

УДК 681.32

Изучение программирования микроконтроллеров в САПР Proteus

С. Ф. Тюрин

Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
tyurinsergfe@yandex.ru; +7-952-32-02-510
Пермский национальный исследовательский политехнический университет
Россия, 614990, г. Пермь, ул. Комсомольский пр., 29

Д. А. Ковыляев

Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
354981@mail.ru; +7-919-458-80-79

Е. Ю. Данилова

Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
ket-eref@yandex.ru; +7-919-458-80-79

А. Ю. Городилов

Пермский государственный национальный исследовательский университет
Россия, 614990, г. Пермь, ул. Букирева, 15
gora830@yandex.ru; +7-902-83-07-328

Рассматривается создание проектов в программе Proteus на основе микроконтроллеров. Исследуется классический микроконтроллер 8051 фирмы Intel, а также микроконтроллер STM32F401RE фирмы изготовителя STMicroelectronics (архитектура ядра – компании ARM). Выполняется разработка простейшей программ для использования на лабораторных занятиях по программированию встроенных систем (например, "интернета вещей") для 8051 – на языке Ассемблер, для STM32F401RE – на языке СИ с использованием специальной среды разработки. Исследование может быть использовано на лабораторных занятиях по программированию встроенных систем.

Ключевые слова: микроконтроллер; программа; IDE.

DOI: 10.17072/1993-0550-2021-2-69-74

Введение

Микроконтроллеры [1–3], ранее именуемые "Микро-ЭВМ", широко применяются в так называемых встроенных системах [4–6], например, в "интернете вещей", в сервисных и промышленных роботах. Микроконтроллеры содержат в одной микросхеме, помимо собственно микропроцессора (процессора), постоянную и оперативную память, ЦАП, АЦП, устройства ввода-вывода, приемо-передатчики, различные преобразователи и др.

Современные 32-разрядные микроконтроллеры обладают большими функциональными возможностями.

В продаже имеется широкий выбор относительно не дорогих готовых плат с микроконтроллерами, содержащими дополнительные микросхемы-загрузчики для подключения к внешнему компьютеру, продаются также различные датчики, твердотельные реле и другое оборудование для изготовления полностью собственной встроенной системы. При программировании современных микроконтроллеров используются IDE (Integrated Development Environment), например, Keil uVision.

Компетенции в области программирования микроконтроллеров весьма востребованы на IT рынке. В нашем университете подобной тематикой занимаются, например, на кафедре радиоэлектроники и защиты информации физического факультета, в ПНИПУ – на кафедре автоматики и телемеханики электротехнического факультета.

1. Исследование микроконтроллера 8051

Микроконтроллер 8051 является "классикой", он разработан в 1980-е годы XX в., но еще до сих пор его архитектура востребована, например, в контроллерах бортовых систем [7]. Для моделирования загружается соответствующая свободно распространяемая система схемотехнического моделирования на сайте фирмы Labcenter Electronics (Великобритания) [8]: (рис. 1).

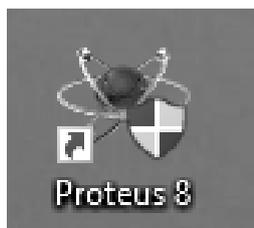


Рис. 1. Символ Proteus Design Suite версия 8

После выполнения необходимых действий по выбору микроконтроллера он появляется на наборном поле (рис. 2).

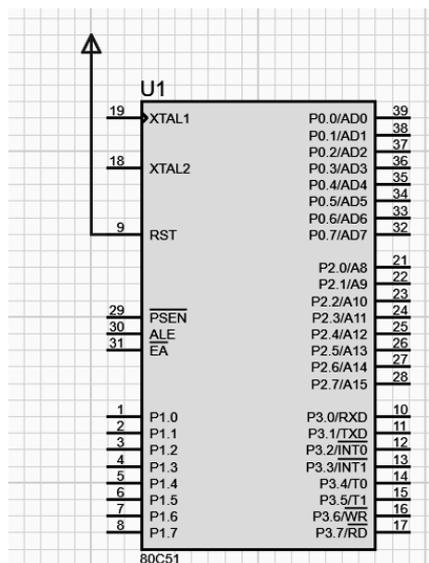


Рис. 2. Микроконтроллер 80C51(8051)

Мы видим обозначения портов ввода-вывода P0,P1,P2, вводы подключения кварце-

вого резонатора XTAL1, XTAL2 для обеспечения работы внутреннего тактового генератора, управляющие входы-выходы, например, сброс RST. Кроме того, автоматически средствами Proteus формируется заготовка программы на языке Ассемблер (при выборе микроконтроллера задаем формирование "прошивки") (рис. 3):

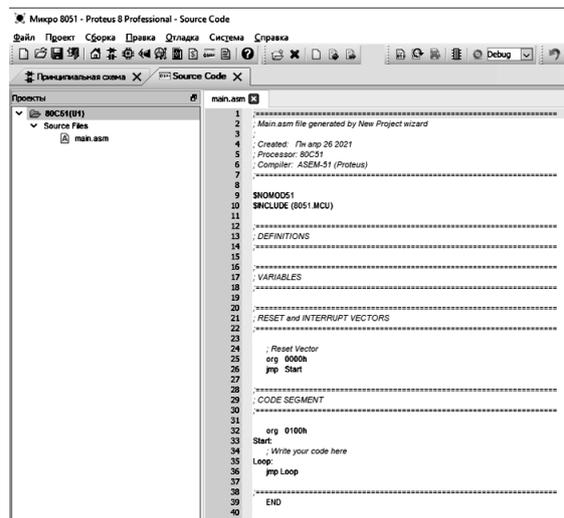


Рис. 3. Заготовка программы на языке Ассемблер

Подключаем необходимый минимум дополнительных элементов, используем ключ для имитации некоторого датчика и светодиод для индикации его состояния (рис. 4):

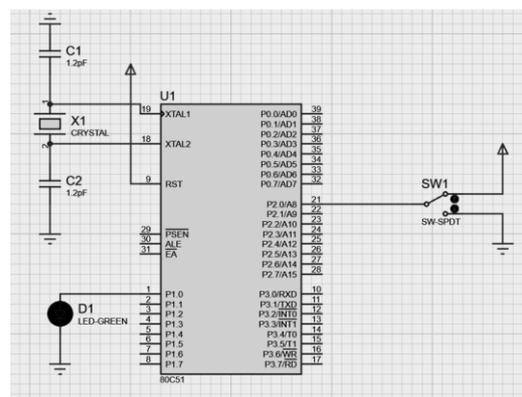


Рис. 4. Подключение кварцевого резонатора, ключа и светодиода

Подготовим программу на языке Ассемблер для ввода информации с порта P2 и вывода на порт P1 (без детализации номера ввода, предполагается, что вводится байт и выводится байт). Это всего одна команда пересылки: MOV P1, P2. Вставляем, компилируем (рис. 5):

```

-----
org 0100h
0100 Start:
-----
0100 Loop:
0100 MOV P1, P2; Write your code here
-----
0103 jmp Loop
-----

```

5 Message(s) PAUSED: 0.000001000s

Рис. 5. Успешная компиляция

Запускаем моделирование, состояние светодиода повторяет вводимый ключом логический уровень:

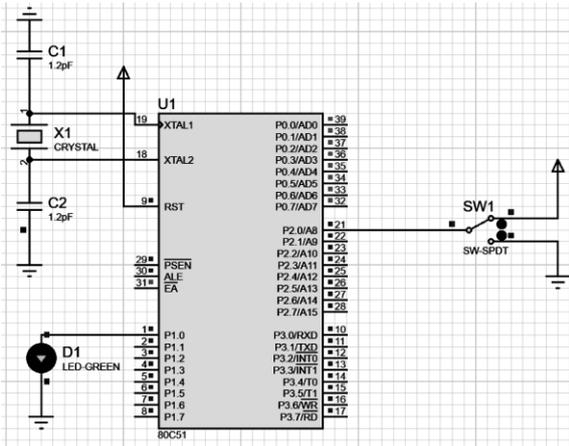


Рис. 6. Проверка работы программы

В качестве дополнительного задания можно увеличить число ключей и светодиодов, довести их до восьми. Очевидно, что программа не изменится. Можно также менять источник и получатель сигнала и изменять программу. Можно ввести обработку входного сигнала.

Разработаем более сложную, универсальную программу – реализации конечного автомата методом Гутмана [9] (рис. 6).

```

-----
; Reset Vector
org 0000h
-----
CODE SEGMENT
-----
org 0100h
0100 Start:
0101 Loop:
0102 Begin: MOV A, P1 ; Считываем и заносим в аккумулятор A
0103 ANL A, #0Bh ; Маскируем ; в R1
0104 MOV R1, A ; в R1
0105 MOV A, R3 ; Внутреннее состояние в R3
0106 ORL A, R1 ; Получаем полное входное слово
0107 MOV R3, A ; Полное входное слово в R3
0108 MOV R2, #0 ; Обнуляем R2 в нем выход
0109 MOV DPTR, #TABL ; Указатель на массив констант
0110 CLR A ; Очищаем аккумулятор
0111 MOV A, #ANDPTR ; Загружаем X0(1) в A
0112 MOV R0, A ; X0(1) в R0
0113 next: ANL A, R1 ; Маскируем входное слово X0(1)
0114 MOV R0, A ; результат в R0
0115 CLR A ; Очищаем аккумулятор
0116 INC DPTR ; Инкрементируем регистр указателя
0117 MOV A, #ANDPTR ; Загружаем X0(1) в A
0118 ORL A, R0 ; Сложение по модулю 2 с предыдущим
0119 INC DPTR ; Инкремент регистра указателя
0120 JNC Check ; Переход если ноль
0121 CLR A ; Очищаем аккумулятор
0122 MOV A, #ANDPTR ; Загружаем Z(1)
0123 ORL A, R2 ; Получаем выходное слово
0124 MOV R2, A ; Сохраняем результат в R2
0125 Check: CLR A ; Очищаем аккумулятор
0126 INC DPTR ; Инкрементируем регистр указателя
0127 MOV A, #ANDPTR ; Загружаем X0(1+1)
0128 ORL A, R2 ; Получаем X0(1+1) в R2
0129 JNC Next ; Переход по нулю
0130 MOV A, R2 ; Выход в R2
0131 ANL A, #00Ch ; Выделяем y2(z+1)/y1(z+1)
0132 MOV R2, R2 ; результат в R3
0133 MOV P2, R2 ; выход на P2
0134 jmp Begin
TABL: DB 00FH,03FH,04FH, 00FH,02H,02H, 00FH,04H,06H, 00FH,05H,04H,00FH,07H,00CH,00FH,00DH,00EH, 00FH,
-----
0145 jmp Loop
-----

```

Рис. 6. Программа реализации конечного автомата

Для проверки программы усложняем схему (рис. 7):

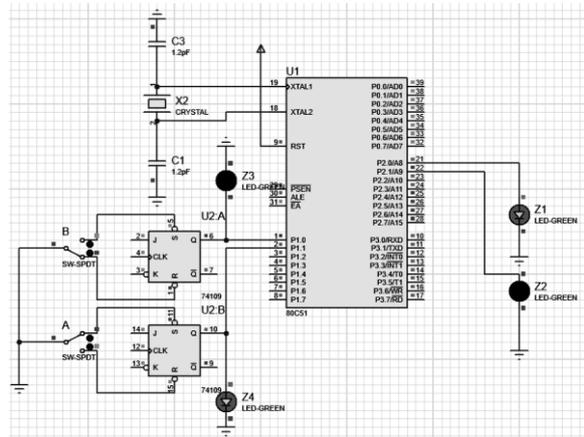


Рис. 7. Проверка работы программа реализации конечного автомата с двумя входами ab и двумя выходами z1z2

Таким образом рассмотренные примеры позволяют ознакомиться с классическим микроконтроллером и соответствующим языком Ассемблер.

Программа может загружаться также в виде шестнадцатеричного файла, такой пример рассмотрен далее.

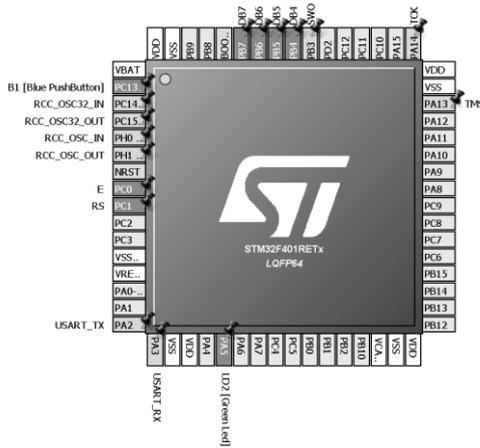
2. Исследование ввода-вывода в микроконтроллере STM32F401RE

Исследуем более сложный микроконтроллер STM32F401RE [10] (рис. 8).

Создание проекта, схемы и программирование здесь гораздо более сложное, чем для 8051. Используется специальное средство STM32CubeMX [11].

Для него, в свою очередь, необходима Java Runtime Environment (версия jre-8u-271-windows-x64) [12].

Необходима также интегрированная среда разработки IDE (Integrated development environment) Keil uVision, входящая в состав MDK (Microcontroller Development Kit) – ARM (Advanced RISC Machine – усовершенствованная RISC-машина) исследуемого микроконтроллера [13].



a)

Peripherals	STM32F401RE	
SRAM in Kbytes	96	
Flash memory in Kbytes	512	
Timers	General-purpose	7
	Advanced-control	1
USART	3	
USB OTG FS	1	
GPIOs	50	
12-bit ADC Number of channels	16	
Maximum CPU frequency	84 MHz	
Operating voltage	1.7 to 3.6 V	

b)
Рис. 8. Микромикроконтроллер STM32F401RE:
a) входы-выходы ("распиновка") микросхемы;
б) характеристики STM32F401RE

После загрузки необходимого ПО запускаем STM32CubeMX, создаем проект и настраиваем микросхему на ввод с PA0, PA1 и вывод на PC0, PC1. По нажатию GENERATE CODE завершается настройка в CubeMX. Нажимаем кнопку Open Project в CubeMX, открывается uVision IDE. Записываем программу ввода-вывода (рис. 9):

```

main.c
87  MX_GPIO_Init();
88  /* USER CODE BEGIN 2 */
89
90  /* USER CODE END 2 */
91
92  /* Infinite loop */
93  /* USER CODE BEGIN WHILE */
94  while (1)
95  {
96    /* USER CODE END WHILE */
97    HAL_GPIO_WritePin(MY_OUT0_GPIO_Port, MY_OUT0_Pin,
98    HAL_GPIO_ReadPin(MY_IN0_GPIO_Port, MY_IN0_Pin));
99    HAL_GPIO_WritePin(MY_OUT1_GPIO_Port, MY_OUT1_Pin);
100   HAL_GPIO_ReadPin(MY_IN1_GPIO_Port, MY_IN1_Pin));
101   HAL_Delay(25);
102
103   /* USER CODE BEGIN 3 */
104   }
105   /* USER CODE END 3 */
106 }
107
108 /**
109  * @brief System Clock Configuration
110  * @retval None
111  */
    
```

Рис. 9. Программа ввода-вывода в STM32F401RE

Создаем схему в Proteus, загружаем программу в виде HEX файла в микроконтроллер (рис. 10):

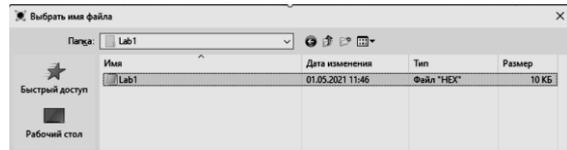


Рис. 10. Автоматически сформированный HEX файл в проекте LAB1

Проверяем работу в Proteus (рис. 11):

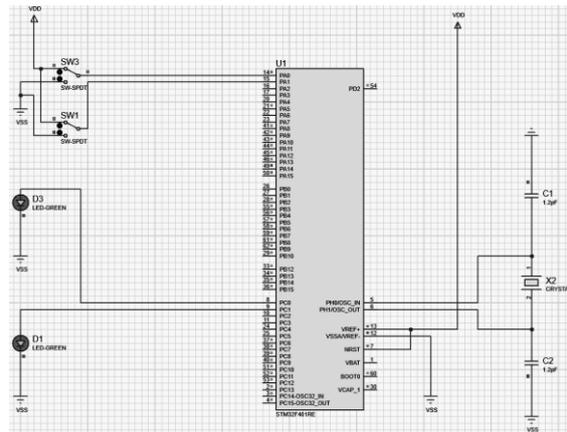


Рис. 11. Схема STM32F401RE для ввода-вывода двух сигналов

Все работает правильно! Пример разработки более сложной программы – реализации конечного автомата в плате NUCLEO-F401RE с микроконтроллером STM32F401RET6 с использованием ресурса Mbed [14] рассмотрен в [15].

3. Реализация эволюционных вычислений в микроконтроллере STM32F401RE

Программа создана для обеспечения встроенного тестирования логики ПЛИС. Она написана в среде Visual Studio 2019 на языке C++. Поэтому осуществлен перенос исходного кода в новую программу в среде uVision IDE. uVision IDE содержит возможность компилировать проекты из исходного кода, написанного на языке C++.

В проекте STM32CubeMX необходимо указать минимальный размер стека 8192 байта и минимальный размер динамической памяти 86016 байт, что граничит с максимальными 96 КБ памяти.

Для выдачи результатов вычислений использован протокол USART. В Proteus создается схема с приемником (рис. 12):

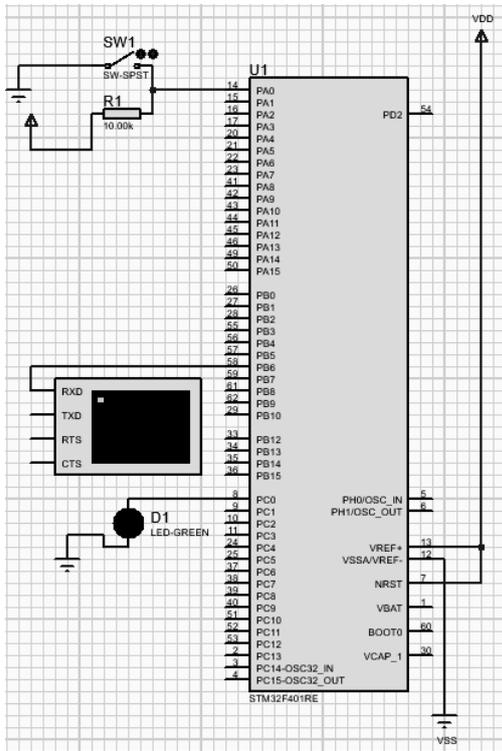


Рис. 12. Схема STM32F401RE с приемником для вывода результатов эволюционных вычислений

Пример выдачи результатов поиска оптимального теста показан на рис. 13.

```

Virtual Terminal
GA: try 1
Seed = 54676824
Ind. Cnt = 3
Ind. Cnt = 4
Solution found!!!!
1
4
5
7

GA: try 2
Seed = 787592950
Ind. Cnt = 3
Ind. Cnt = 4
Solution found!!!!
1
4
5
7
    
```

Рис. 13. Результат расчетов

Выводы

Таким образом, в случае загрузки необходимых программ до начала двухчасового занятия, в его рамках возможно выполнить рассмотренные исследования по одному из контроллеров. Более сложные программы рекомендуются к выполнению в рамках самостоятельной работы студентов (магистров). С целью обеспечения таких занятий методической литературой, авторами подготовлено к изданию учебное пособие. В дальнейшем целесообразно исследовать более сложные вопросы использования рассмотренного микроконтроллера STM32F401RE или платы NUCLEO-F401RE для моделирования встроенного диагностирования логики ПЛИС с помощью эволюционных алгоритмов на базе заданной модели отказов (неисправностей) с учетом количества элементов, необходимости приемов конфигурационной информации и т.д.

Список литературы

1. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. СПб.: БХВ-Петербург, 2010. 797 с.
2. Тюрин С.Ф. Вычислительная техника и информационные технологии. Руководство к лабораторным работам в системе Proteus 7.2. Пермь: Изд-во Перм. гос. техн. ун-та, 2010. 135 с.
3. Вычислительная техника и информационные технологии. Аппаратные средства вычислительной техники: конспект лекций / С.Ф. Тюрин, О.В. Гончаровский, О.А. Громов. Пермь: Изд-во Перм. гос. техн. ун-та, 2011. 324с.
4. Гончаровский О.В. Прототипирование сетевой системы управления. Разработка Windows-приложения удаленного контроллера прототипа работа-официанта на базе PROMOBOT Vol. 4. Пермь: Изд-во ПНИПУ, 2019. URL: <https://elibr.pstu.ru/docview/?fDocumentId=4527> (дата обращения: 11.05.2021).
5. Гончаровский О.В., Каменских А.Н. Встроенные микропроцессорные системы. Макетирование систем управления технических систем: учеб.-метод. пособие. Пермь: Изд-во Перм. нац. исслед. политехн. ун-та, 2020. 146 с.
6. Гончаровский О.В. Проектирование встроенных управляющих систем реального времени. Пермь: Изд-во ПНИПУ, 2013. URL:

- <https://elibr.pstu.ru/docview/?fDocumentId=351> (дата обращения: 10.05.2021).
7. АО "Дизайн Центр "Союз". URL: <https://dcsoyuz.ru/products/mikrokontrollery> (дата обращения: 27.04.2021).
 8. *Proteus Downloads* URL: <https://www.labcenter.com/downloads/> (дата обращения: 23.04.2021).
 9. Тюрин С.Ф., Гончаровский О.В. Программная реализация конечного автомата на языке СИ // Вестник Пермского университета. Математика. Механика. Информатика. 2019. № 3 (46). С. 86–90.
 10. *Stm32f401ret6 reference manual*. URL: <https://www.st.com/resource/en/datasheet/stm32f401re.pdf> (дата обращения: 28.04.2021).
 11. *STM32 – STM32CubeMX*. URL: <https://www.st.com/en/development-tools/stm32cubeMX.html#> (дата обращения: 28.04.2021).
 12. *jre-8u-271-windows-x64*. URL: <https://www.java.com/ru/download/> (дата обращения: 28.04.2021).
 13. *MDK-ARM*. URL: <https://www.keil.com/demo/eval/arm.htm> (дата обращения: 28.04.2021).
 14. *Mbed*. URL: <https://www.mbed.com/en/> (дата обращения: 28.04.2021).
 14. Гончаровский О.В., Тюрин С.Ф. Реализация конечного автомата в микроконтроллере с помощью ресурса *mbed.com*. // Информационно-измерительные и управляющие системы. 2019. Т. 17, № 4. С. 20–27.
 15. *Mbed*. URL: <https://www.mbed.com/en/> (дата обращения: 28.04.2021).

Learning programming of micro-controllers in CAD Proteus

S. F. Tyurin

Perm State University; 15, Bukireva st., Perm, 614990, Russia
Perm National Research Polytechnic University; 29, Komsomolsky Av., Perm, 614990, Russia
tyurinsergfeoyandex.ru; +7 952-320-02-510

D. A. Kovylyayev

Perm State University; 15, Bukireva st., Perm, 614990, Russia
354981@mail.ru; +7-919-458-80-79

E. Y. Danilova

Perm State University; 15, Bukireva st., Perm, 614990, Russia
ket-eref@yandex.ru; +7-919-458-80-79

A. Yu. Gorodilov

Perm State University; 15, Bukireva st., Perm, 614990, Russia
gora830@yandex.ru; +7-902-83-07-328

The creation of projects in the Proteus program based on microcontrollers is considered. A classic 8051 microcontroller from Intel is investigated, as well as an STM32F401RE microcontroller from ARM. The development of the simplest programs for use in laboratory classes on programming embedded systems (for example, the "Internet of things") for 8051 – in assembler language, for STM32F401RE – in C language using a special development environment is carried out. The research can be used in laboratory classes on embedded systems programming.

Keywords: *microcontroller; program; IDE.*