Київський національний університет імені Тараса Шевченка

Коломієць І.С., Оберемок Є.А.

ПРОГРАМУВАННЯ ВБУДОВАНИХ СИСТЕМ

Методичний посібник з виконання лабораторних робіт для студентів факультету радіофізики, електроніки та комп'ютерних систем

Київ 2024

УДК 535.51

Програмування вбудованих систем. Методичний посібник з виконання лабораторних робіт для студентів факультету радіофізики, електроніки та комп'ютерних систем / Коломієць І.С., Оберемок Є.А. – Київ: Видавнича лабораторія факультету радіофізики, електроніки та комп'ютерних систем Київського національного університету імені Тараса Шевченка, 2024. – 75 с.

Рецензенти: кандидат фіз.-мат. наук, доцент Київського національного університету імені Тараса Шевченка Загороднюк С.П.
кандидат фіз.-мат. наук, доцент Київського національного університету імені Тараса Шевченка Борецький В.Ф.

Посібник складений відповідно до навчальної програми курсу «Програмування вбудованих систем», що викладається на факультеті радіофізики, електроніки та комп'ютерних систем. У посібнику наведено основні поняття вбудованих систем та запропоновано лабораторні роботи, що дозволяють на практиці отримати навички з програмування вбудованих систем та закріпити знання, отримані у ході прослуховування відповідного курсу. Вбудовані системи в лабораторних роботах представлені тестувальною платою MC13213-SRB (13213-Sensor Reference Board) від компанії Freescale Semiconductor (NXP), що побудована на основі мікроконтролерної збірки MC13213. Збірка MC13213 поєднує в єдиному корпусі 8-бітний мікроконтролер сімейства HCS08 та 2.4 ГГц радіомодем, що працює під керуванням контролеру. Основні елементи збірки MC13213 також розглянуті у посібнику.

Плата MC13213-SRB дозволяє реалізовувати усі типові функції вбудованих систем: програмувати збір, зберігання та обробку даних; обмін цифровими даними з внутрішніми та зовнішніми пристроями через послідовні дротові (SPI, I^2C, SCI) та бездротові (802.15.4/ZigBee) інтерфейси; визначати програмні та апаратні переривання; введення аналогових даних з аналогових датчиків (вбудованих датчика температури та акселерометру); програмувати світлову та звукову сигналізацію за визначеними подіями (з використанням вбудованих світлодіодів та динаміка); реалізовувати інтерактивну взаємодію з користувачем через стандартні засоби введення/виведення даних – кнопки, клавіатури, LCD дисплеї (для тестової плати MC13213-NCB); програмувати події з прив'язкою до заданих часових інтервалів з використанням багатоканальних таймерів; генерувати та захоплювати цифрові та ШІМ сигнали з використанням таймер/ШІМ модулів.

Для програмування плати MC13213-SRB пропонується використовувати інтегроване середовища розробки CodeWarrior та мови програмування С.

В посібнику наведено опис п’яти лабораторних робіт. Опис кожної роботи містить мету, теоретичні відомості з прикладами програм, завдання до роботи, методичні вказівки щодо виконання роботи та контрольні питання для самостійної роботи студентів. У додатках наведено опис основ роботи у середовищі Code Warrior та таблиця векторів переривань збірки MC13213

Ухвалено Вченою радою факультету радіофізики,
електроніки та комп’ютерних систем
Київського національного університету імені Тараса Шевченка
(протокол №10 від 27 березня 2024 р.)

© Видавництво факультету радіофізики,
електроніки та комп’ютерних систем
Київського національного університету
імені Тараса Шевченка, 2024
© Коломієць І.С., Оберемок Є.А., 2024

Вступ

Даний посібник містить навчальний матеріал, призначений для самостійного вивчення студентами, які практикуються з тематики програмування вбудованих систем. Для закріплення розуміння теоретичного матеріалу студентам пропонується виконати п’ять лабораторних робіт на реальних макетах. Описи робіт наведено в посібнику.

Метою посібника є - на практичних прикладах ознайомити студентів з типовими функціями вбудованих систем та сформувати практичні навички з їх програмування.

Під час підготовки до лабораторної роботи студенти повинні вивчити її опис, ознайомитись із лабораторним макетом, з’ясувати призначення елементів стенду та порядок їх увімкнення.

Результатом виконання лабораторної роботи має бути програма, яка при завантаженні у лабораторний макет (тестувальну плату MC13213-SRB) реалізує усі, сформульовані в описі роботи завдання.

За фактом успішного виконання лабораторної роботи передбачається складання звіту, що має містити:

1. Теоретичні відомості за темою роботи.

2. Хід роботи.

4. Код програми з коментарями.

5. Висновки та аналіз особливостей роботи стенду під управлінням створеної програми.

Загальні відомості про вбудовані системи.

Будь-яка механічна система, електрична система, пристрій або об’єкт, що має у своєму складі компонент управління, виконаний на основі обчислювача, називається системою, що вбудовується, або вбудованою системою (Embedded System). Сучасна вбудована система - це поєднання мікроконтролерного або комп'ютерного (мікрокомп’ютерного) обладнання та програмного забезпечення, які призначені для вирішення конкретного, обмеженого кола задач.

Промислові машини, побутова електроніка та техніка, автомобілі, медичне обладнання, літаки, торгові автомати, іграшки, мобільні пристрої і т.ін. - можливі місця для впровадження вбудованих систем.

Центральний елемент вбудованої системи – мікроконтролер, мікрокомп’ютер або комбінована мікроконтролерна збірка (поєднує у єдиному корпусі один, або декілька мікроконтролерів та інших електронних елементів). Він поєднує у собі центральний процесор (ЦП), оперативну та постійну пам'ять, пристрої введення/виведення (ПВВ), блоки зі спеціальними функціями (лічильники, компаратори, генератори, цифро-аналогові/аналого-цифрові перетворювачі, модеми та інші) і міжмодульні шини.

Вбудовані системи можуть бути програмованими або мати фіксовану функціональність. У системах з обмеженою фіксованою функціональністю, (наприклад система керування гіробордом), часто, може бути відсутній інтерфейс користувача, або наявний простий інтерфейс для відображення стану системи. У програмованих системах (наприклад, смартфон) може бути навпаки складний графічний інтерфейс для забезпечення інтерактивної взаємодії системи та користувача. Такі інтерфейси можуть містити кнопки, світлодіоди, джерела звукових сигналів, датчики, порти розширення, порти введення/виведення даних та сенсори різного роду.

Для попереднього налагодження програмного забезпечення вбудованих систем та відпрацювання їх основних функцій, виробники відповідних

компонент для цих систем розробляють спеціальні тестувальні плати, або плати розробника (development board, development kit). Центральним елементом таких плат завжди є програмований мікроконтролер, мікроконтролерна збірка або мікрокомп'ютер, разом з яким на платі розміщують усю необхідну електронну обв'язку (джерела тактових імпульсів, стабілізатори напруги живлення, контакти та розніми для введення виведення даних). Для наближення умов функціонування контролера та відповідного програмного забезпечення до реальних, тобто, як у схемах вбудованих систем, на тестувальних платах, часто, розміщують і типовий набір периферійних елементів – світлодіоди, кнопки, динаміки, дисплеї, цифрові та аналогові датчики, драйвери стандартних інтерфейсів введення/виведення тощо. Таке рішення розширює можливості щодо всебічності планування та тестування відповідного програмного забезпечення та узгодження функціоналу цього програмного забезпечення з вимогами та потребами майбутнього об'єкту керування.

У лабораторних роботах, розглянутих у посібнику, використовується тестувальна плата MC13213-Sensor Reference Board від Freescale Semiconductor, побудована на базі мікроконтролерної збірки (платформи) сімейства MC1321x.

Основні елементи збірки сімейства MC1321x та тестувальної плати MC13213-SRB коротко розглянуто у наступному розділі. Деякі виділені елементи збірки, програмування яких є предметом розгляду окремих лабораторних робіт, розглядаються детальніше у теоретичних відомостях цих робіт.

Мікроконтролерна збірка сімейства MC1321x

Збірка MC1321x - це ZigBee™ платформа, яка включає в себе 8-бітний мікроконтролер сімейства HCS08 та модем з радіопередавачем малої потужності (1 мВт) радіочастотного діапазону 2,4 ГГц стандарту 802.15.4 під його керуванням. Збірку з контролеру та модему виконано у вигляді єдиного 71-контактного LGA чипу розміром 9x9x1 мм (рис.1 та рис.2). LGA (**L**and **G**rid

Array) - контакти у вигляді плаських металевих площадок. Необхідні з'єднання між контролером та модемом зроблені всередині корпусу збірки MC1321x. Деякі сигнальні лінії виведено назовні головним чином для використання в якості тестових точок. Ці сигнали можуть бути корисними при написанні та відлагодженні програмного забезпечення.

Збірки сімейства MC1321x можуть використовуватися для побудови як простих бездротових вбудованих систем із з’єднанням типу точка-точка, так і для побудови повних мережевих ZigBee систем. Поєднання радіомодема і мікроконтролера в єдину збірку невеликого розміру дозволяє отримати економічно ефективне рішення для побудови бездротової системи керування різними об’єктами, у тому числі розподіленими, з малим споживанням.

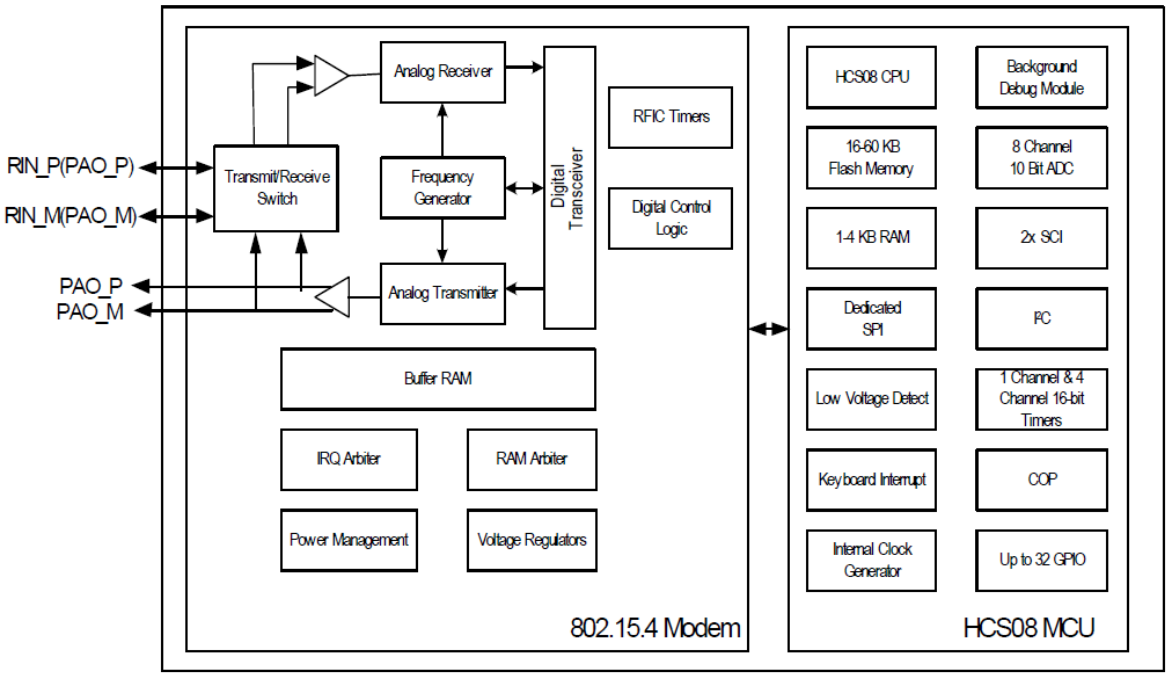


Рис.1. Спрощена блок-схема збірок сімейства MC1321x.

Основна мета поєднання радіомодему та контролеру у одній збірці - побудова на її основі систем управління та мереж з бездротовими комунікаціями. Однак, задача опанування програмування бездротових комунікацій лежить поза межами даного лабораторного курсу, і, тому, обговорення відповідних технічних деталей у даному посібнику не проводиться. Функціонал керуючого мікроконтролера HCS08 збірки повністю

доступний розробнику і дозволяє реалізувати усі необхідні задачі з керування зовнішніми пристроями та взаємодії з периферією без участі модему. Більше того, така периферія значною мірою інтегрована на тестувальній платі MC13213-SRB і використовується у лабораторних роботах для отримання навичок програмування вбудованих систем.

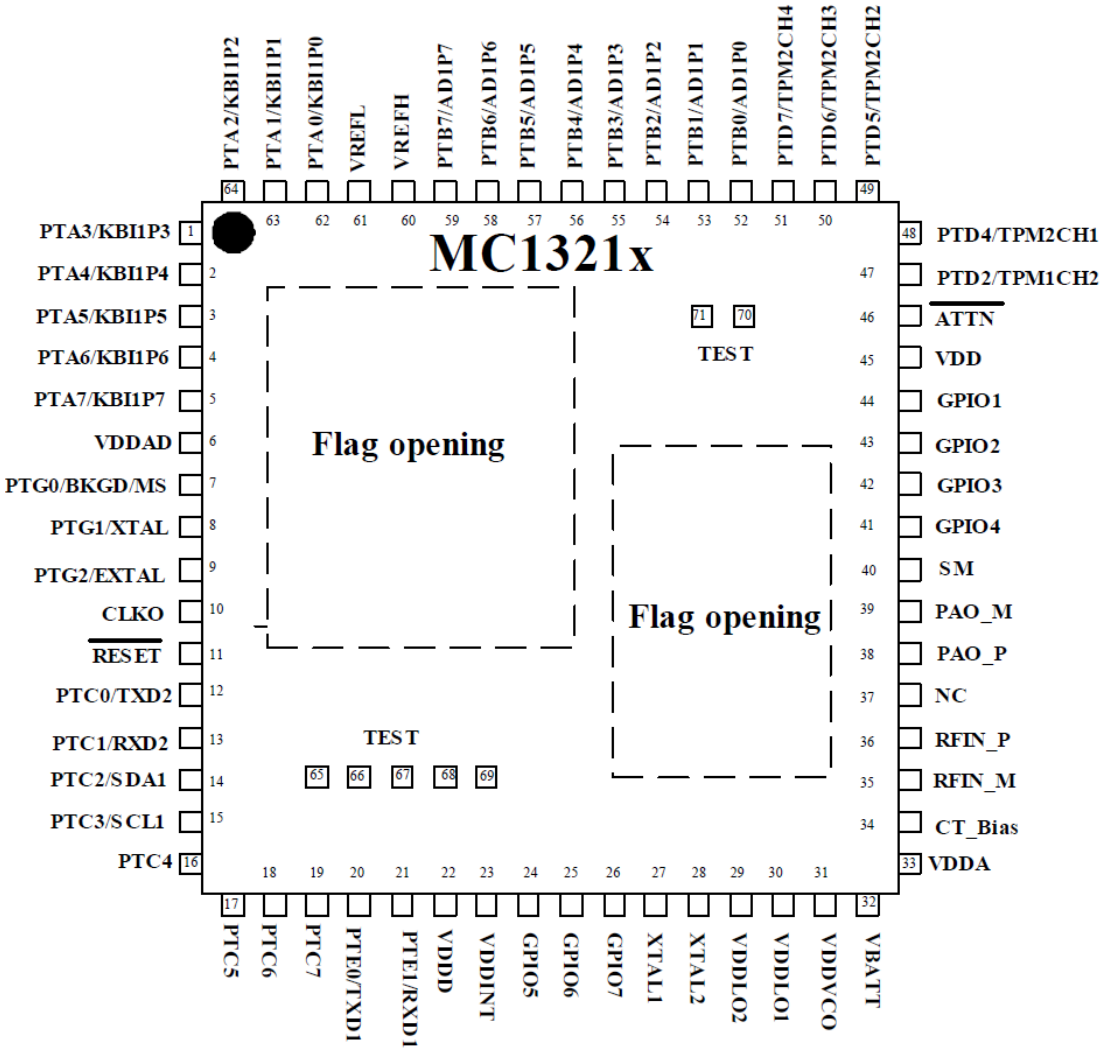


Рис.2. Схема контактів збірок сімейства MC1321x (вигляд зверху).

Збірки сімейства MC1321x відрізняються тільки об’ємом вбудованої пам’яті, а керуючий контролер та периферійні модулі в них однакові.

Більше технічних деталей про будову збірок сімейства MC1321x та про їх відмінності можна знайти у документації [1].

Мікроконтролери сімейства HCS08

Функціональна блок-схема контролерів сімейства HCS08 наведена на рис.3

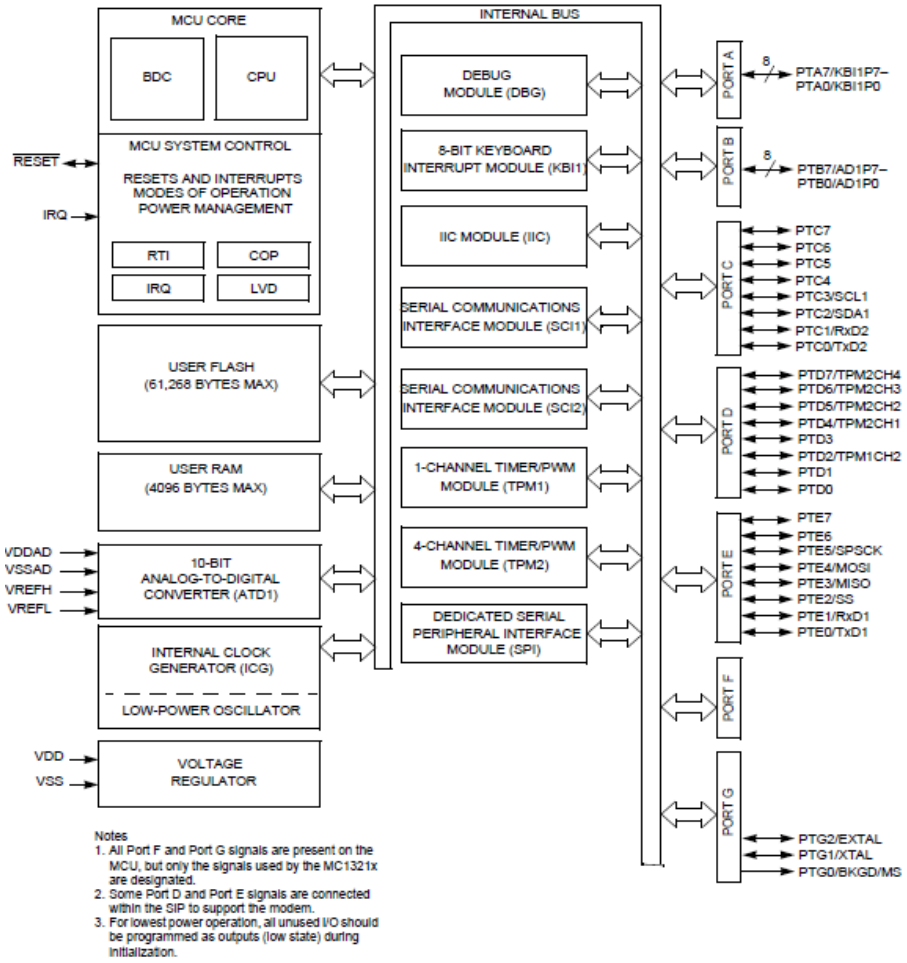


Рис.3. Блок-схема контролера HCS08.

Мікроконтролери сімейства HCS08 - 8-бітні контролери. Максимальна тактова частота системної шини контролера — 20 МГц, а максимальна тактова частота центрального процесора контролера HCS08 становить 40 МГц. Як можна бачити на рис.3, ядро контролера MCU Core (**M**icro **C**ontroller **U**nit Core) складається з центрального процесора CPU (**C**entral **P**rocessing **U**nit) та апаратного фонового контролера налагодження BDC (**B**ackground **D**ebug **C**ontroller), який контролює доступ до ядра контролера HCS08 і, у парі з вбудованим модулем налагодження DBG (**D**e**B**u**G** Module), забезпечує покращену підтримку налагодження та програмування флеш-пам’яті. Модулі

системи контролю RTI (**R**eal-**T**ime Interrupt), COP (**C**omputer **O**perating **P**roperly), LVD (**L**ow-**V**oltage **D**etect), IRQ (**I**nterrupt **R**e**Q**uest) керують скиданням контролера та його взаємодією з джерелами переривань. Зокрема, модуль переривань реального часу RTI може використовуватись для генерування періодичних переривань з програмованою тривалістю періоду, використовуючи внутрішні або зовнішні джерела тактування. Дані переривання корисні для відліку часових інтервалів у програмі або для пробудження контролеру з режиму низького споживання. Слідкуючий таймер COP Watchdog - модуль, що являє собою незалежний таймер-лічильник, призначений для перезавантаження контролеру у тому випадку коли програмний додаток починає працювати не так як очікується. Буквально, watchdog – вартовий пес. COP Watchdog («вартовий пес»-таймер) – один з інструментів, що дозволяє запобігти «зависанню» програми контролера або виконанню нею непередбачуваних дій у випадку випадкового збою. Для уникнення аварійного перезавантаження контролера, програма має періодично скидати лічильник таймеру COP (якщо він задіяний) до того, як даний лічильник сягне певного встановленого максимального значення. За замовчуванням таймер COP після увімкнення та перезавантаження контролера вмикається. COP може бути вимкнений програмно і тоді завдання відслідковування збоїв у роботі програмного забезпечення та їх обробка покладається на програміста. Модуль детектування низької напруги живлення LVD захищає вміст пам’яті контролера при зменшенні напруги живлення нижче встановленого порогового значення та контролює стан системи під час зміни напруги живлення. Даний модуль може бути запрограмований на генерування переривання або скидання контролеру при зменшенні напруги живлення менше порогового значення. Модуль обробки переривань IRQ дозволяє організувати асинхронний режим роботи програмного забезпечення контролера, ігноруючи, або дозволяючи переривання від джерел, під’єднаних до відповідних ліній контролера. При виникненні дозволеного переривання виконання основної програми контролера призупиняється і виконується програмний код підпрограми-обробника переривань, після чого

керування контролером повертається основній програмі. Встановлюючи стан бітів відповідних регістрів центрального процесора та модуля IRQ, програмують дозволи та заборони переривань на встановлених лініях контролера.

Для зберігання програмного коду та даних користувача в контролері доступно 60 кБ енергонезалежної пам'яті USER FLASH та 4 кБ оперативної USER RAM пам'яті.

Контролер HCS08 включає наступні периферійні модулі:

• послідовний периферійний інтерфейс SPI (**S**erial **P**eripheral **I**nterface), через який, всередині збірки MC1321x, контролер під'єднано до модему. Хоча сам інтерфейс SPI контролеру може працювати як у режимі Master так і у режимі Slave, однак, внаслідок його жорсткого з'єднання з інтерфейсом SPI модему, його можна програмувати тільки у режимі Master та застосовувати таке тактування, що підходить до параметрів інтерфейсу SPI модема;

• один триканальний 16-бітовий таймер/широтно імпульсний модулятор (ШІМ) TPM1 (**T**imer/**P**ulse width **M**odulator), та один чотирьох-канальний 16-бітовий таймер/ШІМ - TPM2

• модуль KBI (**K**ey**B**oard **I**nterrupt modul) - 8-бітний порт з підтримкою переривань для підключення пристроїв типу - клавіатура;

• 8-канальний 10-бітовий аналого-цифровий перетворювач ATD (**A**nalog **T**o **D**igital converter);

• два незалежних послідовних інтерфейси передачі даних SCI1 та SCI2 (**S**erial **C**ommunication **I**nterfaces);

• вбудований генератор тактових імпульсів ICG (**I**nternal **C**lock **G**enerator) – підтримує зовнішнє джерело тактових імпульсів або зовнішній кварцовий осцилятор та має внутрішній опорний генератор з двома контрольованими джерелами. Одне з джерел має частоту приблизно 8 МГц, і може бути обране як локальний осцилятор для контролера фонового налагодження BDC. Інше джерело зазвичай має частоту 243 кГц та може бути використане як дешева альтернатива для генератора тактових імпульсів контролера;

• послідовна шина даних для зв'язку інтегральних схем IIC (**I**nter-**I**ntegrated **C**ircuit) із швидкістю 100 кб/с;

• вбудований модуль налагодження у фоновому режимі BDM (**B**ackground **D**ebug **M**odule), що розширює можливості налагодження програмного забезпечення контролера з використанням персонального комп’ютера;

• до 32 ліній введення/виведення загального призначення GPIO (**G**eneral **P**urpose **I**nput/**O**utput) з програмованим резисторами підтягування. Насправді контролер HCS08 у сумі має 7 портів загального призначення, кожен з яких надає по 8 ліній введення/виведення, однак схемотехнічні особливості збірок MC1321x дозволили під’єднати до вихідних пінів(контактів) лише 32 лінії. Деякі з цих ліній поділяються між декількома модулями контролера (див. рис.2 та рис.3). В один і той самий момент кожен контакт збірки, що поділяється між декількома її модулями, може бути використаний як лінія введення/виведення тільки одним модулем.

Більше технічних деталей про будову контролера HCS08, про будову його модулів та про особливості використання контролера у збірці MC13213, також, можна знайти у документації [1].

Тестувальна плата MC13213-Sensor Reference Board

Тестувальна плата MC13213-Sensor Reference Board (далі MC13213-SRB) побудована на основі мікроконтролерної збірки MC13213 рис.4, 5. Периферія, інтегрована на платі: 5 кнопок, 5 світлодіодів, металеві виводи для підключення зовнішніх пристроїв та джерел живлення, датчик температури, акселерометр, динамік, вбудована антена, драйвер емуляції послідовного COM-порту, пам'ять EEPROM (**E**lectrically **E**rasable **P**rogrammable **R**ead-**O**nly **M**emory). На базі даної тестової плати можливо реалізувати всі необхідні типові функції вбудованих систем: програмування операцій, обчислення у реальному часі, збереження та обмін даними, збір цифрових та аналогових даних, у тому числі з використанням механізму переривань, захоплення та генерація цифрових та ШІМ сигналів, тощо.

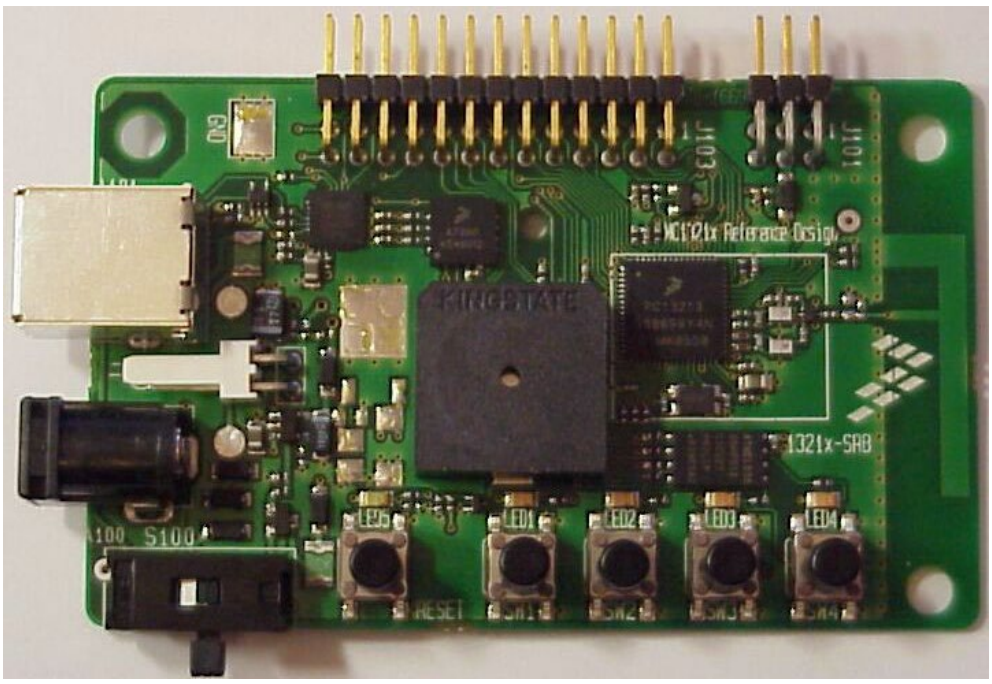


Рис.4. Вигляд тестувальної плати MC13213-SRB (*Sensor Reference Board*).

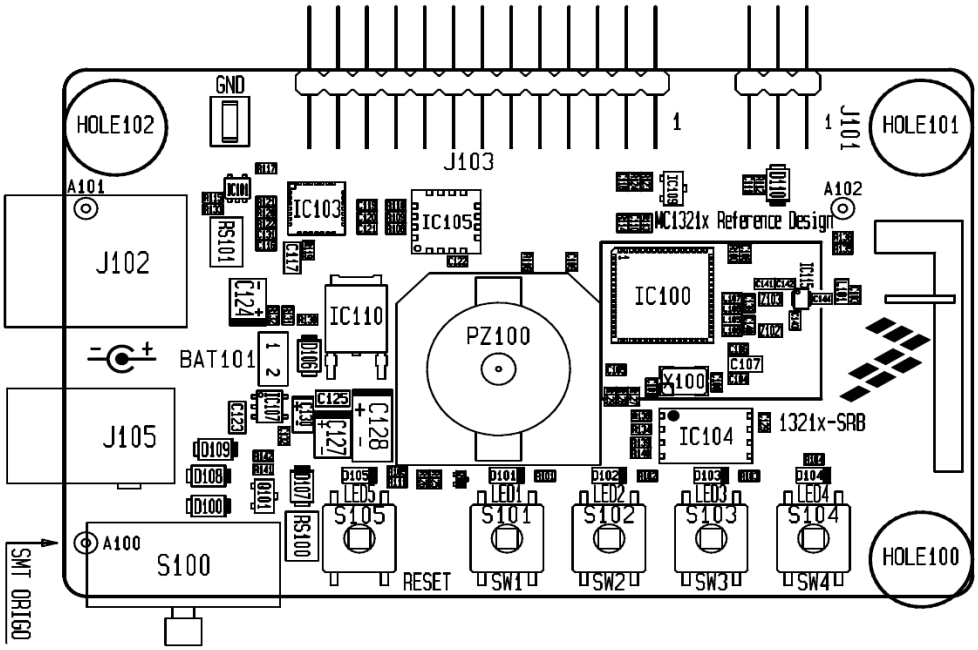


Рис.5. Схема розміщення електронних компонент на тестувальній платі MC13213-SRB.

PZ 100 – динамік; IC 100 - мікросхема MC13213; IC 109 – датчик температури (LM61B); IC 104 - Electrically Erasable Programmable Read-Only Memory (EEPROM); IC 105 – трьохосьовий акселерометр MMA7260Q; S 101-105 – тактові кнопки(TACT); D 101-105 – світлодіоди; S 100 вимикач (ON/OFF).

Лінії введення/виведення збірки MC13213 та їх функціональне призначення на тестовій платі MC13213-SRB наведено у таблиці 1

Таблиця 1. Лінії введення/виведення збірки MC13213 та їх функціональне призначення на тестовій платі MC13213-SRB.

Port A	13213-SRB Functionality	Port B	13213-SRB Functionality
PTA0/KBD0		PTB0/AD0	X-Y-Z Axis Accelerometer (G - Select 1)
PTA1/KBD1		PTB1/AD1	X-Y-Z Axis Accelerometer (G - Select 2)
PTA2/KBD2	(Switch 1)	PTB2/AD2	X-Y-Z Axis Accelerometer (X-OUT)
PTA3/KBD3	(Switch 2)	PTB3/AD3	X-Y-Z Axis Accelerometer (Y-OUT)
PTA4/KBD4	(Switch 3)	PTB4/AD4	X-Y-Z Axis Accelerometer (Z-OUT)
PTA5/KBD5	(Switch 4)	PTB5/AD5	X-Y-Z Axis Accelerometer (Power Save)
PTA6/KBD6	(USB to UART) (CP2102 CTS)	PTB6/AD6	Vout (Temperature Sensor)
PTA7/KBD7	(USB to UART) (CP2102 RTS)	PTB7/AD7	VCC (Temperature Sensor)
Port C	**13213-SRB Functionality**	**Port D**	**13213-SRB Functionality**
PTC0/TxD2	(USB to UART) (CP2102 RXD)	PTD0	ATTNB
PTC1/RxD2	(USB to UART) (CP2102 TXD)	PTD1/RXTXEN	RXTXEN
PTC2/SDA	CS (EEPROM)	PTD2	(Buzzer)
PTC3/SCL	Data (EEPROM)	PTD3/RSTB	RSTB
PTC4	(LED5) Green	PTD4	(LED1) Red
PTC5	CLK (EEPROM)	PTD5	(LED2) Red
PTC6	Data (EEPROM)	PTD6	(LED3) Red
PTC7	(USB Suspend)	PTD7	(LED4) Red
Port E	**13213-SRB Functionality**	**Port G**	**13213-SRB Functionality**
PTE0/TxD1		PTG0/BKGD/MS	(BDM)
PTE1/RxD1		PTG1/XTAL	
PTE2/CEB	CEB	PTG2/EXTAL	(CLKO)
PTE3/MISO	MISO		
PTE4/MOSI	MOSI		
PTE5/SPICLK	SPICLK		
PTE6	GPIO2		
PTE7	GPIO1		

Більше інформації про тестувальну плату 13213-SRB можна знайти у [2].

Обладнання, документація та програмне забезпечення

У лабораторних роботах використовується наступне обладнання, документація та програмне забезпечення:

1. Тестувальна плата 13213-SRB на базі збірки MC13213 компанії Freescale Semiconductor та документація до неї.
2. Програматор PE micro USB BDM Multilink.

3. Інтегроване середовище розробки для вбудованих систем CodeWarrior.

Лабораторна робота № 1. Програмування ліній введення/виведення загального призначення (GPIO)

Мета роботи: Опанувати керування зовнішніми пристроями у вбудованих системах (наприклад, світлові та звукові індикатори) з використанням ліній введення/виведення загального призначення. Набути навичок роботи з інтегрованим середовищем розробки для вбудованих систем CodeWarrior.

Теоретичні відомості

Лінії введення/виведення загального призначення GPIO збірки MC13213

Як зазначено у вступній частині, до частини зовнішніх контактів збірки MC13213 під'єднано лінії введення/виведення загального призначення мікроконтролера HCS08 та лінії введення/виведення внутрішніх модулів контролера (рис.2). Написи типу «PTB0/AD0» біля деяких контактів збірки вказують на те, що до даного контакту під'єднано лінії введення/виведення одночасно декількох модулів контролера, або контролера і модему. Такий спосіб позначення фізичних контактів електронних схем з множинним під'єднанням широко розповсюджений у схемотехніці і у схемотехніці вбудованих систем, зокрема. Кількість під'єднань до одного фізичного контакту мікросхеми може бути істотно більшою за два. Звичайно виходи/входи різних модулів під'єднують до контактів так, щоб модулі не конфліктували один з одним. У більшості випадків в кожен момент часу доступ до спільного контакту надається тільки одному модулю. Виключення, хіба що, можуть складати контакти, до яких під'єднують джерела тактових імпульсів.

До 32 контактів збірки MC13213 можуть підключатись лінії введення/виведення загального призначення контролера. Поняття «загальне призначення», у даному випадку, має на увазі те що, через дані лінії можна вводити/виводити цифрові сигнали від/до широкого кола зовнішніх пристроїв, безвідносно до їх кінцевого функціонального призначення. Завдання

формування та інтерпретації цих сигналів покладається на розробника відповідного програмного забезпечення.

Лінії введення/виведення загального призначення у контролері HCS08 об'єднані в 8-розрядні паралельні двонапрямлені порти: A, B, C, D, E, F та G (рис.3). Виключенням є лінія PTG0 порту G, що працює тільки на виведення. Деякі лінії портів D та E не під’єднано до контактів збірки, а лінії порту F та деякі лінії порту G не використовуються.

Кожен двонапрямлений порт введення/виведення загального призначення керується мінімум чотирма регістрами спеціальних функцій (незалежно по кожній лінії порту), які в спеціалізованих бібліотеках збірок сімейства MC1321x позначаються як:

PTxDD – регістр напрямку передачі (**P**or**t x D**ata **D**irection);

PTxD – регістр даних порту (**P**or**t x D**ata);

PTxPE – регістр керування підтягуванням (**P**or**t x P**ull up **E**nable)

PTxSE – регістр контролю крутизни росту напруги на лінії, при перемиканні (**P**or**t x S**lew rate control **E**nable).

У позначеннях вище, замість літери «x» слід підставити ім’я відповідного порту – A, B, C і т.д.

Встановлення в деяких бітах регістру PTxDD порту *x* логічних «1», налаштує відповідні лінії порту на вивід. У той же час логічні «0» у інших бітах даного регістру налаштують відповідні лінії порту на введення.

Якщо деякі лінії порту *x* налаштовані на виведення, то запис до його регістру даних PTxD 8-бітного слова даних, призведе до встановлення на цих лініях порту високих та низьких рівнів напруги, згідно логічних значень «0» або «1» у відповідних розрядах записаного слова, при цьому стани ліній, налаштовані на введення, не будуть змінені.

В режимі введення, зчитування стану порту *x* у 8-бітну змінну сформує значення її двійкових розрядів відповідно рівнів на вхідних лініях порту.

Гарною практикою програмування є запис даних у регістр даних порту перед тим, як змінити режим передачі порту на виведення. Це гарантує, що на

виході порту ми в жодному разі не зустрінемо старі значення даних, які випадково опинилися в регістрі даних порту.

Регістр керування підтягуванням PTxPE дозволяє вмикати та вимикати резистори підтягування (R pullup) на лініях порту *x*. Резистори підтягування використовуються для підтягування рівня лінії порту до значення, що відповідатиме логічній «1» або логічному «0». Подібне підтягування дозволяє зменшити шуми в системі контролера коли лінію нікуди не під'єднано, або, наприклад, коли до лінії під'єднано кнопку, яка у даний момент розімкнена.

Після перезавантаження збірки MC13213 всі двонапрямлені лінії введення/виведення налаштовуються на введення з вимкненими резисторами підтягування.

У прикладі 1, з використанням синтаксису мови C, запрограмовано наступні дії над лініями введення/виведення загального призначення:

1. порт A налаштовується на виведення та використовується для паралельного виведення 8-бітного слова даних 0b01110101.
2. лінії C7-C5 порту C налаштовані на виведення, лінії C4-C0 - налаштовані на введення, на лініях C4-C2 додатково задіяне підтягування до високого рівня, а на лініях C1-C0 - підтягування до низького рівня.
3. 8-бітні дані зчитуються з порту C у змінну C_byte та з них виділяється частина біт, що відповідає лініям C4-C0 порту, налаштованим на введення.

Приклад 1.

```
PTAD = 0b01110101;  // заносимо дані у регістр даних порту A
PTADD = 0b11111111; // налаштовуємо порт A на виведення, після
                    // чого на виходах порту виставляються рівні,
                    // що відповідають даним в регістрі PTAD.
PTCDD = 0b11100000; // налаштовуємо лінії C7-C5 порту C
                    // на виведення, лінії C4-C0 - на введення
PTCPE = 0b00011100; // лінії C4-C2 порту C підтягуємо до
                    // високого рівня, а лінії C1-C0 - до низького.
C_byte = PTCD;    // зчитуємо байт даних з порту C у змінну C_byte
C4_0data = C_byte & 0b00011111; //виділяємо актуальні вхідні дані.
```

Базова бібліотека для вбудованих систем на базі збірок MC1321x містить макроси для забезпечення роботи з кожним розрядом регістрів портів окремо. Наприклад, звернення (зчитування/запис) до розряду *n* регістру даних порту *x* виглядатиме так: PTxD_PTxDn = 0(1).

Приклад 2.

```
PTDD_PTDD4 = 1;   // занесення логічної «1» у розряд D4 регістру
                  // даних порту D
PTDDD_PTDDD4 = 1; // лінія D4 порту D налаштовується на виведення,
                  // отже на ній виставляється високий рівень
                  // напруги, що відповідає логічній «1»
                  // у відповідному розряді регістру даних порту D.
```

Елементи базової програми на мові С для збірок сімейства MC1321x

Базова програма на мові С для з збірок сімейства MC1321x при використанні середовища розробки CodeWarrior (див. додаток 1) має вигляд:

Приклад 3.

```
#include <hidef.h> /* підключаємо файл-заголовок з визначеннями макросів для */
                   /* роботи з перериваннями */
#include "derivative.h" /* декларування периферійних модулів */

/* визначення макросів для покращення розуміння коду програми */
#define OFF 1
#define ON 0
#define LED_2 PTDD_PTDD5

/* Опис процедур і функцій*/
  void SomeProc(int somevar)
  {
     /* код процедури/функції*/
  }

/* Точка входу в програму */
 void main(void) {
     EnableInterrupts; /* дозвіл переривань */

/* Код, який буде виконано один раз при запуску контролера */
     PTDD = 0xFF;
     PTDDD = 0xFF;
     LED_2 = ON;
     int i = 0;

/*Початок основного безперервного циклу контролера */
    for(;;) {
/* Код основного циклу функціонування контролера */
```

```
        i++;
        if (i == 1000) {
            i = 0; LED_2 = !LED_2;
        }

         SomeProc(100);

        __RESET_WATCHDOG(); /* скидання лічильника таймера COP Watchdog */
    } /* кінець основного безперервного циклу for(;;) */
}/*кінець main ( ) */
```

Як зазначено у загальній частині, за замовчуванням, після вмикання та перезавантаження контролеру, таймер COP Watchdog вмикається. Керування налаштуваннями таймеру COP здійснюється через 8-бітний регістр SOPT (System Options Register). Структура даного регістру наведена на рисунку 6 Призначення відповідних полів (розрядів) даного регістру описано у таблиці 1.

Bit	7	6	5	4	3	2	1	0
R W	COPE	COPT	STOPE		0	0	BKGDPE	
RESET	1	1	0	1	0	0	1	1

Рис.6 Структура системного регістру SOPT.

Таблиця 2. Опис полів регістру SOPT.

Поле	**Опис**
7 COPE	Біт вмикання/вимикання COP Watchdog таймеру (**COP Watchdog Enable**). Встановлюється один раз при завантаженні контролера. За замовчуванням - 1 Можливі значення: 0 – таймер вимкнено; 1 – таймер увімкнено (контролер буде перезавантажуватись кожного разу, коли лічильних таймеру сягне встановленої межі).
6 COPT	Біт встановлення межі лічильника таймеру COP (**COP Watchdog Timeout**). Встановлюється один раз при завантаженні контролера. За замовчуванням - 1 Можливі значення: 0 – короткий часовий період (максимальне значення лічильника - 2^{13} тактів системної шини); 1 – довгий часовий період (максимальне значення лічильника - 2^{18} тактів системної шини).
5 STOPE	Біт вимикає режим зупинки (**Stop Mode Enable**). Встановлюється один раз при завантаженні контролера. За замовчуванням – 0. Якщо режим зупинки вимкнено, а програма користувача намагається виконати інструкцію STOP, відбувається примусове генерування виключення неприпустимого коду операції.

	Можливі значення: 0 – режим зупинки заборонено; 1 – режим зупинки дозволено.
1 BKGDPE	Біт дозволу фонового відлагодження програми (**Background Debug Mode Pin Enable**). Цей біт дозволяє лінії PTG0/BKGD/MS функціонувати як BKGD/MS. Коли біт не встановлений (0), відповідна лінія функціонуватиме як PTG0 – лінія введення загального призначення. Після будь-якого скидання цей контакт завжди використовує функцію BKGD/MS. Можливі значення: 0 – біт не дозволено; 1 – біт дозволено.

Скидання лічильника таймеру COP відбувається шляхом запису будь-якого значення у системний 8-бітний регістр SRS (System Reset Status Register), при цьому вміст регістру не змінюється. Саме це й робить метод __RESET_WATCHDOG() у прикладі 3.

Завдання

1. Вивчити опис мікросхеми MC 13213 та тестувальної плати 13213-SRB. Ознайомитись з підключеними до входів/виходів загального призначення мікросхеми MC 13213 пристроями тестувальної плати (таблиця 1).
2. Встановити (за потреби) на персональний комп’ютер інтегроване середовище розробки CodeWarrior та опанувати основні елементи роботи з ним, відповідно до додатку 1.
3. Встановити драйвери для програматора PE-micro USB BDM Multilink
4. Запрограмувати мікросхему MC 13213 у відповідності до завдань нижче:
   - увімкнути всі червоні діоди LED1-LED4 та звуковий індикатор PZ100 (Buzzer) (див. рис.5 та табл.1);
   - увімкнути зелений діод LED5 та короткочасно звуковий індикатор PZ100 (Buzzer);
5. Завдання на додаткові бали:

- увімкнути червоні діоди LED2 та LED4, при цьому зелений діод LED5 та червоний LED1 мають блимати з різною довільною частотою, а Buzzer вмикатись/вимикатись синхронно з блиманням діоду LED5.
- закоментуйте рядок з викликом __RESET_WATCHDOG() та поспостерігайте за роботою тестової плати при цьому. Поясніть результат спостережень.

Порядок виконання роботи та методичні вказівки

1. Створити порожній проект у середовищі CodeWarrior у відповідності до інструкції у додатку 1.
2. Написати програму для виконання завдання 4.1.
3. Підключити програматор до порту BDM тестувальної плати.
4. Підключіть програматор до USB порту комп'ютера.
5. Підключіть живлення до тестувальної плати та увімкніть її за допомогою відповідного перемикача (далі до закінчення роботи даний перемикач можна використовувати для вимкнення/вмикання тестової плати).
6. Послідовно створіть та налагодьте програми, що реалізують дії, описані у пунктах завдання 4.
7. Підготуйте та здайте звіт.

Контрольні питання

1. Які порти введення/виведення має мікросхема MC 13213, які налаштування для них доступні?
2. Яка периферія інтегрована на тестувальній платі 13213-SRB? До яких портів і до яких пінів мікросхеми MC 13213 вона під'єднана? Які елементи є джерелом даних(сигналів), які – приймачем?
3. Що робить метод __RESET_WATCHDOG() у прикладі 3?
4. Для чого призначені резистори підтягування (Pullup)?

Лабораторна робота № 2. Програмування дій за натисканням кнопок

Мета роботи: Опанувати програмну взаємодію вбудованих систем з зовнішніми джерелами дискретних сигналів на прикладі кнопок. Набути навичок роботи з регістром резисторів підтягування портів (Pullup).

Теоретичні відомості

На тестувальній платі 13213-SRB інтегровано набір нормально-роз'єднаних тактових кнопок TACT DTSM-644 (на схемі рис.5 позначені як S100-105 та Reset, SW1 - SW4). Кнопку S100 (Reset) підключено в ланцюг скидання збірки MC13213 і її натискання дозволяє апаратно перезавантажити збірку. Кнопки S101-104 на платі під'єдано таким чином, що, при натисканні, вони замикають контакти 64 та 1-3 збірки MC13213 на загальну шину («землю» схеми). На схемі рис.2 видно, що контакти 64 та 1-3 збірки MC13213 можуть використовуватись як виділені лінії введення/виведення загального призначення A2-A5 порту A, відповідно. Таким чином, через дані лінії, шляхом натискання кнопок, можна реалізувати взаємодію користувача з програмою контролера. Очевидно, що кожна кнопка, у даному випадку, є джерелом вхідного сигналу, тому, для адекватної взаємодії з нею, як мінімум, необхідно відповідну лінію порту A налаштувати на введення PTADD_PTADDn = 0. Також, необхідно увімкнути для неї підтягування до високого рівня (Pullup) PTAPE_PTAPEn = 1. Тоді, при розімкненій кнопці, зчитування даних з відповідної лінії порту поверне PTAD_PTADn = 1. При натиснутій (замкненій) кнопці дана лінія порту з'єднується із загальною шиною і зчитування даних за нею поверне PTAD_PTADn = 0. Схемотехніка мікросхеми автоматично обмежить струм на даній вхідній лінії і не дасть спровокувати коротке замикання.

Одним з проблемних явищ, яке спостерігається при роботі з кнопками, є так-званий «брязкіт» контактів. Явище проявляється у тому, що при натисканні

кнопки, в момент наближення робочих контактів кнопки один до одного на відстань, достатню для електричного пробою, може відбуватись пробій, якого достатньо для того, щоб контролер зафіксував «натискання» кнопки. Таких пробоїв до моменту фактичного замикання контактів кнопки може бути достатньо багато і швидкодії контролерів вистачає щоб окремо зафіксувати значну їх кількість. Крім того робочі контакти кнопки можуть бути не достатньо надійними (внаслідок зношеності), що також провокує зайві хибні «натискання/розмикання». Таким чином, часто, коли нам здається, що ми натиснули на кнопку лише один раз, програма контролеру може зафіксувати вже багато натискань/розмикань. Ігнорування ефекту «брязкоту», часом, призводить до того, що вбудована система, на перший погляд, не завжди адекватно реагуватиме на натискання кнопок. І, хоча, при проектуванні контролерів застосовуються схемні рішення, що зменшують вплив ефекту «брязкоту» на роботу програм, незайвим буде його врахувати і у самій програмі. Зокрема, наприклад, заборонити опитування стану кнопки протягом декількох циклів роботи контролера після фіксації факту її натискання у програмі.

Завдання

1. Ознайомитись з підключенням кнопок до ліній введення/виведення загального призначення збірки MC 13213 на тестувальній платі (таблиця 1).
2. Ознайомитись з правилами програмування ліній введення/виведення, до яких підключено кнопки типу TACT.
3. Запрограмувати контролер збірки MC 13213 у відповідності до завдань нижче:
   - запрограмувати по натисканню кнопки SW1 вмикання та неперервне світіння діоду LED1. По натисканню кнопки SW2 вимикання діоду LED1. По натисканню кнопки SW3 - блимання діоду LED1 з довільною частотою;

- запрограмувати щоб при кожному наступному натисканню кнопки SW1 світлодіод LED1 циклічно змінював стан у наступній послідовності: увімкнений, блимає, вимкнений;
- запрограмувати дії: кнопка SW4 натиснута – звуковий індикатор (Buzzer) звучить; кнопка SW4 відтиснута - звуковий індикатор (Buzzer) вимкнено.

4. Завдання на додаткові бали:
   - за допомогою світлодіодів на тестовій платі необхідно дослідити вплив резисторів підтягування на результат натискання кнопок.

Порядок виконання роботи та методичні вказівки

1. Створити порожній проект у середовищі CodeWarrior у відповідності до інструкції у додатку 1.
2. Написати програму для виконання завдання 3.1.
3. Підключити програматор до порту BDM тестувальної плати.
4. Підключіть програматор до USB порту комп'ютера.
5. Підключіть живлення до тестувальної плати та увімкніть її за допомогою відповідного перемикача (далі до закінчення роботи даний перемикач можна використовувати для вимкнення/вмикання тестової плати).
6. Послідовно створіть та налагодьте програми, що реалізують дії, описані у пунктах завдання 3.
7. Підготуйте та здайте звіт.

Контрольні питання

1. Які порти введення/виведення збірки MC 13213 використовуються для підключення кнопок інтегрованих на тестувальній платі?
2. Як мають бути налаштовані регістри порту, до якого підключено тактові кнопки?

3. Що таке «брязкіт» контактів? Де і як він може проявитись? Як з ним боротись?
4. Навіщо в даній роботі вмикати резистори підтягування лінії введення до високого рівня? Що буде, якщо їх не вмикати?

Лабораторна робота № 3. Модуль переривань клавіатури KBI

Мета роботи: на прикладі використання модуля переривань клавіатури KBI збірки MC13213 опанувати асинхронну взаємодію вбудованих систем із зовнішніми пристроями через механізм переривань.

Теоретичні відомості

Переривання

Однією з важливих функцій вбудованих систем є відслідковування та зміна стану внутрішніх периферійних модулів та зовнішніх пристроїв. Для забезпечення такої взаємодії можна у керуючій програмі передбачити циклічне опитування та встановлення відповідних ліній введення/виведення. Проте, часто, різні модулі та пристрої мають і різну швидкодію та потребують різної періодичності опитування/зміни стану, що значно ускладнює послідовне (синхронне) програмування обміну даними. Для спрощення такого обміну у більшості процесорів та мікроконтролерів реалізовано механізм «запиту на переривання» IRQ (**I**nterrupt **R**e**Q**uest), який дозволяє «відволікати» центральний процесор для здійснення обміну даними або обробки даних від периферійних пристроїв тільки тоді, коли у цьому виникла потреба. Наприклад: дані від периферійного пристрою щойно з'явились і можуть бути втрачені, якщо їх зараз не обробити; настав час опитати пристрій; необхідно відреагувати на натискання кнопки; лічильник таймеру COP watchdog переповнено, отже необхідно перезавантажити систему тощо.

Переривання бувають програмні та апаратні. Програмні запити на переривання, наприклад, генеруються при викликах процедур і функцій у тілі основної програми контролера. Апаратні запити на переривання у контролерах здійснюються через модулі переривань IRQ (Interrupt Request module), шляхом встановлення на його вхідних лініях заданих рівнів (прапорів). Джерелами переривань можуть бути як внутрішні модулі мікроконтролера (таймер, модуль АЦП тощо), так і зовнішні пристрої (модеми, датчики, інші контролери тощо).

Зазвичай для обробки ситуацій, які викликають переривання, створюються спеціальні підпрограми – обробники переривань. Оскільки час виникнення того, чи іншого апаратного переривання, часто, неможливо передбачити, важливо щоб програма – обробник переривань була негроміздкою та містила мінімальну кількість операцій. Виконання даної умови зменшить імовірність пропустити обробку наступного запиту на переривання та не створюватиме передумови для надмірного накопичення черги з переривань.

Підпрограма-обробник переривання викликається якщо дотримано наступні умови:

- глобальна маска переривання I (Interrupt mask в регістрі ознак процесора) скинута в «0», тобто переривання центрального процесора дозволено (у прикладі 3 до л.р. 1 скидання маски переривання робить процедура EnableInterrupts з однієї із бібліотек збірки MC13213);
- переривання по лінії модуля переривань, до якої підключений периферійний пристрій, що виставив запит на переривання, дозволено.

На рис.7 наведено спрощену схему підсистеми переривань мікроконтролера HCS08. Кожне джерело переривання (буд-який периферійний модуль) у своєму регістрі управління має біт дозволу переривання та прапор запиту на переривання. Ці біти складаються за логікою AND. На вихідному елементі формується логічна «1» тільки в тому випадку, якщо виставлено запит на переривання і це переривання дозволено. Кожній вхідній лінії модуля переривань відповідає свій елемент AND. Усі виходи елементів AND з’єднані із входами елемента OR, на виході якого формується логічна «1», якщо в поточний момент часу існує хоча б один дозволений запит на переривання. Сигнал з виходу елемента OR логічно помножується з інверсією глобальної маски переривання I і подається в центральний процесор. Зрозуміло, що сигнал запиту на переривання до центрального процесору буде сформований, якщо: 1)

глобальна маска переривання I скинута в «0», 2) встановлено прапор запиту переривання з будь-якого джерела і 3) переривання з цього джерела дозволено.

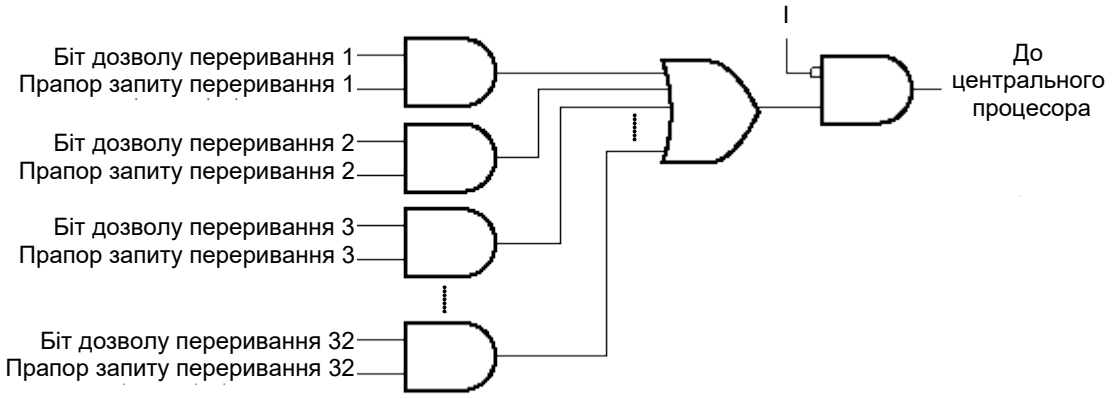


Рис.7. Спрощена структура системи переривань мікроконтролераю.

Якщо кілька пристроїв одночасно виставили запити на переривання (на виході кількох елементів AND встановлено «1»), то підпрограми переривання викликаються мікроконтролером відповідно до їх пріоритетів. Спочатку виконується підпрограма для переривання з найвищим пріоритетом. Потім викликаються підпрограми з меншими пріоритетами у порядку пріоритетності.

Для зручності всі джерела запитів на переривання, пронумеровані. Також, у пам’яті контролера спеціально виділено фіксовану область для збереження адрес підпрограм-обробників переривань (див. табл.1 у додатку 2). Так, наприклад, для зберігання адреси підпрограми-обробника переривання з номером 25 у пам’яті контролера виділена пам’ять, починаючи з адреси $FFCC. По замовчанню, джерелом 25 переривання є таймер реального часу RTI контролера збірки MC13213.

Номер переривання та визначена адреса у пам’яті контролера, за якою зберігатиметься адреса підпрограми-обробника даного переривання, формує вектор переривань. Номер вектору переривання відповідає номеру самого переривання.

У мікроконтролерах HCS08 кожному запиту на переривання (усього може бути до 32 джерел переривань) присвоєно визначений пріоритет, який не можна змінити програмно (це система з жорстким розподілом пріоритетів). Найвищий пріоритет, присвоєний вектору скидання RESET (вектор 0), вектор з номером

31 має найнижчий пріоритет. Таким чином, першим обслуговується вектор переривання з найменшим номером, а потім переривання з високими номерами. Але, в будь-якому випадку, будуть виконані підпрограми для всіх запитів на переривання, які, на момент настання черги для обробки, все ще присутні та дозволені.

Отже, якщо на якійсь лінії модуля переривань IRQ виникає запит на переривання, відповідний прапорець запиту на переривання (рис.7), що має статус лише для читання, буде встановлено модулем в «1». Центральний процесор не відповідатиме, доки відповідний біт дозволу, також, не буде встановлено в «1». Як зазначалось раніше, для того щоб дозволити переривання глобально, біт I у регістрі CCR (**C**ondition **C**ode **R**egister) ознак процесору контролера має бути встановлений в «0». Зауважимо, що після скидання мікроконтролера, глобальна маска переривання (I біт) у CCR спочатку встановлюється в «1», маскуючи (забороняючи) глобальні переривання.

Коли центральний процесор отримує «дозволений» запит на переривання, він виконує поточну інструкцію, перш ніж відповісти на переривання. Послідовність дій при обробці апаратного переривання така ж як і при обробці програмного переривання:

•збереження регістрів процесора в стеку;

•встановлення біта I в CCR для маскування подальших переривань

•отримання вектору переривання з найвищим пріоритетом у черзі переривань;

•виконання програми-обробника переривання.

Поки центральний процесор обробляє поточне переривання, біт I автоматично встановлюється, щоб уникнути можливості запуску обробника іншого переривання, до завершення обробки поточного (це називається вкладенням переривань). Зазвичай біт I відновлюється до «0» після закінчення роботи обробника.

Процедура обслуговування переривань завершується інструкцією повернення з переривання RTI (**R**e**T**urn from **I**nterrupt), яка відновлює регістри

CCR, A, X і PC процесору до їхніх значень до переривання, шляхом зчитування попередньо збереженої інформації зі стеку.

Робота вбудованих систем з використанням механізму переривань забезпечує асинхронну взаємодію у ній пристроїв з центральним процесором контролера та між собою. Асинхронність полягає у тому, що в основній програмі, яку виконує процесор, немає постійної перевірки тієї чи іншої лінії на наявність даних, натомість окремий незалежний модуль, отримавши запит на переривання від деякого пристрою, в будь-який момент може ініціювати зупинку виконання процесором основної програми та його перемикання на обробку термінових даних від пристрою, з наступним поверненням до виконання основного циклу програми.

Модуль сканування переривань клавіатури KBI (Keyboard Interrupt)

Використання кнопок та клавіатур - один із розповсюджених способів для організації взаємодії людини з вбудованими системами. З огляду на швидкодію сучасник мікропроцесорних систем, введення до них даних за допомогою кнопок – дуже повільний процес. Дана особливість дозволила реалізувати взаємодію контролерів та процесорів з клавіатурами з використанням лише однієї лінії переривання, спільної для усіх кнопок. При цьому, навіть якщо кожній кнопці виділена окрема лінія введення, яка саме кнопка викликала переривання легко встановити, просканувавши усі лінії, до яких підключено кнопки. Причому між послідовними актами натискання кнопок, як би швидко не відбувались натискання, мікропроцесорні системи можуть здійснити таке сканування тисячі, мільйони разів.

В деяких мікроконтролерах є спеціалізовані модулі, які є оптимізованими саме для підключення кнопок та клавіатур. Ці модулі мають багато ліній введення але одну на усіх них лінію запиту на переривання. Контролер HCS08 також містить подібний вбудований модуль KBI (Keyboard Interrupt), назва якого говорить сама за себе. KBI – модуль сканування переривання від клавіатури. Його 8 вхідних ліній найчастіше використовуються для

підключення клавіатури, але, також, можуть використовуватися для підключення інших зовнішніх пристроїв, взаємодія з якими потребує використання механізму переривань. Також даний модуль може використовуватись для виведення контролера із режимів низького енергоспоживання (пробудження).

Кожна з 8 вхідних ліній модуля KBI може бути джерелом переривання але усі вони керують лише одним прапором тригера запиту на переривання KBF (**K**ey**B**oard **F**lag) і для них виділено лише один загальний вектор переривання (див. табл.1 додаток 2).

Функціональна схема модуля KBI представлена рис.8.

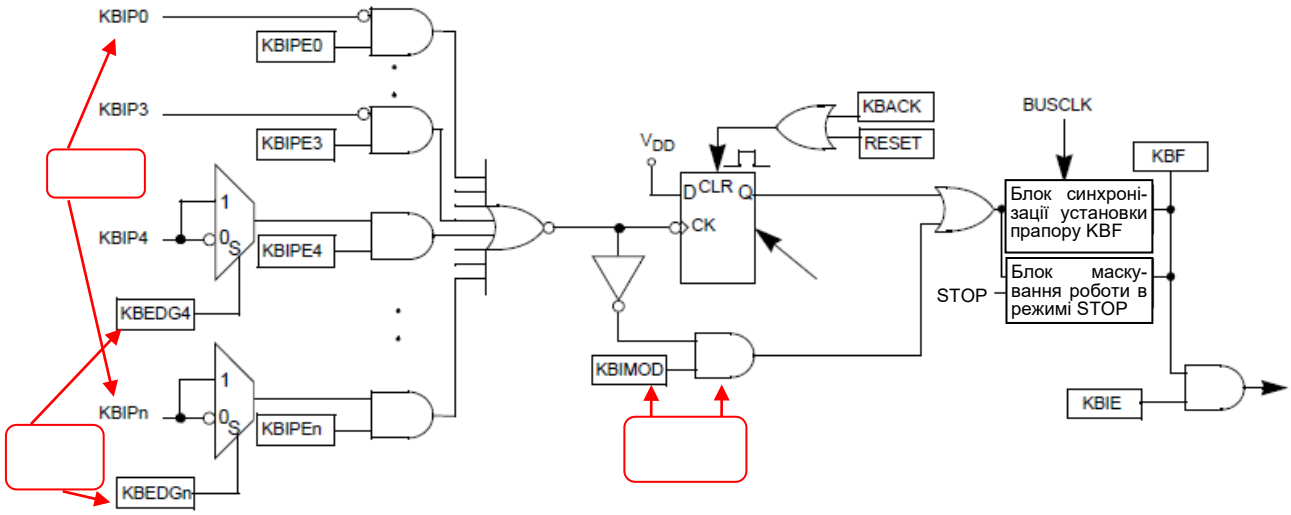


Рис.8. Функціональна схема модуля KBI.

Детектування «активного» сигналу на будь-якому вході KBIPn модуля викликає встановлення прапору запиту на переривання KBF. Конкретний вигляд «активного» сигналу, за яким буде згенеровано запит на переривання, може бути налаштовано для кожної вхідної лінії модуля окремо. Як «активний» можна налаштувати висхідний або спадаючий фронт вхідного сигналу, або ж його рівень (високий/низький). Для цього відповідним чином встановлюють значення бітових полів KBIMOD та KBIEDGn регістру KBISC (**S**tatus and **C**ontrol Register) модуля KBI.

Як можна бачити на рис.2, вхідні лінії модуля KBI та лінії введення/виведення загального призначення, що об’єднані в порт A, поділяють

між собою піни 1-5, 62-64 збірки MC 13213. При цьому, за замовчуванням, дані піни налаштовуються саме як лінії введення/виведення порту A. Для налаштування піну як вхідної лінії модуля KBI необхідно у відповідному розряді регістру KBIPE (**Keyboard Pin Enable for KBI Port Bits**) встановити «1» (KBIPEn = 1).

На чотирьох вхідних лініях модуля KBI (KBIP[7:4]) мікроконтролера HCS08 можна задавати, щоб переривання генерувалось по фронтах вхідного сигналу (висхідному/спадному) або по фронтах і відповідних рівнях вхідного сигналу (високому/низькому). Інші чотири лінії модуля KBIP[3:0] можуть бути налаштовані на генерування переривання тільки за фронтами вхідного сигналу (див. рис.9).

Bit	7	6	5	4	3	2	1	0
R	KBEDG7	KBEDG6	KBEDG5	KBEDG4	KBF	0	KBIE	KBIMOD
W						KBACK		
RESET	0	0	0	0	0	0	0	0

Рис.9. Регістр стану та контролю KBI (KBISC Status and Control Register).

Нижче представлено опис полів регістру стану та контролю KBISC модуля KBI.

Таблиця 3. Опис полів регістру KBISC.

Поле	**Опис**
7:4 KBEDG[7:4]	Біти вибору режиму роботи входів 7-4 (KBIP[7:4]) модуля KBI (**Keyboard Edge Select for KBI Port Bits**). Можливі значення: 0 – переривання за спадаючим фронтом / низьким рівнем; 1 – переривання за висхідним фронтом / високим рівнем. Налаштування працює, якщо генерування переривання входом n дозволене (KBIPEn = 1). Біт керування KBIMOD задає режими роботи модуля з генеруванням переривання за фронтом та рівнем або тільки за фронтом вхідного сигналу на лінії.
3 KBF	Біт прапору переривань клавіатури (**Keyboard Interrupt Flag**) лише для читання. Встановлюється апаратно кожного разу, коли «активний» сигнал (фронт/рівень) виявлено на будь-якому з «дозволених» входів KBI модуля. Прапор скидається шляхом запису «1» в біт керування KBACK. Прапор залишатиметься встановленим, якщо KBIMOD = 1 і на будь-якому дозволеному вході KBI залишається, налаштований через KBEDG, рівень

	сигналу. KBF може бути використаний для програмного визначення факту переривання за входом, навіть якщо переривання від модуля не дозволене (прапор (KBIE = 0)) і відповідне апаратне переривання у процесорі викликатись не буде. Якщо ж переривання від модуля дозволене (KBIE = 1) разом із встановленням прапору KBF буде генеруватись переривання модулем. Можливі значення: 0 – немає необроблених переривань KBI у черзі; 1 – є необроблене переривання від KBI.
2 KBACK	Біт підтвердження оброблення переривання від модуля KBI (**Keyboard Interrupt Acknowledge**). Цей біт для запису (читання завжди повертає 0) використовується для скидання прапору статусу KBF шляхом запису «1» до KBACK. Коли KBIMOD = 1 (переривання за фронтом та рівнем встановлено) і сигнал на дозволеному вході залишається на заданому рівні, KBF постійно встановлюється, так що запис «1» до KBACK не очищатиме прапор KBF доти, поки вхідний рівень не зміниться.
1 KBIE	Біт дозволу переривань від модуля (**Keyboard Interrupt Enable.** Біт управління для читання/запису. Визначає, чи буде модуль KBI виставляти запити на переривання до процесору, коли прапор статусу KBF встановлюється в «1». Якщо KBIE = 0, апаратні переривання не генеруються, але опитування KBF все ще може використовуватися для реєстрації запиту на переривання на входах модуля KBI програмним чином. Можливі значення: 0 - KBI не генерує апаратні переривання (використовується програмне опитування) 1 – KBI генерує запит на апаратне переривання, коли KBF = 1
0 KBIMOD	Біт встановлення режиму детектування переривань клавіатури (**Keyboard Detection Mode**). Цей біт управління для читання/запису задає конкретний вигляд «активного» вхідного сигналу (фронт та рівень або тільки фронт) на лініях [7:4] модуля KBI, за яким буде генеруватись переривання модулем. Лінії [3:0] модуля KBI налаштовуються тільки на генерування переривань за спадаючим або висхідним фронтом вхідного сигналу. Можливі значення: 0 – генерування переривання лише за фронтом вхідного сигналу; 1 – генерування переривання за фронтом та рівнем вхідного сигналу. При KBEDGn = 1 - висхідний фронт/високий рівень; При KBEDGn = 0 – спадаючий фронт/низький рівень.

Нижче наведено приклад програми, що містить код ініціалізації модуля KBI та синтаксис написання обробника переривань за натисканням кнопки:

Приклад 4.

```
#include <hidef.h> /* for EnableInterrupts macro */
```

```
#include "derivative.h" /* include peripheral declarations */
#include "mc13213.h"

/* макроси для покращення розуміння коду програми */
#define Buzz     PTDD_PTDD2 // Звуковий індикатор на pin 2 порту D
#define LED1   PTDD_PTDD4 // Світлодіод на pin 4 порту D
#define SW1     PTAD_PTAD2 // Кнопка 1 на pin 2 порту A
#define SW2     PTAD_PTAD3 // Кнопка 2 на pin 3 порту A
#define ON      0
#define OFF     1
#define DATA_IN_MODE     0
#define DATA_OUT_MODE    1
#define PULLUP_Enable 1
typedef

/*
** ===================================================================
**     Метод      :  KBI1_Init
**     Ініціалізує один модуль KBI (їх може бути декілька в контролерах) та необхідні змінні
**     Режим роботи задіяної вхідної лінії модуля  KBIPE2 «за фронтом» (за замовчанням),
**     не вимагає додаткових налаштувань регістру KBISC.
** ===================================================================
*/

void KBI1_Init(void)
{
  PTADD_PTADD2 = DATA_IN_MODE; // встановлення pin 2 порту A в режим введення
  PTAPE_PTAPE2 = PULLUP_Enable;      // підключення підтягуючого резистору
  PTADD_PTADD3 = DATA_IN_MODE; // встановлення pin 3 порту A в режим введення
  PTAPE_PTAPE3 = PULLUP_Enable;      // підключення підтягуючого резистору

  KBI1PE_KBIPE2=1;  /* підключаємо лінію GPIO PTA2 до входу модуля KBIP2
                       ** та дозволяємо генерувати переривання за відповідним входом
                       модуля KBI */
  KBI1PE_KBIPE3=1;  /* підключаємо лінію GPIO PTA3 до входу модуля KBIP3
                       ** та дозволяємо генерувати переривання за відповідним входом
                       модуля KBI */

  KBI1SC_KBIE = 1;   //  дозволяємо генерувати переривання модулем KBI
} /* KBI1_Init */

/*
** ===================================================================
**     Метод      :  KBI_OnKlick
**     Обробник переривань від модуля KBI
**     VectorNumber_Vkeyboard1 - ім'я номера вектору переривань з таблиці векторів
**     переривань (див. файл заголовку mc13213.h)
**     Переривання модулем буде генеруватись при натисканні кнопки на будь-якому
**     дозволеному вході KBI
** ===================================================================
*/
```

```
interrupt VectorNumber_Vkeyboard1 void KBI_OnKlick(void)
{
     KBI1SC_KBACK = 1;   // скидаємо прапор KBF, підтверджуючи обробку переривання

    if (!SW1)                    // перевіряємо чи переривання викликане саме кнопкою 1
          {   /*  код реакції на натискання кнопки */
            Buzz = ON;          // звук, якщо натиснута кнопка 1
          }
    else   Buzz = OFF;          // вимикаємо звук будь-якою іншою «дозволеною» кнопкою
} /* KBI_OnKlick */

void Delay(int delayms)          // функція програмної затримки
{
          int i;
          for(i=0;i<=delayms*10000;i++)
          { //__RESET_WATCHDOG();        }

void main(void) {

   EnableInterrupts;   // дозволяємо глобальне переривання ЦП (скидаємо ознаку CCR:I = 0)
   SOPT_COPE=0;   // так можна вимкнути WATCH DOG

  PTDDD_PTDDD4 = DATA_OUT_MODE; // лінію, підключену до світлодіоду "на вихід"
  LED1=OFF;                                    // вимикаємо світлодіод перед початком роботи

  KBI_Init();                                  // ініціалізуємо установки модуля KBI

  for(;;) {
    LED1 = !LED2;                    // Блимаємо діодом
    Delay(2);                        // Програмна затримка
//    __RESET_WATCHDOG();       /* годуємо «песика», якщо він працює */
  } /* кінець основного безперервного циклу for(;;) */
} /*кінець main*/
```

Завдання

1. Вивчити опис збірки MC 13213 та тестувальної плати 13213-SRB. Ознайомитись з підключенням кнопок до входів/виходів загального призначення збірки MC 13213 на тестувальній платі (таблиця 1).
2. Ознайомитись з особливостями реалізації та використання механізму переривань модуля KBI у контролері HCS08 збірки MC 13213.
3. Використовуючи модуль переривань KBI:
   - запрограмувати по натисканню кнопки SW1 вмикання та неперервне світіння діоду LED1, по натисканню кнопки SW2 вимикання діоду LED1, по натисканню кнопки SW3 блимання діоду LED1;

- запрограмувати щоб при кожному наступному натисканню кнопки SW1 світлодіод LED1 циклічно змінював стан у наступній послідовності: увімкнений, блимає, вимкнений;
- запрограмувати дію: кнопка SW4 натиснута – звуковий індикатор (Buzzer) звучить; кнопка SW4 відтиснута - звуковий індикатор (Buzzer) вимкнено.

4. Завдання на додаткові бали:
   - закоментуйте рядок скидання прапору KBF в обробнику переривань. Поясніть зміни у результатах роботи програми.

Порядок виконання роботи та методичні вказівки

1. Створити порожній проект у середовищі CodeWarrior у відповідності до інструкції у додатку 1.
2. Написати програму для виконання завдання 3.1.
3. Підключити програматор до порту BDM тестувальної плати.
4. Підключіть програматор до USB порту комп'ютера.
5. Підключіть живлення до тестувальної плати та увімкніть її за допомогою відповідного перемикача (далі до закінчення роботи даний перемикач можна використовувати для вимкнення/вмикання тестової плати).
6. Послідовно створіть та налагодьте програми, що реалізують дії, описані у пунктах завдання 3.
7. Підготуйте та здайте звіт.

Контрольні питання

1. Що таке апаратне переривання і навіщо воно потрібне?
2. Що таке вектор переривань? Який номер вектору переривань модуля KBI
3. Чому в основному тексті програми (приклад 3) явним чином ні де не викликається метод обробника переривань?

4. У чому різниця у налаштуванні режимів генерування переривань за входами [7:4] та входами [3:0] модуля KBI.
5. Що буде, якщо при обробці переривань модуля KBI не скидати прапор KBF?

Лабораторна робота № 4. Модуль таймера/широтно-імпульсного модулятора

Мета роботи: ознайомитись з базовою структурою модулів таймерів/ШІМ мікроконтролерів та опанувати програмне керування роботою таймера в режимі вихідного порівняння (output compare).

Теоретичні відомості

Загальні відомості про модулі таймера/широтно-імпульсного модулятора TPM мікроконтролера HCS08

Для синхронізації різних дій у вбудованих системах, для вимірювання та завдання часових параметрів внутрішніх та зовнішніх сигналів, мікроконтролери, що використовуються у таких системах, містять у своєму складі один або декілька модулів таймерів.

Мікроконтролер збірки MC13213 включає в себе два незалежних модулі TPM (**T**imer/**P**W**M**) - таймер/широтно-імпульсний модулятор (ШІМ), які суміщають у собі, власне, функціонал таймеру, та функціонал широтно-імпульсного модулятора PWM (**P**ulse-**W**idth **M**odulator).

TPM1 - 3-канальний модуль таймер/ШІМ та TPM2 - 5-канальний модуль таймер/ШІМ підтримують типові режими роботи: захоплення входу (*input capture*), вихідне порівняння (*output compare*) та широтно-імпульсну модуляцію (*pulse-width modulation*). Кожен із згаданих модулів дозволяє незалежно обрати джерело тактування, дільник частоти тактування, встановлювати максимальне число тактових імпульсів у періоді лічби та задавати напрям лічби – тільки вгору або вгору/вниз.

Деякі канали модулів TPM збірки MC13213 можуть під’єднуватись до ліній введення/виведення (див. рис.2). Зокрема, канал 2 модуля TPM1 має виділену лінію, яку під’єднано до піну 47 збірки (туди ж під’єднано 2 лінію порту D). У модуля TPM2 виділені лінії мають канали 1-4, які під’єднано до

пінів 48-51 збірки (вони ж лінії 4-7 порту D), відповідно. Якщо канал модуля використовується тільки як програмний таймер (для формування часових інтервалів заданої тривалості), його виділену лінію можна вимкнути, і відповідний пін використовувати як лінію введення/виведення порту D загального призначення.

Структуру модулів TPM, на прикладі TPM1 та його каналів 0 та 1, зображено на рис.10. На схемах і у документації до збірки MC13213 модулі таймер/ШІМ позначають як TPMx, а їх канали як TPMxCHn, де x - номер модуля TPM (1 або 2), а n - номер каналу (наприклад, 0-4).

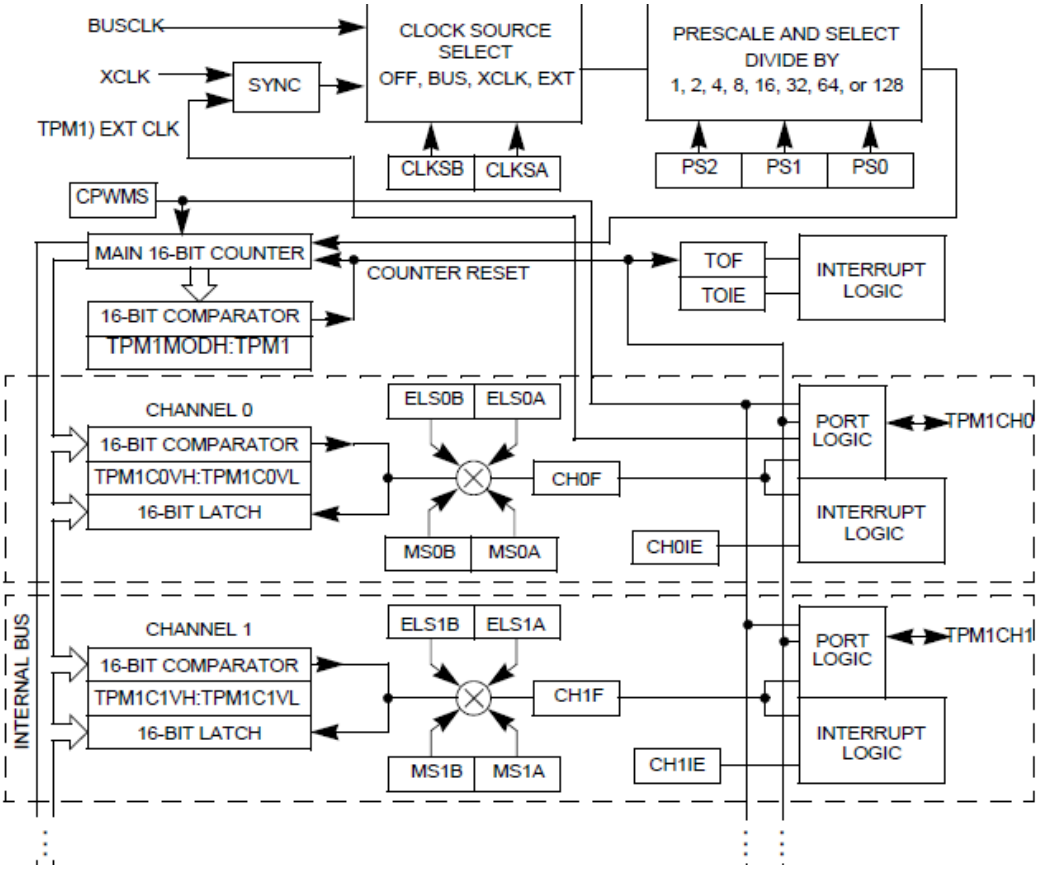


Рис.10. Блок-схема модуля таймера/ШІМ - TPM1.

Кожен модуль TPMx включає в себе:

- 8-бітний регістр стану та керування TPMxSC;
- 16-розрядний лічильник (на рис.10 позначено як *main 16-bit counter*) з регістром відліків TPMxCNT, утвореним з двох 8-бітних частин (TPMxCNTH : TPMxCNTL),

- 16-бітний модуль регістр TPMxMOD, утворений з двох 8-бітних частин (TPMxMODH : TPMxMODL).

  Кожен канал модуля TPMx має:

- 8-бітний регістр стану та керування (TPMxCnSC),
- 16-бітний регістр значень каналу TPMxCnV, утворений з двох 8-бітних частин (TPMxCnVH : TPMxCnVL).

Регістр стану та керування модуля TPMx (Timer x Status and Control Register) TPMxSC.

Загальні налаштування модуля TPMx - вибір джерела тактування (*clock source select*), встановлення дільника тактової частоти (*prescale and select divide*), маскування переривання по подіям «переповнення» лічильника, встановлення режиму широтно-імпульсної модуляції та контроль переповнення лічильника через читання відповідного біту-прапору, відбувається за допомогою регістра стану та керування TPMxSC. Налаштування даного регістру є загальними для всіх каналів модуля TPMx. Загальний опис структури регістру TPMxSC та призначення його полів (бітів) наведено на рис.11. та у таблиці табл.4. Детальний опис полів дано нижче.

Bit	7	6	5	4	3	2	1	0
R	TOF	TOIE	CPWMS	CLKSB	CLKSA	PS2	PS1	PS0
W								
RESET	0	0	0	0	0	0	0	0

Рис.11. Регістр стану та керування TPMxSC.

Таблиця 4. Опис полів регістру стану та керування модулів таймерів TPMx.

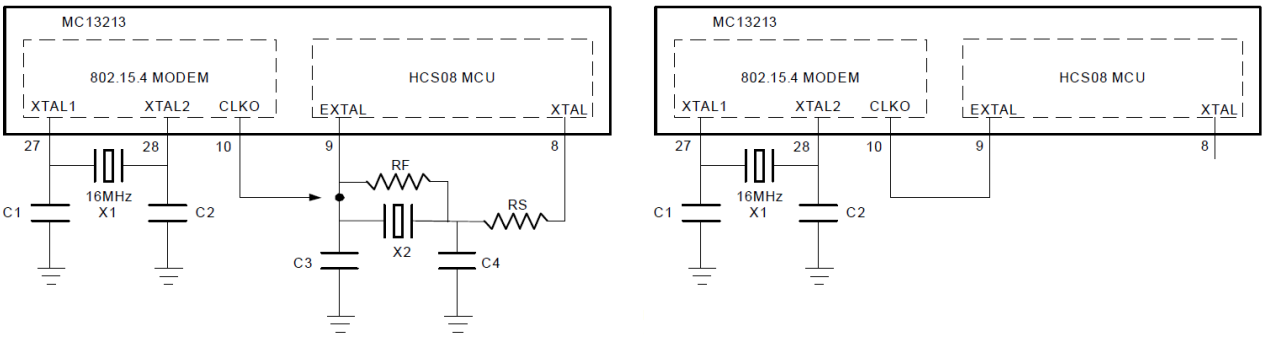
Поле	Опис
7 TOF	Бітове поле прапору переповнення лічильника таймера **(Timer Overflow Flag)**. Даний прапор встановлюється тоді, коли вміст регістру TPMxCNT таймера занулився (став рівним $0000), попередньо досягнувши значення, рівного тому, що записане в модуль-регістрі TPMxMOD. Коли лічильник TPMx налаштовується для циклічної лічби вгору/вниз (CPWMS=1), прапор TOF встановлюється тоді, коли вміст регістру TPMxCNT стає рівним вмісту модуль-регістру і змінює напрям лічби (переходить до наступного нижчого значення). Коли прапор TOF встановлений, його необхідно скинути, спочатку

	прочитавши, а потім, записавши у нього 0. Якщо нове переповнення лічильника таймеру наступає до завершення процедури скидання прапору TOF від попереднього переповнення, прапор залишиться нескинутим навіть після завершення процедури скидання. 0 - лічильник TPM не досяг значення модуля або переповнення. 1 - лічильник TPM був переповнений.
6 TOIE	Бітове поле дозволу переривань по переповненню лічильника таймера **(Timer Overflow Interrupt Enable).** Біт (поле) для читання/запису, встановлення якого дозволяє переривання при переповненні (або досягненні значення, рівному записаному у модуль-регістрі TPMxMOD) лічильника TPMx, коли прапор TOF встановлюється в 1. Якщо переривання не дозволене (TOIE рівне 0), прапор TOF все одно встановлюється при переповненні лічильника таймеру і може бути контрольований програмно.
5 CPWMS	Бітове поле для налаштування режиму ШІМ **(Center-Aligned PWM Select)**. Скидання даного біту (CPWMS = 0) переводить лічильник в режим підрахунку вгору, при цьому кожен канал модуля може бути незалежно налаштований на режим захоплення вхідного сигналу, вихідного порівняння та функцій ШІМ з вирівнюванням до границі періоду, відповідно до налаштування бітів керування MSnB: MSnA у регістрі статусу та керування кожного каналу. Встановлення CPWMS = 1 налаштовує лічильник на лічбу вгору/вниз, при цьому усі канали модуля працюватимуть у режимі формування центрально вирівняної ШІМ з параметрами, встановленими у регістрі статусу та керування кожного каналу.
4:3 CLKS[B:A]	2-бітне поле для вибору джерела тактування модуля TPM (див. табл.5). **(Clock Source Select)**. Запис у поле значення 0:0 вимикає модуль, оскільки при цьому джерело тактування вважається необраним.
2:0 PS[2:0]	3-бітне поле для вибору дільника для частоти джерела тактування **(Prescale Divisor Select).** Частота тактування (лічби) таймера зменшується у встановлену кількість разів у порівнянні з основною частотою джерела тактування (табл.6).

Джерела тактування збірки MC13213

Збірка MC13213 поєднує у собі радіочастотний модем та мікроконтролер. Виробником передбачено можливість їх незалежного тактування від двох окремих кварцових осциляторів X1 та X2 (рис.12.а). У цьому випадку дещо зменшується споживання схеми тактування модему, оскільки її вихід CLKO, який може використовуватись для тактування мікроконтролела, можна вимкнути. В другому варіанті джерелом тактування контролера є вихід CLKO схеми тактування модему (рис.12.б). Вибір такої схеми тактування робить

вбудовану систему на основі збірки MC13213 дешевшою за двох осциляторний варіант, без погіршення стабільності її характеристик у цілому.



а), б)

Рис.12. Спрощена схема підключення джерел тактових імпульсів до збірки MC13213.

На виході CLKO модему можна отримати тактові імпульси з максимальною частотою рівною частоті кварцового осцилятора 16 МГц. Дану частоту можна ділити, програмуючи відповідний регістр керування 0A. В результаті на виході CLKO, можна отримати тактові імпульси з частотами 16 МГц, 8 МГц, 4 МГц, 2 МГц, 1 МГц, 62.5 кГц, 32.786 кГц (використовується за замовчанням) та 16.393 кГц. Внутрішня схемотехніка тактування контролера дозволяє помножувати довільну частоту зовнішнього опорного джерела тактування аж до 32 МГц або ділити, у залежності від потреб системи. Для цього використовуються регістри ICGC1, ICGC2, ICGS1 та ICGTRM внутрішнього генератора тактових імпульсів мікроконтролера (див. приклади у [3]).

Третій варіант передбачає тактування контролера лише від його внутрішнього джерела тактових імпульсів з невисокою стабільністю з частотами 8МГц та 243кГц. Хоча, у даному випадку, дещо зменшується споживання схеми тактування модему і контролера, однак невисока стабільність внутрішнього генератора тактових імпульсів контролера не дозволяє адекватно працювати додаткам, критичним до точності визначення

часових інтервалів. Даний варіант тактування контролера використовується: при відпусканні кнопки перезавантаження збірки, щоб контролер міг запуститись; при перезавантаженні модему, коли вихід CLKO тимчасово недоступний; коли програмно необхідно змінити частоту імпульсів на виході CLKO, та поки у схемі контролера підбирається новий множник для відповідної нової зовнішньої опорної частоти тактування.

Одноосциляторна схема рис.12.б, де вихід схеми тактування модему CLKO є опорним джерелом тактових імпульсів для контролера, є найбільш оптимальним варіантом з точки вартості та точності часових характеристик і саме вона реалізована на тестувальній платі MC13213-SRB.

Вибір джерела тактування та дільника частоти тактування модуля TPM

Вибір джерела тактування модуля TPM здійснюється через завдання контрольних бітів CLKSB та CLKSA регістру стану та керування (див. табл.5). Після кожного перезавантаження контролера збірки MC13213, біти CLKSB:CLKSA встановлюються у 0:0, тобто джерело тактування необране і відповідний модуль TPM неактивний. Зазвичай біти CLKSB:CLKSA встановлюють в 0:1, тоді тактування модуля здійснюється від системної шини контролера. Максимальна частота тактування процесору мікроконтролера 40 МГц а максимальна частота тактування системної шини - 20 МГц. Коли контрольні біти CLKSB:CLKSA встановлені у 1:1, модуль TPMx тактується від зовнішнього джерела тактування, яке має бути під’єднаним до піну каналу 0 модуля (TPMxCH0). Максимальна дозволена частота тактування для зовнішнього джерела становить 1/4 частоти системної шини (*bus rate*). При встановленні бітів CLKSB:CLKSA у 1:0, модуль TPM як джерело квантування використовуватиме так-зване фіксоване системне джерело тактування (*fixed system clock*), частота якого визначається налаштуваннями внутрішнього тактового генератора ICG (*internal clock generator*) мікроконтролера генератора. Дане джерело обирається тільки у специфічних режимах роботи внутрішнього

генератора ICG і не є пріоритетним (більше інформації про дане та інші джерела тактування у документації [1]).

Таблиця 5. Вибір джерела тактування модуля TPM.

CLKSB:CLKSA	Джерела тактування для TPM
0:0	джерело не вибрано (таймер вимкнено)
0:1	джерело – джерело тактування системної шини (bus rate clock (BUSCLK).
1:0	джерело – фіксований системний годинник Fixed system clock (XCLK).
1:1	джерело - зовнішнє (External source (TPM1 Ext Clk)[1,2]

[1] Максимальна дозволена частота для зовнішнього джерела становить 1/4 частоти системної шини (bus rate).

[2] Якщо як джерело тактування TPM вибрано пін каналу TPM1CH0, відповідні біти ELS0B: ELS0A у його регістрі стану та керування TPM1C0SC повинні бути встановлені в 0: 0, так щоб канал 0 не намагався цей же пін використати для конфліктуючої функції.

Для збільшення гнучкості у визначенні та завданні часових параметрів вхідних і вихідних сигналів у модулях TPM передбачена можливість зниження їх робочої частоти тактування шляхом ділення частоти обраного джерела тактування на заданий коефіцієнт. Вибір дільника здійснюється через встановлення 3-бітного поля PS2:PS1:PS0 регістру стану та керування відповідно до таблиці 6.

Таблиця 6. Вибір дільника частоти джерела тактування модуля TPM.

PS2:PS1:PS0	Дільник для частоти джерела TPM
0:0:0	1
0:0:1	2
0:1:0	4
0:1:1	8
1:0:0	16
1:0:1	32
1:1:0	64
1:1:1	128

Регістр лічильника модуля TPMx (Timer x Counter Registers) TPMxCNT

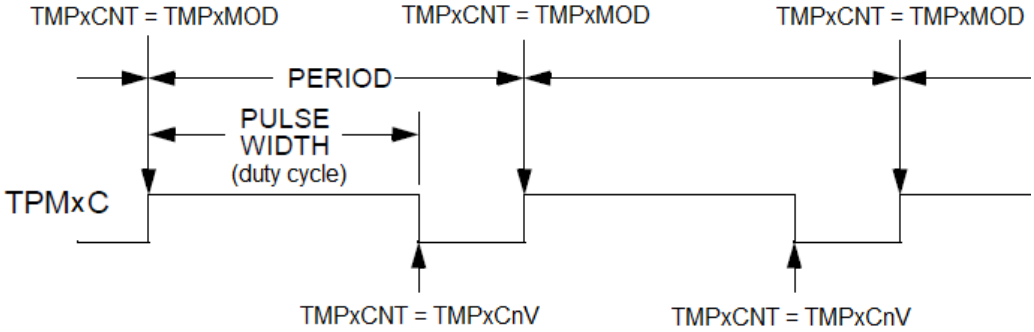
Центральним компонентом модулів TPM є 16-бітний лічильник, окремий для кожного модуля, який може працювати як лічильник вільного відліку, лічильник з обмеженою лічбою або лічильник вгору/вниз. Даний лічильник використовує двобайтний регістр TPMxCNT (TPMxCNTH - старший байт регістру, TPMxCNTL – молодший байт регістру) і є спільним для всіх каналів відповідного модуля x. Лічильник (при роботі в звичайному режимі лічби вгору) забезпечує тактові відліки для всіх режимів роботи таймера. Програмне читання поточного значення лічильника з регістру TPMxCNT, або окремо з його байтів TPMxCNTH: TPMxCNTL можливе в будь-який момент часу і не впливає на процес лічби. Запис даних у старший або молодший байт регістру TPMxCNT лічильника скидає лічильник (регістр TPMxCNT занулюється і відлік починається спочатку), незалежно від змісту даних, що записуються.

Модуль-регістр модуля TPMx (Timer x Counter Modulo Registers) TPMxMOD.

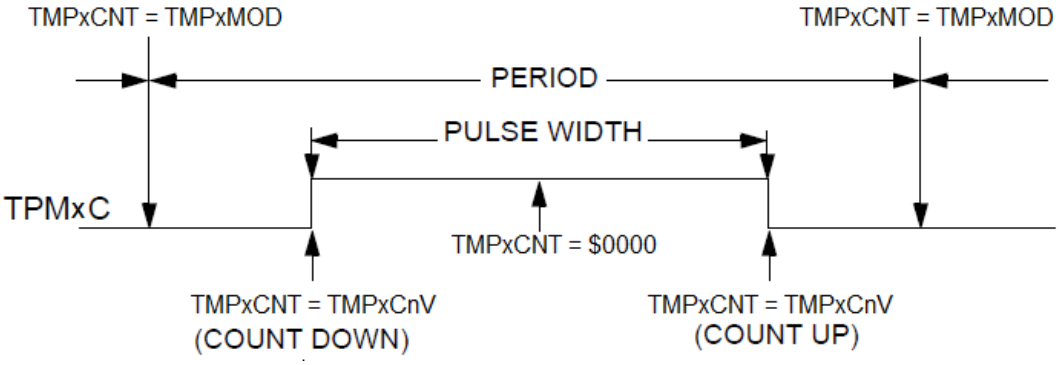
16-розрядний модуль-регістр TPMxMOD лічильника (modulo register), що також утворений з двох восьмирозрядних частин - TPMxMODH: TPMxMODL, дозволяє задавати верхнє значення відліку лічильника. По досягненню у регістрі TPMxCNT значення, рівного тому, що записане у регістрі TPMxMOD, при CPWMS = 0, встановлюється прапор TOF = 1, генерується переривання, якщо воно дозволене (TOIE = 1), і лічба починається спочатку зі значення $0000, а при CPWMS = 1, напрям лічби змінюється вниз. Запис у регістр TPMxMOD значень $ 0000 або $ FFFF переводить лічильник у режим вільного відліку (коли той рахує від $0000 до $FFFF циклічно). Рівність значень регістрів TPMxMOD та TPMxCNT відслідковується компаратором (рис.10).

Регістр значень каналу n модуля TPMx (Timer x Channel n Value Registers) TPMxCnV.

Якщо канал модуля TPM налаштовано на роботу у режимі захоплення, при настанні налаштованої вхідної події (висхідного/спадного фронту, високого/низького рівня сигналу) на піні каналу, в регістр TPMxCnV переноситься поточне значення лічильника з регістру TPMxCNT. При CPWMS = 0 та налаштуванні каналу модуля на режим вихідного порівняння або ШІМ з вирівнюванням по краях періоду (*edge-aligned PWM*), по досягненню у регістрі TPMxCNT значення, рівного тому, що записане у регістрі TPMxCnV, ініціюється встановлення відповідного прапору каналу CHnF = 1 та генерується переривання, якщо воно дозволене (CHnIE = 1), а у режимі ШІМ ще й змінюється рівень на вихідній лінії каналу (рис.13.а).



а)



б)

Рис.13. Формування ШІМ сигналу: а) вирівнюванням по краях періоду, «додатнє» заповнення імпульсу (CPWMS = 0, ELSnA = 0); б) – з центральне вирівнюванням періоду, «додатнє» заповнення імпульсу (CPWMS = 1, ELSnA).

Регістри TPMxCnV значення каналів очищуються шляхом скидання (запису у регістри значення $0000). При налаштуванні каналу на режим вихідного порівняння або ШІМ, запис у регістр TPMxCnV значення більшого за значення регістру TPMxMOD призводитиме до того, що подія рівності TPMxCNT = TPMxCnV ніколи не настане, як і ніколи не виконається дія, пов'язана з цією подією.

Регістр стану та керування каналу n модуля TPMx (Timer x Channel n Status and Control Register) TPMxCnSC.

Регістр TPMxCnSC кожного каналу модулів TPM містить бітове поле CHnF прапору порівняння регістру лічильника TPMxCNT та регістру значень TPMxCnV. Прапор встановлюється коли значення у цих регістрах співпадають. Також регістр TPMxCnSC містить біти керування, які використовуються для налаштування дозволу переривання CHnIE, при встановленні прапора CHnF, для конфігурації каналу та функції вихідної лінії (піну) (див рис.14). Перелік полів регістру та їх призначень коротко описано у таблиці 7 та таблиці 8.

Bit	7	6	5	4	3	2	1	0
R	CHnF	CHnIE	MSnB	MSnA	ELSnB	ELSnA	0	0
W								
RESET	0	0	0	0	0	0	0	0

Рис.14. Регістр стану та керування каналу TPMxCnSC.

Таблиця 7. Опис полів регістру стану та керування TPMxCnSC.

Поле	Опис
7 CHnF	Бітове поле прапору порівняння регістру лічильника TPMxCNT та регістру TPMxCnV значень каналу n (**Channel n Flag**). Якщо канал n налаштовано для захоплення вхідного сигналу, прапор встановлюється (CHnF = 1) коли на вхідній лінії каналу n виникає «активний» фронт/рівень (рівність TPMxCNT = TPMxCnV виконується автоматично, оскільки значення TPMxCNT копіюється у TPMxCnV принастанні відповідної події). При налаштуванні каналу n на режим вихідного порівнянням або ШІМ, прапор встановлюється завжди, коли настає подія TPMxCNT = TPMxCnV, тобто для режиму ШІМ з вирівнюванням по центру прапор встановлюється двічи на період сигналу. Для скидання прапору CHnF треба здійснити зчитування регістру TPMxCnSC, а потім запишіть 0 до CHnF. Якщо наступна подія рівності TPMxCNT = TPMxCnV виникає до завершення скидання прапору, прапор

	залинається встановленим. Спроба запису 1 до CHnF не має ніякого ефекту.
6 CHnIE	Бітове поле дозволу переривання по встановленню прапора CHnF у каналі (**Channel n Interrupt Enable**). CHnIE = 0 - запит на переривання по встановленню прапору не дозволено. CHnIE = 1 - запит на переривання по встановленню прапору дозволено. Значення у даному полі на впливає на встановлення прапору CHnF.
5 MSnB	Бітове поле вибору режиму B для каналу n (**Mode Select B for TPM Channel n**) Вибір режиму B Коли CPWMS = 0, MSnB = 1 налаштовує TPM-канал n в режим ШІМ з вирівнюванням по краях періоду (детальніше про режими у таблиці 8).
4 MSnA	Бітове поле вибору режиму A для каналу n (**Mode Select A for TPM Channel n**) Коли CPWMS = 0, і MSnB = 0, MSnA налаштовує TPM-канал n в режим захоплення або режим порівняння (детальніше про режими у таблиці 8).
3:2 ELSn[B:A]	2-бітове поле встановлення типу «активного» сигналу або ШІМ – рівня/типу фронту.(**Edge/Level Select Bits**). Залежно від робочого режиму для каналу таймера, заданого встановленням полів CPWMS:MSnB:MSnA і показаного в таблиці 8, ці біти задають тип «активного» фронту (спадний/висхідний), який запускає подію захоплення вхідного сигналу. При налаштуванні каналу на режим ШІМ дані біти визначають рівень (іноді його називають - полярність), який (яка) буде заповнювати активну частину (duty cycle) ШІМ сигналу. Встановлення ELSnB:ELSnA = 0:0 від'єднує від каналу пов'язаний з ним пін, і надалі цей пін може використовуватись як лінія введення/виведення загального призначення. Ця функція зазвичай використовується для тимчасового вимкнення вхідного каналу захоплення або для того, щоб звільнити відповідний пін для функцій введення/виведення загального призначення, коли пов'язаний канал таймера налаштовано як програмний таймер. Таке налаштування, необхідне для каналу 0 модуля 1 (TPM1CH0), коли пов'язаний з ним пін обирається як вхід для зовнішнього джерела тактування модуля (див. «Вибір джерела тактування та дільника частоти тактування модуля TPM»).

Таблиця 8. Вибір типу «активного» сигналу для каналів модулів TPM.

CPWMS	MSnB:MSnA	ELSnB:ELSnA	Режим	Конфігурація
X	XX	00	Виділена лінія не використовується каналом і пов'язаний з нею пін може використовуватись як вжід для джерела зовнішнього тактування або як лінія введення/виведення загального користування.	
0	00	01 10 11	Вхідне захоплення	Захоплення тільки по висхідному фронту Захоплення тільки по спадному фронту Захоплення по висхідному або

				спадному фронту
	01	00 01 10 11	Вихідне порівняння	Тільки програмне порівняння (вихідна лінія не задіяна). Інверсія рівню на вихідній лінії по встановленню прапора CHnIE. Низький рівень на вихідній лінії по встановленню прапора CHnIE. Високий рівень на вихідній лінії по встановленню прапора CHnIE.
	1X	10 X1	ШІМ з вирівнюванням по краях	Формування на вихідній лінії імпульсів (duty cycle) високого рівня. Формування на вихідній лінії імпульсів (duty cycle) низького рівня.
1	XX	10 X1	Центрально вирівняне ШІМ	Формування на вихідній лінії імпульсів (duty cycle) високого рівня. Формування на вихідній лінії імпульсів (duty cycle) низького рівня.

Завдання

1. Ознайомитись з особливостями будови модулів таймер/ШІМ ТРМ збірки МС 13213 та з особливостями їх роботи у різних режимах.
2. Запрограмувати один з модулів ТРМ збірки МС 13213 у відповідності до завдань нижче:
   - за натисканням кнопки 1 запустити періодичне вмикання/вимикання діоду 1 з інтервалом Т=1сек. Для відліку часових інтервалів використати один з каналів модулів ТРМ, запрограмувавши його в режимі вихідного порівняння без використання вихідної лінії. Перемикання діоду реалізувати в тілі обробника переривань обраного каналу. Заданий часовий інтервал встановити шляхом підбору значень регістрів TPMxMOD, TPMxCnV та дільника татової частоти джерела.
   - за натисканням кнопки 2 зупиняти таймер.
   - за натисканням кнопки 3 збільшувати період блимання діоду вдвічі.

- за натисканням кнопки 4 зменшувати період блимання діоду вдвічі.

Порядок виконання роботи та методичні вказівки

1. Створити порожній проект у середовищі CodeWarrior у відповідності до інструкції у додатку 1.
2. Написати програму для виконання завдання 2.1.
3. Підключити програматор до порту BDM тестувальної плати (за необхідності скористайтесь допомогою викладача).
4. Підключіть програматор до USB порту комп'ютера.
5. Підключіть живлення до тестувальної плати та увімкніть її за допомогою відповідного перемикача (далі до закінчення роботи даний перемикач можна використовувати для вимкнення/вмикання тестової плати).
6. Послідовно створіть та налагодьте програми, що реалізують дії, описані у пунктах завдання 2.
7. Підготуйте та здайте звіт.

Вказівки для роботи з таймером

В лабораторній роботі пропонується опанувати роботу з модулем TPM на прикладі його програмування у режимі вихідного порівняння без задіяння виділеної вихідної лінії, тобто, у режимі програмного таймера. Для відстеження часових інтервалів пропонується використовувати переривання від відповідного каналу модуля.

Для ініціалізації роботи модуля TPM у режимі програмного таймера необхідно:

1) Скинути налаштування обраного модуля TPMx (використовуємо регістр статусу та керування TPMxSC):

```
TPMxSC = 0; //  TOF=0, TOIE=0, CPWMS=0, CLKSB=0, CLKSA=0, PS2=0, PS1=0, PS0=0.
```

2) Задати режим вихідного порівняння обраного каналу та дозволити переривання в каналі (використовуємо регістр статусу та контролю каналу TPMxCnSC):

TPMxCnSC = 0b01010000; // CHnF=0, CHnIE=1, MSnB=0, MSnA=1, ELSnB=0, ELSnA=0, -, -

3) Встановити значення дільника частоти джерела тактування (через регістр TPMxSC, наприклад 128):

TPMxSC_PS = 0b111; // PS2=1, PS1=1, PS0=1

4) Встановити бажане значення модуль регістра TPMxMOD та регістра значень TPMxCnV, наприклад:

TPMxCnV=0x1E84; TPMxMOD = 0x1E84; //

5) Скинути регістр лічильника:

TPMxCNT = 0;

Для запуску таймера необхідно обрати джерело тактування (наприклад системну шину):

TPMxSC_CLKSA=1; TPMxSC_CLKSB=0; // Bus rate clock (BUSCLK) selected

6) Для зупинення таймера необхідно відключити джерело тактування:

TPM1SC_CLKSA=0; TPM1SC_CLKSB=0;

Приклад визначення обробника переривань каналу 0 таймера TPM1 з використанням відповідного вектору переривань **VectorNumber_Vtpm1ch0**:

Приклад 5.

```
/*
** ==========================================================================
**    Метод      :  TI1_OnTick
**    Обробник переривань від каналу 0 модуля TPM1
**    VectorNumber_Vtpm1ch0 - ім'я номера вектору переривань з таблиці векторів
**    переривань (див. файл заголовку mc13213.h)
**    Переривання буде генеруватись при настанні події порівняння TPM1C0V = TPM1CNT,
**    коли встановиться прапор TPM1C0SC_CH0F
** ==========================================================================
*/
interrupt VectorNumber_Vtpm1ch0 void TI1_OnTick(void)
{
   if (TPM1C0SC_CH0F)  // Читання та перевірка на встановлення прапору у каналі
   {          TPM1C0SC_CH0F =0;}    // Скидання прапора
         /*тут можна писати свій код */    }
```

Наведений обробник переривань викликатиметься кожного разу при настанні події TPM1CNT = TPM1C0V. Очевидно, що частота виклику обробника (частота «тіків» таймера) визначатиметься швидкістю лічби лічильника модуля TPM1 та значеннями, записаними у регістрах TPM1MOD та TPM1C0V.

Оскільки у прикладі як джерело тактування модуля TPMx пропонується обрати системну шину мікроконтролера, саме частота тактових імпульсів системної шини буде визначати швидкість лічби відповідного лічильника. У лабораторній роботі частоту тактування системної шини (по замовчанню має бути 4МГц) рекомендується оцінити експериментально, варіюючи значення регістрів TPM1MOD, TPM1C0V та дільник тактової частоти.

Контрольні питання

1. Поясніть призначення структурних елементів модулів TPM рис.10. Чи залежить це призначення від режиму роботи таймеру?
2. Які номери векторів переривань модуля таймеру та його каналів?
3. Коли генерується глобальне переривання у модулі TPM, а коли переривання за каналом?
4. Чи можна змінювати частоту лічби модуля TPM?
5. Поясніть різницю роботи модуля TPM при налаштуванні у режим захоплення та у режим вихідного порівняння. Що з чим порівнюється у режимі порівняння?

Лабораторна робота № 5. Модуль аналого-цифрового перетворювача ATD

Мета роботи: опанувати принципи роботи з аналоговими сигналами у вбудованих системах.

Теоретичні відомості

Загальні відомості про модуль аналого-цифрового перетворення ATD мікроконтролера HCS08

Аналогові датчики температури, тиску, вологості, витрат, освітленості, швидкості і багато інших, є невід'ємною частиною будь-якої системи управління різними об'єктами – розумними будинками, автомобілями, технологічними виробництвами тощо. Для роботи з такими датчиками кожна вбудована система містить один або декілька модулів аналого-цифрових перетворювачів (АЦП). Мікроконтролер HCS08 мікросхеми MC13213 також має модуль аналого-цифрового перетворювача ATD (**A**nalog-**t**o-**d**igital converter) послідовного наближення SAR (**S**uccessive **A**pproximation **R**egister). Основні характеристики даного АЦП:

• розрядність - 8-/10-bit;

• час одиночного перетворення з 10-bit розрядністю - 14.0 мкс, при тактовій частоті 2 МГц;

• ліве/праве вирівнювання 10-ти розрядного результату перетворення (відносно границь двобайтного регістру де розміщуються результати перетворення);

• беззнакове та знакове представлення результатів перетворення;

• прапор сигналізації завершення перетворення, переривання по завершенню перетворення;

• 8-ми канальний мультиплексор;

• режими одиночної та циклічної вибірки.

Блок-схема модуля АЦП наведена на рисунку 15. На схемі AD1P0 - AD1P7 – вісім вхідних ліній АЦП для підключення аналогових джерел сигналів; V_{REFH}, V_{REFL} – потенціали опорної напруги, яка визначає верхню та нижню межу напруги вхідного сигналу для можливості його коректного перетворення у АЦП. V_{DDAD} – напруга живлення АЦП, V_{SSAD} – потенціал «землі» АЦП.

Вхідні аналогові сигнали мають бути уніполярними і, для коректного перетворення, не мають виходити за межі опорних напруг V_{REFH}, V_{REFL}. На тестувальній платі розробки MC13213-SRB потенціали V_{REFH} та V_{REFL} складають 0В та 5В, відповідно, що задає діапазон аналогових сигналів на вході АЦП: 0 - 5В.

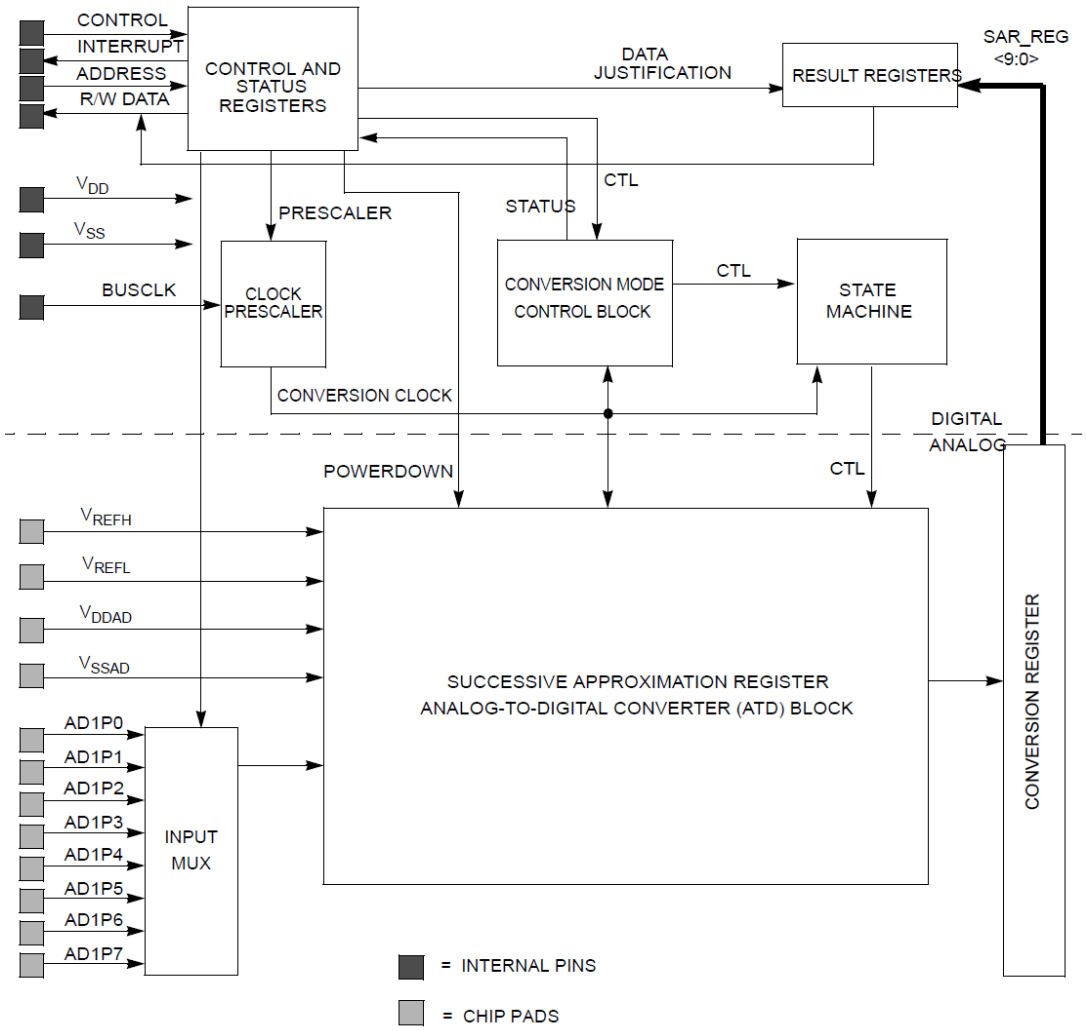


Рис.15. Блок-схема модуля АЦП мікроконтролера HCS08.

Багатоканальність АЦП забезпечує аналоговий мультиплексор INPUT MUX, який дозволяє послідовно підключати до блоку перетворення АЦП один з восьми входів AD1P0 - AD1P7.

Як видно з рисунку 2 та таблиці 1 вхідні лінії АЦП поділяють піни 52-59 збірки MC13213 з лініями введення/виведення загального призначення 0-7 порту B.

АЦП мікроконтролеру HCS08 має сім регістрів для контролю його функцій. Нижче наведено опис основних регістрів та їх полів.

Регістр контролю модуля АЦП (ATD Control) ATD1C

Структура регістру контролю ATDC АЦП наведена на рисунку 16. Поля даного регістру дозволяють вмикати та вимикати АЦП, а, також, налаштовувати його основні параметри перед початком аналого-цифрового перетворення. Будь-який запис до регістру ATD1C припиняє поточне аналого-цифрове перетворення, якщо воно розпочате.

Bit	7	6	5	4	3	2	1	0
R W	ATDPU	DJM	RES8	SGN	PRS			
RESET	0	0	0	0	0	0	0	0

Рис.16. Структура регістру контролю АЦП ATD1C.

В таблиці 9 наведено опис полів регістру ATD1C.

Таблиця 9. Опис полів регістру ATD1C.

Поле	**Опис**
7 ATDPU	Біт вмикання/вимикання АЦП. (**ATD Power Up**), дозволяє програмно увімкнути/вимкнути АЦП. «Вимкнення» АЦП переводить його у режим зниженого енергоспоживання. 1 - вмикає АЦП; 0 - переводить АЦП в режим зниженого енергоспоживання.
6 DJM	Біт керування режимом вирівнювання 10–розрядних даних по двох 8-ми розрядних регістрах ATD1RH та ATD1RL **(Data Justification Mode).** У режимі вирівнювання даних праворуч (Right-justified Mode) молодший біт 10-розрядних даних розміщується в молодшому розряді ATD1RL0 регістру даних ATD1RL, а старший - в першому розряді ATD1RH1 регістру даних ATD1RH. У режимі вирівнювання ліворуч старший біт 10-розрядних даних розміщується в старшому

	розряді ATD1RH7 регістру даних ATD1RH, а молодший - в шостому розряді ATD1RL6 регістру даних ATD1RL. 1 – вмикає режим вирівнювання даних праворуч; 0 – вмикає режим вирівнювання даних ліворуч.
5 RES8	Біт встановлення розрядності АЦП. (**ATD Resolution Select**). 1 – вмикає 8-розрядний режим роботи АЦП; стан біту DJM ігнорується і 8-розрядні дані записуються в регістрі ATD1RH. 0 – вмикає 10-розрядний режим роботи АЦП.
4 SGN	Біт вмикання знакового/беззнакового представлення результату перетворення у регістрах АЦП (**Signed Result Select**). Може застосовуватись тільки для режиму вирівнювання даних ліворуч (DJM = 0). При встановленому біті 10-розрядні дані відображаються на діапазон -512 ($200) до 511($1FF), а 8-розрядні в діапазон -128($80) до 127($7F). В режимі вирівнювання даних праворуч (DJM = 1) біт SGN ігнорується. 1 – знакове представлення даних вирівняних ліворуч; 0 – беззнакове представлення даних вирівняних ліворуч.
3:0 PRS[3:0]	4-бітове поле, що дозволяє масштабувати тактову частоту перетворювача АЦП. (**Prescaler Rate Select**). Дію значень бітів даного поля при заданій тактовій частоті системної шини мікроконтролера ілюструє Таблиця 10.

Таблиця 10. Масштабування тактової частоти перетворення АЦП за допомогою маніпуляції з полем PRS регістру ATD1C.

PRS	**Factor = (PRS +1) × 2**	**Max Bus Clock MHz (2 MHz max ATD Clock)[1]**	**Max Bus Clock MHz (1 MHz max ATD Clock)[2]**	**Min Bus Clock[3] MHz (500 kHz min ATD Clock)**
0000	2	4	2	1
0001	4	8	4	2
0010	6	12	6	3
0011	8	16	8	4
0100	10	20	10	5
0101	12	20	12	6
0110	14	20	14	7
0111	16	20	16	8
1000	18	20	18	9
1001	20	20	20	10
1010	22	20	20	11
1011	24	20	20	12
1100	26	20	20	13
1101	28	20	20	14
1110	30	20	20	15
1111	32	20	20	16

[1] Максимальна частота тактування схеми перетворення АЦП - $maxf_{ATDCLK}$ = 2 МГц. Максимальна частота тактування контролера (max bus clock frequency – $maxf_{BUSCLK}$) розраховується як: $maxf_{BUSCLK} = maxf_{ATDCLK}$ (PRS+1)×2. Тобто, наприклад, для забезпечення максимальної частоти $maxf_{BUSCLK}$= 2МГц при частоті f_{ATDCLK} = 4МГц, дільник PRS має бути встановлений в 0.

[2] Використовуйте дані налаштування якщо максимальна бажана частота f_{ATDCLK}= 1МГц.

[3] При даних налаштуваннях та заданих частотах $maxf_{BUSCLK}$ мінімальна частота тактування f_{ADTCLK} складатиме 500 kГц.

Як можна зрозуміти з рисунку 15, таблиці 10 та приміток до неї, тактова частота перетворювача АЦП f_{ATDCLK} (максимальне значення 2 MHz) – це частота слідування тактових імпульсів схеми послідовного наближення АЦП. Значення f_{ATDCLK} залежить від тактової частоти контролера f_{BUSCLK} та налаштувань дільника PRS. f_{ATDCLK} визначає ширину часового вікна у межах якого відбувається одне повне перетворення в АЦП, яке займає 14 тактів ATDCLK.

Регістр стану та контролю модуля АЦП (Status and Control) ATD1SC

Структура регістру контролю та стану АЦП - ATD1SC наведена на рисунку 17, а опис його полів - у таблиці 11.

Bit	7	6	5	4	3	2	1	0
R	CCF	ATDIE	ATDCO	ATDCH				
W								
RESET	0	0	0	0	0	0	0	1

Рис.17. Регістр стану та контролю АЦП ATD1SC.

Запис у регістр стану і контролю АЦП ATD1SC скидає прапор CCF закінчення перетворення, скидає запити модуля на переривання (якщо вони були) та ініціює нове перетворення.

Таблиця 11 опис полів регістру ATD1SC.

Поле	Опис
7 CCF	Бітове поле прапору завершення перетворення (**Conversion Complete Flag**). Поле тільки для читання, яке встановлюється кожен раз, коли аналого-цифрове перетворення завершене Прапор, також, скидається при будь-якому записі у регістр стану і контролю АЦП ATD1SC, або якщо прочитано який-небуть з регістрів даних: ATD1RH або ATD1RL. 1 = поточне перетворення завершене;

	0 = перетворення ще не завершене.
6 ATDIE	Бітове поле дозволу апаратного переривання від АЦП (**ATD Interrupt Enabled**). Встановлення даного поля дозволяє генерувати АЦП запит на переривання по встановленню прапора завершення перетворення CCF. Запит на переривання здійснюоватиметься доти, доки не буте скинутий прапор CCF. Навіть якщо біт дозволу переривання АЦП не встановлений, опитування стану прапору CCF може бути використане для програмного визначення моменту закінчення перетворення у АЦП. 1 = переривання від АЦП дозволене; 0 = переривання від АЦП заборонене.
5 ATDCO	Бітове поле для налаштувацця АЦП на одноразове чи циклічне перетворення (**ATD Continuous Conversion**). В режимі одноразового перетворення кожне наступне перетворення необхідно ініціювати «вручну», шляхом оновлення (перезапису) регістру ATD1SC АЦП. При неперервному перетворенні, результати перетворення будуть оновлювати регістри даних АЦП кожен раз після завершення чергового перетворення. 1 = встановлення режиму неперервного перетворення; 0 = встановлення режиму одноразового перетворення.
4:0 ATDCH [4:0]	5 – бітне поле вибору каналу з аналоговим сигналом для аналого-цифрового перетворення (**Analog Input Channel Select**). Встановлення бітів поля керує мультиплексором. Таблиця 12 містить припустимі значення поля для вибору відповідних вхідних каналів.

Таблиця 12. Припустимі значення поля ATDCH регістру ATD1SC для вибору заданого вхідного каналу.

ATDCH	Analog Input Channel
00000	AD1P0 (пін 52)
00001	AD1P1 (пін 53)
00010	AD1P2 (пін 54)
00011	AD1P3 (пін 55)
00100	AD1P4 (пін 56)
00101	AD1P5 (пін 57)
00110	AD1P6 (пін 58)
00111	AD1P7 (пін 59)
$08–1D	Reserved (default to V_{REFL})
$1E	V_{REFH}
$1F	V_{REFL}

Регістр «дозволу» вхідних каналів АЦП (ATD Pin Enable) ATD1PE

8-ми розрядний регістр ATD1PE [7:0] призначений для підключення/відключення входів АЦП до/від відповідних виділених ліній збірки. Встановлення n-го біту регістру ATD1PE (ATD1PEn = 1) підключає

виділену лінію до входу AD1Pn АЦП. Скидання біту (ATD1PEn = 0) відключає відповідний вхід АЦП від виділеної лінії і її можна використовувати як лінію введення/виведення загального призначення, програмуючи порт B. Якщо вхід n АЦП відключено від виділеної лінії (ATD1PEn = 0), однак обраний як активний у регістрі ATDCH, АЦП робить аналого-цифрове перетворення напруги V_{REFL}, записуючи результуюче нульове значення до регістрів даних ATD1RH та ATD1RL.

Регістр даних-результатів перетворення АЦП (ATD Result Data) ATD1R

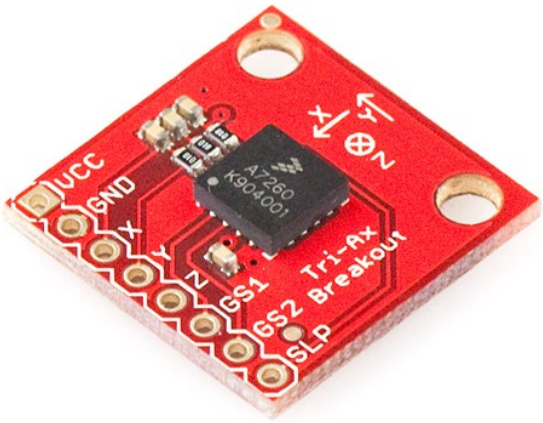
Після закінчення аналого-цифрового перетворення, результати перетворення записується у регістр ATD1R, що складається з дох 8-розрядних частин ATD1RH, ATD1RL. При налаштуванні 8-розрядного режиму роботи АЦП, результат перетворення записується тільки у регістр ATD1RL, незалежно від того - вирівнювання праворуч, чи ліворуч задане. Якщо налаштовано 10-розрядний режим роботи АЦП, при встановленні вирівнювання даних праворуч, молодші 8 розрядів (0-7) 10-розрядного результату перетворення записуються у регістр ATD1RL, а решта старших розрядів 8 та 9, записуються як 0 та 1 розряди в регістр ATD1RH, відповідно. Незадіяні розряди 2-7 регістру ATD1RH заповнюються нулями. При встановленні вирівнювання даних ліворуч, старші 8 розрядів (2-9) 10-розрядного результату перетворення записуються у регістр ATD1RH, а решта молодших розрядів 0 та 1, записуються як 6 та 7 розряди в регістр ATD1RL, відповідно. Незадіяні розряди 0-5 регістру ATD1RL заповнюються нулями.

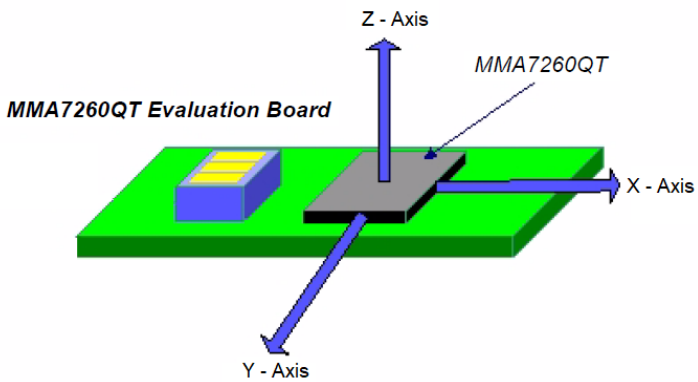
Джерела аналогових сигналів на тестувальній платі 13213-SRB.

Акселерометр MMA7260Q

Як видно з таблиці 1, на тестувальній платі 13213-SRB до входів АЦП [4:2] – AD1P4 - AD1P2 під’єднано виходи акселерометру MMA7260Q - Z-OUT, Y-OUT та X-OUT. На цих виходах генеруються аналогові сигнали, пропорційні прискоренню при русі плати вздовж відповідної просторової координати Z, Y

або X. Напрямки відповідних координатних осей у системі координат акселерометрів, зазвичай вказуються на платі, на які ці акселерометри монтуються, або у документаціях (див. наприклад рис.18).





а) б)

Рис.18. Варіанти відображення системи координат акселерометра а) безпосередньо на друкованій платі, б) в документації до плати.

Спрощена блок-схема акселерометра наведена на рис.19.

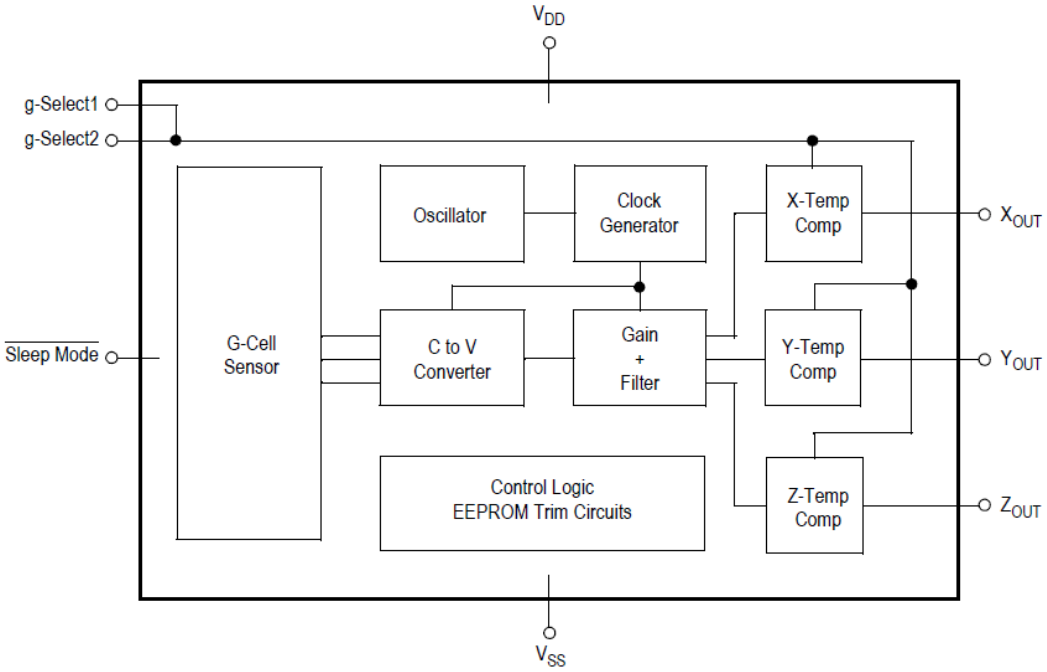


Рис.19. Спрощена блок-схема акселерометра MMA7260Q.

Частота оновлення вихідних сигналів акселерометра MMA7260Q визначається його внутрішнім генератором тактових імпульсів (Clock generator) і складає 11 кГц, тож не має сенсу опитувати виходи акселерометра з частотою, більшою за згадану. Лінії G-Select 1,2 – цифрові вхідні лінії, що дозволяють обирати робочі діапазони (чутливість) вимірюваних прискорень, під'єднано до ліній введення/виведення загального призначення PTB0 та PTB1 порту B збірки MC13213, відповідно (див. табл.1). У таблиці 13 наведено діапазони роботи датчика при відповідних налаштуваннях на лініях G – Select.

Таблиця 13. Вибір діапазонів прискорення (чутливості) акселерометру MMA7260Q.

G – Select 1	**G – Select 2**	**Робочий діапазон**	**Чуливість**
0	0	±1.5g	800mV/g
1	0	±2g	600mV/g
0	1	±4g	300mV/g
1	1	±6g	200mV/g

Лінія Sleep Mode (у таблиці 1 позначена як Power Save), також, є вхідною цифровою і під'єднана до лінії введення/виведення загального призначення PTB5, і, за необхідності, дозволяє переводити акселерометр у «сплячий» режим низького споживання. Низький рівень на лінії Sleep Mode (PTB5 = 0) переводить акселерометр у сплячий режим. За високого рівня на лінії Sleep Mode (PTB5 = 1) акселерометр працюватиме у нормальному режимі.

Акселерометр передбачає живлення від однополярного джерела з номінальною напругою у діапазоні V_{SS} = 2.2-3.6 В і дозволяє вимірювати додатні та від'ємні значення прискорень. Акселерометр відкаліброване так, що нульовим значенням прискорень по координатах відповідатиме вихідний сигнал $V_{OUT} = V_{SS}/2$. Максимальному, у обраному діапазоні, додатньому прискоренню відповідатиме вихідний сигнал $V_{OUT} = V_{SS}$, максимальному від'ємному - сигнал $V_{OUT} = 0$.

Датчик температури LM61B

До шостого входу АЦП (AD1P6) на тестувальній платі 13213-SRB під'єднано вихід аналогового датчику температури LM61B.

Вихідна напруга на виході Vout датчика LM61B змінюється від +300 мВ до +1600 мВ при зміні температури у діапазоні від -30°C до +100°C. Зміна вихідної напруги датчика лінійно пропорційна зміні температурі за Цельсієм (+10 мВ/°C). Датчик LM61B відкалібровано для забезпечення точності ±2,0°C при роботі у діапазоні кімнатних температур та ±3°C у повному діапазоні температур від -25°C до +85°C.

Низьке споживання датчика дозволяє живити його прямо від цифрових виходів контролерів та процесорів. Як видно з таблиці 1, на платі 13213-SRB вхід живлення V_{CC} датчика LM61B під'єднано до лінії загального призначення PTB7 порту B. Таким чином, високий рівень на лінії (PTB7 = 1) вмикатиме датчик, низький рівень (PTB7 = 1) – вимикатиме його. Більше інформації про акселерометр та датчик температури можна дізнатись з їх описів [4,5].

Завдання

1. Ознайомитись з особливостями будови модуля АЦП (ATD) збірки MC 13213 та з особливостями його роботи у різних режимах.
2. Ознайомитись з основними параметрами джерел аналогових сигналів на платі 13213-SRB - акселерометру MMA7260Q та датчику температури LM61B, та з їх підключенням до збірки MC13213 (табл.1).
3. Запрограмувати акселерометр на роботу в діапазоні 1.5g.
4. Запрограмувати увімкнення/вимкнення датчика температури за натисканням кнопки SW1/SW2 та увімкнення/вимкнення акселерометра за натисканням кнопки SW3/SW4. Факт увімкнення датчиків підтвердити запалюванням діодів LED1 та LED3, відповідно.
5. Запрограмувати АЦП мікросхеми MC 13213 у 10-розрядному режимі для послідовного зняття даних із виходів акселерометра та датчика температури, приблизно, раз на 0.1с.

6. Запрограмувати звуковий сигнал при перевантаженнях, що за модулем перевищують g вздовж будь-якої координати (за можливості, перевірити короткочасними рухами тестувальної плати у різних напрямках).
7. Запрограмувати блимання світлодіоду LED2 при зростанні температури та діоду LED4 при зниженні температури.

При виконанні завдань використання переривань не вимагається.

Порядок виконання роботи та методичні вказівки

1. Створити порожній проект у середовищі CodeWarrior у відповідності до інструкції у додатку 1.
2. Написати програму для виконання завдання 4.
3. Підключити програматор до порту BDM тестувальної плати (за необхідності скористайтесь допомогою викладача).
4. Підключіть програматор до USB порту комп'ютера.
5. Підключіть живлення до тестувальної плати та увімкніть її за допомогою відповідного перемикача (далі до закінчення роботи даний перемикач можна використовувати для вимкнення/вмикання тестової плати).
6. Послідовно створіть та налагодьте програми, що реалізують дії, описані у завданнях 4-8.
7. Підготуйте та здайте звіт.

Вказівки для роботи з АЦП

1) Налаштовуємо АЦП на 10-розрядний режим роботи з частотою перетворення 2 МГц, з вирівнюванням беззнакового результату перетворення ліворуч:

```
ATD1C_ATDPU = 1; ATD1C_DJM = 0; ATD1C_RES8 = 0;
ATD1C_SGN=0; ATD1C_PRS = 0x0;
```

2) Підключаємо до 4 каналу АЦП AD1P4 виділену лінію:

```
ATD1PE_ATDPE4 = 1;
```

3) Дозволяємо переривання від АЦП, задаємо режим одиночного перетворення, обираємо як джерело аналогового сигналу вхід 4 АЦП та одночасно запускаємо процес перетворення в АЦП:

```
ATD1SC = 0b01000100; //ATDIE = 1; ATDCO=0x0; ATDCH = 0x4;
```

4) Читаємо 10 розрядний результат перетворення, вирівняний ліворуч, з регістрів даних АЦП, що одночасно призводить до скидання прапору закінчення перетворення. Одразу зшиваємо отриманий результат у змінну RES_10_0:

```
RES_10_0 = (ATD1RH<<2) | (ATD1RL >>6);
```

5) Приклад визначення обробника переривань від модуля АЦП **Atd1I_OnConvReady**, у якому результат перетворення зчитується і одразу зшивається у змінній RES_10_0. Також в обробнику передбачено можливість запуску наступного акту перетворення.

Приклад 7

```
/*
** ==================================================================
**    Метод      :  Atd1I_OnConvReady
**    Обробник переривань від модуля ATD
**    VectorNumber_Vatd1 - ім'я номера вектору переривань з таблиці векторів
**    переривань (див. файл заголовку mc13213.h)
**    Переривання модулем буде генеруватись при завершенні аналого-цифрового
**    перетворення у модулі ATD
** ==================================================================
*/
int RES_10_0;

interrupt VectorNumber_Vatd1 void Atd1I_OnConvReady(void)
{ RES_10_0 = (ATD1RH<<2) | (ATD1RL >>6); /*читаємо результат перетворення у
                                          змінну RES_10_0, одночасно упорядковуємо
                                          біти результату у потрібні місця та скидаючи
                                          прапор CCF (при зчитуванні регістрів даних
                                          прапор CCF скидається автоматично)*/
/*ATD1SC = …… - якщо необхідно, запускаємо новий цикл перетворення зі старими/новими
налаштуваннями у регістрі ATD1SC  */
} /* Atd1I_OnConvReady */
```

Контрольні питання

1. Назвіть основні характеристики АЦП, що використовується в роботі.
2. Назвіть регістри керування АЦП, що використовується в роботі, та призначення їх полів.
3. Які режими представлення даних в регістрах даних АЦП?
4. Які лінії мікросхеми MC13213 використовуються як аналогові входи АЦП в даній роботі?
5. Як вмикається/вимикається датчик температури та акселерометр в даній роботі?
6. У яких межах слід очікувати цифрові значення на виході АЦП у 10-розрядному режимі роботи для температурного датчика та акселерометра?

Рекомендована література

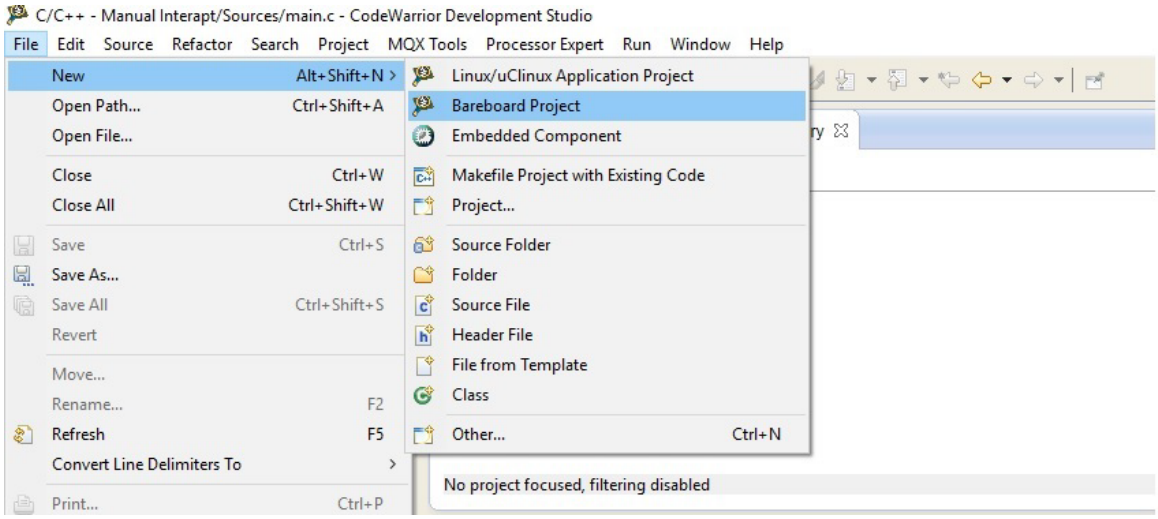
1. MC13211/212/213 ZigBee ™- Compliant Platform - 2.4 GHz Low Power Transceiver for the IEEE® 802.15.4 Standard plus Microcontroller/ Technical Data. https://www.nxp.com/docs/en/data-sheet/MC1321x.pdf
2. MC1321x Evaluation Kit (EVK) Reference Manual - MC1321xEVKRM, https://www.nxp.com/docs/en/reference-manual/MC1321xEVKRM.pdf
3. Freescale Semiconductor. Application Note AN2616. Getting Started with HCS08 and CodeWarrior Using C (Revised for CodeWarrior v3.1 → v5.1). by: Stephen Pickering, East Kilbride. Scotland. https://www.nxp.com/docs/en/application-note/AN2616.pdf
4. MMA7260Q: XYZ AXIS ACCELEROMETER, ±1.5g/2g/4g/6g https://www.nxp.com/docs/en/data-sheet/MMA7260QT.pdf
5. LM61 ±3°C analog output temperature sensor https://www.ti.com/lit/ds/symlink/lm61.pdf?ts=1704324396135&ref_url=https%253A%252F%252Fbr.mouser.com%252F
6. Мікропроцесорні та мікроконтролерні системи : підручник. У 2 ч. Ч. 1. Мікропроцесорні системи [Електронний ресурс] / А. О. Новацький. – Електронні текстові дані (1 файл: 43,8 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 367 с.
7. Мікропроцесорні та мікроконтролерні системи: Ч.2 «Проектування мікропроцесорних систем» [Електронний ресурс] : підручник для студ. освітньої програми «Інтегровані інформаційні системи» за спеціальністю 126 «Інформаційні системи та технології» / А.О. Новацький : КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 20,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 460 с.

ДОДАТОК 1

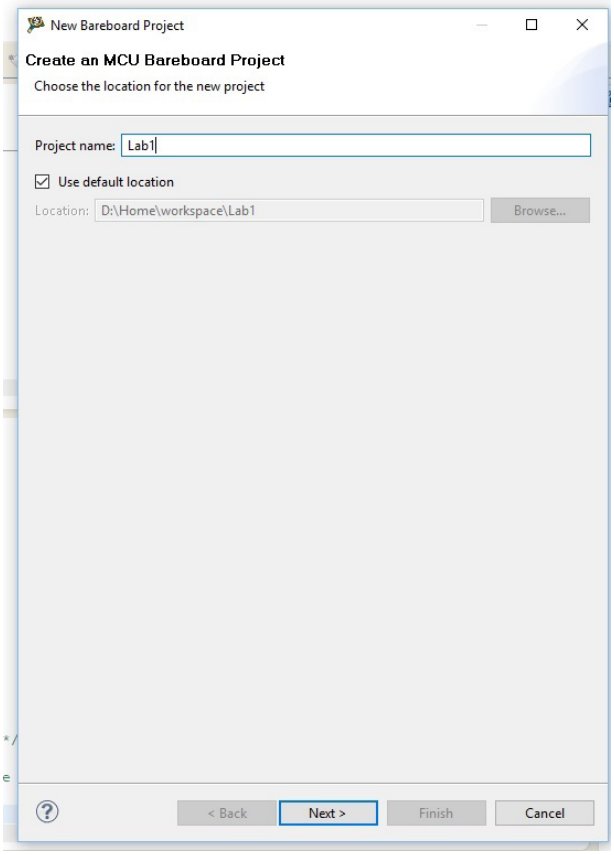
Основи роботи з середовище розробки CodeWarrior в ОС Windows

1. Створення нового проекту

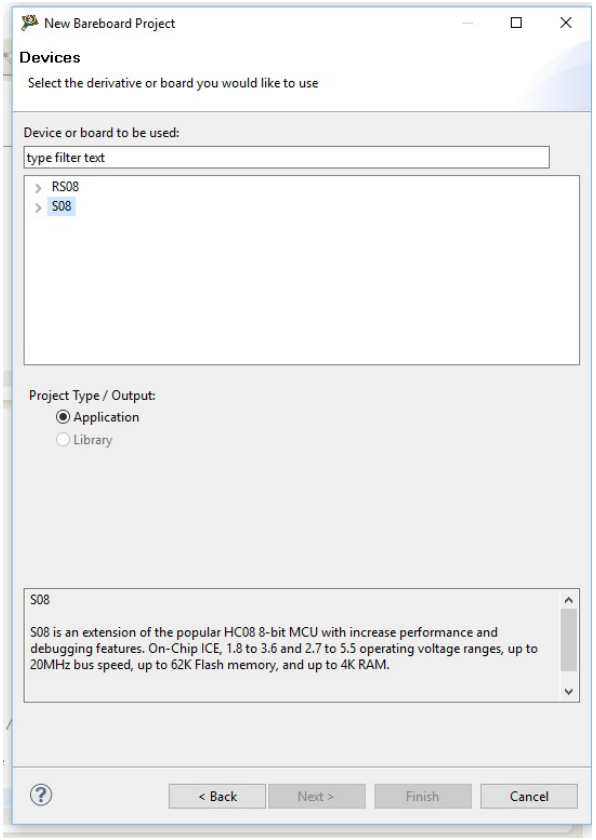
1.1.На вкладці *File* обрати *New -> Bareboard Project*



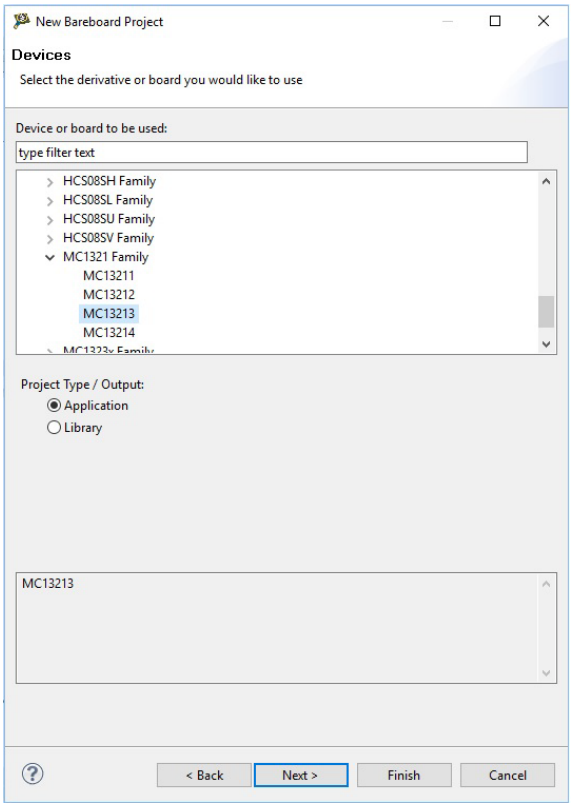
1.2.У вікні *New Bareboard Project* ввести ім'я поетку та натиснути *Next*



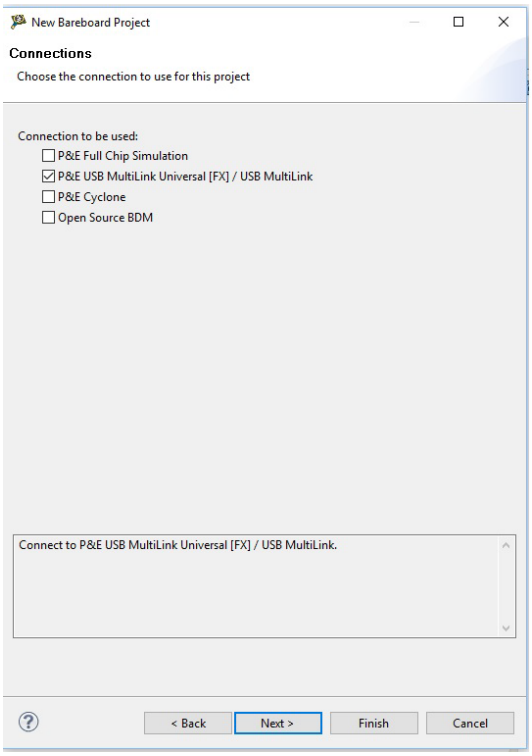
1.3.В розділі *Devices* обрати тип контролера *S08* та натиснути *Next*



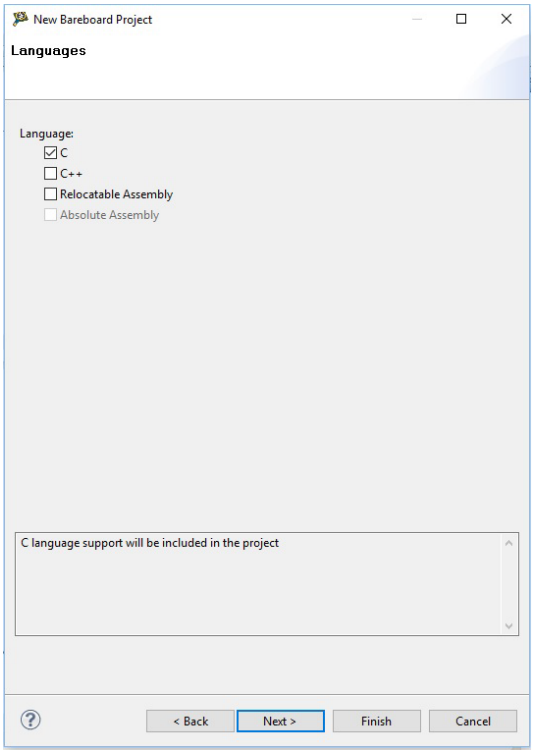
1.4. Розгорнути список сімейства збірок *MC1321 Family* та обрати *MC13213*, натиснути *Next*



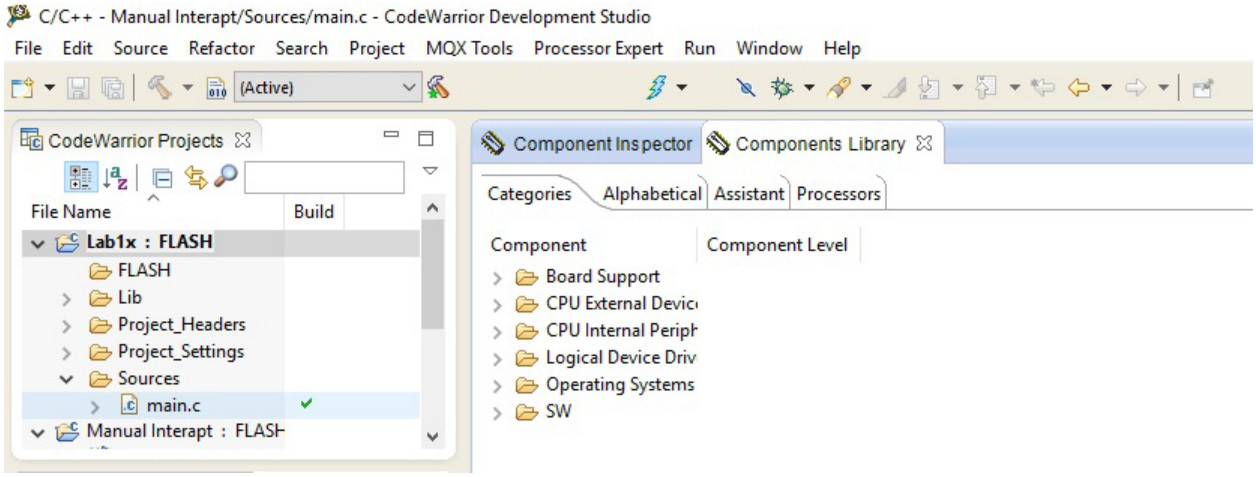
1.5. В розділі *Connections* обрати *P&E USB MultiLink Universal [FX]/USB MultiLink* та натиснути *Next*



1.6. В розділі *Language* обрати *C* та натиснути *Finish*



1.7. У папці *Source*, створеного проекту, відкрити файл *main.c*. Це основний файл проекту, у якому можна писати код програми.



Типовий вигляд новоствореного файлу *main.c*:

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "mc13213.h"
#include "stdio.h"

void main(void) {
      EnableInterrupts;
  for(;;) {
     __RESET_WATCHDOG();    /* feeds the dog */
  } /* loop forever */
  /* please make sure that you never leave main */
}
```

2. Створення програми. Компіляція проекту у CodeWarrior та запис виконавчого коду до збірки MC13213.

2.1. Приклад програми для тестувальної плати 13213-SRB, у якій по замиканню кнопки 1 запалюється світлодіод 1. Розімкнена кнопка гасить діод. (Додаткові приклади у [3])

```
#include <hidef.h> /* for EnableInterrupts macro */
#include "derivative.h" /* include peripheral declarations */
#include "mc13213.h"
#include "stdio.h"

#define LED1 PTDD_PTDD4 //макрос LED1 для зручності
                        //звернення до лінії 4 порту D, до якої
                        //підключено світлодіод 1
#define SW1 PTAD_PTAD2  //SW1 для зручності
                        //звернення до лінії 2 порту A, до якої
                        //підключено кнопку 1
```

```
#define ON 0            //макроси для покращення читабельності
#define OFF 1           //програми
#define UP 0            //макроси для покращення читабельності
#define DOWN 1           //програми

void main(void)
{
     EnableInterrupts;  /* enable interrupts */
     PTDDD_PTDDD4 = 1;  //налаштовуємо лінію 4 порту D на виведення
     PTADD = 0b00000000 //налаштовуємо одночасно усі лінії порту A
                        //на введення.
     PTAPE_PTAPE2 = 1;  //вмикаємо резистор підтягування лінії 2
                         //порту A, щоб при розімкненій кнопці на
                         //лінії 2 порту був високий рівень
     LED_1 = OFF;        //вимикаємо діод.
  for(;;)
     {
          LED1 = SW1; //вмикаємо/вимикаємо світлодіод кнопкою

       __RESET_WATCHDOG(); //годуємо песика, щоб контролер
                            // не перезавантажувався
     } /* loop forever */
} //end main()
```

2.2. Компіляція проекту у CodeWarrior та запис виконавчого коду до збірки MC13213.

Знайдіть у меню CodeWarrior кнопку відлагодження (Debug button) . Розкрийте список доступних варіантів відлагодження. Запустіть виконання проекту. В результаті проект буде зкомпільовано та, у разі відсутності помилок, налагоджувач/програматор *P&E USB MultiLink* завантажить проект у пристрій. Про результати завантаження середовище CodeWarrior повідомить у відповідному вікні.

ДОДАТОК 2

Таблиця 1. Вектори переривання мікроконтролера HCS08 та їх опис

Номер вектору	Адреса	Ім'я вектору	Модуль	Прапор запиту	Біт маски	Опис
26-31	$FFC0 - $FFCA	Unused Vector Space (available for user program)				
25	$FFCC	Vrti	System control	RTIF	RTIE	Real-time interrupt
24	$FFCE	Viic1	IIC	IICIS	IICIE	IIC control
23	$FFD0	Vatd1	ATD	CCF	AIEN	AD conversion complete
22	$FFD2	Vkeyboard1	KBI	KBF	KBIE	Keyboard pins
21	$FFD4	Vsci2tx	SCI2	TDRE TC	TIE TCIE	SCI2 transmit
20	$FFD6	Vsci2rx	SCI2	IDLE RDRF	ILIE RIE	SCI2 receive
19	$FFD8	Vsci2err	SCI2	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI2 error
18	$FFDA	Vsci1tx	SCI1	TDRE TC	TIE TCIE	SCI1 transmit
17	$FFDC	Vsci1rx	SCI1	IDLE RDRF	ILIE RIE	SCI1 receive
16	$FFDE	Vsci1err	SCI1	OR NF FE PF	ORIE NFIE FEIE PFIE	SCI1 error
15	$FFE0	Vspi1	SPI	SPIF MODF SPTEF	SPIE SPIE SPTIE	SPI
14	$FFE2	Vtpm2ovf	TPM2	TOF	TOIE	TPM2 overflow
13	$FFE4	Vtpm2ch4	TPM2	CH4F	CH4IE	TPM2 channel 4
12	$FFE6	Vtpm2ch3	TPM2	CH3F	CH3IE	TPM2 channel 3
11	$FFE8	Vtpm2ch2	TPM2	CH2F	CH2IE	TPM2 channel 2
10	$FFEA	Vtpm2ch1	TPM2	CH1F	CH1IE	TPM2 channel 1
9	$FFEC	Vtpm2ch0	TPM2	CH0F	CH0IE	TPM2 channel 0
8	$FFEE	Vtpm1ovf	TPM1	TOF	TOIE	TPM1 overflow
7	$FFF0	Vtpm1ch2	TPM1	CH2F	CH2IE	TPM1 channel 2
6	$FFF2	Vtpm1ch1	TPM1	CH1F	CH1IE	TPM1 channel 1
5	$FFF4	Vtpm1ch0	TPM1	CH0F	CH0IE	TPM1 channel 0
4	$FFF6	Vicg	ICG	ICGIF (LOLS/LOCS)	LOLRE/LOCRE	ICG

3	$FFF8	Vlvd	System control	LVDF	LVDIE	Low-voltage detect
2	$FFFA	Virq	IRQ	IRQF	IRQIE	IRQ pin
1	$FFFC	Vswi	Core	SWI Instruction	—	Software interrupt
0	$FFFE	Vreset	System control	COP LVD RESET pin Illegal opcode	COPE LVDRE — —	Watchdog timer Low-voltage detect External pin Illegal opcode

Зміст

Вступ 4
Загальні відомості про вбудовані системи. 5
Мікроконтролерна збірка сімейства MC1321x 6
Мікроконтролери сімейства HCS08 9
Тестувальна плата MC13213-Sensor Reference Board 12
Обладнання, документація та програмне забезпечення 14
Лабораторна робота № 1. Програмування ліній введення/виведення загального призначення (GPIO) 16
Теоретичні відомості 16
Лінії введення/виведення загального призначення GPIO збірки MC13213 16
Елементи базової програми на мові С для збірок сімейства MC1321x 19
Завдання 21
Порядок виконання роботи та методичні вказівки 22
Контрольні питання 22
Лабораторна робота № 2. Програмування дій за натисканням кнопок 23
Теоретичні відомості 23
Завдання 24
Порядок виконання роботи та методичні вказівки 25
Контрольні питання 25
Лабораторна робота № 3. Модуль переривань клавіатури KBI 27
Теоретичні відомості 27
Переривання 27
Модуль сканування переривань клавіатури KBI (Keyboard Interrupt) 31
Завдання 36
Порядок виконання роботи та методичні вказівки 37
Контрольні питання 37
Лабораторна робота № 4. Модуль таймера/широтно-імпульсного модулятора 39
Теоретичні відомості 39
Загальні відомості про модулі таймера/широтно-імпульсного модулятора ТРМ мікроконтролера HCS08 39
Регістр стану та керування модуля TPMx (Timer x Status and Control Register) TPMxSC. 41
Джерела тактування збірки MC13213 42

Вибір джерела тактування та дільника частоти тактування модуля TPM .. 44
Регістр лічильника модуля TPMx (Timer x Counter Registers) TPMxCNT 46
Модуль-регістр модуля TPMx (Timer x Counter Modulo Registers) TPMxMOD. 46
Регістр значень каналу n модуля TPMx (Timer x Channel n Value Registers) TPMxCnV. 47
Регістр стану та керування каналу n модуля TPMx (Timer x Channel n Status and Control Register) TPMxCnSC. 48
Завдання..... 50
Порядок виконання роботи та методичні вказівки 51
Контрольні питання 53
Лабораторна робота № 5. Модуль аналого-цифрового перетворювача ATD..... 54
Теоретичні відомості..... 54
Загальні відомості про модуль аналого-цифрового перетворення ATD мікроконтролера HCS08 54
Регістр контролю модуля АЦП (ATD Control) ATD1C..... 56
Регістр стану та контролю модуля АЦП (Status and Control) ATD1SC..... 58
Регістр «дозволу» вхідних каналів АЦП (ATD Pin Enable) ATD1PE 59
Регістр даних-результатів перетворення АЦП (ATD Result Data) ATD1R 60
Джерела аналогових сигналів на тестувальній платі 13213-SRB. 60
Акселерометр MMA7260Q 60
Датчик температури LM61B 63
Завдання..... 63
Порядок виконання роботи та методичні вказівки 64
Контрольні питання 66
Рекомендована література 67
ДОДАТОК 1 68
ДОДАТОК 2 72

Навчальне видання

Коломієць *Іван Сергійович*
Оберемок *Євген Анатолійович*

ПРОГРАМУВАННЯ ВБУДОВАНИХ СИСТЕМ

Методичний посібник з виконання лабораторних робіт
для студентів факультету радіофізики, електроніки
та комп'ютерних систем

Підписано до друку _____. Формат $60x80^{16}$.
Гарнітура Times. Папір офсетний. Друк офсетний.
Наклад 30 примірників. Ум. друк. арк. 4.
Видавнича лабораторія факультету радіофізики,
електроніки та комп'ютерних систем
Київського національного університету імені Тараса Шевченка

2024