

Intelligentcar2022-FZ3B

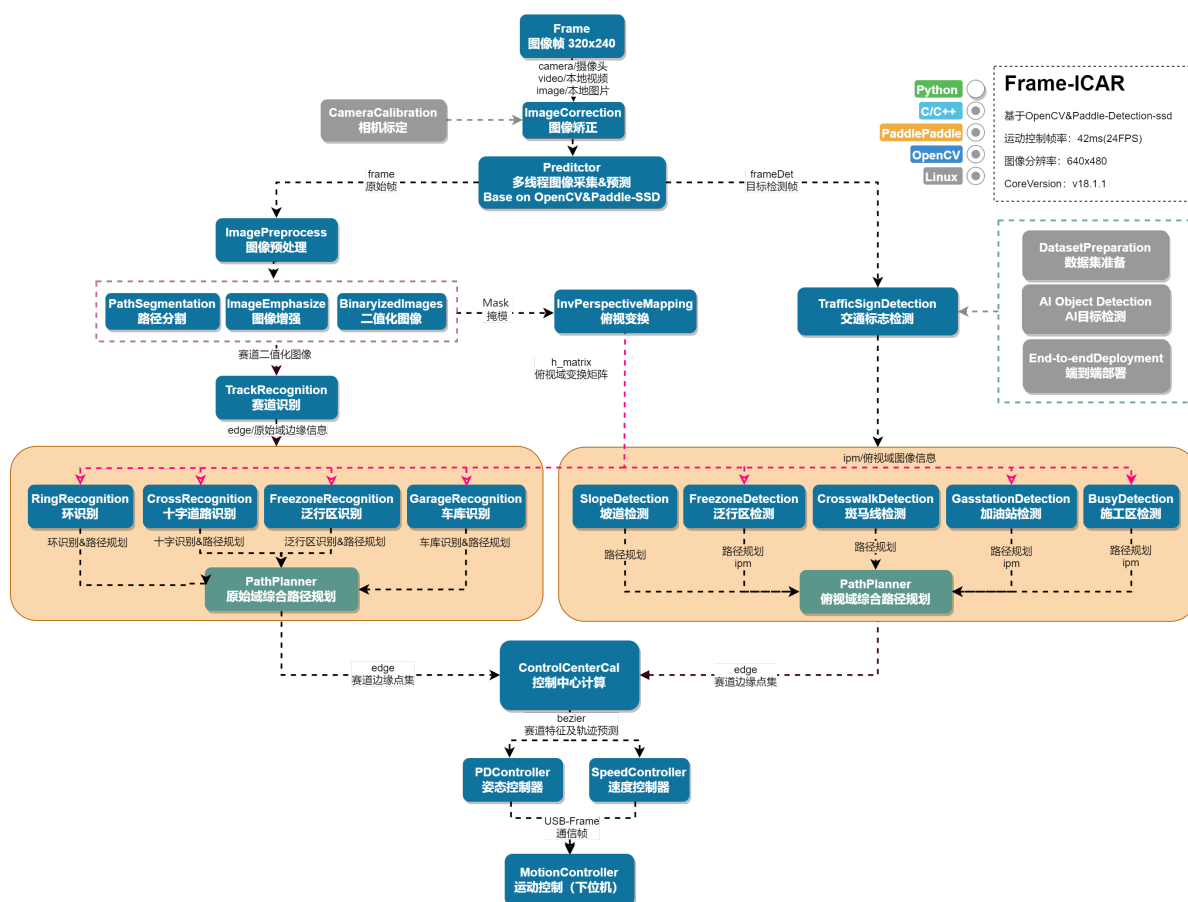
介绍

全国大学生智能汽车竞赛-完全模型组-开源共享软件资源（Edgeboard-FZ3B）

严肃声明！！该软件方案仅限智能汽车竞赛学习交流，本人基于智能车爱好者维度向公众公开源代码，不涉及任何商业行为，请全体技术爱好者共同监督指导！！

同时将继续汇集历届经验，持续为大家共享相关资源，欢迎批评改进，未完待续！！

软件架构

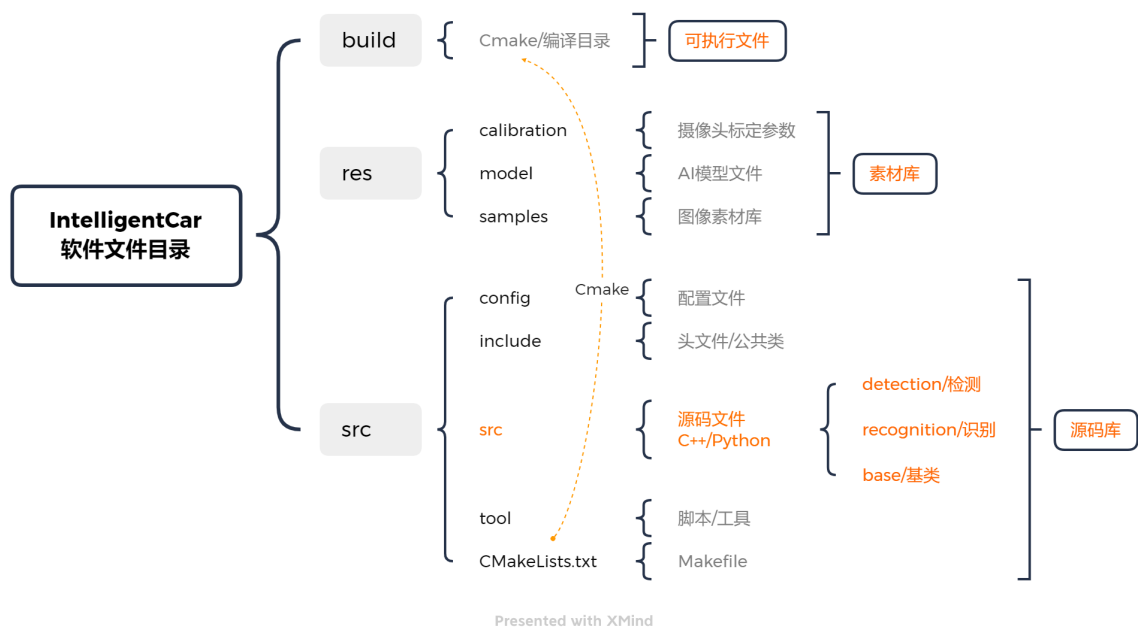


注意：该版本不包含所有功能块，部分内容需要各位参赛选手继续搭建...（环岛/加油站/泛行区）

使用说明

1. 该软件工程基于C++/OpenCV2/Python2.7/PaddlePaddle-Detection搭建；
2. 基于百度智能汽车-完全模型组及赛事开源Edgeboard板卡FZ3B调试开发；
3. 默认将该软件下载到EB：root@EdgeBoard:~/workspace目录下即可启用；
4. 除此之外，该软件支持仿真调试（跑视频）/比赛模式两种方式运行；

文件目录



《智能汽车完全模型组 — EB软件参考目录》						
序号	分类/Type	功能/Func		功能说明/Info	完成率	备注
1	Tools	相机显示	camera_display	> UI显示网格线，机械对称性校验；	100.00%	
2		图像采集	image_collection	> 遥控手柄控制车辆（支持无线/有线）； > 单帧/连续彩图控制；	100.00%	
3		图像合成	image2video	> 图片合成视频；	100.00%	
4	Base	图像预处理	image_preprocess	> 原始图像畸变矫正； > 图像增强去噪； > 图像二值化；	100.00%	
5		逆透视变换	perspective_mapping	> 二维图像透视变换（斜视→俯视）； > 二维坐标透视变换（斜视→俯视）； > 二维图像反透视变换（俯视→斜视）； > 二维坐标反透视变换（俯视→斜视）；	100.00%	
6	DGP 赛道识别	赛道识别	track_recognition	> 搜索赛道边缘； > 搜索岔路坐标； > 计算赛道边缘斜率/方差； > 支持赛道重搜索；	100.00%	
7		十字识别	cross_recognition	> 左/右十字路口识别与路径规划； > 左/右斜入十字路口识别与路径规划；	70.00%	> 问题1：车库和泛行区易识别成十字； > 运动轨迹拟合后不必进行十字处理，稳定性90%
8		车库识别	garage_recognition	> 入库识别与路径规划； > 出库识别与路径规划；	50.00%	> 入库路径规划有缺陷；
9		泛行区识别	freezone_recognition	> 左/右入泛行区识别与路径规划；	90.00%	> 问题1：常规方法识别率较低，需要配合AI检测；
10	AI 赛道检测	施工区检测	busy_detection	> 施工区AI标志检测与任务触发； > 锥桶AI检测与车道线拟合；	100.00%	
11		斑马线检测	crosswalk_detection	> 斑马线AI标志检测；		> 辅助车辆入库； > 问题1：存在AI误检几率；
12		坡道检测	slope_detection	> 坡道AI标志检测； > 坡道图像与车辆姿态优化；		
13	Base	控制中心拟合	controlcenter_cal	> 赛道边缘有效行优化； > 赛道类型判断与运动轨迹拟合； > 加权控制中心计算；	100.00%	
14		运动控制	motion_controller	> 加载Json调试参数； > PD姿态控制器； > 车辆变速控制；	100.00%	
15		串口通信	uart	> CH340/CP2102，USB转串口下位机通信；	100.00%	
16						

程序启动说明

1. 编译代码/Make

```

# Image2video
set(IMAGE2VIDEO_PROJECT_NAME "image2video") 图像合成视频
set(IMAGE2VIDEO_PROJECT_SOURCES ${PROJECT_SOURCE_DIR}/tool/image2video.cpp)
add_executable(${IMAGE2VIDEO_PROJECT_NAME} ${IMAGE2VIDEO_PROJECT_SOURCES})
target_link_libraries(${IMAGE2VIDEO_PROJECT_NAME} PRIVATE pthread )
target_link_libraries(${IMAGE2VIDEO_PROJECT_NAME} PRIVATE ${OpenCV_LIBS})

# CameraDisplay
set(CAMERA_DISPLAY_PROJECT_NAME "camera_display") 摄像头显示校正
set(CAMERA_DISPLAY_PROJECT_SOURCES ${PROJECT_SOURCE_DIR}/tool/camera_display.cpp)
add_executable(${CAMERA_DISPLAY_PROJECT_NAME} ${CAMERA_DISPLAY_PROJECT_SOURCES})
target_link_libraries(${CAMERA_DISPLAY_PROJECT_NAME} PRIVATE pthread )
target_link_libraries(${CAMERA_DISPLAY_PROJECT_NAME} PRIVATE ${OpenCV_LIBS})

# ImageCollection
set(COLLECTION_PROJECT_NAME "image_collection") 遥控采图
set(COLLECTION_PROJECT_SOURCES ${PROJECT_SOURCE_DIR}/tool/image_collection.cpp)
add_executable(${COLLECTION_PROJECT_NAME} ${COLLECTION_PROJECT_SOURCES})
target_link_libraries(${COLLECTION_PROJECT_NAME} PRIVATE pthread )
target_link_libraries(${COLLECTION_PROJECT_NAME} PRIVATE ${OpenCV_LIBS})
target_link_libraries(${COLLECTION_PROJECT_NAME} PRIVATE serial)

#-----
# [ bin ] ==> [ main ]
#-----
set(PROJECT_NAME "icar") 智能车运行
set(INTELLIGENTCAR_CAR_PROJECT_SOURCES ${PROJECT_SOURCE_DIR}/src/icar.cpp)
add_executable(${PROJECT_NAME} ${INTELLIGENTCAR_CAR_PROJECT_SOURCES})
target_link_libraries(${PROJECT_NAME} PRIVATE pthread )
target_link_libraries(${PROJECT_NAME} PRIVATE ${OpenCV_LIBS})
target_link_libraries(${PROJECT_NAME} PRIVATE paddle_full_api_shared)
target_link_libraries(${PROJECT_NAME} PRIVATE serial)

```

编码可执行程序，可修改CMakeLists.txt;

到build路径下cmake:

```

root@EdgeBoard: ~/workspace/intelligentcar-fz3b/intelligentcar/build
root@EdgeBoard:~/workspace/intelligentcar-fz3b/intelligentcar/build# cmake ../src/
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting C compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting C compile features

```

编译代码:

```

r/build
root@EdgeBoard:~/workspace/intelligentcar-fz3b/intelligentcar/build# make icar -j
Scanning dependencies of target icar
[ 50%] Building CXX object CMakeFiles/icar.dir/src/icar.cpp.o
[100%] Linking CXX executable icar
[100%] Built target icar
root@EdgeBoard:~/workspace/intelligentcar-fz3b/intelligentcar/build#

```

2. 修改config文件，启动相关模式

```
{} motion.json M x
src > config > {} motion.json > [ ] record > {} 0 > #debug
25     "CrossEnable": false,
26     "circles": 2,
27     "record": [
28     {
29         "#speedLow": "智能车最低速",
30         "#speedHigh": "智能车最高速",
31         "speedDown": "特殊元素<加油站>|<施工区>减速速度",
32         "speedFreezone": "泛行区速度",
33         "speedGasBusy": "加油站|施工区速度",
34         "speedSlop": "坡道(桥)速度",
35         "#runP1": "一阶比例系数: 直线控制量",
36         "#runP2": "二阶比例系数: 弯道控制量",
37         "#runP3": "三阶比例系数: 弯道控制量",
38         "#turnP": "一阶比例系数: 转弯控制量",
39         "#turnD": "一阶微分系数: 转弯控制量",
40         "#debug": "调试模式使能(存图|看图)",
41         "#saveImage": "存储原始图像使能(非调试模式下)",
42         "#rowCutUp": "图像顶部切行(前瞻距离)",
43         "#rowCutBottom": "图像底部切行(盲区距离)",
44         "#disGarageEntry": "车库入库距离(斑马线Image占比%)[0.0, 1.0]",
45         "#rateTurnFreezone": "泛行区躲避禁行标志的转弯曲率% [-1.0, 1.0]",
46         "#GarageEnable": "车库使能",
47         "#GasStationEnable": "加油站使能",
48         "#BusyAreaEnable": "施工区使能",
49         "#SlopEnable": "坡道使能",
50         "#FreezoneEnable": "泛行区使能",
51         "#RingEnable": "环岛使能",
52         "#CrossEnable": "十字使能",
53         "#circles": "智能车运行圈数"
54     }
55 ]
56 }
```

修改motion.json文件后，不必再次编译即可运行，方便后期调试。可通过debug标志启动仿真（跑视频）模式和比赛（竞速）模式，仿真模式下默认调取本地视频，并且通过VNC观察实时图像运算结果；比赛模式将关闭图像显示和图形绘制（省时间/10ms）。

3. 连接下位机，启动程序

运行icar程序需要连接下位机，否则程序退出，插入ch340/cp2102的USB串口即可。

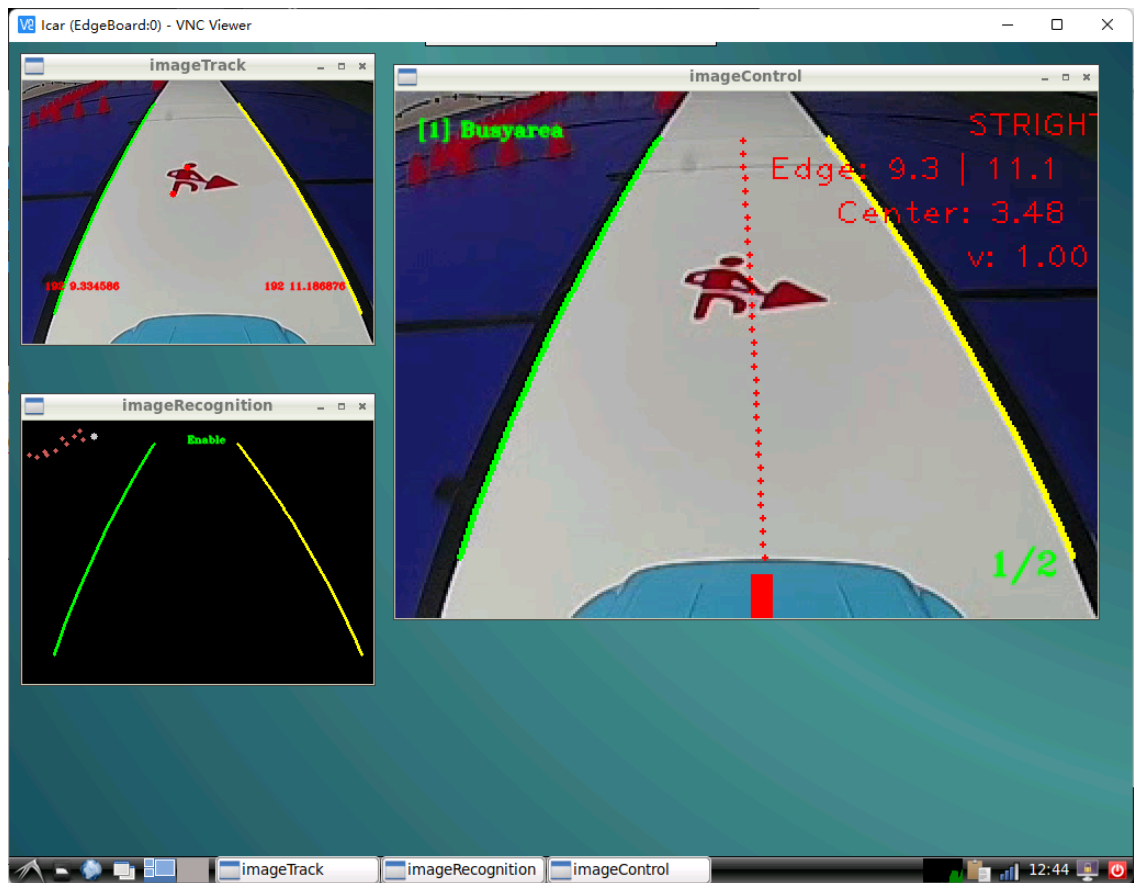
```
root@EdgeBoard: /workspace/intelligentcar-fz3b/intelligentcar/build# ./icar
--- runP1:1.4 | runP2:0.015 | runP3:0
--- turnP:3.5 | turnD:3.5
--- speedLow:1m/s | speedHigh:1.4m/s
Camera Param: frame rate = 23.976 width = 320 height = 240
Config:{"format":"RGB","input_height":300,"input_width":300,"labels_file_name":"label_list.txt","mean":[127.5,127.5,127.5],"model_file_name":"mobilenet-ssd-model","params_file_name":"mobilenet-ssd-params","scale":[0.007843,0.007843,0.007843],"threshold":0.3}
Model Config Init Success !!!
Predictor Init Success !!!
run frame time : 0ms
run frame time : 805ms
run frame time : 63ms
run frame time : 47ms
run frame time : 59ms
run frame time : 57ms
run frame time : 50ms
run frame time : 58ms
run frame time : 66ms
```

配置文件信息

AI模型信息

控制帧率/调试模式

4. 查看运行结果（debug模式）



5. 赛道奔跑（竞赛模式）

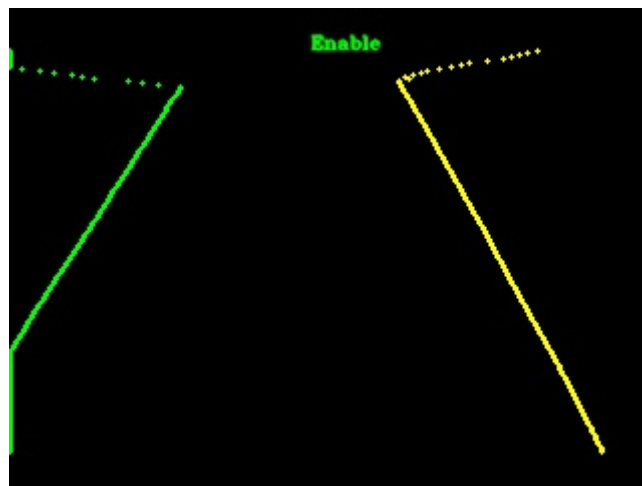
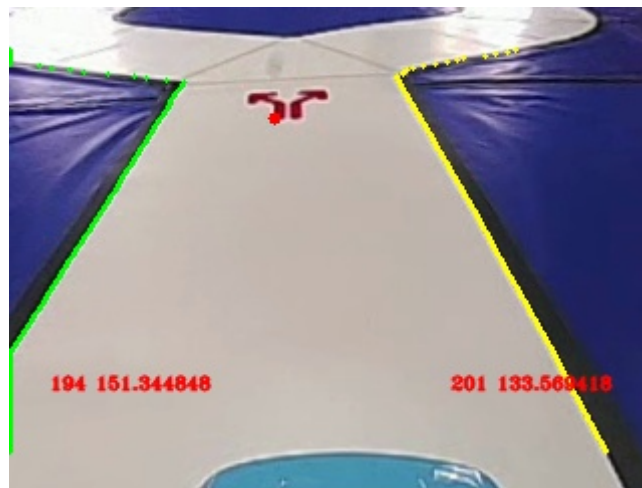
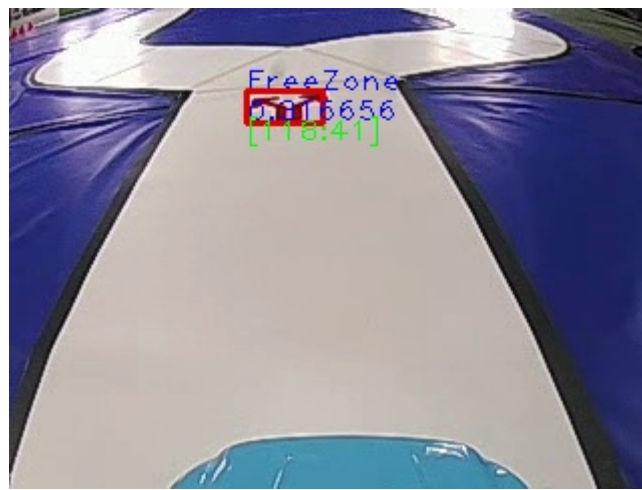
脱机运行：小车在赛道上奔跑时可通过 "nohup" 命令启动程序，此时拔掉网线继续运行。

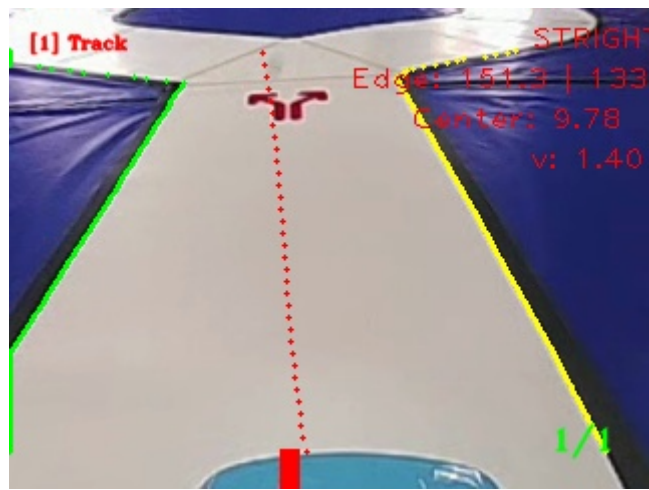
```
root@EdgeBoard: ~/workspace/intelligentcar-fz3b/intelligentcar/build
root@EdgeBoard:~/workspace/intelligentcar-fz3b/intelligentcar/build# nohup ./icar
nohup: ignoring input and appending output to 'nohup.out'
```

若程序无法自行退出，待重新插上网线后，通过Ctrl+C终止进程，或者查询当前进程ID（ps -ef|grep ./icar），并杀死当前进程（kill -9 ID）。

图像处理思路

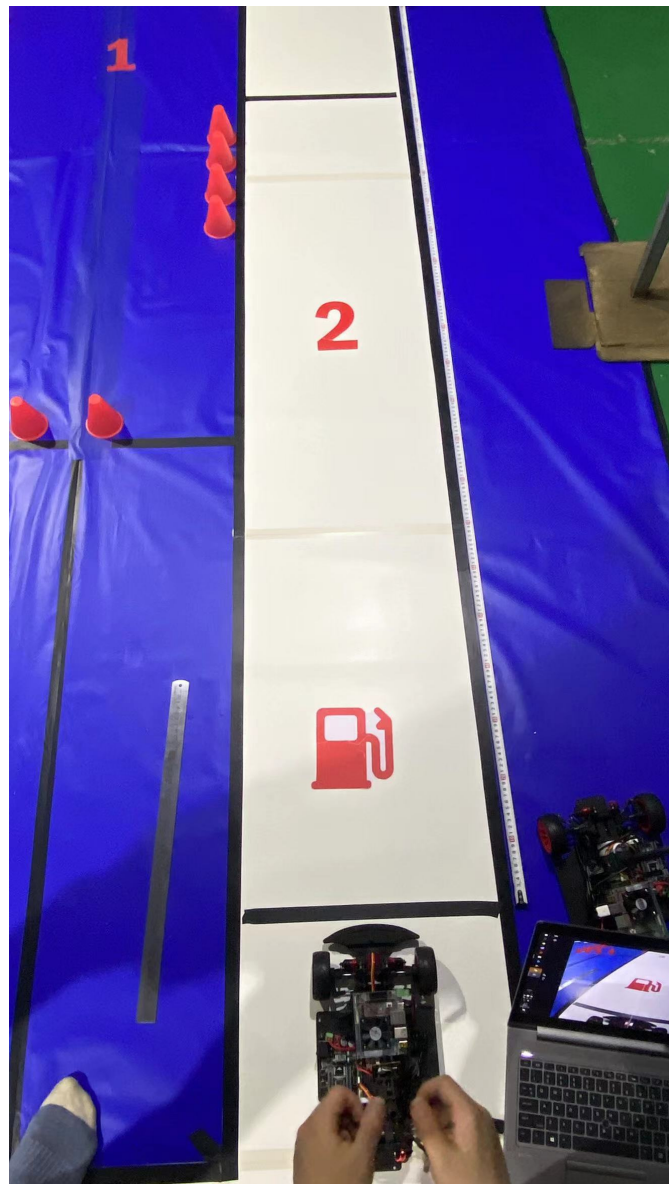
Image输出 → 预处理 → 赛道识别（Track） → 特殊元素识别（Rec/Det） → 路径拟合 → 运动控制
→ 下位机





车模配置

1. 摄像头前瞻距离：1.6~2.5米





2. 下位机设计PID模型/电机模型-控速，控制单位：m/s (float)
3. 下位机舵机控制，TIM/定时器计数上限20000，频率50Hz，占空比控制阈值[500, 2500]。上位机理论控制阈值[1100, 1500, 1900] / 左|中|右，舵机中值及左右阈值在实际的标定值上偏移。