

ОБУЧЕНИЕ ОПТИМИЗАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ МИКРОКОНТРОЛЛЕРОВ: СРЕДА FLOWCODE И ПРАКТИЧЕСКИЕ ИНСТРУМЕНТЫ

Очилов Муроджон Ашуркулович

Доцент Каршинского инженерно-экономического института
ochilov22@mail.ru

Жураев Абурайхон Холикулович

Старший преподаватель Каршинского инженерно-экономического института
aburayxonjurayev75@gmail.com

АННОТАЦИЯ

В статье рассматриваются подходы к обучению оптимизации программного обеспечения для микроконтроллеров с использованием среды визуального программирования Flowcode. Представлены основные принципы работы с микроконтроллерами, методы минимизации использования памяти и повышения быстродействия программ. Освещаются практические аспекты, такие как управление реле, взаимодействие с датчиками (DHT и LDR) и использование ротарного энкодера. Особое внимание уделяется созданию энергоэффективных и функциональных систем на базе Flowcode. В заключение предлагается ряд практических проектов для закрепления знаний, включая интеграцию микроконтроллеров с IoT-платформами.

Ключевые слова: Микроконтроллеры, Flowcode, визуальное программирование, оптимизация программного обеспечения, управление реле, датчики DHT и LDR, ротарный энкодер, энергоэффективность, IoT, практические проекты.

ABSTRACT

The article discusses approaches to teaching software optimization for microcontrollers using the visual programming environment Flowcode. The basic principles of working with microcontrollers, methods for minimizing memory use and increasing program performance are presented. Practical aspects are covered such as relay control, interaction with sensors (DHT and LDR) and the use of a rotary encoder. Particular attention is paid to creating energy-efficient and functional systems based on Flowcode. In conclusion, a number of practical projects are proposed to consolidate knowledge, including the integration of microcontrollers with IoT platforms

Keywords: *Microcontrollers, Flowcode, Visual Programming, Software Optimization, Relay Control, DHT and LDR Sensors, Rotary Encoder, Energy Efficiency, IoT, Hands-On Projects.*

ВВЕДЕНИЕ

Компания Microchip предлагает стартовый набор разработчика PICkit 2 Debug Express, который включает демонстрационную плату с 44-выводным микроконтроллером и один из самых компактных программаторов в мире. Этот комплект разработан для того, чтобы предоставить пользователям все необходимые инструменты для изучения и освоения работы с микроконтроллерами серии PIC. Компактные размеры программатора делают его удобным для повседневного использования, а функциональность набора позволяет реализовывать как простейшие проекты, так и более сложные задачи, требующие программирования на низком уровне.

Важным дополнением к этому является среда разработки Flowcode, созданная компанией Matrix Multimedia. Flowcode предоставляет пользователю графический интерфейс, интуитивно понятный даже для начинающих разработчиков. Благодаря поддержке разработки программ в виде блок-схем, пользователи могут сосредоточиться на функциональности проекта, не углубляясь в сложные технические детали программирования [1, 2]. Это делает Flowcode мощным инструментом для обучения, ускоряя процесс разработки и снижая вероятность ошибок.

ОБСУЖДЕНИЕ

Данная статья ориентирована на начинающих разработчиков, которые только начинают свой путь в изучении микроконтроллеров. Многие из них сталкиваются с необходимостью изучать сложные профессиональные среды, такие как MPLAB IDE, которые, хотя и предоставляют широкий функционал, могут быть сложными для понимания. Flowcode, напротив, предлагает упрощённый подход к проектированию, что делает её идеальной для первых шагов в этой области.

Одним из ключевых преимуществ Flowcode является её совместимость с различными микроконтроллерами, включая популярные модели серии PIC. Среда позволяет быстро разрабатывать и тестировать проекты, экономя время и силы разработчиков [2, 3]. Это особенно важно для образовательных целей, где время на изучение инструментов ограничено, а основной упор делается на практическое применение знаний.

Для демонстрации возможностей набора разработчика PICkit 2 и среды Flowcode мы рассмотрим построение простейшего проекта управления светодиодом. Такой проект позволяет на практике изучить базовые операции, такие как:

- Инициализация портов микроконтроллера;
- Работа с логическими уровнями;
- Управление периферийными устройствами;
- Использование блоков в Flowcode для реализации базовой логики.

В процессе создания проекта мы изучим не только основы работы с микроконтроллерами, но и принципы взаимодействия между программным обеспечением и аппаратной частью. С помощью Flowcode разработчик сможет создавать блок-схемы, которые автоматически преобразуются в программный код, минимизируя вероятность синтаксических ошибок. Это делает процесс обучения более продуктивным и увлекательным.

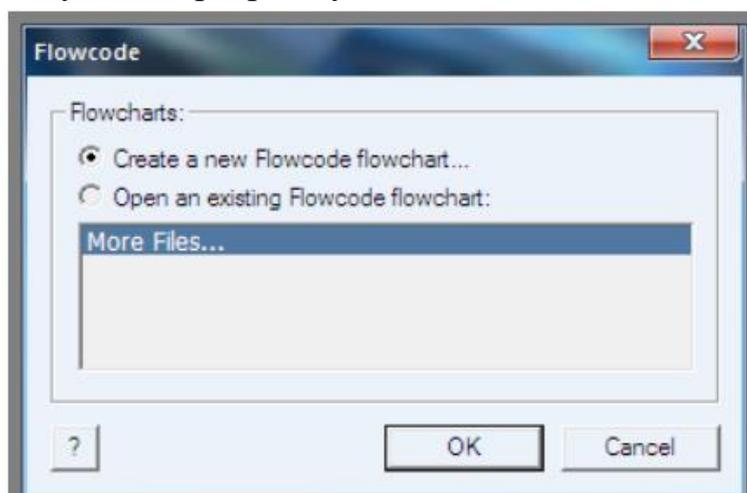
РЕЗУЛЬТАТЫ

Обычно создание проекта начинается с планирования последовательности действий (алгоритма), которые должен выполнять микроконтроллер. В среде разработки Flowcode для данных целей предусмотрен наглядный интерфейс по созданию блок-схемы программы.

В качестве первого примера разработаем систему, которая будет приводить в действие один из двух режимов «моргания» светодиодами в соответствии с положением внешнего переключателя. Алгоритм программы приведен на рис. 1.

Шаг второй — реализация блок-схемы в Flowcode

- Этапы реализации блок-схемы в Flowcode (рис. 2):
- Запустите программу Flowcode.



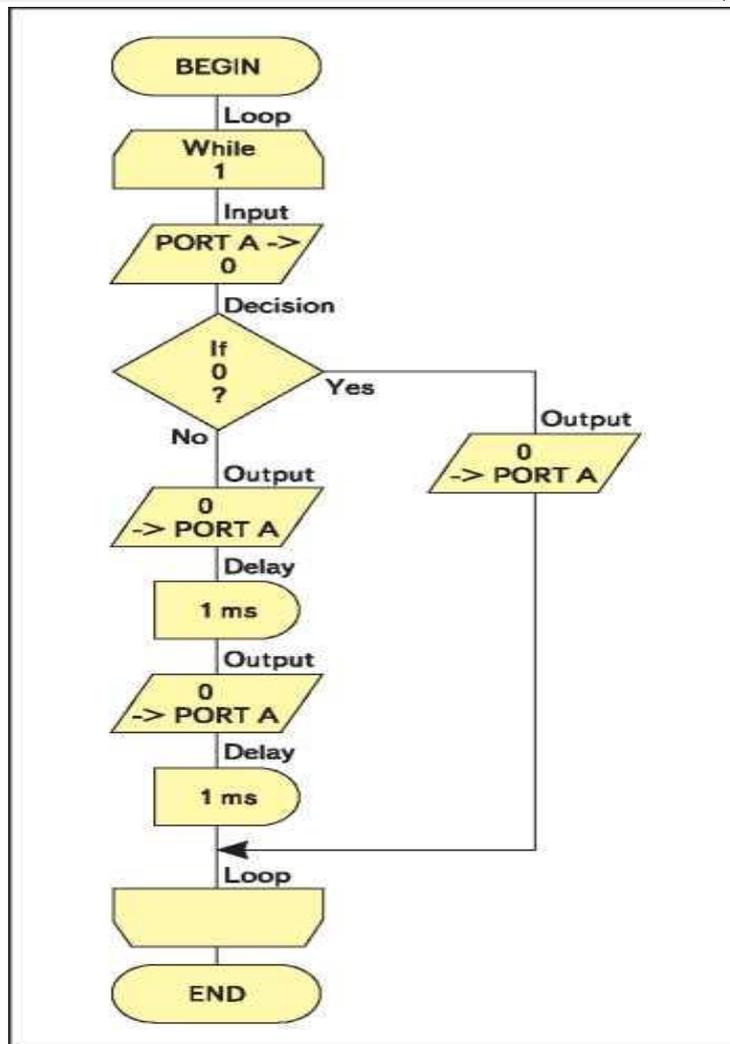
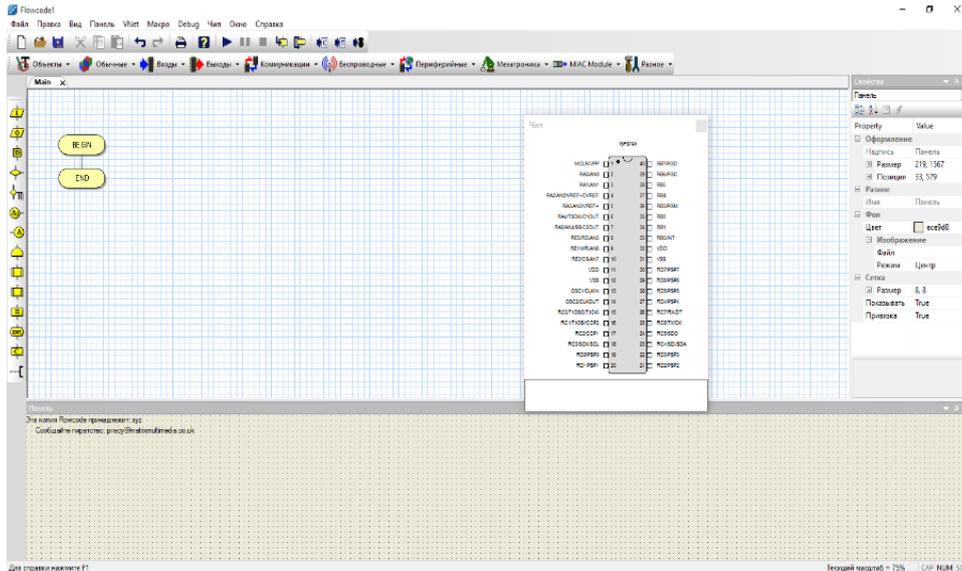


Рис. 2. Блок-схема, созданная во Flowcode

- Нажмите ОК, когда откроется окно Reminder Screen.
- Создайте новый проект (Create a new FlowCode flowchart).
- Выберите в качестве контроллера (Choose a target) PIC16F887 (он

установлен на демонстрационной плате 44-pin Demo Board комплекта PICkit 2 Debug Express).

- Откройте новую рабочую область (workspace) с названием “Main”.
- Добавьте необходимые блоки для блок-схемы. Для этого наведите курсор на нужную иконку в левой панели и, удерживая левую кнопку мыши, перетащите ее в рабочую область.
- Добавьте необходимые элементы (светодиоды LED, переключатель). Для этого щелкните мышью по иконке соответствующего элемента в верхней панели.
- В выпадающем меню окна LEDs («Светодиоды») выберите пункт Component Connections. В меню Connect to Port («Соединение с портом») выберите PORT D (светодиоды демонстрационной платы 44-pin Demo Board выведены на PORT D). Данный диалог должен принять вид, который представлен на рис. 3.

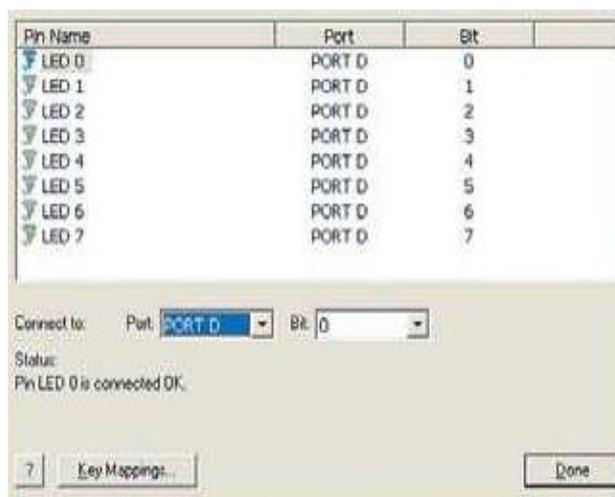


Рис. 3. Выпадающее меню окна LEDs («Светодиоды»)

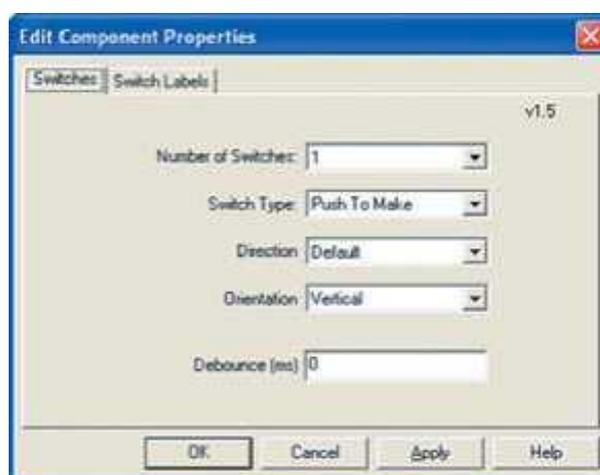


Рис. 4. Выпадающее меню окна Switches («Переключатели»)

• В выпадающем меню окна Switches («Переключатели») выберите пункт Properties («Свойства») и укажите количество кнопок— 1 (рис. 4).

• В пункте Connect to Port выпадающего меню выберите PORT B (кнопка на плате 44-pin Demo Board подключена к PORT B, бит 0). Диалог Connect to Port примет вид, который представлен на рис. 5.

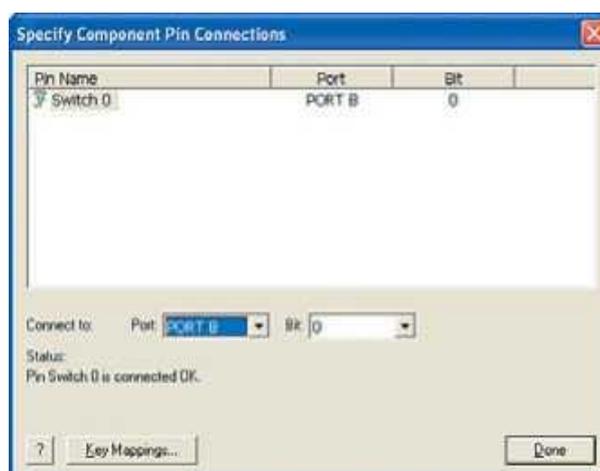


Рис. 5. Диалоговое окно Connect to Port

Настройка элементов блок-схемы

Теперь настало время настройки элементов, составляющих блок-схему. Для настройки элемента нужно дважды щелкнуть на его иконке мышью. Для удобства чтения блок-схемы в строке Display name напишите короткое пояснение о функциональном назначении блока.

Loop («Основной цикл»)



Рис. 6. Основной цикл

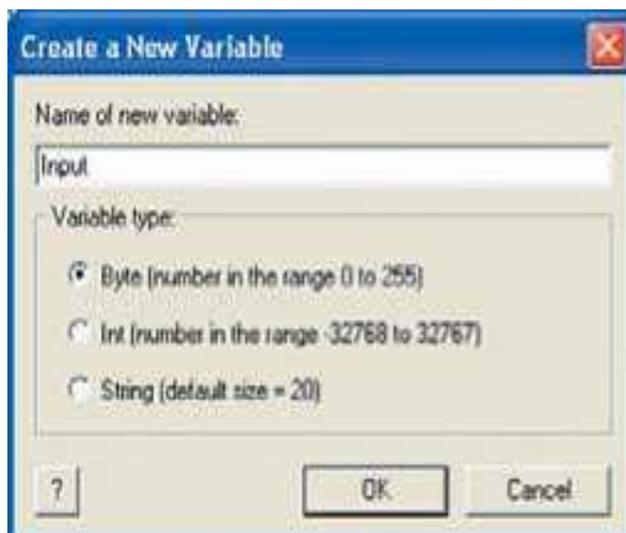


Рис. 7. Добавление новой переменной с названием Input

Input

Для начала необходимо создать переменную, отвечающую за информацию о положении кнопки. Щелкните по Variables («Переменные»), чтобы открыть диалог Variables Manager. Щелкните по Add New Variable («Добавить новую переменную») и добавьте новую переменную с названием Input (рис. 7).

Нажмите ОК. Вернувшись в окно Variables Manager, нажмите кнопку Use Variable. Вернувшись в окно Properties, выберите PORT B, Single Bit (рис. 8).

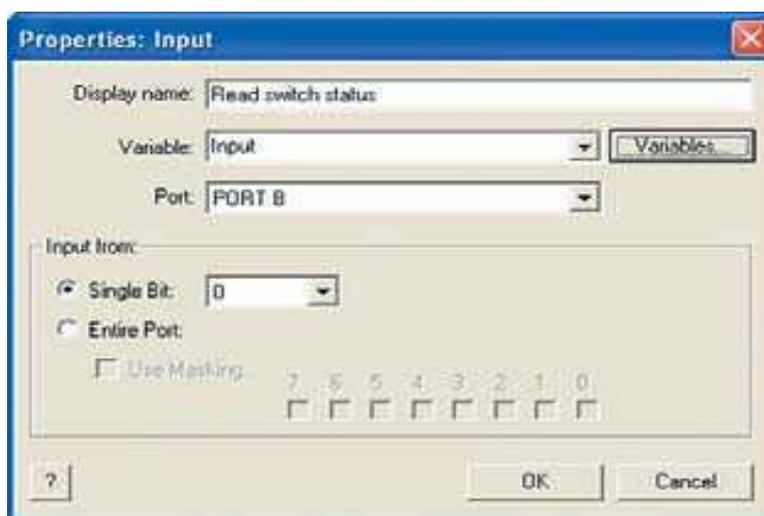


Рис. 8. Настройка новой переменной в окне Variables Manager

Decision

В окне Properties нажмите на Variables и выберите Input, затем щелкните на Use Variable. Дайте название (Display name) — “Is switch pressed?” (рис. 9).

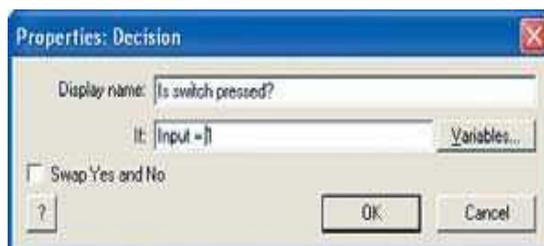


Рис. 9. Введение названия для новой переменной

Ветка No/Output

Чтобы зажечь крайние светодиоды, необходимо выставить в единицу биты 0 и 7 порта PORT D, то есть ввести в PORT D число 129 (27+20) (рис. 10).



Рис. 10. Введение в PORT D числа 129

Ветка Yes /

Первый блок Output

Чтобы зажечь светодиоды с 0 по 3, необходимо ввести в PORT D число 15 (20+21+22+23) (рис. 11).

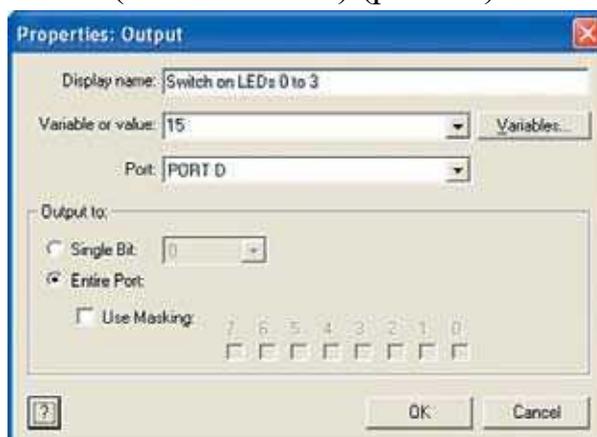


Рис. 11. Введение в PORT D числа 15

Оба блока Delay

Длительность задержки может быть задана в миллисекундах и секундах.

Необходимо задать задержку в полсекунды (рис. 12).



Рис. 12. Задание длительности задержки

Второй блок Output

Чтобы зажечь светодиоды с 4 по 7, необходимо ввести в PORT D число 240 (24+25+26+27) (рис. 13).

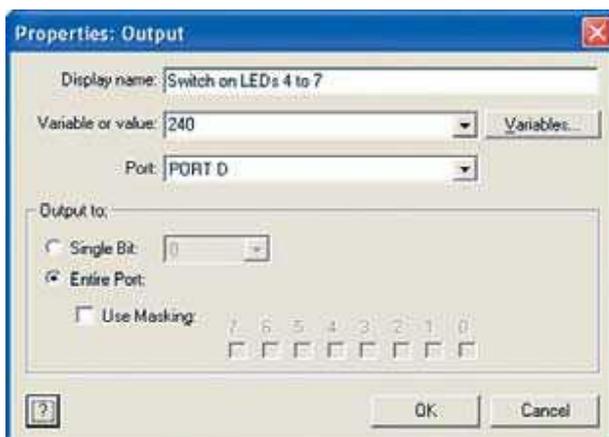
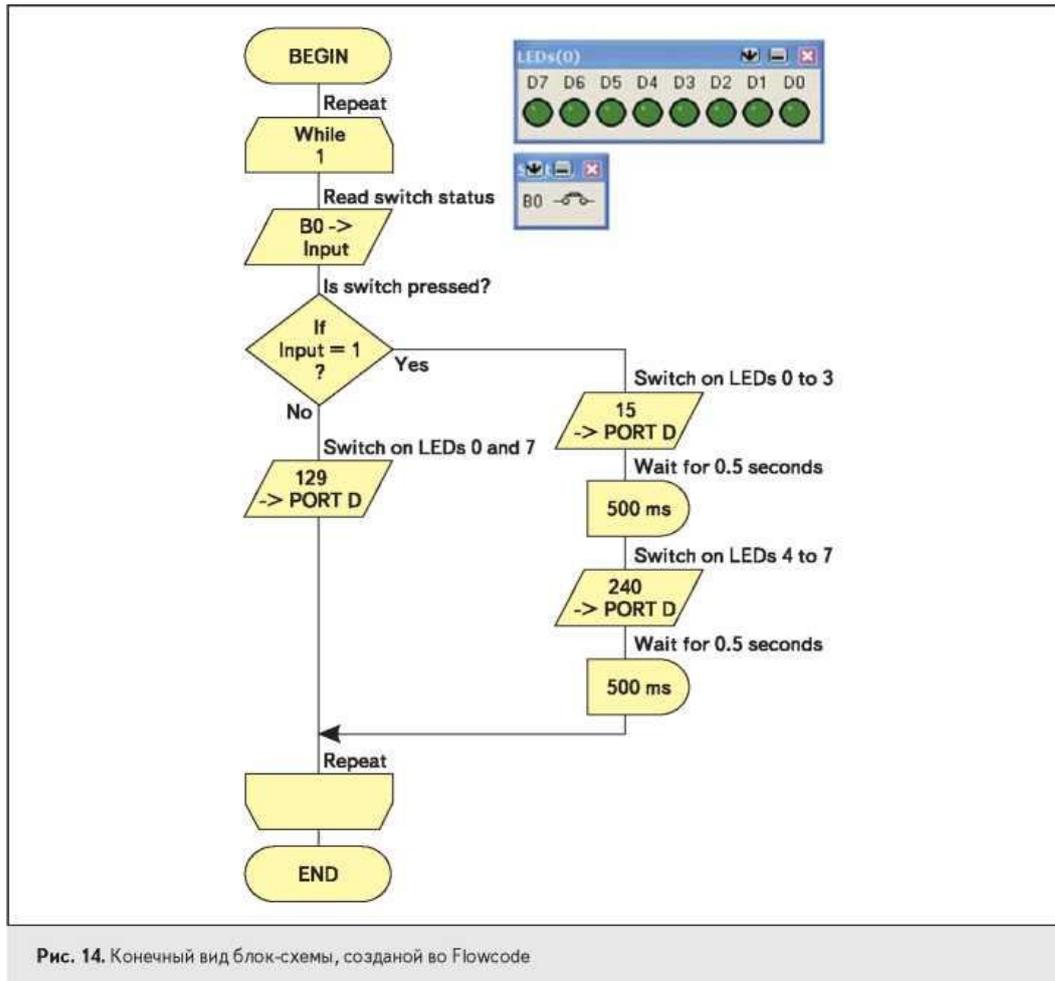


Рис. 13. Введение в PORT D числа 240

Теперь блок-схема должна иметь вид, представленный на рис. 14.

Программа Flowcode предоставляет возможность проверить работу программы через симуляцию, что является ключевым этапом при разработке и тестировании алгоритмов. Симуляция доступна в двух режимах: непрерывная и пошаговая. В первом режиме программа выполняется на полной скорости, что позволяет быстро получить общий результат. Во втором режиме выполнение программы происходит поэтапно, блок за блоком, что полезно для более детального анализа каждого шага.



В обоих режимах на экране отображаются данные в окнах Variables и Call Stack, где можно наблюдать текущие значения всех переменных и состояния стека вызовов. В случае, когда симуляция запускается на полной скорости, данные в этих окнах обновляются не так часто, что позволяет ускорить процесс, но исключает подробный мониторинг переменных. Однако, если скорость симуляции ниже максимальной или выполнение программы идет пошагово, все изменения в переменных и стеке отображаются в реальном времени, обновляясь на каждой итерации. Это дает возможность более детально анализировать поведение программы, отслеживать изменения значений и выявлять потенциальные ошибки или неточности в логике работы алгоритмов.

Теперь посмотрим на симуляцию на полной скорости. Нажмите Run на панели инструментов. Используйте возможность остановки и продолжения работы программы (F5 или Go/Continue в меню Run). При этом можно увидеть текущие значения в окнах Variables и Call Stack, а также крайние свето-диоды (0-й и 7-й светятся). Щелкните по иконке кнопки и убедитесь, что светодиоды 0-3 моргают в противофазе со светодиодами 4-7. Программа работает

корректно.

Если вы не получили нужного результата, вернитесь к началу и повторите все заново.

Программирование PIC-контроллера

Подготовительная часть закончена, вы получили полнофункциональную программу, работающую по заданному алгоритму. Выполните следующие инструкции для программирования контроллера:

- Подключите PICkit2 к ПК через USB.
- Подключите PICkit2 к демонстрационной плате 44-pin Demo Board.
- В меню Chip выберите пункт Compile to Chip (для компиляции необходимо сохранить созданную программу).
- Откроется окно Compile Messages, в котором отображается текущий прогресс. При программировании контроллера засветятся индикаторы Target и Busy на PICkit2. Когда программирование закончится, появится строка Finished, после чего следует нажать кнопку Close.
- Ваша программа должна запуститься. Светодиоды 0-3 должны мигать в противофазе со светодиодами 4-7. При нажатии кнопки должны светиться только светодиоды 0 и 7.

• Незаметно для пользователя программа Flowcode сформировала по графической блок-схеме C-файл и компилировала его в ассемблер. Полученный C- и asm-код можно просмотреть, выбрав в меню Chip соответственно команды “View C” и “View ASM”

либо просмотрев содержание файлов *progname.c* и *progname.asm* в папке, где сохранен данный проект.

ЗАКЛЮЧЕНИЕ

В данной статье показано, как просто создать программу для микроконтроллера по заданному алгоритму с использованием среды Flowcode. Приведенный пример охватывает лишь небольшую часть функциональных возможностей этой платформы [4, 5]. Остались незатронутыми такие аспекты, как выполнение вычислительных операций, работа со строковыми переменными, создание пользовательских макросов, вставка пользовательского C-кода, а также взаимодействие с более сложными периферийными устройствами.

Для изучения этих возможностей Flowcode предлагает подробные справочные материалы, доступные в меню Help, которые помогают пользователям освоить новые инструменты и расширить свои знания.

Дополнительно среда поддерживает широкий набор библиотек, упрощающих внедрение сложных решений, таких как работа с коммуникационными интерфейсами (UART, SPI, I2C) и интеграция с IoT-платформами.

Использование Flowcode в образовательных и исследовательских целях способствует значительному упрощению процесса разработки программного обеспечения для микроконтроллеров [9, 10]. Это не только ускоряет реализацию проектов, но и повышает наглядность изучаемых материалов, что особенно важно для начинающих разработчиков.

Flowcode открывает возможности для создания как простых, так и сложных автоматизированных систем, интеграции с современными технологиями и оптимизации работы микроконтроллеров. Благодаря своей интуитивности и богатому функционалу, Flowcode является эффективным инструментом как для обучения, так и для профессионального использования.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Flowcode User Guide. Official documentation from Matrix TSL. URL: <https://www.matrixtsl.com>
2. Matrix TSL. Flowcode 8: Introduction to Flowcode. – Matrix Technology Solutions, 2020.
3. Mazidi, M. A., Naimi, S., & Naimi, S. The PIC Microcontroller and Embedded Systems Using Assembly and C for PIC18. – Pearson Education, 2016.
4. Barnett, R. H., Cox, L., & O’Cull, L. Embedded C Programming and the Atmel AVR. – Cengage Learning, 2012.
5. Волков, С. А. Микроконтроллеры в автоматизированных системах управления. – Москва: МИЭТ, 2019.
6. Malvino, A., & Bates, D. Electronic Principles. – McGraw-Hill Education, 2016.
7. Алиев, Т. А. Оптимизация программирования микроконтроллеров на языке С. – СПб: Питер, 2018.
8. Ponnusamy, S., & Thiagarajan, R. Optimization of Embedded System Design. – Springer, 2020.
9. Джонсон, С. Микроконтроллеры PIC: от новичка к профессионалу. – Москва: ДМК Пресс, 2020.
10. Palamar, R. IoT и микроконтроллеры: практическое руководство. – Москва: БХВ-Петербург, 2021.
11. Penfold, R. A. Practical PIC Projects. – Bernard Babani Publishing, 2017.
12. Google Scholar. Поиск по ключевым словам: "Flowcode", "PIC"

- microcontroller optimization", "Embedded systems programming". URL: [\[https://scholar.google.com\]](https://scholar.google.com)(<https://scholar.google.com>)
13. Спецификации микроконтроллеров семейства PIC. URL: [\[https://www.microchip.com\]](https://www.microchip.com)(<https://www.microchip.com>)
14. Ochilov, M. A., Juraev, F. D., Maxmatqulov, G. X., & Rahimov, A. M. (2020). Analysis of important factors in checking the optimality of an indeterminate adjuster in a closed system. *Journal of Critical Review*, 7(15), 1679-1684.
15. Jo'rayev, F. D. S., & Ochilov, M. A. (2023). Algorithms for multi-factory polynomial modeling of technological processes. *Chemical Technology, Control and Management*, 2023(1), 59-67./ <https://ijctcm.researchcommons.org/journal/vol2023/iss1/8/>.
16. Ashurkulovich, O. M., & Kholikulovich, J. A. (2024). IMPLEMENTATION OF ROUTING ON A COMPUTER IN A CISCO PACKET TRACER ENVIRONMENT. *American Journal of Innovation in Science Research and Development*, 1(4), 36-43.
17. ОЧИЛОВ, М. А. (2023). РОЛЬ CISCO PACKET TRACER В МОДЕЛИРОВАНИИ И АНАЛИЗЕ СЕРВЕРНЫХ УСЛУГ В КЛИЕНТ-СЕРВЕРНЫХ СЕТЯХ. *Academic research in educational sciences*, 4(11), 275-286.
18. Dustmirzayevich, J. F., & Ashirkulovich, O. M. (2023). THE ROLE OF CISCO PACKET TRACER SOFTWARE IN STUDYING COMPUTER SYSTEMS AND NETWORKS. *PEDAGOGS*, 46(2), 17-27.
19. Ochilov, M. A. (2023). DESIGNING TECHNOLOGICAL PROCESS AUTOMATION SYSTEMS BASED ON SCADA. *Oriental renaissance: Innovative, educational, natural and social sciences*, 3(2), 1025-1032.
20. ЖУРАЕВ, А. (2020). Таълим тизимида педагогик дастурий воситаларни жорий этиш афзалликлари. *UNIVERSITETI XAVARLARI*, 1(1).
21. Жураев, А. Х., & Тожибоев, С. Ж. Ў. (2022). СИМУЛЯТОР ДАСТУРЛАРИДАН ТАЪЛИМ ЖАРАЁНИДА ФОЙДАЛАНИШ. *Oriental renaissance: Innovative, educational, natural and social sciences*, 2(5), 557-565.
22. Juraev, A. K., Jurayev, F. D., Eshkobilov, S. B., Ibragimov, B. S., & Norboev, O. N. (2023). Nonlinear control object identification problems: Methods and approaches. In *E3S Web of Conferences* (Vol. 392, p. 02043). EDP Sciences.
23. Жураев, А. Х. (2022). ЭЛЕКТРОН ЎҚУВ ВА ДИДАКТИК МАТЕРИАЛЛАРНИ ЯРАТИШ ИМКОНИНИ БЕРУВЧИ ДАСТУРЛАР ТАХЛИЛИ. *Academic research in educational sciences*, 3(2), 572-577.

24. Xoliqulovich, J. A. (2023). THE USE OF MATLAB IN TEACHING THE PROCESS OF AUTOMATIC TEMPERATURE ADJUSTMENT IN HIGHER EDUCATION INSTITUTIONS. *Galaxy International Interdisciplinary Research Journal*, 11(2), 351-356.
25. Jurayev, A. K. (2024). THE ROLE OF INFORMATION TECHNOLOGY IN THE TEACHING OF TECHNICAL SCIENCES. *Oriental renaissance: Innovative, educational, natural and social sciences*, 4(5), 842-850.