Internet of Things for Industry and Human Applications

Internet of Things for Smart Building and City

TRAININGS

Funded by the
Erasmus+ Programme
of the European Union

Internet of Things for Industry and Human Applications
Internet of Things for Smart Building and City

Ministry of Education and Science of Ukraine
Odessa National Polytechnic University
Zaporizhzhia National Technical University
National Aerospace University "Kharkiv Aviation Institute"

O. A. Boiko, V. V. Busher, O. V. Drozd, D. A. Maevsky,
O. Ju. Maevskaya, O. M. Martynyuk, A. V. Parkhomenko,
O. M. Gladkova, M. O. Drozd, O. M. Ivanova, S. S. Surkov,
K. V. Zashcholkin

Internet of Things for Industry and Human Applications

Internet of Things for Smart Building and City

Trainings

Edited by D. A. Maevsky

Project
ERASMUS+ ALIOT 73818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP
Internet of Thing: Emerging Curriculum for Industry and Human Applications

2019

UDC 004.415/.416:71/72](076.5)=111
I-73

Reviewers:
Dr. Ah-Lian Kor, Leeds Beckett University, United Kingdom
Prof., DrS. Volodymyr Opanasenko, State Prize Winner of Ukraine, Leading Researcher of the Microprocessor Engineering Department, V.M. Glushkov Institute of Cybernetics of NAS, Ukraine

I-73 Boiko O. A., Busher V. V., Drozd O.V., Maevsky D.A., Maevskaya O.Ju., Martynyuk O.M., Parkhomenko A.V., Gladkova O.M., Drozd M.O., Ivanova O.M., Surkov S.S., Zashcholkin K.V. **Internet of Things for Smart Building and City**: Practicum / Maevsky D.A. (Ed.) – Ministry of Education and Science of Ukraine, Odessa National Polytechnic University, Zaporizhzhia National Technical University, 2019. – 156 p.

ISBN 978-617-7361-94-6.

The materials of the practical part of the study course ITM2 “IoT for Smart building and city”, developed in the framework of the ERASMUS+ ALIOT project “Internet of Things: Emerging Curriculum for Industry and Human Applications” (573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP).

The structure of work on verification of residual knowledge in the discipline, the corresponding practical material, examples of tasks and criteria of evaluation are given. In the learning process, the theoretical aspects of development and implementation of IoT-based systems are presented. The course focuses on the application of the Internet of things technologies in the design and construction of smart building systems.

It is intended for engineers, developers and scientists engaged in the development and implementation of of IoT-based systems, for postgraduate students of universities studying in areas of IoT, cybersecurity in networks, computer science, computer and software engineering, as well as for teachers of relevant courses.

Ref. – 79 items, figures – 66, tables – 6.

Approved by Academic Council of National Aerospace University “Kharkiv Aviation Institute” (record No 4, December 19, 2018).

ISBN 978-617-7361-94-6.

Міністерство освіти і науки України
Одеський національний політехнічний університет
Запорізький національний технічний університет
Національний аерокосмічний університет
ім. М. Є. Жуковського «Харківський Авіаційний Інститут”

Бойко А.О., Бушер В.В., Дрозд О.В., Маєвський Д.А.,
Маєвська О.Ю., Мартинюк О.М., Пархоменко А.В., Гладкова
О.М., Дрозд М.О., Іванова О.М., Сурков С.С., Защолкін К.В.

Інтернет речей
для
індустріальних і гуманітарних застосунків

ІНТЕРНЕТ РЕЧЕЙ ДЛЯ РОЗУМНОГО БУДИНКУ ТА МІСТА

Тренінги

Редактор Маєвський Д.А.

Проект ERASMUS+ ALIOT
“Інтернет речей: нова освітня програма для потреб
промисловості та суспільства”
(573818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP)

2019

УДК 004.415/.416:71/72](076.5)=111
І-73
Рецензенти: Др. А-Ліан Кор, Leeds Beckett University, Велика Британія
Д.т.н., проф. Володимир Опанасенко, Лауреат Державної премії України, провідний науковий співробітник відділу мікропроцесорної інженерії, Інститут кібернетики ім. В.М.Глушкова НАН України

І-73 Бойко А.О., Бушер В.В., Дрозд О.В., Маєвський Д.А., Маєвська О.Ю., Мартинюк О.М., Пархоменко А.В., Гладкова О.М., Дрозд М.О., Іванова О.М., Сурков С.С., Защолкін К.В. **Інтернет речей для розумного будинку та міста.** Практикум / За ред. Д.А. Маєвського – МОН України, Одеський національний політехнічний університет, Запорізький національний технічний університет, 2019. – 156 с.

ISBN 978-617-7361-94-6.

Практичні матеріали навчального модуля «IoT для розумних споруд та міст», наведені в цій книзі, розроблені в рамках проекту ERASMUS+ ALIOT 73818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP “Internet of Things: Emerging Curriculum for Industry and Human Applications”. Курс зосереджений на застосуванні технологій Інтернету речей при проектуванні та розробці систем розумного будинку та розумного міста. Розглядаються питання оцінки ризику в підсистемах Інтернету речей, розробка програмних та апаратних платформ на базі Arduino та Raspberry Pi. Вивчається також побудова систем розумного будинку на базі промислових мікроконтролерів та програмовної логіки. Курс буде корисний працівникам промислових компаній, які займаються розробкою та впровадженням систем розумного будинку або розумного міста. Книга може бути корисною також студентам університетів та викладачам, які проводять заняття по відповідним курсам.

Бібл. – 79, рисунків – 66, таблиць – 6.

Затверджено Вченою радою Національного аерокосмічного університету «Харківський авіаційний інститут» (запис № 4, грудень 19, 2018).
ISBN 978-617-7361-94-6.

ABBREVIATIONS

AS – Artifical System

CPN – Colored Petri Nets

CTL – Concurrent Temporal Logic

DASD – Direct Access Storage Device

DoS – Denial-of-Service

DDoS – Distributed Denial-of-Service

DFD – Data-flow Diagram

EGS – Evolutionary-Genetic System

EE – Expert Evaluation

ERD – Entity-Relationship Diagram

FoT – Fog of Things

GPSS – General Purpose Simulation System

GUI – Graphical User Interface

HMM – Hidden Markov Model

HSMM - Hidden Semi-Markov Model

IDE – Integrated development environment

IoE – Internet of Everything

IoT – Internet of Things

LTL – Linear Temporal Logic

MAS – Multi-Agent Systems

MQTT – Message Queuing Telemetry Transport

MTBF – Mean Time Between Failures

MTTF – Mean Time to Failure

MTTR – Mean Time to Recover

NLP – Natural Language Processing

OPC – Open Platform Communications

OPC UA – Open Platform Communications Unified Architecture

QS – Queuing System

REST – REpresentational State Transfer

RUL – Remaining Useful Lifetime

SA – Service Available

SBC – Smart Building and City

SBS – Smart Building System

SMR – Service May Recover

SMNR – Service May Not Recover

SNA – Service Not Available

TINA – TIme Petri Net Analyzer

TTF – Time to Failure

TTR – Time to Recover

UML – Universal Modeling Language

UML-CSAS – UML-diagrams of Communications, Sequences, Activity, States

XML – eXtensible Markup Language

INTRODUCTION

The manual presents the materials of the practical part of the industrial training module "IoT for Smart Building and City" (ITM2) prepared by the project ERASMUS+ ALIOT 73818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP “Internet of Things: Emerging Curriculum for Industry and Human Applications”[1].

The manual provides a description of trainings and laboratory works, the purpose of which is to get acquainted with the application of the Internet of things technologies in the design and construction of smart building systems and cities.

The first part presents a workshop on the interaction of the subsystems of the Internet of things. Students should perform subsystem partitioning according to the selected criterion. The method of expert assessments determines the risks for the general system of the Internet of things in case of partial or complete failure of one of the subsystems.

Laboratory works presented in the second part are aimed at the realization of software/hardware platform for Smart Building system. Arduino, Raspberry Pi, OpenHAB platforms as well as various sensors are proposed as a basis for design. Processing IDE, Arduino IDE and OS Raspbian are used as development environments. Third and forth parts describe features of communication and protocols, application of FPGA for development of smart systems.

It is intended for engineers, developers and scientists engaged in the development and implementation of of IoT-based systems, for postgraduate students of universities studying in areas of IoT, cybersecurity in networks, computer science, computer and software engineering, as well as for teachers of relevant courses.The authors are grateful to the reviewers, project colleagues, staff of the departments of academic universities, industrial partners for valuable information, methodological assistance and constructive suggestions that were expressed during the discussion of the course program and the materials of the manual.

[1] *The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

1 HIERARCHY AND INTERACTIONS BETWEEN SMART IOT SYSTEMS

1 Seminar

IoT in Smart Homes and Cities Systems

1 Seminar objectives

The objectives are to provide knowledge and practical skills on:

- The structure and composition of smart home systems
- The use of Internet of things technologies for the functioning of smart home systems
- The concept of smart city. Using IoT in smart city systems

2 Seminar preparation

Seminar preparation includes the following steps.

2.1 Assignment (choice) of report subject (analytical review, SDP) and tasks specification.

The report subject is to be agreed with the lecturer. It can be chosen by students on their own based on the following suggested list (can be extended):

- The general concept of building smart home systems;
- Smart home systems and their interactions;
- Using IoT technologies for the functioning of the security system of smart homes;
- Using the Internet of things for the functioning of the climate control system of a smart home;
- Using IoT technologies for the functioning of the lighting control system in a smart home;
- The use of Internet of things technologies for organizing the interaction of smart home systems;
- The Internet of Things and Cloud Technologies in Smart Homes;

– The Internet of things as a means of organizing the interaction of smart homes;

– General principles of the functioning of smart cities;

– The Internet of things as a means of communication of smart city systems;

– Examples of the use of IoT technologies in smart city systems.

2.2 Search of the information about report subject (library, the Internet, resources from department) and primary analysis. The search of the information is conducted using the such keywords: Smart homes, Home appliances, Internet of Things, Intelligent sensors, Building automation, Smart buildings, Smart cities, Home automation, Sustainable living space, Smart urban infrastructure.

Please use reference list [1-10]. Theoretical issues for IoT based transport systems and additional references are described in Part X (sections 36-39) of the book [11].

2.3 Report and presentation plans development. Report plan includes:

- introduction (relevance, reality challenges, brief analysis of the problem - references, purpose and tasks of the report, structure and contents characteristics);
- systematized description of the main report parts (classification schemes, models characteristics, methods, tools, technologies by groups, selection of indexes and criteria for assessment, comparative studies);
- conclusions (established goal achievement, main theoretical and practical results, result validity, ways of further work on the problem);
- list of references;
- appendixes.

2.4 Report writing. The report should stand for 15-20 A4 pages (font size 14, spacing 1.5) including the title page, contents, main text, list of references, appendixes. Unstructured reports or reports compiled directly from Internet sources (more 50%), having incorrect terms and no conclusion shall not be considered.

The work plan, presentation slides and an electronic version of all material related to the work are required to be included in appendixes.

2.5 Presentation preparation. The presentation is to be designed in PowerPoint and be consistent with the report plan (10-15 slides); the time-frame for the presentation is 15 minutes.

– The presentation should include the slides as follows:

– title slide (specification of the educational institution, department, course of study, report subject, authors, date);

– contents (structure) of the report;

– relevance of the issues covered, the purpose and the tasks of the report based on the relevance analysis;

– slides with the details of the tasks;

– report conclusion;

– list of references;

– testing questions.

Each slide should include headers with the report subject and authors. The contents of the slides should include the keywords, figures, formulas rather than the parts from the report. The information can be presented dynamically.

3 Presentation and defence

The presentation should be given at the seminar for 20 minutes including:

- presentation (10-15 minutes);
- discussion (5-10 minutes).

Time schedule can be specified by lecturer.

4 Report assessment

The work is assessed on the following parameters:

a) report text quality (form and contents),

b) presentation quality (contents and style),

c) report quality (contents, logical composition, timing shared among parts, conclusion),

d) fullness and correctness of the answers.

Each student is given an individual mark for the report and the presentation based on the results and responsibility assignment.

5 Recommended literature

1. Hwaiyu Geng, "SMART HOME SERVICES USING THE INTERNET OF THINGS" in Internet of Things and Data Analytics Handbook, Wiley, 2017, doi: 10.1002/9781119173601.ch37

2. Qusay F. Hassan, "Smart Connected Homes" in Internet of Things A to Z: Technologies and Applications, IEEE, 2018, doi: 10.1002/9781119456735.ch13

3. Houbing Song; Ravi Srinivasan; Tamim Sookoor; Sabina Jeschke, "Building Cyber-Physical Systems – A Smart Building Use Case" in Smart Cities: Foundations, Principles, and Applications, Wiley, 2017, pp.605-639, doi: 10.1002/9781119226444.ch21

4. S. Ghosh, "Smart homes: Architectural and engineering design imperatives for smart city building codes" 2018 Technologies for Smart-City Energy Security and Power (ICSESP), Bhubaneswar, 2018, pp. 1-4. doi: 10.1109/ICSESP.2018.8376676

5. B. Kang, S. Kim, M. Choi, K. Cho, S. Jang and S. Park, "Analysis of Types and Importance of Sensors in Smart Home Services" 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1388-1389. doi: 10.1109/HPCC-SmartCity-DSS.2016.0196

6. X. Mao, K. Li, Z. Zhang and J. Liang, "Design and implementation of a new smart home control system based on internet of things" 2017 International Smart Cities Conference (ISC2), Wuxi, 2017, pp. 1-5. doi: 10.1109/ISC2.2017.8090790

7. S. ur Rehman and V. Gruhn, "An approach to secure smart homes in cyber-physical systems/Internet-of-Things," 2018 Fifth International Conference on Software Defined Systems (SDS), Barcelona, 2018, pp. 126-129. doi: 10.1109/SDS.2018.8370433

8. A. FOUNOUN and A. HAYAR, "Evaluation of the concept of the smart city through local regulation and the importance of local initiative," 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 2018, pp. 1-6. doi: 10.1109/ISC2.2018.8656933

9. E. Mardacany, "Smart cities characteristics: importance of buit environments components" IET Conference on Future Intelligent Cities, London, 2014, pp. 1-6. doi: 10.1049/ic.2014.0045

10. M. Abu-Matar and R. Mizouni, "Variability Modeling for Smart City Reference Architectures" 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 2018, pp. 1-8. doi: 10.1109/ISC2.2018.8656967

11. Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 3. Assessment and Implementation /V. S. Kharchenko (ed.) - Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019. – 740 p.

2 Training

Risk estimation in the IoT system using expert assessment method

1 Training objectives

The objectives are to provide knowledge and practical skills on:

- decomposition of the Internet of things systems into subsystems in accordance with a given criterion;
- preparation and conduct of expert assessment methodologies;
- processing and interpretation of expert assessment results;
- preparation of a report on risk assessments
- preparing a presentation in the PowerPoint environment and presenting it to other listeners.

2 Expert assessment method and its processing. Brief theoretical information

In cases of extreme complexity of the problem, its novelty, lack of available information, the impossibility of mathematical formalization of the solution process have to turn to the recommendations of competent professionals who know the problem that means to experts. Their solution of the problem, argumentation, formation of quantitative estimates, processing by formal methods are called the method of expert assessments. This method is used in following practice part, so it is important to present its theoretical ground.

There are two groups of expert evaluations: individual evaluations that are based on using the opinions of individual experts, independent of each other; collective evaluations based on the use of collective expert opinion.

For evaluation, experts can use the following methods:

1. Ranking is the arrangement of objects in ascending or descending order of any property. Ranking allows you to choose from the studied set of factors the most significant.

2. Pair comparison is the establishment of a preference for objects when comparing all possible pairs. It is not necessary here, as in the ranking, to order all the objects, it is necessary in each of the

pairs to identify the more significant object or to establish their equality.

3. Direct evaluation. It is often desirable not only to order (rank the objects of analysis), but also to determine how much one factor is more significant than the others. In this case, the range of changes of characteristics of the object is divided into separate intervals, each of which is assigned a specific score (point), for example, from 0 to 10. That is why the method of direct evaluation is sometimes also called the point method.

Various methods of mathematical statistics are used to analyze the results. These methods can be combined and vary in depending on the type of task and the desired result. For the formation of a generalized evaluation of the group of experts are used average values most often. Sometimes it is necessary to determine how important a particular factor is in terms of some criterion. In this case, at the beginning you need to determine the weight of each factor, and then use the weighted average.

The next step is evaluation processing, which purpose is to obtain a generalized opinion based on the multiple judgments of experts. Processing of expert assessments includes the unification of results, the analysis of the consistency of opinion and the synthesis of generalized opinion. The unified results of expert estimation are the vector of ranks, the vector of relative significance, the matrix of pairwise comparisons, the vector of identifiers and the vector of numerical estimates. Depending on the chosen treatment methods, the results can be converted from strong scales to weaker ones. The main properties of the estimates, reflecting the consistency of expert opinions, are the relative frequency of contradictions, the variation scope and average deviations. To measure these properties, 12 indicators are known that allow to evaluate the pair and multiple consistency of opinions regarding one or more objects of examination. Each indicator of consistency is focused on its own type of assessments, however, cases of analysing consistency with the help of indicators of weaker scales are quite frequent. Synthesis of generalized opinion is carried out in two ways: statistical (arithmetic mean, weighted average, sum of ranks, majority voting) and algebraic (distribution median, Kemeny median, Condorcet principle). The choice of the generalization method is determined by the type of generalized estimates.

In the case of participation of several experts in the survey, differences in their evaluation are inevitable, but the value of this difference is important. Group evaluation can be considered sufficiently reliable only if there is a good consistency in the responses of individual specialists. For the analysis of variance and consistency of estimates, statistical characteristics are used - measures of variation. Most often, as a measure of the variation are used variation range, average linear deviation, standard deviation and variance. These values are calculated based on the results of expert evaluation using the formulas:

variation range $-R = x_{\max} - x_{\min}$;

average linear deviation $-\ a = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \bar{x}\right|$;

standard deviation $-\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}$.

In these formulas: *n* – the number of evaluations, $\bar{x}$ – the average of all evaluations.

2.1 Kendell coefficient of concordance

The Kendell coefficient of concordance can take values ranging from 0 to 1. With full consistency of expert opinions, the coefficient of concordance is equal to one, with complete disagreement - zero. The most realistic is the case of partial consistency of experts.

The average rank of the set of features:

$$\bar{S} = \frac{\sum_{j=1}^{n} S_j}{n}.$$

The deviation dj of the average rank of the j-th attribute from the average rank of the aggregate is included: $d_j = \bar{S} - S_j$

The number of identical ranks assigned by experts to the j-th attribute - tq is determined. The number of groups of the same rank is determined - Q. The coefficient of concordance is determined by the formula:

$$k = \frac{12\sum_{j=1} d_j^2}{m^2(n^3 - n) - m\sum_{i=1}^{m} T_i}$$

where $T_i = \sum_{q=1}^{Q} (t_q{}^3 - t_q)$.

2.2 Method appliance. Generalization of numerical estimates

As already noted, methods based on the calculation of the average (arithmetic or weighted) or on the frequency analysis of the distribution are used to summarize such estimates.

The arithmetic average is the most common way because of its simplicity. The element of the resulting vector is calculated as the arithmetic average of the corresponding estimates:

$$\lambda_i^* = \frac{1}{m}\sum_{j=1}^{n} \lambda_{ij}$$

where λ_{ij} - the relative importance of the i-th object according to the j-th expert.

2.3 The expert assessment method for IoT risk analysis

For the quantitative risk assessment with the help of this scale a method of expert evaluation is applied. The results of risk assessment in IoT subsystems are presented in table 1.

The table contains nine lines and nine columns, in which arithmetic averages the risk estimates given by all experts are registered. In cells with identical line numbers and a column (located on the main diagonal) the risk degrees resulting from influence of any negative factor arising in this subsystem, one of the main executions of functions by this exact subsystem are presented., In cells with the line number of i and column j the risk degrees resulting from impact on a subsystem with number i of the negative factor arising in a subsystem with number j (cross risks) are specified.

As a calculated example, let us take the cell, where the degree of risk arising from the effect on the subsystem smart device from a negative factor, which emerges in the subsystem the IoT system, and the numerical estimates data of only 5 expertsin order to count the average following the formula above:

$$\lambda_9^* = \frac{1}{5}\times(11+8+2+5+1)=5,4. \tag{1}$$

Table 1 Results of Risk Assessment for IoT Smart Systems

		1	2	3	4	5	6	7	8	9	
		The system of IoT	Smart state	Smart area	Smart city	Smart district	Smart house	Smart apartment	Smart room	Smart device	**Average**
1	The IoT system	10,9	7,1	6,0	5,2	3,9	3,0	2,5	2,0	1,3	4,7
2	Smart state	8,9	10,9	7,2	5,8	4,6	2,9	2,3	1,6	1,1	5,0
3	Smart area	8,3	8,7	10,9	7,2	5,8	3,7	2,8	2,1	1,5	5,7
4	Smart city	7,8	7,8	8,4	10,8	7,3	4,2	3,3	2,6	1,8	6,0
5	Smart district	7,0	7,0	7,6	8,4	10,8	6,2	4,4	3,4	2,2	6,3
6	Smart house	6,2	6,4	6,6	7,1	7,4	10,8	7,0	5,1	3,8	6,7
7	Smart apartment	6,0	6,2	6,3	6,7	7,0	8,2	10,8	7,6	5,0	**7,1**
8	Smart room	5,6	5,7	5,9	6,4	6,5	7,6	8,5	10,8	6,0	7,0
9	Smart device	5,4	5,3	5,7	5,7	6,0	6,6	7,1	7,4	10,7	6,6
	Average	**7,3**	7,2	7,2	7,0	6,6	5,9	5,4	4,7	3,7	

An example of calculation by formula (1) is shown in table 2.

The arithmetic mean vector has two main disadvantages. Firstly, it equates all experts, regardless of their competence and other qualities.

And secondly, it accumulates and summarizes all the errors and errors contained in the generalized estimates.

The weighted arithmetic mean as an alternative, if not completely protects against these troubles, then at least provides a means to manage them.

Table 2 Example of calculation

Expert №	Estimations
1	11,0
2	8,0
3	2,0
4	5,0
5	1,0
Average	5,4

This tool are weights assigned to all experts in each specific expertise. The element of the resulting vector is calculated as the weighted average of the corresponding estimates:

$$\lambda_i^* = \sum_{j=1}^{n} v_j \lambda_{ij}$$

where v_j is the normalized weight of the j-th expert ($\sum_{j=1}^{m} v_j = 1$).

The expert's weight can be determined based on his competence or on the analysis of deviations of his assessments, so it can be any linear combination of the coefficients of competence, awareness, degree of reliability, inconsistency and other quantitative indicators of its qualities. However, as weight coputaton process is a pretty complex task, for this survey let us set this value for each expert conditionally, what is presented in Table 3. Based on deviation from the average, for each expert weight`s values were determined in the third cjkumn of this table.

Calculations:

$$\lambda_9^* = (11\times 0{,}1 + 8\times 0{,}2 + 2\times 0{,}2 + 5\times 0{,}4 + 1\times 0{,}1) = 5{,}2\,.$$

The error coputation of 2 methods:

$$\Delta\lambda_9^* = \frac{5{,}4 - 5{,}2}{5{,}4}\times 100 = 3{,}7\%\,.$$

Table 3 Weights by the experts estimation

Expert №	Estimations	Normalized weight
1	11,0	0,1
2	8,0	0,2
3	2,0	0,2
4	5,0	0,4
5	1,0	0,1
Average	5,4	5,2

According to the results and methods of calculation, there are discrepancies and their reasons are clear. However, various approaches to the results generalzation can not be ignored because of the need for more reliable data.

3 Training preparation

Training preparation includes the following steps.

3.1 Selection of the research object for risk assessment from the proposed list (can be expanded by agreement with the teacher):

1) Risk Assessment for Smart State System Components;
2) Risk Assessment for Smart Region System Components;
3) Risk Assessment for Smart City System Components;
4) Risk assessment for the components of the Smart Region of City;
5) Risk Assessment for Smart Home System Components.

3.2 Search of the information about research object (please, use the libraries**,** the Internet, technical reports, other sources) and primary analysis. Please use reference list [1-16].

3.3 Decomposition of the research object to subsystems according to selected sign (functionality, role in the object, territorial location abd other)

3.4 The study of the methodology for conducting research by expert assessments. Please, use the material of section 2 and additional references are described in Part X (sections 36) of the book [17].

3.5 Selection of experts and assessment of their competence. Experts should be selected from among those familiar with the subject area. It is recommended that at least five independent experts be selected.

3.6 Preparation of materials for the work of experts. You should prepare a table for dividing the selected Internet of Things system into subsystems, a risk assessment scale, and instructions for experts.

3.7 Processing the results of peer review. Please use section 2 materials as well as additional materials [18].

3.8 Presentation plans development. Report plan includes:

– introduction (relevance, reality challenges, brief analysis of the problem references, purpose and tasks of the report, structure and contents characteristics);

– systematized description of the main report parts (classification schemes, models characteristics, methods, tools, technologies by groups, selection of indexes and criteria for assessment, comparative studies);

– conclusions (established goal achievement, main theoretical and practical results, result validity, ways of further work on the problem);

– list of references;

– appendixes.

3.9 Presentation preparation. The presentation is to be designed in PowerPoint and be consistent with the report plan (10-15 slides); the time-frame for the presentation is 15 minutes.

The presentation should include the slides as follows:

- title slide (specification of the educational institution, department, course of study, report subject, authors, date);
- contents (structure) of the report;

- relevance of the issues covered, the purpose and the tasks of the report based on the relevance analysis;
- slides with the details of the tasks;
- report conclusion;
- list of references;
- testing questions.

Each slide should include headers with the report subject and authors.

The contents of the slides should include the keywords, figures, formulas rather than the parts from the report.

The information can be presented dynamically.

3.10 Presentation and defence

The presentation should be given at the seminar during 20 minutes including: presentation (10-15 minutes) and discussion (5-10 minutes).

Time schedule can be specified by lecturer.

4. Individual task for assessment

1 Study decomposition ways of the Internet of things system into subsystems;
2 Study the methods and scale of risk assessment;
3 Study the method of expert assessments and results` processes;
4 Carry out an expert assessment of risks in the IoT, according to the method above. For this purpose:
5 Determine the structure of the expert group;
6 Prepare the necessary materials for each expert: a hierarchical list of IOT subsystems; a risk assessment scale; a table for making risk assessments by experts;
7 Organize independent work of the experts;
8 Execute processing of expert estimates;
9 Perform the received results in the presentation form;
10 Make judgments by the received results.

5 Recommended literature

1. D. Maevsky, A. Bojko, E. Maevskaya, O. Vinakov and L. Shapa, "Internet of Things: Hierarhy of smart systems," 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, 2017, pp. 821-827.

2. E. Nakashima and J. Warrick, “Stuxnet was work of U.S. and Israeli experts, officials say,” The Washington Post, June 2, 2012

3. O. Johny, S. Sotiriadis, E. Asimakopoulou and N. Bessis, "Towards a Social Graph Approach for Modeling Risks in Big Data and Internet of Things (IoT),” in 2014 International Conference on Intelligent Networking and Collaborative Systems, Salerno, 2014, pp. 439-444.

4. R. M. Savola, P. Savolainen, A. Evesti, H. Abie and M. Sihvonen, "Risk-driven security metrics development for an e-health IoT application," in 2015 Information Security for South Africa (ISSA), Johannesburg, 2015, pp. 1-6

5. M. St John-Green, R. Piggin, J. A. McDermid and R. Oates, "Combined security and safety risk assessment — What needs to be done for ICS and the IoT," in 10th IET System Safety and Cyber-Security Conference 2015, Bristol, 2015, pp. 1-7.

6. W. Xi and L. Ling, "Research on IoT Privacy Security Risks," in 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, 2016, pp. 259-262.

7. D. Y. Kim, K. Y. Kim, G. K. Park and J. S. Jeong, "A Study on the Implementation of Intelligent Navigational Risk Assessment System with IoT Sensor," in 2016 Joint 8th International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems (ISIS), Sapporo, 2016, pp. 328-333.

8. P. Valerio, "Is the Iot a tech bubble for cities? With more cities joining the smart city revolution and investing in sensors and other iot devices, the risk of a new tech bubble is rising," in IEEE Consumer Electronics Magazine, vol. 5, no. 1, pp. 61-62

9. D. Howard, “Welcome to the post-dotcom era,” netWorker 5, 2 (June 2001), 26-31.

10. D. Nunes et al., "FoTSeC — Human Security in Fog of Things," 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 2016, pp. 743-749.

11. J. Bugeja, A. Jacobsson and P. Davidsson, "On Privacy and Security Challenges in Smart Connected Homes," 2016 European Intelligence and Security Informatics Conference (EISIC), Uppsala, Sweden, 2016, pp. 172-175.

12. A. Shifa, M. N. Asghar and M. Fleury, "Multimedia security perspectives in IoT," 2016 Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, 2016, pp. 550-555.

13. M. Irshad, "A Systematic Review of Information Security Frameworks in the Internet of Things (IoT)," 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1270-1275.

14. R. L. Rutledge, A. K. Massey and A. I. Anton, "Privacy Impacts of IoT Devices: A SmartTV Case Study," 2016 IEEE 24th International Requirements Engineering Conference Workshops (REW), Beijing, 2016, pp. 261-270

15. M. Nawir, A. Amir, N. Yaakob and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," 2016 3rd International Conference on Electronic Design (ICED), Phuket, 2016, pp. 321-326.

16. Anokhin, Alexey. (1996). Методы экспертных оценок. 10.13140/2.1.5043.3282.

17. Internet of Things for Industry and Human Application. In Volumes 1-3. Volume 3. Assessment and Implementation / V. S. Kharchenko (ed.) - Ministry of Education and Science of Ukraine, National Aerospace University KhAI, 2019. – 740 p.

18. "Evaluation", En.wikipedia.org, 2019. [Online]. Available: https://en.wikipedia.org/wiki/Evaluation. [Accessed: 20-Aug- 2019].

2 ENGINEERING OF SOFTWARE/HARDWARE PLATFORM FOR SMART BUILDING SYSTEM

Introduction

In essence, a Smart Building System (SBS) is a complex arrangement of technologies that allow home appliances to intercommunicate and perform certain daily tasks without human intervention. Ideally, such systems allow for control of household elements through a mobile application, alongside the creation of certain automated behavioral patterns and the possibility of complete automation, including independent solutions from elements of the house as a result of data analysis.

Smart home automation technologies are partially connected with one additional current trend – the Internet of Things (IoT). Connecting simple appliances to the network for remote user control is a very popular sphere of development today.

SBS comprise sensors, monitors, interfaces, appliances and devices networked together to enable automation as well as local and remote control of the domestic environment. Controllable appliances and devices include heating and hot water systems (boilers, radiators), lighting, windows, curtains, garage doors, fridges, TVs, washing machines, etc. Sensors and monitors detect environmental factors including temperature, light, motion, and humidity. Control functionality is provided by software on computing devices (smartphones, tablets, laptops, PCs) or through dedicated hardware interfaces (e.g., wall-mounted controls). These different compoments are networked, usually wirelessly, using standart communication protocols. The diversity of available components means the SBS has many possible configurations and by implication, “smartness”.

A growing popularity of smart home technology among users attracts more and more people to the new development. Now, there is a large variety of devices that are associated with a smart home. However, not all devices and systems are certified and compatible, so today the issue of their usage and reliable connection are still exist.

This manual for training contains 4 laboratory works that are aimed to software/ hardware development for SBS.

Laboratory work1
Software/ hardware development based on Arduino platform. Development of interactive graphical interface for interaction with Arduino platform

Objective: to learn the tools for developing the graphical interface for the interaction with Arduino platform.

1 Brief theoretical information

The Arduino Integrated Development Environment (IDE) is a cross-platform application. It is written in Java programming language. It is arising from IDE for Processing and Wiring programming languages. The same as Wiring, Arduino was created to simplify the programming of microcontrollers. By the help of Arduino users who do not have the skills to work with electronics could easily develop their own project based on the microcontroller and other sensors.

On the other hand, Processing IDE was created to give opportunity for students of technical specialization more simple way to deal with graphics.

Arduino and Processing IDE are close related (the first is descendant of second), so they could interact with each other. This fact gives opportunity to create simple graphical interface for convenient management of Arduino platform or create interface which will display data obtained from Arduino on computer screen.

Data can be transmitted between the processing sketch (source code) and the Arduino platform using a serial port. Serial Library is used to work with serial ports in Arduino and Processing. This library contains functions which facilitate the consistent data exchange between Arduino and Processing. Serial is a data type that sends one byte per action. For Arduino, bytes are a data type that can store values from 0 to 255. It works as an integer data type, but with a much shorter range. Large numbers are sent with a splitting into the list of bytes and their subsequent reassembly.

In the examples below, we implement simple programs for demonstrating bidirectional communication between Arduino and Processing.

After opening the file processing.exe, you will see the window shown on Fig. 1. As you can see, it is very similar to the Arduino IDE. This is the window sketch. Sketch is the source code of the program. This window consists of several elements such as: Menu, Toolbar, Tabs, Text editor, Message box, and Console.

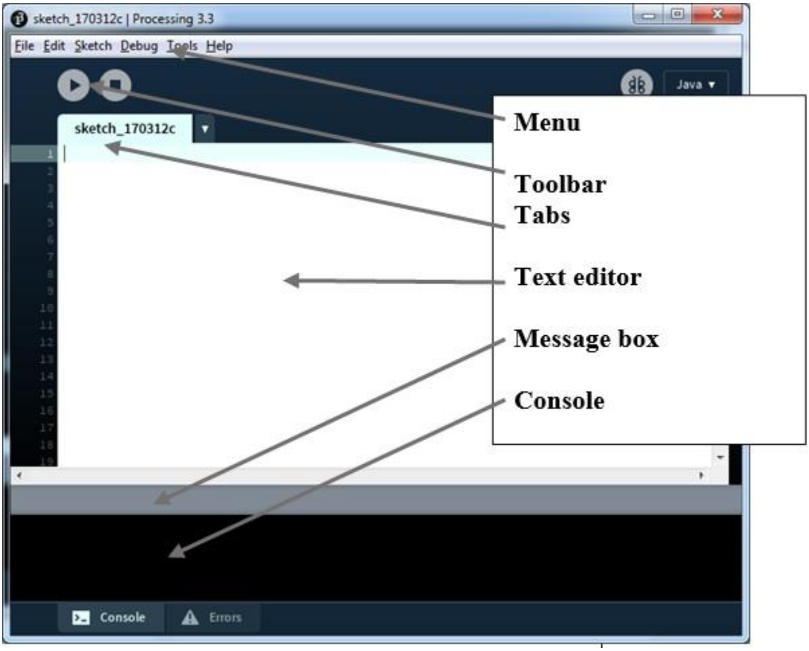


Fig. 1 – Processing Interface

The Processing programming language is a Java dialect, but with some features for working with graphics and interactive content. Processing has taken a lot from PostScript (which served as the basis for the PDF format) and OpenGL (3D graphics specification).

In order to start working with Processing, you must have basic knowledge of the RGB color model and understanding of pixels (picture elements). Also, you have to have some experience in working with Arduino.

To build the first program, you need to learn the basic functions and default variables of Processing programming language (see [4,6] for more details on syntax of Processing language and principles).

Basic functions are:

void setup() and void draw() – the main functions of the program, all other functions are situated inside these two cycles; void setup() is executed once at the beginning of the program; void draw() is a function of an infinite loop, it is executed continuously;

size (X, Y) – creates a workspace (or window) of size X on Y pixels, by default it is 100 by 100 pixels;

background(a) – sets the background color; background(a) is equivalent to recording a background(a, a, a), where the three variables correspond to the main colors of the RGB color model: red, green, and blue;

fill(R, G, B) – indicates the fill color of the figure. You can add the fourth parameter to this function T. It is the transparency, by default, if not specify this parameter, it is 255 (100% - is not transparent);

line(x1, y1, x2, y2) – draws a line from the point (x1, y1) to the point (x2, y2);

rect(x, y, w, h) – draws a rectangle, starting with the left upper point (x, y), with the parameters w - length, h - height, you can also add the fifth parameter t - the radius value for all four corners.;

mousePressed() – is called once after every time a mouse button is pressed*;*

Serial.begin(speed of transmission)- sets the speed of the COM port information as "bit per second" for serial data transmission. In order to maintain communication with the computer, one of the normalized speeds is used: 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, or 115200 (for example, Serial.begin (4800);). By default, the Arduino serial port monitor has a speed of 9600 bps;

Serial.available() – the bytes which is coming from the serial port fall into the microcontroller buffer, where the program can read them. The function returns the number of bytes stored in the buffer. The sequential buffer can store up to 128 bytes.

Returns the value of uint8_t (typedef uint8_t byte;) type – the number of bytes that are in read-only access in the sequential buffer, or 0, if nothing is available;

Serial.read() – reads the next byte from the buffer of the serial port. Returns the first available byte of the input data from the serial port, or -1 if there is nothing to read.

Default variables are:

mouseX/mouseY – contains current horizontal/vertical coordinate of the mouse;

height, width – correspond to the height and width value of the working screen, which are specified by the function size().

2 Execution order and discovery questions

Let’s consider several examples.

Example 1.Turn on / off the embedded LED.

Open the Arduino IDE environment. Copy the code from the example below, compile and download the program to the Arduino board:

```
     char val; // data received from the serial port
     int ledPin = 13; // Initialization of 13 built-
in board led
      void setup() {
      pinMode(ledPin, OUTPUT); // set output mode for
pin
      Serial.begin(9600);  //  start  exchanging  data
with speed 9600 bps.
      }

      void loop() {
      while (Serial.available()) { // if the port is
available
      val = Serial.read(); // read data and write it
in val
      }
      if (val == 'H') { // if H was received
      digitalWrite(ledPin, HIGH); // LED lights up
      } else {
      digitalWrite(ledPin,  LOW);  //  otherwise  it  is
turned off
      }
      }
```

After that, open the Processing environment, copy the program shown in the example below and save the program to the hard drive. (**File -> Save**):

```
     import processing.serial.*;
     Serial arduino;  // create a Serial object

     void setup()
     {
       size(390, 200);
       printArray(Serial.list());
       arduino = new Serial(this, Serial.list()[0],
9600); // We start the serial connection at a speed
of 9600
       delay(1000);
     }

     void draw() {
       background(255);
       // Turn ON/OFF LED
       textAlign(CENTER);
       textSize(32);
       fill(153, 100);
       text("On/Off LED", 197, 42);
       fill(0, 102, 153);
       text("On/Off LED", 195, 40); // create a
shadown for the text
       stroke(102);

         if(mouseOverRect() == true && mousePressed
== true) { //check if mouse over button
           fill(0, 117, 13, 240);
           rect(147, 77, 96, 46, 7);          // Draw
the button
           arduino.write('H');                // send
an H to indicate mouse is over square
         }
         else {
           fill(128, 200);
           rect(147, 77, 100, 50, 7);         // Draw
the button
           fill(0, 200, 51);                       //
change color and
           rect(145, 75, 100, 50, 7);         // Draw
new smalest button to create push effect
           arduino.write('L');                // send
 an L otherwise
         }
     }

     boolean mouseOverRect() { // Test if mouse is
 over button
       return ((mouseX >= 145) && (mouseX <= 245) &&
 (mouseY >= 75) && (mouseY <= 125));
     }
```

Click the **Run button** on the toolbar, which will launch the compilation of the program.

As a result, you will see a window in which the rectangular button is displayed (see Fig. 2.). After clicking on this button, the color will change and the built-in LED 13 will light up.

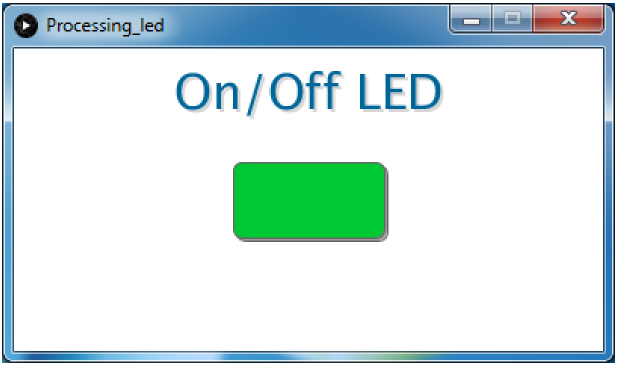


Fig. 2 – Turn on / off the embedded LED

Example 2. Regulation of LED brightness.

The following example allows you to regulate the brightness of LED by changing the position of the slider in Processing.

Connect LED and resistor to the Arduino board and breadboard as shown on Fig. 3.

Open the Arduino IDE environment. Copy the code from the example below, compile and download the program to the Arduino board:

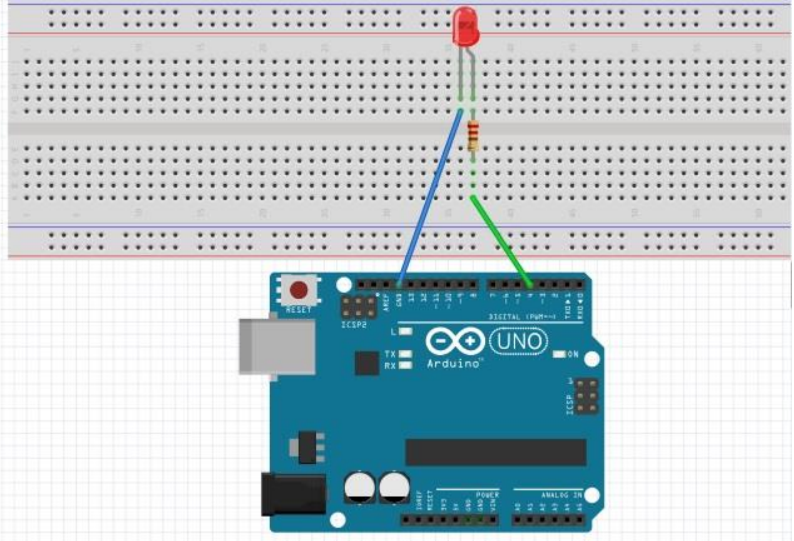


Fig. 3 – Scheme of connection LED and resistor to Arduino

```
      int val; // Data received from the serial port
      int ledPin = 4; // Set the pin to digital I/O 4

      void setup() {
      pinMode(ledPin, OUTPUT); // Set 4 pin as OUTPUT
      Serial.begin(9600);    //    Start    serial
communication at 9600 bps
      }

      void loop() {
      while  (Serial.available())  {  //  If  data  is
available to read,
      val = Serial.read(); // read it and store it in
val
      }

      if (val>=0 && val<=255){
      analogWrite(ledPin,  val);  //  Set  the  LED4
brightness
      }
     Serial.println(val);
      }
```

After that, open the Processing environment, copy the program shown in the example below and save the program to the hard drive. **(File -> Save)**:

```
     import processing.serial.*; // import serial
library
     Serial arduino;  // create object

     int x = 10;

     void setup(){
     size(300, 200);
     printArray(Serial.list());
     arduino = new Serial(this, Serial.list()[0],
9600);
     delay(1000);
     }

     void draw(){
       background(255);

     // LED Brightness
       textAlign(CENTER);
       textSize(32);
       fill(153, 100);
       text("LED Brightness", 147, 42);
       fill(0, 102, 153);
       text("LED Brightness", 145, 40); // create a
shadown for the text

     //draw control slider
       fill(240);
       triangle(20, 115, 270, 95, 270, 115);
       line(20, 115, 270, 115);
       for(int i = 20; i<271; i=i+125){
         stroke(78);
         line(i,95,i,115);
       }
        fill(0,0,255);
        ellipse(x, 110, 20, 30);
        stroke(156);
        noFill();
        ellipse(x, 110, 18, 28);
       if(mouseOver() == true && mousePressed){ //
check if we change slider position
          x=mouseX;
       }
       else if(x>270) {
       x=270;
       }else if(x<20){
         x=20;
       }
     int pos = int (map(x, 20, 270, 0, 255)); //
cover mouse position to 0 from 255
     arduino.write(pos);
     }

     boolean mouseOver(){    // Test if mouse is over
slider
       return((mouseX >= 20) && (mouseX <= 270) &&
(mouseY >= 75) && (mouseY <= 130));
     }
```

Click the **Run button** on the toolbar, which will launch the compilation of the program.

As a result of running this program, a window will appear in which the slider is shown (see Fig. 4). Click on the mouse button and drag the slider to the right. After that, the brightness of the LED will increase. If you drag the slider to the left, then the brightness of the LED will decrease.

Example 3.Working with servo and potentiometer

In the following example, we will consider the interaction of Processing with a servo and potentiometer.

Connect servo and potentiometer to the Arduino board and breadboard as shown on Fig. 5.

Open the Arduino IDE environment. Copy the code from the example below or open a file **Arduino.ino**, compile and download the program to the Arduino board:

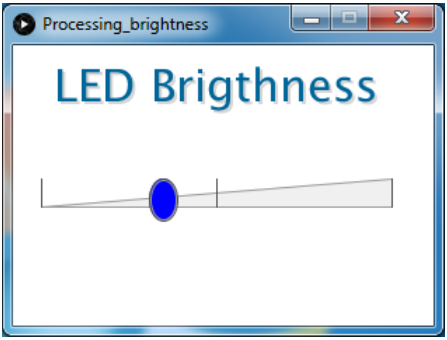


Fig. 4 – Regulation of LED brightness

```
     #include <Servo.h>
     int val=0; // variable where we will record the
value of brightness depending on the position of the
handle of the potentiometer
     int vall; // variable in which we will write the
value of the corner of the turn for Servo
     Servo myServo; // variable for working with
Servo

     void setup() {
       myServo.attach(5); // connect servo to port 5
       Serial.begin(57600); // We start the serial
connection at a speed of 57600
     }

     void loop() {
       while (Serial.available()) { // if data is
available for reading,
         vall = Serial.read(); // read and save them
in vall variable
       }
       myServo.write(vall); // send the value of vall
to servo, it starts rotate
       val=analogRead(A0)/4; // read the value from
the analog input and write it in the variable val
       Serial.write(val); // send the recorded values
to the serial port
       delay(500); // waiting 500 milliseconds before
the next reading
     }
```

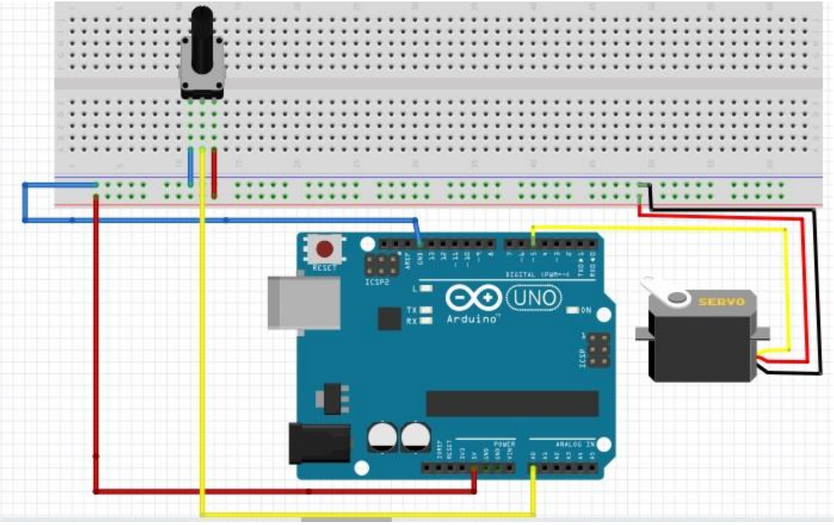


Fig. 5 – Scheme of connection servo and potentiometer to Arduino

After that, open the Processing environment, copy the program shown in the example below and save the program to the hard drive (**File -> Save**):

```
     import processing.serial.*; //connecting serial
library
     Serial myPort;  // creating a Serial object
     int val;        // variable for data sent/read
to/from Arduino

     color rectHighlight;
     color rectColor;
     // Creating variables to check the position of
the cursor
     boolean rectOver1 = false;
     boolean rectOver2 = false;
     boolean rectOver3 = false;

     void setup()
     {
       size(650, 250);
         rectColor = color(255, 0, 0);
         rectHighlight = color(255,100,100);
         printArray(Serial.list());
         myPort = new Serial(this, Serial.list()[0], 
57600);  //creating a serial port named portName and 
a serial communication speed of 57600
         delay(1000);
     }

     void draw()
     {
       update(mouseX, mouseY);
       background(255);
       smooth();
       stroke(0);
       fill(255);
       arc(150, 162, 149, 124, -PI, 0); // draw an 
arc
       // draw lines
       line(75,162,80,162);
       line(225,162, 220,162);
       line(150, 100, 150, 105);
     // Change color when mouse cursor over 0 degree 
rectangle
       if (rectOver1) {
         fill(rectHighlight);
       } else {
         fill(rectColor);
       }
       //create red rectangle
       rect(25, 150, 50, 25); // draw 0 degree 
rectangle

     // Change color when mouse cursor over 180 
degree rectangle
       if (rectOver2) {
         fill(rectHighlight);
       } else {
         fill(rectColor);
       }
       rect(225, 150, 50, 25); // draw 180 degree 
rectangle

     // Change color when mouse cursor over 90 degree
```

```
rectangle
       if (rectOver3) {
         fill(rectHighlight);
       } else {
         fill(rectColor);
       }
       rect(125,  75,  50,  25);  //  draw  90  degree
rectangle

       // Create text at the appropriate coordinates
       textAlign(CENTER);
       fill(0);
       textSize(12);
       text("0 deg", 50, 167);
       text("180 deg", 250, 167);
       text("90 deg", 150, 92);
       textSize(16);
       text("From Processing To Arduino", 150, 20);
       text("From Arduino To Processing ", 480, 20);
       text("Experiment with Servo", 150, 220);
       text("Experiment  with  Potentiometer",  480,
220);
       if ( myPort.available() > 0) {  // if the port
is open
         val = myPort.read();           // read data
from it and save it in val
       }
       textSize(12);
       text("Value", 365, 115); // Value text
       stroke(0);  // color of the line
       line (350, 120, 605, 120);  // markup line
       for  (int  i  =  350;  i  <=  605;  i  =  i+17)  {  //
short lines over 17 pixel
         stroke(0); // short lines color
         line(i, 120, i, 123); // draw short lines
       }
       fill(0); // color of the slider
       noStroke();  // rectangle without outline
       rect(350,  125,  val,  10);  //  draw  a  rectangle
(slider)  that  changes  its  width,  depending  on  the
value obtained in val
       delay(100);
     }
     // Check whether the cursor is above one of the
rectangles
     void update(int x, int y) {
       //0 degree
       if ( overRect(25, 150, 50, 25) ) {
         rectOver1 = true;
       } else {
         rectOver1 = false;
       }
       //180 degree
       if ( overRect(225, 150, 50, 25) ) {
         rectOver2 = true;
       } else {
         rectOver2 = false;
       }

       //90 degree
       if ( overRect(125, 75, 50, 25) ) {
         rectOver3 = true;
       } else {
         rectOver3 = false;
       }
     }

     void mousePressed() {
       if (rectOver1) {               // If the mouse
button is clicked on the 0 degree rectangle,
         myPort.write(0);         // send  a value of
0 to the serial port
       }
         if (rectOver2) {             // If the mouse
button is clicked on a 180 degree rectangle,
         myPort.write(180);        // send a value of
180 to the serial port
       }
        if (rectOver3) {              // If the mouse
button is clicked on the 90 degree rectangle,
         myPort.write(90);          // send the value
of 90 to the serial port
       }
     }
     boolean overRect(int x, int y, int width, int
height) {
       if (mouseX >= x && mouseX <= x+width &&
         mouseY >= y && mouseY <= y+height) {
         return true;
       } else {
         return false;
       }
     }
```

Click the **Run button** on the toolbar, which will launch the compilation of the program.

As a result, you will see a window which represents the Processing working process. On the left there are 3 rectangular buttons for servo operation. On the right there is scale with the value of the potentiometer (Fig. 6).

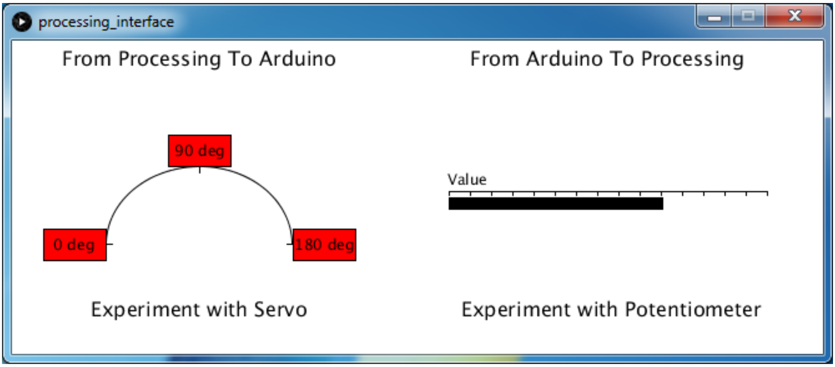
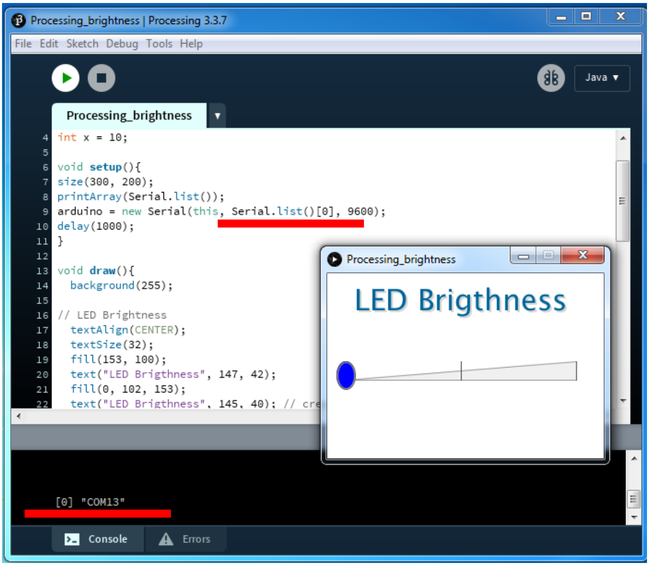

processing_interface
From Processing To Arduino
From Arduino To Processing
90 deg
0 deg
180 deg
Value
Experiment with Servo
Experiment with Potentiometer

Fig. 6 – The result of the work in the Processing

After clicking on the button, Processing sends the value of the turning angle to Arduino. Servo connected to Arduino rotates to the angle value received from Processing. The Experiment with Potentiometer displays data transfer from Arduino to Processing. When we rotating the potentiometer handle connected to Arduino, the length of the slider will change reflecting the new obtained values.

Error handling: if you have more than one device connected to a serial port on your PC, it is possible that during compilation you might see an error according to the serial port connection. In this case, try to change the value at String portName = Serial.list () [0]; from [0] to [1], or [2], it depends on how many devices are connected to the PC. The list of all available serial ports you can see after **Run** the program in the **Console** (see Fig. 7).



```
int x = 10;

void setup(){
size(300, 200);
printArray(Serial.list());
arduino = new Serial(this, Serial.list()[0], 9600);
delay(1000);
}

void draw(){
  background(255);

// LED Brightness
  textAlign(CENTER);
  textSize(32);
  fill(153, 100);
  text("LED Brigthness", 147, 42);
  fill(0, 102, 153);
  text("LED Brigthness", 145, 40); // cre
```

```
[0] "COM13"
```

Fig. 7 – List all the available serial ports in the Console

3 Assignments to laboratory work

1. Read the theoretical information.
2. Connect RGB LED to the Arduino. Create an Arduino program for receiving information from Processing to change the RGB LED color.
3. Create a program for Processing where the interface look like three rectangles or spheres (red, green and blue). When the mouse is pressed on the red rectangular the RGB LED produces the red color, etc.
4. Make a report for laboratory work.

4 Test questions

1. Describe the difference between execution of void setup() and void draw() functions?
2. Which function is intended to set up the background color?

3. Which data is stored in *mouseX*, *mouseY* variables?
4. For what purpose Serial library is used?
5. What are the main functions of the Serial library?
6. What is the default speed of the Arduino serial port monitor?

5 Requirements to the content of the report

1. The objective of the lab.
2. Brief theoretical information.
3. Program code.
4. Screenshots showing the completed tasks.
5. Conclusions to the work.

6 Recommended literature

1. Coburn J. “Getting Started with Arduino: A Beginner’s Guide”. [Electronic resource]. – Access mode: https://www.makeuseof.com/tag/getting-started-with-arduino-a-beginners-guide/

2. Ardino IDE. [Electronic resource]. – Access mode: https://www.arduino.cc/en/main/software

3. Arduino Lesson 3. RGB LEDs. [Electronic resource]. – Access mode: https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds/overview

4. Processing. [Electronic resource]. – Access mode: https://processing.org/

5. Wiring. [Electronic resource]. – Access mode: http://wiring.org.co/

6. Reas C., and Fry B. “Getting Started with Processing”/ Published by O’Reilly Media, Inc., June 2010. P.209

Laboratory work 2
Software/hardware development based on Arduino platform. Working with sensors and actuators

Objective: to learn the basic principles of sensors and how to connect sensors to Arduino.

1 Brief theoretical information

The important components of SBS are different sensors.

An analogue liquid level sensor can be used a water sensor (Fig. 1). The operating voltage of the analog sensor is 5V. Output voltage (sensor reading) depends on the degree of immersion of the sensor in the liquid and on the parameters that affect the voltage transmission coefficient, for example, the conductivity of the liquid.

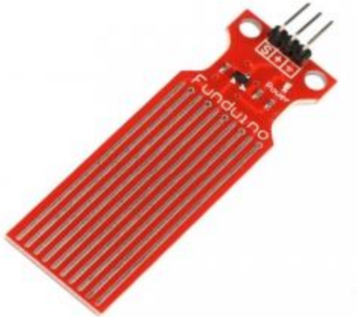


Fig. 1 – An analog liquid level sensor

As a light sensor can be used a light sensor module (Fig. 2) with a threshold comparator. The threshold of the operation of the comparator is regulated by a variable resistor.

Main technical characteristics:

- sensitive element - photoresistor;
- the output of the comparator is more than 15 mA;
- regulation of the trigger threshold by a variable resistor;
- working voltage: from 3.3V to 5V;
- digital output of the comparator (0 and 1).

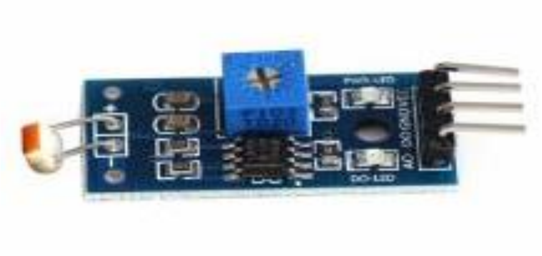


Fig. 2 – Light sensor

PIR (passive infrared sensors) sensors allow motion capture (Fig. 3). Very commonly used in alarm systems. These sensors are small in size, inexpensive, consume little energy, easy to operate, almost not subject to wear.

Main technical characteristics:

- sensor area: up to 6 meters;
- operating voltage: 5 - 9V.

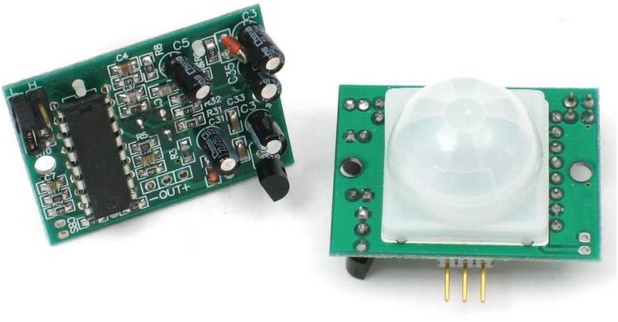


Fig. 3 – Motion sensor

The gas sensor, built on the basis of the gas analyzer MQ-2 (Fig. 4), can detect the presence of hydrocarbon gases in the ambient air (propane, methane, n-butane), smoke (suspended particles that are the result of combustion), hydrogen.

The sensor has 4 contacts (power, ground, digital output, analog output), but also the sensor can have 3 contacts (power, ground, analog output).

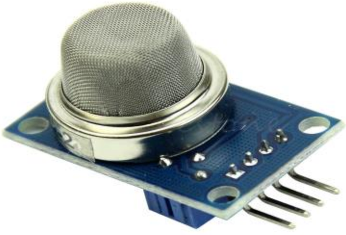


Fig. 4 – Gas sensor

The DHT sensor (Fig. 5) can be selected to measure temperature and humidity. The DHT11 sensor is a digital temperature and humidity sensor that allows you to calibrate the digital signal at the output. Consists of a capacitive humidity sensor and thermistor. Also, the sensor contains an ADC for converting analog values of humidity and temperature.

Main technical characteristics:

– working voltage: from 3.3V to 5V;

– humidity determination 20-95% with 5% accuracy;

– determination of temperature 0-50 degrees. with an accuracy of 2 degrees;

– the polling frequency is not more than 1 Hz (no more than once in 1 sec.).

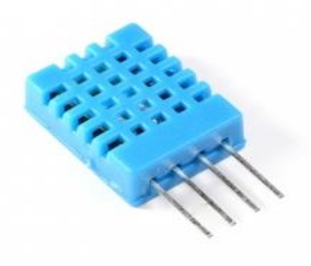


Fig. 5 – Temperature and humidity sensor DHT11

A sound sensor (Fig. 6) is made up of the board on which the outputs are mounted, the sound amplifier, the tuner resistor and the electronic microphone, sensitive to the sound coming in all directions. The regulator of sensitivity (variable resistor) can choose from which sound the sensor will operate.

The Arduino Expansion Card allows to translate sound vibrations into a digital signal. When the membrane oscillates in the microphone from the sound waves, the capacitance of its capacitor changes, which results in a change in the voltage at the outputs of the sound sensor, corresponding to the sound signal.

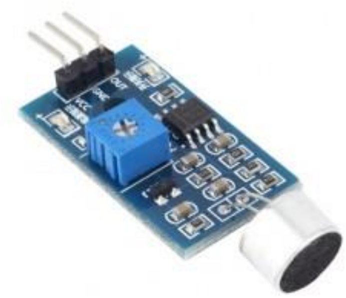


Fig. 6 – Sound sensor

The color tint sensor TCS230 (Fig. 7) for Arduino is capable of recognizing 4 colors and converting the intensity of the color spectrum into an output signal of different frequencies.

In the RGB color palette, any color can be represented as a combination of three main colors: red (R), green (G) and blue (B). Therefore, to determine the color, it is necessary to measure the red, the blue and the green spectrum. As a sensitive element in the sensor TCS230, consisting of an array of photodiodes 8 × 8 - 16 photodiodes in three colors and 16 photodiodes without a filter, is used.

Specifications:

- supply voltage from 2.7 to 5.5 V;
- programmable colors and output frequency of the signal;
- automatic power off function;
- low error of the output frequency - 0,2%.

The sensor is used to determine the color of the object's color at a distance of up to 10 mm, there are four LEDs to illuminate the measurement location on the sensor. On the reverse side of the sensor there are two pads with four contacts. Through these contacts, the sensor TCS230 connects to the Arduino microcontroller. Contacts "S0" and "S1" are used to scale the pulse frequency at the output "OUT" of the sensor color tint.

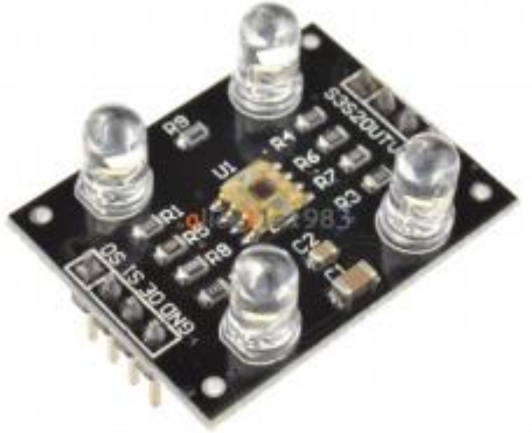


Fig. 7 – Sensor color tint

The ultrasound sensor measures the distance to the object in the same way as bats or dolphins do. The HC-SR04 sensor (Fig. 8) generates a narrow-angle signal at a frequency of 40 kHz and catches the reflected signal (echo). By the time the sound propagation to the object and back can be accurately determine the distance to it.

The HC-SR04 ultrasonic range finder measures from 2 cm to 400 cm, working at temperatures from 0 ° to 60 ° C. The measurement accuracy is ± 1 cm, the operating voltage of the sensor is up to 5.5 V.

The Arduino Vibration Sensor (Fig. 9) is used to determine the presence of external vibrational effects on the module and is used when creating a home alarm system.

This sensor can be issued as a module, for example, SW-420, logo sensors v1.5 or vibration sensor 140s001 for Arduino. The base of the SW-420 sensor is a metal spring inside a plastic tube, which begins to oscillate when vibrated. The module has a signal amplifier, a tuning resistor, for adjusting sensitivity of the sensor and three conclusions for connecting to the microcontroller.

Inside the sensor 801S is a metal ball on a spring that responds to the slightest fluctuations.

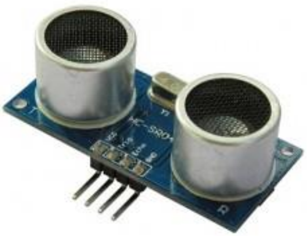


Fig. 8 – Ultrasound sensor

The principle of operation of these sensors can be compared to the clock button - if the contacts on the button are locked when we click on it, then the contacts in the sensor we handle are locked in the event of a vibration.

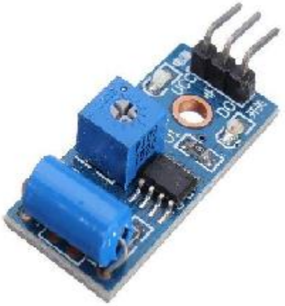


Fig. 9 – Vibration sensor

Soil moisture sensor (Fig. 10) with discrete and analog output. Consists of a sensitive element and a converter board. When drying the soil (the level of drying is set by adjusting the resistor on the board of the converter) to the discrete output logical unit is supplied. From the analogue output, you can get the exact value that characterizes the humidity of the soil, in the range from zero to the magnitude of the supply voltage.

Main technical characteristics:

– dimensions of the sensitive element 60 x 30 mm;

– dimensions of the converter 30 x 15 mm;

– supply voltage 3.3 ... 5 V;

– the length of the wire is 210 mm.

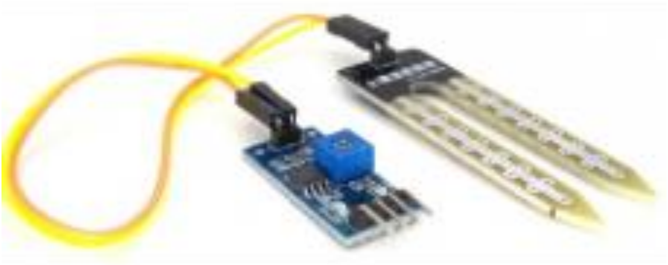


Fig. 10 – Soil humidity sensor

2 Execution order and discovery questions

In order to connect the sensors to Arduino you need to assemble them according to the scheme shown in Fig. 11.

Typically, the sensors have printed which contacts need to be connected:

– VCC - power supply, usually 5V.

– OUT (or something else) - analog output (port connection).

– GND - ground.

If there are 4 contacts, connect 3 (power, ground, analog output).

To work properly and receive temperature and humidity data from the DHT11 sensor, you need to connect the DHT.h and Adafruit_Sensor.h library. To do this, go to the links (https://github.com/adafruit/Adafruit_Sensor and https://github.com/adafruit/DHT-sensor-library), download the libraries and copy the unpacked files into the Arduino IDE's libraries directory.

For the correct operation of the program, the analog outputs of the sensors must be connected to the same ports:

- light sensor to 5;
- liquid level sensor to A1;
- gas sensor to A0;
- motion sensor to 1;
- liquid level sensor to 2.

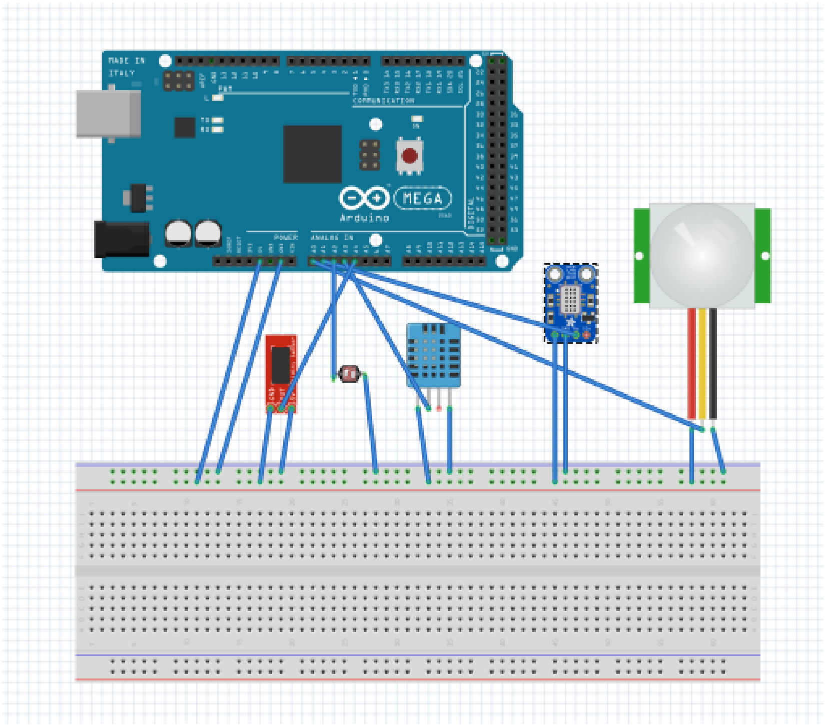


Fig. 11 – General scheme of connecting sensors

```
     #include <DHT.h>
     #include <DHT_U.h>
     #include <Adafruit_Sensor.h>
     #define DHTPIN 3 // stores the port number of
the temperature and humidity sensor

     DHT dht(DHTPIN, DHT11); // object for working
with sensor DHT11
     const int MQ2 = A0;//stores the port number of
the gas sensor
     const int ledPin = 13;//stores the port number
of the diode

     int sensorValue = 0; // stores the read values
of the gas sensors
     int x; // variable for liquid level sensor
     int inputPin = 2;// stores the port number of
```

```
the motion sensor
     int pirState = LOW;      // variable for motion
sensor
     int val = 0;             // variable for motion
sensor
     int buttonState = 0;  // variable for the light
sensor

     void setup() {
       Serial.begin(9600);
       pinMode(ledPin, OUTPUT); // Sets the mode for
13 pin
       dht.begin();
       pinMode(A1, INPUT);// water_level
       pinMode(inputPin, INPUT);// declare sensor as
input
       pinMode(buttonPin, INPUT);
      }

      void loop() {
     Serial.print(dht.readHumidity());
     Serial.print(",");
     Serial.print(dht.readTemperature());
     Serial.print(",");
     sensorValue = analogRead(MQ2);
     Serial.print(sensorValue);
     Serial.print(",");
     x = analogRead(A1);if (x > 100) {
     digitalWrite(13, HIGH); }
     if (x < 100) { digitalWrite(13, LOW); }
     Serial.print(x);
     Serial.print(",");
     val = digitalRead(inputPin);
       if (val == HIGH)
       {
         Serial.print("M_detected,");
          pirState = HIGH;}
       else {
        Serial.print("No_motion,");
         pirState = LOW;}
     buttonState = digitalRead(buttonPin);
     if (buttonState == HIGH)
     {digitalWrite(ledPin, HIGH);
     Serial.print("Light_off,");}
     else
     {digitalWrite(ledPin, LOW);
     Serial.print("Light_on");}
     Serial.println("");
     }
```

After starting the program, the initialization of variables and I/O ports begins. Then the program goes into the main cycle in which it carries out appropriate actions. Table 1 describes the features of the program.

Table 1. Description of program functions

Function name	Function description
Serial.begin(9600)	Opens the serial port, sets the speed to 9600 bps
pinMode(ledPin, OUTPUT)	Sets the mode of operation - output
dht.begin()	Run the humidity and temperature sensor using the DHT.h library.
pinMode(A1, INPUT)	Sets the mode of operation - the input
Serial.print(",")	Displays a port on the monitor for someone
dht.readHumidity()	The function reads the humidity
dht.readTemperature()	The function reads the temperature
analogRead(MQ2)	The function reads the value from the gas sensor
digitalWrite(13, HIGH)	Turns on the light diode
digitalWrite(13, LOW)	Turns off the light diode

3 Assignments to laboratory work

1. Read the theoretical information.
2. Download and install libraries.
3. Assemble the sensors according to the scheme.
4. Write a program for working with sensors that performs the following functions:
- receive temperature and humidity data from DHT11 sensor;
- receive and send values from the liquid level sensor;
- receive and send values for lighting levels;
- receive and send values for the gas level;
- receive and send values of movement.
5. Make a report for laboratory work.

4 Test questions

1. What are the main sensors for Arduino?
2. Where is the motion sensor used?
3. What libraries need to be downloaded for the DTH11 sensor?
4. What is the function of Serial.begin(9600)?
5. What is the function dht.begin() used for?

5 Requirements to the content of the report

1. The objective of the lab.
2. Brief theoretical information.
3. Program code.
4. Screenshots showing the completed tasks.
5. Conclusions to the work.

6 Recommended literature

1. Tinkercad. [Electronic resource]. – Access mode: https://www.tinkercad.com/.
2. Arduino Tutorials. [Electronic resource]. – Access mode: https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials

Laboratory work 3
Raspberry Pi minicomputer implementation as a server for Smart building system

Objective: to learn how to install the Raspbian operating system on Raspberry Pi and how to connect Raspberry Pi to Arduino.

1 Brief theoretical information

Raspberry Pi is a one-size computer with a bank card size, originally designed as a budget system for computer science education, which has subsequently received much wider application and popularity. The first versions were created in 2011. Built on ARM architecture processor.

Raspberry Pi works mainly on operating systems based on Linux kernels, such as: Raspbian (Debian modification); Pidora (Fedora modification); Arch Linux ARM; Kali Linux.

Fig. 1 depicts components typical of the Raspberry Pi 2B model and their location on the board.

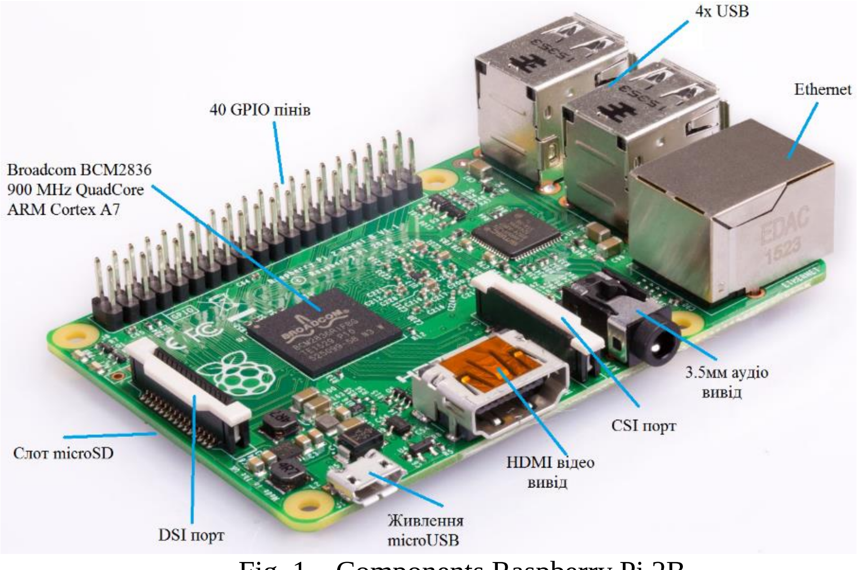


Fig. 1 – Components Raspberry Pi 2B

2 Execution order and discovery questions

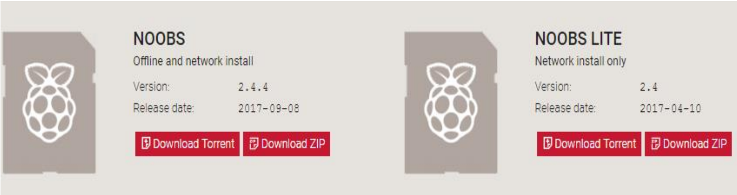
There are two ways to install Raspbian OS on Raspberry Pi:

– Downloading the NOOBS package from the official site to the MicroSD memory card and installing it further.

– Mounting a file-image of the Raspbian OS to the microSD memory card. In this case, you can start work immediately after turning on Raspberry Pi.

Example of installing Raspbian OS using the NOOBS package.

Download the package from the official site https://www.raspberrypi.org/downloads/noobs/. And choose NOOBS "Offline and network install" for installation without the Internet or NOOBS LITE "Network install only" for installation with the Internet (Fig. 2).



NOOBS
Offline and network install
Version: 2.4.4
Release date: 2017-09-08

NOOBS LITE
Network install only
Version: 2.4
Release date: 2017-04-10

Fig. 2 – Selection of the NOOBS package

Before burning NOOBS files to a microSD memory card, it must be formatted in the FAT32 file system.

Next, unpack the NOOBS archive on the microSD memory card.

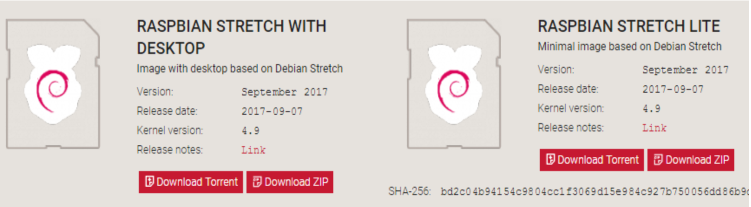
The next step is connecting the peripherals to the Raspberry Pi: USB keyboards, USB mice, monitor (via HDMI), and also install the microSD memory card in the appropriate slot. Lastly, the microUSB power connects.

In the window that appears, choose Raspbian OS (first option). The second option offers the manual partitioning of the memory card, the third option - download directly to the Scratch package - a program for creating computer graphics and animations.

Click "Install" and confirm the data recording on the microSD memory card. Wait for the installation and restart process to complete. In the Configuration Tool window (you can change the settings later), select the "Enable boot to Desktop" option and confirm the choice that will make the LXDE default for bootup. Then click on "Done" and agree to restart. After that everything will be ready to work.

Example of installing Raspbian OS by mounting an image file

First download the OS from the official site: https://www.raspberrypi.org/downloads/raspbian/. It is recommended to select "Raspbian Stretch With Desktop" (Fig. 3).



RASPBIAN STRETCH WITH DESKTOP

Image with desktop based on Debian Stretch

Version:	September 2017
Release date:	2017-09-07
Kernel version:	4.9
Release notes:	Link

RASPBIAN STRETCH LITE

Minimal image based on Debian Stretch

Version:	September 2017
Release date:	2017-09-07
Kernel version:	4.9
Release notes:	Link

SHA-256: bd2c04b94154c9804cc1f3069d15e984c927b750056dd86b9c

Fig. 3 – Downloading the image file

Install the Win32DiskImager utility on your computer (https://sourceforge.net/projects/win32diskimager/) and use the Raspbian OS file to insert a microSD memory card:

– It is necessary to unpack the archive with a file-image on a computer;

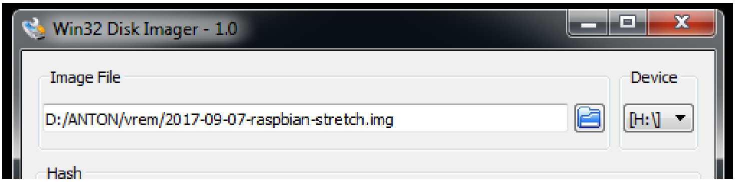
– Format the memory card as a standard Windows device;

– In the Win32DiskImager program, it is necessary to choose an image file of the Raspbian OS (for example: "2017-09-07-raspbian-stretch.img");

– In the "Device" section, select the name of the microSD memory card. Care should be taken if damage to the hard drive can be done by mistake (Fig. 4);

– Finally, select "Write" and wait for the completion of the recording process. Approximate recording time is 5-10 minutes.

–



Win32 Disk Imager - 1.0

Image File: D:/ANTON/vrem/2017-09-07-raspbian-stretch.img

Device: [H:\]

Fig. 4 – Selection of the image file and write device

Next, connect the peripherals to the Raspberry Pi: USB keyboards, USB mice, monitor (via HDMI), and also need to install the microSD memory card in the appropriate slot. Lastly, the microUSB power connects.

Raspbian OS will be downloaded and everything will be ready for work.

If you are using Unix-like OS:

– it is necessary to format the microSD memory card in FAT 32 / ext2;

– execute the command: sudo dd bs = 4m if = / home / user / raspbian.img of = / dev / disk1.

Next, connect the peripherals to the Raspberry Pi: USB keyboards, USB mice, monitor (via HDMI), and also need to install the microSD memory card in the appropriate slot. Lastly, the microUSB power should be connected.

Raspbian OS will be downloaded and everything will be ready for work.

It's important to remember that the Raspberry Pi launch occurs immediately after the MicroUSB connection, and turn off Raspberry Pi as follows:

– Complete all actions;

– Click on the logo Raspberry Pi in the upper left corner;

Select the "Shutdown" option (Fig. 5) and confirm in the new window - "Shutdown" (Fig. 6).

Only after the Raspberry Pi is turned off you can disconnect the MicroUSB cord and change the composition of the periphery connected to the Raspberry Pi.

Example of the first launch of the Raspbian OS

If the download was enabled in the graphical environment, the user's password is not required, but if the password has been changed, it may need to be entered.

In the case of a console mode, you must enter a login and password ("pi" and "raspberry", respectively). To start the graphical environment, you must execute the command "startx".

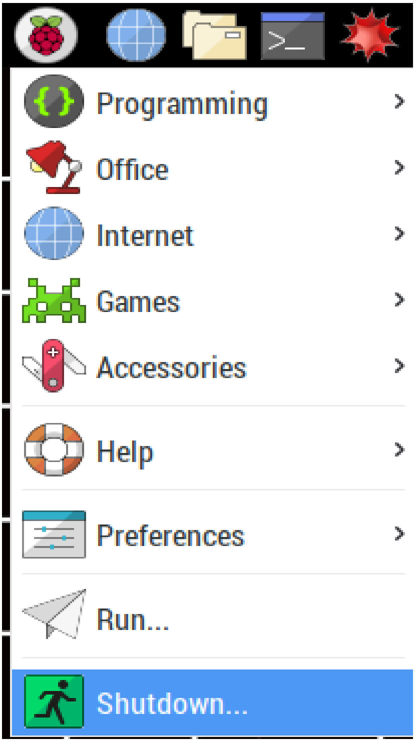


Fig. 5 – Turning off Raspberry Pi

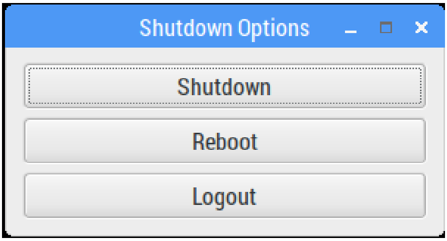


Fig. 6 – Disable Menu for Raspberry Pi

At the top of the screen is a taskbar, a quick access bar and a menu button (Fig. 7).

The Quick Launch toolbar contains:

– Browser;

– Explorer;

– Terminal utility;

– Wolfram packages.

By clicking on the menu button (Raspberry Pi logo) you can view the programs installed on the system (Fig. 8).

The menu contains the following sections:

– Programming: contains programs and tools for programming (Python IDE, Geany, BlueJ, etc.);

– Office: contains "office" programs (OpenOffice package);

– Internet: contains programs and means for work on the Internet;

– Games: contains games (MineCraft);

– Accessories: contains additional accompanying programs (Archiver, calculator);

– Sound & Video: contains video and audio players (VLC Player);

– System Tools: contains system programs (Midnight Commander);

– Help: help;

– Preferences: system setup;

– Run: execute the action;

– Shutdown: switch to the shutdown window.

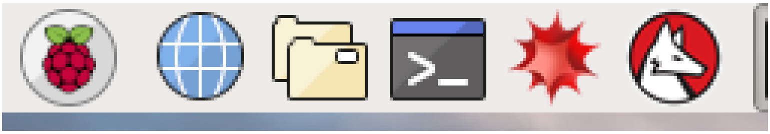


Fig. 7 – Quick start panel

Basically, for most actions in Raspbian (as in any Unix-like OS), the Terminal utility is used (Fig.9). To start it, click on the corresponding icon (Fig. 10).

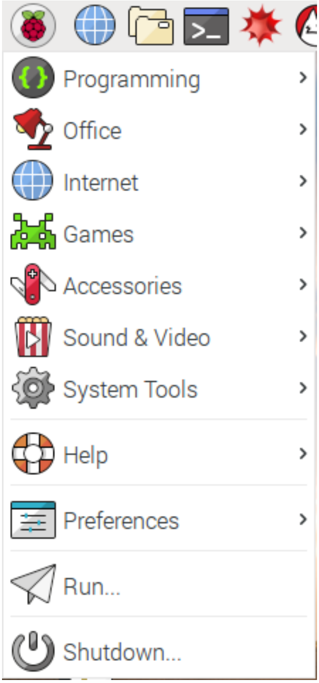
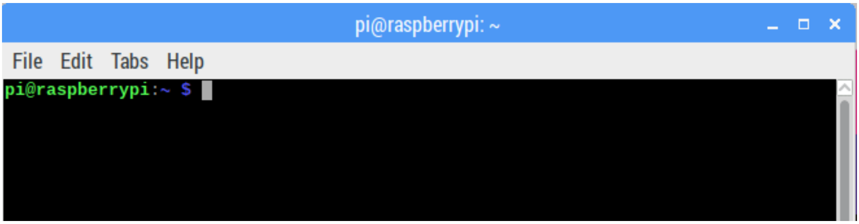


Fig. 8 – Raspberry Pi Menu



```
pi@raspberrypi:~ $
```

Fig. 9 – Window Terminal

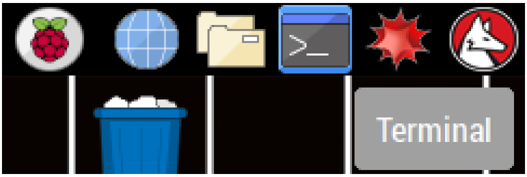
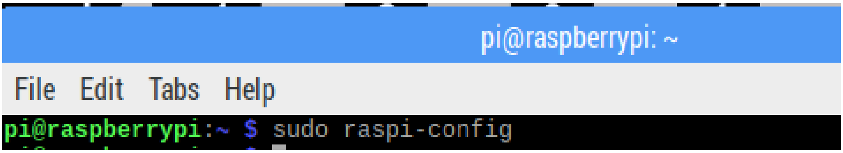


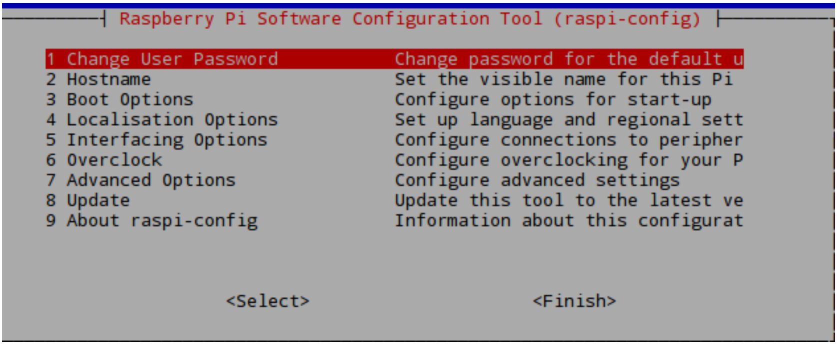
Fig. 10 – Terminal icon

In the process, you may need to change the system configuration, change the user password, turn on the camera, and more. All this can be done using the sudo raspi-config command (Fig. 11). An example of the configuration window is shown in Fig. 12.



```
pi@raspberrypi:~ $ sudo raspi-config
```

Fig. 11 – Run the sudo raspi-config command



```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

1 Change User Password         Change password for the default u
2 Hostname                     Set the visible name for this Pi
3 Boot Options                 Configure options for start-up
4 Localisation Options         Set up language and regional sett
5 Interfacing Options          Configure connections to peripher
6 Overclock                    Configure overclocking for your P
7 Advanced Options             Configure advanced settings
8 Update                       Update this tool to the latest ve
9 About raspi-config           Information about this configurat

              <Select>                   <Finish>
```

Fig. 12 – Setup Window

More about menu options:

– Change User Password - change the password of the user;

– Hostname - the name of the computer on the network;

– Boot options - system boot configuration;

– Localisation Options - language selection and regional settings;

– Interfacing Options - interface settings (camera, SSH, VNC, SPI, I2C, Serial, 1-Wire, GPIO);

– Overclock - increase the frequency of the processor;

– Advanced Options - additional options;

– Update - update the raspi-config program;

– About raspi-config - information about the program.

After completing the setting, choose Finish. The OS will request permission to restart. You must agree and wait for the restart to end.

When changing the system settings, be careful, the error can lead to unwanted consequences.

Example of setting up the Internet

In the case of a network that has a DHCP server and an automatic distribution of addresses (such as a home network), there is no need to do anything else.

If there is no DHCP, you can set network parameters using the console. To do this, you need to run a terminal where you should open the configuration file for the network interfaces using the command: sudo nano / etc / network / interfaces. To the file you need to add the following:

```
     iface <interface> inet static
     address <ip-address>
     netmask <subnet mask>
     gateway <gateway addresses>
     dns-nameservers  <addresses  of  DNS  servers,
separated by a space>
     auto <interface>,
```

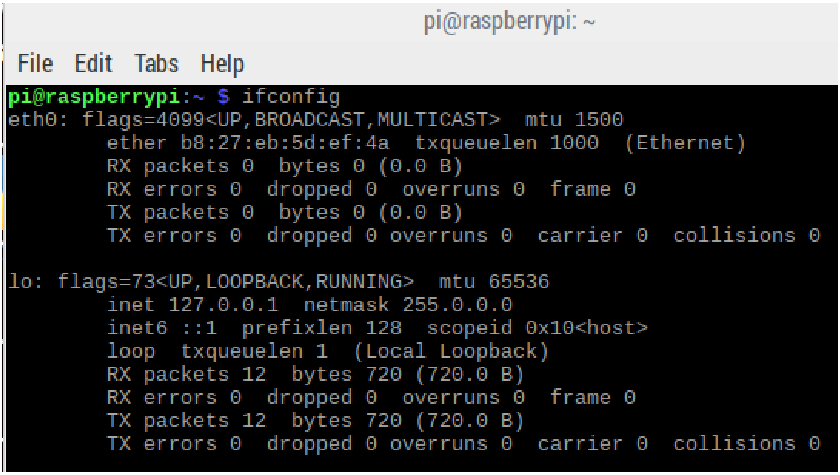
where the interface is a network interface connected to Raspberry P (most often eth0); ip-address is the address to be assigned; subnet mask - mask; the gateway address is the ip address of the computer that is using the gateway network.

Then exit with saving changes, for which you press Ctrl + X, confirm the save by pressing the Y key, and then Enter to confirm the file is overwritten. After changing the file, you must restart Raspberry Pi by executing the sudo reboot command.

To check the network settings, follow the ifconfig command (Fig. 13).

To work with the proxy server, you need certain lines for the / etc / environment (system-wide proxy settings) and /etc/apt/apt.conf (apt-get batch manager settings). To the / etc / environment file, add:
export http_proxy = "http: // host: port", and to the file /etc/apt/apt.conf: Acquire :: http :: proxy "http: // host: port"; .

After performing these actions, you will be able to use network functions, including the apt-get manager.



```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether b8:27:eb:5d:ef:4a  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 12  bytes 720 (720.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 720 (720.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Fig. 13 – An example of executing the ifconfig command

Example of connecting Arduino with a loaded sketch

In order to connect Arduino to Raspberry Pi you need:

– 1. Take sketch (program) from laboratory work # 1 and connected sensors to Arduino. Run the sketch and leave it to work.

– 2. Extract the Arduino USB-wire from your computer.

– 3. Insert the Arduino USB-wire into the running Raspberry Pi (Fig. 14)

– 4. Provide full access to the port to be used by Arduino. To do this write at the terminal: `chmod 777 / dev / ttyACM0`

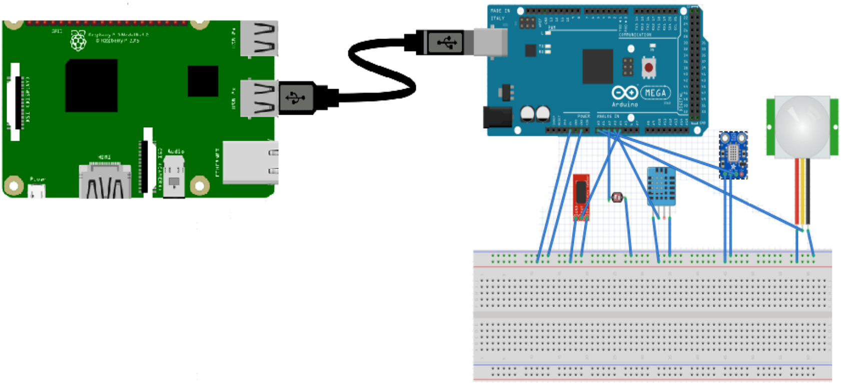


Fig. 14 – Scheme of Arduino to Raspberry Pi connection

3 Assignments to laboratory work

1. Acquainted with theoretical information.
2. Run the Raspbian installation and base configuration on Raspberry Pi.
3. Configure Internet connection to Raspberry Pi.
4. Connect Arduino to Raspberry Pi.
5. Make a report for laboratory work.

4 Test questions

1. What Is Raspberry Pi?
2. On which OS is Raspberry Pi? Which components are typical for Raspberry Pi 2B?
3. What are the ways to install Raspbian OS on Raspberry Pi?
4. How to connect Arduino to Raspberry Pi?

5 Requirements to the content of the report

1. Objective of the lab.
2. Brief theoretical information.
3. Screenshots showing the completed tasks.
4. Conclusions to the work.

6 Recommended literature

1. Raspberry Pi. Downloads. https://www.raspberrypi.org/downloads/
2. Raspberry Pi Azure IoT Online Simulator. URL: https://azure-samples.github.io/raspberry-pi-web-simulator/.

Laboratory work 4
Integration of SBS subsystems based on OpenHAB platform

Objective: to learn how to install and configure OpenHAB on Raspberry Pi.

1 Brief theoretical information

OpenHAB (Open Home Automation Bus) - a Java-based open source smart house implementation project, distributed under the GPLv3 license, and Jetty is used to organize the work of the web-interface.

This open platform is capable of performing functions such as turning on and off the light, controlling sockets, and so on.

OpenHAB provides tools for organizing the work of various systems, equipment and home automation interfaces. At the same time OpenHAB does not depend on protocols and equipment, providing a separate level of abstraction, which allows you to interact with different types of devices and software. To determine the control logic used scripts, written in a special subject-oriented programming language, developed with the help of Eclipse -Xtext.

The main OpenHAB service is the event bus. Modules that do not require state tracking use this bus to share event information with other modules.

There are two main types of events:

1. Commands that initiate any action or change the state of a particular item or device.

2. Status updates that report changes in the state of a particular item or device.

Binding protocols that communicate with real devices should communicate with each other precisely through the event bus. This ensures communication between the modules.

OpenHAB uses a very powerful expression language, which defines scenarios. A script or script is a code block defined by the user and can be called and used in different places. Scripts are placed in the openhab / configurations / scripts folder. The demo.script demo file is in the workspace.

Scripts may also be inside a rule file in the openhab /

configurations / rules folder: they are used to define the rule execution block. Each rule consists of two parts: one contains action switches, in other scripts - for their execution.

For efficient use of scripts OpenHAB provides access:

- to all items (access can be contacted by name);
- to all statuses and teams;
- to standard actions to perform various operations.

The script is identified by the name (in the demo.script file, the name of the demo script). Each script always returns the value that is the result of the last expression contained therein.

Scripts may be called:

- from the rules;
- from the XMPP console;
- from the entries in the Google Calendar.

2 Execution order and discovery questions

Installation and configuration OpenHAB on Raspberry Pi.

To install OpenHAB on Raspberry Pi, you must perform a sequence of actions and commands (using a terminal) as described below.

1) Download key:

```
wget -qO -
'https://bintray.com/user/downloadSubjectPublicKey?username=
openhab '| sudo apt-key add -
```

2) Install the OpenHAB Repository:

```
echo "deb http://dl.bintray.com/openhab/apt-repo stable main" |
sudo tee /etc/apt/sources.list.d/openhab.list
```

3) Update system files:

```
sudo apt-get update
```

4) Install OpenHAB:

```
sudo apt-get install openhab-runtime
```

5) Add a user to the group:

```
sudo usermod -a g openhab pi
```

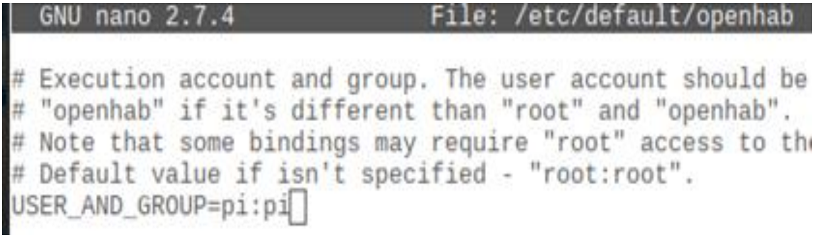
6) Run OpenHAB:

```
sudo systemctl enable openhab
```

7) Install the add-on for GPIO:

```
sudo apt-get install openhab-addon-io-gpio
```

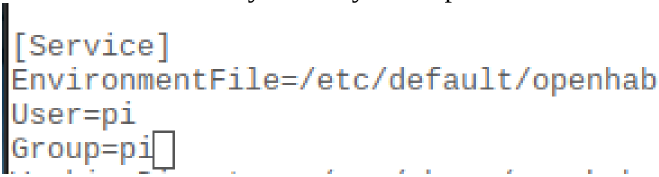
sudo apt-get install openhab-addon-binding-gpio
8) Install add-ons for working with ports:
sudo apt-get install openhab-addons-binding-serial
9) Specify a user name and group name (Fig. 1, Fig. 2):
sudo nano / etc / default / openhab



```
  GNU nano 2.7.4                    File: /etc/default/openhab

# Execution account and group. The user account should be
# "openhab" if it's different than "root" and "openhab".
# Note that some bindings may require "root" access to th
# Default value if isn't specified - "root:root".
USER_AND_GROUP=pi:pi
```

Fig. 1 – Change the user name and group

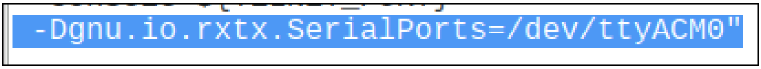
sudo nano /usr/lib/systemd/system/openhab.service



```
[Service]
EnvironmentFile=/etc/default/openhab
User=pi
Group=pi
```

Fig. 2 – Change the user name and group

10) Overload systemd:
sudo systemctl daemon-reload
11) Add port to be used with Arduino:
sudo nano /usr/share/openhab/bin/openhab.sh
In the section "JAVA_ARGS_DEFAULT" add to the end (Fig. 3):

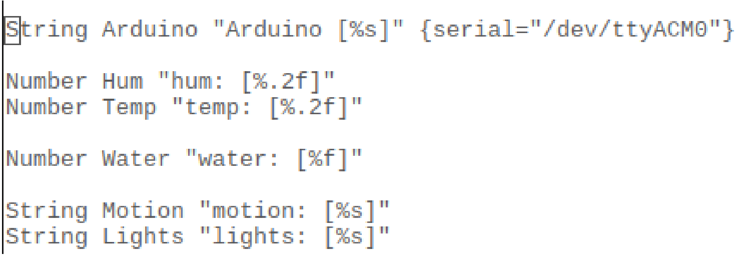


```
 -Dgnu.io.rxtx.SerialPorts=/dev/ttyACM0"
```

Fig. 3 – Adding a port

12) Overload OpenHAB:
sudo service openhab restart

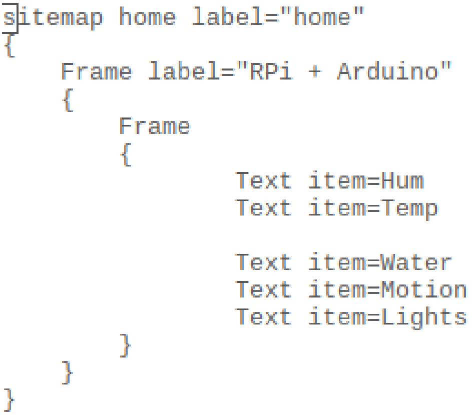
13) Write code to receive data from Arduino (Fig. 4):
sudo nano /etc/openhab/configurations/items/home.items



```
String Arduino "Arduino [%s]" {serial="/dev/ttyACM0"}

Number Hum "hum: [%.2f]"
Number Temp "temp: [%.2f]"

Number Water "water: [%f]"

String Motion "motion: [%s]"
String Lights "lights: [%s]"
```

Fig. 4 – Arduino Data Acquisition Code

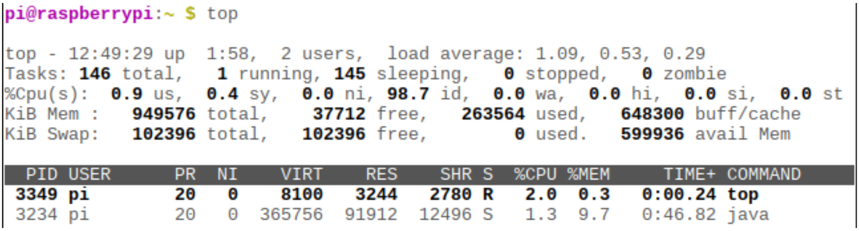
14) Create interface for displaying data (Fig. 5):
sudo nano /etc/openhab/configurations/sitemaps/home.sitemap



```
sitemap home label="home"
{
    Frame label="RPi + Arduino"
    {
        Frame
        {
                Text item=Hum
                Text item=Temp

                Text item=Water
                Text item=Motion
                Text item=Lights
        }
    }
}
```

Fig. 5 – Creating an interface for displaying data

15) Overload OpenHAB:
sudo service openhab restart
16) Find the Java process and remember its number (Fig. 6):
top



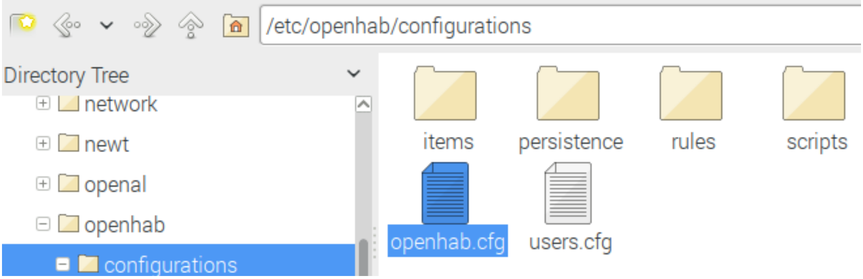
```
pi@raspberrypi:~ $ top

top - 12:49:29 up  1:58,  2 users,  load average: 1.09, 0.53, 0.29
Tasks: 146 total,   1 running, 145 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.9 us,  0.4 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   949576 total,    37712 free,   263564 used,   648300 buff/cache
KiB Swap:   102396 total,   102396 free,        0 used.   599936 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 3349 pi        20   0    8100   3244   2780 R   2.0  0.3   0:00.24 top
 3234 pi        20   0  365756  91912  12496 S   1.3  9.7   0:46.82 java
```

Fig. 6 – Process mapping

17) Stop the Java process. For example, the Java process number is 3234, then:

kill 3234

18) Rename the openhab_default.cfg file at openhab.cfg in /etc/openhab/configurations/ (Fig. 7)

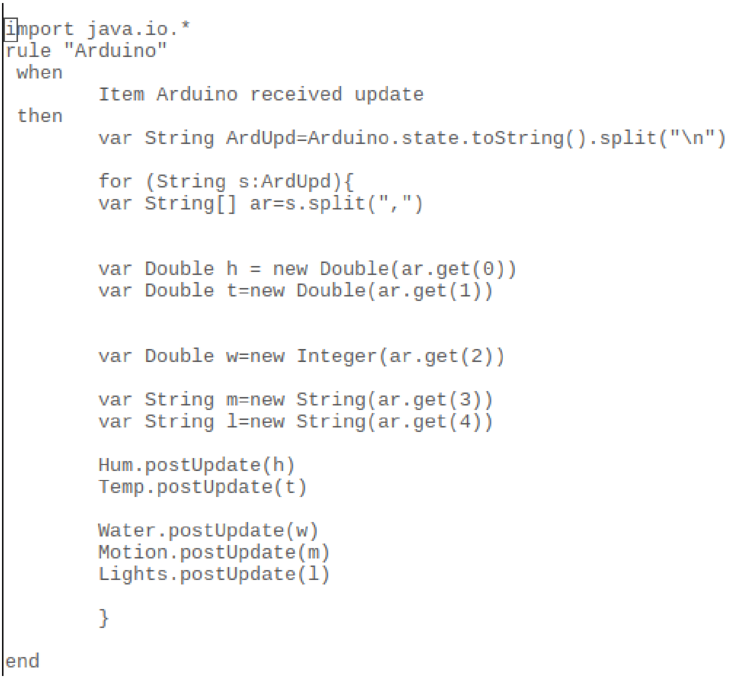


/etc/openhab/configurations

Fig. 7 – File renamed

20) Create OpenHAB rule for sensor control (Fig. 8):

sudo nano etc / openhab / configurations / rules / home.rules



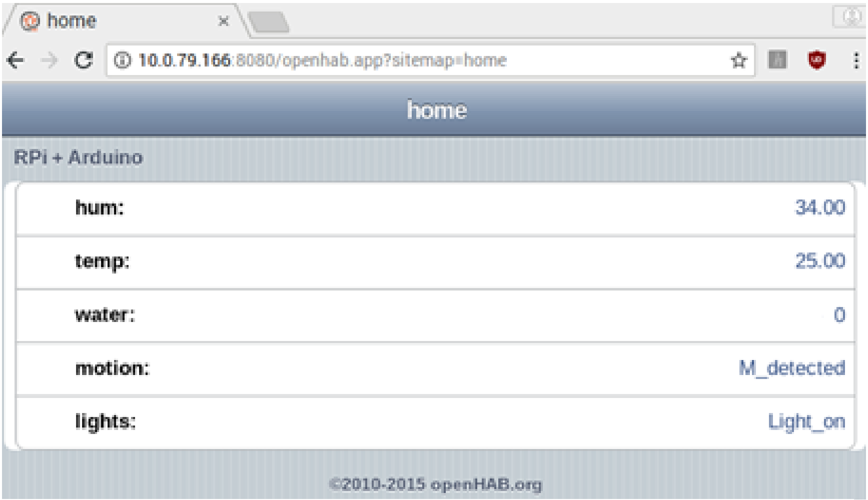
```
import java.io.*
rule "Arduino"
 when
        Item Arduino received update
 then
        var String ArdUpd=Arduino.state.toString().split("\n")

        for (String s:ArdUpd){
        var String[] ar=s.split(",")


        var Double h = new Double(ar.get(0))
        var Double t=new Double(ar.get(1))


        var Double w=new Integer(ar.get(2))

        var String m=new String(ar.get(3))
        var String l=new String(ar.get(4))

        Hum.postUpdate(h)
        Temp.postUpdate(t)

        Water.postUpdate(w)
        Motion.postUpdate(m)
        Lights.postUpdate(l)

        }

end
```

Fig. 8 – Rule code

21) Connect Arduino with a loaded sketch;
22) Provide full access to the port to be used by Arduino:
chmod 777 / dev / ttyACM0
23) Go to the openHab directory:
cd / usr / share / openhab / bin
24) Run the .sh file:
sudo / openhab.sh
24) Go to: RaspberryPi-IP: 8080/openhab.app?sitemap = home (Fig. 9, Fig. 10).

3 Assignments to laboratory work

1. Review the theoretical information.
2. Install and run OpenHAB on Raspberry Pi.
3. Make a report for lab.



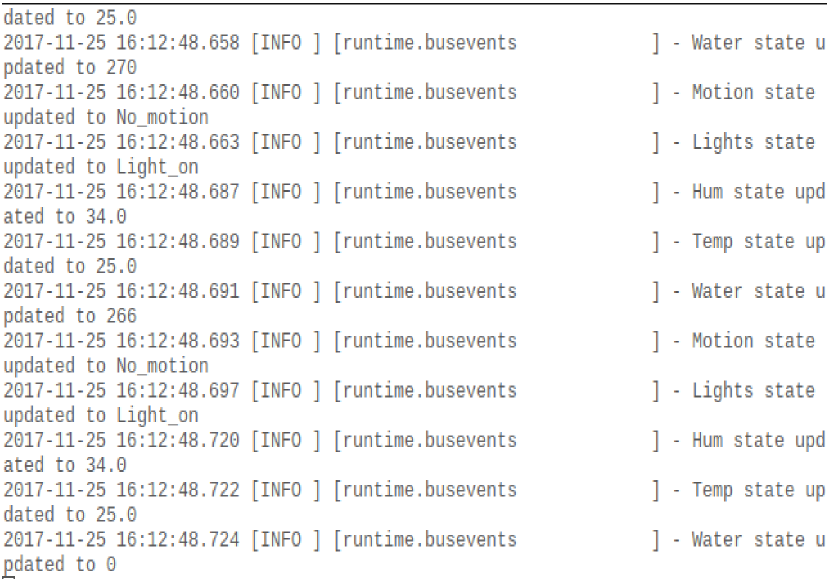
home

RPi + Arduino

hum:	34.00
temp:	25.00
water:	0
motion:	M_detected
lights:	Light_on

©2010-2015 openHAB.org

Fig. 9 – OpenHab page



```
dated to 25.0
2017-11-25 16:12:48.658 [INFO ] [runtime.busevents             ] - Water state u
pdated to 270
2017-11-25 16:12:48.660 [INFO ] [runtime.busevents             ] - Motion state
updated to No_motion
2017-11-25 16:12:48.663 [INFO ] [runtime.busevents             ] - Lights state
updated to Light_on
2017-11-25 16:12:48.687 [INFO ] [runtime.busevents             ] - Hum state upd
ated to 34.0
2017-11-25 16:12:48.689 [INFO ] [runtime.busevents             ] - Temp state up
dated to 25.0
2017-11-25 16:12:48.691 [INFO ] [runtime.busevents             ] - Water state u
pdated to 266
2017-11-25 16:12:48.693 [INFO ] [runtime.busevents             ] - Motion state
updated to No_motion
2017-11-25 16:12:48.697 [INFO ] [runtime.busevents             ] - Lights state
updated to Light_on
2017-11-25 16:12:48.720 [INFO ] [runtime.busevents             ] - Hum state upd
ated to 34.0
2017-11-25 16:12:48.722 [INFO ] [runtime.busevents             ] - Temp state up
dated to 25.0
2017-11-25 16:12:48.724 [INFO ] [runtime.busevents             ] - Water state u
pdated to 0
```

Fig. 10 – Output in the console

4 Test questions

1. What is OpenHAB?
2. What are the scripts used for it?
3. From where the scripts can be called?
4. Which command can stop the process?

5 Requirements to the content of the report

1. Objective of the lab.
2. Brief theoretical information.
3. Screenshots showing the completed tasks.
4. Conclusions to the work.

6 Recommended literature

1. OpenHAB Documentation. [Electronic resource]. – Access mode: https://www.openhab.org/docs/tutorial/

2. Getting Started with OpenHAB Home Automation on Raspberry Pi [Electronic resource]. – Access mode: https://www.makeuseof.com/tag/getting-started-openhab-home-automation-raspberry-pi/

3 DEVELOPMENT OF SMART SBC SYSTEMS

1 Principles of interaction of Smart House & IoT subsystems in the local network and Internet

The most important factors affecting health, productivity and quality of a person's rest are the conditions in which they are located. These terms can be divided into several groups:

1. Air quality, which consists of a comfortable ratio of temperature, humidity, flow velocity and level of harmful impurities in the air;
2. Lighting in the room and surrounding areas;
3. Water and heat supply;
4. Human and Home Security.

1.1 General information

Automated systems that solve problems related to the provision of necessary conditions for these groups, today must take measures to save on all types of resources: electricity, fuel, gas, water. The success is to optimally solve the individual components of this task and in the integrated management of automation systems of premises.

For systems of automation of residential constructions, networks with properties that are more suited to the living conditions and human psychology are developed. It is human interaction, automation systems and individual components that interact with the principles of IoT.

The features of household networks and protocols, above all, are that automatic devices in the house can appear gradually, without a detailed master plan. Therefore, expansion of the network should occur easily, without difficult reprogramming of all other elements and without excessively stringent requirements for cable laying. One of the most popular standards in this area was the development of ABB i-bus®EIB – European Installation Bus. After the unification of many manufacturers, the network was named EIB / KNX.

The network has four main means of communication – an independent low-voltage network, modulation of signals by power cables PowerBus, infrared and radiofrequency communications.

Several companies (Siemens, Moeller) manufacture 868MHz radio-frequency devices (RF-system), which meets the EIB's requirements.

This version of the network allows you to completely eliminate the control wires and significantly reduce the length of the power circuits.

The addresses of each component of the EIB system are subdivided into physical and logical (group) addresses. The physical address determines the location of the component in the system and consists of a sequence of digits separated by a dot corresponding to their level. The first group of digits denotes a segment, the second is a line, and the third is the component number. For example, address 6.11.38 belongs to the thirty-eighth component of the eleventh line of the sixth segment. The physical address of the component is stored in the ROM component and may be modified if it is necessary. The logical address connects components of the system (for example, the sensor and the console). When designing, it is possible to select any logical address from 16 main groups, each of which contains 2048 subgroups. The main groups can be divided by function, ie control of lighting, heating, ventilation, etc. In this way, one component of the system can have not one, but several logical addresses.

The protocol for exchanging data in the EIB / KNX network between devices is based on the reception / transmission of short messages – telegrams. A telegram contains several information blocks – a priority field, an address field, a command field. The transfer rate is 9600 baud, that is to transfer one bit of information to 140 μs. Each device permanently operates a data receiver connected to the line. If a telegram appears on the line, all receivers begin to receive it. Then the telegram analysis is performed. If the address matches the item in the list stored in the memory of the individual device, then this device executes the command and sends a message about its execution.

To prevent collisions (conflicts with information damage) when exchanging messages, CSMA / CA (Carrier Sense Multiple Access / Collision Avoidance) access protocol is used. The EIB protocol accepted exceptionally detection of collisions to block communication. The device that noticed an error sends the Shut Up command (0x0000FFFF) which prevents other devices from transmitting information. Therefore, all participants in the data exchange "silence". In each of them, the generator generates a random pause from 1 to 3 seconds, after which the device again tries to restore the connection, if the network is free. Each device has three attempts.

The EIB system has exceptional flexibility. One cable combines all electrical appliances at home. This simplifies switching systems (especially if some components are connected to the radio network), the cost of design and laying of cables is significantly reduced, reducing the risk of fire. System expansion and function change are achieved by simply rearranging, adding or reprogramming components of the system. Each component of the EIB system can interact with any other component (or simultaneously with a group of components) that are part of the system. The EIB system allows you to control electrical power systems at home both locally (in a specific room) and centralized (from the control panel or computer). Use of timers, light sensors, wind forces, temperatures, movements, etc. makes possible the fully automatic decentralized operation of all house systems depending on the season, day of week (working day / weekend), time of day and specific external conditions.

1.2 Classification and selection of intellectual components, sensors, executive units of Building Management Systems

The EIB / KNX network combines three types of devices:

1. Sensors – analog and relay temperature sensors, motion sensors or presence, illumination, window integrity, pressure, humidity, speed and air quality, etc.;

2. Actuators – power executive devices (relays and TRIAC voltage regulators), modules for transmitting analogue 0 – 10 V or 4 – 20 mA and discrete signals, routers (devices for distributing the network coverage area);

3. Managing devices – buttons, remote controls, personal computers with communication modules, climate control modules, room control modules or buildings with the ability to communicate with external networks GSM, Ethernet, Internet, dispatching consoles.

Table 1 shows some components of the Eaton / Moeller xComfort system with the EIB RF-System. Selection of components for solve typical tasks can possible by this table.

For example, in the basic **system of inflow and exhaust ventilation** (Fig. 1) are needed two universal dimming actuators CDAU-01/04 (CDAE-01/04, CDAE-01/05) for fan motors M1, M3; analogue actuator CAAE-01/01 for water valve driver M2; two temperature sensors CSEZ-01/01 with input channels CTEU-02/01 (T1, T2); binary input unit CBEU-02/03 for differential pressure sensor dP.

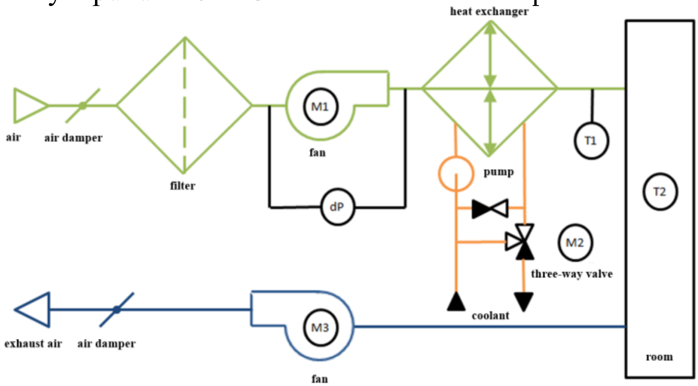
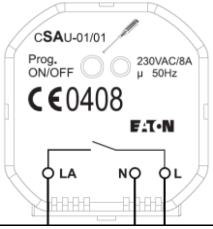
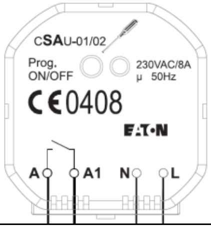
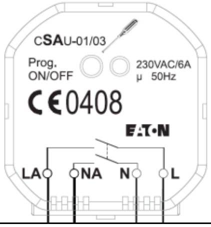
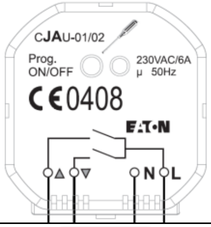
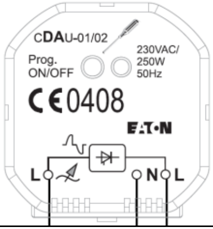
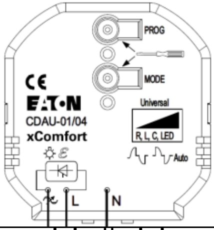
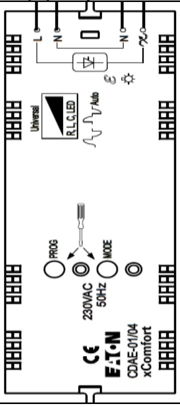
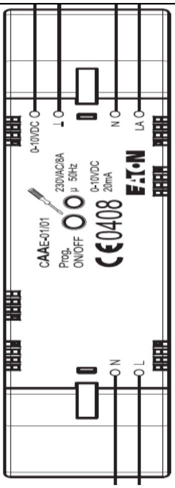


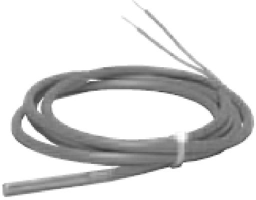
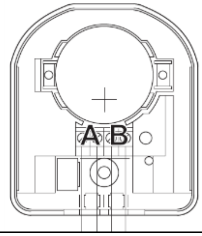
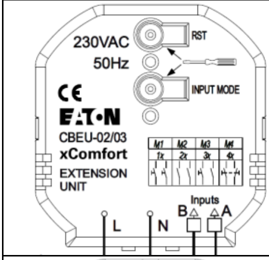
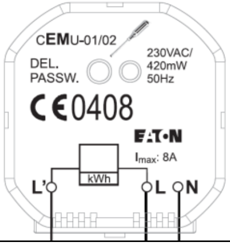
Fig. 1 – Functional diagram of the climate system

Table 1. Components of Eaton / Moeller xComfort

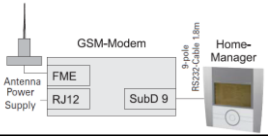
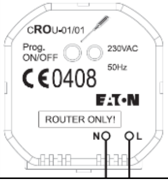
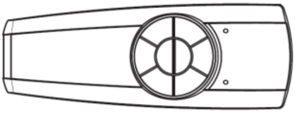
	Type	Technical data
Actuators		
 CSAU-01/01 Prog. ON/OFF 230VAC/8A µ 50Hz CE0408 EATON LA N L	CSAU-01/01	Switching actuator, 8 A, 230 VAC
 CSAU-01/02 Prog. ON/OFF 230VAC/8A µ 50Hz CE0408 EATON A A1 N L	CSAU-01/02	Switching actuator Voltage-Free, 8 A, 230 VAC
 CSAU-01/03 Prog. ON/OFF 230VAC/6A µ 50Hz CE0408 EATON LA NA N L	CSAU-01/03	Switching actuator All-Poles, 6 A, 230 VAC

 CJAU-01/02 Prog. ON/OFF 230VAC/6A µ 50Hz CE0408 EATON N L	CJAU-01/02	Shutter actuator, 6 A, 230 VAC
 CDAU-01/02 Prog. ON/OFF 230VAC/ 250W 50Hz CE0408 EATON L N L	CDAU-01/02	Dimming actuator, 250 VA, 230 VAC
 PROG MODE CE EATON CDAU-01/04 xComfort Universal R, L, C, LED Auto L N	CDAU-01/04	Universal dimming actuator for R, L, C load, LED and phase section energy saving lamp, Fluorescent, Low voltage halogen lamp with electronic and coil transformer, 100 VA, 230 VAC
 L N N Universal R, L, C, LED Auto PROG MODE 230VAC 50Hz CE EATON CDAE-01/04 xComfort	CDAE-01/04 CDAE-01/05	Smart Universal dimming actuator for R, L, C load, LED and phase section energy saving lamp, Fluorescent, Low voltage halogen lamp with electronic and coil transformer, 230 VAC, 250 VA (CDAE-01/04), 500 VA (CDAE-01/05).

 0-10VDC ⊥ N LA 230VAC/8A 50Hz 0-10VDC 20mA CAAE-01/01 Prog. ON/OFF CE0408 N L	CAAE-01/01 CAAE-01/02	Analogue actuator, Power output: 8 A, 230 VAC; Analogue output: 20 мА, 0-10 VDC (CAAE-01/01), 1-10 VDC (CAAE-01/02)

Sensors		
	CSEZ-01/01 CSEZ-01/36	Temperature Sensor Power supply Via temperature input channel CTEU-02/01 Measuring range: -50 to +200°C Sensor element: PT1000
	CSEZ-01/16	VOC Air Quality Sensor Power supply 15-24VDC, Sensor Mixed gas VOC, Output voltage 0-10VDC linear to air quality
	CBEU-02/02	Door / window binary input unit with battery, 2 inputs (A, B).
 230VAC 50Hz RST INPUT MODE CE EATON CBEU-02/03 xComfort EXTENSION UNIT M1 M2 M3 M4 1x 2x 3x 4x Inputs B A L N	CBEU-02/03	Binary input unit, 2 inputs (A, B), 4 modes.
 CEMU-01/02 DEL. PASSW. 230VAC/ 420mW 50Hz CE0408 EATON Imax: 8A kWh L' L N	CEMU-01/02	Energy meter sensor, 8A, 230 VAC

	CSEZ-01/18	Water leakage sensor Power supply 9V via battery 6LR61, Sensor Detection of leakages, sensor is removable, Alarm Acoustic signal emitter, approx. 85dB at 3m
	CSEZ-01/19	Smoke Detector Power supply 9V Sensor Photo-electronic smoke detector, stray light sensor according to Tyndall effect
	CSEZ-02/08	Wind Rain Sensor, 230 VAC, 17 mA, Wind speed: 3–12 m/s
	CSEZ-01/12 CBMA-02/01	PIR Motion sensor, 230 VAC, 15 mA, Detection 2xDual, Area covered: 200 ° (CSEZ-01/12), 110 ° (CBMA-02/01), Approx. 16 m at h 2 m.
	CSEZ-01/14 CSEZ-01/15	Brightness sensor 24 VDC, 3–300 lux… 600–60k lux Output 0–10 VDC Indoor – CSEZ-01/14 Outdoor – CSEZ-01/15
	CSEZ-01/17	Humidity sensor with Temperature PT1000, Power supply 15-24VDC, Relative humidity capacitive sensor, Operating range 0-100% Measuring range 5-95% rel. Humidity, Output voltage 0-10 VDC, Temperature sensor PT1000, Range -20 to + 60°C
	CTEU-02/01	Temperature Input Unit, Power supply 3V via CR2477 N battery Connections 4-pole terminal strip 2 inputs for CSEZ-01/01

Control		
	CTAA-01/04 CTAA-02/04 CTAA-04/04	Push-button. Power supply 3V CR2450N battery. Number of rockers depending on type
	CRMA-00/01	Room Manager
	CHMU-00/02	Home Manager
 GSM-Modem Antenna Power Supply FME RJ12 SubD 9 9-pole RS232-Cable 1.8m Home- Manager	CKOZ-00/02 CKOZ-00/06	GSM-Modem
 CROU-01/01 Prog. ON/OFF 230VAC 50Hz CE0408 EATON ROUTER ONLY! N L	CROU-01/01	RF Router
	CHSZ-12/03	12-Channel Remote Control
	CKOZ-00/03 CKOZ-00/11	Communication Interface USB/RS232

1.3 Organization of the interaction of subsystems and elements of Smart House and IoT based on Moeller / Eaton xComfort

To build a system of intelligent control of a building, it is necessary to pick up sensors, executive and control elements of the system and to perform their connection. Fig. 2 shows an example of an electrical circuit diagram of a system built on the basis of Eaton / Moeller xComfort with the radio network.

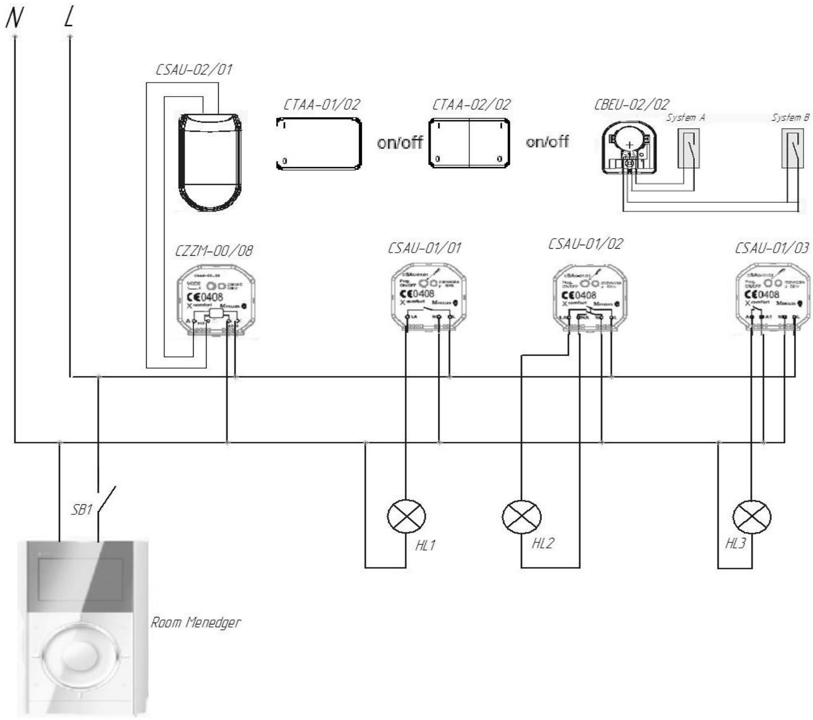


Fig. 2 – Electric circuit diagram of the system

Between the elements, in addition to the wired power connections, there is also radio communication and bilateral exchange of information according to the EIB protocol.

Training 1
System setup without intellectual control unit

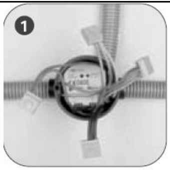
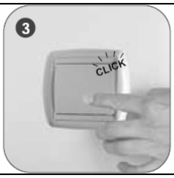
System setup works in several stages, examples for which are listed below.

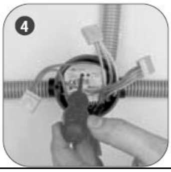
1. Establishing communication between individual elements

1.1. Set the wireless connection manually between the CTAA-01/02 one-button switch and the CSAU-01/01 actuator so that when pressing CTAA-01/02, the CSAU-01/01 is activated and the HL1 lamp is activated.

1.2. Establish a wireless communication manually between the two-button switch CTAA-02/02 and actuators CSAU-01/02, CSAU-01/03 in such a way that when the first button CTAA-02/02 was pressed, the CSAU-01/02 was activated and turned on lamp HL2, and when the second button CTAA-02/02 was pressed, CSAU-01/03 was triggered and the HL3 lamp was turned on.

Table 2. Set up communication of actuators and sensors

Establishing communication	
	Make sure the actuator is connected to the network and ready for operation
	Activate the programming button on the actuator with a short press of no more than 0.5 seconds (the red LED should light up)
	Press the corresponding control element: the switch button, or the remote control, the button on the Room-Manager (with the red diode blinking twice)

	Deactivate the programming button to complete the operation by short pressing not more than 0.5 s (the red diode and the corresponding lamp must go out)
Communication gap	
	Activate the programming button on the actuator with a long press of more than 2 seconds (the red LED should light up)
	Press the appropriate controller: the element of the switch button, or the remote control, the button on the Room Manager (with the red diode blinking twice)
	Deactivate the programming button to complete the operation by briefly pressing no more than 0.5 seconds (the red diode and the lamp should go out)

Training 2
Visualization and configuration of the wireless connection using the Moeller RF-System software environment

The system includes the following radio elements:

1. Remote control 12K – remote control;

2. Room-Manager – room processing and visualization device CRMA data 00/01;

3. Motion Detector – motion sensor CSAU-02/01 with actuator CZZM-00/08;

4. Bin Batt – window sensors with actuator CBEU-02/02;

5. Switching actuator 1 – actuator CSAU-01/01, which is responsible for turning on the HL1 lamp;

6. Switching actuator 2 – actuator CSAU-01/02, which is responsible for turning on the HL2 lamp;

7. Switching actuator 3 – actuator CSAU-01/03, which is responsible for turning on the HL3 lamp;

8. Push-button – wireless switch CTAA-01/02;

9. Push-button 2-fold – wireless switch with two buttons CTAA-02/02;

10. USB Gateway – Communication device, USB interface.

11. Push-button – wireless switch of lighting;

12. Section of lamps 1,2,3 – actuators CSAU-01/01, which are responsible for switching on sections of lamps;

13. CAAE-01/01 – analog control actuator of the ventilation system;

14. CAAE-01/02 – analog control actuator of the heating system;

15. Light sensor CSEZ-01/14;

16. Sensor of relative humidity of air CSEZ-01/17;

17. Air quality sensor CSEZ-01/16;

18. Temperature Input – A two-channel temperature sensor.

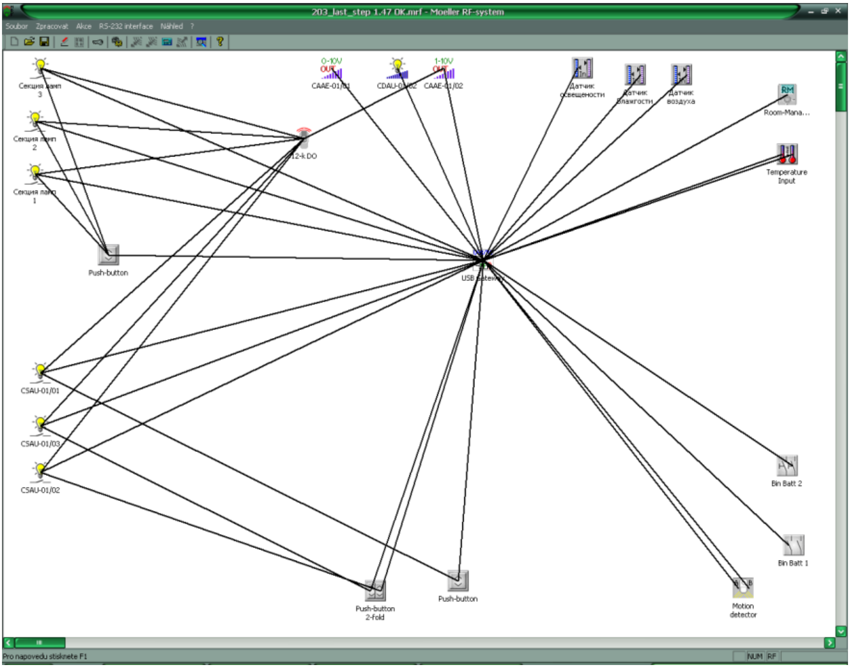


Fig. 3 – The scheme of wireless communications between the elements of the system

All other elements are included in the configuration with the wireless network, and can function independently of the entire system.

The HL1 lamp is activated using the actuator CSAU-01/01. The actuators are the actuating elements that lock and unlock the contacts when the signal is supplied from the switch, or another source of the control signal. The CSAU-01/01 is conFig. d to operate when the control signal is displayed from the CTAA-01/02 switch, the button number 1 in the Room-Manager panel and the button number 1 on the remote control. When the operator (user) activates one or another controller, he sends a radio signal to the actuator CSAU-01/01, which supplies the power to the HL1 lamp. It should be noted that each EIB network device is identified by a unique address, making it impossible to execute commands.

The activation of the HL2 lamp is from the button # 1 of the switch CTAA-02/02, the button # 2 in the Room Manager panel and the button # 2 on the remote control.

The activation of the HL3 lamp is from the button number 2 of the switch CTAA-02/02, the button number 3 on the Room Manager panel and the button number 3 on the remote control.

Homeputer system is used to visualize control over lighting devices and to centralize management. The system is implemented using a personal computer equipped with the USB / RS232 communication interface and Homeputer Studio software.

1. Installing wireless communication between the elements via the USB / RS232 interface and the Moeller RF-System software.

1.1. Connect USB / RS232 communication interface CKOZ-00/03 (CKOZ-00/11) to PC via COM port. Install the Moeller RF-System software and familiarize yourself with the interface and the purpose of the program commands.

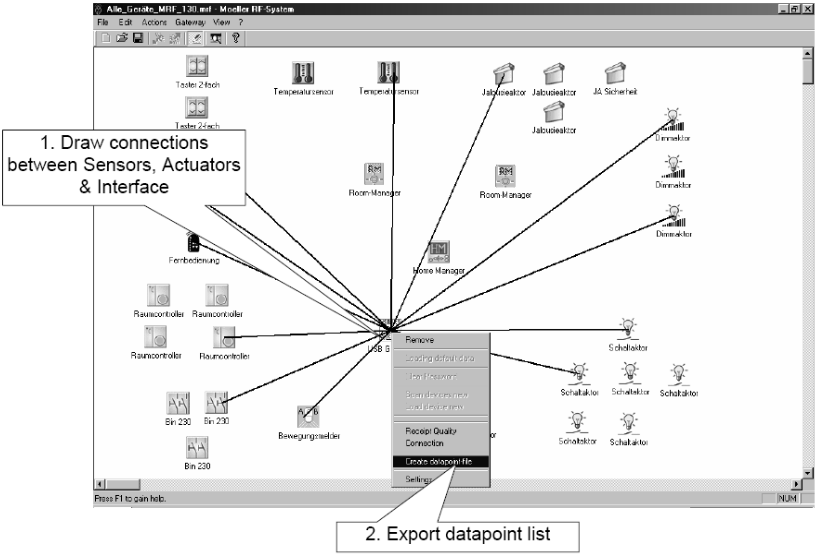
1.2. Use the Scan command to establish contact with all the elements of the stand, assign each item a name that matches their purpose.

1.3. Turn on and off turn-by-turn actuators using Moeller RF-System using the properties of the On and Off elements.

1.4. Use the Connection Mode and Load commands to import the configuration to the items. Clicking on the icon will download the information on the device.

1.5. Use the Connection Mode and Load commands to create a wireless communication system. Check how the system works.

1.6. Using the Create Datapoint-file property of the USB / RS232 Communication Interface, prepare the data for running in the program for controlling and controlling Homeputer Studio (Fig. 4).



1. Draw connections between Sensors, Actuators & Interface

2. Export datapoint list

Fig. 4 – Preparing Datapoint-file

1.7. Using the prepared Datapoint-file, import the configuration of the elements into Homeputer Studio.

Training 3
Development of the Macro program management system in Homeputer Studio

The software of system must solve the following tasks:

1. Support for the temperature of the inflow air and / or the temperature in the room with the processing of signals of temperature sensors in subprograms of PI regulators or relay regulators with a hysteresis loop;

2. Electric heater control with the help of the simistronic electronic relay by the method of pulse-width modulation with a period of 1 minute;

3. Control of water heat exchanger with three-position valve with electric drive;

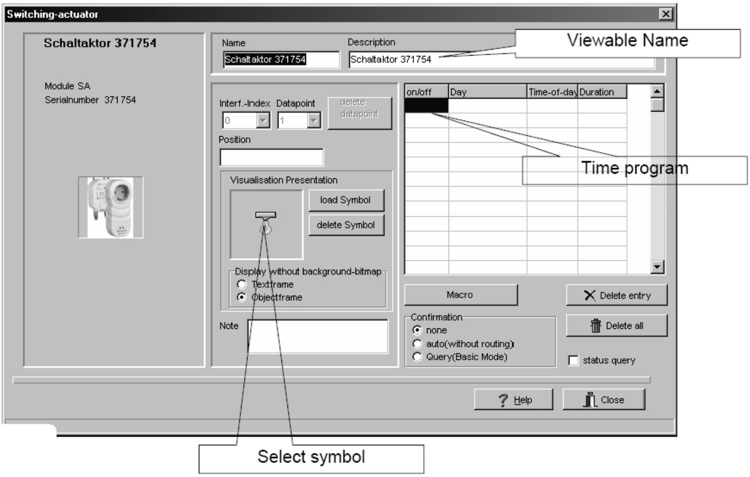
4. Control of two- or three-position flaps in air channels;

5. Speed control of fans, including smooth overclocking when switching on and a smooth stop when switching off the system;

6. Protection of the electric heater from overheating and water from icing, control of air filters, shutdown of the system when triggered by a fire alarm.

To create a virtual model, you need to install the Homeputer Studio program on your computer and get familiar with its interface. The essential condition of the program's operation is the connected USB / RS232 communication interface.

You can double-click the settings menu for each device, where you can set the name, the icon and select the Device's working time (fig. 5).



Switching-actuator

Schaltaktor 371754

Module SA
Serialnumber 371754

Name: Schaltaktor 371754

Description: Schaltaktor 371754

Interf.-Index: 0

Datapoint: 1

Display without background-bitmap: ○ Textframe ● Objectframe

Confirmation: ● none ○ auto(without routing) ○ Query(Basic Mode)

□ status query

on/off	Day	Time-of-day	Duration

Fig. 5 – Device settings menu

An example of a programming procedure is provided for the next task.

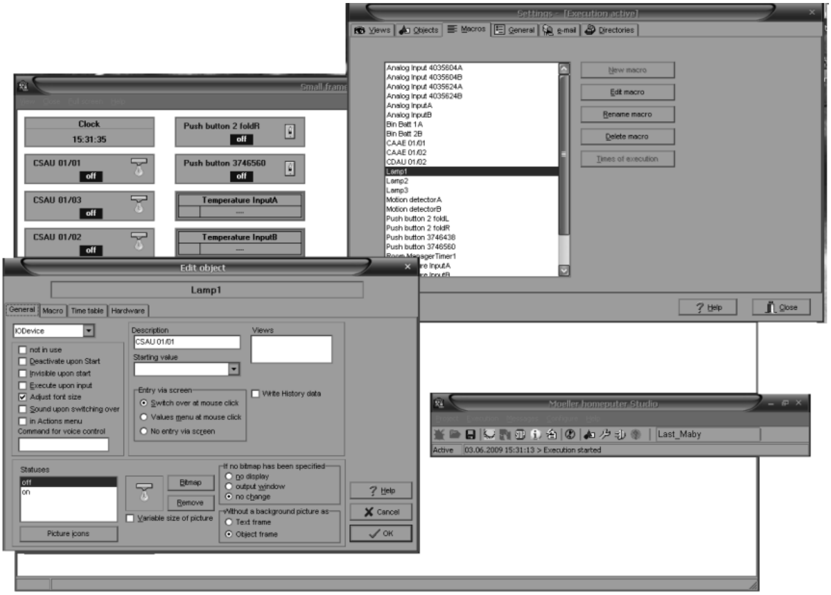
Turning on the light when opening the door.

When the door is opened, the light comes on, when they are closed, the light goes out after 30 seconds. This scenario can be used for lighting in the entrance area.

Elements,which are used to implement the script: a window (door) sensor with an appropriate actuator, actuator CSAU-01/01 or other element of this type, a lamp.

To create a macro, you need to click on the icon on the Homeputer Studio main panel and enter the Settings menu, select the Macros tab. The text of the program is written in the menu of the element, which it concerns directly.

Then in the General tab you can specify the initial conditions of the element, its graphical display and mode of operation (Fig. 6).



Edit object

Lamp1

General | Macro | Time table | Hardware

IODevice

□ not in use
□ Deactivate upon Start
□ Invisible upon start
□ Execute upon input
☑ Adjust font size
□ Sound upon switching over
□ in Actions menu

Command for voice control

Description: CSAU 01/01

Starting value

Views

Entry via screen
● Switch over at mouse click
○ Values menu at mouse click
○ No entry via screen

□ Write History data

Statuses: off, on

□ Variable size of picture

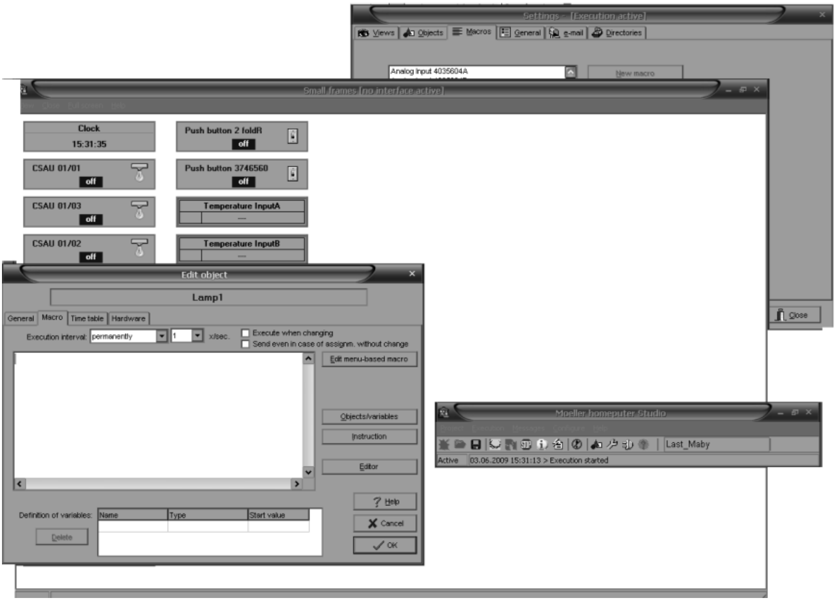
If no bitmap has been specified
○ no display
○ output window
● no change

Without a background picture as
○ Text frame
● Object frame

Active 03.06.2009 15:31:13 > Execution started

Fig. 6 - Configuring the initial values

The Macro tab enters the program text. The Instruction command allows you to select operators to write the program. The Objects command lets you select the items that are mentioned in the program (Fig. 7).



Analog Input 4035604A

Lamp1

Execution interval: permanently 1 x/sec.

□ Execute when changing

□ Send even in case of assignm. without change

Definition of variables:

Name	Type	Start value

Active 03.06.2009 15:31:13 > Execution started

Fig. 7 – Creating a macro

The macro is written in a language close to BASIC, and for the given task its text is given below.

if Bin_Batt_1A switchedon
then
Lamp3 switchon
end-of-if-block
	If the door is open, the actuator is switched on

if Bin_Batt_1A switchedoff
then
Lamp3 switchoff for 30 seconds
end-of-if-block
	If the door is closed, then the actuator is switched off for 30 seconds

When the program is ready, you need to save the changes and execute the Run command.

4. Individual tasks

Select devices from Table 1 and prepare a protocol on the system configuration, setup procedures and macros (if needed) for each task.

4.1 Scenarios of the lighting control system

1. Passage corridors and rooms with multiple entrances

1.1. The problem is that every person who enters and goes out of the room changes the position of the switch at the entrance. Lighting, respectively, must be switched on or off.

1.2. In order to implement the scenario, you must have either pass-through switches (for corridors) or conventional two-way switches and a program in a controller that will implement the Exclusive OR function for many logic signals.

2. Maintaining lighting at the required level.

2.1. This scenario can be used in rooms where it is necessary to maintain a constant level of illumination regardless of the time of day when the level of natural light changes, for example, in the office.

2.2. Elements used to implement the scenario: a light sensor with an appropriate actuator: a symmetric voltage regulator and a halogen lamp or 12, 24 V power supply and a PWM for an LED lamp.

3. Maintaining natural lighting at the required level.

3.1. This scenario can be used in rooms where a constant level of natural light is required irrespective of the time of daylight.

3.2. Elements used to implement the scenario: a light sensor with an appropriate actuator: actuator blind, drive motor.

4. Turning the light on when moving in the room.

4.1. This scenario can be used either as a security alarm, or in long corridors, where the movement is periodic.

4.2. Elements used to implement the script: motion sensor with corresponding executive lighting device.

5. In long corridors, when triggering the motion sensor, several light bulbs must be fired successively in intervals of 3-10 seconds. If after the triggering movement is absent, the lamps should go out in reverse order with the same interval.

5.1. Elements used to implement the scenario: motion sensor with corresponding actuators: relay actuators (for example, CSAU-01/01) and lamps.

6. Advanced light level adjustment scenario due to natural and artificial lighting. Two factors must be taken into account: the actual light level and time of day. Thus, in the light of day, the daylight is controlled by the shutter, in the dark by adjusting the voltage on the incandescent lamps. This scenario will save you a lot of energy.

6.1. Elements used to implement the scenario: light sensor, simulator voltage regulator, halogen lamp, actuator blind, drive motor.

7. Switch on the devices and lighting according to the schedule of work in the room (for example, in the halls of the shopping centers).

7.1. To implement the scenario, an astronomical timer with scheduled work schedules on weekdays and weekends is required, hall layout on the zones, actuators for individual lighting zones.

4.2 Scenarios for controlling climatic installations and the ventilation system

To implement them, humidity sensors, temperature, air quality and analogue actuators that output the task signal to the corresponding devices, astronomical timers are used to schedule the desired temperature at the time of day or season.

8. Support for the required indoor temperature by adjusting the heating performance. This scenario can be used to maintain a comfortable temperature under the influence of various factors: the periodic opening of doors and windows, the heating of the room as a result of the work of the plate or electrical appliances.

9. Support for the required indoor air quality by adjusting the ventilation efficiency. This scenario can be used to maintain air purity with a variety of factors: cooking, increasing the number of people in the room, air pollution by other outsider, the efficiency of the ventilation system may decrease, depending on the presence of people, time of day.

Elements used to implement the scenario: an air quality sensor with an appropriate actuator: an analog actuator, a ventilation control system, a presence sensor, and an astronomical timer can be added.

4.3 Scenarios of the Security subsystem

Organize interaction with lighting, access control and protection subsystems.

Security systems are usually separate systems, equipped with specialized controllers and sensors with a separate network of communications with a security company. But within the frame of Smart House functions several scenarios can be implemented, which work as a supplement to the main security system.

10.1. Switching on the lighting in the room and near the private house, shutting down the roller if the security system is on, but the motion sensor or window sensor has been triggered or the ALARM button is pressed. This feature allows you to frighten an alleged offender.

10.2. Send a message to the owner of the room when triggering movement sensors or window sensors in the mode when the security system or the "No one is" mode is activated. To implement this scenario, you need to conFig. the connection channel of the controller with the Internet or with the mobile operator.

10.3. The mode of presence simulation is implemented to prevent criminal intent in the absence of tenants in the house. A system that remembers the sequence of switching on / off the lighting devices by tenants for a long time (week, month) is considered an effective one. When you click on the "Simulate Presence" button, this sequence is repeated according to the astronomical time and day of the week.

5 Control Questions

1. What are the main methods of communication using EIB / KNX on the physical level?
2. How does the Moeller RF-System (EIB) radio network work?
3. How is the problem of collisions in the serial EIB / KNX tires resolved?
4. What are the main features of the Carrier Sense Multiple Access / Collision Avoidance protocol?
5. What are the main types of devices (which element base) which are used in SMART HOUSE with IoT?
6. What is the system addressing devices in the EIB network?
7. How does the connection and breakdown of communication with the actuator or sensor on the EIB radio network?

8. Which basic parameters of air provide comfortable conditions?
9. What is the principle of operation of the air damper drive?
10. What is the principle of operating a water valve drive?
11. Which sensors are used in climatic systems?
12. How can the fan speed in climatic systems be controlled?
13. How to regulate the power of a water heat exchanger?
14. How can the power of an electric heater be cntrolled?
15. What is the complexity of obtaining a reliable mathematical description of the thermal processes in the room?
16. Which regulators are used in climate control systems?
17. How is energy saving provided during the construction of SMART HOUSE?
18. What is a "macro" and how does macros perform?
19. Why do you think Moeller uses the simplest programming language to prepare macros like BASIC?
20. Which steps are needed to set up the Moeller / Eaton xComfort system?

6 References

1. Ferro, E. (2018). *Smart Solutions for the CNR campus in Pisa*. [online] Openportal.isti.cnr.it. Available at: https://openportal.isti.cnr.it/results?option=com_dnetindexclient&view=doc&id=people______::680a59763c98f06ffc49f48f8abfcc22

2. Ferro, E. and Parodi, O. (2018). *Smart devices and applications for healthy ageing*. [online] Openportal.isti.cnr.it. Available at: https://openportal.isti.cnr.it/results?option=com_dnetindexclient&view=doc&id=people______::cc895aea31a19a92c32bfca2bd7c9746

3. Horyński, M. (2017). Energy management in households as part of the sustainable development of the energy economy. [online] 2017 International Conference on Electromagnetic Devices and Processes in Environment Protection with Seminar Applications of Superconductors (ELMECO & AoS). Available at: https://doi.org/10.1109/ELMECO.2017.8267721

4. Marksteiner, S., Jimenez, V., Valiant, H., Zeiner, H. (2017). An overview of wireless IoT protocol security in the smart

home domain - IEEE Conference Publication. [online] Doi.org. Available at: https://doi.org/10.1109/CTTE.2017.8260940

5. Pang, D., Lu, S. and Zhu, Q. (2018). Design of Intelligent Home Control System Based on KNX/EIB Bus Network. Available at: https://doi.org/10.1109/WCSN.2014.74

6. Park, P., Ergen, S., Fischione, C., Lu, C. and Johansson, K. (2018). *Wireless Network Design for Control Systems : A Survey*. [online] DIVA. Available at: http://kth.diva-portal.org/smash/record.jsf?dswid=3396&pid=diva2:1220077

7. Technical specification EATON RF System. 96 p. Eaton.eu. (2018). [online] Available at: http://www.eaton.eu/ecm/idcplg?IdcService=GET_FILE&allowInterrupt=1&RevisionSelectionMethod=LatestReleased&noSaveAs=0&Rendition=Primary&dDocName=PCT_1572103

8. Vanus, J., Cerny, M. and Koziorek, J. (2015). The proposal of the smart home care solution with KNX components. [online] Available at: https://doi.org/10.1109/TSP.2015.7296410

9. Varghese, S., Kurian, C. and George, V. (2015). A study of communication protocols and wireless networking systems for lighting control application - IEEE Conference Publication. [online] Doi.org. Available at: https://doi.org/10.1109/ICRERA.2015.7418618

4 TECHNOLOGIES OF INTERACTION IN THE SMART BUILDING AND CITY SYSTEMS

Training 1
Authorization in components of the IoT and IoE system with the use of OAuth protocol

1 The purpose and aims of the training

The purpose of this paper is to study the security flaws of the OAuth protocol for IoT and IoE systems. The IoT components in this training are considered by the example of communication modules and lighting control modules on Arduino project collection, which provide some typical of behavioral and synchronize processes of the Smart Building and City systems.

Learning tasks:

- learning the basics of network security, security of HTTP requests and serialization formats;
- study of protocol standards OAuth 1.0 and OAuth 2.0 for authentication of HTTP requests;
- obtaining the knowledge and skills for security, digital signature, and verification of HTTP requests.

Practical tasks:

- implementation of construction of "Signature String", in accordance with the OAuth 1.0 protocol;
- implementation of the signature of the requests and construction of "Authorization" header and in accordance with the OAuth 1.0 protocol;
- simulation of the process using hardware and software;
- acquiring and processing simulation results.

Research tasks:

- research the possibility of changing and verifying a request signed by OAuth using programs such as Fiddler or Charles Proxy;
- analysis and practical evaluation of the OAuth 1.0 protocol.

2 Preparation for the training:

- have an understanding of the purpose and objectives of the training;
- study the theoretical material given here and in additional sources;
- determine the sequence for performing training in accordance with the personal assignment.

2 Theoretical material

The safe interaction of components in IoT smart home and city systems with the access to the REST API [1, 2] of a web service begins with reliable authorization. For a more complete solution to the security problem of third-party applications, OAuth [3, 4] protocol was proposed, the main purpose of which is to perform user input via a standard web browser, as a result, to get an access token that is used for subsequent access to the REST API. The protocol description is given in [3-5], where the advantages and disadvantages of the protocol are noted. In [6] attacks on websites using the technology OAuth were researched. The XAuth authorization protocol guarantees that the data will not be transmitted to the attacker during the exchange with the server. However, there is a risk that a third-party application steals data. The problem is that the client itself can implement malicious functions, such as sending user data to an attacker. The standard authorization sequence using the OAuth protocol can be represented as a UML sequence diagram (Figure 1).

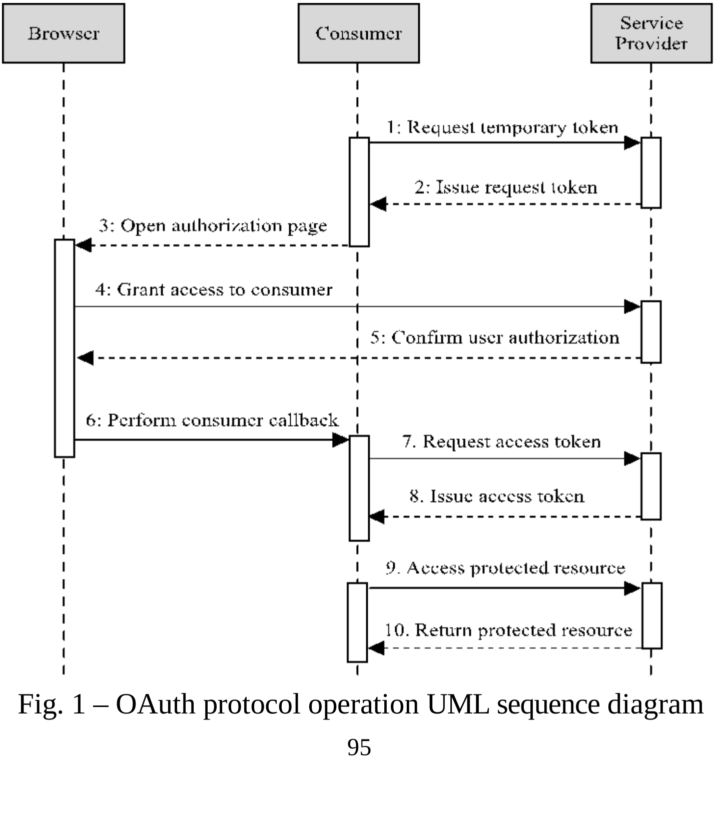

Browser
Consumer
Service Provider
1: Request temporary token
2: Issue request token
3: Open authorization page
4: Grant access to consumer
5: Confirm user authorization
6: Perform consumer callback
7. Request access token
8. Issue access token
9. Access protected resource
10. Return protected resource

Fig. 1 – OAuth protocol operation UML sequence diagram

The paragraphs 3–6 of the diagram imply the need for a web browser to use the OAuth protocol.

The sequence of the OAuth protocol includes the following steps:

- consumer requests access token from server
- then the consumer is redirected to the login page
- consumer enters and is redirected back with access token
- consumer receives an access token, and requests an OAuth token that will be used in the future to provide secure REST API.

In [7], a server-side, easily modified and integrable OAuth library was introduced, supporting the OAuth 1.0a and XAuth protocols. However, during the research of the OAuth protocol specification, a flaw was found in the requirements. The requirement tells us that content must be strictly only URL encoded, or in the other case, it won't be verified. An important part of OAuth 1.0a is implementation of the request signature. According to the OAuth specification [3], the signature can be implemented by three algorithms: RSA-SHA1, HMAC-SHA1 and plain text. The latter option is not secure since the keys can be intercepted by an attacker. Using SSL 2.0 encryption and server certificate verification on the client side is the best, since less other options consume computing resources. Analysis of existing solutions of OAuth server implementations showed that for HTTP POST requests the request body is not used when computing the OAuth signature. This vulnerability allows content to be changed during data transfer. Many third-party libraries do not provide source code, and many of them have licensing problems for commercial use.

The main advantage of OAuth [8] is that the username and password are sent to the server from a web browser window and are not exposed to third party applications. To enter a username and password, a web browser window or a native mobile application is used to ensure that third-party developers do not have unauthorized access to user credentials, especially when the web service provides REST-API to third-party developers. For trusted applications, the web service developer may allow using the XAuth protocol, which allows to provide the username and password directly in order to authenticate the user, and this will be much easier and secure for application developers. But in this case, there is the potential that a third-party application developer will collect user credentials.

This gives the user more reason to trust the application, since the user can be sure that his credentials will not be stolen from the web browser window using third-party applications. In modern mobile applications, the developer can use native mobile applications to get an access token instead of a browser window. In this case, the application that uses the Facebook API first tries to open the official Facebook application, and in its absence will open a browser window.

The responsibility of target applications that use XAuth depends on how much web-service developers trust third-party application developers. After successful authentication in a web browser, a third-party application receives an access token to be used for REST API requests. If the user is already logged in to the web service, he does not need to enter his credentials a second time in order to use the application. The need for authorization depends on the lifetime of the access token. If the access token expires soon, the application updates this token to continue working with the REST API. Authorization is required if the user does not use the application for a very long time. For developing OAuth secure web services, a two- legged or three-legged approach is usually used. The main difference between them is that in a two-legged implementation, only the application is authorized. If the user wants to access the data using Twitter, he works with a three-legged implementation since the access token must be requested for the user in the application by a third party, instead of the token for Twitter.

Issuing OAuth tokens. The first step receiving a valid request token, which is required to authorize the application on the server side.

To receive "Request Token", application validation is required. After successful validation, the new "Request Token" is issued to the user and stored in a temporary database on the server. The third-party application has to open the login page after receiving the "Request token". After successfully obtaining the Request token from the server, the application displays a web page with authorization using the login and password. If the login and password are successfully entered on the web page, then "OAuth verifier" is issued.

After the OAuth verifier is received, which shall be exchanged for the "OAuth token" and the "OAuth secret" that identify and authorize the user. After successfully receiving the response, the user can perform requests to the web service. On the server side there will be an entry in the database that binds the user and the "OAuth token" and the "OAuth secret". In the case of authorization by XAuth, instead of the "Request token|, an "OAuth token" is immediately issued after providing login and password in the request. The received "OAuth Token" and "OAuth Secret" should be stored on the application side.

Implementation of verification of request. After the OAuth token is received, every request that is protected by OAuth must be verified on the server side before processing. Algorithm of verification of OAuth Request contains three important steps: 1) the request is checked in the OAuth database of tokens for an item; 2) if the token is valid, a signature string is created with the OAuth token and its secret part from the database; 3) the hash of the signature string with the token from the database is compared with the signature in the request, and if it does not match, an authorization error is sent to the client.

Implementation of request signature check on the OAuth server. A particularly important part of the Oauth implementation 1.0a is the request signature. Signature OAuth is usually implemented with three algorithms: RSA SHA1, HMAC SHA1 and plain text. The latter option is not secure, as an attacker can capture the keys. However, when using SSL 2.0 encryption and verification of certificates on the client side, from the point of view of performance, this option is the best, because it takes fewer computing resources. To compute the OAuth signature [6] of requests, a string is constructed from the parameters of the "Authorization" header and the URL-encoded parameters of the request.

The parts of HTTP request which are used in OAuth signature are shown in Fig. 3.

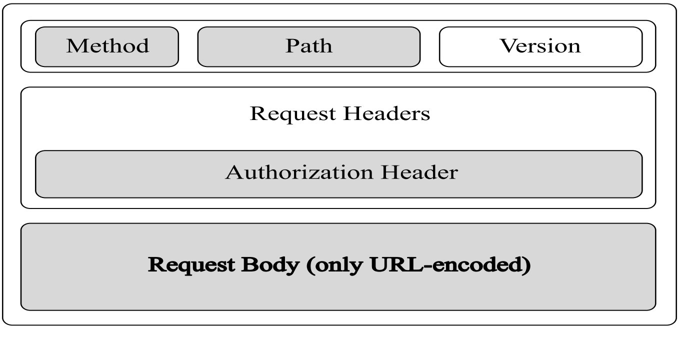


Fig. 3 – The parts of HTTP request which are used in OAuth signature

A limitation of the OAuth protocol is the impossibility of checking the integrity of POST requests, whose entity data format is different from URL-encoded. When using other formats in the request body, verification is not performed. This vulnerability could be exploited by an attacker during a man-in-the-middle attack, which is possible in case, when the data is not encrypted. This vulnerability is also present in the OAuth 2.0 protocol, issuing only temporary tokens. An "man-in-the-middle" attack can also be made here if third-party SSL certificates are allowed to be installed in client's HTTP libraries [9, 10]. A modification of the OAuth protocol [11] is offered, which proposes a verification of the HTTP content [12] of the request.

4 The steps of the training

Step 1. The environment is being prepared for the training. There HTTP server source code and Jar executable are provided. The IoT component of the Smart Building system are being represented by one of the communication or lighting control (Fig. 4) modules from Arduino project collection [13].

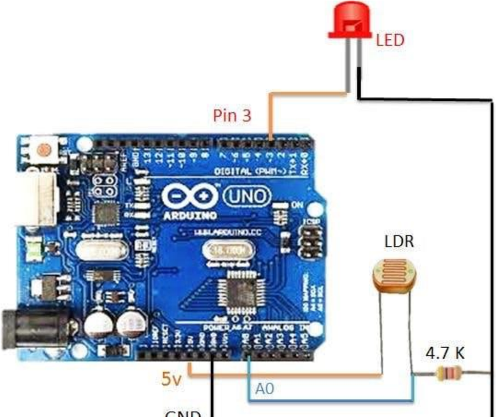


Fig. 4 – Arduino lighting control component

The HTTP server runs on port 9010 by default and can be specified with the --port directive. It is understood that the server can be started, both on the teacher's computer and on the computer that performs the training. To qualify for work, a student must successfully demonstrate work with a teacher's computer. All relative paths below are the concatenation of the base path and relative path strings. To perform training, it is necessary:

- select a programming language and library that implements the HTTP request, prepare the environment;
- implementation of sending the basic HTTP request to the test server using the path "/test" and receiving the "OK" response.

Step 2. The implementation of OAuth 1 signature mechanism is performed. In the training, it is assumed that "OAuth Token" and "OAuth Secret" have already been received by the client and the student needs to implement only the request signature mechanism that signs HTTP requests with text and binary information.

- implementation of the Base String generation, in accordance with the OAuth 1.0 protocol;
- implementation of the construction of the "Authorization" header and the implementation of the signing of requests in accordance with the OAuth 1.0 protocol;
- implementation of sending three HTTP POST requests with following data formats: URL-encoded, JSON, and binary. All three requests must use the path "/oauth1original".

Step 3. Research of OAuth implementation. For this, the following steps are performed:

- investigation of the possibility of modifying a request signed by OAuth using programs like Fiddler or Charles Proxy to a test server;
- investigation of the request modification which is in JSON or binary format for a known service like Twitter, protected by OAuth 1.0a;
- analysis and practical assessment of the OAuth 1.0 implementation approach.

5 Requirements for report content:

The report must include:

- purpose, individual assignment and training research on the security of the OAuth protocol and its modifications;
- selected programming language and HTTP library with which the training is performed;
- the source code of the program and the name of the HTTP library with which the training is performed;
- implementation of the basic request to the server;
- implementation of sending HTTP POST requests signed by OAuth 1 for requests in URL-encoded, JSON, binary formats;
- implementation of construction of "Signature String" for OAuth 1;
- report of each research from step 3;
- conclusions on the results of the research.

6 Test questions and tasks

1. Where and for what are REST APIs used? What modern serialization formats are used?
2. What problem are the proposed training studies?
3. What problems does OAuth solve?
4. What is the difference between OAuth 1.0 and OAuth 2.0?
5. What vulnerability exists in the OAuth 1.0 protocol?
6. What is the proposed solution to overcome vulnerabilities in OAuth 1.0?
7. Can the user modify the data if a developer uses the HTTPs default settings in modern mobile platforms? What does the user need to do?
8. What are the benefits of your chosen programming language and HTTP library which you have chosen during training?

7 References

1. Richardson Leonard, a.R.S., RESTfull Web Services Web services for the real world. O'Reilly Media.
2. Webber Jim, P.S., Robinson Ian, “REST in Practice Hypermedia and Systems Architecture,” O'Reilly Media, 2012.
3. OAuth 1.0 Protocol. IETF RFC 5849 https://tools.ietf.org/html/rfc5849.
4. Nascimento, A.E., OAuth 2.0 Cookbock.
5. Media, O.R., OAuth 2.0: The Definitive Guide
6. Jim Basney, J.G., An OAuth, “Service for Issuing Certificates to Science Gateways for TeraGrid Users,” 2015.
7. Surkov S.S., Martynyuk O.M., Mileiko I.G., “Modification of Open Authorization Protocol for Verification of Request,” Electrotechnic and Computer systems, 2015, no. 19 (95).
8. Hammer, E., OAuth 2.0 and the Road to Hell, 2012.
9. The Transport Layer Security (TLS) Protocol. – IETF RFC 5246 2014; Available from: http://tools.ietf.org/html/rfc5246.
10. HTTP over TLS, IETF RFC 2818 (Informational). 2011; Available from: http://tools.ietf.org/html/rfc2818.
11. Programming Social Applications: Building Viral Experiences with OpenSocial, OAuth, OpenID.
12. Fielding R., a.R.J. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, IETF RFC 7230, 2015; Available from: https://tools.ietf.org/html/rfc7230 (accessed 23.06.2016).
13. A collection of Arduino projects. Join GitHub today, 2019. Available from: https://github.com/mattiasjahnke/arduino-projects; https://create.arduino.cc/projecthub/SURYATEJA/automatic-street-light-controller-27159f

Training 2
HMAC for real-time communication of components of IoT systems

1 The purpose and aims of the training

The aim of the work is to increase the security of communication in Smart Home and Sity systems by signing requests through the HMAC. The IoT components in this training are considered by the example of communication modules and lighting control modules on Arduino project collection, which provide some typical processes of smart systems.

Learning tasks:

– learning the basics of network security, technical capabilities of protocols used in IoT and IoE including Smart Home and Sity systems such as HTTP[1], HTTP/2 [2], MQTT [3], WebSocket [4];

– learning the structure of HMAC;

– learning the protocol standards OAuth 1.0 [5, 6], Hawk [7] and their mechanisms to ensure integrity, security and verification of requests, as well as customer identification;

– learning the features of data serialization formats of HMAC [8];

– the acquisition of knowledge and skills for security, digital signatures and message verification.

Practical tasks:

– implementation of the signature for a request;

– simulation of the process using hardware and software;

– acquiring and processing simulation results.

Research tasks:

– research of your implementation for the vulnerability of changes in the content in the process of transmission using programs such as Charles, Fiddler

– analysis of the vulnerability of changing the content during its communication if it is not part of the digital signature;

– analysis and practical assessment of the implementation

2 Preparation for the training

– have an understanding of the training purpose and objectives;

– study the theoretical material given here and in additional sources;

– determine the sequence for performing training in accordance with the personal assignment.

3 Theoretical material

Currently, IoT systems use protocols such as request / response pattern and real-time protocols publish / subscribe pattern. From the protocols that work in the request / response pattern, the HTTP 1/1 protocol is the most common. In it, the data is transmitted via the path, headers and the request body itself. There are three sections in the request and in the response: starting line, headers, and body (Figure 1).

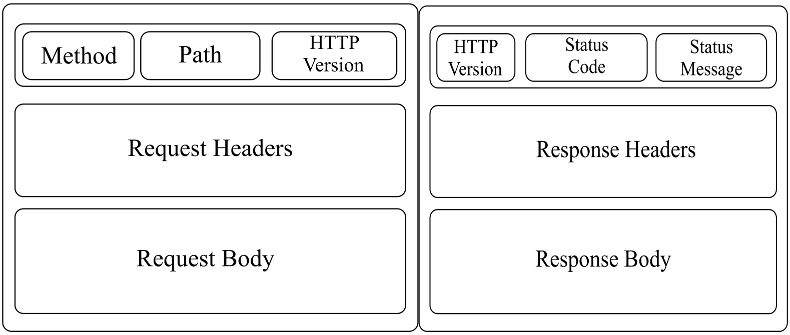


Fig. 1 – Model of HTTP request and HTTP response

The main disadvantage of the "request-response" protocols is that it is impossible to receive updates from the server. For example, for a device whose purpose is to report a change in temperature, implementation of this pattern would imply querying the current state with the smallest possible update interval. This drawback no more exists in the publisher-subscriber pattern, which implies that the device itself will provide updates over the current communication channel.

The easiest option to implement "publisher-subscriber" is the Websocket protocol [4], which provides bidirectional communication between the parties, by wrapping the transmitted and received data in frames. In essence, web sockets are a transport protocol that normalizes the transmission and reception of raw data using the TCP protocol.

The solution of the problem of which response belongs to which query or subscription must be implemented by the protocol developer. This protocol refers to the simple implementation.

However, it requires the implementation of additional functionality, which is the main drawback for its use in IoT systems.

The frame structure of the WebSocket is shown in Fig. 2.

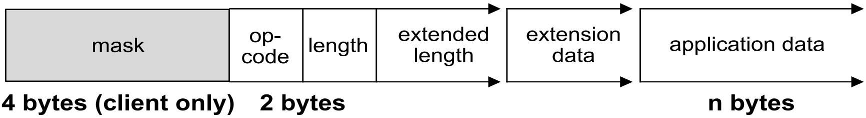


Fig. 2 – Websocket frame structure

A common protocol that implements the publisher-subscriber model is the MQTT [3]. In it, the central element is Broker and customers can post changes by publishing them and receive changes by subscribing to them. Models of the HTTP and HTTP/2 protocols are shown in Fig. 3.

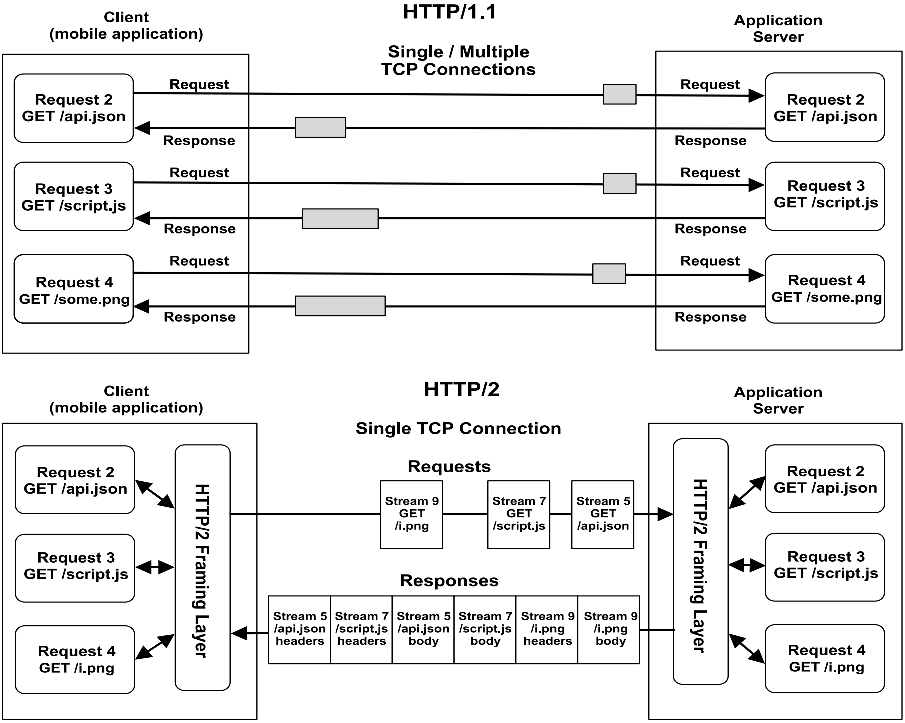


Fig. 3 – HTTP and HTTP / 2 operation model

HMAC structure. Data at the time of transmission through the network is vulnerable to any analysis and attacks. Another case may be at the time of establishing a connection with the server, if a "fake" server is detected on the network. The HMAC mechanism provides a unique digital signature using a hashing algorithm (SHA-1, SHA-256, ...) from known content and secret byte sets that is known to the sending and receiving side. HMAC works in a modern format for detecting forgery of e-mail messages DKIM [9]. HMAC-based mechanism implements HTTP request authorization algorithms, such as OAuth 1.0 [5] and Hawk, which allows a developer to sign HTTP requests, authorizing them and protecting them from modifications.

OAuth and HAWK formats sign most of the HTTP request by collecting data into the Signature String, to which the secret part of the token is added. Particularly important parameters in the OAuth and HAWK digital signature protocols are the *Nonce* and *Timestamp* parameters that determine the uniqueness of the request.

HMAC in IoT for WebSocket, MQTT, HTTP/2

For IoT systems of smart home and sity, the use of OAuth 1.0a / Hawk in its original form is complicated, since these are standards designed for HTTP 1.0/1.1 protocol, which can work by the pattern request / response and do not support server push. In addition, textual representation in HTTP 1 is relatively resource-intensive. To implement the HMAC mechanism, it is necessary to distinguish the Authorization structure, in which unique data for authorization are involved. This structure is similar to the Authorization header in the OAuth protocol.

Serialization of the "Authorization" structure should be done in a format that is more convenient for the sender and receiver to process it. Potentially, the best solution is the serialization of Protobuf format and similar binary serialization formats, as it provides energy efficiency and speed of parsing of the structure.

Message serialization. The developer needs to determine how much data is in content. If the amount of data is large, it is necessary to provide for the possibility of verifying the data using streams. In the case of HTTP/2, the serialized Authorization structure can be base64 encoded and placed in the header (small size). The verification of the content will already occur streaming, as it is provided for by the protocol itself. In the case of using WebSocket or MQTT, to use HMAC, the developer must wrap the message transfer model in a serialization format like Protobuf, or additionally do frame identification for these protocols for "Autorization" or "Content" frame type. To reduce memory consumption, it is necessary to implement a streaming mechanism.

Signing a message. To calculate the message signature on the receiving and transmitting side, the developer needs to get a string or a set of bytes, called "Signature_Base", which includes all fields from the Authorization structure plus a shared secret. To calculate the signature, the signature from "Authorization" structure must be excluded from the resulting byte set. To generate a digital signature in all programming languages, a sequence of operations with the MessageDigest object is used.

Signature algorithms are built into modern programming languages. For more detailed instructions, see the documentation of your chosen programming language. For HMAC systems, the SHA-1 algorithm is proven to be safe. The SHA-1 collision search in an HMAC package may make the system unstable with a high search cost, but a search for the original message may compromise the system.

4 The steps of the training

Step 1. Preparation of computer environment and software for laboratory work. The choices made at this step significantly affect the complexity and time of the laboratory work. The IoT component of the Smart Building system are being represented by one of the communication or lighting control modules from Arduino project collection [10].

To perform laboratory work, it is necessary:

– select the protocol on the basis of which the communication between the module and the recipient from HTTP / 2, MQTT, Websocket will occur;

– select a programming language, prepare the computer environment;

– select library/libraries/frameworks for the selected protocol, for the selected programming language, implementing the client and server sides;

– implement sending "Hello World" from client to server using the selected library.

Step 2. Development of a message signing algorithm. At this step, the structure and format of message signing is selected, based on the speed of work, energy efficiency and ease of implementation:

– select the serialization format of the Authorization structure;

– select and describe the way the message transfer model is signed by the HMAC;

– describe what size data is planned to be transferred.

Step 3. Message Signature Implementation:

– implement the generation of both "Signature Base" and the message signature in accordance with the section "Signing the Message";

– implement the message signing on the client, in accordance with the selected serialization format and the way the message transfer model is signed by HMAC

– implement verification of the message on the server, in accordance with the chosen serialization format and method of the message transfer model signed by HMAC

– make a report on the communication between the client and the server, in which the requests that have been verified on the server, sent earlier, with the wrong signature should be shown

Step 4. Study the resulting implementation of HMAC including the possibility of modifying a request signed by the HMAC implementation obtained using debugging proxies like Wireshark, Fiddler or Charles Proxy between the client and the server.

5 Requirements for report content

The report has to include:

– purpose, individual assignment and laboratory research program;

– selected programming language, communication protocol, library for working with the protocol;

– serialization format of the Authorization structure, transmission format of the message signed by the HMAC, a description of what type and size of data will be transmitted through it;

– the source code of the program in the selected programming language, working with the selected communication protocol using the selected library performing: a) client sending the string "Hello World" to the server; b) server accepting the string "Hello World" from the client; c) signing a message using the selected protocol; d) client which sends to the server "Hello World" string signed by the created protocol based on HMAC; e) server which receives a request from the client with "Hello World" string or any other data that verifies the digital signature of the request from the client;

– conclusions on the results of the research.

6 Test questions and tasks

1. What modern data transfer formats are used in IoT?
2. What modern serialization formats are used for HMAC?
3. What problem are the proposed laboratory studies?
4. To solve what problem is the HMAC protocol created?
5. What protocols are based on HMAC?
6. Why can't HTTP be fully used for IoT?
7. What data transfer data are not protected by a digital signature? What vulnerabilities can this cause?
8. Can the user modify the data if the developer uses the default TLS settings in modern mobile platforms?
9. What data does your implementation of the HMAC protocol verify?
10. What are the advantages of your chosen programming language and data transfer protocol?

7 References

1. Fielding R. Reschke J. *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing, IETF RFC 7230,* 2015; Available from: https://tools.ietf.org/html/rfc7230 (accessed 23.06.2016).
2. M. Belshe, BitGo, and R. Peon. *Hypertext Transfer Protocol Version 2 (HTTP/2), IETF RFC 7540,* 2015; Available from: https://tools.ietf.org/html/rfc7540 (accessed 2.19.2019).
3. Gastón C. Hillar, *MQTT Essentials - A Lightweight IoT Protocol*. April 2017: Packt Publishing.
4. Lombardi, A., *WebSocket Lightweight Client-Server Communications*. 2016: O'Reilly Media.
5. E. Hammer-Lahav, *The OAuth 1.0 Protocol. IETF RFC 5849 (Informational).* (2015).
6. LeBlanc J, *Programming Social Applications: Building Viral Experiences with OpenSocial, OAuth, OpenID*. 2017.
7. Hammer Eran. *HAWK / HTTP Holder-Of-Key Authentication Scheme*. 2019; Available from: https://github.com/hueniverse/hawk.
8. H. Krawczyk, M. Bellare. *HMAC: Keyed-Hashing for Message Authentication*. 1997; Available from: https://tools.ietf.org/html/rfc2104.
9. D. Crocker Ed., et al. *DomainKeys Identified Mail Signatures RFC 6376*. 2011; Available from: https://tools.ietf.org/html/rfc6376.
10. A collection of Arduino projects. Join GitHub today, 2019. Available from: https://github.com/mattiasjahnke/arduino-projects

Training 3
Providing an additional identification level of FPGA-based components in IoT systems of Smart Building and Sity

1 The purpose and aims of the training

The purpose of the training is to research approaches to provide an additional level of protection for FPGA-based components of IoT systems in an aggressive information environment by embedding hidden identifiers in the form of digital watermarks. The IoT components in this training are considered by the example of communication modules and actuator control modules, which provide some typical processes of Smart Building and City systems.

Educational objective: introducing to the methods of embedding a digital watermark (which contains a hidden identifier) into the information object of the program code of FPGA-based components of IoT systems;

Practical objectives: in the special software environment, preparation of a digital watermark containing a hidden identifier; automated analysis of the information object of the FPGA-based component program code and obtaining the information model of the LUT-circuit; implementation (in the Intel (Altera) Quartus environment) of embedding digital watermark bits into the FPGA-based program code information object.

2 Theoretical material

Modern digital computing and control systems (including IoT systems) are built on the basis of both specialized and programmable integrated circuits (ICs). Programmable integrated circuits are usually represented by: a) microprocessors and microcontrollers; b) programmable logic integrated circuits. Currently, the most often used type of programmable logic integrated circuits are FPGA (Field Programmable Gate Array) [1]. These integrated circuits are a set of programmable calculating units, which are arranged in the form of a two-dimensional matrix. The functions of the blocks and their connections are specified by the program code, which is located in the configuration memory FPGA. The modification of configuration memory content leads to the modification of FPGA functioning, i.e. to its reprogramming.

The FPGA chip is formed by a set of elementary programmable units for various purposes: calculating units, memory units, multipliers, input and output units, etc. The most mass in the structure of FPGA are calculating units LUT (Look-Up Table). In a typical FPGA chip, the number of LUT blocks can be from tens of thousands to over a million pieces [2]. Each of the LUT units calculating one programmable logic function of *n* variables (*n* is usually from 4 to 6). The function of the LUT unit is programmed by a 2^n-bit binary program code.

Identification of network components is key value to the security of IoT solutions [3]. In an aggressive information environment, IoT components identification is performed at several levels of abstraction using appropriate network protocols [4]. One promising approach to providing an additional level of identification is the concept of hiding an identifier as a digital watermark in an information object of a network component (node) of an IoT system.

Digital watermark [5] is data that is embedded in an information object to control its use. The technology of digital watermarks is based on the use of steganographic techniques. These techniques hide the fact that the information object (container) contains additional information [6]. Wherein the digital watermark can be read from the container in the presence of a stego-key. Stego-key sets the rules for access to elements of secret information that is embedded in the information object.

For FPGA-based components of IoT systems, it is possible to embedding a hidden identifier in the form of a digital watermark into the information object of the program code of such components. The embedding [7] of a digital watermark is carried out due to the system of equivalent transformations [8], [9] of program codes of a pair of series-connected LUT units.

3 The steps of the training

In the main part of the training, the embedding of a hidden identifier in the program code of an FPGA-based component and the extraction (reading) of this identifier are performed. The following target FPGA components are used in the training: a) actuator control modules Intel Motor Control IP Suite Components for Drive-on-Chip Reference Designs [10] and communication modules Intel Triple Speed Ethernet MegaCore Function [11] shown in Fig. 1. The practical part of the training consists of the next steps.

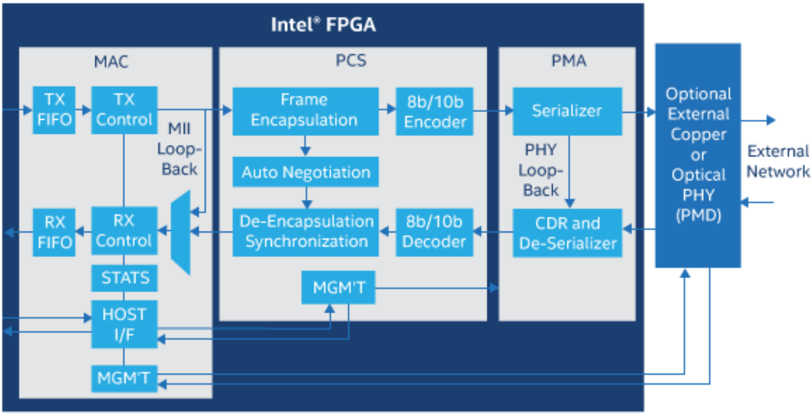


Fig. 1 – Triple-Speed Ethernet IP in an Intel FPGA

Step 1. A binary identifier is formed, intended to be embedding into the program code of an FPGA-based device. The number of identifier bits is selected based on the number of devices (FPGA-based components) included in the system. This takes into account the conditions of the external information environment. For this training, the number of bits of the binary identifier is assumed to be 16.

Step 2. A stego-key is created that is required for the embedding and extraction of the hidden identifier. In the context of this training, it is proposed to use a combination of three parameters as a stego-key: *adr* – the number (address) of the bit of the program code of the LUT units, which, when an identifier is embedding, will act as the direct storage of the bits of the embedded identifier; *threshold* – the threshold number of LUT units, the inputs of which are connected to the output of the current block; *enumeration-rule* – a rule that specifies the order in which the LUT units used to implement the identifier bits are iterated.

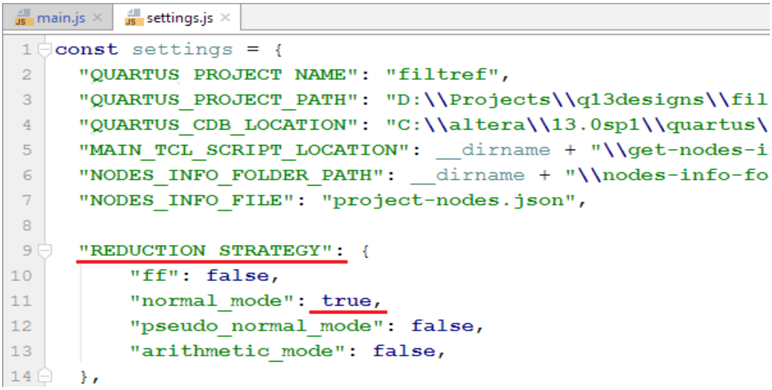
Step 3. The process of setting up the software for automated analysis of the information object of the program code of the FPGA-based component is performed and this analysis is performed.

3.1 In the integrated development environment, it is necessary to open the **LUTIntegrity** software package intended for automated analysis of the information object of the program code of the FPGA based component.

3.2 File settings.js of this package contains the JSON object settings, which is used to configure the analysis tools of the information object of the program code of the FPGA based component.

The value of the QUARTUS_PROJECT_NAME field (Fig. 1) of object settings sets the name of the Quartus project of FPGA device, the information object of whose program code is to be analyzed. The value of the QUARTUS_PROJECT_PATH field of the object settings sets the path to the project.

3.3 The JSON object settings as one of its fields contains the REDUCTION_STRATEGY object (Fig. 2), which is used to set up the selection strategy of target LUT units that are involved in hiding identifier bits. The REDUCTION_STRATEGY object consists of four fields. The boolean values ff, normal_mode, pseudo_normal_mode, arithmetic_mode fields determine whether the target nodes should be included: programmable flip-flops; LUT units in normal mode; LUT units that are in normal mode, but are finite units of a chain of units connected by arithmetic mode transfers; LUT units are in arithmetic mode.



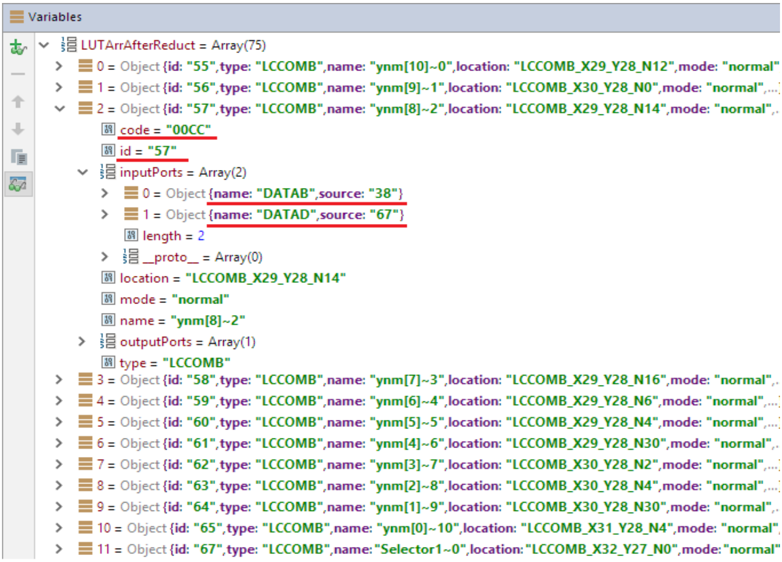
```
const settings = {
  "QUARTUS PROJECT NAME": "filtref",
  "QUARTUS_PROJECT_PATH": "D:\\Projects\\q13designs\\fil
  "QUARTUS_CDB_LOCATION": "C:\\altera\\13.0sp1\\quartus\
  "MAIN_TCL_SCRIPT_LOCATION": __dirname + "\\get-nodes-i
  "NODES_INFO_FOLDER_PATH": __dirname + "\\nodes-info-fo
  "NODES_INFO_FILE": "project-nodes.json",

  "REDUCTION STRATEGY": {
      "ff": false,
      "normal_mode": true,
      "pseudo_normal_mode": false,
      "arithmetic_mode": false,
  },
```

Fig. 2 – Setting the strategy for selecting target LUT units

3.4 You must launch the main file main.js package **LUTIntegrity**. After that, the Intel (Altera) Quartus CAD system will be automatically launched in the background. Using Quartus CAD from an FPGA-based device project, detailed information about the information object of its program code will be obtained

3.5 After launching the main file, it is possible to view information on the input, output and internal nodes of the information model of an FPGA-based device. In the Inodes section of the information model, there is a list of device input nodes. In the Onodes section of the information model, there is a list of output nodes of the device. In the

LUTArrAfterReduct section (Fig. 3) of the information model, there is a list of internal nodes of the project that were selected from the total set of internal nodes in accordance with the selection strategy defined by the REDUCTION_STRATEGY JSON-object.



```
Variables
LUTArrAfterReduct = Array(75)
  0 = Object {id: "55",type: "LCCOMB",name: "ynm[10]~0",location: "LCCOMB_X29_Y28_N12",mode: "normal"
  1 = Object {id: "56",type: "LCCOMB",name: "ynm[9]~1",location: "LCCOMB_X30_Y28_N0",mode: "normal",...
  2 = Object {id: "57",type: "LCCOMB",name: "ynm[8]~2",location: "LCCOMB_X29_Y28_N14",mode: "normal",.
    code = "00CC"
    id = "57"
    inputPorts = Array(2)
      0 = Object {name: "DATAB",source: "38"}
      1 = Object {name: "DATAD",source: "67"}
      length = 2
      __proto__ = Array(0)
    location = "LCCOMB_X29_Y28_N14"
    mode = "normal"
    name = "ynm[8]~2"
    outputPorts = Array(1)
    type = "LCCOMB"
  3 = Object {id: "58",type: "LCCOMB",name: "ynm[7]~3",location: "LCCOMB_X29_Y28_N16",mode: "normal",.
  4 = Object {id: "59",type: "LCCOMB",name: "ynm[6]~4",location: "LCCOMB_X29_Y28_N6",mode: "normal",...
  5 = Object {id: "60",type: "LCCOMB",name: "ynm[5]~5",location: "LCCOMB_X29_Y28_N4",mode: "normal",...
  6 = Object {id: "61",type: "LCCOMB",name: "ynm[4]~6",location: "LCCOMB_X29_Y28_N30",mode: "normal",.
  7 = Object {id: "62",type: "LCCOMB",name: "ynm[3]~7",location: "LCCOMB_X30_Y28_N2",mode: "normal",...
  8 = Object {id: "63",type: "LCCOMB",name: "ynm[2]~8",location: "LCCOMB_X30_Y28_N4",mode: "normal",...
  9 = Object {id: "64",type: "LCCOMB",name: "ynm[1]~9",location: "LCCOMB_X30_Y28_N30",mode: "normal",.
  10 = Object {id: "65",type: "LCCOMB",name: "ynm[0]~10",location: "LCCOMB_X31_Y28_N4",mode: "normal",
  11 = Object {id: "67",type: "LCCOMB",name:"Selector1~0",location:"LCCOMB_X32_Y27_N0",mode:"normal"
```

Fig. 3 – Window for viewing parameters of LUT units

Each element of the LUTArrAfterReduct list has the following information fields: id – global identifier (number) of the node (LUT unit) in the information model of the device; type – node type (in accordance with the involved node selection strategies, this type is always LCCOMB, which corresponds to the designation of the LUT unit in the information model of the FPGA device); name – the name of the node in the device model; location – coordinates of the location of the node in the matrix of the FPGA device; mode – the operation mode of the node; code – hexadecimal value of the program code of the LUT unit; fields inputPorts and outputPorts, the purpose of which was described above in relation to the input and output nodes.

Step 4. The formation of a set of LUT units of elements is being carried out, of which a stego-path can be built – a path for the embedding of a digital watermark in the LUT-circuit space. Stego-path is an ordered set of target LUT units, in the program code of which the identifier formed at the 1st step of this training is directly embedding.

At the 1st training step, a 16-bit binary identifier was formed, intended to be embedding into the program code of an FPGA-based device. Based on this, the stego-path must consist of at least 16 LUT units. Each of them is intended to store one identifier bit. To obtain a set of stego-path LUT blocks, you must perform the following steps.

4.1 Arrange the numbers (id, unique identifiers) of LUT blocks (see section LUTArrAfterReduct visualization of the information object of the program code of an FPGA-based component) in accordance with the *enumeration-rule*, which is a component of the stego-key.

4.2 Remove the blocks from the obtained ordered list of LUT units, the number of connections to the outputs of which exceeds the threshold value *threshold* (is a component of the stego-key).

4.3 From the resulting list of LUT units, remove blocks whose outputs are directly connected to the output nodes of an FPGA-based device. The resulting ordered list of LUT units is a set of LUT units, suitable for the formation of the stego path of embedding a watermark.

Step 5. The digital watermark is embedding into the program code of the LUT units. For the embedding, it is necessary to modify the values of the program codes of the LUT units located on the stego-path. The calculation of the modified values is as follows.

Let $M = <m_1, m_2, m_3, \dots , m_{16}>$ be the ordered sequence of bits of the digital watermark being implemented (hidden identifier), $L= <l_1, l_2, l_3, \dots , l_{16}>$ be the ordered sequence of LUT units forming the stego-path. To perform the implementation of a digital watermark, successively m_i bits are embedded in the program code of l_i units, where $i = 1 \dots 16$. For each pair $<m_i , l_i>$ the following actions are performed:

– if the value of m_i coincides with the value of the bit of the program code of the LUT l_i at the address *adr*, then no modifications of the program code of the LUT block l_i are required;

– if the value of m_i does not coincide with the value of the bit of the program code of the LUT unit l_i at address *adr*, then the following two actions are performed: 1) bitwise inversion of the program code of the l_i unit; 2) compensating rearrangement of bits of all LUT units whose inputs are connected to the output of the l_i unit.

Step 6. The program codes of the LUT units that were modified in the previous step are modified. Code modification is performed in the Intel (Altera) Quartus II or Intel Quartus Prime CAD system for the project, the analysis of which was carried out starting from step 3 of this training. It is necessary to make two copies of this project: a copy for which modifications will be made at the current training step and the original copy, which is necessary for comparison with the modified one. Modification of program codes is performed by the following actions.

6.1. Open in the CAD environment Intel (Altera) Quartus copy of the project to be modified.

6.2. Launch the Chip Planner module, which allows you to view the locations of project nodes in the FPGA matrix (menu Tools → Chip Planner). In the Chip Planner module window that appears, an FPGA matrix consisting of logic blocks is shown. Logical blocks (set of 16 logical elements LE, which include LUT blocks) are shown in blue.

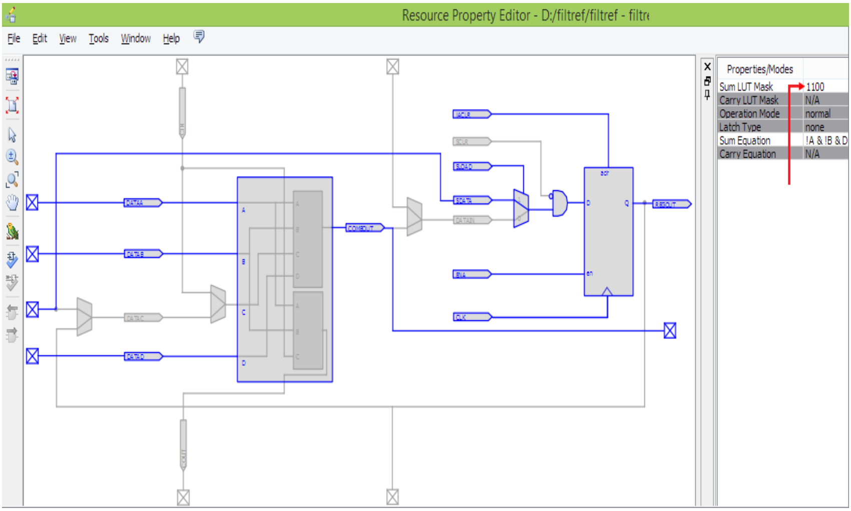
For the matrix defined coordinate system, the beginning of which is located in the lower left corner. Each logical block has two coordinates: the first is the column number of the blocks; the second is the row number of the blocks. The blocks involved in the current project are shown in the Chip Planner window by highlighting in a darker color.

6.3. In the Chip Planner window, you need to find the logical blocks in which the LUT units are located that require modification of the program code. When searching, you should use the X and Y coordinates obtained from the location field of the information model of the LUT block (the list of information model LUTArrAfterReduct, see item 3.5).

6.4. Double click on each of the target logic blocks found to display the logic elements LE (including LUT blocks) that make up the logical blocks.

6.5. In the window that opens, you must select the logical elements (LE), which include the target LUT units. Herewith, you should use the N coordinate obtained from the location field of the information model of the LUT unit (the list of the LUTArrAfterReduct information model, see item 3.5). For each logical element (LE), the coordinates of the LUT blocks are even numbers (0, 2, 4 ...), which increase from top to bottom.

6.6. Next, double-click on the logical element (LE) containing the LUT unit whose program code is to be modified. As a result, the Resource Property Editor module (Fig. 4) is launched, the window of which displays the logic element LE circuit (LUT unit and programmable flip-flop), as well as information on the program code of the LUT unit.



Resource Property Editor - D:/filtref/filtref - filtre
File Edit View Tools Window Help
Properties/Modes

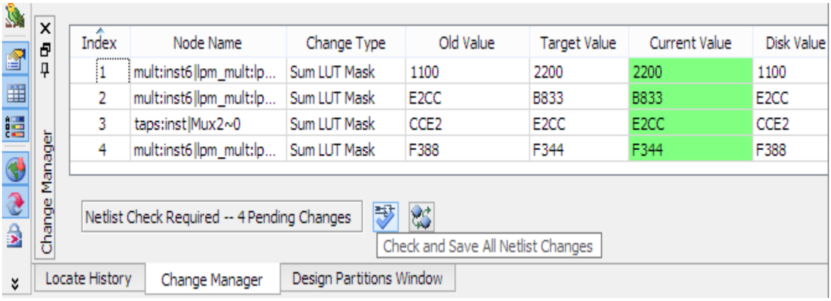
Properties/Modes	
Sum LUT Mask	1100
Carry LUT Mask	N/A
Operation Mode	normal
Latch Type	none
Sum Equation	!A & !B & D
Carry Equation	N/A

Fig. 4 – Window of module Resource Property Editor

6.7. The Sum LUT Mask field of Resource Property Editor contains a hexadecimal representation of the program code of the selected LUT block. This value should be replaced with the value calculated in step 5 of this training.

6.8. As a result of replacing the value of the program code in the Change Manager window of the Chip Planner module, a table appears with information about the source and resulting values of the program codes of the LUT units (Fig. 5).

6.9. The final action of the current training step is to click on the “Check and Save All Netlist Changes” (Fig. 5) button.



Index	Node Name	Change Type	Old Value	Target Value	Current Value	Disk Value
1	mult:inst6|lpm_mult:lp...	Sum LUT Mask	1100	2200	2200	1100
2	mult:inst6|lpm_mult:lp...	Sum LUT Mask	E2CC	B833	B833	E2CC
3	taps:inst|Mux2~0	Sum LUT Mask	CCE2	E2CC	E2CC	CCE2
4	mult:inst6|lpm_mult:lp...	Sum LUT Mask	F388	F344	F344	F388

Netlist Check Required -- 4 Pending Changes

Fig. 5 – Change Manager module

As a result of this, the application of the modifications made to the current project is applied.

Step 7. Perform analysis of the possibility of extracting (reading) a hidden identifier in the presence of a stego-key. To perform this step, it is necessary to build an information model of an FPGA-based device, similarly to how it was done in step 3. The analysis is carried out and conclusions are drawn about the possibility of extracting the identifier hidden in the program code of the device in the presence of a stego-key.

Variants of tasks

All tasks variants require issuing an FPGA component from Intel Motor Control IP Suite Components for Drive-on-Chip Reference Designs [10] and Intel Triple Speed Ethernet MegaCore Function [11].

All tasks variants include:

– introducing to the methods of embedding a digital watermark (which contains a hidden identifier) into the information object of the program code of FPGA-based components of IoT systems;

– in the special software environment, preparation of a digital watermark containing a hidden identifier;

– implementation (in the Intel (Altera) Quartus CAD environment) of embedding digital watermark bits into the FPGA-based program code information object.

4 Test questions and tasks

1. Why should IoT system components be identified?

2. What are the advantages and disadvantages of the open and hidden storage of the identifier of the components of the IoT system?

3. Explain the concept of "digital watermark". What properties should a digital watermark have?

4. What identifiers of components, in what cases, should be used for practical use: open; hidden; combination of open and hidden ids?

5. What parameters are used in the formation of the stego path and stego-key of embedding a digital watermark in an information object of the program code of an FPGA-based device?

6. What Intel (Altera) Quartus CAD tools are used in the process of embedding a digital watermark into an information object of an FPGA-based device program code?

5 References

1. Andina J. FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics / J. Andina. – CRC Press, 2017.

2. Vanderbauwhede W. High-performance computing using FPGAs / W. Vanderbauwhede, K. Benkrid. – New-York: Springer, 2016. – 525 p.

3. Shancang L. Securing the Internet of Things / L. Shancang, Li Da Xu. – Cambridge: Syngress, 2017. – 154 p.

4. Fei Hu (ed.) Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations / Hu Fei (ed.). – Boca Raton: CRC Press, 2016. – 604p.

5. Shih F. Digital Watermarking and Steganography: Fundamentals and Techniques, 2nd edition / F. Shih. – CRC Press, 2017. – 292 p.

6. Fridrich J. Steganography in Digital Media / J. Fridrich. – Cambridge University Press, 2010. – 438 p.

7. Zashcholkin K. The Control Technology of Integrity and Legitimacy of LUT-Oriented Information Object Usage by Self-Recovering Digital Watermark / K. Zashcholkin, O. Ivanova // CEUR Workshop Proceedings. – 2015. – Vol. 1356. – P. 486-497.

8. Drozd A. Use of natural LUT redundancy to improve trustworthiness of FPGA design / A. Drozd, M. Drozd, M. Kuznietsov // CEUR Workshop Proceedings. – 2016. – Vol. 1614. – P. 322-331.

9. Drozd O. Improving of a circuit checkability and trustworthiness of data processing results in LUT-based FPGA components of safety-related systems / O. Drozd, M. Drozd, O. Martynyuk and M. Kuznietsov // CEUR Workshop Proceedings. – 2017. – Vol. 1844. – P. 654-661.

10. Motor Control IP Suite Components for Drive-on-Chip Reference Designs. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/ds_1038_doc_mc.pdf

11. Triple-Speed Ethernet Intel FPGA IP User Guide. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_ethernet.pdf

Training 4
Integrity monitoring of the program code of FPGA-based components in IoT systems of Smart Building and Sity

1 The purpose and aims of the training

The goal of the training is to study the approach to integrity monitoring of the program code of FPGA-based components of IoT systems of Smart Home and Sity through the use of digital watermark technology. The IoT components in this training are considered by the example of communication modules and actuator control modules, which provide some typical processes of Smart Building and City systems.

Educational objective: studying the integrity monitoring method based on embedding a digital watermark into an information object of the program code of FPGA-based components of IoT systems;

Practical objectives: in the environment of special software, preparation of a digital watermark containing integrity monitoring information; implementation in the CAD Intel (Altera) Quartus environment of embedding a digital watermark into an information object of the program code of an FPGA-based component; checking the correctness of the functioning of the integrity monitoring method in cases of absence or presence of distortions of the program code of FPGA-based component.

2 Theoretical material

FPGA chips are widely used as an element base for building IoT system components [1-3]. The functions of the elementary FPGA units and their connections are controlled by a program code located in the configuration memory of the FPGA. Changing the contents of the configuration memory (program code) leads to a change in the behavior of the FPGA chip, that is, to its reprogramming.

This training addresses the problem of ensuring the integrity of the FPGA software code. Most of the known integrity monitoring methods are based on the calculation of the hash sum [4] for an information object whose integrity is monitored. The hash sum is a binary sequence of fixed length, which is obtained as a result of the transformation of an information object (in this case, this information object is the FPGA program code) using one of the special hash functions [5].

The sequence of actions to the integrity monitoring forms two stages: the stage of preparation of the program code for the monitoring and the stage of monitoring itself. At the stage of preparation of the program code for monitoring: a) the hash sum of the program code is calculated; b) the obtained hash sum in some specified way append to the information object of the program code. At the integrity monitoring stage: a) the information object of the program code is separated into its program code and the hash sum; b) a hash sum is calculated for an information object of a program code using a given hash function; c) the newly calculated hash sum is compared with the hash sum that was extracted from the checked object. If these hash sums do not match, it is considered that there is a violation of integrity.

A known approach to monitoring the integrity of the program code, in which the monitoring hash is embedded in the program code FPGA in the form of a digital watermark [6, 7]. The advantages of this approach: a) no additional memory or configuration file fields are required to store the hash; b) information controlling integrity is hidden from an outside observer; c) the fact that integrity monitoring is performed is not obvious to an outside observer.

Using this approach, hash sums are calculated for the entire FPGA program code, and the destination of a digital watermark (including this hash sum) is the program code of a given subset of LUT units. At the same time, integrity monitoring is ensured by the possibility of restoring the original state of the program code when extracting a digital watermark from it [7].

3 The steps of the training

The practical part of the training is to prepare a monitoring digital watermark, embedding it in the FPGA program code and check the integrity of the program code using an embedded watermark. The following target FPGA components are used in the training:

– communication modules Intel Triple Speed Ethernet MegaCore Function [8];

– actuator control modules Intel Motor Control IP Suite Components for Drive-on-Chip Reference Designs (Fig. 1) [9].

The practical part of the training contains of two stages, which consist of the following steps.

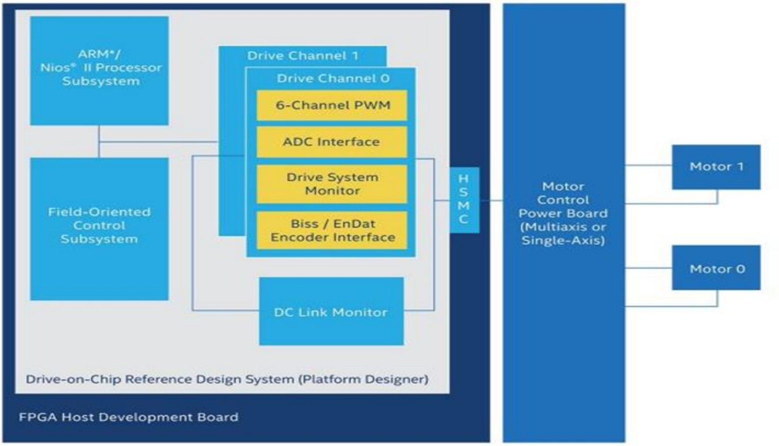


Fig. 1 – Intel Motor Control IP Suite Component

Stage of preparing the FPGA program code for integrity monitoring.

Step 1. The hash sum *H* is calculated for the information object whose integrity monitoring is to be performed. In this training, it is proposed to monitor the integrity of the entire program code of the FPGA device (configuration sof file). It is necessary to find a file in the folder of the studied FPGA-project, the name of which coincides with the name of the project, and the extension has the value sof. Then you should get a hash sum for this file. To obtain a hash sum, it is proposed to use any available online or offline hash sum service.

Step 2. A stego-key is set, containing formally defined rules (see previous training), according to which the bits of a digital watermark will be placed in the LUT-container of an FPGA chip. In this training, the stego-key is proposed to be simplified compared to the previous training. Namely, define the stego-key as a set of two parameters: *adr* – the number (address) of the bit of the program code of the LUT units, which, when an identifier is embedding, will act as the direct storage of the bits of the embedded identifier; *enumeration-rule* – a rule that specifies the order in which the LUT units used to implement the identifier bits are iterated. It is described verbally or as a formula.

Step 3. In the LUT-container space of the FPGA chip (in accordance with the stego-key), the stego-path $L = <l_1, l_2, \dots , l_n>$ is formed – an ordered sequence of LUT units that are intended to embedding the bits of the digital watermark (l_i is the identifier of the LUT units). Stego-path is formed by using the approaches and software discussed in the previous training.

Step 4. The sequential reading of the values of the bits located at a given *adr* address in the program codes of the stego-path LUT units (where the *adr* address is a component of the stego-key) is performed. As a result, an ordered sequence of bits $R = <r_1, r_2, \dots , r_n>$. is formed from the program codes of the units $L = <l_1, l_2, \dots , l_n>$.

Step 5. Using the lossless compression method, the binary sequence $R = <r_1, r_2, \dots , r_n>$ is compressed. As a result, a compressed sequence $R_{com} = <rc_1, rc_2, \dots , rc_m>$ is formed. Moreover, $m < n$.

For compression, it is proposed to use any available online or offline service. The difference in the lengths of the sequences *R* and R_{com} is also determined, which is Δ = m – n. Compression is performed to save the state that the information object of the program code had before embedding a digital watermark. The value of Δ is equal to the maximum possible length of monitoring information embedding in a digital watermark.

Step 6. By concatenation, a digital watermark *DWM* is formed, which contains a compressed R_{com} sequence, a checksum *H* and the information about the position from which the hash sum is in the digital watermark (this information is proposed to be set as an integer m – the length of the sequence R_{com}). Thus, a generated, digital watermark is the following set of sequentially arranged components: $DWM = <m, R_{com}, H>$.

Step 7. The digits of the *DWM* digital watermark are embedding in the program code of LUT $L = <l_1, l_2, \dots , l_n>$ blocks located on the stego-path. Embedding is performed using the approaches and software discussed in the previous training.

Stage of the integrity monitoring of the FPGA program code

Step 1. In the LUT-container space of the FPGA chip (in accordance with the stego-key), the stego-path $L = <l_1, l_2, \dots , l_n>$ is formed. Stego-path is formed by using the approaches and software discussed in the previous training.

Step 2. A digital watermark is read from the LUT container. For this, sequential reading of the values of the bits located at the specified *adr* address in the program codes of the stego-path LUT units (where the *adr* address is a component of the stego-key) is performed. As a result, an ordered sequence of bits $DWM = <d_1, d_2, \dots , d_n>$ is formed from the program codes of the units $L = <l_1, l_2, \dots , l_n>$.

Step 3. From the *DWM* sequence, bits are read that encode the length m of that portion of the digital watermark that contains the compressed initial state of the information object of the FPGA program code.

Step 4. Further, the following m bits are read from the *DWM* sequence, which contain the sequence $R^*_{com} = <rc^*_1, rc^*_2, \dots , rc^*_m>$. This bit sequence contains the compressed initial state of the information object of the FPGA program code.

Step 5. The remaining bits of the *DWM* sequence, containing the hash sum H^*, are being read.

Step 6. The decompression of the sequence R^*_{com} is performed. To do this, the decompression method should be applied, inverse to the compression method used at the stage of embedding the control information in the information object. For decompression, you are invited to use any available online or offline service. As a result, the decompressed binary sequence $R^*_{decom} = <r^*_{1\,decom}, r^*_{2\,decom}, \dots , r^*_{n\,decom} >$ is formed.

Step 7. The initial state of the information object of the FPGA chip program code is being restored. To do this, the LUT units $L = <l_1, l_2, \dots , l_n>$ are iterated over and the bit $r^*_{i\,decom}$, the sequence R^*_{decom}, is embedding into the program codes of the LUT l_i stego-path LUT units. The embedding of bits is performed using the method and software discussed in the previous training.

Step 8. The hash-sum H_{rec} is computed for the information object of the FPGA program code, which was restored in step 7. The hash-sum is calculated using the same hash function that was used at the monitoring digital watermark creation stage.

Step 9. A comparison is made of the H^* and H_{rec} hash-sum values. Their coincidence means no violation of the integrity of the program code of the FPGA chip. The discrepancy between these hash sums indicates that a integrity violation has occurred.

Step 10. Perform several variants of distorting parts of the FPGA chip code. Draw conclusions about whether the studied method fixes integrity violations for each of these options.

Variants of project violations: a) violations of one bit in the program code of any of the LUT units of the FPGA project; b) violations of one bit in the program code of several LUT units of an FPGA project; c) violations of several bits in the program code of several LUT units of an FPGA project; d) violations of one initial connection between LUT units of an FPGA project; e) violations of several original links between the LUT units of the FPGA project.

Variants of tasks

All tasks variants require issuing an FPGA component from Intel Motor Control IP Suite Components for Drive-on-Chip Reference

Designs [8] and Intel Triple Speed Ethernet MegaCore Function [9].

All tasks variants include:

– studying the integrity monitoring method based on embedding a digital watermark into an information object of the program code of FPGA-based components of IoT systems;

– familiarization with the principles of software operation, ensuring the formation of a control digital watermark;

– acquaintance with the principles of operation of the Intel (Altera) Quartus CAD modules that are involved in the embedding and extraction of the monitoring digital watermark;

– in the environment of special software, preparation of a digital watermark containing integrity monitoring information;

– implementation in the CAD Intel (Altera) Quartus environment of embedding a digital watermark into an information object of the program code of an FPGA-based component;

– checking the correctness of the functioning of the integrity monitoring method in cases of absence or presence of distortions of the program code of FPGA-based component;

– research of the possibilities of using a digital watermark to monitoring the integrity of the program code of FPGA-based devices;

– research of the applicability of the studied method in cases of: violation of the program codes of the elementary units of the FPGA-based component; violation of connections between units; combined violations; multiple and single integrity violation.

4 Test questions and tasks

1. What is the reason for the importance of monitoring the integrity of the program code of FPGA-based devices?
2. Describe the main ways to monitor the integrity of the program code.
3. Formulate the basic properties of cryptographic hash functions that determine the effectiveness of their use in the process of monitoring integrity.
4. What are the main ways of storing the check hash sum involved in monitoring the integrity of the program code?
5. State the advantages and disadvantages of using a digital watermark to monitor the integrity of the program code of FPGA-based devices.
6. What is the minimum set of components that should have a digital watermark used in the process of monitoring the integrity of the

program code of FPGA-based devices?

7. What software is used in the process of preparing a digital watermark?

8. What software is used in the process of embedding a digital watermark in the program code of FPGA-based devices?

9. What are the software tools for extracting a digital watermark from software code and performing integrity monitoring?

10. What is the purpose of restoring the original state of the information object of the FPGA program code when performing integrity monitoring?

5 References

1. V. Kharchenko, A. Gorbenko, V. Sklyar, C. Phillips, “Green Computing and Communications in Critical Application Domains: Challenges and Solutions,” in Proc. of the 9th International Conference on Digital Technologies, Zhilina, Slovak Republic, 2013, pp. 191-197.

2. Ajay Rupani1, Gajendra Sujediya, “A Review of FPGA implementation of Internet of Things,” International Journal of Innovative Research in Computer and Communication Engineering, 2016, vol. 4, issue 9, pp. 1-5.

3. Sanjeev Sharma, Revati Deokar, “FPGA Based Cost Effective Smart Home Systems,” International Conference on Advances in Communication and Computing Technology, 2018.

4. M. Bishop, Computer Security, 2nd Edition, Addison-Wesley, Boston, USA, 2018, 1440 p.

5. W. Stallings, Cryptography and Network Security: Principles and Practice, 7th Edition, Pearson Education Limited, Harlow, United Kingdom, 2017, 768 p.

6. J. Katz, Digital signatures. Advances in Information Security, Springer, New York, USA, 2010, 192 p.

7. J. Fridrich, Steganography in Digital Media, USA, New York: Cambridge University Press, 2010, 438 p.

8. Triple-Speed Ethernet Intel FPGA IP User Guide. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_ethernet.pdf

9. Motor Control IP Suite Components for Drive-on-Chip Reference Designs. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/ds_1038_doc_mc.pdf

APPENDIX A

TEACHING PROGRAM OF THE COURSE IT2 “IoT FOR SMART BUILDING AND CITY”

TITLE OF THE COURSE	Code
IoT for the Smart Building and City	**IT 2**

Teacher(s)	Department
Coordinating: Prof. Dmitry Maevsky **Others:** Module ITM2.1: DrS, Prof. Maevsky D. A., Ass. Prof., PhD Maevskaya O. J. Module ITM2.2: Ass. Prof., PhD Parkhomenko A.V.; Ass. Prof., PhD Gladkova O. M. Module ITM 2.3: DrS, Prof. Busher V. A, DrS, Prof. Bojko A. O. Module ITM 2.4: DrS, Prof. Drozd O. V.; Ass. Prof., PhD Martinuk O. M.	ONPU, Institute of Electro mechanics and Energetic Management; ONPU, Computer Engineering ZNTU Software Tools Department

Study cycle	Level of the module	Type of the module
Industrial Training	A	Full-time tuition

Form of delivery	Duration	Langage(s)
Full-time tuition	One semester	English

Prerequisites	
Prerequisites: Systems theory, Probability Theory and Foundations of Mathematical Statistics; Computer Systems and System Analysis, Risk Theory, Theory of Automatic Control; Computer Networks; Information-Networking Technologies, Modeling Foundation knowledge and skills in CAD	**Co-requisites (if necessary):** Experience with IDE

Credits of the module	Total student workload	Contact hours	Individual work hours
4	120	48	72

Aim of the courcs: competences foreseeen by the study programme

The aim of the course is:

- To create a knowledge base for multidisciplinary research systems theory, risk analysis and risk management. The study also expands the current research on systems IoT and hierarchy of smart systems;
- To obtain the knowledge and practical skills of software/hardware engineering applied to development of Smart Building Systems;
- Studying the control, executive and sensor elements used in the development of the Smart Building Management System, as Smart House & IoT objects that are integrated into extensible networks;
- Acquisition of knowledge in wireless and hybrid technologies of component interaction for the IoT SBC systems. Obtaining skills in development and debugging of simulation models of component interaction in the IoT SBC systems.

Learning outcomes of module (course unit)	Teaching/learning methods	Assessment methods
At the end of course, the successful student will be able to: 1. Understand the difference between natural and artificial systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
2. Be able to distinguish the main function of artificial systems and classify these systems according to their main function.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
3. Build hierarchical structures of systems based on their separation on the basis of arbitrary.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
4. To perform a risk assessment of artificial technical systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire
5. Use the method of peer review for risk assessment in IoT systems.	Interactive lectures, Learning in laboratories, Just-in-Time Teaching	Module Evaluation Questionnaire

Appendix A. Teaching program of the course

Themes	**Contact work hours**							**Time and tasks for individual work**	
	Lectures	Consultations	Seminars	Practiacl work	Laboratory work	Placements	**Total contact work**	**Individual work**	**Tasks**
1. Elements of System Theory 1.1. Key concepts of System Theory 1.2. Natural and artificial systems 1.3. Main function of artificial systems 1.4. The concept of an ideal system	2				4			10	Decomposition and main functions of smart building systems Decomposition and main functions of smart city systems
2. Internet of Thing: Hierarhy of Smart Systems. 2.1. Build hierarchical structures of smart systems 2.2. The concept of risk and methods of its evaluation 2.3. Build of risk matrices and its analysis	2				4			10	2.1. Risk analysis of smart building systems 2.2. Risk analysis of smart city systems.
3 Embedded systems as the basis of the IoT infrastructure 3.1 Embedded systems design	1						1	2	Reading literature

techniques.									and preparing presentation
3.2 Hardware /software platforms for embedded systems realization. Software/ hardware development based on Arduino platform. Development of interactive graphical interface for interaction with Arduino platform.					2		2	3	Working on individual tasks and preparing lab's reports
3.3 Software /hardware development based on Arduino platform. Working with sensors and actuators.					2		2	3	
3.4 Protocols and technologies for embedded systems interaction with other devices and Internet	1						1	3	Working on individual tasks and preparing lab's reports
4 Implementation of the software/hardware platform for Smart Building System.									
4.1 The	1						1	2	Reading

development of Smart Building System architecture.									literature and carrying Remote experiments using REIoT complex
4.2 The usage of Raspberry Pi and OpenHAB platforms for Smart Building System control. Raspberry Pi minicomputer implementation as a server for Smart building system. Integration of SBS subsystems based on OpenHAB platform.					4		4	3	Working on individual tasks and preparing labs reports
4.3 The application of the remote laboratory Smart House&IoT for Smart Building System prototyping	1						1	2	Working on individual tasks and preparing labs reports
5 Classification of the Building Management System, the purpose and basic properties of control, executive and sensor elements. 5.1 Introduction in Smart House and IoT systems. 5.2 Classificatio	2				2 2		**6**	9	Reading literature and preparing presentation

n of intellectual components, sensors, executive units of Building Management Systems. 5.3 Subsystems of Smart House: microclimate control, lighting, security in residential and industrial premises.									
6 Organization of the interaction of subsystems and elements of Smart House & IoT. 6.2 Microclimate control – functions, executive and sensor units, 6.3 Lighting – lux meters, astronomical timers, security components as sensor and control units in lighting subsystem. 6.4 Security subsystem – interaction with lighting, access control and protection subsystems. 6.5 Study of the security and lighting control	2				2 2		**6**	9	Working on individual tasks and preparing lab’s reports

system, shutters and blinds based on Theben-Luxor controllers 6.6 Construction of Smart House & IoT subsystems based on Moeller / Eaton xComfort Home-Manager equipment with wireless RF-EIB / KNX networks 6.7 Studying the principles of managing components of Smart House & IoT based on the controller Siemens Logo! with a module for connecting to the KNX network 6.8 Study the principles of configuring intelligent control systems for building and residential engineering systems in the KNX network in the ETS software environment									
7 Technologies of behavioral interaction of device processes in the IoT Smart Building and Sity	2			4			**6**	12	1.6. Features of simulation models of the IoT SBC systems with

(SBC) systems. 7.1 Introduction to behavioral technologies of component processes interaction into the structures of IoT SBC systems. 7.2 Work benches for development and verification of models in interaction of device proccesess in the IoT SBC systems. 7.3 Simulation models of behavior and synchronization for the IoT SBC systems with component integrity monitoring. 7.4 Features of simulation models of behavior and synchronization for the IoT SBC systems with component integrity monitoring. 7.5 Simulation models of behavior and synchronization for the IoT SBC systems with									a changeable traffic in the multipoint environment of wireless dynamic access

component integrity monitoring.									
8-Technologies of synchronize interaction of devices in the IoT SBC systems 8.1 Introduction to synchronize technologies of component interaction in the IoT SBC systems. 8.2 Work benches of development and verification of synchronize models in component interaction of the IoT SBC systems 8.3 Development of simulation synchronize models for the IoT SBC systems with secure authorization. 8.4. Simulation of component synchronize interaction in the IoT SBC systems with secure authorization.	2			4			**6**	12	2.5. Debugging of simulation models of component interaction in the IoT SBC systems with a changeable traffic in the multiroute hybrid environment
On the whole	**16**			**8**	**24**		**48**	**72**	

Assessment strategy	**Weight in %**	**Dead lines**	**Assessment criteria**
Lecture activity,	10	7,14	85% – 100% Outstanding work,

including fulfilling special self-tasks			showing a full grasp of all the questions answered. 70% – 84% Perfect or near perfect answers to a high proportion of the questions answered. There should be a thorough understanding and appreciation of the material. 60% – 69% A very good knowledge of much of the important material, possibly excellent in places, but with a limited account of some significant topics. 50% – 59% There should be a good grasp of several important topics, but with only a limited understanding or ability in places. There may be significant omissions. 45% – 49% Students will show some relevant knowledge of some of the issues involved, but with a good grasp of only a minority of the material. Some topics may be answered well, but others will be either omitted or incorrect. 40% – 44% There should be some work of some merit. There may be a few topics answered partly or there may be scattered or perfunctory knowledge across a larger range. 20% – 39% There should be substantial deficiencies, or no answers, across large parts of the topics set, but with a little relevant and correct material in places. 0% – 19% Very little or nothing that is correct and relevant.
Learning in laboratories	30	7,14	85% – 100% An outstanding piece of work, superbly organised and presented, excellent achievement of the objectives, evidence of original

			thought. 70% – 84% Students will show a thorough understanding and appreciation of the material, producing work without significant error or omission. Objectives achieved well. Excellent organisation and presentation. 60% – 69% Students will show a clear understanding of the issues involved and the work should be well written and well organised. Good work towards the objectives. The exercise should show evidence that the student has thought about the topic and has not simply reproduced standard solutions or arguments. 50% – 59% The work should show evidence that the student has a reasonable understanding of the basic material. There may be some signs of weakness, but overall the grasp of the topic should be sound. The presentation and organisation should be reasonably clear, and the objectives should at least be partially achieved. 45% – 49% Students will show some appreciation of the issues involved. The exercise will indicate a basic understanding of the topic, but will not have gone beyond this, and there may well be signs of confusion about more complex material. There should be fair work towards the laboratory work objectives. 40% – 44% There should be some work towards the laboratory work objectives, but significant issues are likely to be neglected, and there

			will be little or no appreciation of the complexity of the problem. 20% – 39% The work may contain some correct and relevant material, but most issues are neglected or are covered incorrectly. There should be some signs of appreciation of the laboratory work requirements. 0% – 19% Very little or nothing that is correct and relevant and no real appreciation of the laboratory work requirements.
Module Evaluation Quest	60	8,16	The score corresponds to the percentage of correct answers to the test questions

Author	Year of issue	Title	No of periodical or volume	Place of printing. Printing house or intrenet link
Compulsory literature				
M. D. Mesarovic Y. Takahara, editors	1975	General Systems Theory: Mathematical Foundations	Vol.	Elsevier Science
Fraga-Lamas P; Fernandez-Carames TM; Suarez-Albela M; Castedo L; Gonzalez-Lopez M.	2016	A Review on Internet of Things for Defense and Public Safety.	Volume: 16 Issue: 10	Sensors (Basel, Switzerland)
G. Giannopoulos, R. Filippini	2012	Risk Assessment and Resilience for Critical Infrastructures		http://www.moi.gov.cy/moi/cd/cd.nsf/5D9E4DBCF6DBB062C2257A3000294D18/$file/RISK%20ASSESSMENT%20AND%20RESILIENCE%20PROCEEDI

				NGS.pdf
W. Xi and L. Ling	2016	Research on IoT Privacy Security Risks		International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, 2016, pp. 259-262. doi: 10.1109/ICIICII.2016.0069 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7823536&isnumber=7823467
A. Ekelhart, S. Fenz, M. Klemen, E. Weippl	2007	Security Ontologies: Improving Quantitative Risk Analysis		https://www.sba-research.org/wp-content/uploads/publications/2007%20-%20Ekelhart%20-%20Security%20Ontologies%20Improving%20Quantitative%20Risk%20Analysis.pdf
Y. Y. Haimes	2008	Models for risk management of systems of systems	Vol. 1, Nos. 1/2	Int. J. System of Systems Engineering
P. Kertzner, J. Watters, D. Bodeau, A. Hahn	2008	Process Control System Security Technical Risk Assessment	Research Report no. 13	http://www.thei3p.org/docs/publications/ResearchReport13.pdf

		Methodology & Technical Implementation		
T. R. Peltier	2005	Information security risk analysis		CRC Press, Taylor & Francis Group
S. Ziegler, J. Rolim and S. Nikoletsea.	2016	Internet of Things, Crowdsourcing and Systemic Risk Management for Smart Cities and Nations: Initial insight from IoT Lab European Research project		2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Crans-Montana, 2016, pp. 611-616. doi: 10.1109/WAINA.2016.177
Bugeja, A. Jacobsson and P. Davidsson.	2016	On Privacy and Security Challenges in Smart Connected Homes		European Intelligence and Security Informatics Conference (EISIC), Uppsala, 2016, pp. 172-175. doi: 10.1109/EISIC.2016.044
McEwen A., Cassim H.	2014	Designing the Internet of Things		Wiley
Henke K.	2016	Remote and virtual tools in engineering		Zaporizhzhya, Dike Pole
Timmis H.	2011	Practical Arduino Engineering		Berlin, Springer
Horan B.	2013	Practical		NY, Apress

		Raspberry Pi		
Bell C.	2013	Beginning Sensor Networks with Arduino and Raspberry Pi		NY, Apress
Intel Corp.	2016	IoT Path-to-Product: How to Build the Smart Home Prototype		https://software.intel.com/en-us/articles/iot-path-to-product-how-to-build-the-smart-home-prototype
Jiayuan W. Sheng Z.	2012	Smart Home System		https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2012/jw937_sz369/jw937-sz369/jw937_sz369.html
Grasshopper.iics	2014	Introduction to the Internet of Things: What, Why and How		https://www.codeproject.com/articles/832492/stage-introduction-to-the-internet-of-things
Cvjetkovic V., Matijevic M.	2016	Overview of Architectures with Arduino Boards as Building Blocks for Data Acquisition and Control Systems	iJOE, vol. 12 (7)	http://www.online-journals.org/index.php/i-joe/article/view/5818
Ананьев В.А.	2001	Системы вентиляции и кондиционирования. Теория и практика		Moscow: Euroclimate

Харке В. Н.	2006	Умный дом. Объединение в сеть бытовой техники и систем коммуникаций в жилищном строительстве		Moscow: Techosphera
S3-Smart Software Solutions GmbH	2008	Руководство пользователя по программированию ПЛК в CODESYS		CODESYS_V23_RU.pdf
EIB/KNX				http://www.the-ark.kiev.ua
EIB - европейский стиль жизни				http://www.quinta-m.ru
	2017	The Unified Modeling Language		http://www.uml-diagrams.org/
	2017	Edit UML models and diagrams		https://msdn.microsoft.com/en-us/library/dd409405.aspx
	2017	UML Diagram Software - Perfect UML Diagram Examples, Templates, Knowledge, Software, Free Download		https://www.edrawsoft.com/UML-Diagrams.php
Ivo Adan, Jacques Resing	2015	Queueing Systems	182 P.	http://www.win.tue.nl/~iadan/queueing.pdf
Dr. Janos Sztrik	2012	Basic Queueing Theory	193 P,	http://irh.inf.unideb.hu/~jsztik/education/16/SOR_Main_Angol.pdf

	2017	TicToc Tutorial for OMNeT++		https://omnetpp.org/doc/omnetpp/tictoc-tutorial/
	2017	ExtendSim Books		https://www.extendsim.com/sols_books.html
G. Geeraerts		An Introduction to Petri Nets and how to analyse him...	341 P.	http://www.ulb.ac.be/di/ssd/ggeeraer/Tutorial-Petri-Nets-Geeraerts.pdf
	2011	HTTP over TLS, IETF RFC 2818		http://tools.ietf.org/html/rfc2818.
Andina J.	2017	FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics		CRC Press
J. Katz	2010	Digital signatures. Advances in Information Security		Springer, New York, USA
Ajay Rupani1, Gajendra Sujediya	2016	A Review of FPGA implementation of Internet of Things	vol. 4, issue 9, pp. 1-5	International Journal of Innovative Research in Computer and Communication Engineering
Sanjeev Sharma, Revati Deokar	2018	FPGA Based Cost Effective Smart Home Systems		International Conference on Advances in Communication and Computing Technology
V. Kharchenko, A. Gorbenko, V. Sklyar,	2013	Green Computing and Communication		Proc. of the 9th International Conference on

C. Phillips		s in Critical Application Domains: Challenges and Solutions,"		Digital Technologies, Zhilina, Slovak Republic
Additional literature				
Ashish Tiwari	2010	Theory of reals for verification and synthesis of hybrid dynamical systems		In Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation (ISSAC '10). ACM, New York, NY, USA, 5-6. http://dx.doi.org/10.1145/1837934.1837938
Paul A. Fishwick and Bernard P. Zeigler. 2, 1 (January 1992), DOI=http://dx.doi.org/10.1145/ 132277.132280	1992	A multimodel methodology for qualitative model engineering.	Vol. 2, No. 1	ACM Trans. Model. Comput. Simul. Pp. 52-81
Saurabh Mittal and Larry Rainey.	2015	Harnessing emergence: the control and design of emergent behavior in system of systems engineering.		In Proceedings of the Conference on Summer Computer Simulation (SummerSim '15), Saurabh Mittal, Il-Chul Moon, and Eugene Syriani (Eds.). Society for Computer Simulation International, San Diego, CA, USA, 1-10.

Wu Jun, Mei Lei, and Zhong Luo.	2011	Data security mechanism based on hierarchy analysis for internet of things.		In Proceedings of the 2011 International Conference on Innovative Computing and Cloud Computing (ICCC '11). ACM, New York, NY, USA, 68-70. http://dx.doi.org/10.1145/2071639.2071656
Parkhomenko A., Gladkova O., Zalyubovskiy Y., Parkhomenko A.	2017	Engineering of embedded systems		Zaporizhzhya, Dike Pole
Parkhomenko A., Tulenkov A. Sokolyanskii Y. Gladkova O., Zalyubovskiy Y., Parkhomenko A.	2017	Software-hardware platform for IoT technology studying.		Zaporizhzhya, Dike Pole
Parkhomenko A., Gladkova O., Ivanov E., Sokolyanskii A., Kurson S.	2015	Development and application of remote laboratory for embedded systems design	iJOE, vol. 11 (3)	https://www.online-journals.org/index.php/ijoe/article/view/4519/3501
Parkhomenko A., Gladkova O., Kurson S., Sokolyanskii A., Ivanov E.	2015	Internet-based technologies for design of embedded systems	Journal of Control Science and Engineering, vol.3(2)	http://www.davidpublisher.com/Public/uploads/Contribute/55d155b1313c7.pdf
Parkhomenko A., Tulenkov A., Sokolyanskii A.,	2017	Integrated Complex for IoT Technologies Study	Online Engineering & IoT. Lecture	Springer https://doi.org/10.1007/978-3-319-64352-6_31

Zalyubovskiy Y., Parkhomenko A.			Notes in Network and Systems vol. 22	
Parkhomenko A., Tulenkov A., Sokolyanskii A., Zalyubovskiy Y., Parkhomenko A., Stepanenko A.	2018	The application of the remote lab for studying the issues of Smart House systems power efficiency, safety and cybersecurity	Smart Industry & Smart Education. Lecture Notes in Network and Systems vol. 47	Springer https://doi.org/10.1007/978-3-319-95678-7_44
	2019	Motor Control IP Suite Components for Drive-on-Chip Reference Designs.		https://www.intel.com/content/dam/www/programmable/us/en/ pdfs/literature/ds/ds_1038_doc_mc.pdf
	2019	Triple-Speed Ethernet Intel FPGA IP User Guide.		https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_ethernet.pdf

АНОТАЦІЯ

УДК 004.415/.416:71/72](076.5)=111

Дрозд О.В., Маєвський Д.А., Маєвська О.Ю., Мартинюк О.М., Пархоменко А.В., Гладкова О.М., Дрозд М.О., Іванова О.М., Сурков С.С., Защолкін К.В. **Інтернет Речей для розумного будинку та міста.** Практикум / За ред. Д.А. Маєвського – МОН України, Одеський національний політехнічний університет, Запорізький національний технічний університет, 2019. – 155 с.

Практичні матеріали навчального модуля «IoT для розумних споруд та міст», наведені в цій книзі, розроблені в рамках проекту ERASMUS+ ALIOT 73818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP Internet of Thing: Emerging Curriculum for Industry and Human Applications. Курс зосереджений на застосуванні технологій Інтернету речей при проектуванні та розробці систем розумного будинку та розумного міста. Розглядаються питання оцінки ризику в підсистемах Інтернету речей, розробка програмних та апаратних платформ на базі Arduino та Raspberry Pi. Вивчається також побудова систем розумного будинку на базі промислових мікроконтролерів та програмовної логіки. Курс буде корисний працівникам промислових компаній, які займаються розробкою та впровадженням систем розумного будинку або розумного міста. Книга може бути корисною також студентам університетів та викладачам, які проводять заняття по відповідним курсам.

Бібл. – 79, рисунків – 66, таблиць – 6.

ЗМІСТ

ABSTRACT

UDC 004.415/.416:71/72](076.5)=111

Drozd O.V., Maevsky D.A., Maevskaya O.J., Martynyuk O.M., Parkhomenko A.V., Gladkova O.M., Drozd M.O., Ivanova O.M., Surkov S.S., Zashcholkin K.V. **Internet of Things for Smart Building and City.** Practicum / Edited by Maevsky D.A. – Ministry of Education and Science of Ukraine, Odessa National Polytechnic University, Zaporizhzhia National Technical University, 2019. – 156 p.

ISBN 978-617-7361-94-6.

Practical materials of study industrial training module “IoT for Smart building and city” given in this book are developed within project ERASMUS+ ALIOT 73818-EPP-1-2016-1-UK-EPPKA2-CBHE-JP Internet of Thing: Emerging Curriculum for Industry and Human Applications. The course focuses on the application of the Internet of things technologies in the design and construction of smart home systems. The issues of risk assessment in the subsystems of the Internet of things, the development of software and hardware platforms based on Arduino and Raspberry Pi are considered. It's described the construction of smart building systems based on industrial microcontrollers and FPGA.

The course will be useful to engineers of industrial companies that are engaged in the development and implementation of smart building systems. It could be useful for students of universities, lecturers and professors who conduct classes on corresponding courses.

Ref. – 79 items, figures – 66, tables – 6.

CONTENTS

Бойко Андрій Олександрович
Бушер Віктор Володимирович
Дрозд Олександр Валентинович
Маєвський Дмитро Андрійович
Маєвська Олена Юріївна
Мартинюк Олександр Миколайович
Пархоменко Анжела Володимирівна
Гладкова Ольга Миколаївна
Дрозд Мирослав Олександрович
Іванова Олена Миколаївна
Сурков Сергій Сергійович
Защолкін Константин Вячеславович

ІНТЕРНЕТ РЕЧЕЙ ДЛЯ РОЗУМНОГО БУДИНКУ ТА МІСТА

Практикум
(англійською мовою)

Редактор Маєвський Д.А.

Комп'ютерна верстка
В.С. Харченко
О.О. Ілляшенко

Зв. план, 2019
Підписаний до друку 27.08.2019
Формат 60х84 1/16. Папір офс. No2. Офс. друк.
Умов. друк. арк. 8,73. Уч.-вид. л. 9,43. Наклад 150 прим.
Замовлення 270819-4

Національний аерокосмічний університет ім. М. Є. Жуковського
"Харківський авіаційний інститут"
61070, Харків-70, вул. Чкалова, 17
http://www.khai.edu

Випускаючий редактор: ФОП Голембовська О.О.
03049, Київ, Повітрофлотський пр-кт, б. 3, к. 32.
Свідоцтво про внесення суб'єкта видавничої справи до державного реєстру видавців,
виготовлювачів і розповсюджувачів видавничої продукції
серія ДК No 5120 від 08.06.2016 р.

Видавець: ТОВ «Видавництво «Юстон»
01034, м. Київ, вул. О. Гончара, 36-а, тел.: +38 044 360 22 66
www.yuston.com.ua
Свідоцтво про внесення суб’єкта видавничої справи до державного реєстру видавців,
виготовлювачів і розповсюджувачів видавничої продукції
серія ДК No 497 від 09.09.2015 р.