

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ ДЕРЖАВНИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ

Для спеціальності:
5.151 "Автоматизація та комп'ютерно-інтегровані технології"
спеціалізації:
5.151.1 "Обслуговування інтегрованих інтелектуальних систем"

Методичний посібник

для самостійної роботи
з навчальної дисципліни

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІС



2019

Методичний посібник для самостійної роботи з дисципліни
“Програмне забезпечення ІС” для студентів освітньо-професійної програми
«Обслуговування інтелектуальних інтегрованих систем» спеціальності 151
«Автоматизація та комп’ютерно-інтегровані технології»

Укладач: О.А. Задара - Харків: ХДПК, 2019, 53стр.

Затверджено на засіданні циклової комісії інформаційних технологій

Протокол від _____ 2019р. № _____

Голова циклової комісії _____ М.М. Бочарніков

“ _____ ” _____ року

Схвалено методичною радою коледжу

Протокол від _____ 2019р. № _____

Голова методичної ради _____

“ _____ ” _____ 2019року

ЗМІСТ

ВСТУП

Модуль 1. Мікроконтролери PIC

Тема 1. Сімейства мікроконтролерів PIC	4
Тема 2. Фірми-виробники мікроконтролерів.....	5
Тема 3. Організація пам'яті даних мікроконтролерів PIC.....	8
Тема 4. Характеристики мікроконтролера PIC16F84A.....	9
Тема 5. Набір команд мікроконтролера PIC16F84A.....	12
Тема 6. Види та характеристики світлодіодів. Принцип дії.....	14
Тема 7. Спеціальні функції Assembler.....	16
Тема 8. Семисегментні індикатори. Характеристики та принцип дії.....	20

Модуль 2. Мікроконтролери AVR

Тема 9. Технічна документація ATmega8 (Datasheet)	24
Тема 10. Основні директиви препроцесора Cі.....	27
Тема 11. Створення функцій на мові програмування Cі.....	29
Тема 12. Датчики вимірювання. Характеристики датчиків.....	31
Тема 13. Характеристики рідкокристалічних дисплеїв.....	33
Тема 14. Алгоритм включення 4-бітного режиму рідкокристалічного дисплею	
Тема 15. Мікроконтролери серії ATtiny.....	36

Модуль 3. Мікроконтролери STM32

Тема 16. Архітектура мікроконтролерів STM32.....	37
Тема 17. Лінійки сімейства мікроконтролерів STM32.....	41
Тема 18. Порівняльна характеристика STM32 та AVR.....	42
Тема 19. Регістри вводу-виводу STM32.....	43
Тема 20. Статичні характеристики портів вводу-виводу STM32.....	45

Модуль 4. Пристрої з числовим програмним управлінням

Тема 21. Методи програмування пристроїв з ЧПУ.....	47
Тема 22. Системи координат.....	48
Тема 23. Елементи управління Fanuc 0iT	52

ТЕМА 1. СІМЕЙСТВА МІКРОКОНТРОЛЕРІВ PIC

На теперішній час фірма Microchip випускає п'ять сімейств мікроконтролерів PIC:

1. **PIC16 C5X** включає недорогі контролери з мінімальним набором периферійних пристроїв:

2. **PIC12Cxxx** містить МК в мініатюрному 8-вивідному корпусі з вбудованим тактовим генератором і модулем 8-розрядного АЦП (для деяких моделей);

3. **PIC16x/7x/8x/ 9x** об'єднує МК з розвиненою периферією (таймери-лічильники з опціями захоплення/порівняння, широтно-імпульсні модулятори, аналогові компаратори, АЦП, контролери послідовних інтерфейсів);

4. **PIC17C4x/5xx** включає високопродуктивні МК з розширеною системою команд і великої периферією. Мікроконтролери цього сімейства мають вбудований апаратний помножувач 8 г 8, що виконує операцію множення за один машинний цикл;

5. **PIC18Cxxx** - нове сімейство з оптимізованим під використання Cі-компілятора RISC -ядро і частотою внутрішньої шини до 10 МГц.

Споживачам надаються три моделі мікроконтролерів сімейства PIC з різними поєднаннями характеристик і властивостей.

Молодші моделі

До них відносяться МК серії 16C5x і 8-вивідні МК серії 12C5xx, які реалізують нижній рівень можливостей мікроконтролерів сімейства PICMicro і використовуються для створення невеликих і простих додатків. Вони виконують більшість команд старших моделей і програмно сумісні з ними. Молодші моделі зіграли велику роль у розвитку сімейства PICMicro.

Однак обмежені можливості обробки переривань, малий обсяг доступної пам'яті програм і RAM, відсутність внутрісистемного програмування і портів введення /виводу з розширеними функціями, зниження вартості технологічних витрат з'явилися причинами переходу до моделей мікроконтролерів середнього рівня, які продавалися за такою ж (і навіть нижчою) ціні, як і МК молодших моделей.

Середні моделі

Моделі середнього рівня утворюють закінчену лінію мікроконтролерів (серія 16Cxx) з безліччю різних функцій і отримали найбільш широке поширення. Вони мають таку ж базову архітектуру, як мікроконтролери молодших моделей, але істотно відрізняються можливостями обробки переривань.

Старші моделі

Мікроконтролери старших моделей (серії 17Схх) розроблялися для взаємодії з іншими цифровими пристроями. Тому в них відсутні наявні в моделях середнього рівня АЦП і пристрої для підключення датчиків. У контролерах використовуються 16-бітові команди, здатні адресувати по 128К байт пам'яті програм і пам'яті даних (64К слів по 16 біт).

Для читання / запису пам'яті програм в МК використовується модифіковане ядро центрального процесора, що дозволяє різним командам звертатися до всіх регістрів процесора. Це підвищує гнучкість мікроконтролерів і розширює області їх застосування. Молодші та середні моделі такою можливістю не володіють. Крім того, в старших моделях PICMicro є кілька векторів переривань, в той час як в мікроконтролерах середнього рівня використовується тільки один вектор переривання.

Слід зазначити, що різний апаратний склад і характеристики мають не тільки мікроконтролери різних сімейств, а й моделі (версії) одного сімейства. Однак загальним для більшості моделей контролерів є:

RISC-архітектура (з двоступінчастим конвеєром), що забезпечує виконання більшості команд процесора за один машинний цикл;

Гарвардська архітектура (з роздільними шинами даних і програм), що забезпечує одночасний доступ до пам'яті даних і програм;

КМОП технологія, що забезпечує:

- повністю статичний режим роботи, при якому зупинка тактового генератора не приводить до втрати логічних станів внутрішніх вузлів;
- широкий діапазон живлення (2 - 6 В) і температур (-40 + 70 ° С);
- мале енергоспоживання;

Питання для самоперевірки:

1. Які з моделей мікроконтролерів PIC орієнтовані на використання мови високого рівня?
2. Які з мікроконтролерів PIC мають розвинену периферію?
3. Які відмінності мікроконтролерів PIC середніх та старших моделей?
4. Які загальні характеристики у більшості моделей PIC мікроконтролерів?

ТЕМА 2. ФІРМИ-ВИРОБНИКИ МІКРОКОНТРОЛЕРІВ

Microchip Technology - американська компанія, один зі світових лідерів в області виробництва електронних компонентів.

Крім мікроконтролерів Microchip пропонує широкий спектр інших цифрових і аналогових рішень: незалежну пам'ять, мікросхеми програмованої логіки, компоненти для тактування, інтерфейсні рішення, розширювачі портів

введення-виведення, драйвери дисплеїв і світлодіодів, компоненти для бездротового зв'язку, мікросхеми з блоками криптографії, сенсорні рішення, а також операційні підсилювачі, компаратори, АЦП і ЦАП, датчики температури і диму, ультразвукові рішення, драйвери двигунів, високовольтні інтерфейси і рішення для управління живленням.

У 2016 році компанія Microchip придбала фірму **Atmel**, це дозволило Microchip додати в свою програму поставки мікроконтролерів з процесорними ядрами AVR і ARM.

Сайт виробника: www.microchip.com

STMicroelectronics - одна з найбільших в світі компаній, що займаються виробництвом інтегральних мікросхем. Річний дохід її в 2011 році склав 9,73 млрд. Доларів. Компанія має найширший в галузі номенклатурний портфель продукції. Асортимент продукції компанії включає більше 3000 різних найменувань. Це компоненти для зв'язку і телекомунікацій, побутової та промислової електроніки, комп'ютерів і автомобільної галузі.

STMicroelectronics є адресні розробки під конкретне використання. Це системи SoC і SiP на кристалі і в корпусі, чіпи для Set-Top-Box, КМОП-відеокамери, кріптопам'ять для смарт-карт. Такі пристрої збираються під замовлення, і в кожному випадку приймається індивідуальне рішення про доцільність розробки і підтримки.

Сайт виробника www.st.com

Корпорація Nuvoton Technology (NTC) заснована і зареєстрована в Тайвані. Вона створювалася для привнесення інноваційних рішень в сферу аналогових / змішаних сигналів на ринку напівпровідників. Почавши роботу в якості дочірньої компанії Winbond Electronics, Nuvoton Technology в 2008 році відкрила свій офіс і з цього моменту присутній на ринку мікроелектроніки з власним брендом. Тим часом Nuvoton зберегла всі зв'язки з Winbond і успадкувала головна конкурентна перевага - низьку вартість пам'яті.

У складі Nuvoton три підрозділи: мікроконтролери і мікросхеми для аудіо, рішення для хмарних і обчислювальних систем (включаючи персональні комп'ютери), а також власне виробництво пластин. У 2020 році Nuvoton планує придбати напівпровідниковий бізнес Panasonic (контролери живлення і датчики для автоелектроніки, смартфонів і систем безпеки).

Сайт виробника: www.nuvoton.com

Американська фірма Silicon Labs (SiLabs), головний офіс якої знаходиться в Остіні, штат Техас, була заснована в 1998 році і в даний час є одним зі світових лідерів в розробці технологічних і архітектурних рішень мікроелектроніки для спільної обробки аналогових і цифрових сигналів. Silicon Labs випускає широкий спектр продукції спеціального і загального призначення, які використовуються в побутовій техніці, індустріальної і автомобільної промисловості, в телекомунікаційних системах та офісному обладнанні

Крім 8- і 32-розрядних мікроконтролерів Silicon Labs випускає рішення для управління живленням і гальванічної розв'язки, мікросхеми для

радіомовлення і бездротових телекомунікацій, генератори і гуни, мікросхеми для провідних комунікацій та мікросхеми вбудованих дротових модемів ISModem.

Сайт виробника: www.silabs.com

Компанія WIZnet (Південна Корея) займається виробництвом і постачанням унікальних апаратних рішень класу Embedded Internet і Embedded Ethernet. Мікросхеми WIZnet застосовуються для організації віддаленого доступу, в системах IP-телефонії і IP-телебачення, в системах охорони, безпеки і контролю доступу, а також інших додатків, що використовують для комунікації провідні канали зв'язку і мережі Ethernet з розвиненим стеком протоколів TCP / IP.

Основною продукцією компанії Wiznet є інтегральні мікросхеми мережних контролерів: апаратні Ethernet-мости і спеціалізовані мікроконтролери. В останніх на одному кристалі об'єднуються апаратний блок Ethernet-моста з вбудованими рівнями MAC і PHY і процесорне ядро ARM або MSC-51.

Сайт виробника: www.wiznet.co.kr

Компанія FTDI спеціалізується на виробництві мікросхем сполучення мікропроцесорних пристроїв з USB. Рішення FTDI дозволяють організувати обмін даними по USB найпростішим чином завдяки апаратної реалізації протоколу та наявності безкоштовних драйверів для ОС Windows, MAC OS, Linux, Android.

На кристалах FTDI реалізуються мости між USB і такими мікропроцесорними інтерфейсами, як UART, FIFO, JTAG, SPI, PS / 2, I2C, IrDA. У режимі "Bit Bang" вони можуть використовуватися для конфігурації мікросхем програмованої логіки через USB або для введення / виведення з USB цифрових логічних сигналів без використання додаткового мікроконтролера.

Сайт виробника: www.ftdichip.com

Компанія IAR Systems® більше 20 років займається розробкою інтегрованих засобів програмування. Сьогодні IAR Systems® є одним з визнаних світових лідерів з постачання C / C ++ компіляторів для вбудованих мікропроцесорних пристроїв і систем на базі 8051, ARM, AVR, MSP430, HC11 / HC12, PIC, Z80 і багатьох інших 8-, 16-, 32- розрядних процесорів.

Сайт виробника: www.iar.com

Питання для самоперевірки:

1. Яка з компаній придбала виробника мікроконтролерів AVR?
2. Яка з компаній має найбільший спектр виробів у галузі мікроелектроніки?
3. Яка з компаній орієнтована на виробництво продукції для мережевого обладнання?
4. Між якими інтерфейсами реалізуються мости USB від FTDI?

Щоб звернутися до якогось регістру (виконати операцію над ним), необхідно спочатку вказати банк, в якому розташований цей регістр. Банк вибирається за допомогою бітів RP1, RP0 регістра STATUS, який визначений у всіх банках, тобто до регістру STATUS можна звертатися незалежно від того який банк обраний, що є очевидним, інакше ми не змогли б перемикає банки. Деякі важливі регістри спеціального призначення також визначені відразу в декількох банках. Регістри ОЗУ також можуть бути визначені в декількох банках, на карті пам'яті даних

Для реалізації непрямої адресації використовується регістр спеціального призначення FSR, а також фізично не реалізований регістр INDF. Щоб звернутися до будь-якого регістру (наприклад PORTB), необхідно записати його адресу в регістр FSR, після чого вміст регістра PORTB умовно виявиться в регістрі INDF. Умовно тому що звернення до регістру INDF насправді викличе дію з регістром PORTB.

Непряму адресацію зручно застосовувати при зверненні до великої кількості регістрів розташованих послідовно в пам'яті даних. Виконуючи однотипну операцію з регістром INDF, з подальшим інкрементом регістра FSR, можна прооперувати велику кількість регістрів. При використанні непрямої адресації також слід вказати банк місцезнаходження регістру, адреса якого записують в FSR. При даній адресації банк вказується одним бітом IRP регістра STATUS, так як адресація працює в межах 2-х банків.

Пам'ять даних МК розбита на дві області. Перші 12 адрес - це область регістрів спеціальних функцій (SFR), а друга - область регістрів загального призначення (GPR). Область SFR керує роботою приладу.

Питання для самоперевірки:

1. На які ділянки поділяється пам'ять МК PIC16F84A?
2. Для чого використовуються регістри спеціального та загального призначення?
3. Яка різниця між першим та другим банками пам'яті?
4. Для чого використовується регістр STATUS?
5. Які існують способи адресації і як вони реалізуються?

ТЕМА 4. ХАРАКТЕРИСТИКИ МІКРОКОНТРОЛЕРА PIC16F8X

Мікроконтролери підгрупи PIC16F8X відносяться до сімейства 8-розрядних КМОН мікроконтролерів групи PIC16CXXX, для яких характерні низька вартість, цілком статична КМОН-технологія і висока продуктивність. До складу підгрупи входять МК PIC16F83, PIC16CR83, PIC16F84 і PIC16CR84. Основні характеристики МК підгрупи PIC16F8X наведені в табл. 4.1

Параметр	PIC16F83	PIC16CR83	PIC16F84	PIC16CR84
----------	----------	-----------	----------	-----------

Максимальна частота, МГц	10	10	10	10
Flash-пам'ять програм, слів	512	-	1К	-
ПЗП програм, слів	-	512	-	1К
Пам'ять даних, байт	36	36	68	68
Пам'ять даних у РПЗП (EEPROM), байт	64	64	64	64
Таймери	TMR0	TMR0	TMR0	TMR0
Кількість джерел переривань	4	4	4	4
Кількість ліній вводу/виводу	13	13	13	13
Діапазон напруг живлення, В	2,0 – 6,0	2,0 – 6,0	2,0 – 6,0	2,0 – 6,0
Кількість виводів і тип корпусу	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC	18 DIP, SOIC

Усі мікроконтролери підгрупи PIC16F8X використовують гарвардську архітектуру з RISC-процесором, що володіє наступними основними особливостями:

- використовуються тільки 35 простих команд;
- усі команди виконуються за один цикл (400 нс при частоті 10 МГц), крім команд переходу, які вимагають 2 цикли;
- робоча частота 0 Гц ... 10 МГц;
- роздільні шини даних (8 біт) і команд (14 біт);
- 512x14 або 1024x14 пам'ять програм, виконана на ПЗП або електрично перепрограмованій Flash- пам'яті;
- 15 восьмирозрядних регістрів спеціальних функцій (SFR);
- восьмирівневий апаратний стек;
- пряма, непряма і відносна адресація даних і команд;
- 36 або 68 восьмирозрядних регістрів загального призначення (GPR) або ОЗП;
- чотири джерела переривання:
- зовнішній вхід RB0/INT;
- переповнення таймера TMR0;
- зміна сигналів на лініях порту В;
- завершення запису даних у пам'ять EEPROM;
- 64 x 8 електрично перепрограмована EEPROM пам'ять даних з можливістю виконання 1000000 циклів стирання/запису;
- збереження даних у EEPROM протягом як мінімум 40 років.

Мікроконтролери підгрупи PIC16F8X мають розвинуті можливості вводу/виводу:

- 13 ліній вводу-виводу з індивідуальним встановленням напрямку обміну;
- високий що втікаючий/витікаючий струми, достатні для керування світлодіодами;
- максимальний струм, що втікає - 25 мА;
- максимальний струм, що витікає - 20 мА;
- 8-бітний таймер/лічильник TMR0 з 8-бітним програмованим попереднім дільником.

Спеціалізовані мікроконтролерні функції включають наступні можливості:

- автоматичне скидання при увімкненні (Power-on-Reset);
- таймер увімкнення при ініціалізації (Power-up Timer);
- таймер запуску генератора (Oscillator Start-up Timer);
- сторожовий (Watchdog) таймер WDT із власним вмонтованим генератором, що забезпечує підвищену надійність;
- EEPROM біт таємності для захисту коду;
- економічний режим SLEEP;
- обирає користувачем біти для встановлення режиму збудження вмонтованого генератора;
- послідовний пристрій програмування Flash/EEPROM пам'яті програм і даних з використанням тільки двох виводів.

КМОН технологія забезпечує МК підгрупи PIC16F8X додаткові переваги:

- статичний принцип роботи;
- широкий діапазон напруг живлення: 2,0...6,0 В;
- низьке енергоспоживання:
- менше 2 мА при 5В та 4МГц;
- порядку 15 мкА при 2В и 32КГц;
- менше 1 мкА для SLEEP-режиму при 2В.

Мікроконтролери підгрупи PIC16F8X відрізняються між собою тільки об'ємом ОЗП даних, а також об'ємом і типом пам'яті програм. Наявність у складі підгрупи МК із Flash-пам'яттю програм полегшує створення і налагодження прототипів промислових зразків виробів.

ТЕМА 5. НАБІР КОМАНД МІКРОКОНТРОЛЕРА PIC16F84A

Мікроконтролери підгрупи PIC16F8X мають просту й ефективну систему команд, що складає усього з 35 команд.

Кожна команда МК підгрупи PIC16F8X являє собою 14-бітове слово, розділене на код операції (OPCODE), і поле для одного і більше операндів, які можуть брати участь або не брати участь у цій команді. Система команд PIC16F8X є ортогональною і містить у собі команди роботи з байтами, команди роботи з бітами, операції з константами і команди управління. У таблиці 5.10 наведені описи полів команд.

Поле	Опис
f	Адреса регістра
w	Робочий регістр
b	Номер біта в 8-розрядному регістрі
k	Константа
x	Не використовується. Асемблер формує код з x=0
d	Регістр призначення: d=0 - результат у регістрі w d=1 - результат у регістрі f За замовчуванням d=1
label	Ім'я мітки
TOS	Вершина стека
PC	Лічильник команд
PCLATH	Регістр PCLATH
GIE	Біт дозволу всіх переривань
WDT	Сторожовий таймер
/TO	Тайм-аут
/PD	Вимикання живлення
dest	Регістр призначення: робочий регістр w або регістр, заданий у команді
[]	Необов'язкові параметри
()	Зміст
→	Присвоєння
< >	Поле номера біта
€	З набору

Для команд роботи з байтами f позначає регістр, з яким відбувається дія; d - біт, що визначає, куди покласти результат. Якщо d =0, то результат буде поміщений у регістр w, при d=1 результат буде поміщений у регістр "f", згаданий у команді.

Для команд роботи з бітами b позначає номер біта, що бере участь у команді, а f - це регістр, у якому даний біт розташований.

Для команд передачі управління й операцій з константами, k позначає восьми- або одинадцяти-бітну константу.

Майже всі команди виконуються протягом одного командного циклу. У двох випадках виконання команди займає два командних цикли:

- 1) перевірка умови і перехід;

2) зміна програмного лічильника як результат виконання команди.

Один командний цикл складається з чотирьох періодів генератора. Таким чином, для генератора з частотою 4 МГц час виконання командного циклу буде 1 мкс.

немоніка	Опис команди	Цикли	Біти стану	Прим.
ADDWF f, d	Додавання W з f	1	C,DC,Z	1,2
ANDWF f, d	Логічне I W і f	1	Z	1,2
CLRF f	Ініціалізація регістра f	1	Z	2
CLRWF	Очищення регістра W	1	Z	
COMF f, d	Інверсія регістра f	1	Z	1,2
DECF f, d	Декремент регістра f	1	Z	1,2
DECFSZ f, d	Декремент f, пропустити команду, якщо 0	1(2)		1,2,3
INCF f, d	Інкремент регістра f	1	Z	1,2
INCFSZ f, d	Інкремент f, пропустити команду, якщо 0	1(2)		1,2,3
IORWF f, d	Логічне АБО W і f	1	Z	1,2
MOVF f, d	Пересилання регістра f	1	Z	1,2
MOVWF f	Пересилання W у f	1		
NOP -	Холоста команда	1		
RLF f, d	Зсув f уліво через перенос	1	C	1,2
RRF f, d	Зсув f вправо через перенос	1	C	1,2
SUBWF f, d	Віднімання W з f	1	C,DC,Z	1,2
SWAPF f, d	Обмін місцями тетради у f	1		1,2
XORWF f, d	Виключаюче АБО W і f	1	Z	1,2
BCF f, b	Очищення біта в регістрі f	1		1,2
BSF f, b	Встановлення біта в регістрі f	1		1,2
BTFSC f, b	Пропустити команду, якщо біт у f дорівнює нулю	1(2)		3
BTFSS f, b	Пропустити команду, якщо біт у f дорівнює одиниці	1(2)		3
ADDLW k	Додавання константи та W	1	C,DC,Z	
ANDLW k	Логічне I константи та W	1	Z	
CALL k	Виклик підпрограми	2		
CLRWDW -	Ініціалізація сторожового таймера WDT	1	/TO,/P	
GOTO k	Перехід за адресою	2		
IORLW k	Логічне АБО константи та W	1	Z	
MOVLW k	Пересилання константи в W	1		
RETFIE -	Повернення з переривання	2		
RETLW k	Повернення з підпрограми з завантаженням константи в W	2		
RETURN -	Повернення з підпрограми	2		
SLEEP -	Перехід у режим SLEEP	1	/TO,/P	
SUBLW k	Віднімання W з константи	1	C,DC,Z	
XORLW k	Виключаюче АБО константи та W	1	Z	

Примітки до таблиці:

1. Якщо модифікується регістр вводу/виводу (наприклад, MOVF PORTB,1), то використовується значення, яке зчитується з виводів. Наприклад, якщо у вихідному регістрі порта, включеного на ввід, знаходиться "1", а зовнішній пристрій формує на цьому виводі "0", то в розряді даних буде записаний "0".
2. Якщо операндом команди є вміст регістра TMRO (і, якщо допустимо, d=1), то попередній дільник, якщо він підключений до TMRO, буде скинутий.
3. Якщо в результаті виконання команди змінюється лічильник команд або виконується перехід при перевірці умови, то команда виконується за два цикли. Другий цикл виконується як NOP.

Питання для самоперевірки:

1. Яку операцію виконує команда MOVF?
2. Яка різниця між командами BCF та BSF?
3. За допомогою якої команди виконується очищення регістру?
4. Що таке декремент та інкремент?
5. Що таке NOP?

ТЕМА 6. ВИДИ ТА ХАРАКТЕРИСТИКИ СВІТЛОДІОДІВ. ПРИНЦИП ДІЇ.

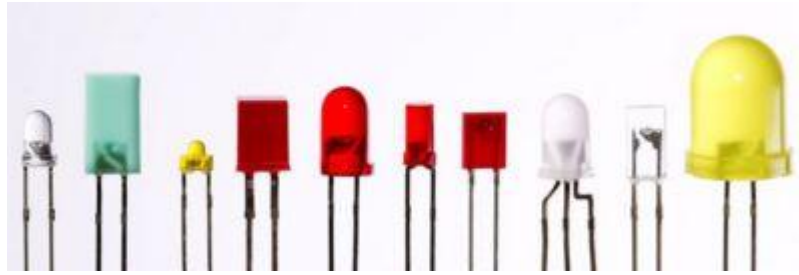
Світлодіоди можуть відрізнятися по безлічі ознак, але все ж види їх класифікують за конструктивними особливостями. Їх можна відрізнити за маркуванням, в якій зазначаються такі позначення:

DIP – індикаторний світлодіод з невеликою силою світлового потоку. Корпус його складається з епоксидної смоли і має форму циліндра з опуклим або запалим верхом, що грає роль лінзи. В середині може бути 1, 2 або 3 кристала різних кольорів. У такому випадку і ніжок висновків буде більше – 2, 3 і 4 відповідно. Його прямим нащадком став діод «піранья», який мав підвищену світловіддачу, проте він сильно нагрівався і мав великий розмір, а тому його виробництво припинилося;

SMD – це вже сучасні елементи, про які сьогодні ми розповімо докладно. Вони мають малі розміри, але яскравість їх значно вище, ніж у попередників. Вони використовуються для освітлювальних приладів, таких як лампи і світлодіодні стрічки;

COB – одне з останніх досягнень в цій області. Являє собою пластину, в якій знаходиться безліч кристалів. Такі елементи мають рівний і яскраве світло і використовуються, крім простих ламп, в прожекторах;

«Cree» – Яскравий компоненти, які виробляє однойменна фірма, яка не передає технологію та право виготовлення нікому. Ліхтарі на таких елементах можуть світити на відстань навіть в 2 км.



Основні характеристики

Струм споживання і його параметр у світлодіода.

Струм світлодіодів залежить від їх типорозміру, а іноді навіть від кольору. Зазвичай цей параметр має значення 0,02 А. Якщо ж в одному корпусі вмонтовано 4 кристала, то і струм зростає відповідно, і буде дорівнює 0,08 А.

Збільшення сили струму сприяє швидкому старінню елемента. Саме для того, щоб стабілізувати цей показник, в сучасних побутових LED-лампах та вбудований драйвер. При підключенні світлодіодних стрічок для цієї мети використовується блок живлення або контролер.

Лампа ближнього світла автомобіля на світлодіодах

При самотійному монтажі схеми до кожного світлодіоду монтується резистор, який обмежує величину струму, тим самим захищаючи його від швидкого виходу з ладу.

Номінальна напруга світлових діодів

Як такого поняття напруги для таких елементів не існує. Краще скористатися іншим терміном - падіння напруги на світлодіоді. Це означає показник, наскільки менше стало напруга при проходженні через елемент. Є усереднені значення цього показника, які залежать безпосередньо від кольору світіння. При синьому, зеленому та білому кольорі це 3 В, а ось для жовтого і червоного – 1,8-2,4 В.

При змішуванні різних кольорів світлодіодів народжується білий

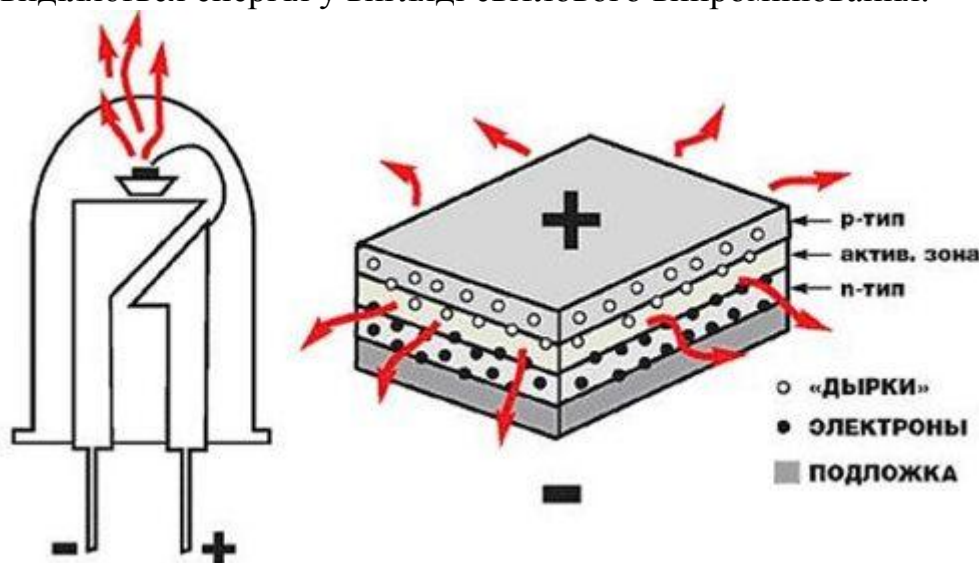
Показники значення опору

В цілому знати значення опору світлодіода не потрібно – ця інформація нічого не дасть. Адже якщо він підключений правильно, то воно незначне, якщо ж ні, то повне. Цікавий факт, що сам по собі цей показник у подібних елементів є динамічним. Це означає, що якщо додати напруги, то опір почне падати і навпаки.

Принцип дії

В основі роботи Led світлодіода лежить р-п-перехід, так званий електронно-дірковий перехід. Робота світлодіода побудована на взаємодії двох напівпровідників р-типу і n-типу. Р - positive, тобто позитивний тип, або дірковий N - negative, тобто негативний тип, або електронний. В результаті пропускання електричного струму в місці зіткнення двох напівпровідників відбувається перехід від одного типу провідності до іншого.

Коли через напівпровідники проходить електричний струм, негативний заряд електронів з'єднуються з іонами позитивно заряджених дірок. У цей момент виділяється енергія у вигляді світлового випромінювання.



Конструкція світлового елемента має 3 основні частини:

- 1) Катод "-" У індикаторних діодах зі слабким світінням він прекрасно видно. Катод має форму прапорця;
- 2) Анод "+" Це тонкий рівний провід;
- 3) Корпус, що складається з міцної прозорої або кольоровий епоксидної смоли.

Питання для самоперевірки:

1. Від чого залежить рівень струму у світлодіодах?
2. На якій технології побудовано принцип роботи світлодіодів?
3. Які основні конструктивні елементи будови світлодіодів?

ТЕМА 7. СПЕЦІАЛЬНІ ФУНКЦІЇ ASSEMBLER

Мікроконтролери підгрупи PIC16F8X мають набір спеціальних функцій, призначених для розширення можливостей системи, мінімізації вартості, виключення навісних компонентів, забезпечення мінімального енергоспоживання і захисту коду від зчитування. У PIC16F8X реалізовані наступні спеціальні функції:

- ініціалізація;
- сторожовий таймер (WDT);
- режим зниженого енергоспоживання (SLEEP);

- вибір типу **генератора**;
- захист коду від зчитування;
- біти ідентифікації;
- послідовне програмування в складі схеми.

У PIC16F8X існують розходження між варіантами ініціалізації:

- ініціалізація при увімкненні живлення;
- ініціалізація за зовнішнім сигналом /MCLR при нормальній роботі;
- ініціалізація за зовнішнім сигналом /MCLR у режимі SLEEP;
- ініціалізація при закінченні затримки таймера WDT при нормальній роботі;
- ініціалізація при закінченні затримки таймера WDT у режимі SLEEP.

Для реалізації ініціалізації при увімкненні живлення в МК підгрупи PIC16F8X передбачений вмонтований детектор увімкнення живлення. Таймер установлення живлення (PWRT) починає відлік часу після того, як напруга живлення перетинає рівень близько 1,2...1...1,8 Вольт. Після закінчення витримки близько 72мс. вважається, що напруга досягла номіналу і запускається інший таймер - таймер запуску генератора (OST), що формує витримку на стабілізацію кварцового генератора. Програмований **біт конфігурації** дозволяє дозволяти або забороняти витримку від вмонтованого таймера встановлення живлення. Витримка запуску змінюється в залежності від екземплярів кристала, від живлення і температури. Таймер на стабілізацію генератора відраховує 1024 імпульсу від генератора, що почав роботу. Вважається, що кварцовий генератор за цей час виходить на режим. При використанні RC генераторів витримка на стабілізацію не відбувається.

Якщо сигнал /MCLR утримується в низькому стані досить довго (довше часу всіх затримок), тоді після переходу /MCLR у високий стан програма почне виконуватися негайно. Це необхідно в тих випадках, коли потрібно синхронно запустити в роботу кілька PIC-контролерів через загальний для всіх сигнал /MCLR.

Мікроконтролери підгрупи PIC16F8X мають вмонтований сторожовий таймер WDT. Для більшої надійності він працює від власного внутрішнього RC-генератора і продовжує функціонувати, навіть якщо основний генератор зупинений, як це буває при виконанні команди SLEEP. Таймер виробляє сигнал ініціалізація. Вироблення таких сигналів може бути заборонена шляхом запису нуля в спеціальний біт конфігурації WDTE. Цю операцію проводять на етапі пропалювання мікросхем.

Номинальна витримка WDT складає 18 мс. (без використання дільника). Вона залежить від температури, напруги живлення, особливостей типів мікросхем. Якщо вимагаються великі затримки, то до WDT може бути підключений убудований предільник з коефіцієнтом ділення до 1:128, який програмується бітами PS2:PS0 у регістрі OPTION. У результаті можуть бути реалізовані витримки до 2,3 секунд.

Команди "CLRWDT" і "SLEEP" обнуляють WDT і лічильник, якщо він підключений до WDT. Це запускає витримку часу спочатку і запобігає на якийсь час вироблення сигналу ініціалізації. Якщо сигнал ініціалізації від WDT усе-таки відбувся, то одночасно обнуляється біт /TO у регістрі статусу. У додатках з високим рівнем завад уміст регістра OPTION піддається збоєм. Тому регістр OPTION повинен обновлятися через рівні проміжки часу.

Деякі зі спеціальних регістрів при скиданні не ініціалізуються. Вони мають випадковий стан при увімкненні живлення і не змінюються при інших видах ініціалізації. Інша частина спеціальних регістрів ініціалізується в "стан ініціалізація" при усіх видах ініціалізації, крім ініціалізації при закінченні затримки таймера WDT у режимі SLEEP. Просто ця ініціалізація розглядається як часова затримка в нормальній роботі. Є ще кілька винятків. Лічильник команд завжди скидається в нуль (0000h). Біти регістра статусу /TO і /PD встановлюються або скидаються в залежності від варіанта ініціалізації. Ці біти використовуються програмою для визначення природи ініціалізації:

Регістр	Адреса	Ініціалізація при увімкненні	Інші види живлення/ініціалізації
W	—	xxxx xxxx	uuuu uuuu
INDF	00h	— — — —	— — — —
TMR0	01h	xxxx xxxx	uuuu uuuu
PCL	02h	0000 0000	0000 0000
STATUS	03h	0001 1xxx	000q quuu
FSR	04h	xxxx xxxx	uuuu uuuu
PORT A	05h	—x xxxx	—u uuuu
PORT B	06h	xxxx xxxx	uuuu uuuu
TRIS A	85h	—1 1111	—1 1111
TRIS B	86h	1111 1111	1111 1111
OPTION	81h	1111 1111	1111 1111
EEDATA	08h	xxxx xxxx	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu
EECON1	88h	—0 0000	—0 q000
EECON2	89h	— — — —	— — — —
PCLATH	0Ah	—0 0000	—0 0000
INTCON	0Bh	0000 000x	0000 000u

Тут: x - невідоме значення;

u - незмінний біт;

"-" - невикористовуваний біт (читається як "0");

q - значення біта залежить від умов ініціалізації.

Режим пониженого енергоспоживання SLEEP призначений для забезпечення дуже малого струму споживання в очікуванні (менше 1 мкА при вимкненому сторожовому таймері). Вихід з режиму SLEEP можливий за зовнішнім сигналом ініціалізації або після закінчення витримки сторожового таймера.

Кристали PIC16F8X можуть працювати з чотирма типами вмонтованих генераторів. Користувач може запрограмувати два конфігураційних біти (FOSC1 і FOSC0) для вибору одного з чотирьох режимів: RC, LP, XT, HS. Тут XT - стандартний кварцовий генератор, HS - високочастотний кварцовий генератор, LP - низькочастотний генератор для економічних задач. Мікроконтролери PIC16F8X можуть тактуватись і від зовнішніх генераторів.

Генератор, побудований на кварцових або керамічних резонаторах, вимагає періоду стабілізації після увімкнення живлення. Для цього вмонтований таймер запуску генератора тримає пристрій у стані ініціалізації приблизно 18 мс. після того, як сигнал на /MCLR ніжці кристала досягне рівня логічної одиниці.

Можливість вибору типу генератора дозволяє ефективно використовувати мікроконтролери сімейства в різних задачах. Застосування RC генератора дозволяє зменшити вартість системи, а низькочастотний LP-генератор скорочує енергоспоживання.

Програмний код, що записаний у кристал, може бути захищений від зчитування за допомогою установки біта захисту (CP) у слові конфігурації в нуль. Уміст програми не може бути прочитаний так, щоб з ним можна було працювати. Крім того, при встановленому біті захисту неможливо змінювати програму. Те ж відноситься і до вмісту пам'яті даних EEPROM.

Якщо встановлений захист, то біт CP можна стерти тільки разом із вмістом кристала. Спочатку буде стерті EEPROM програмна пам'ять і пам'ять даних, і в останню чергу - біт захисту коду CP. При зчитуванні захищеного кристала читання будь-якої адреси пам'яті дасть результат виду 0000000XXXXXXX (двійковий код), де X - це 0 або 1.

Пам'ять даних EEPROM неможливо перевірити після установки біта захисту.

Для вибору різних режимів роботи використовуються біти конфігурації. Мікроконтролери підгрупи PIC16F8X мають 5 або 6 біт конфігурації, які зберігаються в EEPROM і встановлюються на етапі програмування кристала. Ці біти можуть бути запрограмовані (читається як "0") або залишені незапрограмованими (читається "1") для вибору придатного варіанта конфігурації пристрою. Вони розташовані в EEPROM-пам'яті за адресою 2007h. Користувачу варто пам'ятати, що ця адреса знаходиться нижче області кодів і недоступна програмі.

Призначення бітів конфігурації PIC16F84a

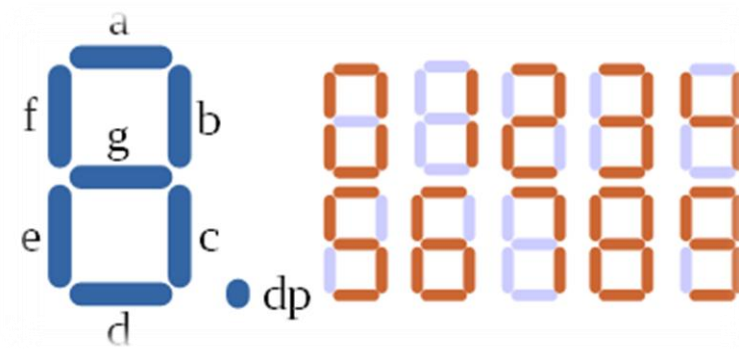
R-u	R-u	R-u	R-u	R-u
CP	/PWRTTE	WDTE	FOSC1	FOSC0
Біт 13:4	Біт 3	Біт 2	Біт 1	Біт 0
Біти 13:4 CP: біт захисту пам'яті програм 0 = пам'ять програм захищена 1 = захист відсутній				
Біт 3 /PWRTTE: біт використання таймера при увімкненні живлення 0 = таймер використовується (є затримка) 1 = таймер не використовується				
Біт 2: WDTE: біт використання сторожового таймера 0 = WDT не використовується 1 = WDT використовується				
Біти 1:0 FOSC1:FOSC0: біт вибору типу генератора 11 = генератор RC 10 = генератор HS 01 = генератор XT 00 = генератор LP				

ТЕМА 8. СЕГМЕНТНІ ІНДИКАТОРИ. ХАРАКТЕРИСТИКИ ТА ПРИНЦИП ДІЇ

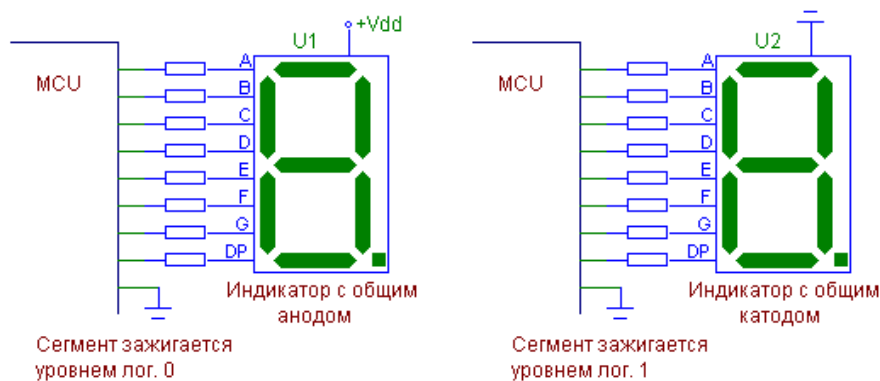
В даний час для відображення інформації все частіше використовуються графічні дисплеї, однак, семисегментний індикатори також не втратили свого значення. Якщо потрібно лише відображення чисел, то вони можуть стати більш привабливим варіантом, тому що прості в управлінні і можуть використовуватися спільно з будь-яким мікроконтролером з достатньою кількістю висновків. Рідкокристалічні семисегментний індикатори мають наднизьким енергоспоживанням (наприклад, в електронному годиннику, разом зі схемою управління працюють від однієї батарейки протягом декількох років).

Свою назву семисегментний індикатори отримали в зв'язку з тим, що зображення символу формується за допомогою семи окремо керованих (підсвічуються світлодіодом) елементів - сегментів. Ці елементи дозволяють відобразити будь-яку цифру 0..9, а також деякі інші символи, наприклад: '-', 'A', 'b', 'C', 'd', 'E', 'F' і інші. Це дає можливість використовувати індикатор для виведення позитивних і негативних десяткових і шістнадцяткових чисел і навіть текстових повідомлень. Зазвичай індикатор має також восьмий елемент - точку, яка використовується при відображенні чисел з десятковою крапкою. Кількість поділок вказує позначають буквами a, b, ..., g (a - верхній елемент, далі букви присвоюються сегментам за годинниковою стрілкою; g - центральний сегмент; dp - точка).

8 незалежних елементів, кожен з яких може перебувати в одному з двох станів - горить або не горить, дають всього $2 ** 8 = 256$ можливих комбінацій. Або 128 комбінацій, кожна з яких може бути з палаючої точкою або без неї.

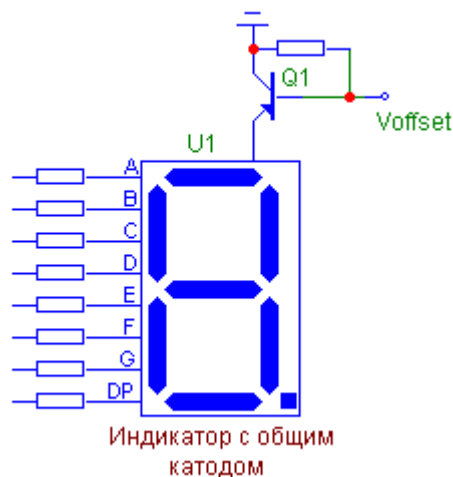


У тому випадку, якщо світлодіоди в індикаторі мають з'єднані разом аноди (схема із загальним анодом), загальний анод підключається до джерела напруги $+V_{DD}$, а катоди світлодіодів - сегментів підключаються до схеми управління (мікроконтролеру), яка відповідає за формування зображення на індикаторі. Запалюються сегменти низьким рівнем (логічний 0) на виводах мікроконтролеру. По відношенню до схеми управління струм до світлодіодів поступає, так що можуть використовуватися інтегральні схеми, які мають виходи з відкритим стоком. Змінюючи величину, яка живить індикатор напруги V_{DD} , можна регулювати яскравість світіння.



Якщо в індикаторі пов'язані один з одним катодом (схема із загальним катодом), то загальний катод підключається до загального проводу схеми, а аноди світлодіодів підключаються до мікроконтролеру. У цьому випадку сегмент запалюється високим рівнем на виході мікроконтролера (логічною 1), для якої струм світлодіода є вихідним.

Регулювати яскравість можна, підключивши загальний вивід індикатора до джерела зміщення напруги $0..V_{DD}$, розрахованого на вхідний струм, наприклад до емітерного повторювача на транзисторі структури р-п-р. Збільшуючи зміщення, зменшується яскравість світіння.



Для формування зображення символу на індикаторі використовують таблицю, яка ставить у відповідність коду символу набір відображуваних сегментів. Наведемо приклад такої таблиці. У цій таблиці код символу - його порядковий номер в таблиці. Набір сегментів, які формують символ, розглядається як двійкове число, сегменту А відповідає молодший біт числа. Якщо біт числа дорівнює 0, то відповідний сегмент не запалюється при відображенні символу, а якщо дорівнює 1, то запалюється. У таблиці також наводиться запис числа, що визначає набір запалюють сегментів, в шістнадцятковій формі.

#	Символ	Рис.	g	f	e	d	c	b	a	HEX
0	0		0	1	1	1	1	1	1	0x3F
1	1		0	0	0	0	1	1	0	0x06
2	2		1	0	1	1	0	1	1	0x5B
3	3		1	0	0	1	1	1	1	0x4F
4	4		1	1	0	0	1	1	0	0x66
5	5		1	1	0	1	1	0	1	0x6D
6	6		1	1	1	1	1	0	1	0x7D
7	7		0	0(1)	0	0	1	1	1	0x07 (0x27)
8	8		1	1	1	1	1	1	1	0x7F

9	9		1	1	0	1	1	1	1	0x6F
10	A		1	1	1	0	1	1	1	0x77
11	b		1	1	1	1	1	0	0	0x7C
12	C		0	1	1	1	0	0	1	0x39
13	d		1	0	1	1	1	1	0	0x5E
14	E		1	1	1	1	0	0	1	0x79
15	F		1	1	1	0	0	0	1	0x71
16	<Пробел>		0	0	0	0	0	0	0	0x00
17	-		1	0	0	0	0	0	0	0x40

Також, світлодіодний індикатор може бути багаторозрядним. В основному на практиці використовуються одно-, двох-, трьох- і чотири-розрядні світлодіодні індикатори:



ТЕМА 9. ТЕХНІЧНА ДОКУМЕНТАЦІЯ ATMEGA8 (DATASHEET)

DataSheet - це технічний опис мікроконтролеру або радіоелектронного компоненту, який розташовано на сайті фірми-виробника пристрою. У зв'язку з тим, що більшість пристроїв виробляється за кордоном, у різних країнах, документація написана на англійській мові і багато термінів потрібно переводити та розбирати.

Розглянемо тлумачення основних характеристик, зазначених у документації на прикладі мікроконтролеру ATmega8

Features

- **High-performance, Low-power Atmel® AVR® 8-bit Microcontroller**
- **Advanced RISC Architecture**
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- **High Endurance Non-volatile Memory segments**
 - 8Kbytes of In-System Self-programmable Flash program memory
 - 512Bytes EEPROM
 - 1Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- **Peripheral Features**
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and QFN/MLF package
 - Eight Channels 10-bit Accuracy
 - 6-channel ADC in PDIP package
 - Six Channels 10-bit Accuracy
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- **Special Microcontroller Features**
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby

Features

Перекладається як "функції". У середовищі електронників просто "фічі".

High Performance, Low Power AVR® 8-Bit Microcontroller

Високопродуктивний, що споживає мало енергії, 8-бітний мікроконтролер.

Advanced RISC Architecture

Розширена RISC архітектура - більше ніж 35 команд.

130 Powerful Instructions - Most Single Clock Cycle Execution

130 команд, більшість з них виконуються за один цикл. Така велика кількість має на увазі орієнтацію цього МК під мови високого рівня, наприклад Си. По-друге, одна команда виконується за один такт генератора. Тобто, при тактовій частоті 1 МГц одна команда буде виконуватися 1 мікросекунду (1 мкс, одну мільйонну частину секунди - 10^{-6}). А при 10 МГц - в десять разів швидше, тобто, 0,1 мкс.

32 x 8 General Purpose Working Registers

32 восьмибітних регістра загального користування.

Така велика кількість регістрів - це перевага, адже регістр - це осередок пам'яті в самому МК. А чим більше такої пам'яті - тим «спритніше» працює МК

Fully Static Operation

Повністю статична структура. Згадуємо про типи пам'яті: динамічної та статичної. Цей пункт запевняє нас, що МК збереже свою працездатність при тактовій частоті нижче сотень герц і навіть при відсутності тактової частоти. Також зазначимо що споживана потужність більшості типів МК безпосередньо залежить від тактової частоти: чим вище тактова частота, тим більше він споживає)

On-chip 2-cycle Multiplier

В даному МК є вбудований помножувач, який примножує числа за два такту.

High Endurance Non-volatile Memory segments

Надійна незалежна пам'ять, побудована у вигляді декількох сегментів.

Згадуємо типи пам'яті: EEPROM и FLASH.

8KBytes of In-System Self-programmable Flash program memory

8 Кбайт вбудованої в МК пам'яті. Пам'ять виконана за технологією Flash. У самому МК є вбудований програматор. Наявність вбудованого програматора цієї пам'яті, дозволяє завантажувати дані в пам'ять, використовуючи простий зовнішній програматор (в найпростішому випадку це п'ять дротів, якими мікроконтролер підключають до LPT порту комп'ютера).

256 Bytes EEPROM

У МК є 256 байт енергонезалежної пам'яті EEPROM. Отже, можна зберегти ще додаткову інформацію, яку можна змінювати програмою МК, без зовнішнього програматора.

1024 Bytes Internal SRAM

У МК є 1024 байт оперативної пам'яті (ОЗУ / RAM).

Write / Erase cycles: 10,000 Flash / 100,000 EEPROM

Пам'ять Flash витримує 10 000 циклів запису / стирання, а пам'ять EEPROM - до 100 000. Простіше кажучи, програму в МК можна змінювати до 10 000 разів, а свої дані в 10 разів більше.

Data retention: 20 years at 85 ° C / 100 years at 25 ° C

Збереження даних в пам'яті МК - до 20 років при температурі зберігання 85 ° С, і 100 років - при температурі 20 ° С.

Programming Lock for Software Security

МК має кілька областей пам'яті (уточнюємо яких), які можна захистити від прочитання установкою спеціальних біт захисту.

Two 8-bit Timer / Counters

One 16-bit Timer / Counter

У МК є два таймера / лічильника: 8 і 16 біт.

Three PWM Channels

Три канали ШІМ

8-channel ADC in TQFP and QFN / MLF package

Eight Channels 10-bit Accuracy

6-channel ADC in PDIP package

Six Channels 10-bit Accuracy

У складі МК є кілька каналів АЦП: 6- для корпусу PDIP і 8 - для корпусу QFN/MLF. Розрядність АЦП - 10 біт.

Byte-oriented Two-wire Serial Interface

Programmable Serial USART

В даному МК реалізований апаратний двопровідний інтерфейс зв'язку USART, байт орієнтований і програмований - є можливість налаштування параметрів інтерфейсу.

Master/Slave SPI Serial Interface

Реалізовано SPI інтерфейс зв'язку, режими Майстер/Підлеглий.

Programmable Watchdog Timer with Separate On-chip Oscillator

Сторожовий таймер з власним автономним генератором.

On-chip Analog Comparator

Аналоговий компаратор.

Power-on Reset and Programmable Brown-out Detection

Реалізовано режими контролю напруги живлення і захист роботи МК при поганому харчуванні (гарантує збільшення надійності роботи всієї системи).

Internal Calibrated RC Oscillator

Вбудований калібрований RC-генератор (можна запустити МК без зовнішніх елементів).

External and Internal Interrupt Sources

Реалізовано кілька типів зовнішніх і внутрішніх переривань.

Five Sleep Modes

П'ять режимів «сну» (зменшення енергоспоживання МК за рахунок відключення деяких внутрішніх вузлів або спеціальних методів уповільнення їх роботи)

ТЕМА 10. ОСНОВНІ ДИРЕКТИВИ ПРЕПРОЦЕСОРА C1

Препроцесор - це спеціальна програма, яка є частиною компілятора мови Сі. Вона призначена для попередньої обробки тексту програми. Препроцесор дозволяє включати в текст програми файли і вводити макроозначення.

Робота препроцесора здійснюється за допомогою спеціальних директив (вказівок). Вони відзначаються знаком решітка #. Після закінчення рядків, що позначають директиви в мові Сі, крапку з комою можна не ставити.

Основні директиви препроцесора

#include - вставляє текст з зазначеного файлу

#define - задає макровизначеннями (макрос) або символічну константу

#undef - скасовує попереднє визначення

#if - здійснює умовну компіляцію при істинності константного виразу

#ifdef - здійснює умовну компіляцію при визначеності константи

#ifndef - здійснює умовну компіляцію при невизначеності константи

#else - гілка умовної компіляції при хибності вираження

#elif - ветка условной компиляции, образуемая слиянием else и if

#endif - конец ветки условной компиляции

#line - препроцессор изменяет номер текущей строки и имя файла

#error - выдача диагностического сообщения

#pragma - действие, зависящее от конкретной реализации компилятора.

Директива **#include** дозволяє включати в текст програми вказаний файл. Якщо файл є стандартною бібліотекою і знаходиться в папці компілятора, він полягає в кутові дужки <>.

Якщо файл знаходиться в поточному каталозі проекту, він вказується в лапках ". Для файлу, що знаходиться в іншому каталозі необхідно в лапках вказати повний шлях.

```
#include <stdio.h>
#include "func.c"
```

Директива **#define** дозволяє вводити в текст програми константи і макроозначення. Загальна форма запису

#define *ідентифікатор* *замена*

Поля *ідентифікатор* и *замена* разделяются одним или несколькими пробелами.

Директива **#define** вказує компілятору, що потрібно підставити рядок, визначену аргументом заміну, замість кожного аргументу ідентифікатор в вихідному файлі. Ідентифікатор статті не замінюється, якщо він знаходиться в коментарі, в рядку або як частина більш довгого ідентифікатора.

```
#include <stdio.h>
#define A 3
int main()
{
    printf("%d + %d = %d", A, A, A+A); // 3 + 3 = 6
    getchar();
    return 0;
}
```

Залежно від значення константи компілятор присвоює їй той чи інший тип. За допомогою суфіксів можна перевизначити тип константи:

U або u представляє цілу константу в беззнаковій формі (unsigned);
F або f дозволяє описати речову константу типу float;
L или l дозволяє виділити цілій константі 8 байт (long int);

```
#define A 280U    // unsigned int
#define B 280LU   // unsigned long int
#define C 280     // int (long int)
#define D 280L    // long int
#define K 28.0    // double
#define L 28.0F   // float
#define M 28.0L   // long double
```

Друга форма синтаксису визначає макрос, подібний функції, з параметрами. Ця форма допускає використання необов'язкового списку параметрів, які повинні знаходитися в дужках. Після визначення макросу кожне наступне входження

ідентифікатор **имя макроса(аргумент 1, ..., аргумент n)**

Директива **#undef** переопределяет константу або препроцесорний макрос, раніше визначений за допомогою директиви **#define**.

#undef name

Давайте подивимося приклад використання директиви **#undef**:

```
#define E 2.71828 // раннє певний макрос
int sumE = E + E; // звернення до макросу
#undef E // тепер E - НЕ макрос
```

Як правило, директива **#undef** використовуються для зняття, раніше визначеної константи або макросу, в невеликій області програми. Це робиться для того, щоб для всієї програми, макрос або константа залишалися, а для деякої області, ці ж макрос або константа можуть бути перевизначені. Небезпечно було б у всій програмі перевизначати константу, але в короткий області, це порівняно безпечно. Директива **#undef** є єдиним способом створення цієї області, так як область дії макросів або констант діє від директиви **#define** до **#undef**.

Питання для самоперевірки

1. За допомогою якої директиви підключаються бібліотеки або програми? Чим відрізняються записи підключення стандартних бібліотек від власноруч створених?
2. Що дозволяє визначити директива **#define**?
3. Як зняти раніше визначену константу або макрос?
4. Чим відрізняється **#define K 50** **#define K 50U**?

ТЕМА 11. СТВОРЕННЯ ФУНКЦІЙ НА МОВІ ПРОГРАМУВАННЯ CІ

Кожна програма у своєму складі повинна мати головну функцію `main()`. Саме функція `main()` забезпечує створення точки входу в об'єктний модуль. Крім функції `main()`, в програму може входити будь-яка кількість функцій. Кожна функція по відношенню до іншої є зовнішньою. Для того, щоб функція була доступна, необхідно, щоб до її виклику про неї було відомо компілятору.

Із поняттям функції у мові Сі пов'язано три наступних компоненти:

- опис функції;
- прототип функції;
- виклик функції.

Опис функції складається з двох частин: заголовка та тіла. Опис функції має наступну форму запису:

```
/* заголовок функції*/  
[тип_результату] <ім'я>([список_параметрів])  
{  
/* оголошення і оператори */  
тіло_функції  
}
```

Тип результату — це тип значення, яке повертається. У випадку відсутності специфікатора типу передбачається, що функція повертає ціле значення (`int`). Якщо функція не повертає ніякого значення, то на місці типу записується специфікатор `void`. У списку параметрів для кожного параметра повинен бути зазначений тип. При відсутності параметрів список може бути порожнім або мати специфікатор `void`.

Тіло функції являє собою послідовність оголошень і операторів, які описують визначений алгоритм. Важливим оператором тіла функції є оператор повернення в точку виклику: `return [вираз]`. Оператор `return` має подвійне призначення. Він забезпечує негайне повернення у викликаючу функцію і може використовуватися для передачі обчисленого значення функції. У тілі функції може бути декілька операторів `return`, але може не бути й жодного. В останньому випадку повернення у викликаючу програму відбувається після виконання останнього оператора тіла функції.

Прототип функції може вказуватися до виклику функції замість опису функції для того, щоб компілятор міг виконати перевірку відповідності типів аргументів і параметрів. Прототип функції за формою такий же, як і заголовок функції, і наприкінці його ставиться крапка з комою `;`. Параметри функції в прототипі можуть мати імена, які не є обов'язковими.

Виклик функції може бути оформлений у вигляді оператора, якщо у функції відсутнє значення, що повертається, або у вигляді виразу, якщо існує значення, що повертається.

У першому випадку оператор має наступний формат:

ім'я_функції(список_аргументів);

Наприклад, $f(x)$;

В другому випадку вираз записується у такий спосіб: $h=f(x)$;

Значення обчисленого виразу є значенням функції, що повертається. Значення, що повертається, передається в місце виклику функції і є результатом її роботи.

Число і типи формальних аргументів повинні співпадати з числом і типом фактичних параметрів функції. При виклику функції фактичні параметри підставляються замість формальних аргументів. Після виклику функції значення аргументу, яке відповідає формальному параметру, використовується в тілі функції, що викликається. Такий спосіб виклику функцій називається виклик по значенню: змінні передаються функції в якості аргументів, їх значення копіюються у відповідні локальні змінні функції (формальні параметри), а самі фактичні змінні в основній програмі не змінюються.

Приклад. Скласти програму для обчислення суми k чисел.

```
#include <iostream.h>
void sum(int); // створюємо прототип функції
int s = 0; // глобальна змінна
void main() // головна функція
{
    int i, b, k; // локальні змінні
    cout << "\nВведіть кількість доданків";
    cin >> k;
    for(i = 0; i < k; i++)
    {
        cout << "\nВведіть новий доданок";
        cin >> b;
        sum(b); // виклик функції
    }
    cout << "\ns = " << s;
}
void sum(int c)
{
    s = s + c;
}
```

У цій програмі змінна s є глобальною, вона доступна з обох функцій програми — $\text{main}()$ та $\text{sum}()$, а змінні i , b , k та c — локальні, доступні тільки у тих функціях, де вони оголошені.

Якщо глобальна і локальна змінні мають одне і теж ім'я, тоді вважається, що оголошені дві різні змінні зі своїми областями використання. При цьому локальна змінна буде видима у тій функції, де вона оголошена, а глобальна у всій програмі за виключенням функції, у якій оголошена локальна змінна.

Питання для самоперевірки

1. Як об'явити створення функції?
2. Як викликати функцію?
3. Що означає, якщо в аргументі функції вказано ключове слово `void`?

4. Для чого використовується оператор return?

ТЕМА 12. ДАТЧИКИ ВИМІРЮВАННЯ. ХАРАКТЕРИСТИКИ ДАТЧИКІВ

Датчик – це електричний апарат, призначений для перетворення безперервної зміни вхідної (контрольованої) неелектричної величини на зміну вихідної електричної величини.

При побудові датчиків використовуються різні фізичні принципи, які значною мірою визначають області раціонального застосування того чи іншого датчика.

Параметричний датчик змінює будь-який зі своїх параметрів під впливом самої вимірюваної величини і вимагає підключення до будь-якого зовнішнього джерела енергії.

Генераторний датчик сам генерує вихідний сигнал і не вимагає підключення до зовнішнього джерела енергії. Як приклади датчиків такого роду можна назвати різні п'єзоелектричні датчики тиску або тахогенераторні, датчики швидкості обертання.

До параметричних датчиків належать:

- резистивні;
- індуктивні;
- трансформаторні;
- ємнісні.

До генераторних датчиків належать:

- термоелектричні;
- індукційні;
- п'єзоелектричні;
- фотоелектричні.

Характеристики датчиків

Функція перетворення вимірювального перетворювача – це залежність вихідної величини даного вимірювального перетворювача від вхідних, що задаються або аналітичним вираженням, або графіком, або таблицею.

Чутливість перетворювача – це іменована величина, що показує, наскільки зміниться вихідна величина при зміні вхідної величини на одну одиницю. Для термопари одиницею чутливості буде мілівольти на 1 градус Кельвіна, для регульованого електродвигуна – обороти за секунду на 1 вольт і т. д. Основною характеристикою є чутливість:

$$S = \Delta Y / \Delta X,$$

де ΔY , ΔX – збільшення вихідної і вхідної величин.

Датчики можуть бути лінійними ($S = \text{const}$) і нелінійними ($S = \text{var}$). В останніх чутливість залежить від вхідної величини.

Найважливішим параметром датчика є поріг чутливості, що являє собою найменше значення вхідної величини, що викликається зміною вихідної величини, яка може бути виміряна.

Роздільна здатність перетворення – це найменша зміна вхідного сигналу, яка може бути виміряна перетворювачем.

Відтворюваність є мірою того, наскільки близькі один до одного результати вимірювань однієї і тієї самої фізичної величини.

Прецизійність є мірою того, наскільки близькі один до одного результати аналогічних вимірювань.

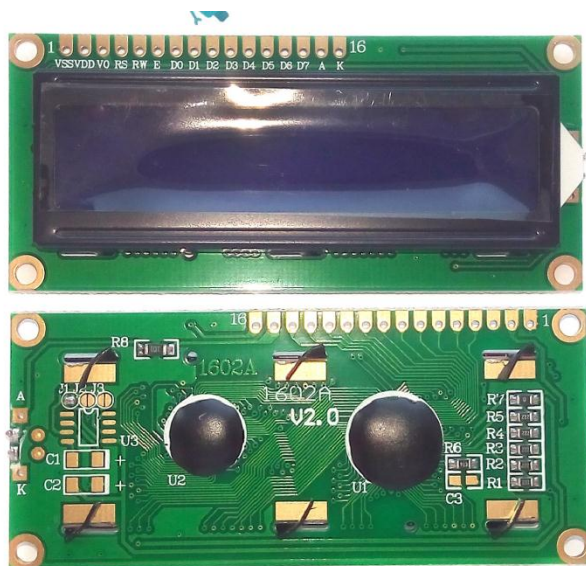
Точність (похибка) вимірювання показує, наскільки показане датчиком значення параметра близько до його істинного значення. Зазвичай точність задається у відсотках від повної шкали вимірювального приладу і в результаті являє собою деяку абсолютну величину. Номінальною характеристикою датчика є залежність вхід-вихід, яка зазначена в паспорті.

Питання для самоперевірки

1. Для чого використовуються датчики, приведіть приклади.
2. Які існують датчики?
3. Що визначає найменшу величину вхідного параметру датчику?
4. Що є номінальною характеристикою датчика?

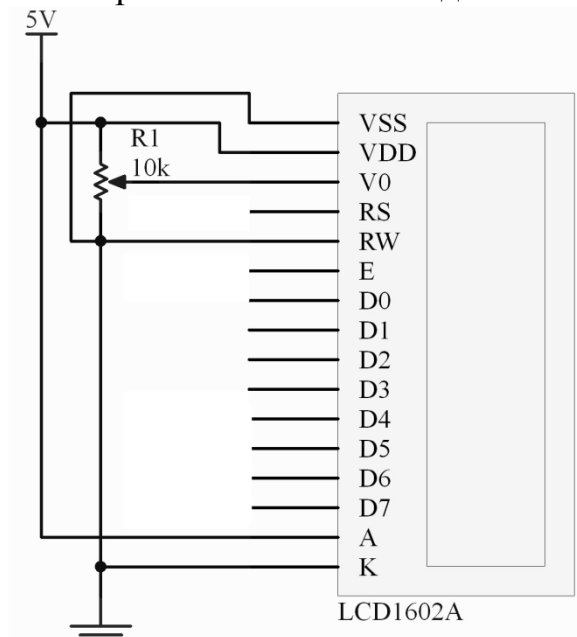
ТЕМА 13. ХАРАКТЕРИСТИКИ СИМВОЛЬНИХ РІДКОКРИСТАЛІЧНИХ ДИСПЛЕЇВ

Символьний рідкокристалічний індикатор 1602 HD44780, це пристрій, який здатний виводити певні символи в два рядки по 16 символів в кожному.



Індикатор виконаний у вигляді модуля, в якому встановлений контролер HD44780, призначений для управління дисплеєм. Також, як ми бачимо, в даному модулі існують 16 контактів, до яких приєднується мікроконтролер. Поставляється даний модуль без контактів. На їх місці існують контактні технологічні отвори. Швидше за все, зроблено це для того, щоб користувачі самі могли вибрати тип контактів, які будуть зручні для роботи з даним модулем.

Розглянемо призначення пінів на дисплеї:



8. Pin assignment

Pin NO.	Symbol	Function		Remark
1	Vss	Power Supply	0V	
2	Vdd		+5V	
3	Vo		For LCD	Variable
4	RS	Register Select (H: Data L: Instruction)		
5	R/W	L: MPU to LCM H: LCM to MPU		
6	E	Enable		
7	DB0	Data Bit 0		
8	DB1	Data Bit 1		
9	DB2	Data Bit 2		
10	DB3	Data Bit 3		
11	DB4	Data Bit 4		
12	DB5	Data Bit 5		
13	DB6	Data Bit 6		
14	DB7	Data Bit 7		
15	A	Anode of LED Unit		
16	K	Cathode of LED Unit		

1 - VSS - загальний провід або "земля"

2 - VDD - живлення.

3 - V0 - це ніжка, за допомогою якої регулюється контрастність дисплея. Тобто контрастність дисплея буде залежати від поданого напруги на дану ніжку. Як правило береться змінний резистор на 10 кОм, підключений крайніми ніжками на загальний провід і на живлення, а з центральної ніжки даного резистора провід йде на ніжку V0 і за допомогою регулювання резистора змінюється контрастність дисплея в модулі.

4 - RS - ніжка, за допомогою якої контролер дисплея буде "знати", які саме дані знаходяться на шині даних. Якщо подаємо на дану ніжку логічний 0, то значить буде команда, якщо 1 - то це дані.

5 - RW - дана ніжка в залежності від логічного стану на ній говорить контролеру дисплея, будемо ми з нього читати або будемо ми в нього писати дані. Якщо буде 0 - то ми в контролер дисплея писатимемо, а якщо 1 - то будемо читати дані з контролера дисплея. Ця функція використовується рідко. Як правило ми завжди тільки пишемо дані в дисплей.

6 - E - це так звана стробуюча шина, коли 1 змінюється в 0, на якій контролер дисплея розуміє, що саме зараз настав момент читання даних на ніжках даних D0 - D7, або передачі даних з модуля в залежності також від стану ніжки RW.

Ніжки D0 - D7 - це паралельна восьмибітна шина даних, через яку і передаються або приймаються дані. Нумери 0 - 7 відповідають однойменним бітам в байті даних. Але також є ще 4-бітний спосіб передачі даних в контролер і з контролера дисплея, коли використовуються тільки ніжки даних D4 - D7, а

ніжки D0 - D3 вже не використовуються. Як правило такий спосіб використовується в цілях економії ніжок порту.

Для управління в DataSheet дисплею діснує таблиця, що представляє собою перелік команд і способи їх реалізації.

У першій колонці даної таблиці знаходяться самі команди. Наступні 2 колонки - це те, в якому стані повинні в момент команди перебувати ніжки RS і RW. Подальші 8 колонок показують нам стан ніжок шини даних D0-D7 потім йде колонка з поясненнями до командам, тобто що саме за допомогою даної команди ми досягнемо.

В останній колонці знаходяться тайминги або тимчасові інтервали, необхідні для того, щоб та чи інша команда або інструкція виконалася. Причому обумовлено, при якій саме частоті генератора це досягається. Тобто, в модулі існує генератор, тактуючий роботу контролера дисплея, який налаштований на певну частоту і дана частота може бути різною.

11. DISPLAY CONTROL INSTRUCTION

Instruction	Instruction Code										Description	ExecutionTime(f osc=270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM set DDRAM address to "00H" from AC	1.52ms
Return Home	0	0	0	0	0	0	0	0	1	-	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed	1.52ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and enable the shift of entire display	38 μ s
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Set display (D) cursor(C) and blinking of cursor(B) on/off	38 μ s
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	-	-	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data	38 μ s
Function Set	0	0	0	0	1	DL	N	F	-	-	Set interface data length of display line (N: 2line/1line)and, display font type F:5X11dots/5X8dots	38 μ s
Set CGRAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter	38 μ s
Set DDRAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter	38 μ s
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF The contents of address counter of address counter can also be read	0 μ s
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM)	38 μ s
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM)	38 μ s

Питання для самоперевірки

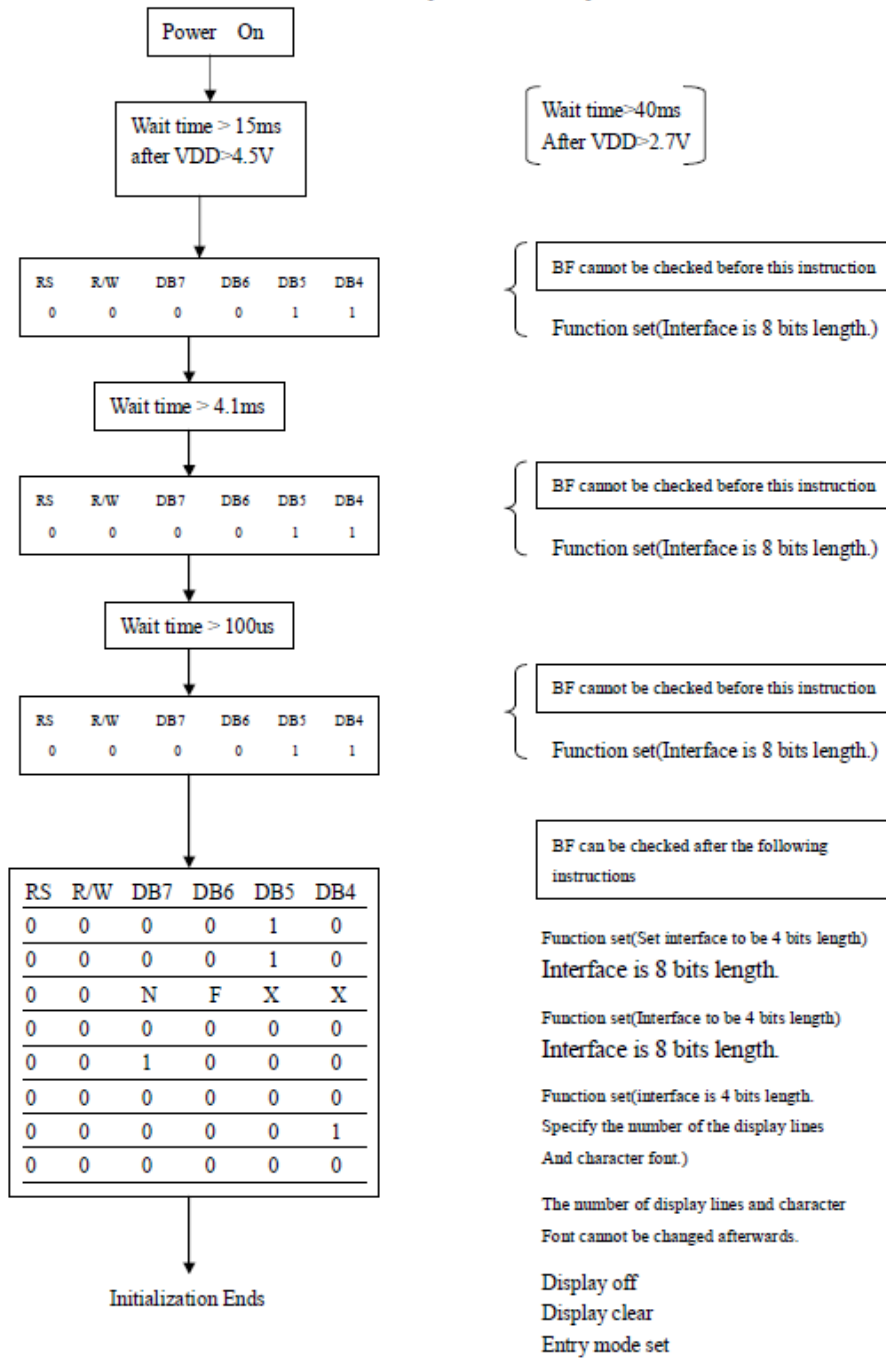
1. Для чого використовуються виводи дисплею D0-D7?
2. За допомогою якого виводу регулюється контрастність дисплею?
3. Як визначити, що саме поступає на вхід D0-D7 – команда чи дані?
4. Як визначити, який процес відбувається – читання чи запис?

ТЕМА 14. АЛГОРИТМ ВКЛЮЧЕННЯ 4-Х БІТНОГО РЕЖИМУ РІДКОКРИСТАЛИЧНОГО ДИСПЛЕЮ

4-бітний режим передачі даних використовується в цілях економії ніжок порту.

Відповідно до технічної документації на дисплей, алгоритм включення 4-бітного режиму має наступну послідовність:

[4-Bit Interface]



Для включення 4-х бітного режиму з дисплею використовуються виводи RS, R/W, DB7, DB6, DB5, DB4.

Після передачі кожної комбінації бітів на виводи дисплею, необхідно дотримуватись вказаного таймінгу у вигляді затримки.

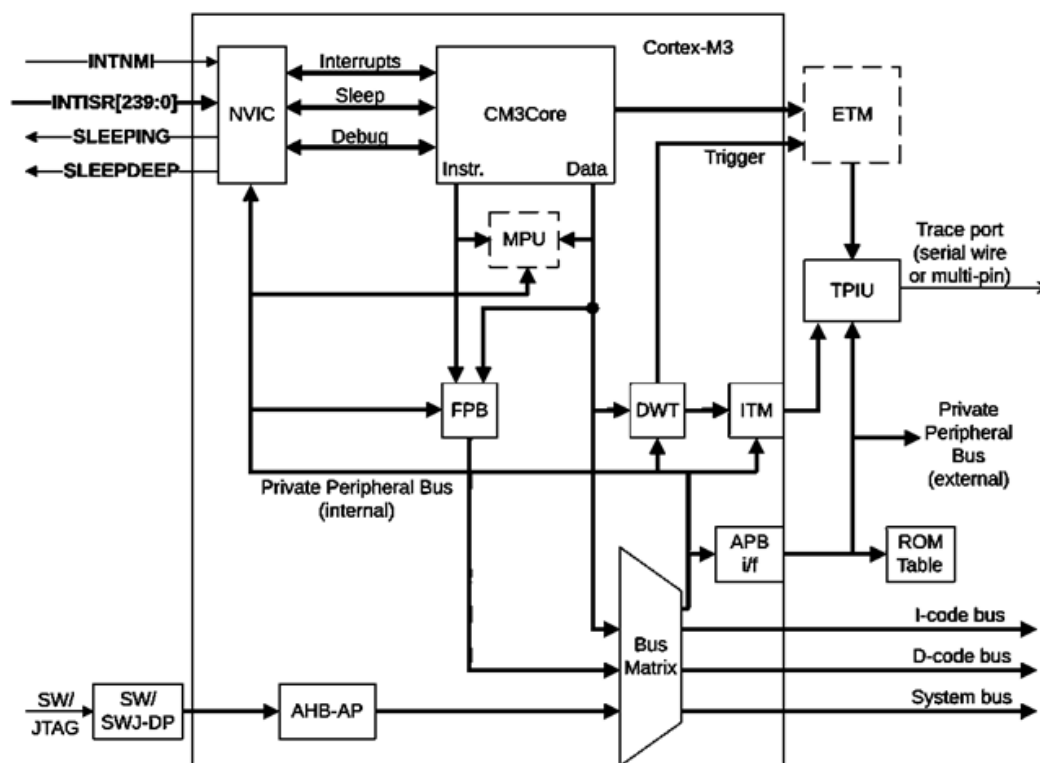
Питання для самоперевірки

1. Для чого використовується 4-х бітний режим роботи дисплею?
2. Які виводи застосовуються для включення 4-х бітного режиму?
3. Чого обов'язково потрібно дотримуватись при включенні 4-х бітного режиму?

ТЕМА 16. АРХІТЕКТУРА МІКРОКОНТРОЛЕРІВ STM32

Розглянемо схему мікроконтролера, на рівні верхнього рівня абстракції, він є найменш деталізованим, але цього достатньо для загального розуміння.

На цьому рівні можна виділити три основних компоненти:



Обчислювальне ядро (Cortex-M3Core)

Виконує інструкції, робить обчислення у своєму АЛУ (арифметико-логічному пристрої). Його Гарвардська архітектура дозволяє одночасно завантажувати інструкції й здійснювати доступ до пам'яті — завдяки цьому, а також триступінчастому конвеєру, більшість інструкцій виконуються за 1 такт. Ядро Cortex-M3 підтримує набір інструкцій Thumb-2, який містить як 32-бітні, так і 16-бітні інструкції для скорочення обсягу коду за рахунок менш далекобійних переходів; має 13 регістрів загального призначення, знижуючи потребу в частому звертанні до пам'яті.

Контролер вкладених переривань (NVIC)

Ця частина відповідає за генерацію переривань на різні події: зовнішні — зміна логічного рівня на вході ніжки, пробудження з режиму сну; і внутрішні — завершення приймання/відправлення даних, переповнення лічильника таймера й т.п. Контролер підтримує до 240 переривань і до 256 рівнів пріоритету, причому вхід в оброблювач переривання займає 12 тактів і переривання можуть бути вкладеними: якщо під час обробки переривання виникає переривання з меншим пріоритетом, то друге буде оброблено через 6 тактів після обробки першого. Крім того, існують немасковані переривання — NMI (Non-Masked Interrupts), які неможливо скинути, не обробивши, і які

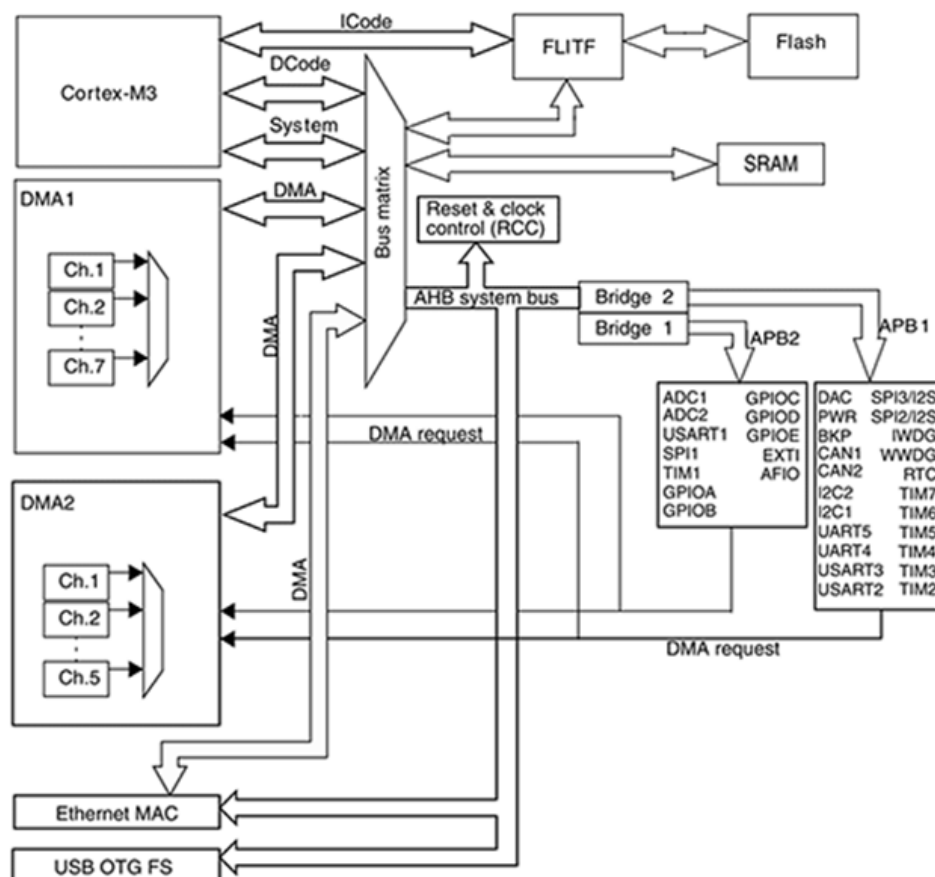
переривають виконання програми незалежно від яких-небудь умов. Такі переривання генеруються при збої зовнішнього джерела тактування (кварцу, керамічного резонатора) і при виявленні некоректної інструкції.

Шинна матриця (Bus matrix)

Сучасні процесори містять безліч різних шин, до яких підключаються інші пристрої системи. На зорі комп'ютерної епохи шиною (bus) називали просто пачку провідників, що з'єднують кілька пристроїв із процесором, який міг одночасно працювати лише з одним пристроєм, а інші в цей час простоювали, тому що шина була на всіх одна. До того ж, усі пристрої були змушено працювати на одній швидкості (самого повільного пристрою), що втримувало ріст продуктивності систем. Сьогодні шини стали складніше їх число збільшилося на порядок. Роздільні шини дають можливість працювати з декількома пристроями одночасно, причому на різних швидкостях: у кожній шині може бути своя швидкість. Можна гнучко управляти енергоспоживанням, відключаючи невикористовувані пристрої й цілі шини. Для керування всією цією купою шин знадобилося вводити спеціальний контролер, керуючий обміном даними між шинами й процесором.

Шинна матриця — це розвиток ідеї простого контролера шини: тут шини з'єднані так, що пристрої можуть взаємодіяти прямо, не через ядро. Також вона управляє доступом до не вирівняних даних (адреси яких не кратні 4, як прийнято в 32-бітних архітектурах) і атомарним доступом до окремих бітів у спеціально виділеному діапазоні (технологія bit-banding)

Тепер розглянемо спрощену блок схему архітектури STM32



Шини ядра STM32:

Icode bus – 32-бітна шина інструкцій — забезпечує зв'язок ядра з інтерфейсом інструкцій Flash.

Dcode bus – шина даних — забезпечує зв'язок ядра з інтерфейсом даних в Flash.

System bus – системна шина ядра — забезпечує зв'язок ядра й периферії

Flash interface (FLITF) інтерфейс Flash-Пам'яті — забезпечує читання, запис, стирання, читання з буфером попередньої вибірки, захист пам'яті (від запису або читання).

DMA (Direct Memory Access) — забезпечує прямий доступ до пам'яті в обхід ядра (потрібно лише задати що, звідки й куди передати, а DMA сам усе вибере й передасть кому треба через матрицю шин)

Reset & Clock Control (RCC) — забезпечує тактування ядра й периферії й скидання контролера.

Периферія підключена до шини APB (PPB), яка з'єднана з АНВ System bus (Advanced High-performance Bus – шина, яка зв'язує матрицю шин і периферійні шини APB) через два так звані «мости» (шини) APB1 і APB2, причому до кожного мосту підключений свій набір периферії. Фішка в тому, що в цих мостів різняться максимальні робочі частоти: APB2 може працювати на частоті ядра (максимум 72 МГц для STM32F10x), а APB — максимум на половині частоти ядра (тобто 36 МГц). Як бачимо, GPIO (виводи загального призначення) підключені до більш швидкої APB2, що дозволяє працювати їм на частоті 50 МГц у МК із частотою 72 МГц; також сюди підключені АЦП (до 1 мільйона вибірок у секунду), USART1 (до 4.5 Мбит/с), SPI1 (до 18 Мбит/с). Менш швидка периферія підключена до APB1. Хоча є виключення — full-speed USB 2.0, наявний тільки в МК лінійки Performance line, підключений до APB1.

Периферійні модулі STM32:

У мікроконтролерах STM32 доступні наступні інтерфейси передачі даних:

FSMC – Інтерфейс підключення зовнішньої SRAM, PSRAM, NOR, NANDROM, RAM, Nandflash, Compact Flash пам'яті, а також для підключення LCD дисплеїв з паралельною шиною.

Ethernet – Інтерфейс для підключення до комп'ютерної мережі.

SPI/(I2S) – Широко розповсюджений послідовний інтерфейс передачі даних, служить для підключення різних мікросхем, карт пам'яті, LCD дисплеїв з послідовною шиною й багато чого іншого.

I2C – Двохпровідна шина передачі даних, служить для підключення різних мікросхем і інших модулів.

USART (UART) – Стандартний універсальний синхронний (асинхронний) приймально-передавальний інтерфейс, служить для обміну даними з іншими пристроями через RS-232, RS-485 шини й т.п.

USB – Сучасний комп'ютерний інтерфейс передачі даних, призначений для підключення до мікроконтролера USB клавіатур, flash накопичувачів, а також для підключення мікроконтролера до ПК, може працювати як у режимі пристрою (Device), так і в режимі хоста (Host).

CAN – Промисловий інтерфейс передачі даних, також використовується в автомобілебудуванні й т.п.

Camera interface – Спеціалізований інтерфейс, призначений для підключення високошвидкісної цифрової камери, з 8-, 10-, 12- або 14-бітною паралельною шиною даних.

SDIO – Інтерфейс для високошвидкісного підключення SD карт пам'яті.

Крім периферійних модулів, відповідальних за обмін даними із зовнішніми пристроями, існують також і інші різні периферійні модулі, значення яких не менш важливе, ніж інтерфейсних модулів:

Таймер – дуже важливий модуль, що дозволяє точно відміряти інтервали часу, генерувати вихідний ШИМ сигнал, підключати енкодер, підраховувати період вхідного сигналу й ін.

АЦП – Аналогово-цифровий перетворювач (має вбудований датчик температури) призначений для виміру значень аналогових напруг, також може працювати як віконний компаратор.

ЦАП – Цифро-аналоговий перетворювач видає на виході аналоговий сигнал відповідний заданому цифровому значенню.

RTC – Годинник реального часу.

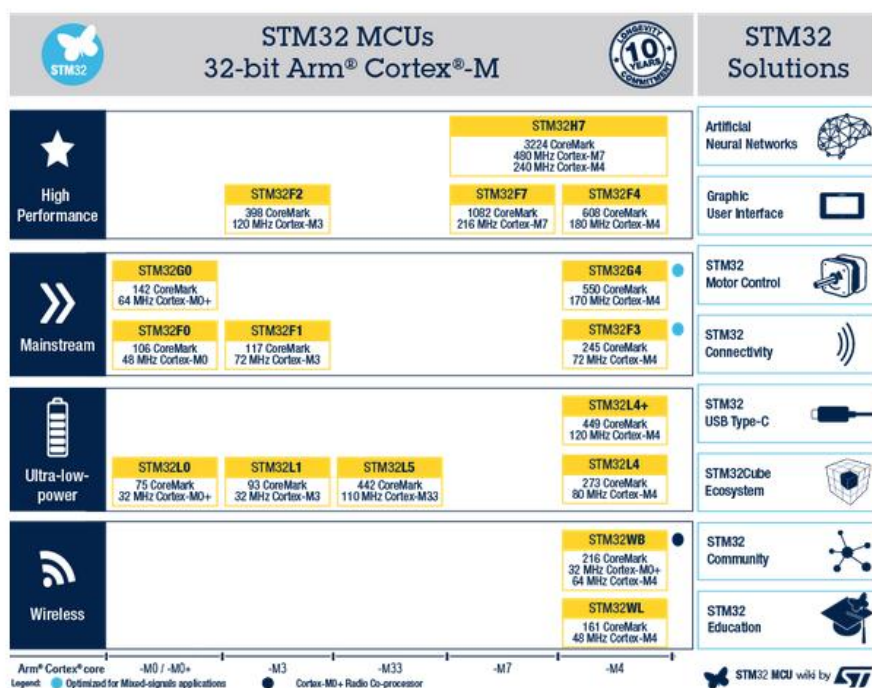
JTAG – Модуль внутрішньо-схемного налагодження програми.

Питання для самоперевірки

1. На якому ядрі побудована архітектура мікроконтролерів STM32?
2. Опишіть основні модулі побудови мікроконтролерів STM32/
3. Які шини забезпечують обмін між основними модулями мікроконтролера?
4. Які модуль забезпечують прямий доступ к пам'яті?
5. Які інтерфейси периферії доступні у МК STM32&

ТЕМА 17. ЛІНІЙКИ СІМЕЙСТВА МІКРОКОНТРОЛЕРІВ STM32

STM32 поділяється на ряд лінійок орієнтованих для застосування в різноманітних пристроях, які мають певні відмінності одна від одної.



Найбільш простою є лінійка F0, в її основі лежить ядро Cortex-M0, орієнтована на використання в домашніх розважальних пристроях, побутовій техніці, промислових приладах. В серії можна відмітити наявність 16-бітного ШИМ контролера для управління двигуном, в старшій моделі наявний датчик ємнісного сенсору та HDMI інтерфейс.

Наступна лінійка F1 заснована на ядрі Cortex-M3, має в арсеналі моделі з інтерфейсом USB 2.0, Ethernet IEEE1588, HDMI інтерфейсом.

Лінійка F2 є найбільш безкомпромісною за ціною та продуктивністю. В своєму арсеналі має інтерфейс USB 2.0, Ethernet, інтерфейс камери.

Лінійка F3, заснована на ядрі Cortex-M4, має на борту швидкий сигма-дельта АЦП з програмованим коефіцієнтом підсилення, швидкі компаратори, FPU. Лінійка оптимізована для обробки та перетворення змішаних сигналів в ланцюгах, як 3-фазний двигун, біометричних приладах, аудіо фільтрах, системах моніторингу.

Найбільш «фаршированою» та потужною є лінійка STM32F4, в основі якої також лежить Cortex-M4. Вона більш орієнтована на використання в медіа апаратурі, але нема перешкод для використання і в інших пристроях.

Лінійка STM32L це мікроконтролери ультра-низького споживання, має в своєму арсеналі моделі з контролером LCD, ємнісним сенсором, USB 2.0 інтерфейс, сенсор температури та ін.

Лінійка STM32T орієнтована на використання в сенсорних пристроях, оскільки має резистивний сенсорний контролер з дуже малим часом відклику.

Лінійка STM32W орієнтована на використання в бездротових пристроях, оскільки має на борту, крім стандартної для Cortex-M3 периферії, також радіо-модуль 2.4 ГГц IEEE 802.15.4.

Судити про застосування кожної моделі можна виходячи з особливостей периферії. Застосування обмежується лише уявою розробників та потужністю мікроконтролера.

ТЕМА 18. ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА STM32 ТА AVR

Розглянемо переваги та недоліки характеристик мікроконтролерів сімейства STM32 та AVR.

Перше питання яке виникає, а чи коректно порівнювати 8-бітну та 32-бітну архітектуру? Таке питання виникає тому що існує асоціація - 8 бітні МК оперують з 8-бітними даними. Це не зовсім так, ат же прості цілочисельні дані і показчики в 8-ми бітних МК є 16-ти бітними. Тому, перебільшуючи, при роботі з такими даними, МК витрачає додаткові такти там, де 32-бітове ядро витрачає всього один. Плюс для доступу до таких даних 8-ми бітний МК виробляє додаткові операції читання/запису і операції зі стеком. На ділі це додатково призводить і до збільшення обсягу прошивки, і до збільшення споживання пам'яті.

В цьому аспекті STM32 мають перевагу, бо 32-бітові МК можуть легко оперувати 8-, 16- і 32-бітових даними за такт. Плюс наявність готових команд для доступу до них і перетворення.

Щодо продуктивності, тут можна звернутися до тесту CoreMark, загальна продуктивність схожих моделей:

ATmega2560 (на частоті 8МГц): 4.25 CoreMark

ATmega644 (на частоті 20МГц): 10.21 CoreMark

STM32F103RB (на частоті 72МГц): 108.26 CoreMark

STM32F051C8 (на частоті 48МГц): 105.61 CoreMark

Як ми бачимо різниці вражаюча.

З потужністю справи такі самі. У порівнянні з 32-біт, 8-бітові МК виробляють в 4 рази більше циклів звернення до пам'яті та більшу кількість команд для копіювання того ж обсягу даних. Також, наприклад, ARM дозволяють виконувати по 2 команди за такт. У 32-біт менше роботи, менше споживання.

Ще додамо модуль FPU, який є на всіх STM32 починаючи з серій F3xx. Значно прискорює обчислення з плаваючою точкою. І ще потім додамо DMA, який на порядок прискорює роботу з периферією і не тільки.

МК	ATmega328P-AU	STM32F103C8T6
Корпус	TQFP-32	LQFP-48
Память	32kB Flash/2kB SRAM	64kB Flash/20kB SRAM
Макс. частота	20MHz (16MHz Arduino)	72MHz
Основная периферия	1xSPI, 1xI2C, 1xUSART	2xSPI, 2xI2C, 3xUSART
Таймеры	1x16-bit, 2x8-bit	4x16-bit
PWM	до 6 каналів	до 16 каналів
ADC/DAC	1x10-bit, 8 каналів. Плюс 1 канал Temp	2x12bit по 10 каналів. Плюс два каналів Vref и Temp
Доп. периферия	QTouch	RTC, USB, CAN, Модуль CRC 32-bit
Доп. информация	EEPROM 1kB	DMA 7 каналів

Цінова політика STMicroelectronics не поступається MicroChip, у таблиці обрані дві моделі за однією ціною, різниця в параметрах очевидна.

Але у STM32 є і недоліки: занадто великий обсяг бібліотек, які рідко бувають корисними та ємкість коду, що для рішення звичайних задач є недоцільним, замість простого запису `digitalWrite(LED1, HIGH)` буде `HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET)`.

ТЕМА 19. РЕГІСТРИ ВВОДУ-ВИВОДУ STM32

В системі STM32 застосовується загальноприйнята назва портів введення-виведення загального призначення, а саме GPIO (General purpose input-output).

Під "загальним призначенням" мається на увазі, що стан виводів встановлюється або зчитується програмно, тобто вивід є входним або вихідним сигналом будь-якого апаратного вузла мікроконтролера.

Порти позначаються GPIOA, GPIOB, GPIOC і т.д. На схемі мікроконтролера відповідні портам висновки позначаються PA0-PA15, PB0-PB15, PC0-PC15 ...

Порти 16ти розрядні, тобто у кожного порту 16 виводів.

Режим і стан кожного виводу можуть бути встановлені окремо, незалежно від інших виводів.

Кожен вивід може використовуватися в режимі:

Input floating - вхід без підтягуючого резистора, кинутий в повітрі вхід.

Input pull-up - вхід з підтягуючим резистором, підключеним до живлення мікроконтролера.

Input pull-down - вхід з підтягуючим резистором, підключеним до загального проводу (землі).

Analog - аналоговий вхід (вхід АЦП, компараторів тощо).

Output open-drain - вихід з відкритим стоком. Функціонально аналогічний виходу з відкритим колектором. При низькому логічному рівні замикає вивід на землю, при високому - кидає в повітрі.

Output push-pull - звичайний активний вихід. При низькому логічному рівні напруга на виводі дорівнює 0, при високому - напруга близько до напруги живлення мікроконтролера, зазвичай + 3 В.

Alternate function push-pull - альтернативна функція виведення в звичайному (активному) режимі.

Alternate function open-drain - альтернативна функція виведення в режимі відкритий стік.

Існує система захисту конфігураційних параметрів порту. Для її активізації необхідно виконати певну послідовність дій над регістрами конфігурації. Розблокування відбувається тільки після скидання.

У нашому мікроконтролері STM32F103C4 існують 37 портів введення-виведення:

GPIOA - 16 виводів: PA0-PA15;

GPIOB - 16 виводів: PB0-PB15;

GPIOC - 3 виводів: PC13-PC15;

GIOD - 2 виводів: PD0, PD1.

Програмне управління портами STM32.

Для кожного порту (16 виводів) є два 32х розрядних регістра конфігурації. Вони утворюють 64 бітний регістр конфігурації порту.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		MODE7[1:0]		CNF6[1:0]		MODE6[1:0]		CNF5[1:0]		MODE5[1:0]		CNF4[1:0]		MODE4[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		MODE3[1:0]		CNF2[1:0]		MODE2[1:0]		CNF1[1:0]		MODE1[1:0]		CNF0[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Молодший регістр конфігурації портів GPIOx_CRL (Port configuration register low)

Старший регістр конфігурації портів GPIOx_CRH (Port configuration register high)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF15[1:0]		MODE15[1:0]		CNF14[1:0]		MODE14[1:0]		CNF13[1:0]		MODE13[1:0]		CNF12[1:0]		MODE12[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF11[1:0]		MODE11[1:0]		CNF10[1:0]		MODE10[1:0]		CNF9[1:0]		MODE9[1:0]		CNF8[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

На кожен вивід відводиться 4 біта, які діляться на 2 бітні поля режиму і конфігурації.

Поле режиму визначає напрямок роботи виведення - вхід або вихід. У разі виходу воно задає максимальну частоту передачі вихідного сигналу. Очевидно, це впливає на енергоспоживання мікроконтролера.

Стан регістрів конфігурації можна захистити від непередбачених змін. Для цього існує регістр блокування конфігурації.

Регістр блокування конфігурації порту GPIOx_LCKR (Port configuration lock register)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Кожному виводу порту відповідає біт блокування LCK0 - LCK15. При установці біта в 1 забороняється зміна відповідних бітів режиму і конфігурації. Після завдання всіх потрібних бітів регістра блокування необхідно активізувати захист. Для цього треба в 16й біт регістра блокування (LCKK) послідовно записати 1, 0, 1. Після цього блокування буде діяти і зміна захищених бітів конфігурації та режиму можливо тільки після скидання мікроконтролера.

Перевірити чи встановити захист конфігурації можна читанням цього ж 16-го біта (LCKK). При активованій захисту два поспіль читання біта повинні дати результат 0, 1.

Доступ до портів відбувається через регістри введення і виведення даних.

Регістр введення даних порту GPIOx_IDR (Port input data register)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Регістр виведення даних порту GPIOx_ODR (Port output data register)

Перший доступний тільки для читання. Біти IDR0 - IDR15 містять стан відповідних виводів порту.

Запис даних у другий регістр встановлює стан виводів порту (ODR0 - ODR15).

Якщо заданий режим виведення - вхід, то стан відповідного біта ODR0 - ODR15 визначає, куди підключений підтягаючий резистор, до шини харчування або землі.

Для управління станом окремих виводів можна скористатися регістрами бітової обробки.

Регістр установки/скидання бітів GPIOx_BSRR (Port bit set/reset register)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Регістр установки/скидання бітів розділений на 2 рівні частини (0-15 і 16-31). Перша призначена для установки виводів в стан 1, друга - в стан 0.

Запис даного з 1 в бітах BS0 - BS15 встановлює відповідні виводи в стан високого рівня.

Запис даного з 1 в бітах BR0 - BR15 переводить відповідні виводи в стан низького рівня.

Таким чином, одним зверненням до регістру можна встановити або скинути потрібні виводи, не змінюючи стан інших. Якщо проводити цю операцію за допомогою регістра виведення даних, то потрібно операції читання, модифікація і запис.

Регістр скидання бітів GPIOx_BRR (Port bit reset register)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Це 16-ти розрядний регістр, який працює точно так, як молодша половина попереднього регістру установки/скидання. Запис слова з 1 в бітах BR0 - BR15 переводить відповідні виводи в стан низького рівня.

Питання для самоперевірки

1. Які існують режими включення портів?
2. Які режими застосовують для передачі вихідного сигналу?
3. Що впливає на енергоживлення мікроконтролеру?
4. Які регістри відповідають за напрямок направлення сигналу?
5. Які біти відповідають за стан направлення сигналу?

6. За допомогою яких реєстрів можна перевести стан виводів у низький рівень?

ТЕМА 21. МЕТОДИ ПРОГРАМУВАННЯ ПРИСТРОЇВ З ЧПУ

Існують три методи програмування обробки для верстатів з ЧПУ:

- ручне програмування;
- програмування на пульті УЧПУ;
- програмування за допомогою CAD / CAM-системи.

Ручне програмування є досить виснажливим заняттям. Однак все технологи-програмісти повинні мати гарне уявлення про техніку ручного програмування незалежно від того, як насправді вони працюють. У нашій країні існує ще чимало підприємств, на яких використовується метод ручного програмування. Дійсно, якщо завод має кілька верстатів з ЧПУ, а виготовляються деталі прості, то грамотний програміст здатний досить успішно працювати і без засобів автоматизації своєї праці.

Метод програмування на пульті УЧПУ набув особливої популярності лише в останні роки. Це пов'язано з технічним розвитком систем ЧПУ, поліпшенням їх інтерфейсу і можливостей. В цьому випадку програми створюються і вводяться прямо на стійці ЧПП з використанням клавіатури і дисплея. Сучасні системи ЧПУ дійсно дозволяють працювати дуже ефективно. Наприклад, оператор верстата може зробити верифікацію УП або вибрати необхідний постійний цикл за допомогою спеціальних піктограм і вставити його в код УП. Деякі системи ЧПУ пропонують інтерактивна мова програмування, який значно спрощує процес створення УП, робить «спілкування» з ЧПУ зручним для оператора.

Третій метод - програмування за допомогою CAD / CAM-системи - дозволяє «підняти» процес написання програм обробки на більш високий рівень. Працюючи з CAD / CAM-системою, технолог-програміст позбавляє себе від трудомістких математичних розрахунків і отримує інструменти, які значно підвищують швидкість написання УП.

Сьогодні для досягнення успіху на ринку промислове підприємство змушене працювати над скороченням терміну випуску продукції, зниженням її собівартості і підвищенням якості. Стрімкий розвиток комп'ютерних та

інформаційних технологій призвело до появи CAD / CAM / CAE-систем, які є найбільш продуктивними інструментами для вирішення цих завдань.

Під CAD-системами (computer-aided design - комп'ютерна підтримка проектування) розуміють програмне забезпечення, яке автоматизує працю інженера-конструктора і дозволяє вирішувати завдання проектування виробів і оформлення технічної документації за допомогою персонального комп'ютера.

Для використання САМ-системи, технолог-програміст застосовує персональний комп'ютер або робочу станцію. Комп'ютерна програма автоматично генерує керуючу програму (G-код). Потім керуюча програма передається тим чи іншим способом в пам'ять стійки верстата з ЧПУ.

САМ-системи можна розділити на дві категорії - з мовним і графічним способом введення інформації. Використовуючи перші, технолог зобов'язаний використовувати мову програмування, подібний Бейсик або С (сі). Ці САМ-системи вимагають програмування і деякі з САМ-систем в силу цього дуже складні для освоєння.

На іншому полюсі розташовуються САМ-системи, де кожен крок обробки задається інтерактивно в графічному режимі. Програміст має зорову зворотний зв'язок протягом кожного кроку завдання програмування. Тому в загальному випадку такі системи більш прості у вивченні і роботі.

ТЕМА 22. СИСТЕМИ КООРДИНАТ ВЕРСТАТІВ З ЧПУ

Для обробки заготовок на верстатах з ЧПУ, також як і на універсальних верстатах, необхідно повідомити різального інструменту та заготівлі певний, як правило досить складний, комплекс узгоджених між собою рухів:

Основні рухи - це рухи виконавчих органів верстата, завдяки яким безпосередньо здійснюється процес зняття стружки ріжучим інструментом з оброблюваної заготовки.

Головним рухом у верстаті є рух, який визначає швидкість різання, швидкість зняття стружки з заготовки. Головний рух може бути обертальним або прямолінійним.

Рух подачі - це рух у верстаті, який забезпечує підведення до інструменту нових ділянок заготовки для зняття з них стружки. Рухів подач в верстаті може бути кілька: поздовжня, поперечна, кругова, тангенціальна.

Рухи ділення - це рух у верстаті, який реалізується для здійснення необхідного кутового (або лінійного) переміщення заготовки щодо інструменту. Ділильний рух може бути безперервним і переривчастим.

Рух обкату - це узгоджений рух ріжучого інструменту і заготовки, при формуванні зачеплення певної кінематичної пари.

Диференціальне рух додається до руху заготовки або інструменту. Скласти можна тільки однорідні рухи: обертальний з обертальним, поступальний з поступальним.

Функціонування верстата з ЧПУ в принципі неможливо без використання певної системи координат, за допомогою якої встановлюються просторові координати будь-якої точки в межах робочої зони верстата. Розглянемо, які системи координат використовуються на верстатах з ЧПУ:

Плоскі системи координат

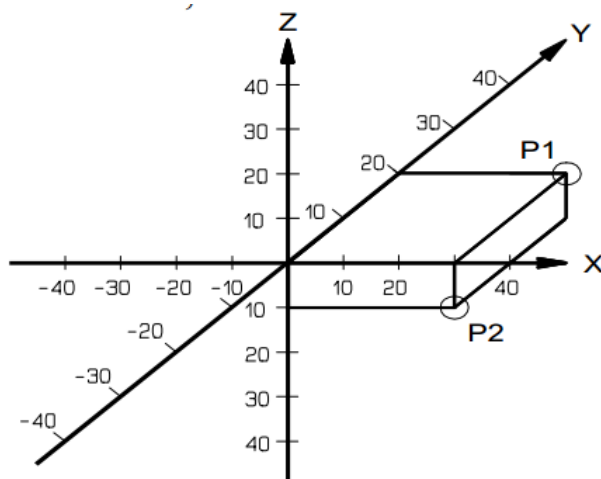
Прямокутна система координат є найбільш поширеною системою координат для верстатів з ЧПУ. Вона містить дві осі координат (двовимірна система) – для визначення положення точок на площині. Для прямокутної системи координат характерні наступні ознаки:

- ☐ координатні осі розташовуються взаємно перпендикулярно;
- ☐ координатні осі мають спільну точку перетину (початок відріку координат);
- ☐ координатні осі мають однаковий геометричний масштаб

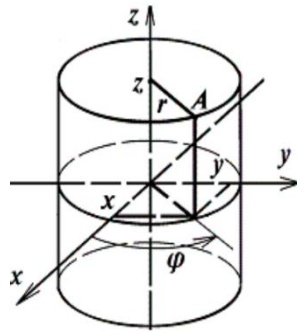
Полярна система координат - двовимірна система координат, в якій кожна точка на площині визначається двома числами - полярним кутом і полярним радіусом. Полярна система координат особливо корисна у випадках, коли відносини між точками простіше зобразити у вигляді радіусів і кутів.

Об'ємні системи координат

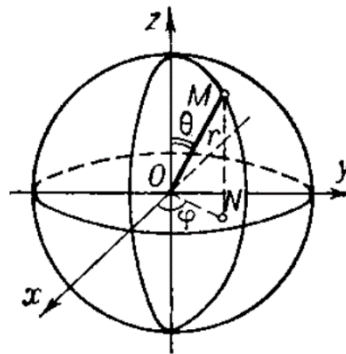
Декартова система координат в просторі (в цьому параграфі мається на увазі тривимірне простір, про більш багатовимірних просторах) утворюється трьома взаємно перпендикулярними осями координат OX , OY і OZ . Осі координат перетинаються в точці O , яка називається початком координат, на кожній осі вибрано позитивний напрямок, вказане стрілками, і одиниця виміру відрізків на осях.



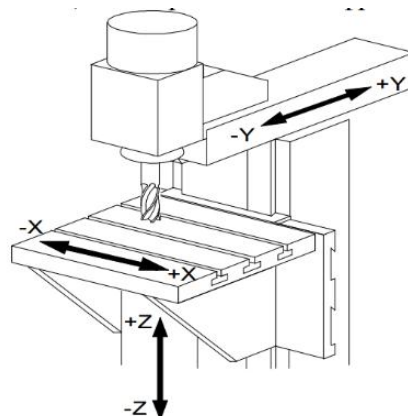
Циліндрична система координат, розширює плоску полярну систему додаванням третьої лінійної координати, званої «висотою» і рівній висоті точки над нульовою площиною подібно до того, як Декартова система розширюється на випадок трьох вимірів. Третя координата зазвичай позначається як, утворюючи трійку координат



Сферичними координатами називають систему координат для відображення геометричних властивостей фігури в трьох вимірах за допомогою завдання трьох координат, де r – відстань до початку координат, θ – зенітний і φ – азимутальний кут відповідно

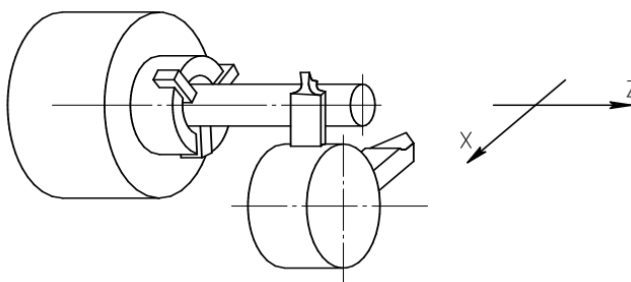


Для фрезерних верстатів з ЧПУ використовується просторова координатна система, як правило, прямокутна. Визначальним фактором для просторового розташування осей координатної системи є орієнтація в просторі осі робочого шпинделя.



Для токарних верстатів з ЧПУ найбільш поширеною є плоска прямокутна система координат з осями Z і X . Як і у випадку з фрезерними верстаками визначальним фактором для просторового розташування осей координатної системи є орієнтація в просторі осі робочого шпинделя, яка

завжди збігається з віссю Z. Позитивним напрямком осі Z є напрямлення від місця кріплення заготовки в шпинделі до ріжучого інструменту.

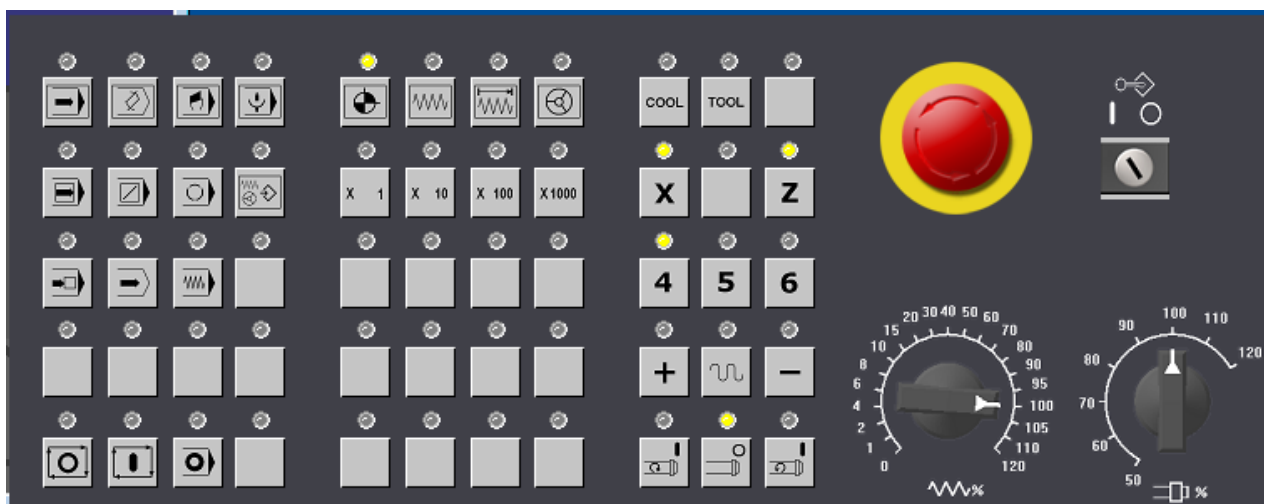


ТЕМА 23. ЕЛЕМЕНТИ УПРАВЛІННЯ FANUC 0IT



POS - кнопка виводу на екран координат розташування різця (абсолютні, відносні або всі);
 PROG - на екран виводиться поточна програма обробки;
 OFFSET SETTING - меню корекції на інструмент;
 SYSTEM - режим для наладки та діагностики верстата. Частіше всього використовується спеціалістами по ремонту ЧПУ верстатів;
 MESSAGE - на екран виводяться повідомлення про помилки. При виникненні аварійної ситуації цей екран виводиться автоматично;

CUSTOM GRAPH - режим графічного контролю (перегляду) УП;
 ALTER - заміна слова УП;
 INSERT - вставка слова УП;
 CAN - стирання останнього введенного символу;
 INPUT - використовується для введення буквено символної інформації;
 DELETE – видалення слова;
 БЛОК ЦИФРОВИХ та СИМВОЛЬНИХ клавіш



	встановлення координат в нуль
	координати по осі X
	координати по осі Z
	режим автоматичної роботи програми
	режим введення програми з зовнішніх пристроїв
	режим для виконання локальних команд
	виконання програми з зовнішньої пам'яті
	обертання шпинделя проти годинникової стрілки
	зупинка шпинделя
	обертання шпинделя за годинниковою стрілкою
	запуск управляючої програми
	зупинка управляючої програми