

TI MSPM0G3507 云台控制系统源码使用说明书 (CCS版)

简介： 本工程基于德州仪器 (TI) **MSPM0G3507** 微控制器开发，专为高精度云台及多轴电机控制设计。项目采用 **CCS (Code Composer Studio)** 开发，利用 **SysConfig** 进行底层配置。核心集成了 **Emm_V5** 闭环步进电机 驱动协议、**PID** 算法、**OLED** 显示 及 时间片调度器，代码结构清晰，非常适合用于电赛控制题或机器人关节控制。

* 1. 快速上手：如何使用 CCS 打开工程？

⚠ 警告：不要直接双击文件夹里的文件打开！请务必按照以下步骤导入工程。

- ① 启动软件：打开 Code Composer Studio (CCS)。
- ② 导入步骤：
 - 点击菜单栏左上角的 **File** -> **Import ...**。
 - 在弹窗中选择 **C/C++** -> **CCS Projects**，然后点击 **Next**。
 - 点击 **Browse...** 按钮，找到本工程的文件夹（选择包含 **.ccsproject** 文件的 **Template_Yuntai** 目录）。
 - 在 "Discovered projects" 列表中，勾选 **Template_Yuntai**。

- 点击 **Finish** 完成导入。
- ③ 编译检查：点击工具栏上的“锤子”图标 ，确保 Output 窗口显示 "Build Finished" 且无错误。
- ④ 可视化配置 (**SysConfig**)：
 - 在左侧资源管理器中找到并双击 **empty.syscfg**。
 - 这里是所有引脚 (GPIO, UART, I2C, PWM) 的“控制中心”。修改引脚定义不需要改代码，直接在这里修改即可！

✳️ 2. 代码架构梳理 (文件导航)

为了让你一眼看懂代码，我们将功能分为了不同层级：

- **My_App** (应用层 - 核心业务逻辑)
 - **scheduler.c**：总调度器。相当于操作系统的“心脏”，负责每隔特定时间（如10ms、100ms）执行一次任务。
 - **motor_app.c**：云台运动逻辑。负责计算云台需要转到什么角度，协调各个电机的动作。
 - **oled_app.c**：界面显示逻辑。决定屏幕上显示什么菜单、参数或状态。
 - **uart_app.c**：通信任务。处理与外部设备（如视觉模块、上位机）的数据交互。
- **My_Driver** (驱动层 - 硬件翻译官)
 - **EmmV5.c**：核心驱动。Emm_V5 闭环步进电机的通信协议库。封装了位置控制、速度控制等指令。
 - **pid_driver.c**：PID 算法库。通用的 PID 计算公式实现。
 - **oled_hardware_i2c.c**：OLED 底层。负责通过 I2C 协议向屏幕发送数据。
- **MSPM0** (系统层)

- `interrupt.c` : 中断服务。定时器中断（用于调度器）和串口中断（用于接收电机反馈）的入口。
- `clock.c` : 系统时钟初始化配置。

✳️ 3. 硬件接口清单 (去哪里看引脚？)

由于 MSPM0 的引脚配置非常灵活，最准确的引脚定义请直接查看工程中的 `empty.syscfg` 文件。

以下是模板默认的典型连接方式（仅供参考，以 SysConfig 为准）：

外设模块	功能	涉及引脚 (SysConfig 名称)	说明
UART (Serial)	电机通信	<code>UART_Motor_TX</code> / <code>RX</code>	核心接口。连接 Emm_V5 转接板。注意 TX 接 RX, RX 接 TX。
I2C	OLED 屏幕	<code>I2C_OLED_SDA</code> / <code>SCL</code>	连接 0.96 寸 OLED 屏幕 (GND, VCC, SCL, SDA)。
UART (Serial)	调试/视觉	<code>UART_Debug_TX</code> / <code>RX</code>	连接电脑串口助手或 OpenMV/树莓派。
GPIO (Input)	按键输入	<code>KEY1</code> , <code>KEY2</code>	用于切换模式或调整 PID 参数。
GPIO (Output)	LED 指示	<code>LED1</code>	系统运行心跳灯。

✳️ 4. 核心功能与 API 调用指南

这里列出了控制云台最常用的“大招”，复制这些函数即可控制硬件。

4.1 闭环电机控制 (`EmmV5.h`)

这是本工程的核心。Emm_V5 闭环步进电机通过串口指令控制，精准且力矩大。

- 初始化

- `void Emm_V5_Init(void)`
 - 位置：在 `motor_app.c` 或初始化阶段调用。
 - 作用：初始化对应的 UART 串口。
- 位置控制 (指定角度)
 - `void Emm_V5_Pos_Control(uint8_t id, uint8_t dir, uint16_t vel, uint8_t acc, uint32_t clk)`
 - 参数详解：
 - `id`：电机 ID (默认 1，多电机需设置不同 ID)。
 - `dir`：方向 (0:逆时针, 1:顺时针)。
 - `vel`：速度 (RPM, 转/分)。
 - `acc`：加速度 (0-255, 0 代表最大加速度)。
 - `clk`：脉冲数 (电机细分后的步数，例如 3200 步一圈)。
 - 实战代码：


```
// 让 ID为1 的电机，顺时针，以 100RPM 的速度，转动 360 度（假设细分16，步距角1.8，则一圈3200步）
Emm_V5_Pos_Control(1, 1, 100, 0, 3200);
```
- 速度控制 (持续转动)
 - `void Emm_V5_Vel_Control(uint8_t id, uint8_t dir, uint16_t vel, uint8_t acc)`
 - 实战代码：


```
// 让 ID为1 的电机以 500RPM 持续旋转
Emm_V5_Vel_Control(1, 0, 500, 0);
```
- 停止
 - `void Emm_V5_Stop_Now(uint8_t id)`：立即急停。

4.2 PID 算法调用 (`pid_driver.h`)

如果你需要做更复杂的闭环（例如配合陀螺仪控制云台稳定），会用到 PID。

- 计算控制量

- `float PID_Realize(PID_TypeDef *pid, float target, float feedback)`

- 用法：在定时器中断或调度任务中调用。

- 示例：

```
float output = PID_Realize(&Pitch_PID,  
                           target_angle, current_angle);  
// 然后将 output 转换为电机速度发送给 EMM_V5
```

4.3 任务调度器 (`scheduler.h`)

不要在 `while(1)` 里写延时死循环，请使用调度器来管理任务。

- 添加新任务：

- 1 打开 `MSPM0/scheduler.c`。
- 2 找到 `Scheduler_Task_List` 数组。
- 3 添加你的任务函数和周期（单位 ms）。

```
static scheduler_task_t Scheduler_Task_List[] =  
{  
    {Motor_Process_Task, 10, 0}, // 10ms 执行一次电  
     机逻辑  
    {OLED_Show_Task, 100, 0}, // 100ms 刷新一次屏  
     幕  
    {My_Custom_Task, 500, 0}, // <--- 你添加的任  
     务：每500ms执行一次  
};
```

4.4 OLED 显示 (`oled_app.h`)

- `OLED_ShowString(x, y, "String")` : 显示文字。

- `OLED_ShowNumber(x, y, number, len, size)` : 显示整数。

- 注意：所有显示内容建议放在 `OLED_Show_Task` 中统一刷新，避免屏幕闪烁。

✳️ 5. 常见问题 (FAQ)

- **Q:** 电机完全不转?
 - A1: 检查 电机供电。Emm_V5 通常需要 12V-24V 独立供电，不能只靠开发板的 5V。
 - A2: 检查 串口线序。开发板的 TX 接驱动板的 RX，RX 接驱动板的 TX。
 - A3: 检查 电机 ID。代码里的 ID 必须和电机内部参数设置的一致（默认通常是 1）。
- **Q:** 编译报错 "Symbol ... not found" ?
 - A: 如果你刚修改了 `.syscfg` 文件，CCS 有时需要一点时间重新生成代码。尝试点击 `Project` -> `Clean`，然后重新 `Build`。
- **Q:** 烧录后程序不运行?
 - A: 检查 `main.c` 中是否开启了总中断 `NVIC_EnableIRQ(...)` 以及是否调用了 `SYSCFG_DL_init()`。本模板默认已配置好。
- **Q:** 如何修改串口波特率?
 - A: 不要去代码里找！双击 `empty.syscfg`，点击左侧的 `UART` 模块，在右侧配置界面直接修改 `Baud Rate` 即可。

祝您开发顺利，云台稳如老狗！

