Hardware components and software tools

Project hardware components and software tools can vary widely depending on the specific project requirements and goals.

Project hardware components

- Microcontrollers (Arduino, Raspberry Pi, etc.)
- Sensors (ultrasonic, camera, light, motion, etc.)
- Actuators (motors & drivers, servos, relays, etc.)
- Displays (LCD, OLED, LED, etc.)
- Communication modules (Wi-Fi, Bluetooth, RFID, etc.)
- Power supply (batteries, power adapters, etc.)

Software Tools:

- Integrated Development Environments (IDEs) (Arduino IDE, Visual Studio Code, etc.)
- Programming languages (C, Python, Matlab, etc.)
- Simulation software (Proteus, LTSpice,Multism, etc.)
- Data visualization and analysis tools (Matplotlib, Tableau, Excel, etc.)
- Version control systems (Git, SVN, etc.)

The choice of hardware components and software tools used in a project depends on the project's requirements and goals.

An Introduction to Controllers

Controllers play a vital role in various industries and applications, serving as the brain behind automated processes. They regulate and maintain desired conditions, making them essential for precision and efficiency.

- ***Understanding Controllers:***

Controllers are electronic devices that monitor and adjust systems based on predefined conditions. They receive input signals from sensors, process the data, and generate output signals to actuate the system. Controllers enable automation, allowing systems to function autonomously without constant human intervention.

- ***Types of controllers:***

There are several types of controllers used in automation and control systems. Each controller type has its own advantages and is suitable for different applications.

Here are some common types of controllers:

❖ On-Off Controllers:

On-off controllers, also known as bang-bang controllers, have a simple output that switches between two states based on upper and lower thresholds. When the measured variable exceeds the upper threshold, the controller output is turned off. When it falls below the lower threshold, the output is turned on. On-off controllers are simple and easy to implement, but they can lead to oscillations and have a limited control range.

- Proportional Controllers: Proportional controllers provide an output that is proportional to the error between the desired setpoint and the measured value. The control action is calculated by multiplying the error by a proportional gain. Proportional control reduces oscillations and improves system response, but it does not eliminate steady-state errors.

- Integral Controllers: Integral controllers, also known as integral action or reset controllers, calculate the sum of the error over time and adjust the output accordingly. The integral control action helps eliminate steady-state errors by continuously integrating the error signal. However, if not properly tuned, integral controllers can cause instability and overshoot.

- Derivative Controllers: Derivative controllers, also known as rate controllers, determine the rate of change of the error. The derivative control action is proportional to the rate of change of the error signal. Derivative control improves system response and stability by anticipating future errors and damping out oscillations. However, derivative control amplifies high-frequency noise and can lead to instability if not properly tuned.

- PID Controllers: PID controllers combine proportional, integral, and derivative control actions for comprehensive control. The PID algorithm calculates the control action by summing the proportional, integral, and derivative terms. PID controllers are widely used due to their ability to handle a wide range of control problems. They provide good control performance, reduced steady-state errors, and improved stability.

- ❖ Intelligent controllers, which use algorithms like fuzzy set theory and/or neural networks algorithms to represent uncertain or imprecise information, have the ability to emulate past experience or learn from the environment.

Each controller type has its advantages and disadvantages, and the choice of controller depends on the specific application and control requirements. Controllers can be implemented using analog components or digital systems.

- ○ ***Digital Controllers:***

Digital controllers, such as those controllers implemented on microcontrollers like Arduino or pic, use digital signals and algorithms to provide control.

Digital controllers revolutionized the field by replacing analog controllers. Unlike analog controllers that operate using continuous signals, digital controllers process discrete signals in the form of binary code. They offer greater accuracy, flexibility, and functionality.

Digital controllers use algorithms to process data, enabling advanced control strategies and precise adjustments.

Digital controllers typically consist of a microprocessor that executes the control algorithm, analog-to-digital converters (ADCs) that convert the process variable signals into digital signals, and digital-to-analog converters (DACs) that convert the digital control signals into analog signals to drive the actuators. The control algorithm is implemented as software running on the microprocessor.

- ***Microcontrollers:***

So, we need a microprocessor with a memory and some peripherals to perform different control algorithms

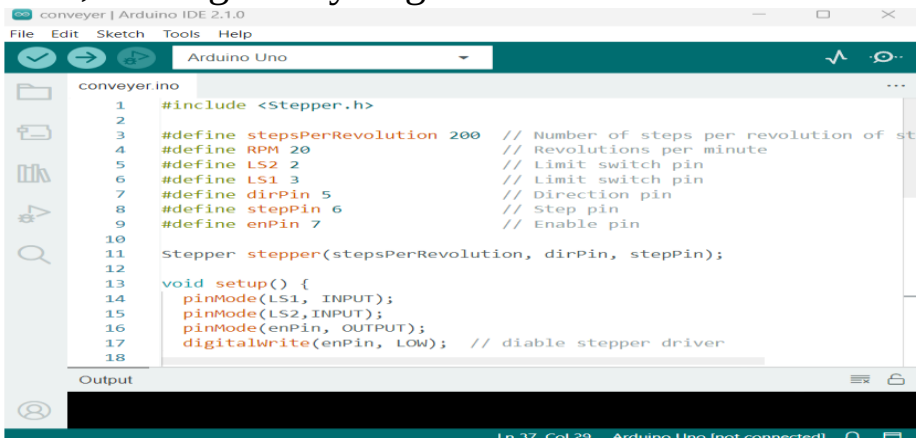
❖ *Arduino*

Arduino is a popular open-source platform that has gained widespread recognition among hobbyists, students, and professionals alike. It offers several advantages that make it an excellent choice for a wide range of projects

When considering a controller for your project, Arduino stands out for several reasons:

1. Cost-effectiveness: Arduino boards are affordable and readily available, making them accessible to individuals with limited budgets. This affordability allows for experimentation and prototyping without breaking the bank.

2. User-friendly: Arduino boards are designed with simplicity in mind. They come with an integrated development environment (IDE) that facilitates programming even for beginners. The IDE provides a user-friendly interface and a vast library of pre-written code, making it easy to get started.



3. Open-source Nature: Arduino is based on an open-source philosophy, which means the hardware and software designs are freely available for modification and distribution. This fosters innovation and encourages the development of new features and enhancements.
4. Extensive Library Support: Arduino has a vast library of pre-written code that simplifies programming tasks. Whether you need to interface with sensors, control motors, or communicate with other devices, chances are there's a library available to help you.

5. Learning Platform: Arduino serves as an excellent learning platform for beginners. Its intuitive interface and extensive documentation enable individuals to understand the basics of electronics and programming while working on tangible projects.

6. Expandability: Arduino boards are highly expandable, allowing you to connect additional modules and shields to extend their capabilities. This flexibility enables you to adapt and scale your project as needed.

○ *Arduino families*

➢ *Nano Family*

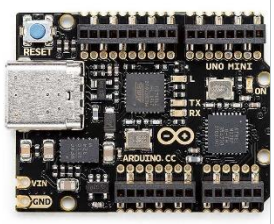
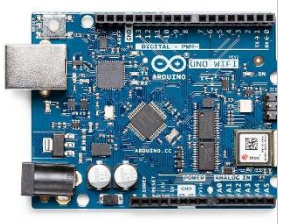
The Nano Family is a set of boards with a tiny footprint, packed with features. It ranges from the inexpensive, basic Nano Every, to the more feature-packed Nano 33 BLE Sense / Nano RP2040 Connect that has Bluetooth® / Wi-Fi radio modules. These boards also have a set of embedded sensors, such as temperature/humidity, pressure, gesture, microphone and more. They can also be programmed with MicroPython and supports Machine Learning.

			
Arduino Nano 33 IoT	Arduino Nano	Arduino Nano 33 BLE	Arduino Nano Every

Classic Family

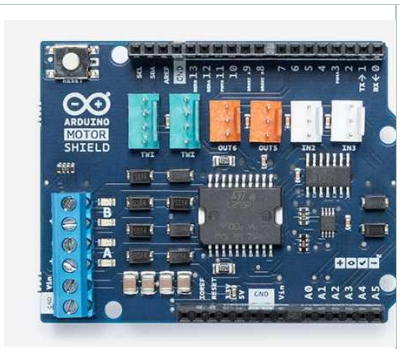
In the Classic Family, you will find boards such as the legendary Arduino UNO and other classics such as Leonardo & Micro. These boards are considered the backbone of the Arduino project and have been a success for many years (and more to come).

Boards

			
Arduino UNO R3	Arduino Mega 2560 Rev3	Arduino Leonardo	Arduino UNO Mini Limited Edition
			

Arduino Due	Arduino Micro	Arduino Zero	Arduino UNO WiFi Rev2

Shields

		
Arduino Motor Shield Rev3	Arduino 4 Relay Shield	Arduino Ethernet Shield Rev2

Arduino Uno: “used Mc**”**

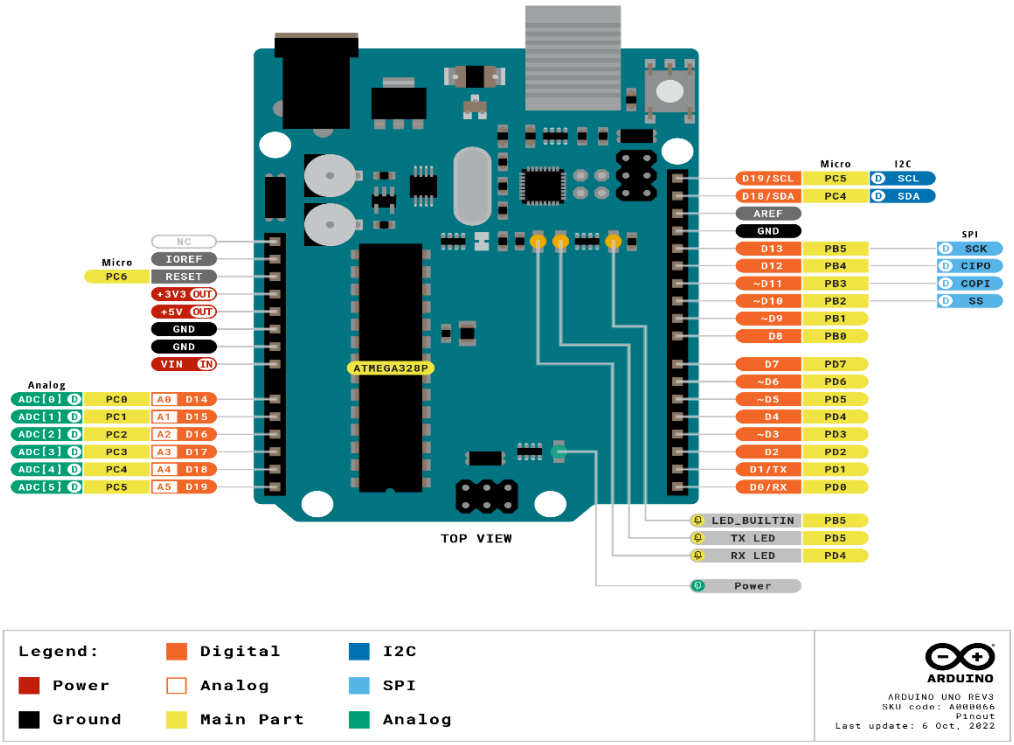
is a microcontroller board based on the ATmega328P datasheet

It has 14 digital input/output pins => of which 6 can be used as PWM outputs, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases

Tech specs

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g



The Arduino Mega 2560

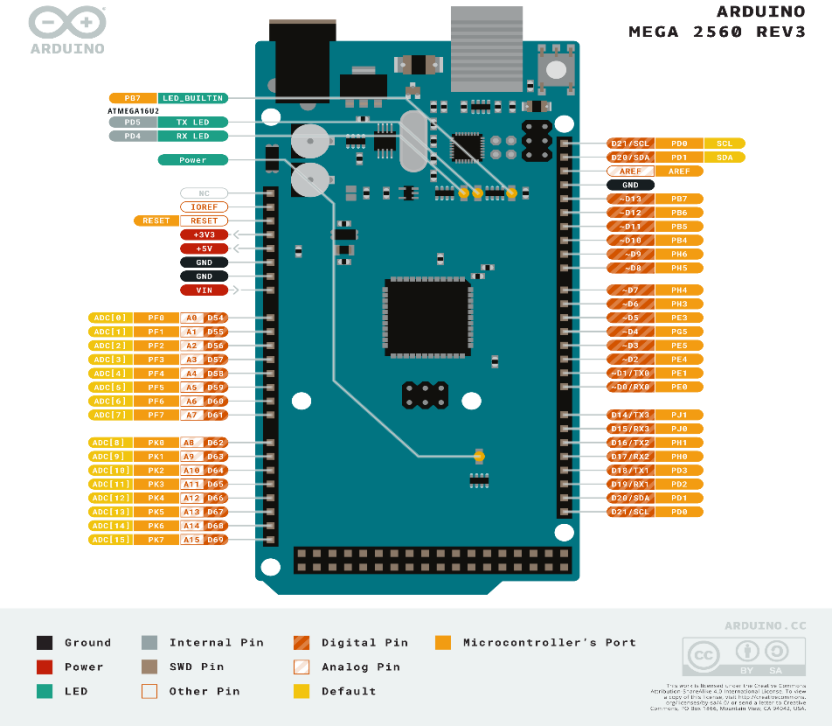
is a microcontroller board based on the ATmega2560.

It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

The Mega 2560 board is compatible with most shields designed for the Uno.

Tech specs

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

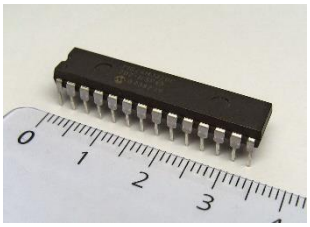


2.1.2.4 Arduino Leonardo:

The Leonardo is Arduino's first development board to use one microcontroller with built-in USB. This means that it can be cheaper and simpler. Also, because the board is handling USB directly, code libraries are available which allow the board to emulate a computer keyboard, mouse, and more!

Arduino Leonardo.

PIC Microcontroller



PIC stands for Programmable Interface Controllers microcontrollers are the world's smallest microcontrollers that can be programmed to carry out a huge range of tasks.

These microcontrollers are found in many electronic devices such as phones, computer control systems, alarm systems, embedded systems, etc.

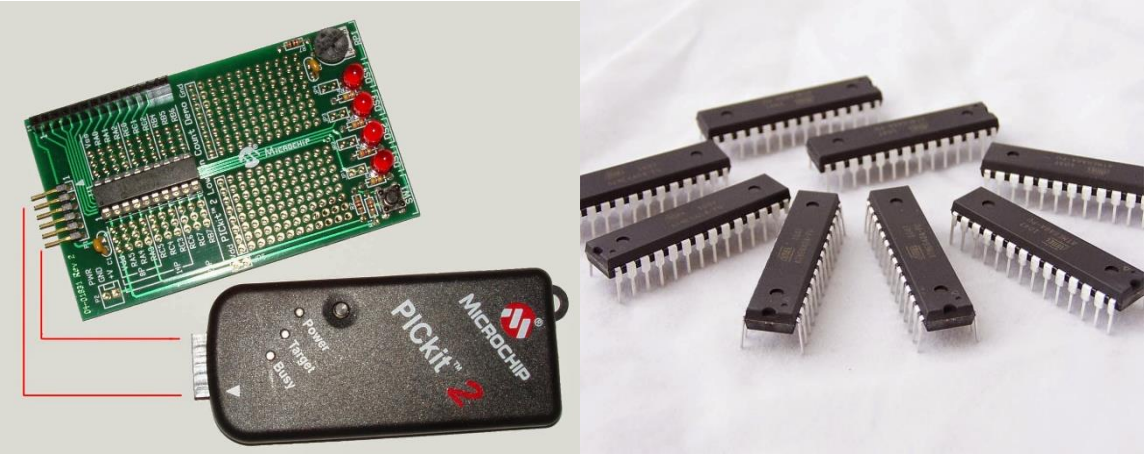
PIC microcontrollers are a very popular family of microcontrollers made by Microchip Technology.

- PIC all models use Flash memory for program storage, and newer models allow the PIC to reprogram itself.
- Program memory and data memory are separated. Data memory is 8-bit, 16-bit and 32-bit.
- Program instructions vary in bit-count by family of PIC, and may be 12, 14, 16, or 24 bits long.
- The instruction set also varies by model, with more powerful chips adding instructions for digital signal processing functions.

- The hardware capabilities of PIC devices range from 8-pin DIP chips up to 100-pin SMD chips, with discrete I/O pins, ADC and DAC modules, and communications ports such as UART, I2C, CAN, and even USB. Low-power and high-speed variations exist for many types.

The manufacturer supplies computer software for development known as MPLAB, assemblers and C/C++ compilers, and programmer/debugger hardware under the MPLAB and PICKit series.

MikroC is the easiest programming software for beginners, because it gives a lot of libraries, sample and debugging tools. Third party and some open-source tools are also available. Some parts have in-circuit programming capability, low-cost development programmers are available as well as high-production programmers.



Architecture of PIC Microcontroller

The PIC microcontroller architecture comprises of CPU, I/O ports, memory organization, A/D converter, timers/counters, interrupts, serial communication, oscillator and CCP module which are discussed in detailed below.