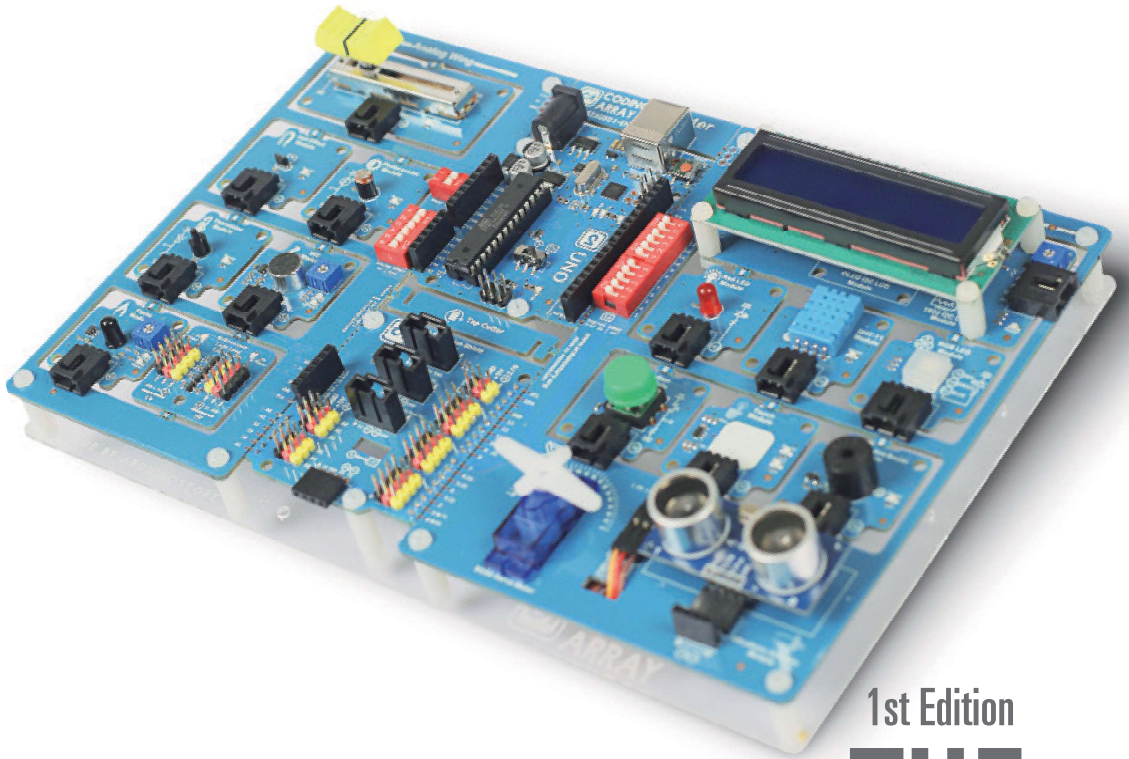


CODING ARRAY

STARTER KIT FOR ARDUINO

- Easier
- Faster
- Safer
- Simple
- Dual mode

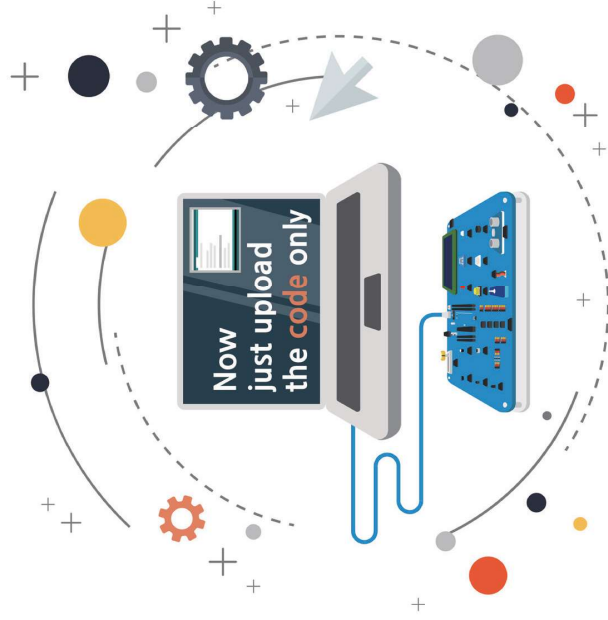


1st Edition
**THE
BEGINNER'S
GUIDE**

 Made By
Arduino Story

CODING ARRAY

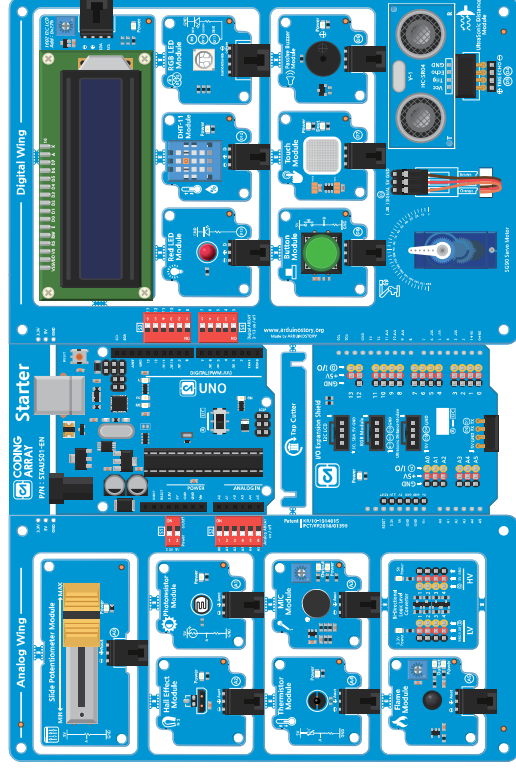
STARTER KIT FOR ARDUINO



Coding array kit is training toys designed to provide easy, fast and safe access to digital computing without the need for electrical engineering knowledge.

- Supplied from the coding array kit "Arduino Story" used in this guidebook.
- This guidebook is version 0.1 modified, translated, and written by Arduino story and owns the Arduino Story.

※ You may not modify, delete or distribute without permission from the owner, and may be subject to legal punishment in the event of a violation.



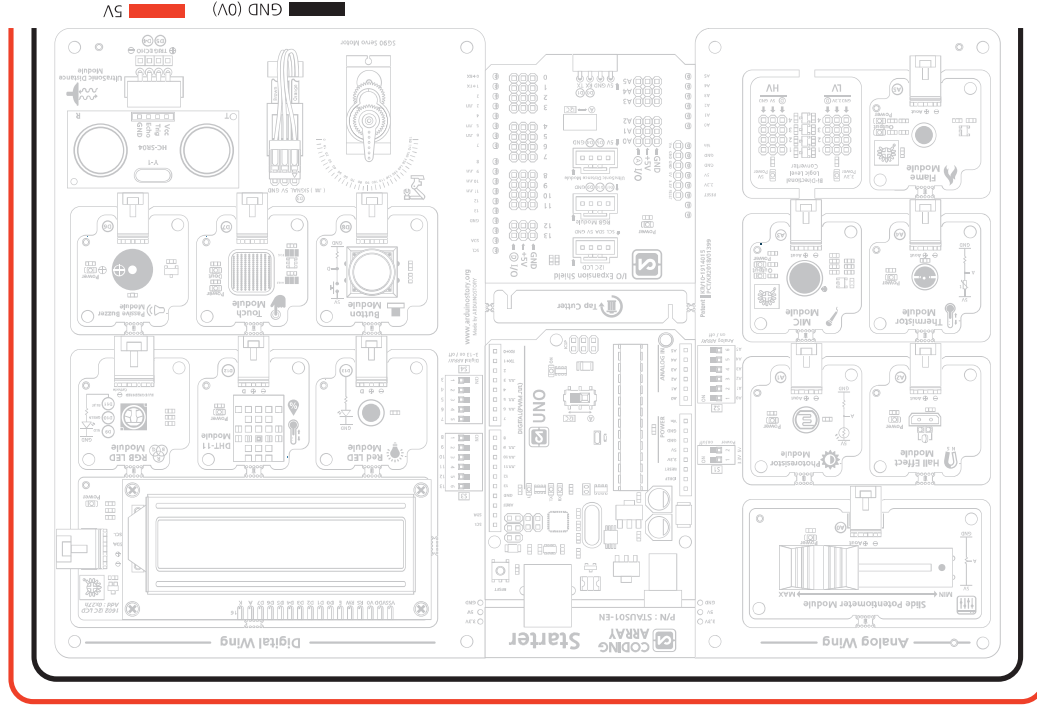
A suitable starter kit for training Arduino beginner is designed to be used without having to configure a separate circuit for 15 input and output units around Arduino Uno.

The modules are arranged in an analog wing on the left side of the Uno Board and a digital wing on the right. You can use and control modules at the same time as you upload sketch files.

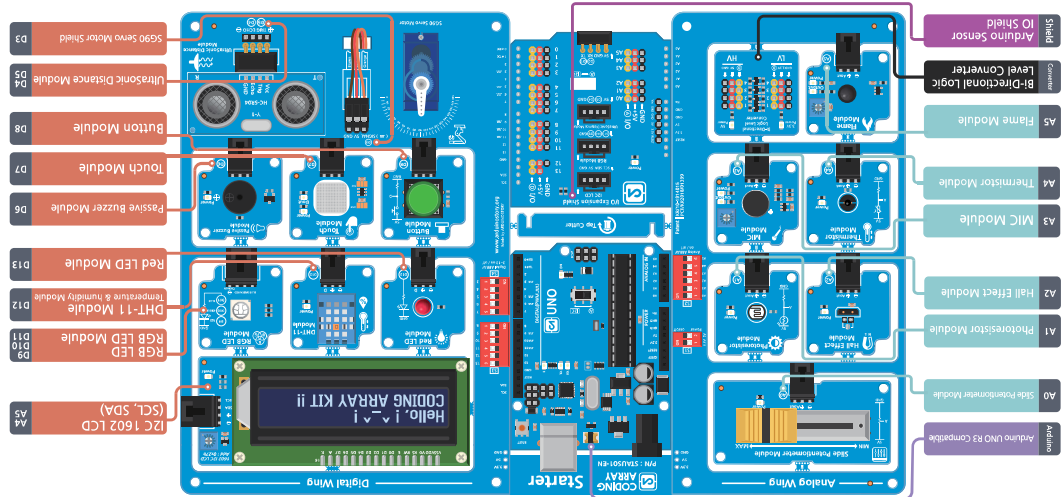
After learning, modules can be disconnected and have the advantages to use in the project.

CAUTION!

Do not disconnect the module using the Tap Cutter before board testing and learning is complete.



Coding Array Kit PINOUT

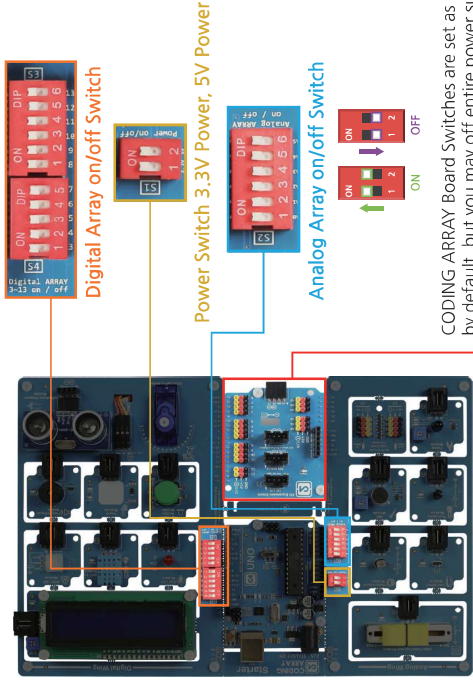


Actuators	Sensors
Slide Potentiometer	Hall Effect Module
Button Module	Photoresistor Module
RGB LED	Thermistor Module
Red LED	MIC Module
Passive Buzzer Module	1602 I2C LCD Module
5G90 Mini Servo Motor	Ultrasonic Distance Module
	Touch Module
	DHT-11 Module
	Temperature & Humidity Module

Platform	Shield	Converter
Microcontroller: Atmega328P	Arduino Sensor IO Shield	Bi-Directional Logic Level Converter
Operating voltage: 5V	DC Current per I/O Pin: Max. 20 mA	
Input voltage (limit): 6-20V	Analog Input Pins: 6	
Flash memory Size: 32 KB (Atmega328P) of which 0.5 KB used by bootloader	Digital I/O Pins: 14 (of which 6 provide PWM output)	
SRAM: 2 KB (Atmega328P)	Input voltage (limit): 6-20V	
EEPROM: 1 KB (Atmega328P)	DC Current for 3.3V: Max. 1000 mA	
LED_BUILTIN: 13		
Clock speed: 16 MHz		

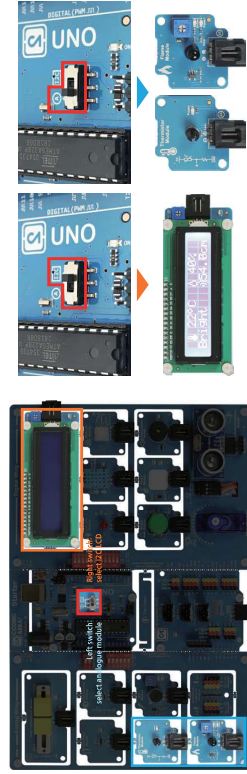
CODING ARRAY Kit's another advantage

To ensure the use of the coding array kit without damage, observe the following




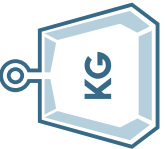
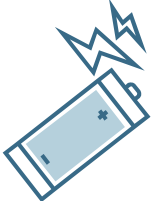
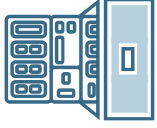
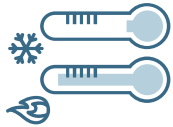
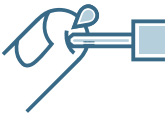
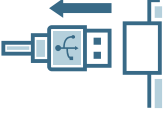


CODING ARRAY Board Switches are set as ON mode by default, but you may off entire power supply if necessary.

By default, aduino analog, digital I/O is connected to the I/O expansion shield. When the switch is turned off analog arrays, digital arrays, additional modules can be newly configured and used.



When shipping the product, it will be shipped to the 'I2C LCD Selection' position. The switch must be moved as shown in the Figure to use the Analog Temperature Sensor (A4) and Analog Fire Detection Sensor (A5) modules.

 <p>Anti-static Packaging Array kits are packaged in anti-static bags.</p>	 <p>Wet Hand Do not touch the product with wet hands.</p>	 <p>Chemicals Keep the product away from water or chemicals.</p>
 <p>Break Do not place heavy or sharp objects on the product.</p>	 <p>Conductive object Take care not to contact any conductive object, such as metal, in the circuit. Risk of damage to the circuit or burn or fire.</p>	 <p>Separate storage Be sure to keep the power separate when you are not using the product.</p>
 <p>Optimal Temperature Do not store and use at extreme hot or cold temperatures.</p>	 <p>Hand care The pins are sharp. Be careful of injuries.</p>	 <p>Disconnect power Turn off the power to connect to the board using a separate module.</p>

CODING ARRAY

STARTER KIT FOR ARDUINO
THE BEGINNER'S GUIDE
1ST EDITION

INDEX

Chapter 1. Prepare

1. Arduino? 12
2. Arduino IDE (Integrated Development Environment) Download 14
3. Arduino Software IDE Open 16
4. Menu Bar Function Overview 18
5. For Windows 20
6. For MAC 22

Chapter 2. Follow Example

1. IDE Structure – setup and loop 26
2. Serial communication 28
3. LED On/Off with Digital Output 30
4. Read button switch values with Digital Input 36
5. Change RGB LED Color Using Digital Output and PWM 42
6. Enables mood light with Electrostatic Touch Sensor 52
7. Read slide variable resistance value with Analog Input 60
8. Play melodies with manual buzzer 68
9. Displaying text on 1602 I2C LCD 76
10. Distance measurement with ultrasonic sensor 92
11. Magnet sensing with Hole sensor 98
12. Light intensity detection and calibrate sensor values with light sensing sensors 102
13. Flame Detection Sensor Detects Fire 116
14. Temperature measurement with NTC thermistor 122
15. Sound Detection with Microphone Sensor 130
16. Temperature and humidity measurement with sensors 136
17. Servo motor control 146

FAQ and solutions

153

CHAPTER 1

- PREPARE

Install Arduino IDE and download example code

Install the IDE installation, an integrated development environment, for use the Arduino board, and check use methods and function of the IDE. Learn how to download and open an example file for using a coding array kit..

1. Arduino?
2. Arduino IDE (Integrated Development Environment) Download
3. Arduino Software IDE Open
4. Menu Bar Function
5. For Windows
6. For MAC

1

CHAPTER 1

Arduino?

Arduino is a typical Microcontroller Unit (MCU) of an open source base (hardware + software) that is easy to use.

Microcontroller



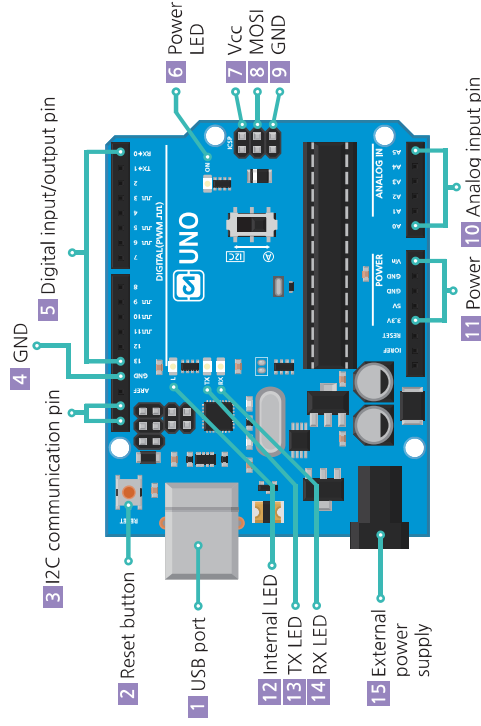
Microcontrollers are made of a single chip to perform a series of tasks in the CPU and memory, and to send the results to the I/O port with an electrical signal.

Arduino was designed in 2005 by Professor Massimo Banzai, who taught interactive design called 'physical computing' at Italy's Interaction Design Institute Ivrea (IDII) that taught art and IT. Professors and students used Parallax's Basic Stamp program, but it was expensive and inconvenient to use. The programming language developed by MIT is open-source and allows even those who are not well-programmed to implement graphics in simple codes, starting with Arduino's idea of how to make it easy for students who are not familiar with hardware to control.

The most used and documented board of Arduino is Uno, the best board to start Arduino. Uno means one in Italian and has been chosen to celebrate the release of Arduino IDE 1.0. The USB communication chip is built into the main processor, so it can be connected directly to the PC's USB, and has 14 digital input/output pins (including 6 PWM pins) and six analog input pins.

Microcontroller	ATmega328P
Dynamic voltage	5V
Input voltage (recommended)	7-12V
Input voltage (limit)	6-20V
Digital input/output pin	14 ea (including 6 PWM pins)
Analog input pin	6 ea
DC current per input/output pin	Max. 20 mA
DC current of 3.3V pin	Max. 1000 mA (150mA Existing Uno)
Flash memory size	32 KB (ATmega328P) 0.5 KB Include Bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock speed	16 MHz
LED_BUILTIN	13

< Uno-compatible board details used in array kits >



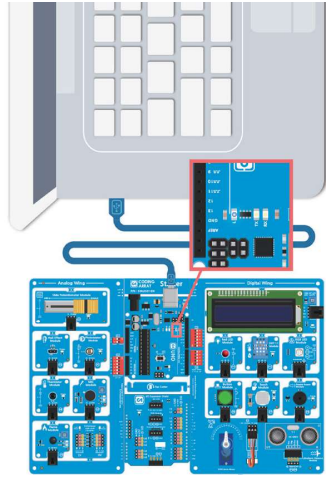
1	USB port	It receives a 5V power supply from the computer and performs serial communication.
2	Reset button	Restart button
3	I2C communication pin	SDA, SCL pins for I2C communication
4	GND	Grounding
5	Digital input/output pin	14 Digital input/output pin (including 6 PWM pins)
6	Power LED	Illuminates when power is supplied.
7	Vcc	Grounding
8	MOSI	Master output.
9	GND	5V Power supply
10	Analog input pin	6 Analog pin
11	Power	3-3V, 5V, Grounding, External power supply
12	Internal LED	Connect digital pin 13
13	TX LED	Indicates that the FTDI chip sends data to the computer.
14	RX LED	Indicates that the FTDI chip receives data from the computer.
15	External power supply	7V to 12V DC power supply. (Battery Pack)

2 CHAPTER 1

Arduino IDE (Integrated Development Environment) Download

STEP 01

Connecting a Board with Computer



Connecting the computer and Arduino with a USB cable allows users to receive 5V power supply to operate the Arduino as well as send and receive data. (However, separate power supplies shall be used for modules requiring a power supply greater than 5 V).

STEP 02

Access the Arduino Software Download Site from the internet



<https://www.arduino.cc/en/Main/Software>

STEP 03

Free download of Arduino Software IDE

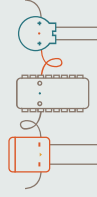
Download the Arduino IDE

Select and download the installation file that matches your computer's operating system.



Contribute to the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



\$3

\$5

\$10

\$25

\$50

OTHER

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Windows and Mac OS users run the program by selecting Download without Donation or Donation

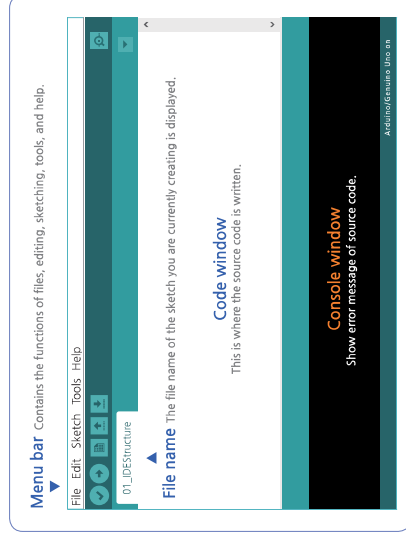


When a compressed file such as "arduino-1.8.5-linux64.tar.xz" is downloaded, Linux users run a terminal to decompress as follows:
Run the install.sh file in the extracted folder.

`tar xvzf [File name]`



Let's open the Arduino IDE installed on the computer and look at the configuration. IDE is largely divided into menu bars, tool bars, code windows, and console windows



- 1) Menu bar: Contains the functions of files, editing, sketching, tools, and help.
- 2) Tool bar: The most commonly used functions of the menu are buttoned together and have the following functions.

	Check : Check and compile code for grammar errors
	Upload : Send code to Arduino board. When the upload button is pressed, the lamp on the board flashes quickly.
	New File : Create a new file for creating a new sketch
	Open : Open saved Sketch file
	Save : Save the currently active sketch
	Serial monitor : Open a new window to show data communication between Arduino and the computer.

- 3) File name: The file name of the sketch you are currently creating is displayed.
- 4) Code window: This is where the source code is written.
- 5) Console window: Show error message of source code.

4

Menu Bar Function

CHAPTER 1

File	Editing	Sketch	Tool	Help
New file		When creating a new sketch		
Open		When opening a saved sketch		
Open Recent Files		View and open a list of recently opened sketches		
Sketchbook		When you open one of the sketches stored in the Sketchbook folder		
Example		When opening a built-in example, library example, and so on		
Close		Exit sketch		
Save		Save Sketch File		
Save As		Save Sketch Files As other name		
Page Setting		Page Setting window for printing		
Print		Print the current sketch		
Setting		Sketchbook folders, font sizes, compiler warnings, etc.		
Exit		Exit sketch at once, and open all when re-run		

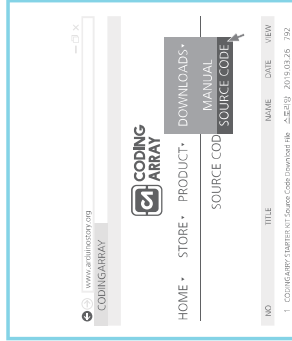
File	Editing	Sketch	Tool	Help
Cancel		Revert to Previous		
Redo		Re-run Cancel		
Crop		Cut Selection		
Copy		Copy Selection		
Copy for Forum		form for posting sources on Arduino Formal Forum notice		
Copy to HTML		Copy to Clipboard as HTML when you want to upload to a webpage		
Paste		Paste cut or copied part to cursor position		
Select All		Select All Code		
Go to line...		Shortcut to a specific code line		
Add/Delete Annotations		Process Annotations // When Displaying or Deleting Annotations		
Add indentation		When adding indentation		
Reduce indentation		When reducing indentation		
Increase Font Size		When increasing the size of the editor's font (Ctrl + over the mouse wheel)		
Decrease Font Size		When reducing the size of the editor's font (Ctrl + below the mouse wheel)		
Find		When looking for a specific character		
Find next		When you grow a specific character and find it in a later sentence		
Find previous		When you grow a specific character and find it in this previous sentence		

File	Editing	Sketch	Tool	Help
Check/compile		Check for code errors and compile		
Upload		Send Code to Arduino Board		
Upload using a programmer		Overwrite to a bootloader on the board		
Export compiled binary		Save as .hex file		
Show Sketch Folder		Open Current Sketch Folder		
Include Libraries		Use the #include to add libraries		
Add File		Add Source Files to Sketch		

File	Editing	Sketch	Tool	Help
Auto Format		Formatting code to look good		
Archive Sketches		Keep a copy of your sketch in a .zip file		
Modify Encoding & Refresh		Reduce encoding differences between editor and other editors where you see the communication data between Arduino and the computer.		
Serial monitor		Graphical representation of communication data between Arduino and computer		
Serial plotter		Send Code to Arduino Board		
WiFi101 Firmware Updater		Select the board you are using		
Board:"Arduino/Genuino Uno"		Choose the computer port to which the Arduino board is connected		
Port		Used to program boards or chips		
Get board information		When using IDE for a MCU other than Arduino		
Programmer : "AVRISP mkII"				
Burning a bootloader				

File	Editing	Sketch	Tool	Help
Getting started				
Environment				
Trouble shooting				
Reference				
Galileo Help				
Getting started				
Trouble shooting				
Edison Help				
Getting started				
Trouble shooting				
Find in Reference				
FAQ				
Visit Arduino.cc				
About Arduino				

Access and help with the various documents and other descriptions provided with Arduino IDE at www.arduino.cc



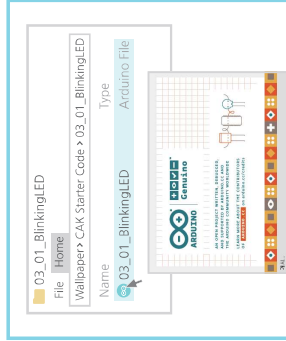
www.arduinoostory.org
DOWNLOADS -> SOURCE CODE
 Access the link above,
 download the sample file

1. Downloading an example file



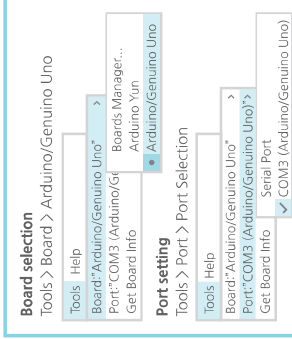
Run a downloaded ZIP
 file and specify the
 folder to be saved as the
 source code.

2. Extracting



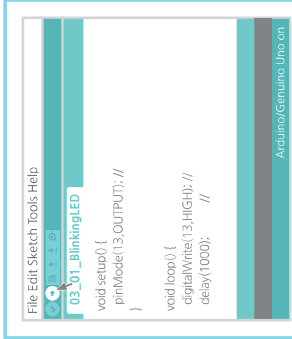
Select and run a learning
 sample file in Custom
 folder ▶ CAK Starter Code.

3. Open Example File1



After a round of connecting/disconnecting
 USB line with IDE program, the board
 shows which port is now being used.
 The port number might differ,
 depending on individual setting.

5. Select Board and Set Ports

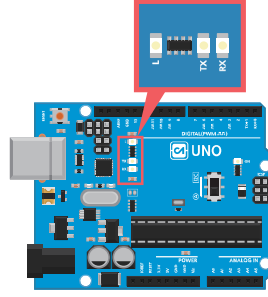


Click to upload Sketch,
 then you may be able to view the outcome.

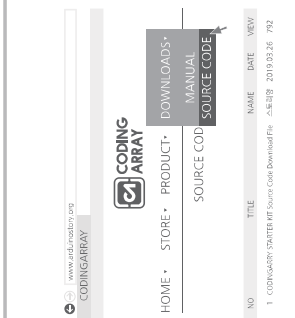
6. Open Example File3



If a message is displayed at the top
 of the IDE console window, it means
 that the upload was successful.



During upload, the TX,RX,LED on the
 onboard flashes quickly and then turns off.



www.arduino.org
DOWNLOADS -> SOURCE CODE
 Access the link above, download the sample file


1. Downloading an example file



Save As: CAK Starter Code.zip
 Tags: Downloads
 FAVORITES: Desktop, Documents, Downloads, iCloud Drive, DEVICE user, Macintosh
 [New Folder] [Cancel] [Export]

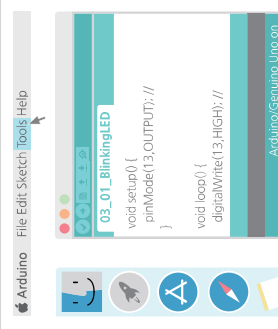
Specify the folder to be saved as the source code.

2. Extracting



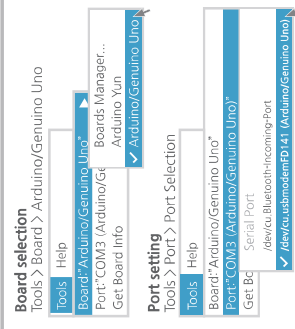
Select and run a learning sample file in Custom folder ▶ CAK Starter Code.

3. Open Example File1



Once you execute your sample file, Sketch file is opened as follows. Please proceed with desired board selection and port setting on the top menu bar 'Tools' before uploading Sketch file into the board.

4. Open Example File2




Board selection
 Tools > Board > Arduino/Genuino Uno
 Tools Help
 Board: "Arduino/Genuino Uno"
 Port: "COM3 (Arduino/Genuino Uno)"
 Get Board Info

Port setting
 Tools > Port > Port Selection
 Tools Help
 Port: "COM3 (Arduino/Genuino Uno)"
 Get Board Info
 Serial Port: /dev/cu.usbmodem1411 (Arduino/Genuino Uno)


After a round of connecting/disconnecting USB line with IDE program, the board shows which port is now being used. The portnum bermight differ, depending on individual setting.

5. Select Board and Set Ports

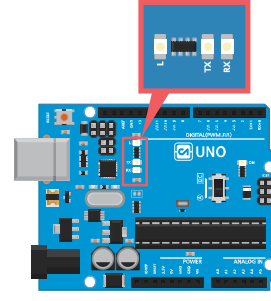


```

03_01_BlinkingLED
void setup() {
  pinMode(13,OUTPUT); //
}
void loop() {
  digitalWrite(13,HIGH); //
}
Arduino/Genuino Uno on
  
```

Click  to upload Sketch, then you may be able to view the outcome.

6. Open Example File3



During upload, the TX:RX LED on the Unoboard flashes quickly and then turns off.



```

03_01_BlinkingLED
void setup() {
  pinMode(13,OUTPUT); //
}
void loop() {
  digitalWrite(13,OUTPUT); //
}
  
```

Done uploading.
 Done uploading.
 Sketch uses 930 bytes (2%) of program memory.
 Arduino/Genuino Uno on

If a message is displayed at the top of the IDE console window, it means that the upload was successful.

CHAPTER 2

- FOLLOW EXAMPLE

Shows the characteristics of the module used in the example file supplied with the board, as well as the default usage code and results.

Example contents

1. IDE structure – setup and loop
2. Serial communication
3. Turn LED on and off with digital output
4. Read button switch values with digital input
5. Change RGB LDE color using digital output and PWM function
6. Implementing moods, etc. with capacitive touch sensors

CODING ARRAY STARTER Kit for Arduino

THE BEGINNER'S GUIDE 1st Edition

IDE structure – setup and loop

Running Arduino IDE creates a sketch file consisting of two parts: void setup and void loop.

IDE structure – setup and loop



- `void setup () { }`
- Runs only once when the program starts.
- Declares variables between medium brackets, includes pinMode settings, etc.

- `void loop () { }`
- The main content of a program that runs repeatedly between medium brackets.
- Even if there are no repeat statements, the medium brackets shall remain blank.



CAK Starter Code > 01_ IDEStructure



Let's run the IDE on Arduino and open an example above.

There are two main screens, void setup {} and void loop {}.

Powering up Arduino will execute the previously uploaded code, which can act as a clearing of the previous code.

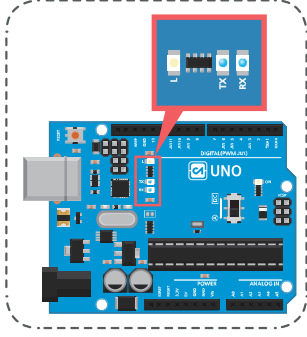
```

1 void setup() {
2   // Put the code that runs only once, such as variable declaration and pinMode, in the
3   medium brackets.
4 }
5
6 void loop() {
7   // Put in the main code that runs repeatedly..
8   // If there is no repeat statement, the parentheses should be left blank.
9
10

```

! Precautions for creating a sketch file

- When you create a sketch file, make sure to write case-sensitive characters.
- At the end of the command statement, a semicolon (;) should be added at all times.
- Comments are part of the program that does not affect the program
- One-line annotation (// content) and multi-line annotation (/* content */).



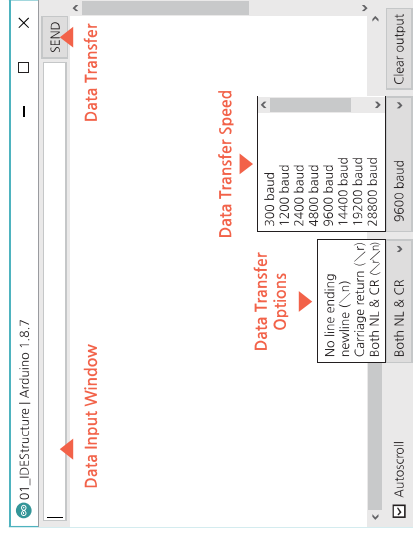
The USB cable allows data to be exchanged between Arduino and the computer, called serial communication (serial communication, UART communication). When uploading sketch files, you can see the RX (:data received) and TX (data sent) lights on the board flickering quickly. Since Arduino's digital No. 0 pin (RX) and Digital No. 1 pin (TX) are used for serial communication, use Pin 2 to connect modules to the digital pin.

Serial communications make it easy to debug the computer to give data to Arduino, check the program results value of Arduino through the computer window, or find and correct errors in the program.

After uploading the sketch file, touch the same icon to the right to display a serial monitor pop-up window.

CAUTION!

do not open the serial window during program upload.



CAK Starter Code > 02_Serial

This sketch shows how to print messages entered on a computer into a serial window.

```

1 void setup() {
2   Serial.begin(9600); // Prepare serial communication. Set the communication speed to 9600.
3 }
4
5 void loop() {
6   Serial.println("Hello Coding Array Kit~!"); // Print Hello Coding Array Kit~! in the serial window.
7   delay(1000); //Wait for 1000 milliseconds (=1 second).
8 }
9

```

Serial.begin (communication speed); the Baud rate is between 300 and 115200. It is usually set to 9600.

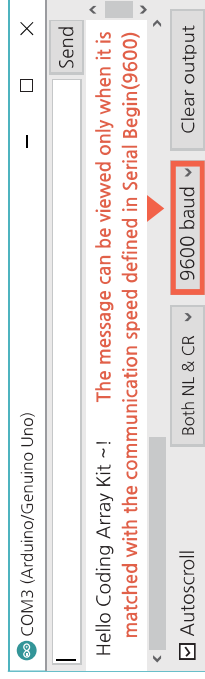
Serial.println (value, format); to change the line after printing the value on the serial monitor.

Values can contain both letters and numbers to be printed. However, the letters must be in between ' or ' .

- Serial.println("XY") outputs A
- Serial.println(3,14159, 0) outputs 3
- Serial.println(3,14159, 2) outputs 3.14.

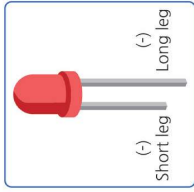
Serial.print (value, format); add values to the serial monitor without changing lines. Format option **delay (milliseconds);** delay the command by milliseconds. 1000 milliseconds = 1 second

Observation of results

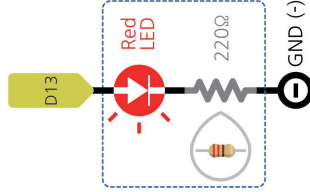


Turn LED on and off with digital output

LED (Light Emitting Diode)



LEDs are semiconductor devices that emit light when current flows through LEDs. LEDs are of type lamp (lead) and surface seal (SMD) type. The red LED used in the array kit is a lamp type and has two legs. A long pin (+) connects to Arduino's No. 13 pin, and a short pin (-) connects to Arduino's GND (ground)



Since the operating voltage of the LED used is 1.6 to 2 volts, the module is also equipped with a resistance (220 ohms) that limits the current at 5 volt supply

CAUTION: ★

LED's do not illuminate when connected with (+) (-) polarity changes.

CAUTION: ★

If the LED is powered without resistance, it will not work or reduce its service life.

LEDs are smaller in size and longer in life compared to light bulbs or fluorescent light bulbs, and use less power, but produce brighter light. It is often used for portable flashlights, lights, billboards, car lights, flat-screen TVs and monitors. Two or more LED lights can be used to implement a beam walker signal lamp or to indicate the device's on/off indicator.

Let's find out about digital output

Arduino Uno's digital input/output/pin

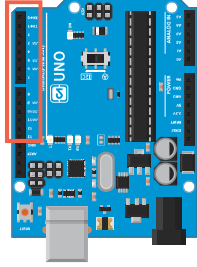


Arduino's input and output signals are largely divided into two categories: digital and analog

Digital signals refer to the high voltage of two signals, HIGH / LOW, 1 / 0, True / False, and On / Off.

First, let's learn about digital output signals.

Arduino has 14 digital I/O pins from 0 to 13. However, since 0 and 1 are connected to the computer, it is preferable to use pins 2 if possible.



Digital input digitalWrite (Pin Number, HIGH);

HIGH	LOW
1	0
5V	0V
On	Off
True	False

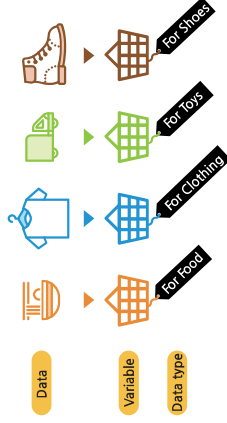
Since digital pins are specified by default as input pins, when used as output pins, the setup declares the output as pinMode (pin number, OUTPUT). digitalWrite (pin number, HIGH) after declaration: can command 5V output to pin number or digitalWrite (pin number, LOW) to 0V output to pin number..

Let's find out about variables

Variables

Variable means the name or space itself of a space that stores values that can change during a program to process or store data.

In order to define a variable in C programming, the data type for the value that goes into the variable must be declared together.



For numeric data types, appropriate numeric data types should be declared according to the size of the data. If the data type is incorrectly declared, the desired result value cannot be obtained.

If a variable is declared at the top of the previous program, it becomes a global variable that can be used in all parts of the program

Variable data type

Type	Scope	byte	Use
void			Function declaration, used when return value is missing ex.) void setup() {} void loop() {}
boolean		1	Use only true or false values ex.) boolean state= true ;
char	-128~127	1	Save character values, one character value is enclosed in single quotes, and stored as ASCII code (number) values. The two examples below store the same values. e.g.) char myChar= 'A'; char myChar = 65; Multiple characters are enclosed in double quotes. e.g.) char array[]="ardinostry"
unsigned char	0~255	1	Same as byte data type. Byte data type is preferred.
byte	0~255	1	Similar to char, but having a positive integer value.
int	-32768~32767	2	Basic data type for storing integers with symbols If data is out of range, it will result in unexpected values and should be replaced with double or long.
unsigned int	0~65535	2	Use for positive integer values
word	0~65535	2	Use for positive integer values
long	-2147483648 ~ 2147483647	4	Use for integer values in a range greater than int
unsigned long	0~4294967295	4	Use for positive integer values in large ranges
short	-32768~32767	2	Use for integer values
float	-3.4028235E+38 ~3.4028235E+38	4	True (numeric) data type
double	-3.4028235E+38 ~3.4028235E+38	4	In Arduino, the same data type as float

CAK Starter Code > 03_01_BlinkingLED

```

1 void setup() {
2   pinMode(13,OUTPUT);           // Set 13 to output pin
3 }
4
5 void loop() {
6   digitalWrite(13,HIGH);       // Give digital signal 1 (HIGH) to pin 13. LED illuminated
7   delay(1000);                 // Wait for 1000 milliseconds (=1 second).
8   digitalWrite(13,LOW);        // Give digital signal 0 (LOW) to pin 13. LED Off
9   delay(1000);                 // Wait for 1000 milliseconds (=1 second)
10 }

```

pinMode(pin number, value);

pin number puts the digital pin number of Arduino, where the (+) pole of the LED is connected. The value specifies the input/output role of the pin. You can give INPUT or OUTPUT or INPUT_PULLUP.

digitalWrite(pin number, value);

can give HIGH (5V) or LOW (0V) digital output value to parts connected to pin number.

CAK Starter Code > 03_02_BlinkingLED2

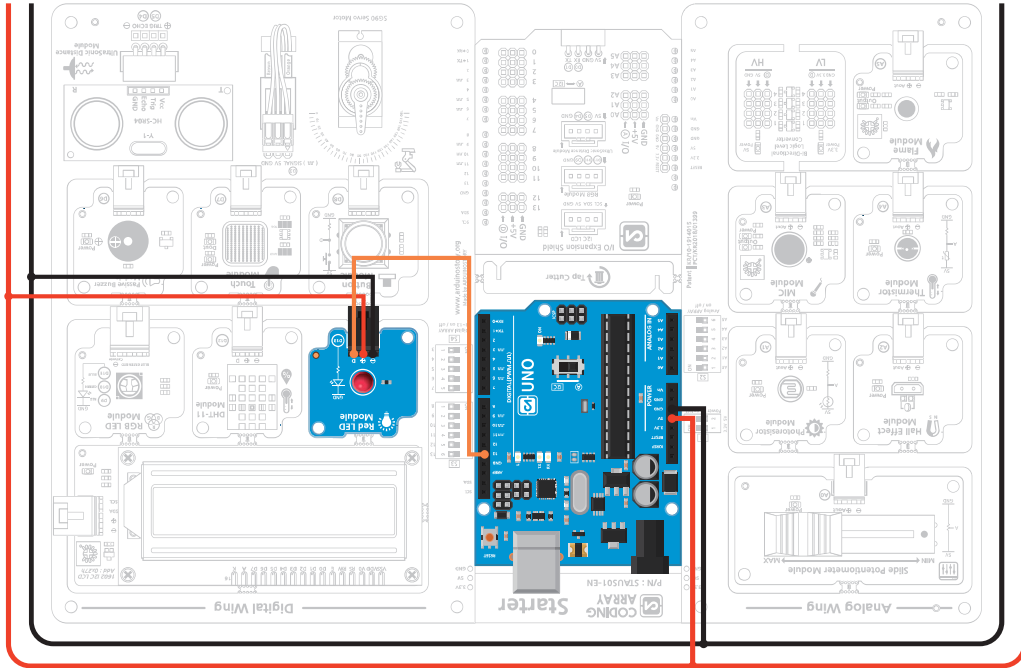
```

1 int redLED = 13;               // Red LED to 13
2
3 void setup() {
4   pinMode(redLED,OUTPUT);     // Set No. 13 to output pin
5 }
6
7 void loop() {
8   digitalWrite(redLED, HIGH); // Give digital signal 1 (HIGH) to pin 13. Red LED illuminated
9   delay(1000);                 // Wait for 1000 milliseconds (=1 second)
10
11  digitalWrite(redLED, LOW);  // Give digital signal 0 (LOW) to pin 13. Red LED Off
12  delay(1000);                 // Wait for 1000 milliseconds (=1 second).
13 }

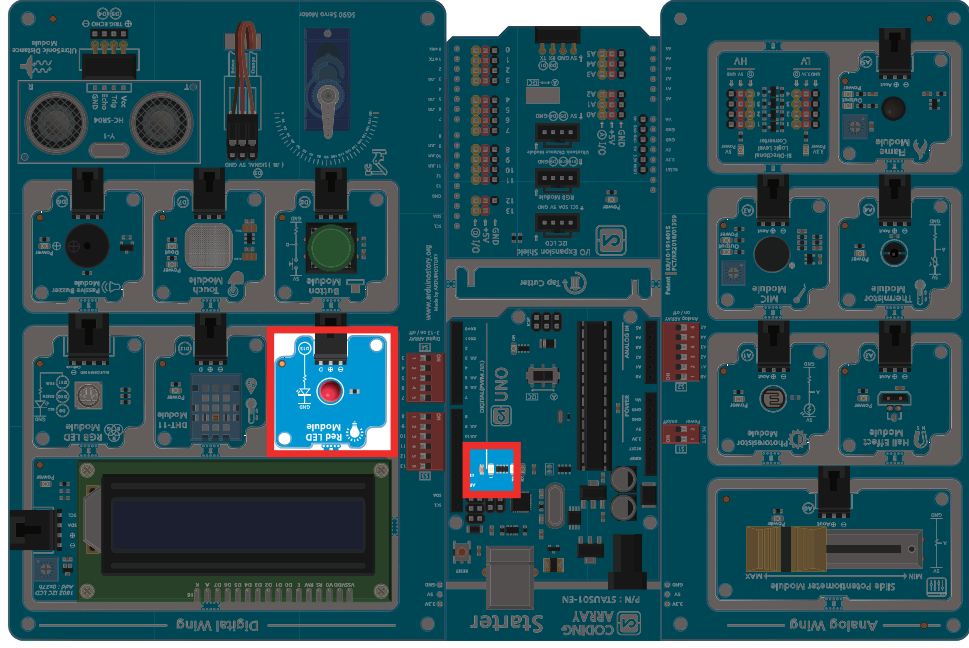
```

int variable name = value;

int is a data type that stores integers. The redLED variable stores pin number 13 and can be represented using variable names instead of pin numbers.



Let's set the red LEDs connected to pin 13 to the output and repeat the execution of the LED turning on for 1 second and off for 1 second, depending on the time settings of the digitalWrite function indicating the digital output and the delay function..



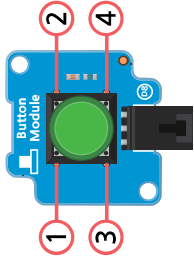
Once the program is uploaded, depending on the digitalWrite function representing the digital output and the time setting of the delay function, the LED can be repeatedly lit for one second and then turned off for one second. At this point, you can see the LED with built-in on the 13th Uno Board flickering. Different delay function times can control the flashing speed of LEDs.

View Results

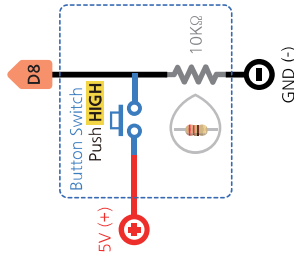
Read button switch values with digital input

Push Button

The button switch is used to open and close the circuit by pressing the button at the top. When a button is pressed, a circuit is connected and an electrical current flows between 1,2 and 3,4. Therefore, when connecting wires, select one from 1,2 and one from 3,4. Button switches are used in everyday life such as game consoles, bus exit notification buttons, keyboard buttons, etc. as well as control of opening and closing circuits.



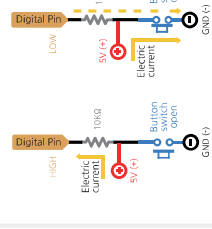
When the circuit of the button switch is open, the Arduino board cannot logically predict the HIGH, LOW for the pin that is not connected, resulting in a floating phenomenon that moves high and low fast. To prevent floating, a button switch can be received as a digital input after a pull-up or pull-down resistance is hung.



The coding array starter kit is connected to a button switch using a pull down resistance that defines the LOW voltage in normal situations without a drive signal. Therefore, the HIGH (1) value is entered when the button switch is pressed and the LOW (0) value is entered if the button switch is not pressed.

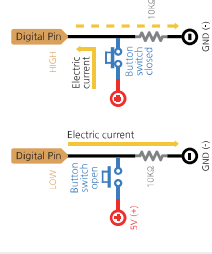
Pull-up circuit

Connect the normal state of the input pin to HIGH. With the resistance connected to power 5V, HIGH returns 1 in `digitalRead`.



Pull-Down circuit

Connect the normal state of the input pin to LOW. With the resistance connected to GND 0V, HIGH returns 1 in `digitalRead`.



CAK Starter Code > 04_01_Button

```

1 /* If condition statement is used
2 * Press button switch to connect circuit and return HIGH (1) value to light LED
3 * If the button switch is not pressed, the circuit will open and the LOW (0) value will be returned
4 * to turn off the LED.
5 */
6
7 int redLED = 13; // Set red LED pin to 13.
8 int Button = 8; // Button switch pin set to 8
9
10 void setup() {
11   pinMode(Button, INPUT); // Set button pin to input
12   pinMode(redLED, OUTPUT); // Set redLED pin to output
13
14   Serial.begin(9600); // Start's serial communication at 9600 speed
15
16 }
17
18 void loop() {
19
20   int sensorVal = digitalRead(Button); // Receive button input value for sensorVal variable.
21   // Variables inserted in void function are regional variables
22
23   Serial.println(sensorVal); // Mark the value of the button one line in the serial window.
24   // If the pull-down resistance is connected, give LOW (0) when the button is open and HIGH (1)
25   // when pressed.
26   // give HIGH (1) value when the button is open and LOW (0) when pressed.
27
28   if (sensorVal == LOW) { // If the button is open,
29     digitalWrite(redLED, LOW); // RedLED OFF
30   }
31   else { // If the button is pressed,
32     digitalWrite(redLED, HIGH); // Turn on the redLED.
33   }
34   delay(10);
35 }

```

digitalRead(pin number, value): The voltage entering the pin is read by HIGH (1) EH as a LOW (0) digital input value. It means that the left and right values are the same.

==