**DIGITAL MULTIMETER USING ARDUINO NANO**

**MINOR PROJECT REPORT 2020**

*Submitted in partial fulfilment of the*

*Requirements for the award of the degree*

*Of*

**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted by:

**ROHAN BANSAL (41476802817/ECE3)**

**SAKSHAM KHATRI (42076802817/ECE3)**

**YADNESH HEMANT REVANKAR (43176802817/ECE3)**

Under the guidance of:

**Ms. Raman Dand**



**Department of Electronics and Communication**

**Guru Tegh Bahadur Institute of Technology**

**Guru Gobind Singh Indraprastha University**

**Dwarka, New Delhi**

**Year 2017-2021**

Table of Contents

1. **DECLARATION**
2. **ACKNOWLEDGEMENT**
3. **INTRODUCTION…………………………………………………………………5**

* MULTIMETER……………………………………………………………..5
* FUNCTIONAL DESCRIPTION……………………………………………5
* BLOCK DIAGRAM………………………………………………………...6

1. **PROJECT DESCRIPTION………………………………………………………7**

* OBJECTIVE
* SCHEMATIC
* COMPONENTS LIST

1. **HARDWARE USED………………………………………………………………9**

* ARDUINO NANO………………………………………………………….9
* ADC MODULE……………………………………………………………..10
* OLED DISPLAY……………………………………………………………11
* CURRENT SENSOR……………………………………………………….12
* CHARGING MODULE…………………………………………………….13

1. **SOFTWARE USED……………………………………………………………….14**

* ARDUINO IDE

1. **IMPLEMENTATION…………………………………………………………….16**

* VOLTAGE MEASUREMENT…………………………………………….16
* RESISTANCE MEASUREMENT…………………………………………16
* CAPACITANCE MEASUREMENT………………………………………17
* INDUCTANCE MEASUREMENT………………………………………..18
* CURRENT MEASUREMENT…………………………………………….19

1. **COMPLETED PROJECT CIRCUIT…………………………………………….…20**
2. **CONCLUSION…………………………………………………………………….21**
3. **REFERENCES…………………………………………………………………….22**

**DECLARATION**

We hereby declare that all the work presented in the **Minor Project Report 2020** entitled “**DIGITAL MULTIMETER USING ARDUINO NANO**” in the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering, Guru Tegh Bahadur Institute of Technology, Guru Govind Singh Indraprastha University, New Delhi is an authentic record of our own work.

Submitted by:

**ROHAN BANSAL (41476802817/ECE3)**

**SAKSHAM KHATRI (42076802817/ECE3)**

**YADNESH HEMANT REVANKAR (43176802817/ECE3)**

**ACKNOWLEDGEMENT**

We would like to express our great gratitude towards our supervisor**, Ms. Raman Dand** who has given us support and suggestions. Without her help, we could not have presented this project up to the present standard. We also take this opportunity to give thanks to all others who gave me support for the project or in other aspects of our study at Guru Tegh Bahadur Institute of Technology.

Submitted by:

**ROHAN BANSAL (41476802817/ECE3)**

**SAKSHAM KHATRI (42076802817/ECE3)**

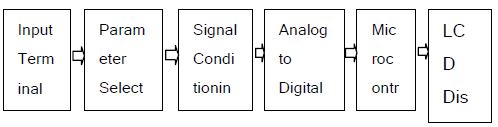
**YADNESH HEMANT REVANKAR (43176802817/ECE3)**

**INTRODUCTION**

**MULTIMETER:** A Multimeter or a Multitester, also known as a VOM (volt-ohm-milliammeter), is an electronic measuring instrument that combines several measurement functions in one unit. A typical multimeter can measure voltage, current, and resistance. Analog Multimeters use a microammeter with a moving pointer to display readings. Digital Multimeters (DMM, DVOM) have a numeric display, and may also show a graphical bar representing the measured value. A multimeter can be a hand-held device useful for basic fault finding and field service work, or a bench instrument which can measure to a very high degree of accuracy. A multimeter is a combination of a multi range DC voltmeter, multi range AC voltmeter, multi range ammeter, multi range ohmmeter and other electrical parameter measuring circuit.

**FUNCTIONAL DESCRIPTION:**

Proposed scheme of the Microcontroller based Digital Multi-meter is as shown in block diagram having following section. –



**Fig.1:** Block Diagram

1. **Parameter Selection Circuit:**

This section is responsible for the selection of the desired parameter and let the micro-controller know about what action should be taken by microcontroller, as here different parameters are required to be measured one at a time.

1. **Signal Conditioning Circuit:**

The purpose of this circuit is to process different type of parameter so that they can be measured by microcontroller. This section has many different kinds of circuit like amplifiers, attenuators, frequency multiplier, precision rectifier, RC oscillators etc. All these are required to convert our input parameter to a state which can be measured by microcontroller without any problem.

1. **Microcontroller Arduino Nano:**

Microcontroller is used to read different kind of parameter at its input through the selection circuit. The main operation of microcontroller is to read data from ADC or any other input as selected by selection circuit. The measured data of whole scheme is processed before displaying it on LCD in a user understandable format.

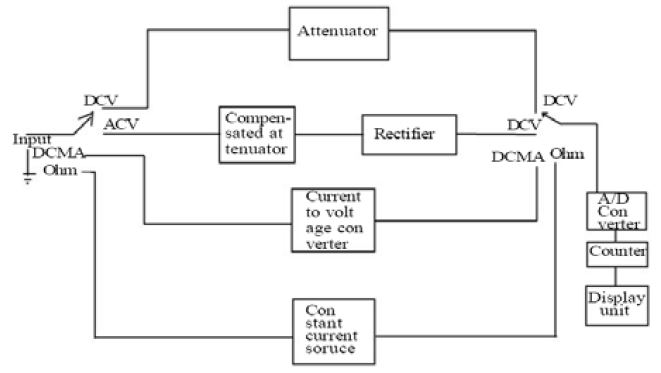
1. **LCD/OLED Display:**

The LCD display is used to display the selected operation and measured value of selected parameter at input terminal.

1. **Power Supply:**

Proposed scheme requires a DC power supply for the operation of our system. This power supply can be made by using a rechargeable battery.

**BLOCK DIAGRAM OF DIGITAL MULTIMETER:**

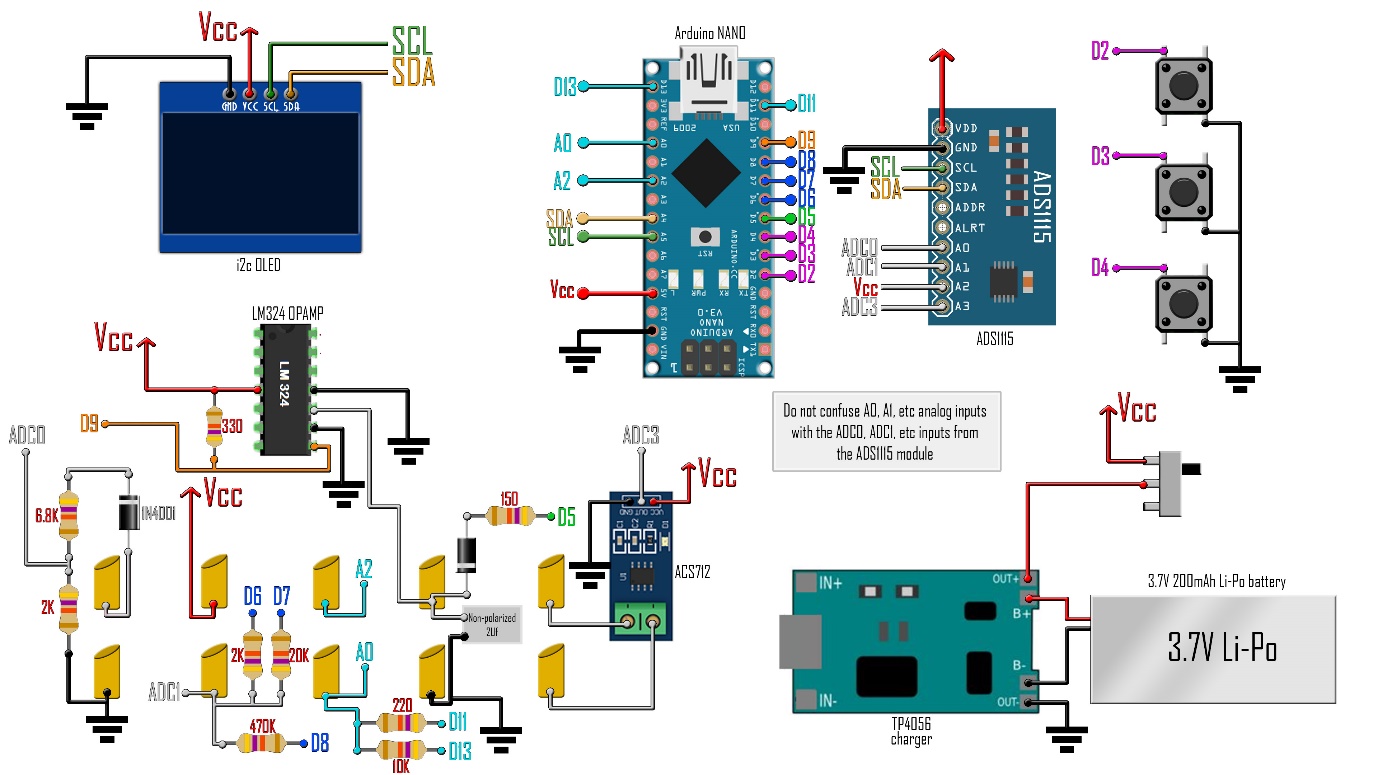
****

**Fig.2:** Digital Multimeter Block Diagram

**PROJECT DESCRIPTION**

**OBJECTIVE:** The objective of this project is to create a low-cost commercial grade Digital Multimeter with the use of Microcontroller Arduino Nano Board.

**SCHEMATIC:**

****

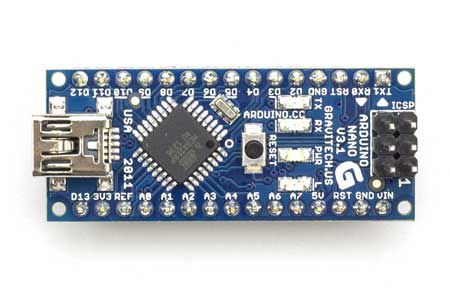
**Fig.3:** Project Schematic

**COMPONENTS USED:**

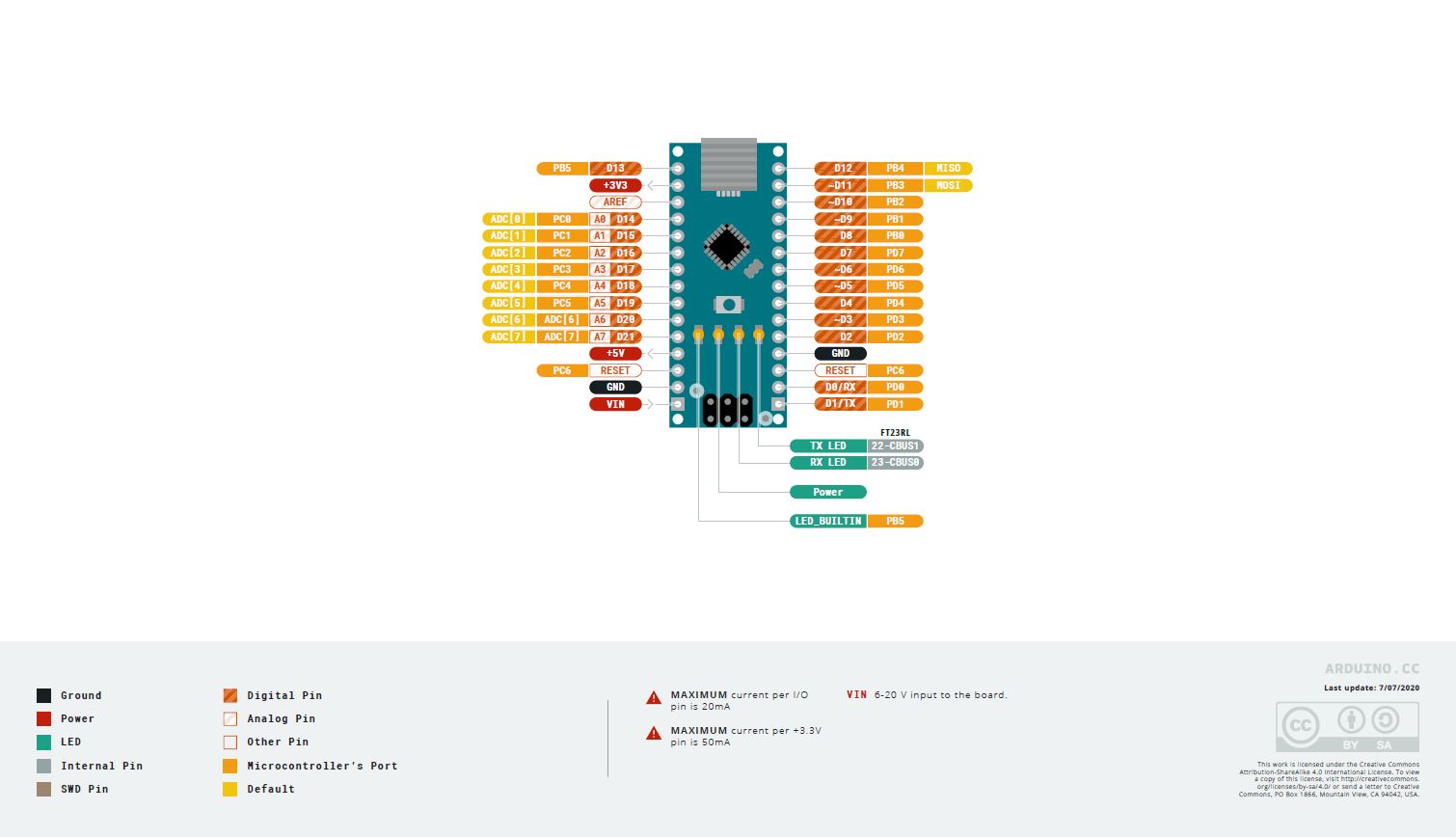
* 1 x Arduino NANO/UNO
* 1 x ADS1115 sensor
* 1 x i2c OLED screen
* 1 x TP4056 charging module
* 1 x ACS712 current sensor
* 1 x LM324 OPAMP
* 10 x female bullet connectors
* 2 x male bullet connectors
* 3 x push buttons
* 1 x sliding switch
* 1 x 3.7V Li-Po battery
* RESISTORS: 1x150, 1x220, 1x330, 2x2K, 1x6.8K, 1x10K, 1x20K, 1x470k
* 2 x 1n4001 diode
* 2 x 1µF nonpolarized capacitor
* wire, solder, soldering-iron, cardboard case

**HARDWARE USED**

1. **Arduino Nano:** The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one.



**Fig.4:** Arduino Nano



**Fig.5:** Nano Pin Configuration

**Power:** The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

**Memory:** The ATmega328 has 32 KB, (also with 2 KB used for the bootloader. The ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

**Input and Output:** Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e., 1024 different values). By default, they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Analog pins 6 and 7 cannot be used as digital pins. Additionally, some pins have specialized functionality:

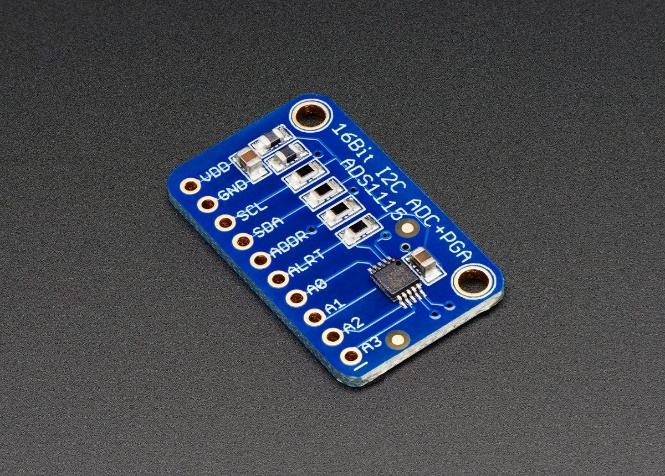
I2C: A4 (SDA) and A5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

1. **ADS1115 ADC Module:** The ADS1115 is an external digital analog converter (ADC) that we can connect to a processor like Arduino to measure analog signals. Arduino has internal ADCs that we use when we use the Arduino analog inputs. On the Arduino Uno, Mini and Nano models, we have 6 ADC of 10 bits. The ADS1115 provides 4 16-bit ADCs, 15 for the measurement and one last for the sign. The ADS1115 is connected by I2C, so it is easy to read. It has 4 addresses, which is chosen by connecting the ADDRESS pin. The interest of using an ADC such as the ADS1115 is to obtain greater precision, in addition to freeing the processor from this burden. In addition, in certain configurations, it is possible to measure negative voltages.



**Fig.6:** ADC Module

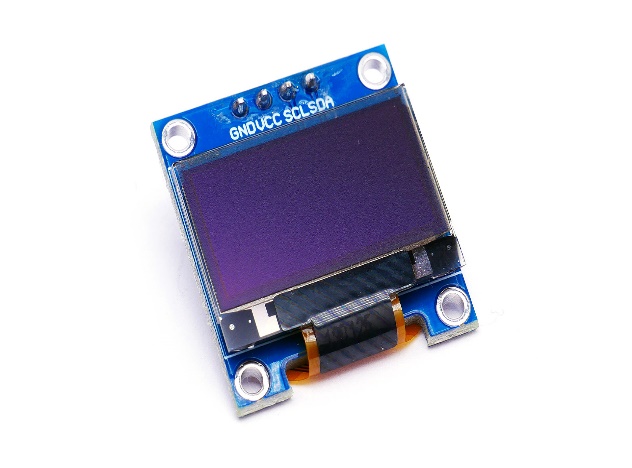
1. **I2C OLED Display:** This 1.3″ I2C OLED Display is an OLED monochrome 128×64 dot matrix display module with I2C Interface. It is perfect when you need an ultra-small display. Comparing to LCD, OLED screens are way more competitive, which has a number of advantages such as high brightness, self-emission, high contrast ratio, slim outline, wide viewing angle, wide temperature range, and low power consumption. It is compatible with any 3.3V-5V microcontroller, such as Arduino. Pin Definition:

1.GND: Power ground

2.VCC: Power positive

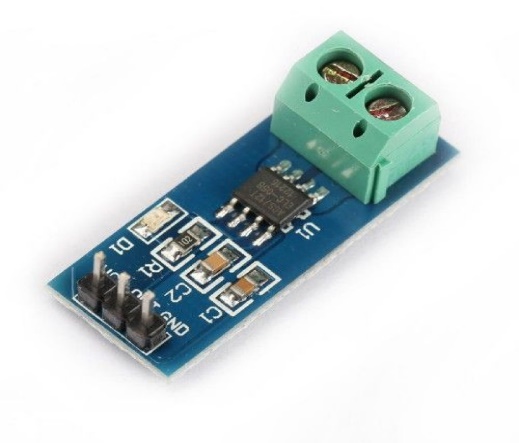
3.SCL: Clock wire

4.SDA: Data wire



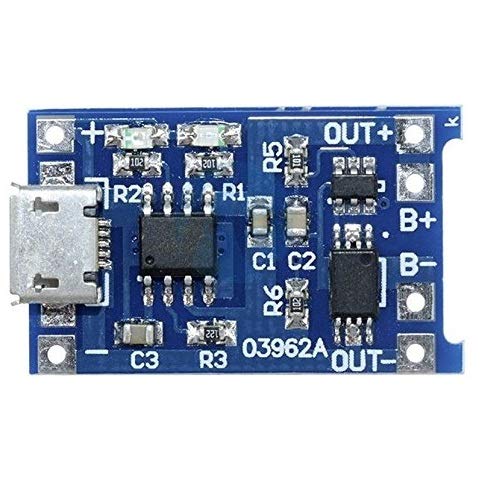
**Fig.7:** OLED Display

1. **ACS712 Current Sensor:** The ACS712 Module uses the famous ACS712 IC to measure current using the Hall Effect principle. The module gets its name from the IC (ACS712) used in the module. These ACS712 module can measure current AC or DC current ranging from +5A to -5A, +20A to -20A and +30A to -30A. We have to select the right range for a project since we have to trade off accuracy for higher range modules. This modules outputs Analog voltage (0-5V) based on the current flowing through the wire; hence it is very easy to interface this module with any microcontroller.



**Fig.8:** Current Sensor

1. **TP4056 Charging Module:** The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter. No blocking diode is required due to the internal PMOSFET architecture and have prevent to negative Charge Current Circuit. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor. The TP4056 automatically terminates the charge cycle when the charge current drops to 1/10th the programmed value after the final float voltage is reached.



**Fig.9:** Charging Module

**SOFTWARE USED**

**ARDUINO IDE:** The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them.

**Writing Sketches:** Programs written using Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

**Sketchbook:** The Arduino Software (IDE) uses the concept of a sketchbook: a standard place to store your programs (or sketches). The sketches in your sketchbook can be opened from the File > Sketchbook menu or from the Open button on the toolbar. The first time you run the Arduino software, it will automatically create a directory for your sketchbook.

**Uploading:** Before uploading your sketch, you need to select the correct items from the Tools > Board and Tools > Port menus. On Windows, it's the COM1 or COM2 (for a serial board) or COM4, COM5, COM7, or higher (for a USB board) - to find out, you look for USB serial device in the ports section of the Windows Device Manager. Once you've selected the correct serial port and board, press the upload button in the toolbar or select the Upload item from the Sketch menu. Current Arduino boards will reset automatically and begin the upload. On most boards, you'll see the RX and TX LEDs blink as the sketch is uploaded. The Arduino Software (IDE) will display a message when the upload is complete, or show an error.

**Libraries:** Libraries provide extra functionality for use in sketches, e.g., working with hardware or manipulating data. To use a library in a sketch, select it from the Sketch > Import Library menu. This will insert one or more #include statements at the top of the sketch and compile the library with your sketch. Because libraries are uploaded to the board with your sketch, they increase the amount of space it takes up. If a sketch no longer needs a library, simply delete its #include statements from the top of your code.

**Some important libraries to be included in this project are: -**

1. **ADS1x15 ADC module library**
2. **GFX graphics core library**
3. **OLED display library**

**IMPLEMENTATION**

In this project we have implemented a circuit which will measure 5 different parameters which are as follows: -

**Voltage Measurement:**

A microcontroller cannot understand analog voltage directly. That is why we have to use an Analog to Digital Converter or ADC (in our case, ADS1115).

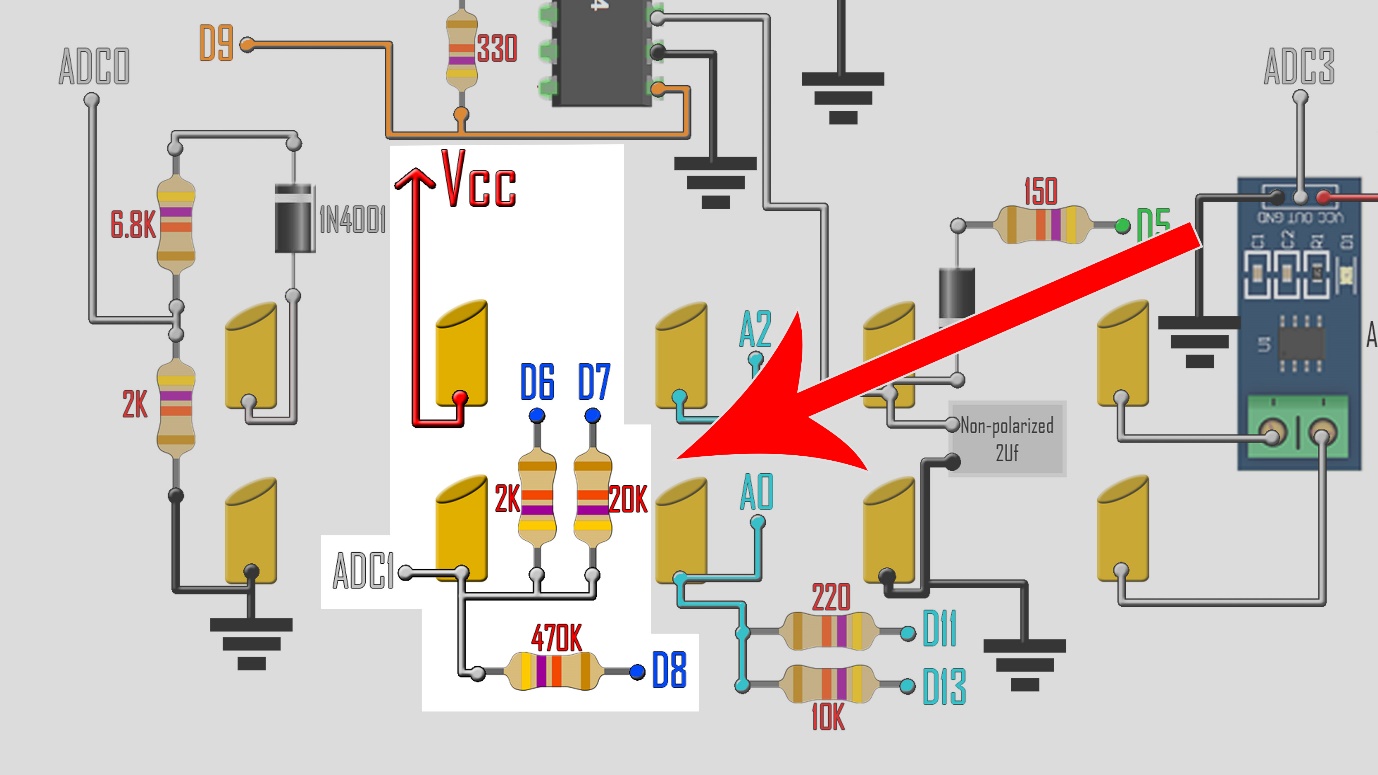
Using ADS1115 ADC converter, the values it will measure will be of high precision.

The circuit is connected in Voltage divider configuration on the input side in order to measure voltage in the range 0V to 20V. This is done to avoid extensive damage to the ADC module will burn out.

**Resistance Measurement:**

We use a basic voltage divider circuit to calculate the resistance. As we know, a voltage divider is made of two resistances (R1 and R2) in series. The output voltage in the middle point is [R2/(R1+R2)]Vin. Using this formula and knowing the value of one of the two resistors and measuring the Vout it is very easy to calculate the resistance of the second resistor.

On our schematic, the resistance measurement is made with the 2k, 20K and 470K resistors connected on pins D6, D7 and D8. By that we have 3 different scales. If we set D6 as OUPTUT and set to LOW, that will our GND for the voltage divider. The other pins, D7 and D8 are set to INPUT so they have high impedance. So, our volage divider is made out of the 2K resistor and the unknown resistor. We measure the voltage on the ADC1 pin and using the formula we get the resistance. We do that for all the scales. We have to make sure to measure the values of the 2k, 20K and 470K resistors so that their exact values are known and put these values later in the code.



**Fig.10:** Voltage divider circuit for Resistance Measurement

**Capacitance Measurement:**

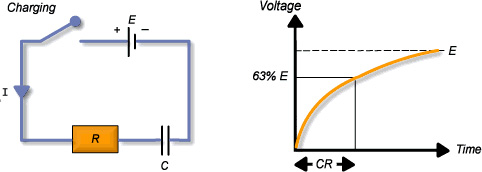
Capacitance is a measure of the ability of "something" to store electrical charge. Arduino capacitance measurement relies on the basic property of capacitors- the time constant. The time constant of a capacitor is defined as the time it takes for the voltage across the capacitor to reach 63.2% of its voltage when fully charged. Larger capacitors take longer to charge, and therefore have larger time constants. An Arduino can measure capacitance because the time a capacitor takes to charge is directly related to its capacitance by the next equation:

**TC = R x C**

**TC is the time constant of the capacitor (in seconds).**

**R is the resistance of the circuit (in Ohms).**

**C is the capacitance of the capacitor (in Farads).**

****

**Fig.11:** Time constant of capacitor

We charge the capacitor using one of the Arduino pins. Then we discharge it through a resistor. The formula tells us that the capacitance value is equal to the time it took to reach 63.2% of the fully charged voltage divided by the resistor value.

So, when capacitance mode is selected, we charge the capacitor, discharge it and count time. When the capacitor reaches 63.2% of Vcc, we stop the time counter and calculate the capacitance value using the formula.

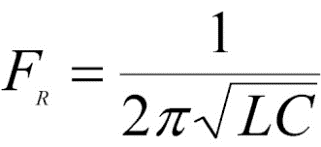
**Inductance Measurement:**

In this circuit we use an LC circuit. An inductor in parallel with a capacitor is called an LC circuit, and it will electronically "ring" like a bell. Well regardless of the frequency or how hard a bell is struck, it will ring at its resonating frequency. We will electronically strike the LC bell, wait a bit to let things resonate, then take a measurement. There is usually some amount of internal resistance so this is generally an RLC circuit.

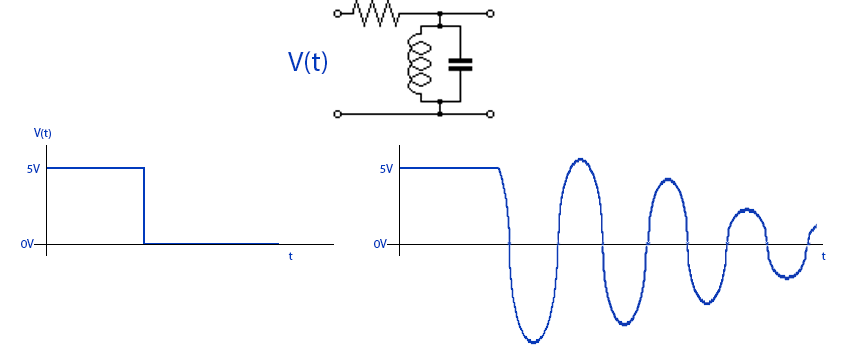
We know that microcontrollers are not good at analysing analog signals. Therefore, we use LM324 op-amp. As soon as the voltage on the LC circuit becomes positive, the LM324 will be floating, which can be pulled high with a pull up resistor. When the voltage on the LC circuit becomes negative, the LM324 will pull its output to ground. It can be noticed that the LM324 has a high capacitance on its output, which is why a low resistance pull up is used.

We will apply a pulse signal to the LC circuit. In this case it will be 5 volts from the Arduino. This pulse will make the circuit to resonate creating a cushioned sinusoidal signal oscillating at the resonant frequency.

The resonant frequency is related with this equation:



We count the time between each pulse of the resonance frequency. If we know the frequency value and also the used capacitance (in this case 2µF), we can get the inductance value and print that to the OLED screen.



**Fig.12:** Pulse signal to the LC circuit

**Current Measurement:**

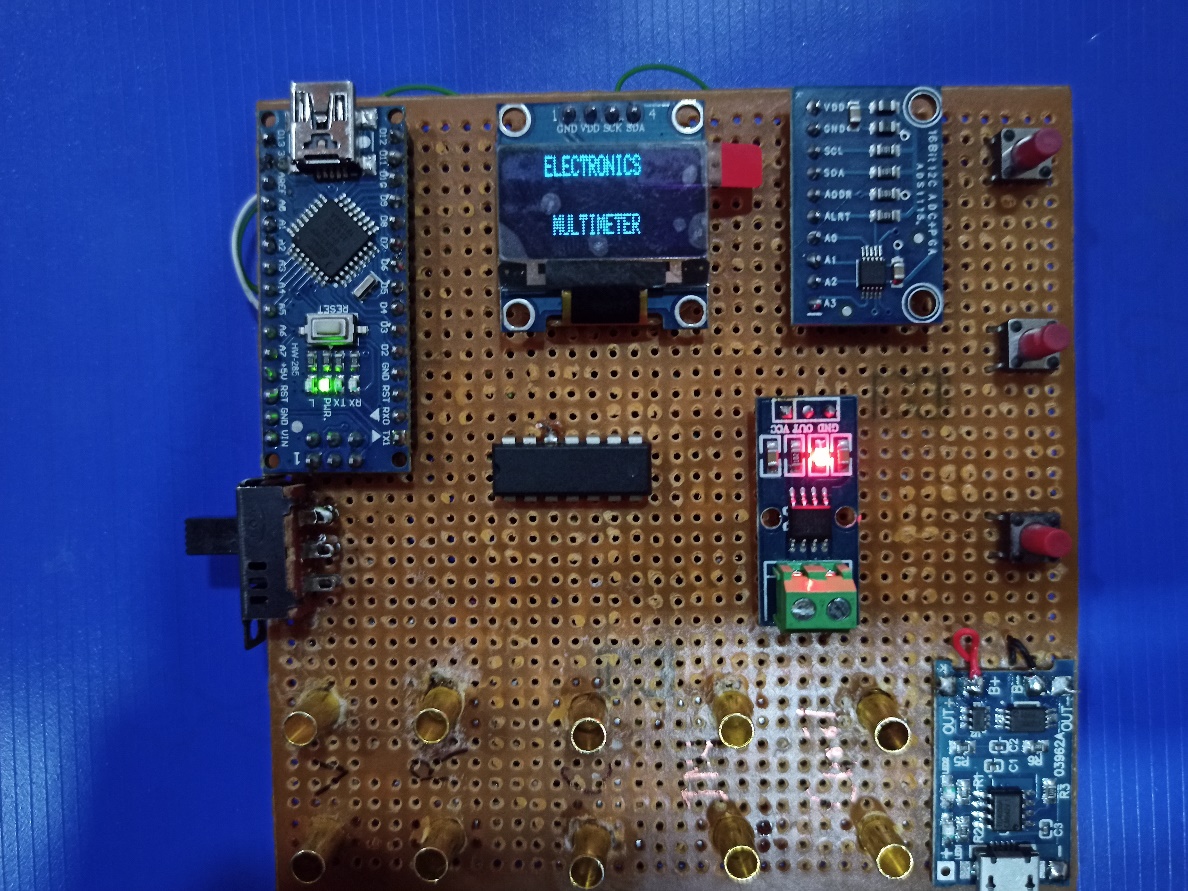
To measure this parameter, we use the ACS712 module. The ACS712 current sensor is an economical solution for measuring current, it works internally with a Hall effect sensor that detects the magnetic field produced by induction of the current flowing through the line being measured.

The sensor gives us a voltage output proportional to the current, in our case, we used ACS712-05A having range of -5A to 5A and resolution of 0.185 V/A. Next is to use the given formula to compute the current:

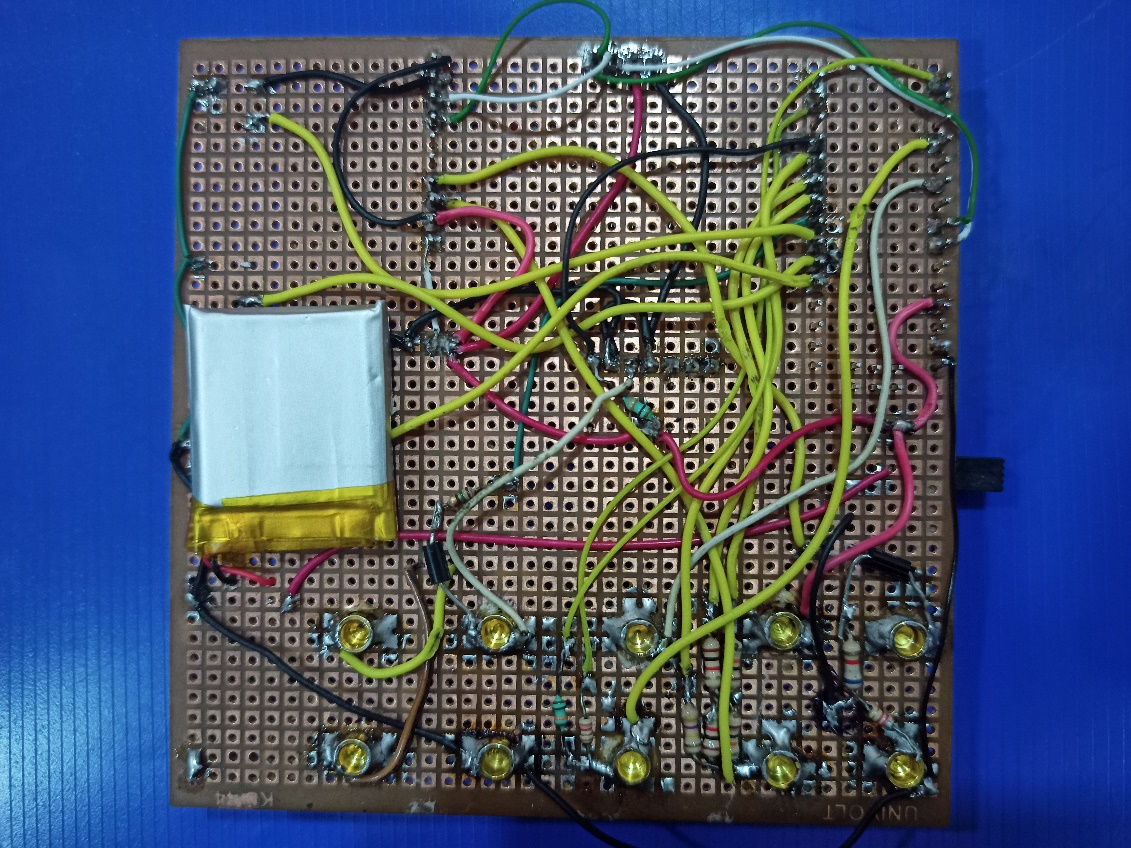
**I=(SensorVoltage-2.5)/resolution**

**COMPLETED PROJECT CIRCUIT**

Front View:



Back View:



**CONCLUSION**

**Arduino IDE Code:**

<https://github.com/rohan1110/Arduino_Multimeter>

**Distinguishing Feature:**

The distinguishing factor in a general purpose commercial multimeter and the multimeter created in this project is that we have included the measurement of 2 additional parameters: **Capacitance and Inductance.**

**Advantages:**

* Cost effective
* Smaller footprint
* Higher accuracy and precision
* Easy to use because of button architecture
* Lower carbon footprint due to rechargeable functionality.

**Drawbacks:**

Measurement of AC current and voltages is not possible.

**REFERENCES**

* [1]Volume-6, Issue-2, March-April 2016 “International Journal of Engineering and Management Research” Page Number: 273-279
* Development of a Digital Multimeter: A Low-Cost Design Approach”
* [https://www.arduino.cc](https://www.arduino.cc/)
* [www.circuitbasics.com](http://www.circuitbasics.com)