RoboVR

SRB WALKATHON

About the game:

In this game, there will be a classic foot race between two robots. The robot to walk the distance in the shortest time wins the race. The specifications and components to build the robot are mentioned below.

Components and its Specifications:

Sr. No.	Components	Specifications
1.	Servo Motor Single shaft (x6)	(RKI 1211), 14kg/cm (High torque metal gear servos)
2.	PCA9685 servo driver	16 channel 12 bit
3.	Arduino Nano	Microcontroller ATmega328
4.	Bluetooth Module HC-05	v2.0+EDR
5.	Battery (x2)	i. 9v (for Arduino) ii. LiPo 5v (2200 mAh 1S 25C/50C)
6.	Jumper wires	M-M, M-F
7.	Metal Horn for Servo 25T	-
8.	Multipurpose Aluminium Standard Servo Bracket	-
9.	Short U Shape Aluminium Servo Bracket	-
10.	Long U Aluminium Servo Bracket	-
11.	L Shaped Interconnect Servo Bracket	-
12.	Large U Beam Aluminium Servo Bracket	-
13.	Robot feet Aluminium Servo Bracket	-

Robot Details:

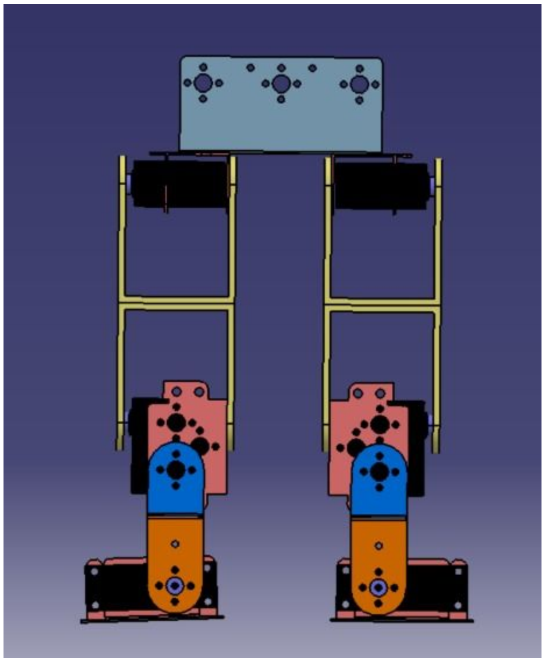
Robot Dimensions: 25cm Height

Robot Control: Wireless

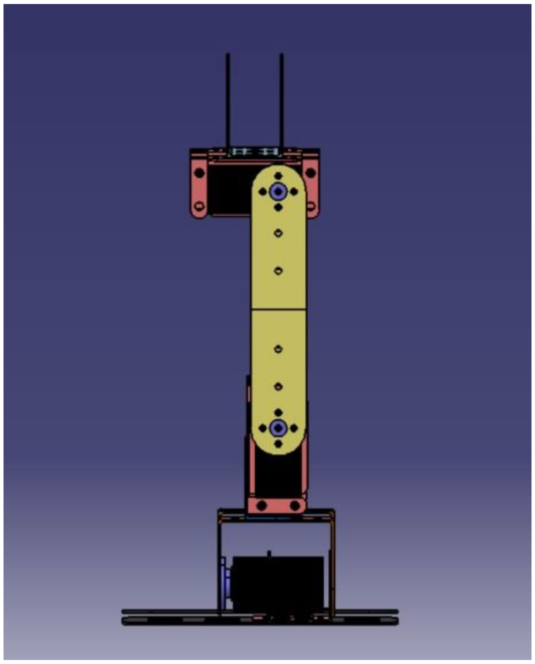
Robot Type: Biped Robot

Mechanical Design:

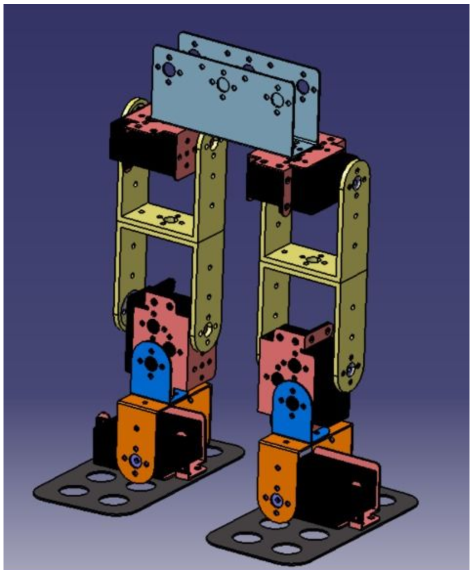
Front View



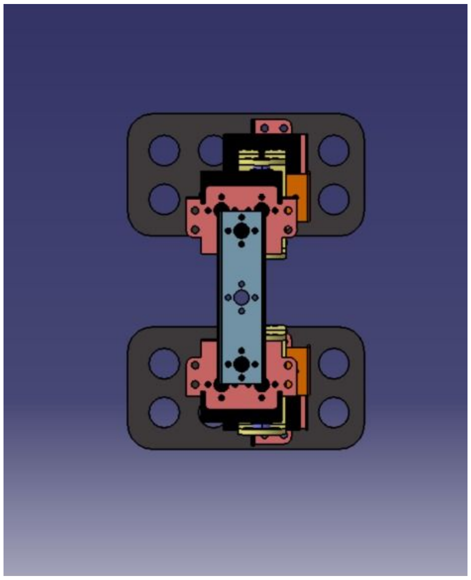
Side View



Isometric View



Top View



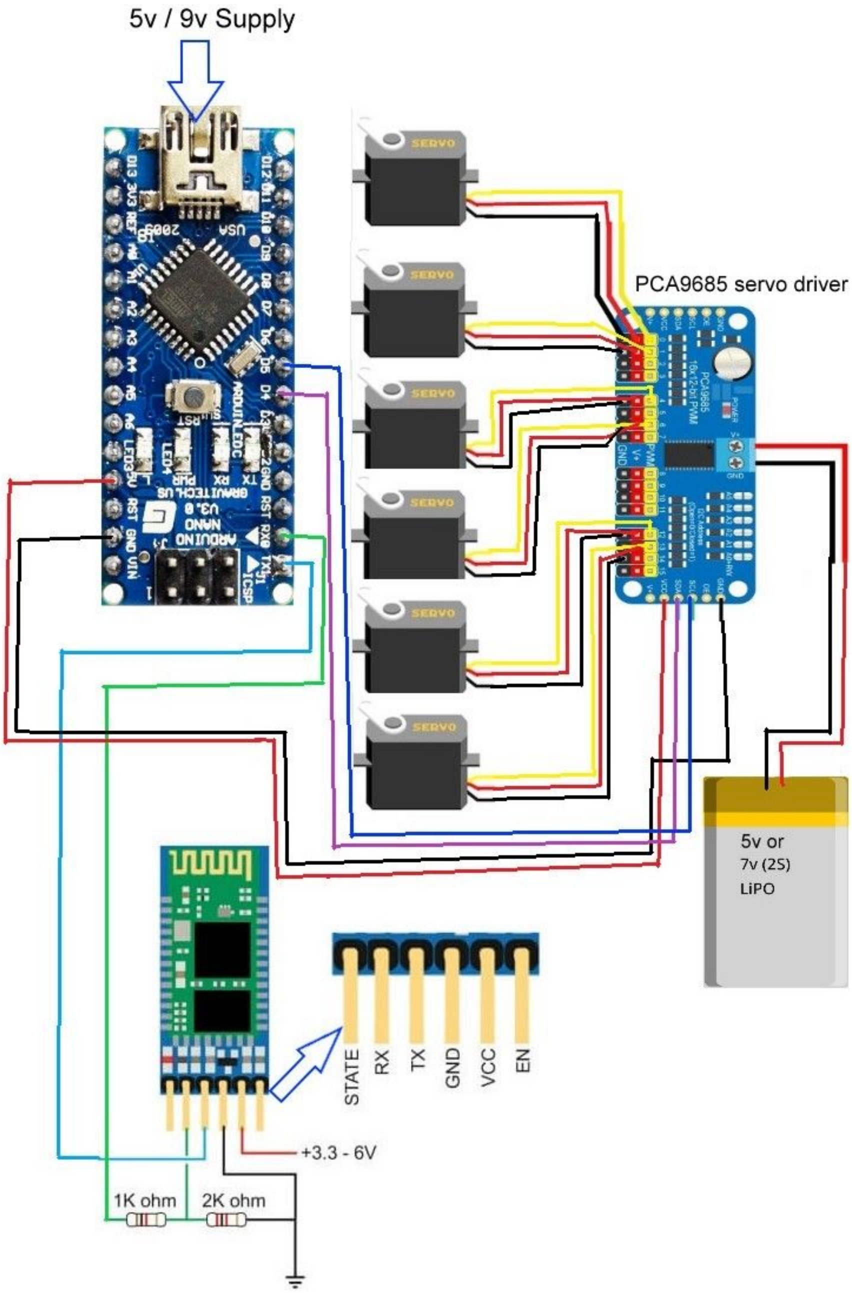
Working:

The robot is called a Biped robot which means two-foot robot. This Biped robot will replicate human locomotions and has 6 degrees of freedom (3 DOF on each leg). The no. of DOFs increases its redundancy for balance and control. We've used rotational joints as we have to replicate human legs. We've used high torque metal gear servos as actuators and used PWM techniques to control the servos. We've used Arduino Nano as a microcontroller. For walking of the robot, the Marching Gait Algorithm is used. For remote operation, we've used an HC05 Bluetooth module control.

The leg is having 3 joints: Hip Joint, Knee joint and Ankle joint. The hip and Knee are having the same 360° angle of rotation, but the ankle joint is different from the two.

Mechanical connections are shown above. The electronic connections and Arduino code are provided below.

Connections:



Codes:

```
for(i=0;i<180;i++)
{
servo.write(i);
}
```

This is the **basic code for rotate any servo motor** attached to any Arduino board.

But calibrating the rotating degrees and deciding which motors should run during the movement of each leg is the most tricky part of coding. It can be done by another Sketch called (Servo_Test). By testing the degree of rotation of each motor through serial communication through the Arduino board, we can calibrate every motor.

Servo Testing Arduino Code:

```
String readString;
#include <Servo.h>
Servo myservo;

void setup()
{
  Serial.begin(9600);
  myservo.attach(27);  //the pin for the servo control
  Serial.println("servo-test"); // so I can keep track of what is loaded
}

void loop()
{
  while (Serial.available())
  {
    delay(11);
    if (Serial.available() >0)
    {
      char c = Serial.read();  //gets one byte from serial buffer
      readString += c; //makes the string readString
    }
  }

  if (readString.length() >0)
  {
    Serial.println(readString);  //so you can see the captured string
    int n;
    char carray[6]; //converting string to number
    readString.toCharArray(carray, sizeof(carray));
    n = atoi(carray);
    myservo.writeMicroseconds(n); // for microseconds
    myservo.write(n); //for degees 0-180
    readString="";
  }
}
```

Full Robot Arduino Code:

```
#include <Servo.h>

char val = '0';

int stop_flag = 3, flag = 70, lp1 = 100, lp2 = 115, lp22 = 95, lp3 = 76, lp33 = 90 , lp4 = 80;

int rp1 = 90, rp2 = 115, rp22 = 90, rp33 = 90;

int  rp3 = 60, rp4 = 150, cp = 84, lhp = 90, rhp = 90;

Servo L1, L2, L22, L3, L33, L4, R1, R2, R22, R3, R33, R4, C, LH, RH;


void setup()

{

  L1.attach(2);

  L2.attach(3);

  L3.attach(4);

  L4.attach(5);

  R1.attach(6);

  R2.attach(7);

  R3.attach(8);

  R4.attach(10);

  L22.attach(22);

  L33.attach(23);

  R22.attach(24);

  R33.attach(25);

  C.attach(28);

  LH.attach(26);
  RH.attach(27);
  Serial.begin(9600);
  Serial.println("Ready");
}
void loop()
{
  initiate();
  command();
}
void initiate()
{
  L1.write(lp1);
  L2.write(lp2);
  L3.write(lp3);
  L4.write(lp4);
  R1.write(rp1);
  R2.write(rp2);
  R3.write(rp3);
  R4.write(rp4);
  L22.write(lp22);
  L33.write(lp33);
  R22.write(rp22);
  R33.write(rp33);
  C.write(cp);
  LH.write(lhp);

  RH.write(rhp);

  flag = 0;

  delay(2500);

}

void command()

{

  if (Serial.available() > 0)

  {

    val = {

      Serial.read()

    };

  }

  if (val == '0')

  {

    walk();

    Serial.write("check");

  }

  if (val == '1')

  {

    stand();

  }

  if (val == '2')

  {
    leftt();
  }
  if (val == '3')
  {
    rightt();
  }
}
void walk()
{
  while (val == '0')
  {
    if (flag == 0)
      ini();
    if (flag == 1)
    {
      //C.write(94);
      left();
      //command();
    }
    if (flag == 2)
    {
      //C.write(76);
      right();
      //command();
    }
  }
}
void ini()
{
  L2.write(105);
  L3.write(64);
  R2.write(125);
  R3.write(72);
  L22.write(105);
  L33.write(102);
  R22.write(80);
  R33.write(78);
  flag = 1;
  rp2=125;
  rp3=72;
  rp22=80;
  rp33=78;
  Serial.println("Ini");
}
void left()
{
  while (flag == 1) //Left Leg
  {
```

```
if (rp1 < 110)
{
  rp1 = rp1 + 2;
  R1.write(rp1);
  //Serial.println(rp1);
}
if (lp2 > 95)
{
  lp2--;
  L2.write(lp2);
  //Serial.println(lp2);
}
if (lp3 > 54)
{
  lp3--;
  L3.write(lp3);
  //Serial.println(lp3);
}
if (lp22 < 115)
{
  lp22++;
  L22.write(lp22);
  //Serial.println(lp22);
}
```

```
if (lp33 < 112)
{
  lp33++;
  L33.write(lp33);
  //Serial.println(lp33);
}

LH.write(110);
RH.write(140);

if (rp2 > 115)
{
  rp2--;
  R2.write(rp2);
  //Serial.println(rp2);
}
if (rp3 > 62)
{
  rp3--;
  R3.write(rp3);
  //Serial.println(rp3);
}
if (rp22 < 90)
{
      rp22++;
      R22.write(rp22);
      //Serial.println(rp22);
    }
    if (rp33 < 88)
    {
      rp33++;
      R33.write(rp33);
      //Serial.println(rp33);
    }
    //Serial.println(++count);
    if (lp2 == 95 && lp3 == 54 && rp1 == 110 && rp2 == 115 && rp3 == 62 && lp22 == 115 && lp33 == 112 && rp22 == 90 && rp33 == 88)
    {
      //Serial.write("Check");
      //for (cp; cp < 84; cp++)
      //{
       // C.write(cp);
        //delay(30);
      //}
      for (rp1; rp1 > 90; rp1--)
      {
        R1.write(rp1);
        delay(30);
      }
      Serial.println("Left_Leg");
      flag++;
      stop_flag = 0;
      delay(120);
    }
    delay(25);
  }
}
void right()
{
  while (flag == 2) //Right leg
  {
    if (lp1 > 74)
    {
      lp1 = lp1 - 2;
      L1.write(lp1);
    }
    if (lp2 < 115)
    {
      lp2++;
      L2.write(lp2);
    }
    if (lp3 < 74)
```

```
{
  lp3++;
  L3.write(lp3);
}
if (lp22 > 95)
{
  lp22--;
  L22.write(lp22);
}
if (lp33 > 92)
{
  lp33--;
  L33.write(lp33);
}

LH.write(50);
RH.write(70);

if (rp2 < 135)
{
  rp2++;
  R2.write(rp2);
}
if (rp3 < 82)
```

```
  {
    rp3++;
    R3.write(rp3);
  }
  if (rp33 > 68)
  {
    rp33--;
    R33.write(rp33);
  }
  if (rp22 > 70)
  {
    rp22--;
    R22.write(rp22);
  }
  if (lp1 == 74 && lp2 == 115 && lp3 == 74 && rp3 == 82 && rp2 == 135 && lp22 == 95 && lp33 == 92
&& rp33 == 68 && rp22 == 70)
  {
    //for (cp; cp < 84; cp++)
    //{
    //  C.write(cp);
    //  delay(30);
    //}
    for (lp1; lp1 < 100;)
    {
        lp1++;

        L1.write(lp1);

        delay(30);

      }

      Serial.println("Right_Leg");

      flag = 1;

      stop_flag = 1;

      delay(120);

    }

    delay(25);

  }

}

void stand()

{

  while (stop_flag == 0)

  {

    if (lp2 > 94)

    {

      lp2--;

      L2.write(lp2);

    }

    if (lp3 > 52)

    {

      lp3--;
    L3.write(lp3);
  }
  if (lp2 == 94 && lp3 == 52)
  {
    for (rp1; rp1 < 100; rp1++)
    {
      R1.write(rp1);
      delay(12);
    }
    flag = 70;
    stop_flag = 2;
    Serial.println("STOP");
  }
  delay(25);
}
while (stop_flag == 1)
{
  if (rp2 > 130)
  {
    rp2--;
    R2.write(rp2);
  }
  if (rp3 < 80)
  {
      rp3++;
      R3.write(rp3);
    }
    if (rp2 == 130 && rp3 == 80)
    {
      for (lp1; lp1 > 110; lp2--)
      {
        L1.write(lp1);
        delay(12);
      }
      flag = 70;
      stop_flag = 2;
      Serial.println("STOP");
    }
    delay(25);
  }
  if (Serial.available() > 0)
  {
    val = {
      Serial.read()
    };
  }
  command();
}
```

```
void leftt()

{

  left();

  val = 1;

  command();

}

void rightt()

{

  right();

  val = 1;

  command();

}
```