

Information Technology of University Class Internet-of-Things-Module

Yaroslav Krainyk¹[0000-0002-7924-3878]

¹ Petro Mohyla Black Sea National University, 10. 68 Desantnykiv str., Mykolaiv, Ukraine
yaroslav.krainyk@chmnu.edu.ua

Abstract. In this paper, new information technology of Internet-of-Things-module (IoT-module) for university class has been investigated. The IoT-module is designed to improve user experience in educational process. It allows lecturer to provide additional content to students and run knowledge testing immediately during classes. The additional information is presented either in multimedia (images and video) or text format. The information technology with the devised module supposes extensive usage of network. Thus, specific communication protocol and has been designed for orchestrating multiple clients using server software. The module employs Ethernet technology for organization of communication with server. The class can be equipped with multiple instances of IoT-module. The usage of the module and information technology provides a scalable solution for modern learning environment enhanced by information-communication technologies. Information technology has been implemented in Proof-of-concept laboratory stand with two IoT-modules based on STM32 F7 development boards. The proposed solution has a huge potential for further improvement and can propose more features for specific learning environments.

Keywords: Internet-of-Things, education, technology, module, synchronization.

1 Introduction

Influence of Internet-of-Things (IoT) on each field of human activity is getting more and more notable. New devices are designed to integrate all types of environments into global network and provide new capabilities to the customers.

Educational environment and specifically educational process that involves usage of information computer technologies (ICT) should stay at the edge of the new technologies to match requirements of the time. It includes both learning of modern advances in the field and employment of contemporary ICT in educational process. From this perspective, IoT offers almost infinite possibilities for effective usage of new devices.

IoT-enabled modules can greatly simplify delivery of educational content to students. Usage of multimedia means has a great effect on the overall perception and understanding of materials during lecture or seminar. Providing such multimedia

Copyright © 2019 for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

modules are available for every attending student, there is a possibility to complement main flow of the classes with additional information on the topic or even personalized content. Another use case for such type of IoT-modules is brief immediate check of the disseminated materials. Not only it gives an opportunity to evaluate students' knowledge, but also is a great basis for analysis on how complicated lecture is for students, what topics are more popular. This is a great source of information for refinement of educational materials.

2 Analysis of scientific sources and industrial solutions

IoT can greatly facilitate in development of Smart Learning Environments (SLE) [1]. SLE may leverage various IoT-devices for improvements in educational process, students experience and involvement, etc. IoT provides multiple enabling means [2] that simplify integration of modern technologies and new features for SLE.

Wireless technologies [3-6] are dominating in the IoT-solutions because they do not severely limit location of device and allow changing device position. Numerous technologies are available in this segment and they can satisfy almost any customer demands. However, IoT-device does not necessarily need to be wireless one. Wired connection also can provide access to global network and its usage does not contradict main ideas of IoT. Ethernet connection provides high-speed reliable channel for data interchange. In this work, throughput and fidelity of data transmission are highly important factors because of multimedia data transfer. That is why Ethernet technology has been chosen as a basis of the network with IoT-modules for education.

Inclusion of ICT in the educational process became a trend from the beginning. However, possibilities for further development with IoT seem almost unlimited. Its integration with another information technologies like Big Data, Machine Learning, Cloud Computing, etc., completes the technology and spreads technical functions available to students and academic staff. That fact rises a lot of issues for integration of education with other fields [7-9]. Displays are the most common ICT-component in the education process. Typically, visual information is perceived much better than other types of information. Therefore, display is almost must-have component for modern IoT-device that targets to be used in education. Display is also a part of the system considered in this paper. Moreover, nowadays, displays are usually accompanied with touch panel or other sensor technology so the user can send commands to the system without any additional tools. Such systems are in education and many other fields. For instance, Briolight company (Mykolaiv, Ukraine) has developed interactive tables for learning anatomy [10]. This very complex system supports different modes and provides great study experience. Nevertheless, it is also oriented on the specific educational segment and cannot be used simultaneously by the group of students. Thus, we are developing solution that is deemed as lightweight device that available to everyone in the class.

The most common base for IoT-devices is embedded systems components. They include microcontrollers, wireless modules with Wi-Fi, Bluetooth, single-board computers, etc. They are limited in computational power and other resources. However,

modern microcontrollers shows decent results even in multimedia processing. At the same time, they consume much less energy in comparison with desktop processors. Therefore, microcontroller with multimedia capabilities can be used in the core of the proposed module.

In this paper, information technology for IoT-module for university class is proposed. IoT-module aims to perform simple functions like displaying text and image as well as quite complex task like video demonstration on the display. This work is a continuation of the results, achieved in [11].

3 Methods and results

The proposed system uses client-server architecture. The server is either pre-installed machine or portable device with necessary software to communicate with clients. IoT-modules run as clients. Data transmission inside the network is handled by the dedicated Ethernet-switch placed inside the classroom. It also provides access to the university network with possibility to exchange data even with external resources, e.g. cloud services. Fig. 1 illustrates the general connection scheme for the network.

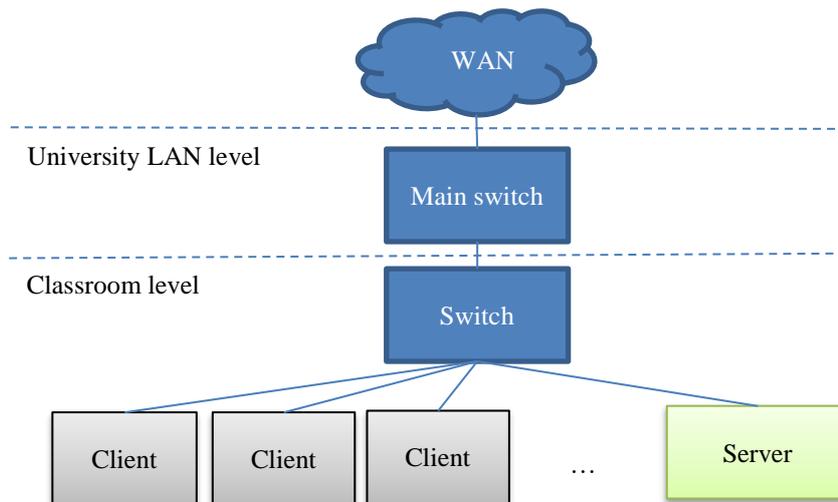


Fig. 1. General scheme for the proposed solution

Devised IoT-module comprises microcontroller, display with sensor touch panel, external RAM module, networking interface for communicating over Ethernet. This is the minimal set of components and it can be completed by other parts depending on the extension demands. Microcontroller manages networking data transmitted or received via network module, manipulates graphical data and shows data on the display, handles user actions on touch panel.

From the educational point of view, the main functions performed by the module are presentation of text and multimedia content (images and video) and test-based knowledge checking. It cannot serve as a complete and universal solution for educational tool but it should be quite effective within specified range of responsibilities.

Placement of the modules in the classroom depends on effective Ethernet cables connection and desks available in the class. The basic idea is to mount module on the surface of the desk. The module is stationary and for the power supply of the module USB-cable can be used or dedicated supply network. Separate interface for microcontroller programming should be routed. This ensures that device can be reprogrammed with adopted software in the future.

Presence of client-server architecture dictates that client software and server software applications are required. In this case, client solution is a firmware. It connects to the server and requests necessary information. At the server side, server software directs clients to perform necessary actions.

Both Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are involved into communication in the network. TCP is used for regulating testing process and image transmission. UDP plays a vital role in video data transmission.

Networking level in the microcontroller's firmware is controlled by lwIP [12] library. Because of the performance reasons and size of the output file, callback-style Application Programming Interface (API) has been chosen for implementation. It is based on usage of Protocol Control Block (PCB) structures. Developer is responsible to set up callback functions in the necessary moments of initialization, connection, and data processing. Communication protocol between server and client will be discussed further.

Most of the work that firmware is responsible for is done automatically after initialization. Thus, developer is responsible for correct initial configuration and organizing main loop, reaction to the interrupts, and setting callbacks. As peripheral interfaces for external memory, display with touch screen, and network interface are configured, a lot of functions related to orchestrating data do not require further control from the developer, e.g. data from dedicated in memory buffer will be automatically sent to the screen. To clarify connections among all main comprised parts of the IoT-module on the firmware level, let us consider them in more details.

Memory is a key part of the IoT-module that is heavily utilized for networking and multimedia functions. Different types of memory are used for the specified purposes. Network functions rely on memory buffer placed in RAM while image processing keeps data in the external DRAM. Thus, when image data are received from network, transfer from RAM to DRAM takes place. The destination matches buffer that is used by display controller to output visual data from memory.

It is convenient to organize processing of touch actions via interrupts. It allows immediately react to the user actions and provide better User eXperience (UX). However, the main issue with touch sensor is that it tends to react too often on the event. Therefore, another solution for touch processing is usage of internal counter and its update (overflow) interrupt. The interrupt occurs in fixed amount of time and does not overreact on the external influence. In the interrupt handler, state of the sensor is

checked and necessary action is applied according to this information. In our case, it is used to identify which item is selected as the answer to the test question.

At the server side, the implementation is different, as socket interface has been used. The server applications launches new thread in which server socket object accepts incoming connections. When new connection is accepted, new thread is created and all further communication with this connection are associated with this thread. This fact indicates that number of communication threads is equal to the number of clients installed in the classroom. As client is connected, the server software initiates all further actions. Lecturer starts the testing from the user interface. All multimedia data transmission begins with the command of lecturer. It greatly simplifies development of the firmware because it just need to handle data and process received data. Most logic functions of the systems are executed at the server side. From the software point of view, instances of socket interface hold network communications. Each thread governs single client socket.

As has been mentioned previously, server communicates with clients through multiple threads. Thus, server application is multi-thread application with the following thread types:

1. Main thread.
2. Server thread.
3. Client thread.
4. Synchronization thread.

Main thread in the thread that is created first as the program run. It starts server thread that is listening for incoming connections on the specific port. When the new connection is available, client thread is launched. However, running the threads is just a part of the problem to deal with. In the communication process, external resources are accessed to send data to clients as well as receive data from them. Therefore, synchronization among the threads are required to coordinate them and keep whole system in the consistent state. This is the main objective of synchronization thread.

The function that executes when the thread is started (e.g. run in Java) is organized as the infinite loop. In the infinite loop thread waits for the user action, sends data to the client, expects reply if necessary. The data to send are identified by user action and internal state of the client thread.

The idea behind synchronization mechanism for discussed systems is in the following. Countdown latch synchronization primitive is supposed to be used for this purpose. Each client thread accesses three latches. The first one is used to postpone execution of the main thread logic to the moment when lecturer selects some action. Its initial count value is equal to one. Synchronization threads notifies all client threads about it by decrementing the latch. Then data interchange section goes where selected content is being sent to the client and reply is received. Single run of the thread body is done at that moment. However, different threads finish execution at the different time. Thus, they decrement another countdown latch to inform synchronization thread about finished work. The initial value for this counter is equal to the number of connected clients. At the moment when the last thread finishes execution of the loop, synchronization thread is aware of the fact that all threads have done the assigned

task. Nevertheless, by the moment when any of the threads goes to the next cycle, new countdown latch should be set. With that intention additional countdown latch wait operator is placed at the end of the client thread cycle. If all threads are finished, synchronization thread creates new instance of countdown latch that is checked in the beginning of the client thread loop and decrements last one in the client threads. This procedure assures that no client thread starts execution before any other thread.

All the transmission from server to clients goes straightforward with no data sent back except the case of testing. If lecturer selects testing option, server needs to receive replies from the clients. Question messages alternate with replies from the user. Client receives single question with answer variants at a time. Next question is available only after reply on the current question.

Comparison of networking features implementation in firmware and server software indicates that they leverage different approaches. Server contains enough computational resources to bring sockets into work while low-level programming interface in the firmware allows reducing size of the output file. The trade-off for this attainment is more complicated program code (more lines of code are required, more structure instances participate in the information exchange, structural complexity of the program, etc.).

TCP and UDP is the basis for communications on the transport layer. However, custom application protocols for correct interpretation of data is compulsory in this case. The following data goes from server to the clients via TCP:

1. Text information to display on screen.
2. Image information.
3. Test information.
4. Scores of the testing.

In opposite direction go only answers on the test questions. As there are only four variants of how to interpret data, the simplest solution is to excrete single bit in the first package to identify type of the message. Several following bytes (e.g. four bytes) should identify total number of data bytes in the message. The rest part of the message is useful information that should be processed according to the message type. To display image on the screen, separate image buffer is available in memory. Image data are transmitted in the raw form in Alpha-Red-Green-Blue (ARGB) format. Test information contains question text, options to display as answers. They are separated from each other with a specific symbol (e.g. #) so corresponding parts can be fluently extracted from the received string. When question and options are available, they are displayed on the screen using high-level API (writing into image buffer is involved). Text information can be transmitted without any additional markers. The last message type (test score) informs student how successful he or she was during the quiz. It is also quite simple in implementation as it contains only number of point or percentage of correct answers that can be represented by single additional byte (e.g. 0-100% correct answers).

Video information is transmitted using UDP due to the performance reasons. Data are sent only from server to clients in this mode. Multicast transmission is employed to send data to solve the problem of delivering all data chunks on the network level.

Although, UDP is not reliable protocol and it cannot prevent possible losses in the network. In context of presentation of video data on the screen, that means appearance of artifacts in the reconstructed frame. Regular transmission of synchronization package is applied to solve this problem. Reception of this package means that writing position in memory buffer is to be moved to the beginning of the buffer. This package is sent after each datagram that represents last chunk of data from image frame extracted from the video at the server side. Synchronization package has fixed width and can be distinguished among other packages by this parameter. Similar method is used in the system to collect image information from multiple sensors [13]

Proof-of-concept system that takes advantages of the proposed information technology has been implemented. STM32F7 Discovery board [14] has been chosen as a hardware platform for IoT-module. It contains all necessary devices in the single board and has wide opportunities for extension. Central point of the processing power for the board is F7-series microcontroller from ST Microelectronics. Firmware for the board is written in C programming language using assist from tools like STM32CubeMX and development environment System Workbench for STM32. The developed laboratory stand includes two boards and Ethernet switch. Two instances of modules are sufficient to check correctness of multi-thread software and all related issues. They are connected using Ethernet cables. Almost any device with Ethernet interface can be used a server. In our case, laptop served as a server machine. Server software has been deployed on the laptop. The software is developed in Java programming language and provides basic user interface to access all described functions of the system. The main form of server application is demonstrated in Fig. 2.

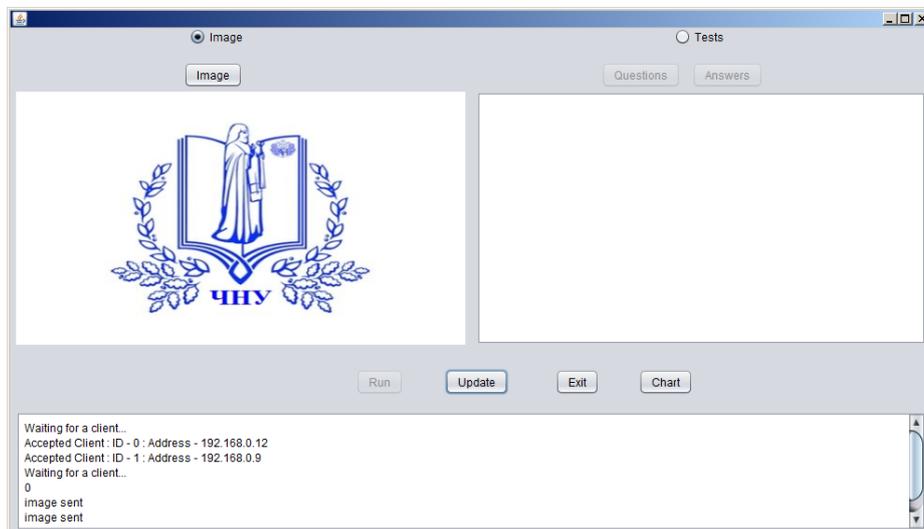


Fig. 2. Main form of the developed application

Lecturer selects image file to send to the IoT-modules, then it can be sent. The image's width and height in pixel must match respective dimensions of the screen. Most

popular image formats (PNG, JPG, BMP) are supported. The result of this function execution is demonstrated in Fig. 3.



Fig. 3. Devices on the stand in the image display mode

The basic testing features are also available to the user. The user prepares file with test in the specific format. First line contains question, three lines following represent variants of answer. Then the pattern repeats so file can contain as much questions as it is necessary. File with answers should be prepared to automatically send score response and collect statistics on the students' performance. In Fig. 4, you can observe devices with test questions shown on the display during the sample test session for two devices. The design interface is simple but it provides access to the necessary functions.

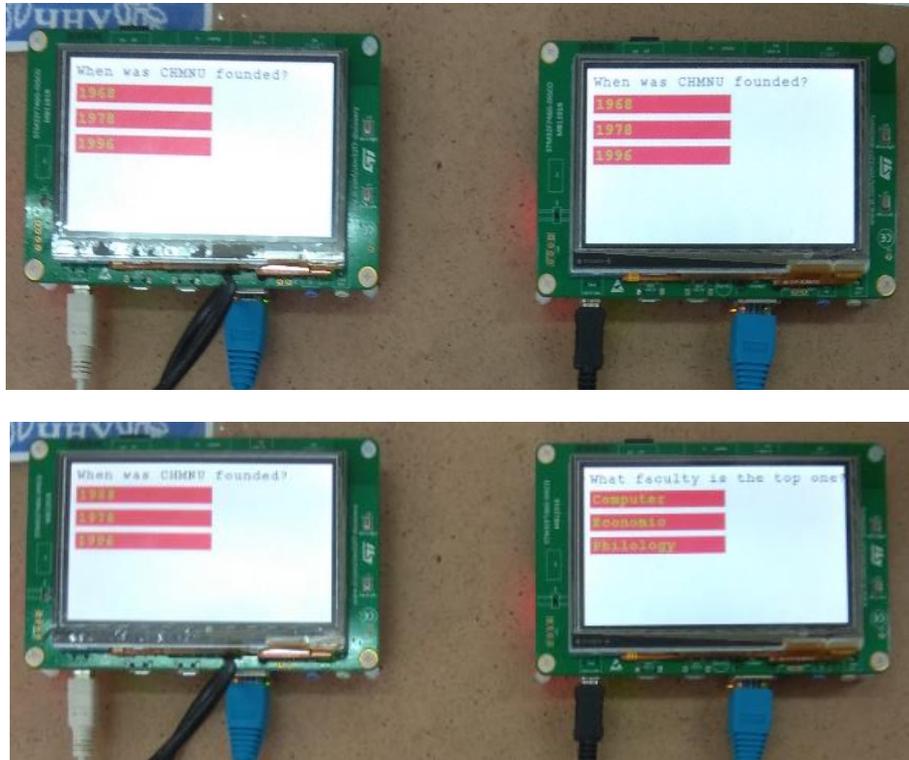


Fig. 4. IoT-modules in knowledge testing mode

The results of passing test for both cases are demonstrated in Fig. 5.



Fig. 5. IoT-modules display results of the test

The proposed information technology and developed IoT-modules based have huge potential for further improvements and depending on components that included into the system. However, it should be tested in more rigorous approaches to identify its reliability and operation conditions [15].

Acknowledgments. The current project has been developed with support of the SEA company (Kyiv, Ukraine). The author is grateful to company representatives for the development board that have been used for the experiments and proof-of-concept exemplar implementation.

4 Conclusions

In this paper, information technology for IoT-module of university classroom has been established. First, general network topology for the educational environment has been devised and considered individual roles of IoT-modules and server in such network. Second, comprised parts for the modules has been proposed to implement necessary functionality. The module's general purpose is to enrich educational experience for lecturer and for students. It augments typical class flow by making possible immediate knowledge testing and demonstration of additional multimedia or text information. Data communication between server and clients (IoT-modules) has been introduced. In the communication phase, application-level protocol is proposed to cover all the proposed functions. Consequently, information technology has been implemented in laboratory stand that includes two clients and allows connection of device with installed server software. According to the devised statements, firmware for IoT-modules has been developed. The stand demonstrates all described features and may be extended during the future investigations on the topic.

References

1. Spector, J.: Smart Learning Environments: Concepts and Issues. In: G. Chamblee & L. Langub (Eds.), Proceedings of Society for Information Technology & Teacher Education International Conference (pp. 2728-2737). Savannah, GA, United States: Association for the Advancement of Computing in Education (AACE) (2016).
2. Alsubaei, F., Abuhusseini, A., Shiva, S.: An Overview of Enabling Technologies for the Internet of Things. In: Hassan, Q. (ed.) Internet of Things A to Z: Technologies and Applications, pp. 79-111, John Wiley & Sons (2018). DOI: 10.1002/9781119456735.ch3
3. Krainyk, Y., Davydenko, Y., Tomas, V.: Configurable Control Node for Wireless Sensor Network. In: Proceedings of 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2-6 July, pp. 258-262. IEEE (2019). DOI: 10.1109/AIACT.2019.8847732
4. Krainyk, Y., Davydenko, Y., Starchenko, V.: Message-level Decoding of Error Patterns for Turbo-Product Codes. In: Proceedings of IEEE 39th International Conference on Electronics and Nanotechnology (ELNANO), 16-18 April, Kyiv, Ukraine, pp. 660-663. IEEE (2019). DOI: 10.1109/ELNANO.2019.8783849
5. Kondratenko, Y., Kondratenko, G., Sidenko, I.: Multi-criteria selection of the wireless communication technology for specialized IoT network. In: 14th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, Workshops, ICTERI, Vol. 2104, pp. 501-516. Kyiv, Ukraine (2018).

6. Kondratenko, Y., Kondratenko, G., Sidenko, I.: Multi-criteria decision making for selecting a rational IoT platform. In: IEEE 9th International Conference on Dependable Systems, Services and Technologies, DESSERT, pp. 147-152. Kyiv, Ukraine (2018).
7. Poliakov, M., Larionova, T., Tabunshchik, G., Parkhomenko, A., Henke, K.: Hybrid Models of Studied Objects Using Remote Laboratories for Teaching Design of Control Systems. *International Journal of Online and Biomedical Engineering* 12(9), 7-13 (2016). DOI: 10.3991/ijoe.v12i09.6128
8. Ursutiu, D., Kuchirka, Y., Samoila, C., Vytvytska, L.: The Online Course and Laboratory for Study of Automatic Control Systems. *International Journal of Online and Biomedical Engineering* 12(1), 24-27 (2016). DOI: 10.3991/ijoe.v12i01.5106
9. He, J., Lo, D. C.-T., Xie, Y., Lartigue, J.: Integrating Internet of Things (IoT) into STEM undergraduate education: Case study of a modern technology infused courseware for embedded system course. In: 2016 IEEE Frontiers in Education Conference (FIE), Erie, PA, USA (2016).
10. Anatomy table | Briolight Innovative electronic solutions, <https://briolight.com/en/anatomy-en/>, last accessed 2019/02/03.
11. Krainyk, Y., Razzhyvin, A., Bondarenko, O., Simakova, I.: Internet-of-Things Device Set Configuration for Connection to Wireless Local Area Network. In: Luengo D., Subbotin S., Arras P, et. al. (eds.) *Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, Ukraine. CEUR-WS, vol. 2353, pp. 885-896 (2019).
12. Dunkels, A.: *Design and Implementation of the lwIP TCP/IP Stack*. Swedish Institute of Computer Science (2001).
13. Zhuravska, I., Musyienko, M., Tohoiev, O.: Development the heat leak detection method for hidden thermal objects by means the information- measuring computer system. In: Luengo D., Subbotin S., Arras P, et. al. (eds.) *Computer Modeling and Intelligent Systems (CMIS-2019)*, Zaporizhzhia, Ukraine. CEUR-WS, vol. 2353, pp. 350-364 (2019).
14. 32F746GDISCOVERY – Discovery kit with STM32F746NG MCU, <https://www.st.com/en/evaluation-tools/32f746gdiscovery.html>, last accessed 2019/02/07.
15. Seniv, M., Yakovyna, V., Symets, I.: Software for visualization of reliability block diagram and automated formulation of operability conditions of technical systems. In: 14th International Conference on Perspective Technologies and Methods in MEMS Design, MEMSTECH-2018, pp. 191-195, Lviv, Ukraine (2018).