**Файлът с кода на програмата е main.c , който се намира \embedded\_systems\_exp\bootup\_GPIOD\_leds\Core\Src\main.c**

Цел да направим нещо подобно на библиотеките, която са ни дадени във HAL за да работим с LED 6,3,4 и 5 намиращи се на PD 12,13,14 и 15.

За целта ще поглеждаме и разясняваме кодът даден от HAL

**Стъпка I :**

Трябва да направим функция, която ще:

1. Енейбалне клока на порт D - \_\_HAL\_RCC\_GPIOD\_CLK\_ENABLE()

Картина, която съдържа текст, екранна снимка, Шрифт, линия

Описанието е генерирано автоматично

Тук използваме do{}while(0u), понеже искаме **кодът да бъде изпълнен веднъж** на извикване . Използва се тази структура главно за инициализиране

Регистърът RCC->AHB1ENR или (\*RCC).AHB1 (Advanced High-performance Bus 1 Peripheral Clock Enable Register) e част от struct RCC , който е отговорен за всички системни клокове и интеръпти .

Set\_bit се използва за да напишем 1 на бита от регистъра , който енейбалва клока на d порта (а именно 3тиа бит от регистъра след маска) .

**#define** SET\_BIT(REG, BIT) ((REG) |= (BIT)).

Този бит е дефинирам като GPIO D(port) EN (enable).

**#define** RCC\_AHB1ENR\_GPIODEN\_Pos (3U)

**#define** RCC\_AHB1ENR\_GPIODEN\_Msk (0x1UL << RCC\_AHB1ENR\_GPIODEN\_Pos) /\*!< 0x00000008 \*/

**#define** RCC\_AHB1ENR\_GPIODEN RCC\_AHB1ENR\_GPIODEN\_Msk

READ\_bit – тази команда прави AND само с бита който сме посочили да видим неговото състояние (а именно RCC\_AHB1ENR\_GPIODEN). Необходимо е да се направи за синхронизация и да сме сигурни че дори да е имало забавяне регистърът е променен.

Нека разясним регистъра AHB1ENR и какви са неговите настройки (за какво са отговорни отделните битове): Той е отговорен за енейбала на доста от периферните устройства, като например GPIO A,B,C,D и E , DMA 1 и 2 , CRC и различните видове RAM, които се намират на Advanced High-performance Bus 1.

Картина, която съдържа текст, линия, екранна снимка, диаграма

Описанието е генерирано автоматично

RCC base address = 0x4002 3800

AHB1ENR address = RCC base address + 0x30 = 0x4002 3830

GPIODEN 🡺 RCC->AHB1enr | = (1<<3)

**Стъпка II :**

Да направим в началото на функцията reset или 0 всеки един от пиновете, които използваме , това се прави за да предотврати грешки или нежелано поведение при включване т.е. :

HAL\_GPIO\_WritePin(GPIOD, LD4\_Pin|LD3\_Pin|LD5\_Pin|LD6\_Pin, *GPIO\_PIN\_RESET*);

За целта трябва да разгледаме функцията HAL\_GPIO\_WritePin :