**Arduino Calculator**

**A Seminar report Submitted**

**in Partial Fulfillment of the Requirements**

**for the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**Electronics & Communication Engineering**

**By**

**Abdullah Aameen**

**(1803231003)**

**Under the Supervision of**

Mr. Dheeraj Singh



**ABES ENGINEERING COLLEGE**

**GHAZIABAD**

**2021-2022**

ACKNOWLEDGEMENT

I, express my sincere thanks to my supervisor, Ms. Himani Garg ECE Department, ABES   
Engineering College, UPTU University for guiding us right from the inception till successful completion of the seminar. I would also like to thank our HOD Prof. (Dr.) Sanjay Kumar Singh for his valuable guidance and cooperation to decide Seminar topic and its content.

Signature: Abdullah Aameen

Name: Abdullah Aameen

Roll No: 1803231003

Date: 02-12-2021

**ABSTRACT**

In this project we are going to make an arduino based calculator, with the help of arduino UNO, 4x4 keypad, 16x2 LCD display, bread board and jumping wires. In this calculator the input is entered by a real time user from the 4x4 keypad then the output will be displayed on 16x2 LCD display.

The function of calculation and control are performed using Arduino UNO , in this calculator only some basic Calculations are done like addition, subtraction, division and multiplication.

To have more complicated calculations then the more arithmetic program should be done carefully For making a more complex calculator it should handle bigger number and more floating point . With the help of code in this project the all calculations process is done in this project we learn use of Arduino, keypad and LCD display.

With the help of arduino we can transform coding program to the multiple components in this the arduino work is important which transfer the coding language to the components by the which output will come.

In this calculator 4x4 keypad is used for the input part and LCD display is used for output the input is entered by a user then output will be displayed on the keypad.

The calculator is an extension of a Mathematician and it is opened up new possibilities with in mathematics. The control and arthimatic program is performed using Arduino software which make this calculator work completely.

**TABLE OF CONTENT**

Acknowledgement

Abstract

**CHAPTER 1: INTRODUCTION : Components in Calculator**

1.1 Arduino UNO

1.2 LCD Structure

1.3 Potentiometer

1.4 Keypad

**CHAPTER 2: LITERATURE REVIEW**

2.1 Links

**CHAPTER 3: CONTENTS**

3.1 Code

3.2 Simulation

**CHAPTER 4: CONCLUSION**

**REFERENCES**

**INTRODUCTION**

In this project we learn how the Arduino Based Calculator works. The calculator reads from left to right. In this C programming is used to perform the working of calculator. The calculator mainly performs some basic calculations like addition, subtraction, multiplication, and division. With the help of Arduino we can transform coding program to the multiple components,.

Arduino work is important which transfer the coding language to the components by the which output will come. In this calculator 4x4 keypad is used for the input part and LCD display is used for output.

The input is entered by a user then output will be displayed on the keypad. The calculator is an extension of a Mathematician and has opened up new possibilities with in mathematics. The Control And Arthimatic Program Is performed using Tinkercad software which make this calculator work completely.

The features of Arduino UNO: lines . The LCD stands for liquid crystal display the LCD displays are used various devices like mobile, computers, TVs and

● 5v operating voltage

● The range of 7v to 12v required input voltage

● The input voltage ranges from 6v to 20V

● 14 digital input/output pins

● 40mA DC current for each I/O pins

● 32 KB flash memory

● 6 analog I/O pins

● SRAM is 2 KB

● EEPROM is 1 KB

● 16 MHz, click speed

**DISCUSSION**

Arduino UNO

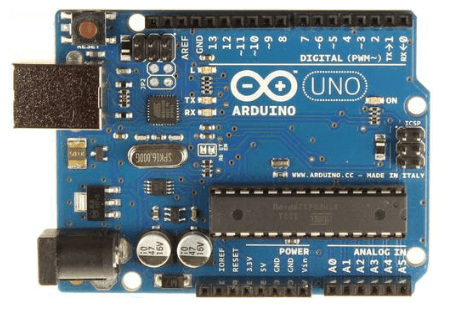
The Arduino UNO is a standard board of Arduino. Here UNO means 'one' in Italian. It was named as UNO to label the first release of Arduino Software. It was also the first USB board released by Arduino. It is considered as the powerful board used in various projects. Arduino.cc developed the Arduino UNO board.

Arduino UNO is based on an ATmega328P microcontroller. It is easy to use compared to other boards, such as the Arduino Mega board, etc. The board consists of digital and analog Input/Output pins (I/O), shields, and other circuits.

The Arduino UNO includes 6 analog pin inputs, 14 digital pins, a USB connector, a power jack, and an ICSP (In-Circuit Serial Programming) header. It is programmed based on IDE, which stands for Integrated Development Environment. It can run on both online and offline platforms.

The IDE is common to all available boards of Arduino.

The Arduino board is shown below.



The components of Arduino UNO board are shown below:

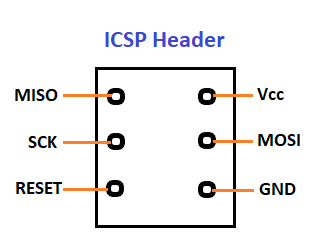


Let's discuss each component in detail.

* **ATmega328 Microcontroller**- It is a single chip Microcontroller of the ATmel family. The processor code inside it is of 8-bit. It combines **Memory (SRAM, EEPROM, and Flash), Analog to Digital Converter, SPI serial ports, I/O lines, registers, timer, external and internal interrupts, and oscillator.**
* **ICSP pin**- The In-Circuit Serial Programming pin allows the user to program using the firmware of the Arduino board.
* **Power LED Indicator**- The ON status of LED shows the power is activated. When the power is OFF, the LED will not light up.
* **Digital I/O pins**- The digital pins have the value HIGH or LOW. The pins numbered from D0 to D13 are digital pins.
* **TX and RX LED's**- The successful flow of data is represented by the lighting of these LED's.
* **AREF-**The Analog Reference (AREF) pin is used to feed a reference voltage to the Arduino UNO board from the external power supply.
* **Reset button**- It is used to add a Reset button to the connection.
* **USB**- It allows the board to connect to the computer. It is essential for the programming of the Arduino UNO board.
* **Crystal Oscillator**- The Crystal oscillator has a frequency of 16MHz, which makes the Arduino UNO a powerful board.
* **Voltage Regulator**- The voltage regulator converts the input voltage to 5V.
* **GND**- Ground pins. The ground pin acts as a pin with zero voltage.
* **Vin**- It is the input voltage.
* **Analog Pins**- The pins numbered from A0 to A5 are analog pins. The function of Analog pins is to read the analog sensor used in the connection. It can also act as GPIO (General Purpose Input Output) pins.

The ICSP header consists of 6 pins.

The structure of the ICSP header is shown below:



It is the top view of the ICSP header.

* **SDA**

It stands for **Serial Data**. It is a line used by the slave and master to send and receive data. It is called as a **data line,** while SCL is called as a clock line.

* **SCL**

It stands for **Serial Clock**. It is defined as the line that carries the clock data. It is used to synchronize the transfer of data between the two devices. The Serial Clock is generated by the device and it is called as master.

* **SPI**

It stands for **Serial Peripheral Interface**. It is popularly used by the microcontrollers to communicate with one or more peripheral devices quickly. It uses conductors for data receiving, data sending, synchronization, and device selection (for communication).

* **MOSI**

It stands for Master Output/ Slave Input.

The MOSI and SCK are driven by the Master.

* **I2C**

It is the two-wire serial communication protocol. It stands for Inter Integrated Circuits. The I2C is a serial communication protocol that uses SCL (Serial Clock) and SDA (Serial Data) to receive and send data between two devices.

3.3V and 5V are the operating voltages of the board.

# Arduino LCD Display

The LCD (**Liquid Crystal Display**) is a type of display that uses the liquid crystals for its operation.

Here, we will accept the serial input from the computer and upload the sketch to the Arduino.

. The characters will be displayed on the LCD.

The library that allows us to control the LCD.

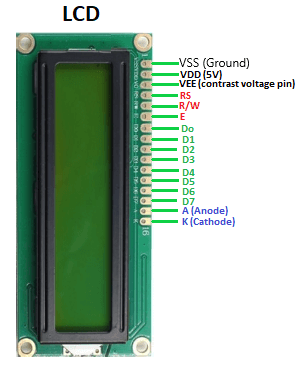
display is called **Liquid Crystal Library**, which is discussed below:

The library is declared as:

### **LCD Structure**

The LCD display has a 16-pin interface.

The structure of the LCD is shown below:



The Liquid Crystal Display has a parallel interface. It means that the microcontroller operates several pins at once to control the LCD display.

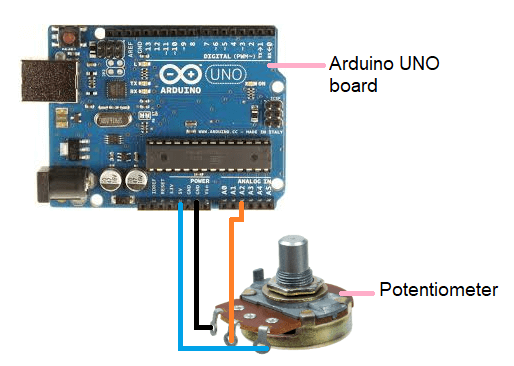
# Arduino Potentiometer

The potentiometer is a device that is used to measure the voltage or electric potential. It provides a variable resistance when the shaft of the device is turned.

Here, we will measure the amount of resistance as an analog value produced by the potentiometer. We will connect the potentiometer to the Arduino uno board.

and will measure the state of the potentiometer. The required code will be uploaded from our computer to the Arduino board.

The variable resistance measured by the potentiometer can be easily read as an analog value into the Arduino board.



## **Potentiometer with LED**

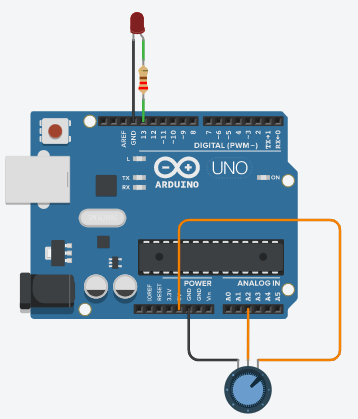
In this example, we will use a potentiometer that controls the value at which LED blinks.

**Hardware Required**

The required components are listed below:

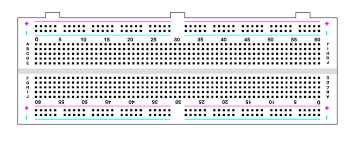
* 1 x red LED
* Arduino UNO R3 board
* 10K Ohm Potentiometer
* Jump wires
* 220 Ohm resistor

Connection

* One outer pin of the potentiometer is connected to ground (**GND**), and other external pin is connected to **5V** of the Arduino board.
* The middle pin of the potentiometer is connected to the analog input pin A2 of the board.
* The positive terminal of the LED is connected in series with 220 Ohm resistor to pin number 13 of the board, and the negative terminal is connected to the GND.
* **Connection diagram**
* The connection diagram is shown below:
* 

Breadboard

A breadboard is **a rectangular plastic board with a bunch of tiny holes in it**. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode).



.4x4 Keypad

4x4 keypad consists **of 4 rows and 4 columns**. Switches are placed between the rows and columns. A key press establishes a connection between the corresponding row and column, between which the switch is placed.



.Jumper wire

Jumper wires are simply wires that have connector pins at each end, **allowing them to be used to connect two points to each other without soldering**.



**Circuit Design**

The design of the circuit is simple to understand but a little bit messy to implement as it involves a lot of connections and wires. The design of the circuit is explained here.

First, the connection of LCD display to Arduino UNO. LCD is used in 4 – bit mode and hence, only 4 data connections are needed. The data pins D4 through D7 (Pins 11 – 14) are connected to digital I/O pins 2 through 5 of the Arduino.

Then, we need to connect the three control pins of LCD i.e. RS, RW and E. RS (Pin 4) is connected to Pin 0 of the Arduino. E (Pin 6) is connected to Pin 1 of the Arduino. RW (Pin 5) is connected to ground.

***Note:***

RS and E are connected to Pins 0 and 1 of Arduino which are usually associated with serial communication. We must be careful while programming the device and hence we must disconnect these two connections while we are [programming Arduino](https://www.electronicshub.org/arduino-program-analysis/) UNO.

The next connections are with respect to power supply of LCD. Pin 1 (VSS) is connected to ground and Pin 2 (VDD) is connected to 5V supply. Pin 3 (VEE) is connected to a POT for adjusting the contrast of display.

Pin 15 and Pin 16 are supply pins for backlight LED. They must be connected to 5V and ground supply.

The next connection is with respect to matrix keypad. The 4 row pins of the keypad (row 1 to row 4) are connected to pins 13 through 10 of Arduino UNO and the 4 column pins (col 1 to col 4) are connected to pins 9 through 6 of Arduino UNO.

**Working Process**

A simple calculator is designed in this project using Arduino UNO board and a keypad matrix. The working of the project is explained here.

The keypad in the project consists of 4 rows and 4 columns and the structure is similar to the image shown below.



A special library called “***Keypad.h***” is used in order to find out which key is pressed. This library must be downloaded separately and must be added to Arduino IDE.

When the key is determined, relevant operations like addition, subtraction, multiplication and division can be performed. A clear screen (of LCD display) button is also included.

Even though the library does the most of the work, we can understand the processing of determining the key.

As mentioned in the circuit design, the rows of the keypad are connected to pins 13 – 10 of Arduino and columns of keypad are connected to pins 9 – 6 of Arduino.

All the row pins are pulled high and all the column pins are pulled low. From now onwards, the Arduino waits for the column pin to become HIGH, which happens if a key is pressed.

Consider, if a key is pressed, the switch closes the connection between the corresponding row and column. Due to the flow of current from high to low potential, the column will become high.

This change in potential at the column, makes the Arduino to understand that the key corresponding to the particular column is pressed.

Now it’s time to find the row. Instead of giving HIGH signal to all the rows at once, Arduino will enable HIGH to one row at a time and check whether there is a HIGH signal detected on the column. If there is no HIGH signal detected on the corresponding column, Arduino will scan for the next row.

This process will repeat until there is a HIGH signal detected on the corresponding column. Once if the HIGH signal is detected on the corresponding column, thus the row is identified.

From the outside, this key detection process seems to be taking long time but really the time taken to complete all the above mentioned process will be in micro seconds. But the average time a human takes for pushing a button is in milli seconds. That’s why, the key detection process will start when the key is pressed and completes before the key is released.

#include <LiquidCrystal.h>

#include <Keypad.h>

const byte ROWS = 4;

const byte COLS = 4;

// Define the Keymap

char keys[ROWS][COLS] = {

{'7','8','9','D'},

{'4','5','6','C'},

{'1','2','3','B'},

{'\*','0','#','A'}

};

byte rowPins[ROWS] = { 0, 1, 2, 3 };// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte colPins[COLS] = { 4, 5, 6, 7 }; // Connect keypad COL0, COL1 and COL2 to these Arduino pins.

Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS ); // Create the Keypad

const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13; //Pins to which LCD is connected

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

long Num1,Num2,Number;

char key,action;

boolean result = false;

void setup() {

lcd.begin(16, 2); //We are using a 16\*2 LCD display

lcd.print("DIY Calculator"); //Display a intro message

lcd.setCursor(0, 1); // set the cursor to column 0, line 1

lcd.print("-CircuitDigest"); //Display a intro message

delay(2000); //Wait for display to show info

lcd.clear(); //Then clean it

}

void loop() {

key = kpd.getKey(); //storing pressed key value in a char

if (key!=NO\_KEY)

DetectButtons();

if (result==true)

CalculateResult();

DisplayResult();

}

void DetectButtons()

{

lcd.clear(); //Then clean it

if (key=='\*') //If cancel Button is pressed

{Serial.println ("Button Cancel"); Number=Num1=Num2=0; result=false;}

if (key == '1') //If Button 1 is pressed

{Serial.println ("Button 1");

if (Number==0)

Number=1;

else

Number = (Number\*10) + 1; //Pressed twice

}

if (key == '4') //If Button 4 is pressed

{Serial.println ("Button 4");

if (Number==0)

Number=4;

else

Number = (Number\*10) + 4; //Pressed twice

}

if (key == '7') //If Button 7 is pressed

{Serial.println ("Button 7");

if (Number==0)

Number=7;

else

Number = (Number\*10) + 7; //Pressed twice

}

if (key == '0')

{Serial.println ("Button 0"); //Button 0 is Pressed

if (Number==0)

Number=0;

else

Number = (Number\*10) + 0; //Pressed twice

}

if (key == '2') //Button 2 is Pressed

{Serial.println ("Button 2");

if (Number==0)

Number=2;

else

Number = (Number\*10) + 2; //Pressed twice

}

if (key == '5')

{Serial.println ("Button 5");

if (Number==0)

Number=5;

else

Number = (Number\*10) + 5; //Pressed twice

}

if (key == '8')

{Serial.println ("Button 8");

if (Number==0)

Number=8;

else

Number = (Number\*10) + 8; //Pressed twice

}

if (key == '#')

{Serial.println ("Button Equal");

Num2=Number;

result = true;

}

if (key == '3')

{Serial.println ("Button 3");

if (Number==0)

Number=3;

else

Number = (Number\*10) + 3; //Pressed twice

}

if (key == '6')

{Serial.println ("Button 6");

if (Number==0)

Number=6;

else

Number = (Number\*10) + 6; //Pressed twice

}

if (key == '9')

{Serial.println ("Button 9");

if (Number==0)

Number=9;

else

Number = (Number\*10) + 9; //Pressed twice

}

if (key == 'A' || key == 'B' || key == 'C' || key == 'D') //Detecting Buttons on Column 4

{

Num1 = Number;

Number =0;

if (key == 'A')

{Serial.println ("Addition"); action = '+';}

if (key == 'B')

{Serial.println ("Subtraction"); action = '-'; }

if (key == 'C')

{Serial.println ("Multiplication"); action = '\*';}

if (key == 'D')

{Serial.println ("Division"); action = '/';}

delay(100);

}

}

void CalculateResult()

{

if (action=='+')

Number = Num1+Num2;

if (action=='-')

Number = Num1-Num2;

if (action=='\*')

Number = Num1\*Num2;

if (action=='/')

Number = Num1/Num2;

}

void DisplayResult()

{

lcd.setCursor(0, 0); // set the cursor to column 0, line 1

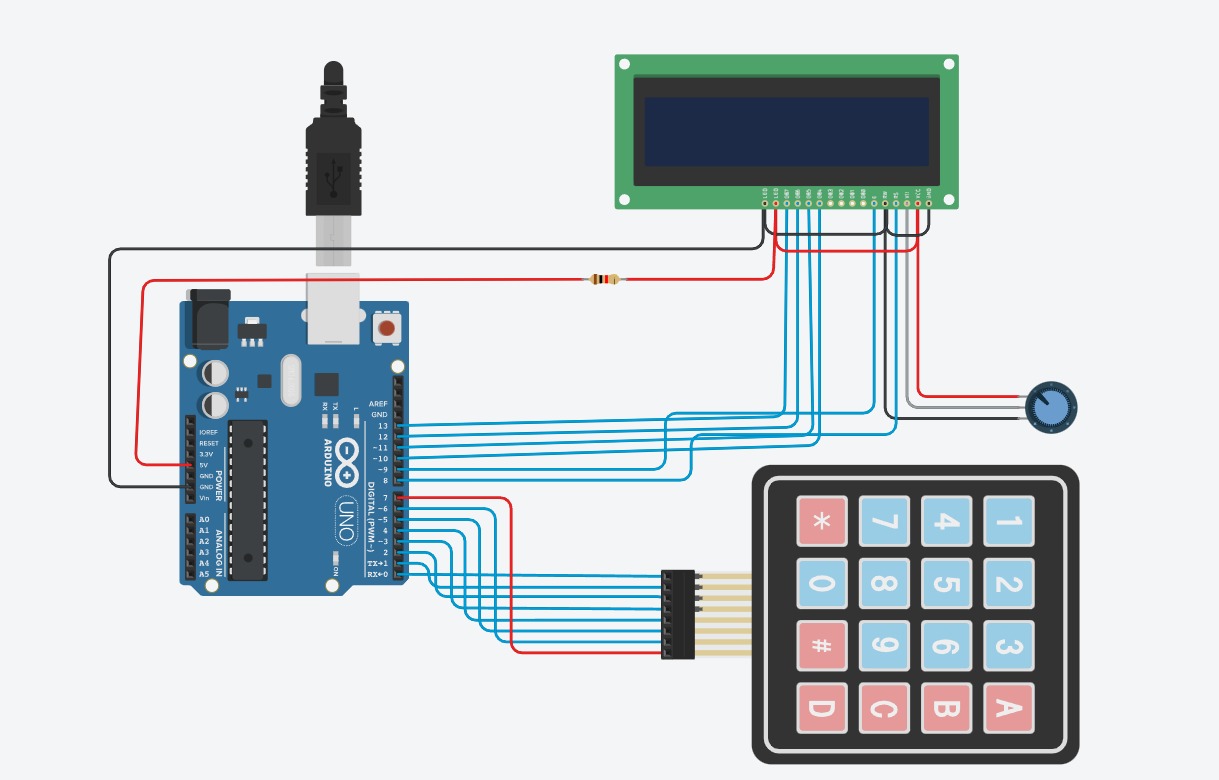
lcd.print(Num1); lcd.print(action); lcd.print(Num2);

if (result==true)

{lcd.print(" ="); lcd.print(Number);} //Display the result

lcd.setCursor(0, 1); // set the cursor to column 0, line 1

lcd.print(Number); //Display the result

} 

**CONCLUSION**

In this the calculator is easy to handle it works by the help of arduino ide software in which we 1st type the program by the help of that program we can operate this calculator . This calculator solves the problem like addition, subtraction, division and multiplication. It cannot perform some basic calculations like square roots, percentages and cos, sine problems for that the program should be carefully programmed, the calculator are solves the calculations faster and accurately. The role of arduino in this calculator is it uploads the program from the system by the help of data cable it runs the program in circuit from input to output it takes the input from keypad and display output on display

**REFERENCES**

* Site: <https://create.arduino.cc/projecthub/123samridhgarg/arduino-calculator-bb57c7>
* Site: <https://www.electronicshub.org/arduino-calculator/>
* Arduino Tutorial Site: <https://www.javatpoint.com/arduino>
* Site: <https://quartzcomponents.com/blogs/electronics-projects/arduino-calculator-using-4x4-keypad>
* Site: [https://www.allaboutcircuits.com/projects/simple-arduino-based-calculator](https://www.allaboutcircuits.com/projects/simple-arduino-based-calculator/)\[/](https://www.allaboutcircuits.com/projects/simple-arduino-based-calculator/)
* Shubham Thati, Arduino Based Calculator, (July 16, 2021), on link: <https://deliverypdf.ssrn.com/delivery.php?ID=012017103006067097119016080125069027113078029065058058030089008028005122005084100125107016062038118063096014094116100086068021029080030038029124011003030030115023071077008008096026017091015015081028117097084122086107066006004125025000004080100006125066&EXT=pdf&INDEX=TRUE>
* Link: <http://203.201.63.46:8080/jspui/bitstream/123456789/6175/1/PR3173.pdf>
* Tummanapally, Shraddha Shree and Sunkari, Saideep, Traffic Data Collection and Analysis based on Wireless Sensor Network (July 12, 2021). Available at SSRN: <https://ssrn.com/abstract=3885102> or <http://dx.doi.org/10.2139/ssrn.3885102>