

A practical approach for introductory embedded systems teaching

Peter Yakimov

***Abstract.** In order to prepare the students for the challenges of their future job the Faculty of Electronic Engineering and Technologies at Technical University of Sofia accepted a new curriculum for Bachelor degree in Electronics which was created after many iterations and discussions with the partners companies and employers organisations. In this paper a part of the course programme in “Practice on open source platforms programming” is presented. The course gives the students theoretical knowledge about embedded systems programming and practical skills in this field. Some examples from the laboratory work that represent the practical approach for introductory teaching in embedded systems programming are given. The practical approach is useful for the beginners. With emphasizing on the practice rather than on the academic theory they accept the material easier. The open source hardware and software - Arduino development board and the software IDE are very useful for the introductory education.*

***Практически подход при начално обучение по вградени системи (Петър Якимов).** За да подготви студентите за предизвикателствата на бъдещата им работа Факултетът по електронна техника и технологии на ТУ-София прие нов учебен план за ОКС „Бакалавър” по специалност „Електроника”, който беше създаден след много обсъждания и разговори с фирмите партньори и организациите на работодателите. Тази статия представя част от учебната програма на дисциплината “Практикум по програмиране на платформи с отворен код”. Дисциплината дава на студентите теоретични знания за програмиране на вградени системи и практически умения в тази област. Представени са някои примери от лабораторните упражнения, които илюстрират практическия подход за начално обучение в областта на програмирането на вградени системи. Практическият подход е полезен за начинаещите. С акцент върху практиката, а не върху академичната теория те приемат материала по-лесно. Платформите с отворен код - системата за развитие Arduino и съответната програмна среда са много подходящи и полезни за началното обучение.*

Introduction

Embedded systems are affecting human lives in many ways. Examples can be found in alarm clocks, automobiles, mobile phones, personal digital assistants, etc. [1]. Embedded systems are also found in cars, airplanes, and robots. They far outnumber traditional computers (which also contain embedded processors) and it is estimated that there will be thousands of embedded devices per person by 2015. Learning to design and program embedded systems is a critical skill that is necessary for many industry and scientific jobs [2]. At the level of today’s technology consumer, there appears to be an increasing desire to interface our technological power-machines to the real

physical world. And power-machines they are, our personal computers, tablets and smart phones – equipped with highly advanced man-machine interaction technologies, communication possibilities, location-determining hardware, acceleration sensors, and more. However, for all their strengths and possibilities, they do not offer the connectivity to the physical world around us that many dream of [3]. So there is a need of a physical interfacing which forced development of projects like Wiring (www.wiring.org.co) and Arduino (www.arduino.cc) that offer immensely popular tools for lower- to intermediate-level software and hardware design. This affects on science and education. Scientists

increasingly use publicly available low-cost digital prototyping systems to create measurement tools and other experimental devices. To witness, a Google Scholar (scholar.google.com) query for articles containing the word “Arduino” in their title, excluding legal document, patents and citations, yielded a result of 490 scholarly articles (query result on September 2, 2013) [3]. The major benefits for using Arduino in an educational setting are [4]:

- Ease of setup – plug and play;
- Many examples for controlling peripherals – preloaded in the IDE;
- Many open source projects to look at;
- Works on Windows, Linux, and Mac;
- Low cost hardware – build or purchase prebuilt;
- Low cost software – free;
- Low maintenance cost;
- Students can prototype quickly;
- Can be programmed in a number of languages including C.

Motivation for the practical approach application

After many iterations and discussions with the partners companies and employers organizations the Faculty of Electronic Engineering and Technologies at Technical University of Sofia accepted a new curriculum for Bachelor degree in Electronics. In the process of preparation lots of curricula from many foreign universities with traditions in education in electronics were investigated. It was created in order to prepare the students for the challenges of their future job. The goal was to emphasize on the embedded systems and especially on the obtaining practical skills in programming. Also it was accepted that the practical training has to begin in the first semester in order to the students to gain experience for the specialized courses in the next years. As a development environment was chosen the Arduino platform which is an open-source hardware, designed to make the process of using electronics in multidisciplinary projects more accessible.

According to the philosophy of the new curriculum a brand new course entitled “Practice on open source platforms programming” with two hours laboratory work per week was included. One major part of it is based on the usage of Arduino. The development board OLIMEXINO-328 based on the microcontroller ATmega328P is the hardware [6]. The aim of the course is with appropriate examples the students to understand the relationship between computer devices and the surrounding world. The experimental work includes an application software writing and

debugging and measuring the response of controlled peripheral circuits. Thus, students acquire practical skills and obtain knowledge about basic electronic circuits and devices, and the possibilities for their program control. During the laboratory work experiments on development boards are carried out. Initially the given task is analyzed and the algorithm is drawn. Then a program is written and run. Thus the students individually find possible errors and after analysis the results conclusions are made and mistakes are corrected. The topics are directly related to the field of the next courses [7].

New course implementation

The very beginning of the course is dedicated to getting knowledge and skills in working with the development board and the IDE. Only a personal computer and a USB cable are needed for this.

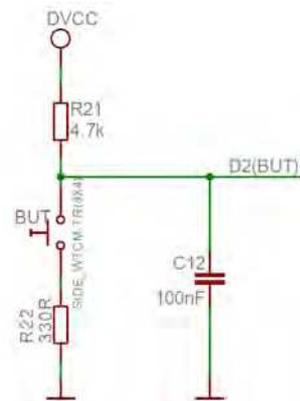


Fig.1. User button with name BUT connected to ATmega328P pin 32 (digital signal D2).



Fig.2. Light emitting diodes connected to ATmega328P pins 17 (LED1 - digital signal D13) and 13 (LED2 - digital signal D9).

Learning the examples from the preloaded in the IDE the students study the basic statements of the programming language for control of digital inputs and outputs. No external tools are needed. The on-board button and LEDs are enough. To use them the students have to know preliminary that the digital input reads logic “0” from the pressed button and to

light the LED a high level must be set from the digital output as it is shown on Fig. 1 and Fig. 2 [6]. Some code examples help to study statements related to digital inputs and outputs control are presented in [7].

With the progress of the course in order to study loops and working with arrays additional components are used. A breadboard and 7-segment LED display with current limiting resistors are enough. This exercise will help the students to understand the operation of one of the most often used indicator elements. A common cathode organization of the display (Fig. 3) is chosen in order to use the same logic levels to control its diodes like the on-board ones as it is depicted in Fig. 2.

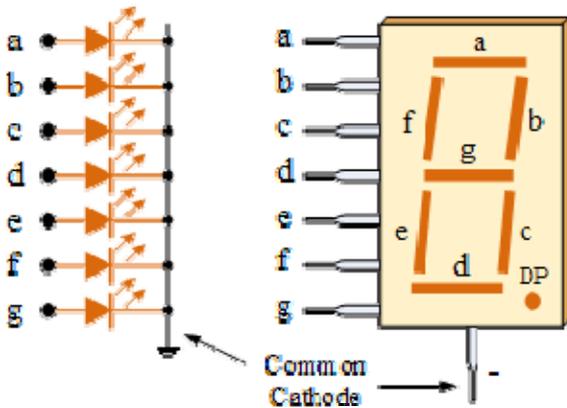


Fig.3. 7-segment LED display.

The following code example illustrates loop organization and using one-dimensional array. The program is intended to turn on successively the segments of the display with 1s delay.

```

/*Setting outputs for every segment*/
int a = 4; // segment a is connected to pin 4
int b = 5; // segment b is connected to pin 5
int c = 6; // segment c is connected to pin 6
int d = 7; // segment d is connected to pin 7
int e = 8; // segment e is connected to pin 8
int f = 9; // segment f is connected to pin 9
int g = 10; // segment g is connected to pin 10
int segment[] = {a, b, c, d, e, f, g};

void setup()
{
  pinMode(a, OUTPUT); // the pin is set as output
  pinMode(b, OUTPUT); // the pin is set as output
  pinMode(c, OUTPUT); // the pin is set as output
  pinMode(d, OUTPUT); // the pin is set as output
  pinMode(e, OUTPUT); // the pin is set as output
  pinMode(f, OUTPUT); // the pin is set as output
  pinMode(g, OUTPUT); // the pin is set as output
}
void loop()
{
  for(int i = 0; i < 7; i++)
  {
    digitalWrite(segment[i], HIGH); // turns the pin on

```

```

    delay(1000); // delay one second
    digitalWrite(segment[i], LOW); // turns the pin off
  }
}

```

The next example is slightly harder than the previous and it is intended to show the using of two-dimensional array. The purpose of the program is to display successively the numbers from 0 to 9 with 1s delay.

```

/*Setting outputs for every segment*/
int a = 4; // segment a is connected to pin 4
int b = 5; // segment b is connected to pin 5
int c = 6; // segment c is connected to pin 6
int d = 7; // segment d is connected to pin 7
int e = 8; // segment e is connected to pin 8
int f = 9; // segment f is connected to pin 9
int g = 10; // segment g is connected to pin 10
int segment[] = {a, b, c, d, e, f, g};

/*Two-dimensional array - table with 7-segment codes of
the numbers from 0 to 9*/
int pattern[10][7] = {
  {1,1,1,1,1,1,0},
  {0,1,1,0,0,0,0},
  {1,1,0,1,1,0,1},
  {1,1,1,1,0,0,1},
  {0,1,1,0,0,1,1},
  {1,0,1,1,0,1,1},
  {1,0,1,1,1,1,1},
  {1,1,1,0,0,0,0},
  {1,1,1,1,1,1,1},
  {1,1,1,1,0,1,1}
};
void setup()
{
  pinMode(a, OUTPUT); // the pin is set as output
  pinMode(b, OUTPUT); // the pin is set as output
  pinMode(c, OUTPUT); // the pin is set as output
  pinMode(d, OUTPUT); // the pin is set as output
  pinMode(e, OUTPUT); // the pin is set as output
  pinMode(f, OUTPUT); // the pin is set as output
  pinMode(g, OUTPUT); // the pin is set as output
}
void loop()
{
  for(int x = 0; x < 10; x++)
  {
    for (int y=0; y < 7; y++)
    {
      digitalWrite(segment[y], pattern [x] [y]);
    }
    delay(1000); // delay one second
  }
}

```

Because of that the world around is analog but the computers and embedded systems are digital devices of great importance is the mastering of the principles of analog-to-digital and digital-to-analog conversion. The Arduino platform gives opportunities the students to obtain practical skills in this area.

The next code example is dedicated to analog-to-

digital conversion studying. A 10k potentiometer is connected between AREF and ground and the voltage derived from its wiper is attached to the analog input A0. The connections and the possibilities of Arduino to convert analog signals are described in [7]. This example also presents the operation of the serial monitor which is used to display the information. The program reads the code which is in the range between 0 and 1023 and it is corresponding to the analog value of the input voltage. The maximum value of the code corresponds to voltage value of 5V. After calculations the digital value of the voltage is determined and displayed. The display is refreshed every second.

```
const int analogInPin = A0; // Analog input pin
long analogIn = 0; // Digital code
float analogValue = 0; // Analog input voltage value

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog input value:
  analogIn = analogRead(analogInPin);
  //calculates the analog voltage value:
  analogValue = analogIn * 5.000 / 1024;
  // print the results to the serial monitor:
  Serial.print("code = ");
  Serial.print(analogIn);
  Serial.print("\t voltage = ");
  Serial.print(analogValue);
  Serial.println("\n");
  delay (1000); // delay one second
}
```

Mastering the principles of the analog-to-digital conversion is a preliminary stage to understanding the conversion of physical quantities from the real world. For this purpose are used sensors with defined transfer characteristics. The next code example represents temperature measurement using the sensor LM35. The connection is shown on Fig. 4. The program reads an analog input pin, converts the result with + 10.0 mV/°C scale factor and prints the results to the serial monitor. The display is refreshed every second.

```
const int analogInPin = A0; // Analog input pin that the
// sensor is attached to
long analogIn = 0; // Digital code
float sensorValue = 0; //sensor analog output voltage
//value
float temperatureValue = 0; // temperature value

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
```

```
analogIn = analogRead(analogInPin);
//calculates the sensor output voltage:
sensorValue = analogIn * 5000 / 1024;
//calculates the temperature:
temperatureValue = sensorValue / 10;
// print the results to the serial monitor:
Serial.print("sensor = ");
Serial.print(sensorValue);
Serial.print("\t temperature = ");
Serial.println(temperatureValue);
delay(1000); // delay one second
}
```

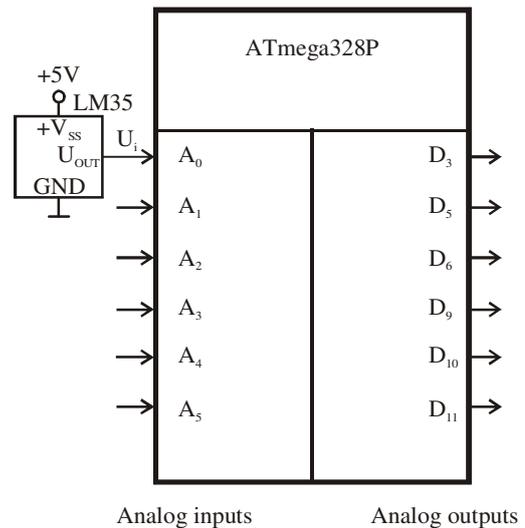


Fig.4. Temperature measurement using analog sensor LM35.

Results

After completing the laboratory work dedicated to programming using Arduino board the students obtain basic skills in using C programming language and the implementation of the program control. Also they realize the philosophy of the embedded systems and their application in all fields of the human activities and especially in the technical area. The students are given basic knowledge about the most popular indicator elements and skills to work with them. The initial explanation of the principles of the sensors and data conversion introduces the students in the world of the modern information technologies.

Conclusion

After the first approbation of the course “Practice on open source platforms programming” from the new curriculum for Bachelor degree in Electronics at the Technical university of Sofia can be concluded that the material attracts the attention of more of the students. The direct program control of simple devices like sensors and indicators convinces the students in the flexibility and the universality of the embedded systems. Emphasizing on the practice rather than

using the academic theory helps the students to accept the material easier. The open source hardware and software - Arduino development board and the software IDE offer friendly environment for beginners and are very useful for the introductory education. The students obtain knowledge and experience that will be in use in the further courses.

Acknowledgements

The paper preparation is supported by Project BG051PO001-3.1.07-0048 "Updating the curricula of specialties in FEET, FTC and MTF of TU-Sofia and create a new joint Master's degree in accordance with the needs of the labor market".

This paper was originally published in the national forum ELECTRONICS 2015, Sofia, Bulgaria, 2015.

REFERENCES

- [1] http://www.researchgate.net/publication/228408026_On_Teaching_Embedded_Systems_Design_to_Electrical_Engineering_Students.
- [2] http://cse.unl.edu/~carrick/courses/2012/236/236_2012_spring_embedded_systems.pdf.
- [3] Lamers, M.H., F.J. Verbeek, and P.W.H. van der Putten. Tinkering in scientific education. Proceedings of 10th International Conference on Advances in Computer

Entertainment Technology (ACE 2013), LNCS 8253, pp. 568–571, 2013.

[4] Jamieson, P. Arduino for teaching embedded systems. Are computer scientists and engineering educators missing the boat?. Available: http://www.users.miamioh.edu/jamiespa/html_papers/fecs_11.pdf.

[5] Banzi, M. Getting Started with Arduino. O'Reilly Media, Inc., 2011, ISBN: 978-1-449-309879.

[6] OLIMEXINO-328 development board. Users Manual. OLIMEX Ltd, 2011

[7] Yakimov, P. Teaching basic skills in embedded systems using open source platforms. Proceedings of the L International Scientific Conference on Information, Communication and Energy Systems and Technologies, June 24-26, 2015, Sofia, Bulgaria, pp. 317-320.

Assoc. Prof. Dr. Peter I. Yakimov received his M.S. and Ph. D. degrees in electronic engineering from the Technical university of Sofia, Bulgaria, in 1984 and in 2003 respectively. He is currently an Associate professor and a Head of the Department of Electronics at the Technical university of Sofia. His research interests are in the area of analog and digital circuits design, embedded systems.
tel.: +359 2 965 3265. e-mail: pj@tu-sofia.bg

Received on: 31.08.2015

ФЕДЕРАЦИЯ НА НАУЧНО-ТЕХНИЧЕСКИТЕ СЪЮЗИ

ИСКАТЕ ЛИ ДА ОТГОВОРИТЕ НА ПРЕДИЗВИКАТЕЛСТВОТА НА ВРЕМЕТО?

ПОТЪРСЕТЕ ФНТС ЗА:

- *Научно-технически конференции, симпозиуми, изложби, панаири и други изяви у нас и в чужбина;*
- *Семинари, курсове и школи за професионална квалификация и преквалификация*
- *Специалисти и консултанти за разработване на проекти, свързани с технологични иновации, приватизацията, финансовата политика и др.;*
- *Информационна и издателска дейност на високо професионално равнище;*
- *Ползване на конферентни и изложбени зали, симултантна и офис техника, научно-технически видеофилми и др.*

ДОВЕРЕТЕ СЕ НА ПРОФЕСИОНАЛИЗМА И КОМПЕТЕНТНОСТТА НИ!

За контакти с ФНТС:

София 1000, ул. Г. С. Раковски № 108, Тел. 987-72-30; факс 986-16-19 и 987-93-60