#include <LiquidCrystal.h>

// select the pins used on the LCD panel

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

char KeyValue[]={'1','2','3','A','4','5','6','B','7','8','9','C','\*','0','#','D'};

byte Row=0, Col=0;

void setup(){

pinMode(10, INPUT); //R1: S1,S2,S3,S4 (1,2,3,A)

pinMode(11, INPUT\_PULLUP); //R2: S5,S6,S7,S8 (4,5,6,B)

pinMode(12, INPUT\_PULLUP); //R3: S9, S10, S11,S12 (7,8,9,C)

pinMode(13, INPUT\_PULLUP); //R4: (\*,0,#,D)

pinMode(A0, OUTPUT); //A1, C1: S1,S5,S9 (1,4,7,\*)

pinMode(A1, OUTPUT); //A2, C2: S2,S6,S10 (2,5,8,0)

pinMode(A2, OUTPUT); //A3, C3: S3,S7,S11 (3,6,9,#)

pinMode(A3, OUTPUT); //A4, C4, S4,S8,S12 (\*,0, #,D)

digitalWrite(A0,HIGH);

digitalWrite(A1,HIGH);

digitalWrite(A2,HIGH);

digitalWrite(A3,HIGH);

lcd.begin(16, 2); // start the library

lcd.setCursor(0,0);

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

lcd.print("Key Martrix Test");

delay(1000);

lcd.clear();

delay(400);

}

void loop() {

static int keypressedcount=0;

byte keyindex=0;

if(keyscan()==true){

keyindex=(Row-1)\*4+Col;

delay(5);

if((keyscan()==true) ){

keyindex==(Row-1)\*4+Col;

lcd.clear();

lcd.setCursor(0,1);

//lcd.print(KeyValue[keyindex-1]);

lcd.print(keyindex);

digitalWrite(A0,LOW);

digitalWrite(A1,LOW);

digitalWrite(A2,LOW);

digitalWrite (A3,LOW);

delayMicroseconds(100);

}

}

}

bool keyscan( )

{

Row=0;Col=0;

bool keypressed = false;

//scan col1

digitalWrite(A0, LOW);

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW){

digitalWrite(A0, HIGH);

Col=1;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW){

digitalWrite(A0, HIGH);

Col=1;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW){

digitalWrite(A0, HIGH);

Col=1;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW){

digitalWrite(A0, HIGH);

Col=1;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 2

digitalWrite(A0, HIGH);

digitalWrite(A1, LOW);

digitalWrite(A2, HIGH);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW){

digitalWrite(A1, HIGH);

Col=2;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW){

digitalWrite(A1, HIGH);

Col=2;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW)

{

digitalWrite(A1, HIGH);

Col=2;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW){

digitalWrite(A1, HIGH);

Col=2;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 3

digitalWrite(A0, HIGH);

digitalWrite(A1, HIGH);

digitalWrite(A2, LOW);

digitalWrite(A3, HIGH);

delayMicroseconds(100);

//Read keys in row.1

if(digitalRead(10)==LOW){

digitalWrite(A2, HIGH);

Col=3;Row=1;

keypressed = true;

return(keypressed);

}

//Read keys in row.2

if(digitalRead(11)==LOW){

digitalWrite(A2, HIGH);

Col=3;Row=2;

keypressed = true;

return(keypressed);

}

//Read keys in row.3

if(digitalRead(12)==LOW){

digitalWrite(A2, HIGH);

Col= 3;Row=3;

keypressed = true;

return(keypressed);

}

//Read keys in row.4

if(digitalRead(13)==LOW){

digitalWrite(A2, HIGH);

Col=3;Row=4;

keypressed = true;

return(keypressed);

}

//scan col 4

digitalWrite (A0,HIGH);

digitalWrite(A1, HIGH);

digitalWrite(A2, HIGH);

digitalWrite(A3, LOW);

if(digitalRead(13)==LOW){

delayMicroseconds(100);

//Read keys in row.1

}

if(digitalRead(10)==LOW){

digitalWrite(A3, HIGH);

Col=4;Row=1;

keypressed = true;

return( keypressed);

//Read keys in row.2

}

if(digitalRead(11)==LOW)

{

digitalWrite(A3, HIGH);

Col=4;Row=2;

keypressed = true;

return(keypressed);

//Read keys in row.3

}

if(digitalRead(12)==LOW){

digitalWrite(A3, HIGH);

Col=4;Row=3;

keypressed = true;

return(keypressed);

//Read

}

//Read keys in row.4

if(digitalRead(13)==LOW){

digitalWrite(A3, HIGH);

Col=4;Row=4;

keypressed = true;

return(keypressed);

}

return(false);

}