

2023 年全国大学生电子设计竞赛

运动目标控制与自动追踪系统（E 题）



2023 年 8 月 4 日

第 7 组 黄惠齐 黄昊林 王富周

指导老师：钟鹏飞

摘要

随着计算机视觉、图像处理等高新技术的发展以及控制理论的不完善，人们对二轴云台的智能化和自主化要求越来越高，使得二轴云台的应用领域不断扩大。云台视觉作为二轴云台智能化系统的一个重要组成部分，其主要任务就是对摄像机获取的图像信息进行检测与跟踪，另外云台的控制转动性能也对目标检测跟踪的准确性和实时性有重要影响。

系统由 OpenMV 嵌入式计算机视觉平台、STM32 单片机控制模块、摄像头模块、显示屏、人机交互系统以及二轴舵机云台和激光笔构成一个控制系统。摄像头模块用于采集图像信息，经由 OpenMV 进行图像数据处理，并通过 PID 精确算法调节舵机，从而控制激光笔的光斑在屏幕上按照设定程序进行移动。本系统实现了以激光笔的光斑作为运动控制目标的自动追踪系统，包括复位功能、沿屏幕运动、沿 A4 纸边缘运动以及自动追踪光斑等功能。

最后为了验证所提控制策略和目标检测跟踪算法的有效性，根据目标检测跟踪实时性要求，对摄像机、舵机控制板等主要硬件进行选型，搭建了用于动态目标检测跟踪的二轴云台视觉系统，通过实际工况中的二轴云台目标跟踪实验，证明了整套二轴云台视觉系统的实用性。

关键词：二轴云台，OpenMV，运动目标控制、目标检测、目标追踪

Abstract

With the development of computer vision, image processing and other high-newtechnology and the continuous improvement of control theory, people have higher requirementsfor the two-axis Pan/Tilt platform's (PTP) intelligence and autonomy, which makes theapplication field of the PTP continue to expand.The PTP vision is an important part of the PTPintelligent system, its main task is to detect and track the image information obtained by thecamera. Meanwhile, the PTP's control rotation performance has an important influence on theon the accuracy and real-time of target detection and tracking.

The closed-loop control system consists of OpenMV embedded computer vision platform, STM32 microcontroller control module, camera module, display module, human-computer interaction system, two-axis steering machine gimbal and laser pointer. The camera module collects image information, processes the image data through OpenMV, and adjusts the servo through the PID precision algorithm to control the spot of the laser pointer and move on the screen according to the set program. This system realizes the functions of reset, movement along the screen, movement along the edge of A4 paper, automatic tracking of the spot and other functions of the automatic tracking system with the spot of the laser pointer as the motion control target, and can accurately complete the task requirements with good robustness.

Finally,in order to verify the effectiveness of the proposed control strategy and target detection and tracking algorithm, according to the real-time requirements of target detectionand tracking, the main hardware such as cameras and steering gear control boards are selected,and set up the PTP for dynamic target detection tracking. Finally, through the PTP targettracking experiment in the actual working condition, the practicability of the whole PTP systemis proved.

Keywords: Two-Axis Pan/Tilt Platform(PTP), OpenMV, moving target control, target detection, target tracking

目 录

一、 系统方案	错误!未定义书签。
1、嵌入式视觉平台的论证与选择.....	1
2、单片机的论证与选择.....	1
2、电机的论证与选择.....	错误!未定义书签。
3、控制算法的论证与选择.....	2
二、理论分析与计算	2
1、矩形位置检测.....	2
2、二轴云台控制分析.....	3
3、控制算法分析.....	3
4、斜线运动控制.....	3
三、 电路与程序设计	4
1、 电路设计.....	4
(1) 系统总体框图.....	错误!未定义书签。
(2) 硬件连接图.....	错误!未定义书签。
四、 测试方案与测试结果.....	5
1、测试方案.....	5
2、测试条件与仪器.....	5
3、测试结果与分析.....	5
(1) 测试结果.....	5
(2) 测试分析和结论.....	6
五、 结论与心得	6
六、 参考文献	7
附录：核心源程序	8

运动目标控制与自动追踪系统（E 题）

【本科组】第 7 组

一、系统方案

系统方案主要描述了针对研究的要求，采用软硬件结合的方法设计了一个运动目标控制与自动追踪系统。该系统由多个模块组成，包括 OpenMV 嵌入式计算机视觉平台、STM32 单片机控制模块、摄像头模块、显示屏幕、二轴舵机云台和激光笔，构成一个控制系统。摄像头模块负责采集图像信息，然后通过 OpenMV 进行图像数据处理。处理后的图像数据用于控制 PID 算法调节舵机，从而实现对激光笔光斑在屏幕上按照设定程序进行移动的控制。下面分别论证这几个模块的选择。

1、嵌入式视觉平台的论证与选择

方案一：采用 OpenMV 嵌入式计算机视觉平台

OpenMV 是一个基于 MicroPython 的嵌入式计算机视觉平台，使用 ARM 微控制器。开发者可以使用 Python 语言编写图像处理和计算机视觉应用，无需深入了解底层硬件编程。它专注于图像处理和计算机视觉任务，提供丰富的图像处理库和示例代码。OpenMV 配备了图像传感器和其他传感器，可以直接连接摄像头、显示屏、Wi-Fi 模块等设备。OpenMV 作为一个简单易用的嵌入式计算机视觉平台，适合初学者和快速原型开发。它的重点是图像处理和计算机视觉任务，并提供丰富的库和示例，使得开发者能够更快速地实现各种视觉应用。

方案二：采用 K210 嵌入式计算机视觉平台

K210 是一种由中国公司寒武纪（Kendryte）开发的系统级芯片（SoC）。它采用 RISC-V 架构，是一种开源的指令集架构，支持低功耗和高性能。K210 芯片主要用于机器学习和人工智能应用，如物体识别、语音识别等。它在物联网设备、智能摄像头、智能家居等边缘计算设备中广泛应用。K210 适用于专业机器学习和边缘计算领域，具备更高的计算性能和灵活性。综合考虑，选择 openMV 嵌入式视觉平台是更迅速处理的选择，因此本设计采用 OpenMV 嵌入式视觉平台。

2、单片机的论证与选择

方案一：采用 STC51 单片机

STC51 是一种基于 Intel 8051 核心的单片机，采用 CISC 架构。它在嵌入式系统中有着广泛的应用，历史悠久，生态丰富。STC51 单片机相对较为简单，易于学习和使用，适合入门级开发者。它拥有一些基本的外设和功能，但在计算能力和性能方面相对较弱。对于较简单的控制任务和低性能要求的应用，STC51 可以是一个经济实惠的选择。

方案二：采用以 ARM Cortex-M4 为内核的 STM32F407 系列控制芯片

STM32 是一种基于 ARM Cortex-M 核心的单片机，采用 RISC 架构。它具有强大的计算能力和丰富的外设，适用于复杂的嵌入式应用。STM32 单片机有多个系列，不同系列的芯片有着不同的性能和功能，满足不同需求。它支持多种编程语言，包括 C 和 C++，也支持 RTOS 系统，适合中高级开发者。STM32 单片机在性能、功耗、稳定性等方面都有优势，适用于较复杂的控制任务和性能要求较高的应用。

综合考虑，选择使用 STM32 单片机控制舵机云台是更合理的选择，因为它提供了更好的性能，所以本设计选择方案二。

2、电机的论证与选择

采用舵机

舵机是一种特殊类型的直流电机，它与普通直流电机相比具有额外的反馈控制回路，通常是一个位置反馈传感器（如编码器）。舵机可以精确地控制电机的位置，通过控制脉宽信号来实现精确的角度调节，通常具有较高的角度分辨率。舵机适用于需要精确位置控制的场景，如机器人、摄像云台、航模等。舵机的结构也相对简单，成本较低，但由于加入了反馈控制，相比普通直流电机稍微贵一些。

综上所述，选择使用舵机作为云台电机是较为合理的选择。舵机具有精确的位置控制能力和较低的成本，在实现云台旋转和角度调节时具有优势。

3、控制算法的论证与选择

方案一：采用开环算法

开环控制是一种简单的控制方法，其中输出信号不受系统反馈信息的影响。在开环控制中，输入信号直接作为控制器的输出信号，用于云台电机控制。开环控制没有对输出信号进行监测和调整的机制，因此无法纠正外部干扰和内部误差，导致系统的稳定性和精确性较差。在开环控制中，系统的性能高度依赖于模型的准确性和环境条件的稳定性。对于复杂系统或存在不确定性的环境，开环控制的性能可能会受到较大影响。

方案二：采用闭环控制

闭环控制是一种基于反馈的控制方法，其中输出信号受系统反馈信息的影响。在闭环控制中，输出信号被反馈给控制器，并与期望信号进行比较，然后根据误差来调整控制器输出，以实现输出信号的精确控制。闭环控制可以实时纠正外部干扰和内部误差，提高系统的稳定性和精确性，使系统能够更好地适应不确定性和变化的环境条件。在闭环控制中，采用合适的控制算法（如 PID 控制算法）可以优化系统的性能，使系统更快地响应控制信号，减少超调和稳态误差。PID 控制算法按比例、积分、微分的函数关系，进行运算，将其运算结果用以输出控制。优点是控制精度高，是控制系统非常普遍的运算方法。对于本系统的控制已足够精确。

综上所述，综合使用开环和闭环控制，并采用 PID 算法控制激光笔是较为合理的选择。闭环控制能够实时纠正误差，提高系统的稳定性和精确性，而 PID 算法能够优化系统的性能，使激光笔具有快速响应和灵活性，适用于实现精确的运动控制。

二、理论分析与计算

1、矩形位置检测

OpenMV 为开发者提供了 Python API 和丰富的图像处理库，这使得实现矩形检测变得相对简单和高效。通过结合 OpenMV 的强大功能和易用性，可以轻松实现矩形检测功能。

OpenMV 对采集到的图像进行预处理，例如灰度化、二值化等。这些预处理步骤有助于减少图像中的噪声并简化后续的图像处理。在预处理后的图像上执行边缘检测算法，例如 Canny 边缘检测，以识别矩形的边缘。基于边缘检测的结果，进行轮廓检测，找出图像中的所有轮廓。对检测到的轮廓进行筛选，通过设置阈值和条件来排除不符合矩形特征的轮廓。对经过筛选的轮廓进行拟合，找出最接近的矩形。OpenMV 通常使用旋转矩形（rotated rectangle）来拟合，因为它可以适应旋转的矩形。

2、二轴云台控制分析

二轴云台是一种常见的云台类型，它可以在两个轴（通常是水平和垂直方向）上进行转动。二轴云台广泛应用于摄像机、激光雷达、传感器等设备的定向控制，以及机器人、无人机等需要精确转动的系统中。

水平轴通常用于实现云台在水平方向上的转动。垂直轴通常用于实现云台在垂直方向上的转动。

3、控制算法分析

采用 PID 算法来光斑在屏幕的位置。开始工作后，实时检测屏幕上光斑的位置，并与设定的目标位置比较，进行 PID 控制。其 PID 结构原理如图 1，PID 分别表示光斑的位置比例（P）、位置误差（I）、位置积分（D）。

P: 对目标位置与当前位置偏差 $e(t)$ 进行调整，系数越大调节速度越快，减小误差，但是过大的比例，会造成舵机速度状态的突变，从而导致云台状态不稳定

I: 加入积分调节，可以消除系统的稳态误差，提高无误差度。系统的稳定性下降，动态响应变慢。

D: 微分调节反应的是光斑的角速度，可以预见偏差变化的趋势具有可预见性因而可以产生超前调节，加入微分调节可以改善系统的动态性能。

PID 控制器由比例单元（P）、积分单元（I）和微分单元（D）组成。其 PID 控制器的传递函数为：

$$K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

其中， K_p , K_i 和 K_d 为 PID 控制器的比例，积分和微分参数。

4、斜线运动控制

设 $A(X_1, Y_1)$ 为起点， $B(X_2, Y_2)$ 为终点。设点 $C(X_3, Y_3)$ 为目标点。

现在对 AB 两点之间进行插值，插值数量为 N 。

$$dx = \frac{X_2 - X_1}{N}, \quad dy = \frac{Y_2 - Y_1}{N}$$

那么 C 的轨迹就是： $\{X_1 + 1dx, X_1 + 2dx, \dots, X_1 + Ndx\}$ 。一系列轨迹点设置好后，接

下来识别到激光笔光斑，对比误差。设识别光斑到的点为 $P(X_4, Y_4)$ ，那么计算设置 PC 的

欧氏距离为 $D = \|P - C\|_2^2$ 。然后进行 PID 控制。当 PC 足够近， D 小于设定阈值，就认为是重合了，目标点切换到下一个点。

三、电路与程序设计

1、电路设计

(1) 系统总体框图

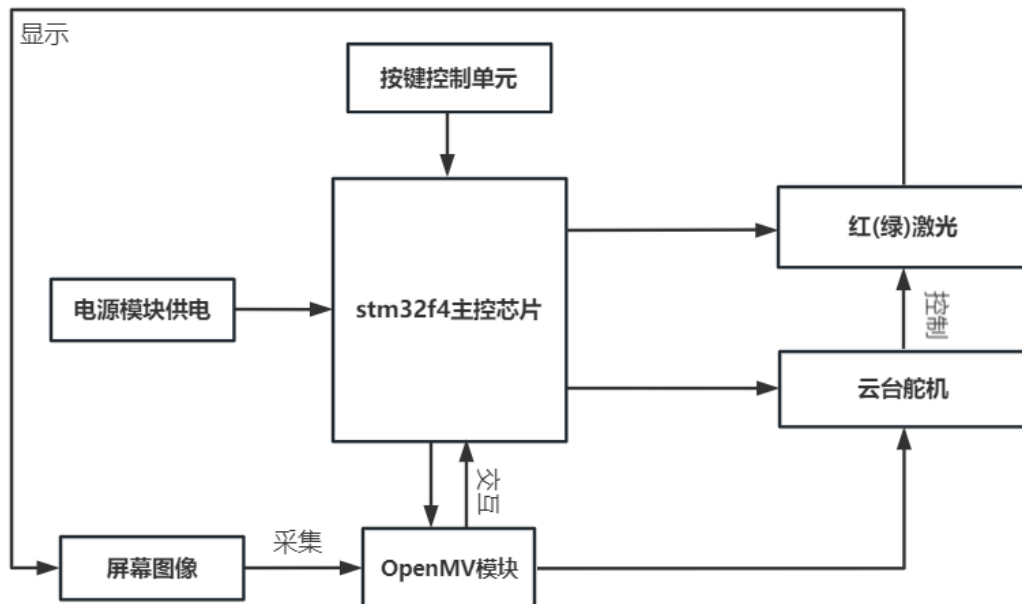


图 1 系统总体框图

(2) 硬件连接图

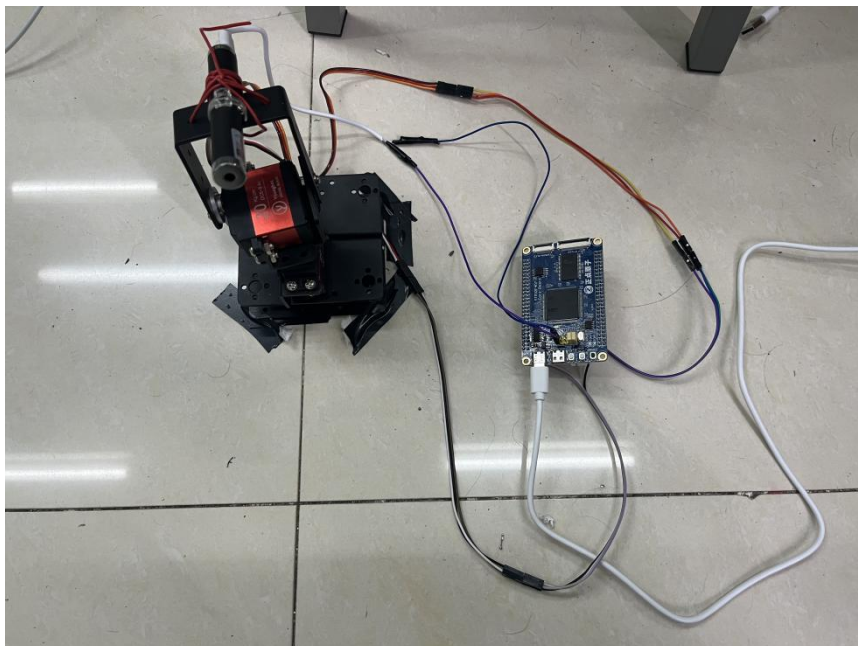


图 2 硬件连接图

2、程序结构与设计

控制二轴云台控制激光笔在屏幕上精准移动时，程序结构如下：

初始化： 在程序开始时，进行系统初始化，包括初始化 OpenMV、舵机、Stm32 单片机和通信接口等。

设定目标位置：在程序中设定激光笔在屏幕上的目标位置。目标位置通过 OpenMV 识别到矩形后，通过计算获得。

主循环： 进入主循环，程序会不断执行以下步骤：

- a. 获取传感器反馈： 读取舵机的位置信息，获取当前云台的姿态。
- b. 计算误差： 将目标位置与当前位置进行比较，计算出误差。即，目标位置与实际位置之间的差距。
- c. PID 控制器计算： 将误差输入 PID 控制器，根据 PID 算法计算出控制信号。PID 控制器会根据比例系数 K_p 、积分系数 K_i 和导数系数 K_d 对误差进行加权计算，得到控制信号。
- d. 控制信号输出： 将 PID 计算得到的控制信号转换为 PWM 信号，并发送给舵机，控制云台的转动。
- e. 限制和安全检查： 对于舵机或电机的控制信号，进行范围限制，确保云台不超出安全转动范围，并避免碰撞。
- f. 重复运行： 以上步骤会在循环中不断运行，直到激光笔移动到设定的目标位置。在运动到目标位置后，控制器可以进入静止状态或继续接收新的目标位置。

优化和调试： 进行系统调试和优化，根据实际情况调整 PID 参数（ K_p 、 K_i 、 K_d ）以及采样频率等参数，使云台运动更加稳定和精确。

中断程序： 当按下中断按钮时，程序会进行中断，舵机停止运动。

四、测试方案与测试结果

1、测试方案

- (2) 硬件测试：测试机械部分能否正常工作，各项参数是否准确。同时，检查电路焊接是否正确，各部分接线是否正确。
- (3) 程序测试：用 keil5、OpenMV IDE 软件编程并通过单片机多次运行调试。
- (4) 硬件软件联调：用实现的编程算法控制硬件系统进行工作。检测整个系统是否实现各项要求。

2、测试条件与仪器

- (2) 测试条件：检查多次，机械部分必须正常工作且参数正确。仿真电路和硬件电路必须与系统原理图完全相同，并且检查无误，硬件电路保证无虚焊。
- (3) 测试仪器：卷尺、秒表。

3、测试结果与分析

1、测试结果

(1) 基本要求的测试

- 1) 设置运动目标位置复位功能。执行此功能，红色光斑能从屏幕任意位置回到原点。光斑中心距原点误差 $\leq 2\text{cm}$

表 B-1 基础部分 1 测试结果

次数	1	2	3	4	5
距离 (cm)	0.8	0.5	0.1	0	0

2) 启动运动目标控制系统。红色光斑能在 30 秒内沿屏幕四周边线顺时针移动一周，移动时光斑中心距边线距离 $\leq 2\text{cm}$ 。

表 B-2 基础部分 2 测试结果

次数	1	2	3	4	5
转动用时间 (s)	11.6	11.5	11.7	11.4	11.8

3) 用约 1.8cm 宽的黑色电工胶带沿 A4 纸四边贴一个长方形，构成 A4 靶纸。将此 A4 靶纸贴在屏幕自定的位置。启动运动目标控制系统，红色光斑能在 30 秒内沿胶带顺时针移动一周。超时不得分，光斑完全脱离胶带一次扣 2 分，连续脱离胶带移动 5cm 以上记 0 分。

表 B-3 基础部分 3 测试结果

次数	1	2	3	4	5
转动用时 (s)	6.4	6.3	6.5	6.6	6.2

4) 将上述 A4 靶纸以任意旋转角度贴在屏幕任意位置。启动运动目标控制系统，要求同 (3)。

自由发挥：利用充电宝供电，实现现代供电技术的使用

2、测试分析和结论

根据上述测试数据，可以得出以下结论：

- (1) 能够很好的完成基本功能和发挥功能。
- (2) 完成任务的时间较快。

综上所述，本设计基本达到设计要求基础部分(1)、(2)、(3)的要求。

五、结论与心得

此次电子设计大赛我们收获颇丰，在整个过程中，我们不仅在通力合作中体会到了团结的重要性。还将理论与实践结合了起来，培养了一定的科研能力，拓宽了知识面。一步一步的解决相应的任务；逐步的实现了任务需求。展望未来，智能车技术必将在更广阔的领域广泛应用，。

在开始设计之前，我们充分理解题目的要求和限制条件，明确了目标是实现云台舵机控制激光运动和自动追踪。构思一个系统架构，包括硬件和软件组件，以及它们之间的通信方式。

在硬件选择与集成上，我们选择了适合的云台舵机、激光模块以及其他所需传感器和电子元件，确保它们能够在物理上集成在一起，同时也考虑了供电和电路连接等问题。

我们学习了云台舵机的控制原理，了解如何通过 PWM 信号或其他方式来控制舵机的角度。在目标检测与追踪算法方面，我们钻研出合适的目标检测和追踪算法用于我们的控制系统。

无论比赛结果如何，从这个项目中学到的知识和经验都是宝贵的。总结项目中遇到的问题和解决方案，为以后的项目积累经验。在这比赛的最后，要感谢学校和学院老师的给力的后勤工作，使我们能够在不被烦扰的情况下，用最充足的时间完成比赛。感谢全国大学生电子设计大赛的组委会，提供这样优秀的平台和机会给我们充足的锻炼。

六、参考文献

- [1] 谭浩强.C 语言程序设计[M].北京:清华大学出版社,2012.
- [2] 黄智伟. 全国大学生电子设计竞赛训练教程. 北京: 电子工业出版社, 2005.
- [3] 王福瑞等. 单片微机测控系统设计大全[M]. 北京航空航天大学出版社,1998(331—337).
- [4] 杨俊, 陈霞, 石健. (M). "基于 PID 控制的云台跟踪系统设计." 自动化与仪器仪表,2019.
- [5] 杨成龙, 龙军, 朱海滨. (M). "运动目标跟踪系统的设计与实现." 电子工程与应用, 2018.

附录：核心源程序

主函数：int main(void)

```
{u8 key;           //保存键值
  delay_init(168); //初始化延时函数
  NVIC_PriorityGroupConfig(NVIC_
PriorityGroup_2);
  LED_Init(); //初始化 LED 端口
  KEY_Init();
  Servo_PWM_Init();
  JG_Init();
/**直接操作库函数实现 IO 控制**/
while(1)
{
  key=KEY_Scan(0); //得到键值
  if(key)
  {
    switch(key)
    {
      case WKUP_PRES:
        i++;
        break;
      case KEY0_PRES:
        j=1;
        break;
    }
  }else delay_ms(10);
  if(j==1)
  {TIM_SetCompare1(TIM3,1415);
    TIM_SetCompare2(TIM3,1820);
  }
  if(i==1)
  {
    switch(state)
    {
      case 0:
        do
        {
          x=x-1;
          delay_ms(10);
          TIM_SetCompare1(TIM3,x);
```

```
          TIM_SetCompare2(TIM3,1940);
        }while(x>=1244);
        state=1,j=0;
        break;
      case 1:
        do
        {
          y=y-1;
          delay_ms(10);
          TIM_SetCompare1(TIM3,1244);
          TIM_SetCompare2(TIM3,y);
        }while(y>=1714);
        state=2,j=0;
        break;
      case 2:
        do
        {
          x=x+1;
          delay_ms(10);
          TIM_SetCompare1(TIM3,x);
          TIM_SetCompare2(TIM3,1705);
        }while(x<=1595);
        state=3,j=0;
        break;
      case 3:
        do
        {
          y=y+1;
          delay_ms(10);
          TIM_SetCompare1(TIM3,1595);
          TIM_SetCompare2(TIM3,y);
        }while(y<=1935);
        state=4,j=0;
    }
  }
```

```

    }
    if(i==2)
    {
        switch(state1)
        {
            case 0:
                do
                {
                    x1=x1-1;
                    delay_ms(10);
TIM_SetCompare1(TIM3,x1);

TIM_SetCompare2(TIM3,y1);
                }while(x1>=1305);
                state1=1,j=0;
                break;
            case 1:
                do
                {
                    y1=y1-1;
                    delay_ms(10);

TIM_SetCompare1(TIM3,1307);

TIM_SetCompare2(TIM3,y1);
                }while(y1>=1767);

                state1=2,j=0;
                break;
            case 2:
                do
                {
                    x1=x1+1;
                    delay_ms(10);
TIM_SetCompare1(TIM3,x1);

TIM_SetCompare2(TIM3,1769);
                }while(x1<=1524);
                state1=3,j=0;
                break;
            case 3:
                do
                {
                    y1=y1+1;
                    delay_ms(10);
TIM_SetCompare1(TIM3,1525);
TIM_SetCompare2(TIM3,y1);
                }while(y1<=1872);
                state1=4,j=0;
        }
    }
}

```