Lesson 12 LCD module

Introduction

In this tutorial, I’ll show you how to connect your LCD and program it in Python, using the RPLCD library. I’ll start with showing you how to connect it in either 8-bit mode or 4-bit mode. Then I’ll explain how to install the library, and provide examples for printing and positioning text, clearing the screen, and controlling the cursor. I’ll also give you examples for scrolling text, creating custom characters, printing data from a sensor, and displaying the date, time, and IP address of your Pi.

Principle

The principle behind the LCD’s is that when an electrical current is applied to the liquid crystal molecule, the molecule tends to untwist. This causes the angle of light which is passing through the molecule of the polarized glass and also cause a change in the angle of the top polarizing filter. As a result a little light is allowed to pass the polarized glass through a particular area of the LCD. Thus that particular area will become dark compared to other. The LCD works on the principle of blocking light. While constructing the LCD’s, a reflected mirror is arranged at the back. An electrode plane is made of indium-tin oxide which is kept on top and a polarized glass with a polarizing film is also added on the bottom of the device. The complete region of the LCD has to be enclosed by a common electrode and above it should be the liquid crystal matter

Hardware Required

- Raspberry Pi
- Breadboard
- Buzzer (Active)
- Potentiometer (3KΩ -10KΩ)
- Resistor (1KΩ)
- Jumper wires

Write to the display in 4-bit mode

In 4-bit mode, only LCD pins D4, D5, D6, and D7 are used for data. These are set in pins data= [D4, D5, D6, D7] on line 2 below:

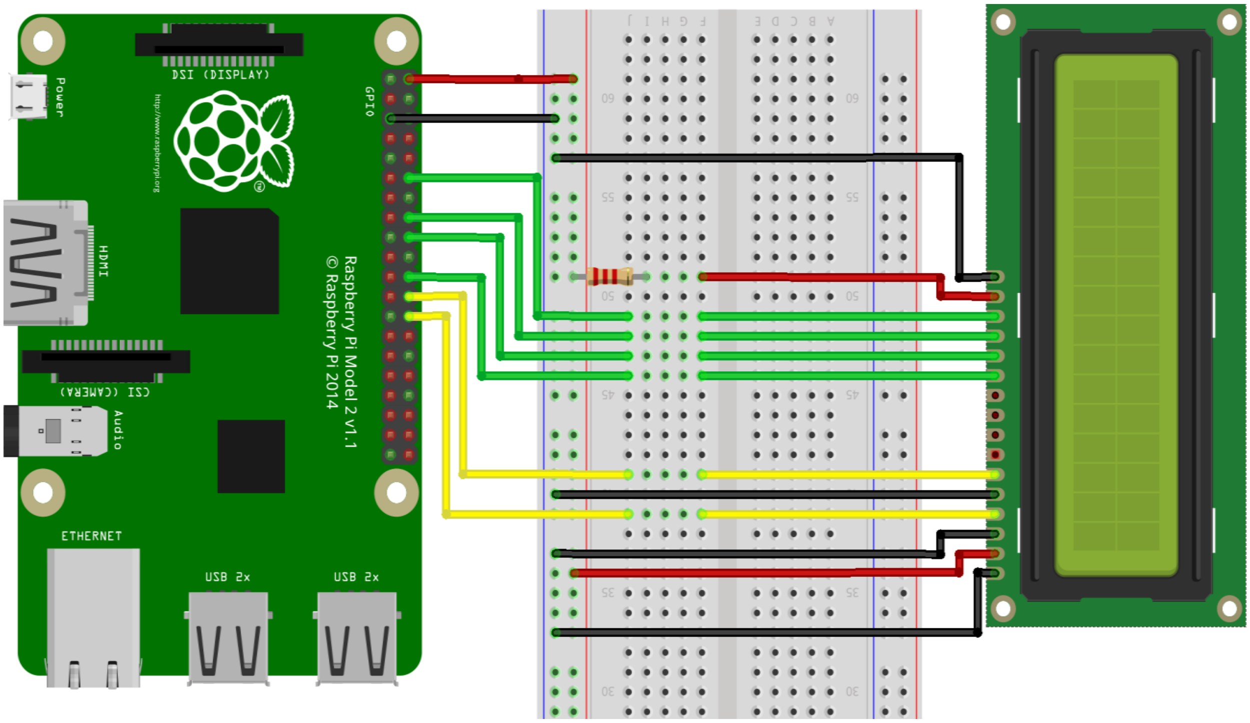
```
from RPLCD import CharLCD

lcd = CharLCD(cols=16, rows=2, pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12])
```

```
lcd.write_string(u'Hello world!')
```

WIRING THE LCD IN 4 BIT MODE

To connect the LCD to your Raspberry Pi in 4-bit mode, set it up like this:



PROGRAMMING THE LCD WITH PYTHON

We'll be using a Python library that provides a lot of useful functions. It called the RLPCD library

PRINT Hello World

```
from RPLCD import CharLCD

lcd = CharLCD(cols=16, rows=2, pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12])

lcd.write_string(u'Hello world!')
```

LINE BREAKS

Text will automatically wrap to the next line if the length of the text is greater than the column length of your LCD. You can also control where the text string breaks to the next line by inserting \n\r where you want the break to occur. The code below will print “Hello” to the top row, and “world!” to the bottom row.

```
from RPLCD import CharLCD

lcd = CharLCD(cols=16, rows=2, pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12])

lcd.write_string(u'Hello\n\rworld!')
```

CLEAR THE SCREEN

The function lcd.clear() will clear the screen. The following code will print “Hello world!” to the screen for two seconds before clearing it:

```
import time

from RPLCD import CharLCD

lcd = CharLCD(cols=16, rows=2, pin_rs=26, pin_e=24, pins_data=[22, 18, 16, 12])

lcd.write_string(u'Hello world!')

time.sleep(2)

lcd.clear()
```

Application

- Weather Monitoring System
- Home Automation
- Ultrasonic Sensor Project