

Московский государственный технический университет
имени Н.Э. Баумана

И.А. Васильев, Е.С. Люминарская, К.В. Селиванов

**МОДЕЛИРОВАНИЕ
МИКРОКОНТРОЛЛЕРА PIC 16 F84A**

Учебно-методическое пособие

Москва
(С) 201_ МГТУ им. Н.Э. БАУМАНА

УДК 004.312

Издание доступно в электронном виде по адресу
ebooks.bmstu.press/catalog/72/book1863.html

Факультет «Фундаментальные науки»
Кафедра «Электротехника и промышленная электроника»

*Рекомендовано Научно - методическим советом
МГТУ им. Н.Э. Баумана в качестве учебно- методического пособия*

Васильев И.А.

Моделирование микроконтроллера PIC 16 F84A: учебно-методическое пособие для выполнения лабораторной работы по дисциплине «Электротехника и электроника» / И.А. Васильев, Е.С. Люминарская, К.В. Селиванов. – Москва: Издательство МГТУ имени Н.Э. Баумана, 2019. –15с.

Изложены основные понятия и методы моделирования микроконтроллера в среде моделирующей программы MULTISIM. Рассмотрены примеры ввода и вывода цифровой информации через порты микроконтроллера. Приведены листинги управляющих программ.

Для студентов 4-го курса факультета ФН, изучающих дисциплину «Электротехника и электроника».

Оглавление

Предисловие	4
1. Краткие теоретические сведения.....	5
2. Задание и порядок выполнения работы.....	10
3 Требования к отчету.....	13
Вопросы для самоконтроля.....	13
Литература.....	13
Приложение	14

ПРЕДИСЛОВИЕ

В настоящее время при разработке микропроцессорных систем (МПС) широко используются микроконтроллеры (примерно в 80% случаев). При этом микроконтроллеры применяются как самостоятельно (с дополнительной аппаратурой), так и в составе сложных контроллеров с развитыми средствами ввода/вывода.

В пособии изучение основ работы микропроцессорной системы управления проводится на примере микроконтроллера PIC16F84A фирмы Microchip. Микроконтроллер PIC16F84A имеет развитую систему модулей для связей с объектом управления и высокую производительность. Данный микроконтроллер подходит для широкого спектра задач: от схем высокоскоростного управления автомобильными и электрическими двигателями до экономичных удаленных приемопередатчиков, показывающих приборов и связанных процессоров.

Лабораторная работа «Моделирование микроконтроллера PIC 16 F84A» предусмотрена учебным планом дисциплины «Электротехника и электроника» для 4-го курса факультета «Фундаментальные науки» направлений подготовки 16.03.01 «Техническая физика» и 24.03.01 «Ракетные комплексы и космонавтика» МГТУ им. Н.Э. Баумана.

Цель лабораторной работы – получение навыков применения соответствующего программного обеспечения для изучения принципов работы микроконтроллера, его моделирования и ввода-вывода информации при проведении экспериментальных исследований.

Задачи лабораторной работы – сформировать у студентов такие компетенции как способность самостоятельно работать в современных компьютерных программах, а также способность моделировать современные электротехнические и электронные устройства для решения профессиональных задач на примере моделирования микроконтроллера PIC 16 F84A в программной среде Multisim Sumulink.

Выполнение лабораторной работы способствует формированию у студентов умений и навыков моделирования и программирования микроконтроллера в программной среде Multisim, а также приобретению знаний об основах его функционирования.

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Микроконтроллером называют микропроцессорное устройство, предназначенное для управления внешними исполнительными механизмами. Для осуществления связи с

внешними исполнительными механизмами (устройствами) контроллер имеет несколько портов ввода-вывода данных. Управляющая программа считывает данные о состоянии внешних устройств с порта ввода данных и посылает управляющие сигналы на внешние устройства в порт вывода (управление по прерыванию в данной работе не рассматривается).

Как правило, порт может быть запрограммирован на ввод или вывод данных по каждому из выводов порта с помощью флагов в регистре управления портом. Таким образом, порт представляет собой двунаправленную шину данных с разрядностью равной разрядности соответствующего порта.

Модель порта в однобитном исполнении

Модель порта в однобитном исполнении изображена на рис. 1. На рассматриваемой модели буферные (без инверсии) усилители $U1$ и $U2$ могут быть переведены в “Z” состояние по потенциалу управляющего входа “С” (вход С помечен буквой «с» на усилителях $U1$ и $U2$). При низком уровне потенциала управляющего входа усилитель переходит в состояние высокого импеданса по выходу и не передает сигнал от входа к выходу, а, следовательно, не создает помехи в работе другого усилителя. При высоком уровне потенциала управляющего входа усилитель переходит в режим передачи сигнала от входа к выходу. Для устранения режима одновременного включения усилителей управляющий вход усилителя $U1$ подключают к переключателю выбора режима $J3$ через инвертор $U3$.

При выборе режима ВЫВОД ($J3$ в верхнем положении) усилитель $U2$ будет активен, а усилитель $U1$ – пассивен (в “Z” состоянии). Ключ $J2$ переключают на МИКРОКОНТРОЛЛЕР. В этом случае сигнал передается от МИКРОКОНТРОЛЛЕРА к ВНЕШНЕМУ УСТРОЙСТВУ (индикаторы работают синхронно).

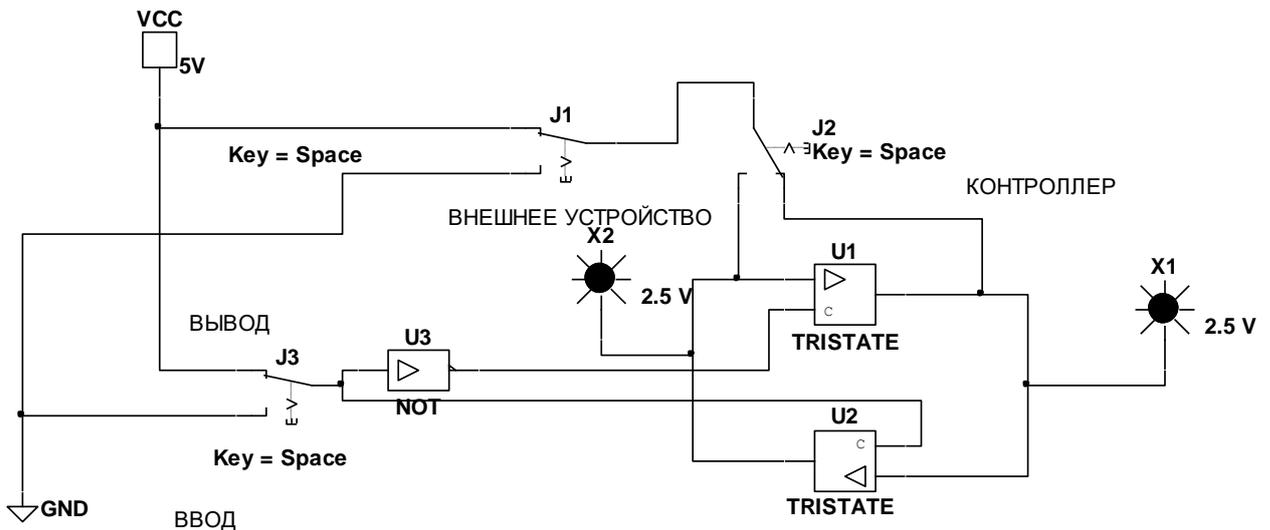


Рис.1. Модель однобитного порта ввода-вывода.

При выборе режима ВВОД (J3 в нижнем положении) усилитель U2 будет пассивен (в “Z” состоянии), а усилитель U1 – активен. Ключ J2 переключают на ВНЕШНЕЕ УСТРОЙСТВО. В этом случае сигнал передается от ВНЕШНЕГО УСТРОЙСТВА к МИКРОКОНТРОЛЛЕРУ (индикаторы работают синхронно). При неправильном выборе источника сигнала индикаторы работать синхронно не будут.

В микроконтроллере переключение режимов ввода и вывода выполняют установкой битов в регистре управления портом.

Микроконтроллер для вывода последовательности чисел

Микроконтроллер осуществляет генерацию последовательности управляющих кодов для смены состояний автомата управления некоторым технологическим процессом.

На рис. 2 изображена схема включения микроконтроллера для моделирования вывода чисел в заданной последовательности в зависимости от тактовых импульсов, генерируемых микроконтроллером.

Микроконтроллер имеет два порта ввода-вывода: порт А (выводы микросхемы 17, 18, 1,2,3) и порт В (выводы микросхемы с 6 по 13 и 15). В управляющей программе эти порты имеют имена (идентификаторы) PORTA и PORTB. Для вывода числа k в порт используют оператор $PORTA = k$; для чтения данных dat – оператор $dat = PORTA$.

Портами называют программно доступные регистры, через которые управляющая программа может считать данные с внешнего устройства или выставить управляющие сигналы на внешнее устройство.

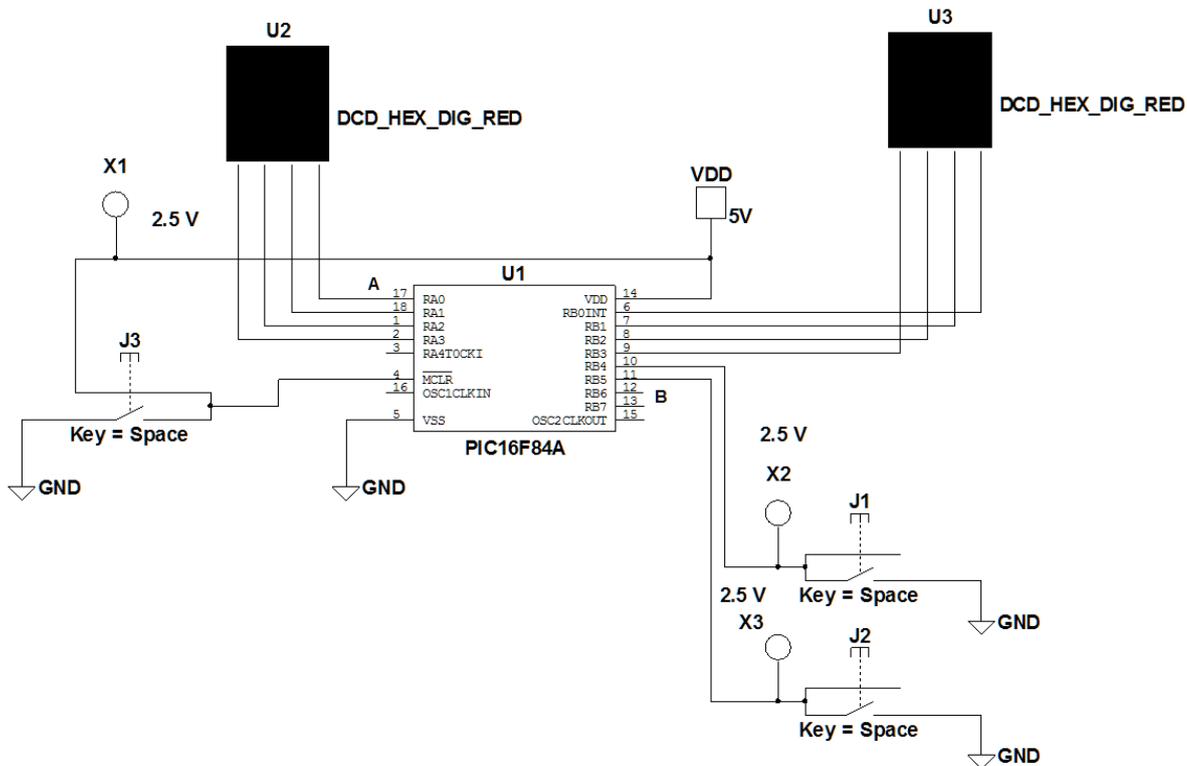


Рис. 2. Схема микроконтроллера для вывода последовательности чисел

К микросхеме контроллера подключено питание (режима питания нет по умолчанию). Кнопка J3 осуществляет общий сброс процессора. Дисплей U2 предназначен для вывода чисел в заданной последовательности.

Дисплей U3 предназначен для вывода номера импульса. Программа управляется двумя внешними устройствами J1 и J2 (см. рис. 2): кнопка J1 – для старта работы автомата, а кнопка J2 – для его остановки. Листинг программы управления автоматом приведен ниже

```

#include<htc.h> //команды препроцессора - подключение макросов и библиотек
//prototype functions
void Pause (int); //прототип функции паузы
void Inc_Port (void); // прототип функции инициализации порта
//end prototype functions
void main()
{
    int n,i,j; //описание целых переменных
    int dat[8] = {1,4,3,7,6,5,0,2}; // задание лабораторной работы
    j=8; //число импульсов задания лабораторной работы
    n = 200; //параметр для функции паузы
    Inc_Port (); //инициализация порта
lb1: while (RB4 == 1); //ожидание флага по кнопке J1
    while (1) //бесконечный цикл вывода последовательности чисел
    {
        for (i=0; i < j; i++) //конечный цикл по числу выводимых чисел
        {
            PORTA = dat[i]; //вывод на дисплей U2 чисел д.з.
            PORTB = i; // вывод на дисплей U3 номера импульса
            Pause (n); //пауза
        }
        while (RB5 == 0) // флаг по кнопке J2 пока кнопка не нажата игнорируется
        {
            goto lb1; //
        }
    }
}

void Pause (int n) //функция паузы
{
    int k; // внутренний параметр паузы
    for( k = 0; k < n ; k++) //цикл по внешнему параметру паузы
    {
        k++; //инкремент
        k--; //декремент
    }
}

void Inc_Port (void) //функция инициализации портов
{
    TRISA = 0x00; //port A output установка флагов на вывод
    TRISB = 0xF0; //portB B7-B4 input B4-B0 output
    RBPU = 1; // кнопка "1" input при разомкнутом ключе
}

```

Для моделирования процесса вывода последовательности студенты должны её инициализировать оператором `int dat[8] = {1,4,3,7,6,5,0,2}`. Задание необходимо выбирать из таблицы (Приложение) в соответствии с порядковым номером студента в списке группы. Значению параметра j нужно присвоить значение количества цифр задания.

Микроконтроллер считывает данные и функционирует в соответствии с внешними командами, посылаемыми оператором с пульта управления. Такой тип управления

называют управлением *по ожиданию значения флага*, а соответствующие биты называют флагами.

Для того, чтобы биты порта могли быть включены на ввод или вывод данных используют регистр управления портом. Имена регистров не являются элементами языка СИ и зависят от применяемого микроконтроллера.

Для данного микроконтроллера при управлении портом В используют регистр с именем TRISB, а портом А – с именем TRISA.

```
void Inc_Port (void) //функция инициализации портов
{
    TRISA = 0x00; //port A output установка флагов на вывод
    TRISB = 0xF0; //portB B7-B4 input B4-B0 output
    RBPU = 1; // кнопка "1" input при разомкнутом ключе
}
```

В микроконтроллере каждый бит порта может быть настроен (сконфигурирован) на ввод или вывод данных по значению соответствующих битов в регистре управления. Если в бите с номером *N* регистра управления установлен "0", то соответствующий бит порта будет открыт на вывод данных, если "1" – на ввод данных (чтение). При чтении данных допускается побитный ввод данных. Идентификаторы RB4 и RB5 в листинге программы означают побитное чтение 4 и 5 битов порта В. Таким образом, идентификатор RB4 (или RB5) равен "0", если кнопка управления J1 (или J2) нажата, и "1", если отпущена.

Оператор `RBPU = 1` выполняет подключение резистора порта (подтягивающего резистора) к источнику питания. Это позволяет иметь "уверенную" единицу на входе бита порта при разомкнутой кнопке.

Первый исполняемый оператор `lb1: while (RB4 == 1)` имеет метку lb1. Этот оператор цикла будет выполняться до смены истинности выражения в скобках. При разомкнутой кнопке J1 с 4 бита порта В считывается "1" и идентификатору RB4 присваивается значение "1", При замыкании кнопки J1 идентификатору RB4 присваивается значение "0". Тогда выражение `RB4==1` становится ложным и программа переходит к исполнению следующего оператора `while (1)`, а микроконтроллер приступает к циклическому выводу

данных и номера импульса. Программа `void Pause (int n)` необходима для устранения быстрого мелькания дисплея.

Оператор цикла `while (RB5 == 0)` предназначен для остановки процесса вывода данных. При отпущенной кнопке значение `RB5` равно "1". Исполняемый оператор игнорируется, и цикл вывода данных продолжается.

После нажатия кнопки `J2` (её нужно удерживать некоторое время, так как в цикле есть функция паузы) программа переходит на метку `lb1` и контроллер ожидает нажатие кнопки `J1`.

2. ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Загрузите программу моделирования порта ввода-вывода "Модель_порта". Положением переключателя `J1` (рис. 1) выберите направление передачи информации и определите передатчик и приемник сигнала управления. Поясните полученный результат. Выключите исполнение программы "Модель_порта".

2. Загрузите программу "MCU_DZ_2". В главном меню выберите пункт `MCU`, далее в всплывающих меню выберите `MCU PIC 16...`, `MCU Code Manager` и нажмите левую кнопку мыши. В появившемся окне найдите закладку "main.c" и щелкните на ней дважды левой кнопкой мыши. Закройте `MCU Code Manager`, нажав на кнопку "OK". Проведите редактирование кода в соответствии с заданием из таблицы в приложении. Введите данные задания в операторе присваивания значения массива. Например, если в задании указан следующий набор цифр (1,4,7,6,3), то число состояний (тактовых импульсов) равно 5 и две строчки ввода данных будут иметь вид:

```
int dat[5] = {1,4,7,6,3}; // задание
j=5; //число импульсов задания.
```

После этого запустите программу и произведите компиляцию нового кода для создания исполняемого кода микропроцессора. Для этого в левом верхнем углу нажмите стрелку в виде зеленого треугольника. Подтвердите запрос на изменение кода, нажав на кнопку "Yes". При отсутствии ошибок и предупреждений в нижнем окне будет выдано

сообщение: 0-Errors, 0-Warnings. Перейдите к закладке MCU_DZ_2, нажав в левом нижнем углу на соответствующую кнопку. Появившаяся в окне схема находится на выполнении в режиме ожидания.

Нажмите на ключ J1 и убедитесь в правильности чередования выводимых цифр и соответствии этих цифр их номерам. Остановите процесс вывода цифр. Для этого нажмите на ключ J2 левой кнопкой мыши и удерживайте её нажатой до остановки процесса.

Вновь запустите программу, нажав левой кнопкой мыши на ключ J1, и остановите процесс с помощью *кратковременного нажатия* левой кнопкой мыши на ключ J3. Убеждаемся в быстром срабатывании кнопки J3.

Вновь запустите программу, нажав левой кнопкой мыши на ключ J1, и попытайтесь с помощью *кратковременного нажатия* левой кнопкой мыши на ключ J2. Убеждаемся в отсутствии управляющего действия.

Выключите исполнение программы MCU_DZ_2, нажав на кнопку "Stop simulation" (кнопка с красным квадратом). Закройте проект для выхода из оболочки Multisim.

3. Создайте новый проект, запустив программу Multisim. В строке инструментов найдите Place MCU (значок с изображением микросхемы с многими выводами). Выберите контроллер PIC16F84A и установите его на листе. В появившемся окне присоединения кода к данной микросхеме введите имя, состоящее из любых нескольких букв. Последовательно нажмите кнопки Next, Next, Finish. Соберите схему, изображенную на рис. 3.

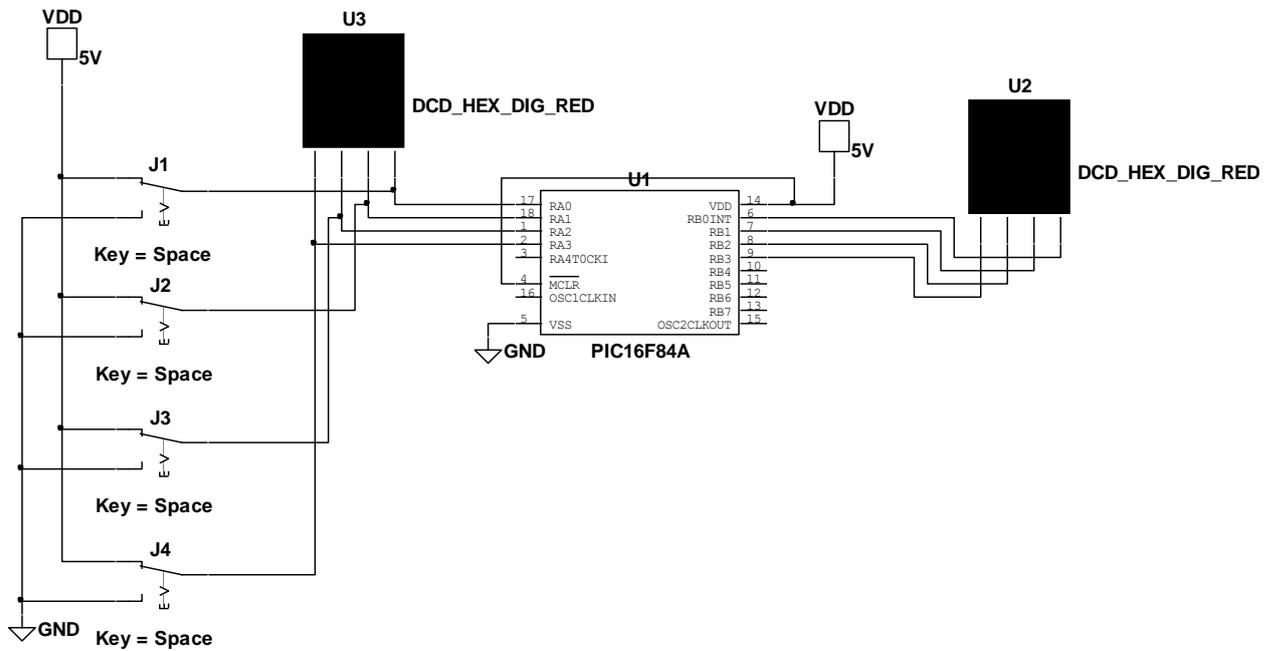


Рис. 3. Новый проект.

В главном меню выберите пункт MCU, далее в всплывающих меню выберите MCU PIC 16..., MCU Code Manager и нажмите левую кнопку мыши. В появившемся окне найдите закладку "main.c" и щелкните на ней дважды левой кнопкой мыши. Закройте MCU Code Manager, нажав на кнопку "OK".

Введите управляющую программу для собранной схемы.

```
#include<htc.h>

void main()
{
    int i;

    TRISA = 0xFF;//port A input

    TRISB = 0x00;//port B output
while(1)
{
    i = PORTA;
    PORTB = i;
}
}
```

Введенная программа считывает биты с цифрового устройства U3 в порт А и выводит их в порт В.

3. ТРЕБОВАНИЯ К ОТЧЕТУ

Отчет должен содержать:

- 1) титульный лист;
- 2) описание цели работы;
- 3) схему микроконтроллера;
- 4) листинг программы;
- 5) выводы по работе.

ПРИЛОЖЕНИЕ

Таблица выбора вариантов заданий.

№ задания	№ импульса	0	1	2	3	4	5	6	7
1. Четный номер по списку группы	Код	4	0	5	6	3	2	7	1
2. Нечетный номер по списку группы	Код	1	3	0	7	2	4	6	5

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Для чего предназначены порты ввода-вывода?
2. Как переключить направление передачи данных в порте ввода-вывода? Для чего служит регистр управления портом?
3. Как осуществляется управление программой по ожиданию флага?
4. Приведите примеры управления программой по ожиданию флага из листинга программы “MCU_DZ_2”. Дайте пояснения.
5. Поясните действие программы ввода и вывода (последний листинг). Выпишите и дайте комментарии к каждой строке листинга программы.

ЛИТЕРАТУРА

1. Джонс М.Х. Электроника – практический курс. Москва Постмаркет, 1999. – 528с.
2. Шпак Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров. Киев МК-Пресс 2006. - 400с.