Using an ESP8266 Microcontroller to Develop a Learning Game

G Chursin and M Semenov

National Research Tomsk Polytechnic University, Lenin Avenue, 30, Tomsk, Russia, 634050

E-mail: sme@tpu.ru

Abstract. This article discusses the possibility of connecting an ESP8266 microcontroller to a learning game developed on the Unity3D game engine. The game is aimed at teaching and developing the skills of students, such as mathematics, physics, programming and the basics of robotics. The paper presents a diagram of the interaction of the microcontroller and the game, as well as a comparative analysis of the Arduino and ESP8266 platforms. Comparison of the Arduino firmware process via USB and ESP8266 over Wi-Fi is carried out. It was possible to establish that the firmware by air takes an average of 2 seconds more, which does not affect the gameplay.

1. Introduction

Traditional education system has remained unchanged for many years. There are various technologies and additional educational courses. However, school education remains unchanged in most schools. Traditional education is based on obtaining basic knowledge, skills and abilities, often without explaining the need to obtain this same knowledge [1]. Obviously, it is necessary to look for ways to renew traditional approaches, to develop active educational methods using modern pedagogical tools and technologies. The authors [2] note that computer games are a promising trend in the development of educational techniques. Globally, games can be divided into two categories: a) basic games for preschool children, where they learn to read, count, perform elementary tasks, b) highly specialized games (simulators) to teach certain competences. Thus, the sphere of educational computer games for schoolchildren is poorly developed.

This work is a continuation of the research [3], where the process of connecting Arduino to Unity3D game engine was shown. The work also presented the results of testing the proposed game on real players.

The aim of the work is to create a computer video game for teaching schoolchildren. A feature of the game is the need to use school knowledge to solve practical problems and puzzles. This will increase the level of motivation and interest in studying the basic educational program. In addition, IT-technologies and the direction of the Internet of Things are actively developing [4]. Therefore, it is necessary to educate students using modern technology. For this purpose, the proposed game technology uses real microcontrollers and various sensors that have physical interaction.

The main contribution of this study to education is the development of an approach to elective classes. We strive to expand the horizons of students with the help of computer games.

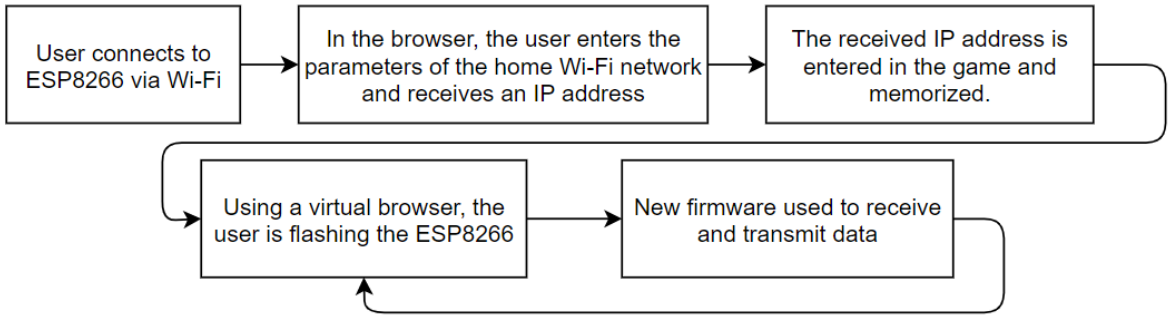

Figure 1. Diagram of a ESP8266 microcontroller connection to the game.

Table 1. Comparing the Connection of the ArduinoUNO Board and the ESP8266 Microcontroller to the Unity3D Environment.

Criterion	Arduino UNO, USB connection	ESP8266, Wi-Fi connection
Simplicity of connection	Firmware requires additional bootloader and operating system tools	Firmware does not require third-party software. Game remains completely isolated from operating system interaction
Stability of connection	USB connection does not cause firmware failures	Network problems can cause connection failures, preventing the game from continuing
Usability	Permanent physical connection required	It is possible to collect game circuits throughout the Wi-Fi coverage area
Universality	Using the firmware program, allows to use only the personal computer	Wireless firmware allows you to use both personal computer and smartphones, tablets
Time of loading of a firmware, s	5 and less	7 and less

2. Algorithm Description

In this work, a feature of the game is the physical communication with microcontrollers as well as various sensors. At the same time, it is important to understand that often the gaming process can take place on a smartphone or tablet. In this case, it is difficult to connect the board via physical cable. To avoid this problem, a Wi-Fi wireless LAN connection is used. A combination of Ardunio platform and Wi-Fi module are the most popular solution is the use. Note that this combination does not allow you to update the firmware without physically connecting the board to your computer. In order to use the microcontroller during the game, this is a prerequisite, as it is necessary to change the firmware depending on the proposed task.

To solve this problem, we propose to use a ESP8266 microcontroller (Espressif Systems CO., LTD, Shanghai, China) a feature of which is the presence of Wi-Fi and Bluetooth modules on the same board as the microprocessor. This allows to eliminate additional connections and reduce the final scheme of the project. The cost of such a microcontroller is comparable to that of the popular Arduino board (manufactured in China) without the modules described above. At the same time, the speed of the processor operation is higher. This card can be pierced without

using a physical cable, but with HTTP. Figure 1 shows a diagram of connecting the ESP8266 microcontroller to the game using Wi-Fi. Initialization can take place with any device that can connect to the Wi-Fi network and the host.

The microcontroller firmware is updated with pre-compiled binaries. Firmware refers to software that manages the microcontroller. The source code for the ESP8266 is developed in the C programming language using ESP8266_RTOS_SDK [5] from Espressif Systems, which includes libraries and a compiler. The file size is only 315 kb on average, able to store a lot of firmware. All firmware is uploaded to the game file archive. Based on the criteria [2], we conducted a comparative analysis of the connection of the ESP8266 and ArduinoUNO boards to the Unity3D game development environment (Table 1). Note also that the microcontroller ESP8266 has more RAM and the processor has a higher frequency [4]. This makes it possible to use more calculations and significantly expand the schemes used.

3. Test game scheme

An Arduino keyboard was used as the first test scheme between the Unity3D game engine and the ESP8266 microcontoller. An Arduino keyboard is a board with multiple keys that can be controlled by a single analog port. The assembly is shown in Figure 2. A developer independently assigns tasks for each key. The principle of operation is to change the signal voltage to the analog input of the microcontroller. The voltage is converted to numbers from 0 to 1023. Depending on the key pressed, values are supplied to the input of the microcontroller. Depending on the incoming values, can be assigned a conditional number to each key. In our case, we assigned the numbers from 1 to 5 as shown in Figure 2. Pressing a key only leads to display on the monitor.

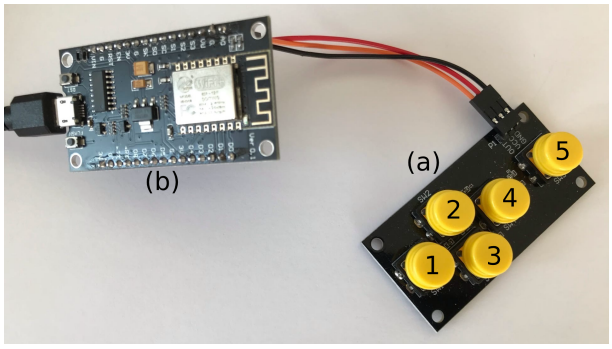

Figure 2. Connecting an Arduino Keyboard (a) to an ESP8266 microcontroller (b).

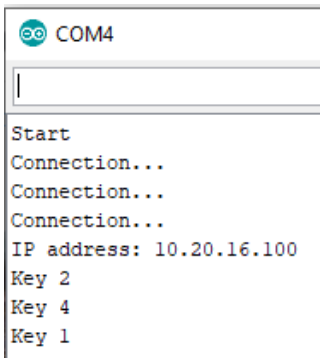

Figure 3. Debug logs showing ESP8266 initialization and keystrokes.

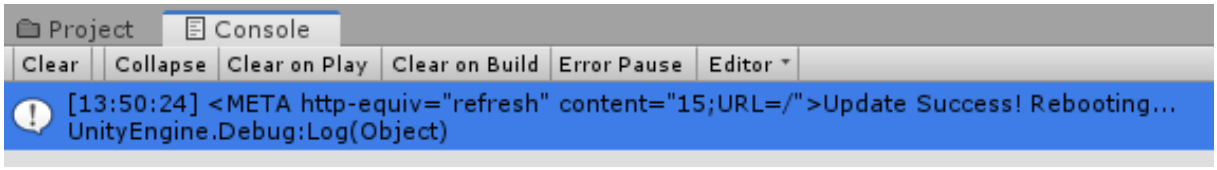

Figure 4. Output response from ESP8266 microcontroller after firmware submission.

In order to be able to flash ESP8266 via Wi-Fi, it is necessary to prepare and download the first firmware over a USB cable. This firmware contains the necessary functionality to update the ESP8266. After that, can update the firmware without using a cable. For this, firstly a new

firmware must be prepared. The source code is compiled and a binary file is created. This binary file is imported into the Unity3D project. After that, the firmware is sent to the microcontroller using an HTTP request.

The figure 3 shows the information after the firmware update. One can see the IP address 10.20.16.100 of the microcontroller, and Key 2, 4, and 1 – keystroke numbers on the board, both serial port and Wi-Fi in Unity3D show and debug keystroke numbers. After sending a new firmware, the board responds to a successful update. The figure 4 shows the successful response to the ESP8266 microcontroller update.

4. Results

A trial version of the game and firmware for the ESP8266 microcontroller were developed. Data transmission between the ESP8266 microcontroller and the Unity3D was performed using the TCP/IP protocol. The use of this protocol allows to be sure of the sequence of transmitted and received data, which is important for the smooth operation of the game. Data transfer rate via USB and Wi-Fi does not differ according to the results of experiments. Combining only two devices allows to create a socket server directly on the microcontroller. This does not affect performance or quality of data transfer.

It is worth noting the peculiarities of using such a scheme. Firstly, the ESP8266 board contains only one analog input. This means that not all sensors can be connected directly to the microcontroller. Secondly, after updating the board over HTTP, the old firmware is erased. Therefore, you must add functionality to each new firmware to update over Wi-Fi. Attention must be paid to the sensors and components used. Poor quality items may cause errors. For example, there are Arduino keyboards on which two different keys give the same signal. This is due to a manufacturing error.

5. Conclusion

Learning through entertainment is becoming a powerful alternative to traditional school and supplementary education practices. Interactive equipment, mobile applications, event and game technologies, design activities are actively used for implementation of training programs. Experiments involving the connection of the ESP8266 microcontroller and the Unity3D development environment suggest that this scheme can be used to develop a learning game. Experiments involving the connection of the ESP8266 microcontroller and the Unity3D development environment suggest that this scheme can be used to develop a learning game [3]. Using the ability to update firmware over wireless communication, with pre-compiled files, makes it possible to simplify the interaction of the game and the board as much as possible, and to port the game to smartphones and tablets. The proposed scheme for connecting a microcontroller to a virtual game makes it possible to eliminate the need to install drivers and libraries on a computer.

The next level of the project development is the creation of a game script, the development of a virtual part of the game. All game and microcontroller interaction adjustments will take place during the game design phase. It is also necessary to consider the options of creating and compiling source code directly during the game. Compiling the code during the game will add programming elements, as well as expand the game.

References

[1] Semenov M, Colen Y S, Colen J and Pardala A 2020 *J. Korean Soc. Math. Educ., Ser. D, Res. Math. Educ.* **23** 47–62
[2] YokeSeng W and Hayati M 2014 *Procedia - Social and Behavioral Sciences* **123** 215–224
[3] Chursin G and Semenov M 2020 *Journal of Physics: Conference Series* **1488** 012023
[4] Patnaikuni D R P 2017 *International Journal of Advanced Research in Computer Science* **8** 2350–2352
[5] ESP8266, Espressif Systems CO., LTD., China `https://docs.espressif.com/`