Introduction

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 pushbuttons. 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
- Dimension: 80 x 58 mm
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:

```cpp
cmd.begin(16, 2);
```

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

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

```cpp
cmd.setCursor(0, 0);
```

**`lcd.print(data)`**

Prints text to the LCD. For example:

```cpp
cmd.print("hello, world!");
```

**`lcd.write(data)`**

Write a character to the LCD.

Example 1

```cpp
/****************************************************************************
*********
Mark Bramwell, July 2010

This program will test the LCD panel and the buttons. When you push the
button on the shield,
the screen will show the corresponding one.

Connection: Plug the LCD Keypad to the UNO (or other controllers)
****************************************************************************
*********/
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7); // select the pins used on the
LCD panel

// define some values used by the panel and buttons
int lcd_key = 0;
int adc_key_in = 0;
#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
```
#define btnSELECT 4
#define btnNONE 5

int read_LCD_buttons(){
    // read the buttons
    adc_key_in = analogRead(0); // read the value from the sensor

    // my buttons when read are centered at these values: 0, 144, 329, 504, 741
    // we add approx 50 to those values and check to see if we are close
    // We make this the 1st option for speed reasons since it will be the most likely result
    if (adc_key_in > 1000) return btnNONE;

    // For V1.1 use this threshold
    if (adc_key_in < 50) return btnRIGHT;
    if (adc_key_in < 250) return btnUP;
    if (adc_key_in < 450) return btnDOWN;
    if (adc_key_in < 650) return btnLEFT;
    if (adc_key_in < 850) return btnSELECT;

    // For V1.0 comment the other threshold and use the one below:
    /*
     * if (adc_key_in < 50) return btnRIGHT;
     * if (adc_key_in < 195) return btnUP;
     * if (adc_key_in < 380) return btnDOWN;
     * if (adc_key_in < 555) return btnLEFT;
     * if (adc_key_in < 790) return btnSELECT;
     */

    return btnNONE; // when all others fail, return this.
}

void setup(){
    lcd.begin(16, 2); // start the library
    lcd.setCursor(0,0); // set the LCD cursor position
    lcd.print("Push the buttons"); // print a simple message on the LCD
}

void loop(){
    lcd.setCursor(9,1); // move cursor to second line "i" and 9 spaces over
    lcd.print(millis()/1000); // display seconds elapsed since power-up

    lcd.setCursor(0,1); // move to the beginning of the second line
    lcd_key = read_LCD_buttons(); // read the buttons

    switch (lcd_key){ // depending on which button was pushed, we perform an action
        case btnRIGHT:{ // push button "RIGHT" and show the word on the screen
            lcd.print("RIGHT ");
            break;
        }
    }
}
```java
    }
    case btnLEFT:
        lcd.print("LEFT "); // push button "LEFT" and show the word on the screen
        break;
    }
    case btnUP:
        lcd.print("UP "); // push button "UP" and show the word on the screen
        break;
    }
    case btnDOWN:
        lcd.print("DOWN "); // push button "DOWN" and show the word on the screen
        break;
    }
    case btnSELECT:
        lcd.print("SELECT"); // push button "SELECT" and show the word on the screen
        break;
    }
    case btnNONE:
        lcd.print("NONE "); // No action will show "None" on the screen
        break;
    }
}
```

Example 2

**Connection:**

Plug the LCD Keypad to the UNO(or other controllers)

Temperature sensor: S(blue) -- A1()

**Note: A0 has been occupied.**

VCC(red) -- VCC

GND(black) -- GND
Sample code

/*******************************************************************************/

Description:
Reads an analog input on pin 1, prints the result to the LCD.
This program takes the temperature sensor LM35 for example.

Connection:
Plug the LCD Keypad to the UNO (or other controllers)
Temperature sensor:
S (blue) -- A1()
    Note: A0 has been occupied.
VCC (red) -- VCC
GND (black) -- GND

******************************************************************************/
```cpp
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7); // select the pins used on the LCD panel

unsigned long tepTimer;

void setup(){
    lcd.begin(16, 2); // start the library
}

void loop(){
    lcd.setCursor(0, 0); // set the LCD cursor position

    int val; // variable to store the value coming from the analog pin
    double data; // variable to store the temperature value coming from the conversion formula
    val=analogRead(1); // read the analog in value:
    data = (double) val * (5/10.24); // temperature conversion formula

    if(millis() - tepTimer > 500){ // output a temperature value per 500ms
        tepTimer = millis();

        // print the results to the LCD
        lcd.print("T: ");
        lcd.print(data);
        lcd.print("C");
    }
}
```

Result
Trouble shooting

Q1. Why my LCD keypad cannot display anything on the Intel Edison while all right on Romeo?

A1: It works well if uploaded by Arduino 1.5.3 version, however, the latest 1.6.* have discard pin Definition for Edison. So you have to add pinMode(); into the setup() like this:

```c
void setup() {
  for(int i=4;i<10;i++){
    pinMode(i,OUTPUT);
  }
  lcd.begin(16, 2); // set up the LCD's number of columns and rows
}
Q2. I do not understand your schematic. There are too many connectors illustrated than are actually on the shield. Could you show me a mapping?

A2: The J1-J8 include the both the user interface, i.e. Analog pins, APC220(Serial) pins, Digital pins, and the pins connected with the lower Arduino card, e.g. Uno/ Leonardo. Here is a simple mapping picture.

Q3. I cannot get the LCD shield to work on Uno. I have a side by side comparison and the other works fine. I get the backlight but no alpha.

A3: