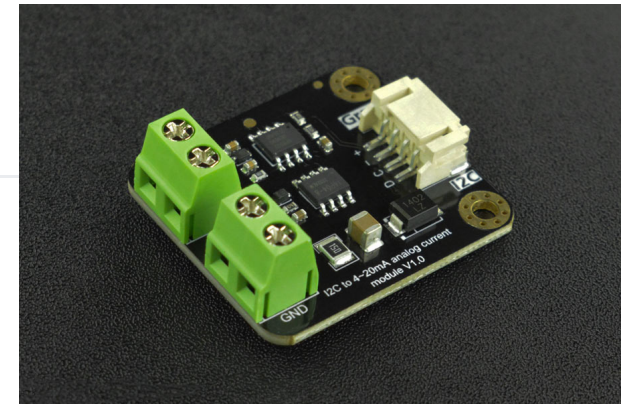


SKU:DFR0972 (<https://www.dfrobot.com/product-2625.html>)

(<https://www.dfrobot.com/product-2625.html>)

Introduction

This is an I2C DAC module that outputs 0-25mA current. It offers more accurate 4-20mA current output after calibration, which can be used with most 4-20mA devices on markets such as motors, inverters, valves, pumps, etc., by Arduino programming.



Features

- I2C signal input, linearly converted to 0-25mA analog current output.
- With Gravity interface, support Arduino library, and can be used for program automation control.
- Configuration can be saved when powered off, and the saved value remains after restarting.

Applications

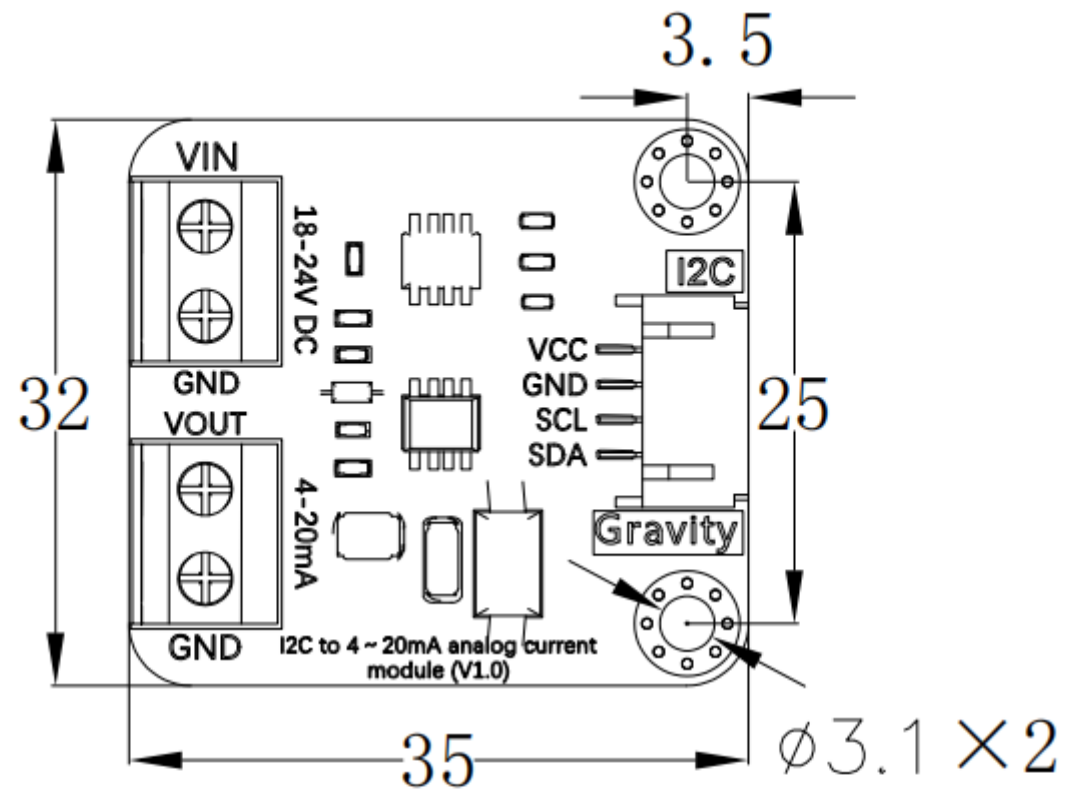
- Intelligent function signal generator.
- Automatic controller for lighting or fountain.
- The automatic and quick transformation of small equipment.
- Automatic debugger for electromechanical equipment.
- Transformation of small test equipment.

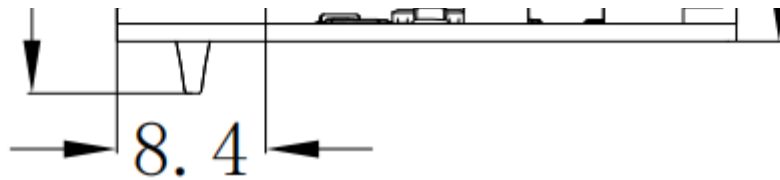
- Transformation of small test equipment.

Specification

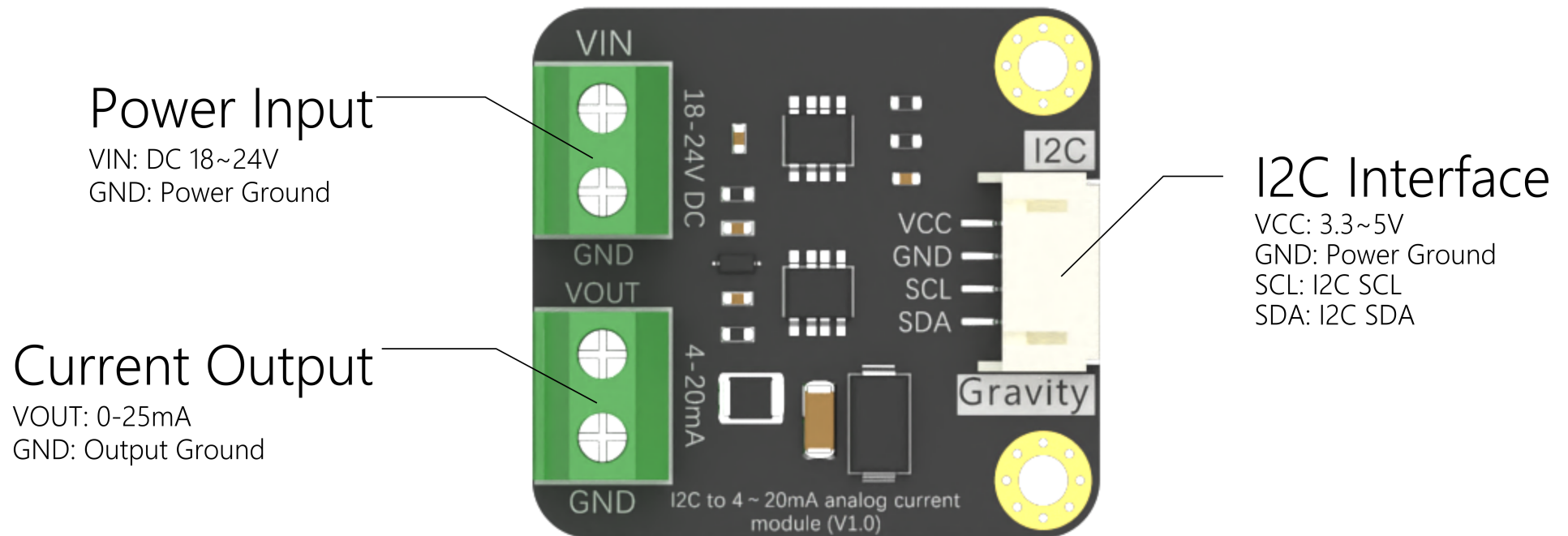
- Chip Type: GP8302
- Input Signal Range: 12Bit, 0x000-0xFFFF (0-25mA)
- Input I2C Signal High Level: 2.7V-5V
- Power Supply: 18V-24V (24V recommended)
- Accuracy: 0.5% before calibration, 0.2% after calibration
- Power Consumption: <5mA
- Start-up Time: <2ms
- Operating Temperature: -40°C to 125°C
- Load Resistance: Maximum 450Ω load capacity for 18V power supply, maximum 750Ω for 24V
- Output Current: 0-25mA
- I2C Address: 0x58

Dimension Diagram





Function Indication



Tutorial

Requirements

- Hardware

• 1 x DFRduino UNO R3 (https://www.dfrobot.com/product_828.html) (or similar) x 1
https://wiki.dfrobot.com/SKU_DFR0972_I2C_4_20mA_DAC_Module

- DFRduino UNO R3 (<https://www.dfrobot.com/product-638.html>) (or similar) x 1
- DFR0972_I2C to 4-20mA Analog Current Module x 1
- Multimeter x 1
- Resistance of 220Ω ~ 650Ω (greater than 1/8 watt) x 1
- Experimental Power supply
- M-M/F-M/F-F Jumper wires
- **Software**
 - Arduino IDE (<https://www.arduino.cc/en/Main/Software>)
 - Download and install the **DFR0972 Library** (https://github.com/cdjq/DFRobot_GP8302) (About how to install the library? (<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>))

Hardware Connection

Step 1: Insert the probe of the multimeter into the jack for current, and change it to DC current, as shown in the figure:

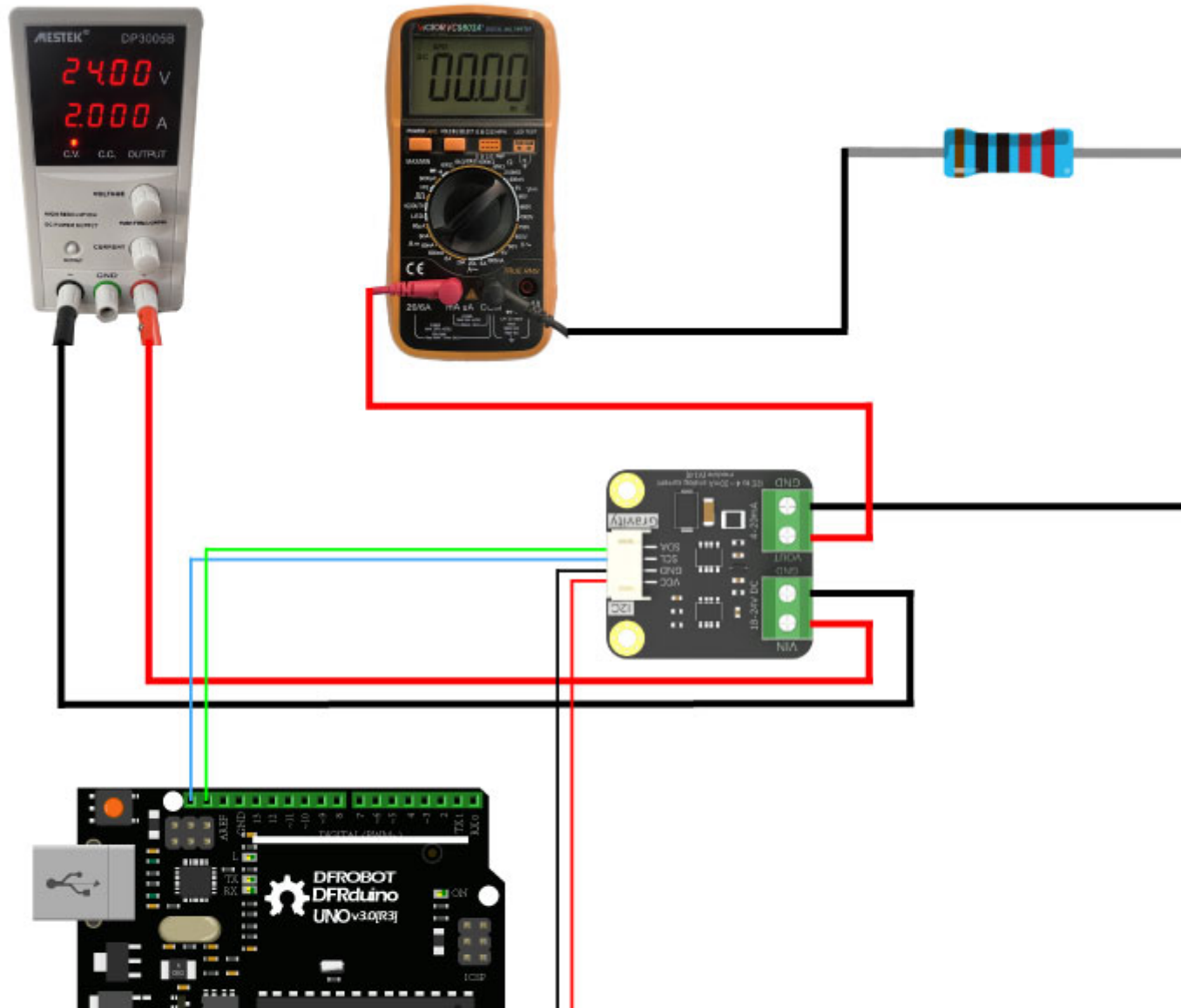


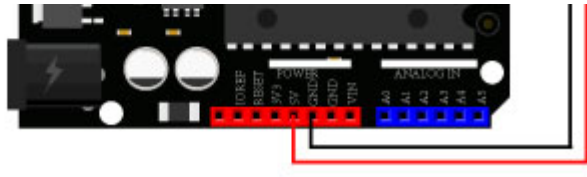


Step 2: Adjust the voltage of the experimental power supply to 24V, as shown in the figure:



Step 3: Connect the UNO main board, the experimental power supply, the current module, the load resistor and the multimeter according to the figure:





The test proved that the result is incorrect when using load meter as load resistor, because the load meter is not a pure resistance load, which will lead to inaccurate output voltage.

Programming

Write and download the following code into the UNO main board to make the current module output a current of 10mA, and save the value in the module. Changing the value in the "module.output(10)" function can change the output current, the range of the value is 0-25, corresponding to 0-25mA respectively.

```
#include "DFRobot_GP8302.h"

DFRobot_GP8302 module;

void setup(){
    Serial.begin(115200);

    while(!Serial){
        ;
    }
    Serial.print("I2C to 0-25 mA analog current module initialization ... ");
    uint8_t status = module.begin(); // use the pin used by the I2C Wire object of the MCU hardware by default

    if(status != 0){
        Serial.print("failed. Error code: ");
        Serial.println(status);
        Serial.println("Error Code: ");
        Serial.println("\t1: _scl or _sda pin is invalid.");
        Serial.println("\t2: Device not found, please check if the device is connected.");
        while(1) yield();
    }
    Serial.println("done!");

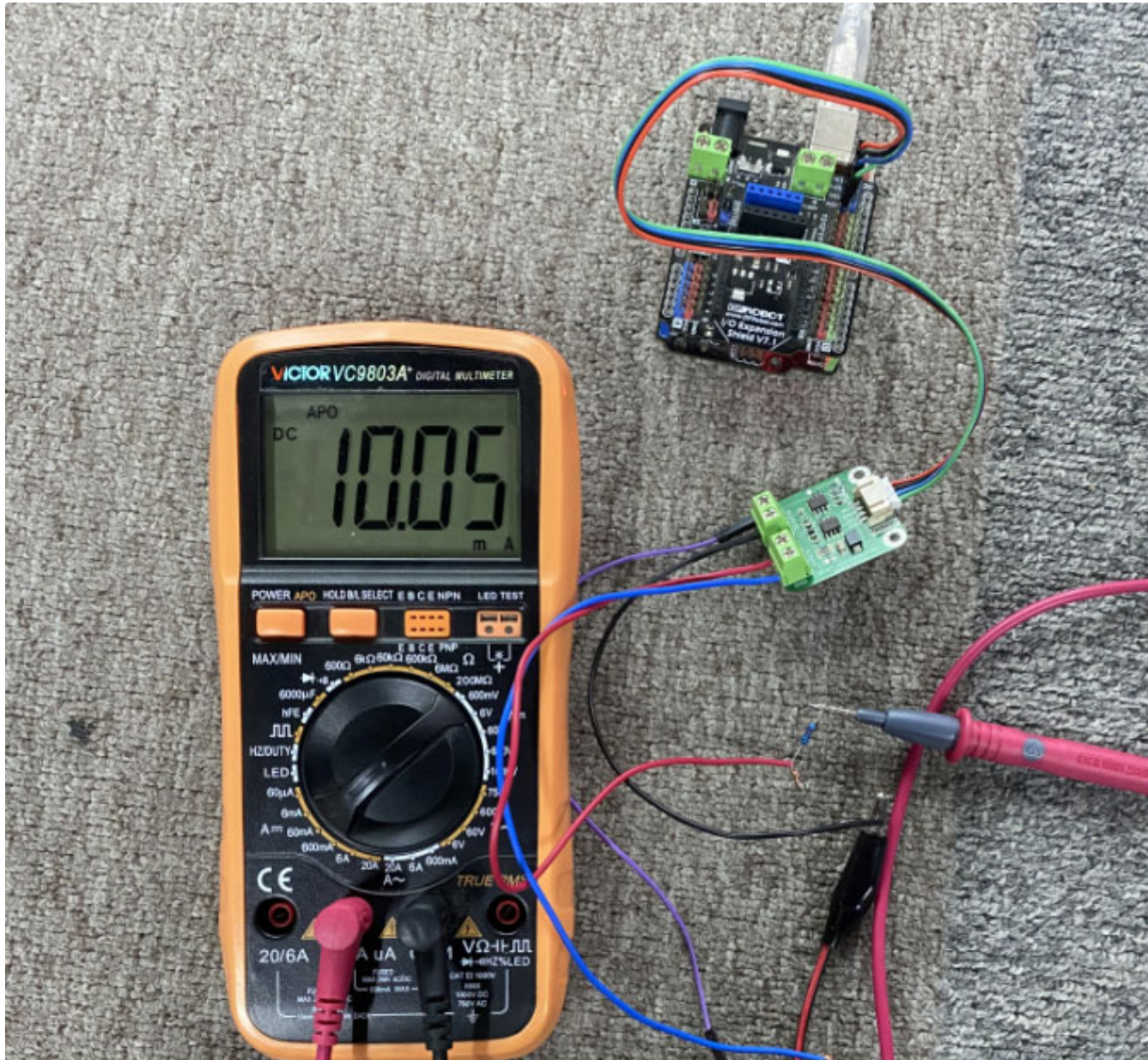
    uint16_t dac = module.output(10); //control the converter module to output a current of 10mA and return the corresponding
    Serial.print("DAC value: 0x"); Serial.println(dac, HEX);
    module.store(); Serial.println("Save current configuration."); //the above current config will be saved and will not be
}

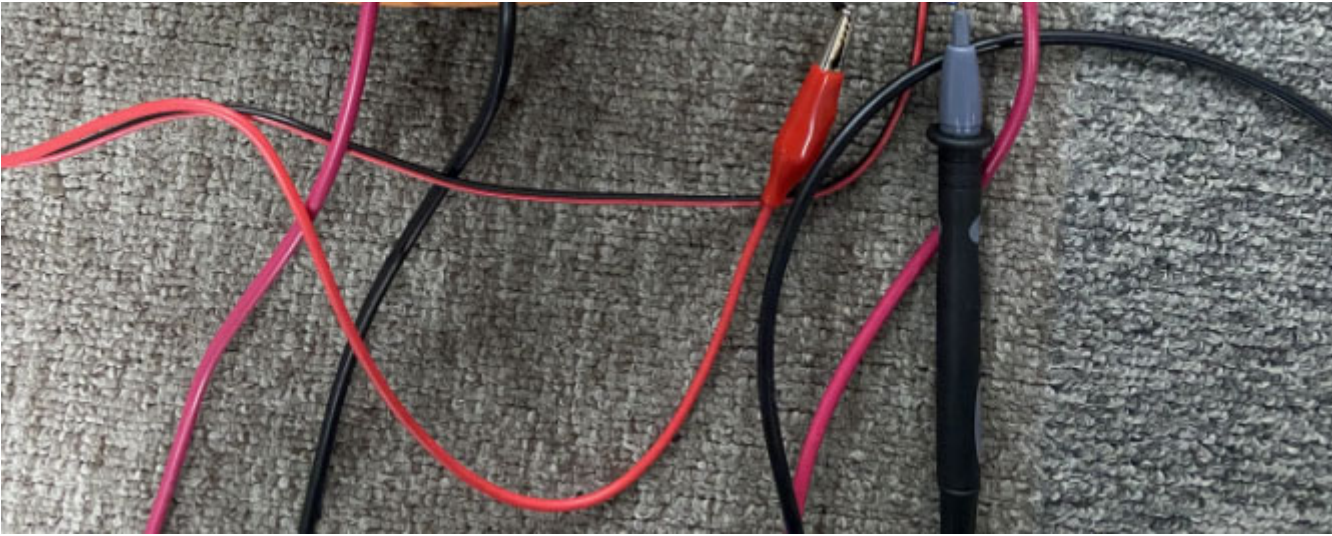
void loop(){
```

```
}
```

Result

As shown in the figure, the multimeter shows an output current of 10.05mA. The error is 0.05mA (0.5%), which is resulted from the accuracy of the multimeter and lack of calibration.





Calibrate to Obtain Accurate Current Within 4-20mA

The module supports outputting current within 0-25mA. And through a two-point calibration of 4mA and 20mA, an accurate current within 4-20mA can be obtained. The Arduino library provides the functions for easy current calibration.

For calibration, first input the exact DAC value for 4mA and for 20mA respectively, then output the current of 4mA and check the actual output value on the ammeter, finally adjust the DAC value until the actual output is exactly 4mA; similarly, output the current of 20mA and check the actual output value on the ammeter, finally adjust the DAC value until the actual output is exactly 20mA. Then the values in 4-20mA are calibrated.

Calibration Steps:

- Connect the wiring as the connection diagram above.
- Write the following code:

```
#include "DFRobot_GP8302.h"

DFRobot_GP8302 module;

void setup(){
  Serial.begin(115200);

  while(!Serial){
    //Wait for USB serial port to connect. Needed for native USB port only
  }

  Serial.print("I2C to 0-25 mA analog current module initialization ... ");

  uint8_t status = module.begin(); // use the pin used by the I2C Wire object of the MCU hardware by default

  if(status != 0){
    Serial.print("failed. Error code: ");
    Serial.println(status);
    Serial.println("Error Code: ");
    Serial.println("\t1: _scl or _sda pin is invalid.");
    Serial.println("\t2: Device not found, please check if the device is connected.");
    while(1) yield();
  }
  Serial.println("done!");

  module.calibration4_20mA(/*dac_4 =*/652, /*dac_20 =*/3265);

  uint16_t dac = module.output(/*current_mA =*/4);
}

void loop() {
```



```
void loop() {  
  
}
```

Calibrate 4mA:

- Set the parameter in the statement to 4 and output 4mA current: `uint16_t dac = module.output(/*current_mA ==*/4);`
- Download the code and check the actual output value of the ammeter.
- If the actual output current is 4.05mA, then lower the DAC value 652 corresponding to 4mA in the code statement, for example, change it to 645: `module.calibration4_20mA(/*dac_4 ==*/652, /*dac_20 ==*/3265);`
- Redownload the code and check the actual current value. Repeatedly modify the DAC value until the measured current is exactly 4mA.
- The calibration of 4mA is complete.

Calibrate 20mA:

- Set the parameter in the statement to 20 and output 20mA current: `uint16_t dac = module.output(/*current_mA ==*/20);`
- Download the code and check the actual output value of the ammeter.
- If the actual output current is 20.05mA, then lower the DAC value 3265 corresponding to 20mA in the statement, for example, change it to 3260: `module.calibration4_20mA(/*dac_4 ==*/652, /*dac_20 ==*/3265);`
- Redownload the code and check the actual current value. Repeatedly modify the DAC value until the measured current is exactly 20mA.
- The calibration of 20mA is complete.

So the calibration of the current within 4-20mA is complete. The calibrated current value within 4-20mA can be output using this statement:

```
uint16_t dac = module.output(/*current_mA ==*/10);
```

Function Library Definition

Function prototype: `uint8_t begin(int scl = -1, int sda = -1)`

Function brief: initialize the current module, and configure the I2C pin.

Parameter: scl IO port pin of the MCU, sda IO port pin of the MCU; when scl and sda use the default formal parameter, they

Return value: 0 init success; 1 the I2C pin of the MCU connected to the I2C to current DAC module is invalid; 2 device not

Function prototype: `float output_mA(uint16_t dac)`

Function brief: set DAC value to control the device to output the current, the range is 0~0xFFF, corresponding to outputting

Parameter: DAC value of 0~0xFFF

Return value: actual current, unit mA

Function prototype: `uint16_t output(float current_mA)`

Function brief: set current value to control the device to output the current, the range is 0-25, corresponding to outputting

Parameter: current value of 0-25

Return value: the hexadecimal DAC value corresponding to the output current value, range is 0~0~0xFFF

Function prototype: `void store()`

Function brief: save the current config, after the config is saved successfully, it will be enabled when the module is powered

Parameter: none

Return value: none

Function Prototype: `module.calibration4_20mA(dac_4, dac_20)`

Function brief: calibrate the current within 4~20mA

Parameter:

dac_4 range 0~4095, the calibration is invalid if the value is out of the range, the DAC value corresponding to the current

dac_20 range 0~4095, the calibration is invalid if the value is out of the range, the DAC value corresponding to the current

dac_4 and dac_20 should meet the condition: $dac_4 < dac_20$

Return value: none

More Documents

DFR0972_Schematics.pdf (<https://dfimg.dfrobot.com/nobody/wiki/8b0137351912abee5b44c26e0943d10d.pdf>)

DFR0972_Dimensions.pdf (<https://dfimg.dfrobot.com/nobody/wiki/164f16980d0aecab768510246e356059.pdf>)

DFR0972_2D_CAD File.rar (<https://dfimg.dfrobot.com/nobody/wiki/e900e91ee222f7ec74672f6efd05a3bb.rar>)

DFR0972_3D Model File.rar (<https://dfimg.dfrobot.com/nobody/wiki/042b66c40f8f104eb1d27e45ba0dfdce.rar>)

GP8302 Datasheet.pdf (<https://dfimg.dfrobot.com/nobody/wiki/aaebe556d0dd83e0a3961830377d79ee.pdf>)

FAQ

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



Get **I2C to 4-20mA Analog Current Module** (<https://www.dfrobot.com/product-2625.html>) from DFRobot Store or **DFRobot Distributor**. (<https://www.dfrobot.com/distributor>)

Turn to the Top