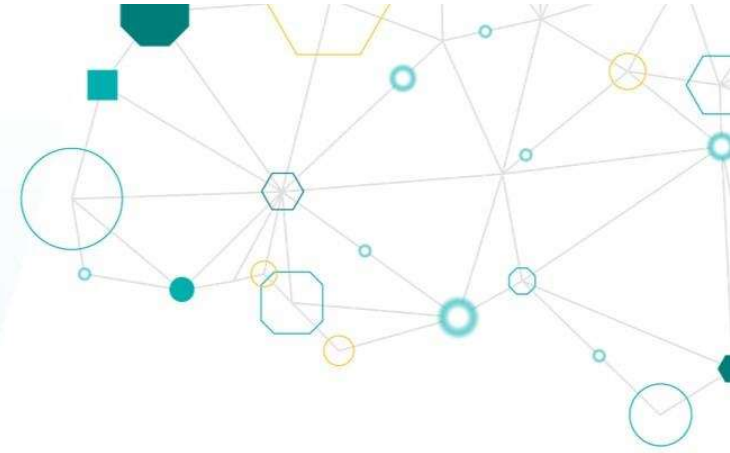
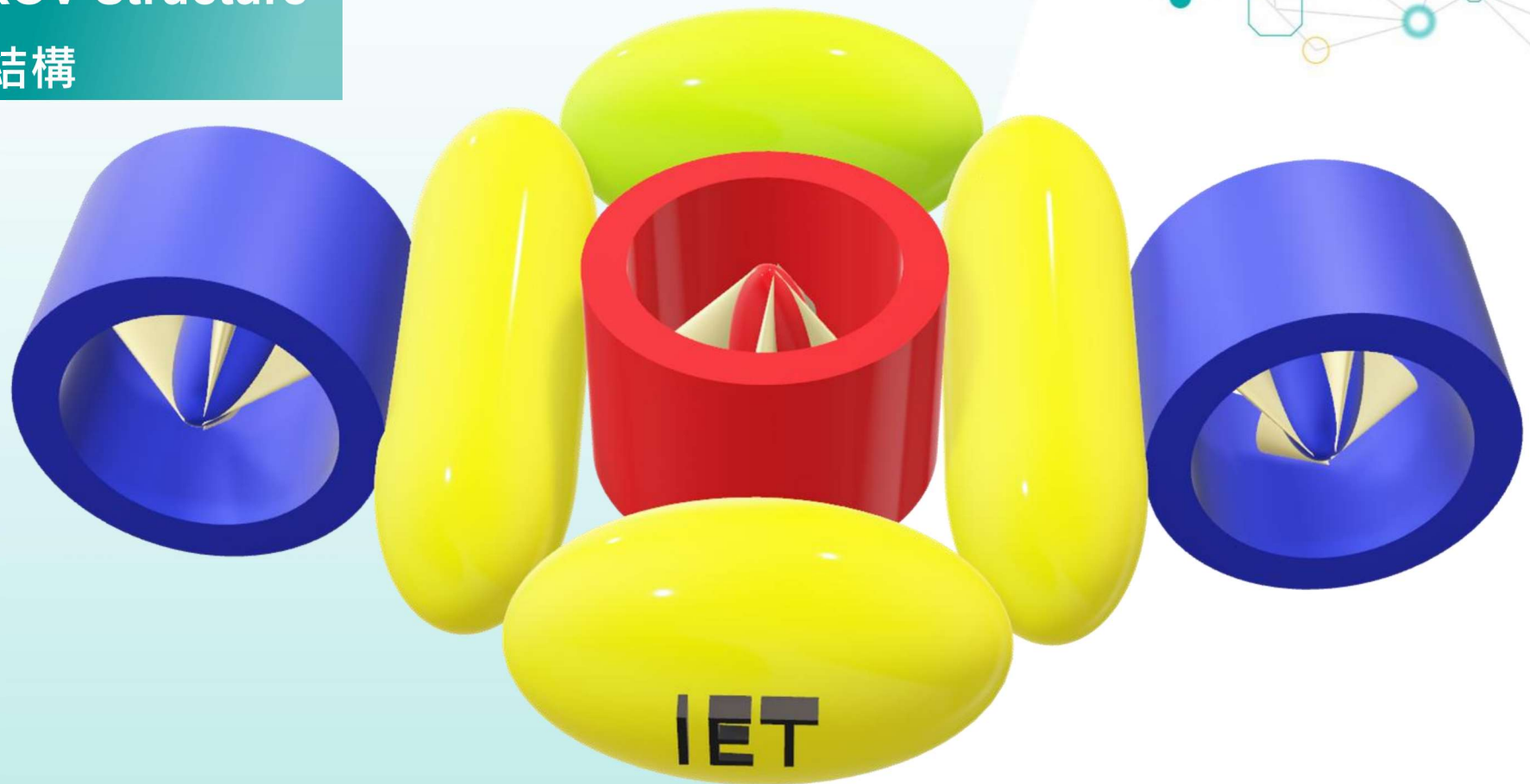


THE IET/MATE HONG KONG REGIONAL - UNDERWATER ROBOT CHALLENGE



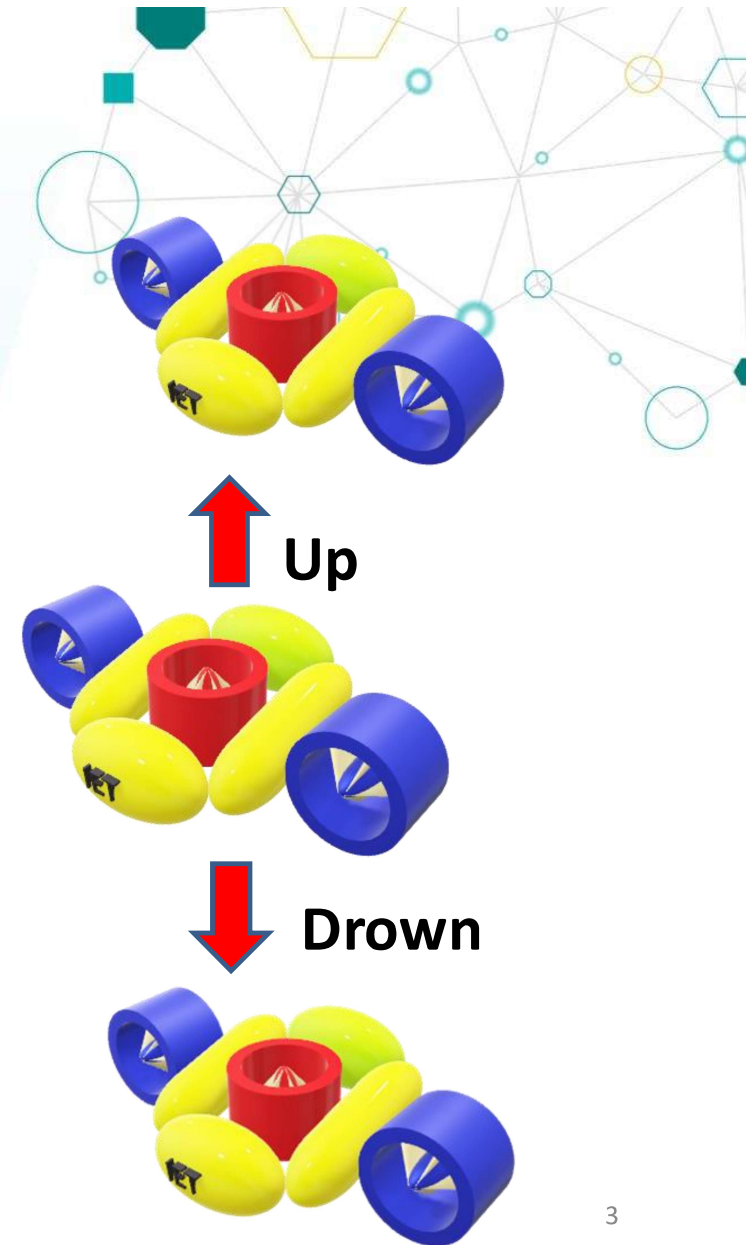
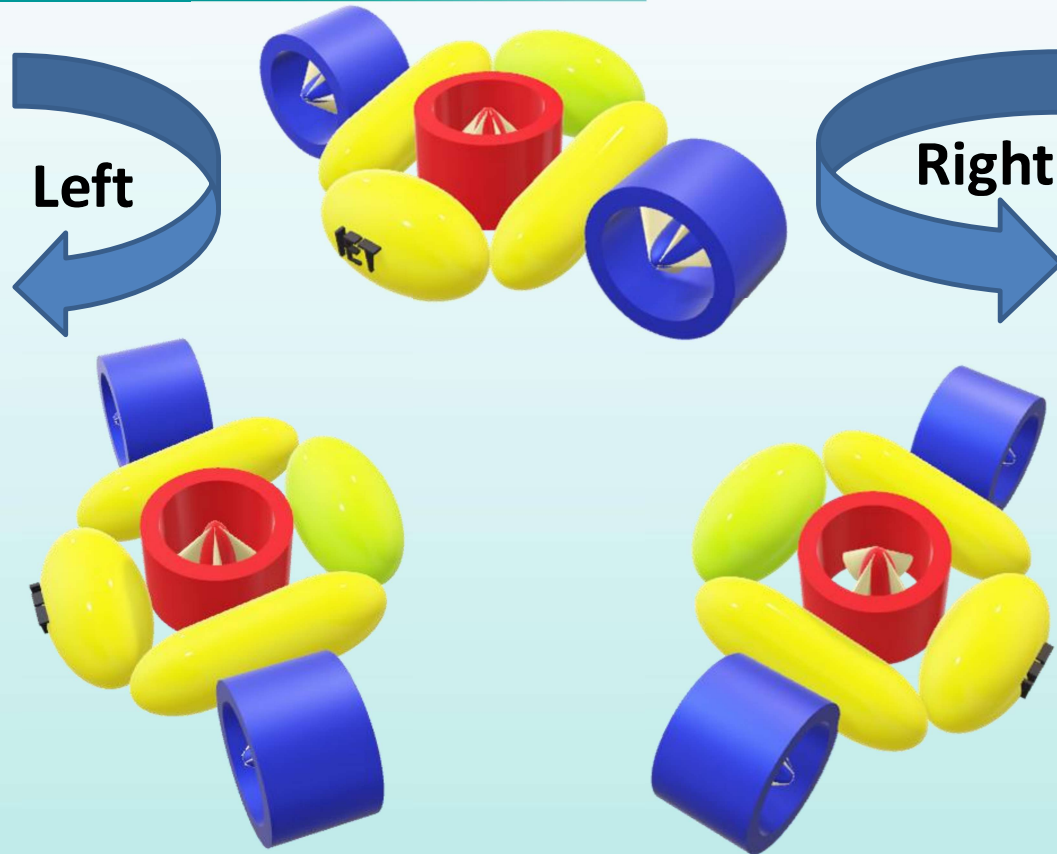
General Programming Workshop

Simplified ROV Structure 簡化的ROV結構



Robot 3D Control

機器人3D控制

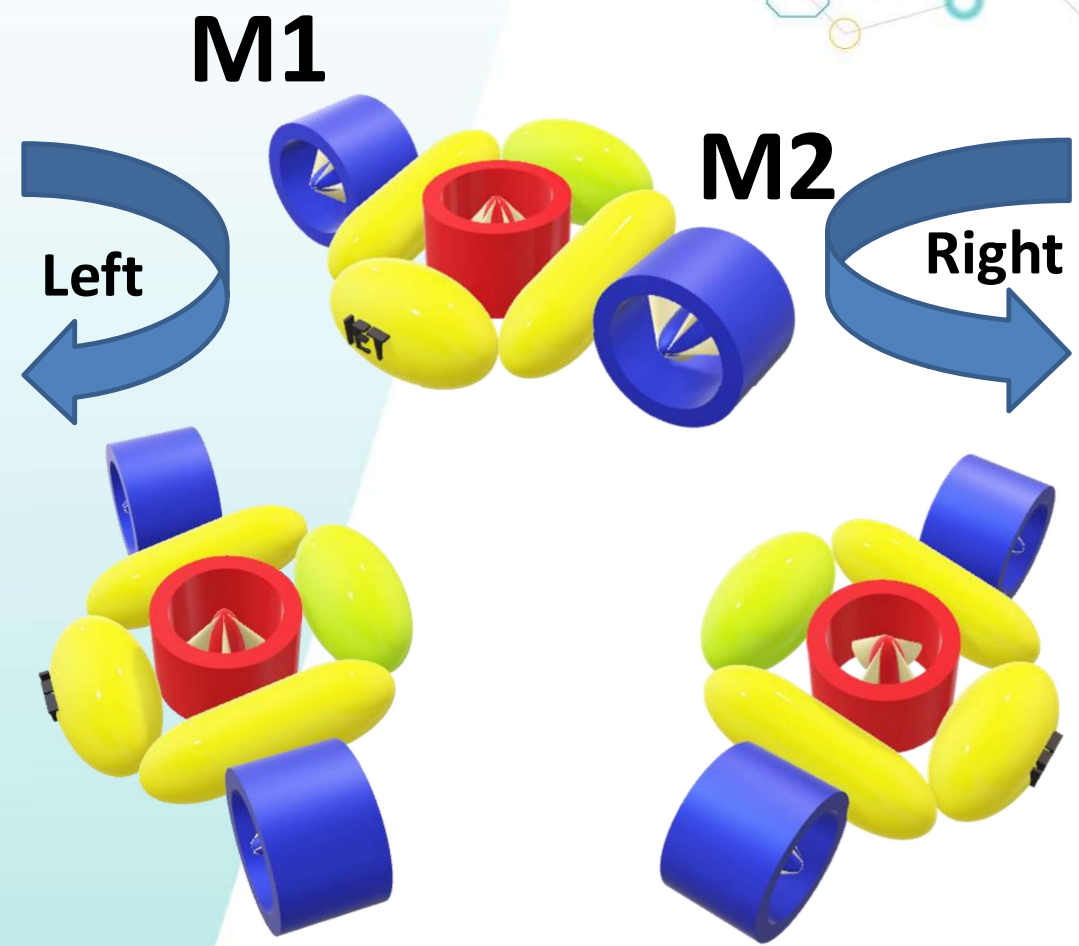




Robot 3D Control

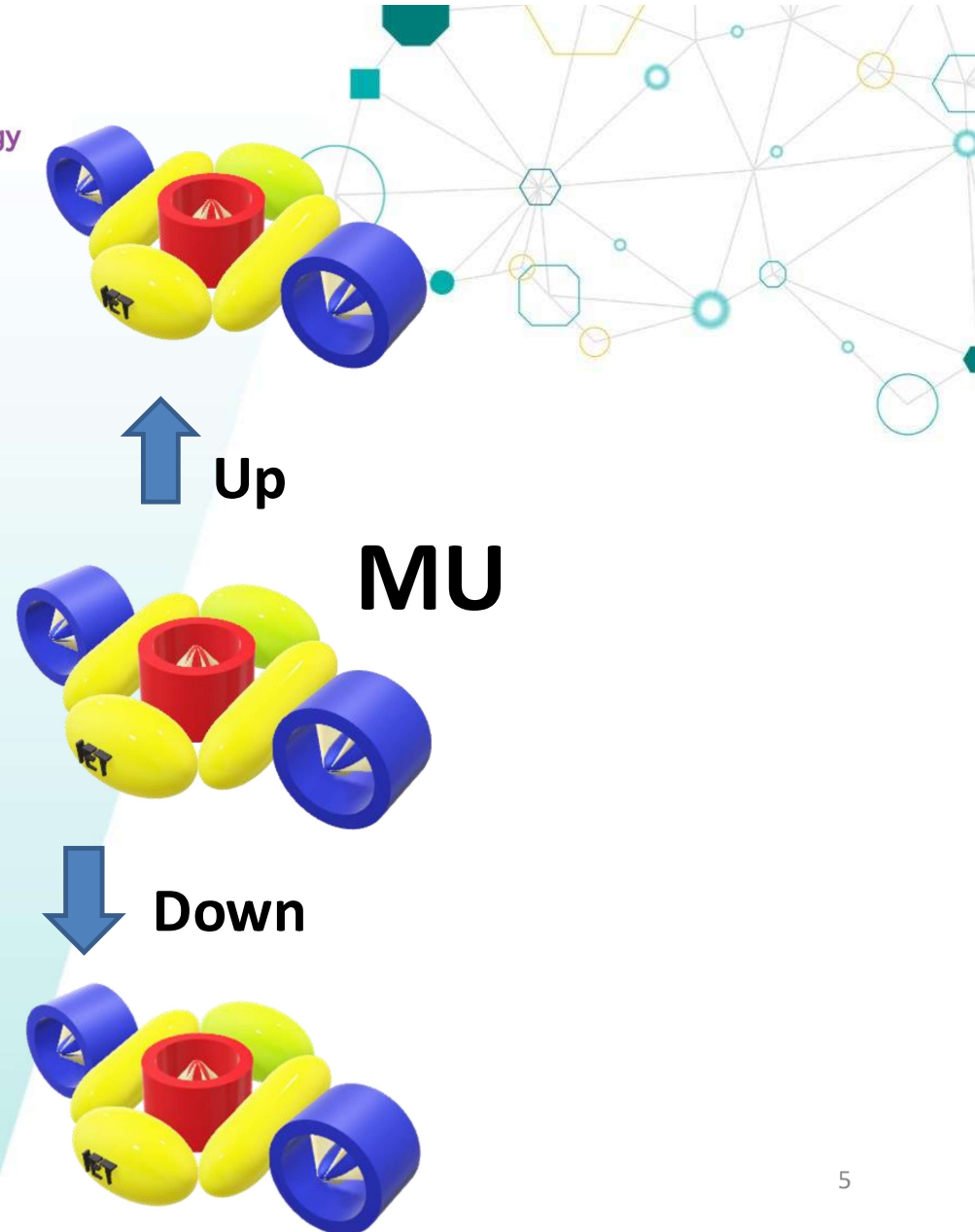
機器人3D控制

Turning 轉向	M1	M2
Right 右	Backward 後退	Forward 向前
Left 左	Forward 向前	Backward 後退

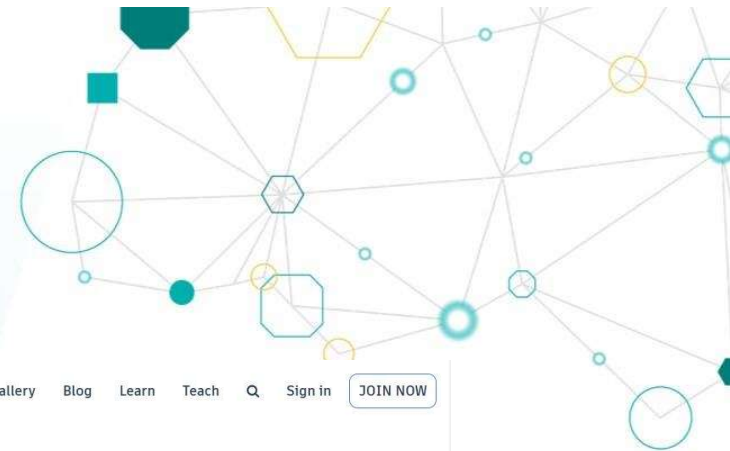


Robot 3D Control 機器人3D控制

Upthrust 上推	MU
Right 右	Backward 後退
Left 左	Forward 向前



Tinkercad

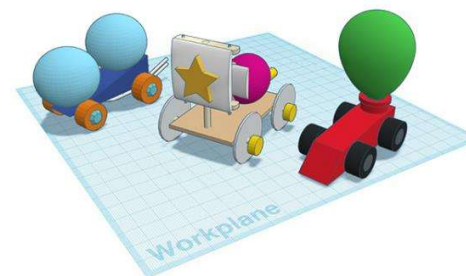


From mind to design in minutes

Tinkercad is a free, easy-to-use web app that equips the next generation of designers and engineers with the foundational skills for innovation: 3D design, electronics, and coding!

Start Tinkering

Join your class



Design a Balloon Powered Car Try it! < || >



Community of 35 million

Join



Fast, free, easy to use

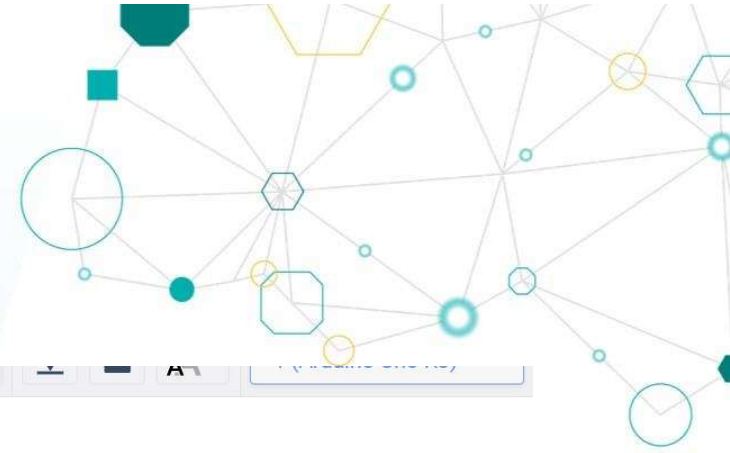
Learn



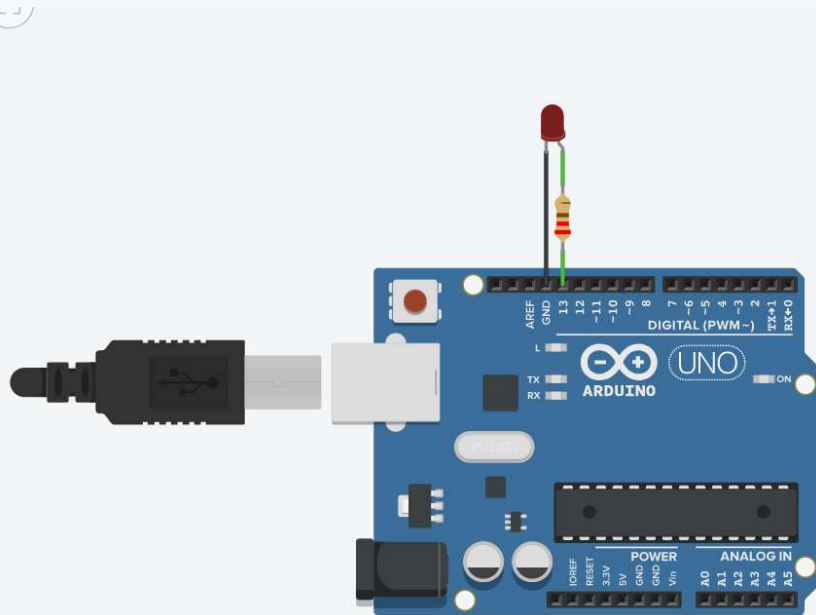
Loved by educators worldwide

Teach

Safeguarding your privacy is fundamental to our mission. Learn how we protect student data in our [Privacy and Security Statement](#)



Arduino Introduction (Turn on LED)



```

1 // C++ code
2 //
3 /*
4   This program blinks pin 13 of the Arduino (the
5   built-in LED)
6 */
7
8 void setup()
9 {
10   pinMode(LED_BUILTIN, OUTPUT);
11 }
12
13 void loop()
14 {
15   // turn the LED on (HIGH is the voltage level)
16   digitalWrite(LED_BUILTIN, HIGH);
17   delay(1000); // Wait for 1000 millisecond(s)
18   // turn the LED off by making the voltage LOW
19   digitalWrite(LED_BUILTIN, LOW);
20   delay(1000); // Wait for 1000 millisecond(s)
21 }

```

Serial Monitor

Arduino Introduction (Turn on LED)

TINKERCAD
AUTODESK
Tinkercad

Gallery Blog Learn Teach Q



燦文 陳

Search designs...

3D Designs

Circuits

Codeblocks

Lessons

Your Classes

Collections

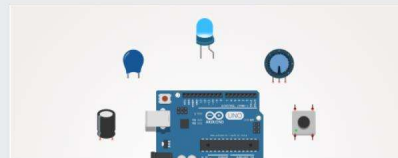
+ Create collection



Educators: If you've ever heard "This is cool! What's next?" while teaching with Tinkercad, here's how to guide your students to the next step when they're ready for professional - grade tools. [Learn more](#)

Circuits

Create new Circuit



Tinker with Circuits on Tinkercad!

Try Circuits

About Us

Support

Follow us

AUTODESK

Arduino Introduction (Turn on LED)

TINKERCAD Turning On LED

All changes saved

Code Start Simulation Send To

Components Basic

Search

9V Battery Coin Cell 3V Battery

1.5V Battery Breadboard Small

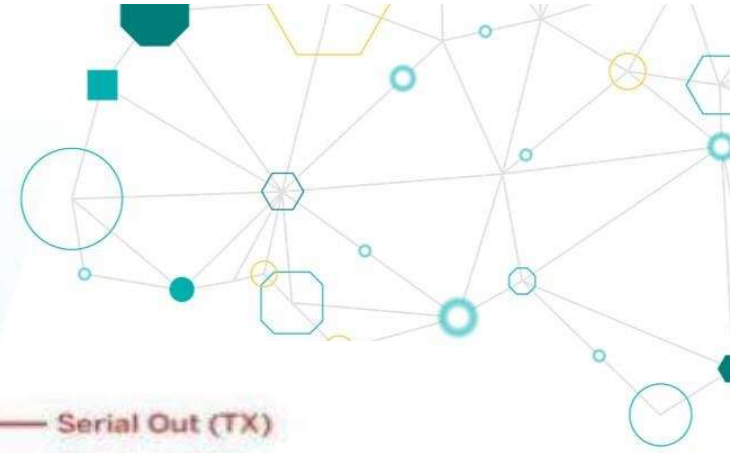
micro:bit Arduino Uno R3

Vibration Motor DC Motor

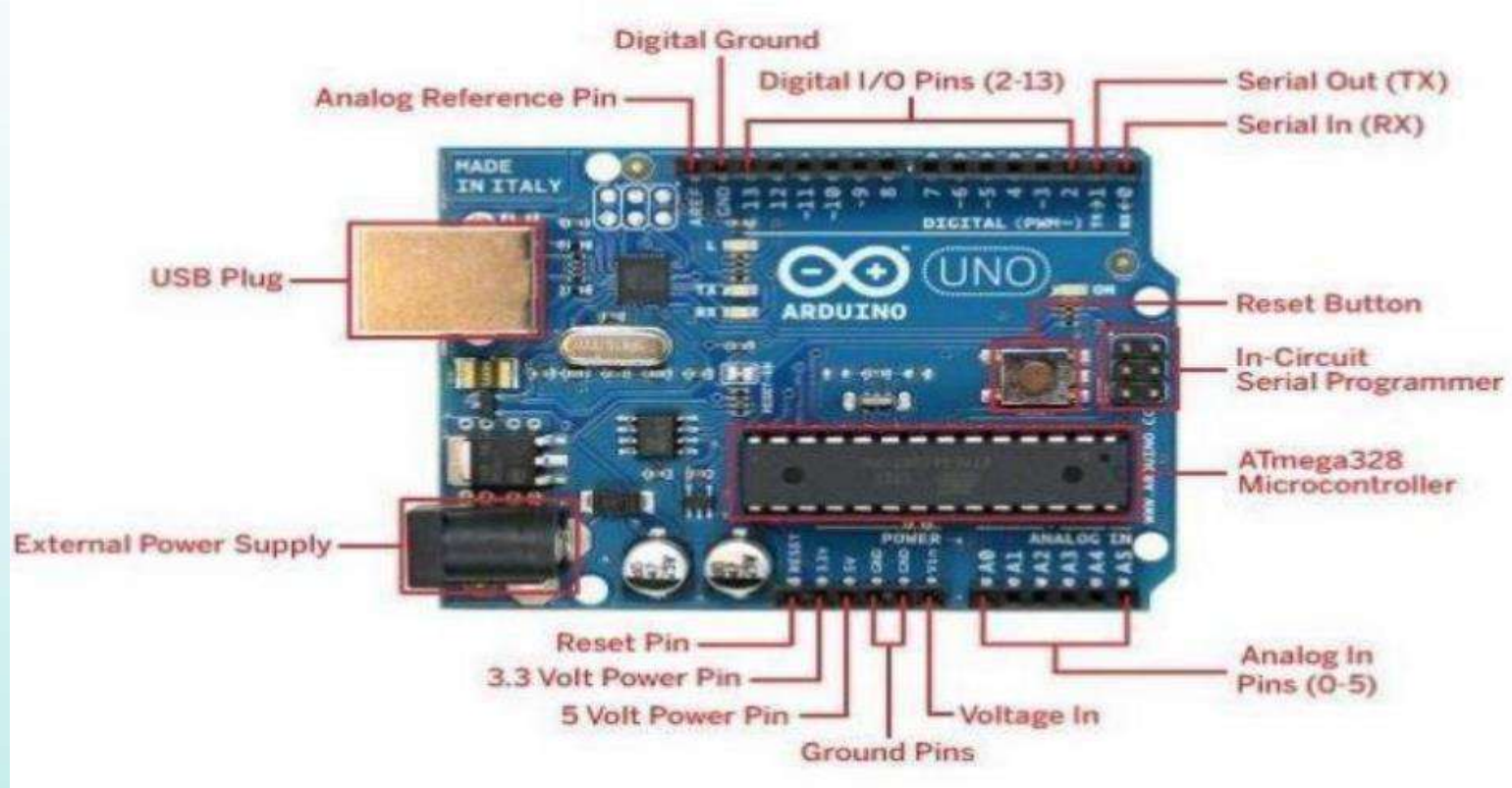
NEW micro:bit component

Add a micro:bit to your design to begin. Code it with blocks and Simulate to see it work. Look in Starters for examples to build with.

<https://www.tinkercad.com/dashboard?type=circuits&collection=designs>

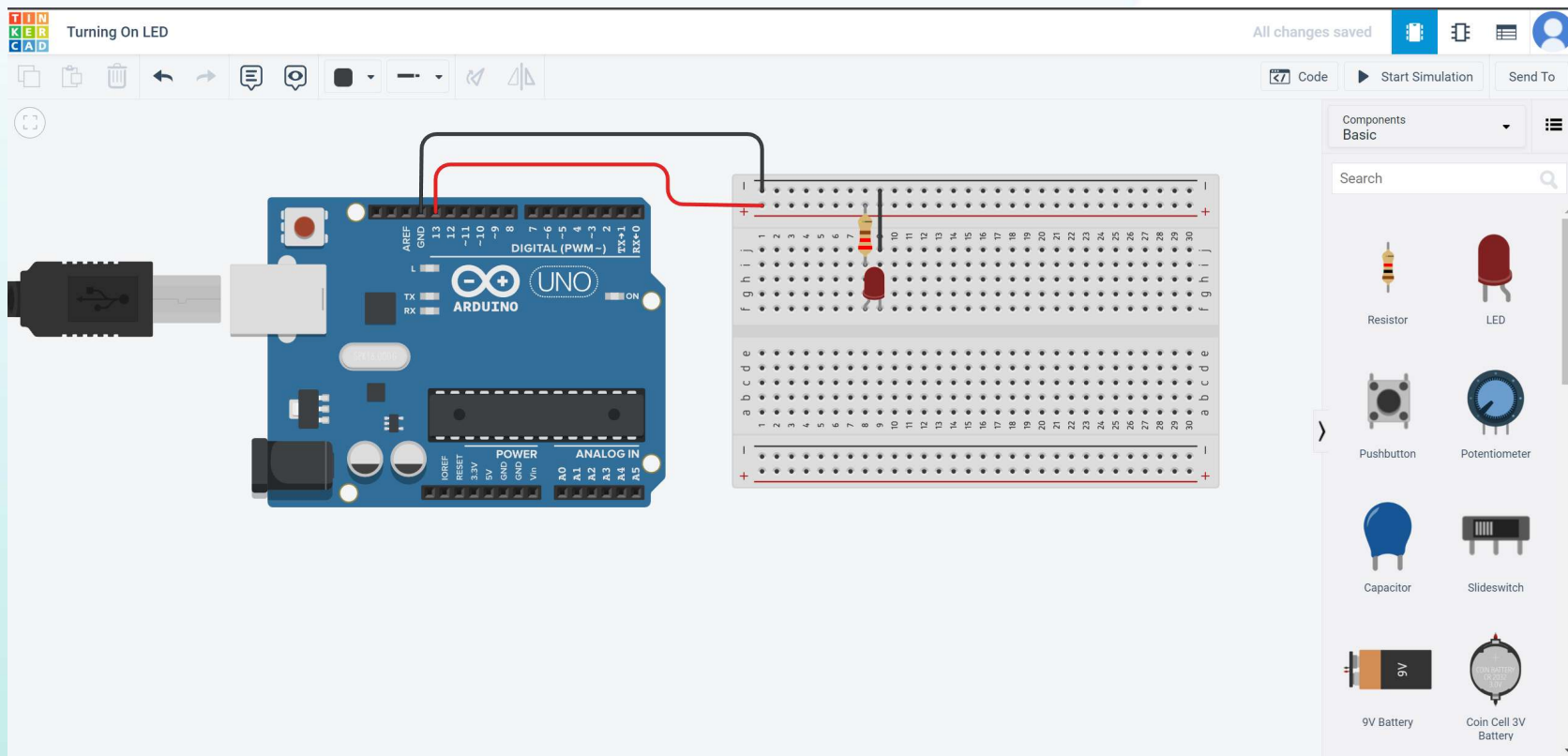


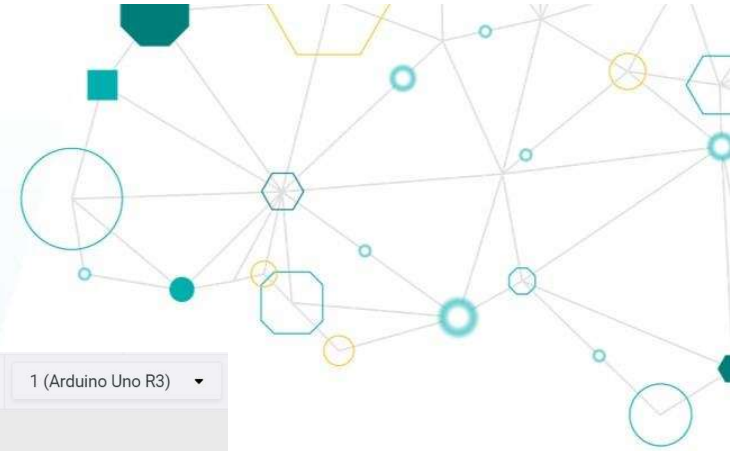
Arduino Introduction (Turn on LED)





Arduino Introduction (Turn on LED)





Arduino Introduction (Turn on LED)

Blocks + Text

Output

Input

Notation

Control

Math

Variables

set built-in LED to HIGH

set pin 0 to HIGH

set pin 3 to 0

rotate servo on pin 0 to 0 degrees

play speaker on pin 0 with tone 6

turn off speaker on pin 0

print to serial monitor hello world with

set RGB LED in pins 3 6

configure LCD 1 type to I2C (MCP2)

print to LCD 1 hello world

set position on LCD 1 to column 0

Serial Monitor

comment Turn the LED ON (HIGH is the voltage level)

set built-in LED to HIGH

wait 1 secs

comment Turn the LED OFF (LOW is the voltage level)

set built-in LED to LOW

wait 1 secs

forever

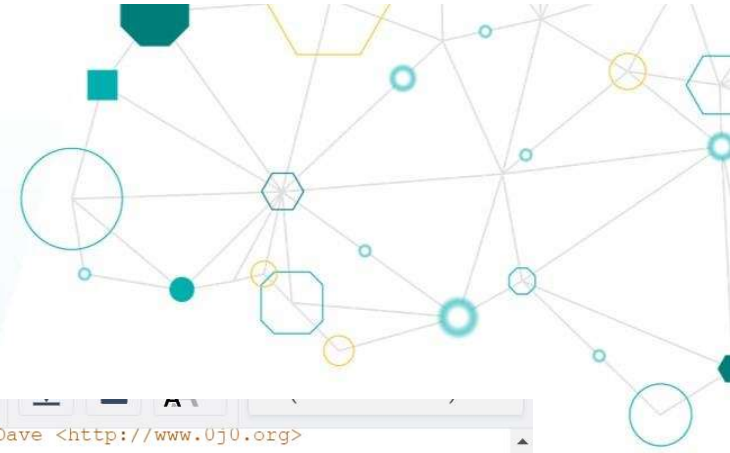
1 (Arduino Uno R3)

```

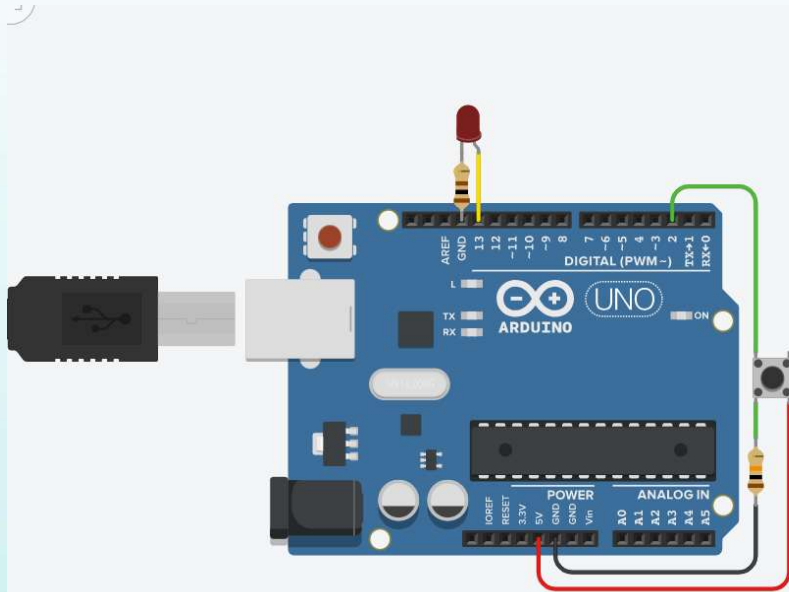
1 // C++ code
2 //
3 void setup()
4 {
5   pinMode(LED_BUILTIN, OUTPUT);
6 }
7
8 void loop()
9 {
10
11   // Turn the LED ON (HIGH is the voltage level)
12   digitalWrite(LED_BUILTIN, HIGH);
13   delay(1000); // Wait for 1000 millisecond(s)
14   // Turn the LED OFF (LOW is the voltage level)
15   digitalWrite(LED_BUILTIN, LOW);
16   delay(1000); // Wait for 1000 millisecond(s)

```

12



Arduino Introduction (Turn on LED by pressing button)



```

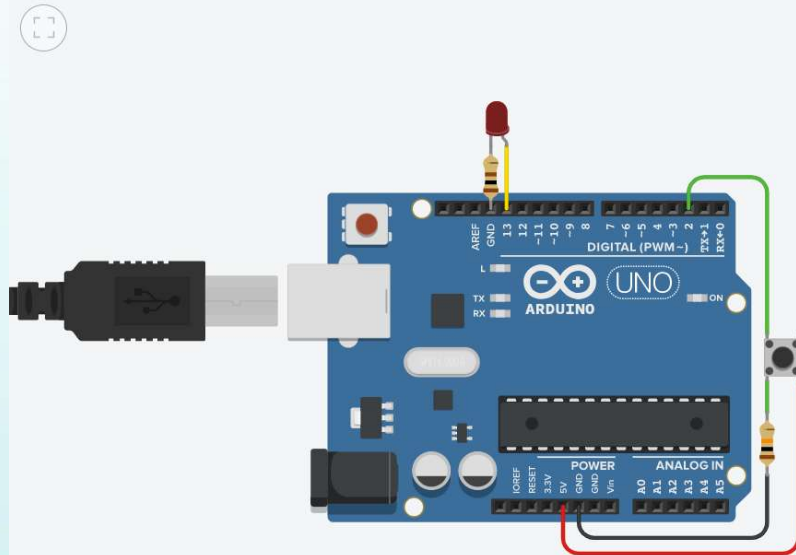
16   created 2005   by DojoDave <http://www.0j0.org>
17
18   modified 30 Aug 2011   by Tom Igoe
19
20   This example code is in the public domain.
21   http://www.arduino.cc/en/Tutorial/Button
22   */
23
24   int buttonState = 0;
25
26   void setup()
27   {
28     pinMode(2, INPUT);
29     pinMode(LED_BUILTIN, OUTPUT);
30   }
31
32   void loop()
33   {
34     // read the state of the pushbutton value
35     buttonState = digitalRead(2);
36     // check if pushbutton is pressed.  if it is, the
37     // buttonState is HIGH
38     if (buttonState == HIGH) {
39       // turn LED on
40       digitalWrite(LED_BUILTIN, HIGH);
41     } else {
42       // turn LED off
43       digitalWrite(LED_BUILTIN, LOW);
44     }
45     delay(10); // Delay a little bit to improve simulation performance
46   }

```

Serial Monitor



Arduino Introduction (Turn on LED by pressing button with time delay)



Text

2 (Arduino Uno R3)

```

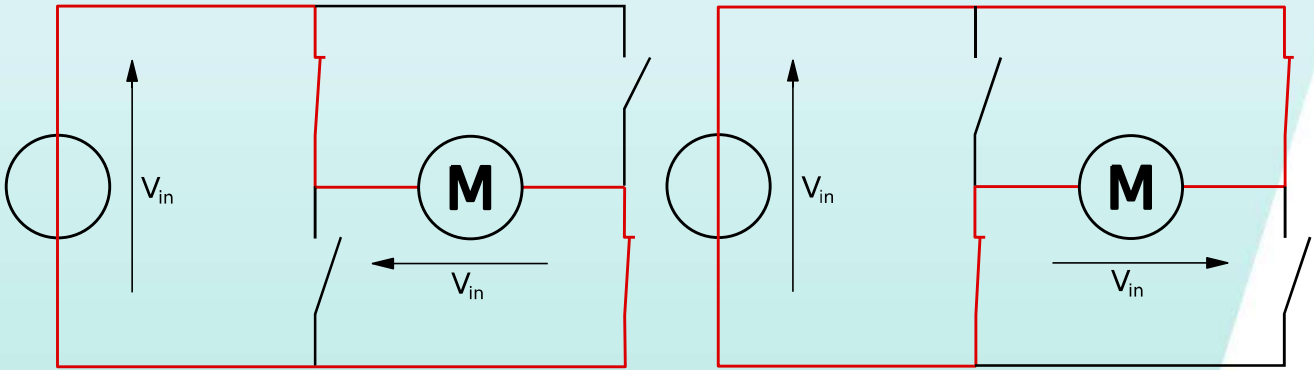
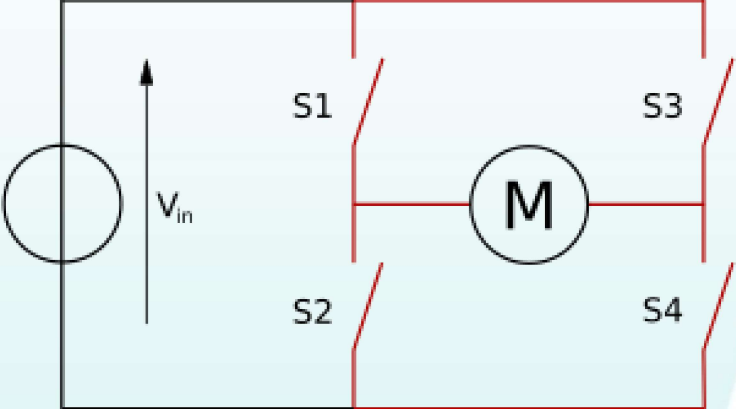
19
20  This example code is in the public domain.
21  http://www.arduino.cc/en/Tutorial/Button
22  */
23
24  int buttonState = 0;
25
26  void setup()
27  {
28    pinMode(2, INPUT);
29    pinMode(LED_BUILTIN, OUTPUT);
30  }
31
32  void loop()
33  {
34    // read the state of the pushbutton value
35    buttonState = digitalRead(2);
36    // check if pushbutton is pressed.  if it is, the
37    // buttonState is HIGH
38    if (buttonState == HIGH) {
39      while (buttonState == HIGH) {
40        // turn LED on
41        digitalWrite(LED_BUILTIN, HIGH);
42        delay(1000); // Wait for 1000 millisecond(s)
43        digitalWrite(LED_BUILTIN, LOW);
44        delay(1000); // Wait for 1000 millisecond(s)
45      }
46    } else {
47      // turn LED off
48      digitalWrite(LED_BUILTIN, LOW);
49    }
50  }

```

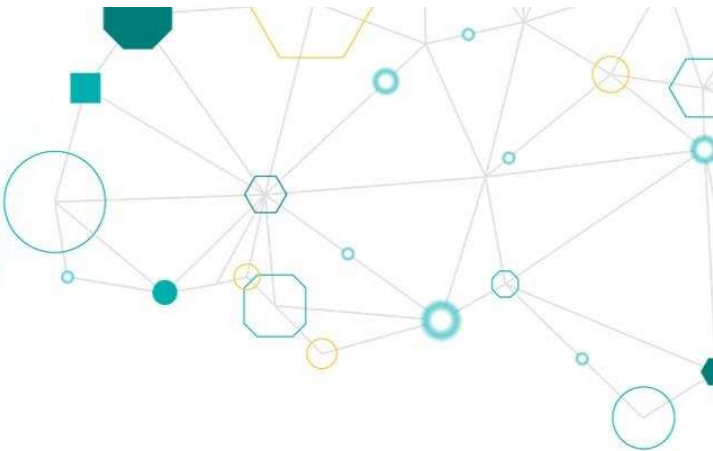
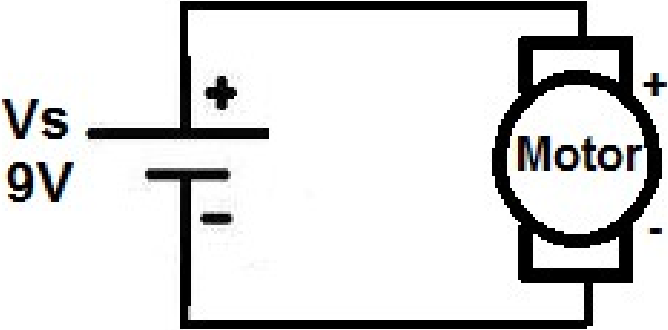
Serial Monitor

Introduction of DC Motor

H-Bridge motor driver



Simple DC motor circuit



H-Bridge Motor Driver

Input 1 (Pin No. 2):-

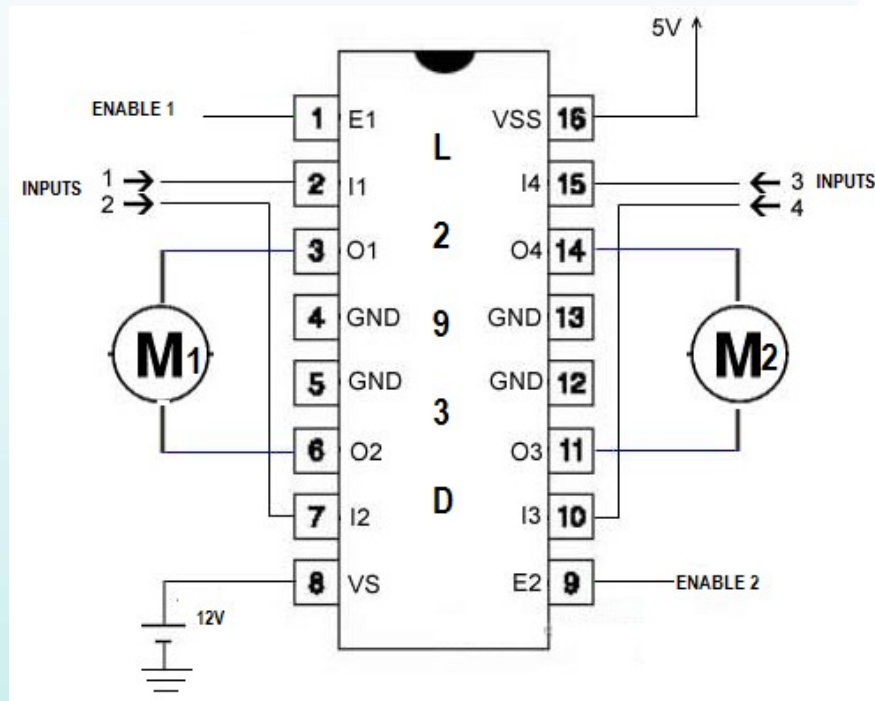
This is an input pin to control the direction of the left motor.

(順時針)

Input 2 (Pin No. 7):-

This is an input pin to control the direction of the left motor.

(逆時針)



Input 3 (Pin No. 15):-

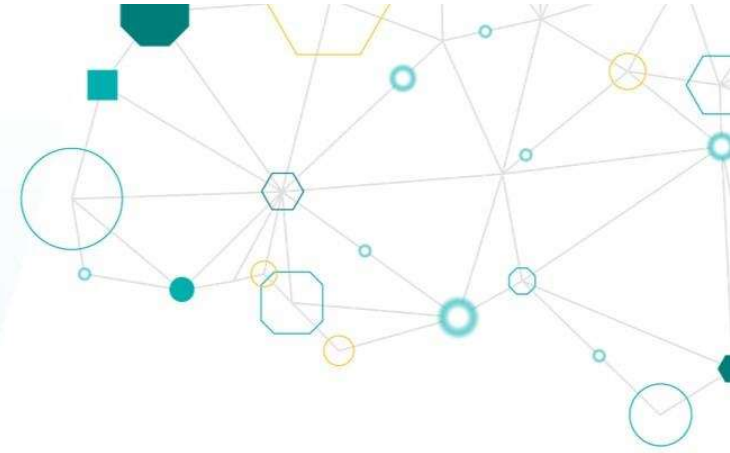
This is an input pin to control the direction of the Right motor.

(順時針)

Input 4 (Pin No. 10):-

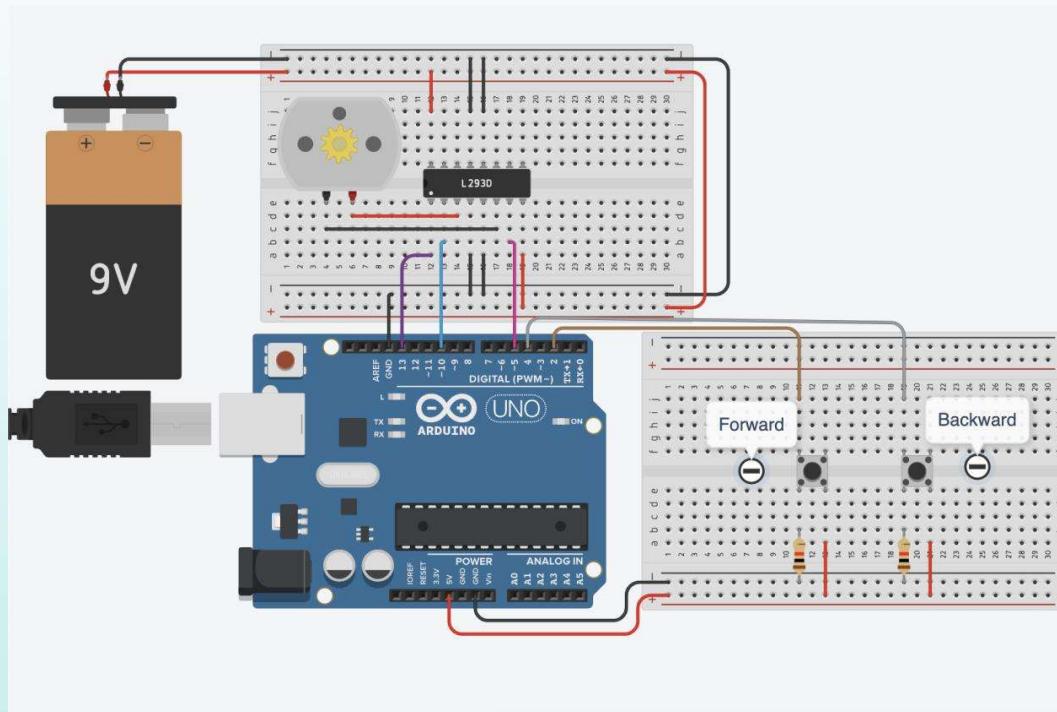
This is an input pin to control the direction of the Right motor.

(逆時針)





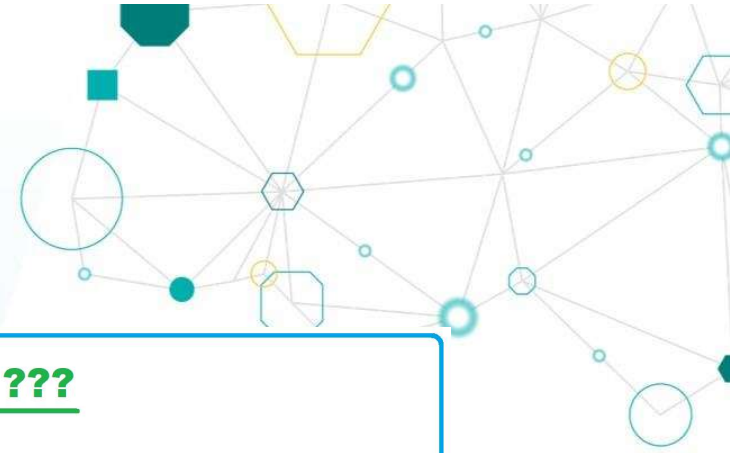
H-Bridge Motor Driver



```

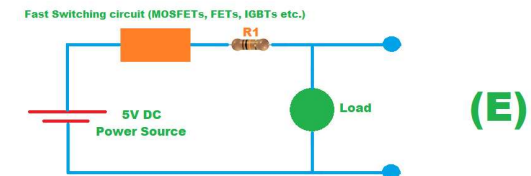
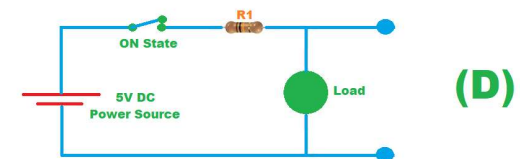
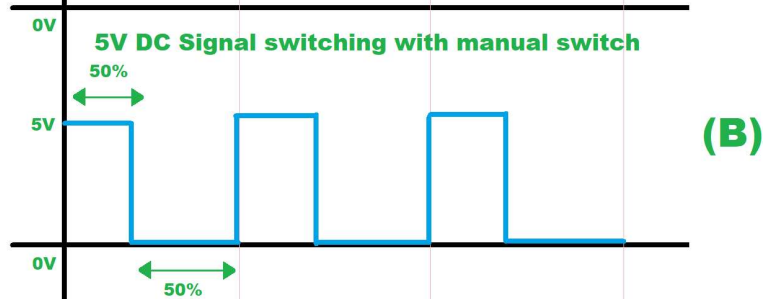
1 // C++ code
2 //
3 int forward_btn = 0;
4
5 int backward_btn = 0;
6
7 void setup()
8 {
9   pinMode(4, INPUT);
10  pinMode(2, INPUT);
11  pinMode(13, OUTPUT);
12  pinMode(10, OUTPUT);
13  pinMode(5, OUTPUT);
14 }
15
16 void loop()
17 {
18   backward_btn = digitalRead(4);
19   forward_btn = digitalRead(2);
20   if (forward_btn != HIGH && backward_btn != HIGH) {
21     digitalWrite(13, LOW); // connected to enable
22     digitalWrite(10, LOW); // connected to input 1
23     digitalWrite(5, LOW); // connected to input 2
24   }
25   if (forward_btn == HIGH) {
26     digitalWrite(13, HIGH); // connected to enable
27     digitalWrite(10, HIGH); // connected to input 1
28     digitalWrite(5, LOW); // connected to input 2
29   }
30   if (backward_btn == HIGH) {
31     digitalWrite(13, HIGH); // connected to enable
32     digitalWrite(10, LOW); // connected to input 1
33     digitalWrite(5, HIGH); // connected to input 2
34   }
35   delay(10); // Delay a little bit to improve simulation performance
36 }

```



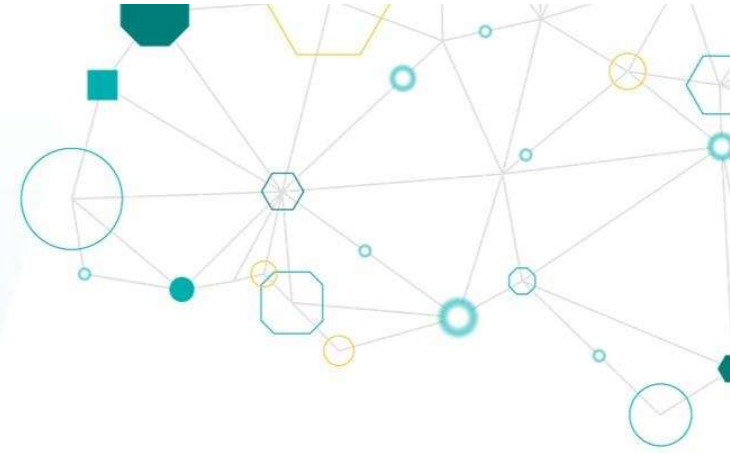
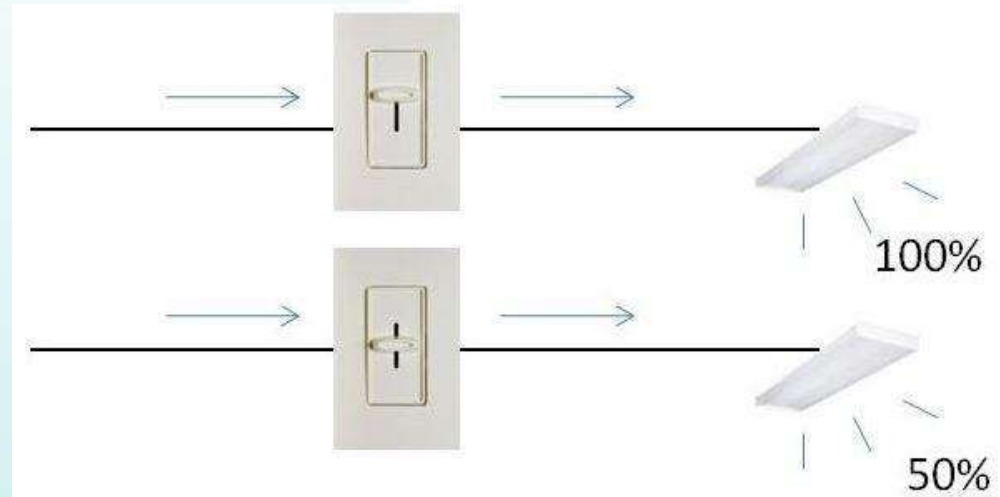
Pulse Width Modulation (PWM)

What is PWM (Pulse Width Modulation) ???



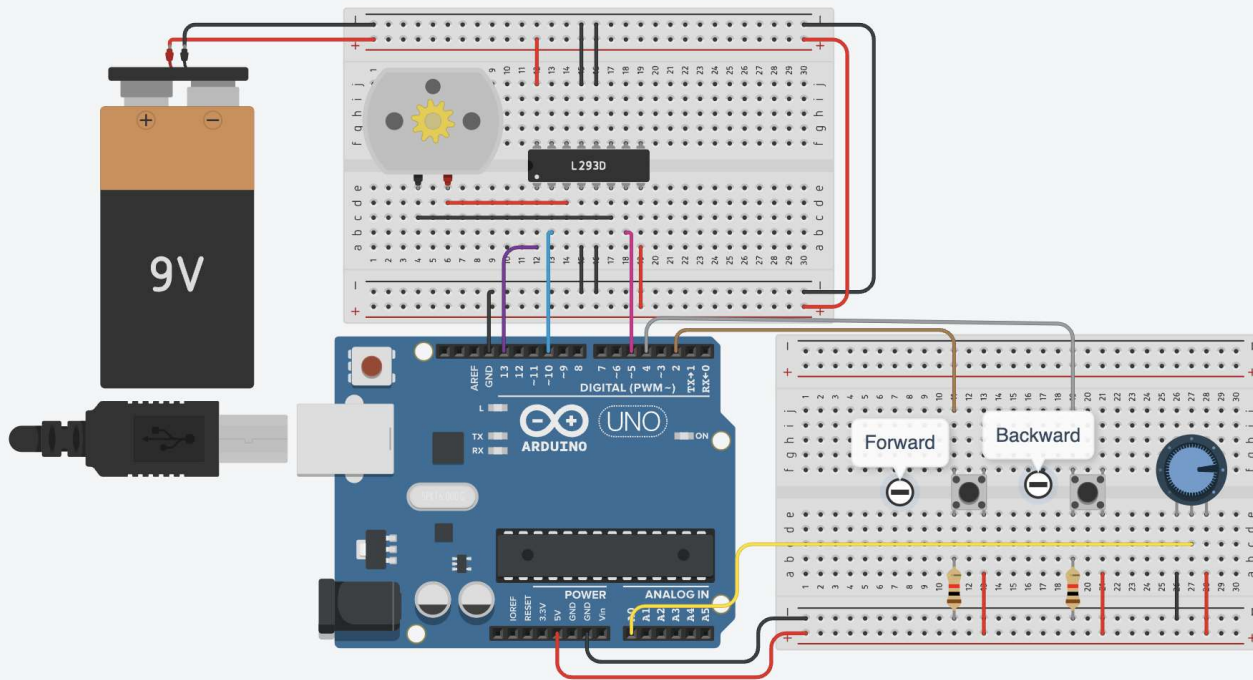
Pulse Width Modulation (PWM)

- Lamp control
- Audio Control
- Motor Speed Control
- etc...





Pulse Width Modulation (PWM) to control one motor's speed

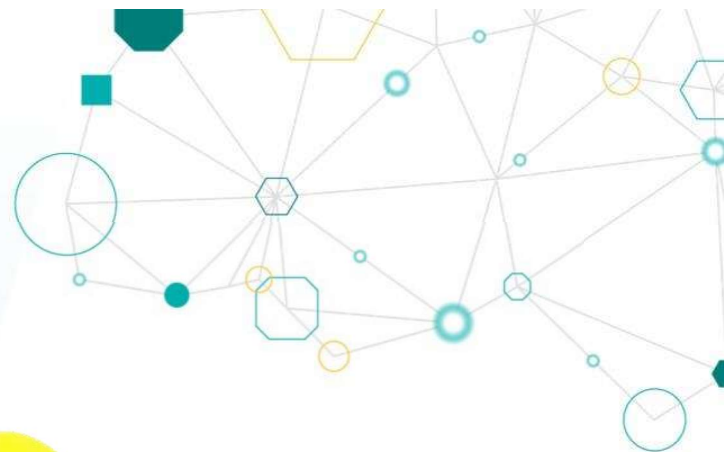


```

1 // C++ code
2 //
3 int forward_btn = 0;
4
5 int backward_btn = 0;
6
7 int varInput = 0;
8
9 void setup()
10 {
11     pinMode(4, INPUT);
12     pinMode(2, INPUT);
13     pinMode(A0, INPUT);
14     pinMode(13, OUTPUT);
15     pinMode(10, OUTPUT);
16     pinMode(5, OUTPUT);
17 }
18
19 void loop()
20 {
21     backward_btn = digitalRead(4);
22     forward_btn = digitalRead(2);
23     /* Map an analog value to 8 bits (0 to 255) */
24     varInput = map(analogRead(A0), 0, 1023, 0, 255);
25     if (forward_btn != HIGH && backward_btn != HIGH) {
26         digitalWrite(13, LOW);
27         digitalWrite(10, LOW);
28         digitalWrite(5, LOW);
29     }
30     if (forward_btn == HIGH) {
31         digitalWrite(13, HIGH);
32         analogWrite(10, varInput);
33         digitalWrite(5, LOW);
34     }
35     if (backward_btn == HIGH) {
36         digitalWrite(13, HIGH);
37         digitalWrite(10, LOW);
38         analogWrite(5, varInput);
39     }
40     delay(10); // Delay a little bit to improve simulation performance
41 }

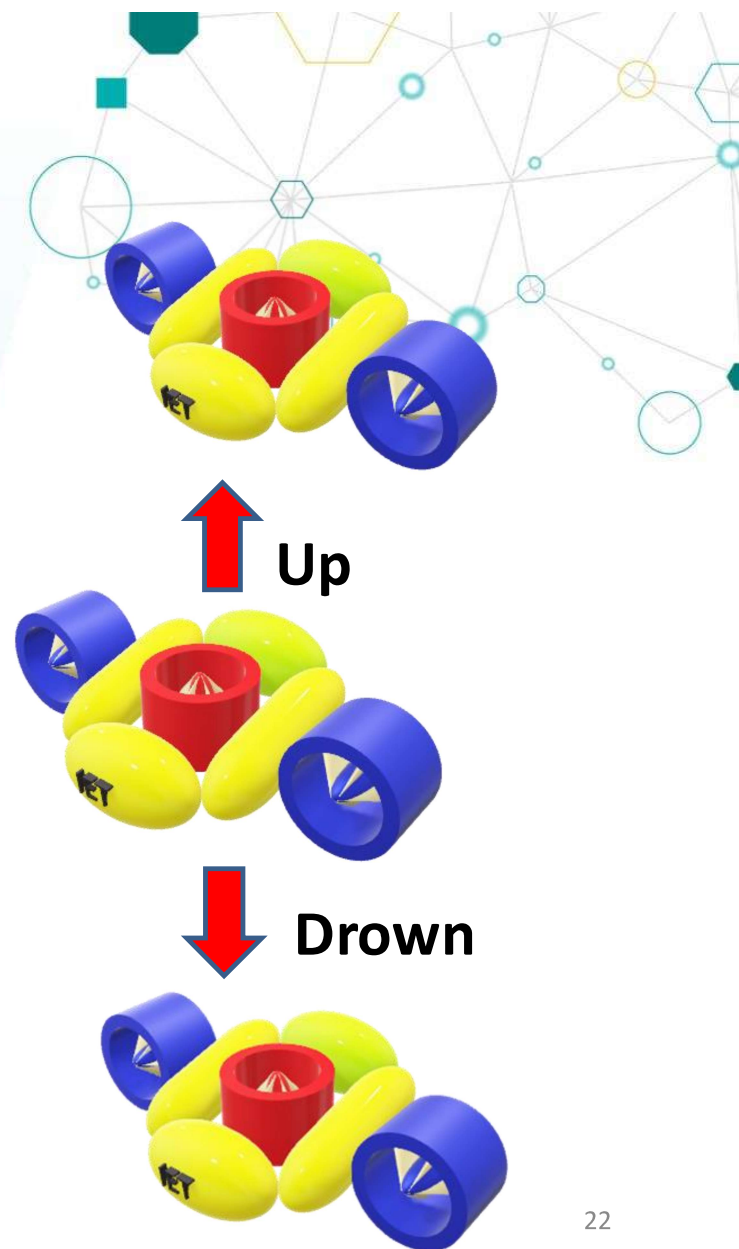
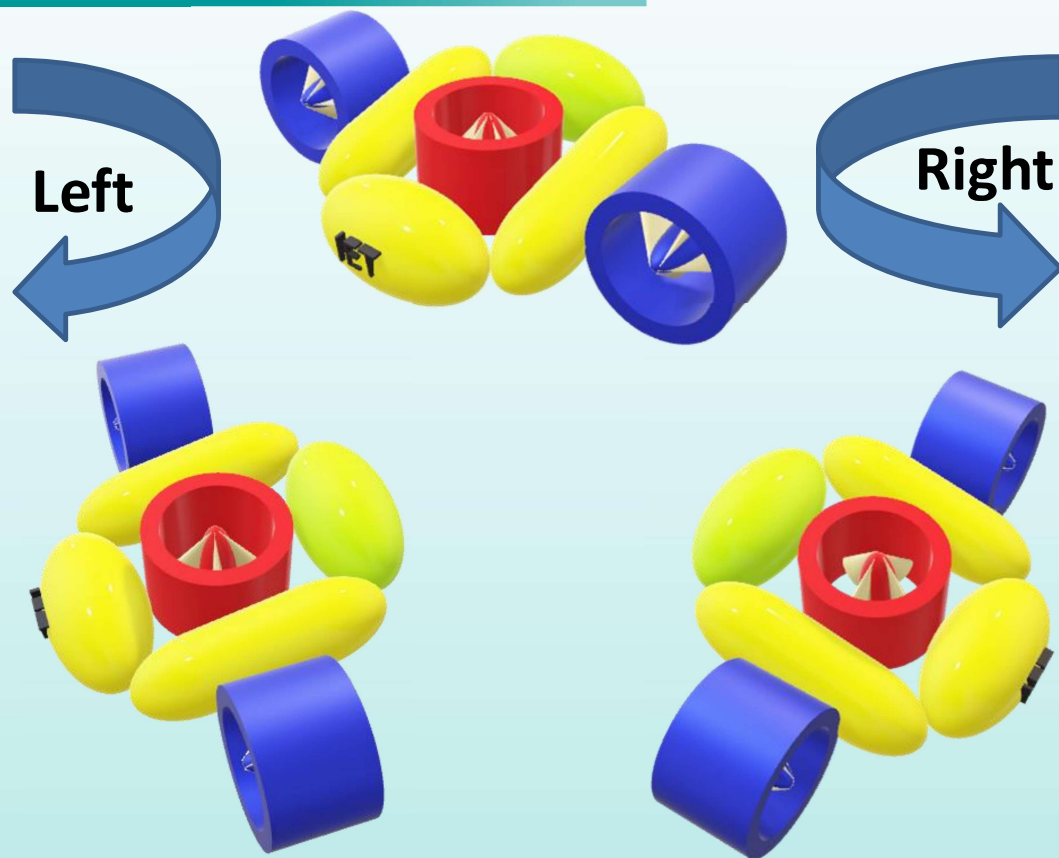
```

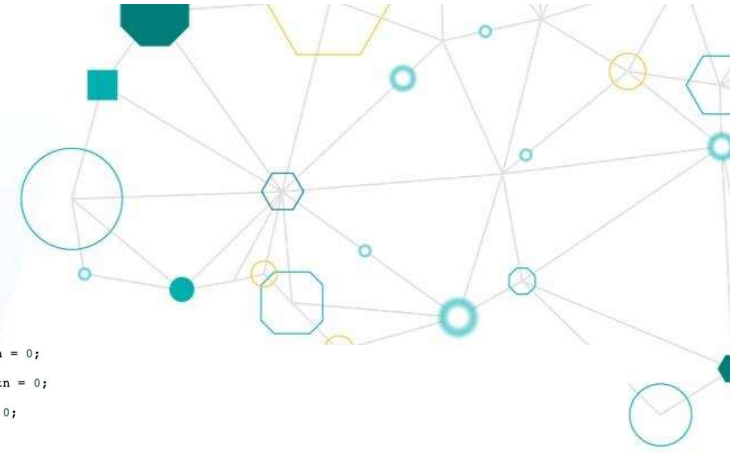
Simplified ROV Structure 簡化的ROV結構



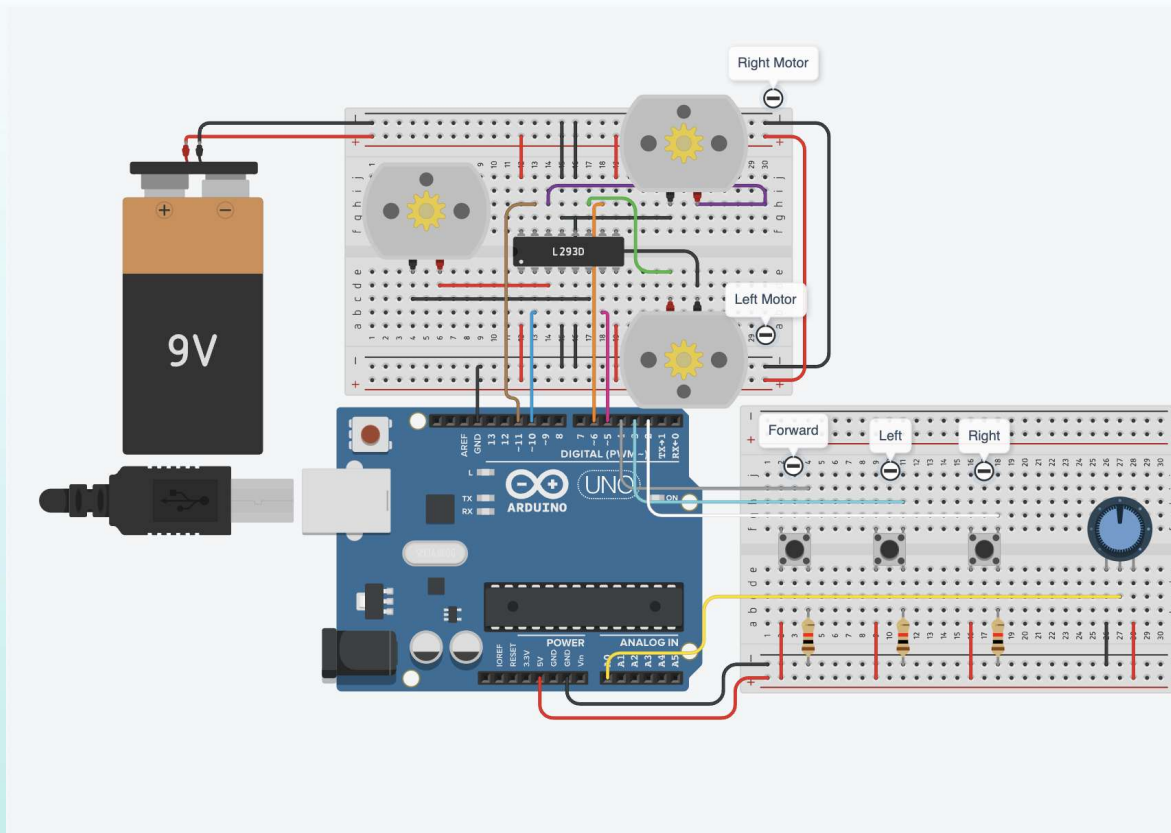
Robot 3D Control

機器人3D控制





Pulse Width Modulation (PWM) to control three motors



```

3 int forward_btn = 0;
4
5 int backward_btn = 0;
6
7 int varInput = 0;
8
9 int Speed = 0;
10
11 void setup()
12 {
13   pinMode(4, INPUT);
14   pinMode(3, INPUT);
15   pinMode(2, INPUT);
16   pinMode(6, OUTPUT);
17   pinMode(11, OUTPUT);
18   pinMode(A0, INPUT);
19   pinMode(10, OUTPUT);
20   pinMode(5, OUTPUT);
21 }
22
23 void loop()
24 {
25   if (digitalRead(4) == HIGH || (digitalRead(3) == HIGH || digitalRead(2) == HIGH)) {
26     if (digitalRead(4) == HIGH) { // Forward button is on
27       digitalWrite(6, HIGH); // input 3 (left motor)
28       digitalWrite(11, HIGH); // input 4 (right motor)
29     }
30     if (digitalRead(3) == HIGH) { // Left button is on
31       digitalWrite(6, LOW);
32       digitalWrite(11, HIGH); // input 4 (right motor)
33     }
34     if (digitalRead(2) == HIGH) { // Right button is on
35       digitalWrite(6, HIGH); // input 3 (left motor)
36       digitalWrite(11, LOW);
37     }
38   } else {
39     digitalWrite(6, LOW);
40     digitalWrite(11, LOW);
41   }
42   varInput = map(analogRead(A0), 0, 1023, 0, 255);
43   if (varInput > 122 && varInput < 132) {
44     digitalWrite(10, LOW);
45     digitalWrite(5, LOW);
46   }
47   if (varInput > 132) {
48     Speed = constrain((abs((varInput - 127)) * 2), 0, 255);
49     analogWrite(10, Speed);
50     digitalWrite(5, LOW);
51   }
52   if (varInput < 122) {
53     Speed = constrain((abs((varInput - 127)) * 2), 0, 255);
54     digitalWrite(10, LOW);
55     analogWrite(5, Speed);
56   }
57   delay(10); // Delay a little bit to improve simulation performance
58 }

```