## SKU:TEL0140 (https://www.dfrobot.com/product-2431.html)

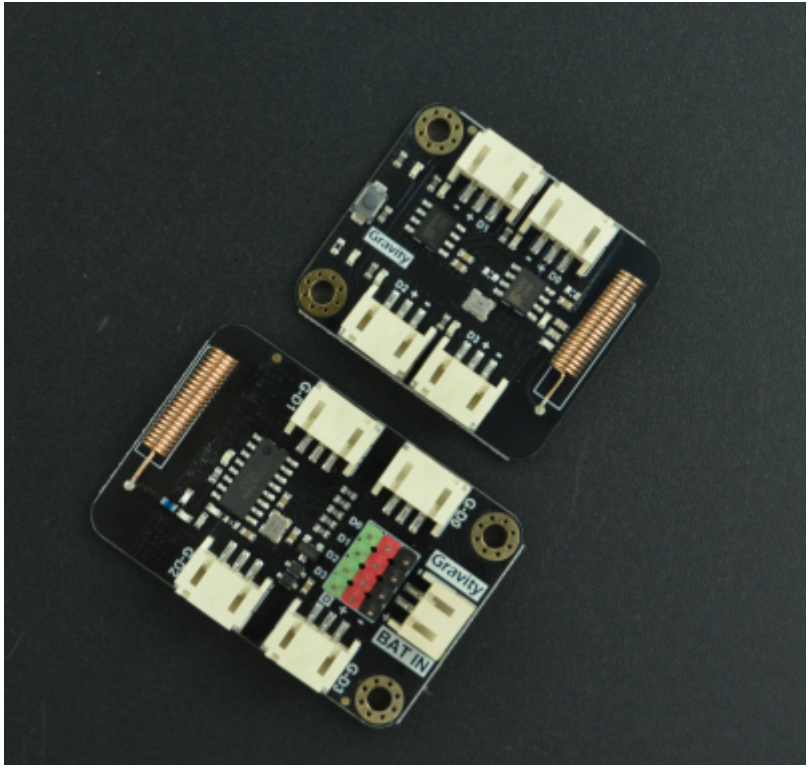
(https://www.dfrobot.com/product-2431.html)

# Introduction

When building a project that needs to transmit some simple signals wirelessly, have you been bothered by the any of the following problems: Bluetooth's cumbersome AT command configuration and high cost, Wifi's signal limitations and high power consumption, IR emission angle and inability to penetrate through the walls...

This Gravity: Digital Wireless Swtich might be just what you need! Based on 433Mhz RF communication technology, the module includes a transmitter and receiver and features the advantages of simple operation, high scalability, strong penetration, and ultra-low standby power

advantages of simple operation, high scalability, strong penetration, and ultra-low standby power consumption. It is able to transmit digital signals and compatible with any controller that can read and write digital signals. These modules are used on a wide variety of applications that require wireless control, such as, remote control, wireless doorbell, wireless signal transmission, upgrading wired buttons to wireless buttons, etc.




The module transmitter can be used without connecting to the controller. After connecting the battery, you can directly use the digital sensor or the switch button as the "trigger button" for signal transmission. Of course, you can also connect the module to a controller and control the signal transmission through the controller's digital pins. The product adopts the EV1527 encoding format and the four-digit key value code can be combined into 15 different states; the

receiver has a corresponding pairing function to ensure that only the paired transmitting device can control the receiver, the receiver supports four working modes: inching, latching, self-locking, and interlocking. It can be paired with EV1527 coded transmitters. One receiver can be paired with up to 32 transmitters. The transmitter and receiver support "one-transmit and multiple-receive" or "one-receive and multiple-transmit" after pairing. Please see the examples part for specific usage.

## Application

- wireless door bell
- remote control
- Deployed as a sensor signal acquisition node
- Wired button upgrade wireless button

## Features

- Simple operation and strong scalability

- Strong penetration
- Ultra-low standby power consumption
- 15 button states of the transmitter

- Pairing function for the receiver
- Support one-transmit and multiple-receive/one-receive and multiple-transmit
- Multiple working modes: inching, latching, self-locking, and interlocking
- Digital signal, support 3.3-5V power input

## Specification

- Working voltage: 3.3 ~ 5.0V DC
- Input signal: digital
- Number of interfaces: 4
- Working frequency: 433Mhz
- Circuit board size: transmitting end 47×32mm, receiving end 37×32mm
- Mounting hole size: inner diameter 3.1mm/outer diameter 6mm

| | | |
|---|---|---|

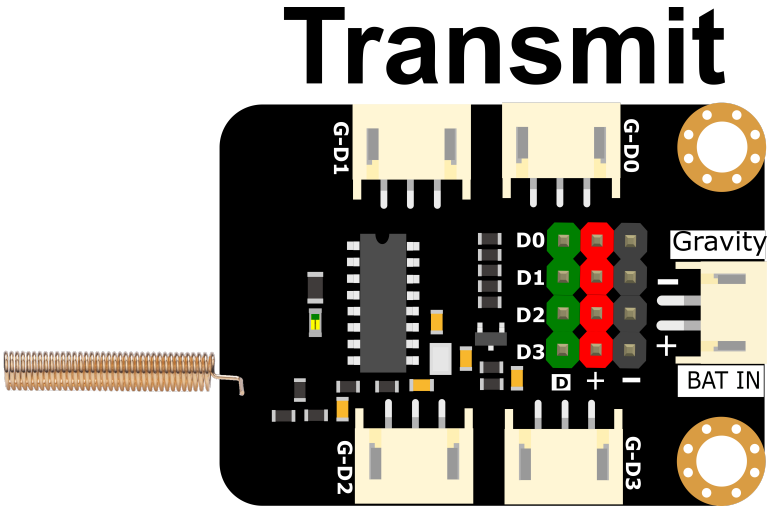
| Operating current<br>Operating current | Transmitter<br>Transmitter | Receiver<br>Receiver |
| --- | --- | --- |
| 5V power supply, working | 10mA | 6mA |
| 5V power supply, standby | <10uA | 3mA |

| | | |
| --- | --- | --- |
| 3.3V power supply, working | 8mA | 5mA |
| 3.3V power supply, standby | <10uA | 3mA |

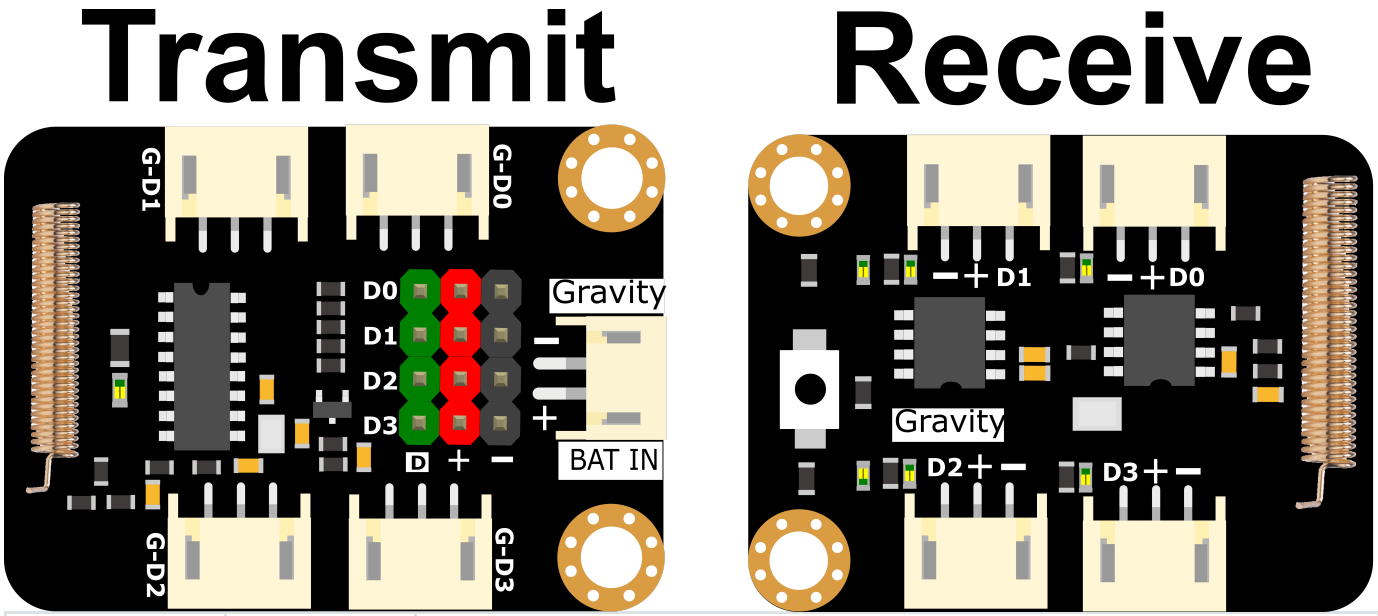
| Status | Transmission distance |
| --- | --- |
| Office， no walls | **15M** |
| Office， 1 walls | **14~15M** |
| Office ， 2 walls | **13~14M** |

- Distance test description
  - The transmitter is connected to a 3.7V lithium battery
  - Receiver connects to UNO controller 5V power supply

- Try to keep the transceiver module at the same height, about 0.5M above the ground
- The antenna stays in the default position
- There is no great attenuation after the signal passes through the wall, and the transmission distance is reduced by about 0.5M each time it passes through the wall
- If the transmitter antenna is re-welded to the state shown in the figure below, the signal transmission distance can be increased to about **25M** (if there is no strong welding ability, and the long transmission distance is not required, it is not recommended to re-weld the antenna, because of this state The lower antenna is easily bent and damaged)



## Board Overview




| Transmitter | Name | Description |
| --- | --- | --- |
| Transmitter | D0 ... D3 | The digital signal input terminal powers the external sensor |

| Transmitter | D0 ~ D3 | powers the external sensor |
| --- | --- | --- |
| **Transmitter** | **Name** | **Description** |
| | | the green pin receives the high-level transmit signal |

| Transmitter | G-D0 ~ G-D3 | Digital signal input terminal<br>After connecting to the controller, it can supply power to the transmitter, DC 3.3-5V<br>receive high-level transmission signal |
| --- | --- | --- |
| Transmitter | BAT IN | External power input 3.3-5V |
| Receiver | D0 ~ D3 | Digital signal output terminal<br>the default low level, the received signal is high level |
| Receiver | Button | Pairing/mode switching |

## Receiver operating instructions

**Pairing**

## Pairing

- step1. Click the button of the receiver(Press and hold for <0.5S), the blue indicator light next to the button will light up, and the receiver waits for pairing.;

- step2. Make any channel of the transmitter continuously transmit signals within six seconds, until the blue indicator light of the receiver flashes 3 times, then the pairing is successful;
- step3. The blue indicator light flashes three times after the pairing is successful, and it will light up again. At this time, the next transmitter can be paired. If there is no next transmitter to be paired, it will automatically exit the pairing mode after waiting for 6S;
- Note: The pairing waiting time is 6S, if the pairing signal from the transmitter is not received within 6S, the pairing mode will be automatically exited.
- Note:**After the pairing is successful, the transmitter must re-transmit the signal and the receiver can receive it**. If it fails, repeat the previous two steps.
- Note: You only need to pair any one of the transmitter D0~D3 signals, and the other three will automatically match.

## Mode switch

**Note: The receiver is in inching mode by default, the following examples all take inching mode as an example. If there is no requirement for using other modes, you can skip this step.**

as an example. If there is no requirement for using other modes, you can skip this step

| Mode | Description |
| --- | --- |
| Latching | After D0 receives the signal once, it stays high until D1~D3 receive the signal |
| Self-locking | Each time D0 receives a signal, the corresponding output state is inverted once, the same is true for D1~D3 |
| Inching | D0 receives the signal and outputs high level, but does not receive the signal low, the same is true for D1~D3 |
| Interlocking | When receives the signal D0, D0 stays at a high level, and all the others are low. The same applies to D1~D3. |

Press and hold the button for 0.5~1.5 seconds, then release it, the blue indicator light flashes twice, indicating that you have entered the mode switching state, and then you can enter different modes according to the different times of pressing the button within 6 seconds:

Press once to enter the latching mode;

Press twice to enter the self-locking mode;

Press twice to enter the self-locking mode;

Press 3 times to enter the inching mode;

Press 4 times to enter the interlocking mode;

According to the mode you need to enter, press the button for the corresponding number of times, and then hold the button for 0.5~1.5 seconds as a confirmation signal. After letting go, the blue indicator light flashes 2 times to set successfully and enter the corresponding working mode. 0.5-1.5 seconds is relatively short, be careful not to press overtime.

### Clear pairing

The receiver can store up to 32 sets of transmitter codes. When there are more than 32 sets, the first paired code will be overwritten; Clear all paired transmitters: Press and hold the button on the receiving end for more than 4 seconds. After releasing your hand, the blue indicator light flashes twice to successfully clear all paired transmitters; if the clearing fails, repeat the above operation.

## Basic Example 1 - Connects Transmitter to Battery and Sensor

- **This sample is mainly used for demonstration: the transmitter is powered by a battery and**

**the sensor's signal is sent out through RF signal, which simulates the function of a wireless doorbell**

- Take Arduino as an example, the other main controllers use the same principle, only need to connect to the corresponding digital port.
- Take inching mode in the receiver as an example, the principles of other modes are the same.

**Requirements**

- **Hardware**
    - DFRuino UNO R3 (https://www.dfrobot.com/product-838.html) x 1
    - Wireless switch transmitter x 1
    - Wireless switch receiver x 1
    - 3.3-5V power supply (https://www.dfrobot.com/product-437.html) x 1
    - Button (https://www.dfrobot.com/product-73.html) x 1
    - Motion Sensor (https://www.dfrobot.com/product-1140.html) x 1

Other sensor application scenarios

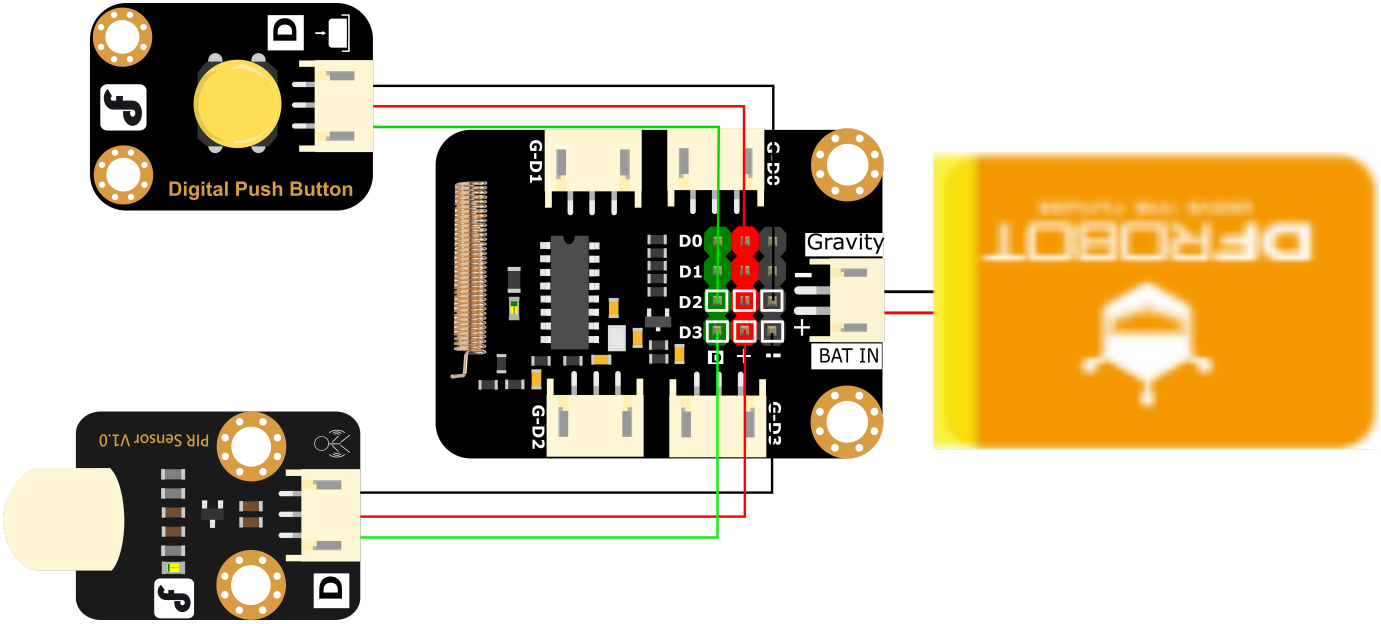
- Crash sensor (https://www.dfrobot.com/product-763.html): position detection-automatic position recovery in scenarios where the closed container is not easy to wire
- Vibration sensor (https://www.dfrobot.com/product-79.html): vibration detection-door opening and closing vibration
- Infrared photoelectric switch (https://www.dfrobot.com/product-1653.html): counting detection-number of people/workpiece counting
- Other digital sensors(Play with your imagination)

**Note: The sensor can be selected arbitrarily, as long as it is digital. The button and motion sensor are used here as a function demonstration to simulate the function of a wireless doorbell.**

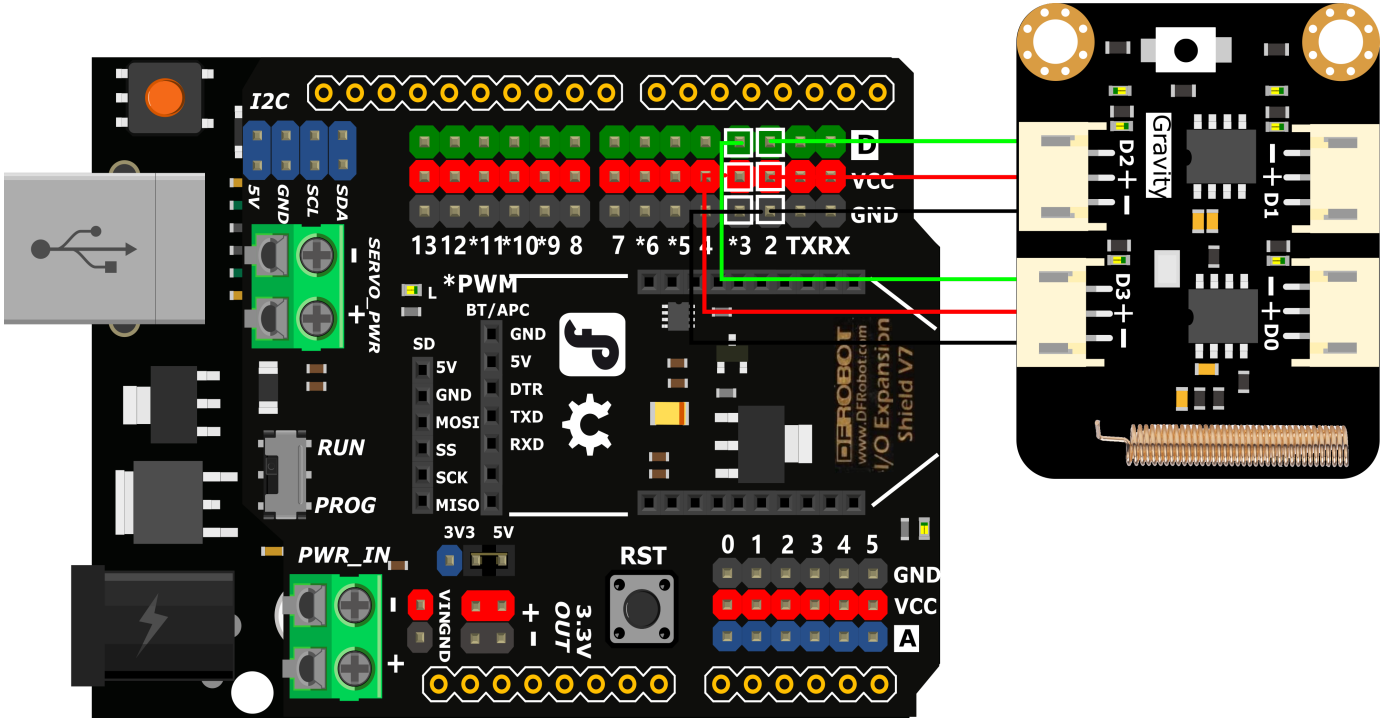
- **Software**
    - Arduino IDE (https://www.arduino.cc/en/Main/Software)

**Connection**

**The transmitting end:**




** Receiving end: **


I2C
5V
GND
SCL
SDA
SERVO_PWR
RUN
PROG
PWR_IN
VINGND
3.3V OUT
3V3
5V
RST
*PWM
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
13 12 *11 *10 *9 8 7 *6 *5 4 *3 2 TXRX
D
VCC
GND
0 1 2 3 4 5
GND
VCC
A
DFROBOT
www.DFRobot.com
I/O Expansion Shield V7
Gravity
D2
D1
D3
D0

## Sample code

```
#define Button_D2 2//Arduino
#define Button_D3 3//Arduino
//#define Button_D2 D2 //ESP32
//#define Button_D3 D3 //ESP32

void setup() {
    Serial.begin(115200);
    pinMode(Button_D2, INPUT);
    pinMode(Button_D3, INPUT);
}
void loop() {
    if (((digitalRead(Button_D2)==1) && (digitalRead(Button_D3)==1))) {
        Serial.println("Someone presses the doorbell");
        delay(3000);
    }
    else if (((digitalRead(Button_D2)==0) && (digitalRead(Button_D3)==1))) {
        Serial.println("Someone passes by but does not press the doorbell");
        delay(100);
    }
```

```
    }
    else if (((digitalRead(Button_D2)==1) && (digitalRead(Button_D3)==0))) {
        Serial.println("Someone has waited for a while and then press the doorbell");
        delay(3000);
    }

    delay(100);
}
```

**Result:**

**When someone passes by and presses the button, the serial port prints "Someone presses the doorbell"; when someone passes by but does not press the doorbell, the serial port prints "Someone passes by but does not press the doorbell"; when someone comes to the door and stands for a while Press the doorbell, and the serial port prints "Someone has waited for a while and then press the doorbell". **

## Basic Example 2 - Connect Transmitter to a Controller

- **This sample is mainly used for demonstration: power the transmitter through the controller, and control the transmitter to transmit signals**

- Take Arduino as an example, the other main controllers use the same principle, only need to connect to the corresponding digital port.

- Take inching mode in the receiver as an example, the principles of other modes are the same.

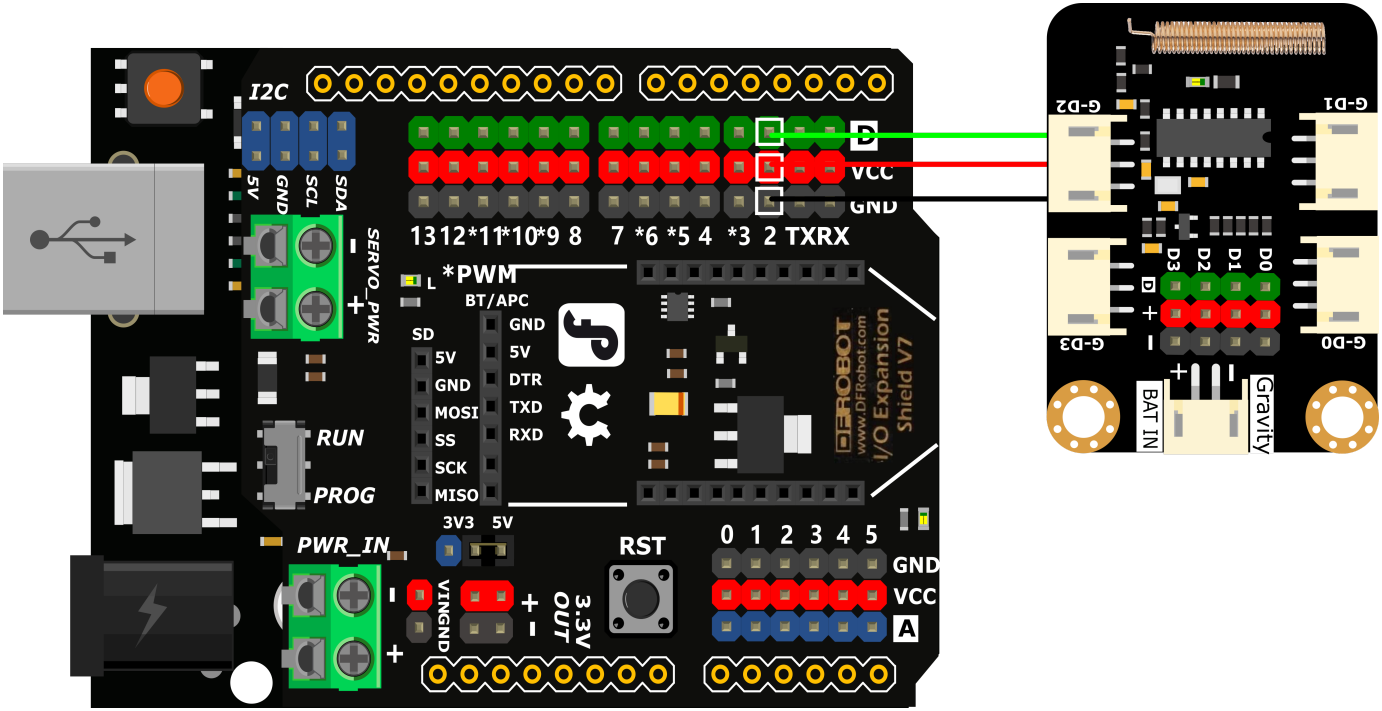
**Requirements**

- **Hardware**
  - DFRuino UNO R3 (https://www.dfrobot.com/product-838.html) x2
  - Wireless switch transmitter x1
  - Wireless switch receiver x1
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)

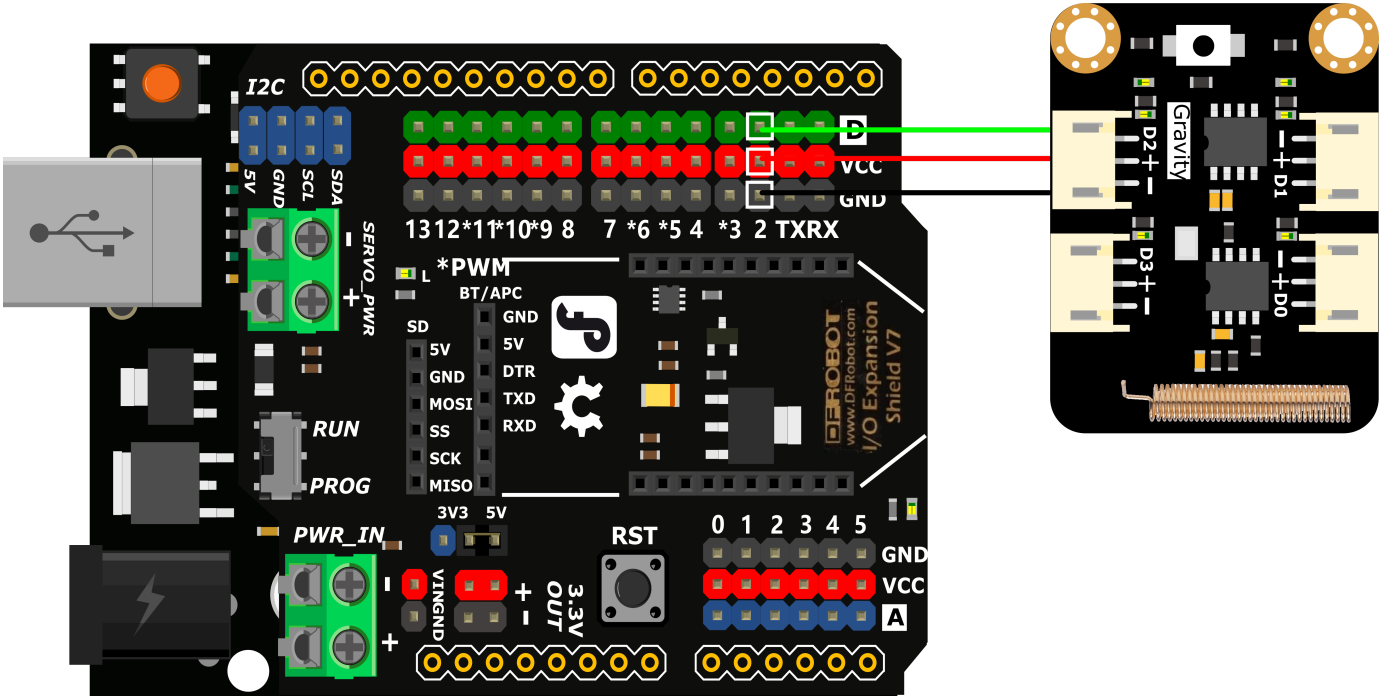
**Connection**

**The transmitting end:**

The transmitting end:




Receiving end:


I2C
5V
GND
SCL
SDA
SERVO_PWR
D
VCC
GND
13 12 *11 *10 *9 8 7 *6 *5 4 *3 2 TXRX
L
*PWM
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
RUN
PROG
PWR_IN
3V3
5V
RST
VINGND
3.3V OUT
0 1 2 3 4 5
GND
VCC
A
DFROBOT
www.DFRobot.com
I/O Expansion Shield V7
Gravity
D2
D1
D3
D0

Sample code

Sample code

### The transmitting end:

```
#define Button_D2 2//Arduino
//#define Button_D2 D2 //ESP32

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(Button_D2, HIGH);
  delay(1000);
  digitalWrite(Button_D2, LOW);
  delay(1000);
}
}
```

### Receiving end:

```
#define Button_D2 2//Arduino
//#define Button_D2 D2 //ESP32

void setup() {
    Serial.begin(115200);
    pinMode(Button_D2, INPUT);
}
void loop() {
    if (digitalRead(Button_D2)) {
        Serial.println("Pressed: D2");
        delay(1000);
    }
}
```

**Result:**

The transmitter G-D2 sends out signals every 1 second, and the receiver D2 receives signals every 1 second.

1 second.

## Advanced Example 1 - Trigger in multiple states

- **Control the transmitter to emit a variety of signals**
- Take Arduino as an example, the other main controllers use the same principle, only need to connect to the corresponding digital port.
- Take inching mode in the receiver as an example, the principles of other modes are the same

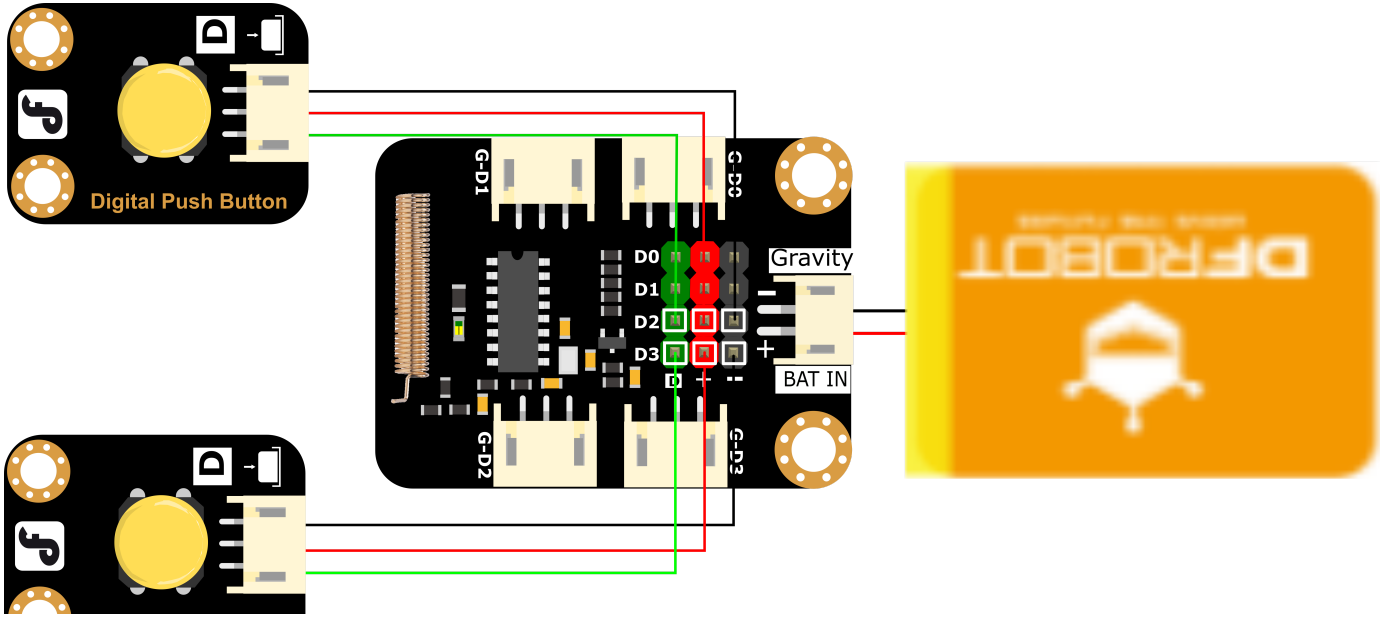
**Requirements**

- **Hardware**
    - DFRuino UNO R3 (https://www.dfrobot.com/product-838.html) x1
    - Wireless switch transmitter x1
    - Wireless switch receiver x1
    - 3.3-5V power supply (https://www.dfrobot.com/product-437.html) x 1
    - Button (https://www.dfrobot.com/product-73.html) x 2

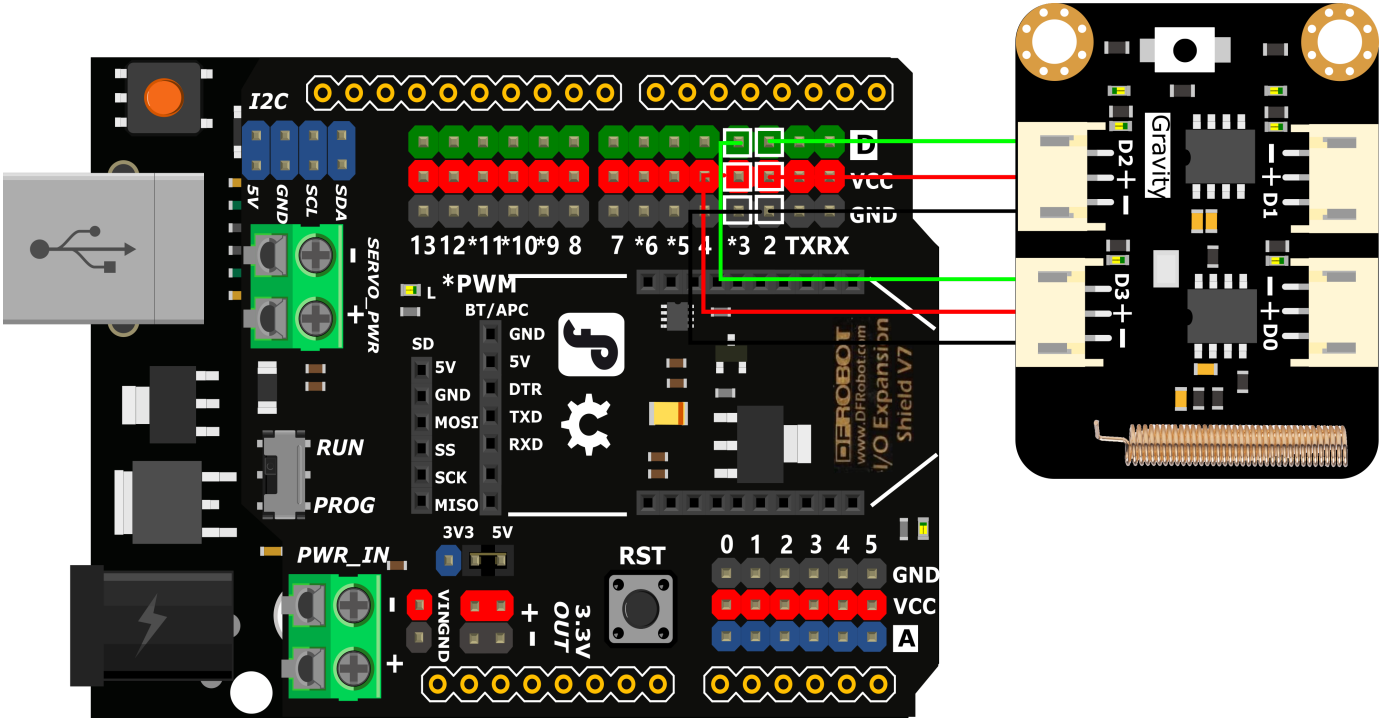
- **Software**
    - Arduino IDE (https://www.arduino.cc/en/Main/Software)

## Connection

### Transmitting end :



Receiving end:



### Sample code

### Receiving end:

```
#define Button_D2 2//Arduino
#define Button_D3 3//Arduino
//#define Button_D2 D2 //ESP32
//#define Button_D3 D3 //ESP32

void setup() {
  Serial.begin(115200);
  pinMode(Button_D2, INPUT);
  pinMode(Button_D3, INPUT);
}
void loop() {
  if (((digitalRead(Button_D2)==0) && (digitalRead(Button_D3)==0))) {
    Serial.println("State:1,Pressed:NONE");
    delay(500);
  }
  if (((digitalRead(Button_D2)==0) && (digitalRead(Button_D3)==1))) {
    Serial.println("State:2,Pressed:D3");
    delay(500);
  }
```

```
  }
  if (((digitalRead(Button_D2)==1) && (digitalRead(Button_D3)==0))) {
    Serial.println("State:3,Pressed:D2");
    delay(500);
  }

  if (((digitalRead(Button_D2)==1) && (digitalRead(Button_D3)==1))) {
    Serial.println("State:4,Pressed:D2&D3");
    delay(500);
  }
  delay(100);
}
```

**Result:**

Press different key combinations, the serial port will print different status values.

## Advanced Example 2 - Transmitter External Power Supply and Jumper Wires

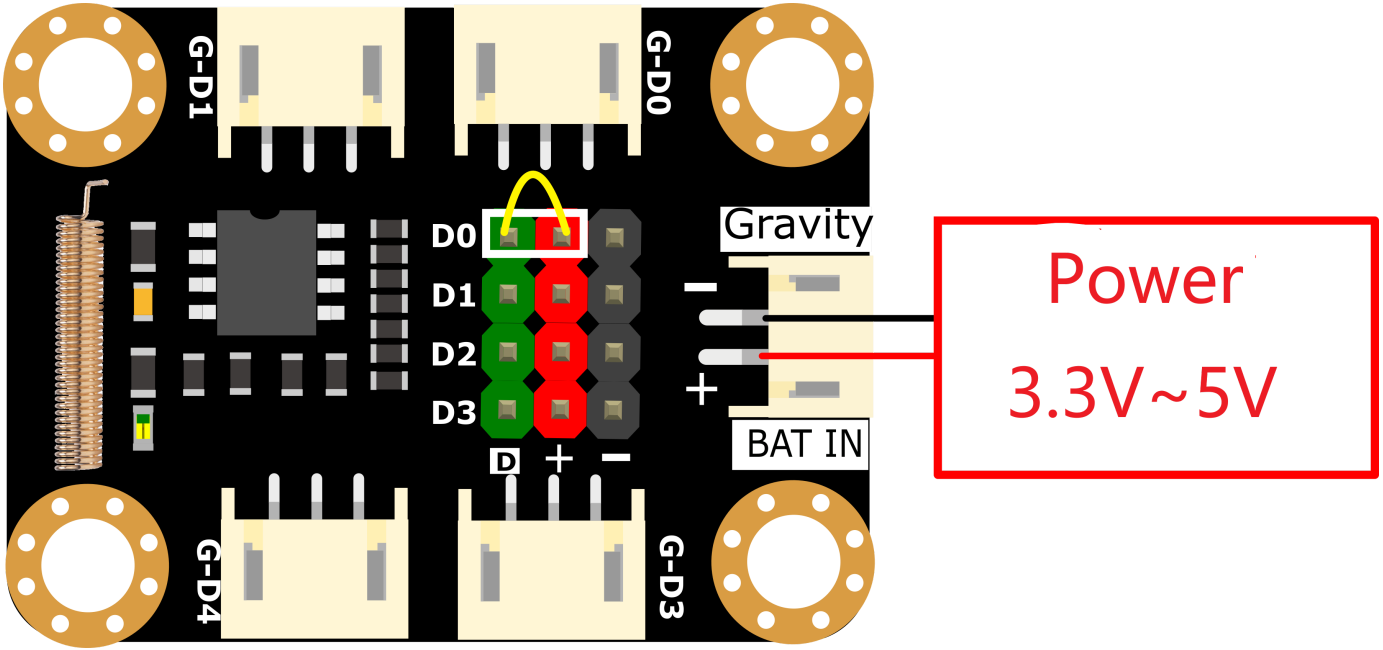
- **This sample is mainly used for demonstration: the transmitter is powered by the battery and the radio frequency signal is sent out by the jumper short circuit**

- Take Arduino as an example, the other main controllers use the same principle, only need to connect to the corresponding digital port.
- Take inching mode in the receiver as an example, the principles of other modes are the same

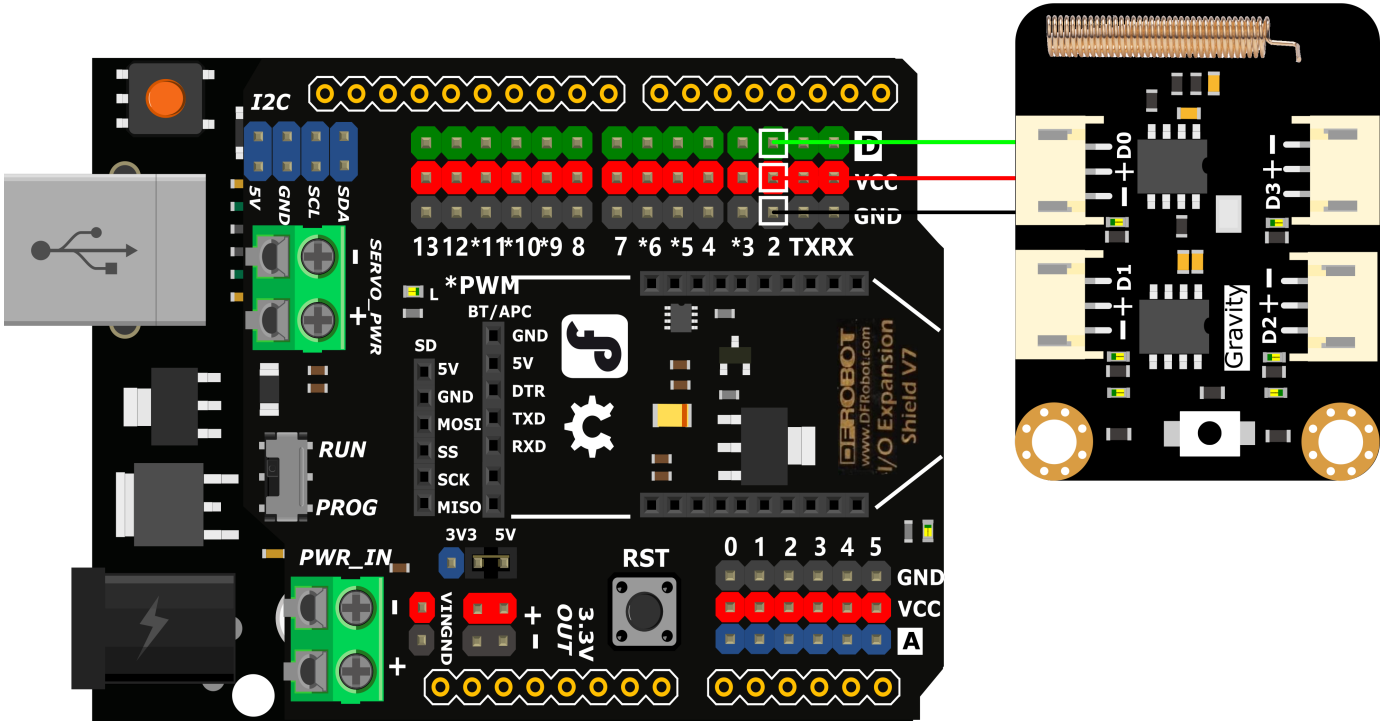
**Requirements**

- **Hardware**
  - DFRuino UNO R3 (https://www.dfrobot.com/product-838.html) x1
  - Wireless switch transmitter x1
  - Wireless switch receiver x1
  - 3.3-5V power supply (https://www.dfrobot.com/product-437.html) x 1
  - Lead / jumper cap x1
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)

**The transmitting end:**




Receiving end:


I2C
5V
GND
SCL
SDA
SERVO_PWR
RUN
PROG
PWR_IN
VINGND
3.3V OUT
3V3
5V
RST
*PWM
BT/APC
SD
5V
GND
MOSI
SS
SCK
MISO
GND
5V
DTR
TXD
RXD
13 12 *11 *10 9 8 7 *6 *5 4 *3 2 TXRX
0 1 2 3 4 5
GND
VCC
A
D
VCC
GND
DFROBOT
www.DFRobot.com
I/O Expansion Shield V7
D0
D1
D2
D3
Gravity

## Sample code

```
#define Button_D2 2//Arduino
//#define Button_D2 D2 //ESP32

void setup() {
    Serial.begin(115200);
    pinMode(Button_D2, INPUT);
}
void loop() {
    if (digitalRead(Button_D2)) {
        Serial.println("Pressed: D2");
        delay(1000);
    }
}
```

## Result:

**1. Connect the power first, then use the lead/jumper cap to short-circuit the red and green pins of the transmitting D0 part, the transmitter will emit a signal, and the corresponding pin**

**pins of the transmitting D0 port, the transmitter will emit a signal, and the corresponding pin of the receiver will receive the signal at the same time**

**2. Use the lead wire/jumper cap to short-circuit the red and green pins of the transmitting end D0 port, and supply power to the transmitter through a controllable power supply（Power supply time>60ms）. The transmitter port connected to the wire will send a signal at the moment of powering on, and the corresponding pin of the receiver will receive the signal at the same time. If the power supply continues to supply power, the transmitter will always be in the transmitting state.**

## Transmitter operating instructions

**Transmitter Power Supply Description:**

- The transmitter cannot be used for charging battery.
- The D0 to D3 pin headers have only power output function and cannot be used as power input terminals, otherwise the module will be damaged
- When using G-D0 to G-D3 interface to connect to the controller, you can leave the external power supply unconnected. If the external power supply is connected at this time, the

power supply unconnected. If the external power supply is connected at this time, the module will automatically select a high-voltage power supply.

- When using G-D0 to G-D3 Gravity interface to connect to the controller for power supply, D0~D3 have voltage output

## FAQ

| |
|---|
| **Q1: The receiver is connected to the controller, the transmitter does not transmit signal but the receiver receives the signal** |
| A: The receiver receives the control of the code, burn the corresponding sample code or burn an empty code to solve the problem |
| **Q2: When using a battery to connect the transmitter and sensor, how to reduce the power consumption as much as possible** |
| A: For some passive sensors, such as button (https://www.dfrobot.com/product-73.html), Collision sensor (https://www.dfrobot.com/product-763.html), magnetic sensor (https://www.dfrobot.com/product-233.html), connect the sensor The red and green wires to the transmitter, disconnect the black wire, which can reduce power consumption |
| **Q3: Can it be used with other transmitters and receivers?** |

| **Q1: The receiver is connected to the controller, the transmitter does not transmit signal but the receiver receives the signal** |
| --- |
| A: Yes, but make sure that the encoding format and rate are the same,different devices D0-D3 may not correspond exactly. The encoding format of the transmitter is EV1527, and the transmission rate is less than 10K; the decoding format of the receiver is EV1527, and the decoding rate is 0~10K. |
| **Q4: Can the receiver directly control the relay?** |
| A: Yes, the positive and negative terminals of any one interface of the receiver is connected to the 3.3/5V power supply, other ports can be directly connected to the relay. |

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum** (https://www.dfrobot.com/forum/).

## More Documents

## More Documents

TEL0140 Schematic
(https://dfimg.dfrobot.com/nobody/wiki/343d58740d9aafcaf2ad42e83b3bbaa0.pdf)

TEL0140 Dimensions
(https://dfimg.dfrobot.com/nobody/wiki/1af09957f25f1cd322fa140f740fae72.pdf)

TEL0142 Schematic
(https://dfimg.dfrobot.com/nobody/wiki/52d903c257c0cb76aa3cda480b619b2a.pdf)

TEL0142 Dimensions
(https://dfimg.dfrobot.com/nobody/wiki/adb9fdec321dac91df2862ce83fda971.pdf)