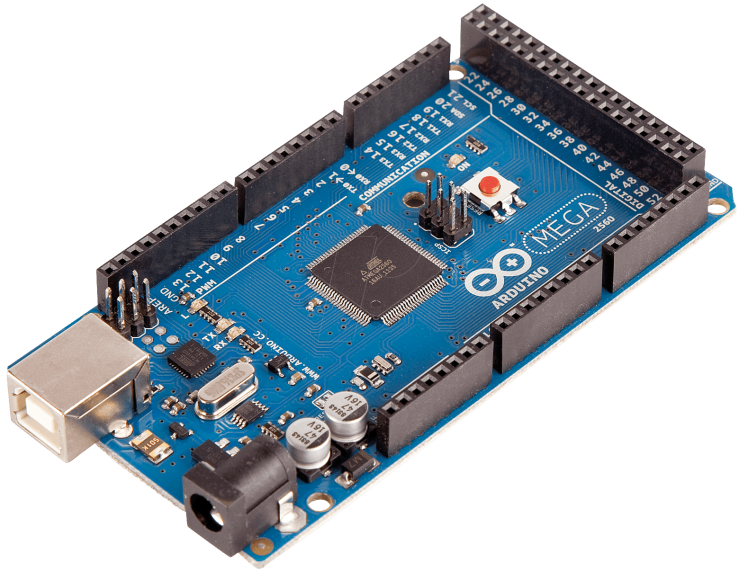
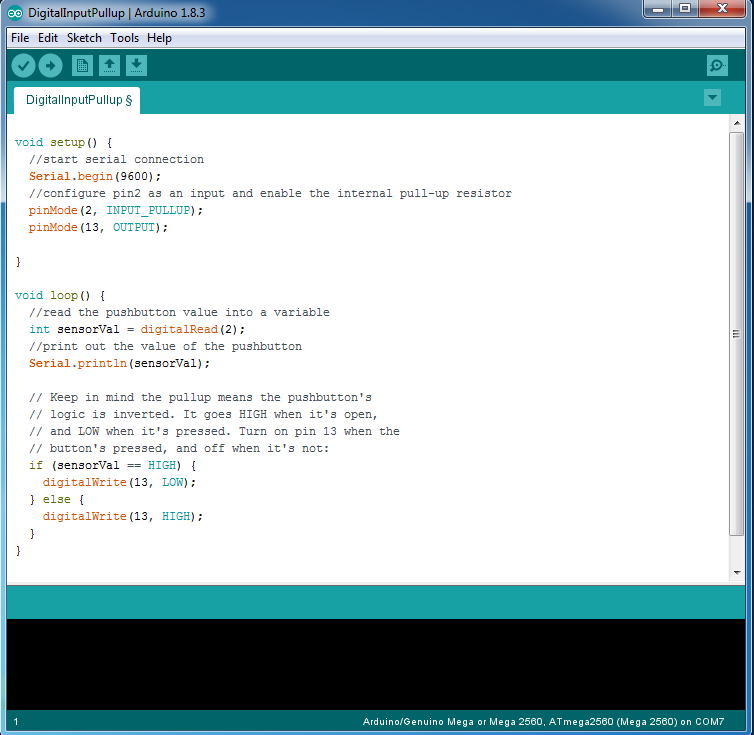
ARDUINO LESSON 1A:

Introduction: What is an Arduino?



*fig 1. Arduino MEGA 2560. The Micro Controller is the square black chip in the middle and is the brains of the device.*

Arduino is an extremely versatile open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board containing a Microcontroller (fig1 above) and a piece of software, or IDE (Integrated Development Environment) which is used to write and upload computer code to the physical board (fig 2 below). By attaching various electronic components and then programming the Arduino we can build an endless variety of circuits that are capable of accomplishing amazing tasks.



*fig 2. The Arduino IDE (Integrated Development Environment) Main screen*



*Arduino powered Quadcopter with GPS and video*

What makes The Arduino system so successful is that unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board – you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of a very common computer programming language C++, making it easier to learn to program. Finally, Arduino provides a standard form factor (shape) that breaks out the functions of the micro-controller into a more accessible package.

What is a Microcontroller?

The best way to explain what a microcontroller is, is to compare it with your computer. Your desktop or laptop computer is made up of multiple parts with the main ones being a CPU, RAM, a hard drive, a keyboard and mouse and a monitor screen. The first step in running a computer program is to take the program information that is stored on the hard drive and copy it to the RAM (RAM and hard drives are both memory but RAM is much faster and can supply the CPU with data much more efficiently) The CPU, which is the brains of the computer then takes program data from the RAM and processes it according to the programs instructions. This "processed" information is then output to the required hardware such as a monitor or printer. To put it simply, the basic function of a computer is to input information, process the information, and then output the new information. A Microcontroller does exactly the same thing but usually on a simpler scale.

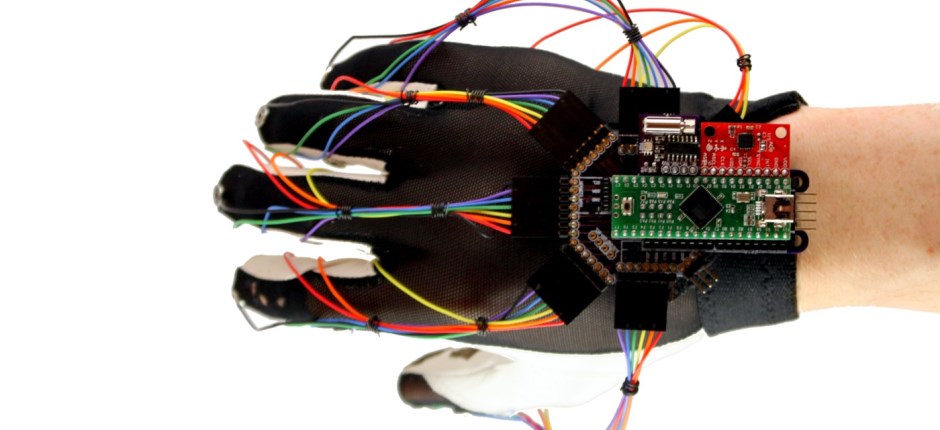
The Arduino microcontroller chip has similar components to a computer such as a CPU, RAM and some long term memory, but they are all built into one chip. Microcontrollers tend to be slower than modern computers, for example the computer you are using right now has a 64-bit CPU running at 3-4 GHz, with 16GB of RAM and 2-4 TB of storage. The Arduino that you will use are 8 bit, run at 16MHz, have 2KB of RAM and 32KB of storage. However, their small size, lower power consumption and low cost make them an excellent choice for many electronic projects.

How do you get the Arduino to do something?

1. Write a program that tells the chip what to do.
2. Then you have to compile it, that is, turn the program description into machine code that the Arduino understands. You may have to debug (find problems and correct them) before the code can be successfully sent to the Arduino.
3. Next you send the program to the chip using a programmer, which will transfer the machine code to the device.
4. Connect any required devices to the Arduino ( inputs such as switches and sensors and outputs such as motors and display screens, LED's) and run the program.

Arduino Projects-What can it Do?

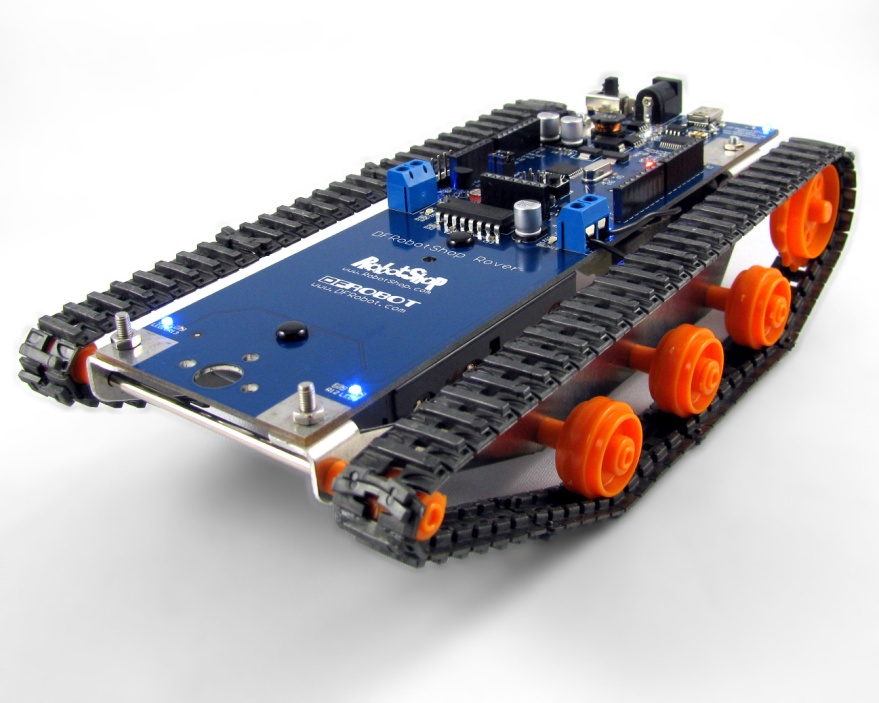
The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbie's, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.



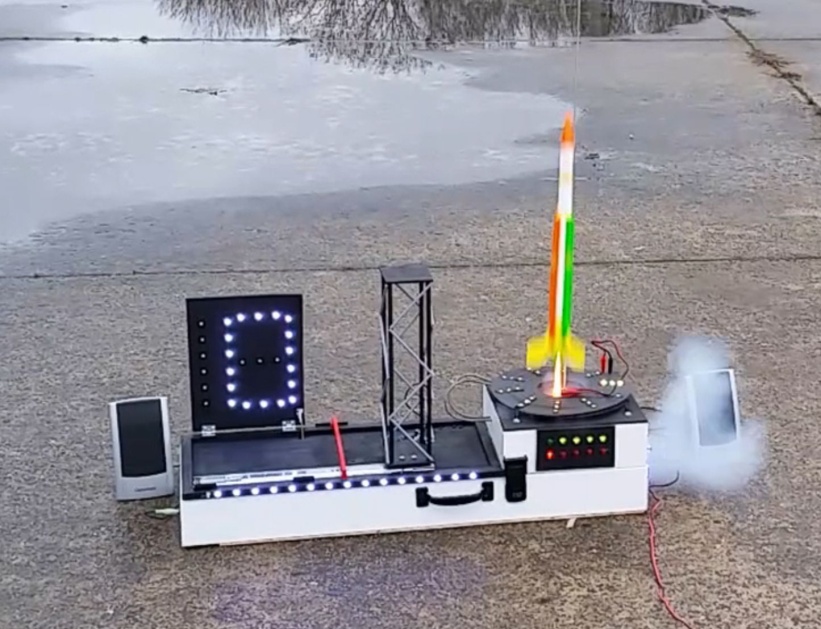
*Arduino Nano powered Key glove*



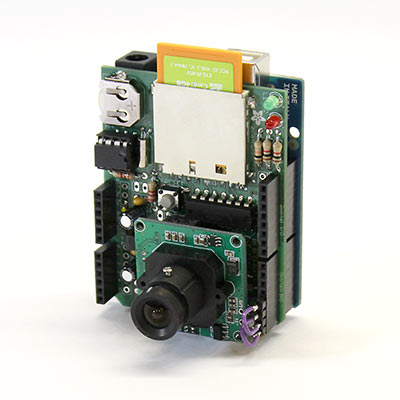
*Arduino MEGA 2560 powered Rubik Cube solver*



*Simple tank Robot*



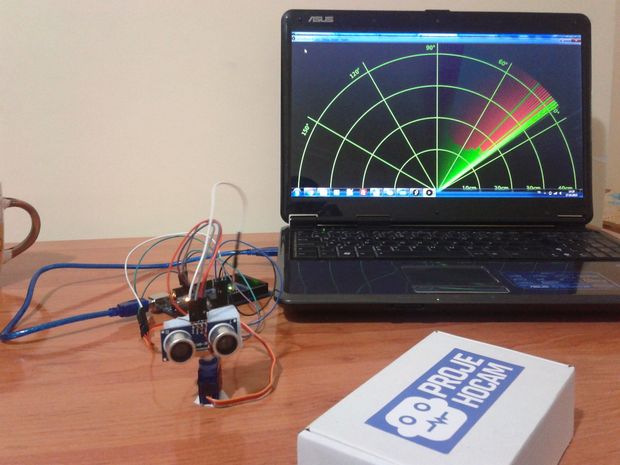
*Remote control Rocket Launcher*



*Remote WI-Fi Camera*



*GPS locator/Guide*



*Ultrasonic Radar*

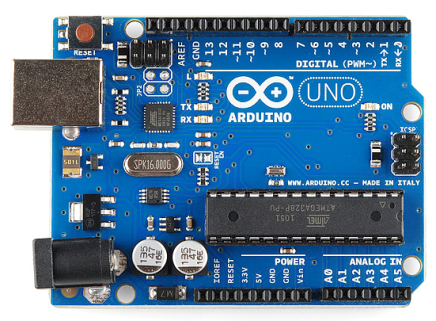


*Programmable light cube*

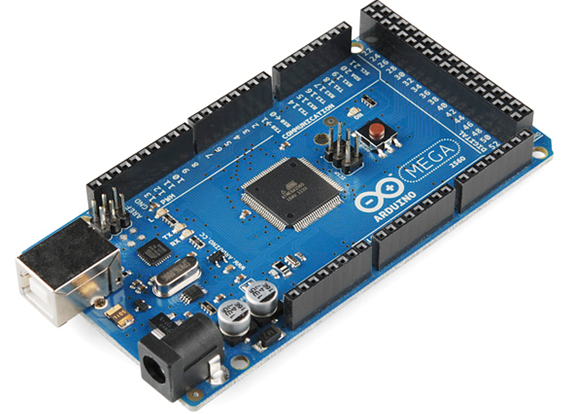
The Arduino Family

Arduino makes several different boards, each with different capabilities. In addition, part of being open source hardware means that others can modify and produce derivatives of Arduino boards that provide even more form factors and functionality. If you’re not sure which one is right for your project, [check this guide](https://www.sparkfun.com/arduino_guide) for some helpful hints. Here are a few options that are well-suited to someone new to the world of Arduino:

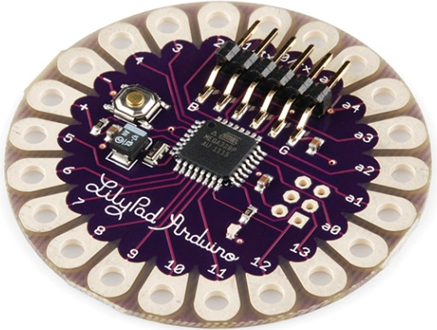
[Arduino Uno (R3)](https://www.sparkfun.com/products/11021)

The Uno is a good choice for your first Arduino. It’s got everything you need to get started and is very inexpensive. Most online projects and information about Arduino's use the UNO so it is especially suited to beginners. 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.

[Arduino Mega 2560 (R3)](https://www.sparkfun.com/products/11061)

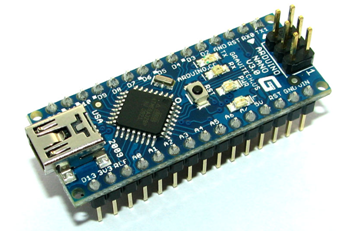
The Arduino Mega is the UNO’s big brother. It has lots (*54!*) of digital input/output pins (14 can be used as PWM outputs), 16 analog inputs, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The large number of pins make this board very handy for projects that require a bunch of digital inputs or outputs (like lots of LEDs or buttons). It also has 4 times the memory of the UNO.

[LilyPad Arduino](https://www.sparkfun.com/products/9266)



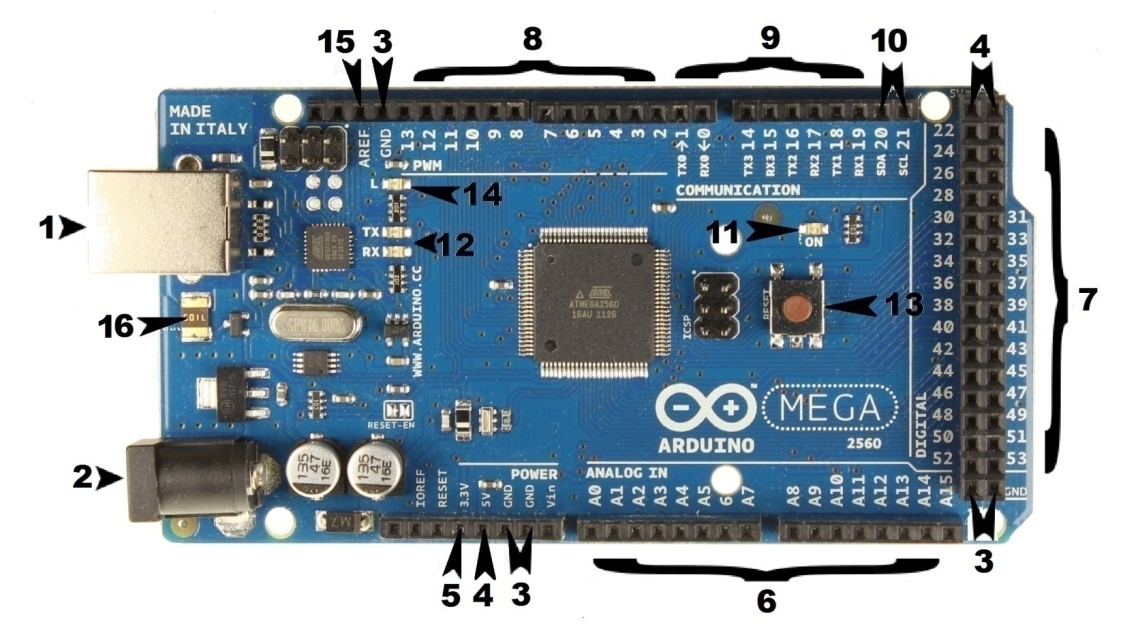
LilyPad is a wearable e-textile technology board. Each LilyPad was creatively designed with large connecting pads and a flat back to allow them to be sewn into clothing with conductive thread. The LilyPad also has its own family of input, output, power, and sensor boards that are also built specifically for e-textiles. They’re even washable.

Arduino Nano

The NANO is Arduino’s smallest development board to use one microcontroller with built-in USB. This means that it can be used for projects that require a very small and/or light microprocessor.

What's on an Arduino board?

The main parts on a MEGA 2560 are shown below:



Main Integrated Circuit

In the middle of the Arduino there is a large black square with metal legs attached to all 4 sides. This is the Microprocessor IC, or Integrated Circuit. Think of it as the brains of the Arduino. The main Integrated Circuit (IC) on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC.

Power

Every Arduino board needs a way to be connected to a power source. The Mega 2560 can be powered from a USB cable coming from your computer ( **1 )** or by a power supply (below) that connects to a barrel jack ( **2 )**.



For most projects power supplied by the USB is fine but some large projects or those involving motors and servos may require a more powerful wall power supply.

The USB connection is also how you will load code onto your Arduino board. More on how to program with Arduino can be found in  the "CE2 Arduino lesson 2 Your First Program"  tutorial.

**NOTE:** Do NOT use a power supply greater than 20 Volts as you will overpower (and thereby destroy) your Arduino. The recommended voltage for most Arduino models is between 6 and 12 Volts.

Pins

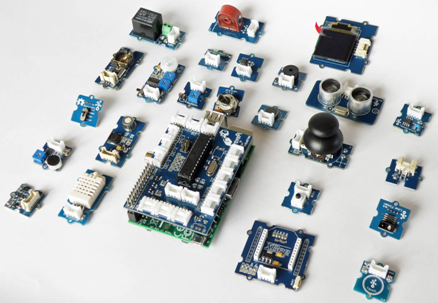
The pins on your Arduino are the places where you connect wires to construct a circuit (probably along with a breadboard and some wire). They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labelled on the board and used for different functions.

* **GND ( 3 )**: Short for ‘Ground’. There are 5 GND pins on the Arduino Mega, any of which can be used to ground your circuit.
* **5V ( 4 ) & 3.3V ( 5 )**: As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the components used with the Arduino run off of the 3 available 5 Volt pins.
* **Analog ( 6 )**: The area of pins under the ‘Analog In’ label (A0 through A15) are Analog In pins. These pins can read the signal from an analog sensor (like a temperature sensor) and convert it into a digital value that we can read.
* **Digital ( 7 )**: Across from the analog pins are the digital pins (22 through 53). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM ( 8 )**: Pins 2 to 13 act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM). Think of these pins as being able to simulate analog output (like fading an LED in and out).
* **TX RX ( 9 )** TX is short for transmit, RX is short for receive. These markings appear in electronics to indicate the pins responsible for [serial communication](https://learn.sparkfun.com/tutorials/serial-communication) of data. There are 4 pairs of RX/TX pins- Communication pins 0 and 1, and pins 14 to 19.
* **SDA/SCL** ( 10 ) The Serial Data Line (SDA) and Serial Clock Line (SCL) are used to communicate with several inputs or outputs at the same time. They are used with I2C capable components.
* **Power LED Indicator** **( 11 )**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong. Time to re-check your circuit!
* **TX/RX LEDS Indicator ( 12 )** These led's indicate when information is going into or out of the Arduino. You will see these flash when a program is being uploaded.
* **Reset Button** **( 13 )**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times.
* **PIN 13 LED** ( 14 ) An in line LED is connected to pin 13 to help in diagnosis or provide an easy way to see output on Pin 13. Any time Pin 13 is activated (HIGH) the led will light.
* **AREF ( 15 )**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.
* **Thermal Fuse** ( 16 ) This helps protect the arduino in case of excessive current draw and short circuits.

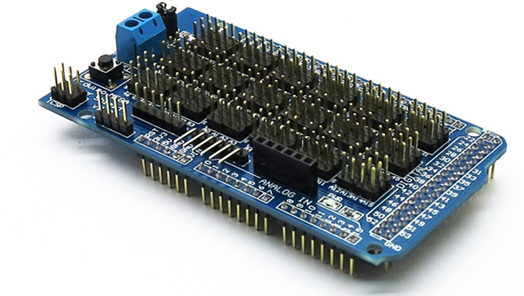
The Extended Family - Arduino Add-ons

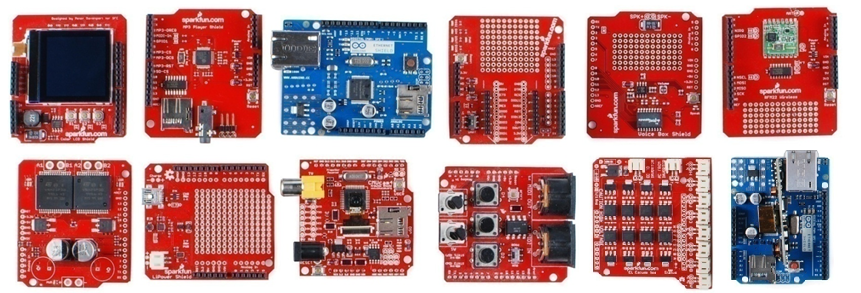
The Arduino can’t do a whole lot on its own – you’ve got to hook it up to something. In this section we’ll introduce basic **sensors** and **shields**, two of the most handy tools to use in bringing your projects to life.

Sensors

The main function of a microprocessor (including all computers) is to take input or information, manipulate it in some way and then output the results. The arduino is designed to work with special inputs called sensors. A sensor is similar to our own senses such as sight, hearing, and touching. With some simple code, the Arduino can control and interact with a wide variety of **sensors** - things that can measure [light](https://www.sparkfun.com/products/9088), [temperature](https://www.sparkfun.com/products/10988), [degree of flex](https://www.sparkfun.com/products/8606), [pressure](https://www.sparkfun.com/products/11207), [proximity](https://www.sparkfun.com/products/242), [acceleration](https://www.sparkfun.com/products/9836), [carbon monoxide](https://www.sparkfun.com/products/9403), [radioactivity](https://www.sparkfun.com/products/11345), [humidity](https://www.sparkfun.com/products/9569), [barometric pressure](https://www.sparkfun.com/products/9721), you name it, you can sense it!

Shields

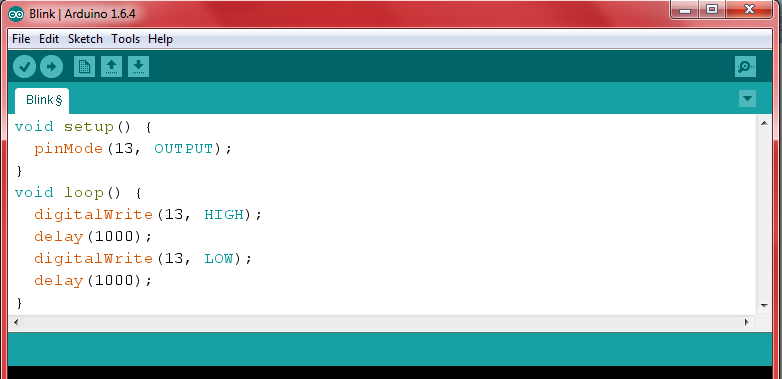
Additionally, there are attachments called **shields** – basically they are pre-built circuit boards that fit on top of your Arduino and provide additional capabilities – controlling motors, connecting to the internet, providing cellular or other wireless communication, controlling an LCD screen, and much more. The mega 2560 shield in your kit (shown on the left) is used mainly for allowing easy connections and adding extra power.



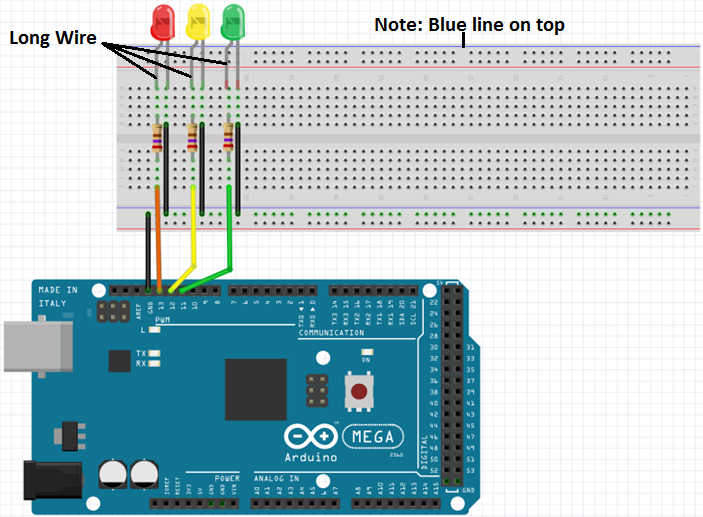
*A partial selection of available shields to extend the power of your Arduino*

Arduino Programming

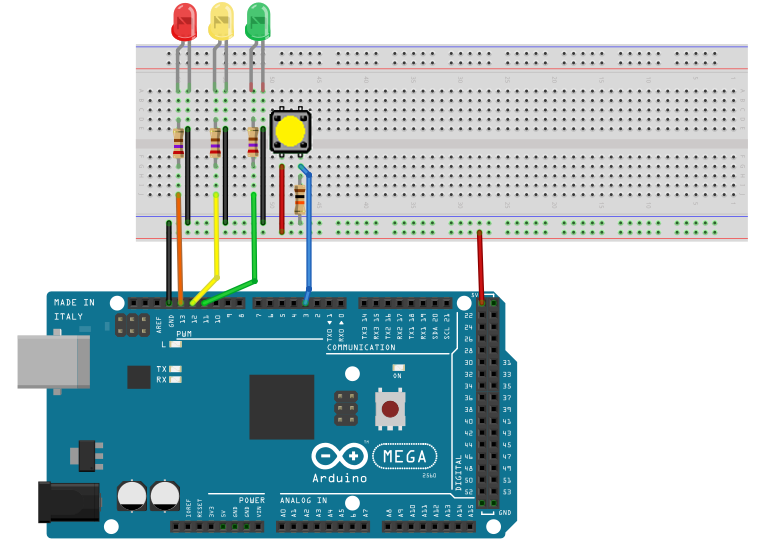
The Arduino is controlled by a user ( you!) developed program that instructs the Arduino to carry out specific instructions. The program you will use is called the Arduino IDE (Integrated Development Environment). The program may seem a bit strange to someone who has never done computer programming before but it is designed with beginners in mind and the support and tutorials available make it easy to learn. For example the first program you will create (below) has only 9 lines of code, 4 of which you don't even have to enter!



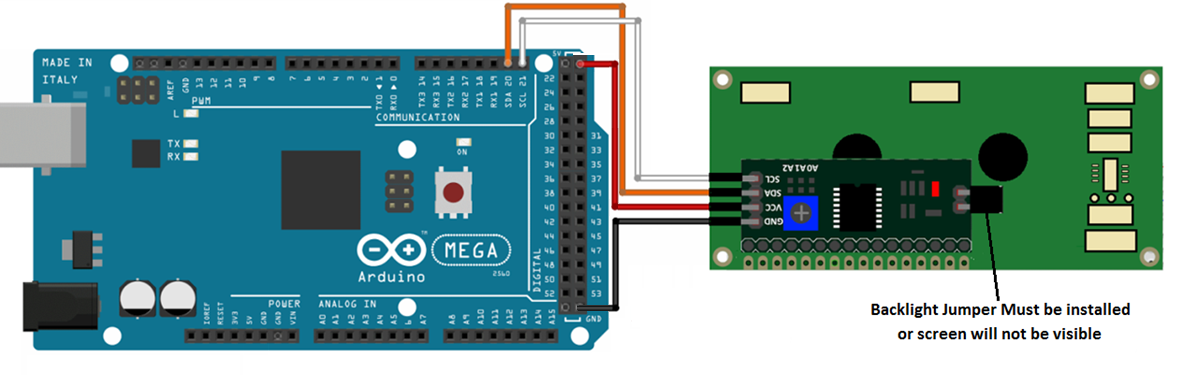
**Circuit 1 - Traffic Light**



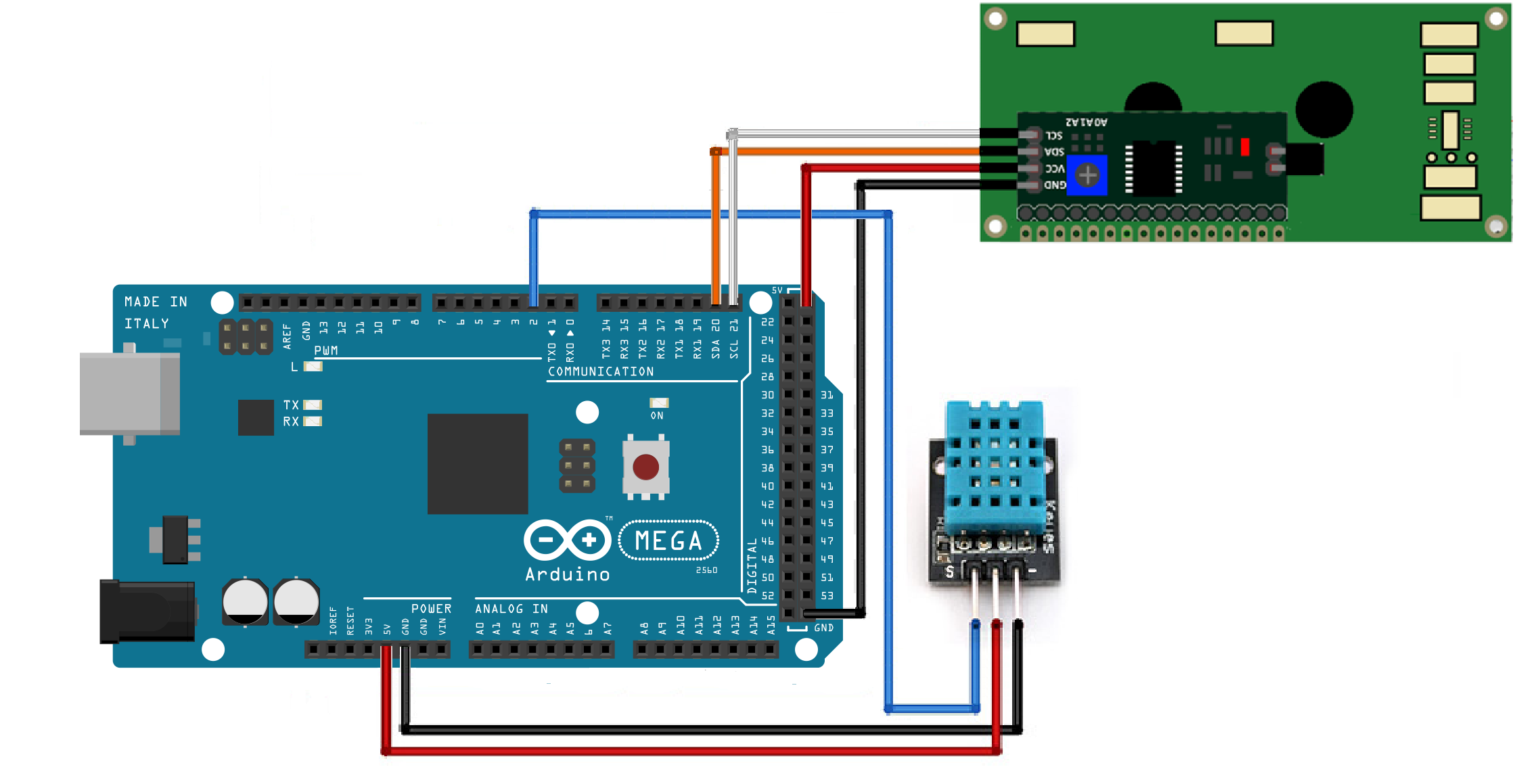
**Circuit 2 - LED's with Switch**

****

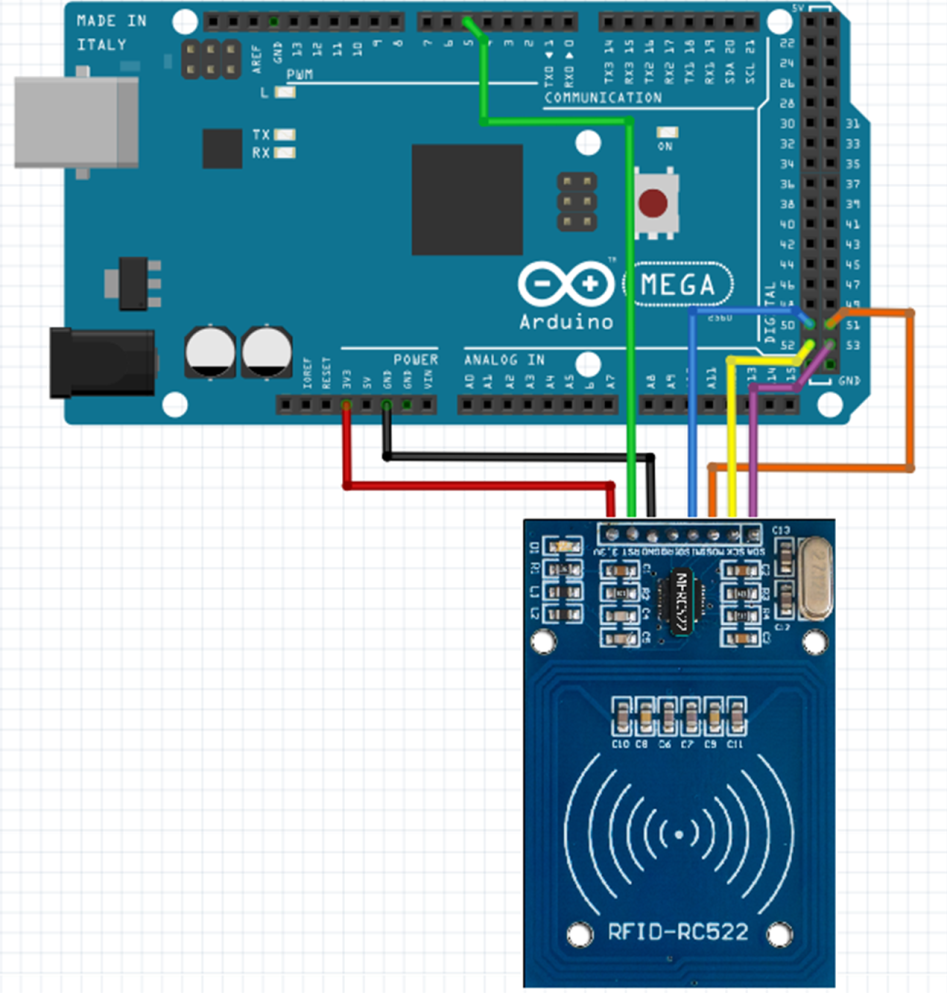
**Circuit 3A - LCD (Add the LCD to Circuit 2)**



**Circuit 3B - LCD with Temperature/Humidity Sensor**

****

**Circuit 4 - RFID (Add to Circuit 3) Note: Red wire is 3.3V NOT 5Volt!**

****