A METHOD OF WRITING HEX CODE TO ATMEGA8 MICROCONTROLLER THROUGH ARDUINO DEVICE

Ruzimov Omon Narimanovich
Tashkent University of Information Technologies named after Muhammad al-Khwarizmi Department of academic activity chief specialist
omonruzimov89@gmail.com

ABSTRACT

In this article shows a method of writing hex code to atmega8 microcontroller through the Arduino device. This atmega8 microcontroller consumes only 5 or 3.7 volts. Using devices without extra costs and provides low voltage of electricity consumption. In addition, it is intended to control remote devices using atmega8 microcontroller and these are very convenient to put in a box. It can also be added to the list of installed systems.

Keywords: Arduino device, Atmega8 microcontroller, controller, hex code, 5, 3.7 voltage, I/O ports.

INTRODUCTION

Nowadays, digital technologies are developing very much. Also writing code to the microcontroller, remote control of devices and their monitoring works are also developing. There are many types of microcontrollers. Atmega8, atmega16, atmega32, atmega328 and so on. These controllers are selected depending on the project. As an example, the microcontroller we chose was the atmega8. Because this microcontroller also works on 3.3 volts apart from that it is small and comfortable. For this reason atmega8 microcontroller was chosen. Figure 1 shows a picture of an atmega8 microcontroller.

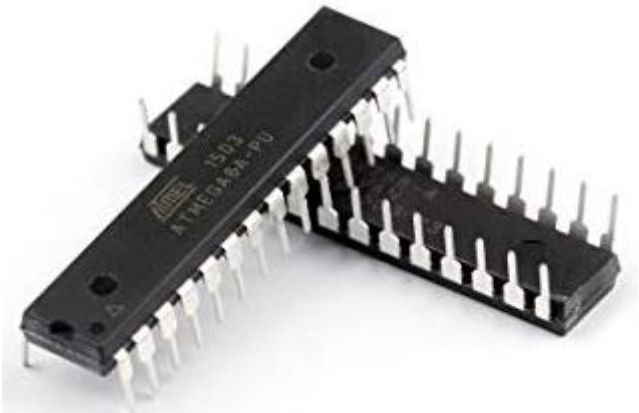

Figure 1. Atmega8 microcontroller

Atmega8 microcontroller has 28 I/O ports. Each port has its own function. All of the I/O pins have secondary functions. These are shown in parenthesis on the pinout diagram. PC6 is almost always used as a reset pin and is not normally available for I/O. PB6 and PB7 are often used for external crystal oscillators, but not in this tutorial. The following Registers are used for reading and writing to the I/O ports. Input/output ports are shown in table 1.

Table 1. List of input/output ports

Register	Type	Description	Notes
DDRB	Read/Write	Port B Data Direction Register	1=output, 0=input
PORTB	Read/Write	Port B Data Register	
PINB	Read only	Port B Input Register	
DDRC	Read/Write	Port C Data Direction Register	1=output, 0=input
PORTC	Read/Write	Port C Data Register	
PINC	Read only	Port C Input Register	
DDRD	Read/Write	Port D Data Direction Register	1=output, 0=input
PORTD	Read/Write	Port D Data Register	
PIND	Read only	Port D Input Register	

To write the code to the Atmega8 controller, we open the functions shown in the image below using the Arduino software and start the workflow.

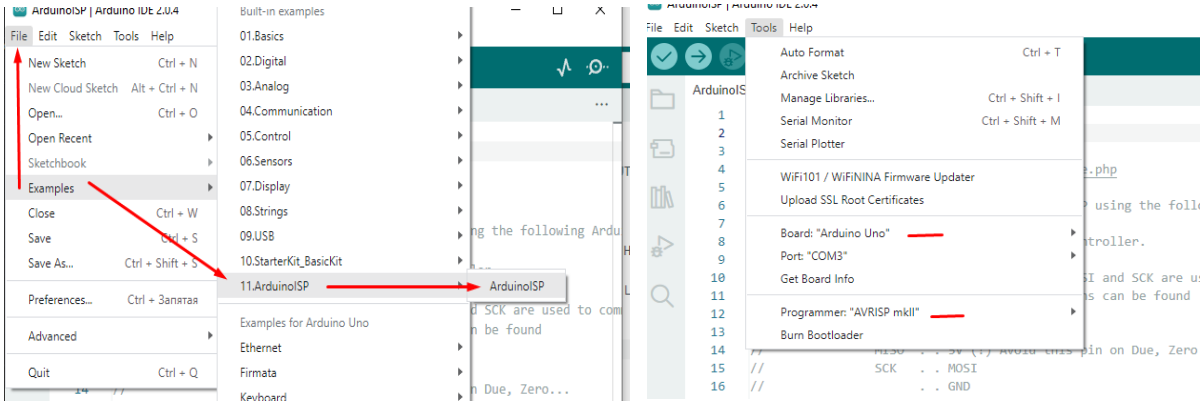

Figure 2. ArduinoISP function and setting

The process after completing the function and setting is as follows. The update operation shown in Figure 3 is executed.

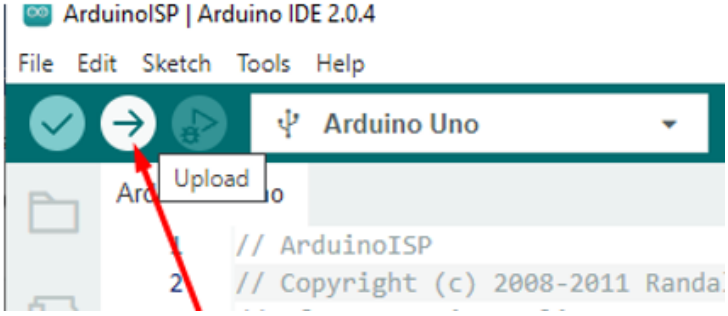

Figure 3. The update operation

All the indicated work is done correctly and the next step is carried out. From the tools function, it performs the settings for the atmega8 controller. Figure 4 shows.

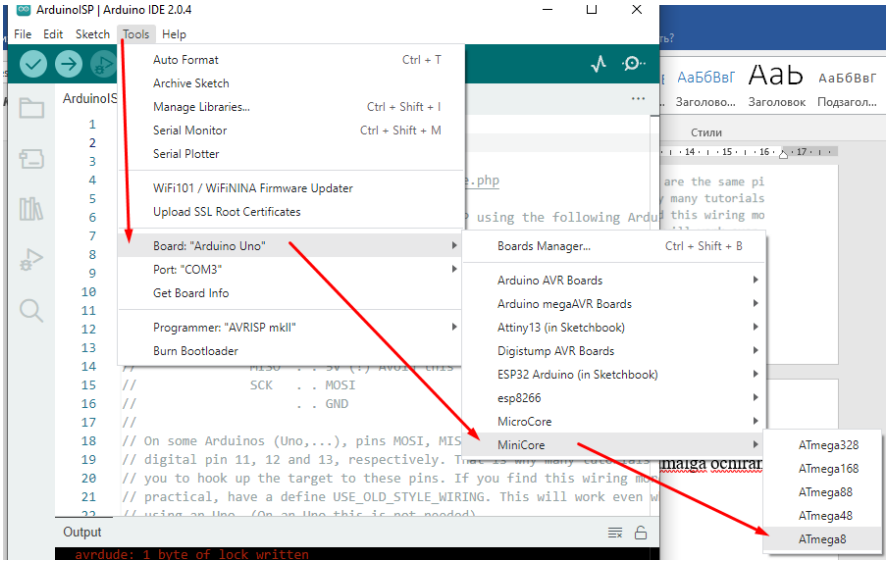

Figure 4. Tools board arduno

In Figure 4, the atmega8 microcontroller is selected from the tools function. After this process is done, the setup process is performed. this process is shown in Figure 5.

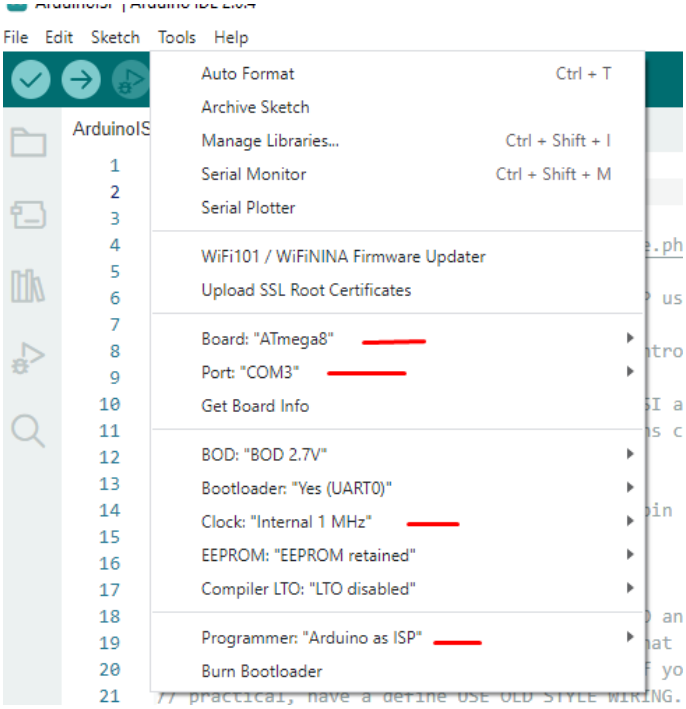

Figure 5. Device setup process

After the settings are made, the Burn Bootloader function is clicked. The code written in the Arduino program is sent to the microcontroller in hex format. When the process is finished, a single update of the atmega8 controller will be performed. The code written in the Arduino program is as follows.

```
void setup() {
  pinMode(10, OUTPUT);
}

void loop() {
  digitalWrite(10, HIGH);
  delay(1000);
  digitalWrite(10, LOW);
  delay(1000);
}
```

After that, the update button is pressed to write the code to the atmega8 controller. It is done as shown in Figure 6.

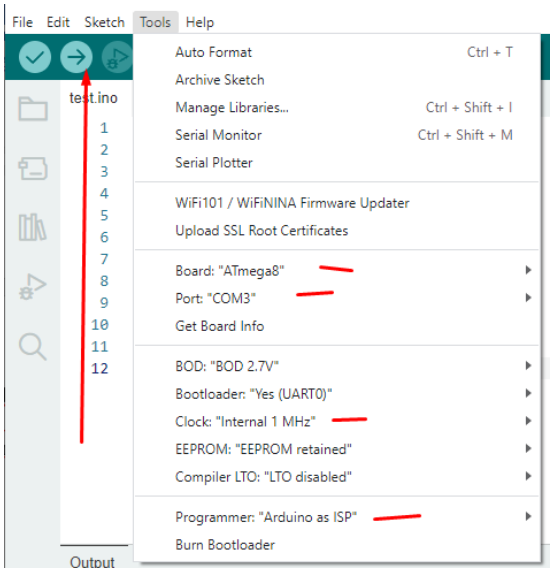

Figure 6. The process of writing code to a microcontroller

After the code is successfully executed, the reset wire on the atmega8 controller is disconnected. After that, the operation process of the microcontroller is shown in Figure 7.

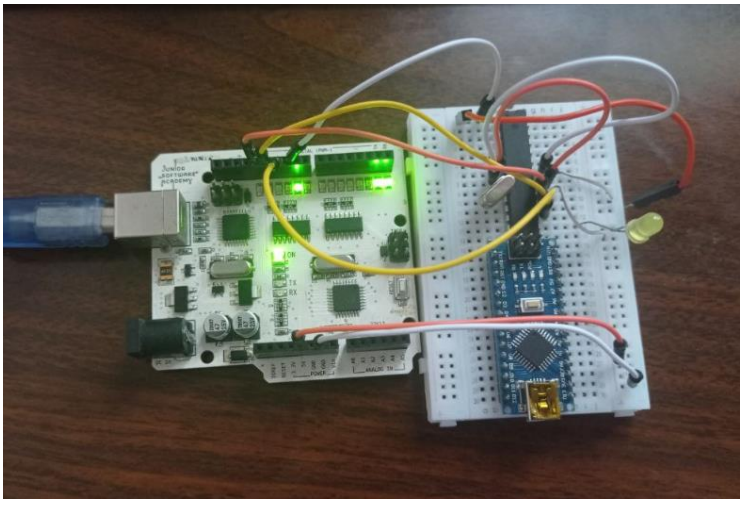

Figure 7. Atmega8 microcontroller working process

CONCLUSION

In conclusion, we can say the following. We can write hex code to atmega8 microcontroller using Arduino software. It is very easy and convenient. In this way a method of writing hex code to atmega8 microcontroller through the Arduino device.

REFERENCES:

1. M. Safdar Munira, Imran Sarwar Bajwa, Sehrish Munawar Cheema. An intelligent and secure smart watering system using fuzzy logic and blockchain // Computers and Electrical Engineering. — 2019. — № 77. — С. 109-119.
2. Mnati M J, Van den Bossche A and Chisab R F 2017 A smart voltage and current monitoring system for three phase inverters using an android smartphone application Sensors.
3. K. Manjunath, “Electrical and Electronic Measuring Instrumentation”, Falcon Publishers, 1st Edition, 2017.
4. M M S, Borre J D and Santos J C 2014 Design and Construction of a Power Meter to Optimize Usage of the Electric Power International Congress of Engineering Mechatronics and Automation (CIIMA) pp 1–5.
5. Abubakar I, Khalid S N, Mustafa M W, Shareef H and Mustapha M 2017 Calibration of ZMPT101B voltage sensor module using polynomial regression for accurate load monitoring ARPN J. Eng. Appl. Sci. 12 1076–84.
6. Breithardt G., Becker R., Seipel L., Abendroth R-R, Ostermeyer J. Non- invasive detection of late potentials in man - a new marker for ventricular tachycardia. // Europ.Heart J.-1981.-V. 2.-P. 1-11.
7. Berdanov U. A, Khujayorov I.Sh, Abdurashidova K.T., Salimova Kh.R., Musadjanova D.A.Using Artificial Intelligence Algorithms for Speech // Therapy SystemsInternational Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075 (Online), Volume-9 Issue-7, May 2020, Page No. 821-825.
8. S. Ahmad, ―Smart Metering and home automation solutions for the next decade, in International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2011, pp. 200–204.
9. M. Weiss, F. Mattern, T. Graml, T. Staake, and E. Fleisch, ―Handy feedback: Connecting Smart Meters with mobile phones,‖ in 8th International Conference on Mobile and Ubiquitous Multimedia, Cambridge, United Kingdom, Nov. 2009.