LAB-05

CSE2020

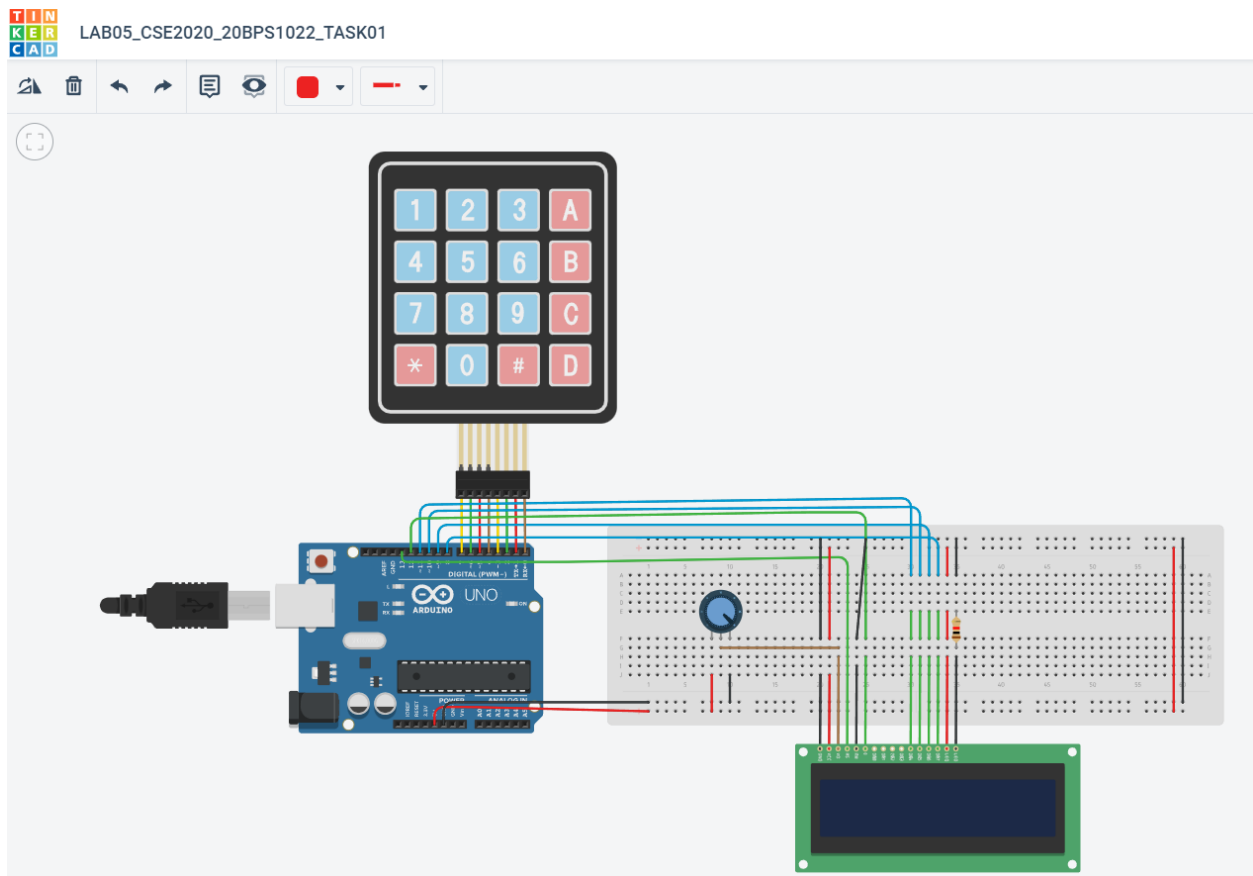# INTRODUCTION TO CPS LAB

Name: Preyash

Reg No.: 20BPS1022   **Date: February 07, 2022**

Task 1: Scientific Calculator using ARDUINO UNO R3, KEYPAD, LCD, RESISTOR, POTENTIOMETER.

Circuit:

Code:

```cpp
#include <Keypad.h>

#include <Wire.h>

#include <LiquidCrystal.h>

LiquidCrystal lcd(13, 12, 11, 10, 9, 8);

long first = 0;

long second = 0;

double total = 0;

char customKey;

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] = {7,6,5,4};

byte colPins[COLS] = {3,2,1,0};

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

void setup()

{

lcd.begin(16, 2);

for(int i=0;i<=3;i++);

lcd.setCursor(0,0);

  lcd.print("Calculator");

delay(4000);
```

```
lcd.clear();

lcd.setCursor(0, 0);

}

void loop()

{

  customKey = customKeypad.getKey();

  switch(customKey)

  {

  case '0' ... '9':

    lcd.setCursor(0,0);

    first = first * 10 + (customKey - '0');

    lcd.print(first);

    break;


  case '+':

    first = (total != 0 ? total : first);

    lcd.setCursor(0,1);

    lcd.print("+");

    second = SecondNumber();

    total = first + second;

    lcd.setCursor(0,3);

    lcd.print(total);

    first = 0, second = 0;

    break;


  case '-':

    first = (total != 0 ? total : first);
```

```
     lcd.setCursor(0,1);

     lcd.print("-");

     second = SecondNumber();

     total = first - second;

     lcd.setCursor(0,3);

     lcd.print(total);

     first = 0, second = 0;

     break;


   case '*':

    first = (total != 0 ? total : first);

    lcd.setCursor(0,1);

    lcd.print("*");

     second = SecondNumber();

     total = first * second;

    lcd.setCursor(0,3);

    lcd.print(total);

    first = 0, second = 0;

     break;


   case '/':

    first = (total != 0 ? total : first);

    lcd.setCursor(0,1);

    lcd.print("/");

     second = SecondNumber();

    lcd.setCursor(0,3);
```
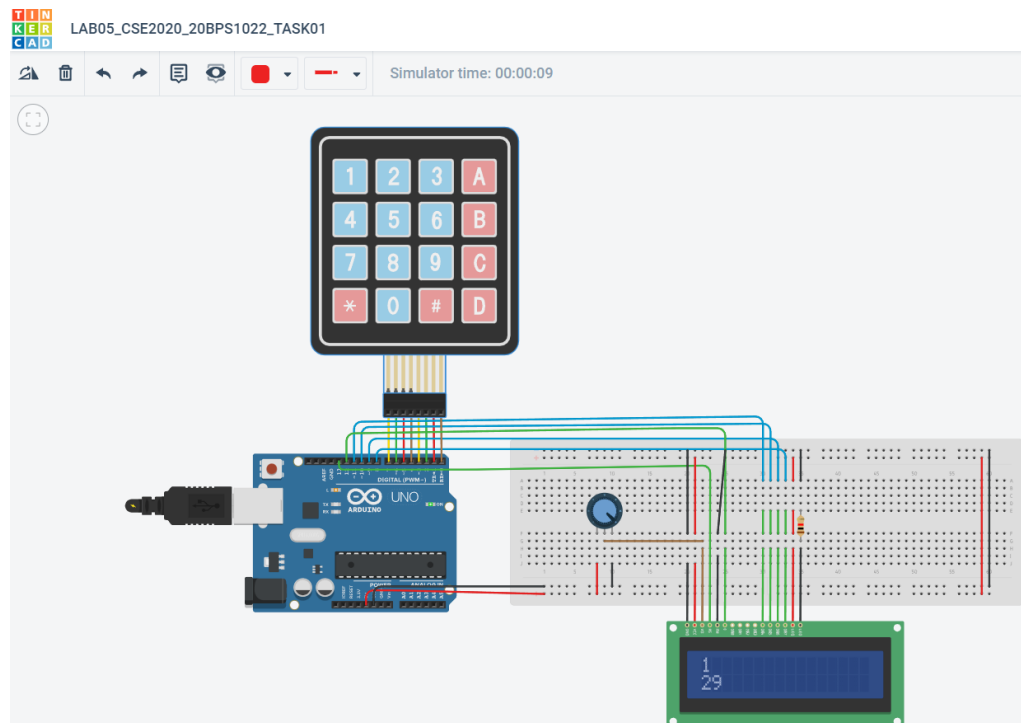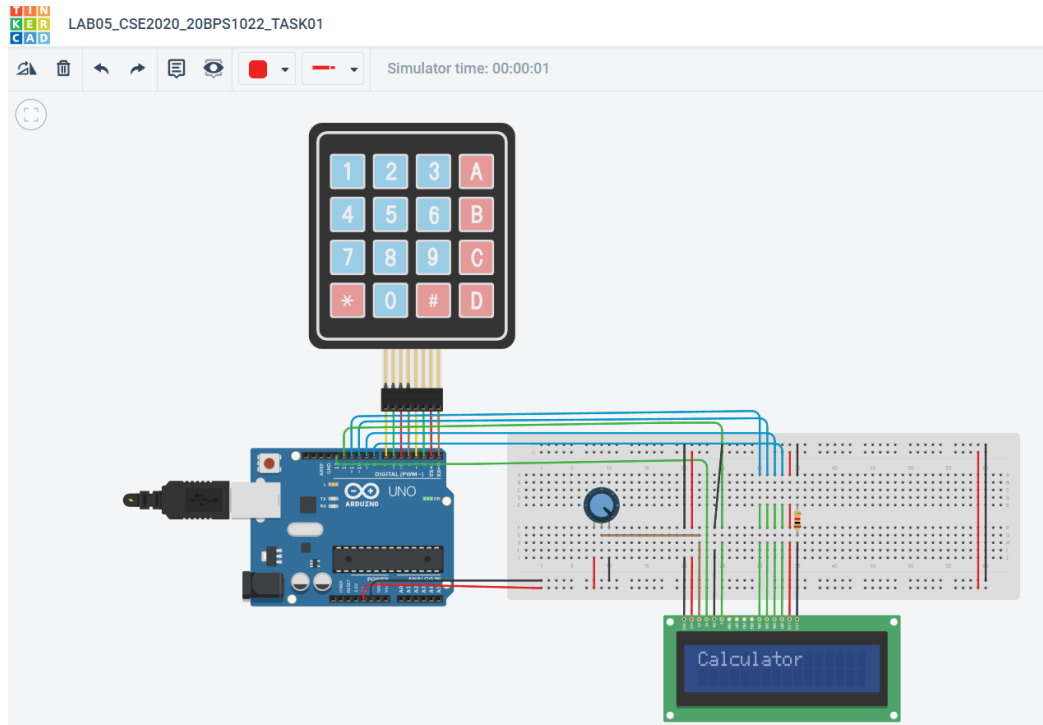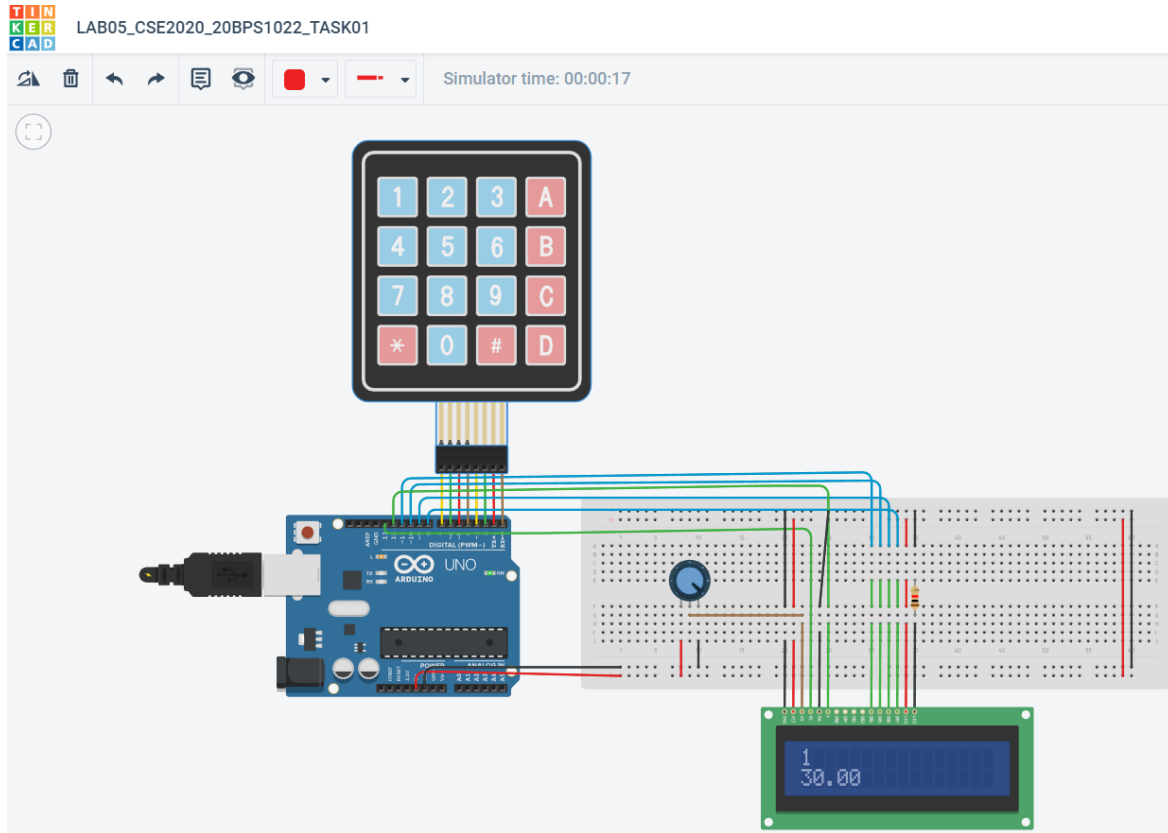
```
      second == 0 ? lcd.print("Invalid") : total = (float)first / (float)second;

      lcd.print(total);

      first = 0, second = 0;

      break;


    case 'C':

      total = 0;

      lcd.clear();

      break;

   }

 }

 long SecondNumber()

 {

   while( 1 )

   {

     customKey = customKeypad.getKey();

     if(customKey >= '0' && customKey <= '9')

     {

       second = second * 10 + (customKey - '0');

       lcd.setCursor(0,2);

       lcd.print(second);

     }

     if(customKey == '=') break;

   }

  return second;

 }
```
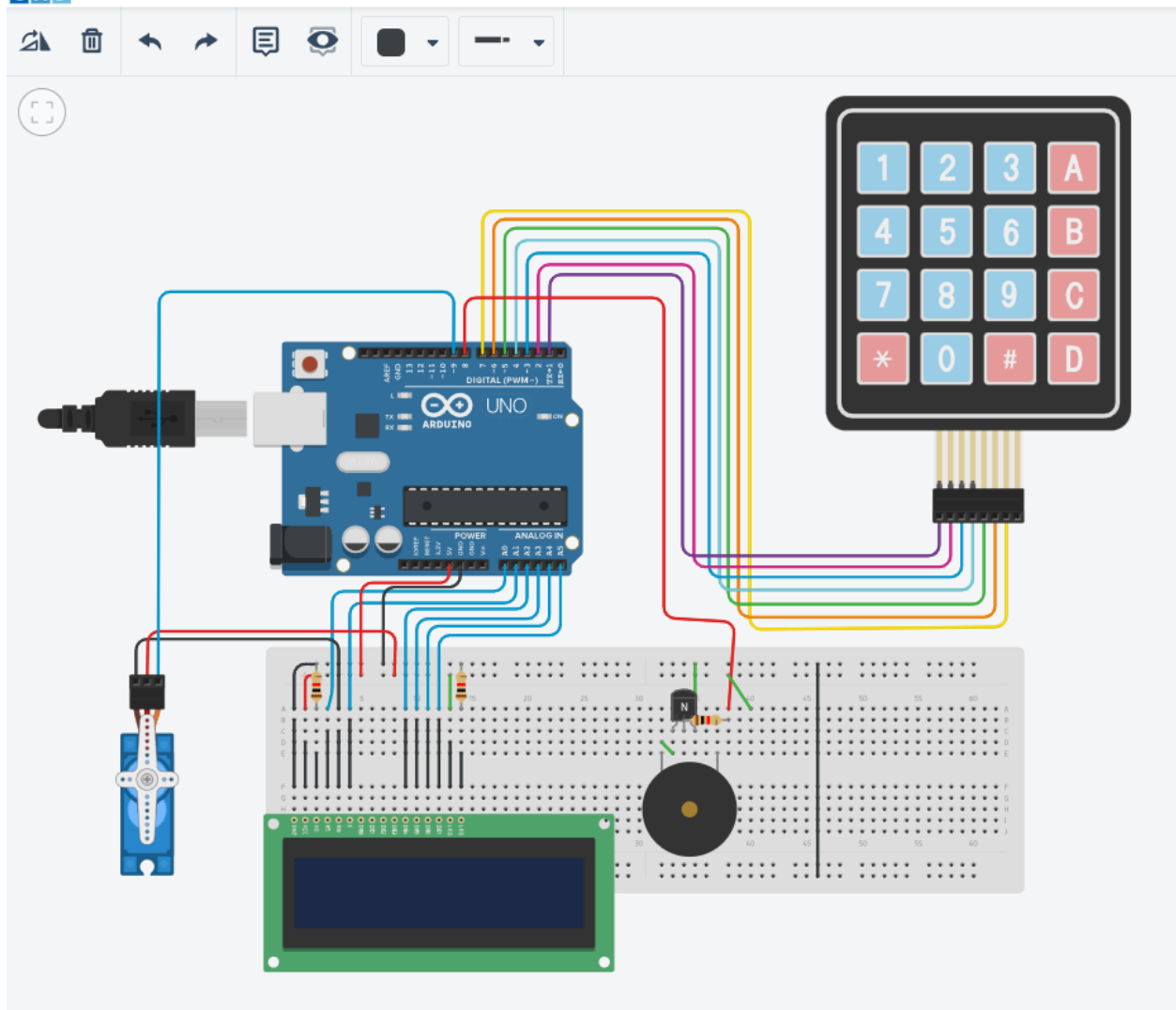
Output:

Adding 1+29

Sum=30

Link:

https://www.tinkercad.com/things/513Ua7vFO0A-lab05cse202020bps1022task01/editel?sharecode=2dr6FXtb7PfGetusqG8mojUCSJ0GdGXIW16ewmNLl4s

Task 2: **Password based door locking system using Arduino, LCD Display, Servo Motor, 4x4 Keypad.**

Circuit:

Code:

```cpp
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>
Servo myservo;
int pos=0;
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);
const byte rows=4;
const byte cols=3;
char key[rows][cols]={
{'1','2','3'},
{'4','5','6'},
{'7','8','9'},
{'*','0','#'}
};
byte rowPins[rows]={1,2,3,4};
byte colPins[cols]={5,6,7};
Keypad keypad= Keypad(makeKeymap(key),rowPins,colPins,rows,cols);
char* password="0129";
int currentposition=0;
int redled=10;
int greenled=11;
int buzz=8;
int invalidcount=12;
void setup()
{
```

```
displayscreen();

Serial.begin(9600);

pinMode(redled, OUTPUT);

pinMode(greenled, OUTPUT);

pinMode(buzz, OUTPUT);

myservo.attach(9);

lcd.begin(16,2);

}

void loop()

{

if( currentposition==0)

{

displayscreen();

}

int l ;

char code=keypad.getKey();

if(code!=NO_KEY)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("PASSWORD:");

lcd.setCursor(7,1);

lcd.print(" ");

lcd.setCursor(7,1);

for(l=0;l<=currentposition;++l)

{

lcd.print("*");
```

```
keypress();

}

if (code==password[currentposition])

{

++currentposition;

if(currentposition==4)

{

unlockdoor();

currentposition=0;

}

}

else

{

++invalidcount;

incorrect();

currentposition=0;

}

}

}


void unlockdoor()

{

delay(900);

lcd.setCursor(0,0);

lcd.println(" ");

lcd.setCursor(1,0);

lcd.print("ACCESS GRANTED");
```

```
lcd.setCursor(4,1);

lcd.println("WELCOME!!");

lcd.setCursor(15,1);

lcd.println(" ");

lcd.setCursor(16,1);

lcd.println(" ");

lcd.setCursor(14,1);

lcd.println(" ");

lcd.setCursor(13,1);

lcd.println(" ");

unlockbuzz();


for(pos = 180; pos>=0; pos-=5)

{

myservo.write(pos);

delay(5);

}

delay(2000);




delay(1000);

counterbeep();


delay(1000);


for(pos = 0; pos <= 180; pos +=5)
```

```arduino
myservo.write(pos);

delay(15);



currentposition=0;



lcd.clear();

displayscreen();



}



void incorrect()

{

delay(500);

lcd.clear();

lcd.setCursor(1,0);

lcd.print("CODE");

lcd.setCursor(6,0);

lcd.print("INCORRECT");

lcd.setCursor(15,1);

lcd.println(" ");

lcd.setCursor(4,1);

lcd.println("GET AWAY!!!");



lcd.setCursor(13,1);

lcd.println(" ");

Serial.println("INCORRECT, YOU ARE UNAUTHORIZED");
```

```
digitalWrite(redled, HIGH);

digitalWrite(buzz, HIGH);

delay(3000);

lcd.clear();

digitalWrite(redled, LOW);

digitalWrite(buzz,LOW);

displayscreen();

}

void clearscreen()

{

lcd.setCursor(0,0);

lcd.println(" ");

lcd.setCursor(0,1);

lcd.println(" ");

lcd.setCursor(0,2);

lcd.println(" ");

lcd.setCursor(0,3);

lcd.println(" ");

}


void keypress()

{




digitalWrite(buzz, HIGH);

delay(50);
```

```
digitalWrite(buzz, LOW);

}


void displayscreen()

{


lcd.setCursor(0,0);

lcd.println("ENTER THE");

lcd.setCursor(1 ,1);


lcd.println(" PASSWORD:");

}


void armservo()

{


for (pos=180;pos<=180;pos+=50)

{

myservo.write(pos);

delay(5);

}

delay(5000);


for(pos=180;pos>=0;pos-=50)

{

myservo.write(pos);

}
```

```
}
void unlockbuzz()
{

digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(200);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
digitalWrite(buzz, HIGH);
delay(80);
digitalWrite(buzz, LOW);
delay(80);
}

void counterbeep()
{
delay(1200);
```

```
lcd.clear();

digitalWrite(buzz, HIGH);


lcd.setCursor(2,15);

lcd.println(" ");

lcd.setCursor(2,14);

lcd.println(" ");

lcd.setCursor(2,0);

delay(200);

lcd.println("GET IN WITHIN:::");


lcd.setCursor(4,1);

lcd.print("5");

delay(200);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

digitalWrite(buzz,LOW);

delay(1000);

//2

digitalWrite(buzz, HIGH);

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

lcd.setCursor(4,1); //2

lcd.print("4");

delay(100);
```

```
lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

digitalWrite(buzz,LOW);

delay(1000);

//3

digitalWrite(buzz, HIGH);

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

lcd.setCursor(4,1); //3

lcd.print("3");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

digitalWrite(buzz,LOW);

delay(1000);

//4

digitalWrite(buzz, HIGH);

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");

lcd.setCursor(4,1); //4

lcd.print("2");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN:");
```

```
digitalWrite(buzz,LOW);

delay(1000);

//

digitalWrite(buzz, HIGH);

lcd.setCursor(4,1);

lcd.print("1");

delay(100);

lcd.clear();

lcd.setCursor(2,0);

lcd.println("GET IN WITHIN::");

digitalWrite(buzz,LOW);

delay(1000);

//5

digitalWrite(buzz, HIGH);

delay(40);

digitalWrite(buzz,LOW);

delay(40);

digitalWrite(buzz, HIGH);

delay(40);

digitalWrite(buzz,LOW);

delay(40);

digitalWrite(buzz, HIGH);

delay(40);

digitalWrite(buzz,LOW);

delay(40);

digitalWrite(buzz, HIGH);

delay(40);
```

```
digitalWrite(buzz,LOW);

lcd.clear();

lcd.setCursor(2,0);

lcd.print("RE-LOCKING");

delay(500);

lcd.setCursor(12,0);

lcd.print(".");

delay(500);

lcd.setCursor(13,0);

lcd.print(".");

delay(500);

lcd.setCursor(14,0);

lcd.print(".");

delay(400);

lcd.clear();

lcd.setCursor(4,0);

lcd.print("LOCKED!");

delay(440);

}
```
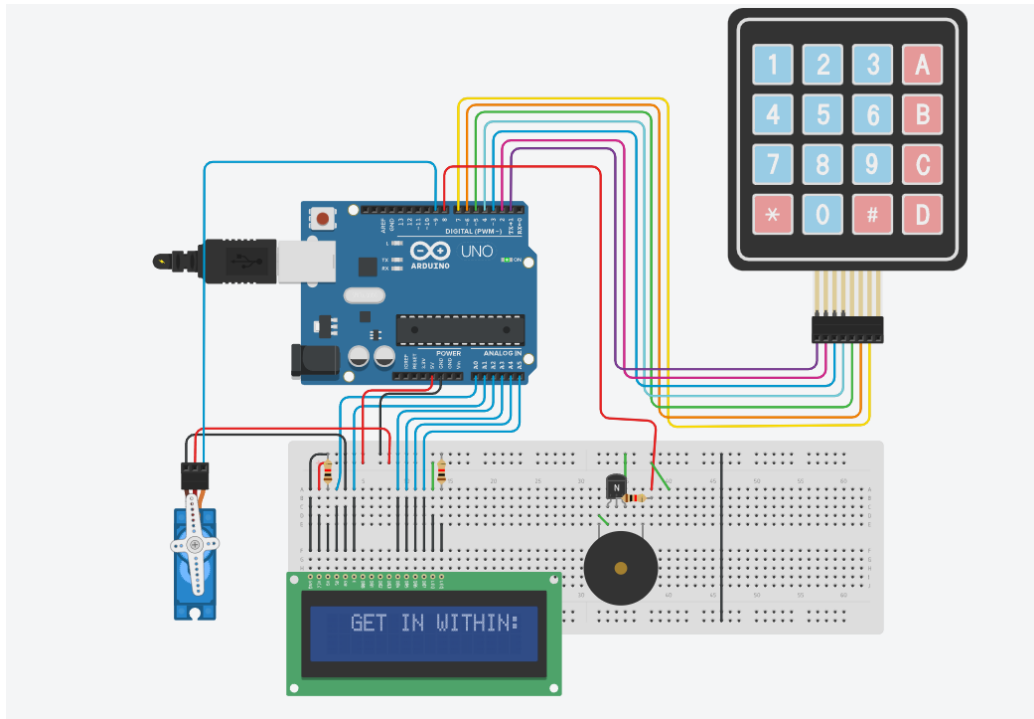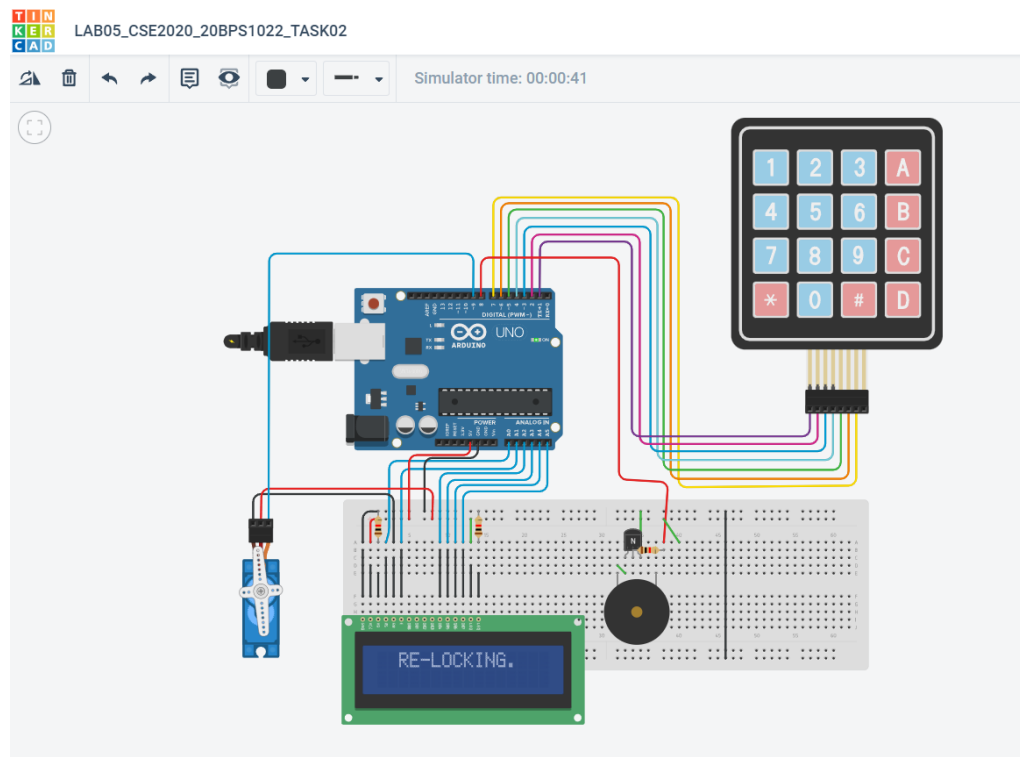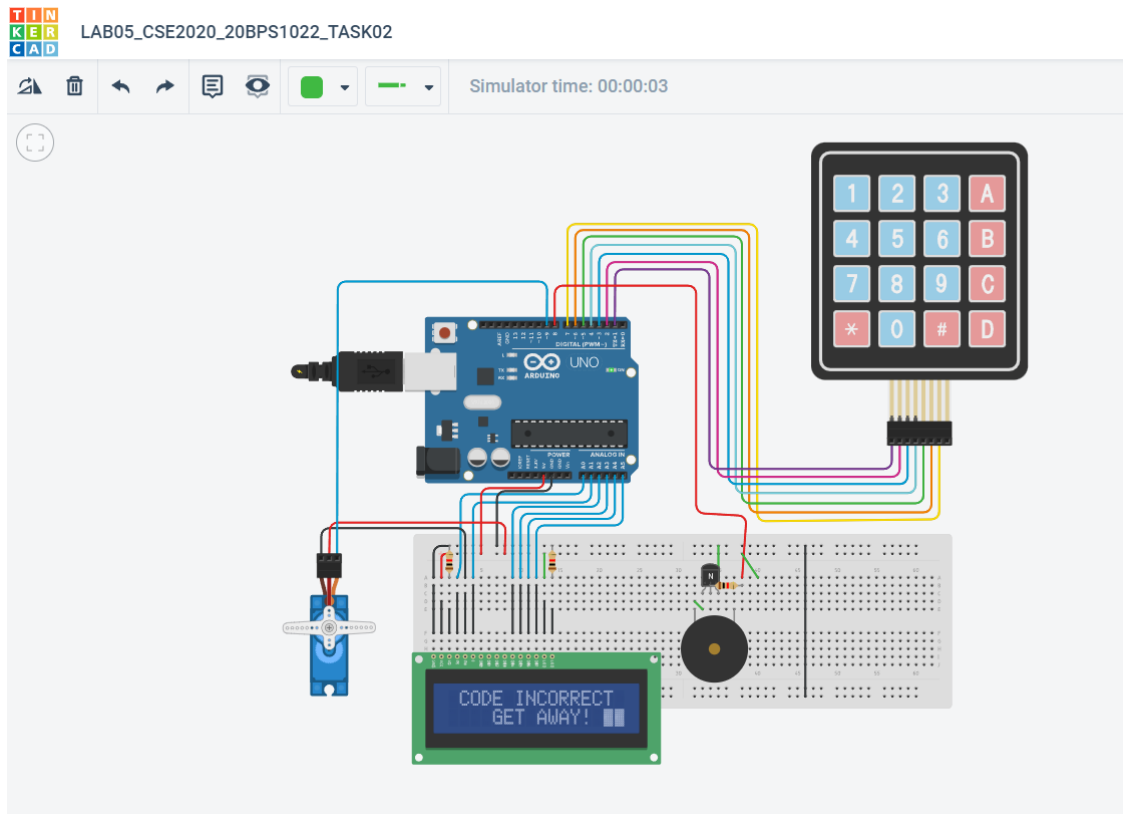
Output:
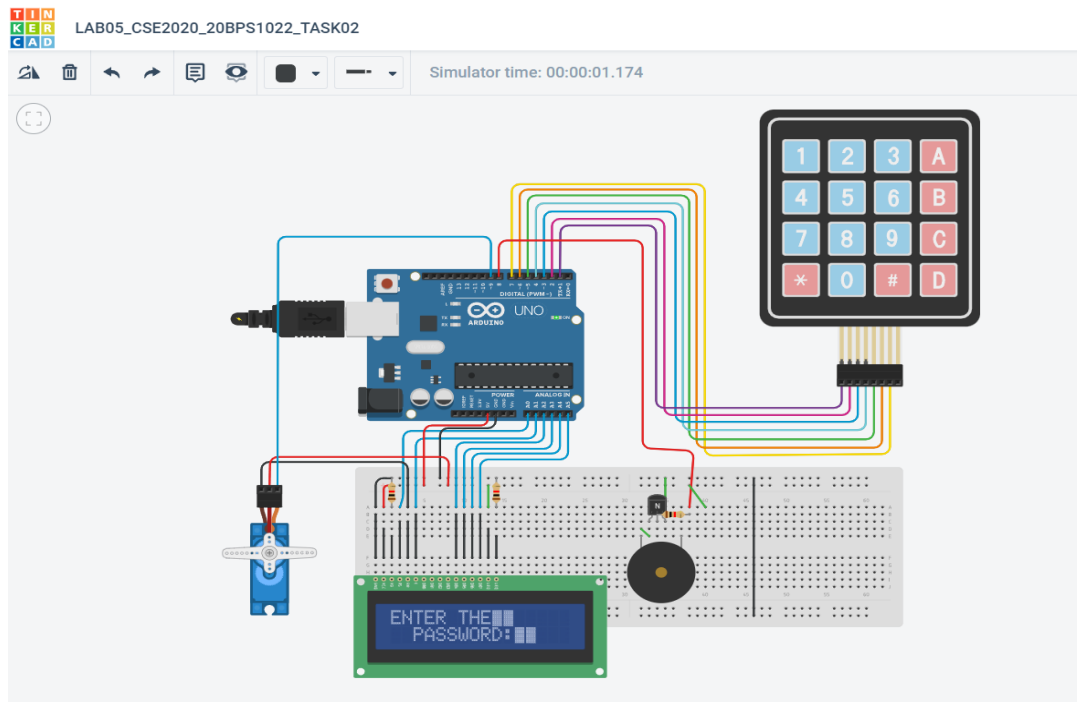
Entering the password
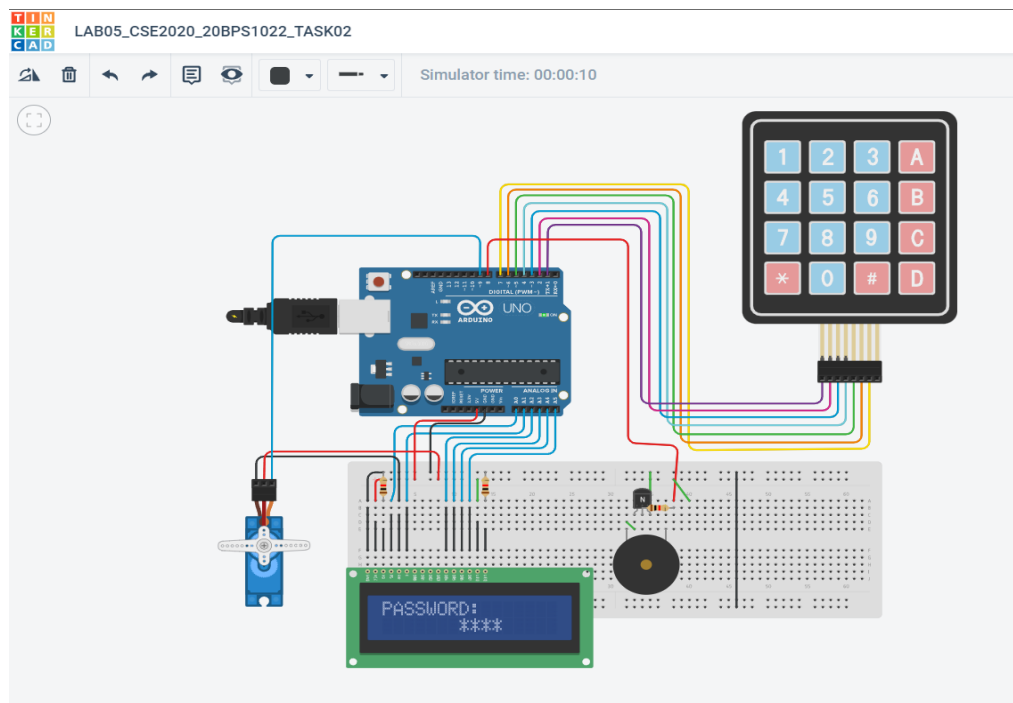
Entered the password

Closing the door

Closed the door

Incorrect password

Result:

All the circuits were successfully built as required.