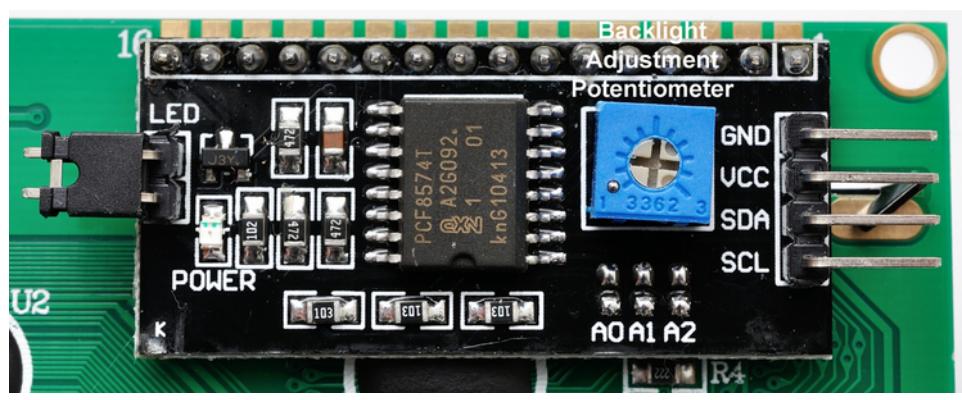


Using a 20x4 i²c Character LCD with RasPiO Duino



i2c LCD 20x4 characters

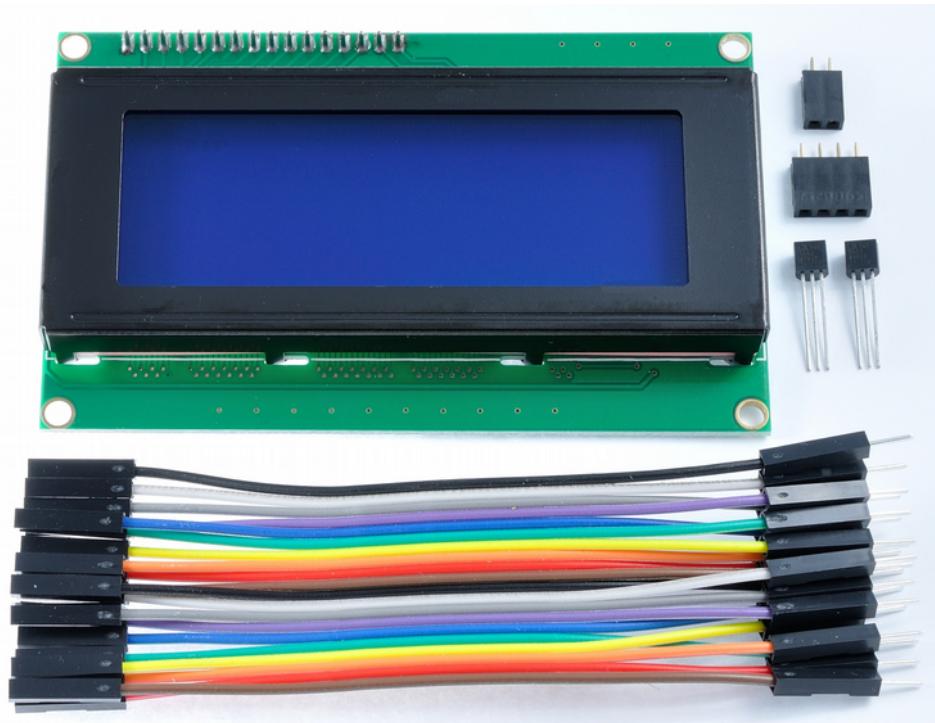
Character LCDs are a great way of showing output from the RasPiO Duino. You can get both bare and i²c displays. i²c displays only require 4 wires, using 2 ports on the Duino. Whereas the bare displays require 11 or more wires to function, using up most of the I/O ports. The displays are the same, but the i²c ones have a little “backpack” that handles the conversion between the Hitachi HD44780 interface and i²c for you. The jumper enables/disables the backlight and the blue potentiometer is for backlight contrast adjustment.



i2c backpack on LCD

The RasPiO LCD20x4 kit has been put together with the RasPiO Duino in mind, although it will work directly from the Raspberry Pi's i²c ports as well. You can find the kit here...

<http://rasp.io/lcd20>



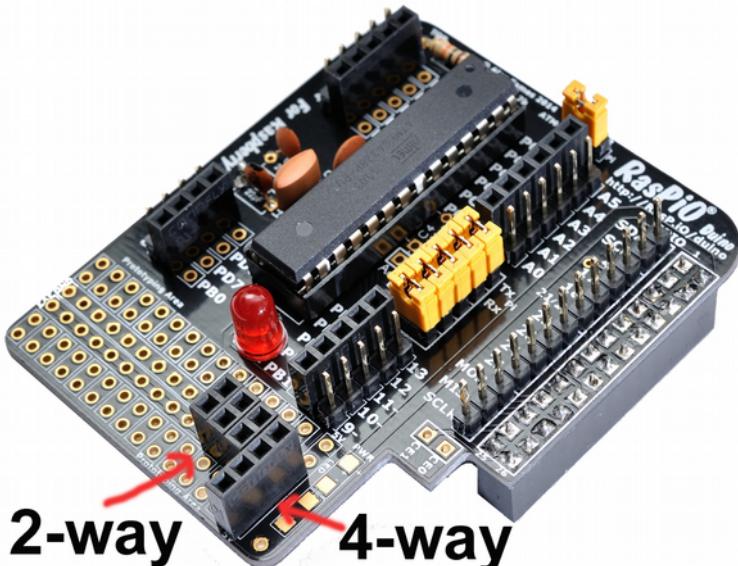
RasPiO LCD20x4 kit contents

The kit contains...

- a lovely 20x4 LCD with blue backlight and pre-soldered i²c backpack
- 2 x TMP-36 analog temperature sensors
- 1 two-way female header
- 1 four-way female header
- 20 M-F jumper wires (10cm)

Soldering the Headers

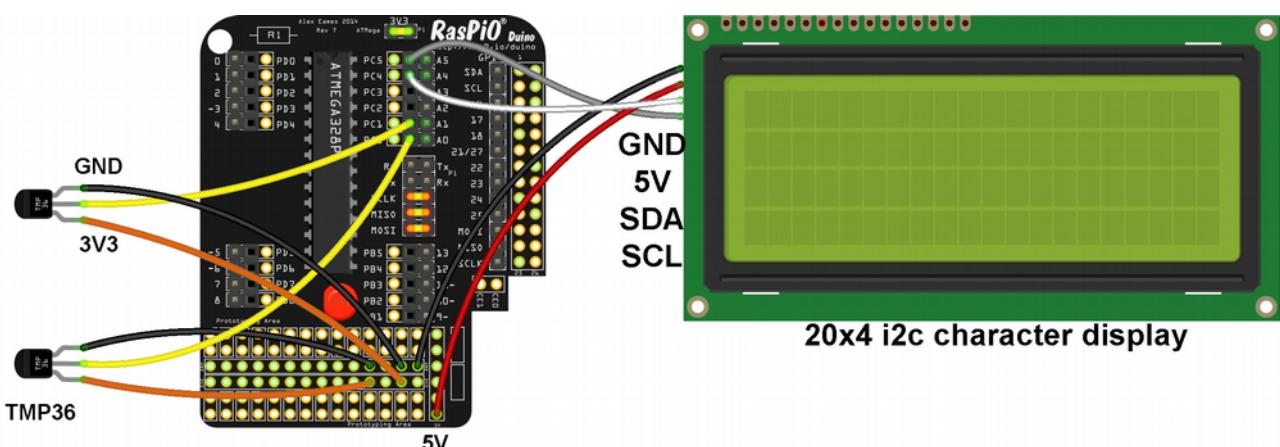
You will probably need to add some extra headers on 5V, 3V3 & GND rails. There is a female 4-way header provided for use on the 5V rail. The additional 2-way female header is to add an extra socket on each of the 3V3 and GND rails. This is what the RasPiO Duino should look like once they're added...



RasPiO Duino with extra headers soldered on

Wiring the Duino to the Display

A5/PC5 on duino	to SCL on display (grey wire)
A4/PC4 on duino	to SDA on display (white wire)
5V on duino	to VCC on display (red wire)
GND on duino	to GND on display (black wire)

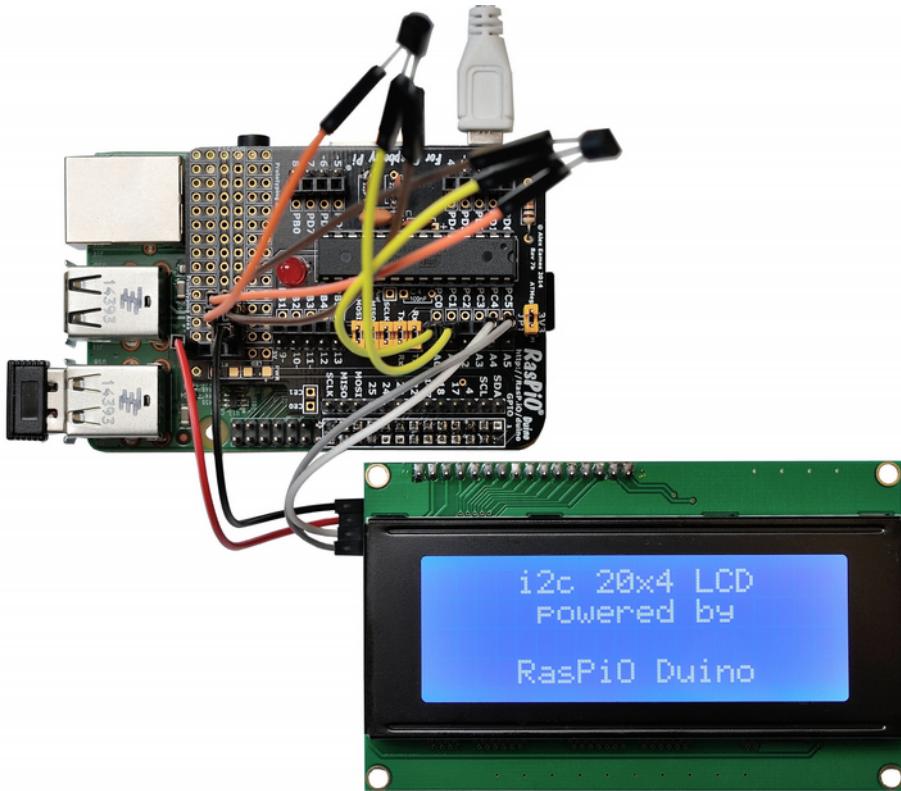


Wiring a 20x4 i2c display and TMP36 sensors

Wiring the Temperature Sensors to the Duino

- Pin 1 to Duino 3V3 (orange wires)
- Pin 2 to Duino A0 or A1 analog port (yellow wires)
- Pin 3 to Duino GND (black wires)

This is what it looks like all wired up using the kit components...



RasPiO LCD20x4 in use

Installing the Software

We're using an open source library called `LiquidCrystal_I2C1602v1` to drive the i2c LCD, so we'll install that next. From the command line, type...

```
curl rasp.io/duinolcd.sh | bash
```

If you want to check out the script before you run it, you can find it here...
https://github.com/raspitv/raspio_arduino/blob/master/duinolcd.sh

The script will download a zip file, unzip it and place the `LiquidCrystal_I2C1602v1` library files in `/usr/share/arduino/libraries`

Then it refreshes your `/home/pi/raspio_arduino` directory to ensure it has the latest files. Then the example sketch `LCD_thermometer_20x4.ino` is placed in `/home/pi/sketchbook` and the script cleans up after itself by deleting all the temporary files it created. When run, your on-screen output should look something like this...

```

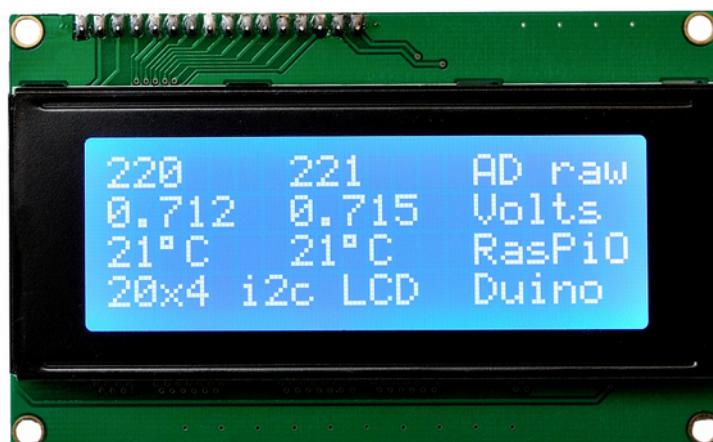
pi@raspberrypi:~/raspio_arduino
File Edit Tabs Help
inflating: __MACOSX/LiquidCrystal_I2C1602V1/_examples
inflating: LiquidCrystal_I2C1602V1/keywords.txt
inflating: __MACOSX/LiquidCrystal_I2C1602V1/_keywords.txt
inflating: LiquidCrystal_I2C1602V1/LiquidCrystal_I2C.cpp
inflating: __MACOSX/LiquidCrystal_I2C1602V1/_LiquidCrystal_I2C.cpp
inflating: LiquidCrystal_I2C1602V1/LiquidCrystal_I2C.h
inflating: __MACOSX/LiquidCrystal_I2C1602V1/_LiquidCrystal_I2C.h
inflating: LiquidCrystal_I2C1602V1/LiquidCrystal_I2C.o
inflating: __MACOSX/LiquidCrystal_I2C1602V1/_LiquidCrystal_I2C.o
inflating: __MACOSX/_LiquidCrystal_I2C1602V1
remote: Counting objects: 4, done.
remote: Total 4 (delta 1), reused 1 (delta 1), pack-reused 3
Unpacking objects: 100% (4/4), done.
From https://github.com/raspitv/raspio_arduino
  97392ac..df27675 master      -> origin/master
Updating 97392ac..df27675
Fast-forward
 LCD_thermometer_20x4/LCD_thermometer_20x4.ino | 80 ++++++
 1 file changed, 80 insertions(+)
 create mode 100644 LCD_thermometer_20x4/LCD_thermometer_20x4.ino
pi@raspberrypi ~ $
```

On-screen output during software installation

Once you've wired your display, you can upload the sketch `LCD_thermometer_20x4.ino` to the RasPiO Duino (remember to Upload Using Programmer) and it should give you 5 seconds of this startup screen...



...followed by a display of the output from both temperature sensors...



Thermometer sketch output

LCD_thermometer_20x4.ino Sketch Walk-through

This sketch is quite long, but most of it deals with what is being 'printed' on the LCD screen. The full sketch [is found here](#). As usual, the first few lines are comment, describing the sketch. After that we have lines 7 & 8...

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

...which ensure that the required libraries are loaded when we compile the sketch. In lines 9-15 we're just declaring variables (nothing new there). In 16 onwards we start doing new things, so let's have a closer look at those parts...

```
16 // set LCD i2c address to 0x27 and 20 char x 4 line display
17 LiquidCrystal_I2C lcd(0x27,20,4);
18
19 void setup()
20 {
21     lcd.init();                      // initialize lcd
22     lcd.backlight();
23     lcd.setCursor(0, 0);             // go to column 0 row 0
24     lcd.print("    i2c 20x4 LCD");   // Print txt on LCD
25     lcd.setCursor(0, 1);
26     lcd.print("    powered by");
27     lcd.setCursor(0, 3);
28     lcd.print("    RasPiO Duino");
29     delay(5000);
30     lcd.clear();
31 }
```

LCD_thermometer_20x4.ino lines 16-31

In line 17 we're setting the i²c address (0x27) of the LCD's i²c backpack. The 20 and the 4 set the number of columns and rows on the screen.

In 19-31 we write our setup function.

21 lcd.**init()**; initialises the LCD.

22 lcd.**backlight()**; turns on the backlight (you can turn it off with lcd.**noBacklight()**; if you ever need to do that).

23 lcd.**setCursor(0, 0)**; sets the current position to column 0, row 0 (top left) of the LCD

24 lcd.**print(" i2c 20x4 LCD")**; prints the supplied text to the LCD. You have to be careful that your text will fit the number of characters available if your output is to make sense.

25-28 deal with more of the same – displaying text.

29 **delay(5000)**; causes a 5 second wait, so that you can appreciate the full glory of the “Splash screen”.

30 lcd.**clear()**; clears the LCD ready for us to start showing our ADC output, voltages and temperatures.

The rest of the sketch uses procedures we've already learned about in the [analog section](#).

In lines 35-56 we're reading the two analog channels 0 and 1, and displaying their raw values on the top row (0) of the LCD.

Lines 58-65 we convert those ADC values into voltages and display those on the second row (1) of the LCD.

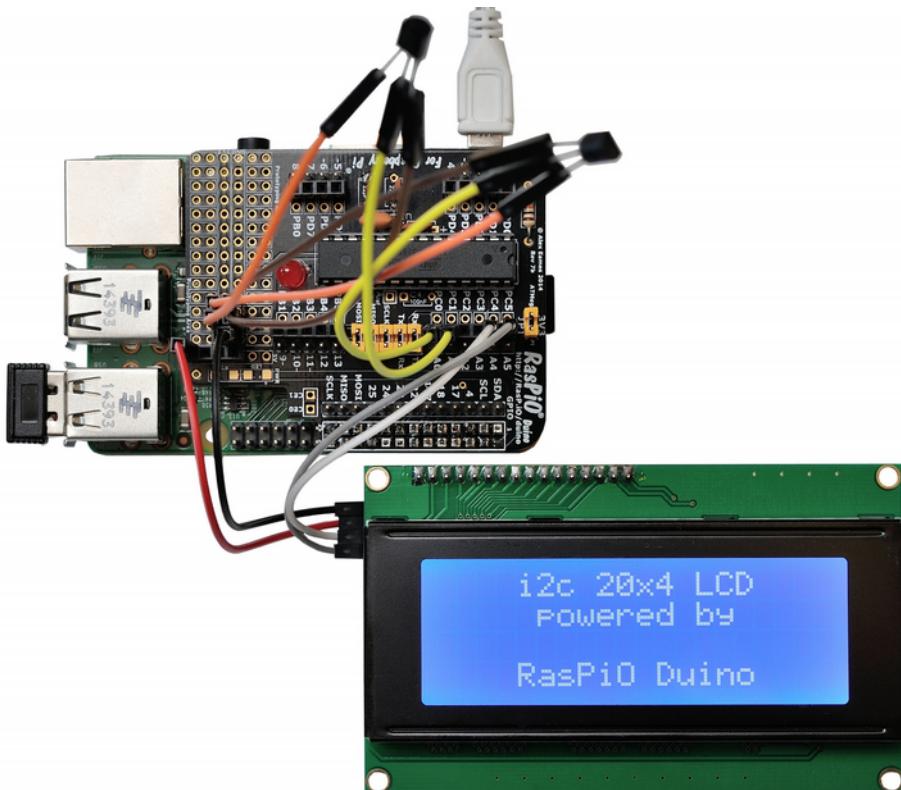
Lines 67-75 we calculate temperature values from the measured voltages and display those temperature values on the third row (2) of the LCD.

Lines 77-78 we just print some constant text on the fourth row (3) of the LCD.

Line 79 we wait half a second before starting the main loop again.

So that's a brief introduction to using the RasPiO LCD20x4 kit with RasPiO Duino. I hope you have fun using it...

<http://rasp.io/lcd20>



RasPiO LCD20x4 running on RasPiO Duino