

Instalando sensor ultrassônico de distância na Beaglebone Black

Olá pessoal, neste tutorial vou mostrar uma forma fácil e rápida de instalar um sensor ultrassônico de distância na [Beaglebone Black](#), e como usar o sistema de arquivos do [Linux](#) para ler os valores adquiridos pelo conversor A/D, e também como isso poderá ser feito pela sua aplicação (aqui vamos exemplificar utilizando Python).

Para demonstrar vou utilizar o sensor de distância ultrassônico **LV-MaxSonar-EZ0** que, entre outras interfaces de comunicação, possui uma saída analógica que entrega uma tensão proporcional à distância do sensor, até a superfície que ele estiver apontando.



Dimensões:

Tamanho: 0,870 "x 0,786" x 0,645 "

Peso: 4,3 g



Especificações Gerais

Resolução:	2,5 cm
Alcance máximo:	6.45 m
A faixa mínima:	0 m ¹
Taxa de amostragem:	20 Hz
baud:	9600
Tensão mínima:	2.5 V
Tensão máxima de operação:	5.5 V
Corrente de alimentação:	2 mA

Este compacto sonar da Maxbotix consegue detectar objetos até 6,45m de distância com uma resolução de 2,5cm. Diferente de outros sensores este sonar não tem zona morta, ou seja, pode detectar objetos muito pequenos e até muito próximos.

O VL-MaxSonar oferece 3 tipos de saídas: largura de pulso digital, saída analógica, saída digital serial assíncrona. O sensor indicará na saída 15cm quando houver objetos entre 0-15cm.

Para que serve um sensor ultrassônico?

Um sensor ultrassônico serve para detectar obstáculos e medir distâncias e tem inúmeras aplicações. Quer saber tudo sobre ele? Chega mais.

Talvez você não o conheça pelo nome, mas provavelmente já ouviu em ação! Eles estão presentes nos famosos “**sensores de estacionamento**” dos automóveis mais modernos. Aqueles que apitam quando você está perto demais de uma parede, outro veículo ou obstáculo. Além de detectar a presença de uma barreira, eles também conseguem medir a distância do obstáculo. Por isso, além dos carros, também são muito usados em fábricas, processos de automação industrial, robótica e até na medicina.

Como funcionam os sensores ultrassônicos?

O **sensor ultrassônico** é composto de um emissor e um receptor de ondas sonoras. Podemos compará-los a um alto-falante e um microfone trabalhando em conjunto. Entretanto, ambos trabalham com ondas de altíssima frequência, na faixa dos **40.000 Hz** (ou 40KHz). Isto é muito, muito acima do que os nossos ouvidos são capazes de perceber. O ouvido humano consegue, normalmente, perceber ondas na entre 20 e 20.000 Hz e por isto o sinal emitido pelo sensor ultrassônico passa despercebido por nós.

O sinal emitido, ao colidir com qualquer obstáculo, é refletido de volta na direção do sensor. Durante todo o processo, o aparelho está com uma espécie de “cronômetro” de alta precisão funcionando. Assim, podemos saber quanto tempo o sinal levou desde a sua emissão até o seu retorno. Como a velocidade do som no ar é conhecida, é possível, de posse do tempo que o sinal levou para ir até o obstáculo e voltar, calcular a distância entre o sensor e o obstáculo. Para isto vamos considerar a velocidade do som no ar (340 m/s) na seguinte equação:

$$d = (V * t) / 2$$

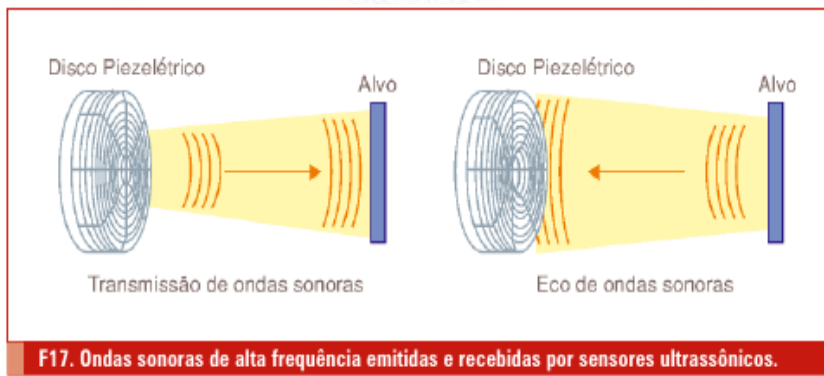
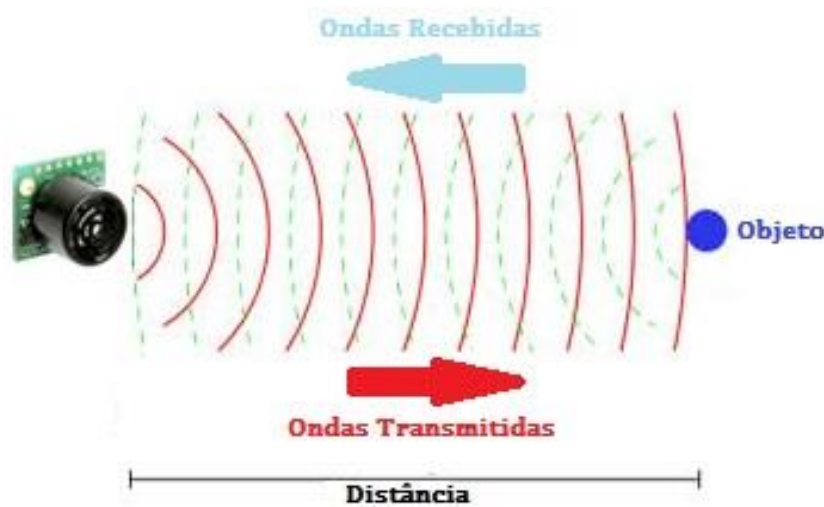
Onde:

d = Distância entre o sensor e o obstáculo (é o que queremos descobrir).

V = Velocidade do som no ar (340 m/s).

t = Tempo necessário para o sinal ir do sensor até o obstáculo e voltar (é o que o nosso módulo sensor ultrassom mede).

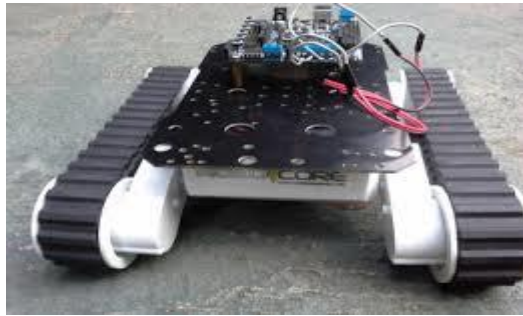
A divisão por dois existe pois o tempo medido pelo sensor é na realidade o tempo para ir e voltar, ou seja, duas vezes a distância que queremos descobrir.



Aplicações:



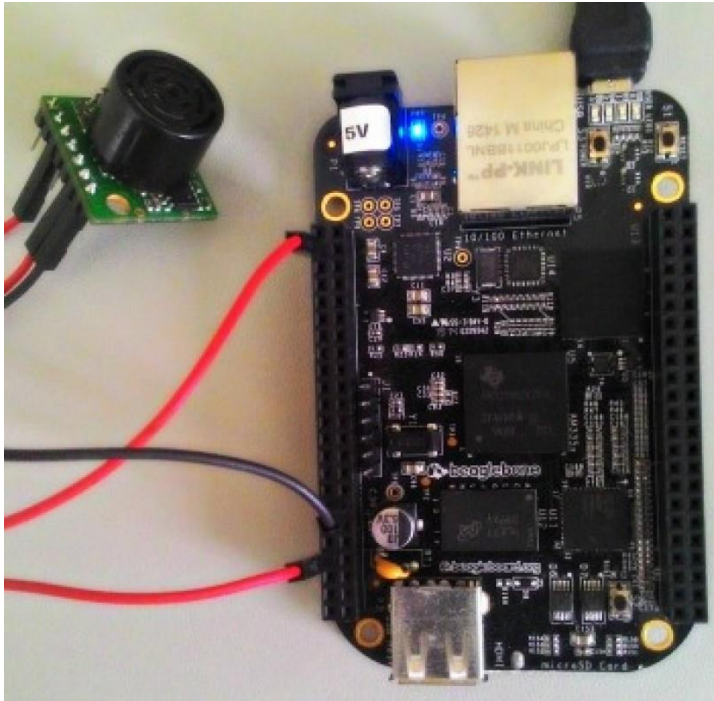
Na indústria



Na robótica



No dia a dia



Conversores analógico/digital são circuitos eletrônicos fundamentais para sistemas embarcados, necessários para transformar sinais contínuos de tensão em representações discretas para que possam ser processados em sistemas digitais.

A **Beaglebone Black** possui sete entradas para o seu conversor A/D de aproximação sucessiva, com resolução de 12 bits, capaz de realizar 200 mil amostras por segundo, e com uma tensão de entrada que pode variar de 0 a 1,8V. Com uma divisão simples então podemos concluir que a **Beaglebone Black** pode detectar variações de tensão de até 0,439mV, suficiente para a maioria das aplicações.

Atenção:

*Não exponha a entrada analógica da **Beaglebone Black** a tensões superiores a 1.8V;*

*Para que o nosso sensor ultrassônico forneça tensões compatíveis com a **Beaglebone Black** tivemos que alimentá-lo com 3,3V;*

*No exemplo estou utilizando a última imagem oficial lançada para a **Beaglebone Black** com **Debian**.*

*Caso você queira clonar o projeto utilizado no exemplo, irá precisar de acesso à internet na **Beaglebone Black**. Isso pode ser feito pela porta USB.*

Caso você execute o comando abaixo, você poderá ver que as entradas analógicas não estão habilitadas.

No Debian

[Clique aqui para saber como instalar sistema debian na beaglebone black](#)

Verificando os slots

```
cat /sys/devices/bone_capemgr.*/slots
```

```
root@beaglebone:~# cat /sys/devices/bone_capemgr.*/slots
0: 54:PF---
1: 55:PF---
2: 56:PF---
3: 57:PF---
4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
root@beaglebone:~# █
```

Como a [Beaglebone Black](#) já possui um device tree overlay (DTO) para o conversor A/D pronto para uso, podemos habilitar as entradas analógicas facilmente apenas com o comando abaixo:

Conversor A/D

```
echo BB-ADC > /sys/devices/bone_capemgr.*/slots
```

```
root@beaglebone:~# cat /sys/devices/bone_capemgr.*/slots
0: 54:PF---
1: 55:PF---
2: 56:PF---
3: 57:PF---
4: ff:P-O-L Bone-LT-eMMC-2G,00A0,Texas Instrument,BB-BONE-EMMC-2G
5: ff:P-O-L Bone-Black-HDMI,00A0,Texas Instrument,BB-BONELT-HDMI
7: ff:P-O-L Override Board Name,00A0,Override Manuf,BB-ADC
root@beaglebone:~# █
```

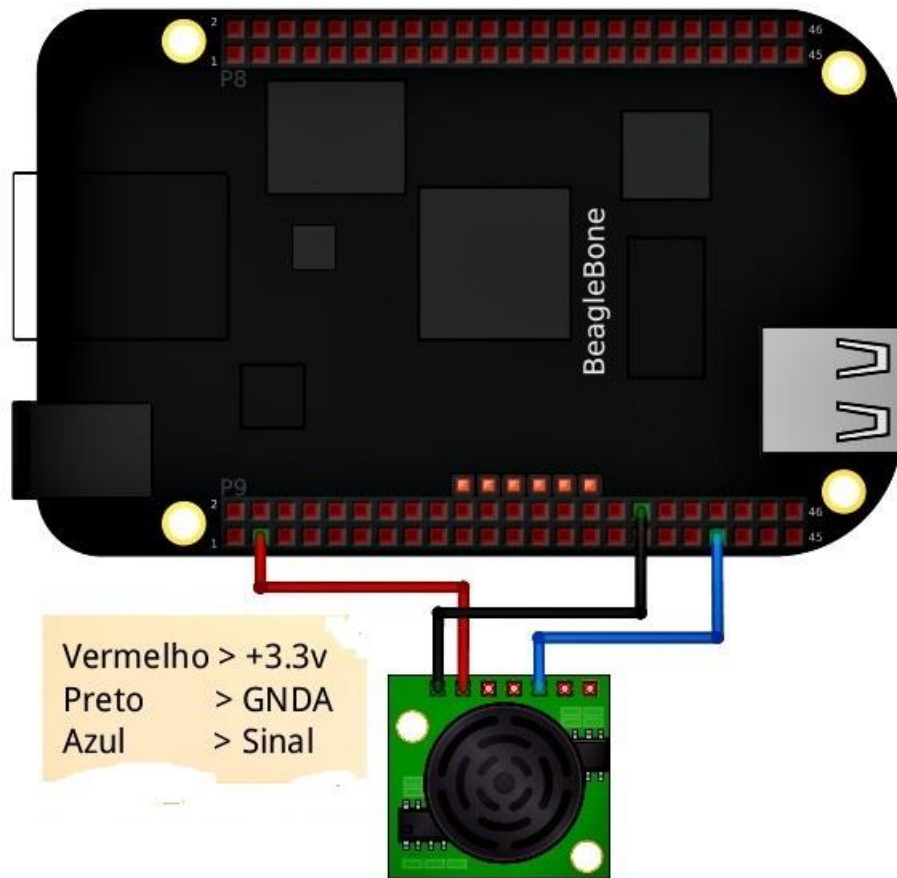
Nova verificação dos slots.

Agora conectando o sensor na [Beaglebone Black](#) conforme:

O VDD do sensor ultrassônico no +3,3V da [Beaglebone Black](#) (P9_03);

O GND do sensor ultrassônico no GNDA_ADC da [Beaglebone Black](#) (P9_34);

O sinal analógico do sensor ultrassônico na entrada AINO da [Beaglebone Black](#) (P9_39);



Conexões entre o sensor e a [Beaglebone Black](#).

Já podemos fazer nossa primeira leitura utilizando o comando:

Verificando os slots

```
cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
```

```
root@beaglebone:~# cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw  
1086
```

Leitura do conversor A/D.

O valor recebido varia linearmente de 0 a 4095 à medida que a tensão na entrada varia de 0 a +1,8V.

Utilizando [Python](#)

Agora vamos desenvolver um pequeno script em [Python](#) 2.7.3 para realizarmos a leitura do sensor ultrassônico, e exibir o resultado em centímetros na tela. O código está no [GitHub \[4\]](#) e para clonar e rodar o projeto basta utilizar os três comandos seguintes:



Clonando e rodando a aplicação

```
git clone https://github.com/igorTavares/bbb_ultrasonic_python.git
cd bbb_ultrasonic_python
python ultrasonic.py
```

```
root@beaglebone:~/bbb_ultrasonic_python# python ultrasonic.py
Voltage = 0.106787109375
Inchs = 16.5681818182
Centimeters = 42.0831818182
root@beaglebone:~/bbb_ultrasonic_python#
```

As explicações estão todas nos comentários do script, assim acho que fica mais fácil de visualizar o funcionamento do programa.

Aplicação em Python

```
#!/usr/bin/env python
```

```
# Modulo para evocar novos processor
import subprocess
```

```
# Recebe a saída do comando "cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw"
output = subprocess.check_output("cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw", shell=True)
```

```
# Regra de tres para achar o valor em volts
voltage = (float(output)*1.8)/4096.0
```

```
# Converte voltage para string para escrever na tela
print "Voltage = " + str(voltage)
```

```
# Equacao do sensor. Vide datasheet.
inchs = (voltage*512.0)/3.3
```

```
# Converte inchs para string para escrever na tela
print "Inchs = " + str(inchs)
```

```
# Converte polegadas para centimetros
centimeters = inchs*2.54
```

```
# Converte centimeters para string para escrever na tela
print "Centimeters = " + str(centimeters)
```

Pronto galera agora é só soltar a imaginação.