Міністерство освіти та науки України Чорноморський національний університет імені Петра Могили

Крайник Я. М., Ухань Є. О.

МІКРОКОНТРОЛЕРИ

Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Методичні рекомендації

Випуск 447



Миколаїв - 2024

УДК 621.38-022.53(076) К 77

Рекомендовано до друку засіданням вченої ради факультету комп'ютерних наук Чорноморського національного університету імені Петра Могили (протокол № 1 від 28 серпня 2024 року).

Рецензент:

Журавська І. М., д-р техн. наук, професор, завідувачка кафедри комп'ютерної інженерії ЧНУ імені Петра Могили.

К 77 Крайник Я. М. Мікроконтролери : метод. рек. до виконання лабораторних робіт з дисципліни «Мікроконтролери» : метод. рек. / Я. М. Крайник, Є. О. Ухань. – Миколаїв : Вид-во ЧНУ ім. Петра Могили, 2024. – 52 с. – (Методична серія ; вип. 447).

> Методичні рекомендації містять відомості щодо початку роботи з мікроконтролерами, зокрема, на відлагоджувальних платах STM32F4 та GlobalLogic Embedded StarterKit. Також наведено відомості щодо основних конструкцій мови та їх використання й особливості. Призначені для використання при вивченні дисципліни «Мікроконтролери» здобувачами технічних спеціальностей закладів вищої освіти.

> > УДК 621.38-022.53(076)

© Крайник Я. М., Ухань Є. О. 2024 © ЧНУ ім. Петра Могили, 2024

ISSN 1811-492X

3MICT

ВСТУП4
Лабораторна робота № 1 Генерація коду ініціалізації. Налаштування проєкту в STM32CubeMX/STM32CubeIDE5
Лабораторна робота № 2 Робота з GPIO та зовнішніми перериваннями15
Лабораторна робота № 3 Реалізація роботи з паралельною шиною
Лабораторна робота № 4 Робота з таймерами. Широтно- імпульсна модуляція (ШІМ)24
Лабораторна робота № 5 Робота з модулем Direct Memory Access (DMA)
Лабораторна робота № 6 Робота з інтерфейсом SPI
Лабораторна робота № 7 Робота мікроконтролерів у режимах низького енергоспоживання
Лабораторна робота № 8 Робота з датчиком за допомогою інтерфейсу I2C42
Лабораторна робота № 9 Операційні системи реального часу46
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

ВСТУП

Мікроконтролери набули широкого розповсюдження при розробці вбудованих систем та характеризуються широкими можливостями щодо інтеграції у системи різних типів. Складність таких систем варіюється від простих, де мікроконтролер виконує лише одну функцію, до систем та пристроїв високої складності, де інтерфейси та внутрішні ресурси мікроконтролерів використовуються максимально ефективно.

Зручний доступ до відповідних апаратних засобів значно спрощує процес тестування та відлагодження апаратного та програмного забезпечення. Для набуття практичних навичок роботи з мікроконтролерами необхідним є розуміння архітектури сучасних контролерів, організація доступу та використання найбільш поширених інтерфейсів пристрою, можливість підключення зовнішніх пристроїв до мікроконтролера тощо. Тому представлені методичні рекомендації орієнтовані саме на те, щоб забезпечити отримання відповідних навичок при виконанні лабораторних робіт з дисципліни.

Розроблені методичні рекомендації орієнтовані на широке використання наборів для початківців GlobalLogic Embedded Starter Kit, які розроблялися спеціалістами компанії GlobalLogic та орієнтовані на те, щоб сприяти швидкому старту у галузі мікроконтролерів для студентів, оскільки надають можливість простого використання найбільш поширених інтерфейсів, а завдяки достатньо потужній цільовій платформі (STM32 серії F4) здатні забезпечити реалізацію потужних апаратнопрограмних застосунків. У ході виконання робіт також надається доступ до шаблону початкового проєкту з рекомендованими налаштуваннями, який можна використовувати для початку роботи. Відповідні набори отримані факультетом та кафедрою у рамках співпраці з миколаївським представництвом компанії, за що автор, кафедра та колектив факультету висловлюють щиру подяку.

Лабораторна робота № 1 Генерація коду ініціалізації. Налаштування проєкту в STM32CubeMX/STM32CubeIDE

Мета роботи: ознайомитися зі STM32IDE та її базою мікроконтролерів, набути навичок використання графічного інтерфейсу програми.

У випадку використання великої кількості модулів у проєкті основними можуть бути функції, пов'язані з ініціалізацією цих модулів та налаштуванням перед початком роботи. Часто такий код включає багато однотипних блоків, що повторюються, але при цьому містять велику кількість параметрів. Тому деякі виробники пропонують спеціальні засоби для генерації початкового проєкту. У випадку STM32 i ST Microelectronics таким засобом є графічний конфігуратор STM32CubeMX. У поточних версіях середовища для програмування STM32CubeIDE такий конфігуратор вже інтегрований у середовище, тому рекомендується використовувати саме його.

Графічний конфігуратор STM32CubeMX дозволяє:

 згенерувати проєкт, що містить код ініціалізації мікроконтролера;

 обирати і активовувати потрібні периферійні модулі та в інтерактивному режимі визначати їх доступність;

 задавати налаштування кожного конкретного модуля, які надалі будуть з'являтися у проєкті;

 проводити редагування згенерованого проєкту відповідно до певних правил.

Робота з візуальним конфігуратором STM32CubeMX/CubeIDE

Для початку роботи слід створити новий проєкт командою File/New Project. Кожен проєкт створюється для конкретного мікроконтролера або для конкретної відлагоджувальної плати, які можна обрати у діалоговому вікні (рис. 1.1a, б).

New Project										
U Selector Board	Selector									
CU Filters					Deckson (
ies :		~	All		 All 		~	More I	Filters 👻	
rinheral Selection		MCU	List: 866 Tems							
Parinhorals Nh	May	MCU	^	Lines	Package	Elach	Ram	Eenrom	10	
ADC 12-bit 0	40	A STM3	2F030C6Tx	STM32F0x0 Value Line	LOFP48	32	4	0	39	
ADC 16-bit 0	21	STM3	2F030C8Tx	STM32F0x0 Value Line	LQFP48	64	8	0	39	
CAN 0 COMP 0	2	STM3	2F030CCTx	STM32F0x0 Value Line	LQFP48	256	32	0	37	
DAC 12-bit 0	3	STM3	2F030F4Px 2E030K6Tx	STM32F0x0 Value Line STM32F0x0 Value Line	TSSOP20	16	4	0	15	
DCMI		STM3	2F030R8Tx	STM32F0x0 Value Line	LQFP64	64	8	0	55	
DSIHOST		STM3	2F030RCTx	STM32F0x0 Value Line	LQFP64	256	32	0	51	
Ethernet		STM3	2F031C4Tx	STM32F0x1	LQFP48	16	4	0	39	
FMPI2C		STM3	2F031C61x 2F031E6Yx	STM32F0X1 STM32F0x1	WLCSP25	32	4	0	20	
FSMC		STM3	2F031F4Px	STM32F0x1	TSSOP20	16	4	0	15	
HRTIM		STM3	2F031F6Px	STM32F0x1	TSSOP20	32	4	0	15	
12C 0	4	STM3	2F031G4Ux	STM32F0x1	UFQFPN28	16	4	0	23	
I2S 0	5	STM3	2F031660X	STM32F0x1	UFQFPN32	16	4	0	27	
LPTIM 0	2	STM3	2F031K6Tx	STM32F0x1	LQFP32	32	4	0	25	
LPUART		STM3	2F031K6Ux	STM32F0x1	UFQFPN32	32	4	0	27	
QUADSPI		STM3	2F038C61X	STM32F0x8	LQFP48 WLCSP25	32	4	0	38	
RTC		STM3	2F038F6Px	STM32F0x8	TSSOP20	32	4	0	14	
SDIO 0	2	STM3	2F038G6Ux	STM32F0x8	UFQFPN28	32	4	0	22	
SDMMC		STM3	2F038K6Ux	STM32F0x8	UFQFPN32	32	4	0	26	
SPDIFRX 0	6	STM3	2F042C41X 2F042C6Tx	STM32F0x2	LOFP48	32	6	0	38	
SWPMI		STM3	2F042C6Ux	STM32F0x2	UFQFPN48	32	6	0	38	
Segment LCD		STM3	2F042F4Px	STM32F0x2	TSSOP20	16	6	0	16	
Timer 16-bit 0	14	STM3	2F042F6Px	STM32F0x2	TSSOP20	32	6	0	16	
Timer 32-bit 0	2	STM3	2F042G6Ux	STM32F0x2	UFQFPN28	32	6	0	23	
UART 0	4	STM3	2F042K4Ux	STM32F0x2	UFQFPN32	16	6	0	28	
USART 0	8	STM3	2F042K6Tx	STM32F0x2	LQFP32	32	6	0	26	
USB Device		STM3	2F042K6UX 2F042T6Yx	STM32F0x2 STM32F0x2	WLCSP36	32	6	0	28	
USB OTG_HS		Y STM3	2F048C6Ux	STM32E0x8	UFOFPN48	32	6	0	37	
New Project										
U Selector Board	Selector									
CU Selector Board st bard Filter /endor :	Selector	Type of B	pard :	MCU Series :						
CU Selector Board S bard Filter /endor : STMicroelectronics	Selector	Type of B Discovery	pard :	MCU Series : All V						
CU Selector Board S and Filter /endor : STMicroelectronics	Selector	Type of B Discovery	pard : · · · · · · · · · · · ·	MCU Series : All V						>>
CU Selector Board S and Filter /endor : STMicroelectronics] Initialize all IP with eripheral Selection	Selector	Type of B Discovery efault Mode	Boards List: 18 It	MCU Series : All V						>>
U Selector Board S and Filter /endor : STMicroelectronics] Initialize all IP with eripheral Selection Peripherals/Conne	Selector	Type of B Discovery efault Mode Nb Max	Board : Boards List: 18 IX	MCU Series : All v ems Refer	nce		м	cu		>:
CU Selector Board st board Filter /endor : STMicroelectronics] Initialize all IP with eripheral Selection Peripherals/Conne Accelerometer	Selector	Type of B Discovery efault Mode	Boards List: 18 It Type Discovery	MCU Series : All v ems Refer ST M33 ST M33	ince IFODISCOVERY		M	CU M32F051R8Tx M32F02008Tv		×:
CU Selector Board S board Filter /endor : STMicroelectronics Initialize all IP with eripheral Selection Peripherals/Conne Accelerometer Analog VO	Selector	Type of B Discovery efault Mode	Boards List: 18 It Type Discovery Discovery	MCU Series : All v ems Refer STM32 STM32 STM33 STM33	ince F0DISCOVERY F0308-DISCO		M ST ST	CU M32F051R8Tx M32F030R8Tx M32F072R8Tx		>:
CU Selector Board Sard Filter Zu Sard Filter Zendor : STMicroelectronics Initialize all IP with eripheral Selection Peripherals/Conne Accelerometer Acadeo Line In Audio Line Out	Selector	Type of B Discovery efault Mode	Boards List: 18 It Type Discovery Discovery Discovery	MCU Series : All v STN32 STN32 STN32 STN32 STN32 STN32	nce F0DISCOVERY F0308-DISCO F0728-DISCO F0728-DISCOVERY		M ST ST ST ST	CU M32F051R8Tx M32F030R8Tx M32F072R8Tx M32F072R8Tx M32F072VCTx		>
CU Selector Board S CU Selector Board S Vendor : STMicroelectronics Initialize all IP with eripherals/Conne Acceleration Acceleration Audio Line Out Button CAN	Selector	Type of B Discovery efault Mode Nb Max 0 0 N/A N/A 0 1 0 1	Boards List: 18 IB Type Discovery Discovery Discovery Discovery Discovery	MCU Series : All V ems Refer STM32 STM32 STM33 STM33 STM33 STM33 STM33 STM33	nce FØDISCOVERY FØ308-DISCO FØ284-DISCO FØ284-DISCO FØ328-DISCOVERY F3348DISCOVERY F3348DISCOVERY F3348DISCOVERY	Ŷ	M ST ST ST ST ST ST	CU M32F051R8Tx M32F030R8Tx M32F0302R8Tx M32F0302CTx M32F334C8Tx M32F334C8Tx		>:
Vew Project TU Selector Board 1 ard Filter Vendor : STMicroelectronics STMicroelectronics Initialize all IP with eripheral Selection Peripherals/Conne Accelerometer Anado Line Out Button CAH Camero	Selector	Type of B Discovery efault Mode Nb Max I N/A 0 0 N/A 0 N/A N/A 0 1 0 0 I N/A	Boards List: 18 II Type Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All	Ince FOJISCOVERY F0308-DISCO F3DISCOVERY F3DISCOVERY F401C-DISCO	Y	M ST ST ST ST ST ST ST	CU M32F051R8Tx M32F030R8Tx M32F030R8Tx M32F030VCTx M32F031VCTx M32F01VCTx M32F01VCTx		>:
Vew Project 20 Selector Board 1 20 Selector Board 1 20 STMIcroelectronics 21 Initialize all IP with 21 StMicroelectronics 21 Initialize 21 Initialize all IP with 21 StMicroelectronics 21 Initialize 21 Initial	Selector	Type of B Discovery efault Mode Nb Max 0 0 1 N/A 0 1 0 0 1 N/A 0 1 0 0 0 0 0 0	Boards List: 18 IX Boards List: 18 IX Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All ~ ems Refer STM32 S	Proce F0DISCOVERY F0308-DISCO F028-DISCO F028-DISCO F028-DISCO F4015-OISCO F	Ŷ	M ST ST ST ST ST ST ST ST ST	CU M32P051887x M32P030887x M32P0302887x M32P303VCTx M32P401VCTx M32P401VCTx M32P401VCTx M32P401VCTx		×
Vew Project 20 Selector Board 2 20 Selector Board 2 20 STMIcroelectronics 21 Initialize all IP with 21 Initialize all IP w	Selector	Type of B Discovery efault Mode Nb Max N/A N/A 0 0 1 N/A 0 1 1 N/A 0 1 N/A 1 0 1 1 N/A 0 128 N/A 128	Boards List: 18 II Boards List: 18 II Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All	Ince FPDISCOVERY F0308-DISCO F0728-DISCO F0728-DISCO F411E-DISCO F411E-DISCO F411E-DISCO F415C-OVERY F407C-DISC1	Ŷ	M ST ST ST ST ST ST ST ST ST ST	CU M32F051R8Tx M32F030K8Tx M32F030VCTx M32F401VCTx M32F401VCTx M32F401VCTx M32F401VCTx M32F407VCTx M32F407VCTx M32F407VCTx		×
Vew Project US selector Board 1 ard Filter Vendor : Thitroelectronics Thitroelectronics Thitroelectronics Peripherals/Conne Accelerometer Accelerometer Accelerometer Accelerometer Accelerometer Compass Digital I/O Eeprom Chart	Selector	Type of B Discovery efault Mode Nb Max N/A 0	Boards List: 18 II Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All v ems Refer STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32 STM32	Ince F00ISCOVERY F0308-DISCO F3DISCOVERY F401C-DISCO F40ISCOVERY F401C-DISCO F401SCOVERY F407C-DISCO F403C-DISCO F403C-DISCO	Ŷ	M ST ST ST ST ST ST ST ST ST ST ST	CU M32F031R8Tx M32F030R8Tx M32F030R8Tx M32F303VCTx M32F430VCTx M32F410VCTx M32F410VCTx M32F407VCTx M32F402VCTx M32F4292TTx		×
Vew Project Diselectory Board 1 bard Filter Initialize all IP with arrpheral Selection Peripherals/Come Accelerometer Andeo Line Out Addo Line Out Compass Digital I/O Exprom Digital I/O Exprom	Selector h their d actors	Type of B Discovery efault Mode Nb N/A 0 0 1 N/A 0 1 0 0 1 N/A 0 1 N/A 0 0 1 N/A 0 N/A 0 1 N/A 0 12	Boards List: 18 IB Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All	Ince FP015COVERY F0308-DISCO F0728-DISCO F0728-DISCO F4012-DISCO F4012-DISCO F4015-COVERY F407C-DISC1 F4015-COVERY F407C-DISC1 F4091-DISCO	Ŷ	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F051887x M32F030887x M32F030887x M32F030467x M32F030467x M32F030467x M32F030467x M32F03047x M32F0307x M37707x M37707x M37707x		>:
Uselector inject 20 Selector 20 Selector 27 Microelectronics 27 Microelectronics 27 Microelectronics 27 Microelectronics 20 Microelector 20 Mi	Selector h their d actors	Type of B Discovery efault Mode Nb Max N/A 0 0 1 N/A 0 0 1 N/A 0	Boards List: 18 B Type Discovery	MCU Series : All ems Refer STM33 STM33 STM33 STM33 STM33 STM33 ST	Ince F0DISCOVERY F0DISCOVERY F0DISCOVERY F0DISCOVERY F401C-DISCO F4DISCOVERY F401C-DISCO F4DISCOVERY F4207C-DISCO F420F0DISCO F420F0DISCO F440F0DISCO F440F0DISCO	Y	M 57 57 57 57 57 57 57 57 57 57 57 57 57	CU M32F030R8Tx M32F030R8Tx M32F030R8Tx M32F030R8Tx M32F134CTFx M32F4101VCTx M32F401VCTx M32F401VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F403VCTx M3		>
Vew Project Jack Filter Fendor I: Thitolize all IP with arripheral Selection Peripherals/Commeter Analog I/O an Accelerometer Analog I/O an Audio Line Out Button Competer Audio I/O Competer Co	Selector h their d actors	Type of B Discovery efault Mode Nb Max N/A N/A 0 1 N/A 0 N/A N/A 0 1 N/A 0 N/A N/A 0 1 N/A N/A N/A N/A N/A N/A N/A N/A N/A N/A	Boards List: 18 It Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All v sems STN432 STN532 STN432	Ince FOISCOVERY FOISCOVERY FOIS-DISCO FO728-01SCO FO728-01SCO FOISCOVERY FOISCOVER FOISCOVE	Y	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F051R8Tx M32F030R8Tx M32F030VCTx M32F30VCTx M32F411VeTx M32F407VGTx M32F407VGTx M32F407VGTx M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32F4080H0k M32H080CTx		>
Verwindert U Selector Jarde Filter STMicroelectronics Thicroelectronics Thicroelectronics Thicroelectronics Thicroelectronics Thicroelectronics Paripherals/conne Acade Line Bu Acade Line Bu	Selector	Type of B Discovery efault Mode N/A	Board : Type Discovery Discove	MCU Series : All ems ems Fritos STM32	1006 1009	Ŷ	M 51 51 51 51 51 51 51 51 51 51 51 51 51	CU M32P03188Tx M32P03088Tx M32P03088Tx M32P030VCTx M32P030VCTx M32P030VCTx M32P030VCTx M32P030VCTx M32P030VCTx M32P030VCTx M32P030VCTx M32P030C		>
Veew Holect U Selector Beard and Filter endor : STMicroelectronics STMicroelectronics I Initialize all IP with tripheral Selection Peripheral/conne Accelerometer Accelerometer Accelerometer Manage Design Mutton Compass Digital I/O Exprom Compass Digital I/O Exprom Carabic Led Display Corrabic Led Display Corpolicy Led	Selector	Nb Max 0 0/A 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10	Boards 1: Type Discovery	MCU Series : All ems Refer STM32 STM32 STM33	Ince FOJSCOVERY FOJSOSCOVERY FOJSOSCO FOJZA-DISCO FOJZA-DISCO FOJZA-DISCOVER FOJZA-DISCOVER FOJZA-DISCOVER FOJZA-DISCO FOJZA-D	Y	M 57 57 57 57 57 57 57 57 57 57 57 57 57	CU M32F051R8Tx M32F030R8Tx M32F030K8Tx M32F030VCTx M32F401VCTx M32F401VCTx M32F401VCTx M32F407VCTx M32F407VCTx M32F408VL M32F4		>
Verwindert U Selector and Filter STMicroelectronics STMicroelectronics I Initialize all IP with tripheral Selection Peripheralk/come Accelerometer Analog I/O Analog I/O Analog I/O Analog I/O Analog I/O Degital Selection Compas Selection Compas Objetal Objetal Objetal Selection Charter Compas Objetal Objetal Selection Charter Compas Objetal Objetal Selection Charter Compas Objetal Objetal Selection Charter Compas Objetal Objetal Selection Charter Compas Objetal Objetal Selection Charter Compas Objetal Objetal Compas Objetal Co	Selector	Nb Max N/A N/A 0 1 0 1 0 1 0 1/A 0 1 N/A 0	Board : Type Discovery	MCU Series : All V ems ems STM33 STM	PR00 F00BSC0VERY F030-05C0 F702E-05C0 F702E-05C0 F702E-05C0 F702E-05C0 F702E-05C0 F702E-05C0 F702E-05C1 F402E-05C1 H02E-05C0 H02E-05C0 H02E-05C0 H02E-05C0 H02E-05C0 H02E-05C0 H02E-05C0	Y	M 51 51 53 53 53 53 53 53 53 53 53 53 53 53 53	CU M32P031R8TX M32P032R8TX M32P032R8TX M32P332VGTX M32P33VGTX M32P33VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX M32P43VGTX		
Veew Holect U Selector (Beard 1 and Filter endor : STMicroelectronics STMicroelectronics Initialize all IP with ripheral Selection Peripheral/come Accelerometer Accelerometer Accelerometer Manager Auto Line Out Button Compass Digital V/O Eeprom Carphic Led Display Carphic Led Display Correscope Joyatick Led Display Led Display L	Selector	Type of B Discovery efault Mode Nb Max 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 12 N/A 0 0 12 N/A N/A 0 12 N/A N/A 0 12 N/A N/A	Boards 1: Type Discovery	MCU Series : All ~ ems Refer 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32 5TM32	Ince F0JSC0VERY F0J8-0JSC0 F0J8-0JSC0 F0J8-0JSC0 F1J8-0JSC0 F4J1E-0JSC0 F4J1E-0JSC0 F4J2E-	Y	M 51 51 51 51 51 51 51 51 51 51 51 51 51	CU M32F051R8Tx M32F030R8Tx M32F030R8Tx M32F032VCTx M32F403VCTx M32F403VCTx M32F403VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32L152R6Tx M32L152R6Tx M32L152R6Tx		
Veew Holect U Selector Bord and Filter STMicroelectronics STMicroelectronics I Initialize all IP with tripheral Selection Peripherals/come Andio I/O D Andio I/O D Andio I/O D Andio I/O D Andio I/O D Audio Line Out Sutton CAR Compass Dagital V/O Exprove Compass Dagital V/O Exprove Compass Compa	Selector	Type of B Discovery Biscovery efault Mode Nb Max I II/A	Board : Type Discovery	MCU Series : All V ems ems STM32 STM	PR06 P005C0VERY P005C0	Y	M 57 57 57 57 57 57 57 57 57 57 57 57 57	CU M32F030875 M32F030875 M32F030875 M32F030875 M32F334CFTx M32F4304CFTx M32F7304CFT		
Veew Project US selector Board - Itard Filter erador : STMicroelectronics STMicroelectronics STMicroelectronics Initialize all IP with erapheral Selection Peripheral Selection Accelerometer Accelerometer Accelerometer Compass Digital VO Compass Digital VO Compass Compass Digital VO Compass Compass Compass Compass Digital VO Compass Compass Digital VO Compass Compa	Selector	Type of B Discovery efault Mode Nb Max N N/A 0 1 N/A 1 0 1 N/A 1 0 1 N/A 1 0 1 N/A 1 0 1 N/A N/A 0 0 0 0 N/A N/A N/A N/A N/A N/A N/A N/A 0 2 N/A N/A N/A N/A 0 2 N/A N/A 0 2 N/A N/A 0 2 N/A N/A	Board : Type Discovery	MCU Series : All Series : Bens Refer STM32	nn@ FR0JSCOVERY FR0JSCOVERY FR0J&DISCO FR0J&DISCOVERY FR0JEDISCOVERY FR0JEDISCOVERY FR0JEDISCO FR0J	Y	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F031R8Tx M32F030R8Tx M32F030R8Tx M32F032R8Tx M32F031CTx M32F431CTx M32F431CTx M32F439CTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx M32F439ZTx		
View Project 20 Selector (Baard Siler erador : STMicroelectronics STMicroelectronics I) nitolize all IP with arriheral Selection Peripherals/come Accelerometer Accelerometer Accelerometer Accelerometer Accelerometer Septon Compass Digital I/O Exprose Septon Compass Digital I/O Exprose Compass Digital I/O Exprose Compass Digital I/O Exprose Compass Digital I/O Exprose Compass Compass Digital I/O Exprose Compass	Selector	Type of B Discovery Discovery efault Mode N/A U N/A <td>Board : Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery</td> <td>MCU Series : All V ems Ems STM32 STM</td> <td>1966 Folia OSCO Forza DISCO Forza DISCO Forza DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Folia DISCO Falle DISCO Folia DISCO FO</td> <td>Y</td> <td>M ST ST ST ST ST ST ST ST ST ST ST ST ST</td> <td>CU M32F030875 M32F030875 M32F030875 M32F3304GTX M32F3304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX</td> <td></td> <td></td>	Board : Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All V ems Ems STM32 STM	1966 Folia OSCO Forza DISCO Forza DISCO Forza DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Falle DISCO Folia DISCO Falle DISCO Folia DISCO FO	Y	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F030875 M32F030875 M32F030875 M32F3304GTX M32F3304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX M32F4304CTX		
Vew Project US selector Board 1 and Filter Ardor : STMcroelectronics STMcroelectronics and the selection Peripherals/Conne Acceleromater Acceleromater Acceleromater Acceleromater Compass Batton Compass Compass Distance Compass Com	Selector	Type of B: Discovery Discovery efault Mode Nb Max N/A N/A 0 10 N/A 0 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 10 0 12 0 12 0 12 0 12 0 12 0 12 0 12 0 12 0 12 0 12 0 13	Board : Type Discovery	MCU Series : All	Ince FR0JSCOVERY FR0JSCOVERY F03JE018CO F03JE018COVERY F13JE018COVERY F40JE018CO F4	Ŷ	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32P031R8Tx M32P030R8Tx M32P030R8Tx M32P130VCTx M32P131VCTx M32P431VCTx M32P431VCTx M32P439VCTx M32P4392VTx M32		
Vew Hoject US selector Bard 1 Sord Filter Vendor : STMicroelectronics TMicroelectronics TMicroelectronics Paripheral Selection Accelerometer Accelerometer Accelerometer Compass Compa	Selector	Type of B Discovery Biscovery Max N/A N/A </td <td>Boards List: 18 II Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery</td> <td>MCU Series : All V STN32</td> <td>Profe FobScoVERY FobScoVERY For Solar Disco For Solar Disco For Solar DiscovErY Folic DiscovErY Folic DiscovErY FobScoVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY</td> <td>Y</td> <td>M ST ST ST ST ST ST ST ST ST ST ST ST ST</td> <td>CU M32F0318875, M32F0308875 M32F0308875 M32F303UCTX M32F303UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32L103CCTX M32L103CCTX</td> <td></td> <td></td>	Boards List: 18 II Type Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery Discovery	MCU Series : All V STN32	Profe FobScoVERY FobScoVERY For Solar Disco For Solar Disco For Solar DiscovErY Folic DiscovErY Folic DiscovErY FobScoVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY LoSCOVERY	Y	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F0318875, M32F0308875 M32F0308875 M32F303UCTX M32F303UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32F430UCTX M32L103CCTX M32L103CCTX		
Vew Hoject US selector Board T Vendor : STMicroelectronics STMicroelectronics and selecton Peripherals/Conne Peripherals	Selector	Type of B Discovery Discovery Fault Mode Nb Max Nb Nb Nb	Board : Type Discovery	MCU Series : All ems ems ems 57103 571	Ince ProjScience	Ŷ	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32P03188Tx M32P03088Tx M32P03008Tx M32P030VCTx M32P03VCTx M32P		
Vew Hoyert CU Selector Board 1 Sector Sector Board 1 STMcroelectronics STMcroelectronics STMcroelectronics Paripheral Selection Accelerometer Accelerometer Accelerometer Accelerometer Compass Audio Line Out Board 1/O Compass Compa	Selector	Type of B Biscovery Discovery Biscovery 0 0 C 0 0 D 0 D D <td>Boards II II</td> <td>MCU Series : All V ems Ems STM32 STM</td> <td>nne FobScovERV FobScov</td> <td>Y</td> <td>M ST ST ST ST ST ST ST ST ST ST ST ST ST</td> <td>CU M32F051R8Tx M32F030R8Tx M32F030R8Tx M32F039VCTx M32F403VCTx M32F403VCTx M32F407VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32L152R6Tx M32L152R6Tx M32L152R6Tx</td> <td></td> <td></td>	Boards II	MCU Series : All V ems Ems STM32 STM	nne FobScovERV FobScov	Y	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32F051R8Tx M32F030R8Tx M32F030R8Tx M32F039VCTx M32F403VCTx M32F403VCTx M32F407VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32F402VCTx M32L152R6Tx M32L152R6Tx M32L152R6Tx		
Uselector Board : Josef Carlos Carlo	Selector	Type of B Biscovery Discovery Biscovery Visit No No No <tr< td=""><td>Board : Type Discovery</td><td>MCU Series : All V ems ems 5TH33</td><td>NOS F030-05C0 F030-05C0 F032-0</td><td>Y</td><td>M 51 51 57 57 57 57 57 57 57 57 57 57 57 57 57</td><td>CU M32P03188Tx M32P03788Tx M32P03788Tx M32P03788Tx M32P13924CTx M32P13924CTx M32P13924CTx M32P13924CTx M32P14914CTx M32P14</td><td></td><td></td></tr<>	Board : Type Discovery	MCU Series : All V ems ems 5TH33	NOS F030-05C0 F030-05C0 F032-0	Y	M 51 51 57 57 57 57 57 57 57 57 57 57 57 57 57	CU M32P03188Tx M32P03788Tx M32P03788Tx M32P03788Tx M32P13924CTx M32P13924CTx M32P13924CTx M32P13924CTx M32P14914CTx M32P14		
Verwindert U Selector Board 1 ard Filter endor : STMcroelectronics STMcroelectronics Initialize all IP with rephrafal/Conne Accelerometer interiore and the selection Peripherals/Conne Accelerometer Compass Digital VO Compass Digital VO Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Compass Digital VO Digital VO Di Digital VO Digital VO Digital VO Di	Selector	Type of B Item of B <t< td=""><td>Board : Type Discovery</td><td>MCU Series : All</td><td>Ince Projos OSC Proze OSC Proz</td><td>Ŷ</td><td>M ST ST ST ST ST ST ST ST ST ST ST ST ST</td><td>CU M32P031R8Tx M32P030R8Tx M32P03008Tx M32P103VCTx M32P100VCTx M32P100VCTx M32P100VCTx M32P100VCTx M32</td><td></td><td></td></t<>	Board : Type Discovery	MCU Series : All	Ince Projos OSC Proze OSC Proz	Ŷ	M ST ST ST ST ST ST ST ST ST ST ST ST ST	CU M32P031R8Tx M32P030R8Tx M32P03008Tx M32P103VCTx M32P100VCTx M32P100VCTx M32P100VCTx M32P100VCTx M32		

Крайник Я. М., Ухань Є. О.

б)

Рис. 1.1 – Вибір цільового пристрою: а – вибір мікроконтролера; б – вибір відлагоджувальної плати

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Після підтвердження вибору користувачеві стає доступним проєкт із принциповим представленням мікроконтролера (рис. 1.2).



Рис.1.2 – Принципове представлення мікроконтролера у STM32CubeMX

Для подальшої роботи рекомендується зберегти проєкт.

Для конфігурування доступні 4 вкладки:

1) Pinout – для налаштування використання модулів та пінів;

2) Clock Configuration – для налаштування тактування пристрою та конкретних модулів;

3) Configuration – для більш детального налаштування модулів та бібліотек;

4) Power Consumption Calculator – для розрахунку споживання енергії за заданої конфігурації та вибору джерела живлення.

На вкладці Ріпоит можна виконувати налаштування пінів в основній робочій області, натиснувши на пін та обравши відповідну функцію (LPUART1_RTS на рис. 1.3). У випадаючому списку, що відображається після натиснення, доступні усі функції, пов'язані з обраним піном. Проте, такий підхід не активує обраний модуль автоматично. Тому рекомендується активовувати пристрої за допомогою дерева на лівій панелі. Оскільки область відображення й область із деревом пристроїв пов'язані, то внесені зміни будуть застосовані. У той же час, для налаштування у режимі входу/виходу/переривання можна використовувати перший підхід.



Крайник Я. М., Ухань Є. О.

Рис. 1.3 – Налаштування піну у візуальному режимі

Для прикладу розглянемо налаштування модуля SPI1 у режимі двонаправленого обміну даними (рис. 1.4).



Рис. 1.4 – Налаштування SPI

Переведення його у режим Full-Duplex призводить до того, що деякі модулі стають або недоступними (DAC – відображений червоним кольором) або їх використання стає обмеженим (RTC – позначено жовтим), але вони все ще можуть бути задіяні одночасно з модулем SPI1 (рис. 1.5).



Рис. 1.5 – Налаштування RTС-модуля

Великою перевагою графічного конфігуратора є те, що він автоматично проводить перевірку сумісності модулів і, таким чином, можливо уникнути ситуації, коли 2 модулі, несумісні між собою, будуть одночасно активовані, що цілком імовірно, наприклад, при об'єднанні коду з двох різних проєктів.

Більш детальне налаштування модулів можна виконати на вкладці Configuration (рис. 1.6). Модулі групуються відповідно до функціональності, яку вони пропонують.

Middlewares					
Multimedia	Control	Analog	Connectivity	System DMA	
			SPI: Serial Perig This IP is correc	heral Interface	

Рис. 1.6 – Налаштування обраних системних модулів

Для кожного пристрою, що відображається як кнопка, доступне окреме діалогове вікно налаштувань (рис. 1.7).

SPI1 Configuration		\times				
🖋 Parameter Settings 🖋 User Constants 🖋 NVIC Settings 🖋 DMA Settings 🖋 GPIO Settings						
Configure the below parameters :						
Search : Search (Crtl+F)	★ ★					
Basic Parameters						
Frame Format	Motorola					
Data Size	8 Bits					
First Bit	MSB First					
Clock Parameters						
Prescaler (for Baud Rate)	2					
* Baud Rate	1.0485 MBits/s					
Clock Polarity (CPOL)	Low					
Clock Phase (CPHA)	1 Edge					
Advanced Parameters						
CRC Calculation	Disabled					
NSSP Mode	Enabled					
NSS Signal Type	Input Hardware					
Prescaler (for Baud Rate)						
BaudRatePrescaler						
1						
	Apply Ok Ca	ancel				

Рис. 1.7 – Налаштування параметрів SPI

Усі параметри, задані користувачем, враховуються за подальшої генерації проєкту, тому слід перевіряти усі задані значення. Слід звернути увагу на наявність кількох вкладок у вікні. Одна з них призначена для того, щоб вказати, що переривання від конкретного пристрою має бути доступним (рис. 1.8).

		er Settings 🛛 💞 User Cons	Farameter Ser
ption Priority	abled	ble	nterrupt Table
	\sim	nterrupt	PI1 global interru

Рис. 1.8 – Налаштування переривання від SPI

Також доступні налаштування Direct Memory Access для обраної конфігурації. Для цього слід додати нову опцію кнопкою Add та обрати доступні варіанти передачі за допомогою DMA. Наприклад, на рис. 1.9. показано, як виглядає налаштування DMA для SPI1 на отримання даних.

	🖉 User Constants 🛛 🍼 🕅	/IC Settings 🗹 DMA Setti	ngs 🎻 GPIO	Settings
MA Request	Channel	Direction	Priority	,
PI1_RX	DMA1 Channel 2	Peripheral To Memor	y Low	
NMA Posicion Softings			A	dd Delete
DMA Request Settings			Ar	dd Delete Memory
DMA Request Settings Mode Normal		Increment Address	Peripheral	id Delete Memory
DMA Request Settings Mode Normal	~	Increment Address	Ai Peripheral	id Delete Memory

Рис. 1.9 – Налаштування DMA для отримання даних

Зверніть увагу, що налаштування каналу та напряму будуть встановлені автоматично.

Проєкт генерується з використанням файлів бібліотек, які мають бути попередньо завантажені. Перевірити, чи доступний необхідний пакет, можна за допомогою команди *Help/Install New Libraries* у вікні New Libraries Manager. Якщо бібліотеки відсутні, то слід їх встановити (рис. 1.10).

STM32CubeL0 Releases		
Firmware Package for Family STM32L0	1.5.0	1.5.0
Firmware Package for Family STM32L0 (Size : 59.1 MB)		1.4.0
Firmware Package for Family STM32L0 (Size : 56.4 MB)		1.3.0
Firmware Package for Family STM32L0 (Size : 16.41 MB)		1.1.2
Firmware Package for Family STM32L0 (Size : 11.42 MB)		1.0.0

Рис. 1.10 – Доступні бібліотеки у вікні New Libraries Manager

Перед генерацією проєкту слід задати відповідні опції вихідного проєкту. STM32CubeMX підтримує роботу з кількома різними середовищами розробки, тому слід переконатись, що обране необхідне IDE у діалоговому вікні, викликаному командою Project/Settings (рис. 1.11).

oject	Code Generato	r Advanced Settings		
 -				
Proje	ct Settings			
litoje	et name			
Prote	ct Location			
_				Browse
Toolo	chain Folder Loc	tion		
Teele				
1 0010	chain / IDE			
3004	13111132		·	
. Sarlar				
Linke	r Settings	0:400		
Minin	num Heap Size	0:000		
Minin	num Stack Size	0x200		
	and Circuit on D			
Mcu a	anu Firmware Po Roforonco	ckage		
STM	32L053R8Tx			
Eirm	waro Packago N	me and Version		
STM	32Cube FW 10	/1.5.0		
5110	520000 TW_00	1010		

Рис. 1.11 – Налаштування генерації проєкту

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Для генерації проєкту слід скористатись командою меню Project/Generate Code.

Розглянемо код, згенерований конфігуратором. Перш за все, слід відзначити, що проєкт буде розділений на папки відповідно до функціональності коду. На рівні розділення коду він буде розбитий між файлами. Слід відзначити, що структура згенерованого проєкту не буде відповідати такому, що був створений безпосередньо у SystemWorkbench. Він містить лише необхідні для компіляції файли, в той час як стандартний проєкт буде посилатись на проєкт-бібліотеку, що має містити усі доступні файли з бібліотеки конкретного пристрою.

Функція main виглядатиме так, як показано на рис. 1.12.

```
int main(void)
  /* USER CODE BEGIN 1 */
  /* USER CODE END 1 */
 /* MCU Configuration-----
  /* Reset of all peripherals, Initializes the Flash interface and the Systick.
 HAL_Init();
  /* Configure the system clock */
 SystemClock Config();
  /* Initialize all configured peripherals */
 MX GPIO Init();
 MX_SPI1_Init();
  /* USER CODE BEGIN 2 */
  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
 while (1)
  /* USER CODE END WHILE */
  /* USER CODE BEGIN 3 */
```

Рис. 1.12 – Згенерований код проєкту у файлі *main.c*

Код головної функції main() містить виклики

- HAL_Init()
- SystemClock_Config()
- MX_Periph_Init()

Також за допомогою коментарів виділені ділянки, у які можна дописувати власний код. Ці частини гарантовано не будуть змінені, якщо користувач вирішить змінити налаштування у STM32CubeMX та перегенерувати його. Збереження усіх інших частин не гарантується.

Завдання

Створити новий проєкт STM32CubeMX для плати STM32L0 Nucleo з налаштуваннями, описаними у ході роботи. У проєкті обов'язково слід задіяти як мінімум 4 периферійні модулі. Також мають бути увімкнені 3 переривання і 2 канали DMA (один на запис, а інший – на читання). Згенерувати проєкт та продемонструвати його структуру у SystemWorkbench for STM32, попередньо виконавши імпорт згенерованого проєкту.

Лабораторна робота № 2 Робота з GPIO та зовнішніми перериваннями

Мета роботи: продемонструвати розуміння і використання функцій НАL для роботи з GPIO та зовнішніми перериваннями.

Хід роботи

Створити проєкт для плати STM32F4Discovery, що використовується у GlobalLogic Embedded Starter Kit. Проєкт має базуватись на STM32Cube. У налаштуваннях проєкту після вибору плати вказати потрібну інформацію, як показано на рис. 2.1.

DE STM32 Proje	ct	_		ı ×
Setup STM32 p	roject			IDE
Project				
Project Name:	stm32_			
🗹 Use default	location			
Location:	E:/Software/stm32_workspace			Browse
Targeted La C C C+ Targeted Bin Executab	nguage + nary Type le () Static Library			
Targeted Pro	oject Type Ibe ○ Empty			
?	< <u>B</u> ack <u>N</u> ext > <u>F</u> inish		Cá	ancel

Рис. 2.1 – Генерація нового проєкту

За необхідності вказати налаштування Firmware Updater, встановивши шлях до директорії з бібліотеками для STM32F4.

Провести налаштування роботи GPIO у частині STM32Cube:

- 3 піни для роботи в режимі виходів;
- 1 пін в режимі входу і зовнішнього переривання.

Для цього слід ознайомитись зі схемою розміщення компонентів на допоміжній платі. Так, відповідно до опису на рис. 2.2, доступні 5 тактових кнопок, серед яких для виконання даної роботи слід обрати одну, наприклад, PC11.



Рис. 2.2 – Підключення кнопок на платі Embedded Starter Kit

PC11 підключено до кнопки, що позначена як SWT1.

Для того, щоб задати опцію, що пін використовується як вхід з реакцією на зовнішнє переривання, рекомендовано використовувати STM32Cube, як це показано на рис. 2.3.



Рис. 2.3 – Вибір функції GPIO

Після цього у налаштуваннях модулю NVIC вказати, що зовнішнє переривання для ліній 10-15 має бути доступне у контролері переривань (рис. 2.4).

Q	~ 🔅	NVIC Mode and Configuration				
Categories A->Z		Configuration				
System Core	~	ONVIC Sequence Code generation				
<u>م</u>		Priority Group 0 bits for pre-emption priority 4 bits f ~	Sort	by Premption Priority ar	nd Sub Prie	
DMA GPIO		Search I Show only enabled interrupts	Force	e DMA channels Interru	pts	
NVIC		NVIC Interrupt Table	Enabled	Preemption Priority	Sub Pri	
✓ RCC		Non maskable interrupt	\checkmark	0	0	
A SYS		Hard fault interrupt	~	0	0	
WWDG		Memory management fault	\checkmark	0	0	
		Pre-fetch fault, memory access fault	\checkmark	0	0	
		Undefined instruction or illegal state	\checkmark	0	0	
Analog		System service call via SWI instruction	\checkmark	0	0	
Timer		Debug monitor	\checkmark	0	0	
Timers		Pendable request for system service	\checkmark	0	0	
Connectivity	>	Time base: System tick timer	\checkmark	0	0	
Connectivity		PVD interrupt through EXTI line 16		0	0	
Multimedia	>	Flash global interrupt		0	0	
		RCC global interrupt		0	0	
Security	>	I2C1 event interrupt		0	0	
		I2C1 error interrupt		0	0	
Computing	>	SPI1 global interrupt		0	0	
		EXTI line[15:10] interrupts	~	0	0	
Middleware	>	SPI3 global interrupt		0	0	
		USB On The Go FS global interrupt	\checkmark	0	0	
		FPU global interrupt		0	0	

Рис. 2.4 – Налаштування переривання GPIO

Зберегти опції та підтвердити генерацію проєкту відповідно до внесених змін. При налаштуваннях за замовчуванням код проєкту буде генеруватись у момент збереження файлу *ioc*.

На платі STM32F4Discovery наявні 4 світлодіоди, які можна використовувати з метою перевірки роботи простих функцій та відлагодження на базовому рівні. Якщо проєкт був згенерований з опцією підтримки доступних функцій плати, то вони вже готові до використання, як показано на рис. 2.5.



Рис. 2.5 – Налаштування світлодіодів

Якщо був згенерований порожній проєкт, то піни PD12-15 необхідно налаштувати як виходи.

С-файли проєкту, призначені для редагування, розташовані у директорії Core/Src. У цій директорії має бути наявним файл *main.c.* Відкривши його у редакторі, можна побачити, що він розбитий на секції коментарями спеціального виду. Такі коментарі використовуються для того, щоб позначити, яка частина програми може бути відредагована і при цьому залишиться незмінною за повторної генерації проєкту. Приклад такої секції показаний на рис. 2.6.



Рис. 2.6 – Додавання користувацького коду у файл проєкту

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Додавати власний код у проєкт слід строго у відповідні секції.

Для початку роботи з платою слід встановити її у плату розширення і підключити блок живлення на 12 В. Після цього можна підключити відлагоджувальну плату до ПК за допомогою mini-USB. Завдяки наявності на платі вбудованого програматора і відлагоджувача STLink, це підключення використовується для того, щоб програмувати Flashпам'ять пристрою. Також зверніть увагу на те, що без підключення джерела живлення до плати розширення її функціональність не буде доступна для основної плати, тому обидва кабелі мають бути підключені одночасно.

У деяких випадках при спробі запуску проєкту може з'явитись повідомлення, що необхідно оновлення інтерфейсу STLink, яке необхідно провести за допомогою майстра, який показаний на рис. 2.7.

STLinkUpgrade 3.3.2	- 🗆 ×
ST-LINK/V2-1 Cefresh device list Open in update mode ST-Link ID: 0670FF303931594E43105446 Current Firmware: Type: STM32 Debug+Mass storage+VCP	Releasing your creativity
Version: V2J25M14 Update to Firmware: V2J35M26 STM32 Debug+Mass storage+VCP	

Рис. 2.7 – Оновлення прошивки STLink

Коли необхідні підключення виконані, можна запустити проєкт у режимі відлагодження. Для цього слід запустити команду контекстного меню Debug As/STM32 Cortex-M C/C++ Application (рис. 2.8).

0	Show in Remote Systems view Run As	>	417 [©] void Error_Handler(void) 418 {
*	Debug As	>	I Local C/C++ Application
	Profile As	>	2 STM32 Cortex-M C/C++ Application
	Team	>	Debug Configurations
	Compare With	>]	

Рис. 2.8 – Запуск проєкту

Командами графічного інтерфейсу, або за допомогою клавіш F6-F8 можна покроково виконувати команди. Проте, оскільки власного функціоналу не було реалізовано, то цей етап можна розглядати як ознайомлення із середовищем розробки. Далі слід вийти з режиму відлагодження.

Обробник зовнішнього переривання вже доданий у проєкт (у файлі stm32f4xx_it.c) і, відповідно до підходу, що пропагується HAL, залишилось реалізувати функцію зворотного виклику (англ. callback). Така функція може бути додана до будь-якого файлу, але зазвичай її додають у main.c. Peaniзація callback'у продемонстрована на рис. 2.9.

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16 t GPIO Pin)
{
    if (GPIO Pin == GPIO PIN 11) {
        if (button_state == 0) {
            HAL GPIO WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_SET);
            HAL GPIO WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
            button_state = 1;
        } else {
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_12, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_13, GPIO_PIN_RESET);
            HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
            button state = 0;
        }
    }
}
/* USER CODE END 4 */
```

Рис. 2.9 – Додавання тестового коду у проєкт

Як видно з рис. 2.9, функція перевіряє номер піну і значення змінної для того, щоб інвертувати значення світлодіодів.

Реалізувати функціональність відповідно до варіанту. Реакцію на натиснення кнопки обов'язково реалізовувати за допомогою зовнішнього переривання.

Варіанти завдань

1. Реалізувати ефект, коли при натисненій кнопці вмикаються світлодіоди по одному, якщо натиснення немає — в іншому напрямку.

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

2. Реалізувати ефект, коли при натисненій кнопці вмикаються всі діоди, а потім вимикаються, якщо натиснення немає – спочатку все вимкнено, а потім вмикається по одному.

3. Якщо кнопка натиснена, то пристрій працює як світлофор, якщо відтиснута – почергово 2 блимання кожним світлодіодом.

4. Якщо кнопка натиснена, то почергово передається 3 світлодіодами сигнал SOS, якщо відтиснута – сигнал OSO.

Лабораторна робота № 3 Реалізація роботи з паралельною шиною

Мета роботи: навчитись реалізувати роботу паралельного інтерфейсу для керування LCD-дисплеєм.

Хід роботи

На допоміжній платі GlobalLogic Embedded Kit встановлено LCDдисплей WH1602, який керується контролером HD44780. Відповідно до рис. 3.1, дисплей підключено за допомогою паралельної шини.



Рис. 3.1 – Схема підключення дисплея

Як зрозуміло з рис. 3.1, для передачі даних використовується 4-бітна шина. Така шина передбачає, що дані передаються у режимі, коли за 1 такт передається лише півбайту. Також наявні 3 службові сигнали:

1) RS (Register Select) – для вибору значення, яке позначає інформація на шині даних (команда або дані);

2) R/W (Read/Write) – для вибору напряму; у даній роботі використовується лише режим запису, тому відповідний пін можна встановити у лог. «0»;

3) Е (Enable) – сигнал дозволу, має бути встановлений в лог. «1» під час передачі даних або команд.

Ознайомитись з особливостями ініціалізації та передачі даних на дисплей (файл LCD Interfacing using HD44780 Hitachi chipset compatible LCD.pdf) та реалізувати завдання за варіантом.

Завдання

1. Вивести ім'я у першому рядку, після чого воно через певний час переміщується у кінець та зникає і з'являється у нижньому рядку, і так само проходить до кінця.

2. Вивести рядок з числами (0-F) таким чином: спочатку з'являється «0», потім «1» і. т. д. Після того, як рядок заповнений, «0» зникає, через деякий час зникає «1» і т. д. Повторити для 2-го рядка.

3. Одночасно у 1-му та 2-му рядку є анімація слова. На початку слова розташовані на різних кінцях та починають зближуватись, переходять до протилежного кінця. Слова однакової довжини.

4. Те саме, що для п. 3, але слова різної довжини. Кожен цикл має починатися одночасно для обох слів.

Лабораторна робота № 4 Робота з таймерами. Широтно-імпульсна модуляція (ШІМ)

Таймери є важливими компонентами мікроконтролера з погляду планування виконання задач, організації затримок, вимірювання часових інтервалів. Коректне управління таймерами на програмному рівні дозволяє підвищити ефективність виконання програм з точки зору швидкодії та реакції на зовнішні події. Окрім цього, вони пов'язані з функціями входів та виходів мікроконтролера. Саме тому робота з таймерами є важливою складовою при написанні комплексних програм для мікроконтролерів.

Мета роботи: отримати навички роботи з таймерами за допомогою бібліотеки HAL, налаштуванню переривань таймерів, організації затримок за допомогою таймерів, а також генерації сигналу ШІМ за допомогою таймерів.

Хід роботи

Створити новий проєкт для STM32F4Discovery у STM32CubeIDE.

Мікроконтролери STM32 серії F4 характеризуються наявністю великої кількості таймерів. У даному конкретному випадку доступно одразу 14 таймерів-лічильників, а також 1 модуль таймера реального часу.

У термінології STM вони поділяються на [1]:

1) таймери з розширеною функціональністю (Advanced-control timers);

2) таймери загального призначення (General purpose timers);

3) базові таймери (Basic timers).

Основні відмінності пов'язані з кількістю налаштувань і їх функціональністю, а також із розрядністю таймерів.

Основною подією, яка використовується при роботі з таймером є оновлення таймера (timer update event). Така подія може призводити до переривання. Затримка з точки роботи з таймером, означає, що контекст виконання програми не має залишатись у обробнику, а може перейти до основного циклу виконання. Такий варіант є більш ефективним щодо використання обчислювального ядра пристрою.

Для того, щоб реалізувати затримку на основі переривання, слід:

- 1) налаштувати параметри таймера;
- 2) дозволити переривання оновлення таймера;
- 3) реалізувати обробник переривання.

Для того, щоб коректно організувати затримки, необхідно навчитись правильно розраховувати значення, зазвичай, для 2 регістрів для таймера:

peгicтру ARR;

- переддільника.

У випадку реалізації вимірювання дуже великих часових інтервалів може знадобитись також підключення регістру підрахунку повторень.

Введемо позначення для параметрів:

*Clk*_{timer} – вхідна частота таймера;

Clk_{update} – частота генерації події оновлення таймера.

Зазвичай, задача зводиться до того, щоб задати значення переддільника та perictpy ARR, щоб отримати необхідну частоту появу події:

$$Clk_{update} = \frac{1}{(PSC-1) \cdot (ARR-1)} \cdot Clk_{timer}$$

Оскільки частота події оновлення таймера є відомою, то достатньо обрати одне зі значень PSC або ARR, а на основі цього обчислити друге значення:

$$(PSC-1) = \frac{1}{(ARR-1) \cdot Clk_{update}} \cdot Clk_{timer}$$

При реалізації за допомогою бібліотеки НАL слід задати відповідні розраховані значення полям структури та ініціалізувати таймер.

```
TIM HandleTypeDef TimHandle;
/* Compute the prescaler value to have TIM3 counter
clock equal to 10 KHz */
  uwPrescalerValue = (uint32 t) ((SystemCoreClock /2) /
10000) - 1;
  /* Set TIMx instance */
  TimHandle.Instance = TIMx;
  /* Initialize TIM3 peripheral as follow:
       + Period = 10000 - 1
       + Prescaler = ((SystemCoreClock/2)/10000) - 1
       + ClockDivision = 0
       + Counter direction = Up
  */
  TimHandle.Init.Period = 10000 - 1;
  TimHandle.Init.Prescaler = uwPrescalerValue;
```

```
TimHandle.Init.ClockDivision = 0;
TimHandle.Init.CounterMode = TIM_COUNTERMODE_UP;
TimHandle.Init.AutoReloadPreload =
TIM_AUTORELOAD_PRELOAD_DISABLE;
if(HAL_TIM_Base_Init(&TimHandle) != HAL_OK)
{
    /* Initialization Error */
    Error_Handler();
}
```

Завдяки встановленому значенню переддільника, вхідна тактова частота «знижується» до 10 кГц, а далі – встановлюється значення регістру перезавантаження, яке понижує частоту події до 1 Гц. Відповідно, у цьому випадку подія буде виникати 1 раз в секунду.

Далі слідує виклик функції дозволу переривання та початку роботи таймера в режимі переривання (у згенерованому проєкті можуть бути розташовані у різних файлах):

Також слід перевірити, що у файлі *stm32f4xx_it.c* є виклик стандартного обробника:

```
void TIMx_IRQHandler(void)
{
    HAL_TIM_IRQHandler(&TimHandle);
}
```

а також реалізована функція зворотного виклику (зазвичай, у файлі *main.c*).

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef
*htim)
{
   BSP_LED_Toggle(LED4);
}
```

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

У такому випадку відбуватиметься переключення стану світлодіоду кожну секунду.

Налаштування ШІМ у таймері. Керування сигналом ШІМ за допомогою таймера забезпечується завдяки наявності регістрів Capture/Compare (CC). У даному випадку вони використовуються для того, щоб визначити стан логічного сигналу на виході каналу ШІМ.

Сигнал ШІМ задається за допомогою таких параметрів:

1) частота;

2) коефіцієнт заповнення.

У випадку використання таймера частота ШІМ буде визначатись частотою сигналу, який отримується після переддільника, а коефіцієнт заповнення – співвідношенням значень у регістрі СС для відповідного каналу та у регістрі автозавантаження.

Для налаштування каналу ШІМ використовується структура типу TIM_OC_InitTypeDef, за допомогою якої вказується поведінка сигналу, а також коефіцієнт заповнення:

```
TIM OC InitTypeDef sConfig;
  /*##
           Configure
                          the
                                   PWM
                                             channels
/* Common configuration for all channels */
  sConfig.OCMode = TIM OCMODE PWM1;
  sConfig.OCPolarity = TIM_OCPOLARITY_HIGH;
sConfig.OCFastMode = TIM_OCFAST_DISABLE;
  sConfig.OCNPolarity = TIM OCNPOLARITY HIGH;
  sConfig.OCNIdleState = TIM OCNIDLESTATE RESET;
  sConfig.OCIdleState = TIM OCIDLESTATE RESET;
  /* Set the pulse value for channel 1 */
  sConfig.Pulse = PULSE1 VALUE;
  if (HAL TIM PWM ConfigChannel(&TimHandle, &sConfig,
TIM CHANNEL 1) != HAL OK)
  {
   /* Configuration Error */
   Error Handler();
  }
  HAL TIM PWM Start(&TimHandle, TIM CHANNEL 1);
```

У цьому випадку PULSE1_VALUE – константа. Для коректної роботи в такому режимі слід виконати виклик функції, що активує генерацію ШІМ (HAL_TIM_PWM_Start).

Варіанти завдань

1. Реалізувати затримку довжиною 2 с та 3 с. Реалізувати генерацію сигналу ШІМ для 2 каналів: зміна першого – від 1% до 99%, другого – від 99% до 1%.

2. Реалізувати затримку довжиною 1 с та 4 с. Реалізувати генерацію сигналу ШІМ для 3 каналів: зміна коефіцієнту заповнення від 1 до 99%.

3. Реалізувати затримку довжиною 3 с та 1 с. Реалізувати генерацію сигналу ШІМ для 3 каналів: починають з рівня 50%, далі 1 і 3 – зменшуються до 0, а 2 – збільшуються до 99%.

4. Реалізувати 2 види затримки – на основі переривання і блокуючу – за допомогою таймера для інтервалу 1 с. Реалізувати генерацію сигналу ШІМ для 3 каналів: спочатку збільшується від 1 до 33% і повертається в 1, далі до 66%, і потім до 99%.

Примітка: для каналів ШІМ у завданні вказана зміна коефіцієнту заповнення.

Лабораторна робота № 5 Робота з модулем Direct Memory Access (DMA)

Робота з модулем DMA ϵ важливою з погляду реалізації програм для сучасних мікроконтролерів, оскільки вона дозволя ϵ розвантажити ядро процесора від виконання однотипних задач, пов'язаних із копіюванням даних та передачі даних з/у периферію.

Мета роботи: навчитись користуватися можливостями DMA за допомогою бібліотеки HAL, освоїти організацію програми при використанні DMA, навчитися визначати швидкодію DMA та порівнювати перевагу від використання DMA і виконанні операцій безпосередньо процесором.

Хід роботи

У цій роботі передбачено виконання наступних етапів:

1) передача даних з однієї області даних в іншу (перезапис даних із масивів);

 використання DMA для передачі даних за допомогою периферійного інтерфейсу;

3) використання DMA для отримання даних з периферійного інтерфейсу.

Для виконання першого етапу рекомендовано виділити адресний простір для двох масивів розміром у декілька кбайт. Один із масивів слід ініціалізувати за допомогою циклу, наприклад, значеннями ітератору. Далі необхідно налаштувати модуль DMA для того, щоб передати дані з ініціалізованого масиву в інший.

Доступ до шин пам'яті мають обидва модулі DMA, тому для передачі в режимі Memory-to-Memory (M2M) можна використовувати будьякий з них.

Розглянемо на прикладі, як реалізувати передачу даних з одного масиву в інший з використанням DMA.

Основна інформація щодо налаштувань DMA міститься у полях структури, що відповідає за ініціалізацію модулю. Для випадку передачі у режимі M2M це може бути виконано таким чином:

DMA_HandleTypeDef hdma;
/* DMA controller clock enable */
__HAL_RCC_DMA2_CLK_ENABLE();

/* Configure DMA request hdma_memtomem_dma2_stream0 on DMA2 Stream0 */

```
hdma.Instance = DMA2 Stream0;
hdma.Init.Channel = DMA CHANNEL 0;
hdma.Init.Direction = DMA MEMORY TO MEMORY;
hdma.Init.PeriphInc = DMA PINC ENABLE;
hdma.Init.MemInc = DMA MINC ENABLE;
hdma.Init.PeriphDataAlignment = DMA PDATAALIGN BYTE;
hdma.Init.MemDataAlignment = DMA MDATAALIGN BYTE;
hdma.Init.Mode = DMA NORMAL;
hdma.Init.Priority = DMA PRIORITY LOW;
hdma.Init.FIFOMode = DMA FIFOMODE ENABLE;
hdma.Init.FIFOThreshold = DMA FIFO THRESHOLD FULL;
hdma.Init.MemBurst = DMA MBURST SINGLE;
hdma.Init.PeriphBurst = DMA PBURST SINGLE;
if (HAL DMA Init(&hdma) != HAL OK)
{
 Error Handler();
}
```

Так виконується ініціалізація модулю DMA для передачі з однієї області пам'яті в іншу. Відповідник наведених вище налаштувань у графічному представлені конфігуратору продемонстрований на рис. 5.1.

	Cor	figuration		
📀 DMA1 🛛 😒 DMA2	✓ MemToMem			
DMA Request	Stream	Direction		Priority
MEMTOMEM	DMA2 Stream 0	Memory To Memory	y Low	
Add Delete DMA Request Settings –			Src Memory	Dst Memory
Mode Normal	\checkmark	Increment Address		
Use Fifo 🔽 Threshol	ld Full 🗸	Data Width	Byte 🗸	Byte ~
		Burst Size	Single \vee	Single ~

Рис. 5.1 – Налаштування DMA

Щойно код ініціалізації додано у проєкт, безпосередньо передача потребує виклику лише однієї функції. Проте, попередньо слід також підготувати області пам'яті, з/у які буде виконуватись читання/запис: Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

```
/* memory regions in RAM */
uint32_t source[50];
uint32_t destination[50];
// init source array in some way
HAL_DMA_Start(&hdma, (uint32_t)&source,
(uint32_t)&destination, 50);
HAL_DMA_PollForTransfer(&hdma, HAL_DMA_FULL_TRANSFER,
HAL_MAX_DELAY);
```

У цьому випадку передача відбувається у режимі опитування (polling) з погляду процесора, що не зовсім виправдано з урахуванням основної переваги DMA – передачі без участі процесора. Тому більш практичним варіантом є сценарій із використанням переривання.

За замовчуванням при використанні передачі DMA за участю переривань автоматично вмикаються усі пов'язані з цим переривання, в т.ч. обробки помилок. Тому достатньо додати загальний обробник переривань DMA і відповідну функцію зворотного виклику.

Розглянемо типовий приклад використання переривання для сигналізації завершення передачі даних:

```
/* initialization code stays the same */
HAL_DMA_RegisterCallback(&hdma,
HAL_DMA_XFER_CPLT_CB_ID, TransferComplete);
HAL_DMA_RegisterCallback(&hdma,
HAL_DMA_XFER_ERROR_CB_ID, TransferError);
HAL_NVIC_SetPriority(DMA_STREAM_IRQ, 0, 0);
HAL_NVIC_EnableIRQ(DMA_STREAM_IRQ);
HAL_DMA_Start_IT(&DmaHandle, (uint32_t)&source,
(uint32_t)&destination, 50);
static void TransferComplete(DMA_HandleTypeDef
*DmaHandle)
{
    BSP_LED_On(LED4);
}
```

У файл опису переривань має бути додано сегмент, що відповідатиме за виклик загального обробника.

```
extern DMA_HandleTypeDef hdma;
void DMA_STREAM_IRQHANDLER(void)
{
```

```
/* Check the interrupt and clear flag */
HAL_DMA_IRQHandler(&DmaHandle);
}
```

Зверніть увагу на те, що передача за допомогою DMA потребує вказання параметру кількості даних, що може призводити до ситуацій, коли певна частина програми не виконується через те, що цього протоколу не дотримано.

Перевіряти дані у пам'яті слід за допомогою представлення View обов'язково з точкою відлагодження у момент, коли дані мають бути записані (якщо програма виконується, то це представлення просто не буде працювати). Для переходу необхідної області можна ввести безпосередньо адресу у різних числових представленнях або, що є більш зручним, ім'я змінної масиву.

Найбільш характерним прикладом використання режиму M2M є передача даних із FLASH-пам'яті до оперативної пам'яті на початку роботи із зображеннями, а також операції з перезаписом графічного буферу.

За роботи з периферійними пристроями обов'язковою умовою є узгодження щодо використання каналу і потоку в модулі DMA. Для цього слід використовувати інформацію, наведену у таблицях на рис. 5.2–5.3 (таблиці з Reference Manual для мікроконтролерів STM32F4).

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	SPI3_RX	-	SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX	-	SPI3_TX
Channel 1	I2C1_RX	-	TIM7_UP	-	TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
Channel 2	TIM4_CH1	-	I2S3_EXT_ RX	TIM4_CH2	I2S2_EXT_ TX	I2S3_EXT_ TX	TIM4_UP	TIM4_CH3
Channel 3	I2S3_EXT_ RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_ RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
Channel 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
Channel 5	UART8_TX ⁽¹⁾	UART7_TX ⁽¹⁾	TIM3_CH4 TIM3_UP	UART7_RX ⁽¹⁾	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX ⁽¹⁾	тімз_снз

Рис. 5.2 – Відповідність каналів DMA та потоків

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1	SAI1_A ⁽¹⁾	TIM8_CH1 TIM8_CH2 TIM8_CH3	SAI1_A ⁽¹⁾	ADC1	SAI1_B(1)	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
Channel 1	-	DCMI	ADC2	ADC2	SAI1_B(1)	SPI6_TX ⁽¹⁾	SPI6_RX ⁽¹⁾	DCMI
Channel 2	ADC3	ADC3	-	SPI5_RX ⁽¹⁾	SPI5_TX ⁽¹⁾	CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX	-	SPI1_RX	SPI1_TX	-	SPI1_TX	-	-
Channel 4	SPI4_RX ⁽¹⁾	SPI4_TX ⁽¹⁾	USART1_RX	SDIO	-	USART1_RX	SDIO	USART1_TX
Channel 5	-	USART6_RX	USART6_RX	SPI4_RX ⁽¹⁾	SPI4_TX ⁽¹⁾	-	USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	-
Channel 7	-	TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX ⁽¹⁾	SPI5_TX ⁽¹⁾	TIM8_CH4 TIM8_TRIG TIM8_COM

Рис. 5.3 – Відповідність каналів DMA та потоків

Слід розуміти, що для одного потоку в конкретний момент часу активним може бути лише один канал. Саме тому можна за наведеними таблицями простежити, що для одного пристрою функції розподілені між різними потоками, але при цьому мають однаковий номер каналу. Для того, щоб уникнути таких конфліктів, можна використовувати конфігуратор, як показано на рис. 5.4 (конфлікти будуть визначені автоматично).



Рис. 5.4 – Активація модулів USART

У проєкті-шаблоні використовуються модулі USART2/3, які сконфігуровані для роботі в асинхронному режимі.

У результаті цих налаштувань для послідовних інтерфейсів має оновитись також налаштування модулю DMA (рис. 5.5).

DMA Mode and Configuration				
	Cor	nfiguration		
📀 DMA1 🛛 📀 DMA2	MemToMem			
DMA Request	Stream	Direction		Priority
USART2_TX	DMA1 Stream 6	Memory To Peripheral	Low	
USARI3_RX	DMA1 Stream 1	Peripheral To Memory	Low	
Add Delete				
DMA Request Settings-				
		Per	ipheral	Memory
Mode Normal	~	Increment Address		
Use Fifo 🔲 Thresh		Data Width Byte	\sim	Byte ~
		Burst Size	\sim	~

Рис. 5.5 – Активація запитів DMA для модулів USART

Обидва потоки асоційовані з модулем DMA1.

Для виконання 2-го та 3-го етапів завдання рекомендується використовувати 2 модулі послідовної передачі даних, один з яких працює на запис, а інший – на читання. Найпростішим варіантом з точки зору кількості підключення є використання інтерфейсів UART або I2C, оскільки вони потребують лише двох з'єднань. Тим не менш, можна використовувати й інші інтерфейси, які будуть сконфігуровані відповідним чином. При цьому, для одного модуля має бути спочатку викликана функція на отримання даних за допомогою DMA, а потім – для іншого викликається функція на передачу. Таким чином, пристрій буде очікувати на надходження даних перед тим, як дані будуть надіслані. Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

На цьому етапі, окрім ініціалізації DMA, необхідно також ініціалізувати два периферійні модулі (ця частина може бути повністю реалізована за допомогою графічного конфігуратору у STM32CubeIDE). Для UART ця частина може виглядати таким чином:

```
UART_HandleTypeDef UartHandle;
UartHandle.Instance = USARTx;
UartHandle.Init.BaudRate = 9600;
UartHandle.Init.WordLength = UART_WORDLENGTH_8B;
UartHandle.Init.StopBits = UART_STOPBITS_1;
UartHandle.Init.Parity = UART_PARITY_NONE;
UartHandle.Init.HwFlowCtl = UART_HWCONTROL_NONE;
UartHandle.Init.Mode = UART_MODE_TX_RX;
UartHandle.Init.OverSampling = UART_OVERSAMPLING_16;
```

HAL_UART_Init(&UartHandle);

Взаємодія між модулями може бути описана як показано у лістингу:

```
HAL_UART_Transmit_DMA(&UartHandle, (uint8_t*)aTxBuffer,
TXBUFFERSIZE)!= HAL_OK
```

HAL_UART_Receive_DMA(&UartHandle, (uint8_t *)aRxBuffer, RXBUFFERSIZE) != HAL_OK

Переривання пристроїв і переривання DMA. Модуль DMA здатен генерувати власні переривання. Саме тому доцільно використовувати переривання DMA для того, щоб відслідковувати стан передачі (додати у проєкт самостійно за прикладом, наведеним вище). Також варто пам'ятати, що при використанні переривань пристроїв вони так само можуть бути активними, тому для досягнення вищої продуктивності їх слід відключити.

Варіанти завдань

Усі завдання передбачають виконання передачі даних в режимі М2М з одного масиву до іншого (розмір масиву – від 1000 елементів). У режимі відлагодження продемонструвати, що дані скопійовані у цільову область (за допомогою вікна Memory).

Запустити передачу даних між двома послідовними інтерфейсами у режимі DMA – і передача, і отримання мають бути виконанні за допомогою DMA. Використовувати переривання та режим відлагодження для того, щоб продемонструвати коректну передачу даних. Код проєкту-шаблону доступний у репозиторії <u>https://github.com/codebreaker7/MC_Lab5_template</u>.

Лабораторна робота № 6 Робота з інтерфейсом SPI

Serial Peripheral Interface (SPI) є одним із найбільш поширених інтерфейсів для керування зовнішніми пристроями, отримання даних з датчиків та ін. Тому робота з цим інтерфейсом є важливою, а код для роботи з одним пристроєм може бути адаптований під інші відповідно до протоколу взаємодії. Дисплеї є одними з типових пристроїв, що керуються по SPI.

Мета роботи: засвоєння навичок керування модулем дисплея за допомогою інтерфейсу SPI.

Хід роботи

Під час виконання даної роботи передбачається використання дисплея на базі контролера ILI9163. Такий дисплей має вбудований графічний буфер, з якого зображення виводиться, тому для його оновлення необхідно передавати дані за допомогою інтерфейсу SPI. Дисплей, що керується таким контролером, підтримує різні формати представлення пікселів, серед яких RGB565 відповідає найбільшій якості.

Робота з дисплеєм передбачає його ініціалізацію та передачу даних у графічний буфер. Ініціалізація полягає у передачі послідовності службових повідомлень, на основі яких виконується налаштування дисплея. З погляду інтерфейсу SPI на даній стадії виконується передача значень із виділеного масиву констант із необхідними часовими затримками між повідомленнями (як це передбачено протоколом керування).

Вивід графічної інформації розпочинається після ініціалізації і може бути визначений як заповнення необхідних областей пам'яті графічною інформацією. При цьому, після вказання початкової адреси, а також ширини і висоти виділеної області, починається (має починатись) видача даних, що представляють пікселі у зображенні. У випадку, коли оновлюється вся область, дані з оперативної пам'яті контролера передаються на дисплей, починаючи з адреси (0, 0) та відповідно до ширини і висоти дисплея. Відповідно, для реалізації виведення і оновлення зображення достатньо реалізувати лише 2 функції. Друга функція може бути реалізована як параметризована для того, щоб забезпечити вивід зображення у довільну область на дисплеї.

З погляду організації пам'яті у програмі, має бути виділений масив, який використовуватиметься як графічний буфер у мікроконтролері і дані з якого передаватимуться на дисплей. Для того, щоб почати працювати безпосередньо з дисплеєм, на початку необхідно подати послідовність команд, які виконають його ініціалізацію. Загалом, можна відзначити, що робота з дисплеєм складається з того, що контролеру дисплея відправляються або команди, або дані (за це відповідає пін АО: якщо слід відправити команду, то його значення – «0», дані – «1»). При ініціалізації дисплея використовуються не лише команди, а і дані, для того, щоб встановити параметри роботи дисплея.

Після того, як дисплей проініціалізовано, на ньому можна побачити зображення типу білий шум (у перший раз за умови, що достатньо довго не було живлення і що не виводилось ніяких зображень). Після цього можна переходити безпосередньо до того, щоб керувати тим, що відображається на екрані. Розмір екрану становить 128×128 пікселів, а для представлення 1 пікселя використовуються 2 байти (формат RGB565). Відповідно, уся пам'ять зображень у такому випадку займатиме понад 32 кбайт. Контролер дисплея відповідає за те, щоб виводити зображення з області графічної пам'яті на дисплей. Для того, щоб записати дані у область графічної пам'яті, необхідно для початку встановити вікно – область, у яку відбуватиметься запис, – а після цього перейти безпосередньо до запису даних у цю область пам'яті.

Відповідно, рекомендується використовувати наступний алгоритм:

- 1) підготувати масив, заповнивши його необхідними даними;
- 2) встановити адресу для виведення;
- 3) оновити відповідну частину екрану, передавши підготовлені дані.

Такий принцип необхідно буде реалізувати відповідно до завдання.

Завдання

Підключити дисплей до плати, запустити програму-приклад та перевірити його роботу. Дисплей зручно підключати за допомогою макетної плати. Живлення та землю можна подавати безпосередньо з плати. На піни VCC, LED, RST слід подати лог. «1». GND – підключити до піна GND на платі. Інші піни підключаються відповідно до налаштувань SPI.

1. Реалізувати прохід зеленої «змійки» по діагоналі. Вона з'являється та зникає через деякий час.

2. Реалізувати обхід червоною «змійкою» периметру дисплея.

3. Реалізувати заповнення дисплея різнокольоровими квадратами з подальшим їх зникненням.

4. Реалізувати виведення на дисплей 3 будь-яких чисел або літер.

Лабораторна робота № 7 Робота мікроконтролерів у режимах низького енергоспоживання

Мікроконтролери часто стають базою для систем, які живляться від автономного джерела живлення, тому актуальним у таких системах є продовження часу роботи від наявного заряду. Найбільш поширеним засобом для подовження часу роботи є використання режимів низького споживання енергії.

Метою роботи є дослідження роботи мікроконтролерів у режимах низького споживання та отримання навичок по використанню типових підходів для роботи з ними.

Хід роботи

Початково, після запуску мікроконтролер завжди буде знаходитись у активному стані RUN. Проте, у цьому стані споживається найбільше енергії, тому доцільно, якщо мікроконтролер у конкретний момент часу не зайнятий виконанням функцій, перевести його у один із режимів низького споживання.

Найбільш простим із точки зору розуміння є перехід у режим Sleep. У цьому режимі відключається лише ядро, а всі інші елементи продовжують повноцінно працювати. Загалом, варто відзначити, що безпосередньо перехід у конкретний стан не потребує написання складних програмних конструкцій, а завдяки бібліотеці НАL забезпечується викликом 1-2 команд. Наприклад, для переходу в режим Sleep використовується функція:

```
void HAL_PWR_EnterSLEEPMode(uint32_t Regulator, uint8_t
SLEEPEntry);
```

Перший параметр у ній не використовується і зберігається лише для сумісності. Другий параметр вказує, у результаті чого відбудеться вихід з стану, переривання чи події. Так, PWR_SLEEPENTRY_WFI вказує, що вихід буде по перериванню, а PWR_SLEEPENTRY_WFE – по події. У першому випадку вихід буде здійснено по виникненню переривання, у другому – по події.

Також, оскільки керування переходом до відповідного режиму відбувається за допомогою регістру PWR, він має бути доступний і підключений до джерела тактової частоти:

```
__HAL_RCC_PWR_CLK_ENABLE();
```

Після цього виклику, відповідний регістр стає доступним. Оскільки бібліотека HAL має активоване системне переривання, що використовується для вимірювання часу, то це переривання необхідно вимкнути переходом до режиму сну, оскільки, інакше, через 1 мс відбудеться вихід з цього режиму. Це забезпечується функцією:

```
HAL SuspendTick();
```

Для переходу в режим Stop, який є більш просунутим режимом низького споживання, оскільки дозволяє відключити не лише ядро, а і периферію, що суттєво впливає на споживання. Перехід до цього режиму забезпечується викликом:

```
void HAL_PWR_EnterSTOPMode(uint32_t Regulator, uint8_t
STOPEntry);
```

Останнім режимом є режим StandBy, який забезпечує найнижче споживання (фактично, мікроконтролер вимикається і відновлює свою роботу через Reset). Перехід до цього стану відбувається за допомогою функцій:

```
HAL_PWR_EnableWakeUpPin(PWR_WAKEUP_PIN1);
HAL_PWR_EnterSTANDBYMode();
```

Перший виклик необхідний для роботи т. зв. Wakeup Pin, проте, він не є обов'язковим.

Для того, щоб наочно переглянути, яким саме є споживання у конкретному випадку, слід підключити до відлагоджувальної плати мультиметр в режимі амперметра до перемички IDD (попередньо слід зняти перемичку та, бажано, зафіксувати щупи мультиметру). Підключення показано на рис. 7.1.



Рис. 7.1 – Підключення амперметра до плати для вимірювання споживання

Завдання

Виконати переведення мікроконтролера у 3 режими низького споживання та переглянути досягнуті показники за допомогою мультиметру.

Лабораторна робота № 8 Робота з датчиком за допомогою інтерфейсу I2C

Inter-Integrated Circuit (I2C) інтерфейс є одним із найбільш популярних апаратних інтерфейсів для підключення датчиків до мікроконтролерів. Це пояснюється його відносною простотою, а також можливістю підключення багатьох пристроїв за допомогою лише двох апаратних пінів. Іншим вагомим фактором є те, що протокол взаємодії з датчиками є доволі схожим, якщо порівнювати основні моменти взаємодії, тому є можливість швидко адаптовувати наявні програмні рішення під поточні потреби щодо підключення сенсорів.

Мета роботи: набути навичок програмування для взаємодії із зовнішніми датчиками за допомогою послідовного інтерфейсу I2C. Зокрема, реалізовувати використання магнетометра у якості компаса/вимірювача кутів.

Хід роботи

Плата розширення GlobalLogic Embedded StarterKit розміщує 9-осьовий датчик STM LSM9DS1, який поєднує функціональність акселерометра, гіроскопа та магнетометра. Саме функціональність останнього буде використовуватись у роботі. Перш за все, слід перевірити підключення датчика за допомогою принципової схеми плати (рис. 8.1).



Рис. 8.1 – Схема підключення 9-осьового датчика

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Згідно з інформацією на схемі, адреса пристрою магнетометра – 0х1Е. Це значення при передачі у функції запису/читання HAL необхідно зсувати на один біт вліво, оскільки останній біт позначає операцію, яку необхідно виконати з отриманими даними. Також зі схеми можна визначити, які саме піни використовуються для взаємодії з датчиком (PB6, PB9), відповідно, ці піни мають бути активовані в режимі альтернативної функції AF. З точки апаратного модуля I2C на стороні мікроконтролера ці піни відповідають модулю I2C3. Відповідно, цей модуль має бути активований (рис. 8.2).



Рис. 8.2 – Активовані піни модулю І2С

У випадку конфігурації периферії за замовчуванням цей модуль буде активований, оскільки він вже використовується для підключення мікрофона. Але, оскільки адреси пристроїв є різними, то конфлікту у цьому випадку немає.

З погляду програмної взаємодії із сенсором, то зручним інструментом у цьому випадку є функції бібліотеки HAL – HAL_I2C_Mem_Write, HAL_I2C_Mem_Read. Вони дозволяють виконувати операції читання та запису з/у регістри пристрою одразу після ініціалізації пристрою.

Функціонування магнетометра визначається багатьма налаштуваннями в регістрах пристрою, серед яких можна виділити такі:

- чутливість;
- частота оновлення даних;

- режим роботи;
- налаштування порогу переривання та типу реакції;
- калібрувальні значення;
- регістри даних.

У ході виконання роботи пропонується використовувати режим постійного перетворення без переривань для виведення значень показань магнетометра та подальшого перетворення з частотою оновлення даних 10 Гц та обраною чутливістю.

Вказані параметри ініціалізуються на початку роботи, після чого передбачається постійне зчитування даних згідно з налаштуваннями. Проте, для коректного використання функціоналу датчика як компаса, необхідно провести калібрування. Під час калібрування виконується накопичення набору даних у масив заданого розміру та аналіз з погляду найбільшого та найменшого значень. Калібрувальне значення, яке необхідно записати у відповідні регістри, обчислюватиметься як

$$offset = \frac{val_{max} + val_{min}}{2}$$

У стані калібрування бажано виконувати обертання плати за осями, відносно яких планується проводити вимірювання. У випадку функціональності компасу, нас цікавитимуть лише вісі X та Y. Відповідно, обертання можна проводити лише в горизонтальній площині. Якщо побудувати графік залежності сирих показників магнетометра, то можна спостерігати такий результат, як показано на рис. 8.3.



Рис. 8.3 – Приклад отриманих у процесі калібрування значень

Як видно, значення не вирівняні відносно початку осі координат.

Також бажано, щоб у процесі калібрування датчик не потрапляв під дію сторонніх завад (металічні предмети), оскільки це може спотворити

результат калібрування. Відповідне значення має бути записане у регістри корегування (offset). У результаті значення, які отримані при вимірюванні сенсором будуть автоматично зменшені на вказані величини для відповідних осей. Особливістю доступного модуля магнетометра є те, що дані та корегувальні значення представлені 16-бітним значеннями, відповідно, одразу 2 регістри (1 регістр представлений 8 бітами) мають бути зчитані для отримання одного вимірювання магнетометра. Комбінування їх для отримання значення вимірювання проводиться відповідно до стандартного підходу:

$$meas = (meas_{high} << 8) + meas_{low}$$

Так само формується корегувальне значення. Для більш детального аналізу роботи сенсору слід звернутись до документації та опису регістрів зокрема.

Завдання

З погляду *виконання роботи*, надається проєкт із реалізованими функціями ініціалізації та калібрування. На початку його необхідно запустити та перевірити, що функції виконуються (наприклад, за допомогою логування, організованому через інтерфейс UART). Слід пересвідчитись у правильності процесу калібрування, проаналізувавши значення до та після калібрування. Після калібрування вони, в середньому, мають знаходиться достатньо близько довкола 0-го значення. Необхідно реалізувати функціонал електронного компасу з урахуванням наведених способів визначення напрямку.

Лабораторна робота № 9 Операційні системи реального часу

Операційні системи реального часу (Real-Time Operating Systems, RTOS) – системи, що призначені для реалізації роботи пристрою із забезпеченням необхідного часу реакції на певні події, для спрощення виконання розподілу задач у системі на базі мікроконтролерів.

Мета роботи: набуття знань і навичок, пов'язаних із первинним налаштуванням операційної системи реального часу (ОСРЧ) FreeRTOS, а також використання базових функцій цієї операційної системи.

Хід роботи

Для того, щоб підключити ОСРЧ FreeRTOS у проєкт CubeIDE слід скористатись можливостями графічного конфігуратора. Відповідні налаштування знаходяться у секції Middleware (рис. 9.1).



Рис. 9.1 – Налаштування FreeRTOS

За замовчуванням ця опція вимкнена, тому слід обрати, на основі якої версії бібліотеки CMSIS буде додано FreeRTOS у проєкт. Налаштування і навіть часткову генерацію допоміжного коду, що буде використовуватись ОС, можна здійснювати повністю у графічному режимі, що спрощує початок роботи (рис. 9.2).

Мікроконтролери. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

FREERTOS Mode and Configuration						
Mode						
Interface CMSIS_V2 ~						
	Configuration					
	5					
Reset Configuration						
🥺 Mutexes	FreeRTOS Heap Usage					
😔 Tasks and Queues	Timers and Semaphores					
🛛 🛇 Config parameters 🔤 🤇	🥺 Include parameters 🛛 📀 User Constants					
Configure the below parameters :	:					
Q Search (CrtI+F) ③ ③						
~ API						
FreeRTOS API	CMSIS v2					
✓ Versions						
FreeRTOS version	10.0.1					
CMSIS-RTOS version	2.00					
✓ Kernel settings						
USE_PREEMPTION	Enabled					
CPU_CLOCK_HZ	SystemCoreClock					
TICK_RATE_HZ	1000					
MAX_PRIORITIES	56					
MINIMAL_STACK_SIZ	ZE 128 Words					
MAX_TASK_NAME_L	LEN 16					
 Mutexes Tasks and Queues Config parameters Configure the below parameters : Configure the below parameters : Search (CrtI+F) Search (CrtI+F) YereeRTOS API Versions FreeRTOS version CMSIS-RTOS version CMSIS-RTOS version Kernel settings USE_PREEMPTION CPU_CLOCK_HZ TICK_RATE_HZ MAX_PRIORITIES MINIMAL_STACK_SIZ MAX_TASK_NAME_L USE_16, BIT_TICKS 	 FreeRTOS Heap Usage Timers and Semaphores Include parameters User Constants CMSIS v2 CMSIS v2 10.0.1 2.00 Enabled SystemCoreClock 1000 56 ZE 128 Words LEN 16 Dischlad 					

Рис. 9.2 – Налаштування параметрів RTOS

Необхідні файли будуть автоматично підключені до проєкту відповідно до обраних параметрів. Загалом, налаштування розподілені між вкладками:

- Config parameters загальні параметри;
- Include parameters;

- User Constants користувацькі константи;
- Tasks and Queues налаштування задач і черг;
- Timers and Semaphores таймери і семафори;
- Mutexes м'ютекси;
- FreeRTOS Heap Usage.

Основною перевагою RTOS є можливість відійти від традиційного підходу з використанням циклів, який не є зручним за масштабування (додавання нових функцій), до підходу на основі задач (tasks). Кожна запущена задача має отримувати свій квант часу на виконання протягом заданого періоду. При цьому, ОСРЧ, а точніше планувальник, бере на себе відповідальність за розподіл часу між задачами. У цьому випадку ми маємо справу з витісняючою ОСРЧ, яка покладає цю функціональність на планувальник.

Для роботи з використанням ОРСЧ бажано обрати інше джерело тактування, аніж SystemClock, тому рекомендується увімкнути один із таймерів-лічильників загального призначення і задати налаштування Timebase Source (рис. 9.3).



Рис. 9.3 – Вибір базового таймера для роботи RTOS

Завдання

Підключити до проєкту функціональність FreeRTOS. Запустити щонайменше 5 періодичних задач, які постійно мають працювати. Реалізувати обмін даними (масив цілих чисел) між двома задачами за допомогою черги (queue). Продемонструвати роботу із семафором за допомогою 3 задач, з яких 1 задача має блокуватись.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Noviello C. Mastering STM32. 2016. 783 p.

2. Gay W. Beginning STM32. Apress, 2018.

3. Bertolotti I. C., Hu T. Embedded software development: The opensource approach. CRC Press, 2016.

4. Ibrahim D. Arm-based microcontroller multitasking projects using the FreeRTOS multitasking kernel. 2021.

5. Вялов С. Р., Дунай А. В. Порівняльний аналіз систем програмування мікроконтролерів серії STM32 у виробах авіаційного застосування. *Авіаційно-космічна техніка і технологія*. 2023. № 4 (спецвип.2). С. 65-69.

6. Krainyk Y. Embedded systems multimedia framework for Microcontroller devices advances in cyber physical systems. 2023. Vol. 8, No. 1. P. 43-49.

7. BSP for GlobalLogic Starter Kit. URL: https://github.com/GlobalLogicEdu/GL-SK-BSP (Last accessed: 20.08.2024).

8. LSM9DS1 9-axis iNEMO inertial module (IMU): 3D magnetometer, 3D accelerometer, 3D gyroscope with I2C and SPI. URL: https://www.st.com/en/mems-and-sensors/lsm9ds1.html (Last accessed: 20.08.2024).

ДЛЯ НОТАТОК

ДЛЯ НОТАТОК

Навчальне видання

Ярослав Михайлович КРАЙНИК, Єгор Олександрович УХАНЬ

МІКРОКОНТРОЛЕРИ

Методичні рекомендації до виконання лабораторних робіт з дисципліни «Мікроконтролери»

Методичні рекомендації

Випуск 447

Редактор О. Михайлова Комп'ютерна верстка, дизайн обкладинки К. Гросу-Грабарчук Друк С. Волинець. Фальцювально-палітурні роботи О. Мішалкіна.

> Підп. до друку 08.10.2024. Формат 60х84¹/₁₆. Папір офсет. Гарнітура «Times New Roman». Друк ризограф. Ум. друк. арк. 3. Обл.-вид. арк. 1,2. Тираж 50 пр. Зам. № 6874.

Видавець і виготовлювач: ЧНУ ім. Петра Могили. 54003, м. Миколаїв, вул. 68 Десантників, 10. Тел.: 8 (0512) 50–03–32, 8 (0512) 76–55–81, e-mail: rector@chmnu.edu.ua. Свідоцтво суб'єкта видавничої справи ДК № 6124 від 05.04.2020.