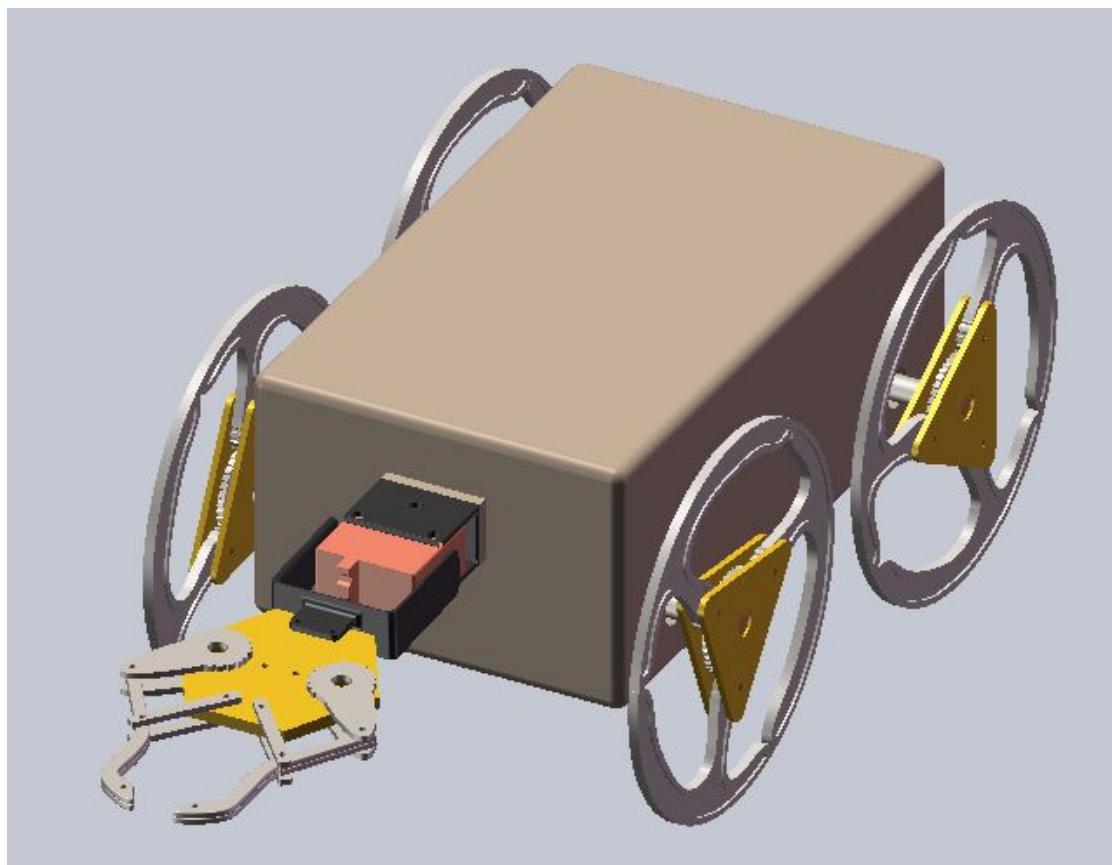


中国科学技术大学2012robogame计划书

之

晴天霹雳队



中国科学技术大学
第十二届RoboGame活动周机器人大赛报名表

院系	信息学院	专业	自动化系		
参赛内容	<input checked="" type="checkbox"/> 机器人献花				
队伍名称	晴天霹雳	指导老师信息	姓 名	邵长星	老师签字
			手机号码	13866184114	
			电子邮箱	starshao@ustc.edu.cn	
参赛队信息					
职务	姓名	学号	手机	学校邮箱	BBS注册ID
队长					
队员					

参赛承诺：

我们保证以本承诺为有约束力的协议，遵守中国科学技术大学第十二届RoboGame组委会的有关规定，认真进行机器人的设计制作等工作，就比赛相关问题积极与组委会交流，服从组委会的活动安排与最终裁判。

对于由本队引发的一切不良后果由本队承担相应责任。

所有参赛队员（签字）：

- 备注：1、请各参赛队填写报名表及计划书，电子版发送至 2012robogame@gmail.com，
 打印纸质版一份交送力学三楼313室张尚斌处（进门右边第一个桌子），截止日期5月20日。
 2、每队参赛人员人数在5名以内。
 3、请各系推荐一名参赛队队长担任该系参赛负责人，其主要职责如下：
 (1) 填写《RoboGame2012各系参赛信息汇总表》并上交（方式同备注1）；
 (2) 负责本系机器人制作经费的领取；
 (3) 负责相关比赛信息的通知，与RoboGame2012组委会的联络工作。
 4、本次活动最终解释权归中国科学技术大学 RoboGame 2012组委会所有。

中国科学技术大学 RoboGame 2012机器人大赛组委会
 2012年4月21日

中国科学技术大学第十二届RoboGame机器人大赛

参赛计划书

队伍名称 : 晴天霹雳

系别 (全称+代号) : 10 自动化系

类别 :

机器人献花

说明

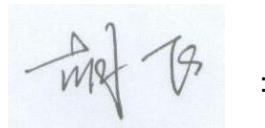
1. 参赛队应在认真阅读此说明各项内容后按要求详细填写。
2. 参赛队在填写本表时只需根据所报项目填写A1 (机器人算24) 或A2 (机器人献花) 表。

承 诺 书

组委会承诺 :

我们组委会保证及时解决各参赛队就比赛相关问题提出的疑问，为各参赛队的制作计划等保密，公正处理机器人比赛相关事务，选拔优秀成员担任比赛裁判，保证裁判的公正。

组委会负责人 (签字)



2012年4月21日

参赛者承诺 :

我们队承诺对本人填写的各项内容保证是本队的原创，没有抄袭他人。

我们保证以本承诺为有约束力的协议，遵守中国科学技术大学第十二届robogame组委会的有关规定，认真进行机器人的设计制作等工作，就比赛相关问题积极与组委会交流，服从组委会的活动安排与最终裁判。

对于由本队引发的一切不良后果由本队承担相应责任。

参赛队员 (签字) :

年 月 日

目 录

官方表格

可行性分析

机械部分

A. 可伸缩叶轮

B. 行星轮

电路部分

程序部分

A2. 机器人献花

队员组成及分工	姓名	学号	分工

机器人的基本设计思路 (想法) 及 创新性

爬楼设计

方案一：采用伸缩式叶轮，平地行进时采用圆形形态，实现平稳行进，爬楼梯时圆形轮子伸缩成叶轮形态（具体看机械部分）爬行楼梯。

优点：可以适合不同类型楼梯。机器人外形简洁。

缺点：机械结构加工稍显复杂

方案二：采用行星轮设计，平地行进时两轮着地，采用齿轮传动，驱动小轮转动。爬楼时整个行星轮绕中心轴转动，以实现爬楼（具体见机械部分）。

优点：有成品，价格便宜，可以在其基础上进行适当改造。

缺点：不能很好适应各种楼梯。

动力设计：

叶轮：平地时两侧转速相同实现平行运动，转弯时通过左右轮子转速调整，使两侧转速不一致实现转弯，爬楼时，：首先驱动叶轮内齿轮，更改叶轮形态，然后驱动前进。

行星轮：平地驱动时通过驱动小轮实现平地前进，爬楼时驱动整个行星轮转动。

寻路设计：

此步骤为传感器的选择和使用，在此一并说明。

由于组委会规定可以使用3个以内自己的信号源，这个相对比较稳定，而且比起图像识别的庞大的数据量有着难以比拟的优势，故必须充分利用之。在鲜花处和转弯处放置超声波、红外等信号源，指引机器人寻找鲜花和行进路线。

上楼时，采用碰撞传感器识别楼梯，当机器人前端两个平行的碰撞传感器同时受到碰撞信号时，机器人已经垂直对准楼梯，可以上楼。然后机器人更改为上楼行进形态。下楼时，因为可以事先摆正机器人位置，故可以简化设计。但是怎样识别垂直下楼也是一个值得考虑的问题。

3.不可避免的还是要用到图像识别技术。由于图像识别数据量庞大，单片机显得无能为力，因此拟用无线模块将数据传至PC上处理。这有一个问题，无线模块一般波特率能达到9600bps，但是在淘宝上发现成品无线模块一般只有2mbps，速率可能达不到要求。也可以把pc放在机器人上，通过图像卡（其通讯速率可以达到400mbps），把程序转移到pc控制台进行。

同时，识别时，路标固然要用，但是单靠图像识别可以识别其方位，但很难识别其距离。经过实地考察，觉得周围的环境可以加以利用而达到更好的控制效果，如：路线周围的草地，路线上铺的瓷砖中红色的部分。

取/持/献花设计：

由于组委会没有要求拿花和取花的动作，故可以简单的设计成一个自由度的机械臂。但是考虑到观赏性，我们设计了两个自由度的机械臂，结构参看机械部分。

取花：机器人行进到放花地点，机械手臂下移至放花处，前方夹取处电机时机械钳收缩夹取献花。手臂上移。

持花：机械手臂保持一定高度，机械钳保持收缩状态。使鲜花

保持正立。

献花：机器人行进到严济慈像处，手臂下移，机械钳舒张将鲜花置于地上。

机器人的可行性和智能性描述

1. 可行性见后文可行性分析
2. 智能性描述：
 1. 对路标，信号源，周围环境的传感，采用闭环负反馈控制，利用PID或鲁棒控制，使其对扰动有较强的抵御能力。
 2. 程序上的稳定性：如遇到摄像头拍不到正确图像时，应能自主旋转，以找寻正确方位，拍摄图像。
 3. 当遇到对方小车时应能主动回避。
等等诸如此类的应变策略。

机器人的具体技术实现
——机械部分
(可附图)

见后文机械部分。

机器人的具体技术实现
——电路部分
(可附图)

见后文电路部分。

机器人的具
体技术实现
——控制部
分
(可附图)

见后文程序部分。

制作时间进度安排	时间	进度
	6月15日——7月10日	听关于机器人的讲座，动手搭建硬件平台
	7月11日——8月5日	硬件平台基本完成，装上单片机进行软件调试
	8月6日——8月30日	优化硬件及算法，对程序中待定的参数进行整定和优化
	9月1日——9月15日	最后的调试和实地测试，与兄弟队伍切磋，达到最佳竞技状态

项目预算	事由	单价（元）	数量	合计（元）
	Freescale xs128	25	6	150
	最小系统板（PCB）	80	2	160
	铝板等材料	300	1	300
	加工费	150	2	300
	摄像头	120	2	240
	蓝牙无线模块	60	1	60
	舵机	40	6	240

电机	80	4	320
超声波传感器	2	10	20
成品行星轮	8	5	40
成品2自由度 机械臂	190	1	190
万用版	5	10	50
大功率电机	150	1	150
大功率电池	80	1	80
视频采集卡	150	1	150
其他 (芯片 等)	200	1	250
备用	300	1	300
总计 (元)			2850

其它说明	无。
------	----

2. 可行性分析

机器人在爬楼梯过程中，机器人除了轮子其他部件不应与楼梯相接处，机器人几何参数与楼梯几何参数必须满足一定的条件，这样机器人才能走上楼梯或穿越障碍。图 4 为六轮移动机器人在爬越楼梯，它的 4 个走行轮和 2 个爬行轮都为驱动轮，在平面上运行时，机器人摆起两个摆臂，以普通的 4 轮方式行驶，爬楼梯时，将通过两个摆臂的摆动，带动两个爬行轮，协调自身的姿态以适应楼梯地形，不会像普通汽车那样被顶起而失效。下面将对小车在穿越不同台阶的位置进行分析，如图 4 所示，为六轮移动机器人在一级台阶上。

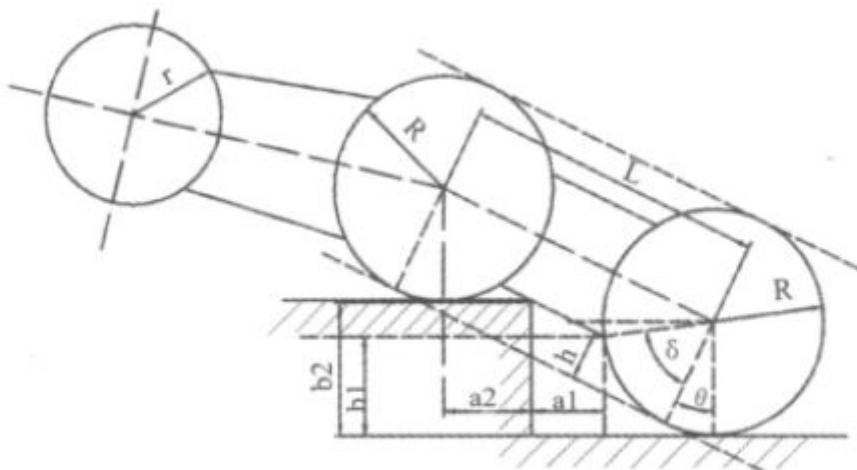


图 4 六轮移动机器人在台阶上

在爬上楼梯的初始阶段和在楼梯的中段，爬行轮总是可以找到台阶支撑，所以机器人可以摆动摆臂，增大车身的倾斜角度，使机器人能够攀越比较陡峭的楼梯，但是当机器人爬到楼梯的上部时，摆臂已经出了楼梯区域，爬行轮再也找不到可支撑的台阶，车身的倾斜角度受到了限制，当然也就限制了机器人爬楼梯的能力。从整个过程考虑，研究机器人的爬楼梯的过程，只分析机器人爬出楼梯的情形(4 轮状态)就可以了，由图 4 可知：

$$\cos \delta = (R-h)/R \quad (1)$$

$$\cos(\delta + \theta) = (R-b_1)/R \quad (2)$$

$$b_2 = L \sin \theta - R + R = L \sin \theta \quad (3)$$

$$b_1 = R - R \cos(\delta + \theta) = R - R \cos \delta \cos \theta + R \sin \delta \sin \theta \quad (4)$$

由 $b_1=b_2$ 得：

$$b^* = L \sin(\arctan \Phi) \quad (5)$$

其中，

$$\Phi = \frac{-R \cos \delta (L - R \sin \delta) + R \sqrt{L^2 - 2RL \sin \delta}}{(L - R \sin \delta)^2 - R^2}$$

b^* 为轮触地条件下的最大台阶高度， L 为前后走行轮中心距， R 为走行轮半径， r 为爬行轮半径， h 为车底净空， b_2 为台阶高度，如果 b_2 和 b_1 相等，则台阶将不会和车体的底部相接触，这个轮触地条件是对台阶而言的。

下面对凸起进行分析，当凸起高度大于车底净空 h 而小于上面的轮触地最大台阶高度，同时凸起的纵深长度 l_c 小于轮触地最小凸起宽度 l_e 时，四轮载体可能被顶起，由图 5 计算得：

$$H = L \sin \theta + R$$

$$m = H - n$$

$$\cos \alpha = m/R$$

$$l_e = L \cos \theta - R \sin(\delta + \theta) - R \sin \alpha \quad (6)$$

其中 n 为凸起的高度，把它转换壕沟宽度为 $l_d=2$

$\sqrt{R^2 + (R - b^*)^2}$ ，如图 6 所示为宽度小于 l_d 的壕沟，六轮移动机器人是可以跨越的，而对于宽度大于轮直径的壕沟，如果壕沟深度小于 b^* ，则根据上面的轮触地条件，六轮移动机器人也是可以跨越的。

图 7 为多级台阶的情形，其中： a_1 为第一级台阶纵深长度， b_1 为第一级台阶的高度， b_2 为第二级台阶的高度。分析可知轮触地

条件为 $b_1 \leq b^{**}$, $b_2 \leq b^*$, $0 \leq a_1 \leq h^*$ 。由图可知：

$$\cos \theta = (b_1 + b_2) / L$$

$$b^{**} = R - R \cos(\delta + \theta) \quad (7)$$

$$h^* = R \cot \theta - (R - h) \csc \theta \quad (8)$$

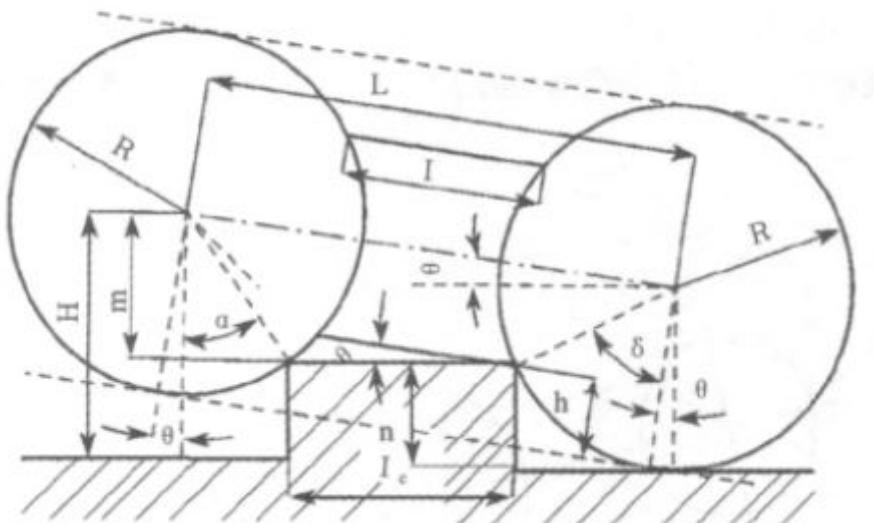


图 5 机器人在凸台上

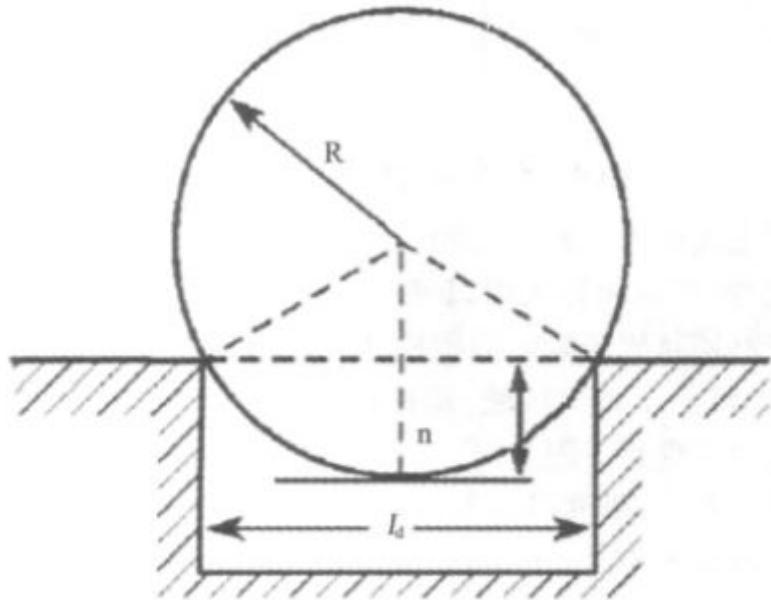


图 6 壕沟宽度和台阶高度的转换

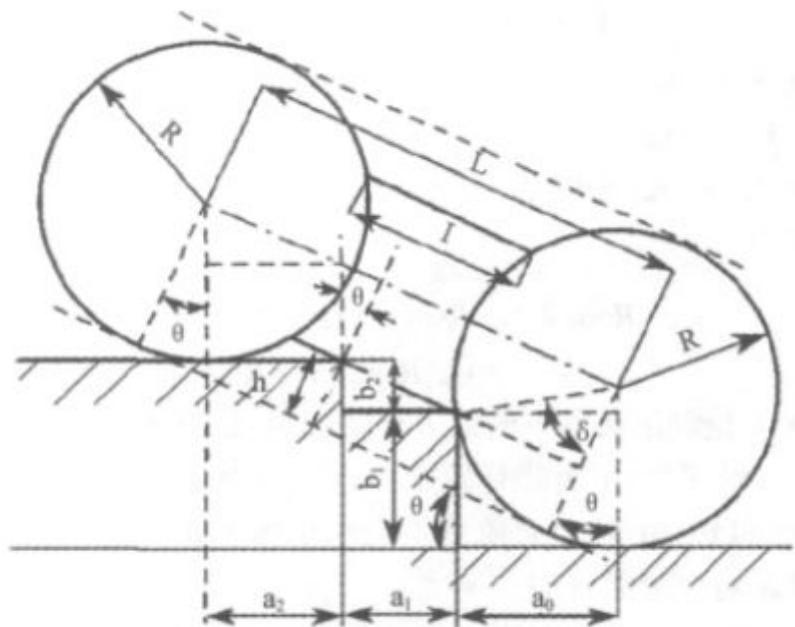


图 7 多级台阶(楼梯)

由以上分析可知,六轮移动机器人可穿越障碍的地形为:

- (1) 对于台阶,高度要小于 b^* 。
- (2) 当凸起高度小于车底净空 h ,台阶要小于 b^* ;大于车底净空,纵深长度要大于 l_e 。
- (3) 对于多级台阶,第一级台阶纵深长度要小于 h^* ,第一级和最后一级台阶高度分别小于 b^{**} 和 b^* 。(5)、(6)、(7) 和 (8) 式分别表示 b^* 、 l_e 、 b^{**} 和 h^* 。

2.2 机器人重心对其倾覆的影响

当机器人在爬越楼梯时，我们应该考虑到其重心对机器人的影响，如果重心的位置不合理，将会导致机器人倾覆，下面对机器人重心位置进行分析。如图 8 所示， θ 为机器人俯仰角， h 为重心距

轮心连线的垂直高度， f 为重心偏离轮心连线垂直中心面的距离， G 为机器人的重心，其不发生倾覆的条件为下式：

$$h \sin \theta + f \cos \theta < \frac{L}{2} \cos \theta$$

可得：

$$\tan \theta < \frac{L - 2f}{2h}$$

由上式可知，当 h 和 f 减小与 L 增大时，机器人俯仰角 θ 的最大值将增大，机器人将不容易倾覆，所以在设计机器人时，应考虑到上面 3 个值对其倾覆的影响。在生产设计时，要合理安排机器人各个部件位置，合理地设计两轮之间的距离和机器人的高度，增大机器人的爬行能力。

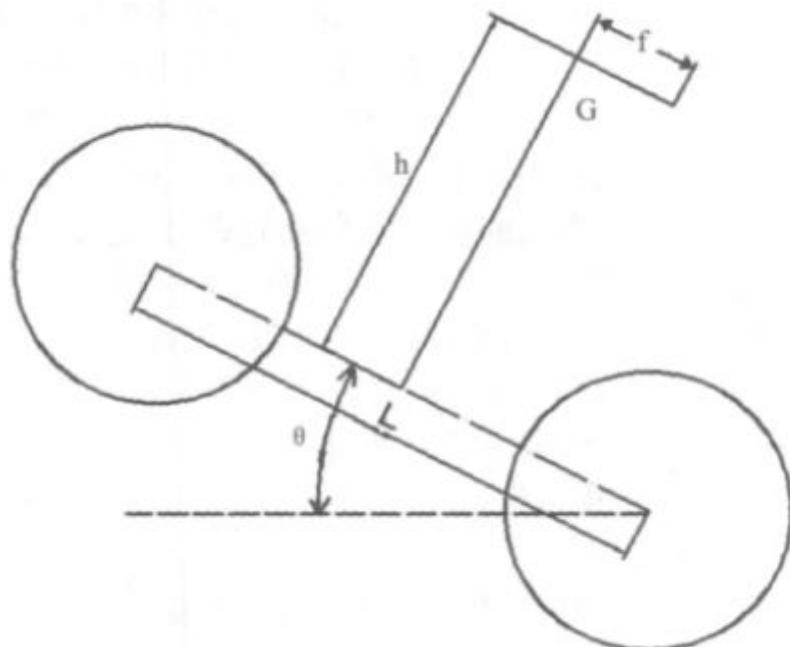


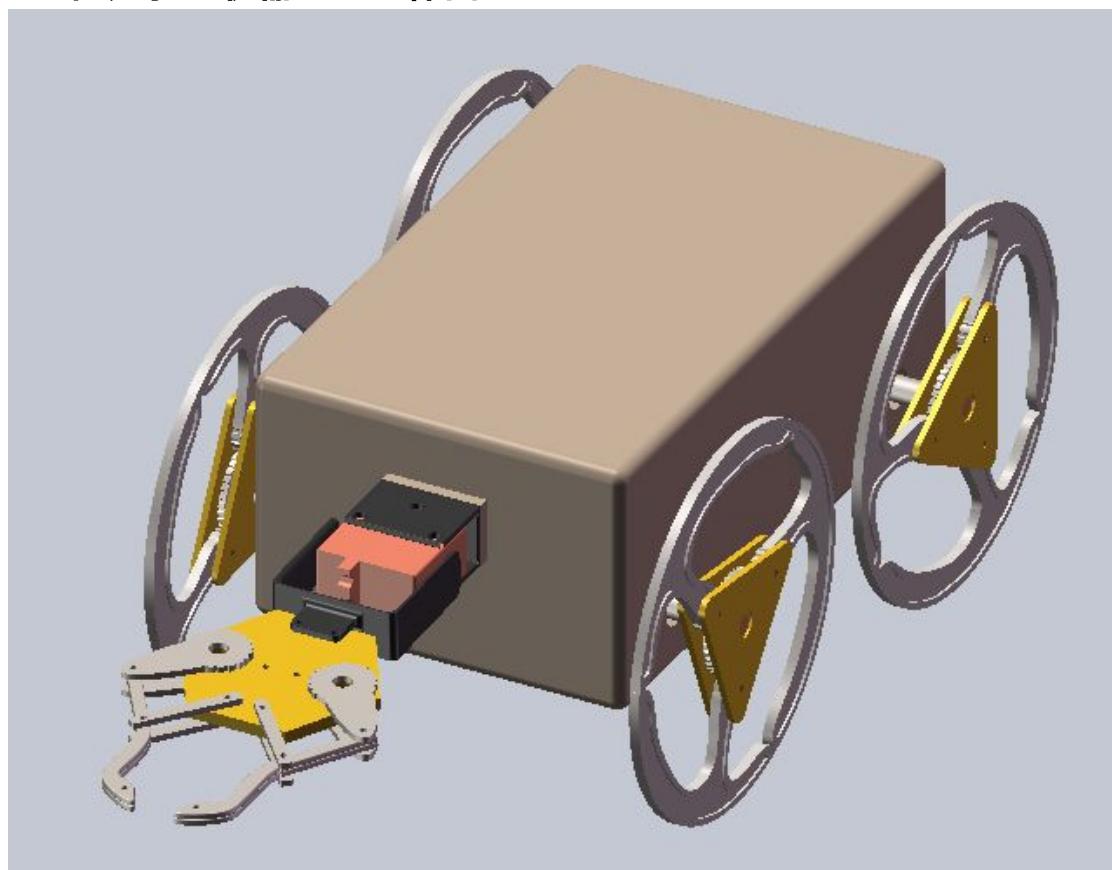
图 8 重心对机器人倾覆的影响

引用自：
《可爬楼梯的移动机器人研究》马学玉

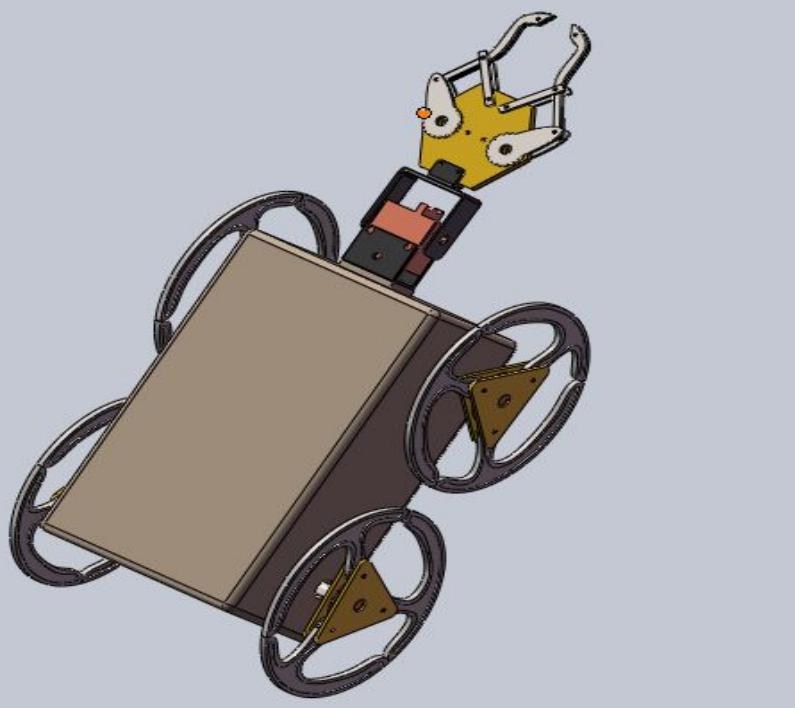
机械部分：

A.方案一：可伸缩叶轮式

1.下面是我们机器人的总体图



(机器人整体效果图)

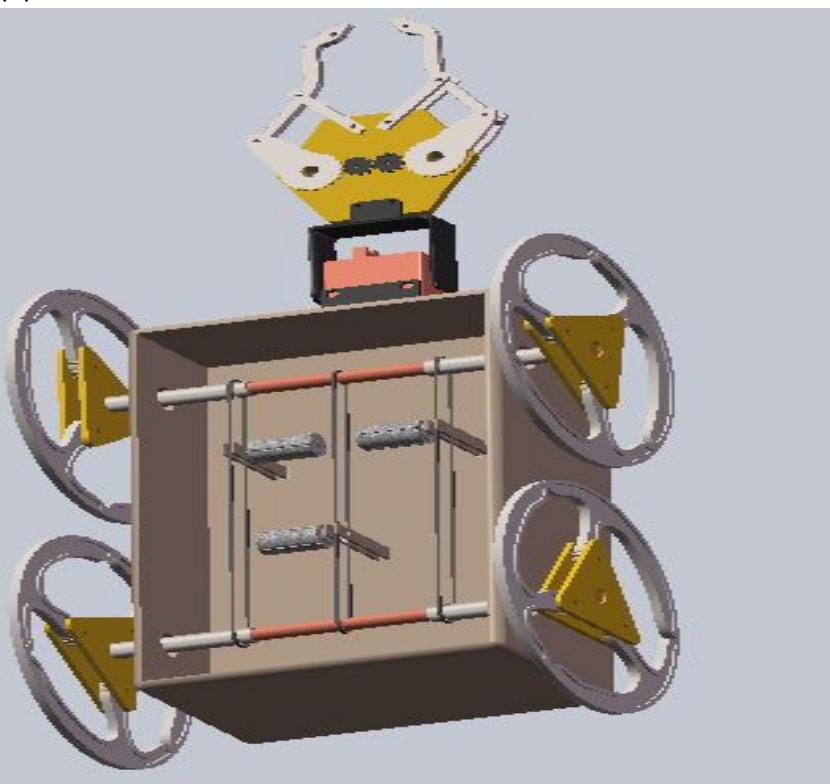


(机器人正面模型)

其中轮子采用可伸缩叶轮，具体将在后面介绍其用法。

机械臂采用两自由度机械臂，具体将在后面介绍。

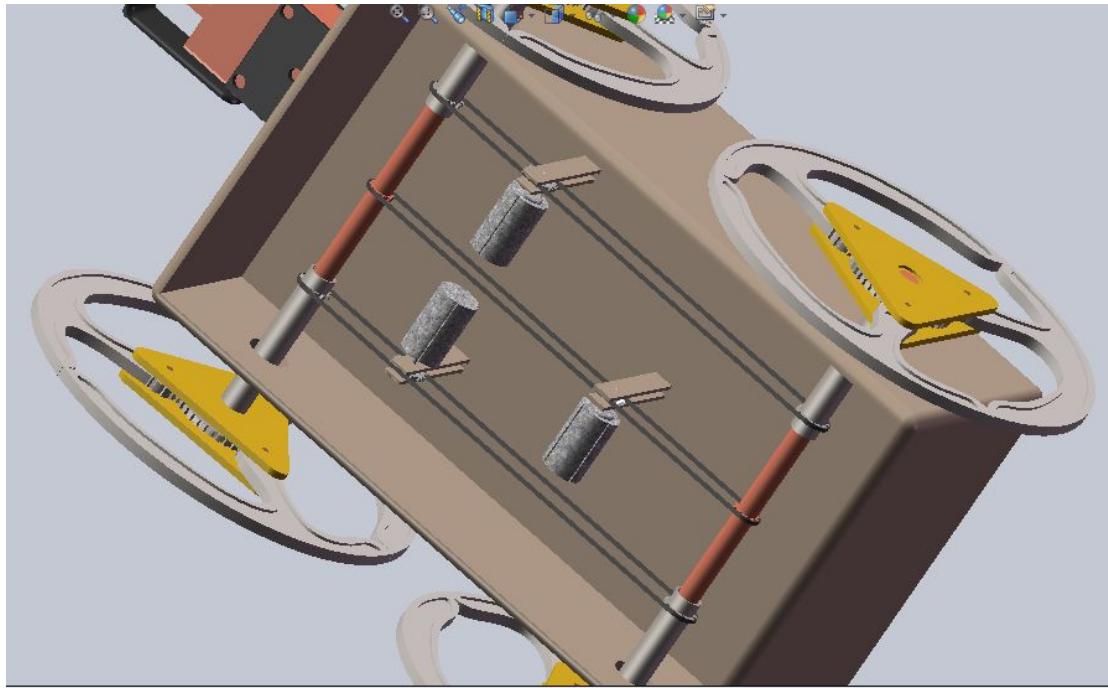
小车背面如下图：



(机器人背面模型)

我们的设计一大特点是使用的驱动相对较少，容易控制。只用了3个电机就可以实现小车的平动和爬楼。

下图可以看得更加清楚：

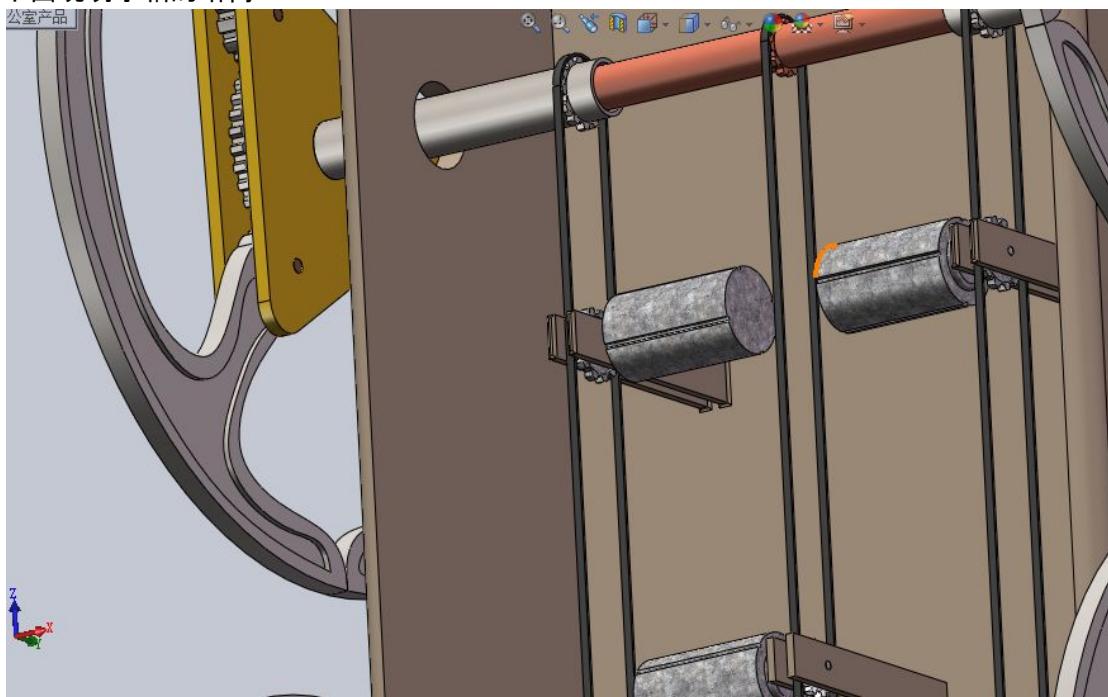


(机器人电机控制概略)

中间的电机同时控制4个可伸缩性叶轮的变形。关于变形详见后文轮子的介绍。

两边的两个电机分别控制两侧的轮子转动。

下图说明了轴的结构：

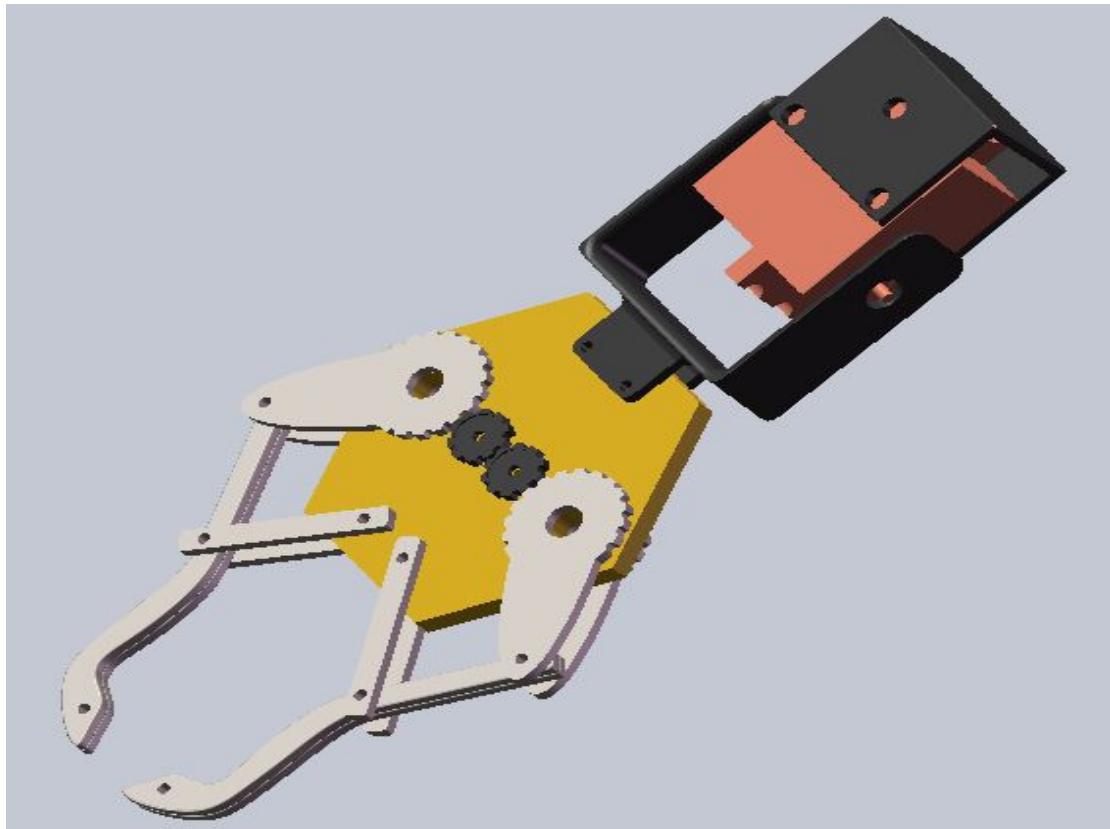


(轴的结构)

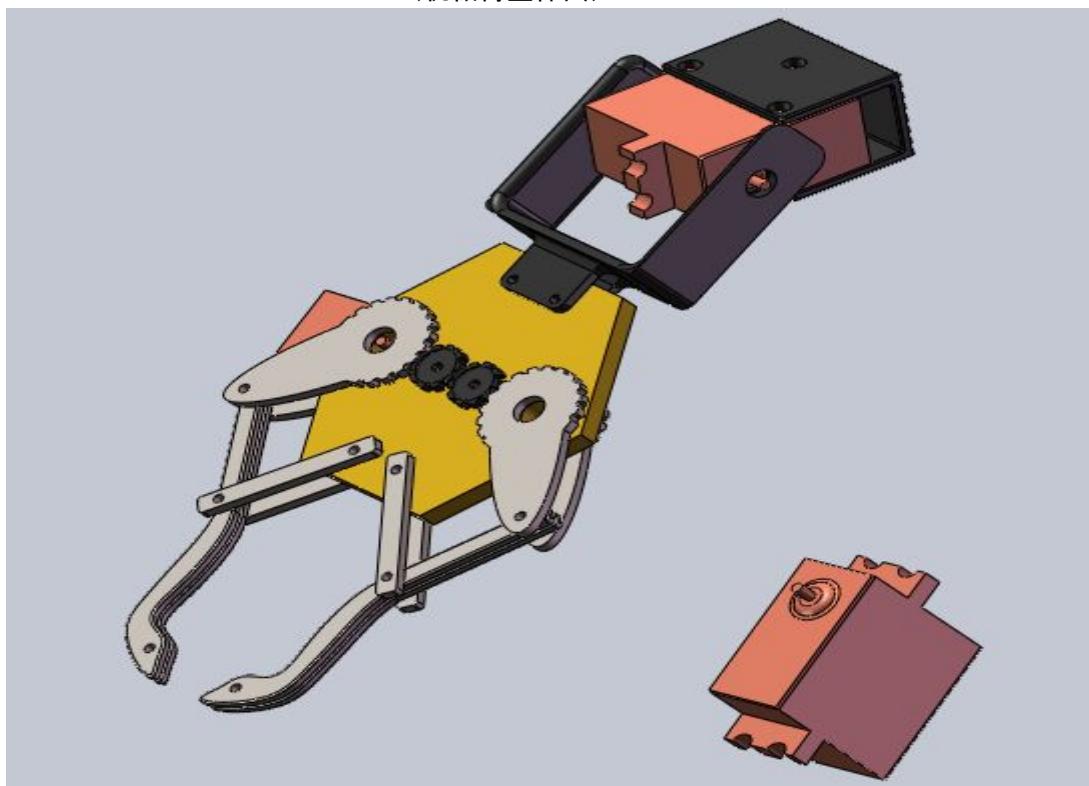
考虑到电机实际可能功率不够，可以适当增加电机的数量。但我们还是倾向于使用比较少的电机，一方面容易控制，另一方面保证了整体的协调性。

2.下面详细介绍机械臂的构造：

机械臂整体图如下：



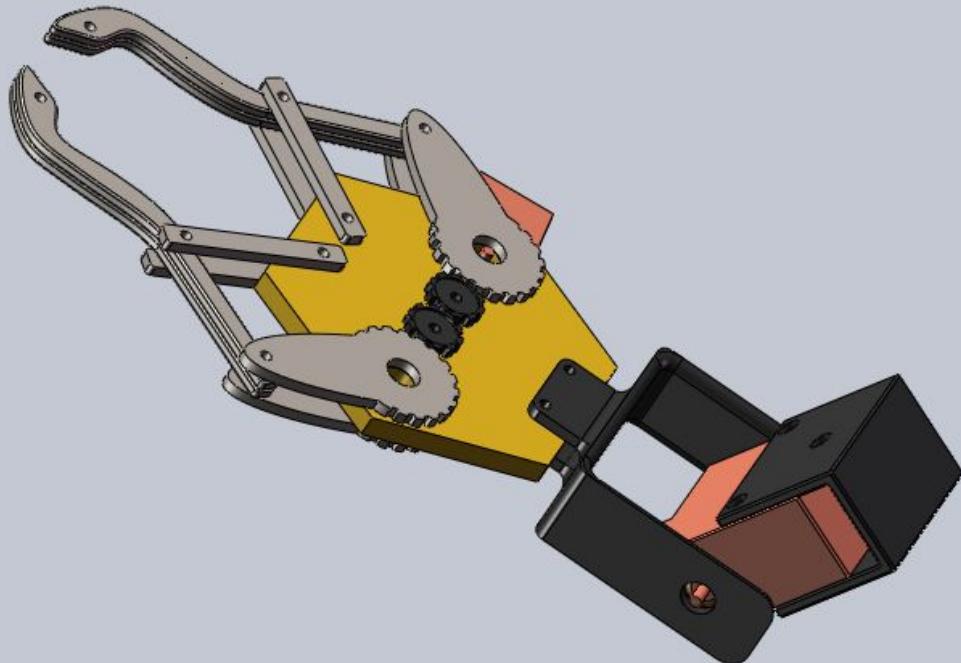
(机械臂整体图)



(机械臂整体模型)

如图，两个舵机，一个放在后臂，控制机械臂上下转动。一个放在前臂，控制前面齿轮系统运动，从而控制夹紧与放松。右下角为电机模型。

后臂的转动如下图：

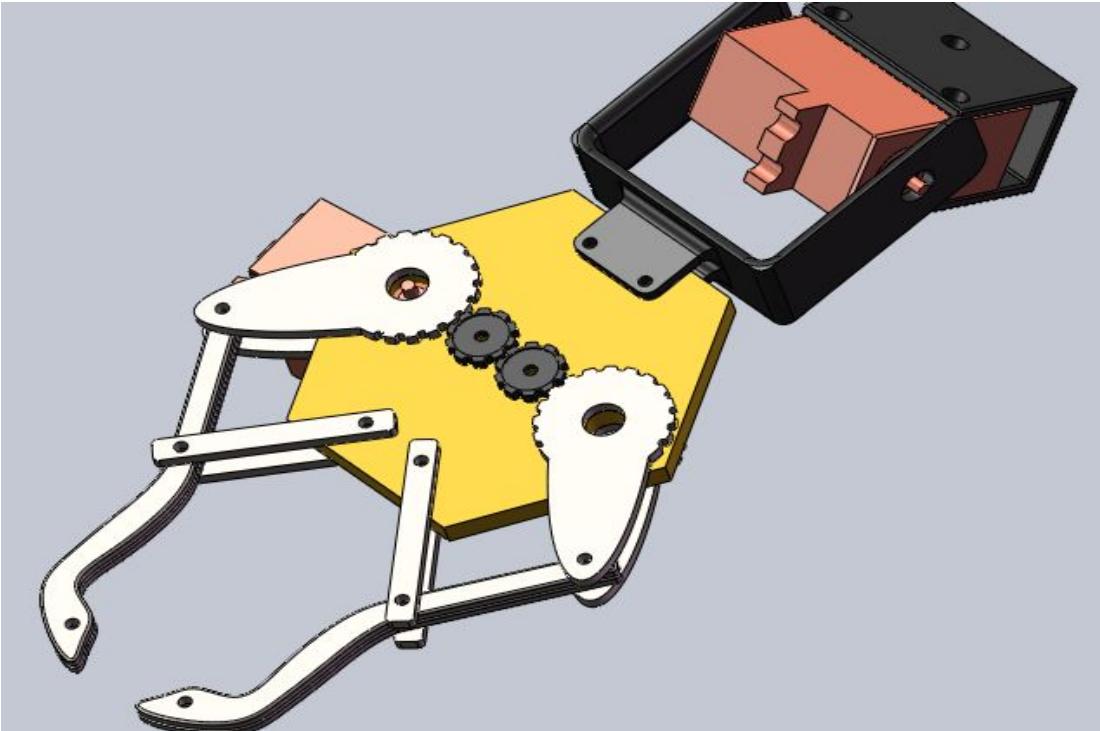


(后臂自由度演示1)

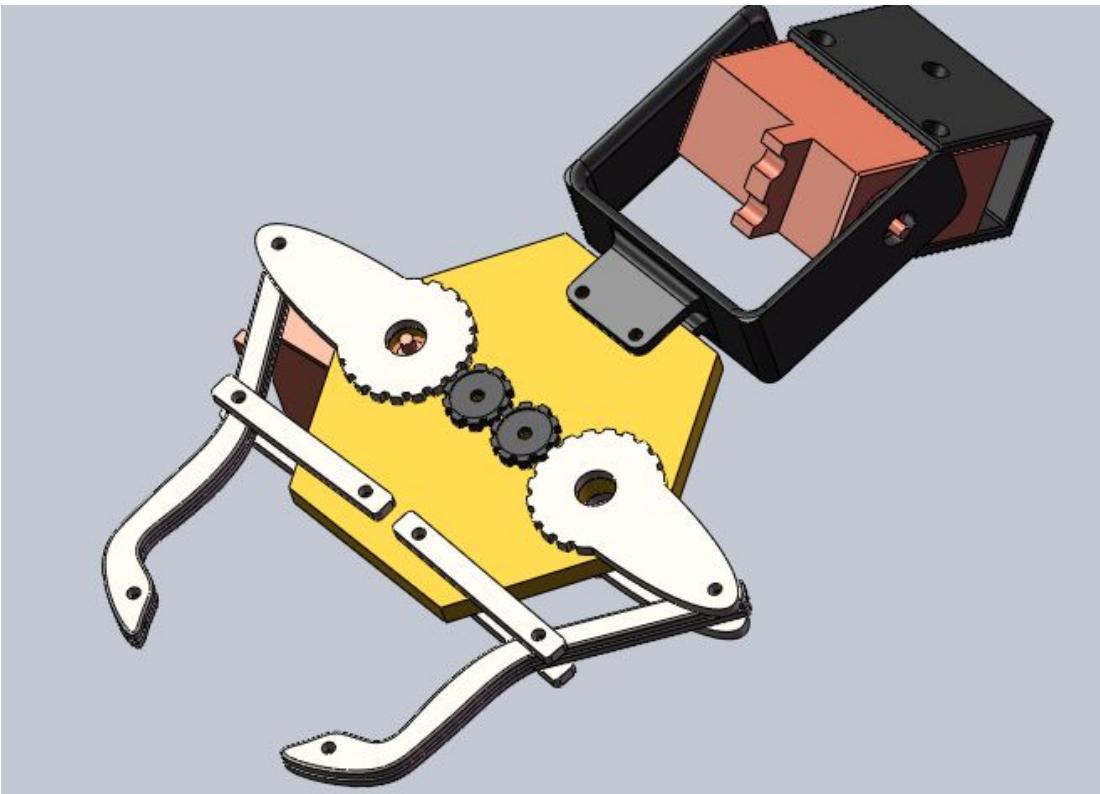


(后臂自由度演示2)

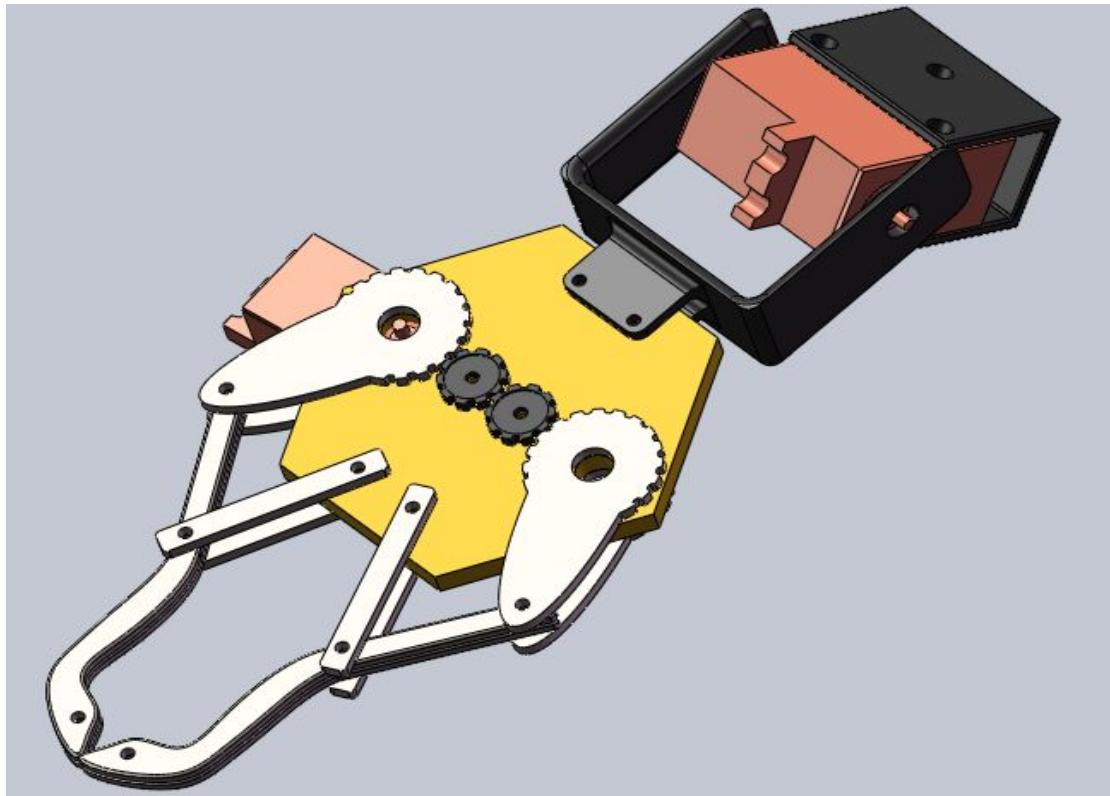
前臂的夹紧和放松如下图：



(前臂自由度演示1)



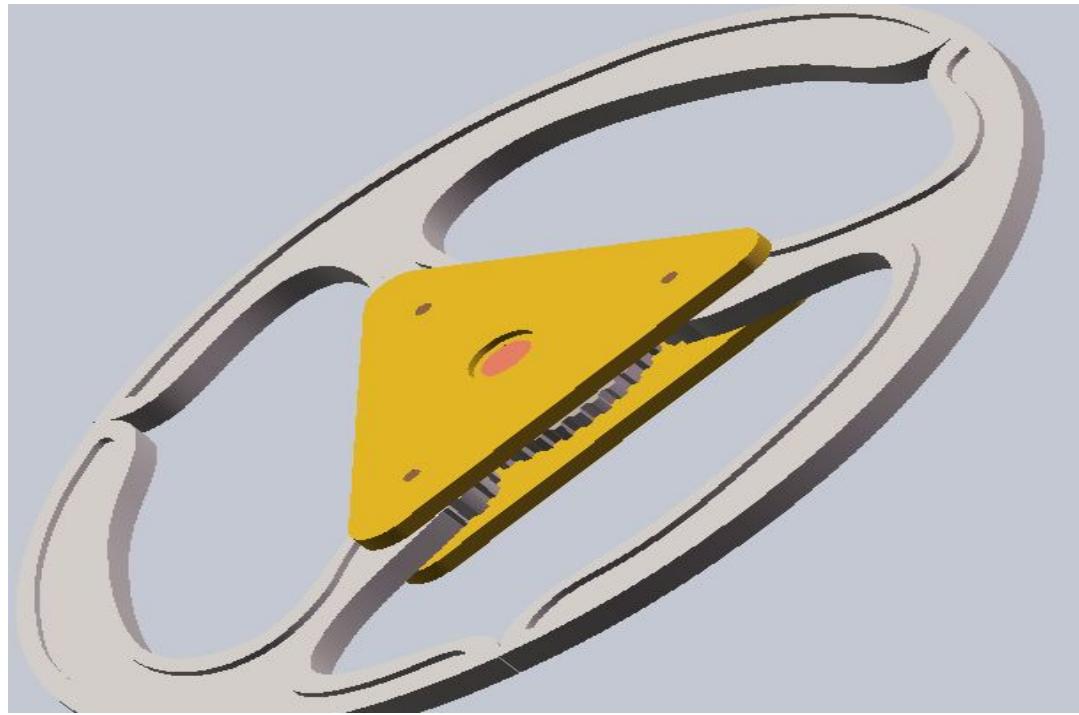
(前臂自由度演示2)



(前臂自由度演示3)

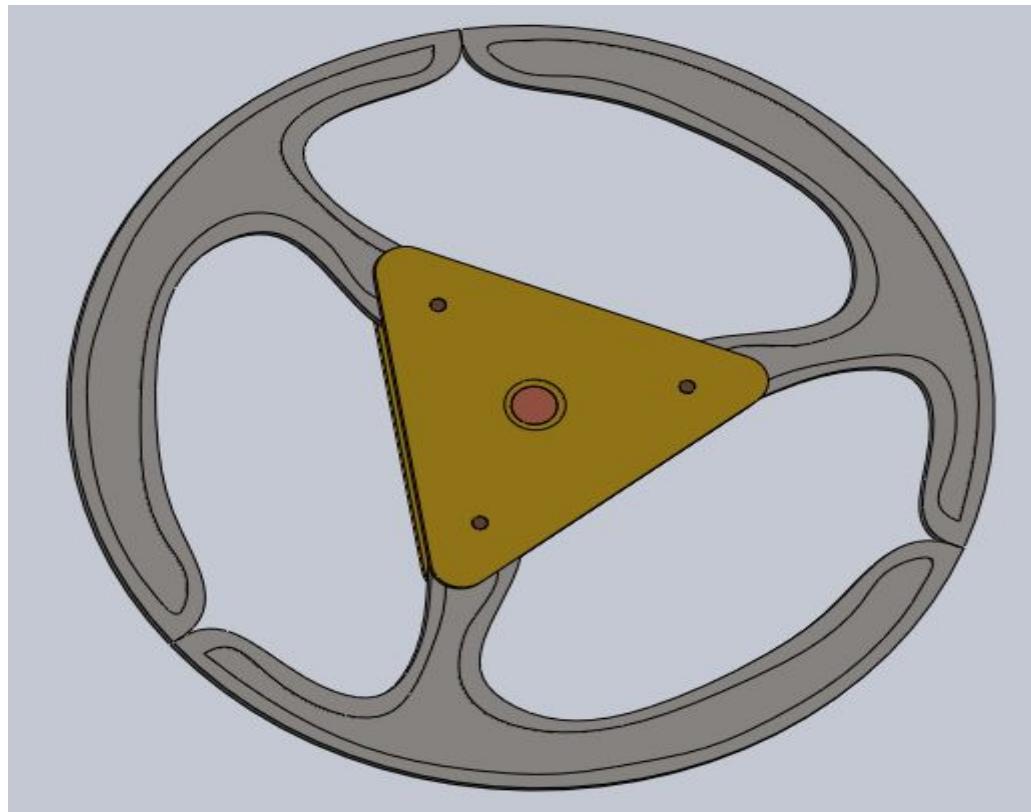
3.下面介绍可伸缩叶轮部分：

可伸缩叶轮整体模型图如下：



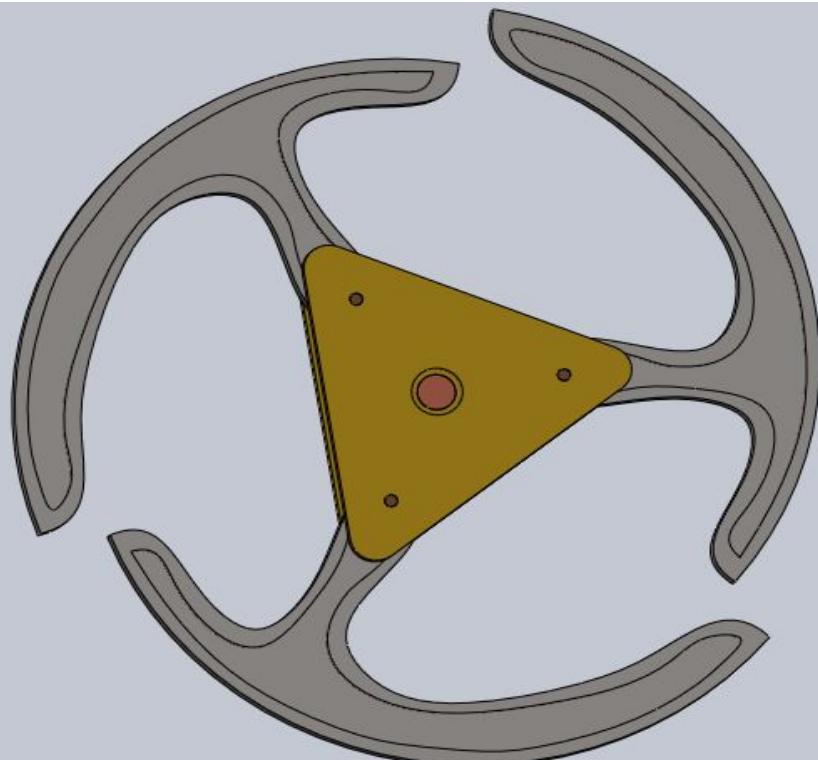
(可伸缩叶落整体图)

轮子走平地时如下图：

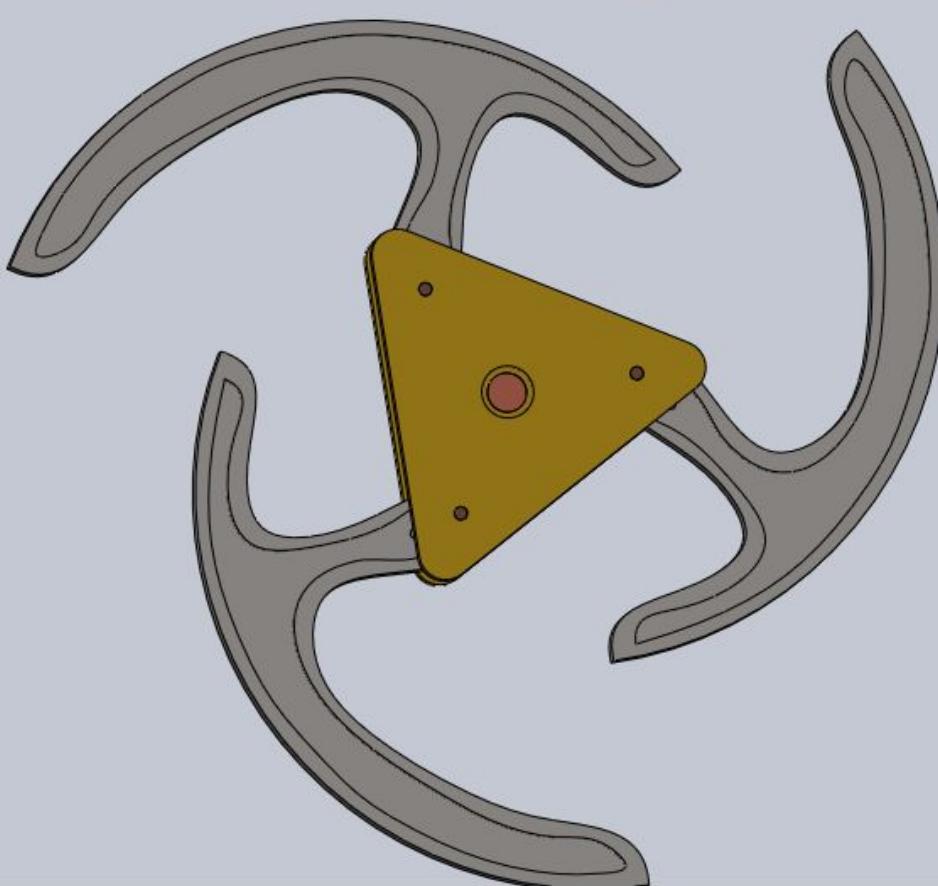


(正常走平地时轮子状态)

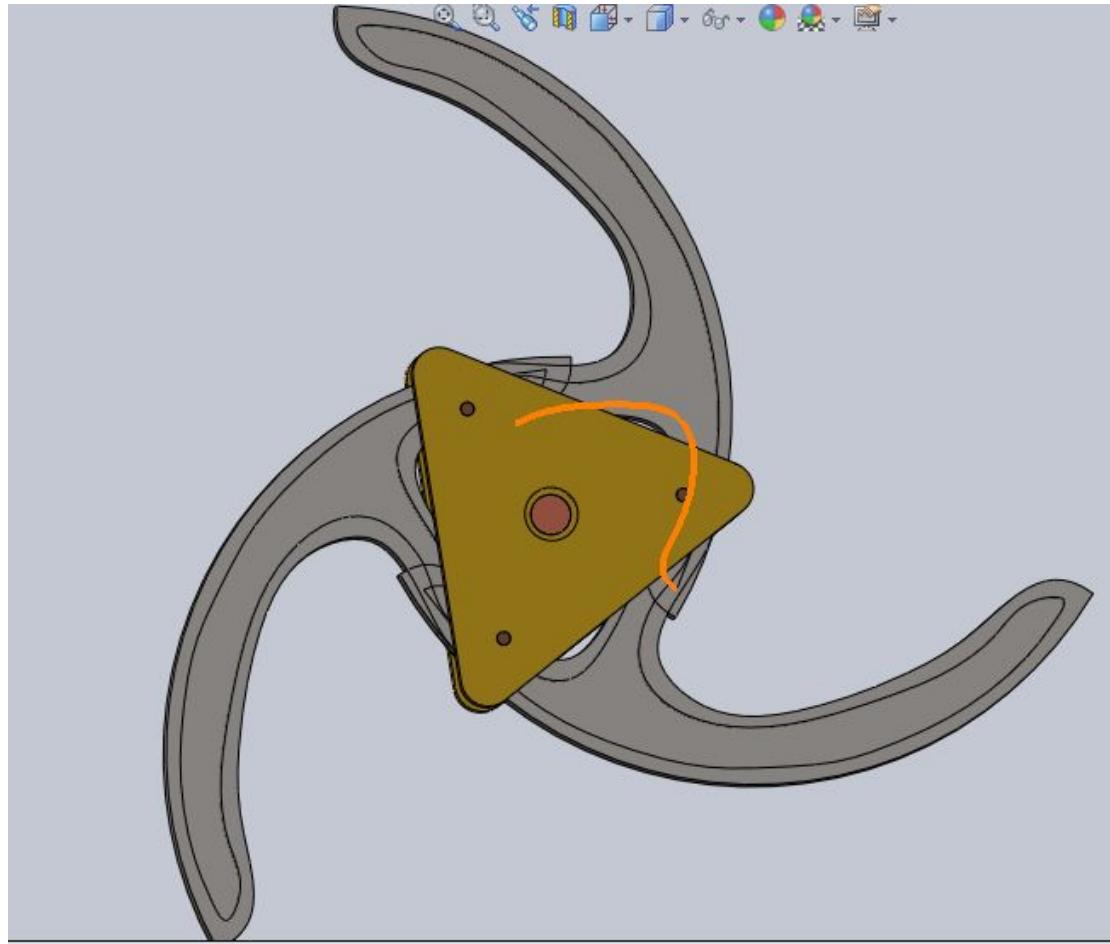
叶轮可以绕中间轴转动，如下图：



(可伸缩叶轮变形演示1)

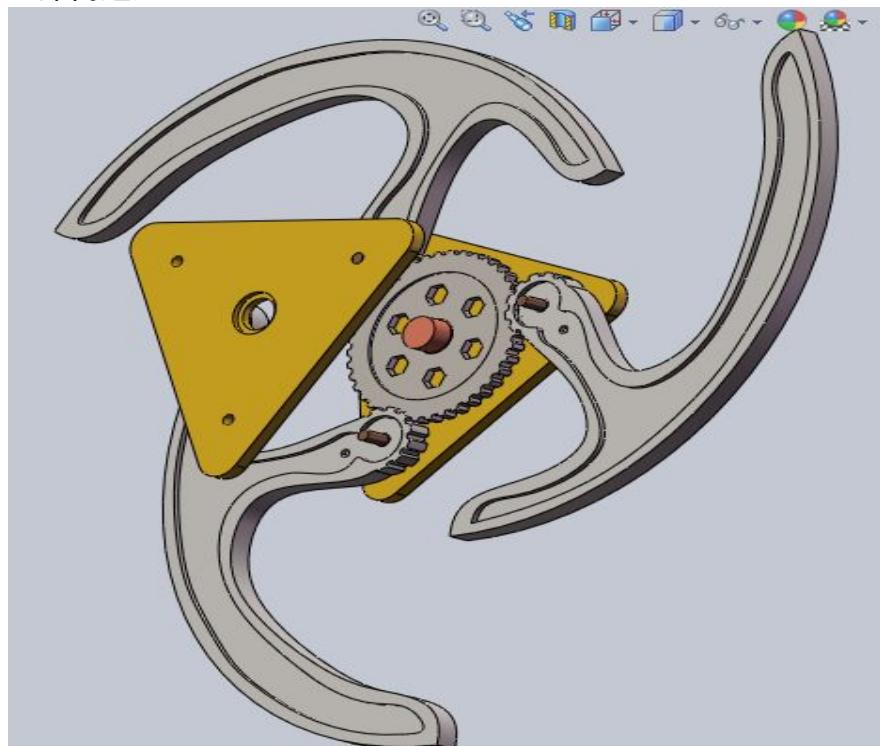


(可伸缩叶轮变形演示2)

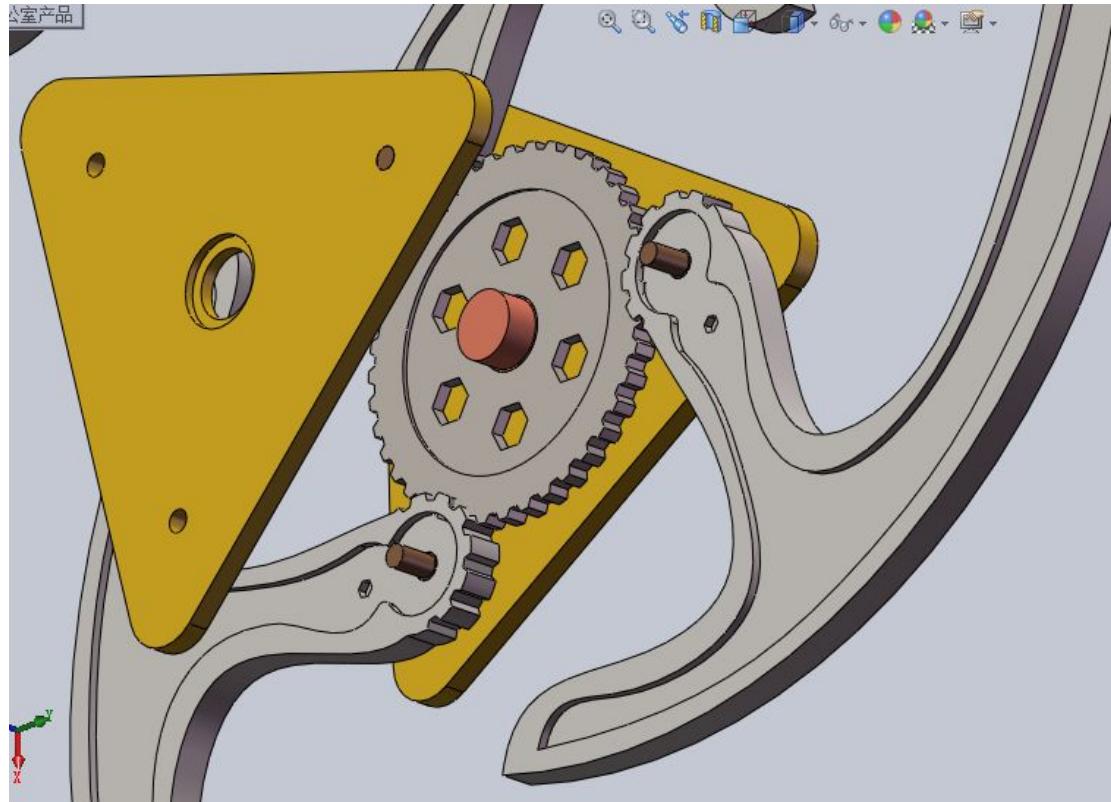


(可伸缩叶轮变形演示3)

内部构造如下：

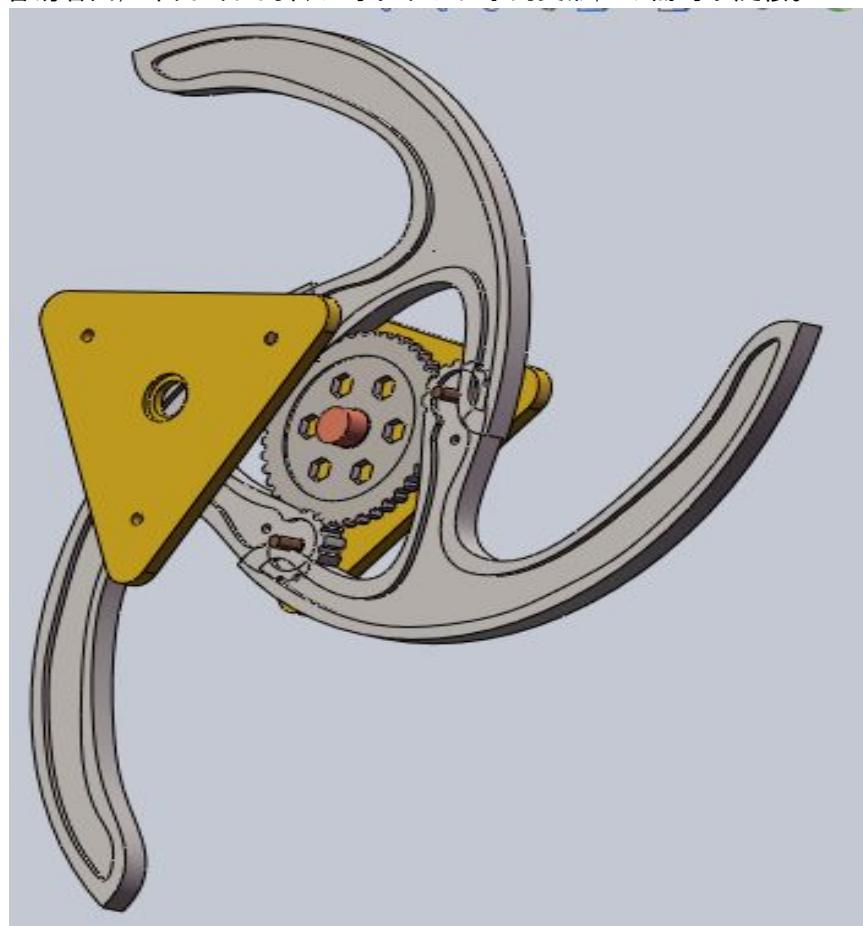


(内部构造)

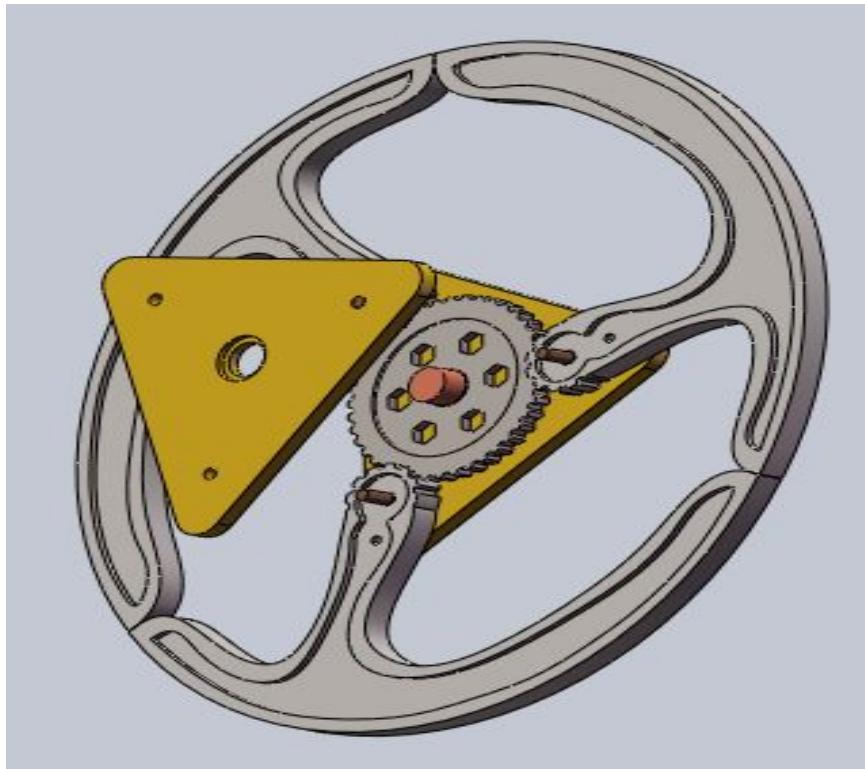


(内部构造大图)

容易看出，中间齿轮的转动可以带动轮子的变形，从而可以爬楼。



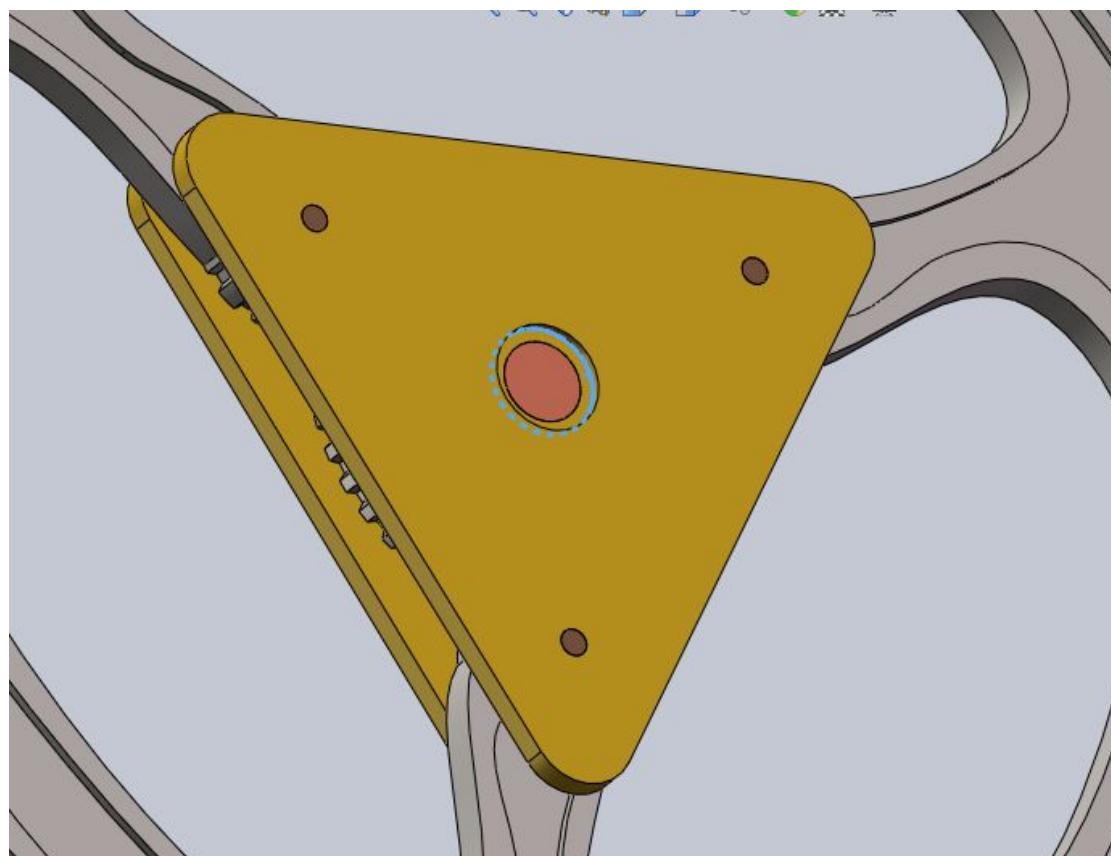
(内部齿轮传动1)



(内部齿轮传动2)

关于具体怎么爬楼，和普通行星轮类似。

下图反映了控制变形的轴和控制整个轮子转动的轴是分开的：

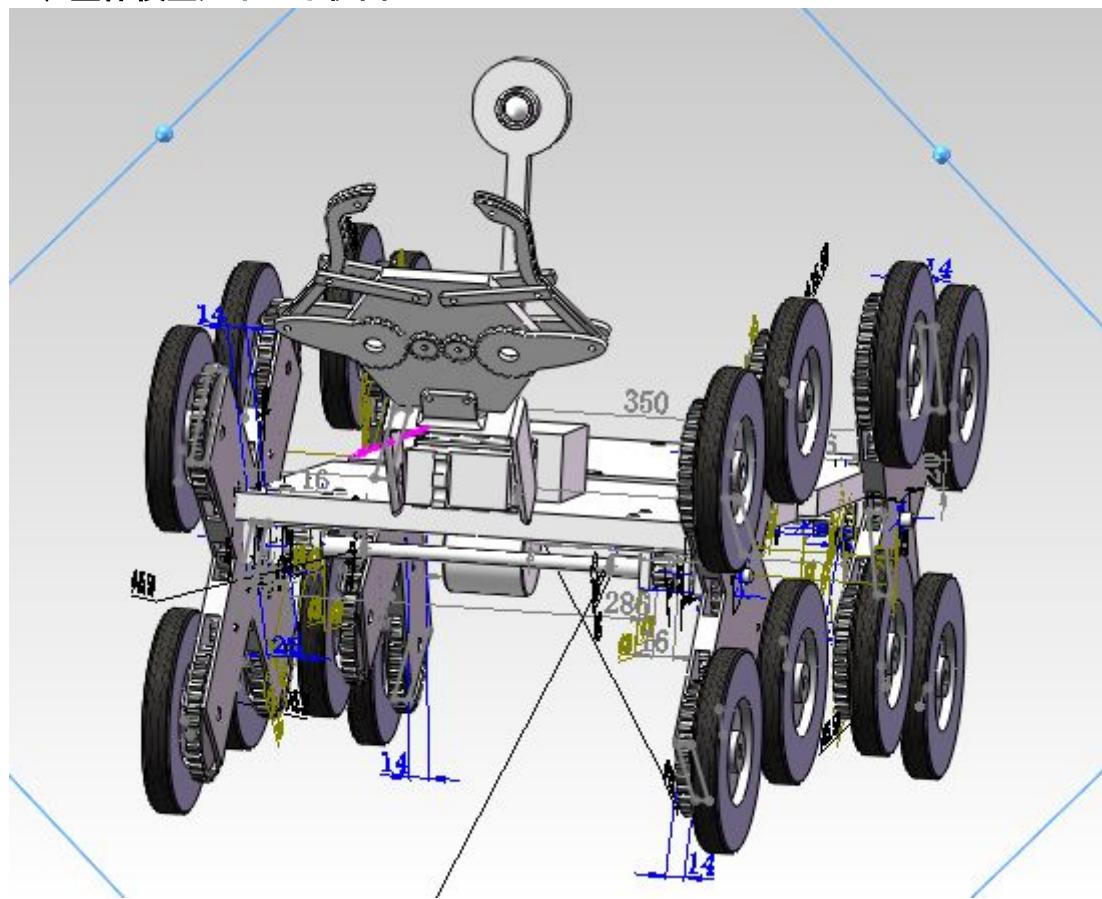


(轴间传动关系图)

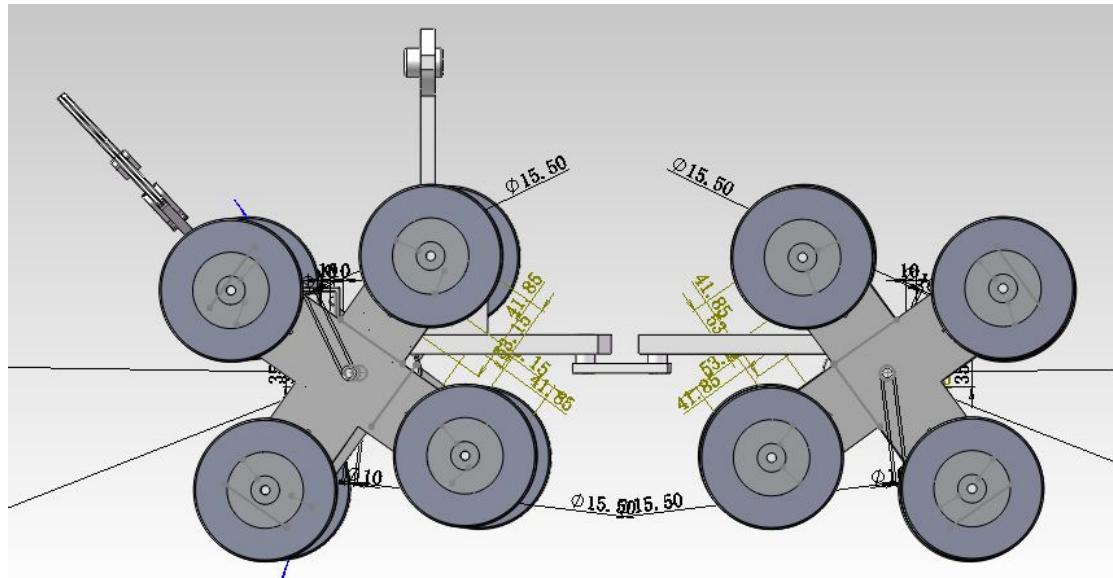
B. 方案二：行星轮式

行星轮机器人方案机械图及其详解

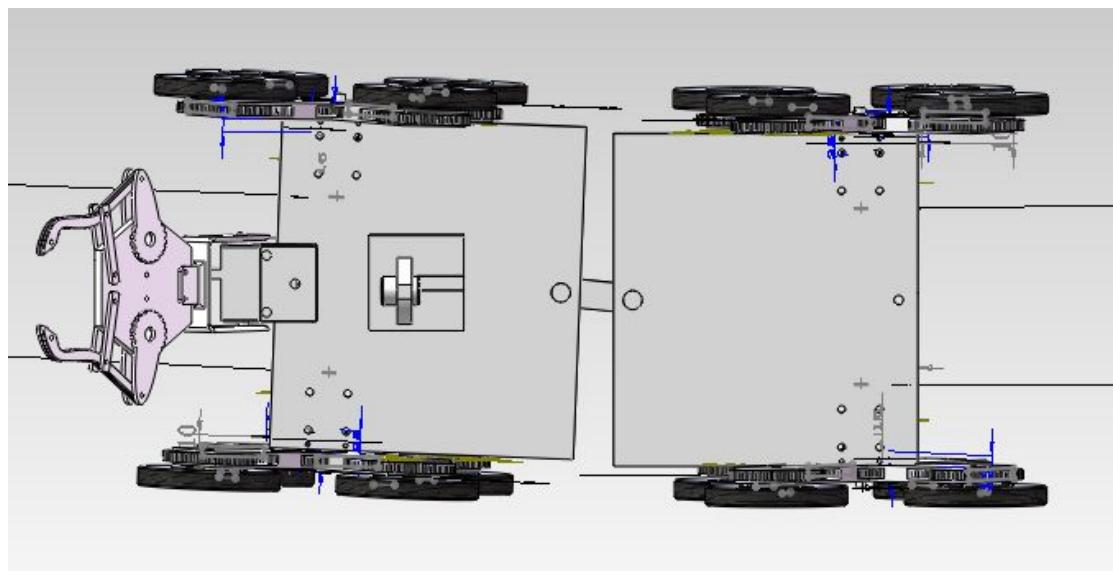
一、整体模型如下三个视图



图一 机器人前视图



图二 机器人侧视图

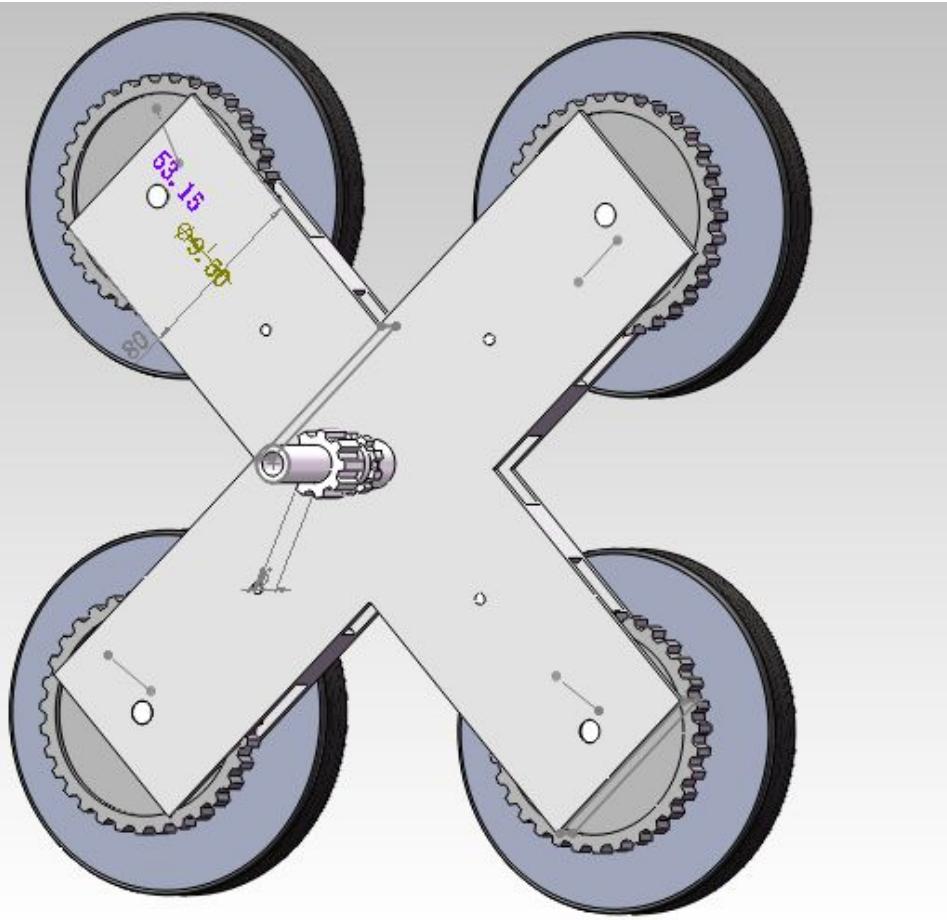


图三 机器人俯视图

二、行星轮机器人详解

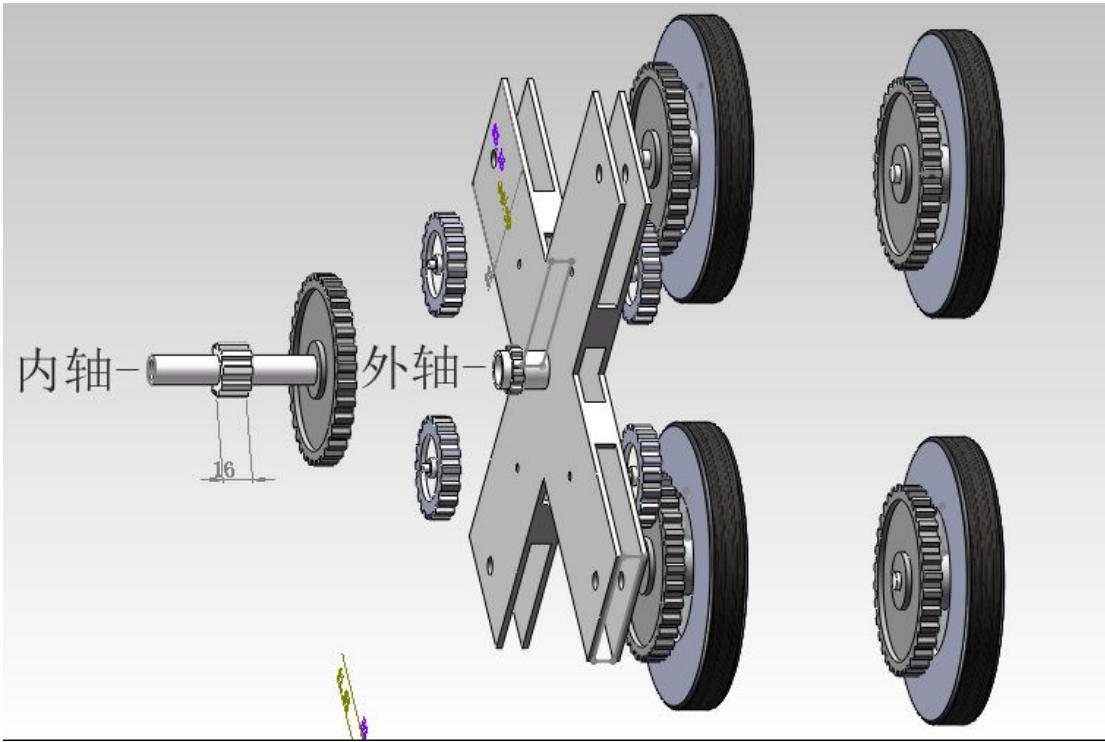
1、行星轮部分（图四）

1.1 行星轮整体



图四 行星轮整体模型

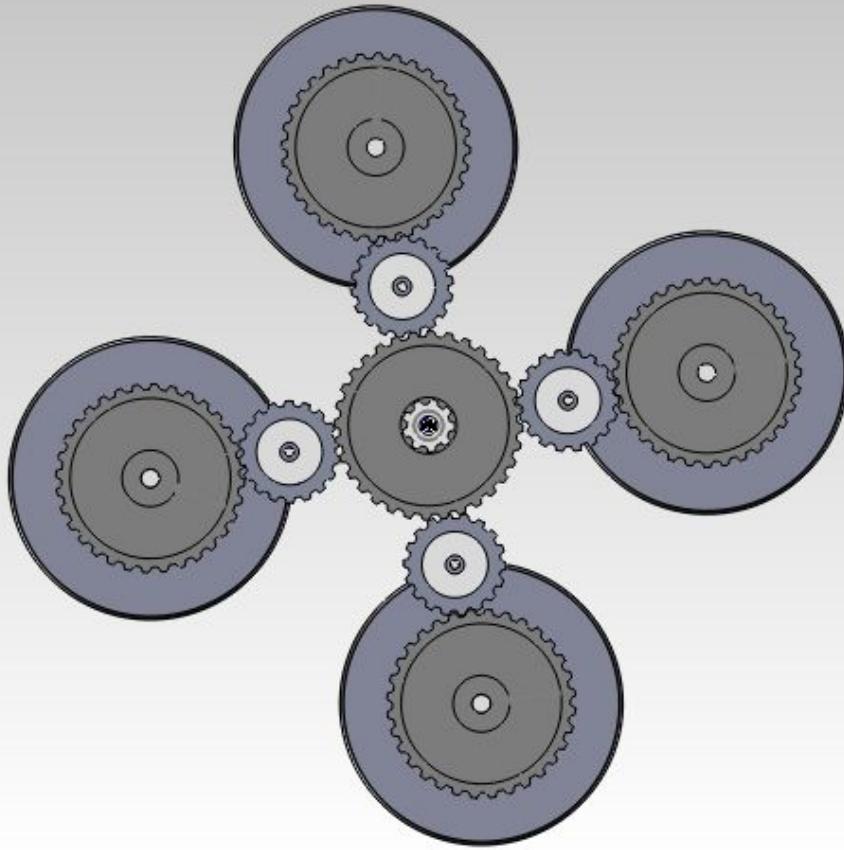
1.2 行星轮的解剖图



图五 行星轮的分解图

根据图五行星轮的分解图，由普通带齿轮小车轮构成一只最小车轮，四个这样的最小车轮构成一只最大车轮，因外形酷似星星，故叫做行星轮。图中外轴与行星轮的十字架固定，当外界为外轴齿轮提供动力时，行星轮将转动，这样通过四个最小车轮的交替着地可以实现机器人攀爬阶梯；内轴与四个传递齿轮相磨合，而传递齿轮再与最小车轮上的齿轮磨合，这样实现了内轴控制四个最小齿轮的转动，当内轴齿轮被给与动力时，将带动最小齿轮的转动，这样机器人就可以在平地上移动。

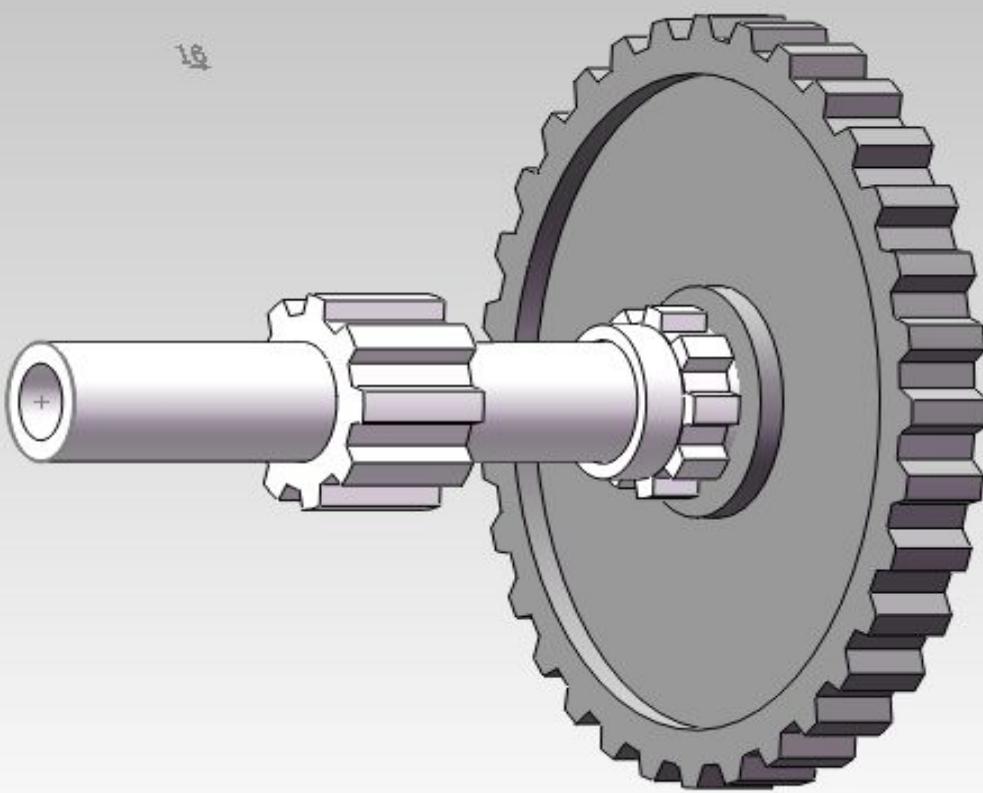
1.3 行星轮内部结构



图六 行星轮的内部结构图

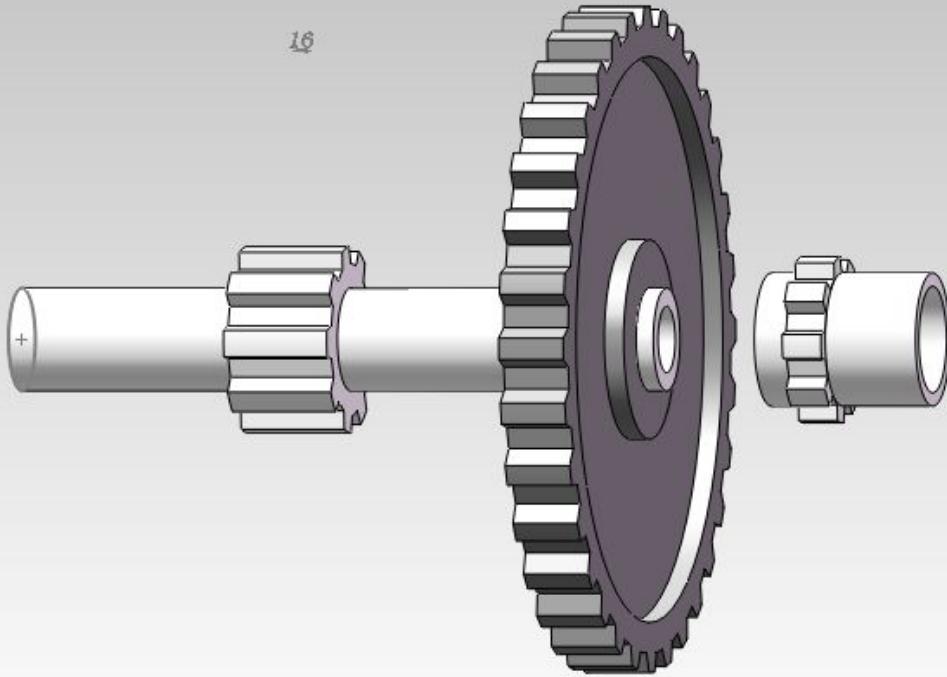
从图六中可以清晰地看出行星轮的内部结构，中间的大齿轮带动周边的四个小齿轮转动，小齿轮起到了传动作用，又带动了外围四个大齿轮转动，外围大齿轮与车轮固定，因此车轮能够转动，经过动态模拟，效果比较出色。

1.4 内轴、外轴



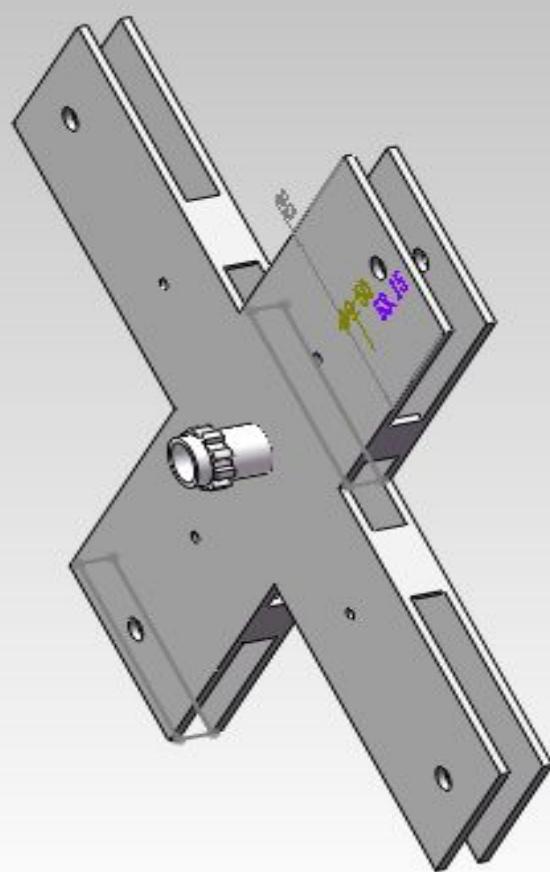
图七 内轴与外轴相磨合

内轴与外轴同心，但两轴互不影响。



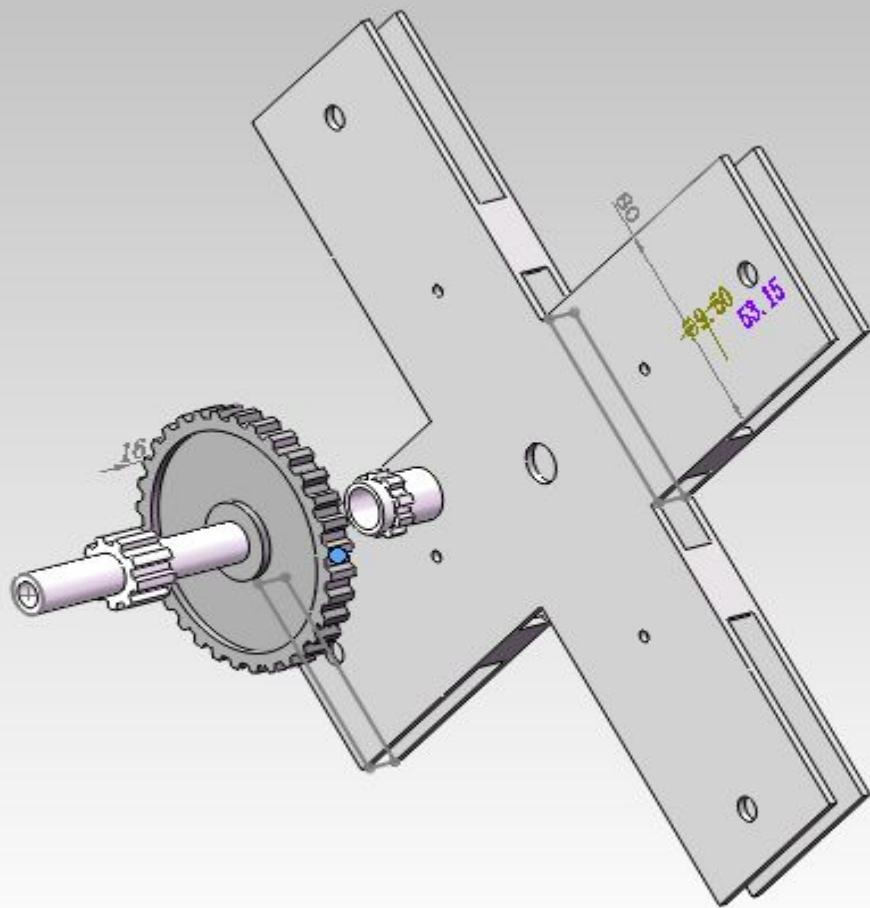
图八 内轴与外轴分开

内轴与大齿轮相粘合固定。



图九 内轴与十字架

内轴与十字架粘合固定，这样，内轴的转动就能带动十字架的转动。



图十

内轴、外轴、十字架同心共轴。

1.5 涉及成品



图 1-1 车轮

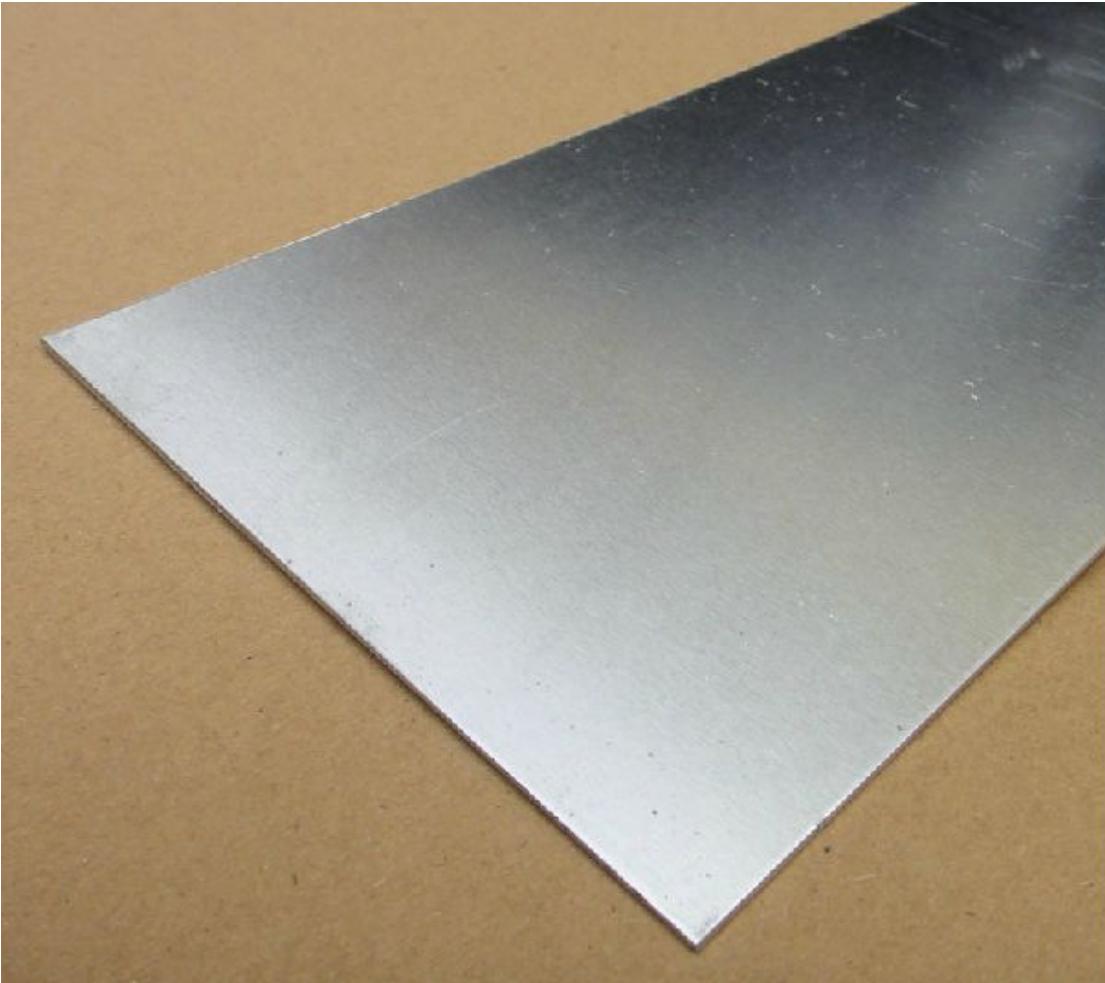


图 1-2 铝板



图 1-3 齿轮



图 1-4 带轴齿轮



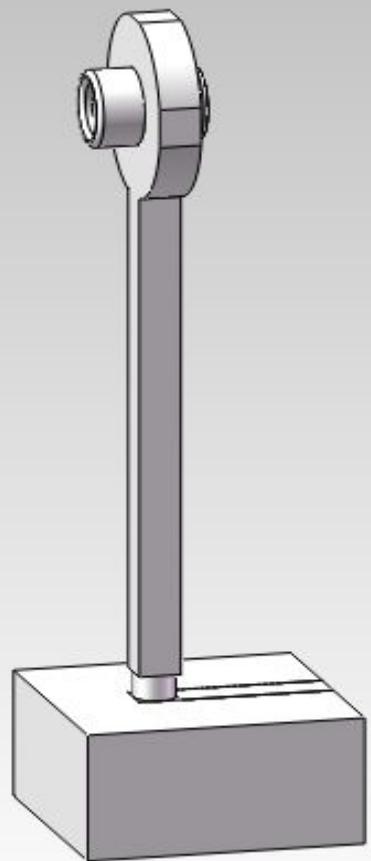
图 1-5 内轴



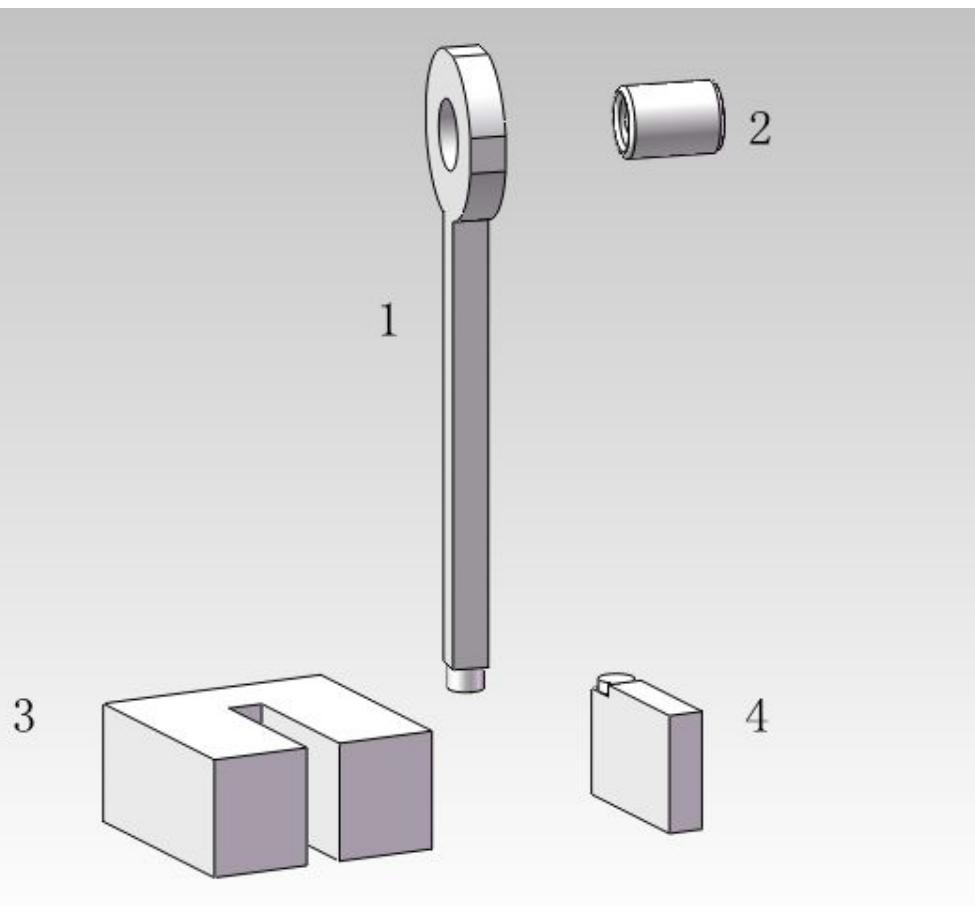
图1-6 外轴

2、摄像头部分

2.1 摄像头整体示意图（图十一）



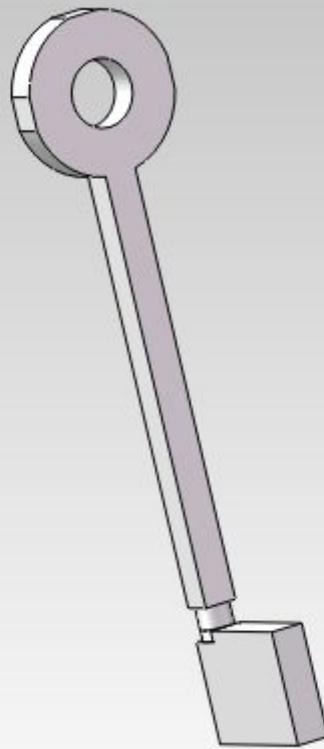
2.2 摄像头结构分解 (图十二)



图十二
图中：1、摄像头支架

- 2、摄像头
- 3、舵机固定体
- 4、舵机

其中，摄像头支架与舵机圆盘部分重合固定，使舵机能够控制支架的转动，如图十三所示。



图十三 舵机与摄像头粘合

舵机固定体，顾名思义，就是固定舵机的装置，可以较好地将舵机牢牢地固定在车身上；摄像头将嵌入摄像头支架的顶部，使之与支架、舵机连为一体。当舵机转动一定角度时，摄像头也将随之转动。

2.3 摄像头装置的假设

计划书中所设计的摄像头十分粗糙简易，具体使用时将会结合具体情况对摄像头进行选择与安装，但大体思路和计划书所述相差不多。

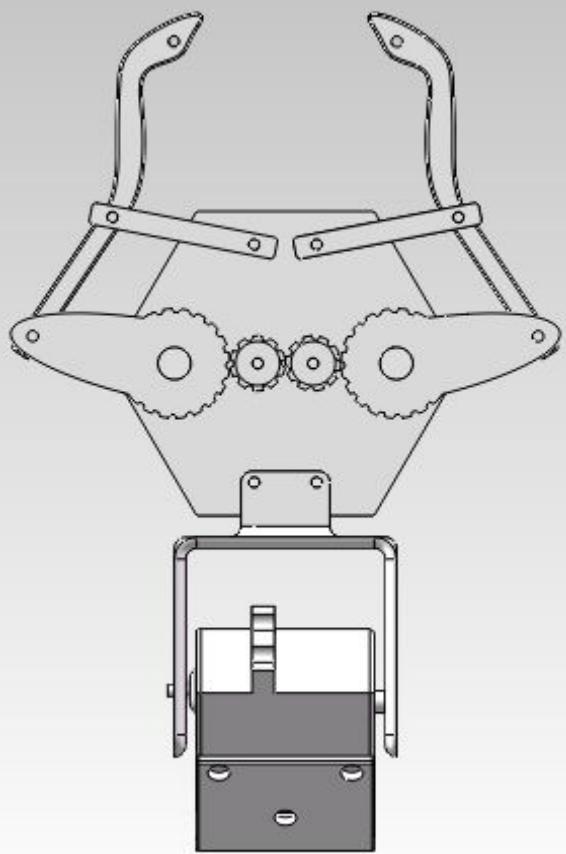
2.4 成品摄像头



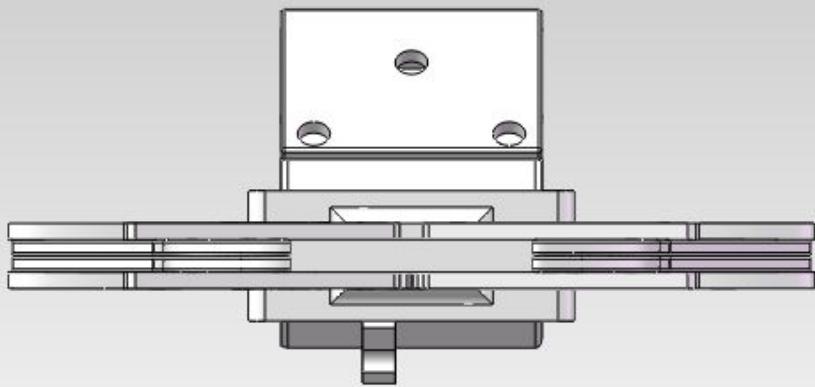
图十四 成品摄像头

3、机械臂部分

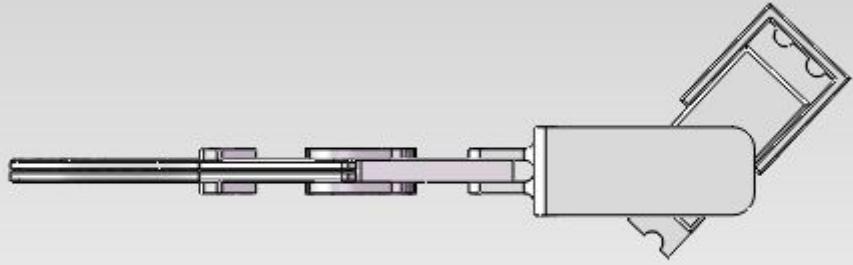
3.1 机械臂整体模型



图十五 机械臂俯视图

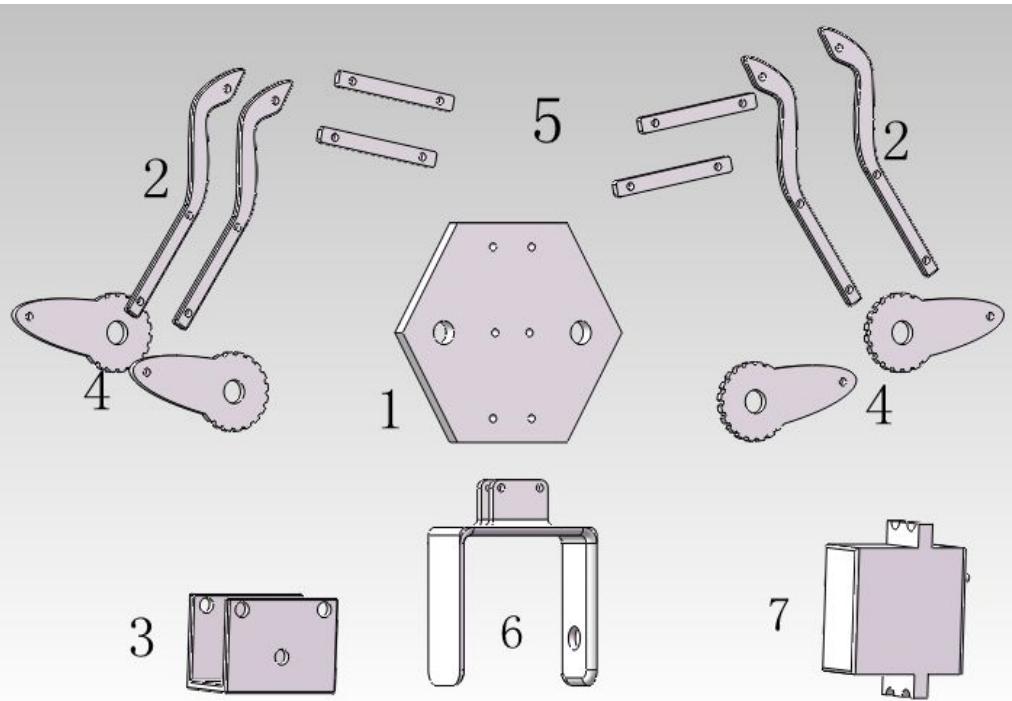


图十六 机械臂前视图



图十七 机械臂侧视图

3.2 机械臂分解图

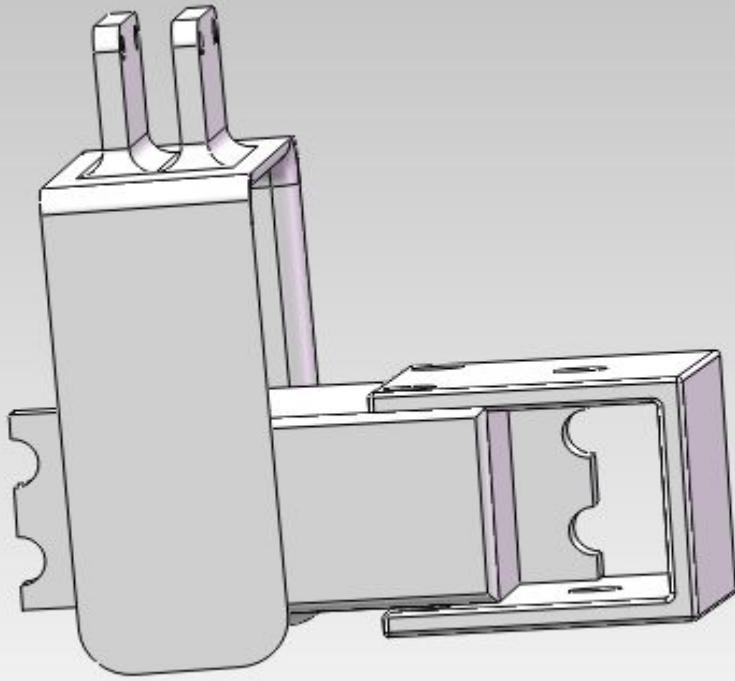


图十八 机械臂分解图

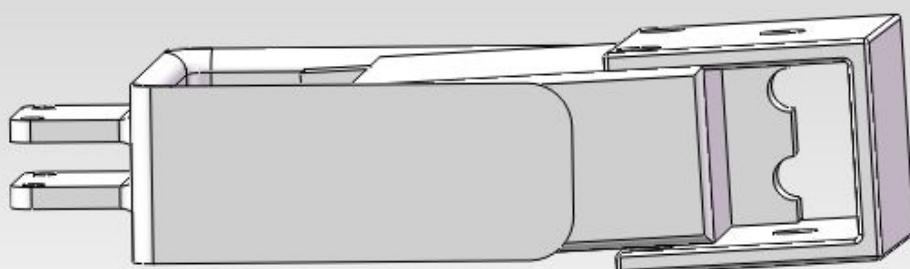
图中：1、机械臂平板

- 2、机械臂前肢
- 3、舵机固定架
- 4、机械臂后肢
- 5、机械臂稳固支架
- 6、机械臂上下控制支架
- 7、

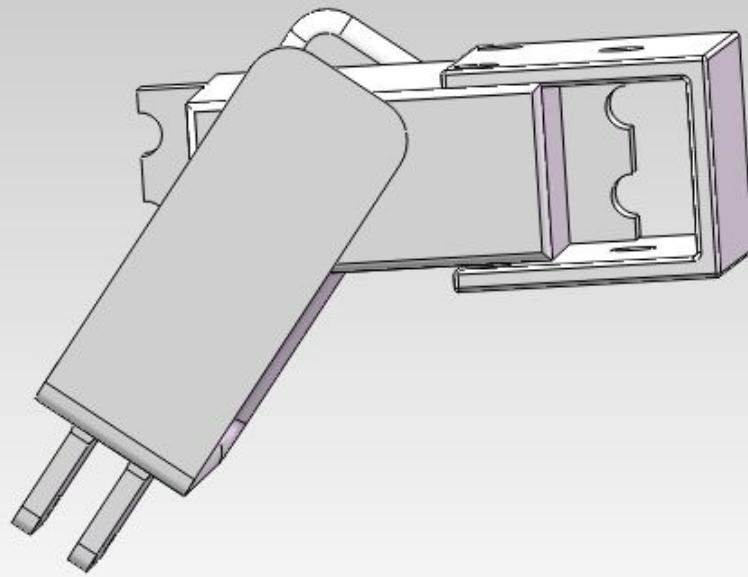
3.3 舵机控制机械臂上下



图十九 舵机控制上仰

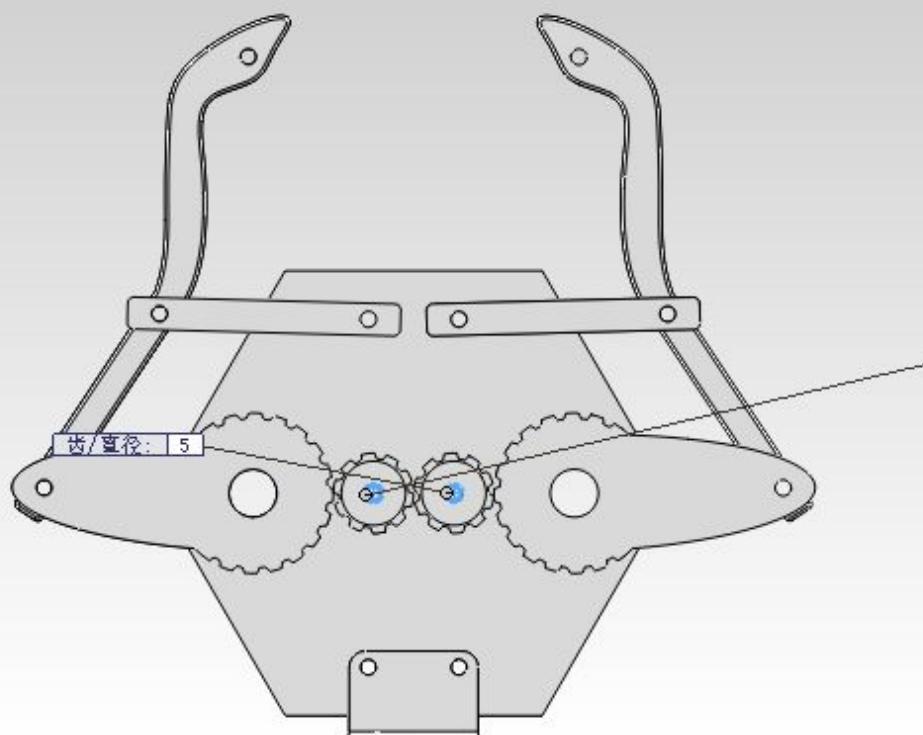


图二十 舵机控制持平

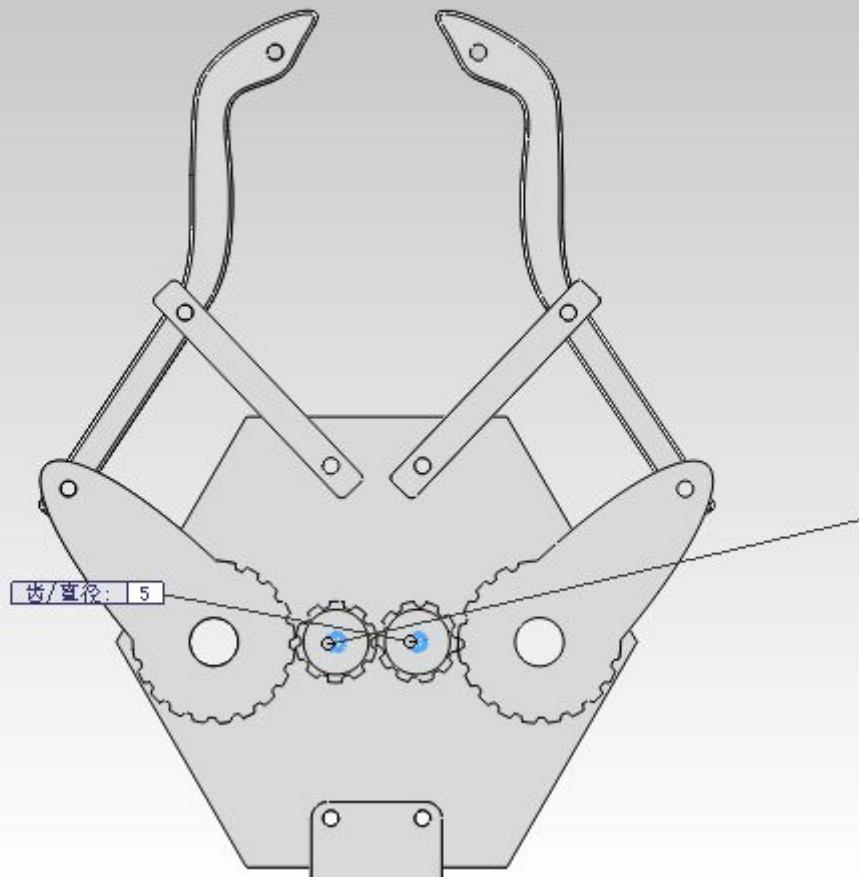


图二十一 舵机控制下倾

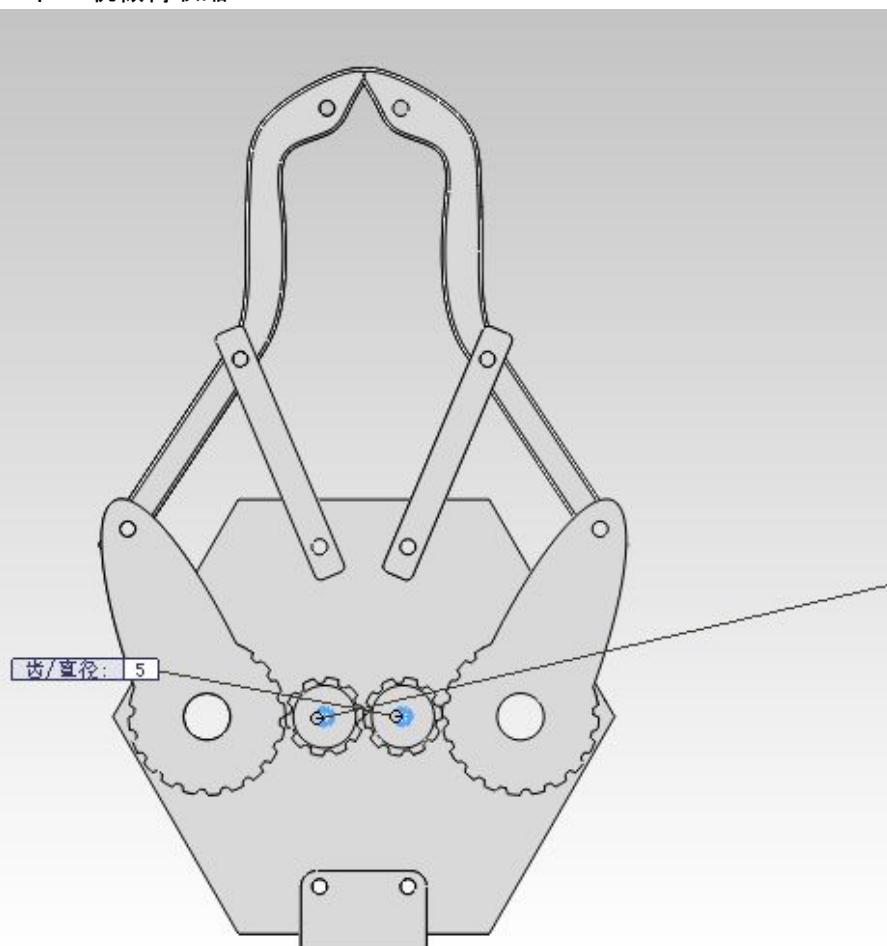
3.4 机械臂夹持



图二十二 机械臂张开情形



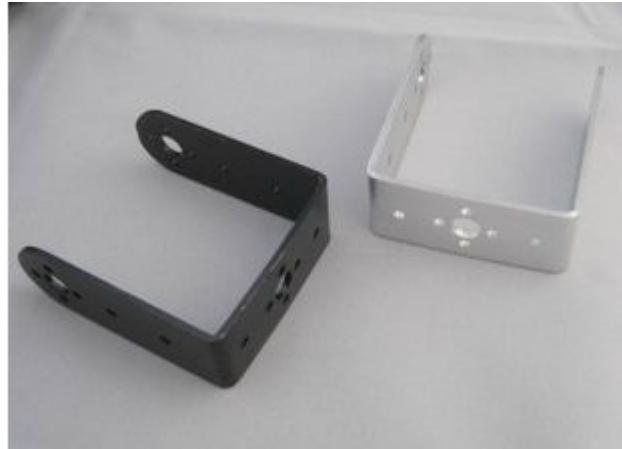
图二十三 机械臂收缩



图二十四 机械臂夹持

图中只给出控制机械臂上下的舵机，事实上还有控制机械臂夹持的舵机，通过舵机控制齿轮部分的转动便可带动后肢、前肢的伸缩。

3.5 机械臂相关实物图



图二十五 舵机固定架



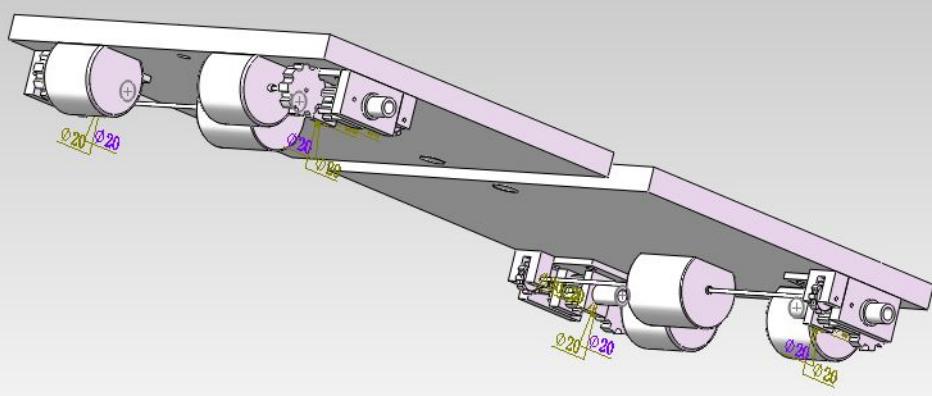
图二十六 机械臂夹持部分



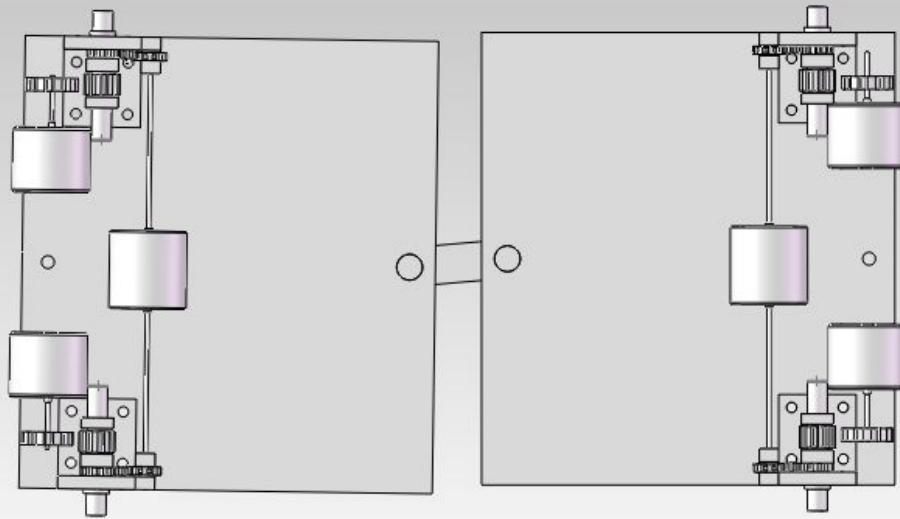
图二十七 舵机

4、车身部分

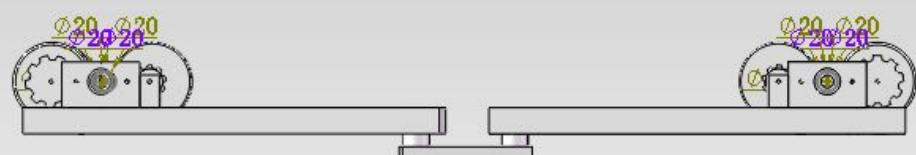
4.1 车身整体模型



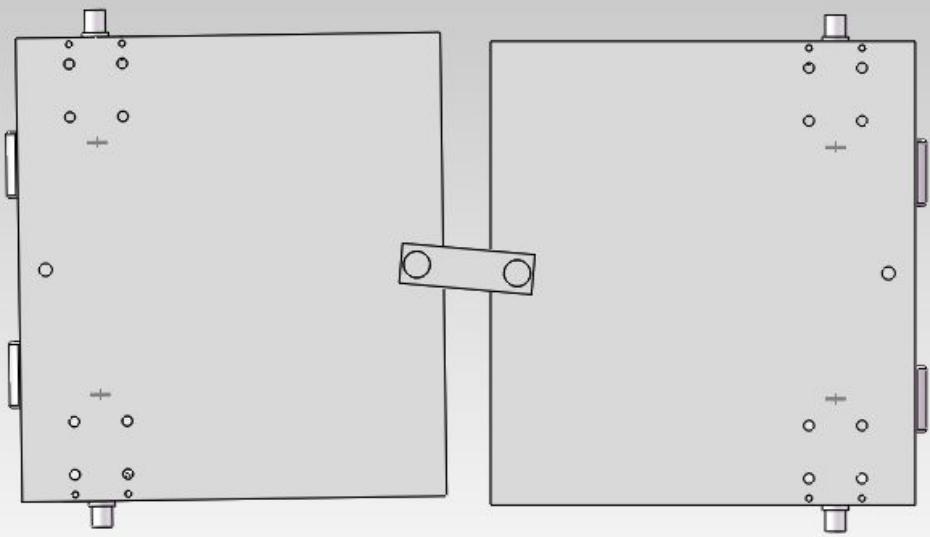
图二十八 车身



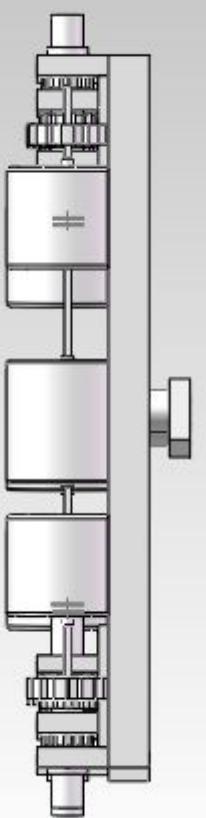
图二十九 车身仰视图



图三十 车身侧视图

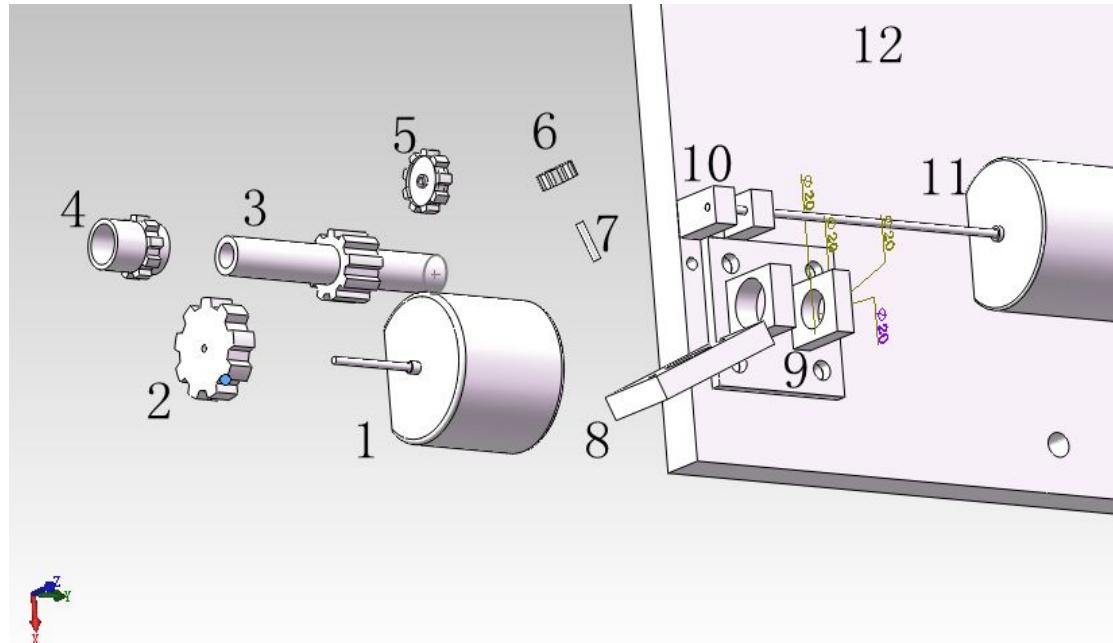


图三十一 车身俯视图



图三十二 车身前视图

4.2 车身分解图

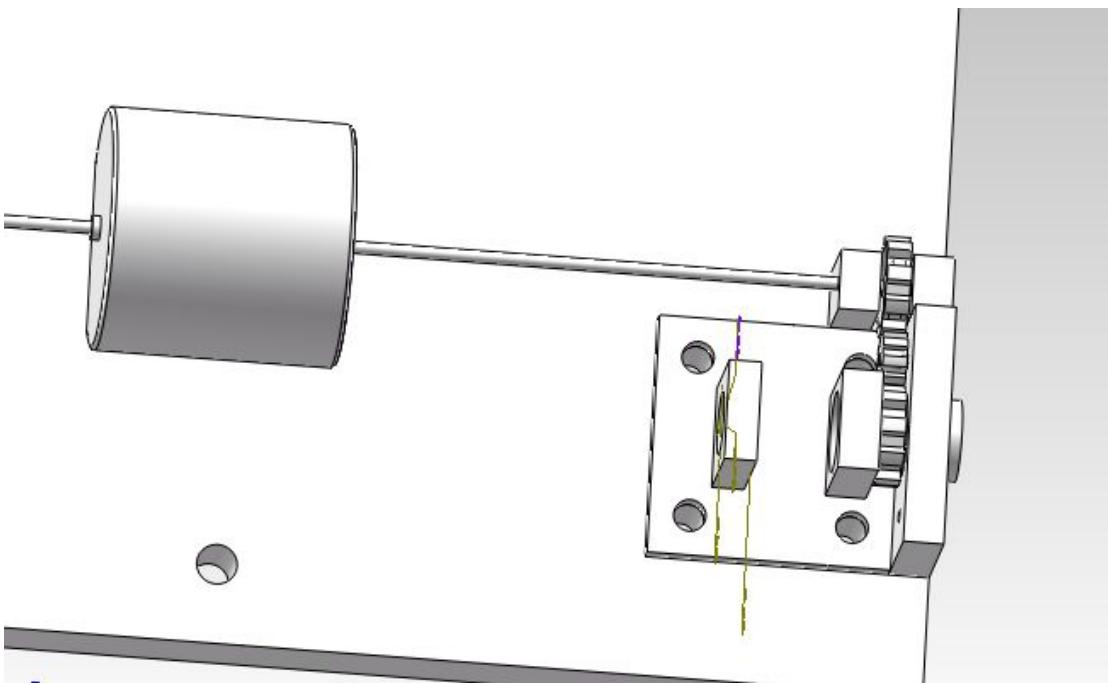


图三十三 车身分解图

图中：1、驱动电机

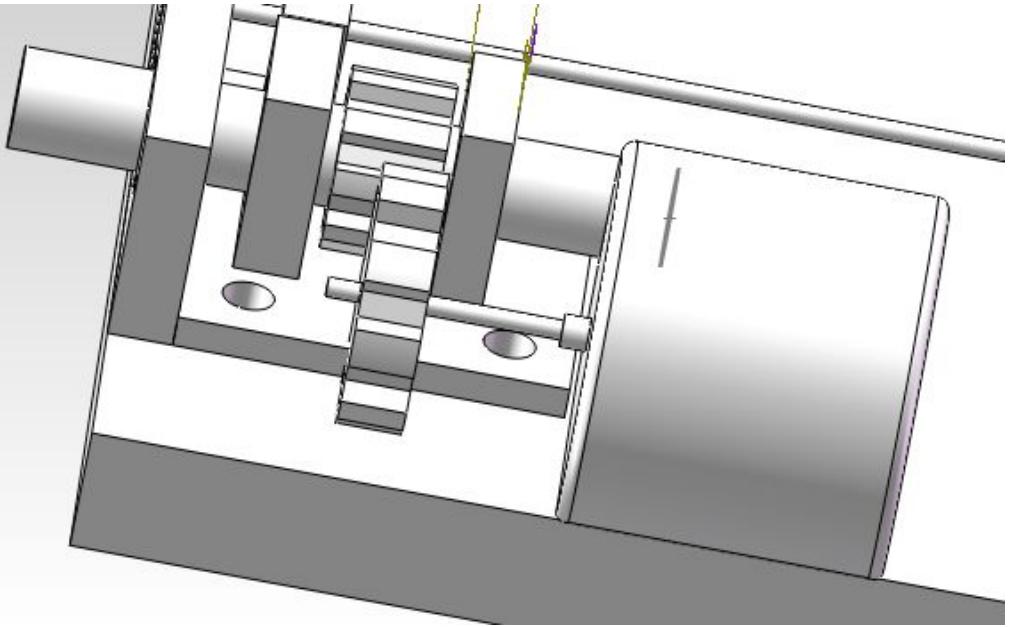
- 2、齿轮
- 3、内轴
- 4、外轴
- 5、齿轮
- 6、传递齿轮
- 7、传递轴
- 8、外轴支架
- 9、内轴支架
- 10、双轴电机支架
- 11、双轴电机
- 12、车身平板

4.3 电机驱动详解



图三十四 外轴电机驱动

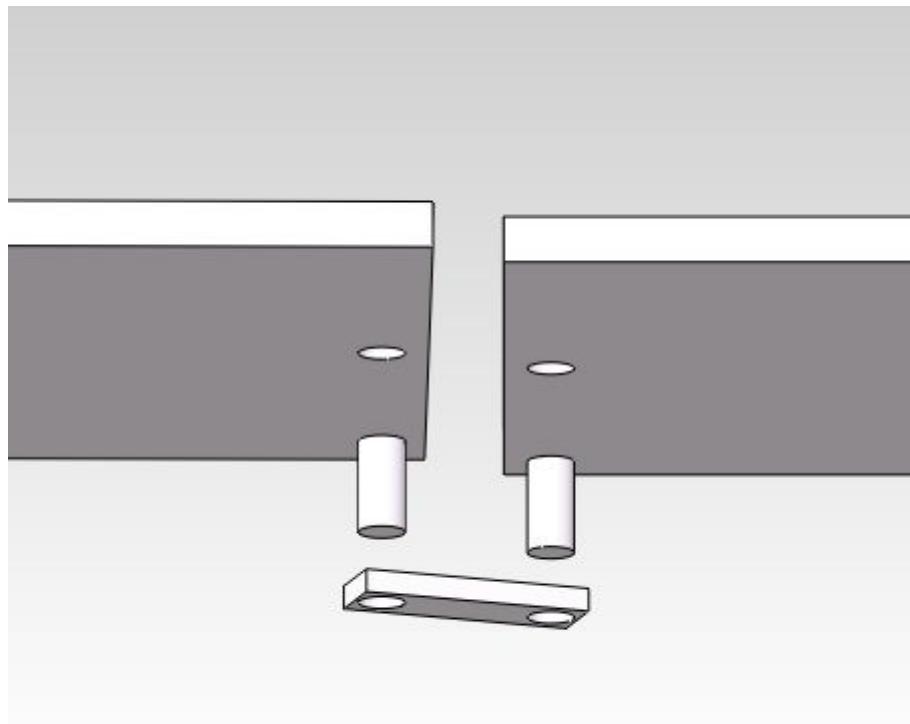
图三十四反应的是电机驱动外轴的情形，电机一段固定一齿轮，通过传递齿轮使得外轴在齿轮的作用下转动起来，从而带动行星轮的转动。由于上下楼梯前后端的行星轮是同步的，因此可以用同一个电机控制相同端的行星轮。



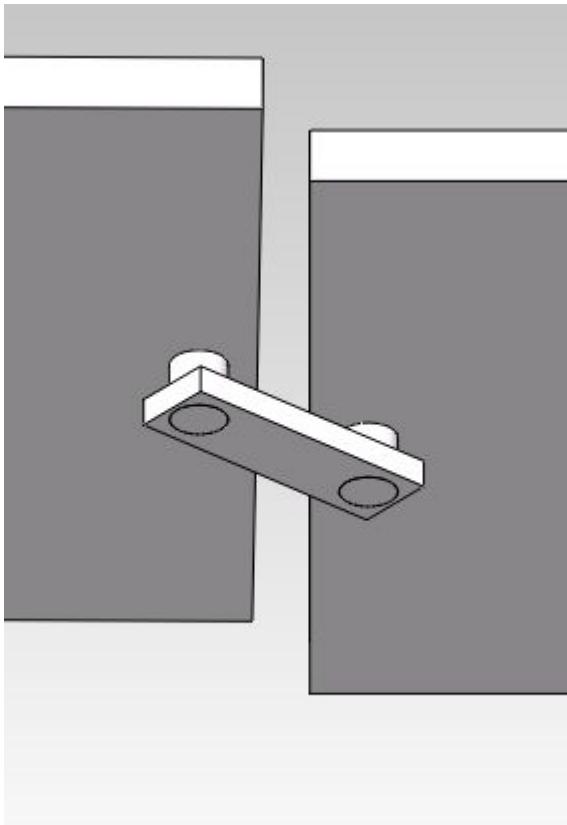
图三十五 内轴驱动

图三十五反应的是电机驱动内轴的情形。由于最小车轮会涉及转弯等实际情况，因此每只行星轮配一只电机，这样通过速度控制实现车的转向问题。当然这种转向方案存在一定问题，如果实际对转向要求很高的话，我们将考虑使用舵机进行方向的控制。

4.4 车身连接轴



图三十六 车身连接轴分解



图三十七 车身连接轴粘合

车身连接轴的使用主要是方便车转动方向。

4.5 车身相关实物图



图三十七 双轴电机



图三十八 铝板

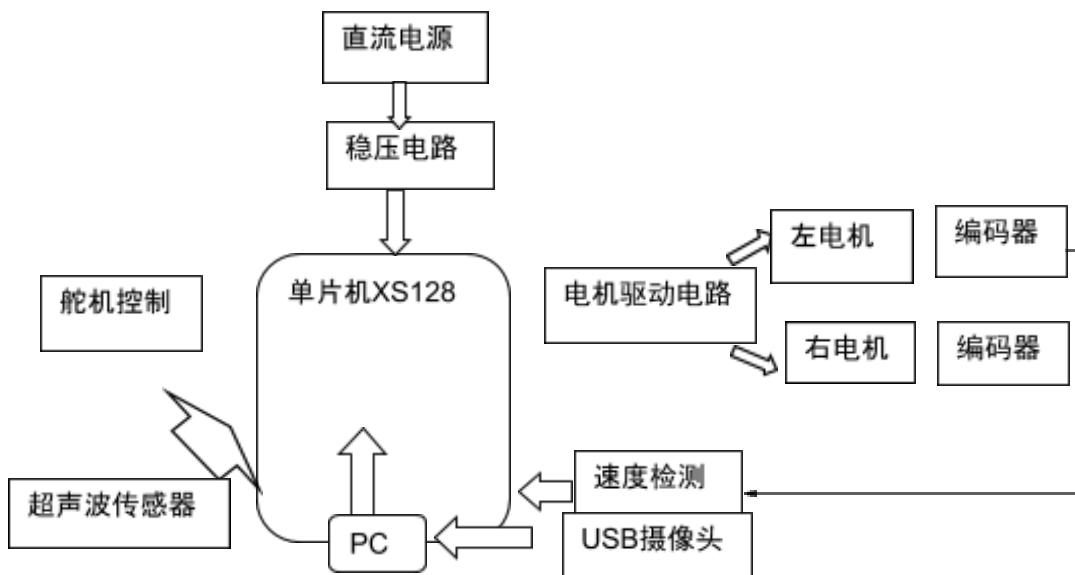


图三十九 驱动电机

电路部分

由于具体的电路需要根据实际的硬件选取来确定，计划书中只能写出电路的各个模块的参考方案，在做机器人的过程中，再根据实际情况进行电路原理方面的调整。

1. 整体电路总框图

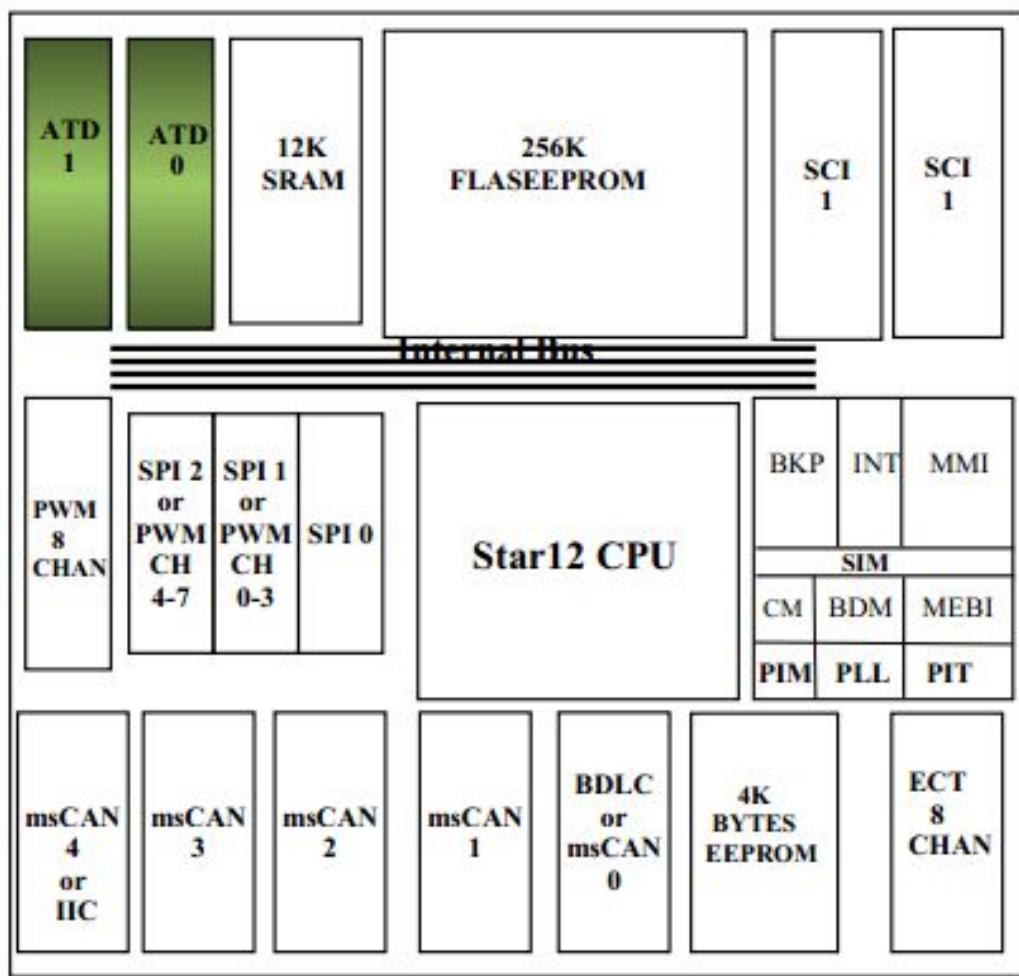


2. 单片机的选取

鉴于我们做过“飞思卡尔智能车”，对飞思卡尔公司的单片机比较熟悉，我们选用MC9S12XS128，作为机器人的主控芯片。

相比之前师兄们所用的AVR系列单片机，这款单片机有很多优势。

XS128系统总框图如下：



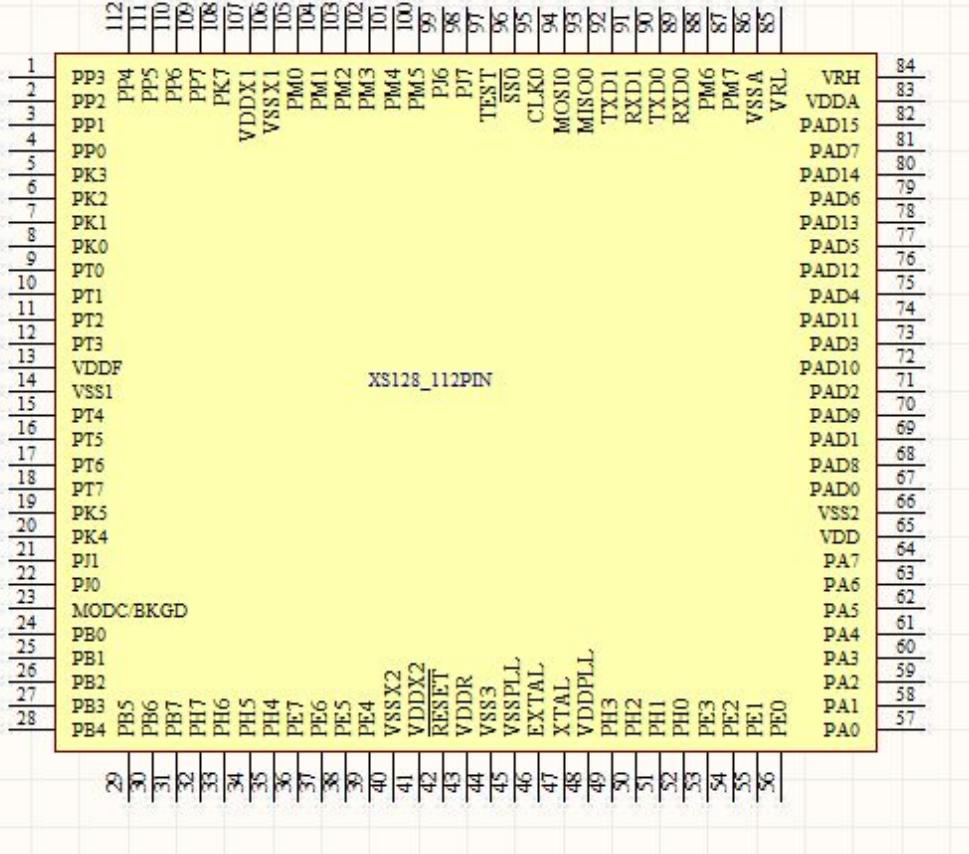
特点：

- ☆ XS128以复杂指令集CISC架构，集成了中断控制器，有丰富的寻址方式，CPU频率80MHz，可超频到120MHz，拥有更快的处理速度，满足我们机器人的需要；
- ☆ 有7个中断优先级并且内核支持优先级的调度，最多可有117个中断源。
- ☆ 具有丰富的外围接口，SCI, SPI, CAN, AD等等
- ☆ PWM调制波有8个输出通道，每个输出通道都可以独立的进行输出，相邻的两个输出通道也可以级联成一个16位的输出通道。每一个输出通道都有一个精确的计数器，一个周期控制寄存器和两个可供选择的时钟源，每个PWM输出通道都能调制出占空比从0-100%变化的波形。
- ☆ ECT模块可对输入波形进行测量，同时也可产生一个输出波形。可用来测量编码器的脉冲个数，来进行速度的控制。
- ☆ A/D模块8,10,12位精度可选，转换速度快，可编程采样时间，左右对齐，有符号/无符号结果输出

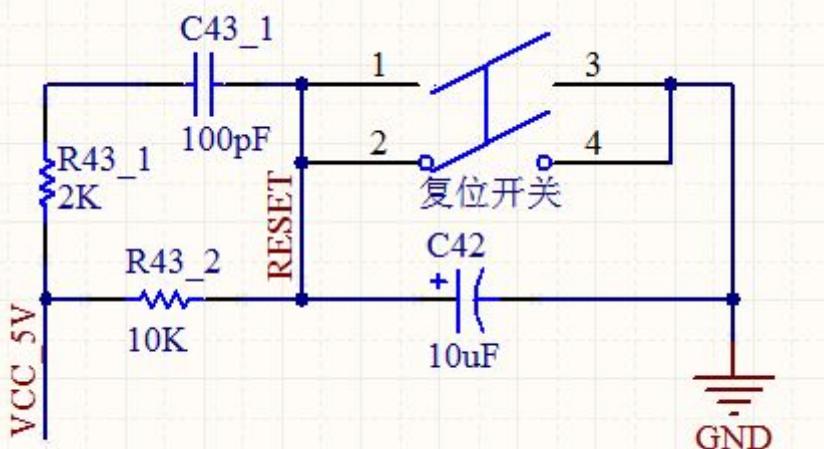
因此XS128上集成的各种模块完全可以满足我们的需要，避免了再用其他芯片的麻烦，电路也更加稳定。

部分单片机电路：

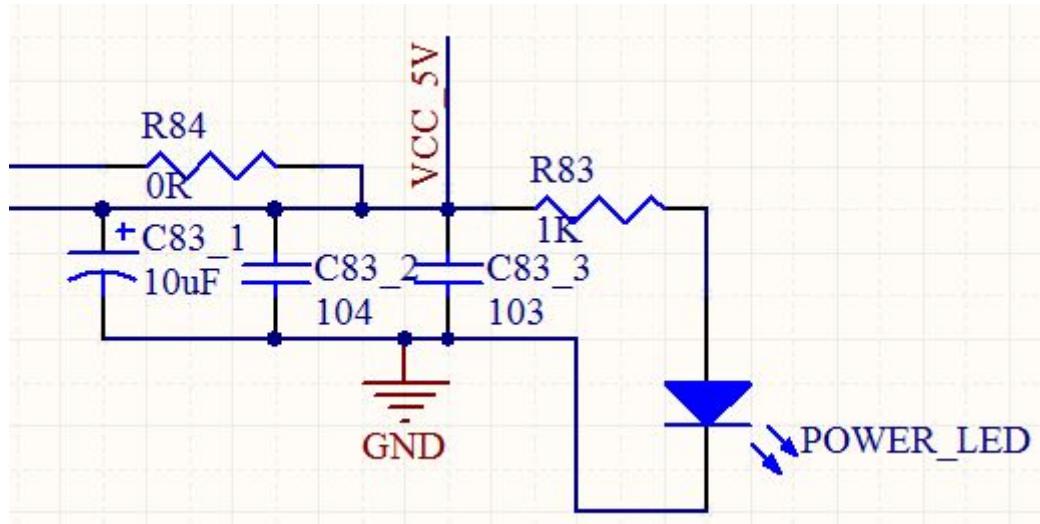
XS128引脚图：



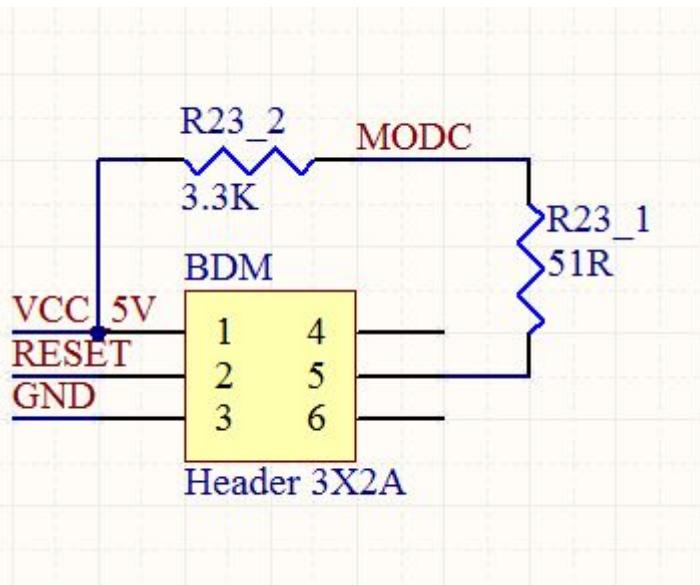
单片机复位电路：



电源电路：



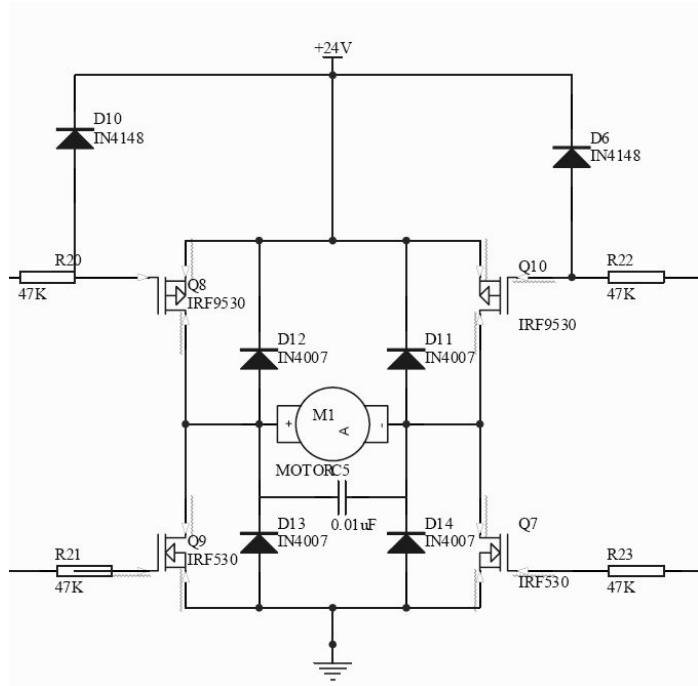
BDM调试电路：



3电机控制

单片机输出的PWM波不能直接驱动直流电机，需要通过电机驱动电路来间接地驱动电机。直流电机驱动电路使用最广泛的就是H形全桥式电路。这种驱动电路可以很方便的实现直流电机的四象限运行，分别对应正转，正转制动，反转，反转制动。

H形全桥式电路原理图如下：



原理分析：

1) 图中当Q8, Q7导通时，电机电流从左端流入，右端流出，此时电机为顺时针转动，当Q9,Q11导通时，电流从右端流入，左端流出，此时电机为逆时针转动。Q8, Q9 (Q11,Q14) 不能同时导通（此时无负载，类似于短路，可能烧坏MOS管）。

2) 电路图中几个二极管都是续流二极管，作用是保护直流电机内部的三极管。因为如果要把电机反转，那么电机可能会产生一个高于电源的反向电动势，可能会损坏内部的三极管。

3增强MOS管的导通和截止是受其栅极G和源极S电压差影响，因此，只需要控制四个MOS管的栅极电压就可以控制电机的运转。栅极电压可以通过芯片来控制。

4) 电机两端并联一只小电容C5，是为了降低电机中感性元件产生的尖峰电压。



我们找到了一款集成芯片BTS7970

特性如下：

输入电压：6V---24V

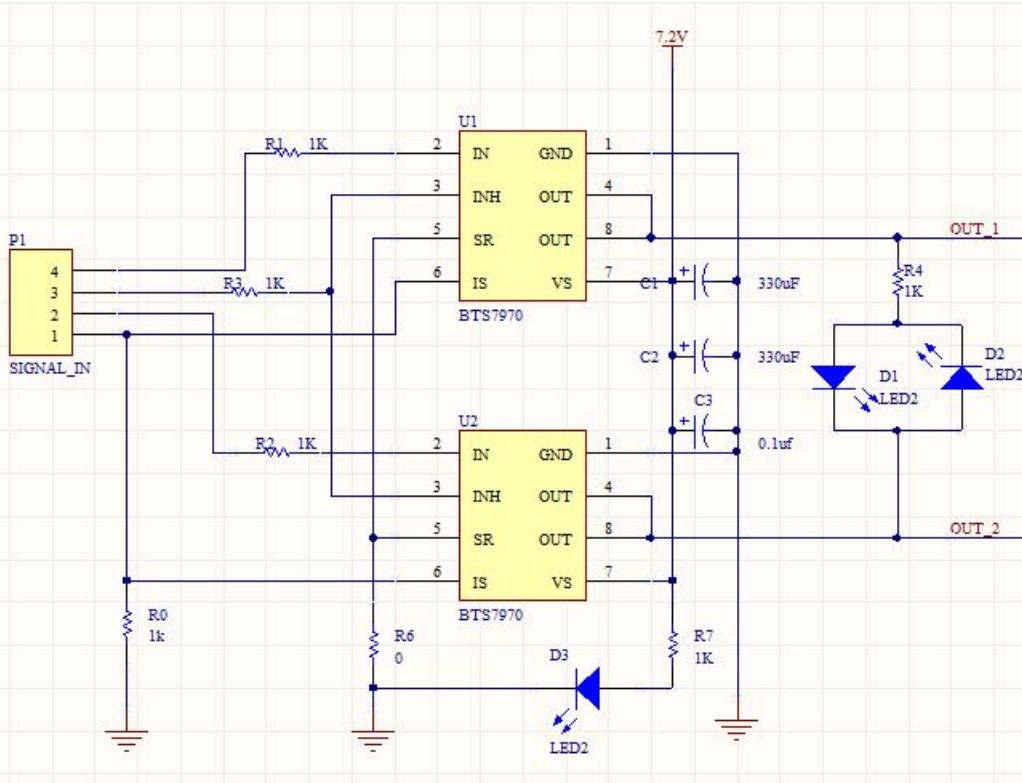
输出电流：最大可达40A

内阻：16毫欧

控制线电压：5V

PWM控制频率：25K

电路原理图如下：

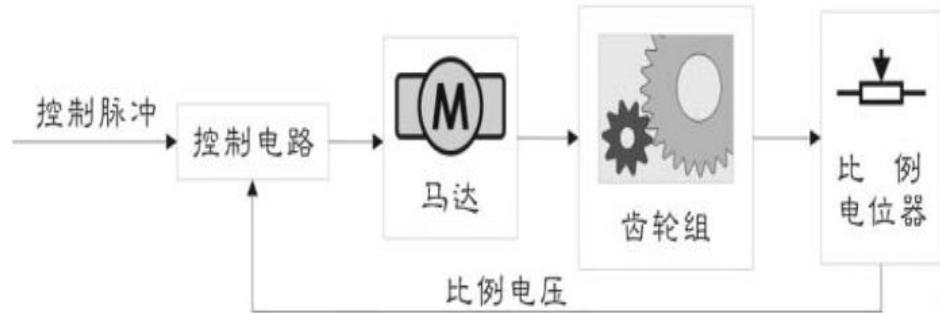


4舵机控制



工作原理：

三根控制线中，分别接电源、地及控制。电源及地为舵机提供能源。当控制脉冲输入后带动马达转动，马达进一步带动齿轮组转动实现减速同时加大输出扭力的功能。同时输出端带动线性比例电位器检测位置，即把转角坐标转换为比例电压并反馈给控制板，控制板将反馈电压与输入脉冲比较，得到的电压差产生纠正脉冲控制马达的转动方向。如此调节直至纠正脉冲趋于零，此时齿轮组输出位置达到期望。



舵机工作原理

2) 对舵机的控制：

舵机的控制信号是周期为20ms的PWM信号，利用占空比的变化改变舵机的位置。其脉宽在0.5ms到2.5ms间变化，对应舵机在0°到180°之间线性变化。这样就可以利用单片机时舵机转到任意角度

输入正脉冲宽度（周期为20ms）	伺服马达输出臂位置
	≈ -90°
	≈ -45°
	≈ 0°
	≈ 45°
	≈ 90°

PWM信号正脉冲宽度与输出位置关系

5图像采集模块

摄像头按照信号类型可分为模拟摄像头、数字摄像头两种。由于单片机普通IO口只能读取

数字信号，故对于模拟摄像头要设计模数转换（ATD）。数字摄像头数据可以直接进单片机。用模拟摄像头的缺点是要自己设计模数转换电路，同步信号分离电路。

摄像头按照传感器，可分为CCD、CMOS。CCD成像质量好，贵。CMOS成像质量略差，便宜。

考虑到单片机处理速度较PC很低，我们采用带有USB接口的摄像机，将摄像机采集的图像信号传输到PC进行处理后，传给单片机，来识别路标和道路等各种图像信息。

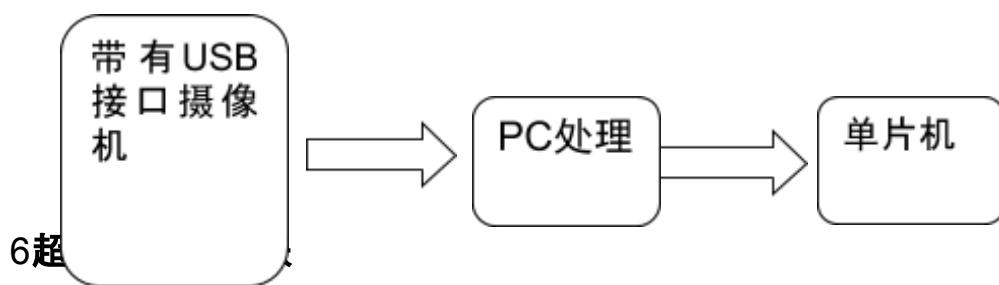


带有USB接口的摄像机

特点：

- 高速USB2.0接口，可达480Mb/s
- 支持静态的图像捕捉（JPG、BMP）
- 图像色彩丰富、逼真
- 支持Windows XP和7 操作系统
- 丰富的外部同步控制资源

图像采集和控制流程：



由于超声波指向性强，能量消耗缓慢，在介质中传播的距离较远，因而超声波经常用于距离的测量，如测距仪和物位测量仪等都可以通过超声波来实现。利用超声波检测往往比较迅速、方便、计算简单、易于做到实时控制，并且在测量精度方面能达到工业实用的要求，因此在移动机器人研制上也得到了广泛的应用。

1) 超声波发生器

为了研究和利用超声波，人们已经设计和制成了许多超声波发生器。总体上讲，超声波发生器可以分为两大类：一类是用电气方式产生超声波，一类是用机械方式产生超声波。电

气方式包括压电型、磁致伸缩型和电动型等；机械方式有加尔统笛、液哨和气流旋笛等。它们所产生的超声波的频率、功率和声波特性各不相同，因而用途也各不相同。目前较为常用的是压电式超声波发生器。

2) 压电式超声波发生器原理

压电式超声波发生器实际上是利用压电晶体的谐振来工作的。超声波发生器内部结构如图1所示，它有两个压电晶片和一个共振板。当它的两极外加脉冲信号，其频率等于压电晶片的固有振荡频率时，压电晶片将会发生共振，并带动共振板振动，便产生超声波。反之，如果两电极间未外加电压，当共振板接收到超声波时，将压迫压电晶片作振动，将机械能转换为电信号，这时它就成为超声波接收器了。



超声波测距模块

3) 超声波测距原理

超声波发射器向某一方向发射超声波，在发射时刻的同时开始计时，超声波在空气中传播，途中碰到障碍物就立即返回来，超声波接收器收到反射波就立即停止计时。超声波在空气中的传播速度为 340m/s ，根据计时器记录的时间 t ，就可以计算出发射点距障碍物的距离(s)，即： $s=340t/2$ 。这就是所谓的时间差测距法。

超声波测距的原理是利用超声波在空气中的传播速度为已知，测量声波在发射后遇到障碍物反射回来的时间，根据发射和接收的时间差计算出发射点到障碍物的实际距离。由此可见，超声波测距原理与雷达原理是一样的。

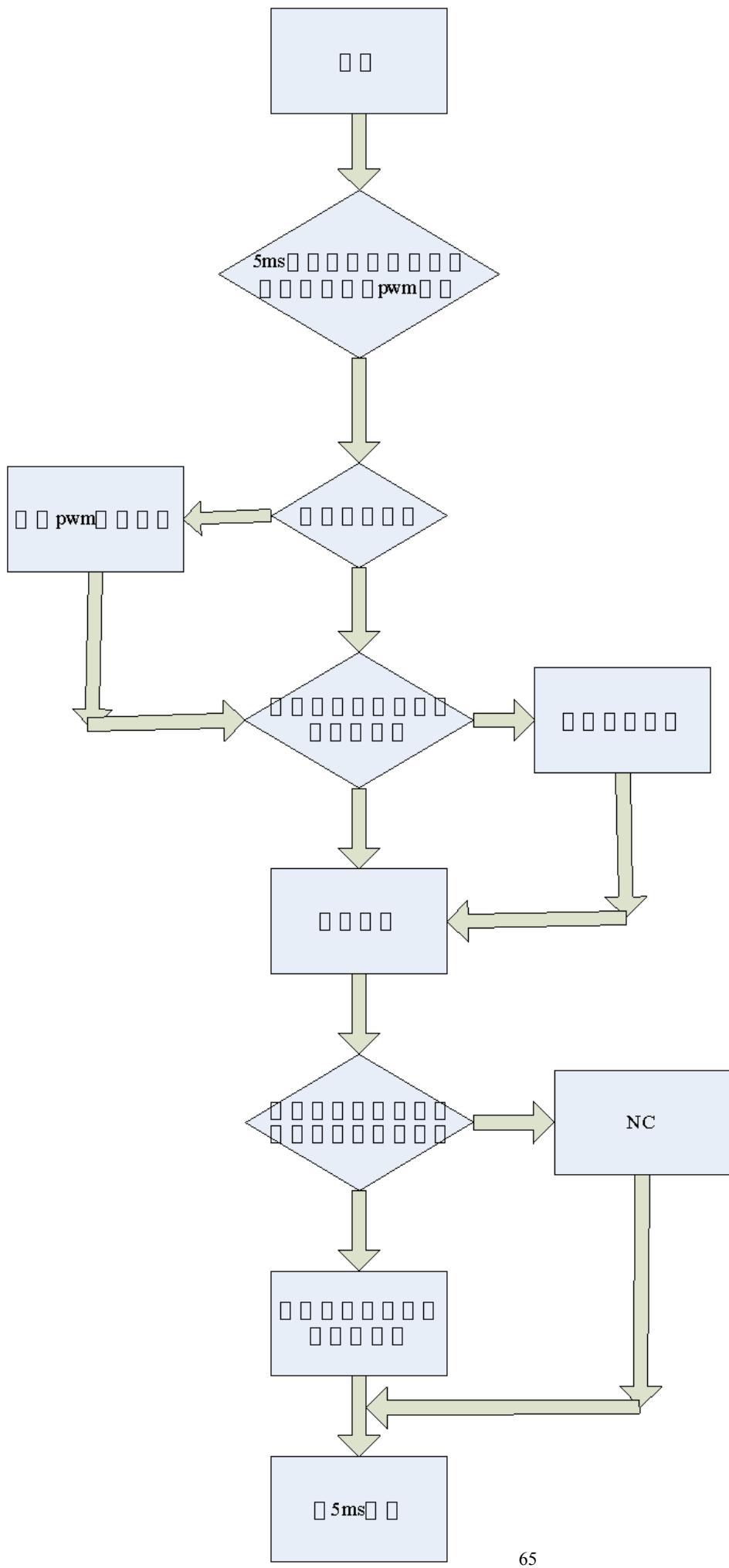
测距的公式表示为： $L=C\times T$

式中 L 为测量的距离长度； C 为超声波在空气中的传播速度； T 为测量距离传播的时间差(T 为发射到接收时间数值的一半)。

程序部分

由于我们做过飞思卡尔智能车，对于freescale xs128芯片比较熟悉，加之它功能丰富，对外设要求较低，性能稳定，故采用之。

下图为基本程序控制流程图，其中编程思想是以中断的形式进行。在一个5ms中断里，分别检测各传感器的返回数值并进行电机舵机的控制。



下面贴出xs128的部分模块代码：

设置锁相环（超频）

```
void PLL_Init(void) {
    CLKSEL_PLLSEL      = 0;
    PLLCTL_PLLON      = 0;
    SYNR_VCOFRQ       = 0b11;
    SYNR_SYNDIV       = 4;
    REFDV_REFFRQ      = 0b11;
    REFDV_REFDIV       = 0;
    PLLCTL_PLLON      = 1;
    _asm nop
    _asm nop
    _asm nop
    _asm nop
    while(!CRGFLG_LOCK);
    CLKSEL_PLLSEL      = 1;
}
```

用于电机和舵机控制的PWM模块

```
void PWM_Init(void)
{
    PWME                = 0x00;
    PWMPOL              = 0xFF;
    PWMCAE              = 0x00;
    PWMPRCLK             = 0x22;
    PWMSCLA              = 10;
    PWMSCLB              = 50;
    PWMCTL_CON01         = 1;
    PWMCLK_PCLK1          = 1;
    PWMPER01              = 10000;
    PWMMDTY01             = DIRECTION_MIDDLE;
    PWME                = 0xFF;
}
```

用于定时的PIT模块

```
void TIM_Init(void) {
    TIOS_IOS0 = 0;
    TCTL4_EDG0B = 1;
    TCTL4_EDG0A = 0;
    TIE_C0I = 1;
    TIOS_IOS1 = 0;
    TCTL4_EDG1B = 0;
    TCTL4_EDG1A = 1;
    TIE_C1I = 1;
    TIE_C1I = 0;
```

```
TSCR1_TEN      =      1;
```

光电编码器方波计数代码

```
PACTL_PAEN    =      1;          // 脉冲累加器使能
PACTL_PAMOD   =      0;          // 设为事件计数方式
PACTL_PEDGE   =      1;          // 捕获上升沿
PACNT         = 0x0000;
```

中断写法

```
#pragma CODE_SEG NON_BANKED
void interrupt VectorNumber_Vtimch0 vsync(void)
{
//代码
    TFLG1_C0F      =      1;
}
AD转换模块
ATDCTL1 = 0x3F;
ATDCTL3 = 0x88;
ATDCTL4 = 0xE1;
uint ADC_GetValue(byte ch)  {
    ATDCTL5 = 0x0F & ch;
    while(!ATDSTAT0_SCF);  Return(ATDDR0);
}
```