International Journal of Information, Engineering & Technology
Volume 12, Issue 4, PP 68-77, ISSN: 2360-9194, December, 2023
DOI: 4272-1454-381-247
Double Blind Peer Reviewed International Research Journal
**Journal Series: Global Academic Research Consortium (garc)**
arcnjournals@gmail.com
https://arcnjournals.org

# Design and Implementation of Digital Voltmeter (0 – 10v) Range Using Arduino Uno Using Serial Monitor as a Display

**Adam Bababe Bukar**
Department of Computer Engineering, Ramat Polytechnic Maiduguri, Nigeria

***Abstract**: A simple digital voltmeter can easily be made using an Arduino uno and a serial monitor as display and It's relatively simple to use an Arduino to measure voltages. Measuring voltage is quite easy using any microcontroller as compared to the measurement of current. A microcontroller cannot understand analog voltage directly, thus an analog-to-digital converter (ADC) is used. The Arduino has several analog input pins that connect to ADC inside the Arduino. AT MEGA 328 which is the brain of the arduino has ten-bit ADC.This implies that it will read voltage from 0 to 5v into digital values from 0 to 1023 which gives approximately a resolution of 4.9mV per unit. A problem arises when the voltage to be measured exceeds 5v. This problem can be overcomed by using a voltage divider circuit which consists of two Resistors of different values connected in series. One end of the series connection is connected to the voltage to be measured and the other end to the ground. This project discusses how to measure a maximum voltage of 10v which is above the reference voltage by using arduino uno. In order to achieve this, the concept of voltage divider rule is applied which divides the input voltage so that the voltage actually input the Arduino is 5V or less. Two Resistors of $R_1$ and $R_2$ are used to create a 2.1 divider which allows us to measure voltages up to 10V.*

***Keywords:** Digital Voltmeter, Design, Divider Circuit, Implementation.*

## 1.0 INTRODUCTION

The design and implementation of digital voltmeters have gained significant attention in the field of electronics due to their versatility and accuracy. In this context, utilizing the Arduino Uno platform to create a digital voltmeter with a range of 0-10V offers an accessible and cost-effective solution for measuring voltage levels in various electronic circuits. Arduino Uno, an open-source microcontroller, provides a programmable and user-friendly interface, making it an ideal choice for developing customized measurement devices. By employing the Serial Monitor as a display, users can easily observe real-time voltage readings, enhancing the usability of the digital voltmeter. This project aligns with the growing demand for innovative and user-friendly measurement tools, contributing to the development of practical applications in electronics and automation.

A digital voltmeter, abbreviated as DVM, is a measuring instrument that displays the value of an AC and DC voltage directly in decimal numbers instead of a pointer deflection on a continuous scale. The data output from a digital voltmeter may be fed directly into memory devices for further computation and storage. DVM is an accurate and versatile instrument used in many laboratory applications. A digital voltmeter (DVM) attains the required measurement by converting the analog input signal into digital and when necessary by discrete-time processing of the converted values. The measurement result is presented in a digital form that can be coded as a decimal BCD code or a binary code the main factors that characterize DVMs are speed automatic operation and programmability in particular they presently offer the best combination of speed and accuracy If compared with other available voltage-measuring instruments moreover capability of automatic operations and programmability make DVMs very useful in application where flexibility high speed and computer controllability are required.

A typical application field is therefore that of automatically operated systems when a DVM is directly interfaced to a digital signal processing (DSP) and used to convert the analog input voltage into a sequence of sampled values it is usually called an analog-to-digital converter (ADC). In this paper we can use arduino's analog to digital converter (ADC) pins to convert analog voltage values into number representation that you can work with. An interface circuit is designed and embedded software is written with Arduino and ATmega 328 microcontroller board so that it functions as a digital voltmeter. The voltmeter software functions as a measured voltage on analog input channel 0 and display it on a serial monitor with accuracy of one decimal place.

The accuracy of an ADC is determined by the resolution. In the case of the arduino uno, there is a 10-bit ADC for doing your analog conversions. 10-bit means that the ADC can subdivide (or quantize) analog signal into $2^{10}$ different values which is equivalent to $2^{10} = 1024$. Hence, the arduino can assign a value from to 1023 for any analog value that you give it. Although it is possible to change the reference voltage, you'll be using the default 10v reference for the analog work. The reference voltage determines the max voltage that you are expecting, and therefore, the value that will be mapped to 1023.

## 2.0 REVIEW OF RELATED WORKS

A digital voltmeter (DVM) attains the required measurement by converting the analog input signal into digital and when necessary by discrete-time processing of the converted values. The measurement result is presented in a digital form that can be coded as a decimal BCD code or a binary code the main factors that characterize DVMs are speed automatic operation and programmability in particular they presently offer the best combination of speed and accuracy If compared with other available voltage-measuring instruments moreover capability of automatic operations and program-mability make DVMs very useful in application where flexibility high speed and computer controllability are required A typical application field is therefore that of automatically operated systems when a DVM is directly interfaced to a digital signal processing (DSP) and used to convert the analog input voltage into a sequence of sampled values it is usually called an analog-to-digital converter (ADC)

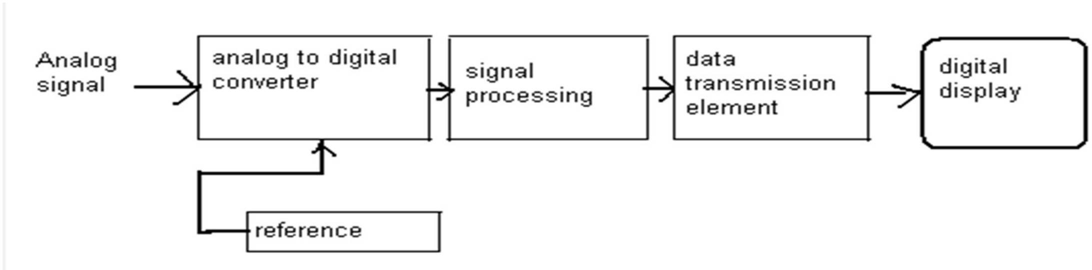



Figure 1.1: basic DVM diagram

A digital voltmeter, abbreviated as DVM, is a measuring instrument that displays the value of an AC and DC voltage directly in decimal numbers instead of a pointer deflection on a continuous scale. The data output from a digital voltmeter may be fed directly into memory devices for further computation and storage. DVM is an accurate and versatile instrument used in many laboratory applications.[1]

In this project we can use arduino's analog to digital converter (ADC) pins to convert analog voltage values into number representation that you can work with. The accuracy of an ADC is determined by the resolution. In the case of the arduino uno, there is a 10-bit ADC for doing your analog conversions. 10-bit means that the ADC can subdivide (or quantize) analog signal into 2^10 different values. If you do the math, you'll find that 2^10 = 1024. Hence, the arduino can assign a value from to 1023 for any analog value that you give it. Although it is possible to change the reference voltage, you'll be using the default 5v reference for the analog work. The reference voltage determines the max voltage that you are expecting, and therefore, the value that will be mapped to 1023. So, with a 5v reference voltage, putting 0v on an ADC pin returns a value of 2.5v returns a value of 512(half of 1023) ,and 5v returns a value of 1023. [2]

In this section the work related to the digital voltmeter is expressed. There are many projects implemented for digital voltmeter design. One of these projects used arduino Nano, resistor 10k ohm, resistor 1k ohm and also used Liquid Crystal Display (LCD) to display the voltage. But not use the potentiometer. The analog sensor on the Nano board senses the voltage on the analog pin and converts it into a digital format that can be processed by the microcontroller. Here, the input voltage is routed to an analog pin through a simple passive voltage divider, and with the values shown it can measure dc voltage in 0v to 55v range.[3]

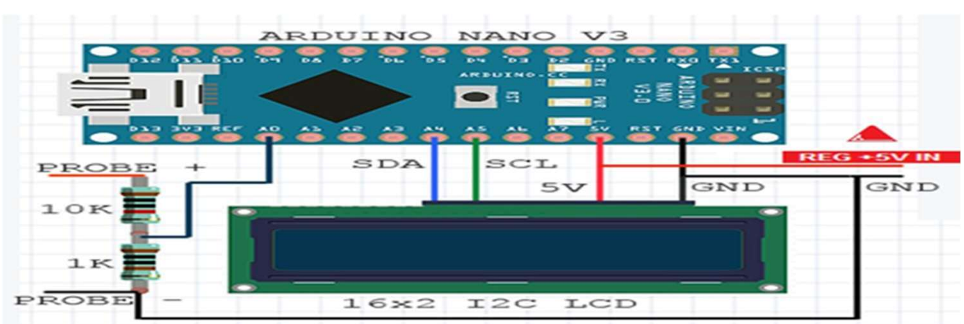



Figure 1.2: circuit digital voltmeter uno

## 2.1 VOLTAGE MEASUREMENT

Is one of the most basic measurements in the electronic measurement. Generally, an electronic measuring instrument is classified roughly by voltage measurement, current (or charge) measurement, and measurement of both (electric power and impedance). For example, the oscilloscope is a measuring instrument that displays the voltage value sensed by the tip of the probe on the screen as a time waveform, and it can be called a voltmeter in a broad sense because the physical value of the target measurement is the voltage. However, at present this is a measuring device focused on the high-speed response and sampling rate exceeding several GHz or more, and generally the precision and resolution of the voltage measurement is not high. The digital multi meter is the most basic ammeter and voltmeter, and various lineups like the hand-held type, bench top type etc. are available. It is most widely used for general voltage measurement because high accuracy and high resolution voltage measurement devices are available, though it does not match the oscilloscope with respect to speed. The electrometer has the function of current and voltage measurement similar to the digital multi meter. When focusing attention on the voltage measurement, it can be positioned with the voltmeter characterized by the high input resistance at the measuring terminal by comparing with the digital multi meter.

## 2.2 ARDUINO UNO 328P

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Shown in Figure 1.3 [4].

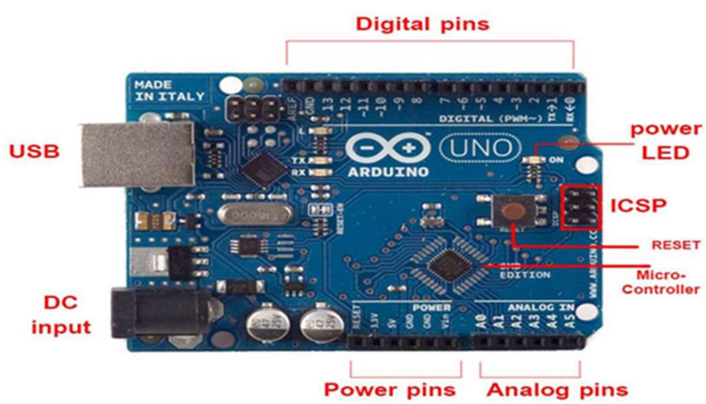



Figure 1.3: Arduino Uno 328p

It contains everything needed to support the microcontroller shown in figure 1.2; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started. [6]

The Uno 328p can be powered Vin the USB connection or with an DC supply of 6 to 20 volts.

Figure 1.4: USB and DC input

If supplied with less than 7V however, the 5V pin may supply less than five volts and the board may become unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.[6]

**The digital pins are as follows:**

**Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. On the Arduino Decimal, these pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip. On the Arduino BT, they are connected to the corresponding pins of the WT11 Bluetooth module. On the Arduino Mini and Lily Pad Arduino, they are intended for use with an external TTL serial module (e.g. the Mini-USB Adapter).

**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 attach Interrupt() function for details.

**PWM**: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function. On boards with an ATmega8, PWM output is available only on pins 9, 10, and 11.

**BT Reset**: 7. (Arduino BT-only) Connected to the reset line of the bluetooth module.

**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. On the Diecimila and LilyPad, 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.[7]


Figure 1.4: digital pins

**Analog Pins**: In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

**I2C**: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website.[7]


Figure 1.5: Analog pins

**Power Pins:** VIN (sometimes labelled "9V"). The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin. Note that different boards accept different input voltages ranges, please see the documentation for your board. Also note that the LilyPad has no VIN pin and accepts only a regulated input.

The regulated power supply used to power the microcontroller 5V and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

**V3**: (Diecimila-only) A 3.3 volt supply generated by the on- board FTDI chip.

**GND**: Ground pins.[7]

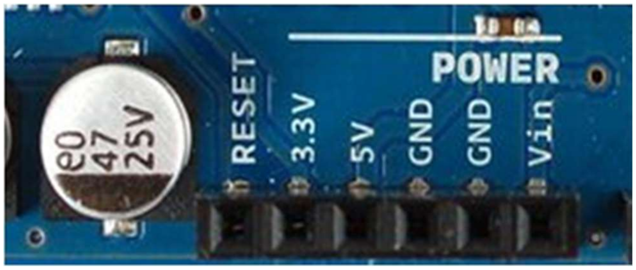



Figure 1.7: power pins

**Other Pins**: AREF: Reference voltage for the analog inputs. Used with analog Reference.

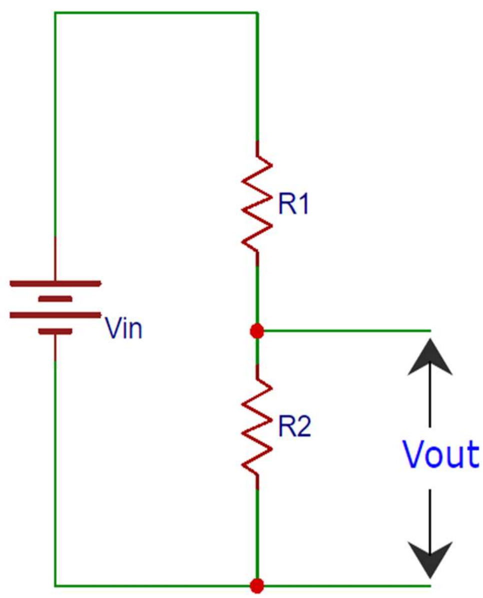
**Reset**: (Diecimila-only) Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. [7].

**3.0 METHODOLOGY**

This project discusses how to measure a maximum voltage of 10v which is above the reference voltage by using arduino uno. In order to achieve this, the concept of voltage divider rule is applied which divides the input voltage so that the voltage actually input to the Arduino is 5V or less. Two Resistors of $R_1$ *and* $R_2$ are used to create a 2:1 divider which allows us to measure voltages up to 10 V.

### 3.1 VOLTAGE DIVIDER CIRCUIT

A **potential divider circuit** is a very common circuit used in electronics where an input voltage has to be converted to another voltage less than it. This circuit is very useful for all analog circuits where variable voltages are required; hence, it is important to understand how this circuit works and how to calculate the values of Resistors.




A voltage divider circuit is very simple circuit consisting of only two resistors ($R_1$ *and* $R_2$) as shown above. The required output voltage ($\boldsymbol{V_{out}}$) can be obtained across the resistor $R_2$. Using these two resistors, one can convert an input voltage to any required output voltage; this output voltage depends on the values of the resistances $R_1$ *and* $R_2$. The formula to calculate $\boldsymbol{V_{out}}$ is shown below.

$$\boldsymbol{V_{out} = V_{in}\frac{R_2}{R_1 + R_2}}$$

Where:

- $V_{in}$ is the input voltage
- $R_1$ is the resistance of the first resistor,
- $R_2$ is the resistance of the second resistor,
- $V_{out}$ Is the output voltage.

Now, considering the problem at hand, the two voltage values are 5v and 10v for $\boldsymbol{V_{out}}$ and $\boldsymbol{V_{in}}$ respectively. Thus, the resistor values are calculated as follows:

$$\frac{5}{10} = 0.5 = \boldsymbol{\frac{R_2}{R_1 + R_2} = scale\,factor}$$
$$0.5(\boldsymbol{R_1 + R_2}) = \boldsymbol{0.5R_1 + 0.5R_2 = R_2}$$
$$\boldsymbol{0.5R_1 = R_2 - 0.5R_2 = 0.5R_2}$$
$$thus, \boldsymbol{R_1 = R_2}$$

Thus, two resistors of the same value (in kΩ) can convert the voltage of 10v to 5v.

## 4.0 RESULTS

### 4.1 COMPONENTS REQUIRED

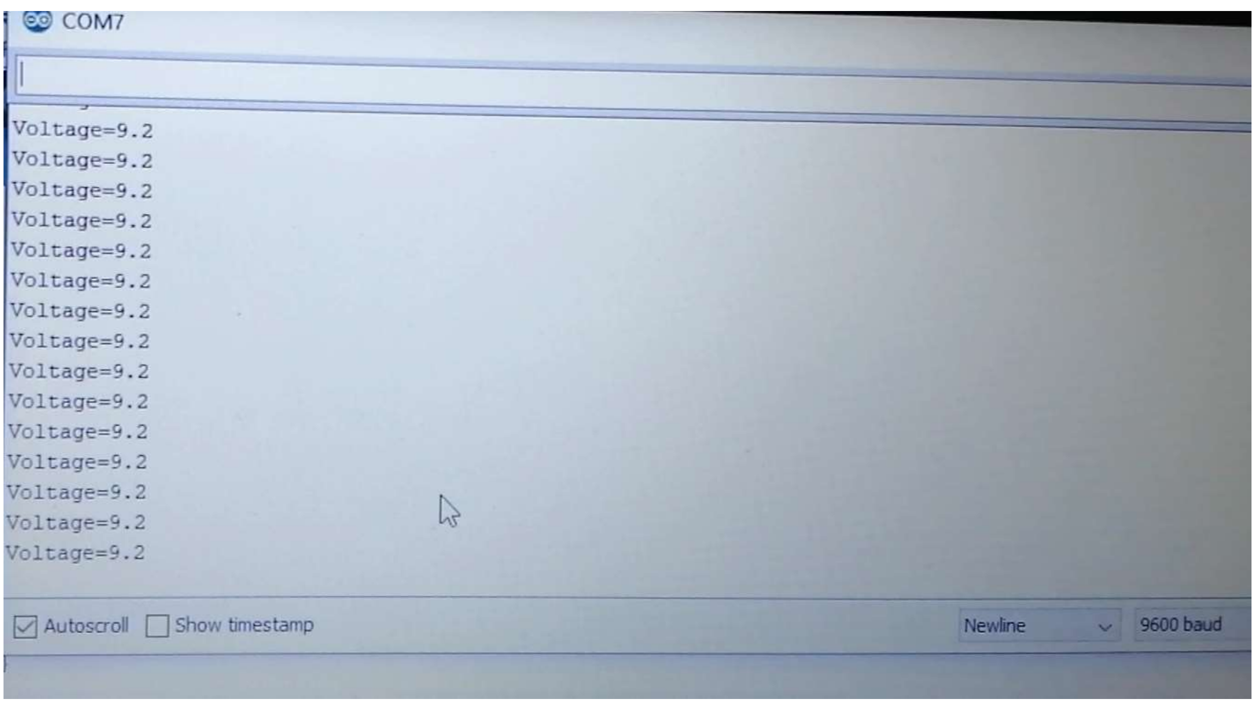
i) Arduino Board; ii) 10v battery; iii) Connecting wires; iv) Breadboard;
v) Two Resistors of 20kΩ and equal value are used, vi) Serial monitor.

### 4.2 C PROGRAM CODING:

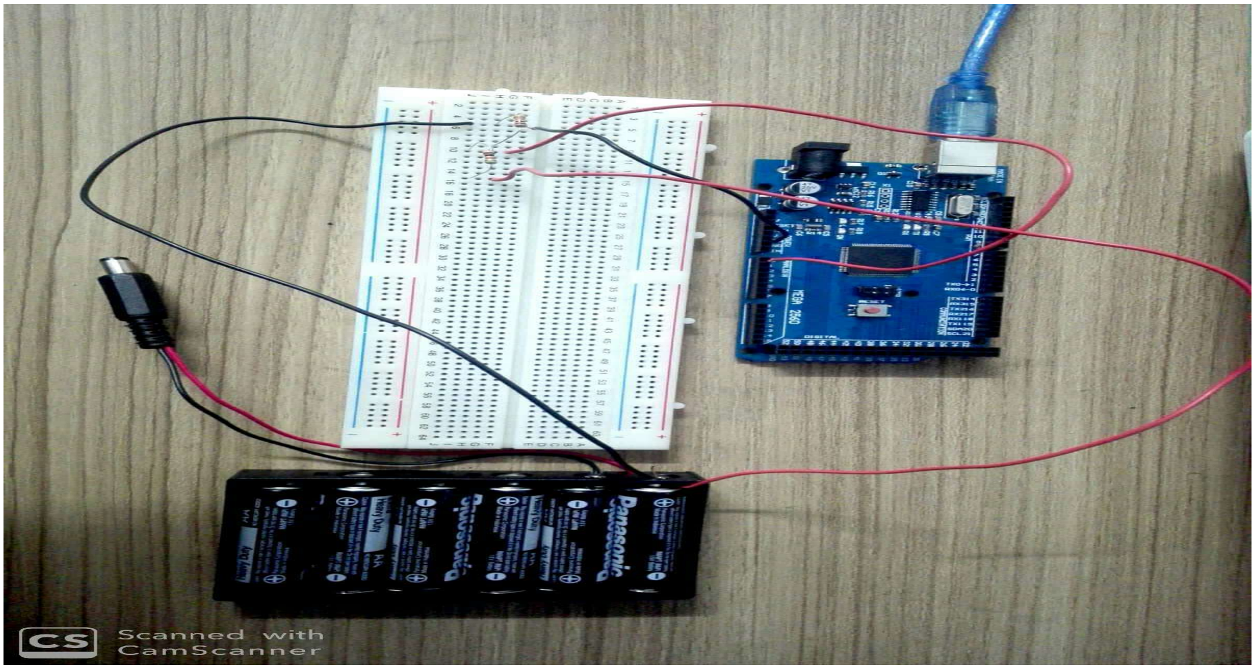
```
int value=0;
float voltage;
float R1=20000.0;
float R2=20000.0;

void setup() {
  Serial.begin(9600);
}
void loop() {
   int sensorValue = analogRead(A0);
   float voltage = sensorValue * (5.0 / 1023.0)*((R1+R2)/R2);
  Serial.print("voltage=");
  Serial.println(voltage,1);
}
```

### 4.3 OUTPUT




### 4.4 CIRCUIT CONNECTION




**5.0 CONCLUSION**

One major consideration before designing the circuit input is the maximum voltage that can be fed into ADC of ATmega328. As stated in the datasheet, ATmega328 or the Arduino can only accept a maximum of 5V. To adapt with the specification of 0-9.9V, a simple external voltage divider circuit is built to drop the voltage feed into it to a reasonable level. The versatility, accuracy, precision, robustness of digital electronic devices. Can not be matched with that of analog devices as the former ones are far more superior. Digital voltmeter is a very simple example of a digital electronic device.

The disadvantage of using this method, the voltage divider based voltmeter is the error of measurement (the data acquired is not as precise as it should be due to the voltage drop across the resistances) hence, we need multiple ranges of voltmeter. In order to reduce the error, the ratio of R1 and R2 in the voltage divider must be minimum. We choose R1 as 100kohm and R2 10kohm then it is possible to measure the voltages up to 50v .

It can be concluded that the project carried out successfully according to the task requirement. The Aduino UNO was able to function as a digital voltmeter and within the range of 0 to 10V as required.

**REFERENCES**

1. Book electronic instrumentation and measurement. Page 142

2. Book exploring arduino : tools and techniques for engineering wizardy,2013,Jeremy blum, page 170.

3. https:// www.codrey.com/arduino - projects / nano - digital - voltmeter.

4. https://store. Arduino.cc/usa/ arduino – uno – rev3.

5. https://www. Arduino. cc/en/products/compare.

6. https://www. Arduino .cc/en/main/arduinoboarduno 328p.

7. https://www. Arduino .cc/en/reference/board.

8. https://www. Arduino .cc/en/tutorial/hello world.

9. www.resistorguide.com/potentiometer/.

10. https://www. Electronics hub.org/digital – arduino – voltmeter/.

11. http:\\WWW.Keysight.Com Voltage Measurements

12. http:\\WWW.engnetbase.Com

13. Beginning C for Arduino by Jack Purdum

14. https://www.arduino.cc/en/Tutorial/HomePage