

Projeto Sensor URM37 com Arduíno

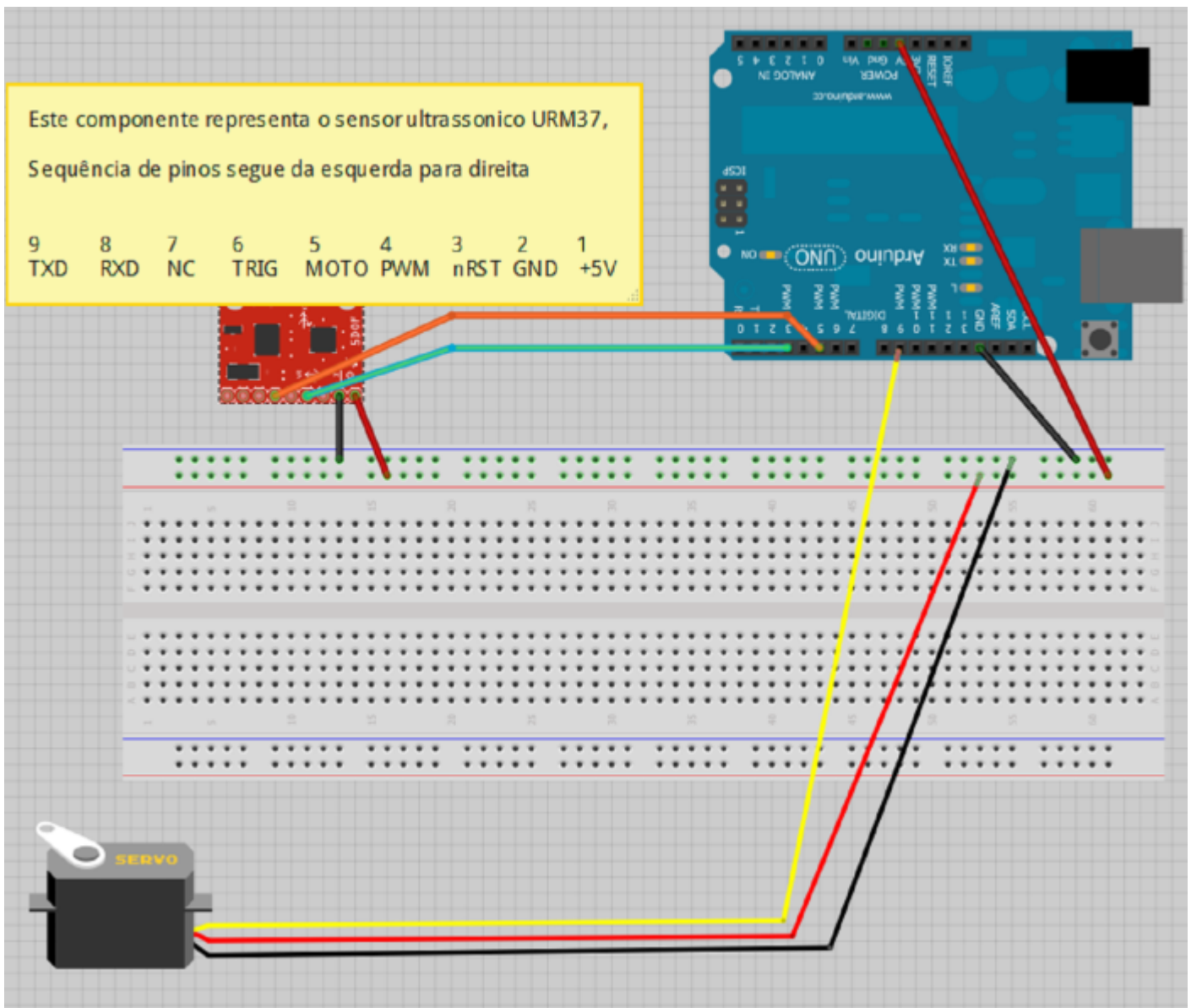
□ schmittjoaopedro . Arduino, Linguagem C □ 19 19+00:00 Março 19+00:00 2013 05+00:00 Agosto 05+00:00 2013 □ 3 Minutes

Neste tópico abordaremos o projeto utilizando um sensor de presença que ativará a movimentação de um motor servo conforme a proximidade com os sensores.

Para esse projeto foram utilizados os seguintes componente:

- Placa Arduíno UNO R3
- Um sensor ultrassônico tipo URM37.
- Um motor servo modelo HTX900

Faça as conexões dos componentes conforme a imagem abaixo, o sensor ultrassônico foi representado utilizando outro componente que contém as pinagens certas, o diagrama acima do componente indica os pinos correspondentes. Pino +5V vai ligado aos 5V do Arduíno, pino GND vai ligado ao terra do Arduíno, pino TRIG e PWM vão ligados a portas arduínos com suporte PWM.



(<https://joaoschmitt.files.wordpress.com/2013/03/projeto-sensor-urm371.png>).

Faça download do projeto para Fritzing: <http://db.tt/r6nimsap> (<http://db.tt/r6nimsap>).

Após certificado de que todas ligações encontram-se conforme o esquema de montagem acima. Basta fazer upload do código abaixo.

Código para upload pronto:

```
#include <Servo.h> // Comando para importação da biblioteca que faz controle do servo motor
int URPWM = 3; // Saída PWM 0-25000US, cada 50US representam um centímetro
int URTRIG=5; // Pino do gatilho PWM
Servo servo1; // Variável do tipo servo que fará dar acesso as funções do servo motor
// # 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)
unsigned long distanciaMedida; // Variável para armanezar o valor lido pelo sensor
int anteriorMenor = 0; // Variável de controle para impedir oscilação contante do motor servo
armazenará um valor menor que o lido
```

```

int anteriorMaior = 0; // Variável de controle para impedir oscilação contante do motor servo
armazenará um valor maior que o lido
unsigned int distancia=0; // Variável que armazenará o valor para impressão e escrita no servo
uint8_t EnPwmCmd[4]={0x44,0x02,0xbb,0x01}; // comando medida / distância /
//Seta os parâmetros iniciais
void setup(){
  Serial.begin(9600); // Inicia a configuração serial com o monitor em uma taxa de 9600 bits
  ModoPWMSetup(); // Invoca função ModoPWMSetup
  servo1.attach(9); // Define o pino 9 para escrita do servo motor
}
//Inicio o loop de repetição infinita
void loop()
{
  ModoPWM(); //Chamada da função ModoPWM
  delay(500); //Delay de 500 milissegundos até o próximo loop
}
void ModoPWMSetup(){
  pinMode(URTRIG,OUTPUT); // puxar um baixo no pino COMP / TRIG
  digitalWrite(URTRIG,HIGH); // Set / para ALTA
  pinMode(URPWM, INPUT); // Enviando comando Ativar modo PWM
  for(int i=0;i<4;i++){ // Loop para iniciar parâmetros da variável definida acima
    Serial.write(EnPwmCmd[i]);
  }
}
void ModoPWM(){ // um puxão baixo no pino COMP / TRIG provocando uma leitura do sensor
  digitalWrite(URTRIG, LOW);
  digitalWrite(URTRIG, HIGH); // leitura / Pin PWM pulsos de saída
  distanciaMedida=pulseIn(URPWM,LOW); // Leitura do valor de pulso de entrada
  if(distanciaMedida==50000){ // a leitura é inválido.
    Serial.print("Invalid");
  }
  else{
    distancia=distanciaMedida/50; // todos os níveis 50us baixo significa um centímetro, (conversão
    para centímetros)
  }
  Serial.print("Distance=");
  Serial.print(distancia);
  if( distancia < 95){ //Verificador para impedir picos de leitura e acionar irregularidades do motor,
  aumentando a precisão
    int servoAngulo = map(distancia, 5, 95, 0, 180); //Mapeamento da leitura do sensor para um valor
    compreendido na faixa do servo motor em angulo de 0 – 180 graus
    if(servoAngulo < anteriorMenor || servoAngulo > anteriorMaior){ //Compara o valor de leitura com o
    menor e maior valor de flag, esse teste aumenta a precisão do motor e diminui instabilidades
    servo1.write(servoAngulo); //Escreve o valor de angulação para ativação do motor servo
    anteriorMenor = servoAngulo – 10; //Atualiza as flags para a leitura do motor
    anteriorMaior = servoAngulo + 10;
  }
}
  Serial.println("cm");
}

```

Basta fazer upload do código abaixo e fazer os testes aproximando a mão do sensor.

Para maior entendimento do funcionamento do sensor URM37: acesse o site:

[http://www.dfrobot.com/wiki/index.php?title=URM37 V3.2 Ultrasonic Sensor \(SKU:SEN0001\)](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))
([http://www.dfrobot.com/wiki/index.php?](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))
[title=URM37 V3.2 Ultrasonic Sensor \(SKU:SEN0001\)](http://www.dfrobot.com/wiki/index.php?title=URM37_V3.2_Ultrasonic_Sensor_(SKU:SEN0001))).

Qualquer dúvida poste nos comentários.

Obrigado

Com as etiquetas :

37,
arduino,
Centimetros,
codigo,
dados,
Esquema,
Leitor,
Leitura,
Ligações,
Montagem,
Projeto,
Sensor,
Ultrasson,
Ultrassonico,
URM,
URM37



Publicado por schmittjoaopedro

Graduado como bacharel em Sistemas de Informação pelo Centro Universitário Católica de Santa Catarina campus Jaraguá do Sul. Formado no Ensino Médio pelo Senai com Técnico em Redes de Computadores Articulado. Atualmente desenvolvedor JEE/Web em Sistemas de Engenharia na WEG. Pesquisador no período de faculdade em Informática pela Católica de Santa Catarina. Contato 47 - 99615 2305 E-mail: schmittjoaopedro@gmail.com Web page: <https://joaoschmitt.wordpress.com/> LinkedIn: <https://www.linkedin.com/in/joao-pedro-schmitt-60847470/> Curriculum lattes: <http://lattes.cnpq.br/9304236291664423> Twitter: @JooPedroSchmitt [Ver todos os artigos de schmittjoaopedro](#)

2 thoughts on “Projeto Sensor URM37 com Arduíno”

Ian villas diz:

24 24+00:00 Julho 24+00:00 2013 às 16:17 | Editar

O codigo esta dando alguns erros.

☐ Responder

schmittjoapedro diz:

5 05+00:00 Agosto 05+00:00 2013 às 13:54 | Editar

Bom dia, que tipo de erros?

☐ Responder

☐