Cristal Liquido. Experiment 17

F541- Prof David M Soares

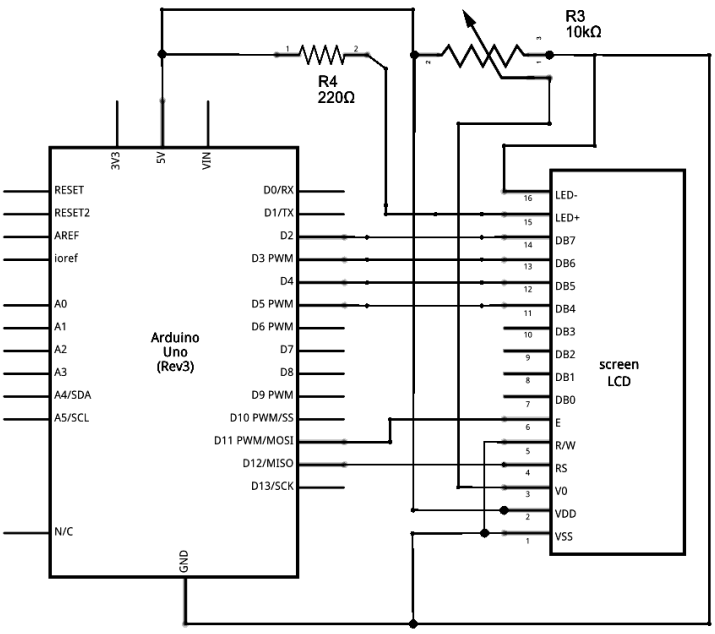
Este programa foi retirado do site:
https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay

```
/*

  LiquidCrystal Library - display() and noDisplay()
 Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
 library works with all LCD displays that are compatible with the
 Hitachi HD44780 driver. There are many of them out there, and you
 can usually tell them by the 16-pin interface.

 This sketch prints "Hello World!" to the LCD and uses the
 display() and noDisplay() functions to turn on and off
 the display.
 The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)

 Library originally added 18 Apr 2008
 by David A. Mellis
 library modified 5 Jul 2009
 by Limor Fried (http://www.ladyada.net)
 example added 9 Jul 2009
 by Tom Igoe
 modified 22 Nov 2010
 by Tom Igoe
 modified 7 Nov 2016
 by Arturo Guadalupi
 This example code is in the public domain.
 http://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay
*/
```

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("hello, world!");
}

void loop() {
  // Turn off the display:
  lcd.noDisplay();
  delay(500);
  // Turn on the display:
  lcd.display();
  delay(500);
}
```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

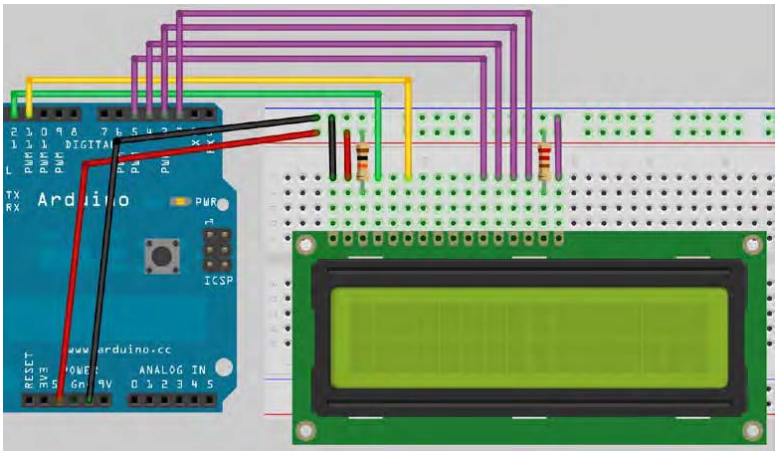
Circuit:

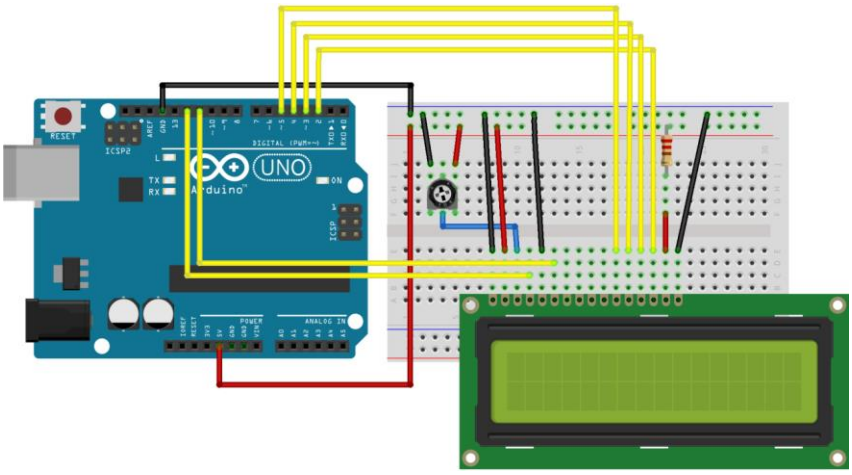


Material :

Arduino or Genuino Board

- LCD Screen (compatible with Hitachi HD44780 driver)
- pin headers to solder to the LCD display pins
- 10k ohm potentiometer
- 220 ohm resistor
- hook-up wires
- breadboard





LCD programs:

The main program loop simply runs seven different demo routines, one by one, before restarting. Each demo routine shows off one set of related routines in the LiquidCrystal.h library:

```
// PROJECT 23
#include <LiquidCrystal.h>
// Initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2); // Create an lcd object and assign the pins

void setup() {
lcd.begin(16, 2); // Set the display to 16 columns and 2 rows
```

```
}
void loop() {
// Run the seven demo routines
basicPrintDemo();
displayOnOffDemo();
setCursorDemo();
scrollLeftDemo();
scrollRightDemo();
cursorDemo();
createGlyphDemo();
}
void basicPrintDemo() {
lcd.clear(); // Clear the display
lcd.print("Basic Print"); // Print some text
delay(2000);
}
void displayOnOffDemo() {
lcd.clear(); // Clear the display
lcd.print("Display On/Off"); // Print some text
for(int x=0; x < 3; x++) { // Loop 3 times
lcd.noDisplay(); // Turn display off
delay(1000);
lcd.display(); // Turn it back on again
delay(1000);
}
}
void setCursorDemo() {
lcd.clear(); // Clear the display
lcd.print("SetCursor Demo"); // Print some text
delay(1000);
lcd.clear(); // Clear the display
lcd.setCursor(5,0); // Cursor at column 5 row 0
lcd.print("5,0");
delay(2000);
lcd.setCursor(10,1); // Cursor at column 10 row 1
lcd.print("10,1");
delay(2000);
lcd.setCursor(3,1); // Cursor at column 3 row 1
lcd.print("3,1");
delay(2000);
}

void scrollLeftDemo() {
lcd.clear(); // Clear the display
lcd.print("Scroll Left Demo");
delay(1000);
lcd.clear(); // Clear the display
lcd.setCursor(7,0);
lcd.print("Beginning");
lcd.setCursor(9,1);
lcd.print("Arduino");
delay(1000);
for(int x=0; x<16; x++) {
lcd.scrollDisplayLeft(); // Scroll display left 16 times
delay(250);
}
}
void scrollRightDemo() {
lcd.clear(); // Clear the display
lcd.print("Scroll Right");
lcd.setCursor(0,1);
lcd.print("Demo");
delay(1000);
lcd.clear(); // Clear the display
lcd.print("Beginning");
lcd.setCursor(0,1);
lcd.print("Arduino");
```

```
delay(1000);
for(int x=0; x<16; x++) {
lcd.scrollDisplayRight(); // Scroll display right 16 times
delay(250);
}
}
void cursorDemo() {
lcd.clear(); // Clear the display
lcd.cursor(); // Enable cursor visible
lcd.print("Cursor On");
delay(3000);
lcd.clear(); // Clear the display
lcd.noCursor(); // Cursor invisible
lcd.print("Cursor Off");
delay(3000);
lcd.clear(); // Clear the display
lcd.cursor(); // Cursor visible
lcd.blink(); // Cursor blinking
lcd.print("Cursor Blink On");
delay(3000);
lcd.noCursor(); // Cursor invisible
lcd.noBlink(); // Blink off
}

void createGlyphDemo() {
lcd.clear();
byte happy[8] = { // Create byte array with happy face
B00000,
B00000,
B10001,
B00000,
B10001,
B01110,
B00000,
B00000};
byte sad[8] = { // Create byte array with sad face
B00000,
B00000,
B10001,
B00000,
B01110,
B10001,
B00000,
B00000};
lcd.createChar(0, happy); // Create custom character 0
lcd.createChar(1, sad); // Create custom character 1
for(int x=0; x<5; x++) { // Loop animation 5 times
lcd.setCursor(8,0);
lcd.write(0); // Write custom char 0
delay(1000);
lcd.setCursor(8,0);
lcd.write(1); // Write custom char 1
delay(1000);
}
}
```

The final function called **createGlyphDemo()** creates a custom character. Most LCDs let you program your own custom characters to them. The standard 16□2 LCD has space to store eight custom characters in memory. The characters are 5 pixels wide by 8 pixels high (a pixel is a picture element, i.e. the individual dots that make up a digital display). The display is cleared and then two arrays of type byte are initialized with the

binary pattern of a happy and a sad face. The binary patterns are 5 bits wide. Then you create the two custom characters using the **createChar()** command. This requires twoparameters: the first is the number of the custom character (0 to 7 in the case of my test LCD, which can store a maximum of 8), and the second is the name of the array that creates and stores the custom characters binary pattern in memory on the LCD:

```
lcd.createChar(0, happy); // create custom character 0
lcd.createChar(1, sad); // create custom character 1
```

A **for** loop will now loop through itself five times. On each iteration the cursor is set to column 8 and row 0, and the first custom character is written to that location using the **write()** command. This writes the custom character within the brackets to the cursor location. The first character, a happy face, is written to the cursor location; after a delay of one second the second character, a sad face, is then written to the same cursor location. This repeats five times to make a crude animation.

```
for(int x=0; x<5; x++) { // loop animation 5 times
lcd.setCursor(8,0);
lcd.write(0); // write custom char 0
delay(1000);
lcd.setCursor(8,0);
lcd.write(1); // write custom char 1
delay(1000);
}
}
```