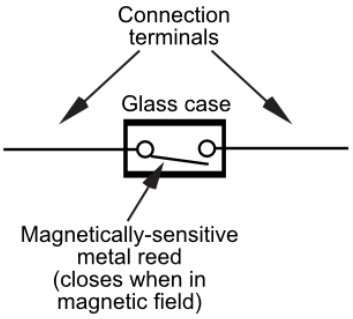
Reed Switch Arduino Demo

Please Note: This product is not sold by Parallax. This demo was created to support the 2013 National microMedic Contest kits, which are no longer available.

Two reed switches and two magnets are included in the 2013 National microMedic Contest Kit. This Board of Education Shield for Arduino demo provides example sketches for simple use of one switch and magnet, or a more advanced use of two switches and magnets together.

The glass reed switch is mechanical switch that is activated when a strong magnetic field is brought near it. The magnetic reed is enclosed in a glass ampoule, and has two connection leads for wiring to a circuit.



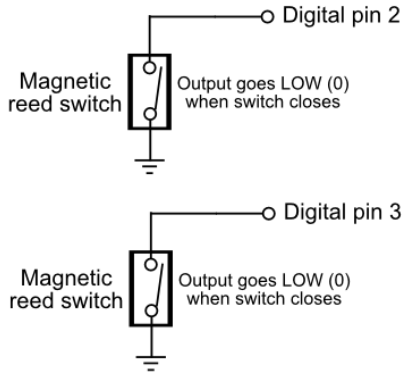
For best results, use a magnet with a strong magnetic field .This increases the sensitivity of the reed switch. The higher the field strength, the further away the magnet can be and still trigger the switch. The Parallax 3/8 Diameter x 1/8 Thick Magnet (#605-00006, discontinued) is an ideal companion for the reed switch. It's both compact and has a high field strength.

The glass enclosure of the reed switch can break if the device is dropped or subjected to a sharp impact.

Connections

The simple example sketch uses a single reed switch connected to Digital pin 2. The Extended example uses an additional reed switch connected to Digital pin 3. To connect the reed switch to

the Arduino Shield, attach one lead to the digital pin indicated, and the other to Gnd, as shown in the figure.



Programming Examples

Simple Programming Example

To use this example, upload the BasicReed sketch to your Arduino, then open the Serial Monitor window. Ensure that the Baud Rate is set at 9600. Pass a magnet within 1" of the reed switch, and watch the value in the Serial Monitor window.

- When the switch is closed (magnet is near), the value is 0 (LOW)
- When the switch is open (magnet is removed), the value is 1 (HIGH)

The “reverse logic” of the sketch has to do with the use of the Arduino’s internal pull-up resistors that are on each input pin. When activated -- as shown in the sketch -- the pin is normally in a HIGH (digital 1) state. Because the other end of the switch is connected directly to ground, when the switch is activated, the pin goes LOW (digital 0).

```
const int switchPin = 2;     //     // Reed switch to digital pin 2

void setup() {
  pinMode(switchPin, INPUT);        // switchPin is an input
  digitalWrite(switchPin, HIGH);    // Activate internal pullup resistor
  Serial.begin(9600);
}

void loop() {
  Serial.println(digitalRead(switchPin));  // Display current value
  delay(200);
}
```

Extended Programming Example

This example demonstrates a more advanced application of using two reed switches, and comparing the number of times both switches are activated. It might be used, for example, to show a complete movement cycle, such as an arm exercise, where it's important that the arm move from one position to another. Place a strong magnet in a fabric cuff, and position a reed switch at each extreme of the movement.

The ExtendedReed sketch includes coding to prevent switch "bounce," which occurs when the reed opens or closes many times in a very short period. This is actually quite common in all mechanical switches. In many instances, the multiple bounces create little or no problem. But in this extended programming example, the bounces will cause miscounts. Code is added to ignore additional switch closures if they occur too soon after the first one.

To use this example, upload the ExtendedReed sketch to your Arduino, then open the Serial Monitor window. Ensure that the Baud Rate is set at 9600. Pass a magnet within 1" of each switch, and watch the values in the Serial Monitor window. The count is updated each time switch 0 is activated.

```
// Reed switch repetition counter

int sense0 = 2;
int sense1 = 3;
int counter0 = 0;
int counter1 = 0;
long lastDebounce0 = 0;
long lastDebounce1 = 0;
long debounceDelay = 500;    // Ignore bounces under 1/2 second

void setup() {
  Serial.begin(9600);
  pinMode(sense0, INPUT);
  digitalWrite(sense0, HIGH);
  pinMode(sense1, INPUT);
  digitalWrite(sense1, HIGH);
  attachInterrupt(0, trigger0, FALLING);
  attachInterrupt(1, trigger1, FALLING);
  Serial.println("Repetition counter");
  Serial.print("Start");
  Serial.print("\t");
  Serial.println("End");
}

void loop() {
  // Nothing here
}

void trigger0() {
  if( (millis() - lastDebounce0) > debounceDelay){
    counter0++;
    Serial.print(counter0);
    Serial.print(" : ");
    Serial.println(counter1);
```

```
    lastDebounce0 = millis();
  }
}

void trigger1() {
  if( (millis() - lastDebounce1) > debounceDelay){
    counter1++;
    lastDebounce1 = millis();
  }
}
```