Basics of Arduino Laboratory

Welcome to the first laboratory. This laboratory is aimed at students with no previous experience with Arduino, robotics or electronics.

Prerequisites

Prerequisites are basic knowledge of programming in C/C++ (syntax, conditions, loops, functions). Also knowledge of prototyping electronics circuits is welcomed.

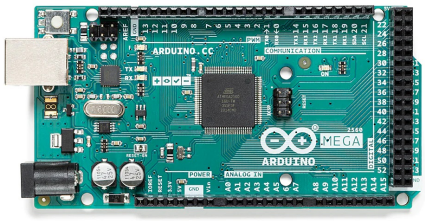
Goals of the laboratory

Goals of this laboratory are set to introduce you with basics of Arduino and prototyping electrical circuits. You will learn:

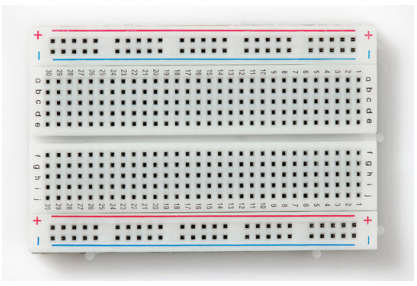
- Work with Arduino and Breadboard
- Creating simple electrical circuits using buttons, LEDs, potentiometers ...
- Using Arduino's Input/Output capabilities
- Reading/Writing digital and analog signal, using PWM
- Using Arduino's serial communication and debugging your code

Hardware used in this laboratory

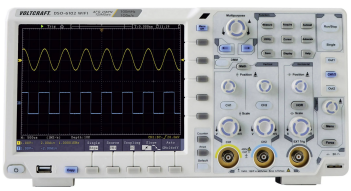
For this laboratory you will need following:



(a) Arduino Mega

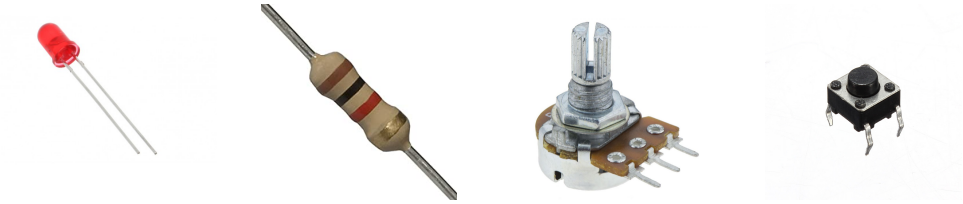


(b) Breadboard



(c) Oscilloscope

Figure 1: Core of the laboratory



(a) LED (b) Resistor (c) Potentiometer (d) Button

Figure 2: Electrical parts

Arduino Mega	-	Contains *ATmega2560* processor. Your code will run on it
Breadboard	-	Used for creating electronics circuits
Oscilloscope	-	Used for displaying electronic signals
LED	-	Light Emitting Diode
Resistor	-	Used for limiting current for electronics parts (like LED)
Potentiometer	-	Variable resistor
Button	-	Used for interrupting circuits

Theory

Arduino overview

Arduino is set of open source hardware and software projects used for fast prototyping of electronics circuits. It offers lots of **boards** (Arduino Mega, Arduino Uno, Arduino Micro, ...), **shields** (Ethernet shield, Motor shield, ...), **sensors** (Gas sensor, Hall sensor, Motion sensor, ...) and **components** (Universal PCBs, Jumping wires, Connectors, ...) that can be used for fast prototyping and learning basic concepts of electronics and programming.

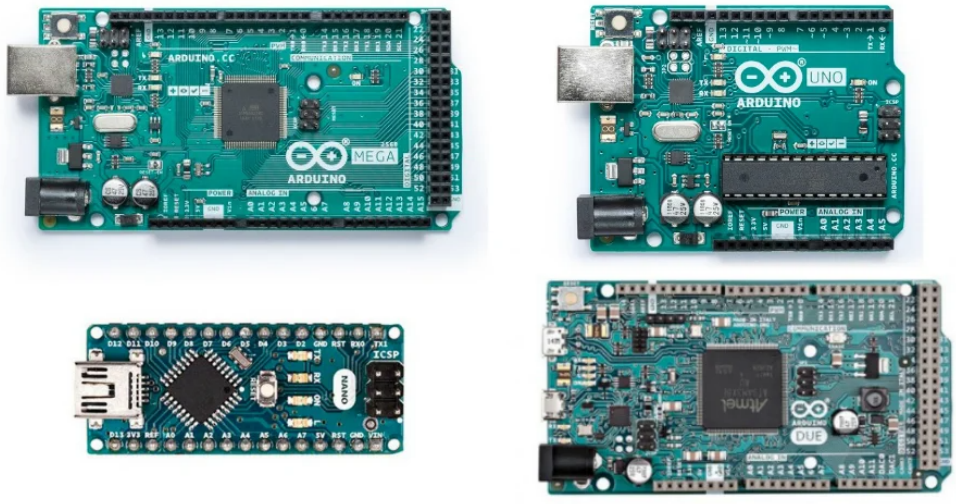


Figure 3: Prewiev of Arduino Boards

Arduino mega

Arduino board (later on just Arduino) used in this course will be Arduino Mega. It's one of the more powerful boards and offers great amount of **GPIO** - General Purpose Input Output pins.

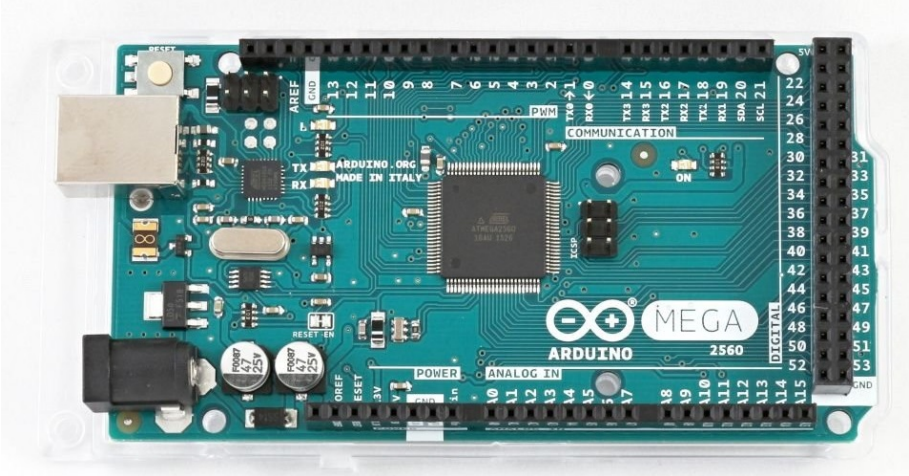


Figure 4: Arduino Mega used in this laboratory

Arduino uses USB type B for communication with PC and power (for power can be also used power-jack). It also offers 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports).

Microcontroller	ATmega2560
Operating Voltage	5V
Clock Speed	16 MHz
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
LED_BUILTIN	13

Table 1: Some of the specifications of Arduino Mega

Arduino IDE

Arduino also offers software for programming it's boards. It can be also programmed by other software, but most convenient way is to use Arduino software. It's called Arduino IDE . There are **2 versions** of it - **Arduino IDE** and **Arduino IDE 2**.

Arduino IDE is older version of these 2 and offers fully functional IDE. It's disadvantage is it's older design. Difference between Arduino IDEs and other editors is that Arduino IDE offers easy way to compile and upload code to the Arduino boards. **Verify/Compile** can be done by button with **Green tick** in left top corner and **Upload** can be done by button with **Green arrow** next to Verify button. Arduino IDE also offers **Serial monitor** and Serial plotter. Both of them are useful for debugging/looking at result from a code that runs on Arduino.

Figure 5: Preview of Arduino IDE

Arduino IDE 2 is newer and still being developed. It offers modern looking IDE, better Serial plotter etc. It's drawback is that it still has bugs in places and it's functionality is not fully polished.

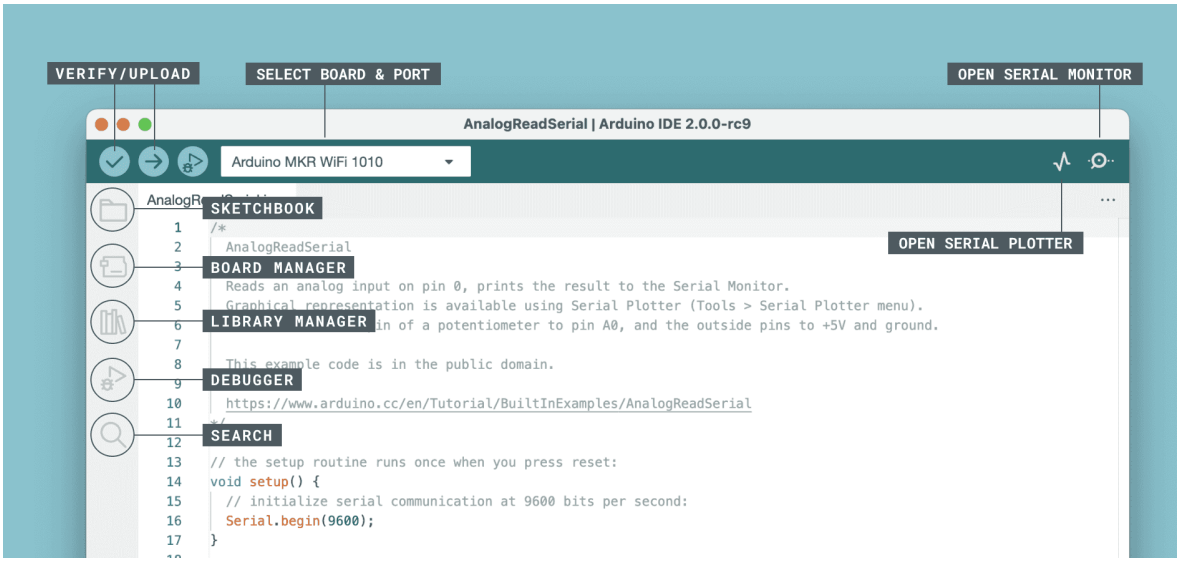


Figure 6: Preview of Arduino IDE 2

Both of the IDEs contain **settings for uploading** your code into Arduino board. You need to make sure that correct **Board** and **Port** is set before uploading. Communication port depends on which USB port you chose and Board is which Arduino board you are using. In this laboratory it is **Arduino Mega**.

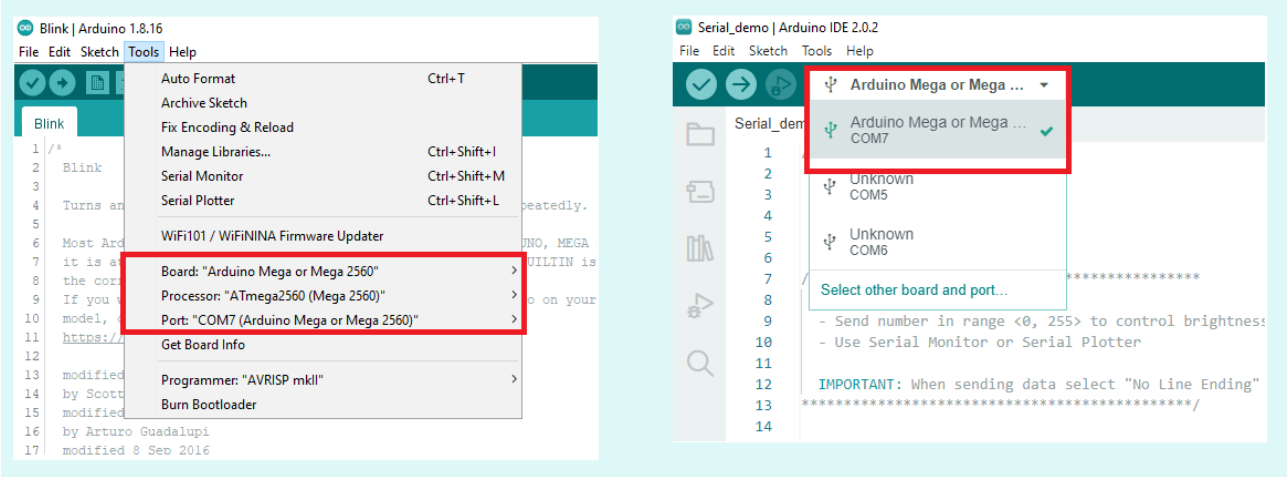


Figure 7: Setting for **Port** and **Board** in both IDEs

Breadboard

Breadboard is construction base used for solderless prototyping of electronics circuits. Even tough on the market is huge amount of variants, working principle of each of them is the same.

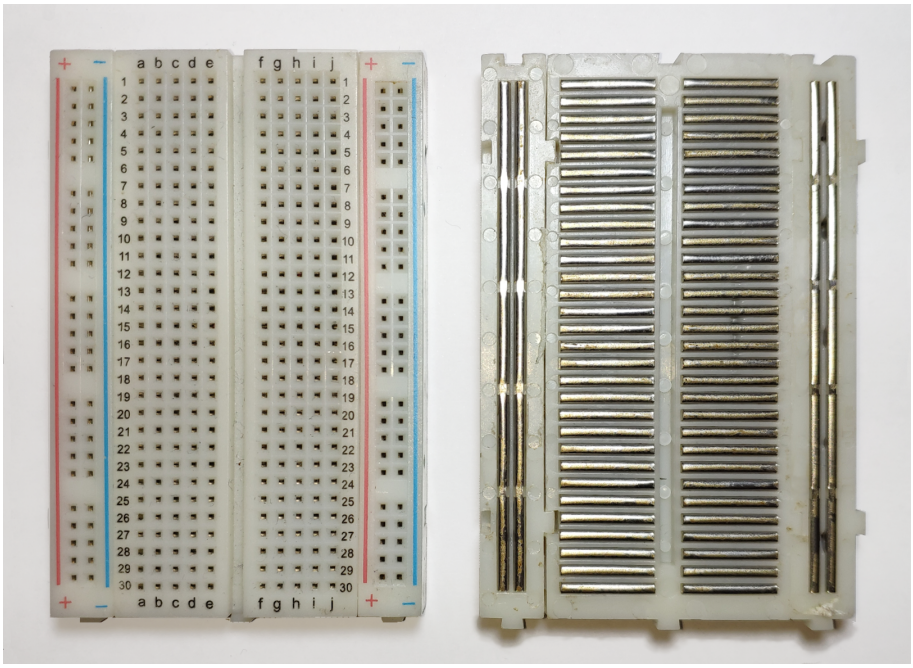


Figure 8: Breadboard

Breadboard consists of 2 types of rails - **horizontal** and **vertical**.

Vertical rails, usually at the sides (bigger boards can have them also in the middle), are used for *supply voltage* and *ground*. They are connected all the way trough the breadboard.

Horizontal rails are used for making the circuit itself. This kind of lines are also connected but not trough center divider.

Connections are demonstrated at Figure 8 where can be see disassembled breadboard from back-side.

Resistor

Resistor is basic passive electronic part. It's goal is usually to **limit current** flow trough circuit or to **accumulate voltage drop**. It has unit of *Ohm* [Ω].

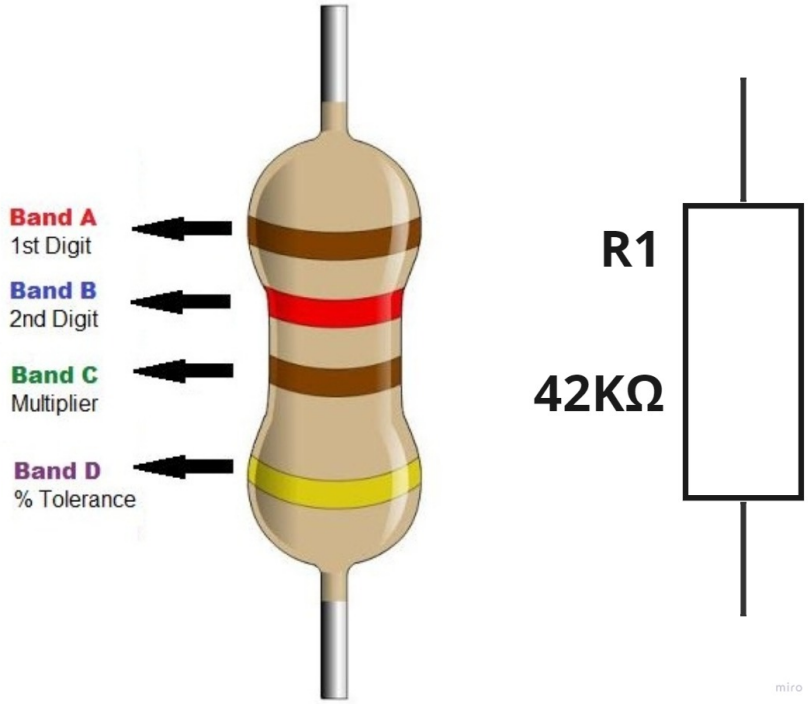


Figure 9: Resistor and it's shematic symbol

There is no point knowing the resistor without knowing the **Ohms law**. It is one of the core electrical laws and in it's principle is really simple. Ohms laws describes relationship between *current*, *voltage* and *resistance*. Ohm law says that current flowing trough circuit is proportional to the voltage applied with constant resistance.

$$I = \frac{U}{R}$$

Good analogy is shown at figure 69. Voltage tries to "push" as much current trough circuit while resistance (and resistor itself) limits

todo sem ten obrazok s panacikmi

todo dat ku kazdej suciatske aj schematicku znacku

Variable Resistor

Variable resistor - as the name suggests - is resistor which value can be changed. Unlike the resistor which has 2 pins, variable resistor(or potentiometer) has 3 pins as can be seen on figure 69. It has the same goal as resistor - limit current/voltage in the ciruit.

It's working pricnciple is simple and goes as follows: two side pins act together as normal resistor. Middle pin - slider

It is used for example as volume control at speakers. As you rotate the potentiometer voltage at slider changes - which can be translated by other electrical circuits into sound level.

LED

LED or Light Emmiting Diode is a type of diode that lights up when current flows trough it. Diode itself as electrical part is used for controlling the flow of current - it allows current to flow only one way. It has 2 pins Anode and Katode. In most applications there needs to be resistor in series with dioden, in order to limi current flow, to prevent destroing/burning the LED. LEDs are used for indication.

TODO image here

Push Button

Another simple electrical part covered in this laboratory is push button. Push button (or just button) acts like element that disconnets part of circuit by air gap. Therefore current cant flow. There are many types of buttons, but button used in this laboratory is momentary mechanical push button - meaning that when not pressed, button returns into its original state (either on/off).

TODO button img a scheme here

In many cases there is either pullup or pulldown resistor in combination with button. Function of these resistor is to ensure a known state for a signal. If there were no pulldown/pullup resistors and button would simply translate signal from one pin to other, in case of vypnuteho button output side would be not connected to anything. Pin would be in state of high impedance and readding analog or logic value would consiste of undefined behaviour.

pullup pulldown image here

Oscilloscope

Digital signal

Digital signal is type of signal that represents data as sequence of discrete values. Vast majority of digital devices in this world use binary values 1/0. This binary value is translated into level of voltage in electical circuits and high/1 can be 12V, 5V, 3.3V, or 1.25V. In case of Arduino mega used in this laboratory it is 5V. logic value 0 is usually in every case transalted onto 0 volts.

TODO image here

PWM

PWM or pulse-width modulation is a method of reducing the average power delivered by an electrical signal, by effectively chopping it up into discrete parts. It has 2 attributes - frequency an duty. Switching frequency has to be fast enough for controlled device (such as motor or LED) for the waveform to be percieved as smooth signal. Duty cycle describes how long from one period is signal "ON" or in logic state 1. 0% duty cycle means no energy is deliverd and signal is at constant 0, 50% duty cycle means signal is half of the cycle in logic 1 andd half in logic 0. And 100% duty cycle means signal is constant 100%. Duty cycle can be in any range from 0 to 100%

TODO img here

Analog signal

Analog signal, unlike digital signal, can nadobudat have any arbitrary value in given range. In case of Arduino mega, given range is <0V,5V>. In order for arduino to be able to work with analog values, voltage level needs to be translated into digital signal. Component that is used for this is called Analog to Digital Converter - or ADC shortened. Arduinos converters resolution is 10-bit, meaning that analog value is stored in 10-bit number. Range for 10 bit number is <0,1023>.

Analog signal img here a mozno aj obrazok na to nejaky range 0-5 0-1023

Programming Arduino

Arduino can programmed by using Arduino IDE and programming language needed to know is C/C++. Arduino offers many built-in functions tahat simplyfy work with its peripherials, therefore the learnign curve is not that steep.

Basic frame of code contains 2 function void setup() and void loop(). As their names suggest, setup() function is used for settuping values, pins, communications, etc. This function runs only once at begginig of each code run. Function void loop() runs after setup() in endless loop. This is the place where you write main part of your code.

TODO here image if basic codde

For each of the mentioned signals - Digital, Analog, PWM - Arduino has built in functions. Arduino can either read of write these signals. Before reading or writing an signal using any pin, it has to be initialized as either input or output. This initialization is usually in setup function. Table bellow contains list of functions necessary to know for basic work with Arduino.

pinMode(pin, INPUT/OUTPUT)	-	Sets pin to be either INPUT or OUTPUT (choose one). Usually used in setup()
digitalWrite(pin, HIGH/LOW)	-	Writes digital value 1/0 (0 Volts, 5 Volts) to pin.
digitalRead(pin)	-	Reads digital value from pin. Returns read value
analogWrite(pin, value)	-	Writes analog value to pin. Value has to be from range <0, 255>. This range corensponds to <0 Volts, 5 Volts>
analogRead(pin)	-	Reads analog value from pin. Returns read value. Value is from range <0, 1023> which corrensponds to <0 Volts, 5 Volts>
delay(value)	-	Stops the program for given number of milliseconds
delayMicroseconds(value)	-	Stops the program for given number of microseconds

Exercises

IDEAS: spravit cheat sheet kde by boli vsetky funkcie ktore sa postupne beru (ako napr pin mode a tak)

TODO: nazvy suborov, cesty pomocou bold

TODO: spravit prehlad toho co sa bude brat

HC-SR04 - Ranging

Measure distance using HC-SR04 sensor.

1. Open file **HC-SR04_ranging.ino** with Arduino IDE using **File -> Open...**
2. Take a look at **HC-SR04_scheme.png** and create circuit as it is shown on image
3. Follow the instructions inside and write your own code that reads length of pulse and converts milliseconds into centimeters
4. Upload your code by clicking on Upload button 

or pressing Ctrl+U
5. Test the sensor by pointing it to different sides and measuring different distances

Appendix

TODO - farby kablikov dorobit stuff na osciloscope sensory na koniec

mozno sem schemy zo vsetkych cviceni?

todo arduino mega scheme

ideas/feedback aby som nezabudol

V labu 1 bych přidal i info o VSCode a Platformio, Arduino IDE 2 a třeba i Arduino CLI, osciloscop

"v tomto díle uvidíte" a v dalších labinách i "v minulých dílech jste viděli"

spravit nejaky feedback papier na ktory by na kazdom cviku napisali co sa pacilo/bolo hard a pod

todo pozriet sa na jitter sonaru