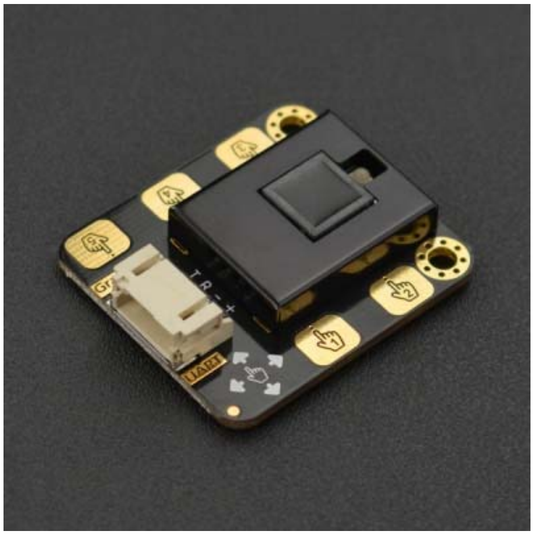


Gesture & Touch Sensor

SKU:SEN0285

Introduction

This sensor module integrates gesture recognition and touch detecting functions in one piece, and provides an adjustable detection range within 0~30cm. When connected to your microcontroller, it can detect 5-way touch signal and 7 kinds of gestures: move left, move right, move forward, move backward, pull up, pull down, pull and remove. Besides, the sensor is also equipped with the function of auto-sleep and wake up. The module comes with the gesture recognition algorithm and provides simple and reliable data output. Use the sensor to directly communicate with upper computer or micro-controllers like Arduino and Raspberry Pi via serial port.

The onboard 5-way touch pad on the sensor can be directly used to detect touch, or you can extend the touch pad with wires to make it perfectly fit in your application. The outer shield for the sensor retains the advantages of Gravity series as well as makes the sensor more durable. This sensor can be used to make smart lamp, DIY intelligent car, or used in interactive projects requiring gesture recognition.

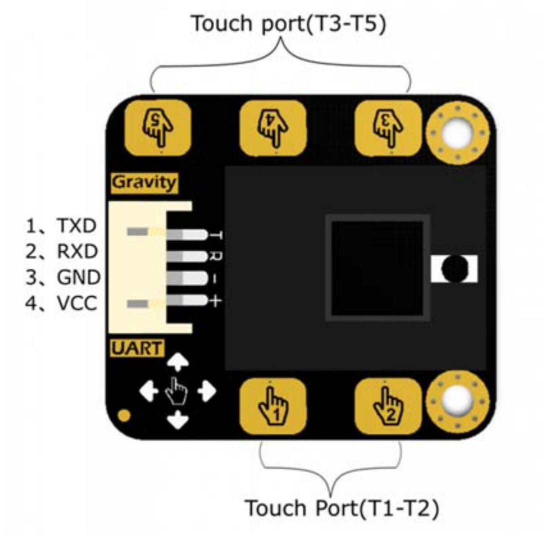
NOTE

- Abnormality might occur when using the module in strong sunlight condition due to its working principle;
- When powered down, the sensor will be initialized automatically and the previous setting will be invalid.
- When sending byte constinuously, there should be at least 200us time interval between the end of the last byte and the starting of the next byte; The sensor can receive/transmit data only in working state, while for processing data, we must make the sensor stay in non-working state. That is to say, if there is an object over the sensor when receiving/transmitting data, we need to move it away for completing the following settings.

Specification

- Power Supply: 3.3V - 6V (recommend 5V）
- Output Voltage: 0 - 3.3V
- Operating Current: about 56.3mA
- Sleep Mode Current: about 40uA
- Output: TTL serial port
- Adapter Plate Size: 35mm×22mm/1.38×0.87”
- Interface: Gravity
- Serial Protocol Format: 9600 band rate; 8 data bits, no parity bit, 1 stop bit

Pinout



Num	Label	Description	Color
1	TXD	Receive Data	Greem
2	RXD	Transmit Data	Blue
3	GND	Ground	Black
4	VCC	Power Supply	Red
5	Touch Port	5-way Touch signal	

Function Description

- Set the sensing height(default 20cm when powered on)

Example: **DFGT.setGestureDistance(20)**---Unit(cm); Max height: 30cm

- Set the sleep enablement or enter auto-sleep mode (sleep mode disabled when powered on)

Example: DFGT.setSleep(4);---unit(s)

- Set gesture sensing enablement (all the seven gestures can be recognized when powered on)

Example(1) DFGT.enableFunction(DFGT_FUN_ALL);---enable all gestures sensing fuction

Example(2) DFGT.disableFunction(DFGT_FUN_RIGHT | DFGT_FUN_LEFT);---disable sensing function of part gestures

Example(3) DFGT.enableFunction(DFGT_FUN_RIGHT | DFGT_FUN_LEFT);---enable sensing function of part gestures

Tutorial

In this tutorial, we will show you the basic usages of the sensor. First of all, connect all parts together and download the codes to Arduino UNO. Open the serial monitor, when we do any one of the seven gestures or touch any touch pad on the sensor, there should be a corresponding reaction appearing on the serial monitor.

Note:

- Default function when powered on: no sleep, 5-way touch, detecting height: 15cm;
- The default detecting height is 15cm, more higher heigth detection needs to be set by yourself;
- Keep a normal speed when gesticulating to let the sensor better understand your gesture.

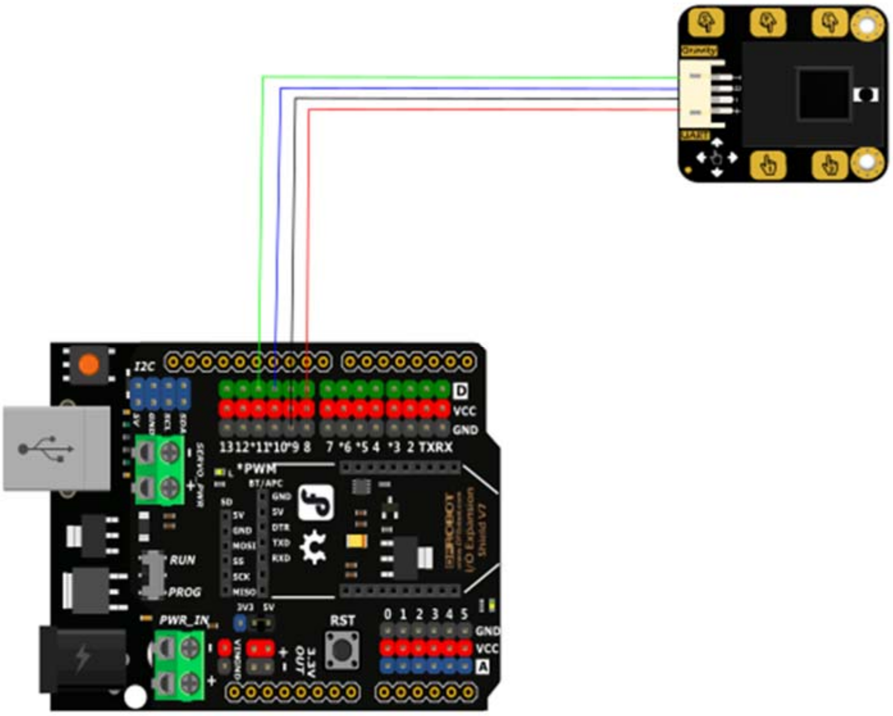
Requirements

- **Hardware**
  - DFRduino UNO R3 x 1
  - IO Sensor Expansion Shield V7.1 x1
  - Gesture&Touch Sensor x1
  - PC x1
  - Connectors
- **Software**
  - Arduino IDE (V1.6.8 above)
  - Download and install the **XXX Library** (About how to install the library?)

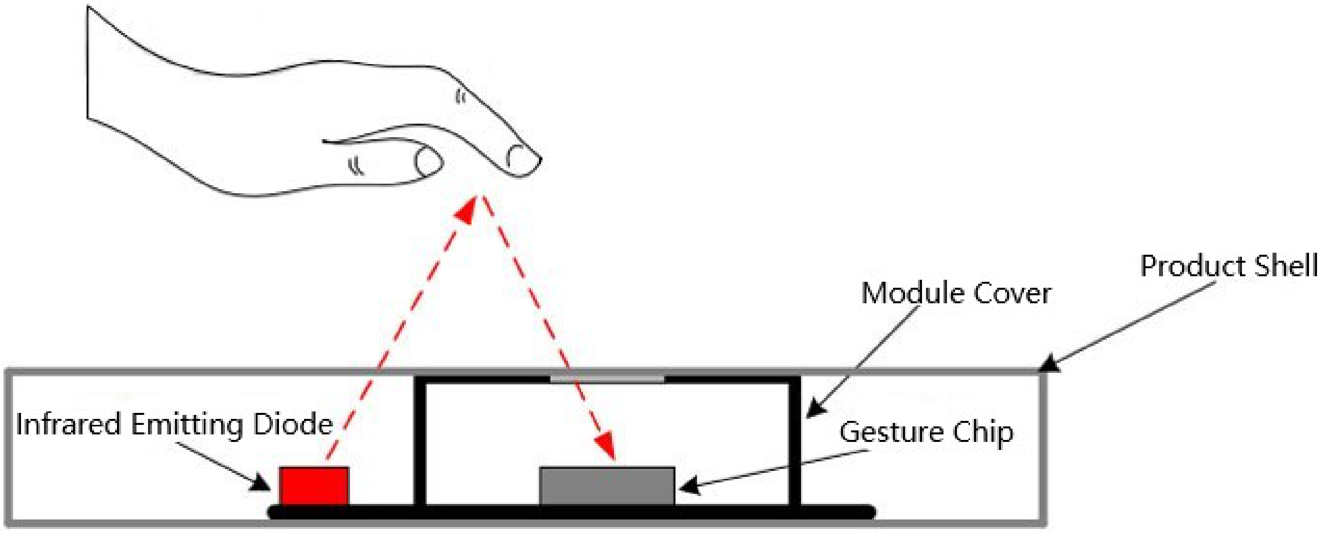
Arduino Debug

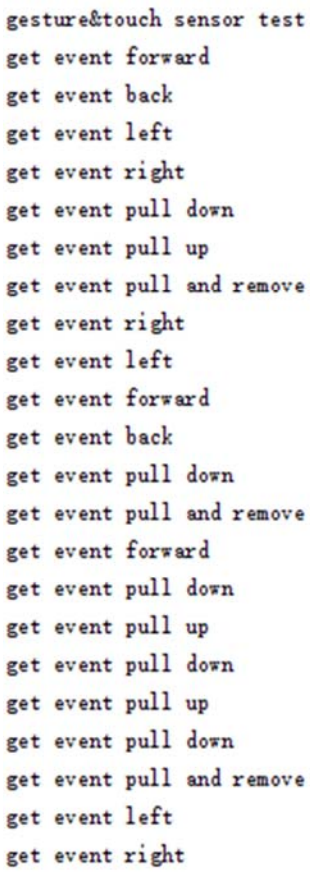
Since the gesture & touch sensor is a serial device and there is only 1 hardware serial port on Arduino, we recommend using the sensor with software serial or you can use device with multiple serial ports such as Arduino Leonardo, Arduino Mega2560 and so on. Here we choose Arduino UNO as the controller and define D10 and D11 as software serial port.

Connection Diagram



How to use gestures recognization (The maxmium height the sensor can detect is 30cm)





```
gesture&touch sensor test
get event forward
get event back
get event left
get event right
get event pull down
get event pull up
get event pull and remove
get event right
get event left
get event forward
get event back
get event pull down
get event pull and remove
get event forward
get event pull down
get event pull up
get event pull down
get event pull up
get event pull down
get event pull and remove
get event left
get event right
```

Arduino Debugging Codes

PC serial tool will be needed here, and the read data will be displayed on the Arduino serial interface.

```
/*
 * file simpleGesture.ino
 *
 * @ https://github.com/DFRobot/DFRobot_Gesture_Touch
 *
 * example for DFRobot Gesture&Touch sensor
 *
 * sensor event will print on your serial monitor
 *
 * for esp32, rx_pin = D5, tx_pin = D6
 *
 * Copyright   [DFRobot](http://www.dfrobot.com), 2016
 * Copyright   GNU Lesser General Public License
 *
 * version  V1.0
 * date  2018-10-19
 */
```

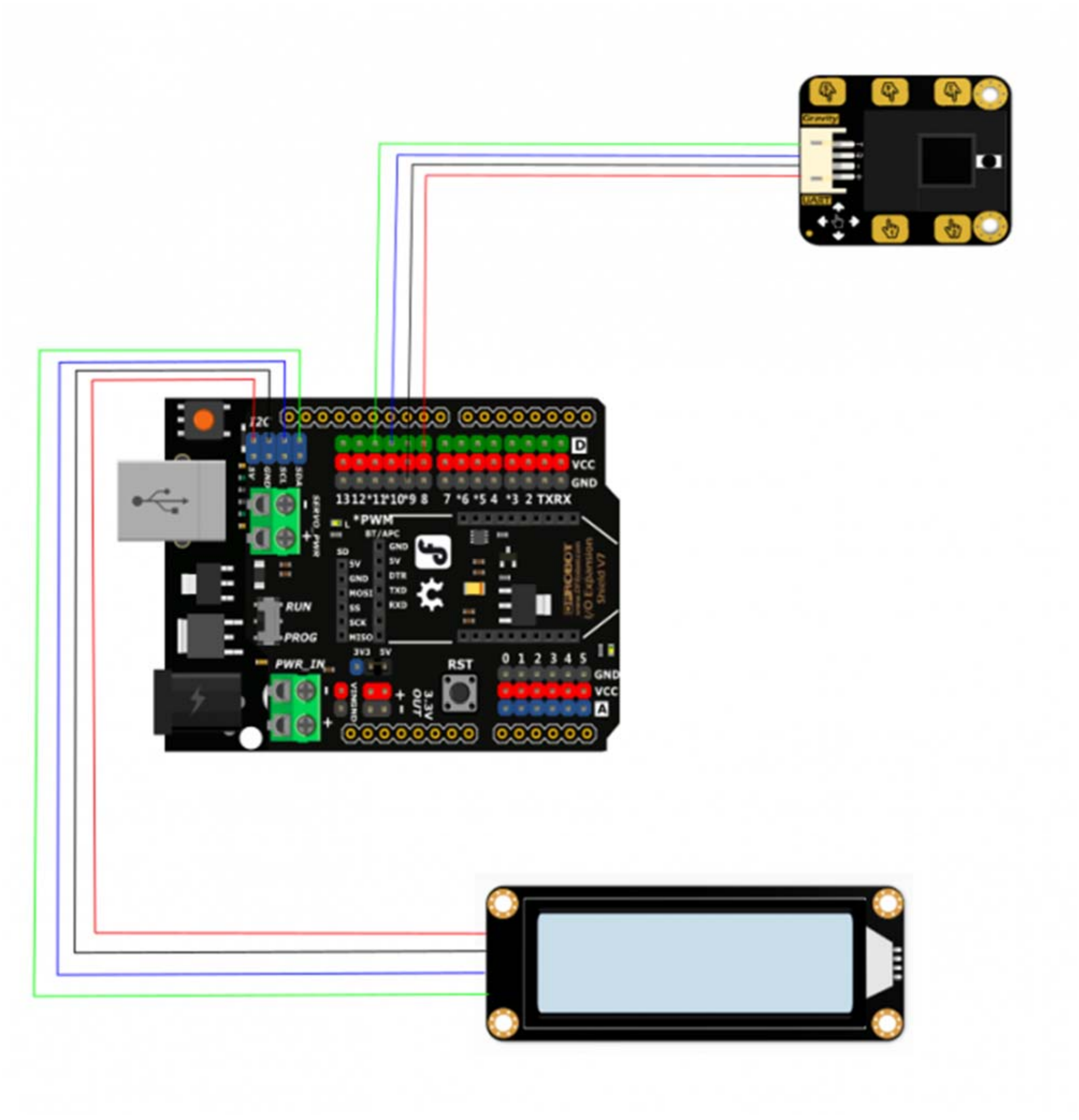
```
#include "DFRobot_Gesture_Touch.h"

#ifdef __AVR__
  #include "SoftwareSerial.h"
  SoftwareSerial    mySerial(10, 11);         // example for uno, use software serial
#elif defined ESP_PLATFORM
  #include "HardwareSerial.h"
  HardwareSerial    mySerial(1);
#endif

DFRobot_Gesture_Touch   DFGT(&mySerial);    // init sensor object, request write and read function

void setup()
{
  Serial.begin(115200);
  mySerial.begin(9600);
  delay(2000);
  while(mySerial.available())  // flush serial buffer
    mySerial.read();
  Serial.println("gesture&touch sensor test");

  DFGT.setGestureDistance(20);             // suggest default value
  // DFGT.enableFunction(DFGT_FUN_ALL);       // enable all functions
  // DFGT.disableFunction(DFGT_FUN_RIGHT | DFGT_FUN_LEFT);    // disable function test
  // DFGT.disableFunction(DFGT_FUN_TOUCH1 | DFGT_FUN_TOUCH2);    // disable function test
  // DFGT.enableFunction(DFGT_FUN_RIGHT | DFGT_FUN_LEFT);     // enable function test
  // DFGT.setSleep(4);                        // set auto sleep time out, in sleep mode, put something
}

void loop()
{
  int8_t    rslt = DFGT.getAnEvent();  // get an event that data saved in serial buffer
  if(rslt != DF_ERR) {
    // DFGT.setSleep(DFGT_SLEEP_DISABLE);  // disable auto sleep
    switch(rslt) {
      case DFGT_EVT_BACK: {
        Serial.println("get event back");
      } break;
      case DFGT_EVT_FORWARD: {
        Serial.println("get event forward");
      } break;
      case DFGT_EVT_RIGHT: {
        Serial.println("get event right");
      } break;
      case DFGT_EVT_LEFT: {
        Serial.println("get event left");
      } break;
      case DFGT_EVT_PULLUP: {
        Serial.println("get event pull up");
      } break;
      case DFGT_EVT_PULLDOWN: {
        Serial.println("get event pull down");
      } break;
      case DFGT_EVT_PULLREMOVE: {
        Serial.println("get event pull and remove");
      } break;
      case DFGT_EVT_TOUCH1: {
        Serial.println("get event touch1");
      } break;
      case DFGT_EVT_TOUCH2: {
        Serial.println("get event touch2");
      } break;
      case DFGT_EVT_TOUCH3: {
        Serial.println("get event touch3");
      } break;
      case DFGT_EVT_TOUCH4: {
        Serial.println("get event touch4");
      } break;
      case DFGT_EVT_TOUCH5: {
        Serial.println("get event touch5");
      } break;
    }
  }
}
```

Functions: recognize the 5-way touch signal and the seven gestures including move right, move left, forward, backward, pull up, pull down, and pull and leave.

Dispaly the results on LCD

In actual application, we may use the sensor outdoor, so here we specially provide an case for you. Li-ion batteries will be the power supply and an LCD will be used to display the measured distance information.

Arduino LCD Connection



Arduino LCD Debugging Code

```
/*
 * file simpleGesture.ino
 *
 * @ https://github.com/DFRobot/DFRobot_Gesture_Touch
 *
 * example for DFRobot Gesture&Touch sensor
 *
 * sensor event will print on your serial monitor
 *
 * for esp32, rx_pin = D5, tx_pin = D6
 *
 * Copyright   [DFRobot](http://www.dfrobot.com), 2016
 * Copyright   GNU Lesser General Public License
 *
 * version  V1.0
 * date  2018-10-19
 */

#include "DFRobot_Gesture_Touch.h"
#include <Wire.h>
#include "DFRobot_LCD.h"
DFRobot_LCD lcd(16,2);  //16 characters and 2 lines of show

#ifdef __AVR__
  #include "SoftwareSerial.h"
  SoftwareSerial    mySerial(10, 11);         // example for uno, use software serial
#elif defined ESP_PLATFORM
  #include "HardwareSerial.h"
  HardwareSerial    mySerial(1);
#endif

DFRobot_Gesture_Touch   DFGT(&mySerial);    // init sensor object, request write and
read function

void setup()
{
  lcd.init();                              // init LCD IIC
  delay(20);
  lcd.init();                             //init LCD IIC
  delay(20);
  mySerial.begin(9600);
  delay(2000);

  DFGT.setGestureDistance(20);             // suggest default value
}

void loop()
{
  lcd.setCursor(0, 0);                 // position the cursor to LCD line 0, column 0
  lcd.print("State:");               //start to display "State" at the LCD line 0,
column 0
```

```
  lcd.setCursor(0, 1);               // poistion the cursor to the LCD line 2, column 
0
  int8_t    rslt = DFGT.getAnEvent();  // get an event that data saved in serial 
buffer
  if(rslt != DF_ERR) {
    switch(rslt) {
      case DFGT_EVT_BACK: {
        lcd.print("Back            ");
      } break;
      case DFGT_EVT_FORWARD: {
        lcd.print("Forward            ");
      } break;
      case DFGT_EVT_RIGHT: {
        lcd.print("Right            ");
      } break;
      case DFGT_EVT_LEFT: {
        lcd.print("Left            ");
      } break;
      case DFGT_EVT_PULLUP: {
        lcd.print("Pull up          ");
      } break;
      case DFGT_EVT_PULLDOWN: {
        lcd.print("Pull Down          ");
      } break;
      case DFGT_EVT_PULLREMOVE: {
        lcd.print("Pull and Remove          ");
      } break;
      case DFGT_EVT_TOUCH1: {
        lcd.print("Touch1          ");
      } break;
      case DFGT_EVT_TOUCH2: {
        lcd.print("Touch2          ");
      } break;
      case DFGT_EVT_TOUCH3: {
        lcd.print("Touch3          ");
      } break;
      case DFGT_EVT_TOUCH4: {
        lcd.print("Touch4          ");
      } break;
      case DFGT_EVT_TOUCH5: {
        lcd.print("Touch5          ");
      } break;
    }
  }
}
```

Result displayed on LCD:

```
State:

Pull Down
```

FAQ

For any questions, advice or cool ideas to share, please visit the **DFRobot Forum**

More Documents

Get **Gesture & Touch Sensor V1.0** from DFRobot Store or **DFRobot Distributor**.