

Display LCD 16x2 com Keypad

Seminário 2 de INF 350 - Prof.: Ricardo dos Santos Ferreira

Grupo:

Juliana Moreno - 75763

Michael Canesche - 68064

Vanessa Vasconcelos - 77427

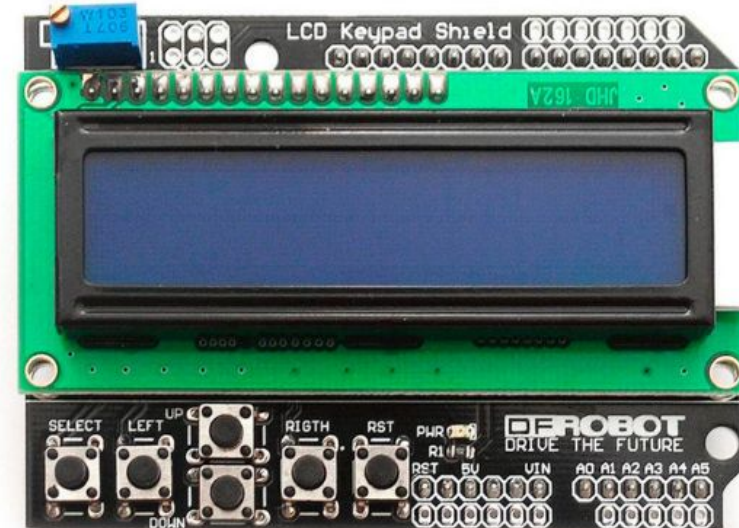


Especificação Geral

Componente chamado shield LCD 16x2 com Keypad

- Possui um display LCD 16x2 com backlight azul
- 5 botões customizáveis e 1 botão de reset
- Potenciômetro para ajuste do contraste
- Furos, acima do display, para soldagem de barras de pinos

Uso da biblioteca: ***LiquidCrystal lcd(8,9,4,5,6,7);***



Relação entre os pinos do Arduino e os sinais do display :

Arduino - Pino		Display - Sinal
4	<==>	DB4
5	<==>	DB5
6	<==>	DB6
7	<==>	DB7
8	<==>	RS
9	<==>	Enable

Pino 7 do Arduino vai no 14 do display (Pino 14: DB7)

Pino 6 do Arduino vai no 13 do display (Pino 13: DB6)

Pino 5 do Arduino vai no 12 do display (Pino 12 : DB5)

Pino 4 do Arduino vai no 11 do display (Pino 11: DB4)

Pino 9 do Arduino vai no 6 (Enable) do display

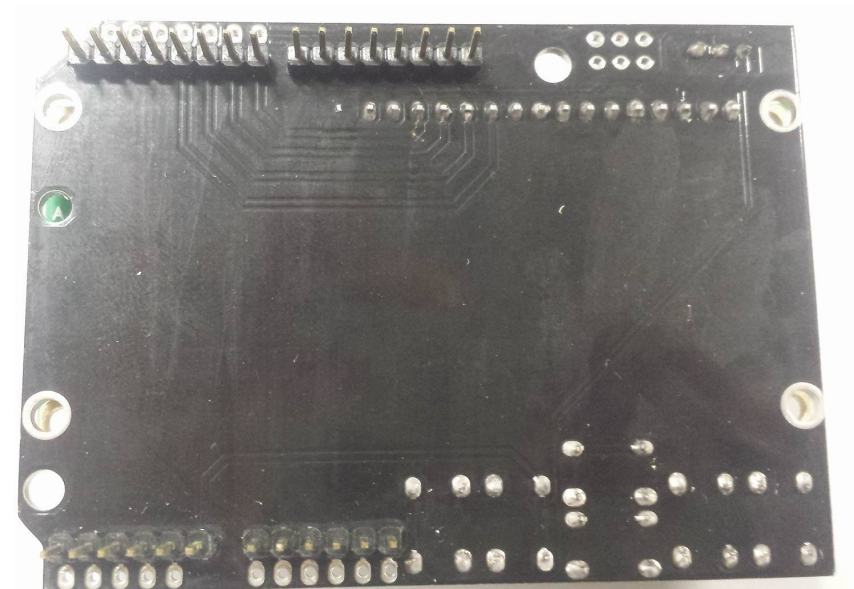
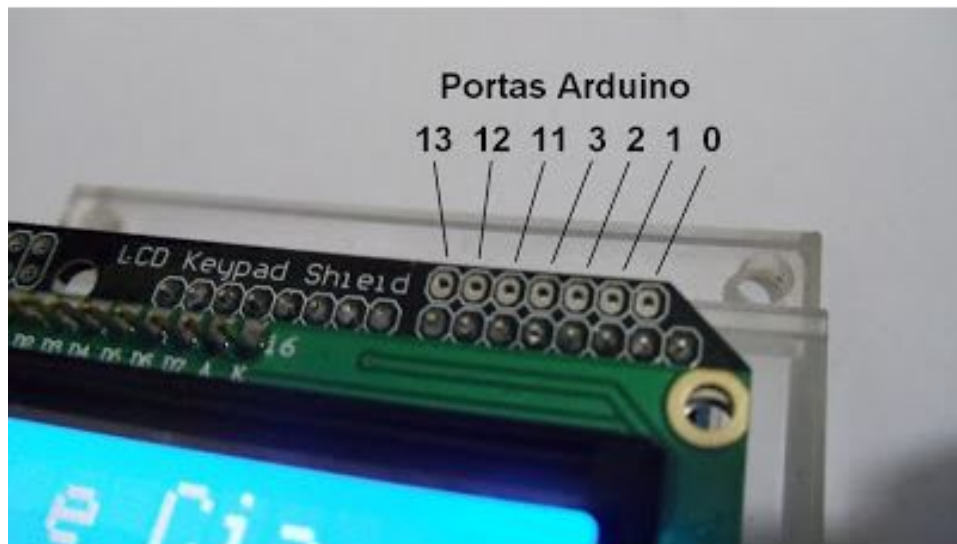
Pino 8 do Arduino vai no 4 (RS) do display

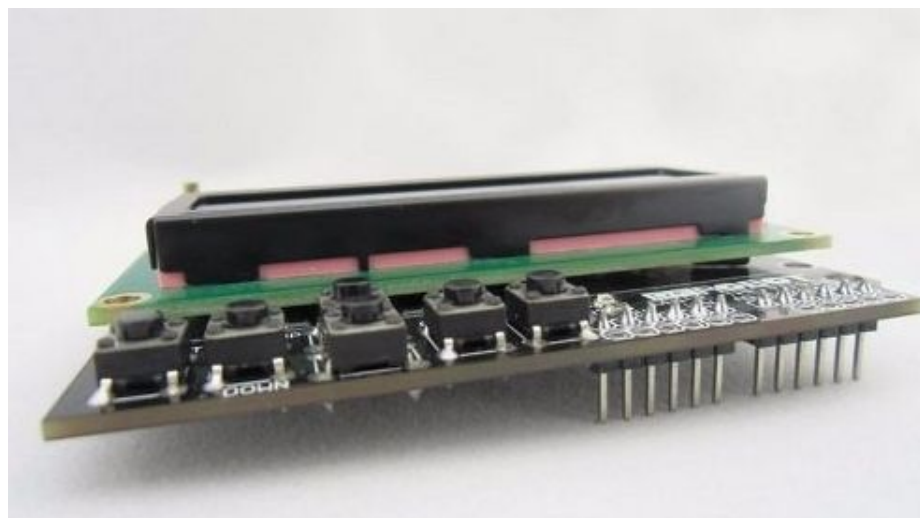
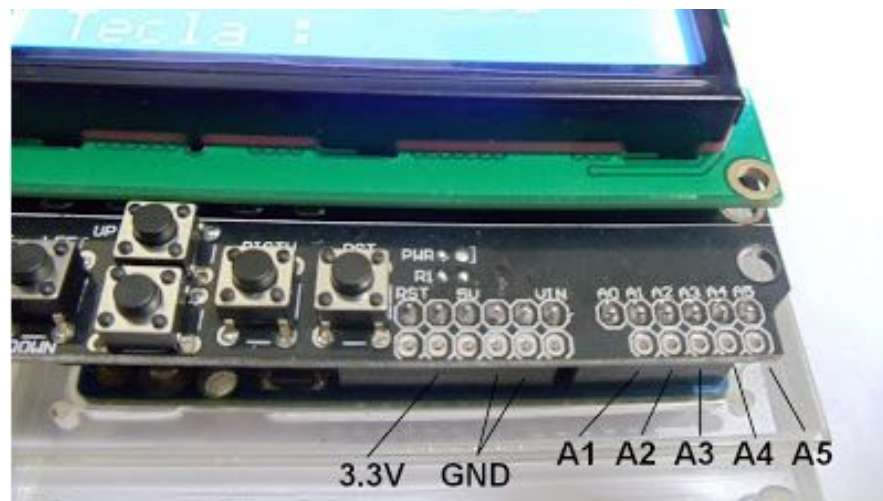


A placa possui furos que permitem a soldagem de uma barra de pinos ou conectores

Usa-se os pinos : **0, 1, 2, 3, 11, 12, 13 (digitais)**, e **A1, A2, A3, A4, A5 (analógicos)** e de energia (**5v, 3.3v, GND**).

Obs: o pino 10 também é utilizado pela placa para uma função específica, no caso o backlight.

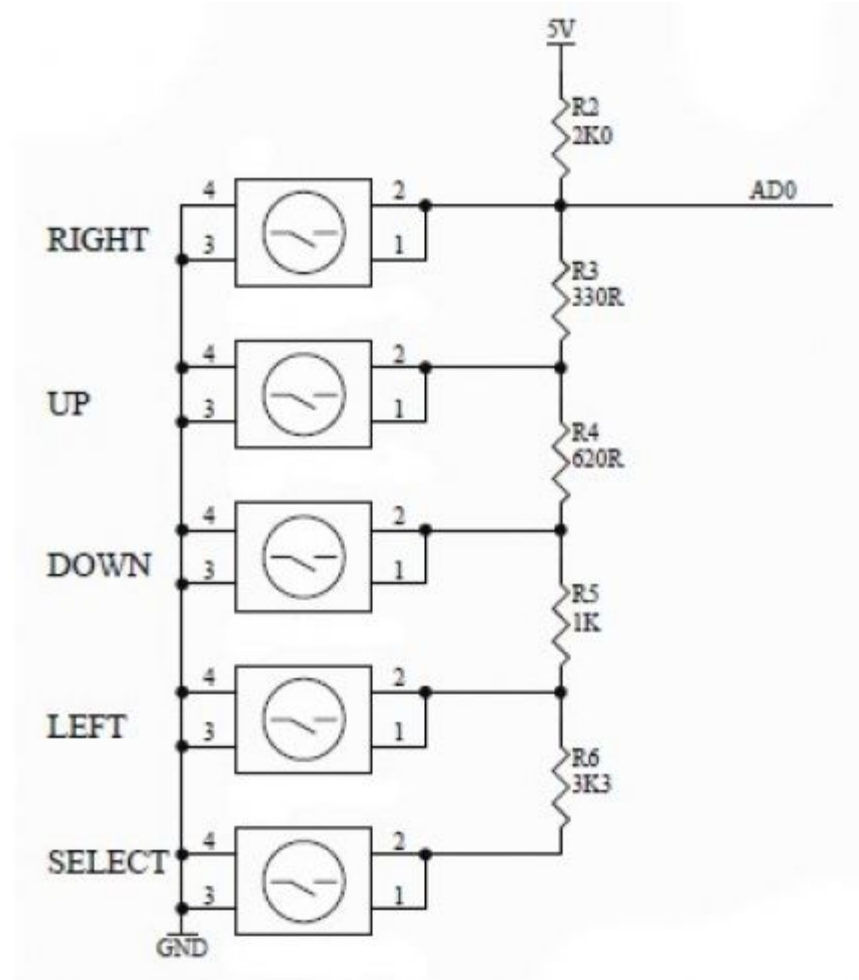




Usa uma associação de resistores
para determinar qual botão foi pressionado.

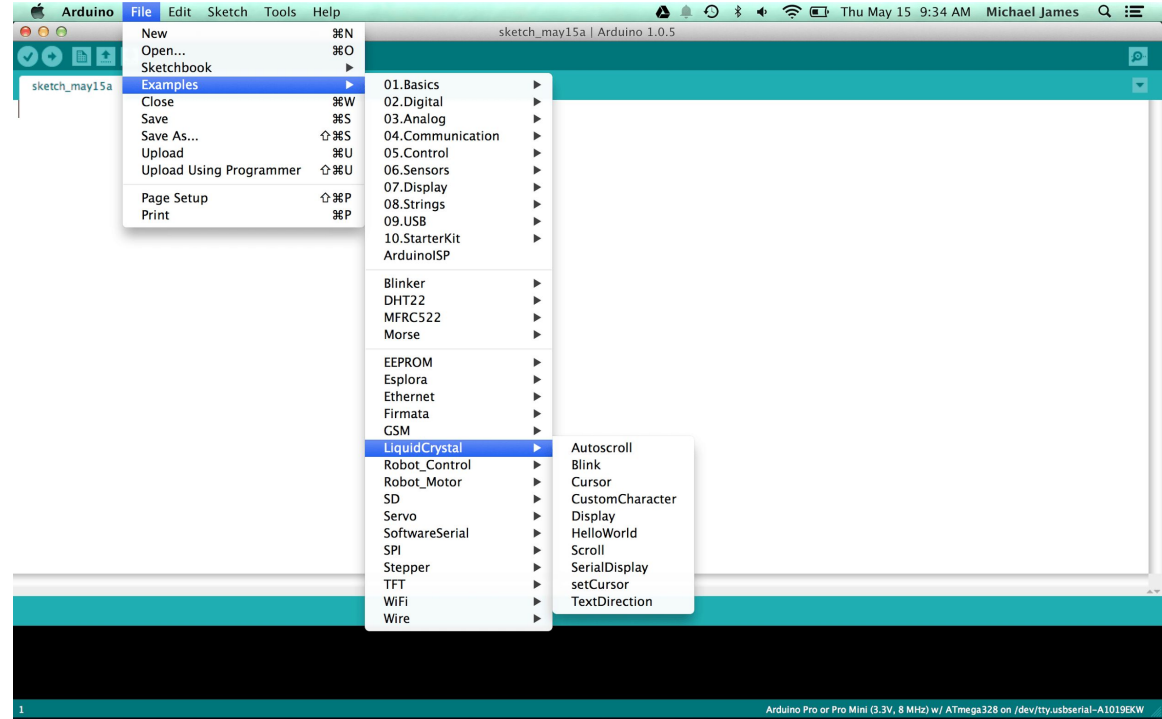
De forma a economizar portas digitais do Arduino

Usam uma única porta analógica, **A0**



Biblioteca

- LiquidCrystal.h
- Default na IDE Arduino



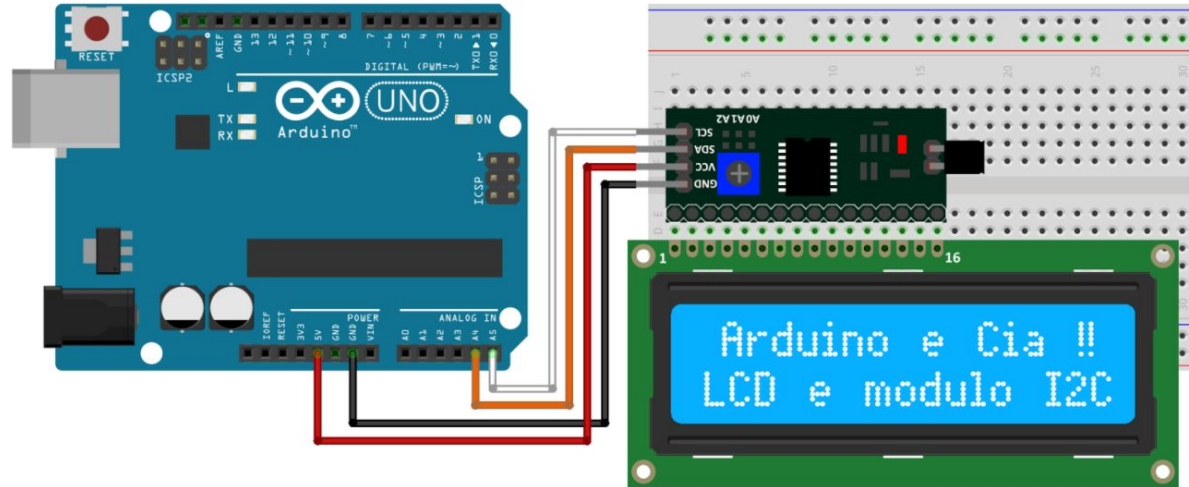
Função	Sintaxe no programa	Descrição
LiquidCrystal()	LiquidCrystal(rs, enable, d4, d5, d6, d7)	Indica como os pinos do display estão interligados na placa Arduino
begin()	lcd.begin(colunas, linhas)	Configura o modelo do display em nosso caso 16×2
clear()	lcd.clear()	Limpa o display e posiciona o cursor no canto superior esquerdo do display
home()	lcd.home()	Posiciona o cursor no canto superior esquerdo do display
setCursor()	lcd.setCursor(coluna, linha)	Posiciona o cursor na linha e coluna indicada no comando
write()	lcd.write(caractere)	Escreve um caractere no display
print()	lcd.print(dado)	Escreve um dado qualquer no display (texto ou número)
cursor()	lcd.cursor()	Exibe o cursor na tela do LCD no formato de um underline

noCursor()	lcd.noCursor()	Ocultar o cursor
blink()	lcd.blink()	Faz com que o cursor fique piscando na tela do Arduino
noBlink()	noBlink()	Faz com que o display pare de piscar
noDisplay()	lcd.noDisplay()	Desliga o display sem perder os dados exibidos
display()	lcd.display()	liga o display e restaura o texto escrito após um comando “noDisplay()”
scrollDisplayLeft()	lcd.scrollDisplayLeft()	Mover o conteúdo do display (texto+cursor) um espaço para a esquerda.
scrollDisplayRight()	lcd.scrollDisplayRight()	Mover o conteúdo do display (texto+cursor) um espaço para a direita.
autoscroll()	lcd.autoscroll()	Ativa a rolagem automática. Cada caractere empurra os anteriores um espaço. Se a direção do texto atual é da esquerda para a direita (o padrão) o visor desloca para a esquerda;

noAutoscroll()	lcd.noAutoscroll()	Desliga o modo automático autoscroll.
leftToRight()	lcd.leftToRight()	Define a direção para o texto ser escrito no LCD da esquerda para a direita, o padrão. Isto significa que os caracteres subsequentes escritos para a exibição vai da esquerda para a direita, mas não afeta o texto enviado anteriormente para o display.
rightToLeft()	lcd.rightToLeft()	Define a direção para o texto ser escrito no LCD da direita para a esquerda. Isto significa que os caracteres subsequentes escritos para a exibição vai da direita para a esquerda, mas não afeta o texto enviado anteriormente para o display.
createChar()	lcd.createChar()	Cria um caractere personalizado (glyph) para uso no LCD. Até oito caracteres de 5×8 pixels são suportadas (numeradas de 0 a 7). O aparecimento de cada caracter personalizada é especificada por uma matriz de oito bytes, uma para cada linha. Os cinco bits menos significativos de cada byte de determina os pixels daquela linha. Para exibir um caractere personalizado na tela, use o comando <code>lcd.write(byte(numero));</code>

Curiosidades

- LCD API 1.0 (<http://playground.arduino.cc/Code/LCDAPI>)
- LiquidCrystal_I2C.h



Demonstração

Adaptação do exercício 3A

- Utilizando DHT



Links:

<http://www.arduinoecia.com.br/2013/08/arduino-shield-lcd-16x2-com-keypad.html>

<http://blog.usinainfo.com.br/shield-lcd-16x2-lcd-keypad-shield-com-botoes-para-arduino/>

<https://playground.arduino.cc/Main/LiquidCrystal>

<http://blog.baudaeletronica.com.br/display-lcd-16x2-conhecendo-biblioteca-liquidcrystal/>

<https://programmingelectronics.com/liquidcrystal-library-functions-tour-part-1/>

<http://www.arduinoecia.com.br/2014/12/modulo-i2c-display-16x2-arduino.html>