

Васильев Игорь Александрович

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО МОДЕЛИРОВАНИЮ ПУЛЬТА ВВОДА ДАННЫХ В ПРОГРАММУ МИКРОПРОЦЕССОРА В СРЕДЕ MULTISIM

В статье рассматривается моделирование подключения простейшего кнопочного пульта ввода данных в программу микропроцессора. Моделирование осуществляется в интегрированной среде разработки электронных схем MULTISIM. Программа микропроцессора разрабатывается на языке С с применением встроенного в MULTISIM компилятора. Микропроцессорное устройство не имеет операционной системы. Поэтому студентам важно иметь практические навыки создания и программирования элементов внешнего интерфейса.

Адрес статьи: www.gramota.net/materials/1/2017/7/5.html

Статья опубликована в авторской редакции и отражает точку зрения автора(ов) по рассматриваемому вопросу.

Источник

Альманах современной науки и образования

Тамбов: Грамота, 2017. № 7 (120). С. 21-23. ISSN 1993-5552.

Адрес журнала: www.gramota.net/editions/1.html

Содержание данного номера журнала: www.gramota.net/materials/1/2017/7/

© Издательство "Грамота"

Информация о возможности публикации статей в журнале размещена на Интернет сайте издательства: www.gramota.net

Вопросы, связанные с публикациями научных материалов, редакция просит направлять на адрес: almanac@gramota.net

УДК 621.382

Педагогические науки

В статье рассматривается моделирование подключения простейшего кнопочного пульта ввода данных в программу микропроцессора. Моделирование осуществляется в интегрированной среде разработки электронных схем *MULTISIM*. Программа микропроцессора разрабатывается на языке *C* с применением встроенного в *MULTISIM* компилятора. Микропроцессорное устройство не имеет операционной системы. Поэтому студентам важно иметь практические навыки создания и программирования элементов внешнего интерфейса.

Ключевые слова и фразы: порт; ввод/вывод данных; микропроцессор; пульт ввода данных; интерфейс; практические навыки; навыки моделирования.

Васильев Игорь Александрович, к.т.н.

Московский государственный технический университет имени Н. Э. Баумана
ivasiliev@rslab.ru

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ ПО МОДЕЛИРОВАНИЮ ПУЛЬТА ВВОДА ДАННЫХ В ПРОГРАММУ МИКРОПРОЦЕССОРА В СРЕДЕ *MULTISIM*

Представленные в статье материалы служат основой для проведения практического занятия по теме «Управление вычислительным процессом в микропроцессорных устройствах».

Автор проводит занятие в компьютерном зале Кафедры электротехники и промышленной электроники МГТУ им. Н. Э. Баумана с использованием лицензионной программы *MULTISIM 10*.

Рассмотренный в статье пример реализации пульта ввода данных в программу микропроцессора используется автором на лекциях и практических занятиях по дисциплине «Цифровая электроника» при подготовке магистров по специальности «Техническая физика».

Перед проведением занятия студенты в процессе лекционной подготовки должны приобрести навыки моделирования в среде *MULTISIM* [4] и программирования на алгоритмическом языке *C* [5]. Студенты должны иметь представление о внутренней и внешней архитектуре микропроцессора *PIC16F84* [3].

Учащиеся и преподаватели могут установить программу с ограниченными правами с сайта разработчика *MULTISIM* (ni.com/multisim/try). При этом нужно пройти процедуру регистрации.

Целью занятия является приобретение студентами практических навыков по разработке кнопочного пульта ввода данных микропроцессорным устройством. При подготовке к занятию студенты повторяют материалы по подключению устройств с параллельным интерфейсом [1] и моделированию пульта управления [2]. Предполагается, что по результатам подготовки к занятию студенты овладеют практическими навыками подключения дисплея и кнопок управления к микропроцессору в среде моделирования *MULTISIM*.

Схема модели микропроцессорного устройства с пультом ввода данных представлена на Рис. 1.

При разработке модели необходимо выполнить процедуру создания нового проекта на основе микросхемы *PIC16F84*. Из меню *MCU* размещаем микросхему *PIC16F84* на наборном поле. После установки микросхемы появится диалоговое окно с тремя шагами.

Для первого шага (Step1) в окне *Workspace path* прописываем путь проекта, например:

C:\TMP\PULT_MCU

Далее в окне *Workspace name* очищаем окно и прописываем имя, например: номер группы и вариант.

Далее кнопка *Next*.

Для второго шага кнопка *Next*.

Для третьего шага кнопка *Next*.

После третьего шага сформируется *main* программа для языка программирования *C*.

Далее переходим на вкладку *Design1* внизу справа поля ввода *MULTISIM*.

Собираем схему по Рис. 1 и запоминаем файл схемы в папке C:\TMP\PULT_MCU с именем «номер группы и вариант».

Далее в окне *Design Toolbox* по ниспадающему пути проекта доходим до синей вкладки *main.c*, дважды щелкаем левой кнопкой и открываем редактор и средства отладки и компиляции. Выше *void main()* помещаем *#include<htc.h>*.

На Рис. 1 элементы *U4*, *U5*, *U6* и *U7* – интерактивные цифровые константы. Они предназначены для набора вводимых в программу микропроцессора числовых данных.

Кнопка *S2* предназначена для ввода данных после их набора и играет роль клавиши *ENTER*.

Кнопка *S1* инициализирует микропроцессор *U1*.

По показаниям дисплея *U8* контролируются вводимые данные.

Дисплеи *U2* и *U3* отображают результат ввода данных и результат вычисления суммы вводимых чисел.

Программа обслуживания микропроцессорного устройства приведена в листинге с комментариями. Эта программа предназначена для ввода двух восьмиразрядных чисел и вычисления их суммы.

```
#include<htc.h> //подключение библиотеки
void main()
```

```

{
char Dat,Dat1,DatL,DatH; // описание идентификаторов
TRISA = 0xFF; // порт A на ввод данных
TRISB = 0x00; // порт B на вывод данных
PORTB = 0; // инициализация порта B
while(1) // внешний цикл
{
while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
DatL = PORTA; // ввод младших битов вводимого первого числа
DatL = DatL & 0x0F; // маскирование младших битов
PORTB = DatL; // вывод младших битов на дисплее U3
while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
DatH = PORTA; // ввод старших битов вводимого первого числа
DatH = DatH << 4; // сдвиг старших битов
Dat = DatH + DatL; // формирование первого введенного числа
PORTB = Dat; // вывод первого числа
while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
DatL = PORTA; // ввод младших битов вводимого второго числа
DatL = DatL & 0x0F; // маскирование младших битов
PORTB = DatL; // вывод младших битов второго числа на дисплее U3
while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
DatH = PORTA; // ввод старших битов вводимого второго числа
DatH = DatH << 4; // сдвиг старших битов вводимого второго числа
Dat1 = DatH + DatL; // формирование второго введенного числа
PORTB = Dat1; // вывод второго числа
while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
PORTB = Dat1+Dat; // вывод суммы двух чисел
}
}

```

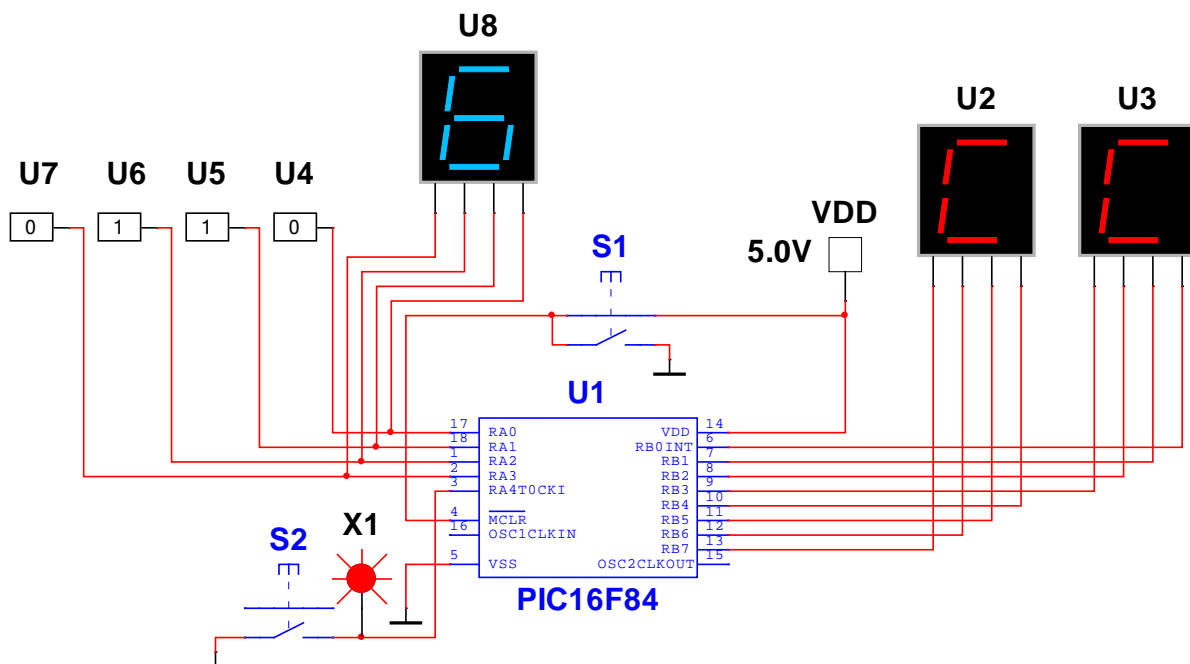


Рисунок 1. Схема моделирования пульта управления

Пульт ввода данных имеет существенное отличие от кнопок «пуск» и «стоп» [Там же]. Функционирование пульта ввода основано на применении пары операторов цикла. Эти операторы опрашивают состояние

вывода кнопки S2 по условию ноль (кнопка нажата) и по условию единица (кнопка отпущена). По умолчанию при открытии порта на ввод его выводы подключаются через подтягивающий резистор к источнику питания микропроцессора. При этом значение бита на порте ввода равно единице даже при отключенном выводе микропроцессора. Рассмотрим функционирование пары операторов цикла, опрашивающих состояние кнопки S2 (ENTER). Для этого запишем фрагмент программы, пронумеруем в нем строки и вставим, для изложения последовательности событий, операцию ввода данных с кнопочного пульта. Эта операция не отражена в листинге программы, т.к. её выполняет оператор при вводе данных в программу микропроцессора.

```
1. // набор данных на кнопочном пульте
2. while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
3. while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
4. // ввод данных кнопочного пульта из строки № 1
5. // набор данных на кнопочном пульте для ввода в строке № 8
6. while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
7. while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
8. // ввод данных кнопочного пульта из строки № 5
9. // набор данных на кнопочном пульте для последующего ввода данных в программу микропроцессора
   после выполнения операторов, расположенных в строках № 10 и № 11
10. while (RA4==0); // цикл по нулю бита A4 (ожидание S2 нажата)
11. while (RA4==1); // цикл по единице бита A4 (ожидание S2 отжата)
```

При наборе данных оператором в строке № 1 программа находится в режиме ожидания нажатия кнопки S2. При этом выполняется оператор цикла в строке № 2. После нажатия кнопки S2 программа микропроцессора переходит на выполнение оператора в строке № 3. Наступает режим ожидания события – кнопка S2 отжата. После отпускания кнопки S2 программа выполняет операторы ввода данных в строке № 4 из строки № 1 (кнопка S2 выполняет функцию клавиши ENTER). При этом наступает событие – ожидание нажатия кнопки S2. Это позволяет оператору набрать данные для их ввода в программу микропроцессора в строке № 5. Далее процесс повторяется. В строке № 8 данные из строки № 5 вводятся в программу микропроцессора. Оператор набирает на пульте данные в строке № 9 и нажимает и отпускает кнопку S2.

Последовательность четырех операторов, расположенных в строках № 2, № 3, № 6 и № 7, позволяет создать цепочку событий для многократного ввода данных в программу микропроцессора.

При этом оператор while (RA4==0); обеспечивает паузу в программе для ввода данных, а оператор цикла while (RA4==1); обеспечивает последовательное выполнение нескольких событий ввода данных т.к. выполнение программы прерывается до тех пор, пока кнопка S2 нажата. При одиночном операторе while (RA4==0); выполнялся бы только ввод данных из первой строки. Программа работает намного быстрее реакции оператора, и все процедуры ввода осуществляются практически мгновенно, с первым введенным значением.

Методически важно обратить внимание учащихся на совпадение принципа действия рассмотренного пульта ввода данных с двухступенчатым (MS, мастер-помощник) триггером.

Список источников

1. **Васильев И. А.** Моделирование подключения внешних устройств с параллельным интерфейсом к микропроцессору в среде MULTISIM // Альманах современной науки и образования. 2016. № 10. С. 17-20.
2. **Васильев И. А.** Практическое занятие по моделированию пульта управления микропроцессором в среде MULTISIM // Альманах современной науки и образования. 2016. № 10. С. 20-23.
3. **Предко М.** PIC-микроконтроллеры: архитектура и программирование. М.: Издательский дом «ДМК Пресс», 2010. 512 с.
4. **Хернтер М. Е.** Multisim 7: современная система компьютерного моделирования и анализа схем электронных устройств. М.: Издательский дом «ДМК Пресс», 2006. 488 с.
5. **Шпак Ю. А.** Программирование на языке C для AVR и PIC микроконтроллеров. Киев: МК-Пресс, 2006. 400 с.

PRACTICAL LESSON ON MODELING THE KEYBOARD FOR DATA INPUT INTO THE MICROPROCESSOR PROGRAM IN THE MULTISIM ENVIRONMENT

Vasil'ev Igor' Aleksandrovich, Ph. D. in Technical Sciences
Bauman Moscow State University
ivasiliev@rslab.ru

The article considers the modeling of connection of the simplest keyboard for data input into the microprocessor program. The modeling is carried out in the integrated environment of electronic circuits development MULTISIM. The microprocessor program is developed in C language using the compiler built in MULTISIM. The microprocessor-based device does not have an operating system, therefore, it is important for students to have practical skills in creating and programming external interface elements.

Key words and phrases: port; data input/output; microprocessor; data input keyboard; interface; practical skills; skills of modeling.