the requirements specific for configuration ~~ones~~ (5.2) deal with technical feature that are not common for some kinds of instruments or in some areas of application.

Notation: (I) – Technical solution acceptable in case of normal severity level

(II) – Technical solution acceptable in case of raised severity level (see 8)

5.1.1 Software identification

Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be ~~only~~ dedicated ~~for~~ to the legal purpose.

The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via communication interface in order to be displayed/printed on another sub-assembly/electronic device.

As an exception ~~for non-interruptible measurements~~ an imprint of the software identification on the instrument/electronic device shall be an acceptable solution ~~under the following circumstances~~if it satisfies the following conditions:

(1) The user interface does not have any control capability to activate the indication of the software identification on the display or the display does not allow technically showing the identification of the software (analogue indicating device or electromechanic~~al~~ counter);

(2) The instrument/electronic device does not have an interface to communicate the software identification;

(3) After production of the instrument/electronic device a change of the software is not possible or only possible, if also the hardware or a hardware component is changed;

The manufacturer of the hardware or the concerned hardware component is responsible that the software identification is correctly marked on the concerned instrument/electronic device.

The software identification and the means of identification shall be stated in the type approval certificate.

The relevant OIML Recommendation should allow or disallow this exception.

Note:

Each measuring instrument in use has to conform to the approved type. The software identification enables surveillance personnel and persons affected by the measurement to determine whether the instrument under consideration is ~~conformant~~conformable.

Example:

(I) The software contains a textual string or a number, unambiguously identifying the installed version. This string is transferred to the display of the instrument when a button is pressed, when the instrument is switched on, or cyclically controlled by a timer.

A version number may have the following structure A.Y.Z. If we consider a flow computer, the number A will represent the version of the core software that is counting pulses, the number Y will represent the version of the conversion function (none, at 15 °C, at 20 °C) and the number Z will represent the language of the user interface.

(II) The software calculates a checksum of the executable code and presents the result as the identification instead of or additional to the string in (I). The checksum algorithm shall be a normalized algorithm e.g. the CRC16 algorithm is an acceptable solution for this calculation.

Solution (II) is suitable, if increased conformity is required (see 5.2.5 (d) and 8.).

5.1.2 Correctness of algorithms and functions

The measuring algorithms and functions of an electronic device shall be appropriate and functionally correct for the given application and device type (accuracy of the algorithms, price calculation according to certain rules, rounding algorithms …).

The measurement result and accompanying information required by specific OIML Recommendations or national legislation shall be displayed or printed correctly.

It shall be possible to examine algorithms and functions either by metrological tests, software tests or software examination (as described in 6.3).

5.1.3 Software protection

5.1.3.1 Prevention of misuse

A measuring instrument shall be constructed in a way that possibilities for unintentional accidental or intentional misuse are minimal. In the framework of this OIML Document, this applies especially to the software. The presentation of the measurement results should be unambiguous for all parties affected.

Note:

Software controlled instruments are often complex in their functionality. The user needs good guidance for correct use and for achieving correct measurement results.

Example:

The user is guided by menus. The legally relevant functions are combined to one branch in this menu. If measurement values might be lost by an action, the user should be warned and requested to do another action before the function is executed. See also 5.2.2.

5.1.3.2 Fraud protection

5.1.3.2.a The legally relevant software shall be secured against unauthorized modification, loading, or changes by swapping the memory device. ~~Technical means, i~~In addition to mechanical sealing, technical means may be necessary to secure measuring instrument having an operating system or an option to load software.

Note:

When the software is stored on an inviolable memory device (which data are unalterable; ~~(~~e.g. sealed ~~masked~~ ROM “Read Only Memory”) the needs for technical means are accordingly reduced.

Example:

(I)/(II) The housing containing the memory devices is sealed or the memory device is sealed on the PCB).

(II) If a rewritable device is used, the write-enable input is inhibited by a switch that can be sealed. The circuit is designed in a way that the write-protection cannot be cancelled by a short-circuit of contacts.

(I) A measuring system consists of two sub-assemblies, one containing the main metrological functions incorporated in a housing that can be sealed. The other sub-assembly is a universal computer with an operating system. Some functions like indication are located in the software of this computer. One relatively easy manipulation – especially, if a standard protocol is used for communication between both software parts – could be swapping the software on the universal computer. This manipulation can be inhibited by simple cryptographic means, e.g. encryption of the data transfer between the sub-assembly and the universal computer. The key for decryption is hidden in the legally relevant program of the universal computer. Only this program knows the key and is able to read, decrypt and use the measurement values. Other programs cannot be used for this purpose as they cannot decrypt the measurement values (see also example in 5.2.1.2.d).

5.1.3.2.b Only documented functions are allowed to be activated by the user interface. The user interface shall be realised in a way that it does not facilitate fraudulent use. The presentation of information shall comply with5.2.2.

Note:

The examiner decides whether all of these documented commands are acceptable.

Example:

(I)/(II) All inputs from the user interface are redirected to a program that filters incoming commands. It only allows and lets pass the documented ones and discards all others. This program or software module is part of the legally relevant software.

5.1.3.2.c Parameters that fix legally relevant characteristics of the measuring instrument shall be secured against unauthorised modification. If necessary for the purpose of verification, the current parameter settings shall be able to be displayed or printed.

Note:

Device-specific parameters may be adjustable or selectable only in a special operational mode of the instrument. They may be classified as those that should be secured (unalterable) and those that may be accessed (settable parameters) by an authorised person, e.g. instrument owner or product vendor.

Type-specific parameters have identical values for all specimen of a type. They are fixed at type approval of the instrument.

Example:

(I)/(II) Device specific parameters to be secured are stored in a non-volatile memory. Its write-enable input is inhibited by a switch that can be sealed~~ or may be software controlled~~.

Refer to examples 5.1.3.2.d (1) to (3) in this chapter.

5.1.3.2.d Software protection comprises mechanical sealing and electronic or cryptographic means making an unauthorized intervention impossible or evident.

Example:

(1) (I) Electronic sealing. The metrological parameters of an instrument can be input and adjusted by a menu item. The software recognises each change and increments an event counter with each event of this kind. This event counter value can be indicated. The initial value of the event counter has to be registered. If the indicated value differs from the registered one, the instrument is in an unverified state (equivalent to a broken seal);

(2) (I)/(II) The software of a measuring instrument is constructed such (see Example 5.1.3.2.a) that there is no way to access the parameters and legally relevant configuration but via a switch protected menu. This switch is mechanically sealed in the inactive position; this makes modification of parameters and legally relevant configuration impossible.

To access the parameters and configuration, the switch has to be switched, inevitably breaking the seal by doing so;

(3) (II) The software of a measuring instrument is constructed such (see Example (a)) that there is no way to access the parameters and legally relevant configuration but by authorised persons. If a person wants to enter the parameter menu item he has to insert his smart card containing a PIN as part of a cryptographic certificate. The software of the instrument is able to verify the authenticity of the PIN by the certificate and allows entering the parameter menu item. The access is registered in an audit trail including the identity of the person (or at least of the smart card used).

Level (II) of the examples for acceptable technical solutions is appropriate, if increased protection against fraud is necessary (see 8).

5.1.4 Support of hardware features

5.1.4.1 Support of fault detection

The relevant OIML Recommendation may require fault detection functions for certain faults of the instrument (addressed in D11 (5.1.2 (b) and 5.3)). In this case, the manufacturer of the instrument shall be required to design checking facilities into the software or hardware parts or provide means by which the hardware parts can be supported by the software parts of the instrument.

If software is involved in fault detection, an appropriate reaction is required. The relevant OIML Recommendation may prescribe that the instrument /electronic device is deactivated or an alarm / report is generated in case a fault condition is detected.

The documentation submitted for type approval shall contain a list of faults that are detected by the software and its exp~~e~~e~~p~~cted reaction. ~~i~~If necessary for understanding, a description of the detecting algorithm.

Example:

(I)/(II) On each start-up the legally relevant program calculates a checksum of the program code and legally relevant parameters. The nominal value of these checksums has been calculated in advance and stored in the instrument. If the calculated and stored values don't match, the program stops execution.

If the measurement is not interruptible the checksum is calculated cyclically controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and registers the time of the event in a log if it exists.

An acceptable checksum algorithm is CRC16.

5.1.4.2 Support of durability protection

It is the manufacturer’s choice to realise durability protection facilities addressed in D11 (5.1.3 (b) and 5.4) in software or hardware or let hardware facilities be supported by software. The relevant OIML Recommendation may recommend appropriate solutions.

If software is involved in durability protection, an appropriate reaction is required. The relevant OIML Recommendation may prescribe that the instrument / electronic device is deactivated or an alarm / report is generated in case durability is detected being jeopardised.

Example:

(I)/(II) Some kinds of measuring instruments need an adjustment after a prescribed time interval for guaranteeing durability of measurement. The software gives a warning when the maintenance interval has elapsed and even stops measuring, if it has been exceeded for a certain time interval.

5.2 Requirements specific for Configurations

The requirements given in this section are based on typical technical solutions in IT though they might not be common in all areas of legal applications. Following these requirements technical solutions are possible that show the same degree of security and conformity to a type as instruments that are not software controlled.

The following specific requirements are needed when certain technologies are employed in measuring systems. They have to be considered in addition to those described in 5.1.

Notation: (I) – Technical solution acceptable in case of normal severity level

(II) – Technical solution acceptable in case of raised severity level (see 8)

5.2.1 Specifying and separating relevant parts and specifying interfaces of parts

Metrologically critical parts of a measuring system – whether software or hardware parts – shall not be inadmissibly influenced by other parts of the measuring system.

This requirement applies, if the measuring instrument (or electronic device or sub-assembly) has interfaces for communicating with other electronic devices, with the user, or with other software parts besides the metrologically critical parts within a measuring instrument (or electronic device or sub-assembly).

5.2.1.1 Separation of electronic devices and sub-assemblies

5.2.1.1.a Sub-assemblies or electronic devices of a measuring system that perform legally relevant functions shall be identified, clearly defined, and documented. They form the legally relevant part of the measuring system.

Note:

The examiner decides whether this part is complete and whether other parts of the measuring system may be excluded from further evaluation.

Example:

(1) (I)/(II) An electricity meter is equipped with an optical interface for connecting a electronic device to read out measurement values. The meter stores all relevant quantities and keeps the values available for being read out for a sufficient time span. In this system only the electricity meter is the legally relevant device. Other legally non-relevant devices may exist and be connected to the interface of the instrument provided requirement 5.2.1.1.b is fulfilled. Securing of the data transmission itself (see 5.2.3) is not required.

(2) (I)/(II) A measuring system consists of the following sub-assemblies:

- a digital sensor calculating the weight or volume;

- a universal computer calculating the price;

- a printer printing out the measurement value and the price to pay.~~;~~

~~- a merchandise management system.~~

All sub-assemblies are connected by a local area network. In this case digital sensor, universal computer, and printer are legally relevant sub-assemblies, ~~the~~ an optionally connected merchandise system is not. The legally relevant sub-assemblies have to fulfil requirement 5.2.1.1.b and – because of the transmission via the network – also requirements contained in 5.2.3. There are no requirements on the merchandise management system.

5.2.1.1.b It shall be shown that the relevant functions and data of sub-assemblies and electronic devices cannot be inadmissibly influenced by commands received via the interface.

This implies that there is an unambiguous assignment of each command to all initiated function or data change in the sub-assembly or electronic device.

Note:

If “legally relevant” sub-assemblies or electronic devices interact with other “legally relevant” sub-assemblies or electronic devices, refer to 5.2.3.

Example:

(1) (I)/(II) The software of the electricity meter (see example (1) of 5.2.1.1.a above) is able to receive commands for selecting the quantities wanted. It combines the measurement value with additional information – e.g. time stamp, unit – and sends this data set back to the requesting device. The software only accepts commands for selection of valid allowed quantities and discards any other command sending back only an error message. There may be securing means for the contents of the data set but they are not required, as ~~the sink of~~ the transmitted data set is not subject to legal control.

(2) (I)/(II)Inside the housing that can be sealed there is a switch that defines the operating mode of the electricity meter: one switch setting indicates the verified mode the other the non-verified mode(other securing means except a mechanical seal are possible; see examples 5.1.3.2.a/.d. When interpreting received commands the software checks the position of the switch: In the non-verified mode the command set that the software accepts is extended compared to the mode described above; e.g. it may be possible to adjust the calibration factor by a command that is discarded in the verified mode.

5.2.1.2 Separation of software parts

OIML TC’s and SC’s may specify in the relevant Recommendation software / hardware / data or part of the software/hardware/data that are legally relevant.

National regulation may prescribe that a specific software/hardware/data or part of the software/hardware/data is legally relevant.

5.2.1.2.a All software modules (programs, subroutines, objects etc.) that perform legally relevant functions or that contain legally relevant data domains form the legally relevant software part of a measuring instrument (electronic device or sub-assembly). The conformity requirement applies to this part (see 5.2.5) and it shall be made identifiable as described in 5.1.1.

If the separation of the software is not possible or needed, the software is legally relevant as a whole.

Example:

(I) A measuring system consists of several digital sensors connected to a personal computer that displays the measurement values. The legally relevant software on the personal computer is separated from the legally non-relevant parts by compiling all procedures realising legally relevant functions into a dynamically linkable library. One or several legally non-relevant applications may call program procedures in this library. These procedures receive the measurement data from the digital sensors, calculate the measurement result, and display it in a software window. When having finished the legally relevant functions, control is given back to the legally non-relevant application.

5.2.1.2.b If the legally relevant software part communicates with other software parts, a software interface shall be defined. All communication shall be performed exclusively via this interface. The legally relevant software part and the interface shall be clearly documented. ~~This implies that a~~All legally relevant functions and data domains of the software ~~are~~ shall be described to enable a type approval authority to decide on correct software separation.

The interface consists of program code and dedicated data domains. Defined coded commands or data are exchanged between the software parts by storing to the dedicated data domain by one software part and reading from it by the other. Writing and reading program code is part of the software interface. The data domain forming the software interface including the code that exports from the legally relevant part to the interface data domain and the code that imports from the interface to the legally relevant part shall be clearly defined and documented. The declared software interface shall not be circumvented.

The manufacturer is responsible for respecting these constraints. Technical means (like sealing) for preventing a program from circumventing the interface or programming hidden commands are not possible. The programmer of the legally relevant software part as well as the programmer of the legally non-relevant part should be instructed by the manufacturer about these requirements.

5.2.1.2.c There shall be an unambiguous assignment of each command to all initiated function or data change in the legally relevant part of the software. Commands that communicate through the software interface shall be declared and documented. Only documented commands are allowed to be activated through the software interface. The manufacturer shall state the completeness of the documentation of commands.

Example:

(I) In the example described in 5.2.1.2.a the software interface is realised by the parameters and return values of the procedures in the library. No pointers to data domains inside the library are returned. The definition of the interface is fixed in the compiled legally relevant library and cannot be changed by any application. It is not impossible to circumvent the software interface and address data domains of the library directly; but this is no good programming practice, is rather complicated, and may be classified as hacking.

5.2.1.2.d Where legally relevant software has been separated from non-relevant software, the legally relevant software shall have priority using the resources over non-relevant software. The measurement task (realised by the legally relevant software part) must not be delayed or blocked by other tasks.

The manufacturer is responsible for respecting these constraints. Technical means for preventing a legally non-relevant program from disturbing legally relevant functions ~~have to~~shall be provided. The programmer of the legally relevant software part as well as the programmer of the legally non-relevant part should be instructed by the manufacturer about these requirements.

Examples:

(1) (I) In the example 5.2.1.2.a/.c the legally non-relevant application controls the start of the legally relevant procedures in the library. Omitting a call of these procedures would of course inhibit the legally relevant function of the system. Therefore the following provisions have been taken in the example system to fulfil the requirement 5.2.1.2.d: The digital sensors send the measurement data in encrypted form. The key for decryption is hidden in the library. Only the procedures in the library know the key and are able to read, decrypt, and display measurement values. If the application programmer wants to read and process measurement values, he is forced to use the legally relevant procedures in the library that perform all legally required functions as a side effect when being called. The library contains procedures that export the decrypted measurement values allowing the application programmer to use them for his own needs after the legally relevant processing has been finished.

(2) (I)/(II) The software of an electronic electricity meter is reading raw measurement values from an analogue-digital converter (ADC). For the correct calculation of the measurement values the delay between the “data ready” event from the ADC to finishing buffering of the measurement values is crucial. The raw values are read by an interrupt routine initiated by the “data ready” signal. The instrument is able to communicate via an interface with other electronic devices in parallel served by another interrupt routine (legally non-relevant communication). Interpreting the requirement 5.2.1.2 for such a configuration, it follows that the priority of the interrupt routine for processing the measurement values shall be higher than that of the communication routine.

Examples from 5.2.1.2.a to 5.2.1.2.c and 5.2.1.2.d (1) are acceptable as a technical solution only for a normal severity level (I). If increased protection against fraud or increased conformity is necessary (see 8), software separation alone is not sufficient and additional means are demanded or the whole software should be considered as under legal control.

5.2.2 Shared indications

A display or printout may be employed for presenting both information from the legally relevant part of software and other information. The contents and layout are specific for the kind of instrument and area of application and have to be defined in the relevant Recommendation. However, if the indication is realised using a multiple windows user interface, the following requirement applies:

Software that realises the indication of measurement values and other legally relevant information belongs to the legally relevant part. The window containing these data shall have highest priority i.e. it shall not be deleted by other software or overlapped by windows generated by other software or minimised or made invisible as long as the measurement is running and the presented results are needed for the legally relevant purpose.

Example:

(I) On a system described in the examples 5.2.1.2.a.a to 5.2.1.2.d the measurement values are displayed in a separate software window. The means described in 5.2.1.2.d guarantee that only the legally relevant program part can read the measurement values. On an operating system with a multiple windows user interface an additional technical means is taken to meet the requirement in 5.2.2: The window displaying the legally relevant data is generated and controlled by procedures in the legally relevant dynamically linkable library (see 5.2.1.2). During measurement these procedures check cyclically that the relevant window is still on top of all other windows that currently exist and bring it on top, if not.

If increased protection against fraud is necessary (II), a printout as an indication alone may not be suitable. There should exist a sub-assembly with increased securing means that is able to display the measurement values.

The use of a universal computer is not appropriate as part of a measuring system if increased protection against fraud is necessary (II). Additional hardware components to the universal computer are necessary to guarantee a sufficient level of protection.

5.2.3 Storage of data, transmission via communication systems

If measurement values are used at another place than the place of measurement or at a later time than the time of measurement they possibly have to leave the measuring instrument (electronic device, sub-assembly) and be stored or transmitted in an insecure environment before they are used for legal purposes. In this case the following requirements apply:

5.2.3.1 The measurement value stored or transmitted shall be accompanied by all relevant information necessary for future legally relevant use.

Example:

(I)/(II) A data set .may include the following entries:

- measurement value including unit;
- time stamp of measurement (see 5.2.3.6);
- place of measurement or identification of the measuring instrument that was used for the measurement;
- unambiguous identification of the measurement e.g. consecutive numbers enabling assignment to values printed on an invoice.

~~The time stamp is read from the clock of the device. Depending on the kind of instrument, or area of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the severity level to be applied (see 5.1.3.2.c).~~

~~The internal clock of a stand-alone measuring instrument tends to have a large uncertainty because there is no means to synchronize it with the global clock. But if the information of time of measurement is necessary for a specific field of application, the internal clock of the measuring instrument shall be reliable as far as possible. Specific means shall be employed to get a reliable clock, such as a specific flag (as an incremental counter) to enhance the time information.~~

5.2.3.2 The data shall be protected by software means to guarantee authenticity, integrity and, if necessary correctness of the information of the time of measurement. The software that displays or further processes the measurement values and accompanying data shall check time of measurement, authenticity, and integrity of the data after having read them from the insecure storage or after having received them from an insecure transmission channel. If an irregularity is detected, the data shall be discarded or marked unusable.

Software modules that prepare data for storing or sending, or that check data after reading or receiving belong to the legally relevant software part.

Example:

(I) The program of the sending device calculates a checksum of the data set (algorithm CRC16, CRC32…) and appends it to the dataset. It uses a secret initial value for this calculation instead of the value given in the standard. This initial value is employed as a key and stored as a constant in the program code. The receiving or reading program also has stored this initial value in its program code. Before using the data set the receiving program calculates the checksum and compares it with that stored in the data set. If both values match, the data set is not falsified. Else the program assumes a falsification and discards the data set.

5.2.3.3 For a high protection level it is necessary to apply cryptographic methods. Confidential keys employed for that purpose shall be kept secret and secured in the measuring instruments, electronic devices, or sub-assemblies involved. Means shall be provided that these keys can only be input or read if a seal is broken.

Example:

(II) The storing or sending program generates an “electronic signature” by first calculating a hash value[5]) and secondly encrypting the hash value with the secret key of a public key system[6]). The result is the signature. It is appended to the stored or transmitted data set. The receiver also calculates the hash value of the data set and decrypts the signature appended to the data set with the public key. The calculated and the decrypted values of the hash value are compared. If they are equal, the data set is not falsified (the integrity is proven). To prove the origin of the data set the receiver must know whether the public key really belongs to the sender i.e. the sending device. Therefore the public key is displayed on the display of the measuring instrument and can be registered once e.g. together with the serial number of the device when it is legally verified in the field. If the receiver is sure that he used the correct public key for decryption of the signature, also the authenticity of the data set is proven.

~~Choose level (II) of the example for acceptable technical solutions, if increased protection against fraud is necessary (see 8).~~

5.2.3.4 Automatic storing

5.2.3.4.a When, considering the application, data storage is required, measurement data must be stored automatically when the measurement is concluded, i.e. when the final value used for the legal purpose has been generated.

The storage device must have sufficient permanency to ensure that the data are not corrupted under normal storage conditions. There shall be sufficient memory storage for any particular application.

When the final value used for the legal purpose results from a calculation, all data that are necessary for the calculation must be automatically stored with the final value.

Note:

[5]) Acceptable algorithms: SHA-1, MD5, RipeMD160, or equivalent.

[6]) Acceptable algorithms: RSA (1024 bit key length), Elliptic Curves (160 bit key length) , or equivalent.

Cumulative measurement values like e.g. electrical energy or gas volume have to be updated currently. As always the same data domain (program variable) is used the requirement concerning the storage capacity is not applicable to cumulative measurements.

5.2.3.4.b Stored data may be deleted if either:

- the transaction is settled;
- these data are printed by a printing device subject to legal control.

5.2.3.4.c After Section 5.2.3.4.b requirements are fulfilled and when the storage is full, it is permitted to delete memorized data when both the following conditions are met:

- data are deleted in the same order as the recording order and the rules established for the particular application are respected,
- deletion is carried out either automatically or after a special manual operation.

5.2.3.5 Transmission delay

The measurement shall not be inadmissibly influenced by a transmission delay. If network services become unavailable, no measurement data shall be lost. The measurement process could be stopped to avoid the loss of measurement data.

Example:

(I)/(II) The sending device waits until the receiver has sent an affirmation of correct receipt of the data set. The sending device keeps the data set in a buffer until this affirmation has been received. The buffer may have a capacity for more than one data set, organised as a FIFO[7]) queue.

5.2.3.6 Time stamp

The time stamp shall be read from the clock of the device. Depending on the kind of instrument, or area of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the severity level to be applied (see 5.1.3.2.c).

The internal clock of a stand-alone measuring instrument tends to have a large uncertainty because there is no means to synchronize it with the global clock. But if the information of time of measurement is necessary for a specific field of application, reliability of the internal clock of the measuring instrument shall be ~~reliable as far as possible.~~ enhanced by ~~S~~specific means ~~shall be employed to get a reliable clock,~~ .

Example:

(II) The reliability of the internal quartz-controlled clock device of the measuring instrument is enhanced by redundancy: A timer is incremented by the clock of the microcontroller that is derived from another quartz. When the timer value reaches a preset value, e.g. 1 second, ~~such as~~ a specific flag ~~(as an incremental counter) to enhance the time information~~of the microcontroller is set and an interrupt routine of the program increments a second counter. At the end of e.g. one day the software reads the quartz-controlled clock device~~,~~ and calculates the difference to the seconds counted by the software. ~~.~~If the difference is within

[7]) FIFO: First in – first out

predefined limits, the software counter is reset and the procedure repeats; but if the difference exceeds the limits, the software initiates an appropriate error reaction.

5.2.4 Compatibility of operating systems and hardware, portability

5.2.4.1 The manufacturer shall identify the hardware and software environment that is suitable. ~~Minimal~~ Minimum resources and a suitable configuration (e.g. processor, RAM, HDD, specific communication, version of operating system, etc) which is necessary for correct functioning shall be declared by the manufacturer and stated in the type approval certificate.

5.2.4.2 Technical means shall be provided in the legally relevant software to prevent operation, if the minimal configuration requirements are not met.The system shall be operated only in the environment specified by the manufacturer for its correct functioning.

For example, in case an invariant environment is specified for the correct functioning of the system, means shall be provided to keep the operating environment fixed. This especially applies to universal computer performing legally relevant functions.

Fixing the hardware, operating system, or system configuration of a universal computer or even exclude the usage of an off-the-shelf universal computer has to be considered in the following cases:

- if high conformity is required (see 5.2.5 (d));
- if fixed software is required (e.g. 5.2.6.3.b for traced software update);
- if cryptographic algorithms or keys have to be implemented (see 5.2.3).

5.2.5 Conformity of manufactured devices to the approved type

The manufacturer shall produce devices and the legally relevant software that conform to the approved type and the documentation submitted. There are different levels of conformity demands:

(a) identity of the *legally relevant functions* described in the documentation (6.1) of each device with those of the type (the executable code may differ);

(b) identity of *parts of the legally relevant source code*, and the rest of the legally relevant software complying with (a);

(c) identity of the *whole legally relevant source code*, and

(d) identity of the *whole executable code*.

The relevant Recommendation shall specify which degree of conformity is suitable. This Recommendation can also define a subset from these conformity degrees.

Except for (d) there may be a software part with no conformity requirements, if it is separated from the legally relevant part according to 5.2.1.2.

Means described in 5.1.1 and 5.2.1 shall be provided to make the conformity evident.

Note :

(a), (b) should be applied in case of normal severity level and (c), (d) should be applied in case of raised severity level

5.2.6 Maintenance and re-configuration

Updating the legally relevant software of a measuring instrument in the field should be considered as:

- A modification of the measuring instrument, when exchanging the software with another approved version;

- A repair of the measuring instrument, when re-installing the same version.

A measuring instrument which has been modified or repaired while in service may require initial or subsequent verification, dependant on national regulation

Software which is not necessary for the correct functioning of the measuring instrument does not require verification after being updated.

5.2.6.1 Only versions of legally relevant software that conform to the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also on the kind of instrument under consideration. The following options 5.2.6.2 and 5.2.6.3 are equivalent alternatives. This issue concerns verification in the field. Refer to chapter 7 for additional constraints.

5.2.6.2 Verified Update

The software to be updated can be loaded locally i.e. directly on the measuring device or remotely via a network. Loading and installation may be two different steps (as shown in Fig. 1) or combined to one, depending on the needs of the technical solution. After update of the legally relevant software of a measuring instrument (exchange with another approved version or re-installation) the measuring instrument is not allowed to be employed for legal purposes before a verification of the instrument as described in chapter 7 has been performed and the securing means have been renewed (if not otherwise stated in the relevant OIML Recommendation or in the approval certificate). A person responsible for verification must be on the installation site of the measuring instrument.

5.2.6.3 Traced Update

The software is implemented into the instrument according to the requirements for Traced Update (5.2.6.3.a to 5.2.6.3.g), if it is in compliance with the relevant OIML Recommendation. Traced Update is the procedure of changing software in a verified instrument or device after which the subsequent verification by a responsible person at place is not necessary. The software to be updated can be loaded locally i.e. directly on the measuring device or remotely via a network. The software update is recorded in an audit trail (see 3.1.2). The procedure of a Traced Update comprises several steps: loading, integrity checking, checking of the origin (authentication), installation, logging and activation.

5.2.6.3.a Traced Update of software shall be automatic. On completion of the update procedure the software protection environment shall be at the same level as required by the type approval.

5.2.6.3.b The target measuring instrument (electronic device, sub-assembly) shall have a fixed legally relevant software that cannot be updated and that contains all of the checking functions necessary for fulfilling Traced Update requirements.

5.2.6.3.c Technical means shall be employed to guarantee the authenticity of the loaded software i.e. that it originates from the owner of the type approval certificate. If the loaded software fails the authenticity check, the instrument shall discard it and use the previous version of the software or switch to an inoperable mode.

Example:

(II) The authenticity check ~~This can be~~ is accomplished ~~e.g.~~ by cryptographic means like a public key system~~signing~~. The owner of the type approval certificate (in general the manufacturer of the measuring instrument) is generating an electronic signature of the software to be updated using the *secret key* in the manufactory. The *public key* is stored in the fixed software part of the measuring instrument. The signature is checked using the *public key* during loading the software into the measuring instrument. If the signature of the loaded software is OK, it is installed and activated; if it fails ~~this test~~the check, the ~~instrument shall~~ fixed software discards it and uses the previous version of the software or switches to an inoperable mode.

5.2.6.3.d Technical means shall be employed to ensure the integrity of the loaded software i.e. that it has not been inadmissibly changed before loading. This can be accomplished by adding a checksum or hash code of the loaded software and verifying it during the loading procedure. If the loaded software fails this test, the instrument shall discard it and use the previous version of the software or switch to an inoperable mode. In this mode, the measuring functions shall be inhibited. It shall only be possible to resume the download procedure, without omitting any step in the flow diagram for Traced Update.

5.2.6.3.e Appropriate technical means, e.g. an audit trail, shall be employed to ensure that ~~.~~Traced Updates of legally relevant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection. ~~This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace Traced Updates of legally relevant software over an adequate period of time (that depends on national legislation).~~

The audit trail shall contain at minimum the following information: success / failure of the update procedure, software identification of the installed version, software identification of the previous installed version, time stamp of the event, identification of the downloading party. An entry is generated for each update attempt regardless of the success.

The storage device that supports the Traced Update shall have a sufficient capacity to ensure the traceability of Traced Updates of legally relevant software between at least two successive verifications in the field/inspection. After having reached the limit of the storage for the audit trail, it shall be ensured by technical means that further downloads are impossible without breaking a seal.

Note:

This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace Traced Updates of legally relevant software over an adequate period of time (that depends on national legislation).

5.2.6.3.f Depending on the needs and national legal legislation it may be necessary that the user or owner of the measuring instrument gives his consent to a download. The measuring instrument shall have sub-assembly / electronic device for the user or owner for expressing this consent, e.g. a push button, before the download starts. It shall be possible to enable and disable this sub-assembly / electronic device e.g. by a switch that can be sealed or by a parameter. If the sub-assembly / electronic device is enabled, each download has to be initiated by the user or owner. If it is disabled no activity by the user or owner is necessary to perform a download.

5.2.6.3.g If the requirements 5.2.6.3.a to 5.2.6.3.f cannot be fulfilled, it is still possible to update the legally non-relevant software part. In this case the following requirements shall be met:

- There is a distinct separation between the legally relevant and non-relevant software according to 5.2.1;
- The whole legally relevant software part cannot be updated without breaking a seal;

- It is stated in the type approval certificate that updating of the legally non-relevant part is acceptable.

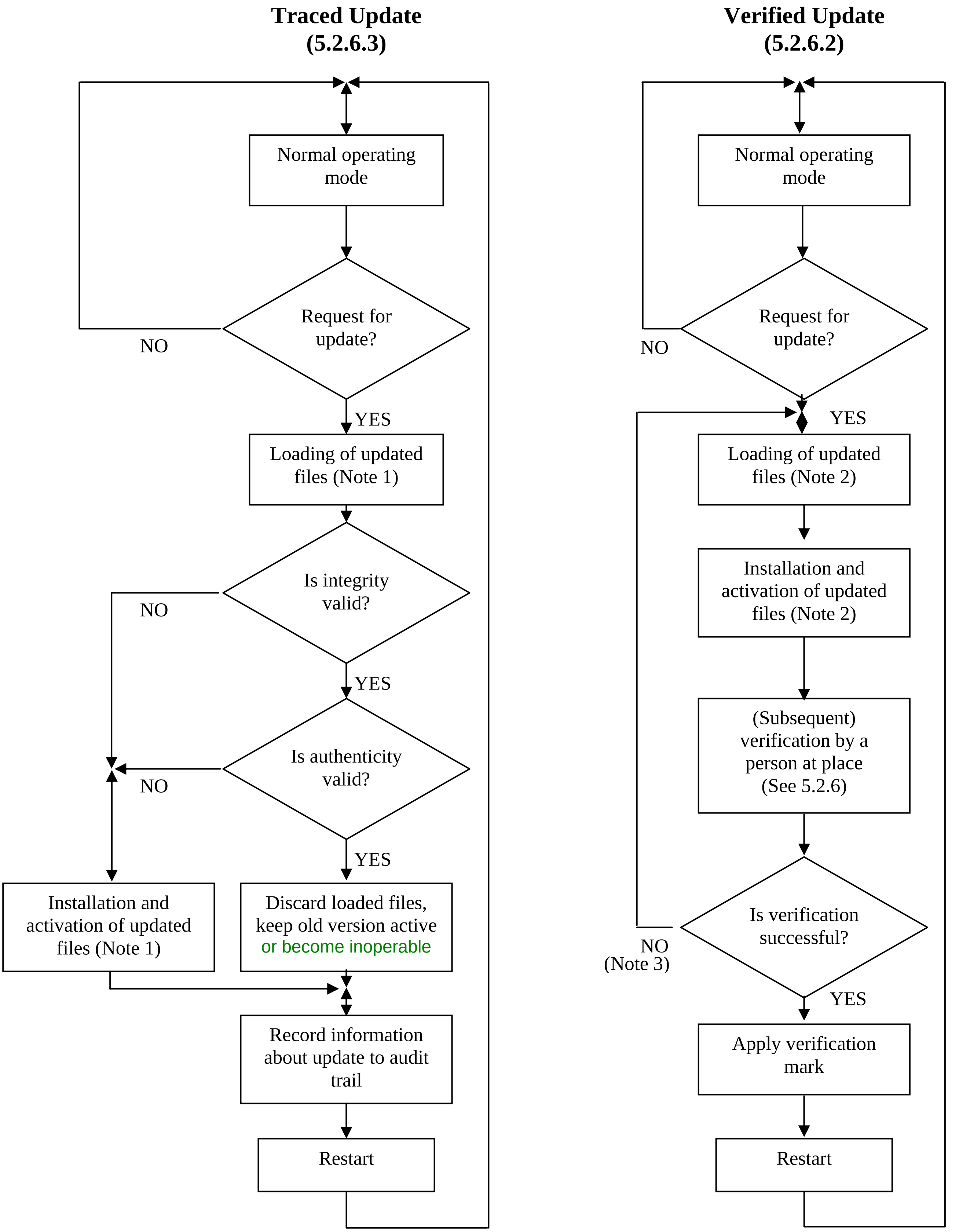
Traced Update
(5.2.6.3)
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 1)
Is integrity valid?
NO
YES
Is authenticity valid?
NO
YES
Installation and activation of updated files (Note 1)
Discard loaded files, keep old version active or become inoperable
Record information about update to audit trail
Restart
Verified Update
(5.2.6.2)
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 2)
Installation and activation of updated files (Note 2)
(Subsequent) verification by a person at place (See 5.2.6)
Is verification successful?
NO
(Note 3)
YES
Apply verification mark
Restart

Figure 1: Software update procedure

Notes:

(1) In case of Traced Update updating is separated into the steps: “loading” and “installing/activating”. This implies that the software is temporarily stored after loading without being activated because it must be possible to discard the loaded software and fall back to the old version, if the checks fail.

(2) In case of Verified Update the software may also be loaded and temporarily stored before installation but depending on the technical solution loading and installation may also be accomplished in one step.

(3) Here only failing of the verification because of the software update is considered. Failing because of other reasons doesn’t require re-loading and re-installing of the software, symbolised by the NO-branch.

5.2.6.4 The relevant OIML Recommendation may require the setting of certain device specific parameter to be available to the user. In such a case, the measuring instrument shall be fitted with a facility to automatically and non-erasably record any adjustment of the device specific parameter, e.g. an audit trail. The instrument shall be capable of presenting the recorded data.

Note:

An event counter is not an acceptable solution.

5.2.6.5 The traceability means and records are part of the legally relevant software and should be protected as such. The software employed for displaying the audit trail (5.2.6.2; 5.2.6.3) belongs to the fixed legally relevant software.

6 TYPE APPROVAL

6.1 Documentation to be supplied for type approval

For type approval the manufacturer of the measuring instrument shall declare and document all program functions, relevant data structures and software interfaces of the legally relevant software part that are implemented in the instrument. There shall not exist any hidden undocumented functions.

The commands and their effects shall be described completely in the software documentation to be submitted for type approval. The manufacturer shall state the completeness of the documentation of commands. If commands can be entered via a user interface, they shall be described completely in the software documentation to be submitted for the type approval.

Furthermore, the application for type approval shall be accompanied by a document or other evi-dence that supports the assumption that the design and characteristics of the software of the measuring instrument comply with the requirements of the relevant OIML Recommendation, in which the general requirements of this Document have been incorporated.

6.1.1 Typical documentation (for each measuring instrument, electronic device, or sub-assembly) basically includes:

- A description of the legally relevant software and how the requirements are met;
  - List of software modules that belong to legally relevant part (Annex B) including a declaration that all legally relevant functions are included in the description;
  - Description of the software interfaces of the legally relevant software part and of the commands and data flows via this interface including a statement of completeness (Annex B);
  - Description of the generation of the software identification;
  - Depending on the validation method chosen in the relevant OIML Recommendation (see 6.4) the source code shall be made available to the testing authority if high conformity or strong protection is required by the relevant OIML Recommendation;
  - List of parameters to be protected and description of protection means;
- A description of suitable system configuration and minimal required resources (see 5.2.4);
- A description of security means of the operating system (password, … if applicable);
- A description of the (software) sealing method(s);
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network etc. Where a hardware component is deemed legally relevant or performs legally relevant functions, this should also be identified;
- A description of the accuracy of the algorithms (like filtering of A/D conversion results, price calculation, rounding algorithms, …);
- A description of the user interface, menus and dialogues;
- The software identification and instructions for obtaining it from an instrument in use;
- List of commands of each hardware interface of the measuring instrument / electronic device / sub-assembly including a statement of completeness;
- List of durability errors that are detected by the software and if necessary for understanding, a description of the detecting algorithms;
- A description of data sets stored or transmitted;
- If fault detection is realised in software, a list of faults that are detected and a description of the detecting algorithm;
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network etc;
- The operating manual.

6.2 Requirements on the approval procedure

Test procedures in the framework of the type approval e.g. described in Document D11 are based on well defined test setups and test conditions and can rely on precise comparative measurements. “Testing” and “validating” software means something different. The accuracy or correctness of software in general cannot be measured in a metrological sense though there are standards how to “measure” software quality [e.g. ISO/IEC 14598]. The procedures described here take into consideration both the needs in legal metrology and well-known validation and test methods in software engineering not having the same goal like e.g. the software developer who is searching errors and optimising performance. As shown in 6.4 each software requirement needs individual adaptation of suitable validation procedures. The effort for the procedure should reflect the importance of the requirement in terms of accuracy, reliability and protection against corruption.

The aim is to validate that the instrument to be approved complies with the requirements of the relevant OIML Recommendation. For software controlled instruments the validation procedure comprises examinations, analysis, and tests and the relevant OIML Recommendation shall include an appropriate selection of methods described in the following.

Methods described in the following focus on the type examination. Verifications of every single instrument in use in the field are not covered by those validation methods. Refer to Chapter 7 Verification for more information.

The methods specified for software validation are described in 6.3. Combinations of these methods forming a complete validation procedure adapted to all requirements defined in Section 5 are specified in Chapter 6.4.

6.3 Validation methods (software examination)

6.3.1 Overview of methods and their application

The selection and sequence of the following methods are not prescribed and may vary in a validation procedure from case to case.

Abbrevia-tion	Description	Application	Preconditions, tools for application	Special skills for performing
AD	Analysis of the documentation and validation of the design (6.3.2.1)	Always	Documentation	-
VFTM	Validation by functional testing of metrological features (6.3.2.2)	Correctness of the algorithms, uncertainty, compensating and correcting algorithms, rules for price calculation	Documentation	-
VFTSw	Validation by functional testing of software features (6.3.2.3)	Correct functioning of communication, indication, fraud protection, protection against operating errors, protection of parameters, fault detection	Documentation, common software tool	-
DFA	Metrological data flow analysis (6.3.2.4)	Software separation, evaluation of the impact of commands on the instrument's functions	Source code, common software tool (simple procedure), tools (sophisticated procedure)	Knowledge of programming languages. Instruction for the method necessary.
CIWT	Code inspection, Walkthrough (6.3.2.5)	All purposes	Source code, common software tool	Knowledge of programming languages, protocols, and other IT issues
SMT	Software module testing (6.3.2.6)	All purposes when input and output can clearly be defined	Source code, testing environment, special software tools	Knowledge of programming languages, protocols, and other IT issues. Instruction for using the tools necessary.

Table 1: Overview of the proposed selected validation methods

Note:

Text editor, hexadecimal editor etc. are considered as “common software tool”.

6.3.2 Description of selected validation methods

6.3.2.1 Analysis of Documentation and Specification and Validation of the Design (AD)

Application:

This is the basic procedure that has to be applied in any case.

Preconditions:

The procedure is based on the manufacturer’s documentation of the measuring instrument. Depending on the demands this documentation shall have adequate scope:

(1) Specification of the externally accessible functions of the instrument in a general form (Suitable for simple instruments with no interfaces except a display, all features verifiable by functional testing, low risk of fraud);

(2) Specification of software functions and interfaces (Necessary for instruments with interfaces and for instrument functions that cannot be functionally tested and in case of increased risk of fraud). The description shall make evident and explain all software functions that may have an impact on metrological features;

(3) Concerning interfaces the documentation shall include a complete list of commands or signals that the software is able to interpret. The effect of each command shall be documented in detail. It shall be described how the instrument reacts on undocumented commands;

(4) Additional documentation of the software for complex measuring algorithms, cryptographic functions, or crucial timing constraints shall be provided, if necessary for understanding and evaluating the software functions;

(5) When it is not clear how to validate a function of a software program the onus to develop a test method should be placed on the manufacturer. In addition, the services of the programmer should be made available to the examiner for the purposes of answering questions.

A general precondition for examination is the completeness of the documentation and the clear identification of the EUT i.e. of the software packages that contribute to the metrological functions (see 6.1.1).

Description:

The examiner evaluates the functions and features of the measuring instrument using the verbal description and graphical representations and decides whether they comply with the requirements of the relevant OIML Recommendation. Metrological requirements as well as software-functional requirements defined in chapter 5 (like e.g. fraud protection, protection of adjustment parameters, disallowed functions, communication with other devices, update of software, fault detection) have to be considered and evaluated. This task may be supported by Software Evaluation Report Format (see Annex B).

Result:

The procedure gives a result for all characteristics of the measuring instrument provided an appropriate documentation has been submitted by the manufacturer. The result should be documented in a chapter related to software in a Software Evaluation Report (see Annex B) included in the Evaluation Report Format of the relevant OIML Recommendation.

Complementing procedures:

Additional procedures should be applied, if examining the documentation cannot give substantiated validation results. In most cases “Validating the metrological functions by functional testing” (see 6.3.2.2) is a complementing procedure.

References:

FDA, Guidance for FDA Reviewers and Industry, 29 May 1998

IEC 61508-7, 2000-3

6.3.2.2 Validation by Functional Testing of the Metrological Functions (VFTM)

Application:

Correctness of algorithms for calculating the measurement value from raw data, for linearisation of a characteristic, compensation of environmental influences, rounding in price calculation etc.

Preconditions:

Operating manual, functioning pattern, metrological references and test equipment.

Description:

Most of the approval and test methods described in OIML Recommendations are based on reference measurements under various conditions. Its application is not restricted to a certain technology of the instrument. Though it doesn't aim primarily on validation of software the test result can be interpreted as a validation of some software parts, in general even the metrologically most important. If the tests described in the relevant OIML Recommendation cover all metrologically relevant features of the instrument, the corresponding software parts can be regarded as being validated. In general no additional software analysis or test has to be applied to validate the metrological features of the measuring instrument.

Result:

Correctness of algorithms valid or invalid. Measurement values under all conditions within MPE or not.

Complementing procedures:

The method is normally an enhancement to 6.3.2.1. In certain cases it may be easier or more effective to combine the method with examinations based on the source code (6.3.2.5) or by simulating input signals (6.3.2.6) e.g. for dynamic measurements.

References:

Various specific OIML Recommendations.

6.3.2.3 Validation by Functional Testing of the Software Functions (VFTSw)

Application:

Validation of e.g. protection of parameters, indication of a software identification, software supported fault detection, configuration of the system especially of the software environment etc.

Preconditions:

Operating manual, software documentation, functioning pattern, test equipment.

Description:

Required features described in the operating manual, instrument documentation or software documentation are checked practically. If they are software controlled, they are to be regarded as validated if they function correctly without any further software analysis. Features addressed here are e.g.

- Normal operating of the instrument if operating is software controlled. All switches or keys and described combinations should be employed and the reaction of the instrument be evaluated. In graphical user interfaces all menus and other graphical elements should be activated and checked;
- Effectiveness of parameter protection may be checked by activating the protection means and trying to change a parameter;
- Effectiveness of the protection of stored data may be checked by changing some data in the file and check whether this is detected by the program;
- Generation and indication of software identification may be validated by practical checking;
- If fault detection is software supported, the relevant software parts may be validated by provoking, implementing or simulating a fault and check the correct reaction of the instrument;
- If configuration or environment of the legally relevant software is claimed to be fixed, protection means can be checked by making unauthorized changes. The software should inhibit these changes or should stop.

Result:

Software controlled feature under consideration OK or not.

Complementing Procedures:

Some features or functions of a software controlled instrument cannot be practically validated as described. If the instrument has interfaces, it is in general not possible to detect unauthorized commands only by trying commands at random. Apart from that a sender is needed to generate these commands. For normal validation level method 6.3.2.1 including a declaration of the manufacturer may cover this requirement. For extended examination level a software analysis like 6.3.2.4 or 6.3.2.5 is necessary.

References:

WELMEC 2.3, 7.2, FDA Guidance for Industry Part 11, August 2003

6.3.2.4 Metrological Dataflow Analysis (DFA)

Application:

Construction of the flow of measurement values through the data domains subject to legal control. Examination of the software separation.

Preconditions:

Software documentation, source code, editor, text search program or special tools. Knowledge of programming languages.

Description:

It is the aim of this method to find all parts of the software that are involved in the calculation of the measurement value or that may have an impact on it. Starting from the hardware port where measurement raw data from the sensor are available, the subroutine is searched that reads them. This subroutine will store them in a variable after possibly having done some calculation. From this variable the intermediate value is read by another subroutine and so forth until the completed measurement value is output to the display. All variables that are used as storage for intermediate measurement values and all subroutines transporting these values can be found in the source code simply by using a text editor and a text search program for finding variable or subroutine names in another source code file than the currently opened in the text editor.

Other data flows can be found by this method e.g. from interfaces to the interpreter of received commands. Furthermore circumvention of a software interface (see 5.2.1.2) can be detected.

Result:

It can be validated whether software separation according 5.2.1.2 is OK or not.

Complementing Procedures:

This method is recommended if software separation is realised and if high conformity or strong protection against manipulation is required. It is an enhancement to 6.3.2.1 to 6.3.2.3 and 6.3.2.5.

References:

IEC 61131-3

6.3.2.5 Code Inspection and Walk Through (CIWT)

Application:

Any feature of the software may be validated with this method if enhanced examination intensity is necessary.

Preconditions:

Source code, text editor, tools. Knowledge of programming languages.

Description:

The examiner walks through the source code assignment by assignment, evaluating the respective part of the code to determine whether the requirements are fulfilled and whether program functions and features are in compliance with the documentation.

The examiner may also concentrate on algorithms or functions that he has identified as complex, error-prone, insufficiently documented etc. and inspect the respective part of the source code by analysing and checking.

Prior to these examination steps the examiner will have identified the legally relevant software part e.g. by applying the metrological data flow analysis (see 6.3.2.4). In general code inspection or walk through is limited to this part. By combining both methods the examination effort is minimal compared to the application of these methods in the normal software production with the objective of producing failure-free programs or optimising performance.

Result:

Implementation compatible with the software documentation and in compliance with the requirements or not.

Complementing Procedures:

This is an enhanced method, additional to 6.3.2.1and 6.3.2.4. Normally it is only applied in spot checks.

References:

IEC 61508-7

6.3.2.6 Software Module Testing (SMT)

Application:

Only if high conformity and protection against fraud is required. This method is applied when functions of a program cannot be examined exclusively on the basis of written information. It is appropriate and economically advantageous in validation of dynamic measurement algorithms.

Preconditions:

Source code, development tools (at least a compiler), functioning environment of the software module under test, input data set and corresponding correct reference output data set or tools for automation. Skill in IT, knowledge of programming languages. Co-operation with the programmer of the module under test is advisable.

Description:

The software module under test is integrated in a test environment i.e. a specific test program module that is calling the module under test and providing it with all necessary input data. The test program receives output data from the module under test and compares them with the expected reference values.

Result:

Measuring algorithm or other tested functions correct or not.

Complementing Procedures:

This is an enhanced method, additional to 6.3.2.2 or 6.3.2.5. It is only profitable in exceptional cases.

References:

IEC 61508-7

6.4 Validation procedure

Validation procedure consists of a combination of analysis methods and tests. The relevant OIML Recommendation may specify details concerning the validation procedure, including:

(a) which of the validation methods described in 6.3 shall be carried out for the requirement under consideration;

(b) how the evaluation of test results shall be performed;

(c) which result should be included in the test report and which should be integrated in the test certificate (see Annex B).

In Table 2 two alternative levels A and B for the validation procedures are defined. Level B implies an extended examination compared to A. A selection between A and B type of validation procedures may be made in the relevant OIML Recommendation - different or equal for each requirement - in accordance to expected:

- Risk of fraud;
- Area of application;
- Required conformity to approved type;
- Risk of wrong measurement result due to operating errors.

Requirement		**Validation procedure A** (normal examination level)	**Validation procedure B** (extended examination level)	**Comment**
5.1.1	Software identification	AD + VFTSw	AD + VFTSw + CIWT	Select »B« if high conformity is required
5.1.2	Correctness of algorithms and functions	AD + VFTM	AD + VFTM + CIWT/SMT	
	Software protection			
5.1.3.1	Prevention of accidental misuse	AD + VFTSw	AD + VFTSw	
5.1.3.2	Fraud protection	AD + VFTSw	AD + VFTSw + DFA/CIWT/SMT	Select »B« in case of high risk of fraud
Support of hardware features				
5.1.4.1	Support of fault detection	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select »B« if high reliability is required
5.1.4.2	Support of durability protection	AD + VFTSw	AD + VFTSw + CIWT + SMT	Select »B« if high reliability is required
Specifying and separating of relevant parts and specifying of interfaces of parts				
5.2.1.1	Separation of electronic devices and sub-assemblies	AD	AD	
5.2.1.2	Separation of software parts	AD	AD + DFA/CIWT	
5.2.2	Shared indications	AD + VFTM/ VFTSw	AD + VFTM/ VFTSw + DFA/CIWT	
5.2.3	Storage of data, transmission via communication systems	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« if transmission of measurement data in open system is foreseen
5.2.3.1	The measurement data must be stored automatically when the measurement is concluded	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« in case of high risk of fraud
5.2.3.2	Transmission delays	AD + VFTSw	AD + VFTSw + SMT	Select »B« in case of high risk of fraud, e.g. transmission in open systems
5.2.4	Compatibility of operating systems and hardware, portability	AD + VFTSw	AD + VFTSw + SMT	
Maintenance and re-configuration				
5.2.6.2	Verified Update	AD	AD	
5.2.6.3	Traced Update	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« in case of high risk of fraud

Table 2: Recommendations for combinations of analysis and test methods for the various software requirements (acronyms defined in Table 1)

6.5 Equipment under test (EUT)

Normally tests will be carried out on the complete measuring instrument (functional testing). If the size or configuration of the measuring instrument does not lend itself to testing as a whole unit or if only a separate device (module) of the measuring instrument is concerned, the relevant OIML Recommendation may indicate that the tests, or certain tests, shall be carried out on the electronic devices or software modules separately, provided that, in case of tests with the devices in operation, these devices are included in a simulated set-up, sufficiently representative of its normal operation. The approval applicant is responsible for the provision of all required equipment and components.

7 VERIFICATION

If a metrological control of measuring instruments is prescribed in a country, there shall be means to check in the field during operation the identity of the software, the validity of adjustment and the conformity to the approved type.

The relevant OIML Recommendation may require carrying out the verification of the software in one or more stage according to the nature of the considered measuring instrument.

The verification of software shall include:

- An examination of the conformity of the software with the approved version (e.g. verification of the version number and checksum);
- An examination that the configuration is compatible with the declared minimal configuration, if given in the approval certificate;
- An examination that inputs/outputs of the measuring instrument are well configured in the software when their assignment is a device specific parameter;
- An examination that the device specific parameters (especially adjustment parameters) are correct.

The procedure of software update is described in 5.2.6.2 and 5.2.6.3.

8 ASSESSMENT OF SEVERITY (RISK) LEVELS

8.1 This chapter is intended as a guide to determine a set of severity levels to be generally applied for tests carried out on electronic measuring instruments. It is not intended as a classification with strict limits leading to special requirements as in the case of an accuracy classification.

Moreover, this guide does not restrict the technical committees and subcommittees from providing severity levels that differ from those resulting from the guidelines set forth in this Document. Different severity levels may be used in accordance with special limits prescribed in the relevant OIML Recommendations.

8.2 Selecting severity levels for a particular category of instruments and area of application (trade, direct selling to the public, health, law enforcement ...), the following aspects can be taken into account:

(a) risk of fraud;

- the consequence and the social and societal impact of malfunction;
- the value of goods to be measured;
- platform used (built for purpose or universal computer);
- exposure to sources of potential fraud (unattended self service device);

(b) required conformity;

- the practical possibilities for the industry to comply with the prescribed level

(c) required reliability;

- environmental conditions;
- the consequence and the social and societal impact of errors;

(d) interest of the defrauder;

- being able to commit fraud can be the sufficient motivational factor;

(e) the possibility to repeat a measurement or to interrupt it.

Throughout the requirements' section (see 5) various examples for acceptable technical solutions are given illustrating the basic level of protection against fraud, conformity, reliability, and type of measurement (marked with (I)). Where suitable also examples with enhanced counter measures are presented that consider a raised severity level of the aspects described above (marked with (II)).

Validation procedure and severity (risk) level are inextricably linked. A deep analysis of the software shall be performed when a raised severity level is required in order to detect software deficiency or security weakness. ~~In~~ On the other hand, mechanical sealing (e.g. sealing of the communication port or the housing) should be considered when choosing the validation procedure.

ANNEX A
Notes and Bibliography
(to be completed)

At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and the users of this Document 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.

The actual status of the standards referred to can also be found on the Internet:

IEC Publications:	http://www.iec.ch/searchpub/cur_fut.htm
ISO Publications:	http://www.iso.org/iso/en/CatalogueListPage.CatalogueList
OIML Publications:	http://www.oiml.org/publications/ (with free download of PDF files).

In order to avoid any misunderstanding, it is highly recommended that all references to standards in OIML Recommendations and International Documents shall be followed by the version referred to (generally the year or date).

Ref.	Standards and reference documents	Description
[1]	International Vocabulary of Basic and General Terms in Metrology (VIM) (1993)	Vocabulary, prepared by a joint working group consisting of experts appointed by BIPM, IEC, IFCC, ISO, IUPAC, IUPAP, and OIML
[2]	The OIML Certificate system for Measuring Instruments	
[3]	OIML D11	
	X.509	
	ISO/IEC 9594-8: 2001	
	ISO 2382-9	
	IEC 61508-4	
	ISO/IEC 14598 and IEC 61508-4	
	IEC 61508-7, 2000-3	
	FDA, Guidance for FDA Reviewers and Industry, 29 May 1998	
	FDA Guidance for Industry Part 11, August 2003	
	WELMEC 2.3, 2.5, 7.2	
	Reference between Draft Measurement Canada Specification (Metrological Software), 8 August 2002	

ANNEX B

Exampleof Software Evaluation Report (informative)

Note:

> The Technical Committees and Sub-Committees developing OIML Recommendations should decide which information shall be included in Test Report and OIML Certificate of Conformity. E.g. the name, version and checksum of the executable file from the following example should be included in the Test Certificate.

Test report no XYZ122344
Validation of Software of the flow meter Tournesol Metering **model** TT100

The software of the measuring instrument was validated to show conformance with the requirements of the OIML Recommendation R-xyz.

The validation was based on the report OIML international document D-SW, where the essential requirements for software are interpreted and explained. This report describes the examination of software needed to state conformance with the R-xyz.

Manufacturer
Tournesol Metering
P.O. Box 1120333
100 Klow
Syldavie
Reference: Mr Tryphon Tournesol

Applicant
New Company
Nova Street 123
1000 Las Dopicos
San Theodorod
Reference: Archibald Haddock

Test Object

The Tournesol Metering meterTT100 is a measuring instrument intended to measure flow in liquids. The intended range is from 1 l/s up to 2000 l/s. The basic functions of the instrument are:

- measuring of flow in liquids,
- indication of measured volume,
- interface to transducer.

The flow meter is described as a built-for-purpose measuring instrument (an embedded system) with storage device of legally relevant data.

The flow meterTT100 is an independent instrument with a transducer connected. The transducer incorporates a temperature compensation. Adjustment of flow rates is possible by calibration parameters stored in a non-volatile memory of the transducer. It is fixed to the instrument and cannot be disconnected. The measured volume is indicated on a display. No communication with other devices is possible.

The embedded software of the measuring instrument was developed by

Tournesol Metering, P.O. Box 1120333, 100 Klow, Syldavie.

The executable file name is “**tt100_12.exe**”.

The validated version of this software is **V1.2c**.

The software version is presented on the display upon device start-up and by pressing the “level” button for 4 seconds.

The source code comprises following legally relevant files:

-	main.c	12301 byte	23 Nov 2003
-	int.c	6509 byte	23 Nov 2003
-	filter.c	10897 byte	20 Oct 2003
-	input.c	2004 byte	20 Oct 2003
-	display.c	32000 byte	23 Nov 2003
-	Ethernet.c	23455 byte	15 June 2002
-	driver.c	11670 byte	15 June 2002
-	calculate.c	6788 byte	23 Nov 2003

The executable file “**tt100_12.exe**” is protected against modification by a checksum. The value of checksum by algorithm **XYZ** is **1A2B3C**.

~~The software version is presented on the display upon device start-up and by pressing the “level” button for 4 seconds.~~

The validation has been supported by following documents from the manufacturer:

- TT 100 User Manual Release 1.6
- TT 100 Maintenance Manual Release 1.1
- Software description TT100 (internal design document, dated 22 Nov 2003)
- Electronic circuit diagram TT100 (drawing no 222-31, date 15 Oct 2003)

The final version of the test object was delivered to National Testing & Measurement Laboratory on 25 November 2003.

Performance of validation

The validation has been performed according to the OIML D-SW (version 1.0). The validation was performed between 1 November and 23 December 2003. A design review was held on 3 December by Dr K. Fehler at Tournesol Meteringhead office in Klow. Other validation work has been carried out at the National Testing & Measurement Lab by Dr K. Fehler and M. S. Problème.

Following requirements have been validated:

- Software identification,
- Correctness of algorithms and functions,
- Software protection,
- Prevention of accidental misuse,
- Fraud protection,
- Support of hardware features,
- Storage of data, transmission via communication systems.

Following validation methods have been applied:

- Analysis of the documentation and validation of the design,
- Validation by functional testing of metrological features,
- Walkthrough, code inspection,
- Software Module testing of module calculate.c with SDK XXX.

Result

Following requirements of the OIML D-SW have been validated without finding faults:

5.1.1, 5.1.2, 5.1.3.2, 5.2.1, 5.2.2.1, 5.2.2.2, 5.2.2.3.

Two commands which were not initially described in the operator's manual were found. The two commands have been included in the operator's manual dated 10 December 2003.

A software fault which limited the month of February to 28 days also in leap year was found in software package V1.2b. This has been corrected in V1.2c.

The result applies to the tested item with Serial No. 1188093-B-2004 only.

Conclusion

The software of the **Tournesol Metering TT100 V1.2c** fulfils the requirements of the OIML R-xyz.

National Testing & Measurement Lab
Software Department

Dr. K.E.I.N. Fehler	M. S.A.N.S. Problème
Technical manager	Technical Officer

Checklist

§(D-SW)	Requirement	+passed	-failed	Remarks
5.1	**General Requirements**			
5.1.1	**Software identification** *Legally relevant software shall be clearly identified.*			
5.1.2	**Correctness of algorithms and functions** The measuring algorithms and functions of a measuring device shall be correct			
5.1.3	**Software protection**			
5.1.3.1	**Prevention of accidental misuse** A measuring instrument – especially the software – shall be constructed in a way that possibilities for unintentional accidental misuse are minimal			
	Fraud protection			
5.1.3.2	Metrologically critically software shall be secured against unauthorized modification, loading, or changes by swapping of hardware memory			
	Support of hardware features			
5.1.4	**Support of fault detection**			
5.1.4.1	It is the manufacturer's choice to realise checking facilities addressed in D11 (5.1.2 (b) and 5.3) in software or hardware or let hardware facilities be supported by software.			
	Support of durability protection			
5.1.4.2	*It is the manufacturer's choice to realize durability protection facilities addressed in D11 (5.1.3 (b) and 5.4) in software or hardware or let hardware facilities be supported by software*			
5.2	**Specific requirements**			
5.2.1	**Specifying and separating relevant parts and specifying interface of parts** *Metrologically critical parts of a measuring system – whether software or hardware parts – shall not be inadmissibly influenced by other parts of the measuring system*			
	Separation of devices and sub-assemblies			
5.2.1.1	Interfaces of these "legally relevant" sub-assemblies and devices shall be clearly defined and documented to show that their relevant functions and data cannot be inadmissibly influenced by commands received via the interface			

5.2.1.2	**Separation of software parts** *If the legally relevant software part communicates with other software parts, a software interface shall be defined. All communication shall be performed exclusively via this interface*			
5.2.2	**Shared indications** *The distinction between information from the legally relevant part of software and other information shall be clear and unambiguous*			
5.2.3	**Storage of data, transmission via communication system** *The data shall be protected by software means to guarantee their identity, correctness of the information of the time of measurement, authenticity, and integrity*			
5.2.3.1	*The measurement data must be stored automatically when the measurement is concluded. The storage device must have a capacity which is sufficient for the intended purpose*			
5.2.3.2	*The measurement must not be inadmissibly influenced by a transmission delay. If network services become unavailable, no measurement data must get lost*			
5.2.4	**Compatibility of operating system and hardware, portability** *The manufacturer of the metrologically relevant software shall identify the hardware and software environment that is suitable*			
5.2.6	**Maintenance and reconfiguration** *Only approved versions of legally relevant software are allowed for use*			
5.2.6.2	**Verified Update** *After update of the legally relevant software of a measuring instrument it is necessary to perform a verification of the instrument and renew the securing means*			
5.2.6.3	**Traced Update** *The software is implemented into the instrument according to the requirements for Traced Update (***5.2.6.2.1** *to* **5.2.6.2.7***) if it is in compliance with national legislation. Traced Update is the procedure of changing software in a verified instrument or device after which the verification is not necessary*			

ANNEX C

Index *(will be completed at the end)*