TA12-100 Arduino AC Current Sensor Tutorial

Overall Description

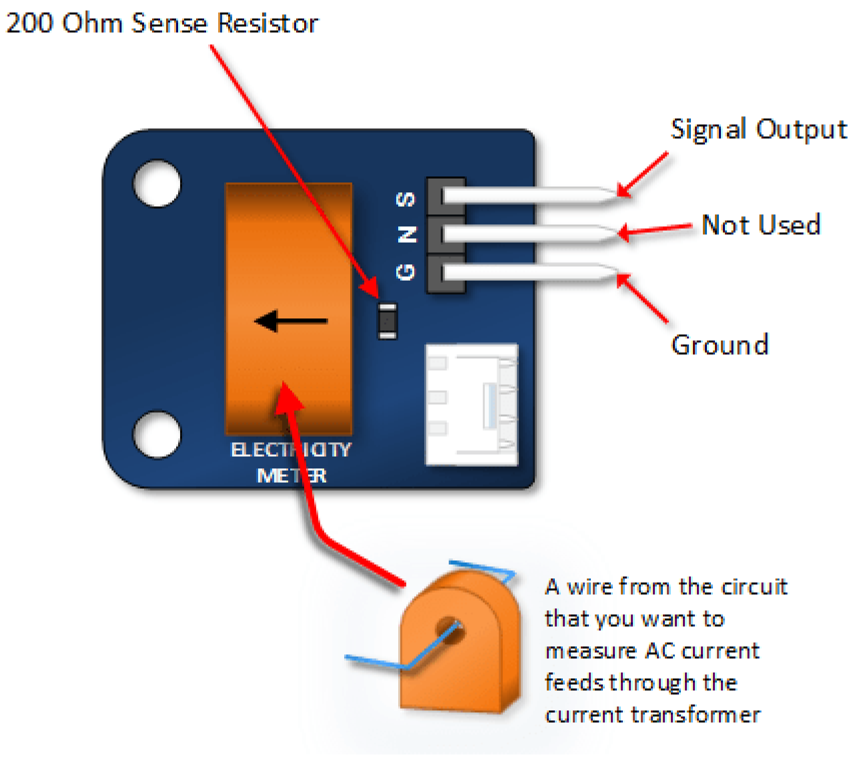
Measuring AC current with an Arduino can be a little tricky. The use of this Itead TA12-100 current transformer makes it possible.

The device uses a 1000:1 voltage current transformer. The output of this transformer has a 200 ohm resistor across its output. The AC current is calculated by measuring the voltage drop across the resistor.

It is particularly useful when measure current at line frequency.

TA12-100 Arduino Current Sensor Pin Outs

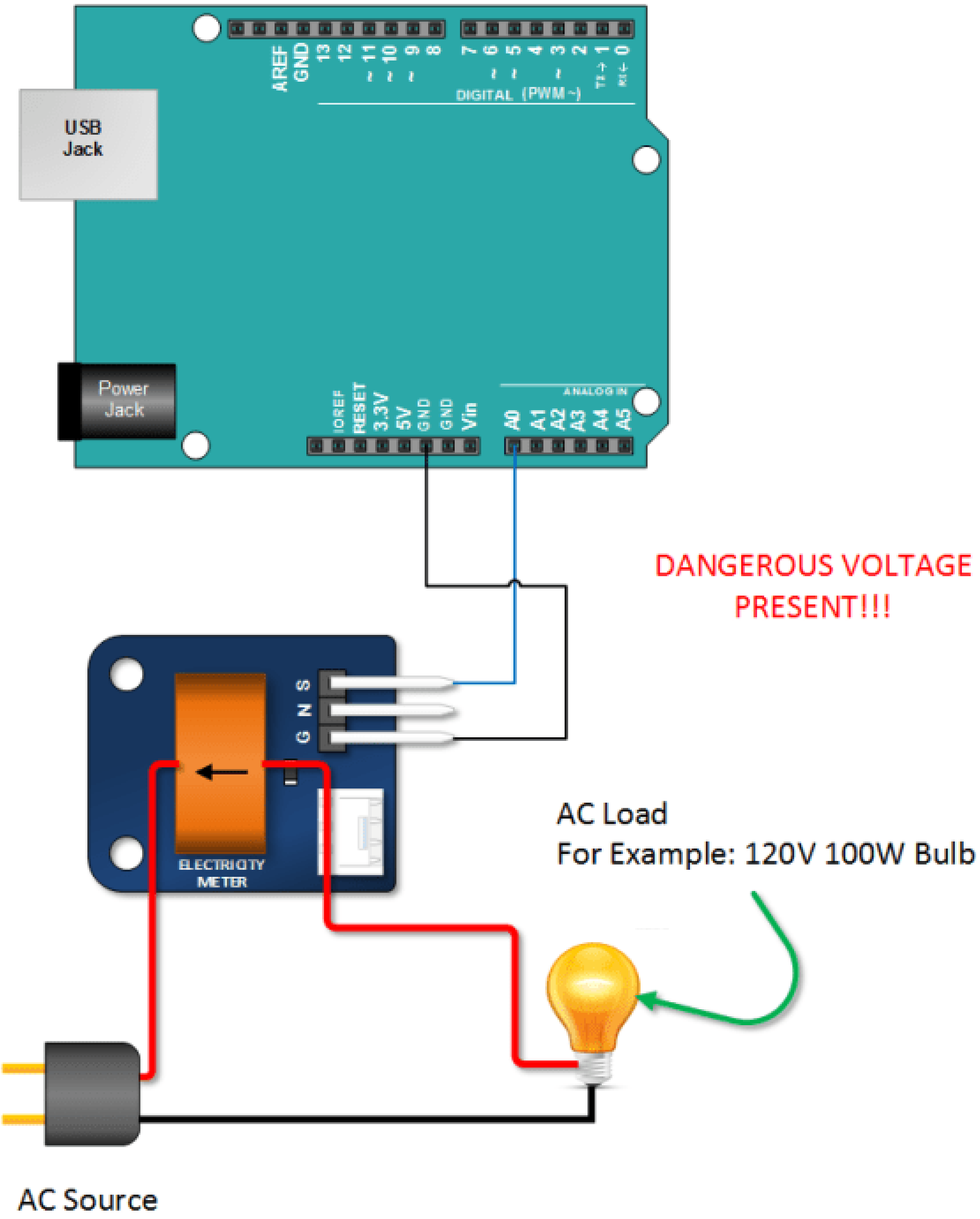
This 5A AC current Sensor requires two connections to your Arduino and it requires you to run the wire that you wish to measure current on through the center of the current transformer.



WARNING – Excercise care.. This tutorial may expose you to lethal voltages.

Connect Your Arduino to AC Current Sensor and AC Load

Use the drawing below: The key here is find an AC Load that won't pull more than 5 amps. For your first test you might want to find something that will draw only about an amp (or 1000 mA).



Copy, Paste and Upload the AC Current Transformer Arduino Sketch

Study this sketch so that you get an idea of how this works. Keep in mind, that the results depend on measuring current of a sinusoidal waveform.

Note that the sketch does following:

- It measures the peak voltage dropped across the 200 ohm transistor that is on the transformer output
- It converts that voltage across the resistor to a current by applying ohms law: (I = E/R)
- It multiplies the peak voltage by 0.707 to get and RMS Voltage across the resistor. (*0.707 as a factor only applies to Sine Waves*).
- It multiplies the RMS current by 1000 to get yield the value going through the wire being measured. (*The current transformer has a 1000 to 1 ratio*).

```
// Henry's Bench TA-1200 AC Current Sensor Tutorial



int sensorTA12 = A0; // Analog input pin that sensor is attached to


float nVPP;   // Voltage measured across resistor

float nCurrThruResistorPP; // Peak Current Measured Through Resistor

float nCurrThruResistorRMS; // RMS current through Resistor

float nCurrentThruWire;     // Actual RMS current in Wire


void setup()

 {

   Serial.begin(9600);

   pinMode(sensorTA12, INPUT);

 }



 void loop()

 {



   nVPP = getVPP();
```

```
/*
Use Ohms law to calculate current across resistor
and express in mA
*/

nCurrThruResistorPP = (nVPP/200.0) * 1000.0;

/*
Use Formula for SINE wave to convert
to RMS
*/

nCurrThruResistorRMS = nCurrThruResistorPP * 0.707;

/*
Current Transformer Ratio is 1000:1...

Therefore current through 200 ohm resistor
is multiplied by 1000 to get input current
*/

nCurrentThruWire = nCurrThruResistorRMS * 1000;
```

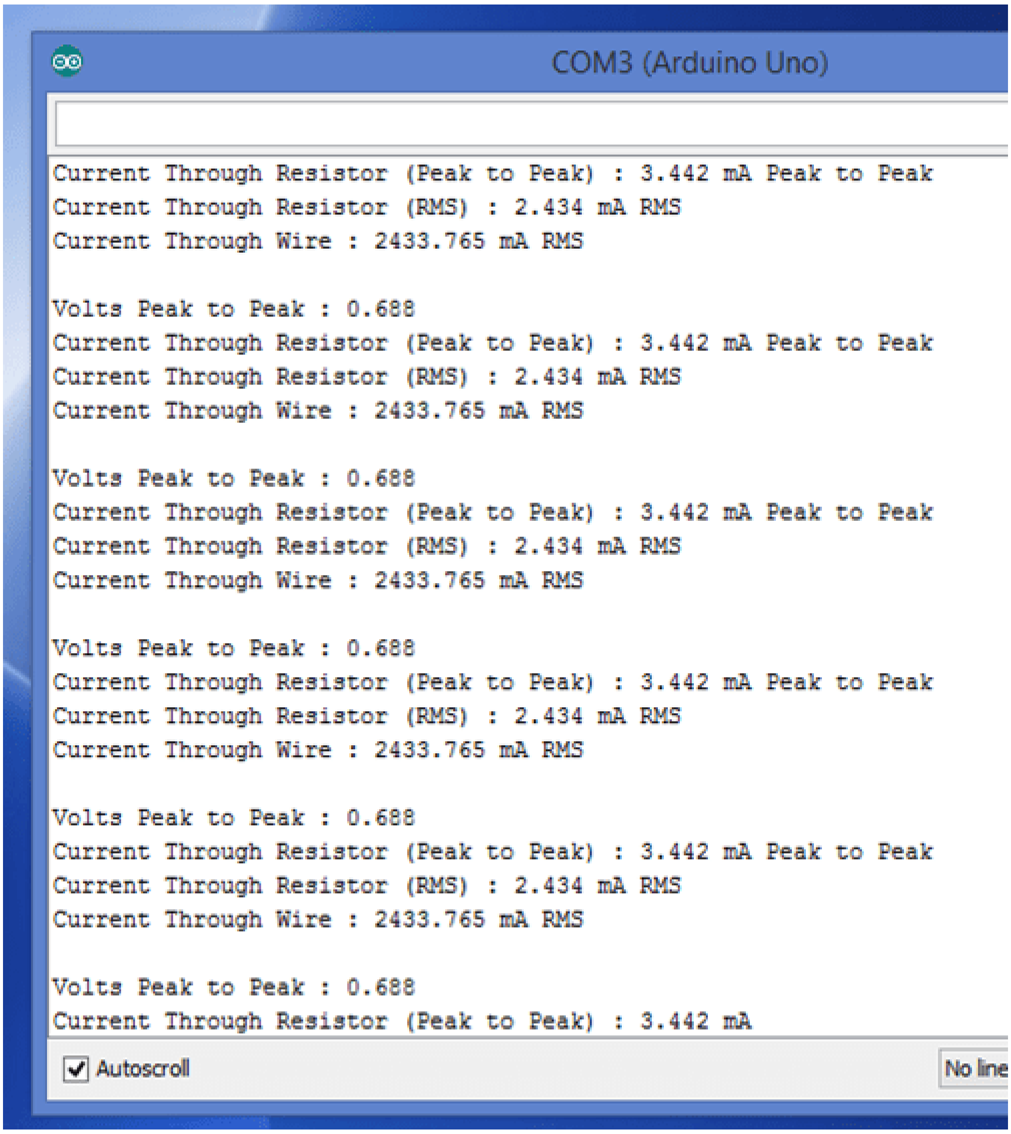
```
  Serial.print("Volts Peak : ");
  Serial.println(nVPP,3);


  Serial.print("Current Through Resistor (Peak) : ");
  Serial.print(nCurrThruResistorPP,3);
  Serial.println(" mA Peak to Peak");

  Serial.print("Current Through Resistor (RMS) : ");
  Serial.print(nCurrThruResistorRMS,3);
  Serial.println(" mA RMS");

  Serial.print("Current Through Wire : ");
  Serial.print(nCurrentThruWire,3);
  Serial.println(" mA RMS");

  Serial.println();



}
```

```
 /************************************
In order to calculate RMS current, we need to know
the peak to peak voltage measured at the output across the
200 Ohm Resistor

The following function takes one second worth of samples
and returns the peak value that is measured
 *************************************/


float getVPP()
{
  float result;
  int readValue;             //value read from the sensor
  int maxValue = 0;          // store max value here
   uint32_t start_time = millis();
   while((millis()-start_time) < 1000) //sample for 1 Sec
   {
       readValue = analogRead(sensorTA12);
       // see if you have a new maxValue
       if (readValue > maxValue)
       {
           /*record the maximum sensor value*/
```

```
          maxValue = readValue;

       }

   }



   // Convert the digital data to a voltage

   result = (maxValue * 5.0)/1024.0;



   return result;

 }
```

Verify Your Sketch Output

Open your Arduino IDS serial monitor.   Your results should look something like the picture below.



```
Current Through Resistor (Peak to Peak) : 3.442 mA Peak to Peak
Current Through Resistor (RMS) : 2.434 mA RMS
Current Through Wire : 2433.765 mA RMS

Volts Peak to Peak : 0.688
Current Through Resistor (Peak to Peak) : 3.442 mA Peak to Peak
Current Through Resistor (RMS) : 2.434 mA RMS
Current Through Wire : 2433.765 mA RMS

Volts Peak to Peak : 0.688
Current Through Resistor (Peak to Peak) : 3.442 mA Peak to Peak
Current Through Resistor (RMS) : 2.434 mA RMS
Current Through Wire : 2433.765 mA RMS

Volts Peak to Peak : 0.688
Current Through Resistor (Peak to Peak) : 3.442 mA Peak to Peak
Current Through Resistor (RMS) : 2.434 mA RMS
Current Through Wire : 2433.765 mA RMS

Volts Peak to Peak : 0.688
Current Through Resistor (Peak to Peak) : 3.442 mA Peak to Peak
Current Through Resistor (RMS) : 2.434 mA RMS
Current Through Wire : 2433.765 mA RMS

Volts Peak to Peak : 0.688
Current Through Resistor (Peak to Peak) : 3.442 mA
```