

1. Запознаване с STM32 Nucleo-64, FreeRTOS, програмиране и управление на светодиод чрез контролера

Целта на упражнението е да се запознаят студентите с развойната платка Nucleo-64, средата за програмиране STM32CubeMX, генериране и изпълнение на прост код.

1.1. Задачи за изпълнение

1. Да се конфигурира, генерира и пусне на устройството примерна програма за управление на светодиод чрез бутон използвайки STM32CubeMX.
2. Да се конфигурира и напише код за управление на светодиод чрез бутон без ползване на примерно генерираният код.
3. Да се конфигурира операционната система FreeRTOS и да се напише програма за управление на светодиод с ШИМ.
4. Да се напише програма за управление на светодиод през ШИМ, като коефициента на запълване да се управлява от потенциометър.
5. Да се напише програма за четене на стойността на ШИМ използвайки серийният интерфейс UART.

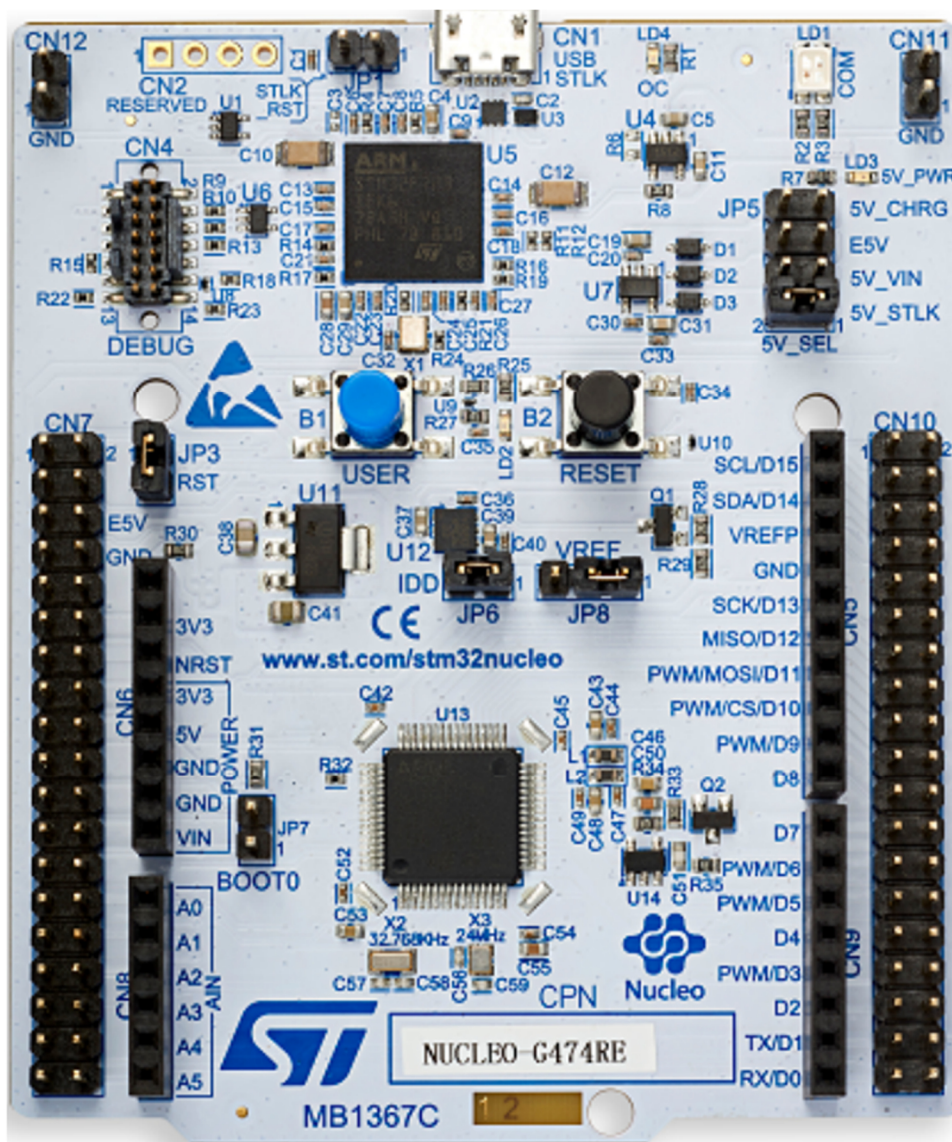
1.2. Запознаване с средата за програмиране и платка NUCLEO-G474RE на STM32

Платката STM32 Nucleo-64 предоставя на потребителите достъпен начин за създаване прототипи като избира от различните комбинации от характеристики за производителност и консумация на енергия, предоставени от микроконтролера STM32.

Платката STM32 Nucleo-64 се предлага с цялостните безплатни софтуерни библиотеки и примери на STM32, налични в пакета STM32Cube MCU Package.

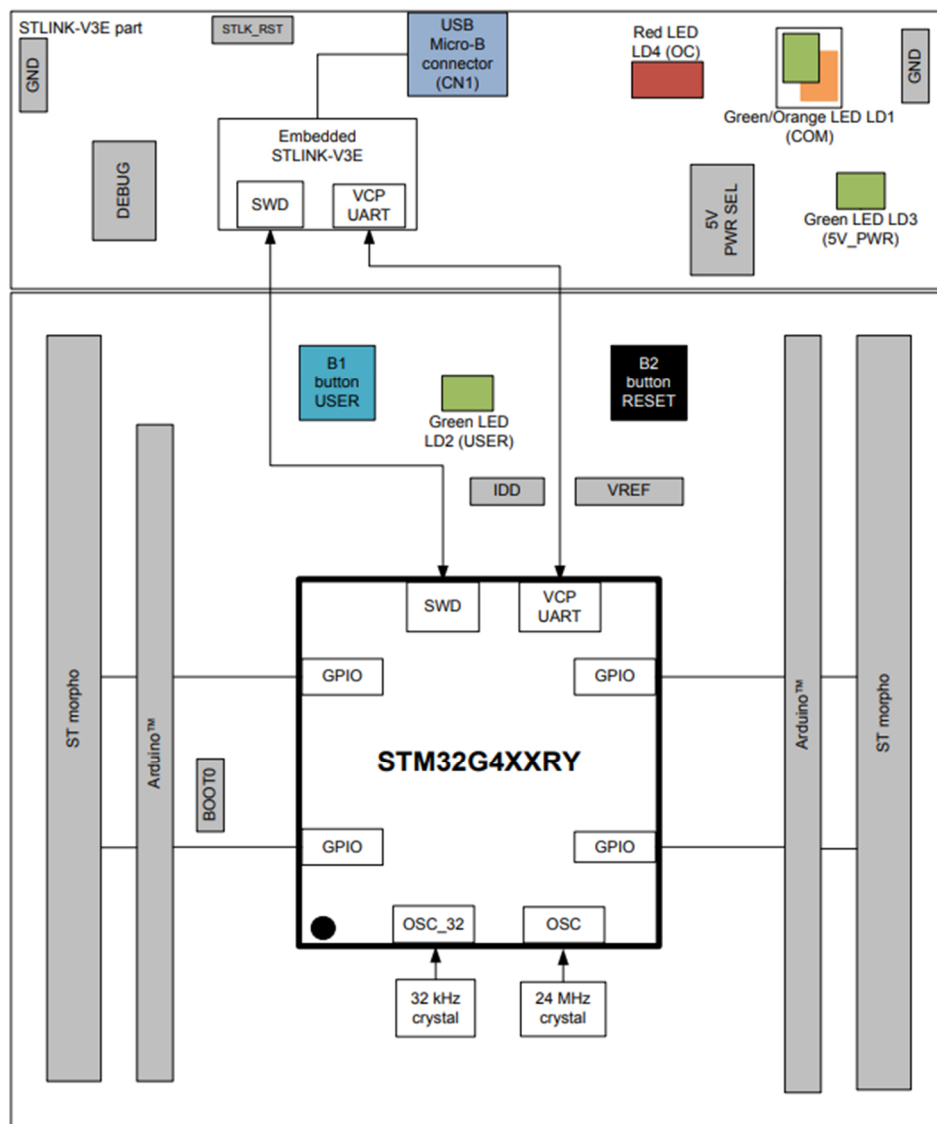
В днешното упражнение ще се запознаете по-подробно със средата за програмиране STM32CubeMX базирана на Eclipse IDE, писане и билдване на код, както и неговото дебъгване.

Като основни задачи ще се програмира код за включване и изключване на предоставеният на платката диод чрез бутон, контролиране на яркостта на диода със ШИМ и потенциометър и четене на текущите стойности за яркостта през ШИМ.

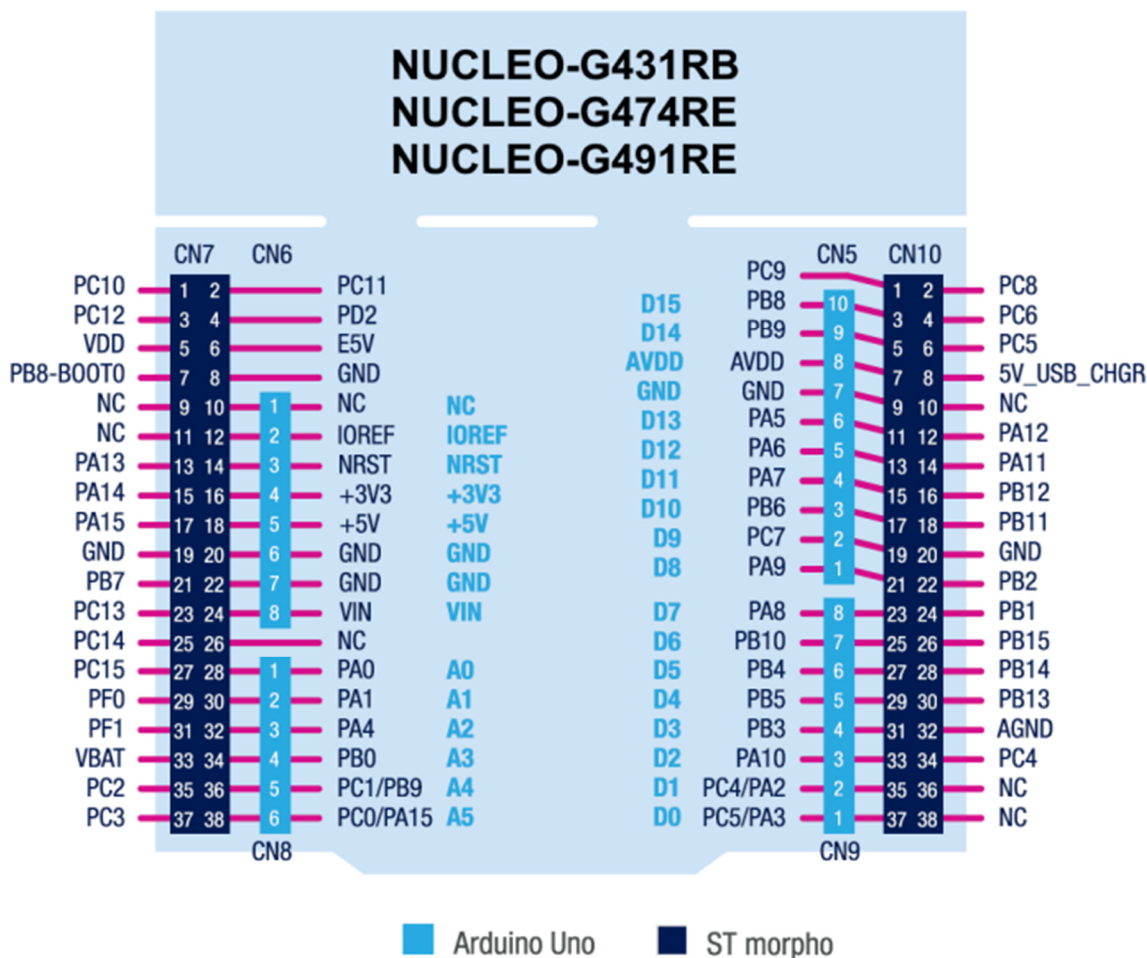


Фигура 1. Платка на микроконтролера STM32 Nucleo-64

Платката на STM32 (фиг. 1) наподобява познатите Arduino платки - има предоставена периферия за по-лесно програмиране през USB, изводи за IO-тата на устройството, два осцилатора с различна честота и др. (фиг 2)



Фигура 2 Блокова схема на микроконтролера STM32 Nucleo-64



Фигура 3. Схема на пиновете на Nucleo-64

1.3. Указания и информация за изпълнение на задачите

Презентацията със стъпки за изпълнение на задачите може да се достъпи в Git страницата на проекта.

1.3.1. **Задача 1** Да се конфигурира, генерира и пусне на устройството примерна програма за управление на светодиода чрез бутон използвайки STM32CubeMX.

Целта на тази задача е да се използва примерен код и конфигурация. За изпълнение се пуска програмата STM32CubeMX и се избира Access Board Selector → Nucleo-G474RE. След избор за стартиране на проекта, на прозореца за Board Project Options се избира отметката Generate demonstration code и се продължава с бутон OK.

Екрана с Pinout & Configuration има генерирани необходимите изходи.

Categories → System Core → GPIO показва подробен изглед на конфигурацията на изходите на устройството, а NVIC показва конфигурираните системни прекъсвания..

Прозорецът Clock configuration показва настройките на тактовата честота на микроконтролера.

Прозорецът Project Manager се използва за конфигуриране на информацията относно проекта. Тук задайте име на проекта, а от Toolchain/IDE се избира средата за програмиране STM32CubeMX.

Избира се Generate Code. Възможно е да се изиска допълнително изтегляне на firmware package, след което проекта се отваря с Open Project. Изберете или създайте папка за workspace-a.

Генерираният код се наблюдава в main.c. Когато състоянието на бутона е натиснато (`BspButtonState == 1`), светодиода сменя състоянието си. `BSP_LED_Toggle`.

Билдването на програмата става с десен клик на проекта → Build Project или от иконката с чукче на лентата с инструменти. Ако всичко е настроено правилно в конзолата ще се появи съобщение за правилно изпълнение.

Следваща стъпка е свързване с платката и изпълняване на кода. За целта от лентата с инструменти при изгледа на конзолата се избира Console (иконка с бяло прозорче и жълт плюс) където се избира Command Shell Console. Задава се име и COM настройки, като от Connection Name → New се избират допълнителни настройки за връзката. От Device Manager на компютъра може да се потвърди на кой COM е свързано устройството.

При свързана платка от лентата с инструменти се избира опцията за debug, без да се правят промени по конфигурацията на настройките, избира се OK и след това Switch.

От долната лента с инструменти се избира опцията Console to monitor и избираме DebugNucleo1.

За изпълнение на кода от лентата с инструменти се избира Resume бутона.

1.3.2. Задача 2 Да се конфигурира и напише код за управление на светодиод чрез бутон

Задачата за изпълнение е същата като предишната, с разликата че конфигурацията на устройството ще бъде ръчно направена а не генерирана от програмата.

За целта се създава нов проект и се избира опцията “Access MCU Selector”. Избира се Commercial Part Number → STM32G474RET6, и се потвърждава избора от MCUs/MPUs List. Когато е избран правилният микропроцесор се избира от горен десен ъгъл Start Project. Следващата стъпка е конфигуриране на изходите за управление. Вграденият бутон Button B1 е свързан с IO порт PC13, избира се като GPIO_EXTI13. Вграденият светодиод LED LD2 е свързан с PA5 като за конфигуриране се избира опцията GPIO_Output.

Чрез прозорецът Pinout & Configuration, System Core → GPIO може да потвърдим конфигурацията, която е направена до момента и да се променят различните им атрибути. За по лесно разпознаване на изходите може да се добавят имена, като например B1 за разпознаване на бутона на изход PC13, и LD2 за вградения светодиод на изход PA5.

От менюто NVIC се избира опцията за разрешаване на прекъсванията от входовете EXTI 15:10.

Менюто Clock Configuration се използва за задаване на тактовата честота на микроконтролера. За да зададем желаната честота от 170Mhz трябва да се конфигурира блок HCLK (Mhz) като стойността му се задава на 170. Програмата автоматично ще изчисли стойностите на останалите блокове.

От меню Project Manager се задава име на проекта Project name → “Lab1_LED_Button” и желаният от нас IDE, Toolchain / IDE → STM32CubeIDE. Когато са избрани тези опции може да генерираме шаблон в който да добавим нашият код за управление. За генериране на шаблона се избира опцията Generate Code.

В папката на проекта се генерират файлове и драйвъри необходими за работата на устройството. Файлът който се използва за програмиране е main.c.

Добавете код за изпълнение на програмата, след което се билдва и пуска по познатият вече начин. При правилно изпълнение на задачата, при натискане на бутона светодиода трябва да се включва и след това при повторно натискане - изключва.

1.3.3. Задача 3 Да се конфигурира операционната система FreeRTOS и да се напише програма за управление на светодиод с ШИМ.

В прозореца Pinout & Configuration трябва да се избере изхода PA5 да бъде от GPIO изход на PWM. За целта променяме видя му от GPIO_Output на TIM2_CH1. В менюто от Categories → Timers се задава конфигурацията на използвания таймер. В Mode се избира на Channel1 опцията PWM Generation CH1. В менюто Configuration → Parameter Settings се задават настройките на брояча → Prescaler (PSC -16 bit value) – 170-1; Counter Period (AutoReload Register) 10 000-1. За конфигуриране на FreeRTOS от меню Categories → Middleware and Software Packs се избира FREERTOS, след което трябва да се конфигурира. В меню “Config Parameters” → “Memory Management settings” се задава настройка TOTAL_HEAP_SIZE [32768]. В меню Advanced Setting → “Newlib Settings” се избира да е активиран параметъра USE_NEWLIB_REENTRANT.

В прозореца Pinout & Configuration трябва да се избере изхода PA5 да бъде от GPIO изход на PWM. За целта променяме видя му от GPIO_Output на TIM2_CH1. В менюто от Categories → Timers се задава конфигурацията на използвания таймер. В Mode се избира на Channel1 опцията PWM Generation CH1. В менюто Configuration → Parameter Settings се задават настройките на брояча → Prescaler (PSC -16 bit value) – 170-1; Counter Period (AutoReload Register) 10 000-1. За конфигуриране на FreeRTOS от меню Categories → Middleware and Software Packs се избира FREERTOS, след което трябва да се конфигурира. В меню “Config Parameters” → “Memory Management settings” се задава настройка TOTAL_HEAP_SIZE [32768]. В меню Advanced Setting → “Newlib Settings” се избира да е активиран параметъра USE_NEWLIB_REENTRANT.

За конфигуриране на задачите на системата се избира меню Tasks and Queues. За създаване на task се избира бутона Add. Създават се два taska “Task1” и ”Task2”. Ако има нужда от допълнителна настройки, с двоен клик се отваря меню за конфигуриране на задачите.

Последна стъпка преди генериране на кода е да се избере в меню Categories → System Core → SYS, където от Mode → Timebase Source се избира TIM6 (може да се избере всяко освен SysTick за код базиран на FreeRTOS).

Във файл main.c се добавя потребителски код. За инициализиране на PWM на канал едно трябва да се извика функцията:

```
HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1)
```

За управление на ШИМ ще се използват двата създадени taska. В StartTask1 да се запише кода за увеличаване на яркостта посредством ШИМ. Задавайки стойност на параметъра CCR1 се управлява ШИМ:

TIM2->CCR1 = <стойност>. Стойност 0 отговаря на 0% коефициент на запълване, а 10000 = 100%.

Да се напише програма която при натискане на бутона B1 да увеличава постепенно яркостта на диода LED LD2 от изключено до включено, посредством ШИМ за период от 1 секунда.

За целта да се използват и двата създадени таска, като единия да управлява включването, а другият изключването.

При правилно изпълнение на задачата, светодиода трябва плавно да се включва при натискане на бутона и след натискане, плавно да се изключва.

1.3.4. Задача 4 Конфигуриране на аналогов вход и управление на яркостта на светодиод с потенциометър.

В това упражнение ще се използват аналоговите I/O-та на устройството. През прозореца за конфигурация на пиновете се избират за конфигурация аналоговите пинове PC3 – ADC2_IN9 и PC0 – ADC2_IN2. По този начин те се конфигурират като входове и чрез 12 битовото АЦП на микроконтролера ще се измерва напрежението. От меню Categories → Analog → ADC 2 → Mode се настройва входа IN6 и IN9 да е Single-ended. От Configuration → Parameter Settings:

ADC Settings → Clock Prescaler се избира да е Asynchronous clock mode divided by 16.

ADC_Regular_ConversionMode → Number of Conversion се слага на 2

След генериране на проекта, трябва да се добави код за изпълнение на задачата. Примерен такъв може да намерите в презентацията към упражнението.

Билдва се и се зарежда кода на платката. В прозореца Live Expression се добавят в adc_raw и adc_mV за четене на стойностите на променливите, когато ги променяме.

При свързване на пин PC3 към захранващо напрежение от 3.3V се наблюдава промяна на стойността adc_raw – 4095 и adc_mV – 3.3V, а при свързване на пин PC0 към GND се наблюдава adc_raw и adc_mV – 0.

Да се свърже потенциометър към платката използвайки захранващо напрежение от 3.3V и да се модифицира кода по такъв начин, че да се чете стойността на напрежението на потенциометъра през пин PC0. Тази стойност да се използва за управление на светодиода посредством ШИМ, като максималната стойност на adc_raw – 4095 да отговаря на 100% а минималната на 0%.

1.3.5. Задача 5 Четене на информация от платката през UART

За изпълнение на упражнението PA2 и PA3 трябва да се конфигурират като UART TX-RX. Да се използва FreeRTOS базирания код като предишните упражнения, както и настройката на тактовата честота.

За конфигурация на протокола, в конфигурационното меню Connectivity → USART2 се проверяват настройките на протокола. Mode → Asynchronous, Hardware control flow → Disabled. От NVIC Settings се включват прекъсванията към протокола.

След генериране на кода да се добави символен низ и две стандартни функции за комуникация през UART които да се викат през колбека за UART:

```
HAL_UART_Transmit(&huart2, (uint8_t *)Message, strlen(Message), 10)
HAL_UART_Receive_IT(&huart2, UART2, rxBuffer, sizeof(UART2_rxBuffer))
```

За четене през STM32CubeMx се свързва по същият начин както преди за дебъгване, но сега ще може да се получават и съобщения през UART.

Примерен код може да се намери в презентацията, след което се билдва се и се пуска на устройството.

Когато се пише на серийният монитор, на всеки 10 символа ще се вижда какво е написано.

При последващи пускания на кода няма нужда да се създава нова връзка с устройството, а се избира вече създадената.

Модифицирайте кода да показва стойността на ШИМ на светодиода при настройка с потенциометъра.