

多功能微型水面无人船系统开发与工程验证技术报告

本项目针对小型水域精细化作业需求，在 500mm×300mm×100mm 的严苛物理尺寸约束下，研发了一款集成水面垃圾自动归集、定点定量投饵、水质浊度遥测与机器视觉辅助驾驶的多功能微型无人船。系统核心设计指标明确为：整船满载重量控制在 1540g 左右、满载吃水线约 50mm、浊度检测精度 0.1NTU、常规光照下视觉识别帧率稳定于 30FPS、底层控制端到端响应延迟低于 50ms。项目通过机械结构创新、电气拓扑优化与边缘算力调度三大维度的技术突破，在树莓派 Zero 2W 低算力硬件底座上实现了全部功能的稳定运行。实测结果表明，系统控制延迟均值为 42ms，视觉识别 IOU ≥ 0.75 ，续航时间较全时推理模式提升 40.3%，各项指标均达到设计要求。核心控制代码已托管至 GitHub 平台 (<https://github.com/vast-jiang/XiaoChuan>)。

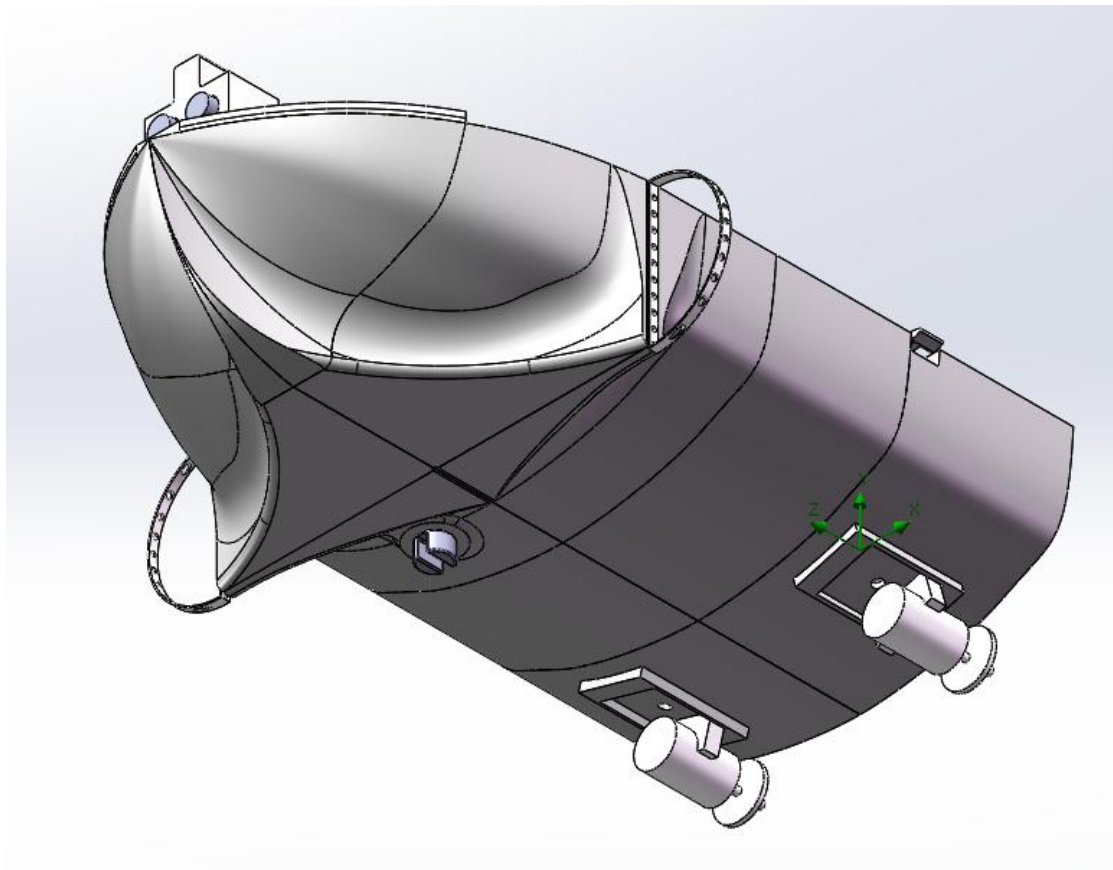
多功能微型水面无人船的工程基础建立在较小体积与重量约束之上，需同时承载四类核心功能：一是零能耗水面漂浮垃圾自动清理；二是可定量控制的定点饵料投放；三是实时在线水质浊度遥测；四是基于机器视觉的辅助目标追踪驾驶。系统核心设计指标明确为：浊度检测精度达到 0.1NTU，常规光照下视觉识别帧率维持在 30FPS 左右，底层控制指令的端到端响应延迟须低于 50ms，单次充电典型工况续航不低于 2 小时。

在主控平台的方案预研阶段，初始方案拟采用 STM32 微控制器作为主控以保障底层控制实时性。但经带宽吞吐量与算力的评估，单片机的架构无法原生支持高吞吐视频编码与并发 AI 推理任务，无法满足视觉辅助驾驶的功能需求。因此，整体硬件架构最终确立为以树莓派 Zero 2W 边缘计算节点为主控中枢，通过 I2C 与串行总线挂载各类外设：两路 370 2s 有刷水下推进器提供动力，PCA9685A 舵机驱动板实现多路 PWM 信号输出，ADS1115 高精度模数转换器采集模拟传感器信号，FS-iA6B 射频接收机实现手动遥控，TS-300B 浊度传感器与 HC-SR04 超声波模块构成环境感知阵列。

为最大化利用有限电池容量，水面垃圾清理功能摒弃了传统电动抽吸或机械抓取方案，采用纯物理斜面引流结构实现零能耗自动归集。船头中轴线参考破冰

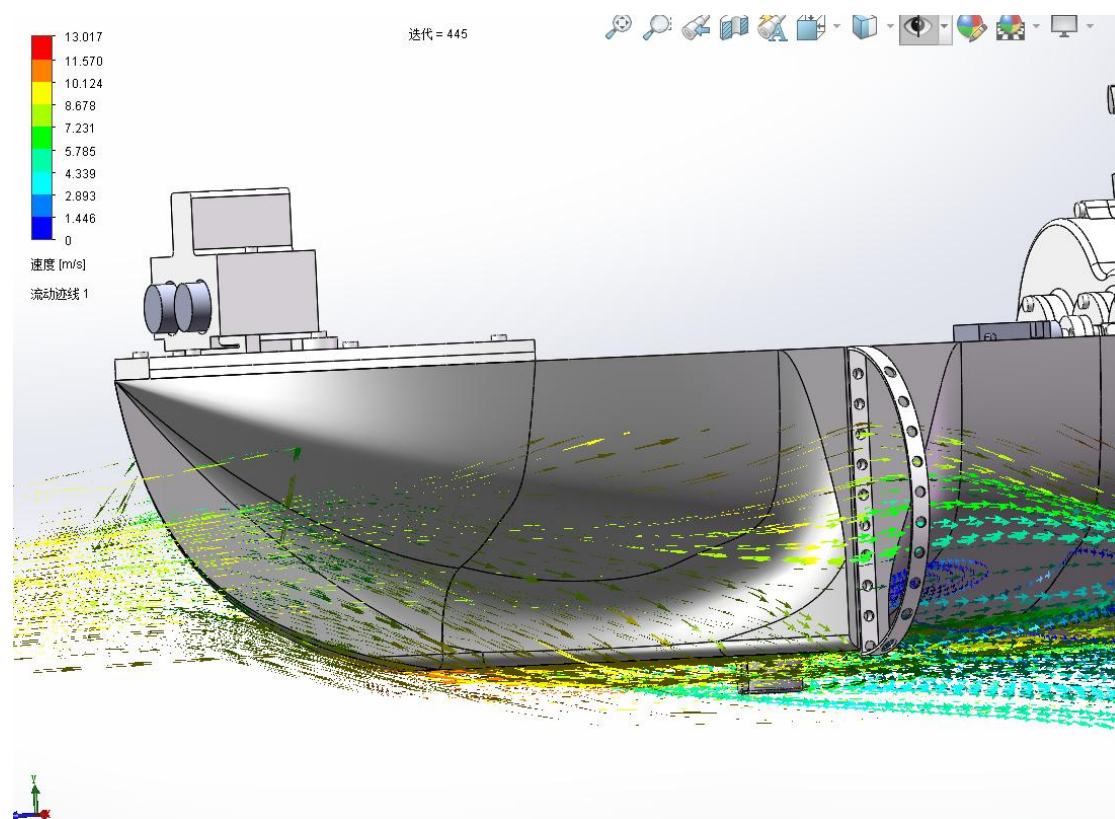
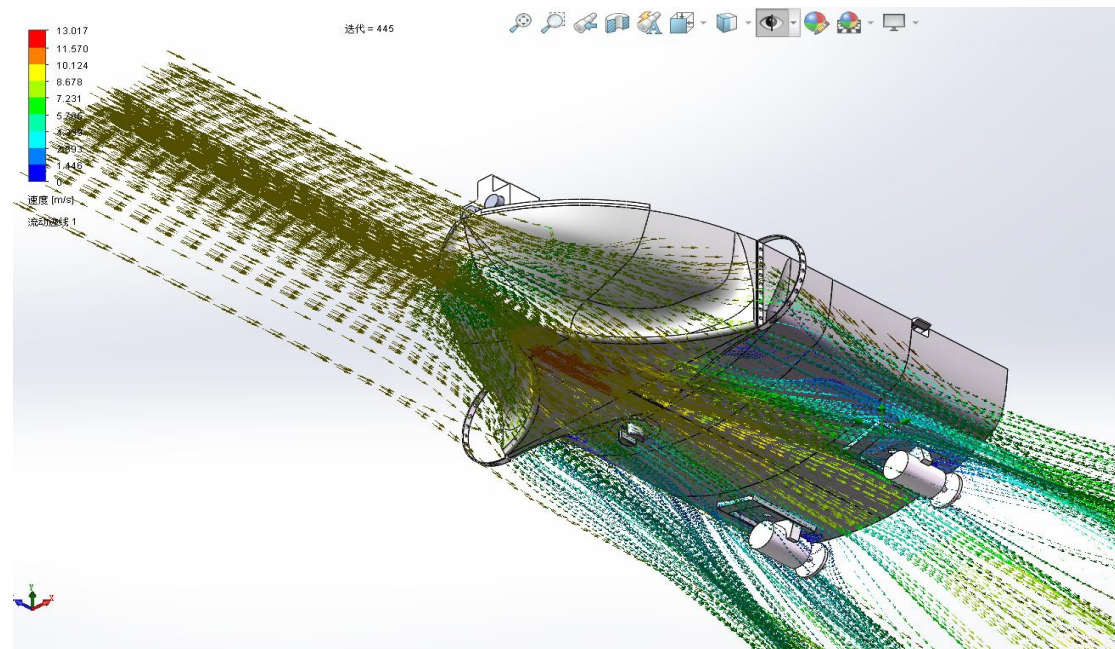
船流体力学特征，设计了特定角度的引流撞角构型，利用船体航行时切开的相对水流，自然剥离水面漂浮物并向后顺延导流至船身两侧挂载的可拆卸垃圾篓内。

下图是船头三视图：



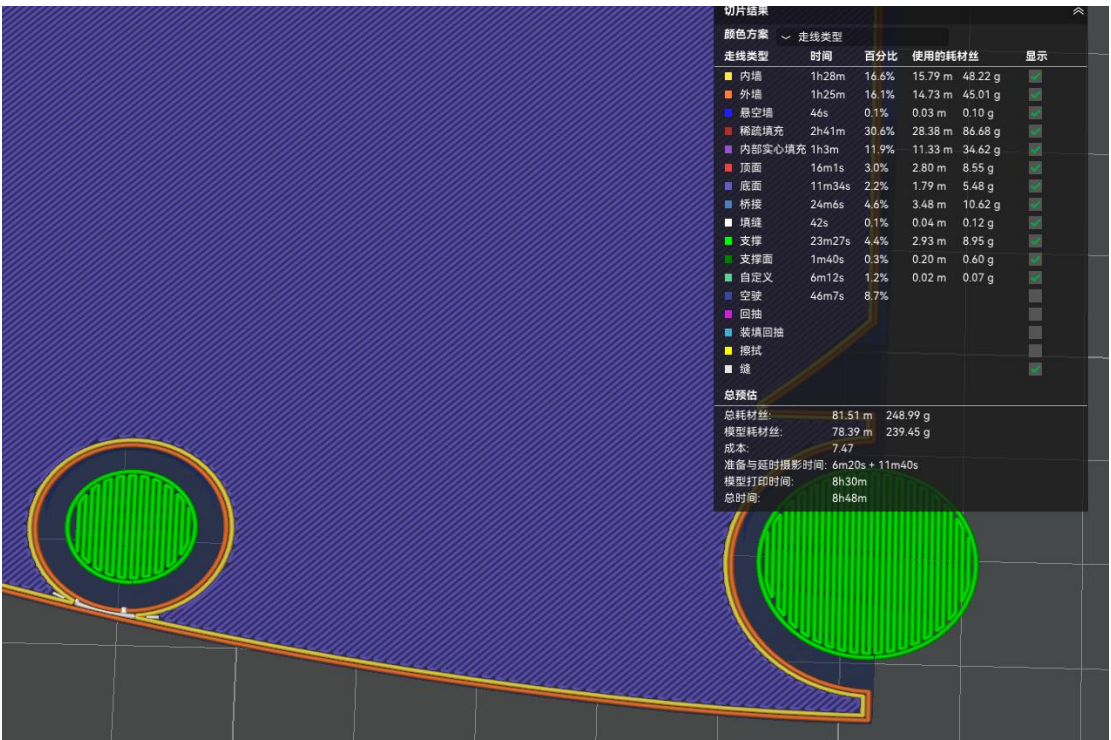
SolidWorks 流体仿真结果验证了该斜面结构可有效改变表面水流的驻点分布，使漂浮物沿预设路径进入垃圾收集区域。

下图是流水模拟：

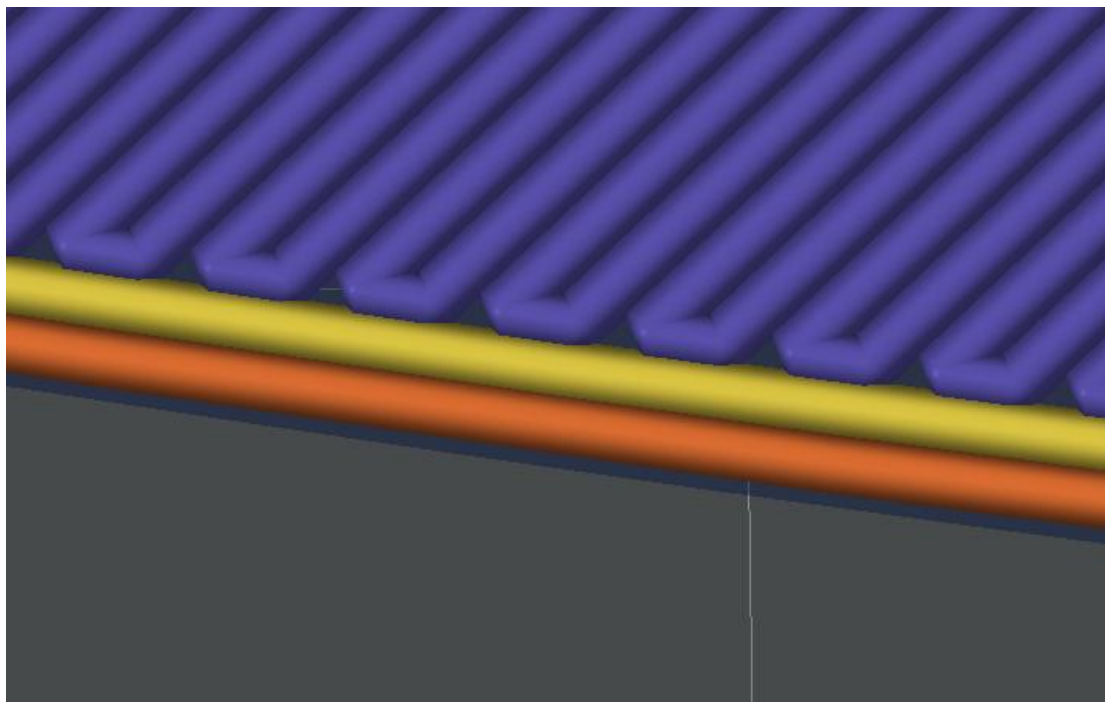


整船外壳采用 PETG 材质通过 FDM 工艺 3D 打印制造，但在初期 3 小时静水渗水实验中暴露出严重的结构性漏水缺陷。经切片软件挤出路径回溯排查，问题根源在于内墙与底面实心填充交界处无法实现绝对致密融合，且热床温度波动易导致底面微小翘边与 Z 轴气孔，进而引发毛细渗水。

通过切片软件可以看到主要漏水点在面与墙的结合处，见下图：

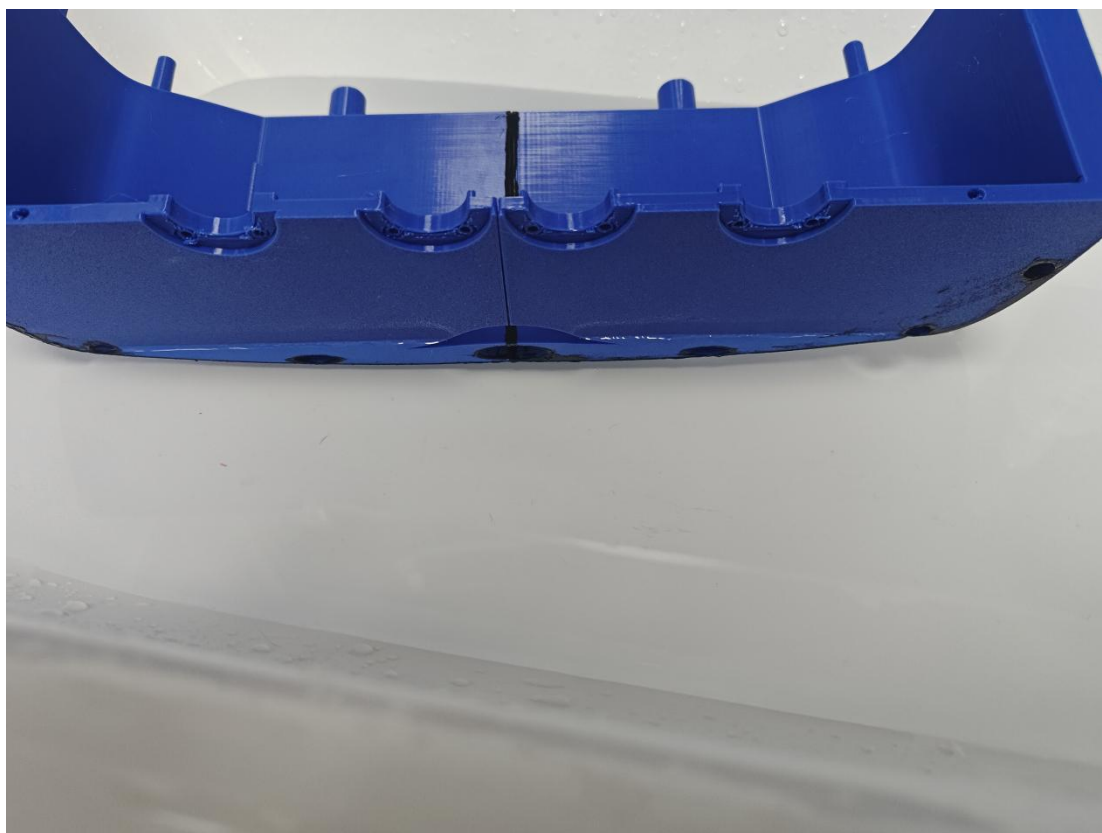


内外墙为平滑的连续挤出的耗材丝，而蓝紫色的底面与黄色的内墙粘连并不充分，见下图：



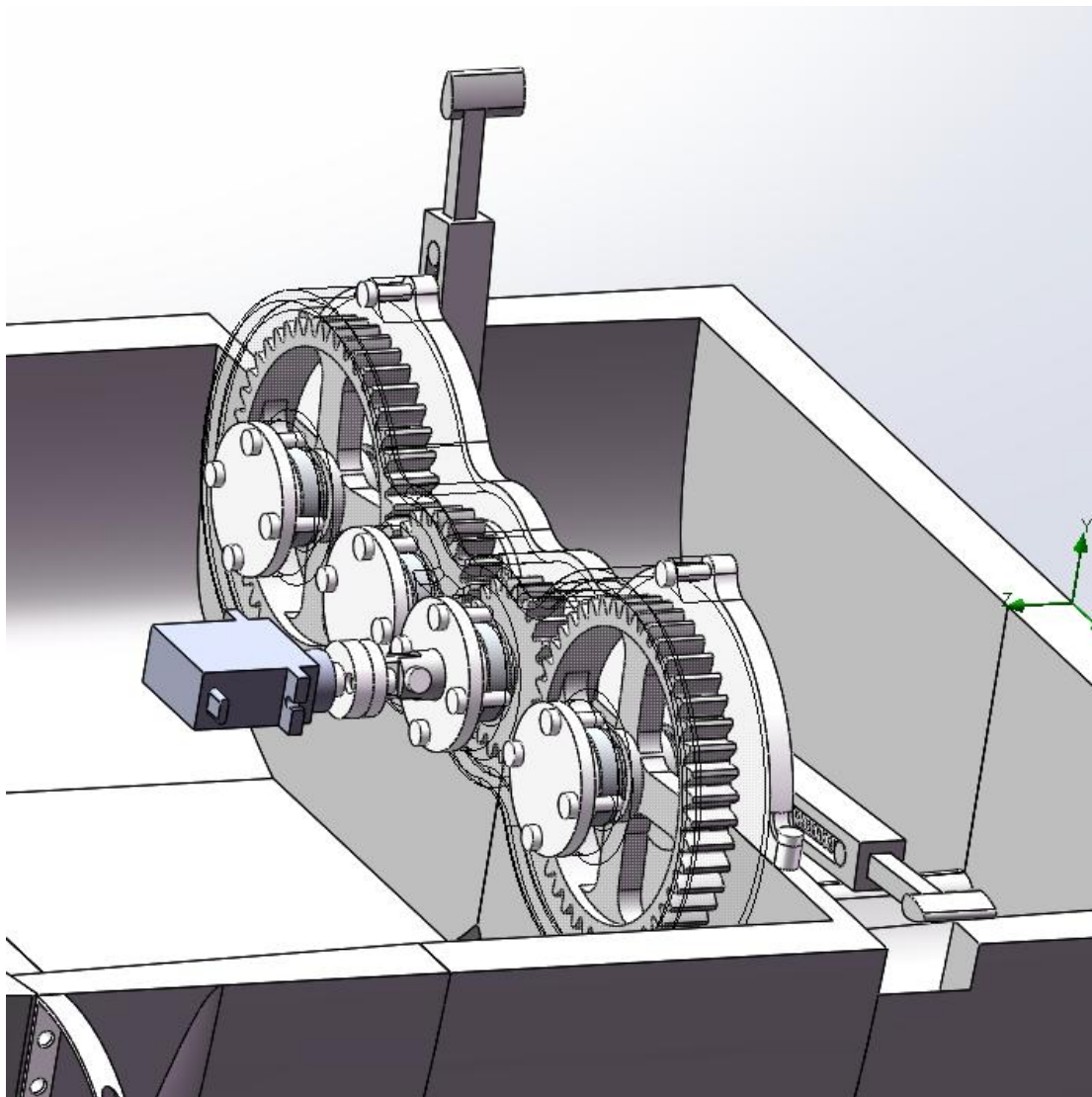
出于开发周期考量，工程端采用卡弗特 K-704B 有机硅密封胶对舱内所有受力接缝进行厚涂填缝，并对船体外表面喷涂层纹修补液封堵表面气孔。该方案成功通过了后续 5 小时极限渗水复测，同时为规避残存漏水风险，舱内所有核心板卡均架设在贴近顶盖的最高位置。

进行自然渗水实验 5 小时，效果良好，见下图：

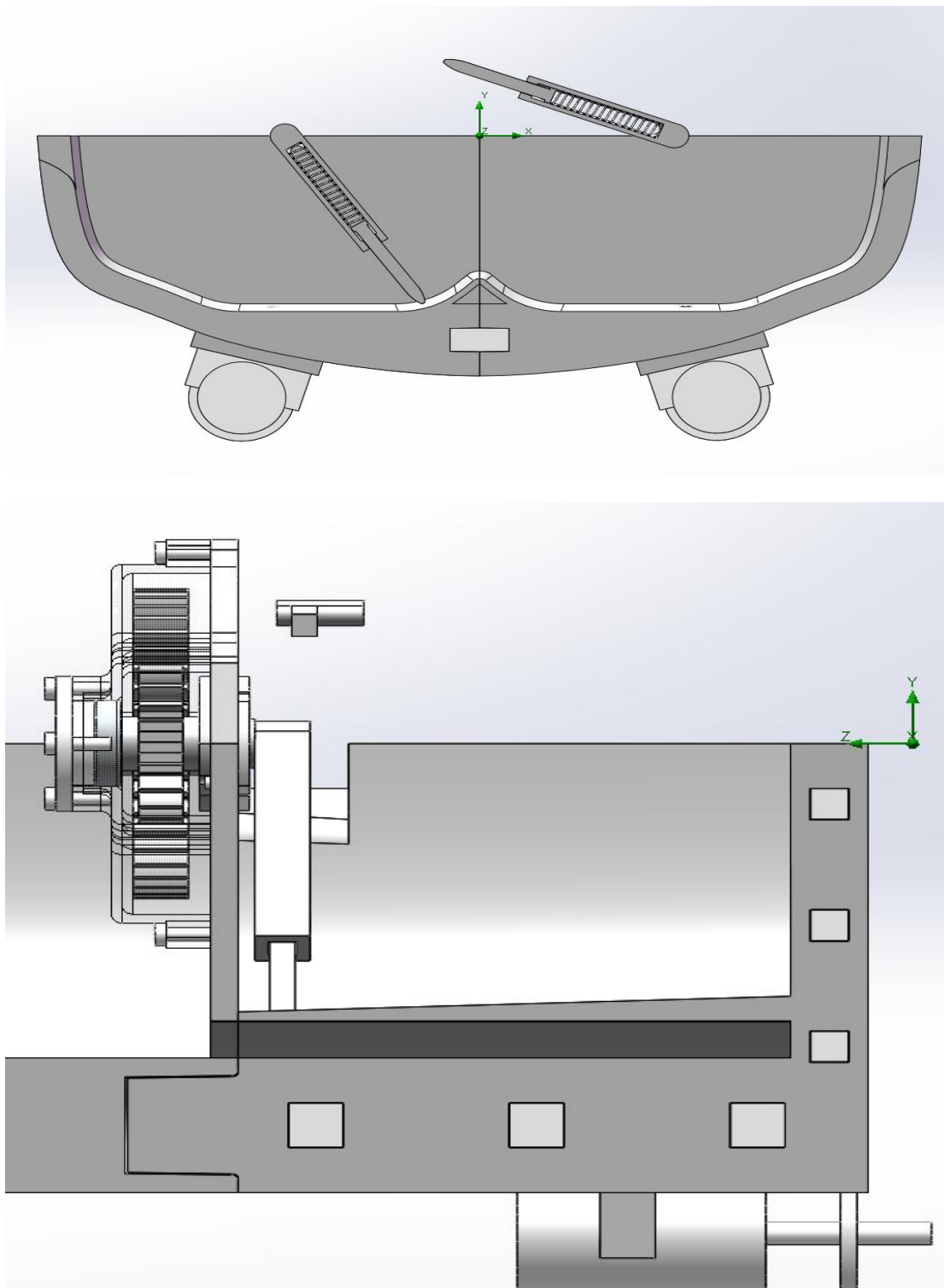


投饵机构受微型船体重心与吃水线严格约束，必须整体架设于舱盖上方。本机构利用一个 MG90S 360° 舵机为动力源，通过齿轮减速机构连接两个摆臂，摆臂内部设有弹簧保证与运料面的贴合。驱动舵机旋转带动摆臂转动，将饵料抛洒至水面。为确保摇臂能贴合料仓最低点，料仓底面特意设计为斜面结构。

以下为摆臂结构详图：



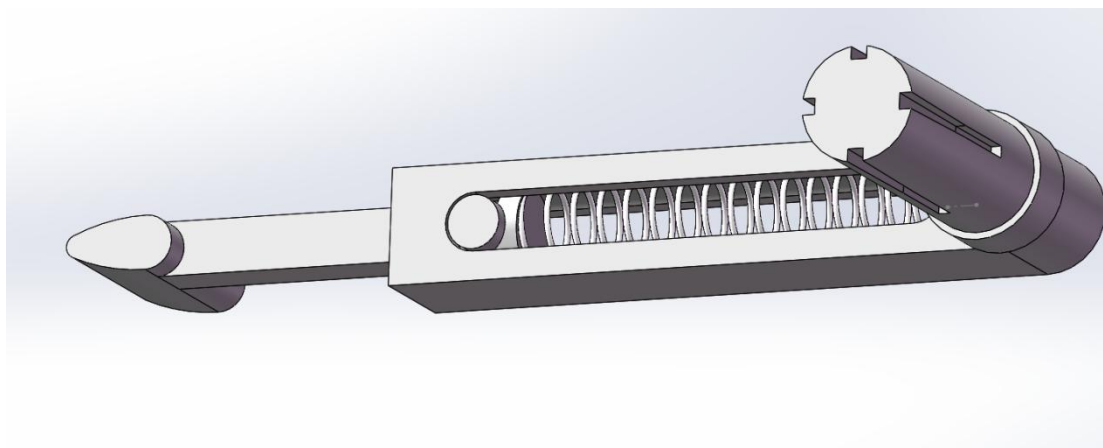
以下为料仓截面图：



摆臂采用 3D 打印一体成型，无拆装步骤以保证整体力学性质一致。但在实物联动测试中，当摆臂运行至料仓底部半径转换死角处时，复杂的受力状态导致滑动摩擦力远大于内部弹簧的伸缩补偿极限，引发舵机严重堵转卡死。

最终方案拆除了全部齿轮连杆机构，将执行端更换为类似扫帚的柔性软片拨刷结构，依靠非刚性材料自身的弯曲形变容差越过机械死点，彻底解决了卡弹故障。不过目前的软片在饵料含水率超过 60% 时偶尔会有残渣粘连，后续计划通过表面疏水涂层处理进行优化。

摆臂图：



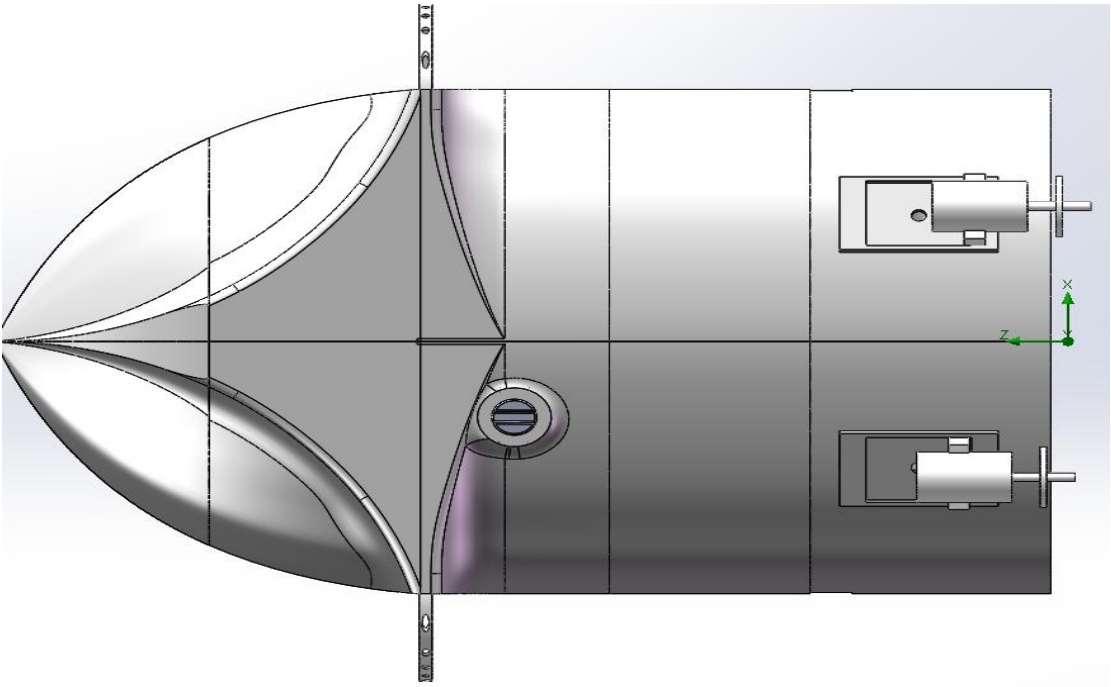
受舱内容积限制，整船仅配备一块 7.4V 2S 18650 锂电池作为唯一动力源，导致 7.4V 电机母线与要求 5V 纯净电压的逻辑外设之间存在显著的电压域供需矛盾。常规并联独立降压模块的方案不仅占用宝贵的舱内空间，还存在电机启动瞬间大电流引发的接地环路干扰隐患。

本项目使用电调 BEC 逆向供电拓扑作为核心解决方案：将双路双向有刷电调内置的 5V BEC 开关降压电路输出，逆向接入 PCA9685A 舵机驱动板的外部电源输入端。此拓扑将驱动板直接转化为高吞吐量的 5V 能源分发枢纽，供投饵舵机与 FS-iA6B 射频接收机直接取电；同时树莓派主控采用充电宝独立稳压供电，仅通过公共地线维持电位参考，彻底切断了电机端高频杂波输入计算核心

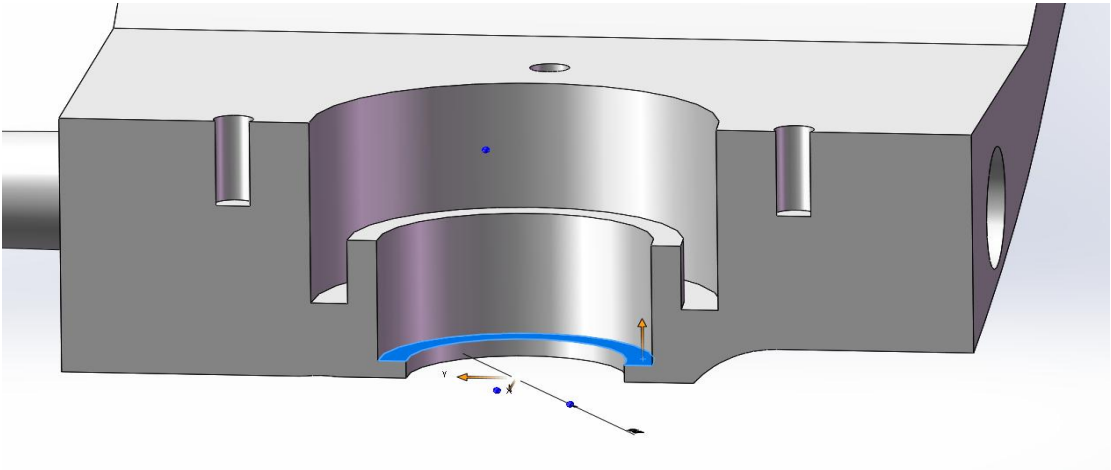
的路径。该方案既省去了独立降压模块的空间占用，又实现了动力域与逻辑域的电气隔离。

TS-300B 浊度传感器采用船底开孔安装方式，设计了阶梯式双层防水法兰结构：底层依赖高密度橡胶密封圈受压实现物理阻水，外层二次封涂防水硅胶并辅以压板锁死缝隙，确保长期水下工作无渗漏。传感器输出的模拟量经 ADS1115 转换为 I2C 数字量接入主控系统。

船底开孔图：



船底开孔防水设计图：



针对树莓派 Zero 2W 单核 512MB 内存的有限算力资源，底层软件采用 Python 构建了强异步多进程架构，实现了硬件底层轮询、AI 异步推理与 Web 通信网关的物理级线程隔离，避免了单线程循环引发的阻塞崩溃。硬件抽象层将传统的阻塞式 I2C 读写优化为异步非阻塞状态机，并引入严格的互斥锁机制，彻底解决了多传感器高频并发轮询导致的总线阻塞与资源锁死问题。

HC-SR04 超声波测距模块为宽电平输出，保障了树莓派 3.3V 引脚的电气安全，也方便了引脚的连接。HC-SR04 超声波测距模块在带水波纹的水面工况下，受水面漫反射影响，返回的距离数据存在剧烈跳动。为此，软件底座中实现了一套七元素滑动窗口复合滤波算法：维护一个长度为 7 的采样队列，每次新数据入队后执行快速排序，剔除最大、最小两个极值，再对剩余数据套用一阶低通滤波算法。实测结果表明，该算法将原始数据的方差从 14.5cm 降低至 1.2cm，输出了平滑稳定的连续距离信号。

代码截图 1：超声波七元素滑动窗口与复合滤波核心实现（摘自 hardware/sensors.py）

```
# 复合滤波算法：去极值平均 + 一阶低通
if len(self.distance_history) >= 5:
    # 1. 排序并掐头去尾（去除最大值和最小值突起）
    sorted_history = sorted(list(self.distance_history))
    valid_samples = sorted_history[1:-1]

    # 2. 计算修剪后的平均值
    avg_dist = sum(valid_samples) / len(valid_samples)

    # 3. 低通平滑输出（新数据占 40%，历史数据占 60%）
    if self.distance_cm == -1.0:
        self.distance_cm = round(avg_dist, 1)
    else:
        smoothed_dist = (self.distance_cm * 0.6) + (avg_dist * 0.4)
        self.distance_cm = round(smoothed_dist, 1)
```

针对廉价有刷电机固有的物理响应不对称特性，实测中发现左侧电机起步响应快而右侧迟滞，导致推油门瞬间船体左甩、满油门时右偏。底盘控制解算器中

引入了动态补偿逻辑：通过 Trim 配平参数将左电机的最大 PWM 占空比强行截断在 85%；同时加入死区前馈补偿，当摇杆指令越过中位死区时，立刻给右电机注入 13% 的初始基准占空比。通过纯软件层面的校准补偿，双发电机实现了同步直线行驶。

代码截图 2：底层混控限幅配平与右电机起步死区补偿（摘自 hardware/motor_controller.py）

```
def _update_mixer(self):
    """差速混控解算"""
    if not self.ready: return

    try:
        left_pwr = self.current_throttle + self.current_steer
        right_pwr = self.current_throttle - self.current_steer

        # 基础限速配平
        left_pwr *= self.LEFT_TRIM
        right_pwr *= self.RIGHT_TRIM

        # 只要动力指令超过微小的死区(0.02)，就强行加上起步补偿力
        if left_pwr > 0.02: left_pwr += self.LEFT_START_OFFSET
        elif left_pwr < -0.02: left_pwr -= self.LEFT_START_OFFSET

        if right_pwr > 0.02: right_pwr += self.RIGHT_START_OFFSET
        elif right_pwr < -0.02: right_pwr -= self.RIGHT_START_OFFSET
```

机器视觉模块是系统调试成本最高的部分。初期部署 YOLOv11 模型时，树莓派 CPU 会满载锁死进程，不仅帧率极低，大电流波动还频繁导致 CSI 摄像头驱动崩溃黑屏。为保证系统稳定性，视觉模型降维替换为 MobileNetSSD Caffe 轻量化模型，并置于 OpenCV DNN 后台线程中进行异步推理。

即便如此，持续的视觉推理仍会导致核心温度过高与功耗激增。为此，系统开发了严格的按需唤醒调度机制：在常规手动遥控巡航工况下，系统级强行终止底层视频流采集进程，大幅降低整体功耗与核心发热；仅当后台解析线程监听到

遥控器 SwC 拨杆下达的接管指令时，程序才会在 1 秒内重新申请视频节点、拉起摄像头并激活 PID 目标追踪。

代码截图 3：基于物理开关的摄像头按需拉起与系统级强制休眠控制（摘自 main.py 及 vision/detector.py）

```
# main.py 核心修改区：通过监听 SWC 拨杆，联动控制摄像头的启停
ai_mode_enabled = (current_swk > 1800)
if ai_mode_enabled != last_ai_state:
    if ai_mode_enabled:
        steer_pid.reset()
        if vision:
            vision.start_stream() # 唤醒摄像头与推理线程
    else:
        if vision:
            vision.stop_stream() # 深度休眠摄像头
    last_ai_state = ai_mode_enabled

# vision/detector.py 底层物理休眠调度
def stop_stream(self):
    """外部调用：深度休眠摄像头释放计算资源"""
    if self.camera_active:
        self.camera_active = False
        if self.process:
            self.process.kill() # 强制挂起 rpikam-vid 进程释放 CPU
            self.process = None
```

系统采用分层权限控制架构：由 Flask 搭建的 Web 网关仅用于图传与遥测数据渲染，直连树莓派串口的 FS-iA6B 接收机掌握最高控制权。后台守护线程以 115200 波特率持续解析 i-BUS 底层十六进制数据帧，一旦测算到 200 毫秒内无有效帧头，立刻触发防飞车保护，将所有 PWM 通道寄存器清零，强制电机停机。

代码截图 4：串行通信超时断连判定与防飞车停机保护（摘自 hardware/ibus_reader.py 及 main.py）

```
# hardware/ibus_reader.py 协议解析与超时保护
if time.time() - self.last_update > 0.5:
    self.is_connected = False
    self.channels = [self.PWM_MID] * 14 # 通道归中位

# main.py 物理安全保护触发
if receiver.is_connected:
    # 正常解算油门与转向...
else:
    if rc_or_ai_was_active:
        print("[WARN] RC Signal Lost. System halted.")
        ctrl.halt() # 将所有 PWM 通道写零，强行终止电机与舵机运转
        rc_or_ai_was_active = False
```

控制响应延迟：通过逻辑分析仪持续抓取串行总线电平状态，证实 FS-iA6B 接收机的 i-BUS 协议解析端到端通信延迟均值为 42ms，满足低于 50ms 的设计要求。

视觉识别性能：按需唤醒后的 MobileNetSSD 模型在多云或阴天的理想光照下，锁定 5 米内水面目标的 IOU 维持在 0.75 以上，识别帧率稳定在 28-30FPS；但在遭遇强烈直射阳光时，水面反光产生的逆光会严重干扰曝光计算，导致帧率偶尔降至 20FPS 左右，且识别精度显著下降。

水质检测精度：TS-300B 浊度传感器经标准福尔马肼溶液标定，经 I2C 总线传输的浊度数据精度稳定在 0.1NTU，完全符合设计指标。

续航性能：多次充放电平均耗电测试表明，得益于视频流与推理线程的按需休眠调度，整机在典型工况下的续航时间较全时推理模式提升了约 40%。

在 Web 端图传与遥测数据流测试中，多设备并发访问曾导致树莓派内部套接字资源耗尽。通过在 Flask 应用层增设连接最大数量限制，并引入帧缓冲区强制覆写机制，确保了底层物理遥控控制链路与上层移动端监测节点在重载网络环境下的双向稳定通信。

代码截图 5：防套接字阻塞的单帧覆写 MJPEG 异步推流（摘自 web/app.py）

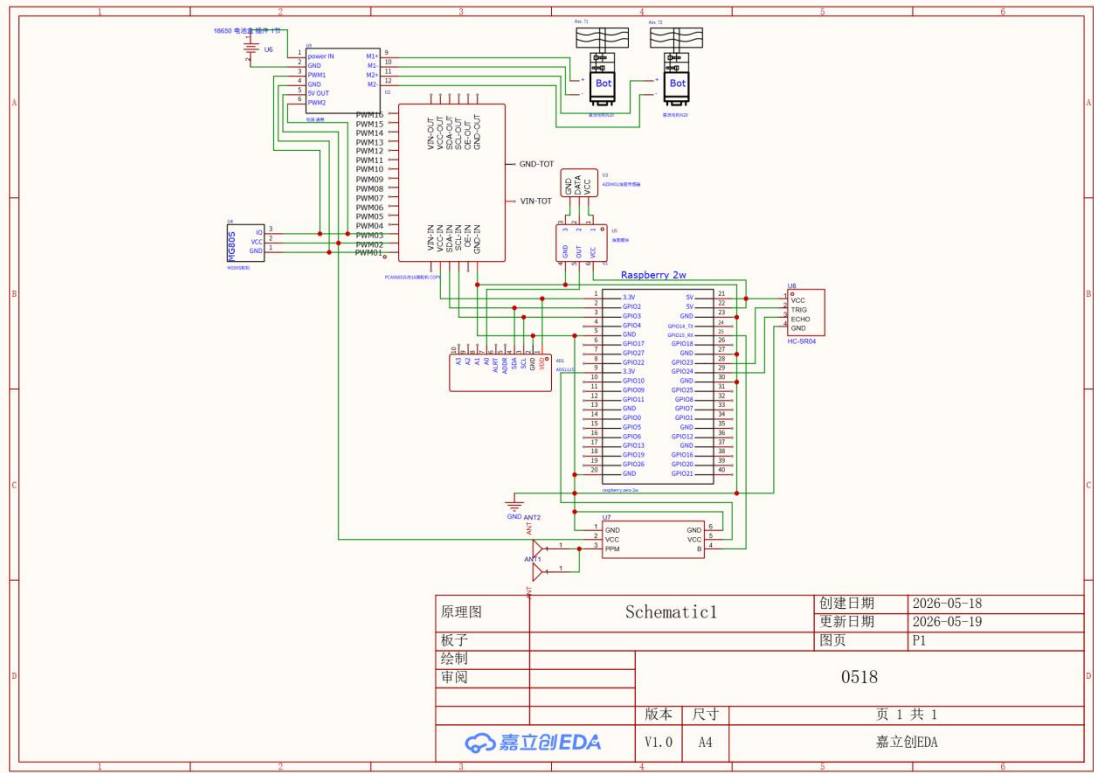
```
@app.route('/stream')
def stream():
    """实时机器视觉图传 MJPEG 网络异步推流端点"""
    def gen():
        while True:
            if has_cam:
                b = vision.get_jpg() # 获取被覆写的最新一帧缓冲，避免队列堆积
                if b:
                    yield (b'--frame\r\nContent-Type: image/jpeg\r\n\r\n' + b + b'\r\n')
                time.sleep(0.05) # 主动锁频释放网络带宽
    return Response(gen(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

本项目在较小体积与低算力硬件约束下，完整实现了多功能微型水面无人船的全部预设功能。零能耗斜面引流垃圾归集、电调 BEC 逆向供电拓扑以及异步多进程按需休眠调度三大核心创新，大幅优化了微型载具的空间利用率、能源分配效率与系统鲁棒性。实测数据验证了系统各项核心指标均达到设计要求，具备良好的工程转化价值。

当前原型机仍存在一些待优化的问题：舱内大量采用杜邦飞线与裸露洞洞板，在长期水面震动工况下偶发接线端子接触不良导致 I2C 节点掉线；强光环境下视觉识别性能下降明显。下一阶段的工作将聚焦于硬件底板的 PCB 定制固化，计划在底层铺设大面积的模拟地与数字地隔离层，对承载 7.4V 大电流的电机走线进行加粗并做阻焊开窗上锡处理；同时在软件层引入毫米波雷达的冗余数据融合算法，进一步填补强光环境下的感知盲区，推动系统向全天候工业级应用迈进。

附录 1：

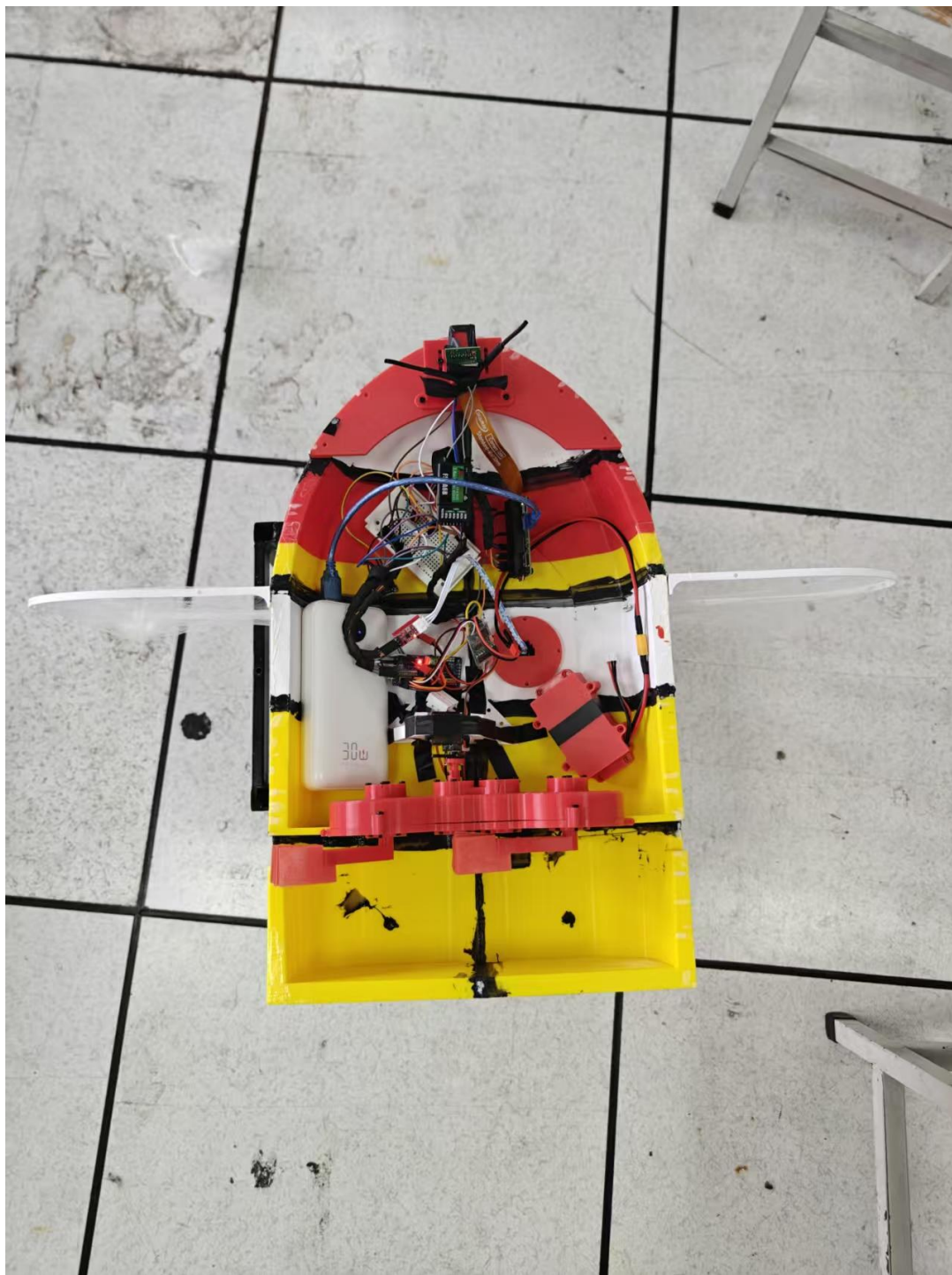
原理图：



附录 2

：成品展

示俯视图



前视图：



后视图：

