

技术报告

夏令营-高中二组

机器人设计需求

第一阶段目标：在步兵车基础上设置抓取物块结构，将场地上障碍块的子弹倒入步兵车弹仓。

第二阶段目标：在第一阶段的基础上将取得的子弹通过发射机构打出去，配合相应战术，与盟友一起 2v2 对战。

对战战术：

在规定时间内，夹取一定量的子弹并完成对对方战车的有效攻击。

正式赛 vs14 联盟：

第一局

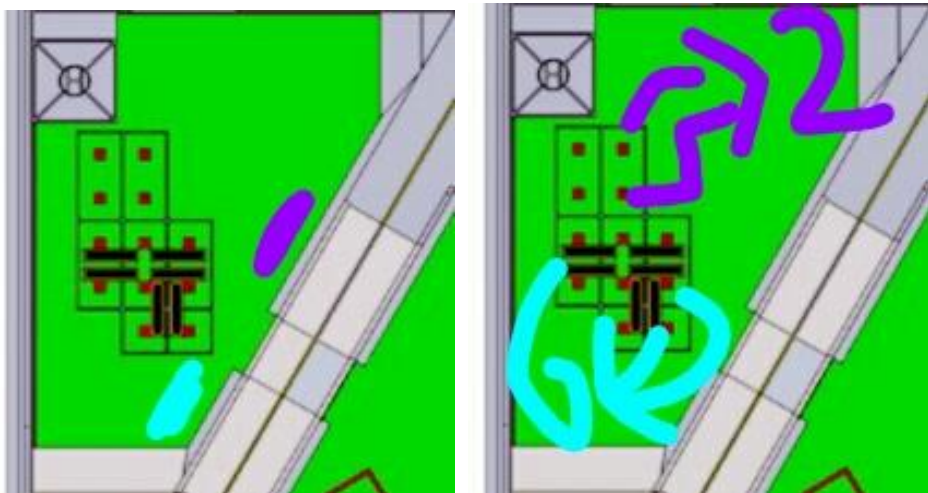
主要战略思想：

防守，引诱对方进攻，打伏击。

保持冷静，快速但稳健地攻击，保持战场主动权在我方手中。

具体战术：

入桥洞时，2 组先进，枪口朝洞内。6 组后进，枪口摄像头朝外倒退进入桥洞。入洞后，二组绕行取弹，六组直接取门口子弹，同时观察桥洞外情况。如有危险，则六组就近蹲守桥洞左，二组蹲守桥洞右，如图：



(蓝色为六组，紫色为二组)

取完两个障碍快内子弹后，二组继续绕行，同时改观察门口，六组取子弹。如图：(紫色为二组路径，蓝色为六组路径，若二组途中发现危险，则二组去紫色位置蹲守，六组去蓝色位置蹲守)

到达蹲守位置后不要妄动，等待一组/四组进入，抓紧攻击进入车辆的左右装甲板，力求打死。若打不死，进攻车辆极可能向六组方向绕，六组需挡住四组，避免拉开距离。

最好的情况是打死四组，2v1，若打不死，则 2v2, 最坏 1v2

第二局和第三局，第一局赢，则守在洞内，取弹求稳，若输，则保持冷静，千万不能焦躁。



正式赛 vs35 联盟

具体战术：

先取两个障碍块内子弹，（同 14 战略）取完后马上到对方桥洞，先朝里射弹试探，发现有击打对象则抓紧时间攻击，如果没有击打对象，则守在洞口，等待对方进攻。

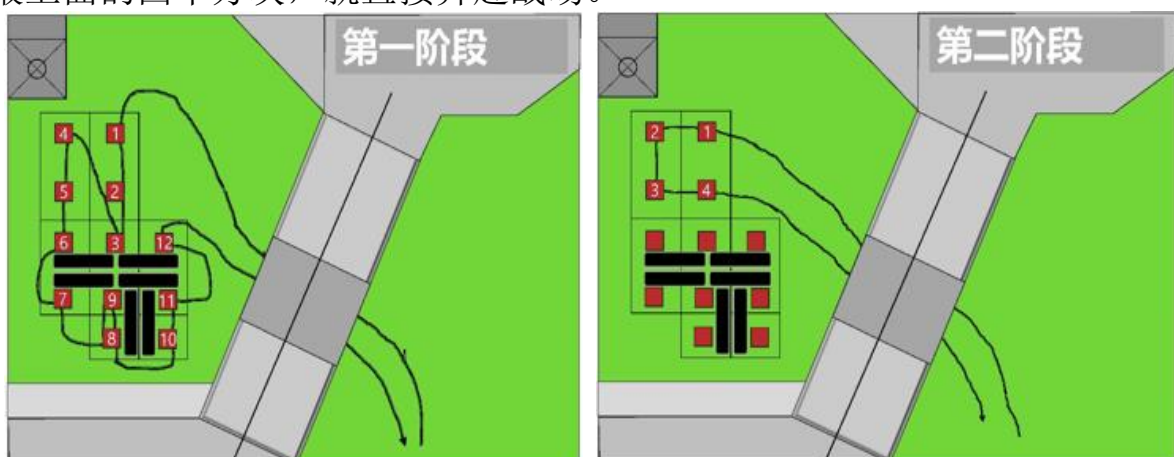
总体方案

经过多次讨论以后，我们组决定使用机械臂抓取前方抬升的方法来完成此次任务。因为如果是前方抬升的话，将会大大降低使用传感器或是使用可旋转视野等方式所需要的时间，因为在视野中就可以很清楚的看出爪子抓的是否准确和实时监控，也能更加有效的处理复杂的应急情况。由于这个原因，所以我们的车子并无巡线的必要和变成全自动的可能。

抓取的路线有两种，分别适用于第一阶段和第二阶段。

第一阶段是根据哪个地方的方块最适合我们的机器夹的原则进行分布的，所以我们先跳过车子容易卡住的减速带区，直上去取最容易拿的六个，再以逆时针的顺序依次夹取。

第二阶段因为考虑到敌方机器人可能采取快攻的方式来打，所以我们组先去取最上面的四个方块，就直接奔赴战场。



动力选择（夹子）

在我们最初的方案设计当中，我们打算给抓取物块的夹子使用舵机进行操控。然而在实际使用中我们发现，由于舵机相对较重，会对抬升机械臂的电机造成更大压力，并且对于舵机的操控我们也没有非常多的经验，我们最终放弃使用电机，改为使用气缸。气缸的优势对于我们的只需要松开和夹紧的夹子是非常明显的：气缸相对容易控制，不需要像电机那样确定一个角度；气管、气缸和电磁阀总重量比舵机相对更轻。因此我们最终确定了使用 125 量程的气缸来操控夹子。

爪子迭代：先是连杆结构，然后是钢轴导轨直夹，最后是工业导轨直夹。整件夹子高度不断降低，夹取效率不断提升。

机械臂

对于机械臂的动力来源，最初我们也是考虑使用舵机操控。但在通电测试前，我们发现，夹子夹了障碍块以后有高达 0.8kg 的质量，再加上有 35cm 的阻力臂长度，对电机的扭矩和拉力是有非常大的要求的。因此，我们特地使用了含减速齿轮组的 1:27 3510 电机来带动机械臂的转动。

机械部分：

综述：在项目开始前，我们先拿到最重要的步兵车. 对所有比赛任务的设计也都是基于它之上的. 在最开始的时候，大家一起讨论规则得到两个方案：

- 1、通过涵道或者其他动力足够强的发动机来从障碍块中吸取子弹小球进入弹仓为第二阶段的步兵车战斗提供战斗力，但是在接下来的设计和操作过程中出现了会使车辆尺寸不符，而且会令步兵车上的物件过于冗杂的问题而被淘汰；
- 2、是通过机械臂上的机械爪来抓取障碍块并运转电机调整角度将障碍块中的子弹倒入弹仓；

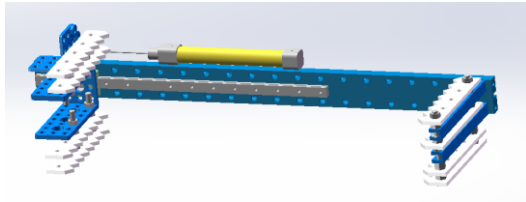
小组全员最后同意了第二方案，但在方案中令如何电机运转并带动机械爪出现两种想法，一种是想在云台与底座连接的空隙中插入一根细棍且在两端安装法兰或滚轴与电机合为一体，在其中再加入机械臂；一种是最终使用的想法，将电机同样固定，但去掉细棍直接通过机械爪两端来连接固定。

电机的固定在加工过程中是一个难点，本来大家计划通过一个合适的平台来固定，但突然就产生了干涉这样的巨大问题. 多亏在赛前发现了问题，我提供了将平台减小，垫高电机的固定方法，最后被淘汰了，换成了通过两个小的型材来固定，效果是相当不错的. 为了配合我组“倒子弹”的方法，我组的弹仓自然也要设计的能够最好的接收子弹，最先的版本是我制作的纸板弹仓，在原有的金属弹仓之上黏结了可配合的斜面纸板，尽可能的多接收子弹，最后的成品将纸板换成了亚克力板并且给出两个可以放置机械臂的凹槽. 步兵车经过大家的修修改改最终有了现在的成绩.

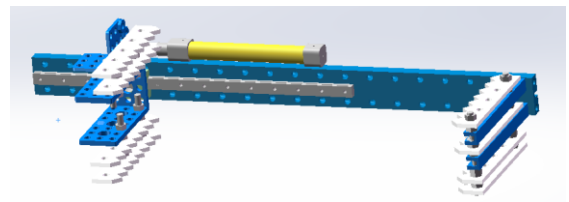
各模块介绍：

我们所设计的机械臂主要分为两个部分还有加上支撑结构。

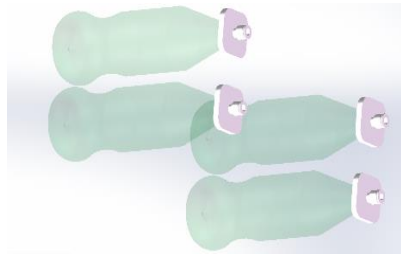
第一部分是障碍块的夹取，是由气动的气缸来控制伸缩夹取，还有一个重要结构是滑道和滑块的使用大大提升了夹取速度。气动的优点非常明显，就是夹取的速度特别快，而且非常紧，可以很好的保证有效的进入下一阶段的抬升，但与此同时也有个弱点就是气动装置对气密性的要求很大，所以很容易就因为疏忽就将气漏完了。



夹子开

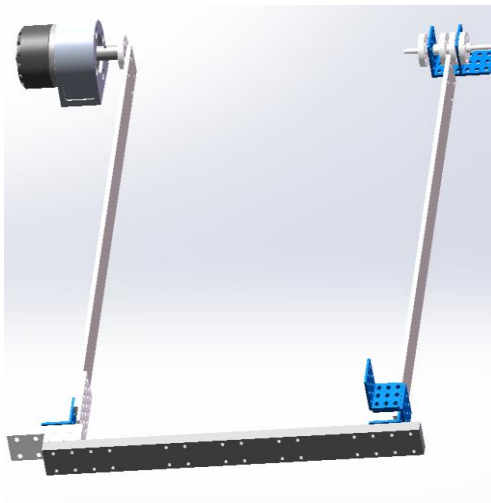


夹子关

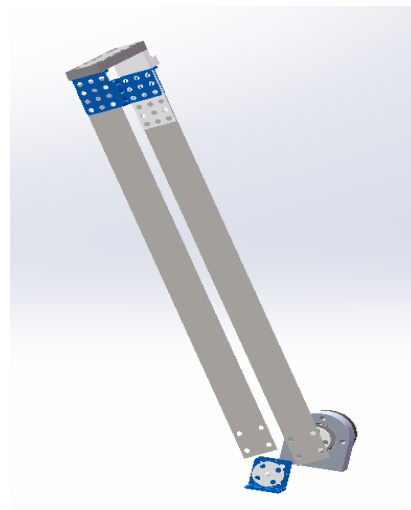


气瓶

第二部分就是抬升结构，我们组因为考虑了电机之间是否同步的问题，所以只采用了单电机的方案，使用的是 RM3510，经过计算它转动所产生的扭矩是足以带动整个箱子的抬升旋转的。

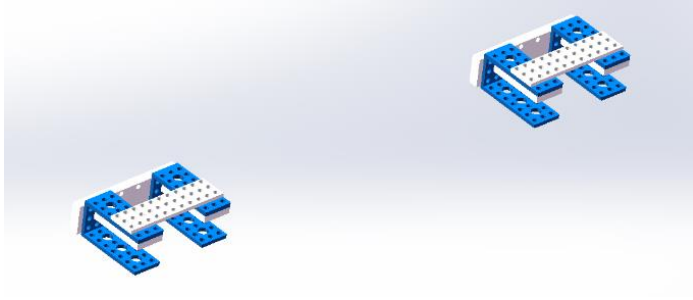


架子降下时

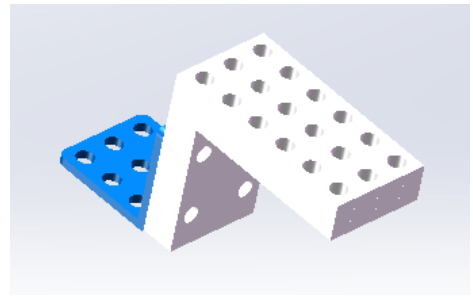


架子抬升时

然后就是支持支撑部分，左右各有两个架子支撑架，还有两个微型限位，保证了机械臂不会因为转动过快而直接撞地导致夹子的损坏。

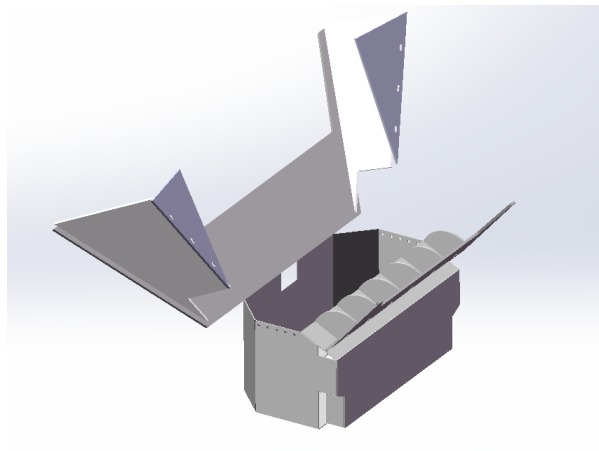


架子支撑架



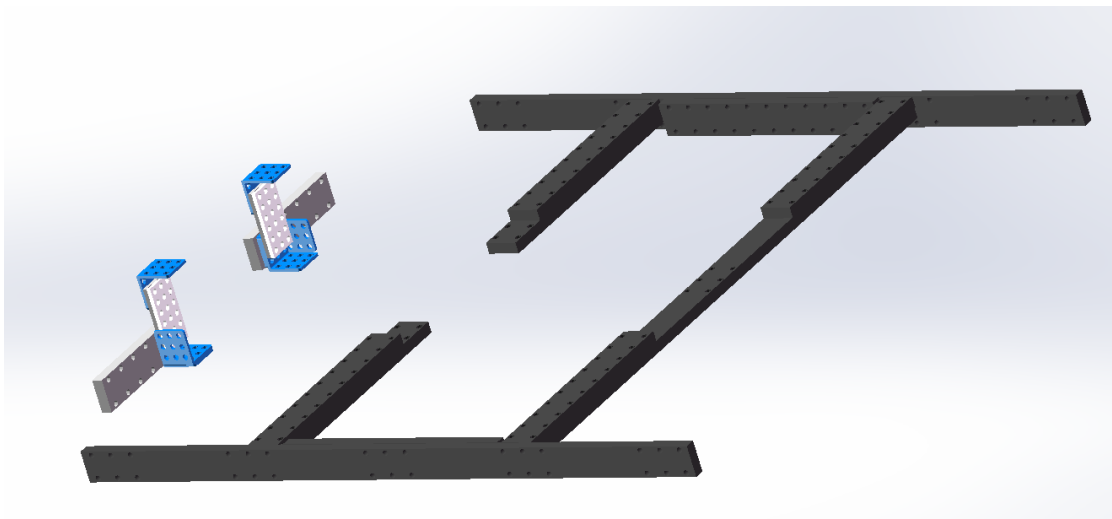
微型限位

因为我们抬升结构的问题，所以我们的弹槽也进行了改进。为保证检录合格，初始状态把机械臂向上搭载在弹仓（扩展结构）上，比赛开始后启动电机将其旋下。



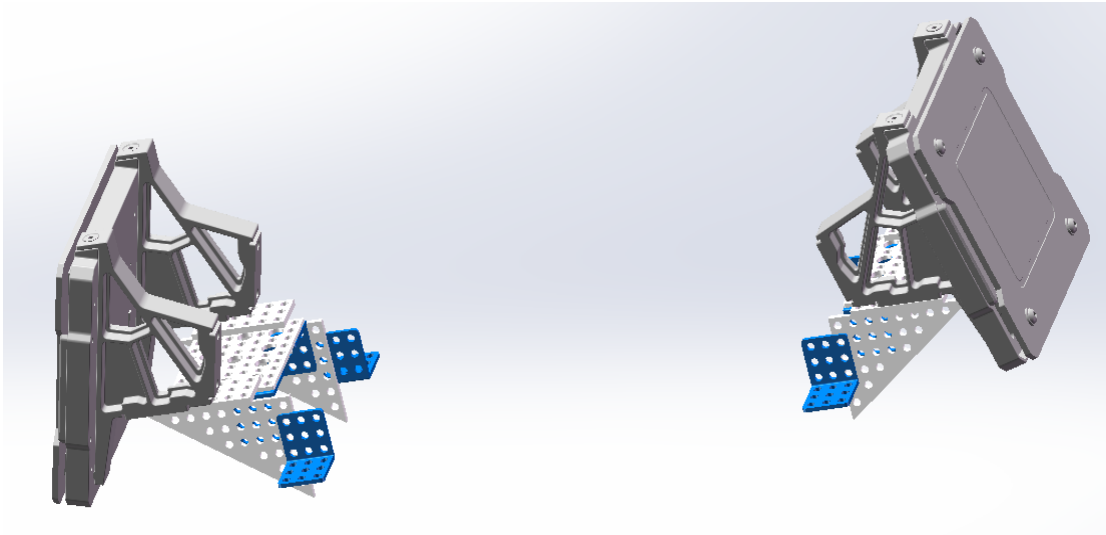
弹槽

防护结构看似不重要也似乎没有技术含量，但是在实战中却发挥了很大的作用。

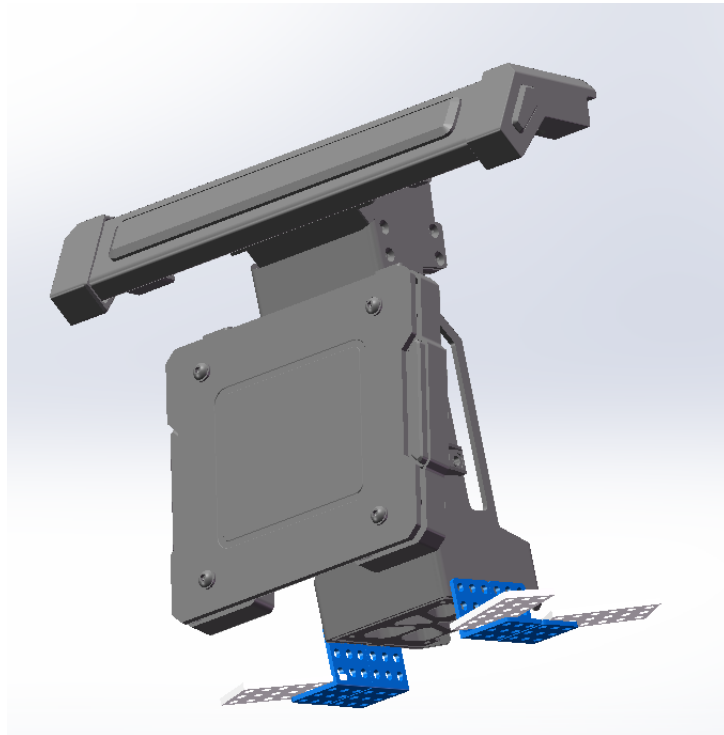


装甲区域

我们也对左右和后面的装甲板作出升级，以适用上方机械臂的伸展和弹仓的扩大。



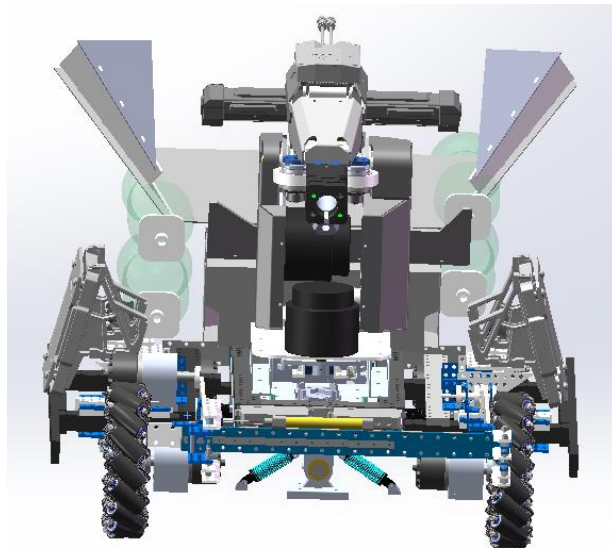
左右装甲板和支架



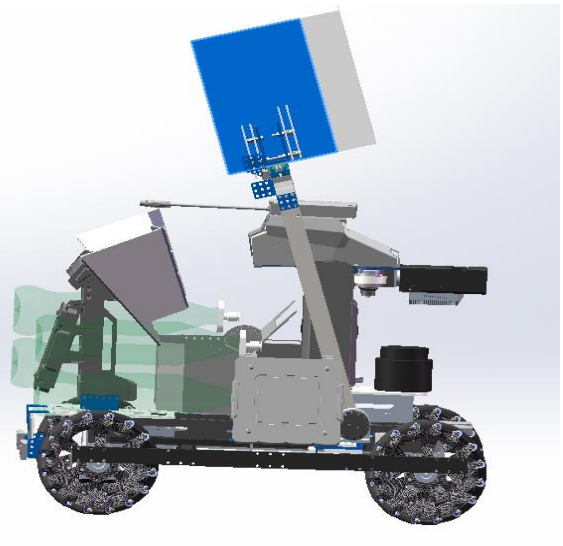
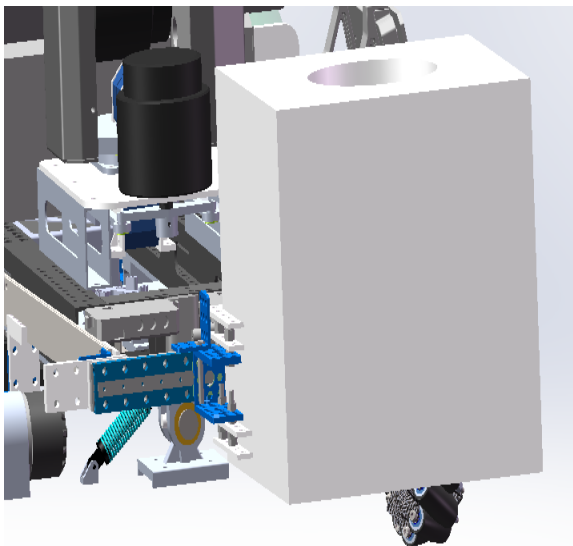
后装甲板拓展结构

我们的战车采用单机械臂夹取结构，将一个由两条臂制成的抓手置于战车正前方，用抓手夹取障碍块再向上倾倒是使障碍块内子弹落入弹仓。

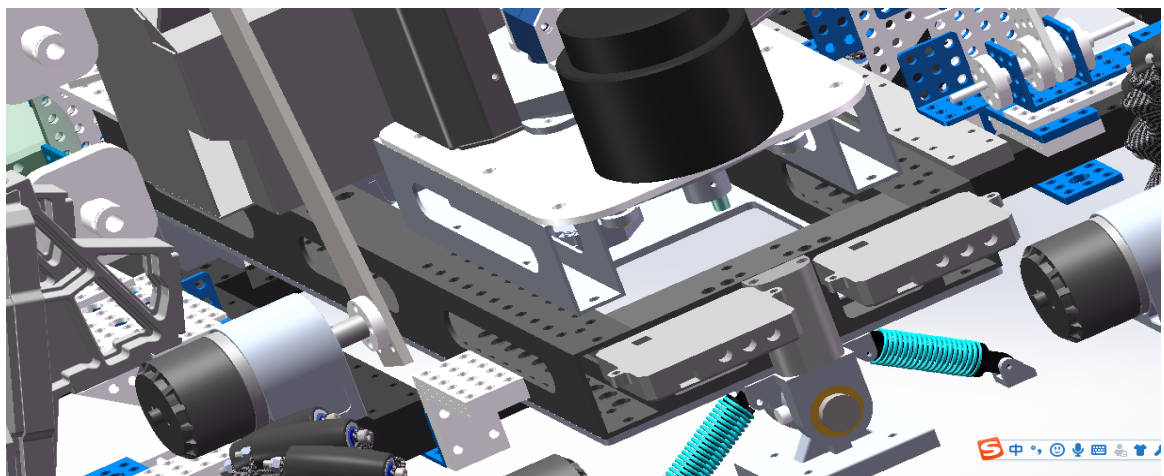
首先对准目标，闭合机械爪；然后启动电机将障碍块向上倾倒是子弹，反转放下障碍块，张开机械爪；抓取下一个目标障碍块。机械臂结构主要的干涉集中在弹仓、云台下方及前轮上方。于是我们在底盘两侧上方用型材的保险杠结构基础上搭建了安置机械臂电机和底座的平台；之后为了防止与前轮和车体干涉，在平台前安置了限位，在弹仓扩展结构上设置了开口。首先整个机械结构选用的是型材以保证强度，后将机械臂部分改为碳纤维板材以减重，提高电机效率。



抓之前



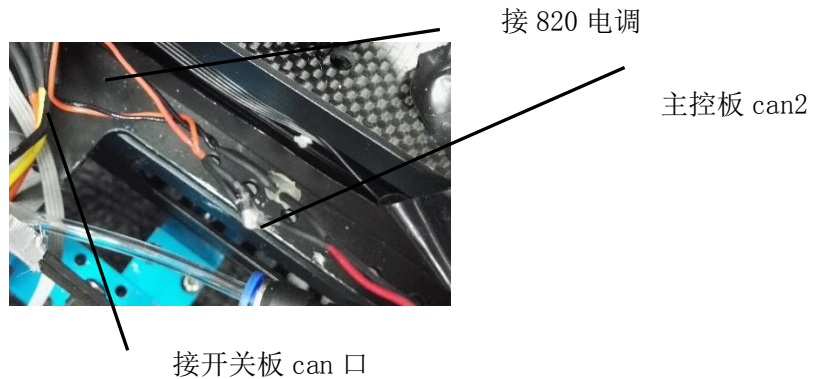
夹取方块及拾取能量小弹丸



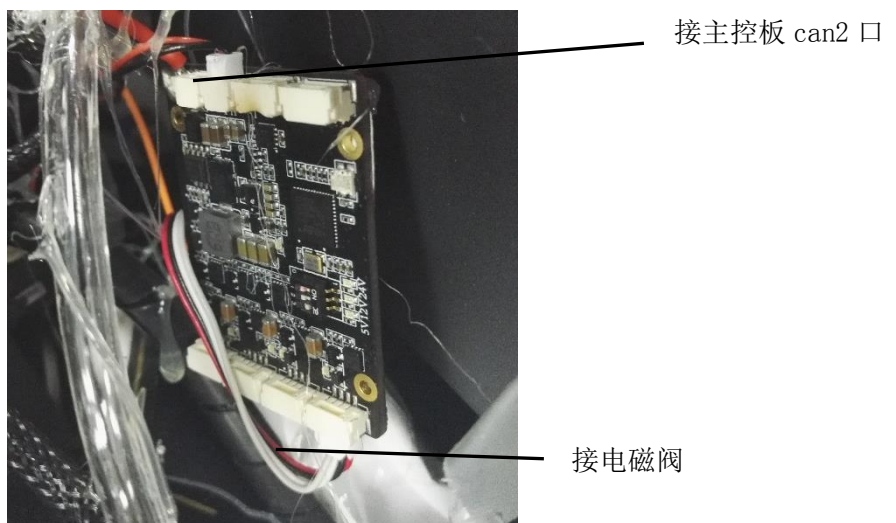
倒计时电机情况

硬件

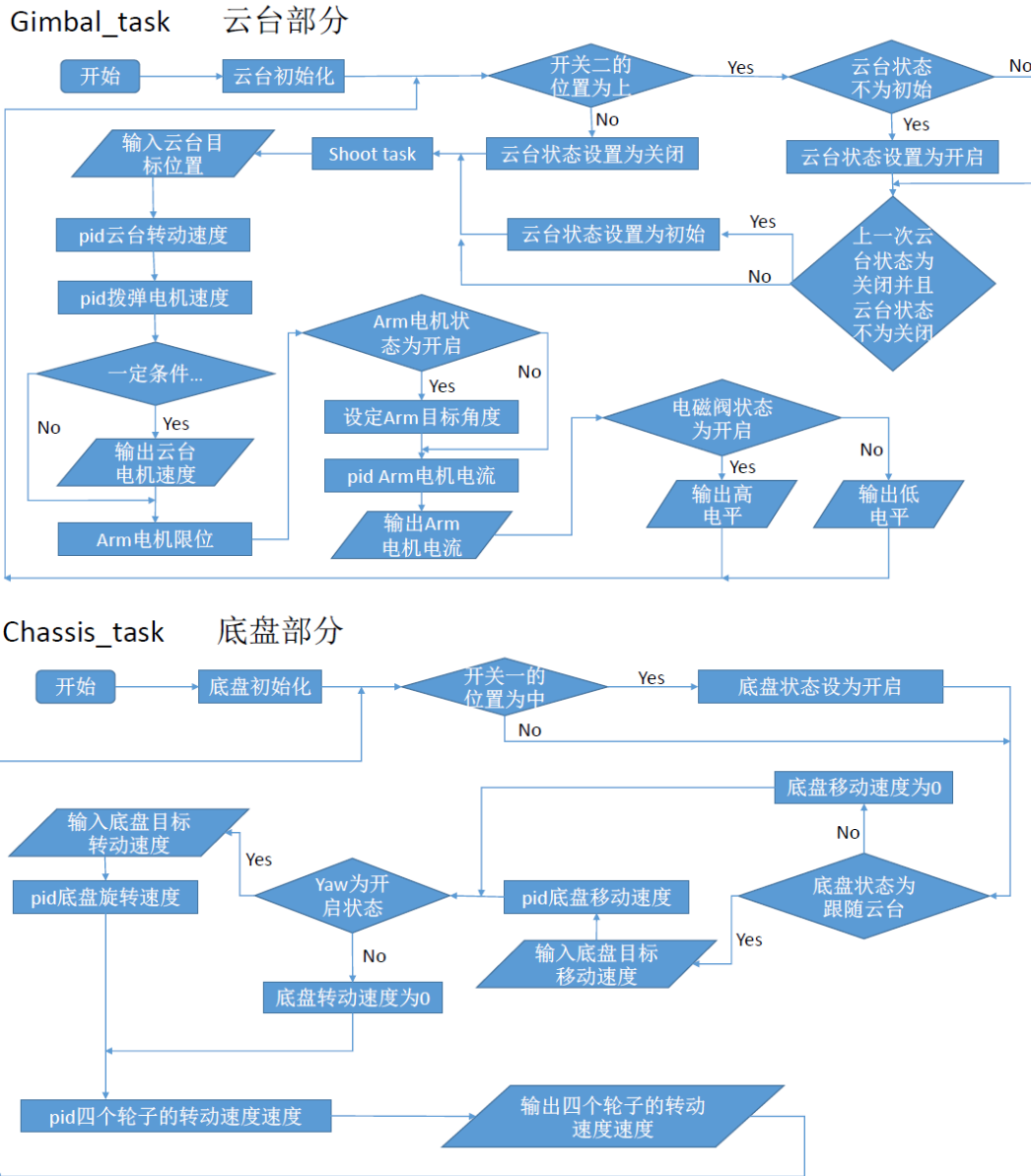
硬件这方面主要由许胤哲和曾柏霖负责，例如 3510 电机电调的使用，开关板的接法以及电磁阀线路的连接。李逸斌主要负责了线路的安装和焊线，包括 4pin 线焊接以及开关板电源的焊接。在具体对硬件操作过程中，我们也碰到了一些问题。比如，我们希望用 can 接口控制机械臂的 3510 电机。3510 的电机在使用时必须连接 820 电调，然后再通过 820 电调的 can 线连接主控板的 can 口。然而主控板只剩下 can2 一路可用，我们又必须在这个 can 口连接开关板，因此最初的想法是将 820 电调的线直接连在开关板的 can 口上。但是在实际操作中却发现，电机换成连接开关板 can 口无论如何也控制不了，然而直接连主控板 can2 却可以。在多次咨询大学生后我们了解到，开关板的 can 口很少用于再发送 can 信号，因此机械臂电机难以接收到发自主控板的 can 指令。可行的办法是利用导线的并联方法，将 can2 并联出去。在此基础上，我们将连接开关板的 can 线和连接 820 电调的 can 线焊在了一次，最后由一根总线连到了主控板 can2（如下图）。



这是我们主控板最后接的情况。唯一我们自己多加的是 can2 口的线。（如图）这是开关板的接线情况。上层 2pin 的是开关板连主控板的 can 线，下面的 4pin 线则是控制电磁阀的线。



软件部分：流程图



本次活动中，开源代码我们未做过多的更改，但在原有基础上我们根据实际的需求做了部分功能的增加。

增加部分为投入实际应用的 Arm 电机的控制、电磁阀的控制和 can 通讯；作为实验的云台、底盘分离控制，pwm 通讯。

最终版本增加的代码如下：

```
gimbal_task.c
void gimbal_task(const void* argu)
{
    if(rc.kb.bit.C == 1||rc.kb.bit.V == 1||rc.kb.bit.B == 1||rc.kb.bit.X == 1)
        target = target-0.2f;
    else if(rc.kb.bit.F == 1)
```

```

        target = target+0.2f;

if(target>=3)
    target=-3;

if(target<=-98&&rc.kb.bit.C == 1)
    target=-98;
if(target<=-93&&rc.kb.bit.V == 1)
    target=-93;
if(target<=-88&&rc.kb.bit.B == 1)
    target=-88;
if(target<=-125&&rc.kb.bit.X==1)
    target=-125;

pid_calc(&pid_arm, moto_arm[0].total_angle/27, target );
tmp_arm_iq = pid_calc(&pid_arm_speed, moto_arm[0].speed_rpm, pid_arm.pos_out);
can_send_arm_iq(tmp_arm_iq);

if(rc.kb.bit.Q == 1)
    can_send_valve_iq(1, 0, 0, 0);
else if(rc.kb.bit.E == 1)
    can_send_valve_iq(0, 0, 0, 0);
}

void can_send_arm_iq(s16 arm_iq)
{
    ARM_CAN.pTxMsg->StdId    = 0x200;
    ARM_CAN.pTxMsg->IDE      = CAN_ID_STD;
    ARM_CAN.pTxMsg->RTR      = CAN_RTR_DATA;
    ARM_CAN.pTxMsg->DLC      = 8;
    ARM_CAN.pTxMsg->Data[0]  = arm_iq >> 8;
    ARM_CAN.pTxMsg->Data[1]  = arm_iq;
    ARM_CAN.pTxMsg->Data[2]  = 0;
    ARM_CAN.pTxMsg->Data[3]  = 0;
    ARM_CAN.pTxMsg->Data[4]  = 0;
    ARM_CAN.pTxMsg->Data[5]  = 0;
    ARM_CAN.pTxMsg->Data[6]  = 0;
    ARM_CAN.pTxMsg->Data[7]  = 0;
    HAL_CAN_Transmit (&ARM_CAN, 1000);
}

```

作为实验的代码如下（未应用在最终版本程序中）：

```

gimbal_task.c
void MX_TIM3_Init(void)

```

```

{

TIM_MasterConfigTypeDef sMasterConfig;
TIM_OC_InitTypeDef      sConfigOC;

htim3.Instance          = TIM3;
htim3.Init.Prescaler    = 179;
htim3.Init.CounterMode  = TIM_COUNTERMODE_UP;
htim3.Init.Period       = 0;//0
htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;

if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
    Error_Handler();

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode     = TIM_MASTERSLAVEMODE_DISABLE;

if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
    Error_Handler();

sConfigOC.OCMode      = TIM_OCMODE_PWM1;
sConfigOC.Pulse       = 0;//0
sConfigOC.OCpolarity  = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode  = TIM_OCFAST_DISABLE;

if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
    Error_Handler();

if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_2) != HAL_OK)
    Error_Handler();

HAL_TIM_MspPostInit(&htim3);
}

```

chassis_task.c

```

void chassis_task(void const* argu)
{
    if(rc.kb.bit.R == 1 && on==0)on = 1;
    else if(rc.kb.bit.R == 0 && on == 1)on = 2;
    else if(rc.kb.bit.R == 1 && on == 2)on = 3;
    else if(rc.kb.bit.R == 0 && on == 3)on = 0;

    if(on == 2)

```

```
mecanum_calc(chassis.vx, chassis.vy, 0, MAX_WHEEL_SPEED, chassis.wheel_speed);  
else  
mecanum_calc(chassis.vx, chassis.vy, chassis.vw, MAX_WHEEL_SPEED, chassis.wheel_speed);  
}
```

Pwm 通讯：原本打算用 Pwm 实现对电磁阀的控制，但因技术上水平有限，存在过多的问题无法解决，所以否定的这个方案，改用 can 对电磁阀进行控制。

云台、底盘分离控制：原本想用此功能实现机器人对抗时更好的控制走位，实际上机器人的水平设计范围仅为 60° 左右；而且安装了麦轮的机器人可以实现全向移动，再次分离控制反而增大了机器人的操控难度，所以否定了此方案。

步兵车机械臂软件研究：（曾柏霖）

我们步兵车抓取子弹的动作由一个 3510 电机带动机械臂来完成。

关于如何使机器人准确完成一系列抓弹动作，我们需要解决以下问题：

- 1、如何通过开关控制输出模块气动电磁阀以达到控制活塞夹住箱子的目的。
- 2、如何使机械臂准确停在弹仓正上方并将子弹倒入弹仓。
- 3、如何调整 PID 参数使机械臂在运动过程中保持一个相对稳定的速度。

问题（1）：

电磁阀的额定电压为 24V，所以我们一开始直接使用开关控制输出模块的 24V 电路直接控制电磁阀，并且从主板直接向开关控制输出模块提供 24V 的电压。然而电磁阀使用几次之后发现开关控制板 24V 电路被烧坏，在检测开关板电源的电压后发现，主板供给开关板的电压在 25V 左右，并且电压并不稳定。于是我们使用开关板的 5V 电压控制继电器，将主板的电源直接输出到电磁阀上，以继电器来控制电路闭合，最终解决了开关板被烧坏的问题。

问题（2）：

由于机械臂在夹住箱子之后质量较大，惯性较大，机械组也未设计减速齿轮，并且机械臂也只由一个电机驱动，于是使其准确停在弹仓正上方也是我们要解决的问题之一。

对于这个问题，我们最初有以下几种方案：

- 1、向电机输出一个额定的电流并使用开关限位。
- 2、利用 3510 电机的编码器通过负反馈控制来精确控制机械臂。
- 3、利用负反馈控制的同时使用开关限位。

然而（1）与（2）各有缺点：

在（1）中，由于机械臂较长，所以箱子在不同位置阻力臂不一样，所以额定电流会使机械臂在举起箱子时逐渐加速，经过测试，当机械臂的电流大小恰好能把箱子从地面抬起时，到弹仓上方时的速度已达到能把子弹从箱子里甩出

去的速度。

在（2）中，由于负反馈会出现超调现象，并且PID的参数需要通过负载大小来调整，所以一旦机械臂结构有所改动，参数需要重新调整。并且PID有时会有微小的超调现象，容易将子弹甩出。

而在（3）中，开关的限位大幅度解决了PID的超调问题，而负反馈控制又很好的解决了额定电流速度无法稳定的问题，于是我们初步决定用方案（3）。

但在与机械组商讨之后，我们发现车上并没有合适的位置安装限位开关，于是我们只能使用方案（2）。

问题（3）：

在最终决定只是用PID来解决限位问题之后，我们开始调整PID的参数。

首先，我们将速度与位置环的Ki与Kd设为0，调整P值的大小，当调整到抬起速度刚好时，出现稳态误差，于是我们加入积分。此时发现积分反应时间较慢，效果不明显，于是我们调低积分时间，调高积分输出限制。但此时，我们发现增大电流时马达转矩与速度会一起增大，需要限制其速度而增大转矩。于是我们将原程序改为以下代码：

```
if(rc.kb.bit.C == 1||rc.kb.bit.V == 1||rc.kb.bit.B == 1||rc.kb.bit.X == 1)
    target = target-0.2f;
else if(rc.kb.bit.F == 1)
    target = target+0.2f;

if(target>-3)
    target=-3;

if(target<-98&&rc.kb.bit.C == 1)
    target=-98;
if(target<-93&&rc.kb.bit.V == 1)
    target=-93;
if(target<-88&&rc.kb.bit.B == 1)
    target=-88; /*多个角度便于调整*/
if(target<-125&&rc.kb.bit.X==1)
    target=-125; /*将机械臂调整到不影响射击的角度*/

pid_calc(&pid_arm, moto_arm[0].total_angle/27, target );
tmp_arm_iq = pid_calc(&pid_arm_speed, moto_arm[0].speed_rpm, pid_arm.pos_out);
can_send_arm_iq(tmp_arm_iq); /*PID输出*/
/*此段程序包含在云台主程序循环中*/
```

调整成此代码之后，当按住相应键位时，会缓慢增加负反馈的目标角度，由此限制其速度，也可以随时手动调整机械臂的位置，有利于操作手的操纵。

四，项目历程简介及最终评估

7月13日，队伍组建完成。

7月14日至16日，完成车周围防护 demo 的设计与搭建，主要零件 solidworks 作图，熟悉代码，机械臂的设计。

7月17日至18日，完成机械臂的设计与搭建，确定动力选型，主要物资购买。

7月20日至23日，程序编写完成，调试电机与气动装置，整车 solidworks 制图完成。

7月24日，第一阶段比赛

7月25日至27日，优化取弹方案，设计第二阶段比赛战术。

7月28日，第二阶段比赛。

最终实现的功能：

从启动去出发到取到第一个障碍块：约 10s

取到障碍块内 50%及以上的子弹

射击流畅，不卡弹。

机器人总体评估：

swot 分析

s（优势）：

- 1，设计常规，简单，易实现。
- 2，编程简单。
- 3，大部分零件可以用现有材料解决，节约成本。
- 4，图传与抓取的方向一致，方便操作。
- 5，抓取用时短。

w（劣势）：

- 1，机械臂力臂过长，需要扭矩大，电机易损伤。
- 2，气动装置有漏气危险。
- 3，支撑结构复杂，且难固定，易造成干涉
- 4，机器人高度限制，有超高的危险。
- 5，弹仓范围难扩大，接不到子弹。
- 6，pid 控制不稳定。

o（机会）：

- 1，电机改为舵机，增大扭矩。
- 2，使用软质，可压缩的材料，扩大空中接住子弹的面积。
- 3，缩短机械臂力臂长度

- 4, 改用其它控制方式, 增加稳定性。
- 5, 气动改为电机, 增加稳定性。

T (威胁):

- 1, 自动化程度低, 对操作手现场操作要求高。
- 2, 机械结构不稳定, 赛前加固机械压缩检查其它部件的时间。
- 3, 与某些组对比没有时间竞争优势。
- 4, 没有应对突发事件的备选方案。

创新点:

- 1, 由于机械臂升起时被弹仓挡板挡住, 机械臂可能超高, 解决方案是使用扫把毛做了一个挡板, 机械臂放下时可以分开扫把毛, 降低高度, 但扫把毛足够密, 子弹不会从缝隙里漏出去。缺点是制作麻烦。
- 2, 采用透明胶带粘成一个类似塑料膜的挡板, 可以被压低, 压低后可以恢复原来高度, 挡住子弹。缺点是当时没有固定好, 塑料膜压低时改变了角度, 导致子弹挡板的效果不好。

关于团队合作:

我认为团队合作是分层级的, 以队长为圆心, 第一层是核心成员, 这类成员通常技术实力强, 认真负责, 明白自己要完成的事和他人要完成的事, 这类成员通常工作主动, 经常提供新的创意和设计, 团队里的核心成员不超过 4 个。

核心成员的外层是普通成员, 这类成员技术实力较弱, 可以完成核心成员交派的任务, 但不能主动意识和规划未来的工作。当核心成员与普通成员交流不善, 或者项目整体时间安排没有做好, 普通成员容易陷入停摆状态, 造成团队效率低下。

普通成员的外层是游离成员, 这类成员常表现出对任务的不上心, 不负责, 团队有需要时找不到人, 自己负责的任务敷衍了事, 推卸责任, 这类成员会被团队边缘化, 甚至默认为不属于团队。

团队合作的专业化水平要提高, 对于一个具体的任务, 参加完成任务的人应该不超过 4 人, 4 人以上容易出现干扰。如果人数较多, 可以考虑采用轮班制度, 但一定要做好两班人员之间的沟通交流, 专人负责的部分, 应该由专人负责到底, 中间最好不要换人。

团队合作沟通很重要, 核心成员要向非核心成员解释清楚下一步具体的技术任务, 项目管理要向每个方面的核心成员确认进度, 队长或者队内外交负责人要表示清楚'整个团队的需求, 主动寻找帮助和解决方案。

个人总结

林卓垠：参加这次 RM 夏令营，我认识到了我自身技术经验的不足，同时也发现了我自己的能力。我在大家身上学到了很多，也基本确定了我未来学习的方向。这次夏令营锻炼了我管理时间，管理物资和与人交际的能力，也让我熟悉了一个研发项目的具体流程，未来我将学习和巩固基本技术知识，使自己在团队中更有发言权，能够更有效地控制团队进度，学习专业的管理，营销方面的知识，未来有可能选择经济管理学院的专业。

曾柏霖：这次参加 robomaster 夏令营，我学到了许多东西，从前我只学习算法，然而因为我们组没有嵌入式，我不得不自己摸索，从软件 PID 的参数调整到嵌入式 can 通讯，pwm 通讯，都是我这次夏令营工作中所运用到的。不仅如此，团队合作的过程也是机器人开发中极为重要的一部分，将机械，硬软件与嵌入进行分工，这也是我不曾经历过的。同时，我还认识了许多朋友，从他们身上学到了许多东西。望下次我还能继续参加 DJI 的 robomaster 夏令营！

王津川：在夏令营的二十天里学到了很多，从对 solidworks 陌生到精通，从一辆普通步兵车到可以将拾取子弹与射击完美结合的步兵车，二十天，让我学会更多的与同伴合作、请求同伴的帮助，能够准确的执行每一项任务，对待问题更多的思考，向同伴们学习，培养了耐心和坚持的品质。来到夏令营就是来学习和发现自己的问题的，二十天当中感觉自己对专业知识有种前所未有的渴望，大家在各方面都不是非常专业的，在项目里都是边学边蒙边摸索的过程中成长的。同样也让自己看到了不少自己身上的态度问题，实在是需要改正。在一个项目中，一个项目管理者是很重要的，我组的项目管理十分精细和合理，虽然有时也会丢丢零件，但是还是比其他组要整齐的多。

王之晗：人要多动脑子，方法总是有很多的。遇到问题不要着急，先思考，再行动。合理安排工作顺序提升效率。大学组的机械脑子很会活，我还有很大的提升空间。

李逸斌：身为队伍的队长，每天需要处理非常多的问题，因此技术相关的东西其实并没有学到特别多。七七八八的学了下硬件的接线、软件的调参还有机械的建模，值得一提的是我现在已经能熟练使用机床加工零件。在以后的学习和研发过程中，我相信自己能再次成为队长带好队伍，并且能独立完成最基础的零件加工装配以及机器调参修改。

洪梓翔：这次夏令营非常难忘，虽然几乎都是互不相识的人，但在短短几天内

就能够合作的非常融洽。所说我们并不是整个高中组里最强的人，但我们都愿意积极学习，并且快速应用在我们的车车上。虽然最后的比赛中老是出状况，但是我们组还是挺乐观的，因为我们来到这不是在学习技术上的东西和在比赛中争取名次，而是在之间成长，学习到一些在学校中学不到的知识。

黄宇轩：我们组有明确分工的程序、机械和项目管理，可以说是很完整很丰富的。虽然大家总体水平都有待提高，但是我们的合作十分愉快，大家都能在自己擅长的领域做出成果。经过这个项目的锻炼我也发现自己有许多不足，比如从头开始学 Solidworks，不过很开心现在已经可以完成我们的建模需求，以后应该会更深入学习机械和建模并尝试新事物。

许胤哲：在这次夏令营中，我对算法进行了更加深入的学习，对以前没有接触过的嵌入式及硬件方面进行了初步的学习，能够用所学的知识完成 can 通讯、开关板的应用和电机的控制。其次，我更好的认识到了自己目前的技术水平，对我以后在技术上的学习提供了参考，同时我也认识到自己和其他营员之间的差距。另一方面，通过此次活动我也对今后的就业方向有了初步的规划。