

# 目录

<b>1</b>	<b>Latex模板</b>	<b>2</b>
1.1	正文 . . . . .	2
1.2	分点 . . . . .	2
1.3	subsubsection . . . . .	2
1.3.1	平稳性 . . . . .	2
1.3.2	平地行驶 . . . . .	2
<b>2</b>	<b>图片插入</b>	<b>3</b>
2.1	单张图片 - figure . . . . .	3
2.2	多张图片 - subfigure . . . . .	4
<b>3</b>	<b>公式与表格</b>	<b>5</b>
3.1	公式 . . . . .	5
3.2	表格 . . . . .	5
<b>4</b>	<b>示例：电控部分设计</b>	<b>6</b>
4.1	总体概述 . . . . .	6
4.1.1	流程图 . . . . .	6
4.1.2	功能简介 . . . . .	6
4.1.3	电子器件 . . . . .	7
4.1.4	操作系统配置 . . . . .	7
4.2	硬件结构 . . . . .	8
4.2.1	硬件结构图 . . . . .	8
4.2.2	主控板介绍 . . . . .	8
4.2.3	动力输出介绍 . . . . .	9
4.3	控制系统 . . . . .	9
4.3.1	程序框架 . . . . .	9
4.3.2	指令层 . . . . .	9
4.3.3	控制层 . . . . .	10
4.3.4	驱动层 . . . . .	10
4.4	当前问题及改进方向 . . . . .	10
4.4.1	当前问题 . . . . .	10
4.4.2	改进方向 . . . . .	11

# 1 Latex模板

## 1.1 正文

这款移动云台机器人主要针对的是工作在复杂路况环境中的科研人员和相关从业人员，例如抢险救灾的消防员、维修管道的维修工人、地质勘探的科研工作者等等，从事着较为危险的工作，在实际开展工作中面对着难以预料的环境，受限于现有的勘探机器人缺乏强大的越障能力，经常会出现工作失灵的情况，于是工作人员亲自去勘探也就最为常见。

空行

加粗与换行

## 1.2 分点

- 没有出色的代替人力的机器人完成相关的工作。咬牙进入危险环境的从业人员极为常见。
- 现有的移动平台的驱动方式往往只有一种，单一的足式很难保证适应各类环境，亟待整合多种足式运动优势的机器人出现。

## 1.3 subsection

基于此，我们需要设计的一款移动云台机器人应当具有如下的特点：

### 1.3.1 平稳性

移动的云台作为勘探仪器的载体，必须要保证移动时的平稳性，避免因剧烈震动导致精密仪器的损坏，否则从业人员也不放心安装精密仪器在机器人上，考虑到实际需求才能争取这些特征人群的认可

### 1.3.2 平地行驶

不仅要能够实现越障功能，移动平台机器人在较为平坦的路面上，具有出色的行走能力同样重要。速度太慢而被抛弃的机器人并不在少数。

## 2 图片插入

### 2.1 单张图片 - figure



图 1. 主体部分-实物搭建图

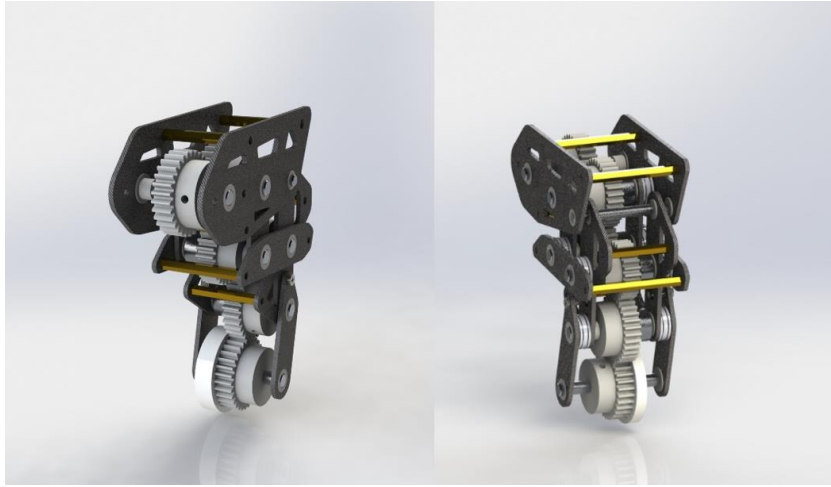
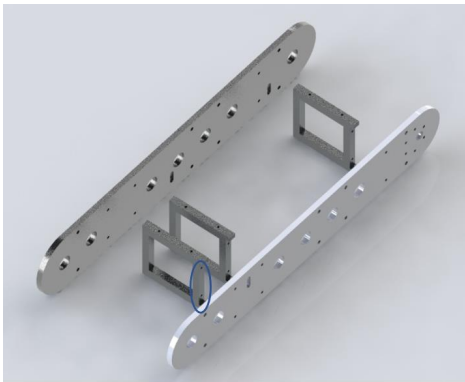


图 2. 齿轮系驱动前足-模型渲染图

## 2.2 多张图片 - subfigure



(a) 框架爆炸图



(b) 框架整合图

图 3. C形足传动框架

## 3 公式与表格

### 3.1 公式

推荐软件：Mathpix Snipping Tool, 截图生成代码。

$$K\varepsilon = \frac{\Delta R_1}{R_1} - \frac{\Delta R_2}{R_2} + \frac{\Delta R_4}{R_4} - \frac{\Delta R_3}{R_3} = K\varepsilon_1 - K\varepsilon_2 + K\varepsilon_4 - K\varepsilon_3$$

式中：

k - 仪器灵敏系数指示值

$\varepsilon$  - 仪器读数 $\mu\varepsilon$

K - 电阻应变片的灵敏系数

### 3.2 表格

不建议使使用，过于复杂。

解决方案：1)excel截图；2)自动生成表格代码（在线or编译器自带）。

Table 1: R1R7组成半桥

载荷N	初始值	5	10	15
仪器读数 $\varepsilon_{\text{仪}}$	0.81	28.36	54.99	81.33

Table 2: R1R2R3R4组成全桥

载荷N		初始值	5	10	15
仪器读数 $\varepsilon_{\text{仪}}$	临界	-0.30	0.77	2.20	4.01
	对接	-4.64	99.34	203.01	306.52

## 4 示例：电控部分设计

### 4.1 总体概述

#### 4.1.1 流程图

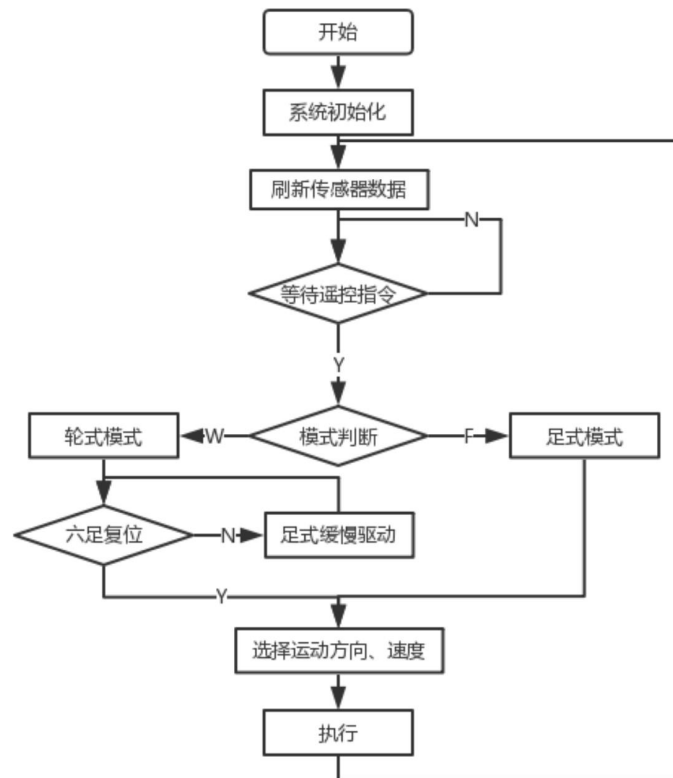


图 4. 控制流程图

#### 4.1.2 功能简介

- 足式驱动

初始化：以两侧轮足平行的上电位置为基准，驱动其中之一侧，使之相差 $180^\circ$ ；

运行：同向同速驱动两侧轮足前进，反向同速驱动两侧旋转。

- 轮式驱动

初始化：驱动两侧轮足，使两侧前轮同时处于最低点；  
运行：同向同速转动前轮前进，差速驱动转弯。

- 遥控部分

右上拨杆最上为轮式模式，此时左上拨杆控制高速、低速和停止，右侧摇杆上下推动控制前进后退，左右推动进行转弯；最下为足式模式，此时右侧摇杆上下推动控制前进后退，仅在无前后运动速度的情况下左右推动进行原地旋转控制；中间为停止状态，对任意输入不响应。

#### 4.1.3 电子器件

电子器件	功能
单片机 STM32F427IIH6	控制核心
遥控接收系统 DT7&DR16 2.4GHz	遥控接收
直流无刷电机 M3508	足式驱动
减速电机 37GB520R-12V-300	轮式驱动
C620 电子调速器	直流无刷电机驱动模块
L298n	直流减速电机驱动模块

图 5. 电子器件清单

#### 4.1.4 操作系统配置

RTOS 操作系统任务	优先级
遥控器控制任务 RControlTask	osPriorityHigh
CAN1 电机控制任务 Can1_Task	osPriorityAboveNormal
2ms 定时任务 Timer_Task	osPriorityAboveNormal
红绿灯闪烁任务 ledGreenTask, ledRedTask	osPriorityNormal

图 6. 操作系统任务及其优先级

## 4.2 硬件结构

### 4.2.1 硬件结构图

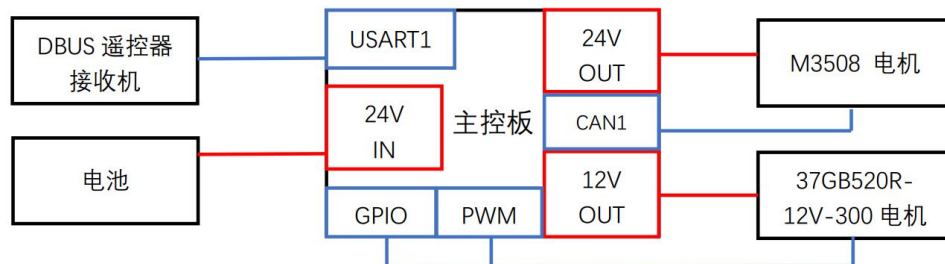


图 7. 硬件结构框图

### 4.2.2 主控板介绍

该开发板主控芯片为STM32F427IIH6芯片，通过24v直流电源供电，板载可控电源输出，有丰富的扩展接口和通信接口，可以满足控制需求。在本项目中，用到了看门狗、定时中断，串口通信、CAN通信，GPIO、PWM接口。

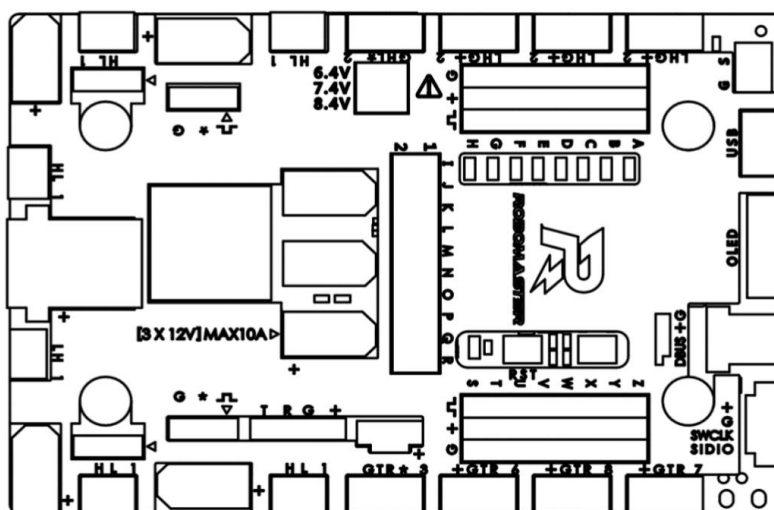


图 8. 主控开发板



### 4.2.3 动力输出介绍

M3508无刷直流电机工作电压为24v，最高功率可达220w，最大扭矩5N\*m，有感FOC控制在任何转速下都能提供稳定的扭矩，让机器人在快速响应的同时保持平稳的动力。带有编码器，配合C620电调使用，可以得到角度、速度、转矩、温度等参数，通过CAN总线控制，高效方便。该电机可自动感应高温、断线等异常并报警，同时反馈故障，使用安全。37GB520R-12V-300直流减速电机工作电压为12v，额定力矩2.25kg\*cm，通过PWM控制，可实现一定范围内的调速，但占空比直接影响输出电压，过低会使电机无法工作。

## 4.3 控制系统

### 4.3.1 程序框架

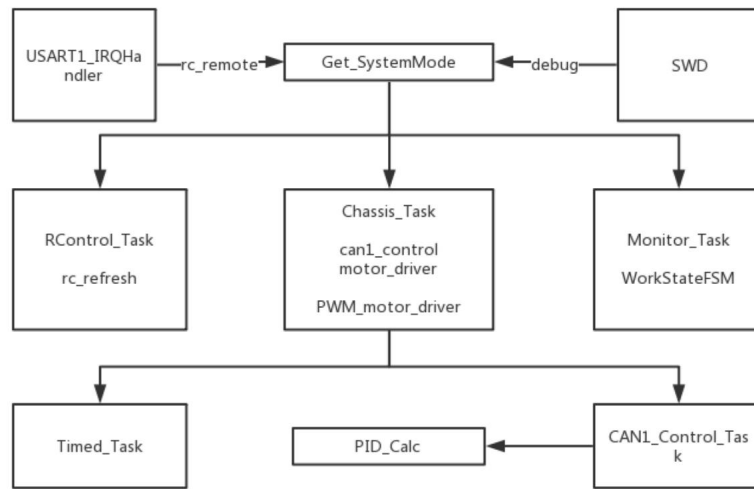


图 9. 程序框图

### 4.3.2 指令层

指令层为程序的顶层，该层又分为两部分：遥控器数据读取、整车运行状态机。以遥控器串口中断作为外部输入，是人机交互的接口部分，为了实现人对该六足机器人的控制，该中断为第一优先级，在任何时刻第一时间响应人的指令。

整车运行状态监测在定时器中断中进行，包括控制模式切换（足式运行模式、轮式运行模式、以及预留的自动模式）、运行状况状态机（开机启动初始化、电机堵转等应急反应、干扰或潜在bug的异常抛出），使整个控制条理清楚，为进一步开发预留了合理的空间。另外，该程序添加了IWG(看门狗复位)，在意外情况下紧急重启，不会发生失控状况。

### 4.3.3 控制层

中间层为控制层，包括足式运行控制、轮式运行控制、红绿状态指示灯三个部分，控制层接收指令层的命令和处理驱动层反馈的数据。

对足式驱动电机采用经典的PID控制，主控板与电子调速器之间通过CAN总线通信，主控读取编码器值并解算得到角度、速度，以目标值与实际值之差为输入量，通过双闭环PID算法得到电流作为输出量即控制量，发出电流指令。对轮式驱动电机采用IO控制转向及锁死、PWM控制速度，设置高速、中速、低速三个档位。红绿状态指示灯按照状态机状态出现不同的闪烁情况。

### 4.3.4 驱动层

驱动层为程序底层，包括串口、定时器、CAN总线、GPIO等外设的初始化与驱动，实现与电子调速器之间的数据收发，串行通讯设备之间的数据收发，PWM波输出等底层驱动。在驱动层和控制层之间学习了骆学长的IOPool中间层进行数据交换，读写完全分离，保证了不会出现误修改问题。

## 4.4 当前问题及改进方向

### 4.4.1 当前问题

单侧轮足的耦合运动设计，使得该六足机器人在足式模式下不能实现差速转弯，否则将打乱两侧轮足相差 $180^\circ$ 的结构，使平台颠簸，损伤机器人。仅能采用原地旋转的方式进行原地转弯。

轮式驱动在前足，故如果同速反向驱动电机，机器人将会绕两前足中心旋转，而不是绕整个平台的中心旋转；由于后轮是单向旋转的普通从动轮，后轮在转动过程中会受到平行于轴的侧向力，引起较大的摩擦。

由于编码器是针对转子而言，且减速比为19:1，故如果运行过程中出现异常并重启，机器人将无法找到自己的当前位置，对两侧轮足的相对角度控制造成困难。

#### 4.4.2 改进方向

针对以上第二个问题，可以根据尺寸和速度进行运动学解算，在旋转时给两轮进行速度补偿，使之绕平台中心旋转。

针对以上第三个问题，可以寻找合适的传感器，并找到合适的安装位置，来实现绝对位置的定位。比如磁电接触开关、触碰开关等。

说明，该工程为配套改装足式驱动电机为M3508后的程序，源码为文件“六足-stm32”，改自RoboMaster开源代码。配套展示用42GA775F-12V-110直流电机的Arduino源码见文件“六足-Arduino”。

附,关键代码:

```
void complete_PID_Calc(complete_PID_Regulator_t *pid){
    pid->errorCurr = pid->target - pid->feedback;
    pid->error[pid->addr_error] = pid->errorCurr;
    pid->errorSum += pid->errorCurr;
    pid->addr_error = (pid->addr_error + 1)%80;
    pid->errorSum -= pid->error[pid->addr_error];
    pid->componentKp = pid->kp * pid->errorCurr;
    MINMAX(pid->componentKp, -pid->componentKpMax, pid->componentKpMax);
    pid->componentKi = pid->ki * pid->errorSum;
    MINMAX(pid->componentKi, -pid->componentKiMax, pid->componentKiMax);
    pid->componentKd = pid->kd * (pid->errorCurr - pid->errorLast);
    MINMAX(pid->componentKd, -pid->componentKdMax, pid->componentKdMax);

    pid->errorLast = pid->errorCurr;

    pid->output = pid->componentKp + pid->componentKi + pid->componentKd;
    MINMAX(pid->output, -pid->outputMax, pid->outputMax);
}
```

图 10. 双闭环PID角度、速度控制

```
if(s_CMFRCount == 1)
{
    IOPool_getNextRead(CMFRxIOPool, 0);
    CMFRThisAngle = IOPool_pGetReadData(CMFRxIOPool, 0)->angle;

    if(isCMFRFirstEnter==1) (CMFRLastAngle = CMFRThisAngle;isCMFRFirstEnter = 0;return;) //初始化时，记录下当前编码器的值

    if(CMFRThisAngle<=CMFRLastAngle)
    {
        if((CMFRLastAngle-CMFRThisAngle)>3000)//编码器上溢
            CMFRRealAngle = CMFRRealAngle + (CMFRThisAngle+8192-CMFRLastAngle) * 360 / 8192.0 / CMReduction;
        else//反转
            CMFRRealAngle = CMFRRealAngle - (CMFRLastAngle - CMFRThisAngle) * 360 / 8192.0 / CMReduction;
    }
    else
    {
        if((CMFRThisAngle-CMFRLastAngle)>3000)//编码器下溢
            CMFRRealAngle = CMFRRealAngle - (CMFRLastAngle+8192-CMFRThisAngle) * 360 / 8192.0 / CMReduction;
        else//正转
            CMFRRealAngle = CMFRRealAngle + (CMFRThisAngle - CMFRLastAngle) * 360 / 8192.0 / CMReduction;
    }
}
```

图 11. 编码器溢出算法