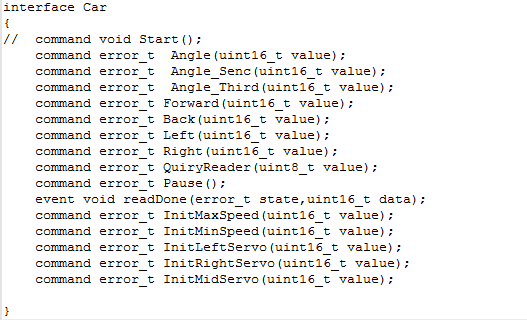
小车控制说明：

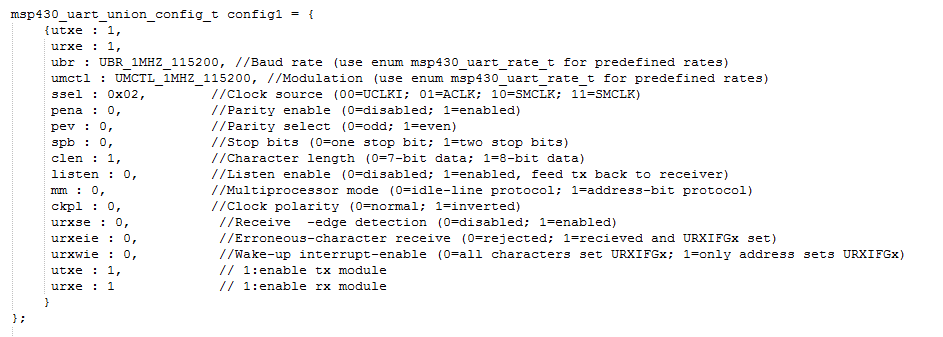
小车模块以及接口示意



其中烧录主模块与小车模块通过接口Car连接，对于该接口，有下列对应函数：

这些需要在Car的模块中实现，Car模块通过串口与小车上电路连接，HplMsp430Usart0C是串口模块与CarC通过两个接口连接，通过调用接口HplMsp430Usart中的tx函数来发送操控小车的信息，注意tx一次发送一个byte数据，因此发送7个byte的命令的时候，每次发完需要通过函数isTxEmpty来检查是否空闲，再去发送下一个byte。HplMsp430UsartInterrupts主要是用来调试串口信息发送的，可以调用它的txDone函数来查看信息是否发送完成。

Msp430Uart0C模块与CarC通过接口Resource连接（http://www.btnode.ethz.ch/static\_docs/tinyos-2.x/nesdoc/eyesIFXv2/ihtml/tos.interfaces.Resource.html）。该模块主要用来获取对串口的操作权限。每次获取小车的操作命令后，通过调用该模块的request函数获取资源，成功后会启动granted事件，因此需要编写该事件来写入串口信息，写入信息需要调用上面说到的接口HplMsp430Usart，先通过setModeUart写入配置信息（如下），然后enableUart，然后设置控制器状态U0CTL &= ~SYNC;最后通过上面tx函数来实现写入小车命令。



对于HplMsp430GeneralIOC，需要将Car中定义的HplMsp430GeneralIO接口连接到HplMsp430GeneralIOC的Port20。(类似Car.P20-> GIO.Port20)

整体函数调用的流程：在BlinkToRadio中AMReceiver收到Radio接收到的命令，然后拿出对应的数据，决定call接口Car中对应的小车命令，在Car中实现的小车命令需要调用Resource接口的request函数去获取串口资源从而触发事件granted，然后自己编写的granted事件先配置串口信息，然后通过tx函数写入控制命令。最后写完调用Resource接口的release函数释放资源。

手柄控制说明：



烧录模块BlinkToRadioC连接ButtonC模块，ButtonC模块通过定义6个HplMspGeneralIO接口与HplMspGeneralIOC的6个端口相连，对于ButtonC模块定义端口Button，提供一下命令调用以及调用完成的事件（事件默认定义就可以default event void Button.pinvalueADone(error\_t error){}），具体时间操作需要在BlinkToRadioC中改写。



对于start命令作为初始化所有button端口命令，先调用HplMspGeneralIO接口的clr函数清空数据，然后调用makeInput设置端口方向是输入。

对于每一个获取button输入值的命令pinvalueA，调用HplMspGeneralIO的get函数获取按钮的输入值。然后会signal一个完成的事件，这个事件被BlinkToRadio模块检测到然后得到ButtonA是否按下，从而在BlinkToRadio的无线发射部分将命令发送出去。

（<http://www.btnode.ethz.ch/static_docs/tinyos-2.x/nesdoc/eyesIFXv2/ihtml/tos.chips.msp430.pins.HplMsp430GeneralIO.html>）

Button操作流程：BlinkToRadio模块调用命令pinvalueA去获取Button中连接A端口的按钮是否被按下，从而触发事件pinvalueADone，BlinkToRadio在该事件中记录A端口的输入情况，最终在利用AMSender将该数据打包发送出去。

摇杆操作：

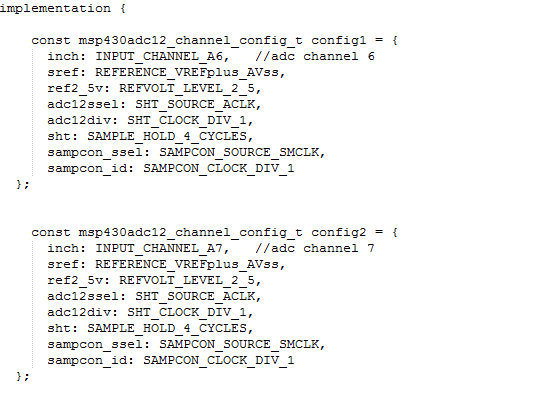
摇杆定义成JoyStick的通用类，因此在BlinkToRadio中定义两个JoyStick来读取摇杆在X,Y轴上的输入。BlinkToRadio通过接口Read<uint16\_t>中的read命令来读取摇杆输入。在JoyStick中定义两个组件AdcReadClientC作为X，Y轴ADC的输入。在JoyStick中定义模块JoyStickP，该模块提供两个接口AdcConfigure<const msp430adc12\_channel\_config\_t\*>，作为X,Y轴的配置，将AdcReadClientC的对应接口AdcConfigure连接到JoyStickP的对应接口上。然后在JoyStickP中通过定义命令

async command const msp430adc12\_channel\_config\_t\* AdcConfigure1.getConfiguration() {

return &config1;

}

配置adc信息



（<http://www.btnode.ethz.ch/static_docs/tinyos-2.x/nesdoc/eyesIFXv2/chtml/tos.chips.msp430.adc12.AdcReadClientC.html>）

tinyos-main/doc/txt/tep101.txt有对adc模块的说明，可以参考一下。

因此摇杆部分是BlinkToRadio通过接口Read<uint16\_t>来连接JoyStick中的AdcReadClientC，AdcClientC的AdcConfigure接口连接到JoyStickP模块的AdcConfigure，在JoyStickP中定义了命令getConfiguration配置对应adc。