Коммерциялық емес акционерлік қоғам

АЛМАТЫ ЭНЕРГЕТИКА ЖӘНЕ БАЙЛАНЫС УНИВЕРСИТЕТІ

Кафедра электроники

МИКРОКОНТРОЛЛЕРЛЕР

5В071600 – Прибор жасау мамандығының студенттері үшін зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Алматы 2014

ҚҰРАСТЫРҒАН: К.О. Амантаев. Микроконтроллерлер. 5В071600 – Прибор жасау мамандығының студенттері үшін зертханалық жұмыстарды орындауға бойынша әдістемелік нұсқаулықтар. – Алматы: АУЭС, 2014. – 35 с.

Бұл әдістемелік нұсқауда Atmel фирмасының AVR сериясындағы ATMega8535 микроконтроллерінің архитектурасы туралы, динамикалық сипаттамасы және функционалдық мүмкіндіктері жайында қысқаша мәлімет берілген. AVR STUDIO бағдарламаны пайдалану арқылы ассемблер және Си бағдарламасында қолданушы контроллер жадын өңдеуге, оны ретке келтіруге және сақтауға мүмкіндік береді. AVR-микроконтроллеріне аумағына қатысты танмал периферия және типтік интерфейстармен жүзеге асуыдың сұлбасы мен бағдарламалық шешімдері келтірілген. Әртүрлі функцияларды бағдарламалық жүзеге асырудың мысалдары қарастырылған: EEPROM-жадының ішкі және сыртқы деректері, аналогтық-цифрлық түрлендіруі, кеңжолақты импульсты модуляторыдың (КИМ) көмегімен қозғалтқышты цифрлық басқару, таймер-өлшеуішінің және жітісегмент индикаторының әртүрлі режимдерде жұмыс істеуі.

Әдісемелік нұсқау 5В071600 – Прибор жасау мамандардың студенттеріне арналған лекция материалдарын бекіту мақсатында құрастырылған.

Сұлба 14, кесте 7, әдебиет көрсеткіші – 6 атау.

Пікір беруші: АЭБ кафедрасының доценті Калиева С.А.

«Алматы энергетика және байланыс университеті» коммерциялық емес акционерлік қоғамының 2014 ж. баспаның қосымша жоспары бойынша басылды.

Кіріспе

Әдістемелік нұсқаудың мақсаты Atmel фирмасының ATMega8535 микроконтроллерінде ассемблерде және Си арқылы бағдарламалаудың тәжірибелік дағдыларына ие болуды және микропроцессорлық басқару жүйелерімен әзірлеудің теориялық білімдерін бекіт болып табылады.

Команда жүйесі және AVR ядросының архитектурасы IAR Systems-нің жоғарғы дәрежедегі бағдарламалау тілімен компилятор әзірлеуші-фирмасы арқылы жасалды. Нәтижесінде AVR-бағдарламаларын ассемблер тілінде жазылған бағдарламалармен салыстырғанда Си тілінде тезірек және қысқа жазу мүмкіндігі пайда болды.

PIC тобындағы микроконтроллерлер команданы 4 тактта орындаған кезде, AVR сериялы микроконтроллерлерде көптеген командалар бір тактта орындайды.

Алайда, сонымен қатар командалар көлемі көбейді: кейде әрекет ету шектелген облыстарда мекендеу әдісінің әрқайсысына дерлік өз командасы пайда болды.

32 жедел тіркелімнің болуы, бірнеше жағдайларда жедел жадыға мүлдем жүгінбеуге және стекті қолданбауға мүмкіндік береді. Си тілін пайдалану іс жүзінде микроконтроллерлердің әр түрлі құрылымдарын бағдарламалау ерекшелігі көзқарасы бойынша теңестіреді.

AVR сериялы микроконтроллерлер құрылысының әмбебаптығымен, контроллердің әр түрлі түрлеріне құрылымының сабақтастығымен, сұлба техникасын және бағдарламасын микросұлбаға «тігу» оңайлығымен ерекшеленеді.

1 Зертханалық жұмыс № 1. AVR STUDIO бағдарламалау ортасы

Жұмыс мақсаты: Atmel AVR Studio фирмасының микроконтроллерлерінің бағдарламалық ортасымен бағдарламалауды және ретке келтірумен танысу.

1.1 Қысқаша мағлұмат

Зертханалық стендтің құрамына AVR ATmega8535 8-разрядты микроконтроллер және оның перифериялық құрылғысын және көптеген фунционалдық мүмкіндіктерін зерттеу үшін қажетті элементтер кіреді (1.1 сурет).

Модульдің беттік панелінде төмендегілер орналасқан:

- кернеудің бар болу индикациясының сәуледиоды бар «Желі» ауыстырып қосқышы;
- микроконтроллердің енгізу-шығару порттарымен байланысты клеммалы микроконтроллер микросұлбасы;

- микроконтроллерге логикалық дабылдар беруге арналған шығу клеммалары бар S1-S8 ауыстырып қосқыштары;

- микроконтроллерге логикалық дабылдар беруге арналған шығу клеммалары бар S9, S10 батырмалары;

- микроконтроллерге аналогты кернеу беруге арналған шығуклеммалары бар RP1 потенциометр;

- 50 Гц төменжиілікті тікбұрышты сигнал генераторының микросұлбасы және генератордың шығу клеммасы;

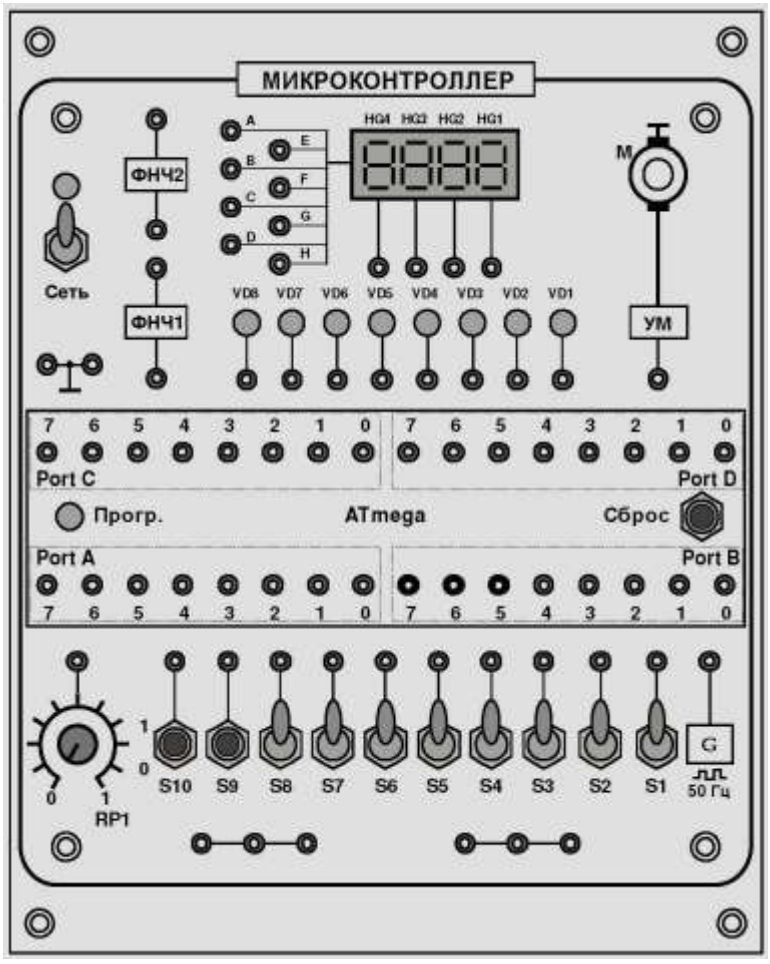
- кернеу көзіне қосуға арналған клеммалары бар VD1 - VD8 сәуледиодтары (мысалы, микроконтроллерге);

- қуат күшейткіші және оған басқарушы кернеу беру үшін клеммасы бар М тұрақты ток электр қозғалтқышы;

- A, B, C, D, E, F, G, H сегменттеріне, сонымен қатар индикатордың әр сегменттің ортақ нүктесіне кернеу беру клеммалары бар 7-сегментті 4 таңбалы индикатор;

- микроконтроллер шығысында ЕИМ-таңбаларды сүзуге арналған төменгі жиіліктегі 2 сүзгіш.

Модульдың сырт жағында 220В 50 Гц кернеуін беруге арналған ажырату, сонымен қатар модульді ДК-ге USB интерфесі бойынша қосуға арналған ажырату орналасады.



1.1 сурет - «Микроконтроллер» модулінің ішкі түр

1.1 кесте - ATmega8535 микроконтроллерінің қысқаша сипаттамасы

Параметр	Мағынасы
Кварцтық резонатор жиілігі	8 МГц
Электр қоректенудің кернеуі	2,7 - 5,5 В
Ішкі Flash – жадының көлемі	8 кБайт
Энергиядан тәуелсіз жадының көлемі	512 Байт
Ішкі ЖЖҚ көлемі	512 Байт
32 бағдарламаланатын кіріс/шығыс	4 портта 32
JTAG - интерфейс	жоқ
8-биттік таймерлер/ЕИМ бар санауыштар	2 дана
16- биттік таймерлер/ЕИМ бар санауыштар	1 дана
10-разрядтық аналогты-сандық түрлендіргіш	бар
АСТ каналдар саны	8
Аналогты компаратор	бар
Ішкі үзілім көздері	3 дана
USART әмбебап қабылдау таратқыш	бар
SPI -интерфейс	бар
TWI -интерфейс	бар

Бағдарламалар жазудың танымал әдісіне төмендегілер жатады:

а) Микроконтроллердің машиналық кодында.

Бағдарламалар ең тезірекетті болып табылады, алайда оларды жазу үшін бағдарламашының жоғары біліктілігі мен процессордың құрылымын терең білуі талап етіледі.

б) Ассемблерде.

Ассемблерде бағдарлама жазу да бағдарламашының жоғары біліктілііін талап етеді. Әдетте ассмблердің тілі шағын процессордың нақты бір түріне қатты байланғанын және бір өндірушінің әр түрлі шағын процессорлары үшін айтарлықтай ерекшеленуі мүмкін екенін атап өткен жөн.

в) Жоғары деңгейдегі тілде (мысалы үшін, Си).

Мұндай бағдарламалар әдетте кросс-платформалы болып табылады, яғни іс жүзінде әр түрлі типтегі шағын процессорлар үшін синтаксисі бойынша ерекшеленбейді. Пайдаланушы бағдарламаны жоғары деңгейдегі тілде жазады, ал компилятор оны ассамблерге және нақты шағын процессордың машиналық кодына айналдырады.

1.1.1 AVR Studio ортасында жоба құру.

Ассамбле рде бағдарламалау үшін AVR Studio v4 бағдарламалық жасақтамасы пайдаланылады. Бағдарлама жазу үшін жоба құру қажет.

1. AVR Studio бағдарламасын қосу. Бағдарламаны қосуға арналған таңбаша жұмыс үстелінде немесе «Пуск» мәзірінде болады. Пайда болған

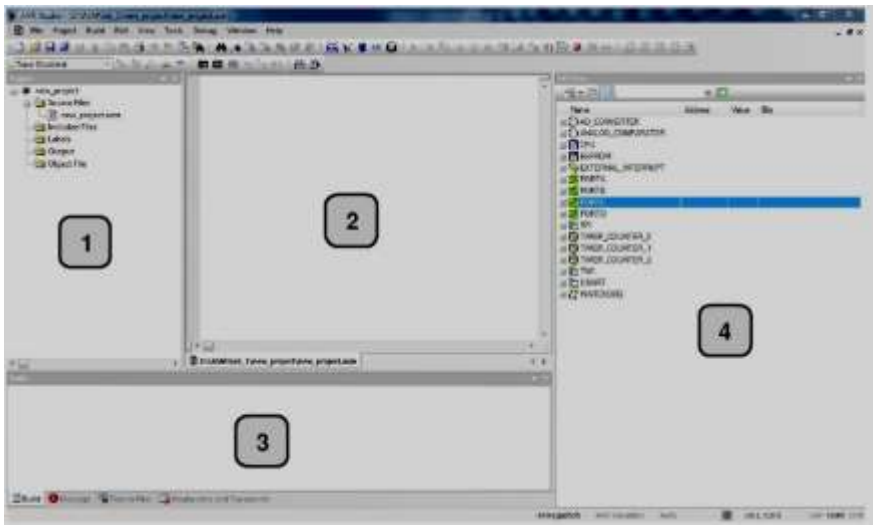
бағдарламаның сәлемдесу терезесінде жаңа жоба немесе бұрыннан бар жобаны ашу ұсынылады.

2. Жаңа жобаны таңдағанда, бағдарламаның тілін - Atmel Avr Assembler немесе Avr GCC таңдау ұсынылатын терезе пайда болады (1.2-сурет). Бұл жерде Atmel Avr Assembler-ді таңдап, жобаның атын және *.asm. үлгаюы бар инициализацияланған файлдың атын көрсету қажет. Жоба аты мен инициализацияланған файлдың атының сәйкес келуіне кеңес беріледі. Файлдың, жобаның немесе жол атында кириллица таңбаларын жібермеу өте маңызды болып табылады. Осыдан кейін «Next» батырмасын басу қажет.

1.2 сурет – Бағдарлама тілін таңдау

3. Бағдарлама тілі мен жоба атын таңдағаннан кейін платформа (Debug platform) және құрылғы (Device) таңдау терезесі пайда болады. Бұл жерде AVR Simulator және жұмыста қолданылатын ATmega8535 процессорын таңдау қажет.осыдан кейін «Finish» батырмасын басу қажет.

4. «Finish» батырмасын басқанда бағдарламада бірнеше облыстар ашылады (1.3 сурет)



1.3 сурет - AVR Studio бағдарламасының жұмыс терезесі

Бұл:

- «Project» жобасының«1» терезесі. Бұл жерде аты және қосылған файлдар мен кітапханалар тізімінен тұратын жобаның құрылымы көрсетіледі;
- тікелей бағдарламаны теру жүзеге асырылатын орталық жұмыс облысы («2»терезе);
- «built» бағдарламаның ағымдағы хаттар терезесі(«3» терезе);
- «I/O View» микроконтроллерінің тіркелім терезесі («4» терезе).

1.2 Жұмыстың орындалу тәртібі

1. Микроконтроллердің PORTC-да 01010101 немесе 10101010 сандар тұжырымын жүзеге асыратын бағдарламаны PORTB0 кірісіндегі дабыл жағдайына байланысты теру талап етіледі.

ATmega8535 үшін бағдарлама листингтің төмендегі түрі бар:

```
;Кірістер: PORTB0
;Кәрәсер: PORTC0...PORTC7
.include "m8535def.inc" ; ATmega8535 стандартты кітапханасының қосылуы
.cseg                   ; сегмент кодының басы
.org 0
reset:
ldi r16,0xFF
out DDRC,r16        ; PORTC шығысқа тағайындау
clr r16
out DDRB,r16       ; PORTC енгізуге тағайындау
main:
sbis PINB,0           ; егер PINB0 -де  – логикалық «0», онда
rjmp PINB0_is_0     ; "PINB0_is_0"-ге өту
ldi r16,0xAA          ;әйтпесе PORTC-ға шығару
out PORTC,r16        ; 0xAA сандар (1010 1010)
rjmp main            ;әрі қарай –main-ға қайту
PINB0_is_0:          ;егер PINB0 – логикалық «0», онда
ldi r16,0x55          ; PORTC-ға шығару
out PORTC,r16        ; 0x55 сандар (0101 0101)
rjmp main            ; әрі қарай –main-ға қайту
```

Бағдарламаны тергеннен кейін оның ассемблерленуін жасау қажет, яғни құрастыру, онда ассемблер жазылған бағдарламаның синтаксисін жасайды және қате болмаған жағдайда ассемблер кодын шағын процессордың машиналық кодына түрлендіреді. Сонымен бірге ары қарай микроконтроллерге жазылатын *.hex ұлғаюы бар файл жасалады.

Жазылған бағдарламаны ассемблерлендіру үшін F7 «Assemble»батырмасын басу қажет. «built» AVR Studio ағымдағы хаттар

терезесінде қате болмаған жағдайда операцияның сәтті аяқталғаны жөнінде көрсетіледі, ал қате болған жағдайда олардың тізімі мен бағдарлама мәтініндегі жағдайы шығады.

2. Бағдарламаны симуляторда ретке келтіру.

AVR Studio бағдарламалық жасақтамаға бақылаушы микросызбасының тікелей бағдарламалауын жүзеге асырмай бағдарламаны реттеуге, алгоритмде қателер табуға және дәлсіздікті жөндеуге болатын микроконтроллерлердің симуляторы орнатылған. Сонымен қатар симулятор үй жағдайларында бағдарламаларды жазу кезінде қажетті болуы мүмкін.

AVR Studio симуляторы микроконтроллердің, оның порттарының, таймерлерінің, аналогты-санды түрлендіргіштердің, үзілімдердің және т.б. симуляциясын жүзеге асырады. Симулятор жобаның hex-файлымен жұмыс істегендіктен, эмуляторды іске қосу үшін бағдарлама жазып, одан hex-файлды жасауға мүмкіндік бермейтін анық қателерді жою керек.

Бағдарламаны симуляторда реттеу үшін оны жазып болған соң Ctrl+F7 «Assemble and Run» пернелердің үйлесімін басу керек, осыдан кейін бағдарламаны ассемблирлеуі және эмулятордың іске қосылуы болады.

Сәтті компиляциядан кейін бағдарламаның басталуы сары нұсқармен белгіленеді. Бағдарламаны реттеу үрдістерімен басқару үшін AVR Studio меню жолында «debug» мәзірі мен симуляторды басқарудың функционалді батырмаларының тақтасы орналасқан.

1.2 кесте – Эмуляторды басқару батырмасын белгілеу

Атауы	Пернелердің үйлесімі	Белгілеуі
Start Debugging (реттеуді бастау)	Ctrl+Shift+Alt+F5	Бағдарламаны эмуляторда реттеу үрдісін бастау
Stop Debugging (реттеуді тоқтату)	Ctrl+Shift+F5	Эмуляторда бағдарламаны реттеу үрдісін тоқтату
Run (эмуляцияны іске қосу)	F5	Эмулятор іске қосылады және экранда контроллердің әдеттегі өзгерулерінің бейнелеуінсіз эмуляцияны циклдік түрде жүргізеді.
Break (эмуляцияны тоқтату)	Ctrl+F5	Команда эмуляторды деректерді жоғалтуынсыз уақытша тоқтатады
Reset (эмуляцияны тоқтату)	Shift+F5	Команда эмуляторды деректерді жоғалтумен уақытша тоқтатады
Step into (алға қадам жасау)	F11	Команда тек бір нұсқаулықты шағарады.
Auto Step (авто орындау)	Alt+F5	Команда эмуляцияны қадамды автоматты тәртіпте орындайды.
Toggle breakpoint (тоқтату нүктесі)	F9	Команда ішінде бағдарлама тоқтатылатын тоқтату нүктесін қосады
Remove all program breakpoints	-	Команда бағдарламаны тоқтатудың барлық белсенді нүктелерін өшіреді.

Бағдарламаны симуляциялау кезінде Step into (1.2 кесте) командасын пайдалана отырып, нұсқаулықты қадамды түрде орындау ұсынылады.

Сонымен бірге микроконтроллер регистрінің және енгізу/шығару порттарының ішіндегісін бақылау қажет.

Регистрлер қалпын және микроконтроллердің жеке құрылғаларын бақылау «4» терезеде жүзеге асырылады. Әр құрылғыны ашып, оның басқару және бақылау регистрлерінің ішіндегісін көруге болады.

3. Бағдарламаны микроконтроллерге жазу

Ол үшін қажет:

- «Микроконтроллер» модулінің «Желі» ауыстырып қосқышын оған қоректендіру кернеуін беру үшін қосу.

- «Tools» AVR Studio мәзіріне «Program AVR» пунктін қосу онда «Connect» біріктіру әдісін көрсету.

Select AVR Programmer пайда болған терезесінде келесіні белгілеу:

- программатор типі STK500;

- ПК диспетчерін иелендірген сол COM port-ын белгілеу;

- Connect үстіңгі батырмасын басу, содан кейін микроконтроллердің AVR-Studio ортасына қосылуы жүзеге асырылады, және контроллердің бағдарламалау терезесі пайда болады;

- Бағдарламалау терезесінде main салымын таңдау керек, онда Device and Signature Bytes терезесінде - ATmega8535 шығару керек;

- «Program» салымына өтіп, онда «Flash» бағанын таңдау.

Бұл бағанда жобаның .hex файлына жолды көрсетіп, содан кейін

«Program» батырмасын басып микроконтроллерге бағдарламаны жазу керек.

Бағдарламаны жазу аяқталған соң берілген алгоритмге сәйкес микроконтроллердегі оның жұмысын тексеру керек.

1.3 Есеп мазмұны

Жұмыс мақсаты.
Бағдарлама листингі.
Қорытындылар.

1.4. Бақылау сұрақтары

1) Қалай AVR Studio ортасында жоба құрылады?
2) Бағдарламаны жинау қай жерде іске асырылады?
3) Бағдарлама синтаксисін тексеру қалай жүзеге асырылады?
4) AVR симуляторының негізгі белгілеуі қандай?
5) hex-файлды қалай алу керек?
6) hex-файлын микроконтроллерге жазу қалай жүзеге асырылады?

2 Зертханалық жұмыс № 2. Стек және уақытты бағдарламалық кідірту

Жұмыс мақсаты: стекті пайдалану бойынша теориялық және тәжірибелік материалды игеру, уақытты кідіртумен бағдарламаларды жазудың тәжірибелік дағдыларына ие болу.

2 Қысқаша мәліметтер

2.1 Стек көрсеткіші

Стекті көрсеткіш – бұл «Last In - First Out» LIFO (соңғы келдің – бірінші болып шығасың)қағидасы іске асырылған буфер болып саналатын арнайы регистр. Стек екі /-разрядты SPH, SPL регистрлерінде жүзеге асырылған.

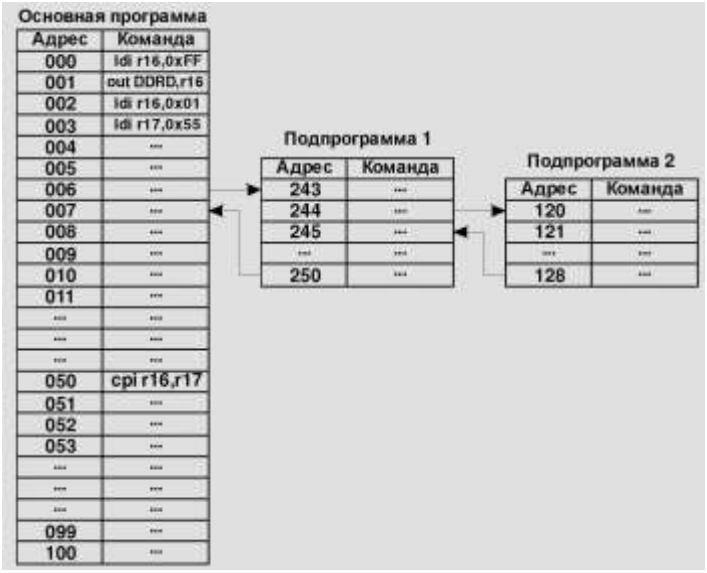
Стек биіктігінің адресін пайдаланушы өз бетімен көрсету керек, сонымен бірге стек биіктігі ОЗУ санашықтың соңында орналасқанын көпшілік мақұлдаған. Бұл бағдарлама аймағы кездейсоқ стек аймағымен жабылып қалмасын деп жасалады, себебі ондай жағдай болса, стек аймағына жазылған қайтару адресі жоғалып кетеді.

Бағыныңқы бағдарламаны орындау кезінде келесі өтуге басқа бағыныңқы бағдарламаға сұрату болатын жағдай болуы мүмкін (2.1 сурет).

Негізгі бағдарламадан 1 бағыныңқы бағдарлама өту болған кезде, стек басына 007 қайтару адресі жазылады. Егер әрі қарай 1 бағыныңқы бағдарламадан 2 бағыныңқы бағдарламаға өту болса, онда стекке 1 бағыныңқы бағдарламаға қайтарудың 245 адресі жазылады.

2 бағыныңқы бағдарламасы аяқталған кезде бірінші болып 245 адресі, ал содан кейін 1 бағыныңқы бағдарламасы аяқталған кезде 007 адресі оқылады.

rcall нұсқамасын шақыру кезінде SPH:SPL регистріне бағыныңқы бағдарламадан қайтару адресі сақталатын жад ұяшығының адресі жазылады да, қайтару адресі автоматты түрде стекка жазылады, ал ret нұсқамасын шақырған кезде регистрде көрсетілген SPH:SPL адресі бойынша стектен оқу жүзеге асырылады.



2.1 сурет – Стекті толтыру қағидасы

Стекті нұсқағышын көрнекі пайдалануларының бірі уақытты бағдарламалық кідіртуді жүзеге асыру болып табылады.

Бағдарламалық кідіртудің мәні санашықты циклдік түрде бірдей әрекетті жасауды мәжбүрлеу болып табылады, мысалы бағдарламаны әрі қарай орындауға бұл әрекетті орындаудың аяқталуы кезінде әрі қарай өтумен нөлден максимумға дейін қандай да бір санның инкременті.

Төмендегі әдіспен жүзеге асырылған бағдарламалық кідірту уақытының жоба есебін көрейік:

```
    clr r16
met_1:
    inc rl6
    cpi rl6,0xFF
    brne met_l
```

РОН r16 регистрінің мағынасы, 16 г мағынасы 0xFF немесе 255 мағынасына жетпегенше 1-ге көбейеді. Сонымен бірге үнемі met_l белгісіне қайту жүреді, және c16 г-қа жазылған мағына 0xFF (немесе 255) мағынасына жеткен кезде ғана бағдарламалардың келесі командаларын орындау рұқсат етіледі. Сөйтіп бағдарлама біраз уақытқа «ілінісіп» қалады. Бұл «ілінісу» бағдарламалық кідірту болып табылады.

Командаларды орындау уақытын біле отырып, осы бағдарламалық кідіртудің уақытын есептеуге болады. Қолданылған командалар келесі есеп уақытына ие:

- clr r16: 1 санашық такті;
- inc r16: 1 санашық такті;
- cpi rl6,0xFF: 1 санашық такті;
- brne met_l: 1 санашық такті, егер шарт орындалмайтын болса және 2 такт, егер жағдай орындалатын болса.

Циклға түскен кезде санашық тактілерінің жалпы саны N=4 болады, яғни 1-ден 255-ке дейінгі сандарды санағанда тактілердің жалпы саны N=4x255=1020 циклдеріне теңеседі. Содан кейін 16 г есебінің аяғында brne

met_i жалған шартын орындау кезінде нақтыланбай қалған бір циклді есептеу керек те clr r 16 командасын орындауға бір такт қосу қажет. Тактілердің жалпы саны барлығы N=4x255-1+1=1020 цикл.

Егер алынған санды f_{osc} (бұл жағдайда – 8 МГц) кварцы резонатордың тербелісінің жиілігіне кері санашықтың бір тактісін орындау уақытына көбейтсе онда Тз кідірісінің уақыты мынаған тең:

$$T_З = N \cdot \frac{1}{f_{osc}} = 1020 \cdot \frac{1}{8 \cdot 10^6} = 127{,}5 \text{ мкс.}$$

Елеулі уақытша кідірістерді жүзеге асыру үшін үлкен сандар немесе енгізілген циклдермен жұмыс жасау керек болады. Енгізілген циклдерді пайдаланумен бағдарламалық кідірістің мысалы төменде көрсетілген: Мысал, жүгіруші оттың бағдарламасын PORTC-қа жазу. Сонымен бірге порт разрядтарын ауыстырып қосуларының арасындағы кідіріс уақыты бағдарламалық кідірістің көмегімен беріледі.

```
        .include"m8535def.inc"     ; Atmega 8535 стандартты кітапханасы
        .cseg   ; код сегментінің басталуы
        . org 0
        ldi r16, low (RAMEND)    ; мекенжай бойынша стек шегіне орналастыру
        ldi rl7,high(RAMEND)     ; ОЗУ-дың үлкен торына spl,rl6
        out sph,rl7
        ldi rl6,0xFF             ; енгізу/шығару порттарын анықтап қосу
        out DDRC,rl6                  ; PORTC - шығаруға
        ldi rl6,0x01             ; 1 санын РОН r16-ға енгізу
main:                       ; бас бағдарламаның басталуы
        out PORTC,r16       ; r16 мәнін PORTC шығару
        in rl7,SREG         ; SREG регистрін сақтау
        rcall delay         ; кешіктіру ішкі бағдарламысын шақыру
        out SREG,rl7        ; delay-ден қайтарудан кейін SREG қалпына келтіру
        rol r16                  ; r16 солға циклдық жылжуы
        rjmp main          ; main-ға оралу
delay:                  ; №1 ішкі бағдарламасы
        clr r18                  ; r18 регистрін тазалау
met1:
        rcall delayl         ; delayl-ға ауысу
        inc r18             ; r18 инкременті
        cpi r18,0x0F         ; r18 мәнін 15 санымен салыстыру
        brne metl           ; егер мән r18≠5 болса, metl-қа ауысу
        ret                ; болмаса – негізгі бағдарламаға қайтып келу
delay1:                         ; №2 ішкі бағдарламасы
        clr r19                  ; r19 регистрін тазалау
met 2:
        rcall delay2           ; delay2-ге ауысу
        inc r19                  ; r19 инкременті
        cpi r19,0xFF         ; r19 мәнін 255 санымен салыстыру
        brne met2           ; егер мән r19≠255 болса, met2-ге ауысу
```

```
        ret                 ; болмаса - №1 ішкі бағдарламаға қайтып келу
delay2:                     ; №3 ішкі бағдарлама
        clr r20             ; r20 регистрін тазалау
met3:
        inc r20             ; r20 инкременті
        cpi r20,0xFF        ; r20 мәнін 255 санымен салыстыру
        brne met3           ; егер мән r20≠255 болса, met3-ке ауысу
        ret                 ; болмаса - №2 ішкі бағдарламаға қайтып келу
```

Бірінші циклге кіргенде шектеулі met1 белгісі мен brne met1 шарты бар r18 регистрінің инкрементінің алдында delay1 ішкі бағдарламасын шақыру жүзеге асырылады және мұнда операция қайталынады және delay2 ішкі бағдарламасына ауысу жүргізіледі. Осы арқылы алдымен delay2 ішкі бағдарламасы циклінің орындалуы, содан кейін ғана delay1 ішкі бағдарламасындағы r19 инкременті жүзеге асырылады, бұдан кейін қайтадан delay2-ге және т.с.с. ауысу жүзеге асырылады.

2.2 Бағдарламаны дисассембрлеу

Бағдарламаны дисассембрлеу «View» AVR STUDIO мәзіріндегі disassembler тармағын таңдау арқылы жүзеге асырылады. Мұндай жағдайда экранда әрбір пәрменнің мекенжайы мен қажетті қызметтік ақпарат көрсетілген бағдарлама пайда болады.

Бір мезгілде экранға мәліметтер жадын (Data) сақтауға қажет Toggle memory window (шақыру үшін Alt+0 комбинациясын теру қажет) жадының мазмұнын бақылау терезесін шығару қажет. Бұл жадының соңғы облысында стектің көрсеткіші орналасатын болады.

2.3 Есептің мазмұны

Жұмыс мақсаты.

Кешіктірудің бағдарламасын жазу және оның жұмысын қалпына келтіру.

Дисассемблер кестесі мен стек құрамын келтіру.

Тұжырымдар.

2.4 Бақылау сұрақтары

1) Стек дегеніміз не?
2) Оның негізгі міндеті мен толтырылу қағидаты қандай?
3) Стек көрсеткішінің функциялары қандай?
4) Стек қандай пәрмендерде пайдаланылады?
5) Ішкі бағдарламаның салыну тереңдігі қандай?
6) Стек қандай жад түріне жатады?
7) Дисассемблерді қандай жағдайларда пайдалану қажет?

3 Зертханалық жұмыс №3. Т0/Т2 таймерлері. Ендік-импульстік модуляция режимі

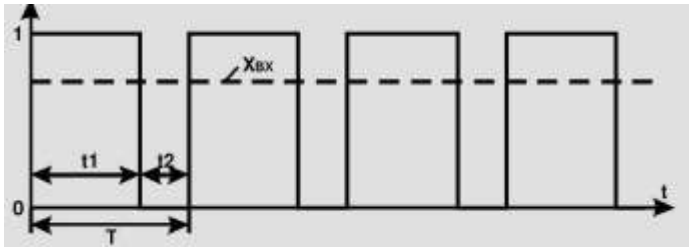
Жұмыс мақсаты: ендік-импульстік модуляция режиміндегі Atmega8535 микроконтроллердің 8-разрядтық таймерлердің жұмысына қатысты теориялық және тәжірибелік материалдарды игеру.

3.1 Теориядан қысқаша мәліметтер

3.1.1 ЕИМ режимі.

Таймерлер/есептеуіштер уақыт интервалдарын есептеу режимінде де, ендік-импульстік модуляция (ЕИМ) режимінде де жұмыс істей алады.

Ендік-импульстік модуляция дегеніміз - логикалық «0» мен модуляция мерзімі ішіндегі логикалық «1» дабылдарының көмегімен дабылдың орташа мәнін қалыптастыру (3.1 сурет).



3.1 сурет – Дабылдың ендік-импульстік модуляцияның қағидаты

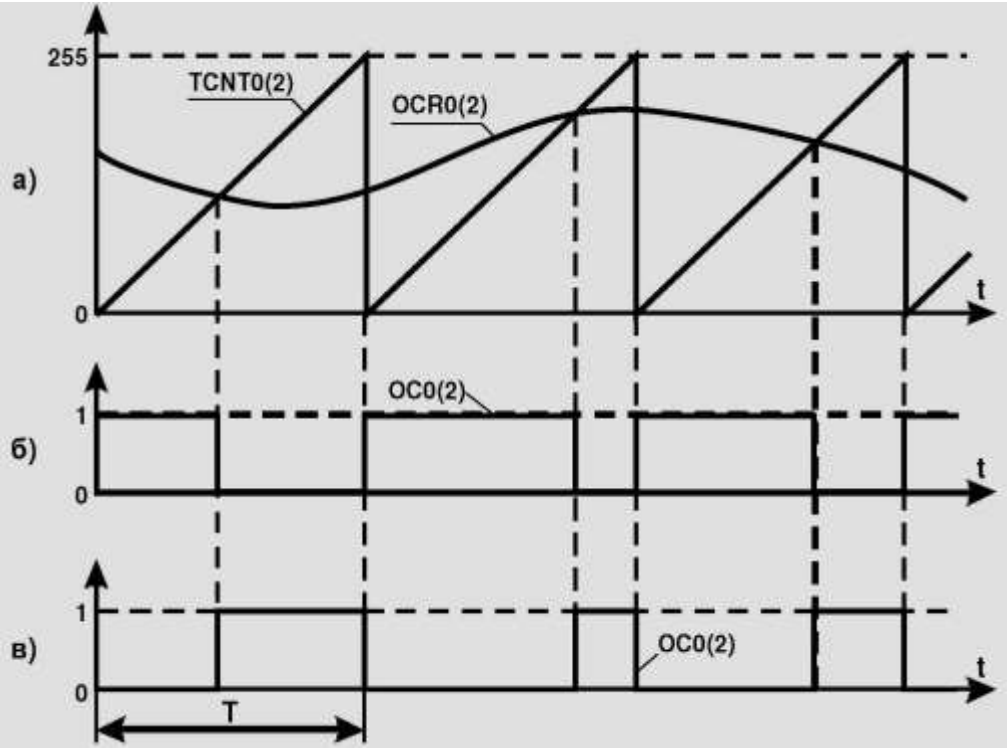
ЕИМ логикалық дабылдардың көмегімен микроконтроллер шығуында 0-ден 1-ге дейін өзгертуге мүмкіндік беретін аналогтік дабылды алуға көмектеседі. Бұл дабылдың мәнін келесі формула бойынша есептеуге болады:

$$X_{\text{BX}} = 1 \cdot \frac{t_1}{T} + 0 \cdot \frac{t_2}{T}.$$

Atmega 8535 микроконтроллердің әрбір таймерінде микроконтроллердің енгізу\шығару порттарына қосылатын және оларды ЕИМ режимінде таймерлердің жұмысы кезінде баламалы функция ретінде пайдаланылатын тұжырымдары бар. Осылайша Т0 таймердің шығуына РВ3 шығуы, ал Т2 таймерінің шығуына PD7 шығуы қосылған.

Atmega 8535 микроконтроллердегі Т0 және Т2 таймерлер екі ендік-импульстік модуляция режимінде жұмыс істейді: жылдам ЕИМ мен кезеңдік-түзетуші ЕИМ.

Жылдам ЕИМ режимінде TCNT0(2) таймердің есеп регистрі Т0 мен Т2 таймерінің TCCR0(2) басқару регистрінде орнатылатын алдын ала бөлгіші бар әрбір импульстің өз мәнін инкременттейтін ара тәріздес ұңғысын (3.2, а сурет) қалыптастыруды жүзеге асырады. Есеп регистрі 255 мәніне жеткен кезде ол автоматты түрде нөлдік көрсеткішке жетеді.



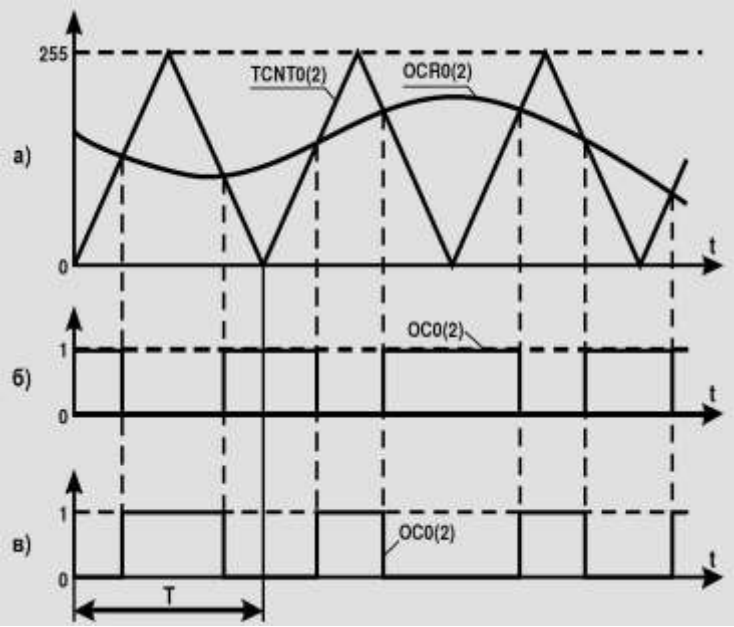
3.2 сурет - «Жылдам ЕИМ» режиміндегі Т0 мен Т2 таймерлер жұмысының қағидасы

Т0 мен Т2 таймерлерін OCR0(2) салыстыру регистрлерінде 0-ден 255-ке дейінгі кез-келген санды жазуға болады. Бұл санмен TCNT0(2) есеп регистрінің мәні салыстырылады және олар тең болған жағдайда, ОС0 немесе ОС2 таймерін TCCR0(2) таймерін басқару регистрінің баптауына сай таймердің шығарылуын ауыстыру жүзеге асырылады. TCNT0(2)=OCR0(2) таймерлері тең келген жағдайда, ОС0(2) таймердің шығуы нөлдік көрсеткішке жетсе, ЕИМ инверттелген емес болады (3.2, б суреті), ал тең болған жағдайда, шығу логикалық «1» қалпында бекітілсе, ЕИМ инверттелген қалыпта болады (3.2, в сурет).

Кезеңдік-түзетуші ЕИМ режимінде TCNT0(2) есеп регистрі үдемелі және төмендейтін фронттары бар ара тәріздес ұңғысының дабылын қалыптастырады. Алдымен регистр өзінің мәнін 0-ден 255-ке дейін инкременттенеді, ал кейінірек оның декременті 255-тен 0-ге дейін төмендейді (3.3, а сурет).

OCR0(2) салыстыру регистрінде бар мәндері TCNT0(2) шот регистрінде мәндерден жоғары болған жағдайда, онда ОС0(2) таймер шығуы нөлдік көрсеткішке түседі, ал ЕИМ инверттелген емес болады (3.3, б сурет), олай болмаған жағдайда, ЕИМ инверттелген болады (3.3, в сурет).

Кезеңдік-түзетуші ЕИМ режимінде ЕИМ жиілігінің дабылы жылдам ЕИМ режиміндегіден төмен болғанына қарамастан, кезеңдік-түзетуші ЕИМ-нің рұқса ету мүмкіндігі жоғарырақ және кезеңдік-түзетуші ЕИМ-ді модуляциясының көбірек нақтылығына жету үшін пайдаланылады.



3.3 сурет - «Кезеңдік-түзетуші ЕИМ» режиміне Т0 мен Т2 таймерлері жұмысының қағидаты

3.1.1.1 ЕИМ режиміндегі Т0 таймерінің регистрлері.

Ендік-импульстік модуляция режимі қалған Т таймері жұмысының режимдері TCCR0 басқару регистрінде көрсетіледі.

3.1 кесте - Т0 TCCR0 таймерін басқару регистрі

бит	7	6	5	4	3	2	1	0
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00

Таймер жұмысы режимін орнатуға жауап беретін биттер WGM01:WGM00. WGM01=0, WGM00=1 орнату кезінде кезеңдік-түзетуші ЕИМ режимі белсендіріледі, ал WGM01=1, WGM00=1 орнату кезінде жылдам ЕИМ режимі белсендіріледі.

Ендік-импульстік модуляция режимін белсендіру кезінде COM01 және COM00 биттер микроконтроллер РВЗ ретінде пайдаланылатын Т0 таймерін шығарудың тәртібін анықтайды. Шығу тәртібі осы биттерге сай 3.2 және 3.3-кестелерінде көрсетілген.

3.2 кесте – Жылдам ЕИМ режиміндегі Т0 таймері РВЗ (ОС0) шығу жағдайы

C0M01	C0M00	Сипаттамасы
0	0	Порттың қалыпты жұмыс істеуі, ОС0 шығаруы өшірілген
0	1	Бит комбинациясы тапсырысқа қойылған
1	0	Үйлесуі кезіндегі ОС0 тазалау. ЕИМ инверттелген емес
1	1	Үйлесуі кезіндегі ОС0 тазалау. Инверттелген ЕИМ

3.3 кесте – Кезеңдік-түзетуші ЕИМ режимінде РВЗ (ОС0) шығу жағдайы

COM01	COM00	Сипаттамасы
0	0	Порттың қалыпты жұмыс істеуі, ОС0 шығаруы өшірілген отключен
0	1	Бит комбинациясы тапсырысқа қойылған
1	0	Шотта үйлесуі кезіндегі ОС0 тазалау. ЕИМ инверттелген емес
1	1	Шотта үйлесуі кезіндегі ОС0 бекіту. Инверттелген ЕИМ

ЕИМ жиілігінің дабылы CS02...CS00 биттарымен бекітіледі. Бұл биттар жиілік тапсырмасы мен бөлгіш коэффициентінің қорын анықтайды (3.4 кесте).

3.4 кесте – Жиілік тапсырмасы қорын бекіту

CS02	CS01	CS00	Сипаттамасы
0	0	0	Таймер тоқтатылған
0	0	1	Қоры – процессор жиілігінің генераторы. (fто = f_{cLk}/1)
0	1	0	Қоры - процессор жиілігінің генераторы. (fто = f_{cLk}/8)
0	1	1	Қоры - процессор жиілігінің генераторы. (fто = f_{cLk}/64)
1	0	0	Қоры - процессор жиілігінің генераторы. (fто = f_{cLk}/256)
1	0	1	Қоры процессор жиілігінің генераторы. (fто = f_{cLk}/1024)
1	1	0	Т0 шығаруындағы сыртқы дабыл. Дабылдың төмендейтін фронты.
1	1	1	Т0 шығаруындағы сыртқы дабыл. Дабылдың үдемелі фронты.

3.4-кестеге сай ЕИМ келесі жолмен саналады:

- жылдам ЕИМ режимі үшін: f = fто/256;
- кезеңдік-түзетуші ЕИМ режимі үшін: f= fто/512.

3.1.1.2 ЕИМ режиміндегі Т2 таймерінің регистрлері.

Ендік-импульстік модуляция режимі қалған Т2 таймері жұмысының режимдері TCCR2 басқару регистрінде көрсетіледі.

Т2 таймерінің TCCR2 регистрінің басқарушы битін сипаттау мен тағайындау Т0 таймерінің TCCR0 регистрі сипатталған биттерге үйлес. ЕИМ жиілігінің дабылы CS22...CS20 биттарымен бекітіледі. Бұл биттар жиілік тапсырмасы мен бөлгіш коэффициентінің қорын анықтайды.

ЕИМ режимін Т0 немесе Т2 таймерінде іске қосу үшін қажет:

- TCCR0(2) басқару регистрін нөлдік көрсеткішке дейін таймерді тоқтату;
- OCR0(2) салыстыру регистрі мен TCNT0(2) есеп регистрін нөлдік көрсеткішке жеткізу;
- TCCR0(2) басқару регистрінде ЕИМ режимін бекіту мен оның жұмысына жиілікті таңдау.

3.2 Жұмысты орындау тәртібі

1. Белгіленген алгоритміне сай бағдарламаны әзірлеу, симуляторда жұмысын қалыпқа келтіру мен микроконтролердің жадына «тігу».

Т0 таймері жедел ЕИМ режимінде жұмыс істейді. Биттердің қалпына байланысты PORTA таймер Т0 ЕИМ шығуына қосылатын светодиод жарықтығы реттеледі. PORTD0 логикалық дабылымен ЕИМ интервенттелгеннен интервенттелген емес негізгі ауысады.

```
; ЕИМ Т0 таймеріне
; енгізулер:
; PORTA0...PORTA7    - таймер бекіту тапсырмасы
; PORTD0   - интервенттелген (1)/ интервенттелген емес (0) ЕИМ
; шығулары:
; P0RTB3    -  ЕИМ Т0 таймеріне
.include"m8535def.inc"   ;ATmega8535 стандартты кітапханасын қосу
.cseg            ; код сегментінің басталуы
.org$0                  ; 0 мекенжайы бойынша
rjmp reset       ; және reset-ке ауысу
reset:
cli      ; барлық бөлінулерге тыйым салу
ldi r16,low(RAMEND)      ; аяқталу мекенжайы стегінің көрсеткішіндегі жазба
ldi r17,high(RAMEND)     ; мәліметтер жады
out spl,r17
out sph,r17
ldi r16,0x01         ; шығаруды енгізу порттарын іске қосу.
out P0RTD,r16              ; PORTD0 – ақпаратты енгізуге
clr r16
out DDRD,r16
ldi r16,0xFF
out P0RTA,r16            ; PORTA - ақпаратты енгізуге
clr r16
out DDRA,r16
out P0RTB,r16
ldi r16,0x08
out DDRB,r16              ; P0RTB3 – ақпаратты шығаруға
clr r16
out TCCR0,r16              ; Т0 таймерін басқару регистрін алып тастау
out OCR0,r16               ; Т0 таймерін салыстыру регистрін алып тастау
out TCNT0,r16              ; Т0 таймері шоты регистрін алып тастау
ldi r16,0x69          , жылдам инверттелген емес режимді инверт бекіту
out TCCR0,r16              ; Т0 таймері ЕИМ
sei                  ; барлық алып тастауға рұқсат беру
main:                ; негізгі бағдарламының басталуы
in r16,PINA          ; PORTA енгізу\шығару порт мәндерін есептеу
```

```
out OCR0,r16            ; және оны ТО таймерін салыстыру регистріне жіберу
in r16,PIND         ; PORTD енгізу\шығару порт мәндерін есептеу
andi r16,0x01           ; және PORTD0 битін бөлу
cpi r16,0x01            ; егер PORTD0=l
breq met1           ; онда met1 белгісіне ауысу
ldi r17,0x69        ; болмаса r17 жазбасы инверттелген емес ЕИМ-ге
rjmp met2           ; және  met2 белгісіне ауысу
met1:                   ; met1 белгісі бойынша
ldi r17,0x79        ; r17 мәні жазбасы инверттелген ЕИМ-ге
met2:                   ; met2 белгісі бойынша
out TCCR0,r17     , T0 таймер жұмысы бекітілген режимін бекіту
rjmp main           ; және main-ге алу
```

3.3 Есептің мазмұны

Жұмыс мақсаты.

Екі режимдегі ЕИМ жұмысының уақытша диаграммасы.

ЕИМ режиміне бағдарламаны жазу мен оның жұмысын қалпына келтіру.

Тұжырымдар.

3.4 Бақылау сұрақтары

1) Ендік-импульстік модуляция дегеніміз не?
2) Жедел ЕИМ дегеніміз қандай режим?
3) Кезеңдік-түзетуші ЕИМ дегеніміз қандай режим?
4) Қандай жағдайларда ЕИМ инверттелген бола алады?
5) Қандай пульсті меандр деп атайды?
6) Үлгіленген дабылдың орташа мәні қалай есептелінеді?
7) Импульстік генератордың қуыстылығы дегеніміз не?

4 Зертханалық жұмыс №4. ICCavr ортасында микроконтроллерді бағдарламалау

Жұмыс мақсаты: Imagecraft ICCavr бағдарламалық жасақтаманы және микроконтроллерді бағдарламалау үрдісін оқып үйрену.

4.1 Қысқаша мәліметтер

4.1.1 ICCavr бағдарламалық жасақтама.

«ImageCraft C» бағдарламасы (бұдан әрі - ICCavr) Си тілінде AVR микроконтроллерлерінде жұмыс істеуге арналып әзірленген бағдарлама. Бұл бағдарламалаудың нәтижесінде құрамында микроконтроллер процессорының пәрмендер коды бар, компиляциядан кейін «*.hex» файлына айналатын,

«*.c» (және жобаның файлы «*.prj») кеңеюіне ие қолданбалы бағдарлама листингі пайда болды.

4.1.1.1 ICCAVR бағдарламасын іске қосу.

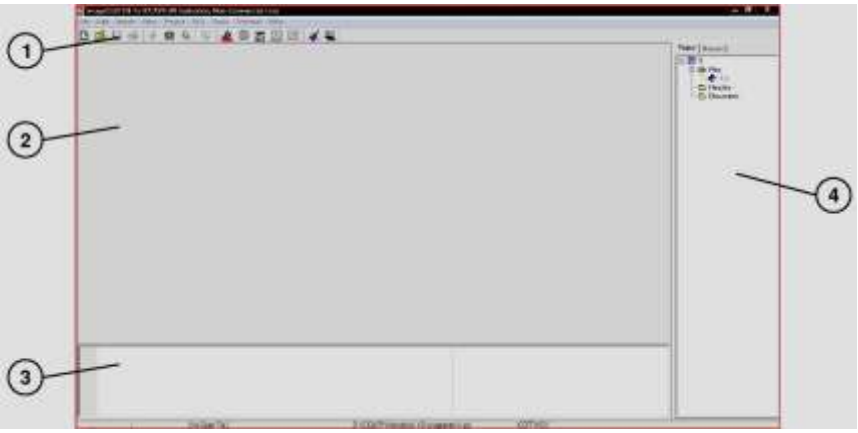
Бағдарламаның іске қосу мәзірінен тауып, оны жүктеу (Іске қосу (пуск) - Бағдарламалар - ImageCraft Development Tools - ICCAVR). Бұдан кейін 4 облысы бар бағдарлама терезесі (4.1 сурет) ашылад:

1 – мәзір батырмалары;

2 – қолданбалы бағдарламаны енгізу облысы;

3 - компиляция облысы;

4 – жоба файлдарының облысы.



4.1 сурет - ICCavr бағдарламасының терезесі

4.1.1.2 Қолданбалы бағдарлама жобасын құрастыру.

Жаңа жобаны құрастыру терезесін ашу: Project мәзірінде New жолын таңдаңыз (бұдан әрі мұндай операциялар Project – New жазылатын болады). Пайда болған «Қапшық» («Папка») жолында жоба сақталынатын қапшықтың атауын, ал «Файл атауы» жолында жобаның атауын, мысалы, «lab1», содан кейін «Сақтау» батырмасын басу қажет. Файлдар облысында жобаның атауы пайда болады.

4.1.1.3 Қолданбалы бағдарламаның үлгісін құрастыру.

Бағдарламаны енгізуді ең дұрысы  белгісі – шеберінің көмегімен немесе бағдарламаның барлық міндетті элементтерін енгізуді жеңілдететін мәзірдегі «Tools - Application Builder» тармағы арқылы бастаған жөн. Экранда шебер терезесі пайда болады.

Бұл ең қарапайым бағдарламада осы терезеге енгізу үшін контроллер түрін («Target CPU» тармағында M8535 таңдау) және кварцты резонатордың жиілігін («Xtal») - 8 МГц енгізу жеткілікті. Жалпы жағдайда осы мәзірде порттардың, таймерлердің, АЦП, UART және SPI бірізді беру құрылғыларының, сыртқы бөлгіштердің жұмыс параметрлерін тиісті салымды көрсете отырып, таңдауға болады. «Options» батырмасы бағдарлама листингіне «main» бас функциясына жолды, ал «Preview» батырмасы бағдарлама үлгісін қарап шығуға мүмкіндік береді. Параметрлерді

енгізгеннен кейін «ОК» батырмасын басу арқылы шебер терезесін жабуға болады.

Бағдарламаның енгізу облысында бастапқы листингі бар қолданбалы бағдарламаснда «Untitled - 0» салымы пайда болады.

Бағдарламаны дискте сақтау қажет: File - Save және файлдың атауын «*.c» кеңейтуін міндетті түрде мысалы, «test.c» түрінде көрсете отырып, енгізу қажет. Бұдан кейін бағдарлама терезесінде элементтердің түсі өзгереді:

а) комментарий символдарының түсі жасыл;

б) қосылатын файлдар символдарының түсі көк;

в) негізгі сөздер жартылай қарайтылған қаріппен және т.с.с. белгіленетін болады.

4.1.1.4 Бағдарламаны өзіндік функциялармен толықтыру.

Бұл кезеңде бағдарламаға, мысалы, портқа дабылдарды енгізіп және кері шығаруға, бөлуді өңдейтіндерді анықтауға, математикалық операцияларды орындауға және т.с.с. арналған өзіндік функцияларды енгізу қажет. Соның ішінде микроконтроллерлердегі бағдарламаның басты функциясының main() құрамында әдетінше оны орындаудың шексіз циклі бар, мысалы, келесі нысанда көрсетілген:

```
void main(void)
{
while(1) {;}
}
```

4.1.1.5 Жобаға файлды қосу.

Жобаға «*.c» кеңейтіуі бар файлды қосу үшін Project - Add Files мәзіріндегі тармақты таңдап, біздің мысалды көрсетілгендей «test.c» файлын бағдарлама файлы түрінде көрсету қажет. Бұдан кейін экранның оң жақ бетінде «Files» жолының жоба файлдары терезесінде қолданбалы бағдарлама файлының атауы пайда болады.

4.1.1.6 Компилятор параметрлері.

Файлды жобаға қосқаннан кейін компилятордың базалық баптау мәндерін орнату қажет. Бұл үшін Project – Options мәзірінде тармақты таңдап, контроллердің қажетті параметрлерін енгізу қажет. Параметрлер терезесі “Options” пайда болады.

Минималды нұсқада параметрлерді таңдау үшін тек контроллер түрін енгізу жеткілікті: «Target» салымының «Device Configuration» жолында «AtMega8535» контроллерін таңдау қажет.

4.1.1.7 Бағдарламаның компиляциясы.

Мәзірде «Build Project» батырмасын немесе құрал-саймандар панелінен тиісті батырманы таңдау қажет. Бұл батырманы басқан кезде бағдарлама листингін, жоба файлының компиляциясын, тұтастыру мен «*.hex» (біздің жағдайда «test.hex») «*.hex» кеңейтуін алатын микроконтроллер пәрмені кодтарындағы орындалатын файлды құрастыруды тексеруді жүзеге асырады. Орындаудың ағымдағы кезеңі, бағдарламадағы қателер туралы хабарламалар бағдарламаның компиляция облысында көрсетіледі. Егер бағдарламада

қателер көрсетілсе, оларды жою қажет. Сонымен бірге компилятор қабылданған бағдарламаның көлемін көрсетеді (бағдарлама микроконтроллерде неше орын алатындығын жалпы көлемінен пайызбен көрсетеді).

4.1.2 Бағдарламаны симуляторда тексеру.

Бұл үшін микроконтроллердің, барлық енгізу\шығару порттардың, есептеуштердің\таймерлердің, бөлгіштердің, ЕИМ мен АЦП жұмысын эмуляциясын жүзеге асыратын «AVR Studio» бағдарламасын пайдалануға болады.

4.1.2.1 Орындалып жатқан файлды «AVR Studio» бағдарламасына жүктеу.

Жұмыс үстелінен AVR Studio бағдарламасының жарлығын тауып ау немесе мәзірдегі «Іске қосу» («Пуск») батырмасын жүктеу (Бағдарламалар - Atmel AVR Tools - AVR Studio). Экранда жаңа жобаны енгізу немесе ағымдағы жобаны ашу ұсынысы келтірілген бағдарламаның диалогтік терезесі пайда болады.

«Open» батырмасын басып, орындалып жатқан әзірленген «*.hex» кеңейтуі бар, мысалы, «test.hex» бағдарламаны ашамыз. Содан кейін бағдарлама бағдарламаның эмуляциясын жүзеге асыру үшін файл құрастырып, оған «*.aps» (біздің жағдайда «test_hex.aps») кеңейтілуі бар атауын береді. Экранда бұл файлды дискке жазуға арналған диалог терезесі пайда болады, «ОК» батырмасын басамыз, содан кейін «Debug platform» платформасы мен «Device» құрылғысын таңдау терезесі пайда болады.

«Debug platform» тармағында «AVR Simulator», ал «Device» тармағында стендте (Atmega8535) қолданылатын контроллер түрін таңдаймыз. «Finish» батырмасын басамыз.

Содан кейін үш жұмыс облысынан тұратын бағдарламаны қалпына келтіру терезесі пайда болады:

-экранның сол жақ бөлігінде енгізу\шығару регистрлерін қарауға мүмкіндік беретін «I/O View» ашу;

- дизассемблеу бағдарламасы көрсетілген «Disassembler» терезесінің оң жақ бөлігінде ассемблер пәрмені түріндегі бастапқа бағдарлама;

- экранның төменгі бөлігінде орындалатын іс-әрекеттер бойынша коментарийлерге арналған «Message» хабарламалар терезесі көрсетіледі.

Алғашқы екі терезені қарастырайық. Дизассемблер терезесінде бағдарлама туралы келесі ақпарат болады:

- әрбір жолда контроллер орындайтын бір нұсқаулақ болады;

- бірінші бағанада пәрмендер жазылатын пәрмендер есептегішінің он алтылық мәні көрсетіледі;

- екінші бағанада пәрменнің он алтылық мәні бар коды көрсетіледі;

- үшінші және төртінші бағаналарда пәрмен мен операнданың символдық атауы көрсетіледі;

- соңғы бесінші бағанада орындалып жатқан пәрмен бойынша қысқа комментарий көрсетіледі.

FFFF пәрмені жазылған жолдарда пәрмен болмайды.

Терезенің сол жақтағы жоғарғы бұрышында ағымдағы орындалып жатқан пәрменнің сары нұсқары (стрелка) орналасқан.

Регистрлер терезесі барлық құрылғылардың ағашынан тұрады (4.2-сурет). Әрбір құрылғыға қарама-қарсы «+» белгісі тұрады. Құрылғына ашу барысында барлық регистрлер көрінеді: басқарушы регистрлер, мәліметтер регистрлері және т.с.с. Мысалы, енгізу\шығарудың А регистрінің құрамында үш регистр бар: PORTA мәліметтерінің регистрі, DDRA бағытының регистрі мен PINA портының шығуы. Порт белгіленуінің оң жағында оның ағымдағы қалпы ол алтылық мәні бар сан мен биттік бейне түрінде шығарылады. Тышқан арқылы биттердің мәнін «0» немесе «1» түрінде белгілеуге болады. Осы арқылы сыртқы дабылдың әсеріне эмуляция жүргізіледі.

4.1.2.2 Эмуляторды іске қосу.

«AVR Studio» бағдарламасы оны нақты уақыт режимінде және қадамдардан тұратын режим арқылы іске қосуға мүмкіндік береді. «Debug» қалпына келтіру мәзірінде бағдарламаны іске қосудың барлық нұсқалары орналасқан, мысалы:

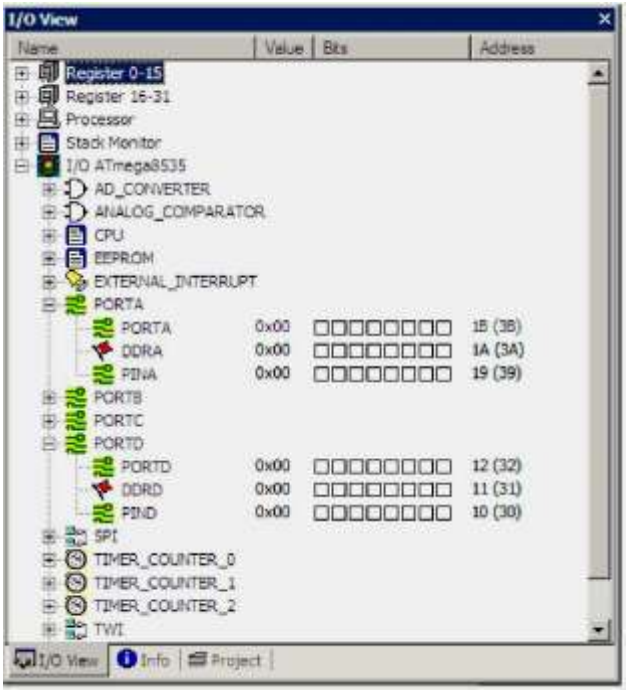
 - Run, бағдарламаны нақты уақыт режимінде іске қосу, нәтижесі бағдарламаны тоқтатқаннан кейін ғана көрінетін болады;

 - Break, бағдарламаны тоқтату, қарап шыққаннан кейін бағдарламаның орындалуын жалғастыруға болады;

 - Reset, бағдарламаның баптауын алып тастау, бағдарлама басында пәрмендер есептегішін орнату;

 - Step Over, қадамдық орындау, бұл ретте бағдарлама әрбір пәрменнен кейін тоқтатылады, нұсқар (стрелка) ағымдағы пәрменді көрсетеді;

 - AutoStep, ағымдағы орындаудың кезеңдерін қарап шығу мүмкіндішімен үздіксіз орындау бағдарламасын іске қосу.



4.2 сурет – Енгізу\шығару регистрлерінің терезесі

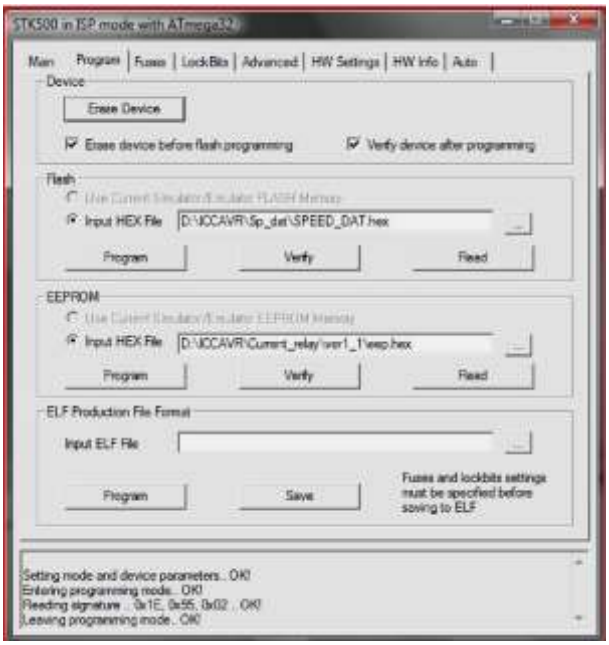
Бағдарламаны  «AutoStep» батырмасын басу арқылы іске қосып, оның үздіксіз орындалуын және уақыт ішіндегі регистрлер индикациясын аламыз. Дизассамблер терезесіндегі сары нұсқар (стрелка) ағымдағы орындалып жатқан батырманы көрсетеді. Бағдарламаның орындалуын «Break» батырмасын басу арқылы тоқтатып, енгізу\шығару регистрлері мәндерін, яғни шығыс дабылдарының түрлі мәндерін белгілеу арқылы өзгертуге болады. Содан ары бағдарламаны «AutoStep» батырмасын басу арқылы қайтадан іске қосып,осы әсерлерге микроконтроллерге реакциясын көруге болады.

4.1.2.3 Бағдарламаны микроконтроллерге жазу.

Бағдарламаныны симуляторда тексергеннен кейін оны міндетті түрде микроконтроллерге жазу керек. Бағдарламаны микроконтроллерге жазу үшін лабараториялық стендте AVR Studio бағдарлама қамсыздандыруы қолданылады. Бұл бағдарлама микроконтроллерді бағдарламалауға, микроконтроллерден бағдарламаны санауға, жазуға және ішкі бағдарламалық ТСҚ микроконтроллерін оқуға, бит бағдарламалауын қорғауға, генератор импульсінің ырғақ режимін және микроконтроллер бастау режимін өзгертуге мүмкіндік береді. Кіріс деретер форматының бағдарламалаушысы hex-файл, Intel (Intel Standard или Intel Extended) фирмасының стандарты болып табылады. AVR Studio бағдарламасының мәзір жолында «Tools - Program AVR - Auto Connect» пунктін таңдау қажет .Егер «Микроконтроллер» модулі дербес компьютерге қосылған және сәйкес драйвер орнатылған болса ,онда бағдарламалаушының шақыру терезесі болып табылады (4.3 сурет) . Бұл терезеде Main қосымша бетіндегі бағдарламаналушы Atmega8535 құралын таңдау қажет, таңдауымызды тексеру керек «Read Signature» батырмасын басып,сәйкестендіру арқылы.

Бұдан кейін «Program» қосымша бетіне өтуге болады(4.4 сурет).Бұл қосымша бетте Flash микроконтроллер ішіндегі мәліметті өшіруге («Erase Device»батырмасы),сондай-ақ Flash және EEPROM микроконтроллер жадына бағдарламалап қоюға болады.

4.3 сурет - AVR Studio бағдарламасыы. Қосымша бет «Main»



4.4 сурет - AVR Studioбағдарламасы. Қосымша бер «Program»

Flash жадының микроконтроллеріне бағдарлама құрастырмасы және жазылуынан алынған hex файл жазуын іске асыру қажет. Ол үшін жазылып жатқан файл таңдау керек *.hex кеңеюіменен және Flash жұмыс терезесіндегі «Program» батырмасын басу қажет. Бағдарлама барысындағы мәлімет терезенің төменгі жағында кескінделеді. Микроконтроллер

бағдарламалауында «Микроконтроллер» модулі алдыңғы панелде орналасқан, түсті диод «Прог» жылтылдау керек.

4.3 Тапсырма

1. «бегущий огонь» бағдарламасын жөрмеп, қарапайым әдіспен уақыт кідірісі жүзеге асады (бағдарлама циклына бос оператор енгізу).

```
// Листинг бағдарламасы 2
#include <iom8535v.h> #include <macros.h>
void port_init(void)
{
PORTD = 0x01;    // 0 бит портын қосу D: PORTD=0000 0001
DDRD  = 0xFF;
}
void init_devices(void)
        {
        CLI ();
        port_init();
        MCUCR = 0x00;
        GICR  = 0x00;
        TIMSK = 0x00;
        SEI () ;
        }
void main(void)
{ unsigned int i;       //айдымалдылығын мәлімдеу i. init_devices();
while(1)
        {
        for (i=l;i;i++) {;}    //Бос цикл
        PORTD=PORTD«l;    //Жаңа порт тағайындалу тапсырмасы  D. if
(PORTD==0)
        PORTD=l; //Егер мәні 0 болса, жеке мән иемден
        }
}
```

4.4 Есептеме мазмұны

Жұмыс мақсаты.
Листинг бағдарламасы.
Қорытынды.

4.5 Бақылау сұрақтары

1) Си тілінің басты артықшылығы неде?
2) «Кросс-платформенный язык» деген нені білдіреді?
3) ICCAVR бағдарламасы неге арналған?
4) AVR Studio бағдарламасы кандай функция орындайды?
5) Бағдарлама құрастырмасы қалай жүзеге асады?

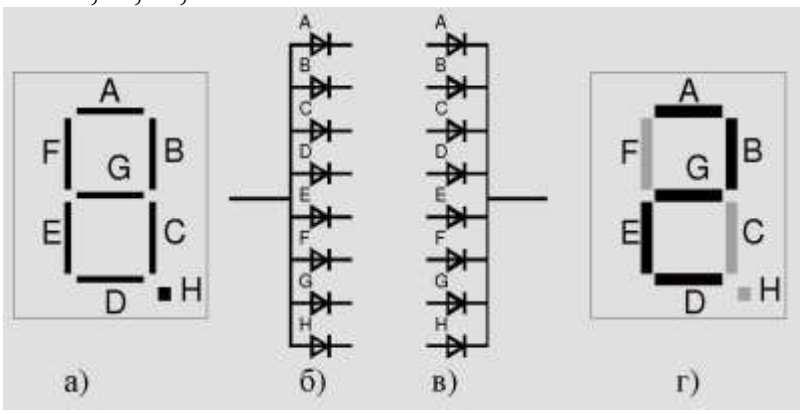
6) hex-файл форматы не үшін қажет?

5 Зертханалық жұмыс №5. Жетісегменттеу жетісегментті индикатормен басқару

Жұмыс мақсаты: жетісегментті түстідиод индикация базасының динамикалық индикациясын оқып үйрену.

5.1 Қысқаша мәлімет

Цифрлық мәліметт индикациясы көбінесе жетісегментті индикатор базасында орындалады. Мұндай индикаторлар өз тістідиод (сегментін) ұсынады A, B ... H (5.1а, суреті). Индикаторлар ортақ анодты (5.1,б, сурет) және ортақ катодты (5.1в, сурет) болады. Кез келген сегмент индикаторын қосу үшін, мысалы, A сегменті ортақ катодты сызбада, осы сегмент анодына кернеу қорегі кедергі арқылы беріліп және нолінші ортақ катод көзімен байланытырамыз. Цифр индикациясы үшін «2» (5.1 г, сурет) анод сегментіне қорек беру қажет A, B, D, E и G.



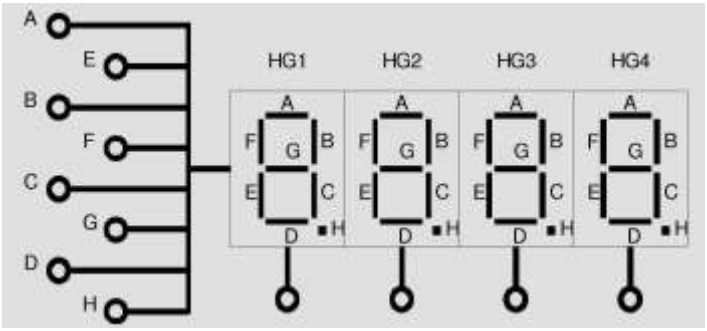
а - жетісегментті индикатор, б –ортақ анодты сызба,
в –ортақ катодты сызба, г - цифр индикациясы «2».
5.1 сурет - Жетісегментті индикатор

Жұмыста жетісегментті 4 индикатор қолданылады HG1...HG4 (5.2 сурет) ортақ катодты сызба бойынша. Қысқарту мақсатында клемма бағдарламасы HG1 ... HG4 катод индикатор түйіндерімен логикалық инвертор арқыла байланысқан. Бұл катод индикаторларын басқаруда дұрыс логикалық қолданыс береді.

Егер HG1, HG2, HG3 индикаторларына логикалық бірліктер берсек, онда кандай да бір A...H сегментінде, мысалы «1», онда ол уш индикаторда бірмезетте іленеді.

Динамикалық индикация индикаторлардағы әр түрлі сандарға жарық түсіруге арналған, мысалы, HG1-HG3 индикаторында «123» сандардың тізбектей: алғашында HG1индикаторына «1» цифрын, одан кейін HG2 екінші индикаторға «2» цифрын және ары қарай HG3 үшінші индикаторға «3» цифрын енгізеді. Егер осы тұжырымдаманы 25 Гц жылдамдықпен

орындасақ, ол адам көзіне көрінбейді. Яғни біздің жағдайда жиілік ауысуыны 25х3=75 Гц-тен кем болмау қажет.



5.2 – сурет. Индикаторды қосу схемасы.

5.2 Жұмыстың орындалу реті

1. Бағдарламаны теру және анықтау, кандай ең аз мәнінде NUM макроанықтауышы жылтылдау нышаны басталады. «123» санының алгоритмін іске асыратын бағдарлама. Жиілі ауысуы кідіріс бағдарламасымен анықталады. A-H сегменті PC0-PC7түйініне сәйкесінше қосылған, ал HG1-HG3 индикатор- PD0-PD2 шығысына сәйкес.

```
// 7 жетісегментті индикатордағы динамическалық индикация
// Қорытынды хабарлама "123" таймер қолдануынсыз
// Кіріс: жоқ
//Шығыс: PORTC - сегментті индикатор
//        PORTD - индикаторы HG1, HG2, HG3
#include <iom8535v.h>
#include <macros.h>
#define NUM 1000
void main(void)
{
unsigned int i;
DDRC =0xFF;      // порт С - кіріс
PORTC=0x00;      // барлық сегменттер жарқырамайды
DDRD =0x07;      // үш кіші D -бит порты шығыста
PORTD=0x00;      // индикаторлар таңдалынбаған
for (;;)                  // үздіксіз цикл
{                    // "1" цифрын шығару HG1 индикаторына
PORTC=0x06;      // 0000 0110 - сегменттер "B" және "C" ,  C портына
PORTD=0x01;      // HG1 индикаторын шығару
for (i=0; i<NUM;  i++);          // бағдарламалық кідіріс 5 мс
// "2" цифрын шығару HG2 индикаторына
PORTC=0x5B;      // 0101 1011 - сегменттер "a","b","d","e","g"
PORTD=0x02;      // HG2 индикаторын шығару
for (i=0; i<NUM;  i++);          // бағдарламалық кідіріс 5 мс
// "3" цифрын шығару HG3 индикаторына
PORTC=0x4F;      // 0100 1111 - сегменттер "a","Ь","c","d","g"
```

```
PORTD=0x04;      // HG3индикаторының шығару
for (i=0; i<NUM;  i++);      // бағдарламалық кідіріс 5 мс
}
}
```

2. Бағдарламаны теру және анықтау, кандай ең аз коэффициент мәнінде T0 таймер бөлгіш жылтылдау нышаны басталады.

Берілген жағдайда ғылими ауыспалы мақсаттылық индификациясы символ коды жене индикатордың кезекті номер беруі, «сегментті» порт түрінде берілген. Төменде динамикалық индикация «123» сандары, T0 таймерінің тоқтатылуымен берілген.

```
// 7 жетісегментті индикатордағы динамическалық индикация
//  T0 таймерін қолданумен
// Қорытынды хабарлама "123
// Кіріс: жоқ
 //Шығыс: PORTC - сегменті
//        PD0...PD2 - HG1...HG3
#include <iom8535v.h>
#include <macros.h>
unsigned char cl=0x06,      // код цифры "1"
c2=0x5B,                    // код цифры "2"
c3=0x4F;                    // код цифры "3"
unsigned char n;          // индикатор номері
// ------------ T0 таймерінің үзіліс өңдеуі -----------------
#pragma interrupt_handler indie:10
void indie(void)
{
PORTD&=0xF8 ;   // барлық индикаторларды өшіремізHG1-HG3
switch (n)
{
case 0:
PORTC=cl ;       // код "1"
PORTD=0x01 ;     // индикатор HG1
break;
case 1:
PORTC=c2;       // код "2"
PORTD=0x02;     // индикатор HG2
break;
case 2 :
PORTC=c3;       // код "3"
PORTD=0x04;     // индикатор HG3
}
n++ ;                    // индикатор номерінің ауысуы
n%=3 ;
}
```

```
void main(void)
{
DDRC =0xFF;              // порт С - кіріс
PORTC=0x00;      // барлық сегменттер жарқырамайды
DDRD =0x07 ;              // үш кіші D -бит порты шығыста
PORTD=0x00;      // индикаторлар таңдалынбаған
TIMSK=0x01;         // Т0 таймерінің толып кетуіне үзіліс рұқсаты
SEI() ;      // үзіліске ортақ рұқсат – 7 битін орнату SREG регистр
TCCR0=0x04 ;       // =0000 0100, бөлгіш коэффициенті256,
// жиілік үзілісі8000000/256/256=122 Гц,
// жиілік санын жаңарту (3 цифры): 122/3=40 Гц
for (;;) ;
}
```

3 таймер 0 бағдарламасын жарық түсірі үшін өзгерту:

- туылған жылың;
- туылған күнің» жене ай (мысалы, 11.09).

PA0 кіріс сигналы индикациясы өзгереді :туылган жылы немесе туылған күні.

5.3 Есептеме мазмұны

Жұмыс мақсаты.
Индикатор сызбасы және оның қосылуы.
Листинг бағдарламасы.
Қорытынды.

5.4 Бақылау сұрақтары

1) Жетісигментті индикотордың қандай сызбалары бар?
2) Жетісегментті индикатор қалай жұмыс істейді?
3) Индикоторларды басқаруда қандай порт іске қосылған?
4) Индикатор бір сивволдың екінші символға өтуін қалай жүзеге асырады?
5) Как осуществить мерцание индикаторов?
6)Динамикалық индикация деп нені айтады?

6 Зертханалық жұмыс №6. Аналогтік-цифрлық түрлендіргіш

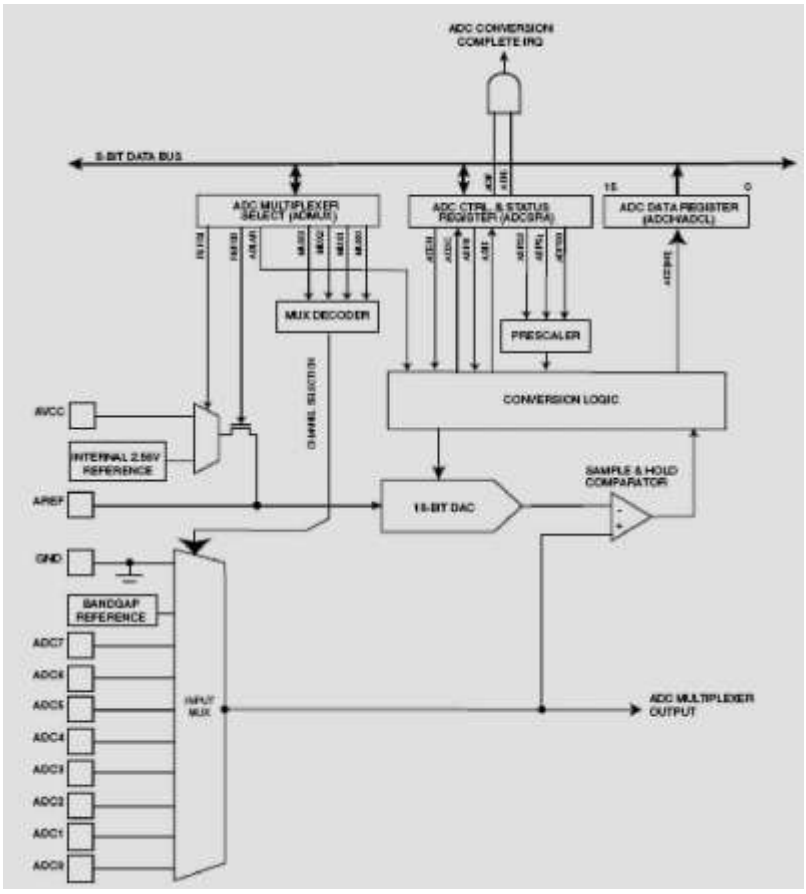
Жұмыстың мақсаты: ATmega8535 микроконтроллерінің АЦП сызбасын және аналог сигналының түрлену принципін, сондай- ақ АЦП бағдарламалауын оқып үйрену.

6.1 Қысқаша мәлімет

ATMega8535 микроконтроллері 8 аналоготі сигналға дейін енгізуге және өңдеуге болады. Ол үшін 10-разрядьтты АЦП тізбектей құрамына кіреді. АЦП негізгі параметрлері мынадай:

- абсолют қателік: ±2;
- интегралды сызықты емес: ±0.5;
-ауысу уақыты: 65.. .260 мкс.

АЦП құрылымдылық схемасы 6.1- суретте келтірілген.



6.1 сурет - АЦП микроконтроллерінің структуралық сызбасы

АЦПқұрамында:

- дешифратор каналымен 8 каналды аналогты мультиплексор;
- таңдалынданған сызбада кернеу көзі 2,56В ;
- АЦП тізбектелген жанасушы;
-сигнал ауысуы уақытының өзгеруі.

А портында альтернативті функциялық контроллер қорданылады,ADC0...ADC7 аналогті сигналды енгізуге, тұжырым сәкесінше

PA0...PA7.Назарға алған жөн, порттағы биттер, АЦП кірісі ретінде бағдаламаланып қолданылады, бірақ мұнда катайтатын резисторлар өшірілуі керек, ендеше кернеудің бұрмалануына алып келеді.

6.1.1 АЦП регистрлары.

АЦП жұмысын басқару 4 регистрдің көмегімен жүзеге асырылады:

- регистр каналының таңдауы және ADMUX кернеу көзі;
- регистр басқаруы және ADCSRA статусы;
- екі регистрдің дерегіADCH, ADCL (АЦП 10-разрядты шығыс коды).

6.1.1.1 Регистр ADMUX каналының таңдауы.

АЦП кіріс канал номері және ADMUX кернеу көзі берілген. Екі үлкен разяд осы регистрдың ИОН кернеу көзін анықтайды.

Тіреуіш кернеуді кернеу анықтайды, кіріс коды максималға тең (10-разряд үшін АЦП - ол 1023). 4 кіші разряд регистры АЦП каналының номерін таңдауға көмектеседі. Аналог сигналының өзгеру нәтежиесі 8-ми екілік разряд регистрында жалғастырылады. Басқару нәтежиесінің өзгеруіне (10-шы биттік код) ADLAR разряды ADMUX регистры атқарады. Егер бұл разрядта «1» қолданылса, нәтежиесі сол жақ шекара бойымен 10 разряд сөзіне ауысады, егер «0» - оң жақ шекарасымен лақтырылса, ADCH:ADCL регистрларында орналастырылады.

6.1.1.2 Басқару регистры және ADCSRA статусы.

ADCSRA регистры АЦП жұмысын қосуға және өшіруге мүмкіндік береді, жұмыс режимін таңдап (жеке немесе үздіксіз), АЦП жұмысының ауысуының аяқталуына рұқсат етіп, сондай- ақ аналог сигналы өзгеру ұзақтыған цифрлы код ретінде беру.

6.1 кесте - Битті ADCSRA регистры

Разряд	Аталуы	Сипаттамасы
7	ADEN	АЦП мүмкіндігі (1 - қосулы, 0 - өшірулі)
6	ADSC	Қайта өңдеуді бастау (1 – қайта өңдеуді бастау)
5	ADFR	АЦП жұмыс режимін таңдау
4	ADIF	АЦП жалаушасының қалпы
3	ADIE	АЦП тоқтату мүмкіндіктері
2...0	ADPS2:ADPS0	Қайта өңдеу жиілігін таңдау

АЦП жұмысының рұсатын ADCSRA регистры ADEN биті арқылы «1» логикалық жағдайына орнатуға болады.

АЦП-ны қосуға ADSC «1» бит ADCSRA регистрын орнату керек.

АЦП жұмысының екі режимі бар:

-жеке ауысу режимі, әр қосылу қолданушының өзгертуімен болса;

-үздіксіз ауысу режимі, косылу белгілі бір уақыт интервалымен үздіксіз орындалса.

АЦП режимінің жұмысы ADFR разрядының жағдыйымен анықталады. Егер ол «1»- де қосылған болса, АЦП үздіксіз режимде жұмыс жасайды. Бұл режимде қосылу автоматты түрде әрбір ауысу сайын орындалады. Егер разряд ADFR «0»-де жиналса, АЦП жұмыс режимі жеке ауысумен және әрбір қосылу қолданушының командасы бойынша жүзеге асады.

АЦП циклының ішкі жұмысы тактылы генератор микроконтроллер бөлгіші арқылы атқарылады. АЦП жұмыс жиілігінің бөлгіш коэффициенті бөлінді және разряд ADPS2..ADPS0 жағдайымен ADCSRA регистрі арқылы анықталады.

АЦП үзілісті жұмысымен рұқсат үзілуі «1» орнатумен жүзезе асырылады.. ADIE разряды ADCSRA регистры I (7 бит) жалаушасында SREG регистрінді орнатылады. Басқа да үзілісті жалауша сияқты, ADIF флагы құралын қосу бағдарламасы кезінде АЦП немесе «1» логикалық бағдарманы қосу арқылы орындалады.

АЦП циклының жұмысы аяқталғанда, нәтежиесі екі 8-разрядымен ADCH және ADCL регистрына жазылады. АЦП өзгеру нәтежиесін оқу алдында ADCL кіші регистрді оқу керек, одан кейін ADCH- улкен регистрін.

АЦП жұмысын орындау үшін келесу көрсетілген әрекетті ретімен орындау керек:

- коэффициентті таңдап бөліндіні бөл;
- жұмыс режимін таңда;
- АЦП жұмысына рұсат бер;
-канал номерін таңда;
- АЦП- ны іске қос («1» битке ADSC орнат).

Өзгертуді орындау алдында («1» флагын ADIF регистры ADCSRA-ын орнатқан жағдайда) АЦП мәліметтер регистрында сақталады.

6.2 Жұмыстың орындалу реті

1. Аналог сигналы ADC3 (3-й канал АЦП - вывод PA3) каналын кабылдауын, 10-разряд басында ауысуын, кейін 8-разряд және D портымен түйінделуін бағдарлама ретінде жазу керек.

```
#include <iom8535v.h>
#include <macros.h>
void port_init(void)
{
DDRA  = 0x00;                    // A портын енгізу
PORTD = 0x00;
DDRD  = 0xFF;            // D портын түйіндеу
}
void adc_init(void)   // АЦП инициализациясы
{
ADMUX = 0x03;     //АЦП (PA3) каналын таңдаймыз,
```

```
                        // AREF- кернеу көзі, оң нәтежиені теңастіру
ACSR   = 0x80;            // Компаратор қолайсыз
ADCSRA = 0xCE;          // 1100 1110
                          // 1-ші бит – қайта бөлгіш = 64
                         // 3-ші бит -  АЦП үзілісіне рұқсат
                         // 5-ші жұмыс режимі-жеке өзгеру,
                          // 6-ші – АЦП іске қосу
}                        // 7-ші – АЦП жұмыс рұқсаты
// АЦП өзгеруінің өңдеуіне бағдарламалау
// тоқтату векторы=15
#pragma interrupt_handler adc_isr:15
void adc_isr(void)
{
int v;
CLI();
v=ADCL;                 //кіші бит өзгеру нәтежиесін есептейміз,
v|=(int)ADCH << 8;     // сосын оны үлкенін және оларды  қосамыз
v>>=2;                  // оның разрядқа жылжытамыз
                          // 8-разряд нәтежиесін аламыз
PORTD=v;                // D портына түйіндейміз
ADCSRA|=0x40;          // қайта өзгеруге қосамыз
SEI();
}
void init_devices(void)         // периферийлы құралдарының инициализациясы
{
CLI();
port_init();                     // порт  инициализациясы
adc_init();                     // АЦП инициализациясы
SEI(); }
void main(void)
{
init_devices();
while (1);
}
```

6.3 Есептеме мазмұны

Жұмыс мақсаты.
АЦП сызбасы.
Листинг бағдарламасы.
Қорытынды.

6.4 Бақылау сұрақтары

1) Аналогты сигналды сандауы қалай орындалады?

2) АЦП-ны қандай регистрлар басқарады?
3) Тіреуіш кернеудің басты қызметі?
4) Для чего предусмотрено выравнивание результата?
5) АЦП-ны қалай қосуға және жіберуді сандауды орындауға болады?
6) Бөлгіш қандай функция атқарады?
7) АЦП-ны принциалды сұлбасы?

Әдебиеттер тізімі

1 Баранов В.Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы. - М.: Додэка – XXI, 2004. – 288 с.

2 Ревич Ю. В. Практическое программирование микроконтроллеров Atmel AVR н,а языке ассемблера.— 2-е изд., испр. — СПб.: БХВ-Петербург, 2011. — 352 с.

3 Евстифеев А.В. Микроконтроллеры семейства Classic фирмы ATMEL. – М.: Додэка - XXI, 2006. – 288 с.

4 Мортон Дж. Микроконтроллеры AVR. Вводный курс. – М.: Додэка-XXI, 2006. – 356 с.

5 Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. – СПб.: БХВ-Петербург, 2011. – 352 с.

6 Трамперт В. AVR-RISC микроконтроллеры. – К.: МК-Пресс, 2006. – 464 с.

7 Белов А. В. Разработка устройств на микроконтроллерах AVR. Книга + видеокурс. — СПб.: Наука и Техника, 2013. — 528 с.

Мазмұны

2014 ж. қосымша жоспар, реті 31

Амантаев Канат Омирзакович

МИКРОКОНТРОЛЛЕРЛЕР

5В071600 – Прибор жасау мамандығының студенттері үшін
зертханалық жұмыстарды орындау бойынша әдістемелік нұсқаулықтар

Редакторы Б.С. Қасымжанова
Стандарттау бойынша маман Н.Қ. Молдабекова

Басуға _____ қол қойылды
Таралымы 50 дана.
Көлемі 2,3 есептік баспа табақ

Пішіні 60х84 1/16
Баспаханалық қағаз № 1
Тапсырыс_____Бағасы 1150 тг

«Алматы энергетика және байланыс университеті»
коммерциялық емес акционерлік қоғамының
көшірмелі- көбейткіш бюросы
050013, Алматы, Байтурсынова көшесі, 126 үй