



Robocon/ROV Training

EZ (Easy) Arduino Programming (Tinkercad)

Tasks for you



State Transit

Design sequential logic circuits!



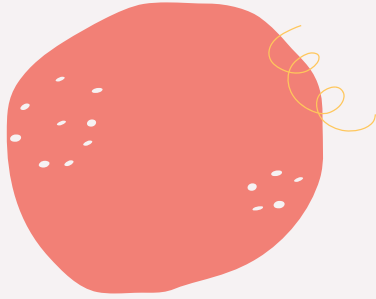
Simple I/O

Learn to use input and output devices



Communication

Learn to use TX/RX

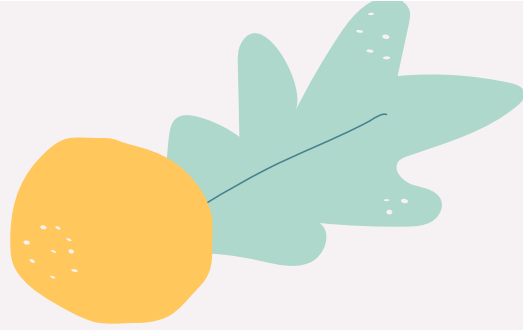


Overview

- Basic Arduino Program
- Simple I/O
- Simple Finite State Machine
- Simple Communication

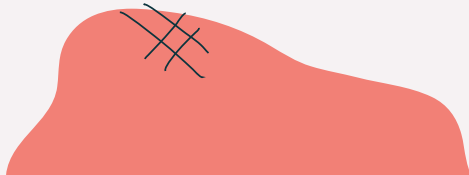
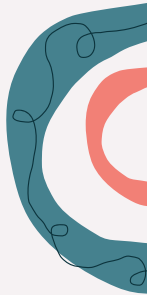
If you are familiar with Arduino, you can skip part 0!





00.

First program - Blink





Learn from it

```
//Structure  
void setup() {  
    //runs once  
    //setting & initial state  
}  
  
void loop() {  
    //runs over and over again forever  
}
```



Print a message

```
Serial.begin(Baud);  
Serial.print("Arduino:");  
Serial.println("Hello.");
```

***Baud** is defined as the number of signal units per second

Serial (Important)

Used for communication between the Arduino board and a computer or other devices

<http://arduino.cc/reference/en/language/functions/communication/serial/>





Add a variable and Print it

```
bool isOn; //declare a variable called isOn which is a boolean type  
isOn = true; //assign a value to isOn  
Serial.print("isOn: ");  
Serial.println(isOn);
```

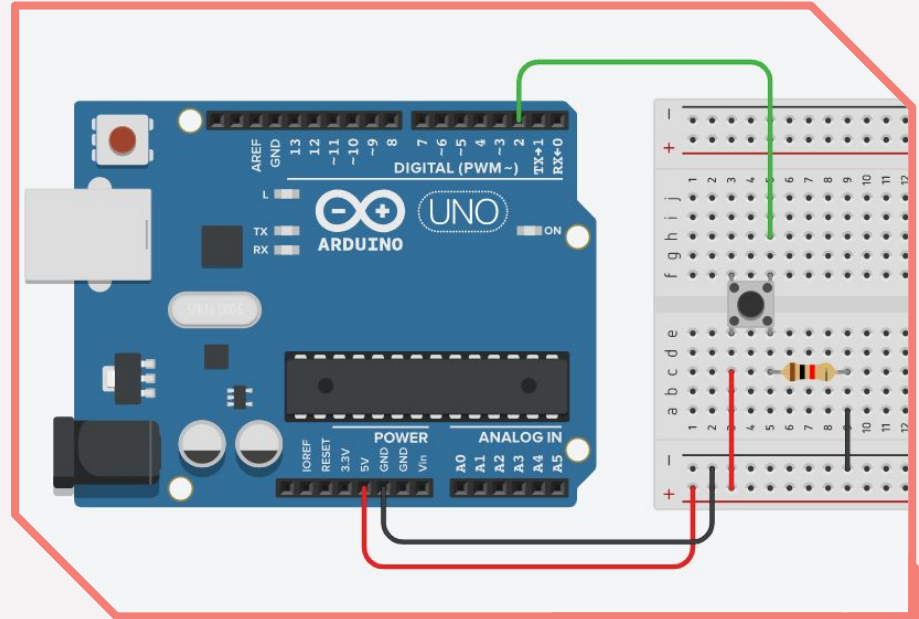
*We have other different variable type to use. E.g. int (integer), float (float point)...

Control led with a button

Pull Down
(You can also use pull up)
Prevent floating

Other Options:

1. Internal pull-up
(`pinMode(pin, INPUT_PULLUP);`)
2. Use `digitalWrite` to
turn on pullup resistors
(`digitalWrite(pin, HIGH);`)




Source: <https://www.arduino.cn/thread-13186-1-1.html>



Control led with a button

Read signal from the button

```
const int buttonPin = 2;  
int buttonState = 0;  
pinMode(buttonPin, INPUT);  
  
buttonState = digitalRead(buttonPin);  
Serial.println(buttonState);
```





Control led with a button

Use “if-else” to control the led with button.

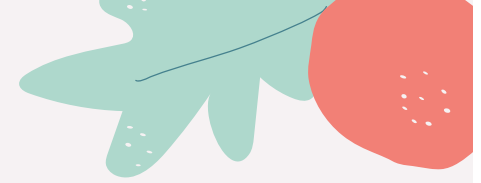
```
if (buttonState == HIGH) {  
    // turn LED on:  
    digitalWrite(ledPin, HIGH);  
} else {  
    // turn LED off:  
    digitalWrite(ledPin, LOW);  
}
```

=>

```
digitalWrite(ledPin, buttonState);
```



Conclusion



- Know the structure of an Arduino program (setup & loop)
- Set the pin mode
 - `pinMode(pinNum,mode);`
- Use Serial Monitor
 - `Serial.println("Hello World.");`
- Write a digital signal
 - `digitalWrite(pinNum, HIGH/LOW);`
- Assign a variable
- Use if-else
- Use a button

The background features abstract, organic shapes in red, orange, and green. The red shape is in the top left, the orange shape is in the top right, and the green shape is in the bottom right. The text is positioned on the left side of the slide.

01.

Simple I/O Program

Get feedback & Control Something!



Digital pin? Analog pin?

In Arduino IDE, we have four basic functions:

`digitalRead()` for reading HIGH / LOW.

`digitalWrite()` for control HIGH / LOW output of pins.

`analogRead()` for reading analog signal. It will return 0 – 1023 value.

`analogWrite()` for control analog output of pins. It can be used on **PWM** pins.

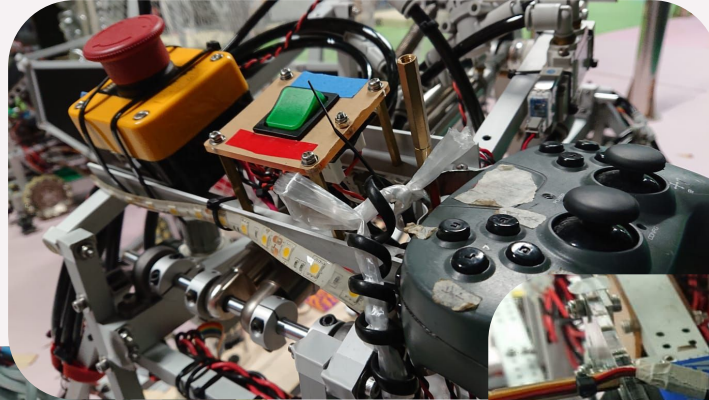
So far, we can see there **only are HIGH or LOW value in digital pin**, which mean 5V(3.3V for other different board) and 0V respectively.

But in analog pin, a **analog signal can provide more information and control on voltage**, a value ranging from 0 to 1023 (0V-5V).



Input Device

1. Distance Sensor
2. Ultrasound (TRASH)
3. Gyro
4. Others



An abstract graphic on the left side of the slide. It features a teal shape with a dark blue curved line and a black 'X' mark. Next to it is a yellow crescent shape. At the bottom right, there is a large red shape with some white dots.

Input Device

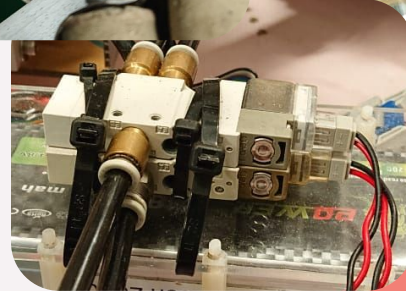
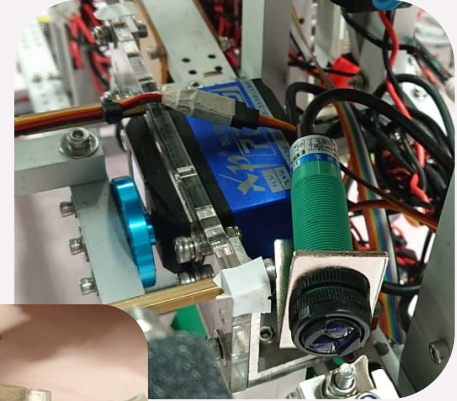
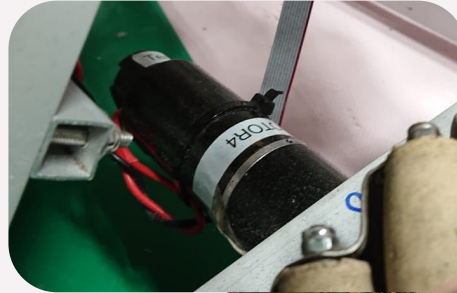
Read signal from different pins.

`digitalRead()` for reading HIGH / LOW.

`analogRead()` for reading analog signal. It will return 0 – 1023 value.

Output Device

1. Solenoid
2. Servo
3. Motor
4. Others

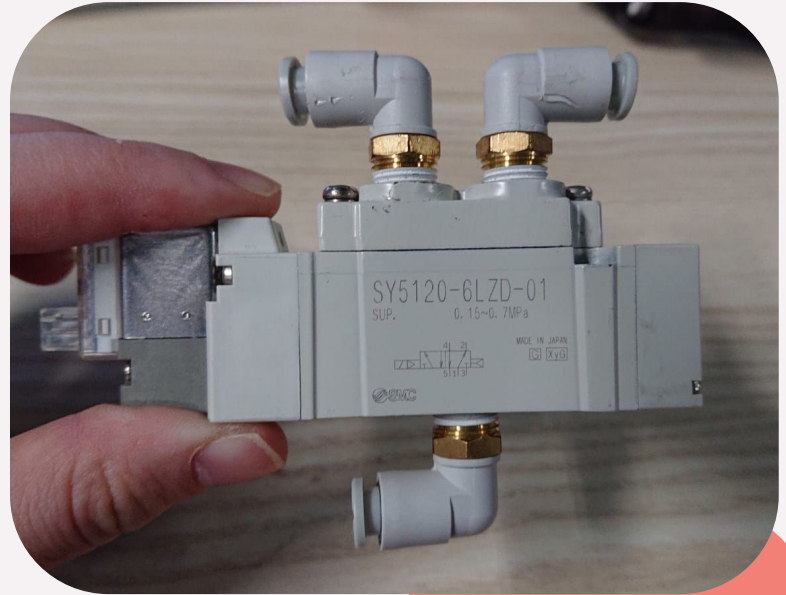


Control pneumatic components



A gripper from ROV 2019

+



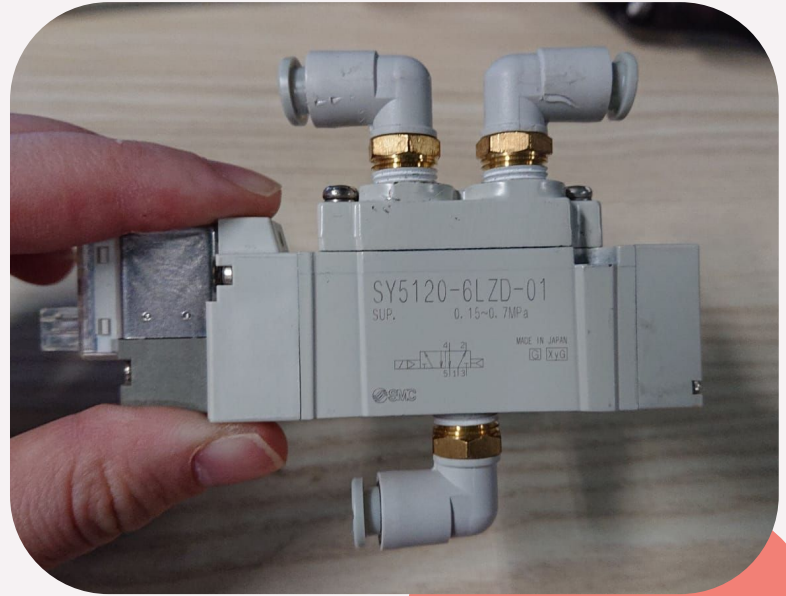
Pneumatic Solenoid Valve

Control pneumatic components



A gripper from ROV 2019

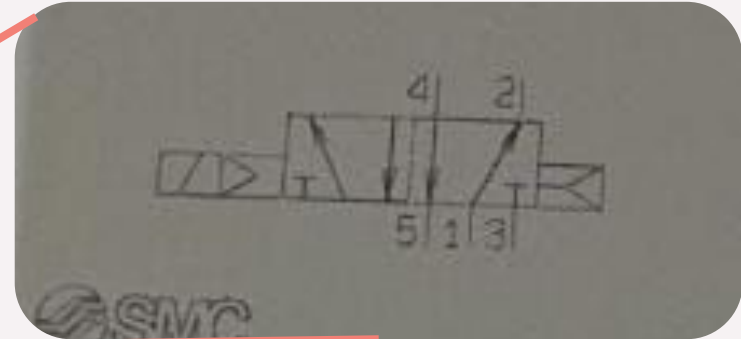
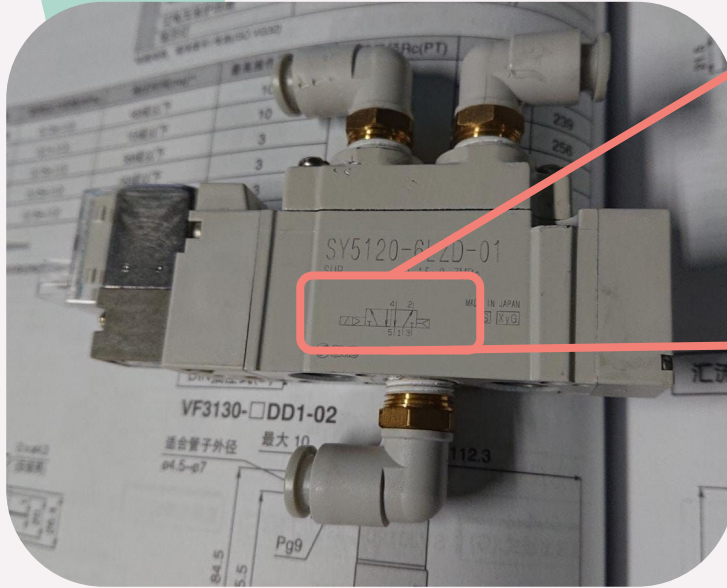
+



Pneumatic Solenoid Valve

Control pneumatic components

- What is Pneumatic Solenoid Valve (電磁閥)?



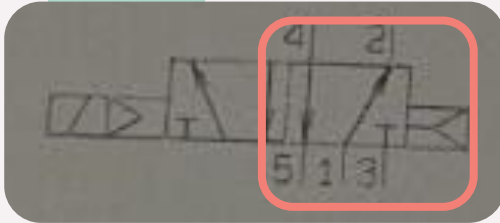
12V -> Change State

<https://www.youtube.com/watch?v=JbK1yNCY8tA>

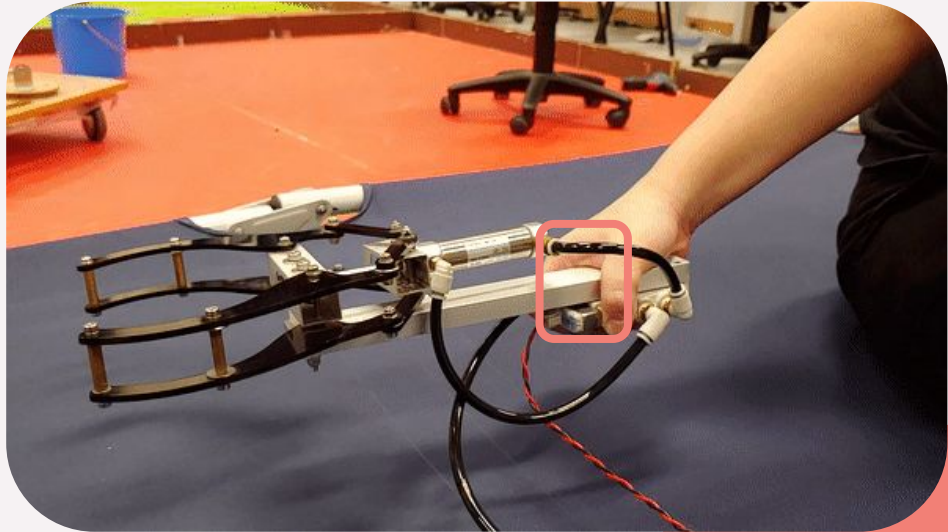
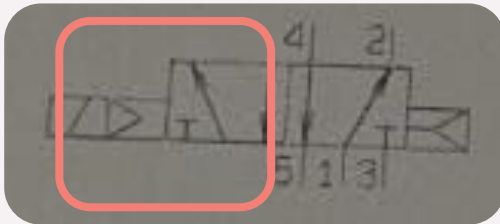
Control pneumatic components

- What is Pneumatic Solenoid Valve (電磁閥)?

Initial State



12V -> Change State



Control pneumatic components

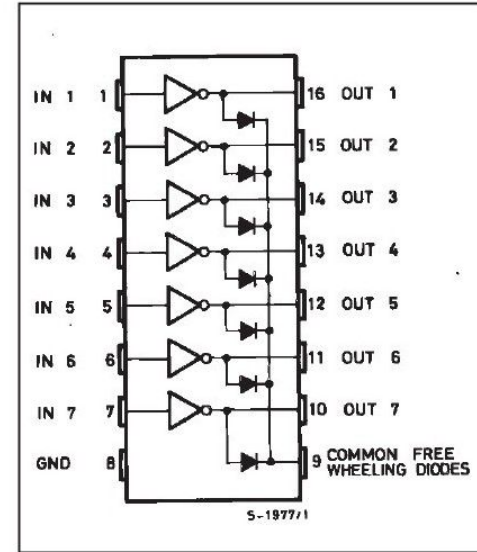
- Pneumatic Solenoid Valve (12V)
- Arduino (5V)
- **How do we control it with Arduino?**



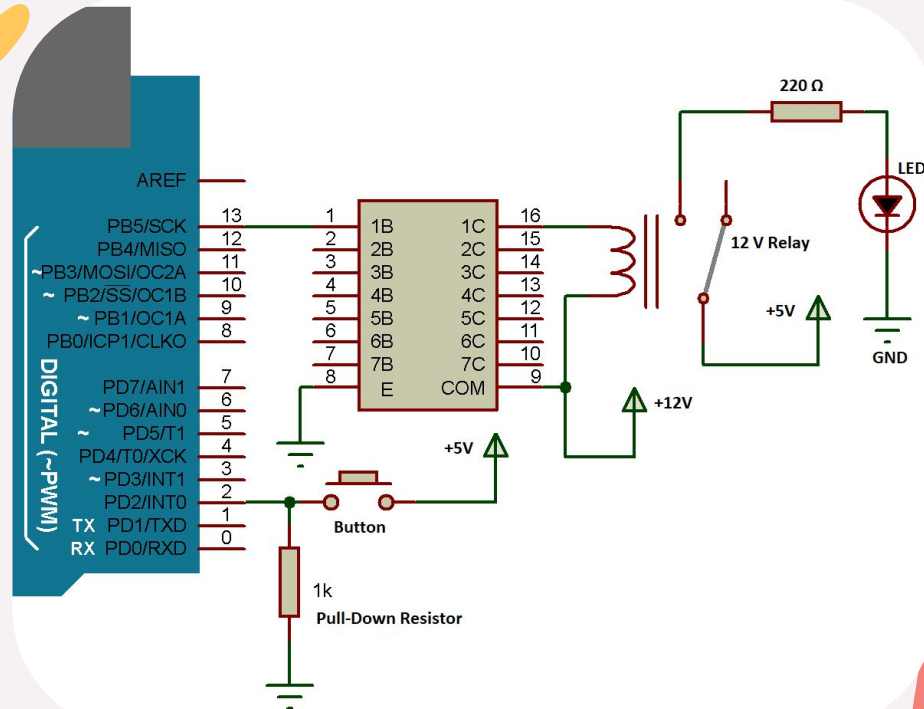
ULN2003AN

- We use ULN2003AN to control pneumatic components (12V)
- What is ULN2003AN?
 - An array of seven NPN **Darlington transistors** capable of 500 mA, 50 V output (From Wiki)
- It is okay that you don't understand, just treat it as a chip that help you to **control a high-voltage component with low-voltage.**

PIN CONNECTION



ULN2003AN



Source:

<https://pijaeducation.com/arduino/relay/interfacing-of-relay-with-arduino-using-uln2003/>

Servo

```
#include <Servo.h>  
Servo myServo;  
myServo.attach(pinNum);  
myServo.write(position);
```



<https://www.arduino.cc/reference/en/libraries/servo/>

Task 1: Simple I/O

Digital



Pushbutton

Control



LED

Analog



Potentiometer

Control



Micro Servo



02.

State Transit

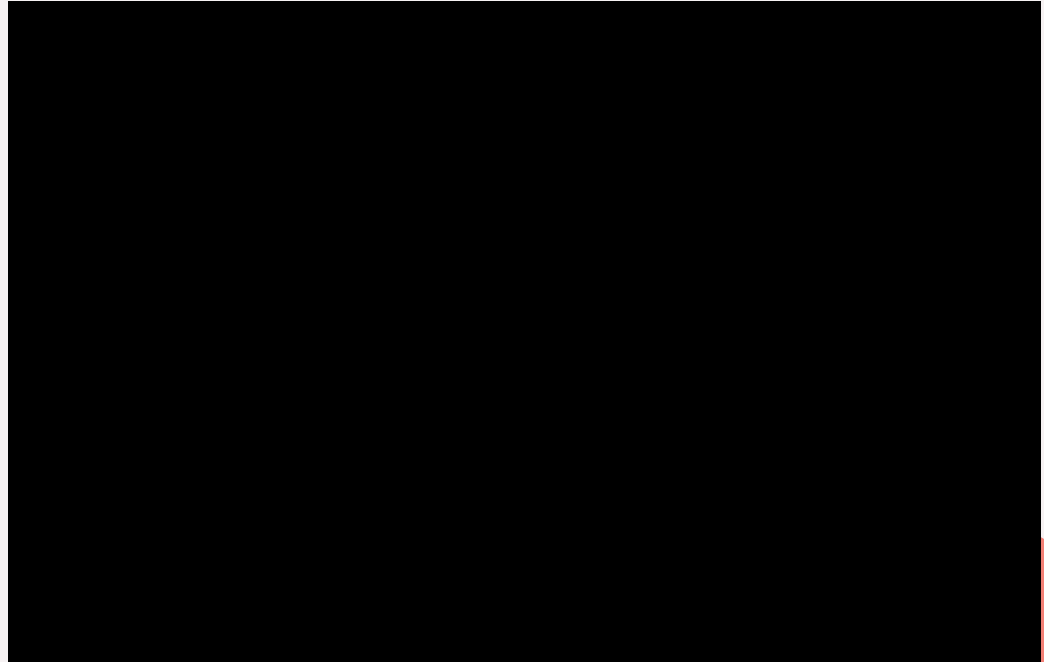
Design sequential logic circuits!



From last 2 years training

Step:

1. Rotary pneumatic actuator
2. Servo
3. Pneumatic cylinder
4. Servo
5. Pneumatic cylinder
6. ...



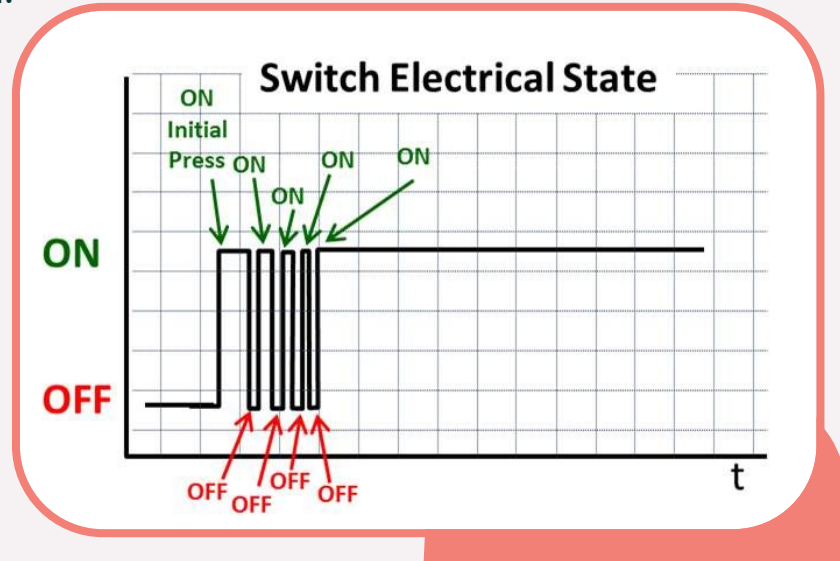
State toggle + debounce

Reason of using toggle:

No one would keep on pressing the button.

Reason of debounce:

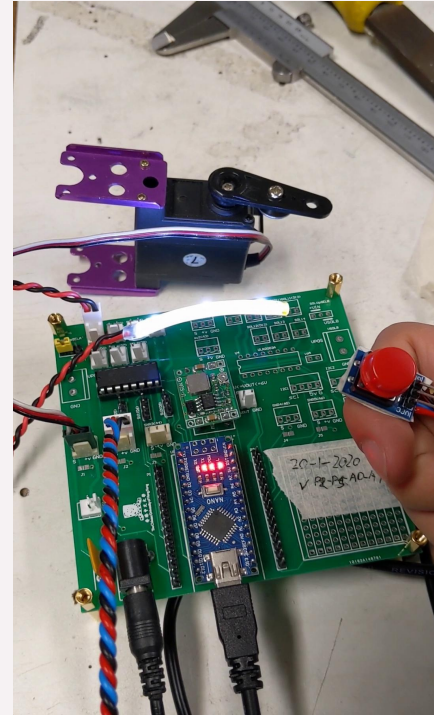
Prevent interference of noise



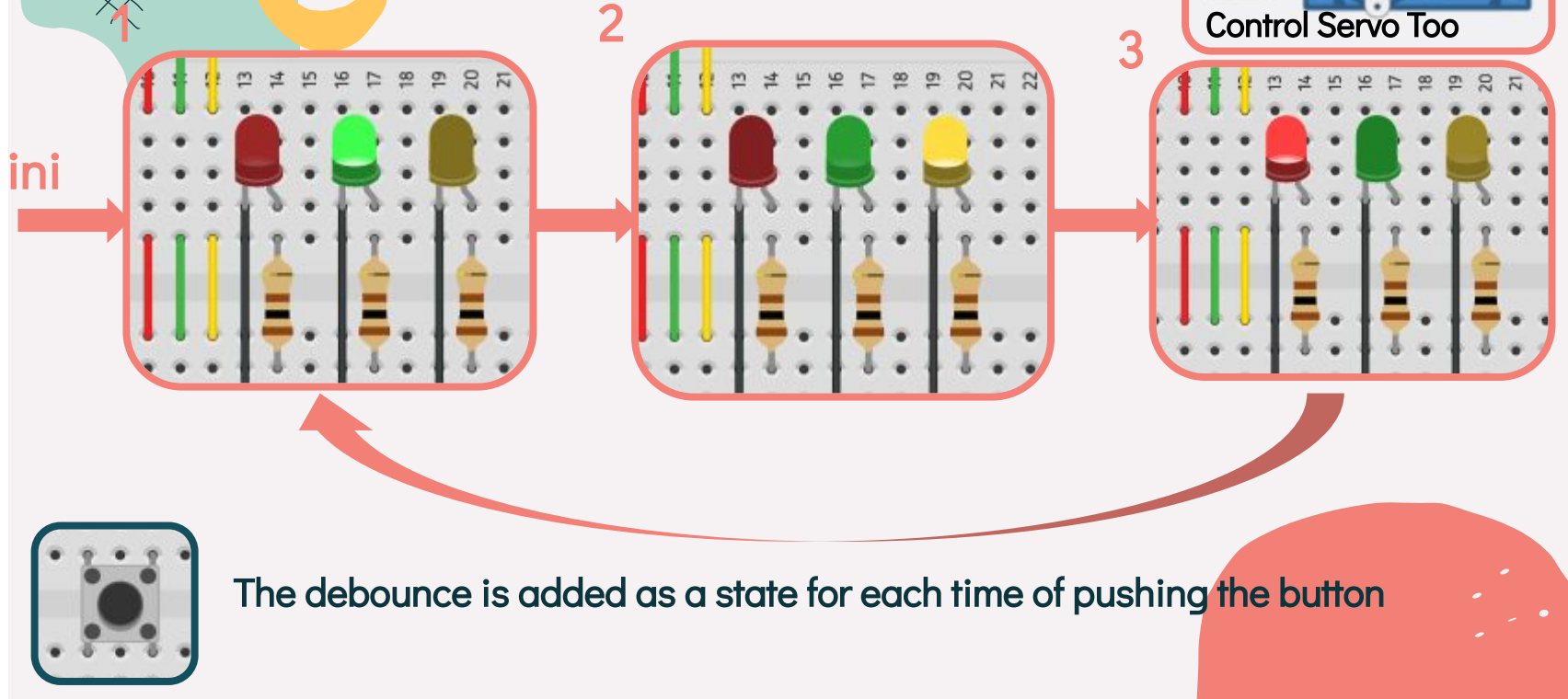
State Transit

```
const int maxState = 4;
int curState = 0;
switch(curState){
    case 0: ....
    case 1: ....
}

//Pseudo code
if(Button){
    curState = (curState + 1) % maxState;
}
```



Task 2: Traffic Light



Finite State Machine

- It can perform **predetermined sequence of actions**, depending on a sequence of events that takes place
- A mathematical **model** of computation
- A set of **states** and how to get from one state to another
- An ideal representation of a computer/ machine
 - It can be in **exactly one of the states** at a time
 - A state describes the computer at any given point
 - A large but **finite** number of states
- It represents legal steps of a process
 - Valid **inputs**
 - Valid **outputs**
 - Some **computation**

Reference: ENGG1100 Introduction Engineering Design, Lecture 5: Finite-State Machine, Term 2, 2019



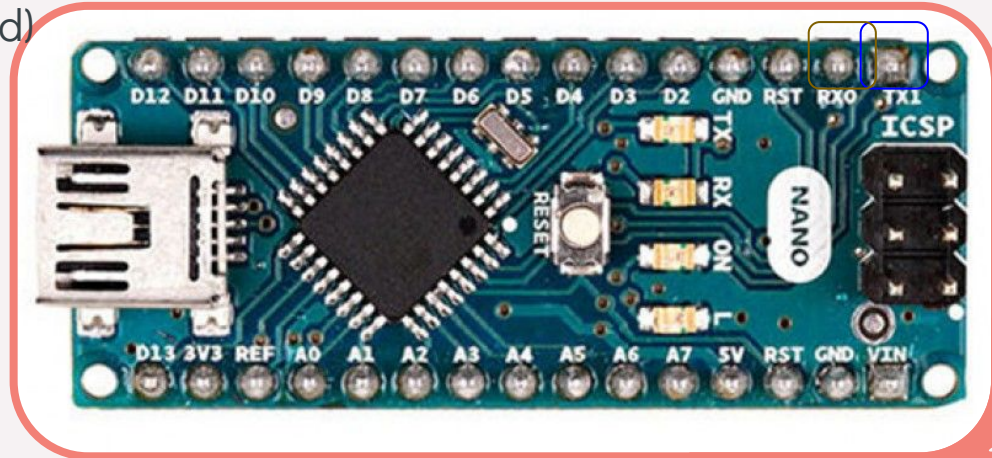
03.

Arduino TX/RX

Communication

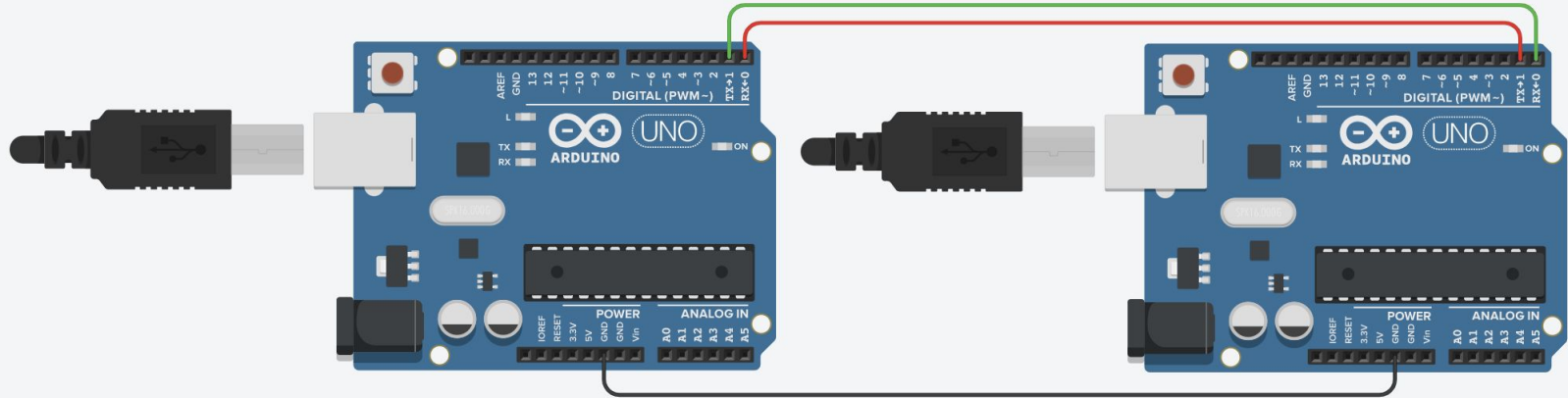
Serial

- Used for communication between Arduino board and other device
- All Arduino boards have at least one serial port
- Serial communication on pins TX/RX uses TTL logic levels (5V or 3.3V depending on the board)



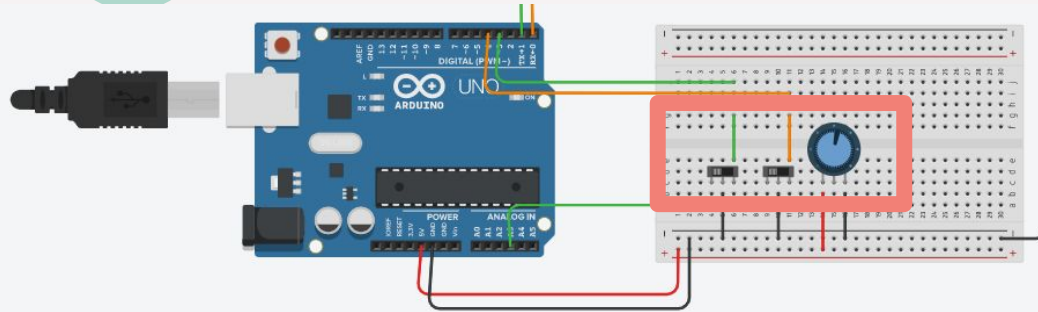
Example: Communication between 2 Arduino

TX => RX
RX => TX



Task 3: Serial Communication

Sender



2 Slide Switch
1 Potentiometer

It will send 6 bytes to another Arduino.

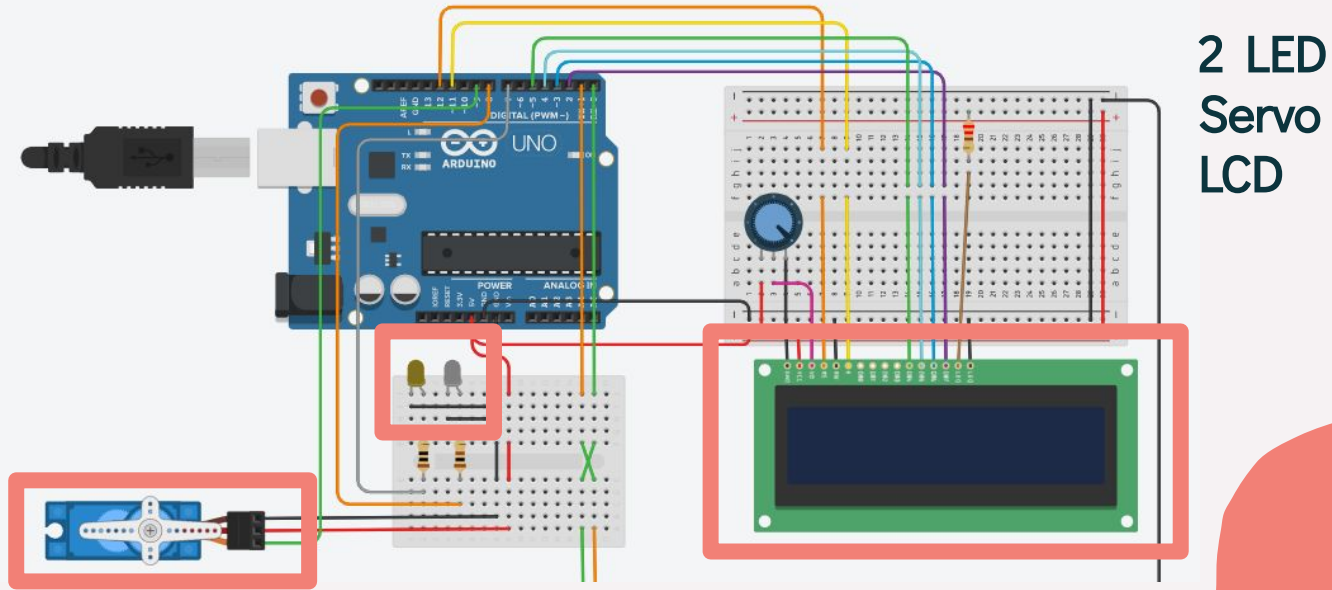
State of 2 switch

111023

Potentiometer Value

Task 3: Serial Communication

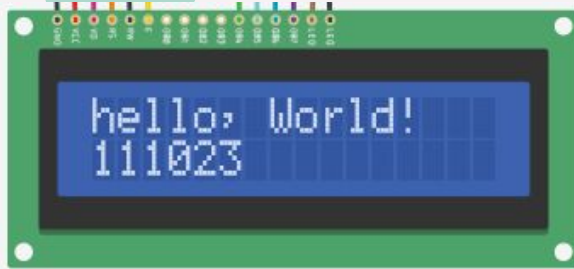
Receiver



2 LED
Servo
LCD

Task 3: Serial Communication

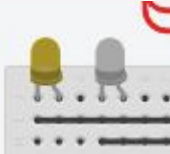
Receiver



Display the message from sender

It will control led and servo.

1: turn on
0: turn off

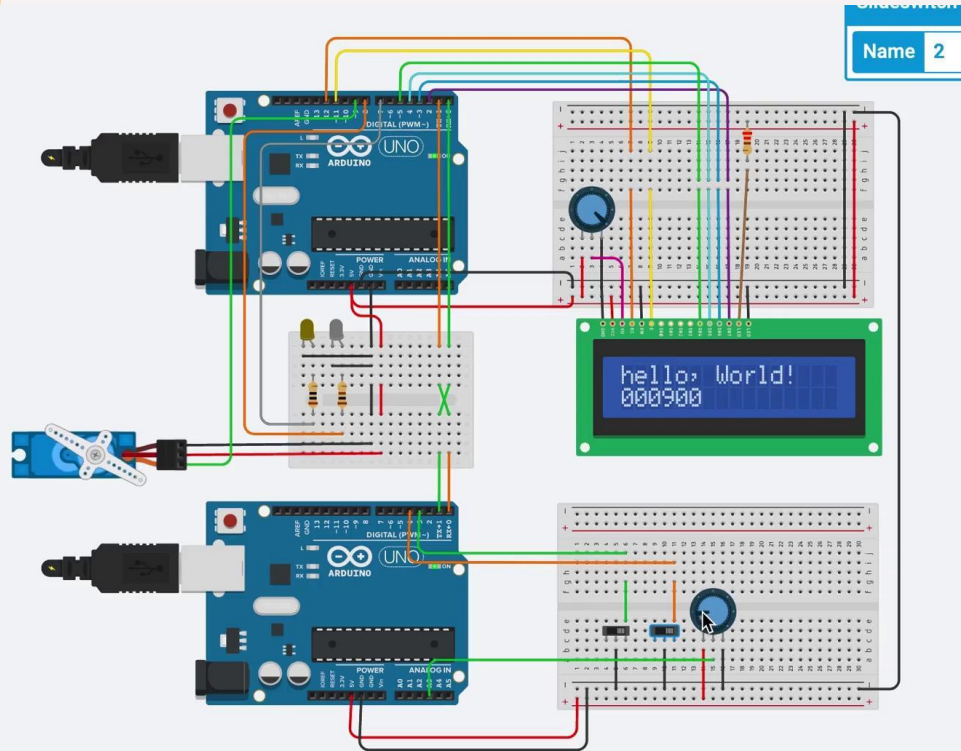


111023

0: the servo at 0 degree.
1023: the servo at 180 degree.



Task 3: Serial Communication



Other Programming



PID Control (Motor)

PS4 Controller

OpenCV (ROV)

...



Reference

An article by our past member

- <https://proj.cse.cuhk.edu.hk/cseblog/introduction-to-arduino%ef%bb%bf/>



Thanks!


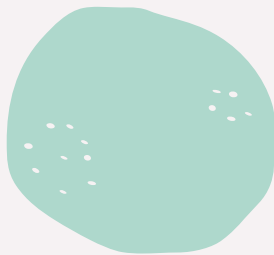


powershuttlecuhk



cuhkrobocon

If you have any problem, feel free to open an issue in the Github repository.



CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik