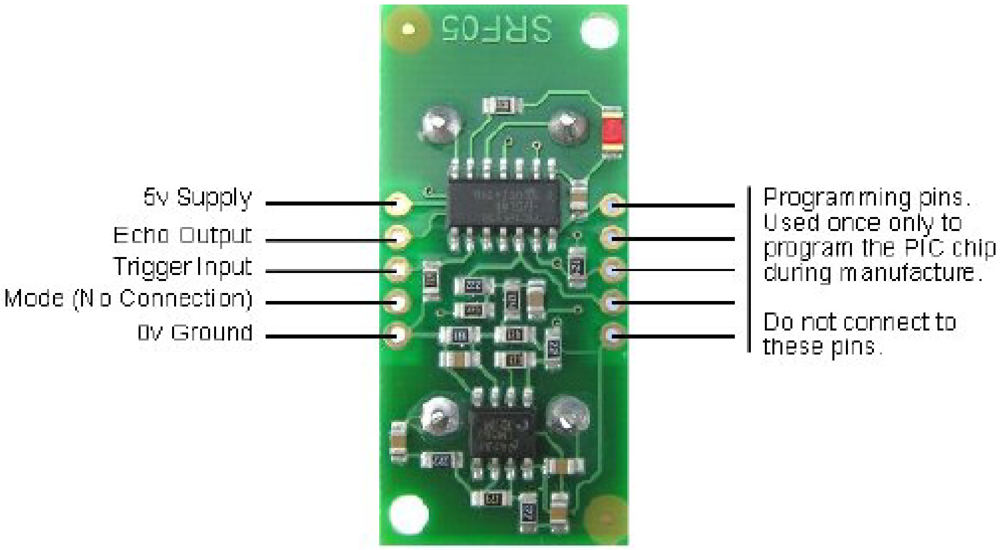
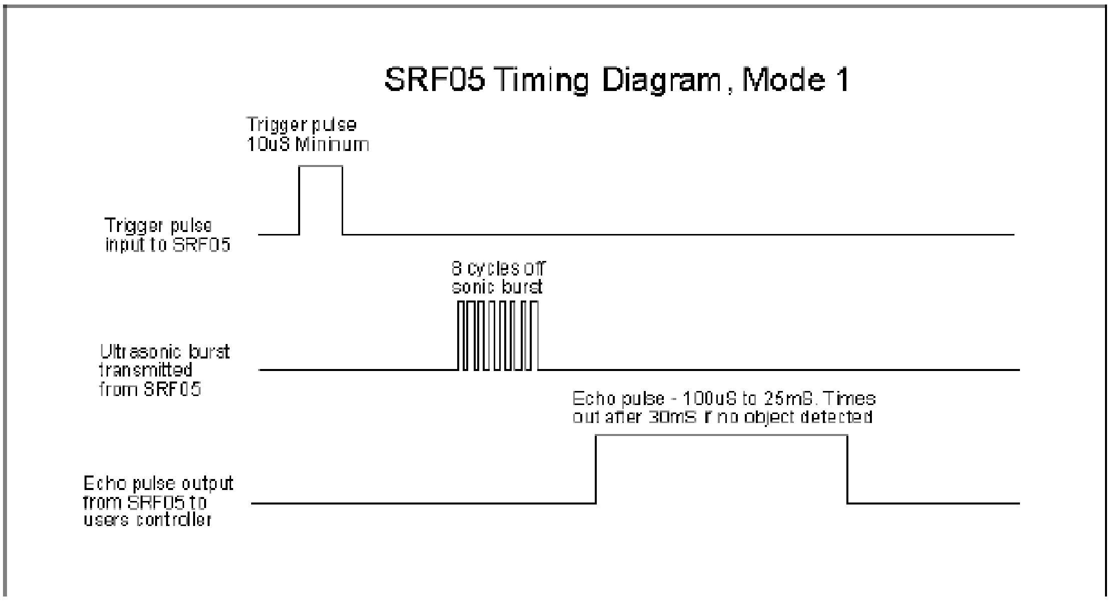
Ultrasonic Sensor

- SRF05
    - SRF04에 비해 3~4 미터 사정거리 증가
    - Mode 1(SRF04와 호환, Separate TRIG, ECHO)

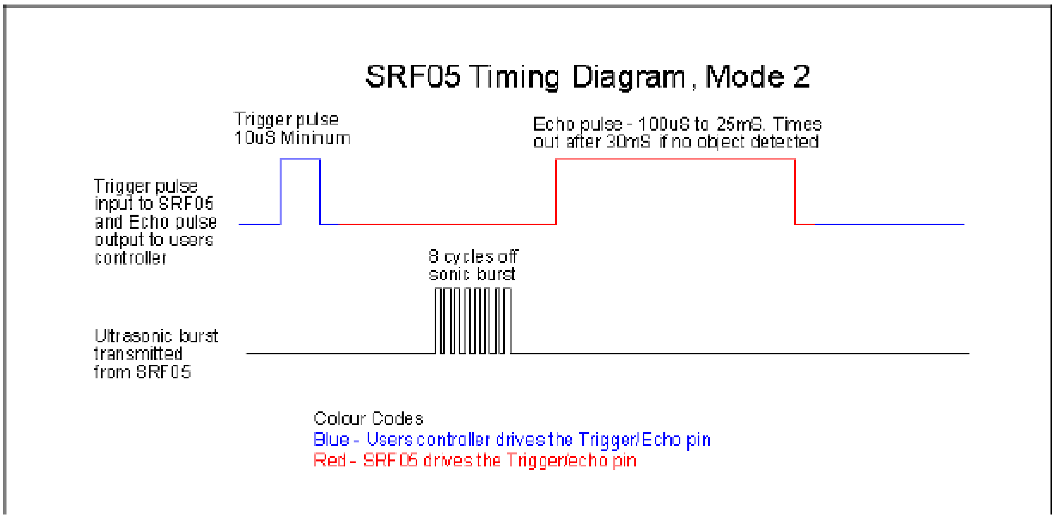

SRF05
5v Supply
Echo Output
Trigger Input
Mode (No Connection)
0v Ground
Programming pins. Used once only to program the PIC chip during manufacture.
Do not connect to these pins.

Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)

        - Uses separate TRIG and ECHO pins, simplest mode to use
        - leave the MODE pin unconnected

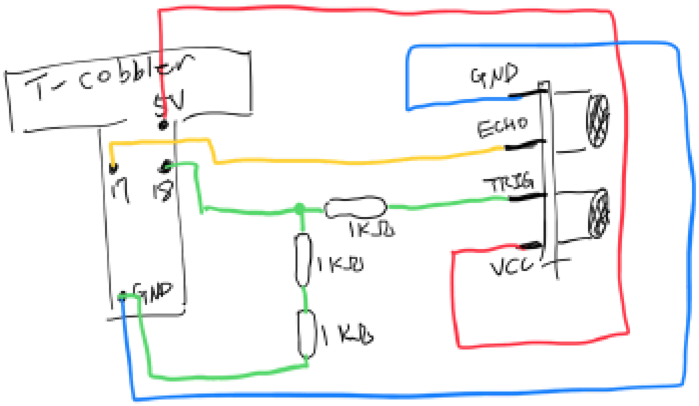

SRF05 Timing Diagram, Mode 1
Trigger pulse 10uS Minimum
Trigger pulse input to SRF05
8 cycles of sonic burst
Ultrasonic burst transmitted from SRF05
Echo pulse - 100uS to 25mS. Times out after 30mS if no object detected
Echo pulse output from SRF05 to users controller

    - Mode 2 : Single pin for both TRIG and ECHO
        - uses a single pin for both TRIG and ECHO signals
        - designed to save valuable pins on embedded controllers
        - connect MODE pin to the 0V GROUND pin
        - ECHO and TRIG appear on same pin


SRF05 Timing Diagram, Mode 2
Trigger pulse 10uS Minimum
Echo pulse - 100uS to 25mS. Times out after 30mS if no object detected
Trigger pulse input to SRF05 and Echo pulse output to users controller
8 cycles of sonic burst
Ultrasonic burst transmitted from SRF05
Colour Codes
Blue - Users controller drives the Trigger/Echo pin
Red - SRF05 drives the Trigger/echo pin

- Calculating Distance
    - Supply short 10uS pulse to the TRIG input to start ranging
    - SRF05 will send out an 8 cycle burst of ultrasound at 40KHz and raise its echo line high, listens for echo.
    - As soon as it detects echo, lowers echo line again
    - ECHO line : a pulse whose width is proportional to the distance to the object
    - Timing the pulse = calculate the range in inches/centimeters
    - If nothing detected, SRF05 will lower its echo line anyways after about 30mS
- SRF04 : Provides echo pulse proportional to distance - width of pulse is measured in uS
    - divide by 58 will give cm result
    - divide by 148 will give inches result
- SRF05 : can be triggered as fast as every 50ms, or 20 times each second
    - wait 50ms before next trigger, even srf05 detecs a close object and the echo pulse is shorter
    - this is to ensure ultrasonic beep has faded away and will not cause a false echo or the next ranging
- 남은 5개 핀에 아무것도 연결하지 말 것(Flash Memory 관련된 일 할 때 사용)
- Changing beam pattern and beam width
    - 안됨

Prototype Circuit



Code

```
# uss_test.py
```

```python
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

TRIG = 11
ECHO = 12
/*
count = 0
success = 0
LIMIT = 101
MAX_SUCCESS = 152
MIN_SUCCESS = 148
*/

def setup():
    GPIO.setmode(GPIO.BOARD)
    GPIO.setup(TRIG, GPIO.OUT)
    GPIO.setup(ECHO, GPIO.IN)

def distance():
    GPIO.output(TRIG, 0)
    time.sleep(0.000002)

    GPIO.output(TRIG, 1)
    time.sleep(0.00001)
    GPIO.output(TRIG, 0)

    while GPIO.input(ECHO) == 0:
        a = 0

    time1 = time.time()

    while GPIO.input(ECHO) == 1:
        a = 1

    time2 = time.time()

    during = time2 - time1
    return during * 340 / 2 * 100

def loop():
    while True:
        dis = distance()
        print dis, 'cm'
        time.sleep(0.05) # Q3 : Sleep time change(50ms)
    /*  # Q4
    global count
    while count < (LIMIT+1):
        dis = distance()
        measure(dis)
        print dis, 'cm'
        time.sleep(0.05)
        count += 1
    */

def destroy():
    GPIO.cleanup()
```

```
/*
def measure(dis):
    global success
    if (dis > MIN_SUCCESS) and (dis < MAX_SUCCESS):
        success += 1
*/

if __name__ == "__main__":
    setup()
    try:
        loop()
    except KeyboardInterrupt:
        destroy()
```

```
/********************************
* Ultra Sonic Raning module Pin VCC should
* be connected to 5V power.
******************************/
#include <wiringPi.h>
#include <stdio.h>
#include <sys/time.h>
#define Trig 0
#define Echo 1

void ultraInit(void)
{
    pinMode(Echo, INPUT);
    pinMode(Trig, OUTPUT);
}

float disMeasure(void)
{
    struct timeval tv1;
    struct timeval tv2;

    long time1, time2;
    float dis;

    digitalWrite(Trig, LOW);
    delayMicroseconds(2);

    digitalWrite(Trig, HIGH);
    delayMicroseconds(10); //发出超声波脉冲
    digitalWrite(Trig, LOW);

    while(!(digitalRead(Echo) == 1));
    gettimeofday(&tv1, NULL); //获取当前时间

    while(!(digitalRead(Echo) == 0));
    gettimeofday(&tv2, NULL); //获取当前时间

    time1 = tv1.tv_sec * 1000000 + tv1.tv_usec; //微秒级的时间
    time2 = tv2.tv_sec * 1000000 + tv2.tv_usec;

    dis = (float)(time2 - time1) / 1000000 * 34000 / 2; //求出距离

    return dis;
```

```c
}
int main(void)
{
    float dis;
    if(wiringPiSetup() == -1){ /* when initialize wiring failed,
                                            print messageto screen */
        printf("setup wiringPi failed !");
        return 1;
    }

    ultraInit();

    while(1) {
        dis = disMeasure();
        printf("%0.2f cm\n",dis);
        delay(1000);
    }
    return 0;
}
```

- 3 : Modify the above code so that distance is measured at every 50ms
- 4 : For a given target distance, repeat measurement 100 times and evaluate how your measurements were successful. For example, aim at wall with 2m distance, and take 10 samples like : 2.0 .... 1.9. If you set the threshold range 1.9 ~ 2.1m, the measurement success ratio becomes 70%
- 5 : Discuss the limit and fault of the above code