

## Basketball\_base\_serial

整个串口节点基于生产者消费者模型（中文 wiki <https://zh.wikipedia.org/wiki/%E7%94%9F%E4%BA%A7%E8%80%85%E6%B6%88%E8%B4%B9%E8%80%85%E9%97%AE%E9%A2%98>）。主要还是利用了 boost 库中 io\_service 和 serial\_port，其中 SerialPort 是星爷写的，家政，篮球都在用，用了好几年，非常稳定。SerialNode 是廖神基于篮球组的情况进行了一定的修改完成的。

注意，校验和由于历史原因是对 255 取模

basketball\_base\_serial/

./include

./SerialPort.h 串口类的声明，利用了 boost::asio::io\_service 模块;并设置了一参数，如文件 '/dev/ttyUSB0'，波特率 115200，注意使用时对 /dev/ttyUSB0 文件权限的修改，否则没有读写权限将无法正常运行

./SerialNode.h 串口节点的声明

./src/

./SerialPort.cpp 实现了读写回调函数

./SerialNode.cpp 具体实现串口节点，设置了读回调函数，并将下位机返回的数据发布于 /RecvData/1 话题上，此外，订阅 /robot\_cmd 话题，计算校验和，写入串口

./mainNode.cpp 运行串口节点

注意，如果运行中电脑断电或者其他异常情况的发生，会使得串口文件名发生变化，调试过程中应注意这个问题。如果开两个串口尤其要注意这个问题

## basketball\_bringup

为了便于启动放了一些 launch 文件。关于 roslaunch 详情见 <http://wiki.ros.org/roslaunch>

./launch

start\_robot.launch

启动 odom 节点，driver 节点，shovel 节点，shoot 节点，base\_serial 节点。

start\_robot\_show.launch

启动

rviz，basketball\_trajectory\_srv，basketball\_maker\_drawing，basketball\_direction\_maker 节点

注意因命名空间的原因产生话题名不一致的情况，可能需要使用 <remap>

## basketball\_description

由 solidworks 转换过来的 urdf 图，会在 rviz 中绘制出来

## basketball\_direction\_maker

在 rviz 中画出机器人上的方向标，便于前期调试

## basketball\_maker\_drawing

在 rviz 中画出篮球，定位柱，便于调试。

basketball\_draw\_maker 提供了 BasketballDrawMarker 类实现了画图的一些方法。

basketball\_maker\_srv 主要提供给外部接口，有三个话题 /moved\_trajectory,/mark\_ball,/mark\_cylinder

## basketball\_trajectory\_srv

通过 tf 接受消息后向/moved\_trajectory 话题发布消息在 rviz 中画出机器人的轨迹

## basketball\_msgs

放了一些 msg 和 srv

message : basketball\_position, mark\_post\_position, robot\_message, robot\_order, robot\_state

service : basketball\_nav\_srv, basketball\_position\_srv, basketball\_shoot\_srv, basketball\_shovel\_srv, ControlState, mark\_post\_position\_srv

## basketball\_driver

处理速度，订阅了两个话题 /cmd\_move 接受全局坐标系下的请求，/cmd\_move\_robot 接受机器人坐标系下的请求;类型 geometry\_msgs/Twist 发布了 /robot\_cmd 话题,消息类型 basketball\_msgs/robot\_message

## basketball\_odom

里程计节点。订阅串口节点发布的/RecvData/1 话题，发布话题/odom 同时发布 tf。/odom 是里程计，/base\_link 是机器人。清零里程计的功能放在了界面里，这个节点只会接收和发布里程数据

## basketball\_strage

scripts/

cylinder\_state\_pkg/ 提供给状态机找定位柱的接口

robot\_find\_pkg/ 提供给状态机找球的接口

robot\_shovel\_srv/ 提供给状态机控制铲子的接口

robot\_state\_class/ 提供给状态机一些简单常用的状态类

robot\_state\_pkg/ 提供获取当前姿态的接口

robot\_move\_pkg/ 提供状态机移动的接口

由于分工，我负责维护最后两个包。主要介绍 robot\_move\_pkg。

./config.py 机器人移动的一些配置参数，如速度，阈值等

./go\_along\_circle.py 机器人绕着圆圈走，主要是给/cmd\_move\_robot 话题下发速度实现的，有四种模式 mode 1: 向右前方转，mode 2: 向左前方转，mode 3: 向右后方转，mode 4: 向左后方转

./linear\_move.py 实现了边跑边转，并进行速度基于位移的插值，同时通过对角度和位移进行拟合来实现旋转与直线移动的同步

./low\_speed\_move.py 低速下的边跑边转，使用的是过去的老方案：向下发一固定速度，同时进行位移的计算，与目标距离进行比对，小于阈值速度不变，超出阈值速度给反注意！！如果速度大，阈值小，会造成机器人在目标点附近来回抖动

./move\_in\_robot.py 同上，唯一不同的是在机器人坐标下进行移动。

`./move_a_distance.py` 老方案，全局坐标系下进行 x, y 某一方向上直线移动，用的是老方案

`./turn_an angular.py` 机器人旋转一定的角度，原理同上，并提供给外部转到某一姿态的接口

`./interpolation_function/`

`./config_spline.py` 插值的一些参数

`./cubic_spline.py` 三次样条插值，但是被淘汰了

`./growth_curve.py` 基于细菌生长曲线的插值  $\frac{a}{1 + c \times e^{-bx}}$  a 代表移动过程中的最大速度;起始速度为  $a/(1+c)$ ; b 影响达到最大速度的时间，b 越大，越快达到最大速度。