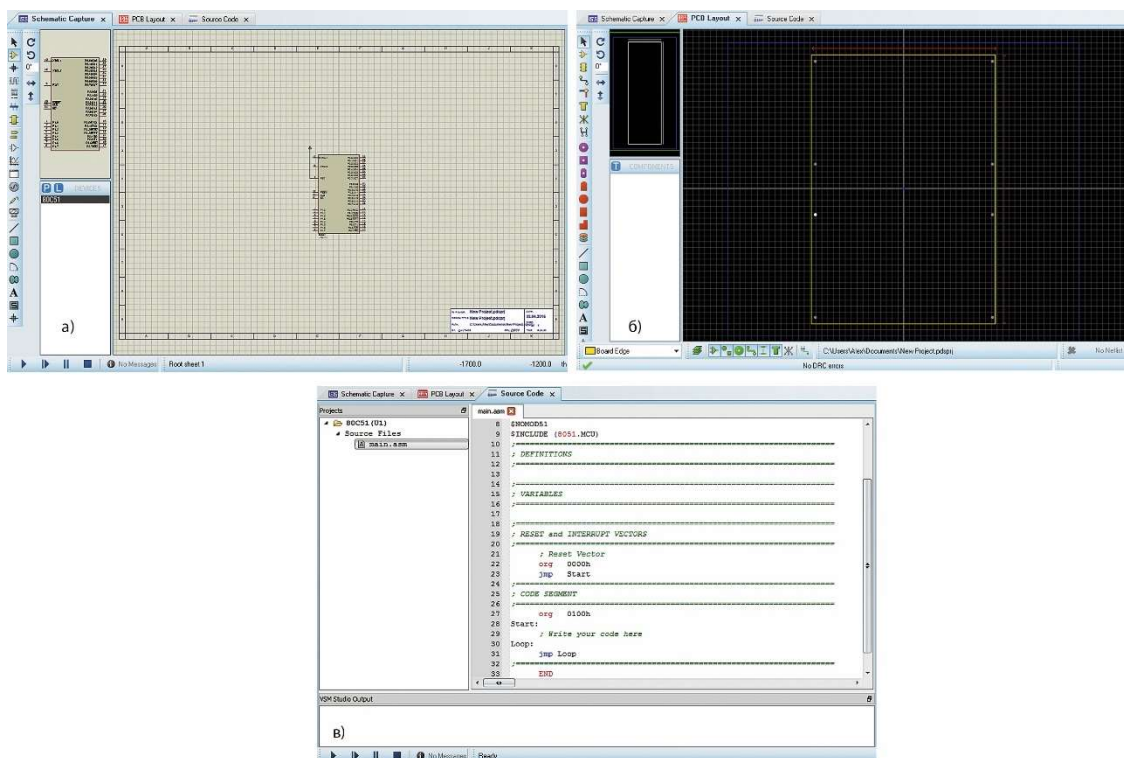


Результаты работы мастера New Project Wizard:

- а) рабочее поле чертежа;
- б) контур печатной платы;
- в) заготовка программного кода для микроконтроллера



Рабочее поле чертежа (рисунок 11а) — вкладка **Schematic Capture**. При этом на чертеже уже будет размещено условное графическое обозначение микроконтроллера выбранного семейства;

- контур печатной платы (рисунок 11б) — вкладка **PCB Layout**;
- заготовку программного кода для микроконтроллера (рисунок 11в) — вкладка **Source Code**.

Проектирование схемы, включающей микроконтроллер без помощи мастера

Проект схемы электрической принципиальной, в котором присутствует микроконтроллер, можно создать и без использования мастера — при помощи кнопки **ISIS** верхней панели инструментов Proteus. В результате откроется новая вкладка **Schematic Capture**, в рабочем поле которой и будет выполняться разработка схемы.

Выбор компонентов из базы данных для последующего их размещения в рабочей области программы производится в окне **Pick Devices** (рисунок 12). Данное окно можно открыть командой контекстного меню **Place/Component/From Libraries** или посредством нажатия на кнопку **P** на панели **DEVICES** (по умолчанию данная панель расположена в левой части программы и содержит список имеющихся в проекте компонентов). Открыть

Рисунок 12

Библиотека микроконтроллеров семейства 8051 программы Proteus



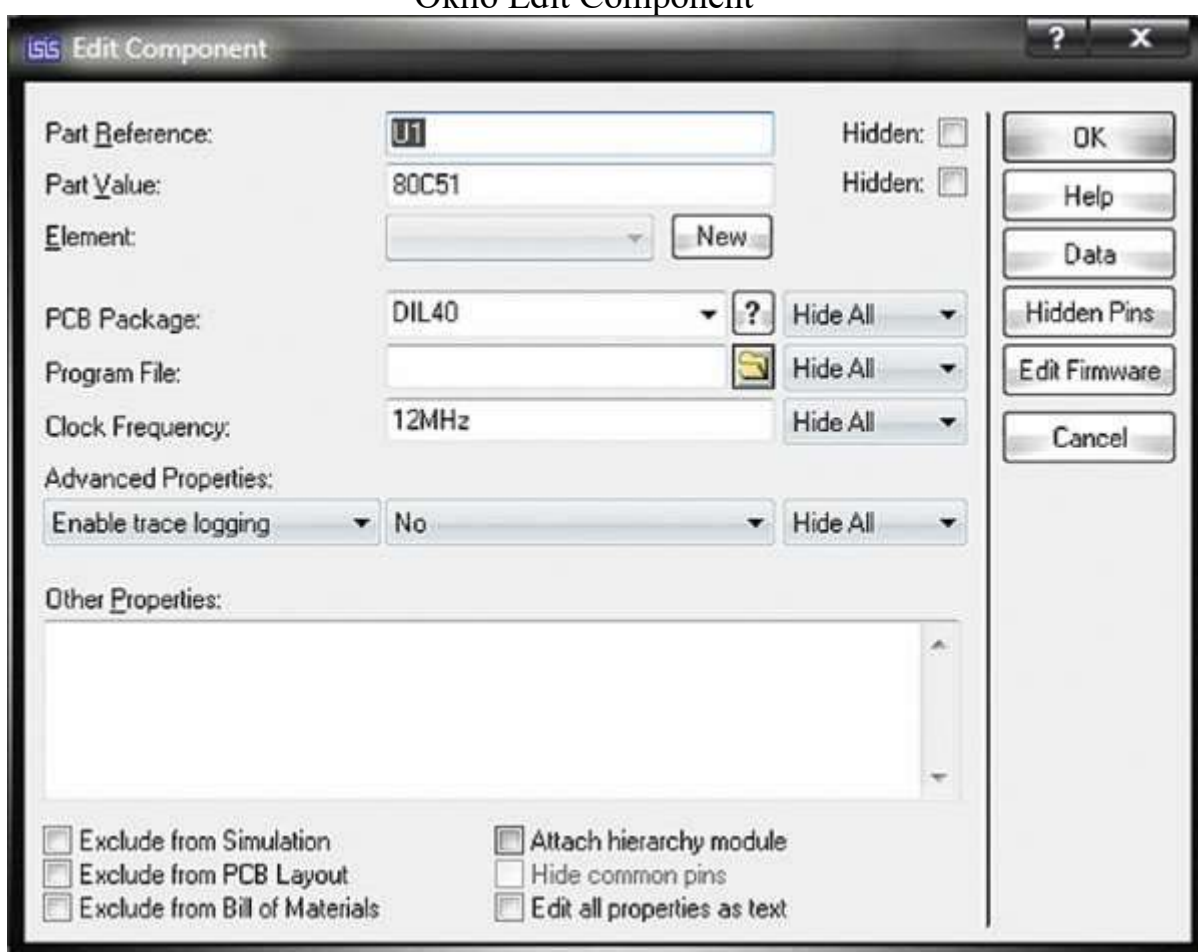
необходимо в окне **Pick Devices** нажать кнопку **OK**. После чего данное окно будет закрыто, а символ компонента будет прикреплен к курсору мыши, при помощи которого необходимо поместить символ в нужное место на схеме (щелкнуть в нужном месте схемы левой кнопкой мыши).

Кварцевый резонатор, конденсаторы и элементы питания в данном случае в схему можно не добавлять, так как они эмулируются программно. Однако если вы будете разрабатывать проект до его логического конца, то есть до изготовления печатной платы, элементы придется добавить.

Параметры размещенного на схеме символа микроконтроллера при необходимости редактируются в окне **Edit Component** (рисунок 13). Данное окно можно открыть путем двойного щелчка левой кнопки мыши по уже размещенному в рабочем поле программы символу компонента.

Рисунок 13

Окно Edit Component

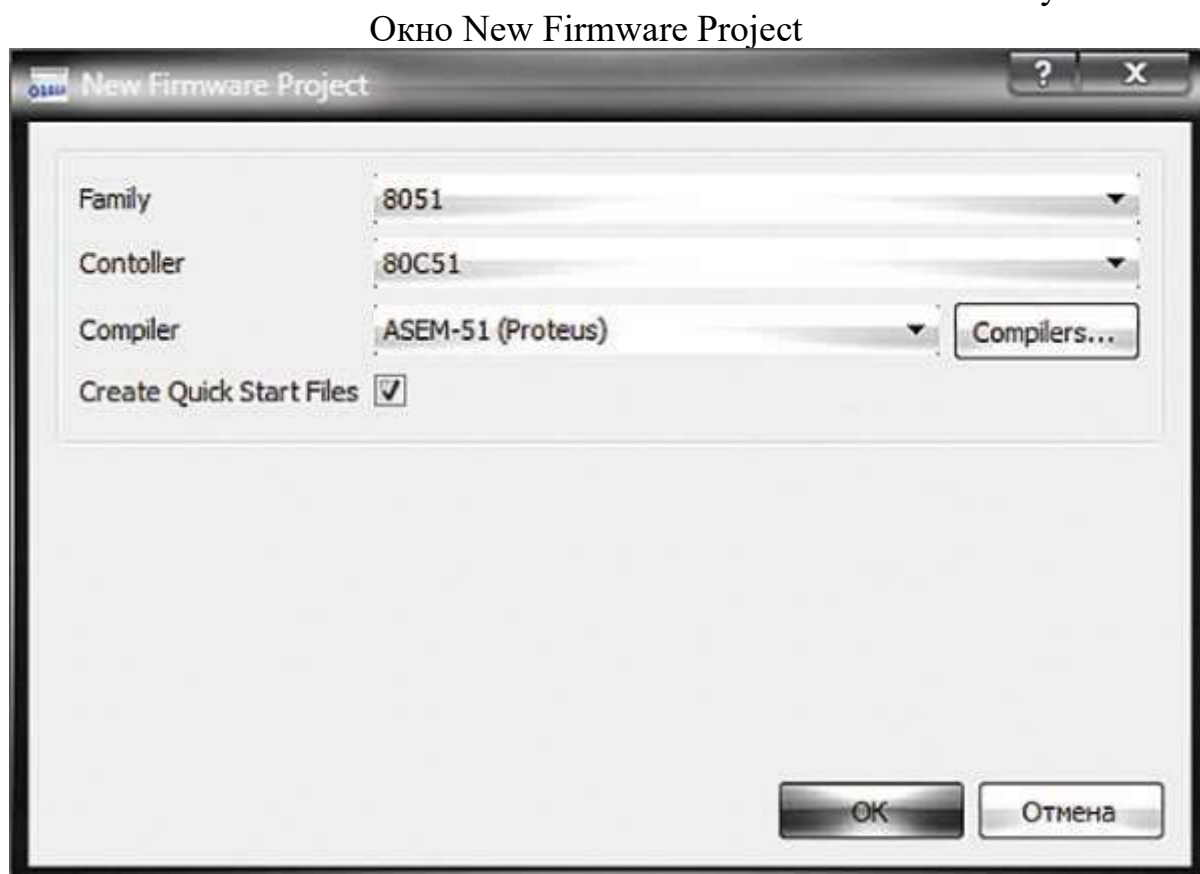


Открыть вкладку, на которой в процессе проектирования будет вноситься код программы инициализации микроконтроллера, можно следующим образом. Выделите при помощи левой кнопки мыши символ микроконтроллера в рабочем поле проекта, при помощи правой кнопки мыши вызовите контекстное меню и выберите в нем пункт **Edit Source Code**. В результате в проект будет добавлена вкладка **Source Code** и открыто окно

выбора микроконтроллера **New Firmware Project** (рисунок 14), в котором устанавливаются следующие параметры:

- **Family**— семейство микроконтроллера;
- **Controller**— модель микроконтроллера;
- **Compiler**— инструменты ассемблера/компилятора;
- **Create Quick Start Files**— автоматическое создание заготовки программного кода для микроконтроллера.

Рисунок 14



В проект можно также добавить и уже подготовленный ранее *.asm*-файл. Для этого на панели дерева проектов **Projects** выберите при помощи левой кнопки мыши проект, к которому необходимо добавить *.asm*-файл, при помощи правой кнопки мыши вызовите контекстное меню и выберите в нем пункт **Add Files**. В результате выполненных действий будет открыто окно проводника **Windows**, где необходимо выбрать на диске компьютера нужный *.asm*-файл, а затем нажать на кнопку «Открыть».

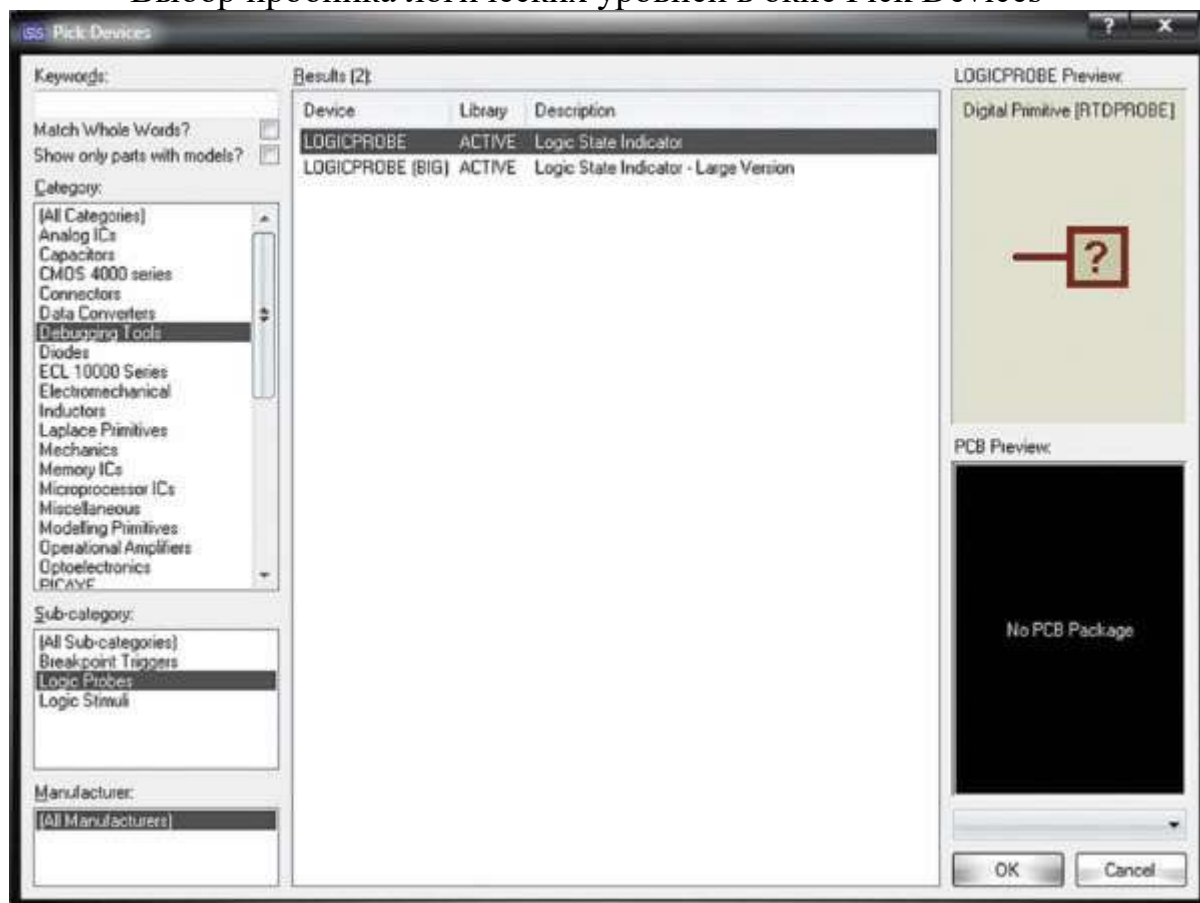
Моделирование схемы, включающей микроконтроллер, и трансляция программного кода

Для наглядной демонстрации работы программы инициализации микроконтроллера воспользуемся цветными пробниками логических уровней 0 и 1. Для того чтобы добавить пробник в рабочее поле проекта, необходимо вызвать окно **Pick Devices** (рисунок 15) и в поле **Category** выбрать из списка библиотеку **Debugging Tools**, в поле **Sub-category** выбрать строку **Logic**

Probes, а в поле **Results** — строку LOGICPROBE, затем нажать кнопку **ОК**. В результате окно **Pick Devices** будет закрыто, а символ пробника будет прикреплен к курсору мыши, при помощи которого необходимо поместить символ в требуемое место на схеме — щелкнуть в нужном месте схемы левой кнопкой мыши столько раз, сколько пробников надо разместить.

Рисунок 15

Выбор пробника логических уровней в окне Pick Devices



Подсоединим пробники к исследуемым линиям портов микроконтроллера. В результате при появлении на выходе линии порта значения логической единицы пробник будет подсвечен красным цветом, при появлении же значения логического нуля пробник будет подсвечен синим цветом. Также на пробниках визуальнo отображаются значения 0 и 1.

После того как в рабочей области проекта собрана схема (рисунок 16), а на вкладке **Source Code** введен код программы (рисунок 17), можно запускать моделирование.

Рисунок 16

Демонстрационная схема с использованием микроконтроллера и
цветных пробников логических уровней

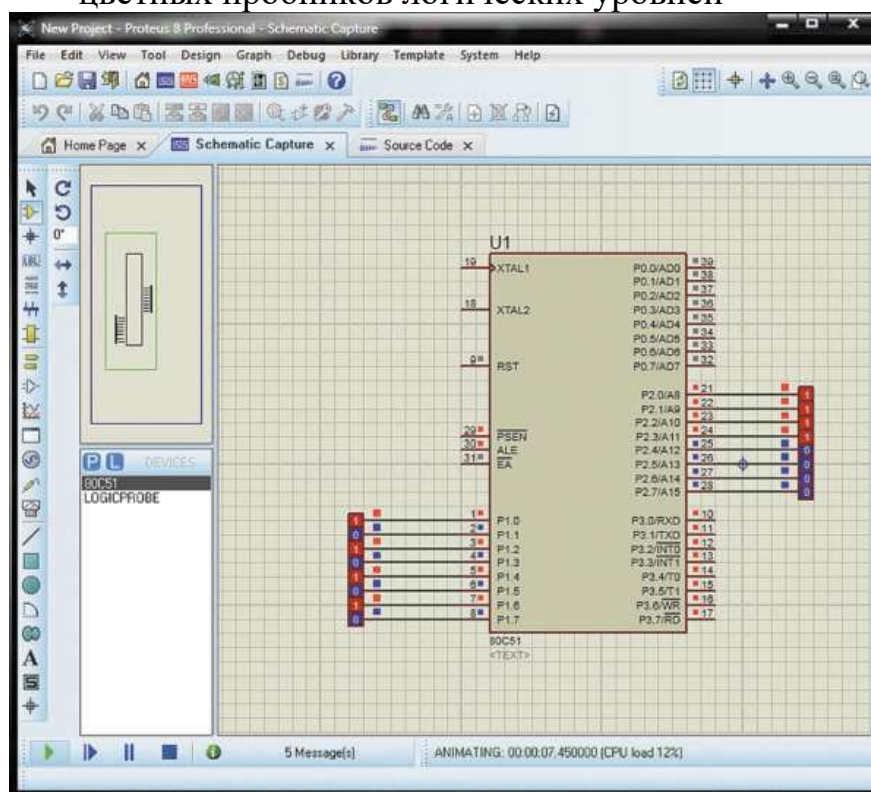


Рисунок 17

Программа инициализации микроконтроллера

```

9  $NOMOD51
10 $INCLUDE (8051.MCU)
11
12
13
14 ; Reset Vector
15 org 0000h
16 jmp Start
17
18 ;-----
19 ; CODE SEGMENT
20 ;-----
21
22 org 0100h
23 Start:
24 mov R5, #00001111b
25 mov R2, R5
26 mov R4, #01010101b
27 mov P1, R4
28 Loop:
29 jmp Loop
30 ;-----
31 END
32
33

```

VSM Studio Output
Processed 162 lines.
Compiled successfully.

Процесс моделирования запускается при помощи кнопки **Run the simulation**, которая находится в левом нижнем углу окна программы. Для того чтобы временно приостановить процесс симуляции, используйте кнопку **Pause the simulation, or start up at time 0 if stopped** (кнопка находится в левом нижнем углу окна программы). Остановить моделирование можно при помощи кнопки **Stop the simulation**.

В результате, если компилятор в листинге программы не обнаружит ошибок, на диске вашего компьютера в рабочей папке проекта будут созданы следующие файлы:

- *.hex— файл машинного кода (рисунок 18);
- *.asm— файл с ассемблерным кодом программы (рисунок 19);
- *.sdi— файл отладки программы инициализации, содержимое которого можно также просмотреть на вкладке **8051 CPU Source Code** (рисунок 20).

Рисунок 18

hex-файл машинного кода

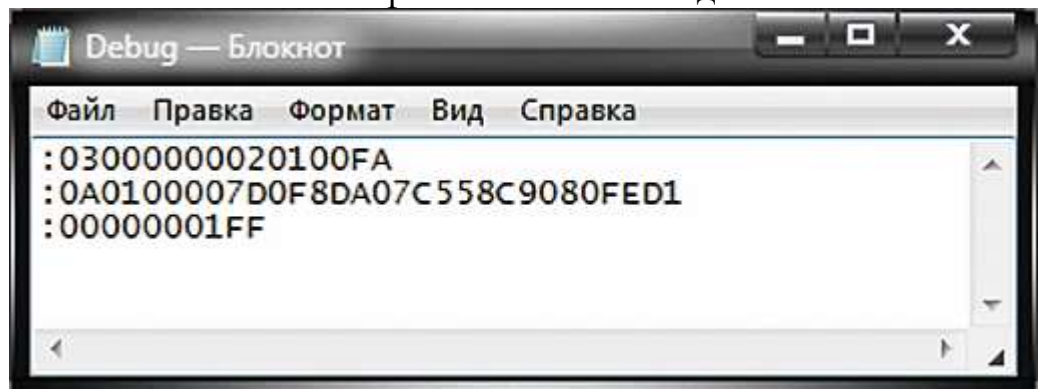
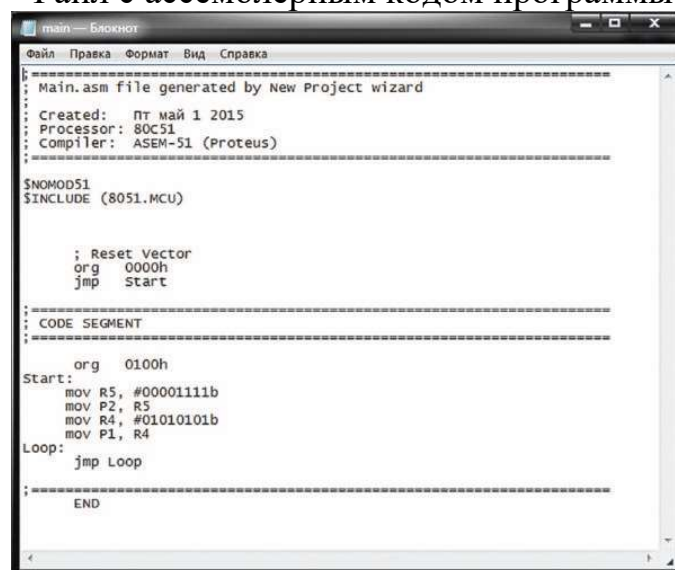


Рисунок 19

Файл с ассемблерным кодом программы



Вкладка 8051 CPU Source Code

```

1 -----
2 -----
3 -----
4 -----
5 -----
6 -----
7 -----
8 -----
9 -----
10 -----
11 -----
12 -----
13 -----
14 -----
15 -----
16 -----
17 -----
18 -----
19 -----
20 -----
21 -----
22 -----
23 -----
24 -----
25 -----
26 -----
27 -----
28 -----
29 -----
30 -----
31 -----
32 -----
33 -----
34 -----
35 -----
36 -----
37 -----
38 -----
39 -----
40 -----
41 -----
42 -----
43 -----
44 -----
45 -----
46 -----
47 -----
48 -----
49 -----
50 -----
51 -----
52 -----
53 -----
54 -----
55 -----
56 -----
57 -----
58 -----
59 -----
60 -----
61 -----
62 -----
63 -----
64 -----
65 -----
66 -----
67 -----
68 -----
69 -----
70 -----
71 -----
72 -----
73 -----
74 -----
75 -----
76 -----
77 -----
78 -----
79 -----
80 -----
81 -----
82 -----
83 -----
84 -----
85 -----
86 -----
87 -----
88 -----
89 -----
90 0000 02 01 00
91 -----
92 -----
93 -----
94 -----
95 -----
96 -----
97 0100
98 0100 7D 0F
99 0102 8D A0
100 0104 7C 55
101 0106 8C 90
102 0108
103 010B 80 FE
104 -----
105 -----

```

Программа инициализации микроконтроллера пишется на языке программирования asm51. Это язык ассемблера, специально предназначенный для написания программ для микроконтроллеров семейства x51. Написанная программа транслируется с помощью ассемблера asm51, в результате чего будут получены *hex*-файл для загрузки в память программ и листинговый файл с кодами команд и адресами расположения этих кодов. Файл листинга формируется ассемблером в процессе трансляции и представляет собой начальный ассемблерный файл, дополненный следующей информацией. Перед каждой командой размещен номер ее строки в ассемблерном тексте, адрес ячейки памяти программ, в которой размещен код операции команды,

а после этого адреса — один, два или три байта самой команды. Также в файле листинга находится таблица имен и меток.

Если в ходе компиляции кода программы инициализации микроконтроллера обнаружены ошибки, они будут отмечены компилятором в файле **.lst*. При этом *hex*-файл не будет сгенерирован до тех пор, пока ошибки не устранены. В файле листинга для каждой обнаруженной ошибки компилятор указывает причину ошибки и номер строки, в которой она обнаружена. На рис. 21 представлен фрагмент файла **.lst*, в котором отладчик сообщает об ошибке номер 2 (неопределенный идентификатор), состоящей в том, что в коде программы неверно указано название порта ввода/вывода данных (в используемом микроконтроллере нет порта с названием P6). Отладка программы инициализации выполняется на вкладке **8051 CPU Source Code**, которая отображается после приостановки процесса симуляции. Для того чтобы временно приостановить симуляцию схемы, используйте кнопку **Pause the simulation, or start up at time 0 if stopped** (кнопка находится в левом нижнем углу окна программы). Отчет об ошибках, полученных в результате трансляции программного кода, отображается в нижней части вкладки **Source Code** на панели **VSM Studio Output** (рисунок. 22).

Рисунок 21

Фрагмент файла листинга **.lst*

```

74: 1      N      D4      RS1      BIT      0D4H
75: 1      N      D5      F0       BIT      0D5H
76: 1      N      D6      AC       BIT      0D6H
77: 1      N      D7      CY       BIT      0D7H
78: 1
79: 1      N      0000      RESET    CODE    000H
80: 1      N      0003      EXTI0    CODE    003H
81: 1      N      0008      TIMER0   CODE    00BH
82: 1      N      0013      EXTI1    CODE    013H
83: 1      N      0018      TIMER1   CODE    018H
84: 1      N      0023      SINT     CODE    023H
85:
86:
87:
88:
89:      N      0000      ; Reset Vector
90: 0000 02 01 00      org 0000h
91:
92:
93:      ;=====
94:      ; CODE SEGMENT
95:      ;=====
96:      N      0100      org 0100h
97: 0100      start:
98: 0100 7D 0F      mov R5, #00001111b
99: 0102 8D A0      mov P2, R5

ASEM-51 V1.3                                     Copyright (c) 2002 by w.w. Heinz

Line I  Addr  Code      Source
100: 0104 7C 55      mov R4, #01010101b
101: 0106 8C 00      mov P6, R4
      ^
      @@0000 symbol not defined @@0000
102: 0108      Loop:
103: 0108 80 FE      jmp Loop
104:
105:
106:      ;=====
      END
  
```

Рисунок 22

Отчет об ошибках, полученных в результате трансляции программного кода

```

VSM Studio Output
asm.exe "..\main.asm" "Debug.HEX" Debug.lst /INCLUDES:"C:\Program Files\Labcenter Electronics\Proteus 8 Professional\Tools\ASSEMBLER"
MCS-51 Family Macro Assembler: ASEM-51 V1.3
main.asm(77): symbol not defined
1 error detected
make: *** [Debug.HEX] Error 1
Error code 2
  
```

Состояние памяти и регистров микроконтроллера можно просмотреть при помощи следующих команд основного меню программы:

- **Debug/8051 CPU Registers** (рисунок 23а);
- **Debug/8051 CPU SFR Memory** (рисунок 23б);
- **Debug/8051 CPU Internal (IDATA) Memory** (рисунок 23в).

Рисунок 23

Состояние:

а) регистров;

б) памяти SFR;

в) внутренней памяти микроконтроллера

а)

8051 CPU Registers - U1		8051 CPU Internal (IDATA) Memory - U1		8051 CPU SFR Memory - U1	
8051 CPU Registers - U1					
PC	INSTRUCTION				
0108	SJMP 0108				
ACC B	DPTR	SP	CA-rs0-P		
00	00	0000	07	00000000	
R0	R1	R2	R3	R4	R5
00	00	00	00	55	0F 00 00
P0	P1	P2	P3	SCON	SBUF
FF	55	0F	FF	00	00
TMR0	TMR1	TMOD	TCON	PCON	
0000	0000	00	00	00	
IE	IP	TMR2	TCON	RCAP	
00	00	0000	00	0000	

б)

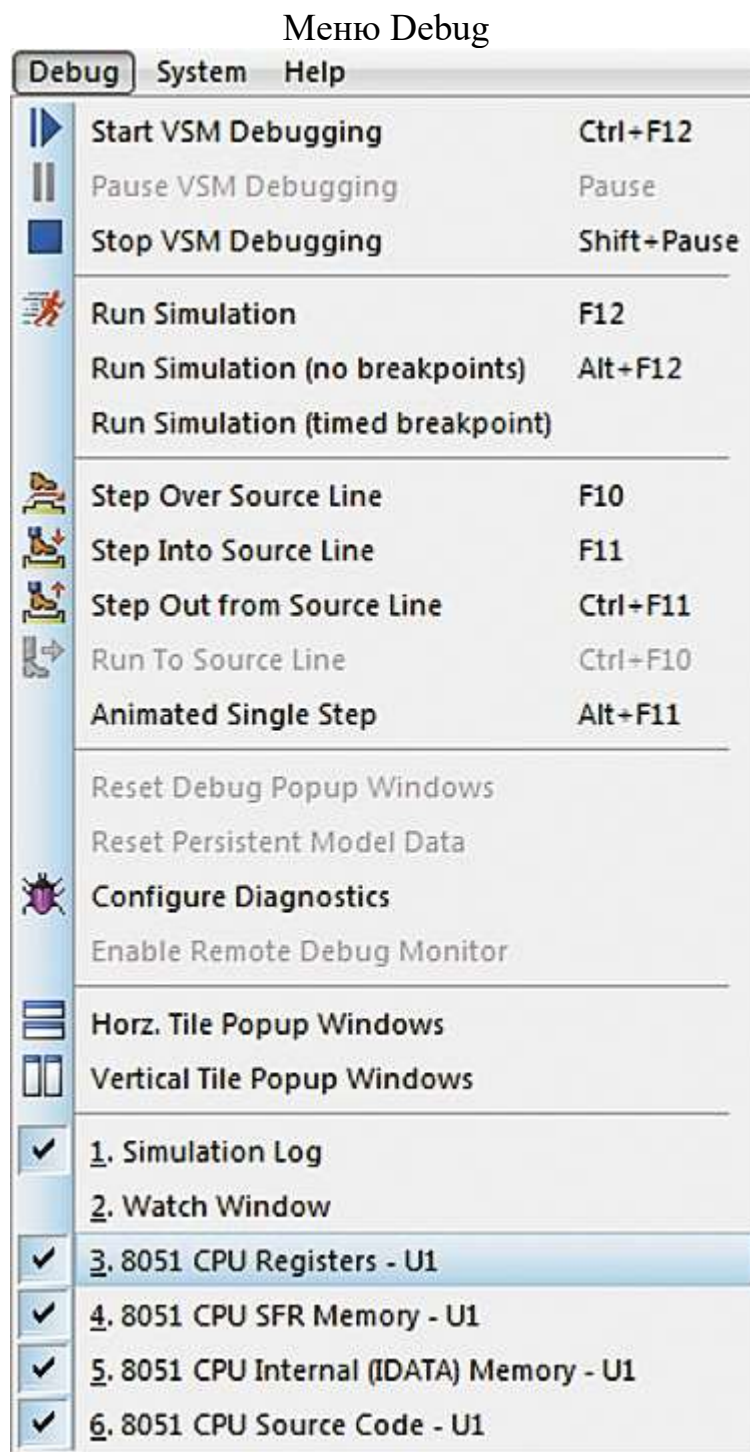
8051 CPU Registers - U1		8051 CPU Internal (IDATA) Memory - U1		8051 CPU SFR Memory - U1	
8051 CPU SFR Memory - U1					
80	FF	07	00	00	00
94	00	00	00	00	00
A8	00	00	00	00	00
BC	00	00	00	00	00
D0	00	00	00	00	00
E4	00	00	00	00	00
F8	00	00	00	00	00

в)

8051 CPU Registers - U1		8051 CPU Internal (IDATA) Memory - U1		8051 CPU SFR Memory - U1	
8051 CPU Internal (IDATA) Memory - U1					
00	00	00	00	00	00
14	00	00	00	00	00
28	00	00	00	00	00
3C	00	00	00	00	00
50	00	00	00	00	00
64	00	00	00	00	00
78	00	00	00	00	00

Меню **Debug** (рисунок 24) доступно для вызова во время приостановки симуляции схемы.

Рисунок 24



Проанализируем работу демонстрационной схемы, представленной на рисунке 16. На вкладке **main.asm** (рисунок 17) при помощи команды **mov** были даны указания программе инициализации микроконтроллера записать в порт P2 следующие значения линий порта P2.7–P2.0 — «00001111», а в порт P1 — значения линий порта P1.7–P1.0 — «01010101». После запуска моделирования при помощи цветных пробников мы можем проверить

правильность работы программы. Как видно из рис. 16, на выводах портов микроконтроллера P1 и P2 действительно присутствует та комбинация, которую мы указали в коде программы.

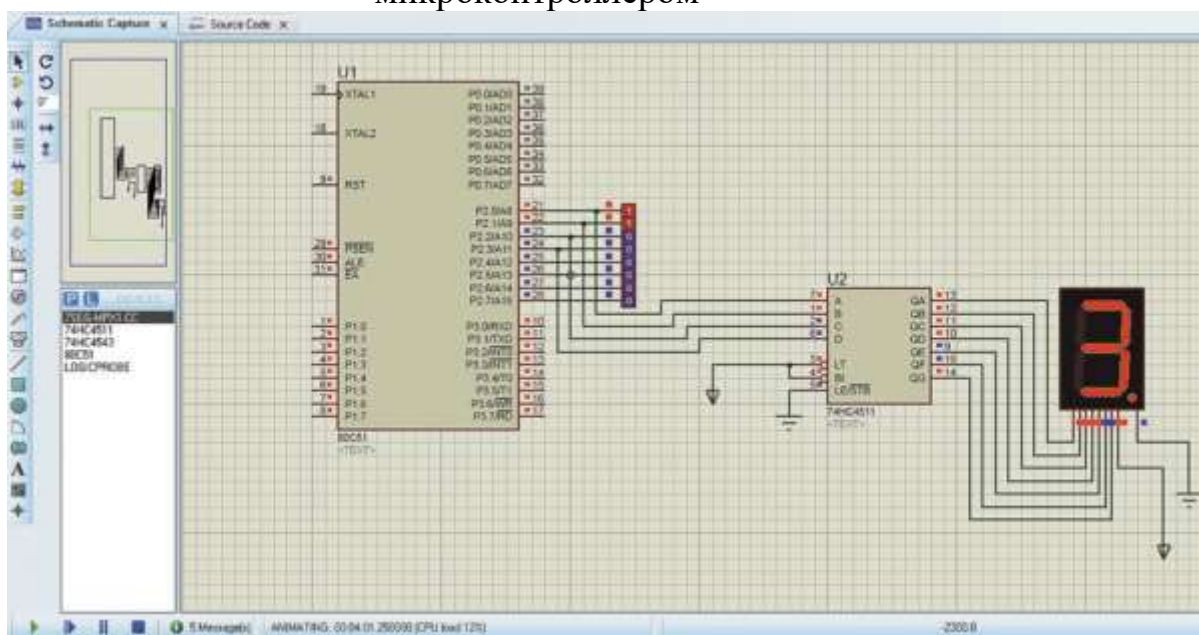
Сопряжение микроконтроллера x51 с микросхемами 7-сегментных индикаторов в Proteus

Система, в которой используется микроконтроллер, может не только чем-то управлять, но и что-то отображать. Чаще всего в качестве узла отображения в схеме используют знаковосинтезирующие индикаторы. По способу формирования символов индикаторы бывают 7-сегментными или матричными. Для отображения цифр удобнее использовать 7-сегментный индикатор. Такие индикаторы широко применяются в самых разных устройствах цифровой техники, от калькулятора до электронных часов. Матричные индикаторы обычно используют для отображения букв. Каких-либо стандартных правил сопряжения микроконтроллера с индикаторами не существует, и в каждом конкретном случае сопряжение может выполняться по-разному.

На рисунке 25 представлен пример соединения микросхемы индикатора с микроконтроллером. Принцип соединения микроконтроллера с элементами схемы такой же, как и для других компонентов схемы.

Рисунок 25

Сопряжение микросхемы 7 сегментного индикатора с микроконтроллером



Для того чтобы отобразить цифру, нужно «зажечь» определенные сегменты индикатора. При использовании в схеме микроконтроллера сделать это можно программным способом. Разные комбинации светящихся элементов индикатора, обеспечиваемые внешней коммутацией, позволяют отображать цифры от 0 до 9 и десятичную точку. В представленном примере у индикатора элементы имеют общий катод.

Заключение

Proteus - среда для проектирования и отладки электронных устройств, в т.ч. выполненных на основе микроконтроллеров различных семейств. Предоставляет возможности ввода схемы в графическом редакторе, моделирования её работы и разработки печатной платы, включая трехмерную визуализацию её сборки. Уникальной чертой среды Proteus является возможность эффективного моделирования работы разнообразных микроконтроллеров (PIC, 8051, AVR, HC11, ARM7/LPC2000 и др.) и отладки микропрограммного обеспечения.

Среда PROTEUS имеет огромную библиотеку электронных компонентов, а недостающие - можно сделать самостоятельно. Предусмотрена поддержка SPICE-моделей, которые часто предоставляются производителями электронных компонентов.

В комплект профессиональной версии входят инструменты USBCONN для подключения моделируемой схемы к реальному USB порту компьютера и COMPIR для подключения к COM-порту ПК.

Среда PROTEUS совместима с популярными средами разработки микропрограммного обеспечения.

Список использованных источников

1. ISIS Help. Labcenter Electronics, 2014.
2. Пашкевич А. П., Чумаков О. А. Микро-процессорные системы управления. Часть БГУИР, 2006.
3. 8051 CROSS ASSEMBLER USER'S MANUAL, MetaLink Corporation Chandler, Arizona, USA.
4. Максимов А. Моделирование устройств на микроконтроллерах с помощью программы ISIS из пакета PROTEUS VSM // Радио. 2005. № 4.
5. Довгун В. П. Компьютерное моделирование электронных цепей и устройств. Курс лекций. Лекция 1. Общая характеристика программ компьютерного моделирования электронных цепей. Красноярск, СФУ Институт космических и информационных технологий, 2008.
6. Магда Ю. С. Микроконтроллеры серии 8051: практический подход. М.: ДМК Пресс, 2008.