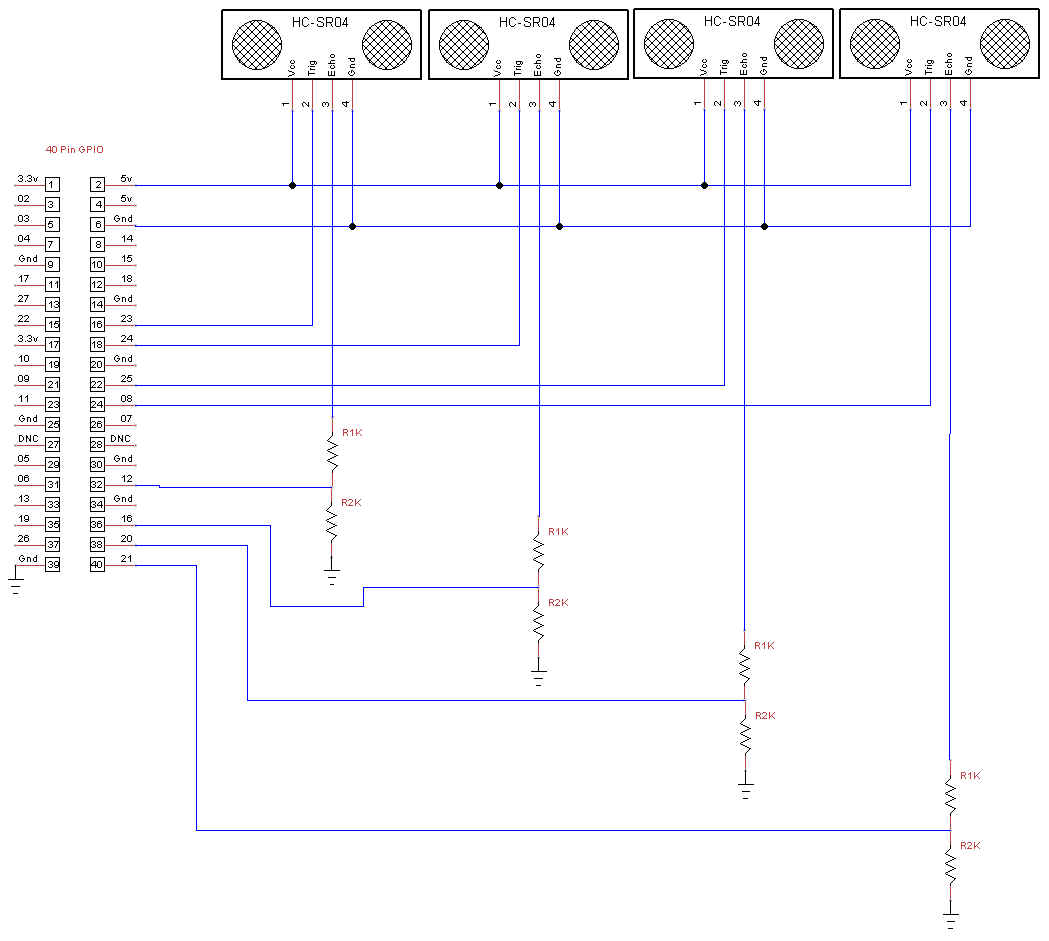


|  |  |  |  |
| --- | --- | --- | --- |
| **Location** | **Module** | **GPIO Pin #** | **Wire Color (Rainbow cable)** |
| Toilet #1 (E107) Ceiling | Ultrasonic Sensor #1 | #16(Trig), #32(Echo) | Black, White |
| Toilet #1 (E107) Wall | Ultrasonic Sensor #2 | #18(Trig), #36(Echo) | Grey, Purple |
| Toilet #2 (E108) Ceiling | Ultrasonic Sensor #3 | #22(Trig), #38(Echo) | Orange, Yellow |
| Toilet #2 (E108) Wall | Ultrasonic Sensor #4 | #24(Trig), #40(Echo) | Brown, Red |
|  |  | #2(Common VCC for all Ultrasonic Sensors) | Green (In each keystone jack, used Pin #3 (white-green) as common VCC(+ve)) |
|  |  | #39(Common Ground for all Ultrasonic Sensors) | Blue (In each keystone jack, used Pin #6 (green) as common Ground (-ve)) |
| Toilet #1 (E107) | Relay | #13 | Grey |
| Toilet #2 (E108) | Relay | #15 | White |
|  | Relay | #4(VCC) | Purple |
|  | Relay | #6(Gnd) | Black |
| Remark: E107 Toilet nearby entrance, E108 toilet nearby kitchen. | | | |
|  |  |  |  |

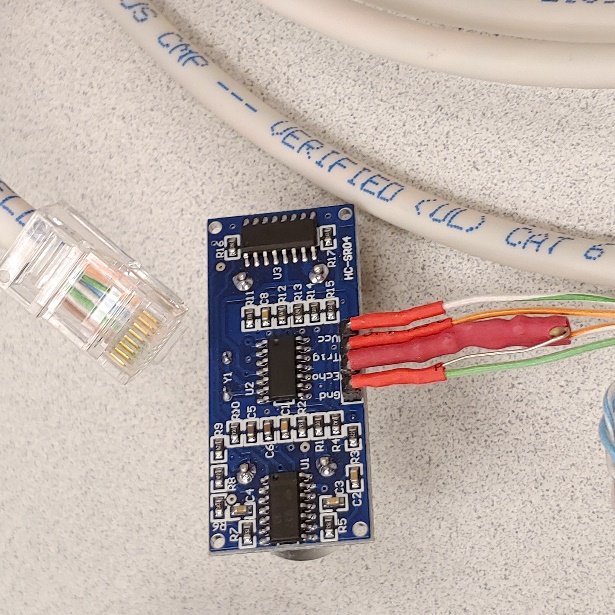
Multiple Ultrasonic Sensors to Raspberry Pi

To connect 4 Ultrasonic HC-SR04 Distance sensors is straight forward , you need to connect 5V and ground to each sensor ,  
  
Then you need to connect 4 gpio pins one to each sensor trigger pin, and finally you need to connect each of the echo pins to a 4 gpio pins using a potential divider circuit. (R1K is 1KΩ resistor, R2K is 2KΩ resistor)

Some thing like this



|  |  |  |
| --- | --- | --- |
| RJ45 Jack Pin Assignment | | |
| Pin # | Colour | Description |
| 1 | White-Orange | Echo |
| 2 | Orange | Trigger |
| 3 | White-Green | VCC (+ve) |
| 6 | Green | Ground (-ve) |



Download and Install the Raspbian Operating System in the MicroSD card

<https://www.raspberrypi.org/documentation/installation/installing-images/README.md>

Getting started with the Raspberry Pi

<https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started>

Installed additional packages and enable some service(interface) for this project:

1. sudo apt-get update
2. sudo apt-get upgrade -y
3. Install Apache Web server, sudo apt-get install apache2
4. Install PHP, sudo apt-get install php
5. Install xrdp (optional, for remote desktop connection): sudo apt-get install xrdp
6. enable ssh service, sudo raspi-config -> select Interfacing Option -> Select P2 SSH to enable SSH.
7. create the cron table, crontab –e and put the following lines at the buttom:

45 07 \* \* 1 sudo shutdown -r now # Weekly reboot the Raspberry Pi

@reboot sudo /usr/bin/python /home/pi/ADP/adp\_tls.py > /dev/null 2>&1

@reboot ps -ef | grep adp\_tls.py | grep -v grep | awk '{print $2}' > /home/pi/ADP/adp\_tls.pid

1. create a script file under /etc/init.dcd

pi@adptl01:~ $ sudo vi /etc/init.d/relayOff

The content of this file is:

#! /bin/sh

### BEGIN INIT INFO

# Provides: relayOff

# Required-Start: $syslog

# Required-Stop:

# X-Start-Before: rmnologin

# Default-Start: 2 3 4 5

# Default-Stop:

# Short-Description: Turn Off GPIO pins 13 & 15

# Description: Provide limited super user privileges to specific users.

### END INIT INFO

set -e

case "$1" in

start)

echo "Turning Off Relay...."

# Turn 13,15 on which keeps relay off

/usr/bin/gpio mode 2 out

/usr/bin/gpio mode 3 out

/usr/bin/gpio write 2 0

/usr/bin/gpio write 3 0

echo "Turned Off Relay" >> /home/pi/relay.log

date >> /home/pi/relay.log

;;

stop|reload|restart|force-reload|status)

;;

\*)

echo "Usage: $N {start|stop|restart|force-reload|status}" >&2

exit 1

;;

esac

exit 0

As you saved the file, issue the following command to activate the script.

pi@adptl01:~ $ *sudo update-rc.d relayOff defaults*

1. make a directory “ADP” under the /home/pi

pi@adptl01:~ $ mkdir –p ADP/log/oldlog

1. Place the Python program “adp\_tls.py” under /home/pi/ADP directory.
2. Reboot the Raspberry Pi

pi@adptl01:~ $ sudo shutdown –r now