

ARM Coretex-M

# 펌웨어 설계



## 디바이스 드라이버 API 구조



한국기술교육대학교  
온라인평생교육원

### 학습목표

- 디바이스 드라이버 및 STM32 라이브러리의 HAL 드라이버에 대해 설명할 수 있다.
- HAL 드라이버 구조에 대해 설명할 수 있다.

### 학습내용

- 디바이스 드라이버 API의 소개
- HAL 드라이버 분석

# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🌈 디바이스 드라이버의 정의

#### 디바이스 드라이버

하드웨어나 장치를 제어하기 위한 소프트웨어

- ... 주로 운영체제가 존재하는 시스템에서 운영체제에 포함되어 있음
- ... 운영체제가 없더라도 하드웨어를 제어하는 소프트웨어를 통틀어 디바이스 드라이버라고 함
- ... 같은 하드웨어 장치라도 리눅스는 리눅스용 디바이스 드라이버, 윈도우는 윈도우용 디바이스 드라이버가 필요함

예

> Software Library

LAN9000 Diagnostic and Setup Utilities (ETX, ISMC9000, MIIPHY, CIS and ENABLERS)  
 LAN91C111 DOS ODI Driver Version 4.5  
 LAN91C111 Linux Driver Version 2.03 Source  
 LAN91C111 Packet Driver Version 1.2C  
 LAN91C111 VxWorks Driver Version 2.1  
 LAN91C111 Windows 98 Driver Version 3.12B  
 LAN91C111 Windows CE Driver 5.04/4.x for X86  
 LAN91C111 Windows CE Driver Version 3.0  
 LAN91C111 Windows XP Embedded Driver Version 5.1.1.0

LAN91C111칩의 디바이스 드라이버들

- ... 예를 들어 LAN91C111이라는 이더넷 컨트롤러 칩의 디바이스 드라이버는 리눅스용 드라이버 뿐만 아니라 VxWorks라는 RTOS용 드라이버, 윈도우용 드라이버 등이 따로 제공됨

# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🎯 디바이스 드라이버의 역할



리눅스의 예로 디바이스 드라이버의 역할을 상세히 알아봄

## 리눅스 커널

### 커널 코어

- ... 멀티 프로세서 제어와 같은 운영체제 본연의 역할을 하는 소프트웨어

### 디바이스 드라이버

- ... 하드웨어 디바이스를 직접 제어하는 소프트웨어

# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🌈 디바이스 드라이버의 역할



리눅스의 예로 디바이스 드라이버의 역할을 상세히 알아봄

## 리눅스 커널

### 커널 코어

- ... 멀티 프로세서 제어와 같은 운영체제 본연의 역할을 하는 소프트웨어

### 디바이스 드라이버

- ... 하드웨어 디바이스를 직접 제어하는 소프트웨어

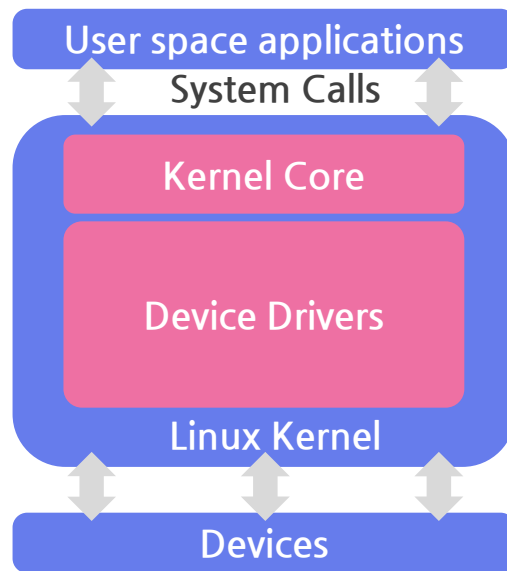
# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🌈 디바이스 드라이버의 역할

#### 리눅스 커널상의 디바이스 드라이버



- ...> 리눅스 커널은 사용자 영역의 어플리케이션과 시스템 콜이라는 인터페이스로 통신함
- ...> 실제로 리눅스 커널의 수만 개의 C 언어 소스 파일로 구성되어 있고 그 중 70~80%가 디바이스 드라이버 소스 파일임
  - 실제 사용하지 않는 오래된 디바이스의 드라이버도 포함되어 있기 때문임

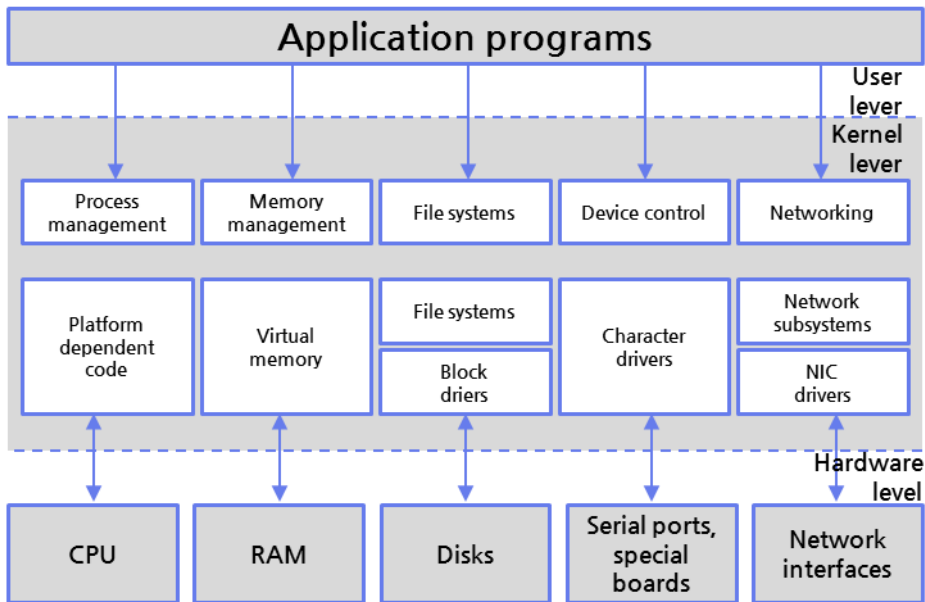
# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🌈 디바이스 드라이버의 분류

#### 솔라리스 커널의 구조



- ...> 그림은 리눅스의 한 종류인 솔라리스 커널의 구조임
- ...> 제어하는 디바이스의 종류에 따라 디바이스 드라이버를 분류할 수 있음

# 디바이스 드라이버 API의 소개



## ⚙ 디바이스 드라이버란 무엇인가?

### 🟡 디바이스 드라이버의 분류

#### 솔라리스 커널의 디바이스 드라이버

드라이버라는 이름의 모듈인 블록 드라이버, 캐릭터 드라이버, 네트워크 (NIC) 드라이버 등이 있음

##### 블록 드라이버

하드디스크, CD-ROM, USB 메모리와 같은  
♦ 저장소를 다루는 드라이버로 파일시스템과  
연관되어 동작함

##### 캐릭터 드라이버

시리얼 포트와 같은 전반적인 장치를 다루는  
♦ 드라이버

##### 네트워크 드라이버

이더넷 카드와 같은 네트워크 관련된 장치를  
♦ 다루는 드라이버

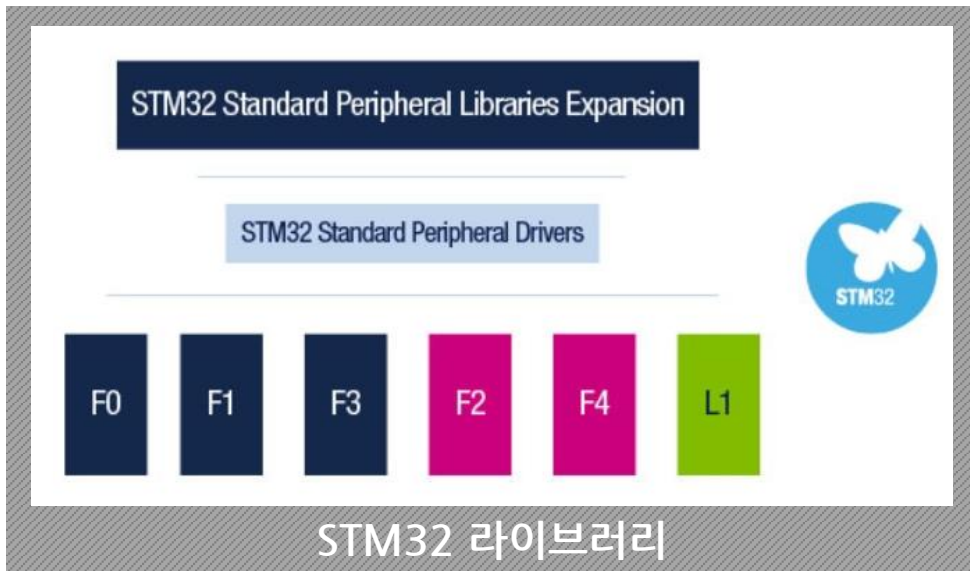


# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버란?



STM32 라이브러리는 운영체제를 사용할 수도 있고 사용하지 않을 수도 있음



운영체제와 독립적으로  
하드웨어 장치들을 다루는  
디바이스 드라이버를 제공함

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버란?

HAL 드라이버라는 용어를 사용함

- ... HAL은 Hardware Abstraction Layer의 약자로 하드웨어 추상화 계층으로 해석됨

장치의 종류에 상관없이  
사용할 수 있는  
정규화된 인터페이스 제공

- ... 실제로 STM32 라이브러리는 HAL 드라이버를 제공함으로써 STM32 시리즈의 다른 CPU로 교체되더라도 HAL 드라이버 위쪽의 소프트웨어는 수정할 필요 없이 사용가능하도록 하고 있음
- ... 즉, STM32F429 CPU를 사용하던 시스템을 STM32F103 CPU로 교체하더라도 기존 코드는 재사용이 가능함

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버의 종류

STM32 라이브러리에서 제공하는 HAL 드라이버의 종류

HAL 드라이버 이름	약어	풀이
HAL System Driver		
HAL ADC Driver	ADC	Analog to Digital Convertor
HAL CAN Driver	CAN	Controller Area Network
HAL CEC Driver	CEC	Consumer Electronic Controller
HAL CORTEX Driver	CORTEX	ARM Cortex M
HAL CRC Driver	CRC	Cyclic Redundancy Check
HAL CRYPT Driver	CRYPT	Cryptography
HAL DAC Driver	DAC	Digital to Analog Convertor
HAL DCMI Driver	DCMI	Digital Camera Interface
HAL DFSDM Driver	DFSDM	Digital Filter for Sigma-Delta Modulators

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버의 종류

STM32 라이브러리에서 제공하는 HAL 드라이버의 종류

HAL 드라이버 이름	약어	풀이
HAL DMA Driver	DMA	Direct Memory Access
HAL DSI Driver	DSI	Display Serial Interface
HAL ETH Driver	ETH	Ethernet
HAL EXTI Driver	EXTI	Extended Interrupts and events controller
HAL FLASH Driver		
HAL FMPI2C Driver	FMPI2C	Fast Mode Plus I2C
HAL GPIO Driver	GPIO	General Purpose Input/Output
HAL HASH Driver		
HAL HCD Driver	HCD	USB Host Controller Driver
HAL I2C Driver	I2C	Inter-integrated Circuit

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버의 종류

STM32 라이브러리에서 제공하는 HAL 드라이버의 종류

HAL 드라이버 이름	약어	풀이
HAL I2S Driver	I2S	Inter-integrated Sound
HAL IRDA Driver	IRDA	InfraRed Data Association
HAL IWDG Driver	IWDG	Independent Watchdog
HAL LPTIM Driver	LPTIM	Low Power Timer
HAL LTDC Driver	LTDC	LCD-TFT Display Controller
HAL MMC Driver	MMC	Multi-Media Card
HAL MSP Driver	MSP	MCU Specific Package
HAL NAND Driver	NAND	NAND Flash memory
HAL NOR Driver	NOR	Nor Flash memory
HAL PCCARD Driver	PCCARD	PCCARD/compact flash memories

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버의 종류

#### STM32 라이브러리에서 제공하는 HAL 드라이버의 종류

HAL 드라이버 이름	약어	풀이
HAL PCD Driver	PCD	USB Peripheral Controller Driver
HAL PWR Driver	PWR	Power Controller
HAL QSPI Driver	QSPI	Quad SPI
HAL RCC Driver	RCC	Reset and Clock Controller
HAL RNG Driver	RNG	Random Number Generator
HAL RTC Driver	RTC	Real Timer Clock
HAL SAI Driver	SAI	Serial Audio Interface
HAL SD Driver	SD	Secure Digital
HAL SMARTCARD Driver	SMARTCARD	
HAL SMBUS Driver	SMBUS	System Management Bus

# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버의 종류

STM32 라이브러리에서 제공하는 HAL 드라이버의 종류

HAL 드라이버 이름	약어	풀이
HAL SPDIFRX Driver	SPDIFRX	SPDIF Receiver
HAL SPI Driver	SPI	Serial Peripheral Interface
HAL SRAM Driver	SRAM	Static RAM
HAL TIM Driver	TIM	Timer
HAL UART Driver	UART	Universal Asynchronous Receiver/Transmitter
HAL USART Driver	USART	Universal Synchronous Receiver/Transmitter
HAL WWDG Driver	WWDG	Window WatchDog

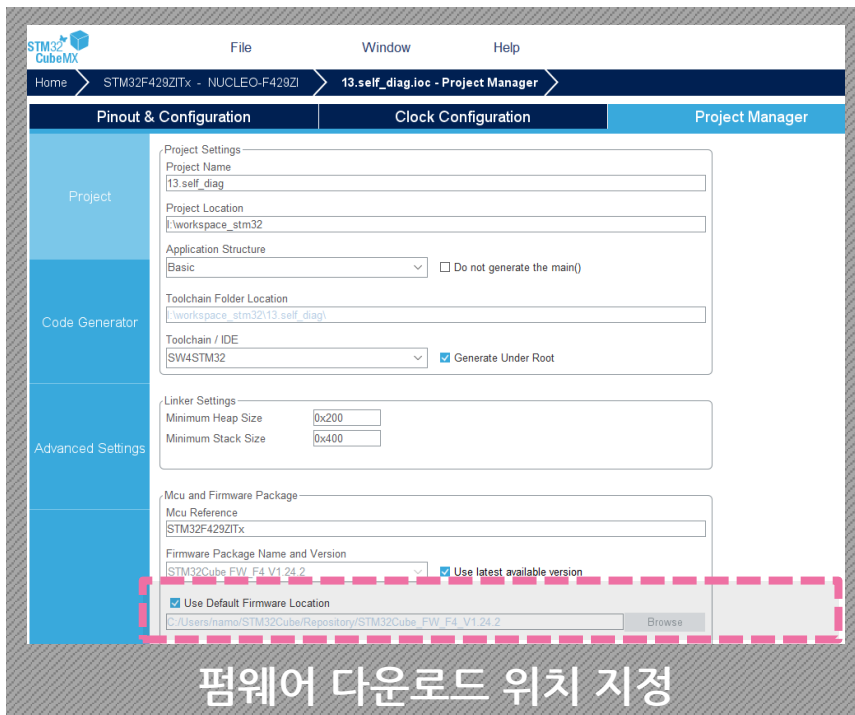
# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버 소스 코드

- ▶ 앞에서 설명한 수많은 종류의 HAL 드라이버의 소스 코드를 찾아볼 수 있음
- ▶ CubeMX에 지정한 펌웨어 다운로드 저장소에 STM32 라이브러리 전체가 있으며 그 중에 HAL 드라이버 소스 코드도 포함되어 있음



### 펌웨어 다운로드 저장소

- ▶ CubeMX의 Project Manager에서 지정함
- ▶ Mcu and Firmware Packages의 Firmware Location에 위치함



# 디바이스 드라이버 API의 소개



## ⚙ HAL 드라이버

### 🌈 HAL 드라이버 소스 코드

C:\₩사용자₩내이름₩STM32Cube  
₩Repository  
₩STM32Cube\_FW\_F4\_V1.24.2  
펌웨어 다운로드 저장소의 기본 위치

펌웨어 다운로드 저장소₩Drivers  
₩STM32F4xx\_HAL\_Driver

HAL 드라이버 소스 코드의 위치



CubeMX는 소스를 생성할 때  
이 장소에 있는 HAL 드라이버들을 복사함

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 HAL API의 이름 규칙

- ... 많은 HAL 드라이버 API들은 일정한 규칙을 가지고 이름 지어져 있음
- ... 이 규칙을 **Naming Rule**이라고 함
  - STM사에서 제공하는 문서인 “Description of STM32F4 HAL and LL drivers”에 설명되어 있음

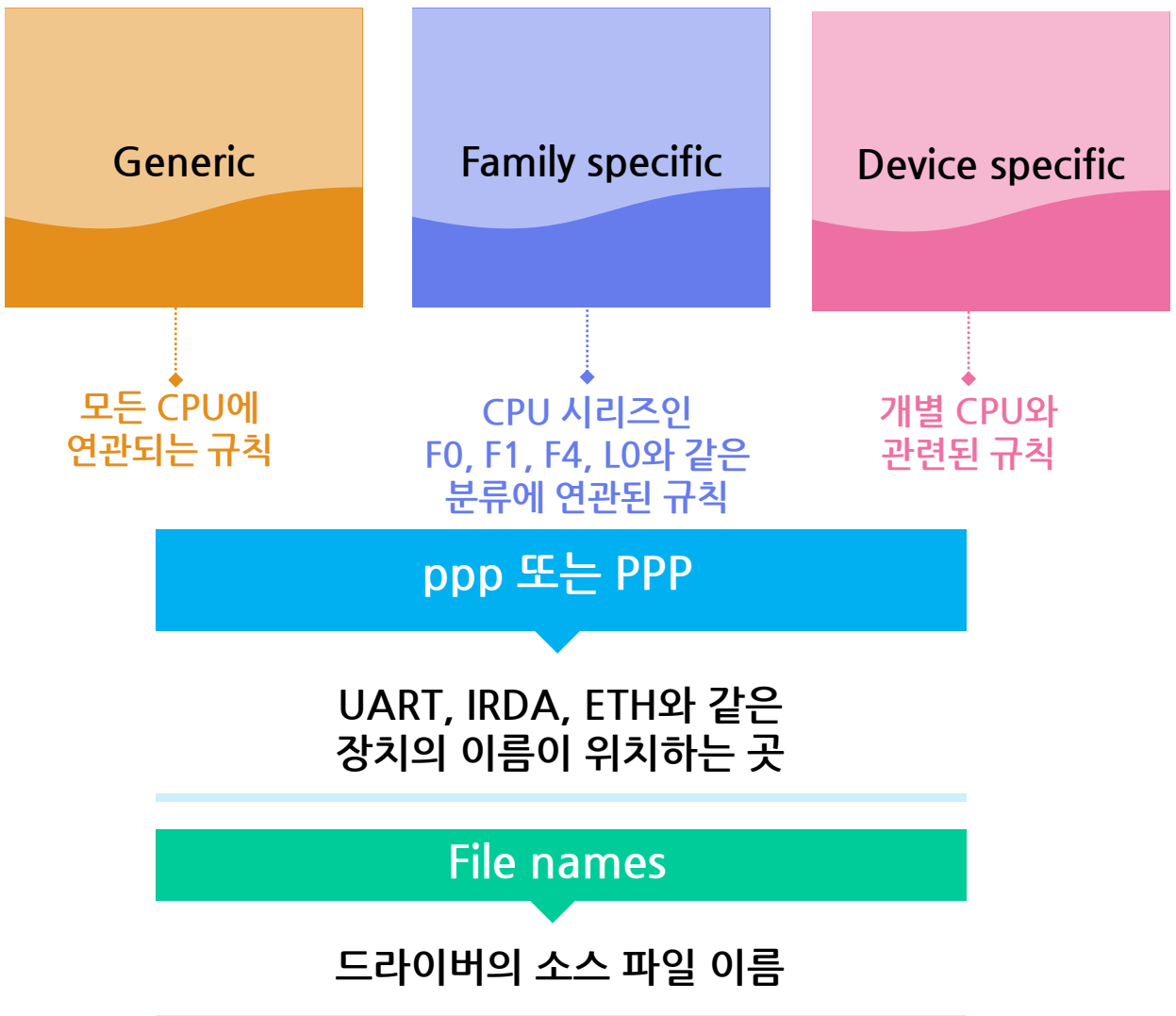
	Generic	Family specific	Device specific
File names	Stm32f4xx_hal_ppp(c/h)	Stm32f4xx_hal_ppp_ex(c/h)	Stm32f4xx_hal_ppp_ex(c/h)
Module name	HAL_PPP_MODULE		
Function name	HAL_PPP_Function HAL_PPP_FeatureFunction_MODE	HAL_PPPEX_Function HAL_PPPEX_FeatureFunction_MODE	HAL_PPPEX_Function HAL_PPPEX_FeatureFunction_MODE
Handle name	PPP_HandleTypeDef	NA	NA
Init structure name	PPP_InitTypeDef	NA	PPP_InitTypeDef
Enum name	HAL_PPP_StructnameTypeDef	NA	NA

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 HAL API의 이름 규칙



- ... Function name의 HAL\_로 시작하고 장치 이름인 PPP\_ 다음 함수 이름이나 함수 이름\_MODE가 추가됨
- ... Handle name의 Handle은 은 뒤에서 설명할 주변장치 핸들 구조체로 PPP\_로 시작하여 Handle 이름이 옴
- ... Typedef 지정자를 사용한 변수의 이름은 Typedef로 끝나는 규칙이 있음

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 HAL 드라이버의 데이터 구조

- ...> HAL 드라이버들은 공통적으로 다음과 같은 종류의 데이터 구조체를 가짐
  - 주변 장치 핸들 구조체(Peripheral handle structures)
  - 초기화 및 설정용 구조체(Initialization and configuration structures)
  - 특정 작업 구조체(Specific process structures)
- ...> 위 데이터 구조체들을 HAL 드라이버 API들을 사용하여 주변장치들을 제어함

#### 주변 장치 핸들 구조체

- ...> 주변장치의 설정이나 관련 레지스터를 제어하는 용도

#### 초기화 및 설정용 구조체

- ...> 주변장치의 초기화 및 동작 조건을 설정하기 위한 용도

#### 특정 작업 구조체

- ...> 특정한 작업 수행을 위해 API 함수 내에서 사용되는 구조체

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 주변 장치 핸들 구조체

- 주변장치의 설정, 레지스터와 관련된 사항을 다루기 위한 용도로 다음과 같은 형태의 이름을 가짐

```
PPP_HandleTypeDef *handle;
```

### → 주변장치 핸들 구조체의 예

```
typedef struct
{
    TIM_TypeDef          *Instance; /*!< Register base address */
    TIM_Base_InitTypeDef Init;      /*!< TIM Time Base required parameters */
    HAL_TIM_ActiveChannel Channel;  /*!< Active channel */
    DMA_HandleTypeDef    *hdma[7]; /*!< DMA Handlers array
                                     This array is accessed by a @ref DMA_Handle_index */
    HAL_LockTypeDef      Lock;      /*!< Locking object */
    __IO HAL_TIM_StateTypeDef State; /*!< TIM operation state */
} TIM_HandleTypeDef;
```

stm32f4xx\_hal\_tim.h에 정의된 타이머 장치 핸들 구조체

하지만 다음 장치들은 핸들 구조체가 없음



GPIO

SYSTICK

NVIC

PWR

RCC

FLASH

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 초기화 및 설정용 구조체

- 주변장치의 초기화 및 동작 조건을 설정하기 위한 용도로 다음과 같은 형태의 이름을 가짐

```
PPP_InitTypeDef *handle;
```

### → 초기화 및 설정용 구조체의 예

```
typedef struct
{
    uint32_t Pin;    /*!< Specifies the GPIO pins to be configured.
                     This parameter can be any value of @ref GPIO_pins_define */

    uint32_t Mode;   /*!< Specifies the operating mode for the selected pins.
                     This parameter can be a value of @ref GPIO_mode_define */

    uint32_t Pull;    /*!< Specifies the Pull-up or Pull-Down activation for the selected
                     pins.
                     This parameter can be a value of @ref GPIO_pull_define */
}GPIO_InitTypeDef;
```

stm32f4xx\_hal\_gpio.h에 정의된 GPIO 초기화 및 설정용 구조체

```
uint32_t Speed;    /*!< Specifies the speed for the selected pins.
                   This parameter can be a value of @ref GPIO_speed_define */

uint32_t Alternate; /*!< Peripheral to be connected to the selected pins.
                   This parameter can be a value of @ref
GPIO_Alternate_function_selection */
}GPIO_InitTypeDef;
```

stm32f4xx\_hal\_gpio.h에 정의된 GPIO 초기화 및 설정용 구조체

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🎯 특정 작업 구조체

- 특정한 작업을 수행하기 위해 API 함수 내에서 사용되는 구조체로 다음과 같은 형태의 이름을 가짐

```
HAL_PPP_Process  
(PPP_HandleTypeDef* hadc, PPP_ProcessConfig* sConfig)
```

# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🎯 HAL 인터럽트 핸들러

- HAL 드라이버는 API 함수 이외에도 인터럽트 핸들러로 가지며, 인터럽트 핸들러는 다음과 같은 형태를 가짐

HAL\_PPP\_IRQHandler



위 함수는 stm32f4xx\_it.c에서  
호출하는 구조로 되어 있음

### stm32f4xx\_it.c에서 호출하는 구조

CubeMX를 통해 NVIC (인터럽트)를 활성화하면  
자동으로 stm32f4xx\_it.c에 인터럽트 핸들러가 생성되는 구조

- 예를 들어 I2C1의 event interrupt를 활성화하면  
stm32f4xx\_it.c에 다음과 같은 인터럽트 핸들러가 생성됨

```
void I2C1_EV_IRQHandler(void)
{
    /* USER CODE BEGIN I2C1_EV_IRQn 0 */

    /* USER CODE END I2C1_EV_IRQn 0 */
    HAL_I2C_EV_IRQHandler(&hi2c1);
    /* USER CODE BEGIN I2C1_EV_IRQn 1 */

    /* USER CODE END I2C1_EV_IRQn 1 */
}
```



# HAL 드라이버 분석



## ⚙ HAL 드라이버 개요

### 🌈 HAL 콜백 함수

- HAL 드라이버는 인터럽트 핸들러와 비슷한 형태의 콜백(callback) 함수라는 것을 제공함
- 콜백 함수는 해당하는 인터럽트 핸들러를 추적하면 포함되어 있으나 이른바 weak attribute 속성을 가지는 빈 함수 형태로 되어있어 사용하려면 사용자가 직접 재정의해주어야 함
- 콜백 함수는 크게 3가지 종류로 나뉨

### 콜백 함수

Callback functions	Example
HAL_PPP_Msplnit() / _Deinit()	Ex: HAL_USART_Msplnit() Called from HAL_PPP_Init() API function to perform peripheral system level initialization(GIOs, clock, DMA, interrupt)
HAL_PPP_ProcessCpltCallback	Ex: HAL_USART_TxCpltCallback Called by peripheral or DMA interrupt handler when the process completes
HAL_PPP_ErrorCallback	Ex: HAL_USART_ErrorCallback Called by peripheral or DMA interrupt handler when an error occurs

#### HAL\_PPP\_Msplnit() / \_Deinit() 함수

초기화하거나 해제할 때 호출되는 콜백 함수

#### HAL\_PPP\_ProcessCpltCallback() 함수

인터럽트 핸들러가 수행된 다음에 호출되는 콜백 함수

#### HAL\_PPP\_ErrorCallback() 함수

해당 주변장치에 에러가 발생했을 때 호출되는 콜백 함수

# HAL 드라이버 분석



## ⚙ HAL API의 종류

### 🌈 HAL API의 2가지 종류

- ... HAL API는 크게 **일반(generic) HAL API**와 **확장(extension) HAL API**로 나뉨

#### 일반 HAL API

- ... 모든 STM32 CPU에 공통적으로 사용되는 일반적인 함수

#### 확장 HAL API

- ... 특정 모델의 CPU에서만 동작하는 함수 또는 특정한 기능의 함수
- ... 주로 stm32f4xx\_hal\_ppp\_ex.c와 같은 형태의 파일에 저장되어 있음

# HAL 드라이버 분석



## ⚙ HAL API의 종류

### 🌈 일반 HAL API

#### ... 초기화/해제 함수

- 주변장치를 초기화하고 클럭, GPIO, Alternate function, DMA, 인터럽트 등을 설정
- 형태 : HAL\_PPP\_Init(), HAL\_PPP\_Deinit()

#### ... 입출력 함수

- 주변장치의 입출력을 하는 함수
- 형태 : HAL\_PPP\_Read(), HAL\_PPP\_Write(), HAL\_PPP\_Transmit(), HAL\_PPP\_Receive()

#### ... 제어 함수

- 주변장치의 설정을 동적으로 변경하거나 다른 동작 모드로 전환하는 기능
- 형태 : HAL\_PPP\_Set(), HAL\_PPP\_Get()

#### ... 상태 및 에러 함수

- 실행 도중 주변장치의 상태와 데이터의 상태를 확인하거나 에러가 발생한 경우  
에러의 종류를 알아보는데 사용하는 함수
- 형태 : HAL\_PPP\_GetState(), HAL\_PPP\_GetError()

# HAL 드라이버 분석



## ⚙ HAL API의 종류

### 🌈 확장 HAL API

#### 확장 HAL API

특정 모델의 CPU에서만 동작하는 함수



이는 다시 특정 패밀리(혹은 시리즈)에서만 동작하는 API와  
특정 디바이스에서만 동작하는 API로 나눌 수 있음

# HAL 드라이버 분석



## ⚙ HAL API의 종류

### 🌈 확장 HAL API

#### ... 특정 확장 HAL API의 예

특정 패밀리의 ADC에 있는 기능을 구현한 API로 함수명에 Ex가 포함됨

### 확장 HAL API

Function Group	Common API Name
HAL_ADCEx_Injected Start()	This function starts injected channel ADC conversions when the polling method is used
HAL_ADCEx_Injected Stop()	This function stops injected channel ADC conversions when the polling method is used
HAL_ADCEx_Injected Start_IT()	This function starts injected channel ADC conversions when the interrupt method is used
HAL_ADCEx_Injected Stop_IT()	This function stops injected channel ADC conversions when the interrupt method is used
HAL_ADCEx_Injected Config Channel()	This function configures the selected ADC injected channel (corresponding rank in the sequencer and sample time)

# HAL 드라이버 분석



## ⚙ HAL API의 종류

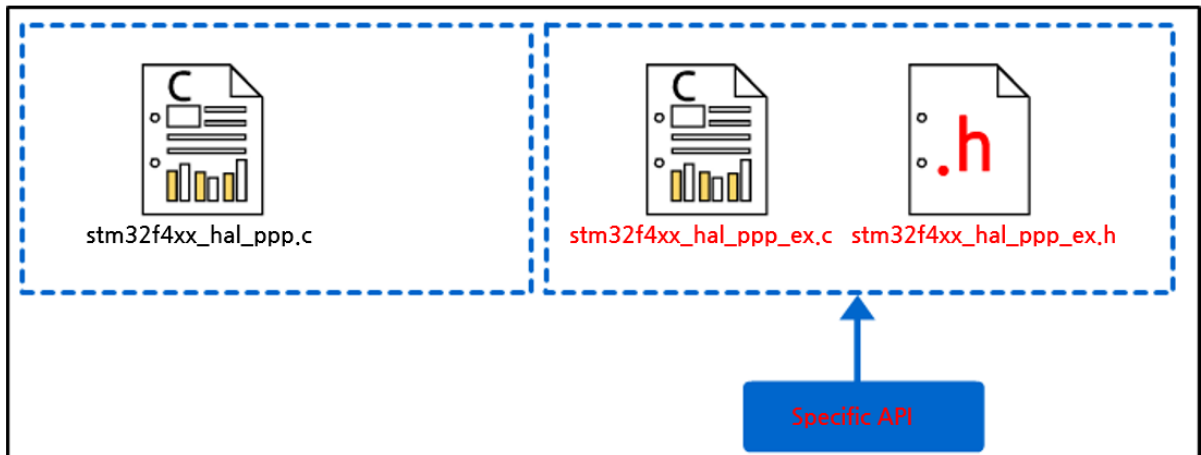
### 🌈 확장 HAL API

#### → 특정 디바이스 확장 HAL API의 예

특정 디바이스에서만 적용되기 때문에 조건부 컴파일 옵션이 포함되어 있음

### 특정 디바이스 확장 HAL API

Adding device-specific functions



Example: stm32f4xx\_hal\_flash\_ex.c/h

```
#if defined(STM32F427xx) || defined(STM32F437xx) || defined(STM32F429xx) ||  
Defined(STM32F429xx)  
HAL_StatusTypeDef HAL_FLASHEx_OB_SelectPCROP(void);  
HAL_StatusTypeDef HAL_FLASHEx_OB_DeSelectPCROP(void);  
#endif /* STM32F427xx || STM32F437xx || STM32F429xx || STM32F429xx */
```

## 요점노트

## 1. 디바이스 드라이버 API의 소개



- 디바이스 드라이버 API의 소개
  - 디바이스 드라이버는 하드웨어나 장치를 제어하기 위한 소프트웨어를 말함
  - STM32 라이브러리는 HAL 드라이버를 제공함으로써 STM32 시리즈의 모든 CPU의 소프트웨어는 호환이 가능함
  - CubeMX에 지정한 펌웨어 다운로드 저장소에 STM32 라이브러리 전체가 있으며 그 중에 HAL 드라이버 소스 코드도 포함되어 있음

## 요점노트

## 2. HAL 드라이버 분석



- HAL 드라이버 분석
  - 수많은 HAL 드라이버 API들은 일정한 규칙을 가지고 이름 지어져 있음
  - HAL 드라이버는 주변 장치 핸들 구조체, 초기화 및 설정용 구조체, 특정 작업 구조체로 구성되어 있음
  - HAL 드라이버는 API 함수 이외에도 인터럽트 핸들러와 콜백 함수로 이루어져 있음
  - HAL API는 크게 일반(generic) HAL API와 확장(extension) HAL API의 2가지로 나뉨