**ПМГ „Академик Боян Петканчин“ – Хасково**

**НП „ИТ КАРИЕРА“**

**Модул 8 - Разработка на софтуер**

**Проект -Arduino Calculator**

**Project Link - https://www.tinkercad.com/things/5LCP2IIIVFK-calculator**

**Изготвено от Деян Димитров Делчев**

Съдържание

[Описание на проекта: 2](#_Toc69574585)

[Блокова схема на проекта: 3](#_Toc69574586)

[Електрическа схема: 3](#_Toc69574587)

[Монтажна схема: 4](#_Toc69574588)

[Списък съставни части: 4](#_Toc69574589)

[Сорс код и фунционалност: 5](#_Toc69574590)

[Описание на фунционлноста: 5](#_Toc69574591)

[Сорс код: 8](#_Toc69574592)

# Описание на проекта:

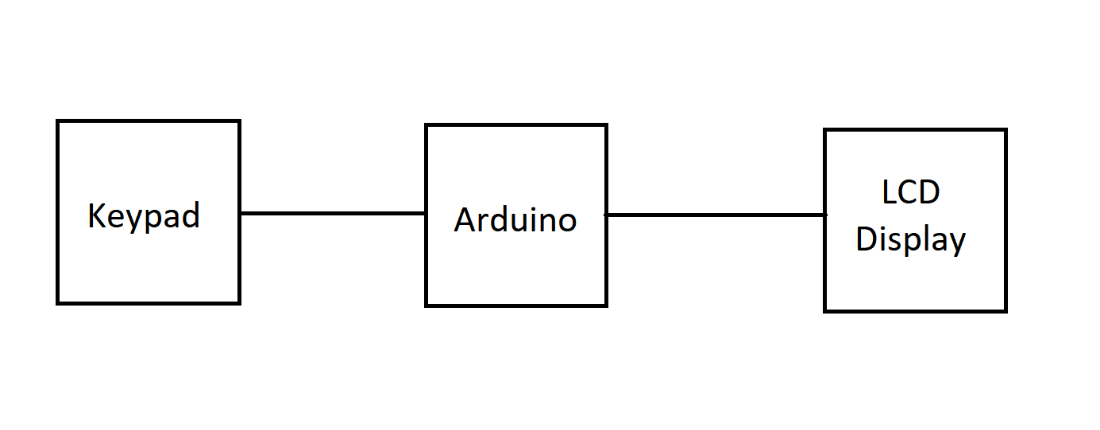
Проектът е разработен на онлайн средата Tinkercad, който представлява симулатор на електрически вериги.

Проектът е симулация на калкулатор.Контролера в симулацията е Arduino Uno R3. При натискане на бутон от клавиатурата, Arudino-то визуализира знака,който е натиснат,на LCD дисплей и пресмята израза при натискане на бутон „#“.При натискане на бутона дисплея се изчиства и показва меню, което показава кой бутон за какво отговаря.

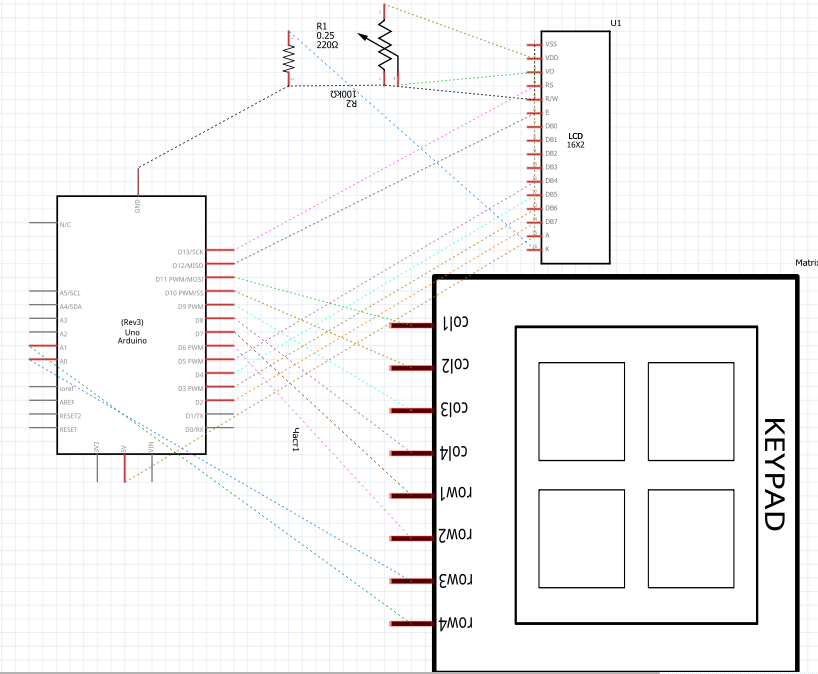


Цялата фунционалност се управлява от Arduino-то, което изпълнавя пограмата за калкулатора.(Сорс кодът е надолу). Програмата се грижи за Получаването от входа от клавиатурата, визуализацията на израза, и пресмятането.

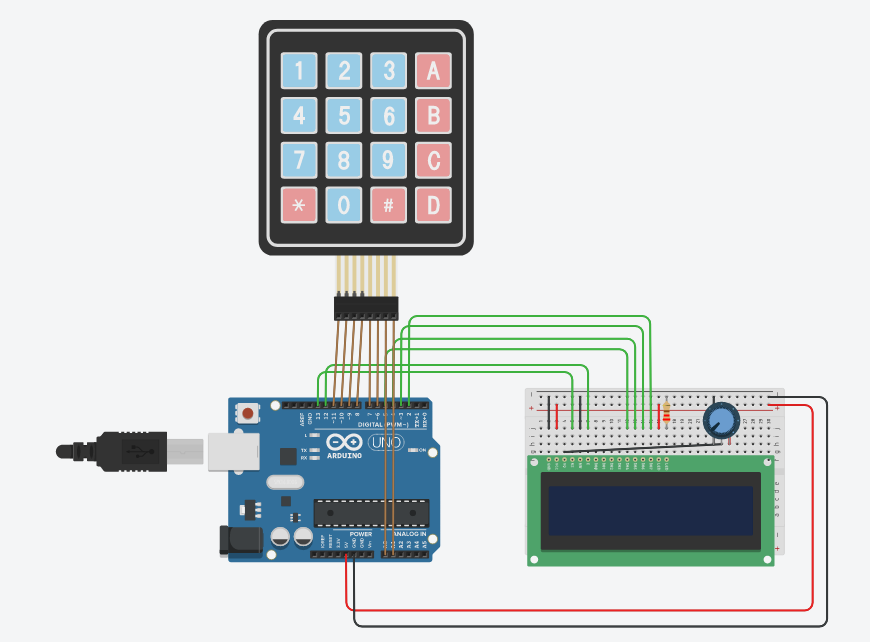
# Блокова схема на проекта:



# Електрическа схема:



# Монтажна схема:

****

# Списък съставни части:

|  |  |
| --- | --- |
| Компоненти | Количество |
| Arduino Uno R3 | 1 |
| KeyPad 4X4 | 1 |
| LCD Display | 1 |
| Резистор 220Ω | 1 |
| Потенциометър 10КΩ | 1 |

# Сорс код и фунционалност:

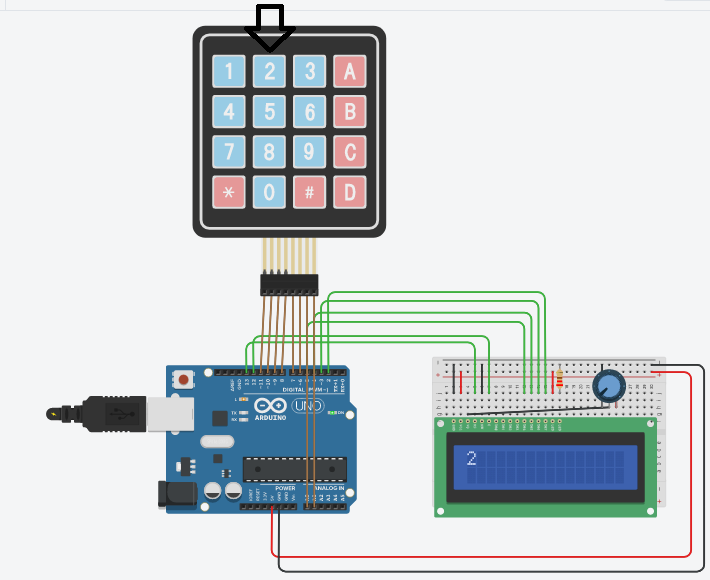
## Описание на фунционлноста:

При пускане на проекта излиза меню, което разяснява какво правят бутоните на клавиатурата. *(Фиг. 1)*

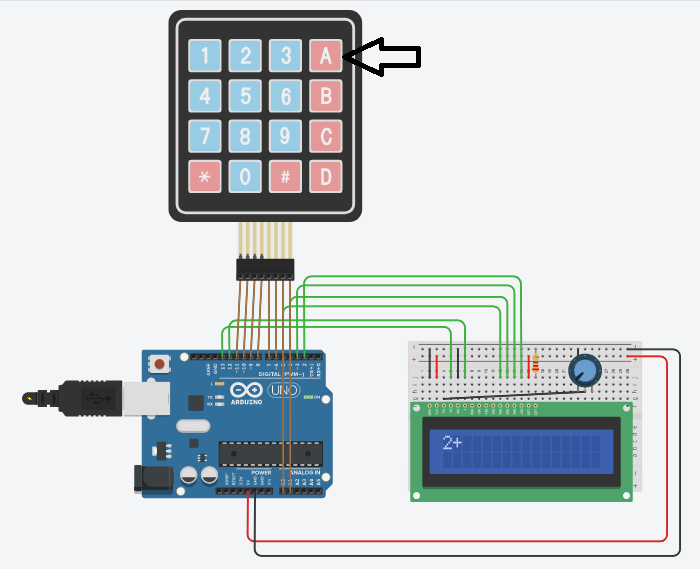
*(Фиг. 1)*

Клавиатурата се състои от 16 бутона *(Фиг. 2)*. Цифри от 0 до 9, букви от А до D и знаците \* #. След като натиснеш цифра се визуализира на дисплея *(Фиг. 3)*. Буквите отговарят на основните знаците запресмятане + - \* /, като А отговаря на +,В отговаря на – и така нататък. *(Фиг. 4)*. Бутонът # отговаря на =, пресмята стойноста на израза и я разпечатва. *(Фиг. 5 и 6) )*. Бутонът \* изчиства диспля и данинте от предишния израз и рапечатва менюто от *(Фиг. 1).* Калкулатора има възможността да пресмята сложни изрази с повече от един знак. Потенциуметърът служи за регулиране на яркоста на дисплея.

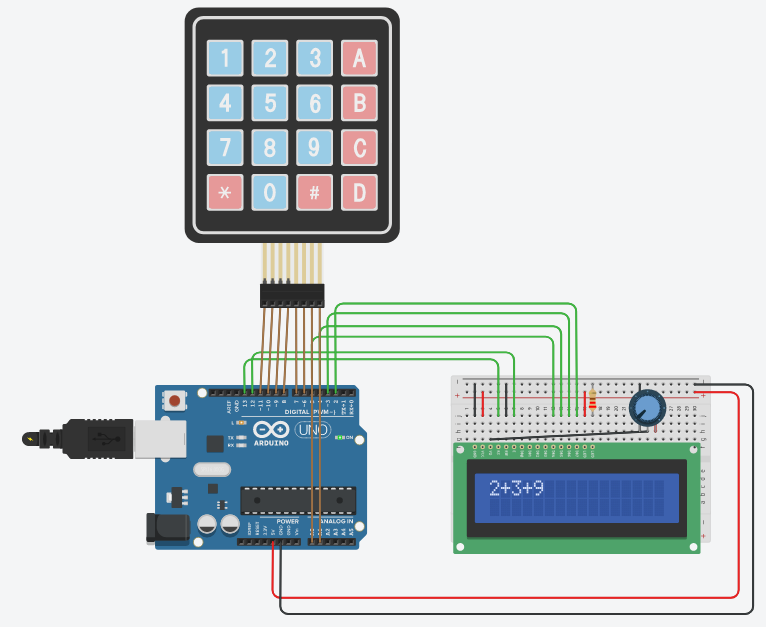
*(Фиг. 2)*

**

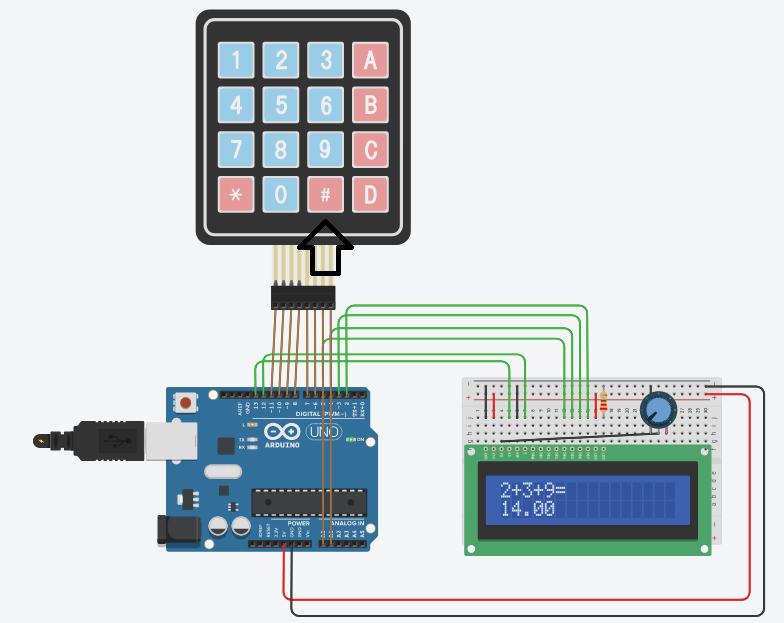
*(Фиг.3)*

*.* 

*(Фиг.4)*

**

*(Фиг.5)*

**

*(Фиг.6)*

## Сорс код:

#include<LiquidCrystal.h>

#include <Keypad.h>

LiquidCrystal lcd(13, 12, 5, 4, 3, 2);//display

int symbCount = 0;

char symb[8];//array,where symb are stored

double nums[8];//array,where numbers are stored

String temp = "";//string where current number is stored

char n = 0;

//initialize keypad

const byte ROWS = 4;

const byte COLS = 4;

String text = "";//string where the the expression is stored

bool stMenu=true;

char hexaKeys[ROWS][COLS] = {

{'1', '2', '3','A'},

{'4', '5', '6','B'},

{'7', '8', '9','C'},

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

};

byte rowPins[ROWS] = {11,10, 9, 8};

byte colPins[COLS] = {7,6,A0, A1};

Keypad keypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);//keypadd

void setup()

{

lcd.begin(16, 2);

Serial.begin(9600);

StartMenu();//pring starting menu

}

void loop()

{

if(FindNumber())printNumber();

}

void printNumber()//print the expression

{

lcd.setCursor(0, 0);

lcd.print(text);

delay(1000);

}

bool FindNumber()//main logic

{

char num = keypad.getKey();//read the press buttun

switch (num)//if it is a letter, it is converted to symbol or action

{

case 'A'://A->+;add + in symbols arry; add current number in the number array

num='+';

nums[symbCount] = temp.toDouble();

symb[symbCount] = num;

temp = "";

symbCount++;

break;

case 'B'://B->-;add - in symbols arry; add current number in the number array

num='-';

nums[symbCount] = temp.toDouble();

symb[symbCount] = num;

temp = "";

symbCount++;

break;

case 'C'://C->\*;add \* in symbols arry; add current number in the number array

num='\*';

nums[symbCount] =temp.toDouble();

symb[symbCount] = num;

temp = "";

symbCount++;

break;

case 'D'://D->/;add / in symbols arry; add current number in the number array

num='/';

nums[symbCount] = temp.toDouble();

symb[symbCount] = num;

temp = "";

symbCount++;

break;

case '\*'://\* calls method clean

num=' ';

Clean();

break;

case'#'://# calls calculate method

num='=';

nums[symbCount] = temp.toDouble();

temp="";

Calculate();

break;

default:

temp+=String(num);

break;

}

if(num&&num!=' ')//if button is not \* or #,then he print the new expresion

{

text+=String(num);

if(num!='=')lcd.clear();

return true;

}

return false;

}

void Calculate()//calculates the using rules of arithmetic \*/first,+- second

{

lcd.setCursor(0, 1);

int len = symbCount;

for (int i = 0; i < len; i++)//searchs for \*/

{

if (symb[i] == '\*')

{

double temp = nums[i] \* nums[i + 1];//if it finds,it calculate the answer

nums[i] = temp;//puts on first, delete the second spot and shifts the array

Shift(i+1 );

i--;

}

else if (symb[i] == '/')

{

double temp = (nums[i] / nums[i + 1]);//if it finds,it calculate the answer

nums[i] = temp;//puts on first, delete the second spot and shifts the array

Shift(i + 1);

i--;

}

}

for (int i = 0; i < len; i++)//searchs for +-

{

if (symb[i] == '+')

{

double temp = nums[i] + nums[i + 1];

nums[i] = temp;//puts on first, delete the second spot and shifts the array

Shift(i+1 );

i--;

}

else if (symb[i] == '-')

{

double temp = nums[i] - nums[i + 1];//if it finds,it calculate the answer

nums[i] = temp;//puts on first, delete the second spot and shifts the array

Shift(i + 1);

i--;

}

}

lcd.print(nums[0],8);

}

void Clean()//clears infotmation about the last expression

{

text="";

lcd.clear();

symbCount = 0;

for(int i=0;i<8;i++)

{Serial.println(nums[i]);

nums[i]=0;

symb[i]='\0';

}

StartMenu();

}

void Shift(int n)//shift the array from given index

{

for (int i = n; i <= symbCount; i++)

{

nums[i ] = nums[i+1];

symb[i - 1] = symb[i];

}

nums[symbCount] = 0;

symb[symbCount-1] = '\0';

symbCount--;

}

void StartMenu()//print start menu

{

lcd.setCursor(0, 0);

lcd.print("A->+ B->- C->\*");

lcd.setCursor(0, 1);

lcd.print("D->/ #->= \*->CE");

stMenu=true;

}void Shift(int n)

{

for (int i = n; i <= symbCount; i++)

{

nums[i ] = nums[i+1];

symb[i - 1] = symb[i];

}

nums[symbCount] = 0;

symb[symbCount-1] = '\0';

symbCount--;

}

void StartMenu()

{

lcd.setCursor(0, 0);

lcd.print("A->+ B->- C->\*");

lcd.setCursor(0, 1);

lcd.print("D->/ #->= \*->CE");

stMenu=true;

}