Arduino Lesson 12. LCD Displays - Part 2
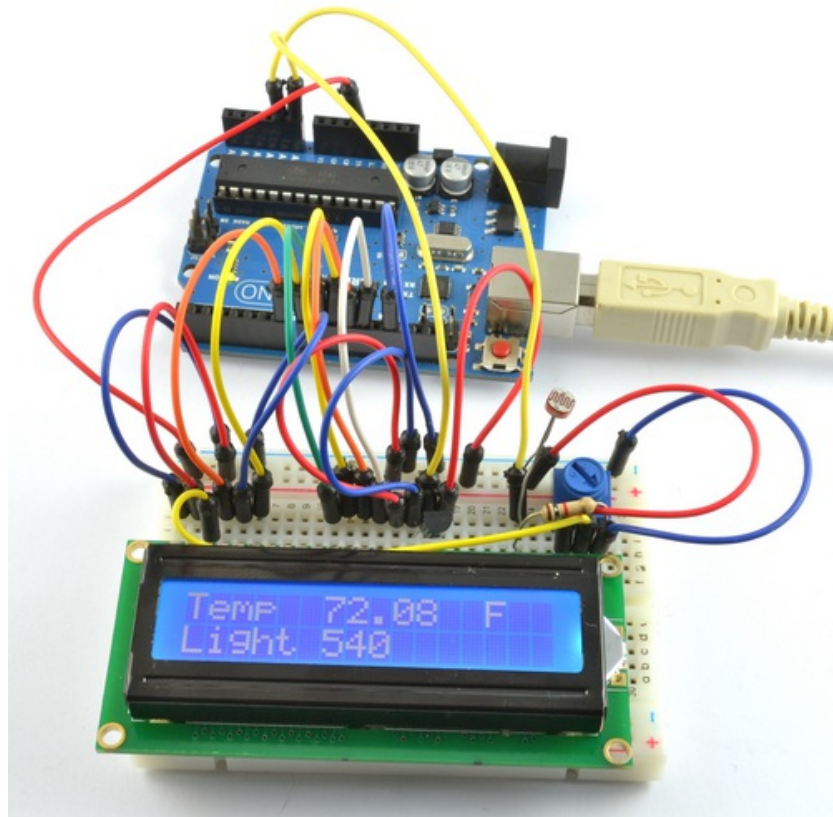
Created by Simon Monk
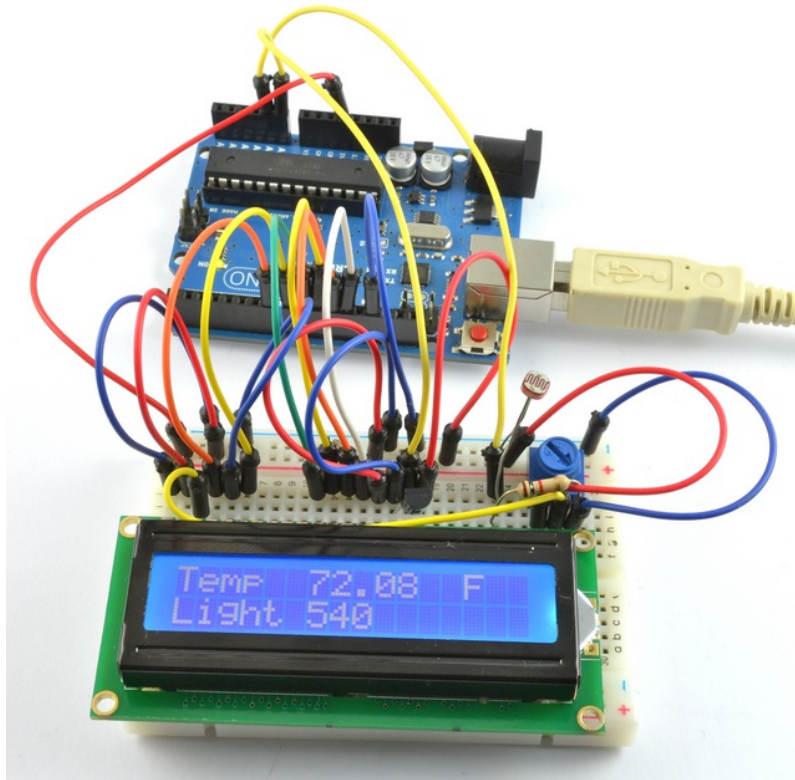

Last updated on 2013-06-22 06:45:32 PM EDT

Guide Contents

Overview

In this lesson, you will build on what we have learnt in lesson 11 and use a LCD display to show the temperature and light intensity.
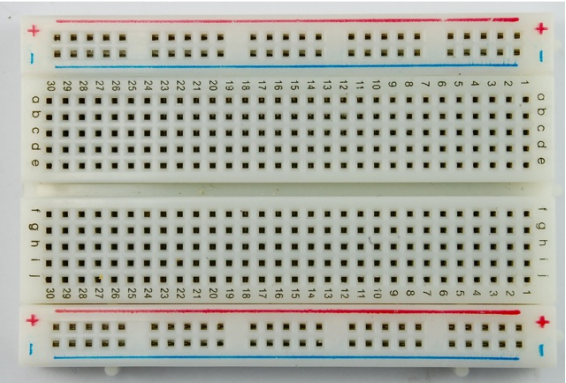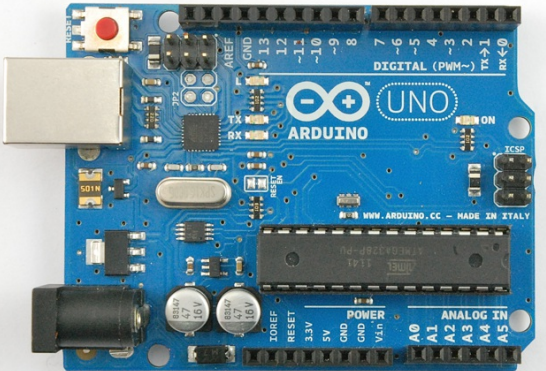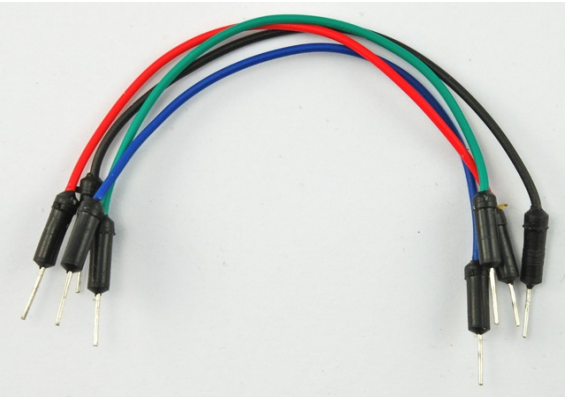

Light intensity is measured using the same photocell that you used in lesson 9.

To measure the temperature, you will use a temperature measurement chip. This device has just three leads two for 5V and GND and the third lead is connected directly to an analog input on the Arduino.

Parts

	Part	Qty
	LCD Display (16x2 characters)	1
	10 kΩ variable resistor (pot)	1
	1 kΩ Resistor (brown, black, red stripes)	1
	Photocell (Light Dependent Resistor)	1

	TMP36 temperature sensor	1
	Half-size Breadboard	1
	Arduino Uno R3	1
	Jumper wire pack	1

The TMP36 looks just like the PN2222 transistor, but if you look at the flat side of the package body, you should see it labelled as a TMP36.

Breadboard Layout

The breadboard layout is based on the layout from lesson 11, so if you still have this on the breadboard it will simplify things greatly.
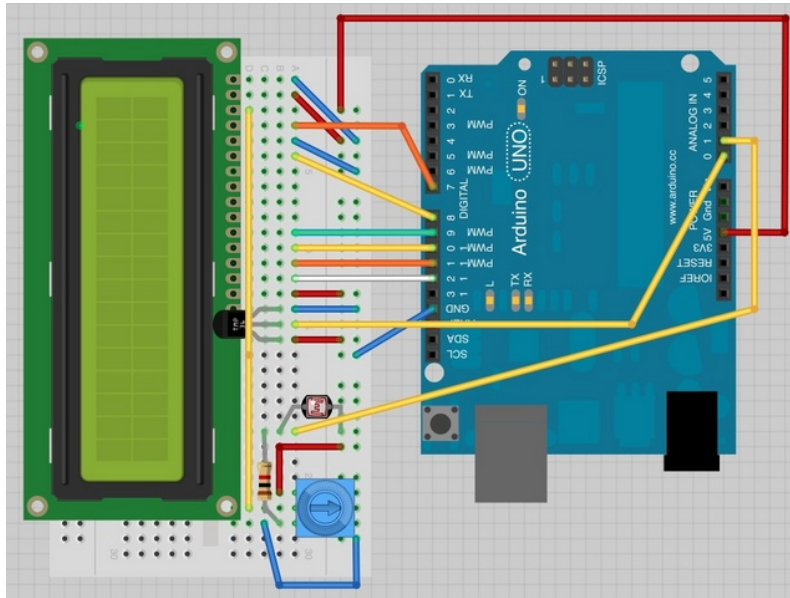
There are a few jumper wires that have been moved slightly on this layout. In particular, those near the pot.

The photocell, 1 kΩ resistor and TMP36 are all new additions to the board. The TMP36 has its curved face towards the display.

Arduino Code

The sketch for this is based on that of lesson 11. Load it up onto your Arduino and you should find that warming the temperature sensor by putting your finger on it will increase the temperature reading.

Also if you wave your hand over the photocell blocking out some of the light, the reading will decrease.

```
/*
Adafruit Arduino - Lesson 12. Light and Temperature
*/

#include <LiquidCrystal.h>

int tempPin = 0;
int lightPin = 1;

//             BS  E  D4 D5  D6 D7
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

void setup()
{
  lcd.begin(16, 2);
}

void loop()
{
  // Display Temperature in C
  int tempReading = analogRead(tempPin);
  float tempVolts = tempReading * 5.0 / 1024.0;
  float tempC = (tempVolts - 0.5) * 100.0;
  float tempF = tempC * 9.0 / 5.0 + 32.0;
  //        ----------------
  lcd.print("Temp      F ");
  lcd.setCursor(6, 0);
  lcd.print(tempF);

  // Display Light on second row
  int lightReading = analogRead(lightPin);
  lcd.setCursor(0, 1);
  //        ---------------
  lcd.print("Light        ");
  lcd.setCursor(6, 1);
  lcd.print(lightReading);
  delay(500);
}
```

I find it useful to put a comment line above the 'lcd' command.

```
//              BS  E  D4 D5  D6 D7
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

This makes things easier if you decide to change which pins you use.

In the 'loop' function there are now two interesting things going on. Firstly we have to convert the analog from the temperature sensor into an actual temperature, and secondly we have to work out how to display them.

First of all, let's look at calculating the temperature.

```
int tempReading = analogRead(tempPin);
float tempVolts = tempReading * 5.0 / 1024.0;
float tempC = (tempVolts - 0.5) * 100.0;
float tempF = tempC * 9.0 / 5.0 + 32.0;
```

The raw reading from the temperature sensor is first multiplied by 5 and then divided by 1024 to give us the voltage (between 0 and 5) at the 'tempPin' analog input.

To convert the voltage coming from the TMP36 into a temperature in degrees C, you have to subtract 0.5V from the measurement and then multiply by 100.

To convert this into a temperature in Fahrenheit, you then have to multiply it by 9/5 and then add 32.

Displaying changing readings on an LCD display can be tricky. The main problem is that the reading may not always be the same number of digits. So, if the temperature changed from 101.50 to 99.00 then the extra digit from the old reading is in danger of being left on the display.

To avoid this, write the whole line of the LCD each time around the loop.

```
//       ----------------
lcd.print("Temp       F ");
lcd.setCursor(6, 0);
lcd.print(tempF);
```

The rather strange comment serves to remind you of the 16 columns of the display. You can then print a string of that length with spaces where the actual reading will go.

To fill in the blanks, set the cursor position for where the reading should appear and then print it.

Exactly the same approach is used for displaying the light level. There are no units for the light level, we just display the raw reading from the analog read.

Other Things to Do

Try changing the example so that it displays the temperature in degrees C rather than F.

About the Author

Simon Monk is author of a number of books relating to Open Source Hardware. The following books written by Simon are available from Adafruit: Programming Arduino (http://adafru.it/1019), 30 Arduino Projects for the Evil Genius (http://adafru.it/868) and Programming the Raspberry Pi (http://adafru.it/aM5).