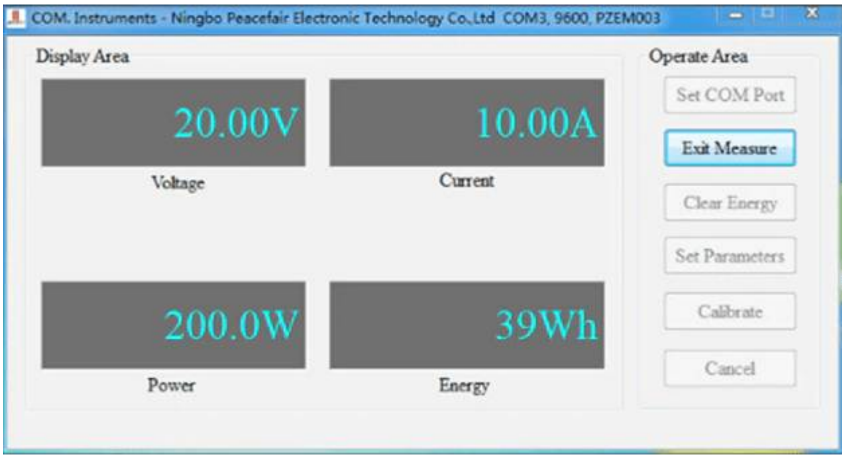
PZEM-017 DC Energy Meter with Arduino

6 janvier 2020 https://solarduino.com/pzem-017-dc-energy-meter-with-arduino/

https://youtu.be/vvQcoQesAD0

PZEM-017 is a DC communication module that can measure DC power up to 300VDC and current measurement is subject to external shunt installed ranges 50A, 100A, 200A and 300A. It is a module that made from Peacefair, a very famous Chinese brand with good quality and price that specialize in Metering products. This module **can measure Voltage, Current, Power and Energy.**

PZEM-017 does not have Display on it, that is why it is cheap to own. It has a built in RS485 communication interface using Modbus-RTU protocol similar to most industrial devices. The measured values can be displayed on your PC using UART to RS485 converter with its come-along software. You can download the software ***here*** !!!



COM. Instruments - Ningbo Peacefair Electronic Technology Co.,Ltd COM3, 9600, PZEM003

Display Area

Voltage: 20.00V

Current: 10.00A

Power: 200.0W

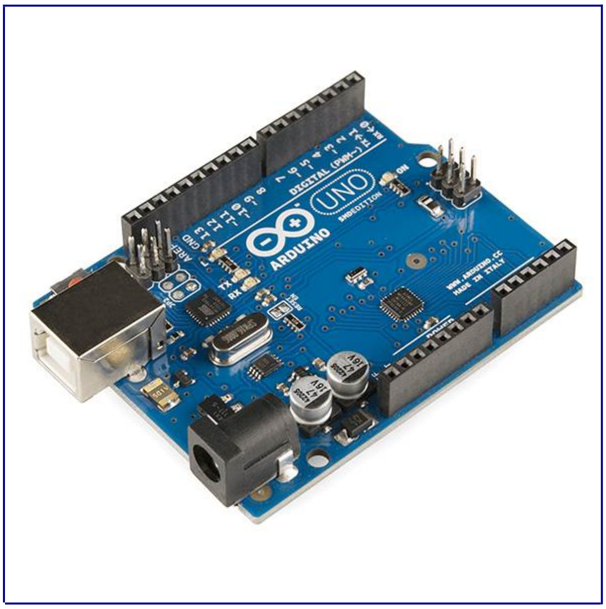
Energy: 39Wh

It sounds complicated and useless, but not with Arduino microprocessor board**. Today I will show you how easily to read the measurement values using Arduino UNO.** You can use the values for further processing by adding Arduino Code to the measured values such as recording them using **Datalogging Shield**, or display them using **LCD Display Shield** just to name a few. It acts as the DC Power sensors for Arduino so that only minimum wiring and work is required comparing you buy and wire all the modules individually and create their own individual part of codes. Has it convinced you? Let’s get started !

Warning ! You may now dealing with high voltage and high power source ! We assumed that you have the basic electrical knowledge and know what you are dealing with. You may need guidance from experienced guys if you are new to electrical work. Safety and Precaution must be always have in mind. We shall not be responsible for anything happening to you.

PZEM-017 DC Energy Module

PZEM-017 is a DC Energy Module that can support up to 300Vdc and the current measurement is subject to external shunt installed. It can be up to 50A, 100A, 200A and 300A. You can get it at a very competitive price at our affiliate link ***here*** !!!



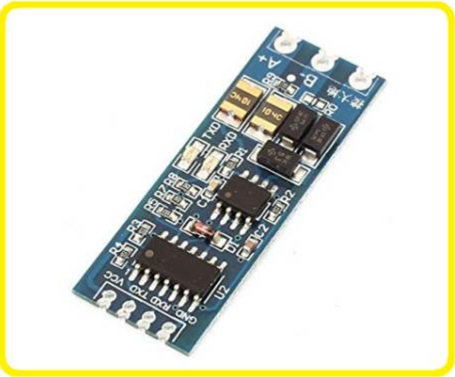
Arduino UNO (compatible board)

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

Since this DC Energy Measurement Device uses RS485 communication, we need a converter module that can convert **RS485 to Serial** communication in order to communicate with Arduino board. We need a **UART TTL to RS485 converter module.** Generally there are **2 types** of RS485 converter module **1)** only 2 communication output pins **named Tx and Rx** and **2) 4**

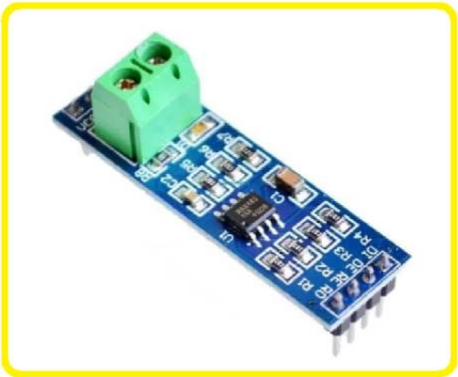
communication output pins named DI, DE, RE and RO). In this example we are using the **4 pins module (Type 2).**

Type 1



2 communication pins : Tx & Rx

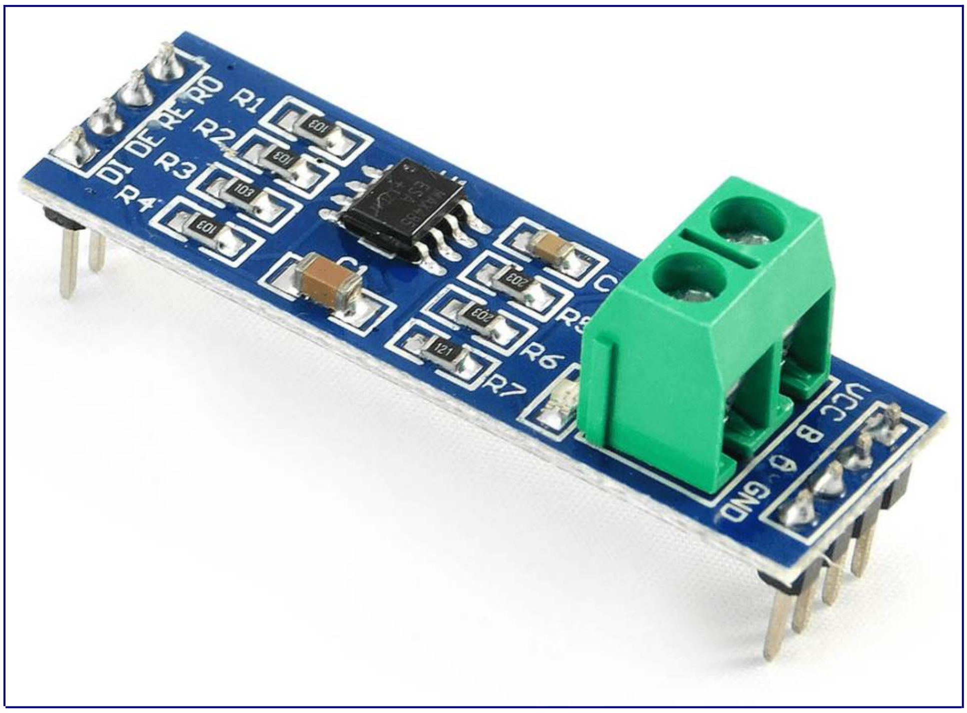
Type 2



4 communication pins : DI, DE, RE & RO

Arduino UNO has only 1 physical Serial (TTL UART) port which you use the USB cable connecting the Arduino IDE code software to your Arduino Board. It allows you to communicate with Arduino board such as upload code to arduino board and read values from Arduino board to Serial Monitor. It occupied pins 0 (Rx) and pin 1 (Tx). We cannot use Arduino UNO anymore and we need Arduino Board that has more Serial Ports such as **Arduino Mega**. In this example we will be using Arduino Mega board.

If you slowly used to understanding the code which debug code is no longer required, then you can switch to Arduino UNO with LCD Display Shield (for example) and don’t depend on Serial Monitor for troubleshooting. However, I will put 2 codes below which 1 with Arduino UNO (if you trust the code without error), and code with Arduino Mega which I will use as an example. Both codes are with LCD Display Shield.



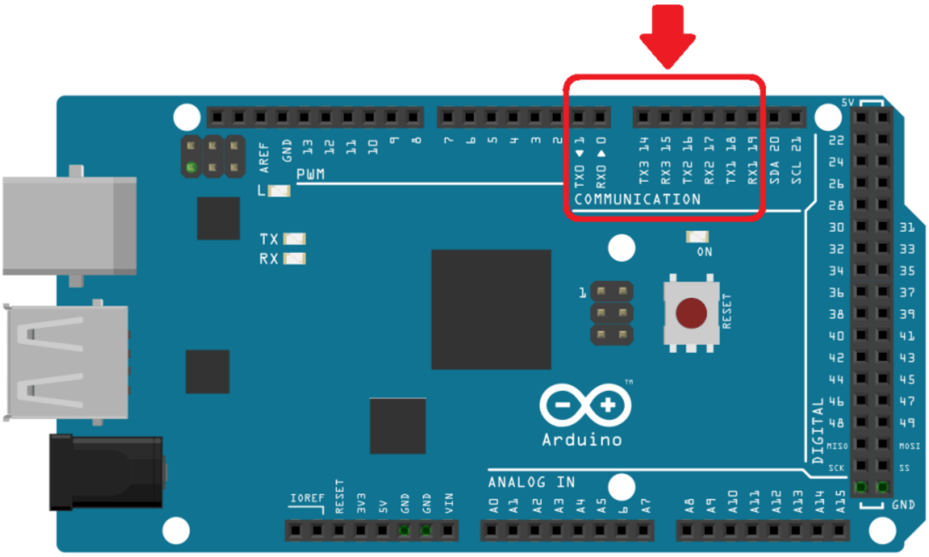
MAX485 UART TTL RS485 converter

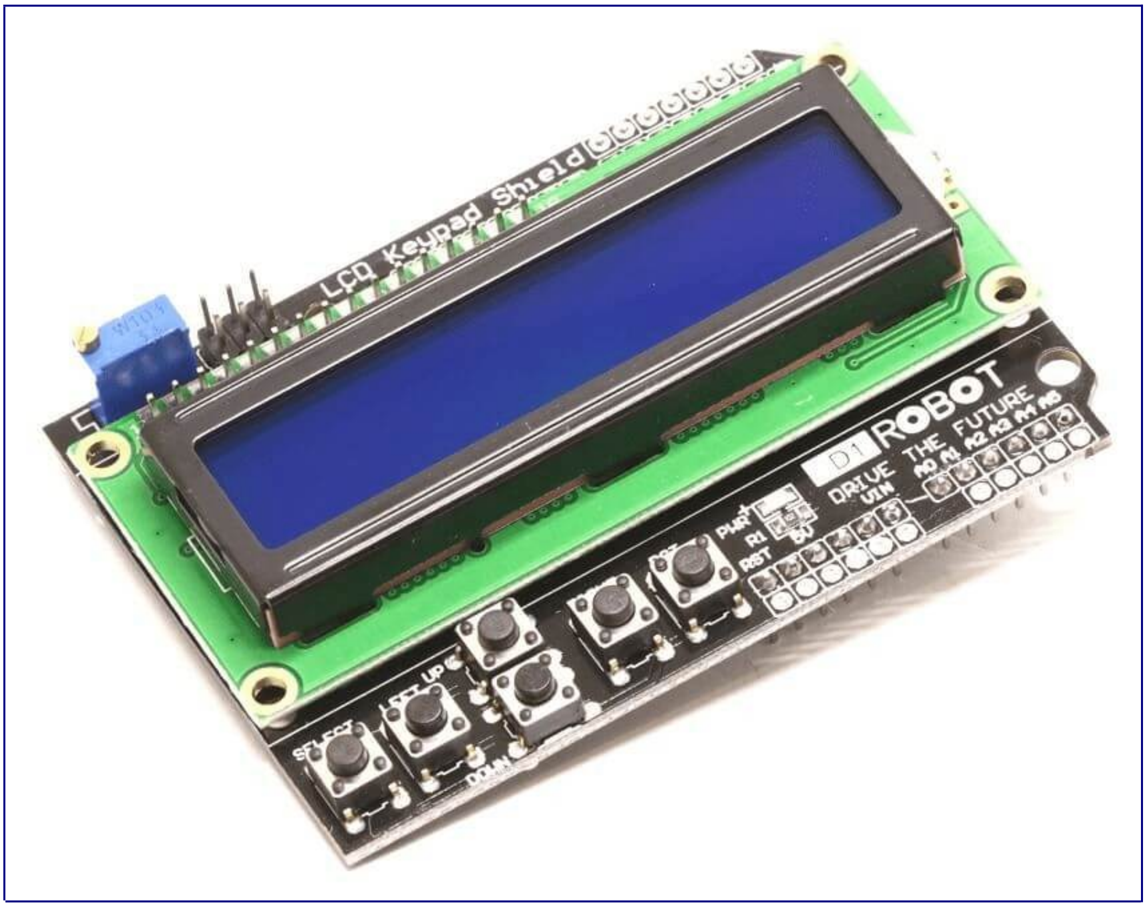
This is a module that converts between RS485 communication and Serial (UART TTL) communication. This module enables Arduino to communicate, read or provide command to devices that using RS485 communication. Quickly grab this module at our affiliate link ***here*** !!!

Arduino Mega 2560 (compatible board)

This is an upgraded and superior Arduino Board compare to Arduino UNO. It has 16 Analog Pins, 40 Digital Pins, 4 Serial ports, and 1 separate I2C port. It has large memory and act facter compare to Arduino UNO. Come grab some at our affiliate link ***here*** !!!

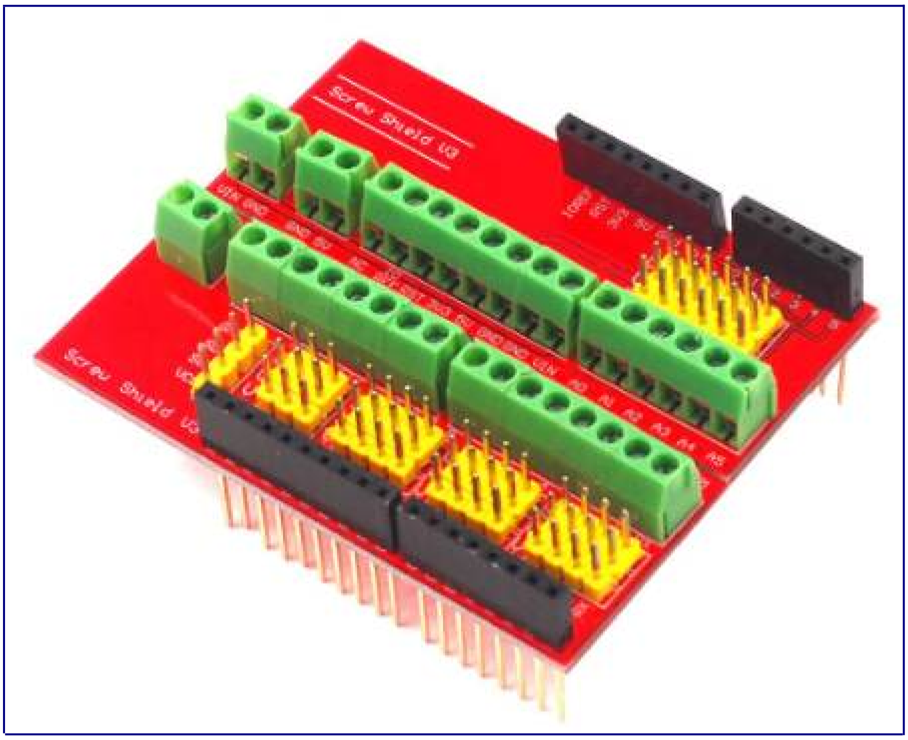
Arduino has the ability to communicate with RS485 devices using Serial Ports with the help of the RS485 to Serial Converter. Arduino UNO has only **1 Serial Port** which is **pin 0 (Rx) and pin 1(Tx)**. As for Arduino Mega 2560, it has 4 Serial Ports name **Serial, Serial1, Serial2 and Serial3**. **Serial Port has the same pin locations which are pin 0 (Rx) and 1 (Tx), Serial1 Port at pins 19 (Rx) and 18 (Tx), Serial2 Port at pin 17 (Rx) and 16 (Tx) and lastly Serial3 Port at pins 15 (Rx) and 14 (Tx).**





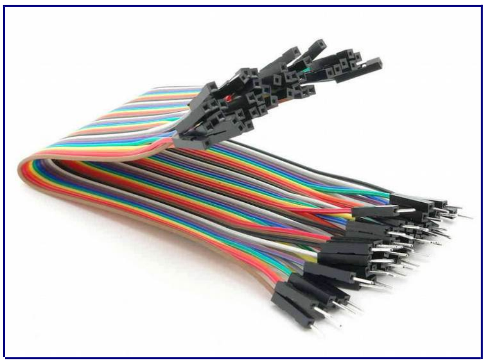
LCD Display Shield

This is a shield that allows the output value of your Arduino board to be displayed on the screen. Since it is a shield, you can just stack it on Arduino board without the need of extra wiring for the LCD Display. You can get the LCD Display board at our affiliate link ***here*** !!!.



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 board or in between the shields which made it very convenient to use. You can get it at our affiliate link ***here*** !!

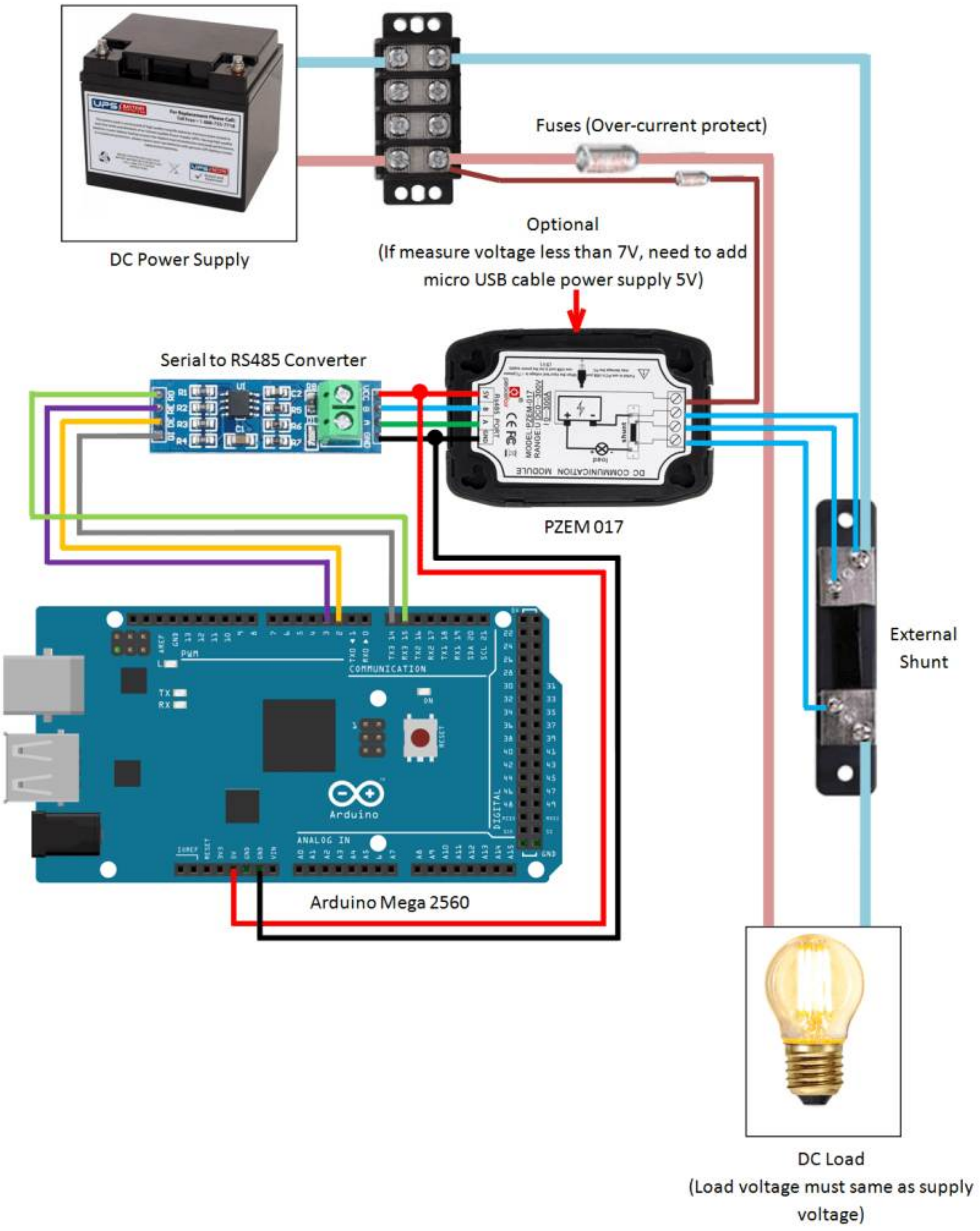


Dupont line Wires

In order to connect wiring between Arduino board and module fast and easy, you need the **dupont line cables**. These cables are suitable during testing and troubleshoot where a lot of wire modifications. You can get it at our affiliate link ***here*** !!!

Hardware Connection

You are dealing with high power or high voltage DC power source, any exposed conductors need to be isolated and fused. Turn the measurement power source OFF during hardware connection. I recommend to use **ferrule lug for terminal tight connection** and to prevent stranded conductor accidentally touching adjacent cable line which might cause short circuit and fire risk. Furthermore, no conductive materials are placed around the workplace and use non-metallic enclosure or project box.

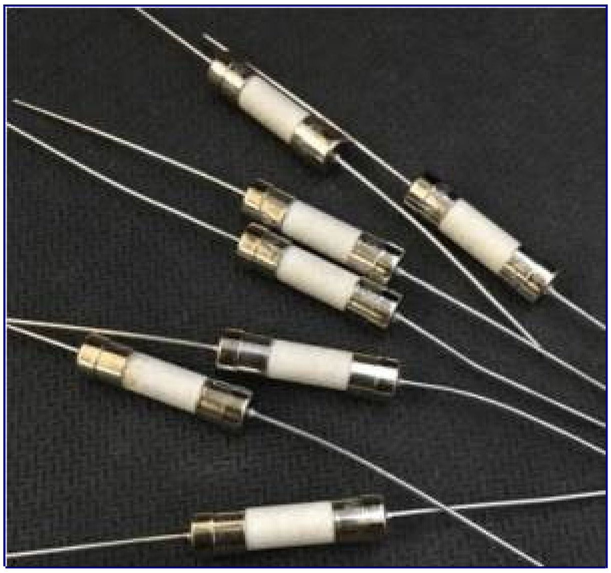

DC Power Supply
Fuses (Over-current protect)
Optional
(If measure voltage less than 7V, need to add micro USB cable power supply 5V)
Serial to RS485 Converter
PZEM 017
External Shunt
Arduino Mega 2560
DC Load
(Load voltage must same as supply voltage)

Connection is quite direct. First, connect 5V power supply from Arduino board to Serial-RS485

Converter and meter. Optionally you can get the separate 5V power supply. Secondly, connect the 2-wire (A & B) RS485 between the converter and the meter; A connect to A terminal while B connects to B terminal. The DI and RO are the transmission and receiver line which connect to pin 14 (Tx) and pin 15 (Rx) of Serial3 respectively.

The PZEM 017 energy meter may comes with different rating of shunt, the setting can be set via **come-along PC Software (download from above). The upstream of signal cable and power cable need to be fused to protect from over-current that may cause fire risk.** The fuse rating must be lower than the current carry capacity of cable. It is highly recommend to use ferrule lugs at cable terminals.

You need the connector that can secure cables and isolate from accidental touch. Get the **fast connector** at our affiliate link ***here*** !!!



You need a **fuse** to protect your safety and cables!! You can get it ***here*** !! The ampere rating depends on your cable. Kindly refer to ampacity of your cable and application.

Ferrule Lugs and Crimper Set

Ferrule lugs are used at terminals for tight and secure connection. Besides, it also prevent stranded cables accidentally touching adjacent power line that may cause short circuit. Grab 1 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 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 is 2 **source codes file attached** which one is with Arduino Mega and another one is with Arduino UNO. Both are with LCD display shield function. By default, the LCD Shield button function has occupied Analog pin A0. I have used it for Reset Energy purpose and to go to next page since 1 display cannot show all values. To reset energy, press Select Button and Left Button to proceed.

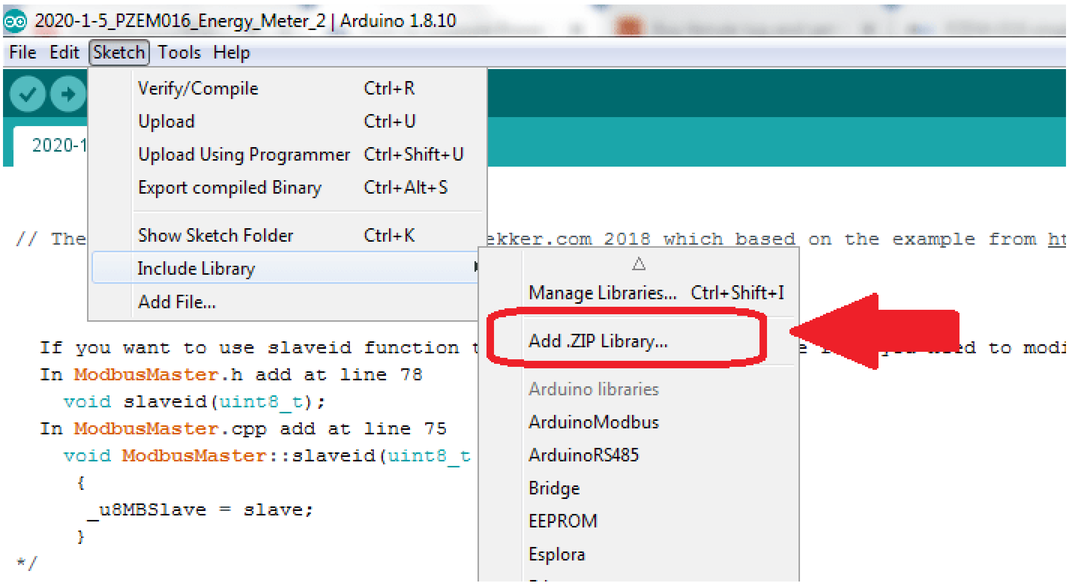
First Thing First

The core of this code is from EvertDekker.com. I just applied and amended it to match with the LCD Display shield. I **have spent weeks to make it work** and have been searching for workable code but non of them seems working for me. All the time the Arduino failed to communicate with the device even from the web page above. It definitely shows mistake somewhere.

Finally I manage to find out the main mistake that many people out there just skipped the little thing and did not tell you. **You need to set the device address FIRST**. For a successful RS485 communication, you need to send data to correct device address (the meter). Reference from many websites, we thought that device address was 0x01 by default but I also have been changing a lot of device address but the protocol can support up to about 250 device addresses which I don't think I can check 1 by 1. Luckily there is code to change the device address by sending broadcast command to all. I then just need to change the address to 0x01 at first start. Then everything goes smoothly.

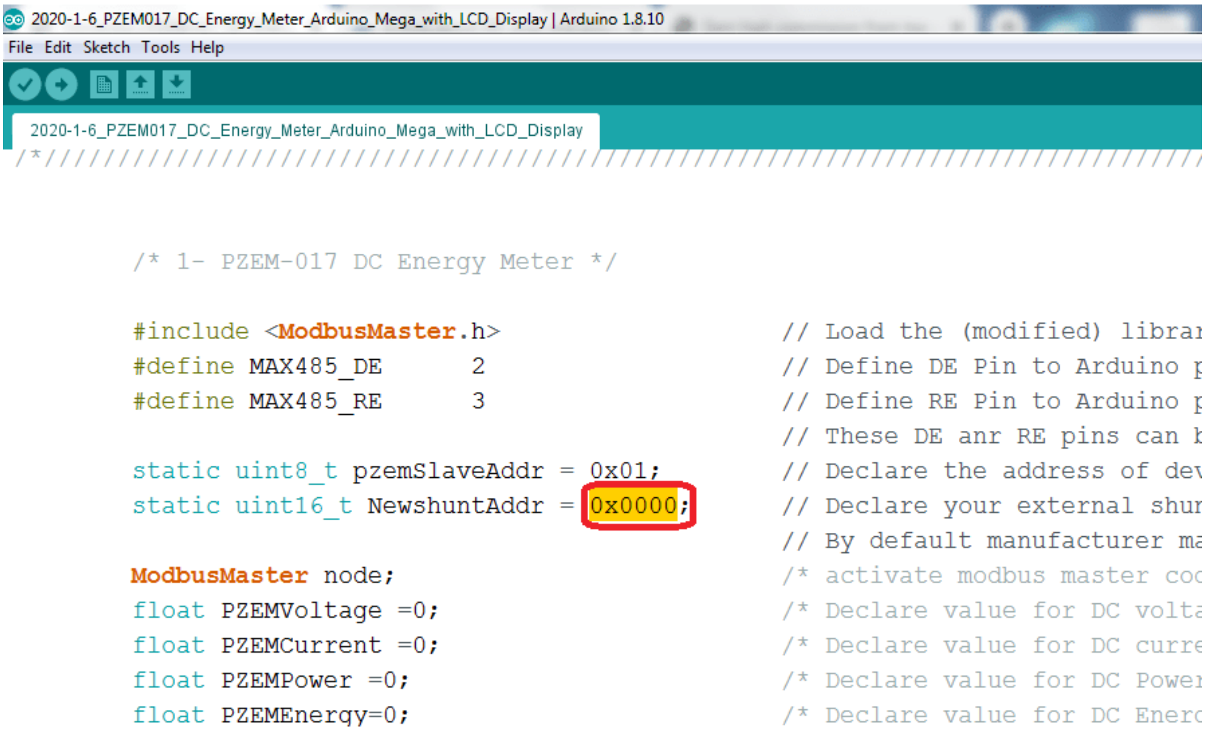
I have make this code : **changeAddress(0XF8, 0x01); active all the time**. Whenever you uploaded the code or whenever the arduino board got its power back on, it will change the address to 0x01. 0XF8 is the broadcast address and the device will change to address 0x01. Once the device has been successful changed address, you can put back to double slash symbol “//” to make the code inactive. However, I believe you can leave it ON with no issue.

Second thing before you upload the code, you need to install **(Modified) Modbus Master Library** . If you have installed the original ModbusMaster.h library from library management, kindly look into the **folder and delete it or it will clash with this library**. The Modified Modbus Master Library I get it from ***EvertDekker.com*** and I have uploaded the zip file ***here*** so that you can download and install it. When you open the .ino file or the Arduino IDE software, go to Sketch > Include Library > Add.ZIP Library…. and locate the zip file. Make sure you download it to your PC first (preferably download at desktop)


2020-1-5_PZEM016_Energy_Meter_2 | Arduino 1.8.10
File Edit Sketch Tools Help
2020-1
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
ArduinoModbus
ArduinoRS485
Bridge
EEPROM
Esplora

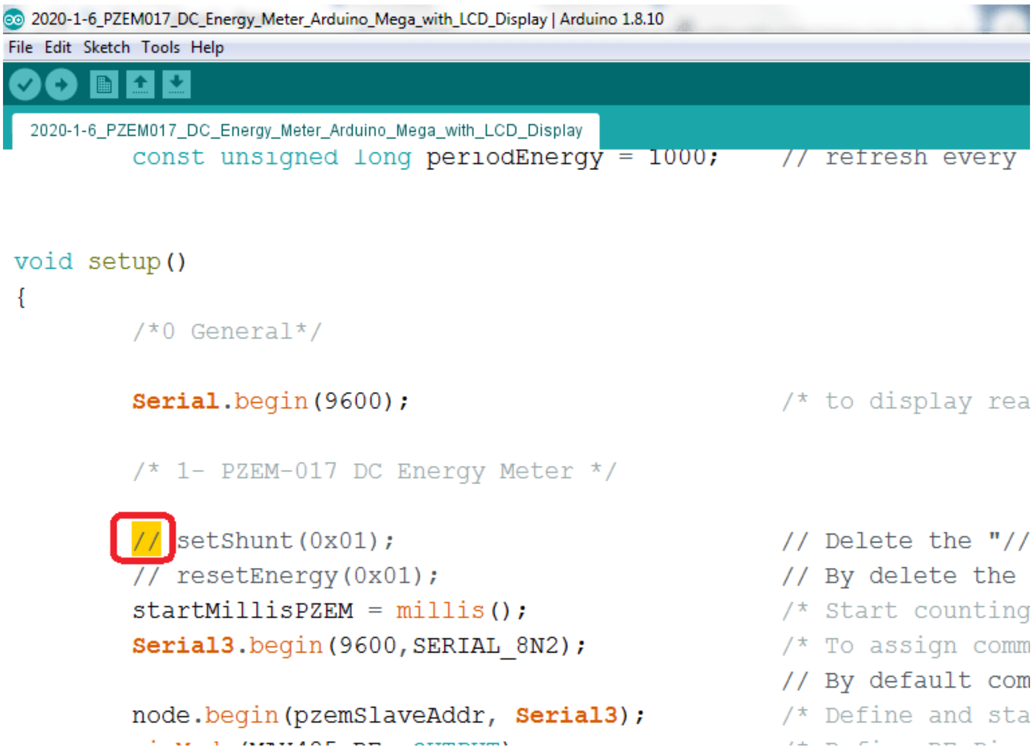
```
// The                                     ekker.com 2018 which based on the example from ht


  If you want to use slaveid function t                                    e                    d to modi
  In ModbusMaster.h add at line 78
    void slaveid(uint8_t);
  In ModbusMaster.cpp add at line 75
    void ModbusMaster::slaveid(uint8_t
     {
      _u8MBSlave = slave;
     }
*/
```

Another thing to set is the shunt setting based on your external shunt rating. The external shunt rating are 50A, 100A, 200A and 300A. I believe manufacturer has set for you in the factory. But if they don’t, no problem, you can also set inside the code. Just look for shunt setting, change the code 0x0000 (default 100A) to other value based on your external shunt setting. 0x0001 for 50A, 0x0002 for 200A and 0x0003 for 300A shunt.


2020-1-6_PZEM017_DC_Energy_Meter_Arduino_Mega_with_LCD_Display | Arduino 1.8.10
File Edit Sketch Tools Help
2020-1-6_PZEM017_DC_Energy_Meter_Arduino_Mega_with_LCD_Display

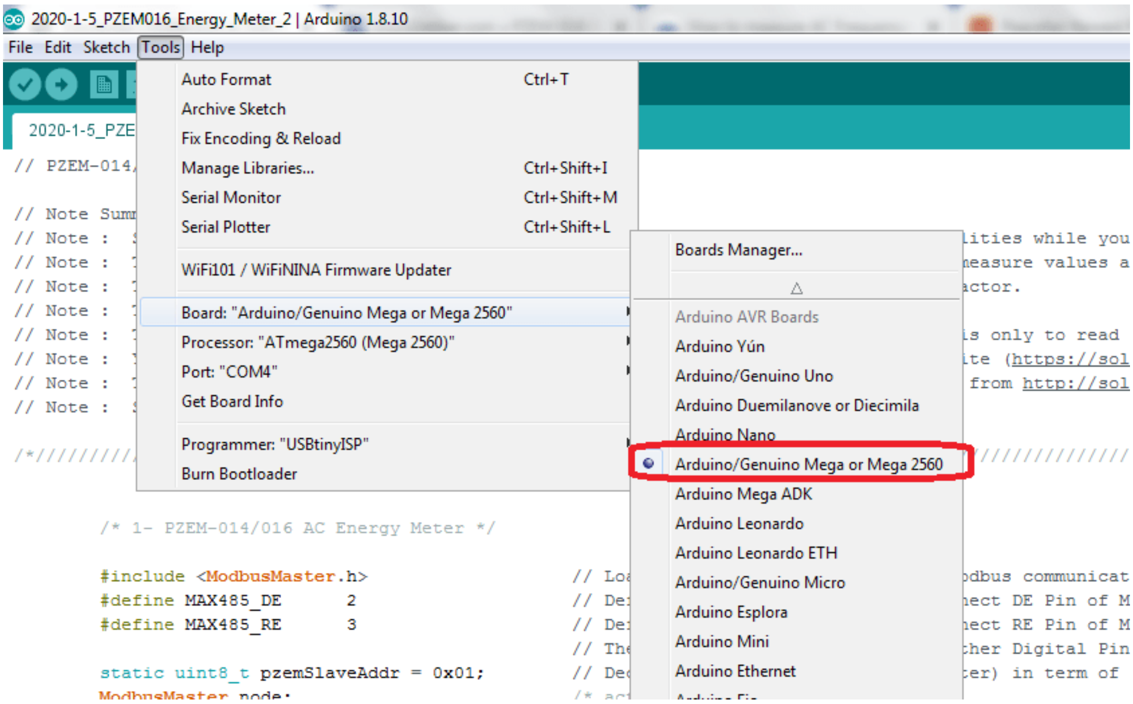
```
/*//////////////////////////////////////////////////////////////////////////////

        /* 1- PZEM-017 DC Energy Meter */

        #include <ModbusMaster.h>                         // Load the (modified) librai
        #define MAX485_DE      2                          // Define DE Pin to Arduino p
        #define MAX485_RE      3                          // Define RE Pin to Arduino p
                                                          // These DE anr RE pins can k
        static uint8_t pzemSlaveAddr = 0x01;              // Declare the address of dev
        static uint16_t NewshuntAddr = 0x0000;            // Declare your external shur
                                                          // By default manufacturer ma
        ModbusMaster node;                                /* activate modbus master coc
        float PZEMVoltage =0;                             /* Declare value for DC volta
        float PZEMCurrent =0;                             /* Declare value for DC curre
        float PZEMPower =0;                               /* Declare value for DC Powei
        float PZEMEnergy=0;                               /* Declare value for DC Enerc
```

After you set the value, you have to un-comment another code named **setShunt(0x01) to allow the code to active.** Just delete the “//” in front of code **setShunt(0x01);**


2020-1-6_PZEM017_DC_Energy_Meter_Arduino_Mega_with_LCD_Display | Arduino 1.8.10
File Edit Sketch Tools Help
2020-1-6_PZEM017_DC_Energy_Meter_Arduino_Mega_with_LCD_Display

```
        const unsigned long periodEnergy = 1000;    // refresh every


void setup()
{
        /*0 General*/

        Serial.begin(9600);                          /* to display rea

        /* 1- PZEM-017 DC Energy Meter */

        //setShunt(0x01);                            // Delete the "//
        // resetEnergy(0x01);                        // By delete the
        startMillisPZEM = millis();                  /* Start counting
        Serial3.begin(9600,SERIAL_8N2);              /* To assign comm
                                                     // By default com
        node.begin(pzemSlaveAddr, Serial3);          /* Define and sta
```

For this example, we are using **Arduino Mega 2560** board, make sure you have changed the board. When in the .ino file, go to **Tools> Board: > Arduino Mega 2560**. Compile and upload and you are done !!



Board: "Arduino/Genuino Mega or Mega 2560"

Processor: "ATmega2560 (Mega 2560)"

Port: "COM4"

Programmer: "USBtinyISP"

● Arduino/Genuino Mega or Mega 2560

```
/* 1- PZEM-014/016 AC Energy Meter */

#include <ModbusMaster.h>
#define MAX485_DE      2
#define MAX485_RE      3

static uint8_t pzemSlaveAddr = 0x01;
```

Warning !!

<u>Start the Arduino board first before Power from measurement</u>

You are not recommended to provide measurement power before Arduino is started or LCD is displayed. You also cannot reset the Arduino board during measurement. If you want to reset Arduino Board (by pressing Reset button), make sure the energy meter measurement power source is disconnected or else it might have error. If you forget to turn off by mistake, the LCD will go blank. Just turn off measurement power source and press Reset button again. Once reset and LCD display back to normal then turn on measurement power supply.

<u>Unplug Tx and Rx cable during code uploading for Arduino UNO</u>

If you want to use Arduino UNO replacing Arduino Mega, kindly read the following. Since arduino UNO sharing the same Serial port to communicate with Energy Meter and to communicate with PC. It is best to unplug the Tx and Rx pin so that during code uploading, the port is only dealing with PC while Energy Meter communication is disconnected. Once code is uploaded completely, you may connect both cable in. Optionally you can disconnect the measurement power source so that it don’t have the 5V power for communication with Arduino board.