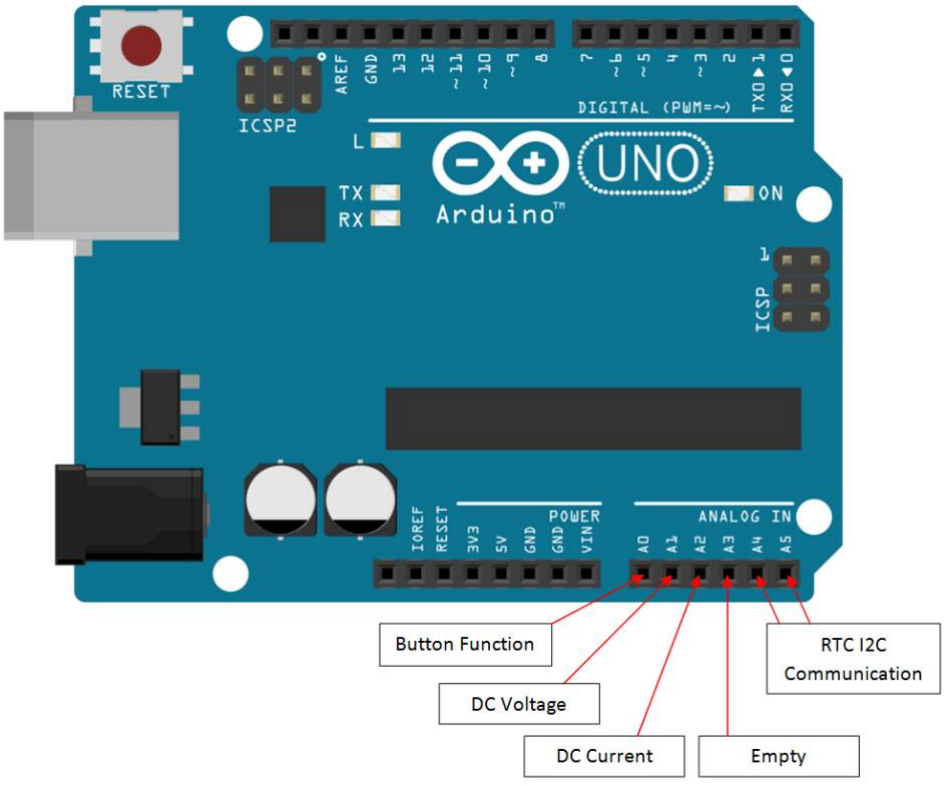
Mesure d’énergie électrique en courant continu

DIY DC Energy Meter with Arduino

12 janvier 2020 https://solarduino.com/diy-dc-energy-meter-with-arduino/

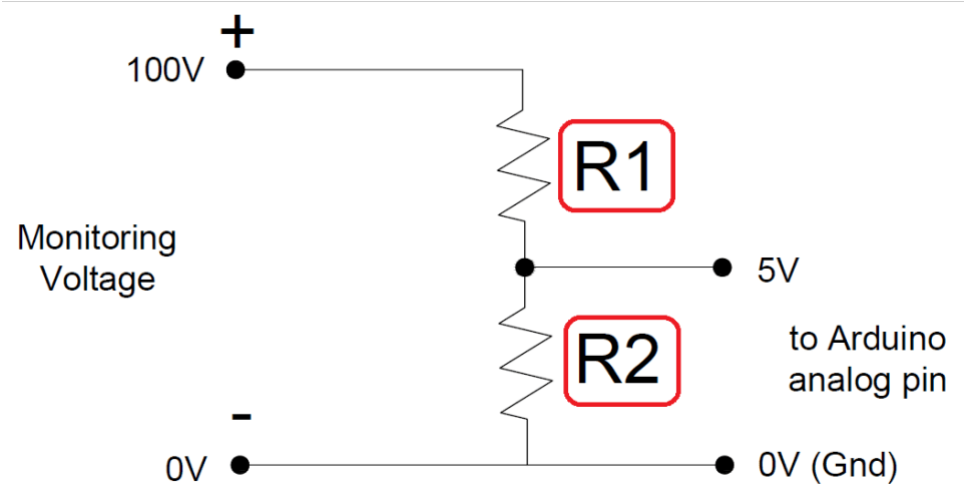
Energy Meter is a very useful device that displays important information of electrical parameters. There are 4 important electrical parameters in a Direct Current (DC) Energy Meter, which are **DC Voltage, Current, instantaneous Power and accumulate Energy consumption**. This device is used in DC energy generation or DC load such as Solar PV System. To cut everything short, in order to obtain all the 4 parameters above, we need 2 sensors, the **DC voltage sensor** and **DC Current Sensor**.

Arduino has the ability to measure **DC voltage and DC current (via module) using analog input pins**. For Arduino UNO, there are **6 analog input pins** (A0-A5) where you need separate pin for each measurement. If you stacked up a LCD Display Shield, Analog Pin A0 is automatically occupied by the button function. If you are going a step further by adding Datalogger Shield, Analog Pin A4 and A5 are also occupied for I2C communication for the Real Time Cloak module in the Datalogger Shield. Technically it left Analog Pin A1 to A3 for DC Current and DC Voltage pin. In this project, I will set A1 to measure DC Voltage and A2 to measure from current module. **Do not reverse the voltage polarity which may damage the pins.**

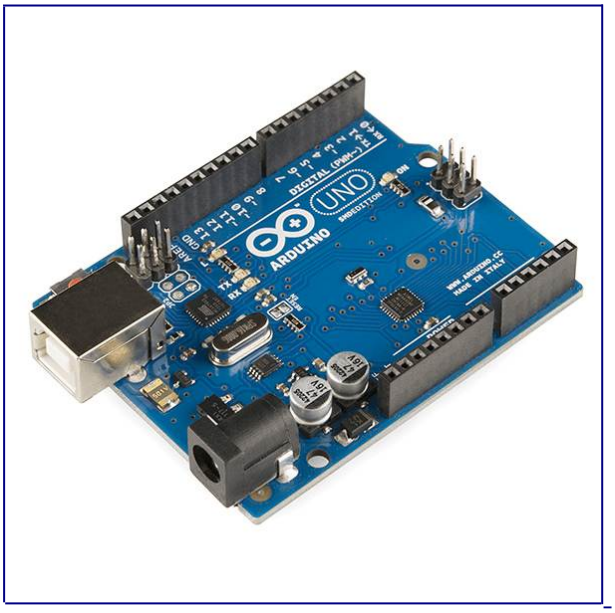


Measure Voltage using voltage divider method

If your measure DC voltage value which is not more than 5.0V, you can directly **c**onnect to the analog pin without any modification. In order to measure higher voltage such as 18Vdc, 48Vdc, 100Vdc or even 500Vdc, **voltage divider method** is used to split and reduce the measurement voltage into a 5V range. The voltage divider consists of **two resistors connected in series** as shown in the diagram below. All you need is just 2 resistors with different resistance values.



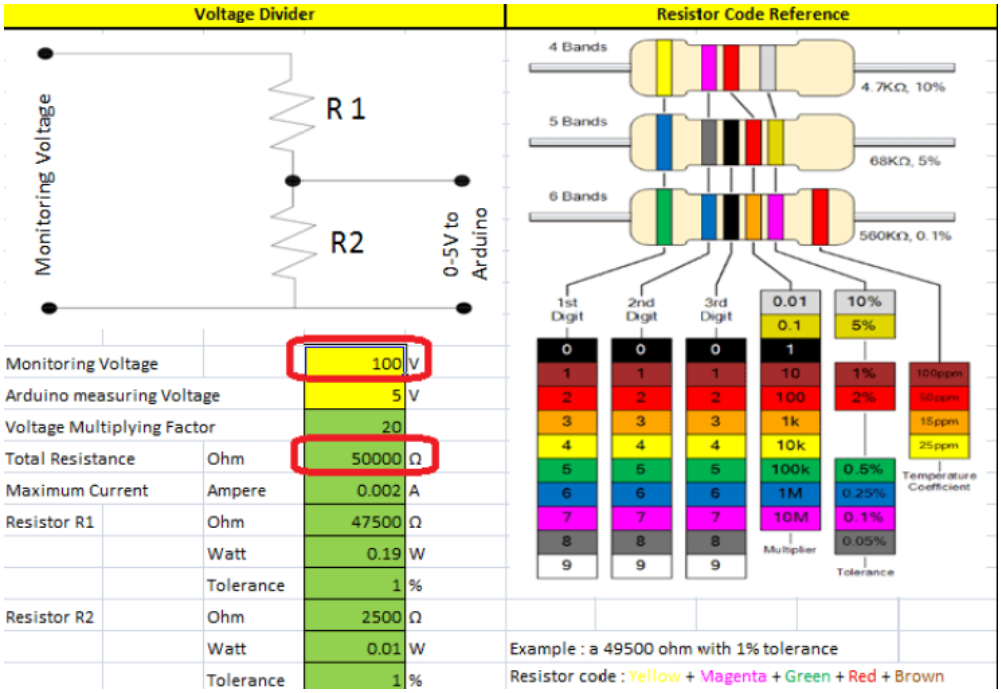
When the voltage is applied across the pair resistor of different resistance, it creates a **voltage drop** based on each resistor and it can be used as reference value and it is directly proportional to the total voltage value. Higher resistance R1 tends to have larger voltage drop while smaller resistance R2 will have smaller voltage value which is within acceptable range of Arduino.



Arduino UNO (compatible board)

If you still not yet own an Arduino Micro-controller Board, you can get it cheap at our affiliate link ***here*** !!!

For easy calculation, below we have attached an excel sheet that can automatically calculate the value of the resistors. First, key in the **Monitoring Voltage**. It is the maximum value of your DC voltage monitoring range. The next step is to key in the **Total Resistance** value. It is recommend that the total resistance ranges from 50,000 ohm to 300,000 ohm.



Total Resistance is **flexible**. However, small total resistance will have large consumption and heat dissipation for high voltage measurements which end up you have to buy larger wattage resistor. In the contrary, too large total resistance may end up voltage measurement processing too slow or not accurate in low voltage measurements.

As a guide, **100Vdc and below** use Total Resistance value of **50,000 ohm, 200Vdc** and below key in **100,000 ohm**, and voltage less than **500Vdc** key in **300,000 ohm**. Once you keyed in Monitoring Voltage and Total Resistance values, the Excel sheet automatic calculate all the relevant specs for the two resistors.

There are **3 important information** that need to be specified during the resistor purchase. You may purchase the resistor at our affiliate link ***here*** !!

1) Resistance value

You will end up getting 2 resistors of different resistance value. You might get weird resistance value in the calculation sheet, for example 44,565 ohm and 5,325 ohm. It is just a guideline and all you need to do is to get a round figure which is closer but **larger value for resistor R1** and **smaller value for resistor R2**. Example: 45,000 ohm and 5,300 ohm. This is to ensure monitoring voltage can be measured within 0-5V range.

2) Resistor Wattage

Similar to other products, resistor itself has its **withstand's wattage rating**. Oversize wattage rating is always better than using small wattage rating on large consumption which may lead to resistor burnt. Resistor R1 always has more heat dissipation than resistor R2 thus it is not strange that both

resistors are not with same wattage requirement. Just get a value larger than the wattage rating as specified in the excel sheet. The standard wattage of a resistor is: 1/8 Watt (0.125W), 1/4 Watt (0.25W), 1/2 Watt (0.5W), 1 Watt, 2 Watt and 5 Watt.

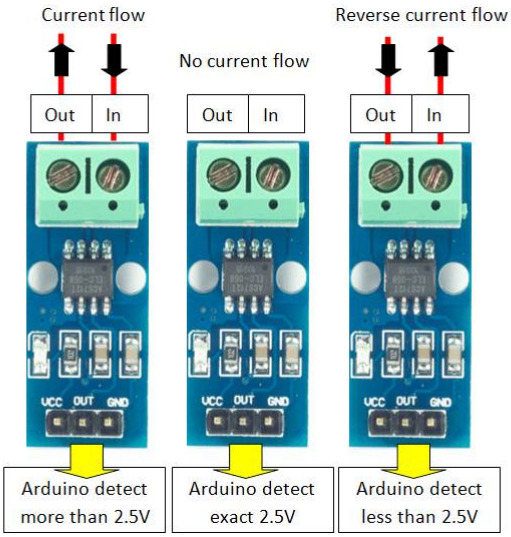
3) The resistance tolerance in %.

Resistor Tolerance is the percentage of deviation from the stated resistance value. If you do not have a Multimeter to measure the actual resistance value, you will need to purchase a better accuracy resistor. Resistors are very small and cheap component; if possible get most accurate ones. 1% is widely use and suitable for monitoring and measuring purposes.

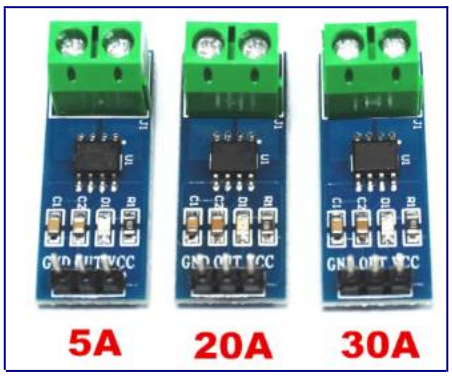
You may purchase the resistor at our affiliate link ***here*** !!

Measure Current using Module

The current sensor that is widely used for Arduino is the **ACS712 Current Sensor Module**. It utilizing hall-effect phenomenon which voltage is produced from the movement of current within the region of magnetic field. The voltage produced by hall effect is directly proportional to the applied current making it suitable to estimate the applied current from the voltage sensed.



The sensor can measure current in 2 direction. Reverse current will not damage the sensor but the voltage produced will be in reduced. As we know, Arduino analog input only read **positive integer values**. In order to measure 2 direction, the zero point should be at half the total voltage range (0 to 5V) which is 2.5V. This is true if the supply voltage to the sensor is 5V.

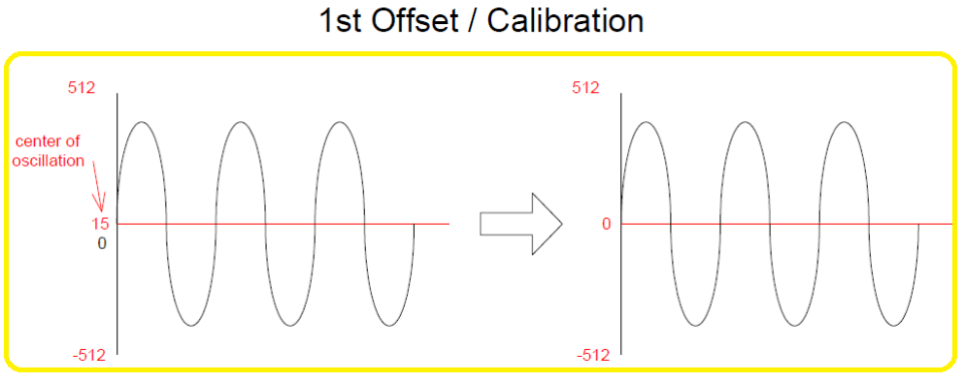


The standard **ACS712 Current Sensor Module** rated at **5A, 20A and 30A** which are suitable to most applications. You may get them by our affiliate link ***here*** !!! The 5A module has the resolution of 185mV/ampere, 20A module has 100mV/ampere while 30A module has the resolution of 66mV/ampere.

Assume using 5V supply as standard, the 5A module will provide voltage at 2.5V +/- 0.925V, 20A module will provide voltage at 2.5V +/- 2.0V while 30A module will provide voltage at 2.5V +/- 2.0V. These modules require **direct contact** which I think is a major drawback. **It has to be connected in series to the measured value**. The wiring of the existing system need to be altered in order to fit the module into the existing system.

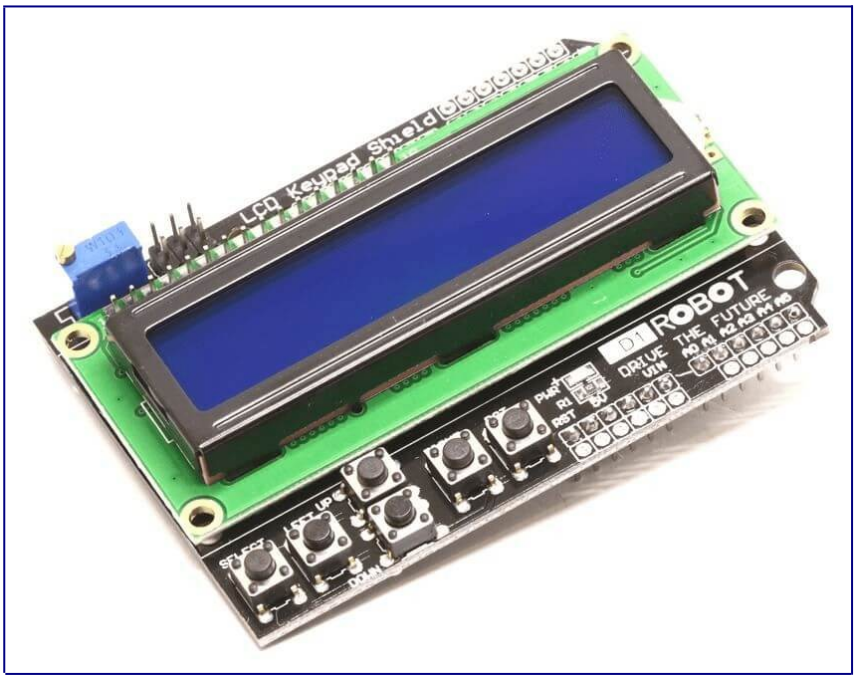
Fortunately, there is also a hall-effect sensor type with split core transformer type (as picture on left). It is the **Hall-Effect Split-Core Sensor HSTS016L** module. The model ranges from **10A up to 200A**. With split core current sensor type, not alteration on the existing system required. You can get it via our affiliate link ***here*** !!! The output voltage of this sensor is **2.5V +/- 0.625V** with decent accuracy.

Unlike Voltage Sensor, Current Sensor requires **initial offset setting**. You may need to offset the value by checking the initial false current when no current draws during Arduino startup. Each sensor has its own deviation error. When there is no current sensed, the sensor might not be 100% at middle point of voltage value. Some might be remaining at few milli voltages **above / below the middle point** even though after averaging. This might be due to voltage supplied not in exact 5V or due to the sensor itself.

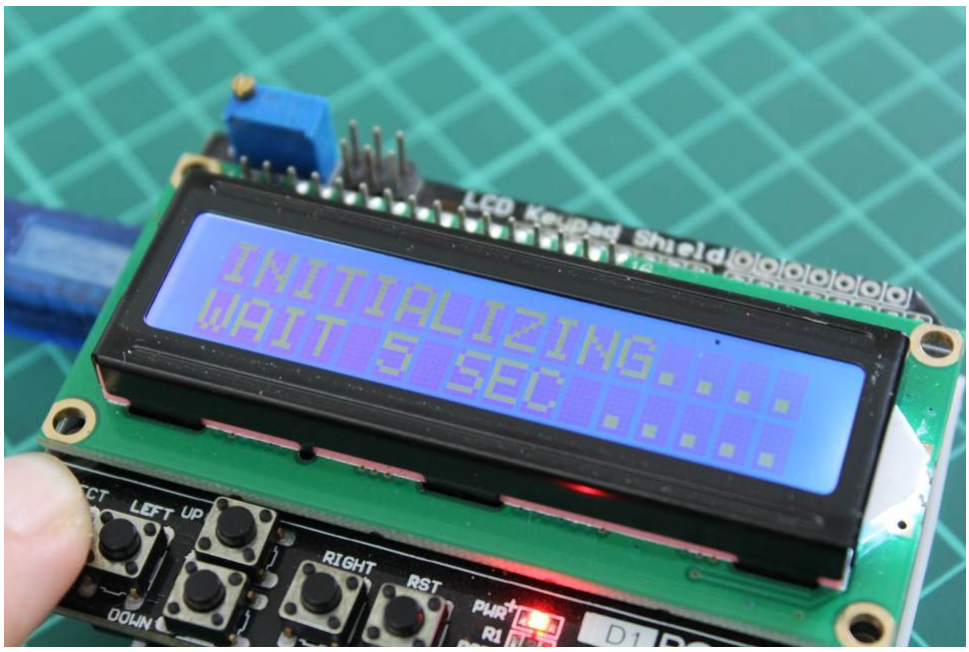


Secondly, current sensor is also a **sensitive sensor**. The output reading of the sensor seems to have **electrical noises and its** value fluctuates all the time even when there is no current detected. It is more obvious if the measurement is in a smaller time frame. In order to greatly reduce this phenomenon, multiple samples must be taken for **averaging must be done.**

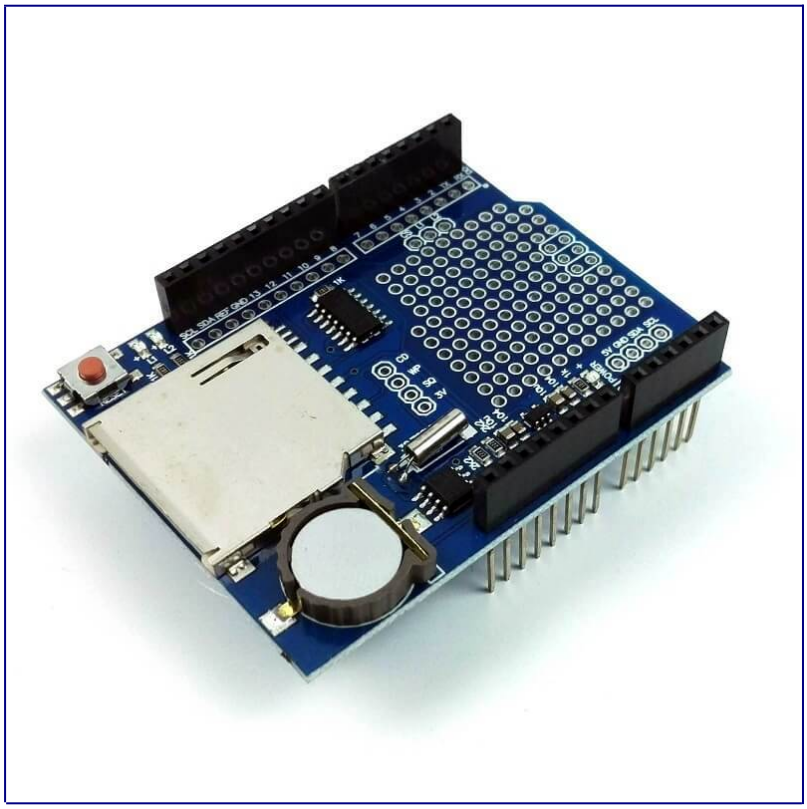
Our code is designed to display a value which is derived from **averaging 1000 samples in every second.** Each sample is recorded every 1 milli second (0.001 second). The single averaged value is then to be displayed at Serial Monitor and LCD Display. With this, the fluctuation of value is way lesser compare to taking 1 sample reading every second. Make sure the sensor cables are tight because minor movement of wires might affects on the wire terminal connections thus affecting the accuracy reading.



I recommend you to add a 16X2 **LCD Display Shield** which can be directly fit on to the top of the Arduino board without the need of extra wiring for the LCD Display. Without the LCD Display, you can only monitor the measured current value on PC via Serial Monitor. You can get the LCD Display board at our affiliate link *here* !!!.



The good news is you **do not need to manually calibrate** the offset settings if you got the **LCD Display Shield** with you. Below we have attached the code that utilizes the button function that could **automatically calibrate** by itself when you pressed the **SELECT Button**. You may download from the end of this page below.



Datalogger Shield

If you plan to record the data in a proper way, you may consider this Datalogger Shield. It allows your arduino to record your data in SD Card. Datalogger shield is often installed together with LCD

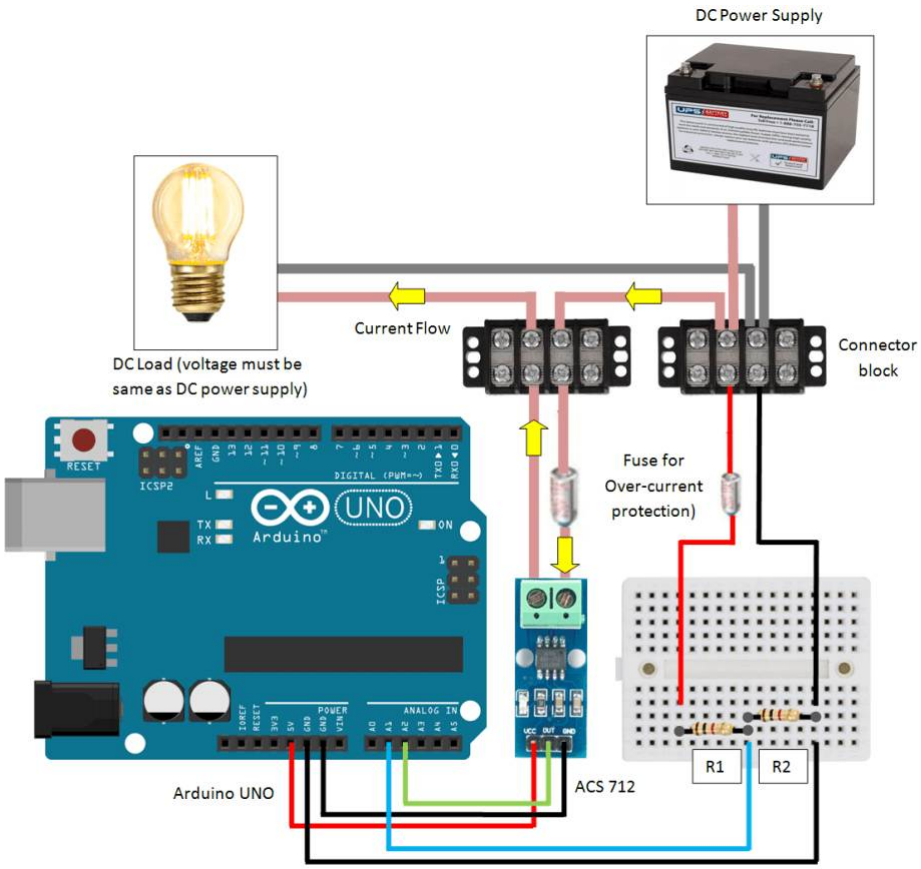
Display shield. Please find it at our affiliate link ***here*** !!! For more about this Datalogger Shield, kindly visit our post ***here***.

Hardware Connection

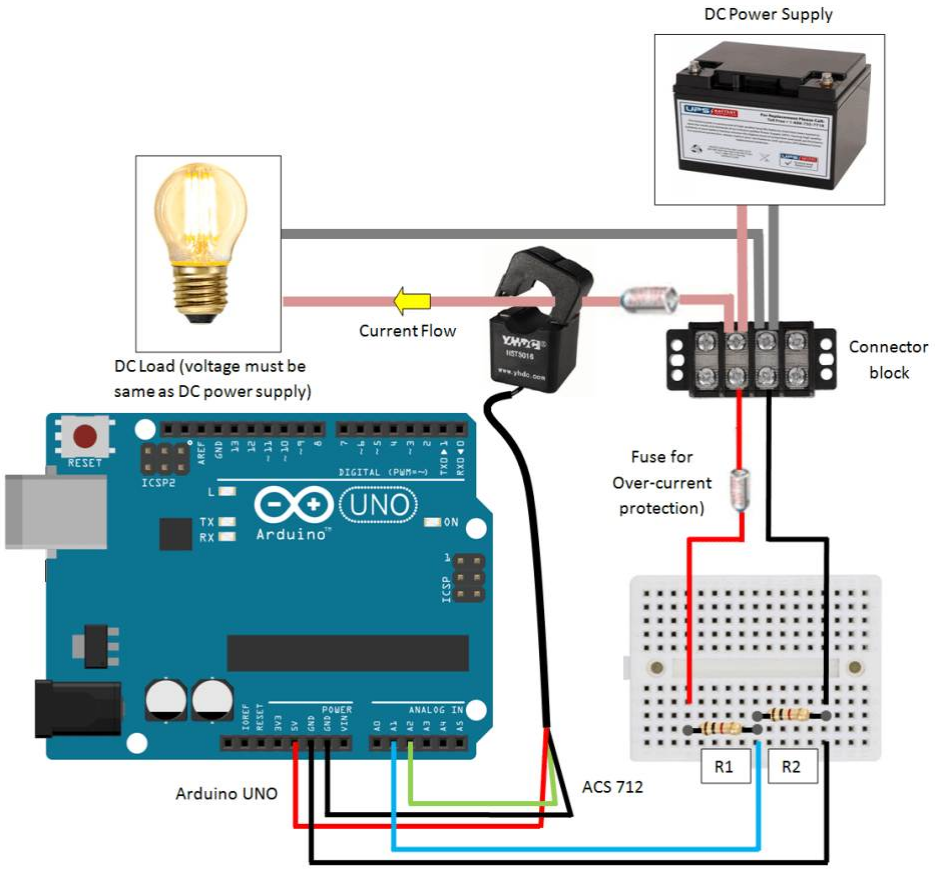
Once you get your **Resistors**, **current sensor module** and **Arduino** Board ready, you may start to do hardware wiring. Below is the schematic of the whole wiring. You may also need some tools and accessories. Be sure your connection cable is tight and module shall be installed in such a way no movement at all.

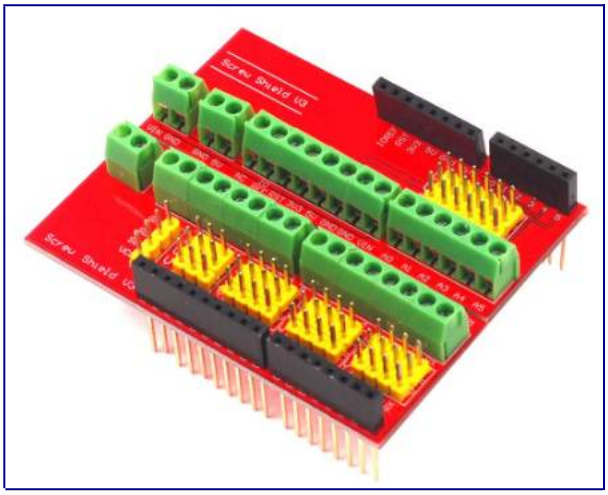
You can stack up screw shield, LCD Display Shield, and Datalogger Shield on top of Arduino UNO. No additional wiring is required as the shields are meant for ad**ding function without need of extra wiring.**

Wiring using ACS712 current module



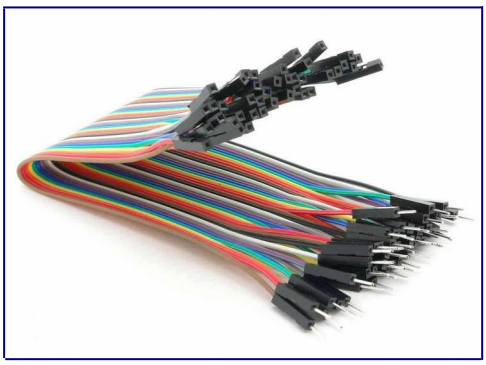
Wiring using HSTS016L Split-Core Sensor module





Screw Shield / Expansion Shield

When there are a lot of wiring around especially more than 1 sensor, sharing pins will be difficult as existing pins (ground and 5V) are limited. This shield provides a lot of convenient terminals for each of the input and output pins. The shield can be mounted directly on top of the Arduino Uno board or in between the shields which made it very convenient to use. You can get it at our affiliate link ***here*** !!!



Dupont Line Wires

You may need Dupont Line Wires to connect Arduino board and Sensor Module. It is available at our affiliate link ***here*** !!!

Software Codes

The final step would be adding source code onto Arduino board. I assume you have installed the Arduino Software. If you still have not installed the software, the link ***here*** can bring you to the official download site. Once you have downloaded the software, you may download the code file (.ino) for this application below (right click save link).

There are **2 source code files attached**, the first source code is DC Energy Meter code with LCD Display Shield without Datalogger Shield. The second source code is for people that have both LCD Display Shield and Datalogger Shield. **In order to use the Datalogger Shield, make sure you download and install the library as required. You can visit my post about datalogger shield for detail *here*** !!!.

Once the code has been uploaded to the arduino board, the output can be shown in LCD Display. As for second code user, besides display, the data is also saved to SD card every minute by default. I will not display the code here because it is long. You can download the .ino file to see for your own. Almost all code lines are with explanation.