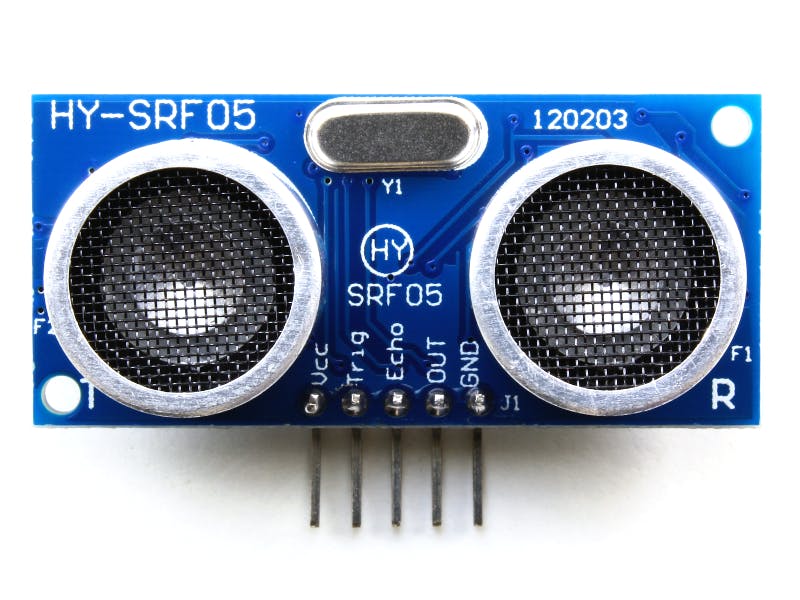
**Sobre o sensor SRF05**



**Descrição**

O **Sensor Ultrassônico de Distância HY-SRF05** foi desenvolvido para aperfeiçoar projetos de robótica, microeletrônica, identificação de presença, é ideal para calcular a distância com precisão de objetos, com operação entre ~2cm à ~450cm.

**Sensor Ultrassônico de Distância HY-SRF05 Shield**

O **Sensor Ultrassônico de Distância HY-SRF05** é composto por um emissor e um receptor, com capacidade de medir distâncias de ~2cm até ~450cm, com uma precisão de aproximadamente 3mm. Este sensor emite sinais ultrassônicos que refletem no objeto a ser atingido e retornam ao sensor, precisando a distância do alvo, tomando o tempo de trânsito do sinal.

A velocidade do sinal ultrassônico emitida pelo **Sensor Ultrassônico de Distância HY-SRF05** corresponde a velocidade do som, que é de aproximadamente 340 m/s, assim, se o sensor estiver a uma distância x do objeto, o sinal percorrerá uma distância equivalente a 2x, ou seja, a onda é enviada pelo sensor e rebatida no obstáculo, logo ela percorre 2 vezes a distância procurada.

O **Sensor Ultrassônico de Distância HY-SRF05** possui um circuito pronto com emissor e receptor acoplados e 5 pinos (VCC, Trigger, ECHO, FORA, GND) para medição, auferindo maiores possibilidades de utilização.

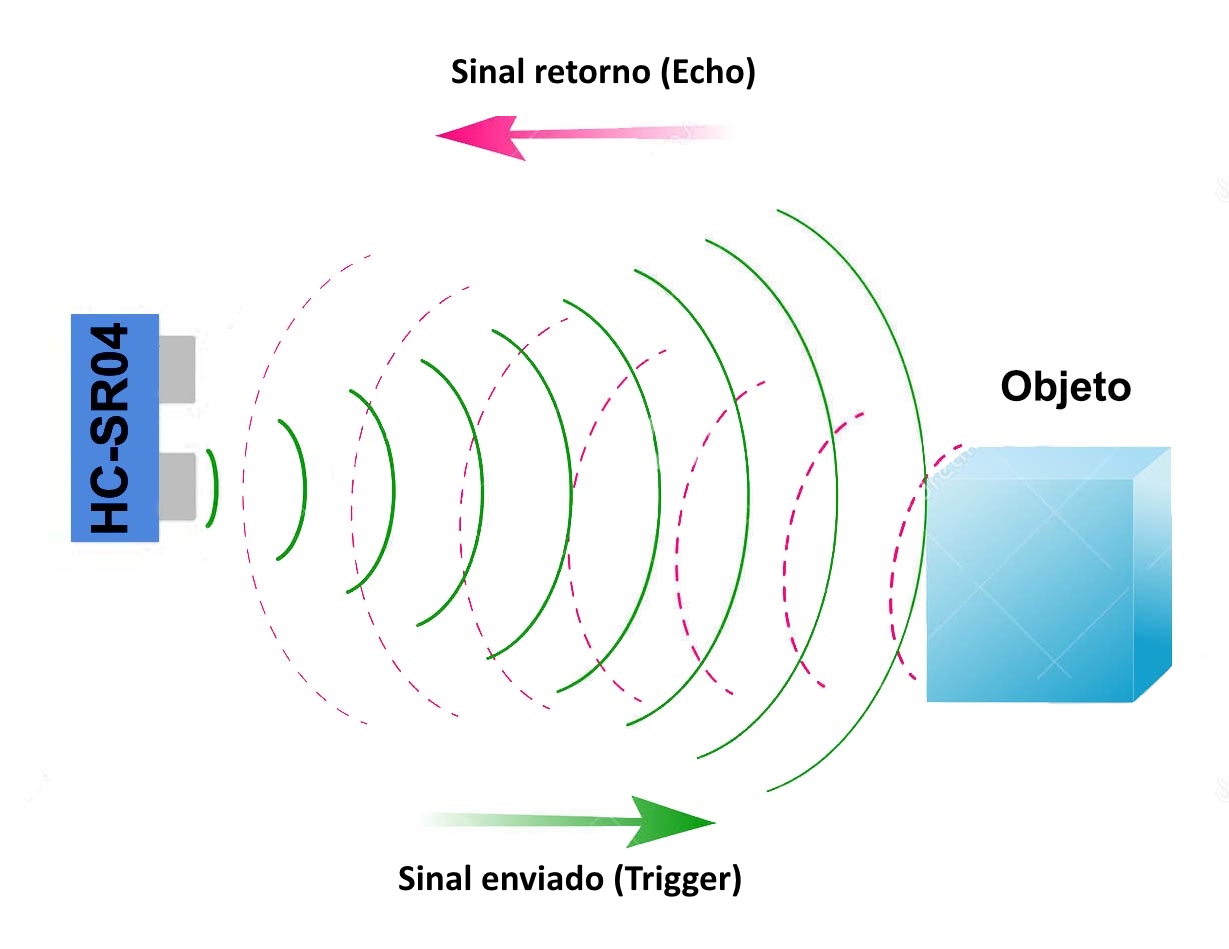
**Características**

* Faixa de detecção: 3 cm-4 m
* Melhor em ângulo de 30graus
* Interface compatível com tijolo eletrônico
* Fonte de alimentação 5VDC
* Placa de ensaio amigável
* Dupla transduce
* Biblioteca pronto
* Tensão de funcionamento: 5 V (DC)
* Corrente de estática: menos de 2mA.
* Sinal de saída: sinal de freqüência elétrica, alto nível de 5 V, baixo nível de 0 V
* Ângulo Sensor: não mais do que 15graus.
* Distância de detecção: 2 cm-450 cm.
* Alta precisão: até 3mm
* Tentador de trabalho:-20C \_ \_ \_ + 60C
* Peso: 10
* Tamanho: 45×21 mm (L x W)
* Conector Pin: 5

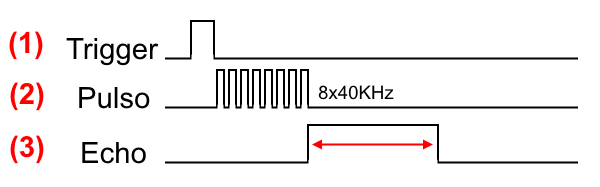
**Pinos:**

1. VCC
2. Trig (T)
3. Echo (R)
4. Fora
5. GND

**Funcionamento**



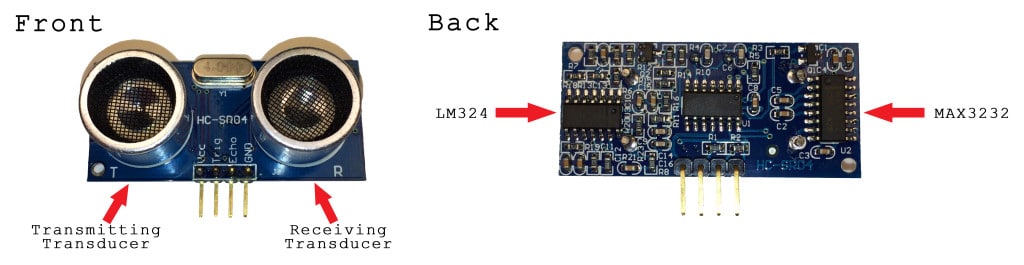
Primeiramente é enviado um pulso de 10µs, indicando o início da transmissão de dados. Depois disso, são enviado 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação ***Distância = (Tempo echo em nível alto \* velocidade do som) /2***



----------------------------------------------------------------------------------------------------------

Testar com Arduino uno: <http://arduinoccbr.blogspot.com/2013/06/ola-neste-artigo-iremos-explicar-como.html>

----------------------------------------------------------------------------------------------------------



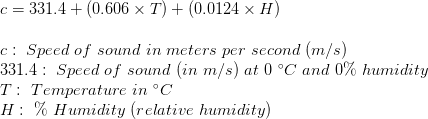
Na parte frontal do telêmetro ultrassônico existem dois cilindros de metal. Estes são transdutores. Os transdutores convertem forças mecânicas em sinais elétricos. No telêmetro ultrassônico, há um transdutor de transmissão e um transdutor de recebimento. O transdutor transmissor converte um sinal elétrico no pulso ultrassônico, e o transdutor receptor converte o pulso ultrassônico refletido novamente em um sinal elétrico. Se você olhar na parte traseira do telêmetro, verá um IC atrás do transdutor de transmissão chamado MAX3232. Este é o IC que controla o transdutor de transmissão. Atrás do transdutor receptor está um IC rotulado como LM324. Este é um Op-Amp quad que amplifica o sinal gerado pelo transdutor receptor em um sinal forte o suficiente para transmitir ao Arduino.

Para iniciar uma medição de distância, precisamos enviar um sinal alto de 5V ao pino Trig por pelo menos 10 µs. Quando o módulo recebe esse sinal, ele emitirá 8 pulsos de som ultrassônico a uma frequência de 40 KHz do transdutor transmissor. Em seguida, aguarda e ouve no transdutor receptor o sinal refletido. Se um objeto estiver dentro do alcance, os 8 pulsos serão refletidos de volta ao sensor. Quando o pulso atinge o transdutor receptor, o pino de eco emite um sinal de alta tensão.

A duração desse sinal de alta tensão é igual ao *tempo total que* os 8 pulsos levam para viajar do transdutor transmissor e voltar ao transdutor receptor. No entanto, queremos apenas medir a distância do objeto, e não a distância do caminho percorrido pelo pulso do som. Portanto, dividimos esse tempo pela metade para obter a variável de tempo na equação d = sxt acima. Como já sabemos a velocidade do (s) som (s), podemos resolver a equação para a distância.

Como a temperatura é uma variável na equação da velocidade do som acima (c = 331,4 + (0,606 x T”celsius”) + (0,0124 x H)), a temperatura do ar ao redor do sensor afeta nossas medições de distância. Para compensar isso, tudo o que precisamos fazer é adicionar um [termistor](https://www.amazon.com/gp/product/B018QL5LPI/ref=as_li_qf_asin_il_tl?ie=UTF8&tag=circbasi-20&creative=9325&linkCode=as2&creativeASIN=B018QL5LPI&linkId=519fc4970f527fe4f5484e0b63e8e1a9" \t "_blank" \o "Termistor de 100K Ohm)  ao nosso circuito e inserir suas leituras na equação. Isso deve fornecer maior precisão às nossas medições de distância. Um termistor é um resistor variável que altera a resistência com a temperatura.





c = 331.4 + (0.606 \times 20) + (0.0124 \times 50)\\  c = 344.02 \ m/s 

**Referências**

<http://www.circuitbasics.com/how-to-set-up-an-ultrasonic-range-finder-on-an-arduino/>

**Instalando e fazendo funcionar a biblioteca do HC-SR04 no Matlab**

No matlab vá em Add-Ons e instale Legacy HC-SR04 Add-On Library for Arduino. Siga o tutorial <https://www.mathworks.com/matlabcentral/fileexchange/57898-legacy-hc-sr04-add-on-library-for-arduino>.

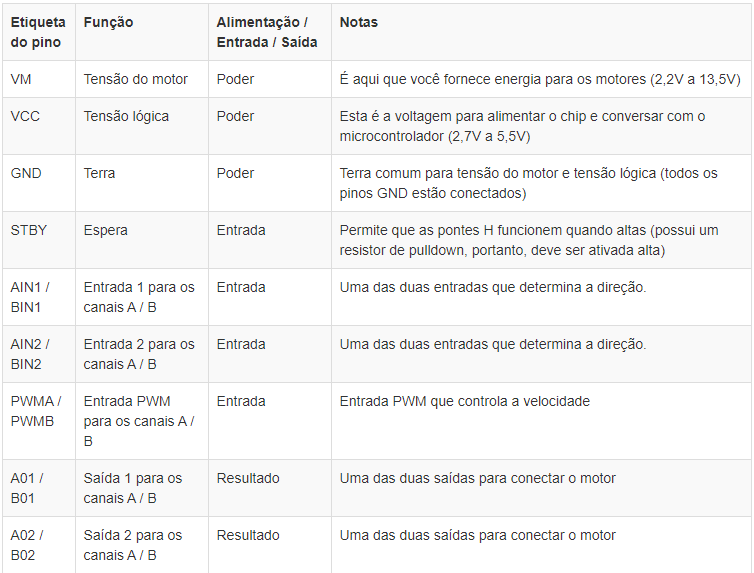
Após a instalação conforme o tutorial adicione a biblioteca [ttps://github.com/JRodrigoTech/Ultrasonic-HC-SR04/archive/master.zip](https://github.com/JRodrigoTech/Ultrasonic-HC-SR04/archive/master.zip) na pasta C:\ProgramData\MATLAB\SupportPackages\R2018a\**3P.instrset\arduinoide.instrset\arduino-1.8.1\libraries.** Pronto, use a vontade.

<http://tixplicando.blogspot.com/2015/10/criando-um-robo-para-derrubar-obstaculos.html>

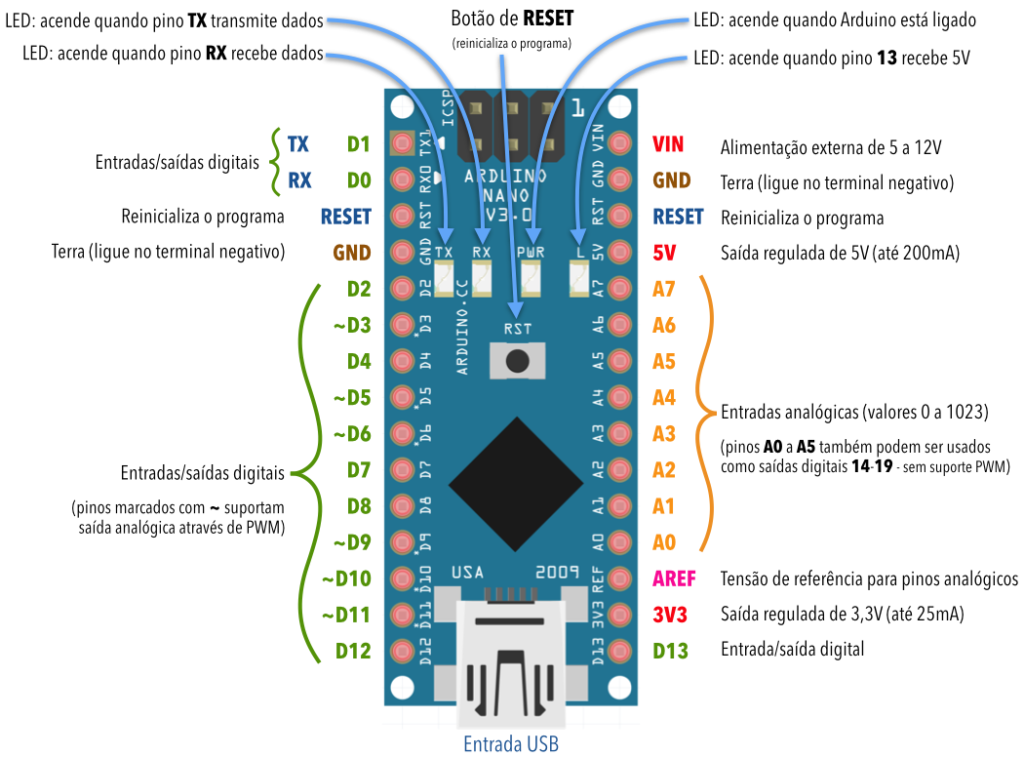
**Driver TB6612FNG**

<http://tixplicando.blogspot.com/2015/10/criando-um-robo-para-derrubar-obstaculos.html>

<https://learn.sparkfun.com/tutorials/tb6612fng-hookup-guide/all> :



**Arduino Nano**



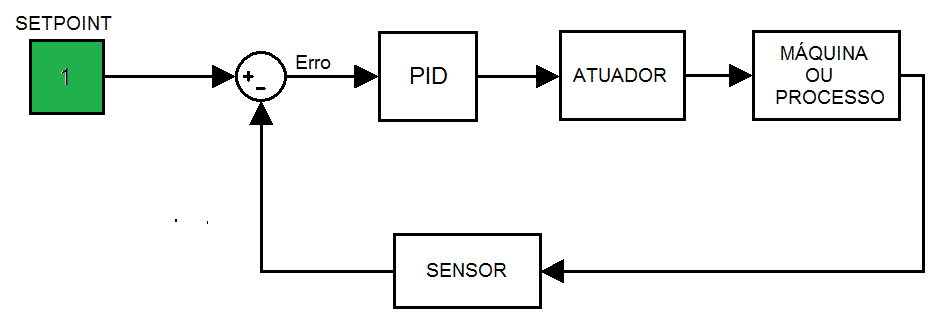
Para rodar o scket precisa baixar a placa atmel m32p n Arduino

------------------------------------------------------------------------------------------------------------------------------

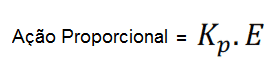
PID com Arduino, bom vídeo: <https://www.youtube.com/watch?v=txftR4TqKYA>

<http://professor.pucgoias.edu.br/SiteDocente/admin/arquivosUpload/18706/material/Lab%20Projeto.pdf>

<https://www.embarcados.com.br/controle-pid-em-sistemas-embarcados/>



A ação proporcional trabalha corrigindo o erro do sistema, multiplicando o ganho proporcional pelo erro, dessa forma agindo com uma maior amplitude de correção a fim de manter a estabilidade da variável.



Para corrigir o efeito windup o controlador PID deve possuir em seu algoritmo rotinas de "reset" da ação integral, que impede que o termo integral continue a ser atualizado quando a saída atinge seu limite máximo.

A ação derivativa tem sua resposta proporcional à taxa de variação da variável do processo, aumentando a velocidade de resposta do sistema caso a presença do erro seja detectada.

------------------------------------------------------------------------------------------------------------------------------

“Documentação da librarie: ”<http://brettbeauregard.com/blog/2011/04/improving-the-beginners-pid-introduction/>