LAB MANUAL 2

Getting Started with Arduino

Disclaimer: The content is curated from online/offline resources and used for educational purpose only

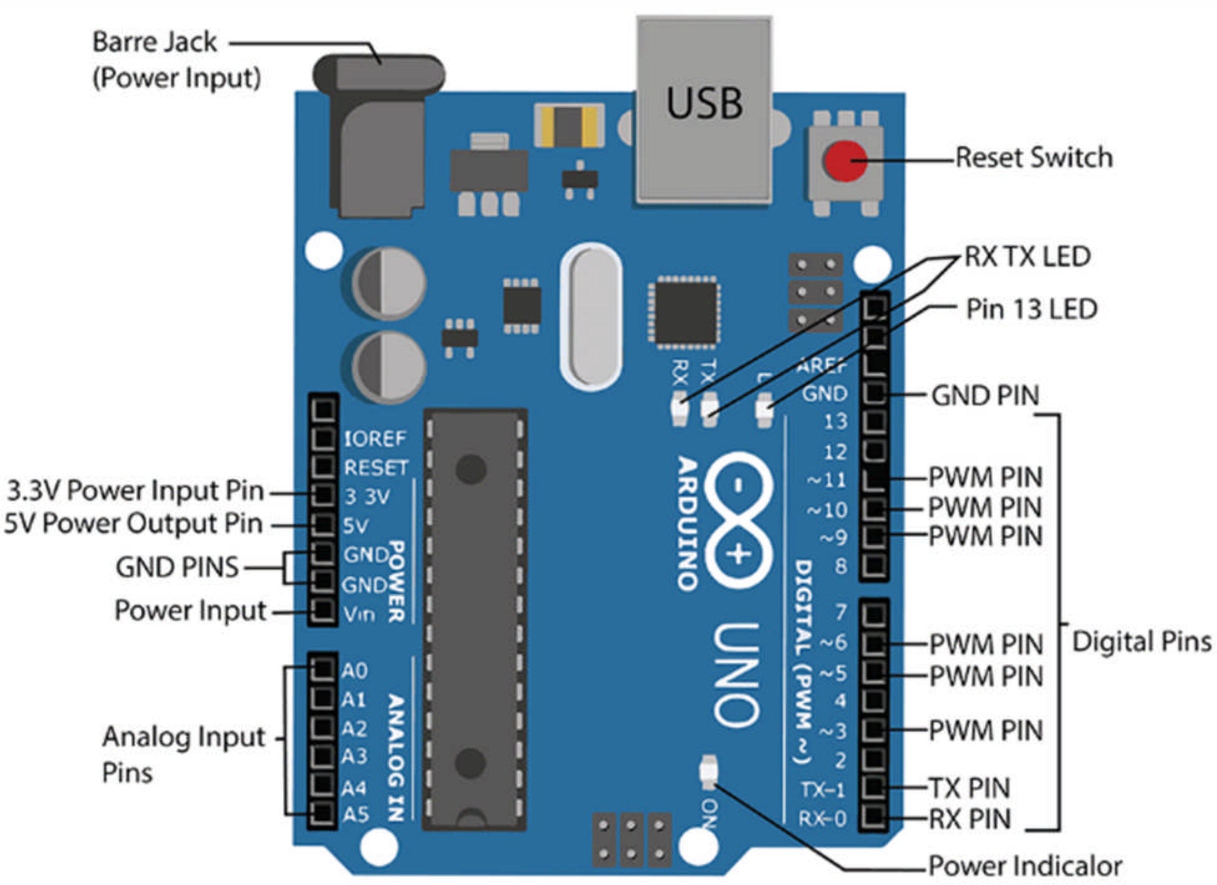
About Arduino board-

A microcontroller is embedded inside of a system to control a singular function in a device. It does this by interpreting data it receives from its I/O peripherals using its central processor. The temporary information that the microcontroller receives is stored in its data memory, where the processor accesses it and uses instructions stored in its program memory to decipher and apply the incoming data. It then uses its I/O peripherals to communicate and enact the appropriate action.

Arduino boards are available in different sizes, form factors, different no. of I/O pins etc. Some of the commonly known and frequently used Arduino boards are Arduino UNO, Arduino Mega, Arduino Nano, Arduino Mini, Arduino Leonardo and Arduino Lilypad.

Arduino UNO is a basic and inexpensive Arduino board and is the most popular of all the Arduino boards with a market share of over 50%. Arduino UNO is considered the best prototyping board for beginners in electronics and coding.

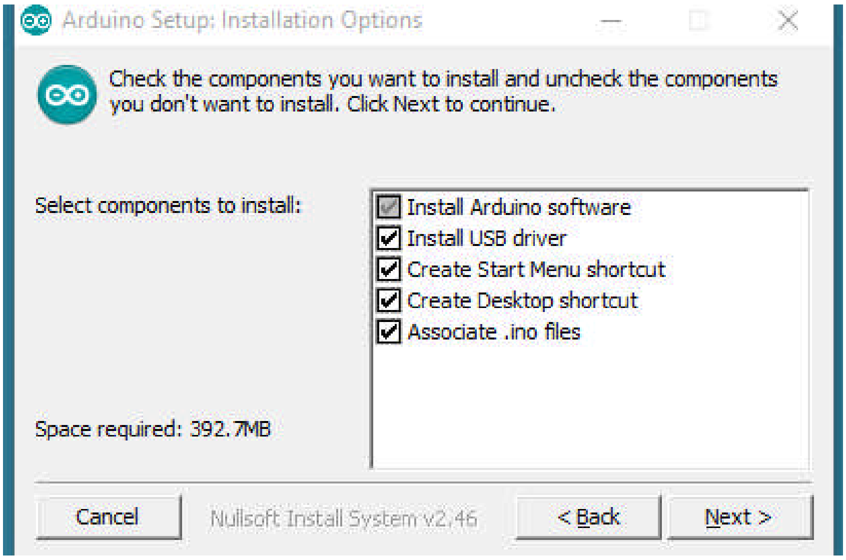
- UNO is based on ATmega328P microcontroller, which is an 8-bit microcontroller based on the AVR architecture.
- It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a USB connection, a power jack, a reset button and more. It contains everything needed to support 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.



Step-1

Download Arduino Software – Integrated Development Environment

Access the Internet – In order to use your Arduino, you will first need to download Arduino software from the Internet (it is free!). This software, known as the Arduino integrated development environment (IDE), allows us to program Arduino to do what we want. It is called integrated, because we can write the code, compile and execute it from the same program window. You can download the latest or a previous version of the software from Arduino.cc, the repository for Arduino development. We can also code online using Arduino Web Editor, with the code stored on the cloud: http://arduino.cc/en/Main/Software



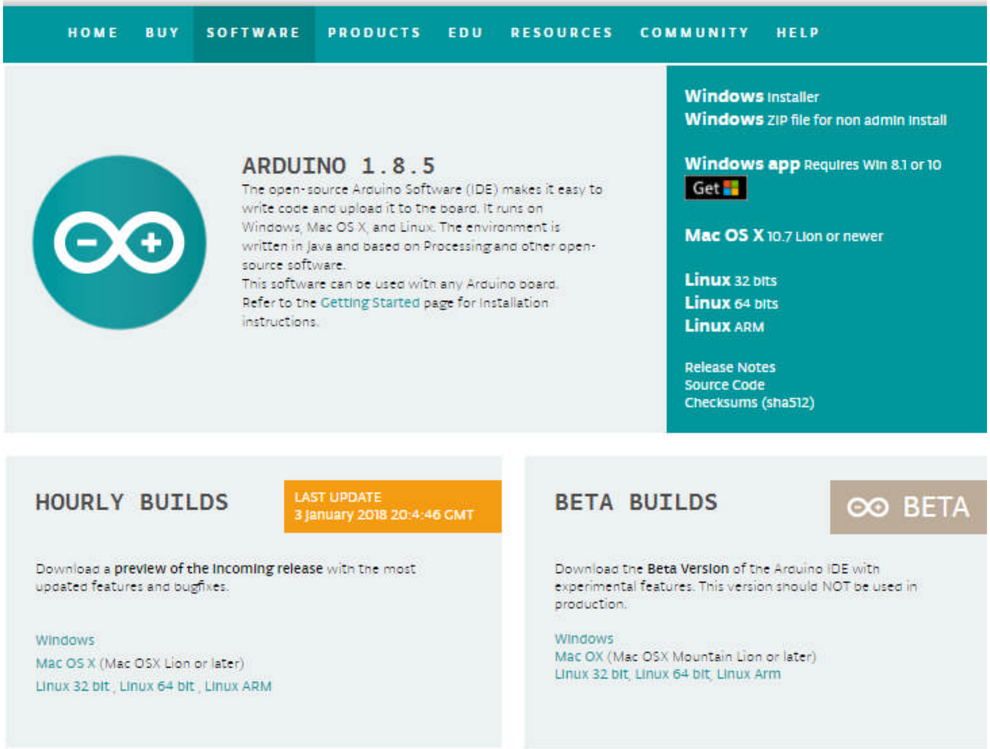
Arduino Setup: Installation Options

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

☑ Install Arduino software
☑ Install USB driver
☑ Create Start Menu shortcut
☑ Create Desktop shortcut
☑ Associate .ino files

Space required: 392.7MB



ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for installation instructions.

Windows Installer
Windows ZIP file for non admin install

Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

HOURLY BUILDS

LAST UPDATE
3 January 2018 20:4:46 GMT

Download a **preview of the incoming release** with the most updated features and bugfixes.

Windows
Mac OS X (Mac OSX Lion or later)
Linux 32 bit , Linux 64 bit , Linux ARM

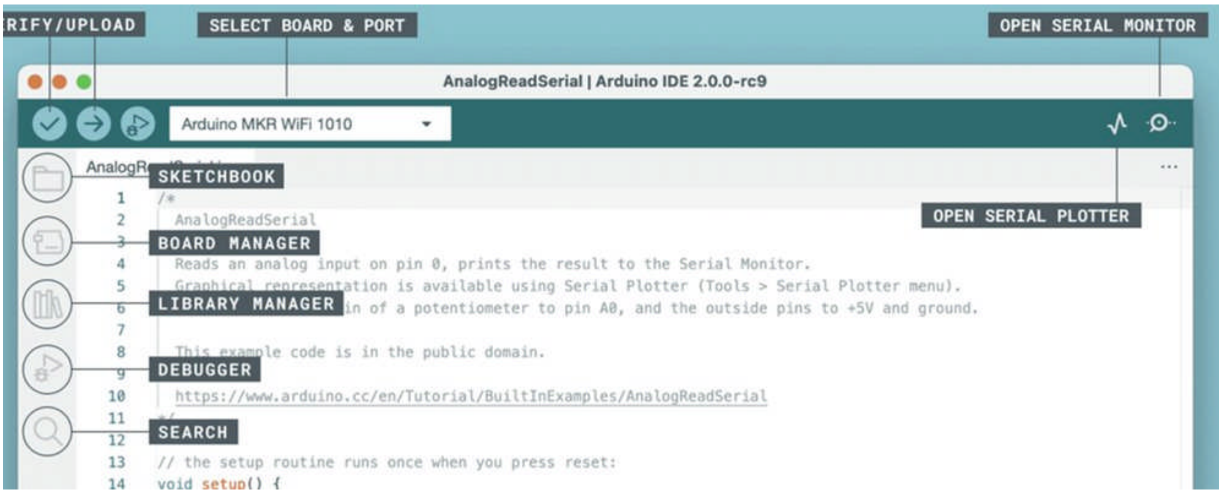
BETA BUILDS

BETA

Download the **Beta Version** of the Arduino IDE with experimental features. This version should NOT be used in production.

Windows
Mac OX (Mac OSX Mountain Lion or later)
Linux 32 bit, Linux 64 bit, Linux Arm

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.



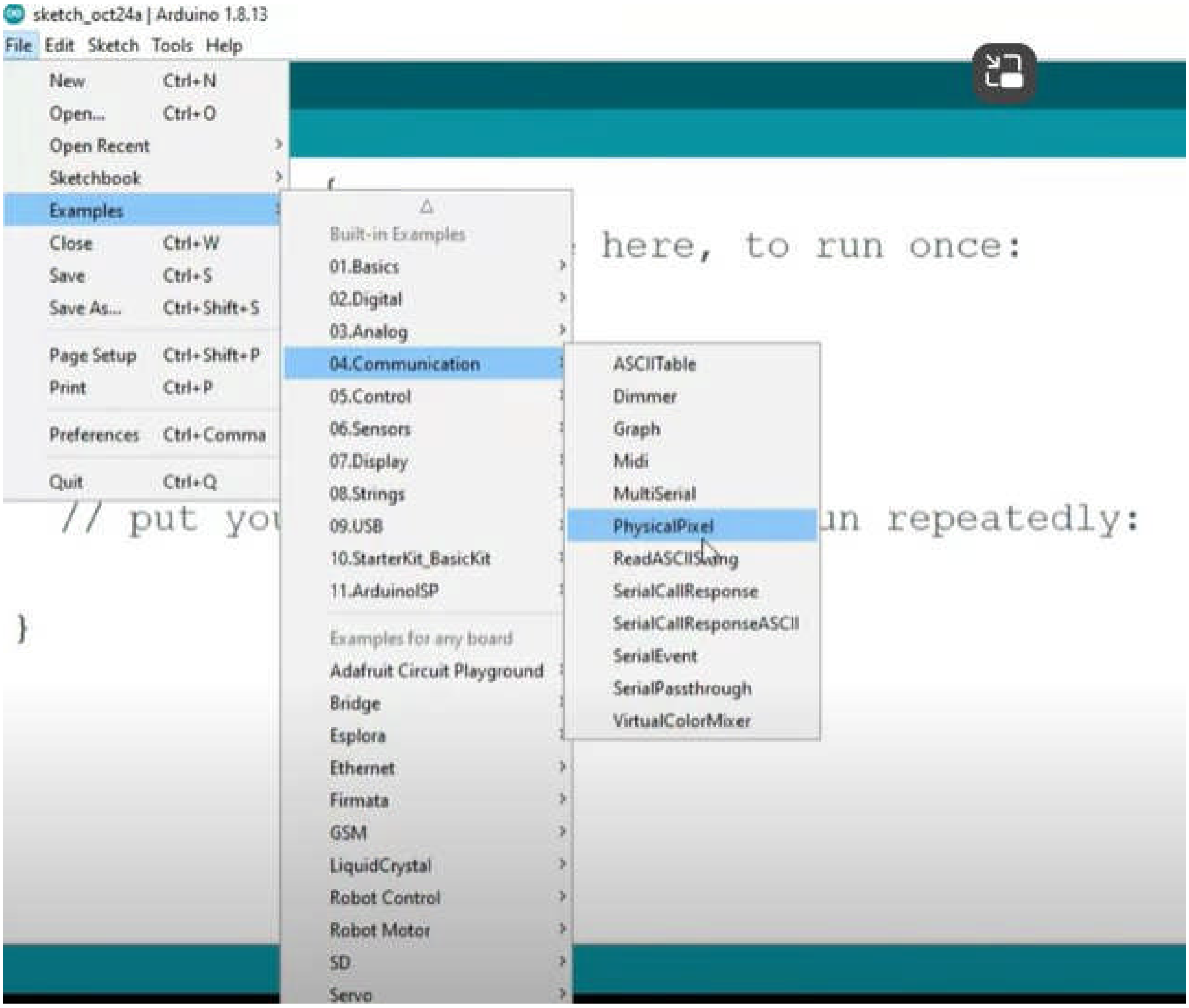
AnalogReadSerial | Arduino IDE 2.0.0-rc9

```
/*
  AnalogReadSerial

  Reads an analog input on pin 0, prints the result to the Serial Monitor.
  Graphical representation is available using Serial Plotter (Tools > Serial Plotter menu).
                  in of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.

  https://www.arduino.cc/en/Tutorial/BuiltInExamples/AnalogReadSerial
*/

// the setup routine runs once when you press reset:
void setup() {
```

Step-2

Procedure for practical setup-

- Check out: http://arduino.cc/en/Guide/HomePage
- Download & install the Arduino environment (IDE)
- Connect the board to your computer via the UBS cable
- If needed, install the drivers (not needed in lab)
- Launch the Arduino IDE
- Select your board
- Select your serial port
- Open the examples select communication and then physicalPixcel and run code
- Open our window terminal and follow the steps in the next slide.



Procedure for Port working verification :

Paste code in Arduino ide-

```
void setup() {
  // Start the serial communication with a baud rate of 9600
  Serial.begin(9600);

  // Give some time for the Serial Monitor to start
  delay(1000);

  // Print a message to the Serial Monitor
  Serial.println("Arduino is connected and working properly.");
}
```

```
void loop() {
  // Print a message every second
  Serial.println("Arduino is still connected.");
  delay(1000);
}
```

Steps to Upload and Verify:

1. **Open the Arduino IDE**:
   - Launch the Arduino IDE on your computer.
2. **Select Your Board**:
   - Go to Tools -> Board and select the appropriate Arduino board you are using (e.g., Arduino Uno).
3. **Select Your Port**:
   - Go to Tools -> Port and select the correct port to which your Arduino is connected. This is usually something like COM3 on Windows or /dev/ttyUSB0 on Linux.
4. **Upload the Code**:
   - Copy the verification code above into the Arduino IDE.
   - Click the upload button (right arrow icon) in the Arduino IDE to upload the sketch to your board.
5. **Open the Serial Monitor**:
   - After uploading the code, open the Serial Monitor by clicking the magnifying glass icon in the top right corner of the Arduino IDE or by going to Tools -> Serial Monitor.
6. **Check Output**:
   - Set the baud rate in the Serial Monitor to 9600.
   - If the Arduino is correctly connected, you should see the messages "Arduino is connected and working properly." and "Arduino is still connected." being printed every second.