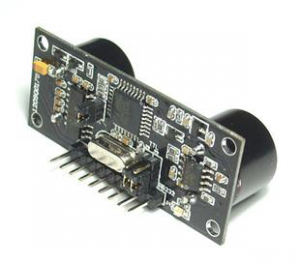
URM37 Sensor Ultrassônico V3.2 (SKU: SEN0001)

|  |
| --- |
| **Conteúdo**   [ [esconder](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)) ]   * [1 Introdução](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Introduction) * [2 Especificação](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Specification)   + [2,1 Comparar com outro sensor ultra-sónico](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Compare_with_other_ultrasonic_sensor) * [3 requierments Hardware](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Hardware_requierments) * [4 Ferramentas usadas](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Tools_used) * [5 Software](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Software) * [6 de selecção de modo](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Working_Mode_Selection)   + [6,1 Modo 1: modo de controle de série passiva](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Mode_1:_Serial_passive_control_mode)   + [6,2 configuração de Jumper para saída RS232 e TTL](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Jumper_setting_for_RS232_and_TTL_output)   + [6,3 Modo 2: modo de disparo Autónoma](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Mode_2:_Autonomous_trigger_mode)   + [6,4 Modo 3: PWM modo de controle passivo](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Mode_3:_PWM_passive_control_mode)   + [6,5 O esboço de modo PWM controle passivo](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#The_sketch_for_PWM_passive_control_mode) * [7 protocolo de controle de série](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Serial_control_protocol) * [8 Esboço Arduino](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Arduino_Sketch) * [9 Recursos](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001)#Resources) |

Introdução

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor1.png)

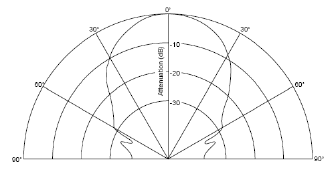
[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor1.png)

URM37 V3.2 manual do sensor ultra-sônico - Rev 2.2

URM37 Sensor Ultrassônico V3.2 usa um processador AVR industrial nível como a unidade de processamento principal. Ele vem com uma correção de temperatura, que é muito original em sua classe.

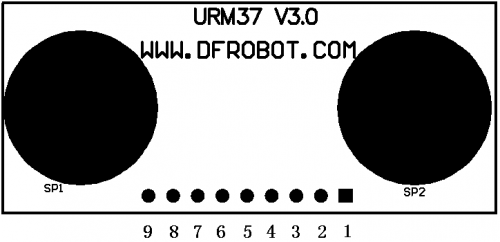
Especificação

* Alimentação: +5 V
* Corrente: <20mA
* Temperatura de trabalho: -10 ℃ ~ 70 ℃
* Faixa de detecção: 4cm-5m
* Resolução: 1cm
* Interface: RS232 (TTL), PWM
* Controle servo: Uma saída de controle servo
* Modo de operação: de série; (PWM) modo de controle passivo; modo autônomo; ON / OFF
* Sensor de temperatura: 12 bits de leitura da porta serial
* Tamanho: mm 22 milímetros × 51
* Peso: 30 g

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor2.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor2.png)

Figura 1: URM37 Largura Boca V3.2

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor3.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor3.png)

Figura 2: Definição URM37 Pin V3.2

1. **+ VCC** - +5 V Potência   
2. **GND** Ground -   
3. **RST** - Reset   
4. **PWM** - PWM Saída 0-25000US, cada 50US representam um centímetro   
5. **MOTO** - saída de sinal de Servo controle   
. 6 **COMP / TRIG**

**COMP** - On / OFF modo, quando a distância de detecção é inferior a um valor pré-estabelecido, este pino puxa baixo.

**TRIG** - PWM ou RS232 pino do gatilho

7. **NC**   
8. **RXD** - RS232, TTL comunicação   
. 9 **TXD** - RS232, TTL comunicação

**Comparar com outro sensor ultra-sónico**

[Compare com SRF08 e XL-MaxSonar-WRC1 sensor de distância ultra-Avaliação](http://www.dfrobot.com/community/2012/06/1671/)

Requierments Hardware

1. 1 × URM37 V3.2 Sensor Ultrasonic
2. 1 × microcontrolador Arduino
3. 1 × Expansão IO Shield para Arduino (V5)
4. 1 × cabo USB

Ferramentas usadas

* 4 × jumper

Software

* Arduino IDE

Seleção do modo de trabalho

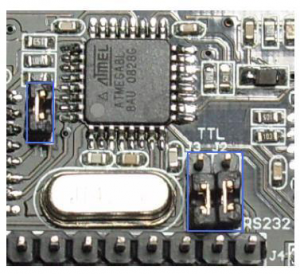
O modo de funcionamento pode ser alterado ao escrever 0x01, 0x00, 0x02 ou a EEPROM através da porta serial.

**Modo 1: modo de controle de série passiva**

Neste modo, o sensor está sempre à espera de comando a partir da porta de série. Cada vez que recebe um comando, ele irá retornar a distância e esperar o próximo comando. O grau em que o comando será utilizado para controlar um servo motor para girar grau correspondente. Por favor note que este modo está sempre ligado. Ele não pode ser ligar ou desligar.

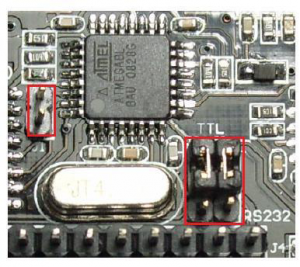
**Configuração de jumper para a saída RS232 e TTL**

A selecção de RS232 TTL ou o nível de saída é ligado por três pontes mudando (J1, J2, J3). Um diagrama abaixo ilustra a configuração:

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor4.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor4.png)

Modo RS232

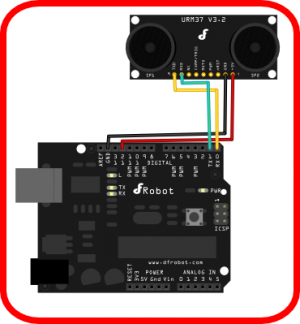
[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor5.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor5.png)

Modo TTL

*Aviso: Não ligue para TTL MCU quando o modo de saída está definido para RS232, isso irá danificar permanentemente o aparelho.*

Este recurso está disponível apenas para Rev2 e depois. Se não existirem pontes na parte de trás do sensor, o sensor é Rev1 e, portanto, não suportando esta característica.

[](http://www.dfrobot.com/wiki/index.php/File:URM37_ConnectionDiagram.png)

**Modo 2: modo de disparo Autónoma**

Neste modo, o sensor irá fazer um sensor de leitura a cada 25ms e comparar a leitura com um limiar (pré-definido, o utilizador é capaz de definir este valor pela escrita EEPROM), se a leitura é igual ou menor do que o limiar, o pino COMP / TRIG terá saída baixa. Enquanto isso, o pino PWM output a leitura à distância, cada nível 50us baixo significa 1cm, através da contagem do número destes pulsos, a distância pode ser calculada. Este modo pode ser usado apenas como um interruptor ON / OFF.

**Modo 3: PWM modo de controle passivo**

Neste modo, um puxão baixo no pino COMP / TRIG irá desencadear uma leitura do sensor. A largura do pulso é proporcional ao grau de rotação do servo. Depois de uma leitura do sensor de sucesso, Pin PWM irá gerar pulsos, cada 50us representa 1cm. Se a leitura é inválido, um pulso 50000us será devolvido.

**O esboço de modo PWM controle passivo**

[?](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | / / # Editor: Jiang de DFRobot  / / # Data: 2012/09/18    / / # Nome do produto: scanner de ultra-sons  / / # SKU do produto: SEN0001  / / # Versão: 0,2    / / # Descrição:  / / # O Esboço para digitalizar 180 área de grau 4-500 centímetros faixa de detecção    / / # Conexão:  / / # Pino 1 VCC (V3.2 URM) -> VCC (Arduino)  / / # Pino 2 GND (URM V3.2) -> GND (Arduino)  / / # 4 Pin PWM (URM V3.2) -> Pino 3 (Arduino)  / / # Pino 6 COMP / TRIG (URM V3.2) -> Pino 5 (Arduino)  / / #  int URPWM = 3; / / Saída PWM 0-25000US, cada 50US representam um centímetro  int URTRIG = 5; / / pino do gatilho PWM    unsigned int Distância = 0;  uint8\_t EnPwmCmd [4] = {0x44, 0x02, 0xbb, 0x01};    comando medida / distância /    vazio setup () {                                 / / inicialização de série    Serial.begin (9600);                         / / Define a taxa de transmissão para 9600    PWM\_Mode\_Setup ();  }    vazio loop ()  {   PWM\_Mode ();   atraso (20);  }                      / / PWM função de configuração do modo    vazio PWM\_Mode\_Setup () {    pinMode (URTRIG, OUTPUT);                     / / puxar um baixo no pino COMP / TRIG    digitalWrite (URTRIG, HIGH);                  / Set / para ALTA      pinMode (URPWM, INPUT);                      / / Enviando comando Ativar modo PWM      para ( int i = 0; i <4; i + +) {        Serial.write (EnPwmCmd [i]);     }  }    vazio PWM\_Mode () {                              / / um puxão baixo no pino COMP / TRIG provocando uma leitura do sensor      digitalWrite (URTRIG, LOW);      digitalWrite (URTRIG, HIGH);               / leitura / Pin PWM pulsos de saída        unsigned longo DistanceMeasured = pulseIn (URPWM, LOW);        se (DistanceMeasured == 50000) {              / / a leitura é inválido.        Serial.print ( "inválido" );     }      mais {        = Distância DistanceMeasured/50;           / / todos os níveis 50us baixo significa um centímetro     }    Serial.print ( "Distância =" );    Serial.print (Distância);    Serial.println ( "CM" );  } |

[](http://www.dfrobot.com/wiki/index.php/File:URM37_Connection.jpg)

Protocolo de controle de série

Configuração de série: taxa de Port: 9600; Paridade: none; Stop bit: 1

Comando: Comando é composto de quatro bits, comando + + DATA0 data1 + soma. Soma = 8 bits Baixa da soma de comando + + DATA0 dados1.

|  |  |  |
| --- | --- | --- |
| **Formato de comando** | **Função** | **Descrição** |
| 0x11 + NC + NC Soma + (Amostra: 0x00 0x00 0x11 0x11) | Habilitar leitura de temperatura de 16 bits | Ler a temperatura, o formato de dados de retorno será:  0x11 + elevada (temperatura) + baixa (temperatura) SUM + Se a temperatura é superior a 0, os primeiros quatro bits de alta será tudo 0. Se a temperatura é inferior a 0, os primeiros quatro bits de alta será todos 1. Os últimos quatro bits de alta, juntamente com os bits de baixa temperatura significa 12bits. A resolução é de 0,1. Quando a leitura é inválido, ele retorna 0x11 0 0 xFF xFF SUM + |
| 0x22 + Grau + NC + SUM (Amostra: 0x00 0x00 0x22 0x22) | Permitir leitura à distância de 16 bits | O grau em que o comando é utilizado para controlar um servo motor para girar grau correspondente.  Grau:. 0-46 stands para 0-270 graus, por exemplo, três stands para 18 graus  formato de dados de retorno será: 0x22 + Alta (distância) + Baixo (distância) SUM +. Quando a leitura é inválido, ele retorna 0x22 0 0 xFF xFF SUM + |
| 0x33 + Add + NC + SOMA | Ativar interna leitura EEPROM | Dados de retorno será 0x33 + Add + dados + SOMA. |
| 0x44 + Add + dados + SUM (Amostra: 0x02 0x44 0x01 0xbb) Ativar modo PWM | Ativar interna escrita EEPROM | Dados escritos só pode 0-255.  Endereço 0x00 0x02 é usado para configurar o modo. 0x00 - distância-limite (Baixo) 0x01 - distância-limite (Alta) 0x02 - Modo de Operação (0xaa de modo autônomo) (0xbb para PWM modo de controle passivo) O retorno formato de dados será: 0x44 + Add + dados + SOMA |

Nota: NC significa quaisquer dados, SUM significa soma, Adicionar stands de endereço.

*1. PWN\_ON deve ser ajustado para alta para permitir sensor.*

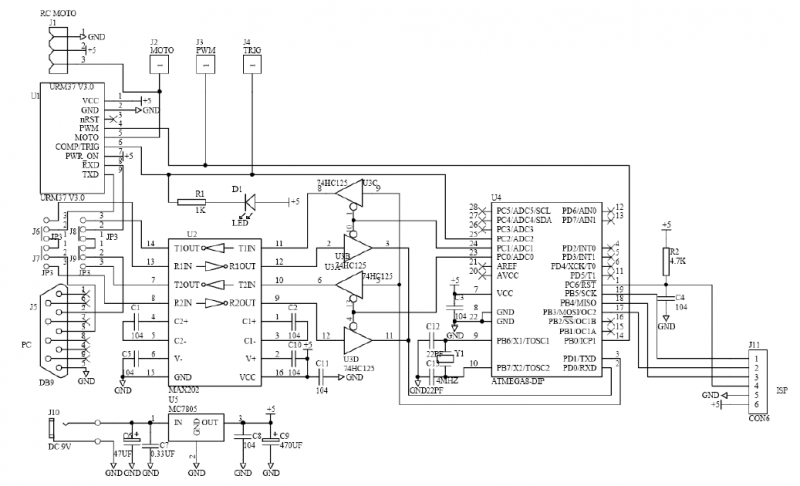
**Exemplos:** Função para calcular a temperatura:

[?](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))

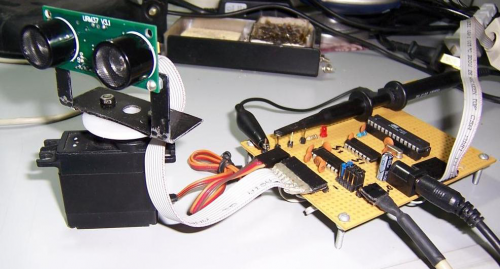
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | IF (HightByte> = 0xF0)  {  Temperatura = ((HightByte-0xF0) \* 256-Lowbyte) / 10   }  Outro  {  Temperatura = ((HightByte) \* 256-Lowbyte) / 10  } |

Servo controle de tabela de referência de comando:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dezembro | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| HEX | 0 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0E | 0E | 0F |
| Grau | 0 | 6 | 12 | 18 | 24 | 29 | 35 | 41 | 47 | 53 | 59 | 65 | 70 | 76 | 82 | 88 |
| Dezembro | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| HEX | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 1A | 1B | 1C | 1D | 1E | 1F |
| Grau | 94 | 100 | 106 | 112 | 117 | 123 | 129 | 135 | 141 | 147 | 153 | 159 | 164 | 170 | 176 | 182 |
| Dezembro | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 |  |
| HEX | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 2A | 2B | 2C | 2D | 2E |  |
| Grau | 188 | 194 | 200 | 206 | 211 | 217 | 223 | 229 | 235 | 241 | 247 | 252 | 258 | 264 | 270 |  |

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor6.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor6.png)

[](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor7.png)

[http://www.dfrobot.com/wiki/skins/common/images/magnify-clip.png](http://www.dfrobot.com/wiki/index.php/File:URM37_V3.2_Ultrasonic_Sensor7.png)

Figura controle UMRV3.2 um servo fornece 270 área de digitalização grau

**V3.2 Baixar Companheiro Ajuda:**  
[Ajuda-Mate](http://www.dfrobot.com/image/data/SEN0001/URMV3.2HelpMate.rar)

Esboço Arduino

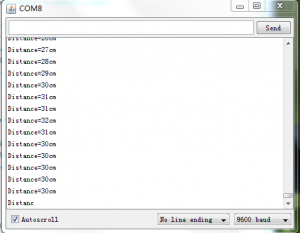
Use um servo para controlar a posição, eo sensor de ultra-som para julgar o distance.Here é o esboço.

[](http://www.dfrobot.com/wiki/index.php/File:Ultrasonic_sensor_kit.jpg)

NOTA: Por favor, coloque os jumpers de sensores para o modo TTL. Veja acima a imagem indicando TTL

[?](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93 | / / # Editor: Jiang de DFRobot  / / # Data: 2012/07/24    / / # Nome do produto: Kit de scanner de ultra-som  / / # SKU do produto: SEN0001  / / # Versão: 0,2    / / # Descrição:  / / # O Esboço para digitalizar 180 área de grau 4-500 centímetros faixa de detecção    / / # Conexão:  / / # Pino 1 VCC (V3.2 URM) -> VCC (Arduino)  / / # Pino 2 GND (URM V3.2) -> GND (Arduino)  / / # 4 Pin PWM (URM V3.2) -> Pino 3 (Arduino)  / / # Pino 6 COMP / TRIG (URM V3.2) -> Pino 5 (Arduino)  Modo / / # Pin: PWM  / / # Modo de Funcionamento: PWM modo de controle passivo.  / / # Se é a primeira vez que usá-lo, por favor, certifique-se que os dois jumpers para a direita  / / # Lado do dispositivo são definidos para o modo TTL. Você também vai encontrar um jumper secundário em  / / # Do lado esquerdo, você deve quebrar esta ligação ou você pode danificar o aparelho.    # Include <Servo.h> / / Incluir biblioteca Servo  Myservo Servo;                                      / / cria o objeto servo para controlar um servo    int pos = 0;                                          / / variável para armazenar a posição do servo  int URPWM = 3;                                        / / Saída PWM 0-25000us, a cada 50us representam um centímetro  int URTRIG = 5;                                       / / pino do gatilho PWM  se boolean = verdadeiro ;                                    / / cria uma variável booleana  unsigned longo tempo ;                                 / / cria uma variável de tempo  unsigned longo urmTimer = 0;                          / temporizador / para a gestão do sensor de leitura taxa de flash    unsigned int Distância = 0;  uint8\_t EnPwmCmd [4] = {0x44, 0x22, 0xbb, 0x01};          comando medida / distância /    vazio setup () {                                      / / inicialização de série    Serial.begin (9600);                              / / Define a taxa de transmissão para 9600    myservo.attach (9);                                / / Pino 9 para controlar servo    PWM\_Mode\_Setup ();  }    vazio loop () {     se (millis () - tempo > = 20) {                           intervalo / / 0,02 segundo       tempo = millis ();                                 / / obtém a hora atual do programa       se (até) {                                        / juiz / a condição        se (pos> = 0 && pos <= 179) {          pos = pos 1;                                  / / em passos de 1 grau          myservo.write (pos);                         / / diz servo de ir para a posição na variável 'pos'        }        se (pos> 179) para cima = false ;                       / / atribuir a variável novamente      }       mais {        se (pos> = 1 && pos <= 180) {          pos = pos-1;          myservo.write (pos);        }        se (pos <1) para cima = verdadeiro ;      }     }       se (millis ()-urmTimer> 50) {       urmTimer = millis ();       PWM\_Mode ();     }   }     vazio PWM\_Mode\_Setup () {    pinMode (URTRIG, OUTPUT);                            / / puxar um baixo no pino COMP / TRIG    digitalWrite (URTRIG, HIGH);                         / Set / para ALTA      pinMode (URPWM, INPUT);                             / / Enviando comando Ativar modo PWM      para ( int i = 0; i <4; i + +) {        Serial.write (EnPwmCmd [i]);     }  }    vazio PWM\_Mode () {                                     / / um puxão baixo no pino COMP / TRIG provocando uma leitura do sensor      digitalWrite (URTRIG, LOW);      digitalWrite (URTRIG, HIGH);                      / leitura / Pin PWM pulsos de saída        unsigned longo DistanceMeasured = pulseIn (URPWM, LOW);        se (DistanceMeasured == 50000) {                     / / a leitura é inválido.        Serial.print ( "inválido" );     }      mais {        = Distância DistanceMeasured/50;                  / / todos os níveis 50us baixo significa um centímetro     }    Serial.print ( "Distância =" );    Serial.print (Distância);    Serial.println ( "CM" );  } |

[](http://www.dfrobot.com/wiki/index.php/File:The_distance_measured.png)

Recursos

* [Arduino Biblioteca de milesburton (IDE 0023 e abaixo)](http://milesburton.com/URM37_Ultrasonic_Distance_Measurement_Library)
* [Arduino Biblioteca de Lauren (Apenas Arduino IDE 1.0)](http://www.dfrobot.com/image/data/SEN0001/URM37%20library%20for%20Arduino%20IDE%201.0.rar)