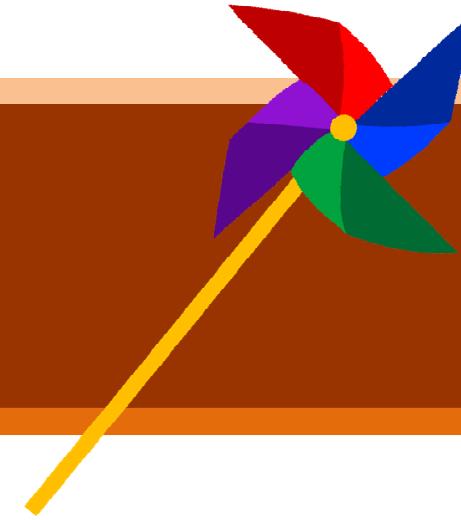


Project 9

Moterized Pinwheel



Discover: transistors, high current/voltage loads

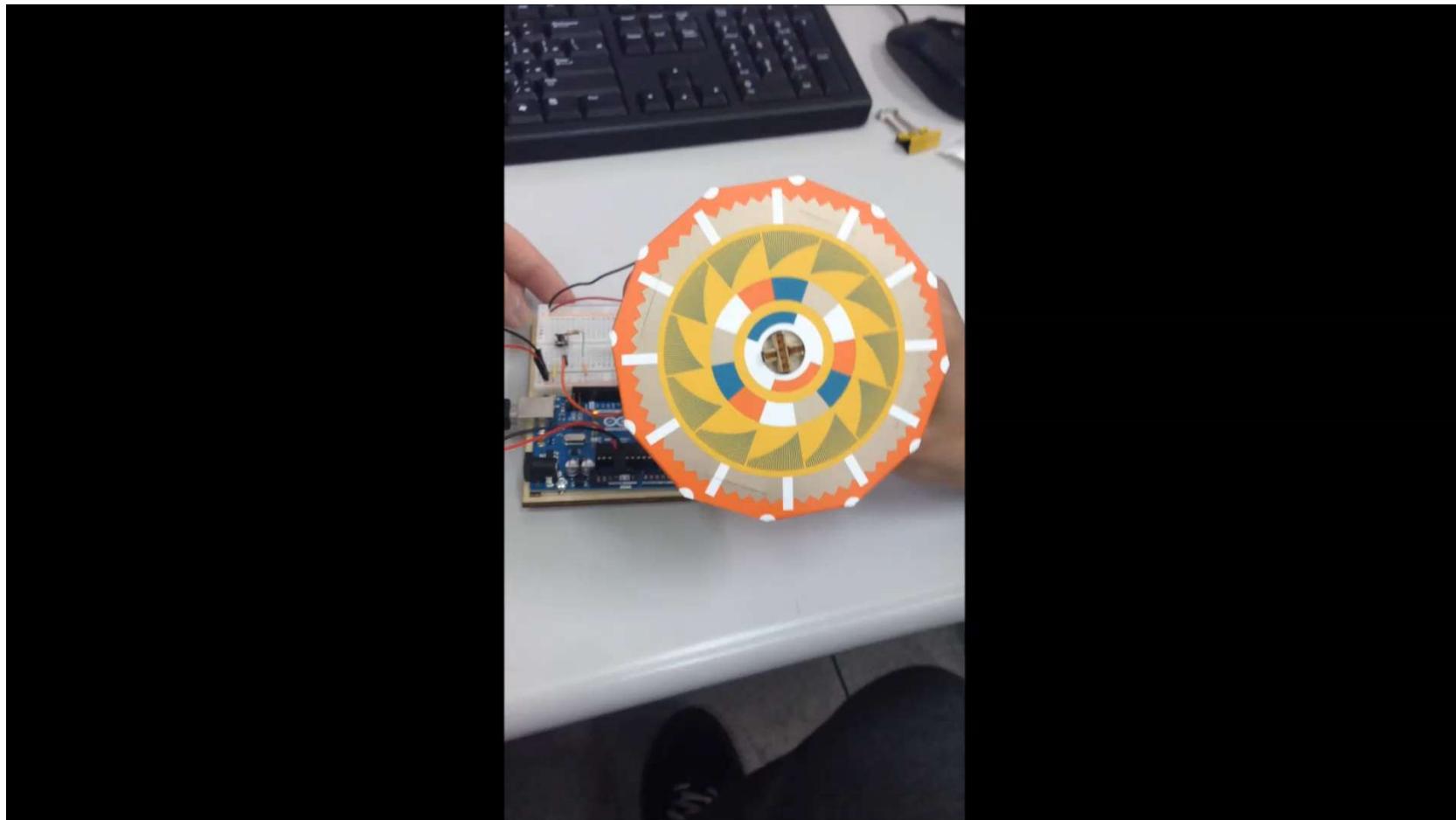
Introduction

- Controlling motors with an Arduino is more complicated than just controlling LEDs for a couple of reasons.
 - Motors require more **current** than the Arduino's output pins can supply
 - Motors can generate their own current through a process called **induction**, which can damage circuit.
- Motors make it possible to move physical things, making your projects much more exciting.
 - They're worth the complications!

Motor Control

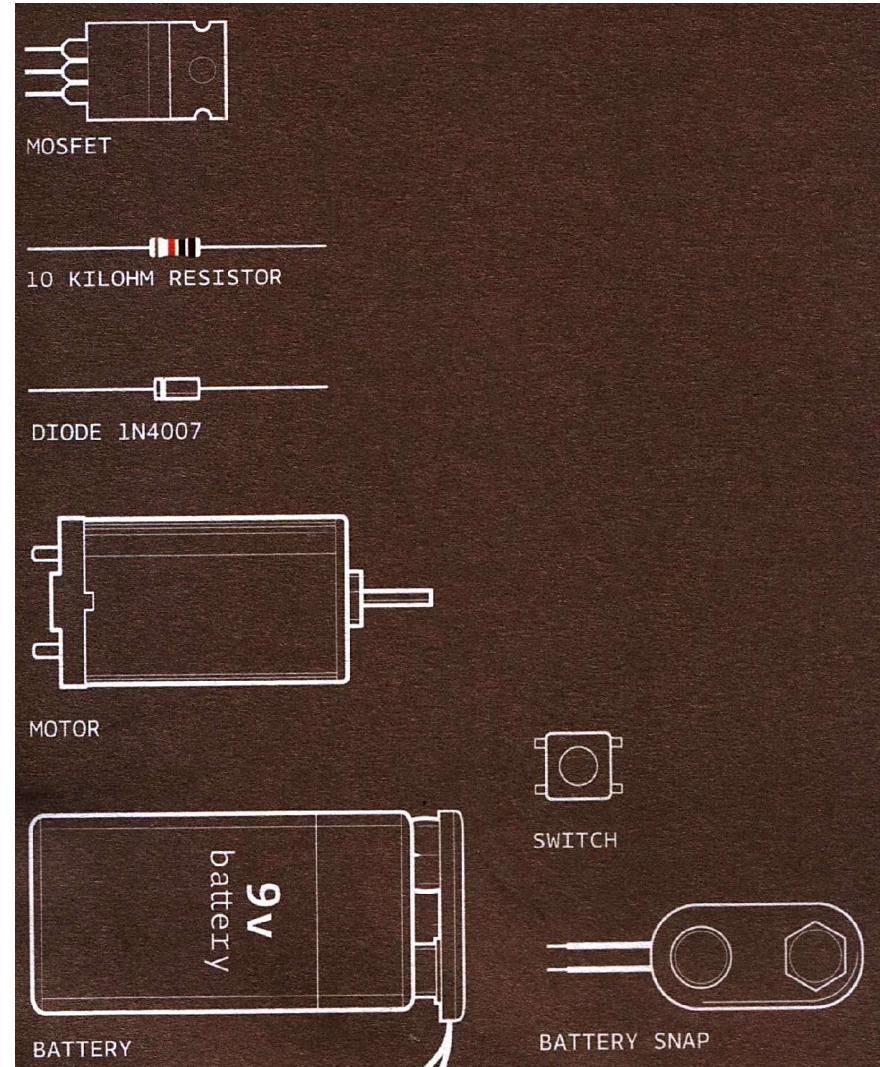
- Motors requires current than a Arduino can provide (**40mA** at most)
→ use an external 9v battery.
- MOSFET is a transistor (digital switch), and allow you to control high current or voltage power sources from Arduino.
 - How a transistor works: provide voltage to the **gate** pin → close the circuit between the other two pins (**source** and **drain**).
- A motor is a type of inductive device.
 - **Induction:** changing electrical current in a wire → a changing magnetic field around the wire.
 - When a motor is given electricity a tightly wound coil create a magnetic field which cause the shaft to spin around.

Demo



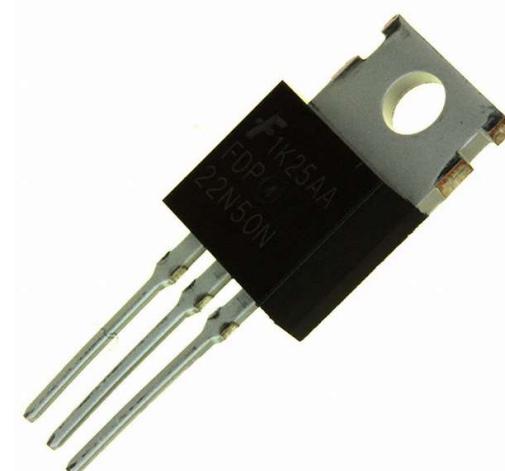
Ingredients

- 1x 電晶體(MOSFET)
- 1個 $10K\Omega$ 電阻(RESISTOR)
- 1x 二極體(DIODE IN4007)
- 1x 馬達(MOTOR)
- 1x 開關(SWITCH)
- 1x 9V電池(BATTERY)
- 1x 電池扣(BATTERY SNAP)



MOSFET

- 金屬氧化物半導體場效電晶體(MOSFET): Metal-Oxide-Semiconductor Field-Effect Transistor
- 金屬氧化物半導體場效電晶體是一種可以廣泛使用在類比電路與數位電路的場效電晶體。
- MOSFET依照其通道極性的不同，可分為
 - 電子占多數的N通道型
 - 通常被稱為N型金氧半場效電晶體 (NMOSFET)
 - 電洞占多數的P通道型
 - P型金氧半場效電晶體 (PMOSFET) 。



MOSFET (cont.)

- 利用氧化層在MOSFET的G(閘極)端子和其他電極之間形成絕緣，讓在DS(汲極·源極)之間形成PN接合區，如此即形成類似二極體的結構。 C_{gs}, C_{gd} 取決於氧化層的靜電容量，而 C_{ds} 的容量則依內部等效二極體的接合容量不同而異。

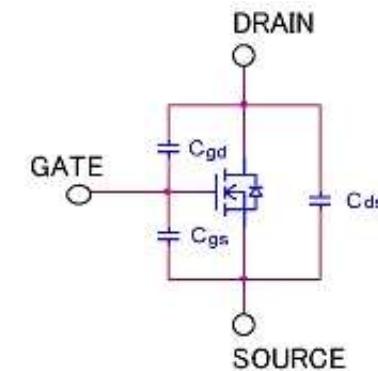
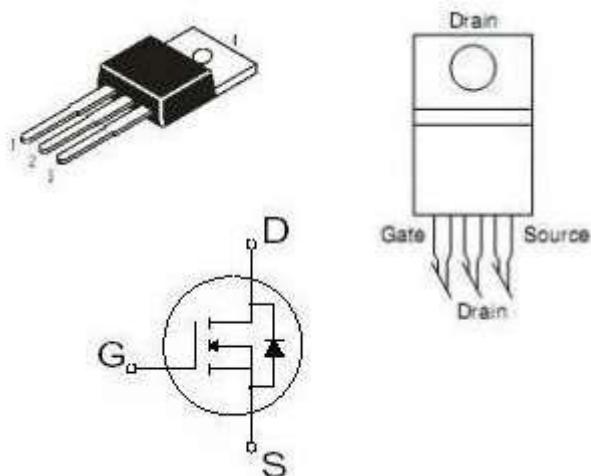
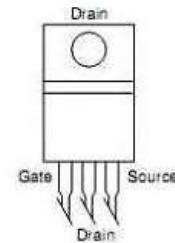
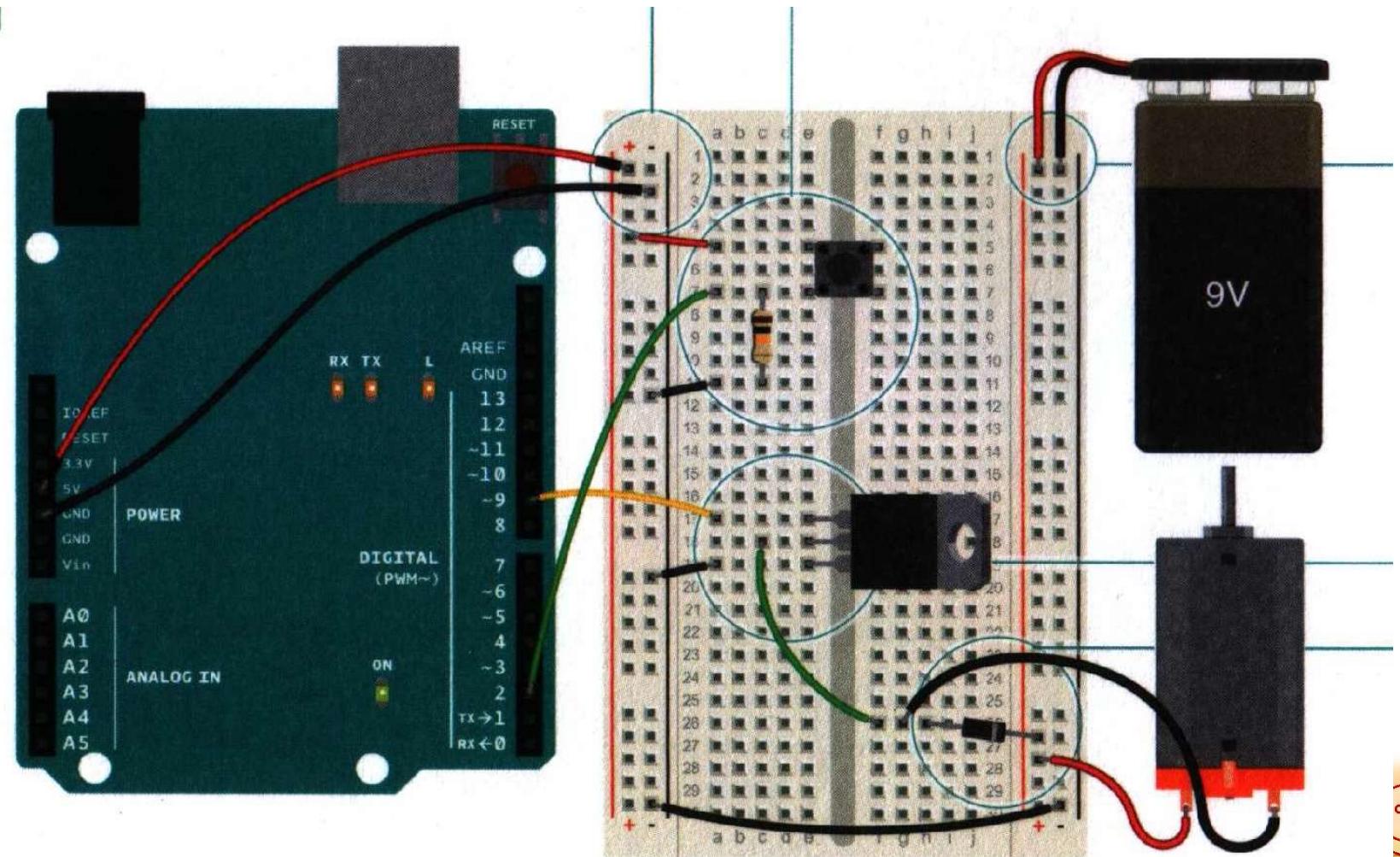


圖1: MOSFET的容量類型

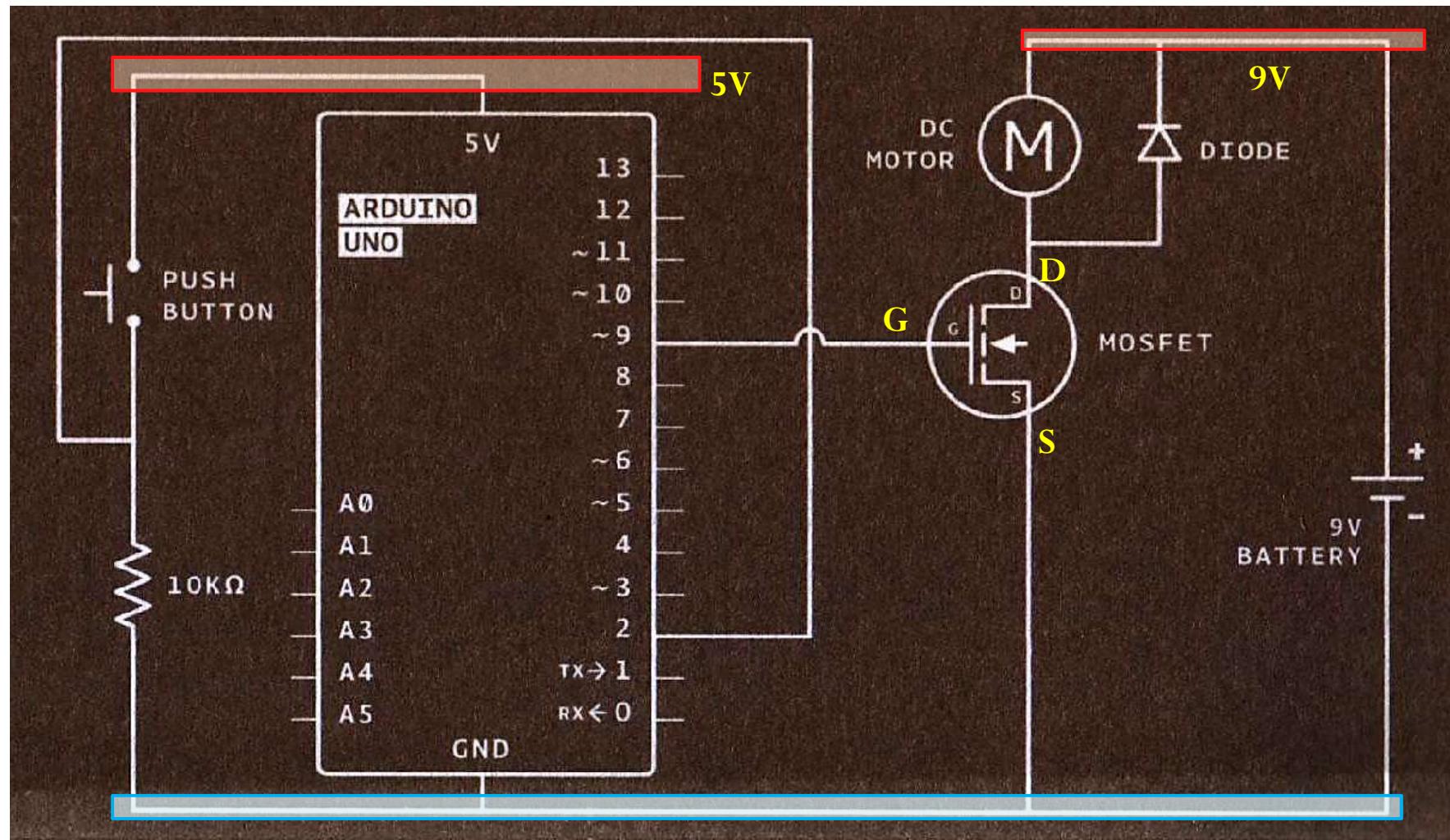
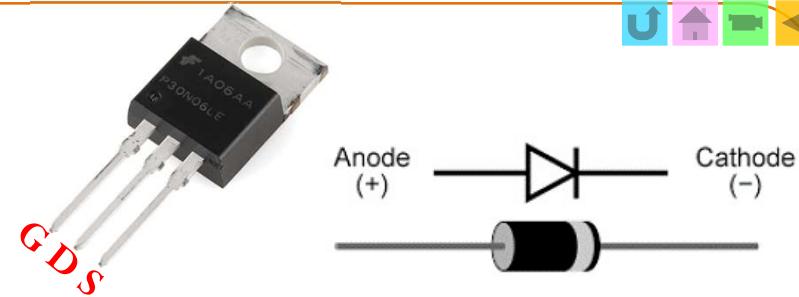
表1: MOSFET的容量特性

符號	公式	意義
C_{iss}	$C_{gs} + C_{gd}$	輸入容量
C_{oss}	$C_{ds} + C_{gd}$	輸出容量
C_{rss}	C_{gd}	歸返容量

The Top View of the Circuit



Schematic Diagram



The diode is for **back-voltage** protection.

The Codes

```
1 const int switchPin = 2;  
2 const int motorPin = 9;  
3 int switchState = 0;  
  
4 void setup() {  
    pinMode(motorPin, OUTPUT);  
    pinMode(switchPin, INPUT);  
}  
7 }
```

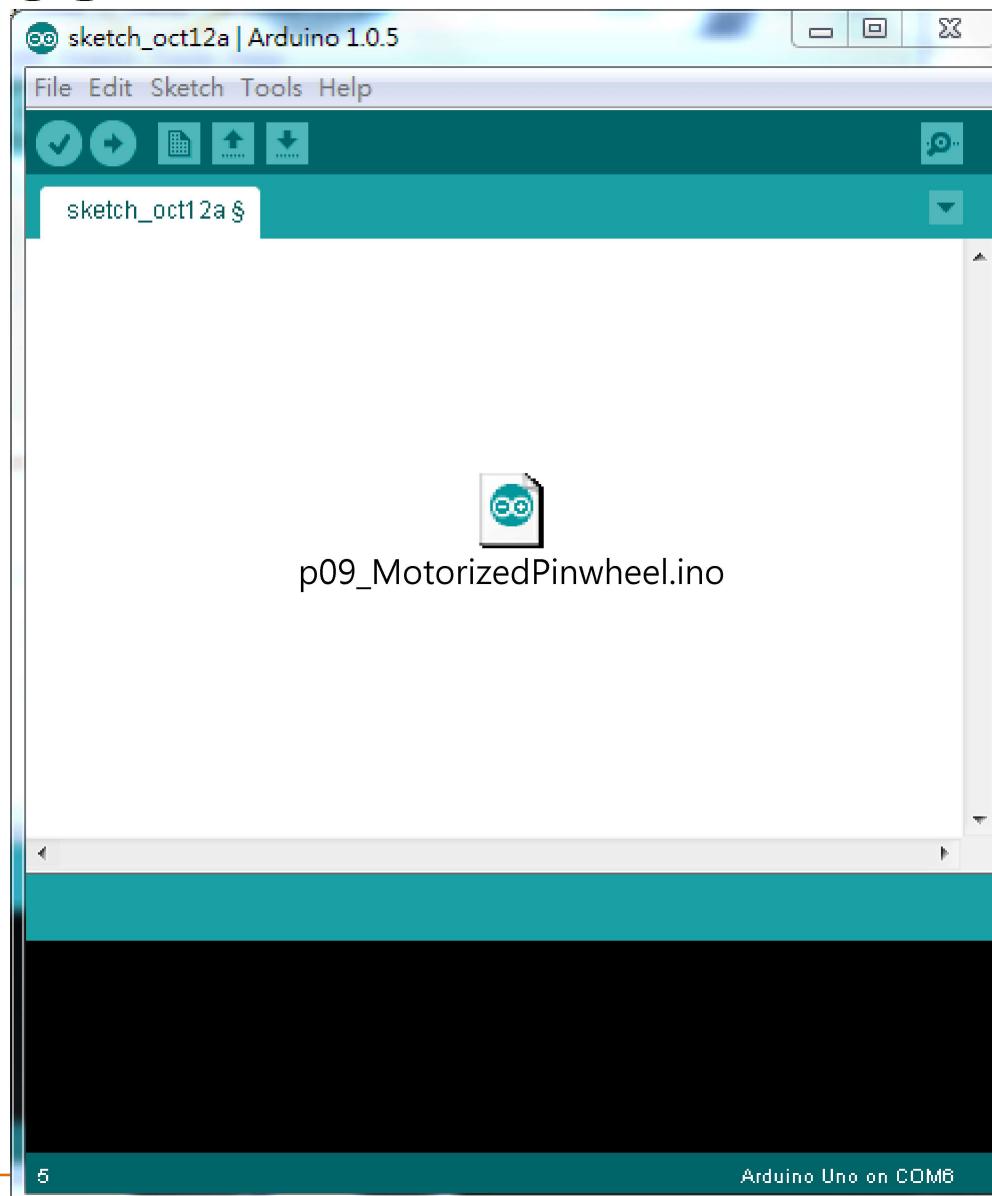


The Codes (cont.)

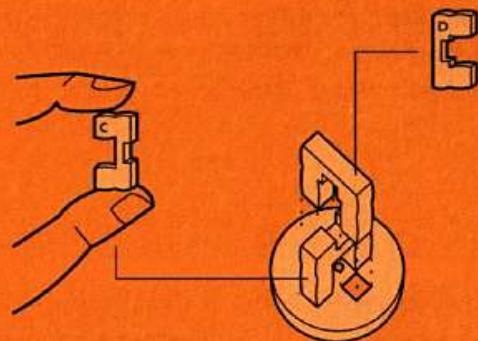
```
8 void loop(){
9     switchState = digitalRead(switchPin);
10
10    if (switchState == HIGH) {
11        digitalWrite(motorPin, HIGH);
12    }
13    else {
14        digitalWrite(motorPin, LOW);
15    }
16 }
```



Codes

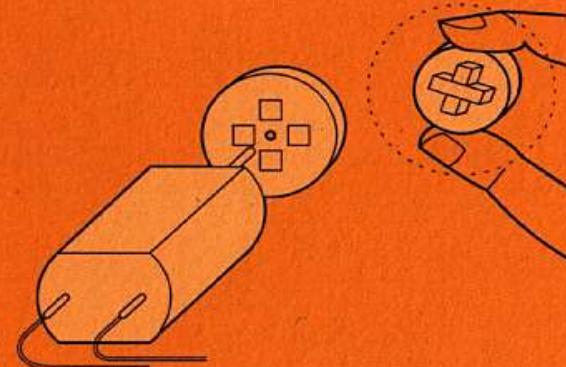


Finish Up



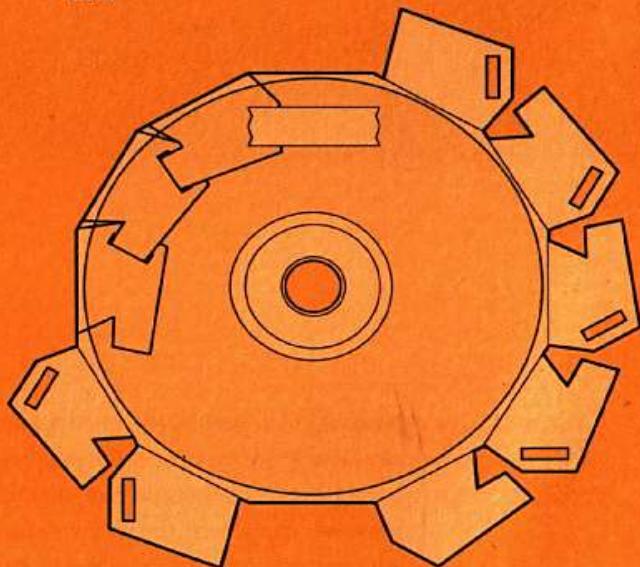
1

Snap part C into part B, and then gently press part D on to them.



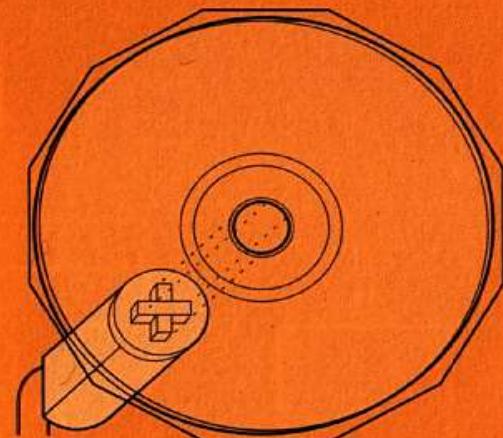
2

Gently press the motor shaft into the hole in the back of part B.



3

Place the paper disk on the CD and secure it using the flaps on the back.



4

Attach the CD to the cross formed by parts B and D. Use a drop of glue to keep the CD from coming off.



What if?

- Motor speed control
 - try hooking up a **potentiometer** to an analog input and use that to PWM the pin that controls the transistor.
 - What do you think will happen to the motor's speed if you vary the voltage it's getting?
- Using your patterns on your spinner, can you get different visual effects?

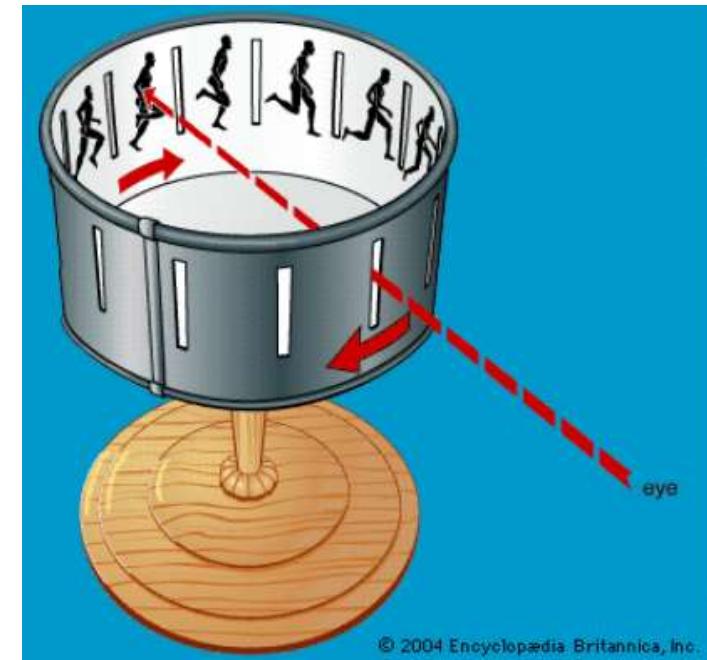
Project 10

Zoetrope (旋轉畫筒)

Discover: H-bridges

Introduction

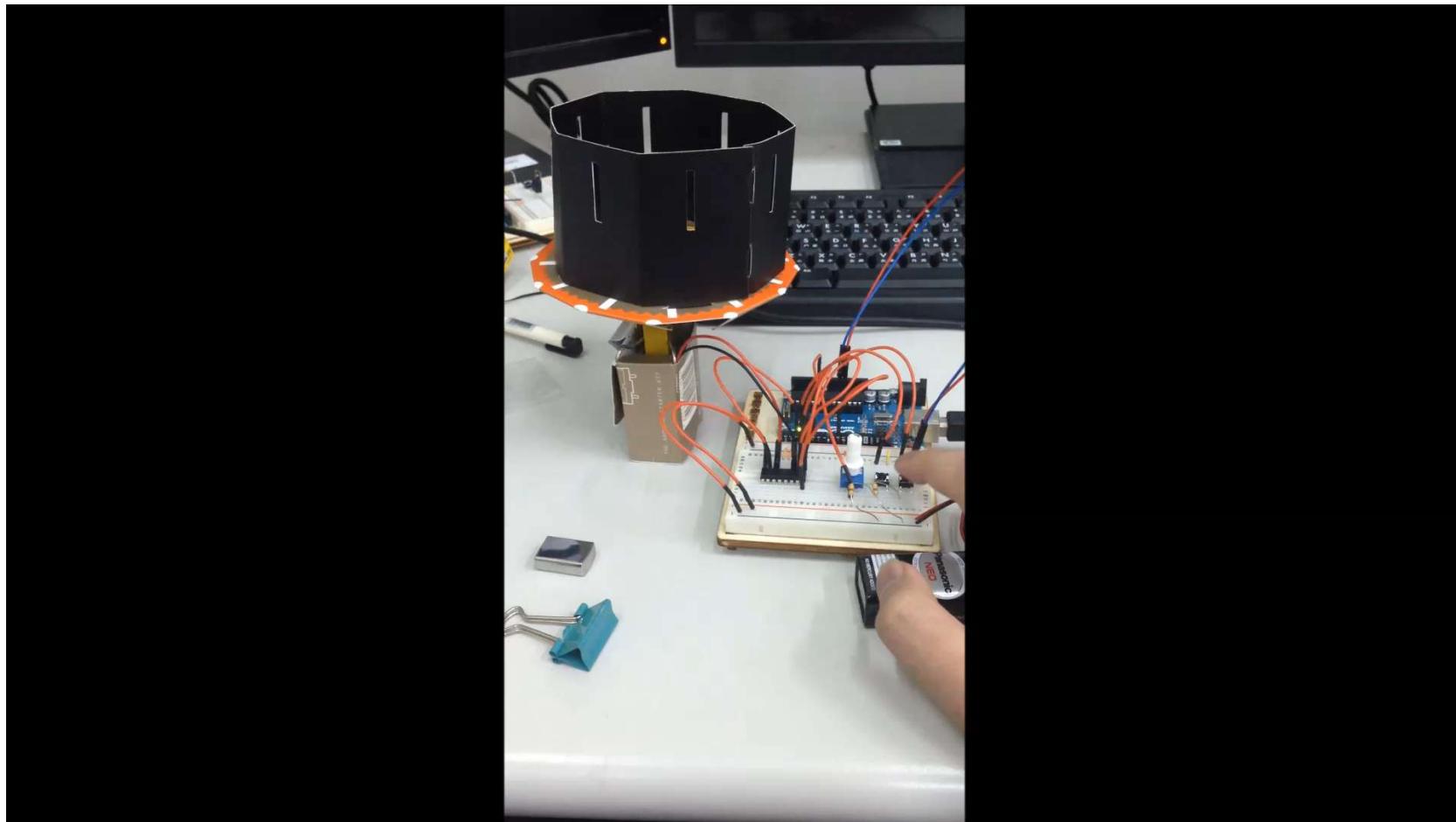
- Before the internet, television, even before movies, some of the first moving images were created with a tool called a **zoetrope**.
- Zoetropes create the illusion of motion from a group of still images that have small changes in them.
 - Originally, these novelties/were **spun by hand**, or with a **cranking mechanism**.



Introduction

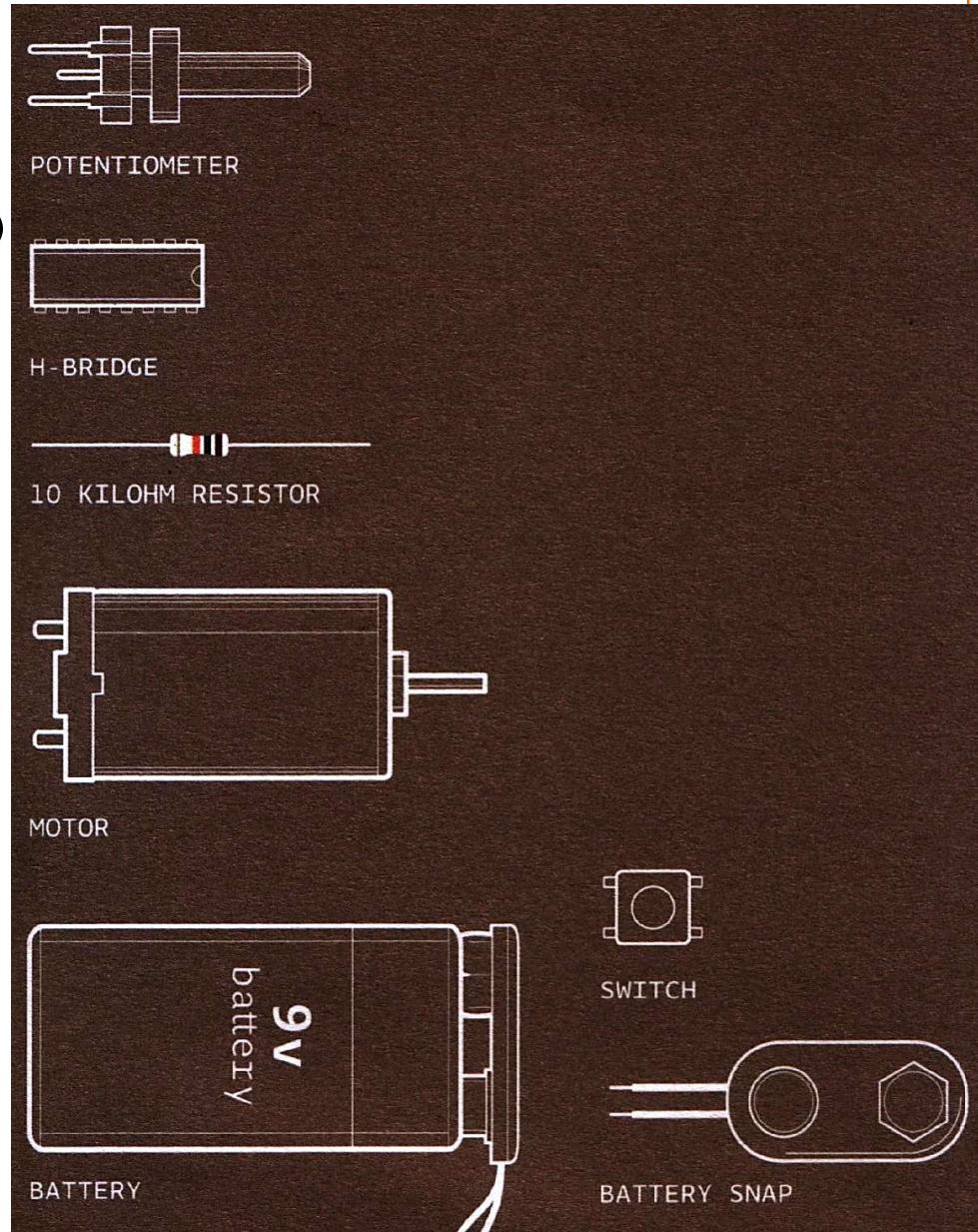
- Based on the previous project, we control the rotation of a motor use an **H-Bridge** (by reversing the polarity of the motor).
- An H-Bridge is a type of component know as Integrated Circuit (IC) that has a lot of transistor built in.

Demo



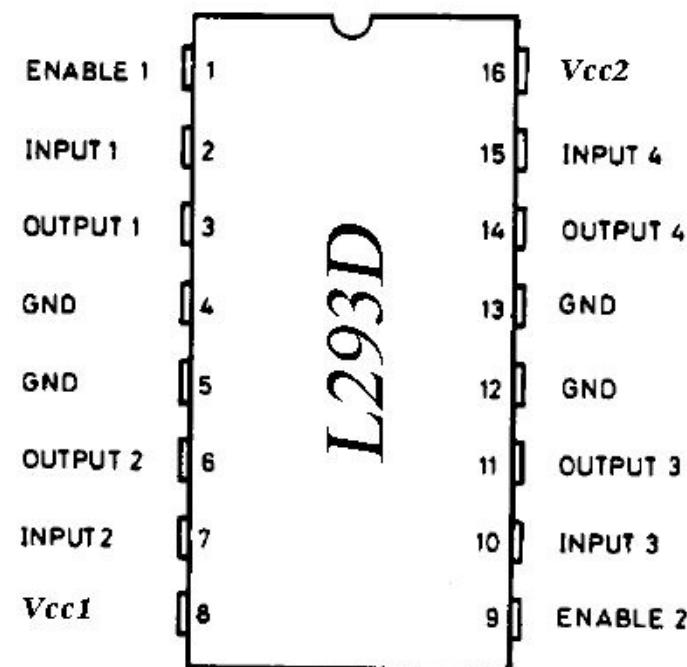
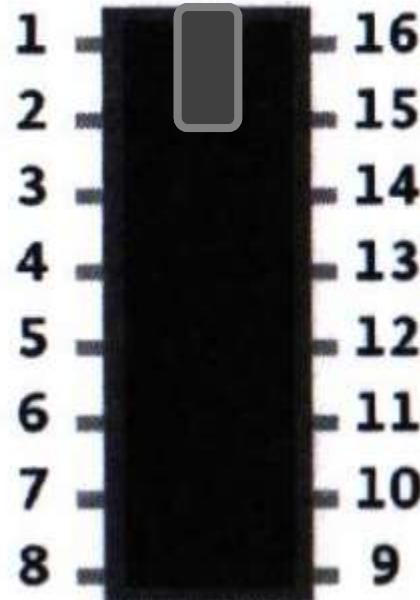
Ingredients

- 電位器(POTENTIOMETER)
- H橋(H-BRIDGE, L293D)
- 2個10K Ω電阻(RESISTOR)
- 馬達(MOTOR)
- 2個開關(SWITCH)
- 9V電池(BATTERY)
- 電池扣(BATTERY SNAP)



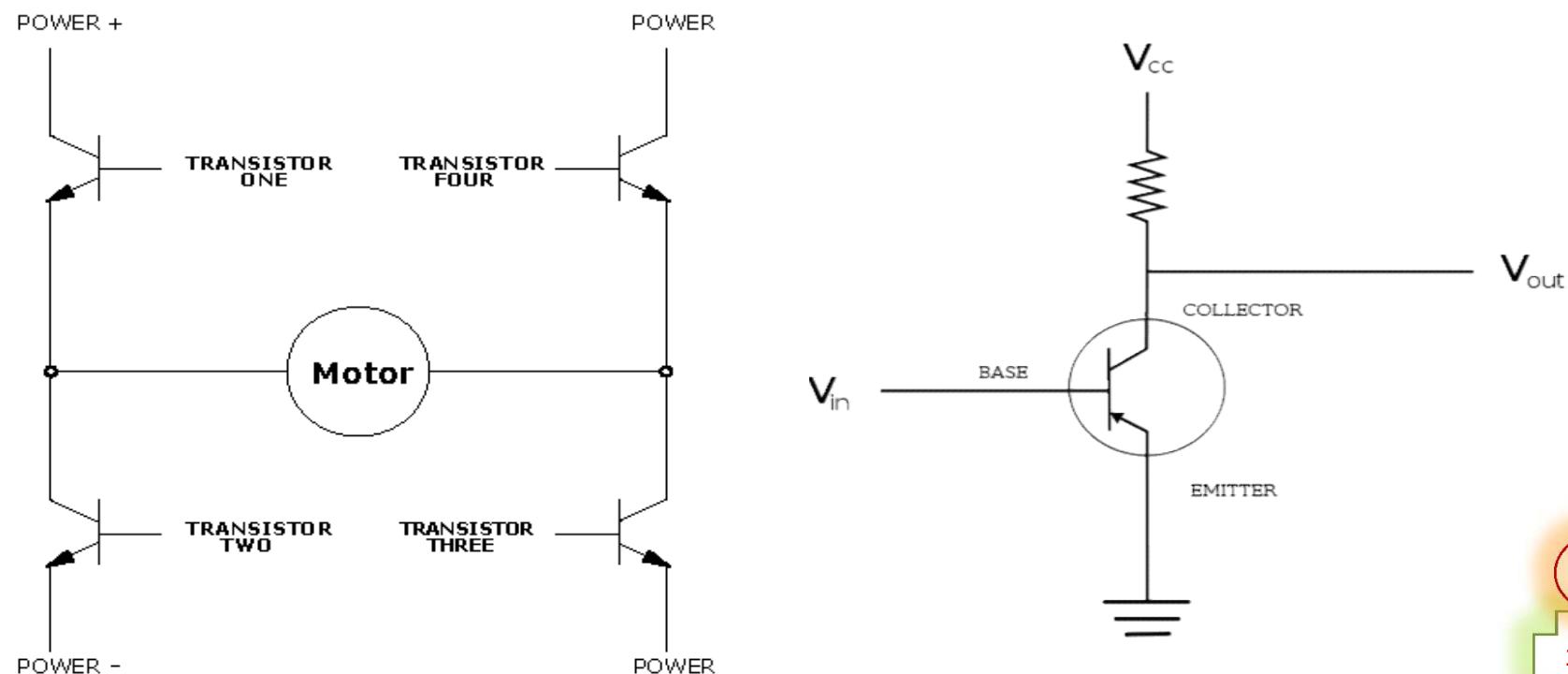
H-Bridge

- An H bridge is an electronic circuit that enables a voltage to be applied across a load in either direction.
 - These circuits are often used in robotics and other applications to allow DC motors to run forwards and backwards

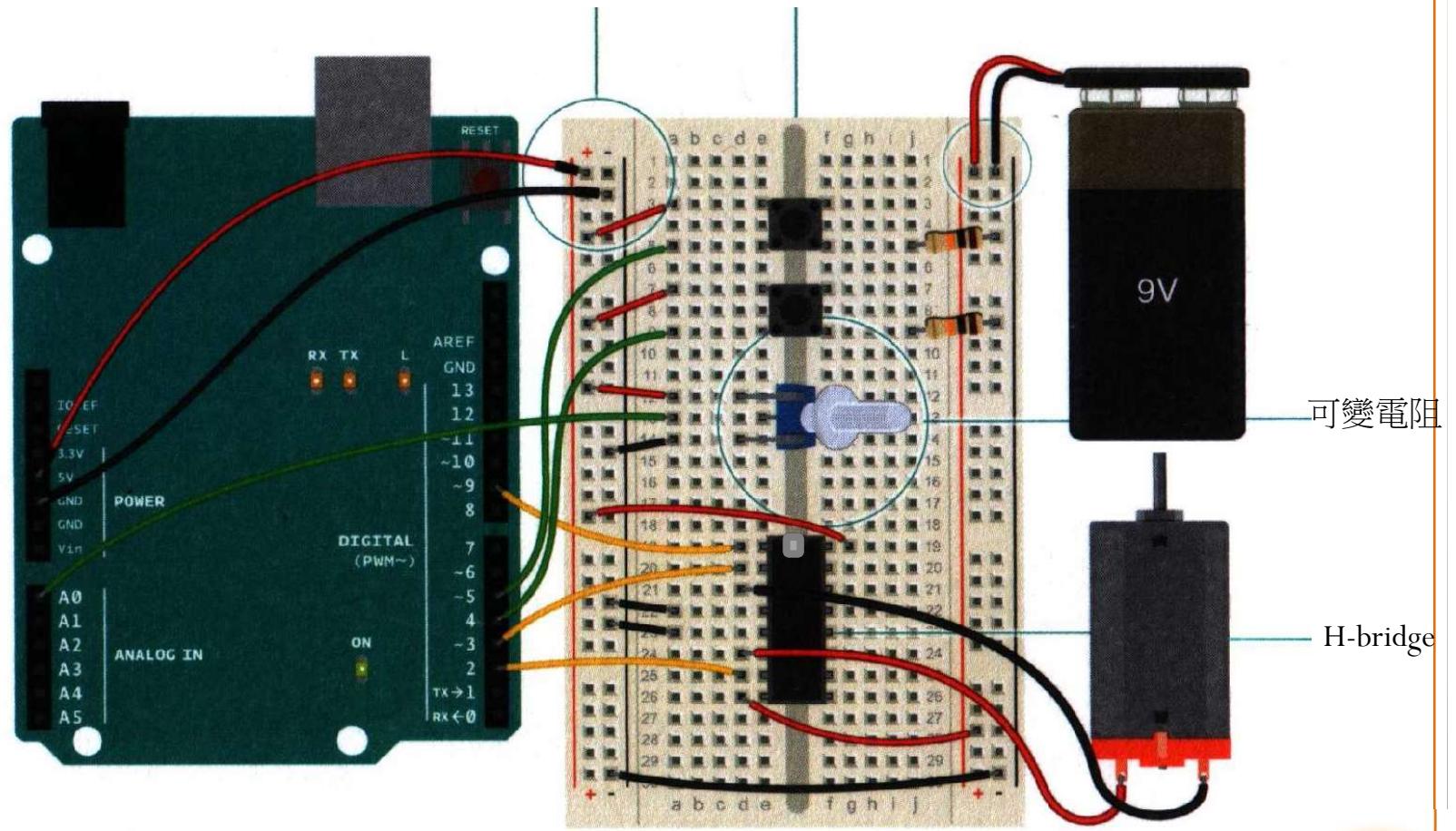


H-Bridge

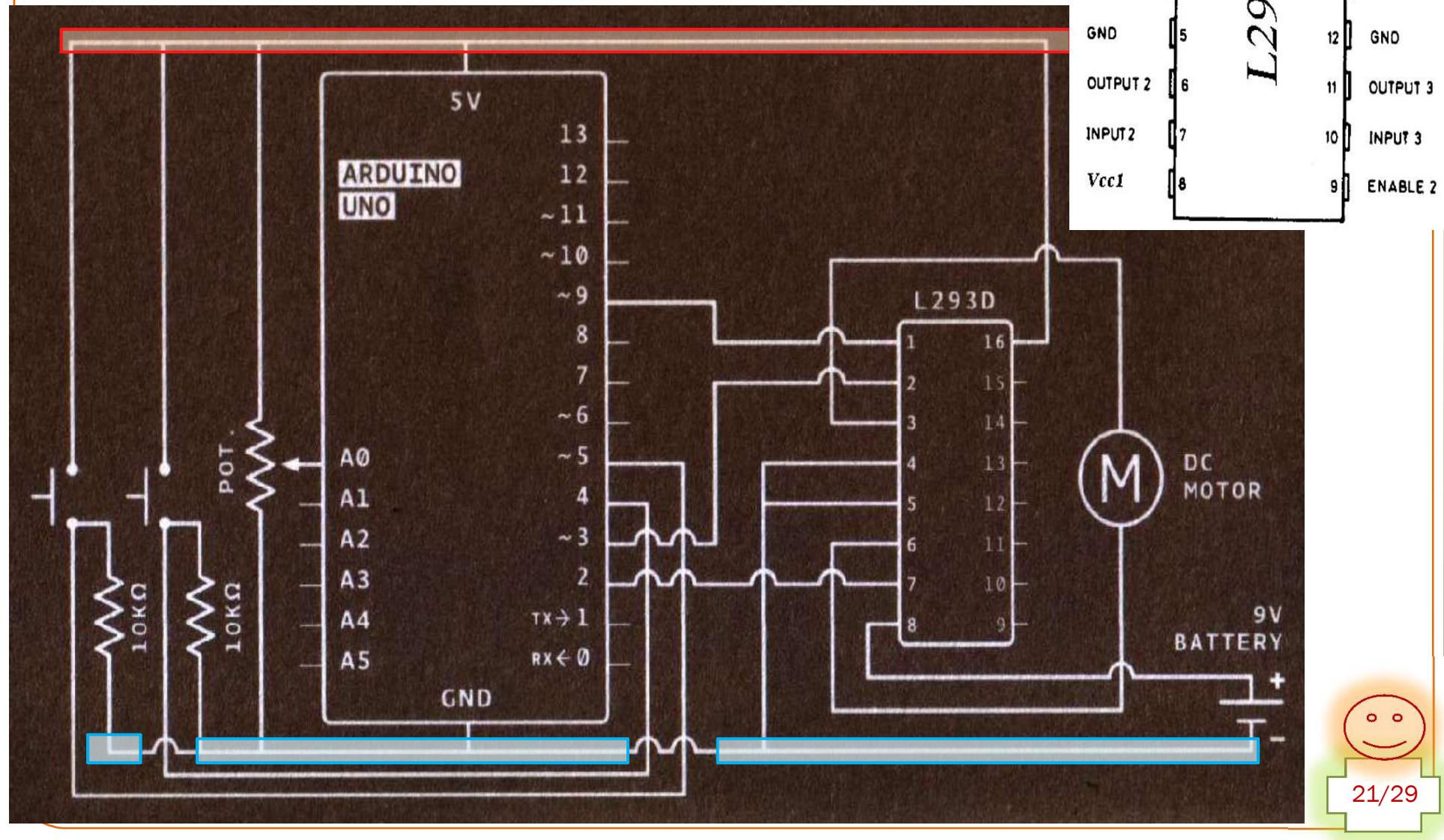
- H橋是由四個電晶體組成，然後由MCU輸入到基極（ Base ）的電位決定電晶體集極（ Collector ）與射極（ Emmiter ）是否導通，可以把它想成一個電子式的開關。



The Top View of the Circuit

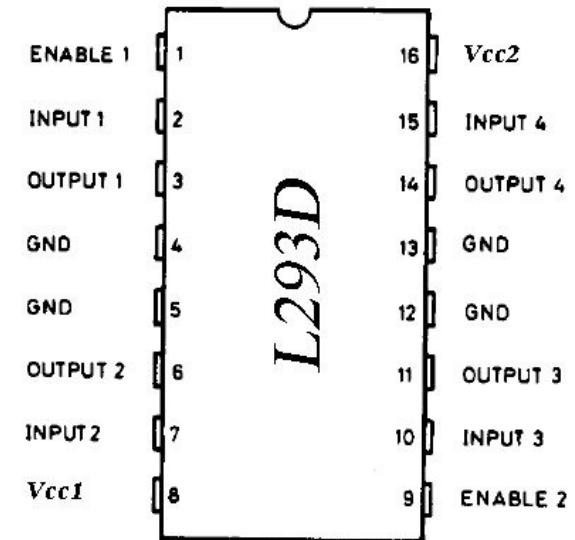


Schematic Diagram



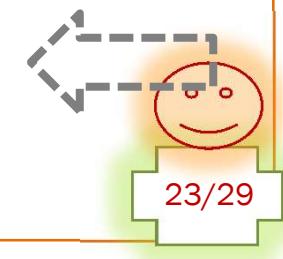
The Codes

```
1 const int controlPin1 = 2;
2 const int controlPin2 = 3;
3 const int enablePin = 9;
4 const int directionSwitchPin = 4;
5 const int onOffSwitchStateSwitchPin = 5;
6 const int potPin = A0;
7 int onOffSwitchState = 0;
8 int previousOnOffSwitchState = 0;
9 int directionSwitchState = 0;
10 int previousDirectionSwitchState = 0;
11 int motorEnabled = 0;
12 int motorSpeed = 0;
13 int motorDirection = 1;
```



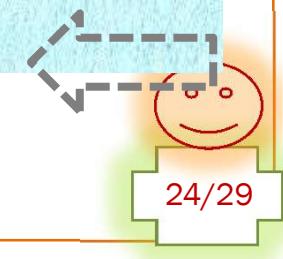
The Codes (cont.)

```
14 void setup(){
15     pinMode(directionSwitchPin, INPUT);
16     pinMode(onOffSwitchStateSwitchPin, INPUT);
17     pinMode(controlPin1, OUTPUT);
18     pinMode(controlPin2, OUTPUT);
19     pinMode(enablePin, OUTPUT);
```



The Codes (cont.)

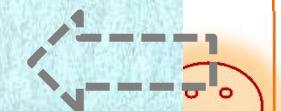
```
20 digitalWrite(enablePin, LOW);  
21 } //注意：後面對enablePin pin是用analogWrite來啟用PWM  
  
22 void loop(){  
23     onOffSwitchState =  
        digitalRead(onOffSwitchStateSwitchPin);  
24     delay(1);  
25     directionSwitchState =  
        digitalRead(directionSwitchPin);  
26     motorSpeed = analogRead(potPin)/4;
```



The Codes (cont.)

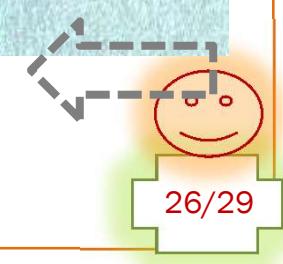
```
27  if(onOffSwitchState != previousOnOffSwitchState){  
28      if(onOffSwitchState == HIGH){  
29          motorEnabled = !motorEnabled;  
30      }  
31  }
```

```
32  if (directionSwitchState !=  
     previousDirectionSwitchState) {  
33      if (directionSwitchState == HIGH) {  
34          motorDirection = !motorDirection;  
35      }  
36  }
```



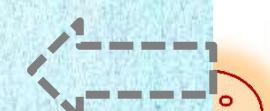
The Codes (cont.)

```
37  if (motorDirection == 1) {  
38      digitalWrite(controlPin1, HIGH);  
39      digitalWrite(controlPin2, LOW);  
40  }  
  
41  else {  
42      digitalWrite(controlPin1, LOW);  
43      digitalWrite(controlPin2, HIGH);  
44  }
```

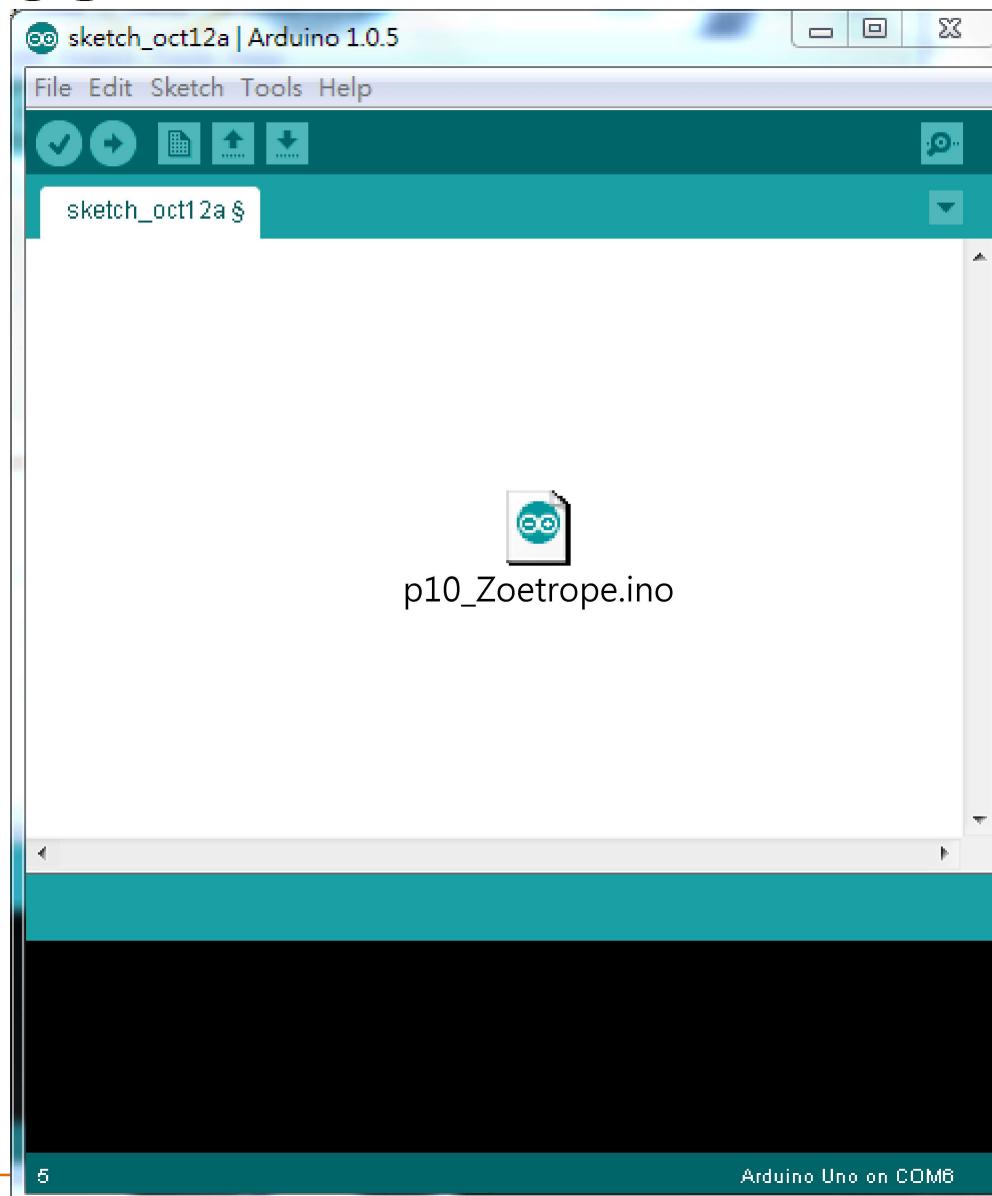


The Codes (cont.)

```
45  if (motorEnabled == 1) {  
46      analogWrite(enablePin, motorSpeed);  
47  }  
48  else {  
49      analogWrite(enablePin, 0);  
50  }  
  
51  previousDirectionSwitchState =  
     directionSwitchState;  
52  previousOnOffSwitchState = onOffSwitchState;  
53 }
```

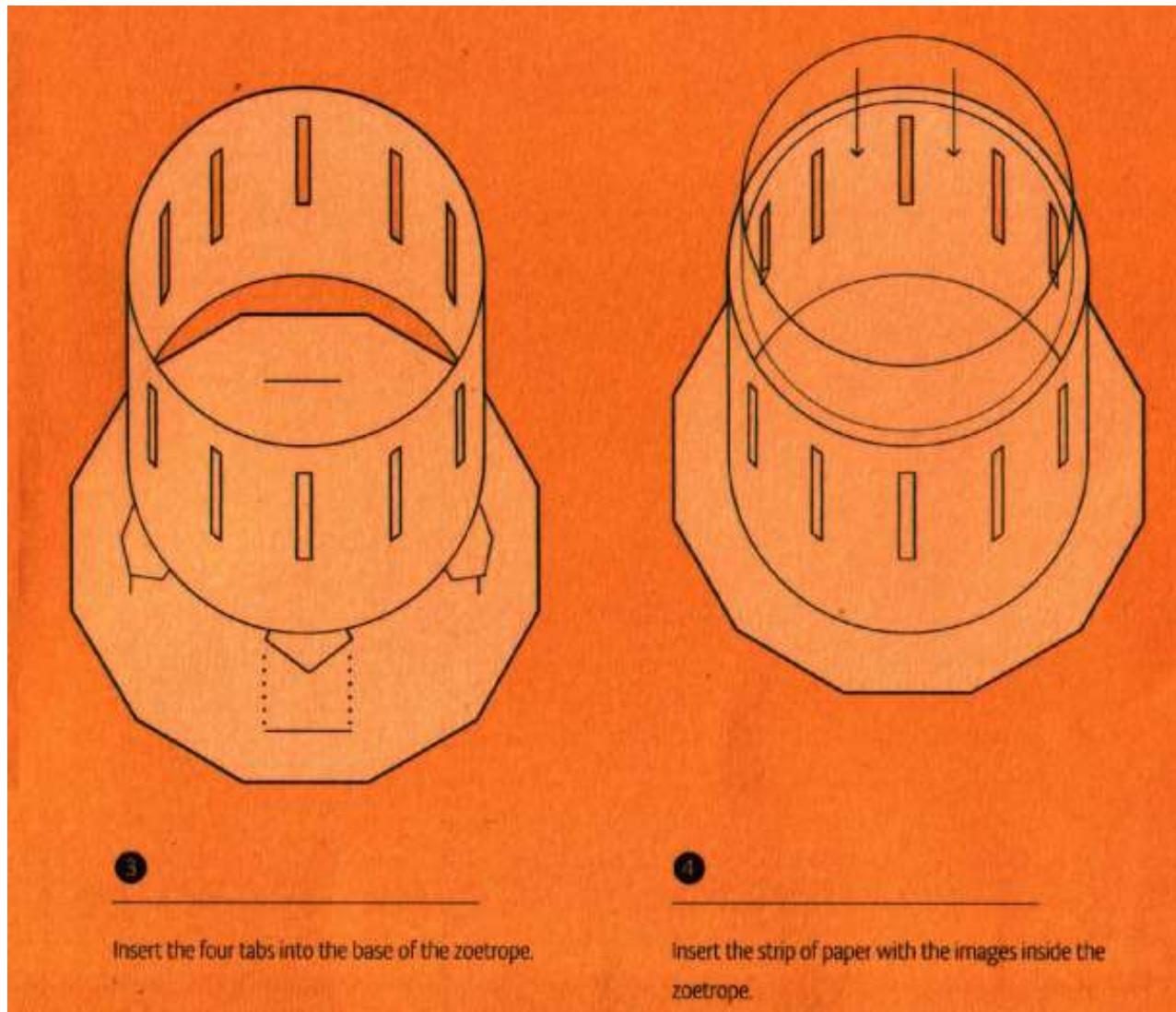


Codes



28/29

Finish Up



29/29