Федеральное государственное автономное образовательное учреждение
высшего образования
«ТЮМЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

РАЗРАБОТКА СИСТЕМ РЕАЛЬНОГО ВРЕМЕНИ

Методические рекомендации по выполнению лабораторных работ
для обучающихся по направлению подготовки
15.03.06 Мехатроника и робототехника
Профиль: автоматизированные системы управления технологическим процессом
форма обучения очная

Введение

Целью изучения дисциплины является формирование единого комплекса понятий, определений и положений о сущности и закономерностях функционирования систем реального времени. Задачи изучения дисциплины: подготовка студентов для научной и практической деятельности в разработки и сопровождения систем реального времени.

Лабораторная работа № 1.

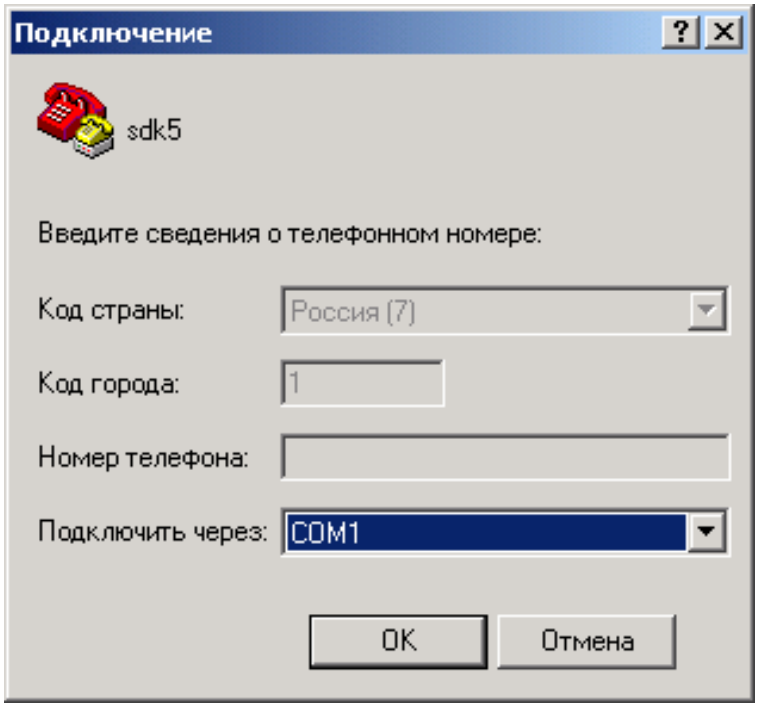
Стенд SDK-5.0 может быть запрограммирован конечным пользователем. Количество циклов программирования ограничено числом циклов записи используемого в стенде SDK-5.0 микроконтроллера — PIC18F458. Число циклов записи указано в документации на микроконтроллер. Для того чтобы конечный пользователь мог запрограммировать микропроцессор стенда SDK-5.0, он поставляется вместе со встроенным загрузчиком, позволяющим загружать пользовательские программы без дополнительного оборудования.

Цель данной практической работы – получить практические навыки записи программы в стенд SDK-5.0 и обзор ограничений, налагаемых на пользовательскую программу, связанных с корректной работой на стенде.

Программирование стенда SDK-5.0

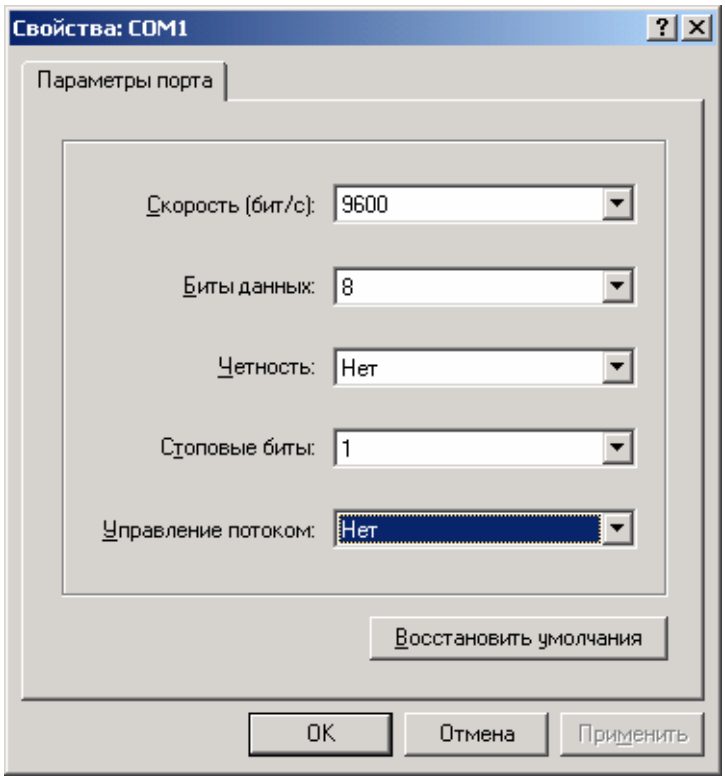
Программу не надо передавать при каждом старте стенда, она сохраняется в его энергонезависимой памяти. Для того чтобы загрузить в стенд пользовательскую программу, нужно воспользоваться любой терминальной программой, позволяющей передавать текстовые файлы. В данной практической работе будет описана работа с программой HyperTerminal, входящей в программное обеспечение ОС MS Windows. Для других терминальных программ надо провести аналогичные настройки. Программа HyperTerminal находится для английской версии ОС в меню Programs→Accessories→Communication и для русской версии ОС в меню Программы→Стандартные→Связь.

Сначала в HyperTerminal надо создать соединение. Назовите его sdk5 и укажите, какой последовательный канал будет использоваться для соединения со стендом.



Если до этого вы ни разу не запускали программу HyperTerminal, то перед этим придется ввести код страны, код города и телефон для соединения. Так как эта информация не используется для работы со стендом, то можете ввести любые сведения.

После указания последовательного канала надо указать его параметры, а именно скорость работы и т.п. Сделайте параметры такими, какие показаны на рисунке.



После выполнения этих действий соединение считается созданным. Его надо сохранить в меню программы HyperTerminal File→Save или Файл→Сохранить. После этого в меню, в котором находится программа HyperTerminal, появится папка HyperTerminal, в которой будет созданное соединение sdk5. При его запуске будет запускаться HyperTerminal с параметрами, введенными при создании соединения.

Когда соединение создано и HyperTerminal запущен, можно подключить контроллер SDK-5.0 к персональному компьютеру шлейфом RS232. После этого контроллеру можно включить питание. При включенном питании нельзя соединять или отсоединять шлейф RS232.

После включения контроллер переходит в состояние ожидание ввода пользователя. Если в течение нескольких секунд пользователь не прервет ожидания, то контроллер запустит на исполнение пользовательскую программу. Если программы в памяти контроллера нет, то поведение стенда неопределенно. Во время ожидания контроллер передает таймер обратного отсчета в последовательный канал. Для прерывания ожидания пользователю надо передать какие-нибудь данные контроллеру. Для этого во время нахождения в терминале достаточно нажать любую клавишу на клавиатуре. Ее код будет отправлен по последовательному каналу и контроллер SDK-5.0 прервет свое ожидание.

Признаком окончания ожидания является передача в терминал контроллером символа двоеточия. После этого контроллер готов принять пользовательскую программу. Для ее передачи в HyperTerminal следует воспользоваться функцией “Передача→Отправить текстовый файл…”. Укажите в открывшемся окне hex-файл «sample.hex». После окончания приема контроллер выведет в терминал символ закрывающей скобки и запустит пользовательскую программу на исполнение. Если за время передачи файла не было никаких ошибок, то в терминале появится строка из двух символов «:)». Если

загружаемая программа содержала в себе данные EEPROM, то после программирования она не запускается. Для запуска программы следует перегрузить стенд. При следующем старте стенд запустит на исполнение последнюю переданную ему программу, если не будет прерван для программирования другой программы.

Программа-терминал необходима только для программирования пользовательской программы. Для работы стенда она не является необходимой.

Демонстрационная программа

Тест предназначен для проверки периферийных устройств контроллера SDK-5.0. В тесте проверяется работа светодиода, семисегментного индикатора, кнопочной клавиатуры, DIP-переключателей, звукоизлучателя, последовательного канала, устройств I2C (термодатчика и часов реального времени).

Тест запускается автоматически при запуске (перезапуске) контроллера. Особенности теста:

1. при старте контроллер должен подать сигнал звукоизлучателем;
2. светодиод должен моргать примерно два раза в секунду все время работы;
3. если контроллер принял без ошибок один байт по последовательному каналу, он пищит звукоизлучателем и возвращает этот же байт (echo). Последовательный канал должен быть настроен с такими же параметрами, какие используются при программировании стенда встроенным загрузчиком;
4. в зависимости от состояния DIP-переключателей на семисегментный индикатор могут быть выведены данные от различных устройств:
    a. если переключатель 1 установлен в ON, а остальные в OFF, на семисегментный индикатор выводятся данные о состоянии клавиатуры: 1, если соответствующая кнопка нажата, иначе 0;
    b. если переключатель 2 установлен в ON, а остальные в OFF, производится тестирование последовательного канала: на семисегментный индикатор выводится значение последнего правильно принятого байта. Если еще не принят ни один байт, на семисегментном индикаторе ничего не отображается. При ошибке вместо значения на семисегментном индикаторе отображаются прочерки;
    c. если переключатель 3 установлен в ON, а остальные в OFF, на семисегментном индикаторе отображается десятичное значение температуры, полученное от термодатчика. При ошибке вместо значения отображаются прочерки;
    d. если переключатель 4 установлен в ON, а остальные в OFF, производится тестирование часов реального времени: отображается время в формате "MM.CC". Значение часов обнуляется при каждом старте контроллера. При ошибке вместо значения отображаются прочерки;
    e. при других положениях DIP-переключателей на семисегментный индикатор ничего не выводится.

Лабораторная работа № 2.

Работа в среде разработки MPLAB

Целью работы является знакомство с языком программирования, компиляция программы, запись программы в память стенда.

Программирование стенда SDK-5.0

Данный комплекс практических работ предназначен для ознакомления с архитектурой и системой команд RISC процессоров (на примере микропроцессора PIC18F458). Наиболее полно изучить особенности процессора, можно программируя только на языке низкого уровня. Поэтому лабораторные работы выполняются на языке Ассемблер.

Учебный лабораторный комплекс SDK-5.0 представляет собой микропроцессорный стенд SDK-5.0, подключенный к персональному компьютеру через интерфейс RS232C (COM-порт ПК). Стенд SDK-5.0 построен на базе микропроцессора PIC18F458 и имеет в своем составе разнообразные устройства, предназначенные для ввода, обработки и вывода информации в цифровом и аналоговом виде. Стенд SDK-5.0 может работать полностью автономно от ПК.

Краткое описание микроконтроллера:

- программная память 32 Кб (Flash)
- память данных 1536 байт (SRAM)
- EEPROM 256 байт
- 4 таймера/счетчика
- SPI, I2C, USART
- 10-битный аналого-цифровой преобразователь (8 каналов)
- аналоговый компаратор (2)

Периферийные устройства:

- Часы реального времени
- Термодатчик
- Звукоизлучатель
- 4 восьмисегментных индикатора
- 4 кнопки
- DIP-переключателя

Средства разработки программ.

Есть два (как минимум) способа написания и компиляции программ для стенда.

1. Для выполнения лабораторных работ удобно использовать среду разработки MPLAB IDE фирмы Microchip.

   1.1. Создать новый проект (меню Project->Project Wizard).

   1.2. На первом шаге выбрать устройство PIC18F458.

1.3. На втором шаге выбрать язык разработки Microchip MPASM Toolsuite. (если необходимо указать путь к компилятору “…\MPLAB IDE\MCHIP_Tools\mpasmwin.exe”)

1.4. На третьем шаге указать имя проекта и папку для проекта.

1.5. На четвертом шаге можно добавить к проекту существующие файлы.

1.6. Для добавления нового файла необходимо выполнить команду Project-> Add new file to project

Добавить файл к проекту можно в любой момент, для этого нужно щелкнуть правой

клавишей мыши в окне проекта на нужной группе файлов и в выпавшем меню выбрать «Add files».

После того как к проекту будет подключен (или создан заново) файл с текстом

программы ее можно скомпилировать (получить HEX-файл).

Для компиляции программы нужно нажать F10 (Ctrl+F10) или воспользоваться пунктом меню Project->Make (Project->Build All).

2. Другой вариант написания и компиляции программы состоит в том, что текст программы пишется в любом простом текстовом редакторе (например, Блокнот). А компиляция производится "вручную" с помощью программы MPASMWIN.EXE. Это оконное Windows-приложение позволяющее задать параметры компиляции программы.

Для того чтобы записать программу в память стенда можно использовать любую терминальную программу, например HyperTerminal. Настройка этой программы для работы со стендом приведена в руководстве пользователя.

После включения питания стенд начинает исполнять специальную программу-загрузчик, расположенную в начале программной памяти. Эта программа предназначена для записи в память стенда пользовательских программ (через интерфейс RS232C). После старта загрузчик в течение 4 секунд ожидает ввода программы пользователя. Если же за это время попытки загрузить программу не было, управление передается последней загруженной в стенд пользовательской программе.

Простейшая программа.

Простейшая программа для стенда SDK-5.0 имеет следующий вид:

```
;*****************************************************************************
*************
```

```
LIST P=PIC18F458 ;директива определения процессора

#include <P18F458.INC> ;включение файла с определениями имен регистров процессора

ORG 0x0200 ;программа должна располагаться

;в памяти с адреса 0x0200 (после загрузчика)

bcf RCON, IPEN ;отключить приоритеты прерываний

bcf INTCON, GIE ;запретить все прерывания
```

```
movlw 'H' ;передаваемый символ
movwf TXREG ;записать его в регистр TXREG (начать передачу)
btfss PIR1, TXIF ;проверить состояние буфера передачи
bra $-2 ;если он не пуст - ждать освобождения
movlw 'i' ;передаваемый символ
movwf TXREG ;записать его в регистр TXREG (начать передачу)
btfss PIR1, TXIF ;проверить состояние буфера передачи
bra $-2 ;если он не пуст - ждать освобождения
movlw '!' ;передаваемый символ
movwf TXREG ;записать его в регистр TXREG (начать передачу)
bra $ ;бесконечный цикл
END ;конец текста программы
;*********************************************************************************
*************
```

Программа передает в последовательный порт строку “Hi!” и затем входит в бесконечный цикл. Так как стенд не находится под управлением операционной системы, то простой выход из программы приведет к неконтролируемой выборке команд микропроцессором из памяти, что может привести к нежелательным последствиям и даже выходу стенда из строя. Поэтому рекомендуется все программы либо завершать бесконечным циклом, либо строить их таким образом, чтобы они работали по бесконечному алгоритму.

Результат работы программы будет виден в окне терминальной программы, которая использовалась для записи программы в память стенда.

Лабораторная работа № 3.

Целью работы является написание простейшей программы на языке ассемблер, реализующей управление светодиодным индикатором на стенде SDK 5.0. Знакомство с механизмом прерываний. Изучение документации от производителя на микроконтроллер PIC18.

Общие сведения.

АЛУ микроконтроллеров PICMICRO начинает выборку команд на выполнение начиная с адреса 0x00 в памяти. Этот адрес называется вектором сброса, т.е. адресом в памяти программ куда переводится счетчик команд при сбросе микроконтроллера. Вектор прерываний (адрес в памяти программ куда переводится счетчик команд при возникновении прерывания) расположен по адресу 0x08. Поскольку лабораторный стенд SDK 5.0 содержит встроенный загрузчик, который располагается в диапазоне адресов от 0x00 до 0x200, то векторы сброса и прерываний смещаются на 200, т.е. вектор сброса будет 0x200, а вектор прерываний 0x208. Это необходимо учитывать при написании своих программ. Несоблюдение этих требований приведет к перезаписи области данных со встроенным загрузчиком и выходом стенда из строя.

Для указания интерпретатору адреса памяти начиная с которого размещается программа используется директива ORG. Например “ORG 0x200” перед началом программы указывает интерпретатору на размещение нижеследующей программы начиная с адреса 0x200 в памяти команд.

Постановка задачи

Задачей данной практической работы является написание программы, реализующей периодическое мигание светодиодом, расположенном на стенде SDK 5.0 (зеленый светодиод в верхнем правом углу).

Входные данные

Для реализации этой задачи необходимо знать следующие аспекты:

1. Форматы регистров конфигурации
2. К какому выводу контроллера подключен светодиод

Источники информации

1. Документация к микроконтроллеру PIC18FXX2_manual.pdf

Для решения задачи будут задействован периферийный модуль таймера TMR0. Конфигурирование контроллера производится путем установки соответствующих битов в регистрах конфигурации. Для выполнения данного задания будут задействованы следующие регистры контроллера:

- INTCON
- TRISB
- PORTB
- T0CON
- TMR0L

Задание для самостоятельной работы

1. Используя документацию на контроллер изучить назначение регистров конфигурации и битов конфигурации.
2. Скомпилировать программу, представленную ниже и загрузить ее в контроллер.
3. Модифицировать программу, изменив время в течение которого горит светодиод (изменить частоту мигания)

```
LEDCON equ 0x00

LEDONOFF equ 0x01

LIST P=PIC18F458

#include <P18F458.INC>

ORG 0x0200

goto START
```

```
ORG 0x208
btfsc INTCON, TMR0IF
call TMRINTSERVICE, 1
retfie
START:
bcf RCON, IPEN
clrf INTCON
bsf INTCON, TMR0IE
bsf INTCON, GIE
clrf LEDCON
clrf PORTB
movlw 0x00
movwf TRISB
call LEDON, 1
bra $
LEDON:
bsf PORTB, RB4
bsf LEDCON, LEDONOFF
bcf T0CON, TMR0ON
clrf TMR0L
movlw 0x82
movwf T0CON
return 1
LEDOFF:
bcf PORTB, RB4
bcf LEDCON, LEDONOFF
bcf T0CON, TMR0ON
clrf TMR0L
movlw 0x85
movwf T0CON
return 1
TMRINTSERVICE:
```

```
bcf INTCON, TMR0IF

btfss LEDCON, LEDONOFF

goto ON

goto OFF

ON:

call LEDON, 1

goto ENDONOFF

OFF:

call LEDOFF, 1

ENDONOFF:

return 1

END
```

Пояснения к коду программы

В начале программы объявляются символические константы адресам в памяти данных:

```
LEDCON equ 0x00

LEDONOFF equ 0x01
```

Регистр LEDCON будет предназначен для конфигурирования светодиода. Данный регистр будет содержать только один флаг (бит конфигурации) LEDONOFF который отражает текущее состояние светодиода (1 – горит, 0 – не горит).

Далее прописана директива, подключающая модуль с символическими константами, специфичными для микроконтроллера PIC18F458. Файл P18F458.INC содержит определение символических констант для адресов регистров. Найдите этот файл в каталоге MP LAB и изучите его.

```
LIST P=PIC18F458

#include <P18F458.INC>
```

Программа состоит из блоков:

- START – метка перехода при старте контроллера
- LEDON – подпрограмма включения светодиода
- LEDOFF – подпрограмма выключения светодиода
- TMRINTSERVICE – подпрограмма обслуживания прерываний

Рассмотрим подпрограмму LEDON.

```
LEDON: ;Символическая метка

bsf PORTB, RB4 ;Установка бита RB4 порта PORTB
```

bsf LEDCON, LEDONOFF ;Установка бита LEDONOFF регистра LEDCON

bcf T0CON, TMR0ON ;Выключение таймера TMR0

clrf TMR0L ;Очистка счетчика таймера

movlw 0x82 ;Запись конфигурации таймера TMR0L в рабочий регистр (аккумулятор)

movwf T0CON ;Пересылка конфигурации таймера TMR0 из рабочего регистра в регистр T0CON

return 1 ;Возврат из подпрограммы

Микроконтроллер PIC18 содержит пять портов ввода/вывода:

- PORTA
- PORTB
- PORTC
- PORTD
- PORTE

Каналы портов (выводы контроллера) обозначаются RB0-RB7 для PORTB и соответственно для других портов. Светодиод подключен к выводу RB4 порта PORTB. В данной подпрограмме зажигается светодиод путем установки бита RB4 регистра PORTB. Затем конфигурируется и включается таймер TMR0. В конце происходит выход из подпрограммы и возврат в основную программу.

Обратите внимание, что после вызова подпрограммы происходит зацикливание программы. Это делается командой:

bra $

Команда «bra» предназначена для безусловного перехода. Метка «$» является ссылкой на адрес текущей команды, т.е. фактически происходит зацикливание программы. Все программы, разрабатываемые для микроконтроллеров должны иметь замкнутый цикл работы, чтобы не происходило произвольного выбора команд из памяти. Это может привести к непредсказуемым результатам или выходу контроллера и периферийных устройств из строя.

При переполнении счетчика таймера TMR0 происходит прерывание и переход к вектору прерываний 0x208:

ORG 0x208

btfsc INTCON, TMR0IF ;Проверка флага TMR0IF регистра INTCON

call TMRINTSERVICE, 1 ;Вызов обработчика прерываний

retfie ;Возврат к основной программе

При возникновении прерывания устанавливается соответствующий флаг. Последовательно проверяя эти флаги, выясняется какое именно прерывание произошло, и принимается решение о вызове программы обслуживающей прерывание.

Отчет по практической работе:

Вверху первой страницы указать фамилии исполнителей и номер группы.

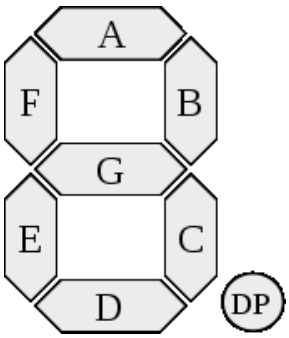
Отчет должен содержать ответы на следующие вопросы:

1. Назначение регистров, используемых в программе и назначение битов регистров;
2. Назначение команд, используемых для написания программы;
3. Каким образом была изменена частота мигания светодиода, указать конфигурация каких регистров была изменена для этого.

Лабораторная работа № 4.

Семисегментный индикатор.

Целью работы является написание программы на языке ассемблер, реализующей управление семисегментным индикатором на стенде SDK 5.0. Отображение буквенно-цифровой информации. Принцип построения изображения на нескольких семисегментых индикаторах.



Общие сведения.

Семисегментный индикатор, как говорит его название, состоит из семи элементов индикации (сегментов), включающихся и выключающихся по отдельности. Включая их в разных комбинациях, из них можно составить упрощённые изображения арабских цифр. Изредка на семисегментном индикаторе отображают буквы.

Сегменты обозначаются буквами от A до G; восьмой сегмент — десятичная точка, имеющая название DP (digital point), предназначенная для отображения дробных чисел.

В обычном светодиодном индикаторе девять выводов: один идёт к катодам (минусу) всех сегментов, и остальные восемь — к аноду (плюсу) каждого из сегментов. Эта схема называется «схема с общим катодом», существуют также схемы с общим анодом.

Многоразрядные индикаторы часто устроены по матричному принципу. Выводы всех одноимённых сегментов всех разрядов соединены вместе. Чтобы выводить информацию на такой индикатор, управляющая микросхема должна циклически подавать ток на общие выводы всех разрядов, в то время как на выводы сегментов ток подаётся в зависимости от того, зажжён ли данный сегмент в данном разряде. Таким образом, чтобы получить десятиразрядный экран микрокалькулятора, нужны всего восемнадцать выводов (8 анодов и 10 катодов) — а не 81.

Постановка задачи

Задачей данной практической работы является написание программ, реализующих следующие функции:

1. Отображение цифр от 0 до 9 и букв A, B, C, D, E, F на каждом из четырех семисегментных индикаторов. Для демонстрации – показать один из символов на семисегментном индикаторе соответствующем варианту студента.
2. Последовательно отобразить цифры от 0 до 9 на семисегментном индикаторе соответствующем варианту студента. После отображения цифр также последовательно отобразить латинские буквы A, B, C, D, E, F. Дойдя до буквы соответсвующей варианту студента – перевести индикатор в режим мигания. Распределение вариантов:
    a. 1 вариант – индикатор № 1, буква «C»
    b. 2 вариант – индикатор № 2, буква «D»
    c. 3 вариант – индикатор № 3, буква «E»
    d. 4 вариант – индикатор № 4, буква «F»
3. Отображение на четырех семисегментых индикаторах какого-либо слова или числа. Для переключения между индикаторами использовать таймер TMR1. Алгоритм работы программы следующий:
    a. Отобразить символ на первом индикаторе
    b. Выполнить задержку при помощи таймера
    c. Выключить первый индикатор
    d. Отобразить символ на втором индикаторе
    e. Выполнить задержку при помощи таймера
    f. И таким же образом в цикле для всех индикаторов.
4. Реализуйте задержку при помощи таймера TMR0 и сравните результаты

Входные данные

Для реализации этой задачи необходимо знать следующие аспекты:

3. Форматы регистров конфигурации
4. К какому выводу контроллера подключены выводы семисегментых индикаторов:
    a. Катод индикатора 1 подключен к выводу RA5 порта PORTA
    b. Катод индикатора 2 подключен к выводу RE0 порта PORTE
    c. Катод индикатора 3 подключен к выводу RE1 порта PORTE
    d. Катод индикатора 4 подключен к выводу RE2 порта PORTE
    e. Аноды сегментов подключены к выводам RD0 – RD7 порта PORTD

Источники информации

2. Документация к микроконтроллеру PIC18FXX2_manual.pdf

Ниже приведен пример программы которая реализует последовательно отображение цифр от 1 до 3 и переход в режим мигания по достижению заданной цифры. Необходимо модифицировать программу для выполнения заданий 1 и 2.

```
INPUT equ 0x4F

LIST P=PIC18F458

#include <P18F458.INC>

ORG 0x0200

goto START

START:

clrf TRISE

clrf PORTE

clrf TRISD

clrf PORTD

bsf PORTE, RE2

call d0, 1

call wait, 1

call d1, 1

call wait, 1

call d2, 1

call wait, 1

call d3, 1

call wait, 1

goto START

bra $

d0:

movlw 0x3F

movwf PORTD

return 0

d1:

movlw 0x06

movwf PORTD

return 0
```

```
d2:
movlw 0x5B
movwf PORTD
return 0
d3:
movlw 0x4F
movwf PORTD
return 0
wait:
movff LATD, WREG
xorlw INPUT
btfsc STATUS, Z
goto lighting
clrf INTCON
movlw 0x04
movwf T0CON
bsf T0CON, TMR0ON
btfss INTCON, TMR0IF
bra $-1
bcf T0CON, TMR0ON
goto endwaiting
lighting:
clrf  INTCON
clrf  TMR0L
btfss LATE, RE2
goto ON
goto OFF
ON:
movlw 0x84
movwf T0CON
btfss INTCON, TMR0IF
bra $-1
```

```
bsf LATE, RE2
bcf T0CON, TMR0ON
goto lighting
OFF:
movlw 0x84
movwf T0CON
btfss INTCON, TMR0IF
bra $-1
bcf LATE, RE2
bcf T0CON, TMR0ON
goto lighting
endwaiting:
return 1
END
```

Пример реализации задержки при помощи таймера TMR1:

```
wait:
bcf PIR1, TMR1IF
movlw 0xFF
movwf TMR1L
movlw 0xF0
movwf TMR1H
movlw 0xB0
movwf T1CON
bsf T1CON, TMR1ON
cycle:
btfss PIR1, TMR1IF
goto cycle
bcf T1CON, TMR1ON
return 1
```

Прокомментируйте в отчете разницу в использовании таймеров TMR0 и TMR1

Лабораторная работа № 5.

Используя знания, полученные в ходе выполнения предыдущей практической работы сделать бегущую строку на четырех семисегментных индикаторах из четырех букв.

Список литературы

1. Конюх, В. Л. Проектирование автоматизированных систем производства : учебное пособие / В. Л. Конюх. - Москва : КУРС : ИНФРА-М, 2019. - 312 с. - ISBN 978-5-905554-53-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1027253 (дата обращения: 25.05.2020). – Режим доступа: по подписке.
2. Прокопенко, А. В. Синтез систем реального времени с гарантированной доступностью программно-информационных ресурсов [Электронный ресурс] : монография / А. В. Прокопенко, М. А. Русаков, Р. Ю. Царев. - Красноярск: Сиб. федер. ун-т, 2013. - 92 с. - ISBN 978-5-7638-2748-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/492781 (дата обращения: 25.05.2020). – Режим доступа: по подписке.
3. Корнеев, И. К. Технические средства управления : учебник / И. К. Корнеев, Г. Н. Ксандопуло. — Москва : ИНФРА-М, 2019. — 200 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-003620-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/991843 (дата обращения: 25.05.2020). – Режим доступа: по подписке.