ПРАКТИКУМ по ЕУСКУ

ЦЕЛ:

Задачата на тези занятия е студентите да усъвършенстват на практика уменията си да програмират сравнително прости микроконтролери в елементарни схеми. Избран е PIC16F716, представител на контролерите PIC от среден клас, поради малкият брой инструкции (35) и относително ниската цена. Основното ядро и инструкциите за този клас контролери на Microchip са еднакви, т.е същите задачи могат да се изпълняват и за други контролери на фирмата – PIC16C84, PIC16F84A и т.н.

ЗАДАЧИ:

1. Разучаване на принципната схема на макета (KIT16Fxxx) и съставяне на алгоритъм за управление на четири-разрядна динамична индикация с бутони.
2. Запознаване със структурата на микроконтролера PIC16Fxxx, задаване на режима му на работа с конфигурационната дума, с Option регистъра и т.н.
3. Програмиране на портовете A и B като входове и изходи, смяна на изходното ниво на изходите управляващи сегментите.
4. Задаване на временни интервали с програмни закъснения (цикли), с помощта на таймера и с прекъсване.
5. Написване и проверка на програма за управление на динамичната индикация без използване на прекъсване.
6. Като предната точка, но с подпрограма за прекъсване.
7. Написване и проверка на програма за "четене" на бутоните.
8. Разучаване на учебната програма KITxxx.asm, написване и тестване на собствена програма за работа на макета като хронометър, измерител на рефлекси и др.
9. Представяне и анализ на резултатите от практикума.

Всички задачи изискват студентите предварително да са се подготвили за това което ще тестват по време на занятията!

РЕД НА РАБОТА:

Задачи 1 и 2 са за самостоятелна подготовка и се изпълняват преди занятията в лабораторията. Използват се материали на Microchip - www.microchip.com, както и материалите подготвени за практикума на страницата на ЕУКУ. Основното е да се разберат модулите на контролера и начинът за задаване на режимите на работа – тип осцилатор, коефициенти на делене на таймера, управление на "кучето" и т.н. Трябва да се прегледат и основните команди на контролера.

Разучават се основните команди за управление на асемблиращата програма, дефиниране на константи и променливи, означаване на двоични, десетични и шестнадесетични числа и др. Удобно е да се ползват примерите от страницата на ЕУКУ.

Задача 3 изисква написването на програма за инициализация на портовете. За пример може да се използва някоя от приложените програми. Основното при тази задача е да се "проиграе" последователността - написване на програмата, асемблирането и програмирането на контролера. За да се стигне до програмиране трябва да се отстранят грешките които открива асемблерът. Задачата е изпълнена, ако на индикацията се изпише зададена произволна комбинация от сегменти.

Задача 4 е за задаване на временни интервали необходими за управлението на динамичната индикация. Тя се изпълнява непосредствено след или едновременно със задача 3 – комбинациите от светещи сегменти да се сменят през произволно зададени интервали от време. Може да се реализират и трите възможност за закъснение – програмно закъснение чрез многократно изпълнение на цикъл, програмна проверка за препълване на таймера и реализиране на прекъсване от таймера. Задачата е изпълнена когато се реализира програма предизвикваща мигане на сегментите с произволна честота.

Задача 5 е подпрограма част от крайната цел която е хронометър или часовник. Преди всичко трябва да се напише подробен алгоритъм за начинът на работа на

динамичната индикация. Задават се регистрите от паметта в които се съхраняват цифрите които ще се изобразяват, регистрите в които са комбинациите от сегменти за всяка цифра, както и броячите които задават индицираната цифра в даден момент. Би трябвало, като се пише тази програма "да се има едно наум", че паралелно с нея е най-добре да се проверяват и бутоните. Задачата е изпълнена когато на индикацията може да се показват четири различни цифри. Допълнително изискване е скоростта на "въртене" на отделните разряди да се променя и да се установи границата при която окото не възприема "трепкането" на индикацията. Най-лесно това става като смяната на разрядите става през различни интервали от време като на индикаторите се показва стойността на тези интервали. Продължителността на интервалите трябва да се сменя през не по-малко от 0,5s за да се възприеме и запомни стойността на интервала.

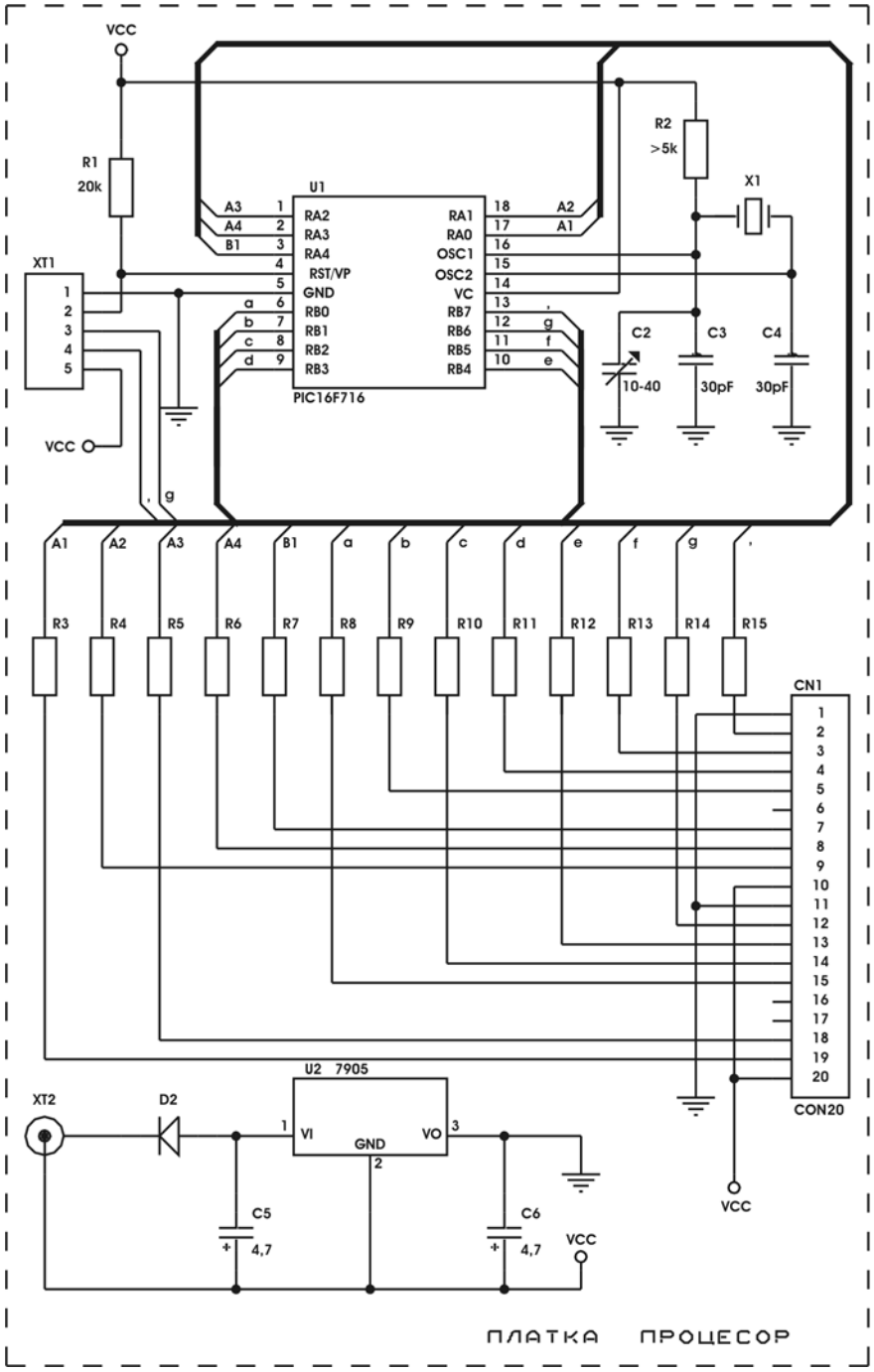
Задача 6 се изпълнява аналогично на задача 5, но се реализира с прекъсване. На таймера се задава стойност съобразена с броя на разрядите и честотата която окото не възприема (няма трепкане). При препълване на таймера (преминаване от FF в 00) съответен флаг се установява в единица и ако е разрешено се генерира прекъсване. Разрешението се дава от специален регистър чрез установяването на съответен флаг. Има общ флаг GBE който отговаря за забрана/разрешение на всички прекъсвания и отделни флагове за различните прекъсвания.

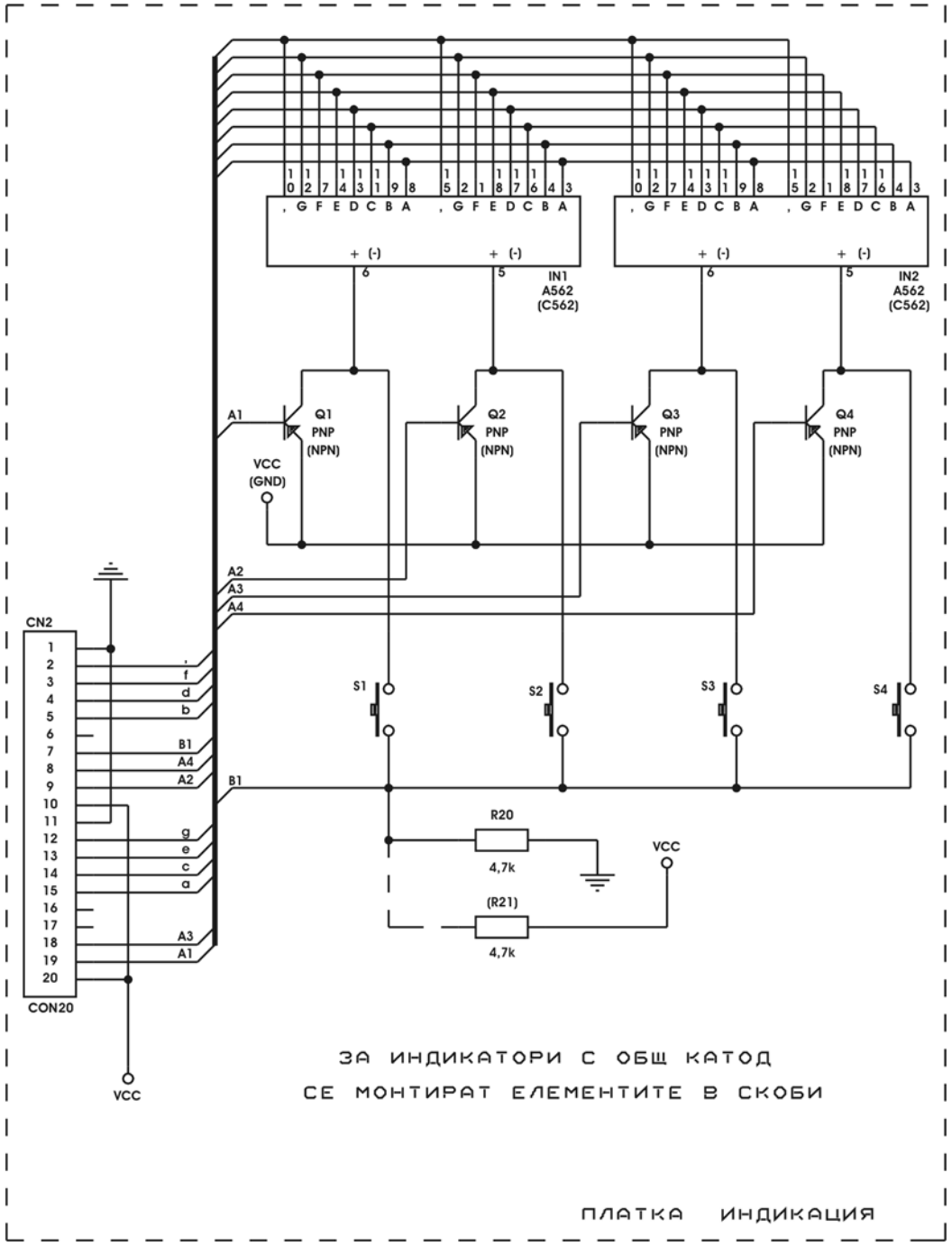
Подпрограмата за прекъсване винаги започва (при този тип процесори) от адрес 04h и завършва с инструкция RETFIE която означава връщане от прекъсване. В тази подпрограма трябва да се обслужва индикацията като се предвиди място и за четене на бутоните (предвидено за задача 7). Програмата трябва да е написана така, че времето през което е включен някой индикатор да не зависи от хода на изпълнение на програмата – в участъка където се сменят индикаторите да няма условни преходи. Друго важно при написването на тези програми е да се запомнят всички регистри които се използват от програмата за прекъсване и в нейния край да се възстановят. Запомнят се W и Status регистъра защото почти всички команди ги променят. Ако не се възстановяват използваните регистри, основната програма няма да продължи при тези условия при които е била прекъсната.

Задача 7 е за четене на бутоните. Тя може да се изпълнява и едновременно със задачи 5 и 6. Това е по-правилният подход, защото с помощта на индикацията може да се установи работоспособността на подпрограмата за бутоните. Друга възможност да се провери как работи програмата е чрез наблюдение с осцилоскоп на някой изход на контролера който да се управлява синхронно с натискането на бутон – отделен изход за всеки от бутоните. Принципната схема предопределя как да се четат бутоните – те са свързани паралелно на индикаторите и при включването на съответен разряд само "неговият" бутон влияе на състоянието на входа RA4. Това може да става по време на обслужването на индикацията или като отделна фаза–след като се "извъртят" четирите индикатора да има фаза отделена на бутоните. И двата подхода са възможни. При по-нискочестотни кварцови резонатори подходът с отделна фаза може да доведе до "трепкане" на индикацията. Задачата е изпълнена когато при натискане на бутон номерът (1-4) му се показва на първия индикатор.

Задача 8 е с най-голяма свобода на действие. Като се разучи примерната програма и при необходимост се използват подпрограми от нея, трябва да се направи собствена програма. Тя може да измерва времето между две натискания на бутони или нещо подобно. Трябва да включва коментираните по-горе програми за индикация и за четене на бутони и главна програма която да изпълнява основната задача – например хронометър. Трябва да се анализира точността на измерването като се отчете грешката от забавяне (поради изпълнение на подпрограмите).

Задача 9 е при приключване на практикума и заверка на семестъра. Това става **при редовно участие в занятията и представяне на работеща версия на програмата**. Работоспособността и се демонстрира от студента който трябва да може да прави и малки промени – например смяна на изписването на цифрите или на реда на въртене на индикацията и т.н.


VCC
R1
20k
U1
PIC16F716
RA2
RA3
RA4
RST/VP
GND
RB0
RB1
RB2
RB3
RA1
RA0
OSC1
OSC2
VC
RB7
RB6
RB5
RB4
A3
A4
B1
A2
A1
a
b
c
d
e
f
g
R2
>5k
X1
C2
10-40
C3
30pF
C4
30pF
XT1
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
CN1
CON20
U2 7905
VI
GND
VO
XT2
D2
C5
4,7
C6
4,7
ПЛАТКА ПРОЦЕСОР


IN1
A562
(C562)
IN2
A562
(C562)
Q1
PNP
(NPN)
Q2
PNP
(NPN)
Q3
PNP
(NPN)
Q4
PNP
(NPN)
VCC
(GND)
S1
S2
S3
S4
R20
4,7k
(R21)
4,7k
CN2
CON20
VCC
ЗА ИНДИКАТОРИ С ОБЩ КАТОД
СЕ МОНТИРАТ ЕЛЕМЕНТИТЕ В СКОБИ
ПЛАТКА ИНДИКАЦИЯ

KIT16Fxxx

Комплект за запознаване с процесорите на фирмата MICROCHIP

Въведение

Избраният процесор PIC16F716, може многократно да се препрограмира и е с относително ниска за качествата си цена. Доколкото програмните модели на процесорите от фамилията PIC16xxx са много близки, преминаването от един към друг не представлява сериозно затруднение.

Описание

На печатната платка е реализирана схема която включва стабилизатор за 5V, микроконтролер PIC16F716 и място за елементите необходими за реализацията на тактовия генератор. При разработването на схемата са предвидени всички възможни тактови генератори за микроконтролера - RC, LP, XT и HS. Естествено те не могат да работят едновременно. В този смисъл на схемата R2 от една страна и X1, C4 от друга не могат да работят едновременно.

През ограничителни резистори са изведени PORTA (RA0-RA4) и PORTB(RB0-RB7). Това позволява платката да се използва за реализацията на различни устройства. В този вариант стойностите на резисторите са подбрани за работа с динамична светодиодна индикация. За захранването е необходимо постоянно напрежение 7,5-12V. На самата платка то се стабилизира на 5V. За захранване може и да се използва и програматорът PICKIT2.

За реализация на четири разрядната динамична индикация е приложена друга платка включваща необходимите елементи за индикатори с общ катод или общ анод и четири бутона.

Варианти

В този вариант е използван RC-генератор за около 2MHz. Може да се използва и някой от другите схеми на генератор. Типът на генератора се задава при програмиране на конфигурационната дума.

Според желанието схемата за динамична индикация може да бъде с индикатори с общ катод или общ анод. На печатната платка са предвидени мостчета които се превключват в зависимост от типа на индикаторите. Сменя се и типа на ключовите транзистори (NPN->PNP). Естествено се налагат и промени в програмата.

Ако е необходимо непрекъснато захранване, например, ако схемата ще се използва като часовник или друго подобно устройство, което да запазва информацията при спиране на тока, се използва акумулатор. В случая акумулаторът трябва да е с напрежение 3,6V. Неговото зарядно напрежение е 4,2-4.4V. Поради това е важно последователно след стабилизатора който е за 5V да се включи диод, а последователно на акумулатора се включва резистор 1-1.5k за да се ограничи токът на заряд и разряд. Естествено капацитетът на акумулатора трябва да се съобрази с тока който се консумира или в софтуера да се предвиди разпознаване на наличието на мрежово захранване и при липсата на захранване да се изключва индикацията.

Инструкция за монтаж

При самостоятелно монтиране на елементите трябва да се работи с поялник със заземен връх, за да се избегне повреда на микроконтролера. Стойностите показани на схемата не са критични, но все пак трябва да са в границите около 30% спрямо дадените. Транзисторите работят при ток 100-120 mA. Необходимият коефициент на усилване по ток да е поне 15-20 (при Uke < 0.5V). Когато се използва RC осцилатор стойностите на елементите се подбират съобразно графиките в описанието на микроконтролера. Стабилна честота се получава при капацитети по-големи от 30pF и резистори по-големи от 5к. **Не може да се очаква, че точна работа на часовник може да се осигури от RC генератор!**

Коментари по програмното осигуряване

В този вариант микроконтролерите **PIC16F716** са програмирани с KIT_716.asm - програма за часовник при осцилаторна честота около 2 MHz. Ако платката ще се използва за прецизен часовник трябва честотата да е по-ниска с оглед на по-малката консумация. За тази цел се ползват резонатори 32768 kHz. Тази ниска честота и динамичната индикация налагат ред ограничения. Основната тактова честота на микроконтролера е четири пъти по-ниска от осцилаторната (8192 Hz). За да не се усеща работата на динамичната индикация честотата с която се сменят индикаторите трябва да е по-висока от 45-50Hz. При четири индикатора това прави около 200 Hz. Така тактовете (команди) с които разполага процесора за да смени разряда, да отчете времето, да провери дали е натиснат бутон и всичко останало са около 40, а това е сериозно ограничение за програмата (програмиста). Естествено при по-високата (2MHz) тактова честота такива проблеми няма.

При 32768 kHz таймерът предизвиква прекъсване на 42 такта, а една секунда се отброява за 195 прекъсвания. За да работи точно, тактовата честота трябва да е 42x195=8190. Тъй като генераторът работи на малко по-висока честота (8192) часовникът ще избързва с около 15-20 s на ден. За да се избегне този недостатък програмно може да се отнема 1s на определени часове. За това се използва програмата **AdjTime**.

При експерименти трябва да се има предвид, че времето се отчита в програмата за прекъсване предизвикано от таймера. Когато се влезе в програмата за прекъсване всички регистри които се използват в нея трябва да се запазят непроменени при излизане. В предложената програма се запомнят само регистрите W и S, които винаги се използват. В програмата за прекъсване се използва и указателя за косвена адресация Fsr. Той може да не се запазва защото в основната програма (извън прекъсването) не се ползва. Ако в някои приложения има косвена адресация тогава и Fsr трябва да се запазва.

Във всички случаи при промени в програмата за прекъсване трябва внимателно да се отчитат необходимите тактовете. При застъпване, когато идва ново прекъсване преди да е завършила програмата за предишното, ефектът се проявява веднага - индикацията започва да мига, но когато броят на тактовете е около границата на застъпването се появяват грешки в отчитането на времето които трудно се забелязват.

Инструкции за програмирането на контролера (с програматор)

Преди да се започнат самостоятелни експерименти се препоръчва работещата програмата да се прочете от микроконтролера и да се запише във файл. С този файл може да се програмира чипът в случай на съмнения в изправността му.

Освен програмата трябва да се програмира и конфигурационната дума. Особено важно е в нея да се укаже типа на генератора и работата на WDT (таймер за “кучето”). Начинът на програмиране на конфигурационната дума се определя от използвания програматор. При PICKIT2 записването става заедно с основната програма, ако в сорса на програмата с __CONFIG е зададено съдържанието на конфигурационната дума.

ВАЖНО ! Броят на презаписите на контролера, макар и голям, е ограничен. Това означава, че експериментите трябва да се обмислят внимателно за да не се "хаби" от ресурса за дреболии.

Програматорът PICKIT-2 се управлява или с MPLAB или с програма PICKIT2. И двете са на Microchip и са свободни за ползване. При някои версии на MPLAB има проблем с програмирането. Тогава се налага писането и асемблиране да се правят с MPLAB, а програмирането с PICKIT2.

Когато се настройва програмното осигуряване, собственото захранване на платката може да е изключено, а захранването да се взема от програматора.