

Placa de desenvolvimento Interface LCD para Arduino UNO Rev3 compatível (ID PR.FT9_5)

Manual

V 1.0 – Agosto 2023

Índice

1. Introdução.....	3
2. Especificações.....	3
3. Diagrama de conexões.....	4
4. Informações gerais.....	6
5. Programação.....	13
6. Diagrama esquemático.....	23

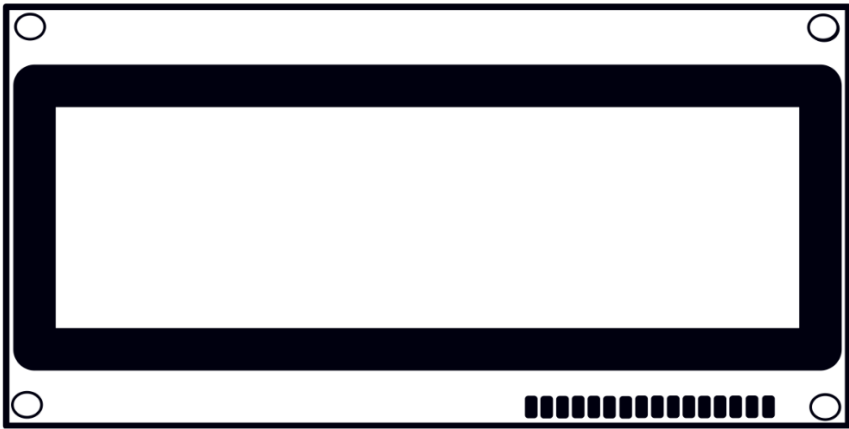
1. Introdução

Este manual descreve as especificações técnicas da placa de desenvolvimento interface LCD para Arduino UNO Rev3 compatível (ID PR.FT9_5). Inclui informação essencial sobre os seus componentes e exemplos de programação para facilitar a interação inicial. Os itens deste produto encontram-se atualizados, com as últimas versões estáveis.

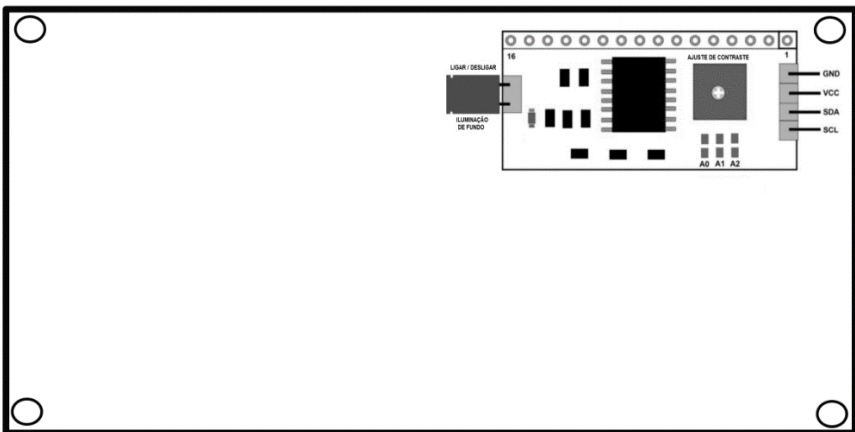
2. Especificações

- Display LCD com 20 caracteres por 4 linhas
- “Trimmer” para ajuste do contraste
- Interruptor para ligar/desligar iluminação de fundo
- Botão de Reset
- 3 botões configuráveis de gestão das entradas (3 entradas digitais e uma entrada analógica)
- Dimensões 68 x 53 mm.

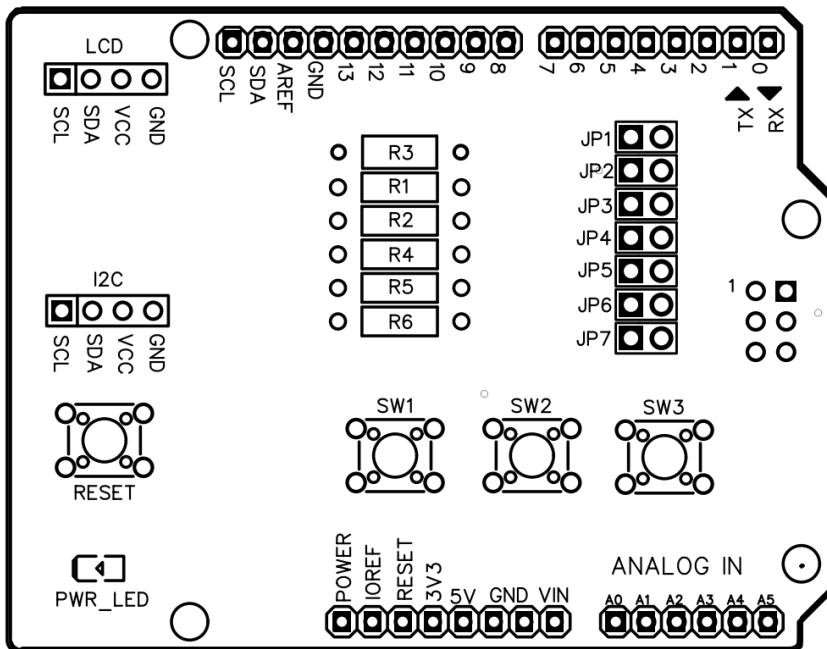
3. Diagrama de conexões



Display LCD 20x04 I2C – Vista frente



Display LCD 20x04 com interface I2C - Vista trás

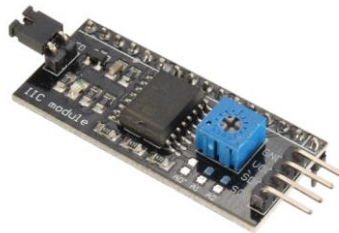


Placa de desenvolvimento Interface LCD para Arduino UNO Rev3 compatível

4. Informações gerais

Display LCD 20x04

O display LCD 20x04 com interface I2C tem no centro do módulo o circuito integrado PCF8574. Este CI de 8 bits converte os dados I2C do microcontrolador em dados paralelos exigidos pelo display LCD.



Módulo conversor de dados I2C (do microcontrolador) para dados paralelos (LCD)

Na parte lateral, o módulo I2C tem quatro pinos para ligação à placa (conector LCD) que suporta o protocolo de comunicação. Estes quatro pinos são: GND, VCC, SDA e SCL.



Pinos do barramento I2C do display LCD

O pino SDA (Serial Data) é o responsável pela transferência de dados. Por sua vez, o pino SCL (Serial Clock) é utilizado como temporizador.

O módulo I2C conta também com dois pinos para controle da luz de fundo (backlight) do visor do display, que se encontra ligados por um *jumper*. Assim, a luz de fundo estará sempre ligada. Se remover este *jumper* desligará a luz de fundo do display.



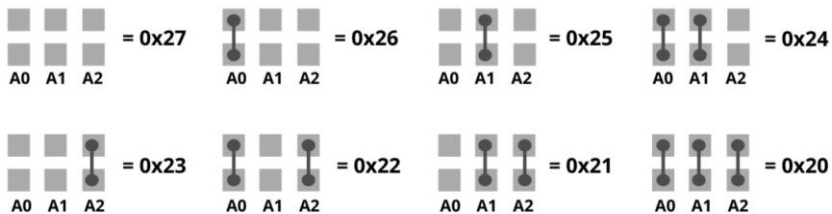
Controle da luz de fundo (backlight) do visor do display

Além disto, a placa conta com um “*Trimmer*” para ajuste fino do contraste do display e um barramento para configuração do endereço I2C.



Trimmer para ajuste do contraste do display LCD

Por defeito, o display LCD tem um endereço em formato hexadecimal configurado pelo fabricante do dispositivo (originalmente o endereço é o 0x27) mas poderá ser alterado conforme configuração dos contatos A0, A1 e A2



Configuração do endereço I2C do dispositivo display LCD

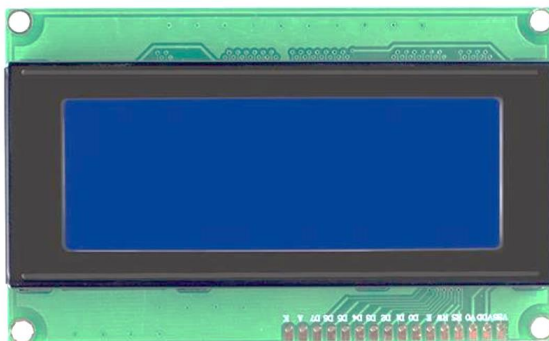


Imagem frontal do display LCD 20x04



Imagem traseira do display LCD 20x04

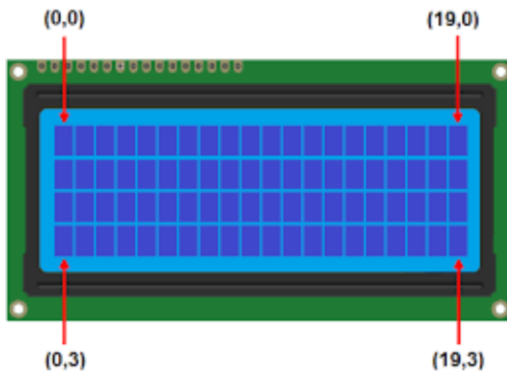


Imagem da matriz de caracteres (4 linhas e 20 colunas)

Placa de desenvolvimento Interface LCD para Arduino UNO

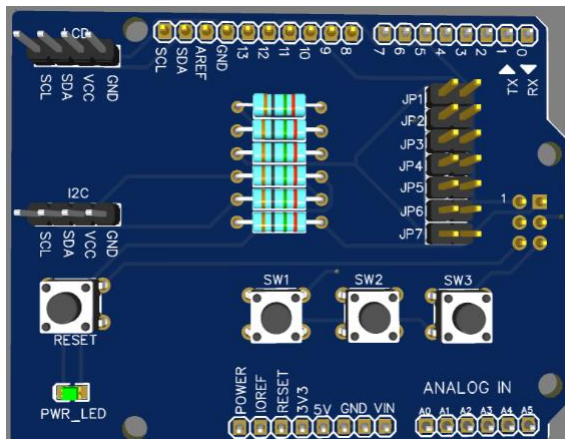


Imagem frontal da placa de desenvolvimento Interface LCD para Arduino UNO

A placa é tem um formato *shield* compatível com o Arduino e acessibilidade a todos os seus pinos através de *stackable header* fêmea.

Depois de ligada, exibirá um LED verde (PWR_LED):

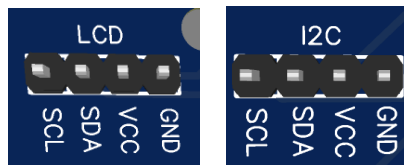


LED que assinala ligação da placa de Interface LCD para Arduino UNO

Para fazer interface com o display LCD 20x04 existem duas possibilidades com conetores (4 *pin header* macho) designados por LCD e I2C. No *pin header* LCD poderá conetar o display LCD 20x04 através do cabo Dupont de 4 pinos fêmea-fêmea.

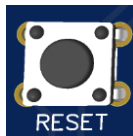


Cabo de Interface entre a placa e o display LCD 20x04



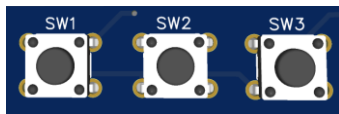
Interface LCD (para display LCD 20x04) e adicional I2C para ligação de outros dispositivos I2C

A placa dispõe de um botão de Reset para reiniciar o microcontrolador:



Botão RESET

Na parte central da placa encontram-se os três botões de pressão (SW1, SW2 e SW3) e sete *header jumpers* (JP1, JP2, JP3, JP4, JP5, JP6 e JP7) para configuração manual de algumas linhas pinos digitais (10, 11, e 12) ou analógico (A3) através desses botões.



3 botões configuráveis de gestão das entradas através de jumpers

5. Programação

Para descarregar o software Arduino IDE siga o link:

<https://www.arduino.cc/en/software>

Para comunicar entre a placa de desenvolvimento interface LCD e o Display LCD utilizamos o protocolo I2C (*Inter-Integrated Circuit*) e a biblioteca para Arduino Wire.h

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

O protocolo I2C necessita de dois pinos (SDA – dados, SCL – clock) para poder ligar até 127 dispositivos ao barramento de 7 bits, funcionando em hierarquia mestre-escravo. Assim, normalmente o microcontrolador Arduino é o mestre e todos os dispositivos e sensores ligados, os escravos. Porém, para evitar conflito de dados no barramento, o dispositivo mestre é quem realiza a coordenação de toda a comunicação, tendo a capacidade de enviar e solicitar informações aos escravos.

Cada dispositivo tem um endereço associado em formato hexadecimal. Esse endereço é dado pelo fabricante do dispositivo (no caso do LCD é o 0x27) mas pode ser alterado (consultando a documentação técnica que cada fabricante disponibiliza) no caso de haver repetições de endereços no barramento I2C. Diversos componentes e módulos foram desenvolvidos para este tipo de interface.

Mais informação sobre a biblioteca Wire.h no link:

<https://www.arduino.cc/reference/en/language/functions/communication/wire/>

Mais informação sobre a biblioteca LiquidCrystal_I2C.h no link:

https://github.com/johnrickman/LiquidCrystal_I2C

ou na biblioteca referência Arduino LiquidCrystal-I2C.h:

<https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/>

Exemplo de programa de teste do LCD:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4); //endereço I2C do display 0x27
int l = 0;

void setup() {
  lcd.init();          //Inicia LCD
  lcd.backlight();    //Activa backlight
}

void loop() {
  // Limpa LCD
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(".....");
  lcd.setCursor(0,1);
  lcd.print("  LCD LIGADO  ");
  lcd.setCursor(0,2);
  lcd.print("  INICIA TESTE  ");
  lcd.setCursor(0,3);
  lcd.print(".....");
  delay(2000);
```

```
lcd.clear();  
lcd.setCursor(0,0);
```

```
lcd.noDisplay();  
delay(500);  
// liga display:  
lcd.display();  
delay(500);
```

```
lcd.clear();  
while(i<20)  
{  
  lcd.setCursor(1,0);  
  lcd.print("X");  
  i = i + 1;  
  delay(100);  
}  
i = 0;
```

```
while(i<20)  
{  
  lcd.setCursor(i,1);  
  lcd.print("X");  
  i = i + 1;  
  delay(100);  
}  
i = 0;
```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("ABCDEFGHILMNOPQRST");  
lcd.setCursor(0,1);  
lcd.print("UVWXYZ");  
lcd.setCursor(0,2);  
lcd.print("abcdefghijklmnpqrst");  
lcd.setCursor(0,3);  
lcd.print("uvwxyz");  
delay(2000);
```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print(" 1234567890 ");  
lcd.setCursor(0,1);  
lcd.print(" 0987654321 ");  
delay(2000);
```

```
lcd.setCursor(0,0);  
lcd.print(" !$%&/()=?! ");  
lcd.setCursor(0,1);  
lcd.print(" |@#~!$%&/% ");  
delay(2000);
```

```
lcd.clear();  
lcd.setCursor(0,1);  
lcd.print(" TESTE OK! ");  
  
delay(3000);
```



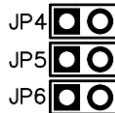
```
}
```

A placa dispõe de três botões para desenvolver as aplicações e experiências. Existem duas formas de interagir com estes botões através de *jumper*s.



Exemplo de *jumper* aberto e fechado.

A primeira forma faz uso de três linhas digitais do Arduino, designadamente nos pinos 10, 11 e 12. Neste caso, é necessário colocar os *jumper*s nas posições JP4, JP5 e JP6 da placa.



Configuração digital dos botões

Para usar os botões com as entradas digitais, ative as resistências *pull-up* internas através do código de programação do microcontrolador. Faça isso para cada botão com o seguinte código exemplo:

```
#define P1 10
```

```
#define P2 11
```

```
#define P3 12
```

```
pinMode(P1, INPUT_PULLUP);
```

```
pinMode(P2, INPUT_PULLUP);
```

```
pinMode(P3, INPUT_PULLUP);
```

Os botões funcionarão numa lógica negativa, ou seja, o valor em repouso é 1. Quando o botão é pressionado o valor passa a 0. O exemplo de código seguinte permite o teste dos botões SW1,SW2 e SW3 em modo digital.

Exemplo 1 – Programa de teste de botões em modo digital:

```
/*  
  Botões ligados modo DIGITAL (JP4, JP5 e JP6 devem ter jumper)  
*/  
  
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27,20,4); //endereço I2C do display 0x27  
#define P1 10  
#define P2 11  
#define P3 12  
int i = 0;  
  
void setup() {  
  lcd.init();      //incia LCD  
  lcd.backlight(); //ativa backlight  
  
  // ativa modo entrada dos pinos  
  pinMode(P1, INPUT_PULLUP);  
  pinMode(P2, INPUT_PULLUP);  
  pinMode(P3, INPUT_PULLUP);
```

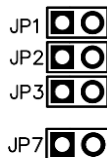
```

lcd.clear();
//mostra mensagem inicial
lcd.print("Bem-vindo!");
}

void loop() {
//cursos linha 1
lcd.setCursor(0, 1);
// mostra botões pressionados
if ( digitalRead(P1)==0 ) lcd.print("SW1 pressionado.");
if ( digitalRead(P2)==0 ) lcd.print("SW2 pressionado.");
if ( digitalRead(P3)==0 ) lcd.print("SW3 pressionado.");
}

```

A segunda forma de utilização dos botões da placa envolve uma linha analógica, neste caso a do pino A3. Neste caso, as linhas digitais permanecem livres para outros fins. Para ativar este modo é necessário colocar os *jumpers* nas posições JP1, JP2, JP3 e JP7.



Configuração analógica dos botões

Poderá observar no diagrama do circuito formado pelas resistências R4, R5 e R6 formam um divisor de tensão.

Quando nenhum dos botões é pressionado, o pino A3 tem uma tensão de +5 V sem circulação de corrente elétrica. Se pressionarmos o botão SW1, o pino A3 será ligado a GND e a tensão passa a 0V. Se pressionarmos o botão SW2, o nó entre as resistências R5 e R6 fica ligado a GND, portanto, o pino A3 terá uma tensão de aproximadamente de 1,6 V. Finalmente, quando se pressiona o botão SW3, a resistência R6 ficará ligada a GND e no pino A3 podemos medir uma tensão de 3,3 V.

Resumidamente, para cada botão, o conversor analógico digital (ADC) do Arduino terá diferentes tensões. No código, a função **analogRead(Pino_3)** irá ler um valor de tensão diferente de acordo com o botão que está a ser pressionado.

Botão pressionado	Tensão em A3	Valor Digital
SW1	0 V	0
SW2	1,6 V	328
SW3	3,3 V	676
Nenhum	5 V	1023

Tabela de conversão entre tensão analógica e valor digital do ADC de 10 bits

O exemplo de código seguinte permite o teste dos botões SW1, SW2 e SW3 em modo analógico.

Exemplo 2 – Programa de teste de botões em modo analógico:

```
/*
  Botões ligados Modo ANALÓGICO ( JP1, JP2, JP3 e JP7 devem ter jumper)
*/

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,20,4); //endereço I2C do display 0x27

#define Pino_3 3 //pino analogico

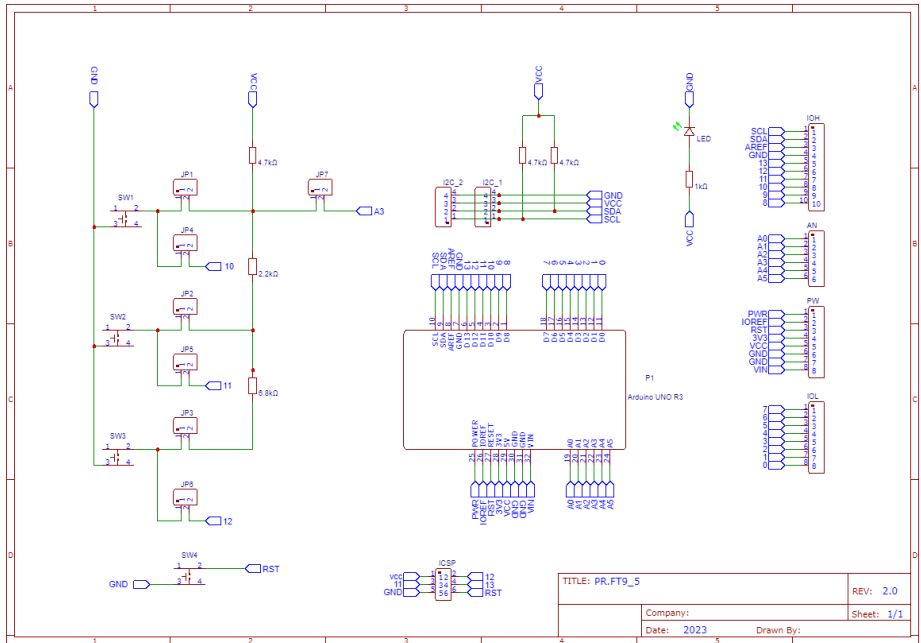
void setup() {
  lcd.init();          //inicia LCD
  lcd.backlight();    //ativa backlight
  // mensagem inicial
  lcd.clear();
  //mostra mensagem inicial
  lcd.print("Bem-vindo!");
}

void loop() {
  //posiciona cursor
  lcd.setCursor(0, 1);
  // mostra
  int Valor = analogRead(Pino_3);

  if ( Valor < 164 )
    lcd.print("SW1 pressionado.");
}
```

```
else if ( Valor < 502 )  
lcd.print("SW2 pressionado.");  
else if ( Valor < 850 )  
lcd.print("SW3 pressionado.");  
else  
lcd.print("Pressione botao!");  
}
```

6. Diagrama esquemático



Esquema elétrico do circuito da placa de interface

CE Declaration of Conformity

Company: Shenzhen Scope Corporation, Ltd.

Address: 12/13 Floors, C2 Building, I Park, No. 1001, College Road, Nanshan, Shenzhen, Guangdong, China

Product Name: LCD Interface Development Board

Product Model: XUGU-E012 (AB.AZ00121)

Directives and Standard applicable:

Our samples match EMC directive 2014/30/EU and ROHS EU2011/65+AMD EU 2015/863

EMC Test Standards :EN 55032:2015+A11:2020+A1:2020

EN 55035:2017+A11:2020

RoHS Test Standards : EN 62321-5:2014,

EN62321-4:2014+A1:2017,

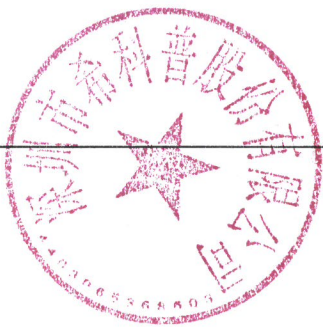
EN 62321-7-1:2015

EN 62321-7-2:2017,

IEC 62321-6:2015,

IEC 62321-8:2017

Signature: _____



A handwritten signature in black ink, consisting of stylized, cursive characters, positioned over a horizontal line that extends from the signature area to the date field.

Date: ___ Oct. 11, 2023 ___