

Cronômetro baseado em microcontrolador

Introdução

O presente relatório tem como objetivo apresentar o desenvolvimento de um cronômetro digital baseado em microcontrolador, descrevendo detalhadamente o problema a ser automatizado, os componentes utilizados, o diagrama de blocos do sistema, a tabela de entradas e saídas, e a projeção do diagrama de fluxo do programa.

Problema a ser automatizado

O projeto tem como finalidade a implementação de um cronômetro digital capaz de realizar a contagem de tempo em minutos e segundos, utilizando quatro displays de sete segmentos, três botoeiras e um diodo emissor de luz (LED) como iluminação de fundo. O sistema deve permitir iniciar e pausar a contagem, resetar o tempo e controlar o acionamento da luz de fundo.

Ao ser energizado, o cronômetro deve iniciar no estado 00:00. A contagem é exibida nos quatro displays, sendo dois dígitos destinados aos minutos e dois aos segundos. O sistema deve operar até o valor máximo de 99:59, retornando automaticamente a 00:00 ao ultrapassar esse limite.

As funcionalidades são definidas conforme segue:

- **Botoeira START/PAUSE:** inicia e pausa a contagem;
- **Botoeira RESET:** zera o cronômetro (00:00);
- **Botoeira BACKLIGHT:** alterna o estado do LED de fundo entre ligado e desligado.

Componentes e dispositivos a serem utilizados

Componente	Quantidade	Função	Preço
Microcontrolador (Arduino Uno)	1	Controle central do sistema	R\$ 56,90
Display LCD 16x2	1	Exibição de minutos e segundos	R\$ 18,00
Botoeiras (push buttons)	3	Controle de funções (start/pause, reset e backlight)	R\$ 3,00 – R\$ 9,00
LED (luz de fundo)	1	Iluminação de fundo	R\$ 1,00 – R\$ 3,00

Resistores 220 Ω	2	Limitação de corrente dos segmentos	R\$ 0,80
Resistores 10 k Ω	3	Resistores de pull-up/pull-down para botoeiras	R\$ 2,00 – R\$ 6,00
Transistor (NPN)	1	Acionamento do LED de backlight	R\$ 0,50 – R\$ 1,00
Fonte de Alimentação 5V	1	Alimentação do sistema	R\$ 18,00

Tabela com Entradas e Saídas

Tipo	Sigla	Função	Pino
Entrada	BTN_START	Botão para iniciar ou pausar a contagem	8
Entrada	BTN_RESET	Botão para zerar o cronômetro (00:00)	9
Entrada	BTN_BACKLIGHT	Botão para ligar/desligar o LED de fundo	10
Saída	LCD_RS	Pino RS do display LCD 16×2	2
Saída	LCD_E	Pino E (Enable) do LCD	3
Saída	LCD_D4	Bit 4 de dados do LCD	4
Saída	LCD_D5	Bit 5 de dados do LCD	5
Saída	LCD_D6	Bit 6 de dados do LCD	6
Saída	LCD_D7	Bit 7 de dados do LCD	7
Saída	LED_BACK	LED backlight	11

Tabela com Variáveis

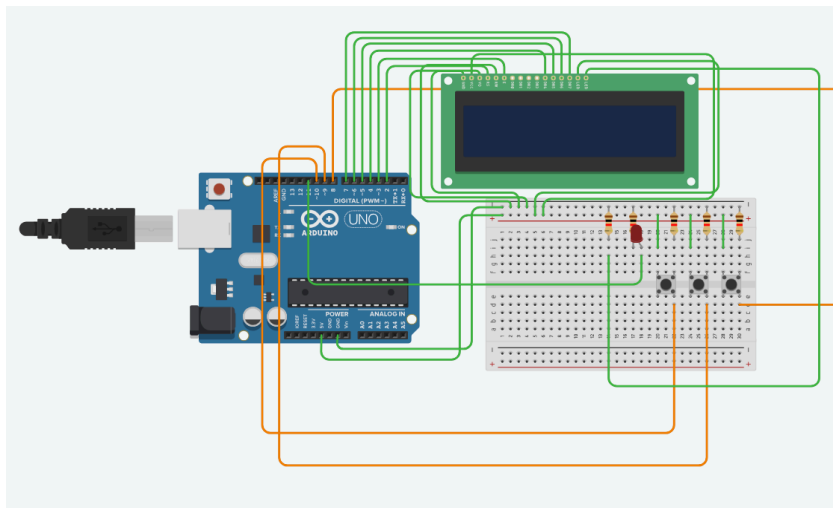
Tipo	Sigla	Função
Variável	min	Contador de minutos (0 a 99)
Variável	sec	Contador de segundos (0 a 59)
Variável	rodando	Indica se o cronômetro está em execução
Variável	luzAtiva	Indica se o LCD e o LED estão acesos
Variável	tempoDecorrido	Guarda o tempo acumulado em ms
Variável	tempoAnterior	Marca o tempo do último loop (para cálculo incremental)

Projetos realizados

Devido a falta de portas digitais (PWM) na placa arduino, realizamos dois projetos, sendo um utilizando um display 16x2 e outro utilizando 4 displays de 7 segmentos, entretanto sem botões. Seguem os respectivos projetos:

Projeto 1:

<https://www.tinkercad.com/things/aGMKlZ0LQ6a-sizzling-hillar-vihelmo/editel?returnTo=https%3A%2F%2Fwww.tinkercad.com%2Fdashboard%2Fdesigns%2Fcircuits>



Código:

```
1  #include <LiquidCrystal.h>
2
3  // LCD nos pinos 2-7
4  LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
5
6  // Botões
7  const int botaoStart = 8;
8  const int botaoReset = 9;
9  const int botaoLuz   = 10;
10
11 // LED indicador (backlight fake)
12 const int ledPin     = 11;
13
14 // Variáveis do cronômetro
15 unsigned long tempoAnterior = 0;
16 unsigned long tempoDecorrido = 0;
17 bool rodando = false;
18 bool luzAtiva = true;
19
20 // Debounce
21 bool lastStart = HIGH;
22 bool lastReset = HIGH;
23 bool lastLuz   = HIGH;
24
25 void setup() {
26     lcd.begin(16, 2);
27
28     pinMode(botaoStart, INPUT_PULLUP);
29     pinMode(botaoReset, INPUT_PULLUP);
30     pinMode(botaoLuz,   INPUT_PULLUP);
31
32     pinMode(ledPin, OUTPUT);
33     digitalWrite(ledPin, HIGH); // luz começa ligada
34
35     lcd.setCursor(0, 0);
36     lcd.print("Cronometro");
37     delay(800);
38     lcd.clear();
39 }
40
```

```

41 void loop() {
42
43     // ===== START / PAUSE =====
44     bool s = digitalRead(botaoStart);
45     if (s == LOW && lastStart == HIGH) {
46         rodando = !rodando;
47         if (rodando) tempoAnterior = millis();
48
49         // REMOVIDO: digitalWrite(ledPin, rodando ? HIGH : LOW);
50         // Agora o LED só responde ao botão da luz
51
52         delay(200);
53     }
54     lastStart = s;
55
56     // ===== RESET =====
57     bool r = digitalRead(botaoReset);
58     if (r == LOW && lastReset == HIGH) {
59         tempoDecorrido = 0;
60         rodando = false;
61         delay(200);
62     }
63     lastReset = r;
64
65     // ===== LUZ LCD + LED =====
66     bool l = digitalRead(botaoLuz);
67     if (l == LOW && lastLuz == HIGH) {
68         luzAtiva = !luzAtiva;
69
70         if (luzAtiva) {
71             lcd.display();
72             digitalWrite(ledPin, HIGH);
73         } else {
74             lcd.noDisplay();
75             digitalWrite(ledPin, LOW);
76         }
77         delay(200);
78     }
79     lastLuz = l;
80

```

```

81 // ===== ATUALIZA TEMPO =====
82 if (rodando) {
83     unsigned long agora = millis();
84     tempoDecorrido += (agora - tempoAnterior);
85     tempoAnterior = agora;
86 }
87
88 // Conversão MM:SS (99:59 máx)
89 unsigned long totalSeg = tempoDecorrido / 1000;
90 unsigned int minutos = totalSeg / 60;
91 unsigned int segundos = totalSeg % 60;
92
93 // Limite máximo: 99:59
94 if (minutos > 99) {
95     minutos = 0;
96     segundos = 0;
97     tempoDecorrido = 0;
98 }
99
100 // ===== VISOR DE 4 DÍGITOS =====
101 lcd.setCursor(0, 0);
102 lcd.print("Tempo:");
103
104 lcd.setCursor(0, 1);
105
106 if (minutos < 10) lcd.print("0");
107 lcd.print(minutos);
108 lcd.print(":");
109
110 if (segundos < 10) lcd.print("0");
111 lcd.print(segundos);
112
113 delay(50);
114 }
115

```

Projeto 2:

https://www.tinkercad.com/things/b8U4z2Yr43R/editel?sharecode=WXR511PUF81PTyEQxptVnRzs4aLGxODYN_NCKf9dJuM

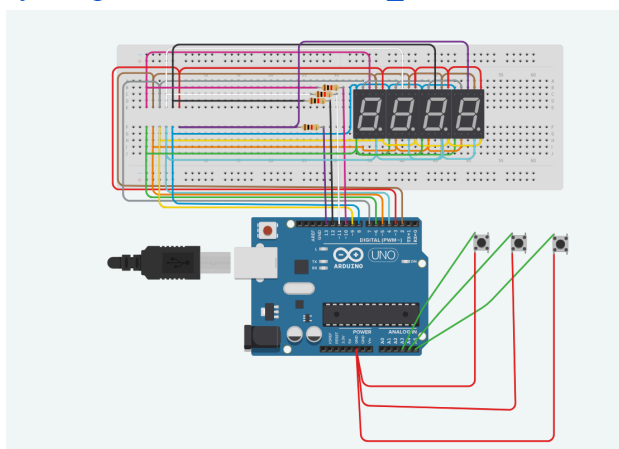


Diagrama de blocos

O seguinte diagrama de blocos demonstra o fluxo de entrada das botoeiras, processamento no microcontrolador e ativação do circuito de leds de acordo com as entradas.

