UNIT 3

IOT AND ARDUINO PROGRAMMING

1.1 INTRODUCTION TO THE CONCEPT OF IOT DEVICES

The Internet of Things Functional Blocks is the foundation of the IoT ecosystem. Companies are fast discovering ways to leverage the Internet of Things (IoT) to improve their efficiency as it grows in popularity. While the Internet of Things has numerous advantages, businesses are trying to comprehend how to incorporate technology into their work and daily lives. This article will look at that the IoT functional blocks and how they interact to produce a functional IoT system.

The Internet of Things (IoT) is a rapidly growing industry, with more and more devices becoming connected on a regular basis. The Internet of Things devices range from small sensors to huge machines and systems, but they all have one thing in common: they generate data. IoT systems use functional blocks to make sense of this data and extract value from it. These blocks are distinct components of the IoT system that carry out specialized functions.

In this article, we will discuss more about functional blocks in IoT.

IoT Functional Blocks

IoT systems are composed of a number of building blocks, including sensors/actuators, connectivity, security, services, etc. The functional blocks are responsible for sensing, verification, actuation, management, and communication.

These functional blocks are made up of devices that handle interactions between a web server and the client, enable controls and monitoring functions, manage data transfer, secure the IoT system through authentication and various purposes, and offer an interface for monitoring and managing various concepts. Let’s gather more information about the IoT Functional Blocks.

Sensor/Actuator block

The sensor/actuator block serves as the data entry point in an IoT system. Sensors collect data from their surroundings, whereas actuators drive physical processes. Sensors gather data on temperature, humidity, light, motion, and other variables, whereas actuators turn on lights, open

doors, and control machines. These gadgets work together to collect data and operate in the physical world.

Connectivity Block

Once the sensor/actuator block has collected data, it must be sent to the remainder of the system. This is when the connection block enters the picture. The connectivity block is in charge of creating and managing communication channels amongst IoT system devices. This can be accomplished with the use of several technologies such as Wi-Fi, Bluetooth, ZigBee, and cellular networks.

Data Processing Block

The obtained data is examined and processed in the data processing block. This block is in charge of filtering out noise and irrelevant data, converting the data into an easily studied format, and recognizing patterns and anomalies in the data. This block can also execute real-time analysis, enabling speedy data-driven decisions.

Application Block

The application block is the component of the IoT system that gives value to the end user. This block is in charge of utilizing the processed data to provide a specified function or service. An application block, for example, could be used to provide insights into energy usage in a building or to adjust the temperature in a greenhouse.

Security Block

The security block is in charge of assuring the IoT system's security and protection against illegal access. This block is in charge of authentication and authorization, as well as data encryption during transmission and storage. It also handles intrusion detection and response, assisting in the prevention and mitigation of threats.

Management Block

The management block is in charge of overseeing the overall operation of the IoT system. This block is capable of handling device configuration, firmware updates, and system monitoring.

It can also give analytics and reporting, allowing system administrators to understand how the system is operating and find areas for improvement.

Advantages of IoT Functional Blocks

IoT Functional Blocks provide various advantages to enterprises and people interested in implementing IoT solutions.

These advantages include −

Scalability

IoT Scalability is built into Functional Blocks, allowing enterprises to add new devices and services to their IoT system as needed. The capacity to scale assures that an IoT system can grow and react to changing business needs and future technology.

Interoperability

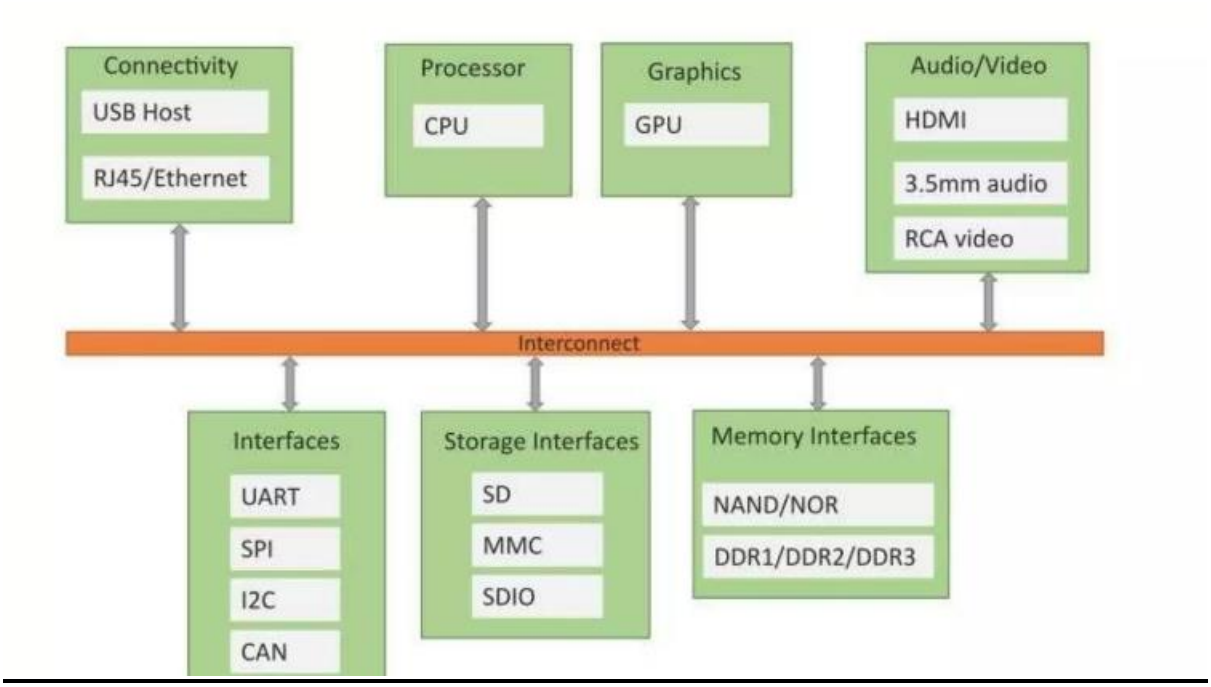
IoT Functional Blocks is a standardized architecture for developing IoT solutions. This standardization means that devices and services from various suppliers may function seamlessly together, enhancing interoperability and lowering integration costs.

Modularity

Because IoT Functional Blocks are modular, they may be swapped, modified, or added as needed. Because of this versatility, enterprises can select the optimal components for their IoT system and easily swap out components as needed.

Flexibility

IoT Functional Blocks offer a variety of deployment choices. Depending on their needs, businesses can implement an IoT system on-premises, in the cloud, or in a hybrid approach.



1.2 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 were initially created to help the students with the non-technical background.

The designs of Arduino boards use a variety of controllers and microprocessors.

The Arduino board consists of sets of analog and digital I/O (Input / Output) pins, which are further interfaced to **breadboard, expansion boards,** and other **circuits**. Such boards feature the model, Universal Serial Bus (**USB**), and **serial communication interfaces**, which are used for loading programs from the computers.

It also provides an **IDE** (Integrated Development Environment) project, which is based on the Processing Language to upload the code to the physical board.

The projects are authorized under the **GPL** and **LGPL**. The **GPL** is named **as GNU General Public License.** The licensed **LGPL** is named as **GNU Lesser General Public License**. It

allows the use of Arduino boards, it's software distribution, and can be manufactured by anyone.

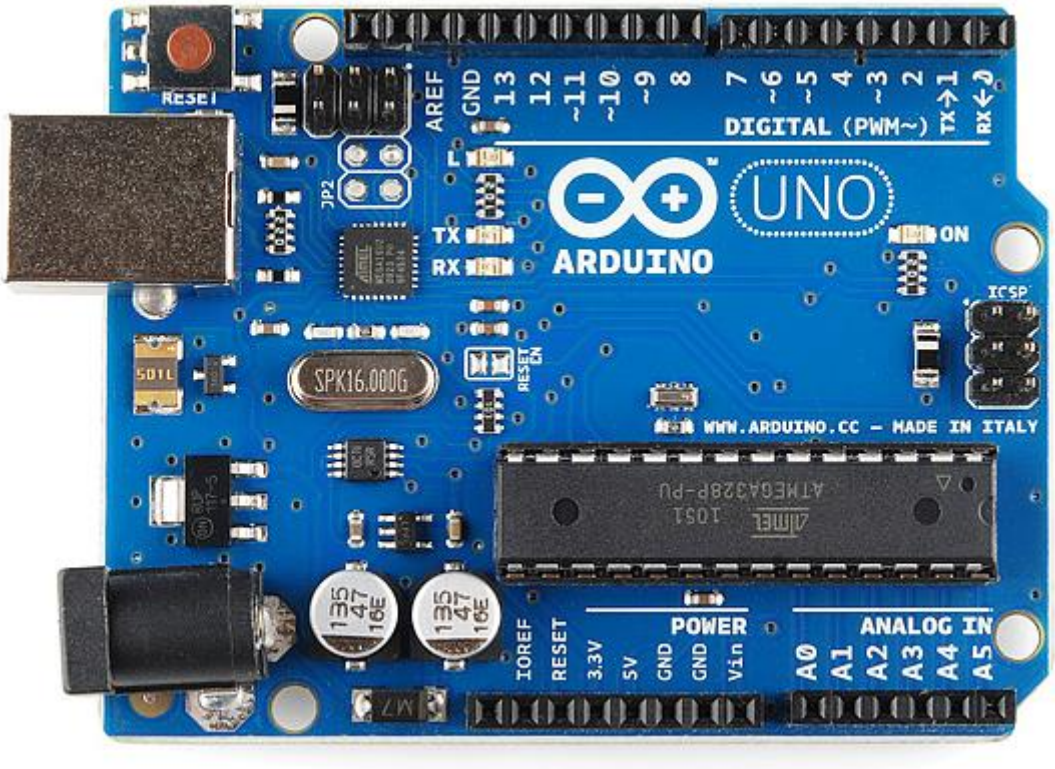
It is also available in the form of self practicing kits.

The Arduino is used for various purposes, such as:

Features

The features of Arduino are listed below:

- Arduino programming is a simplified version of C++, which makes the learning process easy.
- The Arduino IDE is used to control the functions of boards. It further sends the set of specifications to the microcontroller.
- Arduino does not need an extra board or piece to load new code.
- Arduino can read analog and digital input signals.
- The hardware and software platform is easy to use and implement.



Arduino is a prototype platform (open-source) based on an easy-to-use hardware and software. It consists of a circuit board, which can be programed (referred to as a microcontroller) and a ready-made software called Arduino IDE (Integrated Development Environment), which is used to write and upload the computer code to the physical board.

The key features are −

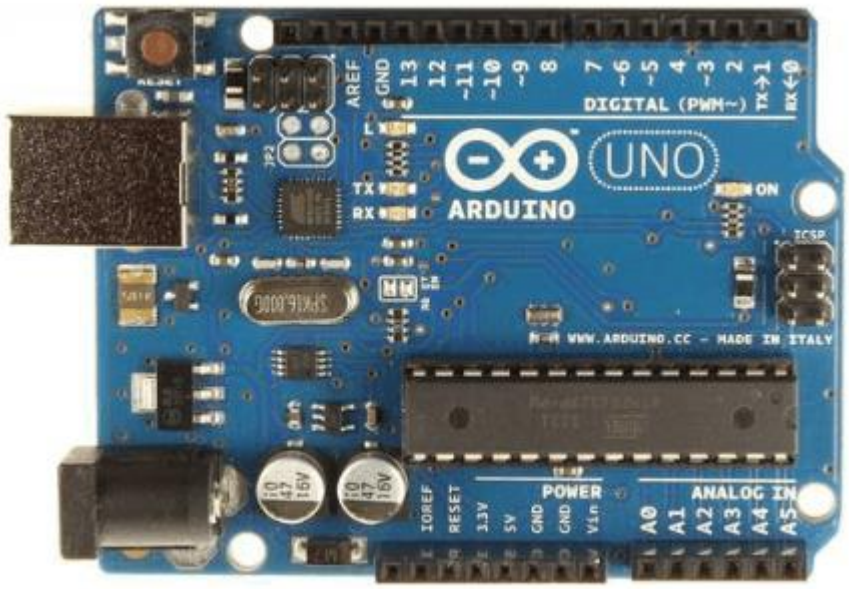
- Arduino boards are able to read analog or digital input signals from different sensors and turn it into an output such as activating a motor, turning LED on/off, connect to the cloud and many other actions.
- You can control your board functions by sending a set of instructions to the microcontroller on the board via Arduino IDE (referred to as uploading software).
- Unlike most previous programmable circuit boards, Arduino does not need an extra piece of hardware (called a programmer) in order to load a new code onto the board. You can simply use a USB cable.
- Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program.
- Finally, Arduino provides a standard form factor that breaks the functions of the micro-controller into a more accessible package.

The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

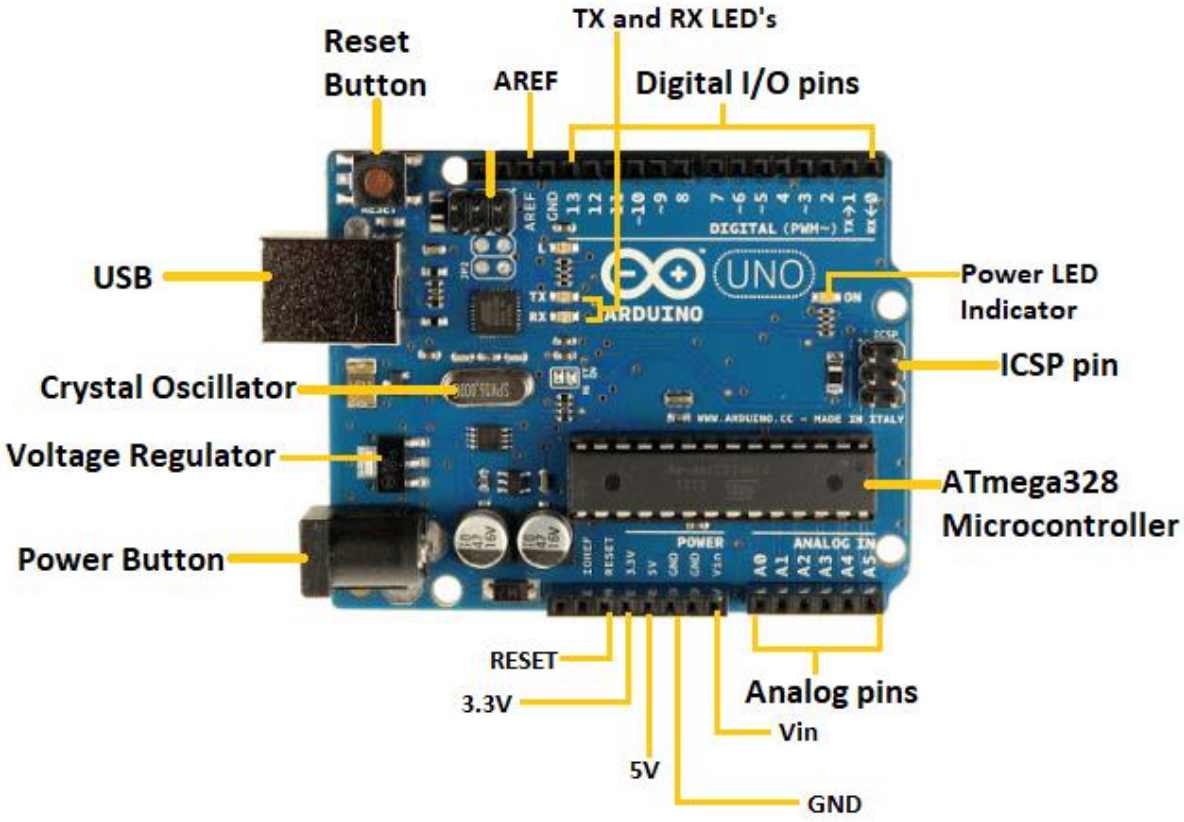
Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power 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 Arduino board is shown below:



The components of Arduino UNO board are shown below:



Let's discuss each component in detail.

- **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, timer, 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.
- **Power LED Indicator**- The ON status of LED shows the power is activated. When the power 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 power 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 powerful 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) pins.

Why is Arduino recommended over other boards for beginners?

The USB port in the Arduino board is used to connect the board to the computer using the USB cable. The cable acts as a serial port and as the power supply to interface the board. Such dual functioning makes it unique to recommend and easy to use for beginners.

What is the main difference between Arduino UNO and Arduino Nano?

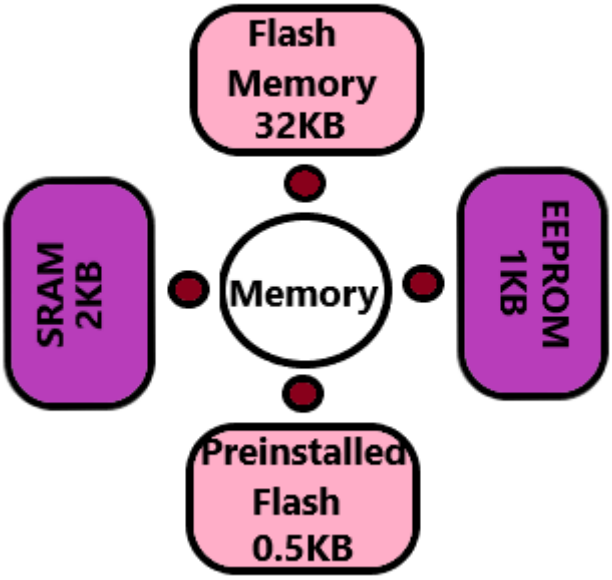
The Arduino Nano has a compact size and mini USB cable than the Arduino UNO.

What is the main difference between Arduino UNO and Arduino Mega?

The Arduino UNO is a standard board recommended to beginners, while Arduino Mega is used for complex projects due to its greater memory space.

Memory

The memory structure is shown in the below image:



The preinstalled flash has a bootloader, which takes the memory of 0.5 Kb.

Here, SRAM stands for Static Random Access Memory, and EEPROM stands for Electrically Erasable Programmable Read-Only Memory.

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 pis include 6 PWM pins, 6 analog pins, and 8 digital I/O pins.
- The PWM pins are Pulse Width Modulation capable pins.
- The crystal oscillator present in Arduino UNO comes with a frequency of 16MHz.
- It also has a 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 power from the external power supply. It can also draw power from the USB.

How to get started with Arduino UNO?

We can program the Arduino UNO using the Arduino IDE. The Arduino IDE is the Integral Development program, which is common to all the boards.

We can also use Arduino Web Editor, which allows us to upload sketches and write the code from our web browser (Google Chrome recommended) to any Arduino Board. It is an online platform.

The USB connection is essential to connect the computer with the board. After the connection, the PWR pins will light in green. It is a green power LED.

The steps to get started with Arduino UNO are listed below:

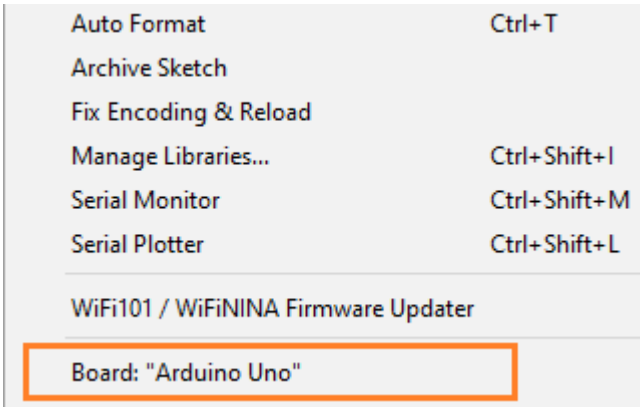
- Install the **drivers** of the board.

As soon we connect the board to the computer, Windows from XP to 10 will automatically install the board drivers.

But, if you have expanded or downloaded the zip package, follow the below steps:

1. Click on **Start** -> **Control Panel** -> **System and Security**.
2. Click on **System** -> **Device Manager** -> **Ports (COM &LPT)** -> **Arduino UNO (COMxx)**. If the COM &LPT is absent, look **Other Devices** -> **Unknown Device**.
3. Right-click to **Arduino UNO (COmxx)** -> **Update Driver Software** -> **Browse my computer for driver software.**
4. Select the file "**inf**" to navigate else, select "**ArduinoUNO.inf**" .
5. Installation Finished.

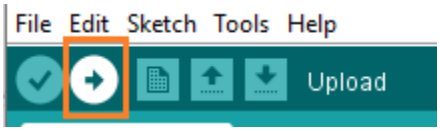
- Open the code or sketch written in the Arduino software.
- Select the type of board.
- Click on **'Tools'** and select **Board**, as shown below:



- Select the port. Click on the **Tools -> Port** (select the port). The port likely will be **COM3** or higher. For example, **COM6**, etc. The **COM1** and **COM2** ports will not appear, because these two ports are reserved for the hardware serial ports.

- Now, **upload** and **run** the written code or sketch.

To upload and run, click on the button present on the top panel of the Arduino display, as shown below:



1.3 TYPES OF ARDUINOS

There are different Arduino types, each offering unique features and capabilities. From basic microcontrollers to more advanced modules, let's explore some of the popular types of Arduino boards.

1. Arduino Uno R3

The Arduino Uno R3 is a popular board among DIY electronics that offers features, such as 14 digital input/output pins, 6 analog pins, and an ICSP (In-Circuit Serial Programming) header.

- It runs on the ATmega328P 16MHz microchip providing up to 5V voltage supply to attached components.
- While it uses a USB-B connector for the computer interface, this doesn't mean that projects requiring advanced skills cannot be built with it.
- The key specs include 2kB SRAM memory capability, 32kB flash storage space, and 1KB EEPROM chip along with UART, and I2C SPI communication capabilities which can also be replaced in case of any problem.
- It is a great choice for anyone wanting to get into DIY electronics and programming. It can be used for projects both simple and complex.

2. Arduino Nano

The Arduino Nano is highly popular among developers due to its small size yet considerable capabilities. It shares many of the same features as the UNO, but in a smaller form, making it great for use on breadboards and other tight spaces. The main differences from the larger model include:

- There is no onboard DC power jack connection. Instead, it relies on a mini USB port, which also handles UART bridge chip duties with FTDI's dedicated FT232 converter included. This means there's no need for the ATMega16U2 chipset.
- This combination provides powerful control at both modest cost and minimum space requirements, allowing creative designs to come within reach.

3. Arduino Micro

The Arduino Micro offers features similar to the Leonardo but is significantly smaller and only has 12 analog I/O pins together with its 20 digital ones.

- This board measures just 18mm wide by 48mm long. It is one of the smallest boards ever created by Arduino. You can create a keyboard, mouse, or other HID devices that require minimal space with this Arduino board.
- It comes equipped with an ATmega32U4 microcontroller containing 2.5kB SRAM, 32KB flash memory, and 1 KB EEPROM for data storage along with UART, IC2 & SPI communication protocols. It also has special built-in functions such as HID connectivity.

4. Arduino Leonardo

The Arduino Leonardo is different from the other boards. Given below are its specifications.

- It uses an ATmega32U4 chip in place of ATMega328P. It offers more pins for IO (20), PWM (7), and analog input signals (12).
- Instead of needing a second processor or a USB to UART bridge chip, this board has built-in USB communication. This allows it to connect directly with computers as either Human Interface Devices or Virtual COM port serial/CDC ports. This virtual connection enables us to program our Leonardo via its bootloader.

5. Arduino Micro

Arduino Micro is perfect for those who need the power of an Arduino Leonardo in a breadboard-friendly size.

- It offers all of the same functionalities as its larger counterpart, with only one slight difference.
- It doesn't have a DC input jack. However, you can still use this tiny board to act like an HID or Virtual COM port device just as you would on any other Leonardo model.

To understand the concept of Arduino boards and the different types of Arduino even better, you can take this comprehensive Arduino course.

6. Arduino Mega2560 Rev3

The Arduino Mega 2560 stands out amongst the other boards.

- It is designed for those situations which require many I/O or peripherals.
- With a powerful ATMega2560 processor, this board boasts an impressive 54 I/O pins (15 of them can be used as PWM outputs), 16 analog inputs, and 4 UARTs.
- Furthermore, there is more flash storage and SRAM on this basic Arduino than any other.
- Due to its capability, it has grown in popularity among open-source CNC makers and 3D printer builders.
- It can be easily implemented into various open-source Programmable Logic Controllers projects.

7. Arduino Nano 33 BLE

The Arduino Nano 33 BLE is an incredibly powerful board with the same form factor as its predecessor, the Arduino Nano.

- Featuring Nordic Semiconductors' advanced NRF52480 32-bit ARM Cortex-M4 processor running at 64MHz and 1MB flash plus 256KB SRAM memory capacity, it's a great choice for applications that need more processing power and storage capabilities than traditional boards provide.
- The most impressive attribute of this device has to be Bluetooth Low Energy (BLE) support.
- It allows users to leverage both client/host configurations with their compatible devices while taking advantage of energy efficiency benefits offered by connected wearables development.

8. Arduino Due

The Arduino Due is a more powerful upgrade of the popular Arduino Mega with its 54 Digital I/Os and 12 Analog inputs.

- Running at 84MHz, this 3.3V-only board offers an Atmel SAM3x8E Arm Cortex-M3 processor along with USB OTG, DAC, and JTAG interfaces for extra connectivity options.
- It's important to note that this device's I/O pins are not 5V tolerant so be sure you keep that in mind when getting started on your project.

9. LilyPad Arduino Board

Leah Buechley and SparkFun collaborated to create the LilyPad Arduino board, a wearable e-textile technology. This creative design comes complete with large connecting pads and a flat back, allowing it to be easily sewn into clothing using conductive thread. Alongside this are I/O, power, and sensor boards created for use in e-textiles.

10. Arduino Bluetooth

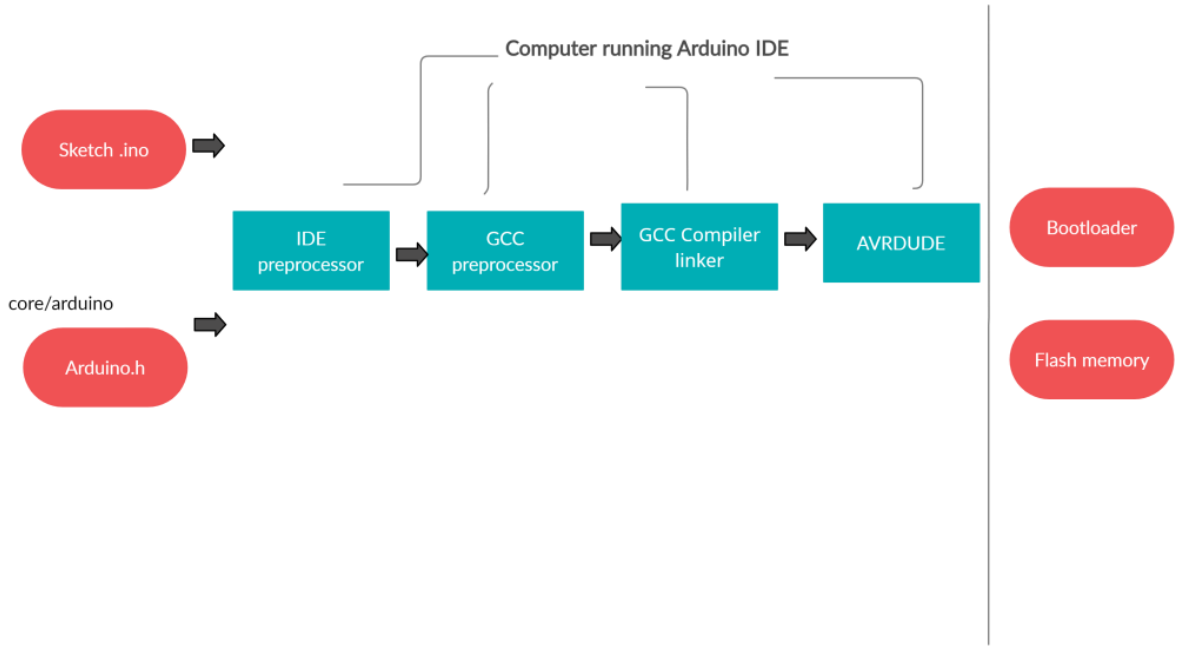
Arduino Bluetooth is one of the popular types of Arduino boards. The Arduino BT board is powered by the ATmega168 microcontroller and includes components such as digital pins (16), analog pins (6), a crystal oscillator (16MHz) reset button, screw terminals for power connections, and an ICSP header. Programming with this Bluetooth-enabled controller can be done wirelessly via a Bluetooth connection.

11. Arduino MKR FOX 1200

This board is an ideal choice if your project requires Sigfox connectivity. The Arduino MKR FOX 1200 provides a cost-effective solution that can be used in the European region, allowing you to collect data 24/7 without needing any maintenance or supervision. Using a SAMD21 MCU and Microchip's Smart RF ATA852o for its SIGFOX connection, this board offers seamless integration with projects requiring SIGFOX capabilities.

1.4 ARDUINO TOOL CHAIN

The USB device refer to connect to your Computer USB and loading with a program called Arduino Sketch IDE. The communication is seamless. Let discuss on the flow.

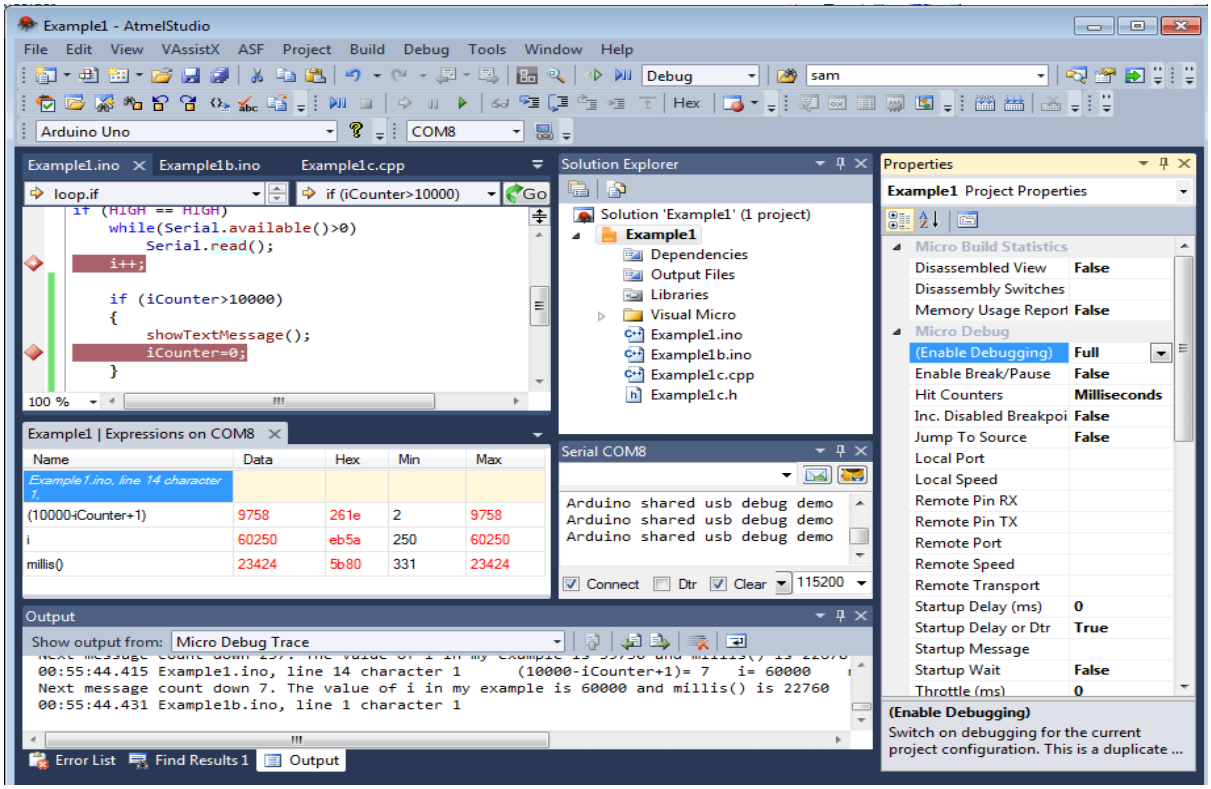


- Sketch code press upload button. The Arduino toolchain is ran to perform the uploading of the code.
- Arduino sketches held the file in **.ino** extension in a folder with same name.
- Arduino IED starts to perform the following tasks
- Arduino IDE preprocessor assembles the files on the sketch. 1 file is found in the folder.
- As there are many boards with different pins layout, there is a folder /hardware/Arduino/variant folder that kept all other type of Arduino pin setup.
- Combining all the files, GCC compiler (which is open-source C++ compiler) bundled part of Arduino distribution.
    - Preprocessor interprets all the #if and #define commands and determines what actually goes into the build.
    - Next, the code is compiled and linked into a single executable file for type of proessor used by board.
    - After compiler finished, another piece open source called avrdude actually sends the executable code saved as hexadecimal binary to the board over USB serial interface.
- There is a program called bootloader on Arduino board runs every briefly when Arduino is reset.

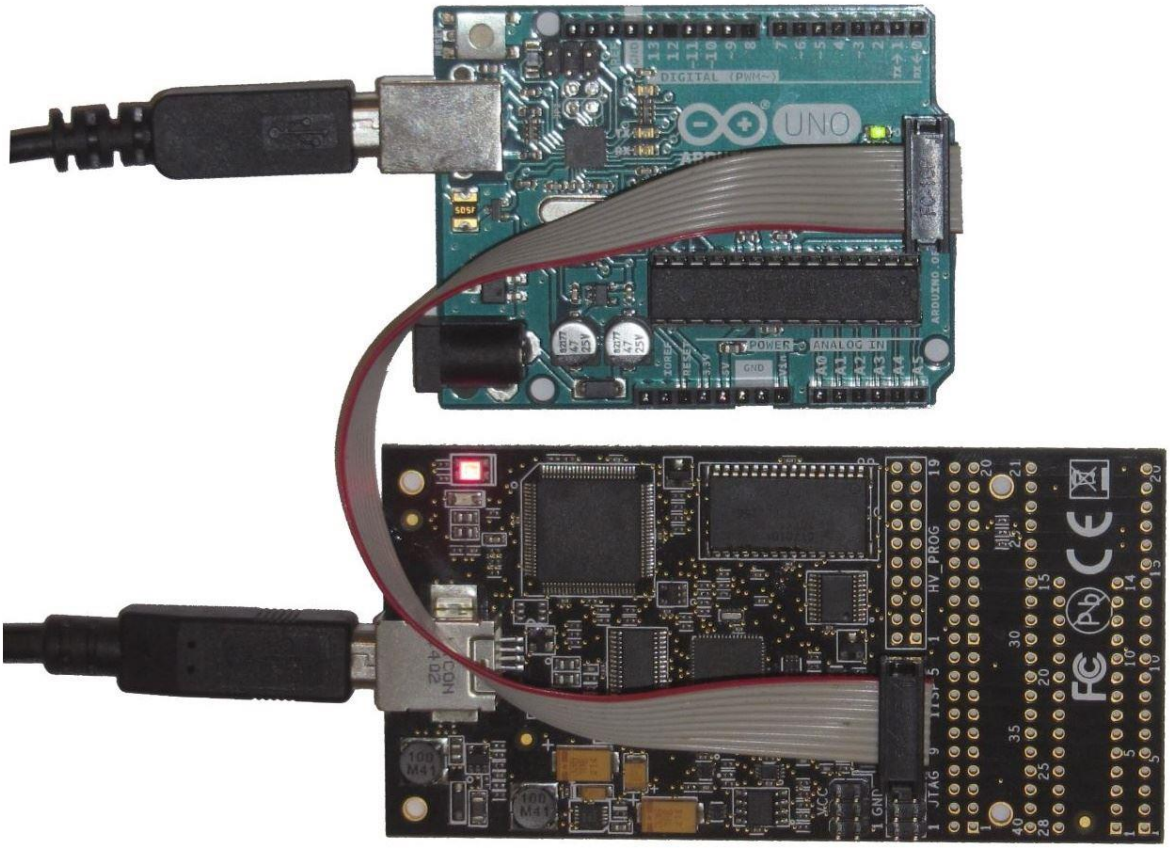
- When serial communication starts, the hardware serial link forces a reset to give the bootloader chance to check for any incoming sketches.
- If sketch exist, Arduino programs will unpack the hexadecimal into binary.
- It stores the sketch in the flash memory.

AVR Studio

Since bootloader removes the needs to have special MCU dependent emulator, but you can directly upload using AVRISPv2, AV Dragon and USBtinyISP to flash to the AVRboard bypassing the bootloader. This can speed up the board loading time as bootloader spends second to check incoming program each time.



- AVR Studio is the software for programming the microcontrollers used in Arduinos.
- Using hardware programmer emulator to upload rather than USB.
- Emulator is more complex than Sketch DE.
- Most of the board comes with 6-pin header that can be directly connect to ay AVR Dragon to rograms actually running on ATmega chip.



Finally, you can skip Bootloader if you are comfortable using AVR tools to uplad your program into Arduino board.

1.5 ARDUINO PROGRAMMING STRUCTURE

The Arduino software is open-source. The source code for the Java environment is released under the GPL and the C/C++ microcontroller libraries are under the LGPL.

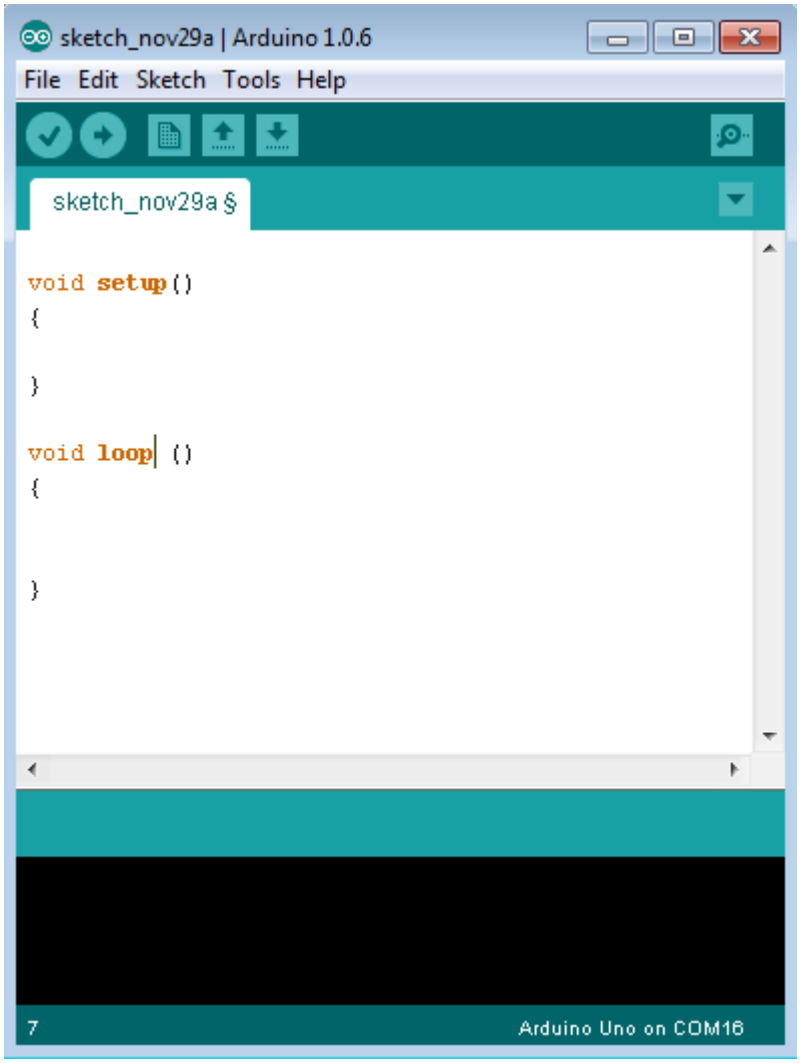
Sketch − The first new terminology is the Arduino program called "**sketch**".

Structure

Arduino programs can be divided in three main parts: **Structure, Values** (variables and constants), and **Functions**. In this tutorial, we will learn about the Arduino software program, step by step, and how we can write the program without any syntax or compilation error.

Let us start with the **Structure**. Software structure consist of two main functions −

- Setup( ) function
- Loop( ) function



Void setup ( ) {

}

- **PURPOSE** – The **setup()** function is called when a sketch starts. Use it to initialize the variables, pin modes, start using libraries, etc. The setup function will only run once, after each power up or reset of the Arduino board.
- **INPUT** – -
- **OUTPUT** – -
- **RETURN** – -

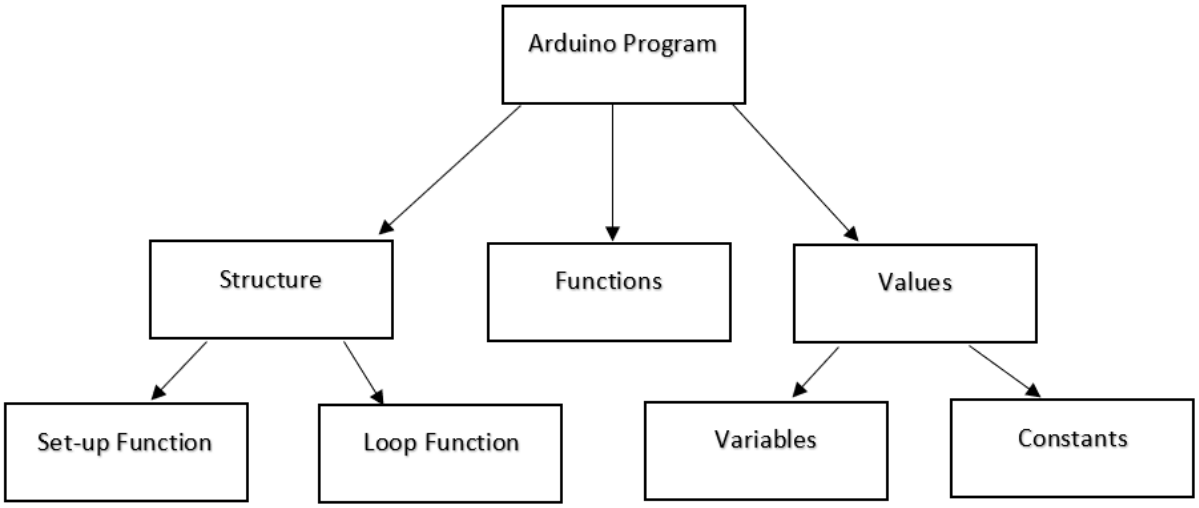
Void Loop ( ) {

}

- **PURPOSE** − After creating a **setup()** function, which initializes and sets the initial values, the **loop()** function does precisely what its name suggests, and loops consecutively, allowing your program to change and respond. Use it to actively control the Arduino board.
- **INPUT** − -
- **OUTPUT** − -
- **RETURN** − -

Arduino Programing Overview

The Arduino program is divided into three main parts that are structure, values, and functions.

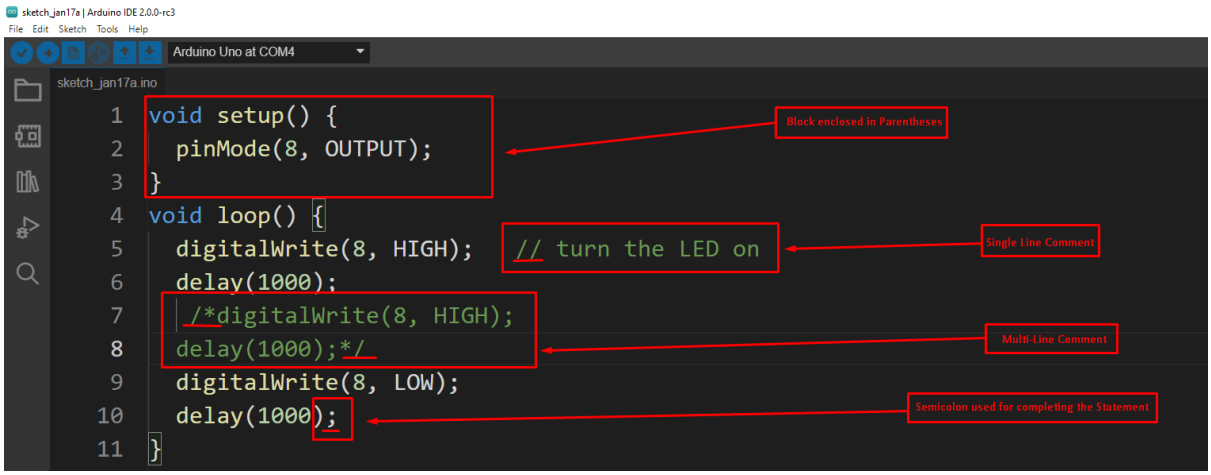


When writing a code, the important thing is following the syntax of the language being used because in order to run the code successfully the correct syntax is necessary. So, when writing a program for Arduino following syntax should be followed:

- To complete the statement a **semicolon ";"** is used at the end of the statement.
- To enclose the block **parenthesis** "{}" are used. Block in a program contains some statements, declaration of the variables, functions, or loops.
- Comments can be written for each statement in the code to better understand the statement functionality. It can be done by using double **forward slash "//"** at the start of the comment if there is only a single **line comment**. However, if there are **multi line comments** in a row, a **forward slash asterisk "/*"** at the start

and **asterisk forward slash "*/"** at the end of the comment. Comments can also be used to exclude any statement.

The figure below given gives a much better understanding of the syntax used for coding in Arduino software:



After understanding the syntax, let's move towards how to use variables in the Arduino program and what type of variables are used in program structure. To store any values which will be used in the program can be a number or an alphabet.

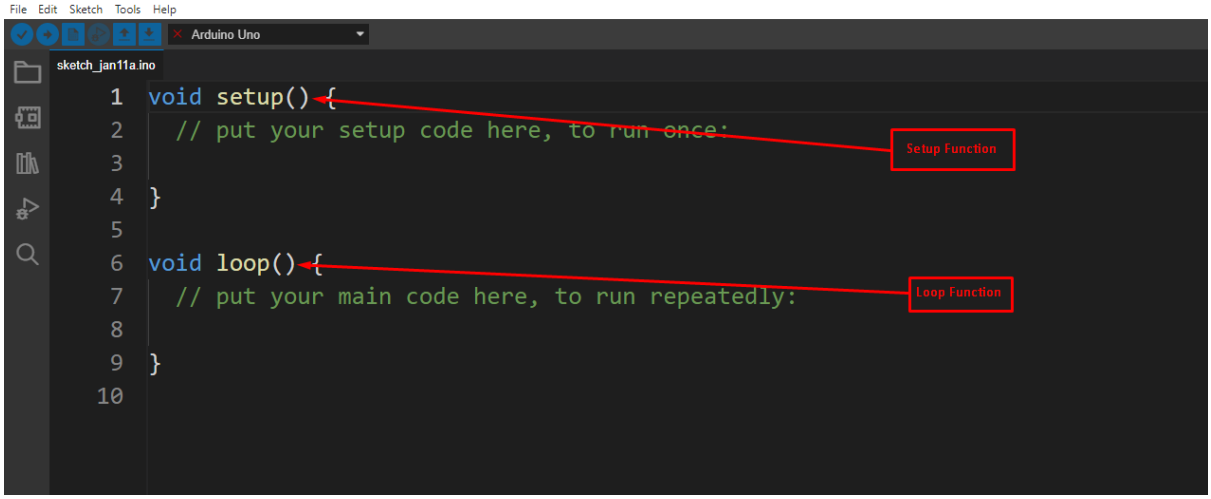
Using the variables gives the option of saving, changing, updating and accessing the information when the program is running. There are different types of variables that can be used including char, int, double, string, float, unsigned int, long and unsigned long.

The following are operators used in the programing of Arduino:

- For **assigning** any value to a variable or a character equal to "=" sign is used
- There are different **mathematical operators** like percentage, multiply, subtraction, addition can be used (%, +, *, -, /)
- For **comparison** of the different values the operators like less than equal to, greater than equal to, equal to, less than, greater than are used (==, ,=)
- **Logical operators** are used to define the conditional statements such as AND (&&), NOT(!) and OR (||) operators

Arduino Program structure

The Arduino program structure is divided into two functions: the set-up function and the loop function.



The **setup()** function contains initialization of the libraries, variables used for the code. Similarly, pin modes of the Arduino are also declared in this function. It also initializes the communication between the Arduino board and the computer. It only runs once.
The **loop()** function keeps on repeating the instructions and actively controls and monitors the Arduino.

1.6 INTRODUCTION TO ARDUINO SHIELDS

Arduino shields are the boards, which are plugged over the Arduino board to expand its functionalities. There are different varieties of shields used for various tasks, such as Arduino motor shields, Arduino communication shields, etc.

Shield is defined as the hardware device that can be mounted over the board to increase the capabilities of the projects. It also makes our work easy. For example, Ethernet shields are used to connect the Arduino board to the Internet.

The pin position of the shields is similar to the Arduino boards. We can also connect the modules and sensors to the shields with the help of the connection cable.

Arduino motor shields help us to control the motors with the Arduino board.

Backward Skip 10sPlay VideoForward Skip 10s

Why do we need Shields?

The advantages of using Arduino shields are listed below:

- It adds new functionalities to the Arduino projects.
- The shields can be attached and detached easily from the Arduino board. It does not require any complex wiring.
- It is easy to connect the shields by mounting them over the Arduino board.
- The hardware components on the shields can be easily implemented.

Types of Shields

The popular Arduino shields are listed below:

- **Ethernet shield**
- **Xbee Shield**
- **Proto shield**
- **Relay shield**
- **Motor shield**
- **LCD shield**
- **Bluetooth shield**
- **Capacitive Touchpad Shield**

Let's discuss the shields that are listed above:

Ethernet shield

- The Ethernet shields are used to connect the Arduino board to the Internet. We need to mount the shield on the top of the specified Arduino board.
- The USB port will play the usual role to upload sketches on the board.
- The latest version of Ethernet shields consists of a micro SD card slot. The micro SD card slot can be interfaced with the help of the SD card library.

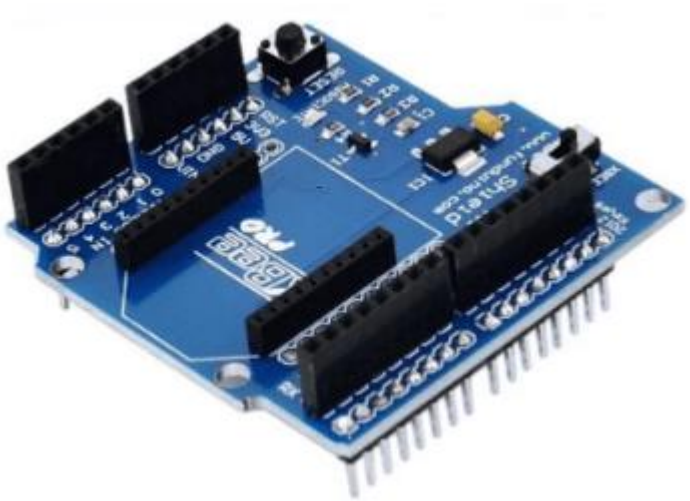
The Ethernet shield is shown below:



- We can also connect another shield on the top of the Ethernet shield. It means that we can also mount two shields on the top of the Arduino board.

Xbee Shield

- We can communicate wirelessly with the Arduino board by using the Xbee Shield with Zigbee.
- It reduces the hassle of the cable, which makes Xbee a wireless communication model.
- The Xbee wireless module allows us to communicate outdoor upto 300 feet and indoor upto 100 feet.
- The Xbee shield is shown below:



- It can also be used with different models of Xbee.

Proto shield

- Proto shields are designed for custom circuits.
- We can solder electronic circuits directly on the shield.
- The shield consists of two LED pads, two power lines, and SPI signal pads.
- The IOREF (Input Output voltage REFerence) and GND (Ground) are the two power lines on the board.

The proto shield is shown below:



- We can also solder the SMD (Surface Mount Device) ICs on the prototyping area. A maximum of 24 pins can be integrated onto the SMD area.

Relay shield

- The Arduino digital I/O pins cannot bear the high current due to its voltage and current limits. The relay shield is used to overcome such situation. It provides a solution for controlling the devices carrying high current and voltage.
- The shield consists of four relays and four LED indicators.
- It also provides NO/NC interfaces and a shield form factor for the simple connection to the Arduino board.
- The LED indicators depicts the ON/OFF condition of each relay.
- The relay used in the structure is of high quality.
- The NO (Normally Open), NC (Normally Closed), and COM pins are present on each relay.
- The relay shield is shown below:



- The applications of the Relay shield include remote control, etc.

Motor shield

- The motor shield helps us to control the motor using the Arduino board.
- It controls the direction and working speed of the motor. We can power the motor shield either by the external power supply through the input terminal or directly by the Arduino.
- We can also measure the absorption current of each motor with the help of the motor shield.
- The motor shield is based on the L298 chip that can drive a step motor or two DC motors. L298 chip is a full bridge IC. It also consists of the heat sinker, which increases the performance of the motor shield.

- It can drive inductive loads, such as solenoids, etc.
- The operating voltage is from 5V to 12V.

The Motor shield is shown below:



- The applications of the motor shield are intelligent vehicles, micro-robots, etc.

LCD shield

- The keypad of LCD (Liquid Crystal Display) shield includes five buttons called as up, down, left, right, and select.
- There are 6 push buttons present on the shield that can be used as a custom menu control panel.
- It consists of the 1602 white characters, which are displayed on the blue backlight LCD.
- The LED present on the board indicates the power ON.
- The five keys present on the board helps us to make the selection on menus and from board to our project.

The LCD shield is shown below:



- The LCD shield is popularly designed for the classic boards such as Duemilanove, UNO, etc.

Bluetooth shield

- The Bluetooth shield can be used as a wireless module for transparent serial communication.
- It includes a serial Bluetooth module. D0 and D1 are the serial hardware ports in the Bluetooth shield, which can be used to communicate with the two serial ports (from D0 to D7) of the Arduino board.
- We can install Groves through the two serial ports of the Bluetooth shield called a Grove connector. One Grove connector is digital, while the other is analog.

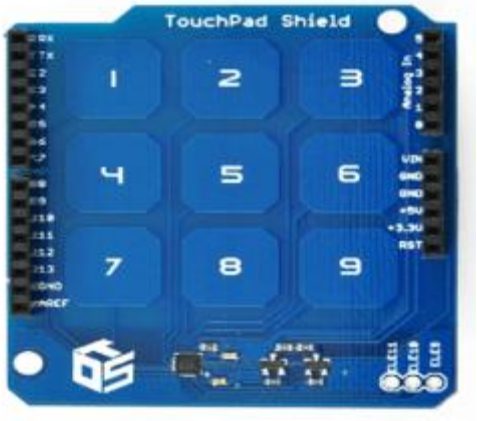
The Bluetooth shield is shown below:



- The communication distance of the Bluetooth shield is upto 10m at home without any obstacle in between.

Capacitive Touchpad shield

- It has a touchpad interface that allows to integrate the Arduino board with the touch shield.
- The Capacitive touchpad shield consists of 12 sensitive touch buttons, which includes 3 electrode connections and 9 capacitive touch pads.
- The capacitive shield is shown below:



- The board can work with the logic level of 3.3V or 5V.
- We can establish a connection to the Arduino project by touching the shield.

1.7 INTEGRATION OF SENSORS AND ACTUATORS WITH ARDUINO

Arduino is the open-source platform used for making custom electronics projects. Before learning about the Arduino Sensors, lets first understand the sensors as a general term.

What are the sensors?

The sensors are defined as a machine, module, or a device that detect changes in the environment. The sensors transfer those changes to the electronic devices in the form of a signal.

A sensor and electronic devices always work together. The output signal is easily readable by humans.

Nowadays, Sensors are used in daily lives. For example, controlling the brightness of the lamp by touching its base, etc. The use of sensors is expanding with new technologies.

What is a sensor made of?

The sensor is a device, which is made up of **Single Crystal Silicon**. It is considered as a widely used semiconductor material. It has superior mechanical stability, machinability, etc. It can also combine electronics and sensing elements on the same substrate.

Where are the sensors used?

The sensors are used to measure the physical quantities, such as pressure, temperature, sound, humidity, and light, etc.

An example of sensors is Fire Alarm, a detector present on the fire alarm detects the smoke or heat. The signal generated from the detector is sent to the alarming system, which produces an alert in the form of alarm.

The types of detectors are smoke detectors, heat detectors, carbon monoxide detectors, multi-sensors detectors, etc.

How are the sensors used in Arduino?

The data signal runs from the sensor to the output pins of the Arduino. The data is further recorded by the Arduino.

What are the types of sensors in Arduino?

Some of the types of sensors in Arduino are listed below:

- **Light sensor**

The light sensor is used to control the light. It is used with LDR (Light Dependent Resistor) in Arduino.

- **Ultrasonic sensor**

The ultrasonic sensor is used to determine the distance of the object using SONAR.

- **Temperature sensor**

The temperature sensor is used to detect the temperature around it.

- **Knock Sensor**

The knock sensor is used to pick the vibrations of the knocking. It is a common category of a vibration sensor.

- **Object Detection Sensor**

It is used to detect the object by emitting infrared radiations, which are reflected or bounced back by that object.

- **Tracking Sensor**

It allows the robots to follow a particular path specified by sensing the marking or lines on the surface.

- **Metal Touch Sensor**

It is suitable for detecting the human touch.

- **Water Level Sensor**

It is used to measure the water or the depth of the water level. It is also used to detect leaks in containers.

- **Vibration Sensor**

The vibration sensor is used to measure the vibrations.

- **Air Pressure sensor**

It is commonly related to meteorology, biomedical fields. It looks like the below image:



- **Pulse Sensor**

The pulse sensor is used to measure the pulse rate. It looks like the below image:



- **Capacitive soil moisture sensor**

It is used to measure the moisture level of the soil.

- **Microphone sensor**

The microphone sensor in Arduino is used to detect the sound. The **analog** and **digital** are the two outputs of this module. The digital output sends the high signal when the intensity of sound reaches a certain threshold. We can adjust the sensitivity of a module with the help of a potentiometer.

- **humidity sensor**

The humidity sensor is used to monitor weather conditions.

- **Motion sensor**

The motion sensor detects the movement and occupancy from the human body with the help of Infrared radiation.

- **Vibration sensor**

The vibration sensor is used to detect the vibrations.

- **Sound sensor**

The sound sensor is suitable to detect the sound of the environment.

- **Pressure Sensor**

The pressure sensor is used to measure the pressure. The sensor in Arduino measures the pressure and displays it on the small LCD screen.

- **Magnetic field sensor**

The magnetic field sensor measures the magnetic field strength and produces a varying voltage as the output in Arduino.