FY B.Tech Trimester- 2

Academic Session: 2021-22

Project Based Learning Activity

Course Name: BEEE

Course Code: ECE1022A

Basic Arduino Calculator

Shreerang Mhatre

FY Division: 11, Batch: K3

(1032211745)

(111056)

School of Electronics and Communication Engineering
MITWPU,

BASIC ARDUINO CALCULATOR

Introduction:

Calculator:
A calculator is a device that performs arithmetic operations on numbers. The simplest calculators can do only addition, subtraction, multiplication, and division. More sophisticated calculators can handle exponential operations, roots, logarithm s, trigonometric functions, and hyperbolic functions. Internally, some calculators actually perform all of these functions by repeated processes of addition.

The first solid-state electronic calculator was created in the early 1960s. Pocket-sized devices became available in the 1970s, especially after the Intel 4004, the first microprocessor, was developed by Intel for the Japanese calculator company Busicom. They later became used commonly within the petroleum industry (oil and gas).

Input:-
Electronic calculators contain a keyboard with buttons for digits and arithmetical operations; some even contain "00" and "000" buttons to make larger or smaller numbers easier to enter. Most basic calculators assign only one digit or operation on each button; however, in more specific calculators, a button can perform multi-function working with key combinations.

Display output:-
Calculators usually have Liquid Crystal Display (LCD) as output in place of historical Light Emitting Diode (LED) displays and Vacuum Florescent Display (VFD).
Large-sized figures are often used to improve readability; while using Decimal seperater (usually a point rather than a comma) instead of or in addition to vulgar fractions. Various symbols for function commands may also be shown on the display. Fractions such as $\frac{1}{3}$ are displayed as decimal approximations.

Arduino:
Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.

The Arduino platform has become quite popular with people just starting out with electronics, and for good reason. Unlike most previous programmable circuit boards, the Arduino does not need a separate piece of hardware (called a programmer) in order to load new code onto the board -- you can simply use a USB cable. Additionally, the Arduino IDE uses a simplified version of C++, making it easier to learn to program. Finally, Arduino provides a standard form factor that breaks out the functions of the micro-controller into a more accessible package.

What Does it Do?
The Arduino hardware and software was designed for artists, designers, hobbyists, hackers, newbies, and anyone interested in creating interactive objects or environments. Arduino can interact with buttons, LEDs, motors, speakers, GPS units, cameras, the internet, and even your smart-phone or your TV! This flexibility combined with the fact that the Arduino software is free, the hardware boards are pretty cheap, and both the software and hardware are easy to learn has led to a large community of users who have contributed code and released instructions for a **huge** variety of Arduino-based projects.
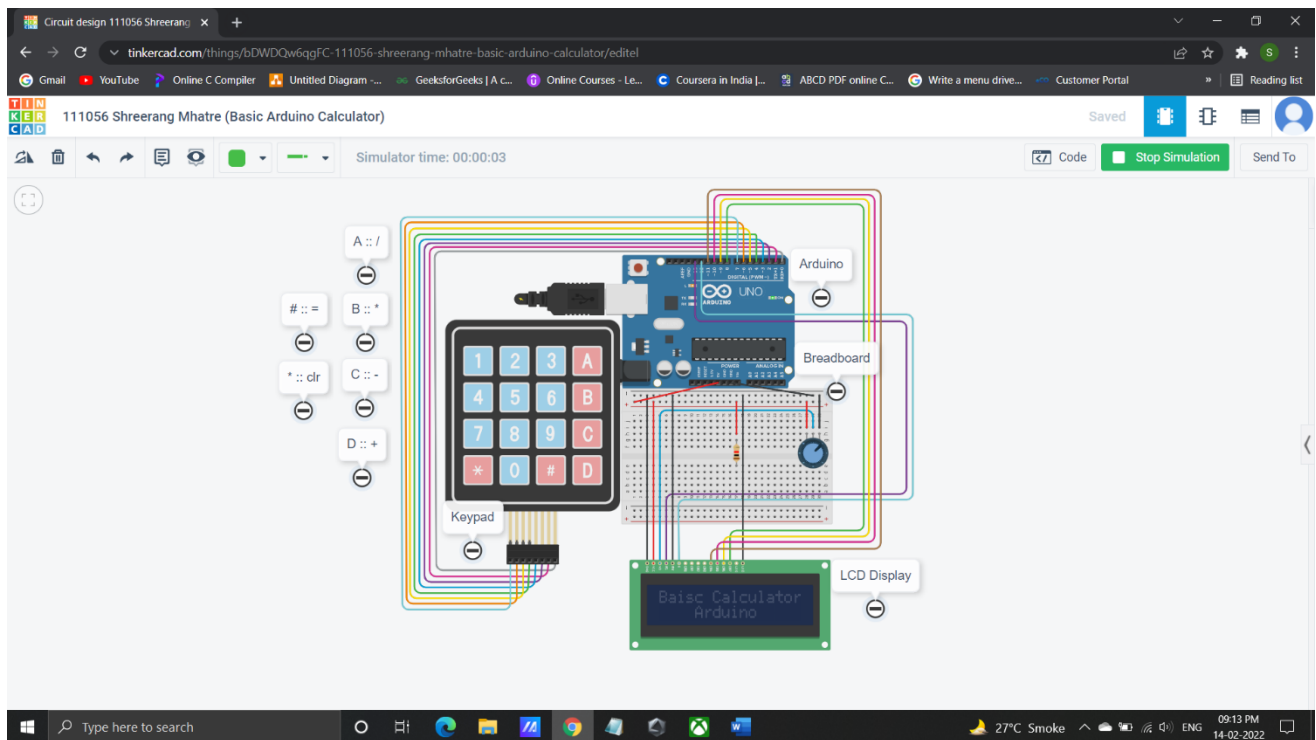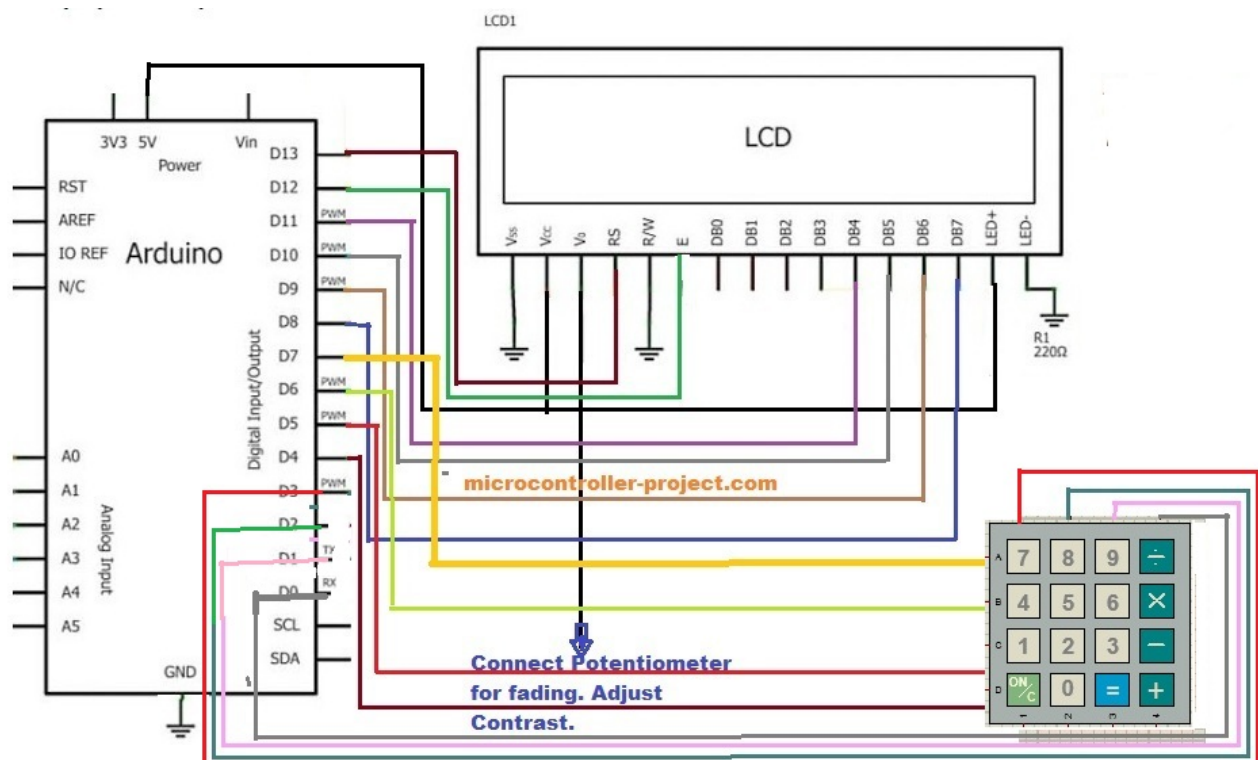
Arduino UNO R3:
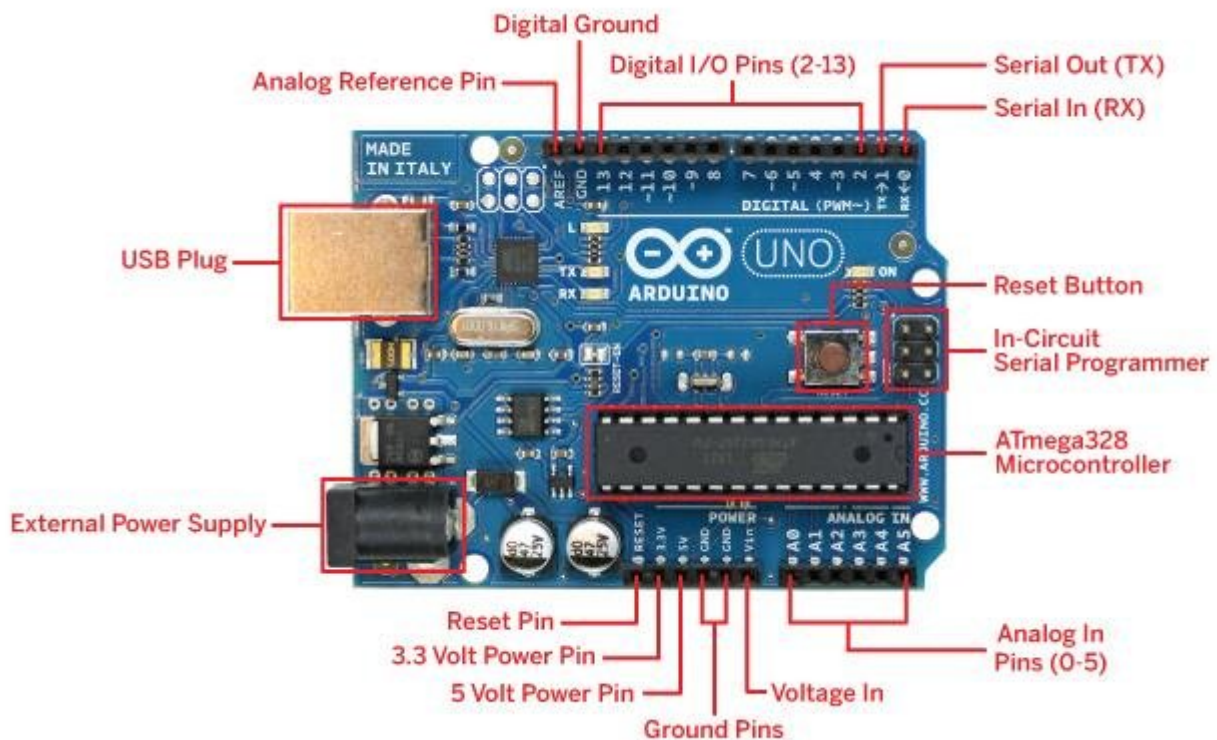
My Project:

Nowadays Calculators plays a vital role everywhere. Even for a small calculations we seek for it. So here is a simple Calculator designed using Tinkercad Software. Here in this project is the calculator takes input from a user in real-time via a keypad and displays output on a LCD display module, the control, arithmetic algorithm and calculation functions are performed using a Arduino UNO R3. The prototype of the system is configured and the simulation results for basic mathematical functions of calculator are expressed with the help of Tinkercad Software.

Basic Arduino Calculator on Tinkercad:

Circuit Diagram:

Component List:

1)Arduino Uno R3
2)Breadboard
3)Liquid Crystal Display (LCD 16 x 2)
4)Keypad 4 x 4
5)Potentiometer – 250Kohm
6)Resistor – 1Kohm
7)Connecting Wires

Operation Principle:

This simple device starts with a cleared LCD screen and waits for key input from the keypad. It forms numbers out of keys entered by the user. As soon as the user presses an operation button, it memorizes the first number and operation desired and continues the acquisition of the second number. When the user is done inputting the second number, after pressing the equals button on the keypad, the program performs the requested operation and prints the result. After that, it sits still and waits for the CLEAR button ('C') to be pressed so it can start over. (The user can also reset the program at any point in time.)

Working Principle:

Arduino uno is used as microcontroller in the project. Arduino takes the input from user and after analyzing the instructions produces output. The output is than displayed on 16×2 lcd. 4×4 numeric keypad is used as input. User presses the buttons on keypad to give input to Arduino calculator. Arduino calculator takes two digits and an operator as input. Arduino calculator then identifies the operator, computes results according to the operator and then displays the result on 16×2 lcd screen.

Lcd 16×2 is interfaced in 4-bit mode with arduino uno. Arduino pins D13, D12, D11, D10, D9 and D8 are occupied by 16×2 lcd. 4×4 Keypad rows are connected to pins D4, D5, D6 & D7 of arduino uno. Columns of keypad are connected to pins D0, D1, D2 & D3 of arduino uno.

A Potentiometer is used to control the contrast of the LCD display. A Resistor of 1 K ohm is used to give appropriate current to the lcd display. Connecting wires are connected accordingly and breadboard is used for better connections of the circuit.

Liquid Crystal library **(<LiquidCrystal.h>)** contains predefined functions which could be used in arduino code to easily interface and operate 16×2 lcd with arduino uno. I am interfacing 16×2 lcd in 4-bit mode with arduino uno. A special library called **(<keypad.h>)** is used in order to find out which key is pressed.

Arduino Calculator Code /System code:

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

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);  // setting the expander address, number of lcd columns, number of lcd
                                          rows
long first = 0;
long second = 0;
double total = 0;
int posit = 0 ;
char customKey;
const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = { // define keypad equivalents
  {'1','2','3','/'},
  {'4','5','6','*'},
  {'7','8','9','-'},
  {'C','0','=','+'}
};
byte rowPins[ROWS] = {7 ,6 ,5 ,4}; // connect rows from keypad to Arduino
byte colPins[COLS] = {3, 2, 1, 0}; // connect columns from keypad to Arduino

Keypad customKeypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); // class Keypad
                                                                              instance initialization
void setup(){
  lcd.begin(16,2);
  lcd.setCursor(3,0);
  lcd.print("Shreerang");
  lcd.setCursor(5,1);
  lcd.print("Mhatre");
  delay(3000);
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  lcd.print("Baisc Calculator");
  lcd.setCursor(4,1);
  lcd.print("Arduino");
  delay(2000);
lcd.clear(); //clears the LCD screen and positions the cursor in the upper-left corner.
}
void loop()
{
  customKey = customKeypad.getKey();
  switch(customKey)
  {
  case '0' ... '9': // get first value for calculation
    lcd.setCursor(0,0);
    first = first * 10 + (customKey - '0');
    lcd.print(first);
    posit++;
    break;

  case '+':
    first = (total != 0 ? total : first);
    lcd.setCursor(posit,0);
    lcd.print("+");
```

```cpp
      posit++;
      second = SecondNumber(); // get second number
      total = first + second;
      lcd.setCursor(1,1);
      lcd.print(total);
      first = 0, second = 0; // reset the values
      posit=0;
      break;

    case '-':
      first = (total != 0 ? total : first);
      lcd.setCursor(posit,0);
      lcd.print("-");
      posit++;
      second = SecondNumber();
      total = first - second;
      lcd.setCursor(1,1);
      lcd.print(total);
      first = 0, second = 0;
      posit=0;
      break;

    case '*':
      first = (total != 0 ? total : first);
      lcd.setCursor(posit,0);
      lcd.print("*");
      posit++;
      second = SecondNumber();
      total = first * second;
      lcd.setCursor(1,1);
      lcd.print(total);
      first = 0, second = 0;
      posit=0;
      break;

    case '/':
      first = (total != 0 ? total : first);
      lcd.setCursor(posit,0);
      lcd.print("/");
      posit++;
      second = SecondNumber();
      lcd.setCursor(1,1);

      second == 0 ? lcd.print("Error") : total = (float)first / (float)second;

      lcd.print(total);
      first = 0, second = 0;
      posit=0;
      break;

    case 'C':
      total = 0;
      first = 0;
      second = 0;
      posit = 0;
      lcd.clear();
      break;
    }
}
```

```cpp
long SecondNumber()
{
  while( 1 )
  {
    customKey = customKeypad.getKey();
    if(customKey >= '0' && customKey <= '9')
    {
      second = second * 10 + (customKey - '0');
      lcd.setCursor(posit,0);
      lcd.print(second);
    }
    if(customKey == 'C') {
      total = 0;
      first = 0;
      second = 0;
      posit = 0;
      lcd.clear();
      break;
    }

    if(customKey == '='){
      lcd.setCursor(0,1);
        lcd.print("=");

      posit = total;
      lcd.clear();
    lcd.setCursor(0,1);
        lcd.print("=");
      break;
    }
  }
  return second;}
```

Applications:

1) A simple arithmetic calculator is implemented in this project using Arduino UNO, matrix keypad and LCD.

2) Can be extended to perform complex calculations, but the logic must be carefully designed in order to solve those calculations. This may even result in increased number of switches.

3) Used to perform daily mathematical calculations etc.

4)Getting Knowledge of how a calculator works and also knowing how to make one using an online simulation platform like Tinkercad.

Conclusion:

Although this project is a simple calculator made with an Arduino, it mainly explains how to use a keypad to acquire characters and form a whole number out of individually entered characters. It also explains how to control an LCD connected to an Arduino and combine the two into a functional calculator.

This program is limited by the Arduino's platform variables and math, Arduino has limitations when it comes to big numbers and floating point. For example, when it comes to floating point numbers, you have float and double. Double should have bigger precision than float, but on Arduino that is not the case. So using double instead of float will not give higher precision unless you are using an Arduino Due.

Tinkercad link:

https://www.tinkercad.com/things/bDWDQw6qgFC-111056-shreerang-mhatre-basic-arduino-calculator/editel

References:

https://learn.sparkfun.com/tutorials/what-is-an-arduino/all

https://www.learnelectronicsindia.com/post/arduino-diy-calculator-using-a-4-4-keypad-and-16-2-lcd-display

https://www.engineersgarage.com/calculator-using-arduino/