

Workshop Arduino

Prof. Simão





Números



- Conforme a aplicação, talvez seja necessário trabalhar com números em bases diferentes da decimal
- O pequeno programa a seguir exibe um conjunto de números em diferentes bases de numeração

```
lint numero;
 3void setup() {
 4 Serial.begin(9600);
 5
 6 }
 8 void loop() {
 9
10 for (numero = 0; numero < 33; numero++) {
11
    Serial.print(numero, DEC);
12
    Serial.print("\t");
13
    Serial.print(numero, BIN);
14
    Serial.print("\t");
    Serial.println(numero, HEX);
15
16
    delay(500);
17 }
18 while(1){
19
20 }
21 }
```



```
Para ver o
resultado da
execução, clique
Ferramentas /
Monitor Serial – e,
SE precisar, ajuste
a velocidade de
comunicação para
9600 bps
```



EXPERIMENTOS

Monitor serial



- O monitor serial é um programa de comunicação serial, capaz de receber e enviar comandos.
- Vamos experimentar

TI Aplicada

000010 1000011 0000

000001010000101000000000

0100000 1100110

4 leds, 4 resistores



fritzing



```
void loop(){
  if (Serial.available())
    switch(Serial.read())
    case 't':
      digitalWrite(ledAmarelo, 0);
      digitalWrite(ledVermelho, 0);
      digitalWrite(ledVerde, 0);
      digitalWrite(ledAzul, 0);
      Serial.println("Todos apagados");
      break:
    case 'T':
      digitalWrite(ledAmarelo, 1);
      digitalWrite(ledVermelho, 1);
      digitalWrite(ledVerde, 1);
      digitalWrite(ledAzul, 1);
      Serial.println("Todos acesos");
      break;
```



```
Use o monitor
serial para
interagir com seu
circuito (os
comandos
poderiam também
vir via Bluetooth
ou outra conexão).
```

Tl Aplicada

simao@ufpr.br - 2022







- Baseado em seus experimentos anteriores, crie um semáforo contendo 6 leds, nas cores usuais de um semáforo.
 - Quando o led vermelho de um estiver aceso o led verde o outro estará aceso
 - Uma variação é colocar um temporizador para que os pedestres atravessem, antes que o segundo sinal abra, após o primeiro fechar

O que você precisará?

- Arduino configurado na IDE
- Uma placa de experimentação e fios
- 6 leds, sendo 2 vermelhos, 2 verdes e 2 amarelos
- 6 resistores para limitar corrente/ tensão, conforme já visto



Sequência



- Desligue o cabo USB
- Ligue os componentes na placa
- Conecte os fios à placa do Arduino; anote as portas digitais utilizadas para cada led
- Elabore o código, lembrando de declarar as portas como saída no setup() e depois escrevendo níveis LOW ou HIGH em cada porta dentro do loop()





```
23 void loop() {
24
    // semáforos 1 e 2 vermelhos, pedestre atravessando
25
    digitalWrite(vm1, HIGH);
26
    digitalWrite(vm2, HIGH);
27
    delay(1000);
28
    //abre o semáfor 1, semáforo 2 fechado
29
    digitalWrite(vm1, LOW);
30
    digitalWrite(vd1, HIGH);
31
    delay(1000);
32
    //prepara para fechar o semáforo 1
33
    digitalWrite(vd1, LOW);
34
    digitalWrite(am1, HIGH);
35
    delay(1000);
36
    digitalWrite(am1, LOW);
37
    //fecha o semáforo 1 e abre o 2
38
    digitalWrite(vm1, HIGH);
39
    digitalWrite(vm2, LOW);
40
    digitalWrite(vd2, HIGH);
41
    delay(1000);
42
    //prepara para fechar o semáforo 2
43
    digitalWrite(vd2, LOW);
44
    digitalWrite(am2, HIGH);
45
    delay(1000);
46
    digitalWrite(am2, LOW);
47 }
```

simao@ufpr.br - 2022



Como ficou sua ligação?



Documente!

Use o Fritzing!

TI Aplicada

					Untitled Sk	etch.fzz - Fri	tzing - [V	isão Protob	oard]				_	
<u>\</u> rquivo	<u>E</u> ditar	<u>C</u> omponente	<u>V</u> isāo	Janela	<u>R</u> oteamen	to <u>Aj</u> uda								
f Welcome		📖 Protoboard			emático			< > Cód	igo		Con	nponentes		₽
									0		Q	Core Parts		
											CORE	Básico		
											MINE	բու 🗛		MM
::											œ		R 🔜	,0
-	un.	15	8	52	26 S	40	54	5	12	8	4	Entrada		
													<u> </u>	
			••••								seeed			
	• • • • •	• • • • • • • • • •	• • • • •			• • • • • • •	• • • • • •			• • • •	(inters		- 0	
												(n) 🔪 (•	
			••••			•••••	•••••				Pro	priedades		c
< • • • • • • • • • • • • • • • • • • •	• • • • • •	5 15 • • • • • • • • • •	* * * * * R	N .	8 8	• • • • • • • • 9	4 • • • • • •	8	8 8	8	Bre	adboard1		
										::	Bread	: iboard1	v. 4	
C. 91	0										Place	ment		
Trit	zາກດ]									locati	on -0.052 +	0.000 *	in
											rotati	on 0.0 +	degrees	
	(A	2			Não eviete	na li ana ña a v				×00		Bloquea	do	
	Ċ,				Nao existe	ni ligações p	ara rotea				Prop	riedades		
dicnota	Girar	Inverter								compthat	famíl	a breadboard	đ	
onto							(x,y)=(-0.170, -0	.517) in 137	% O —	\oplus	tama	nho full+		Ŧ

Na aba de componentes digite UNO. Selecione a placa e arraste para o desenho (use CTRL + '-') para ajustar o tamanho na tela.



Clique sobre o pino 5V na placa do Arduino e arraste até a barra positiva. O Fritzing vai indicar a ligação com um fio.



O programa utiliza a última cor utilizada como padrão. Portanto, no seu caso a cor do fio poderá não ser azul como a do exemplo ao lado.

TI Aplicada





* no caso de a cor do fio não ser esta



Na aba de componentes, digite LED, selecione-o e arraste-o para o *protoboard*.



* não se preocupe com a cor, dá para mudar depois

0000010

0100000



Coloque mais um LED no *protoboard*. Depois, na aba de propriedades, ajuste a cor do LED para amarelo.

		Untitled Ske	tch.fzz* - Fritzing - [Vi	ão Protoboard]				- 0	×
<u>A</u> rquivo <u>E</u> ditar	<u>C</u> omponente <u>V</u> isão	Janela <u>R</u> oteament	to <u>Aj</u> uda						
f Welcome	📖 Protoboard	Esquemático	■ PCB	< > Código		Compon	ientes		₽×
									۹ ∗≣
						CORE	i		
	RESET ICSP2					MINE			
	TX =	Arduino"				⊡⊙			
		451				\$		be B25	1
	1 (B) (B)						528	Galileo	
		ANALOG IN				seeed		@∞	
	•		a 🗐					Galileo' 📒	
	- 0 2 	8 8 2	я н	8 8		Propried	lades		₽×
						família	led		
						corrente			-
						cor	Yellow (595nm)		Ŧ
	= • • • • • • • • • • • • • • • • • • •				n 	pacote	0805 [SMD]		Ŧ
fritzing		2 R R R	10 10 10 10 10 10 10 10 10 10 10 10 10 1	8 8		número do component	te		
					-	Marcadore	25		
						led, red led,	, indicator, fritzing	core, smd	
r (>		Rotea	amento completo		⊘器	Conexões			
Adicnota Girar	Inverter				Complhar	con.			
						nome			
			(x,y)=(7.622, 0.8	67) in 100 % 🕒 🖳	Ð	tipo			

Na aba de componentes, procure resistor e arraste um para o *protoboard*. O valor pode ser ajustado depois.





1 100001 0100

Ajuste o valor do resistor na aba de propriedades (para o valor que você está usando). As cores mudam automaticamente.





0100000

1010000 (



- Seu desenho deverá ter 6 leds (2 vermelhos, 2 verdes e 2 amarelos), e seis resistores limitadores de corrente.
- Você terá usado 6 portas do Arduino. Ligue-os de acordo com a numeração que você usou em seu código, para a que a documentação fique correta.



Vamos trabalhar com o sensor de ultrassom

TI Aplicada





simao@ufpr.br - 2022

Medidor de distância



 Baseado em seus experimentos anteriores, crie um medidor de distância

O que você precisará?

- Arduino configurado na IDE
- Uma placa de experimentação e fios
- Um emissor receptor de ultrassom SR-04



Sequência



- Desligue o cabo USB
- Ligue os componentes na placa
- Conecte os fios à placa do Arduino; anote as portas utilizadas
- Elabore o código

simao@ufpr.br - 2022



Ligação



O pino VCC do SR-04 é ligado ao 5V da placa do Arduino.

O pino GND do SR-04 é ligado ao GND da placa do Arduino.

O pino TRIG do SR-04 é ligado na digital 4 da placa do Arduino.

O pino ECHO do SR-04 é ligado na digital 3 da placa do Arduino.

(Se usar outras portas troque no código exemplo a seguir)



Quem faz todo o trabalho é a biblioteca incluída no código, "Ultrasonic.h".

Código

SE ocorrer um erro de compilação indicando que a biblioteca não está instalada, veja o slide seguinte.

נוסוססס סוסססס וווסטס סוסססס ווסוווו סוסססס ווסטוס סוסססט וסוססוו סוסססט ווסוסט סוסססס ווטווטו סוסססס ווסטסס ו אוטווווו

'l Aplicada

SE for necessário incluir a biblioteca



<u>Sketch</u> Ferramen <u>t</u> as Ajuda			
Verificar/Compilar	Ctrl+R		
Carregar	Ctrl+U		
Carregar usando programador	Ctrl+Shift+U		
p Exportar Binário compilado	Ctrl+Alt+S		
^S Mostrar a página do Sketch	Ctrl+K		
Incluir Biblioteca	•	Δ	
l Adicionar Arquivo		Gerenciar Bibliotecas	Ctrl+Shift+I
in AC electrical system		Adicionar biblioteca . ZIP	

Além da óbvia pesquisa na internet, você poderá encontrar bibliotecas em: https://www.arduino.cc/reference/en/libraries/



Código



Quem faz todo o trabalho é a biblioteca incluída no código, "Ultrasonic.h".

O valor da distância está sendo enviado à saída Serial.

Para vê-lo, vamos ligar o Monitor Serial.

Basta clicar no ícone à esquerda da tela:



simao@ufpr.br - 2022

Ferramentas / Monitor serial



	/dev/ttyUSB0	
		Enviar
Distancia em CM: O Distancia emIniciando Distancia em CM: O Distancia em CM: 4 Distancia em CM: 5 Distancia em CM: 6 Distancia em CM: 6 Distancia em CM: 11 Distancia em CM: 11 Distancia em CM: 11		×
Auto-rolagem	Nenhum final-de-linha	9600 velocidade 🔹 Deleta a saida

Melhorando



medeDistanciaCMPol 1 #include <Ultrasonic.h> 3 #define trig 4 4 #define echo 3 6 Ultrasonic ultrasonic(trig, echo); 8 void setup() 9 { 10 11 12 13 14 } 15

30 }

	/dev/ttyUSB0	
		Enviar
Distancia em Pol: 963 Distancia em CM: 2458 Distancia em Pol: 963 Distancia em Pol: 7 Distancia em Pol: 1 Distancia em CM: 7 Distancia em Pol: 0 Distancia em CM: 7 Distancia em CM: 7 Distancia em CM: 6 Distancia em Pol: 2 Distancia em Pol: 2 Distancia em Pol: 2 Distancia em CM: 6 Distancia em CM: 6 Distancia em Pol: 2 Distancia em CM: 6 Distancia em Pol: 2 Distancia em CM: 2459 Distancia em CM: 2459		
Auto-rolagem	Nenhum final-de-linha 💌 9600 velocida	ade 🔹 Deleta a saida

Sem biblioteca



- Se você tiver informações a respeito do módulo que está utilizando (ou se você desenvolveu o módulo...), poderá controlar suas características diretamente, sem utilizar uma biblioteca.
 - Em geral as características do módulo ou de seu circuito integrado de controle estão disponíveis na forma de *datasheets* (folhas de dados)

Ultrassom





```
3
4 void setup()
5 {
6 Serial.begin(9600);
7 pinMode(pinoTrigger, OUTPUT);
8 pinMode(pinoEcho, INPUT);
9 }
```

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion .You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.



Ultrassom



11 float mede() digitalWrite(pinoTrigger,LOW); delayMicroseconds(3); digitalWrite(pinoTrigger,HIGH); delayMicroseconds(10); digitalWrite(pinoTrigger,LOW); float tempoUs = pulseIn(pinoEcho, HIGH); return (tempoUs / 58); 22 void loop() float distancia = mede(); Serial.print("Distancia medida: "); Serial.print(distancia); Serial.println(" centimetros"); 28 delay(500); 29 }

The Timing diagram is shown below. You only need to supply a short 10uS pulse to the trigger input to start the ranging, and then the module will send out an 8 cycle burst of ultrasound at 40 kHz and raise its echo. The Echo is a distance object that is pulse width and the range in proportion . You can calculate the range through the time interval between sending trigger signal and receiving echo signal. Formula: uS / 58 = centimeters or uS / 148 =inch; or: the range = high level time * velocity (340M/S) / 2; we suggest to use over 60ms measurement cycle, in order to prevent trigger signal to the echo signal.









O que você precisará?

- Arduino configurado na IDE
- Uma placa de experimentação e fios
- Um sensor de temperatura e umidade DHT-11



simao@ufpr.br - 2022



Ligação



O pino VCC do DHT11 é o pino 1, e é ligado ao 5V da placa do Arduino.

O pino GND do DHT11 é o pino 4, e é ligado ao GND da placa do Arduino.

O pino de saída do DHT11 é ligado na entrada analógica A1 da placa do Arduino.



1 = VCC 2 = Saída3 = sem uso 4 = GND

1100001 0100001





Sequência



- Desligue o cabo USB
- Ligue os componentes na placa
- Conecte os fios à placa do Arduino; anote as portas utilizadas
- Elabore o código

Tl Aplicada

tempUmidadeDHT11 1 #include "DHT.h" 3 #define sensor Al 4 #define tipo DHT11 6 DHT dht(sensor, tipo); 8 void setup() 9 { Serial.begin(9600); 10 Serial.println("Iniciando...\n\n"); 11 12 dht.begin(); 13 } 14 15 void loop() 16 { 17 18 float temperatura = dht.readTemperature(); 19 float umidade = dht.readHumidity(); 20 21 if (isnan(temperatura) || isnan(umidade)) 22 23 Serial.println("Comunicação falhou!\n"); 24 } 25 else 26 { 27 Serial.print("Temperatura = "); Serial.print(temperatura); 28 Serial.print(" C, e umidade = "); 29 Serial.print(umidade); 30 31 Serial.println(" %"); 32 33 delay(2000); 34

Código



Quem faz todo o trabalho agora é a biblioteca incluída no código, "DHT.h".

Para ver o resultado, vamos ligar o Monitor Serial.

Basta clicar no ícone à esquerda da tela:



35 }

Ferramentas / Monitor serial



	/dev/ttyUSB0	- • •
		Enviar
Iniciando		
Temperatura = 26.50 C, e umidade = 58.00 % Temperatura = 26.40 C, e umidade = 58.00 % Temperatura = 26.40 C, e umidade = 58.00 %		
🗹 Auto-rolagem	Nenhum final-de-linha 🔻 9600 velocidade 💌 Del	eta a saida

Sem biblioteca



- Também é possível ligar este sensor sem usar uma biblioteca.
 - Pesquise...

TI Aplicada





simao@ufpr.br - 2022

O que você precisará?

- Arduino configurado na IDE
- Uma placa de experimentação e fios
- Um sensor de som KY-037



Ligação



O pino A0 do KY-037 é ligado na entrada analógica A0 da placa do Arduino.

O pino G do KY-037 é ligado ao GND da placa do Arduino.

O pino + do KY-037 é ligado ao 5V da placa do Arduino.

Sequência



- Desligue o cabo USB
- Ligue os componentes na placa
- Conecte os fios à placa do Arduino; anote as portas utilizadas
- Elabore o código

Código



			s	omKY037 /	Arduino 1.8.
Агс	quivo	<u>E</u> dita	r <u>S</u> ketch	Ferramen <u>t</u> a	as Ajuda
0	0				
s	somKY	037			
1 2 3	void	setup(() {		
4	Seria	l .begi	.n (9600) ;		
6	}				
/ 8 9	void	loop()	{		
10 11	Seria	l.prin	itln(analo	gRead(AO));	
12	delay	(100);			
14	}				

Para ver o resultado, vamos ligar o Plotter Serial.

Ferramentas / Plotter Serial

0100000

1101001

0000010

0000010

Ferramentas / Plotter Serial



Display LCD





simao@ufpr.br - 2022

Pino	Identificação	Função
1	VSS	Ligação com o GND (0Volts)
2	VDD	Ligação com o VCC (+5Volts)
3	V0	Tensão de controle do contraste do LCD
4	RS	Seletor de registro (register select), utilizado para controle do display
5	RW	Seletor leitura/ escrita (<i>read/ write</i>), utilizado para controle do <i>display</i> . No código, este pino é visto como opcional em algumas aplicações, de foma que pode aparecer nos circuitos conectado ao GND, sem conexão ao Arduino.
6	E	Seletor de habilitação (enable), utilizado para controle do display
7	D0	Bit menos significativo da palavra de dados, data 0
8	D1	Bit 1 da palavra de dados
9	D2	Bit 2 da palavra de dados
10	D3	Bit 3 da palavra de dados
11	D4	Bit 4 da palavra de dados
12	D5	Bit 5 da palavra de dados
13	D6	Bit 6 da palavra de dados
14	D7	Bit mais significativo da palavra de dados, data 7



Nem todos os *displays* possuem um LED de luz de fundo (*backlight*). Portanto, podem não possuir os dois pinos que seguem (o pino 15 e o pino 16 – neste caso tem somente os 14 pinos anteriores), relativos à conexão do LED.

A propósito, se você não conectar o LED de *backlight*, o *display* funcionará normalmente, só terá menos luminosidade e, em algumas aplicações, em função da iluminação do ambeinte, um contraste mais difícil de perceber.

15	А	Ânodo do LED de <i>backlight</i> , ligar ao +5V
16	К	Cátodo do LED de <i>backlight</i> ; ligar o GND (0V) por meio de resistor limitador de corrente (pode acontecer de o resistor estar incluído no módulo, mas é sempre bom prevenir o resistor também pode ser ligado no pino 15, ao invés de no 16, é indiferente).



simao@ufpr.br - 2022

fritzing

Display LCD



O display ao lado segue a nomenclatura e numeração mostradas na tabela acima. Às vezes, os identificadores vem impressos:



Em geral a correspondência entre o sinal/ identificador e o número do pino não muda, PORÉM, pode haver mudança na seguência de pinos:



Um exemplo de declaração no IDE do Arduino:

```
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
```

Quer dizer que a ligação Arduino x Display, será: o pino 12 do Arduino no pino 4 do display; o pino 11 do Arduino no pino 6 do display; o pino 5 do Arduino no pino 11 do display; ...

simao@ufpr.br - 2022

Tl Aplicada



fritzing



//Adaptado de: <u>https://docs.arduino.cc/learn/electronics/lcd-displays</u> //A declaração a seguir informa o compilador que iremos utilizar a biblioteca 'LiquidCrystal' #include <LiguidCrystal.h> /* * Os termos, usados nas constantes a seguir, rs, en, d4, d5, d6 e d7 referem-se às conexões de controle do display. Cada display pode tê-los em posições diferentes * os números utilizados são os pinos do Arduino que você vai utilizar, e podem ser modificados se você desejar */ **const int** rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2; /* * A declaração a seguir cria um objeto 'lcd', baseado na biblioteca LiquidCrystal, e informa, na sequência, os pinos de controle * As formas de inicialiação / controle, são: * LiquidCrystal(rs, enable, d4, d5, d6, d7) * LiquidCrystal(rs, rw, enable, d4, d5, d6, d7) * LiquidCrystal(rs, enable, d0, d1, d2, d3, d4, d5, d6, d7) * LiquidCrystal(rs, rw, enable, d0, d1, d2, d3, d4, d5, d6, d7) * Cada uma destas formas está ligada a diferentes conexões físicas e diferentes possibilidades de uso do display */ LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

```
TI Aplicada
                const int colunas = 16, linhas = 2;
                /*
                * Para desenhar caracteres especiais:
                    https://maxpromer.github.io/LCD-Character-Creator/
                 *
                    https://omerk.github.io/lcdchargen/
                 *
                 */
                byte smiley[8] = {
                 B01110,
                 B10111,
                 B11110,
                 B11111,
                 B11000,
                 B11111,
                 B11010,
                 B10010,
                };
                void setup()
                  lcd.begin(colunas, linhas);
                  lcd.createChar(0, smiley);
```



```
void loop()
  lcd.clear();
  lcd.display();
  lcd.setCursor(4, 0);
  lcd.print("CyberRex");
  lcd.setCursor(0, 1);
  lcd.print("E.M. Omar Sabbag");
  delay(1000);
  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(500);
  lcd.noDisplay();
  delay(500);
  lcd.display();
  delay(1000);
```

```
lcd.clear();
lcd.home();
for (int pos = 0; pos<16; pos++)</pre>
ł
   lcd.setCursor(pos, 0);
   lcd.write(byte(0));
   delay(100);
}
for (int pos = 15; pos>0; pos--)
   lcd.setCursor(pos, 1);
   lcd.write(byte(0));
   delay(100);
for (int pos = 0; pos<16; pos++)</pre>
   lcd.clear();
   lcd.setCursor(pos, 0);
   lcd.write(byte(0));
   delay(150);
}
```





```
for (int pos = 15; pos>0; pos--)
{
    lcd.clear();
    lcd.setCursor(pos, 1);
    lcd.write(byte(0));
    delay(150);
  }
}
//Depois de testar este, vejam este: <u>https://create.arduino.cc/projecthub/aqzuonyt/arduino-dino-</u>
game-using-lcd-663aeb
```

Este código para controle do *display* está disponível em: https://tiaplicada.ufpr.br/wp-content/uploads/2022/10/displaylcd.zip

1010000



O que você aprendeu, que será necessário para os próximos passos:

 Utilizar bibliotecas de funções
 Pesquisar formas de controlar os módulos sem usar bibliotecas Tl Aplicada



Parabéns!

