

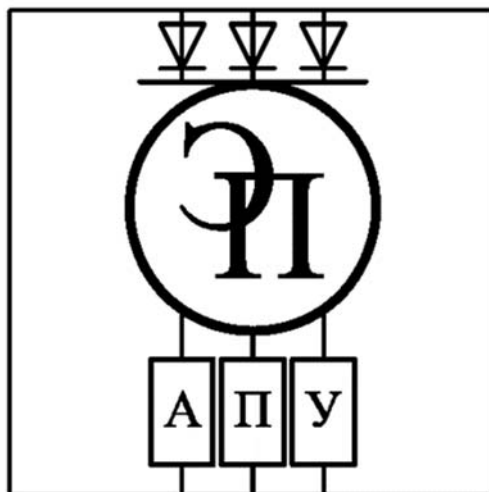
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и автоматизация промышленных установок»

# МИКРОПРОЦЕССОРНАЯ ТЕХНИКА ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ

*Методические рекомендации к лабораторным работам  
для студентов специальности*

*1-36 01 03 «Технологическое оборудование машиностроительного  
производства»  
очной и заочной форм обучения*



Могилев 2021

УДК 658.26  
ББК 31.19  
М75

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Электропривод и автоматизация промышленных установок» «3» февраля 2020 г., протокол № 7

Составитель О. А. Капитонов

Рецензент канд. техн. наук, доц. С. В. Болотов

Методические рекомендации предназначены для студентов специальности 1-36 01 03 «Технологическое оборудование машиностроительного производства» очной и заочной форм обучения.

Учебно-методическое издание

## МИКРОПРОЦЕССОРНАЯ ТЕХНИКА ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ

Ответственный за выпуск

Г. С. Леневский

Корректор

Е. А. Галковская

Компьютерная верстка

Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский  
университет, 2021

## Содержание

Введение.....	4
Инструкция по охране труда при проведении лабораторных работ .....	5
1 Лабораторная работа № 1. Изучение технических параметров, устройства и порядка работы с учебным оборудованием .....	6
2 Лабораторная работа № 2. Изучение организации ввода-вывода дискретных сигналов в микропроцессорных устройствах .....	11
3 Лабораторная работа № 3. Изучение организации семисегментной индикации в микропроцессорных устройствах .....	15
4 Лабораторная работа № 4. Изучение организации жидкокристаллической индикации в микропроцессорных устройствах .....	22
5 Лабораторная работа № 5. Изучение организации клавиатуры в микропроцессорных устройствах .....	24
6 Лабораторная работа № 6. Изучение организации устройств вывода аналоговой информации в микропроцессорных устройствах.....	27
7 Лабораторная работа № 7. Разработка и отладка программ управления технологическими объектами в реальном времени (часть 1).....	34
8 Лабораторная работа № 8. Разработка и отладка программ управления технологическими объектами в реальном времени (часть 2).....	36
Список литературы .....	40

## Введение

Целью учебной дисциплины является формирование специалистов, умеющих обоснованно и результативно применять существующие и осваивать новые методы использования микропроцессоров для управления технологическим оборудованием машиностроительного производства.

В результате освоения учебной дисциплины студент будет знать:

- классификацию микропроцессоров и архитектурные особенности микропроцессорных комплектов, выпускаемых промышленностью;
- структуру микропроцессорных систем управления, их аппаратную реализацию и состав программного обеспечения;
- методику проектирования микропроцессорных систем управления, расчета и выбора микропроцессорных средств;
- теоретические основы и принципы действия систем автоматического управления с использованием микропроцессоров;
- пути повышения экономических и экологических показателей двигателей внутреннего сгорания путем использования микропроцессорных контроллеров.

В результате освоения учебной дисциплины студент будет уметь:

- пользоваться методикой проектирования микропроцессорных систем управления;
- анализировать и производить сравнительную оценку вариантов рассматриваемой системы с использованием микропроцессора;
- осуществлять программирование и отладку микропроцессорных систем управления.

В результате освоения учебной дисциплины студент овладеет:

- методами использования микропроцессоров в системах управления ;
- методами внедрения управляющих комплексов и многомашинных (компьютерных) сетей в технологическом оборудовании машиностроительного производства.

## **Инструкция по охране труда при проведении лабораторных работ**

При проведении лабораторных работ имеются четыре основных вида опасностей по возможности нанесения существенного вреда здоровью студентов, лаборантов, преподавателей и других лиц, которые выполняют на лабораторном оборудовании определенные работы:

- 1) напряжение переменного тока 220 В частотой 50 Гц (агрегат электромашинный, станция управления, ПЭВМ);
- 2) напряжение постоянного тока 220 В (агрегат электромашинный);
- 3) вращающиеся части (агрегат электромашинный);
- 4) элементы лабораторного комплекса, имеющие рабочую температуру 100 °С и более.

### **Запрещается:**

- выполнение лабораторной работы без прохождения инструктажа по технике безопасности;
- выполнение лабораторной работы без разрешения преподавателя, проводящего лабораторную работу;
- подача напряжения питания на комплекс без разрешения преподавателя;
- коммутация органов управления комплекса при поданном напряжении питания без разрешения преподавателя;
- во время выполнения лабораторных работ шуметь, громко разговаривать, заниматься посторонними делами, быть одетыми в одежду с длинными рукавами, длинными полами.

# **1 Лабораторная работа № 1. Изучение технических параметров, устройства и порядка работы с учебным оборудованием**

**Цель работы:** изучить состав, функциональную схему учебного стенда; ознакомиться со структурой и принципом работы микроконтроллера, изучить порядок работы со стендом.

## ***1.1 Ход выполнения работы***

1.1.1 Изучить функциональную схему учебного стенда.

1.1.2 Изучить порядок работы с учебным стендом.

1.1.3 Изучить расположение рабочих органов и разъемов подключения учебного стенда.

1.1.4 Ответить на контрольные вопросы.

## ***1.2 Краткие теоретические сведения***

В микропроцессорной технике выделился самостоятельный класс интегральных схем – микроконтроллеры, которые предназначены для встраивания в приборы различного назначения. От класса однокристальных микропроцессоров их отличает наличие встроенной памяти, развитые средства взаимодействия с внешними устройствами.

Процессор Cortex-M4F, разработанный фирмой ARM, – это последнее поколение процессоров для встраиваемых систем. Он был разработан для того, чтобы получить дешевую платформу, необходимую для создания микроконтроллеров, с уменьшенным числом выводов и малым энергопотреблением, в сочетании с высокими вычислительными возможностями и уменьшенным временем отклика на прерывания.

ARM Cortex-M4F – это 32-битный RISC-процессор, характеризующийся исключительной эффективностью кода, обеспечивая при этом высокую производительность.

Процессор поддерживает набор DSP инструкций, которые позволяют производить эффективную обработку сигналов и выполнение комплексных алгоритмов. Он содержит FPU (floating point unit – блок вычислений с плавающей запятой) одинарной точности, который ускоряет разработку программного обеспечения и предотвращает насыщения.

Линейка микроконтроллеров STM32F407xx построено на базе процессора Cortex-M4F. Эти устройства оснащены большим набором периферийных устройств. Ввод/вывод информации можно осуществлять в параллельном виде через порты, количеством до 9, а также через широкую номенклатуру последовательных интерфейсов, таких как SPI, I<sup>2</sup>C, CAN, USB, USART. Поддерживает возможность подключения к нему карт памяти. Входящие в состав контроллера таймеры, количеством до 14, позволяют решать задачи,

связанные с выдержками времени, формированием ШИМ-сигналов, подсчетом частоты и длительности импульсных сигналов. А совместно со средствами ввода/вывода аналоговых сигналов позволяют использовать их в составе систем управления статическими преобразователями энергии.

Основу структурной схемы (рисунок 1.1, на примере контроллеров STM32F4xx) образует процессор Cortex-M4 – именно он производит вычислительные операции и управляет обменом информации между всеми элементами, входящими в состав микроконтроллера. Блок управления и синхронизации обеспечивает формирование тактовых импульсов для всех узлов микроконтроллера и формирует сигнал сброса. Внутренняя двунаправленная системная магистраль связывает между собой основные узлы и устройства микроконтроллера: резидентную флэш-память, резидентную память данных, блок прерываний, таймеров, последовательные и параллельные порты ввода/вывода, АЦП и ЦАП. Внутренняя системная магистраль представляет собой совокупность 32-разрядных шин: АНВ1, АНВ2, АНВ3 – высокоскоростные магистрали (с тактовой частотой до 168 МГц), к которым могут подключаться быстрые узлы контроллера (контроллеры USB и Ethernet интерфейсов, память, контроллеры внешней памяти и прямого доступа к памяти), APB1, APB2 – шины периферийных устройств (тактовые частоты для контроллеров STM32F4xx 42 и 84 МГц соответственно), к которым подключаются все остальные периферийные блоки.

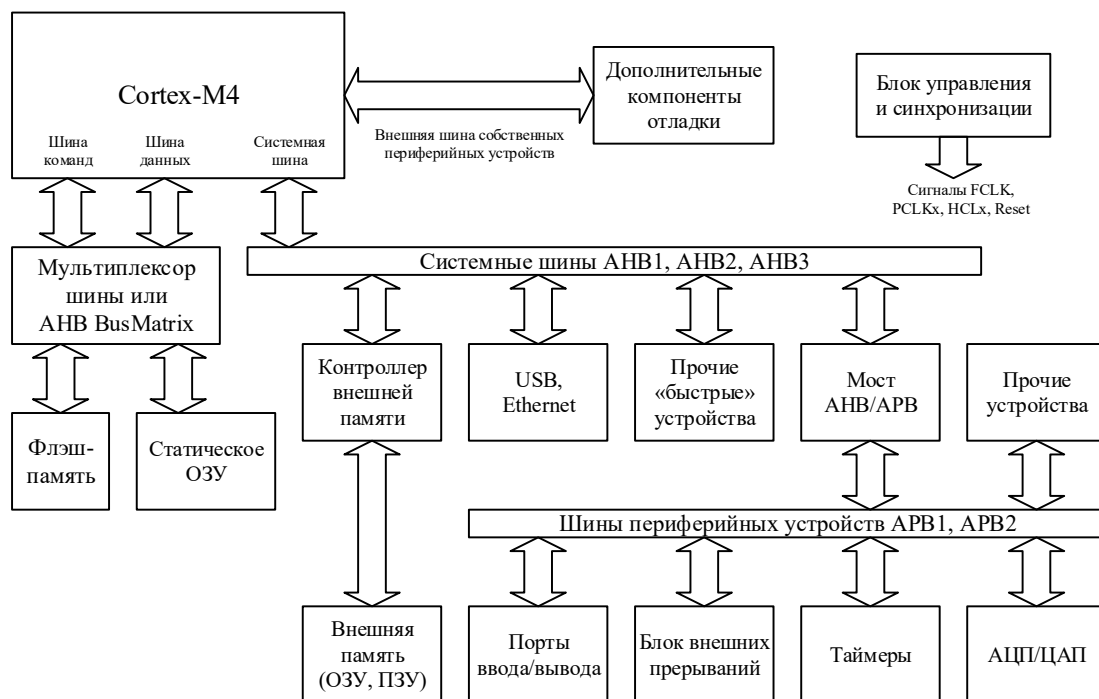


Рисунок 1.1 – Упрощенная структурная схема микроконтроллеров STM32F4xx

Структурная схема учебного стенда приведена на рисунке 1.2.

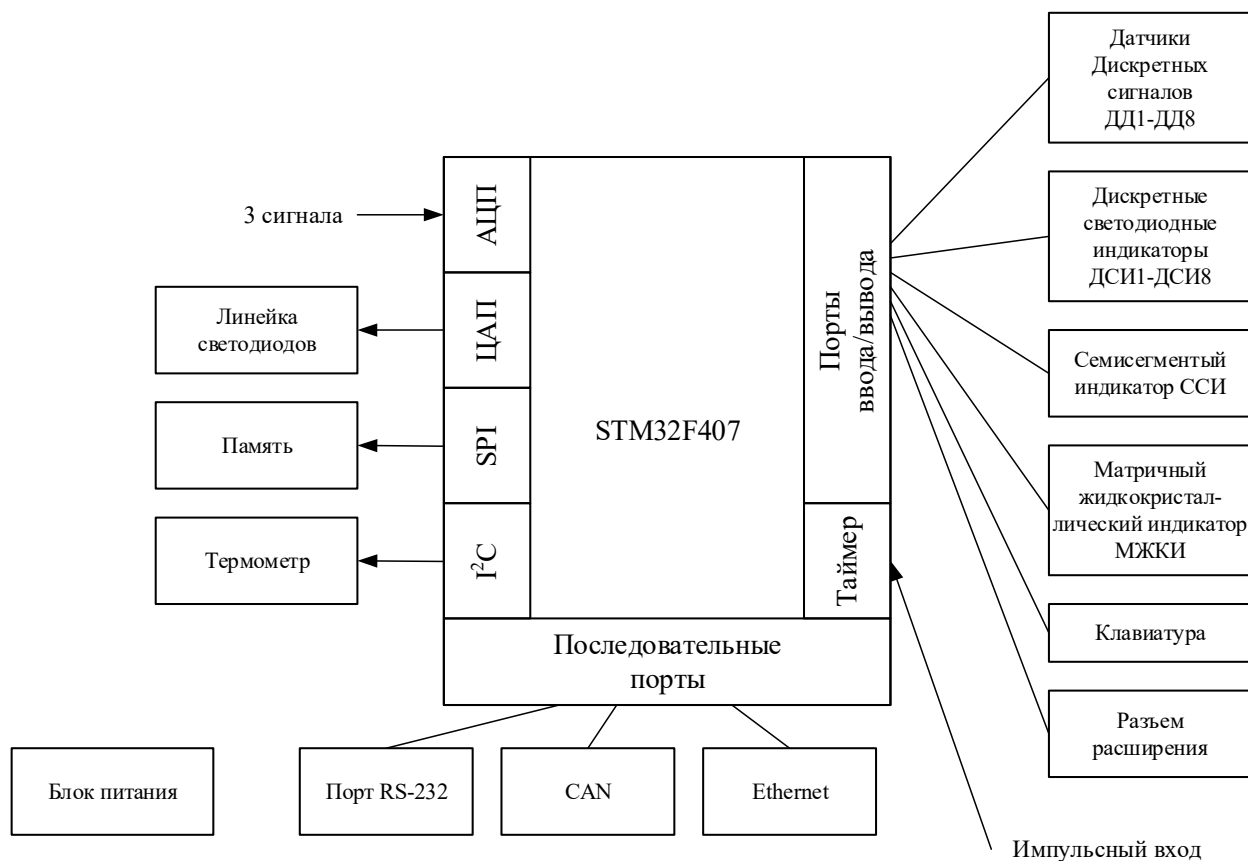


Рисунок 1.2 – Структурная схема учебного стенда

Стенд построен на базе микроконтроллера STM32F407ZET6 семейства ARM Cortex-M4. Контроллер обладает большим количеством встроенных периферийных устройств, поэтому все органы управления и индикации подключены непосредственно к выводам микроконтроллера.

Стенд позволяет исследовать ввод и вывод информации, представленной в различном виде. Так, для исследования вывода используются:

- дискретные светодиодные индикаторы;
- четырехсимвольный семисегментный светодиодный индикатор;
- устройство матричной жидкокристаллической индикации;
- светодиодная линейка.

Для исследования ввода используются:

- датчики дискретных сигналов;
- двенадцатикнопочная клавиатура;
- три аналоговых сигнала.

На плате организован ряд последовательных интерфейсов: I<sup>2</sup>C – используется для обмена информацией с датчиком температуры, SPI – используется для подключения к электрически стираемому ППЗУ, для связи с другими устройствами, например с персональным компьютером. Кроме того, стенд оснащен последовательными портами RS232, CAN и Ethernet.

Кроме того, предусмотрена возможность подключения внешних модулей расширения при помощи разъема.

Для питания платы используется внешний блок питания на 12 В.



### ***Органы управления и индикации учебного стенда СУ-ARM-СМ4.***

Необходимо обратить внимание на то, что учебный стенд выполнен в бескорпусном исполнении – на металлических стойках расположена основная печатная плата. Все органы расположены непосредственно на поверхности печатной платы. Внешний вид учебной платы приведен на рисунке 1.3, расположение элементов поясняется на рисунке 1.4.

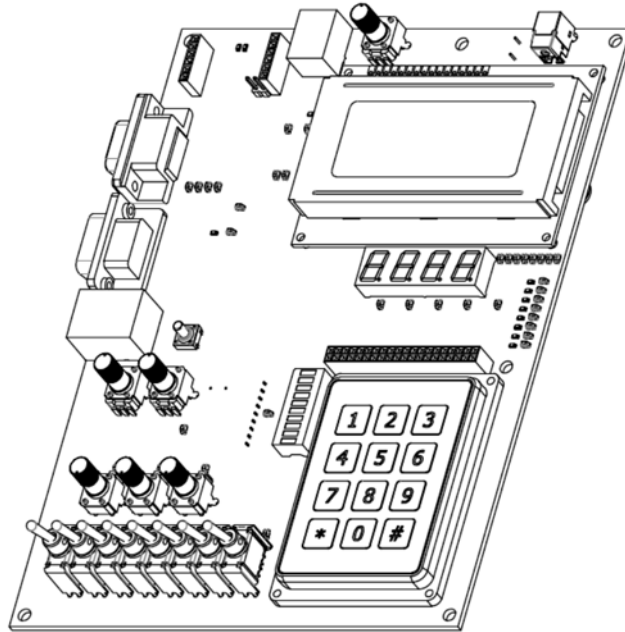


Рисунок 1.3 – Внешний вид платы стенда

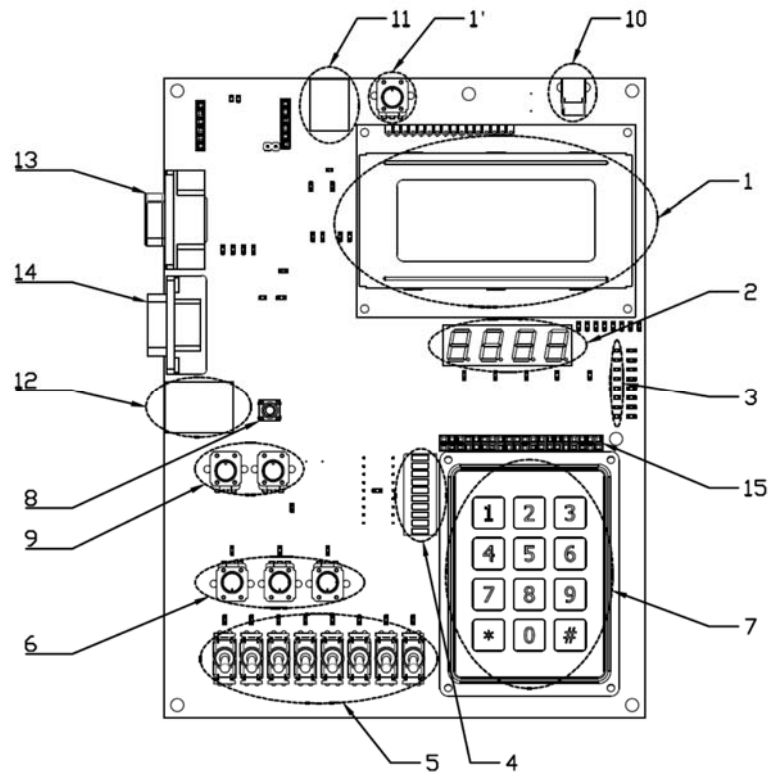


Рисунок 1.4 – Расположение органов на плате

Как видно, на плате расположены:

- матричный жидкокристаллический индикатор МЖКИ (1), контрастностью которого можно управлять при помощи переменного резистора (1');
- четырехсимвольный семисегментный светодиодный индикатор ССИ (2);
- дискретный светодиодный индикатор (набор из восьми светодиодов ДСИ1 ... ДСИ8) (3) (нумерация светодиодов приведена на рисунке 1.5);
- линейка светодиодов – индикатор аналогового сигнала на выходе ЦАП – ЛСИ (4);
- датчики дискретных сигналов (набор из 8 переключателей ДД1...ДД8 (5) (нумерация переключателей приведена на рисунке 1.6);
- имитаторы аналогового сигнала на входах АЦП АД1...АД3 (6) (нумерация аналоговых сигналов приведена на рисунке 1.7 );
- двенадцатикнопочная клавиатура (7);
- кнопка сброса (8);
- для регулирования параметров входного импульсного сигнала на плате установлены резисторы (9).

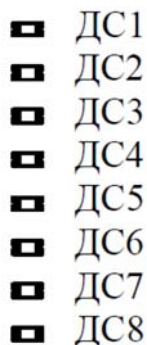


Рисунок 1.5 – Расположение светодиодов ДСИ

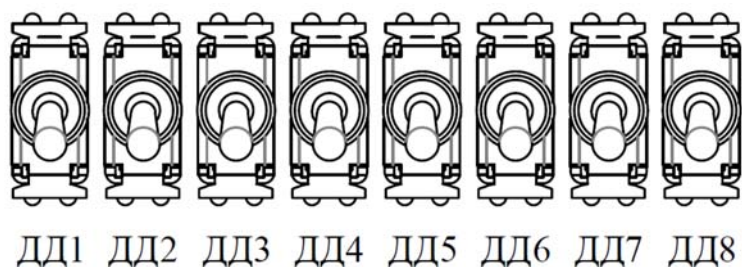


Рисунок 1.6 – Расположение датчиков ДД

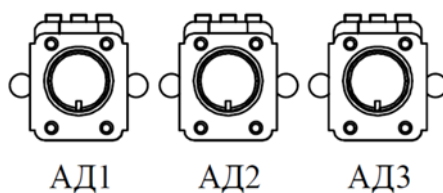


Рисунок 1.7 – Расположение имитаторов аналоговых сигналов

Кроме того, на плате находятся разъемы:

- для подключения питания (10);
- отладки (11);
- Ethernet (12);
- разъем RS-232, предоставляющий возможность подключения стенда к персональному компьютеру (13);
- разъем CAN-интерфейса (14);
- разъем расширения для подключения дополнительных внешних устройств (15).

### ***1.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода работы по пунктам с указанием результата выполнения пункта;
- вывод.

### ***Контрольные вопросы***

1 Перечислите характерные черты архитектуры однокристальных микроконтроллеров.

2 Какие структурные элементы входят в состав микроконтроллеров Cortex-M4, на примере семейства STM32F4xx ?

3 Каковы основные структурные элементы учебного стенда?

4 Опишите органы управления лабораторного стенда.

5 Каков порядок работы со стендом?

## **2 Лабораторная работа № 2. Изучение организации ввода-вывода дискретных сигналов в микропроцессорных устройствах**

**Цель работы:** изучить структуру и особенности портов микроконтроллера, схему подключения входных и выходных сигналов к микроконтроллеру, особенности программирования ввода-вывода дискретных сигналов на языке программирования микроконтроллеров. Составить программу ввода, обработки по заданному алгоритму и вывода дискретных сигналов.

### ***2.1 Ход работы***

2.1.1 Изучить электрическую принципиальную схему к лабораторной работе.

2.1.2 Разработать программу в соответствии с индивидуальным заданием с помощью среды STM32 CubeMX.

2.1.3 Отладить программу в среде Keil uVision 5.

2.1.4 Загрузить программу в учебный стенд.

2.1.5 Исследовать работу датчиков дискретных сигналов и дискретных светодиодных индикаторов в соответствии с индивидуальным заданием.

2.1.6 Оформить отчет по лабораторной работе.

2.1.7 После выполнения работы необходимо ответить на контрольные вопросы.

## **2.2 Краткие теоретические сведения**

На рисунке 2.1 изображена электрическая принципиальная схема к лабораторной работе. В схеме дискретные датчики ДД1. ...8 оформлены в виде набора переключателей SA1. ...SA8, подключенных к младшим 8 разрядам порта F (PF.8...PF.15) микроконтроллера. При этом датчик ДД1 подключен к PF.15, ДД2 к PF.14 и т. д.

Дискретные индикаторы оформлены в виде набора светодиодов ДСИ1...ДСИ8 (VD6, VD9...VD13, VD15, VD18), подключенных к младшим 8 разрядам порта D (PD.0...PD.7) микроконтроллера. Индикатор ДСИ1 при этом подключен к PD.0, ДД2 к PD.1 и т. д.

## **2.3 Варианты индивидуальных заданий к лабораторной работе**

Разработать программу для учебного стенда СУ-МК, позволяющую выполнять следующие действия.

2.3.1 Если ДД1 = 0, то ДСИ1 = 0 и ДСИ2 = 0; иначе, если ДД2 = 0, то ДСИ1 = 1, ДСИ2 = 0, если ДД2 = 1, то ДСИ1 = 0, ДСИ2 = 1.

2.3.2 Если ДД1 = 1 и ДД2 = 0 то ДСИ1 = 0 и ДСИ2 = 0, если ДД1 = 0 и ДД2 = 1, то ДСИ1 = 1, ДСИ2 = 1, если ДД1 = ДД2, то ДСИ1 = 0, ДСИ2 = 1.

2.3.3 Если ДД1 = 1, то ДСИ1 = 0 и ДСИ2 = 0; иначе, если ДД2 = 1, то ДСИ1=1, ДСИ2 = 0, если ДД2 = 0, то ДСИ1 = 0, ДСИ2 = 1.

2.3.4 Если ДД1 = 1 и ДД2 = 1 то ДСИ1 = 0 и ДСИ2 = 0, если ДД1 = 0 и ДД2 = 0, то ДСИ1 = 1, ДСИ2 = 1, если ДД1 ≠ ДД2, то ДСИ1 = 1, ДСИ2 = 0.

2.3.5 Если ДД1 = 1, то ДСИ1 = 0 и ДСИ2 = 0; иначе, по приходу на ДД2 положительного импульса длительностью более 1 с, ДСИ1 = 1 и ДСИ2 = 1.

2.3.6 Если ДД2 = 0, то ДСИ1 = 0 и ДСИ2 = 0; иначе, по приходу на ДД1 отрицательного импульса длительностью более 1,5 с, ДСИ1 = 1 и ДСИ2 = 1.

2.3.7 Если ДД2 = 1, то ДСИ1 = 0 и ДСИ2 = 0; иначе, если ДД1 = 1, то ДСИ1 = 1 и ДСИ2 = 0, если ДД1 = 0, то ДСИ1 = 0 и ДСИ2 = 1; опрос ДД1 организовать с устранением дребезга контактов путем введения временной задержки.

2.3.8 Если ДД1 = 0, то ДСИ1 = 0 и ДСИ2 = 0; иначе, если ДД2 = 1, то ДСИ1 = 0 и ДСИ2 = 1, если ДД2 = 0 то ДСИ1 = 1 и ДСИ2 = 0; опрос ДД2 организовать с устранением дребезга контактов путем подсчета заданного числа совпадений значений сигнала.

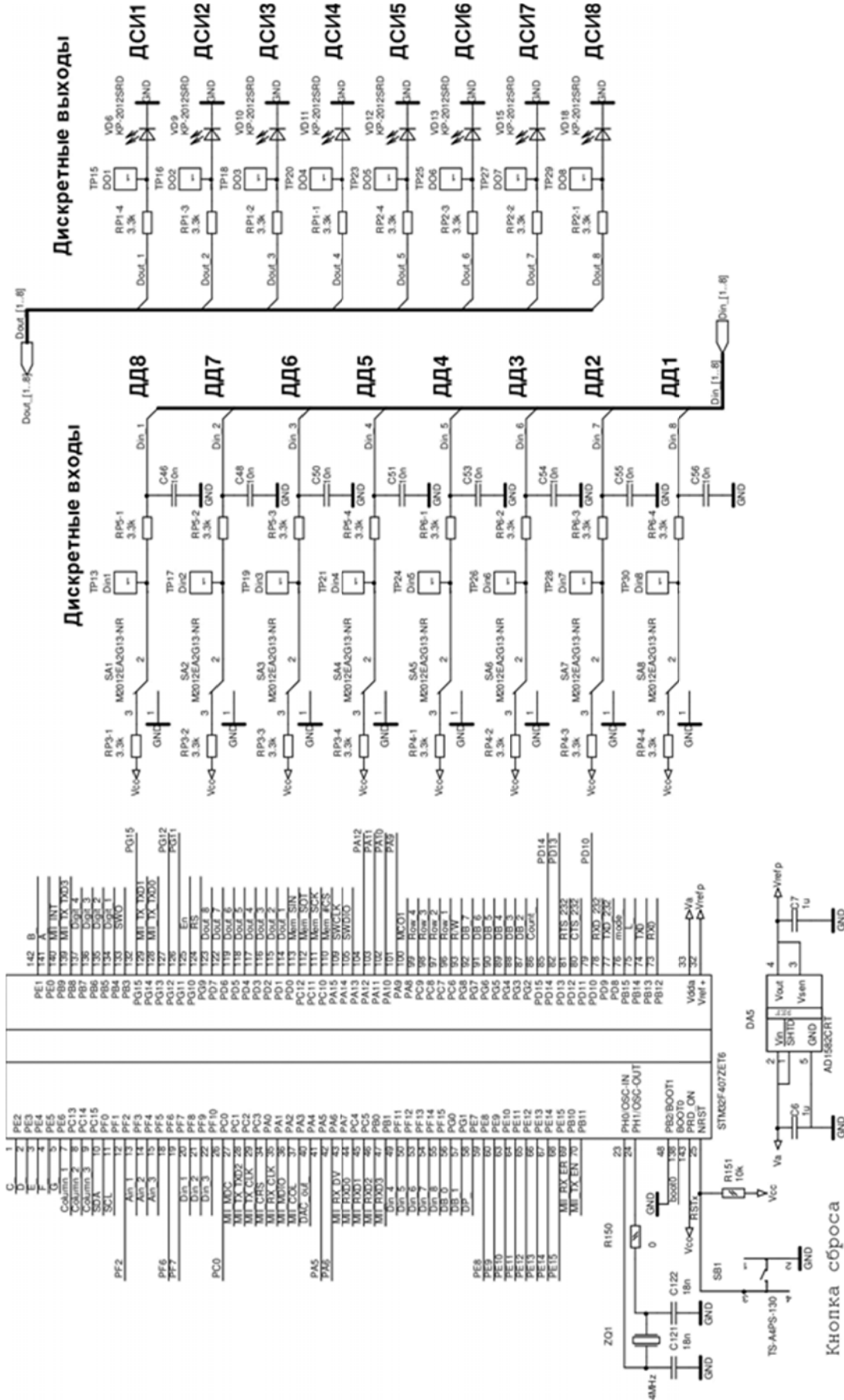


Рисунок 2.1 – Схема электрическая принципиальная к лабораторной работе

2.3.9 Если  $ДД1 = 0$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ ; иначе, если  $ДД2 = 1$ , то  $ДСИ1 = 1$  и  $ДСИ2 = 1$ , если  $ДД2 = 0$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ ; организовать периодический опрос  $ДД2$  с применением таймера; период опроса 10 мс.

2.3.10 Если  $ДД1 = 1$  и  $ДД2 = 1$  то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ , если  $ДД1 = 0$  и  $ДД2 = 0$ , то  $ДСИ1=1$ ,  $ДСИ2 = 1$ , если  $ДД1 \neq ДД2$ , то  $ДСИ1=1$ ,  $ДСИ2 = 0$ ; организовать периодический опрос  $ДД1$  и  $ДД2$  с применением таймера; период опроса – 20 мс.

2.3.11 Если  $ДД2 = 1$ , то  $ДСИ1 = 1$  и  $ДСИ2 = 1$ ; иначе, если  $ДД1 = 1$ , то  $ДСИ1 = 1$  и  $ДСИ2 = 0$ , если  $ДД1 = 0$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 1$ ; опрос  $ДД1$  организовать с устранением дребезга контактов путем введения временной задержки.

2.3.12 Если  $ДД1 = 1$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ ; иначе, если  $ДД2 = 1$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 1$ , если  $ДД2 = 0$ , то  $ДСИ1 = 1$  и  $ДСИ2 = 0$ ; опрос  $ДД2$  организовать с устранением дребезга контактов путем подсчета заданного числа совпадений значений сигнала.

2.3.13 Если  $ДД1 = 0$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ ; иначе, по приходу на  $ДД2$  положительного импульса длительностью менее 3 с,  $ДСИ1 = 1$  и  $ДСИ2 = 1$ .

2.3.14 Если  $ДД2 = 1$ , то  $ДСИ1 = 0$  и  $ДСИ2 = 0$ ; иначе, по приходу на  $ДД1$  отрицательного импульса длительностью более 4,5 с,  $ДСИ1 = 1$  и  $ДСИ2 = 1$ .

2.3.15 Если после прихода на  $ДД1$  положительного импульса длительностью более 3 с на  $ДД2$  придет отрицательный импульс длительностью более 2,5 с,  $ДСИ1 = 1$  и  $ДСИ2 = 1$ , иначе  $ДСИ1 = 0$  и  $ДСИ2 = 0$ .

## 2.4 Пример программы

В качестве примера приведен участок программы с циклом `while(1){}`, в котором осуществляется опрос выводов 8 и 9 порта F. В том случае, если на обоих выводах будет присутствовать уровень логической 1, то будет выдана логическая 1 на линию 1 порта D. В противном случае на линию 1 порта D будет выдан логический 0.

```
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
if ((HAL_GPIO_ReadPin(GPIOF, GPIO_PIN_8) == 1) & (HAL_GPIO_ReadPin(GPIOF, GPIO_PIN_9) == 1))
{
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_1, GPIO_PIN_SET);
}
else
{
    HAL_GPIO_WritePin(GPIOD, GPIO_PIN_1, GPIO_PIN_RESET);
}
}
/* USER CODE END 3 */
```

## **2.5 Содержание отчета**

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание выполнения хода работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### **Контрольные вопросы**

- 1 Как реализован ввод дискретных сигналов в лабораторном стенде?
- 2 Как реализован вывод дискретных сигналов в лабораторном стенде?
- 3 Как подключить тактовую кнопку к микроконтроллеру?
- 4 Включаются ли в разработанной Вами программе потягивающие резисторы портов ввода-вывода микроконтроллера?
- 5 Для чего нужны резисторы, включенные последовательно со светодиодами, в блоке вывода дискретных сигналов?
- 6 Какие функции могут выполнять порты ввода-вывода микроконтроллеров семейства STM32F407?
- 7 Что такое альтернативные функции порта ввода-вывода?
- 8 Какие источники тактового сигнала могут быть у микроконтроллера STM32F407ZET6?
- 9 Каковы особенности настройки системы тактирования шин микроконтроллера STM32F407ZET6?
- 10 Зачем портам ввода-вывода и шинам микроконтроллера нужен тактовый сигнал?

## **3 Лабораторная работа № 3. Изучение организации семисегментной индикации в микропроцессорных устройствах**

**Цель работы:** ознакомиться с документацией на устройство динамической индикации, изучить схему сопряжения микроконтроллера с устройством динамической индикации, разработать и отладить программу вывода информации на устройство динамической индикации.

### **3.1 Ход работы**

- 3.1.1 Изучить электрическую принципиальную схему к лабораторной работе.
- 3.1.2 Разработать программу в соответствии с индивидуальным заданием.
- 3.1.3 Отладить программу в среде Keil uVision5.
- 3.1.4 Загрузить программу в учебный стенд STM32.

3.1.5 Исследовать работу динамической индикации в соответствии с индивидуальным заданием.

3.1.6 Оформить отчет по лабораторной работе.

### 3.2 Краткие теоретические сведения

#### **Устройства цифровой индикации.**

Для отображения цифровой индикации наибольшим распространением пользуются светодиодные семисегментные индикаторы (рисунок 3.1).

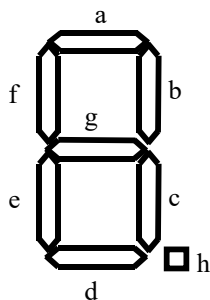


Рисунок 3.1 – Семисегментный символьный индикатор

Сегменты индикатора расположены в виде восьмерки. Иногда добавляют еще один восьмой сегмент – десятичную запятую. Засвечивая группы сегментов, можно получить все цифры и некоторые символы.

Конструктивно индикаторы оформляются в виде светодиодных модулей с общим катодом или с общим анодом (рисунок 3.2).

При построении систем отображения информации различают два подхода: статическая и динамическая индикация.

Статическая индикация состоит в постоянной засветке каждого индикатора от одного источника информации (рисунок 3.3). В такой системе каждый индикатор подключен через свой дешифратор и регистр-защелку к шине данных. Каждый регистр адресуется с помощью устройства дешифрации адреса. Такая схема предполагает значительные аппаратные затраты, т. к. на каждый индикатор необходим по крайней мере один регистр.

Сущность динамической индикации заключается в поочередном циклическом подключении каждого индикатора к источнику данных (рисунок 3.4). При использовании такой схемы включения значительно сокращаются аппаратные затраты. Но при этом необходимо обеспечить достаточное время свечения, для того, чтобы не уменьшалась яркость свечения индикаторов. С другой стороны, необходимо обеспечить достаточно быстрое переключение индикаторов, чтобы не было заметно мерцание.



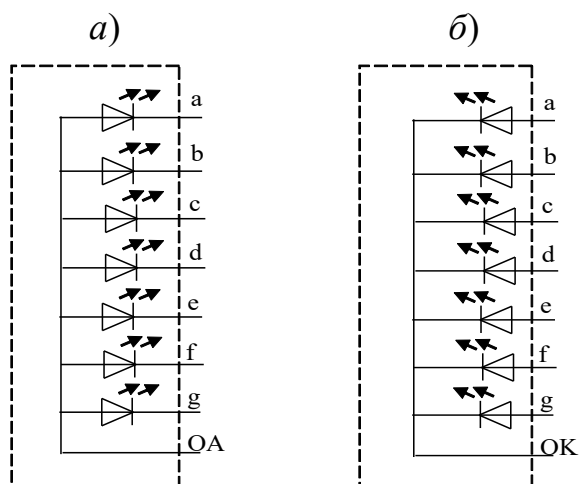
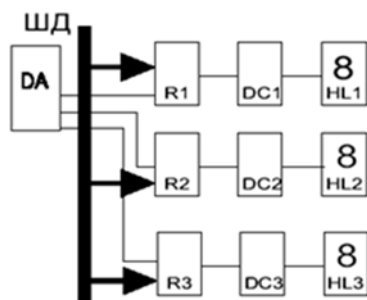
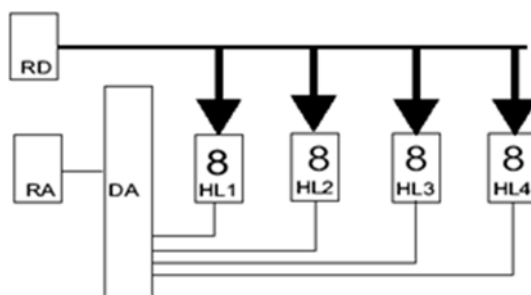


Рисунок 3.2 – Электрические принципиальные схемы семисегментных индикаторов с общим анодом (а) и с общим катодом (б)



DA – дешифратор адреса, необходим для выборки соответствующего регистра; R1-R3 – регистры, в которых временно хранится значение кода числа для отображения (соответствующий регистр выбирается DA); DC1-DC3 – семисегментные дешифраторы, преобразующие двоичный код в семисегментный код; HL1-HL3 – семисегментные индикаторы; ШД – шина данных, по ней осуществляется передача данных на индикацию

Рисунок 3.3 – Устройство статической индикации



RD – регистр данных для временного хранения отображаемого числа либо символа; RA – регистр адреса для временного хранения двоичного кода адреса выбираемого индикатора; DA – для преобразования адреса, задаваемого двоичным кодом в позиционный код; HL1-HL4 – семисегментные индикаторы

Рисунок 3.4 – Устройство динамической индикации

В обоих случаях семисегментные дешифраторы из схемы индикации можно исключить. В этом случае функцию дешифрации можно переложить на микроконтроллер. Это, с одной стороны, несколько усложнит программную реализацию индикации, но при этом, можно выводить на индикацию не только цифры, но и другие символы.

### *Электрическая принципиальная схема к лабораторной работе.*

Электрическая принципиальная схема к лабораторной работе представлена на рисунке 3.5.

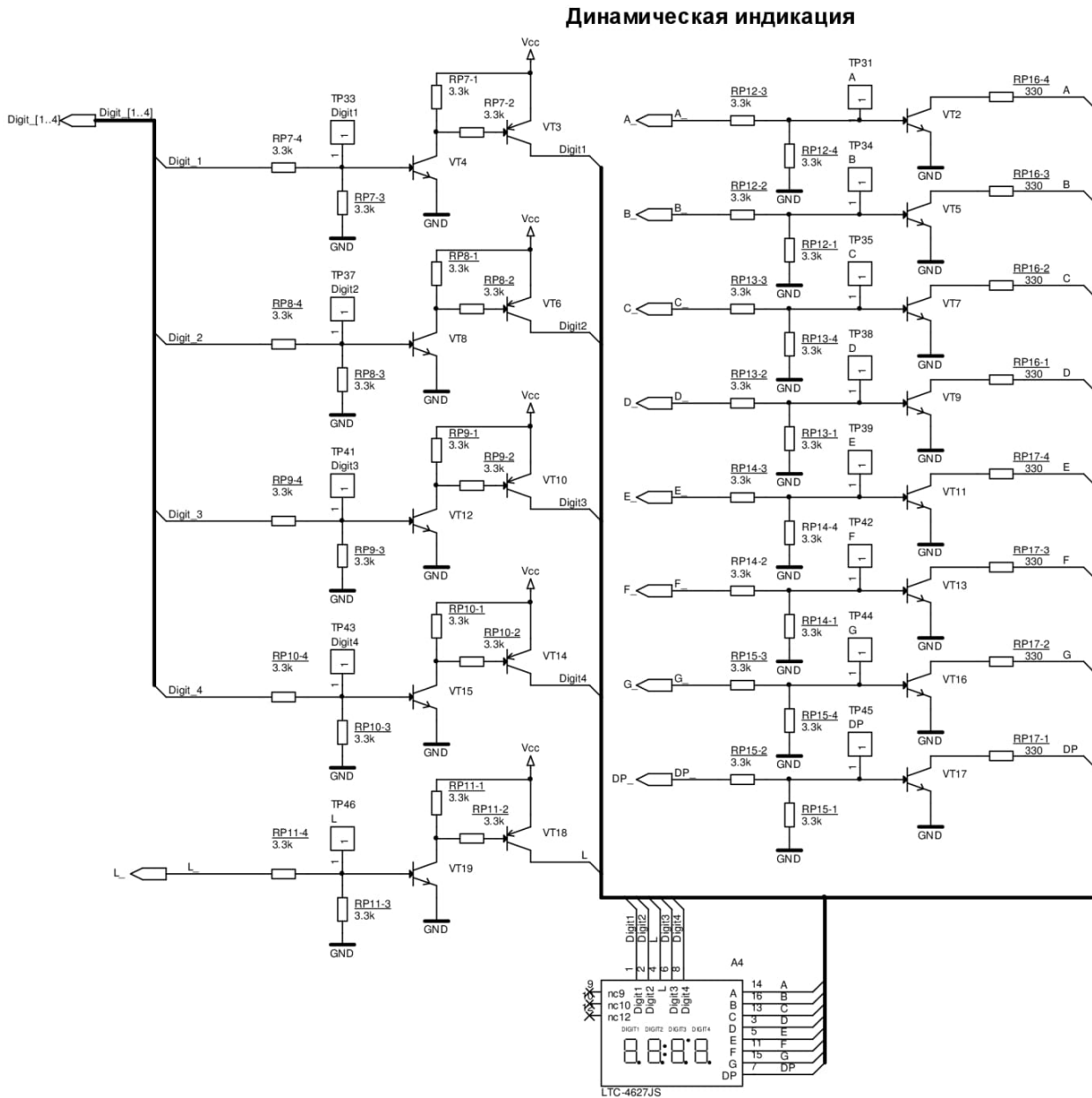


Рисунок 3.5 – Схема электрическая принципиальная к лабораторной работе

В схеме дискретные датчики ДД1...8 оформлены в виде набора переключателей SA1 ...8, подключенных к старшим восьми разрядам порта F (PF.8...PF.15) микроконтроллера. При этом датчик ДД1 подключен к PF.15, ДД2 и PF14 и т. д.

В качестве семисегментного символьного индикатора ССИ А4 используется индикатор LTC-4627JS.

### ***3.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### ***3.4 Варианты индивидуальных заданий к лабораторной работе***

Разработать программу для учебного стенда СУ-МК, позволяющую выводить результат вычислений на светодиодный символьный индикатор (ССИ).

3.4.1 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти их сумму и результат вывести на ССИ в десятичной форме.

3.4.2 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти их разность и результат вывести на ССИ в десятичной форме.

3.4.3 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти их произведение и результат вывести на ССИ в десятичной форме.

3.4.4 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти их частное и результат вывести на ССИ в десятичной форме.

3.4.5 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИЛИ и вывести на ССИ в десятичной форме.

3.4.6 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции И и вывести на ССИ в десятичной форме.

3.4.7 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ и вывести на ССИ в десятичной форме.

3.4.8 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИЛИ и вывести на ССИ в десятичной форме.

3.4.9 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции И и вывести на ССИ в десятичной форме.

3.4.10 Рассматривая состояние дискретных датчиков ДД3...ДД6 и ДД7...ДД10 как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ и вывести на ССИ в десятичной форме.

3.4.11 Найти количество включенных ДД, умножить это количество на 3 и вывести результат на ССИ в десятичной форме.

3.4.12 Найти количество выключенных ДД, умножить это количество на 2 и вывести результат на ССИ в десятичной форме.

3.4.13 Найти количество включенных ДД, умножить это количество на 4 и вычесть количество выключенных ДД, вывести результат на ССИ в десятичной форме.

3.4.14 Отобразить состояние ДД на ССИ поразрядно, включенный ДД, отображая единицей в соответствующем разряде ДД, выключенный – нулем.

3.4.15 Отобразить инверсное состояние ДД на ССИ поразрядно, включенный ДД, отображая нулем в соответствующем разряде ДД, выключенный – единицей.

### ***3.5 Пример реализации динамической индикации***

```
while (1)
```

```
{
```

```
/* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 3 */
```

```
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_4, GPIO_PIN_SET);
HAL_Delay(2);
```

```
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
```

```

HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_SET);
HAL_Delay(2);

```

```

HAL_GPIO_WritePin(GPIOB, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
HAL_Delay(2);

```

```

HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_0, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_1, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_2, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_3, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_4, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_5, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_6, GPIO_PIN_SET);
HAL_GPIO_WritePin(GPIOE, GPIO_PIN_7, GPIO_PIN_RESET);
HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, GPIO_PIN_SET);
HAL_Delay(2);

```

```

}
/* USER CODE END 3 */

```

### ***Контрольные вопросы***

- 1 Как организуется вывод цифровой информации в микроконтроллерных системах?
- 2 В чем принцип действия семисегментных светодиодных индикаторов?
- 3 Статическая индикация.
- 4 Динамическая индикация.
- 5 Особенности программной реализации динамической индикации.

## 4 Лабораторная работа № 4. Изучение организации жидкокристаллической индикации в микропроцессорных устройствах

**Цель работы:** изучить особенности программной и аппаратной реализации жидкокристаллической индикации с помощью языка программирования С, составить программу заданной временной функции, перевести ее в машинные коды, записать в память программ микроконтроллера и выполнить.

### 4.1 Ход работы

4.1.1 Изучить электрическую принципиальную схему к лабораторной работе.

4.1.2 Разработать программу в соответствии с индивидуальным заданием.

4.1.3 Отладить программу в среде uVision.

4.1.4 Загрузить программу в учебный стенд.

4.1.5 Исследовать программную и аппаратную реализацию жидкокристаллической индикации, режимы работы индикаторов и порядок программирования микроконтроллера в соответствии с индивидуальным заданием.

4.1.6 Оформить отчет по лабораторной работе.

### 4.2 Краткие теоретические сведения

#### ***Жидкокристаллический индикатор Volumin BC1604A-GPLCW.***

Жидкокристаллический модуль BC1604A представляет собой алфавитно-цифровой дисплей. Тип сегментов – матричный. Алфавитно-цифровые (или символьные) индикаторы в своем составе обязательно имеют знакогенератор, который при поступлении соответствующей команды формирует те или иные последовательности микрокоманд, обеспечивающие воспроизведение на экране дисплея выбранных стандартных символов (цифры, буквы и знаки). Кроме того, система команд модуля обеспечивает возможность программирования пользователем до восьми символов в дополнение к стандартным. В состав модуля входит контроллер KS0066 (или совместимый с ним, например HD44780), который может потенциально управлять двумя строками по 40 символов в каждой (для модулей с четырьмя строками используются два однотипных контроллера).

На рисунке 4.1 изображена электрическая принципиальная схема к лабораторной работе.

В схеме дискретные датчики ДД1...ДД8 оформлены в виде набора переключателей SA1...SA8, подключенных к старшим восьми разрядам порта F (PF.8...PF.15) микроконтроллера.



В качестве матричного жидкокристаллического индикатора А1 используется индикатор BC1604A-GPLCW. Индикатор подключен к портам микроконтроллера: 8-битная шина данных к выводам порта G (PG.0...PG.7), сигналы управления к выводам PG.8...PG10 (R/W, RS и E соответственно).

### ***4.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### ***Контрольные вопросы***

- 1 Какова физическая сущность работы жидкокристаллических индикаторов?
- 2 Какова процедура инициализации индикатора?
- 3 Какие символы можно выводить на знакосинтезирующий индикатор?
- 4 Как осуществляется установка курсора в заданное положение?
- 5 Как вывести символ на индикатор?
- 6 Как вывести строку символов на индикатор?

## **5 Лабораторная работа № 5. Изучение организации клавиатуры в микропроцессорных устройствах**

**Цель работы:** изучить особенности программной и аппаратной реализации опроса матричной клавиатуры с помощью языка программирования C, составить программу, перевести ее в машинные коды, записать в память программ микроконтроллера и выполнить.

### ***5.1 Ход работы***

- 5.1.1 Изучить электрическую принципиальную схему к лабораторной работе.
- 5.1.2 Разработать программу в соответствии с индивидуальным заданием.
- 5.1.3 Отладить программу в среде uVision.
- 5.1.4 Загрузить программу в учебный стенд.
- 5.1.5 Исследовать программную и аппаратную реализацию ввода информации с помощью клавиатуры и порядок программирования микроконтроллера в соответствии с индивидуальным заданием.
- 5.1.6 Оформить отчет по лабораторной работе.



## 5.2 Краткие теоретические сведения

Клавиатура предназначена для ввода информации в микроконтроллерное устройство, т. е. для организации интерфейса с человеком. Фактически, клавиатуры представляют собой наборы дискретных переключателей. Особенностью клавиатур является достаточно большое количество таких переключателей (обычно более 10). Самым простым способом организации клавиатур является линейный (рисунок 5.1). При таком построении клавиатуры сигнал от каждой кнопки вводится в систему через устройства ввода информации (порт ввода/вывода) по отдельной линии. Недостаток такого способа заключается в том, что большое количество переключателей требует использования большого количества линий параллельных портов.

Одним из наиболее распространенных способов уменьшения количества требуемых линий для подключения клавиатур к параллельным портам является организация клавиатур по принципу матричного шифратора, в узлах которого размещены коммутационные элементы – клавиши (рисунок 5.2).

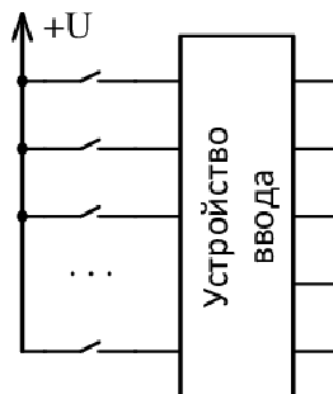


Рисунок 5.1 – Линейная организация клавиатур

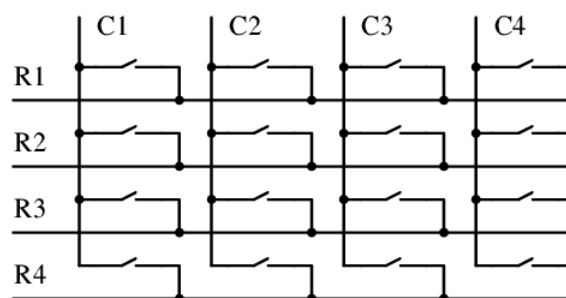


Рисунок 5.2 – Матричная организация клавиатур

Из рисунка 5.2 видно, что шестнадцатиклавишную клавиатуру можно подключить, используя только восемь линий параллельного порта.

Работает матричная клавиатура следующим образом: на одну из линий R1 – R4 подаются сигналы, т. е. происходит сканирование одного из рядов клавиатуры. Если ни одна клавиша не нажата, то сигнал на линиях C1 – C4 отсутствует. Если клавиша

в сканирующем ряду нажата, то на соответствующей линии C1...C4 появляется поданный на линию Ri сигнал. Таким образом, зная, какой ряд в данный момент сканируется и на какой из линий C1...C4 появится сигнал, можно определить, какая клавиша нажата в данный момент.

### 5.3 Электрическая принципиальная схема к лабораторной работе

На рисунке 5.3 изображена электрическая принципиальная схема к лабораторной работе.

Клавиатура организована по матричному принципу и выполнена в виде клавиатурного блока А3 (96АВ-102F). Строки клавиатуры подключены к выводам 6...9 порта С (РС.6...РС.9) микроконтроллера, столбцы – к выводам 13...15 порта С.



### LCD дисплей

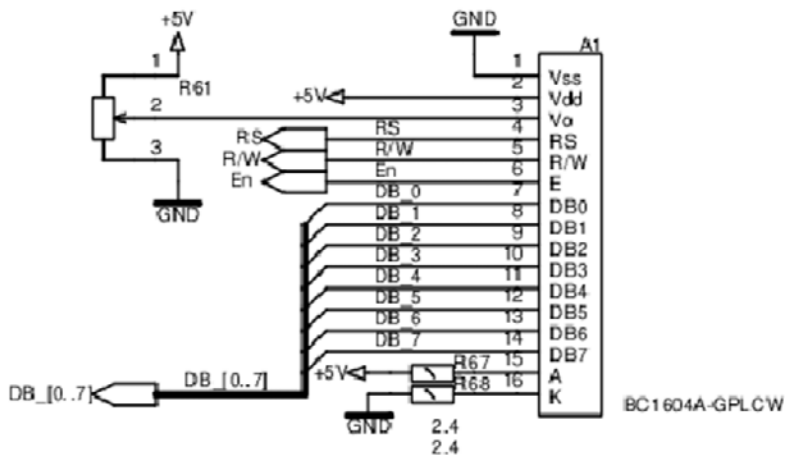


Рисунок 5.3 – Подключение матричной клавиатуры и ЖКИ

### ***5.4 Варианты индивидуальных заданий к лабораторной работе***

Разработать программу для учебного стенда, позволяющую вводить данные с клавиатуры, выполнять определенные действия над данными и выводить поясняющие надписи и результат вычислений на индикатор. Клавишу «#» использовать как подтверждение ввода, а клавишу «\*» - как сброс операции.

5.4.1 Ввести последовательно два трехзначных числа, найти их разность и вывести результат на МЖКИ.

5.4.2 Ввести последовательно два трехзначных числа, найти их сумму и вывести результат на МЖКИ.

5.4.3 Ввести последовательно два трехзначных числа, найти их произведение и вывести результат на МЖКИ.

5.4.4 Ввести последовательно два трехзначных числа, найти их частное и вывести результат на МЖКИ.

5.4.5 Ввести последовательно два трехзначных числа, найти результат поразрядной операции «ИЛИ» и вывести его на МЖКИ.

5.4.6 Ввести последовательно два трехзначных числа, найти результат поразрядной операции «И» и вывести его на МЖКИ.

### ***5.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### ***Контрольные вопросы***

- 1 Для чего используются клавиатуры?
- 2 По какому принципу чаще всего организуют клавиатуры?
- 3 Как реализуется опрос матричной клавиатуры?

## **6 Лабораторная работа № 6. Изучение организации устройств вывода аналоговой информации в микропроцессорных устройствах**

**Цель работы:** ознакомиться с техническими характеристиками и принципом действия АЦП, изучить схему ввода аналоговых сигналов в микроконтроллер, разработать и отладить программу для сбора аналоговой информации.

## **6.1 *Ход работы***

- 6.1.1 Изучить электрическую принципиальную схему к лабораторной работе.
- 6.1.2 Разработать программу в соответствии с индивидуальным заданием.
- 6.1.3 Отладить программу в среде uVision.
- 6.1.4 Загрузить программу в учебный стенд.
- 6.1.5 Исследовать работу аналогово-цифрового преобразователя в соответствии с индивидуальным заданием.
- 6.1.6 Оформить отчет по лабораторной работе.

## **6.2 *Краткие теоретические сведения***

Аналогово-цифровой преобразователь (АЦП) – один из самых важных электронных компонентов в измерительном и тестовом оборудовании. АЦП преобразует напряжение (аналоговый сигнал) в код, над которым микропроцессор и программное обеспечение выполняют определенные действия. Даже если Вы работаете только с цифровыми сигналами, скорее всего Вы используете АЦП в составе осциллографа, чтобы узнать их аналоговые характеристики.

Существует несколько основных типов архитектуры АЦП, отличающихся по сложности реализации, быстродействию, помехоустойчивости, хотя в пределах каждого типа существует также множество вариаций. Различные типы измерительного оборудования используют различные типы АЦП. Например, в цифровом осциллографе используется высокая частота дискретизации, но не требуется высокое разрешение. В цифровых мультиметрах нужно большее разрешение, но можно пожертвовать скоростью измерения.

Системы сбора данных общего назначения по скорости дискретизации и разрешающей способности обычно занимают место между осциллографами и цифровыми мультиметрами. В оборудовании такого типа используются АЦП последовательного приближения либо сигма-дельта АЦП. В системах, где основным критерием является быстродействие, применяют АЦП параллельного преобразования. Однако преобразователи этого типа сложны в реализации. Для  $n$ -разрядного АЦП необходимо  $2^{n-1}$  компараторов и параллельный делитель напряжения, который вырабатывает  $2^{n-1}$  уровней квантования. Кроме того, существуют также интегрирующие АЦП с высокими разрешением и помехоподавлением. Такой АЦП состоит из двух преобразователей. Измеряемое напряжение преобразуется в длительность импульса, а потом длительность импульса преобразуется в цифровой код.

Как видно из рисунка 6.1, блок содержит в своем составе три АЦП и температурный сенсор. В качестве источника питания используется напряжение

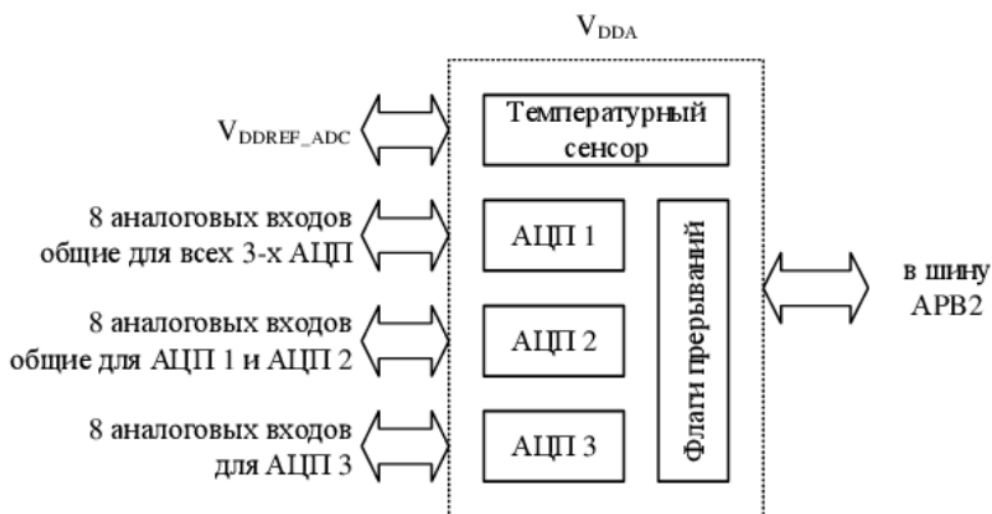


Рисунок 6.1 – Структурная схема блока ввода аналоговых сигналов

**Включение/отключение АЦП.** АЦП включается путем установки бита ADON в регистре ADC\_CR2. Когда бит устанавливается в первый раз, он выводит АЦП из режима Power-down (выключен). Преобразование запускается, когда устанавливается бит SWSTART или JSWSTART. Остановить преобразование и перевести АЦП в режим Power-down можно очистив бит ADON. В этом режиме АЦП почти не потребляет энергии (только несколько микроампер).

**Тактирование АЦП.** Если рассмотреть структурную схему контроллера STM32F407, приведенную в документации на него, легко увидеть, что АЦП подключен к шине APB2, которая и является основным источником тактовых импульсов для АЦП. АЦП имеет две схемы тактирования:

1) для аналоговой части – источником является сигнал ADCCLK, общий для всех АЦП; этот источник формируется на основании тактового сигнала APB2 уменьшенного программируемым предделителем, позволяющим АЦП работать на частотах  $f_{\text{clk2}}/2, /4, /6$  или  $/8$ . Максимальное значение частоты сигнала ADCCLK для контроллера STM32F407 составляет 36 МГц, номинальное – 30 МГц;

2) для цифровой части (используется для доступа к регистрам) – этот тактовый сигнал равен частоте шины APB2. Тактовый сигнал цифрового интерфейса может быть разрешается/запрещается индивидуально для каждого АЦП при помощи регистра RCC\_APB2ENR. Как видно из рисунка, для включения тактовый сигналов АЦП 1/2/3 используются биты 8...10 этого регистра.

**Выбор канала.** У АЦП 16 мультиплексированных каналов. Каналы АЦП микроконтроллеров STM32 делятся на две группы: обычные каналы (*regular*) и инжектированные (*injected*). Преобразования можно объединить в две группы: обычные и инжектированные. Группа состоит из последовательности преобразований, которые можно выполнить на любом канале и в любом порядке. Например, можно реализовать преобразование последовательности в следующем порядке: ADC\_IN3, ADC\_IN8, ADC\_IN2, ADC\_IN2, ADC\_IN0, ADC\_IN2, ADC\_IN2, ADC\_IN15.

*Обычная группа* состоит из 16 преобразований. Обычные каналы и их порядок в последовательности преобразования должны быть выбраны в регистре ADC\_SQRx. Общее количество преобразований в обычной группе должно быть записано в битах L[3:0] регистра ADC\_SQR1. Результат преобразования таких каналов сохраняется в регистре ADC\_DR.

*Инжектированная группа* состоит из четырех преобразований. Инжектированные каналы и их порядок в последовательности преобразования должны быть выбраны в регистре ADC\_ISQR. Общее количество преобразований в инжектированной группе должно быть записано в битах L[1:0] регистра ADC\_ISQR. Результат преобразования таких каналов сохраняется в соответствующем регистре ADC\_JDRx (где x=1..4).

Если регистры ADC\_SQRx или ADC\_ISQR будут изменены во время преобразования, текущее преобразование сбрасывается и посылается новый импульс запуска на АЦП для преобразования вновь выбранной группы.

**Режим одиночного преобразования.** В этом режиме АЦП выполняет одно преобразование. Этот режим начинается с битом CONT в 0 с помощью:

- установки бита SWSTART в регистре ADC\_CR2 (для обычного канала);
- установки бита ISWSTART (для инжектированного канала);
- внешнего триггера (для обычного или инжектированного канала).

После окончания преобразования выбранного канала:

а) если был преобразован обычный канал:

- преобразованные данные сохраняются в 16-битном ADC\_DR регистре;
- устанавливается флаг EOC (конец преобразования);
- если бит EOC IE установлен, генерируется прерывание;

б) если был преобразован инжектированный канал:

- преобразованные данные сохраняются в 16-битном регистре ADC\_IDR1;
- устанавливается флаг IEОС (конец инжектированного преобразования);
- если бит IEОСIE установлен, генерируется прерывание.

После чего АЦП останавливается.

**Режим непрерывного преобразования.** В этом режиме АЦП начинает новое преобразование, как только он заканчивает текущее.

Этот режим начинается с битом CONT в 1 с помощью внешнего триггера или установкой бита SWSTRT в регистре ADC\_CR2 (только для обычных каналов). После каждого преобразования:

а) если была преобразована обычная группа каналов:

- последние преобразованные данные сохраняются в 16-битном регистре ADC\_DR;
- устанавливается флаг EOC (конец преобразования).

б) если бит EOC IE установлен, генерируется прерывание.

Инжектированные каналы не могут быть преобразованы непрерывно. Единственным исключением является ситуация, когда инжектированный канал сконфигурирован на автоматическое преобразование после обычных каналов в непрерывном режиме.

**Режим сканирования.** Этот режим используется для сканирования группы аналоговых каналов.

Режим сканирования выбирается путем установки бита SCAN в регистре ADC\_CR1. После того, как этот бит был установлен, АЦП проверяет все каналы, выбранные в регистрах ADC\_SQRx (для обычных каналов) или в регистре ADC\_ISQR (для инжектированных каналов). Одиночное преобразование выполняется для каждого канала из группы. После окончания преобразования следующий канал в группе автоматически преобразуется. Если бит CONT установлен, преобразование обычного канала не останавливается на последнем выбранном канале из группы, но продолжится снова с первого выбранного канала.

Если бит DMA установлен, используется контроллер прямого доступа к памяти (DMA) для передачи преобразованных данных обычной группы каналов (сохраненных в регистре ADC\_DR) в статическое ОЗУ после каждого преобразования обычного канала.

Бит EOC в регистре ADC\_SR устанавливается:

- в конце каждой последовательности обычной группы, если бит EOCs сброшен в 0;
- в конце каждого преобразования обычного канала, если бит EOCs установлен в 1.

Данные, преобразованные из инжектированного канала, всегда сохраняются в регистрах ADC\_IDRx.

**Управление инжектированными каналами.** Возможны два варианта инжектирования.

*Инжектирование по триггеру.* Для того, чтобы использовать инжектирование по триггеру, должен быть сброшен бит IAUTO в регистре ADC\_CR1:

- необходимо запустить преобразование группы обычных каналов с использованием либо внешнего сигнала либо установкой бита SWSTART в регистре ADC\_CR2;
- если появляется внешний сигнал запуска инжектированных каналов или во время преобразования группы обычных каналов будет установлен бит ISWSTART, текущее преобразование сбрасывается и последовательность инжектированных каналов переключается в режим одиночного сканирования;
- после окончания этого процесса продолжается преобразование обычных каналов с прерванного канала.

*Автоинжектирование.* Если бит IAUTO сброшен, то каналы в инжектированной группе автоматически преобразуются после группы обычных каналов. Эту особенность можно использовать для преобразования последовательности из 20 преобразований запрограммированной в регистрах ADC\_SQRx и ADC\_ISQR. В этом режиме внешний триггер для инжектированных каналов должен быть выключен. Если бит CONT также установлен в дополнение к биту IAUTO, обычные каналы непрерывно преобразуются следом за инжектированными каналами.

### ***Прерываемый режим преобразования.***

Обычная группа.

Этот режим разрешается путем установки бита DISCEN в регистре ADC\_CR1. Он может быть использован для преобразования короткой последовательности из  $n$  преобразований ( $n < 8$ ), которая является частью последовательности преобразований, выбранной при помощи регистров ADC\_SQRx. Значение  $n$  задается путем записи битов DISCNUM[2:0] в регистре ADC\_CR1.

Когда появляется внешний сигнал запуска, он запускает  $n$  преобразований, выбранных в регистрах ADC\_SQRx, пока все преобразования из последовательности не будут выполнены. Общая длина последовательности определяется битами L[3:0] в регистре ADC\_SQR1.

Например:

- $n = 3$ , преобразуемые каналы = 0,1,2,3,6, 7, 9, 10;
- первый сигнал: преобразуемая последовательность 0, 1, 2. Событие EOC (конец преобразования) генерируется после каждого преобразования;
- второй сигнал: преобразуемая последовательность 3, 6, 7. Событие EOC (конец преобразования) генерируется после каждого преобразования;
- третий сигнал: преобразуемая последовательность 9, 10. Событие EOC (конец преобразования) генерируется после каждого преобразования;
- четвертый сигнал: преобразуемая последовательность 0, 1,2. Событие EOC (конец преобразования) генерируется после каждого преобразования.

Инжектированная группа.

Этот режим разрешается путем установки бита IDISCEN в регистре ADC\_CR1. Он может быть использован для преобразования последовательности, указанной в регистре ADC\_ISQR, канал за каналом, после внешнего сигнала синхронизации.

Когда появляется внешний сигнал запуска, он запускает преобразование следующего канала, выбранного в регистре ADC\_ISQR, пока все преобразования из последовательности не будут выполнены. Общая длина последовательности определяется битами JL[3:0] в регистре ADC\_ISQR.

Например:

- $n = 1$ , преобразуемые каналы = 1,2,3;
- первый сигнал: преобразуется канал 1;
- второй сигнал: преобразуется канал 2;
- третий сигнал: преобразуется канал 3 и генерируется событие IEOC (конец преобразования);
- четвертый сигнал: преобразуется канал 1, 2.

### ***6.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.



#### **6.4 Варианты индивидуальных заданий к лабораторной работе**

Разработать программу для учебного стенда, позволяющую вводить данные с датчиков аналоговых величин АД1 – АД3, выполнять определенные действия над данными, и выводить поясняющие надписи и результат вычислений на матричный жидкокристаллический индикатор (МЖКИ) или на символный светодиодный индикатор (ССИ).

6.4.1 Ввести данные с аналогового датчика АД1. Привести код, считываемый с АЦП, к значениям измеряемой величины. Минимальное значение измеряемой величины 5 соответствует коду АЦП 0, максимальное значение измеряемой величины 516 соответствует коду АЦП 4095. Вывести результат на ССИ.

6.4.2 Ввести данные с аналогового датчика АД 1. Выполнить усреднение по пяти результатам эксперимента. Вывести результат на ССИ.

6.4.3 Ввести данные с аналогового датчика АД 1. Привести код, считываемый с АЦП, к значениям измеряемой величины. Минимальное значение измеряемой величины – 300, соответствует коду АЦП 0; максимальное значение измеряемой величины – 300, соответствует коду АЦП 4095. Вывести результат на МЖКИ.

6.4.4 Ввести данные с аналогового датчика АД 1. Вывести на МЖКИ состояние аналоговой величины: при превышении  $2/3$  максимального значения измеряемой величины – больше нормы; при уровне измеряемого сигнала менее  $1/3$  максимального значения измеряемой величины – меньше нормы; в остальных случаях – норма.

6.4.5 Ввести данные с аналоговых датчиков АД2 и АД3. Вывести на МЖКИ предупреждение, если  $АД2 < АД3$ .

6.4.6 Ввести данные с аналоговых датчиков АД1, АД2 и АД3. Вывести на ССИ предупреждение, если  $АД2 - АД3$  меньше АД1.

6.4.7 Ввести данные с аналогового датчика АД 1. Вывести на ССИ состояние аналоговой величины: при превышении  $2/3$  максимального значения измеряемой величины – больше нормы; при уровне измеряемого сигнала менее  $1/3$  максимального значения измеряемой величины – меньше нормы; в остальных случаях – норма.

6.4.8 Ввести данные с аналоговых датчиков АД2 и АД3. Вывести на ССИ предупреждение, если  $АД2 > АД3$ .

6.4.9 Ввести данные с аналогового датчика АД 1. Выполнить усреднение по 5 результатам эксперимента. Вывести результат на МЖКИ.

6.4.10 Ввести данные с аналоговых датчиков АД2 и АД3. Найти сумму показаний. Вывести результат на ССИ.

6.4.11 Ввести данные с аналоговых датчиков АД2 и АД3. Найти сумму показаний. Вывести результат на МЖКИ.

6.4.12 Ввести данные с аналоговых датчиков АД2 и АД3. Вывести на ССИ предупреждение, если  $АД2 - АД3$  больше максимального значения измеряемых величин.

6.4.13 Ввести данные с аналоговых датчиков АД2 и АД3. Вывести на МЖКИ предупреждение, если АД2 – АД3 меньше максимального значения измеряемых величин.

6.4.14 Ввести данные с аналоговых датчиков АД1, АД2 и АД3. Вывести на МЖКИ предупреждение, если АД2 и АД3 больше АД1.

6.4.15 Ввести данные с аналоговых датчиков АД1, АД2 и АД3. Вывести на МЖКИ результат выполнения операции  $(АД2 + АД3 - АД1) / 2$ . Если результат меньше 0, вывести 0, дробную часть результата отбросить.

### ***Контрольные вопросы***

- 1 Какие схемы АЦП применяют в быстродействующих системах?
- 2 Какие схемы АЦП применяют для повышения помехоустойчивости?
- 3 Как работают АЦП, построенные по схеме сравнения?
- 4 Из каких структурных элементов состоит АЦП, встроенный в микроконтроллер STM32F407?
- 5 Каковы основные параметры АЦП, встроенного в микроконтроллер STM32F407?
- 6 В каких режимах может работать АЦП, встроенный в микроконтроллер STM32F407?
- 7 Каким образом можно изменить режим работы АЦП ?
- 8 На какие группы делятся каналы АЦП и чем они отличаются ?

## **7 Лабораторная работа № 7. Разработка и отладка программ управления технологическими объектами в реальном времени (часть 1)**

***Цель работы:*** изучить методику разработки и отладки программ для микроконтроллера для управления технологическим объектами в реальном времени.

### ***7.1 Ход работы***

- 7.1.1 Изучить понятие систем мягкого и жесткого реального времени.
- 7.1.2 Изучить порядок работы с отладчиком в IDE  $\mu$ Vision 5.
- 7.1.3 Выполнить программу в соответствии с заданием в режиме пошаговой отладки.
- 7.1.4 Оформить отчет о выполнении работы.
- 7.1.5 Ответить на контрольные вопросы.

## 7.2 Краткие теоретические сведения

Системы жёсткого и мягкого реального времени.

Операционные системы реального времени иногда делят на два типа – системы жёсткого реального времени и системы мягкого реального времени.

Операционная система, которая может обеспечить требуемое время выполнения задачи реального времени даже в худших случаях, называется операционной системой жёсткого реального времени. Система, которая может обеспечить требуемое время выполнения задачи реального времени в среднем, называется операционной системой мягкого реального времени.

Системы жёсткого реального времени не допускают задержек реакции системы, т. к. это может привести к потере актуальности результатов, большим финансовым потерям или даже авариям и катастрофам. Ситуация, в которой обработка событий происходит за время, большее предусмотренного, в системе жёсткого реального времени считается фатальной ошибкой. При возникновении такой ситуации операционная система прерывает операцию и блокирует её, чтобы, насколько возможно, не пострадала надёжность и готовность остальной части системы. Примерами систем жёсткого реального времени могут быть бортовые системы управления (на самолёте, космическом аппарате, корабле и пр.), системы аварийной защиты, регистраторы аварийных событий.

В системе мягкого реального времени задержка реакции считается восстановимой ошибкой, которая может привести к увеличению стоимости результатов и снижению производительности, но не является фатальной. Примером может служить работа компьютерной сети. Если система не успела обработать очередной принятый пакет, это приведёт к остановке на передающей стороне и повторной посылке (в зависимости от протокола). Данные при этом не теряются, но производительность сети снижается.

Основное различие систем жёсткого и мягкого реального времени можно охарактеризовать так: система жёсткого реального времени никогда не опоздает с реакцией на событие, система мягкого реального времени не должна опаздывать с реакцией на событие.

Часто операционной системой реального времени считают лишь систему, которая может быть использована для решения задач жёсткого реального времени. Это определение означает наличие у ОСРВ необходимых инструментов, но также означает, что эти инструменты необходимо правильно использовать.

Большинство программного обеспечения ориентировано на «мягкое» реальное время. Для подобных систем характерно:

- гарантированное время реакции на внешние события (прерывания от оборудования);
- жёсткая подсистема планирования процессов (высокоприоритетные задачи не должны вытесняться низкоприоритетными, за некоторыми исключениями);
- повышенные требования к времени реакции на внешние события или реактивности (задержка вызова обработчика прерывания не более десятков микросекунд, задержка при переключении задач не более сотен микросекунд).

Классическим примером задачи, где требуется ОСРВ, является управление роботом, берущим деталь с ленты конвейера. Деталь движется, и робот имеет лишь маленький промежуток времени, когда он может её взять. Если он опоздает, то деталь уже не будет на нужном участке конвейера, и следовательно, работа не будет выполнена, несмотря на то, что робот находится в правильном месте. Если он подготовится раньше, то деталь ещё не успеет подъехать, и он заблокирует ей путь.

Также для операционных систем иногда используется понятие «интерактивного реального времени», в котором определяется минимальный порог реакции на события графического интерфейса, в течение которого оператор-человек способен спокойно, без нервозности, ожидать реакции системы на данные им указания.

### ***7.3 Содержание отчета***

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### ***Контрольные вопросы***

- 1 Поясните различие между системами мягкого и жесткого реального времени.
- 2 Изложите порядок работы с отладчиком в среде  $\mu$ Vision 5.
- 3 В каких случаях целесообразно применение отладчика?
- 4 Приведите примеры систем управления реального времени.

## **8 Лабораторная работа № 8. Разработка и отладка программ управления технологическими объектами в реальном времени (часть 2)**

**Цель работы:** изучить связь между аналоговыми регуляторами и цифровыми фильтрами, формы представления цифровых фильтров, особенности их программной реализации, реализацию с помощью языка программирования, составить программу реализации заданного цифрового фильтра, используя язык программирования микроконтроллера, перевести ее в машинные коды, записать в память программ микроконтроллера и проверить работу, используя аналоговые входы и выходы.

## **8.1 *Ход работы***

8.1.1 Изучить электрическую принципиальную схему к лабораторной работе.

8.1.2 Разработать программу в соответствии с индивидуальным заданием.

8.1.3 Отладить программу в среде  $\mu$ Vision.

8.1.4 Загрузить программу в учебный стенд.

8.1.5 Исследовать работу цифроаналогового преобразователя в соответствии с индивидуальным заданием.

8.1.6 Оформить отчет по лабораторной работе.

## **8.2 *Краткие теоретические сведения***

Цифровые системы строятся на базе комплекса средств вычислительной техники, основными элементами которого являются: ЦВМ; устройства ввода; устройства вывода.

Функции ЦВМ могут выполнять: ЭВМ (компьютеры); DSP-цифровые сигнальные процессоры; ЦУ на жесткой логике. Первые относятся к универсальным устройствам управления, вторые специализированы для приложений, третьи разрабатываются для конкретных устройств (например, цифровой фильтр, имеющийся в каждом АЦП).

Устройствами ввода и вывода в случае состыковки с аналоговыми сигналами являются АЦП и ЦАП, а в случае состыковки с цифровыми сигналами – порты и интерфейсы.

В системах с ЦВМ последние могут выполнять роли:

- регулятора;
- регулятора и устройства сравнения;
- корректирующего устройства;
- самого объекта.

Если ЦВМ универсальная (ЭВМ), то возможно построение многофункциональных САУ, когда одна ЦВМ обслуживает комплекс составляющих объект устройств. Во всех случаях ЦВМ предоставляет легко доступные информационные потоки, позволяющие кроме прямого управления осуществлять функции:

- контроля;
- оптимизации;
- координации;
- организации всех процессов.

Пропорционально-интегрально-дифференцирующий (ПИД) регулятор – устройство в управляющем контуре с обратной связью. Используется в системах автоматического управления для формирования управляющего сигнала с целью получения необходимых точности и качества переходного процесса. ПИД-регулятор формирует управляющий сигнал, являющийся суммой трёх слагаемых, первое из которых пропорционально разности входного сигнала и сигнала

обратной связи (сигнал рассогласования), второе – интегралу сигнала рассогласования, третье – производной сигнала рассогласования.

Если какие-то из составляющих не используются, то регулятор называют пропорционально-интегрирующим, пропорционально-дифференцирующим, пропорциональным и т. д.

#### ***Пропорциональная составляющая.***

Пропорциональная составляющая вырабатывает выходной сигнал, противодействующий отклонению регулируемой величины от заданного значения, наблюдаемому в данный момент времени. Он тем больше, чем больше это отклонение. Если входной сигнал равен заданному значению, то выходной равен нулю.

Однако при использовании только пропорционального регулятора значение регулируемой величины никогда не стабилизируется на заданном значении. Существует так называемая статическая ошибка, которая равна такому отклонению регулируемой величины, которое обеспечивает выходной сигнал, стабилизирующий выходную величину именно на этом значении. Например, в регуляторе температуры выходной сигнал (мощность нагревателя) постепенно уменьшается при приближении температуры к заданной, и система стабилизируется при мощности, равной тепловым потерям. Температура не может достичь заданного значения, т. к. в этом случае мощность нагревателя станет равна нулю, и он начнёт остывать.

Чем больше коэффициент пропорциональности между входным и выходным сигналом (коэффициент усиления), тем меньше статическая ошибка, однако при слишком большом коэффициенте усиления при наличии задержек (запаздывания) в системе могут начаться автоколебания, а при дальнейшем увеличении коэффициента система может потерять устойчивость.

#### ***Интегрирующая составляющая.***

Интегрирующая составляющая пропорциональна интегралу по времени от отклонения регулируемой величины. Её используют для устранения статической ошибки. Она позволяет регулятору со временем учесть статическую ошибку.

Если система не испытывает внешних возмущений, то через некоторое время регулируемая величина стабилизируется на заданном значении, сигнал пропорциональной составляющей будет равен нулю, а выходной сигнал будет полностью обеспечиваться интегрирующей составляющей. Тем не менее, интегрирующая составляющая также может приводить к автоколебаниям при неправильном выборе её коэффициента.

#### ***Дифференцирующая составляющая.***

Дифференцирующая составляющая пропорциональна темпу изменения отклонения регулируемой величины и предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем. Отклонения могут быть вызваны внешними возмущениями или запаздыванием воздействия регулятора на систему.

### 8.3 Содержание отчета

Отчет о выполнении лабораторной работы оформляется на компьютере в текстовом редакторе. Отчет должен содержать:

- цель работы;
- описание хода выполнения работы по пунктам;
- листинг программы для микроконтроллера;
- вывод.

### 8.4 Варианты индивидуальных заданий к лабораторной работе

Разработать программу для учебного стенда, позволяющую вырабатывать управляющее воздействие в соответствии с заданными параметрами ПИД-регулятора. Входное воздействие считывается с одного из аналоговых датчиков АД1-АД3 через АЦП. Управляющее воздействие выводить на выход аналогового сигнала через ЦАП. Вывести числовые значения, соответствующие входному воздействию и управляющему воздействию на индикатор: ССИ или МЖКИ. Предусмотреть возможность генерации тестового воздействия в соответствии с состоянием дискретных датчиков ДД1-ДД8.

8.4.1 ПИД-регулятор со следующими параметрами:  $K_p = 0,4$ ;  $T_c = 0,2$  с;  $T_d = 0,8$  с;  $T_i = 12$  с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие – пилообразный сигнал с уровнем сигнала от  $+0,5$  до  $-0,5$  максимального, сгенерировать при  $ДД4 = 1$ . Значение управляющего воздействия вывести на ССИ.

8.4.2 ПИ-регулятор со следующими параметрами:  $K_p = 0,3$ ;  $T_c = 0,4$  с;  $T_i = 28$  с. Входное воздействие снимается с аналогового датчика АД 1. Тестовое воздействие - треугольный сигнал с уровнем сигнала от 0 до 0,5 максимального, сгенерировать при  $ДД1 = 1$ . Значение управляющего воздействия вывести на ССИ.

8.4.3 ПИД-регулятор со следующими параметрами:  $K_p = 0,8$ ;  $T_c = 0,1$  с;  $T_d = 0,4$  с;  $T_i = 24$  с. Входное воздействие снимается с аналогового датчика АД2. Тестовое воздействие - ступенчатый сигнал с уровнем сигнала от 0 до 0,5 максимального, сгенерировать при  $ДД2 = 1$ . Значение управляющего воздействия вывести на МЖКИ.

8.4.4 ПД-регулятор со следующими параметрами:  $K_p = 1$ ;  $T_c = 0,4$  с;  $T_d = 1,2$  с. Входное воздействие снимается с аналогового датчика АД 1. Тестовое воздействие - пилообразный сигнал с уровнем сигнала от 0,5 до 0 максимального, сгенерировать при  $ДД7 = 1$ . Значение управляющего воздействия вывести на ССИ.

8.4.5 ПИ-регулятор со следующими параметрами:  $K_p = 1$ ;  $T_c = 0,5$  с;  $T_i = 21$  с. Входное воздействие снимается с аналогового датчика АД3. Тестовое воздействие – треугольный сигнал с уровнем сигнала от 0 до 0,5 максимального, сгенерировать при  $ДД3 = 0$ . Значение управляющего воздействия вывести на МЖКИ.

8.4.6 ПИД-регулятор со следующими параметрами:  $K_p = 1$ ;  $T_c = 0,4$  с;  $T_o = 1$  с;  $T_i = 10$  с. Входное воздействие снимается с аналогового датчика АД3. Тестовое воздействие – бивал пилообразный сигнал с уровнем сигнала от 0

до 0,5 максимального, сгенерировать при  $ДД4 = 0$ . Значение управляющего воздействия вывести на МЖКИ.

8.4.7 ПД-регулятор со следующими параметрами:  $K_p = 0,8$ ;  $T_c = 0,5$  с;  $T_o = 0,8$  с. Входное воздействие снимается с аналогового датчика АД2. Тестовое воздействие – ступенчатый сигнал с уровнем сигнала от 0 до 0,5 максимального, сгенерировать при  $ДД2 = 0$ . Значение управляющего воздействия вывести на ССИ.

8.4.8 ПИ-регулятор со следующими параметрами:  $K_p = 0,4$ ;  $T_c = 0,4$  с;  $T_i = 45$  с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие – ступенчатый сигнал с уровнем сигнала от -0,5 до +0,5 максимального, сгенерировать при  $ДД5 = 1$ . Значение управляющего воздействия вывести на ССИ.

8.4.9 ПИД-регулятор со следующими параметрами:  $K_p = 0,5$ ;  $T_c = 0,5$  с;  $T_d = 1,2$  с;  $T_i = 36$  с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие - пилообразный сигнал с уровнем сигнала от 0 до 0,5 максимального, сгенерировать при  $ДД3 = 1$ . Значение управляющего воздействия вывести на МЖКИ.

8.4.10 ПД-регулятор со следующими параметрами:  $K_p = 0,3$ ;  $T_c = 0,2$  с;  $T_d = 0,5$  с. Входное воздействие снимается с аналогового датчика АД2. Тестовое воздействие – треугольный сигнал с уровнем сигнала от 0,5 до 0 максимального, сгенерировать при  $ДД1 = 0$ . Значение управляющего воздействия вывести на МЖКИ.

8.4.11 ПИ-регулятор со следующими параметрами:  $K_p = 0,5$ ;  $T_c = 0,6$  с;  $T_d = 60$  с. Входное воздействие снимается с аналогового датчика АД2. Тестовое воздействие – пилообразный сигнал с уровнем сигнала от -0,5 до +0,5 максимального, сгенерировать при  $ДД6 = 0$ . Значение управляющего воздействия вывести на ССИ.

### ***Контрольные вопросы***

- 1 Где применяются цифровые системы управления?
- 2 Какую роль играют АЦП и ЦАП в цифровых системах управления?
- 3 Опишите структуру ПИД-регулятора.
- 4 Какие существуют схемы алгоритмов реализации цифровых фильтров?
- 5 Опишите преимущества и недостатки последовательной и параллельной схем построения алгоритмов цифровой фильтрации.

### **Список литературы**

- 1 **Беккер, В. Ф.** Технические средства автоматизации. Интерфейсные устройства и микропроцессорные средства: учебное пособие / В. Ф. Беккер – 2-е изд. – Москва: РИОР: ИНФРА-М, 2019. – 152 с.
- 2 **Гуров, В. В.** Микропроцессорные системы : учебное пособие / В. В. Гуров. – Москва: ИНФРА-М, 2019. – 336 с.