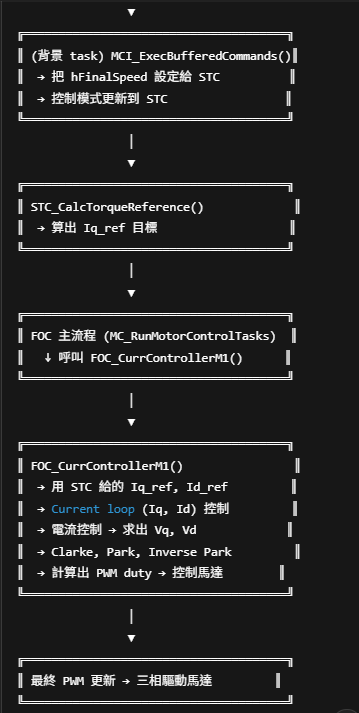
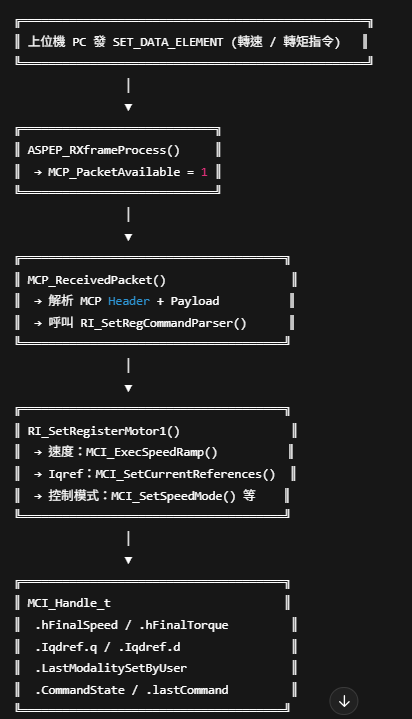
外部指令控制馬達的Code流程

上位機 => MotorPilot



1. **MototPilot 能發送的資料(原始資料 包含配對、連線確認)**

BEACON 初始化配對 #define BEACON (0x5)

PING 維持連線 #define PING (0x06)

DATA\_PACKET 同步控制資料包（例如轉速、轉矩命令） #define DATA\_PACKET (0x09)

NACK 回傳錯誤 #define NACK (0xF)

DATA\_PACKET

------------------------------------------------------------------------------------------------------------

| Header (4 bytes / 32 bit) | Data Payload (N bytes) | CRC (2 bytes, optional) |

------------------------------------------------------------------------------------------------------------

Header 結構

Bit 0~3 : PacketType → ID\_MASK (0xF)，例 DATA\_PACKET == 0x9

Bit 4~16 : Length 欄位 (packet length in bytes)

Bit 17~27 : Reserved

Bit 28~31 : CRC-4 of header

1. **ASPEP**

Aspep.c /ASPEP\_RXframeProcess() 確認和解析MoterPilot 傳來的原始封包 🡪 把DATA\_PACKET 放入buf緩衝區 🡪

1. **MCP**

🡪再由mcp.c /MCP\_ReceivedPacket 查看buf緩衝區的packet並使用 🡪

判斷packet內容後呼叫對應處理：

- RI\_SetRegCommandParser() → 設定 Register（如轉速命令）

- MCI\_StartMotor()

- MCI\_StopMotor()

- MCI\_StopRamp()

- MCI\_Clear\_Iqdref()

🡪 RI\_SetRegCommandParser() 決定 regID 並呼叫 RI\_SetRegisterMotor1()

🡪 RI\_SetRegisterMotor1() 根據 regID → 決定要設定 PID 參數 / 控制模式 / Iqref / 速度 ramp / Torque ramp 再將參數寫入相關結構 🡪

(背景LOOP task) mc\_interface.h/ MCI\_ExecBufferedCommands()

1. **STC**

MCI\_ExecBufferedCommands() 🡪 speed\_torq\_ctrl.c/ STC\_SetControlMode +

STC\_ExecRamp (寫入hFinalSpeed / hFinalTorque )

1. **MAIN LOOP**

在mc\_tasks.c/MC\_RunMotorControlTasks中呼叫 mc\_tasks\_foc.c/TSK\_MediumFrequencyTaskM1();

🡪 TSK\_MediumFrequencyTaskM1() 呼叫MCI\_ExecBufferedCommands() & FOC\_CalcCurrRef(M1);

🡪 FOC\_CalcCurrRef(M1) 呼叫 STC 算出 Iq\_ref (hTeref)並寫入FOCVars[bMotor].Iqdref

1. **呼叫FOC\_CurrControllerM1(); 🡪 控制馬達**

(ADC轉換完成的中斷)stm32g4xx\_mc\_it.c/ ADC1\_2\_IRQHandler 中呼叫TSK\_HighFrequencyTask()

🡪 mc\_tasks.c/TSK\_HighFrequencyTask()呼叫mc\_tasks\_foc.c/ FOC\_HighFrequencyTask()

🡪 FOC\_HighFrequencyTask() 呼叫 FOC\_CurrControllerM1();

**Torque補充**

上位機（MotorPilot / GUI） → 發指令（例如 SetTorqueRamp）

→ ASPep RX ISR → aspep.c 處理封包

→ RI\_SetRegisterMotor1()

→ MC\_ProgramTorqueRampMotor1\_F()

→ MCI\_ExecTorqueRamp\_F()

float\_t hFinalTorque = (FinalTorque \* (float\_t)CURRENT\_CONV\_FACTOR);

MCI\_ExecTorqueRamp(pHandle, (int16\_t)hFinalTorque, hDurationms);

→ MCI\_ExecTorqueRamp()

→ 設定 Iqref ramp → 控轉矩