

Министерство образования Республики Беларусь
Учреждение образования
«Могилевский государственный университет имени А.А.Кулешова»

СИСТЕМА РАЗРАБОТКИ PROTEUS

Обзор
студента 3 курса специальности 1-40-01-01
«Программное обеспечение информационных технологий»
заочной формы получения образования, группы 5242201
Станислава Васильевича Микулича

Могилев
2023

Оглавление

Введение.....	3
Разработка электрических схем с использованием микроконтроллеров в программной среде Proteus.....	5
Проектирование схемы, включающей микроконтроллер, при помощи мастера New Project Wizard	5
Моделирование схемы, включающей микроконтроллер, и трансляция программного кода.....	17
Сопряжение микроконтроллера x51 с микросхемами 7-сегментных индикаторов в Proteus	25
Заключение	26
Список использованных источников	27

Введение

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 COMPIM для подключения к COM-порту ПК.

Среда PROTEUS совместима с популярными средами разработки микропрограммного обеспечения, в т.ч.:

- CodeVisionAVR (только МК AVR)
- IAR (любые МК)
- ICC (МК AVR, msp430, ARM7)
- WinAVR (МК AVR)
- Keil (МК 8051 и ARM)
- HiTECH (МК 8051 и PIC)

Отличительные особенности:

- Разработка схемы электрической принципиальной (ввод в графическом редакторе)
- Моделирование схемы с использованием разнообразных виртуальных приборов
- Разработка печатной платы, включая 3D-визуализацию её сборки
- совместное моделирование работы микроконтроллера, исполняющего заданную программу, и окружающих его аналоговой и цифровой схем
- широкие отладочные возможности, в т.ч. доступ к содержимому регистров и памяти, задание точек останова программы, пошаговое выполнение
- отладка на уровне исходного кода (Си, Бейсик, Ассемблер, в зависимости от типа используемого для отладки файла с испытываемым микропрограммным обеспечением)
- поддержка нескольких семейств микроконтроллеров от разных производителей, в т.ч.:
 - PIC12, PIC16, PIC18 и PIC24 (Microchip).
 - 8051/8052, в т.ч. производные от них, выпускаемые Philips и Atmel

- AVR, Tiny AVR и Mega AVR (Atmel)
- ARM7, в т.ч. LPC2000 (NXP)
- HC11 (Freescale) и микроконтроллерные модули BASIC Stamp (Parallax)

- ведется работа по добавлению поддержки других МК

Преимущества

- Выполнение всех этапов разработки электронного устройства на основе микроконтроллера в единой среде
- Возможность написания, отладки и тестирования микропрограммного обеспечения еще до физического изготовления опытного образца системы.
- Генерируемые диагностические сообщения (например, при выполнении непредусмотренной инструкции) как со стороны ЦПУ, так и со стороны моделей устройств ввода-вывода, позволяют обнаружить сложные в поиске ошибки программирования
- Ускоряет процесс разработки электронного устройства
- Поддержка совместной работы с аппаратными устройствами, подключенными через порт компьютера

Разработка электрических схем с использованием микроконтроллеров в программной среде Proteus

Proteus является так называемой средой сквозного проектирования, что позволяет создавать устройство, начиная с проектирования его принципиальной схемы и заканчивая изготовлением печатной платы, с возможностью контроля на каждом этапе производства.

Первый этап проектирования узла печатной платы в системе Proteus — разработка схемы электрической принципиальной, которая выполняется в редакторе ISIS. На этой стадии проектирования производится выбор необходимых компонентов, их размещение в рабочем поле чертежа, связь компонентов при помощи цепей и шин. При необходимости можно модифицировать свойства компонентов, добавлять текстовые надписи.

Рассмотрим процесс моделирования схем с использованием микроконтроллеров на примере микроконтроллера 80C51.

После создания пустого листа схемы его нужно заполнить символами необходимых компонентов из библиотеки. В Proteus создать новый проект схемы можно при помощи команды **File/New Project**. Необходимо отметить, что по умолчанию при формировании нового проекта запускается мастер **New Project Wizard**.

Проектирование схемы, включающей микроконтроллер, при помощи мастера **New Project Wizard**

Работа мастера состоит из нескольких этапов, на которых указываются название проекта и его месторасположение на диске компьютера (можно создать проект с чистого листа или на основе имеющихся разработок, поставляемых с системой), задается необходимость создания разработки ISIS (при этом указывается формат чертежа) и/или ARES, необходимость включения в проект определенного микроконтроллера. По окончании работы мастера в проект будет добавлено несколько вкладок, в рабочей области которых и будет проводиться дальнейшее проектирование схемы.

Рассмотрим более подробно работу с мастером **New Project Wizard**. Для этого запустим его при помощи команды **File/New Project** основного меню Proteus. Количество шагов мастера зависит от выбора переключателя в окне **New Project Wizard: Start**. В том случае если выбрана позиция **From Development Board** (рисунок 2), работа мастера состоит всего из одного шага, на котором для создания нового проекта предлагается использовать одну из имеющихся разработок, поставляемых вместе с Proteus. Выбор разработки производится в поле **Development Board** из одноименного списка путем выделения ее названия левой кнопкой мыши. В выпадающем списке **Microcontroller Family** можно задать семейство микроконтроллера. В таком случае в списке **Development Board** будут показаны разработки, в которых используется микроконтроллер указанного семейства. При этом описание выбранной разработки отображается в поле **Details**. Также необходимо указать название нового проекта (поле **Name**) и его месторасположение на диске компьютера (поле **Path**). А затем нажать кнопку **Finish** в нижней части

окна **New Project Wizard: Start**. В результате будет открыт новый проект, содержащий схему (рисунок 3а) и исходный код программы микроконтроллера (рисунок 3б), которые можно модифицировать на свое усмотрение.

Рисунок 2.

Окно New Project Wizard: Start, установка переключателя в позицию From Development Board

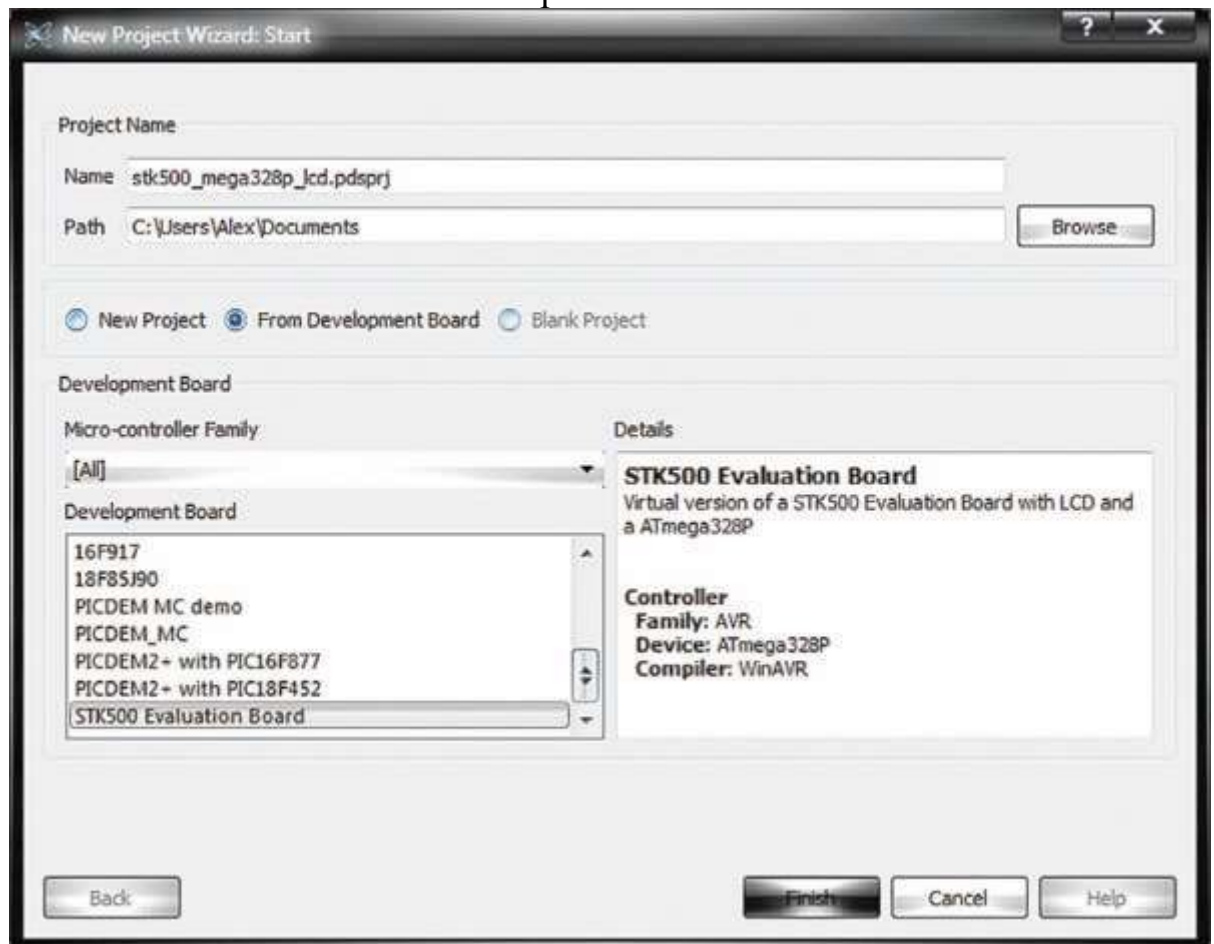
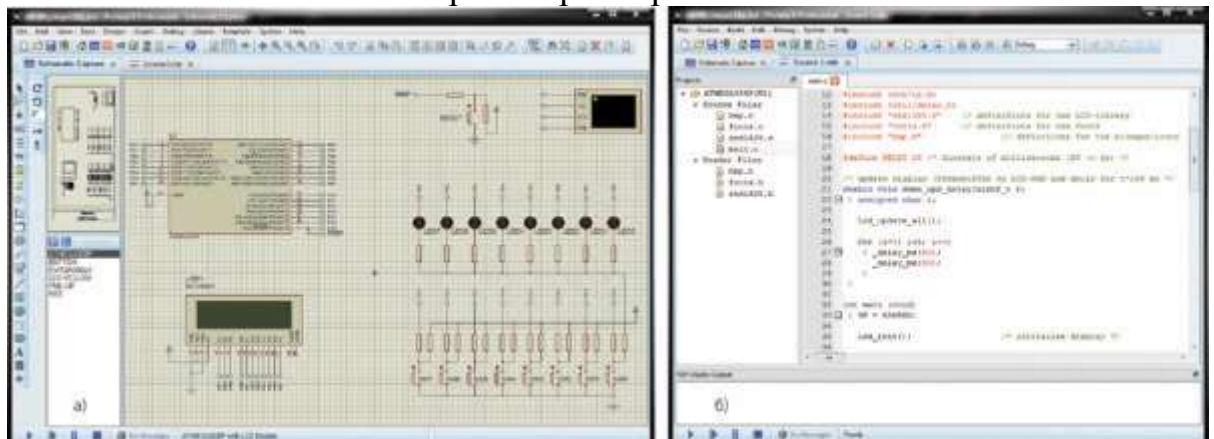
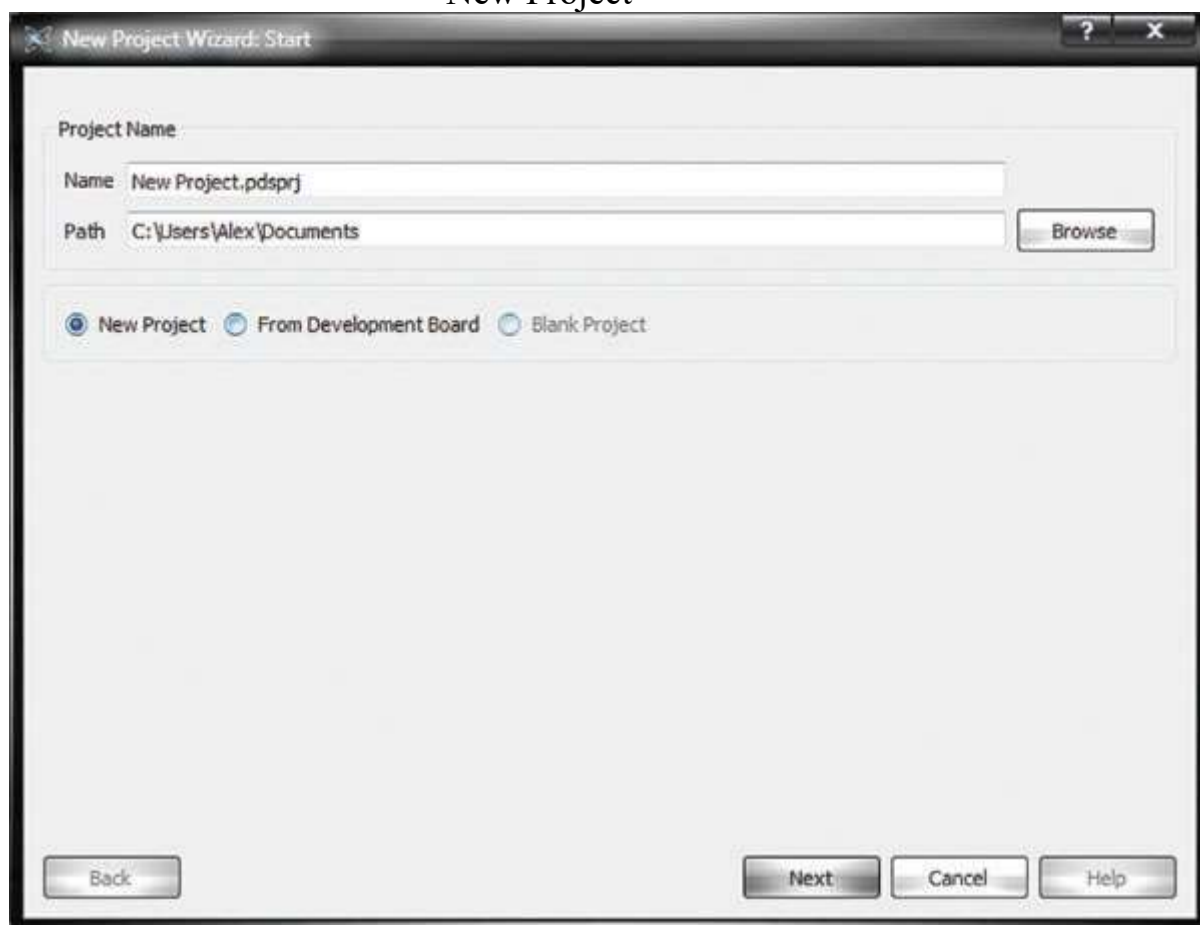


Рисунок 3.

Новый проект, созданный на основе уже имеющейся разработки: а) схема электрическая принципиальная; б) исходный код программы микроконтроллера



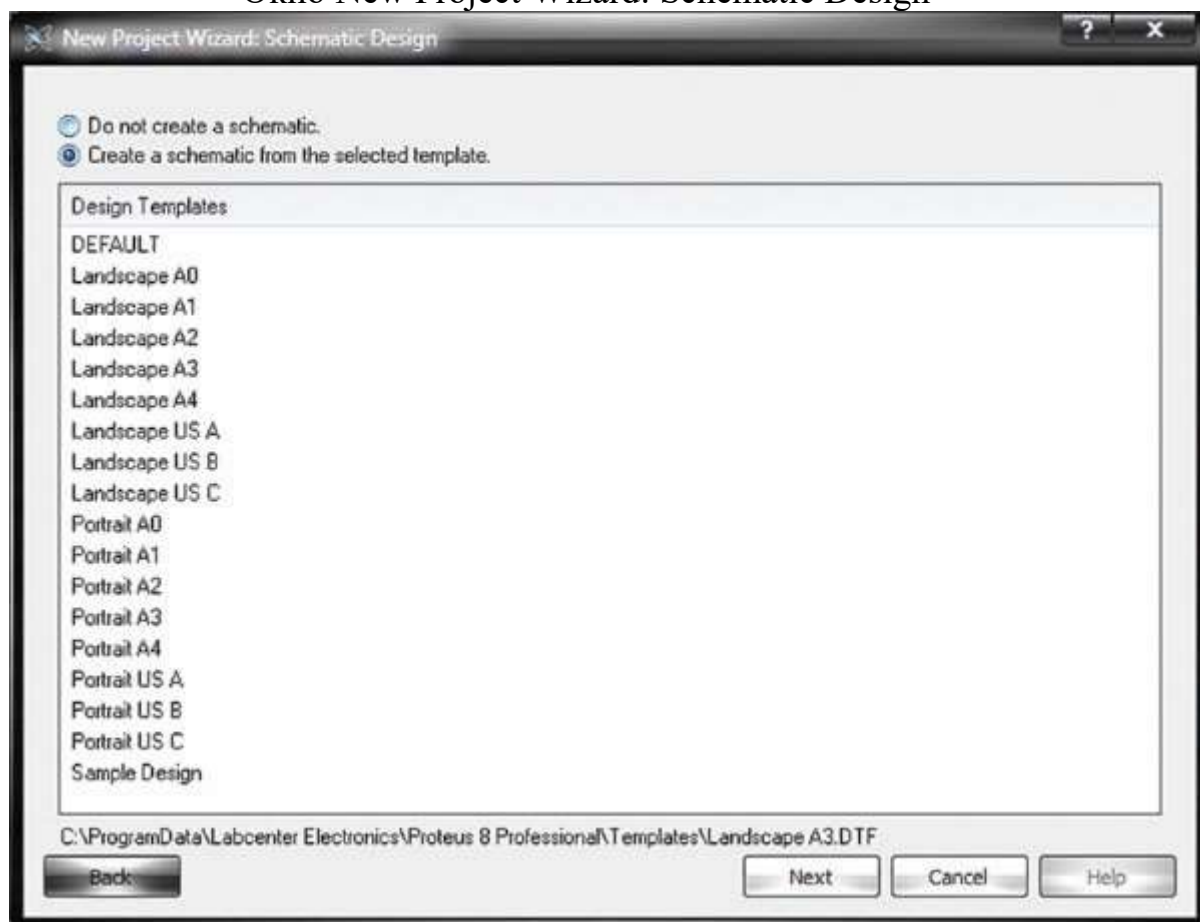
Окно New Project Wizard: Start, установка переключателя в позицию New Project



В случае если в окне **New Project Wizard: Start** переключатель установлен в позицию **New Project**, работа мастера будет состоять из шести шагов, на первом из которых (рисунок 4) пользователю предлагается указать название нового проекта (поле **Name**) и его месторасположение на диске компьютера (поле **Path**). После того как соответствующие поля заполнены, необходимо нажать кнопку **Next**, в результате будет открыто следующее окно мастера — **New Project Wizard: Schematic Design** (рисунок 5). На втором шаге мастер предложит указать необходимость создания разработки ISIS посредством установки переключателя в одну из двух позиций:

- **Do not create a schematic**— не создавать проект ISIS;
- **Create a schematic from the selected template**— создать проект ISIS (при этом в поле **Design Templates** задается формат чертежа).

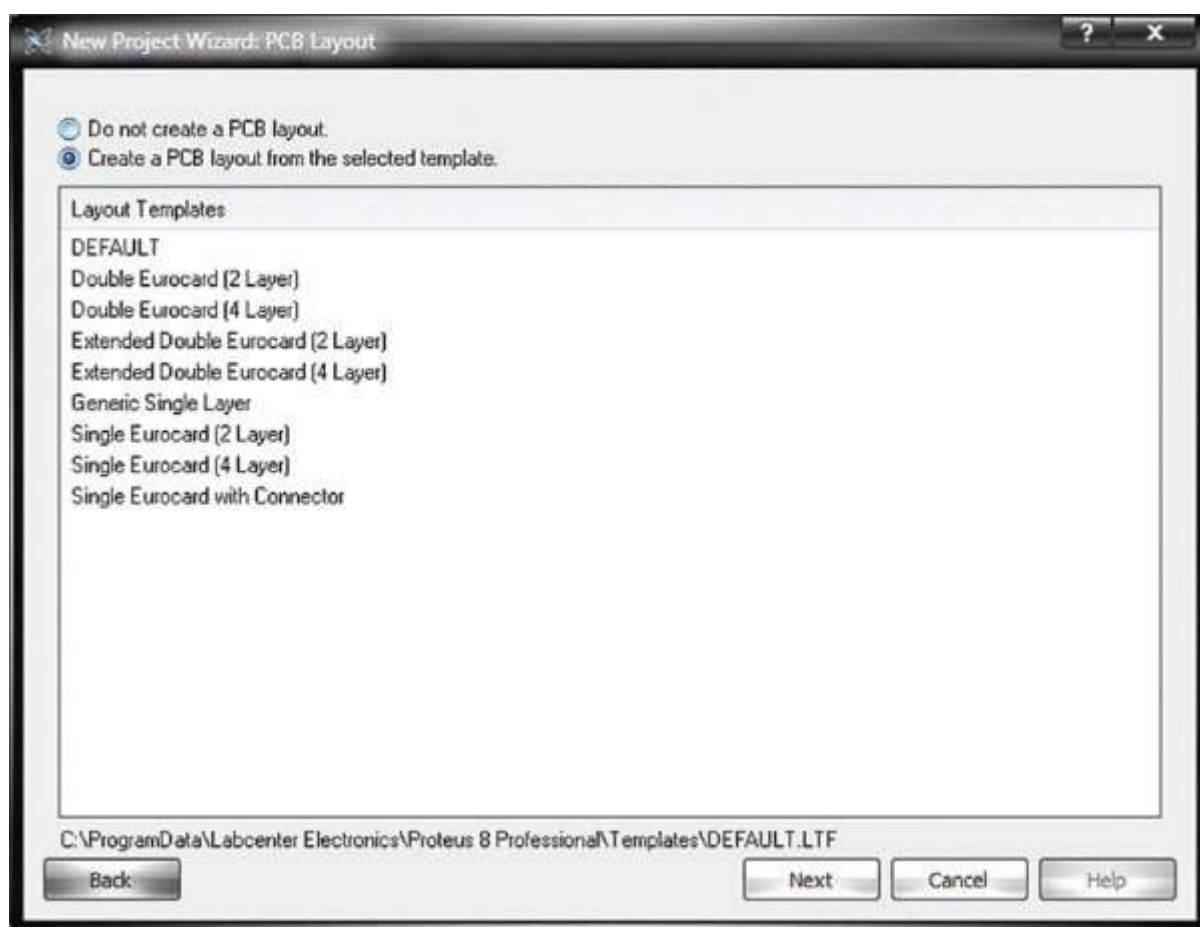
Окно New Project Wizard: Schematic Design



После установки всех параметров следует нажать кнопку **Next**, в результате чего будет открыто третье окно мастера — **New Project Wizard: PCB Layout** (рисунок 6). В нем посредством установки переключателя в нужную позицию необходимо произвести выбор одного из параметров:

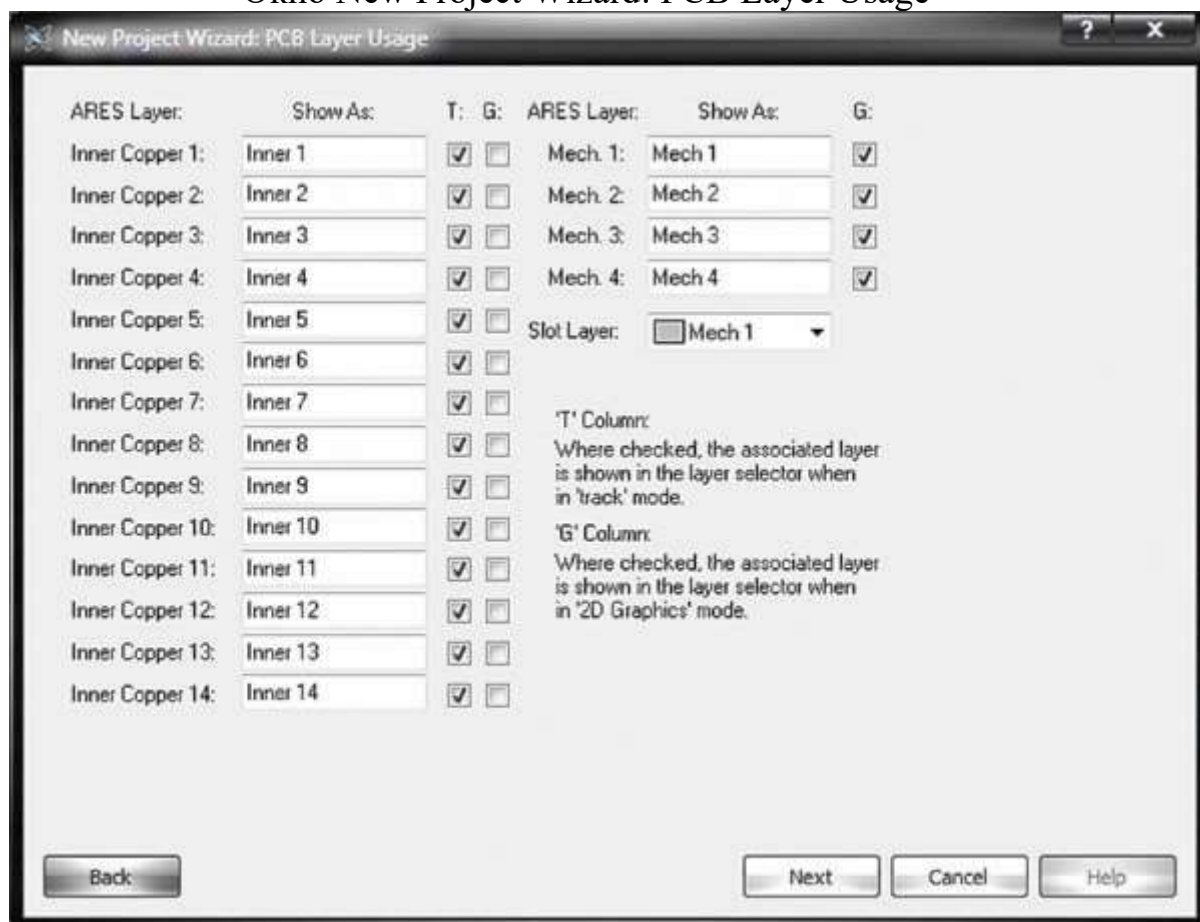
- **Do not create a PCB layout**— не создавать проект PCB;
- **Create a PCB layout from the selected template**— создать проект PCB (при этом в поле **Layout Templates** производится выбор шаблона проекта).

Окно New Project Wizard: PCB Layout



А затем нажать кнопку **Next** для перехода к следующему шагу мастера — **New Project Wizard: PCB Layer Usage** (рисунок 7), на котором выполняется настройка слоев платы.

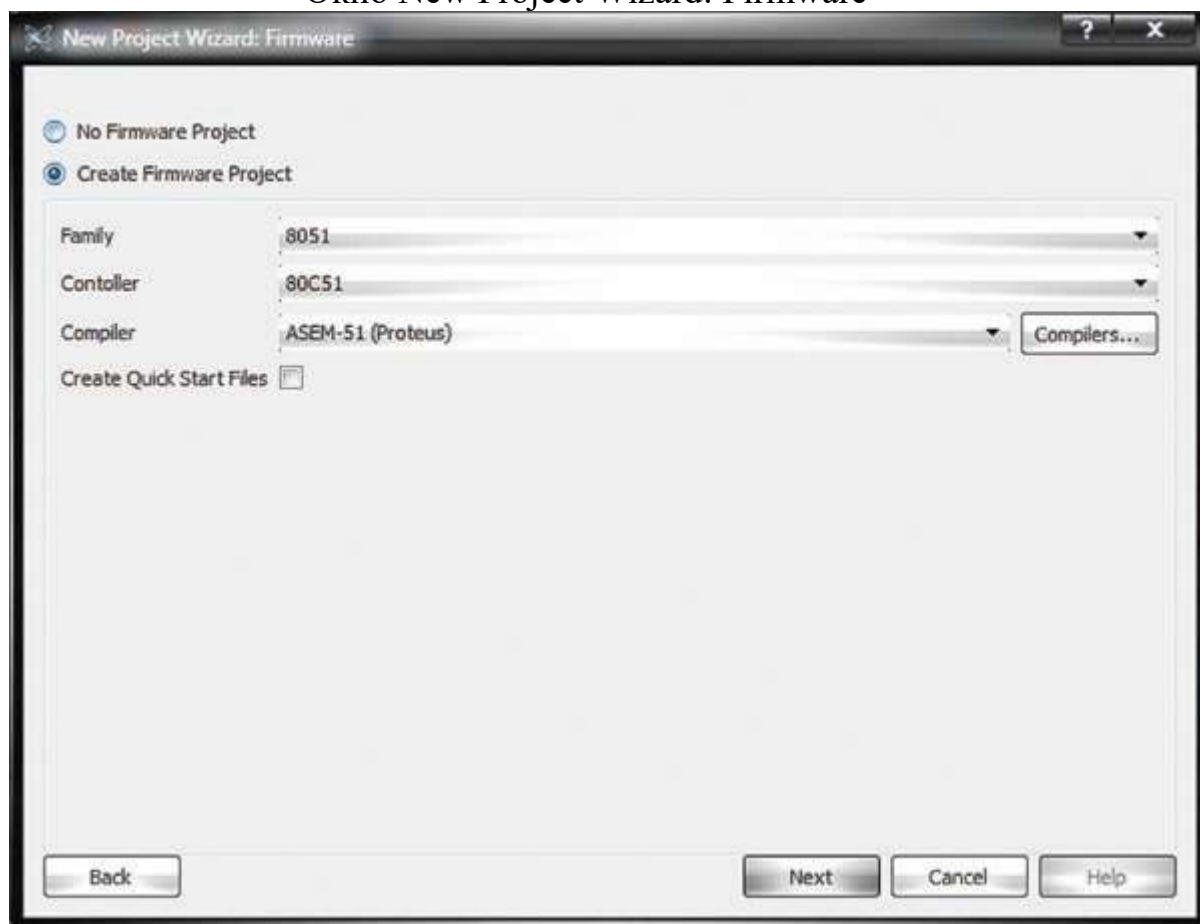
Окно New Project Wizard: PCB Layer Usage



Пятый шаг работы мастера **New Project Wizard: Firmware** (рисунок 8) — выбор микроконтроллера. На этом этапе путем установки переключателя в нужную позицию задается необходимость использования микроконтроллера в проекте схемы:

- **No Firmware Project**— микроконтроллер не используется;
- **Create Firmware Project**— создать проект с использованием микроконтроллера.

Окно New Project Wizard: Firmware

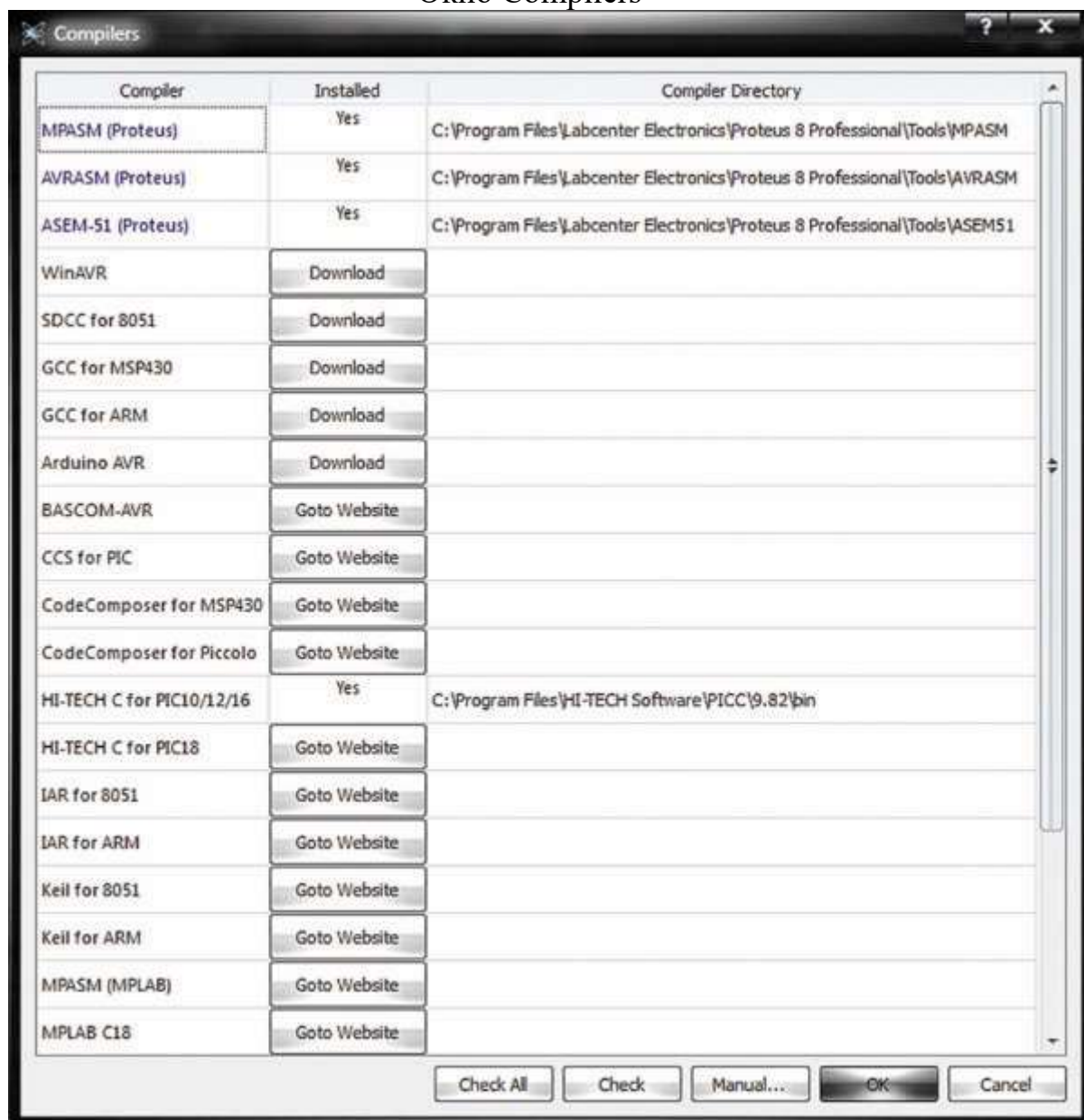


В случае выбора второго параметра доступны следующие поля:

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

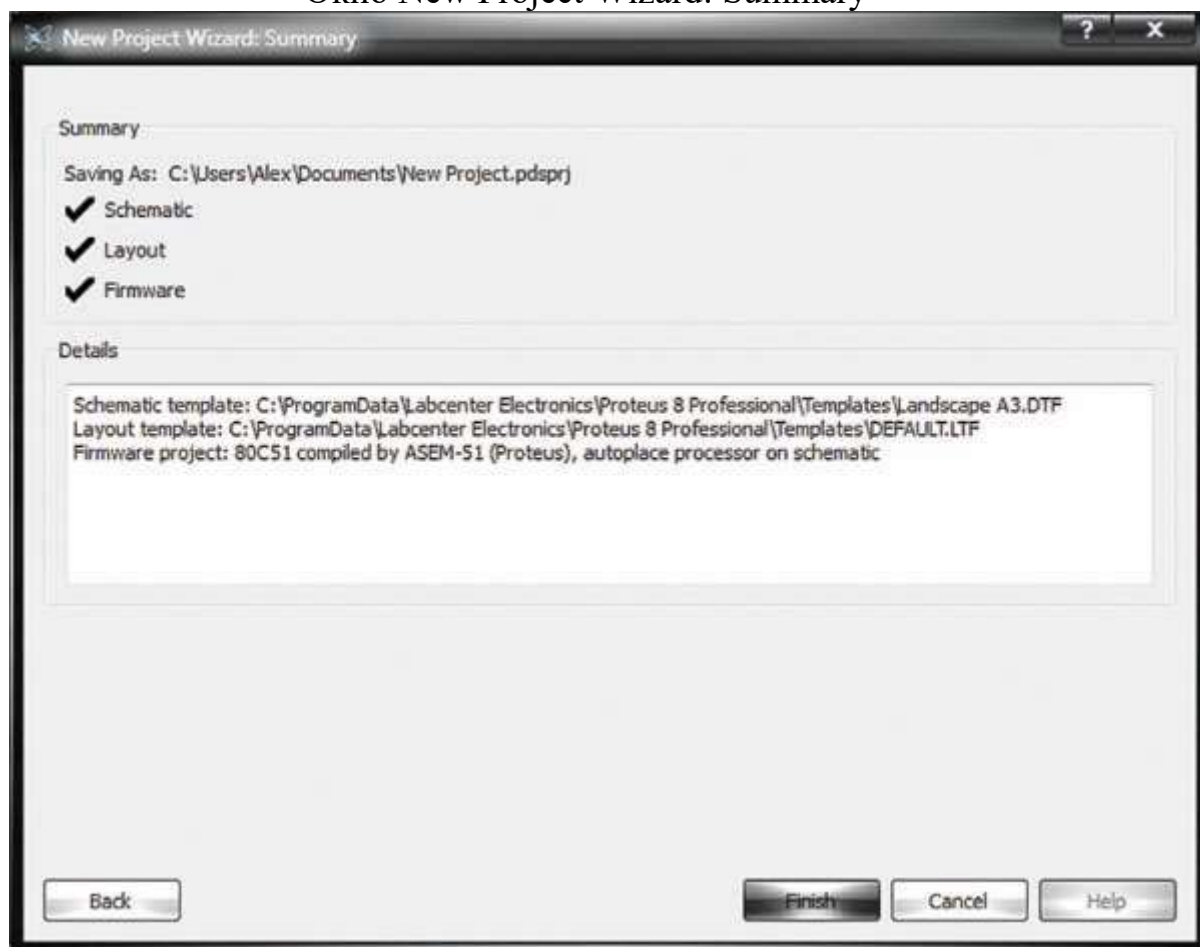
Используйте кнопку **Compilers** в поле **Compiler** для добавления компилятора (рисунок 9).

Окно Compilers



Шестой шаг работы мастера — подведение итогов. В окне **New Project Wizard: Summary** (рисунок 10) выводится вся информация о создаваемом проекте. Просмотрите внимательно настройки проекта и, если вас все устраивает, нажмите кнопку **Finish** для окончания работы с мастером. В противном случае вернитесь при помощи кнопки **Back** к предыдущим шагам мастера для внесения изменений в настройки проекта.

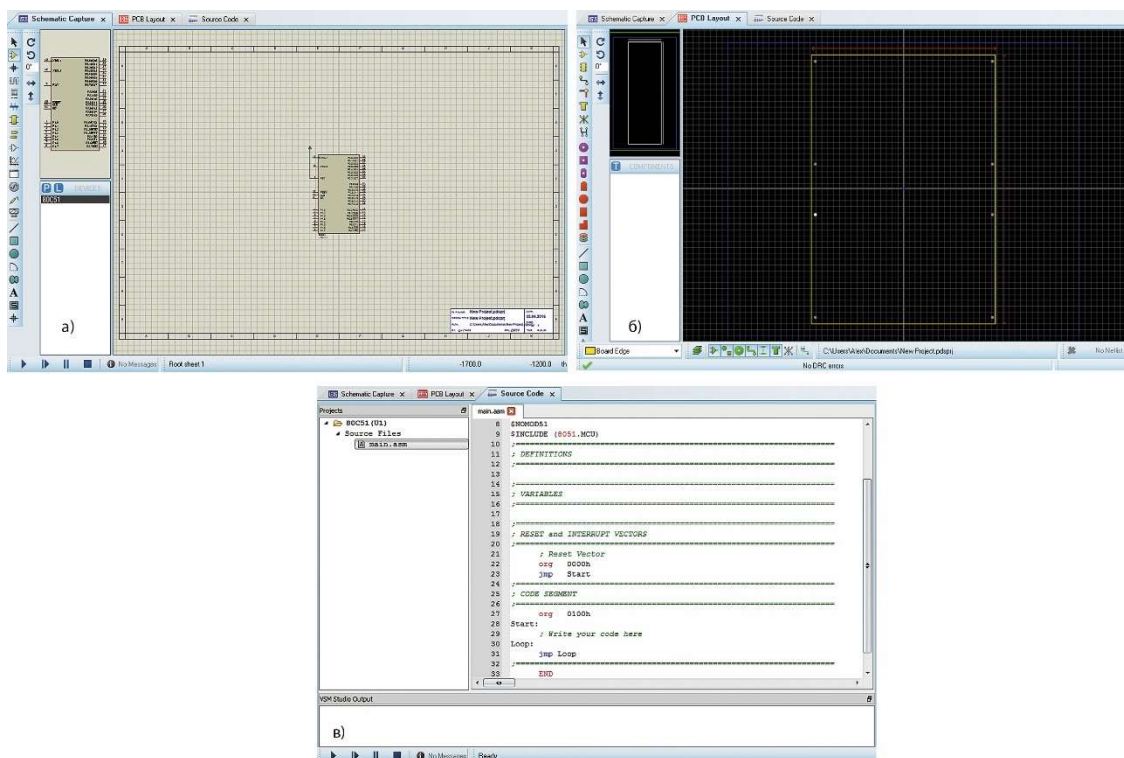
Окно New Project Wizard: Summary



По окончании работы мастера система на основе заданных установок создаст новый проект, который может содержать:

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

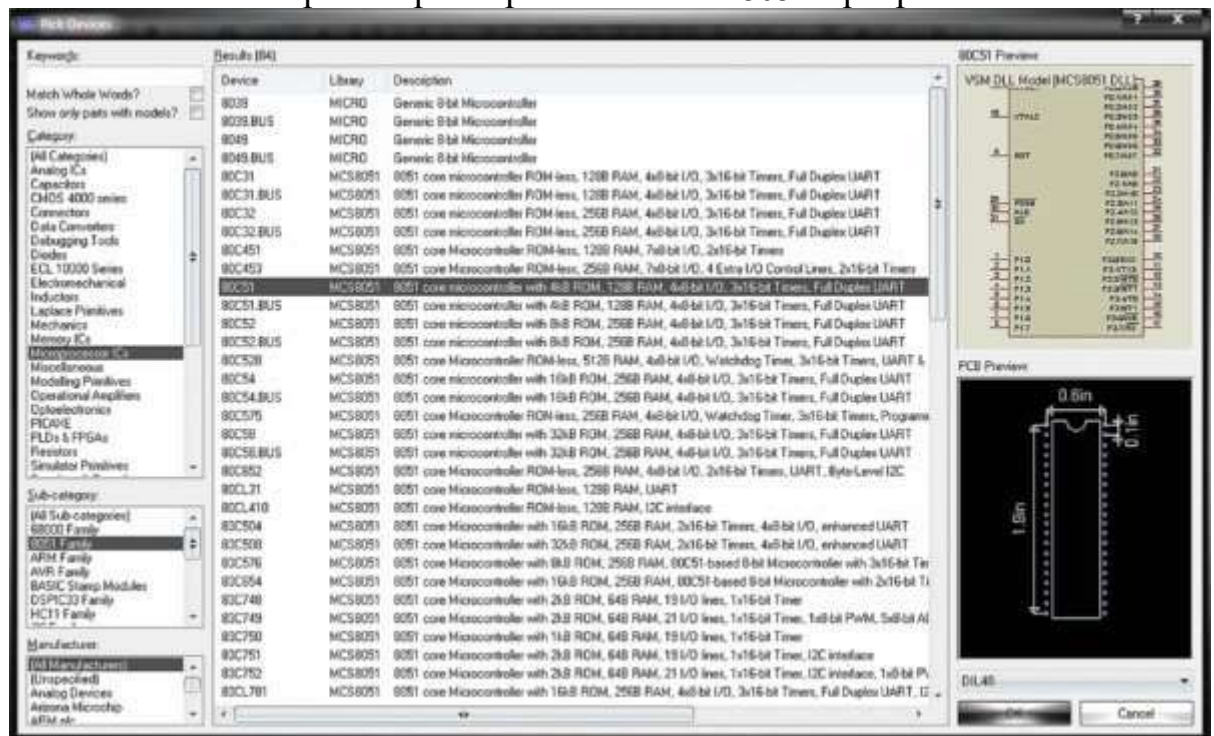
Проект схемы электрической принципиальной, в котором присутствует микроконтроллер, можно создать и без использования мастера — при помощи кнопки **ISIS** верхней панели инструментов Proteus. В результате откроется новая вкладка **Schematic Capture**, в рабочем поле которой и будет выполняться разработка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

панель можно нажатием кнопки **Component Mode** на левой панели инструментов редактора ISIS.

Рисунок 12

Библиотека микроконтроллеров семейства 8051 программы Proteus



Для того чтобы добавить микросхему микроконтроллера в рабочее поле проекта, необходимо в левой верхней части окна **Pick Devices** в поле **Category** выбрать из списка библиотеку **Microprocessor ICs**. Пакет **Microprocessor ICs** позволяет включать в эмуляцию смешанной схемы определенные микроконтроллеры с возможностью написания и отладки программного кода. Выбор библиотеки из списка производится посредством щелчка левой кнопки мыши по строке с ее названием. Ниже поля **Category** находится поле **Sub-category**, в котором таким же способом задается семейство микроконтроллеров выбранной библиотеки. В поле **Results** отображаются все компоненты выбранного семейства. Выбор компонента производится посредством выделения при помощи левой кнопки мыши строки с его названием в поле **Results**. В поле **Manufacturer** можно выбрать производителя микроконтроллера. Если производитель не имеет значения — укажите значение **All Manufacturers** в этом поле. Для ускорения поиска компонентов можно воспользоваться строкой фильтра **Keywords**, расположенной в верхнем левом углу окна **Pick Devices**. После того как выбор микроконтроллера сделан, его условное графическое обозначение отобразится в поле предварительного просмотра **Preview**. Посадочное место компонента будет показано в поле **PCB Preview**. Если для выбранного микроконтроллера предусмотрено несколько посадочных мест, то все возможные варианты будут доступны для выбора из выпадающего меню, расположенного под полем **PCB Preview**. Для того чтобы разместить выбранный микроконтроллер на схеме, необходимо в окне **Pick Devices** нажать кнопку **OK**. После чего данное окно

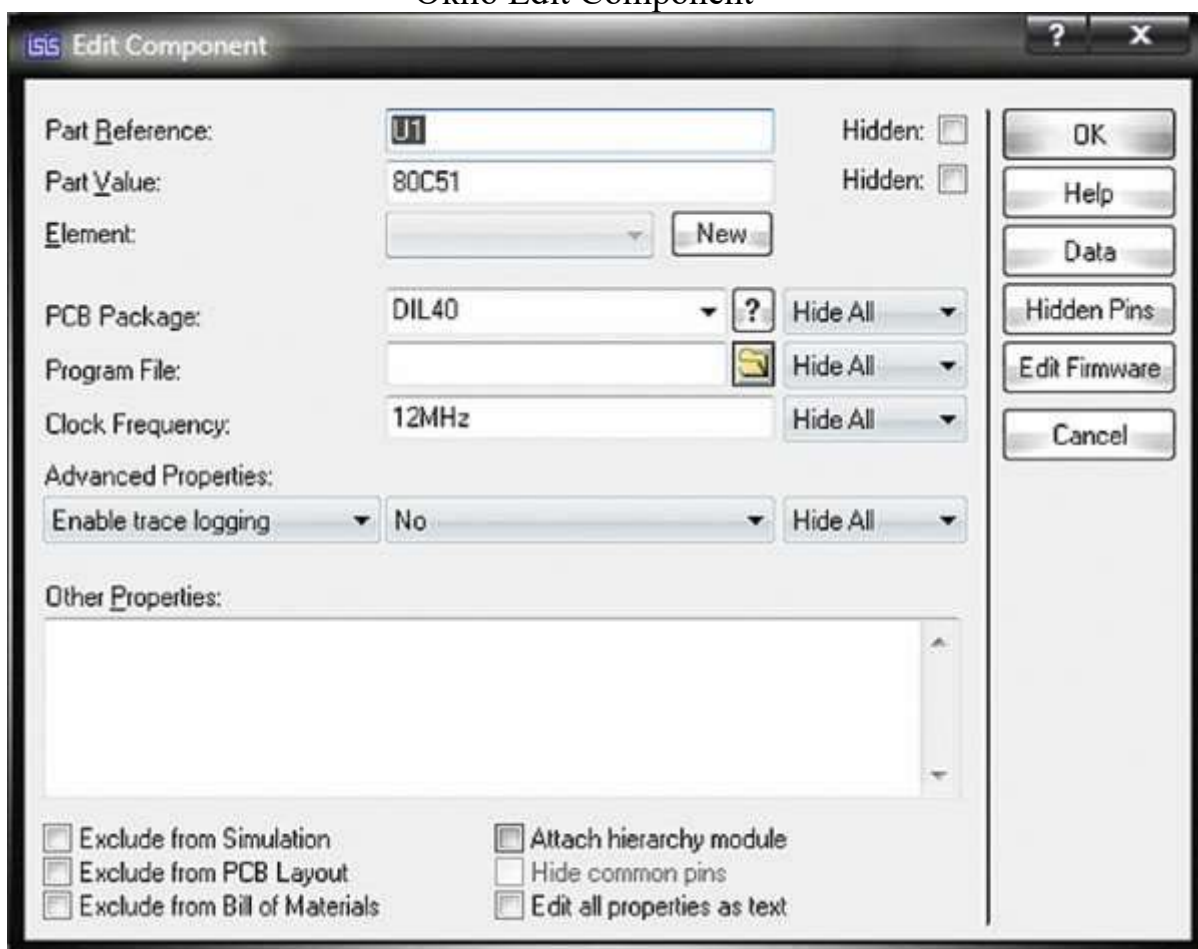
будет закрыто, а символ компонента будет прикреплен к курсору мыши, при помощи которого необходимо поместить символ в нужное место на схеме (щелкнуть в нужном месте схемы левой кнопкой мыши).

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



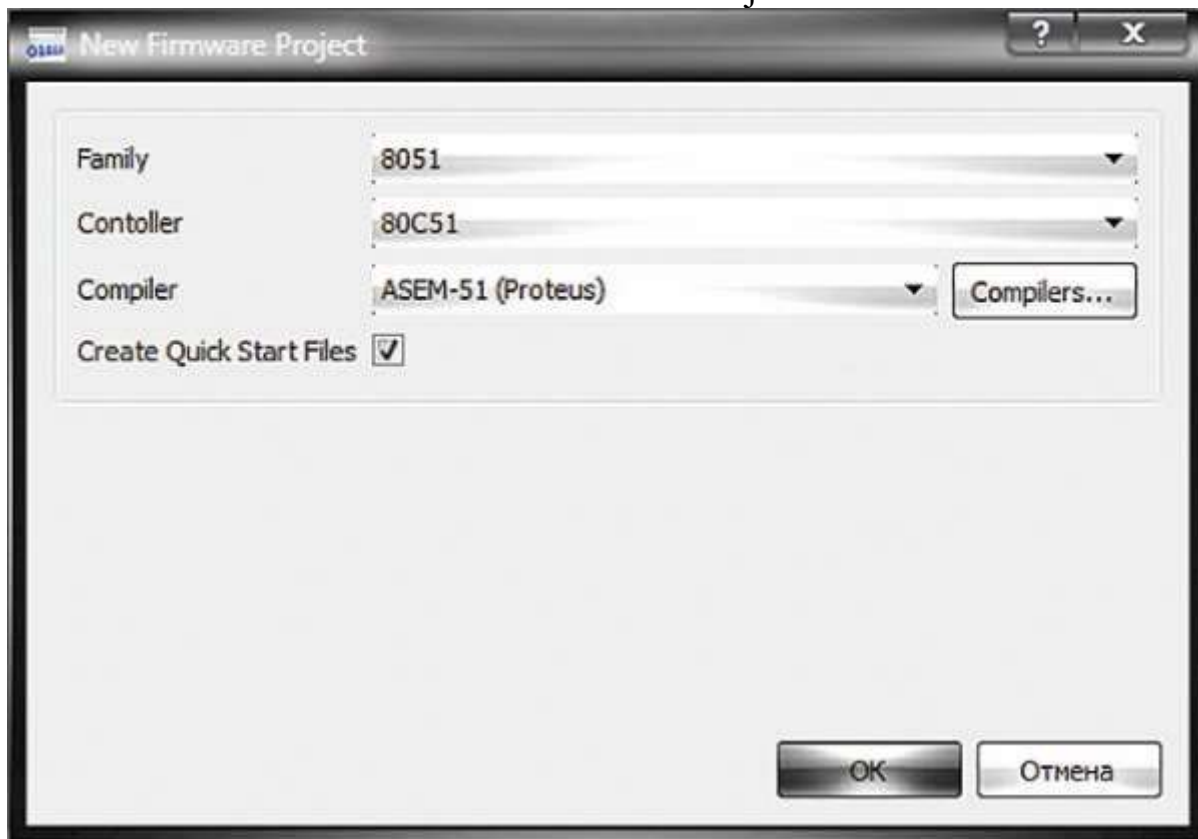
Открыть вкладку, на которой в процессе проектирования будет вноситься код программы инициализации микроконтроллера, можно следующим образом. Выделите при помощи левой кнопки мыши символ микроконтроллера в рабочем поле проекта, при помощи правой кнопки мыши вызовите контекстное меню и выберите в нем пункт **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

Окно New Firmware Project



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».

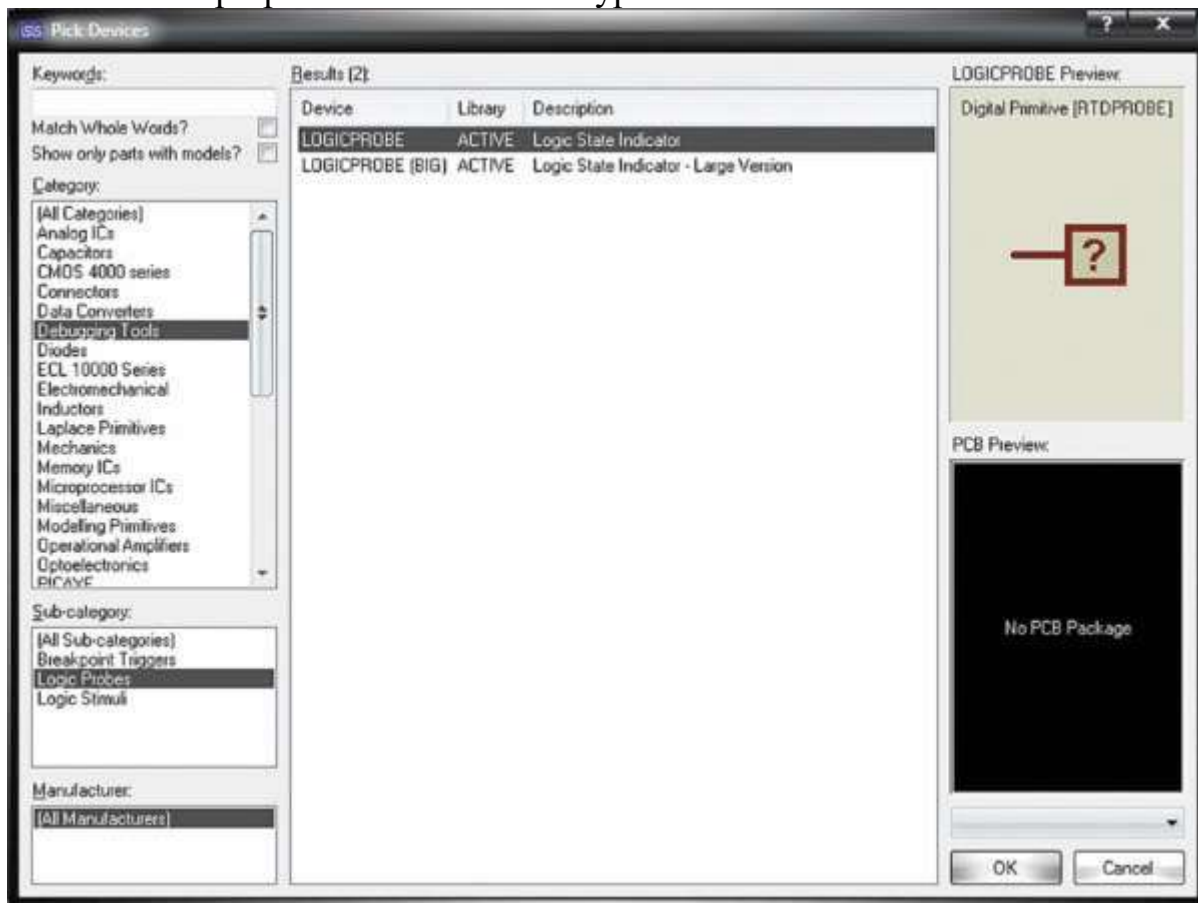
Моделирование схемы, включающей микроконтроллер, и трансляция программного кода

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 **OK**. 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

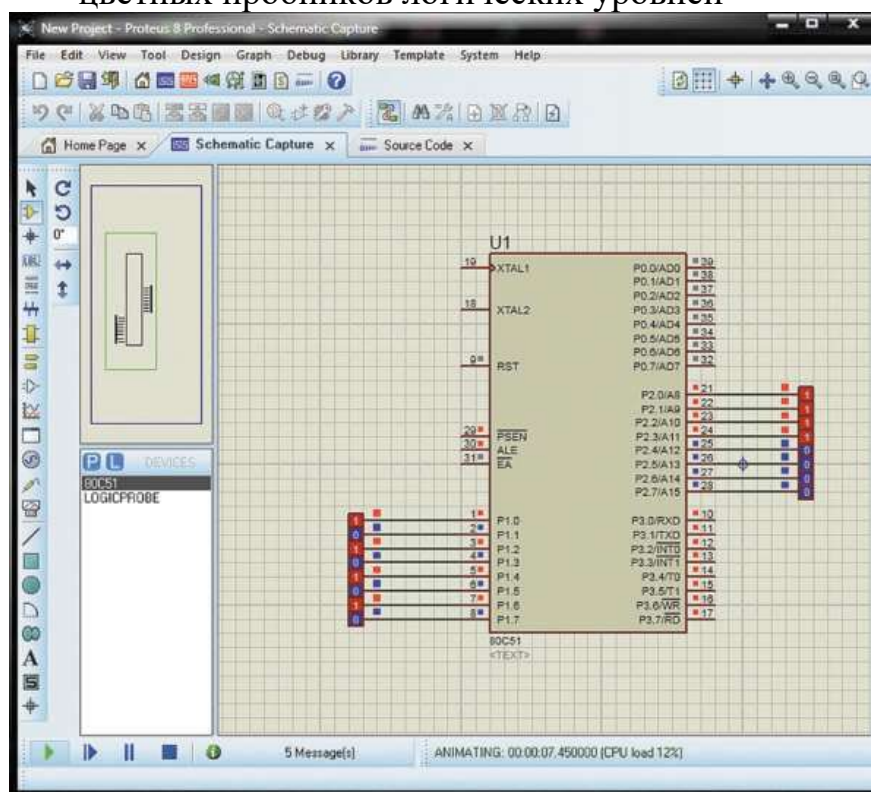


Рисунок 17

Программа инициализации микроконтроллера

The screenshot shows the Proteus 8 Professional Source Code window. The file 'main.asm' is open, showing the assembly code for the microcontroller. The code includes a reset vector and a loop that initializes the microcontroller. The status bar at the bottom indicates 'VSM Studio Output: Processed 162 lines. Compiled successfully.'

```

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 P2, R5
26 mov R4, #01010101b
27 mov P1, R4
28 Loop:
29 jmp Loop
30
31 ;-----
32 END
33

```

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

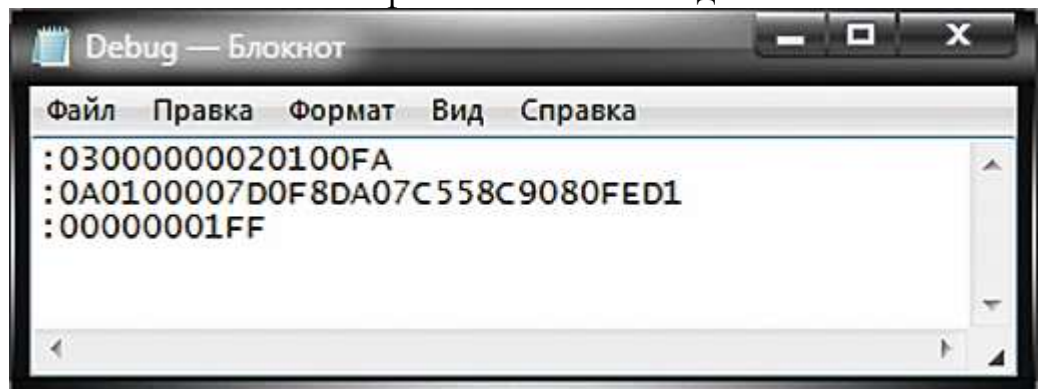
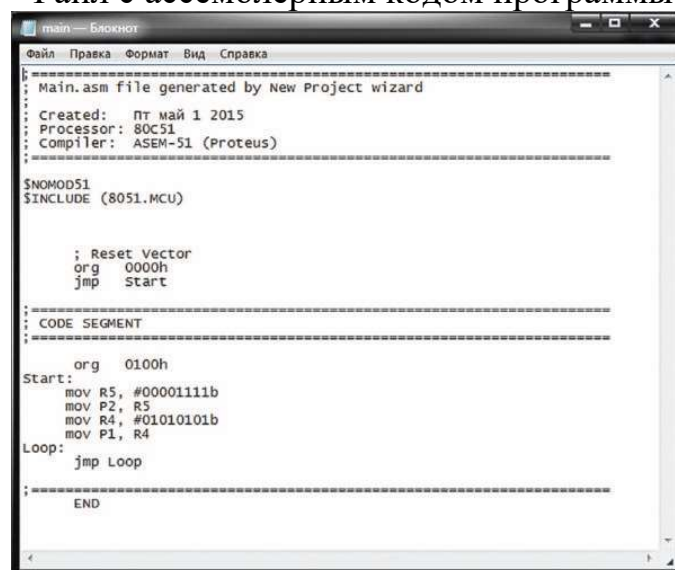


Рисунок 19

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 -----
91 -----
92 -----
93 -----
94 -----
95 -----
96 -----
97 -----
98 -----
99 -----
100 -----
101 -----
102 -----
103 -----
104 -----
105 -----

; Main.asm file generated by New Project wizard
; Created: 17 май 1 2015
; Processor: 80C51
; Compiler: ASM51 (Proteus)

$NOMOD51
$INCLUDE (8051.MCU)
; 8051 processor definition file

P0 DATA 080H
P1 DATA 081H
P2 DATA 082H
P3 DATA 083H
P4 DATA 084H
P5 DATA 085H
P6 DATA 086H
P7 DATA 087H
P8 DATA 088H
P9 DATA 089H
P10 DATA 08AH
P11 DATA 08BH
P12 DATA 08CH
P13 DATA 08DH
P14 DATA 08EH
P15 DATA 08FH
P16 DATA 090H
P17 DATA 091H
P18 DATA 092H
P19 DATA 093H
P20 DATA 094H
P21 DATA 095H
P22 DATA 096H
P23 DATA 097H
P24 DATA 098H
P25 DATA 099H
P26 DATA 09AH
P27 DATA 09BH
P28 DATA 09CH
P29 DATA 09DH
P30 DATA 09EH
P31 DATA 09FH
P32 DATA 0A0H
P33 DATA 0A1H
P34 DATA 0A2H
P35 DATA 0A3H
P36 DATA 0A4H
P37 DATA 0A5H
P38 DATA 0A6H
P39 DATA 0A7H
P40 DATA 0A8H
P41 DATA 0A9H
P42 DATA 0AAH
P43 DATA 0ABH
P44 DATA 0ACH
P45 DATA 0ADH
P46 DATA 0AEH
P47 DATA 0AFH
P48 DATA 0B0H
P49 DATA 0B1H
P50 DATA 0B2H
P51 DATA 0B3H
P52 DATA 0B4H
P53 DATA 0B5H
P54 DATA 0B6H
P55 DATA 0B7H
P56 DATA 0B8H
P57 DATA 0B9H
P58 DATA 0BAH
P59 DATA 0BBH
P60 DATA 0BCH
P61 DATA 0BDH
P62 DATA 0BEH
P63 DATA 0BFH
P64 DATA 0C0H
P65 DATA 0C1H
P66 DATA 0C2H
P67 DATA 0C3H
P68 DATA 0C4H
P69 DATA 0C5H
P70 DATA 0C6H
P71 DATA 0C7H
P72 DATA 0C8H
P73 DATA 0C9H
P74 DATA 0CAH
P75 DATA 0CBH
P76 DATA 0CBH
P77 DATA 0CBH
P78 DATA 0CBH
P79 DATA 0CBH
P80 DATA 0CBH
P81 DATA 0CBH
P82 DATA 0CBH
P83 DATA 0CBH
P84 DATA 0CBH
P85 DATA 0CBH
P86 DATA 0CBH
P87 DATA 0CBH
P88 DATA 0CBH
P89 DATA 0CBH
P90 DATA 0CBH
P91 DATA 0CBH
P92 DATA 0CBH
P93 DATA 0CBH
P94 DATA 0CBH
P95 DATA 0CBH
P96 DATA 0CBH
P97 DATA 0CBH
P98 DATA 0CBH
P99 DATA 0CBH
P100 DATA 0CBH

; Reset Vector
org 0000h
jmp Start

; CODE SEGMENT
org 0100h
Start:
mov R5, #00001111b
mov R4, #01010101b
mov R3, R4
Loop:
jmp Loop

```

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

```

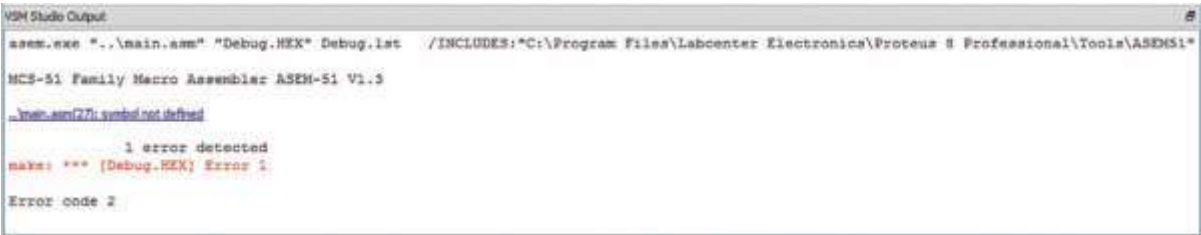
74: 1      N      D4      RS1      BIT      0D4H
75: 1      N      D5      F0       BIT      0D5H
76: 1      N      D6      AC       BIT      0D6H
77: 1      N      D7      CY       BIT      0D7H
78: 1
79: 1      N      0000      RESET    CODE    000H
80: 1      N      0003      EXTI0    CODE    003H
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



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).

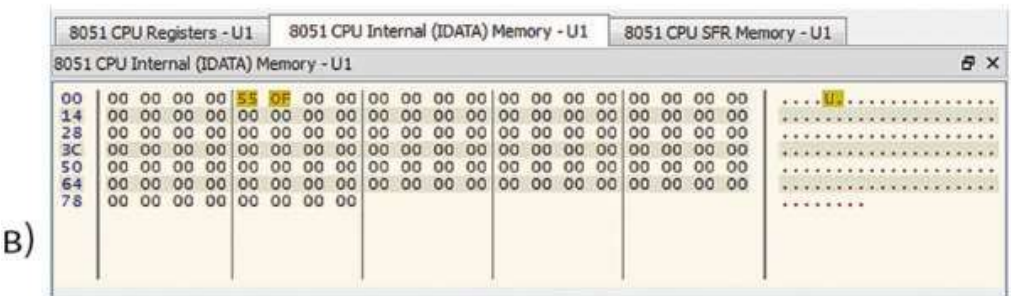
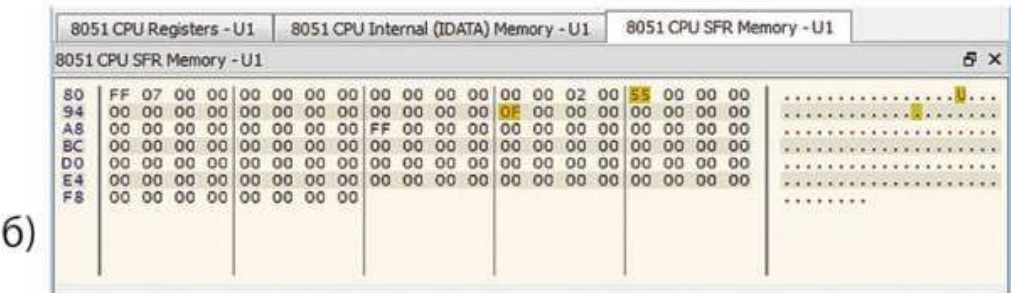
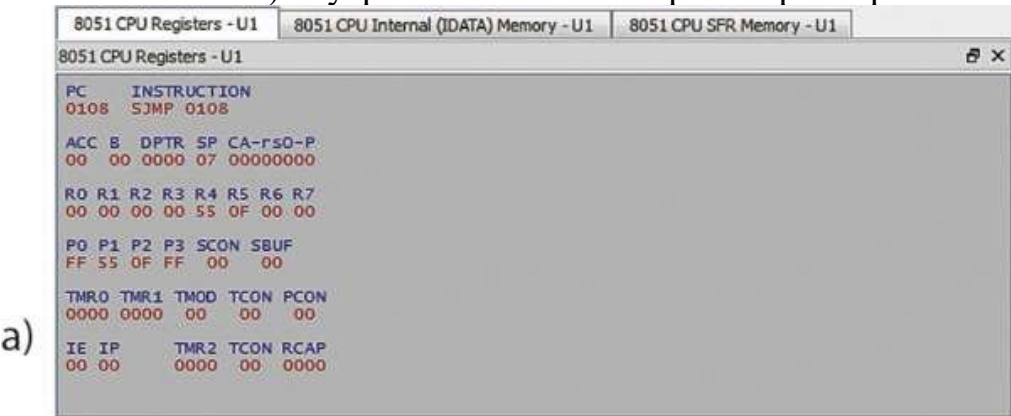
Рисунок 23

Состояние:

а) регистров;

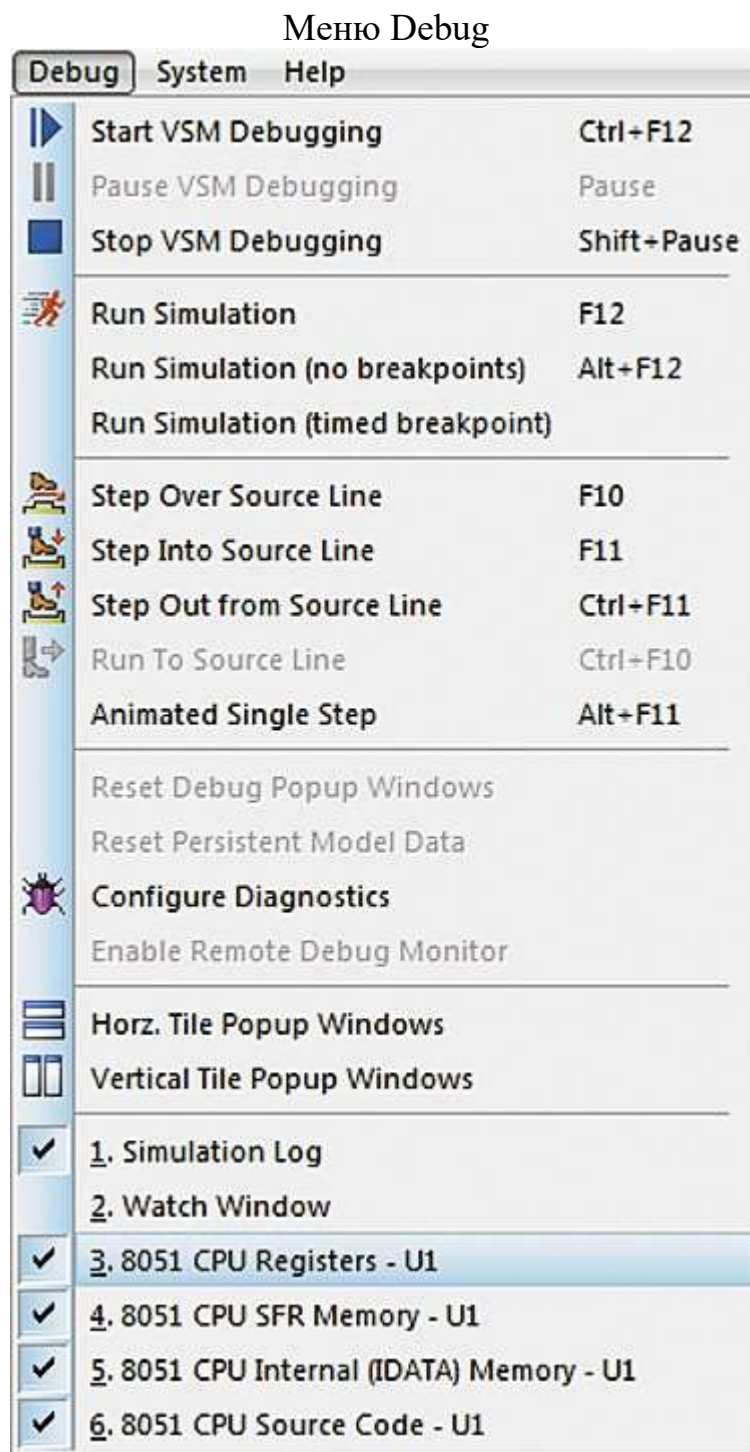
б) памяти SFR;

в) внутренней памяти микроконтроллера



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.

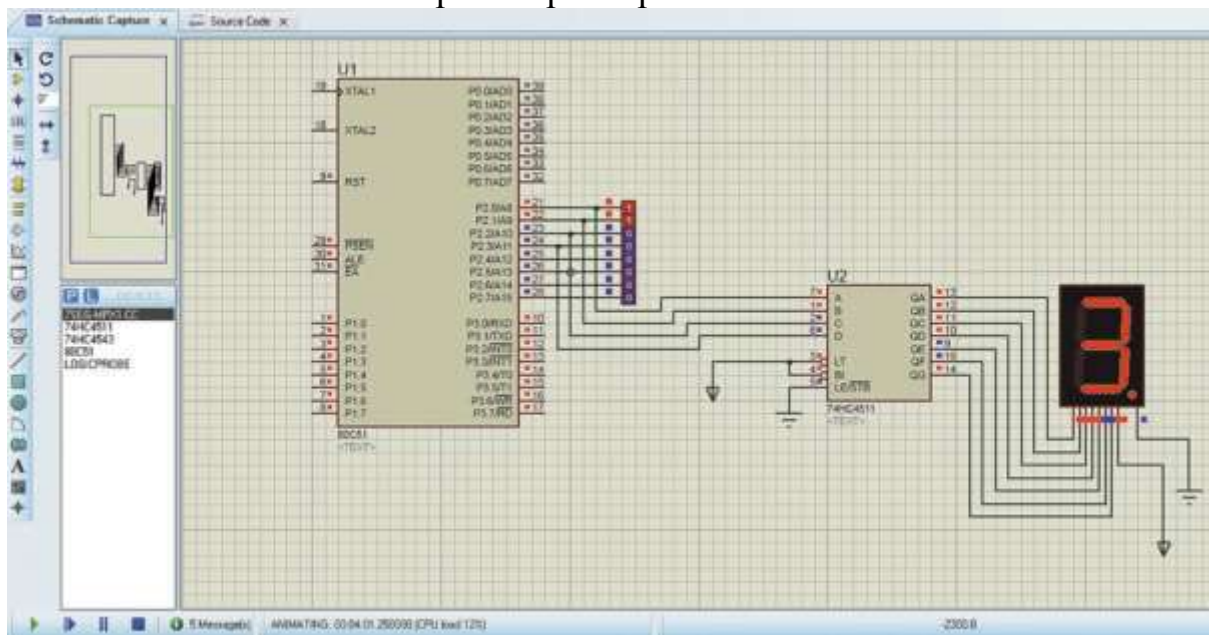
Сопряжение микроконтроллера x51 с микросхемами 7-сегментных индикаторов в Proteus

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 знакосинтезирующие 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 COMPIМ 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.