

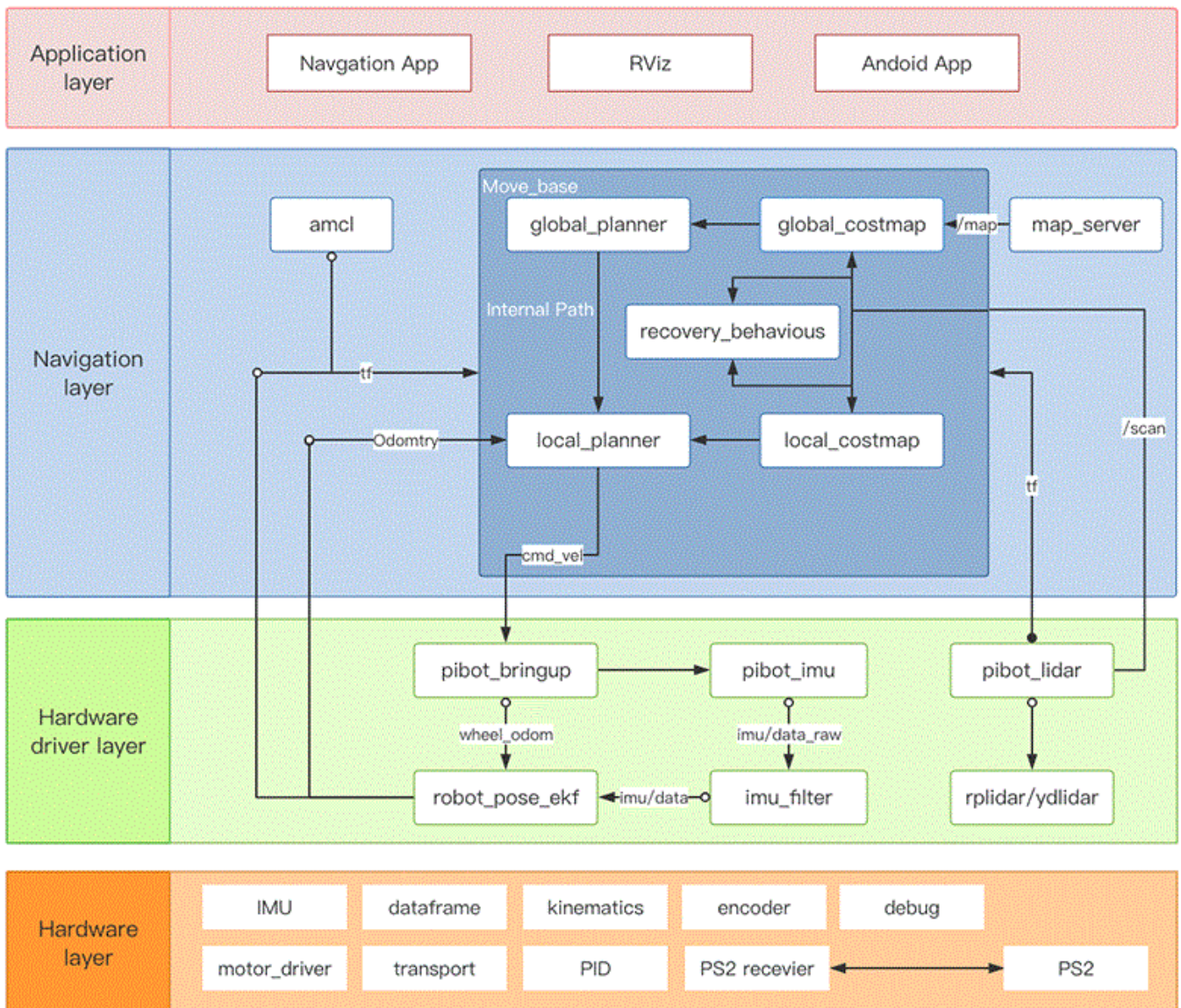
- 1. 概述
- 2. 软件框架
- 3. 下位机开发环境
 - 3.1 环境搭建
 - Arduino
 - STM32F1
 - STM32F4
 - 3.2 环境配置
 - Arduino
 - STM32F1
 - STM32F4
- 4. 代码分析
- 5. 参数配置
 - 5.1 默认参数
 - apollo
 - zeus
 - apolloX
 - hades
 - hera
 - hadesX
 - 5.2 配置参数
- 6. 电机方向和编码器方向软件调整

1. 概述

- PIBOT下位机支持多种主板，Arduino Mega2560、STM32F1及STM32F4等
- PIBOT下位机支持多种运动模型(差分、全向、麦克纳姆轮)，只需要调整编译参数即可
- PIBOT下位机支持不同的参数的机器人，执行设置相关参数即可

2. 软件框架

系统框架



橙色部分为下位机的功能模块

3. 下位机开发环境

- **Arduino Mega 2560**为主控单元，使用**Visual studio code+Platform IO**进行开发，支持**Windows**和**ubuntu**环境
- **STM32F1**为主控单元，使用**Keil**进行开发
- **STM32F4**为主控单元，**Ubuntu**下使用**Visual studio code**进行开发

3.1 环境搭建

Arduino

具体请参考[Visual Studio Code插件PlatformIO IDE开发Arduino](#)

STM32F1

具体请参考[PIBOT的STM32F1的环境配置与编译](#)

STM32F4

具体请参考[PIBOT的ubuntu下stm32 C/C++模版及配置编译](#)

3.2 环境配置

Arduino

使用[Visual studio code](#)打开附带下位代码文件夹，在[platformio.ini](#)修改相应模型以及使用的电机控制器

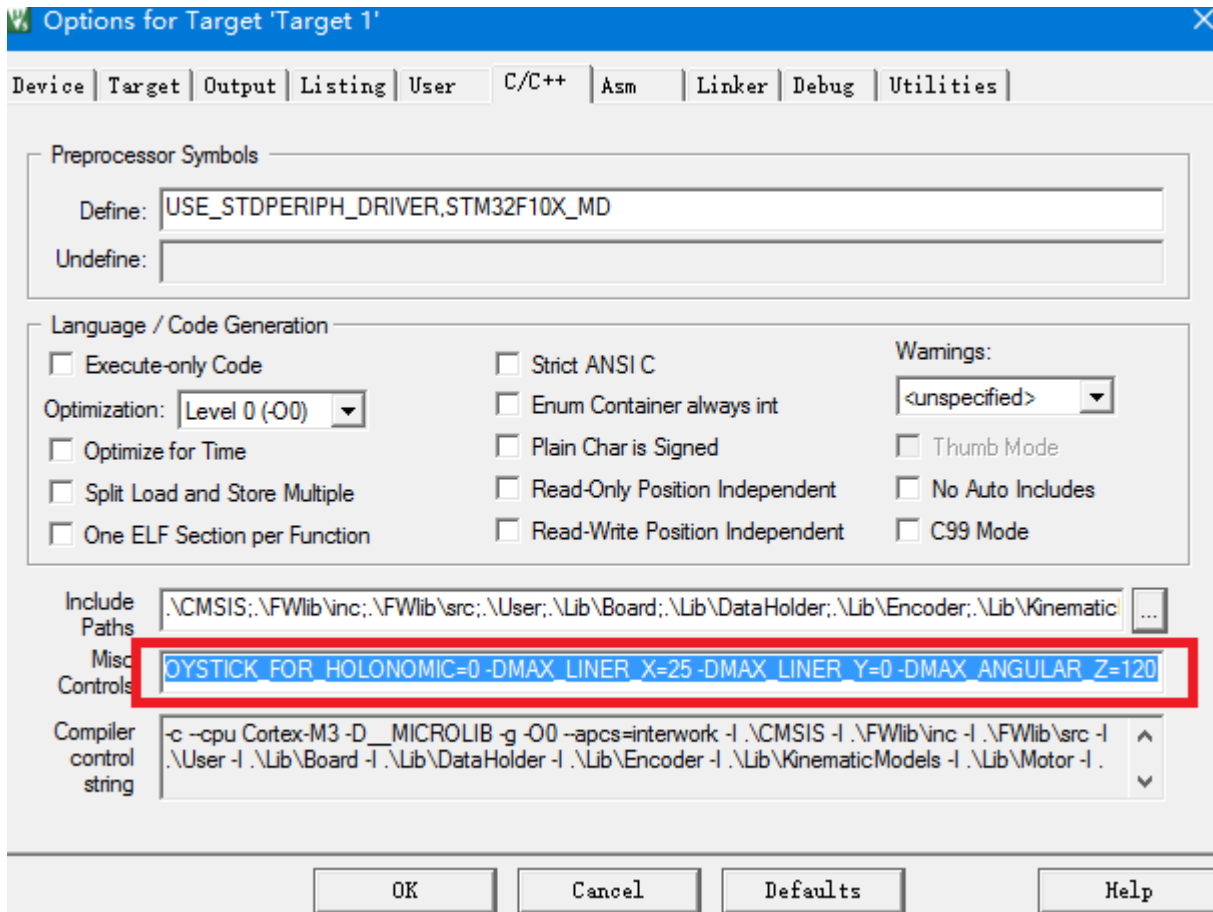
apollo

```
models =  
  -D ROBOT_MODEL=ROBOT_MODEL_DIFF  
  -D MOTOR_CONTROLLER=COMMON_CONTROLLER
```

zeus

```
models =  
  -D ROBOT_MODEL=ROBOT_OMNI_3  
  -D MOTOR_CONTROLLER=COMMON_CONTROLLER
```

STM32F1



同Arduino

STM32F4

在param.mk的修改相关配置，下图分别代表四种车型选择一种即可

```
#models
#DDEFS += -DROBOT_MODEL=ROBOT_DIFF_2WD -DMOTOR_CONTROLLER=COMMON_CONTROLLER
#DDEFS += -DROBOT_MODEL=ROBOT_DIFF_4WD -DMOTOR_CONTROLLER=COMMON_CONTROLLER
#DDEFS += -DROBOT_MODEL=ROBOT_OMNI_3 -DMOTOR_CONTROLLER=COMMON_CONTROLLER
DDEFS += -DROBOT_MODEL=ROBOT_MECANUM -DMOTOR_CONTROLLER=COMMON_CONTROLLER
```

4. 代码分析

以Arduino为例的代码分析具体请参考[ROS机器人底盘\(7\)-Firmware的代码分析\(1\)](#) [ROS机器人底盘\(8\)-Firmware的代码分析\(2\)](#) [ROS机器人底盘\(24\)-嵌入式部分框架设计与实现](#)

5. 参数配置

运动参数出厂时都内置在板子的EEPROM/FLASH中

5.1 默认参数

默认参数出厂会固化到板子,用户可以根据主机环境校准微调,可以参考PIBOT的控制及校准

apollo

- Arduino

The screenshot shows the 'Dynamic Reconfigure' window for the '/piBOT driver'. On the left, there is a 'Filter key:' field and buttons for 'Collapse all' and 'Expand all'. Below these is a tree view with 'piBOT driver' selected. A 'Refresh' button is at the bottom left. The main area contains a list of parameters with sliders and input boxes:

Parameter	Min	Max	Current Value
wheel_diameter	10	300	63
wheel_track	50	500	175
encoder_resolution	100	5000	1980
do_pid_interval	1	80	10
kp	0	10000	75
ki	0	32000	2500
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	200	40
max_v_liner_y	0	200	0
max_v_angular_z	0	200	150

- STM32F1/STM32F4

The screenshot shows the 'Dynamic Reconfigure' window for the '/piBOT driver' on an STM32F1/STM32F4 system. The interface is similar to the Arduino version but includes a message box at the bottom. The parameters and their values are:

Parameter	Min	Max	Current Value
wheel_diameter	10	300	63
wheel_track	50	500	175
encoder_resolution	100	5000	1980
do_pid_interval	1	80	10
kp	0	10000	250
ki	0	32000	2500
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	200
max_v_liner_x	0	200	40
max_v_liner_y	0	200	0
max_v_angular_z	0	200	150

(System message might be shown here when necessary)

zeus

- STM32F4

The screenshot shows a 'Dynamic Reconfigure' window for the '/pibot_driver'. It features a 'Filter key' field, 'Collapse all' and 'Expand all' buttons, and a 'Refresh' button. A list on the left shows 'pibot_driver' is selected. The main area displays a table of parameters with sliders and input boxes.

Parameter	Min	Max	Current Value
wheel_diameter	10	500	58
wheel_track	50	1000	230
encoder_resolution	100	10000	3960
do_pid_interval	1	80	10
kp	0	10000	320
ki	0	32000	2700
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	200	40
max_v_liner_y	0	200	40
max_v_angular_z	0	200	200

apolloX

- STM32F4

The screenshot shows a 'Dynamic Reconfigure' window for 'rqt_reconfigure...Param - rqt'. It features a 'Filter key' field, 'Collapse all' and 'Expand all' buttons, and a 'Refresh' button. A list on the left shows 'pibot_driver' is selected. The main area displays a table of parameters with sliders and input boxes.

Parameter	Min	Max	Current Value
wheel_diameter	10	500	96
wheel_track	50	1000	350
encoder_resolution	100	32000	3960
do_pid_interval	1	80	10
kp	0	10000	250
ki	0	32000	1200
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	500	50
max_v_liner_y	0	500	0
max_v_angular_z	0	2000	200
imu_type	0	255	69

hades

- STM32F4(轮子直径58MM)

The screenshot shows the 'rqt_reconfigure__Param - rqt' window for the '/piBOT_driver'. The 'Dynamic Reconfigure' panel is active, displaying a list of parameters for the 'piBOT_driver' node. The parameters are:

Parameter	Min	Max	Current Value
wheel_diameter	10	500	58
wheel_track	50	1000	470
encoder_resolution	100	10000	3960
do_pid_interval	1	80	10
kp	0	10000	320
ki	0	32000	2700
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	200	40
max_v_liner_y	0	200	40
max_v_angular_z	0	200	200

- STM32F4(轮子直径76MM)

The screenshot shows the 'rqt_reconfigure__Param - rqt' window for the '/piBOT_driver'. The 'Dynamic Reconfigure' panel is active, displaying a list of parameters for the 'piBOT_driver' node. The parameters are:

Parameter	Min	Max	Current Value
wheel_diameter	10	500	77
wheel_track	50	1000	470
encoder_resolution	100	32000	3960
do_pid_interval	1	80	10
kp	0	10000	320
ki	0	32000	2700
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	500	50
max_v_liner_y	0	500	50
max_v_angular_z	0	2000	250
imu_type	0	255	69

hera

- STM32F4

The screenshot shows the Dynamic Reconfigure window for the `/pibot_driver`. The interface includes a filter key, collapse/expand buttons, and a list of parameters with sliders and input fields. The current values are as follows:

Parameter	Min	Max	Current Value
wheel_diameter	10	500	82
wheel_track	50	1000	338
encoder_resolution	100	10000	3960
do_pid_interval	1	80	10
kp	0	10000	320
ki	0	32000	2700
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	200	40
max_v_liner_y	0	200	0
max_v_angular_z	0	200	200

Below the parameters, there is a message box: (System message might be shown here when necessary)

hadesX

- STM32F4

The screenshot shows the Dynamic Reconfigure window for the `/pibot_driver` in the hadesX environment. The interface includes a filter key, collapse/expand buttons, and a list of parameters with sliders and input fields. The current values are as follows:

Parameter	Min	Max	Current Value
wheel_diameter	10	500	150
wheel_track	50	1000	565
encoder_resolution	100	32000	3168
do_pid_interval	1	80	10
kp	0	10000	250
ki	0	32000	2750
kd	0	1000	0
ko	0	1000	10
cmd_last_time	0	1000	250
max_v_liner_x	0	500	50
max_v_liner_y	0	500	50
max_v_angular_z	0	2000	200
imu_type	0	255	69

Below the parameters, there is a message box: (System message might be shown here when necessary)

- wheel_disameter 轮子直径

- `wheel_track` `apollo`:轮距 `zeus`:轮子所在圆直径 `hades`:轮子矩形长宽之和 `hera` 左右轮距*系数
- `encoder_resolution` 轮子旋转一周编码器变化值的绝对值(一般为4 \times 减速比编码器分辨率, 如4 * 11 * 90 固件程序做了4倍频)
- `do_pi_interval` 计算pid的间隔时间, 固定值10
- `kp ki kd`
- `ko` 为一个系数, 实际PID参数为`kp/ko ki/ko kd/ko`
- `cmd_last_time` 命令激励的超时时间, 即超过该时间没有新的命令会机器人会停止
- `max_v_liner_x max_v_liner_y max_angular_z` 底层速度限制, 遥控器键盘或者导航层下发的速度会被该值限制
- `imu_type` 固定值69

5.2 配置参数

配置参数需要通过ROS上位机的界面配置

```
pibot_bringup
pibot_configure
```

或者

```
roslaunch pibot_bringup bringup.launch
rosvrun rqt_reconfigure rqt_reconfigure
```

6. 电机方向和编码器方向软件调整

移植PIBOT下位机或者由于电机或编码器接线问题, 导致电机控制时一直转。原因是给定方向的PWM值导致编码器一直反向变化, 所以需要调整电机线或者编码器接线, 通过设置软件编译参数也可以达到同样效果具体可以参见[关于运动控制方向的补充](#)