Code for ultrasonic and mositure sensor

```
import RPi.GPIO as GPIO
import time
import detect as st
GPIO.setmode(GPIO.BCM)

TRIG = 23
ECHO = 24
BUZZ = 17
print ("Distance Measurement In Progress")
channel = 27
buzzer = 17

GPIO.setmode(GPIO.BCM)
GPIO.setup(channel,GPIO.IN)
GPIO.setup(buzzer,GPIO.OUT)

GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)



def get_distance():
    GPIO.output(TRIG, False)
    print ("Waiting For Sensor To Settle")
    time.sleep(1)

    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)
```

```
    while GPIO.input(ECHO)==0:
        pulse_start = time.time()

    while GPIO.input(ECHO)==1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start
    distance = pulse_duration * 17150
    distance = round(distance, 2)

    return distance



def callback(channel):
    if GPIO.input(channel):
        print("Water not Detected!")
        GPIO.output(buzzer,GPIO.LOW)
    else:
        print("Water Detected!")
        GPIO.output(buzzer,GPIO.HIGH)



GPIO.add_event_detect(channel, GPIO.BOTH, bouncetime=300)  # let us know when the pin goes HIGH or LOW
GPIO.add_event_callback(channel, callback)  # assign function to GPIO PIN, Run function on change


try:
    while True:
```

```
        time.sleep(0.5)
        callback(channel)

        print(get_distance())

        if(get_distance() < 40):
            #GPIO.output(BUZZ,GPIO.HIGH)
            #time.sleep(1)
            #GPIO.output(BUZZ,GPIO.LOW)
            #time.sleep(1)
            st.main()
```

except KeyboardInterrupt: # If there is a KeyboardInterrupt (when you press ctrl+c), exit the program

```
    print("Cleaning up!")
    gpio.cleanup()
```

Print results

```
            for c in det[:, 5].unique():
                n = (det[:, 5] == c).sum()  # detections per class
                s += f"{n} {names[int(c)]}{'s' * (n > 1)}, "  # add to string
                text_speech.say(names[int(c)])
                d=ut.get_distance()
                if(d<40):
                    text_speech.say(names[int(c)])
                    text_speech.say(d)
                    text_speech.runAndWait()
```