



# Crystal Clear Electronics

The development of the Crystal Clear Electronics curriculum was supported by the European Commission in the framework of the Erasmus + programme in connection with the “Developing an innovative electronics curriculum for school education” project under “2018-1-HU01-KA201-047718” project number.



## Erasmus+

The project was implemented by an international partnership of the following 5 institutions:

- Xtalin Engineering Ltd. – Budapest
- ELTE Bolyai János Practice Primary and Secondary Grammar School – Szombathely
- Bolyai Farkas High School – Târgu Mureş
- Selye János High School – Komárno
- Pro Ratio Foundation working in cooperation with Madách Imre High School – Šamorín



**XTALIN**



### Copyrights

This curriculum is the intellectual property of the partnership led by Xtalin Engineering Ltd., as the coordinator. The materials are designed for educational use and are therefore free to use for this purpose; however, their content cannot be modified or further developed without the written permission of Xtalin Engineering Ltd. Re-publication of the materials in an unchanged content is possible only with a clear indication of the authors of the curriculum and the source of the original curriculum, only with the written permission of Xtalin Engineering Ltd.

**Contact** <http://crystalclearelectronics.eu/en/>  
[info@kristalytisztaelektronika.hu](mailto:info@kristalytisztaelektronika.hu)

# 09 - Microcontrollers I.

Written by Gergely Lágler, Viktor Vincze

English translation by Xtalín Engineering Ltd.

Revised by Szabolcs Veréb

By the end of this chapter, our microcontroller will be able to blink an LED, based on a pre-written software.

Let's get right to the point! We're going to turn the normal learning system upside down here, and try the device at first, so later, when you try to understand its internal structure and operation, you will already have some tangible experience. We will use the microcontroller Atmega16A, and we will implement all the examples of the curriculum with the help of this device.

So, we will learn about how the microcontroller works step-by-step during practice. This knowledge will be enough to implement simple devices on your own, however we have to note that, the curriculum doesn't substitute any kind of vocational training.

What is a microcontroller? What do you use it for? Where do they use them in the industry? How many of them are there? What's the difference between models? Why are we using this?



Figure 1 - ATmega16A microcontroller

Author: Pengo [CC BY-SA 3.0 (<https://creativecommons.org/licenses/by-sa/3.0/>)]  
[https://commons.wikimedia.org/wiki/File:ATmega8\\_01\\_Pengo.jpg](https://commons.wikimedia.org/wiki/File:ATmega8_01_Pengo.jpg)

The microcontroller is also an IC, in other words an integrated circuit. This means that inside there are several super small components on a minuscule semiconductor plate (chip), which together are capable of complicated and complex operations.

## PC ANALOGY

The easiest way to imagine microcontrollers is if we compare them to a computer (PC) or a laptop, because they have lots of things in common, even though computers are not micro-programmed devices. The small parts of the microcontroller have the same tasks as the bigger devices inside a PC, however they are



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

*This is xxx personal copy - distribution is strictly prohibited.*

<http://crystalcleelectronics.eu> | All rights reserved Xtalín Engineering Ltd.

doing this while they are integrated into very tiny case (this is where their name comes from). A program (created by us) runs on microcontrollers and PC as well, and this software determines the operation of the device.

## PROCESSING UNIT

In a PC, the processor completes the basic operations, for example the addition, subtraction, multiplication, division. Inside the microcontroller, these tasks are also completed by the central processing unit (CPU).

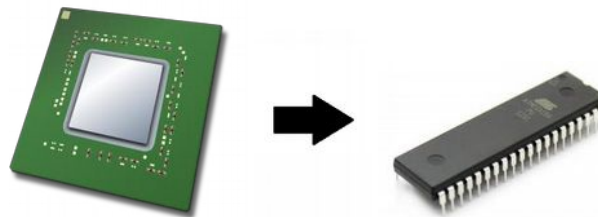


Figure 2 – Processor from a PC, and a microcontroller

## NON-VOLATILE MEMORY

Hard disk drives (HDD) and SSDs are non-volatile memories, which are used in PCs. Huge amount of data can be stored on them, for example, our programs and our files. The data is retained, even after turning off the computer, that's why we call them non-volatile memories.

Similar units can be found inside the microcontroller. The part, which contains the program, is called FLASH memory or program memory. The part, which can be used to store other data, is called EEPROM memory.

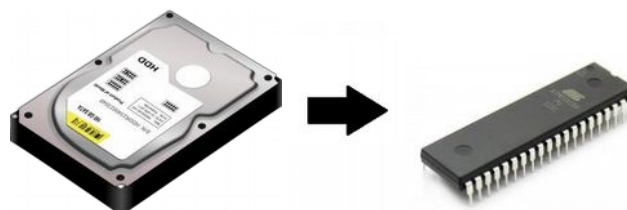


Figure 3: HDD from a PC, and microcontroller with integrated FLASH

## ADDITIONAL NON-VOLATILE MEMORIES

External storage options can be connected to the PC (if we need more memory), for example, external HDD, USB drive, SD card.

Similar options for microcontrollers also exist, as an example, an external „EEPROM”, or the SD card.

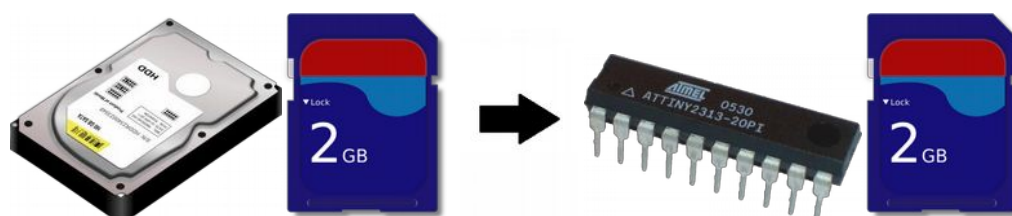


Figure 4: Typical external memories connected to a PC and to a microcontroller



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

This is xxx personal copy - distribution is strictly prohibited.

<http://crystalcleelectronics.eu> | All rights reserved Xtaln Engineering Ltd.

## VOLATILE MEMORY

The data, stored in the PC's volatile memory, won't be retained after the PC is turned off, but this memory is capable for faster operation than the non-volatile, so it gives faster operation to the processor. This is the so-called RAM memory.

The same thing can be found in the microcontroller too.

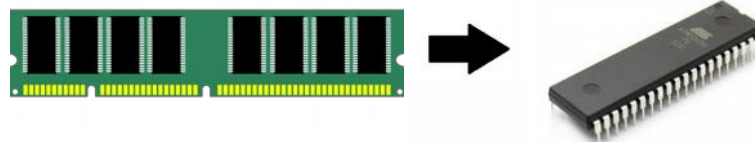


Figure 5: PC RAM memory, microcontroller with integrated RAM

## PERIPHERALS

We can connect different data input and output units, in other words, peripherals, to the PC, for example monitor, keyboard, mouse, printer, camera, scanner, etc., moreover these devices can be operated easily by connecting a plug, or installing a driver software.

Different peripherals can be connected to the microcontroller too, however, these are usually more primitive ones. Peripherals can be sorted into two groups in terms of their direction, they can be outputs, or inputs.

Example for simple outputs: small display, buzzer, LED strip. Typical inputs: push buttons, different sensors (temperature sensor, pressure sensor, current sensor, etc.).

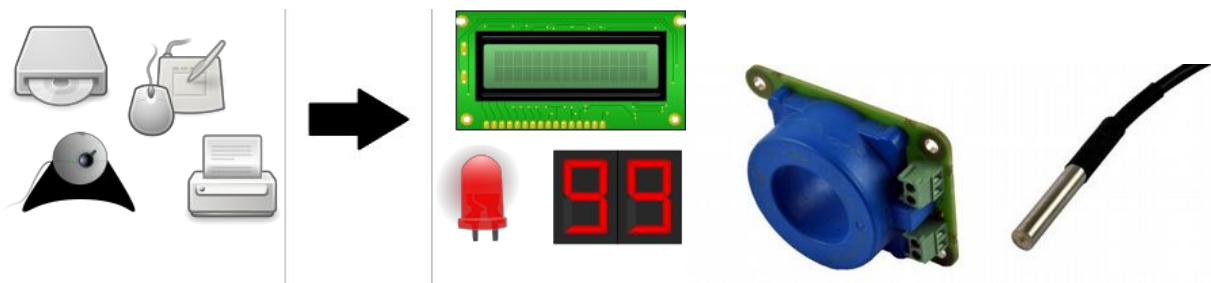


Figure 6: Comparison of peripherals

These devices will complete our desired task only, for example indicate, measure or change the given physical information, if they are connected properly to our microcontroller and we have to write a software for the microcontroller, which can handle the given periphery.

## SUMMARY

There are several reasons for the existence of microcontrollers. These are small components, though they have much smaller computing capacity, their energy consumption is also a lot less, and they don't contain moving components that wear out, so their lifetime is excessively high, and their operation is much more reliable. Of course, reliability requires reliable accessories and software as well.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

To be able to design a microcontroller circuit for serious industrial systems, years of learning and experience is required. Thus, we will only introduce a small segment of the basic building blocks to provide a practical background for learning even more complex theoretical basics.

Apart from the similarities of the PC and the microcontroller, their differences include size, price, reliability, handling, and complexity of the realizable task.

## AREAS OF APPLICATION

Microcontrollers, or very similar devices implement operation controlling and regulating tasks in most devices around us, that have electronic automation. A microcontroller can control a digital microwave oven, a washing machine, a dishwasher, a sensor or the control unit of an intelligent home, an elevators' interface, a card access control panel, a barcode scanner, a credit card reader, or most devices that can be found in hospitals or vehicles, for example, patient's guardian devices, blood pressure meters, defibrillators, vehicle controllers, electronics of windshield wipers, airbags, window lifters, electronically adjustable mirrors, ABS, AC, etc. A similar device can be found in mobile phones, modern radios, televisions too.

You can see that most electronic devices around us, and safety critical devices, contain a microcontroller. Learning about the Atmega16A microcontroller in this curriculum, will allow you to use a microcontroller for a wide variety of tasks, and based on that knowledge you will able to learn to use a more complicated microcontroller on your own.

### Types of micro-programmed devices

Micro-programmed device is the uniting name for those electronic components, for which the user or the designer has to write a program, in order to be operational. While the vast majority of micro-programmed devices are microcontrollers, there are several other types of devices as well, which differ in terms of structure or operation, and they can be more effective in a special area. For example, digital signal processors (DSP) have outstanding performance in the area of measurement technology. Similar devices are the CPLDs, FPGAs, Application processors, etc. For now, these are mentioned only for reference. The easiest way to learn about the internal structure, operation and programming of these, is through an electrical engineering course.

### Different architectures

The internal structure of the individual micro-programmed devices, i.e. the system architecture can be different depending on the main building blocks the device contains. Similarly, staying at the PC analogy, one of the most typical difference can be the data length of the processing unit. There are PCs, which can work with 32- or 64-bit long data in a single step. This feature defines the operating system, that can be installed on the PC, the installable software and even the efficiency of the PC. Similarly, we can find different CPUs in microcontrollers with different data length. 8-bit microcontrollers are very widespread, and probably it will remain like this, but we can meet 16-bit microcontrollers too, and longer data length can be found in more and more devices too. The ATmega16A is an 8-bit microcontroller.



*This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).*

## DIFFERENT MICROCONTROLLER MANUFACTURERS

Microcontrollers have been used since the 1970s, nowadays several billions of these devices are produced. The most common manufacturers are: Microchip, ST Microelectronics, NXP Semiconductors, Texas Instruments, Analog Devices. Previously Atmel and Freescale were prestigious names too, but in the recent years, they were bought by Microchip and NXP.

## THE ATMEGA16A MICROCONTROLLER

### PACKAGING OF THE ATMEGA16A

On the website of the curriculum under the extra materials section you can download the datasheet of the ATmega16A, that we will refer to. You will also find a reference to the manufacturer's website.

You can find more information about the package (Packaging information) on pages 325-327. Out of these, we have also chosen the through-hole one, to make it easy to use on a breadboard. Atmel marks this package with a code 40P6, which means that, the component has a total of 40 pins, and the distance between the two lines of pins is 0.6 inch (15.24 mm). This aligns perfectly with a breadboard, where the distance between rows is 0.1 inch (2.54 mm).

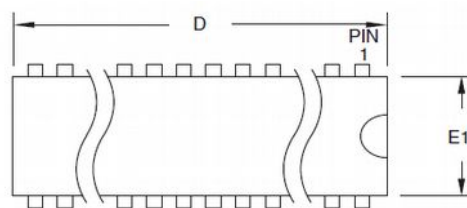


Figure 7: the DIP package (part of the datasheet)

Usually, this package is called DIP-40, which indicates a 40-pin component, with two lines of pins on the casing (Dual In-line Package).

### The complete marking of the manufacturer

We can look up the exact item number, on page 324 of the datasheet, under the ordering information. You can see that the ATmega16A-PU version has the 40P6 package.

### PIN ASSIGNMENT AND BASIC WIRING OF ATMEGA16A-PU

The pin assignment can be found on page 3 of the datasheet (Pin configuration, or Pinout):



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

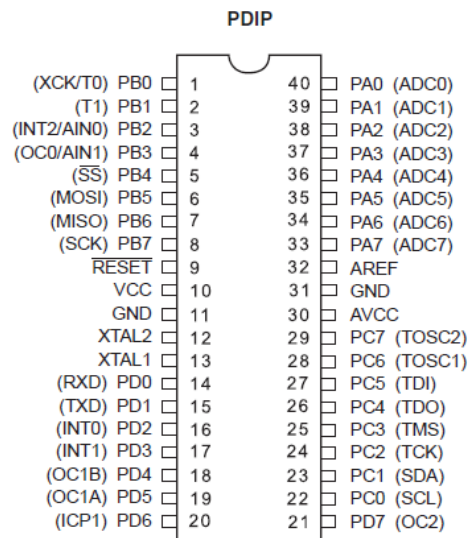


Figure 8: ATmega16A-PU pin assignment (part of the datasheet)

The datasheet can help you a lot to read the figure above, every pin's precise description can be found inside the overview section, under "pin description". Because, we learn about every single function, one-by-one, now it isn't necessary to understand the operation of every single pin and the circuit that is behind them (referred by an abbreviation in parentheses).

### Power Supply Pins

Since the microcontroller is an active component, it needs proper supply voltage for its operation, which we need to connect to its dedicated pins. Most microcontrollers, including the ATmega16A, have multiple power supply pins. For correct operation we need to connect the appropriate voltage to every single power supply pin.

The pins labelled GND are the „ground” pins, the VCC and the AVCC are the supply pins. In every microcontroller's datasheet, they indicate the operational voltage range, in our case, it can be found in the electrical characteristics section, on page 281.

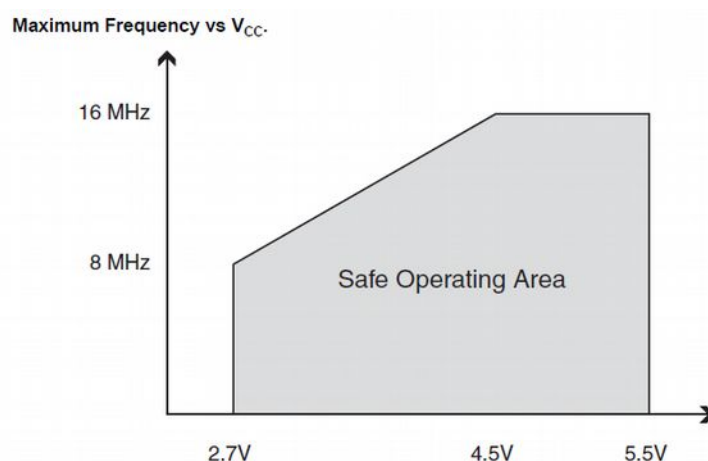


Figure 9: Safe operating area of the microcontroller (part of the datasheet)



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

The figure shows the safe operating area, which means our device will operate safely between 2,7 and 5,5 V of supply voltage. You can see in the figure as well, what clock frequencies can be used for different supply voltages.

## The clock signal

The clock signal consists of consecutive 0, 1, 0, 1 signal, a so-called square wave.

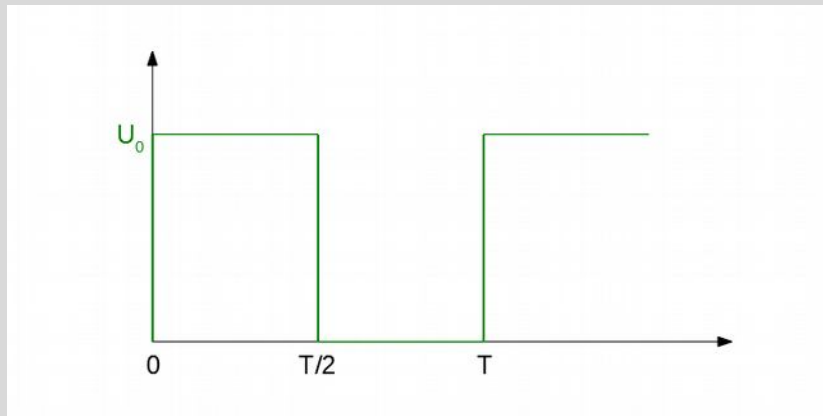


Figure 10: the clock signal

The change between the levels, is called the rising and falling edge of the square wave. Most of our devices are sensitive for rising edge, that's when it processes the next operation inside. Microcontrollers are no different. The clock signal is usually produced by a special circuit, which is used only for this purpose. In our case, we have several options to choose from, we will mention two of them.

At first, we will use the so-called internal RC oscillator which is built into the microcontroller. Its advantage is that no external circuit is needed to create a clock signal, but on the other hand the internal oscillator is not very accurate and cannot be used to create faster clocks. We can read about it in detail on page 28 of the datasheet, where you can see that the internal oscillator's highest frequency is 8 MHz, this is the maximum that the internal circuit is capable of.

We can also connect a proper quartz crystal to the microcontroller's XtalIn and XtalOut (the "xtal" expression refers to the crystal) pins, in this case, the microcontroller uses the crystal to produce the clock signal. For this purpose, several crystals are manufactured, one crystal can only produce a given frequency. Among others a 16 MHz crystal is available too.

## Operating speed of the microcontroller

We will set the clock signal's frequency at the time of programming, and this will give the operating speed of the microcontroller. Higher frequency means faster speed, which means faster operations, but the power consumption will increase by a little too. So, the 8 MHz frequency means that, in 1 second, this device could complete 8 million additions.

We can also see from the datasheet that the clock frequency can be set to 16 MHz too. This is twice as fast, but at least 4.5 V of power supply voltage is needed. For now, we are not aiming for low power



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).



consumption or very high speeds, so for simplicity's sake we will use the 8 MHz internal oscillator but give 5 V of power supply to the microcontroller.

### Absolute maximum ratings

In every case, it's worth to examine, what the maximum physical limits are (voltage, current, etc.) for our device. These are not for operation, but they are the values that the device's supply pins, other pins, or casing can tolerate without damage. This information can be found on page 279 of the microcontroller's datasheet.

#### 27.1 Absolute Maximum Ratings\*

Operating Temperature .....	-55°C to +125°C
Storage Temperature .....	-65°C to +150°C
Voltage on any Pin except $\overline{\text{RESET}}$ with respect to Ground .....	-0.5V to $V_{CC}+0.5V$
Voltage on $\overline{\text{RESET}}$ with respect to Ground .....	-0.5V to +13.0V
Maximum Operating Voltage .....	6.0V
DC Current per I/O Pin .....	40.0mA
DC Current $V_{CC}$ and GND Pins .....	200.0mA PDIP and 400.0mA TQFP/MLF

Figure 11: Absolute maximum ratings (part of the datasheet)

For us, every piece of data can be important later, at the time of installing a periphery, we will have to look back here and use them for the calculations:

- Maximum operating voltage: If we connect a voltage bigger than 6 V to the power supply, even for a short period of time, our microcontroller may be damaged permanently or become completely faulty.
- Voltage on any pin: it is prohibited to connect a voltage to a single pin, which is lower than -0.5 V (with respect to GND), or higher than the power supply (5V in our case) by 0.5 V.
- Maximum current on each pin (DC current per I/O pin): a current bigger than 40 mA is not allowed to flow in either direction.
- Maximum current on supply pins (DC current VCC and GND pins): a current, bigger than 200 mA, is not allowed to flow on either power supply pins.

### Digital Inputs and Outputs

All general-purpose pins (PA0...7, PB0...7, PC0...7, PD0...7) can be used as digital inputs or digital outputs.

A digital input can determine whether the connected voltage is a logic low or high value. We can get a logic low value if the connected voltage is „0 V” or a value near to „0 V”, we can get a logic high value, if the connected voltage is near to the power supply (VCC, for example 5 V).

A digital output connects logical low or logical high to the pin depending on what we previously set in the software, or how we change the value on the output during operation.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

## Exact definition of logic low and logic high

The question may well arise, for example, in case of 5 V power supply, what level will be surely detected as 0 and what level will be surely detected as 1, and in case of an output, what voltage levels are guaranteed. We can find the solution for this question inside the „Electrical characteristics”, on page 279 of the datasheet:

27.2 DC Characteristics						
$T_A = -40^{\circ}\text{C}$ to $85^{\circ}\text{C}$ , $V_{CC} = 2.7\text{V}$ to $5.5\text{V}$ (Unless Otherwise Noted)						
Symbol	Parameter	Condition	Min	Typ	Max	Units
$V_{IL}$	Input Low Voltage except XTAL1 and RESET pins	$V_{CC}=2.7 - 5.5$	-0.5		$0.2 V_{CC}^{(1)}$	V
$V_{IH}$	Input High Voltage except XTAL1 and RESET pins	$V_{CC}=2.7 - 5.5$	$0.6 V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
$V_{OL}$	Output Low Voltage <sup>(3)</sup> (Ports A,B,C,D)	$I_{OL} = 20\text{mA}$ , $V_{CC} = 5\text{V}$			0.7	V
		$I_{OL} = 10\text{mA}$ , $V_{CC} = 3\text{V}$			0.5	V
$V_{OH}$	Output High Voltage <sup>(4)</sup> (Ports A,B,C,D)	$I_{OH} = -20\text{mA}$ , $V_{CC} = 5\text{V}$	4.2			V
		$I_{OH} = -10\text{mA}$ , $V_{CC} = 3\text{V}$	2.2			V

Figure 12: Typical input and output voltage levels (part of the datasheet)

The input will indicate logic 0, if the connected voltage is bigger than -0.5 V, and smaller than  $0.2 \cdot V_{CC}$ , in our case  $0.2 \cdot 5\text{V} = 1\text{V}$ .

The input will indicate logic 1, if the connected voltage is bigger than 3 V, and smaller than 5.5 V, in our case.

In case of logic 0, it is guaranteed that the output is lower than 0.7 V, if the current, that flows through, is smaller than 20 mA.

In case of logic 1, it's guaranteed that the voltage of the output will be bigger than 4.2 V, if the current, that flows through, is less than 20 mA.

## THE PROGRAMMER DEVICE

In order to execute a series of instructions by the microcontroller, i.e. a program, this program must be implemented first and then loaded on the microcontroller. For these operations we need a PC, a proper programmer device and a microcontroller with appropriate supply voltage.

Several programmer devices are compatible with this microcontroller, while some of them are capable to complete more useful tasks as well. Let's learn about 3 programmer and debugger devices. We'll only need one of the three to program the microcontroller.

All three devices can be connected to the microcontroller in several ways according to the standards, we will use the one called JTAG, because, every task can be implemented with this, which will be discussed in this curriculum. We will connect the proper pins of the programmer device's JTAG plug to the appropriate pins of the microcontroller.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

### AVR Dragon

This programmer has to be connected to the PC or laptop through USB cable. As you can see in the following picture, a 10-pole ribbon cable has to be connected to the 10 pole „JTAG” labelled plug, then it has to be connected to the microcontroller as instructed. (Detailed English description is available for the AVR Dragon on the website.)



Figure 13: AVR Dragon (part of the datasheet)

You have to be careful during connecting the ribbon cable, because pin 1 of the ribbon cable (it's usually marked with a triangle) has to be connected to pin 1 of the pin row which can be found on the programmer.

### ATATMEL-ICE-Basic

This programmer has to be connected to the PC or laptop through a micro USB cable. A proper 10 pole ribbon cable has to be connected to the „AVR” labelled, 10 pole, double density plug, then this has to be connected to the microcontroller, the way it's written down later on.



Figure 14: ATATMEL ICE Basic

### ATATMEL-ICE-Basic-PCBA

The ATATMEL-ICE-BASIC-PCBA is the previous device's version without case, so the handling of the connectors is the same. Devices without case are much more sensitive to the different physical impacts, for example mechanical impacts, humidity, electronic discharges (ESD, in other words electrostatic discharge).

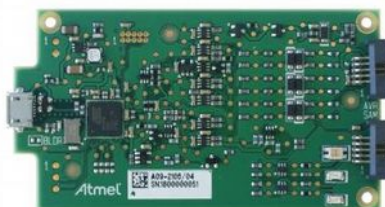


Figure 15: ATATMEL-ICE-Basic-PCBA



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

## JTAG plug and the microcontroller

Connection of the JTAG programmer and the microcontroller is made easy by an adapter shown in the picture below:

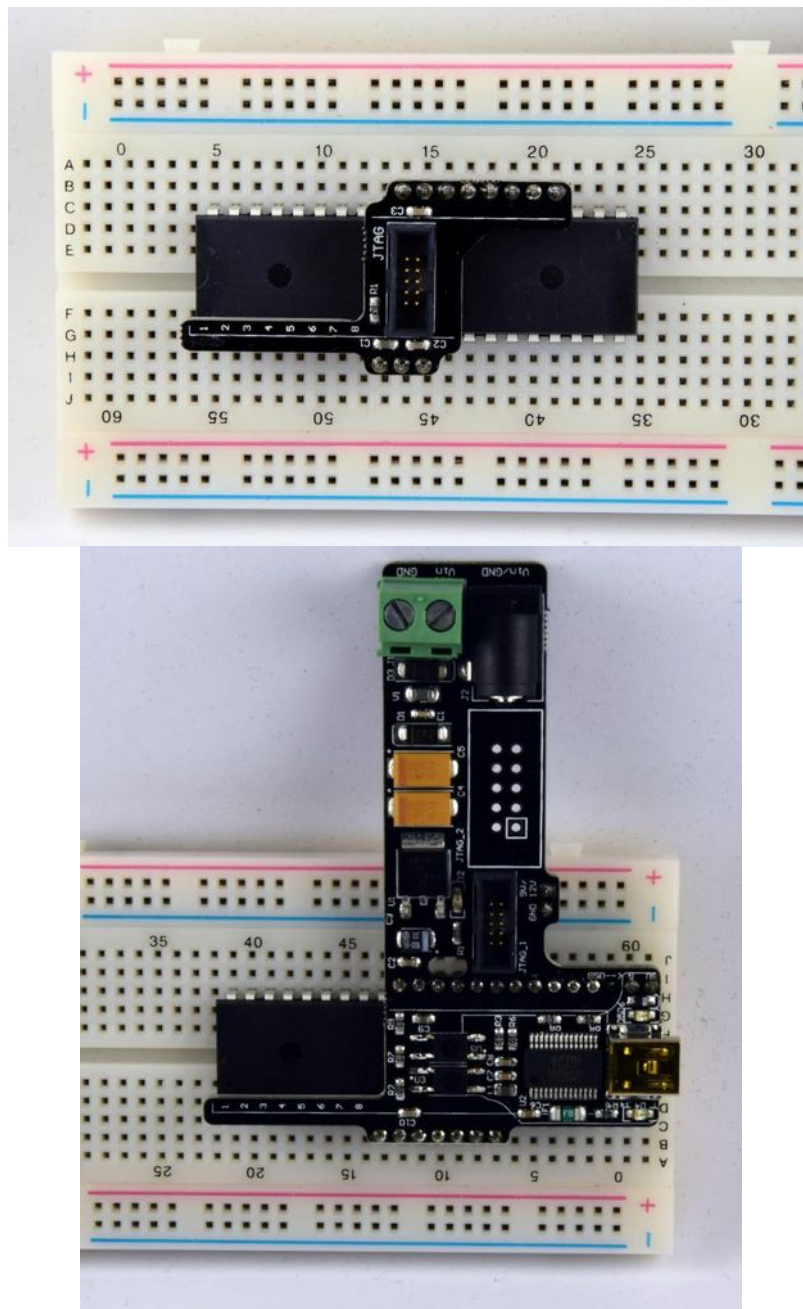


Figure 16: Different programmer adapters

Press the adapter into the breadboard, over the microcontroller, according to the markings. Make sure, that the label 1 of the adapter is over the microcontroller's number 1 pin! The pin rows, which are on the adapter, are exactly connected to the 5 V, GND and JTAG pins of the microcontroller, and the wires of the printed circuit connect them to the ribbon cable plug, which is on the adapter.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

Connect the programmer to the proper JTAG plug. (the ATATMEL-ICE-Basic programmer should be connected to the JTAG\_1 labelled plug, the AVR Dragon should be connected to the JTAG\_2 labelled plug.)

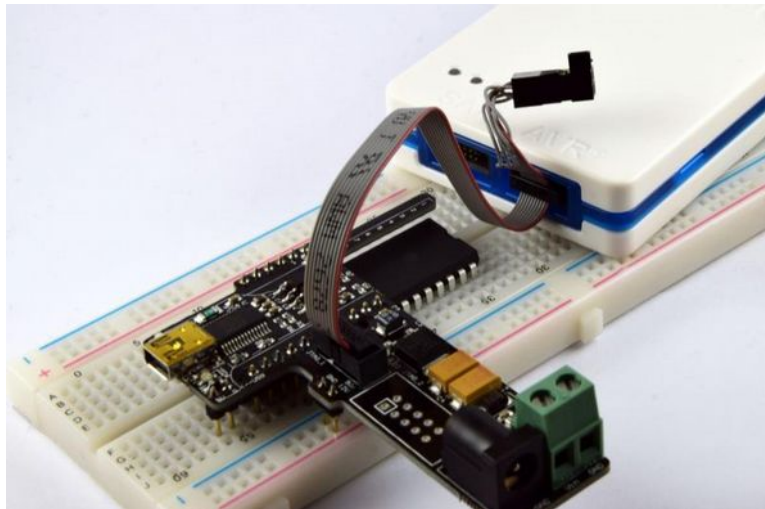


Figure 17: Connecting the ATATMEL-ICE-Basic programmer to the adapter

### Structure of programmer adapters

The adapter connects the microcontroller's proper pins to the appropriate pins of the programmer. We're talking about a few wires, but through them, we're not only able to program the microcontroller, but we can read out a great amount of data, during its operation. Later we'll see that it makes easier to implement plans.

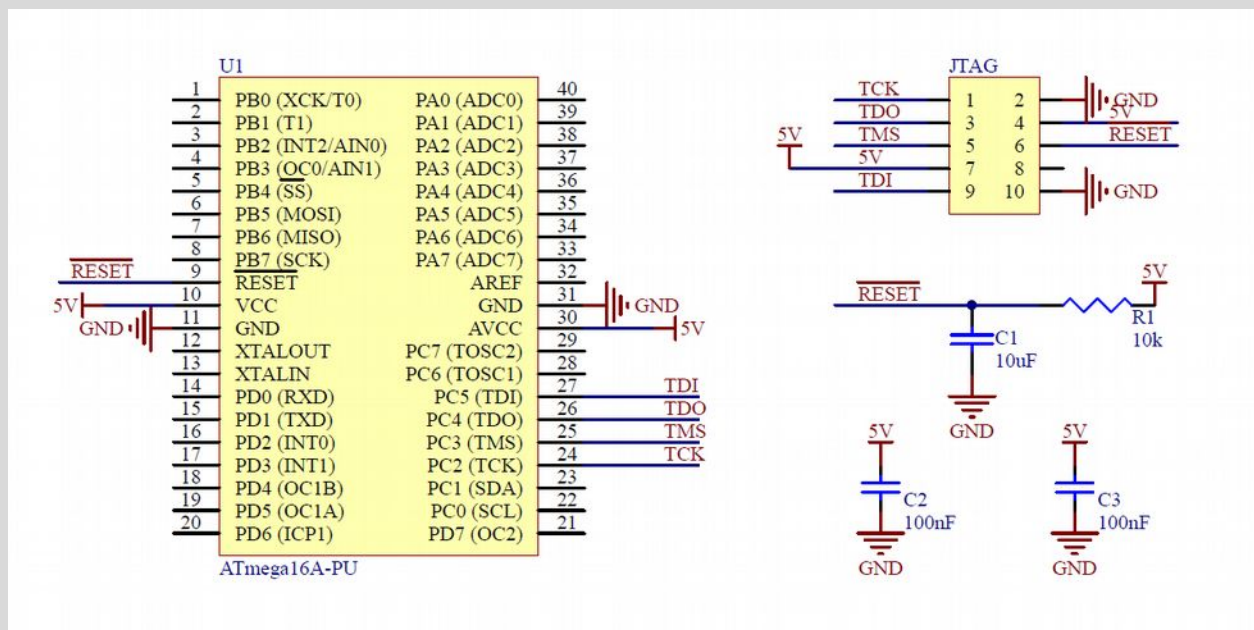


Figure 18: Wiring diagram of the JTAG programmer adapter

(We connect the GNDs together inside all device, because this is the reference voltage, at all the places a signal is considered to be low or high level comparing to this level.)



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

We connect the VCC wire to the programmer, in order to make sure that the programmer is able to check whether we provide the right supply voltage for the microcontroller. (In our case, it's 5 Volts). If the supply voltage were too low, then the programmer would do damage to the controller, that's why we need to check this automatically.

For example, the /RESET pin is used by the programmer, if he would like to reboot the controller. It's defined in the JTAG standard, that the TCK, TDO, TMS, TDI and TRST pins are used for communication, out of these, the ATmega16A uses only the first four.

## INTEGRATED 5V POWER SUPPLY IN THE ADAPTER

The larger adapter type gives further help for constructing the layout, because there is a circuit on it, which is very similar to the previously learnt 7805 circuit. It generates the 5 V power supply required by the microcontroller and connects the voltage to the right pins of the microcontroller through the breadboard (This can be checked with a multimeter).

We can connect a supply voltage to the input of the supply circuit in several ways, the point is that the voltage has to be between 7 V and 20 V, the ideal range is the 7-12 V.

We can connect a DC adapter - if its plug is proper - to the J2 marked plug. Or we can connect a 9 V battery to the green screw terminals, marked with J1. These cases have their disadvantages, that there isn't a current limit in the system, an unwanted short circuit, or badly connected components may damage something.

The ideal situation is, if we can provide voltage for the J1 terminal from a laboratory power supply, and we set the current limit of the power supply to a small value at first, about 70-100 mA.

When we connect the power supply, though we have to watch for the polarity. The adapter also contains a serial diode on the supply input, so in case of inverse connection, current won't flow, nothing will be damaged, while, in case of correct polarity, a green LED will light up on the adapter.

The USB connector, on the adapter, will be used in later parts of the curriculum. Make sure that, the section of the pin row, next to the USB connector, is not connected to anything on the breadboard for the time being.

## ADD AN LED

Let's connect an LED and a 330 Ohm resistor between the microcontroller's PA0 pin and the GND, so that, the LED lights up later, when 5 V appears on the pin! (In case of reverse polarity, the LED won't light up.)



*This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).*

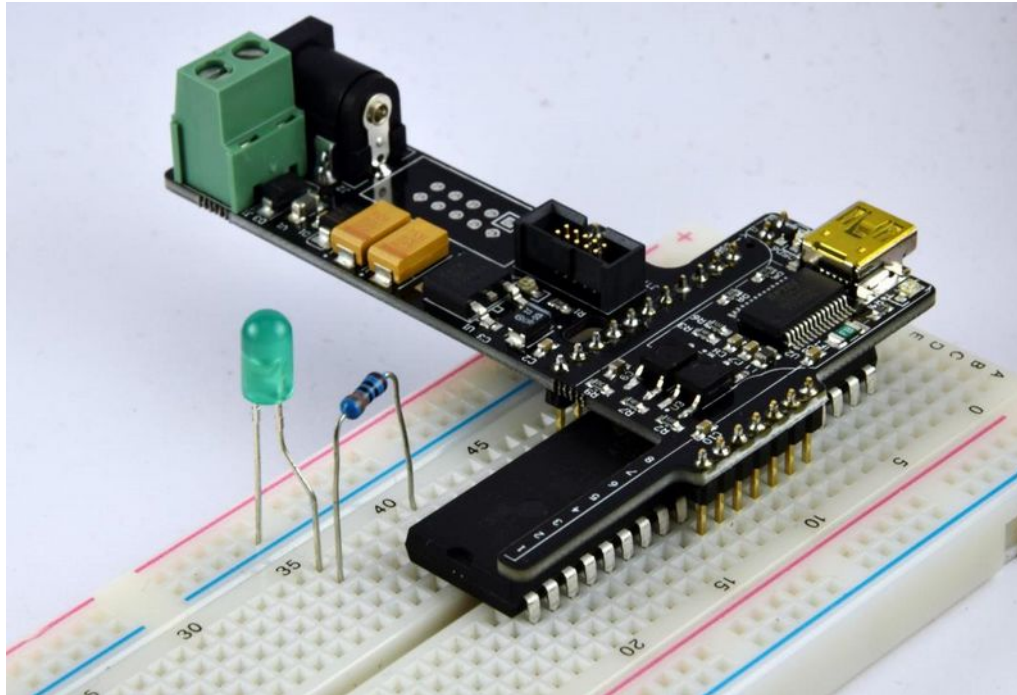


Figure 19: Connecting the LED to the PA0 pin of the microcontroller

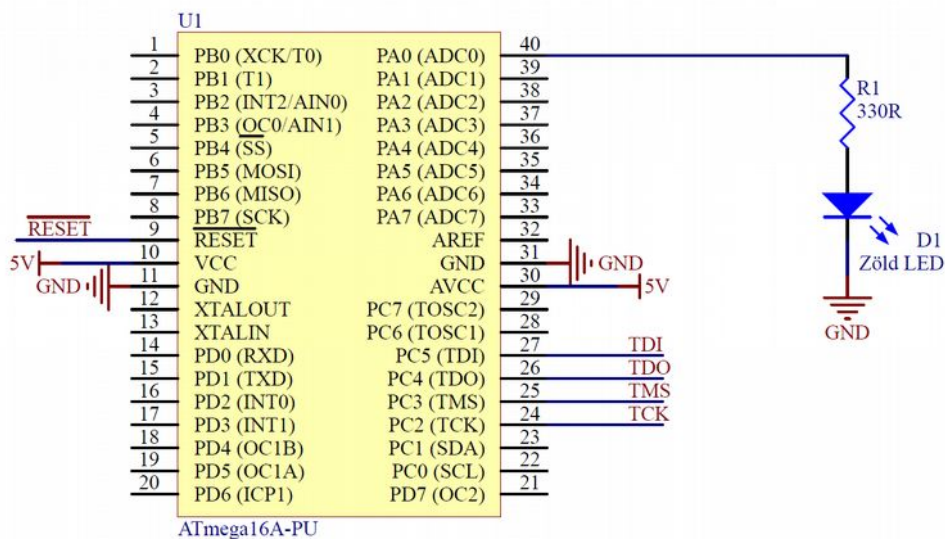


Figure 20: Schematic with LED

## INSTALLATION OF THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

Our hardware is assembled, but we still need to load the firmware (general name of small sized software, that run on microcontrollers), that will make it work properly. The Atmel company provides a software for this reason, which can be installed on a PC, this is called development environment.

During the curriculum, we will use the version v6.2.1563 of the Atmel Studio, which can be downloaded from the <https://www.microchip.com/mplab/avr-support/avr-and-sam-downloads-archive> website. The Microchip continuously develops the software, if you feel like that, you can use the latest version of the



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).

software too. However, in this case, we cannot guarantee that the example programs will run without errors, because of compatibility issues. In case you have problems during downloading or installing, you can find a detailed guide on the curriculum's website at the actual part.

### Programming the blinking LED software

For the sake of simplicity, the project, that contains the first software compressed, can be downloaded from the website. Understanding this software code and writing our own code will be started only in the later parts of the curriculum, however, now we'll learn how we can upload the code to the microcontroller. After we unpack the file, open the *CE9\_1\_LED\_blink\_delay* Atmel Studio solution file.

Name	Date modified	Type	Size
Debug	02/12/2019 20:04	File folder	
Headers	02/12/2019 20:04	File folder	
Source	02/12/2019 20:04	File folder	
CE9_1_LED_blink_delay	02/12/2019 16:04	ATMEL Studio 6.2 ...	1 KB
CE9_1_LED_blink_delay.atsuo	02/12/2019 16:15	ATSUO File	17 KB
CE9_1_LED_blink_delay	02/12/2019 16:04	ATMEL Studio 6.2 ...	7 KB

Figure 21: Files belonging to our first project

Make sure, that the microcontroller's pins get the right supply voltage, and the programmer is connected properly to the microcontroller. Connect the programmer to your computer. If that's the first time you program the device, definitely choose the „Tools” menu and then the „Device programming” option.

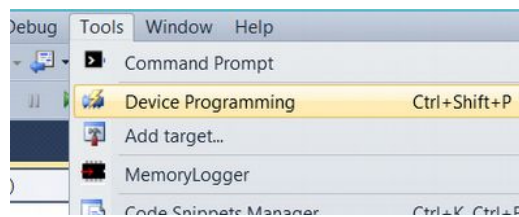


Figure 22: Here you can find the Device programming menu

Let's set the parameters of our system here.

- Tool: Atmel-ICE
- Device: ATmega16
- Interface: JTAG

Then press the button „Apply” to read some data directly from the microcontroller. We can see the default setup of the microcontroller if we click on „Fuses” inside this window. Here, let's find the clock signal frequency („SUT-CKSEL”), and change it to the 8 MHz clock signal from the internal oscillator. **Don't change any other values!** In order to make the change take effect, click on the „Program” button, then close the „Device programming” window.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).



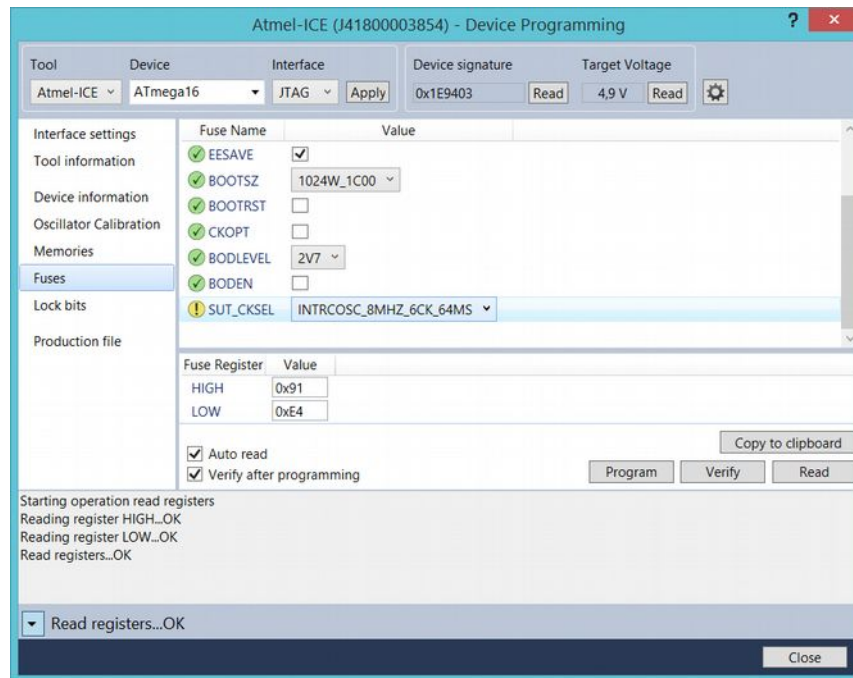


Figure 23: The Device programming window, and setting the clock signal frequency

The pre-written software can be easily loaded to the device, if we first click on the „Start Debugging” then the „Stop Debugging” buttons.

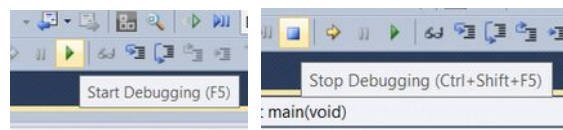


Figure 24: Uploading and launching the code

If everything is working correctly, then our computer doesn't throw any error message, and our LED blinks with a period of 2 seconds. After this, we now have a free path to understand programming and attaching a wide variety of peripherals!

## Useful tips

Never write accented letters, and special characters in a file's name or directory. Because of that, the developer environment can work unexpectedly, it may happen, that we cannot download our program to the microcontroller.

During development, always take small steps. If you change only a little from a previous working version, it will be easier to detect any errors in the new sections, whether they are hardware or software errors.



This project was supported by the European Commission. The content of this publication does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the author(s).