# Starting to play with Arduino

Start from unboxing

- This page contains how to setup the software environment to start playing with Arduino.
- The example used will be the most commonly used model "Arduino UNO" in most popular starter's kits.
- The same procedure also apply to most other Arduino Products.

### Pre-requisite:

1. Having a set up Arduino board on hand. (Can purchase through Spark fun)

### **Objectives:**

1. Setup the environment and see something happening

#### **Descriptions:**

1. Several key elements you have to know on Arduino UNO (What is Type B USB connector):



- a. The official stable release is 1.0.5, which in most cases is the one you want to get.
- b. Download according to which system you are using(Windows, Mac, Linux).
- c. For windows specifically, you need to install the Arduino IDE before you plug the board onto your PC, for how windows handles device Drivers differently.
- 3. Connect the Arduino UNO to your Desktop/Laptop via USB
  - a. If you purchase an Arduino, it usually comes with the correct cable
  - b. In case you've lost the cable or the cable doesn't work, what you need is a Type A to Type B USB cable. (What are these)
  - c. The computer will recognize the device
    - i. In windows its COM# (# being a number)

- ii. in UNIX-like system it's USBTTY (Mac/Linux/Solaries/etc.)
- 4. Start the Arduino IDE a. Should be something that looks like this:
  - \_ 🗆 🗙 💿 sketch\_mar29a | Arduino 1.5.4 File Edit Sketch Tools Help Verify your programme 1 ø æ sketch\_mar29a • Upload your programmevoid setup() { . // put your setup code here, to run once: } void loop() { // put your main code here, to run repeatedly: } •
  - b. Select which interface you are using for the Arduino

     For Windows, it should be COM# (# stands for a number)

ii. For Unix-like system (usually Mac for design students) it should be /dev/tty.usbmodem##### (# stands for a number or a character)



- c. Select the Board-type
  - i. In our case Arduino UNO



d. Go to File Example and under 01.Basics Blink

pen the example sketch			
🥺 sketch_mar29a   Arduino 1.	5.4		_ 🗆 X
File Edit Sketch Tools H	elp		
New	Ctrl+N		
Open	Ctrl+O		
Sketchbook	•		
Examples	•	01.Basics 🕨 🕨	AnalogReadSerial
Close	Ctrl+W	02.Digital 🕨 🕨	BareMinimum
Save	Ctrl+S	03.Analog 🌙	Blink
Save As	Ctrl+Shift+S	04.Communication	DigitalReadSerial
Upload	Ctrl+U	05.Control	Fade
Upload Using Programmer	Ctrl+Shift+U	06.Sensors	ReadAnalogVoltage
Page Setup	Ctrl+Shift+P	07.Display 🕨 🕨	
Print	Ctrl+P	08.Strings 🔹 🕨	
		09.USB	
Preferences	Ctrl+Comma	10.StarterKit 🕨 🕨	
Quit	Ctrl+Q	ArduinoISP	
		Bridge 🕨 🕨	
		EEPROM 🕨	
		Esplora 🕨 🕨	
		Ethernet 🕨 🕨	<b>_</b>
		Firmata 🕨 🕨	
		GSM 🕨	
		LiquidCrystal 🔹 🕨	
		Robot Control 🔹 🕨	
		Robot Motor 🔹 🕨	
		RobotIRremote 🔹 🕨	
		SD 🕨	o Uno on COM1
		Servo 🕨	
		SoftwareSerial 🔹 🕨	
		SPI 🕨	
		Stepper 🕨 🕨	
		TFT 🕨	
	Ì	WiFi 🕨	
		Wire 🕨	

#### ii. You will see a programme like this

```
Example programme from Arduino Blink
/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
This example code is in the public domain.
*/
// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;
// the setup routine runs once when you press reset:
void setup() {
    // initialize the digital pin as an output.
    pinMode(led, OUTPUT);
```

```
}
// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);
                            // wait for a second
  digitalWrite(led, LOW);
                            // turn the LED off by making the voltage LOW
  delay(1000);
                            // wait for a second
}
```

- iii. Upload the programme to Arduino iv. You should see the LED onboard blinking



v. Try modify the "1000" in the delay function and see what happens with the blinking.

## Appendix:

• Types of USB connectors