

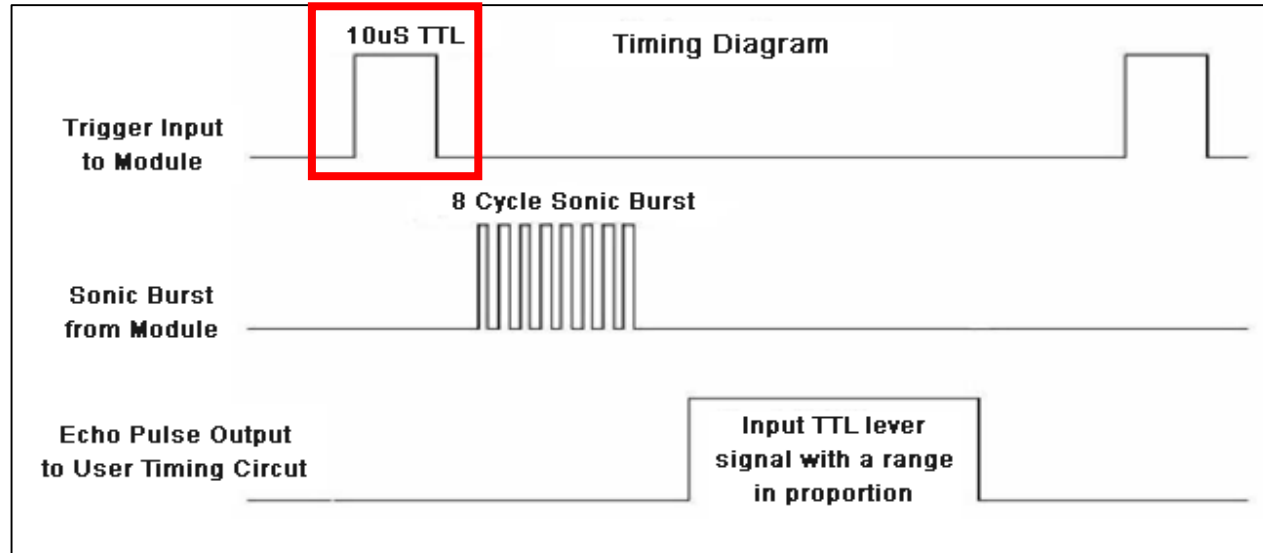
안드로이드_초음파센서



초음파 센서 동작 방식

- 초음파를 발생하고 물체에 부딪혀 돌아오는 시간을 측정함으로써 거리를
센싱
 - Trig 핀 출력 신호와 Echo 핀 입력 신호를 이용
 - 최소 10us HIGH 레벨 신호를 통한 IO trigger(Trig핀이 출력)를 사용하며, 모듈은 자동으로 8번의 40kHz 신호를 보내고 펄스 신호가 돌아오는지 여부를 검출
 - 만약, HIGH 레벨을 통해 신호가 돌아오면(Echo핀이 감지), HIGH 출력 IO 지속 시간은 되돌아 오는 초음파가 보내지는 시간
 - 시험 거리는 다음과 같이 계산
 - 초음파 속도: 340 m/s
 - $\text{us} / 58 = \text{centimeters}$

타이밍 다이어그램



- 센싱을 시작할 때, 트리거 입력에 짧은 10uS 펄스를 제공
- 모듈은 40kHz의 초음파의 버스트(Burst)를 8차례 전송하고, 그것의 에코를 수신
- 전송하는 트리거 신호와 받는 에코 신호 사이의 시간 간격을 통하여 그 범위를 계산
- 센싱이 시작되고 다음 센싱이 시작 되기까지의 간격, 즉, 측정 주기는 60ms 이상으로 사용하며, 이는 에코 신호 중에 트리거 신호가 전달되는 것을 방지하기 위함

라즈베리파이_초음파 센서

```
1. #include <stdio.h>
2. #include <wiringPi.h>
3. #include <unistd.h>

4. #define Trigger 4
5. #define Echo 17

6. float getDistance(void)
7. {
8.     float fDistance;
9.     int nStartTime, nEndTime;

10.    digitalWrite(Trigger, LOW);
11.    delayMicroseconds(2);

12.    digitalWrite(Trigger, HIGH);
13.    delayMicroseconds(10);
14.    digitalWrite(Trigger, LOW);

15.    while(digitalRead(Echo) == LOW);
16.    nStartTime = micros();

17.    while(digitalRead(Echo) == HIGH);
18.    nEndTime = micros();

19.    fDistance = (nEndTime - nStartTime) / 58.0;

20.    return fDistance;
21.}
```

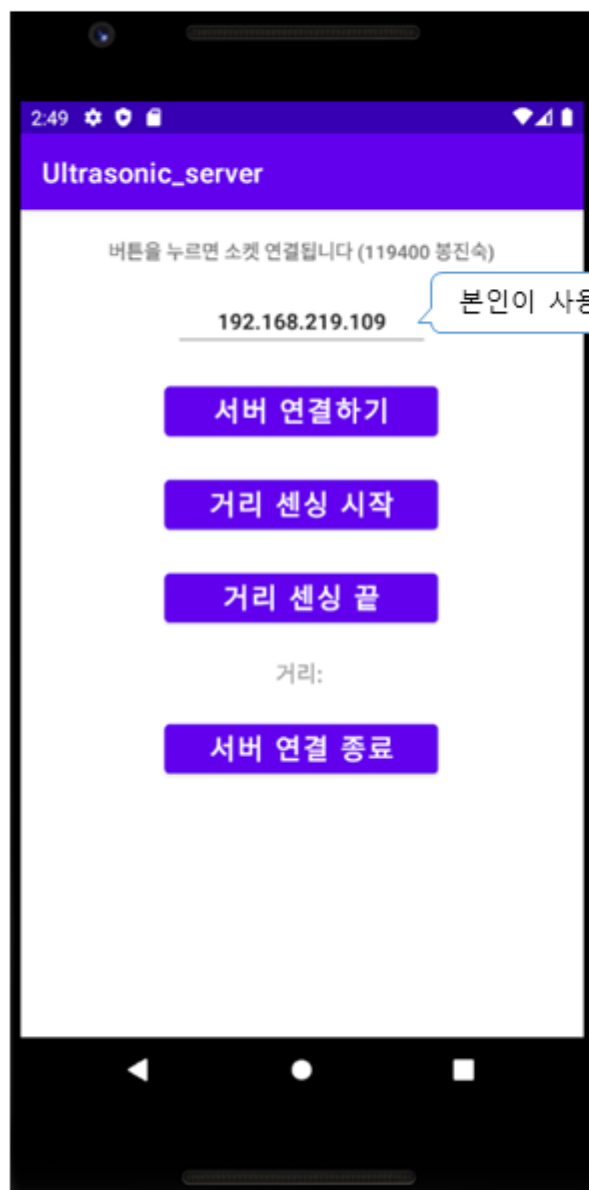
```
21. int main(void)
22. {
23.     if(wiringPiSetupGpio() == -1)
24.         return -1;

25.     pinMode(Trigger, OUTPUT);
26.     pinMode(Echo, INPUT);

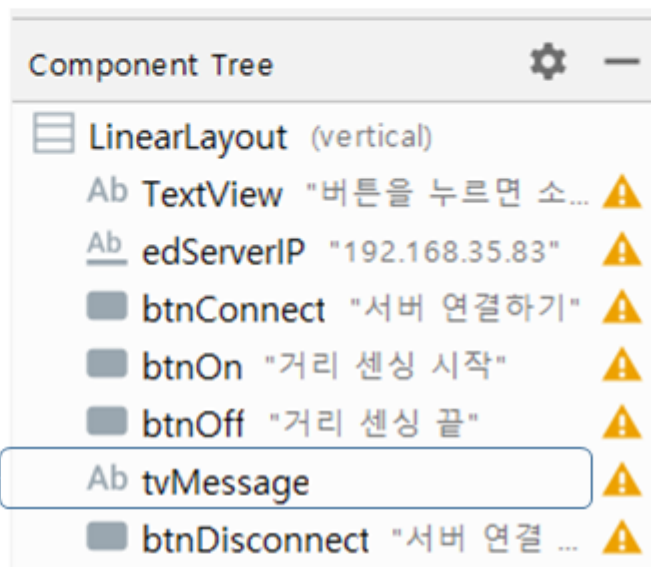
27.     float fDistance = 0;

28.     while(1){
29.         if(digitalRead(Echo) == LOW){
30.             fDistance = getDistance();
31.         }else{
32.             pinMode(Echo, OUTPUT);
33.             digitalWrite(Echo, LOW);
34.             pinMode(Echo, INPUT);
35.         }
36.         printf("Distance : %.2fcm\n", fDistance);
37.         delay(200);
38.     }
39.     return 0;
40. }
```

초음파 센서_클라이언트 (ultrasonic_client)



본인이 사용할 라즈베리파이 IP



<TextView

```
android:id="@+id/tvMessage"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:layout_marginTop="20dp"  
android:hint="거리 : "  
android:textSize="16sp"  
android:textStyle="bold" />
```

초음파 제어_클라이언트 (ultrasonic_client)

```
package com.bong.ultrasonic_client;

import ...

public class MainActivity extends AppCompatActivity {
    Socket socket;
    OutputStream os;
    InputStream is;
    BufferedReader in;
    PrintWriter out;

    ConnectThread thread;

    Button btnConnect, btnOn, btnOff, btnDisconnect;
    EditText edServerIP;
    TextView tvMessage;

    Handler handler = new Handler();

    [ Timer timer;
    boolean threadStop;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvMessage = (TextView) findViewById(R.id.tvMessage);
        edServerIP = (EditText) findViewById(R.id.edServerIP);
        btnConnect = (Button) findViewById(R.id.btnConnect);
        btnDisconnect = (Button) findViewById(R.id.btnDisconnect);
        btnOn = (Button) findViewById(R.id.btnOn);
        btnOff = (Button) findViewById(R.id.btnOff);
```

```
        btnConnect.setEnabled(true);
        btnOn.setEnabled(false);
        btnOff.setEnabled(false);
        btnDisconnect.setEnabled(false);
```

```
        btnConnect.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                String addr = edServerIP.getText().toString().trim();

                thread = new ConnectThread(addr);
                threadStop = false;
                thread.start();

                btnConnect.setEnabled(false);
                btnDisconnect.setEnabled(true);
                btnOn.setEnabled(true);
                btnOff.setEnabled(true);
            }
        });
```

```
        btnDisconnect.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                thread.setStop();

                tvMessage.setText("거리 : ");
                btnConnect.setEnabled(true);
                btnDisconnect.setEnabled(false);
                btnOn.setEnabled(false);
                btnOff.setEnabled(false);
            }
        });
```

초음파 제어_클라이언트 (ultrasonic_client)

```
btnOn.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        btnConnect.setEnabled(false);  
        btnDisconnect.setEnabled(true);  
        btnOn.setEnabled(false);  
        btnOff.setEnabled(true);  
  
        new Thread(){  
            public void run(){  
                timer = new Timer();  
                timer.schedule(new TimerTask() {  
  
                    @Override  
                    public void run() {  
                        out.println("on");  
                        out.flush();  
                    }  
                }, 1000, 1000);  
            }  
        }.start();  
    }  
});
```

```
btnOff.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        btnConnect.setEnabled(false);  
        btnDisconnect.setEnabled(true);  
        btnOn.setEnabled(true);  
        btnOff.setEnabled(false);  
  
        new Thread(){  
            public void run(){  
                timer.cancel();  
                out.println("off");  
                out.flush();  
            }  
        }.start();  
    }  
});
```

초음파 제어_클라이언트 (ultrasonic_client)

//소켓 연결할 스레드 정의

```
class ConnectThread extends Thread {
```

```
    String hostname;
```

```
    public ConnectThread(String addr) {
        hostname = addr;
    }
```

```
    public void run() {
        try {
            int port = 9999;

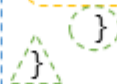
            socket = new Socket(hostname, port);
            os = socket.getOutputStream();
            is = socket.getInputStream();
            in = new BufferedReader(new InputStreamReader(is));
            out = new PrintWriter(os);

            readServer();
        } catch (Exception ex) {
            ex.printStackTrace();
            try {
                socket.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```

```
    public void readServer() {
        try {
            while (!threadStop) {
                String msg1 = in.readLine();

                handler.post(new Runnable() {
                    @Override
                    public void run() {
                        tvMessage.setText("거리: "
                            + msg1 + "cm");
                    }
                });
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
    public void setStop() {
        if (socket.isConnected()) {
            try {
                timer.cancel();
                threadStop = true;
                socket.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}
```



AndroidManifest

```
activity_main.xml x MainActivity.java x AndroidManifest.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.bong.ultrasonic_server">
4      <uses-permission android:name="android.permission.INTERNET"/>
5
6      <application
7          android:allowBackup="true"
8          android:icon="@mipmap/ic_launcher"
9          android:label="Ultrasonic_server"
10         android:roundIcon="@mipmap/ic_launcher_round"
11         android:supportsRtl="true"
12         android:theme="@style/Theme.Ultrasonic_server">
13         <activity
14             android:name=".MainActivity"
15             android:exported="true">
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER" />
20             </intent-filter>
21         </activity>
22     </application>
```