

To follow
the speed of
the world

Try!

Basic Arduino Programming

with C - language

Written by Ji Hu Jeon

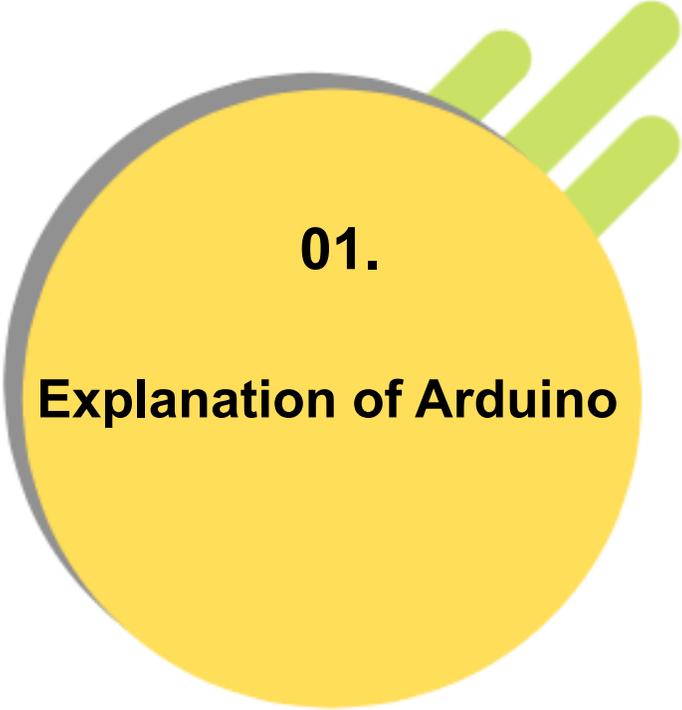
“Wait that side, and people who intend to
and leave hard can arrive in mountaintop all.
The mountain is conquered to a person who rise.”

Shakespeare

Contents

1. Explanation of Arduino
2. Setting Development Environment
 - 2.1 - How to Download Arduino Uno
 - 2.2 - How to use Tinkercad
3. Arduino Circuits & Sensors
 - 3.1 - Arduino Uno
 - 3.2 - Breadboard
 - 3.3 - LED
 - 3.4 - Jumper wires
 - 3.5 - Servo motor
 - 3.6 - Resistors
 - 3.7 - LCD
 - 3.8 - DC motor & H-bridge motor driver
 - 3.9 - Push button
4. Examples of circuits
 - 4.1 - LED & Resistor
 - 4.2 - Button & Resistor, Servo motor
 - 4.3 - DC motor & H-bridge motor driver
 - 4.4 - LCD



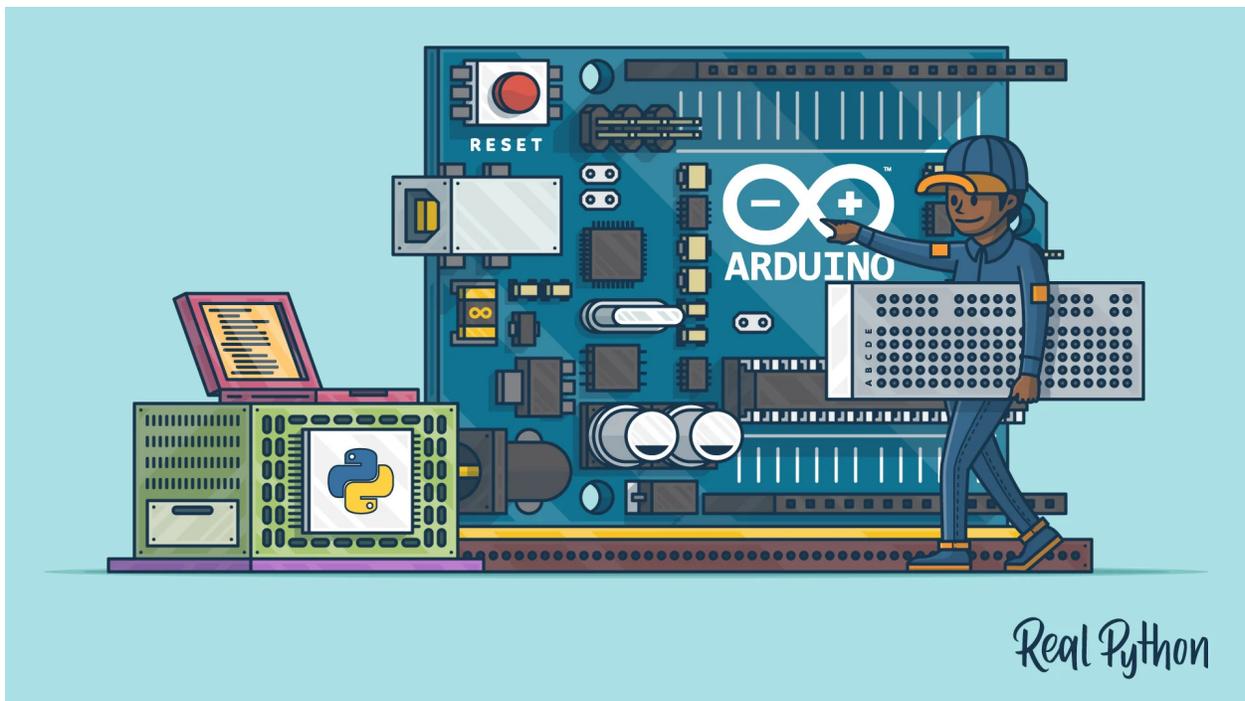


01.

Explanation of Arduino



Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. Hardware products are licensed under a CC-BY-SA license, while software is licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially from the official website or through authorized distributors.





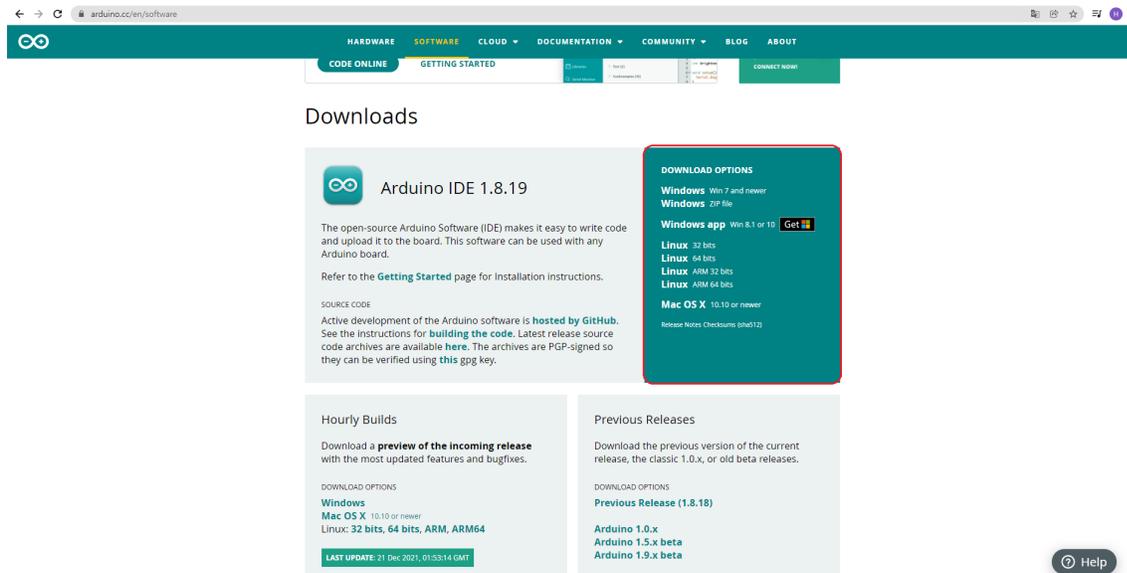
02.

**Setting Development
Environment**



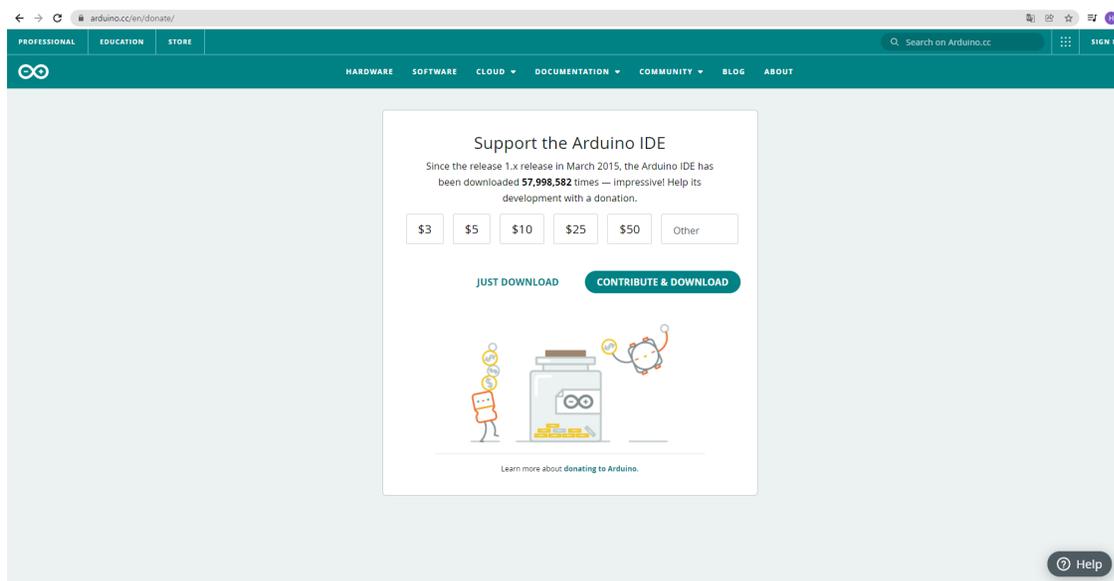
2.1 - How to Download Arduino Uno

To download the arduino software, open a web browser and go to the arduino official site <https://www.arduino.cc/en/software>. Check the options fits with your pc or laptop and click the download option that fits.



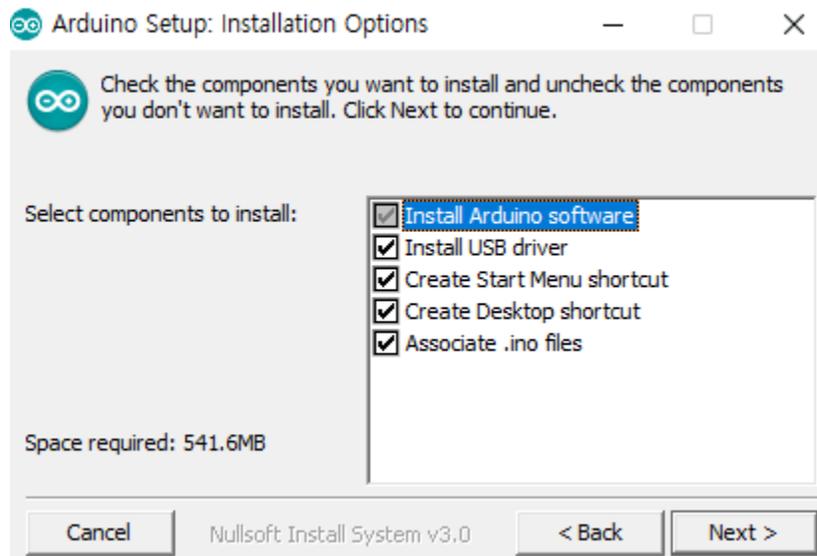
The screenshot shows the Arduino IDE 1.8.19 download page. The main heading is "Downloads". The primary content area features the Arduino IDE 1.8.19 logo and a description: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions." Below this is a "SOURCE CODE" section stating that active development is hosted by GitHub and provides links for building the code and verifying source code archives. A "DOWNLOAD OPTIONS" box is highlighted with a red border, listing options for Windows (Win 7 and newer, ZIP file), Windows app (Win 8.1 or 10), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), and Mac OS X (10.10 or newer). Below the main content are sections for "Hourly Builds" (with a "LAST UPDATE: 21 Dec 2021, 01:53:14 GMT" badge) and "Previous Releases" (listing Arduino 1.0.x, 1.5.x beta, and 1.9.x beta). A "Help" button is visible in the bottom right corner.

Click “just download”. There is no other restriction of service without contribution or donation.



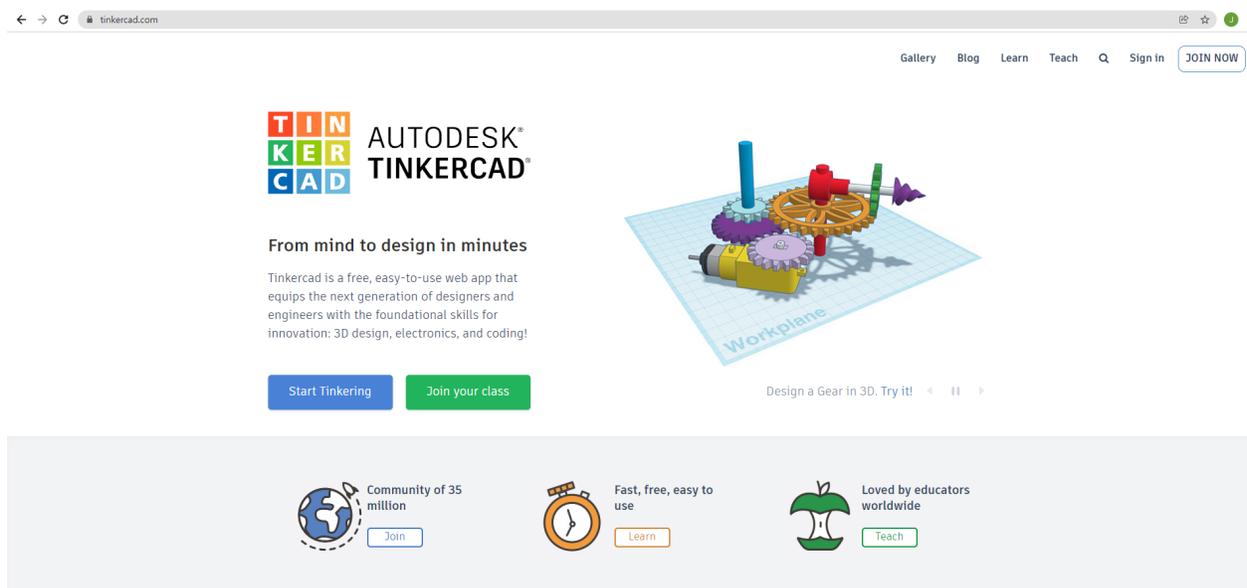
The screenshot shows the Arduino donation page. The main heading is "Support the Arduino IDE". Below the heading is a paragraph: "Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **57,998,582** times — impressive! Help its development with a donation." There are five buttons for donation amounts: "\$3", "\$5", "\$10", "\$25", "\$50", and "Other". Below these buttons are two buttons: "JUST DOWNLOAD" and "CONTRIBUTE & DOWNLOAD". At the bottom of the donation area is an illustration of a robot and a stack of coins, with the text "Learn more about donating to Arduino." and a "Help" button in the bottom right corner.

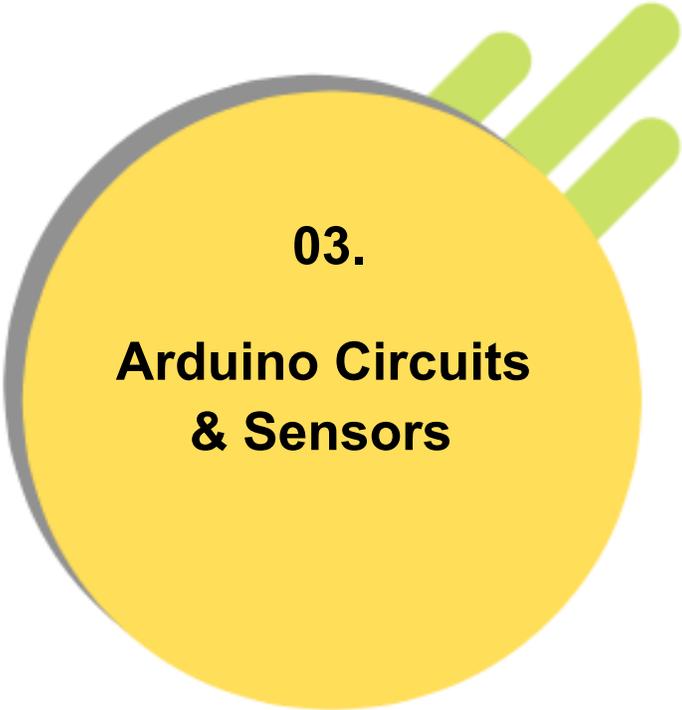
For creating desktop and start menu shortcuts, it is unnecessary to check it.



2.2 - How to use Tinkercad

Tinkercad is another option for arduino but it is only available for simulation. It is not able to connect with hardware. It is a free website (<https://www.tinkercad.com/>) and able to use it after signing in.

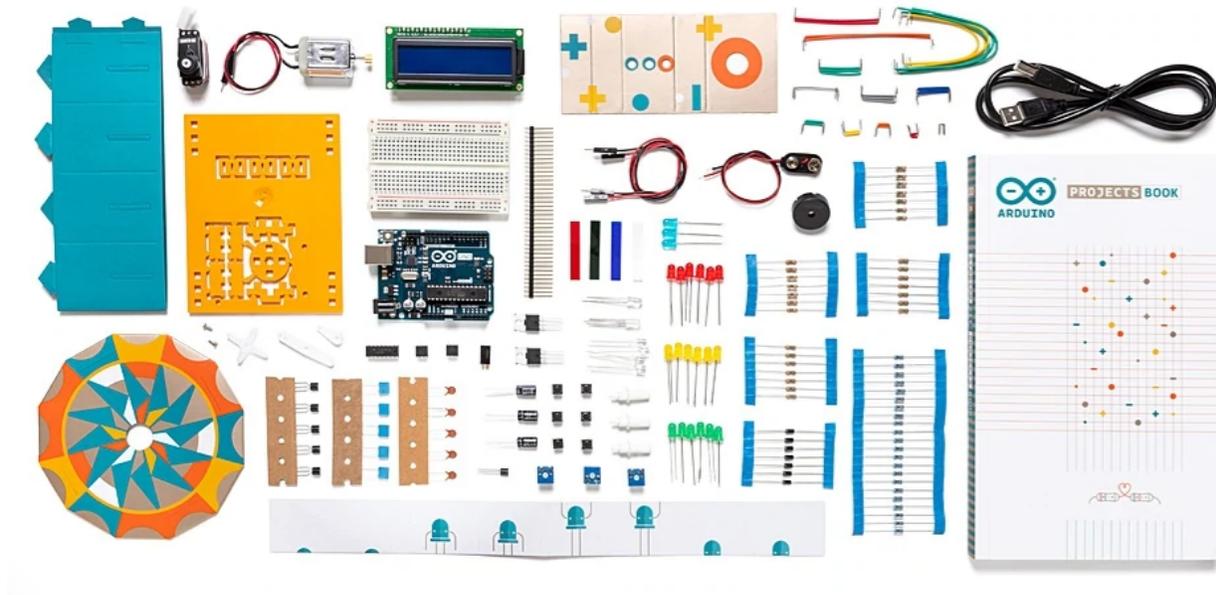




03.

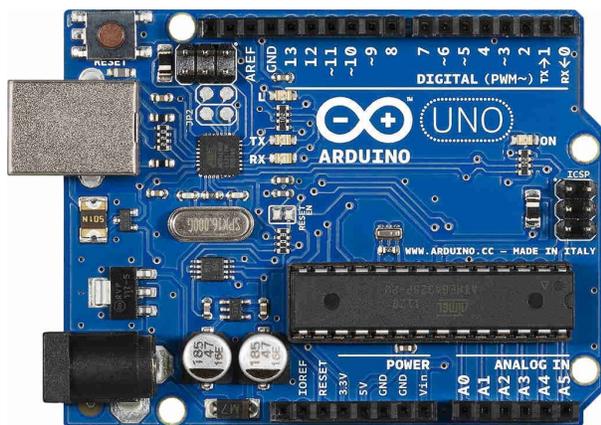
**Arduino Circuits
& Sensors**

Arduino starter kit has Uno R3, Breadboard, Jumper wires, LED, Push button, Servo motor, Resistors, LCD, DC motor, H-bridge motor driver.



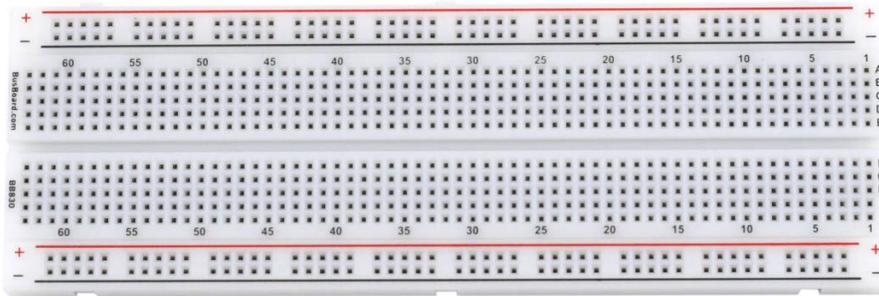
3.1 - Arduino Uno

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button.



3.2 - Breadboard

A breadboard is a construction base for prototyping of electronics. Originally the word referred to a literal bread board, a polished piece of wood used when slicing bread.



3.3 - LED

LED(Light emitting diode) is a semiconductor light source that emits light when a voltage is applied.



3.4 - Jumper wires

A jumper wire is an electrical wire or group of them in a cable with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components without soldering.



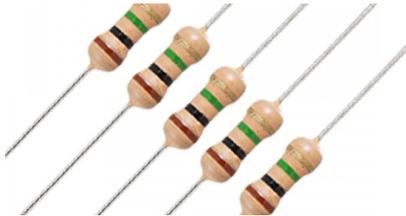
3.5 - Servo motor

A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.



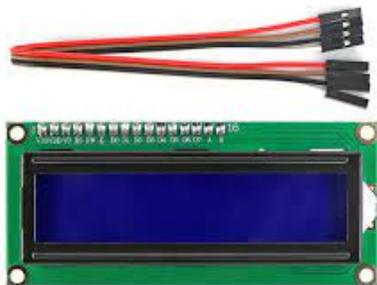
3.6 - Resistors

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.



3.7 - LCD

A liquid-crystal display is a flat-panel display or other electronically modulated optical device that uses the light-modulating properties of liquid crystals combined with polarizers which is an optical filter that lets light waves of a specific polarization pass through while blocking light waves of other polarizations.



3.8 - DC motor & H-bridge motor driver

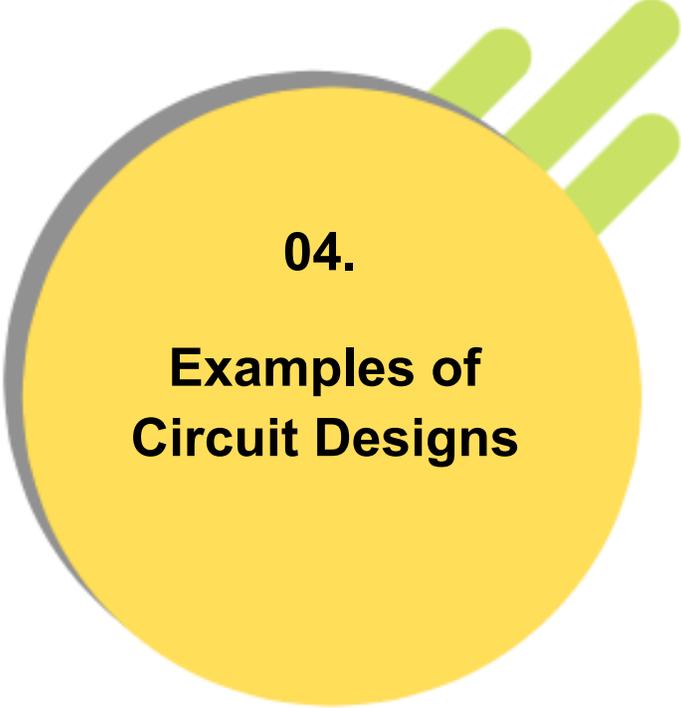
An H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. These circuits are often used in robotics and other applications to allow DC motors to run forwards or backwards.



3.9 - Pushbutton

Pushbutton is a simple switch to control some parts of a machine or a process. It is a simple switch to control some parts of the machine and it is made of hard materials to push the button easily.



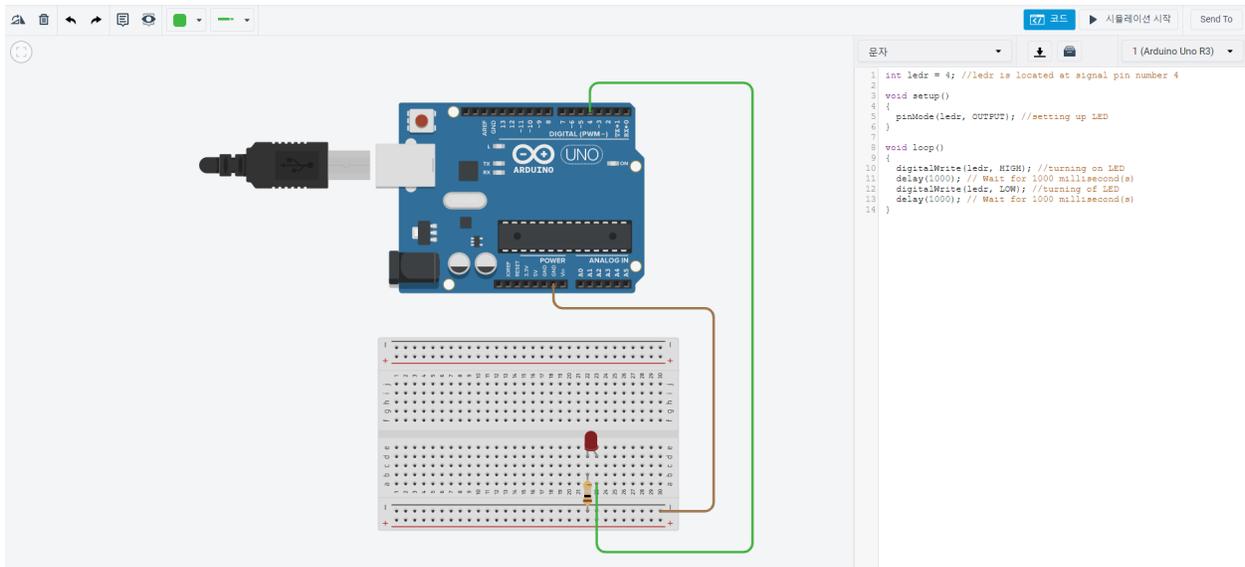


04.

**Examples of
Circuit Designs**



4.1 - LED & Resistor



Code:

```
int ledr = 4; //ledr is located at digital pin number 4
```

```
void setup()
```

```
{
```

```
  pinMode(ledr, OUTPUT); //setting up LED
```

```
}
```

```
void loop()
```

```
{
```

```
  digitalWrite(ledr, HIGH); //turning on LED
```

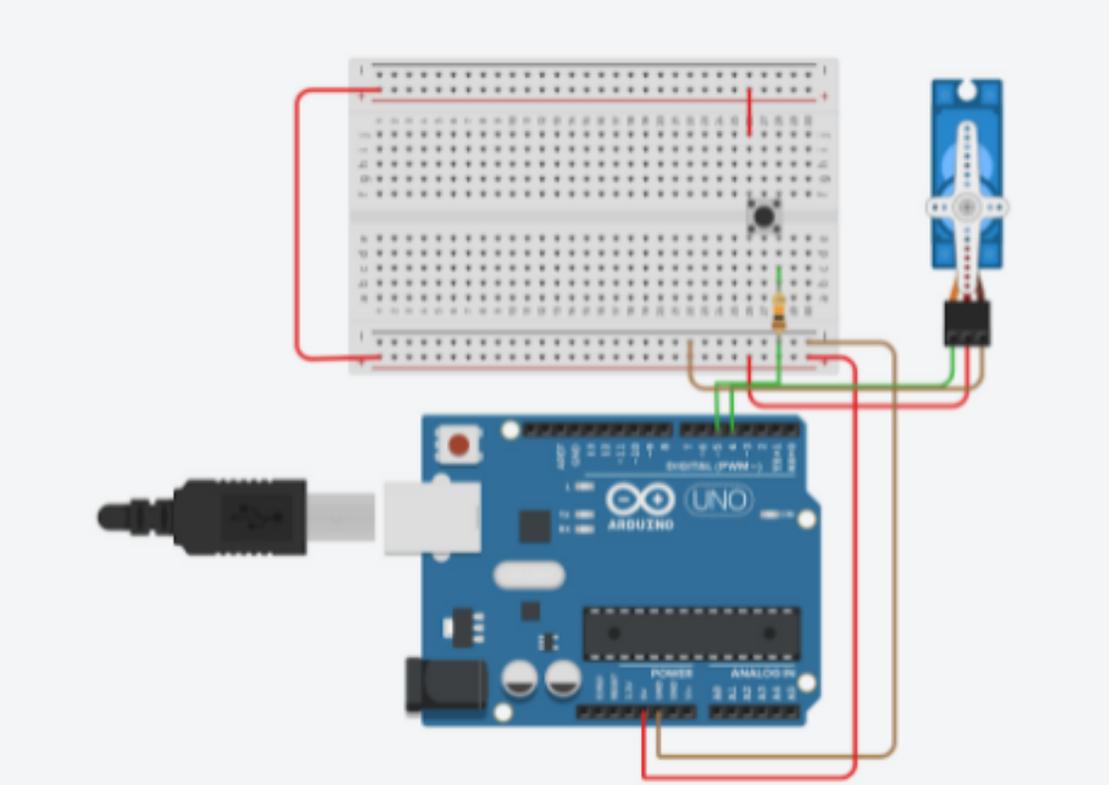
```
  delay(1000); // Wait for 1000 millisecond(s)
```

```
  digitalWrite(ledr, LOW); //turning of LED
```

```
  delay(1000); // Wait for 1000 millisecond(s)
```

```
}
```

4.2 - Button, servo motor, resistor



Code:

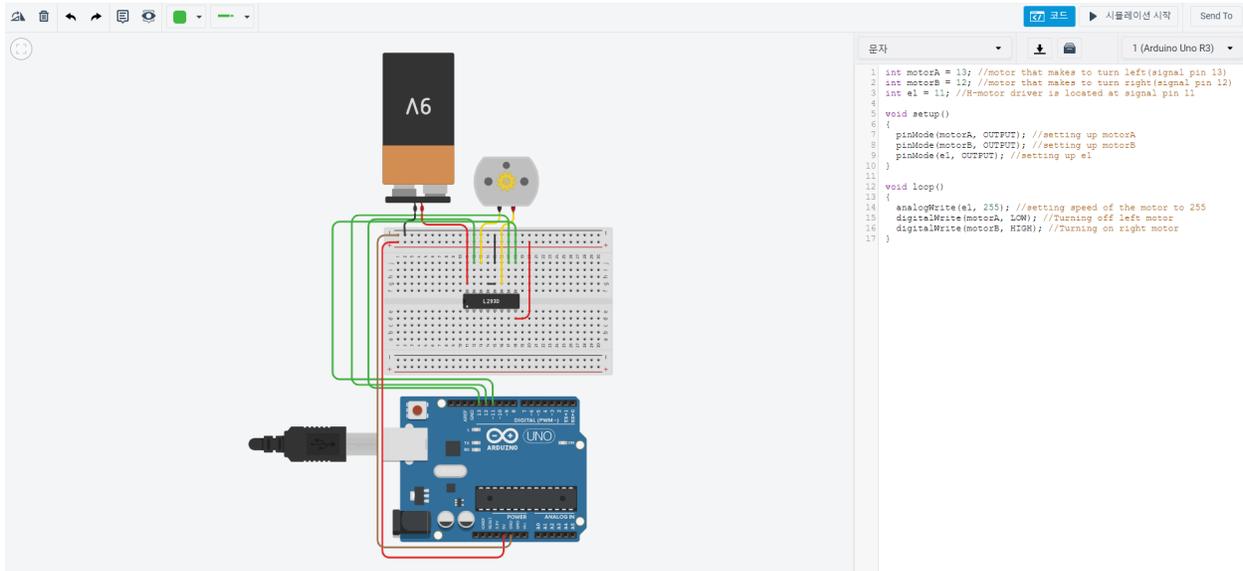
```
#include<Servo.h> //headerfile to use servo motor

int pb = 5; //pushbutton is connected at digital pin number 5
int status = 0; //declare any variable
Servo motor; //define servo motor

void setup()
{
    motor.attach(4); //servo motor is connected to signal pin number4
    pinMode(pb, INPUT); //setting up push button
}

void loop()
{
    status = digitalRead(pb); //set status as the variable of pushed
pushbutton
    if(status==HIGH) // if button is pushed
    {
        motor.write(45); //move motor 45 degrees
        delay(1000); //Wait for 1000 millisecond(s)
    }
    else // if not
    {
        motor.write(0); //Return motor to original position
        delay(1000); //Wait for 1000 millisecond(s)
    }
}
```

4.3 - DC motor & H-bridge motor driver



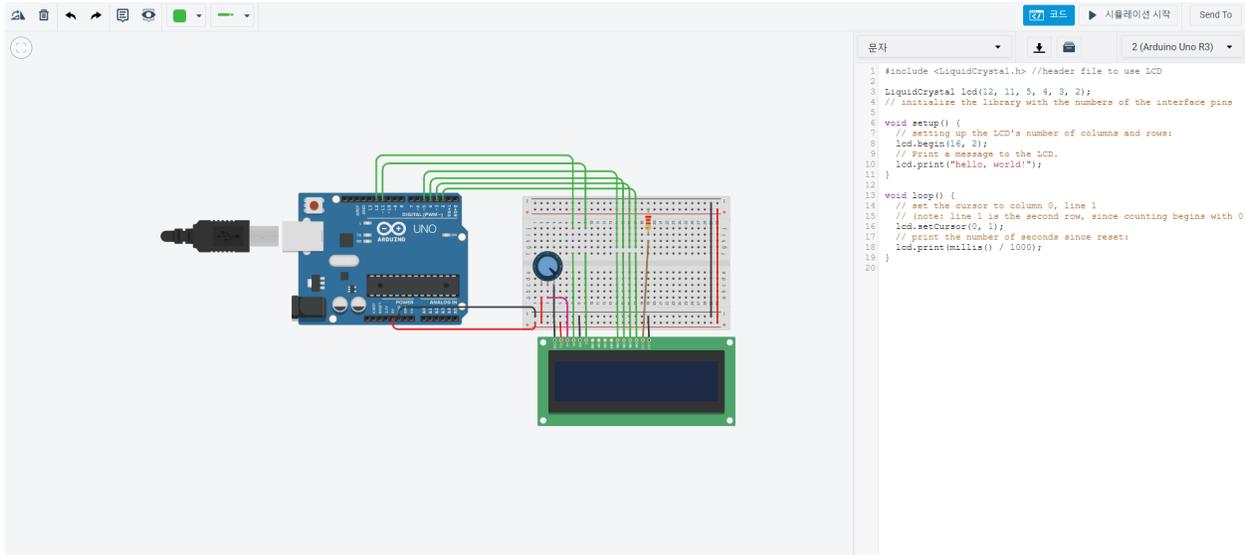
Code:

```
int motorA = 13; //motor that makes to turn left(digital pin 13)
int motorB = 12; //motor that makes to turn right(digital pin 12)
int e1 = 11; //H-motor driver is located at signal pin 11
```

```
void setup()
{
  pinMode(motorA, OUTPUT); //setting up motorA
  pinMode(motorB, OUTPUT); //setting up motorB
  pinMode(e1, OUTPUT); //setting up e1
}
```

```
void loop()
{
  analogWrite(e1, 255); //setting speed of the motor to 255
  digitalWrite(motorA, LOW); //Turning off left motor
  digitalWrite(motorB, HIGH); //Turning on right motor
}
```

4.4 - LCD



Code:

```
#include <LiquidCrystal.h> //header file to use LCD
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
// initialize the library with the numbers of the interface pins
```

```
void setup()
```

```
{
```

```
  lcd.begin(16, 2);
```

```
  // setting up the LCD's number of columns and rows
```

```
  lcd.print("hello, world!");
```

```
  // Print a message to the LCD
```

```
}
```

```
void loop()
```

```
{
```

```
  lcd.setCursor(0, 1);
```

```
  // set the cursor to column 0, line 1
```

```
  // (note: line 1 is the second row, since counting begins with 0)
```

```
  lcd.print(millis() / 1000);
```

```
  // print the number of 1000 milliseconds since reset
```

```
}
```