**Embedded System?**

So, you guys have some doubts and most of you people have answer that embedded system is any system which has one microprocessor/microcontroller which is run by some sort of software to control hardware, **but**! can we control hardware using our laptop or mobile like our overly enhanced cameras, answer is yes but do laptop and mobile comes under definition of embedded system?

**Definition:** It is a microcontroller-based system that is designed to control a function or range of functions, and is not meant to be programmed by the end user.



**Ahh Wait, tell us what embedded system actually means!**

**So now let us focus on what embedded system is and what it is not!**

* A microprocessor sitting inside a traditional computing system (like desktop, laptop, server)
* Any kind of supercomputers.
* Even above given points too held no meaning if we are controlling any home appliances using website on laptop or through mobile app.
* So now our laptop and mobile may act as application specific.



**Now what?**

We then narrow down to a point that embedded system are application specific and they are not manufactured for general purpose.

**For example**

* We have one gaming pc consisting better graphics card like Nvidia 1080ti graphics card, whooping 1tb of solid-state drive, 16gb of ram, intel processor i8 etc.
* So now this system is made for complex programs, animation, high end gaming, better experience in all sense, and thus it will cost handsomely.
* Eventually, we are not going to control Diwali led strips with above given system, instead we will run high end software like fusion360, play GTA-5 etc.

**So, above system Is built for general purpose!**

What embedded system actually are they are meant to be application specific

**My imagination of embedded system limits me to given below examples:**

a) **Consumer Segment**: Refrigerator, washing machine, A/C machine, camera, microwave oven, TV, security system, etc.

b) **Office Automation**: Printers, Fax machines, photocopying machines, scanners, biometric, scanner, surveillance camera, etc.

c) **Automobiles**: Air bags, anti-lock braking system (ABS), engine control, door lock, GPS system, vehicular ad-hoc network (VANET), etc.

d) **Communication**: Mobile phones, network switches, Wi-Fi hotspots, telephones,

MODEM, etc.

e) **Miscellaneous**: Automatic door locks, automatic baggage screening,

surveillance systems, intelligent toilet, etc.

Microcontrollers

Microcontroller is part and parcel of embedded system

Let us check what is available in market

1. Intel 8051:

The Intel MCS-51 (commonly termed 8051) is a single chip microcontroller (MCU) series developed by Intel in 1980 for use in embedded systems. Intel’s original MCS-51 family was developed using N-type metal-oxide-semiconductor (NMOS).

1. PIC series

PIC is a family of microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division. The name PIC initially referred to Peripheral Interface Controller, then it was corrected as Programmable Intelligent Computer.

1. AVR MICROCONTROLLERS:

AVR is a family of microcontrollers developed since 1996 by Atmel, acquired by Microchip Technology in 2016. These are modified Harvard architecture 8-bit RISC single-chip microcontrollers. AVR was one of the first microcontroller families to use on-chip flash memory for program storage, as opposed to one-time programmable ROM, EPROM, or EEPROM used by other microcontrollers at the time.

Why Arduino?

More and more makerspaces around the world are looking to add coding and electronics to their maker education programs.  One of the best ways to do this is by integrating an Arduino board into makerspace projects and lessons.

We’ve found that a lot of maker educators haven’t taken the plunge into coding or Arduino because they think programming is scary.  Because of this, we wanted to make sure this tutorial was written for the absolute beginner with no experience whatsoever.

This tutorial is a high level view of all the parts and pieces of the Arduino ecosystem.  In future posts, we will take you step by step in creating your first simple Arduino project.

What is Arduino?

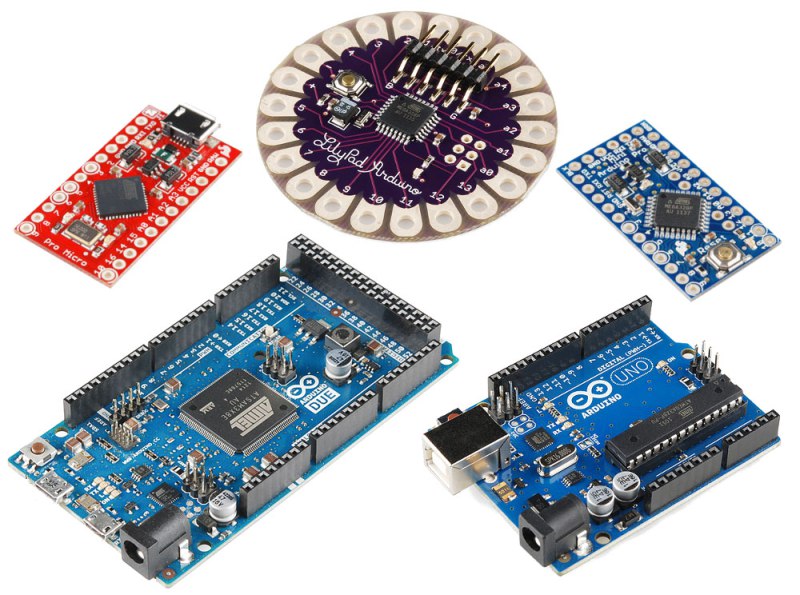
Arduino is an open source programmable circuit board that can be integrated into a wide variety of makerspace projects both simple and complex.  This board contains a [microcontroller](https://en.wikipedia.org/wiki/Microcontroller) which is able to be programmed to sense and control objects in the physical world.   By responding to sensors and inputs, the Arduino is able to interact with a large array of outputs such as LEDs, motors and displays.  Because of it’s flexibility and low cost, [Arduino](https://www.arduino.cc/) has become a very popular choice for makers and makerspaces looking to create interactive hardware projects.

Arduino was introduced back in 2005 in Italy by Massimo Banzi as a way for non-engineers to have access to a low cost, simple tool for creating hardware projects.  Since the board is [open-source](https://en.wikipedia.org/wiki/Open-source_hardware), it is released under a Creative Commons license which allows anyone to produce their own board.  If you search the web, you will find there are hundreds of Arduino compatible clones and variations available but the only official boards have Arduino in it’s name.

Types Of Arduino?

Arduino is a great platform for prototyping projects and inventions but can be confusing when having to choose the right board.  If you’re brand new to this, you might have always thought that there was just one “Arduino” board and that’s it.  In reality, there are many variations of the official Arduino boards and then there are hundreds more from competitors who offer clones.  But don’t worry, we’re going to show you which one to start with later on in this tutorial.

Below are a few examples of the different types of Arduino boards out there.  The boards with the name Arduino on them are the official boards but there are also a lot of really great clones on the market as well.  One of the best reasons to buy a clone is the fact they are generally less expensive than their official counterpart.  [Adafruit](https://www.adafruit.com/) and [Sparkfun](https://www.sparkfun.com/" \t "_blank) for example, sell variations of the Arduino boards which cost less but still have the same quality of the originals.  One word of caution, be careful when buying boards from companies you don’t know.



Arduino Uno Pinout?

## Board Breakdown

Here are the components that make up an Arduino board and what each of their functions are.

1. **Reset Button** – This will restart any code that is loaded to the Arduino board
2. **AREF** – Stands for “Analog Reference” and is used to set an external reference voltage
3. **Ground Pin** – There are a few ground pins on the Arduino and they all work the same
4. **Digital Input/Output** –  Pins 0-13 can be used for digital input or output
5. **PWM** – The pins marked with the (~) symbol can simulate analog output
6. **USB Connection** – Used for powering up your Arduino and uploading sketches
7. **TX/RX** – Transmit and receive data indication LEDs
8. **ATmega Microcontroller** –  This is the brains and is where the programs are stored
9. **Power LED Indicator** – This LED lights up anytime the board is plugged in a power source
10. **Voltage Regulator** – This controls the amount of voltage going into the Arduino board
11. **DC Power Barrel Jack** – This is used for powering your Arduino with a power supply
12. **3.3V Pin** – This pin supplies 3.3 volts of power to your projects
13. **5V Pin** – This pin supplies 5 volts of power to your projects
14. **Ground Pins** –  There are a few ground pins on the Arduino and they all work the same
15. **Analog Pins** –  These pins can read the signal from an analog sensor and convert it to digital

**How do I power my Arduino?**

The recommended unit is our [9V 2A Arduino Power Supply](http://www.modmypi.com/arduino/accessories/power-supplies/arduino-universal-power-supply-9v-2a). This charger offers a nice stable supply voltage, and ensures that the Arduino's voltage regulators aren't pushed too hard. This unit should extend the life of your Arduino, and allow you to complete the majority of low-end hacking.

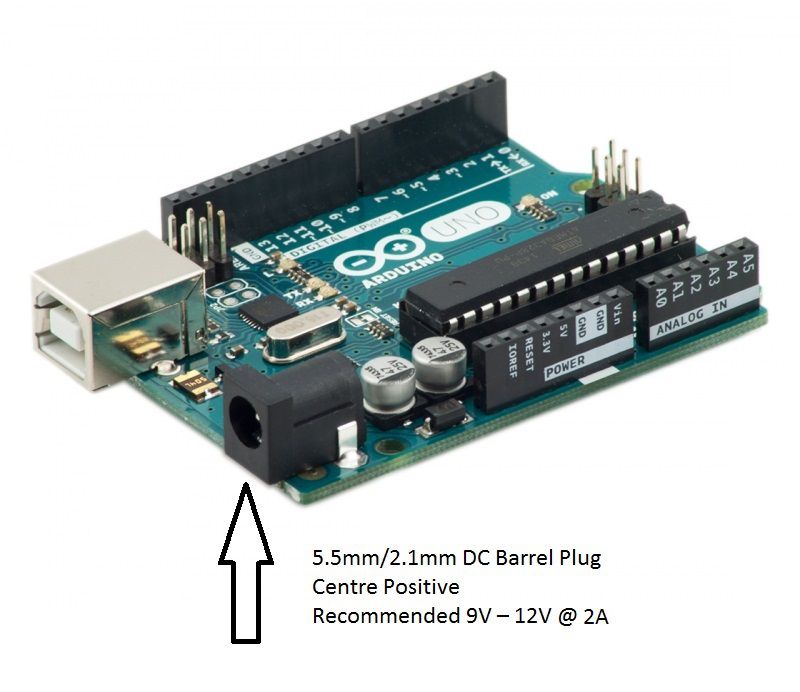
We also offer a [12V 2A Arduino Power Supply](http://www.modmypi.com/arduino/accessories/power-supplies/arduino-universal-power-supply-12v-2a-high-voltage). This unit is for high power applications, or if you’re looking to drive a 12V circuit via the Vin pins. This is the top end of the recommended voltage range, so users are advised that this should only be used if actually required.

**Mode 1 – DC Barrel Plug 5.5mm/2.1mm (Recommended 9V – 12V @ 2A)**

For a more concise explanation of the various requirements see below, but the simplest answer is the following. The Arduino features a DC (Direct Current) barrel jack input port for power. The barrel jack has the following requirements:

* The Adaptor must be DC (Direct Current) not AC (Alternating Current)
* The barrel plug must be center positive (The middle pin of the plug has to be positive)
* The barrel plug must have an inside diameter (ID) of 2.1mm
* The barrel plug must have an outside diameter (OD) of 5.5mm or less.
* The operating voltage is recommended to be between 9V and 12V
* The operating current is recommended to be 0.5A to 2A.
* Barrel connector length of 9.5mm or greater is sufficient.

This is a common size barrel, and you’ll see 5.5mm/2.1mm featured on a lot of our items.



**Operating Voltage**

The Arduino can operate on a supply through this port of 6 to 20V. However, the operative word here is “can”.

If the board is supplied with less than 7V, the unit’s performance could become unstable, and the 5V I/O pin may end up supplying less than 5V, which could cause additional circuitry to function incorrectly.

Vice versa, using an over-voltage power supply up to 20V will cause the regulators on the board to run at full-whack, dissipating the extra voltage as heat. This is both inefficient, and could cause over-heating of the Arduino.

**Therefore, the recommended voltage is 9V to 12V.**

It should be noted. The Vin pin on the Power Pins I/O will copy the voltage input supplied through the power jack, and act as an output of that voltage. In this way, you effectively have a customizable output voltage pin on the Arduino which will replicate the input voltage of your power supply. For example, if you have a 9V relay, you could run this directly from your Arduino Vin pin as long as you have a 9V barrel power supply connected.

**Operating Current**

The minimum requirement is 250mA which should just about run an Arduino with some measure of stability. However, if you want to power any external devices (Servo’s, LED’s, USB devices etc.) then 0.5A to 2A will enable a greater range of use. A charger with a higher current rating will ensure you enough juice to make each component of the circuit function correctly.

**Mode 2 – Through the USB Port (5V @ 500mA)**

The Arduino can also be powered via the USB port on the front of the unit (known as back-powering in the Pi world!). USB powering should only be attempted using a stable 5V supply. It should also be noted that there is a current limit on the USB ports of 500mA, so any current draw greater than this might cause instability.

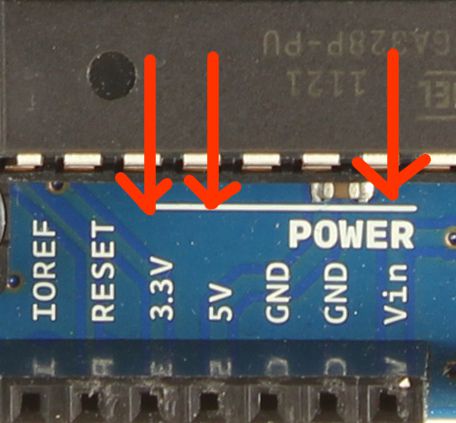
Supplying a voltage to the USB ports essentially dumps that voltage directly to the 5V rail on the Arduino (directly powering the 5V pin). Care should be taken when powering in this manner, as supplying voltage to the USB ports bypasses the 5V regulators, which can damage your board if an incorrect voltage is applied!

Additionally, the Arduino features a resettable polyfuse on the USB ports, with an overcurrent protection of 500mA. This somewhat limits the applications that can be utilized when powering via USB

If the Arduino detects that there is a barrel source and USB source supplying voltage at the same time, the Arduino automatically switches to the barrel source as long as it supplies a sufficient voltage (over 6.6V). Therefore, as long as a barrel supply is connected (and of sufficient voltage) the user does not need to worry about the applied USB voltage.

**Mode 3 – Via the I/O**

The Arduino features a number of power pins as shown below. These function as either inputs or outputs depending on how the Arduino is being powered!



The power pins are as follows:

* Vin. Vin can be utilized as a voltage input (instead of using the barrel adaptor or USB). The voltage should within the 9V – 12V, and it is regulated internally by the board to 5V. Vin can also be used as a voltage output, copying the voltage supplied via the barrel adaptor or USB.
* 5V. This pin outputs a regulated 5V from the regulator on the board. This pin can be supplied with power either from the barrel adaptor, the USB connector, or the Vin pin of the board. You can supply voltage via the 5V pin, however this bypasses the regulator, and can damage your board. Maximum current draw is 0.8A.
* 3V3. A 3.3V output supply generated by the on-board regulator. This pin can be supplied with power either from the barrel adaptor, the USB connector, or the Vin pin of the board. Maximum current draw is 150mA.
* GND. Ground pins.

You can utilize a range of supplies to power the Arduino via the I/O e.g. leads from a battery, external circuit, spliced power supply etc. As Vin is regulated, it’s safe to power using this method, although you lose the additional functionality gained from using Vin as an output if this is required.

Link

[https://youtu.be/tCaiPQxt3vM](https://youtu.be/tCaiPQxt3vM" \t "_blank)