

智能节点与智能车通讯接口 使用说明

硬件连接

智能车转向与速度的控制采用 ARM Cortex-M3 内核的 STM32F103 系列控制器，通过接收节点发送过来的指令来控制转向和速度。节点与智能车的通讯采用 UART 接口，硬件连接方式如下：

STM32F103 的 A2 (TX2) 连接到节点 2 号引脚(RX);

STM32F103 的 A3 (RX2) 连接到节点 4 号引脚(TX);

组件接口

为了能正常控制智能车的行驶，需要通过节点发送控制指令到智能车控制器。在节点中，我们提供的相应的 API 接口来实现该功能，不同的接口由不同的组件提供，因此在使用接口时，必须在配置文件中包含相关组件。

主要组件：

- **CarC:** 提供了 Car 和 CarControl 两个接口，以及实现这些接口功能需要的其他组件；
- **CarP:** 包含 Car 和 CarControl 两个接口中所有命令的具体实现；

相关接口：

- **CarControl:** 主要有两个命令和事件，控制过程的启动 `start()` 与停止 `stop()`，以及对应的事件 `startDone()` 和 `stopDone()`。在与小车通讯之前，必须先调用 `start()` 命令，之后便可发送不同的控制指令给小车，当确定控制结束后，长时间不需要在控制小车时，调用 `stop()` 命令即可。若想再次发送控制指令，必须要再次调用 `start()` 命令；
- **Car:** 小车的控制指令包含在这个接口中，包括角度控制、前进速度、后退速度、停止。另外，出于对小车安全考虑，我们在小车底层驱动代码中设置一些保护措施，最大速度、最小速度、左偏最大角、右偏最大角、角度中间值。由于每辆车机械安装造成的差异，这些保护参数可能不同，因此，在该接口中还提供了初始化配置命令，主要有初始化最大速度、初始最小速度、初始化左偏最大角、初始化右偏最大角、初始化角度中间值；

接口命令具体说明：

接口名称	命令或事件	输入参数	返回值	功能	说明
CarControl	command error_t start()	无	SUCCESS 正常启动 FAIL 其他情况	控制指令 启动命令	正常启动后会触发事件 event void startDone()
	command error_t stop()	无	SUCCESS 正常 FAIL 其他情况	停止控制指令	正常停止后会触发事件 event void stopDone()
	Event void startDone(error_t error)	start () 命令传入的参数	SUCCESS 正常 FAIL 其他情况	启动完成应该执行的操作	事件的实现 用户自行定义
	event void stopDone(error_t error)	stop () 命令传入的参数	SUCCESS 正常 FAIL 其他情况	停止完成应该执行的操作	事件的实现 用户自行定义
Car	command error_t Angle(uint16_t value);	舵机转角值	SUCCESS 正常 FAIL 其他情况	转向角度	输入范围内的一个角度值（1800，3700），舵机即转到相应的角度
	command error_t Forward(uint16_t value);	前进速度值	SUCCESS 正常 FAIL 其他情况	前进速度	输入范围内的一个速度值（300,600），电机即转到相应的速度
	command error_t Back(uint16_t value);	后退速度值	SUCCESS 正常 FAIL 其他情况	后退速度	输入范围内的一个速度值（300,600），电机即转到相应的速度
	command error_t Pause();	无	SUCCESS 正常	电机停止	电机停止转动

			FAIL 其他情况		
	command error_t InitMaxSpeed(uint16_t value);	最大速度	SUCCESS 正常 FAIL 其他情况	初始化最大速度	最大速度，默认值 600，可根据实际需要自行修改
	command error_t InitMinSpeed(uint16_t value);	最小速度	SUCCESS 正常 FAIL 其他情况	初始化最小速度	最大速度，默认值 300，可根据实际需要自行修改
	command error_t InitLeftServo(uint16_t value);	左偏最大值	SUCCESS 正常 FAIL 其他情况	初始化左偏最大值	左偏最大值 3700，可根据舵机安装角度调整
	command error_t InitRightServo(uint16_t value);	右偏最大值	SUCCESS 正常 FAIL 其他情况	初始化右偏最大值	右偏最大值 1800，可根据舵机安装角度调整
	command error_t InitMidServo(uint16_t value);	角度中间值	SUCCESS 正常 FAIL 其他情况	初始化角度中间值	舵机中间值 1800，可根据舵机安装角度调整

使用例程

- 1、例程提供三个文件，分别是配置组件 CarC.nc、模块组件 CarP.nc、接口 Car.nc。要正常使用 Car 接口提供的命令，首先第一步是要将这三个文件都添加到源文件目录下。

- 2、在源文件顶层目录文件中添加组件声明和导通连接：

例程代码：

```
components CarSendC as App;
components CarC;
```

```
App.Car -> CarC.Car;
```

```
App.CarControl -> CarC.CarControl;
```

其中 CarSendC 是用作测试的组件，在使用中只要替换自己的组件即可。

- 3、在 CarSendC 测试组件中要声明使用接口说明：

例程代码：

```
uses interface Car;
```

```
uses interface SplitControl as CarControl;
```

- 4、在 CarSendC 测试组件中需要先调用 CarControl.start(); 命令才能进行后面指令发送，否则发送的指令无效。当确定不需要在对小车做任何操作时，可以调用 CarControl.stop(); 当再次控制小车时，需要重新调用 CarControl.start(); 命令

例程代码：

```
call Car.Forward(300); //前进速度 300  
call Car.Angle(2850); //舵机角度值 2850
```

- 5、第一次发送命令需要先发送初始化配置命令，命令格式与其他命令相同，若不发送默认配置，则使用底层驱动自带的默认配置！

例程代码：

```
call Car.InitMaxSpeed(600); //初始化最大速度为 600  
call Car.InitRightServo(1800); //初始化右偏最大值为 1800
```