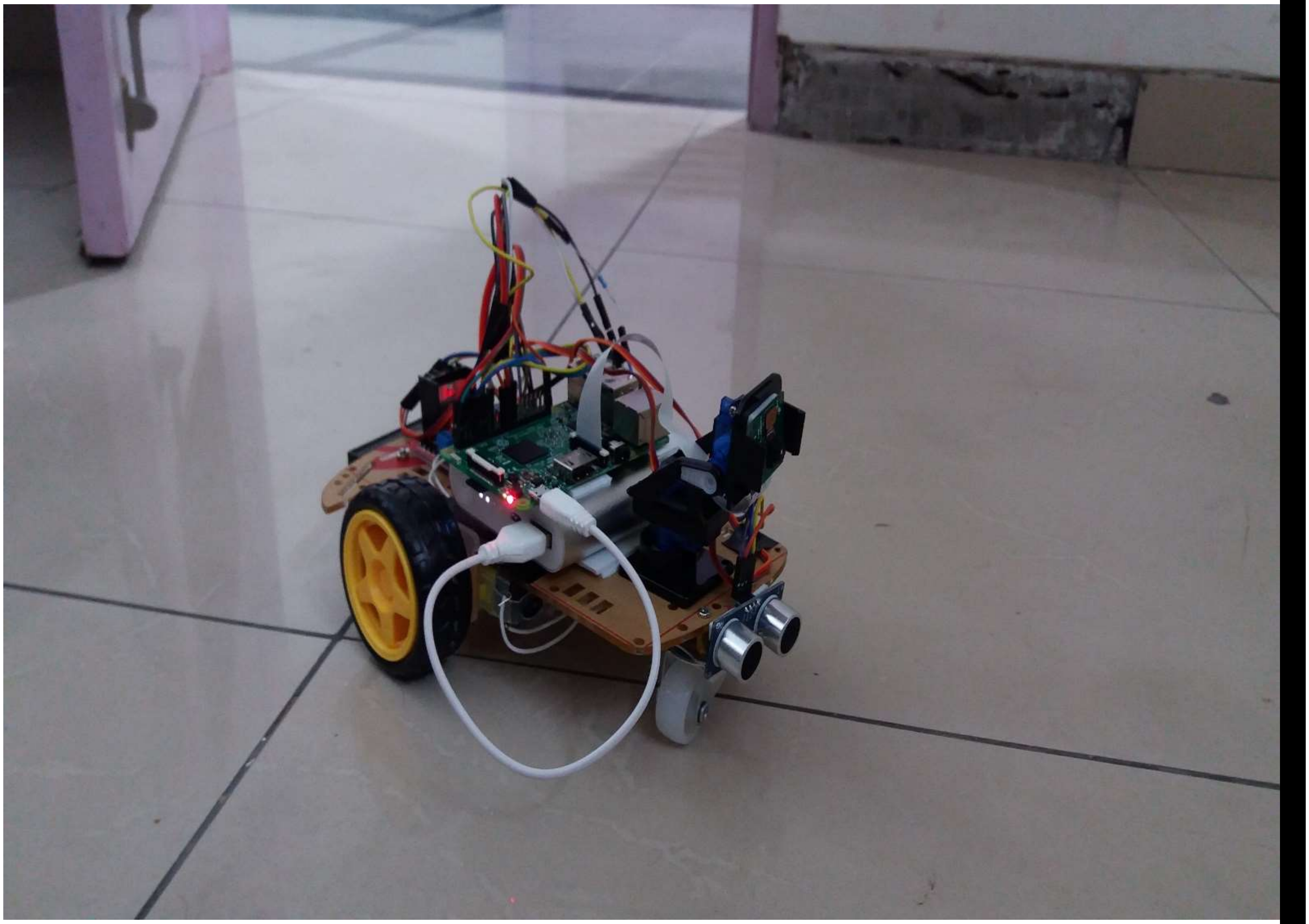


SURVIELLANCE BOT

DLC ROJECT

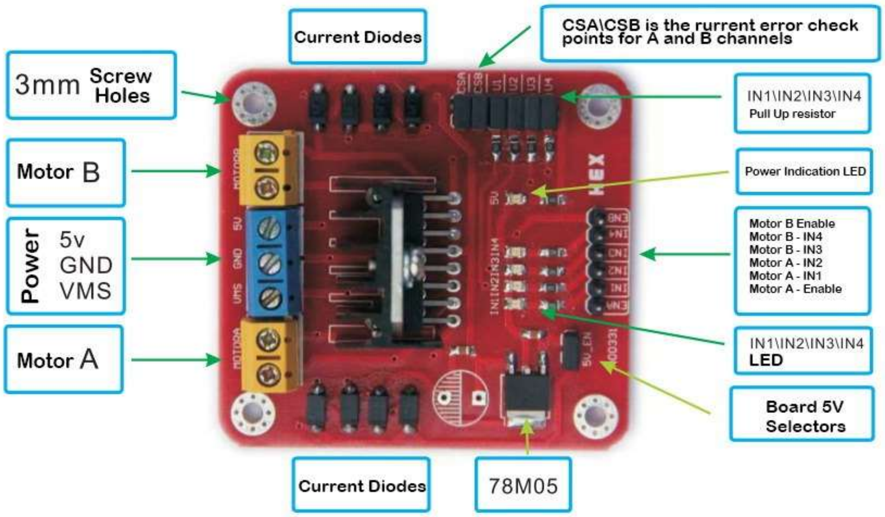
Build an autonomous, 3WD Raspberry Pi Car with camera stream and obstacle detection.

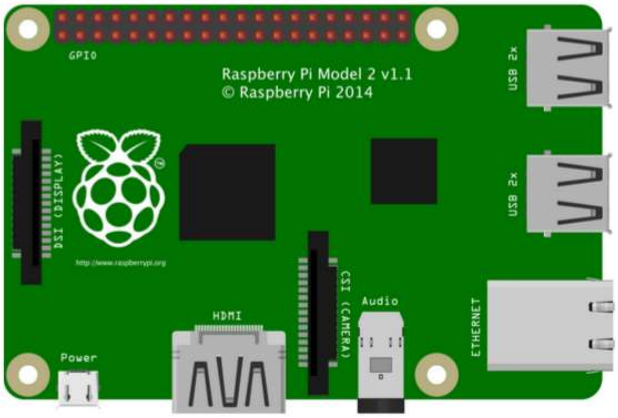
Group Members are: -

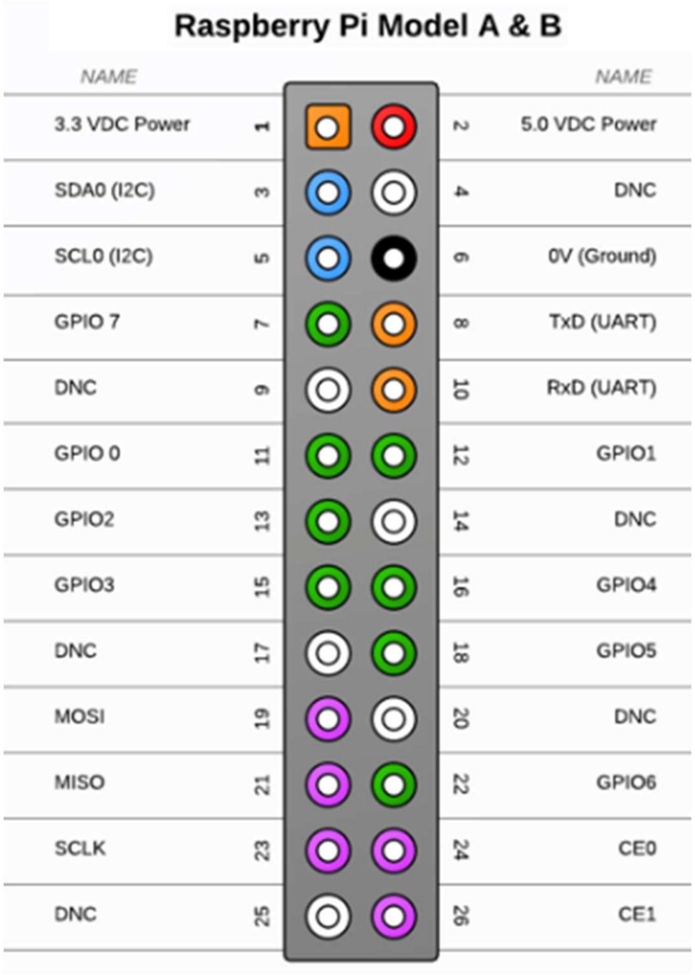
1. **Akshat Khanna (S2) (U10115FCS046)**
2. **Yash Sangai (S7) (U10115FCS250)**
3. **Mohit Singh (S5) (U10115FCS224)**
4. **Fayaz Maknojia (S3) (U10115FCS255)**

SURVIELLANCE BOT

Schematic

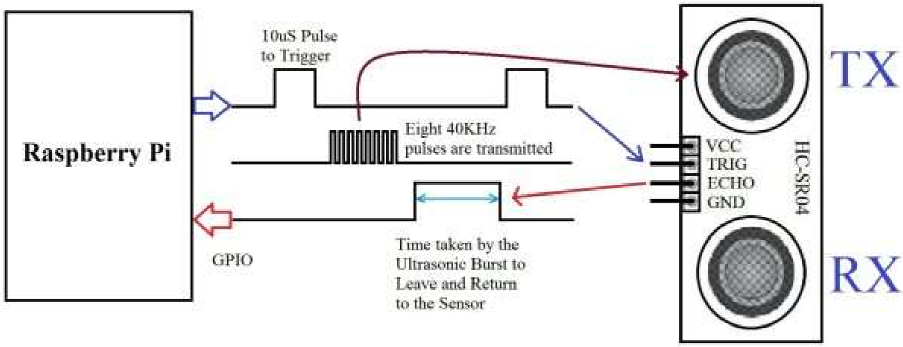


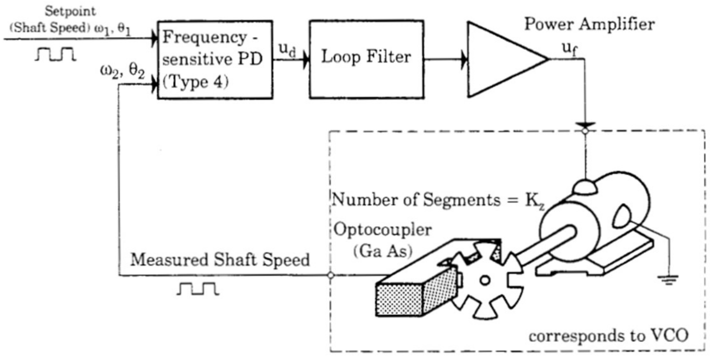




Raspberry Pi Model A & B

NAME			NAME
3.3 VDC Power	1	2	5.0 VDC Power
SDA0 (I2C)	3	4	DNC
SCL0 (I2C)	5	6	0V (Ground)
GPIO 7	7	8	TxD (UART)
DNC	9	10	RxD (UART)
GPIO 0	11	12	GPIO1
GPIO2	13	14	DNC
GPIO3	15	16	GPIO4
DNC	17	18	GPIO5
MOSI	19	20	DNC
MISO	21	22	GPIO6
SCLK	23	24	CE0
DNC	25	26	CE1





Working: -

1. Stepper Driver

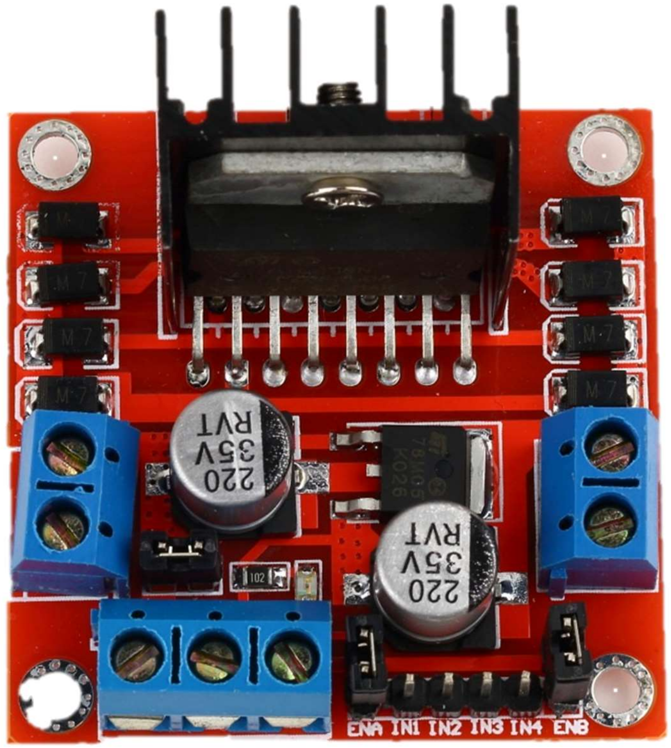
Internally the L298N consists of four independent power amps with 5-volt digital inputs and four high current, high voltage power amplifiers capable of driving single DC motors, and both unipolar and bi-polar stepper motors.

The four amplifiers are usually used in pairs forming an H-bridge to switch polarity for to control the direction of a single DC motor or as two pairs of H-bridges a bi-polar stepper motor.

The four power amplifiers and grouped in pairs of two with individual enable pins (ENA, ENB) and individual current sense pins (CSA, CSB) for each pair. The current sense pins in general can be tied to ground, but one can insert low value resistor, whose voltage reading is proportional to current.

ENA, ENB, and In1-In4 are all standard 5-volt TTL logic making connection to most micro-controllers easy. ENA will turn on A1 and A2 when with a digital HIGH (5-volts) and off when LOW (0 volts); the corresponding outputs will be floating when off. Same is true of ENB, In3 and In4. ENA and ENB can be connected directly together to enable both channels at once or simply tied to +5 volts and both channels making all four outputs active at all times.

A 5-volt TTL level input to In1, In2 In3, or In4 will produce a corresponding output of Vm (motor voltage) minus about a volt.



2. Servo Motor

Servos are controlled by sending an electrical pulse of variable width, or **pulse width modulation** (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90 degrees in either direction for a total of 180 degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the **motor** determines position of the shaft, and based on the duration of the pulse sent via the control wire; the **rotor** will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position. Shorter than 1.5ms moves it to 0 degrees, and any longer than 1.5ms will turn the servo to 180 degrees.

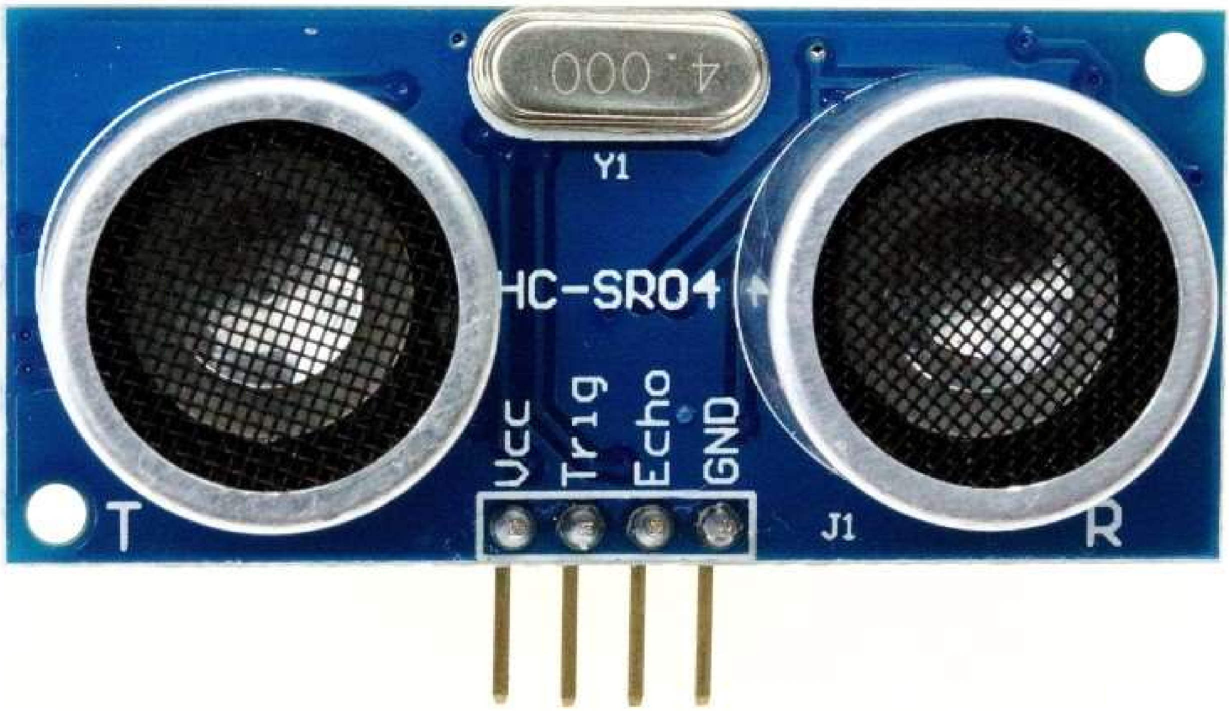
When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position.

3. Ultrasonic Distance Sonar

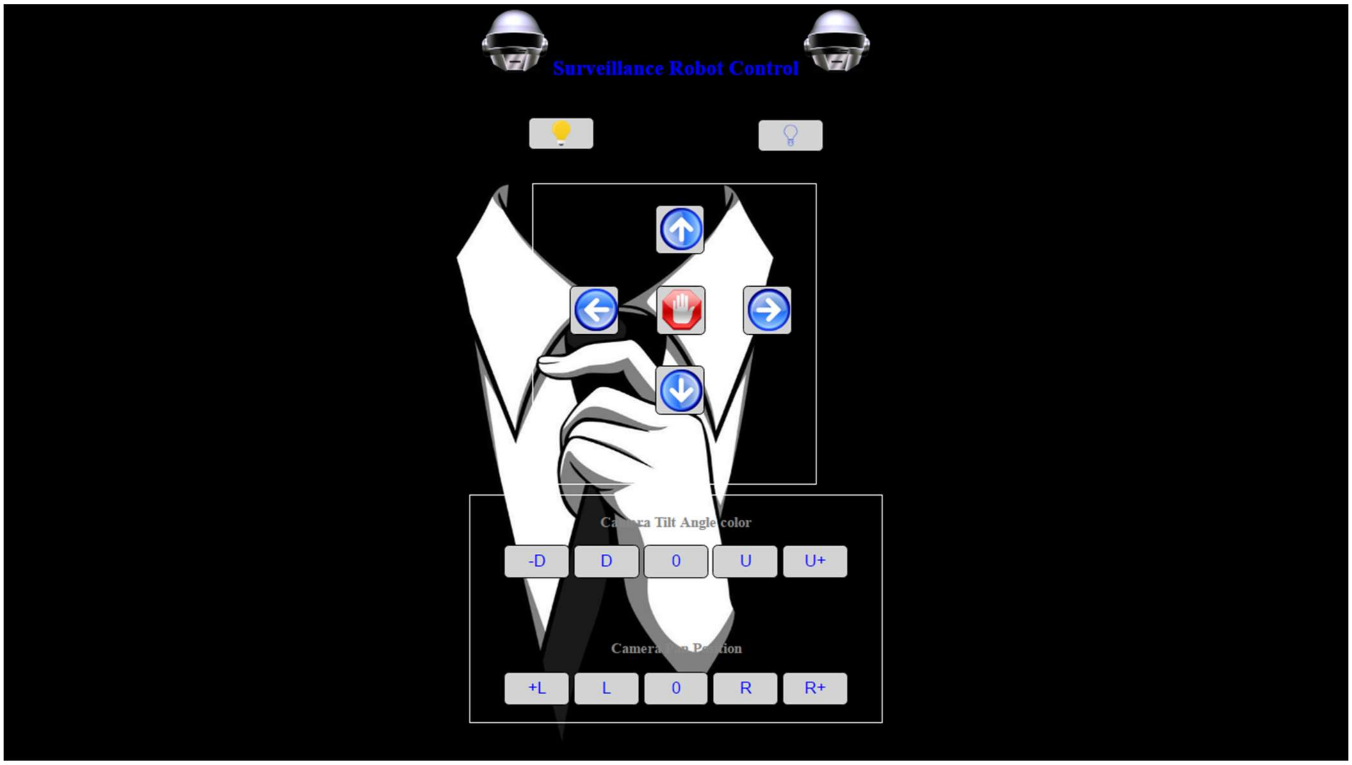
It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

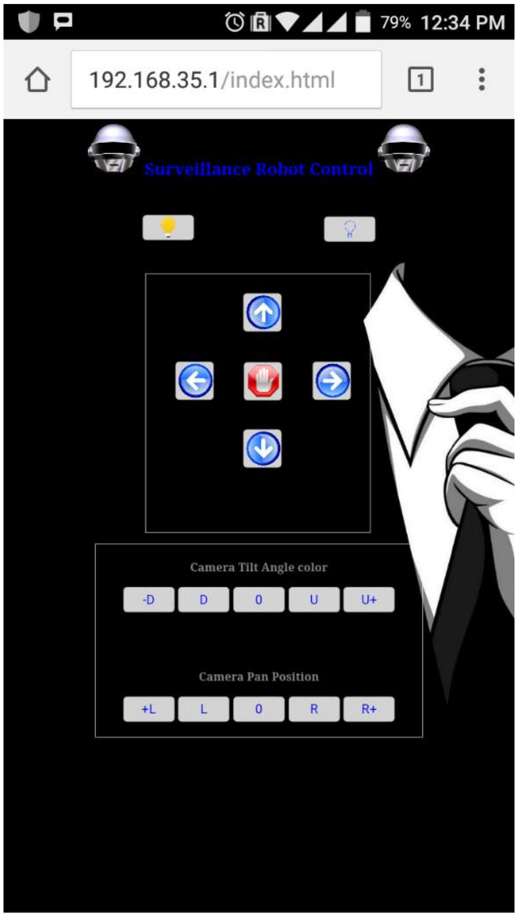
The HC-SR04 Ultrasonic Module has 4 pins, Ground, VCC, Trig and Echo. The Ground and the VCC pins of the module needs to be connected to the Ground and the 5 volts pins on the Pi Board respectively and the trig and echo pins to any Digital I/O pin on the Pi Board.

In order to generate the ultrasound you need to set the Trig on a High State for 10 µs. That will send out an 8 cycle sonic burst which will travel at the speed sound and it will be received in the Echo pin. The Echo pin will output the time in microseconds the sound wave traveled.



Website


Surveillance Robot Control
Camera Tilt Angle color
-D
D
0
U
U+
Camera Pan Position
+L
L
0
R
R+


79% 12:34 PM
192.168.35.1/index.html
1
Surveillance Robot Control
Camera Tilt Angle color
-D
D
0
U
U+
Camera Pan Position
+L
L
0
R
R+

Coding..

1. **Website**

```
<html>

<head>

</head>


<style>

body {background: url(wallpapers.jpg)}

h1 {color:blue}


button {

        color: blue;

        background:lightgrey;

        border: 1px solid #000;

        border-radius: 8px;

        position: center;

}


</style>


<body>


<div style="text-align:center">


<h1><img style="height: 100px" src="robot.png"> Surveillance Robot Control   <img style="height: 100px"src="robot.png"> </h1>

<br><br>
```

```
<form action = "button.php" method = "POST">

        <button style="height: 50px; width: 100px" type = "submit" name ="lighton" id="lighton" value = "Ture" ><img style="height: 40px"src="bulbon.png"></button>

        <img hspace="20" style="padding-left: 200px">

        <button style="height: 50px; width: 100px" type = "submit" name ="lightoff" id="lightoff" value = "Ture" ><img style="height: 35px"src="bulboff.png"></button>

</form>

<br><br>


<span style="display:inline-block;padding:5px;border:1px solid #fff; font-size: 140%;font-weight: bold;">

        <br>


        <form action = "button.php" method = "POST">

        <img hspace="5" style="padding-left: 1px">

                <button style="height: 75px; width: 75px"type = "submit" name ="fwd" id="fwd" value = "Ture" ><img style="height: 65px"src="forward.png"></button>

        </form>

                <br>

        <form action = "button.php" method = "POST">


                <img hspace="20" style="padding-left: 5px">


                <button style="height: 75px; width: 75px"type = "submit" name ="left" id="left" value = "Ture"><img style="height: 65px"src="left.png"></button>


                <img hspace="20" style="padding-left: 5px">


                <button style="height: 75px; width: 75px" type = "submit" name ="stop" id="stop" value = "Ture"><img style="height: 63px"src="stop.ico"></button>


                <img hspace="20" style="padding-left: 5px">
```

```
                    <button style="height: 75px; width: 75px" type = "submit" name ="right" id="right" value = "Ture"><img style="height: 65px"src="right.png"></button>


                    <img hspace="10" style="padding-left: 5px">

          </form>

          <br>

          <form action = "button.php" method = "POST">

                    <img hspace="5" style="padding-left: 1px">

                    <button style="height: 75px; width: 75px" type = "submit" name ="rev" id="rev" value = "Ture"><img style="height: 65px"src="reverse.png"></button>

          </form>

                    <br><br><br>



</span>

<br><br>


<span style="display:inline-block;padding:5px;border:1px solid #fff; font-size: 140%;font-weight: bold;">



      <p style="color:#888888">Camera Tilt Angle color</p>

                    <form action = "button.php" method = "POST">

                    <img hspace="18" style="padding-left: 5px">


                    <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="LT" id="LT" value = "Ture">-D</button>


                    <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="LMT" id="LMT" value = "Ture">D</button>


                    <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="MT" id="MT" value = "Ture">0</button>
```

```
                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="RMT" id="RMT" value = "Ture">U</button>


                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="RT" id="RT" value = "Ture">U+</button>


                        <img hspace="18" style="padding-left: 5px">

                        </form>

            <br><br>

                <p style="color:#888888">Camera Pan Position</p>

                        <form action = "button.php" method = "POST">


        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="LP" id="LP" value = "Ture">+L</button>


                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="LMP" id="LMP" value = "Ture">L</button>


                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="MP" id="MP" value = "Ture">0</button>


                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="RMP" id="RMP" value = "Ture">R</button>


                        <button style="height: 50px; width: 100px; font-size: 25px" type = "submit" name ="RP" id="RP" value = "Ture">R+</button>

        </form>

               <p></p>

</span>

</body>

</html>

</xmp>
```

2. **Button.php**

```
<?php
$file1 = "buttonStatus1.txt";
$file2 = "buttonStatus2.txt";
$file3 = "buttonStatus3.txt";
$file4 = "buttonStatus4.txt";

if (isset($_POST['fwd']))
{
$handle = fopen($file1,'w+');
$onstring = "forward";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if(isset($_POST['left']))
{
$handle = fopen($file1,'w+');
$offstring = "Left Turn";
fwrite($handle, $offstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['right']))
{
$handle = fopen($file1,'w+');
$onstring = "Right Turn";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if(isset($_POST['rev']))
{
$handle = fopen($file1,'w+');
$offstring = "Reverse";
fwrite($handle, $offstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['stop']))
{
$handle = fopen($file1,'w+');
$onstring = "Stop";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if(isset($_POST['lighton']))
```

```
{
$handle = fopen($file2,'w+');
$offstring = "OFF";
fwrite($handle, $offstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['lightoff']))
{
$handle = fopen($file2,'w+');
$onstring = "ON";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['LP']))
{
$handle = fopen($file3,'w+');
$onstring = "Left pan";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['LMP']))
{
$handle = fopen($file3,'w+');
$onstring = "Left mid pan";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['MP']))
{
$handle = fopen($file3,'w+');
$onstring = "Mid pan";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['RP']))
{
$handle = fopen($file3,'w+');
$onstring = "Right pan";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['RMP']))
```

```
{
$handle = fopen($file3,'w+');
$onstring = "Right mid pan";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['LT']))
{
$handle = fopen($file4,'w+');
$onstring = "Left tilt";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['LMT']))
{
$handle = fopen($file4,'w+');
$onstring = "Left mid tilt";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['MT']))
{
$handle = fopen($file4,'w+');
$onstring = "Mid tilt";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['RMT']))
{
$handle = fopen($file4,'w+');
$onstring = "Right mid tilt";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
else if (isset($_POST['RT']))
{
$handle = fopen($file4,'w+');
$onstring = "Right tilt";
fwrite($handle,$onstring);
fclose($handle);

header( "refresh:0.1; index.html" );
}
?>
```

3. Raspberry pi

```
import RPi.GPIO as GPIO
import urllib2
import time
from subprocess import call

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(19,GPIO.OUT)
GPIO.setup(21,GPIO.OUT)
GPIO.setup(23,GPIO.OUT)
GPIO.setup(24,GPIO.OUT)
GPIO.setup(40,GPIO.OUT)
true = 1
while(true):

    response1 = urllib2.urlopen('http://192.168.35.1/buttonStatus1.txt')
    status1 = response1.read()
    response2 = urllib2.urlopen('http://192.168.35.1/buttonStatus2.txt')
    status2 = response2.read()
    response3 = urllib2.urlopen('http://192.168.35.1/buttonStatus3.txt')
    status3 = response3.read()
    response4 = urllib2.urlopen('http://192.168.35.1/buttonStatus4.txt')
    status4 = response4.read()

    if status1=='forward':
            GPIO.output(19 , False )
            GPIO.output(21 , True)
        GPIO.output(23 , False)
            GPIO.output(24 , True )
        print status1
    if status1=='Stop':
            GPIO.output(21 , False)
        GPIO.output(19 , False)
            GPIO.output(23 , False)
            GPIO.output(24 , False)
        print status1
    if status1=='Left Turn':
        GPIO.output(21 , True)
        GPIO.output(19 , False)
        GPIO.output(23 , False)
        GPIO.output(24 , False)
        print status1
    if status1=='Right Turn':
        GPIO.output(21 , False)
        GPIO.output(19 , False)
        GPIO.output(23 , False)
        GPIO.output(24 , True)
        print status1
    if status1=='Reverse':
        GPIO.output(21 , False)
        GPIO.output(19 , True)
        GPIO.output(23 , True)
    GPIO.output(24 , False)
    print status1

if status2=='ON':
    GPIO.output(40 , False)
if status2=='OFF':
    GPIO.output(40 , True)
if status3=='Left pan':
    call (["echo 0=0% > /dev/servoblaster"], shell=True)

if status3=='Left mid pan':
    call (["echo 0=15% > /dev/servoblaster"], shell=True)

if status3=='Mid pan':
    call (["echo 0=35% > /dev/servoblaster"], shell=True)

if status3=='Right mid pan':
    call (["echo 0=58% > /dev/servoblaster"], shell=True)

if status3=='Right pan':
    call (["echo 0=100% > /dev/servoblaster"], shell=True)

if status4=='Left tilt':
    call (["echo 1=75% > /dev/servoblaster"], shell=True)

if status4=='Left mid tilt':
    call (["echo 1=68% > /dev/servoblaster"], shell=True)

if status4=='Mid tilt':
    call (["echo 1=60% > /dev/servoblaster"], shell=True)

if status4=='Right mid tilt':
    call (["echo 1=50% > /dev/servoblaster"], shell=True)

if status4=='Right tilt':
    call (["echo 1=40% > /dev/servoblaster"], shell=True)
```