Ex. No: **Introduction to Arduino platform and programming**
Date:

AIM:

To study the basics of Arduino Uno board and Arduino IDE 2.0 software.

Hardware & SOFTWARE TOOLS REQUIRED:

S.No.	Hardware & Software Requirements	Quantity
1	Arduino IDE 2.0	1
2	Arduino Uno Board	1

INTRODUCTION TO ARDUINO:

Arduino is a project, open-source hardware, and software platform used to design and build electronic devices. It designs and manufactures microcontroller kits and single-board interfaces for building electronics projects. The Arduino boards re initially created to help students with the non-technical background. The designs of Arduino boards use a variety of controllers and microprocessors. Arduino is an easy-to-use open platform for creating electronic projects. Arduino boards play a vital role in creating different projects. It makes electronics accessible to non-engineers, hobbyists, etc.

The various components present on the Arduino boards are a Microcontroller, Digital Input/output pins, USB Interface and Connector, Analog Pins, reset buttons, Po r buttons, LEDs, Crystal oscillators, and Voltage regulators. Some components may differ depending on the type of board. The most standard and popular board used over time is Arduino UNO. The ATmega328 Microcontroller present on the UNO board makes it rather po rful than other boards. There are various types of Arduino boards used for different purposes and projects. The Arduino Boards are organized using the Arduino (IDE), which can run on various platforms. Here, IDE stands for Integrated Development Environment. Let's discuss some common and best Arduino boards.

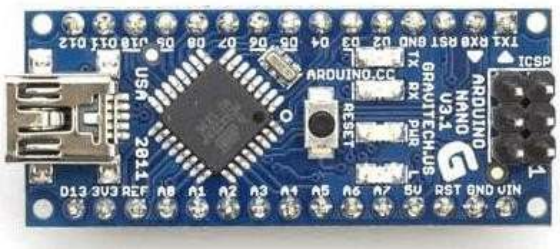
TYPES OF ARDUINO BOARDS

1) Arduino UNO

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a po r jack, and an ICSP (In-Circuit Serial Programming) header. It is the most used and of standard form from the list of all available Arduino Boards.

2) Arduino Nano

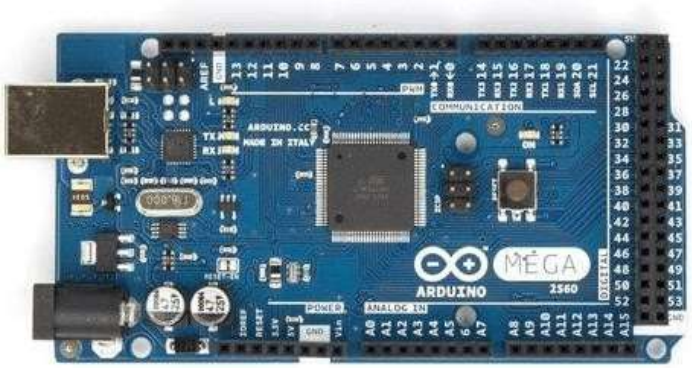
The Arduino Nano is a small Arduino board based on ATmega328P or ATmega628 Microcontroller. The connectivity is the same as the Arduino UNO board. The Nano board is defined as a sustainable, small, consistent, and flexible microcontroller board. It is small in size compared to the UNO board. The devices required to start our projects using the Arduino Nano board are Arduino IDE and mini-USB. The Arduino Nano includes an I/O pin set of 14 digital pins and 8 analog pins. It also includes 6 Po r pins and 2 Reset pins.



3) Arduino Mega

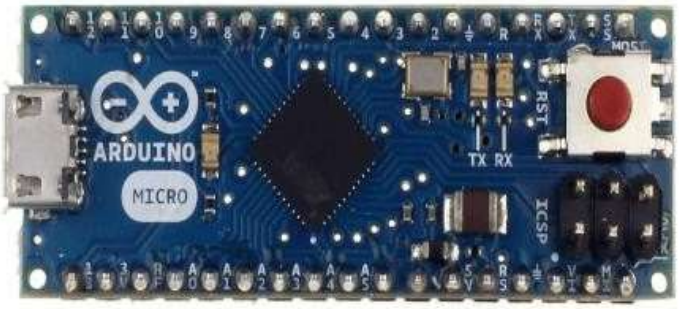
The Arduino Mega is based on the ATmega2560 Microcontroller. The ATmega2560 is an 8-bit microcontroller. need a simple USB cable to connect to the computer and the AC to DC adapter or battery to get started with it. It has the advantage of working with more memory space. The Arduino Mega includes 54 I/O digital pins and 16 Analog Input/Output (I/O), ICSP header, a reset

button, 4 UART (Universal Asynchronous Reciever/Transmitter) ports, USB connection, and a po r jack.



4) Arduino Micro

The Arduino Micro is based on the ATmega32U4 Microcontroller. It consists of 20 sets of pins. The 7 pins from the set are PWM (Pulse Width Modulation) pins, while 12 pins are analog input pins. The other components on board are a reset button, a 16MHz crystal oscillator, an ICSP header, and a micro-USB connection. The USB is built in the Arduino Micro board.



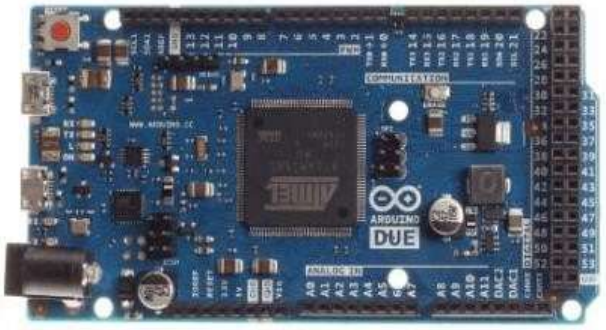
5) Arduino Leonardo

The basic specification of the Arduino Leonardo is the same as the Arduino Micro. It is also based on the ATmega32U4 Microcontroller. The components present on the board are 20 analog and digital pins, a reset button, a 16MHz crystal oscillator, an ICSP header, and a micro USB connection.

6) Arduino Due

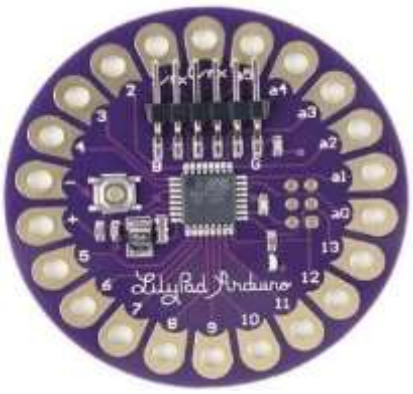
The Arduino Due is based on the 32-bit ARM core. It is the first Arduino board that has been

developed based on the ARM Microcontroller. It consists of 54 Digital Input/Output pins and 12 Analog pins. The Microcontroller present on the board is the Atmel SAM3X8E ARM Cortex-M3 CPU. It has two ports, namely, a native USB port and a Programming port. The micro side of the USB cable should be attached to the programming port.



7) Arduino Lilypad

The Arduino LilyPad was initially created for arable projects and e-textiles. It is based on the ATmega168 Microcontroller. The functionality of Lilypad is the same as other Arduino Boards. It is a round, light ight board with a minimal number of components to keep the size of the board small. The Arduino Lilypad board was designed by Sparkfun and Leah. It was developed by Leah Buechley. It has 9 digital I/O pins.



8) Arduino Bluetooth

The Arduino Bluetooth board is based on ATmega168 Microcontroller. It is also named as **Arduino BT board**. The components present on the board are 16 digital pins, 6 analog pins, reset button, 16MHz crystal oscillator, ICSP header, and screw terminals. The screw terminals are used for po r. The Arduino Bluetooth Microcontroller board can be programmed over the Bluetooth as a wireless connection.

9) Arduino Diecimila

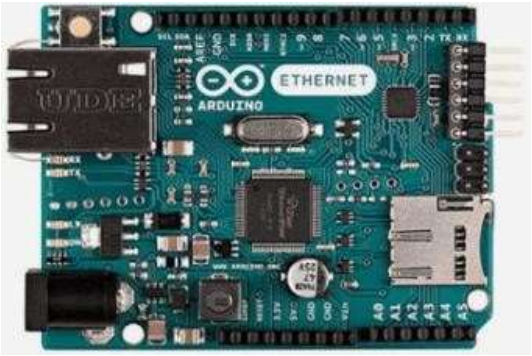
The Arduino Diecimila is also based on the ATmeg628 Microcontroller. The board consists of 6 analog pin inputs, 14 digital Input/Output pins, a USB connector, a po r jack, an ICSP (In-Circuit Serial Programming) header, and a reset button. can connect the board to the computer using the USB and can po r on the board with the help of an AC to DC adapter. The Diecimila was initially developed to mark the 10000 delivered boards of Arduino. Here, Diecimila means 10,000 in Italian.

10) Arduino Robot

The Arduino Robot is called as the tiny computer. It is widely used in robotics. The board comprises of the speaker, five-button, color screen, two motors, an SD card reader, a digital compass, two potentiometers, and five floor sensors. The Robot Library can be used to control the actuators and the sensors.

11) Arduino Ethernet

The Arduino Ethernet is based on the ATmega328 Microcontroller. The board consists of 6 analog pins, 14 digital I/O pins, crystal oscillator, reset button, ICSP header, a po r jack, and an RJ45 connection. With the help of the Ethernet shield, can connect our Arduino board to the internet.



12) Arduino Zero

The Arduino Zero is generally called as the 32-bit extension of the Arduino UNO. It is based on ATmel's SAM21 MCU. The board consists of 6 analog pin inputs, 14 digital Input/Output pins, a USB connector, a po r jack, and an ICSP (In-Circuit Serial Programming) header, UART port pins, a po r header, and AREF button. The Embedded debugger of Atmel is also supported by the Arduino Zero. The function of Debugger is to provide a full debug interface, which does not require additional hardware.


RESET
SCL
SDA
AREF
GND
13
12
11
10
9
8
7
6
5
4
3
2
TX→1
RX←0
DIGITAL (PWM~)
JTAG
ZERO
ARDUINO
RX
TX
L
ON
DEBUG
SPI
SWD
ARDUINO.CC
MADE IN ITALY
IOREF
RESET
3V3
5V
GND
VIN
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

13) Arduino Esplora

The Arduino Esplora boards allow easy interfacing of sensors and actuators. The outputs and inputs connected on the Esplora board make it unique from other types of Arduino boards. The board includes outputs, inputs, a small microcontroller, a microphone, a sensor, a joystick, an accelerometer, a temperature sensor, four buttons, and a slider.

14) Arduino Pro Micro

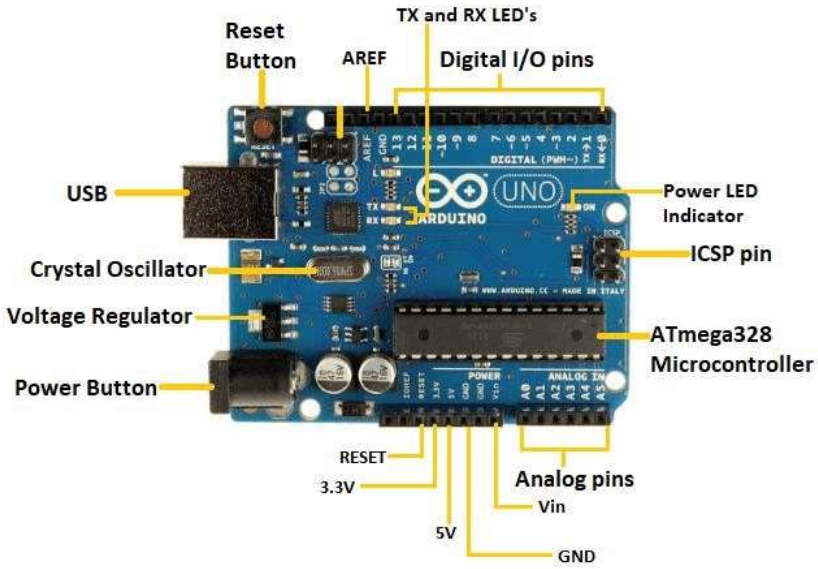
The structure of Arduino Pro Micro is similar to the Arduino Mini board, except the Microcontroller ATmega32U4. The board consists of 12 digital Input/output pins, 5 PWM (Pulse Width Modulation) pins, Tx and Rx serial connections, and 10-bit ADC (Analog to Digital Converter).

<u>Introduction To Arduino Uno:</u>

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered a po rful board used in various projects. Arduino. cc developed the Arduino UNO board. Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits. The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a <u>USB</u> connector, a po r jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms. The IDE is common to all available boards of Arduino.

The Arduino board is shown below:

The components of Arduino UNO board are shown below:



- **ATmega328 Microcontroller**- It is a single-chip Microcontroller of the ATmel family. The processor code inside it is of 8-bit. It combines **Memory (SRAM, EEPROM, and Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timers, external and internal interrupts, and oscillator.**
- **ICSP pin** - The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.
- **Po r LED Indicator**- The ON status of the LED shows the po r is activated. When the po r is OFF, the LED will not light up.
- **Digital I/O pins**- The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pins.
- **TX and RX LED's**- The successful flow of data is represented by the lighting of these LED's.
- **AREF**- The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external po r supply.

- **Reset button**- It is used to add a Reset button to the connection.
- **USB**- It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.
- **Crystal Oscillator**- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a po rful board.
- **Voltage Regulator**- The voltage regulator converts the input voltage to 5V.
- **GND**- Ground pins. The ground pin acts as a pin with zero voltage.
- **Vin**- It is the input voltage.
- **Analog Pins**- The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) pin.

Technical Specifications Of Arduino Uno

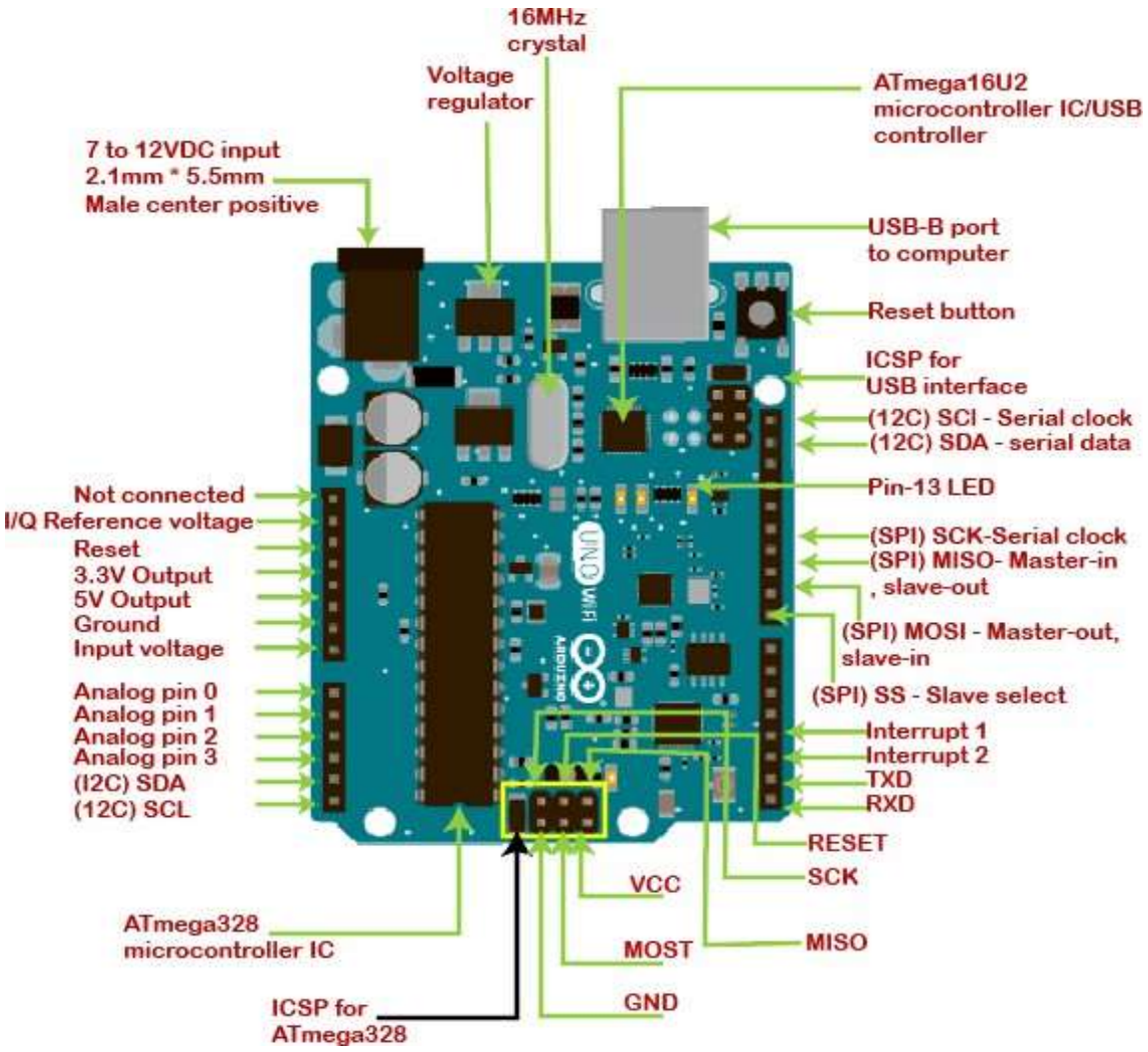
The technical specifications of the Arduino UNO are listed below:

- There are 20 Input/Output pins present on the Arduino UNO board. These 20 pins include 6 PWM pins, 6 analog pins, and 8 digital I/O pins.
- The PWM pins are Pulse Width Modulation capable.
- The crystal oscillator present in Arduino UNO comes with a frequency of 16MHz.
- It also has an Arduino-integrated WIFI module. Such Arduino UNO board is based on the Integrated WIFI ESP8266 Module and ATmega328P microcontroller.
- The input voltage of the UNO board varies from 7V to 20V.
- Arduino UNO automatically draws po r from the external po r supply. It can also draw po r from the USB.

Arduino Uno Pinout

The Arduino UNO is a standard board of Arduino, which is based on an **ATmega328P** microcontroller. It is easier to use than other types of Arduino Boards.

The Arduino UNO Board, with the specification of pins, is shown below:



Pin Description

ATmega328 Microcontroller- It is a single chip Microcontroller of the ATmel family. The processor core inside it is of 8-bit. It is a low-cost, low po red, and a simple microcontroller. The Arduino UNO and Nano models are based on the ATmega328 Microcontroller.

Voltage Regulator: The voltage regulator converts the input voltage to 5V. The primary function of voltage regulator is to regulate the voltage level in the Arduino board. For any changes in the input voltage of the regulator, the output voltage is constant and steady.

GND - Ground pins. The ground pins are used to ground the circuit.

TXD and RXD: TXD and RXD pins are used for serial communication. The TXD is used for transmitting the data, and RXD is used for receiving the data. It also represents the successful flow of data.

USB Interface: The USB Interface is used to plug-in the USB cable. It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.

RESET: It is used to add a Reset button to the connection.

SCK: It stands for **Serial Clock**. These are the clock pulses, which are used to synchronize the transmission of data.

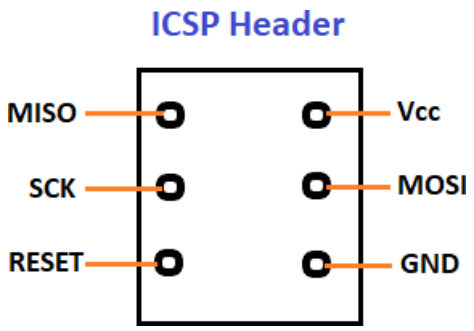
MISO: It stands for **Master Input/ Slave Output**. The save line in the MISO pin is used to send the data to the master.

VCC: It is the modulated DC supply voltage, which is used to regulate the IC's used in the connection. It is also called as the primary voltage for IC's present on the Arduino board. The Vcc voltage value can be negative or positive with respect to the GND pin.

Crystal Oscillator- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a po rful board.

ICSP: It stands for **In-Circuit Serial Programming**. The users can program the Arduino board's firmware using the ICSP pins. The program or firmware with the advanced functionalities is received by microcontroller with the help of the ICSP header. The ICSP header consists of 6 pins.

The structure of the ICSP header is shown below:



SDA: It stands for **Serial Data**. It is a line used by the slave and master to send and receive data. It is called as a **data line,** while SCL is called as a clock line.

SCL: It stands for **Serial Clock**. It is defined as the line that carries the clock data. It is used to synchronize the transfer of data bet en the two devices. The Serial Clock is generated by the device and it is called as master.

SPI: It stands for **Serial Peripheral Interface**. It is popularly used by the microcontrollers to communicate with one or more peripheral devices quickly. It uses conductors for data receiving, data sending, synchronization, and device selection (for communication).

MOSI: It stands for Master Output/ Slave Input. The MOSI and SCK are driven by the Master.

SS: It stands for **Slave Select**. It is the Slave Select line, which is used by the master. It acts as the

enable line. **I^2C:** It is the two-wire serial communication protocol. It stands for Inter Integrated Circuits. The I^2C is a serial communication protocol that uses SCL (Serial Clock) and SDA (Serial Data) to receive and send data bet en two devices. **3.3V and 5V** are the operating voltages of the board.

Introduction To Arduino Ide 2.0:

The Arduino IDE 2.0 is an open-source project, currently in its beta-phase. It is a big step from it's sturdy predecessor, Arduino IDE 2.0, and comes with revamped UI, improved board & library manger, autocomplete feature and much more.

In this tutorial, will go through step by step, how to download and install the software.

Download the editor

Downloading the Arduino IDE 2.0 is done through the Arduino Software page. Here will also find information on the other editors available to use.

Requirements

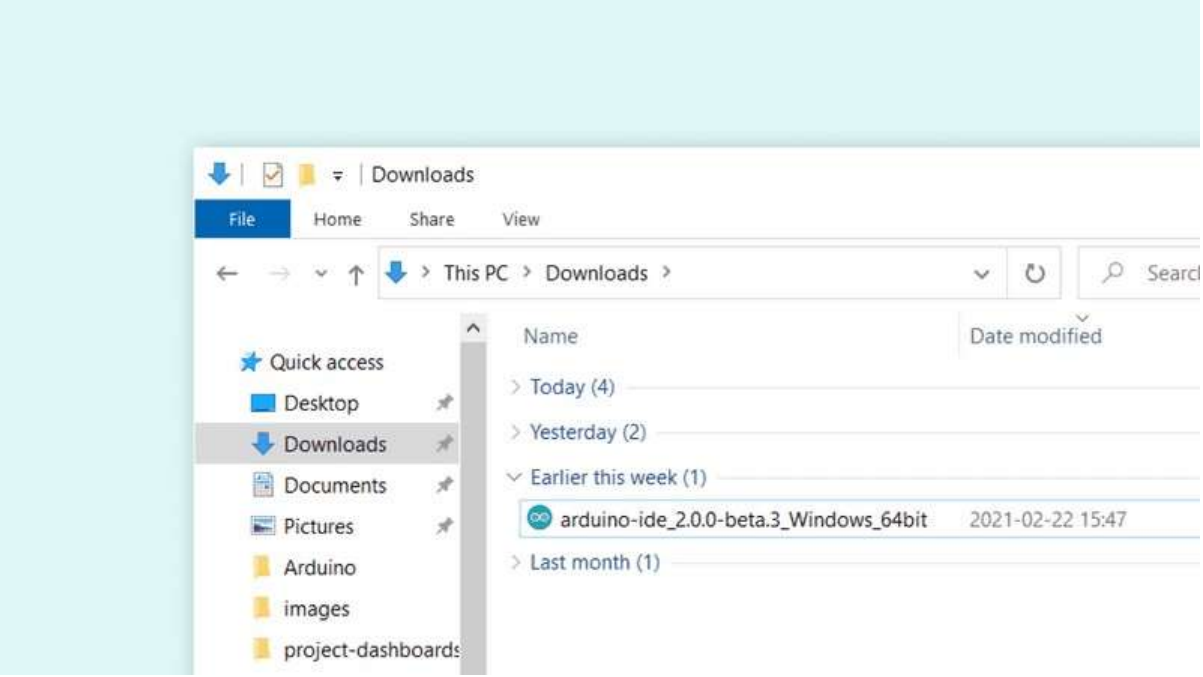
- **Windows** - Win 10 and ne r, 64 bits
- **Linux** - 64 bits
- **Mac OS X** - Version 10.14: "Mojave" or ne r, 64 bits

Installation

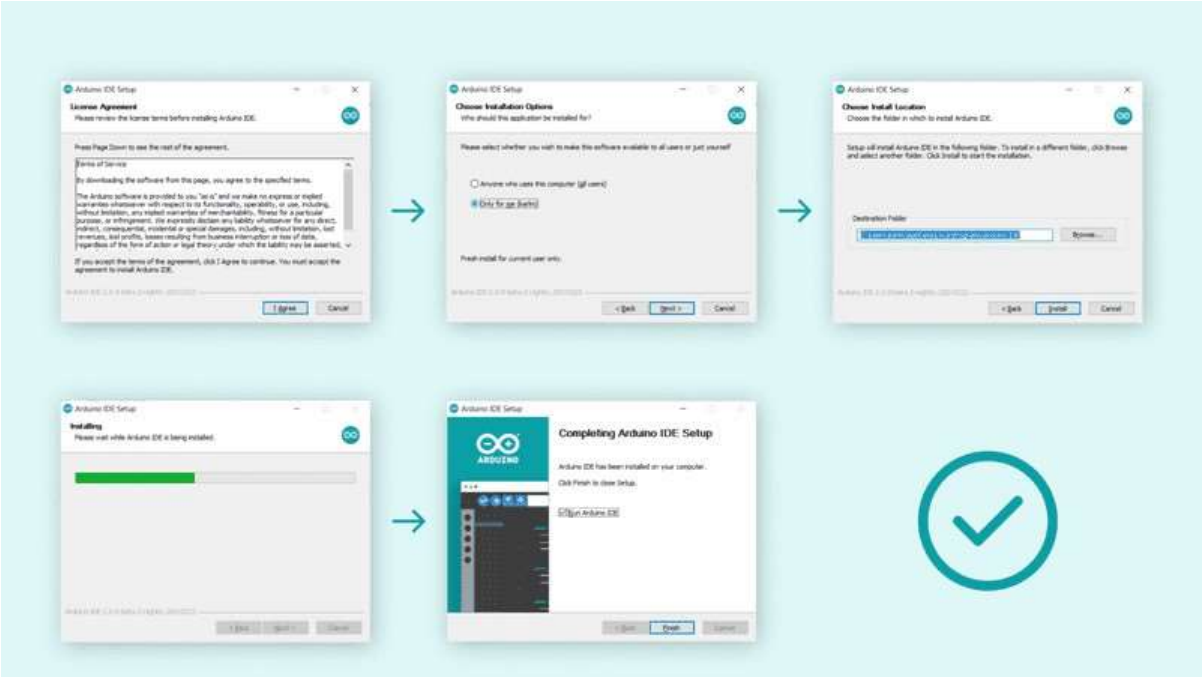
Windows

Download URL: https://www.arduino.cc/en/software

To install the Arduino IDE 2.0 on a Windows computer, simply run the file downloaded from the software page.



Follow the instructions in the installation guide. The installation may take several minutes.



How to use the board manager with the Arduino IDE 2.0

The board manager is a great tool for installing the necessary cores to use the Arduino boards. In this quick tutorial, will take a look at how to install one, and choosing the right core for the board!

Requirements

- Arduino IDE 2.0 installed.

Why use the board manager?

The board manager is a tool that is used to install different cores on the local computer. A core is written and designed for specific microcontrollers. As Arduino have several different types of

boards, they also have different type of microcontrollers.

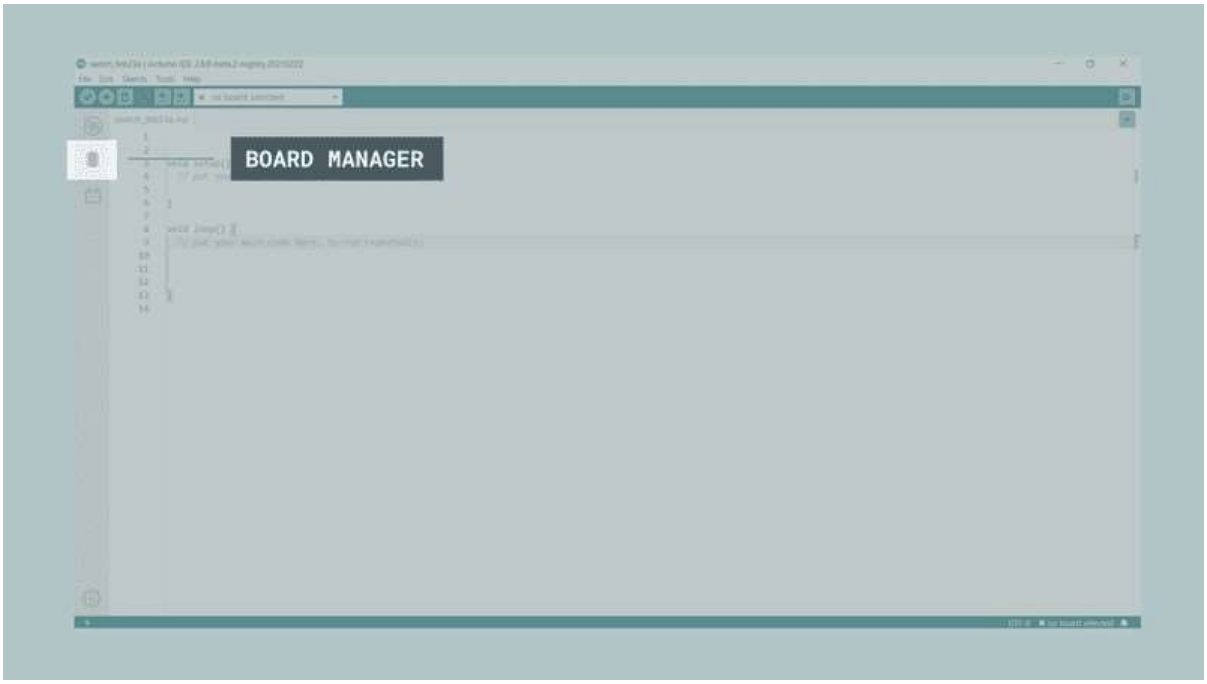
For example, an Arduino UNO has an **ATmega328P**, which uses the **AVR core**, while an Arduino Nano 33 IoT has a **SAMD21** microcontroller, where need to use the **SAMD core**.

In conclusion, to use a specific board, need to install a specific core.

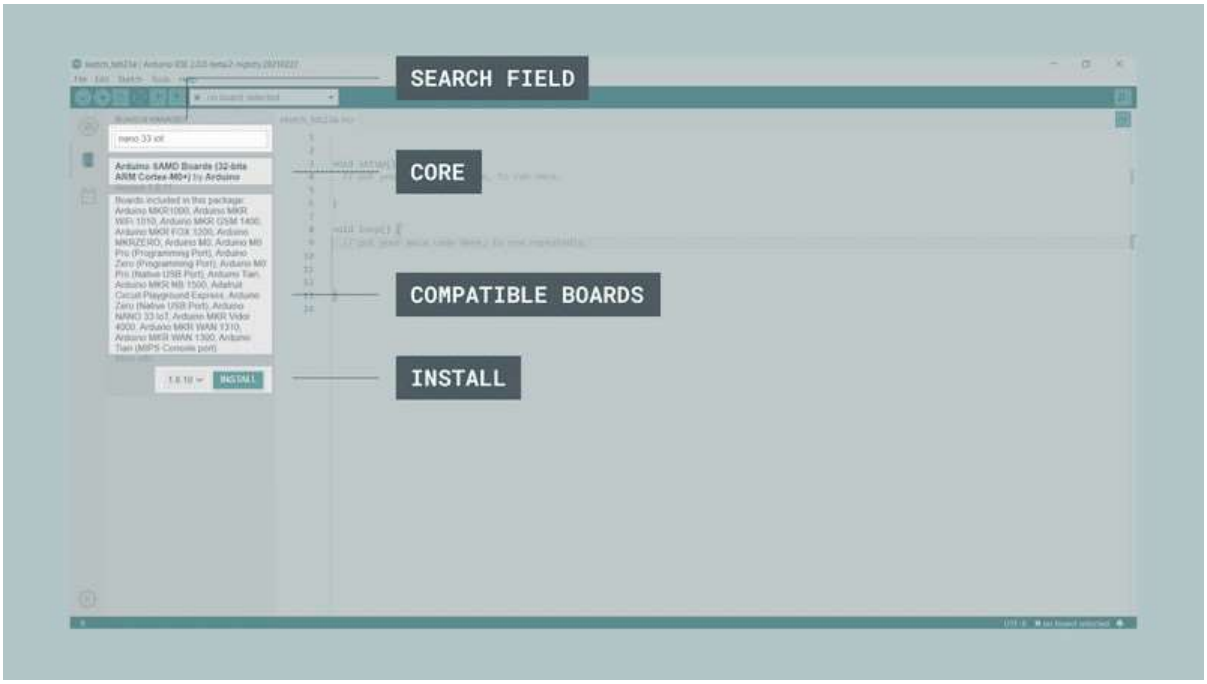
Installing a core

Installing a core is quick and easy, but let's take a look at what need to do.

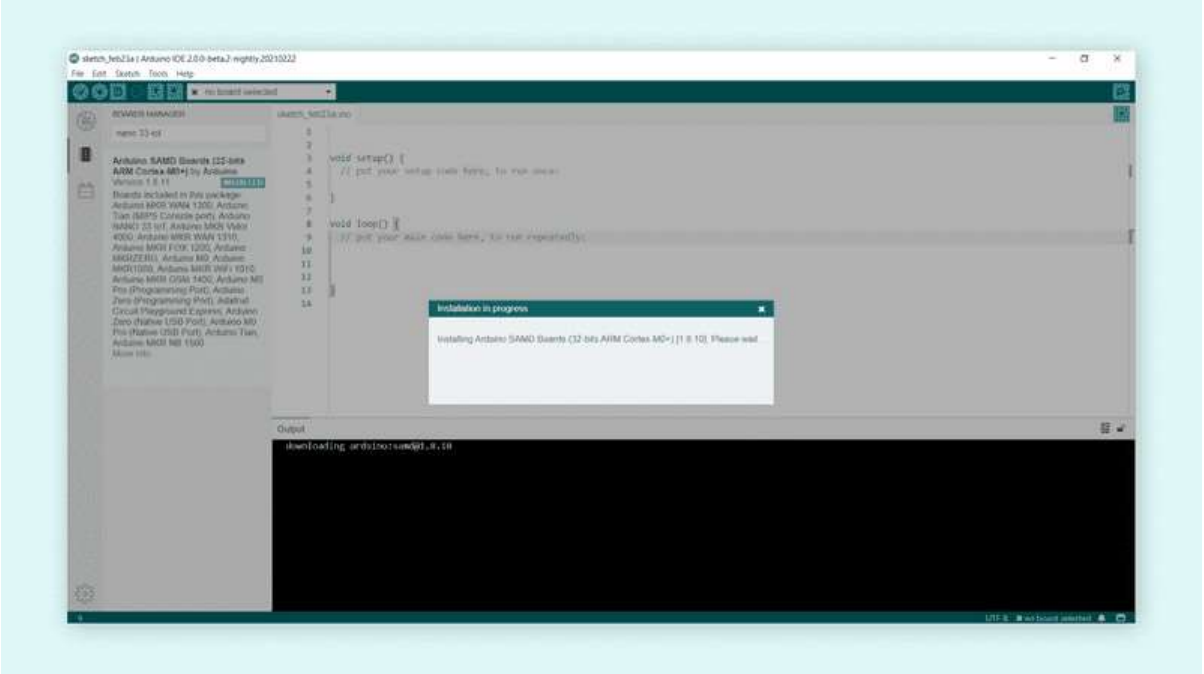
1. Open the Arduino IDE 2.0.
2. With the editor open, let's take a look at the left column. Here, can see a couple of icons. Let's click the on the **"computer chip"** icon.



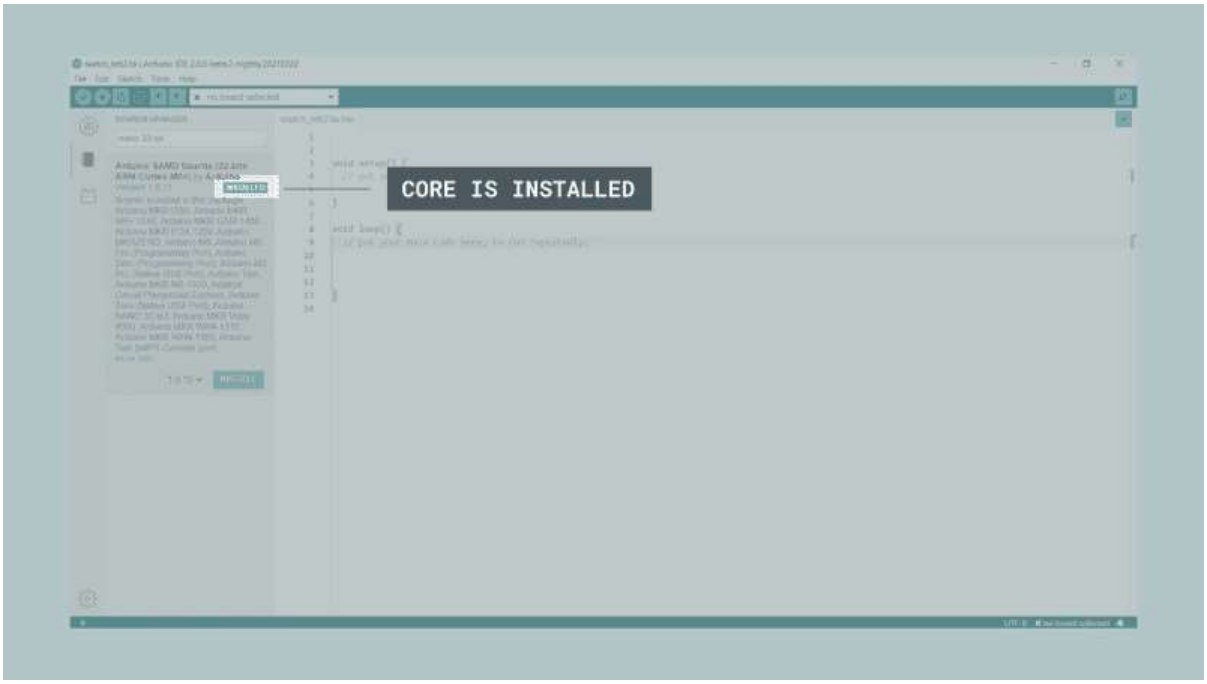
1. A list will now appear of all available cores. Now let's say are using an **Nano 33 IoT** board, and want to install the core. Simply enter the name in the search field, and the right core (SAMD) will appear, where the Nano 33 IoT features in the description. Click on the **"INSTALL"** button.



4. This will begin an installation process, which in some cases may take several minutes.



5. When it is finished, can take a look at the core in the boards manager column, where it should say **"INSTALLED"**.



How to upload a sketch with the Arduino IDE 2.0

In the Arduino environment, write **sketches** that can be uploaded to Arduino boards. In this tutorial, will go through how to select a board connected to the computer, and how to upload a sketch to that board, using the Arduino IDE 2.0.

Requirements

- Arduino IDE 2.0 installed.

Verify VS Upload

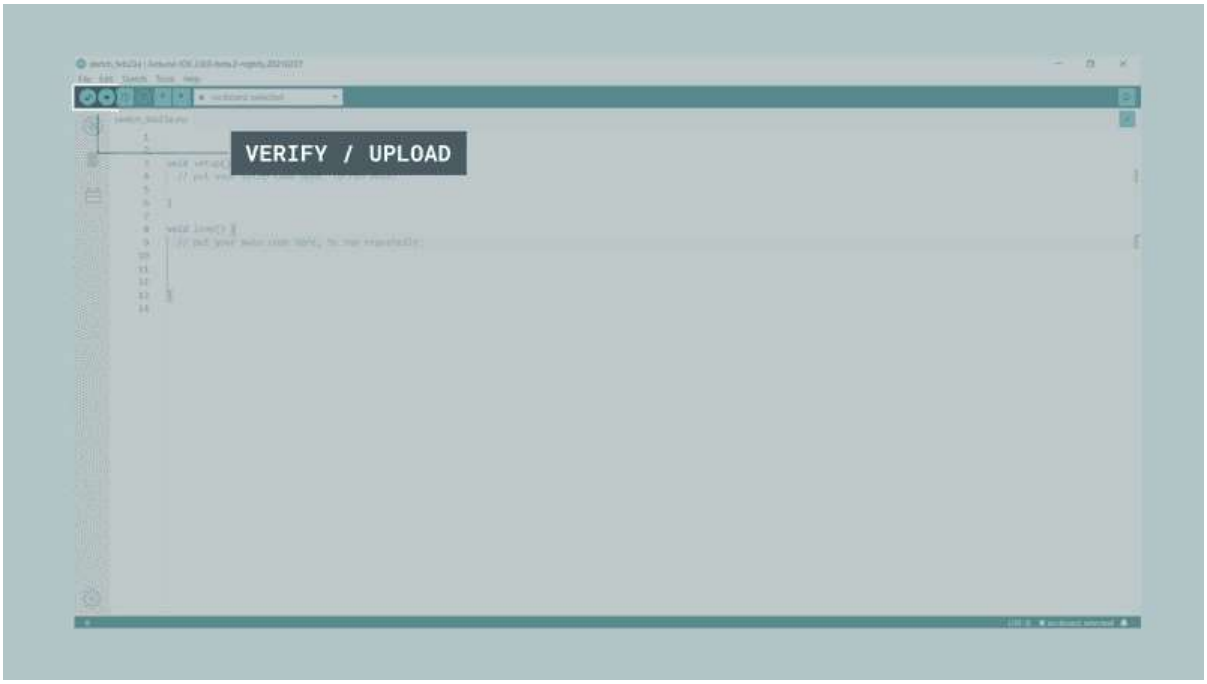
There are two main tools when uploading a sketch to a board: **verify** and **upload.** The verify tool simply goes through the sketch, checks for errors and compiles it. The upload tool does the same, but when it finishes compiling the code, it also uploads it to the board.

A good practice is to use the verifying tool before attempting to upload anything. This is a quick way of spotting any errors in the code, so can fix them before actually uploading the code.

Uploading a sketch

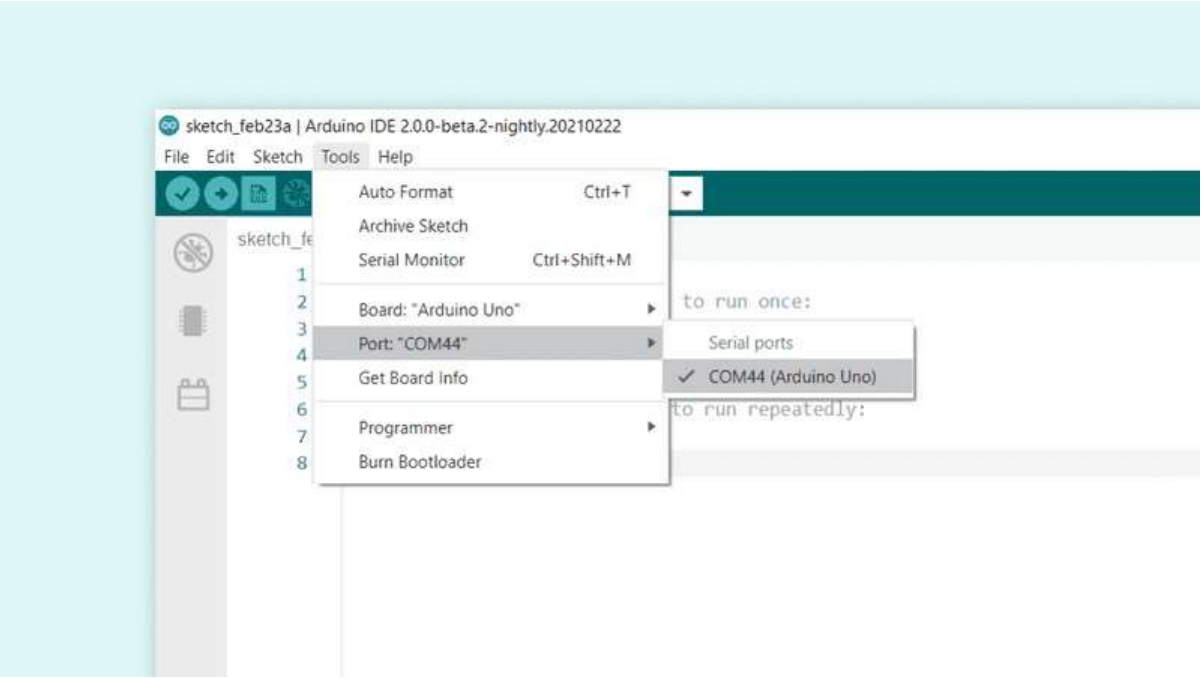
Installing a core is quick and easy, but let's take a look at what need to do.

1. Open the Arduino IDE 2.0.
2. With the editor open, let's take a look at the navigation bar at the top. At the very left, there is a **checkmark** and an **arrow pointing right**. The checkmark is used to **verify**, and the arrow is used to **upload.**



3. Click on the verify tool (checkmark). Since are verifying an empty sketch, can be sure it is going to compile. After a few seconds, can see the result of the action in the console (black box in the bottom).

1. Now know that our code is compiled, and that it is working. Now, before can upload the code to our board, will first need to select the board that are using. can do this by navigating to **Tools > Port > {Board}**. The board(s) that are connected to the computer should appear here, and need to select it by clicking it. In this case, our board is displayed as **COM44 (Arduino UNO)**.



5. With the board selected, are good to go! Click on the **upload** button, and it will start uploading the sketch to the board.
6. When it is finished, it will notify in the console log. Of course, sometimes there are some complications when uploading, and these errors will be listed here as ll.

have now uploaded a sketch to the Arduino board!

How to install and use a library with the Arduino IDE 2.0

A large part of the Arduino programming experience is the **use of libraries.** Thousands of libraries can be found online, and the best-documented ones can be found and installed directly through the editor.

Requirements

- Arduino IDE 2.0 installed.

Why use libraries?

Libraries are incredibly useful when creating a project of any type. They make our development experience much smoother, and there almost an infinite amount out there. They are used to interface with many different sensors, RTCs, Wi-Fi modules, RGB matrices and of course with other components on the board.

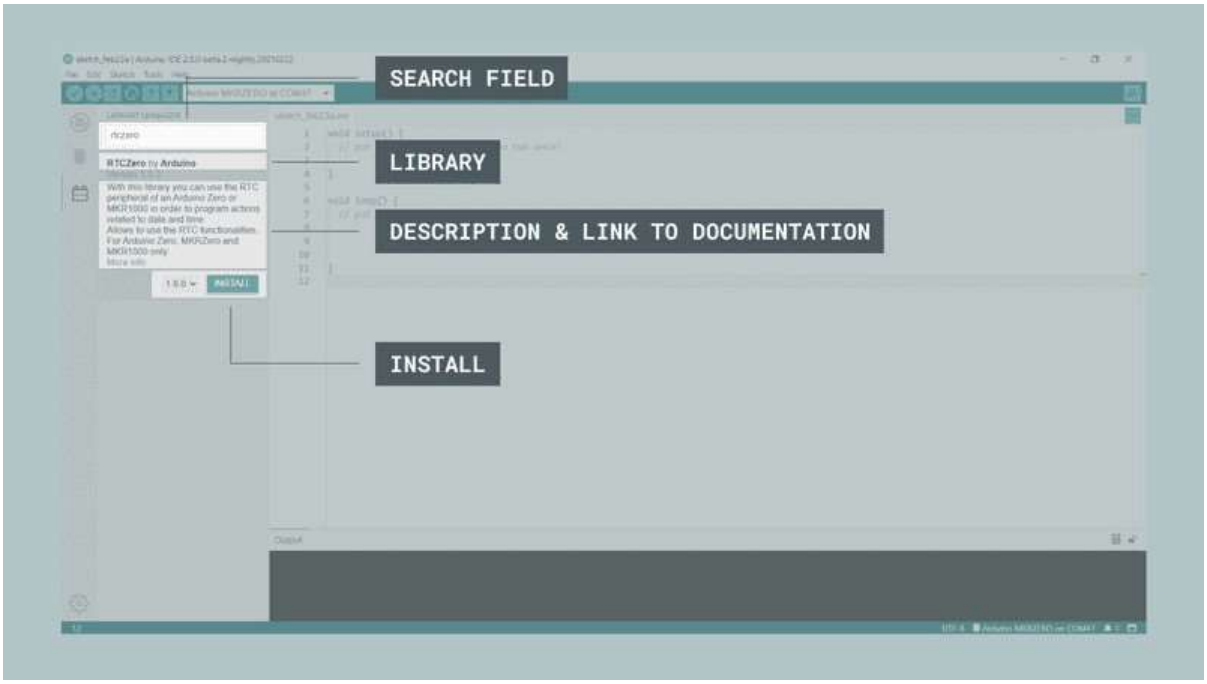
Arduino has many official libraries, but the real heroes are the Arduino community, who develop, maintain and improve their libraries on a regular basis.

Installing a library

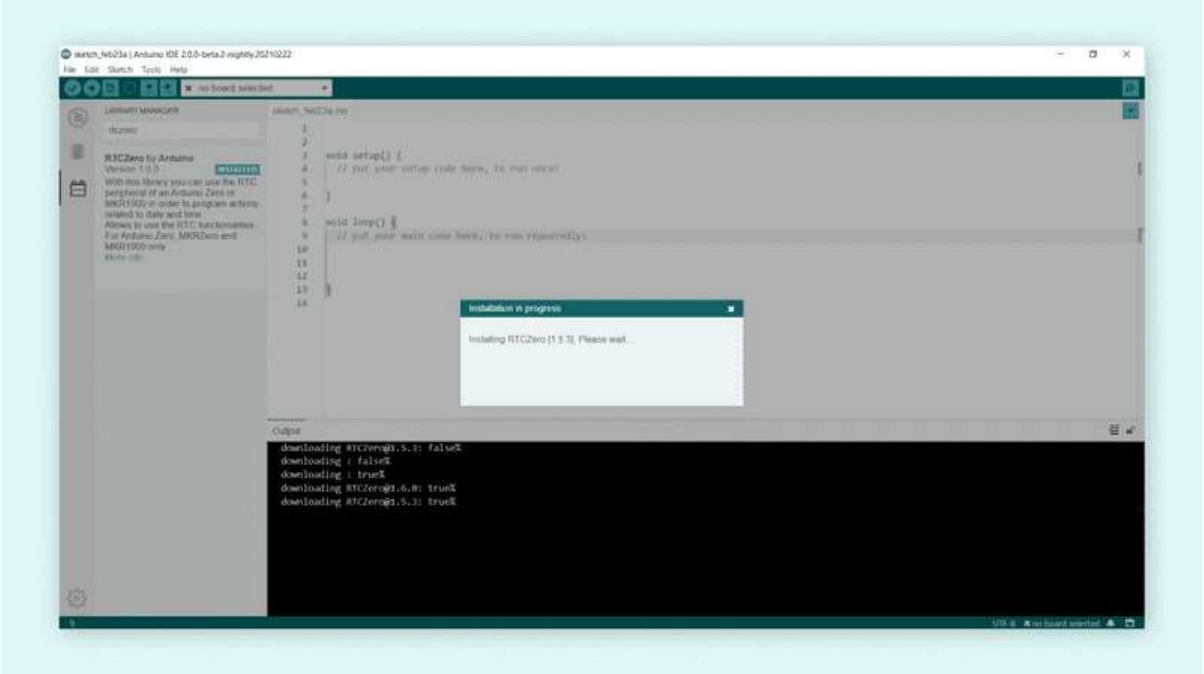
Installing a library is quick and easy, but let's take a look at what need to do.

1. Open the Arduino IDE 2.0.
2. With the editor open, let's take a look at the left column. Here, can see a couple of icons. Let's click the on the **"library"** icon.

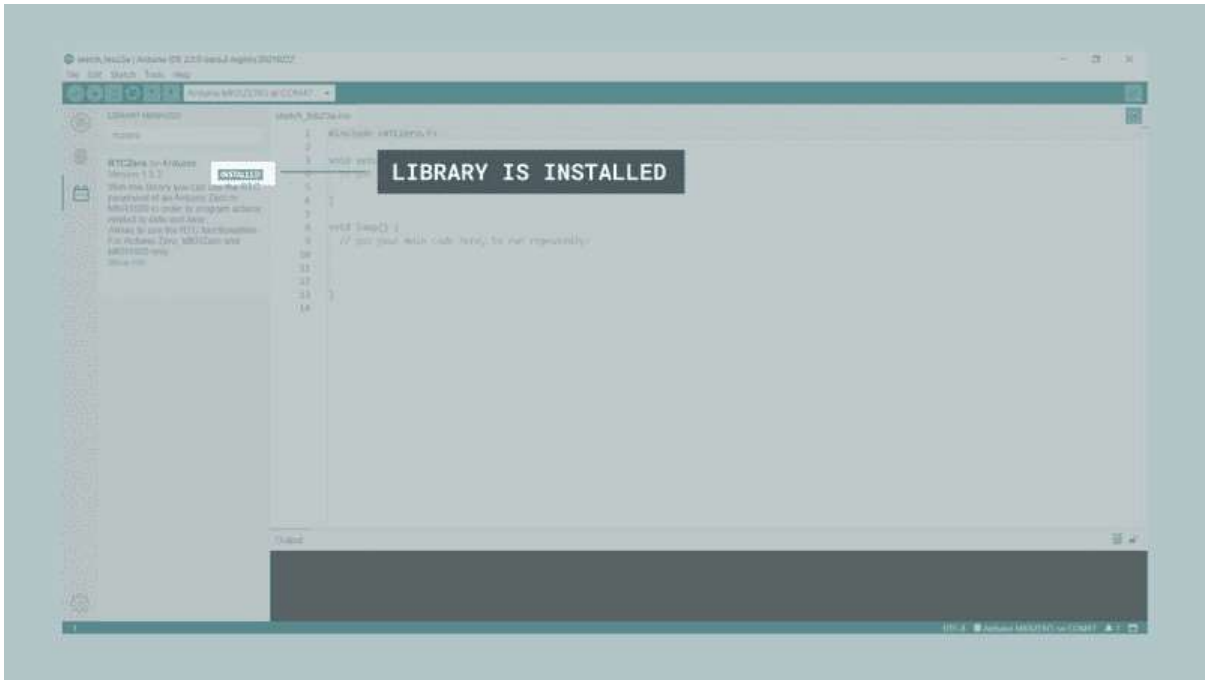
3. A list will now appear of all available libraries, where can also search for the library want to use. In this example, are going to install the **RTCZero** library. Click on the **"INSTALL"** button to install the library.



4. This process should not take too long, but allow up to a minute to install it.



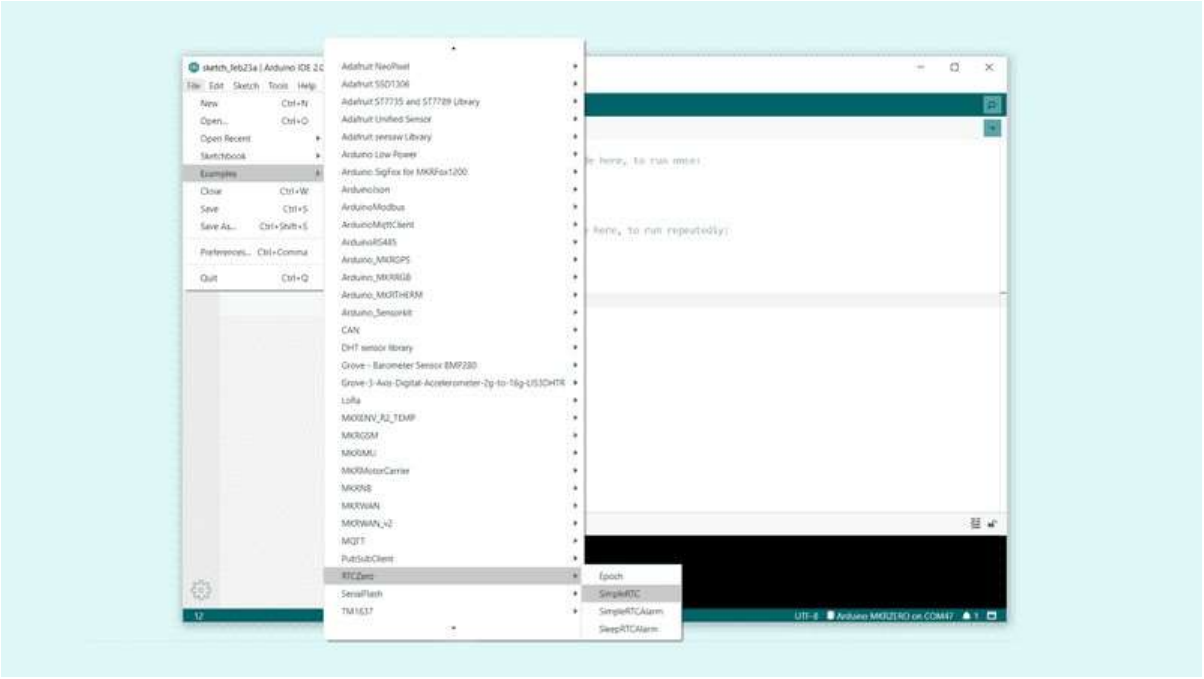
5. When it is finished, can take a look at the library in the library manager column, where it should say **"INSTALLED"**.



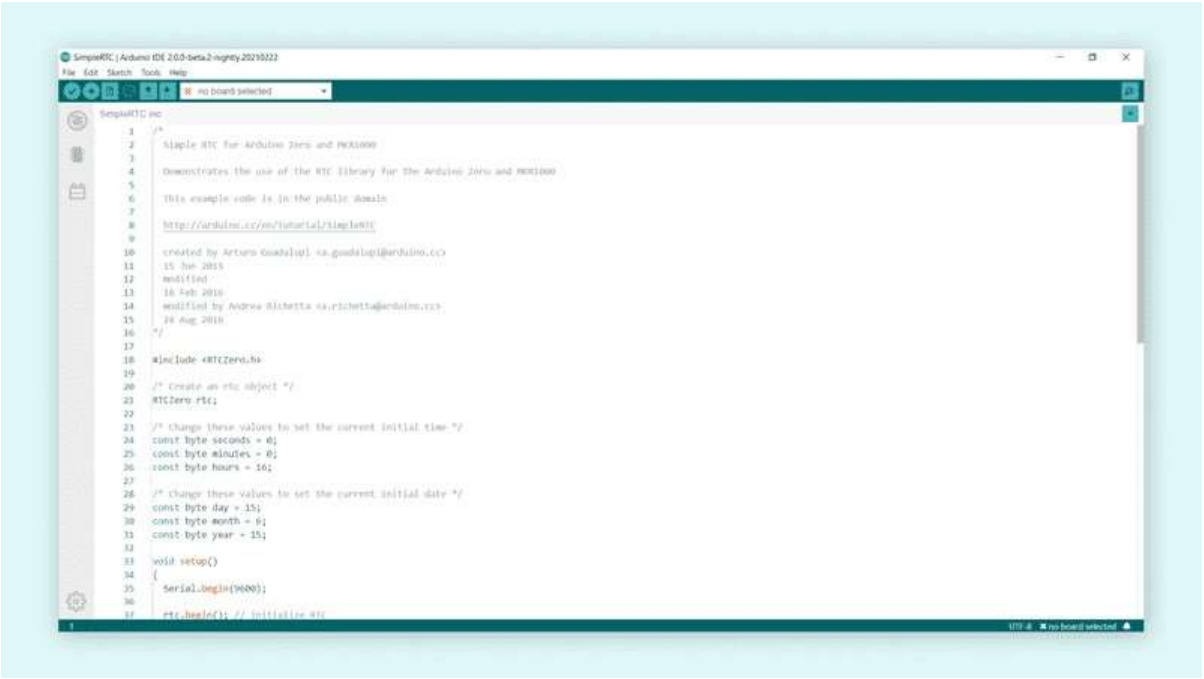
Including a library

To use a library, first need to include the library at the top of the sketch.

Almost all libraries come with already made examples that can use. These are accessible through **File > Examples > {Library} > {Example}**. In this example, are choosing the **RTCZero > SimpleRTC**.



The chosen example will now open up in a new window, and can start using it hover.



Conclusion

In conclusion, this detailed introduction to the Arduino platform and programming provides a clear foundation for understanding its components, functionalities, and applications. From its versatile hardware and intuitive software to its beginner-friendly programming structure, Arduino empowers users to innovate across various fields like robotics, IoT, and automation.

The explanation offered not only highlights the practical aspects of Arduino but also underscores its significance in modern technology and education. With its open-source nature and robust community support, Arduino continues to inspire creativity and problem-solving, making it an essential tool for both novices and experts in embedded systems development.