DOI https://doi.org/10.30525/978-9934-588-47-1.2

A MODERN TECHNOLOGY OF TEACHING AND TESTING MICROCONTROLLER SYSTEM DESIGNERS

Bondarenko V. E.

INTRODUCTION

Microcontrollers (MC) are computers executed on one microchip. They are used in many spheres of human activity. Microcontrollers are intended for management of various electronic devices. The interaction between units of a micro-controller is made according to a program. Such a program is incorporated in the microcontroller. Unlike microprocessors of personal computers, microcontrollers contain built-in additional devices. These devices carry out tasks under the control of the microprocessor nucleus of the microcontroller.

An increased demand for microcontroller systems requires training of design systems experts. This problem was discussed in[1] which described the training platform of the PIC (Programmable Interface Controller) microcontroller, made by the Microchip Technology Incorporation. Hedle[2] discusses a course in microcontroller system design which can help students to improve learning results.

It helps students to develop knowledge and technical skills in microcontroller system design. Hardware was developed for the laboratory using the Motorola 68HC11 microcontroller. The group of authors[3] share their experience of teaching microcontroller system design with the help of the SK40C PIC starter kit. General problems of

[1] Hsiung S., Ritz J., Jones R., Eiland J. Design and Evaluation of a Microcontroller Training System for Hands-on Distance and Campus-Based Classes. Journal of Industrial Technology. 2010, v. 26, № 4, p. 1–10.

[2] Hedley M., Barrie S. An Undergraduate Microcontroller Systems Laboratory. IEEE Transactions on Education , 1998, v. 41, № 4, p. 17–25.

[3] Che Fai Yeong, Hisyam A. R., Su E., Lee Ming A Hands-on Approach To Teaching Microcontroller. *Systemics, Cybernetics and Informatics,* 2013, v. 11, № 1, p. 55–59.

technology of satellite and mobile communication in modern distance education are discussed in[4,5].

Analysis shows that existing works only partially address the microcontroller system design problem.

Therefore, this article features a technology of teaching microcontroller system designers. This technology is based on a Web-site, which is used for receiving theoretical knowledge. It uses a simulator which can help to project variants of a system and its software.

The technology uses the hardware (Arduino Uno), which allows students to construct real microcontroller systems. The final stage of design is development of a project printed-circuit board.

This technology can improve education quality**,** thanks to more effective reflecting of all sides of microcontroller system design.

In view of the aforesaid, this paper describes a technology of teaching microcontroller system design. This educational technology can be applied to all forms of training. However, it is particularly helpful for part-time and distance forms of learning. Such students do not have stable contact with teachers. As usual, they are located at large distances from an educational institution. However, the stable contacts of students with the tutors determine the quality of education.

This paper is organized as follows:

Section 1 is an introduction outlining the problem of teaching microcontroller systems designers. Section 2 describes the structure of a training complex, which consists of several subsystems that guarantee quality of the students' training as microcontroller system designers. Section 3 contains the main results of the paper. It describes the technology of expert teaching of micro-controller system design. Section 4 describes in detail seven projects which allow students to receive practical skills in microcontroller system designing. Section 5 presents the conclusion and the main results of the paper.

1. The structure of training complex

This technology is based on the use of a training complex. It allows students to study the discipline "Designing of microcontroller systems"

[4] Bondarenko V.E. Technology of Satellite and Mobile Communication in Modern Distance Education. *Second International Conference Modern* (e -) Learning, July, 2007, Bulgaria. Proceedings, ITHEA, Sofia, 2007, c. 120–127.

[5] Bondarenko V.E. Mobile Communication Technology as a Tool of Educational Process. *Information Technology and Knowledge,* 2007, v.1, № 1, p. 78–80.

in stationary and remote modes. This is useful especially for the students of distance courses. The general structure of the training complex is shown in Figure 1.

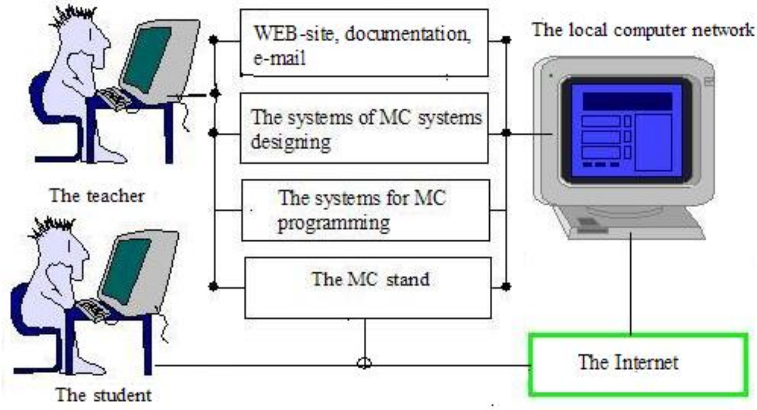

WEB-site, documentation, e-mail
The local computer network
The systems of MC systems designing
The systems for MC programming
The MC stand
The teacher
The student
The Internet

Figure 1. General structure of the training complex

As it is shown in Figure 1, the training complex consists of several subsystems. Each of the subsystems forms a certain technological stage.

1. The first stage allows students to see Web-site with theoretical papers. They can find the literature about the designing of microcontroller systems. They can receive a tutor's consultations by email and Skype. Finally, students receive tasks for designing.

2. The second stage is projection of the circuit of the received task. Such projection is made with the help of the core ISIS of the Suit Proteus Professional Version 8.6 (the firm Labcenter Electronics). The example of such projection is shown in Figure 2.

3. The third stage is necessary for designing the software which provides the correct work of the projected system. All software is written in the programming language C. This software is projected with the help of the integrated development environment CodeVisionAVR Version 3.12 (the firm HP Info Tech). A result of such projection is the program which can be written in the microcontroller memory of the circuit that was projected at stage 2.

4. The fourth stage is the simulation stage. It is based on stages 2 and 3. A student can simulate on the computer the functioning of the

projected microcontroller system. They can estimate the quality of the project in different modes.

5. The fifth stage allows students to realize the project on the stand (Arduino Uno) with the real equipment, where students can investigate the quality of the project functioning.

6. The sixth stage gives possibility to make a printed board for the projected system. Such a printed board is projected with the help of the core ARES of the Suit Proteus Professional Version 8.6.

2. The technology of the work with the training complex

First, students study the theoretical material about the AVR-microcontrollers. Such documentation is placed on the system web-site.

Further, students receive the tasks of growing complexity. The tasks will allow students to receive professional skills in microcontroller systems designing. These tasks are formulated in seven laboratory works. The laboratory works allow students to comprehend most aspects of the microcontroller system design. The student receives a task for designing (stage 1).

They develop the basic scheme of the projected device, using the core ISIS of the Suit Proteus Professional Version 8.6 (the firm Labcenter Electronics) (stage 2).

The proposed alarm system can be used for a bike or scooter. It is assembled on the AVR ATmega8 microcontroller. An example of such a scheme of a sound alarm system, which was worked out on the basis of the microcontroller ATMega8, is presented in Figure 2.

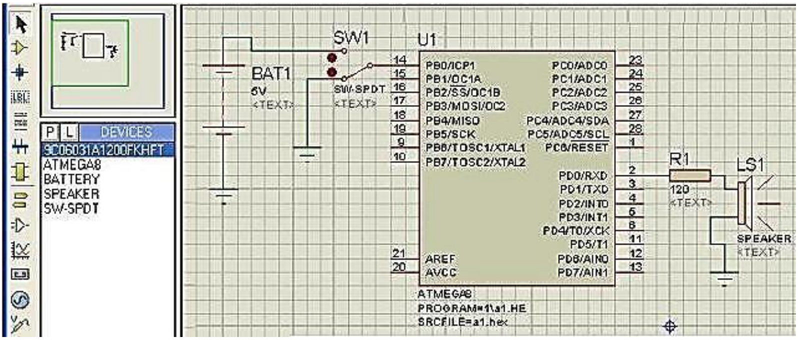


Figure 2. The scheme of the alarm system

Next, the student develops the program for the microcontroller (stage 3). The program is made with the help of the integrated environment CodeVisionAVR Version 3.12 (the firm HP Info Tech). This program ensures the solution of necessary tasks in the projected microcontroller system.

The program of the speaker sound has the following form. The smaller delay in the ***delay_ms*** function, the higher the sound frequency. This program is written in C language.

```
#include <mega8.h>
#include <delay.h>
void main(void)
{ DDRD=0xFF;
while (1)
{  if(PINB.0)  // The SW1switch check
{PORTD=0b11111111;
delay_ms(1);  //Sound frequency control
PORTD=0b00000000;
delay_ms(1);  //Sound frequency control
}}
```

The program is connected to the microcontroller project in the core ISIS of the Suit Proteus Professional, and the modelling of the projected system is carried out (stage 4). If the functioning of the model does not correspond to the technical requirements, the student comes back to the design of the basic scheme (stage 2), or to the development of the program and carries out necessary updating (stage 3). If a student feels the necessity to receive additional theoretical knowledge, they go back to stage 1. This process is repeated until the project model meets the technical requirements.

Further, the projected system is realized on the stand (Arduino Uno) with the real equipment (stage 5)**,** to investigate the quality of its functioning. Such realization will be shown in the next section. If the project does not work according to the requirements, it is carried out the transition to the above described design stages (stages 1–4). Where it is carried out: 1) reading of theoretical papers, 2) the updating of the scheme, 3) the input of the changes in the program, 4) modelling of the new variant of the project, 4) realizing of a new variant of the project on the stand.

When the project works adequately to the requirements, it is carried out the designing of a printed-circuit board for projected system (the stage 6). Such a printed-circuit board is projected with the help of the core ARES of the Suit Proteus Professional Version 8.6. The core ARES uses a scheme of a system which was projected with the help of the core ISIS (the stage 2).

3. Development of laboratory works based on the platform Arduino Uno

3.1. Objectives for the practical works

The stand for seven laboratory works uses the following equipment:

the Arduino Uno, a breadboard, a sensor module hydrocarbon gases and smoke (MQ-2), a sensor temperature and humidity (DHT-11), the LCD 16×2 with the I2C module, a module real-time clock (DS3231), an infrared receiver, a remote controller, a buzzer (speaker), LEDs, resistors, connecting wires.

All necessary equipment is shown in Figure 3.

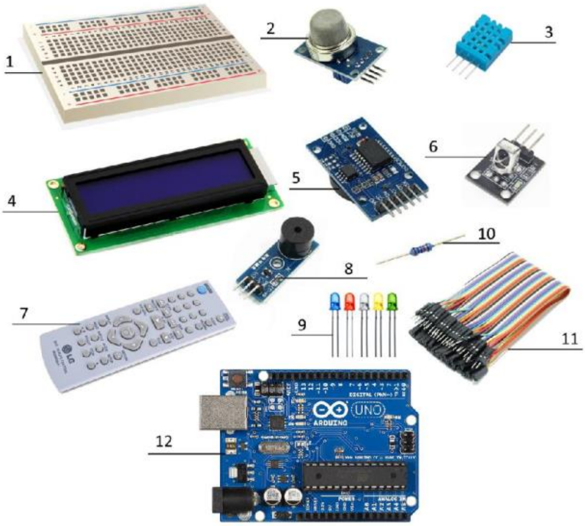


Figure 3. Equipment, which is needed for laboratory works: a breadboard (1), a gas sensor MQ-2 (2), a temperature and humidity sensor DHT-11 (3), an LCD module (4), a real-time clock (5), an infrared receiver (6), a remote controller (7), a buzzer (8), LEDs(9), resistors (10), connecting wires (11), the Arduino Uno (12)

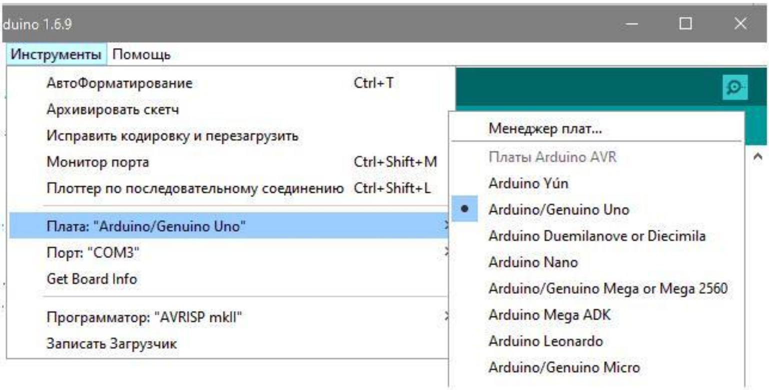
3.2. Laboratory work 1

This work has several goals: 1) Familiarization with the board Arduino Uno. 2) The installation of the necessary software. 3) Design of the first project which includes a circuit with the LEDs and C-program. This program allows the blinking of the LEDs.

The equipment of this work is: the Arduino Uno board, resistors, LEDs, connecting wires.

To get started with the Arduino Uno board, students download and install the software Integrated Development Environment (IDE). The software is on the official website http://www.arduino.cc/en/Main/Software.

We install the IDE**,** and it is launched. Further**,** we choose the Arduino Uno board (Figure 4) and the COM3 port (Fig. 4.3), to which it connects.


duino 1.6.9
Инструменты
Помощь
АвтоФорматирование Ctrl+T
Архивировать скетч
Исправить кодировку и перезагрузить
Монитор порта Ctrl+Shift+M
Плоттер по последовательному соединению Ctrl+Shift+L
Get Board Info
Записать Загрузчик
Менеджер плат...
Платы Arduino AVR
Arduino Yún
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino/Genuino Micro

Плата: "Arduino/Genuino Uno"

Порт: "COM3"

Программатор: "AVRISP mkII"

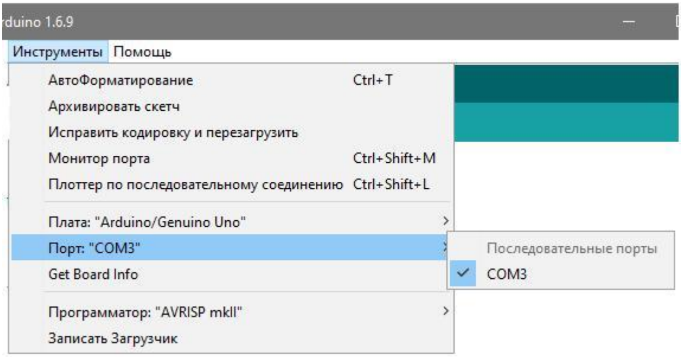
● Arduino/Genuino Uno

Figure 4. The choice of the working board Arduino Uno in the IDE

The essence of the first laboratory work is to connect three LEDs and to write a sketch (C-program) that will light the LEDs alternately at intervals of five seconds.

You can keep in mind that LEDs have two contacts («+» and «-»).

A resistor needs to be set between the LED and the voltage source to prevent combustion of the LED.


rduino 1.6.9
Инструменты
Помощь
АвтоФорматирование Ctrl+T
Архивировать скетч
Исправить кодировку и перезагрузить
Монитор порта Ctrl+Shift+M
Плоттер по последовательному соединению Ctrl+Shift+L
Get Board Info
Записать Загрузчик
Последовательные порты

Плата: "Arduino/Genuino Uno"

Порт: "COM3"

✓ COM3

Программатор: "AVRISP mkII"

Figure 5. The choice of the working port (COM3) of the Arduino Uno in the IDE

The contact «+» will be connected with the information port of the Arduino Uno which has voltages from 0 to + 5V through a resistor to control the power supply of the LED. The contacts «-» of the LEDs should be connected with the common rail «-». Connect in such manner: the green LED to the port nine, yellow LED to the port 10 and red LED to the port 11.

Assemble the circuit as shown in Figure 6.

In order to light the LED**,** it is sufficient to apply the voltage (+5V) to the port to which this LED is connected. For this we should determine the port position (*OUTPUT*) for information transfer with the help of function "***pinMode (port_number, OUTPUT)***" and then submit to this port the voltage (+5 V) with the help of the function "***digitalWrite (port_number, HIGH)***".

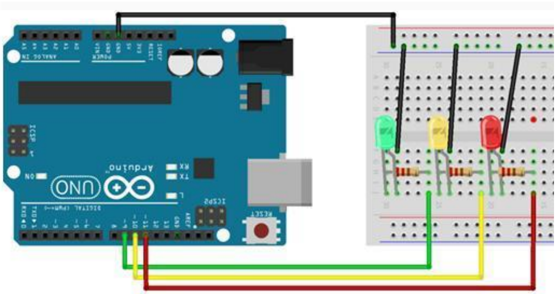


Figure 6. The circuit of laboratory work 1

The LED can be switched off with the help of the function “***digitalWrite (port_number, LOW)***”.

Now we write the sketch (C program) and upload it to the project.

```
void setup () {
Serial.begin (115200)
// Set the speed of information exchange with Arduino
pinMode (9, OUTPUT);
// Set the port 9 to position of information transfer
pinMode (10, OUTPUT);
// Set the port 10 to the position of information transfer
pinMode (11, OUTPUT);
// Set the port 11 to the position of information transfer
}
void loop () {
digitalWrite (11, LOW); // The red LED turns off
digitalWrite (9, HIGH); // The greenLED lights
delay (5000); // delay 5 seconds

digitalWrite (9, LOW); // The green LED turns off
digitalWrite (10, HIGH); // The yellow LED lights
delay (5000); // delay 5 seconds
digitalWrite (10, LOW); // The yellow LED turns off
digitalWrite (11, HIGH); // The red LED lights
delay (5000); // delay 5 seconds
}
```

After downloading this sketch and the system activation, the LEDs will alternatively glow. First, the green LED lights up (Figure 7), the yellow LED will glow after five seconds, then the red LED will glow after next five seconds.

The objective of this work is to learn how to work with the sensor DHT-11, to implement data reading of the current temperature and humidity, to bring this information to the monitor of the serial port.

The equipment of this work consists of the equipment of the previous laboratory work, a temperature and humidity sensor DHT-11, resistors and connecting cables.

The work with additional modules and sensors demands libraries. We need a library for work with the temperature / humidity sensor which is used the laboratory work. This library can be found on various websites and forums.

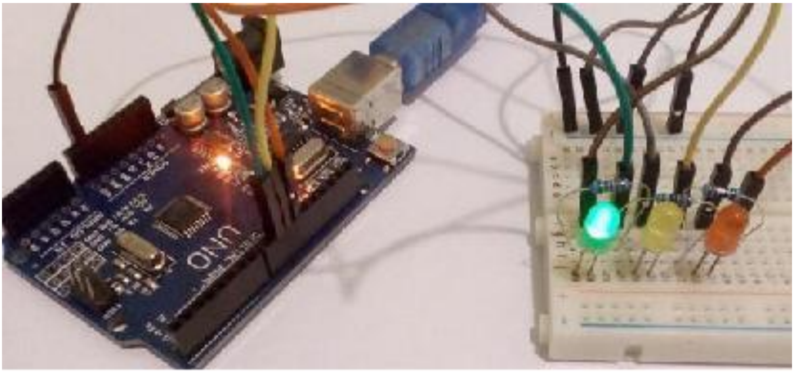


Figure 7. The demonstration of the green LED glows

3.3. Laboratory work 2

We should add the library to our IDE after loading the library.

The attaching **of** the library to the sketch is made with the line: ***#include <stDHT.h>***.

Now we connect the sensor. It has four contacts, but we will use only the first, second and fourth. The first contact is connected to the bus voltage (+ 5V). Contact two is an informational contact. It is connected directly to the second port of the board Arduino Uno. The fourth contact is connected to the voltage bus (- ground connection).

To prevent deterioration of the sensor, between the + 5V voltage contact on the sensor output and information output**,** you need to install the resistor.

The LEDs that were used in the previous laboratory work will signal the surrounding temperature. The green LED lights up if the temperature is below 20 degrees. The yellow LED will switch on if the temperature is between 20 and 30 degrees. And the red LED will switch on if the temperature is above 30 degrees.

The current temperature can be obtained by using the function "***sens.readTemperature (port_number)***". The humidity can be found with the help of the function "***sens.readHumidity (port_number)***".

Assemble the circuit as shown in Figure 8.

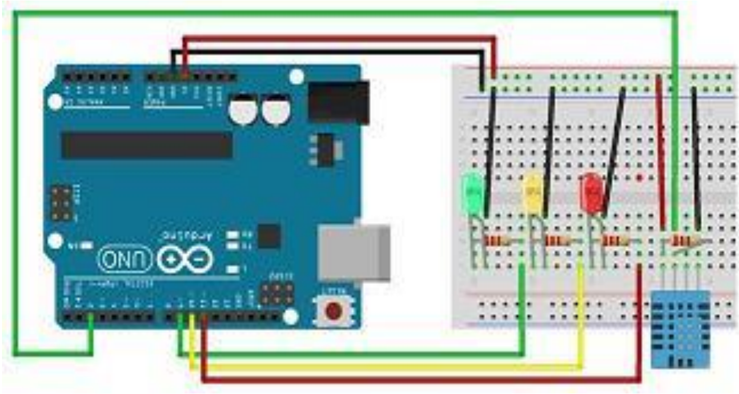


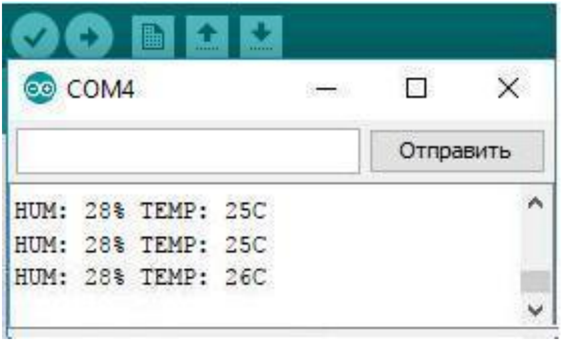
Figure 8. The circuit of laboratory work 2

We turn to the writing of the sketch:

```
#include <stDHT.h> // The library uses with the probe DHT-11
DHTsens (DHT11); // Specify the type of temperature sensor
void setup () {
Serial.begin (115200)
// Set the speed of information exchange with Arduino
pinMode (2, INPUT);
// Set the port 2 receiving position information
pinMode (9, OUTPUT);
// Set the port 9 to position information transfer
pinMode (10, OUTPUT);
// Set the port 10 to the position information transfer
pinMode (11, OUTPUT);
// Set the port 11 to the position information transfer
digitalWrite (2, HIGH);
// Serve a voltage on the port 2
}
void loop () {
int t = sens.readTemperature (2) - 2;
// The variable to store the current temperature
int h = sens.readHumidity (2);
// The variable to store the current humidity
// The green LED lights if the current temperature is below of 20 degrees
if (t <20) {
digitalWrite (9, HIGH);
}
else {
digitalWrite (9, LOW);
}

// The yellow LED lights if the current temperature is from 20 to 30 degrees
if (t> = 20 && t <30) {
digitalWrite (10, HIGH);
}
else {
digitalWrite (10, LOW);
}
The red LED lights if the current temperature is above 30 degrees
if (t> = 30) {
digitalWrite (11, HIGH);
}
else {
digitalWrite (11, LOW);
}
// The output information collected on the Serial Port Monitor
Serial.print( "HUM:");
Serial.print (h); // Output humidity
Serial.print( "%");
Serial.print( "TEMP:");
Serial.print (t); // Output Temperature
Serial.print( "C");
Serial.print( "\ n");
}
```

We download the sketch and observe data of humidity and temperature on the monitor (Fig. 9).



```
HUM: 28% TEMP: 25C
HUM: 28% TEMP: 25C
HUM: 28% TEMP: 26C
```

Figure 9. The monitor of the serial port

3.4. Laboratory work 3

The objective of this work is the connection of the LCD, to show information collected by sensor DHT-11.

This work needs the following equipment: equipment of the previous laboratory work, an LCD size 16×2 with the I2C module, connecting wires.

ATmega microcontrollers support the I2C interface. The SDA data line on the board Arduino is connected to analog port 4**,** and the clocking SCL line is connected to analog port 5. The Arduino has the regular library ***"Wire"*** for the work with the I2C protocol.

The I2C module allows us to control devices that require an 8-bit bus with the help of just two contacts using the protocol I2C. Thanks to this module, the quantity of ports which are needed to connect the LCD to Arduino **is** reduced from 16 to 4.

We should download and connect the library ***"LiquidCrystal_I2C"*** to interact with the LCD via the I2C module.

The sketch must define the LCD address and its dimension. This is done by using the function "***LiquidCrystal_I2C lcd (0x3F, 16,2)***", where 0h3F is the display address and 16,2 is the LCD size. After this, the display is initialized by the function "***lcd.init ()***" and turn**s** on the lights using function "***lcd.backlight ()***".

We should put the cursor of the display to the necessary position using the function "***lcd.setCursor (position in row, position in column)***" before the information output. Output symbols are written on display with the help of function "***lcd.print ()***".

Typically, such displays have a limited number of characters available for output. We can do new characters with the help of an array of the type “byte” and the function “***lcd.createChar ()***”.This array will meet the desired pixels on the display (1 in needed bits).

We can write the Celsius degree symbol (°) as such array *degree* in the sketch:

```
bytedegree [8] =
{ 0b01100,
0b10010,
0b10010,
0b01100,
0b00000,
0b00000,
0b00000,
0b00000
};
```

General connection diagram is shown in Figure 10.

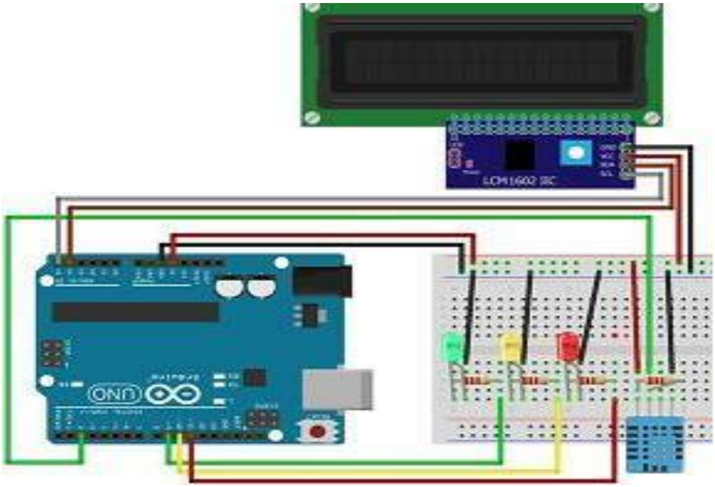


Figure 10. The circuit of laboratory work 3

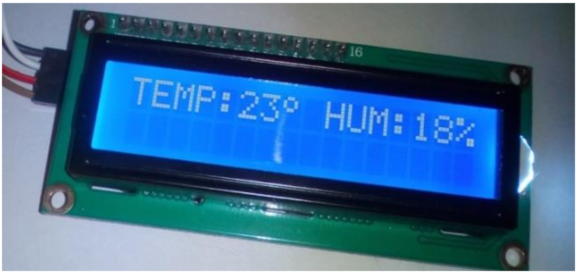
The sketch is shown below.

```
#include <stDHT.h>
// The library for using with temperature and humidity sensor
```

```
DHT sensor (DHT11);
// Specification of the type of temperature sensor
#include <Wire.h>
// The library for work with I2C module
#include <LiquidCrystal_I2C.h>
// The library for work with display
LiquidCrystal_I2Clcd (0x3F, 16,2);
// Specification of the address and the size of the display
bytedegree [8] =
{
0b01100,
0b10010,
0b10010,
0b01100,
0b00000,
0b00000,
0b00000,
0b00000
};
void setup () {
Serial.begin (115200)
// Setting the speed of information exchange with Arduino
pinMode (2, INPUT);
// Settingof the port 2 for receiving information
pinMode (9, OUTPUT);
// Setting the port 9 for information transfer
pinMode (10, OUTPUT);
// Setting the port 10 for information transfer
pinMode (11, OUTPUT);
// Setting the port 11 for information transfer
digitalWrite (2, HIGH);
// Serve voltage on port 2
lcd.init ();
// Initialize display
lcd.backlight ();
// Enables backlight display
}
void loop () {
int t = sens.readTemperature (2) - 2;
// variable to store the current temperature
```

```
int h = sens.readHumidity (2);
// The variable for storing the current humidity
// The green LED lights if the current temperature is below 20 degrees
if (t <20) {
digitalWrite (9, HIGH);
}
else {
digitalWrite (9, LOW);
}
// The yellow LED lights if the current temperature is 20 to 30 degrees
if (t> = 20 && t <30) {
digitalWrite (10, HIGH);
}
else {
digitalWrite (10, LOW);
}
//The red LED lights if the current temperature above 30 degrees
if (t> = 30) {
digitalWrite (11, HIGH);
}
else {
digitalWrite (11, LOW);
}
// Output collected information on the display
lcd.setCursor (0, 0);
// Set the cursor to the zero position
lcd.print ("TEMP:");
lcd.print (t);
// Output temperature
lcd.createChar (1, degree);
lcd.setCursor (7, 0);
lcd.print (char (1));
// Create the degree symbol and its output
lcd.print ( "HUM:");
lcd.print (h);
lcd.print( "%");
// Output of information collected on the Serial Port Monitor
Serial.print( "HUM:");
Serial.print (h);
Serial.print( "%");
Serial.print( "TEMP:");
Serial.print (t);
Serial.print( "C");
Serial.print( "\ n");
}
```

The results of this sketch are shown in Figure 11. The title TEMP shows the current temperature. The title HUM shows the level of humidity.



TEMP:23° HUM:18%

Figure 11. The results of laboratory work 3

3.5. Laboratory work 4

The objective of this work is the connection of the gas sensor MQ-2 to the Arduino Uno board. We should connect one red LED that will indicate a high concentration of hydrocarbon gases and smoke in the air. The projected system will give to the LCD information about the current level of the gas concentration.

This work needs the following equipment: the equipment of the previous laboratory work, a gas sensor module (MQ-2), a LED, connection wires.

A student should connect the module using these contacts: the first (the voltage + 5V), the second (ground) and the fourth (analog information). The fourth contact is connected to analog port A1.

We should load the library "***TroykaMQ***" for work with this module. The operator of the microcontroller program "***#define PIN_MQ2 port_number***" specifies the port number to which the information contact of the sensor is connected. The function "***MQ2 mq2 (PIN_MQ2)***" creates the object for use with gas sensors. We should transmit to this object the port number. The function “***pinMode (port_number, INPUT)***” sets the number of the information reception port.

The module *MQ-2* must be calibrated in order to receive correct values while scanning information from the environment. Such calibration is made in two stages. First we use the function “***mq2.heaterPwrHigh ()***” for heating of the probe**,** and then values are calibrated using the function “***mq2.calibrate ()***”.

The calibration takes a minute, so the screen can display a primitive animation which is determined by such code:

```
lcd.setCursor (3, 0);
lcd.print( "LOADING");
delay (1000);
lcd.print( ".");
delay (1000);
lcd.print( ".");
delay (1000);
lcd.print( ".");
delay (1000);
lcd.setCursor (10, 0);
lcd.print ( "");
```

We can use the function “***analogRead (port_number)***” for determination of the gas concentration level in the air.

The LED will indicate a high concentration of gas and smoke. It is connected to port 6. If the gas concentration exceeds 300 units, the LED switches on.

Assemble the circuit as shown in Figure 12.

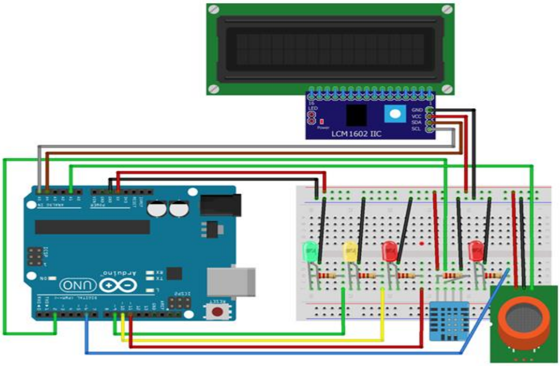


Figure 12. The circuit of laboratory work 4

The results of the system work are shown in Figure 13.

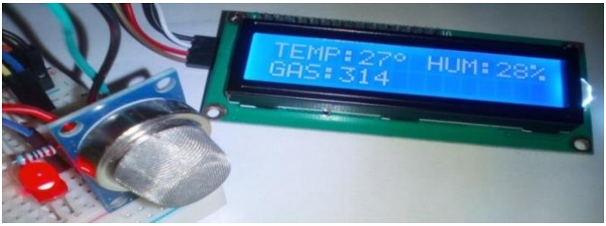


Figure 13. The results of laboratory work 4

3.6. Laboratory work 5

The objective of this work is to connect a real-time clock, set it up and display the current time.

This work needs the following equipment: the equipment of the previous laboratory work, the Real Time Clock Module (DS3231), connecting wires.

The Real-time clock module is self-powered by batteries. This allows you to store data for a long time without an external supply of the module. The DS3231 uses the first four contacts to connect. Connect the **GND** contact to the ground**,** the **VCC** contact should connect to +5**V**, **SDA** and **SCL** contacts are connected to the ports on Arduino Uno.

We must download library "***iarduino_RTC***" for using the module DS3231. Determination of the module type is made with the help of the function "***iarduino_RTC time (RTC_DS3231)***". The clock starts with this function "***time.begin ()***". The first time you must set the time. This is done by using the function "***time.settime (seconds, minutes, hours)***". After successfully setting the clock, you must remove or comment out this function.

The results of this system work are shown in Figure 14.

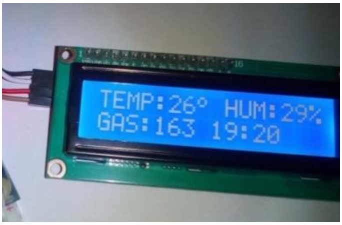


Figure 14. The results of laboratory work 5

3.7. Laboratory work 6

The objective of this work is to connect an infrared (IR) sensor so as to enable setting the clock from the remote control.

This work requires the following equipment: equipment from the previous laboratory work, an infrared sensor, a remote controller, connection cables.

The IR module uses all three contacts to the connection. The first contact connects with the ground, the second contact connects with the voltage +5V**,** and the third contact connects to port 3.

It is possible to control the Arduino Uno via a remote controller. You need to know remote key codes in the hexadecimal form. We can find them online if a remote controller is from the known brand. In our case, the type of the remote controller is LG AKB33659511. The codes of the main keys for the remote control are given.

We must download library “***IRremote***” to use the infrared module. The function “***IRrecvirrecv (port_number)***” defines a port that is connected to the module.

3.8. Laboratory work 7

The objective of this work is to connect a buzzer (speaker) for an audio alert about the high gas level. The alarm project will enable the alert at the time which will be set with the help of the remote controller.

This work needs the following equipment: the equipment from the previous laboratory work, the buzzer, connecting wires.

The buzzer (speaker) uses three contacts for connection with the system. The first contact connects to the ground**.** The second contact receives information from port 9, from which a melody will be transmitted. The third contact is connected to the voltage +5V.

We should load library ***“NewTone”*** for using the speaker. The function “***NewTone (port_number, pitch, duration)***” makes sounds.

We can take the example code of the manual clock setting of the previous laboratory work to create an alarm.

Figure 15 shows the process of setting up the alarm clock from the remote controller.

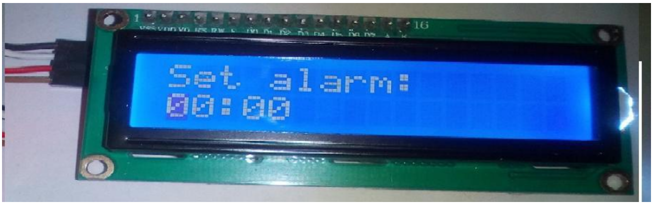


Figure 15. The alarm clock setting

It is necessary to design the printed-circuit board of the project after a successful final test of the project. The scheme designed in the core ISIS is transferred to the core ARES. The optimal tracks of the printed-circuit board will be designed with the help of the core ARES of the Suite Proteus Professional. Development of the project is completed at this stage.

However, the realization of the project is possible on the next levels of teaching (a course project design or a bachelor degree project design).

The considered technology is intended not only for full-time students. Such students can work directly in their university laboratories. It can be used in remote training, which is especially important for distance students.

The Arduino Uno board is not expensive, and all students can have such a microcontroller board at home. Internet communication (Skype, E-mail) with a tutor can help to recognize and solve technical problems.

CONCLUSIONS

The technology of remote microcontroller study was developed at the Kiev International University. So far, the implementation of this technology has been carried out only partially. Manufacturing of a printed-circuit board has not been realized yet**,** and it is given as a home task for advanced students.

This technology is used for the preparation of specialists of various forms of training. However, we observe the efficiency of the technology in the distance form of training. Distance students do not have stable possibilities of direct communication with teachers and this technology makes such communication more active.

The implementation of this technology has resulted in an increase in students’ grades by 20% on average. With distance students, the increase was 27% on average. In the future, the technology will be further developed for training and testing in the disciplines Computer Electronics and Computer Circuit Engineering.

SUMMARY

This paper describes a teaching technique of microcontroller system designing. Designing of microcontroller systems demands theoretical knowledge as well as practical experience. Therefore, this technique has several stages.

First, students receive theoretical knowledge of basic principles of microcontroller system designing. Students can receive this knowledge from the university WEB-site. There are many multimedia courses. They explain in what manner students can use equipment and software. Further, students learn the C-programming for AVR microcontrollers. The next stage of the learning is working out a project. A project has a system circuit and C-programs which support the microcontrollers functions. Further, a project is simulated with the help of the ISIS subsystem of the Proteus simulator. Such simulation helps students to see the quality of the project functioning. They can look through several variants of the project and select the optimal variant of a circuit and a C program. The next stage of designing projects is their realization with the help of real equipment. The authors propose a technique of receiving practical experience in microcontroller system design. This technique is based on the microcontroller board Arduino Uno. The Arduino Uno is a microcontroller board which uses the ATmega328P microcontroller. The Arduino Uno allows students to construct real microcontroller devices. A real project should be realized on a printed-circuit board. Such a printed-circuit board is projected in the ARES subsystem of the Proteus simulator. Seven laboratory works of different complexity developed by the authors lead students step-by-step to receiving professional skills. These works allow students to receive the first experience in microcontroller systems designing. Besides teaching, this technique has tests which can estimate students' practical skills. This technique allows students to do a lot of projects of various complexity. It will give students a deeper experience in the designing of microcontroller systems.

REFERENCES

1. Hsiung S., Ritz J., Jones R., Eiland J. *Design and Evaluation of a Microcontroller Training System for Hands-on Distance and Campus-Based Classes.* Journal of Industrial Technology. 2010. v. 26. № 4. P. 1–10.

2. Hedley M., Barrie S. An Undergraduate Microcontroller Systems Laboratory. *IEEE Transactions on Education.* 1998. v. 41. № 4. P. 17–25.

3. Che Fai Yeong, Hisyam A.R., SuE., Lee Ming A Hands-on Approach To Teaching Microcontroller. *Systemics, Cybernetics and Informatics*. 2013. v. 11. № 1. P. 55–59.

4. Bondarenko V.E. Technology of Satellite and Mobile Communication in Modern Distance Education. Second International Conference Modern (e -) Learning, July, 2007, Bulgaria. Proceedings. ITHEA. Sofia, 2007. c. 120–127.

5. Bondarenko V.E. Mobile Communication Technology as a Tool of Educational Process. *Information Technology and Knowledge*. 2007. v. 1. № 1. p. 78–80.

Information about author:
Bondarenko V. E.,
Doctor of Technical Sciences, Professor,
National University of Life and Environmental Sciences of Ukraine
15, Heroiv Oborony Str., Kyiv, 03041, Ukraine