

IOT102 Arduino Project

Nguyễn Trí Trường Sơn - HE151167 - AI1603

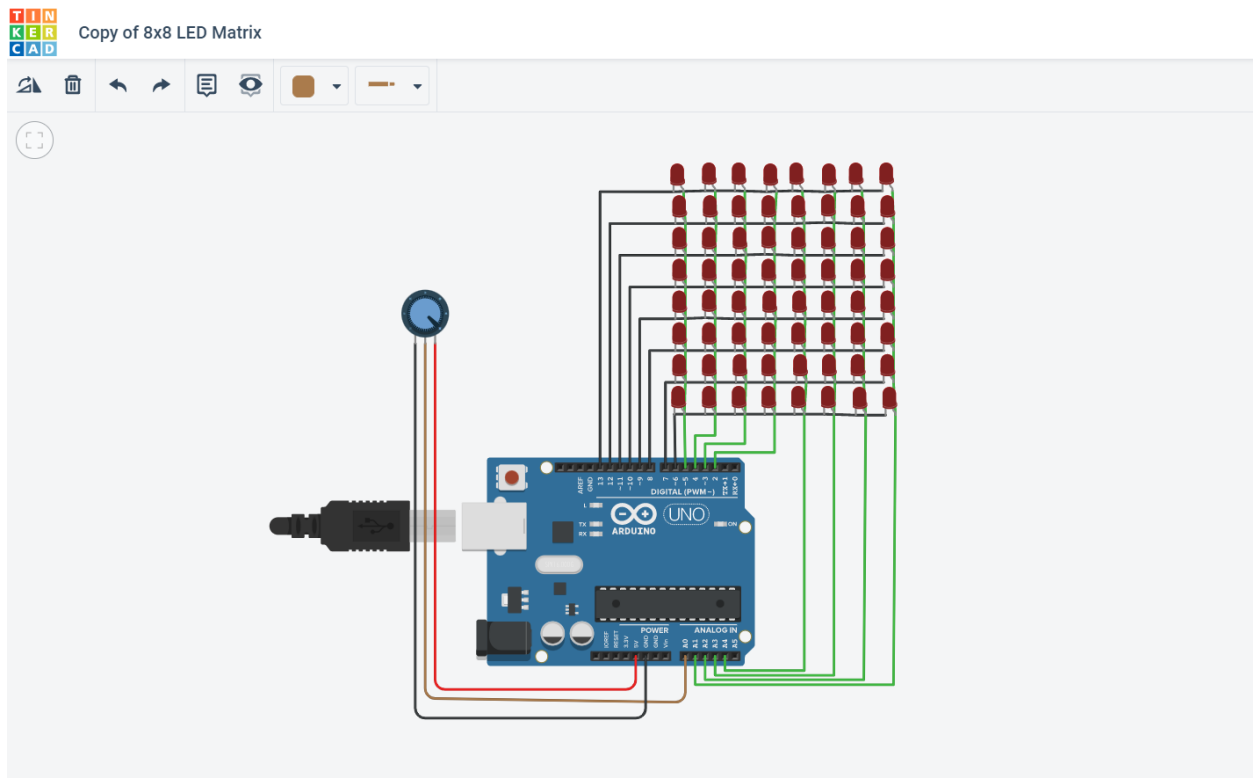
#Project 1: Điều chỉnh số hiển thị bằng led matrix 8x8

Mô tả:

Đây là một số hiệu ứng cơ bản với sự kết hợp giữa led matrix 8x8 và biến trở. Bằng cách xoay biến trở thì một số sẽ được hiển thị tương ứng với góc quay. Góc ngoài cùng bên trái tương đương với số 0 và bên phải tương đương với số 9.

Phần cứng:

- Arduino UNO
- Breadboard
- Led Matrix 8x8
- Potentiometer



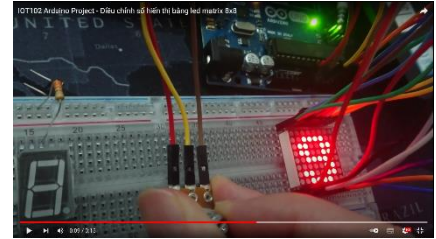
Note: Mạch trên em thấy khá giống nên em lấy tạm nhưng mà các cổng nó sai nhé thầy. Em hơi lười nên không muốn sửa.

Link:

- Tinkercard: <https://www.tinkercad.com/things/4IOWj7ACIYI-copy-of-8x8-led-matrix/editel?sharecode=DDecePrjSYVOhIme4jljxwPY9h2vQz43tVD0Lucqj0k>

- Youtube Video: <https://youtu.be/TKmVzbOTiSA>

Code:



```
#define ROW_1 2
#define ROW_2 7
#define ROW_3 19
#define ROW_4 5
#define ROW_5 13
#define ROW_6 18
#define ROW_7 12
#define ROW_8 16

#define COL_1 9
#define COL_2 8
#define COL_3 4
#define COL_4 17
#define COL_5 3
#define COL_6 10
#define COL_7 11
#define COL_8 6

const byte rows[] = {
  ROW_1, ROW_2, ROW_3, ROW_4, ROW_5, ROW_6, ROW_7, ROW_8
};
const byte cols[] = {
  COL_1, COL_2, COL_3, COL_4, COL_5, COL_6, COL_7, COL_8
};

byte so0[] = {
  B11111111,
  B11000011,
  B10011001,
  B10010001,
  B10001001,
  B10011001,
  B10011001,
  B11000011
};
byte so1[] = {
  B11111111,
  B11100111,
```

```
B11100111,  
B11000111,  
B11100111,  
B11100111,  
B11100111,  
B10000001  
};  
byte so2[] = {  
    B11111111,  
    B11000011,  
    B10011001,  
    B11111001,  
    B11110011,  
    B11001111,  
    B10011111,  
    B10000001  
};  
byte so3[] = {  
    B11111111,  
    B11000011,  
    B10011001,  
    B11111001,  
    B11100011,  
    B11111001,  
    B10011001,  
    B11000011  
};  
byte so4[] = {  
    B11111111,  
    B11110011,  
    B11100011,  
    B11010011,  
    B10110011,  
    B10000001,  
    B11110011,  
    B11110011  
};  
byte so5[] = {  
    B11111111,  
    B10000001,  
    B10011111,  
    B10000011,  
    B11111001,  
    B11111001,  
    B10011001,  
    B11000011  
};  
byte so6[] = {
```

```
B11111111,  
B11000011,  
B10011001,  
B10011111,  
B10000011,  
B10011001,  
B10011001,  
B11000011  
};  
byte so7[] = {  
  B11111111,  
  B10000001,  
  B10011001,  
  B11110011,  
  B11110011,  
  B11100111,  
  B11100111,  
  B11100111,  
  B11100111  
};  
byte so8[] = {  
  B11111111,  
  B11000011,  
  B10011001,  
  B10011001,  
  B11000011,  
  B10011001,  
  B10011001,  
  B10011001,  
  B11000011  
};  
byte so9[] = {  
  B11111111,  
  B11000011,  
  B10011001,  
  B10011001,  
  B11000001,  
  B11111001,  
  B10011001,  
  B11000011  
};  
  
void setup()  
{  
  // Open serial port  
  Serial.begin(9600);  
  
  // Set all used pins to OUTPUT  
  // the display will be very dim.  
  for (byte i = 2; i <= 13; i++)
```

```

pinMode(i, OUTPUT);
pinMode(A2, OUTPUT);
pinMode(A3, OUTPUT);
pinMode(A4, OUTPUT);
pinMode(A5, OUTPUT);
}
void drawMatrix(byte data[]){
// Turn on each row in series
for (byte i = 0; i < 8; i++) {
digitalWrite(rows[i], HIGH); //initiate whole row
// Turn on each point in row
for (byte a = 0; a < 8; a++){
// if You set (~data[i] >> a) then You will have positive
digitalWrite(cols[a], (data[i] >> a) & 0x01); // initiate whole column
digitalWrite(cols[a], HIGH); // reset whole column
}
digitalWrite(rows[i], LOW); // reset whole row
// otherwise last row will intersect with next row
}
}
void hienthi(int count){
switch (count){
case 0:
{
drawMatrix(so0);
break;
}
case 1:
{
drawMatrix(so1);
break;
}
case 2:
{
drawMatrix(so2);
break;
}
case 3:
{
drawMatrix(so3);
break;
}
case 4:
{
drawMatrix(so4);
break;
}
case 5:

```

```
{
    drawMatrix(so5);
    break;
}
case 6:
{
    drawMatrix(so6);
    break;
}
case 7:
{
    drawMatrix(so7);
    break;
}
case 8:
{
    drawMatrix(so8);
    break;
}
case 9:
{
    drawMatrix(so9);
    break;
}

}

}

void loop() {
    int a = analogRead(A0);
    int count = a/110;
    hienthi(count);
}
```

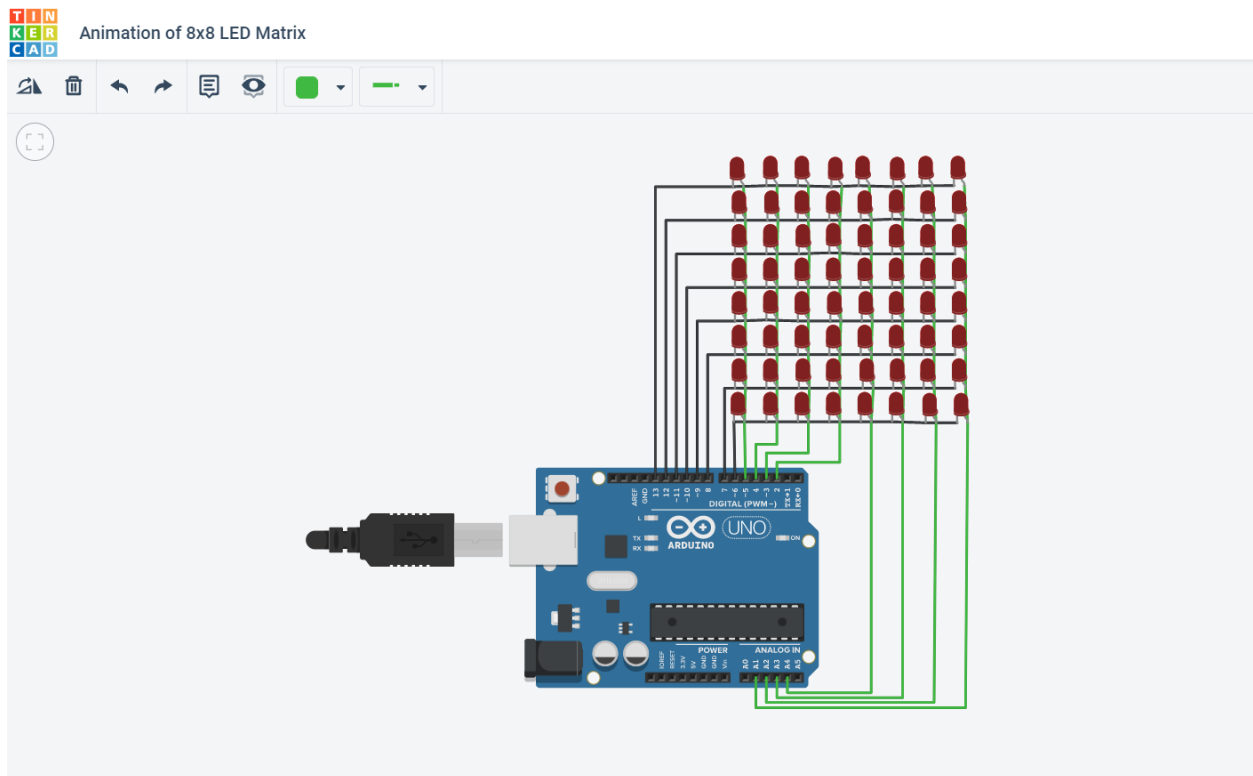
#Project 1.2: Hình ảnh động, chữ chạy từ phải sang trái

Mô tả:

Đây là một hiệu ứng cơ bản của led matrix 8x8. Chữ SON sẽ chạy lần lượt từ phải qua trái.

Phần cứng:

- Arduino UNO
- Breadboard
- Led Matrix 8x8



Link:

- Tinkercard: <https://www.tinkercad.com/things/j76oXijHgZ2-copy-of-copy-of-8x8-led-matrix/editel?sharecode=uyBPNn2PjCCA9QL93XsxWiMiODDpn7moXglwa1-ub84>

- Youtube Video: <https://youtu.be/0od36968Vms>

Code:

```
#define ROW_1 2
#define ROW_2 7
#define ROW_3 19
#define ROW_4 5
#define ROW_5 13
```



```

#define ROW_6 18
#define ROW_7 12
#define ROW_8 16

#define COL_1 9
#define COL_2 8
#define COL_3 4
#define COL_4 17
#define COL_5 3
#define COL_6 10
#define COL_7 11
#define COL_8 6

const byte rows[] = {
    ROW_1, ROW_2, ROW_3, ROW_4, ROW_5, ROW_6, ROW_7, ROW_8
};
const byte cols[] = {
    COL_1, COL_2, COL_3, COL_4, COL_5, COL_6, COL_7, COL_8
};

const byte chuSon[][8] = {
{
    B11111111,
    B11111111,
    B11111110,
    B11111110,
    B11111111,
    B11111111,
    B11111110,
    B11111111
}, {
    B11111111,
    B11111110,
    B11111100,
    B11111100,
    B11111110,
    B11111111,
    B11111100,
    B11111110
}, {
    B11111111,
    B11111100,
    B11111001,
    B11111001,
    B11111100,
    B11111111,
    B11111001,
    B11111100
}

```



```
},{  
  B11111111,  
  B11111000,  
  B11110011,  
  B11110011,  
  B11111000,  
  B11111111,  
  B11110011,  
  B11111000
```

```
},{  
  B11111111,  
  B11110000,  
  B11100110,  
  B11100111,  
  B11110000,  
  B11111110,  
  B11100110,  
  B11110000
```

```
},{  
  B11111111,  
  B11100001,  
  B11001100,  
  B11001111,  
  B11100001,  
  B11111100,  
  B11001100,  
  B11100001
```

```
},{  
  B11111111,  
  B11000011,  
  B10011001,  
  B10011111,  
  B11000011,  
  B11111001,  
  B10011001,  
  B11000011
```

```
},{  
  B11111111,  
  B10000111,  
  B00110011,  
  B00111111,  
  B10000111,  
  B11110011,  
  B00110011,  
  B10000111
```

```
},{  
  B11111111,  
  B00001111,
```

```
B01100110,  
B01111110,  
B00001110,  
B11100110,  
B01100110,  
B00001111  
,{  
B11111111,  
B00011110,  
B11001100,  
B11111100,  
B00011100,  
B11001100,  
B11001100,  
B00011110  
,{  
B11111111,  
B00111100,  
B10011001,  
B11111001,  
B00111001,  
B10011001,  
B10011001,  
B00111100  
,{  
B11111111,  
B01111000,  
B00110011,  
B11110011,  
B01110011,  
B00110011,  
B00110011,  
B01111000  
,{  
B11111111,  
B11110000,  
B01100110,  
B11100110,  
B11100110,  
B01100110,  
B01100110,  
B11110000  
,{  
B11111111,  
B11100001,  
B11001100,  
B11001100,  
B11001100,
```

```
B11001100,  
B11001100,  
B11100001  
,  
{  
B11111111,  
B11000011,  
B10011001,  
B10011001,  
B10011001,  
B10011001,  
B10011001,  
B10011001,  
B11000011  
,  
{  
B11111111,  
B10000111,  
B00110011,  
B00110011,  
B00110011,  
B00110011,  
B00110011,  
B00110011,  
B10000111  
,  
{  
B11111111,  
B00001110,  
B01100110,  
B01100110,  
B01100110,  
B01100110,  
B01100110,  
B01100110,  
B00001110  
,  
{  
B11111111,  
B00111001,  
B10011000,  
B10011000,  
B10011001,  
B10011001,  
B10011001,  
B00111001  
,  
{  
B11111111,  
B01110011,  
B00110001,  
B00110000,  
B00110010,  
B00110011,  
B00110011,  
B01110011
```

```
},{  
  B11111111,  
  B11100111,  
  B01100011,  
  B01100001,  
  B01100100,  
  B01100110,  
  B01100111,  
  B11100111
```

```
},{  
  B11111111,  
  B11001110,  
  B11000110,  
  B11000010,  
  B11001000,  
  B11001100,  
  B11001110,  
  B11001110
```

```
},{  
  B11111111,  
  B10011100,  
  B10001100,  
  B10000100,  
  B10010000,  
  B10011000,  
  B10011100,  
  B10011100
```

```
},{  
  B11111111,  
  B00111001,  
  B00011001,  
  B00001001,  
  B00100001,  
  B00110001,  
  B00111001,  
  B00111001
```

```
},{  
  B11111111,  
  B01110011,  
  B00110011,  
  B00010011,  
  B01000011,  
  B01100011,  
  B01110011,  
  B01110011
```

```
},{  
  B11111111,  
  B11100111,
```

```
B01100111,  
B00100111,  
B10000111,  
B11000111,  
B11100111,  
B11100111  
,{  
B11111111,  
B11001111,  
B11001111,  
B01001111,  
B00001111,  
B10001111,  
B11001111,  
B11001111  
,{  
B11111111,  
B10011111,  
B10011111,  
B10011111,  
B00011111,  
B00011111,  
B10011111,  
B10011111  
,{  
B11111111,  
B00111111,  
B00111111,  
B00111111,  
B00111111,  
B00111111,  
B00111111,  
B00111111  
,{  
B11111111,  
B01111111,  
B01111111,  
B01111111,  
B01111111,  
B01111111,  
B01111111,  
B01111111  
,{  
B11111111,  
B11111111,  
B11111111,  
B11111111,  
B11111111,
```

```

B11111111,
B11111111,
B11111111
});
void setup()
{
    // Open serial port
    Serial.begin(9600);

    // Set all used pins to OUTPUT
    // This is very important! If the pins are set to input
    // the display will be very dim.
    for (byte i = 2; i <= 13; i++)
        pinMode(i, OUTPUT);
    pinMode(A2, OUTPUT);
    pinMode(A3, OUTPUT);
    pinMode(A4, OUTPUT);
    pinMode(A5, OUTPUT);
}
void drawMatrix(byte data[]){
    // Turn on each row in series
    for (byte i = 0; i < 8; i++) {
        digitalWrite(rows[i], HIGH); //initiate whole row
        // Turn on each point in row
        for (byte a = 0; a < 8; a++){
            // if You set (~data[i] >> a) then You will have positive
            digitalWrite(cols[a], (data[i] >> a) & 0x01); // initiate whole column
            digitalWrite(cols[a], HIGH); // reset whole column
        }
        digitalWrite(rows[i], LOW); // reset whole row
        // otherwise last row will intersect with next row
    }
}

void loop() {
    int count;
    for(int i = 0; i < 30; i++){
        count = 0;
        while(count < 200){
            drawMatrix(chuSon[i]);
            count++;
        }
    }
}

```

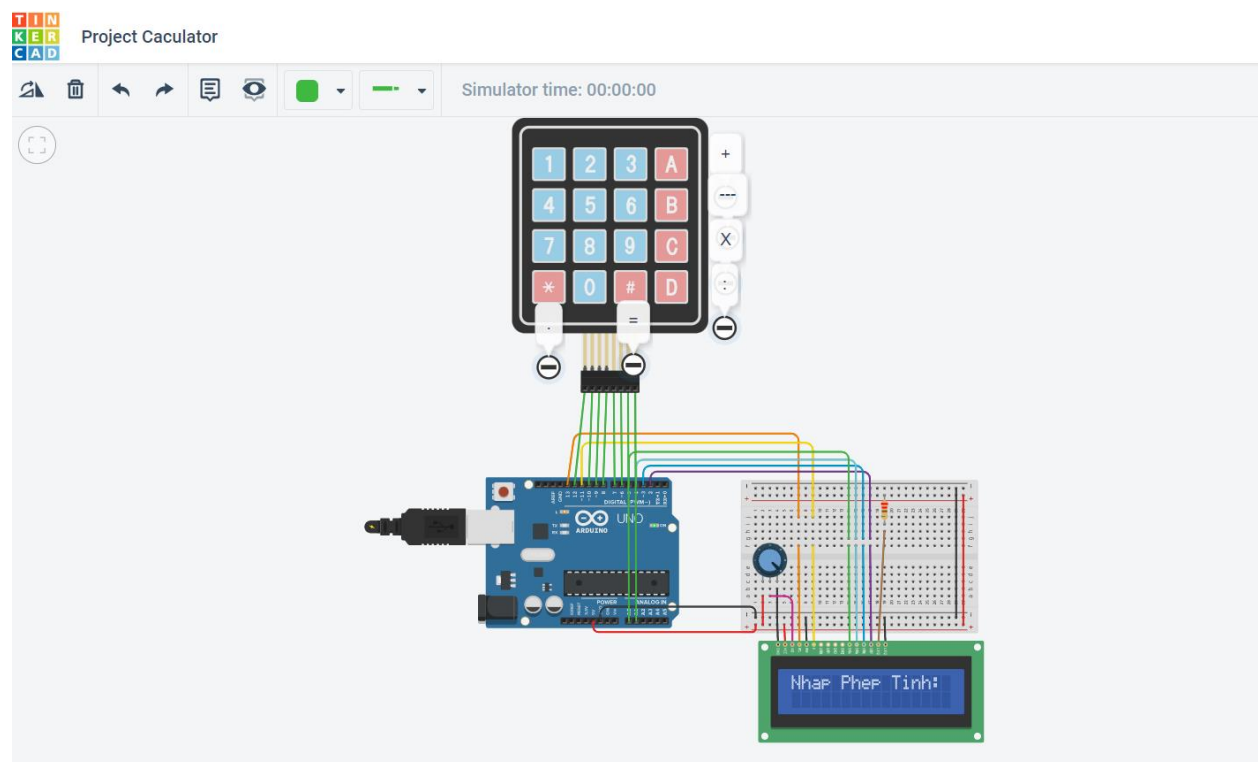
#Project 2.1: Caculator arduino tính tổ hợp các phép tính.

Mô tả:

Đây là một máy tính cầm tay với sự kết hợp giữa keypad và LCD. Bằng cách Nhập một phép tính hoặc tổ hợp các phép tính kết quả sẽ được hiển thị trên LCD. Nhấn phím bất kỳ để thực hiện phép tính khác.

Phần cứng:

- Arduino UNO
- Breadboard
- Resistor 220 Ω
- Potentiometer
- Keypad 4x4
- LCD 16x2



Link:

- Tinkercard: <https://www.tinkercad.com/things/213WMdGe2V9-project-caculator/editel?sharecode=L2Ez4ktYM7jm7A4q6uwxuT6qb0AuuvUZoZkSyveEaeM>

- Youtube Video: <https://youtu.be/6W9oKzKiUjs>

Code:



```

// include the library code:
#include <LiquidCrystal.h>
#include <Keypad.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', '+'},
  {'4', '5', '6', '-'},
  {'7', '8', '9', 'x'},
  {'.', '0', '=', ':'}
};

byte rowPins[ROWS] = {12, 10, 9, 8};
byte colPins[COLS] = {7, 6, A0, A1};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);
}

float result(String str) { // hàm trả về kết quả cuối cùng
  String cal = "";
  float number[16];
  int countNum = 0;
  bool check = true;
  float x = 0;
  float chia = 10;
  for (int i = 0; i < str.length(); i++) { // tách từng số và phép tính ra 2 mảng
    if (str[i] == '=') {
      number[countNum] = x;
    }
    if (str[i] == '+' || str[i] == '-' || str[i] == 'x' || str[i] == ':') { // tách phép tính
      cal += str[i];
      number[countNum] = x;
      countNum++;
      x = 0;
      check = true; // reset lại các giá trị
      chia = 10;
    }
    if (str[i] >= '0' && str[i] <= '9' && check == true) {
      int tmp = str[i] - 48;
      x = 10 * x + tmp*1.0; // cộng dồn để tạo số trước hàng thập phân
    }
  }
}

```



```

}

if (str[i] == '.') {
    check = false;
    i++;
}
if (check == false) {
    int tmp = str[i] - 48;
    x = x + (tmp / 1.0) / chia;           //tạo số sau dấu .
    chia *= 10;
}

}

String cal2 = "";
float number2[16];
int countNum2 = 0;
float tmp = 0;
int k = 0;
if (cal[0] == '+' || cal[0] == '-') {    // check phép tính đầu là + hoặc trừ
    cal2 += cal[0];
    number2[countNum2] = number[0];
    countNum2++;
    k++;
    if (cal.length() == 1)
        number2[countNum2] = number[1];
}
for (k; k < cal.length(); k++) {        // tính toán ưu tiên phép nhân, chia và tạo 2 mảng số và +,-

    if (cal[k] == '+' || cal[k] == '-') {
        if (cal[k - 1] == 'x' || cal[k - 1] == ':')
            cal2 += cal[k];
        else {
            cal2 += cal[k];
            number2[countNum2] = number[k];
            countNum2++;
        }

    } else {
        float a;
        if (tmp != 0)
            a = tmp;
        else
            a = number[k];
        float b = number[k + 1];
        float rs;
        if (cal[k] == 'x') {
            rs = a * b;

```

```

    } else {
        rs = a / b;
    }
    if (cal[k + 1] == 'x' || cal[k + 1] == ':') {
        tmp = rs;
    } else {
        number2[countNum2] = rs;
        countNum2++;
        tmp = 0;
    }
}
}
if(cal[cal.length()-1] == '+' || cal[cal.length()-1] == '-'){
    number2[cal2.length()] = number[cal.length()];
}

float result = number2[0];
if (cal2.length() != 0)
    for (int i = 0; i < cal2.length(); i++) {    // cộng dồn lần cuối
        if (cal2[i] == '+') {
            result += number2[i + 1];
        } else {
            result -= number2[i + 1];
        }
    }
return result;
}

void loop() {
    lcd.clear();
    String str = "";
    int count = 0;
    int x = 0;
    lcd.setCursor(0, 0);
    lcd.print("Nhap Phep Tinh:");
    while (true) {
        char customKey = customKeypad.getKey();
        if (customKey) {
            if (customKey == '=') {    // dấu = để bắt đầu tính
                lcd.setCursor(x, 1);
                lcd.print(customKey);
                str += customKey;
                delay(2000);
                break;
            }
            lcd.setCursor(x, 1);
            lcd.print(customKey);
            str += customKey;    // cộng dồn chuỗi

```

```
    x++;  
  }  
}  
  
lcd.clear();  
  
while (true) {  
  char customKey = customKeypad.getKey();  
  if (customKey) {  
    break;  
  } else {  
    lcd.setCursor(0, 0);  
    lcd.print("Ket Qua La: ");  
    lcd.setCursor(0, 1);  
    lcd.print(result(str));    // truyền chuỗi vào và in ra kết quả  
  }  
}  
  
}
```

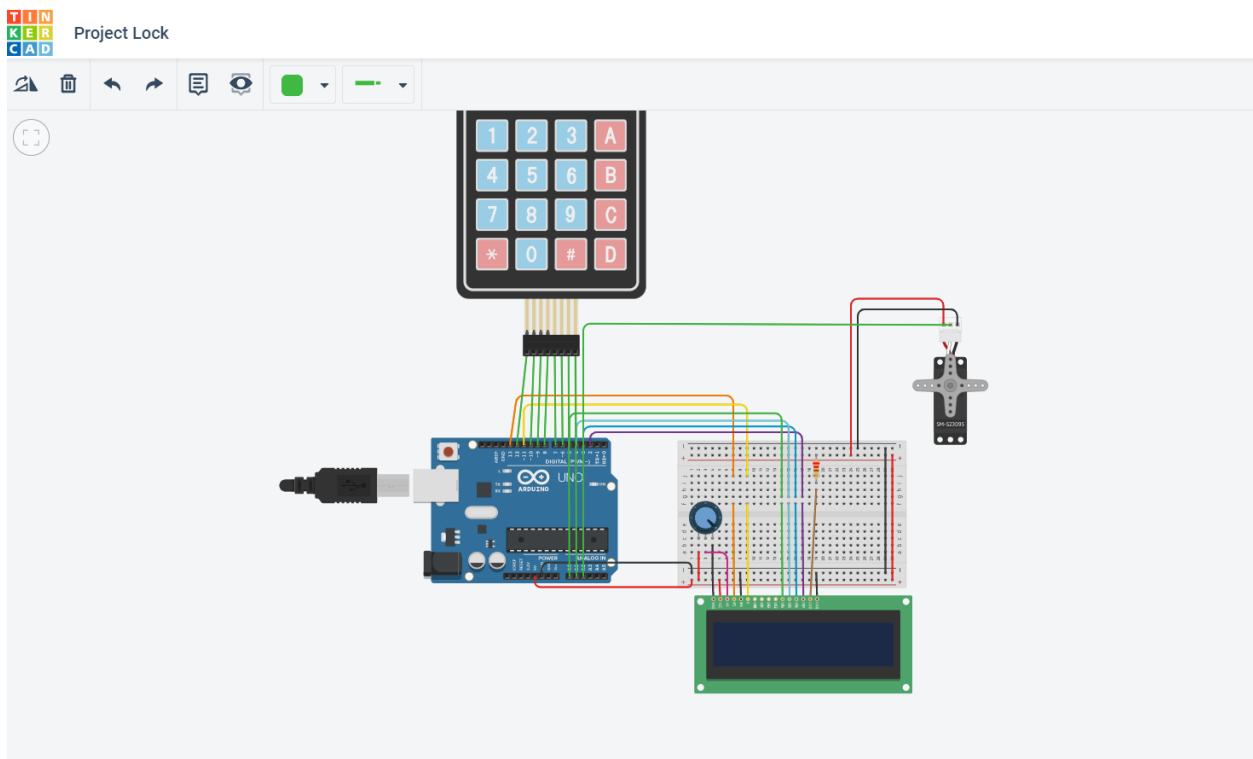
#Project 2.2: SMART LOCK Kết hợp giữa Keypad, LCD, Servo.

Mô tả:

Đây là smart lock với sự kết hợp giữa keypad và LCD và Servo. Bằng cách thiết lập mật khẩu sử dụng keypad với chức năng tạo mật khẩu, nhập mật khẩu và đổi mật khẩu. Khi nhập mật khẩu chính xác thì Servo sẽ quay tương ứng với việc mở khóa.

Phần cứng:

- Arduino UNO
- Breadboard
- Resistor 220Ω
- Potentiometer
- Keypad 4x4
- LCD 16x2
- Servo



Link:

- Tinkercard: https://www.tinkercad.com/things/6DA7iBKlu4-copy-of-project-caculator/editel?sharecode=g0rKMoeXBxhK3fKwla3Q5CMX0hkEyOIT7_BP7x1f5HA



- Youtube Video: <https://youtu.be/nG469IBdyus>

Code:

```
// include the library code:
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Servo.h>
Servo myservo;
int pos = 0;

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {12, 10, 9, 8};
byte colPins[COLS] = {7, 6, A0, A1};

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

String pass = "";
void setup() {
  myservo.attach(A2);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  Serial.begin(9600);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Mat Khau Moi:");
  int x = 0;
  char customKey;
  while (true) {
    customKey = customKeypad.getKey();
    if (customKey) {
      if (customKey == '#') {
        break;
      }
    }
  }
}
```

```

    }
    if (customKey == 'C') {
        x--;
        pass.remove(x);
        lcd.setCursor(x, 1);
        lcd.print(" ");
    } else {
        lcd.setCursor(x, 1);
        lcd.print("*");
        pass += customKey;
        x++;
    }
}
}
}
lcd.clear();
menu();
}

void done() {    //Xoay servo
    for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
        // in steps of 1 degree
        myservo.write(pos);           // tell servo to go to position in variable 'pos'
        delay(15);                     // waits 15ms for the servo to reach the position
    }
    for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
        myservo.write(pos);           // tell servo to go to position in variable 'pos'
        delay(15);                     // waits 15ms for the servo to reach the position
    }
}

void nhapMk() {    //Nhập mật khẩu để check
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Mat Khau:");
    int x = 0;
    String pass2 = "";
    char customKey;
    while (true) {
        customKey = customKeypad.getKey();
        if (customKey) {
            if (customKey == '*') {
                return;
            }
            if (customKey == '#') {
                break;
            }
            if (customKey == 'C') {
                x--;
            }
        }
    }
}

```

```

        pass2.remove(x);
        lcd.setCursor(x, 1);
        lcd.print(" ");
    } else {
        lcd.setCursor(x, 1);
        lcd.print("*");
        pass2 += customKey;
        x++;
    }
}
}
}
if (pass2.equalsIgnoreCase(pass)) {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Mo Khoa.");
    done();
    lcd.clear();
    menu();
} else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Sai Mat Khau.");
    delay(3000);
    lcd.clear();
    menu();
}
}
}
void newMk() {          // Tạo mật khẩu mới
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Mat Khau Cu:");
    int x = 0;
    String pass2 = "";
    char customKey;
    while (true) {
        customKey = customKeypad.getKey();
        if (customKey) {
            if (customKey == '*') {
                lcd.clear();
                menu();
                return;
            }

            if (customKey == '#') {
                if (pass2.equalsIgnoreCase(pass)) {          //Check mật khẩu cũ
                    Serial.println("OK");
                    break;
                } else {

```

```

        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Sai, Nhap Lai:");
        Serial.println("Sai");
        pass2 = "";
        x = 0;
    }
} else if (customKey == 'C') {
    x--;
    pass2.remove(x);
    lcd.setCursor(x, 1);
    lcd.print(" ");
} else {
    lcd.setCursor(x, 1);
    lcd.print("*");
    pass2 += customKey;
    x++;
}
}
}
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Mat Khau Moi:");
x = 0;
String passNew = "";
customKey;
while (true) {
    customKey = customKeypad.getKey();
    if (customKey) {
        if (customKey == '*') {
            lcd.clear();
            menu();
            return;
        }
        if (customKey == '#') {
            lcd.clear();
            lcd.setCursor(0, 0);
            String passNew2 = "";
            lcd.print("Nhap Lai:");
            x = 0;
            while (true) {
                customKey = customKeypad.getKey();
                if (customKey) {
                    if (customKey == '*') {
                        lcd.clear();
                        menu();
                        return;
                    }
                }
            }
        }
    }
}

```



```

    if (customKey == '#') {
        if (passNew2.equalsIgnoreCase(passNew)) {           // Check lại mật khẩu mới
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Thanh Cong.");
            Serial.println("OK");
            pass = passNew;
            delay(3000);
            lcd.clear();
            menu();
            return;
        } else {
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Sai, Nhap Lai:");
            x = 0;
            Serial.println("Sai");
            passNew2 = "";
        }
    } else if (customKey == 'C') {
        x--;
        passNew2.remove(x);
        lcd.setCursor(x, 1);
        lcd.print(" ");
    } else {
        lcd.setCursor(x, 1);
        lcd.print("*");
        passNew2 += customKey;
        x++;
    }
}
}
}

} else if (customKey == 'C') {
    x--;
    passNew.remove(x);
    lcd.setCursor(x, 1);
    lcd.print(" ");
} else {
    lcd.setCursor(x, 1);
    lcd.print("*");
    passNew += customKey;
    x++;
}
}
}
}

void menu() {

```

```
lcd.setCursor(0, 0);  
lcd.print("1.Mo Khoa");  
lcd.setCursor(0, 1);  
lcd.print("2.Mat Khau Moi");  
}  
void loop() {  
  char customKey;  
  customKey = customKeypad.getKey();  
  if (customKey) {  
    if (customKey == '1') {  
      nhapMk();  
    } else if (customKey == '2') {  
      newMk();  
    }  
  }  
}
```

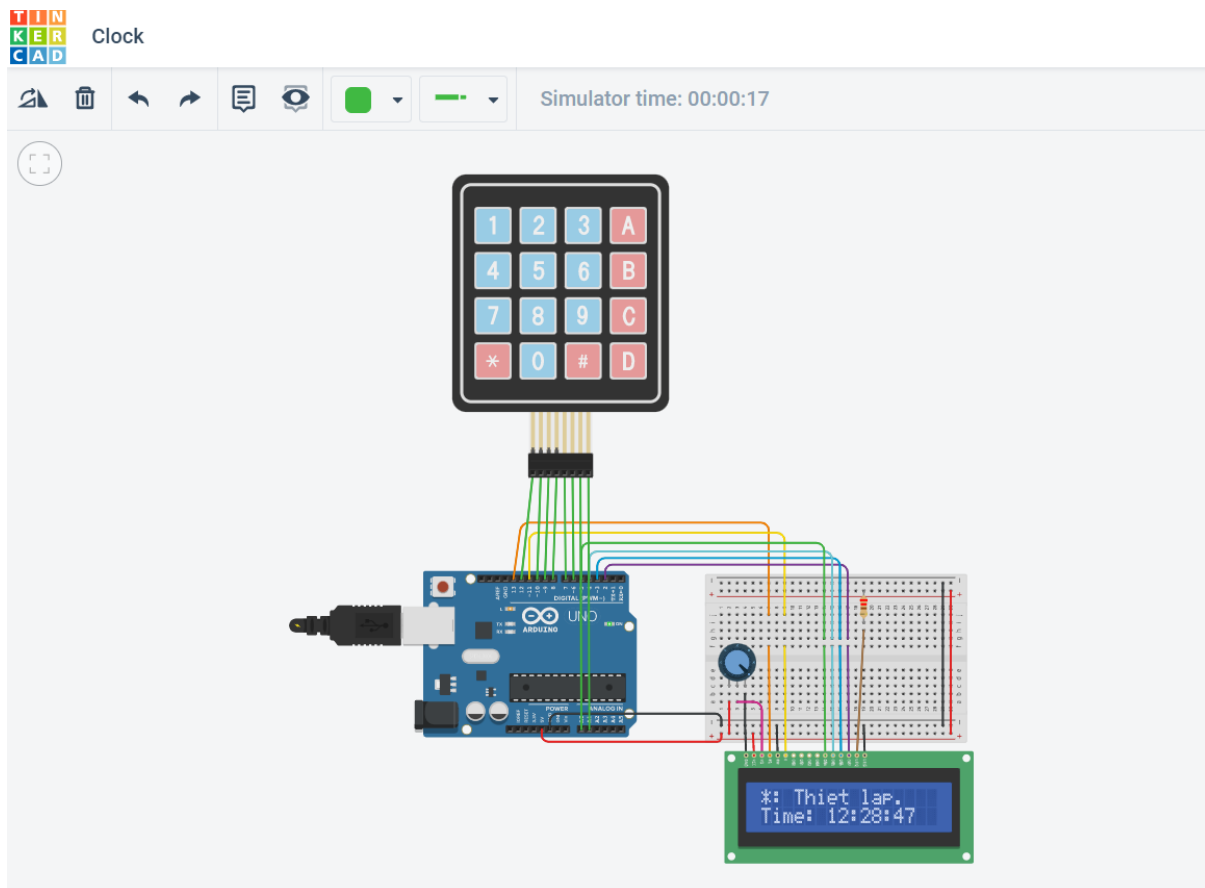
#Project 2.3: Hiển thị đồng hồ trên LCD (Có thể thiết lập thời gian).

Mô tả:

Đây là đồng hồ với sự kết hợp giữa keypad và LCD. Bằng cách thiết lập thời gian với keypad với phím '*'.

Phần cứng:

- Arduino UNO
- Breadboard
- Resistor 220Ω
- Potentiometer
- Keypad 4x4
- LCD 16x2



Link:

- Tinkercard: <https://www.tinkercad.com/things/9LT7c6zJkna-copy-of-project-lock/editel?sharecode=FLhg6skdGgqI3eZ7hesYd0m9pAK25c6sXksnpjndH9s>
- Youtube Video: <https://youtu.be/C3ZRyLwa4Yc>



Code:

```
// include the library code:
#include <LiquidCrystal.h>
#include <Keypad.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
const byte ROWS = 4;
const byte COLS = 4;
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {12, 10, 9, 8};
byte colPins[COLS] = {7, 6, A0, A1};

Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
int gio = 0, phut = 0, giay = 0;

void setup() {
  lcd.begin(16, 2);
  lcd.clear();
  Serial.begin(9600);
}

void hienthi() {
  for (gio; gio <= 23; gio++) {
    for (phut; phut <= 59; phut++) {
      for (giay; giay <= 59; giay++) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("*: Thiet lap.");
        lcd.setCursor(0, 1);
        lcd.print("Time: ");
        if (gio <= 9) {
          lcd.print(0);
          lcd.print(gio);
          lcd.print(":");
        } else {
          lcd.print(gio);
```

```

    lcd.print(":");
}
if (phut <= 9) {
    lcd.print(0);
    lcd.print(phut);
    lcd.print(":");
} else {
    lcd.print(phut);
    lcd.print(":");
}
}
if (giay <= 9) {
    lcd.print(0);
    lcd.print(giay);
} else {
    lcd.print(giay);
}
for (int i = 0; i < 937; i++) {          // Em để 937 nó gần giống 1 giây
    char customKey = customKeypad.getKey();
    if (customKey) {
        if (customKey == '*') { // dấu = để bắt đầu tính
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Gio:");
            setGio();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Phut:");
            setPhut();
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Giay:");
            setGiay();
        }
    }
    delay(1);
}
if (giay == 59) {
    giay = 0;
    break;
}
}
if (phut == 59) {
    phut = 0;
    break;
}
}
if (gio == 23) {
    gio = -1;
}

```

```

    }
}

}

int parseNum(String str) {           // Ép chuỗi về số
    if (str.length() > 2) {           // Check chữ số
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Khong Hop Le!");
        delay(2000);
    }
    int rs = 0;
    for (int i = 0; i < str.length(); i++) {    // Ép kiểu
        int tmp = str[i] - 48;
        rs = 10 * rs + tmp;
    }
    return rs;
}

void setGio() {                       // Chỉnh giờ
    int rs = gio;
    lcd.setCursor(0, 1);
    int x = 0;
    String str = "";
    char customKey;
    while (true) {
        customKey = customKeypad.getKey();
        if (customKey) {
            if (customKey == '#') {
                break;
            }
            if (customKey == 'C') {           //Xóa 1 kí tự
                x--;
                str.remove(x);
                lcd.setCursor(x, 1);
                lcd.print(" ");
            }
            else {
                lcd.setCursor(x, 1);
                lcd.print(customKey);
                str += customKey;
                x++;
            }
        }
    }
    rs = parseNum(str);

    if (rs > 23 || rs < 0) {
        lcd.clear();
    }
}

```

```

    lcd.setCursor(0, 0);
    lcd.print("Nhap Lai Gio:");
    setGio();
} else {
    gio = rs;
}

}

void setPhut() {           // Chỉnh phút
    int rs = phut;
    lcd.setCursor(0, 1);
    int x = 0;
    String str = "";
    char customKey;
    while (true) {
        customKey = customKeypad.getKey();
        if (customKey) {
            if (customKey == '#') {
                break;
            }
            if (customKey == 'C') {
                x--;
                str.remove(x);
                lcd.setCursor(x, 1);
                lcd.print(" ");
            }
            else {
                lcd.setCursor(x, 1);
                lcd.print(customKey);
                str += customKey;
                x++;
            }
        }
    }
    rs = parseNum(str);

    if (rs > 60 || rs < 0) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Nhap Lai Phut:");
        setPhut();
    } else {
        phut = rs;
    }

}

void setGiay() {           // Chỉnh giây
    int rs = giay;

```

```
lcd.setCursor(0, 1);
int x = 0;
String str = "";
char customKey;
while (true) {
  customKey = customKeypad.getKey();
  if (customKey) {
    if (customKey == '#') {
      break;
    }
    if (customKey == 'C') {
      x--;
      str.remove(x);
      lcd.setCursor(x, 1);
      lcd.print(" ");
    } else {
      lcd.setCursor(x, 1);
      lcd.print(customKey);
      str += customKey;
      x++;
    }
  }
}
rs = parseNum(str);

if (rs > 60 || rs < 0) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Nhap Lai Giay:");
  setGiay();
} else {
  giay = rs;
}

}

void loop() {

  hienthi();
}
```


#Project 3: Cảnh báo va chạm.

Mô tả:

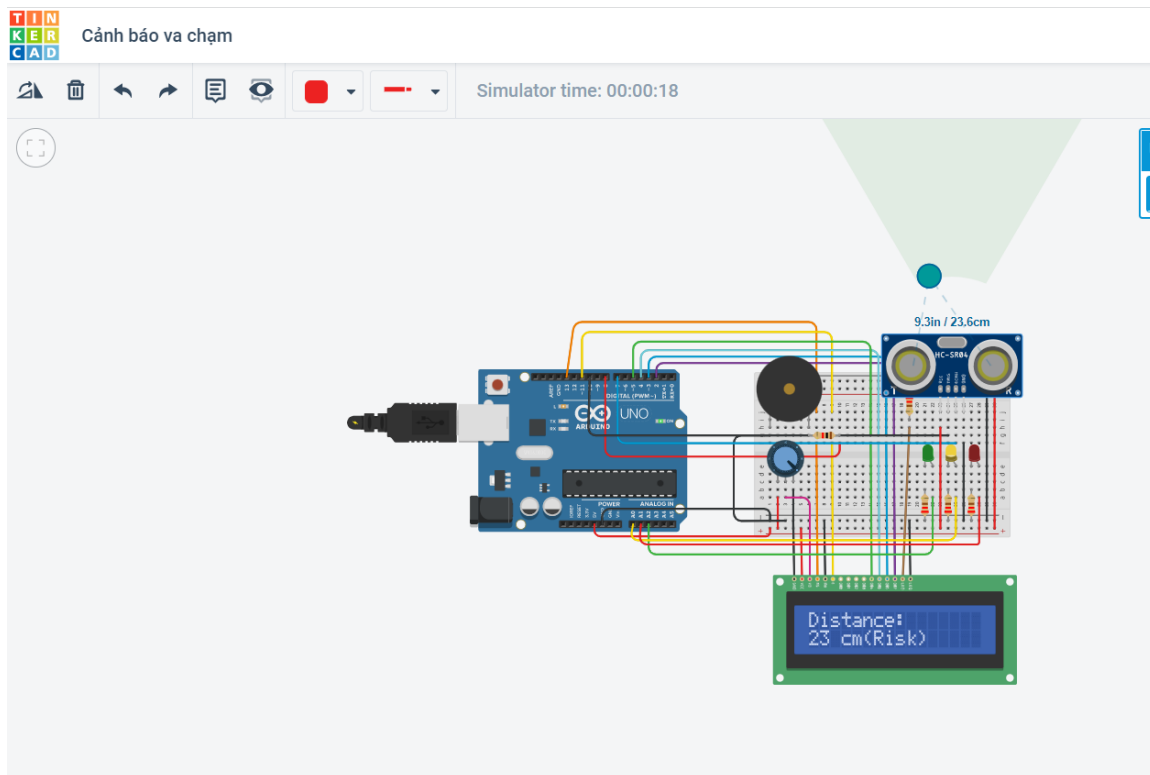
Dùng cảm biến siêu âm xác định khoảng cách đến vật cản.

Dùng còi để cảnh báo, dùng đèn led để hiển thị thêm trạng thái.

Hiển thị thêm thông tin trên LCD.

Phần cứng:

- Arduino UNO
- Breadboard
- 3 Resistor 220 Ω
- Resistor 1k Ω
- Potentiometer
- Ultrasonic distance sensor
- 3 Led (xanh, vàng, đỏ)
- Buzzer
- LCD 16x2



Link:

- Tinkercard: https://www.tinkercad.com/things/35m0ql596x7-copy-of-project-lock/editel?sharecode=8yTR1sziLGj7ScA0V76xSFovby6AoYQq_r_HLIOv6Q

- Youtube Video: <https://youtu.be/HqNgzDFmlgM>

Code:



```
// include the library code:
#include <LiquidCrystal.h>

const unsigned int TRIG_PIN = 10;
const unsigned int ECHO_PIN = 7;
const unsigned int BAUD_RATE = 9600;

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 13, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int dis() {          // Trả về khoảng cách đo được
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  const unsigned long duration = pulseIn(ECHO_PIN, HIGH);
  int distance = duration / 29 / 2;
  return distance;
}

void warning(int dis) {          // In ra Lcd khoảng cách hiện tại
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Distance:");
  lcd.setCursor(0, 1);
  lcd.print(dis);
  lcd.print(" cm(Warning)");
}

void music() {          // Nhạc em copy cho vào

  if (dis() > 15) {          // Nếu hơn 15cm thì thoát nhạc
    return;
  }

  warning(dis());          // In ra khoảng cách
  int melody[] = {262, 294, 330, 349, 392, 440, 494, 523, 587, 659, 698, 784, 880, 988};
```

```
tone(8, melody[2], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[7], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(500);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[2], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
```

```
tone(8, melody[7], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(500);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[2], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[7], 500); delay(125);
tone(8, melody[8], 500); delay(250);
tone(8, melody[8], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[10], 500); delay(250);
tone(8, melody[10], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[9], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[5], 500); delay(375);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
```

```
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[5], 500); delay(500);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[7], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[7], 500); delay(125);
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(500);

if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(125);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[7], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
```

```
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(500);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[2], 500); delay(125);
tone(8, melody[3], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[7], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(500);
if (dis() > 15) {
    return;
}
warning(dis());

tone(8, melody[2], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
```

```
    return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[8], 500); delay(250);
tone(8, melody[8], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[10], 500); delay(250);
tone(8, melody[10], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[9], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[5], 500); delay(375);

if (dis() > 15) {
    return;
}
warning(dis());

tone(8, melody[2], 500); delay(125);
tone(8, melody[4], 500); delay(125);
tone(8, melody[5], 500); delay(250);
tone(8, melody[5], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[8], 500); delay(250);
tone(8, melody[8], 500); delay(250);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[10], 500); delay(250);
```

```
tone(8, melody[10], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[9], 500); delay(125);
tone(8, melody[8], 500); delay(125);
tone(8, melody[9], 500); delay(125);
tone(8, melody[5], 500); delay(375);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[7], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[8], 500); delay(250);
tone(8, melody[9], 500); delay(125);
tone(8, melody[5], 500); delay(375);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[5], 500); delay(125);
tone(8, melody[7], 500); delay(125);
tone(8, melody[6], 500); delay(250);
tone(8, melody[6], 500); delay(250);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[7], 500); delay(125);
tone(8, melody[5], 500); delay(125);
tone(8, melody[6], 500); delay(750);
if (dis() > 15) {
  return;
}
warning(dis());
tone(8, melody[9], 500); delay(750);
tone(8, melody[10], 500); delay(750);
if (dis() > 15) {
  return;
}
}
```



```

warning(dis());
tone(8, melody[9], 500); delay(125);
tone(8, melody[9], 500); delay(250);
tone(8, melody[11], 500); delay(250);
tone(8, melody[9], 500); delay(125);
tone(8, melody[8], 500); delay(500);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[8], 500); delay(750);
tone(8, melody[7], 500); delay(750);
if (dis() > 15) {
    return;
}
warning(dis());
tone(8, melody[6], 500); delay(125);
tone(8, melody[7], 500); delay(250);
tone(8, melody[6], 500); delay(250);
tone(8, melody[5], 500); delay(500);
}

void setup() {
    // set up the LCD's number of columns and rows:
    pinMode(TRIG_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);
    Serial.begin(BAUD_RATE);
    pinMode(A0, OUTPUT);
    pinMode(A1, OUTPUT);
    pinMode(A2, OUTPUT);
    lcd.begin(16, 2);
    lcd.clear();
}

void loop() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);

    const unsigned long duration = pulseIn(ECHO_PIN, HIGH);
    int distance = duration / 29 / 2;
    lcd.clear();
    if (duration == 0) {

```

```
    lcd.setCursor(0, 0);
    lcd.print("Warning");
}
else
{
    if (distance > 30) {           // Em chỉ để khoảng 30 cho dễ demo
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Distance:");
        digitalWrite(A2, HIGH);
        digitalWrite(A1, LOW);
        digitalWrite(A0, LOW);
        lcd.setCursor(0, 1);
        lcd.print(distance);
        lcd.print(" cm(Safe)");
    } else if (distance <= 30 && distance >= 15 ) {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Distance:");
        digitalWrite(A2, LOW);
        digitalWrite(A1, LOW);
        digitalWrite(A0, HIGH);
        lcd.setCursor(0, 1);
        lcd.print(distance);
        lcd.print(" cm(Risk)");
    } else {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("Distance:");
        digitalWrite(A2, LOW);
        digitalWrite(A1, HIGH);
        digitalWrite(A0, LOW);
        lcd.setCursor(0, 1);
        lcd.print(distance);
        lcd.print(" cm(Warning)");
        music();
    }
}

delay(500);
}
```