## The Liquid Crystal Displace (LCD) Shield

The pictures below show how to attach the LCD shield to the Arduino. You should be careful not to plug it too far into the input pins of the Arduino and connect the shield board with metallic parts of the Arduino as this can short some of the components (allow current to flow freely without resistance) which can cause the module not to work as expected.

|  |  |
| --- | --- |
| C:\Users\s2807774\Documents\GitHub\ArduinoTutes\ims\wiring\lcd1.jpg | C:\Users\s2807774\Documents\GitHub\ArduinoTutes\ims\wiring\lcd2.jpg |

No let’s take a look at some code. This example shows how to use both the LCD display to output characters, as well as reading input from the select, left, up, down and right buttons.

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

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

void setup() {

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

// Print a message to the LCD.

lcd.setCursor(0,0); //set the cursor to column 0, row 0

lcd.print("LCD Key Shield");

lcd.setCursor(0,1); //set the cursor to column 0, row 1

lcd.print("Press Key:");

}

void loop() {

int x;

x = analogRead (0); //read button input from Analog pin 0

lcd.setCursor(10,1); //move cursor to row 1, column 10

if (x < 60)

lcd.print ("Right ");

else if (x < 200)

lcd.print ("Up ");

else if (x < 400)

lcd.print ("Down ");

else if (x < 600)

lcd.print ("Left ");

else if (x < 800)

lcd.print ("Select");

}

Here we include the LiquidCrystal.h library header file. In global space we create an LiquidCrystal object called lcd and provide a list of interface pins (these are the pins to use with this module and the Arduino Uno).

In our setup function, we call lcd.begin(16, 2), the begin function takes two parameters, the number of columns (our shield has space for 16 characters from left to right) and the number of rows (out lcd shield provides two rows). The method for writing to the screen includes setting the cursor location to a particular row and column, before calling the print function.

In the next lines we write lcd.setCursor(0, 0), to set our cursor to the top left of the screen. Then lcd.print(“LCD Key Shield”) prints this string to the top left hand side of our LCD screen. The setCursor function takes the column value then the row value as parameters, and the print function can take any Arduino string, char, byte, long or int.

In the loop function, we detect input from the various buttons on the shield. If a particular button is detected, we write which button was pressed to the lcd screen. To detect a button press, we perform an analogue read from analogue input pin 0. This returns a value between 0 and 1023. If the value is in the range [0-59] then the right button was pressed. The table below lists the ranges for different button presses.

|  |  |
| --- | --- |
| Range (inclusive) | button |
| 800-1023 |  |
| 0-59 | Right button |
| 60-199 | Up button |
| 200-399 | Down button |
| 400-599 | Left button |
| 600-799 | Select button |