

## **Practical – 1**

**Aim: -** Familiarization with Arduino/Raspberry Pi and perform necessary software installation.

**Theory: -**

### **What is Arduino?**

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming. As soon as it reached a wider community, the Arduino board started changing to adapt to new needs and challenges, differentiating its offer from simple 8-bit boards to products for IoT applications, wearable, 3D printing, and embedded environments. All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software, too, is open-source, and it is growing through the contributions of users worldwide.

### **Why Arduino?**

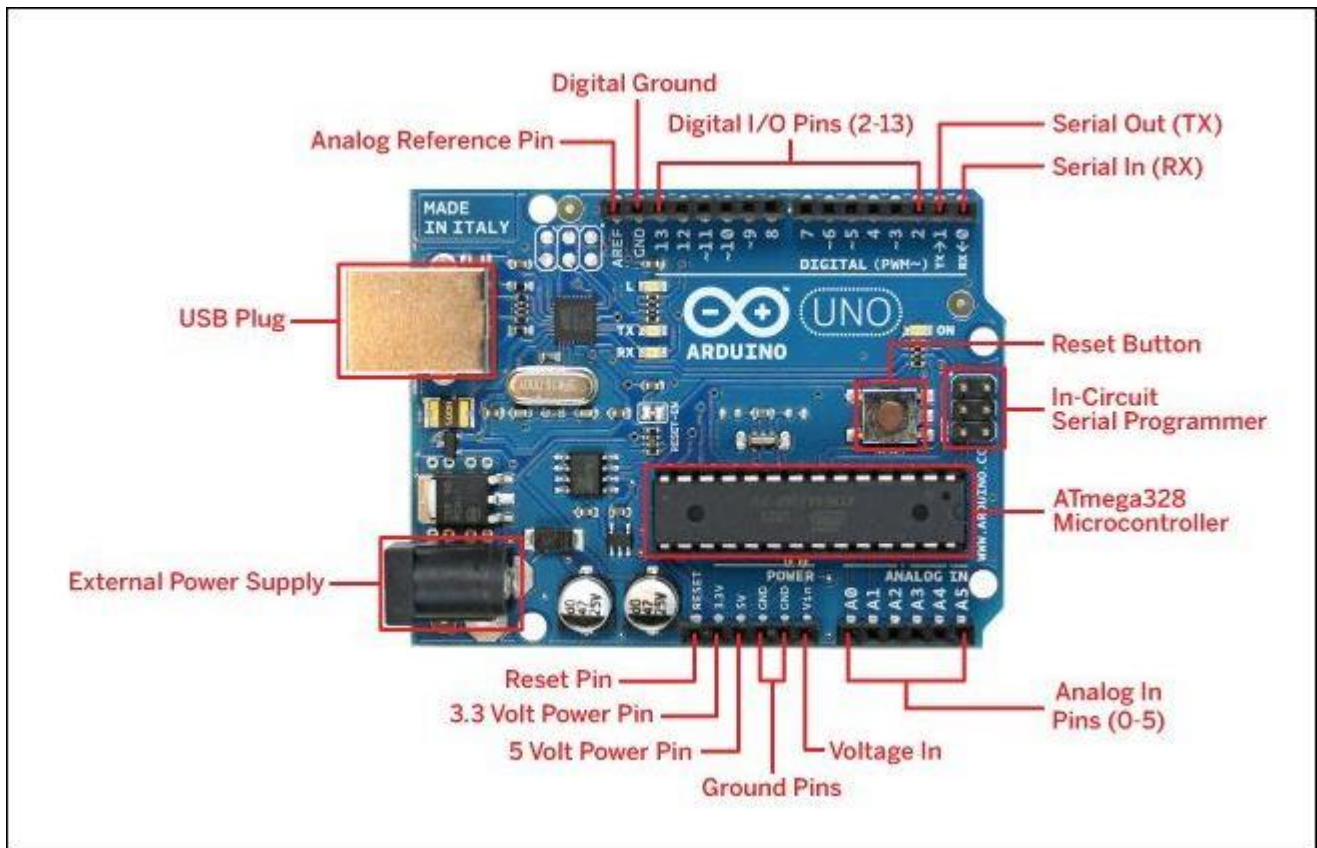
Thanks to its simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. It runs on Mac, Windows, and Linux. Teachers and students use it to build low cost scientific instruments, to prove chemistry and physics principles, or to get started with programming and robotics. Designers and architects build interactive prototypes, musicians and artists use it for installations and to experiment with new musical instruments. Makers, of course, use it to build many of the projects exhibited at the Maker Faire, for example. Arduino is a key tool to learn new things. Anyone - children, hobbyists, artists, programmers - can start tinkering just following the step by step instructions of a kit, or sharing ideas online with other members of the Arduino community.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- **Inexpensive** - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- **Cross-platform** - The Arduino Software (IDE) runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- **Simple, clear programming environment** - The Arduino Software (IDE) is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with how the Arduino IDE works.
- **Open source and extensible software** - The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.
- **Open source and extensible hardware** - The plans of the Arduino boards are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can build the breadboard version of the module in order to understand how it works and save money.

### **Arduino Uno R3 Pin Diagram: -**

The **Arduino Uno R3 pin diagram** is shown below. It comprises 14-digit I/O pins. From these pins, 6-pins can be utilized like PWM outputs. This board includes 14 digital input/output pins, Analog inputs-6, a USB connection, quartz crystal-16 MHz, a power jack, a USB connection, resonator-16Mhz, a power jack, an ICSP header an RST button.



Arduino Uno Pin Diagram

## Power Supply

The power supply of the Arduino can be done with the help of an exterior power supply otherwise USB connection. The exterior power supply (6 to 20 volts) mainly includes a battery or an AC to DC adapter. The connection of an adapter can be done by plugging a center-positive plug (2.1mm) into the power jack on the board. The battery terminals can be placed in the pins of Vin as well as GND. The power pins of an **Arduino board** include the following.

**Vin:** The input voltage or Vin to the Arduino while it is using an exterior power supply opposite to volts from the connection of USB or else **RPS (regulated power supply)**. By using this pin, one can supply the voltage.

**5Volts:** The RPS can be used to give the power supply to the microcontroller as well as components which are used on the Arduino board. This can approach from the input voltage through a regulator.

**3V3:** A 3.3 supply voltage can be generated with the onboard regulator, and the highest draw current will be 50 mA.

**GND:** GND (ground) pins

## Memory

The memory of an ATmega328 microcontroller includes 32 KB and 0.5 KB memory is utilized for the Boot loader), and also it includes SRAM-2 KB as well as EEPROM-1KB.

## **Input and Output**

We know that an Arduino Uno R3 includes 14-digital pins which can be used as an input otherwise output by using the functions like pin Mode (), digital Read(), and digital Write(). These pins can operate with 5V, and every digital pin can give or receive 20mA, & includes a 20k to 50k ohm [pull up resistor](#). The maximum current on any pin is 40mA which cannot surpass for avoiding the microcontroller from the damage. Additionally, some of the pins of an Arduino include specific functions.

## **Serial Pins**

The serial pins of an Arduino board are TX (1) and RX (0) pins and these pins can be used to transfer the TTL serial data. The connection of these pins can be done with the equivalent pins of the ATmega8 U2 USB to TTL chip.

## **External Interrupt Pins**

The external interrupt pins of the board are 2 & 3, and these pins can be arranged to activate an interrupt on a rising otherwise falling edge, a low-value otherwise a modify in value

## **PWM Pins**

The PWM pins of an Arduino are 3, 5, 6, 9, 10, & 11, and gives an output of an 8-bit PWM with the function analog Write ().

## **SPI (Serial Peripheral Interface) Pins**

The SPI pins are 10, 11, 12, 13 namely SS, MOSI, MISO, SCK, and these will maintain the **SPI communication** with the help of the SPI library.

## **LED Pin**

An Arduino board is inbuilt with a LED using digital pin-13. Whenever the digital pin is high, the LED will glow otherwise it will not glow.

## **TWI (2-Wire Interface) Pins**

The TWI pins are SDA or A4, & SCL or A5, which can support the communication of TWI with the help of Wire library.

## **AREF (Analog Reference) Pin**

An analog reference pin is the reference voltage to the inputs of an analog I/Ps using the function like analog Reference ().

## **Reset (RST) Pin**

This pin brings a low line for resetting the microcontroller, and it is very useful for using an RST button toward shields which can block the one over the Arduino R3 board.

## **Communication**

The communication protocols of an Arduino Uno include SPI, I2C, and **UART serial communication**.

## **UART**

An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.

## **I2C**

An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.

## **SPI Pins**

The SPI communication includes MOSI, MISO, and SCK.

### **MOSI (Pin11)**

This is the master out slave in the pin, used to transmit the data to the devices

### **MISO (Pin12)**

This pin is a serial CLK, and the CLK pulse will synchronize the transmission of which is produced by the master.

### **SCK (Pin13)**

The CLK pulse synchronizes data transmission that is generated by the master. Equivalent pins with the SPI library is employed for the communication of SPI. ICSP (in-circuit serial programming) headers can be utilized for programming **ATmega microcontroller** directly with the boot loader.

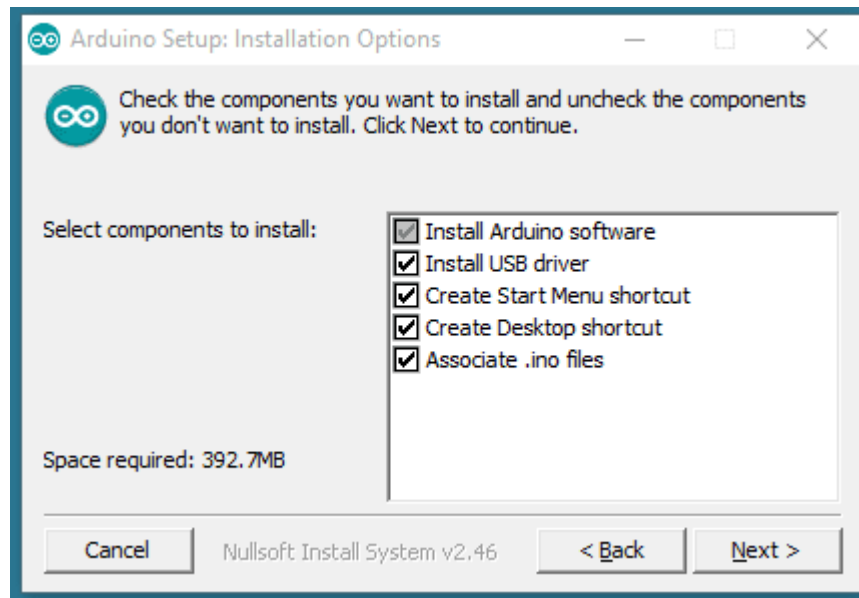
## **Installation of Arduino: -**

### **Install the Arduino Software (IDE) on Windows PCs: -**

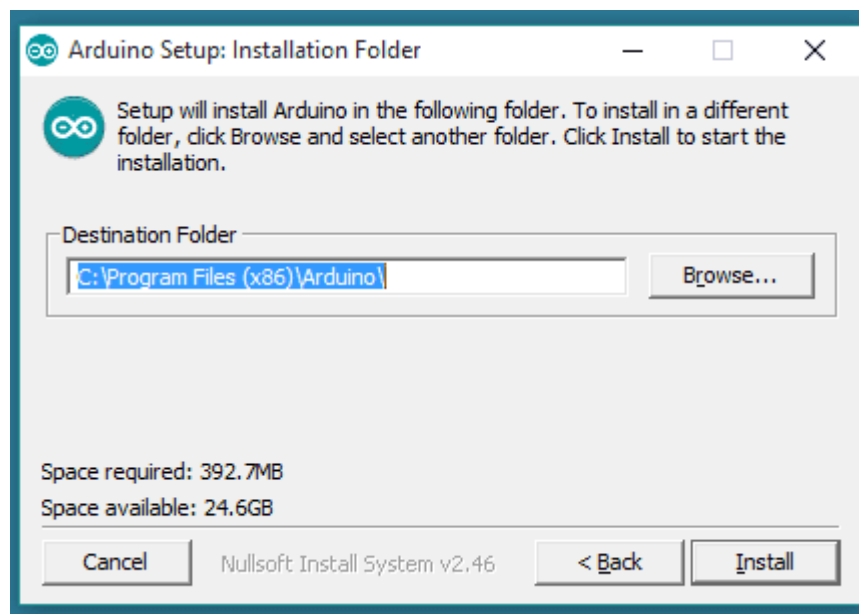
#### **Download the Arduino Software (IDE)**

Get the latest version from the download page. You can choose between the Installer (.exe) and the Zip packages. We suggest you use the first one that installs directly everything you need to use the Arduino Software (IDE), including the drivers. With the Zip package you need to install the drivers manually. The Zip file is also useful if you want to create a portable installation.

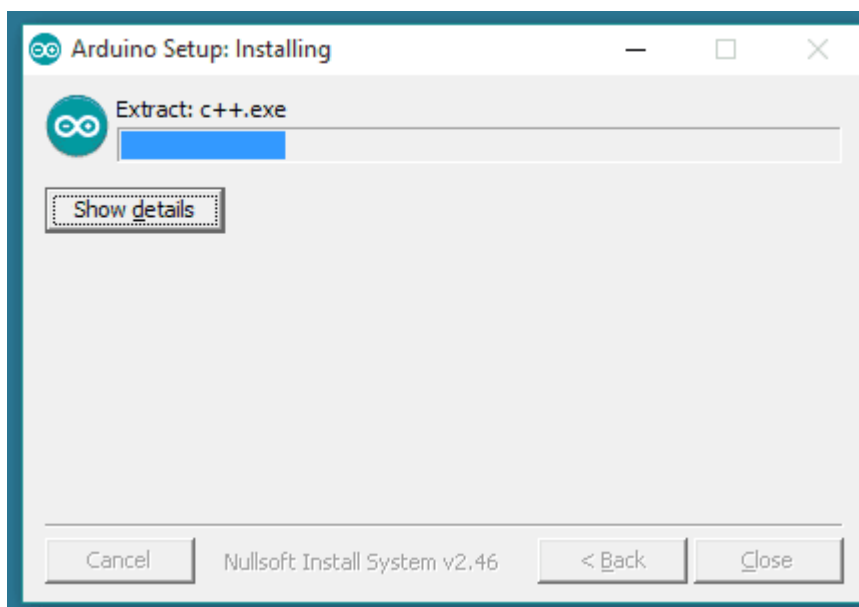
When the download finishes, proceed with the installation and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install



Choose the installation directory (we suggest to keep the default one)



The process will extract and install all the required files to execute properly the Arduino Software (IDE)

### **Proceed with board specific instructions**

When the Arduino Software (IDE) is properly installed you can go back to the Getting Started Home and choose your board from the list on the right of the page.