

하드웨어 제원서

조향장치

13.02.2026

배한진

iwdhanjin@gmail.com | 문서 번호: HW-SPEC-001 | 버전: 1.0 | 2026-02-13

목차

1	개요	1
1.1	적용 범위	1
1.2	참조 문서	1
2	시스템 개요 및 동작 흐름	1
2.1	동작 순서 요약	5
2.2	주요 하드웨어 구성	6
3	MCU 사양	6
4	메모리	6
5	핀 배치	7
5.1	GPIO 및 기능 할당	7
5.2	GPIO 설정 요약	7
6	주변장치	8
6.1	UART	8
6.2	DMA	8
6.3	타이머	8
6.4	인터럽트	8
7	드라이브 및 패키지 사용	9
7.1	스테퍼 모터 드라이브 (L298)	9
7.2	릴레이 패키지 (SN74HC574)	9
7.3	펌프 제어 패키지 (pump_ctrl + TIM5)	9
8	통신 프로토콜 인터페이스	9
8.1	물리 계층	9
8.2	프레임 형식 (AA 33 프로토콜)	10
8.3	명령 코드 (CMD)	10
8.4	응답 코드 (RSP)	10
8.5	소프트웨어 인터페이스	11
9	개발 환경	11
10	문서 이력	11

1 개요

본 문서는 조향장치 프로젝트에서 사용하는 하드웨어의 제원을 정의한다. 메인 컨트롤러는 STMicroelectronics의 STM32F446RETx를 사용하며, 2축 스템퍼 제어 및 RS485 통신, 오리진 센서 입력을 지원한다.

1.1 적용 범위

- MCU 및 메모리 사양
- 핀 배치 및 GPIO 할당
- 사용 주변장치(UART, TIM, DMA) 설정
- 개발/빌드 환경

1.2 참조 문서

- STM32F446xx Reference Manual (ST)
- STM32F446RETx Datasheet
- control_v1.ioc (STM32CubeIDE 설정, 조향장치 프로젝트)

2 시스템 개요 및 동작 흐름

본 시스템은 T-rail 위에 공병(빈 병)을 올려놓고, Y축으로 맨 뒤까지 이동한 뒤 2 mm 후진한 다음, X-Y축을 이동시키며 노즐 1번부터 36번 밑으로 순차 이동하여 각 노즐에서 향수 등을 분사하는 방식이다.

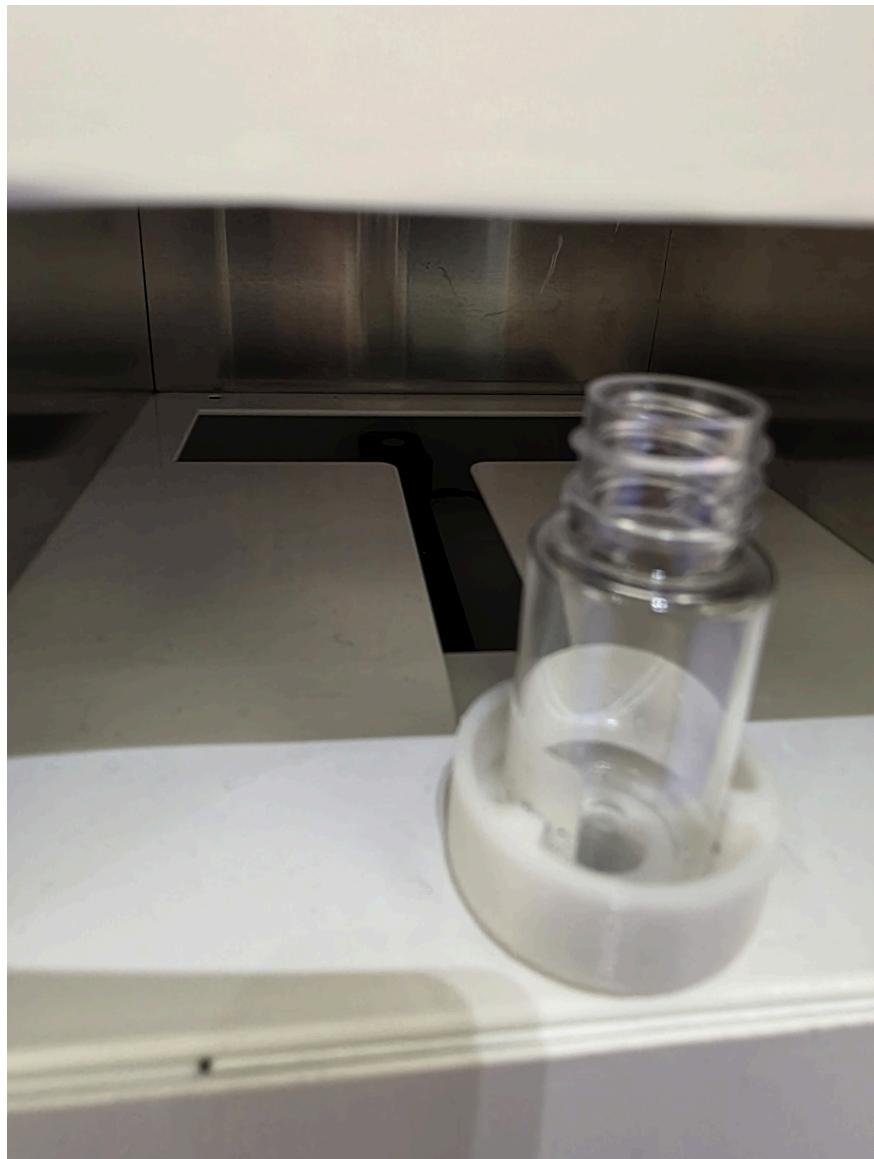


그림 1: T-rail: 공병을 옮겨 X·Y축 이동으로 노즐 밑으로 이송하는 레일/플랫폼

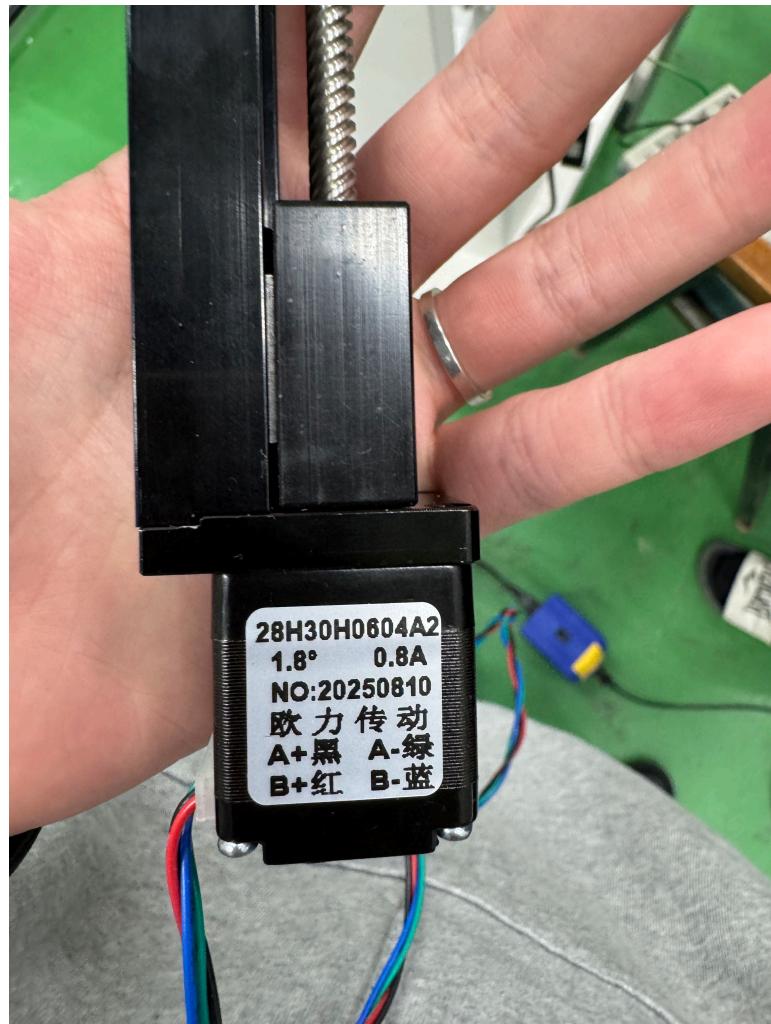


그림 2: 스텠퍼 모터 28H30H0604A2 (1.8° , 0.8 A). 리드스크류 직결, 선형 구동용.

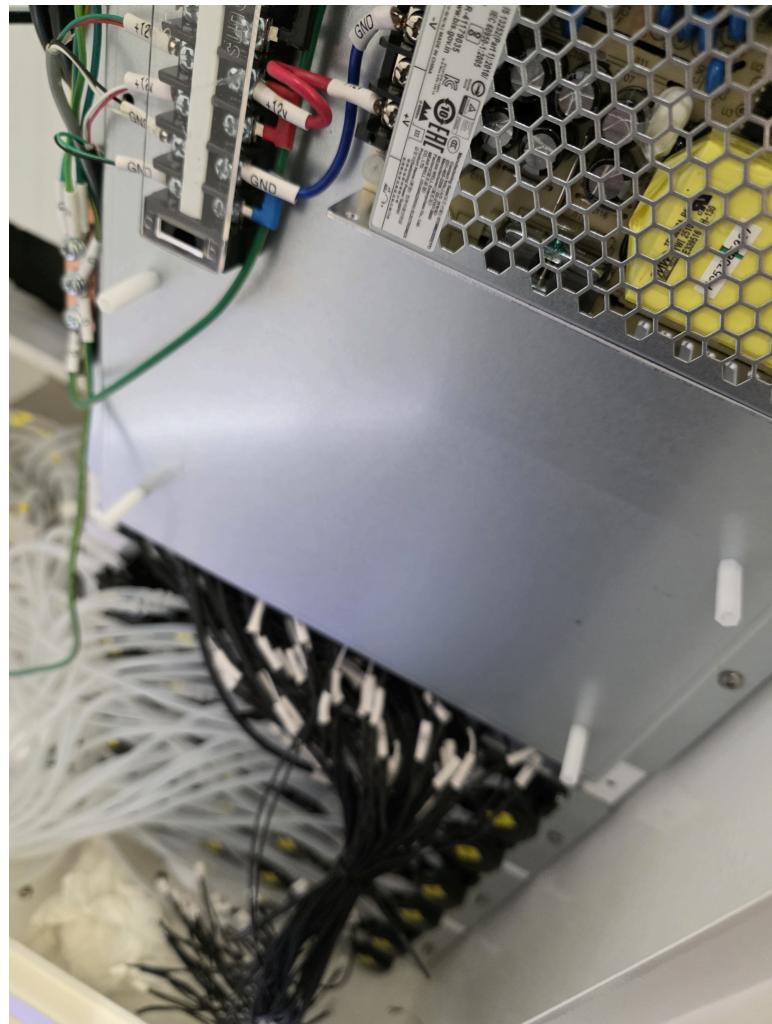


그림 3: 펌프/DC 모터 및 전원: MEAN WELL 스위칭 전원 (+12 V), 단자대 배선, 유체 투브 및 전선 번들.

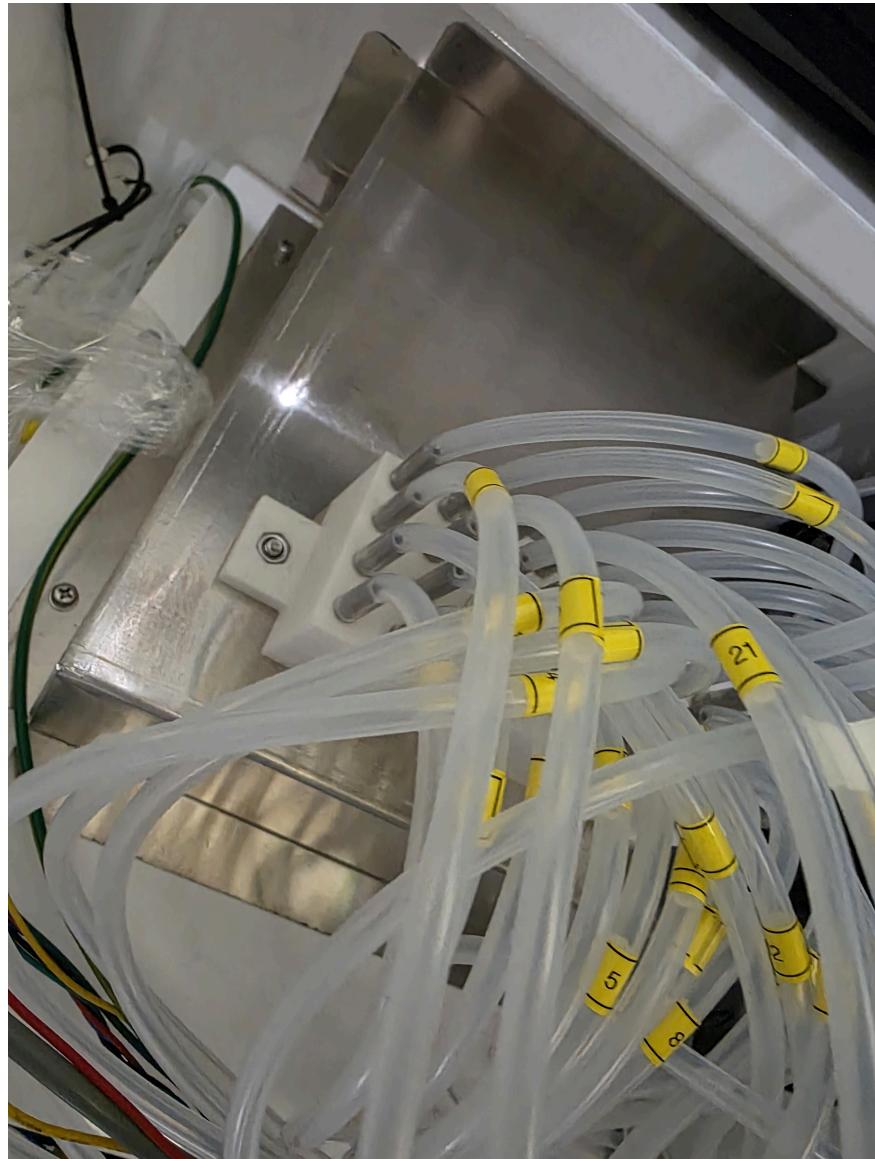


그림 4: 노즐 매니폴드: 1 ~ 36번 채널이 노란 띠로 표시된 투명 투브로 각 노즐에 유체 공급.

2.1 동작 순서 요약

1. 공병 적재 — T-rail 위 흰색 원형 홀더에 공병을 세움.
2. Y축 후방 이동 — Y축 스테퍼로 공병을 맨 뒤(원점 방향)까지 이동.
3. 2 mm 후진 — 리미트 도달 후 2 mm만큼 후진하여 오리진 스위치에서 이탈.
4. X·Y축 스캔 — X·Y축을 이동시키며 병을 노즐 1번 아래부터 36번 아래까지 순차 위치시킴.
5. 분사 — 각 위치에서 해당 노즐(펌프)으로 정량 분사 (통신 프로토콜 CMD_MAKE_PERFUME 등으로 제어).

2.2 주요 하드웨어 구성

구성요소	설명
T-rail	공병 이송용 레일/플랫폼. 그림 1 참조.
스테퍼 모터	모델 28H30H0604A2. 1.8° 스텝, 0.8 A/상. A+黑 A-绿 B+红 B-蓝. 리드스크류 직결 선 형 구동. 그림 2
펌프/전원	MEAN WELL 스위칭 전원 +12 V. 단자대로 배분. 펌프·DC 모터 및 유체 투브. 그림 3
노즐 매니폴드	1~36번 채널, 노란 띠 번호 표시. 투명 투 브로 각 노즐에 유체 공급. 그림 4

3 MCU 사양

항목	내용
Part Number	STM32F446RET6
패키지	LQFP64
코어	ARM Cortex-M4F, FPU, 180 MHz max
제조사	STMicroelectronics
패밀리	STM32F4 Series

4 메모리

메모리	주소	용량
Flash	0x0800_0000	512 KB
RAM	0x2000_0000	128 KB

링커 스크립트 기준 최소 힙/스택: Heap 0x200 (512 B), Stack 0x400 (1 KB). 애플리케이션 요구에 따라 조정 가능.

5 핀 배치

5.1 GPIO 및 기능 할당

포트	핀	라벨	기능
PA	0	WKUP	TIM5 CH1 (PWM 입력 등)
PA	2	USART2_TX	UART2 송신
PA	3	USART2_RX	UART2 수신
PA	4	X_A	X축 스텔퍼 A상 출력
PA	5	X_B	X축 스텔퍼 B상 출력
PA	6	X_A_	X축 스텔퍼 A상 (2nd 드라이버)
PA	7	X_B_	X축 스텔퍼 B상 (2nd 드라이버)
PA	8	RTX_EN_485	RS485 방향 제어 (TX Enable)
PA	9	USART1_TX	UART1 송신 (RS485)
PA	10	USART1_RX	UART1 수신 (RS485)
PA	11	X_ORG_SWIN	X축 오리진 센서, EXTI, Pull-up
PA	13	SWDIO	디버그 SWD
PA	14	SWCLK	디버그 SWD
PB	1	Y_ORG_SWIN	Y축 오리진 센서, EXTI, Pull-up
PB	4	Y_A	Y축 스텔퍼 A상 출력
PB	5	Y_B	Y축 스텔퍼 B상 출력
PB	6	Y_A_	Y축 스텔퍼 A상 (2nd 드라이버)
PB	7	Y_B_	Y축 스텔퍼 B상 (2nd 드라이버)
PH	0	OSC_IN	외부 발진자 입력
PH	1	OSC_OUT	외부 발진자 출력

5.2 GPIO 설정 요약

- 스테퍼 출력(X_A, X_B, Y_A, Y_B 등): Push-Pull, Very High Speed, Pull-up (해당 시).

- 오리진 스위치(X_ORG_SWIN, Y_ORG_SWIN): EXTI, Falling Edge, Pull-up.
- RTX_EN_485: Push-Pull, Very High Speed, 기본 Low(수신 대기).

6 주변장치

6.1 UART

인스턴스	용도	비고
USART1	RS485 통신	TX/RX: PA9/PA10, 방향: PA8 (RTX_EN_485)
USART2	일반 UART	TX/RX: PA2/PA3

모드: Asynchronous. 전송 속도(baud rate) 등은 소스 코드 및 CubeMX 설정에 따름.

6.2 DMA

요청	인스턴스	방향	모드
USART1_RX	DMA2 Stream 2	Periph → Memory	Circular
USART2_RX	DMA1 Stream 5	Periph → Memory	Circular

정렬: Peripheral/Memory 모두 Byte. 우선순위: Very High.

6.3 타이머

타이머	역할	설정
TIM5	PWM 출력 (CH1)	PA0, Prescaler=2-1, Period=1000-1, 내부 클록
TIM6	베이스 타이머	Prescaler=1000-1, Period=3-1

6.4 인터럽트

- EXTI1: Y_ORG_SWIN (PB1)
- EXTI15_10: X_ORG_SWIN (PA11)
- DMA1_Stream5: USART2 RX
- DMA2_Stream2: USART1 RX
- TIM5, TIM6_DAC: 타이머 인터럽트
- 우선순위 그룹: NVIC_PRIORITYGROUP_4

7 드라이브 및 패키지 사용

7.1 스텝 모터 드라이브 (L298)

- 드라이버 보드: L298 기반 큰 모터 드라이버 보드 사용 (L298_Drv 정의).
- 모터: 4선 2상 스텝 모터, X축·Y축 각 1대. 1상 여자(Full Step), 2상 여자, 1-2상 여자 (Half Step) 시퀀스 지원.
- **GPIO:** X축 PA4/PA5/PA6/PA7 (X_A, X_B, X_A, X_B), Y축 PB4/PB5/PB6/PB7 (Y_A, Y_B, Y_A, Y_B). 포트 ODR 하위 4비트로 1상 스텝 출력.
- **패키지:** Core/My_Lib/ST_MOTOR/ (st_motor.c, st_motor.h). 흄은 오리진 스위치 폴링 + 리리즈(backoff) 후 Stm_X_0rg_move() / Stm_Y_0rg_move() 로 수행. 50 pulse = 1 mm 기준, X_Start=3000(60 mm), Y_Start=8500(170 mm) 등 절대 펄스 값 사용.

7.2 릴레이 패키지 (SN74HC574)

- 구성: SN74HC574 5개 → 40 출력 중 36채널 사용. 채널 번호 1 ~ 36 (1-based).
- 연결: 데이터 버스 PC0 ~ PC7 (8비트), 래치 클록 CLK1 ~ CLK5 → PB11 ~ PB15 (574별 1개씩).
- **API:** Relay574_Init(), Relay574_AllOff(), Relay574_Set(ch, on), Relay574_Get(ch). 펌프 선택 시 해당 채널만 ON, 작업 종료 시 AllOff.

7.3 펌프 제어 패키지 (pump_ctrl + TIM5)

- **PWM:** TIM5 CH1 (PA0), 블루터 0 ~ 999 (Period=1000-1). 펌프 속도 제어.
- 방향: PC8(CW), PC9(CCW). PumpDir에 따라 한쪽만 활성화.
- 풀로우: Pump_Init() → Pump_QueueMl(pump_no, dir, ml)로 1 ~ 36번 펌프·ml 큐잉 → Pump_StartQueued()로 실행. 펌프별 유량 테이블(ml/ms)로 시간 환산 후 TIM5 PWM + Relay574 채널 선택으로 동작. Pump_Tick(now_ms)로 주기 호출.

8 통신 프로토콜 인터페이스

8.1 물리 계층

통신	UART / 핀	역할
호스트 프로토콜	USART2, PA2(TX)/PA3(RX)	USBシリアル変換 연결. AA 33 ... FC 프로토콜 수신·응답. Receive-to-Idle DMA 사용.
RS485	USART1, PA9(TX)/PA10(RX), PA8(RTX_EN_485)	방향 제어: TX 시 PA8=High, RX 대기 시 Low. My_uart (ch2)에서 사용.

8.2 프레임 형식 (AA 33 프로토콜)

- 프레임 구조: ST0 ST1 | ID | CMD | LEN | DATA... | CRC16(LE) | END
 - ST0 ST1 = 0xAA 0x33, END = 0xFC.
 - CRC 범위: ID ~ DATA (LEN 포함). CRC16 Modbus (poly 0xA001), Little Endian 2바이트.
- 형식 A: LEN 있음 (2+1+1+1+LEN+2+1 바이트). 데이터 있는 명령.
- 형식 B: LEN 없음 (2+1+1+2+1 = 7바이트). 데이터 없는 명령. 오인 파싱으로 릴레이 차터링 방지용으로 두 형식 모두 파싱.

8.3 명령 코드 (CMD)

명령	코드	비고
CMD_MAKE_PERFUME	0x80	N×4바이트: perfume_no(16b BE), vol_0.1ml(16b BE). N=1 ~ 5. 펌프 큐 후 DONE까지 대기.
CMD_MOVE_ZEROPOINT	0x10	ACK 후 미구현 (ERR_UNSUP).
CMD_MOVE_HOME	0x11	ACK 후 미구현 (ERR_UNSUP).
CMD_MOVE_POINT	0x12	ACK 후 미구현 (ERR_UNSUP).
CMD_PUMP_START	0x20	ACK 후 미구현 (ERR_UNSUP).
CMD_PUMP_STOP	0x21	ACK 후 미구현 (ERR_UNSUP).
CMD_CATE_CHANGE	0x30	ACK 후 미구현 (ERR_UNSUP).

8.4 응답 코드 (RSP)

응답	코드	DATA 내용
RSP_ACK	0xC0	req_cmd, status (0=OK, 1=BUSY 등)
RSP_DONE	0xC1	req_cmd, result
RSP_ERR	0xC2	req_cmd, err_code, detail

에러 코드: BAD_FRAME(0x01), BAD_CRC(0x02), BAD_LEN(0x03), UNSUP(0x04), BUSY(0x05), PARAM(0x06).

8.5 소프트웨어 인터페이스

- 초기화: Proto_Init(&huart2, 0x01) — USART2 사용, 슬레이브 ID 0x01.
- 수신: UART Rx 콜백(또는 Receive-to-Idle 완료)에서 수신 바이트를 Proto_OnRxBytes(data, len)으로 전달. 내부 512바이트 버퍼, 프레임 동기 후 consume_one_frame()으로 처리.
- 주기: 메인 루프에서 Proto_Tick(HAL_GetTick()) 호출. MAKE_PERFUME 등 비동기 동작 완료 시 RSP_DONE 전송.

9 개발 환경

항목	내용
IDE/설정	STM32CubeIDE 6.15.0, control_v1.ioc (조향장치)
툴체인	ARM GCC (arm-none-eabi-gcc) 14.2.1-1.1.1
디버그	OpenOCD 0.12.0-7.1, SWD (PA13/PA14)
HAL	STM32F4xx HAL
링커 스크립트	STM32F446RETX_FLASH.ld

환경 변수는 프로젝트 루트의 .stm32env에서 참조 가능하다 (ARM_GCC_PATH, OPENOCD 등).

10 문서 이력

버전	날짜	변경 내용
1.0	2026-02-13	초안 작성 (조향장치 기준)