

SKU:DFR0971 (<https://www.dfrobot.com/product-2613.html>)

(<https://www.dfrobot.com/product-2613.html>)

Introduction

This 2-Channel I2C DAC module can output a voltage of 0-10V, which is a standard driver in industrial applications. It can be widely used with most 0-10V devices on markets such as light controllers, inverters, valves, pumps, etc., by Arduino programming. Besides, this module supports 8 different I2C addresses, which can be changed using the on-board DIP switch, allowing a cascade of 16 devices.



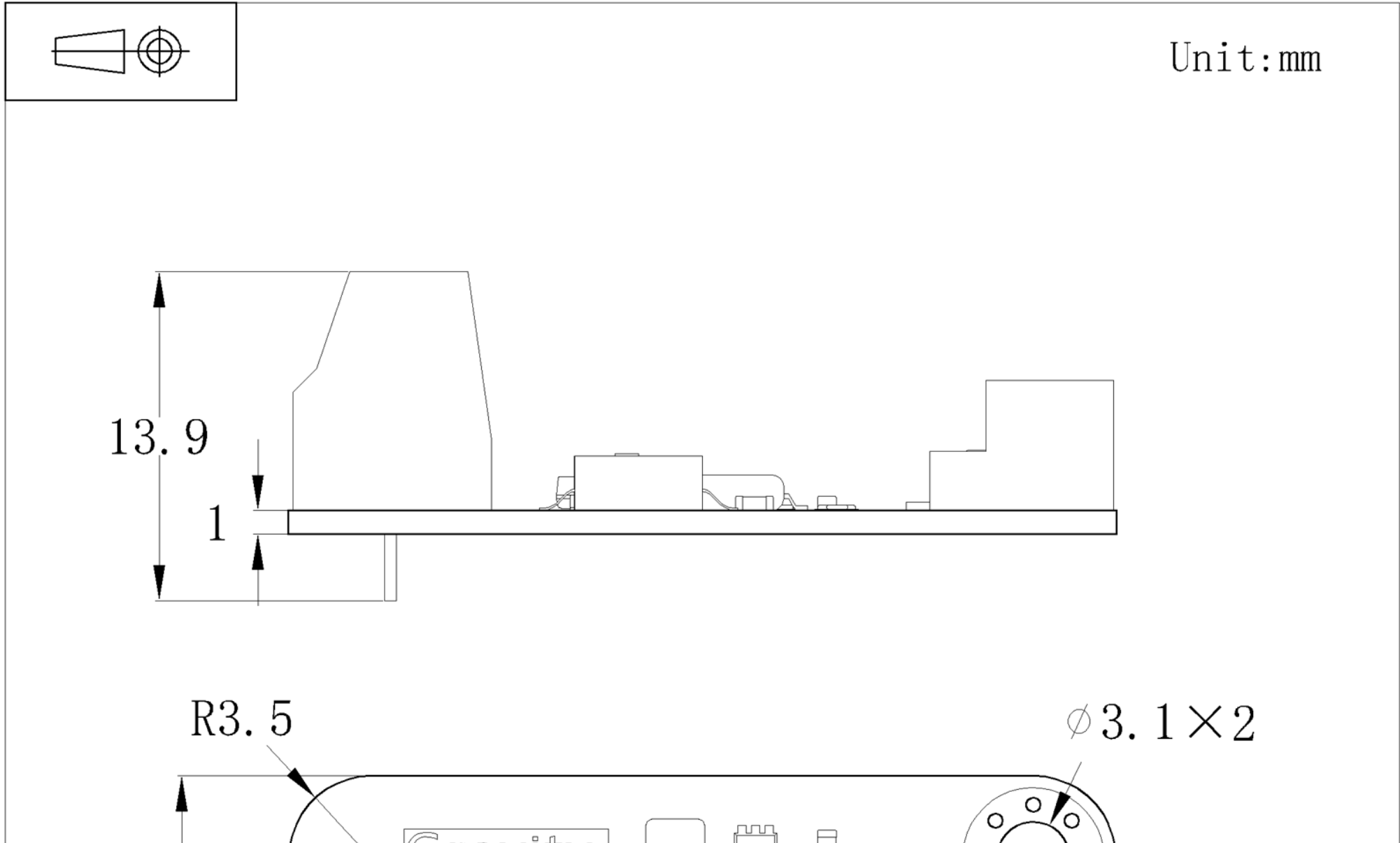
Features

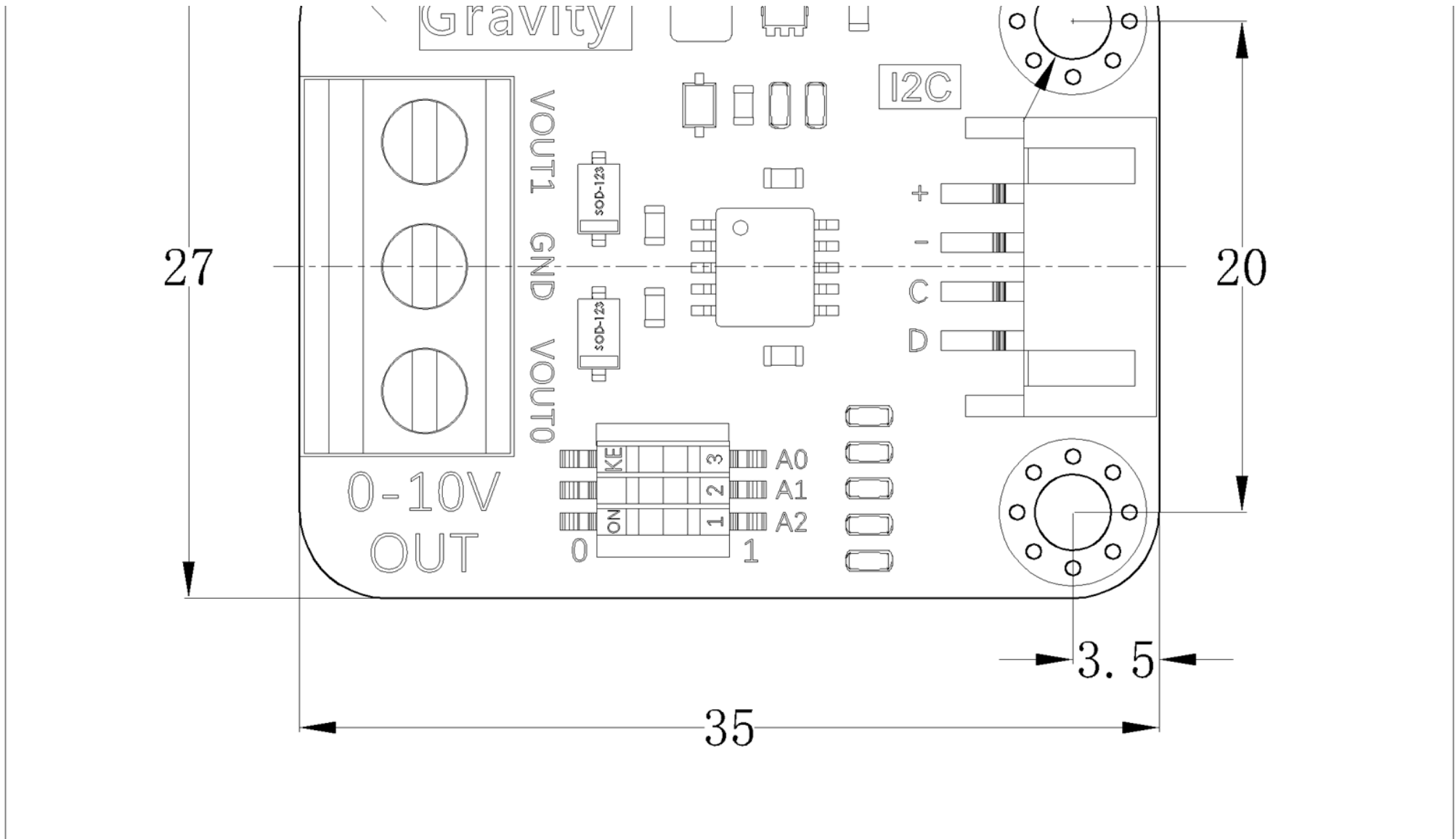
- Support 3.3V-5V power supply
- Output voltage error is less than 0.5%
- Output short-circuit protection, the chip enters protection mode and stops output when the output and GND are shorted.
- Two-channel 0-5V/0-10V output, can connect to and control standard analog voltage devices
- Gravity interface, I2C communication, Arduino control, can be used for program automation control
- Adjustable 8-way I2C addresses, can control 16 devices at the same time

Applications

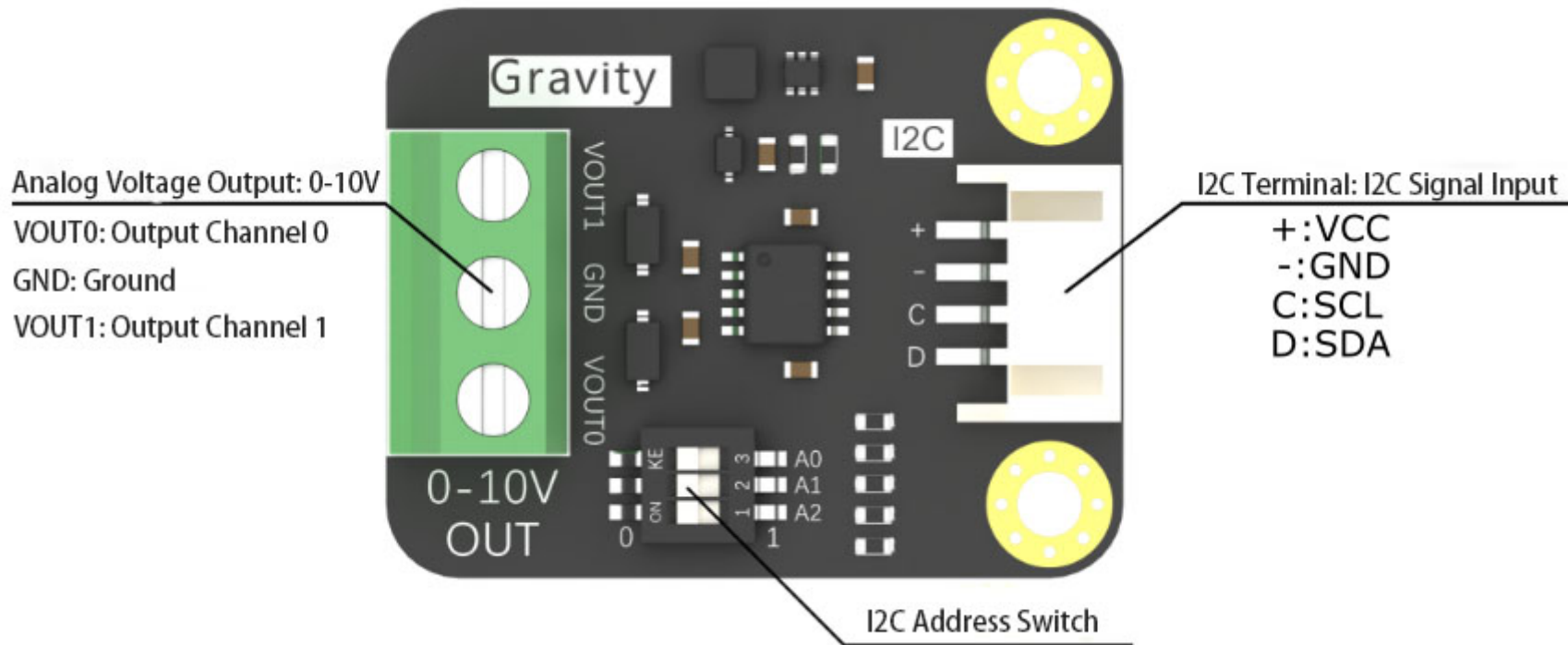
- Intelligent function signal generator.
- Automatic controlling of lighting or fountain.
- The automatic and quick transformation of small equipment.
- Automatic debugger for electromechanical equipment.
- Transformation of small test equipment.

Dimension Diagram





Board Overview



Function Description

Name	Description	Remarks
I2C Terminal	I2C Signal Port	
I2C Address	Adjust the DIP switch values of A0, A1 and A2 to	The correspondence between I2C address and DIP switch

Switch	change the module I2C address	is shown in the table below
Analog Voltage Output	VOUT0 and VOUT1 can output 0-10V or 0-5V independently	

I2C Address Table

A0	A1	A2	I2C Address
0	0	0	0x58
0	0	1	0x59
0	1	0	0x5A
0	1	1	0x5B
1	0	0	0x5C
1	0	1	0x5D
1	1	0	0x5E
1	1	1	0x5F

Specification

- Operating Voltage: 3.3V~5V
- I2C Interface: PH2.0-4P (Gravity Line Sequence)
- Chip Type: GP8403
- Input Signal: 12Bit (0x000 - 0xFFFF)
- Input I2C Signal High Level: 2.7V - 5V
- VOUT0 Output Voltage: 0-10V/0-5V

- VOUT1 Output Voltage: 0-10V/0-5V
- Output Voltage Error: < 0.5%
- Linearity Error of Output Voltage: 0.1%

- Power Consumption: < 4mA
- Output Short-circuit Protection

Tutorial

Software Requirements

- Arduino IDE (<https://www.arduino.cc/en/Main/Software>)
- Install SDK: click to enter FireBeetle 2 ESP32-E WIKI page (https://wiki.dfrobot.com/FireBeetle_Board_ESP32_E_SKU_DFR0654) to find the SDK installation tutorial
- Download and install the **DFRobo_GP8403-main Library** (https://github.com/DFRobot/DFRobot_GP8403) (About how to install the library? (<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>))

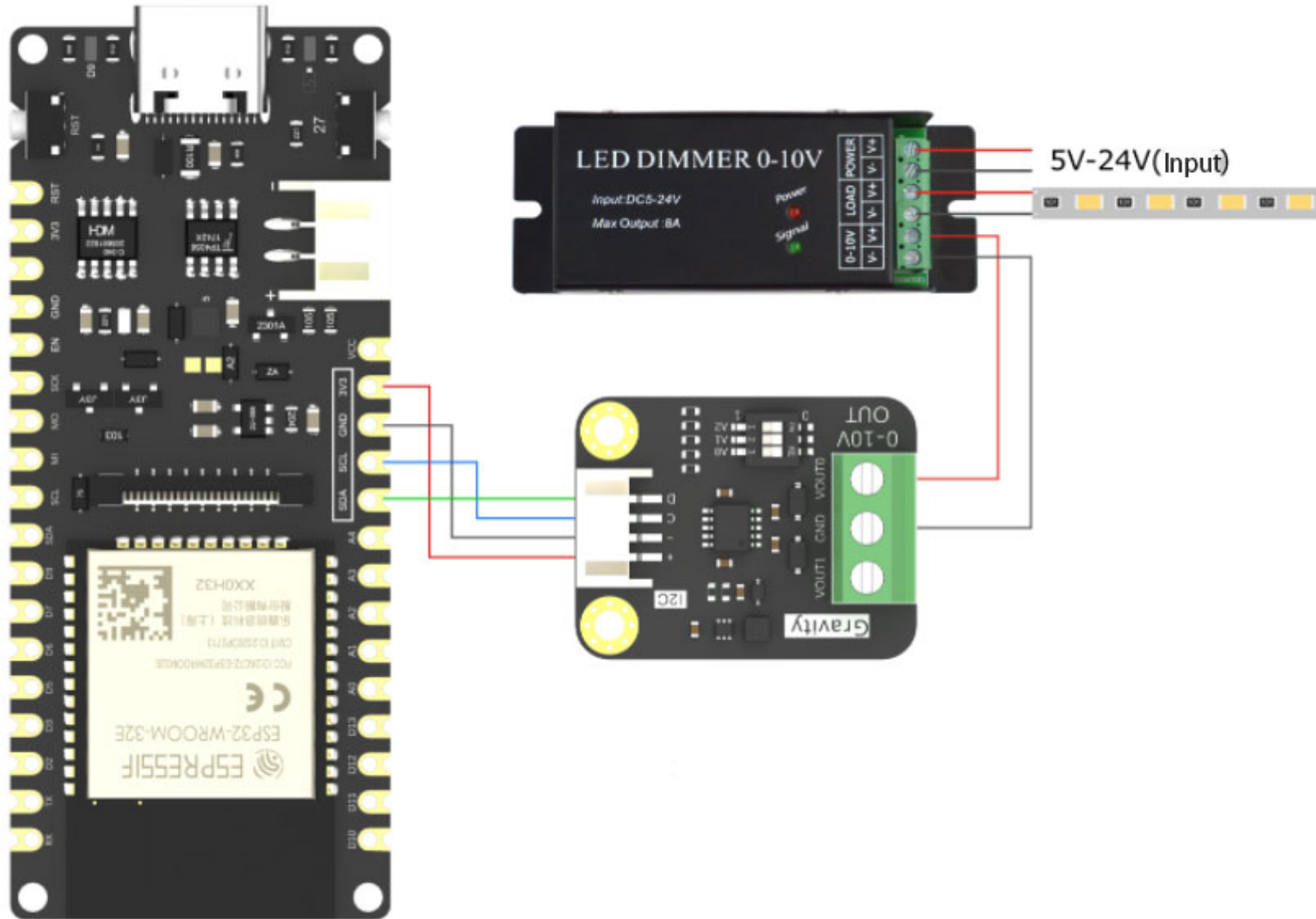
Hardware Requirements

- FireBeetle 2 ESP32-E (<https://www.dfrobot.com/product-2195.html>) x 1
- DFR0971 2-Channel I2C to 0-10V DAC Module x 1
- LED Dimmer x 1
- 5V LED Strip (<https://www.dfrobot.com/product-2403.html>) x 1
- M-M/F-M/F-F Jumper wires
- Oscilloscope x 1

Control LED Strip Brightness

Hardware Connection

Hardware Connection



Wring Description

FireBeetle 2: 3V3 to DAC module: +

FireBeetle 2: GND to DAC module: -

FireBeetle 2: SCL to DAC module: SCL

FireBeetle 2: SDA to DAC module: SDA

The DAC module: VOUT0 to LED dimmer (0-10V): V+

The DAC module: GND to LED dimmer (0-10V): V-

The LED dimmer (LOAD): V+ to 5V LED Strip: +

The LED dimmer (LOAD): V- to 5V LED Strip: -

The LED dimmer (POWER): V+ to 5V-24V power input: 5V +

The LED dimmer (POWER): V- to 5V-24V power input: 5V -

Sample Code

The following code will demonstrate how to output a 3.5V signal through the OUT0 port to provide the LED strip with signal voltage for adjusting the brightness of the strip connected to the driver. Two APIs are used in this sample code:

API Name: `setDACOutRange (e0utPutRange range)`

API Function: set the range of the output voltage, 5V or 10V;

Sample: `dac.setDACOutRange(dac.e0utputRange10V)`

API Name: `setDACOutVoltage(uint16_t data,uint8_t channel);`

API function: set the DAC value corresponding to the output voltage. data : Output voltage values, 5V range : 0-5000; 10V range : 0-10000 channel : channel select, 0: channel 0; 1: channel 1; 2: all the channels

10000. CHANNEL 1. CHANNEL SELECT, 0. CHANNEL 0, 1. CHANNEL 1, 2. ALL THE CHANNELS.

Sample: `dac.setDACOutVoltage(3500,0) // output 3.5V via channel 0.`

```
#include "DFRobot_GP8403.h"
DFRobot_GP8403 dac(&Wire,0x58);

void setup() {
  Serial.begin(115200);
  while(dac.begin()!=0){
    Serial.println("init error");
    delay(1000);
  }
  Serial.println("init succeed");
  dac.setDACOutRange(dac.eOutputRange10V);//Set the output range as 0-10V
  dac.setDACOutVoltage(1433,0);//The DAC value for 3.5V output in OUT0 channel
  delay(1000);
  dac.store(); //Save the set 3.5V voltage inside the chip
}

void loop(){

}
```

Result

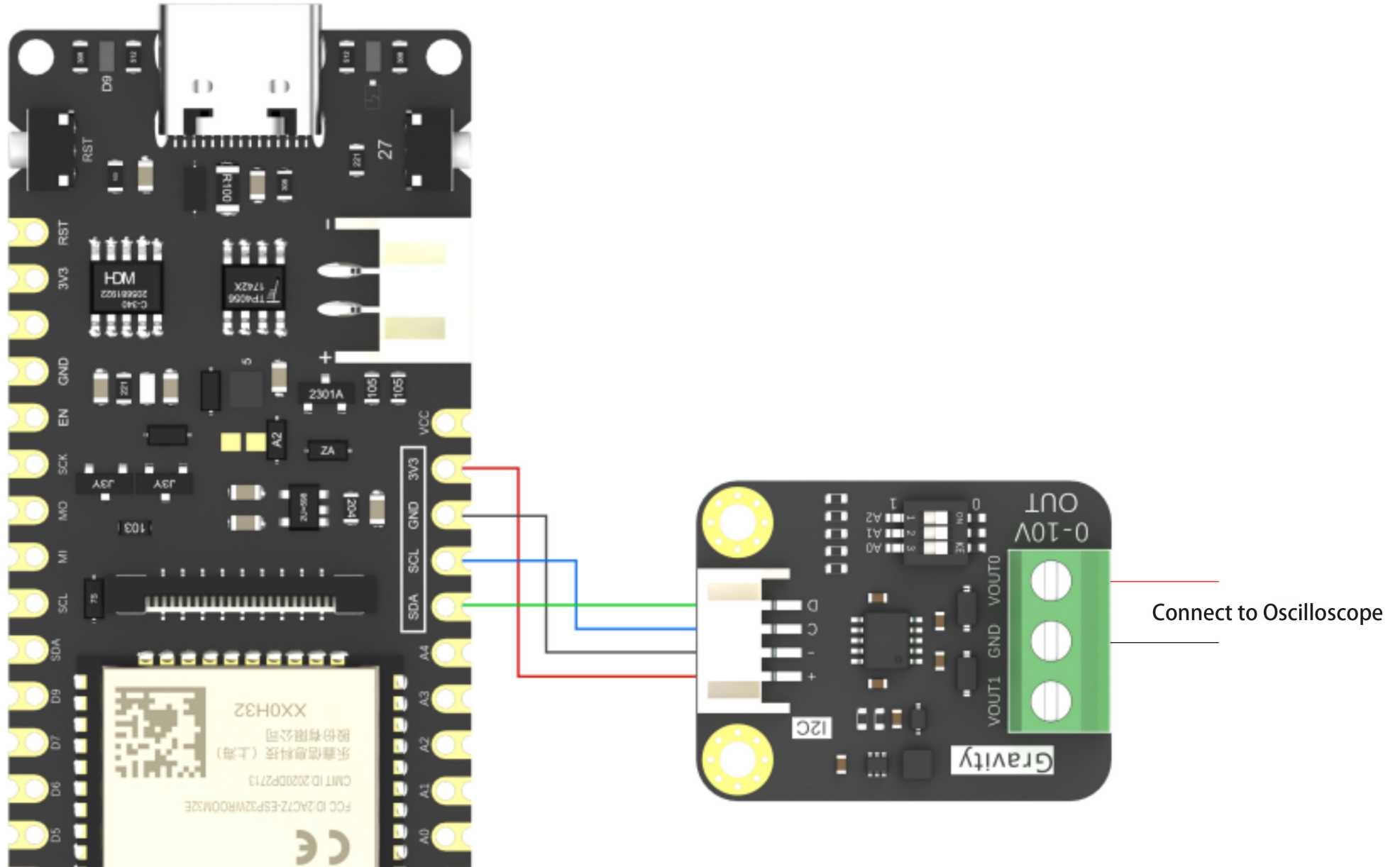
After downloading the code, the strip was lit, and at this time, the output voltage of channel 0 measured by a multimeter is 3.5V.

Output Sine Wave

This sample will show how to output a sine wave with a frequency of 10Hz and an amplitude of 5.12V. It should be noted that the waveform is generated by code simulation, which has a relatively large error, so it's just for learning. Connect the oscilloscope to the output and the generated sine waveform can be seen.

generated sine waveform can be seen.

Hardware Connetion





Wring Description

FireBeetle 2: 3V3 to DAC module: +

FireBeetle 2: GND to DAC module: -

FireBeetle 2: SCL to DAC module: SCL

FireBeetle 2: SDA to DAC module: SDA

Sample Code

This sample code demonstrates how to output a sine waveform with an amplitude of $\pm 2.5V$ and a frequency of 10Hz. API `dac.outputSin()` is used in the code. By setting different parameters of the API, the sine waveform with different voltage amplitude and frequency can be obtained, the maximum voltage is $\pm 5V$ and the frequency is 0-100Hz.

API Name: `outputSin(amp, freq, offset, channel)`

API Function: set the sine wave parameter

API Parameters:

`amp` sine wave amplitude, range: 0-5000 for 0-5V (within 10V), 0-2500 for 0-2.5V (within 5V)

`freq` sine wave frequency, range: 0-100Hz

offset DC offset of sine wave, range: 0-5000 for 0-5V (within 10V), 0-2500 for 0-2.5V (within 5V)

channel channel select, 0: channel 0, 1: channel 1, 2: all the channels

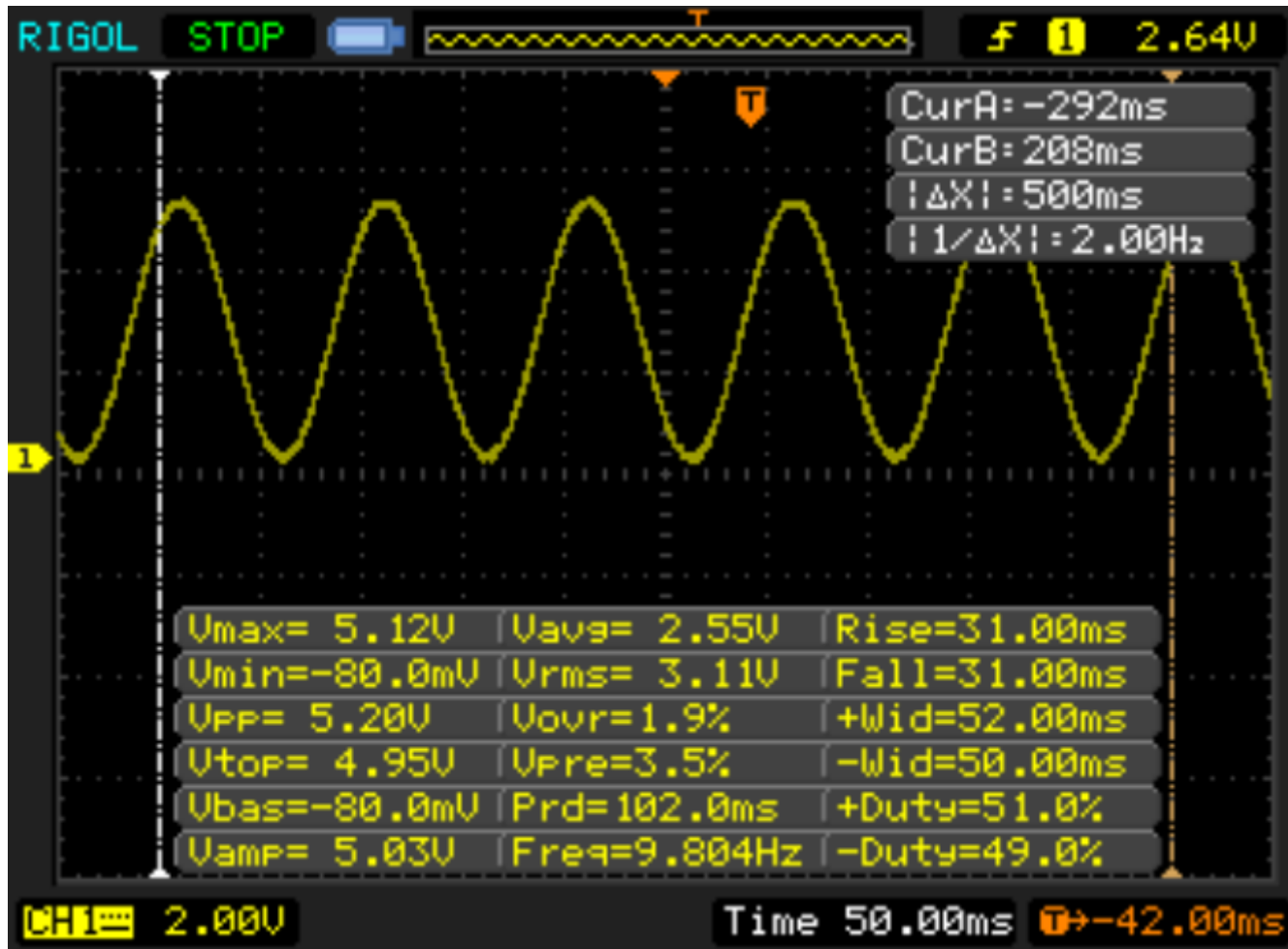
```
#include "DFRobot_GP8403.h"
DFRobot_GP8403 dac(&Wire,0x58);

void setup() {
  Serial.begin(115200);
  while(dac.begin()!=0){
    Serial.println("init error");
    delay(1000);
  }
  Serial.println("init succeed");
  //Set DAC output range
  dac.setDACOutRange(dac.e0outputRange5V);
}

void loop(){
  dac.outputSin(2500, 10, 2500, 0);//(voltage value, frequency (0~100Hz), voltage value, channel number)
}
```

Result

After downloading the code, a sine waveform with a frequency of 9.804Hz and an amplitude of $\pm 2.5V$ can be observed on the oscilloscope display.



API Function Library

```
/**
 * @fn begin
 * @brief Initialize the module
 */
uint8_t begin(void);

/**
 * @fn setDACOutRange
 * @brief Set DAC output range
 * @param range DAC output range, 5V or 10V
 * @return NONE
 */
void setDACOutRange(eOutPutRange range);

/**
 * @fn setDACOutVoltage
 * @brief Set output DAC voltage of different channels
 * @param data the DAC value corresponding to the voltage to be output
 * @param channel Output channel setting, 0: channel 0; 1: channel 1; 2: all the channels
 * @return NONE
 */
void setDACOutVoltage(uint16_t data,uint8_t channel);

/**
 * @brief Save the set voltage inside the chip
 */
void store(void);

/**
 * @brief Call the function to output sine wave
```



```
-----  
* @param amp Set sine wave amplitude Vp  
* @param freq Set sine wave frequency f  
* @param offset Set sine wave DC offset Voffset  
* @param channel Output channel: 0: channel 0; 1: channel 1; 2: all the channels  
  
*/  
void outputSin(uint16_t amp, uint16_t freq, uint16_t offset, uint8_t channel);  
/**  
* @brief Call the function to output triangle wave  
* @param amp Set triangle wave amplitude Vp  
* @param freq Set triangle wave frequency f  
* @param offset Set triangle wave DC offset Voffset  
* @param dutyCycle Set triangle (sawtooth) wave duty cycle  
* @param channel Output channel: 0: channel 0; 1: channel 1; 2: all the channels  
*/  
void outputTriangle(uint16_t amp, uint16_t freq, uint16_t offset, int8_t dutyCycle, uint8_t channel);  
/**  
* @brief Call the function to output square wave  
* @param amp Set square wave amplitude Vp  
* @param freq Set square wave frequency f  
* @param offset Set square wave DC offset Voffset  
* @param dutyCycle Set square wave duty cycle  
* @param channel Output channel: 0: channel 0; 1: channel 1; 2: all the channels  
*/  
void outputSquare(uint16_t amp, uint16_t freq, uint16_t offset, int8_t dutyCycle, uint8_t channel);
```

More Documents

DFR0971-Schematics.pdf (<https://dfimg.dfrobot.com/nobody/wiki/aa329c20d23132e901274bafe98e8c79.pdf>)

DFR0971_Dimensions.pdf (<https://dfimg.dfrobot.com/nobody/wiki/e35a5fe3b130c261cecfce89b103fd0d.pdf>)

DFR0971_3D File.rar (<https://dfimg.dfrobot.com/nobody/wiki/7909b96bf04413e7ffbd6937959f3536.rar>)

DFR0971_2D_CAD File.rar (<https://dfimg.dfrobot.com/nobody/wiki/1f3f07397315b570cea12d297f12b693.rar>)

DFR0971_GP8403 Datasheet.pdf (<https://dfimg.dfrobot.com/nobody/wiki/79daa2f3c729e78f0580b474fe33a5e3.pdf>)

FAQ

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

 Get **2-Channel I2C to 0-10V DAC Module** (<https://www.dfrobot.com/product-2613.html>) from DFRobot Store or **DFRobot Distributor**.
(<https://www.dfrobot.com/distributor>)

Turn to the Top