

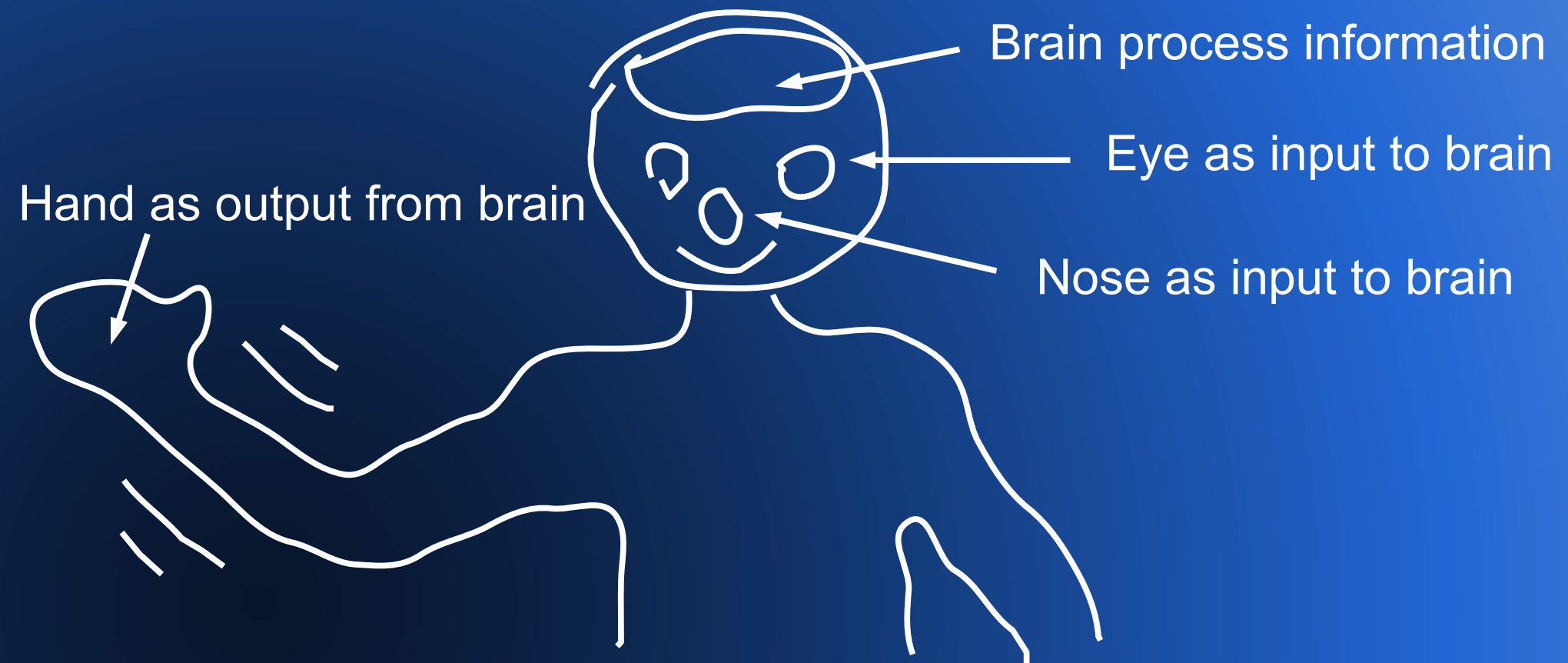
Introduction to Microcontroller

What is Microcontroller ?

In its simplest form, a microcontroller is a tiny computer that you can program to process inputs and outputs.

Introduction to Microcontroller

Compare microcontroller with human brain



Introduction to Arduino

Arduino UNO microcontroller is a tiny computer that you can be programmed to process inputs and outputs going to and from the Arduino processing unit using the Arduino programming language

Introduction to Arduino Hardware

- The Arduino microcontroller can be used to develop *stand-alone objects* or connected to a computer or LED displays, buttons, switches, motors, temperature sensors, distance sensors, etc.
- To program the Arduino you can use the *Arduino IDE* (Integrated Development Environment). In the case of the Arduino, the language is C. In the Arduino world, programs are known as 'sketches'.
- The Arduino hardware and software are both *Open Source*, which means the code, the schematics, design, etc. are all open for anyone to take freely and do what they like with it.
- The Arduino can also be extended with the use of '*Shields*' which are circuit boards containing other devices (e.g. LCD Displays, Ethernet connections, etc.) that you can simply slot into the top of your Arduino to get extra functionality.

Introduction to Arduino Hardware



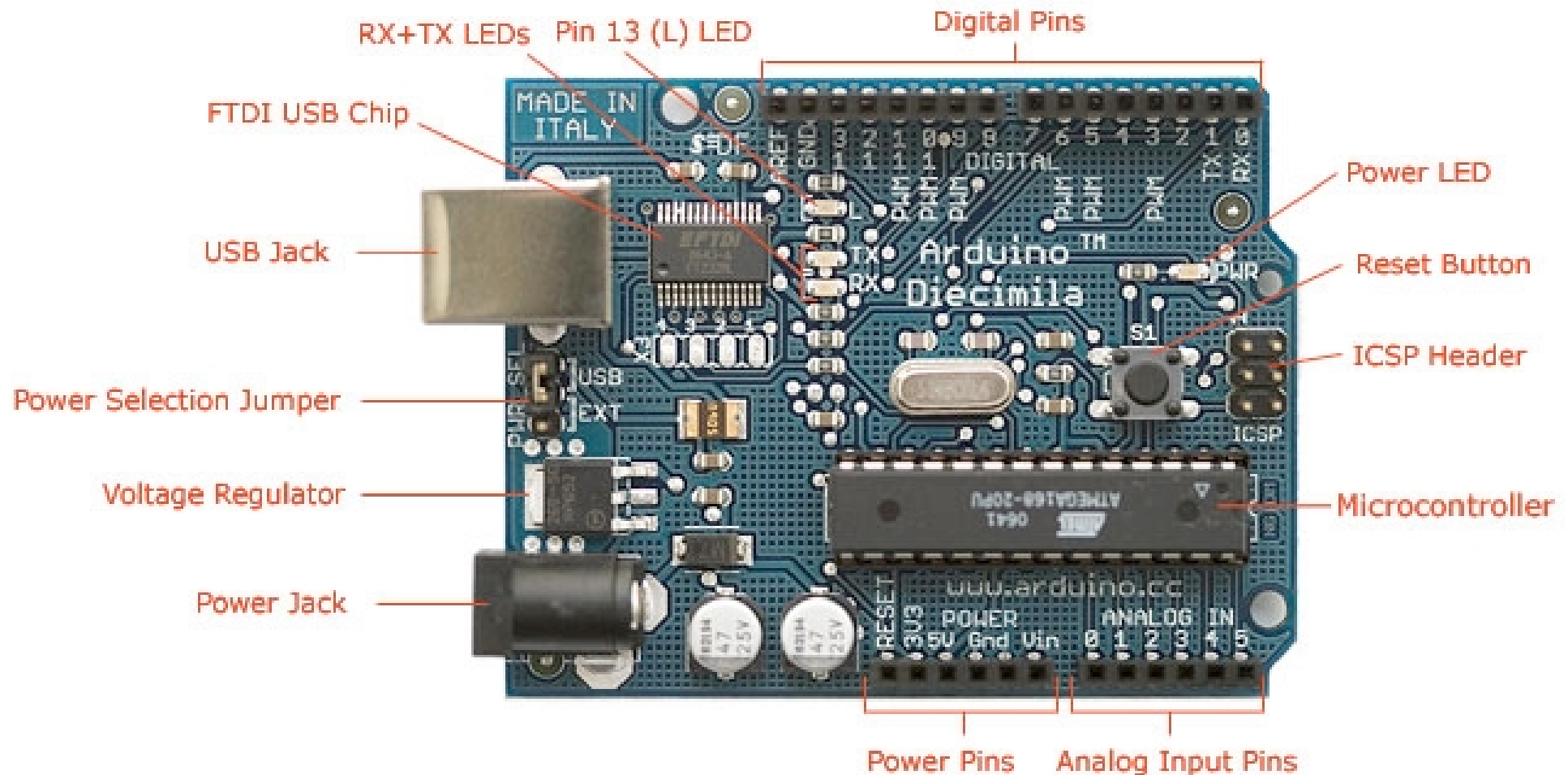
- Arduino Uno - This is the latest revision of the basic Arduino USB board
- It connects to the computer with a standard USB cable

Introduction to Arduino Hardware

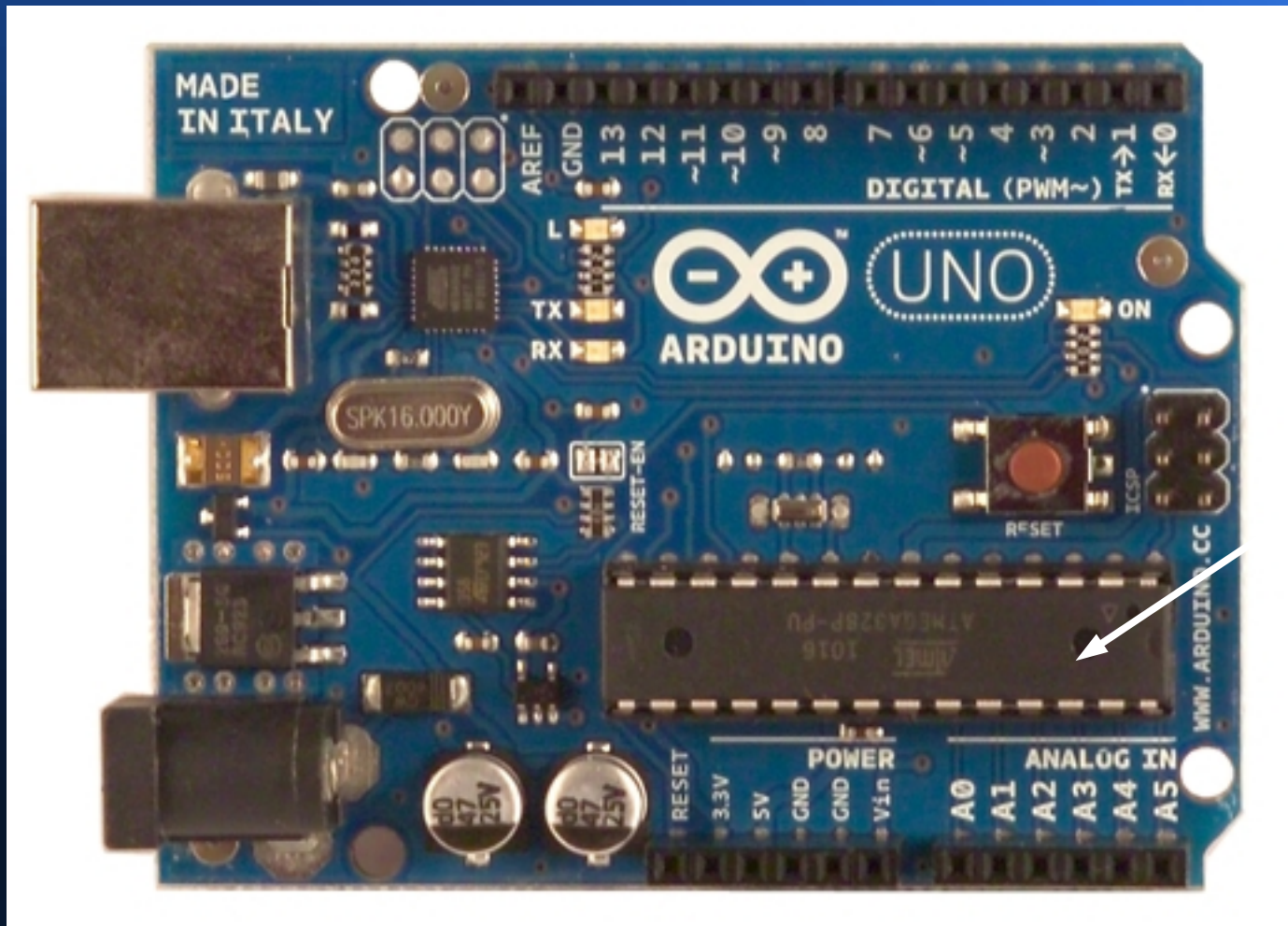
Specifications

- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Digital I/O Pins 14
- Analog Input Pins 6
- Flash Memory 32 KB
- Clock Speed 16 MHz

Introduction to Arduino Hardware



Introduction to Arduino Hardware



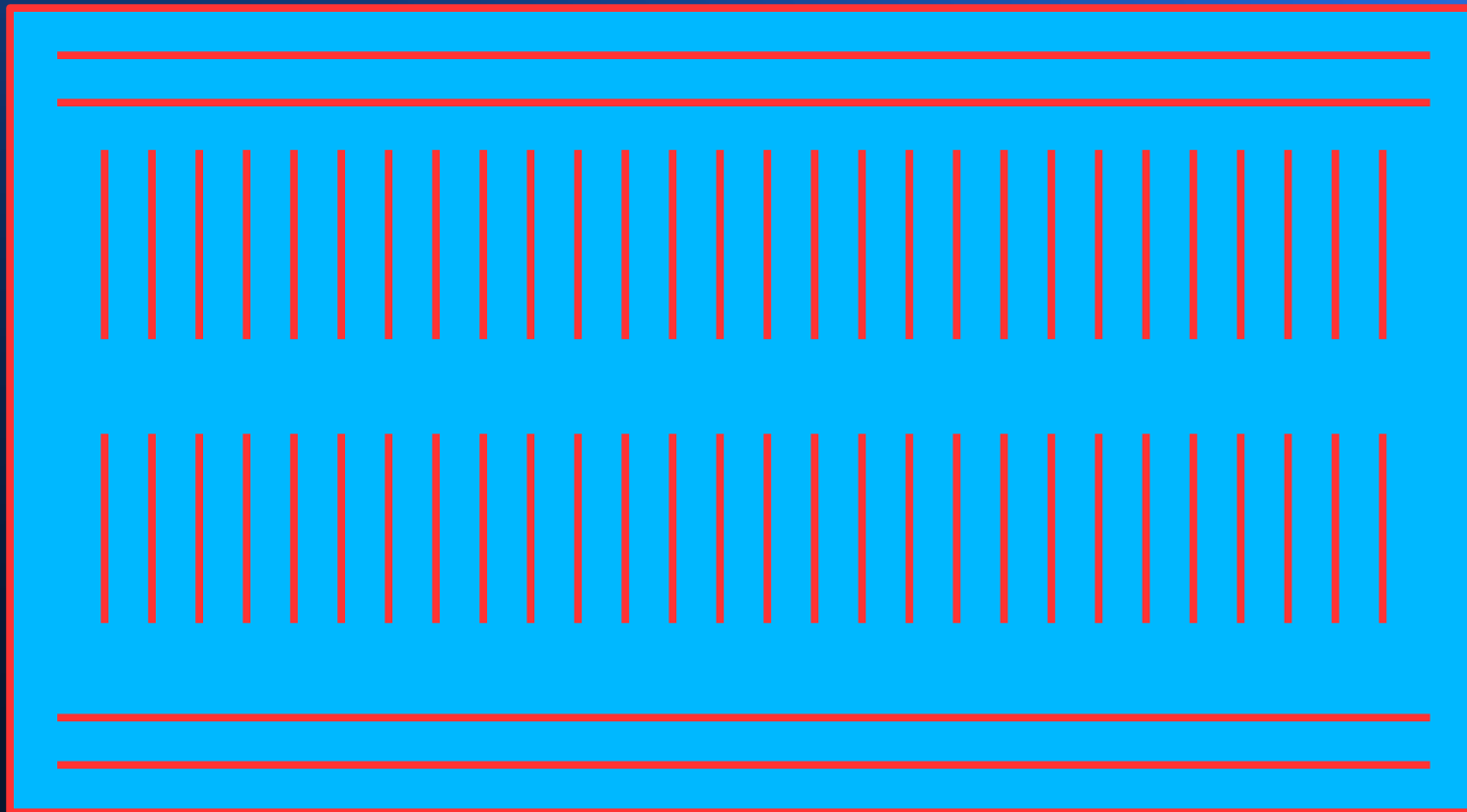
To PC ←

Atmega
328

To power ←

Introduction to Parts

Breadboard



Introduction to Parts

Resistor

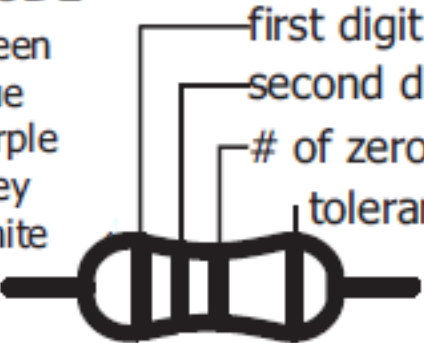
A resistor restricts the amount of current that can flow through a circuit. Resistor have color coding to determine the resistance.

RESISTOR COLOR CODE

0 - Black	5 - Green
1 - Brown	6 - Blue
2 - Red	7 - Purple
3 - Orange	8 - Grey
4 - Yellow	9 - White




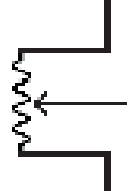



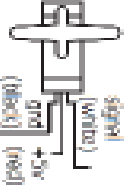
first digit
second digit
of zeros
tolerance

20%	- none
10%	- silver
5%	- gold

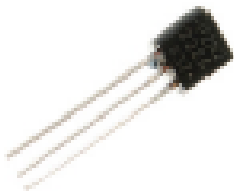
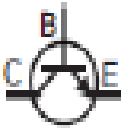






Orange-Orange-brown - 330 ohms
brown-black-orange - 10 000 ohms (10k)

Introduction to Parts

Component	Description	Image	Sym
Photoresistor	A photoresistor, light dependent resistor (LDR) or cadmium sulfide (CdS) cell is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a photoconductor. It needs to be in a voltage divider before it provides a useful input.		
Potentiometer	A potentiometer is a three-terminal resistor with a sliding contact that forms an adjustable voltage divider. If only two terminals are used (one side and the wiper), it acts as a variable resistor or rheostat. Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment.		
Push button	Push button is a simple switch mechanism. Buttons are typically made out of hard material, usually plastic or metal. The surface is usually flat or shaped to accommodate the human finger or hand, so as to be easily pushed.		
Servo Motor	Takes a timed pulse and converts it into an angular position of the output shaft. The connector has 3 wires coming out - White for signal, red for +ve & black for -ve.		

Introduction to Parts

Transistor	Uses a small current to switch or amplify a much larger current. Make sure to hook up in the right way round for the 3 leads - Base, Collector, Emitter. A current limiting resistor is often needed on the base pin.		
LED	A light-emitting diode (LED) is a semiconductor light source. LEDs are used as indicator lamps in many devices. Emits light when a small current is passed through it. (only in one direction). The longer lead connects to positive.		
Piezo Speaker	In a piezo 'speaker, a piece of piezoelectric material is attached to a diaphragm. If a voltage is applied across a piece of piezoelectric material it will flex, or change dimension in one direction. A pulse of current will cause it to click. stream of pulses will cause it to emit a tone.		

Getting Started

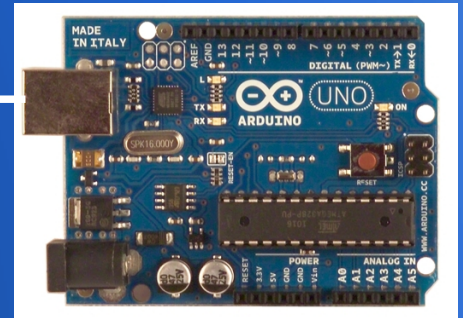
Quick Start

- Download the Arduino environment
- Connect the board to PC using USB cable
- Install the drivers
- Launch the Arduino application
- Open an example program.
- Select your board & your serial port
- Upload the program

Getting Started

- Download the Arduino environment
 - Download the following Software.
 - <http://arduino.googlecode.com/files/arduino-1.0-windows.zip>
- Connect the board to PC using USB cable

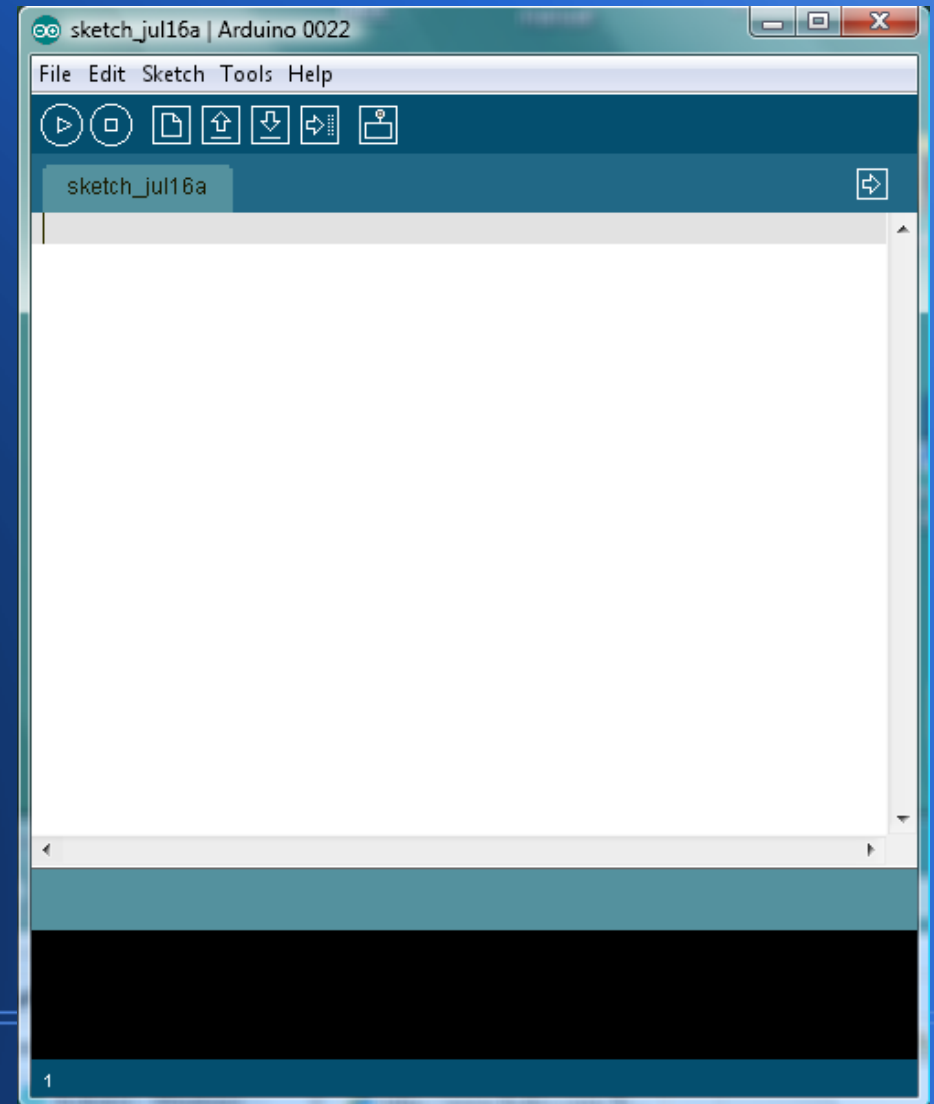
To PC via USB cable ←



- Install the drivers
 - Control Panel → System and Security.
 - System window → Device Manager → Ports (COM & LPT).
 - Select open port named "Arduino UNO" → Update driver
 - Navigate to and select the Uno's driver file, named "ArduinoUNO.inf", located in them "Drivers" folder of the Arduino Software download

Getting Started

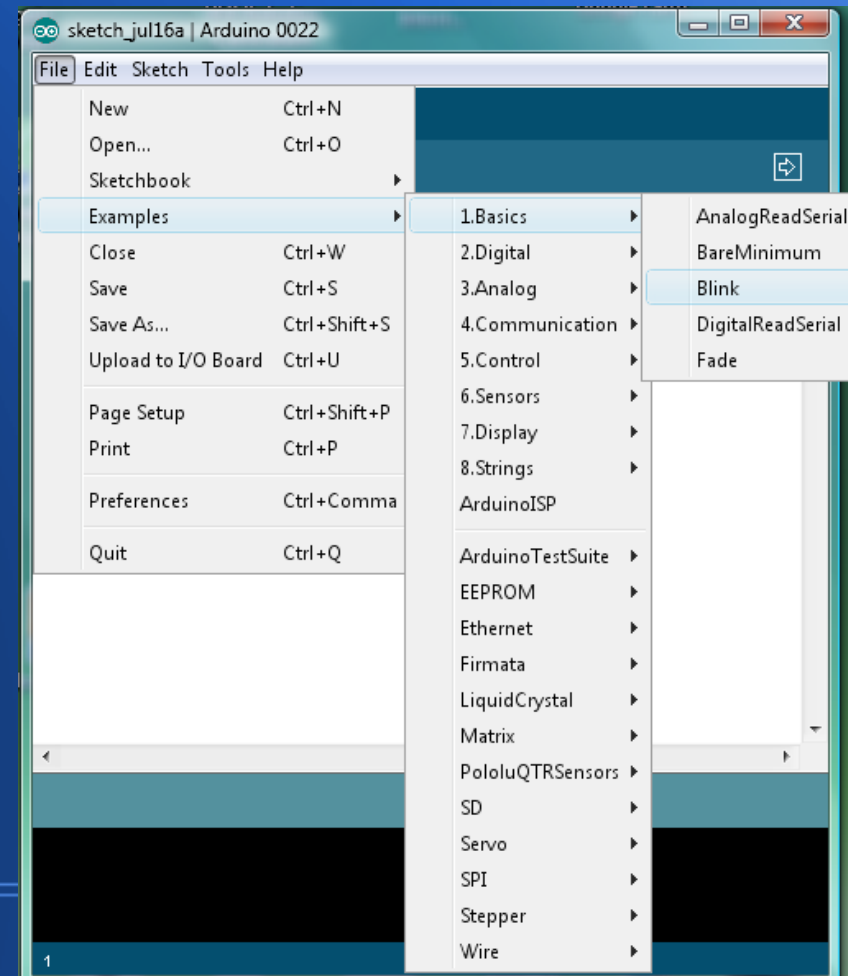
Launch the Arduino application



Getting Started

Open an example program

- LED blink : File > Examples > 1.Basics > Blink.



Getting Started

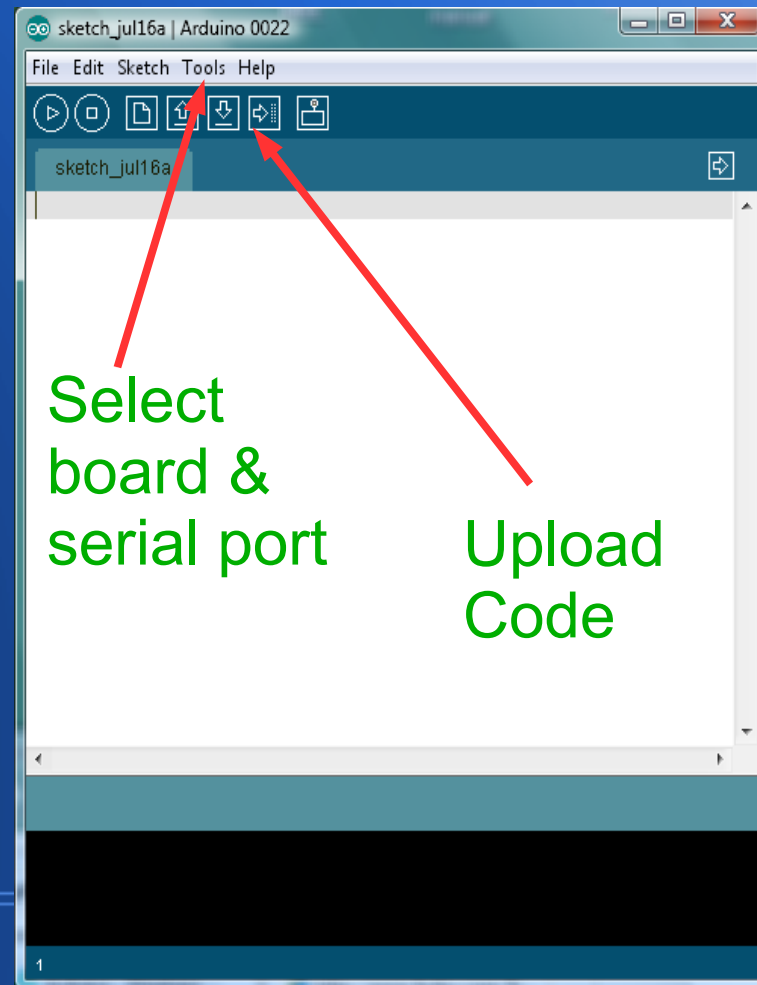
- Select your board & your serial port
- Upload the program

Wait a few seconds - you should see the RX and TX leds on the board flashing.

If the upload is successful, the message "Done uploading." will appear in the status bar.

A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange).

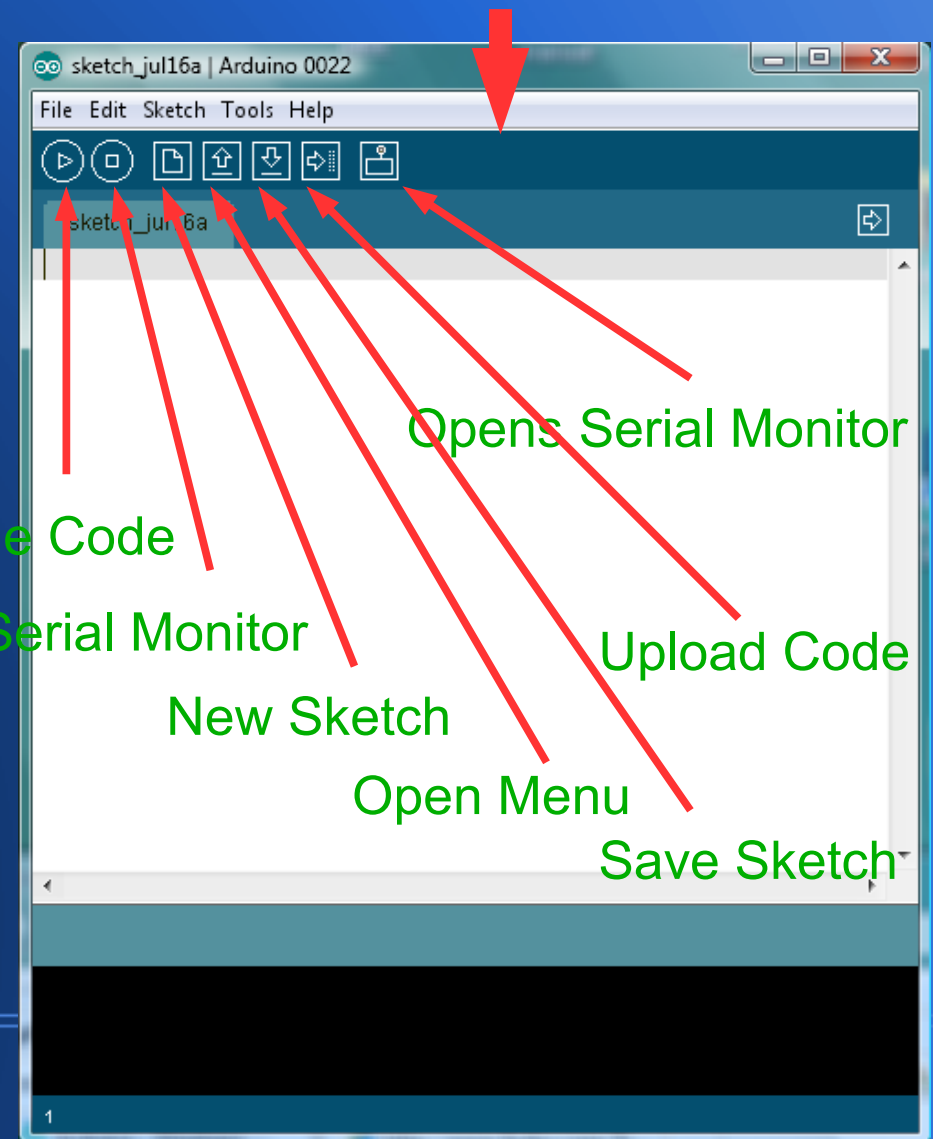
If it does, congratulations! You've gotten Arduino up-and-running.



Introduction to Arduino Software

- **Verify/Compile** - Checks your code for errors.
- **Stop** - Stops the serial monitor, or unhighlight other buttons.
- **New** - Creates a new sketch.
- **Open** - Presents a menu of all the *sketches* in your sketchbook. Clicking one will open it within the current window.
- **Save** - Saves your sketch.
- **Upload to I/O Board** - Compiles your code and uploads it to the Arduino I/O board.
- **Serial Monitor** - Opens the serial monitor.

Arduino IDE



Introduction to Arduino Software

Arduino Programming – First Step

1. Sketch - A sketch is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board.

2. Comments

The first few lines of the Blink sketch are a comment:

```
/*  
 * The basic Arduino example.  
 */
```

There's another style for short, single-line comments --> //

Introduction to Arduino Software

3. Each Arduino program (often called a sketch) has two required functions (also called routines).

- `void setup(){ }`

- The `setup()` is called once, when the sketch starts. It's a good place to do setup tasks like setting pin modes or initializing libraries. All the code between the two curly brackets will be run once when your Arduino program first runs.

- `void loop(){ }`

- This function is run after `setup` has finished. After it has run once it will be run again, and again, until power is removed.

Introduction to Arduino Software

4. FUNCTIONS

To create modular pieces of code that perform a defined task and then return to the area of code from which the function was "called", when one needs to perform the same action multiple times in a program.

DIGITAL I/O

--- pinMode(pin, mode) - Used to set a pin's mode. The mode can either be INPUT or OUTPUT.

--- digitalWrite(pin, value) - Once a pin is set as an OUTPUT, it can be set either HIGH (pulled to +5 volts) or LOW (pulled to ground).

ANALOG I/O

int analogWrite(pin, value)

- Some of the Arduino's pins support pulse width modulation (3, 5, 6, 9, 10, 11). This turns the pin on and off very quickly making it act like an analog output. The value is any number between 0 (0% duty cycle ~0v) and 255 (100% duty cycle ~5 volts).

Introduction to Arduino Software

Syntax - formatting requirements

- ;
(semicolon) - Each line of code must be ended with a semicolon.
- { }
(curly braces) - Used to define when a block of code starts and ends.
- //
(single line comment) - For writing notes to explain what each line of code does.
Everything until the end of the line will be ignored by program.
- /* */
(multi-line comment) - Everything between these two symbols will be ignored.
- #define - allows naming to a constant value before the program is compiled.
- #include - is used to include outside libraries in your sketch.

Control Structures – Controlling what runs next

if
if...else
for
while