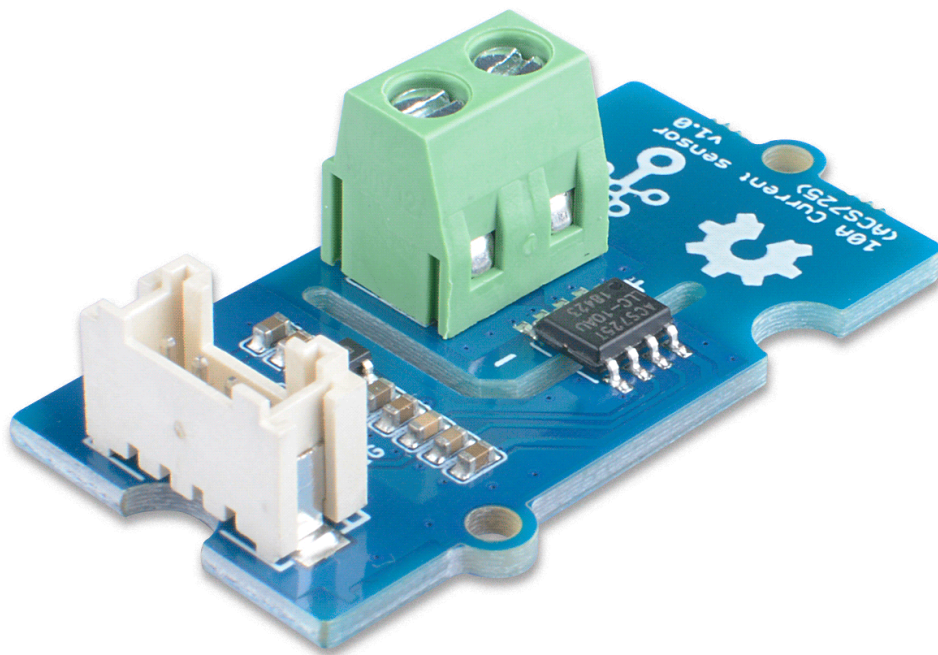


# Grove - 10A DC Current Sensor (ACS725)



The Grove - 10A DC Current Sensor (ACS725) is a high precision DC current sensor based on ACS725. The Allegro™ ACS725 current sensor IC is an economical and precise solution DC current sensing in industrial, automotive, commercial, and communications systems.

The Grove - 10A DC Current Sensor (ACS725) can measure the DC current up to 10A and has a base sensitivity of 264mV/A. This sensor do not support AC current, if you want to measure the AC load please check the:

[Grove - ±5A DC/AC Current Sensor \(ACS70331\)](https://www.seeedstudio.com/Grove-5A-DC-AC-Current-Sensor-ACS70331-p-2928.html)

[<https://www.seeedstudio.com/Grove-5A-DC-AC-Current-Sensor-ACS70331-p-2928.html>]



[<https://www.seeedstudio.com/Grove-10A-DC-Current-Sensor-ACS725-p-2927.html>]

## Feature

- High-bandwidth 120 kHz analog output for faster response times in control applications
- Industry-leading noise performance with greatly improved bandwidth through proprietary amplifier and filter design techniques
- 1.2 mΩ primary conductor resistance for low power loss and high inrush current withstand capability
- Differential Hall sensing rejects common-mode fields
- Analog output

## Specification

Parameter	Value
Supply voltage	3.3V / 5V
Operating ambient temperature	-40 – 150°C
Storage temperature	- 65°C – 165°C
Working Voltage	<400V
Current sensing range	0 – 10A
Sensitivity	264mV/A(Typ.)
Output interface	Analog
Input interface	Screw terminal

## Working Principle

There are two types of current sensing: direct and indirect. Classification is mainly based on the technology used to measure current.

### **Direct sensing:**

- Ohm's Law

### **Indirect sensing:**

- Faraday's Law of Induction
- Magnetic field sensors

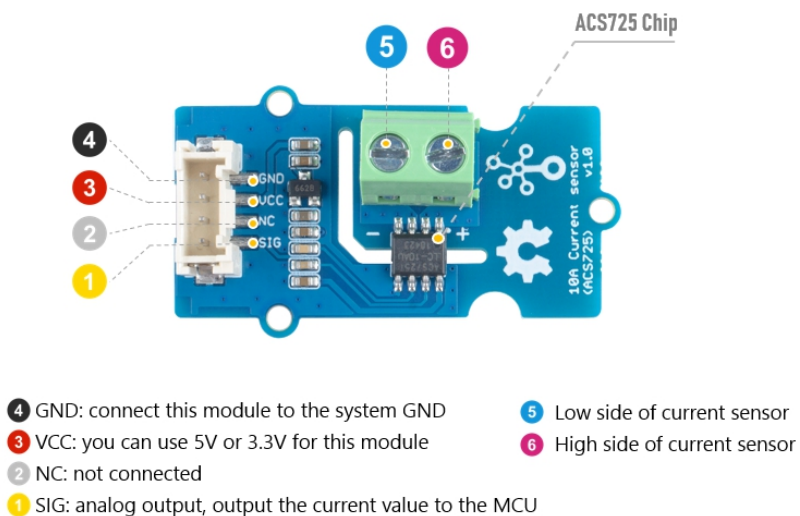
- Faraday Effect

The Grove - 10A DC Current Sensor (ACS725) uses magnetic field sensors technology. And there are three kinds of Magnetic field sensors technology:

- Hall effect
- Flux gate sensors
- Magneto-resistive current sensor

The Grove - 10A DC Current Sensor (ACS725) is based on the Hall principle, differential Hall sensing rejects common-mode.

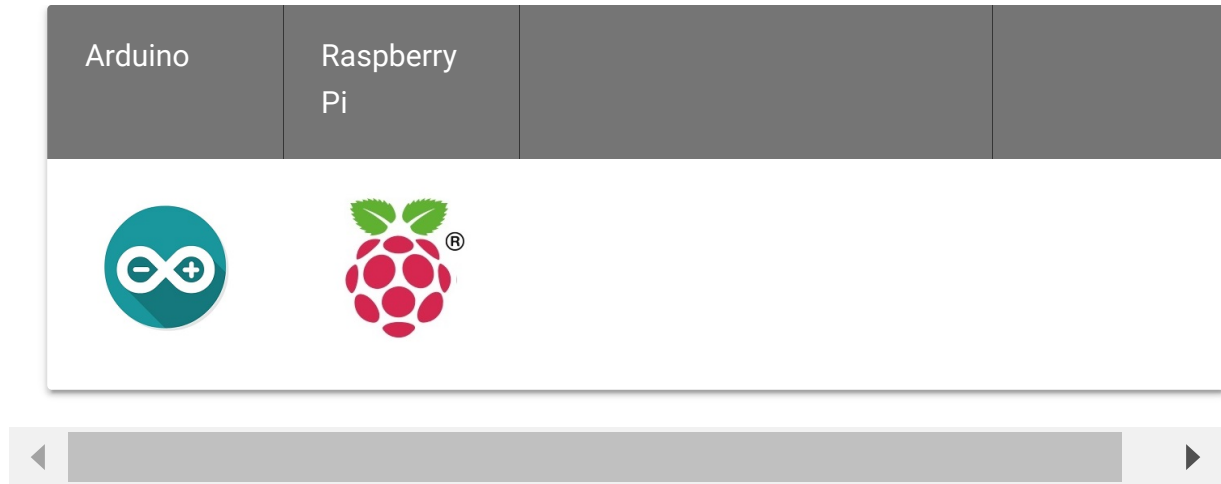
## Hardware Overview



[[https://files.seeedstudio.com/wiki/Grove-10A\\_Current\\_Sensor-ACS725/img/pinout.jpg](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/img/pinout.jpg)]

**Figure 3. Pinout**

## Platforms Supported



## Getting Started



### **Danger**

The human body is forbidden to touch the module during the test, otherwise there is danger of electric shock.

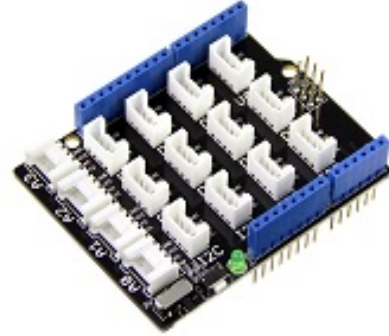
## Play With Arduino

### **Materials required**

Seeeduino V4.2



Base Shield



Get ONE Now

[<https://www.seeedstudio.com/Seeeduino-V4.2-p-2517.html>]

Get ONE Now

[<https://www.seeedstudio.com/Base-Shield-V2-p-1378.html>]

In addition, you can consider our new [Seeeduino Lotus M0+](https://www.seeedstudio.com/Seeeduino-Lotus-Cortex-M0-p-2896.html) [<https://www.seeedstudio.com/Seeeduino-Lotus-Cortex-M0-p-2896.html>], which is equivalent to the combination of Seeeduino V4.2 and Baseshield.



#### Note

**1** Please plug the USB cable gently, otherwise you may damage the port. Please use the USB cable with 4 wires inside, the 2 wires cable can't transfer data. If you are not sure about the wire you have, you can click [here](https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html) [<https://www.seeedstudio.com/Micro-USB-Cable-48cm-p-1475.html>] to buy

**2** Each Grove module comes with a Grove cable when you buy. In case you lose the Grove cable, you can click [here](https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html) [<https://www.seeedstudio.com/Grove-Universal-4-Pin-Buckled-20cm-Cable-%285-PCs-pack%29-p-936.html>] to buy.

## Hardware Connection

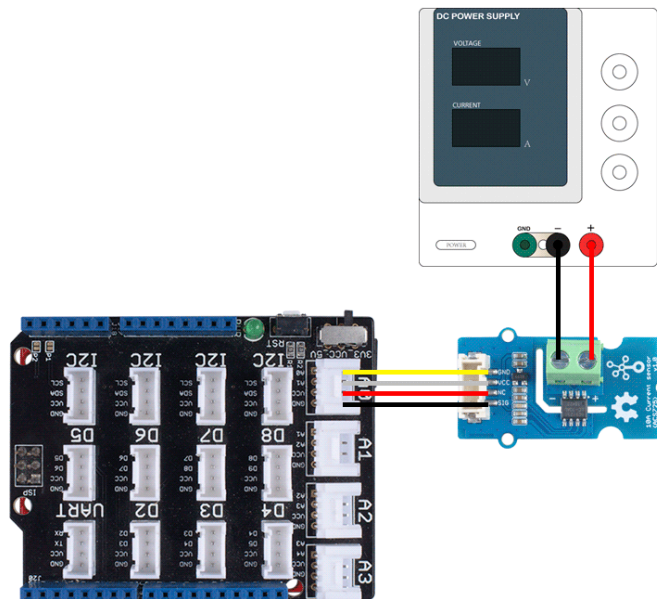
- **Step 1.** Connect the Grove - 10A DC Current Sensor(ACS725) to the **A0** port of the Base Shield.
- **Step 2.** Connect the positive and negative poles of the circuit to be tested to the corresponding positive and negative poles of the screw terminal.



### Tip

If you reverse the positive and negative poles, the reading will be reversed. This sensor need calibration before use, so please do not power on the circuit first.

- **Step 3.** Plug Grove - Base Shield into Seeeduino.
- **Step 4.** Connect Seeeduino to PC via a USB cable.



[[https://files.seeedstudio.com/wiki/Grove-10A\\_Current\\_Sensor-ACS725/img/101020616-connect.png](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/img/101020616-connect.png)]

**Figure 4.** We use the DC Power Supply in this demo, please set the current to 0A or do not power on it at first

## Software



### Attention

If this is the first time you work with Arduino, we strongly recommend you to see [Getting Started with Arduino](https://wiki.seeedstudio.com/Getting_Started_with_Arduino/) [https://wiki.seeedstudio.com/Getting\_Started\_with\_Arduino/] before the start.

- **Step 1.** Download the [Grove Current Sensor](https://github.com/Seeed-Studio/Grove_Current_Sensor) [https://github.com/Seeed-Studio/Grove\_Current\_Sensor] Library from Github.
- **Step 2.** In the /example/ folder, you can find the demo code. Here we take the **Grove\_10A\_Current\_Sensor** for instance. Just click the [Grove\\_10A\\_Current\\_Sensor.ino](https://github.com/Seeed-Studio/Grove_Current_Sensor/blob/master/examples/Grove_10A_Current_Sensor/Grove_10A_Current_Sensor.ino) [https://github.com/Seeed-Studio/Grove\_Current\_Sensor/blob/master/examples/Grove\_10A\_Current\_Sensor/Grove\_10A\_Current\_Sensor.ino] to open the demo. Or you can copy the following code:

```
1  #ifdef ARDUINO_SAMD_VARIANT_COMPLIANCE
2      #define RefVal 3.3
3      #define SERIAL SerialUSB
4  #else
5      #define RefVal 5.0
6      #define SERIAL Serial
7  #endif
8  //An OLED Display is required here
9  //use pin A0
10 #define Pin A0
11
12 // Take the average of 500 times
```

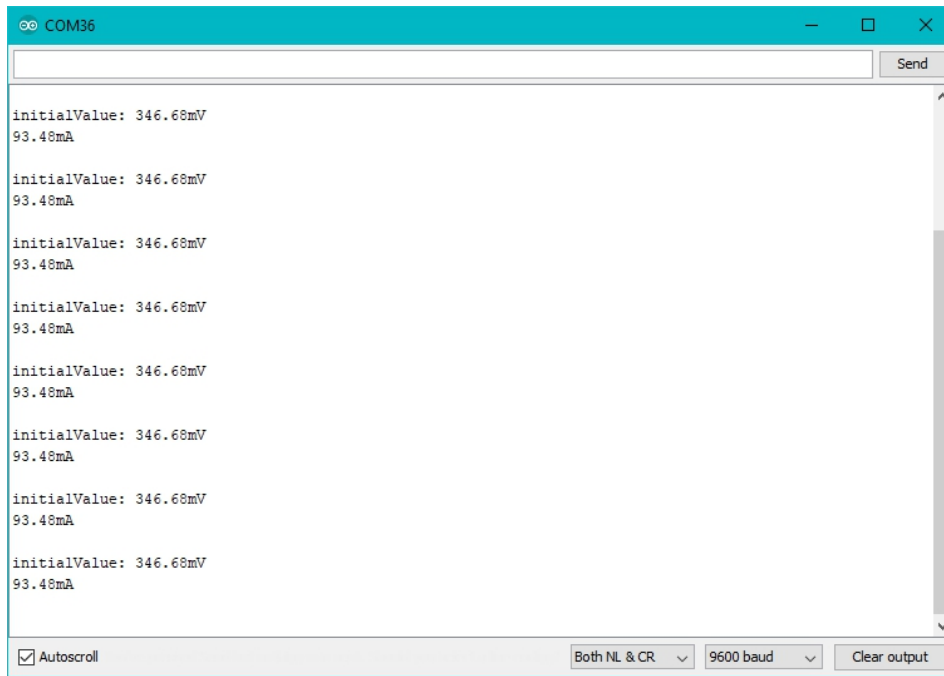




```
13  const int averageValue = 500;
14
15  long int sensorValue = 0;
16  float sensitivity = 1000.0 / 264.0; //1000mA per 264mV
17
18
19  float Vref = 322; //Vref is zero drift value, you need
20  void setup()
21  {
22    SERIAL.begin(9600);
23  }
24
25  void loop()
26  {
27    // Read the value 10 times:
28    for (int i = 0; i < averageValue; i++)
29    {
30      sensorValue += analogRead(Pin);
31
32      // wait 2 milliseconds before the next loop
33      delay(2);
34    }
35
36
37    sensorValue = sensorValue / averageValue;
38
39
40    // The on-board ADC is 10-bits
41    // Different power supply will lead to different refer
42    // example: 2^10 = 1024 -> 5V / 1024 ~= 4.88mV
43    //          unitValue= 5.0 / 1024.0*1000 ;
44    float unitValue= RefVal / 1024.0*1000 ;
45    float voltage = unitValue * sensorValue;
46
47    //When no Load,Vref=initialValue
48    SERIAL.print("initialValue: ");
49    SERIAL.print(voltage);
50    SERIAL.println("mV");
51
52    // Calculate the corresponding current
53    float current = (voltage - Vref) * sensitivity;
```

```
54
55 // Print display voltage (mV)
56 // This voltage is the pin voltage corresponding to the
57 /*
58 voltage = unitValue * sensorValue - Vref;
59 SERIAL.print(voltage);
60 SERIAL.println("mV");
61 */
62
63 // Print display current (mA)
64 SERIAL.print(current);
65 SERIAL.println("mA");
66
67 SERIAL.print("\n");
68
69 // Reset the sensorValue for the next reading
70 sensorValue = 0;
71 // Read it once per second
72 delay(1000);
73 }
```

- **Step 3.** Upload the demo. If you do not know how to upload the code, please check [How to upload code](https://wiki.seeedstudio.com/Upload_Code/) [https://wiki.seeedstudio.com/Upload\_Code/].
- **Step 4.** Open the **Serial Monitor** of Arduino IDE by click **Tool->Serial Monitor**. Or tap the **Ctrl + Shift + M** key at the same time. Set the baud rate to **9600**.
- **Step 5. Calibration**  
When there is no current flowing, the sensor will still have a small output value. We call this value **zero offset**.



[[https://files.seeedstudio.com/wiki/Grove-10A\\_Current\\_Sensor-ACS725/img/ca1.jpg](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/img/ca1.jpg)]

**Figure 5.** *The zero offset of this board is 346.68mV ,  
Converted into current is 93.48mA*

Due to the presence of zero offset, the sensor will also have a reading when there is no current. So we set a parameter **Vref** to fix it, you can find it in the code block above.

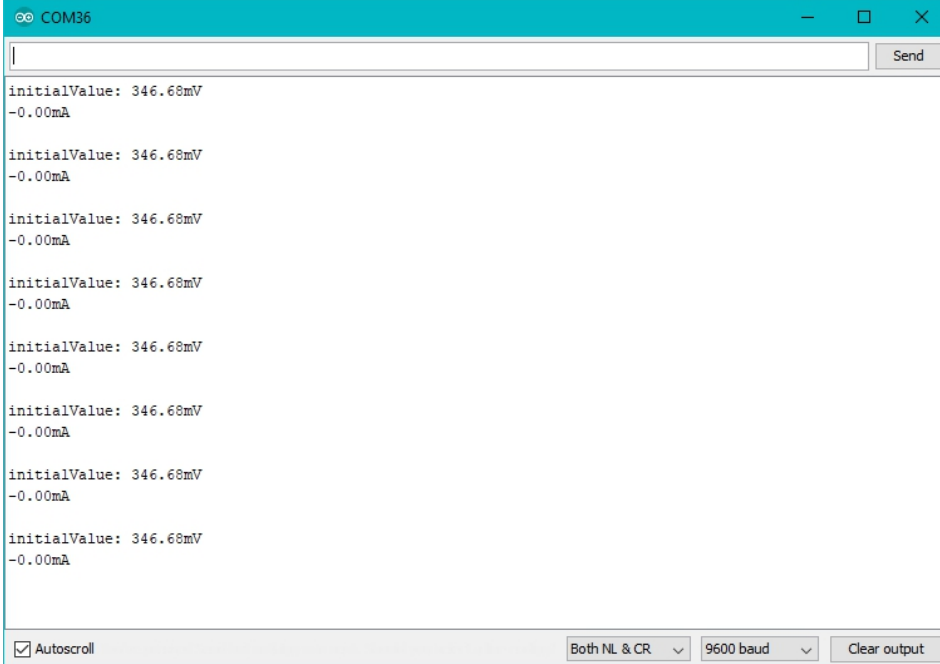
Line 19:

```
1 float Vref = 322;  
2 //Vref is zero drift value, you need to change this value
```

In the demo code, we set the Vref to 322, however, the zero offset value varies from board to board. As you know, the board we use in this demo is 346.68. So let's modify the Line 21:

```
float Vref = 346.68;
```

Then save the code and upload the code again, follow the Step 2. and Step 3. Now let's see:

A screenshot of a serial terminal window titled "COM36". The window displays a series of eight lines of text, each consisting of two lines: "initialValue: 346.68mV" followed by "-0.00mA". At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, a dropdown menu set to "Both NL & CR", a dropdown menu set to "9600 baud", and a "Clear output" button. A "Send" button is located at the top right of the terminal area.

```
initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA

initialValue: 346.68mV
-0.00mA
```

[[https://files.seeedstudio.com/wiki/Grove-10A\\_Current\\_Sensor-ACS725/img/ca2.jpg](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/img/ca2.jpg)]

**Figure 6.** Now the current zero offset turns to 0mA


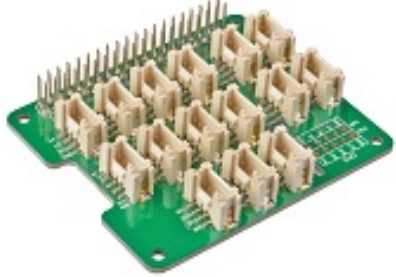
When the current output becomes to 0mA or a small value, you have completed the calibration.

- **Step 6.** Now it's all yours, you can power up the current. Please feel free to use it, remember this is a 10A DC Current Sensor, current cannot exceed 10A!

If you want to know the calculation formula of the result, please refer to the [FAQ Q1](#) [#faq]

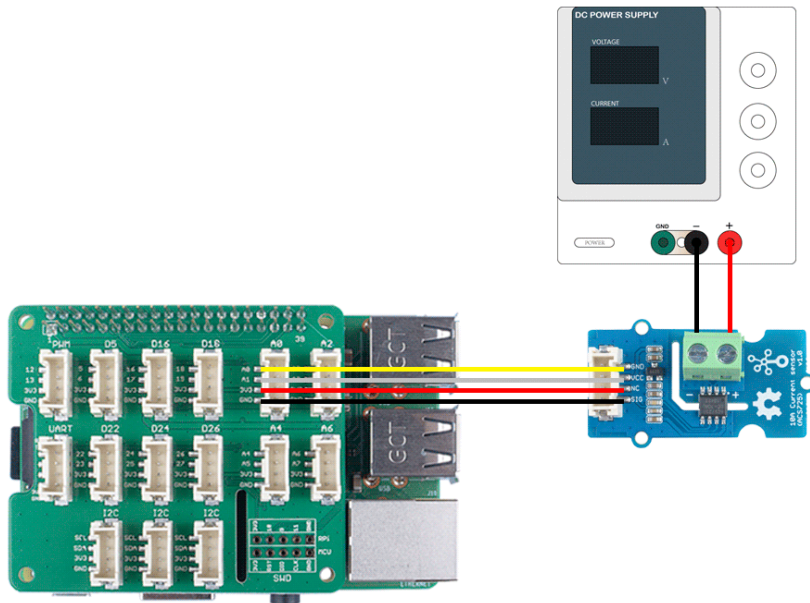
## Play with Raspberry

## Materials required

Raspberry pi	Grove Base Hat for RasPi
	
<p><a href="https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html">Get ONE Now</a> [<a href="https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html">https://www.seeedstudio.com/Raspberry-Pi-3-Model-B-p-2625.html</a>]</p>	<p><a href="https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html">Get ONE Now</a> [<a href="https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html">https://www.seeedstudio.com/Grove-Base-Hat-for-Raspberry-Pi-p-3186.html</a>]</p>

## Hardware Connection

- **Step 1.** Plug the Grove Base Hat into Raspberry Pi.
- **Step 2.** Connect the Grove - 10A DC Current Sensor(ACS70331) to port **A0** of the Base Hat.
- **Step 3.** Connect the positive and negative poles of the circuit to be tested to the corresponding positive and negative poles of the screw terminal.



[[https://files.seeedstudio.com/wiki/Grove-10A\\_Current\\_Sensor-ACS725/img/101020616-connect2.png](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/img/101020616-connect2.png)]

**Figure 7.** We use the DC Power Supply in this demo, please set the current to 0A or do not power on it at first



#### Tip

If you reverse the positive and negative poles, the reading will be reversed. This sensor need calibration before use, so please do not power on the circuit first.

- **Step 4.** Power the Raspberry Pi via the Micro-USB cable.



#### Attention

You can power the Raspberry Pi by computer USB port or DC adapter, however, if you are using the Raspberry pi 3B+, we strongly recommend you to power it by DC adapter, if you use the USB port of the PC, you may damage the Raspberry Pi 3B+.

## Software

- **Step 1.** Follow [Setting Software](#) [[https://wiki.seeedstudio.com/Grove\\_Base\\_Hat\\_for\\_Raspberry\\_Pi/#installation](https://wiki.seeedstudio.com/Grove_Base_Hat_for_Raspberry_Pi/#installation)] to configure the development environment.
- **Step 2.** Download the source file by cloning the [grove.py](#) [<https://github.com/Seeed-Studio/grove.py>] library.

```
1 cd ~
2 git clone https://github.com/Seeed-Studio/grove.py
```

- **Step 3.** Execute following commands to run the code.

```
1 cd grove.py/grove # to enter the demo file folder
2 python grove_current_sensor.py 0 10A # to run the demo
```

Then the terminal will output as following:

```
1 pi@raspberrypi:~/grove.py/grove $ python grove_current_s
2 pin_voltage(mV):
3 324
4 current(mA):
5 8.0
6 ()
7 pin_voltage(mV):
8 324
9 current(mA):
10 8.0
11 ()
12 pin_voltage(mV):
13 323
14 current(mA):
15 4.0
16 ()
```

```
17 pin_voltage(mV):
18 324
19 current(mA):
20 8.0
21 ()
22 pin_voltage(mV):
23 324
24 current(mA):
25 8.0
26 ()
27 pin_voltage(mV):
28 324
29 current(mA):
30 8.0
31 ()
32 ^CTraceback (most recent call last):
33   File "grove_current_sensor.py", line 200, in <module>
34     main()
35   File "grove_current_sensor.py", line 185, in main
36     time.sleep(1)
37 KeyboardInterrupt
```

Tap `Ctrl + C` to quit.



#### Note

Please note the second command, There are two parameters after the file name:

- **0** means the sensor is connected to port A0. If you connect the sensor to port A2, then you need to change this parameter to 2. This parameter has a range of 0-7, but if you use the Grove base hat, you can only use 0/2/4/6 because of the physical limitations of the interface.
- **10A** means the current sensor type is 10A DC



Sensor	Current type	Parameter Value
Grove - 2.5A DC Current Sensor(ACS70331)	DC	2.5A
Grove - $\pm$ 5A DC/AC Current Sensor (ACS70331)	DC	5A_DC
	AC	5A_AC
Grove - 10A DC Current Sensor (ACS725)	DC	10A

*This series has three current sensors, the parameter list is as above*

!!! Note Please note that the DC current sensor of 10A will have a large error when measuring a small range, so it is recommended that you provide a current of more than 200mA for testing. In addition, the measurement environment will affect the accuracy, such as the supply voltage ripple to be as small as possible.

- **Step 4 Calibration.**

When there is no current flowing, the sensor will still have a small output value. We call this value zero offset. As you can see, in the step 3, the zero offset of this board is 324mV, converted into current is 8mA.

Due to the presence of zero offset, the sensor will also have a reading when there is no current. So we set a parameter **Vref** to fix it, you can find it in the **python grove\_current\_sensor.py**. For the Grove - 10A DC Current Sensor (ACS725), we set the **Vref** to

322 by default, however the zero offset varies from board to board. That's why we need to do the calibration first.

Check the python code below.

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  #
4  # The MIT License (MIT)
5  # Copyright (C) 2018 Seeed Technology Co.,Ltd.
6  #
7  # This is the library for Grove Base Hat
8  # which used to connect grove sensors for Raspberry Pi.
9  '''
10 This is the code for
11     - `Grove - 2.5A DC current sensor <https://www.see
12     - `Grove - 5A AC/DC current sensor <https://www.see
13     - `Grove - 10A current sensor      <https://www.see
14 Examples:
15     .. code-block:: python
16         import time
17         from grove_current_sensor import Current
18         pin = 0
19         sensor_type = "2.5A"
20         #if use 10A current sensor input: pin = 0 , sen
21         if (sensor_type == "2.5A"):
22             sensitivity = 1000.0 / 800.0
23             Vref = 260
24         if (sensor_type == "5A_DC"):
25             sensitivity = 1000.0 / 200.0
26             Vref = 1498
27         if (sensor_type == "5A_AC"):
28             sensitivity = 1000.0 / 200.0
29             Vref = 1498
30         if (sensor_type == "10A"):
31             sensitivity = 1000.0 / 264.0
32             Vref = 322
33         averageValue = 500
34         ADC = Current()
35         while True:
```



```
36         if(sensor_type == "5A_AC"):
37             pin_voltage = ADC.get_nchan_vol_milli_d
38             current = ADC.get_nchan_AC_current_data
39         else:
40             temp = ADC.get_nchan_current_data(pin,s
41             current = temp[0]
42             pin_voltage = temp[1]
43
44             current = round(current)
45             print("pin_voltage(mV):")
46             print(pin_voltage)
47             print("current(mA):")
48             print(current)
49             print()
50             time.sleep(1)
51
52     '''
53
54     import sys
55     import time
56     from grove.i2c import Bus
57
58     ADC_DEFAULT_IIC_ADDR = 0X04
59
60     ADC_CHAN_NUM = 8
61
62     REG_RAW_DATA_START = 0X10
63     REG_VOL_START = 0X20
64     REG_RTO_START = 0X30
65
66     REG_SET_ADDR = 0XC0
67
68     __all__ = ['Current', 'Bus']
69
70     class Current():
71         '''
72         Grove Current Sensor class
73         '''
74
75         def __init__(self, bus_num=1, addr=ADC_DEFAULT_IIC_ADDR):
76             '''
```

```
77     Init iic.
78     Args:
79         bus_num(int): the bus number;
80         addr(int): iic address;
81     '''
82     self.bus = Bus(bus_num)
83     self.addr = addr
84
85     def get_nchan_vol_milli_data(self,n,averageValue):
86         '''
87         Get n chanel data with unit mV.
88         :param int n: the adc pin.
89         :param int averageValue: Average acquisition fr
90     Returns:
91         int: voltage value
92     '''
93     val = 0
94     for i in range(averageValue):
95         data = self.bus.read_i2c_block_data(self.ad
96         val += data[1]<<8|data[0]
97     val = val / averageValue
98     return val
99
100    def get_nchan_current_data(self,n,sensitivity,Vref,
101    '''
102    2.5A/5A DC/10A currrent sensor get n chanel dat
103    :param int n: the adc pin.
104    :param float sensitivity: The coefficient by wh
105    :param int Vref: Initial voltage at no load.
106    :param int averageValue: Average acquisition fr
107    Returns:
108    int: current value
109    '''
110    val = 0
111    for i in range(averageValue):
112        data = self.bus.read_i2c_block_data(self.ad
113        val += data[1]<<8|data[0]
114    val = val / averageValue
115    currentVal = (val - Vref) * sensitivity
116    return currentVal,val
117
```

```
118     def get_nchan_AC_current_data(self,n,sensitivity,Vref
119         '''
120         5A current sensor AC output and get n chanel da
121         :param int n: the adc pin.
122         :param float sensitivity: The coefficient by wh
123         :param int Vref: Initial voltage at no load.
124         :param int averageValue: Average acquisition fr
125         Returns:
126             int: current value
127         '''
128         sensorValue = 0
129         for i in range(averageValue):
130             data=self.bus.read_i2c_block_data(self.addr
131             val=data[1]<<8|data[0]
132             if(val > sensorValue):
133                 sensorValue=val
134                 time.sleep(0.00004)
135             currentVal = ((sensorValue - Vref) * sensitivit
136             return currentVal
137
138 ADC = Current()
139 def main():
140     if(len(sys.argv) == 3):
141
142         pin = int(sys.argv[1])
143         sensor_type = sys.argv[2]
144         if (pin < 8 and (sensor_type == "2.5A" or senso
145             if (sensor_type == "2.5A"):
146                 sensitivity = 1000.0 / 800.0
147                 Vref = 260
148             if (sensor_type == "5A_DC"):
149                 sensitivity = 1000.0 / 200.0
150                 Vref = 1498
151             if (sensor_type == "5A_AC"):
152                 sensitivity = 1000.0 / 200.0
153                 Vref = 1498
154             if (sensor_type == "10A"):
155                 sensitivity = 1000.0 / 264.0
156                 Vref = 322
157                 averageValue = 500
158
```

```

159         while True:
160
161             if(sensor_type == "5A_AC"):
162                 pin_voltage = ADC.get_nchan_vol_mil
163                 current = ADC.get_nchan_AC_current_
164             else:
165                 temp = ADC.get_nchan_current_data(p
166                 current = temp[0]
167                 pin_voltage = temp[1]
168
169                 current = round(current)
170                 print("pin_voltage(mV):")
171                 print(pin_voltage)
172                 print("current(mA):")
173                 print(current)
174                 print()
175                 time.sleep(1)
176
177         else:
178             print("parameter input error!")
179             print("Please enter parameters for example:
180             print("parameter1: 0-7")
181             print("parameter2: 2.5A/5A_DC/5A_AC/10A")
182
183     else:
184         print("Please enter parameters for example: pyt
185         print("parameter1: 0-7")
186         print("parameter2: 2.5A/5A_DC/5A_AC/10A")
187
188
189 if __name__ == '__main__':
190     main()

```

You can modify the **Vref** at line 156 of the code block above:

```

1         if (pin < 8 and (sensor_type == "2.5A" or sensor_
2             if (sensor_type == "2.5A"):
3                 sensitivity = 1000.0 / 800.0
4                 Vref = 260

```

```

5         if (sensor_type == "5A_DC"):
6             sensitivity = 1000.0 / 200.0
7             Vref = 1498
8         if (sensor_type == "5A_AC"):
9             sensitivity = 1000.0 / 200.0
10            Vref = 1498
11        if (sensor_type == "10A"):
12            sensitivity = 1000.0 / 264.0
13            Vref = 322
14        averageValue = 500

```

As you can see, for the 10A Current Sensor the default **Vref** is 322, and in the **Step 3**, we can find it when there is no current the zero offset value is 324mV. So let's change it into 324.

```

1         if (sensor_type == "10A"):
2             sensitivity = 1000.0 / 264.0
3             Vref = 324

```

Now, let's run this demo again.

```

1 pi@raspberrypi:~/grove.py/grove $ python grove_current_s
2 pin_voltage(mV):
3 324
4 current(mA):
5 0.0
6 ()
7 pin_voltage(mV):
8 325
9 current(mA):
10 4.0
11 ()
12 pin_voltage(mV):
13 324
14 current(mA):
15 0.0
16 ()

```

```

17 pin_voltage(mV):
18 323
19 current(mA):
20 -4.0
21 ()
22 pin_voltage(mV):
23 324
24 current(mA):
25 0.0

```

Well, better than before, now you can measure the current more accurately 😊

## FAQ

**Q1#** What's the current calculation formula?

**A1:** If you think the **principle part** [#working-principle] is very complicated, let's put it in a easy way. The current in the circuit to be tested excites the magnetic field, which causes the resistance value of the GMR elements change. And the resistance change in the bridge causes a change in the voltage at the output of the chip. We call the voltage output as **V<sub>IOUT</sub>**.

$$V_{\{IOUT\}} = Sens \times I_P + V_{\{IOUT(Q)\}}$$

**Sens:** Sens is the coefficient that converts the current into an output voltage. For this module it is 264mA/V.

**I<sub>p</sub>:** I<sub>p</sub> is the current value in the circuit to be tested, Unit mA.

**V<sub>IOUT(Q)</sub>:** V<sub>IOUT(Q)</sub> is the voltage output when the I<sub>p</sub> is 0mA(which means there is no current in the circuit to be tested), Unit mV.

Here comes the current value:



$$I_P = \{V_{\{IOUT\}} - V_{\{IOUT(Q)\}} \over Sens\}$$

Now, Let's review the figure 5, we will explain why the current value of the output is not 0 when the actual current value in the circuit to be tested is 0. As you can see in the figure 5, the **initialValue** is 346.68mV, which is the **V<sub>IOUT</sub>**; the current is 93.48mA, which is the **I<sub>p</sub>**. As for the **V<sub>IOUT(Q)</sub>**, it is the **Vref** we set in the code. In figure 5, it is 265. And the **Sens** is 264mA/V, which is 264mA/1000mV. Now, just do some math:

$$\{346.68mV - 265mV \over 264mA/1000mV\} = 93.48mA$$

So, in the figure 6, when we set the **Vref** to 346.68, the **I<sub>p</sub>** turns to 0mA.

## Schematic Online Viewer



## Resources

- **[ZIP]** [Grove - 10A DC Current Sensor \(ACS725\) Schematic file](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/res/10A%20Current%20Sensor%20(ACS725).zip)  
[https://files.seeedstudio.com/wiki/Grove-10A\_Current\_Sensor-ACS725/res/10A%20Current%20Sensor%20(ACS725).zip]
- **[PDF]** [ACS725 Datasheet](https://files.seeedstudio.com/wiki/Grove-10A_Current_Sensor-ACS725/res/ACS725.pdf)  
[https://files.seeedstudio.com/wiki/Grove-10A\_Current\_Sensor-ACS725/res/ACS725.pdf]

# Tech Support

Please submit any technical issue into our [forum](#)

[<https://forum.seeedstudio.com/>]



[[https://www.seeedstudio.com/act-4.html?](https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts)

[utm\\_source=wiki&utm\\_medium=wikibanner&utm\\_campaign=newproducts](https://www.seeedstudio.com/act-4.html?utm_source=wiki&utm_medium=wikibanner&utm_campaign=newproducts)]  
oducts]