注：使用Google ProtoBuffer、NetMQ的依赖包括

1. AsyncIO.dll （针对上位机.NetFramework 4.0框架，这里必须使用0.1.20.0版本）

添加方法：VS2010安装NuGet管理器（拷贝文件夹中的NuGet.Tools.vsix，点击安装重启VS2010即可），在菜单“工具→NuGet程序包管理器→程序包管理器控制台”中输入命令“Install-Package AsyncIO –Version 0.1.20.0”等待下载安装完成即可。

1. Google.Protobuf.dll 版本必须为3.5.1.0，不要擅自下载安装最新的版本，请直接拷贝添加引用即可。
2. NetMQ.dll 当前版本为3.3.3.4，与AsyncIO.dll 版本有依赖关系；该依赖项版本不同所依赖的AsyncIO 版本也不同，理论上根据资料查询对应下载安装相依赖的版本即可，但是由于 AsyncIO 的版本与 .NetFramework 框架版本又有关系，且目前暂不清楚是否与 Google.Protobuf 版本也有关系；所以不建议下载安装其他的版本，暂时使用当前版本即可。（NetMQ是使用C#对ZeroMQ的实现）

PS：[1]. 生成 .cs文件工具（proto文件夹）使用方法

<1>.手动新建编辑 .proto文件

<2>.点击 protoAll.bat 批处理执行文件后在”csharp”文件夹中可找到相关的 .cs文件，将文件添加到项目中即可

具体说明与使用方法，请详细阅览：<https://www.cnblogs.com/Leo_wl/p/7126321.html>，并参考或查询官网等资料。

1. <https://developers.google.com/protocol-buffers/docs/reference/csharp/>
2. <https://netmq.readthedocs.io/en/latest/>
3. <https://github.com/zeromq/netmq/>

[2]. SetEvn文件夹是基于原始ROS系统环境变量新改造的针对于新消息的“环境变量”设置项目。使用方法相同，以“管理员身份”点击运行bin→Release→SetEvn.exe即可：

程序会新注册两个系统环境变量

<1>. NETMQ\_LOCAL\_URI 用于绑定本地IP，格式为：tcp://本地IP:11312

<2>. NETMQ\_MASTER\_URI 用于绑定远程服务器IP，格式为：tcp://服务器IP:11312

消息说明：

1. 车速消息

topic：”/yida/robot/showspeed”

type：float

proto：RobotSpeedMessage.proto

message RobotSpeed {

required float linear\_x=1;

required float angular\_z=2;

}

1. 车体电压消息

topic：”/yida/robot/batteryinfo”

type：float

proto：RobotBatteryMessage.proto

message RobotBattery {

required float linear\_x=1;

}

1. 车轮驱动消息

topic：”/wheelDriveInfo”

type：string

proto：WheelDriveMessage.proto

message WheelDrive {

required string data=1;

}

1. 车体控制模式消息

topic：”/yida/robot/control\_mode”

type：int32

proto：RobotCtrlModeMessage.proto

message RobotCtrlMode {

required int32 data=1;

}

1. 车体控制消息

topic：”/yida/robot/motor\_control2”

type：string/int32

proto：RobotCtrlMessage.proto

message RobotCtrl {

required string direction=1;

required int32 level=2;

}

direction：前进 - forward

后退 - backward

左转 - left

右转 - right

原地左转掉头 – leftTurnAround

原地右转掉头 – rightTurnAround

1evel：1 – 慢 / 2 – 中 / 3 – 快

1. 云台控制消息

topic：”/yunTai\_ctrl”

type：string/int32

proto：YunTaiCtrlMessage.proto

message YunTaiCtrl {

required string direction=1;

required int32 number=2;

required int32 step=3;

}

direction：左转 – left

右转 – right

上仰 – up

下俯 – down

number：云台编号（从1开始）

step：云台每次旋转的幅度（例如：10度/次）

1. 相机控制消息

topic：”/camera\_ctrl”

type：string/int32

proto：CameraCtrlMessage.proto

message CameraCtrl {

required string cameraType=1;

required string command=2;

required int32 number=3;

required string scaleType=4;

required int32 scaleStep=5;

}

cameraType：可见光 – visual

红外 – infrared

command：拍照 – capture（相机执行拍照命令并传送图片）

调焦 – focus（相机执行调焦命令）

录像 – record（相机无动作，由上位机将预览图数据进行编码存储为影视文件）

number：相机编号（从1开始）

scaleType：调焦类型，当 command 为 focus 时，此项不为空

放大 – ZoomIn

缩小 – ZoomOut

scaleStep：调焦倍数，当 command 为 focus 时，此项不为0

1 – 默认最小 放大/缩小 倍数

2 – 放大/缩小 2倍

以此类推…

1. 云台/相机 归位消息

topic：”/preset\_reset”

type：int32

proto：PresetResetMessage.proto

message PresetReset {

required int32 data=1;

}

data：赋值为1时，云台复位到初始位（例如：直视正前方），相机焦距回复到初始值（例如：0）

1. 车体自检消息

topic：”/robotSelfCheck”

type：int32

proto：RobotSelfCheckMessage.proto

message RobotSelfCheck {

required int32 data=1;

}

1. 一键返航（回仓充电）消息

topic：”/robotReturn”

type：int32

proto：RobotReturnMessage.proto

message RobotReturn {

required int32 data=1;

}

1. 看路摄像头图片消息

topic：”/road\_camera\_img”

type：int32/bytes

proto：RoadCameraImgMessage.proto

message RoadCameraImg {

required int32 number=1;

required bytes data=2;

}

data：图片bytes[]，需要转化为Google.Protobuf.ByteString后赋值使用

1. 检测摄像头图片消息

topic：”/detect\_camera\_img”

type：string/int32/bytes

proto：DetectCameraImgMessage.proto

message DetectCameraImg {

required string cameraType=1;

required int32 number=2;

required string imgType=3;

required bytes data=4;

}

imgType：预览图 – live

拍照图 – capture

1. 超声波消息

topic：”/UltraSound”

type：string/int32

proto：UltraSoundMessage.proto

message UltraSound {

required string name=1;

required int32 flag=2;

}

1. Request/Response消息

topic：”/req\_res”

type：string

proto：ReqResMessage.proto

message ReqRes {

required string reqType=1;

optional string data=2;

}

reqType：connectionDetect – 心跳请求

task – 下发任务请求

taskStop – 停止任务请求

taskCancel – 取消任务请求

data：当 reqType 为 task 时，值不为空；为 taskStop / taskCancel 时，值可为空；其他情况下值为1

1. 任务状态消息

topic：”/task\_status”

type：int32/float/string

proto：TaskStatusMessage.proto

message TaskStatus {

required int32 taskid=1;

required float start\_x=2;

required float start\_y=3;

required float start\_z=4;

required float end\_x=5;

required float end\_y=6;

required float end\_z=7;

required float dis=8;

required float speed=9;

required float dis\_left=10;

required float dis\_right=11;

required float road\_width=12;

required int32 control\_flag=13;

required string log=14;

}

1. 机器人位置消息

topic：”/robot\_pose”

type：int32/float

proto：RobotPoseMessage.proto

message RobotPose {

required int32 id=1;

required float x=2;

required float y=3;

required float z=4;

required float anglex=5;

required float angley=6;

required float anglez=7;

required float prob=8;

required float x1=9;

required float y1=10;

required float x2=11;

required float y2=12;

required float x3=13;

required float y3=14;

required float x4=15;

required float y4=16;

required int32 road\_line\_status=17;

required float robot\_road\_angle=18;

required float distance=19;

required float turn\_angle=20;

}

1. 检测结果消息

topic：”/detect\_result”

type：int32/float/bytes/string/bool

proto：DetectResultMessage.proto

message DetectResult {

required int32 camid=1;

required int32 picid=2;

required float x=3;

required float y=4;

required float z=5;

required bytes equipimage=6;

required bytes nameplates=7;

required string equipid=8;

required string result=9;

required bool success=10;

}

1. DebugSystem Log调试消息

topic：”/log”

type：string/int32

proto：LogMessage.proto

message Log {

required string name=1;

required int32 flag=2;

}

1. 上位机上线消息

topic：”/monitorOnLine”

type：string（data：”monitor\_is\_online”）

proto：MonitorOnLineMessage.proto

message MonitorOnLine {

required string data=1;

}

20. web发给路径规划

TaskPlanMessage.proto

message TaskPlan

{

required int32 taskId=1; #任务号

required string robotId=2; #机器人号

}

21.路径规划发给web

TaskPlanStatusMessage.proto

message TaskPlanStatus

{

required int32 status=1; 0：开始 1：结束

}