Experiment No. 03 – Interfacing Displays and Keypads

Part 1 – Interfacing a LCD Display

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x Breadboard
4. 1x LCD (16x2 LCD Display)
5. 1x 10kΩ Potentiometer
6. 1x 220Ω Resistor

Schematic Diagram:

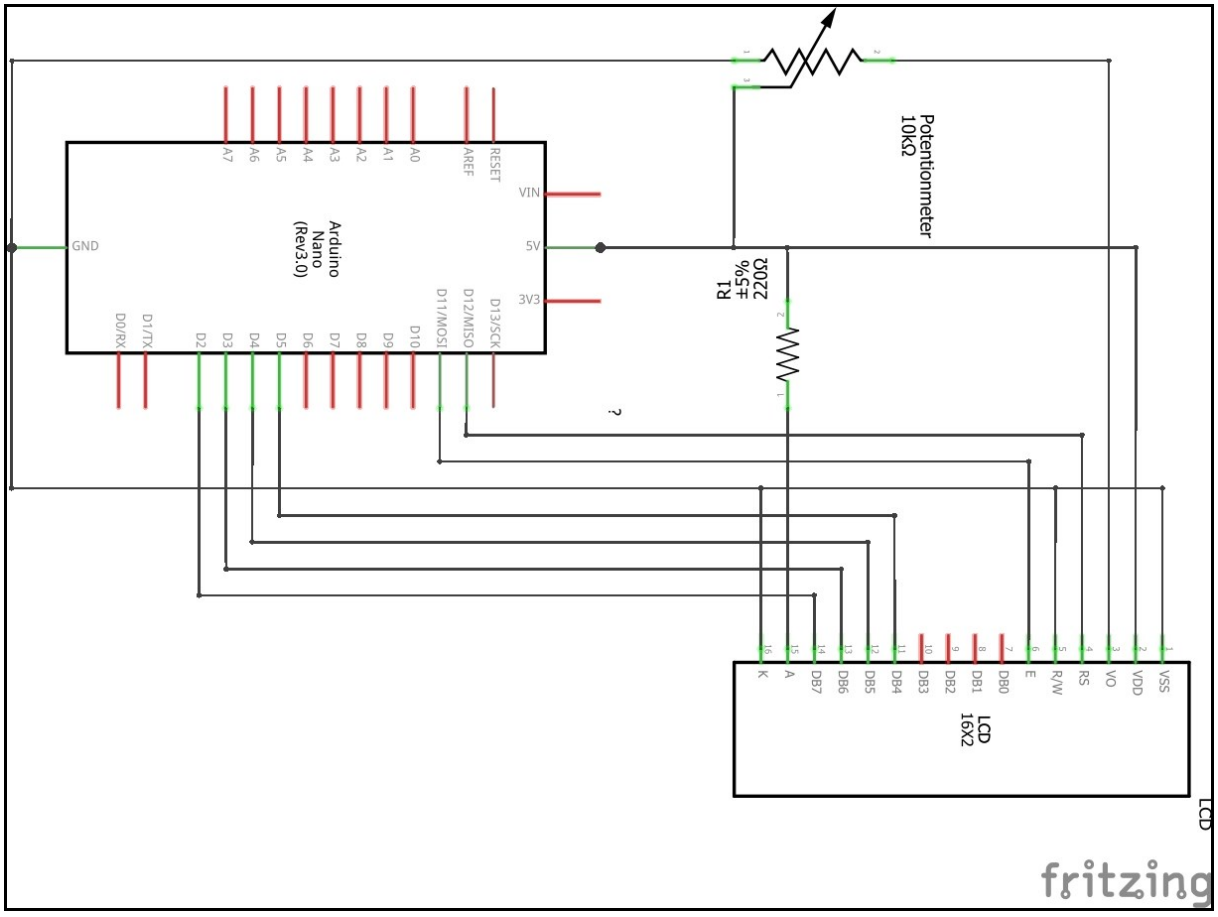

Figure 1 - Exp. 03 - Part 1 – LCD Display interfacing

Experiment Setup: Connect the PINS as per the following -

- LCD RS pin to digital pin 12
- LCD Enable pin to digital pin 11
- LCD D4 pin to digital pin 5
- LCD D5 pin to digital pin 4
- LCD D6 pin to digital pin 3
- LCD D7 pin to digital pin 2
- Upload the code and turn the potentiometer to see the output.

Code:

```
#include <LiquidCrystal.h>
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(fibonacci(millis() / 1000));
}
int fibonacci(int i){
  if(i==0 || i==1){
    return 1;
  }
  else return fibonacci(i-1)+fibonacci(i-2);
}
```

Code Listing 1 - Code for a 16x2 LCD Display interfacing

Expected Output:

- The LCD should show the text “Hello World” after turning the potentiometer to a suitable value.
- On the 2nd row of the display it will print a Fibonacci series after 1 sec intervals.

Appendix:

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

A 16x2 LCD means it can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix. This LCD has two registers, namely, Command and Data.

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD.

Reference: Kushagra, "16 x 2 LCD Datasheet | 16x2 Character LCD Module PINOUT", *Engineersgarage.com*, 2019. [Online]. Available: https://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet. [Accessed: 13- May- 2019].

Part 2 – Interfacing a 4x4 Keypad

Equipment List:

1. 1x Arduino Nano
2. 1x USB 2.0 to USB type A connecting cable
3. 1x KeyPad (4x4 Keypad)

Schematic Diagram:

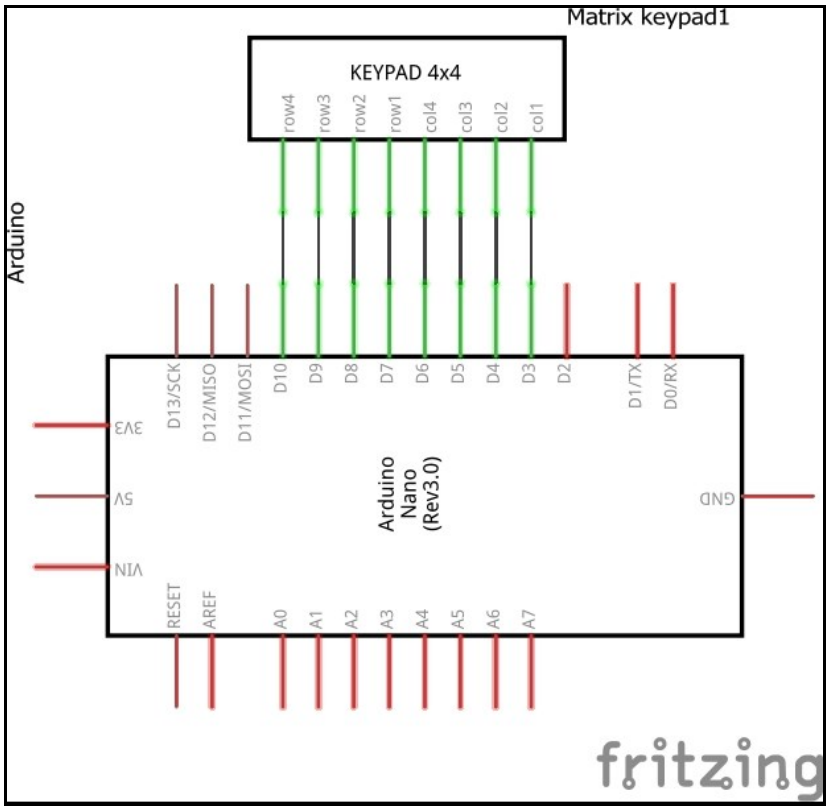

Figure 2 - Exp. 03 - Part 2 – 4x4 Keypad Interfacing

Experiment Setup: Connect the PINS as per the following -

- PIN 1 with digital D10
- PIN 2 with digital D9
- PIN 3 with digital D8
- PIN 4 with digital D7
- PIN 5 with digital D6
- PIN 6 with digital D5
- PIN 7 with digital D4
- PIN 8 with digital D3
- Upload the code.
- Open the serial monitor and press any key to see the output.

Expected Output:

- The Serial Monitor will output the keys pressed in the keypad.

Code:

```
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {10, 9, 8, 7}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {6, 5, 4, 3}; //connect to the column pinouts of the keypad

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

void setup(){
  Serial.begin(9600);
}

void loop(){
  char key = keypad.getKey();

  if (key){
    Serial.println(key);
  }
}
```

Code Listing 2 – Code for Keypad interfacing

Appendix:

A 4X4 KEYPAD will have EIGHT TERMINALS. In them four are ROWS of MATRIX and four are COLUMNS of MATRIX. These 8 PINS are driven out from 16 buttons present in the MODULE. Those 16 alphanumeric digits on the MODULE surface are the 16 buttons arranged in MATRIX formation.

Reference: "4x4 Keypad Module Pinout, Configuration, Features, Circuit & Datasheet", *Components101.com*, 2019. [Online]. Available: https://components101.com/misc/4x4-keypad-module-pinout-configuration-features-datasheet. [Accessed: 13- May- 2019].

Tasks:

1. Connect a LCD with an Arduino and do the following:
    a. Display the reading of a LM35 sensor on the LCD from the rightmost corner.
2. Connect a LCD and a KeyPad with an Arduino and do the following:
    a. Display any keys that are pressed from the keypad on the LCD display.
    b. Arrange the characters displayed such that if the 1st row of display fills up then it will start printing on the 2nd row. If the 2nd row is also filled up then it will start overwriting from the 1st row again.