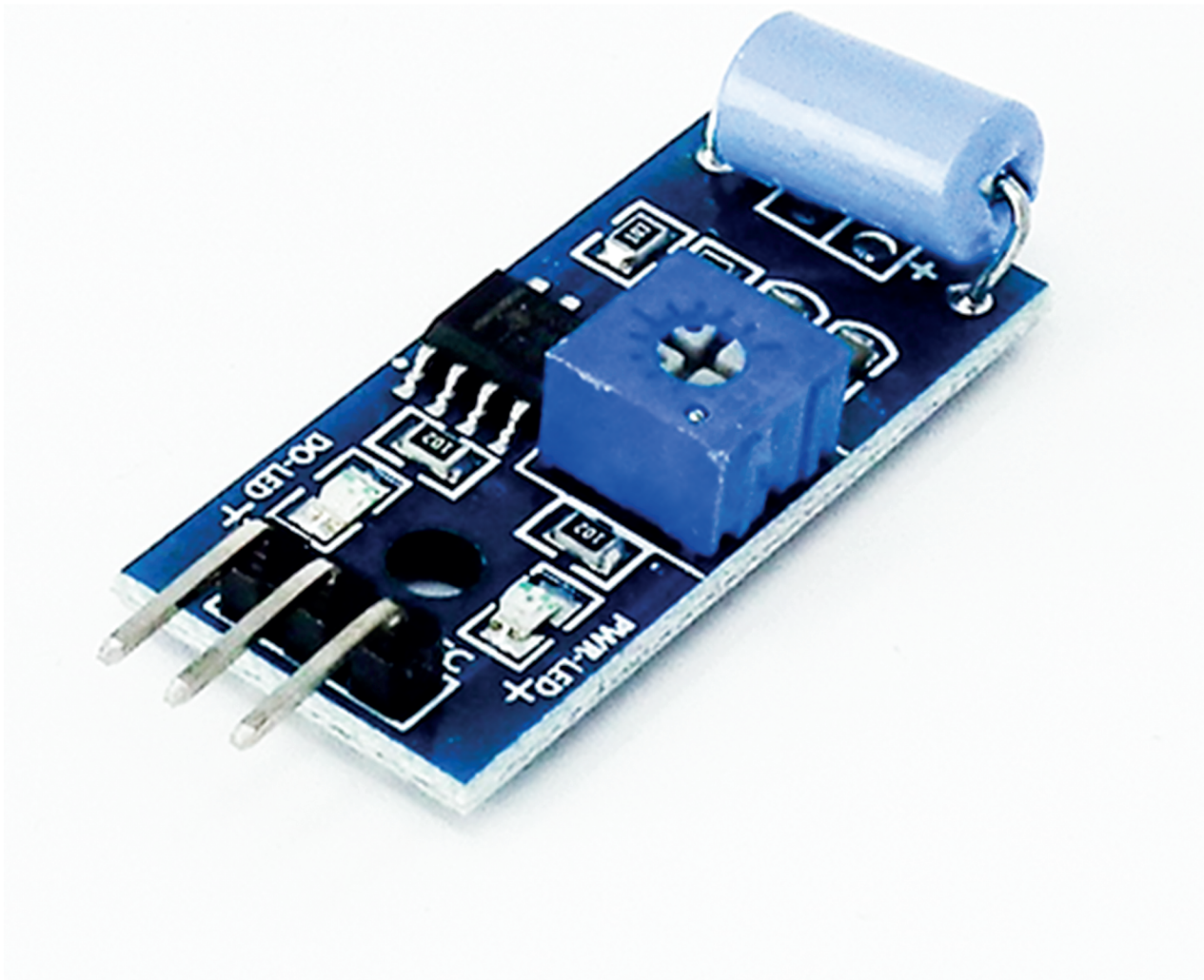


SW420 Vibration Schüttel Erschütterung Sensor Modul Datenblatt



Contents:

- 1. Description**
- 2. Features**
- 3. Applications**
- 4. Connecting**
- 5. Example Code**

1. Description

This sensor module produce logic states depends on vibration and external force applied on it. When there is no vibration this module gives logic LOW output. When it feels vibration then output of this module goes to logic HIGH. The working bias of this circuit is between 3.3V to 5V DC.

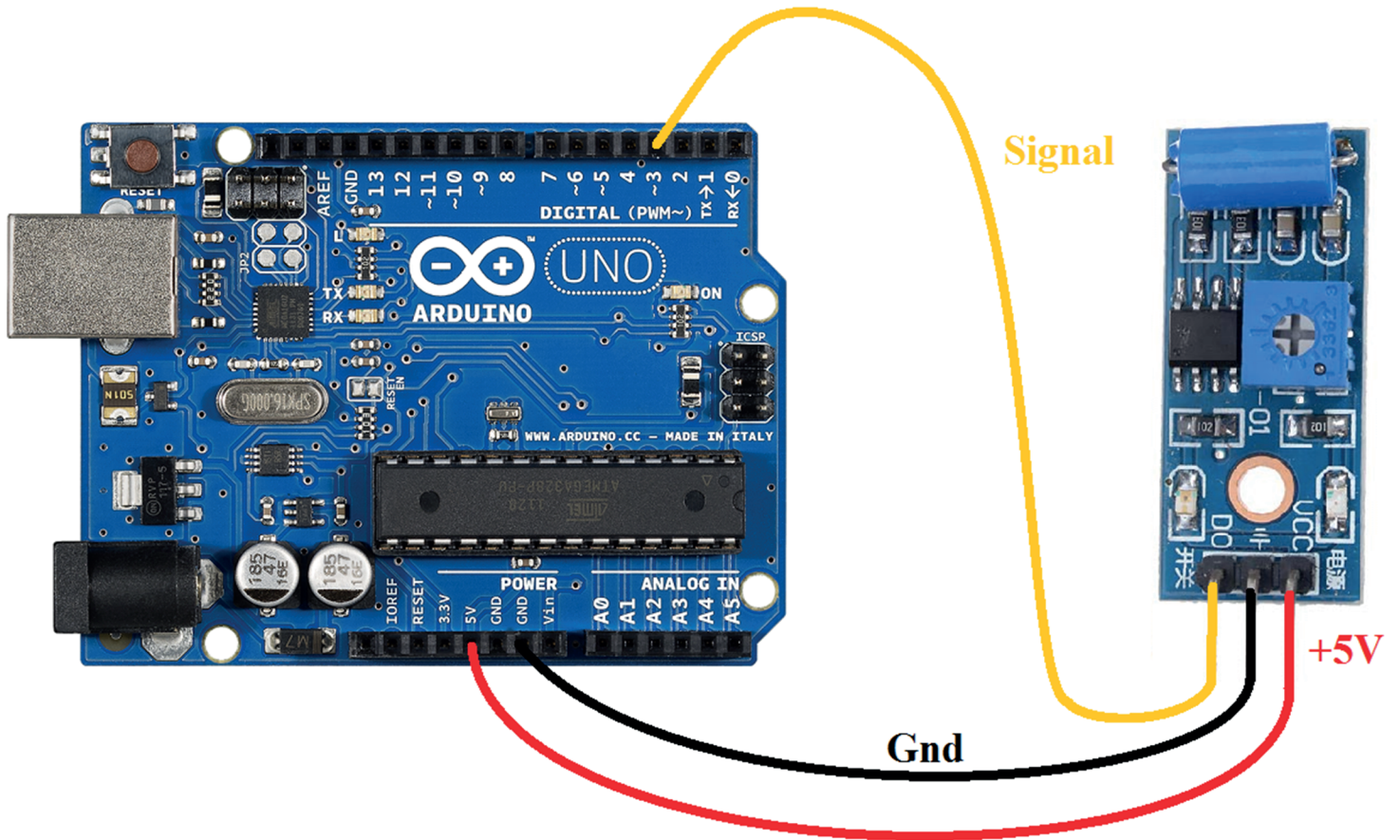
2. Features

- The default state of the switch is close
- Digital output Supply voltage:3.3V-5V
- On-board indicator LED to show the results
- On-board LM393 chip
- SW-420 based sensor, normally closed type vibration sensor
- Dimension of the board: 3.2cm x 1.4cm

3. Applications

- Vibration detecting
- Burglary protection system
- Object Movement detecting
- Triggering effect reported theft alarm
- Smart car
- Earthquake alarm
- Motorcycle alarm

4. Connecting



5. Example Code

Arduino Code for Logic State Output from sensor module, here onboard LED of Arduino indicates the presence of vibration.

```
int vibr_pin=3;
int LED_Pin=13;
void setup() {
  pinMode(vibr_pin, INPUT);
  pinMode(LED_Pin, OUTPUT);
}

void loop() {
  int val;
  val=digitalRead(vibr_pin);
  if(val==1)
  {
    digitalWrite(LED_Pin, HIGH);
    delay(1000);
    digitalWrite(LED_Pin, LOW);
    delay(1000);
  }
  else
  digitalWrite(LED_Pin, LOW);
}
```

Arduino Code for Value Reading and serial printing Vibration value, this code turns ON the onboard LED when measurement goes greater than 1000, you can adjust this threshold to your need.

```
int LED_Pin = 13;
int vibr_Pin =3;

void setup(){
  pinMode(LED_Pin, OUTPUT);
  pinMode(vibr_Pin, INPUT); //set vibr_Pin input for measurment
  Serial.begin(9600); //init serial 9600
  // Serial.println("-----Vibration demo-----");
}
void loop(){
  long measurement =TP_init();
  delay(50);
  // Serial.print("measurment = ");
  Serial.println(measurement);
  if (measurement > 1000){
    digitalWrite(LED_Pin, HIGH);
  }
  else{
    digitalWrite(LED_Pin, LOW);
  }
}

long TP_init(){
  delay(10);
  long measurement=pulseIn (vibr_Pin, HIGH); //wait for the pin to get HIGH and returns measurement
  return measurement;
}
```