M16 软件设计

1. 振镜

1.1. MCU 端

源码文件： src\rp2040\xy2_100.c \src\rp2040\xy2_stepper.c
xy2_100.c 主要用 RP2040 来 PIO 驱动 XY2 协议（基于第一版来修改）
xy2_stepper.c 基于 stepper.c 实现相关函数来修改的

函数：
int update_vir_postion_info(uint8_t gpio_step_num, uint32_t position, uint16_t count, uint8_t mode)
功能：当振镜移动指令，BC 轴位置有变化，该函数把相应位置信息映射到振镜的位置。
参数：
uint8_t gpio_step_num 区分那个轴，根据配置来定，B 轴为振镜 X 轴、C 轴为振镜 Y 轴。（gpio_step_num 的值小为 X 轴）
uint32_t position 振镜移动，记录当前位置
uint16_t count 移动过程中 count 动态变化信息
uint8_t mode 区分 count 的模式（single、double）。

1.2. HOST 端

源码文件： klippy\extras\galvo_config.py （add）
galvo_config.py 实现振镜的驱动 IO 的配置，命令 SET_POS_GALVO 控制振镜的位置定点，查询 QUERY_POS_GALVO 振镜的位置。命令：SET_POS_GALVO B=0 C=0
配置振镜的参数在[printer]打印机控制的高级设置的：
hradiation_angle: 0.35 //振镜的角度（弧度单位）
focus_distance: 160 //焦距 （毫米单位）
magnify_factor: 100 //放大倍数 （角度值取值范围小，通过放大系数可以放大，获取好的精度）

2. 扩展 ABC 轴

基于 G1 命令-振镜处理部分信息.pdf 文档需求来修改相关代码。

涉及修改源码文件：klippy\chelper__init__.py　itersolve.c　itersolve.h
trapq.c　trapq.h　kin_galvo.c （add）
klippy\kinematics\corexy_galvo.py （add）
klippy\extras\gcode_move.py　gcode.py
klippy\steppery.py

振镜运动方程： y=f * θ 的镜头组（f 为焦距， θ 为振镜偏转角度）。
针对 BC 轴，在 MCU 中通 update_vir_postion_info 函数映射实际振镜上。 配置 BC 轴的电机需要使用龙头集板的 RP2040 上的 GPIO 口。并且需要配置虚拟电机类型，一定要配置 step_mvirtualmode: 1

3. E 轴关联激光输出

基于 M16 项目概括_08062024.pdf 文档 P21 页需求来修改相关代码。

3.1. MCU 端

源码文件： src\rp2040\stepper_pwm.c
stepper_pwm.c 基于 stepper.c 实现相关函数来修改的
函数:
void update_next_pwm_ctrl_data(uint8_t runstep, uint16_t count, uint32_t inter_pulse_ticks)
功能： E 轴的运动，转换成调激光功能
参数：
uint8_t runstep　表示 E 轴运动步进阶段
uint8_t runstep　表示 E 轴 count
uint32_t inter_pulse_ticks　当前运动快慢

增加指令：
“set_pwm_onf oid=%c onf=%c”，根据 G0 和 G1 来给 onf 转不同值。
“set_pwm_power oid=%c mod=%c pwmv=%hu pticks=%u”，根据模式 M3 或 M4，当前功率，在 M4 模式当前速度计算出 interval，在计算变化功率时，这个值为分子。实际运动过程 interval 值当分母。

3.2. HOST 端

源码文件： klippy\kinematics\extruder.py
klippy\chelper__init__.py
klippy\chelper\trapq.c trapq.h
klippy\chelper\stepcompress.c stepcompress.h
klippy\chelper\itersolve.c

klippy\extras\gcode_move.py
klippy\toolhead.py
klippy\stepper.py

klippy\extras\force_move.py
klippy\extras\manual_stepper.py

针对 E 轴，在 MCU 中通 set_pwm_pulse_width（半导体激光）
或 set_pwm_pulse_width_fiberlaser（光纤激光）
函数映射激光上。 配置 E 轴的电机需要配置与激光控制同一个 MCU 的上的 GPIO 口。并且需要配置虚拟电机类型，一定要配置 step_mvirtualmode: 2

4. 增加光纤激光支持

4.1. MCU 端

源码文件： src\rp2040\fiberlaser_ctrl.c fiberlaser_ctrl.h

src\rp2040\xy2_100.c 中 增加在 PIO1 上两个状态机。一个状态机实现脉冲重复频率 PRR，主要函数 pwm_program_init。另一个状态机实现动态调功率，主要函数 fiber_set_power_program_init。

src\rp2040\stepper_pwm.c 中 根据 printer_*.cfg 配置选择当前激光类型：半导体激光、光纤激光，调用不同控制函数。

光纤激光控制函数：

```
void direct_set_pwm_pulse_width_fibertype(uint8_t pwd_oid, uint32_t val,
    uint8_t   pwm_on_off);
    uint8_t pwd_oid    指定光纤激光
    uint32_t val          光纤激光的强度
    uint8_t   pwm_on_off  光纤激光开关
```

4.2. HOST 端

源码文件： klippy\extras\fiblaser_link.py （add）

根据不同类型激光，选择不同配置文件。 fiblaser.cfg 是 光纤激光配置文件；pwmlaser.cfg 是半导体激光配置文件。

[include fiblaser.cfg]

#[include pwmlaser.cfg]

fiblaser_link.py 通过 FiberLaserLink 类函数_handle_connect_bind，配置 E 轴与光纤激光关联。

5. 主机升级 MCU 固件

按 BOOTSEL 键上电，让 MCU 进入升级 U 盘模式；在主机端终端上用命令 lsblk 查看设备，一般类似 /dev/sda1 设备。

sudo mkdir /mnt/usb （运行一次，如果/mnt/usb 目录存在，就不行再运行）

sudo mount /dev/sda1 /mnt/usb //挂载 U 盘到/mnt/usb 目录

sudo cp 更新固件文件名 /mnt/usb //等待文件拷贝完成

最后卸载：

sudo umount /mnt/usb

6. 版本说明：

版本	日期	描述
V1.0	2024/09/29	初始版本

7. 参考文档

1、XY2-100 协议，网上链接：doc.php (newson.be)，本地查看
2、光纤激光协议，网上链接：MANUAL (newson.be)，本地查看
3、声光调 Q 脉冲光纤激光器:本地查看