Introducing 32-bit microcontroller technologies to a technology teacher training programme

D. Rihtaršič & S. Kocijancic

University of Ljubljana
Ljubljana, Slovenia

ABSTRACT: In our everyday life, we are accustomed to the use of contemporary electronic devices with continuously growing potential. However, such technologies are rarely introduced to technology education even in higher education. Since this technology is developing rapidly, the gap between the everyday implementation and the technology education programmes is also growing rapidly. In practical laboratory exercises, 8-bit microcontroller evaluation kits still dominate. On the other hand, one can find much more powerful 32-bit microcontrollers, which are capable of controlling contemporary devices such as SD memory cards, graphical LCDs, touch pads, intelligent sensors, etc. The reason for this is the more complex architecture of those devices, which led to rather complicated programming. This paper outlines some interesting examples of 32-bit microcontroller implementation, and how they can be introduced into a technology teacher training programme with a new programming environment that introduces the use of drivers. The pros and cons of 32-bit microcontrollers are considered in the conclusions of the paper.

INTRODUCTION

Technology has been developing rapidly, particularly in recent decades. What is great for everyday life usage is that such technologies allow us to complete various tasks easier, simpler and faster. Everyone uses devices that execute orders by sensing our touch, movements or even gestures and such devices are ubiquitous (mp3s, mobile phones, automotive applications, etc). On the other hand, those devices are rarely introduced into engineering education programmes.

Eight-bit microcontrollers still dominate in educational practice in contrast with 32-bit microcontrollers or even ARMs, which are mostly implemented in contemporary electronics devices. It is very hard to follow the development of technology within the educational context. However, experts in the technology educational field indicate that it is very important to expose interesting and relevant problems for the effectiveness of teaching [1-3].

The use of 8-bit microcontrollers is well settled in the education field all over the world, as well as in hobby workshops [4-7]. They have a simple architecture and still manage to be part of amazing projects. That is why it is much easier to explain their working processes and easier to program them. In the very first stage of introducing microcontrollers to students, 8-bit microcontrollers are more than welcome. They are appropriate for beginners to get some basic ideas about how microcontrollers works and, as is well known, the greater the knowledge base they have, the easier it will be for the microcontrollers to be applied in a new problem situation.

However, not everything can be explained in detail because it would require too much time in the educational process and would consequently be inefficient. So, the teacher is not able to introduce implementation of contemporary devices such as SD memory cards, touch screens and other sophisticated sensors and actuators. However, things are going to change; big steps in this direction are being undertaken. The ATMEL Corporation is one of the worldwide leaders in the design and manufacture of microcontrollers and, in 2011, they launched a programming environment called AVR Studio 5 [8]. This integrated development environment (IDE) combines all 8-bit, 16-bit and 32-bit microcontrollers with AVR architecture.

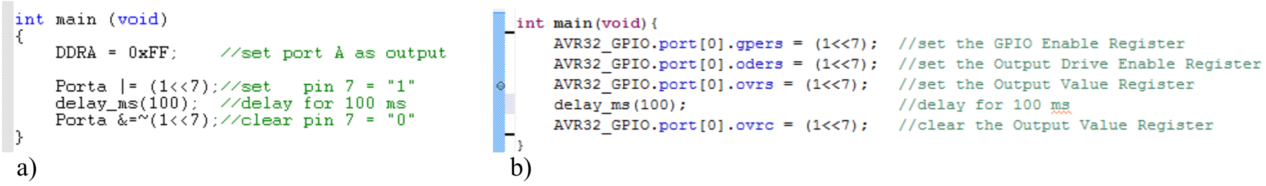
Following in the footsteps of others IDEs, they also included intuitive code writing, which is very helpful for beginners and saves a lot of programming time. Among the most powerful features, there are drivers that can be loaded into a microcontroller to support other components [9]. For that reason, the user does not need to know how to control the device exactly bit by bit, but can manipulate it with a few functions which provide perfect operation. Of course, not all devices are supported (and probably never will be), but it is a great start towards usage in the education field or even for hobbyists.

INTRODUCING 32-bit MICROCONTROLLERS

Introducing microcontrollers to trainee teachers of technology by way of 8-bit microcontrollers has some advantages. Those microcontrollers have a much simpler architecture and, it is, therefore, easier to understand their functioning. This contributes to the first steps of programming microcontrollers. Nevertheless, there are several 8-bit microcontrollers in DIP IC packages, which make implementation simpler and cheaper for students' projects.

However, it has turned out that these microcontrollers are becoming less associated with contemporary external devices, such as memory cards, touch pads and various sophisticated sensors. A crucial point is also the ability to connect the microcontroller to a PC as a USB device, which many 8-bit microcontrollers do not support.

Producers of a wide range of devices have started to use more powerful 32-bit microcontrollers such as ARMs, but these devices are more complex and harder to programme and are, therefore, used mostly in the professional domain [10]. The difference in the programming code of the simplest task to set output pin 7 of Port A and clear it after 100 ms delay is shown in Figure 1.



```
int main (void)
{
    DDRA = 0xFF;     //set port A as output

    Porta |= (1<<7);//set   pin 7 = "1"
    delay_ms(100);   //delay for 100 ms
    Porta &=~(1<<7);//clear pin 7 = "0"
}
```

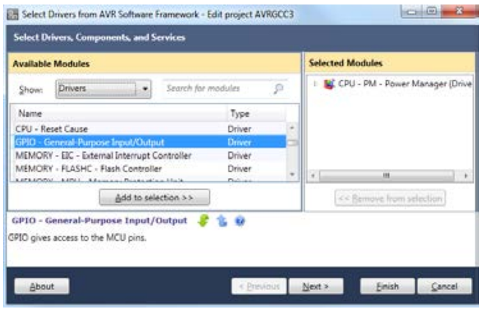
a)

```
int main(void){
    AVR32_GPIO.port[0].gpers = (1<<7);  //set the GPIO Enable Register
    AVR32_GPIO.port[0].oders = (1<<7);  //set the Output Drive Enable Register
    AVR32_GPIO.port[0].ovrs = (1<<7);   //set the Output Value Register
    delay_ms(100);                      //delay for 100 ms
    AVR32_GPIO.port[0].ovrc = (1<<7);   //clear the Output Value Register
}
```

b)

Figure 1: The difference in programming; a) 8-bit and b) 32-bit microcontroller.

In 2011, the ATMEL Corporation launched the most sophisticated programming environment in this field so far. It is called AVR Studio 5, integrated development environment (IDE) designed to develop and debug embedded Atmel AVR applications. The biggest step towards less skilled users is the integration of drivers. The driver is a set of many functions that serves to control some modules inside a microcontroller or even outside it. For example, the previous task looks much more easier in the AVR Studio 5 environment. Importing the GPIO driver in this project is shown in Figure 2, and the task to control the output pin 7 is represented with programming code in Figure 3.



Select Drivers from AVR Software Framework - Edit project AVRGCC3

Select Drivers, Components, and Services

Available Modules

Show: Drivers

Name	Type
CPU - Reset Cause	Driver
GPIO - General-Purpose Input/Output	Driver
MEMORY - EIC - External Interrupt Controller	Driver
MEMORY - FLASHC - Flash Controller	Driver

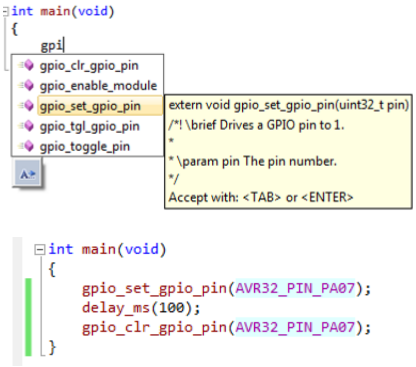
Selected Modules

CPU - PM - Power Manager (Drive

GPIO - General-Purpose Input/Output

GPIO gives access to the MCU pins.

Figure 2: Importing the drivers into the project.



```
int main(void)
{
    gpi
```

gpio_clr_gpio_pin
gpio_enable_module
gpio_set_gpio_pin
gpio_tgl_gpio_pin
gpio_toggle_pin

```
extern void gpio_set_gpio_pin(uint32_t pin)
/*! \brief Drives a GPIO pin to 1.
*
* \param pin The pin number.
*/
```

Accept with: <TAB> or <ENTER>

```
int main(void)
{
    gpio_set_gpio_pin(AVR32_PIN_PA07);
    delay_ms(100);
    gpio_clr_gpio_pin(AVR32_PIN_PA07);
}
```

Figure 3: Intuitive code writing (top) and final code with a driver's functions (bottom).

This example shows a big difference that came with the new IDE. In Figure 3, intuitive code writing, which helps to find functions easier and enables the coding to be done faster, is presented. The programme also provides a short description of the function, which is useful for beginners in this field. Figure 3 also shows the final code for the problem presented earlier. It can clearly be seen that this kind of programming is appropriate even for those who are dealing with 32-bit microcontrollers for the first time. Of course, in subsequent steps of programming things get complicated.

However, from the educational point of view the crucial steps are the very first ones. They must be as easy as possible and the results must be seen straight away.

INTRODUCING EVERYDAY TECHNOLOGY INTO EDUCATION

It is very important for students to be involved in problems, which are relational to our everyday life. One of the possible challenges is how to measure acceleration, while some other action is performing. Acceleration is also a theme in physics and this experiment could contribute to understanding the issue since students would be actively involved in the experiment. Furthermore, many contemporary cellular phones have such a s ensor. The acceleration sensor is embedded to provide the data, the SD card stores the data, and transfers it to the PC for further analysis. To introduce the problem, it was decided to implement an EVK-1101 evaluation board, which is equipped with several sensors including the accelerometer, and it has an SD memory card socket [11]. This board is powered by a 32-bit microcontroller 32UC3B0256 and can be programmed in AVR Studio 5. The board EVK-1101 is shown in Figure 4.

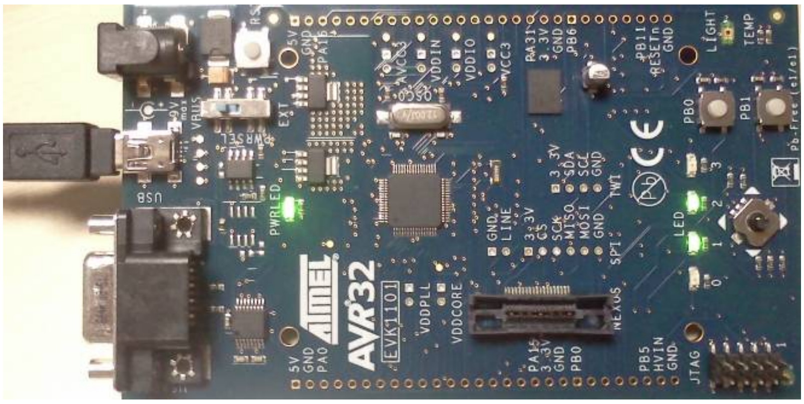


Figure 4: Evaluation board EVK-1101.

In this project, drivers such as ADC, TIMING, SD/MMC and others are uploaded for easier management of the hardware. After a little programming time, it was already possible to obtain the acceleration data, saved them to the SD card and sent them to the PC, where they were presented as is shown in Figure 5.

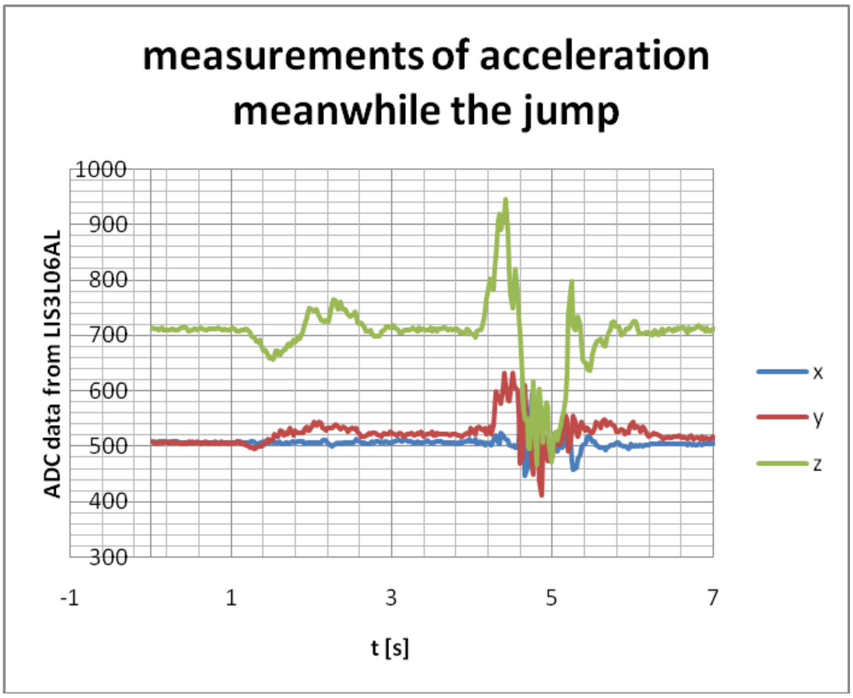


Figure 5: Representation of acceleration data during the jump.

Students can now recapture their actions and compare them with the sensor's response. They must also take into consideration the system force (gravity), which is not an easy task for them. It should be explained that the result to demonstrate the data could be hidden in the signals. Figure 5 shows three curves that represent acceleration in each direction (x - [forward/back], y - [left/right], z - [up/down]). The sensor's response at no acceleration presented is 512, which is right in the middle of its range and is clearly seen at the start of the x and y curves.

The green curve starts at 710, which represents gravity-related acceleration. The first wave on the green curve is from squatting to do a jump. After the jump, the steady signal at the same level is followed because there was no motion. The next action is jumping what can be seen with the tip around 4.4 s. Then, the fuzzy signal can be seen at around the same level as the red and green curves. This is not a coincidence because the student is now *falling* back to the floor with gravity acceleration. Looking from the student's point of view, the system force must be subtracted.

The result is zero gravity for that moment as he/she is off the ground. But that is not all. Looking at the red signal, it can be seen that during the student's action the red curve goes slightly above zero acceleration, which means that the student was angled slightly to the left. Why is that? It transpires that the student suffered an injury in the past on his/her right leg and until then, transferred the weight to the left leg by angling to the left. This fault is clearly seen right at the time of jump.

Only one opportunity to integrate modern technology into teacher training education has been presented here, but these devices offer much more. Other interesting devices can be used to maintain contact with fast developing technology. For example, one can use multi-touch pads without any buttons, which work as capacitor sensors or a graphical display as shown in Figure 6.

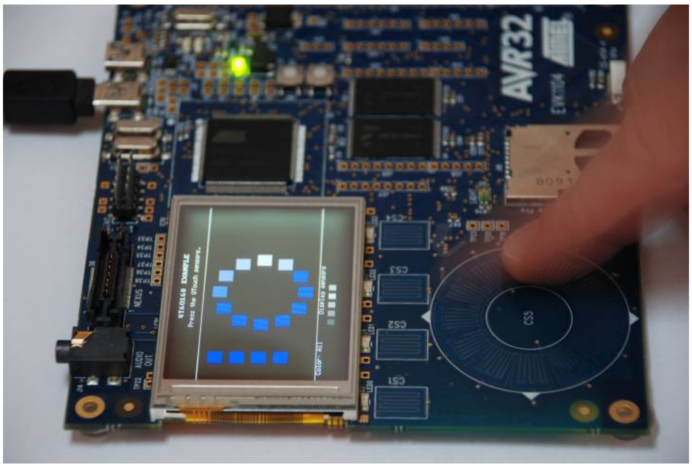


Figure 6: Use of graphical display and capacitor touch pad.

CONCLUSIONS

It cannot be argued that 8-bit microcontrollers still dominate in education field and enjoy major support around the world. However, if one wishes to educate students to become competitive teachers, they must be familiar with the technology that surrounds us all, and should follow the guidelines that are dictated by the development of technology. The authors believe that a new programming environment with the implementation of drivers, is a great step in introducing 32-bit microcontrollers for beginners, and it gives the teacher an excellent opportunity to implement these devices in an educational process.

Naturally, there were some obstacles, because not all devices are supported yet, the very popular alphanumeric display for example. Another deficiency of new IDE is that users cannot write their own driver and save it for another use in a project. However, it must be confirmed that such an approach will help to integrate modern technology into teacher education, and will help to reduce the gap between engineering and technology educational and contemporary technologies.

REFERENCES

1. Korbel, S. and Janes, V., Interesting applications of Atmel AVR Microcontrollers. *IEEE Euromicro Symposium on Digital System Design* (DSD'04), France (2004).
2. Hill. A.M., Problem solving in real-life contexts: An alternative for design in technology education. *Inter. J. of Technol. and Design Educ.*, 5, **3**, 1-18 (1998).
3. Savery, J. and Duffy, T.M., *Problem Based Learning: An Instructional Model and its Constructivist Framework*. In: Wilson, B.G. (Ed), Designing Constructivist Learning Environments. Englewood Cliffs, NJ: Educational Technology Publications (1996).
4. Weiss, B., Gridling, G. and Proski, M., A case study in efficient microcontroller education. *Proc. 2005 Workshop on Embedded Systems Educ.*, 22 September, 36-43 (2005).
5. Giurgiutiu, V., Lyons, J., Rocheleau, D. and Liu, W., Mechatronics/microcontroller education for mechanical engineering students at the University of South Carolina. *Mechatronics 2005*, 15, 9, 1025–1036 (2005).
6. Kocijančič, S., Kušar, T. and Rihtaršič, D., Introducing programming languages through data acquisition examples. *Inter. J. of Emerging Technol. Learning*, 3, **2**, 28-33 (2008).

7. Murovec, B. and Kocijančič, S., A USB-based data acquisition system designed for educational purposes. *Inter. J. of Engng. Educ*., 20, **1**, 24-30 (2004).
8. Atmel Corporation, Tools and Software: AVR Studio 5, 1 June 2011, xhttp://www.atmel.com/dyn/products/tools_ca rd.asp?tool_id=17212
9. Atmel Corporation, The new Atmel AVR Studio 5, 1 June 2011, http://www.atmel.com/microsite/avr_studio_5/default.a sp?source=redirect
10. Sharad S., Introducing embedded design concepts to freshmen and sophomore engineering students with LEGO MINDSTORMS NXT. *Proc. 2007 IEEE Inter. Conf. on Microelectronic Systems Educ*., San Diego, 119-120 (2007).
11. Atmel Corporation, EVK1101 Schematics, 1 June 2011, http://www.atmel.com/dyn/resources/prod_documents/EVK11 01_Schematics_RevB.pdf