Training Laboratories with Online Access on the ITMO.cLAB Platform

Alexey Platunov[a], Arkady Kluchev[a], Vasiliy Pinkevich[a], Vladislav Kluchev[b] and Maxim Kolchurin[a]

[a] *ITMO University, Kronverksky prospekt 49, Saint Petersburg, 197101, Russian Federation*
[b] *LMT Ltd., Birzhevaya liniya 16, Saint Petersburg, 199034, Russian Federation*

Abstract

In the era of total digitalization, an important application area is the creation of platforms for remote educational and scientific laboratory research on real equipment. The online laboratory ITMO.cLAB is a multi-level infrastructure that integrates laboratory equipment for various purposes (from microcontrollers and FPGAs to analytical instruments and technological units) with cloud services for the preparation, performing a full-scale and mixed experiment, with simulation and modeling tools, information and methodological databases, logs of experiments and checkpoint tests. The platform is actively developing. Now it supports laboratory equipment from two very different application areas: embedded computing and thermophysical experiment.

Keywords

remote learning, remote laboratory, virtual laboratory, learning situation, embedded systems, laboratory equipment, physical experiment

1. Introduction

Using the achievements of modern info-communication technologies, many systems for remote communication, training, and various services appear and develop in the world. These services have become firmly established in many areas of people's lives and are taken for granted. Their absence is perceived as an anachronism.

One of these areas is higher education. The classical learning process ties students to a specific place and time of classes. At the same time, modern technologies allow organizing almost complete remote communication between teachers and students. At the same time, a huge number of video courses, books, articles, and various blogs on almost any topic are available to students online. This significantly reduces the value of full-time attendance at lectures, as well as (to a lesser extent) practical and laboratory studies.

In addition to live communication with the teacher (professor), the student needs to access the laboratory equipment. Let us consider the arising problems using the example of studying

---

Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, December 10–11, 2020, Online & Saint Petersburg, Russia

aeplatunov@itmo.ru (A. Platunov); kluchev@itmo.ru (A. Kluchev); vupinkevich@itmo.ru (V. Pinkevich); vakluchev@gmail.com (V. Kluchev); maxim.kolchurin@gmail.com (M. Kolchurin)
0000-0003-3003-3949 (A. Platunov); 0000-0002-3892-8424 (A. Kluchev); 0000-0002-8635-5026 (V. Pinkevich); 0000-0002-3052-2200 (V. Kluchev); 0000-0002-7061-9357 (M. Kolchurin)

CEUR Workshop Proceedings (CEUR-WS.org)

embedded computing systems.

The small and cheap development boards that could be delivered to every student are too simple for any complicated laboratory work. The required devices are quite complex and expensive, and also not so widespread. In some cases, real training devices can be replaced with simulators. However, even in this case, distance learning faces the following problems:

1. Licenses for proprietary simulators (even academic ones) are expensive. It is relatively easy to equip a computer lab with such licenses, but it will be problematic to provide each student with a license for remote work.
2. Free and open-source simulators, for example, QEMU [1], or university developments, such as GEM-5 [2], require a lot of computing resources. Also, these simulators usually require ad-hoc extensions to solve specific educational tasks.
3. Simulators, even simulating the device as a whole [3], give an "ideal" model of execution. A real device always has hardware errors or is prone to interference, etc., which makes its programming much more realistic.

Similar problems exist in other groups of technical and natural sciences. In any case, the approaches considered above do not allow each student to deploy their own full-fledged substitution of a university laboratory.

Having this, we think that the best way to organize remote laboratory work is to provide remote access to real laboratory equipment for a large number of users. The relevance of this approach is confirmed by the presence of actively developing solutions in the field of creating remote laboratories both for the study of information technologies [4, 5] and in other areas [6, 7].

2. ITMO.cLAB cloud laboratory

We have developed the ITMO.cLAB cloud lab as a service to provide students and teachers with remote access to educational laboratory equipment anytime and anywhere. Any device with an internet browser can be used, including laptops, tablets, and even smartphones.

ITMO.cLAB is designed to solve the following range of educational, scientific, and technical problems:

1. Providing hardware access for conducting ITMO distance online courses with a lot of students (thousands of people).
2. Providing ITMO students with laboratory access for conducting experiments during scientific and pre-diploma practices.
3. Supporting students' scientific and technical creativity in various informal ITMO associations.
4. Providing access to laboratory equipment on a commercial basis for universities, colleges, school clubs, special interest groups, commercial companies, and start-ups, as well as DIY enthusiasts.

“Virtual laboratory” allows you to carry out lab work and coursework, implement the practical experiments for a graduation project, set up experiments during scientific research, or test various ideas.

Fields of ITMO.cLAB applications:

1. Support for training courses and research related to the design of modern computing systems: embedded and cyber-physical systems, Internet of Things-based systems, Industry 4.0 systems, etc.
2. Support for training courses and research in natural sciences (for example, lab works in thermal physics, optics, etc.).

One of the key features of ITMO.cLAB is support for different scenarios of working with laboratory equipment:

1. Batch processing of requests for access to equipment. The scenario can be implemented if the experiment is carried out in a fully automatic way. Applications with the necessary input data are queued and processed as the equipment becomes available. This achieves maximum equipment utilization with relatively short processing waiting times. Depending on the duration of one experiment, the number of equipment installations can be several times or tens of times less than the number of users simultaneously working with them. This is much better than can be achieved with access to equipment via a “remote desktop”.
2. Real-time access. This scenario is implemented if there is a need for interactive interaction with the laboratory equipment during the experiment. The user is provided with information about the experiment (values, graphs) in real-time and handles to control the installation.

3. ITMO.cLAB architecture

ITMO.cLAB includes three levels of components: client, infrastructure, and executive levels. The core of the system is the infrastructure layer, which provides (Fig. 1):

- registration and authentication of users;
- access for different categories of users to the necessary system components;
- resources sharing during laboratory equipment access (queues of requests, batch processing);
- storage and processing of data about the equipment access sessions (for example, automatic testing of the students’ lab assignments).

The infrastructure level is an information system that is hosted on an Internet server or in the “cloud”. Some of the components of the client and physical layers are also in the "cloud".

The "cloud" part of ITMO.cLAB uses a microservice architecture, which allows the flexible scalability of the system by distributing individual system components across different servers.

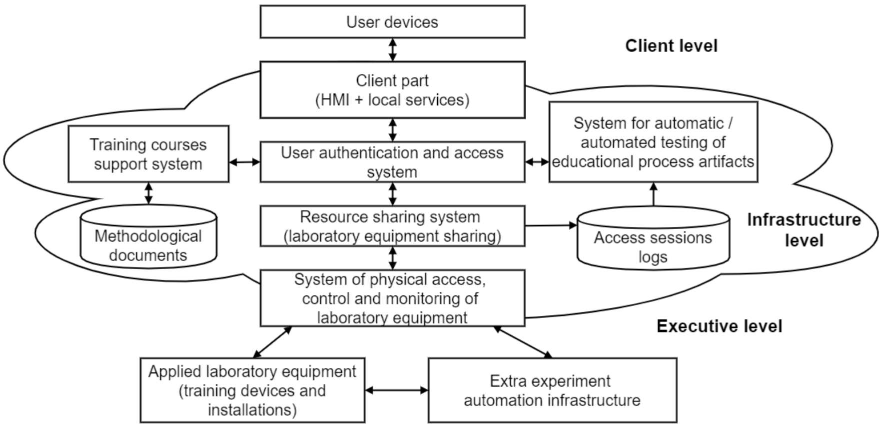


Figure 1: ITMO.cLAB organization in the applied viewpoint

The number of users depends on the allocated computing resources and can reach several thousand people.

The client level is the user interface. Access is available via an internet browser, so the client device requirements are minimal. The executive level includes laboratory equipment connected to ITMO.cLAB and additional hardware and / or software components that allow remote control (for example, turning on / off the power, loading software into the device, etc.) and debugging (including step-by-step at the source level, nested debug, using special test agents and logging tools) [8, 9, 10]. In addition, it includes information system components that support a particular type of equipment.

With the modular architecture, new components can be added to ITMO.cLAB to support a variety of laboratory equipment and simulators. In addition to physical equipment, a variety of simulators can be connected at the executive level. The throughput of the system at this level is easily scalable when connecting new instances of laboratory equipment.

Now the prototype of the ITMO.cLAB system has been implemented, which supports connection of the SDK-1.1M training microprocessor devices. These devices are used during courses at the Faculty of Software Engineering and Computer Systems related to embedded systems. Work with devices is carried out in batch mode. Also, integrating the TFK-4.0U thermophysical controllers is in progress. These controllers are used for courses in the physics of thermal processes at the Faculty of Cryogenic Engineering. The TFK-4.0U controller will be used in real-time mode.

ITMO.cLAB is currently deployed in test mode on one of the servers in ITMO Technopark. According to the technical characteristics of the server equipment, the current prototype allows connecting 100-200 students simultaneously.

Further in the paper, the hardware and software of the current version of the ITMO.cLAB system will be considered in more detail.

4. Laboratory cloud infrastructure

ITMO.cLAB is implemented as a web application and has a web interface for browser access. The advantage of such a solution is the possibility of remote access from anywhere in the world from any platform (Windows, Mac OS, Linux, or iOS and Android).

The cloud infrastructure is divided into front-end and back-end. The web interface is implemented using the Vue.js Javascript framework using the Vuetify.js graphics library. The selected set of technologies allows you to create an adaptive user interface that works equally well on any device.

The backend uses such technologies as REST API, Kotlin, Ktor. To ensure modularity and scalability of the system, its functionality is divided into microservices. Examples of tasks solved by microservices are:

1. Maintenance of one pool of educational laboratory devices.
2. Serving tasks for pools of educational laboratory devices.
3. Maintenance of a database with educational documentation and tests.
4. Monitoring microservices and collecting event logs.

In the future, it is planned to add new functions related to project management: source code repository, bug tracker, continuous integration system, project management system, etc., which will systematize the educational process and provide students and workgroups with free services.

In the current version of ITMO.cLAB, three types of users are implemented: student, teacher, and administrator. The student has access to educational literature and documentation, can take tests connected to lab work, and also has remote access to laboratory equipment. Depending on the educational course, a specific type of laboratory equipment is available to the student. Thus, students studying embedded systems will work with the SDK-1.1M, and physics students will work with the TFK-4.0U thermophysical controller.

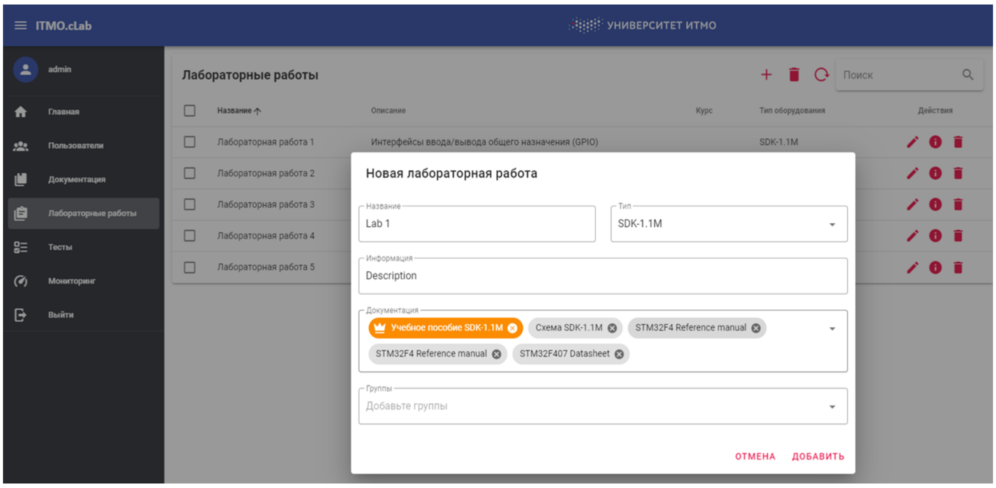
The teacher's responsibilities include interacting with student users and their groups, namely: adding, editing, deleting. Also, the teacher can create new laboratory work, tests, documentation, and assign them to study groups (Fig. 2).

The ITMO.cLAB system provides teachers with the ability to record and control the execution of test assignments and laboratory work. For example, a teacher can see in real-time what stage of laboratory work a student is at. If the laboratory work is performed using TFK-4.0U, the teacher can view the execution of the experiment of any student. In the case of performing work on the SDK-1.1M, the teacher can see information about the queue for downloading programs to controllers.

5. Laboratory equipment

5.1. SDK-1.1M

SDK-1.1M is a complex platform for technical creativity and study of electronics produced by LMT Ltd. and developed jointly with the professors of the Faculty of Software Engineering and



Лабораторные работы

☐	Название	Описание	Курс	Тип оборудования	Действия
☐	Лабораторная работа 1	Интерфейсы ввода/вывода общего назначения (GPIO)		SDK-1.1M	
☐	Лабораторная работа 2				
☐	Лабораторная работа 3				
☐	Лабораторная работа 4				
☐	Лабораторная работа 5				

Новая лабораторная работа

Название: Lab 1
Тип: SDK-1.1M
Информация: Description
Документация: Учебное пособие SDK-1.1M, Схема SDK-1.1M, STM32F4 Reference manual, STM32F4 Reference manual, STM32F407 Datasheet
Группы: Добавьте группы

Figure 2: The example of creating a laboratory work assignment

Computer Systems of ITMO University [11, 12]. It is used during classes on computer science, embedded and cyber-physical systems design.

Unlike boards that can be purchased for little money in numerous online stores, SDK-1.1M's technical solutions meet the standards of industrial controllers. This allows students to get familiar with real modern microprocessor systems, as well as use the SDK-1.1M as a key element of a "smart home" system or robots of various types and purposes. Also, SDK-1.1M has methodological support and example projects that will help in the study of microprocessor technology.

SDK-1.1M has a modular architecture. There are several options for its computing core including microcontrollers and microprocessors of various complexity. Also, there are expansion slots for connecting boards with own SDK-X form factor and expansion boards of the Arduino standard.

The basic model SDK-1.1MC.407 is based on the 32-bit STM32F407VG microcontroller with an ARM architecture processor. SDK-1.1M has a wide range of external peripherals and interfaces (Fig. 3) including external EEPROM 1 Kbit, real-time clock, OLED display, accelerometer, electromagnetic sound emitter, a set of signal LEDs (green, yellow, red), I2C keyboard with 12 buttons, MicroSD-card slot, UART, RS-485, Ethernet 10/100M, USB.

When organizing remote access to microprocessor devices, the following main options are available:

1. Limited access to device hardware. In-system programming is used (the microcontroller has a resident bootloader-monitor) and no external equipment is required to control the hardware.
2. No restrictions on access to hardware capabilities. Requires external equipment to load the program into the device and restart it, as well as a test harness for working with external interfaces.
3. Extended external environment: connecting logic analyzers, combining several devices, etc.

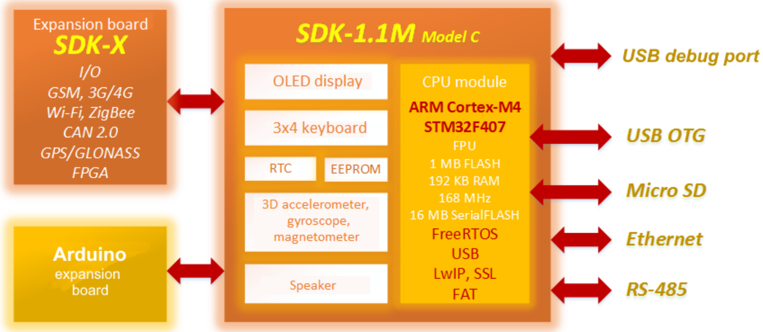


Figure 3: Organization of the SDK-1.1M

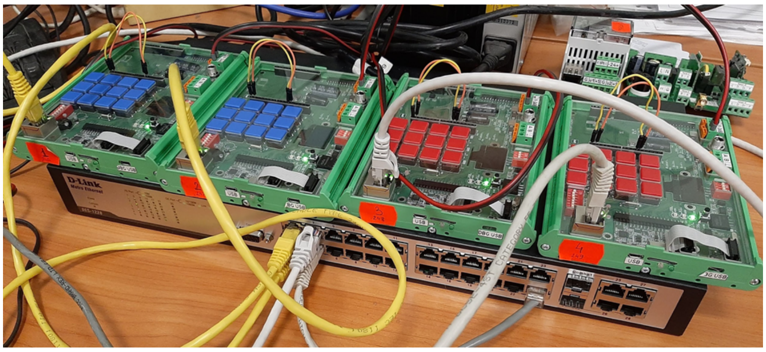


Figure 4: Pool of SDK-1.1M controllers connected to ITMO.cLAB

The first option was chosen as the base one since it provides fair enough capabilities for the educational process purposes, but seriously simplifies the requirements for the hardware of the virtual laboratory. However, an expansion module with an integrated logic analyzer has been developed, and it is planned to support it within ITMO.cLAB for advanced debugging when working with discrete signals, I2C, SPI, UART interfaces, etc.

The device is connected to the network via the Ethernet interface (Fig. 4). A special bootloader software was developed for the device, which is connected to the ITMO.cLAB server. The MQTT protocol is used to communicate with the server.

The bootloader accepts execution requests, which include a software image for the microcontroller. The loader uses the In-Application-Programming method: reprogramming is performed through one of the application interfaces using a program running in the microcontroller's own memory. Since the built-in FLASH memory of the microcontroller has a limited rewriting resource, the user program is placed and executed in RAM.

One of the main problems with remote programming of microcontrollers in this scenario is the inability to visually observe the results of the loaded program execution (for example, the blinking of an LED or a waveform on an oscilloscope). To solve this problem, an event tracing library was developed. It provides the ability to record three types of events: discrete, string, and data. Discrete events are timestamps in microseconds with an event ID and value (1 or 0).

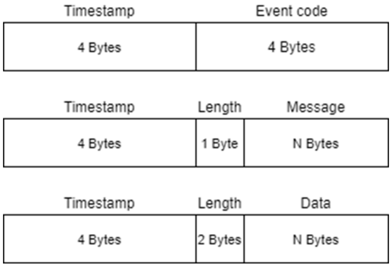


Figure 5: Structures of the event trace records

```
/****** SDK-1.1M Trace start (14.11.2020 15:20:42) ******/
 250.802:  Start of the first cycle
 1505.801: End of first cycle
 1806.802: Start of the second cycle
 4962.801: End of second cycle
 /***************** SDK-1.1M Trace stop *****************/
```

Figure 6: Visualization of user program execution trace on SDK-1.1M

Lines and data are written to memory in the following order: timestamp, length, data. For each type of event, 20 KB of CCMRAM-memory of the microcontroller is allocated. Each event type has its own packaging structure in memory. Discrete events are recorded in pairs of 4 bytes: timestamp and event code. Lines and data are written in the following order: timestamp (4 bytes), 1 or 2 bytes of length, and N bytes of a string or data (Fig. 5).

After completing the trace, the CRC32 is calculated and the microcontroller is reset to bootloader mode. 8 seconds are given for the execution of the user program (this time is limited by the watchdog timer). If the user program does not call the trace termination function, or if the execution took more than 8 seconds, it is reset to the bootloader without saving the trace information.

The trace buffer is read from the controller immediately after a reset. The received data can be interpreted into text logs and diagrams in a web application (Fig. 6).

The algorithm of a student's work in the laboratory is as follows:

1. Download the project template for SDK-1.1M from the repository on GitHub.
2. Develop a program according to the assignment.
3. Compiled the project and generate an executable image.
4. Upload the received image via the web interface.
5. Get a text log and an event diagram.

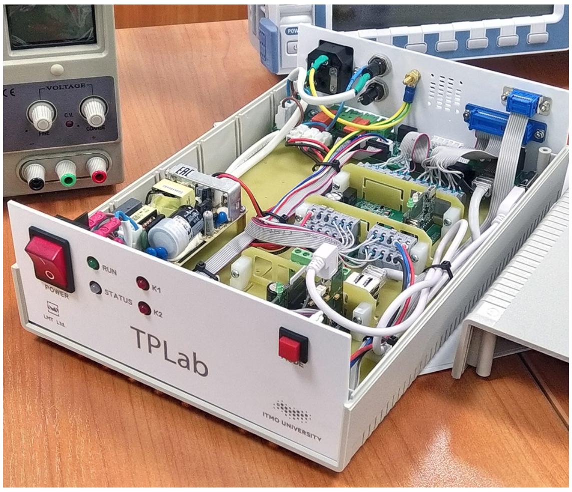


Figure 7: The TFK-4.0U controller

5.2. TPLab

TPLab thermophysical laboratory is a hardware / software complex integrated with the ITMO.cLAB cloud laboratory, which can be used to solve a wide range of problems in the field of thermophysical experiments of a technological, research, and educational nature. The use of a cloud laboratory offers many advantages, such as creating a network of controllers to collect data from remote objects, using the controller under the control of portable personal devices without the need for physical connection to the device, remote monitoring, and organization of remote laboratory work.

The TPLab thermophysical laboratory uses the TFK-4.0U controller [13] (Fig. 7) as a measuring device. It is developed by LMT Ltd. together with the professors of the School of Biotechnology and Cryogenic Systems at ITMO University. It is a part of the TFK family of controllers, which have proven themselves well as devices for educational and industrial applications [14, 15].

The TFK-4.0U controller is built with a modular principle using the LMTFusion platform of distributed infrastructure and industrial automation produced by LMT Ltd. [12] The possibility of the simultaneous connection of temperature sensors of various types is provided, which is necessary for studying thermal processes, properties of temperature sensors, for performing precision measurements of thermophysical characteristics. Also, the TFK-4.0U controller can control several channels of heaters and other actuators for the automatic execution of experiments (Fig. 8).

To measure physical characteristics a TFK-4.0U uses an MA842 analog input-output module, to which thermocouples, resistance temperature detectors (RTDs), and digital temperature sensors of the DS1820 type are connected. The SCG-3.6 controller is used as a communication module, which provides interaction with user terminals and external Internet services, including ITMO.cLAB, provides an interface for I / O expansion modules.

The measuring range of thermo-electromotive force in the controller allows using most stan-

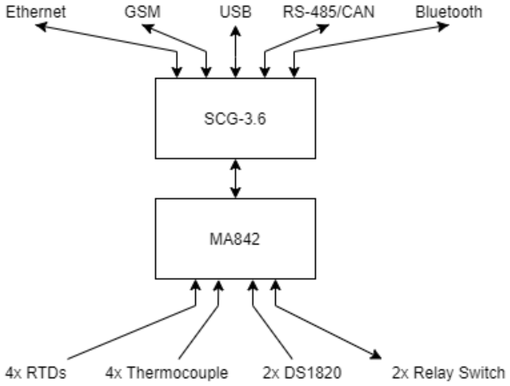


Figure 8: Structure and external interfaces of the TFK-4.0U controller

dard and custom types of thermocouples. To ensure the required resolution used series connection of the differential amplifiers and 18-bit multi-channel analog-to-digital converter (ADC). To ensure operation with resistance thermometers, a digital-to-analog converter (DAC), a high-precision voltage-to-current converter, and an analog multiplexer are used. This circuit allows measuring resistance up to 700 ohms.

6. Conclusion

The organization of educational and scientific laboratories with remote access using real equipment hardware is a complex and relevant segment of activities at the connection of sociocyberphysical systems, educational technologies, experiment automation, and numerous applied scientific and technical fields. The development of such laboratory complexes and platforms requires the study and formalization of known and new scenarios of use, as well as scenarios of the behavior of various categories of users (students, teachers, researchers, laboratory assistants and service personnel, system administrators) [16]. Another area of work is the development of means and technologies for remotely conducting an experiment, which requires the division of terminal laboratory equipment and usage scenarios into some categories and the proposal of a reasonable number of unified solutions. Another area to be considered is the development of automated support and analysis tools for remote educational laboratory activities and scientific experiments, as well as the creation of special methods for remote group project activities of students.

The team of authors plans to develop the present work, first of all, in the second direction and is interested in cooperation with colleagues to cover all areas in such an important and complex area as remote educational and scientific laboratories.

References

[1] QEMU processor emulator. URL: https://www.qemu.org/ (acessed 01.11.2020)
[2] GEM5 simulator. URL: https://www.gem5.org/ (acessed 01.11.2020)

[3] Proteus Simulator. URL: https://www.labcenter.com/ (acessed 01.11.2020)

[4] S. Raivo, S. Sven. Learning Situations and Remote Labs in Embedded System Education. 14th International Workshop on Research and Education in Mechatronics, 2013.

[5] E. Bodegom, E. Jensen, D. R. Sokoloff. Adapting RealTime Physics for Distance Learning with the IOLab. The Physics Teacher, 2019, vol. 57, no. 6, pp. 382-386.

[6] Pastor Vargas, R. et al. Structured remote laboratory development. Technologies Applied to Electronics Teaching (TAEE), 2012. 314-319. 10.1109/TAEE.2012.6235457.

[7] Long J., Chenery K., Stannard, W. Physics Practicals for Distance Education in an Undergraduate Engineering Course. Proceedings of the 2012 AAEE Conference, 2012

[8] Kluchev A.O., Makovetskaya N.A. Problems of testing system software for distributed information and control systems. Scientific and Technical Journal of SPb SITMO(TU), 2003, vol. 3, no. 4 (10), pp. 67–70 (in Russian).

[9] Bolgarov I.S., Patiev A.A., Kluchev A.O. Resident and service software for industrial network I/O controllers. Scientific and technical Journal of SPb SITMO(TU), 2003, vol. 3, no. 4 (10), pp. 30–33 (in Russian).

[10] Pinkevich V., Platunov A. Model-Driven Functional Testing of Cyber-Physical Systems Using Deterministic Replay Techniques. Proceedings 2018 IEEE Industrial Cyber-Physical Systems (ICPS) - 2018, pp. 141-146

[11] Kluchev A.O., Platunov A.E., Dergachev A.M. Application of laboratory stands SDK in education. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2019, vol. 19, no. 1, pp. 184–187 (in Russian). doi: 10.17586/2226-1494-2019-19-1-184-187

[12] LMT Ltd. URL: https://lmt.spb.ru (acessed 01.11.2020)

[13] Baranov I., Tambulatova E., Platunov A., Kluchev A., Pluzhnikova D., Krylov V., Zhovnitsky V., Kluchev V. Automated Training Laboratory Bench for Studies of Thermophysical Properties and Thermal Processes Based on a Programmable Controller SDK-1.1M. CEUR Workshop Proceedings - 2020, Vol. 2590, pp. 1-8

[14] E. S. Platunov, I. V. Baranov, A. E. Platunov. A complex of automated devices for thermal measurements. Scientific Instrumentation Journal, 2003, vol. 13, no. 3, pp. 19–24 (in Russian).

[15] E. S. Platunov, I. V. Baranov, A. E. Platunov. The thermophysical laboratory of low temperatures. Journal of Instrument Engineering. 2009. Vol. 52, N 5. P. 65-68 (in Russian).

[16] Boldyreva E.A., Penskoi A.V., Platunov A.E. Formalization and evolution of learning path in embedded systems. Wave Electronics and its Application in Information and Telecommunication Systems (WECONF 2020) - 2020, pp. 9131496.