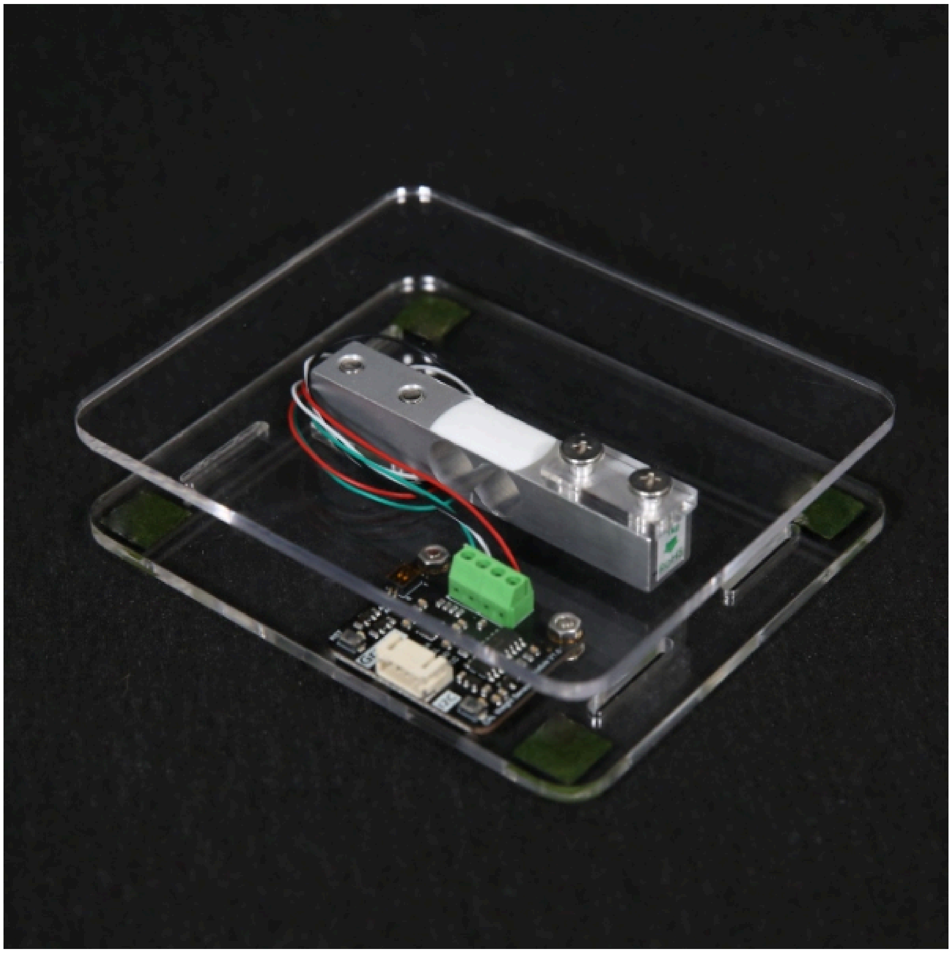
SKU:KIT0176 (https://www.dfrobot.com/product-2289.html)

(https://www.dfrobot.com/product-2289.html)



Introduction

The weight sensor is a practical sensor close to daily life scenes. It can make many interesting applications by acquiring the weight information of objects, such as household electronic scales, daily necessities remaining alarms, and Internet of Things weighing platforms.

This weight sensor kit is equipped with all the accessories needed to build the scale, which reduces the difficulty of structure construction, so that anyone can quickly create an electronic scale to obtain the weight information of the object. In addition, because everyone has different installation methods and usage habits, the sensor needs to be calibrated. The traditional method is to use a separate calibration code to obtain the calibration value. To optimize this problem, the signal adapter board in this kit is built-in The calibration algorithm can calibrate the weight information with one key, and there is no need to run a separate code for calibration.

Note: The maximum range of the sensor is 1kg, please do not exceed the range, otherwise the sensor will be permanently damaged, which will cause inaccurate data.

Features

- Complete accessories, convenient and quick to build the scale
- Built-in calibration algorithm
- Weight calibration can be done by pressing the button
- Weight calibration can be done through API
- Gravity I2C communication mode, compatible with 3.3V and 5V
- Compatible with Arduino series, micro:bit, ESP32, Raspberry Pi

Application

- Simple electronic scale
- Daily necessities weighing
- Daily necessities remaining alarm
- Internet of Things weighing platforms

Specification

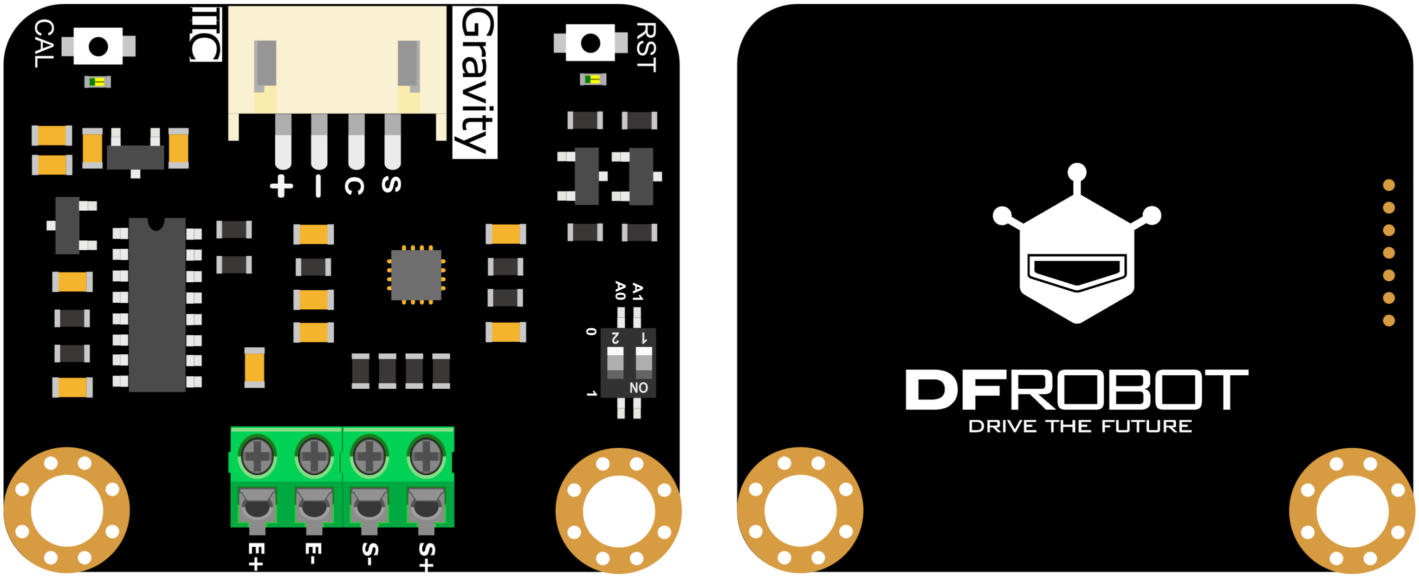
- Measuring Range: 1g-1000g
- Working Voltage: 3.3V/5V
- Working Current: <20mA
- AD Conversion Chip: HX711
- Synthesis Deviation: <±0.2% g
- Overall Dimensions after Assembly: L x W x H: 120 x 100 x 30 mm/ 4.72 x 3.94 x 1.18 inch
- Signal Transfer Board Size: 31 x 37mm / 1.22 x 1.46 inch
- Communication Method: I2C

- I2C Address: 0 x 64 / 0 x 65 / 0 x 66 / 0 x 67

After the address is modified, it will take effect after the sensor is powered off and restarted

A0	A1	Address
0	0	0X64
1	0	0X65
0	1	0X66
1	1	0X67

Board Overview



SN	Name	Description		SN	Name	Description
1	VCC	+		6	E+	Sensor red wire
2	GND	-		7	E-	Sensor black wire
3	SCL	I2C Clock Line		8	S-	Sensor white wire

4	SDA	I2C Data Line		9	S+	Sensor green wire

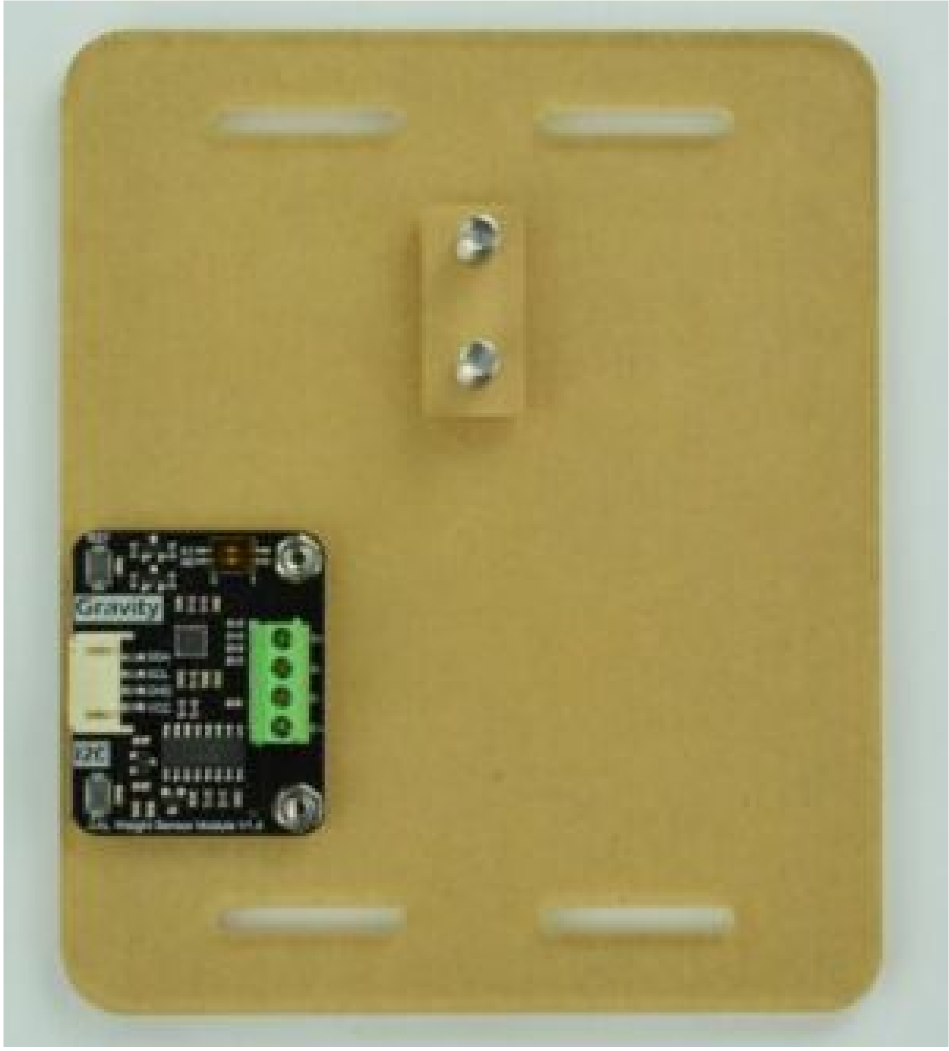
SN	Name	Description		SN	Name	Description
5	CAL Button	Automatic Calibration		10	RST Button	Tare

Operation Instruction

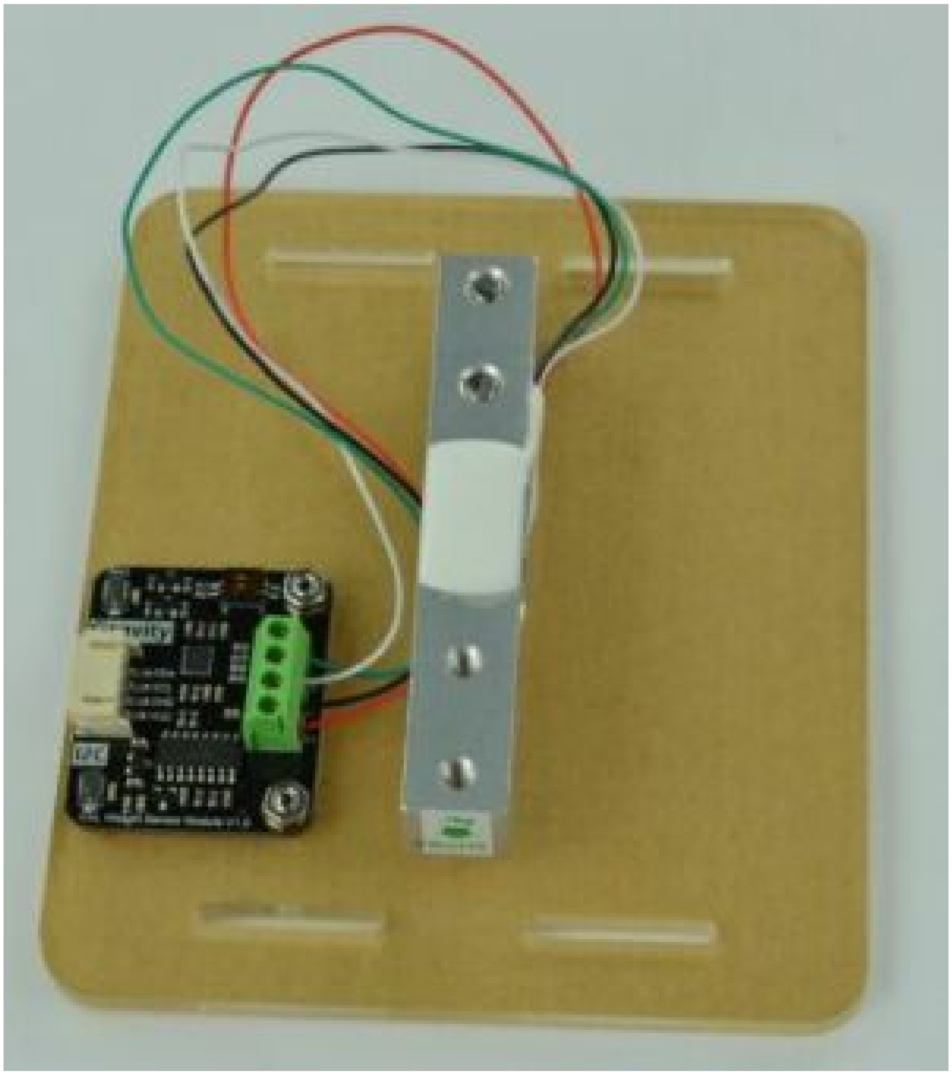
- During calibration and measurement, the most accurate data will be obtained when the object is placed in the middle area of the scale.
- After pressing the rst key, no matter how many objects are placed on the sensor, the sensor will clear the current weight value, and the weight of the objects placed later will accumulate from 0, thus achieving the "tare" function.
- Every time you reset/restart the Arduino or open the serial port monitor, the system will "tare" once by default.
- The sensor calibration method: put an object of known weight on the sensor, and then "tell" the sensor what the weight of the object is, then the sensor will be calibrated (single-point calibration).
- See FAQ (https://wiki.dfrobot.com/HX711_Weight_Sensor_Kit_SKU_KIT0176#target_11) for other instructions.

Assembly

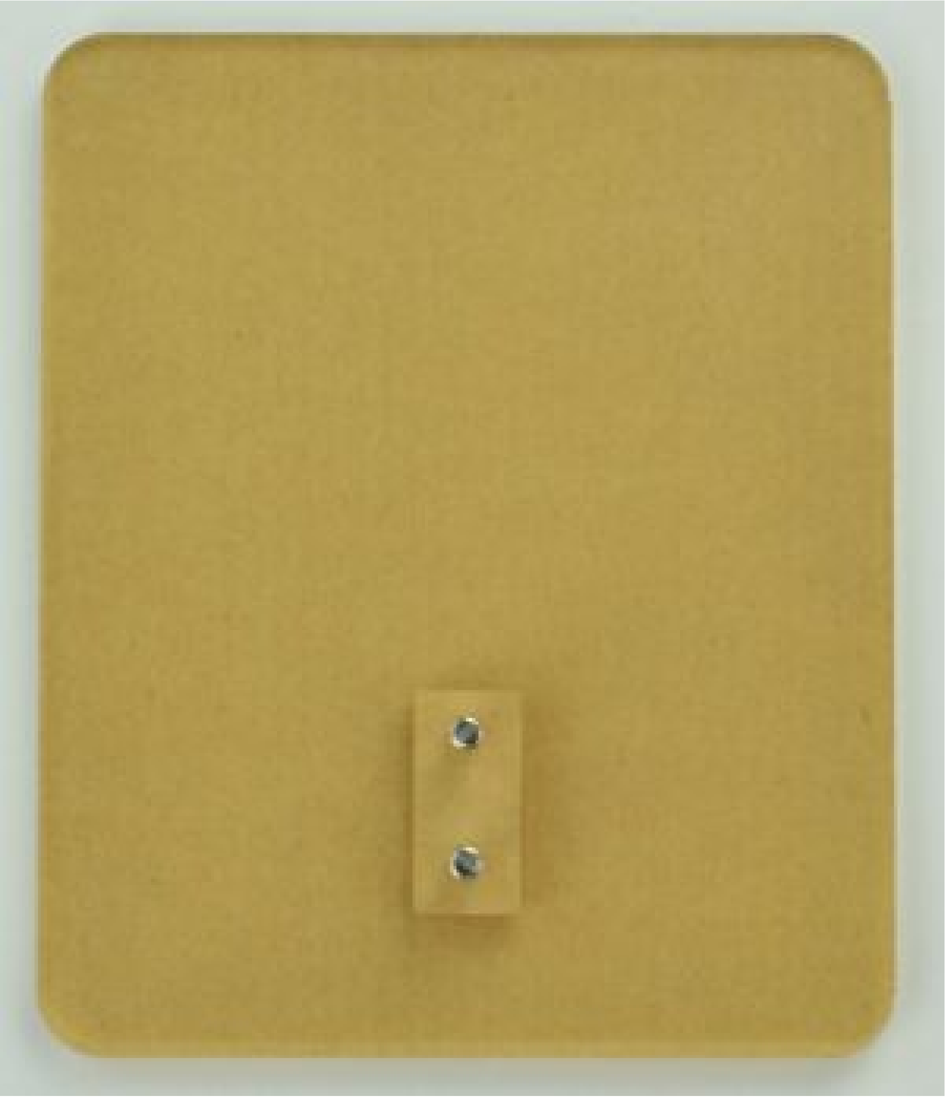
- Step1. Fix the adapter board on the base with 3MM screws, and pass the 5mm screws through the bottom plate and the gasket with 5mm aperture.



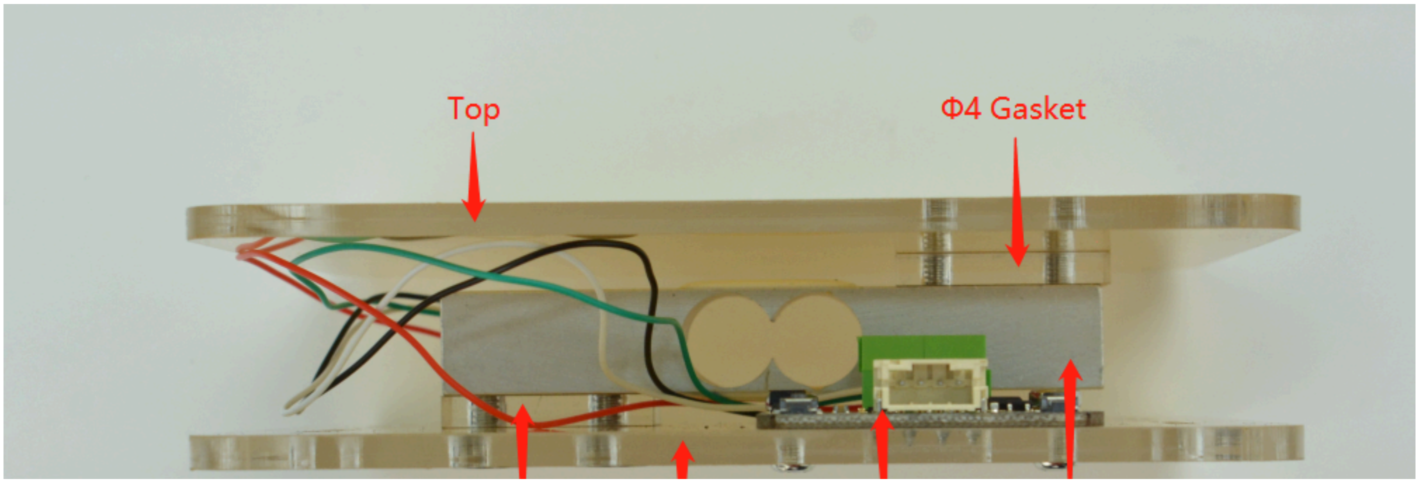
- Step2. Fix the weighing rod and gasket on the base, connect the board and the sensor, **note that the sticker with the green arrow should be downward**.



- Step3. Pass the 4mm screw through the top plate and the spacer with 4mm hole diameter.



- Step4. Fix the top plate and gasket on the weighing beam, and finally paste the gasket on the bottom plate.



Tutorial

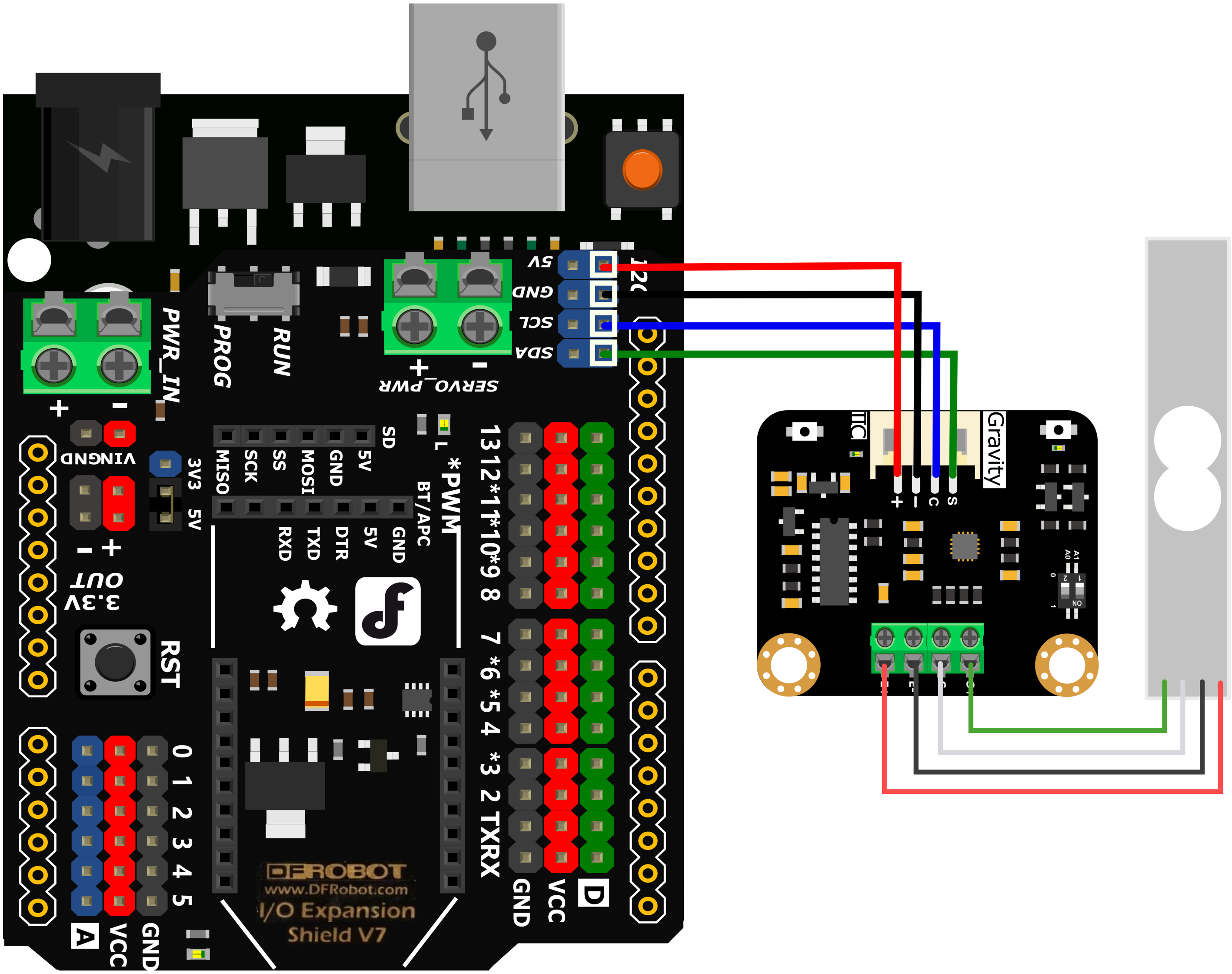
Requirements

- **Hardware**
  - DFRduino UNO R3 (https://www.dfrobot.com/product-838.html) (or similar) x 1
  - IO Expansion Shield for Arduino V7.1 (https://www.dfrobot.com/product-1009.html) x1
  - Weight Sensor Kit (https://www.dfrobot.com/product-2289.html) x1
  - Object with known weight
- **Software**
  - Arduino IDE (https://www.arduino.cc/en/Main/Software)
  - Download and install the **DFRobot HX711 I2C Library** (https://github.com/dfrobot/DFRobot_HX711_I2C) (About how to install the library? (https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0))
- Common API Function

```
/*!
   * @brief Constructor
   * @param pWire I2c controller
   * @param addr  I2C address(0x64/0x65/0x660x67)
   */
  DFRobot_HX711_I2C(TwoWire * pWire = &Wire,uint8_t addr = HX711_I2C_ADDR);

  /**
   * @brief Initialize function
   * @return Return 1 when initialization succeeds. Otherwise, return error co
   */
  int begin(void);

  /**
   * @brief Get object weight
   * @param The number of time to take avereage
   * @return Return the obtained weight，Unit: g
   */
  float readWeight(uint8_t times = 12);

  /**
   * @brief Get calibration value
   * @return Return the obtained calibration value
   */
  float getCalibration();

  /**
   * @brief Set/update calibration value
   * @param Value
   */
  void  setCalibration(float value);

  /**
   * @brief Set the sensor auto-calibration trigger threshold(g)
   * @param Threshold data
   */
```

```
void  setThreshold(uint16_t threshold);

/**
 * @brief Set the sensor calibration weight for auto-calibration(g)
 * @param Weight data
 */
void  setCalWeight(uint16_t triWeight);

/**
 * @brief Enable calibration (by software）
 */
void enableCal();

/**
 * @brief Tare (by software)
 */
void peel();

/**
 * @brief Wait for the sensor to complete calibration
 * @return Calibation completed, return: True
 *         Calibration failed, return: False
 */
bool getCalFlag();
```

Connection Diagram



Auto calibration - Button

Calibration can be performed at any time by button while the sensor is working. The calibration steps are as follows:

- Step 1. Assemble the kit and empty the objects on the sensor (if it is an additional tray, you can keep the tray)

- Step 2. Connect the sensor correctly according to the wiring diagram, and upload the sample code. After the upload is successful, open the serial monitor and adjust the baud rate to 9600

- Step 3. Press the cal button on the board, the indicator next to cal button will light up, at this time it enters the state of waiting for calibration
- Step 4. Put an object weighing 100g on the scale within 5s (100g is the default value, it can be modified), the cal light will go out after placing the object for 1s, at this time it enter the calibration state. After the calibration is completed, the indicator light will flash three times and go out

(After the object is placed, there will be a delay of 1 second for the anti-shake time, so do not worry. After the object is placed, do not touch the object before the calibration is completed)

- **Note: If there is no object placed within 5S after pressing the cal key, the indicator light will continue to light for 5S and then go out after 5S. This calibration is invalid. If it continues to fail, please check Q6 in the FAQ (https://wiki.dfrobot.com/HX711_Weight_Sensor_Kit_SKU_KIT0176#target_11) .**
- **When opening the serial monitor, the arduino will automatically reset and clear, so you must strictly follow the above steps**

Download and install the **DFRobot HX711 I2C Library** (https://github.com/dfrobot/DFRobot_HX711_I2C).

```
#include <DFRobot_HX711_I2C.h>

//DFRobot_HX711_I2C MyScale(&Wire,/*addr=*/0x64);
DFRobot_HX711_I2C MyScale;

float Weight = 0;

void setup() {
  Serial.begin(9600);
  while (!MyScale.begin()) {
    Serial.println("The initialization of the chip is failed, please confirm w
    delay(1000);
  }
  //// Set the calibration weight when the weight sensor module is automatical
  MyScale.setCalWeight(100);
  // Set the trigger threshold (G) for automatic calibration of the weight ser
  // This value cannot be greater than the calibration weight of the setCalWei
  MyScale.setThreshold(30);
  // Obtain the calibration value. The accurate calibration value can be obtai
  Serial.print("the calibration value of the sensor is: ");
  Serial.println(MyScale.getCalibration());
  MyScale.setCalibration(MyScale.getCalibration());
  delay(1000);
}

void loop() {
  Weight = MyScale.readWeight();
  Serial.print("weight is: ");
  if(Weight > 0.5){
    Serial.print(Weight, 1);
  }
  else{
    Serial.print(0, 1);
  }
  Serial.println(" g");
  delay(1000);
  }
```

Auto calibration - Software

you can also calibrate the sensor through software by using the enableCal () function. (Button calibration method, you can calibrate at any time when the sensor is working, please see the previous step for the method).

- Step 1. Assemble the kit and remove the other objects on the sensor except the scale surface
- Step 2. Connect the sensor correctly according to the wiring diagram, and upload the sample code. After the upload is successful, open the serial monitor and adjust the baud rate to 9600
- Step 3. After the CAL light on the board turns on or the serial port prints "Please put the object within 5s", put an object weighing 100g on the scale (100g is the default value, it can be modified)
- Step 4. After waiting for a few seconds, it will output the calibrated weight value.
- **When opening the serial monitor, the arduino will automatically reset and clear, so you must strictly follow the above steps**

```
#include <DFRobot_HX711_I2C.h>

//DFRobot_HX711_I2C MyScale(&Wire,/*addr=*/0x64);
DFRobot_HX711_I2C MyScale;

void setup() {
  Serial.begin(9600);
  while (!MyScale.begin()) {
    Serial.println("The initialization of the chip is failed, please confirm w
    delay(1000);
  }
  //Set the calibration weight when the weight sensor module is automatically 
  MyScale.setCalWeight(100);
  // Set the trigger threshold (G) for automatic calibration of the weight ser
  // This value cannot be greater than the calibration weight of the setCalWei
  MyScale.setThreshold(30);

  //Start sensor calibration
  Serial.println("Please put the object within 5S");
  MyScale.enableCal();
  long time1 = millis();
  //Wait for sensor calibration to complete
  while(!MyScale.getCalFlag()){
       delay(1000);
       if((millis()-time1) > 7000){
          Serial.println("The calibration failed this time, and no object is d
          break;
       }
  }
  //Obtain the calibration value. The accurate calibration value can be obtain
  Serial.print("the calibration value of the sensor is: ");
  Serial.println(MyScale.getCalibration());
  MyScale.setCalibration(MyScale.getCalibration());
}

void loop(){
  float Weight = MyScale.readWeight();
  Serial.print("weight is: ");
  if(Weight > 0.5){
    Serial.print(Weight, 1);
  }
  else{
    Serial.print(0, 1);
  }
  Serial.println(" g");
  delay(1000);
}
```

Quick Start Example

This method is suitable for quick experience, taking the most common DFRduino UNO R3 + IO sensor expansion board V7.1 as an example for quick demonstration (this method requires at least two main controller board)

It is recommended to use 100g-500g objects for calibration in actual use, the data will be more accurate.

- Step 1. Assemble the kit and remove the objects on the sensor (if it comes with an additional tray, you can keep the tray).
- Step 2. Connect the sensor correctly according to the wiring diagram, and upload the sample code. After the upload is successful, open the serial monitor and adjust the baud rate to 9600.
- Step 3. Press the cal button on the board, the indicator next to cal button will light up, at this time it enters the state of waiting for calibration.
- Step 4. Put the DFRduino UNO R3 + IO sensor expansion board V7.1 on the scale within 5s, (DFRduino UNO R3 + IO sensor expansion board V7.1 = 50g) the cal light will go out after placing the object for 1s, at this time it enter the calibration state. When the calibration is completed, the indicator light will flash three times and go out (After the object is placed, there will be a delay of 1

second for the anti-shake time, so do not worry. After the object is placed, do not touch the object before the calibration is completed).

- **Note: If there is no object placed within 5S after pressing the cal key, the indicator light will continue to light for 5S and then go out after 5S. This calibration is invalid. If it continues to fail, please check Q6 in the FAQ (https://wiki.dfrobot.com/HX711_Weight_Sensor_Kit_SKU_KIT0176#target_11) .**
- **When opening the serial monitor, the arduino will automatically reset and clear, so you must strictly follow the above steps**

```
#include <DFRobot_HX711_I2C.h>

//DFRobot_HX711_I2C MyScale(&Wire,/*addr=*/0x64);
DFRobot_HX711_I2C MyScale;

float Weight = 0;

void setup() {
  Serial.begin(9600);
  while (!MyScale.begin()) {
    Serial.println("The initialization of the chip is failed, please confirm w
    delay(1000);
  }
  //// Set the calibration weight when the weight sensor module is automatical
  MyScale.setCalWeight(50);
  // Set the trigger threshold (G) for automatic calibration of the weight ser
  // This value cannot be greater than the calibration weight of the setCalWei
  MyScale.setThreshold(30);
  // Obtain the calibration value. The accurate calibration value can be obtai
  Serial.print("the calibration value of the sensor is: ");
  Serial.println(MyScale.getCalibration());
  MyScale.setCalibration(MyScale.getCalibration());
  delay(1000);
}

void loop() {
  Weight = MyScale.readWeight();
  Serial.print("weight is: ");
  if(Weight > 0.5){
    Serial.print(Weight, 1);
  }
  else{
    Serial.print(0, 1);
  }
  Serial.println(" g");
  delay(1000);
  }
```

Compatibility

MCU	Work Well	Work Wrong	Untested	Remarks
Arduino uno	√			
FireBeetle-ESP8266	√			
FireBeetle-ESP32	√			
mPython	√			
Raspberry Pi	√			
micro:bit	√			

FAQ

- During calibration and measurement, the most accurate data will be obtained when the object is placed in the middle area of the scale.
- After pressing the rst key, no matter how many objects are placed on the sensor, the sensor will clear the current weight value, and the weight of the objects placed later will accumulate from 0, thus achieving the "tare" function.
- **Every time you reset/restart the Arduino or open the serial port monitor, the system will "tare" once by default**
- The sensor calibration method: put an object of known weight on the sensor, and then "tell" the sensor what the weight of the object is, then the sensor will be calibrated (single-point calibration).
- Q1. How to display negative weight information?

A: Replace the loop function in the code.

```
void loop() {
  Serial.print("weight is: ");
  Serial.print(MyScale.readWeight(), 1);
  Serial.println(" g");
  delay(1000);
}
```

- Q2. How to judge whether the automatic calibration is successful?	
A: When the calibration is successful, the indicator above the calibration button will flash three times and go out. Otherwise, the calibration fails.	
- Q3. Although the automatic calibration is successful, the output weight information has a small deviation (<0.5g).	
A: Sometimes the object is not stably placed, which will cause a small error, you can calibrate it again.	
- Q4. After the automatic calibration is successful, the output weight information has a large deviation (>3g), although multiple calibrations have been performed.	
A: Please check that the four leads of the sensor (weighing beam) are not entangled too much. Too much winding will affect the accuracy of the data. Make sure that the scale surface is not tilted.	
- Q5. What should the weight of the calibration object be?	
A: Take the middle value. For example, if the measurement range is about 0-600g, the weight of the calibration object is preferably 300g (other weights are also possible, the middle value is the ideal value).	
- Q6: When calibrating a 100g object, why the calibration is not started after	

the 100g object is placed, and the calibration fails all the time.	
A: A "trigger weight" is required for calibration. Press the calibration button and the sensor will only start calibration when it detects that something is put on and exceeds the "trigger weight". When in the first use, because there is no calibration, the data read of the 100g object by sensor may only be 20g. At this time, you need to reduce the value of MyScale.setThreshold(30) (change 30 to 10 or lower), and then upload the code again.	
- Q7: No matter whether a object is placed or not, the sensor output data is always 0.	
A: This is caused by placing the object on the scale before the Arduino completes the reset. Solution: Take off the object on the scale, press the reset button of Arduino or the rst button of the sensor, and put the object on again, then it can be used normally.	
Q8: Why use the setCalibration() function to set the calibration value to 2000, but use the getCalibration() function to check the current calibration value is not 2000.	
A: The getCalibration() function can only obtain the calibration value of the "key-press calibration" or enableCal() function, but cannot obtain the calibration value set by the setCalibration() function	

More Documents

- Weight Sensor Pinout (https://dfimg.dfrobot.com/nobody/wiki/4bbc4ac20c277d367712cda8acc346b2.pdf)
- Weight Sensor Dimension (https://dfimg.dfrobot.com/nobody/wiki/cc8433ddd2eff8c23e212303067a3c3f.p

df)

- Mindplus Library file (https://gitee.com/chenqi1233/ext-dfrobot_-hx711_-i2-c)

- Drawings (https://dfimg.dfrobot.com/nobody/wiki/cbdcad2977f3a7d4a7210f1d91991402.jpg)