

# Relay module experiment

## Overview



This is a simple button control relay experiment. Delay() function is not used to eliminate jitter and improve the running efficiency of the program.








## Specification

Null

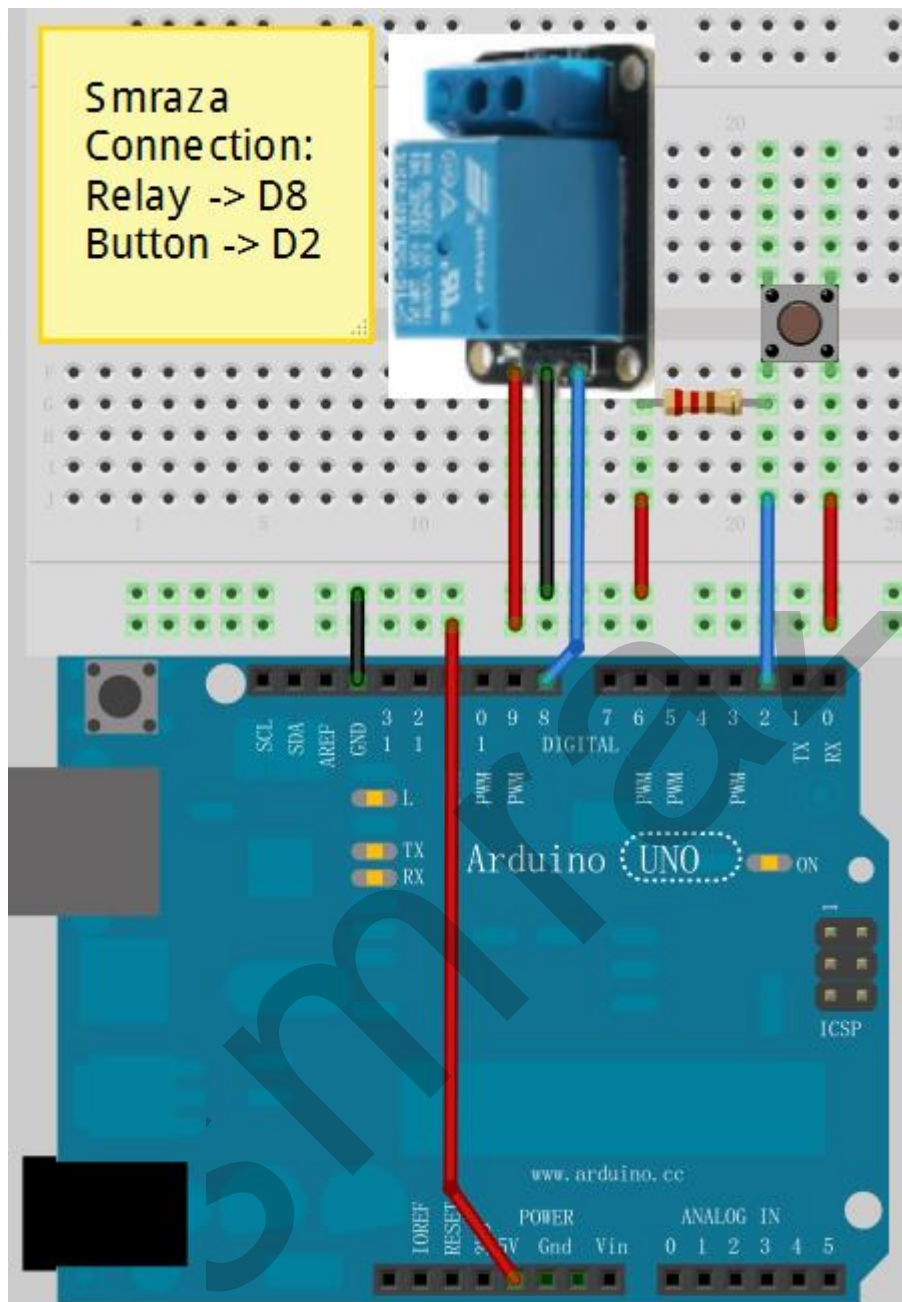
## Pin definition

UNO R3		Relay module
5VC	->	VCC
GND	->	GND
D8	->	IN

## Hardware required

Material diagram	Material name	Number
	Relay module	1
	Button	1
	10KΩ resistor	1
	USB Cable	1
	UNO R3	1
	Breadboard	1
	Jumper wires	Several

## Connection diagram



## Sample code

Note: sample code under the **Sample code** folder

```
const int buttonPin = 2;
```

```
const int relay = 8;
```

```
int relayState = HIGH;           //Relay state
```

```
int buttonState;                 //Key state
```

```
int lastButtonState = LOW;      //Last time the key data
```

```
long lastDebounceTime = 0; // Last output pin trigger time
long debounceDelay = 50;   //Elimination of jitter, if the output is not stable increase in
time
```

```
void setup() {
  pinMode(buttonPin, INPUT);
  pinMode(relay, OUTPUT);
  digitalWrite(relay, relayState);
}

void loop() {
  int reading = digitalRead(buttonPin); //Read button data

  if (reading != lastButtonState) {
    lastDebounceTime = millis();
  }

  if ((millis() - lastDebounceTime) > debounceDelay){
    if (reading != buttonState) {
      buttonState = reading;

      if (buttonState == HIGH) {
        relayState = !relayState;
      }
    }
    digitalWrite(relay, relayState);
    lastButtonState = reading;
  }
}
```

### Language reference

**Tips** : click on the following name to jump to the web page.

If you fail to open, use the Adobe reader to open this document.

[const](#)

[millis\(\)](#)

### Application effect

When the button is pressed, the state of the relay will be changed.