УДК 621.391
DOI: 10.18799/29495407/2024/4/75
Шифр специальности ВАК: 2.2.4

Введение в автоматное программирование микроконтроллеров STM32

В.А. Вдовин, И.В. Трубин, В.Г. Трубин✉, П.Д. Шендрик

Новосибирский Государственный Технический Университет, Россия, г. Новосибирск

✉trubin@corp.nstu.ru

Аннотация. Рассматривается проблема реализации псевдомногозадачности на микроконтроллере STM32, что является актуальной задачей для повышения эффективности работы устройств. Одним из возможных решений является использование автоматного программирования, что и описывается в статье. Приведены и разработаны детализированные фрагменты кода, которые иллюстрируют различные подходы к программированию микроконтроллеров и обеспечивают более полное понимание темы. Показано, как подключить отладочную плату BluePill к персональному компьютеру через преобразователь USB-UART.

Ключевые слова: цифровые системы управления, микроконтроллер, автоматное программирование, псевдомногозадачность, конечные автоматы, UART, преобразователь USB-UART, BluePill, STM32F103, ASCII, moserial, Terminal

Для цитирования: Введение в автоматное программирование микроконтроллеров STM32 / В.А. Вдовин, И.В. Трубин, В.Г. Трубин, П.Д. Шендрик // Известия Томского политехнического университета. Промышленная кибернетика. – 2024. – Т. 2. – № 4. – С. 21–25. DOI: 10.18799/29495407/2024/4/75

UDC 621.391
DOI: 10.18799/29495407/2024/4/75

Introduction to automatic programming of STM32 microcontrollers

V.A. Vdovin, I.V. Trubin, V.G. Trubin✉, P.D. Shendrik

Novosibirsk State Technical University, Novosibirsk, Russia

✉trubin@corp.nstu.ru

Abstract. The article deals with the problem of implementing pseudo-multitasking on the STM32 microcontroller, which is an urgent task to improve the efficiency of devices. One of the possible solutions is the use of automatic programming. It is described in the article. Detailed code fragments are provided and developed. This illustrates various approaches to programming microcontrollers and provides a more complete understanding of the topic. The article shows as well how to connect the BluePill debugging board to a personal computer via a USB-UART converter.

Keywords: digital control systems, microcontroller, automatic programming, pseudo-multitasking, finite-state machines, UART, USB-UART converter, BluePill, STM32F103, ASCII, moserial, Terminal

For citation: Vdovin V.A., Trubin I.V., Trubin V.G., Shendrik P.D. Introduction to automatic programming of STM32 microcontrollers. *Bulletin of the Tomsk Polytechnic University. Industrial Cybernetics*, 2024, vol. 2, no. 4, pp. 21–25. DOI: 10.18799/29495407/2024/4/75

Введение

При разработке цифровых систем управления [1], как правило, используются микроконтроллеры, которые выполняют целый ряд определённых задач. Большая часть недорогих микроконтроллеров имеют одно вычислительное «ядро», которое выполняет заданную программу последовательно, шаг за шагом. Однако задачи обычно требуется выполнять одновременно – генерировать импульсы, обмениваться данными с устройствами, выполнять опрос датчиков, анализировать данные, проводить измерения с помощью АЦП и так далее.

Таким образом, при написании программ для микроконтроллеров приходится обеспечивать так называемую «псевдомногозадачность». В данной статье предлагается к рассмотрению способ автоматного программирования на примере решения типовой задачи.

Концепция

В основе автоматного программирования лежит концепция разбиения всех задач на малые подзадачи, каждая из которых не требует большого количества времени на своё выполнение.

При выполнении бесконечного цикла *while()* в главной функции *main()* происходит попеременный вызов функций (конечных автоматов), каждая из которых представляет собой пошаговую реализацию своей конкретной задачи. Внутри каждого такого автомата, в момент его вызова, происходит выполнение только одной подзадачи, необходимой для последовательного выполнения всей задачи.

Конечные автоматы оформлены в форме конструкции *switch()*, а подзадачи – в виде их *case*'ов. При этом для соблюдения последовательного выполнения подзадач вводится переменная-счётчик, называемая *state*, по которой и работает оператор *switch()*. Схематичное представление изображено на рис. 1.

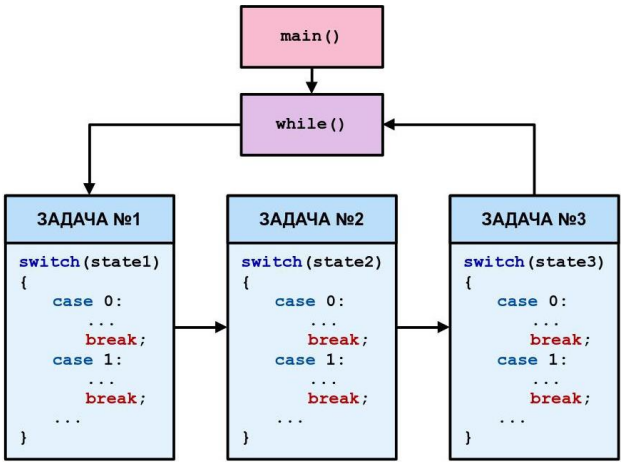

Рис. 1. *Концепция автоматного программирования*
Fig. 1. *Concept of automatic programming*

Пример реализации

Рассмотрим автоматное программирование на примере решения типовой задачи. Необходимо последовательно отправить содержимое массива размером 100 байт через интерфейс *UART* [2] при получении символа «P». Схематичное представление этой задачи изображено на рис. 2, где МК – микроконтроллер, а ПК – персональный компьютер.

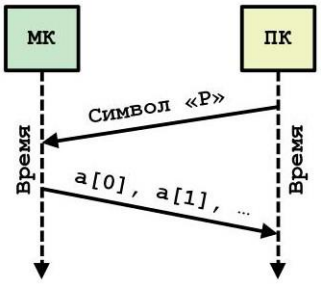

Рис. 2. *Схематичное представление задачи*
Fig. 2. *Schematic representation of the task*

Всю инициализацию микроконтроллера сделаем в отдельном файле *«main_init.c»*.

Выполним инициализацию интерфейса *USART1*.

Блок кода 1. *Файл «main_init.c»*

```
// Настройки по умолчанию:
// 8 инф. бит, 1 стоповый бит, контроля чётно-
сти нет
// Включить тактирование USART1
RCC->APB2ENR |= RCC_APB2ENR_USART1EN;
// PA9 (TX1) AFIO Push-Pull, 10MHz. PA10 (RX1)
HiZ, 10MHz
GPIOA->CRH &= ~(GPIO_CRH_MODE9 |
GPIO_CRH_CNF9);
GPIOA->CRH &= ~(GPIO_CRH_MODE10 |
GPIO_CRH_CNF10);
GPIOA->CRH |= (GPIO_CRH_MODE9_0 |
GPIO_CRH_CNF9_1 | GPIO_CRH_CNF10_0);
// Выбор скорости работы порта (9600 бит/с)
USART1->BRR = 7500; // PCLK2 / Baud = 72000000
/ 9600 бод
// // Включение USART, передатчика и приемника
USART1->CR1 = USART_CR1_UE | USART_CR1_TE |
USART_CR1_RE;
```

После инициализации рассмотрим описание главного файла *«main.c»*. Так, сначала необходимо выполнить подключение заголовочных файлов.

Блок кода 2. *Файл «main.c»*

```
// Описание регистров микроконтроллера
#include "stm32f10x.h"
// Файл с инициализацией микроконтроллера
#include "main_init.c"
```

Далее идёт основная программа. Стандартным решением можно считать код, написанный на языке программирования Си в блоке кода 3.

В нем ожидание получения символа «P» осуществляется путём «зависания» в цикле до тех пор, пока не будет получен соответствующий символ. После получения символа начинается отправка всех байт последовательно. Для ожидания завершения передачи одного байта используется такое же «зависание» в цикле, пока загруженный символ не будет передан в линию *UART*.

Блок кода 3. *Файл «main.c»*

```
// Байтовый массив с данными
uint8_t a[100];
// Номер байта массива
uint8_t i;

int main() {
  // Бесконечный цикл while
  while(1) {
    // Ожидать получения символа
    while (!(USART1->SR & USART_SR_RXNE));
    // Если пришёл нужный символ
    if (USART1->DR == 'P') {
      // Цикл на 100 итераций
      for (i = 0; i < 100; i++) {
        // Ожидать готовности передатчика
        while (
         !(USART1->SR & USART_SR_TXE)
        ) {};
        // Отправить очередной байт
        USART1->DR = a[i];
      }
    }
  }
}
```

В автоматном же программировании ключевым условием является то, что ни одна задача не вводит микроконтроллер в состояние «зависания» для ожидания наступления какого-либо события.

Автоматное программирование позволяет «освободить» микроконтроллер от ожидания наступления необходимого события. Делается так, чтобы микроконтроллер выполнял необходимые действия лишь при наступлении события, а всё остальное время мог выполнять другие задачи.

Пример решения рассмотренной выше задачи в автоматном программировании представлен в блоке кода 4.

Проверка

Так как описываемая в статье программа предполагает обмен данными по интерфейсу *UART*, для получения наглядного результата можно воспользоваться *USB-UART* [3] преобразователем, который позволит ПК обмениваться данными с микроконтроллером.

Блок кода 4. *Файл «main.c»*

```
// Байтовый массив с данными
uint8_t a[100];
// Номер байта массива
uint8_t i;
// Состояние конечного автомата
uint8_t state = 0;

int main() {
  // Бесконечный цикл while
  while(1) {
    // Конечный автомат
    switch(state) {
      // Шаг №0 — ожидание нужного символа
      case 0:
        // Если пришёл символ из линии
        if (USART1->SR & USART_SR_RXNE) {
          // Если пришёл нужный символ
          if (USART1->DR == 'P') {
            // Перейти к шагу отправки
            state = 1;
            // Сбросить номер байта
            i = 0;
          }
        }
        break;
      // Шаг №1 — отправка байт
      case 1:
        // Если передатчик готов
        if (USART1->SR & USART_SR_TXE) {
          // Отправить очередной байт
          USART1->DR = a[i++];
          // Если всё передано
          if (i == 100) state = 0;
        }
        break;
    }
  }
}
```

Вывод преобразователя *VCC* используется для указания ему уровня логической единицы (3.3 В или 5 В). Так как проверка выполнялась на отладочной плате *BluePill* [4] с микроконтроллером *STM32F103* [5, 6], у которого уровень логической единицы равен 3.3 В, вывод *VCC* был подключен к выводу *3V3*.

Схема подключения изображена на рис. 3, а на рис. 4 можно увидеть фото собранной схемы.

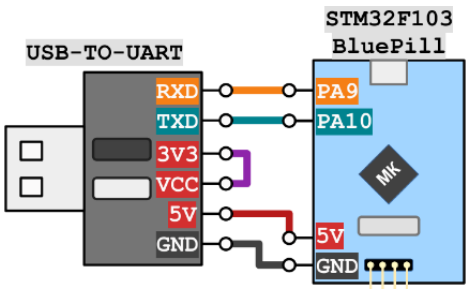

Рис. 3. *Схема подключения отладочной платы BluePill к преобразователю USB-UART*
Fig. 3. *Scheme of the BluePill debugging board connection to the USB-UART converter*

Необходимо чем-нибудь заполнить массив «a», состоящий из 100 байт. В качестве численных значений, например, можно использовать символы, согласно таблице ASCII (так как каждый из символов в этой кодировке умещается в 1 байт). В данном примере массив заполнен цифрами от 0 до 9, согласно таблице ASCII.

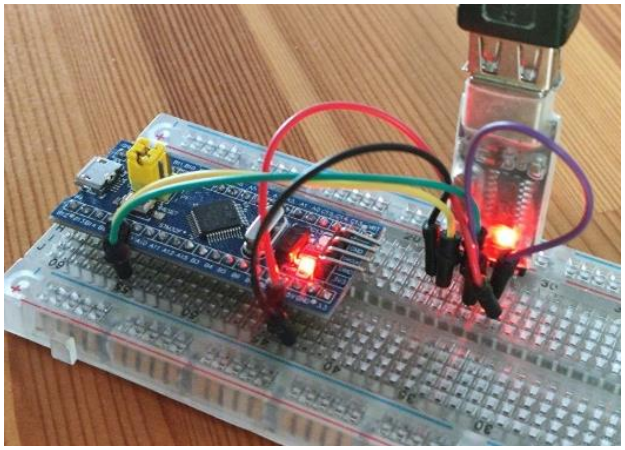

Рис. 4. *Фото собранной схемы*
Fig. 4. *Photo of the assembled circuit*

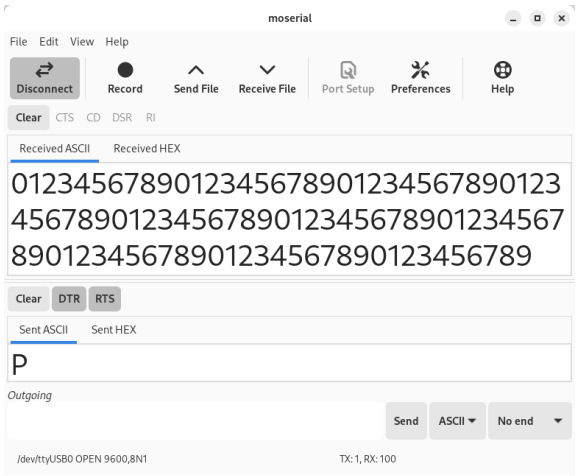

Рис. 5. *Результат обмена данными в программе moserial*
Fig. 5. *Result of data exchange in the moserial program*

Для того чтобы ПК мог взаимодействовать с преобразователем *USB-UART*, необходимо установить на него программу для работы с виртуальными последовательными портами (так называемыми *COM*-портами). В операционных системах (ОС) на базе ядра *GNU/Linux* с этой задачей отлично справляется программа *moserial* (рис. 5).

В ОС *Windows* можно воспользоваться программами *PuTTY* или *Terminal* [7].

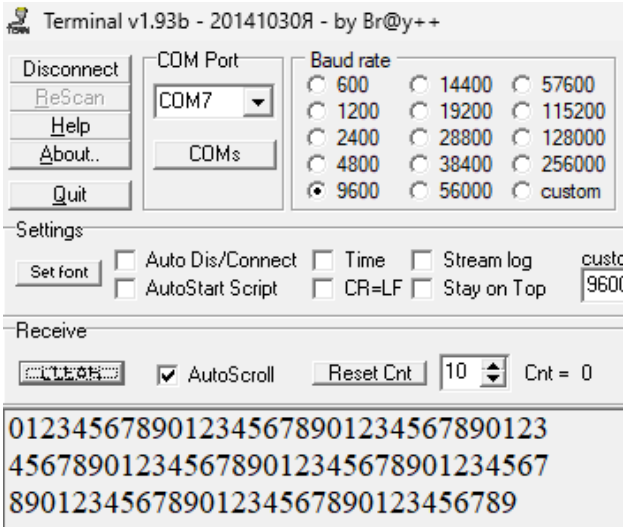

Рис. 6. *Результат обмена данными в программе Terminal*
Fig. 6. *Result of data exchange in the Terminal program*

Заключение

- Рассмотрена концепция автоматного программирования для микроконтроллеров *STM32*.
- Предложен способ простой реализации псевдомногозадачности.
- Приведены примеры кода на языке программирования Си, выполняющие одну и ту же типовую задачу разными способами.
- Выполнена проверка работоспособности приведённого кода на примере популярной отладочной платы *BluePill*.

СПИСОК ЛИТЕРАТУРЫ

1. Разработка цифровых систем управления. URL: http://kb-au.ru/?page_id=141 (дата обращения: 05.11.2024).
2. Близнюк А.Е., Трубин В.Г., Саблина Г.В. Представление и отображение информации при работе с микроконтроллерами STM32 // Автоматика и программная инженерия. – 2022. – № 3 (41). URL: http://kb-au.ru/wp-content/uploads/AaSI-3-2022-7.pdf (дата обращения: 05.11.2024).
3. Жмудь В.А., Трубин И.В., Трубин М.В. Обмен данными между компьютером и микроконтроллером STM32F100 по последовательному интерфейсу связи RS-232 // Автоматика и программная инженерия. – 2015. – № 1 (11). URL: http://jurnal.nips.ru/sites/default/files/АИПИ-1-2015-6.pdf (дата обращения: 05.11.2024).
4. Blue Pill. URL: https://www.belchip.by/sitedocs/31025.pdf (дата обращения: 05.11.2024).
5. Документация на микроконтроллер STM32F103. URL: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf (дата обращения: 05.11.2024).

6. Reference manual STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced Arm®-based 32-bit MCUs. URL: https://www.st.com/content/ccc/resource/technical/document/reference_manual/59/b9/ba/7f/11/af/43/d5/CD00171190.pdf/files/CD00171190.pdf/jcr:content/translations/en.CD00171190.pdf (дата обращения: 05.11.2024).
7. Terminal v1.93b URL: https://sites.google.com/site/terminalbpp/ (дата обращения: 05.11.2024).

Информация об авторах
Владислав Андреевич Вдовин, студент, кафедра автоматики, Новосибирский государственный технический университет, Россия, 630073, г. Новосибирск, пр. Карла Маркса, 20; vladislickness@mail.ru
Игорь Витальевич Трубин, старший преподаватель, кафедра защиты информации, Новосибирский государственный технический университет, Россия, 630073, г. Новосибирск, пр. Карла Маркса, 20; i.trubin@corp.nstu.ru; https://orcid.org/0009-0001-1466-6001
Виталий Геннадьевич Трубин, старший преподаватель, кафедра автоматики, Новосибирский государственный технический университет, Россия, 630073, г. Новосибирск, пр. Карла Маркса, 20; trubin@corp.nstu.ru
Павел Дмитриевич Шендрик, студент, кафедра автоматики, Новосибирский государственный технический университет, Россия, 630073, г. Новосибирск, пр. Карла Маркса, 20; pasha.shendrick@yandex.ru

Поступила: 10.11.2024
Принята: 20.12.2024
Опубликована: 28.12.2024

REFERENCES

1. *Development of digital control systems*. (In Russ.) Available at: http://kb-au.ru/?page_id=141 (accessed: 5 November 2024).
2. Bliznyuk A.E., Trubin V.G., Sablina G.V. Representation and display of information when working with STM32 microcontrollers. *Automatics & Software Enginery*, 2022, no. 3 (41). (In Russ.) Available at: http://kb-au.ru/wp-content/uploads/AaSI-3-2022-7.pdf (accessed: 5 November 2024).
3. Zhmud V.A., Trubin I.V., Trubin M.V. Exchange of data between the computer and the microcontroller STM32F100 by serial communication interface RS-232. *Automatics & Software Enginery*, 2015, no. 1 (11). (In Russ.) Available at: http://jurnal.nips.ru/sites/default/files/АИПИ-1-2015-6.pdf (accessed: 5 November 2024).
4. *Blue Pill*. Available at: https://www.belchip.by/sitedocs/31025.pdf (accessed: 5 November 2024).
5. Documentation for the STM32F103 microcontroller. Available at: https://www.st.com/resource/en/datasheet/stm32f103c8.pdf. (accessed: 5 November 2024).
6. *Reference manual STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced Arm®-based 32-bit MCUs*. Available at: https://www.st.com/content/ccc/resource/technical/document/reference_manual/59/b9/ba/7f/11/af/43/d5/CD00171190.pdf/files/CD00171190.pdf/jcr:content/translations/en.CD00171190.pdf (accessed: 5 November 2024).
7. *Terminal v1.93b*. Available at: https://sites.google.com/site/terminalbpp/ (accessed: 5 November 2024).

Information about the authors
Vladislav A. Vdovin, Student, Novosibirsk State Technical University, 20, Karl Marks avenue, Novosibirsk, 630073, Russian Federation; vladislickness@mail.ru
Igor V. Trubin, Senior Lecturer, Novosibirsk State Technical University, 20, Karl Marks avenue, Novosibirsk, 630073, Russian Federation; i.trubin@corp.nstu.ru; https://orcid.org/0009-0001-1466-6001
Vitaly G. Trubin, Senior Lecturer, Novosibirsk State Technical University, 20, Karl Marks avenue, Novosibirsk, 630073, Russian Federation; trubin@corp.nstu.ru
Pavel D. Shendrik, Student, Novosibirsk State Technical University, 20, Karl Marks avenue, Novosibirsk, 630073, Russian Federation; pasha.shendrick@yandex.ru

Received: 10.11.2024
Revised: 20.12.2024
Accepted: 28.12.2024