

AZ-Delivery

Welcome!

Thank you for purchasing our *AZ-Delivery AZ-Micro Controller ATmega328P Board*. On the following pages, we will introduce you to how to use and set-up this handy device.

Have fun!



AZ-Delivery

Table of Contents

Introduction.....	3
Specifications.....	4
Pinout.....	5
AZ-Micro Controller.....	6
Communication interfaces.....	11
Power Pins.....	12
Power, Current and Voltage Limitations.....	13
Pins description.....	16
I/O pins.....	17
Serial communication pins.....	17
Analog pins.....	17
I2C pins.....	18
SPI pins.....	18
AREF pin.....	19
On-board LED.....	19
How to set-up Arduino IDE.....	20
Sketch examples.....	24
Blinking LED.....	24
PWM - Pulse Width Modulation.....	25

Az-Delivery

Introduction

The microcontroller board is great device intended for electronics learning or prototyping and learning programming. A microcontroller board is actually a microcontroller, but assembled in a way that you don't need extra components with it.

If single microcontroller is used, its needed to build stable DC power supply, external programmer, and reset circuit, and many other things. The ATmega328P board got all of this in one. And the most powerful thing about ATmega328P 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 to learn internal working of onboard microcontroller in order to program it. Just connect the microcontroller board via USB cable to the PC, install and start Arduino IDE, search and upload program to the board and that's it.

There are endless code and library examples already written and available online. Also there are numerous other microcontroller compatible boards, like shields, or many sensors built that are easily connected to the microcontroller board.

Az-Delivery

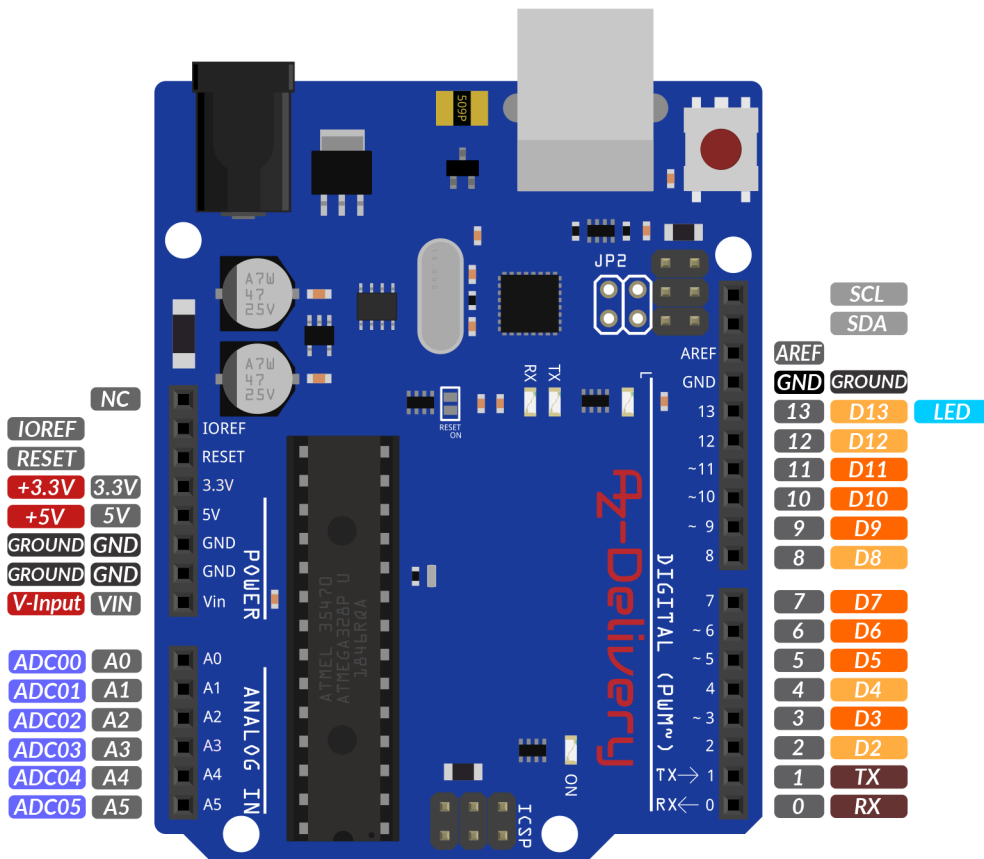
Specifications

Operating voltage	5V
Input voltage (recommended)	7-12V
DC input voltage (limit)	6-20V
Digital I/O pins	14 (6 PWM Outputs, 6 PWM Digital I/O)
Analog inputs	6
DC current per I/O	20mA
DC current for 3.3V pin	50mA
Flash memory	32KB (bootloader 0.5KB, SRAM 2KB)
EEPROM	1KB (ATMega328P)
Clock speed	16MHz
Built in LED	connected to I/O pin 13
Dimensions	69x54x15mm (2.7x2.1x0.5in)

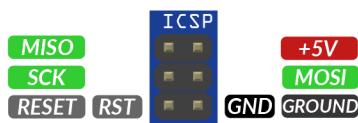
AZ-Delivery

Pinout

The AZ-Micro Controller has 44 pins. The pinout is shown on the following image:



- Digital pins
- Digital pins (supports PWM)
- Analog pins (Can be used as Digital pins)
- I2C bus (Wire)
- SPI bus
- Serial interfaces



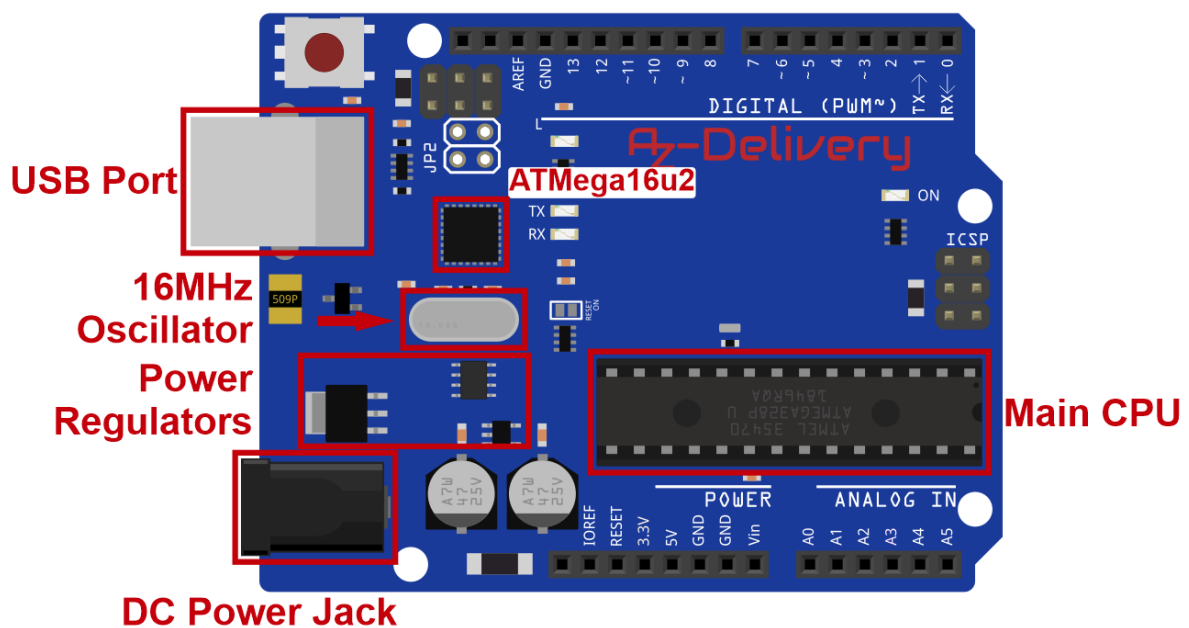
AZ-Delivery

AZ-Micro Controller

The AZ-Micro Controller is a microcontroller board based on the ATmega328 which has 20 digital input/output pins. 6 pins (D3, D5, D6, D9, D10, and D11) can be used as PWM outputs and 6 analog inputs, one serial communication line (D0, D1). Analog pins can also be used as digital pins.

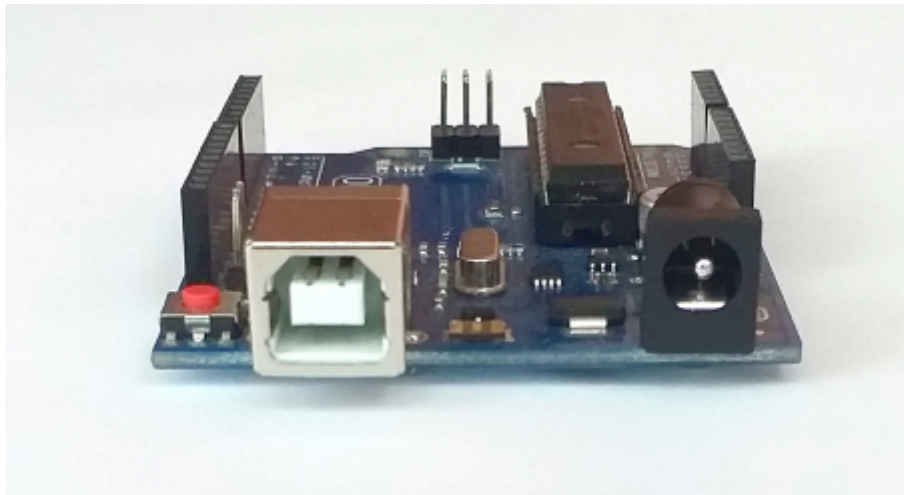
AZ-Micro Controller board features main microcontroller ATmega328P with 16MHz quartz oscillator. It uses another microcontroller ATmega16u2 to communicate with PC via USB. ATmega16u2 has built in USB communication interface, and USART serial interface, so it represents bridge between main microcontroller ATmega328 and your PC.

Also, the USB port is used to power the microcontroller board.



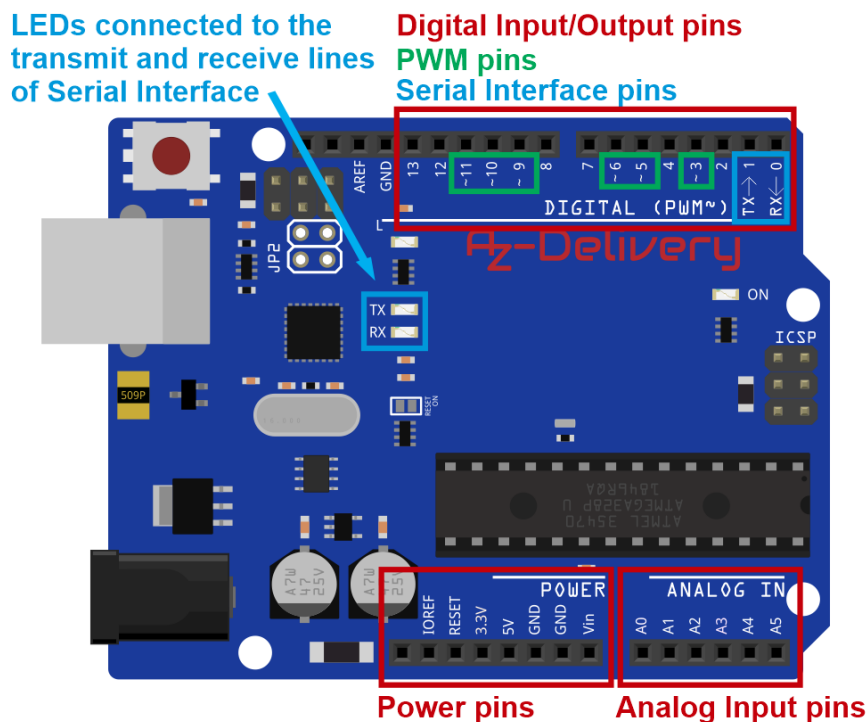
Az-Delivery

The board features DC voltage regulators for both, +5V and +3.3V. 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.



Az-Delivery

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

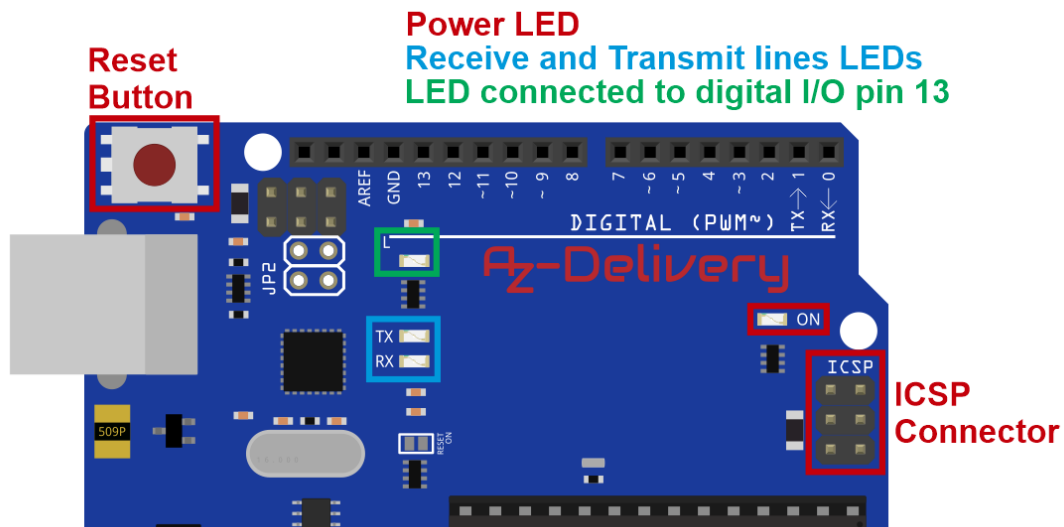


Digital input/output pins 0 and 1 are connected to receive and transmit lines of USART serial interface respectively. It is suggested never to use these digital I/O pins as digital inputs or outputs, because serial interface is used every time when new program is uploading to on-board microcontroller or the microcontroller sends data to the Serial Monitor of the Arduino IDE. Using these pins as digital inputs or outputs can get many errors while reprogramming, or when electronic parts or devices are connected to, they will work incorrectly.

Az-Delivery

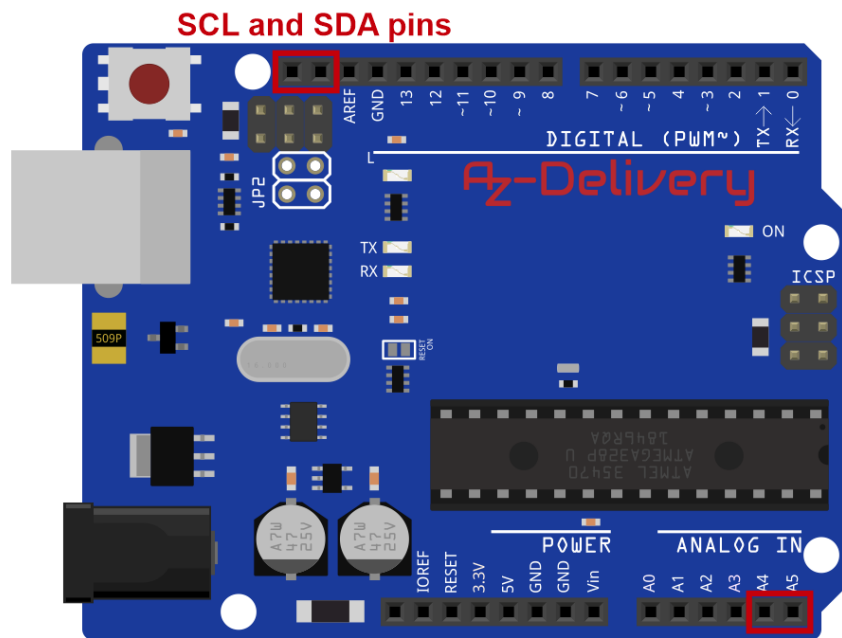
Onboard ATmega328P there is RESET button for main microcontroller, ICSP connector is used when ATmega328 is programmed externally via some other programmer.

One LED is turned on when microcontroller board is connected to the power. One LED is connected to the digital input/output 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

ATMega328P 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 (the I2C is also called: TWI - Two Wire Interface).



Alternative functions of A4 and A5 are SDA and SCL connections

Communicaton interfaces

The 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

VIN - This is voltage input pin and serves as another external power supply for main microcontroller when it is not powered by USB port.

GND - Ground

5V - +5V voltage output

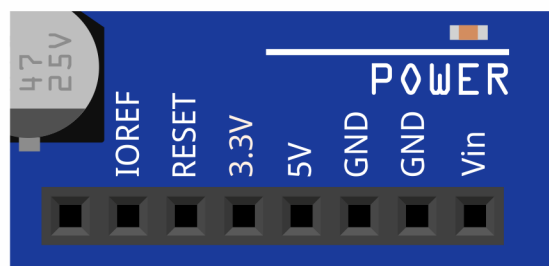
3.3V - +3.3V voltage output

RES - RESET, connecting this pin to GND (LOW) resets the main microcontroller.

IOREF - This pin serves 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 microcontroller board works (+5V for microcontroller).

Last pin is not connected to anything.

Following image show power pin header from right to left:





Power, Current and Voltage Limitations

Voltage limits:

Supplying the micro controller board with power can be done either through USB port, DC Power jack or through power pins header at VIN pin.

When powering the microcontroller board via DC Power jack or VIN and GND pins, it has the following input voltage limitations:

Recommended input voltage limits (power DC jack): 7-12V.

These input voltages can be sustained indefinitely.

Absolute voltage limits for powering the ATmega328P: 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-board voltage regulator, which could cause it to overheat. Short periods, however, are fine. Temperature can be checked by touching the voltage regulator with finger. If it feels too hot to comfortably touch, a voltage source within the recommended limits has to be used in order to reduce heat build-up.

Caution: Before touching any electro-static discharge (ESD) sensitive parts on the ATmega328P (which is pretty much all of the ATmega328P), touch the metal part of the USB plug first to ground yourself out to the board and safely discharge any static voltage that might have build-up.

Az-Delivery

Voltage limits on input/output pins: -0.5 to +5.5V max. Before reading a voltage on the microcontroller board or from digital or analog input, ensure it is between 0 and 5V. If it is outside these limits, bring down the voltage using a voltage divider. This scales the input voltage to allow for analog or digital readings of voltages otherwise outside the allowed range. If input signal is digital, and it is not needed to take scaled analog readings, another technique is to clip (cut the top off) 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 10k Ω 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 the board when powered from a USB port is: 500mA.

The microcontroller board has a “resettable polyfuse that protects 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 microcontroller board is limited by the on-board voltage regulator, and/or input power supply, whichever provides less power. Assuming the power supply is going to the microcontroller board can provide 7-12V and $\geq 1A$. If this is the case, the 5V power is limited strictly by on-board voltage regulator.

Az-Delivery

Total max current draw across the microcontroller board 5V pin and GND: as specified above.

Total max current per input/output pin: 40mA.

Sum of currents out of ALL input/output pins combined is: 200mA!!!

NOTE: This is the one that usually gets people, as it may be the least understood! Despite the fact that voltage regulator on-board the ATmega328P 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 **cannot exceed 200mA**. So, if powering 10 LEDs at 20mA each, via Analog or Digital pins, the limit is hit! Any more than that and the damage may occur to microcontroller. A work around if more current is needed, is to use transistors. The microcontroller board input/output pins can then use a very low current to activate a transistor, which then turns a higher current on and off from the 5V pin directly (which is connected straight to the output of the on-board linear voltage regulator), to the device you want to control. This way, the sum of total output from analog/digital pins is kept below 200mA, while allowing up to the 500mA-1A limit from the 5V pin.



Pins description

The 5V pin is used to provide regulated output voltage of 5VDC. It supplies the controller and other components on the board. It can be obtained from VIN pin, USB cable or another regulated 5V voltage supply. Another voltage regulation is provided by 3.3V pin. Maximum current that can be drawn is 50mA.

The VIN is the input voltage supplied to the board which ranges from 7V to 20V. The voltage provided by the power jack can be accessed through this pin. However, the output voltage through this pin to the board will be automatically set up to 5V.

There are 5 ground pins available on the board which makes it useful when more ground pins are required for a project.

The Reset pin is used to reset the board. Setting this pin to LOW will reset the board.

Az-Delivery

I/O pins

Serial communication pins

Serial pins RXD and TXD are used to transmit and receive serial data. The Rx represents the transmission of data while Tx is used to receive data.

There are four UART interfaces (Universal Asynchronous Receiver Transmitter) where Serial 0 contains RX(0) and TX(1).

A *SoftwareSerial* library allows for serial communication on any of the ATmega328P digital pins.

Analog pins

There are 6 analog pins incorporated on the board labeled as A0 to A5. It is important to note that all these analog pins can also be used as digital I/O pins. Each analog pin comes with 10bit resolution. These pins can measure from 0-5V. However, the upper value can be changed using AREF and *analogReference()* function from the Arduino IDE.

Az-Delivery

I2C pins

Two pins A4 and A5 support I2C communication where A4 represents SDA (Serial Data Line) mainly used for holding the data and A5 represents SCL (Serial Clock Line) mainly used for providing data synchronization between the devices. Additional two pins are connected to these pins.

SPI pins

The SPI stands for Serial Peripheral Interface used for the transmission of data between the controller and other peripheral components. Four pins 12(MISO), 11(MOSI), 13(SCK), 10(SS) are used for SPI communication.

The SPI is also available through two ICSP headers.

AREF pin

The AREF stands for Analog Reference Voltage which is a reference voltage for analog inputs. The function used for AREF in Arduino IDE is *analogReference()*.

On-board LED

This board comes with built-in LED connected to digital pin 13. HIGH value at this pin will turn the LED on and LOW value will turn it off. Simple *Blink* and *Fade* examples from the Arduino IDE can be used for testing the ATmega328P for the first time.

Az-Delivery

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. The Arduino IDE version used for this eBook is **1.8.15**.



 **Arduino IDE 1.8.15**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

DOWNLOAD OPTIONS

- Windows** Win 7 and newer
- Windows** ZIP file
- Windows app** Win 8.1 or 10 
- Linux** 32 bits
- Linux** 64 bits
- Linux** ARM 32 bits
- Linux** ARM 64 bits
- Mac OS X** 10.10 or newer

[Release Notes](#) [Checksums \(sha512\)](#)

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

Az-Delivery

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`.

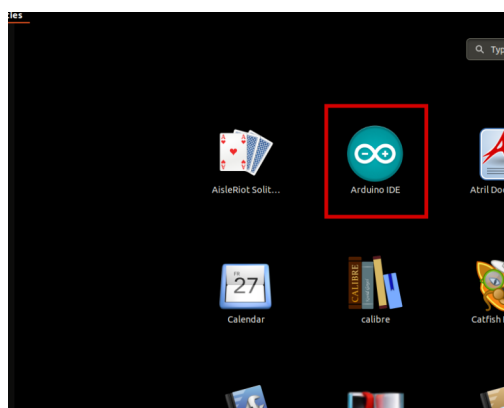
To run the first script in the terminal, open the terminal in the extracted directory and run the following command:

```
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.



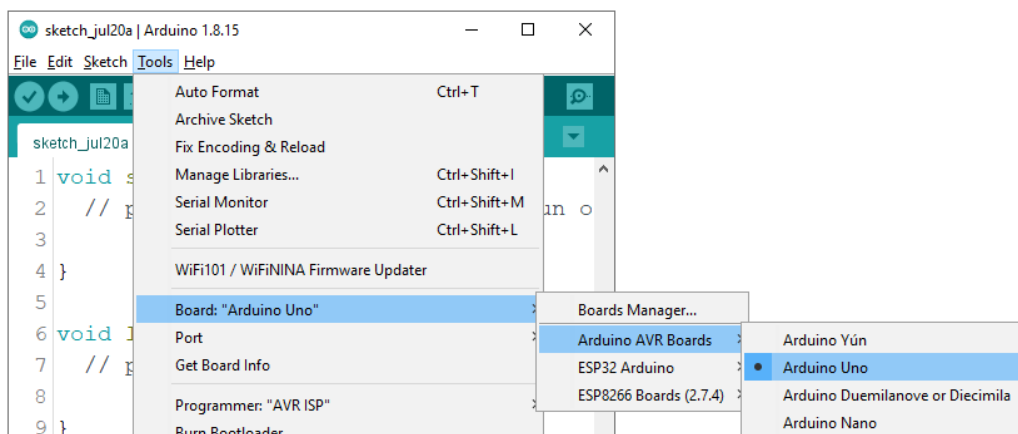
Az-Delivery

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 the PC can detect an microcontroller. 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:



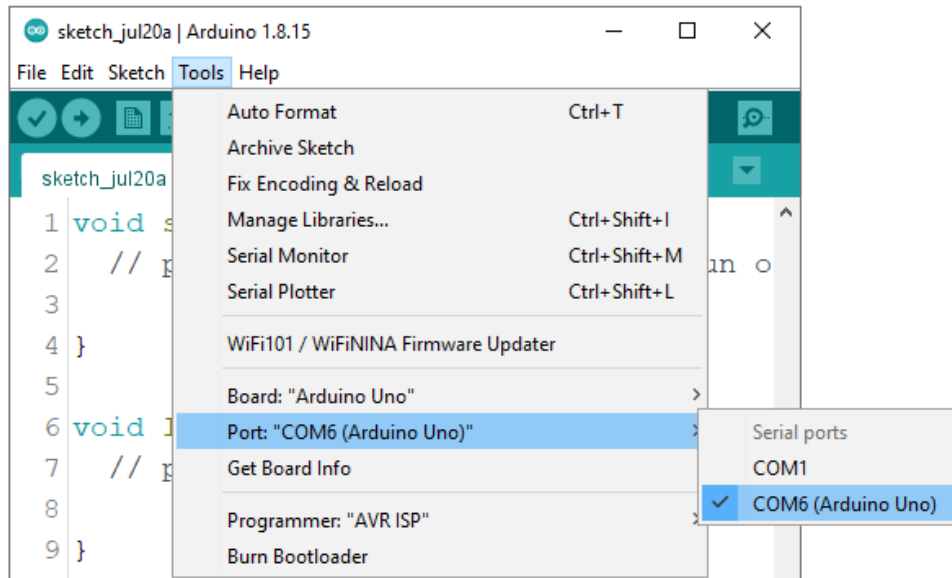
The port to which the microcontroller board is connected has to be selected.

Go to: *Tools > Port > {port name goes here}*

and when the microcontroller board is connected to the USB port, the port name can be seen in the drop-down menu on the previous image.

Az-Delivery

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 integer number between 0 and 9.

Az-Delivery

Sketch examples

Blinking LED

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```


Az-Delivery

PWM - Pulse Width Modulation

```
int led = 13; // Built-in LED is connected to pin 13
int brightness = 0;
int fadeAmount = 5;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  delay(30);
}
```

AZ-Delivery

Now is the time to learn and make the Projects on your own. 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>

Have Fun!

Impressum

<https://az-delivery.de/pages/about-us>