



4WS4WD 使用手册

最后更新日期：2020 年 8 月 17 日

岑汝平

更新日志:

20200817（岑汝平、舒康）

1. 查找小车硬件厂家和购买链接
2. ROS 测试节点

20200813（岑汝平、舒康）

1. 车硬件结构
2. 小车通讯协议

1、小车硬件结构



图 1-1 小车实物图

如图 1-1 所示，小车主控板负责控制底盘上的电机、超声波模块及其他 IO 设备，如图 1-2 所示，小车包含 4 个行进电机和 4 个转向电机，8 个电机均采用 CAN 总线通讯协议，小车行进电机采用的是安普斯的轮毂电机、转向电机采用的是深圳飞特模型公司的 SMCL 系列电机。轮毂电机资料可查看附件“LSDB 系列伺服轮毂驱动器说明书 3V2-20200801.pdf”，转向电机资料可查看附件“SMCL 系列舵机资料包.zip”。

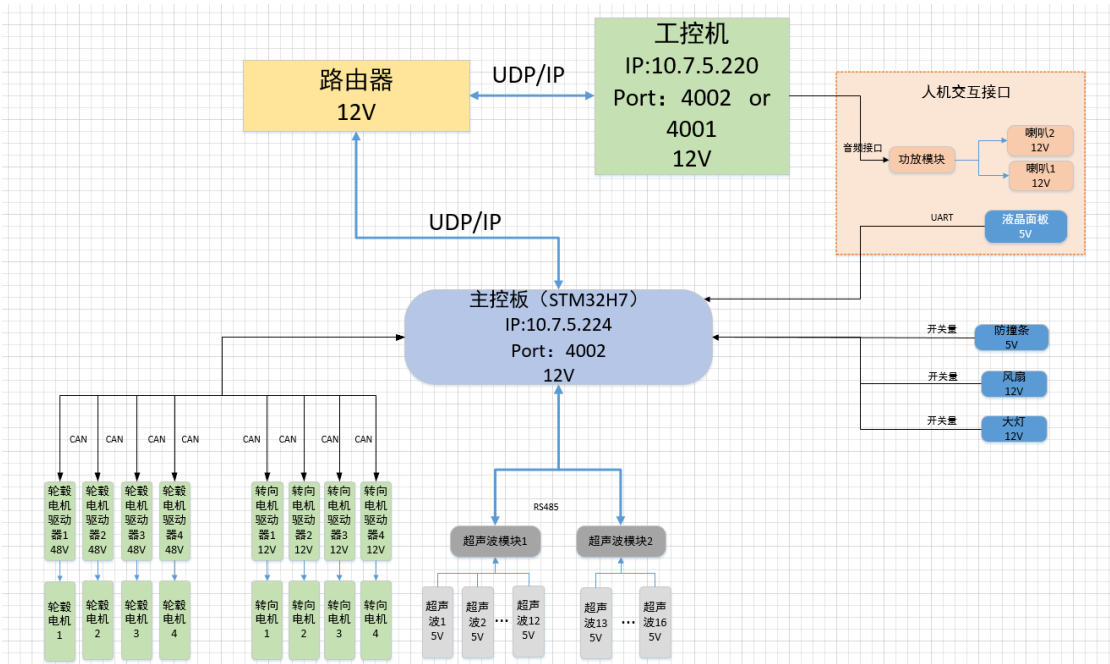


图 1-1 小车硬件结构框图

小车全车周围配备了 16 路超声波模块，超声波通过 CAN 总线与主控板相连，其余 IO 量，如大车车灯和风扇由主控板通过控制继电器实现，防撞条通过 IO 端口连接到主控制板。

小车的局域网端处于 10.7.5.1 网段，主控板的 IP 为 10.7.5.224，工控机 IP 为 10.7.5.220。小车上电以后主控板会向 10.7.5.220:4002 和 4001 端口发送数据，因此在与主控板建立通讯的时候必须将电脑的 IP 地址设置为 10.7.5.220，通讯方式为 UDP 通讯。

1.1 远程访问小车

小车开机以后会对外产生一个名为“**T2 001**”的 wifi，使用笔记本连接上这个 wifi(密码为: gaoxinxing),该 wifi 会给电脑分配一个 192.168.201.1 这个网段的地址，192.168.201.1 则是小车工控机的 IP。

通过 ssh 命令 “ssh gosunyun@192.168.201.1” 密码为:gsy2017 即可远程连接上该工控机。

2、小车通讯协议

小车的帧协议采用帧头+数据+校验的方式

2.1 帧头的结构

```
struct _TNBHead
{
    u16 u16MagicCode; /* "GS" */
    u16 u16Size; /* 总字节数, 包括帧头与数据 */
    u16 u16CmdId; /* 指令 ID */
    u16 u16CRC; /* CRC 校验 */
};
```

u16 MagicCode	“GS”
u16 Size	帧头+数据总字节数
u16 CmdId	指令: 例如 0x7426 代表运动控制帧
u16 CRC	帧数据的 CRC16 校验, 见附 1

帧头部分一共占用 8 个 Byte, 针对不同的命令需要设置不同的 **u16CmdId**。发送数据的时候采用小端模式（低位在前）

2.2 运动控制指令

下发控制指令的时候 帧头中的 CMD_ID 固定为 **0x7426**

/* PC->MCU(S0,S1): 运动控制命令 */

```
struct Move_Ctrl
{
    TNBHead head;
    u8 Cmd; /* 运动控制命令 */
    u16 Spd_l; /* 本体线速度 */
    u16 Spd_a; /* 本体角速度 */
};
```

成员说明:

head	见 1.1 帧头结构
Cmd	见运动控制指令枚举
Spd_l	Bit15: 符号位, 1 代表负, 0 代表正 Bit14~bit0: 本体线速度的绝对值大小 单位: mm/s 本体向前运动为正, 向后运动为负
Spd_a	Bit15: 符号位, 1 代表负, 0 代表正 Bit14~bit0: 本体角速度的绝对值大小 单位: $0.1^{\circ}/s$ 本体朝向逆时针偏转为正, 顺时针为负。

运动控制指令枚举:

```
typedef enum
{
    BD_STOP = 0,      //停止, 此命令停止过程会递减速度最终停止下来, 有一定缓冲。
    BD_RUN,           //运动, 表示运动, 方向和速度由 Spd_l, Spd_a 决定
    BD_BREAK,         //刹车, 与 BS_STOP 相比, 没有任何缓冲, 直接停止。
} BD_CMD;
```

注: 关于停止, 有三种方式:

- 1) BD_ST: 此方式下, 忽略速度参数, 带缓冲滑行尽快的停止下来;
- 2) BD_RUN: 此方式下, 若速度参数给 0, 由速度环将速度调节到 0;
- 3) BD_BREAK: 此方式, 将忽略参数, 忽略速度环调节, 强行将速度直接置 0;

控制命令返回数据格式, 返回数据帧头中的 CMD_ID 固定为 **0x7426**

控制指令反馈:

```
/* MCU(s0,s1)->PC: 运动速度上报 */
struct Move_ST
{
    TNBHead head;
    u16 Spd_l; /* 本体当前线速度 */
    u16 Spd_a; /* 本体当前角速度 */
};
```

2.3 里程、角度查询

下发里程、角度查询指令的时候 帧头中的 CMD_ID 固定为 **0x7600**, 返回数据帧头中的 CMD_ID 与下发的 CMD_ID 相同。

查询命令

```
/* PC->MCU(s1): 位置, 角度查询 */
struct Dist_Get
```

```

    {
        TNBHead head;
    };
反馈:
/* MCU(S1)->PC: 位置, 角度反馈 */
struct Dist_Rsp
{
    TNBHead head;
    s32 speedI; /* 本体总运动里程 单位 mm 向前为正, 向后为负 */
    s32 angleI; /*本地总转动角度, 单位 0.1 度, 逆时针为正, 顺时针为负
                逆时针 0~1800 顺时针 0~-1800*/
};
注:角度和里程都是从机器人上电开始为基准统计, 关机后, 下次开机将重新统计。

```

2.3 超声波查询

下发超声波查询指令的时候 帧头中的 CMD_ID 固定为 **0x7601**, 返回数据帧头中的 CMD_ID 与下发的 CMD_ID 相同。

查询命令

```

/* PC->MCU(s1): 获取超声波数据 */
struct Ultra_Get
{
    TNBHead head;
};
反馈:
/* MCU(S1)->PC: 反馈超声波数据 */
struct Ultra_Rsp
{
    TNBHead head;
    u16 ultra_data[8]; /*8 个超声波数据, 单位 mm, 0—7 分别是: 左, 左前,
                        前, 右前, 右, 右后, 后, 左后。当数据为 0xFFFF 时, 表示对应的超声波
                        异常了 */
};

```

2.4 防撞条查询

帧头中的 CMD_ID 固定为 **0x7602**, 返回数据帧头中的 CMD_ID 与下发的 CMD_ID 相同。

查询命令

```

/* PC->MCU(s1): 获取超声波数据 */

```

```

struct AnticollisionBar_Get
{
    TNBHead head;
};
反馈:
/* MCU(S1)->PC:  反馈超声波数据 */
struct AnticollisionBar_Rsp
{
    TNBHead head;
    U8 status; /*防撞条状态 0: 未撞击 1 撞击*/
};

```

2.5 声紧急制动开关控制

帧头中的CMD_ID 固定为 **0x7603**, 返回数据帧头中的CMD_ID 与下发的CMD_ID 相同。

设置命令

```

/* PC->MCU(s1):  超声近距底层急停设置 */
struct UltraBreak_Set
{
    TNBHead head;
    U8 cmd; /*0-关闭,  1-开启*/
};
反馈:
/* MCU(S1)->PC:  对设置命令的反馈 */
struct UltraBreak_Rsp
{
    TNBHead head;
    U8 status; /* 超声紧急自动制动功能开关状态,  0-关闭,  1-开启 */
};

```

2.6 撞条紧急制动开关控制

帧头中的CMD_ID 固定为 **0x7604**, 返回数据帧头中的CMD_ID 与下发的CMD_ID 相同。

设置命令

```

/* PC->MCU(s1):  防撞条底层急停设置 */
struct Anticollision_Set
{
    TNBHead head;
};

```



```

    U8 cmd; /* 0-关闭, 1-开启 */
};
反馈:
/* MCU(S1)->PC: 对设置命令的反馈 */
struct Anticollision_Rsp
{
    TNBHead head;
    U8 status; /* 防撞条紧急自动制动功能开关状态 0-关闭, 1-开启 */
};

```

2.7 驱动器异常反馈

帧头中的 CMD_ID 固定为 **0x7605** 返回数据帧头中的 CMD_ID 与下发的 CMD_ID 相同。

查询命令

```

/* PC->MCU(s1): 查询驱动器异常状态 */
struct MotorDrvSt_Get
{
    TNBHead head;
    U8 Driver; /* 0-左侧电机驱动器, 1-右侧电机驱动器 */
};

```

反馈:

```

/* MCU(S1)->PC: 反馈驱动器异常状态(异常时主动上报) */
struct MotorDrvSt_Rsp
{
    TNBHead head;
    U8 Driver; /* 0-左侧电机驱动器, 1-右侧电机驱动器 */
    U32 err; /* 驱动器异常状态值 */
};

```

驱动器异常状态值表

Bit 位	异常内容
0	过流(对驱动器本身而言)
1	过压
2	编码器异常
3	欠压
4	过载(对电机而言)
5~31	预留

3、小车硬件清单与厂家

1) 防撞条

<https://item.taobao.com/item.htm?spm=a1z0d.6639537.1997196601.26.55a77484IC00S0&id=606690470476>

2) 超声波模块

<https://item.taobao.com/item.htm?spm=a1z0d.6639537.1997196601.48.55a77484IC00S0&id=541503665657>

3) 大扭力舵机

<https://item.taobao.com/item.htm?spm=a1z0d.6639537.1997196601.4.62957484PVYz9y&id=540489504846>

4) 轮毂电机

安普斯

5) 工控机 12V DC 2.5mm

<https://item.taobao.com/item.htm?spm=a230r.1.14.54.3e6d49aaevQhZR&id=534931094408&ns=1&abbucket=8#detail>

6) 电池

深圳锂神科技

7) 风扇 12V

8) 路由器

9) 点火开关

10) 启动按钮