

Projeto de Microprocessadores

Luiz Augusto Feiten Cisne

December 11, 2023

Contents

1	Modelagem e Projeto do Sistema	2
2	Projeto do hardware	4
3	Projeto do software	5

Modelagem e Projeto do Sistema

O objetivo do projeto em desenvolvimento é criar um detector de intrusos através de um sensor de proximidade construído a partir de um fototransistor. Para implementar o sistema, vamos usar um AT89S51 (8051), um Arduino Uno, alguns LEDs, um display LCD 16x2 e um sensor óptico reflexivo TCRT-5000.

Vamos usar o Arduino uno como gravador de software para o AT89S51, para isso será necessário usar as portas de comunicação serial MOSI(P1.5), MISO(P1.6) e SCK(P1.7), a porta de RST(P1.9) será usada para evitar recarregar o código no Arduino. Para que possa ser possível realizar esse tipo de comunicação vamos usar e modificar o arquivo de exemplo da IDE do Arduino arduinoISP(in circuit serial programming) e para de fato gravar os programas na memória flash do AT89S51 usaremos o avrdude que é um programa de linha de comando usado para gravar dados nos microprocessadores da ATMEL. Por falta de outro componente vou estar usando um cristal oscilador de quartzo de 16MHz conectado em paralelo as portas XTAL1(pino 18) e XTAL2(pino 19) juntamente com 2 capacitores, também em paralelo, de 22 pF.

Para garantir um melhor funcionamento do sistema vamos ligar a única entrada do sensor na porta P3.0 e vamos usar comunicação serial UART pra receber e tratar os dados de detecção de entrada do sensor.

Quando o sensor emitir um sinal, dizendo que existe algo perto, o status do sistema vai atualizar para "Near", e após 10 segundos nesse estado o sistema entrara em estado de "warning",que é estado de alerta máximo. E enquanto o sensor não enviar nada o status vai ser mantido com "far" indicando que não existem objetos no alcance da detecção. Cada um dos estados terá seu respectivo nome exibido no display de LCD, em caso de mudança de estado o cursor apaga o e escreve o nome do estado para qual foi feita a alteração.

O LCD vai estar vai ter suas portas RS, RW e E ligas respectivamente a P2.0, P2.1 e P2.3 o resto dos conectores de dados do LCD (D0 até D7) vai estar ligado na p0. Sendo assim vamos usar um protocolo de comunicação com o LCD de oito entradas de dados.

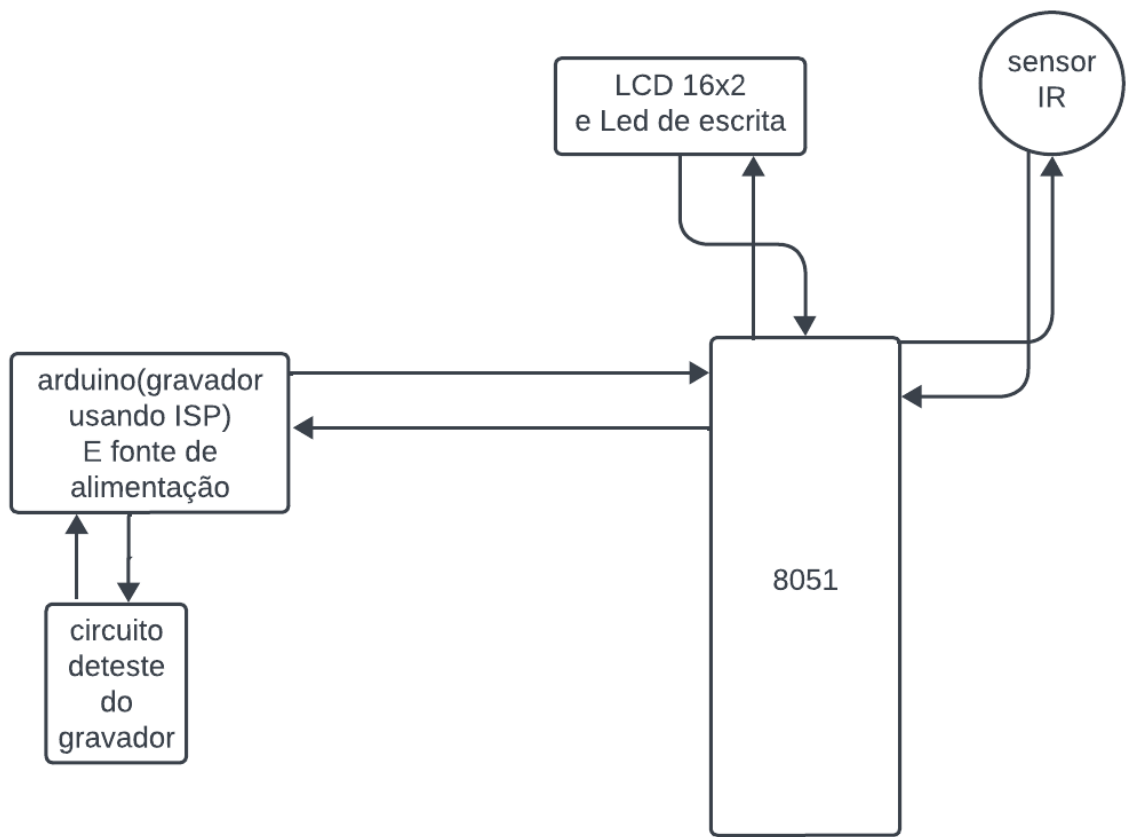


Figure 1.1: Diagrama de blocos

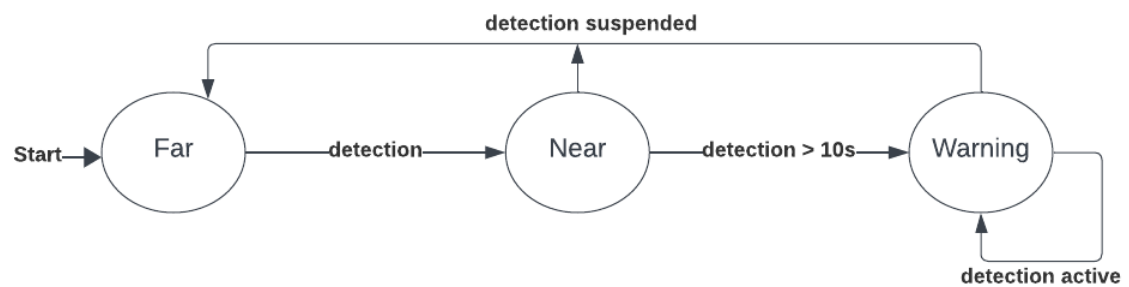


Figure 1.2: Diagrama de estados

Figure 2.1: Esquemático do circuito



Projeto do software

A seguir, está o código em C para o projeto com o microcontrolador 8051:

```
1 #include <reg51.h>
2
3 #define LCD P0
4 sbit IR = P3^0;
5 sbit LED = P1^1;
6
7 sbit RS = P2^0; // Pino RS do LCD
8 sbit RW = P2^1; // Pino RW do LCD
9 sbit E = P2^2; // Pino E do LCD
10
11 // Variáveis para comunica o serial (UART)
12 #define BAUD_RATE 9600
13 #define FREQ_OSC 16000000
14
15 volatile bit motionDetected = 0;
16 unsigned int nearTime = 0;
17
18 void timer1_init();
19 void uart_init();
20 void lcd_init();
21 void lcd_cmd(unsigned char command);
22 void lcd_data(unsigned char dado);
23 void lcd_string(char *str);
24 void delay_timer0(unsigned int milliseconds);
25
26 void main() {
27     timer1_init();
28     uart_init();
29     lcd_init();
30
31     lcd_cmd(0x80); // Posiciona o cursor na primeira linha
32     lcd_string("Status");
33
34     while (1) {
35         if (IR == 0) { // Sensor ligado (detec o)
36             LED = 0; // Liga o LED
37
38             if (!motionDetected) {
39                 nearTime = 0; // Reseta o contador quando algo
```

```

        detectado
        lcd_cmd(0xC0); // Posiciona o cursor na segunda
        linha
        lcd_string("Near      ");
    }

    LED = 1; // Desliga o LED

    while (IR == 0) {
        motionDetected = 1; // Sinaliza que um objeto
        est sendo detectado
        delay_timer0(100); // Atraso para evitar
        contagem r pida de tempo
        nearTime++;

        if (nearTime >= 4000) { // Se passarem mais de
        10 segundos (100 * 100 ms)
            lcd_cmd(0xC0); // Posiciona o cursor na
            segunda linha
            lcd_string("Warning");
        }
    }
}
else {
    motionDetected = 0; // Reseta a flag quando n o h
    movimento
    nearTime = 0; // Reseta o contador quando nada
    detectado
    lcd_cmd(0xC0); // Posiciona o cursor na segunda
    linha
    lcd_string("Far      ");
}
}

}

void timer1_init() {
    TMOD |= 0x20; // Timer 1, modo 2 (8 bits auto reload)
    TH1 = 256 - (FREQ_OSC / (12 * 32 * BAUD_RATE)); //
    Configura o do Timer para a taxa de transmiss o
    TR1 = 1; // Inicia o Timer 1
}

void uart_init() {
    TMOD &= 0x0F; // Limpa os bits relacionados ao Timer 1
    TMOD |= 0x20; // Timer 1, modo 2 (8 bits auto reload)
    SCON = 0x50; // Modo 1, habilita a recep o
    TH1 = 256 - (FREQ_OSC / (12 * 32 * BAUD_RATE)); //
    Configura o do Timer para a taxa de transmiss o
    TR1 = 1; // Inicia o Timer 1
    EA = 1; // Habilita as interrup es globais
    ES = 1; // Habilita as interrup es serial

```

```

80 }
81
82 void lcd_init() {
83     delay_timer0(15); // Aguarda o tempo de inicializa o do
                        LCD
84     lcd_cmd(0x38); // Inicializa o do LCD em modo 8 bits, 2
                        linhas, fonte 5x7
85     lcd_cmd(0x0C); // Exibe o cursor como sublinhado
86     lcd_cmd(0x06); // Incrementa o cursor ap s a escrita
87     lcd_cmd(0x01); // Limpa o display
88 }
89
90 void lcd_cmd(unsigned char command) {
91     RS = 0; // Seleciona o modo de comando
92     RW = 0; // Define a opera o de escrita
93     E = 1; // Ativa o pino E
94     LCD = command; // Envia o comando para o barramento de dados
95     delay_timer0(5); // Aguarda o tempo de execu o do comando
96     E = 0; // Desativa o pino E
97 }
98
99 void lcd_data(unsigned char dado) {
100     RS = 1; // Seleciona o modo de dados
101     RW = 0; // Define a opera o de escrita
102     E = 1; // Ativa o pino E
103     LCD = dado; // Envia os dados para o barramento
104     delay_timer0(5); // Aguarda o tempo de execu o do comando
105     E = 0; // Desativa o pino E
106 }
107
108 void lcd_string(char *str) {
109     while (*str) {
110         lcd_data(*str); // Envia cada caractere da string para o
                        LCD
111         str++;
112     }
113 }
114
115 void delay_timer0(unsigned int milliseconds) {
116
117     // Calcular o valor necess rio para o Timer 0
118     // F rmula: Valor = 65536 - (tempo * (Frequ ncia do cristal
                        ) / 12 / 1000)
119     unsigned int countValue = 65536 - (milliseconds * (FREQ_OSC /
                        12) / 1000);
120
121     // Configurar o Timer 0 em modo 1
122     TMOD &= 0xF0; // Limpar as configura es existentes
123     TMOD |= 0x01; // Modo 1 (temporizador de 16 bits com recarga
                        autom tica)
124

```



```

125 // Configurar os registradores de contagem
126 TH0 = countValue >> 8;
127 TL0 = countValue & 0xFF;
128
129 // Iniciar o Timer 0
130 TR0 = 1;
131
132 // Aguardar at que o Timer 0 atinja o estouro
133 while (!TF0);
134
135 // Desligar o Timer 0
136 TR0 = 0;
137 TF0 = 0; // Limpar a flag de estouro
138 }

```

Listing 3.1: