```
#include <Servo.h>
#include <IRremote.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
Servo myservo1;//
Servo myservo2;//
Servo myservo3;//
Servo myservo4;//
Servo myservo5;//
Servo myservo6;//

int servoPosition;
int analogValue;

void slowServo1(int degInitial, int degFinal);
void slowServo2(int degInitial, int degFinal);
void slowServo3(int degInitial, int degFinal);
void slowServo4(int degInitial, int degFinal);
void slowServo5(int degInitial, int degFinal);
void slowServo6(int degInitial, int degFinal);

void addZero(int setNumber,int column, int row);

void  sine_gen1();
void  sine_gen2();
void  sine_gen3();
void  sine_gen4();
void  sine_gen5();
int sine1;
int angle1;
int sine2;
int angle2;
int sine3;
int angle3;

static int initial4=58;//set for horizontal scan
static int final4=59;//set for horizontal scan
static int initial3;
static int final3;

int analogA1 = A1;
int analogA2 = A2;
int analogA3 = A3;
int analogA4 = A0;// do not use A4 - i2C
int analogA5 = A7;// do not use A5 -i2C
int analogA6 = A6;
int Joint1;
int Joint2;
int Joint3;
int Joint4;
int Joint5;
int Joint6;
int receiver = 3; // Nano Signal Pin of IR receiver

static bool execute0;
static bool execute1;
static bool execute2;
static bool execute3;
static bool execute4;
static bool execute5;
static bool execute5_2;
static bool execute5_3;
static bool execute5_4;
static bool execute_Analog;
static bool execute_Sine;
static bool execute_Sweep;

void translateIR(); // takes action based on IR code received

void Home_Task2();
void Analog_Set_Task1();
void HomeToFive();
void FiveToHome();
void HomeToOne();
void OneToHome();
void HomeToTwo();
void TwoToHome();
void HomeToThree();
void ThreeToHome();
void HomeToFour();
void FourToHome();
void HomeToFive2();
void Five2ToHome();
void Five3ToHome();
void HomeToFive3();
void Five4ToHome();
void HomeToFive4();
void setupHome();

void horizontalScan();

IRrecv irrecv(receiver);    // create instance of 'irrecv'
decode_results results;     // create instance of 'decode_results'
void setup()
{
Serial.begin(9600);
delay(5000);
setupHome();
Serial.begin(9600);
Serial.println("IR Receiver Button Decode");
irrecv.enableIRIn(); // Start the receiver
lcd.init();
lcd.backlight();
```

```
}
void loop()
{

//************Infra Red Remote Control*****************
if (irrecv.decode(&results) )// have we received an IR signal?
     {
     translateIR();
     irrecv.resume(); // receive the next value
     Serial.print("results = ");
     Serial.println(results.value, HEX);
     }
 if(execute_Analog==true)
       Analog_Set_Task1();

 if(execute_Sine==true)
  {
   sine_gen1();
   sine_gen2();
   sine_gen3();
  }
 if(execute_Sweep==true)
  {
   horizontalScan();
  }
}
//************sine wave functions
void  sine_gen1()
{            //SINE WAVE GEN
// frequency 1/(360*delay_time)
angle1 = angle1 + 2;  if (angle1 > 360)  angle1 = 0;
//sine1 = 90+90 * sin((angle1 * PI / 180));
sine1 = 90+25*sin((90+angle1) * 0.0174532925);
myservo1.write(sine1);
//sine1 = abs(sine);
lcd.setCursor(0,0);
  lcd.print("      ");
  lcd.setCursor(0,0);
  lcd.print("1=");
  lcd.setCursor(2,0);
  lcd.print(angle1);
}

void  sine_gen2()
{            //SINE WAVE GEN
// frequency 1/(360*delay_time)
angle2 = angle2 + 2;  if (angle2 > 360)  angle2 = 0;
//sine = 90+90 * sin((angle2 * PI / 180));
sine2 = 100+30*sin(angle2* 0.0174532925);
myservo4.write(sine2);
//sine2 = abs(sine);
lcd.setCursor(6,1);
  lcd.print("     ");
  lcd.setCursor(6,1);
  lcd.print("4=");
  lcd.setCursor(8,1);
  lcd.print(angle2);
}

void  sine_gen3()
{            //SINE WAVE GEN
// frequency 1/(360*delay_time)
angle3 = angle3 + 2;  if (angle3 > 360)  angle3 = 0;
//sine = 90+90 * sin((angle2 * PI / 180));
sine3 = 90+2*sin((180+angle3)* 0.0174532925);
myservo2.write(sine3);
//sine2 = abs(sine);
lcd.setCursor(6,0);
  lcd.print("     ");
  lcd.setCursor(6,0);
  lcd.print("2=");
  lcd.setCursor(8,0);
  lcd.print(angle3);
}

void horizontalScan()
{

int initial1=35;
int final1=145;
slowServo1(initial1,final1);

if(final4>19)
  {
    initial4=initial4-10;
    final4=final4-10;
    slowServo4(initial4,final4);
    final3=final4-10;
    initial3=initial4-10;
  }
else
  {
    initial3=initial3+2;
    final3=final3+4;
    if(final3<30)
       slowServo3(initial3,final3);

  }

initial1=145;
final1=35;
slowServo1(initial1,final1);

if(final4>19)
  {
    initial4=initial4-10;
    final4=final4-10;
    slowServo4(initial4,final4);
    final3=final4-10;
    initial3=initial4-10;
  }
else
  {
    initial3=initial3+2;
    final3=final3+4;
    if(final3<30)
       slowServo3(initial3,final3);
   }
if(final3>30 && final4<=19)
  {
  execute_Sweep=false;
  slowServo1(final1, 90);
  slowServo4(final4,108);
  slowServo3(final3,3);
  initial4=58;
  final4=59;
  }
}

//********************IR
Control*******************************************
void translateIR() // takes action based on IR code received
// describing Remote IR codes
{

switch(results.value)
  {
  case 0xFFA25D: Serial.println("POWER");
  setupHome();
  break;

  case 0xFFE21D: Serial.println("FUNC/STOP");
  execute_Analog=true;
  break;

  case 0xFF629D: Serial.println("VOL+"); break;

  break;

  case 0xFF22DD: Serial.println("FAST BACK");

  break;

  case 0xFF02FD: Serial.println("PAUSE");
  execute_Sweep=true;
  slowServo4(108,58);// enable for horizonal scan
  break;

  case 0xFFC23D: Serial.println("FAST FORWARD");

  break;
```

```
case 0xFFE01F: Serial.println("DOWN");

break;

case 0xFFA857: Serial.println("VOL-");

break;

case 0xFF906F: Serial.println("UP");

break;

case 0xFF9867: Serial.println("EQ");
execute_Sine=true;
break;

case 0xFFB04F: Serial.println("ST/REPT");
execute0=false;
execute1=false;
execute2=false;
execute3=false;
execute4=false;
execute5=false;
execute5_2=false;
execute5_3=false;
execute5_4=false;
execute_Analog=false;
execute_Sine=false;
execute_Sweep=false;
Serial.print("execute_Sweep = ");
Serial.print(execute_Sweep);
break;

case 0xFF6897: Serial.println("0");
if (execute0==false )
  {
  HomeToOne();
  delay(500);
  OneToHome();
  delay(500);
  HomeToFive();
  delay(500);
  FiveToHome();
  delay(500);

  HomeToTwo();
  delay(500);
  TwoToHome();
  delay(500);
  HomeToFive2();
  delay(500);
  Five2ToHome();
  delay(500);
  HomeToThree();
  delay(500);
  ThreeToHome();
  delay(500);
  HomeToFive3();
  delay(500);
  Five3ToHome();
  delay(500);


  HomeToFour();
  delay(500);
  FourToHome();
  delay(500);
  HomeToFive4();
  delay(500);
  Five4ToHome();
  delay(500);
  }
  execute0=true;
break;

case 0xFF30CF: Serial.println("1");
if (execute1==false )
  {
  HomeToOne();
  delay(500);
  OneToHome();
  delay(500);
  }
  execute1=true;
break;

case 0xFF18E7: Serial.println("2");
if (execute2==false )
  {
  HomeToTwo();
  delay(500);
  TwoToHome();
  delay(500);
  }
break;
execute2=true;

case 0xFF7A85: Serial.println("3");
if (execute3==false )
  {
  HomeToThree();
  delay(500);
  ThreeToHome();
  delay(500);
  }
execute3=true;
```

```
break;
case 0xFF10EF: Serial.println("4");
if (execute4==false )
  {
  HomeToFour();
  delay(500);
  FourToHome();
  delay(500);
  }
execute4=true;
break;

case 0xFF38C7: Serial.println("5");
if (execute5==false )
  {
  HomeToFive();
  delay(500);
  FiveToHome();
  delay(500);
  }
execute5=true;
break;

case 0xFF5AA5: Serial.println("6 or 5_2");
 if (execute5_2==false )
  {
  HomeToFive2();
  delay(500);
  Five2ToHome();
  delay(500);
  }
execute5_2=true;
break;

case 0xFF42BD: Serial.println("7 or 5_3");
if (execute5_3==false )
  {
  HomeToFive3();
  delay(500);
  Five3ToHome();
  delay(500);
  }
execute5_3=true;
break;


case 0xFF4AB5: Serial.println("8 or 5_4");
if (execute5_4==false )
  {
  HomeToFive4();
  delay(500);
  Five4ToHome();
  delay(500);
  }
  execute5_4=true;
  break;

  case 0xFF52AD: Serial.println("9 reset execute bools to false");

  break;

  case 0xFFFFFFFF: Serial.println(" REPEAT");

  break;

  default:
  Serial.println(" other button   ");
  }// End Case

} //END translateIR

void HomeToOne()
{
int i;

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,115);
  delay(500);
  slowServo2(85,97);
  delay(500);
  slowServo3(45,21);// Added to Raise Arm
  delay(500);
  slowServo4(108,92);
  delay(500);
  slowServo5(80,108);
  delay(500);
  slowServo6(152,120);
  delay(500);
  slowServo3(21,11);
  delay(500);
  slowServo6(120,152); //close J5
  delay(500);


}

void OneToHome()
{
  slowServo3(11,45);// changed from (12,21)
  delay(500);
  slowServo6(152,152);
  delay(500);
  slowServo5(108,80);
  delay(500);
  slowServo4(92,108);
  delay(500);
  slowServo2(97,85);
  delay(500);
  slowServo1(115,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);// Added to lower Arm
  delay(500);

}
void HomeToTwo()
{
int i;

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,101);
  delay(500);
  slowServo2(85,94);
  delay(500);
  slowServo3(45,21);// Added to Raise Arm
  delay(500);
  slowServo4(108,92);
  delay(500);
  slowServo5(80,91);
  delay(500);
  slowServo6(152,120);
  delay(500);
  slowServo3(21,8);
  delay(500);
  slowServo6(120,152);
  delay(500);

}

void TwoToHome()
{
slowServo3(8,21);
  delay(500);
slowServo6(152,152);
  delay(500);
slowServo5(91,80);
  delay(500);
slowServo4(92,108);
  delay(500);
slowServo3(21,45);// Added to Raise Arm
  delay(500);
slowServo2(94,85);
  delay(500);
slowServo1(101,92);
  delay(500);
Home_Task2();
delay(500);
slowServo3(45,0);

}
```

```
void HomeToThree()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,83);
  delay(500);
  slowServo2(85,97);
  delay(500);
  slowServo3(45,23);// Added to Raise Arm
  delay(500);
  slowServo4(108,92);
  delay(500);
  slowServo5(80,79);
  delay(500);
  slowServo6(152,120);
  delay(500);
  slowServo3(23,10);
  delay(500);
  slowServo6(120,152);
  delay(500);

}

void ThreeToHome()
{

  slowServo3(10,23);
  delay(500);
  slowServo6(152,152);
  delay(500);
  slowServo5(79,80);
  delay(500);
  slowServo4(92,108);
  delay(500);
  slowServo3(23,45);// Added to Raise Arm
  delay(500);
  slowServo2(97,85);
  delay(500);
  slowServo1(83,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);

}


void HomeToFour()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,69);
  delay(500);
  slowServo2(85,97);
  delay(500);
  slowServo3(45,23);// Added to Raise Arm
  delay(500);
  slowServo4(108,92);
  delay(500);
  slowServo5(80,64);
  delay(500);
  slowServo6(152,120);
  delay(500);
  slowServo3(23,12);
  slowServo6(120,152);
  delay(500);

}

void FourToHome()
{

  slowServo3(12,23);
  delay(500);
  slowServo6(152,152);
  delay(500);
  slowServo5(64,80);
  delay(500);
  slowServo4(92,108);
  delay(500);
  slowServo3(23,45);// Added to Raise Arm
  delay(500);
  slowServo2(97,85);
  delay(500);
  slowServo1(69,92);
  delay(500);
  Home_Task2();
  delay(500);
  slowServo3(45,0);

}


void HomeToFive()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,92);
  delay(500);
  slowServo2(85,100);
  delay(500);
  slowServo3(45,28);// Added to Raise Arm
  delay(500);
  slowServo4(108,67);
  delay(500);
  slowServo5(80,80);
  delay(500);
  slowServo3(28,8);
  delay(500);
  slowServo6(152,120);
  delay(500);

}

void FiveToHome()
{

slowServo3(8,28);
  delay(500);
slowServo6(120,152);
  delay(500);
slowServo5(80,80);
  delay(500);
slowServo4(67,108);
  delay(500);
  slowServo3(28,45);// Added to Raise Arm
  delay(500);
  slowServo2(100,85);
  delay(500);
  slowServo1(92,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);

}

void HomeToFive2()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,92);
  delay(500);
  slowServo2(85,100);
  delay(500);
  slowServo3(45,30);// Added to Raise Arm
  delay(500);
  slowServo4(108,78);
  delay(500);
  slowServo5(80,80);
  delay(500);
  slowServo3(30,19);
  delay(500);
  slowServo6(152,120);
  delay(500);

}
```

```
void Five2ToHome()
{
slowServo3(19,30);
  delay(500);
slowServo6(120,152);
  delay(500);
slowServo5(80,80);
  delay(500);
slowServo4(78,108);
  delay(500);
  slowServo3(30,45);// Added to Raise Arm
  delay(500);
  slowServo2(100,85);
  delay(500);
  slowServo1(92,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);
}


void HomeToFive3()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,92);
  delay(500);
  slowServo2(85,97);
  delay(500);
  slowServo3(45,40);// Added to Raise Arm
  delay(500);
  slowServo4(108,76);
  delay(500);
  slowServo5(80,80);
  delay(500);
  slowServo3(40,16);
  delay(500);
  slowServo6(152,120);
  delay(500);

}

void Five3ToHome()
{
slowServo3(16,40);
  delay(500);
slowServo6(120,152);
  delay(500);
slowServo5(80,80);
  delay(500);
slowServo4(76,108);
  delay(500);
  slowServo3(40,45);// Added to Raise Arm
  delay(500);
  slowServo2(97,85);
  delay(500);
  slowServo1(92,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);
}


void HomeToFive4()
{

  slowServo3(0,45);
  Home_Task2();
  slowServo1(92,92);
  delay(500);
  slowServo2(85,91);
  delay(500);
  slowServo3(45,43);// Added to Raise Arm
  delay(500);
  slowServo4(108,76);
  delay(500);
  slowServo5(80,80);
  delay(500);
  slowServo3(43,14);
  delay(500);
  slowServo6(152,120);
  delay(500);

}

void Five4ToHome()
{
slowServo3(14,43);
  delay(500);
slowServo6(120,152);
  delay(500);
slowServo5(80,80);
  delay(500);
slowServo4(76,108);
  delay(500);
  slowServo3(43,45);// Added to Raise Arm
  delay(500);
  slowServo2(91,85);
  delay(500);
  slowServo1(92,92);
  delay(500);
  Home_Task2();
  slowServo3(45,0);
}
//******************Home Position**********************
```

```
void Home_Task2()
{


  myservo1.write(92); // Joint 1 89 (home)
  myservo2.write(85); // Joint 2 107 (home)
  myservo3.write(45); // Joint 3 24 (home)
  myservo4.write(108); // Joint 82 (home)
  //myservo4.write(92); // Joint 82 (home)
  myservo5.write(80); // Joint 5 84 (home)
  myservo6.write(152); // Joint 6 1666 (home) closed
}

void setupHome()
{
 myservo1.attach(9);//Joint 1
 myservo1.write(92); // Joint 1 89 (home)
 delay(500);
 myservo2.attach(8);//Joint 2
 myservo2.write(85); // Joint 2 107 (home)
 delay(500);
 myservo3.attach(7);//Joint 3
 myservo3.write(3);
 //slowServo3(4,90);
 delay(500);
 //slowServo3(0,45);
 myservo4.attach(6);// Joint 4
 myservo4.write(108);
 //slowServo4(108,58);// enable for horizonal scan
 delay(500);
 myservo5.attach(5);// Joint 5
 myservo5.write(80); // Joint 5 84 (home)
 delay(500);
 myservo6.attach(4); // Joint 6
 myservo6.write(163); // Joint 6 1666 (home) closed
 delay(500);
}


//*****Servo Movement Controls***************************
void slowServo1(int degInitial, int degFinal)
{
int i;

if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(50);
    myservo1.write(i);
    }
  }
else
  {
```

```
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo1.write(i);
    }
  }
  lcd.setCursor(0,0);
  lcd.print("     ");
  lcd.setCursor(0,0);
  lcd.print("1=");
  lcd.setCursor(2,0);
  lcd.print(i);
}
void slowServo2(int degInitial, int degFinal)
{
int i;
if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(100);
    myservo2.write(i);
    }
  }
else
  {
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo2.write(i);
    }
  }
  lcd.setCursor(6,0);
  lcd.print("     ");
  lcd.setCursor(6,0);
  lcd.print("2=");
  lcd.setCursor(8,0);
  lcd.print(i);
}



void slowServo3(int degInitial, int degFinal)
{
int i;
if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(50);
    myservo3.write(i);
    }
  }
else
```

```
  {
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo3.write(i);
    }
  }
  lcd.setCursor(0,1);
  lcd.print("     ");
  lcd.setCursor(0,1);
  lcd.print("3=");
  lcd.setCursor(2,1);
  lcd.print(i);
}

void slowServo4(int degInitial, int degFinal)
{
int i;
if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(50);
    myservo4.write(i);
    }
  }
else
  {
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo4.write(i);
    }
  }
  lcd.setCursor(6,1);
  lcd.print("     ");
  lcd.setCursor(6,1);
  lcd.print("4=");
  lcd.setCursor(8,1);
  lcd.print(i);

}

void slowServo5(int degInitial, int degFinal)
{
int i;
if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(50);
    myservo5.write(i);
    }
  }
```

```
else
  {
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo5.write(i);
    }
  }
  lcd.setCursor(0,0);
  lcd.print("     ");
  lcd.setCursor(0,0);
  lcd.print("5=");
  lcd.setCursor(2,0);
  lcd.print(i);
}
void slowServo6(int degInitial, int degFinal)
{
int i;
if(degInitial<degFinal)
  {
  for (i=degInitial; i<degFinal; i++)
    {
    delay(50);
    myservo6.write(i);
    }
  }
else
  {
  for (i=degInitial; i>degFinal; i--)
    {
    delay(50);
    myservo6.write(i);
    }
  }
  if (i>150)
  {
   lcd.setCursor(13,0);
  lcd.print("   ");
  lcd.setCursor(13,0);
  lcd.print("6=");
  lcd.setCursor(15,0);
  lcd.print("C");
  }
  else if(i<125)
  {
  lcd.setCursor(13,0);
  lcd.print("   ");
  lcd.setCursor(13,0);
  lcd.print("6=");
  lcd.setCursor(15,0);
  lcd.print("O");
  }
}
```

```
//**********Sets Fine Adjustment Using Trim Pots**************


void Analog_Set_Task1()
{
 //***************Home Position***************************
   //J1=90, J2=85, J3=0, J4=103, J5=83, J6=162  (closed)


   analogA1=analogRead(A1);
   Joint1= map(analogA1, 0, 1023, 0, 180);
   myservo1.write(Joint1);
   lcd.setCursor(0,0);
    lcd.print("     ");
    lcd.setCursor(0,0);
    lcd.print("1=");
    lcd.setCursor(2,0);
    lcd.print(Joint1);

   analogA2=analogRead(A2);
   Joint2= map(analogA2, 0, 1023, 0, 180);
   myservo2.write(Joint2);
   lcd.setCursor(6,0);
    lcd.print("     ");
    lcd.setCursor(6,0);
    lcd.print("2=");
    lcd.setCursor(8,0);
    lcd.print(Joint2);

   analogA3=analogRead(A3);
   Joint3= map(analogA3, 0, 1023, 0, 180);
    myservo3.write(Joint3);
    lcd.setCursor(0,1);
    lcd.print("     ");
    lcd.setCursor(0,1);
    lcd.print("3=");
    lcd.setCursor(2,1);
    lcd.print(Joint3);

   analogA4=analogRead(A0);
    Joint4= map(analogA4, 0, 1023, 0, 180);
    myservo4.write(Joint4);
    lcd.setCursor(6,1);
    lcd.print("     ");
    lcd.setCursor(6,1);
    lcd.print("4=");
    lcd.setCursor(8,1);
    lcd.print(Joint4);

   analogA5=analogRead(A7);
   Joint5= map(analogA5, 0, 1023, 0, 180);
    myservo5.write(Joint5);
    lcd.setCursor(13,0);
    lcd.print("   ");
    lcd.setCursor(13,0);
    lcd.print(Joint5);

   analogA6=analogRead(A6);
   Joint6= map(analogA6, 0, 1023, 0, 180);
   myservo6.write(Joint6);
}
//**********Display Requirement***********************
void addZero(int setNumber,int column, int row)
{
lcd.setCursor(column, row);

if (setNumber >= 0 && setNumber < 10)
  {
    lcd.print("0");
    lcd.setCursor(column+1,row);
    lcd.print(setNumber);
  }
  else
  {
  lcd.print(setNumber);
  }
}
```