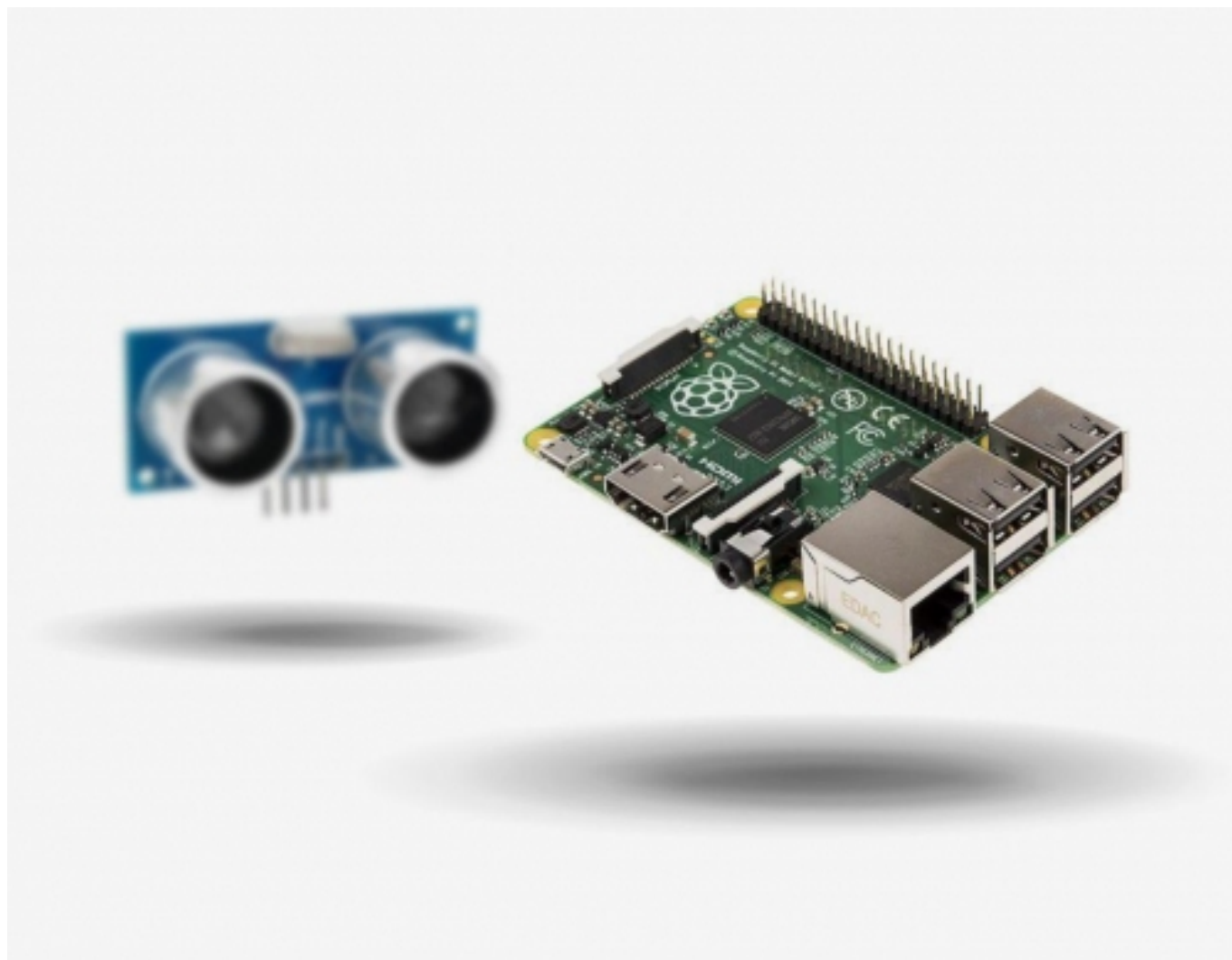


## ساخت فاصله‌سنج رسیبری پای با سنسور تشخیص فاصله آلتراسونیک HC-SR04



برای بسیاری از پروژه‌های در فضای باز، اندازه‌گیری فاصله ضروری و سودمند است. این ماژول‌های کوچک با آلتراسوند فاصله را اندازه‌گیری می‌کنند و به طرز شگفت‌انگیزی دقیق هستند. این آموزش اتصال و کنترل سنسور تشخیص فاصله آلتراسونیک HC-SR04 با استفاده از برد رسیبری پای را نشان می‌دهد.

در آموزش "[راه‌اندازی ماژول فاصله‌سنج آلتراسونیک توسط آردوینو](#)" روابط مربوط به سنسور آلتراسونیک و راه‌اندازی آن با برد آردوینو به‌طور کامل آورده شده است. درواقع در این آموزش راه‌اندازی ماژول آلتراسونیک، راه‌اندازی LCD کارکتری 2 در 16 و فاصله‌سنجی با دقت بالا و در نظر گرفتن عنصر دمایی آورده شده است.

همچنین، ساخت یک فاصله‌سنج توسط آردوینو، سنسور تشخیص فاصله آلتراسونیک SR05، زنگ اخبار و چند ال ای دی را هم می‌توانید در آموزش "[ساخت فاصله‌سنج آردوینو با زنگ اخبار و ال ای دی](#)" مطالعه کنید.

### قطعات موردنیاز

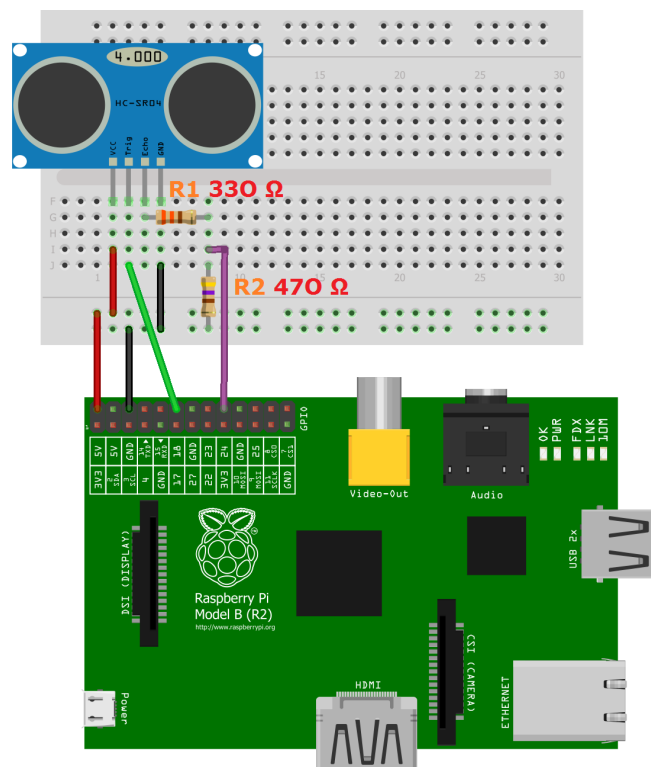
1 عدد	سنسور تشخیص فاصله آلتراسونیک HC-SR04
1 عدد	برد رسیبری پای 2
1 عدد	مقاومت 470 اهم
1 عدد	مقاومت 330 اهم

## شماتیک مدار

چهار عدد پین روی سنسور تشخیص فاصله آلتراسونیک HC-SR04 وجود دارد که به برد رسیبری پای متصل می‌شود:

- VCC به پین شماره 2 (5 ولت) رسیبری پای
- GND به پین شماره 6 (GND) رسیبری پای
- TRIG به پین شماره 18 (GPIO 18) رسیبری پای
- پایه ECHO سنسور را توسط مقاومت 330 اهم به پین شماره 24 (GPIO 24) وصل کنید و توسط مقاومت 470 اهم هم به پین شماره 6 (GND) وصل نمایید.

دلیل این اتصال این است که پایه‌های GPIO فقط ولتاژ 3.3 ولت را تحمل می‌کند. علت اتصال پایه GPIO24 با استفاده از مقاومت به GND هم این است که سیگنال ارسالی به این پین مشخص و ثابت باشد. اگر هیچ پالسی از سوی سنسور ارسال نشود این پایه صفر یا GND است و در غیر این صورت این پایه یک است. اگر این پایه به GND وصل نباشد، زمانی که هیچ سیگنالی ارسال نشود مقدار این پایه نویزی و بین 0 و 1 متغیر است. شماتیک مداری این اتصالات در شکل زیر نمایش داده شده است:



fritzing

## برنامه‌نویسی برای کنترل سنسور

اول از همه، باید کتابخانه GPIO Python نصب شود.

برای استفاده از ماژول، یک اسکریپت جدید ایجاد می‌کنیم:

```
sudo nano ultrasonic_distance.py
```

که شامل متن زیر می‌باشد:

```
Libraries#  
  
import RPi.GPIO as GPIO  
  
import time  
  
(GPIO Mode (BOARD / BCM#  
(GPIO.setmode(GPIO.BCM  
  
set GPIO Pins#  
GPIO_TRIGGER = 18  
GPIO_ECHO = 24  
  
(set GPIO direction (IN / OUT#  
(GPIO.setup(GPIO_TRIGGER, GPIO.OUT  
(GPIO.setup(GPIO_ECHO, GPIO.IN  
  
:()def distance  
set Trigger to HIGH #  
(GPIO.output(GPIO_TRIGGER, True  
  
set Trigger after 0.01ms to LOW #  
(time.sleep(0.00001  
(GPIO.output(GPIO_TRIGGER, False  
  
()StartTime = time.time  
()StopTime = time.time  
  
save StartTime #  
:while GPIO.input(GPIO_ECHO) == 0  
()StartTime = time.time  
  
save time of arrival #  
:while GPIO.input(GPIO_ECHO) == 1  
()StopTime = time.time  
  
time difference between start and arrival #  
TimeElapsed = StopTime - StartTime  
(multiply with the sonic speed (34300 cm/s #  
and divide by 2, because there and back #  
distance = (TimeElapsed * 34300) / 2
```

```
        return distance

    :__if __name__ == '__main__':
        :try:
            :while True:
                ()dist = distance
                (print ("Measured Distance = %.1f cm" % dist
                    (time.sleep(1

Reset by pressing CTRL + C #
:except KeyboardInterrupt:
    ("print("Measurement stopped by User
        ()GPIO.cleanup
```

سپس، دستور زیر را اجرا می‌کنیم:

```
sudo python ultrasonic_distance.py
```

بنابراین، با این برنامه هر ثانیه فاصله اندازه‌گیری می‌شود تا زمانی که کاربر Ctrl+C را بزند و برنامه را متوقف کند. از این برنامه در بسیاری از پروژه‌های کاربردی متفاوت می‌توانید استفاده کنید.

تصاویر و فیلم‌های اجرای این پروژه را با ما به اشتراک بگذارید و پاسخ گوی سؤالات و پیشنهادهای شما عزیزان خواهیم بود.

ترجمه شده توسط تیم الکترونیک صنعت بازار | منبع: سایت [raspberrypi.com](http://raspberrypi.com)