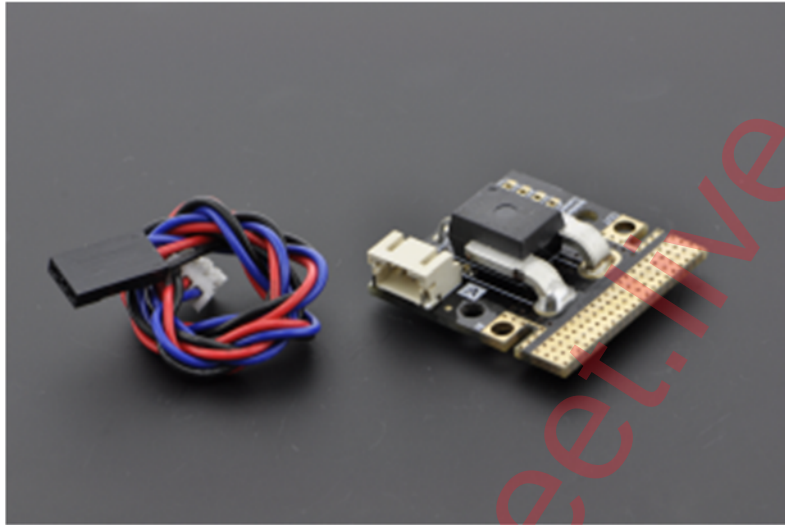


50A Current Sensor(SKU:SEN0098)



Contents

- 1 Introduction
- 2 Specifications
- 3 PinOut
- 4 Tutorial
 - 4.1 Connection Diagram
 - 4.2 DC load
 - 4.2.1 Sketch
 - 4.2.2 Result
 - 4.3 AC Load
 - 4.3.1 Sketch
 - 4.4 Result of AC current
- 5 Trouble Shooting

Introduction

This is a breakout board for the fully integrated Hall Effect based linear ACS758 current sensor. The sensor gives precise current measurement for both AC and DC signals. The thickness of the copper conductor allows survival of the device at high overcurrent conditions. The ACS758 outputs an analog voltage output signal that varies linearly with sensed current. This current sensor enables you to monitor currents in your project every second for energy saving or circuit protecting purposes.

Specifications

- Operating Voltage(analog): 5V
- Peak Measuring Voltage:3000V(AC),500V(DC)
- Current Measuring Rang:-50~50A
- Sensitivity:40 mV/A
- Operating Temperature: -40~150°C
- Dimension:34x34mm

PinOut

old version



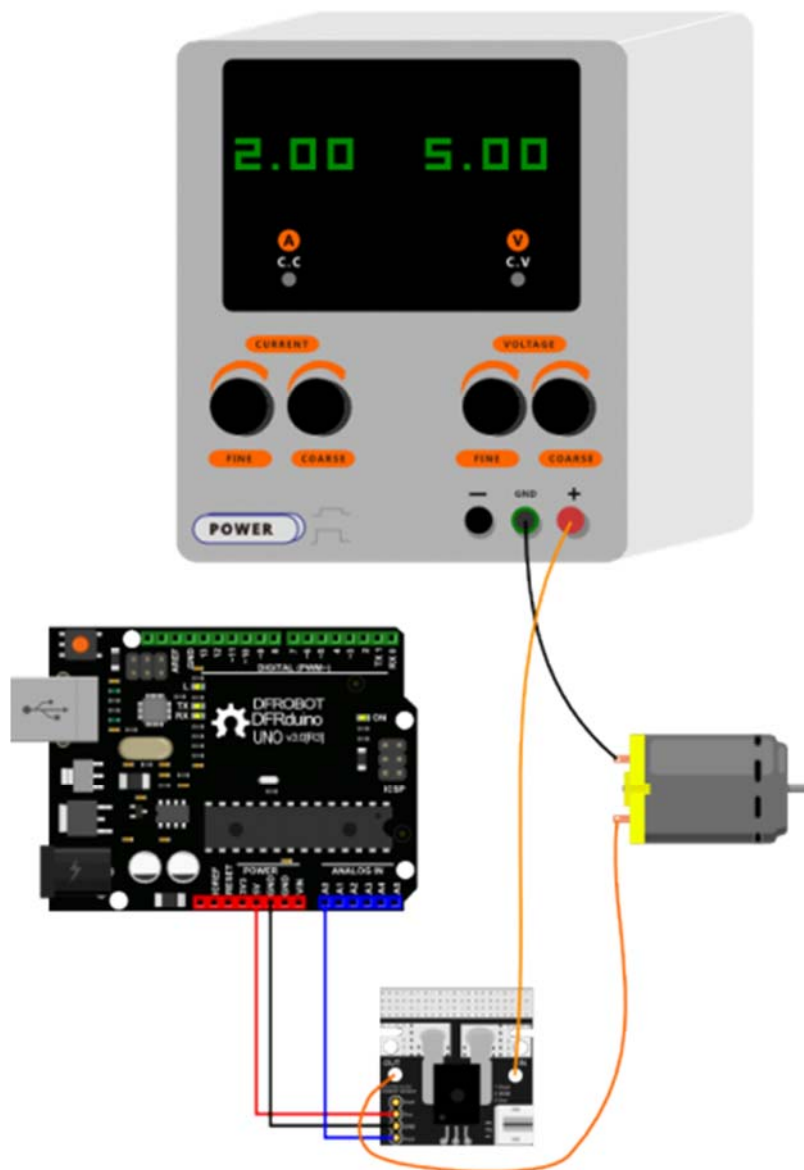
new version



Tutorial

This tutorial is going to test DC and AC current. The result of AC current is its effective value, i.e. the same as an Ammeter reading.

Connection Diagram



DANGER: This diagram can also be applied to AC current connection. But notice that the module should be **cascaded** on **one** electric wire (+/ -). But if you connect it to two electric wires in parallel, you will short circuit and lead to danger.

DC load

Sketch

```
/*
50A Current Sensor(DC)(SKU:SEN0098) Sample Code
This code shows you how to get raw datas from the sensor through Arduino and
convert the raw datas to the value of the current according to the datasheet;

Smoothing algorithm (http://www.arduino.cc/en/Tutorial/Smoothing) is used to
make the outputting current value more reliable;

Created 27 December 2011 @Barry Machine @www.dfrobot.com
*/

const int numReadings = 30;
float readings[numReadings];      // the readings from the analog input
int index = 0;                    // the index of the current reading
float total = 0;                  // the running total
float average = 0;                // the average

float currentValue = 0;

void setup()
{
  Serial.begin(115200);
  for (int thisReading = 0; thisReading < numReadings; thisReading++)
    readings[thisReading] = 0;
}

void loop()
{
  total= total - readings[index];
  readings[index] = analogRead(0); //Raw data reading
  //Data processing:510-raw data from analogRead when the input is 0;
  // 5-5v; the first 0.04-0.04V/A(sensitivity); the second 0.04-offset val;
```

```

    readings[index] = (readings[index]-512)*5/1024/0.04-0.04;

    total= total + readings[index];
    index = index + 1;
    if (index >= numReadings)
        index = 0;
        average = total/numReadings;    //Smoothing algorithm (http://www.arduino.
cc/en/Tutorial/Smoothing)
        currentValue= average;
        Serial.println(currentValue);
        delay(10);

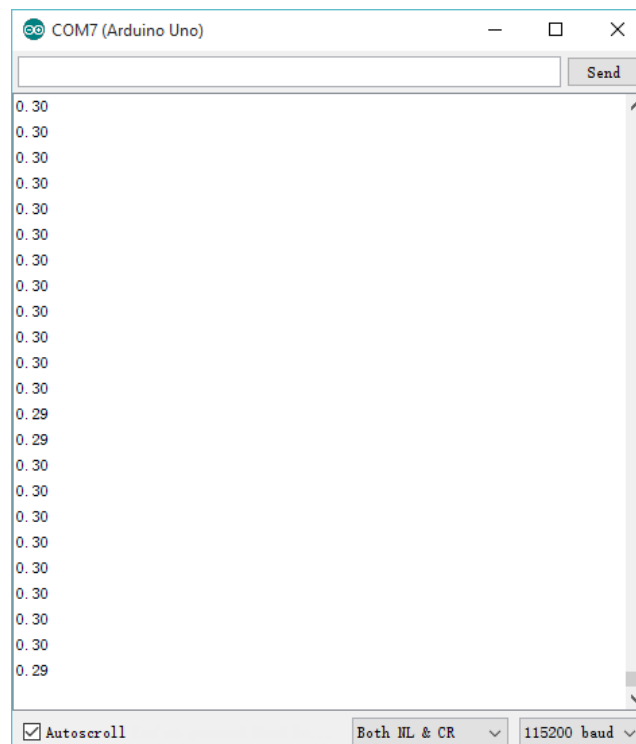
}

```

Result

After uploading the sample sketch, open the serial monitor, when the current is 0A, you may found the reading is not 0A, then if you need amore ccurate readings, you have to revise the code with the reading when the input current is 0A. These are the steps:

1 Check the initial current value when the input is 0A;

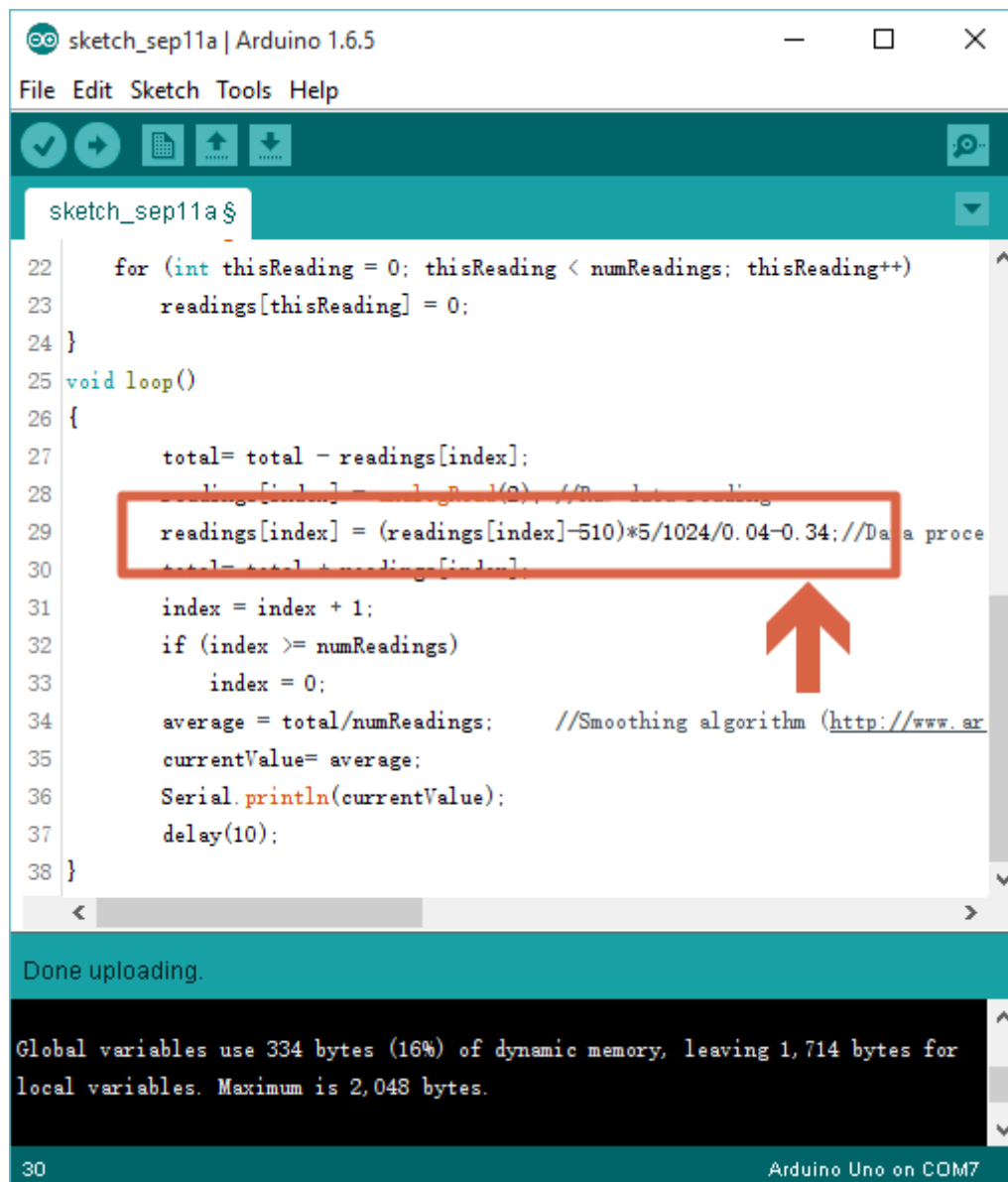


2 Caculate the initial current value:

```
-0.34 = -0.04 - ( +0.30 );
```

3 Revise the sample sketch:

```
readings[index] = (readings[index]-510)*5/1024/0.04-0.34; //
```



```
sketch_sep11a | Arduino 1.6.5
File Edit Sketch Tools Help

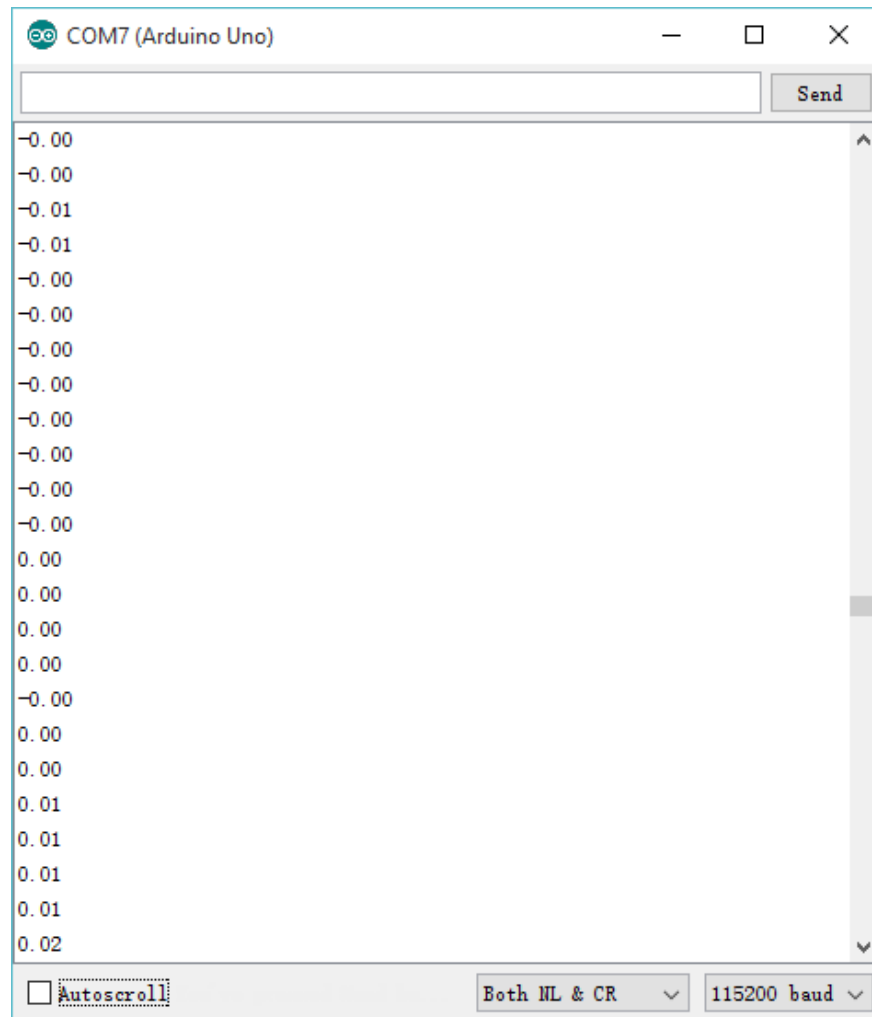
sketch_sep11a $
22   for (int thisReading = 0; thisReading < numReadings; thisReading++)
23       readings[thisReading] = 0;
24   }
25   void loop()
26   {
27       total= total - readings[index];
28       readings[index] = analogRead(0); //Data a proce
29       readings[index] = (readings[index]-510)*5/1024/0.04-0.34; //Data a proce
30       total= total + readings[index];
31       index = index + 1;
32       if (index >= numReadings)
33           index = 0;
34       average = total/numReadings;    //Smoothing algorithm (http://www.ar
35       currentValue= average;
36       Serial.println(currentValue);
37       delay(10);
38   }

Done uploading.

Global variables use 334 bytes (16%) of dynamic memory, leaving 1,714 bytes for
local variables. Maximum is 2,048 bytes.

30 Arduino Uno on COM7
```

4 Upload your new sketch again, then you will find the readings are around 0A, if not, please do the steps above several times to make it.



5 As shown above, it can be applied to detect the current value accurately, and output analog voltage signals.

AC Load

Sketch

```
1 float reading = 0;  
2 float currentValue = 0;  
3
```

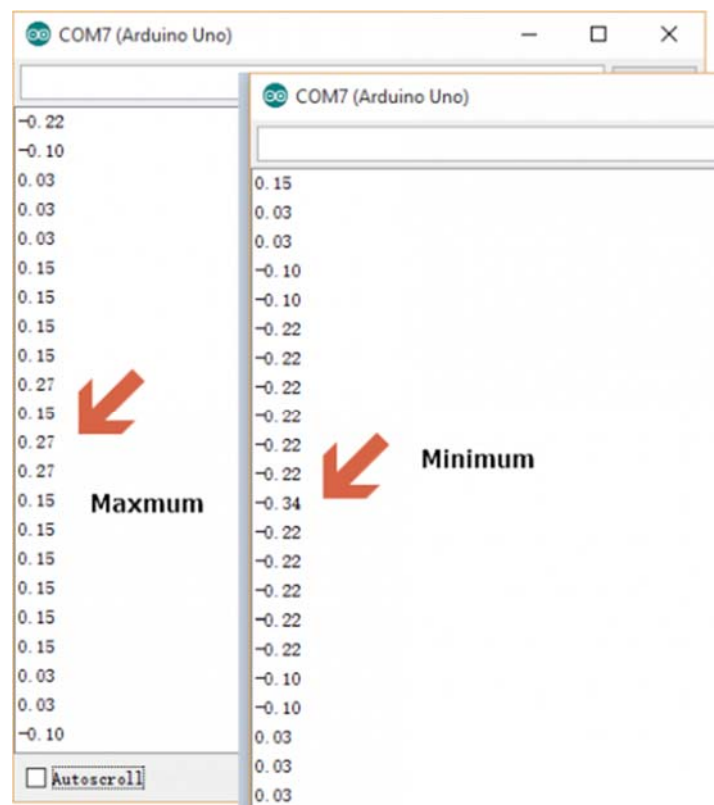
```

4 void setup(){
5   Serial.begin(115200);
6 }
7 void loop() {
8   reading = analogRead(*); //Raw data reading
9   currentValue = (reading - 510) * 5 / 1024 / 0.04 - 0.34;
10  Serial.println(currentValue);
11  delay(2);
12 }

```

Result of AC current

I tested with a lamp @220V~, and got the result shown below: [Read more about the experiment.](#)



Look at the data, it's a Sine wave data described the effective AC current, and you can read the extreme data which are the maximum value 0.27 and the minimum value -0.34, use the maximum absolute one as the (Effective) AC current.

Trouble Shooting

Q1. We see that there are holes to insert sockets, what kind of sockets will fit the holes?

A1. [An example to test a 40W lamp using this sensor.](#)

For any question/advice/cool idea to share, please visit [DFRobot Forum](#).