**硬件设计大赛终审报告**

**作品名称：手势控制星云投影**

**队伍成员：赵英竹、吴荻、黄舒睿**

目录

[一、作品创意及功能特点介绍 2](#_Toc82345176)

[二、所用模块和器件： 3](#_Toc82345177)

[三、各模块具体方法介绍 3](#_Toc82345178)

[四、系统调试过程 5](#_Toc82345179)

[五、展示视频说明 6](#_Toc82345180)

[六、作品效果评估总结 6](#_Toc82345181)

[七、原创性声明及分工 7](#_Toc82345182)

[附录 源代码 7](#_Toc82345183)

# 一、作品创意及功能特点介绍

《星空日记》的结尾，何晓东的毕设用投影展示了星团，他用一系列手部动作控制星团上下左右移动、旋转、伸缩等，我们希望模仿这一段视频，设计出一个能够检测手部姿态，并随手部姿态改变投影灯的灯光方向，进而改变星云图案的星空投影灯。这种设计也可以有很多实际应用，在现实生活中，很多舞台的灯都是连接在机械臂上，靠灯光师移动机械臂来改变灯光方向，如果可以直接用手势控制灯光的移动和指向，调试灯光的过程将会轻松很多。

在我们的硬设作品中，我们设计制作了一个可以检测手部姿态的手套，和一个带有LED灯的三舵机云台。戴上手套之后，通过改变手部姿态（包括全方位转动手掌，张开握拳等），控制云台，将灯光打向指定的位置。在云台外围套上镂刻有星空图案的灯罩，最终在墙壁上投影出星云的变化。

手势控制星空投影灯具体可实现以下功能：

1、通过手掌方向，改变灯的指向，进而改变投影出的星云位置。水平向转动手掌，抬起和落下手掌，手套会利用粘贴在手背位置上的陀螺仪解算手部姿态，感知手掌的方向。云台下方的两个舵机会一起转动，带动机械臂，将灯光打向指定的方向。灯光指向会随手部的运动上下左右移动，灯光指向可以做到与手背的方向完全一致。星云投影的位置也会随之变化；

2、通过手指姿态，让星云旋转。改变手指的姿态，手套会利用粘贴在手指上的弯曲度传感器感知手指的弯曲程度，云台上最上方的舵机会随手指的弯曲程度而转动不同的角度，也会根据手指的张开和握紧改变旋转方向，手张开时星云投影将按顺时针旋转，手握拳时星云投影将按逆时针旋转，转动的度数由手指的弯曲角度决定；

3、通过摇杆控制灯光的颜色和切换模式。三个方向的摇杆可以分别打开三种颜色的灯光，可选择点亮单色光、双色光或者三色光，剩下一个方向可以关闭三种颜色的光（x轴正方向亮红灯，负方向亮绿灯，y轴负方向亮蓝灯，正方向关灯）。手在运动或者做动作时，可以随时利用摇杆控制各色灯光的亮灭。按下摇杆可以切换模式，默认为模式一（通过手掌方向，改变灯的指向），按下复位键可以切换到模式二（通过手指姿态，转动星云），再次按下可以恢复为模式一，实现平移和旋转的切换。

# 二、所用模块和器件：

Arduino Uno开发板\*2、蓝牙（HC-05）\*2、弯曲度传感器（flex4.5、flex2.2）、陀螺仪（MPU9250）、舵机（MG90S）\*3、RGBLED灯\*1、游戏摇杆模块\*1、机械零部件（各种形状铁片、齿轮若干）、手套\*1。

# 三、各模块具体方法介绍

手势控制投影仪整体完成度高，主要由两大部分组成。一部分感知并收集手部数据，控制灯的亮灭，切换各种模式，为控制端。另一部分控制主装置的云台运动，两部分的数据通过两个蓝牙通信传输。

1. **感知和收集手部数据、切换模式部分：**

这部分包括一块Arduino Uno开发板，摇杆模块、蓝牙、LED灯、MPU9250陀螺仪、两个弯曲度传感器。

1. 陀螺仪功能的实现：

陀螺仪贴于手背中部，用于实现模式一（系统启动时默认状态）：星云投影位置的改变。当陀螺仪的X轴指向东方，Y轴指向北方，Z轴指向天时，陀螺仪输出的三个角度值均为0°。改变陀螺仪方向，在较准和姿态解算之后，可以通过读取陀螺仪传来的和角度有关的数据，进行换算后得到陀螺仪转动的角度值（以欧拉角的形式），通过将陀螺仪的角度线性对应到舵机旋转的范围之内，将陀螺仪的极限值与舵机的极限值相对应，得到下方两个舵机应该旋转的角度，并通过蓝牙发送给舵机。

1. 弯曲度传感器功能的实现：

两个弯曲度传感器分别贴于大拇指和无名指处，用于实现模式二：星云投影的旋转及旋转方向的改变。弯曲度传感器可以返回一个其表面可变电阻的阻值，这一阻值随弯曲度数的变化是线性的，可在使用前用模拟引脚读出0度和90度时的阻值，使用时将阻值与这两个值对比推算得弯曲度数，这样就可以通过当前电阻值得到大拇指和无名指的弯曲角度。两个角度再按照两个传感器的精确程度（即阻值差值大小）加权平均，减小误差，最终得到较为精确的弯曲角度，通过蓝牙发送给上方舵机。

1. 摇杆模块功能的实现：

摇杆的按动与向四个方向的移动用来控制模式的切换以及灯的亮暗与颜色。系统启动时，默认为模式一。按下摇杆按钮，程序会通过蓝牙给云台输出一个切换模式的信号，进行模式一、二的切换。摇动摇杆时，向x轴正方向亮红灯，负方向亮绿灯，y轴负方向亮蓝灯，正方向关灯。将摇杆先后摇向多个方向，灯会显示出这几个方向对应颜色的复合光；将摇杆遥向y轴正方向，三种灯同时关闭。这一部分主要实现了星云模式的切换以及星云投影颜色的改变。

4、蓝牙功能的实现：

此部分蓝牙为主端，在运行模式1时，通过蓝牙执行一次loop函数传输两个舵机需要转动的角度值（每次发送会设置延迟，以保证从端接收到消息有时间移动到相应位置，不至于在使用过程中出现手部动作比云台快很多的情况）。在运行模式二时，执行一次loop函数传输一个舵机转动的角度值，同样也设置了延迟。在按下摇杆按钮时，会通过蓝牙想从端发出信号，保证两方同时切换模式。

1. **云台运动部分：**

这部分包括一块Arduino Uno开发板，蓝牙和三个舵机。

1. 云台搭建：

整个云台由三个舵机以及一些L形、U形、“一”字形的支架，转盘，齿轮，自行组装而成，LED灯通过与转盘相连固定在最高层舵机上。云台整体结构美观，能保证在三个舵机转动时有较好的稳定性。

1. 舵机功能的实现：

云台的运动由三个舵机相互协调配合。舵机转动的实现调用了servo库，通过调试PWM输出时的毫秒值，使其可以在0-180度范围内转动。其中最底层舵机的转动对应陀螺仪的水平转动，也即手部沿x、y方向的运动；中层舵机的转动对应陀螺仪在竖直方向的运动，也即手部沿x、z方向的运动；最高层舵机的转动对应弯曲度传感器不同程度的弯曲角度。模式一中底层和中层的舵机同时转动，达到改变星云投影的位置的目的，舵机每次转动一度会暂停15ms，这样保证舵机转动较为平缓。高层的舵机带动转盘和LED灯一起转动，达到使星云全方位快速旋转的目的。

1. 蓝牙功能的实现：

通过蓝牙接收从主端传输过来的数据，从而控制三个舵机的运动，实现人机交互。

# 四、系统调试过程

在对系统进行调试的过程中，我们发现并解决了很多问题。

在各部分分开调试过程中，我们发现舵机在转动时并不会覆盖到0-180度，而是只能转动140度左右，我们通过查阅函数功能，利用map函数将PWM输出对应到0-180度的毫秒值，解决了这一问题。在调试陀螺仪时，我们发现陀螺仪输出值不准且不够稳定，会出现自增的效果，我们首先对陀螺仪进行了磁场的校准。之后查找了有关陀螺仪姿态解算的资料，并通过map函数直接将输出值匹配到舵机的角度值，解决了陀螺仪输出不稳定导致云台姿态与手部姿态的匹配不够精确的问题。

在进行联调的过程中，我们发现很多相互干扰的问题。在同一块uno板上连接舵机、LED灯和蓝牙后，从端蓝牙总会无故断开。查找资料后我们发现，舵机在复位和快速移动时，输出电流过大，将电源电压拉低，从而会使得蓝牙重启，与主端蓝牙断开。我们通过反复调试，找到了供电的最大值，通过将舵机移动过程变平缓，并将LED灯转移到从端，蓝牙断电的问题得到了解决。起初我们将摇杆模块的Rx、Ry和button分别放在A3、A4和A5引脚，但是发现陀螺仪失效了，经过调试，我们发现占用A4和A5串口会使得I2C通信受到影响，导致陀螺仪的GetAngle函数不能正常工作。通过调整接线，用数字引脚读按键，最终解决了这一问题。在利用摇杆模块的button时，最初的设想是通过attachInterrupt来进行外部中断从而调整云台的转动模式，但利用中断函数时发现中断函数会对蓝牙通信传输产生影响，导致蓝牙传输出现问题，后来经过查阅相关资料，发现进行中断时Arduino板接受的串口数据可能丢失导致蓝牙传输数据出现故障，最终的解决办法是通过digitalRead获得button的电平状态并与上一次的电平状态进行比较，从而判断button是否被按下，解决了这一问题。

# 五、展示视频说明

**视频一：云台运动展示**

首先进行两个蓝牙的连接，连接成功后，云台自动复位。

步骤一：进行模块一调试。在水平方向上转动手掌，观察到云台随着手掌同向运动；接着在竖直方向上先抬起后下压手腕，观察到云台跟随手部动作，先抬起后复位。重复上述动作，云台的方向和手掌的方向非常一致。

步骤二：进行模块二调试。按下摇杆按钮。从模式一切换到模式二，最高层舵机开始转动。手握拳，观察到舵机逆时针旋转；手张开，观察到舵机顺时针转动。重复上述动作，验证准确性。

步骤三：进行LED灯的调试。将摇杆依次往四个方向推动，LED灯亮起红、蓝、绿三色后熄灭。重复上述动作，验证准确性。

步骤四：按下摇杆按钮，云台复位到初始状态，开始再次进行模式一，展示完成。

**视频二：星云投影展示**

根据视频一的步骤，以蓝紫光为例，展示了星云投影位置变换、全方位旋转的效果，得到了较好的视觉美感体验。

（以上视频中同样有字幕说明）

# 六、作品效果评估总结

该作品完成度较高，成功实现了初审报告中计划的全部功能。通过手部动作控制星空投影，操作简单，方便上手；多个单片机联动、蓝牙通信与舵机共同配合实现的云台转动、投影变换又体现出设计应有的技术性。作品在保证一定实用价值的基础上提高了艺术美感与观赏体验，使之兼有机关转动的精巧严谨与星河璀璨的无边浪漫。

在完成该作品的过程中，我们在产品创意和实现方式的设计、硬件基本功能实现和协同过程中都遇到了很多问题。通过反复查阅资料和耐心调试，我们逐个地解决了这些很麻烦的问题，很多的功能比如陀螺仪的姿态解算、舵机的转动等都是在比较了很多同类的产品、比较了大量的库之后得到的最精确的结果，云台的支架也是反复调整，保证机械结构不会影响精确度。产品最终达到了很好的视觉效果，基本和最初的设想一致，没有因为技术困难，而不去实现初审时的设想。通过这次硬设，我们对很多硬件技术，陀螺仪滤波、I2C通信、脉宽调制等的理解也更加深刻了，也更加了解了硬件设计地特点，有了调试的耐心。

# 七、原创性声明及分工

该作品调用了Servo库（舵机）、Wire库、SoftwareSerial库（蓝牙通信）和JY901库（陀螺仪）；作品灵感来源于北京大学招生宣传片《星空日记》。

赵英竹：搭建云台。查找资料，调试弯曲度传感器、舵机。写部分代码。

吴荻：查找资料，调试舵机和陀螺仪，写部分代码。

黄舒睿：查找资料，撰写报告，调试舵机。

# 附录 源代码

主端

#include<Wire.h>

#include<Servo.h>

#include <JY901.h>

#include <SoftwareSerial.h>

#define tx 3

#define rx 4

const int FLEX\_PIN1 = A0; // 电压采集接口

const int FLEX\_PIN2 = A1; // 电压采集接口

const float VCC = 5.0; // 模块供电电压，ADC参考电压为V

const float R\_DIV = 100000.0; // 分压电阻为100KΩ

const float STRAIGHT\_RESISTANCE\_L = 14100.0; // 平直时的电阻值

const float BEND\_RESISTANCE\_L = 23500.0; // 90度弯曲时的电阻值

const float STRAIGHT\_RESISTANCE\_S = 31000.0; // 平直时的电阻值

const float BEND\_RESISTANCE\_S = 66000.0; // 90度弯曲时的电阻值

const int R = 5,G = 6,B = 7;

float average\_angle = 0;

float ab = 0;

float ay, az;

const int SW = 2,Rx = A2,Ry = A3;

SoftwareSerial bt(rx, tx); // RX, TX

int val;

int flag = 0;

int valset = 1;

void setup ( )

{

JY901.StartIIC();

Wire.begin ( );

Serial.begin (9600);

bt.begin(9600);

pinMode(FLEX\_PIN1, INPUT);

pinMode(FLEX\_PIN2, INPUT);

pinMode(SW,INPUT\_PULLUP);

pinMode(R,OUTPUT);

pinMode(G,OUTPUT);

pinMode(B,OUTPUT);

delay (1000);

Serial.println ( "Taking Values from the sensor" );

delay (1000);

//attachInterrupt(0,Isr,FALLING);

}

void loop ( )

{

int x = analogRead(Rx);

int y = analogRead(Ry);

if(x<50)

{

digitalWrite(R, HIGH); //将摇杆向x轴正方向推，亮红灯

}

if(x>976)

{

digitalWrite(G, HIGH); //将摇杆向x轴负方向推，亮绿灯

}

if(y<50)

{

digitalWrite(B, HIGH); //将摇杆向y轴正方向推，亮蓝灯

}

if(y>976)

{

digitalWrite(R, LOW);

digitalWrite(G, LOW);

digitalWrite(B, LOW); //将摇杆向y轴负方向推，灯灭

}

val=digitalRead(SW);

Serial.println(val);Serial.println(valset);

if((val == 0) && (valset!=0))

{if(flag == 0)

{

flag = 1;

bt.write(250);

}

else if(flag == 1)

{

flag = 0;

bt.write(250);

}

delay(1000);

}

valset = val;

if(flag == 0){

JY901.GetAngle();

az = (float)JY901.stcAngle.Angle[2]/32768\*180;

az = map (az, -180, 180, 0, 180);

Serial.print("az:");Serial.println (az);

bt.write(az);

delay (200);

JY901.GetAngle();

ay = (float)JY901.stcAngle.Angle[1]/32768\*180;

ay = map(ay, -90, 90, 0, 180);

Serial.print("ay:");Serial.println(ay);

Serial.println(" ");

bt.write(180-ay);

delay (200);

}

if(flag == 1){

int flexADC1=analogRead(FLEX\_PIN1);

float flexV1 = flexADC1\* VCC / 1024.0;

float flexR1 = R\_DIV \* (VCC / flexV1-1.0);

float angle1 = map(flexR1, STRAIGHT\_RESISTANCE\_L, BEND\_RESISTANCE\_L,0, 90.0);

Serial.println("Bend1: "+String(angle1) +" degrees");

Serial.println();

int flexADC2 = analogRead(FLEX\_PIN2);

float flexV2 = flexADC2\* VCC / 1024.0;

float flexR2 = R\_DIV \* (VCC / flexV2-1.0);

float angle2 = map(flexR2, STRAIGHT\_RESISTANCE\_S, BEND\_RESISTANCE\_S,0, 90.0);

Serial.println("Bend2: "+String(angle2) +" degrees");

Serial.println();

average\_angle = Average\_angle(angle1,angle2);

Serial.println("Average angle: "+String(average\_angle)+" degrees");

Serial.println();

ab = map(average\_angle,0,95,0,180);

ab = abs(ab);

bt.write(ab);

delay(400);

}

}

float Average\_angle(float angle1,float angle2)

{

float Dif\_L,Dif\_S,angle\_ave = 0;

Dif\_L = BEND\_RESISTANCE\_L - STRAIGHT\_RESISTANCE\_L;

Dif\_S = BEND\_RESISTANCE\_S - STRAIGHT\_RESISTANCE\_S;

angle\_ave = Dif\_L / (Dif\_L + Dif\_S) \* angle1 + Dif\_S / (Dif\_L + Dif\_S) \* angle2;

return angle\_ave;

}

从端

#include <SoftwareSerial.h>

#include <Servo.h>

#define tx 2

#define rx 3

SoftwareSerial bt(rx, tx); //RX, TX

Servo myservo1, myservo2, myservo3;

float message;

float ay = 0, az = 0;

float pre\_ay = 0, pre\_az = 0;

float bend = 0;

float pre\_bend = 0;

int readstate = 0;

int state = 0; //0控制下面两个舵机，1控制上面一个舵机

void setup() {

Serial.begin(9600);

bt.begin(9600);

pinMode(tx, OUTPUT);

pinMode(rx, INPUT);

myservo1.attach(11, 250, 4600);

myservo2.attach(9, 250, 4600);

myservo3.attach(10, 250, 4600);

delay(2000);

}

void loop() {

if (bt.available() > 0)

{

message = bt.read();

Serial.println(message);

if (message == 250)

{

if (state == 0)

{

state = 1;

}

else if (state == 1)

{

state = 0;

int pos;

for (pos = pre\_az; pos >= 0; pos -= 1)

{

myservo2.write(pos);

delay(20);

} b

for (pos = pre\_ay; pos >= 0; pos -= 1)

{

myservo3.write(pos);

delay(20);

}

for (pos = pre\_bend; pos >= 0; pos -= 1)

{

myservo1.write(pos);

delay(20);

}

}

Serial.print("change");

}

else if (state == 0)

{

if (readstate == 0)

{

az = message;

int pos;

if (pre\_az < az)

{

for (pos = pre\_az; pos <= az; pos += 1)

{

myservo2.write(pos);

delay(20);

}

}

if (pre\_az > az)

{

for (pos = pre\_az; pos >= az; pos -= 1)

{

myservo2.write(pos);

delay(20);

}

}

Serial.print("az:");

Serial.println(az);

readstate = 1;

pre\_az = az;

}

else if (readstate == 1)

{

ay = message;

int pos;

if (pre\_ay < ay)

{

for (pos = pre\_ay; pos <= ay; pos += 1)

{

myservo3.write(pos);

delay(20);

}

}

if (pre\_ay > ay)

{

for (pos = pre\_ay; pos >= ay; pos -= 1)

{

myservo3.write(pos);

delay(20);

}

}

Serial.print("ay:");

Serial.println(ay);

readstate = 0;

pre\_ay = ay;

}

}

else if (state == 1)

{

bend = message;

int pos;

if (pre\_bend < bend)

{

for (pos = pre\_bend; pos <= bend; pos += 1)

{

myservo1.write(pos);

delay(20);

}

}

if (pre\_bend > bend)

{

for (pos = pre\_bend; pos >= bend; pos -= 1)

{

myservo1.write(pos);

delay(20);

}

}

Serial.print("bend:");

Serial.println(bend);

pre\_bend = bend;

}

}

}