



编码器智能小车底盘

源码说明手册

V1.0



关注塔克创新微信公众号，获取更新资讯

烟台塔克电子科技有限公司

版权申明

本手册版权归塔克创新所有，并保留一切权力，受法律保护。未经(书面形式)同意，任何单位及个人不得擅自摘录或修改本手册部分或全部内容，违者我们将追究其法律责任。

版本说明

版本	日期	内容说明
V1.0	2024/6/18	第一次发布

塔克媒体

塔克官网	www.xtark.cn
淘宝店铺	https://xtark.taobao.com
塔克哔哩	https://space.bilibili.com/511052131
销售邮箱	sales@xtark.cn

本手册为塔克通用的编码器智能小车底盘源码使用说明，包含麦轮、四轮差速、两轮差速、阿克曼、三轮全向等底盘控制，支持 PS2 手柄遥控和上位机软件串口控制。

1. 硬件说明

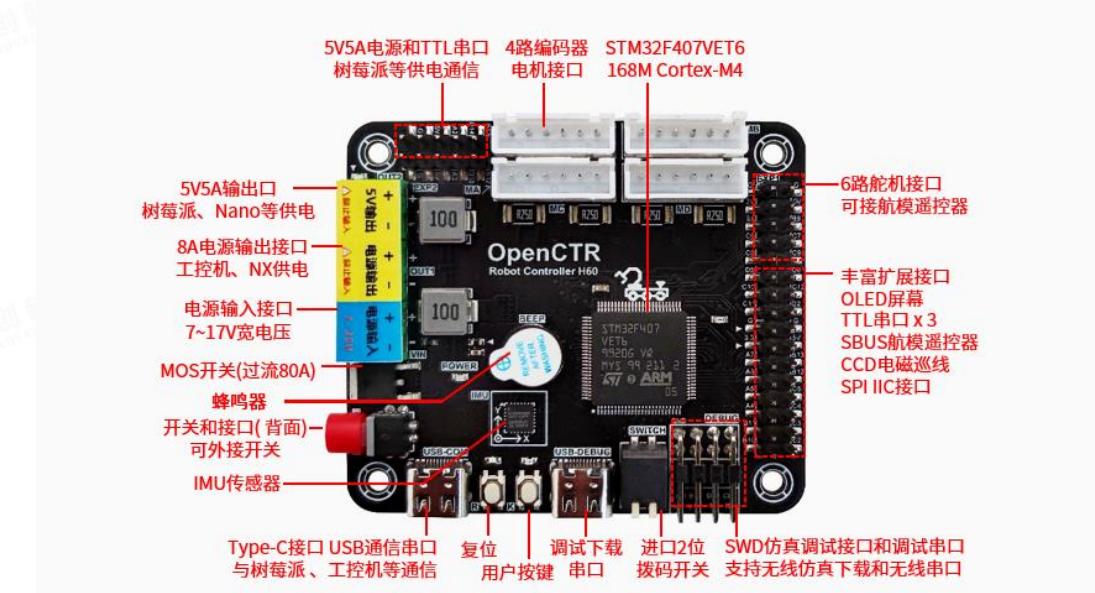
源码基于 OpenCTR H60 控制器开发，采用 R20 系列底盘测试，使用 MC520 13 线霍尔编码器电机。其它自行准备的硬件平台，请参考移植。



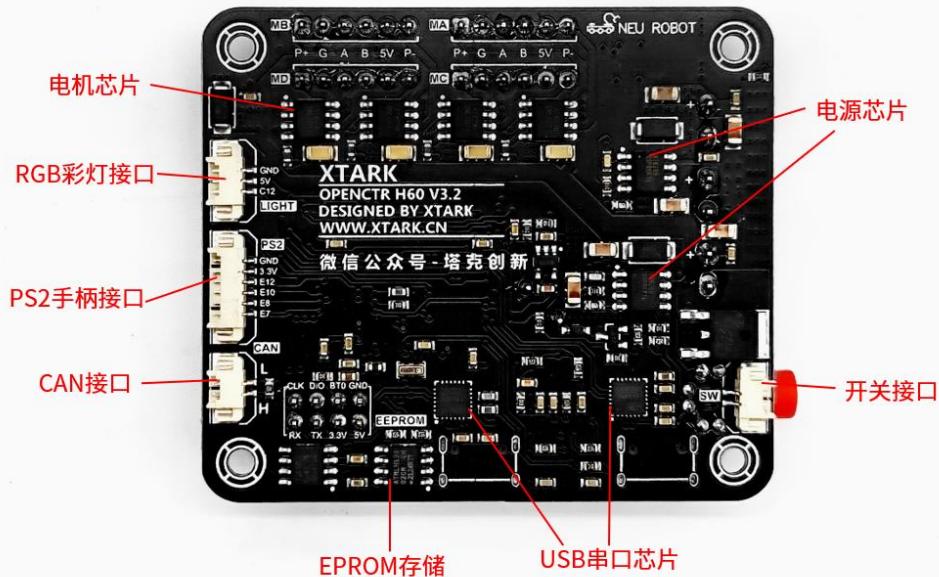
1.1. 控制器说明

OpenCTR H60 机器人运动控制器，OpenCTR (Open-source Controller Module for Robot) 为专用于机器人开发的开源控制器，体积小巧，外观精致。一个板子可以控制麦轮、阿克曼、差速、全向轮、履带等多种底盘。

OpenCTR H60 控制器采用 STM32F407 作为主控制，板载 IMU 加速度陀螺仪传感器，支持四路编码器直流电机，板载两路 5V 电源，最大电流 5A，具有丰富的板载资源和扩展接口，非常适合 ROS 机器人开发，可以和树莓派、Jetson 系列、工控机等 ROS 主控组成 ROS 机器人。控制板正面资源配置如下图所示。

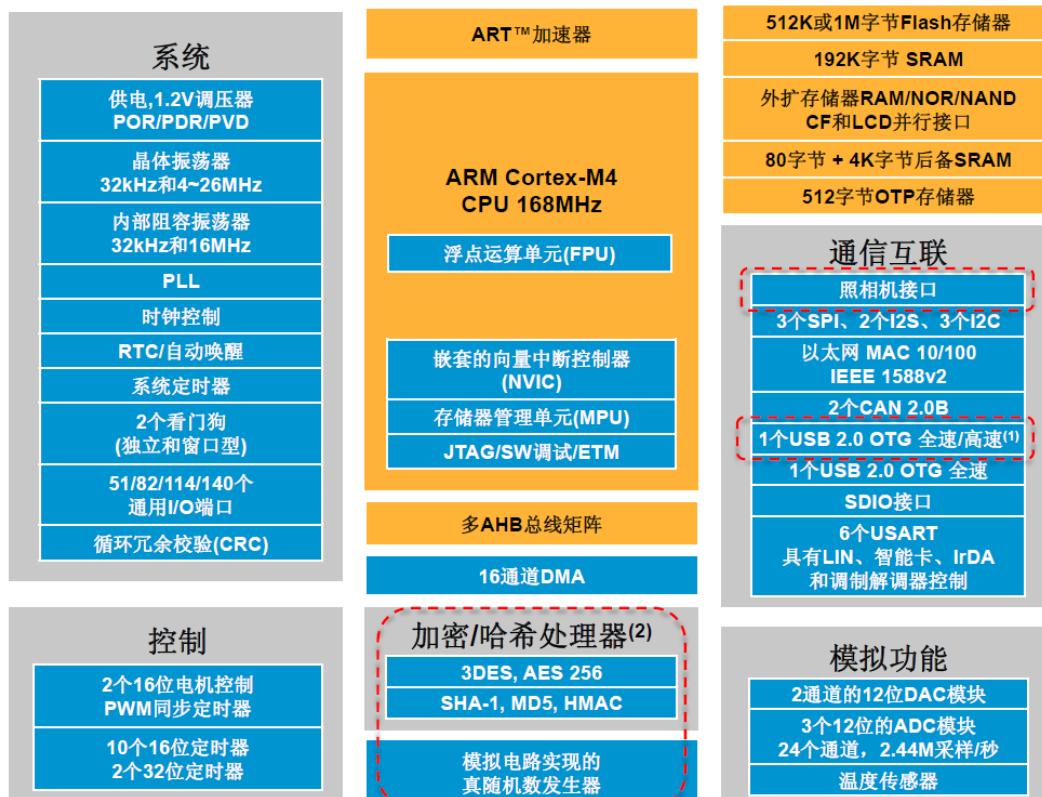


背面资源和接口如下图所示。



OpenCTR 控制器提供原理图，具有 SWD 调试接口，支持 USB 串口程序下载，可实现 STM32 编程二次开发。提供板载资源及外设示例代码，方便用户学习使用。

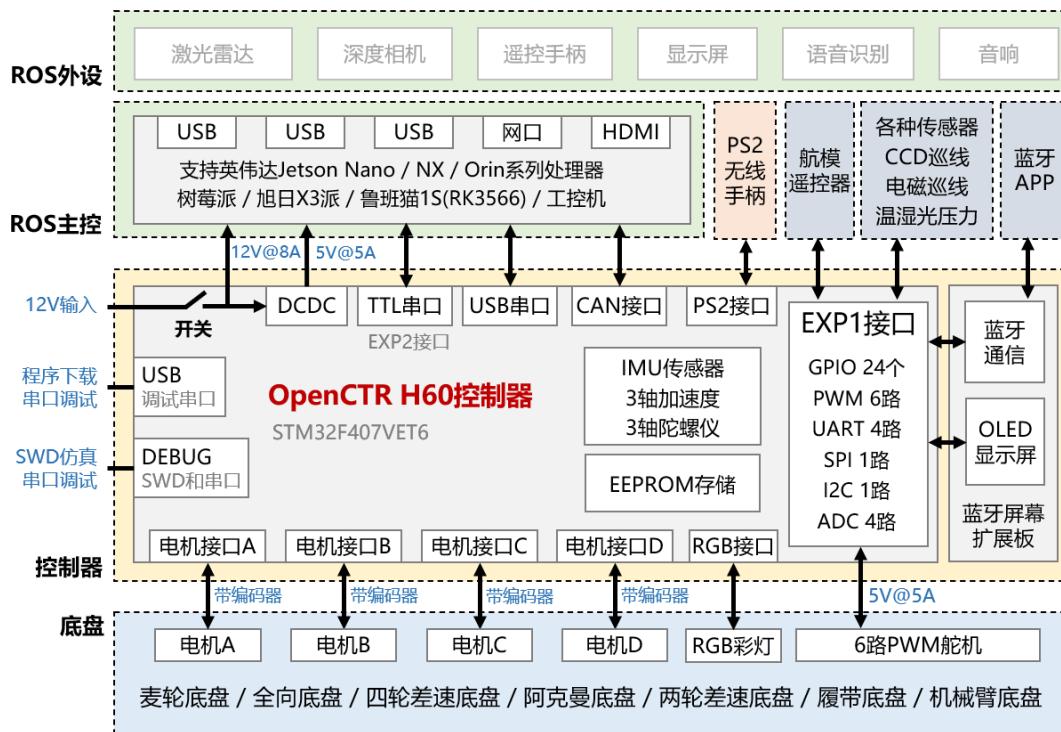
OpenCTR 控制器主控采用 ST 公司 STM32F407VET6，其采用 ARM 的 Cortex-M4 内核，168M 主频，片内 FLASH 容量为 512K，片内 SRAM 容量为 192K，集成 FPU 和 DSP 指令。STM32F40x/41x 系列系统框图如下图所示。



STM32F407VET6 处理器片内资源及外设如下图所示。

Peripherals	STM32F405RG	STM32F405VG	STM32F405ZG	STM32F407Vx	STM32F407Zx	STM32F407Ix
Flash memory in Kbytes		1024		512	1024	512
SRAM in Kbytes	System			192(112+16+64)		
	Backup			4		
FSMC memory controller	No			Yes		
Ethernet		No			Yes	
Timers	General-purpose			10		
	Advanced-control			2		
	Basic			2		
Random number generator				Yes		
Communication interfaces	SPI / I ² S			3/2 (full duplex)		
	I ² C			3		
	USART/UART			4/2		
	USB OTG FS	No			Yes	
	USB OTG HS	Yes			Yes	
CAN				2		
Camera interface		No			Yes	
GPIOs	51	82	114	82	114	140
12-bit ADC				3		
Number of channels	16	16	24	16	24	24
12-bit DAC				Yes		
Number of channels				2		
Maximum CPU frequency				168 MHz		
Operating voltage				1.8 to 3.6 V ⁽¹⁾		
Operating temperatures				Ambient temperatures: -40 to +85 °C / -40 to +105 °C		
				Junction temperature: -40 to +125 °C		
Package	LQFP64	LQFP100	LQFP144	LQFP100	LQFP144	UFBGA176 LQFP176

OpenCTR H60 控制器组成机器人硬件连接系统框图如下所示。



1.2. 硬件连接说明

小车例程相关简单，只需要连接编码器电机、PS2 手柄即可。阿克曼车型需要连接舵机。

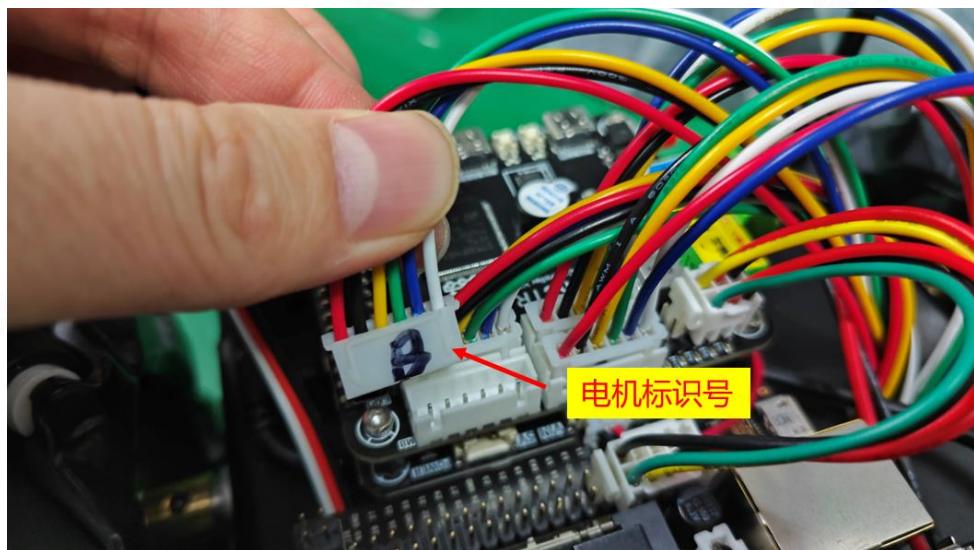
1.2.1. 直流编码器电机

以 MC520 直流减速编码器电机为例说明，本店其它类型电机接口定义相同，可直接替换使用。

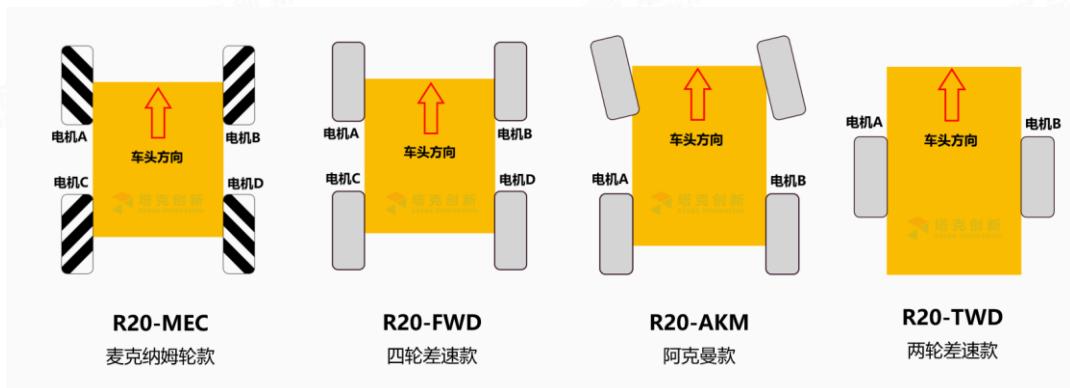
两种编码器接口定义相同，线材和控制板接口均可共用。



电机连接线参考下图所示，具体使用的硬件端口，可参考程序源码。



机器人底盘电机位置编号如下图所示，只有底盘电机插入正确的控制器位置，STM32 程序才可正确运行，否则会导致电机控制异常。

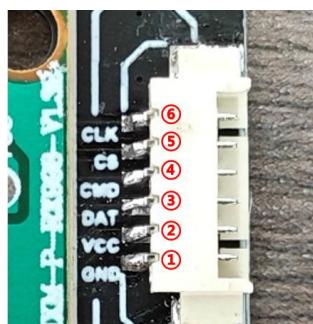


1.2.2. PS2 无线手柄

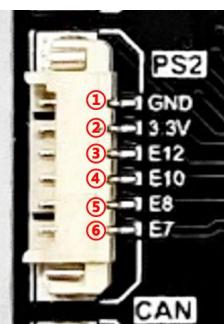
程序支持 PS 手柄控制，OpenCTR H60 可配合本店 PS2 手柄使用，采用塔克定制小巧的接收器，连接方便。



PS2 手柄接口定义及控制板接口定义如下图，通过专用排线连接，如下图所示。



PS2手柄接收器接口定义



OpenCTR H60
手柄接口定义



PS2 手柄引脚说明如下表所示。

引脚	定义	引脚说明
1	GND	手柄电源地信号。
2	VCC	接收器工作电源，供电电压范围3~5V。
3	DAT	信号从手柄到手机，8bit串行数据，在时钟的下降沿进行传送。读取信号需要在时钟由高到低的变化过程中进行。
4	CMD	信号从主机到手柄，8bit串行数据，也是在时钟的下降沿进行传送。此信号与DI/DAT信号相对应。
5	CS	手柄工作使能信号，在通信期间，设置为低电平。
6	CLK	时钟信号，由主机提供，用于保持数据同步。

1.2.3. PWM 舵机

阿克曼车型转向舵机，舵机连接位置如下图所示，PB8 和 PB9 接口均可使用，选择一个连接即可，默认连接 PB8 舵机接口。



2. 基本使用介绍

程序源码支持 PS2 无线手柄控制，支持上位机串口控制，并且将解析的里程计数据通过串口发送到上位机。

注意在使用上位机软件控制时，请务必拔下 PS2 手柄接收器，否则会影响使用。

2.1. PS2 手柄控制

机器人可通过 PS2 手柄控制，首先确保手柄接收器已经连接到 OpenCTR 控制器。接收器的红灯常亮，绿灯闪烁为等待连接状态。

此时打开 PS2 手柄开关，PS2 手柄会自动连接接收器，连接成功后，接收器的绿灯为常亮状态。



注意，如果手柄只有红灯亮，请按“MODE”按键进入“红绿灯模式”再进行相关操作，否则操作无效。



PS2 手柄按键定义，如下图所示，这里只用了部分按键。注意，手柄在连接状态下，5分钟无按键按下，自动启动休眠模式。按下 START 即可唤醒手柄。



右侧摇杆控制机器人 X 和 Y 速度，即摇杆上推前进，下推后退，左右控制机器人横向移动，左右控制仅适用于麦轮等全向移动机器人，非全向移动机器人无效。左侧摇杆左右移动控制机器人转向，上下移动无效。

同时左右摇杆也是按键，可以调节机器人速度，按“左侧摇杆按键”机器人速度减小，速度减小到最小时不能再减小，并有蜂鸣器鸣叫提示。按“右侧摇杆按键”速度增大，增大到最大时不能再增加，并有蜂鸣器鸣叫提示。



2.2. 电脑软件控制

小车程序支持塔克上位机串口软件助手控制。

2.2.1. 塔克串口通信协议

首先介绍下通信协议。串口传输是单片机开发，机器人制作常用输出传输手段，串口传输一般包括两种传输方式，一种基于编码器数据传输，例如 `printf` 函数的传输，这种方式简单直观适合文本传输，传输数据效率较低。另一种是基于 16 进制原始数据传输，适合数据传输。

为了方便软件工具和各个产品进行通信，塔克基于 16 进制数据传输规范了一套通信协议 X-Protocol 协议，塔克产品生态均遵循塔克通用标准的 X-Protocol 传输协议，包含 ROS 串口通信协议，蓝牙 APP 控制协议。配套的串口调试软件、串口示波器软件也均使用此协议。协议内容如下。



帧头：双帧头，抗干扰强。

帧长：根据数据长度设定，包含帧头、校验和等全部数据长度。

帧码：用户根据功能设定，标识帧的唯一性。

数据：高位在前，长度可变，内容自由组合 8 位，16 位，32 位数据。

校验和：前面数据累加和的低 8 位。

如下是示例帧，传输的数据是 1000, -1000, 10。

AA 55 | 0B | 01 | 03 E8 FC 18 00 0A | 14
 帧头 帧长度 帧号 1000 -1000 10 校验和

控制协议

串口控制协议基于塔克 X-Protocol 传输协议，上位机软件向 OpenCTR 控制器发送的控制指令，速度为目标速度，单位为 m/s，为了方便传输，这里放大了 1000 倍，具体帧定义如下表所示。

机器人速度指令，包含机器人X, Y, W三个速度											
数据位	1	2	3	4	5	6	7	8	9	10	11
定义	帧头1	帧头2	帧长	帧号	速度X 高8位	速度X 低8位	速度Y 高8位	速度Y 低8位	速度W 高8位	速度W 低8位	校验位 累加和
数据	0xAA	0x55	11	0x50	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	0xXX

数据反馈协议

程序通过串口向上位机发送里程计数据，速度数据为里程计解算的速度，单位为 m/s，为了方便传输，这里放大了 1000 倍。具体协议见下表。

综合数据，包含IMU，机器人实时速度，电池电压数据等。										
数据位	1	2	3	4	5	6	7	8	9	10
定义	帧头1	帧头2	帧长	帧号	加速度X 高8位	加速度X 低8位	加速度Y 高8位	加速度Y 低8位	加速度Z 高8位	加速度Z 低8位
数据	0xAA	0x55	25	0x10	±32768	±32768	±32768	±32768	±32768	±32768
数据位	11	12	13	14	15	16	17	18	19	20
定义	陀螺仪X 高8位	陀螺仪X 低8位	陀螺仪Y 高8位	陀螺仪Y 低8位	陀螺仪Z 高8位	陀螺仪Z 低8位	速度X 高8位	速度X 低8位	速度Y 高8位	速度Y 低8位
数据	±32768	±32768	±32768	±32768	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍	m/s放大1000倍
数据位	21	22	23	24	25					
定义	速度W 高8位	速度W 低8位	电池电压 高8位	电池电压 低8位	校验位 累加和					
数据	m/s放大1000倍	实际电压放大100倍	0xXX							

2.2.2. 上位机串口软件控制

前面已经了解到串口控制详细内容，可以通过塔克提供的串口助手软件看机器人发送的里程计数据。使用前请安装好串口驱动，选择对应 COM 端口，波特率为 230400，选中“16 进制”和“协议”选项，显示界面如下图所示。



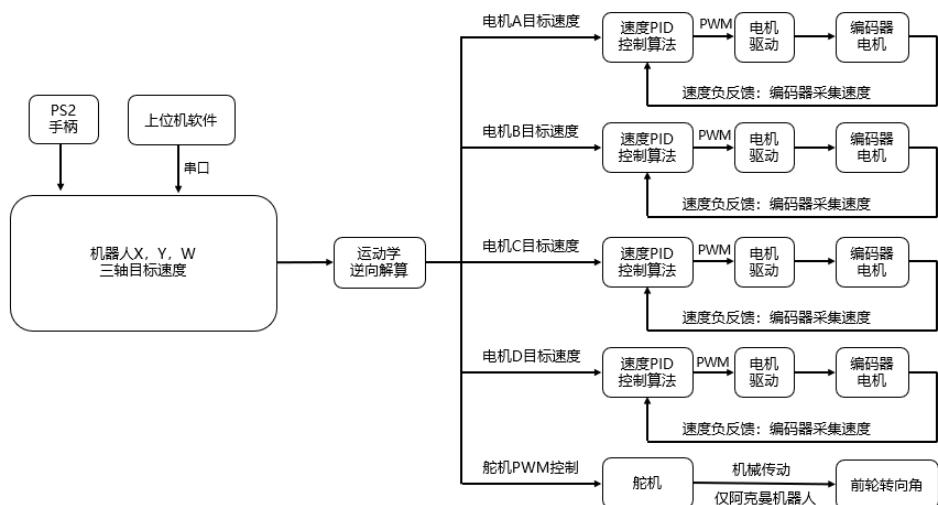
速度控制栏，拖动滑块可以实现对机器人底盘的速度控制，帧编码是 80(0x50)，XV 代表 X 轴速度（前进后退），XY 代表 Y 轴速度（横向速度，仅支持全向轮），VW 代表转向速度。归零按键可快速设置机器人速度为 0，机器人停止运动。

3. STM32 源码说明

本章节对小车源码进行简要说明。

3.1. 控制流程图

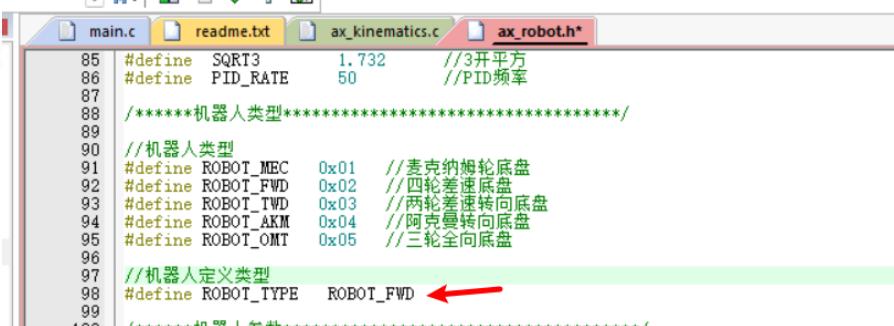
机器人运动控制原理都是通过改变机器人目标速度来实现控制。机器人目标速度经过运动学逆向解算，计算获得电机目标速度，作为电机速度 PID 控制器的输入，经过 PID 运算后，通过 STM32 定时器输出 PWM 电机控制信号给电机驱动，电机驱动控制电机转动，编码器采集电机实时速度，反馈给 PID 控制器，实现速度闭环控制。机器人 STM32 电机控制流程图如下所示。



注意不同机器人类型有差异，麦克纳姆轮，四轮差速机器人需要四个电机，三轮全向机器人只需要三个电机，两轮差速和阿克曼机器人需要两个电机，另外阿克曼需要舵机转向。

3.2. 编译说明

源码通过条件编译的方式实现支持多种类型小车底盘。可以通过如下宏定义小车类型。



```
main.c readme.txt ax_kinematics.c ax_robot.h*
85 #define SQRT3 1.732 //3开平方
86 #define PID_RATE 50 //PID频率
87
88 /*****机器人类型*****
89 //机器人类型
90 #define ROBOT_MEC 0x01 //麦克纳姆轮底盘
91 #define ROBOT_FWD 0x02 //四轮差速底盘
92 #define ROBOT_TWD 0x03 //两轮差速转向底盘
93 #define ROBOT_AKM 0x04 //阿克曼转向底盘
94 #define ROBOT_OMT 0x05 //三轮全向底盘
95
96 //机器人定义类型
97 #define ROBOT_TYPE ROBOT_FWD ←
98
99 **** I 1 *****
```

不同类型机器人大部分代码是共用的，主要区别是运动学模型解析函数和机器人底盘参数。另外阿克曼由于是舵机转向，不能实现原地旋转，控制方式与其它几款有差异，具体参考代码。

3.3. 运动学模型分析

软件支持多种机器人底盘，不同型号机器人具有不同特点。麦克纳姆轮和三轮全向机器人可以实现 360 度全向移动，除了前进和转向运动，还可以实现平面 360 度任意方向的移动。四轮和两轮差速机器人，通过左右两边轮子速度差实现转向。阿克曼机器人则是像汽车一样，通过舵机控制前轮实现转向。

关于各种机器人底盘的运动学模型详细说明，可以查看《运动学模型分析教程》相关文档和视频教程。

4. 其它问题

本章讲述底盘程序的使用过程中涉及的一些问题。

4.1. 其它电机类型和底盘测试

底盘源码以 MC520 13 线霍尔编码器电机为例开发，其它 MC130 编码器 TT 马达、MC310 编码器电机小车请参考移植，控制原理相同，主要是电机减速比，轮距，轴距等参数不同，可根据自己的电机或小车实际情况修改。

```
*****机器人参数*****
//麦轮机器人参数
#define MEC_WHEEL_BASE 0.182 //轮距, 左右轮的距离
#define MEC_ACLE_BASE 0.124 //轴距, 前后轮的距离
#define MEC_WHEEL_DIAMETER 0.080 //轮子直径
#define MEC_WHEEL_RESOLUTION 1560.0 //编码器分辨率(13线), 减速比30, 13x30x4=1560
#define MEC_WHEEL_SCALE (PI*MEC_WHEEL_DIAMETER*PID_RATE/MEC_WHEEL_RESOLUTION) //轮子速度m/s与编码器转换系数

//四轮差速机器人参数
#define FWD_WHEEL_BASE 0.162 //轮距, 左右轮的距离
#define FWD_WB_SCALE 1.75 //轮距系数, 轮距系数与机器人的总负载、轮胎与地面的相对摩擦系数、转弯半径及质心高度有关
#define FWD_WHEEL_DIAMETER 0.065 //轮子直径
#define FWD_WHEEL_RESOLUTION 1560.0 //编码器分辨率(13线), 减速比30, 13x30x4=1560
#define FWD_WHEEL_SCALE (PI*FWD_WHEEL_DIAMETER*PID_RATE/FWD_WHEEL_RESOLUTION) //轮子速度m/s与编码器转换系数

//两轮差速机器人参数
#define TWD_WHEEL_DIAMETER 0.0724 //轮子直径
#define TWD_WHEEL_BASE 0.206 //轮距, 左右轮的距离
#define TWD_WHEEL_RESOLUTION 1560.0 //编码器分辨率(13线), 减速比30, 13x30x4=1560
#define TWD_WHEEL_SCALE (PI*TWD_WHEEL_DIAMETER*PID_RATE/TWD_WHEEL_RESOLUTION) //轮子速度m/s与编码器转换系数

//阿克曼机器人参数
#define AKM_WHEEL_BASE 0.165 //轮距, 左右轮的距离
#define AKM_ACLE_BASE 0.175f //轴距, 前后轮的距离
#define AKM_WHEEL_DIAMETER 0.075 //轮子直径
#define AKM_WHEEL_RESOLUTION 1560.0 //编码器分辨率(13线), 减速比30, 13x30x4=1560
#define AKM_TURN_R_MINI 0.35f //最小转弯半径( L*cot30-W/2 )
#define AKM_WHEEL_SCALE (PI*AKM_WHEEL_DIAMETER*PID_RATE/AKM_WHEEL_RESOLUTION) //轮子速度m/s与编码器转换系数

//三轮全向机器人参数
#define OMT_WHEEL_DIAMETER 0.058 //轮子直径
#define OMT_WHEEL_RADIUS 0.206 //机器人半径, 轮子与机器人中心距离
#define OMT_WHEEL_RESOLUTION 1560.0 //编码器分辨率(13线), 减速比30, 13x30x4=1560
#define OMT_WHEEL_SCALE (PI*OMT_WHEEL_DIAMETER*PID_RATE/OMT_WHEEL_RESOLUTION) //轮子速度m/s与编码器转换系数
```