```
#Libraries

import RPi.GPIO as GPIO #library for Raspberry Pi GPIOs

import time #library to use sleep function

import board

import digitalio

import adafruit_character_lcd.character_lcd as characterlcd


#GPIO Mode (BOARD / BCM)

GPIO.setmode(GPIO.BCM)


# Modify this if you have a different sized character LCD

lcd_columns = 16

lcd_rows = 2


# Raspberry Pi Pin Config:

lcd_rs = digitalio.DigitalInOut(board.D5)

lcd_en = digitalio.DigitalInOut(board.D6)

lcd_d4 = digitalio.DigitalInOut(board.D12)

lcd_d5 = digitalio.DigitalInOut(board.D13)

lcd_d6 = digitalio.DigitalInOut(board.D16)

lcd_d7 = digitalio.DigitalInOut(board.D17)


# Initialise the lcd class

lcd = characterlcd.Character_LCD_Mono(

    lcd_rs, lcd_en, lcd_d4, lcd_d5, lcd_d6, lcd_d7, lcd_columns, lcd_rows)


#set GPIO Pins

TRIGGER = 19  # board pin as trigger

ECHO = 20     # board pin as echo


#set GPIO direction (IN / OUT)
```

```
GPIO.setup(TRIGGER, GPIO.OUT)

GPIO.setup(ECHO, GPIO.IN)

lcd.clear()


#function distance will use 2 GPIOs to trigger and echo to calculate distance using the distance formula

def distance():

    # set Trigger to HIGH

    GPIO.output(TRIGGER, True)


    # set Trigger after 0.01ms to LOW

    time.sleep(0.00001)

    GPIO.output(TRIGGER, False)


    StartTime = time.time()

    StopTime = time.time()


    # save StartTime

    while GPIO.input(ECHO) == 0:

        StartTime = time.time()


    # save time of arrival

    while GPIO.input(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


#simple if statement
if __name__ == '__main__':
    #simple try exception programming
    try:
        while True:
            dist = distance()#we accept the value in a variable dist
            print ("Measured Distance = %.1f cm" % dist)#display dist
            lcd.clear()
            lcd.message = ("Dist.:%.1f cm" % dist)
            time.sleep(2)


        # Reset by pressing CTRL + C
    except KeyboardInterrupt:
        print("Measurement stopped by User")
        GPIO.cleanup()#finally GPIO cleanup to flush all the buffers of the GPIOs used in this code
```