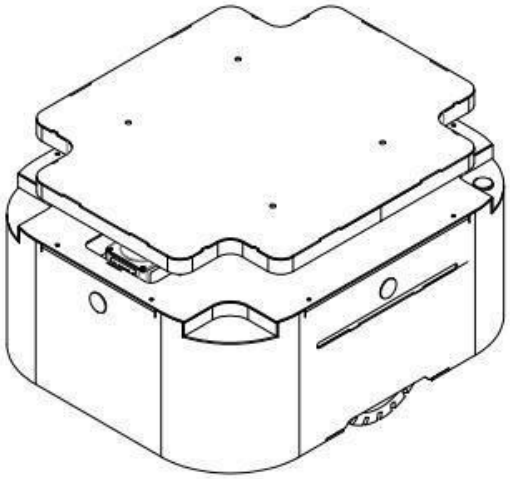
DT-01系列底盘通讯协议



浙江智澜科技有限公司
地址：浙江省杭州市余杭区人工智能小镇广东省佛山市南海区东软华南**IT**创业园
邮箱：**zlkjlin@163.com**

目录

一. 串口设置

数据位	波特率	校验位	停止位	通讯频率
8bit	115200	无	1bit	50HZ

二. 数据格式

内容：	Head	Len	Type	Cmd	Num	Data	Theck
字节：	2	1	1	1	1	2n	2

Head：协议头，固定2个字节： 0xDE 0xED

Len： 帧长度，1 个字节，从Head到Check的总长度

Type：产品类型，1个字节

Cmd：命令类型，1 个字节

Num：数据个数，1个字节

Data：实际数据，长度 0-254 之间,2的倍数

Check：校验码，2字节。从Head到Data的和（和校验：Check=Head+Len+Type+Cmd+Num+Data）

三. 协议指令表

命令	功能表
CMD	上位机下发指令解析
0X00	查询指令,发送该指令，串口返回底盘参数信息.
0X01	开启或关闭机器人20ms数据上传命令.
0X02	机器人速度设置,可通过该指令控制底盘移动
0X03	车子急停功能设置
0X04	回充指令，开启或者关闭回充（无回充模块，请忽视该条指令）
0X05	查询机器人详细故障信息
0X06	修改芯片储存的硬件数据
0X07	防撞杆撞击解除，机器人将解除急停
DT-01上传指令解析	
0X80	机器人上传底盘状态指令（ 该指令仅供查询时使用），包括（机器人固件号、电机功率、电机减速比、编码器线数、轴距、轮子直径、电池容量）
0X81	机器人每20ms返回指令包括（线速度，角速度、轮子速度、陀螺仪、编码器增量、电压、底盘状态、灯条状态），对应0X01指令，开启底盘数据上传。底盘上传模式分两种

	1：里程反馈模式：线速度、角速度、陀螺仪加速度（有IMU时）、陀螺仪角速度（有IMU时）、电压、底盘状态、灯条状态等
	2:速度反馈模式：左右轮速度、左右轮编码器增量值、陀螺仪加速度（有IMU时）、陀螺仪角速度（有IMU时）、电压、底盘状态、灯条状态等。
0X85	查询产品故障数据，机器人将返回包括电机故障数据、电压故障数据、遥控器接收机故障数据等数据
0X86	返回修改成功后芯片储存的硬件数据，机器人将返回修改成功后芯片储存的硬件数据包括固件号数据、电机功率数据、电机减速比数据、编码器线数数据、左右轮距、轮子直径、电池容量、电池电压数据、电机最大转速、左电机系数、右电机系数、发货日期等数据。
0X87	撞击解除状态返回，机器人将返回当前急停状态

四.上位机下发指令案例说明

4.1 Cmd：0x00　　(查询数据0x00)

(查询芯片储存的硬件数据0x00)

描述：查询芯片储存的硬件数据，机器人将返回芯片储存的硬件数据包括固件号数据、电机功率数据、电机减速比数据、编码器线数数据、左右轮距、轮子直径、电池容量、电池电压数据、电机最大转速、左电机系数、右电机系数、发货日期等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x0A 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x00

Num：数据个数，0x01

Data：实际数据,0x00

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 0A 02 00 01 00 00 D8 01

4.2 Cmd：0x01　　(20ms底盘上传数据指令)

描述：开启或关闭机器人20ms上传数据；

Head：固定帧头，0xDE 0xED

Len： 帧长度， 0x0A

Type：产品类型，0x02

Cmd：命令类型，0x01

Num：数据个数，0x01

Data：为0X0000时关闭20ms上传，

为0X0001是开启20ms上传里程反馈模式；

为0X0002是开启20ms上传速度反馈模式；

Check：校验码，2字节。从Head到 Data的和

命令示例：ED DE 0A 02 01 01 00 00 D9 01（关闭速度上传反馈模式）

ED DE 0A 02 01 01 01 00 DA 01（开启线速度，角速度反馈模式）

ED DE 0A 02 01 01 02 00 DB 01（开启速度反馈模式）

4.3 Cmd：0x02　　(底盘速度指令设置)

描述：底盘速度设置，分两种模式，模式1位X轴速度和Z轴速度设置（为线速度模式），模式2位左轮　速度设置和右轮速度设置（轮速控制模式）。

Head：固定帧头：0xDE 0xED

Len：　帧长度，　0x10

Type：产品类型，0x02

Cmd：命令类型，0x02

Num：数据个数，0x04

Data0：为0x0000时为线速设置模式即设置的是X轴速度和Z轴速度。

Data1：Vx即X轴速度。(int16)(mm/s)

Data2、Data3：Vz即Z轴速度。(float)(rad/s)

Data0 ：为0x0001时为轮速设置模式即设置的是左轮速度和右轮速度。

Data1：左轮速度。(int16)(mm/s)

Data2：右轮速度。(int16)(mm/s)

Data3：0x0000。　占位

Check：校验码，2字节。从Head到Data的和

命令示例：ED DE 10 02 02 04 00 00 C8 00 00 00 00 00 AB 02（线速设置模式Vx=200,Vz=0）

ED DE 10 02 02 04 01 00 C8 00 C1 00 00 00 6D 03（轮速设置模式LSp=200,RSp=193）

4.4 Cmd：0x03　　(底盘急停设置)

描述：设置机器人急停等设置。Head：固定帧

头： 0xDE 0xED

Len： 帧长度，0x0A

Type：产品类型，0x02

Cmd：命令类型，0x03

Num：数据个数，0x01

Data0：为急停控制数据，0：不急停，1：急停。

Check：校验码，2字节。从Head到Data的和

命令示例：ED DE 0A 02 03 01 00 00 DB 01

4.5 Cmd：04 (开启自动充电模式0x04)

描述：设置机器人开启自动充电模式；

Head：协议头，固定2个字节： 0xDE 0xED

Len： 帧长度，0x0A

Type：产品类型，0x02

Cmd：命令类型，0x04

Num：数据个数，0x01

Data：为0X0000时关闭回充，

为0X0001是开启回充；

Check：校验码，2字节。从Head到 Data的和

命令示例：ED DE 0A 02 04 01 00 00 DC 01（关）

ED DE 0A 04 04 01 01 00 DF 01（开）

4.6 Cmd：0x05

(查询故障0x05)

描述：查询产品故障数据，机器人将返回包括电机故障数据、电压故障数据、遥控器接收机故障数据、传感器故障数据等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x0A 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x05

Num：数据个数，0x01

Data：实际数据，0x00 0x00 （2字节）

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 0A 02 05 01 00 00 DD 01

4.7 Cmd：0x06

(修改芯片储存的硬件数据0x06)

描述：修改芯片储存的硬件数据，机器人将修改芯片储存的硬件数据包括固件号数据、电机功率数据、电机减速比数据、编码器线数数据、左右轮距、轮子直径、电池容量、电池电压数据、电机最大转速、左电机系数、右电机系数、发货日期等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x20 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x06

Num：数据个数，0x0C

Data：实际数据、

1、固件号数据 2字节（uint16_t）：例如2000则为V2000

2、电机功率数据 2字节（uint16_t）：例如60W则为0x3C

3、电机减速比数据　　2字节（uint16_t）：例如25减速比则为0x19

4、编码器线数数据　　2字节（uint16_t）：例如1000线则为0x3E8

5、左右轮距数据　　2字节（uint16_t）：例如376毫米则为0x178

6、轮子数据　　2字节（uint16_t）：例如150毫米则为0x96

7、电池容量数据　　2字节（uint16_t）：例如20AH则为0x14

8、电池数据　　2字节（uint16_t）：例如24V则为0x18

9、电机最大转速数据　2字节（uint16_t）：例如3000RBM(转/分钟)则为0xBB8

10、左电机系数　　2字节（uint16_t）：例如1.0则为1.0*10=10，则为0x0A

11、右电机系数　　2字节（uint16_t）：例如1.0则为1.0*10=10，则为0x0A

12、发货日期数据　　2字节（uint16_t）：例如23年7月(日期)则为0x2307

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 20 02 06 0C 00 20 3C 00 19 00 E8 03 78 01 96 00 14 00 18 00 B8 0B 0A 00 0A 00 07 23 9B 05

4.8 Cmd：0x07

（撞击解除指令0x07）

描述：防撞杆撞击解除，机器人将解除急停。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x0A 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x07

Num：数据个数，0x01

Data：实际数据，0x00 0x00 （2字节）

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 0A 02 07 01 00 00 DF 01

五.底盘数据上传

5.1 Cmd：0x80(底盘返回参数信息，一般只需要查询一次即可)

(查询芯片储存的硬件数据0x80)

描述：查询芯片储存的硬件数据，机器人将返回芯片储存的硬件数据包括固件号数据、电机功率数据、电机减速比数据、编码器线数数据、左右轮距、轮子直径、电池容量、电池电压数据、电机最大转速、左电机系数、右电机系数、发货日期等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x20 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x80

Num：数据个数，0x0C

Data：实际数据、

1、固件号数据	2字节（uint16_t）：	例如2000则为V2000
2、电机功率数据	2字节（uint16_t）：	例如60W则为0x3C
3、电机减速比数据	2字节（uint16_t）：	例如25减速比则为0x19
4、编码器线数数据	2字节（uint16_t）：	例如1000线则为0x3E8
5、左右轮距数据	2字节（uint16_t）：	例如376毫米则为0x178
6、轮子数据	2字节（uint16_t）：	例如150毫米则为0x96
7、电池容量数据	2字节（uint16_t）：	例如20AH则为0x14
8、电池数据	2字节（uint16_t）：	例如24V则为0x18
9、电机最大转速数据	2字节（uint16_t）：	例如3000RBM(转/分钟)则为0xBB8
10、左电机系数	2字节（uint16_t）：	例如1.0则为1.0*10=10，则为0x0A
11、右电机系数	2字节（uint16_t）：	例如1.0则为1.0*10=10，则为0x0A
12、发货日期数据	2字节（uint16_t）：	例如23年7月(日期)则为0x2307

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 00 20 80 0C 00 20 3C 00 19 00 E8 03 78 01 96 00 14 00 18 00 B8 0B 0A 00 0A 00 07 23 13 06

5.2 Cmd：0x81　　(底盘50HZ数据回传)

描述：机器人 20ms 返回信息；

Head：协议头，固定 2 个字节：　0xDE　0xED

Len：　为 0x30。

Type：为 0x04。

Cmd：命令类型，0x81

Num：为 0x15。

Data

里程数据模式。

0-1 线速度 Vx	4byte(int32)(mm/s)
2-3、角速度 Vz	4byte(flaot)(mm/s)
4-5、陀螺仪加速度 AccX	4byte(float)(g=9.8)
6-7、陀螺仪加速度 AccY	4byte(float)(g=9.8)
8-9、陀螺仪加速度 AccZ	4byte(float)(g=9.8)
10-11、陀螺仪加速度 GyrX	4byte(float)(rad)
12-13、陀螺仪加速度 GyrY	4byte(float)(rad)
14-15、陀螺仪加速度 GyrZ	4byte(float)(rad)
16、电压*10	2byte(uint16)(V)
17、底盘状态	2byte(uint16)
18、灯条 1-2 状态	2byte(uint16)低字节灯条 1 状态，高字节灯条 2 状态
19、灯条 3-4 状态	2byte(uint16)低字节灯条 3 状态，高字节灯条 4 状态

速度数据模式。

0、左轮速度	2byte(int16)(mm/s)
1、右轮速度	2byte(int16)(mm/s)
2、左轮编码器增量	2byte(int16)
3、右轮编码器增量	2byte(int16)
4-5、陀螺仪加速度 AccX	4byte(float)(g=9.8)
6-7、陀螺仪加速度 AccY	4byte(float)(g=9.8)
8-9、陀螺仪加速度 AccZ	4byte(float)(g=9.8)
10-11、陀螺仪加速度 GyrX	4byte(float)(rad)

12-13、陀螺仪加速度 GyrY 4byte(float)(rad)

14-15、陀螺仪加速度 GyrZ 4byte(float)(rad)

16、电压*10 2byte(uint16)(V)

17、底盘状态 2byte(uint16)(看下表)

18、灯条 1-2 状态 2byte(uint16)低字节灯条 1 状态，高字节灯条 2 状态

19、灯条 3-4 状态 2byte(uint16)低字节灯条 3 状态，高字节灯条 4 状态

Check：校验码，2 字节。从 Head 到 Data 的和

底盘状态	
低 8 位(底盘状态)(bit)	高 8 位(回充状态)(数值)
Bit0: (1：急停按下，0:急停未按下)	0：寻找中心信号 1：已找到中心信号 2：信号丢失 3：铜片接触 4：对接成功 5：对接错误（碰撞或充电过程中脱落） 6:寻充超时 7:退出充电状态（如果是回充成功的时候退出充电状态，车子会向前走一段距离，离开充电桩。）
Bit1: (1：电压故障，0:无故障)	
Bit2: (1：电机故障，0:无故障)	
Bit3: 前防撞杆状态	
Bit4: 后防撞杆状态	
Bit5: 软件急停状态	
Bit6: 抱闸 1 状态	
Bit7: 摆闸 2 状态	

5.3 Cmd：0x85

(返回故障0x85)

描述：查询产品故障数据，机器人将返回包括电机故障数据、电压故障数据、遥控器接收机故障数据等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x10 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x85

Num：数据个数，0x04

Data：实际数据：

1、电机故障数据 2字节（uint16_t）

0x01（二进制0000 0000 0000 0001）:左电机故障

0x02（二进制00000 0000 000 0010）:右电机故障

0x03（二进制00000 0000 000 0011）:左右电机故障

2、电压故障数据 4字节（uint16_t）

低电压故障：0x01（二进制0000 0000 0000 0001） 2字节（uint16_t）

高电压故障：0x02（二进制0000 0000 0000 0010） 2字节（uint16_t）

故障电压值：测量的故障电压值*10 2字节（uint16_t）

3、遥控器接收机故障数据 2字节（uint16_t）

0x01（二进制0000 0000 0000 0001）:无接收机信号

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 10 02 85 04 00 00 00 00 00 00 00 00 66 02

5.4 Cmd：0x86

(返回修改成功后芯片储存的硬件数据0x86)

描述：返回修改成功后芯片储存的硬件数据，机器人将返回修改成功后芯片储存的硬件数据包括固件号数据、电机功率数据、电机减速比数据、编码器线数数据、左右轮距、轮子直径、电池容量、电池电压数据、电机最大转速、左电机系数、右电机系数、发货日期等数据。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x20 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x86

Num：数据个数，0x0C

Data：实际数据、

1、固件号数据 2字节（uint16_t）：例如2000则为V2000

2、电机功率数据 2字节（uint16_t）：例如60W则为0x3C

3、电机减速比数据 2字节（uint16_t）：例如25减速比则为0x19

4、编码器线数数据 2字节（uint16_t）：例如1000线则为0x3E8

5、左右轮距数据 2字节（uint16_t）：例如376毫米则为0x178

6、轮子数据 2字节（uint16_t）：例如150毫米则为0x96

7、电池容量数据 2字节（uint16_t）：例如20AH则为0x14

8、电池数据 2字节（uint16_t）：例如24V则为0x18

9、电机最大转速数据 2字节（uint16_t）：例如3000RBM(转/分钟)则为0xBB8

10、左电机系数 2字节（uint16_t）：例如1.0则为1.0*10=10，则为0x0A

11、右电机系数　　2字节（uint16_t）：例如1.0则为1.0*10=10，则为0x0A

12、发货日期数据　　2字节（uint16_t）：例如23年7月(日期)则为0x2307

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 20 02 86 0C 00 20 3C 00 19 00 E8 03 78 01 96 00 14 00 18 00 B8 0B 0A 00 0A 00 07 23 1B 06

5.5 Cmd：0x87

(撞击解除状态返回指令0x07)

描述：撞击解除状态返回，机器人将返回当前急停状态。

Head：固定帧头 0xDE 0xED

Len： 帧长度，0x0A 整个发送数据的字节数

Type：产品类型，0x02 AGV系列类型为0x02

Cmd：命令类型，0x87

Num：数据个数，0x01

Data：实际数据：

0x00 （机器人当前为运动状态）

0x01 （机器人当前为急停状态）

Check：校验码，2字节。从Head到Data的和

指令示例：ED DE 0A 02 87 01 00 00 5F 02

六. 小端模式说明以及表示方法

小端模式：Little-Endian就是低位字节排放在内存的低地址端，高位字节排放在内存的高地址端。

（本文　数据存储传输都是以小端模式）

比如32dit宽数字0x12 34 56 78在内存中的表示

形式。低地址---- 高地址

0x78 I 0X56 IOX34 I 0X12

比如16bit宽的数0x12 34在内存中的表示

形式。低地址---- 高地址

0x34 I 0x12

七：ROS包使用说明

7.1:ROS包的使用（ROS包名称：ros_agv3_msg）

我们需要实现对ROS包的编译，编译这个ros包之前，需要下载相关的配置包，如下图：

```
zlkjlin@zlkjlin-TM1801:~/catkin_ws1$
zlkjlin@zlkjlin-TM1801:~/catkin_ws1$ sudo apt-get install ros-kinetic-serial
ros-kinetic-serial        ros-kinetic-serial-utils
zlkjlin@zlkjlin-TM1801:~/catkin_ws1$ sudo apt-get install ros-kinetic-serial
[sudo] zlkjlin 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列【新】软件包将被安装:
  ros-kinetic-serial
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 37 个软件包未被升级。
需要下载 39.2 kB 的归档。
解压缩后会消耗 148 kB 的额外空间。
获取:1 http://packages.ros.org/ros/ubuntu xenial/main amd64 ros-kinetic-serial amd64 1.2.1-0xenial-2019
已下载 39.2 kB，耗时 5秒 (7,661 B/s)
正在选中未选择的软件包 ros-kinetic-serial。
(正在读取数据库 ... 系统当前共安装有 293103 个文件和目录。)
正准备解包 .../ros-kinetic-serial_1.2.1-0xenial-20191214-001149+0000_amd64.deb  ...
正在解包 ros-kinetic-serial (1.2.1-0xenial-20191214-001149+0000) ...
正在设置 ros-kinetic-serial (1.2.1-0xenial-20191214-001149+0000) ...
```

图1（根据上图红色框安装相关配置包）

安装完相关的包，我们开始进行ros的编译：

```
catkin_create_pkg          catkin_find_pkg            catkin_init_workspace      catkin_make_isolated       catkin_p
catkin_find                catkin_generate_changelog  catkin_make                catkin_package_version     catkin_t
zlkjlin@zlkjlin-TM1801:~/catkin_ws1$ catkin_make
Base path: /home/zlkjlin/catkin_ws1
Source space: /home/zlkjlin/catkin_ws1/src
Build space: /home/zlkjlin/catkin_ws1/build
Devel space: /home/zlkjlin/catkin_ws1/devel
Install space: /home/zlkjlin/catkin_ws1/install
####
#### Running command: "make cmake_check_build_system" in "/home/zlkjlin/catkin_ws1/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/zlkjlin/catkin_ws1/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.12", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/zlkjlin/catkin_ws1/build/test_results
-- Found gtest sources under '/usr/src/gmock': gtests will be built
-- Found gmock sources under '/usr/src/gmock': gmock will be built
```

图 2（根据上图红色框框找到对应的ros工作空间编译）

编译完后，可以启动相关的ros包，我们启动ros包，可以看到如下指令（记得在启动ros包之前记得source相关的工作空间）：

```
zlkjlin@zlkjlin-TM1801:~/catkin_ws1/src$ rosrun ros_agv3_msg ros_agv3_talker
[ INFO] [1623900991.326713916]: Serial Port initialized
Vx:0 Vz:0.00 Ax:0.00 Ay:0.00 Az:0.00 Gx:0.00 Gy:0.00 Gz:0.00 Vo:222 St:192 L12:30583 L34:0
[ INFO] [1623900991.352627210]: ----------------------------------------------------

Vx:0 Vz:0.00 Ax:0.00 Ay:0.00 Az:0.00 Gx:0.00 Gy:0.00 Gz:0.00 Vo:222 St:192 L12:30583 L34:0
[ INFO] [1623900991.372619283]: ----------------------------------------------------

Vx:0 Vz:0.00 Ax:0.00 Ay:0.00 Az:0.00 Gx:0.00 Gy:0.00 Gz:0.00 Vo:222 St:192 L12:30583 L34:0
[ INFO] [1623900991.392553042]: ----------------------------------------------------
```

图 3（启动ros包，可以看到如下数据，说明启动成功）

启动ROS包之后，我们就可以通过ROS主题来订阅发布相关的消息了，如下：

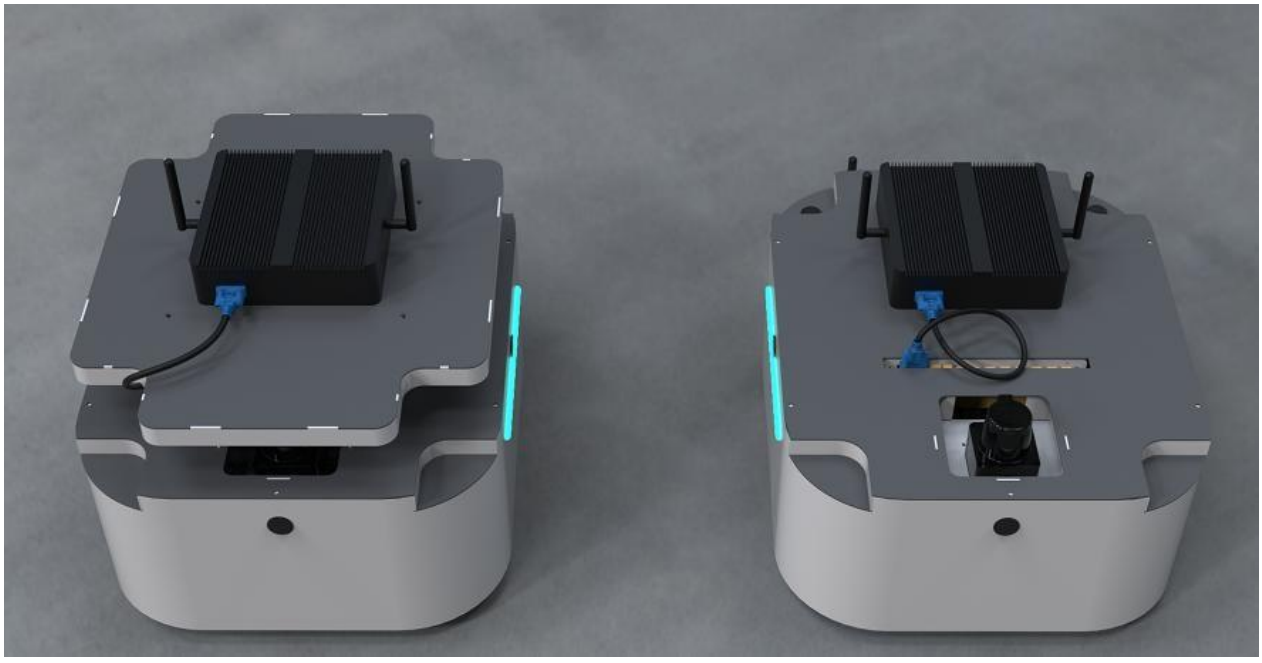
```
zlkjlin@zlkjlin-TM1801:~$ rostopic echo /agv_info
Vx: 698
Vz: 0.0111111113802
AccX: 0.0
AccY: 0.0
AccZ: 0.0
GyrX: 0.0
GyrY: 0.0
GyrZ: 0.0
Voltage: 220
State: 192
Light12: 257
Light34: 0
```

图 4（这里通过主题输出相关的数据）

相关节点内容如下：

Vx：	X轴线速度（mm/S）	GyrX：	X轴角速度（0表示未接入设备）
Vz：	Z轴角速度（rad/S）	GyrY：	Y轴角速度（0表示未接入设备）
ACCX：	X轴加速度（0表示未接入设备）	GyrZ：	Z轴角速度（0表示未接入设备）
ACCY：	Y轴加速度（0表示未接入设备）	Voltage：	电压值（220表示：22.0V）
ACCZ：	Z轴加速度（0表示未接入设备）	state：	底盘状态
Light12:	灯条管1	Light34:	灯条管2

通讯接线示意图，如下：



（如图红色框框为底盘232串口与主机连接）

7.2：注意事项

7.2.1:如果出现权限被拒绝问题，串口打开失败怎么解决？

请到对应的dev/文件下，对相关的串口进行权限设置，例如对ttyUSB0进行权限设置，可以用：sudo chmod 777 -R ttyUSB0。

7.2.1:如果出现串口号找不到的情况，怎么解决？

可以重复插拔串口，看看串口号是多少，不一定是ttyUSB0，有时候可能是ttyUSB1，在对应的串口程序里改过来，重新编译就可以。