

# Project Calculator

By Swaraj Lahiri

IT Student

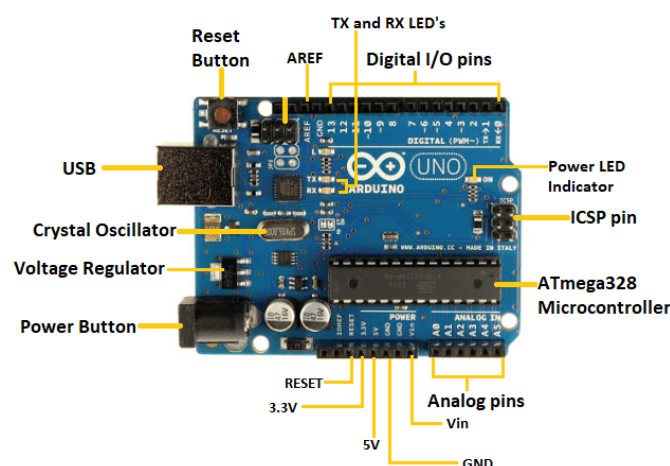
JIS College of Engineering

## Abstract

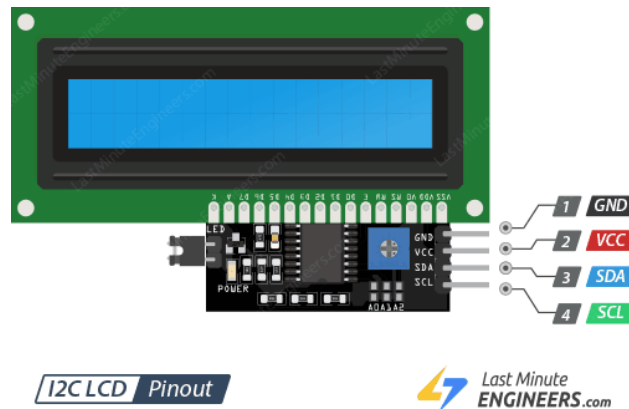
In the daily life, the common mathematical functions like addition, multiplication, subtraction and division are very common. Everyday we need to calculate the total amount we need to pay to the shopkeeper, the change we need to get from the shopkeeper. If you are having a shop, it is very bad to keep the customers the waiting while you calculate the whole thing. During discounts, we have to check how much the product actually costs, and most of the times we think about something else and most of the times, it is very difficult for someone to calculate the whole thing on mind. It gives us a lot of stress. Also, during work, we need to work on very long calculations, which is often very troublesome for do it in the mind. It also creates a mess trying to have a rough area for the calculations. Thus, we made the calculator which allows to get the answers in microseconds. Just type the numbers and then click on the desired operations, and on the go, we have the answers ready to prevent any stressful situations. This project uses an Arduino Uno, which acts as the Central Processing Unit for the calculator. A keypad in order to provide the Calculator with the numbers and also a LCD display which helps the user see what they are giving in the screen and gives out the results.

## Requirements

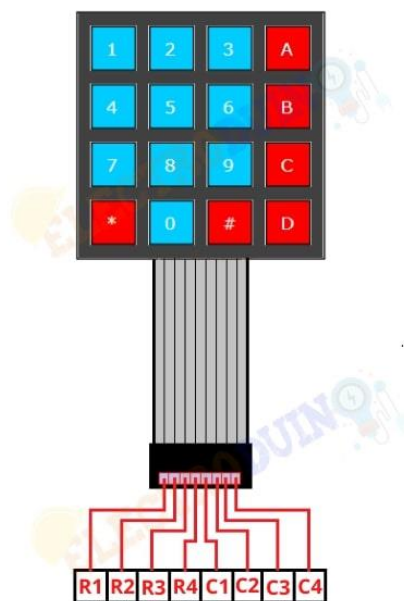
- Arduino Uno: Arduino is an open-source development platform that allows us to program it using the Arduino IDE. It contains the Microcontroller ATmega328P. The microcontroller acts as the Central Processing Unit for the calculator in this project.



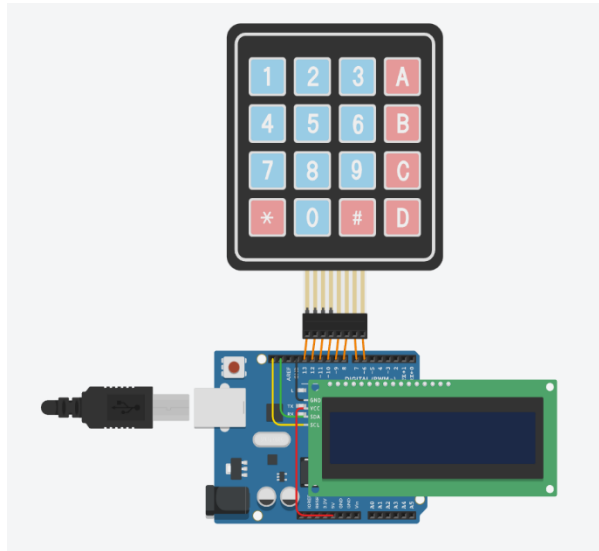
- **LCD :** The LCD (Liquid Crystal Display) is a display device the desire output programmed by the Programmer. In this project it serves as the display for the calculator. It helps us to enter the numbers needed to the calculation and then shows us the result. The module used here follows I2C communication which reduces the number of pins occupied by the LCD.



- **Keypad(4x4):** This keypad module contains 16 switches inside and through the help of a matrix, helps us to accept the inputs from the user. It contains all the numbers from 0 to 9 and also contains extra letters such as A,B,C,D. We used this module in order to accept the numbers from the user and also accept the mathematical operations to be carried out. This keypad helps us to accept requests from the user without any device in the middle for example the Serial Monitor in the laptop.



## Circuit Diagram



## Code

```
#include<Adafruit_LiquidCrystal.h>
#include <Keypad.h>
```

```
Adafruit_LiquidCrystal lcd(0);
```

```
const byte ROWS = 4;
const byte COLS = 4;
```

```
char keys [ROWS] [COLS] = {
  {'1', '2', '3', '+'},
  {'4', '5', '6', '-'},
  {'7', '8', '9', 'C'},
  {'*', '0', '=', '/'}
};
```

```
byte rowPins[ROWS]={13,12,11,10};
byte colPins[COLS]={9,8,7,6};
```

```
Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```

boolean presentValue = false;
boolean next = false;
boolean final = false;
String num1, num2;
int answer = 0;
char op;

void setup()
{
  lcd.begin(16,2);
  lcd.setBacklight(1);
  lcd.print("Welcome!");
  lcd.setCursor(0,1);
  lcd.print("Calculator Ready");
  delay(100);
  lcd.clear();
  Serial.begin(115200);
}

void loop() {
  char key = myKeypad.getKey();

  if (key != NO_KEY && (key == '1' || key == '2' || key == '3' || key == '4' || key == '5' || key == '6' || key == '7'
  || key == '8' || key == '9' || key == '0'))
  {
    if (presentValue != true)
    {
      num1 = num1 + key;
      int numLength = num1.length();
      lcd.setCursor(0, 0);
      lcd.print(num1);
    }
    else
    {
      num2 = num2 + key;
      int numLength = num2.length();
      int numLength1 = num1.length();
      lcd.setCursor(1 + numLength1, 0);
      lcd.print(num2);
      final = true;
    }
  }

  else if (presentValue == false && key != NO_KEY && (key == '/' || key == '*' || key == '-' || key == '+'))
  {
    if (presentValue == false)
    {
      int numLength = num1.length();
      presentValue = true;
      op = key;
      lcd.setCursor(0 + numLength, 0);
      lcd.print(op);
    }
  }
}

```

```

    }
}

else if (final == true && key != NO_KEY && key == '=') {

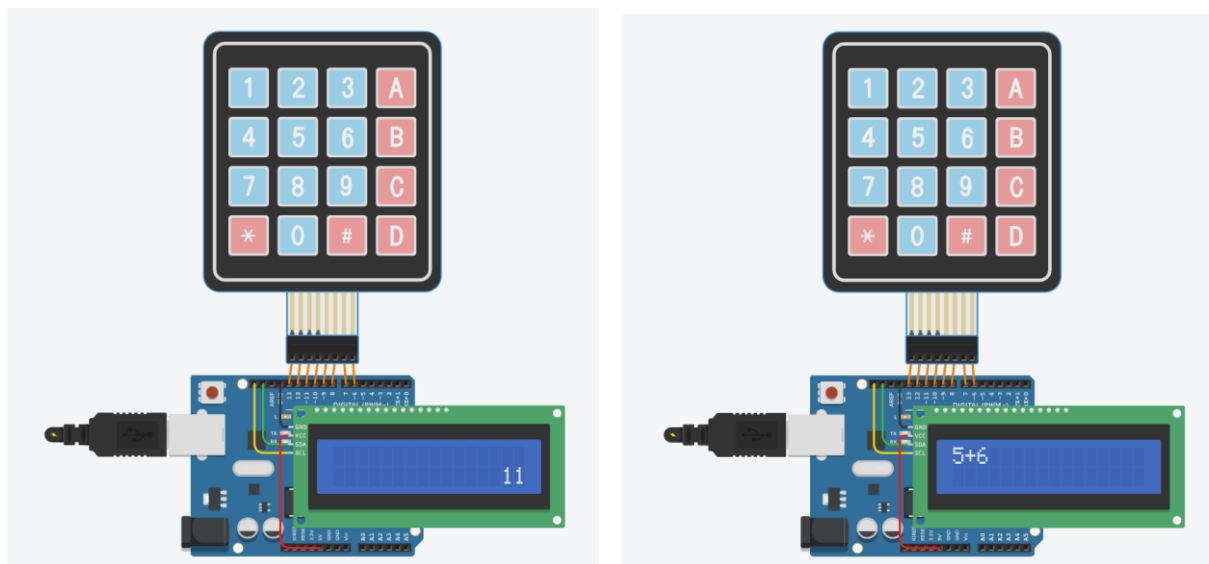
    if (op == '+') {
        answer = num1.toInt() + num2.toInt();
    }
    else if (op == '-') {
        answer = num1.toInt() - num2.toInt();
    }
    else if (op == '*') {
        answer = num1.toInt() * num2.toInt();
    }
    else if (op == '/') {
        answer = num1.toInt() / num2.toInt();
    }
    lcd.clear();
    lcd.setCursor(16, 1);
    lcd.autoscroll();
    lcd.print(answer);
    lcd.noAutoscroll();

}
else if (key != NO_KEY && key == 'C') {
    lcd.clear();
    presentValue = false;
    final = false;
    num1 = "";
    num2 = "";
    answer = 0;
    op = '';
}
}

```

## Working

The keypad allows us to accept the Data from the user. The keypad contains (4\*4) 16 keys contain the digits 0 to 9 and also extra keys, i.e. A, B, C, D, \*, #. The A represents Additions, B represents Subtraction, C represents Clearing the screen and D presents Division, \* represents Multiplication and # represents Equals to which gives us the answer to the required mathematical operation. This



input is processed in the ATmega328P, and the inputted numbers and the operations are displayed on the LCD display. The results are calculated by the microcontroller (ATmega328P) and the results are displayed on the next line.

## Conclusion

This device can do many other things with more resources. With more keys we can include automated Discount calculator and other things that will make life more efficient so that we can release a lot of load from our mind and give time to the things we really care about.