UDC 621.317.1

O. M. Velychko[1, 2], DSc, **V. V. Gaman**[2], **O. V. Hrabovskii**[1], PhD, **T. B. Gordiyenko**[1], DSc

[1]*Odesa State Academy of Technical Regulation and Quality, Odesa*
[2]*State Enterprise "Ukrmetrteststandard", Kyiv*

FEATURES OF TESTING OF THE BUILT-IN SOFTWARE OF MEASURING INSTRUMENTS

Protection against unauthorized modification or its components, measurement data and protection against external unintentional and accidental interference are particularly important for measuring instruments (MIs) software. The possibilities of ensuring the security of built-in MI software, as well as MI as a whole, are considered. The analysis of the main features of built-in MI software on the basis of documents, guidelines and recommendations of international and regional organizations in the field of legal metrology are shown. The characteristics of MI software in accordance with international and regional documents, guidelines and recommendations are the establishment and regulation of such tests, which must be provided in the testing methods for specific MI with embedded software.

Keywords: *software, measuring instruments, testing, legal metrology, normative base.*

О. М. Величко, д.т.н., **В. В. Гаман, О. В. Грабовський,** к.т.н., **Т. Б. Гордієнко,** д.т.н.

ОСОБЛИВОСТІ ТЕСТУВАННЯ ВБУДОВАНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАСОБІВ ВИМІРЮВАЛЬНОЇ ТЕХНІКИ

Сучасний рівень розвитку інформаційних технологій робить все більший вплив на спеціалізоване програмне забезпечення (ПЗ) для засобів вимірювальної техніки (ЗВТ). Вимоги та порядок тестування ПЗ для ЗВТ викладені в документах міжнародних і регіональних організацій у галузі законодавчої метрології. У більшості випадків спеціалізоване ПЗ є одним із ключових компонентів такого ЗВТ. З огляду на це, необхідні ефективні методи тестування ПЗ ЗВТ з урахуванням ризиків і загроз, пов'язаних з його використанням.

ПЗ для ЗВТ має бути розроблене таким чином, щоб забезпечити максимальну придатність для правильного застосування самого ЗВТ. Залежно від типу конструкції ЗВТ, чи вбудованого ПЗ у ЗВТ, чи ЗВТ на основі універсального комп'ютера, використовуються різні підходи та заходи щодо захисту ПЗ, його компонентів і вимірювальних даних. Проведений аналіз основних особливостей ЗВТ із вбудованим ПЗ на основі документів, керівництв і рекомендацій міжнародних і регіональних організацій у галузі законодавчої метрології.

ПЗ для ЗВТ повинно надавати всю необхідну інформацію щодо його ідентифікації, заходів, що вживаються для забезпечення безпеки та придатності ПЗ. Найбільші вимоги щодо цього встановлюються для ПЗ ЗВТ, що використовуються як побутові прилади обліку витрат різноманітних ресурсів. Розглянуто можливості забезпечення безпеки як вбудованого в ЗВТ ПЗ, так і ЗВТ загалом. Розглянуті різні варіанти розміщення ПЗ у ЗВТ і особливості його тестування, однак основну увагу сконцентровано на вбудованому ПЗ у ЗВТ.

Для ПЗ ЗВТ особливо важливі захист від несанкціонованої його модифікації чи його компонентів, даних вимірювань та захист від зовнішніх ненавмисних та випадкових втручань. Характеристиками ПЗ ЗВТ згідно з міжнародними та регіональними документами, керівництвами і рекомендаціями є встановлення та регламентація таких випробувань, які необхідно передбачити в методах тестування для конкретних ЗВТ із вбудованим ПЗ.

Ключові слова: *програмне забезпечення, засоби вимірювальної техніки, випробування, законодавча метрологія, нормативна база.*

О. Н. Величко, д.т.н., **В. В. Гаман, О. В. Грабовский,** к.т.н., **Т. Б. Гордиенко,** д.т.н.

ОСОБЕННОСТИ ТЕСТИРОВАНИЯ ВСТРОЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СРЕДСТВ ИЗМЕРИТЕЛЬНОЙ ТЕХНИКИ

Для программного обеспечения (ПО) средств измерительной техники (СИТ) особенно важны защита от его несанкционированного изменения или его компонентов, защита данных измерений и *защита от внешних непреднамеренных или случайных влияний. Рассмотрены возможности обеспечения безопасности встроенного ПО СИТ, а также СИТ в целом. Проведен анализ основных особенностей встроенного ПО МИ на основе документов, руководств и рекомендаций международных и региональных организаций в области законодательной метрологии. Характеристики ПО MI в соответствии с международными и региональными документами, руководствами и рекомендациями являются установлением и регулированием таких испытаний, которые должны быть указаны в методах испытаний для конкретного MI со встроенным ПО.*

Ключевые слова: *программное обеспечение, средства измерительной техники, испытания, законодательная метрология, нормативная база.*

DOI 10.32684/2412-5288-2020-1-16-36-41

Introduction

The modern level of the development of information technology (IT) is having an increasing influence on specialized software for measuring instruments (MIs). The document of the International Organization of Legal Metrology (OIML) [1], the recommendation of the Eurasian Cooperation of the State Metrology Institute (COOMET) [2], the document and guideline of the European Organization for Cooperation in Legal Metrology (WELMEC) [3, 4] establish the main requirements and procedures for testing software for MIs.

Specialized software for MI has become one of the key components of modern MI. The basic MI configuration is of two types: based on a universal computer and built-in MI itself. The requirements for the respective two types of software are different, since built-in MI software is more secure. It is much easier to implement an extraneous intentional change to universal computer software. In addition, the MI used for accounting must have built-in MI software with a higher protection class.

It is relevant to select effective methods for testing specialized software for MI in order to identify the main risks and threats associated with its use. For this, it is necessary to analyze the basic requirements of international and regional documents, guidelines and recommendations and highlight the most critical ones. This is, first of all, advisable to do for built-in MI software.

Analysis of publications and research

An analysis of the requirements of regulatory support regarding specialized software for MI is shown in [5]. An analysis of the features of software testing methods for MI according to the requirements of [1, 4] is described in [6]. The algorithms for testing software for MI and their main components for OIML requirements [1] and WELMEC requirements [3, 4] are considered in [7]. An analysis of the regulatory framework at the national level for testing software for MI described in [8] in order to test its suitability for conformity assessment of MI. However, these works are not aimed at research the security of built-in MI software.

The security issues, risk assessments and threats associated with the use of MI software are discussed in [9-12]. Those research focus on testing methods according the WELMEC requirements [4] and international standards. However, these works do not take into account the OIML requirements [1] and the possibility of using the built-in MI software.

An approach aimed at automatically checking the parameters for built-in MI software in accordance with the OIML requirements [1] is proposed in [13]. The general criteria for assessing the safety and protection of software components are considered. However, this work does not take into account the WELMEC requirements [4].

Thus, in the described researches, no analysis was made of the complete fulfillment of all OIML requirements [1], in particular, with regard to testing of the built-in MI software.

Setting objectives

The purpose of this research is to develop approaches to ensure security built-in MI software, taking into account the features of its testing. This will increase the protection of MI software against unauthorized changes to the software and its components, measurement data, unintentional and / or accidental interference. This will help to protect built-in MI software from unauthorized changes to the software and its components, measurement data, unintentional and/or accidental interference.

Main features of the bilt-in MI software

Depending on the type of construction of the MI, from the built-in MI software to MI on the basis of a universal computer, different approaches and measures for protecting the software, its components and measurement data are used.

The built-in MI software (Figure 1) has the following features:

they are built for specific measurable purposes, no other tasks can be fulfilled;

they are built on the basis of microcontrollers that contain the components required for work and have a fixed set of commands;

the user can enter only those commands that are installed by the manufacturer;

the display only shows the information that is installed by the manufacturer;

only the communication interfaces (Ethernet, USB, RS-232) that are installed by the manufacturer are used;

the measuring and hardware part may be placed in a protective case that may be sealed.

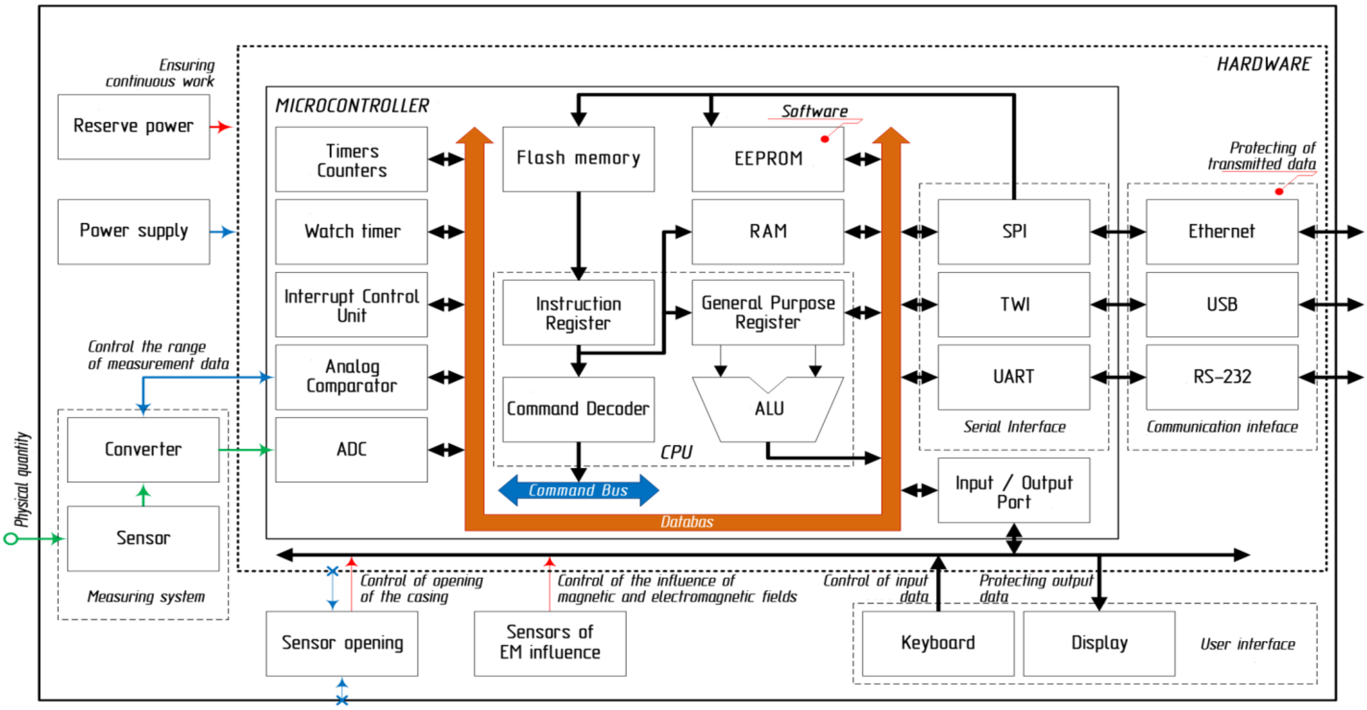


Figure 1 – General structure diagram of built-in MI software

These features provide the following software security capabilities (as well as the MI as a whole):

there is a possibility to block hardware reboot of the built-in software;

when using specialized microcontrollers that have built-in analog-to-digital converter (ADC) channels, measuring data from sensors or transducers comes directly to the microcontroller nodes for further processing, which excludes the possibility of interfering with these data;

it is possible to use microcontrollers with built-in specialized modules to provide protection: rapid calculation of CRC16/CRC32, random number generator, resource access separation, memory protection (MPU), etc.;

without interference in the hardware part of the device, it is impossible to spoil the data and commands transmitted between the microcontroller and the user interface, and it is impossible to enter commands not specified by the manufacturer;

through communication interfaces, only those commands and data that are associated with the measurement process are identified and transmitted by the manufacturer, in addition, some microcontrollers contain an integrated module input/output data port and bridge peripheral;

in the event of MI failures associated with software, it is possible to restart it on a watch command;

the hardware part can be placed in the protective case, in case of which a record will be created in the event log;

if necessary it is possible to place sensors of magnetic and electromagnetic (EM) fields near sensitive elements of the measuring system, in case of exceeding the established norm, a record will be created in the event log;

a reserve power may be used to ensure continuous operation.

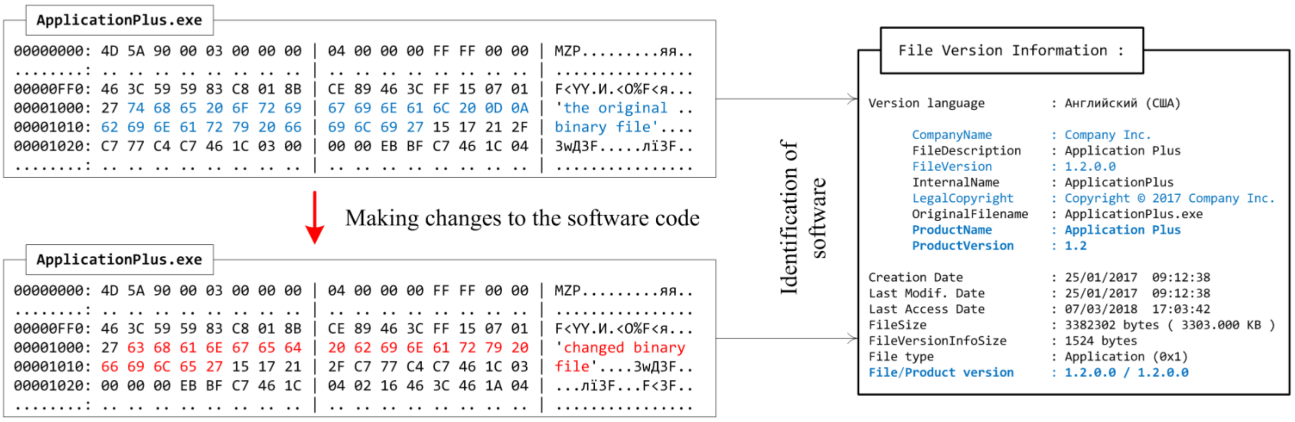
If MI has built-in software that is hosted on:

microchips, permanent memory devices without erasure (PROM);

microchips, permanent memory devices with ultra-violet light (EPROM);

PROM or EPROM microcontrollers;

permanent memory devices with the ability to erase (EPROM, Flash) in the absence of data interfaces;


Making changes to the software code
Identification of software

```
ApplicationPlus.exe
00000000: 4D 5A 90 00 03 00 00 00 | 04 00 00 00 FF FF 00 00 | MZP.........яя..
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. .. | ................
00000FF0: 46 3C 59 59 83 C8 01 8B | CE 89 46 3C FF 15 07 01 | F<YY.И.<O%F<я...
00001000: 27 74 68 65 20 6F 72 69 | 67 69 6E 61 6C 20 0D 0A | 'the original ..
00001010: 62 69 6E 61 72 79 20 66 | 69 6C 69 27 15 17 21 2F | binary file'....
00001020: C7 77 C4 C7 46 1C 03 00 | 00 00 EB BF C7 46 1C 04 | ЗwДЗF.....лїЗF..
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. .. | ................
```

```
ApplicationPlus.exe
00000000: 4D 5A 90 00 03 00 00 00 | 04 00 00 00 FF FF 00 00 | MZP.........яя..
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. .. | ................
00000FF0: 46 3C 59 59 83 C8 01 8B | CE 89 46 3C FF 15 07 01 | F<YY.И.<O%F<я...
00001000: 27 63 68 61 6E 67 65 64 | 20 62 69 6E 61 72 79 20 | 'changed binary
00001010: 66 69 6C 65 27 15 17 21 | 2F C7 77 C4 C7 46 1C 03 | file'....ЗwДЗF..
00001020: 00 00 00 EB BF C7 46 1C | 04 02 16 46 3C 46 1A 04 | ...лїЗF...F<3F..
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. .. | ................
```

```
File Version Information :

Version language            : Английский (США)

     CompanyName            : Company Inc.
     FileDescription        : Application Plus
     FileVersion            : 1.2.0.0
     InternalName           : ApplicationPlus
     LegalCopyright         : Copyright © 2017 Company Inc.
     OriginalFilename       : ApplicationPlus.exe
     ProductName            : Application Plus
     ProductVersion         : 1.2

Creation Date               : 25/01/2017  09:12:38
Last Modif. Date            : 25/01/2017  09:12:38
Last Access Date            : 07/03/2018  17:03:42
FileSize                    : 3382302 bytes ( 3303.000 KB )
FileVersionInfoSize         : 1524 bytes
File type                   : Application (0x1)
File/Product version        : 1.2.0.0 / 1.2.0.0
```

Figure 2 – Software identification for built-in MI software

or there is a constructive (hardware or software) blocking of microcontroller software overwriting, that is, there is no possibility of interference with software without opening the case (breakdown of the device's safety seal), then such software is considered to be unaltered during operation.

Basic requirements for bilt-in MI software

Unchanged software is considered to be the hardware unit of the device and is considered in the same volume as other components, nodes, elements of the electrical circuit, etc. The software documents submitted for conducting the tests for compliance with the requirements of the technical project must contain, including the justification for assigning the software to the same.

Software should be designed in such a way as to ensure maximum suitability for the correct application of the MI, including:

not having un-documented functions or commands that may affect the metrological characteristics or performance of the MI;

eliminate intentional or unintentional actions of the user or others through interfaces that may distort measurement results.

These requirements apply to the main embedded software, as well as to additional or auxiliary software, if it is used by the MI and may affect the measurement results, storage, or transmission of measurement results while operating this tool. Additional (auxiliary) software must also be identified and protected, that is, it needs to perform the same testing as for the main software. Software for MI must provide all necessary information regarding the identification of the software, the measures taken to ensure the security and suitability of the software.

Main features of identification of the bilt-in MI software

It is unimaginable to consider in more detail the identification, security and applicability of MI software. Software identification is a sequence of characters that characterizes this software, is clearly associated with it and distinguishes it from software.

The main purpose of identifying common-use software is the protection of the copyright of the developer, which specifies the following data: the name of the developer, the trademark, the product name, the version of the product, etc. (Figure 2). For some developers, especially if the software is freeware, copyright protection is not important, so they do not specify copyright information.

For software, MI identification is used to verify and confirm the integrity and integrity of the software. For this purpose only the author's information is not enough, the possible change of the code of the array or files in no way affects the author's information. The identification should be such that it would be possible to track software changes, that is, each change should lead to a change of identification. It is possible to implement the calculation of the checksum using the algorithm of the cyclic redundancy code or the hashing of a specific data block. Any change to the software will change the software code. This will be the result of interference in MI.

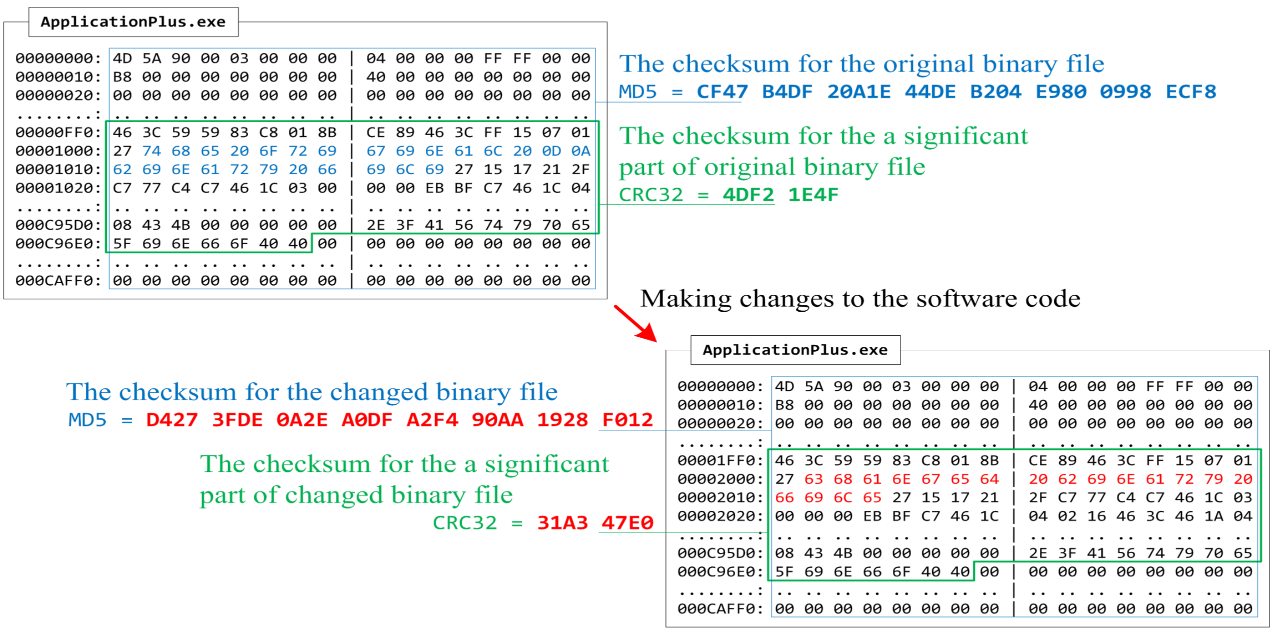
For example, firmware microprocessor for embedded software, legally significant part of the file or the entire files is used (Figure 3). When any significant part of the software was changed, the value of the checksum was changed.

For built-in MI software, the use of cyclic redundancy code CRC-16 or CRC-32 is considered sufficient because:

its implementation is relatively simple;

does not require a lot of resources;

the checksum function can also be used to control the integrity of the data being transmitted.



ApplicationPlus.exe

```
00000000: 4D 5A 90 00 03 00 00 00 | 04 00 00 00 FF FF 00 00
00000010: B8 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
00000FF0: 46 3C 59 59 83 C8 01 8B | CE 89 46 3C FF 15 07 01
00001000: 27 74 68 65 20 6F 72 69 | 67 69 6E 61 6C 20 0D 0A
00001010: 62 69 6E 61 72 79 20 66 | 69 6C 69 27 15 17 21 2F
00001020: C7 77 C4 C7 46 1C 03 00 | 00 00 EB BF C7 46 1C 04
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
000C95D0: 08 43 4B 00 00 00 00 00 | 2E 3F 41 56 74 79 70 65
000C96E0: 5F 69 6E 66 6F 40 40 00 | 00 00 00 00 00 00 00 00
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
000CAFF0: 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00
```

The checksum for the original binary file
MD5 = CF47 B4DF 20A1E 44DE B204 E980 0998 ECF8

The checksum for the a significant
part of original binary file
CRC32 = 4DF2 1E4F

Making changes to the software code

ApplicationPlus.exe

```
00000000: 4D 5A 90 00 03 00 00 00 | 04 00 00 00 FF FF 00 00
00000010: B8 00 00 00 00 00 00 00 | 40 00 00 00 00 00 00 00
00000020: 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
00001FF0: 46 3C 59 59 83 C8 01 8B | CE 89 46 3C FF 15 07 01
00002000: 27 63 68 61 6E 67 65 64 | 20 62 69 6E 61 72 79 20
00002010: 66 69 6C 65 27 15 17 21 | 2F C7 77 C4 C7 46 1C 03
00002020: 00 00 00 EB BF C7 46 1C | 04 02 16 46 3C 46 1A 04
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
000C95D0: 08 43 4B 00 00 00 00 00 | 2E 3F 41 56 74 79 70 65
000C96E0: 5F 69 6E 66 6F 40 40 00 | 00 00 00 00 00 00 00 00
........: .. .. .. .. .. .. .. .. | .. .. .. .. .. .. .. ..
000CAFF0: 00 00 00 00 00 00 00 00 | 00 00 00 00 00 00 00 00
```

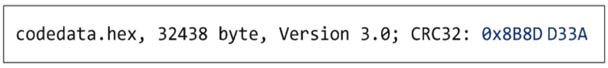
The checksum for the changed binary file
MD5 = D427 3FDE 0A2E A0DF A2F4 90AA 1928 F012

The checksum for the a significant
part of changed binary file
CRC32 = 31A3 47E0

Figure 3 – Software code change indication for built-in MI software

For MI with an operating system, use of CRC is not appropriate; since it has a low level of counterfeit protection it is possible to change the code so that the checksum remains unchanged. For such tools, use hash functions, such as MD5, SHA-1, SHA-2, and RIPEMD-256.

Software identification may include the name of the software product, version, file name and file size or data array, checksum, and algorithm used (Figure 4).



```
codedata.hex, 32438 byte, Version 3.0; CRC32: 0x8B8D D33A
```

Figure 4 – Program identification option.

The choice of the algorithm to calculate the checksum depends on the hardware implementation and the adequate level of protection of components of the software from forgery.

Producers of MI with software, software developers should provide the ability to display identification when the device is turned on or by a user command. In the absence of indication or printer, it should be possible to read the identification through the interface.

As an exception, the identification may be indicated on the marking of MI, if it is not possible to display the value of the identifier on the display or there is no display, there is no communication interface, and after the device is made the software change is not possible or possible when replacing the hardware part. The software that has a specific identifier is identified.

Software protection in the broadest sense is a set of measures aimed at preventing the unauthorized use, study, distribution and modification of software, as well as protection against accidental interference. For MI software, protection against unauthorized modification of software and its components, measurement data, protection against unintentional and accidental interventions is important, namely:

source software code;

measuring data from measuring system sensors;

command entered by the user;

measurement data displayed on the display;

measuring data and gauge coefficients stored in the device’s long-term memory;

measurement data transmitted through communication channels.

Main requirements to the bilt-in MI software testing

An extended classification of software testing approaches and methods based on test design approaches is considered in [14]. The document OIML D 31 [1] sets the general requirements for testing of MI with special software. The requirements of the document do not cover all the technical requirements that are individual for each type of MI. In particular, some indicators that are tested for MI software based on universal computers are not tested for MI with built-in software, as they are already a priori taken into account during design.

Therefore, it is necessary to consider the most critical requirements of document OIML D 31 [1], in particular: requirements for software protection against prevention of accidental misuse of MI and authorization protection.

The document OIML D 31 provides an approval procedure for the MI type and the methods and software testing software depending on the established risk level. These risks are described in detail in [7]. In order to specify the requirements for software and to provide testing methods, it is necessary to use additional requirements of document OIML D 31.

The documentation provided by the manufacturer of the MI (software developer) under the approved type must contain information sufficient to verify compliance with the requirements of the OIML D 31 document. This documentation should allow the development of specific test methods for the MI built-in software. The software must be designed in such a way as to ensure the best fit for the correct application of the MI.

Conclusion

An analysis of the peculiarities of MIs with the built-in software was carried out. For this purpose, all the main components of such a MI are presented. The possibilities of providing the built-in software security as well as MIs in general are considered.

Different variants of placement of software in the MI and features of its testing are considered. Characteristics for software of MIs in accordance with international documents are set testing of which a necessary to provide in the methods for specific MI with the built-in software.

A feature of the MI software is the need for reliable protection against unauthorized modification of software and its components, measurement data, protection against unintentional and random interventions.

Consequently, it is necessary to take into account such the most critical requirements of document OIML D 31 and WELMEC 7.2 guide as requirements for software protection against prevention of accidental interference in MI and authorization protection.

References

1. OIML D 31. General Requirements for Software Controlled Measuring Instruments. OIML. 2019. 63 p. URL: https://www.oiml.org/en/files/pdf_d/d031-e19.pdf.

2. COOMET R/LM/10. COOMET Recommendation. Software for Measuring Instruments: General Technical Specifications. COOMET. 2004. 10 p.

3. WELMEC 7.1. Informative Document. Development of Software Requirements. WELMEC. 2005. 47 p. URL: http://www.welmec.org/fileadmin/user_files/publications/WG_07/7-1_FRPO.pdf.

4. WELMEC 7.2. Software Guide (Measuring Instruments Directive 2014/32/EU). WELMEC. 2019. 132 p. URL: https://www.welmec.org/fileadmin/user_files/publications/WG_07/Guides/WELMEC_Guide_7.2_Software_Guide_2019.pdf.

5. Velichko O. N. Normative base for certification of measurement provision software. Measurement Techniques. 2007. 50 (4). P. 364–371.

6. Velichko O. N. Basic tests, stages, and features in monitoring measuring instrument software. Measurement Techniques. 2009. 52 (6). P. 566–571.

7. Velychko O., Gordiyenko T., Hrabovskyi O. Testing of measurement instrument software on the national level. Eastern-European Journal of Enterprise Technologies. Information and controlling systems. 2018. 2/9(92). P. 13–20.

8. Velychko O., Gaman V., Gordiyenko T., Hrabovskyi O. Testing of measurement instrument software with the purpose of conformity assessment. Eastern-European Journal of Enterprise Technologies. Information and controlling systems. 2019. 1/9(97). P. 19–26.

9. Peters D., Grottker U., Thiel F., Peter M., Seifert J.-P. Achieving Software Security for Measuring Instruments under Legal Control. Position Papers of the Federated Conference on Computer Science and Information Systems. 2014. 3. P. 123–130.

10.Esche M., Thiel F. Software Risk Assessment for Measuring Instruments in Legal Metrology. Proceedings of the Federated Conference on Computer Science and Information Systems. 2015. 5. P. 1113–1123.

11.Sadiq M., Khalid I. R., Mohd W. A., Jung S. Software risk assessment and evaluation process (SRAEP) using model based approach. International Conference on Networking and Information Technology. 2010. 7 p.

12.Peters D., Peter M., Seifert J.-P., Thiel F. A Secure System Architecture for Measuring Instruments in legal metrology. Computers Open Access Journal. 2015. 4. P. 61–86.

13.Thiel F., Grottker U., Richter D. The challenge for legal metrology of operating systems embedded in measuring instruments. OIML Bull. 2011. 52(1). P 5–14.

14. SWEBOK. Guide to the Software Engineering Body of Knowledge. Version 3.0. IEEE Computer Society, 2014. 335 p.

Надійшла до редакції 24.06.2020