

C620 电调 Demo 程序说明

Demo 功能说明

该 Demo 程序以 RoboMaster 主控开发板为控制板 以两种控制模式(PWM 或 CAN) 来驱动 C620 电调控制 M3508 电机。通过按键交互，来改变电机的转速。

环境平台介绍

本程序以 RoboMaster 主控开发板(以下简称 RM 主控板)为例进行演示.如图 1



图 1. RoboMaster 主控开发板([链接](#))

使用 Keil5 μ Vision IDE 作为开发环境.主控 MCU 是 ST 的 STM32F427.该 STM32 系列单片机除了普通固件库以外,还具有 STM32CubeMX 工具和 STM32 HAL 固件库.本文就使用了 STM32CubeMX 工具与其对应的 stm32F4 系列 firmware package 生成外设驱动的初始化代码.软件预览图如图 2 所示:

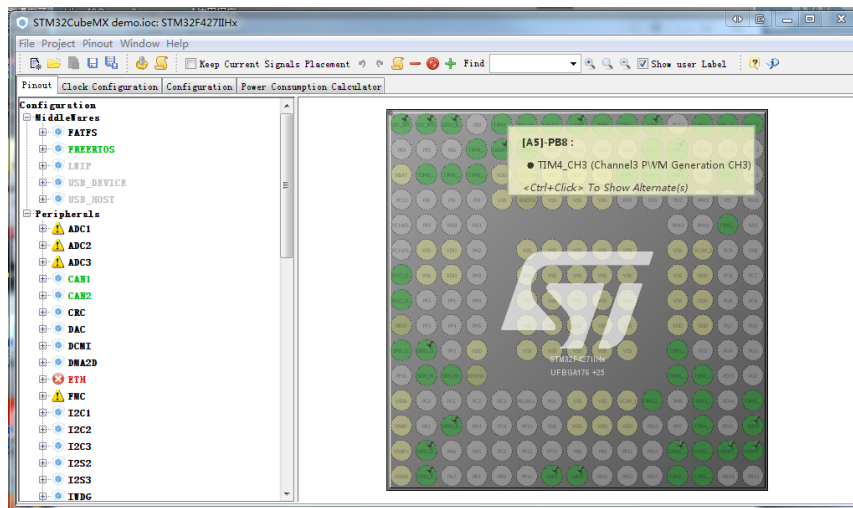


图 2.stm32cubeMX 软件([点击获取](#))

1. 在网页最下角下载 cubemx.需要登录 ST 账号,如果没有则要先注册.如果觉得麻烦,可以找网上的资源,下载完安装就行。
2. 在网页中间偏下可以找到 EMBEDDED SOFTWARE, 这个里面例如 STM32CubeF0 就是 HAL 固件库包。
3. cubeMX 软件需要 java 平台.确保已经安装好 JRE (Java Runtime Environment)。
4. <http://www.stm32cube.com/>一个中文 cube 入门学习网站(非官方),也有软件和固件库包等资源下载。

固件库安装说明

下载好固件库的 zip 包, 打开 cubemx 软件,依次是 help->install new libraries->from local ... 选择 zip 包 open 即可.安装成功后可以在左侧看到对应器件系列固件库版本号的 checkbox 变成绿色,如图下图所示 :

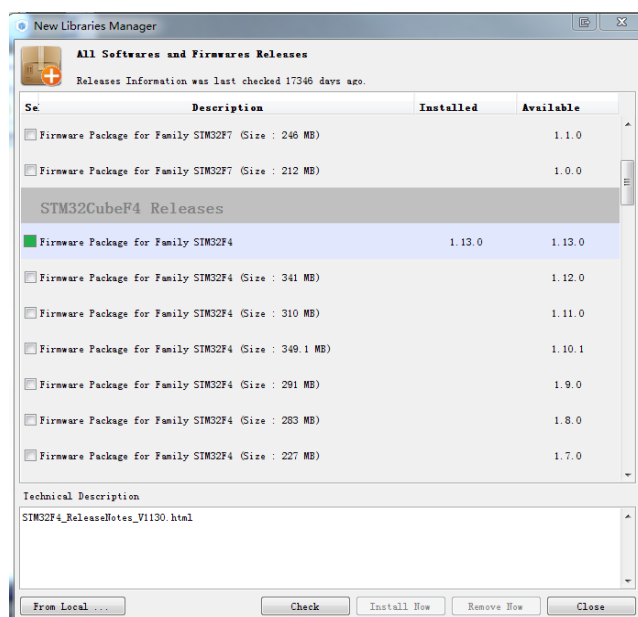


图 3. F4 系列 Firmware package

注意:必须装 pack,不能只安装 cubemx.因为各个系列的 pack 里才有该系列的外设驱动代码,否则无法生成初始化代码。

硬件连线说明

1. 将 C620 电调的三相输出线(3 色)和 7-Pin 数据线(黑色)连接 M3508 电机。
2. 根据需要选择电调 2 种控制方式之一, CAN 或者 PWM.接线图详见电调使用说明书。

a) 使用 CAN 指令控制的连接方式:

将 CAN 信号线 (2-Pin) 分别连接电调和 RM 主控板的 CAN1 信号端口。C620 电调的 PWM 信号端口不接。

b) 使用 PWM 信号的连接方式:

将 PWM 信号线 (3-Pin) 分别连接电调和 RM 主控板,如图 4 标注的 PWM 端口 (C1,C2,C3 或 C4)。C620 电调的 CAN 信号端口不接。

3. 将 C620 电调的电源线(黑色)连接到 RM 主控板的 24V 电源端口。

4. 仔细检查接线后,为 RM 主控板供电(24V)。

注意:不允许同时连接 CAN 信号线和 PWM 信号线,否则会导致电调控制失常!

本程序分别 PWM 和 CAN 协议作为示例。

使用 TIM5 作为 PWM 信号端口时, 引脚在 RM 主控板圈红色处如图下图所示:

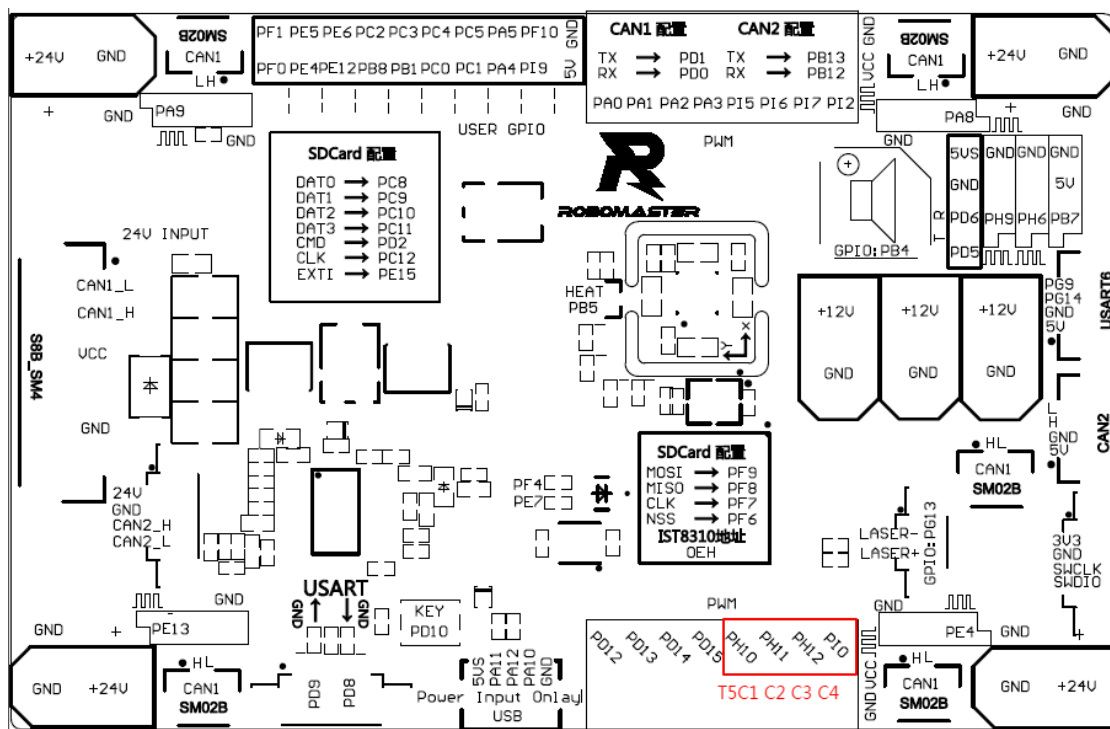


图 4. RM 主控开发板引脚示意图

程序框架

根据 cube 配置生成 Keil5 工程,已经将各部分源码文件分组了.本示例 Demo 添加一个

Bsp 组,在里面添加用户自己的代码文件.生成的代码结构大致如下:

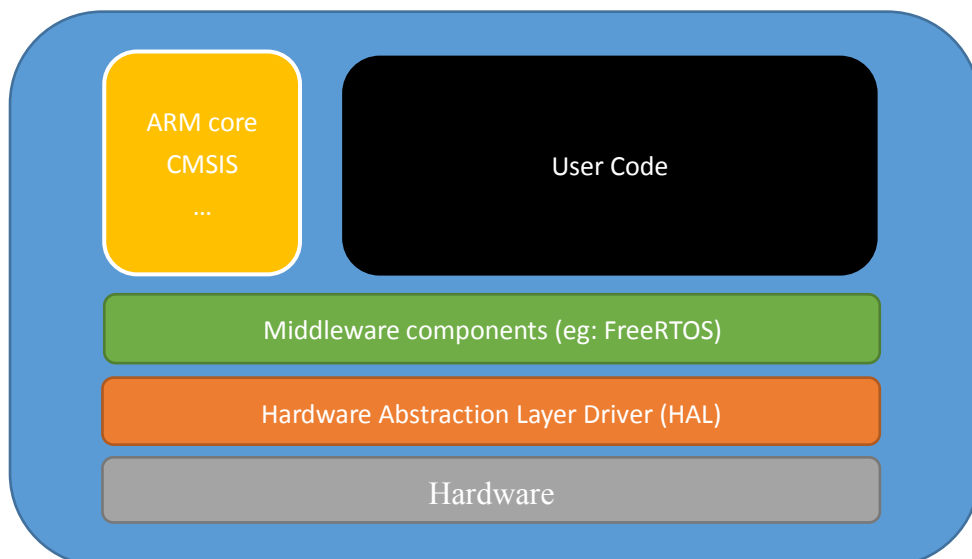


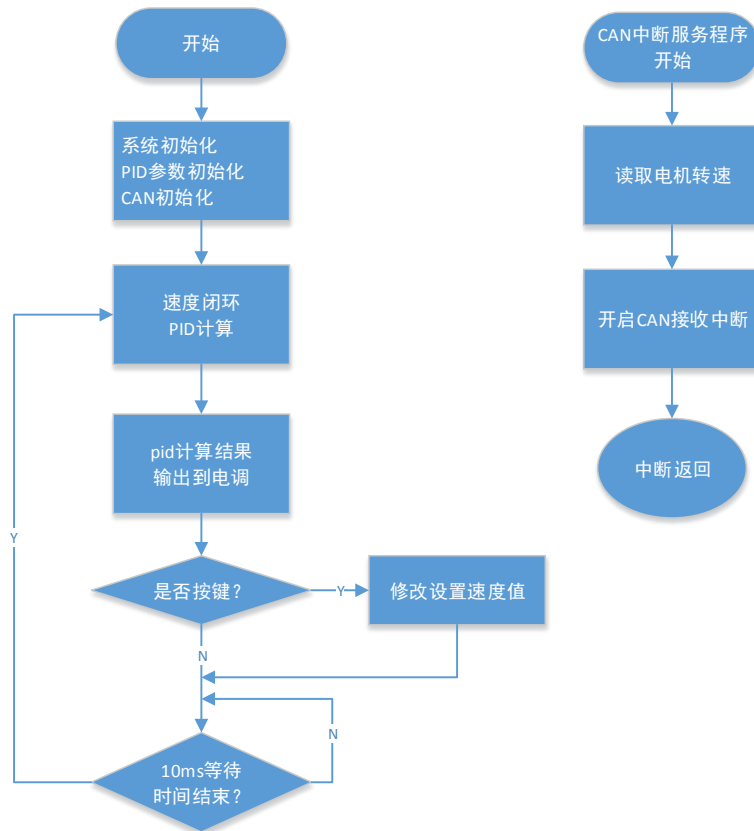
图 3.整个工程代码层次

本示例 Demo 勾选 FreeRTOS 之后,cubemx 生成的工程已经移植好了 FreeRTOS 实时系统.在 BSP 组里添加 bsp_can.c/h , pid.c/h 等文件.其中 bsp_can 主要写了 CAN 滤波器初始化,按格式接收与发送等函数, 还有 RX 接收中断部分. Pid 部分就是包含了 pid 的参数初始化还有计算函数等. 本文的主要控制代码放在 freertos.c 里面的 StartDefaultTask 函数里.

其中可以使用宏定义#define CAN_CONTROL 和#define PWM_CONTROL 来切换控制方式.

注意: 使用 CAN 控制时,电调对电机是电流闭环控制,所以需要在程序中从电调获取电机转速并增加电机转速 PID 调节控制. 使用 PWM 控制时,电调对电机是转速闭环控制,只需要给定符合标准的 PWM 信号就行,其中 PWM 信号的脉宽决定电机转速.

对于电机的 CAN 控制速度闭环控制, 示例程序流程如下:



程序运行

1. 使用 CAN 控制电机转动

- 按照硬件连接说明中的使用 CAN 指令控制的连接方式说明连接硬件
- 在 freertos.c 文件中使用宏定义 `#define CAN_CONTROL`，屏蔽 `#define PWM_CONTROL`
- 编译工程，并下载程序到 RM 主控板
- 通过 RM 主控板上的按键操作，即可对电机进行调速

2. 使用 PWM 信号控制电机转动

- 按照硬件连接说明中的使用 PWM 信号控制的连接方式连接硬件
- 在 freertos.c 文件中使用宏定义 `#define PWM_CONTROL`，屏蔽 `#define CAN_CONTROL`

CAN_CONTROL

- c. 编译工程，并下载程序到 RM 主控板
- d. 通过 RM 主控板上的按键操作，即可对电机进行调速

注意: 请使用动力电池为该系统供电(如果使用电源箱 ,可能因电机反电动势过高 ,造成 RM 主控开发板中做电源保护的二极管被击穿)。