

## **LCD Keypad Shield for Arduino**



### **DESCRIPTION:**

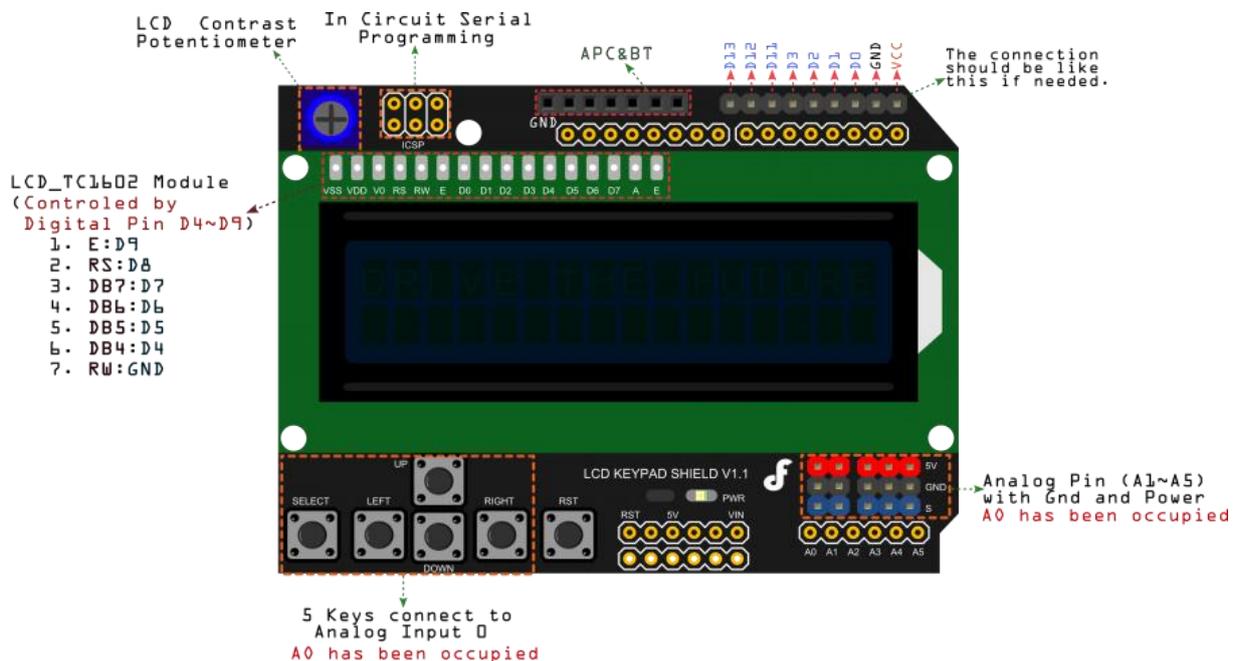
This is a very popular LCD Keypad shield for Arduino or Freeduino board. It includes a 2x16 LCD display and 6 momentary push buttons. Pins 4, 5, 6, 7, 8, 9 and 10 are used to interface with the LCD. Analog Pin 0 is used to read the push buttons. The LCD shield supports contrast adjustment and backlit on/off functions. It also expands analog pins for easy analog sensor reading and display.

The LCD Keypad shield is developed for Arduino compatible boards, to provide a user-friendly interface that allows users to go through the menu, make selections etc. It consists of a 1602 white character blue backlight LCD. The keypad consists of 5 keys — select, up, right, down and left. To save the digital IO pins, the keypad interface uses only one ADC channel. The key value is read through a 5 stage voltage divider.

### **Specification**

- Operating Voltage:5V
- 5 Push buttons to supply a custom menu control panel
- RST button for resetting arduino program
- Integrate a potentiometer for adjusting the backlight
- Expanded available I/O pins
- Expanded Analog Pinout with standard DFRobot configuration for fast sensor extension
- Dimensions: 3.15 in x 2.28 in x 0.79 in (8.0 cm x 5.8 cm x 2 cm)
- Weight: 1.94 oz (55 g)

## Pinout



Instruction for D4 To D10 and Analog Pin 0

| Pin             | Function                  | Instruction   |
|-----------------|---------------------------|---|
| Digital 4(D4)   |                           |   |
| Digital 5(D5)   | D4~D7 are used as DB4~DB7 | Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the LCD. |
| Digital 6(D6)   |                           |   |
| Digital 7(D7)   |                           |   |
| Digital 8(D8)   |                           |   |
| Digital 9(D9)   | Enable                    | Choose Data or Signal Display   |
| Digital 10(D10) | LCD Backlight Control     | Starts data read/write  |
| Analog 0(A0)    | Button select             | Select, up, right, down and left  |

The main reason this Arduino LCD shield is easy to use is because it is fully compatible with the [LiquidCrystal Arduino Library](#). All you have to do is just change the Pin numbers of the LCD from the sample LCD code and you are ready to go.

Another major advantage of this LCD & Keypad Shield is that the switches are all connected to only one ADC channel. That's right only one pin is required to read which one of the five switches is pressed. The pressed switch can be detected by reading analog value through a 5 stage voltage divider circuit. The shield also has onboard contrast control pot. You can also control the back-light intensity or turn it ON & OFF via Arduino.

## Compatibility

LCD & Keypad Shield is compatible with following Arduino Boards:

- Arduino UNO
- Arduino Mega2560
- Arduino Diecimila
- Duemilanove

## Power Required

The LCD & Keypad Shield requires 5 volt power supply. It worked well when I used external adaptor but while using USB supply it works well most of the time. Sometimes if my PC's USB is also powering some devices then the current required for the back-light control may decrease also this may cause voltage to drop & the 16x2 Alphanumeric LCD may not work properly. So make sure you are not powering lot of devices via USB port.

## LCD & Keypad Shield Pin Allocations

- A0 : All five Switches (Select, Up,Down, Left, Right)
- D4 : LCD DB4
- D5 : LCD DB5
- D6 : LCD DB6
- D7 : LCD DB7
- D8 : LCD RS
- D9 : LCD Enable
- D10: LCD Backlight Control

## LCD Interfacing

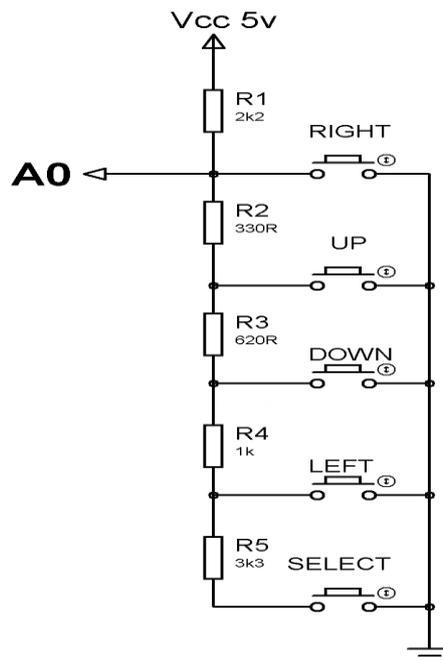
The LCD is interfaced to the Arduino in 4 bit mode i.e. only four data lines are required for interfacing along with Rs & enable. RW is connected directly to ground.

The LiquidCrystal Library that comes with the official Arduino Distribution software can be used for this LCD shield as well. Want to test it? Just open Hello world Example located at "Files -> Examples -> LiquidCrystal -> HelloWorld" and replace LiquidCrystal lcd(12, 11, 5, 4, 3, 2); withLiquidCrystal lcd( 8, 9, 4, 5, 6, 7 ); load the code and you are good to go.

## LCD Backlight Control

Arduino's D10 pin is connected to backlight so you control the backlight by controlling the D10 pin. So if you set D10 pin HIGH the LCD backlight will be fully ON & by setting it LOW the backlight will be OFF. D10 pin is also capable of PWM i.e. Pulse width Modulation. So using PWM you can vary the intensity of the backlight as per your requirement.

## Reading Switches



This is a bit complex to understand. As said earlier all five switches are connected to one Analog pin of the Arduino. This is done by using a chain of resistors that causes different voltage to be received by A0 depending on the switch being pressed. The right hand side image shows the schematic of the keypad section. Here is a list of voltages received on A0 pin as per the switch being pressed:-

- No Switch Pressed: 5 Volts
- UP Switch: 0.71V
- Down Switch: 1.61V
- Left Switch: 2.47V
- Select Switch: 3.62V

So if you read the Analog voltage using the “analogRead()” function you can determine which switch has been pressed. Though this method is a bit lengthy and tedious it does leave lot of free pins for other applications.

## **Library Explanation**

### **Function Explanation**

#### **LiquidCrystal(rs, enable, d4, d5, d6, d7)**

Creates a variable of type LiquidCrystal. The display can be controlled using 4 or 8 data lines. If the former, omit the pin numbers for d0 to d3 and leave those lines unconnected. The RW pin can be tied to ground instead of connected to a pin on the Arduino; if so, omit it from this function's parameters. for example:

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

#### **lcd.begin(cols, rows)**

Initializes the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands. for example:

```
lcd.begin(16, 2);
```

#### **lcd.setCursor(col,row)**

Set the location at which subsequent text written to the LCD will be displayed. for example:

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

#### **lcd.print(data)**

Prints text to the LCD. for example:

```
lcd.print("hello, world!");
```

#### **lcd.write(data)**

Write a character to the LCD.

**Schematic :**

