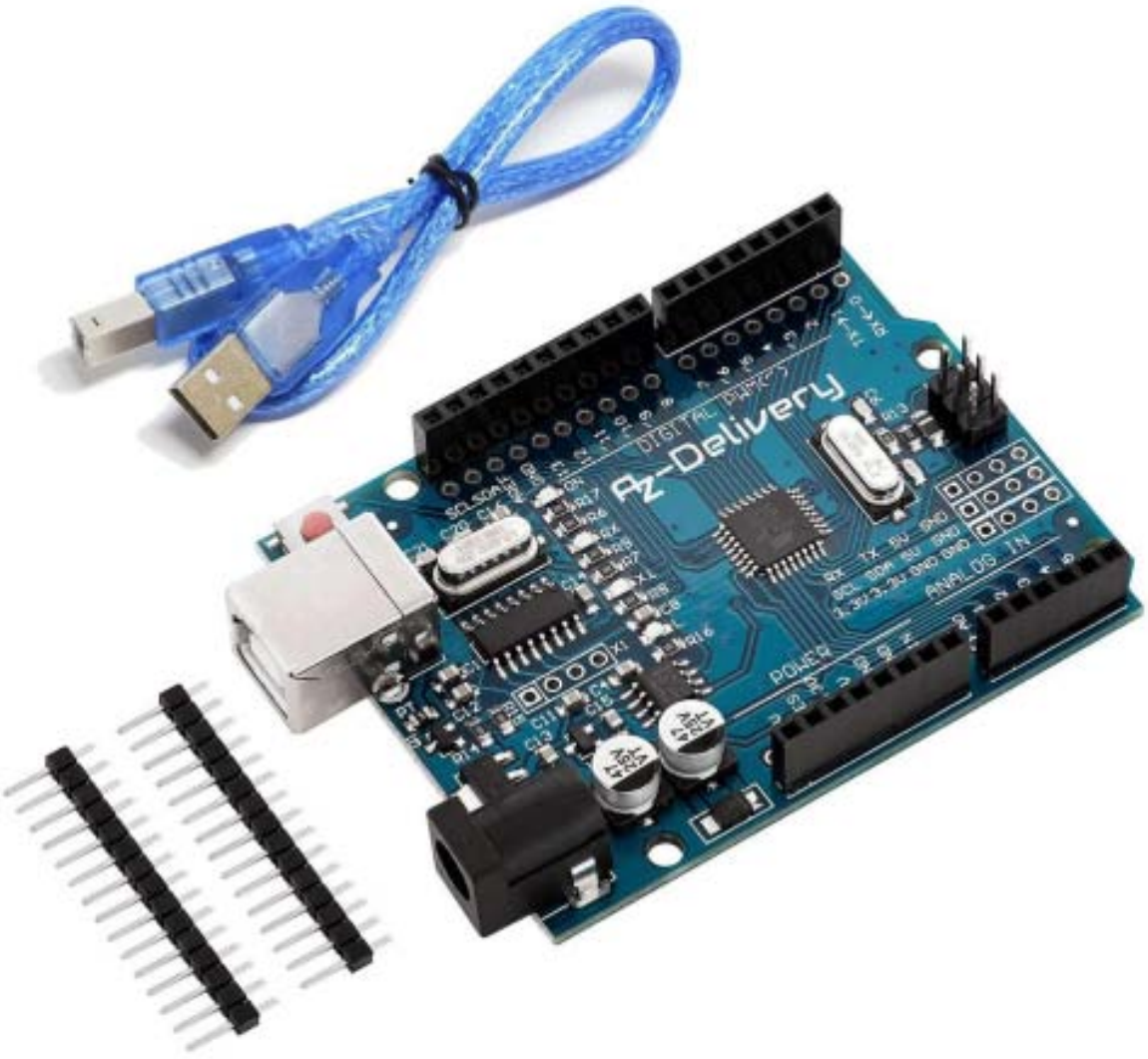

AZ-Delivery

Table of Contents

Introduction

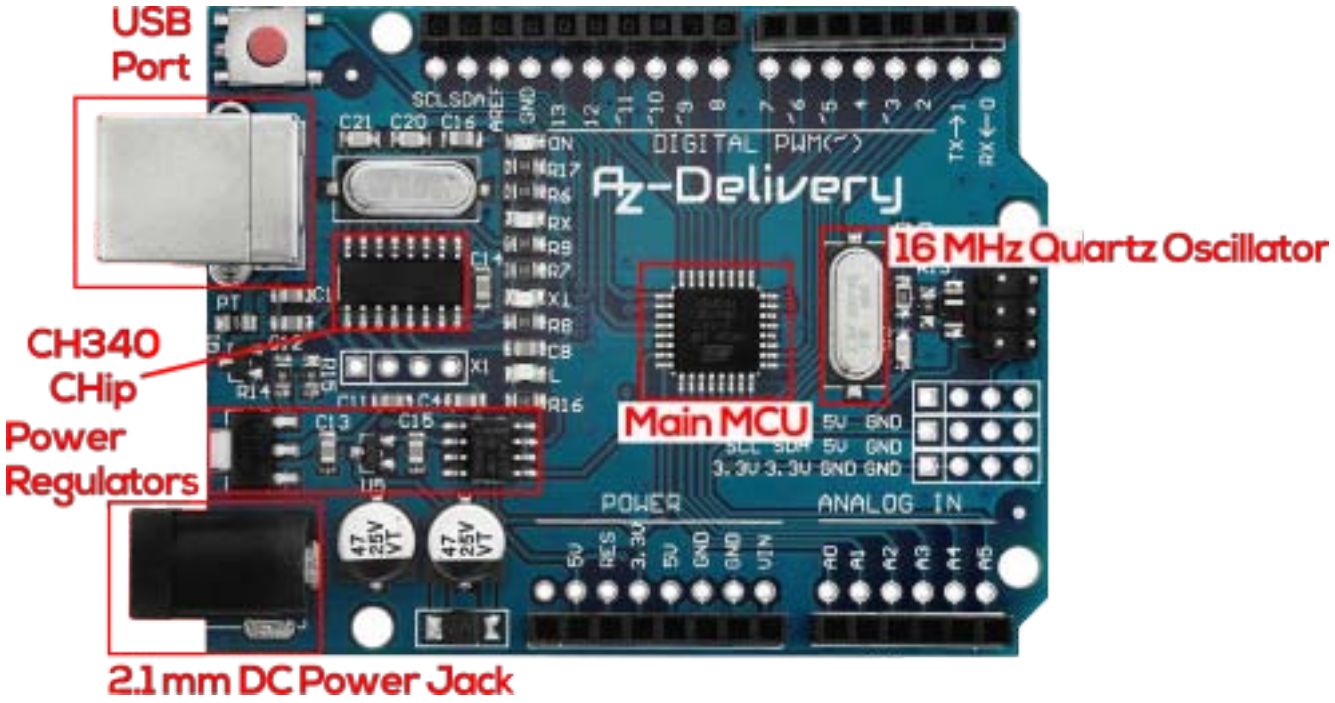
This handy Microcontroller Board is intended for electronics learning or prototyping and programming. This Board is actually a microcontroller, but assembled in a way that you don't need extra components with it. If you use single microcontroller, you will need to build stable DC power supply, and external programmer, and reset circuit, and many other things. The most powerful thing about this microcontroller board compatible is that there is Arduino IDE (Integrated Development Environment) with endless number of code examples already written for it, in a way that everyone can understand. There is no need for you to learn internal working of onboard microcontroller in order to programm it. Just connect your Microcontroller board, via USB cable to your PC, install and start Arduino IDE, search and upload programs that you need, to your board and that's it. There are endless code and library examples already written online, you just need to search for it. Also there are numerous other boards, like shields, or many sensors built in a way so you can easily connect them to your Microcontroller board. Just search our online store az delivery.de and you'll find more than you need.

Specifications

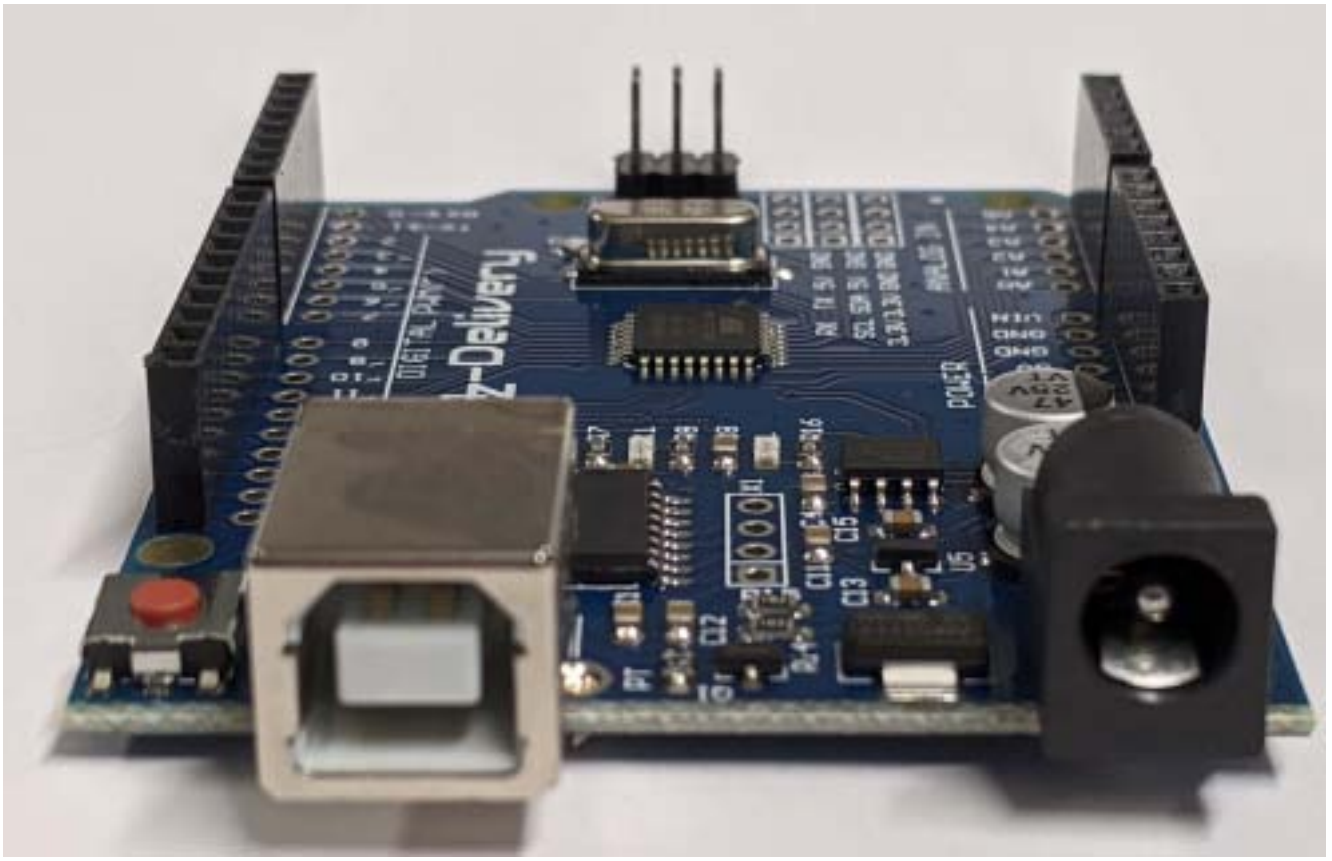
Operating voltage	5V
Input voltage	7-12V
Microcontroller	ATmega328P
Digital I/O Pins	14 (6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	32KB (0.5KB used by bootloader)
SRAM	2KB (ATmega328P)
EEPROM	1KB (ATmega328P)
Clock Speed	16MHz
Dimensions	70x55x13mm(2.7x2.1x0.5in)

Features

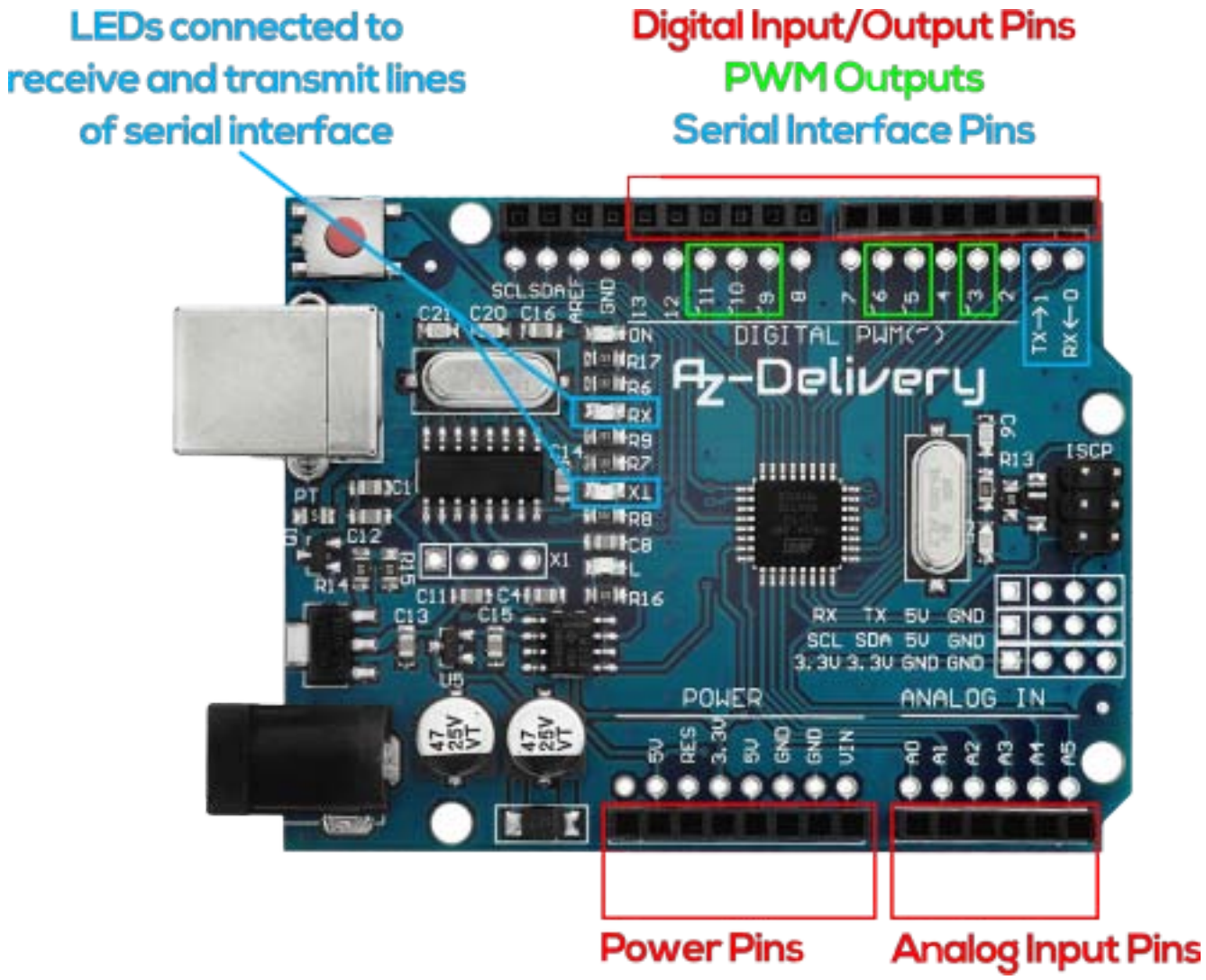
This Microcontroller Board features main microcontroller ATMega328P with 16MHz quartz oscillator. It uses CH340 chip to make communication like USART serial interface but via USB. CH340 chip connect PCs USB port with USART serial interface of microcontroller and thus enableS to program microcontroller via USB port.



AZ-Delivery Microcontroller Board features DC voltage regulators for both +5V and +3.3V. The external DC power supply can be connected to the DC Power jack on board with voltage in range from 7V up to 12V, and voltage regulators will lower and stabilize it to the +5V and +3.3V.



This Microcontroller Board is built in a way that separates digital I/O pins from analog input pins. So there are 6 analog input pins, and separated 14 digital I/O pins. 6 of 14 digital I/O pins can be used as PWM outputs (Pulse Width Modulation). Those pins are labeled with tilde sign “~” (D3, D5, D6, D9, D10, and D11).

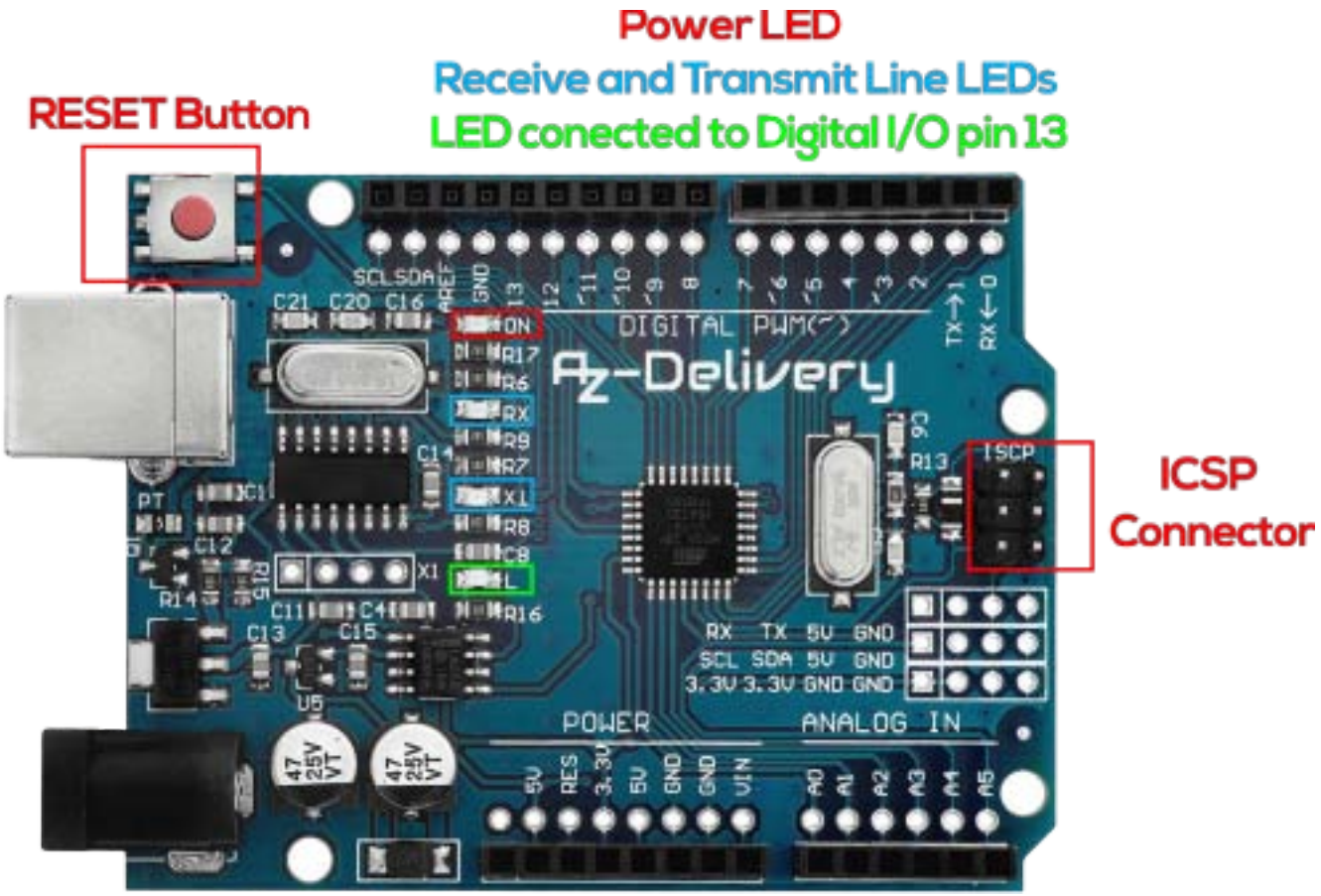


Digital I/O pins 0 and 1 are connected to receive and transmit lines of USART serial interface respectively. These digital I/O pins cannot be used as digital inputs or outputs, because Serial Interface is used every time when a new program is uploaded into this microcontroller board.

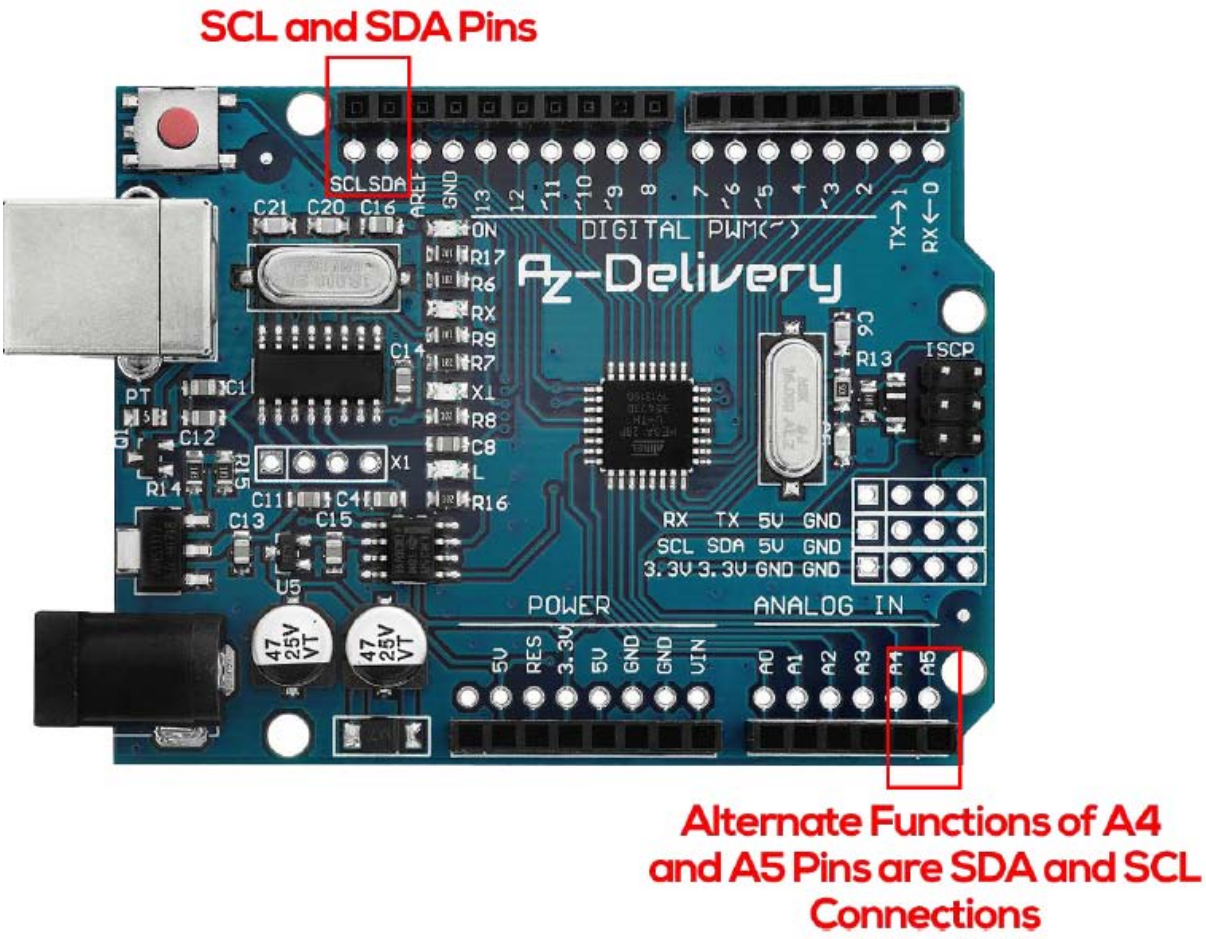
Connections for these analog and digital pins on these boards are established by female or male headers (like on image on previous page).

This Microcontroller Board has on-board RESET button for main microcontroller, ISCP connector which is used for programming the ATMega328 externally via some other programmer, and four on-board SMD LEDs.

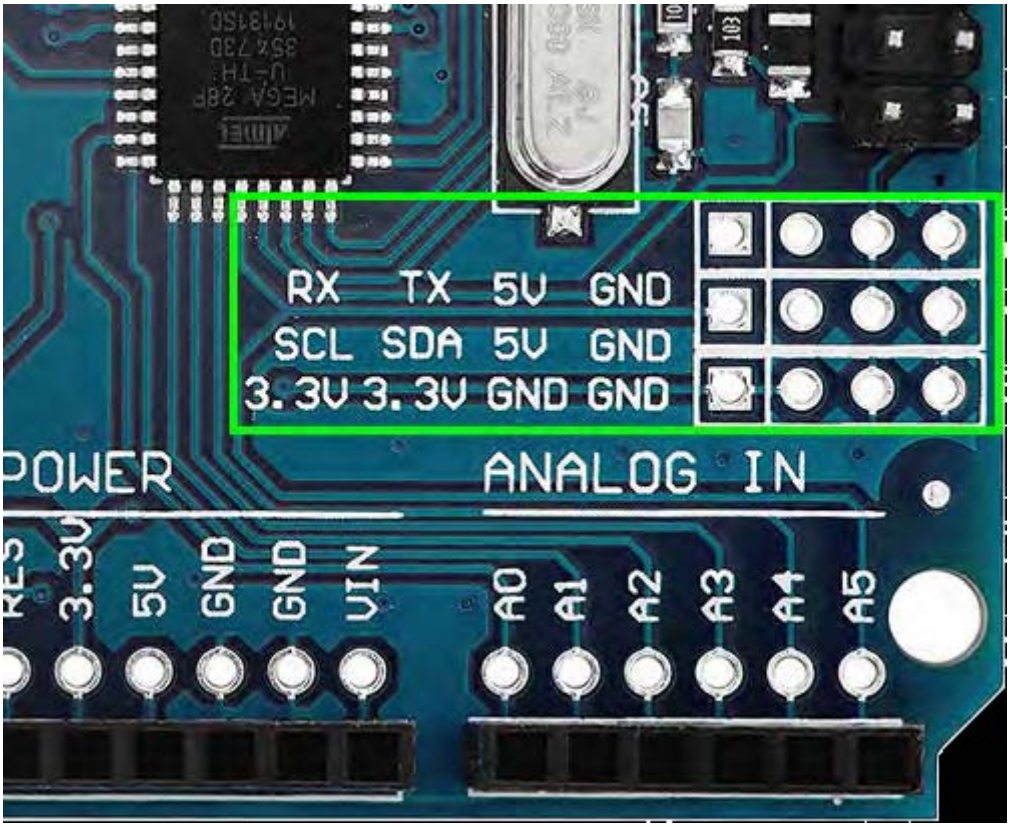
One LED is turned on when connection is established between this board and the power supply. One LED is connected to the digital I/O pin 13, and other two are connected to receive and transmit lines of USART serial interface, and flashing when those lines are used.



AZ-Delivery Microcontroller Board features 2 more additional pins, SDA and SCL, which are connected to the A4 and A5 pins respectively. These pins (A4 and A5) are used in I2C interface communication (another name for it is: TWI - Two Wire Interface).



This handy board provides another 3 extra 4 pin male header connectors. The connectors are shown on the image bellow:



Communication interfaces

Digital I/O pins D0 and D1 have alternative functions. They are connected to receive and transmit lines of serial interface.

There are two more communication interfaces supported by ATMega328P microcontroller, Serial Peripheral Interface - SPI and Inter-Integrated Circuit interface - I2C (or TWI - Two Wire Interface)

For SPI interface digital I/O pins D10, D11, D12 and D13 are used. Their functions are SS, MOSI, MISO and SCK respectively.

For I2C interface analog input pins A4 and A5 are used, or two additional pins SDA and SCL. Their functions are SDA and SCL respectively.

Power Pins

Power pin header from right to left:

VIN - This is voltage input pin, it server as another external power supply for main microcontroller when it is not powered by USB port **GND** - GrouND, 0V

GND - GrouND, 0V

5V - voltage output, +5V

3.3V - voltage output, +3.3V

RES - RESET, drive this pin to LOW (connect it to GND) to reset main microcontroller

5V - This pin is called IOREF and servers as voltage output, actually it is voltage reference pin for external boards (or shields) connected to the ATmega328p board. It tells shields on which logic level board work (+5V for ATmega328p).

- Last pin has no connection (NC)

Voltage Limitations

Voltage Input Limits

Input power: to power this Microcontroller Board, either plug it in to a USB port, or use input voltage source from DC power jack or via Power pins header going to VIN pin. When powering the ATmega328p via the power DC jack or VIN pin, it has the following input voltage limitations:

Recommended input voltage limits (DC power jack): 7-12V. These input voltages can be sustained indefinitely.

Absolute voltage limits for powering the microcontroller: 6-20V
Below 7V may cause the 5V levels on the board to waver, fluctuate, or sag, causing board instability and less accurate analog readings when using analogRead().

Sustained voltage levels above 12V will cause additional heating on the linear voltage regulator of the Board, which could cause it to overheat. Short periods, however, are fine.

Voltage limits on I/O pins

If its needed to read the voltage on this Board's digital or analog input pin, make sure it is between 0 and 5V. If it is outside these limits, the voltage divider can be used to lower the voltage. This scales the input voltage and allow analog or digital readings of voltages otherwise outside the allowed range. If the input signal is digital, and there is no need to take scaled analog readings, another technique is to clip (cut the top off of) the input voltage, rather than scale it. Since AVR microcontrollers have internal clamping diodes, this can be done by simply adding a single resistor in series with the pin. By adding a 10kΩ resistor in series with the input pin (any input pin) permits input voltages as low as -10.5V or as high as +15.5V.

Current Output Limits

Total maximum current draw from these Boards when powered from a USB port is 500mA. These Boards have a *resettable polyfuse* that protects your computer's USB ports from shorts and overcurrent. Total maximum current draw when powered via external power supply is 1A.

Note: If not powered by USB, the total 5V current limit coming out of the ATmega328p is limited by the on-board voltage regulator, in this case it cannot exceede the 1A.

Total maximum current per input/output pin is 40mA.
Sum of currents out of ALL input/output pins combined is 200mA!

Note: Despite the fact that your voltage regulator on this Board may permit up to 1A draw across the 5V and GND pins, the sum of all currents going into or out of the input/output pins (all Analog and Digital pins combined) of the ATMega328P microcontroller itself **should not exceed 200mA**.

How to set-up Arduino IDE

If the Arduino IDE is not installed, follow the *link* and download the installation file for the operating system of choice.

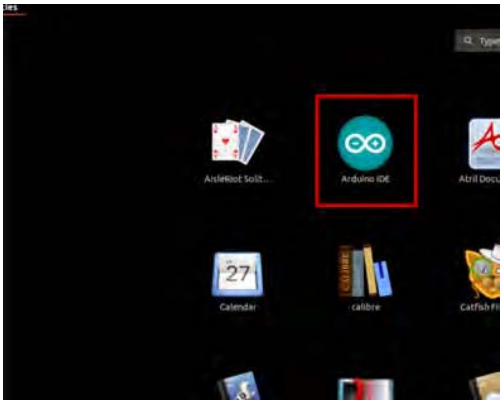
For *Windows* users, double click on the downloaded *.exe* file and follow the instructions in the installation window.

For *Linux* users, download a file with the extension *.tar.xz*, which has to be extracted. When it is extracted, go to the extracted directory and open the terminal in that directory. Two *.sh* scripts have to be executed, the first called *arduino-linux-setup.sh* and the second called *install.sh*.

To run the first script in the terminal, open the terminal in the extracted directory and run the following command:
sh arduino-linux-setup.sh user_name
user_name - is the name of a superuser in the Linux operating system. A password for the superuser has to be entered when the command is started. Wait for a few minutes for the script to complete everything.

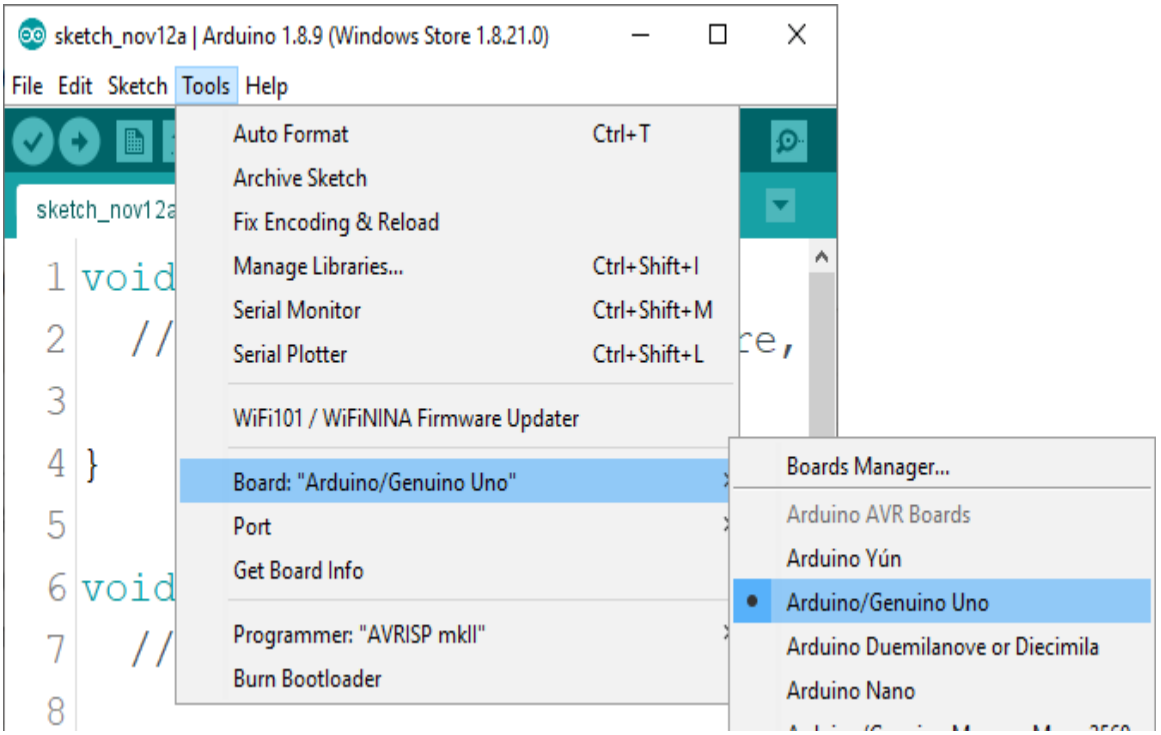
The second script, called *install.sh*, has to be used after the installation of the first script. Run the following command in the terminal (extracted directory): **sh install.sh**

After the installation of these scripts, go to the *All Apps*, where the *Arduino IDE* is installed.



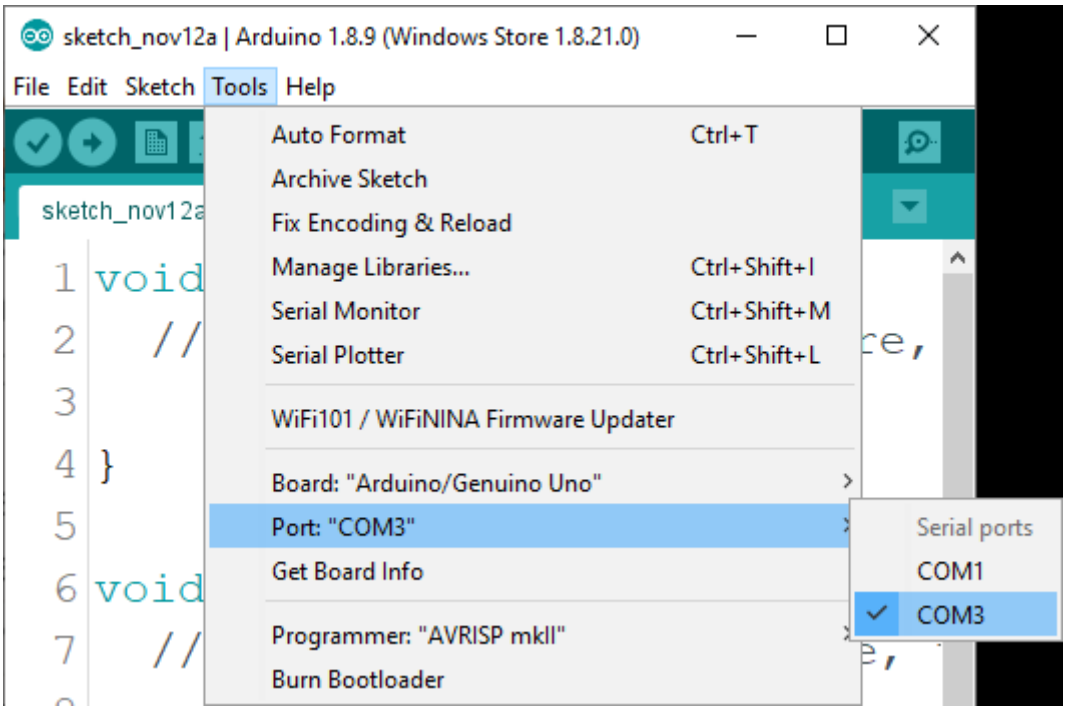
Almost all operating systems come with a text editor preinstalled (for example, *Windows* comes with *Notepad*, *Linux Ubuntu* comes with *Gedit*, *Linux Raspbian* comes with *Leafpad*, etc.). All of these text editors are perfectly fine for the purpose of the eBook.

Next thing is to check if your PC can detect a microcontroller board. Open freshly installed Arduino IDE, and go to:
Tools > Board > {your board name here}
{your board name here} should be the *Arduino/Genuino Uno*, as it can be seen on the following image:



The port to which the board is connected has to be selected. Go to: *Tools > Port > {port name goes here}* and when the board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

If the Arduino IDE is used on Windows, port names are as follows:



For *Linux* users, for example port name is */dev/ttyUSBx*, where *x* represents the integer number between *0* and *9*.

Additional setup

In order to use AZ-Delivery microcontroller board with Arduino IDE, follow a few easy steps. Before setting the Arduino IDE, the driver for the USB to Serial communication chip (CH340) has to be installed. If the driver is not installed automatically, there is a support page that contains the drivers for Windows/Mac or Linux and can be chosen depending on which one is used. Drivers can be downloaded from the following link.

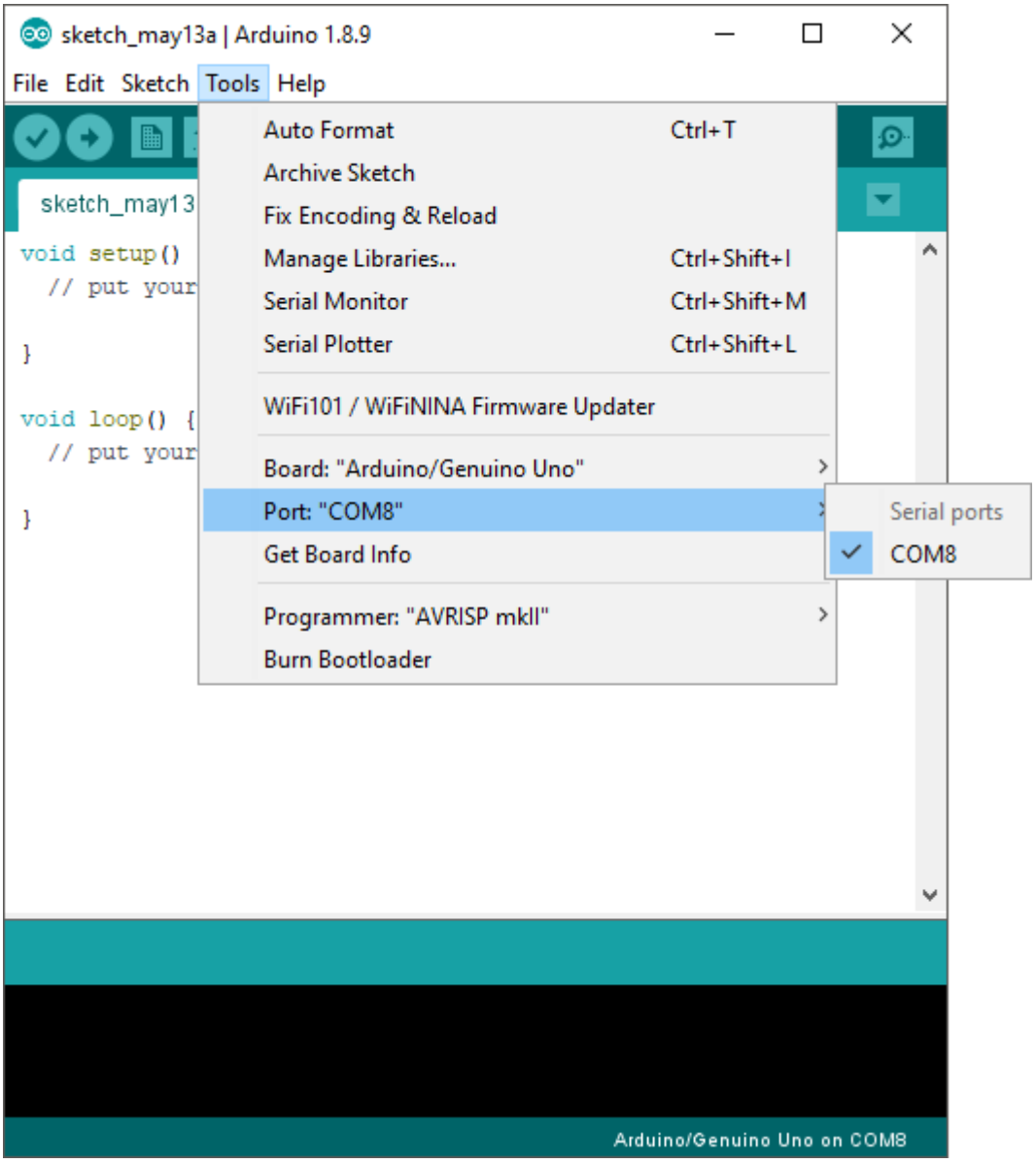
Sketch examples

When Arduino IDE is opened the example is called empty sketch. A sketch is program example, where the code is written. It has two essential parts, setup() function and loop() function, and it can have any number of other functions as well.

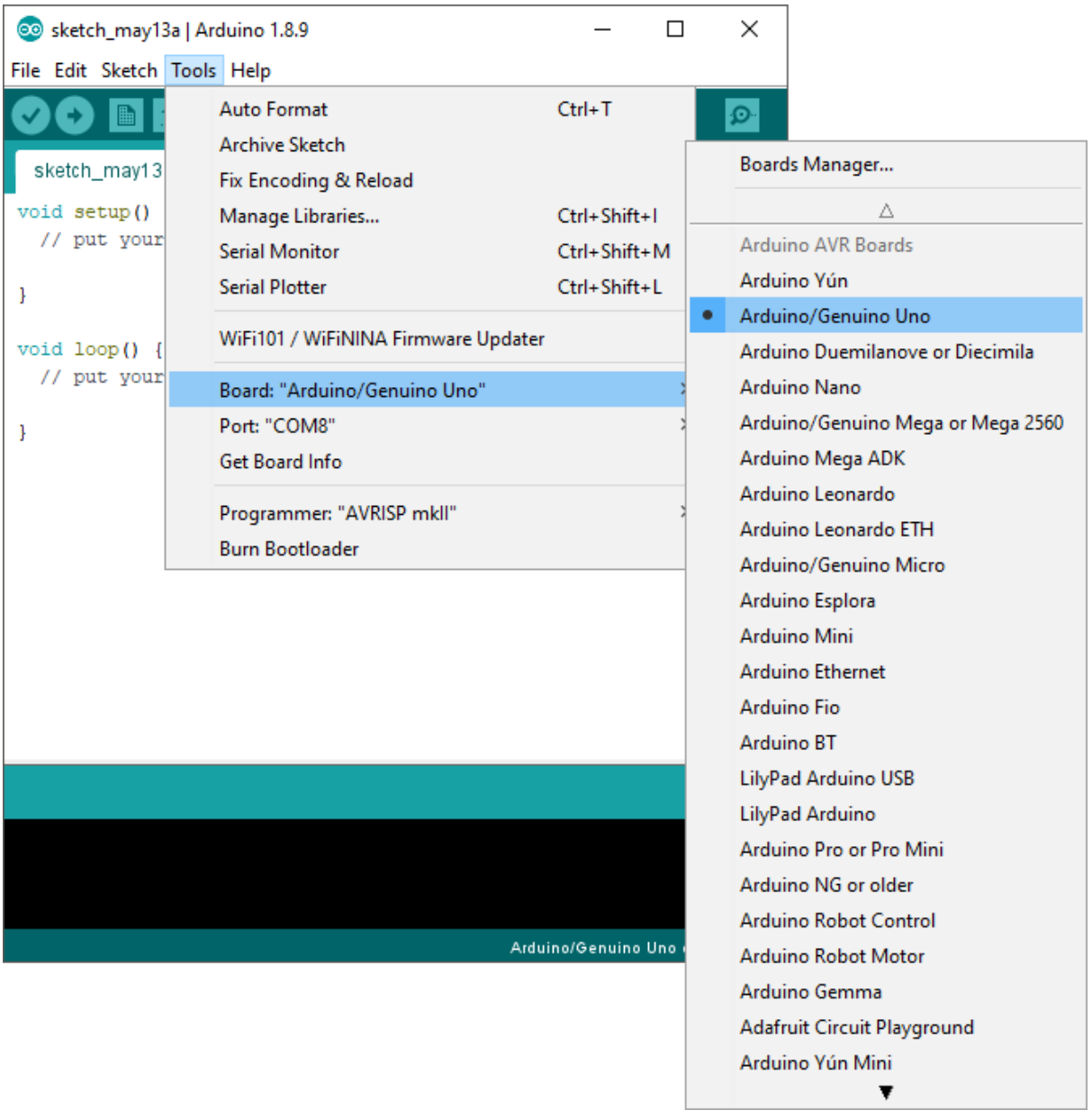
setup() function runs only once, at the beginning of program execution, after powering up the board, or when the the board is reset. In this function setup makes all initializations, for example declare the state of digital input/output pins or setting up analog input pins, or setting up serial interface for serial communication, etc.

loop() function runs after setup() and it runs indefinitely, over and over again, so called *endless loop* function. Actually it runs all time while the board is connected to the power. This is because programs in electronics devices should never reach the end, because if that happen that means that device is as good as turned off. Here the logic is written, algorithms on which the application for ATmega328p board works.

When the AZ-Delivery microcontroller Board is connected via USB cable to the PC, first thing in Arduino IDE is to select the USB port on which our Microcontroller Board is connected.

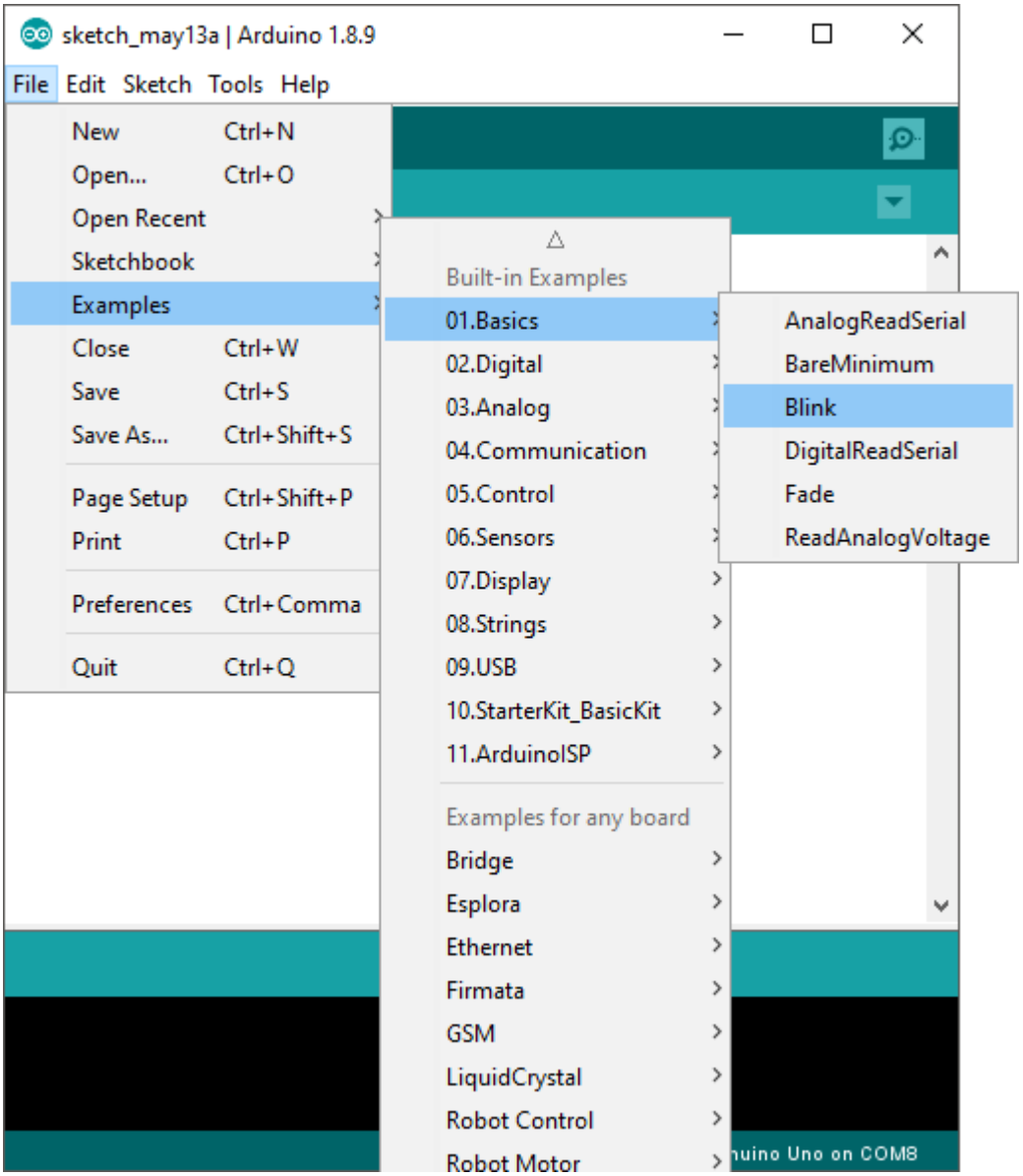


Then select the microcontroller board that is used.

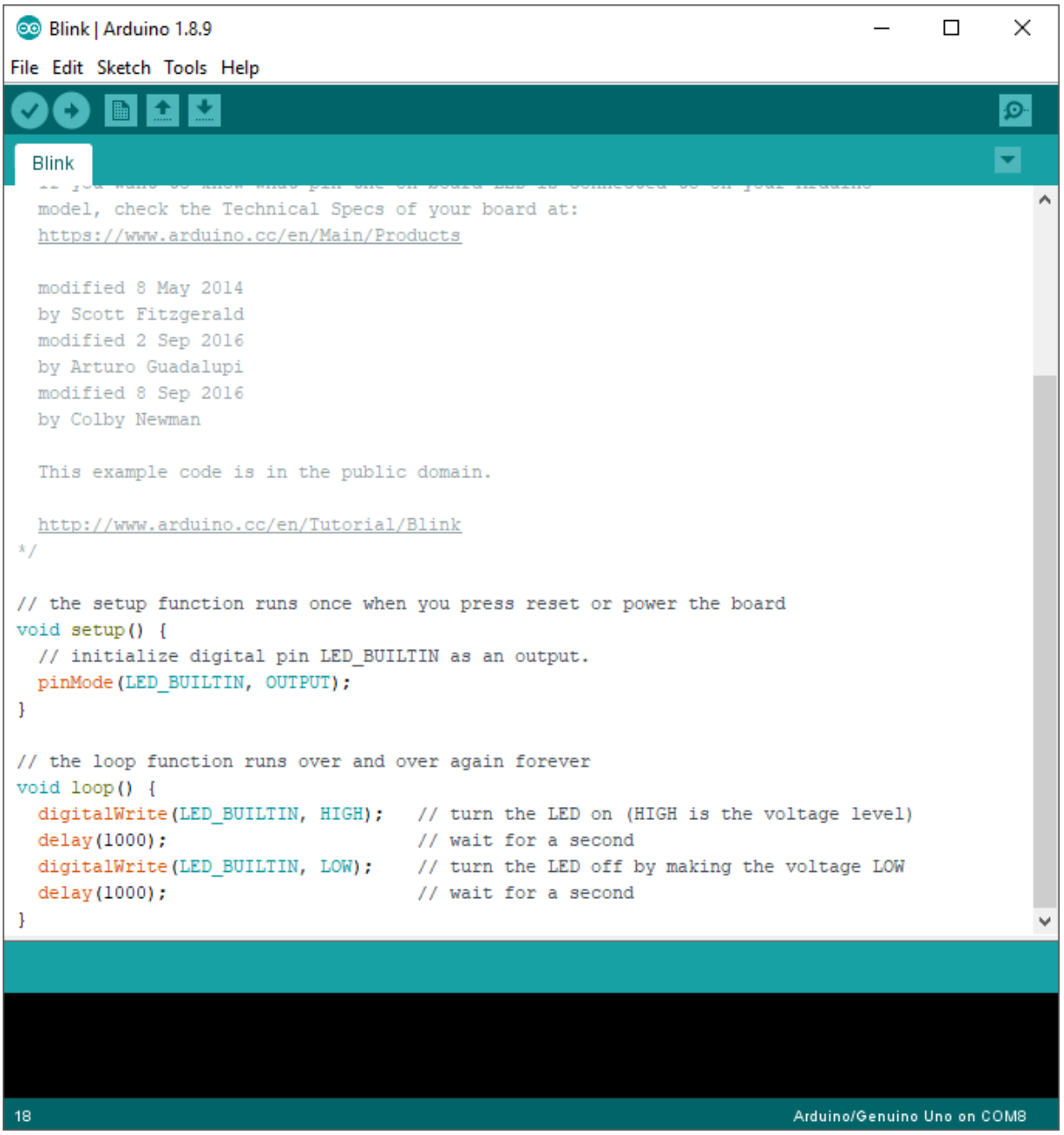


Application example

And now the programming can start. Arduino IDE comes with many prewritten sketch examples, which can be used. Here the *BLINK* sketch example is used. Go to *File* > *Examples* > *01.Basics* > *Blink*.

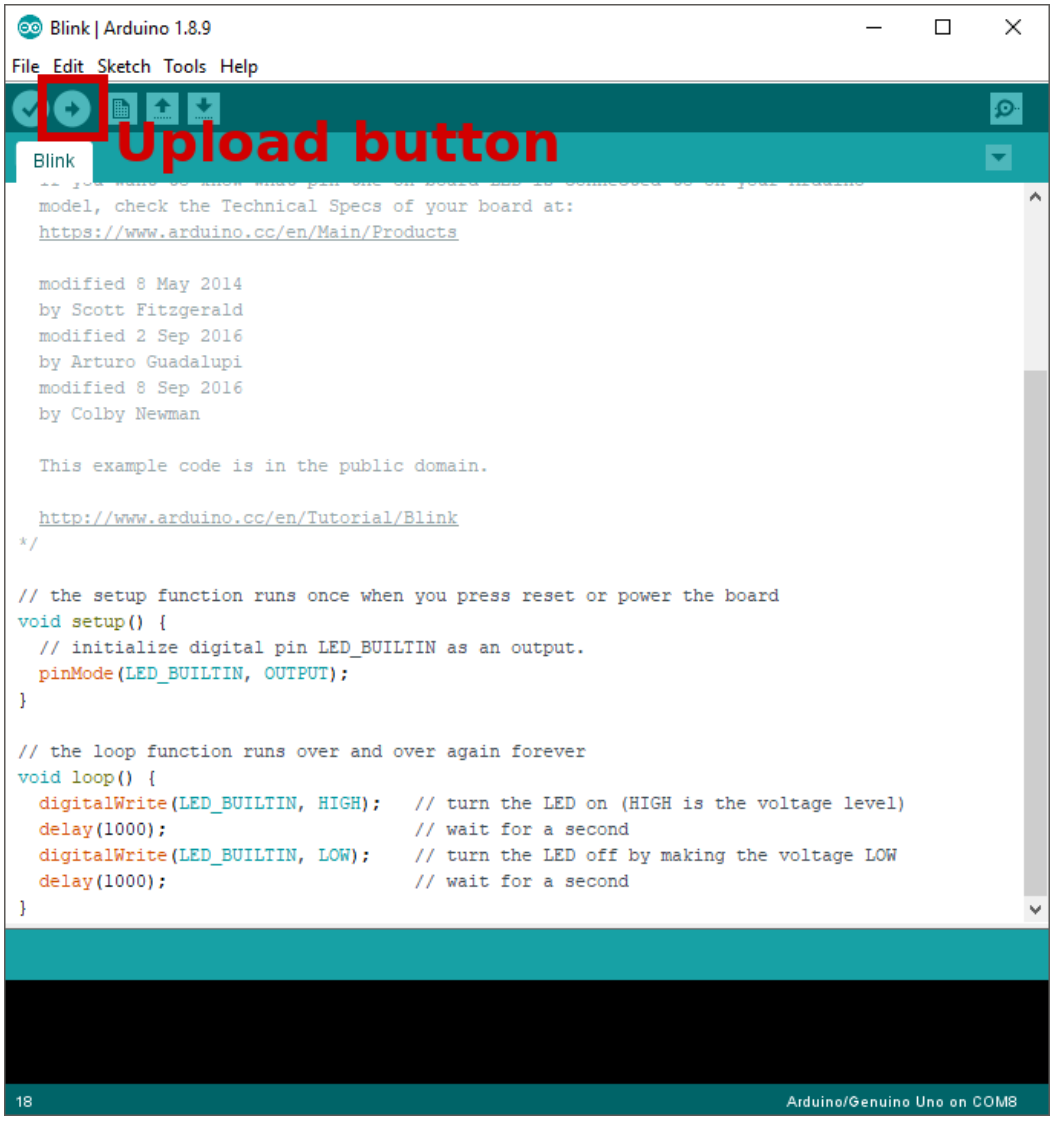


A new window with new sketch example will open:



What this sketch does is turn ON an on-board LED connected to the digital I/O pin 13 (D13), for one second, and than turn it OFF for one second. This turning on and off is called blinking, thus this sketch name.

When programing is done, compile the code and upload it to your Microcontroller Board by pressing the upload button.



After this, the onboard LED should start blinking in an interval of one second.

Now it is the time to learn and make your own projects. You can do that with the help of many example scripts and other tutorials, which can be found on the Internet.

If you are looking for the high quality microelectronics and accessories, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

https://az-delivery.de