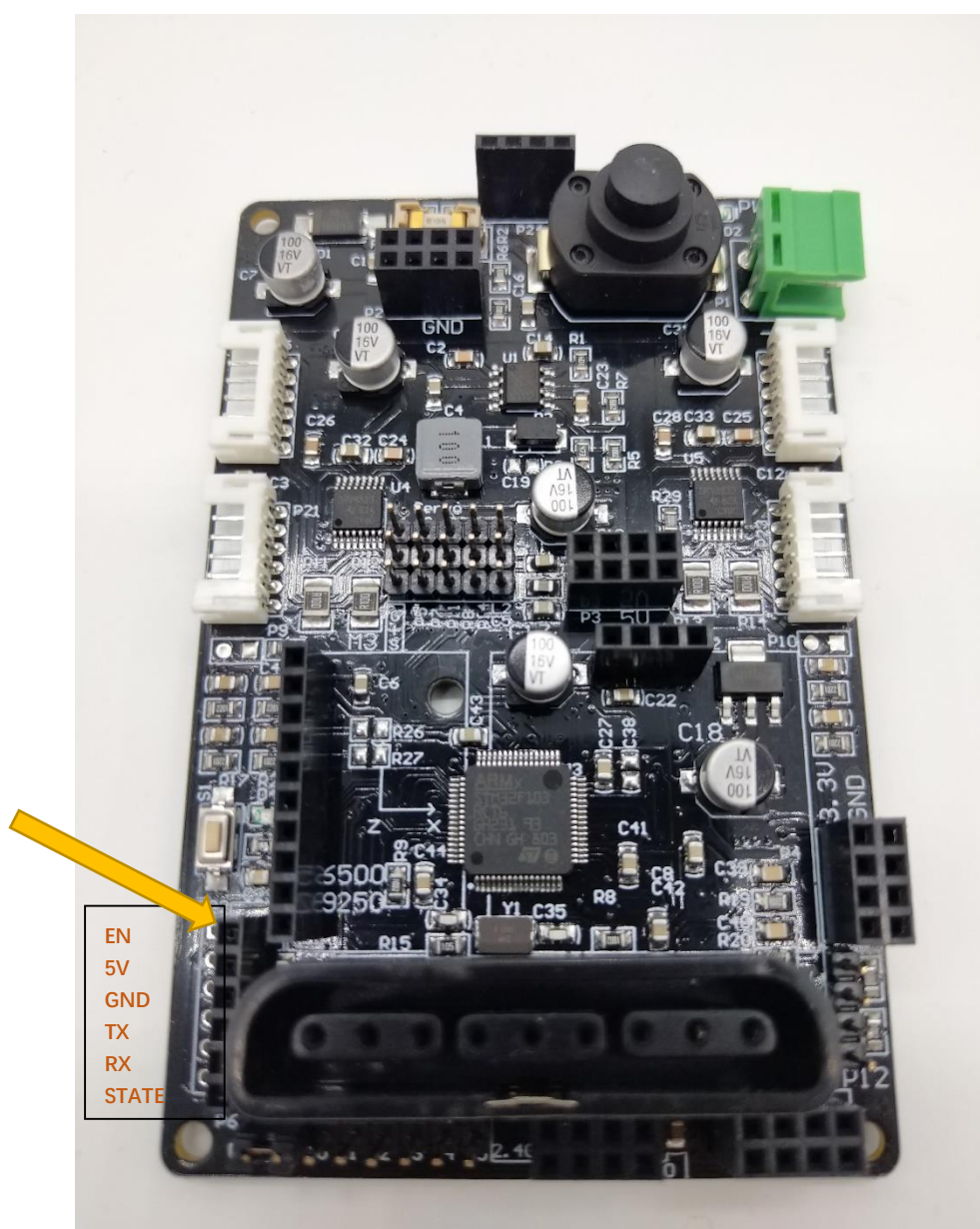


串口控制 应用笔记

概述

主板上引出了串口 2，默认焊接了 6pin 的座子，可以用于串口蓝牙模块的通讯。串口蓝牙模块接受手机 app 发送来的指令数据后通过串口输入 STM32 控制器中，STM32 接收后控制轮子运动。如果不使用蓝牙，可以将这套机制用于其他用途，例如使用其他串口设备如 arduino，pc，树莓派等设备来控制麦车的运动。

管脚定义



排母从上到下的管脚分别是

EN(蓝牙使能，蓝牙专用，默认不接，引出到旁边空焊盘但是未连接至单片机)
5V： 5v 对外供电（若外部设备自带供电则请勿接此线以免烧坏）
GND：地线，必须接否则无法通讯
TX： 外部设备或者蓝牙接收端，已经连至 STM32 的 uart2 的 Rx 管脚(PA3)
RX： 外部设备或者蓝牙发送端，已经连至 STM32 的 uart2 的 Tx 管脚(PA2)
STATE(蓝牙状态输出，蓝牙专用，默认不接，引出到旁边空焊盘但是未连接至单片机)

注：1、进入蓝牙模式需要用跳线帽或者杜邦线短接 PC4 和 PC10.

2、外部模块的 Rx 与 Tx 此处无需交叉相连，此文档的说明以及板子背面的丝印的 Rx Tx 均是针对外部设备而言的

协议说明

在出厂默认固件中，蓝牙模块发送数据至单片机遵循下列的协议，数据格式如下：

数据内容	数据类型及长度	数值说明
帧头	uint8_t	0xff 固定值
协议版本	uint8_t	0xfe 固定值
数据类型	uint8_t	0x01 固定值，表示速度
X 方向速度，右边为正	int16_t (little endian)	速度值， -1024~1023
Y 方向速度，前进为正	int16_t (little endian)	速度值， -1024~1023
R 方向速度，顺时针为正	int16_t (little endian)	速度值， -1024~1023
帧尾	uint8_t	0x00 固定值

注：1、little endian 指小端序，即先发送低字节再发送高字节，arm 内核为小端序，直接按顺序发即可，大端序的平台需要倒字节序

2、发送速度值大小无限制，但是内部处理过程中只有-1000~1000 的数据才是有效的，大于这个范围以-1000 或 1000 算，实际使用中发送-1024~1023 比较方便。

示例：数据帧格式： 0xff 0xfe 0x00 [15:0] (x 方向速度，两字节) [15:0] (y 方向速度，两字节) [15:0] (r 自旋速度，两字节) 0x00 共 10byte

数据发送波特率为 9600bps，8 位数据，无校验位，1 停止位。

编程示例

知道了数据协议后只要按照该协议发送所需的速度值给主板，即可控制电机以设定的速度运动。

附 python demo，适用于 windows/linux/树莓派等平台。需要安装 pyserial 库：

```
import serial
import struct
import os
import sys
```

```
fd=serial.Serial("COM3",9600,timeout=100)
#‘COM3’需要根据所在平台以及实际串口号修改，linux 下 usb 串口一般为/dev/ttyUSBx,
树莓派自带串口一般是/dev/ttyAMA0

head=b'\xff\xfe\x01'
tail=b'\x00'

while True:
    speedx=900#设置 x 轴速度
    speedy=-800#设置 y 轴速度
    speedr=999#设置 x 轴速度
    package=struct.pack('<3s3hs',head,speedx,speedy,speedr,tail)#利用 struct 模块进行
    编码
    fd.write(package)//输出速度
```

注：默认固件带有蓝牙模式下的掉线检测，即一段时间（200ms）未收到串口数据将会设置电机速度为 0，故需要一直发送速度数据，也可以在 config.h 中设置 #define OFFLINE_DECT 0 关闭或者修改 #define TIMEOUT_TH 200 设置超时时间。