

**Fundação Edson Queiroz**

**Universidade de Fortaleza**

**Centro de Ciências Tecnológicas**

**PROJETOS DE SISTEMAS MICROCONTROLADOS (N812-9)  
(RELATÓRIO 3)**

**VICTOR KAINÃ DE LIMA DIAS (2216839)**

**Sumário**

1. **Introdução............................................................................1**
2. **Objetivos..............................................................................1**
3. **Metodologia.........................................................................1  
   3.1 Montagem do Circuito no PROTEUS.........................1  
   3.2 Configuração dos Componentes..................................2  
   3.3 Desenvolvimento do Código em C...............................2  
   3.4 Otimização do Circuito.................................................2  
   3.5 Simulação e Testes.........................................................2**
4. **Revisão Bibliográfica..........................................................2  
   4.1 Princípio do Funcionamento do Sensor Ultrassônico2  
   4.2 Cálculo da Distância.....................................................2  
   4.3 Comunicação Serial Bluetooth.....................................3**
5. **Desenvolvimento..................................................................3  
   5.1 Montagem do Circuito..................................................3  
   5.2 Configuração dos Pinos e Componentes.....................3  
   5.3 Desenvolvimento do Código em C...............................3**
6. **Resultados/Conclusões........................................................6**
7. **Bibliografia..........................................................................7**

**1. Introdução**

Este relatório descreve o desenvolvimento e implementação de um projeto de sistema microcontrolado para medição de distância utilizando um sensor ultrassônico, um microcontrolador 8051, comunicação serial Bluetooth e um display LCD. O projeto foi realizado em duas etapas: uma no ambiente de simulação do software PROTEUS e outra em uma implementação física.

**2. Objetivos**

O objetivo principal deste projeto foi desenvolver um sistema que fosse capaz de medir distâncias usando um sensor ultrassônico, exibir os resultados em um display LCD e transmitir os dados via comunicação serial Bluetooth para um dispositivo móvel.

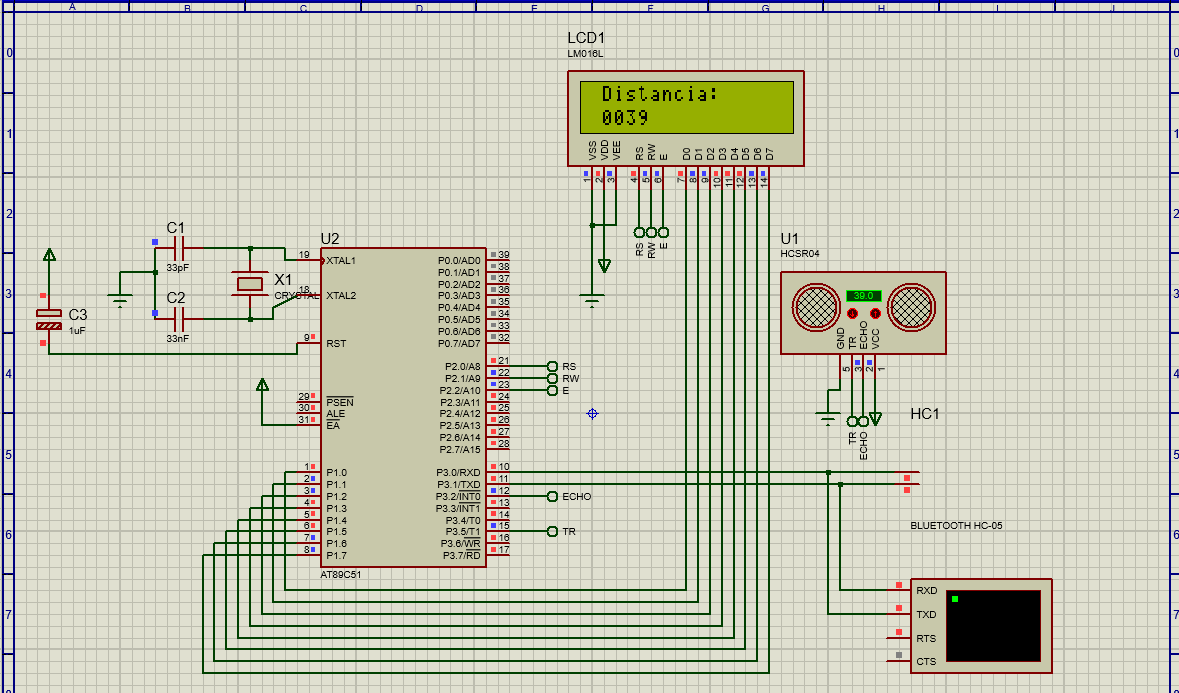
1. Medir a distância utilizando um sensor ultrassônico: Implementar a medição de distâncias através do sensor ultrassônico HC-SR04, aproveitando o método de pulso e eco para determinar a distância de objetos.
2. Exibir os resultados da medição em um display LCD de 16x2: Desenvolver um sistema que mostre as distâncias medidas em tempo real em um display LCD, facilitando a visualização direta dos resultados.
3. Transmitir os dados medidos via comunicação serial Bluetooth: Configurar a comunicação serial Bluetooth com o módulo HC-05, permitindo que os dados das medições de distância sejam enviados para um dispositivo móvel.
4. Implementar e testar o sistema em um ambiente de simulação (PROTEUS): Simular o circuito completo no software PROTEUS para verificar o funcionamento correto antes de proceder com a implementação física.
5. Desenvolver um aplicativo móvel para receber e exibir os dados transmitidos: Criar uma aplicação no App Inventor para receber os dados de distância enviados pelo módulo Bluetooth e exibir essas informações de maneira clara e intuitiva.
6. Otimizar o circuito para implementação prática: Ajustar e refinar o circuito físico utilizando componentes apropriados para garantir que o sistema seja eficiente e funcional quando implementado fora do ambiente de simulação.

**3. Metodologia**

A metodologia adotada para este projeto seguiu uma abordagem estruturada e eficiente, conforme os seguintes passos:

**3.1 Montagem do Circuito no PROTEUS**

Inicialmente, o circuito foi projetado e montado no software PROTEUS **(Figura 1).**



**FIGURA 1 - FONTE: AUTOR**

Os componentes utilizados foram um microcontrolador 8051, um sensor ultrassônico HC-SR04, um módulo Bluetooth HC-05, um display LCD 16x2 e os componentes necessários para a alimentação e conexões.

**3.2 Configuração dos Componentes**

Cada componente foi configurado de acordo com as especificações do datasheet. Os pinos do microcontrolador foram conectados aos componentes de acordo com o esquemático do circuito. O sensor ultrassônico foi conectado aos pinos de trigger **P3.5** e echo **P3.2**, o módulo Bluetooth foi conectado aos pinos de comunicação serial do microcontrolador (**P3.0 – RXD, P3.1 – TXD)**, e o display LCD foi configurado para a comunicação paralela (**PORTA 1)**.

**3.3 Desenvolvimento do Código em C**

O código em linguagem C foi desenvolvido para controlar o funcionamento do sistema. Ele inclui funções para inicializar o display LCD, enviar sinais de trigger para o sensor ultrassônico, medir o tempo de ida e volta do sinal ultrassônico, calcular a distância, exibir os resultados no display LCD e transmitir os dados via comunicação serial Bluetooth. Incluímos as bibliotecas **<8051.h>, <stdio.h>,** e **<serial.h>** e utilizamos o compilador SDCC para compilar o código.

**3.4 Otimização do Circuito**

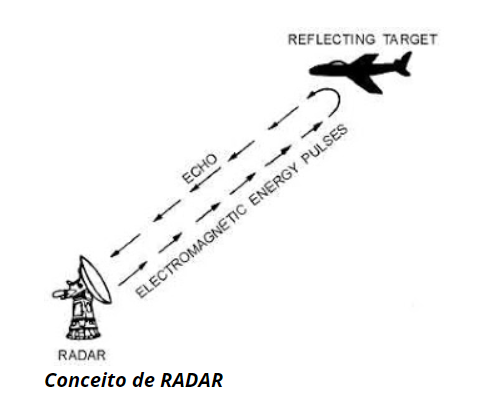
Durante o processo de desenvolvimento, foram realizadas otimizações no circuito para garantir o funcionamento correto do sistema e melhorar a precisão das medições. Isso incluiu ajustes nos parâmetros do sensor ultrassônico, configuração dos temporizadores do microcontrolador, calibração dos componentes do circuito e o delay para o LCD.

**3.5 Simulação e Testes**

Após a montagem do circuito e o desenvolvimento do código, foram realizados testes de simulação no PROTEUS para verificar o funcionamento correto do sistema. Foram realizadas várias simulações para validar o comportamento do sistema em diferentes condições e cenários.

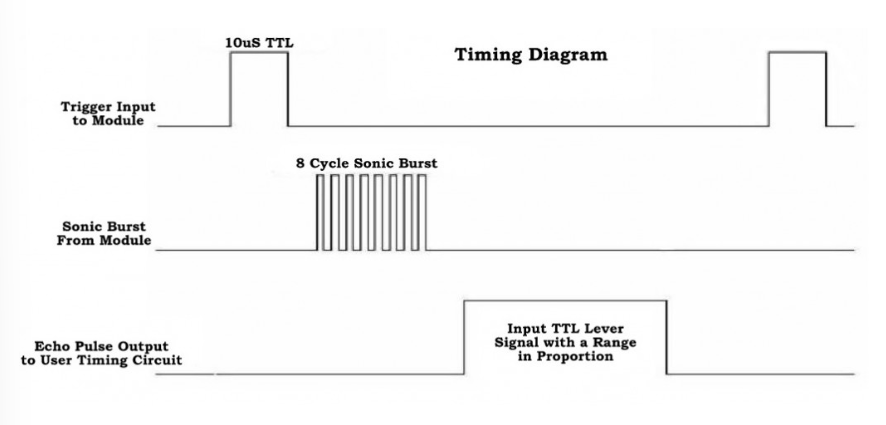
**4. Revisão Bibliográfica**

**4.1 Princípio do Funcionamento do Sensor Ultrassônico**  
O sensor ultrassônico HC-SR04 utiliza o princípio do eco para medir distâncias (**FIGURA 2)**. Ele emite um pulso ultrassônico e mede o tempo que leva para o sinal retornar após ser refletido por um objeto. Esse tempo é proporcional à distância do objeto ao sensor.

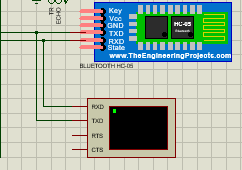


**FIGURA 2 - FONTE: ELECTRONICSHUB.ORG**

**4.2 Cálculo da Distância**  
A distância pode ser calculada usando a fórmula: Distância (cm) = Tempo (µs) / 58. Isso leva em consideração a velocidade do som no ar, que é de aproximadamente 343 metros por segundo (**FIGURA 3)**.



**FIGURA 3 - FONTE: ELECTRONICSHUB.ORG**

**4.3 Comunicação Serial Bluetooth**  
O módulo Bluetooth HC-05 é utilizado para transmitir os dados de distância medidos pelo microcontrolador para um dispositivo móvel, como um smartphone ou tablet. Ele utiliza comunicação serial para enviar os dados em tempo real **(FIGURA 4 E 5)** .

**FIGURA 4**

**FONTE: AUTOR**



**FIGURA 5 – FONTE: AUTOR**

**5. Desenvolvimento**

**5.1 Montagem do Circuito**  
Após a simulação e teste no PROTEUS, o circuito foi montado fisicamente em uma placa de prototipagem **(FIGURA 6)**. Todos os componentes foram conectados de acordo com o esquemático do circuito.



**FIGURA 6**

**FONTE: AUTOR**

**5.2 Configuração dos Pinos e Componentes**  
Os pinos dos componentes foram conectados aos pinos do microcontrolador de acordo com o esquemático do circuito. Foram utilizados jumpers para fazer as conexões entre os componentes.

**5.3 Desenvolvimento do Código em C**  
O código em linguagem C foi desenvolvido para controlar o funcionamento do sistema. Ele inclui funções para inicializar o display LCD, enviar sinais de trigger para o sensor ultrassônico, medir o tempo de ida e volta do sinal ultrassônico, calcular a distância, exibir os resultados no display LCD e transmitir os dados via comunicação serial Bluetooth.

*// Código completo em C desenvolvido para o microcontrolador 8051*

*// Inclui funções para inicializar o display LCD, medir a distância com o sensor ultrassônico e enviar dados via comunicação serial Bluetooth*

*// Desenvolvido por Victor Kainã de Lima Dias e Gabriel Pontes Araújo*

*#include <8051.h>*

*#include <stdio.h>*

*#include <serial.h>*

*// Definição da velocidade do som em cm por segundo*

*#define velocidade\_som 34300*

*// Definição dos pinos do display LCD*

*#define LCD\_Data P1*

*\_\_sbit \_\_at (0xB5) trig;*

*\_\_sbit \_\_at (0xB2) echo;*

*\_\_sbit \_\_at (0xA0) LCDrs;*

*\_\_sbit \_\_at (0xA1) LCDrw;*

*\_\_sbit \_\_at (0xA2) LCDen;*

*// Função para inicializar a comunicação serial*

*void inicia\_serial() {*

*TMOD |= 0x20; // Configura o Timer 1 em modo 2 (8-bit auto-reload)*

*TH1 = 0xFD; // Define a taxa de transmissão para 9600 bps*

*TL1 = 0xFD;*

*SCON = 0x50; // Configura o modo de operação serial (modo 1)*

*TR1 = 1; // Inicia o Timer 1*

*}*

*// Função para enviar um caractere via comunicação serial*

*void send\_char\_serial(char c) {*

*SBUF = c; // Carrega o caractere no buffer serial*

*while (!TI); // Espera até que a transmissão esteja concluída*

*TI = 0; // Reseta a flag de transmissão*

*}*

*// Função para enviar uma string via comunicação serial*

*void send\_string\_serial(char\* str) {*

*int i = 0;*

*while (str[i] != '\0') {*

*send\_char\_serial(str[i]);*

*i++;*

*}*

*}*

*// Função para gerar um atraso de 1 µs*

*void Delay\_us() {*

*TL0 = 0xF5;*

*TH0 = 0xFF;*

*TR0 = 1; // Inicia o Timer 0*

*while (!TF0); // Aguarda a interrupção do Timer 0*

*TR0 = 0; // Para o Timer 0*

*TF0 = 0; // Reseta o flag de overflow do Timer 0*

*}*

*// Função para inicializar o Timer 0*

*void inicia\_timer() {*

*TMOD |= 0x01; // Configura o Timer 0 em modo 1 (16-bit)*

*TF0 = 0;*

*TR0 = 0;*

*}*

*// Função para enviar um pulso de trigger de 10 µs ao sensor ultrassônico*

*void send\_trigger\_pulse() {*

*trig = 1; // Define o pino de trigger como HIGH*

*Delay\_us(); // Gera um atraso de 10 µs*

*trig = 0; // Define o pino de trigger como LOW*

*}*

*// Função para gerar um atraso em milissegundos*

*void delay(unsigned int k) {*

*int i, j;*

*for (i = 0; i < k; i++)*

*for (j = 0; j < 112; j++);*

*}*

*// Função para enviar um comando para o display LCD*

*void LCD\_Command(char Command) {*

*LCD\_Data = Command;*

*LCDrs = 0;*

*LCDrw = 0;*

*delay(2);*

*LCDen = 1;*

*delay(2);*

*LCDen = 0;*

*delay(3);*

*}*

*// Função para enviar um caractere para o display LCD*

*void LCD\_Char(char Data) {*

*LCD\_Data = Data;*

*LCDrs = 1;*

*LCDrw = 0;*

*delay(2);*

*LCDen = 1;*

*delay(2);*

*LCDen = 0;*

*delay(3);*

*}*

*// Função para exibir uma string no display LCD*

*void LCD\_String(unsigned char \*str) {*

*int i;*

*for (i = 0; str[i] != 0; i++) {*

*LCD\_Char(str[i]);*

*}*

*}*

*// Função para exibir uma string em uma posição específica no display LCD*

*void LCD\_String\_xy(unsigned char row, unsigned char pos, unsigned char \*str) {*

*if (row == 1)*

*LCD\_Command((pos & 0x0F) | 0x80);*

*else if (row == 2)*

*LCD\_Command((pos & 0x0F) | 0xC0);*

*LCD\_String(str);*

*}*

*// Função para inicializar o display LCD*

*void inicia\_LCD() {*

*delay(1000);*

*LCD\_Command(0x30); // Configura o display para modo de operação*

*LCD\_Command(0x30);*

*LCD\_Command(0x30);*

*LCD\_Command(0x38); // Define o modo de operação: 8 bits, 2 linhas*

*LCD\_Command(0x0C); // Liga o display e desliga o cursor*

*LCD\_Command(0x01); // Limpa o display*

*LCD\_Command(0x06); // Define o modo de entrada*

*LCD\_Command(0x80); // Posiciona o cursor na primeira linha*

*}*

*// Função para medir a distância usando o sensor ultrassônico*

*int medir\_distancia() {*

*int range = 0;*

*long int timerval;*

*float fator\_correcao = 1.02; // Ajuste para calibrar o sensor*

*send\_trigger\_pulse(); // Envia um pulso de 10 µs*

*while (!echo); // Aguarda o pino de echo ser ativado*

*TR0 = 1; // Inicia o Timer 0*

*while (echo && !TF0); // Aguarda o pino de echo retornar LOW*

*TR0 = 0; // Para o Timer 0*

*timerval = TH0;*

*timerval = ((timerval << 8) | TL0); // Lê o valor do Timer 0*

*if (timerval < velocidade\_som)*

*range = (int)((timerval \* velocidade\_som \* fator\_correcao) / (2 \* 1000000));*

*else*

*range = 0;*

*return range;*

*}*

*// Função principal do programa*

*void main() {*

*int range = 0;*

*int i = 0, j;*

*int soma\_distancias = 0;*

*int num\_leituras = 10; // Número de leituras para média*

*unsigned char str[8] = "0000"; // String para exibir a distância*

*inicia\_LCD(); // Inicializa o display LCD*

*LCD\_String\_xy(1, 1, (unsigned char \*)"Distancia:"); // Exibe "Distancia:" na primeira linha*

*inicia\_timer();*

*inicia\_serial(); // Inicializa a comunicação serial*

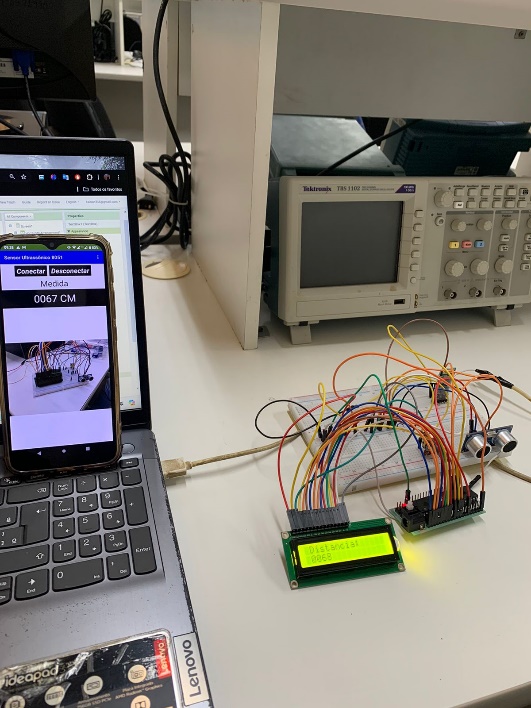
*while (1) {*

*soma\_distancias = 0;*

*for (j = 0; j < num\_leituras; j++) {*

*soma\_distancias += medir\_distancia(); // Mede a distância e acumula*

*delay(10); // Pequena pausa entre as leituras*

 *}*

*range = soma\_distancias / num\_leituras; // Calcula a média das leituras*

*// Converte o número para uma string de 4 dígitos*

*for (i = 3; i >= 0; i--) {*

*str[i] = 0x30 | (range % 10);*

*range = range / 10;*

*}*

*LCD\_String\_xy(2, 1, str); // Exibe a distância no LCD*

*send\_string\_serial(str); // Envia a distância pela comunicação serial*

*send\_char\_serial('\n'); // Envia uma nova linha para finalizar a mensagem*

*delay(100);*

*}*

*}*

**6. Resultados/Conclusões**

Durante o desenvolvimento e testes do sistema, foram observados os seguintes resultados:

* **Simulação no PROTEUS**: Todos os componentes foram corretamente simulados, e o comportamento do sensor ultrassônico, display LCD e comunicação serial Bluetooth foram validados. A medição da distância foi precisa e consistente com os cálculos teóricos.
* **Implementação Física**: Após a montagem física do circuito, o sistema funcionou conforme esperado. As medições de distância foram exibidas no display LCD e transmitidas para um dispositivo móvel via Bluetooth. O aplicativo no smartphone exibiu as distâncias em tempo real, permitindo a visualização e monitoramento das medições (**FIGURA 7).**

**FIGURA 7 – FONTE: AUTOR**

**Conclusões Finais**:

* O projeto alcançou os objetivos propostos, demonstrando a viabilidade de usar um sensor ultrassônico, um microcontrolador 8051 e comunicação Bluetooth para medir distâncias.
* A integração entre hardware e software foi bem-sucedida, proporcionando uma solução funcional para a medição de distância.
* Sugere-se melhorias futuras, como a calibração mais precisa do sensor e a adição de funcionalidades, como alarme sonoro ou visualização em gráficos.

**7. Bibliografia**

1. "Ultrasonic Module HC-SR04 Interfacing with 8051." ElectronicWings. Disponível em: <https://www.electronicwings.com/8051/ultrasonic-module-hc-sr04-interfacing-with-8051>
2. "Ultrasonic Rangefinder using 8051." ElectronicsHub. Disponível em: <https://www.electronicshub.org/ultrasonic-rangefinder-using-8051/>
3. "Interface a Ultrasonic Sensor with 8051 Microcontroller." HackProjects. Disponível em: <https://hackprojects.wordpress.com/forum/8051-2/interface-a-ultrasonic-sensor-with-8051-microcontroller/>