Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

**4**

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **1** | Write a sketch to interface Arduino with a 3 x 4 Matrix Keypad. The display of the pressed key should be displayed on the LCD. |
| **2** | Write a sketch that works as security keypad lock. Set any password. If the input password matches with the set password, Green LED should glow, otherwise Red LED will glow. |
| 3 | Write a sketch to control the LEDs based on their Labelled Number. (Take input from Keypad and there are 10 LED’s in all labelled from 0 to 9) |

Submitted On:

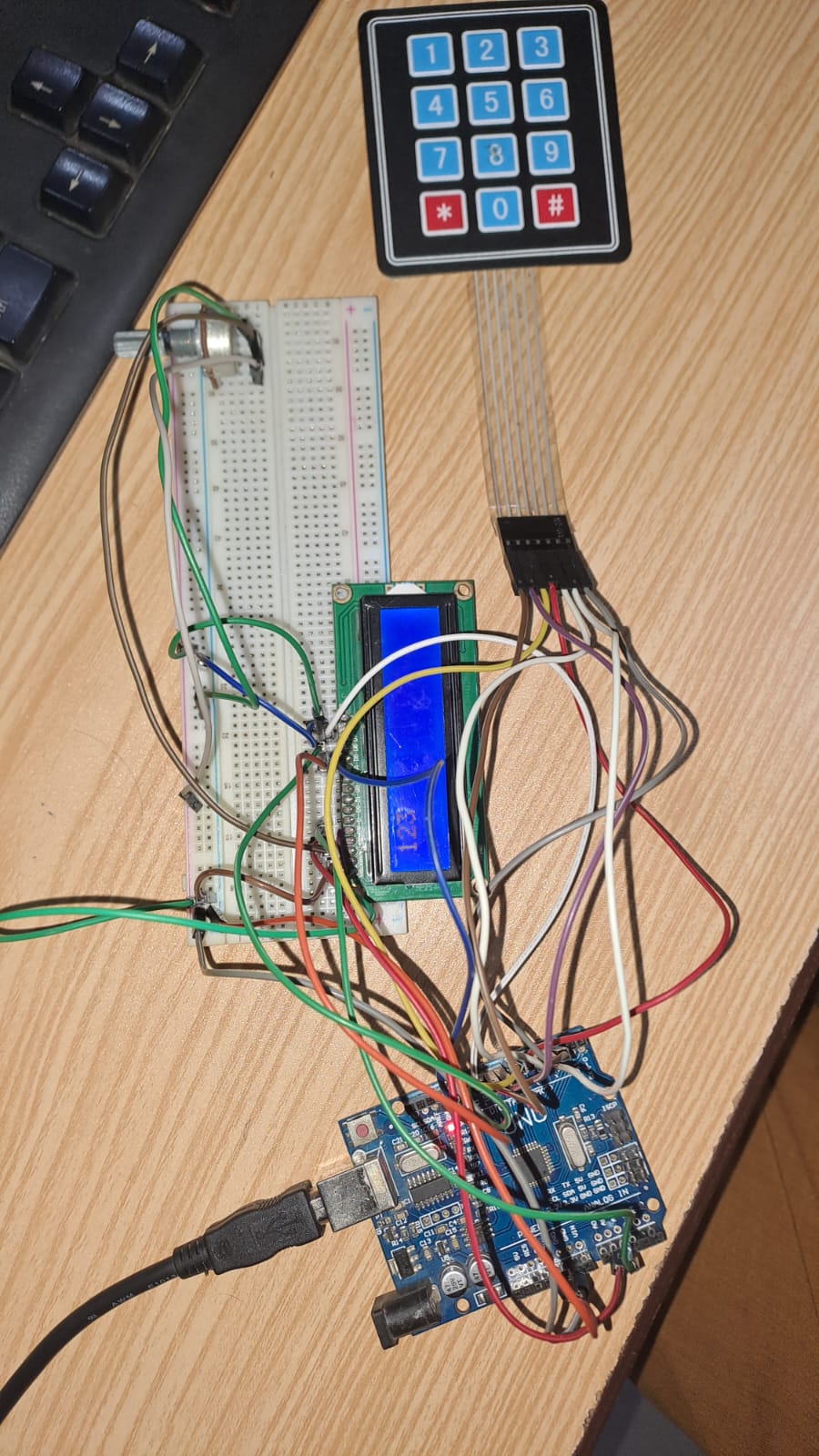
22 December 2023

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: DD/MM/YY)

**Task 1**

Write a sketch to interface Arduino with a 3 x 4 Matrix Keypad. The display of the pressed key should be displayed on the LCD.

#include <LiquidCrystal.h>

#include <Keypad.h>

LiquidCrystal lcd(A1, A2, 13, 12, 11, 10);

const byte ROWS = 4;

const byte COLS = 3;

char keys[ROWS][COLS] =

{

// Four Rows of Keypad.

// Three Columns of Keypad

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

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

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

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

};

byte rowPins[ROWS] = {2,3,4,5};

byte colPins[COLS] = {8,7,6};

Keypad keypad = Keypad(

makeKeymap(keys), rowPins,

colPins, ROWS, COLS );

void setup()

{

Serial.begin(9600);

lcd.begin(16, 2);

}

void loop()

{

char key =

keypad.getKey();

if (key)

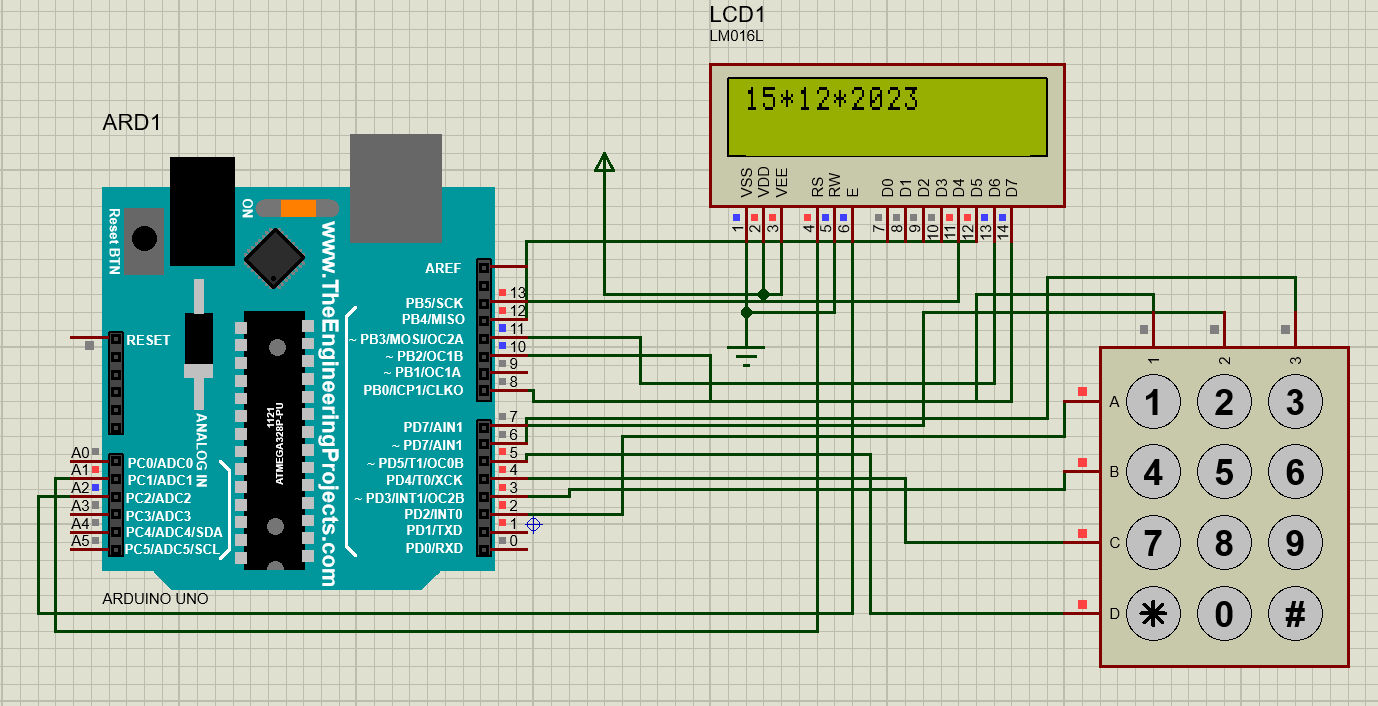
{

Serial.println(key);

lcd.print(key);

}

}



**Task 2**

Write a sketch that works as security keypad lock. Set any password. If the input password matches with the set password, Green LED should glow, otherwise Red LED will glow.

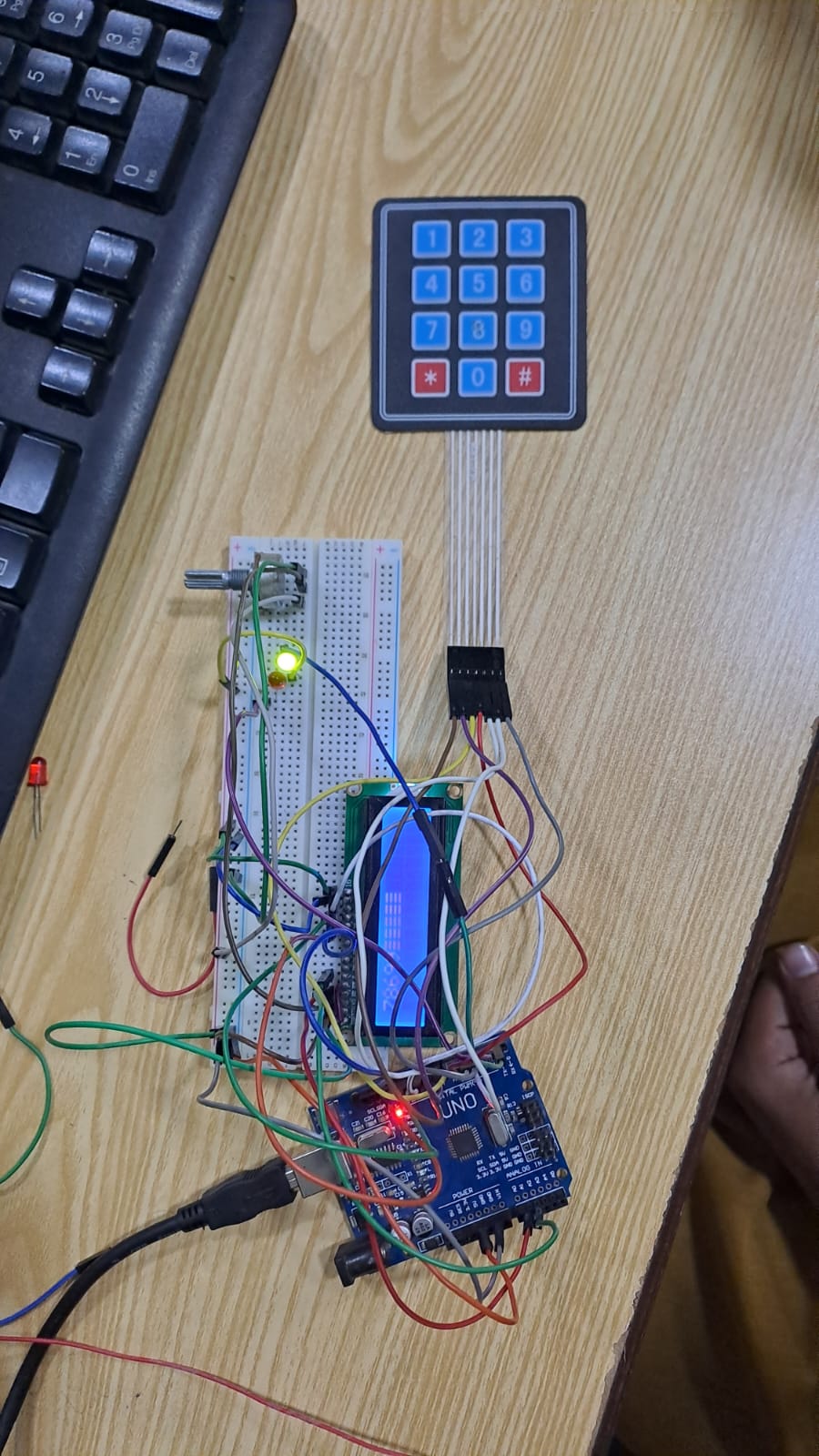
#include <Keypad.h>

char\* secretCode = "1234";

int position = 0;

const byte rows = 4;

const byte cols = 3;

char keys[rows][cols] =

{

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

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

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

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

};

byte rowPins[rows] = {2, 3,4,5};

byte colPins[cols] = {8,7,6};

Keypad keypad = Keypad(makeKeymap(keys),

rowPins, colPins, rows, cols);

int redPin = 12;

int greenPin = 9;

void setup()

{

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

setLocked(true);

}

void loop()

{

char key = keypad.getKey();

if (key == '\*' || key == '#') // Reset Buttons.

{

position = 0;

setLocked(true);

}

if (key == secretCode[position])

{

position ++;

}

if (position == 4)

{

setLocked(false);

}

delay(100);

}

void setLocked(int locked)

{

if (locked)

{

digitalWrite(redPin, HIGH);

digitalWrite(greenPin, LOW);

}

else

{

digitalWrite(redPin, LOW);

digitalWrite(greenPin, HIGH);

}

}

**Task 3**

Write a sketch to control the LEDs based on their Labelled Number. (Take input from Keypad and there are 10 LED’s in all labelled from 0 to 9)

#include <Keypad.h>

const byte rows = 4;

const byte cols = 3;

char keys[rows][cols] =

{

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

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

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

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

};

byte rowPins[rows] = {9,8,7,6};

byte colPins[cols] = {5,4,3};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);

int Pin1 = 13;

int Pin2 = 12;

int Pin3 = 11;

int Pin4 = 10;

int Pin5 = 2;

int Pin6 = A3;

int Pin7 = A4;

int Pin8 = A0;

int Pin9 = A1;

int Pin0 = A2;

void setup()

{

pinMode(Pin1, OUTPUT);

pinMode(Pin2, OUTPUT);

pinMode(Pin3, OUTPUT);

pinMode(Pin4, OUTPUT);

pinMode(Pin5, OUTPUT);

pinMode(Pin6, OUTPUT);

pinMode(Pin7, OUTPUT);

pinMode(Pin8, OUTPUT);

pinMode(Pin9, OUTPUT);

pinMode(Pin0, OUTPUT);

}

void loop()

{

char key = keypad.getKey();

if (key == '\*') // Reset Buttons.

{

  digitalWrite(Pin1,HIGH);

  digitalWrite(Pin2,HIGH);

  digitalWrite(Pin3,HIGH);

  digitalWrite(Pin4,HIGH);

  digitalWrite(Pin5,HIGH);

  digitalWrite(Pin6,HIGH);

  digitalWrite(Pin7,HIGH);

  digitalWrite(Pin8,HIGH);

  digitalWrite(Pin9,HIGH);

  digitalWrite(Pin0,HIGH);

}

else if (key == '1')

{

  digitalWrite(Pin1,HIGH);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '2')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,HIGH);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '3')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,HIGH);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '4')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,HIGH);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '5')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,HIGH);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '6')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,HIGH);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '7')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,HIGH);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '8')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,HIGH);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

else if (key == '9')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,HIGH);

  digitalWrite(Pin0,LOW);

}

else if (key == '0')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,HIGH);

}

else if (key == '#')

{

  digitalWrite(Pin1,LOW);

  digitalWrite(Pin2,LOW);

  digitalWrite(Pin3,LOW);

  digitalWrite(Pin4,LOW);

  digitalWrite(Pin5,LOW);

  digitalWrite(Pin6,LOW);

  digitalWrite(Pin7,LOW);

  digitalWrite(Pin8,LOW);

  digitalWrite(Pin9,LOW);

  digitalWrite(Pin0,LOW);

}

}

