Microcontrollers and optical sensors for education in optics and photonics

Dressler, Paul, Wielage, Heinz, Haiss, Ulrich,
Vauderwange, Oliver, Wozniak, P., et al.

Paul Dressler, Heinz Wielage, Ulrich Haiss, Oliver Vauderwange, P. Wozniak, Dan Curticapean, "Microcontrollers and optical sensors for education in optics and photonics," Proc. SPIE 9188, Optics Education and Outreach III, 91880F (15 September 2014); doi: 10.1117/12.2061836

Event: SPIE Optical Engineering + Applications, 2014, San Diego, California, United States

Microcontrollers and Optical Sensors for Education in Optics and Photonics

Paul Dreßler[1*], Heinz Wielage[1], Ulrich Haiss[1], Oliver Vauderwange[1], P. Wozniak[1] Dan Curticapean[1]
[1] Offenburg University, Badstr. 24, 77652 Offenburg/Germany

ABSTRACT

The digital revolution is going full steam ahead, with a constantly growing number of new devices providing a steady increase in complexity and power. Most of the success is based on one important invention: the microprocessor/microcontroller.

In this paper the authors present how to integrate microcontrollers and optical sensors in the curricula of media engineering by combining subjects of media technology, optics, information technology and media design. Hereby the aim is not to teach these topics separate from each other, but to bring them together in interdisciplinary lectures, projects and applications.

Microcontrollers can be applied in various ways to teach content from the fields of optics and photonics. They can be used to control LEDs, displays, light detectors and infrared sensors, which makes it possible to build measuring instruments like e.g. a lux meter, a light barrier or an optical distance meter.

The learning goals are to stimulate the student's interest in the multiplicity of subjects related to this course and to support a deeper understanding of the close connections between them. The teaching method that the authors describe in their paper turned out to be very successful, as the participants are motivated to bring in their own ideas for projects, they spend more time than requested and as many students return to the courses as tutors. It is an example for effectual knowledge transfer and exchange of ideas among students.

1. INTRODUCTION

About three years ago at the University of Applied Sciences Offenburg a new project was launched: Programming microcontrollers in a hands-on seminar, which contains theoretical contents alternating with practical tasks, whereat focus is on the practical issues ("learning by doing"). Although at this time programmable microcontrollers weren't as widespread as now, a growing number of students have been taking this course since then. Today, with Arduino, Raspberry Pi and others, programmable microcontrollers are booming and a multiplicity of new boards keeps coming to market.

This boom can be compared with the rise of the wide range of personal computer in the 1980s – each with a different CPU, a different look and different possibilities. We are very glad to having anticipated this trend, when we started our project in 2011.

As our university offers several courses of studies with a wide range of subjects – like mathematics, information technology, natural science, electrical engineering and media design – it is always a big topic how to combine these different faculties. The aim is not to teach the subjects separate from each other, but to bring them together in interdisciplinary lectures, projects and applications. The micro-controller project is a successful combination of several of these subjects.

Our microcontrollers are programmed in the programming language *C*, which is a good choice, for all students learn the basics of this language in their first semester. Equipped with this previous knowledge the students are able to start their own programming projects after only a short time. The student's certificate of performance at the end of the semester is

[*] Corresponding author: paul.dressler@hs-offenburg.de

Optics Education and Outreach III, edited by G. Groot Gregory, Proc. of SPIE Vol. 9188,
91880F © 2014 SPIE · CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2061836

to gather in small groups and to develop a microcontroller program, which meets certain demands. Thereby they are encouraged to contribute their own ideas and interests.

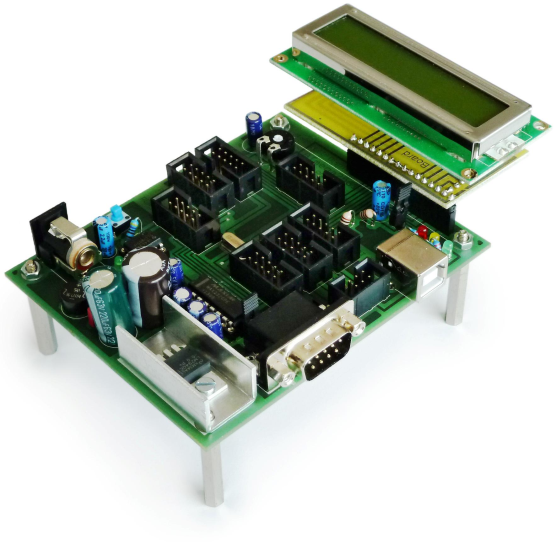

Figure 1: Microcontroller board with display

The learning goals are to stimulate the student's interest in the multiplicity of subjects related to this course and to support a deeper understanding of the close connections between them. By now we are focusing particularly on issues of optics and photonics. Therefore microcontrollers can be applied in various ways: They can be used to control LEDs, displays, light detectors and infrared sensors, which makes it possible to build measuring instruments like e.g. a lux meter, a light barrier or an optical distance meter.

2. THEORETICAL BACKGROUND

2.1 Structure and function of a microcontroller

A microcontroller is a "computer on a chip" – a single chip that contains an entire simple, small computer. Usually it is constructed for a particular purpose, e.g. for controlling a traffic light system, a digital clock etc.

The microcontroller's centerpiece is the central processing unit (CPU), which is also called microprocessor and which primarily consists of the arithmetic and logic unit (ALU) and the control unit (CU). Its task is to perform all operations on data by means of a program:

The *ALU* loads its data from *memory* and also stores it there afterwards. The *control unit* decodes the given instructions of the program, e.g. jump instructions or bitwise operations. Subsequently the *ALU* executes these instructions and manages the data in- and output. All these peripheral units are built in a microcontroller, for the aim is to integrate as much (programmable) functionality as possible into one single chassis. The microcontrollers we use are reprogrammable, which is a precondition for our educational purpose: Our students can freely test their written programs or program fragments by trial and error. At our laboratory we use the AVR Atmega 2561 by Atmel. Reasons for this decision will be discussed in the following chapter.

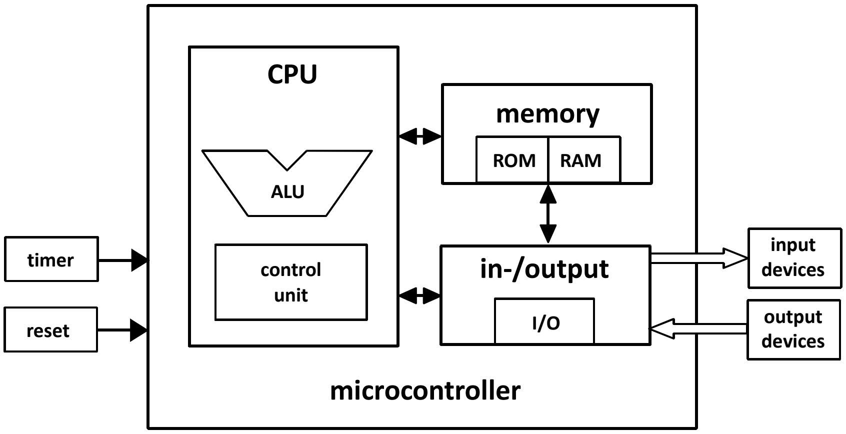

Figure 2: Microcontroller system, drawn by author

2.2 Input/Output

Our microcontroller provides 64 pins which can be independently declared either as input or as output registers. This setting is controlled by software. Eight pins are grouped into one port. As the pins work digitally, they can differentiate between two levels:

Being used as an output register, a pin can be pulled to "low", which means that is connected with ground and set to 0V. The second level, identified with "high", connects the pin with the supply voltage of the controller. Typically this is 5V.

Working as an input register, a pin can distinguish between the two voltage levels "ground" (0V) and "Vcc" (5V), whereas the threshold lies around 2,5V.

For interpreting analog signals an analog-digital converter (ADC) is required. The AVR Atmega 2561 contains a 10-bit ADC, which determines the discrete values by using successive approximation in a range between "ground" and "Vcc". In our seminar both digital and analog sensors are used. [2]

3. DECISION FOR AN AVR ATMEGA MICROPROCESSOR

As the microcontroller scene has been growing constantly over the last years, there exists a big variety of programmable microcontrollers with a wide range of possibilities. Even complete boards, which contain everything you need for a quick start into the field of microcontrollers, are commercially available for only a few dollars. So it would stand to reason to choose one of these boards for the microcontroller project at our university.

3.1 Decision for a self-made microcontroller

Despite of this progress, we decided not to use an "out of the box" solution, but to develop and build a microcontroller board on our own. The main reason for this decision was our request to use a board which fits perfectly to the teaching methods of our seminar: Our learning goal for the students is not to get quick programming results, but to stimulate a general comprehension of the topic and to create knowledge, that is not bound to a special type of microcontroller. So afterwards the students are able to work with microcontrollers, no matter of which brand.

The leading thought when developing our board was to give it a clear structure, so it would be easy to understand its functions and signal paths. One example is the visible 8-bit architecture, which can be found all over the board: The LED panel consists of 8 LEDs, the keyboard has 8 buttons and the LED matrix is built up of 8 columns and 8 rows. This logical structure is not only helpful for understanding the hardware, but especially for programming the microcontroller, because many calculations and addressing have to be done in binary arithmetic.

3.2 Why not Arduino?

Of course our board is not the first one to be designed especially for beginners. Boards like “Arduino” or the “Wiring board” have managed to bundle the complex hard- and software of a microcontroller in a user-friendly package. But their target audience are mainly artists in the broadest sense, who don’t want to familiarize themselves with electronics, bus protocols and hardware-near programming. Their aim is to get quick and simple results for their ideas. On Arduino instead of writing plain code you work with descriptive commands, which are easier to use, but on the other hand have limited possibilities and are much more power consuming. For example, the C code for switching one pin of a microcontroller port, which looks like this

```
PORTB |= ( 1 << PB0 );
```

is not intuitive at all, but works on every controller by using only one or two clock cycles. To do the same on an Arduino board you can write

```
digitalWrite( 9, HIGH );
```

which takes about 50 clock cycles and does not explain what is going on inside the machine. But it’s easy to understand.

At this time there is one other popular microcontroller board beside Arduino: Raspberry Pi. That board is also adequate for beginners and allows a wide range of application possibilities. To work with it, it has to be connected to some hardware like a monitor and keyboard and then can be programmed in the programming language *Python*. As our students are already familiar with the language *C* and as our aim is to keep it simple, the Raspberry Pi doesn’t fit our “beginners” seminar.

Apart from this seminar we could experience both Arduino and Raspberry Pi to be great platforms for students, who want to develop further projects.

4. SETUP

4.1 The development board (hardware)

The development board consists of seven modules: Most important is the microcontroller board at the bottom. It is equipped with seven plugs for in- and output devices and an integrated LCD panel (figure 3, item 1). The microprocessor AVR Atmega 2561 from Atmel Company is situated on the back side of the board.

4.2 On-board in- and output devices

The development board has been designed to provide an easy entry into the field of microcontroller technology and programming. The devices are structured in a way that makes their functions easy to understand and allows a wide range of applications.

Our board is provided with three on-board input devices:

- 8 push buttons (figure 3, item 2)
- 8 switches (item 3)
- a 3x4 numeric keypad (item 4)

and four output devices:

- 8 LEDs (item 5)
- four 7-segment displays (item 6)
- an 8x8 LED matrix (item 7)
- an LCD module with two rows each with
  16 characters (on item 1)

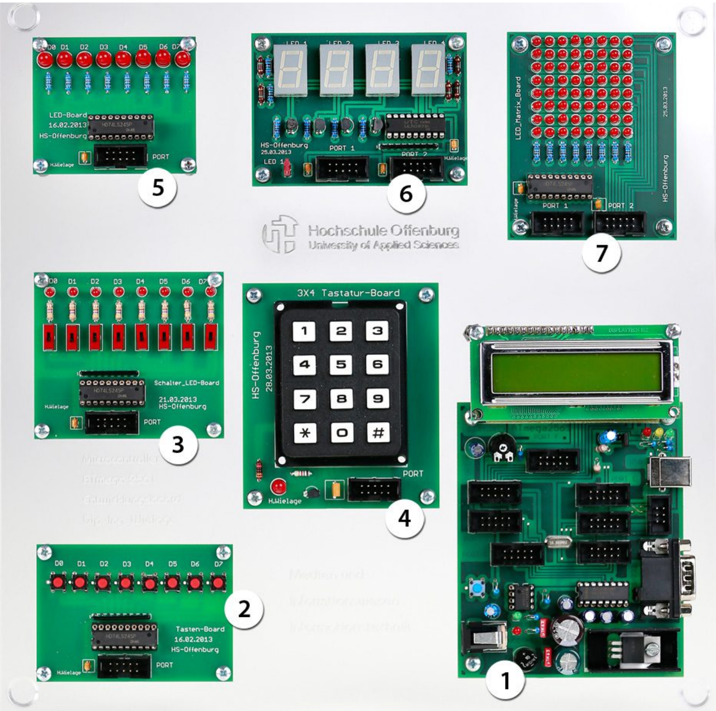

Figure 3: Realized application setup with Atmega2561 system architecture. (picture taken by author)

As an example the authors would like to take a closer look at the 7-segment display board (6):

This output device can display numbers and some characters with four 7-segment displays, which are controlled via two ports (16 pins). The students can use this module in two ways with different levels of difficulty.

A simple possibility is to use just one of the four displays. In this case the display can be triggered directly analog: Each pin switches on and off one segment of the number in exactly the same way as one pin would switch on and off one LED of the LED panel. Seven segments and one decimal point add up to 8 pins, so exactly one port is required.

To tap the full potential of the 7-segment display board it can be used as a four-digit display. As two ports do not suffice for an analog control, the signal has to be multiplexed. The student's first steps into this advanced method may be accompanied by several errors, but to gain insight into time-division multiplexing (TDM) delivers important knowledge about communication and media technology.

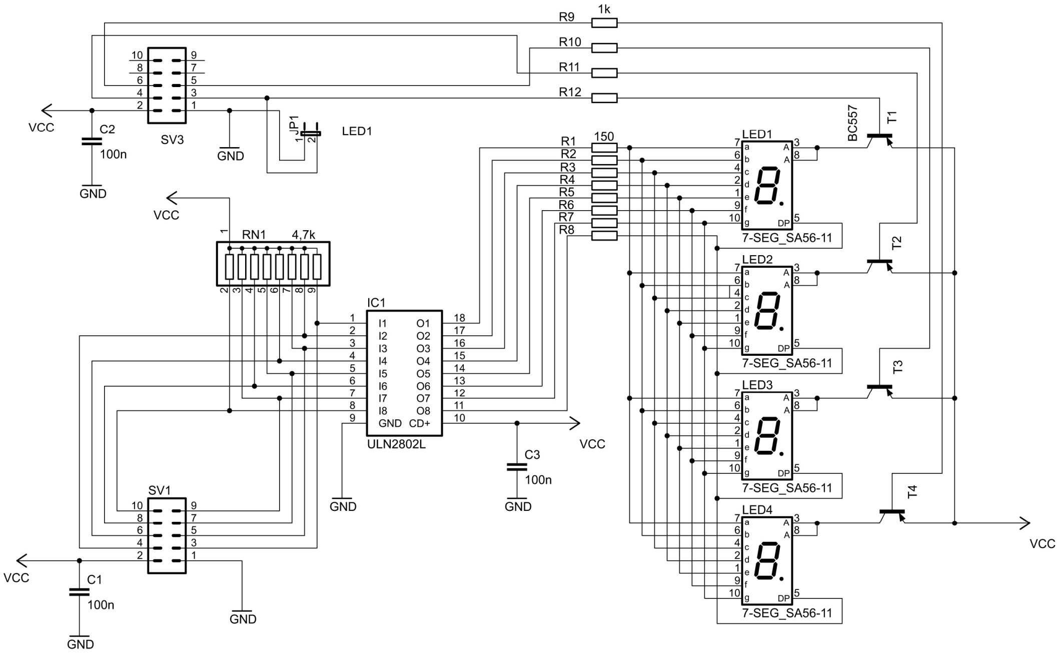

Figure 4: Circuit diagram of the 7-segment display board, drawn by author

4.3 External devices: sensors

To support our students in their understanding of optics we are additionally using several sensors. For technical measurement reasons they are not part of the development board, but are connected as external input devices to a port of our microcontroller.

In the matter of controlling these devices there are three types of sensors:

- Sensors, which deliver an analog signal,
- sensors, which deliver a digital signal,
- ready-to-use modules with a built-in circuit board.

The hardest way: Analog sensors (light sensor BPW42)

Signals from analog light sensors like the BPW42 are difficult to handle when quick results are in the focus. They require additional components, have to be digitalized via an ADC and need to be converted to get the correct units. But from an educational point of view, they are great to demonstrate how integration and conversions can be done.

The analog light sensor shows a linear relationship between the irradiance and the phototransistor current. When using the displayed circuit (Figure 5) the sensor delivers an output voltage of $U_L = I_C\ (R_V+R_P)$ which is proportional to the irradiance E_e (in *W/m*^{2}). If required, the irradiance has to be converted into the illuminance E_v (in *lux*). Afterwards the calculated voltage has to be adapted to the 10-bit A/D converter. [6]

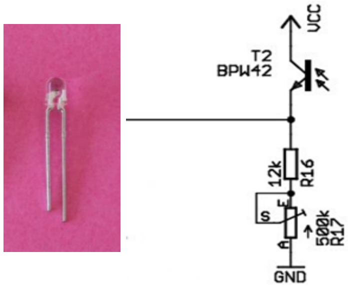

Figure 5: left: light sensor BPW42 (picture taken by author), right: required circuit (drawn by author)

A simpler way: Digital Sensors (infrared light sensor TSOP 1730)

The TSOP 1730 is an infrared sensor with a built-in analog-digital converter, which either sends a logical 1 or a logical 0. So the receiver's output signal S_{OUT} can be passed directly to the microcontroller. To connect it with the microcontroller an additional capacitor has to be put into the circuit. The sensor works with a carrier frequency of 30 kHz and a maximum data rate of 2400 bps. Besides measuring irradiance another possibility for an application could be to decode signals of a remote control. [7]

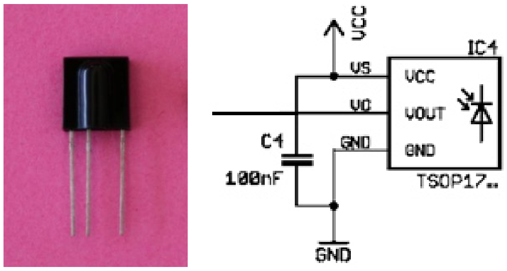

Figure 6: left: light sensor TSOP 1730 (picture taken by author), right: required circuit (drawn by author)

For the development of appropriate algorithms for measuring illuminance or temperature the students have to possess a basic knowledge of optics and electrical engineering.

The easiest way: Ready-to-use modules (PIR module HC-SR501)

The third type of sensor can be used for a very quick start into working with optical sensors: The HC-SR501 (or DYP-ME003), a passive infrared movement detector, integrates a sensor module on a circuit board, which just has to be connected to the microcontroller via three plugs (ground, +5V, output signal). When detecting a movement the module sends a voltage of 3,3V, when the movement stops it returns to 0V ("low"). [4]

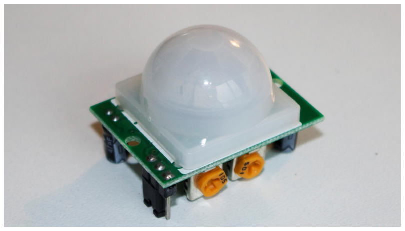

Figure 7: PIR module HC-SR501 (picture taken by author)

4.4 Software

Along with electrical engineering, optics and mathematics, the subject of information technology is an important part of our microcontroller project. This includes knowledge about the syntax of the programming language *C*, the ability to structure the own source code and the skill of operating the IDE: The integrated development environment is an application package for the PC combining editor, assembler, compiler, linker and simulator applications under one uniform surface, which can be operated more easily. For programming microcontrollers there is a variety of development environments for all different types of microcontroller hardware. All of them have in common that they are useful tools, which offer fast access to important functionalities, liberating the developer from recurring and formal tasks and helping him to manage his work results, so he can focus completely on his real task: the development of microcontroller applications. [8]

To transfer the written source code to the microcontroller, the code first is being compiled into machine code in *hex* format and then loaded to the controller's memory via a programmer. At this project we are using the development platform *AVR Studio 4* by *Atmel* in combination with the compiler *WinAVR*.

5. APPLICATIONS

Unlike other microcontrollers the Atmega 2561 can be programmed without restrictions. Together with the modular structure of our development board it offers a big variety of thinkable applications. In our courses after a short theoretical instruction the students are confronted with several small tasks, which are reasonably simple to solve, e.g.:

Controlling an LED

Task: The LED connected to Pin 0 of Port C shall be switched on by pushing a button, which is connected to Pin 0 of Port D, and off by releasing it.

Example source code for a solution:

```
//Button activates LED

#include <avr/io.h>           // adding library (input/output etc.)
#include <util/delay.h>       // adding the "delay" functionality

int main(void){

  DDRC |= (1<<PA0);           // declare Pin 0 of Port C as output
  PORTC |= (0<<PA0);          // switch off Pin 0
  DDRD |= (0<<PA0);           // declare Pin 0 of Port D as input

  while(1)                    // repeat forever
  {
    while (bit_is_clear(PIND, PD0)) //while button on Pin 0 of Port D is pressed
    {

      PORTC |= (1<<PA0);      // switch on the LED
      _delay_ms(50);          // wait 50ms
      PORTC &= ~(1<<PA0);     // switch it off
    }
  }
  return 0;
}
```

A short and clear code snippet like this has proven to be most suitable for getting used to the syntax of the programming language *C* and of the control commands for our microcontroller. After a couple of similar finger exercises with a constantly increasing level of difficulty the students are finally able to manage programs with more complex structures and functionalities.

At the end of the semester the student get together in groups to work on their final programming projects – their certificates of performance. Therefor the students can rent developing boards to take them home and to delve into programming. A majority of the students takes advantage of this offer. So until now they did submit several nice and useful projects. The students are free to include using sensors into their projects (for projects like e.g. a light barrier, a lux meter, a digital watch or math games) or to get by with the on-board modules of the developing board (for projects like simple games, a stop watch, a clock and date display including leap years etc.)

6. RESULTS

After three years of microcontroller programming, our course has increasingly established itself. The good results of the student’s programming tasks confirm this: Not only that the students are able to get along with their tasks very well – in addition many of them are highly motivated to go further into the subject by coming up with their own ideas how to enhance their projects. Notably among the participants of our seminar there are also many female students. The feedback sheets our students returned were very positive and had one common tenor: “Finally a really ‘hands-on’ course”.

Considering the taxonomy of learning goals proposed by Bloom [3] this achievement of supporting the students in realizing their own ideas reaches level K5 (Synthesis: Composition of individual elements to a whole) to K6 (evaluation) on his 6-level-scala.

REFERENCES

[1] Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K.A., Mayer, R. E., Pintrich, P. R., Raths, J. R., Wittrock, M . C, [A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom’s Taxonomy of Educational Objectives], Addison-Wesley, New York (2001).

[2] Corporation, Amtel, "Datasheet of Atmega640/1280/1281/2560/2561 Complete", 18 February 2014, http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf (21 July 2014)

[3] Bloom, B.S., [Taxonomie von Lernzielen im kognitiven Bereich], Beltz Verlag, Weinheim und Basel (1972).

[4] ***, "Specification of DYP-ME003", 3 June 2013, http://elecfreaks.com/store/download/datasheet/sensor/DYP-ME003/Specification.pdf (22 July 2014)

[5] ***, "Hands-On Optics", 10 June 2013, http://www.osa.org/en-us/membership_education/youth_education/hands-on_optics (05 July 2014)

[6] Telefunken, Inc., "Datasheet BPW 42", 5 Dezember 2000, http://www.datasheet4u.net/download.php?id=555151 (21 July 2014)

[7] Semiconductors, Vishay, "TSOP17 Photo Modules for PCM Remote Control Systems", 23 December 2004, http://pdf.datasheetcatalog.com/datasheet/vishay/82030.pdf (21 July 2014)

[8] Wikipedia, Die freie Enzyklopädie, "Integrierte Entwicklungsumgebung", 13 July 2014, https://de.wikipedia.org/wiki/Integrierte_Entwicklungsumgebung (21 July 2014)