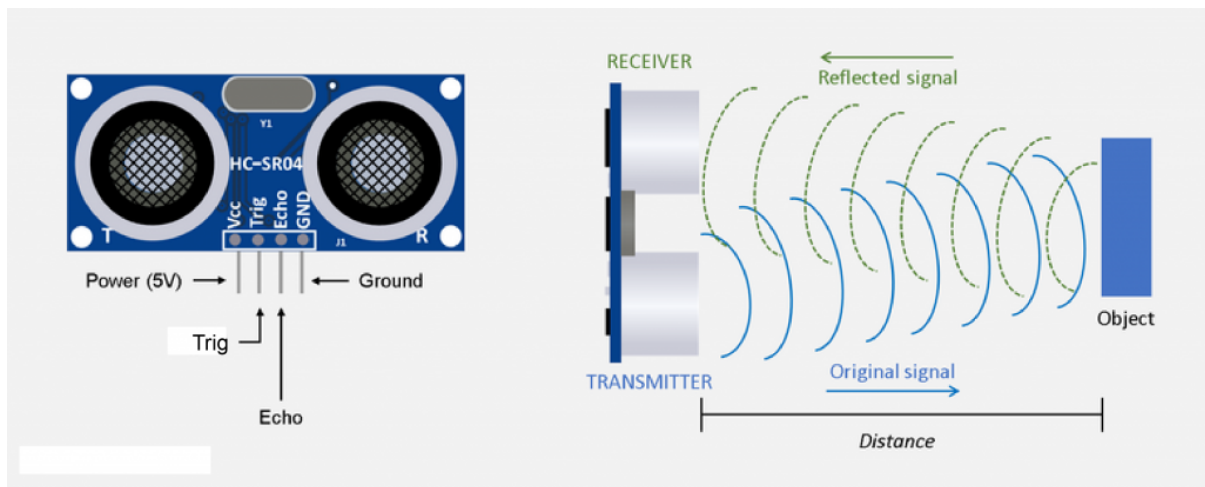
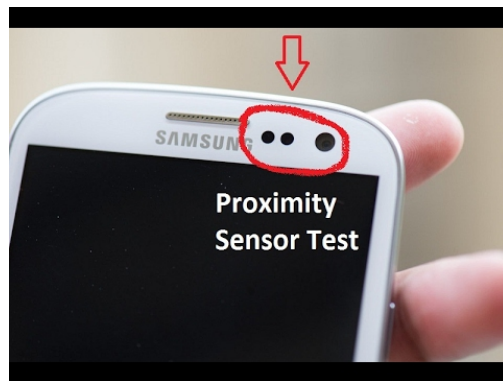
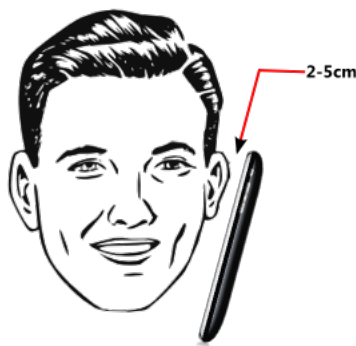


## Raspberry Pi Pico နှင့် HCSR-04 Ultrasonic Sensor တို့အား တွဲဖက်အသုံးပြုခြင်း



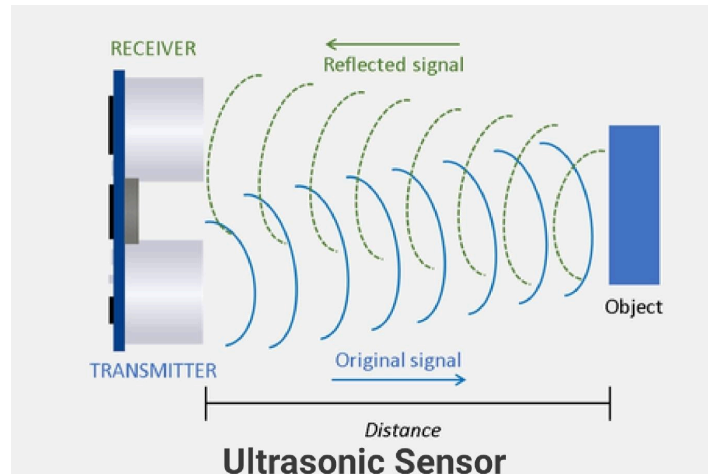
Robot တွေတည်ဆောက်တဲ့နေရာမှာ Ultrasonic Sensor တွေကို အသုံးပြုကြပါတယ်။ Ultrasonic Sensor က သူနဲ့ သူ့ရှေ့မှာရှိနေတဲ့ အရာတစ်ခုကြားထဲက အကွားအဝေး distance ကိုတိုင်းတာနိုင်ပါတယ်။ အဲဒီလိုတိုင်းတာခြင်းအားဖြင့် robot နဲ့အတားအဆီးတစ်ခုခုကြားထဲက အကွားအဝေးကို သိရှိနိုင်မှာဖြစ်ပါတယ်။ Obstacle Avoiding Robot တွေမှာလည်း Ultrasonic Sensor တွေကိုအသုံးပြုကြပါတယ်။



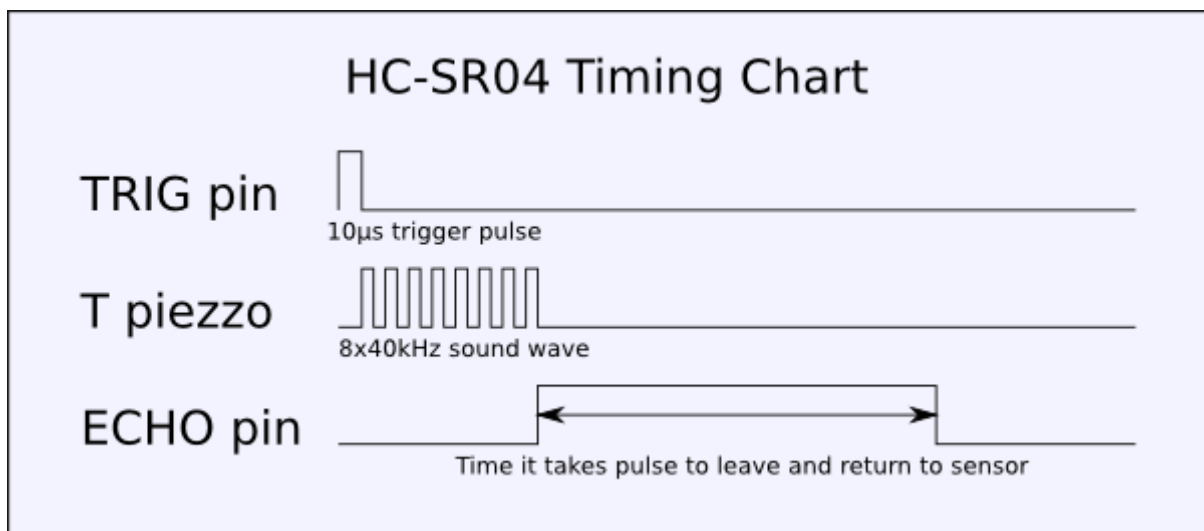
ပုံ() Proximity Sensor in Smart Phones

ကျွန်တော်တို့သုံးနေတဲ့ ဖုန်းထဲမှာလည်း နားနားကပ်ပြီးဖုန်းနားထောင်နေတဲ့အခါမှာ မလိုအပ်တဲ့ ပါဝါကုန်ဆုံးမှုမရှိအောင်နဲ့ touch screen နဲ့ထိမိပြီး မှာမနှိပ်အောင် “Proximity Sensor” ပါဝင်ပါတယ်။ ဒါလည်း နား(object) နဲ့ ဖုန်းကြားထဲက အကွားအဝေးကိုတိုင်းတာပြီး အဲဒီလို ပြဿနာတွေ မဖြစ်အောင် ဖုန်း screen ကို off လုပ်ပေးပါတယ်။

Ultrasonic Sensor ကလူနားနဲ့ မကြားနိုင်တဲ့ 40 kHz ရှိသော ultrasound ကိုထုတ်လွှတ်ပေးပါတယ်။ ထုတ်လွှတ်လိုက်တဲ့ wave က object တစ်ခုနဲ့ တိုက်မိပြီးတော့ echo (ပဲ့တင်သံ) တစ်ခုပြန်လာပါတယ်။ Ultrasonic Sensor ဖြစ်တဲ့အတွက်ကြောင့် လူနားနဲ့တော့ မကြားနိုင်ပါဘူး။ Ultrasonic Sensor က ထုတ်လွှတ်ပေးလိုက်တဲ့ ultrasound က object တစ်ခုနဲ့ သွားတိုက်စေတဲ့ ကြာချိန် “Travel Time” နဲ့ ultrasound အသံရဲ့ speed (“Speed of Sound”) အပေါ်မူတည်စဉ်းစားပြီးတော့ sensor နဲ့ object ရဲ့ ကြားထဲက အကွားအဝေးကိုတွက်ချက်ယူပါတယ်။



HC-SR04 Ultrasonic Sensor မှာ Pin (၄) ခုရှိပါတယ်။ VCC, GND, Trig နဲ့ Echo pin တို့ဖြစ်ပါတယ်။ Trig ကတော့ Trigger Pin ဖြစ်ပါတယ်။ Trigger Pin က 40 kHz ရှိတဲ့ ultrasound ကို ထုတ်လွှတ်ပေးပါတယ်။ ဘယ်လိုထုတ်ပေးလိုက်တာလဲဆိုတာကို ကြည့်ဖို့နဲ့ ပရိုဂရမ်ရေးဖို့ရန်အတွက် HC-SR04 ရဲ့ Timing Chart လေးကို ဖတ်ကြည့်ရအောင်လားဗျာ။



ပထမဦးဆုံးအဆင့် အနေနဲ့ Trigger ကနေ 10 us ကြာတဲ့ pulse တစ်ခုထုတ်လွှတ်ပေးလိုက်ပါတယ်။ ပရိုဂရမ်ရေးတဲ့အခါမှာ 10 us ကြာတဲ့ pulse တစ်ခုကို ထုတ်လွှတ်ဖို့ရန်အတွက် အဲဒီ Trigger Pin ကို “1” ပေးလိုက်ပြီးတော့ delay အနေနဲ့ 10 us ပေးလိုက်မှာဖြစ်ပါတယ်။ ဒါမှသာ Trigger Pin ကနေ 10 us ရှိတဲ့ pulse တစ်ခုထွက်လာမှာဖြစ်ပါတယ်။

၎င်းနောက် 40 kHz ရှိတဲ့ Ultrasound Wave စုစုပေါင်း (၈) ခုထုတ်လွှတ်ပေးလိုက်ပါတယ်။ အဲဒီ Sound Wave သွားနေတဲ့ Speed က “Speed of Sound” ပါ။ (အဲဒီ Speed of Sound ကို အီကွေးရှင်း ထဲထည့်တွက်ပြီးတော့ ပရိုဂရမ်ထဲမှာ ထည့်ရေးရမှာပါ။

Sensor ရှေ့မှာသာ object တစ်ခုရှိနေမယ်ဆိုရင် စောနက လွှတ်ထုတ်လိုက်တဲ့ sound wave တွေက အဲဒီ object တစ်ခုနဲ့ရိုက်ခိုက်မိပြီးတော့ ပဲ့တင်သံ (echo signal) ပြန်ထွက်လာပါတယ်။ အဲဒီ Signal တွေပြန်လာတယ် မလာဘူးဆိုတာကို Ultrasonic Sensor ရဲ့ Echo pin ကနေ စစ်ပေးရ မှာဖြစ်ပါတယ်။ Signal က object ကိုသွားတိုက်ပြီး ပြန်လာတဲ့ကြာချိန်ကို တွက်လိုက်မယ်။ ပြီးတော့ သွားပြီးပြန်လာတဲ့အချိန် (အသွားအပြန်ကြာချိန်) ဖြစ်တာကြောင့် 2 နဲ့ထပ်စားပေးလိုက်ပြီး တော့ sensor နဲ့ object ရဲ့ကြားထဲက အကွာအဝေးကို တွက်ယူနိုင်တဲ့ အီကွေးရှင်းက အောက်ပါ အတိုင်းထွက်လာပါတယ်။

$$\text{Distance} = (\text{time} \times \text{velocity of sound (350 m/s)} / 2)$$

(At 0 degree C) Speed of sound = 331.5 m/s

General formula for speed of sound at certain temperature (Temp = 331.5 + (0.6 x Temp)

(At 30 degree C) Speed of Sound = 350 m/s (nearly)

Convert to microseconds and centimeters :

$$\text{Speed of Sound} = 350 * 100 / 1000000 = 0.035 \text{ cm/us}$$

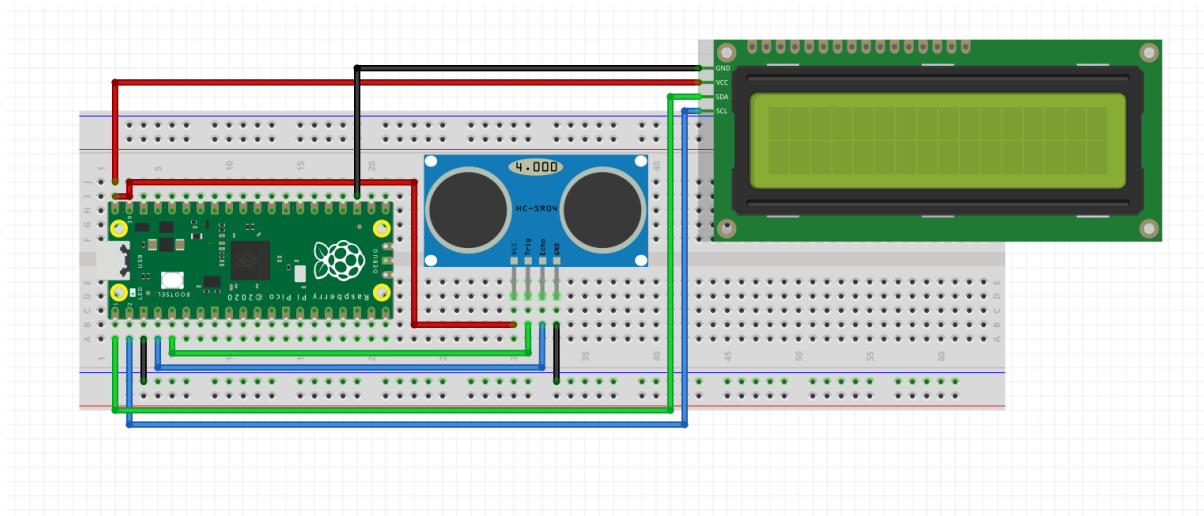
ဒါကြောင့် အပူချိန် ၃၀ ဒီဂရီ ဆဲစီးယက်မှာ အသံဟာ ၁ မိုက်ခရိုစက္ကန့်တွင် 0.035 စင်တီမီတာသွား နိုင်ပါတယ်။

If a signal and its echo take **X** microseconds to do the round trip, then the total distance is converted is –

$$X * 0.035 = X / 28.57$$

Since ultrasound travels from sensor -> object -> sensor. Half is to get to the actual distance = (X/2) / 28.57

Raspberry Pi Pico နှင့် Ultrasonic Sensor (HC-SR04) တို့ပိုင်ယာကြိုးချိတ်ဆက်ရမည့် ပုံစံ -



### simple\_distance.py

```
from machine import Pin
import utime

trigger = Pin(3, Pin.OUT)
echo = Pin(2, Pin.IN)

while True:
    trigger.value(0)
    utime.sleep_us(2)
    trigger.value(1)
    utime.sleep_us(5)
    trigger.value(0)
    while echo.value() == 0:
        signaloff = utime.ticks_us()
    while echo.value() == 1:
        signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    distance = (timepassed * 0.0343) / 2
    print("The distance ",distance,"cm")
    utime.sleep(1)
```

### distance\_lcd.py

```
from machine import I2C, Pin
from lcd_api import LcdApi
from pico_i2c_lcd import I2cLcd
import utime
import time

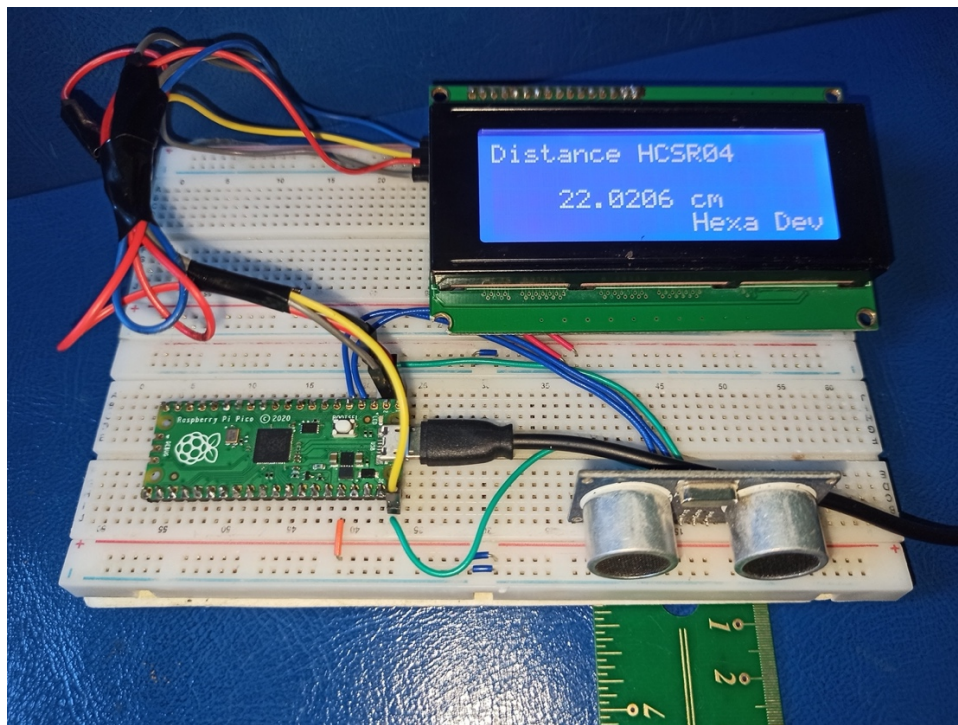
trigger = Pin(3, Pin.OUT)
echo = Pin(2, Pin.IN)
i2c = I2C(0, sda = Pin(0), scl = Pin(1), freq = 400000)
```

```

lcd = I2cLcd(i2c, 0x27, 4, 20)

lcd.move_to(12, 3)
lcd.putstr("Hexa Dev")
while True:
    trigger.value(0)
    utime.sleep_us(2)
    trigger.value(1)
    utime.sleep_us(5)
    trigger.value(0)
    while echo.value() == 0:
        signaloff = utime.ticks_us()
    while echo.value() == 1:
        signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    distance = (timepassed * 0.0343) / 2
    print("The distance ", distance, "cm")
    lcd.move_to(0,0)
    lcd.putstr("Distance HCSR04")
    lcd.move_to(4,2)
    lcd.putstr(str(distance) + " cm ")
    utime.sleep(1)

```



ပုံ() Raspberry Pi Pico နှင့် Ultrasonic Sensor (HCSR-04) တို့အား လက်တွေ့စမ်းသပ်ထားစဉ်

#### Electrical Parameters of HC-SR04

Working Voltage	DC 5V
Working Current	15 mA
Working Frequency	40 kHz
Max Range	400 cm
Min Range	2 cm
Measuring Angle	15 degree
Trigger Input Signal	10us TTL Pulse
Echo Output Signal	Input TTL Level Signal and the range in proportion
Dimension	45 x 20 x 15 mm