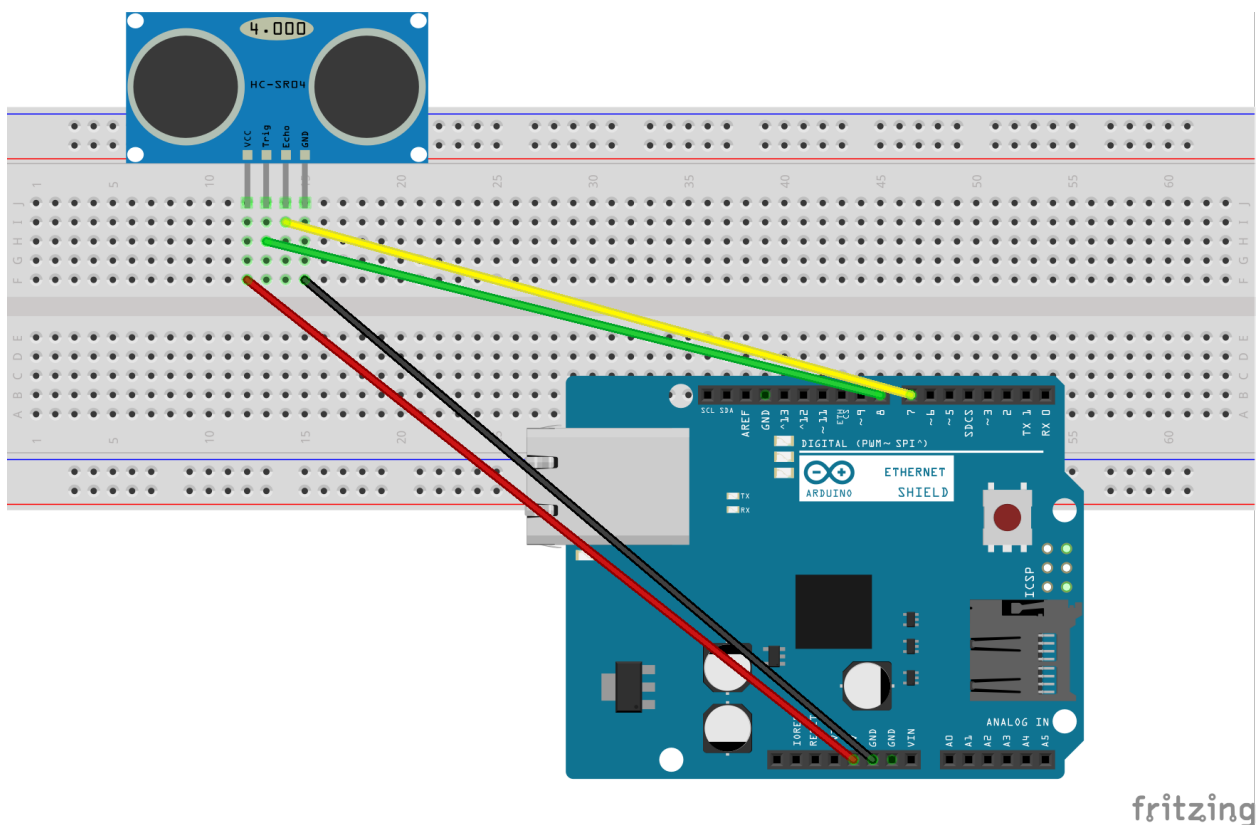


## Sensor de distância ultrassônico – SR-04

Objetivo: controlar o sensor ultrassônico SR-04 de forma a medir distâncias.

Como funciona: o sensor possui dois elementos ultrassônicos, sendo um deles o emissor e o outro o receptor. Chamamos de ultrassom as frequências que estão acima daquelas que podem ser ouvidas pelo ser humano normal, em geral aceita como de 20kHz (ou 20.000 Hertz). Portanto, frequências acima de 20.000 Hz são genericamente chamadas de ultrassom. Nosso módulo sensor utiliza frequência de aproximadamente 40kHz. Quando o pino de controle identificado como *Trigger* (gatilho) fica em nível alto por pelo menos 10us (microsegundos), o módulo envia um ‘trem’ (*burst*) de 8 pulsos pelo emissor e aguarda detecção do retorno deles pelo receptor (identificado como *Echo*, eco).

Como ligar:



O pino VCC do módulo é ligado ao +5V do Arduino

O pino GND do módulo é ligado ao pino GND do Arduino

O pino Trigger do módulo foi conectado ao pino 3 do Arduino (se você escolher outro, troque o código)

O pino Echo do módulo foi conectado ao pino 4 do Arduino (se você escolher outro, troque o código)

Para controlar o sensor, de acordo com a folha de dados (datasheet), precisamos:

- colocar o pino Trigger em nível alto por pelo menos 10uS
- ler o tempo de retorno no pino Echo
- dividir o tempo de retorno por 58 para saber a distância em centímetros (ou, por 148 para saber em polegadas)

```

//pinos em que está ligado o sensor
int pinoTrigger = 3; //vai ligar no pino trigger do sensor, utilizado para
                        enviar um pulso ultrassônico
int pinoEcho = 4; //vai ligar no pino echo do sensor, utilizado para ler a
                        reflexão do pulso enviado

void setup()
{
    Serial.begin(9600); // para usar o monitor serial
    //configurar os pinos para ligação do sensor
    pinMode(pinoTrigger, OUTPUT); //liga o pino do trigger como saída, para enviar
                                    um sinal
    pinMode(pinoEcho, INPUT); //liga o pino do echo como entrada, para receber o
                                    sinal refletido
}

//a função 'mede' aciona o sensor e mede o tempo de resposta, retornando a
    medida em centímetros
float mede()
{
    //primeiramente, desliga a emissão de ultrassom
    digitalWrite(pinoTrigger, LOW);
    //aguarda 0,000003 segundos (3 microsssegundos)
    delayMicroseconds(3);
    //coloca em alto o pino trigger, o que faz com que o sensor emita ultrassom
    digitalWrite(pinoTrigger, HIGH);
    //aguarda 0,000010 segundos (10 microssegundos), conforme o datasheet
    delayMicroseconds(10);
    //desliga a emissão de ultrassom
    digitalWrite(pinoTrigger, LOW);
    // aguarda pelo echo; tempoUs corresponde ao tempo em microssegundos que
        demorou para o sinal voltar
    float tempoUs = pulseIn(pinoEcho, HIGH); // microseconds
    //conforme o datasheet consultado, divide o tempo por 58 para saber a
        distância em centímetros - se dividir por 148 dá a distância em polegadas
    //retorna o resultado da divisão
    return (tempoUs / 58); // assume velocidade do som no ar de 340 m/s = 2.9ms/mm
}

void loop()
{
    //chama a função de medição de distância e armazena o valor em centímetros que
a função retornou na variável distancia
    float distancia = mede();
    Serial.print("Distancia medida: ");
    Serial.print(distancia); //imprime valor decimal
    Serial.println(" centimetros");
    //aguarda 0,5 segundo antes de medir novamente
    delay(500); // 500 ms; o datasheet recomenda aguardar pelo menos 60ms antes de
        efetuar nova medição
}

```

Monte o circuito sugerido, compile o programa e teste. Ligue o monitor serial para ver o resultado da medição.

**Desafio:** como fazer e o que ajustar no código para efetuar a medição de altura de uma pessoa ao invés de uma medição de distância?