1. **Для яких мікроконтролерів призначена бібліотека CMSIS?**

***CMSIS*** – бібліотека, стандартна для всіх МК з ядром ARM Cortex. Стандартизується ARM Ltd. Різні виробники МК з цим ядром доповнюють CMSIS файлами з описом периферійних модулів, специфічних для МК, які вони випускають.

1. **Для яких мікроконтролерів призначена бібліотека Standard Peripherals Library?**

Інакшим помічним засобом є бібліотека ***Standard Peripheral Library*** (SPL), яка є продуктом компанії STMicroelectronics і призначена для полегшення програмування мікроконтролерів виробництва цієї компанії.

1. **Який файл утворюється у результаті компіляції у Keil uVision?**

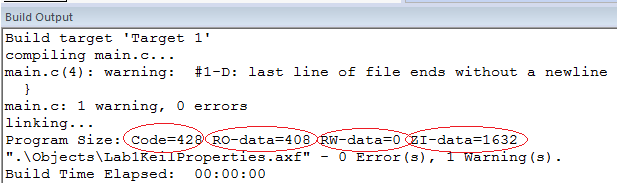


Рис. 7. Результати компіляції проекту

З рис. 7 бачимо, що будь-який файл повинен закінчуватися пустим рядком. Процес компіляції був успішним і у папці .\Objects\ знаходиться файл з кодом для мікроконтролера, Lab1KeilProperties.axf. Code означає розмір вихідного коду у байтах, RO-data – (від read only) – дані, доступні лише для читання, RW-data – дані для запису, ZI-data (Zero Initialized) – дані, ініціалізовані нулями.

1. **З яких файлів складається бібліотека CMSIS?**

Різні виробники МК з цим ядром доповнюють CMSIS файлами з описом периферійних модулів, специфічних для МК, які вони випускають.

1) спільну для усіх мікроконтролерів з ядром ARM Cortex M, файли:

core\_cmN.c

core\_cmN.h;

N позначає цифру, що характеризує покоління Cortex. Оскільки мікроконтролер STM32F407VG у складі STM32F4DISCOVERY має архітектуру Cortex M-4, то для даної оціночної плати потрібно застосувати файли core\_cm4.c + core\_cm4.h;

2) специфічну для виробника – файли:

system\_<конкретний\_МК>.с

system\_<конкретний\_МК>.h

<конкретний\_МК>.h, саме цей останній файл підключають у проект, він, у свою чергу, включає core\_cmN.h і system\_<конкретний\_МК>.h.

Для плати STM32F4DISCOVERY матимемо такий набір файлів:

system\_stm32f4xx.c

system\_stm32f4xx.h

stm32f4xx.h

1. **Яку структуру має бібліотека Standard Peripherals Library?**

Для конфігурації будь-якої периферії у SPL визначені структури вигляду <ім’я\_периферії>\_InitTypeDef (структура визначена у відповідному .h-файлі).

Для портів вводу/виводу є структура GPIO\_InitTypeDef, наведена на рис. 18:

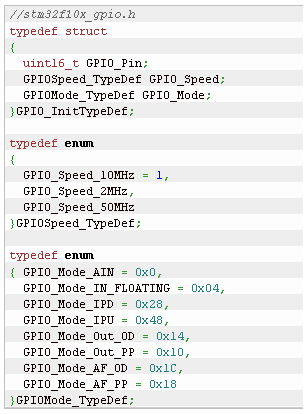
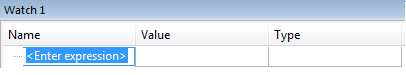


Рис. 18. Структура GPIO\_InitTypeDef і можливі значення її полів

1. **Як відстежити значення змінної у середовищі Keil?**

Нехай потрібно перевірити значення тих чи інших змінних (в нашому поточному проекті наразі вибір невеликий – змінна delay\_c). Для цього слід вибрати View = Watch Windows = >Watch 1 (або Watch 2) і у вікні, що при цьому появиться (рис. 13, а) ввести ім’я змінної – буде показано її поточне значення (рис. 13, б).



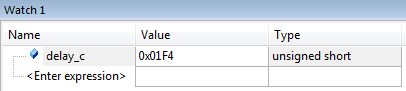


Рис. 13. Вікна для відстеження поточних значень змінних/виразів

1. Які інструменти для трасування коду є у середовищі Keil?
2. Назвіть типову структуру проекту у Keil uVision, відповідь об**ґ**рунтуйте.
3. **Як задати тактування периферійного модуля із застосуванням CMSIS?**

RCC->AHB1ENR |= RCC\_AHB1ENR\_GPIODEN;

1. **Як задати тактування периферійного модуля із застосуванням Standard Peripherals Library?**

RCC\_AHBPeriphClockCmd(RCC\_AHBPeriph\_PPPx, ENABLE); RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_PPPx, ENABLE); RCC\_APB1PeriphClockCmd(RCC\_APB1Periph\_PPPx, ENABLE);

1. Чому робота з периферійними модулями повинна починатися з увімкнення тактування?
2. **Як сконфігурувати порт у CMSIS?**

Оскільки світлодіоди керуються лише портом D, надалі замість абстрактної назви GPIOx застосовуємо GPIOD. GPIOD->MODER дозволяє сконфігурувати напрямок даних. Для роботи зі світлодіодом слід сконфігурувати порт *на вихід*. З документації мікроконтролера (пункт 8.4.1), фрагмент якої показано на рис. 16, видно, що для цього слід задати значення 01 (тут значенням керує комбінація з 2 бітів).

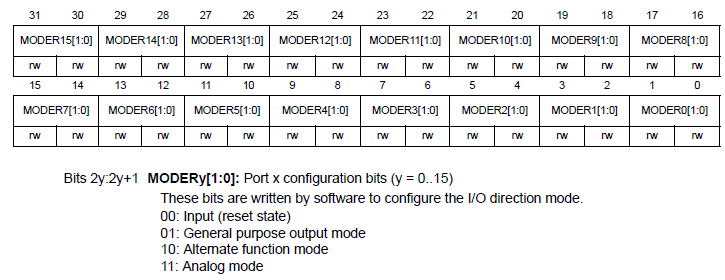


Рис. 16. Можливі значення регістра GPIOD->MODER

1. Як сконфігурувати порт у SPL?
2. **Які є порти вводу/виводу загального призначення на платі STM32F$DISCOVERY?**

Портів вводу/виводу загального призначення (GPIO – General Purpose Input Output) може бути різна кількість, у нашому випадку (для плати STM32F4DISCOVERY) є 5 портів GPIO: A, B, C, D і E. Порти, як бачимо, іменують великими латинськими літерами.

1. Якими регістрами конфігурується кожен порт?

Кожен порт є 16-бітним і зв'язаний з 8 32-бітними регістрами:

4 регістрами конфігурації GPIOx\_MODER, GPIOx\_TYPER, GPIOx\_ORD (тут *x* заміняє ім’я порту);

2 регістрами даних GPIOx\_ODR, GPIOx\_IDR;

2 регістрами вибору додаткових функцій GPIOx\_AFRH, GPIOx\_AFRL.

1. **Для чого призначений регістр GPIOx->MODER?**

Оскільки світлодіоди керуються лише портом D, надалі замість абстрактної назви GPIOx застосовуємо GPIOD. GPIOD->MODER дозволяє сконфігурувати напрямок даних. Для роботи зі світлодіодом слід сконфігурувати порт *на вихід*. З документації мікроконтролера (пункт 8.4.1), фрагмент якої показано на рис. 16, видно, що для цього слід задати значення 01 (тут значенням керує комбінація з 2 бітів)

1. **Для чого призначений регістр GPIOx->SPEEDR?**

Регістр GPIOD->SPEEDR відповідає за швидкість (є 4 рівні швидкості, відповідно, рівень швидкості кодується 2 бітами). Значення 00 відповідає найменшій швидкості, а оскільки швидкість не є принциповою для прикладу блимання світлодіодами, достатньо задати значення 0 для всіх виводів. Тому GPIOD->SPEEDR = 0.

1. **Яку структуру має регістр GPIOx->ODR?**

У 32-бітному регістрі GPIOD->ODR старші 16 бітів не використовуються (зарезервовані), а молодші 16 відповідають якраз виводам 16-розрядного порту (рис. 17).

Щоб засвітити світлодіод, треба подати 1 на відповідний пін. Відповідно якщо GPIOD->ODR = 0x9000, це відповідає числу 1001000000000000 у двійковій системі, тобто, згідно зі схемою на рис. 14, таким чином засвічуємо синій і зелений світлодіоди.

1. **Опишіть поля структури GPIO\_InitTypeDef.**

Для конфігурації будь-якої периферії у SPL визначені структури вигляду <ім’я\_периферії>\_InitTypeDef (структура визначена у відповідному .h-файлі).

Для портів вводу/виводу є структура GPIO\_InitTypeDef, наведена на рис. 18:

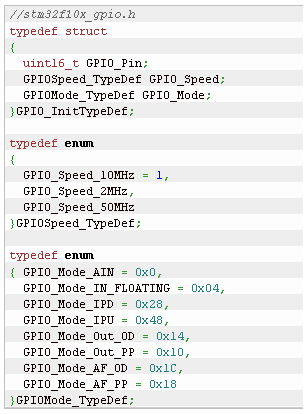


Рис. 18. Структура GPIO\_InitTypeDef і можливі значення її полів

1. Що означає перше поле у структурі GPIO\_InitTypeDef?