

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Белгородский государственный технологический университет
им. В. Г. Шухова
Кафедра технической кибернетики

Утверждено
научно-методическим советом
университета

**ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО
МИКРОКОНТРОЛЛЕРАМ СЕМЕЙСТВА
CORTEX-M3**

Учебное пособие по проведению лабораторных работ на отладочных
платах фирмы Миландр для студентов направлений бакалавриата
27.03.04 – «Управление в технических системах», 15.03.04 –
«Автоматизация технологических процессов и производств»,
15.03.06 – «Мехатроника и робототехника»

Белгород
2016

УДК 004.3 (07)
ББК 32.81 я7
М38

Рецензенты:

Составители: д-р техн. наук, проф. В.Г. Рубанов
канд. техн. наук, проф. А.С. Кижук
ст. преп. Д.А. Бушуев
зав. лаб. Е.Б. Кариков
инж. Е.П. Добринский

М38 Лабораторный практикум по микроконтроллерам семейства Cortex-M: Методическое пособие по проведению лабораторных работ на отладочных платах фирмы Миландр /сост.: В.Г. Рубанов, А.С. Кижук, Д.А. Бушуев, Е.Б. Кариков, Е.П. Добринский. – Белгород: Изд-во БГТУ, 2016. – с.

Учебное пособие содержит лабораторные работы для практического изучения микроконтроллеров с архитектурой Cortex-M3 на отладочных платах фирмы Миландр. Содержатся теоретические сведения, рекомендации к выполнению, контрольные вопросы для подготовки к допуску и защите, порядок выполнения и структура отчета по каждой лабораторной работе.

Учебное пособие предназначено для студентов третьего курса направлений бакалавриата 27.03.04 – «Управление в технических системах», 15.03.04 – «Автоматизация технологических процессов и производств», 15.03.06 – «Мехатроника и робототехника» по дисциплинам изучающих микропроцессорную технику.

Издание публикуется в авторской редакции.

УДК 004.3 (07)
ББК 32.81 я7

© Белгородский государственный
технологический университет
(БГТУ) им. В.Г. Шухова, 2016

Содержание

Введение	4
Лабораторная работа №1	5
Лабораторная работа №2.....	16
Лабораторная работа №3.....	24
Лабораторная работа №4.....	33
Лабораторная работа №5.....	40
Лабораторная работа №6.....	44
Лабораторная работа №7.....	51
Библиографический список	60

Введение

Современный микроконтроллер – это размещенная на кристалле сложная цифровая система, в состав которой входит 8- 16- или 32-разрядный процессор, внутренняя память программ, широкий набор интерфейсных и периферийных устройств, портов ввода-вывода, таймеров, аналого-цифровых преобразователей и др. Эти однокристалльные системы ориентированы, в первую очередь, на выполнение функций управления различными устройствами и процессами, почему и названы микроконтроллерами. Сфера их применения весьма широка: от современной бытовой техники до сложнейших систем управления технологическими процессами и робототехническими комплексами. За последние пять лет выпуск микроконтроллеров в мире увеличился в три раза и приблизился к уровню в 2 миллиарда штук в год. Вместе с тем в РФ в последнее время началась широкая компания по импортозамещению, что особенно актуально в высокотехнологичных отраслях промышленности и производства. Поэтому данный курс ориентирован на изучение микроконтроллеров с отладочными платами, именно отечественного производства, группы компаний «Миландр», созданных на базе организованного в 1995 г. в Зеленограде ЗАО «ПКК Миландр». Компания специализируется на выпуске микроконтроллеров и микропроцессоров (8-, 16- и 32-разрядных), микросхем памяти (ОЗУ и ПЗУ), интерфейсных микросхем, радиочастотных микросхем, контроллеров питания, АЦП, ЦАП, электронных модулей, радиоаппаратуры и т.д.

В данном учебном пособии рассматривается ряд приложений для микроконтроллера серии 1986VE9х архитектуры Cortex-M3.

Лабораторная работа №1

Изучение архитектуры отладочной платы микроконтроллера 1986BE93У

Цель работы

1. Изучение принципиальной схемы связи микроконтроллера с элементами периферии отладочной платы;
2. Изучение связи портов с периферией интегральных микросхем отладочной платы;
3. Изучение назначения и характеристик интерфейсов микроконтроллера 1986BE93У фирмы Миландр.

План работы

1. Ознакомление с основными характеристиками микроконтроллера 1986BE93У.
2. Ознакомление с техническим описанием и архитектурой отладочной платы для микроконтроллера 1986BE93У.
3. Изучение основных элементов периферии платы, их функций и характеристик.

Основные теоретические сведения и содержание работы

1. Общие сведения об отладочной плате

Демонстрационно-отладочная плата предназначена для:

- демонстрации функционирования и оценки производительности микроконтроллера 1986BE93У и его основных периферийных модулей;
- демонстрации функционирования интерфейсных микросхем CAN и COM (RS-232) интерфейсов;
- отладки собственных проектов с применением, установленных на плате блоков;
- программирования памяти программ микроконтроллеров 1986BE93У.

Для программирования памяти программ микроконтроллеров 1986BE93У применяется внешний внутрисхемный программатор ULINK2 (Keil) или JEM-ARM-V2(Phyton).

Питание платы осуществляется от адаптера постоянного тока напряжением 5 вольт или от шины USB.

2. Техническое описание отладочной платы

Основные характеристики микроконтроллера 1986BE93У

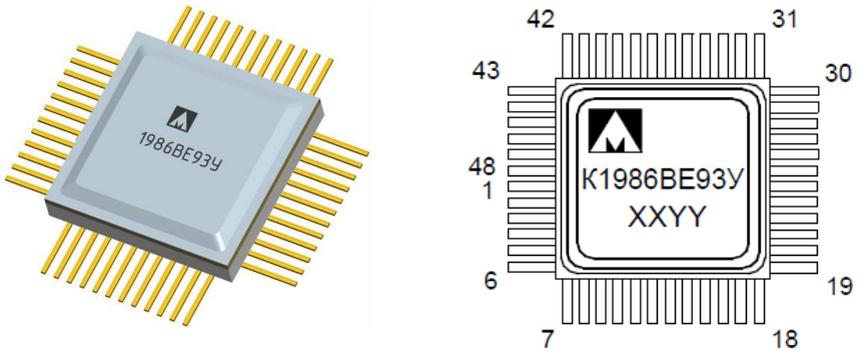


Рис. 1.1. Микроконтроллер 1986BE93У в металлическом корпусе Н16.48-1В

Характеристики микроконтроллера 1986BE93У (рис. 1.1):

Ядро:

- ARM 32-битное RISC-ядро Cortex™-M3 (рис. 1.2) ревизии 2.0, тактовая частота до 80 МГц, производительность 1.25 DMIPS/МГц (Dhrystone 2.1) при нулевой задержке памяти.

Память:

- встроенная энергонезависимая Flash-память программ размером 128 Кбайт;
- встроенное ОЗУ размером 32 Кбайт.

Питание:

- внешнее питания 2,2 ÷ 3,6 В;
- встроенный регулируемый стабилизатор напряжения на 1,8 В для питания ядра.

Аналоговые модули:

- два 12-разрядных АЦП, 4 канала;
- температурный датчик;
- двухканальный 12-разрядный ЦАП;
- встроенный компаратор, 2 входа.

Периферия:

- контроллер USB интерфейса с функциями работы Device и Host;
- контроллеры интерфейсов: UART – 2шт., SPI – 1шт., CAN - 2шт.;
- три 16-разрядных таймер-счетчика с функциями ШИМ и регистрации событий;
- 30 пользовательских линий ввода-вывода.

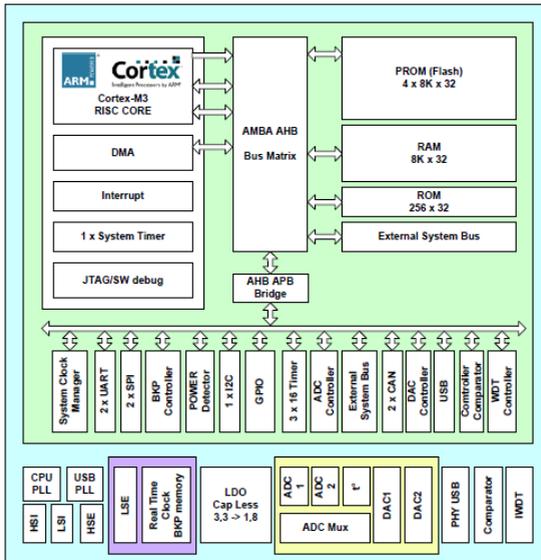


Рис. 1.2. Структурная блок-схема микроконтроллера 1986BE93У

Состав отладочной платы

Внешний вид платы представлен на рис 1.3.

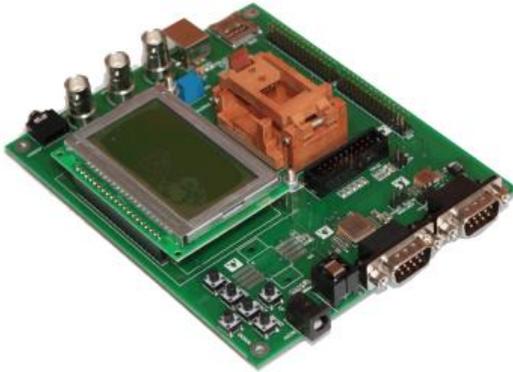


Рис. 1.3. Внешний вид отладочной платы
Компоненты, установленные на отладочной плате, показаны на рис.4.

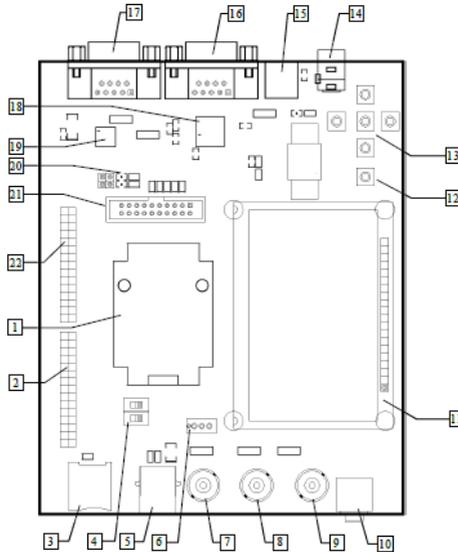


Рис. 1.4. Компоненты отладочной платы

№ на рис. 4	Описание компонентов платы
1.	Контактная площадка для микроконтроллера 1986BE93У. Микроконтроллер должен быть установлен в спутник-держатель
2.	Разъем X25 портов А,Е,Ф микроконтроллера
3.	Разъем для карты памяти microSD
4.	Переключатели выбора режима загрузки

5.	Разъем USB-B
6.	Подстроечный резистор 1кОм на 2-м канале АЦП
7.	Разъем BNC внешнего сигнала на 2-м канале АЦП
8.	Разъем BNC внешнего сигнала на 1-м входе компаратора
9.	Разъем BNC выхода ЦАП
10.	Разъем Audio 3.5мм выхода ЦАП через звуковой усилитель MC34119L (установлен на оборотной стороне)
11.	ЖК индикатор 128x64
12.	Кнопка RESET
13.	Кнопки UP, DOWN, LEFT, RIGHT, SELECT
14.	Разъем питания 5В
15.	Фильтр питания
16.	Разъем RS-232
17.	Разъем CAN
18.	Приемо-передатчик RS-232 ADM3232EARN (установлен на оборотной стороне)
19.	Приемо-передатчик CAN ATA6660 (установлен на оборотной стороне)
20.	Набор светодиодов на порту F
21.	Разъемы отладки JTAG-A и JTAG-B
22.	Разъем X24 портов В,С,D микроконтроллера

Подключение портов микроконтроллера к периферийным элементам платы:

➤ Набор светодиодов (См. №20, рис. 1.4):

Порт МК	№ вывода МК	Светодиоды
PF0	44	Светодиод VD2
PF1	45	Светодиод VD3

➤ Кнопки UP, DOWN, LEFT, RIGHT, SELECT (См. №13, рис.4):

Порт МК	№ вывода МК	Кнопки
PD0	17	Кнопка UP
PD1	18	Кнопка DOWN
PE3	10	Кнопка LEFT
PE6	6	Кнопка RIGHT
PC0	23	Кнопка SELECT

➤ ЖК индикатор 128x64 (См. №11, рис.4):

Порт МК	№ вывода МК	Вывод индикатора (№ вывода)
PA0	41	Вывод DB0 (4)
PA1	40	Вывод DB1 (5)

PA2	39	Вывод DB2 (6)
PA3	38	Вывод DB3 (7)
PA4	37	Вывод DB4 (8)
PA5	36	Вывод DB5 (9)
PA6	35	Вывод DB6 (10)
PA7	34	Вывод DB7 (11)
PF0	44	Вывод E1 (12)
PF1	45	Вывод E2 (13)
PF2	46	Вывод RES (14)
PF3	47	Вывод R/W (15)
PF4	48	Вывод A0 (16)
PD3	20	Вывод E (17)

- Разъем для карты памяти microSD (См. №3, рис. 1.4):

Порт МК	№ вывода МК	Вывод microSD (№ вывода)
PF0	44	Вывод CMD (3)
PF1	45	Вывод CLK (5)
PF2	46	Вывод CD/DAT3 (2)
PF3	47	Вывод DAT0 (7)

- Разъем BNC внешнего сигнала на 2-м канале АЦП (См. №7, рис. 1.4):

Порт МК	№ вывода МК
PD2	19

- Разъем BNC внешнего сигнала на 1-м входе компаратора (См. №8, рис. 1.4):

Порт МК	№ вывода МК
PE2	11

- Разъем BNC выхода ЦАП1 (См. №9, рис. 1.4):

Порт МК	№ вывода МК
PE0	14

- Разъем Audio 3.5мм выхода ЦАП1 через звуковой усилитель MC34119L (См. №10, рис. 1.4):

Порт МК	№ вывода МК
PE0	14

- Разъем RS-232C через приемо-передатчик RS-232 ADM3232EARN (См. №16, 18, рис. 1.4):

Порт МК	№ вывода МК	Вывод ADM3232EARN (№ вывода)
PB5	18	Вывод TD1 (11)
PB6	19	Вывод RQ1 (12)

№ контакта разъема RS-232C	Вывод ADM3232EARN (№ вывода)
2(TXD)	Вывод RD1 (13)
3(RXD)	Вывод TQ1 (14)

- Разъем CAN через прямо-передатчик CAN ATA6660 (См. №17, 19, рис. 1.4):

Порт МК	№ вывода МК	Вывод ADM3232EARN (№ вывода)
PA6	5	Вывод TXD (1)
PA7	6	Вывод RXD (4)

№ контакта разъема CAN	Вывод ATA6660 (№ вывода)
2(CANL)	Вывод CANL (6)
7(CANH)	Вывод CANH (7)

- Разъем USB-B (См. №5 рис. 1.4):

Порт МК	№ вывода МК	№ контакта разъема USB-B
DP	2	3(D+)
DN	3	2(D-)

3. Техническое описание периферийных устройств - ЖК индикатор 128x64

ЖК индикатор имеет 20 контактный разъем для подключения.

Таблица обозначения контактов:

Вывод	Обозначение	Назначение
1	Ucc	Питание цифровой части модуля
2	GND	Общий вывод (0 В, земля)
3	Uo	Вход питания ЖК-панели
4-11	DB0—DB7	Шина данных
12	E1	Выбор кристалла 1
13	E2	Выбор кристалла 2

14	RES	Сброс (начальная установка)
15	R/W	Выбор: Чтение/Запись
16	A0	Выбор: Команда / Данные
17	E	Стробирование данных
18	Uee	Выход DC-DC преобразователя
19	A	+ питания подсветки
20	K	- питания подсветки

Некоторые особенности подключения:

1. Цифровая часть, ЖК-панель и подсветка запитываются отдельно друг от друга.
2. Вход питания ЖК-панели (3) необходимо соединить с выходом DC-DC преобразователя (18) через потенциометр для возможности настройки контраста. Либо через резистор ~4.7 КОм.
3. Внутри ЖК-панели находятся два контроллера KS-0108. Поскольку размер матрицы 128x64, то каждый контроллер отвечает за свою половину экрана и активизируется соответствующим пином E1 и E2. Необходимо учитывать, что запись или чтение сразу в/из обоих контроллеров не допускается!

Память:

1. Модуль содержит ОЗУ для хранения данных, выводимых на ЖКИ, размером 64x64x2 бит (по 64x64 бит на каждый кристалл). Для выбора нужного кристалла используются выводы E1,E2.
2. ОЗУ разбито на 8 страниц размером по 64x8 бит каждая. Каждой светящейся точке на ЖКИ соответствует логическая «1» в ячейке ОЗУ модуля.

Контроллер дисплея понимает 7 команд:

1. Включение / выключение дисплея вне зависимости от данных в ОЗУ и внутреннего состояния;
2. Установка номера строки ОЗУ, которая будет отображаться в верхней строке дисплея;
3. Установка номера страницы ОЗУ;
4. Установка адреса ОЗУ для последующих обращений;
5. Чтение статуса состояния;
6. Запись данных;

7. Чтение данных.

Приемо-передатчик RS-232 ADM3232EARN

Принципиальная схема приемо-передатчика приведена на рис. 1.5.

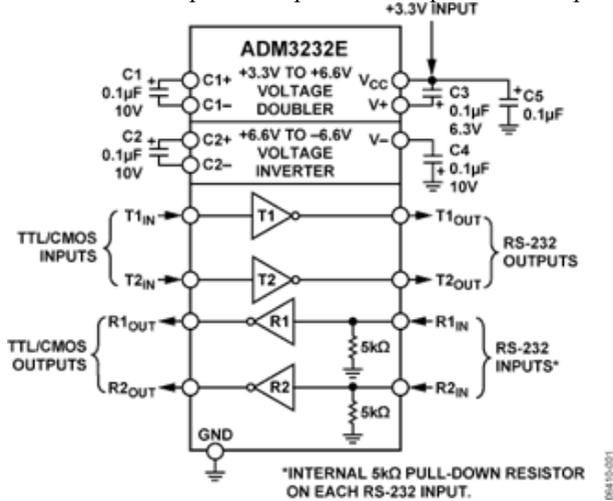


Рис. 1.5. Принципиальная схема приемо-передатчика

Характеристики ADM3232EARNZ:

Производитель	Analog Devices Inc
Тип	Transceiver
Количество приемников/передатчиков	2/2
Интерфейс	RS232
Напряжение питания	3 V ~ 5.5 V

Приемо-передатчик CAN ATA6660

Структурная схема приемо-передатчика приведена на рис. 6.

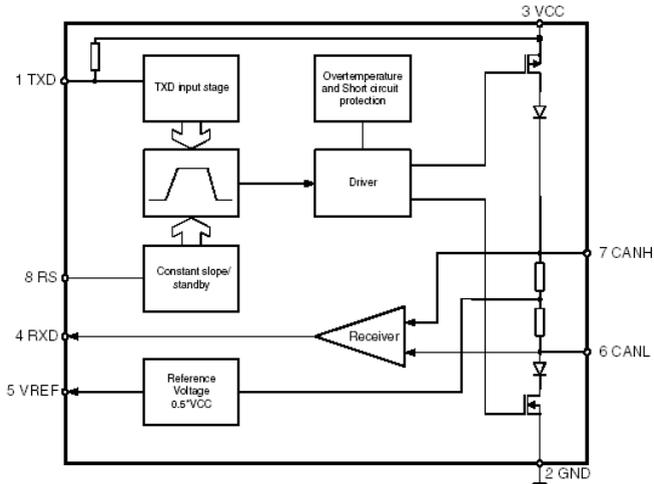


Рис.1.6. Структурная схема приемопередатчика

Характеристики АТ6660:

- Напряжение питания 5 V.
- Максимальная высокоскоростная передача данных до 1 Мбод.
- Управляемая скорость нарастания фронтов.
- Дежурный режим.
- Вход TXD совместим с 3.3В логикой.
- Защита от короткого замыкания.
- Защита от превышения температуры.
- Защита линий от перенапряжений от -40 В до +40 В.
- Высокоскоростной дифференциальный приемник с широким допустимым диапазоном синфазного сигнала (-10 В... +10 В) для высокой электромагнитной устойчивости (EMI).
- Полностью управляемые линии шины CANH и CANL для минимизации электромагнитных излучений (EME).
- Высокий уровень защиты от электростатических разрядов (ESD) на выводах CANH, CANL: 8 кВ (модель человеческого тела HBM), 300 В (машинная модель MM).

Примечание: вся необходимая документация об отладочной плате и периферийных устройствах выдается в электронном виде.

Порядок выполнения лабораторной работы

1. Подготовка и допуск к лабораторной работе.
2. Освоение теории по указанной в методическом пособии литературе, до начала лабораторной работы.
3. Запись в тетрадь вопросов по теории.
4. Обсуждение непонятных вопросов с преподавателем.
5. Освоение материала из спецификации (документации) на МК.
6. Зарисовка структурных-схем изучаемых блоков.
7. Запись в тетрадь вопросов по документации.

Содержание отчета

1. Описание платы по структурной схеме, составленной из имеющейся принципиальной схемы.
2. Описание периферийных элементов отладочной платы с пояснением функций каждого из них.
3. Описать назначение и функции разъемов и переключателей платы.
4. Выводы.

Контрольные вопросы на защиту

1. Устройство и характеристики микроконтроллера 1986BE93У, порты ввода/вывода.
2. Назначение разъемов на плате.
3. Интерфейсы микроконтроллера 1986BE93У, характеристики, отличия.
3. Основные элементы периферии, их характеристики и назначение.
4. Принципиальная схема платы, назначение элементов их характеристики

Лабораторная работа №2

Создание проекта для работы с МК 1986BE93У в среде Keil MDK

Цель работы

Ознакомиться с настройкой проекта в среде разработки программ Keil MDK

Основные положения

Программирование микроконтроллеров осуществляется с помощью специализированных пакетов программ, называемых средами разработки. Одной из таких сред разработки является Keil MDK. Для начала работы необходимо создать проект, который будет содержать код программы для микроконтроллера.

Процедура создания проекта состоит в выполнении следующих действий. Открываем вкладку "Project" и выбираем "New uVision Project". В появившемся окне выбираем микроконтроллер и нажимаем "OK" (рис. 2.1).

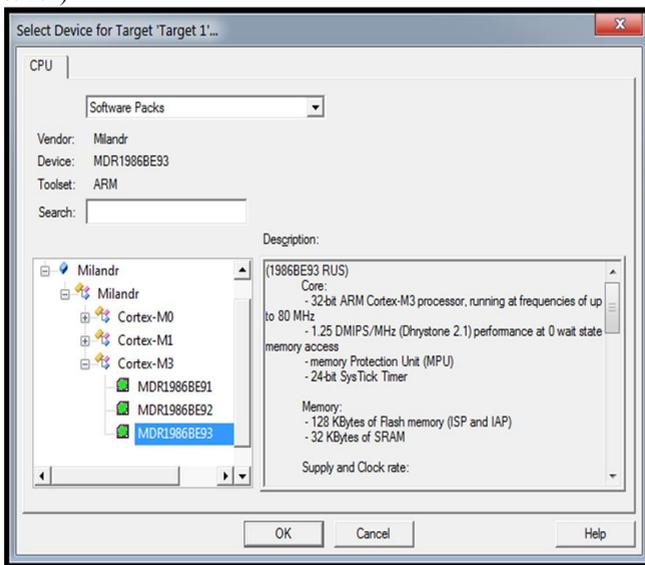


Рис. 2.1. Выбор микроконтроллера в среде Keil MDK

После появления окна, в котором выбираются нужные нам библиотеки для работы с отладочной платой (рис. 2.2). Нам понадобятся библиотеки для работы с портами и для тактирования этих портов. В списке "Drivers" следует поставить галочки на "PORT"

и "RST_CLK", а в списке "Device" поставить галочку на "Startup_MDR1986BE9x" и нажать "OK".

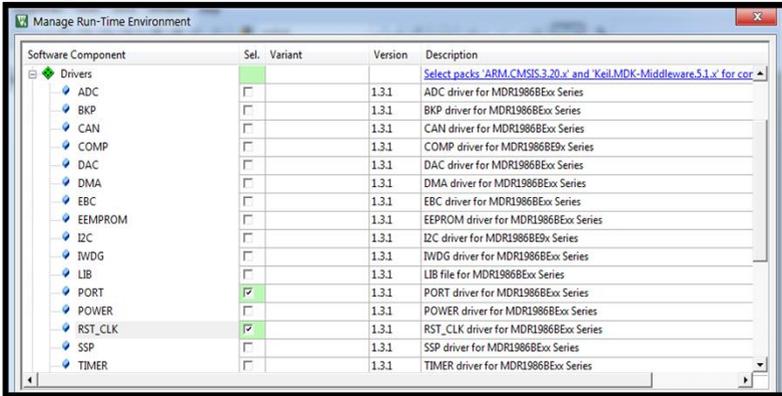


Рис. 2.2. Выбор библиотек в среде Keil MDK

Заходим во вкладку "File" и создаем новый файл выбрав "New". Сохраняем этот файл как "main.c" и добавляем его в проект.

Следующим шагом будет настройка и подключение отладочной платы, которая описана в лабораторной работе №1.

Написание программы для мигания светодиода.

При подключении светодиода к микроконтроллеру возможны два следующих случая:

1. светодиод подключен катодом к микроконтроллеру (рис.2.3);
2. светодиод подключен анодом к микроконтроллеру (рис.2.4).

Рассмотрим первый вариант:

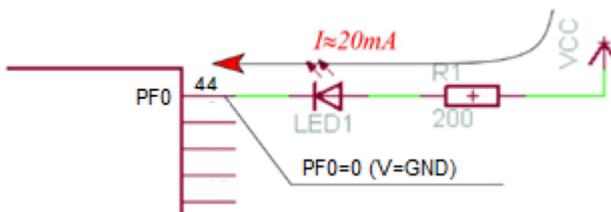


Рис. 2.3. Схема подключения светодиода с общим анодом

Для того, чтобы светодиод начал излучать свет, необходимо создать разность потенциалов между анодом и катодом. Анод подключен к положительному выводу источника питания (VCC). Таким образом, когда PF0 сбрасывается в «0», на нем образуется нулевой потенциал (GND) и через светодиод начинает протекать ток. Если вывод PF0 = 1 (VCC), то разности потенциалов не будет и

светодиод не излучает свет.

Рассмотрим второй вариант:

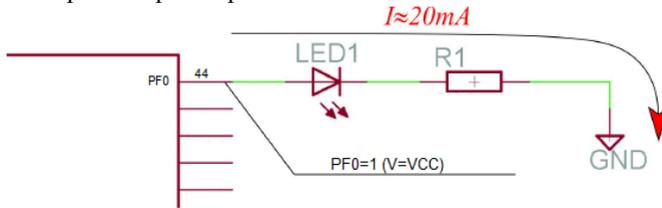


Рис. 2.4. Схема подключения светодиода с общим катодом

Катод подключен к отрицательному выводу источника питания (GND). При сравнении с предыдущей схемой становится ясно, что когда PF0 сбрасывается в «0» – светодиод не будет излучать свет, а когда PF0 = 1 светодиод будет излучать свет.

Подключаем заголовочные файлы "MDR32F9Qx.h", "MDR32F9Qx_config.h", "MDR32F9Qx_port.h", "MDR32F9Qx_rst_clk.h" (рис. 2.3).

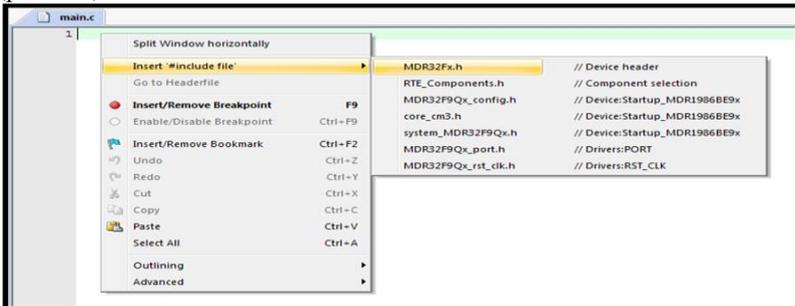


Рис. 2.5. Подключаем заголовочных файлов в среде Keil MDK

Открываем электрическую схему отладочной платы (приложение №1) и определяем к какому порту подсоединен светодиод (рис. 2.4).

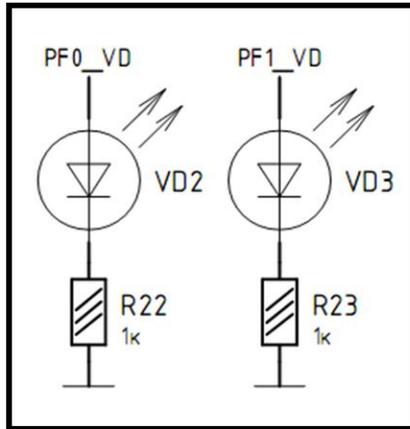


Рис. 2.6. Подключение светодиодам к портам МК на отладочной плате

На электрической схеме отладочной платы можем увидеть, что 2 светодиода подключены к выходам PF0 и PF1. Следовательно нам нужно настроить порт "F". Для этого необходимо написать процедуру, с помощью которой произведем настройку и тактирование порта.

```
void init_PORTF(void)
{
    PORT_InitTypeDef LED; //Создаем идентификатор для порта F
    RST_CLK_PCLKcmd(RST_CLK_PCLK_PORTF, ENABLE);
//Включаем тактирование порта F
    LED.PORT_SPEED = PORT_SPEED_MAXFAST; //Настройка
    скорости порта
    LED.PORT_FUNC = PORT_FUNC_PORT; //Режим работы как
    порт
    LED.PORT_MODE = PORT_MODE_DIGITAL; //Режим работы
    цифровой
    LED.PORT_OE = PORT_OE_OUT; //Настраиваем порт на выход
    LED.PORT_Pin = PORT_Pin_0; //Выбор нулевого пина
    LED.PORT_PULL_DOWN = PORT_PULL_DOWN_OFF;
//подтяжка порта к земле выключена
    LED.PORT_PULL_UP = PORT_PULL_UP_OFF; //Подтяжка порта
    к питанию выключена
    PORT_Init(MDR_PORTF, &LED); //Применение настроек к
    порту F
}
```

Для детальной настройки порта необходимо открыть спецификацию микроконтроллера, которая находится в директории

"Milandr\Документация \Спецификации\Микроконтроллеры и микропроцессоры\1986\1986BE9X.pdf".

Напишем основную функцию для мигания светодиодом:

```
int main(void)
{
    uint8_t i;
    while(1)
    {
        PORT_SetBits(MDR_PORTF, PORT_Pin_0); //Включение 0
        пина порта F
        for(i=0;i<=1000000;i++){ //Задержка
        PORT_ResetBits(MDR_PORTF, PORT_Pin_0); //Отключение
        0 пина порта F
        for(i=0;i<=1000000;i++){
            }
        }
    }
}
```

Компилируем программу и загружаем её на микроконтроллер. Светодиод на плате должен мигать.

Вычислительная мощность современных микроконтроллеров позволяет использовать встраиваемые операционные системы реального времени (ОСРВ) при необходимости выполнения нескольких задач одним устройством. Существует большое количество свободно распространяемых ОСРВ, совместимых с микроконтроллерами архитектуры Cortex-M. Одной из наиболее распространенных ОСРВ является FreeRTOS.

Основные особенности ОС:

- FreeRTOS фактически можно считать ОС жесткого реального времени, но это будет зависеть от приложения, в котором она используется.
- Планировщик может быть вытесняющим или кооперативным. Все зависит от конфигурации ОС. Кооперативный планировщик не имеет приоритетов и просто передает управление от одной задачи к другой. Вытесняющий планировщик выбирает задачу с максимальным на данный момент приоритетом, его работа основана на приоритетах, которые присваиваются задачам при их создании.
- Имеется возможность использования сопрограмм (co-routine).
- Динамическое планирование осуществляется через равные промежутки времени, которые равны тикам системного таймера. Частота тиков задается на этапе разработки приложения.

- Алгоритм планирования основан на выборе самого высокого на данный момент приоритета. Если имеется больше чем одна задача с самым высоким приоритетом, то выполнение происходит поочередно.

Межпроцессорное взаимодействие:

- *Очередь* – это простой FIFO буфер, который может хранить фиксированное число элементов, известного размера. Очереди – это, по сути, независимые объекты, которые могут иметь множество писателей, и читателей, без боязни прочитать/записать битые данные. При чтении данных можно указать время, в течение которого процесс должен находиться в ожидании получения новых данных. При записи данных мы также можем указать данное время, но уже для ожидания места в очереди.
- *Синхронизация* – можно использовать бинарные, счетные и рекурсивные семафоры, а также мьютексы. Они используются атомарно, т.е. в момент использования межзадачного взаимодействия прерывания запрещены, планировщик временно приостанавливает свою работу.
- *Предотвращение блокирования или тупика в задачах* – осуществляется в FreeRTOS либо не блокированием задачи, либо с блокированием на фиксированный промежуток времени. Задачам, которые находятся в конце времени блокировки и все еще не могут получить доступ к ресурсу, API возвращает ошибку. Таймаут на каждом блоке уменьшает вероятность тупика для доступа к ресурсам.
- *Запрещение прерывания* – имеются критические секции в которых запрещены прерывания. Критические секции в пределах задачи могут быть вложенными. Каждая задача отслеживает количество вложений. Имеется возможность уступить процессор внутри критической секции (при использовании кооперативного планировщика).
- *Приостановка планировщика* – может использоваться для получения полного доступа к микроконтроллеру.
- *Выделение памяти.* Во FreeRTOS имеется три модели выделения памяти. Первая модель предусматривает простое выделение памяти при создании каждой задачи, но не имеет механизма освобождения или повторного использования памяти (поэтому задачи не могут быть удалены). Вторая модель позволяет выделять и распределять память и использует самый пригодный для распределения памяти

алгоритм, однако не может комбинировать смежные свободные части памяти. Третья модель обеспечивает обертки вокруг функций `malloc ()` и `calloc ()`.

- *Борьба с инверсией приоритетов.* FreeRTOS не имеет никаких способов борьбы с инверсией приоритетов.

Для работы любой ОС необходим эталон времени. Этот эталон используется для генерации периодических прерываний, которые позволяют прерывать работу задачи и давать ядру возможность периодически выполнять свою работу. В данной ОС этим эталонным интервалом времени называется тик (Tick), который используется в качестве базовой единицы измерения времени в системе. Основу приложения составляет ядро системы, обеспечивающее необходимую функциональность, и прикладные задачи пользователя. Запуск приложения осуществляется в два этапа:

- Настройка приложения, создание задач пользователя.
- Запуск планировщика задач.

Основная функциональность системы заключается в планировщике задач, который осуществляет распределение процессорного времени между задачами, выбирает задачу, запускаемую в настоящий момент времени. Планировщик задач в случае необходимости сохраняет контекст текущей активной задачи и восстанавливает контекст задачи, назначенной к исполнению.

Контекст задачи – это набор данных, содержащий всю необходимую информацию для возобновления выполнения задачи с того места, где она была ранее прервана. Контекст задачи обычно содержит значение рабочих регистров микроконтроллера, счетчика команд, слово состояния. Такое переключение контекстов является основным механизмом работы ОС при переходе от выполнения одной задачи к выполнению другой.

Запуск планировщика осуществляется вызовом `vTaskStartScheduler`. **До запуска планировщика задачи не переключаются и не выполняются!** Для остановки планировщика используется вызов `vTaskEndScheduler`, который осуществляет остановку всех задач, освобождение памяти и далее программа продолжает работу с места следующего за вызовом `vTaskStartScheduler`.

Иерархически FreeRTOS содержит два уровня:

- Ядро системы: `tasks.c`, `queue.c`, `list.c`. Дополнительно для использования со-routine нужен файл `croutine.c`. Плюс несколько заголовочных файлов в папке `include`.

- Абстрактный уровень представления аппаратного обеспечения (Hardware Abstract Layer HAL) для комбинации компилятор-микроконтроллер. Этот уровень находится в файлах `port.c` и `portmacro.h`. Он зависит от используемого микроконтроллера и компилятора.

Для сборки своего приложения потребуется поместить в одно место указанные выше файлы, а также файлы из папки, в которой содержится порт микроконтроллера. После этого настроить соответствующим образом файл `FreeRTOSConfig.h`. После этого можно приступать к написанию задач пользователя.

Порядок выполнения лабораторной работы

1. Ознакомиться с особенностями работы в среде Keil MDK с использованием средств CMSIS и SPL Library.
2. Создать новый проект в среде Keil MDK и подключить к нему CMSIS и SPL Library.
3. Добавить в проект одну из свободно распространяемых операционных систем (например, FreeRTOS, CoOS или ChibiOS) и реализовать с их помощью управление включением светодиодов по короткому и длительному нажатиям на одну из кнопок.
4. Продемонстрировать преподавателю результаты работы.

Содержание отчета

1. Принципиальная электрическая схема для используемой периферии микроконтроллера.
2. Листинг программы.
3. Выводы.

Контрольные вопросы на защиту

1. Укажите способы программирования FLASH-памяти микроконтроллеров МИЛАНДР.
2. Каковы назначение и возможности библиотек CMSIS и SPL?
3. Какие механизмы распределения памяти применяются в ОСРВ?
4. Какие способы межзадачного взаимодействия существуют в ОСРВ?

Лабораторная работа №3

Программирование портов ввода/вывода GPIO

Цель работы

1. Познакомиться с принципом работы портов ввода/вывода, научиться управлять светодиодами и считывать сигнал с кнопок и датчиков с дискретным выходом, начать изучение среды и языка программирования микроконтроллеров 1986VE93У фирмы Миландр.
2. Ознакомиться с принципом вывода информации на ЖК-дисплей;
3. Изучить устройство и принцип работы матричной клавиатуры.

План работы

1. Настройка порта на вывод, мигание светодиодом;
2. Настройка порта на ввод, подключение кнопок и датчиков с дискретным выходом;
3. Вывод информации на ЖК-дисплей;
4. Подключение матричной клавиатуры.

Основные теоретические сведения и содержание работы

Порт – физическое или логическое соединение, через которое принимаются и отправляются данные.

Порт ввода-вывода – используется в микропроцессорах при обмене данными с аппаратным обеспечением. Порт ввода-вывода сопоставляется с тем или иным устройством и позволяет программам обращаться к нему для обмена данными.

Применительно к микроконтроллерам каждый порт ввода/вывода состоит из определенного числа выводов, через которые микроконтроллер может передавать или принимать цифровые сигналы.

В микроконтроллере 1986VE93У имеется шесть портов ввода-вывода, которые обозначаются латинскими буквами «А», «В», «С», «D», «Е», «F».

Задание направления передачи данных через любой контакт порта ввода-вывода может быть произведено программно в любой момент времени.

Светодиод – это полупроводниковый прибор, излучающий некогерентный свет при пропускании через него электрического тока.

1. Настройка проекта в среде Keil uVision 5 и мигание светодиода.

Процесс создания программы управления свечением светодиода описан в лабораторной работе №2.

2. Настройка порта на ввод, подключение кнопок и датчиков с дискретным выходом

Аналогично подключению светодиода для подключения кнопок к МК

существуют две схемы:

- 1) схема с pull-up резистором;
- 2) схема с pull-down резистором.

Подтягивающие резисторы (pull-up и pull-down) необходимы для того, чтобы вывод МК при разомкнутой кнопке имел строго определенный потенциал, а не «болтался в воздухе». Без этого резистора вывод МК будет находиться в третьем состоянии (потенциал на выводе не определен), а при попытке определения состояния вывода, будет считана или «1», или «0» (зависит от внешних наводок на плате).

Рассмотрим схему с pull-up резистором более детально, так как именно она и используется в нашей отладочной плате (рис.3.1).

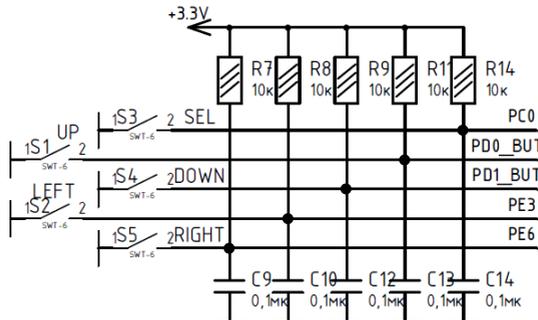


Рис. 3.1. Подключение кнопок к МК на отладочной плате

Когда кнопка разомкнута вывод PD0, через подтягивающий резистор R9 имеет потенциал $V_{CC}=+3.3V$ ($PD0 = 1$). При замыкании кнопки на потенциал вывода МК будут оказывать влияние V_{CC} (через $R9 = 10k$) и GND (сопротивления практически нет). Таким образом, потенциал GND превысит V_{CC} и при чтении состояния вывода получим $PD0=0$.

Схема с pull-down резистором работает «зеркально» относительно схемы с pull-up резистором.

Кроме датчиков с выходом типа «сухой контакт» существуют также датчики и дискретным выходом типа NPN или PNP. Схема подключения таких датчиков с TTL выходом не отличается от схемы подключения обычной кнопки, необходимо только учесть «0» или «1» присутствует на выходе датчика при его срабатывании.

Если же выход датчика рассчитан на работу с 12В и 24В, в данном случае, чтобы подключить такой датчик к микроконтроллеру требуется согласующий каскад на транзисторе или делитель напряжения (рис.3.2).

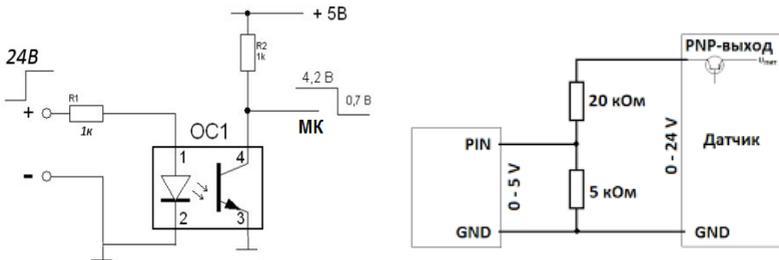


Рис. 3.2. Подключение датчиков к микроконтроллеру через делитель напряжения и схему на транзисторе с оптронной развязкой

При выполнении лабораторной работы первое, что необходимо сделать, это определить к какому из портов подключены кнопки (см. лаб. раб. №1). В данной версии на отладочной плате имеется 5 кнопок: «UP», «DOWN», «LEFT», «RIGHT», «SELECT», подключенные к портам PD0, PD1, PE3, PE6, PC0 соответственно. Данные кнопки будем использовать в качестве конечных выключателей – датчиков с дискретным выходом. Для понятия принципа работы порта ввода выберем кнопку «UP».

В качестве индикации срабатывания кнопки-датчика будем использовать светодиод «PF0», который должен загореться при нажатии на кнопку.

Следовательно, нам нужно в программе настроить порты «F» и «D». Для этого напишем процедуру, в которой произведем настройку и тактирование порта.

```
void init PORTF(void)
{
    PORT_InitTypeDef LED;    //Создаем идентификатор для порта
    F
    RST CLK PCLKcmd(RST CLK PCLK PORTF, ENABLE); //Включаем
    тактирование порта F
    LED.PORT_SPEED = PORT_SPEED_MAXFAST; //Настройка скорости
    порта
    LED.PORT_FUNC = PORT_FUNC_PORT;    //Режим работы порт
```

```

LED.PORT_MODE = PORT_MODE_DIGITAL; //Режим работы цифровой
LED.PORT_OE = PORT_OE_OUT;          //Настраиваем порт на
вывод
LED.PORT_Pin = PORT_Pin_0;          //Выбор нулевого пина PF0
LED.PORT_PULL_DOWN = PORT_PULL_DOWN_OFF; //Выключаем
подтяжку порта порта к земле
LED.PORT_PULL_UP = PORT_PULL_UP_OFF; //Выключаем подтяжку
порта порта к питанию
PORT_Init(MDR_PORTF, &LED);        //Применение настроек к
порту F
}

void init PORTD(void)
{
PORT_InitTypeDef BUT;              //Создаем идентификатор для порта
F
RST_CLK_PCLKcmd(RST_CLK_PCLK_PORTD, ENABLE); //Включаем
тактирование порта D
BUT.PORT_SPEED = PORT_SPEED_SLOW; //Настройка скорости
порта
BUT.PORT_FUNC = PORT_FUNC_PORT;     //Режим работы порт
BUT.PORT_MODE = PORT_MODE_DIGITAL; //Режим работы цифровой
BUT.PORT_OE = PORT_OE_IN;           //Настраиваем порт на ввод
BUT.PORT_Pin = PORT_Pin_0;          //Выбор нулевого пина PD0
BUT.PORT_PULL_DOWN = PORT_PULL_DOWN_OFF; //Выключаем
подтяжку порта порта к земле
BUT.PORT_PULL_UP = PORT_PULL_UP_OFF; //Выключаем подтяжку
порта порта к питанию
PORT_Init(MDR_PORTD, & BUT);       //Применение настроек к
порту D
}

```

Главная функция для управления дискретным портом ввода имеет следующий вид:

```

int main(void)
{
if ( !PORT_ReadInputDataBit(MDR_PORTD,PORT_Pin_0) )
{
PORT_SetBits(MDR_PORTF, PORT_Pin_0); //Включение «PF0»
while ( ! PORT_ReadInputDataBit(MDR_PORTD,PORT_Pin_0) ) {};
}
else PORT_ResetBits(MDR_PORTF, PORT_Pin_0);
}

```

Компилируем программу и загружаем её в микроконтроллер. Светодиод на плате будет загораться при нажатии на кнопку «UP».

3. Вывод информации на ЖК – дисплей

Часто пользователю требуется получать некоторую визуальную информацию с электронного устройства. Если информация может быть представлена в символьном виде, то одним из вариантов ее

отображения является использование символьных жидкокристаллических индикаторов (ЖКИ, или LCD в зарубежном обозначении).

Типы индикаторов, классифицированные по физическому принципу формирования изображения:

1. Лампа накаливания.
2. Газоразрядная лампа.
3. Электронно-лучевая трубка (люминесценция при бомбардировке электронами).
4. Излучение светодиода (LED, СИД).
5. Поляризация света жидкими кристаллами (LCD, ЖК).

На отладочной плате «Миландр» установлен ЖК индикатор с матрицей MT-12864J. Внутри ЖК-панели находятся два контроллера KS-0108. Поскольку размер матрицы 128x64, то каждый контроллер отвечает за свою половину экрана и активизируется соответствующим пином E1 и E2. **Необходимо учитывать, что запись или чтение сразу в/из обоих контроллеров не допускается!**

На экране можно выводить пользовательские символы. Размер одного символа по горизонтали и вертикали зависит от выбора пользователя. Для создания символа необходимо начертить табличку и закрасить нужные клетки. В закрашенных ставим 1, а в пустых 0. Затем формируем строки в двоичном формате. На рисунке 3.3. показано двоичное представление кириллической буквы «П» и символа «смайлик».

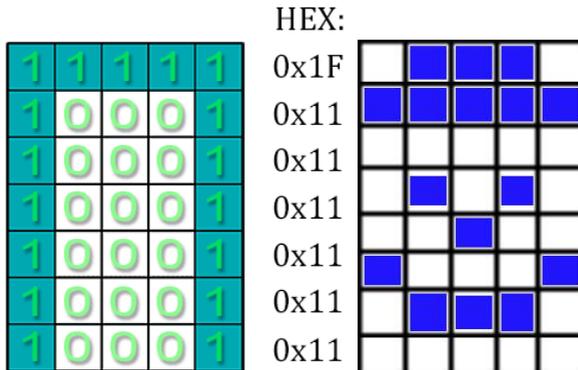


Рис. 3.3. Вывод символа пользователя на ЖК индикатор

Для работы с ЖК индикатором необходимо воспользоваться предоставляемым «Миландр» демо-проектом «EV1986VE3Test.uvproj».

Существует несколько способов управления ЖК-индикатором по параллельному 4-х или 8-ми битному интерфейсу, или, при наличии дополнительного преобразователя, с помощью последовательных интерфейсов (I2C, SPI). На рисунке 3.4. показано подключение ЖК-индикатора к МК на отладочной плате. Назначение выводов индикатора подробно описано в лабораторной работе №1.

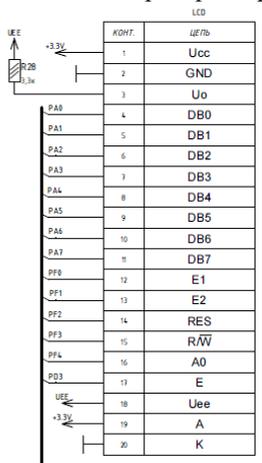


Рис. 3.4. Подключение ЖК-индикатора к микроконтроллеру 1986VE93У

4. Подключение матричной клавиатуры.

При решении некоторых задач необходимо предусмотреть ввод информации. Для решения данной задачи и применяются различные типы клавиатур:

- Одна механическая кнопка
- Несколько механических кнопок
- Матрица кнопок
- Матрица кнопок с отдельным контроллером, например клавиатура персонального компьютера.
- Аналоговые клавиатуры.
- Сенсорные клавиатуры.
- Оптические клавиатуры

Все вышеперечисленные типы клавиатур имеют свои преимущества и недостатки. Например, для построения небольших задач, где не требуется большое количество устройств ввода оптимальным выбором будет использование нескольких кнопок, для более сложных, с целью экономии портов ввода микроконтроллера

необходимо использовать более эффективные решения, одним из которых является матричная клавиатура.

В данной лабораторной работе будем использовать матричную клавиатуру 4x4. Внешний вид приведен на рис 3.5.



Рис. 3.5. Матричная клавиатура 4x4

Данная клавиатура представляет из себя 16 кнопок подключенных как показано на рис 3.6.

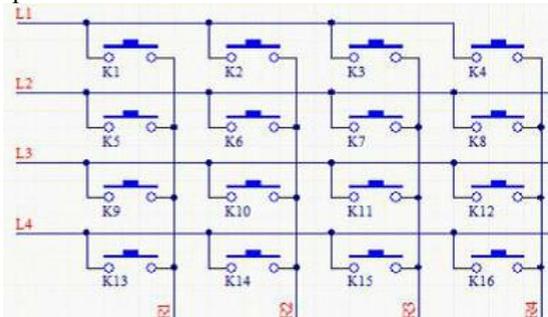


Рис. 3.6. Схема подключения кнопок матричной клавиатуры

Принцип работы матричной клавиатуры достаточно прост, микроконтроллер поочередно подает логическую единицу на каждый из 4-х столбцов, считывая значения на каждом из 4-х входов, и только на одном входе будет высокий уровень.

Если в микроконтроллере нет внутренних подтягивающих резисторов, то для подключения клавиатуры необходимо на каждый вход подключить подтягивающий резистор, чтобы на нем

присутствовал уровень логического нуля, если этого не сделать на выводе микроконтроллера, будет, так называемое «третье состояние», стабильной работы при этом добиться не удастся. Схема подключения приведена на рис 3.7.

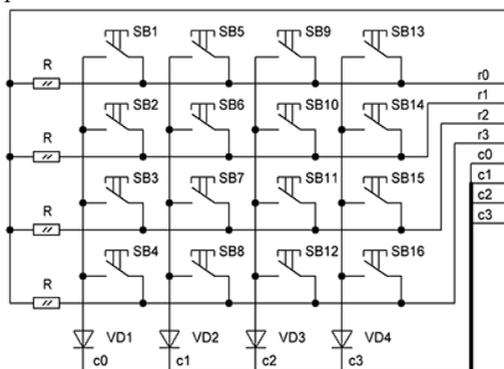


Рис. 3.7. Схема подключения матричной клавиатуры с использованием «подтягивающих» резисторов

Содержание отчета

1. Описание создания нового проекта в среде Keil uVision 5;
2. Схемы подключения светодиодов к микроконтроллеру, листинг программы для управления светодиодом.
3. Схемы подключения кнопок и датчиков с дискретным выходом к микроконтроллеру, листинг программы для работы с кнопкой.
4. Схема подключения ЖК индикатора к микроконтроллеру, листинг программы для работы с ЖК индикатором.
5. Схема подключения матричной клавиатуры к микроконтроллеру, листинг программы для работы с клавиатурой.
6. Выводы.

Контрольные вопросы на защиту

1. Схемы подключения светодиодов к микроконтроллеру, особенности и отличия.
2. Схемы подключения кнопок и датчиков с дискретным выходом к микроконтроллеру, особенности и отличия. Какая схема подключения эффективней и надежней?
3. Основные способы подключения ЖК индикатора к микроконтроллеру, типы ЖК индикаторов, физические принципы построения индикаторов, особенности работы.

Принцип работы графического дисплея МТ-12864J, используемого в отладочных платах фирмы Миландр.

4. Типы клавиатур, их отличия преимущества и недостатки, подключение матричной клавиатуры к микроконтроллеру, особенности работы.

Лабораторная работа №4

Работа с аналоговыми сигналами

Цель работы

Исследование возможностей использования встроенных АЦП и ЦАП 1986ВЕ93У

Основные положения

Аналого-цифровой преобразователь (АЦП) – это устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал), чаще всего – двоичный (рис. 4.1). Обратное преобразование осуществляется при помощи цифро-аналогового преобразователя (ЦАП).

В качестве аналогового сигнала может выступать любая физическая непрерывно меняющаяся величина либо ее эквивалент. Чаще всего в качестве входного сигнала используется эквивалентный сигнал напряжения для получения цифровой информации о температуре, токе, влажности и т.д.

Большинство аналого-цифровых преобразователей являются линейными, то есть диапазон входных значений, отображаемый на выходное цифровое значение, связан по линейному закону с этим выходным значением. Основой построения АЦП является аналоговый компаратор.

Разрешение АЦП – минимальное изменение величины аналогового сигнала, которое может быть преобразовано данным АЦП. Обычно измеряется в вольтах.

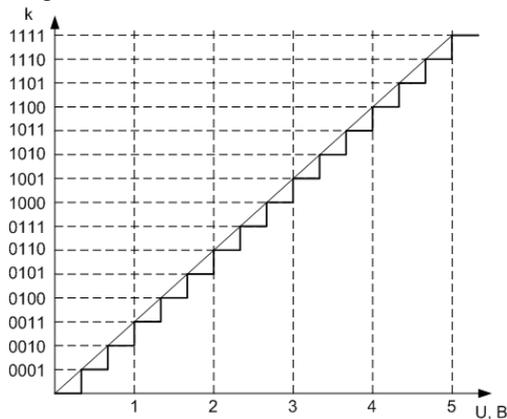


Рис. 4.1. Квантование сигналов в АЦП

Разрядность АЦП характеризует количество дискретных значений, которые преобразователь может выдать на выходе. Измеряется в битах. Например, АЦП, способный выдать $2^8=256$ дискретных значений (0...255), имеет разрядность 8 бит. Разрешение по напряжению равно разности напряжений, соответствующих максимальному и минимальному выходному коду, делённой на количество выходных дискретных значений.

$$d = \frac{U_{\max} - U_{\min}}{2^N},$$

где N – разрядность АЦП.

При этом напряжение на входе преобразователя можно оценить, зная полученное цифровое значение аналого-цифрового преобразования *Value*

$$U = \frac{Value}{d}.$$

На практике разрешение АЦП ограничено отношением сигнал/шум входного сигнала. При большой интенсивности шумов на входе АЦП различение соседних уровней входного сигнала становится невозможным, то есть ухудшается разрешение. При этом реально достижимое разрешение описывается эффективной разрядностью (effective number of bits — ENOB), которая меньше, чем реальная разрядность АЦП. При преобразовании сильно зашумленного сигнала младшие биты выходного кода практически бесполезны, так как содержат шум.

Дискретизацией сигнала называется измерительное преобразование непрерывного сигнала $x(t)$ в последовательность мгновенных значений этого сигнала $X(kT)$ (рис. 4.2), соответствующих определенным моментам времени kT (T – шаг дискретизации).

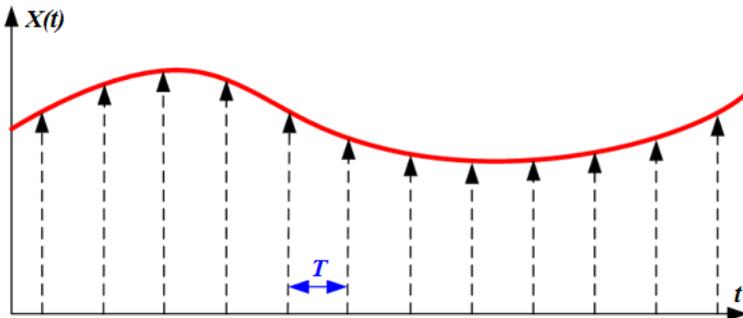


Рис. 4.2. Квантование сигнала по времени

Дискретизацию сигнала по времени можно проводить с постоянным шагом $T = \text{const}$ или с переменным шагом $T = \text{var}$.

Частота дискретизации – частота, с которой производится аналого-цифровое преобразование сигнала.

Время преобразования – время от начала преобразования до появления на выходе АЦП соответствующего кода.

Опорное напряжение – напряжение, соответствующее максимальному выходному коду.

Поскольку реальные АЦП не могут произвести аналого-цифровое преобразование мгновенно, входное аналоговое значение должно удерживаться постоянным, по крайней мере, от начала до конца процесса преобразования (этот интервал времени называют время преобразования). Эта задача может решаться путем использования специальной схемы на входе АЦП – устройства выборки-хранения (УВХ). УВХ, как правило, хранит входное напряжение на конденсаторе, который соединен с входом через аналоговый ключ: при замыкании ключа происходит выборка входного сигнала (конденсатор заряжается до входного напряжения), при размыкании – хранение. Как правило, модули АЦП содержат встроенное УВХ.

В микроконтроллере 1986ВЕ93У реализовано два 12-разрядных АЦП (рис. 4.2). С помощью АЦП можно оцифровать сигнал от 16 внешних аналоговых выводов порта D и от двух внутренних каналов, на которые выводятся датчик температуры и источник опорного напряжения. Скорость выборки составляет до 512 тысяч преобразований в секунду для каждого АЦП.

В качестве опорного напряжения преобразования могут выступать:

- питание АЦП с выводов AUCC и AGND;
- внешние сигналы с выводов ADC0_REF+ и ADC_REF-.

Контроллер АЦП позволяет:

- оцифровать один из 16 внешних каналов;
- оцифровать значение встроенного датчика температуры;
- оцифровать значение встроенного источника опорного напряжения;
- осуществить автоматический опрос заданных каналов;
- выработать прерывание при выходе оцифрованного значения за заданные пределы;
- запускать два АЦП синхронно для увеличения скорости выборки.

Для осуществления преобразования требуется не менее 28 тактов синхронизации CLK. В качестве синхросигнала может выступать

частота процессора CPU_CLK, либо частота ADC_CLK, формируемая в блоке «Сигналы тактовой частоты». Выбор частоты осуществляется с помощью бита Cfg_REG_CLKS. Частота CPU_CLK формируется из частоты процессорного ядра делением на коэффициент Cfg_REG_DIVCLK[3:0]. Максимальная частота CLK не может превышать 14 МГц.

Регистры, используемые для управления АЦП, представлены в таблице 4.1. Для включения АЦП необходимо установить бит Cfg_REG_ADON. С целью снижения тока потребления вместо собственного источника опорного напряжения в АЦП может использоваться источник датчика температуры. Для этого необходимо включить блок датчика температуры и источник опорного напряжения, установив бит TS_EN в единицу. После включения можно использовать источник опорного напряжения для первого и второго АЦП вместо их собственных, при этом достаточно установить биты ADCx_OP в единицу. При преобразовании аналогового сигнала нужно выводы, используемые АЦП у порта D, сконфигурировать как аналоговые и \отключить внутренние подтягивающие резисторы.

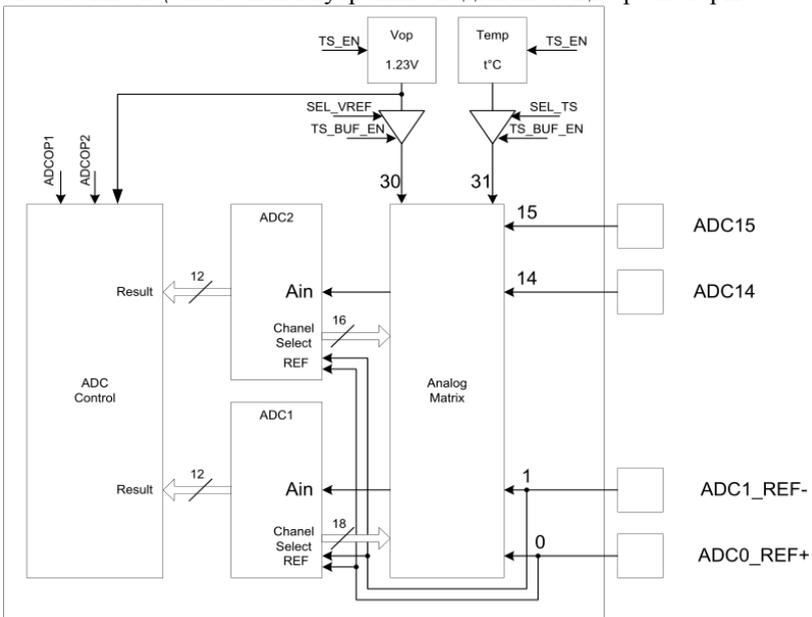


Рис. 4.3. Структурная схема АЦП МК 1896BE9x

Таблица 4.1.

Базовый адрес	Название	Описание
0x4008_8000	MDR_ADC	Контроллер ADC
Смещение		
0x00	MDR_ADC->ADC1_CFG	Регистр управления ADC1
0x04	MDR_ADC->ADC2_CFG	Регистр управления ADC2
0x08	ADC1_H_LEVEL	Регистр MDR_ADC->ADCx_H_LEVEL верхней границы ADC1
0x0C	ADC2_H_LEVEL	Регистр MDR_ADC->ADCx_H_LEVEL верхней границы ADC2
0x10	ADC1_L_LEVEL	Регистр MDR_ADC->ADCx_L_LEVEL нижней границы ADC1
0x14	ADC2_L_LEVEL	Регистр MDR_ADC->ADCx_L_LEVEL нижней границы ADC2
0x18	ADC1_RESULT	Регистр MDR_ADC->ADCx_RESULT результата ADC1
0x1C	ADC2_RESULT	Регистр MDR_ADC->ADCx_RESULT результата ADC2
0x20	ADC1_STATUS	Регистр MDR_ADC->ADCx_STATUS статуса ADC1
0x24	ADC2_STATUS	Регистр MDR_ADC->ADCx_STATUS статуса ADC2
0x28	ADC1_CHSEL	Регистр MDR_ADC->ADCx_CHSEL выбора каналов перебора ADC1
0x2C	ADC2_CHSEL	Регистр MDR_ADC->ADCx_CHSEL выбора каналов перебора ADC2

В микроконтроллере реализовано два ЦАП (рис. 4.3). Регистры, используемые для управления АЦП, представлены в таблице 4.2. Включение ЦАП осуществляется установкой бита `Cfg_ON_DACx` в единицу, а используемые выводы ЦАП порта E должны быть сконфигурированы как аналоговые с отключенными подтягивающими резисторами. Оба ЦАП могут работать независимо или совместно. При независимой работе ЦАП (бит `Cfg_SYNC_A=0`) после записи данных в регистр данных `DACx_DATA` на выходе `DACx_OUT` формируется уровень напряжения, соответствующий записанному значению. При синхронной работе (бит `Cfg_SYNC_A=1`) данные обоих ЦАП могут быть обновлены одной записью в один из регистров `DACx_DATA`. ЦАП может работать от внутренней опоры `Cfg_M_REFx=0`, тогда ЦАП формирует выходной сигнал в диапазоне от 0 до напряжения

питания AU CC . В режиме работы с внешней опорой Cfg_M_REFx=1 ЦАП формирует выходное напряжение в диапазоне от 0 до значения DACx_REF.

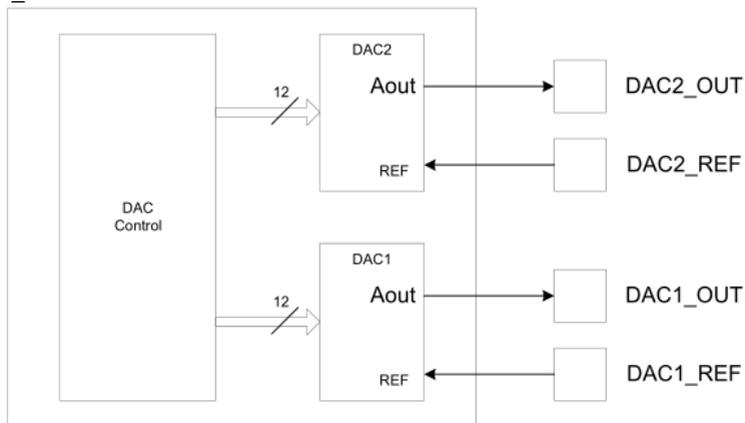


Рис. 4.4. Структурная схема ЦАП МК 1896BE9x

Таблица 4.2.

Базовый адрес	Название	Описание
0x4009_0000	MDR_DAC	Контроллер DAC
Смещение		
0x00	MDR_DAC->CFG	Регистр управления DAC
0x04	MDR_DAC->DAC1_DATA	Регистр данных DAC1
0x08	MDR_DAC->DAC2_DATA	Регистр данных DAC2

или $\frac{U_{\text{вых}}(s)}{U_{\text{вх}}(s)} = W(s)$, где $u_{\text{вх}}, U_{\text{вх}}(s)$ – входное напряжение, $u_{\text{вых}}, U_{\text{вых}}(s)$ – выходное напряжение, снимаемое с выхода ЦАП.

1. Продемонстрировать преподавателю результаты работы.

Содержание отчета

1. Принципиальная электрическая схема для используемой периферии микроконтроллера.
2. Листинг программы.
3. Выводы. $u_{\text{вых}} = f(u_{\text{вх}})$

Контрольные вопросы на защиту

1. Какие типы АЦП и ЦАП существуют? К какому типу принадлежат АЦП и ЦАП в МК?
2. Какими являются максимальные частоты аналоговых сигналов АЦП и ЦАП в МК? Назовите остальные характеристики АЦП и ЦАП для МК 1986ВЕ9х.
3. Какой должна быть частота дискретизации аналогового сигнала? Сформулируйте теорему Котельникова.
4. Начертите схему простейшего 5-разрядного ЦАП с резисторной матрицей.

Лабораторная работа №5

Изучение таймеров общего назначения

Цель работы

Исследование возможностей использования встроенных таймеров 1986VE93У

Основные положения

Таймеры предназначены для формирования временных интервалов, позволяя микропроцессорной системе работать в режиме реального времени. Таймеры представляют собой цифровые счётчики, которые подсчитывают импульсы либо от высокостабильного генератора частоты, либо от внешнего источника сигнала, в этом случае таймер называют счётчиком внешних событий. К системной шине микропроцессора таймеры подключаются при помощи параллельных портов.

Как правило, в микропроцессорной системе в качестве генератора частоты выступает генератор внутренней синхронизации микроконтроллера. Частота генератора задает минимальный временной промежуток, который может определять таймер. Интервалы времени, задаваемые с помощью таймера, могут иметь строго определенные дискретные значения. Разрядность цифрового счётчика таймера определяет максимальный интервал времени, который может задать таймер.

Все блоки таймеров выполнены на основе 16-битного перезагружаемого счетчика, который синхронизируется с выхода 16-битного делителя. Перезагружаемое значение хранится в отдельном регистре. Счет может быть прямой, обратный или двунаправленный (сначала прямой до определенного значения, а затем обратный). Каждый из трех таймеров микроконтроллера содержит 16-битный счетчик, 16-битный делитель частоты и 4-канальный блок захвата/сравнения. Их можно синхронизировать системной синхронизацией, внешними сигналами или другими таймерами. Помимо составляющего основу таймера счетчика, в каждый блок таймера также входит четырехканальный блок захвата/сравнения. Данный блок выполняет как стандартные функции захвата и сравнения, так и ряд специальных функций. Таймеры с 4 каналами схем захвата и ШИМ с функциями формирования «мертвой зоны» и аппаратной блокировки. Каждый из таймеров может генерировать прерывания и запросы DMA.

Состав и особенности таймеров:

- 16-битный счетчик; счёт прямой, обратный или двунаправленный;
- 16-разрядный программируемый предварительный делитель частоты;
- до четырех независимых 16-битных каналов захвата на один таймер, Каждый из которых может захватить (скопировать) текущее значение таймера при изменении некоторого входного сигнала, при этом, в случае захвата имеется дополнительная возможность генерировать прерывание и/или запрос DMA;
- четыре 16-битных регистра сравнения (совпадения), которые позволяют осуществлять непрерывное сравнение, с дополнительной возможностью генерировать прерывание и/или запрос DMA при совпадении;
- имеется до четыре внешних выводов, соответствующих регистрам совпадения со следующими возможностями:
 - сброс в НИЗКИЙ уровень при совпадении;
 - установка в ВЫСОКИЙ уровень при совпадении;
 - переключение (инвертирование) при совпадении;
 - при совпадении состояние выхода не изменяется;
 - переключение при некотором условии.

Таймер предназначен для того, чтобы подсчитывать циклы периферийной тактовой частоты F_{dts} или какие-либо внешние события и произвольно генерировать прерывания, запросы DMA или выполнять другие действия. Значения таймера, при достижении которых будут выполнены те или иные действия, задаются восемью регистрами совпадения. Кроме того, в микроконтроллере имеются четыре входа захвата, чтобы захватить значение таймера при изменении некоторого входного сигнала, с возможностью генерировать прерывание или запрос DMA.

Таймер содержит основной 16-битный счетчик CNT, блок регистров управления и четыре канала схем захвата/ШИМ.

Таймер позволяет работать в режимах:

- таймер;
- расширенный таймер, с объединением нескольких таймеров;
- режим захвата;
- ШИМ.

Режим таймера

Таймеры построены на базе 16-битного счетчика, объединенного с 16-битным предварительным делителем. Скорость счета таймера зависит от значения, находящегося в регистре делителя.

Счетчик может считать вверх, вниз или вверх и вниз (счёт прямой, обратный, двунаправленный).

Базовый блок таймера включает в себя:

- основной счетчик таймера (TIMx_CNT);
- делитель частоты при счете основного счетчика (TIMx_PSC);
- основание счета основного счетчика (TIMx_ARR).

Сигналом для изменения CNT может служить как внутренняя частота TIM_CLK, так и события в других счетчиках, либо события на линиях TxCN_i данного счетчика.

Чтобы запустить работу основного счетчика, необходимо задать:

- начальное значение основного счетчика таймера – TIMx_CNT;
- значение предварительного делителя счетчика – TIMx_PSG, при этом основной счетчик будет считать на частоте CLK, равной $TIMx_CLK/(PSG+1)$;
- значение основания счета для основного счетчика TIMx_ARR;
- режим работы счетчика TIMx_CNTRL, для чего необходимо выбрать:

- выбрать источник события переключения счетчика EVENT_SEL;
 - режим счета основного счетчика CNT_MODE (значения 00 и 01 при тактировании внутренней частотой, значения 10 и 11 при тактировании внешними сигналами);
 - направление счета основного счетчика DIR;
- разрешить работу счетчика CNT_EN.

По событиям совпадения значения основного счетчика с значением нуля или значением основания счета генерируется прерывание и запрос DMA, которые могут быть замаскированы.

Режим ШИМ

Включение режима сравнения для определенного канала осуществляется путем записи «0» в поле CAPnPWM регистра управления каналом CN_y_CNTRL. При работе в режиме ШИМ выходной сигнал может формироваться на основании сравнения значения в регистре CCR и значения основного счетчика CNT, или значения регистров CCR, CCR1 и также значения основного счетчика CNT, Полученный сигнал может без изменения выдаваться на выходы CNxO и nCNxO. Либо с применением схемы DEAD TIME Generator формируются управляющие сигналы с мертвой зоной. У каждого канала есть два выхода – прямой и инверсный. Для каждого выхода формируется как сигнал для выдачи, так и сигнал разрешения выдачи, т.е. если выход канала должен всегда выдавать тот или иной уровень, то на выводе разрешения выдачи CNxOE (для прямого) и на CNxNOE (для инверсного) должны формироваться единицы. Если канал работает на вход (например, режим захвата), то там всегда должен

быть "0" для прямого канала. Сигналы OE формируются по тем же принципам, что и просто выходные уровни, но у них есть собственные сигналы разрешения вывода SELOE и nSELOE, в которых можно выбрать постоянный уровень, либо формируемый на основании REF.

Порядок выполнения лабораторной работы

1. Ознакомиться с особенностями конфигурирования режимов работы таймеров с использованием средств CMSIS и SPL Library.
2. Подать сигнал с функционального генератора сигналов на вход АЦП отладочной демоплаты (**Необходимо согласовать амплитуду сигнала с опорным напряжением на выводах ADC_REF- и ADC_REF+, в противном случае возможно повреждение оборудования!**).
3. Создать новый проект в среде Keil MDK и реализовать измерение частоты входного сигнала и заданную преподавателем функциональность.
4. Вывести результаты на LCD дисплей.
5. Продемонстрировать преподавателю результаты работы.

Содержание отчета

1. Принципиальная электрическая схема для используемой периферии микроконтроллера.
2. Листинг программы.
3. Выводы.

Контрольные вопросы на защиту

1. Назовите режимы работы таймера МК 1986BE9x.
2. Как зависят от частоты тактирования МК максимальная и минимальная частоты работы таймера?
3. Для каких задач следует использовать режим ШИМ?
4. Как реализованы программные таймеры OSCP? Приведите пример использования и настройки.

Лабораторная работа №6

Обмен информацией по интерфейсу USB

Цель работы

Изучение принципов последовательной передачи данных с использованием универсальной шины USB;

Основные положения

Существует ряд различных спецификаций USB: от 0.7(выпущена в 1994 году) до 3.1 (2013 год). В этой лабораторной работе будет использоваться наиболее популярный на текущее время USB 2.0.

В этой спецификации регламентируется три режима работы:

- *Low-speed* 10 – 1500 Кбит/с (клавиатуры, мыши, прочие устройства ввода),
- *Full-speed* 0.5 – 12 Мбит/с (аудио- и видеоустройства),
- *High-speed* 25 – 480 Мбит/с (видеоустройства, устройства хранения информации).

USB обеспечивает взаимодействие между хостом (от англ. host — «хозяин, принимающий гостей») и подключенными устройствами (attachment devices). Хост является инициатором обмена данными, в то время как устройство лишь отвечает на запросы хоста.

Вся информация передается кадрами, которые отправляются через равные промежутки времени. Каждый кадр включает в себя пакет SOF (Start of Frame), транзакции и EOF (End of Frame) (рис 6.1).

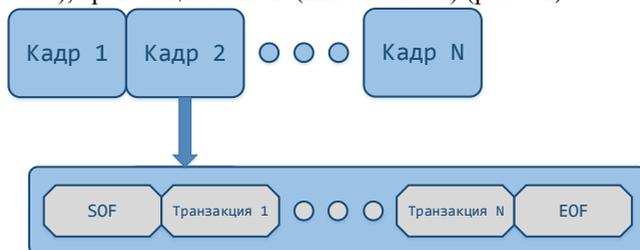


Рис. 6.1. Передача данных по протоколу USB

Транзакция – это логическая единица обмена данными, которая в свою очередь состоит из следующих пакетов (рис. 6.2):

- Идентифицирующий пакет (Token packet), в котором содержится информация об идентификаторе пакета, адресе устройства на шине USB, номере конечной точки и контрольной сумме.

- Пакет данных (Data packet), состоящий из идентификатора пакета, передаваемых данных и контрольной суммы.
- Статус пакет (Handshake packet), состоящий только из идентификатора пакета, значение которого отражает состояние передачи транзакции.



Рис. 6.2. Транзакция в протоколе USB

Конечная точка (Endpoint) – это часть **USB-устройства**, которая имеет уникальный идентификатор и является получателем или отправителем информации, передаваемой по шине USB. Другими словами, это буфер, сохраняющий несколько байт. Данные, которые хранятся в конечной точке, могут быть либо принятыми данными, либо данными, ожидающими передачу. Хост также имеет буфер для приема и передачи данных, но хост не имеет конечных точек.

Конечная точка имеет следующие основные параметры:

- частоту доступа к шине;
- допустимую величину задержки обслуживания;
- требуемую ширину полосы пропускания канала;
- номер конечной точки;
- способ обработки ошибок;
- максимальный размер пакета, который конечная точка может принимать или отправлять;
- используемый конечной точкой тип посылок;
- направление передачи данных.

Любое USB-устройство имеет конечную точку с нулевым номером (Endpoint Zero). Эта точка позволяет хосту опрашивать устройство с целью определения его типа и параметров, выполнять инициализацию и конфигурирование устройства (рис. 6.3).

Кроме нулевой точки, устройства обычно имеют дополнительные конечные точки, которые используются для обмена данными с хостом. Дополнительные точки могут работать либо только на прием данных от хоста (входные точки, IN), либо только на передачу данных хосту (выходные точки, OUT). Число дополнительных конечных точек устройств определяется режимом передачи. Для низкоскоростных устройств допускается наличие одной или двух дополнительных конечных точек, а для высокоскоростных – до 15 входных и 15 выходных дополнительных точек.

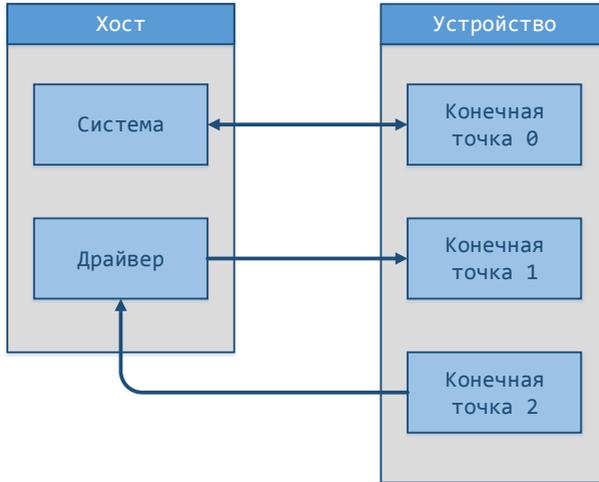


Рис. 6.3. Взаимодействие между хостом и устройствами USB

Нулевая точка устройства доступна после того, как устройство подключено к шине, включено и получило сигнал сброса по шине (bus reset). Все остальные конечные точки после включения питания или сброса находятся в неопределенном состоянии и недоступны для работы до тех пор, пока хост не выполнит процедуру конфигурирования устройства. В зависимости от вида обрабатываемых передач, различают четыре вида конечных точек:

Управляющие передачи (control transfers) — небольшие, как правило, передачи, используемые для конфигурирования, управляющих команд и запроса статуса (рис. 6.4). Это единственный тип двунаправленного обмена – здесь передается информация и от хоста к устройству, и от устройства к хосту.

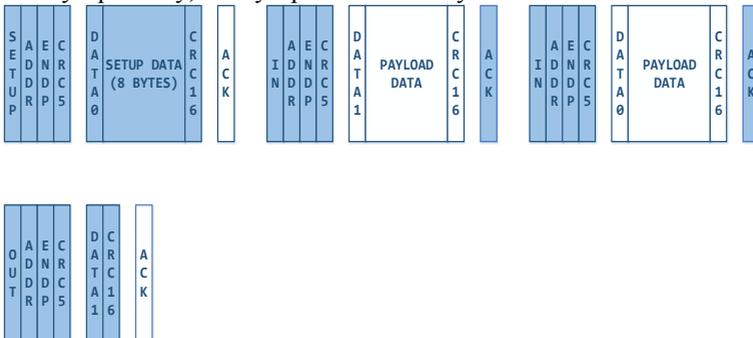


Рис. 6.4. Структура данных управляющей передачи

Передачи массивов данных (bulk transfers) – используется для обмена данными (рис. 6.5). Гарантирование целостности данных и гарантированная доставка данных для данного типа канала, реализованного «в железе». Однако скорость передачи данных по такому каналу ограничена.



Рис. 6.5. Передача массивов данных

Изохронные передачи (isochronous transfers) – тип, не гарантирующий доставку данных (рис. 6.6). Используется когда данных много и важно, чтобы они были обработаны за определённое время, но допускаются потери данных, например, мультимедиа: веб-камеры, USB-колонки.

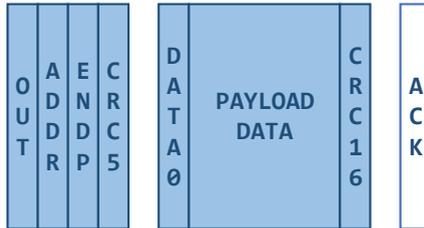


Рис. 6.6. Изохронная передача данных

Передачи по прерыванию (interrupt transfers) – используются для реализации подобия «прерываний» (рис. 6.7). Такие «прерывания» являются логическими, и никак напрямую не связаны с аппаратными прерываниями МК или прерываниями ОС.

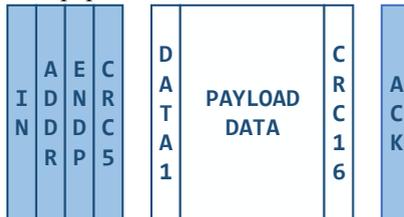


Рис. 6.7. Передача данных по прерыванию

Время на шине USB измеряется в кадрах или микрокадрах (рис. 6.8). Кадры появились в USB 1, один кадр — одна миллисекунда. Микрокадры появились в USB 2, в одном кадре 8 микрокадров.

Инфраструктура USB планирует выполнение передач по прерыванию и изохронных передач на конкретные (микро) кадры так, чтобы гарантировать запрошенный интервал времени между передачами. Планирование не имеет права использовать более 90% от кадра или более 80% от микрокадра. Оставшееся время (минимум 10%/20%, хотя может быть и больше, если распланировано не всё время) занимают активные управляющие передачи и, по остаточному принципу, передачи массивов данных.

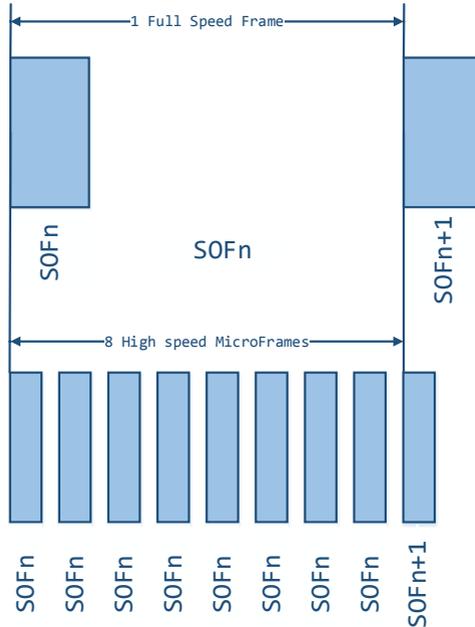


Рис. 6.8. Поток кадров USB

Для конфигурации и работы с USB понадобятся следующие библиотеки: RST_CLK, TIMER, USB Library, USB.

Установка драйвера виртуального COM-порта

Открыть панель управления и перейти в диспетчер устройств. При первом подключении плата обнаружится как неизвестное устройство.

Для установки драйвера необходимо перейти в **Свойства** устройства на вкладку **Драйвер** и нажать **Обновить**.

После чего выбрать пункт **Выполнить поиск драйверов на этом компьютере**. Далее выбрать пункт **Выбрать из списка уже установленных драйверов**. После чего выбрать тип оборудования **Порты (COM и LPT)**. Выбрать в поле **Изготовитель (Стандартные**

порты), а в поле *Модель* выбрать *Последовательный порт* и нажать кнопку *Установить с диска*, где в диалоговом окне выбрать файл из раздаточного материала **MDRVComport.inf**.

Создание приложения

Для работы с устройством написать код на стороне хоста (ПК). Для этого в комплект раздаточного материала входит библиотека для работы с COM портом в асинхронном режиме. Она состоит из файла-заголовка (*.h), файла статически скомпонованной библиотеки (*.lib), и файла динамически скомпонованной библиотеки (*.dll). Для подключения библиотеки нужно включить заголовочный файл в проект, указать программе-линковщику на расположение *.lib файла и скопировать *.dll файл в директорию с исполняемым файлом.

Для записи в порт используется функция `void write(void const* write_buf, size_t size, ok_callback ok_func, error_callback error_func);` где параметрами являются:

`write_buf` – указатель на область памяти, в которой расположено сообщение для отправки,

`size` – размер сообщения для отправки (в байтах),

`ok_func` – функция, которая будет вызвана в случае успешной передачи,

`error_func` – функция, которая будет вызвана в случае ошибки.

Для чтения из порта используется функция `void read(void* read_buf, size_t size, ok_callback ok_func, error_callback error_func);`

`read_buf` – указатель на область памяти, в которую будет помещено сообщение после приема,

`size` – размер сообщения для отправки (в байтах),

`ok_func` – функция, которая будет вызвана в случае успешного приема,

`error_func` – функция, которая будет вызвана в случае ошибки,

Функция чтения будет ожидать сообщения размера `size` в течении времени, которое задается функцией `void set_timeout(const unsigned int& value)`, где `value` – время ожидания (в секундах).

Функции `read` и `write` являются композитными, поэтому нельзя повторно вызывать `read` и `write` до тех пор, пока не сработали функции `callback` предыдущих вызовов (Смотри исходный код библиотеки и документацию по `boost::asio::async_write/read`).

Порядок выполнения лабораторной работы

1. Ознакомиться с особенностями конфигурирования режимов работы модуля USB МК с использованием средств CMSIS и SPL Library.
2. Создать новый проект в среде Keil MDK и реализовать обмен данными с ПК.
3. Продемонстрировать преподавателю результаты работы.

Содержание отчета

1. Принципиальная электрическая схема для используемой периферии микроконтроллера.
2. Листинг программы.
3. Выводы.

Контрольные вопросы на защиту

1. Каковы возможности МК серии 1986BE9x для передачи данных?
2. Как организуется передача данных по интерфейсу USB?
3. Какие вы знаете интерфейсы последовательной и параллельной передачи данных, в чем специфика их применения?
4. В каких режимах возможно подключение МК по шине USB?

Лабораторная работа №7

Исследование динамики цифровых систем автоматической стабилизации на базе МК 1986ВЕ93У

Цель работы

1. Исследование цифровой системы автоматической стабилизации угловой скорости вращения вала электродвигателя постоянного тока в статическом и динамическом режимах.
2. Овладение навыками экспериментального снятия статических характеристик и определения передаточных коэффициентов элементов исследуемой системы с использованием микроконтроллера.
3. Экспериментальное определение динамических характеристик системы автоматической стабилизации.
4. Реализация типовых цифровых законов управления на микроконтроллере 1986ВЕ93У фирмы Миландр.

Основные положения и описание лабораторного макета

В систему автоматической стабилизации угловой скорости вращения вала электродвигателя постоянного тока входят:

- микроконтроллер «МК» 1986ВЕ93У, который является управляющим устройством;
- отладочная плата производства фирмы Миландр, выполняющая функции отображения и задатчика;
- двигатель постоянного тока с независимым возбуждением М, являющийся объектом управления;
- тахогенератор постоянного тока или инкрементальный энкодер, предназначенные для преобразования действительного значения скорости вращения вала двигателя М в пропорциональную ЭДС постоянного тока или количество импульсов за единицу времени;
- генератор постоянного тока G и потенциометр RP1, с помощью которых имитируется механическая нагрузка.

Основные характеристики электродвигателя М (СЛ-221А) указаны в табл. 7.1.

Таблица 7.1

Технические характеристики двигателя СЛ-221А

$U_{ном}$ В	$P_{ном}$ Вт	$n_{ном}$ об/мин	$M_{ном}$ мН м
110	13	3600	34

В качестве генератора G могут выступать электродвигатели СЛ221А или СЛ369, основные характеристики которых указаны в табл. 7.1 и 7.2 соответственно.

Таблица 7.2

Технические характеристики двигателя СЛ-369

$U_{ном}$, В	$P_{ном}$, Вт	$n_{ном}$, об/мин	$M_{ном}$, мН м
110	55	3600	147

Конструктивно один выход вала электродвигателя М соединен с валом генератора G, а второй с валом инкрементального энкодера или тахогенератора постоянного тока. Генератор постоянного тока подключается с помощью модуля управления генератором (см. принципиальную схему на рис. 7.1) МНГ содержащим электромагнитное реле 1393219-3 [6], служащее для реализации возможности осуществления ступенчатого возмущающего воздействия при замыкании тумблера SB1. Уровень нагрузочного воздействия задается с помощью потенциометра RP1 и измеряется по параметрам вольтметра V и амперметра А. Функцию усилительно-преобразовательного устройства УПУ выполняет полупроводниковый усилитель, выполненный в виде модуля МДУ1, принципиальная схема которого представлена на рис. 7.1. В нем для гальванической развязки используется транзисторная оптопара 4N33 [7], а в качестве силового ключа выступает IGBT-транзистор IRGB14C40LPBF [8]. В МК 1986BE93У выполняются функция обработки информации с инкрементального энкодера и организации управляющего воздействия. В схеме обработки сигнала с энкодера конденсатор С1 используется для уменьшения влияния дребезга, а резистор R6 для подтягивания к линии питания. С помощью отладочной платы фирмы Миландр реализуется устройство индикации текущей скорости вращения (дисплей) и заданной, а также задающее устройство (при помощи кнопок отладочной платы или матричной клавиатуры). Также микроконтроллер используется для идентификации элементов системы по п. 2. и п. 3 плана работы.

Функциональная схема и структурная схемы исследуемой цифровой системы изображены на рис. 7.2 и рис. 7.3 соответственно.

Объект управления ОУ – электродвигатель постоянного тока СЛ-221А представлен в виде модели третьего порядка [9-12], в которой возмущающим воздействием является момент сопротивления M_c , зависящий от тока якоря генератора постоянного тока – $I_{яг}$. Усилительно-преобразовательное устройство УПУ – полупроводниковый усилитель и измерительно-преобразовательное устройство ИзПУ – инкрементальный энкодер представлены в виде квазилинейных безынерционных элементов, поскольку частота изменения входных сигналов энкодера и УПУ на порядок больше, чем

частота изменения входа и выхода объекта управления соответственно.

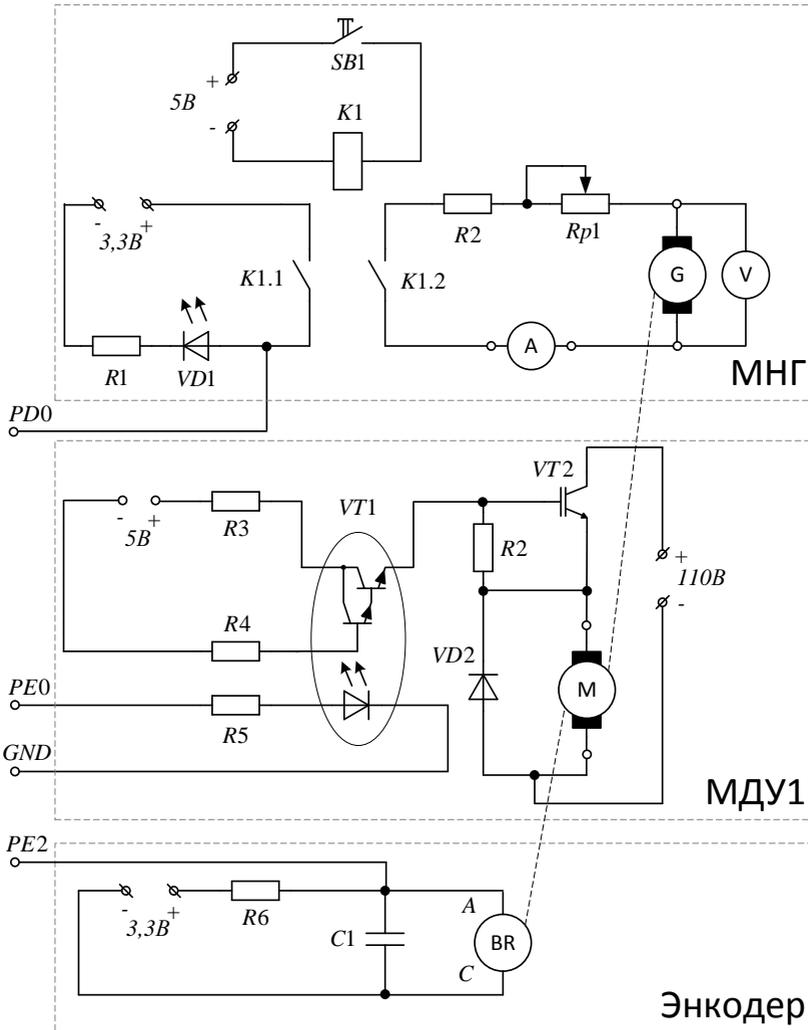


Рис. 7.1. Принципиальная схема лабораторного макета

В лабораторной работе используются опыт и наработки по предыдущим лабораторным работам (№3, №5, №6). Для получения информации о частоте вращения при помощи инкрементального

оптического энкодера необходимо в микроконтроллере МК реализовать подсчет количества импульсов и времени их прохождения и поделить на количество импульсов приходящихся на один оборот его вала [13-14], т.е.

$$\omega = 2\pi \frac{n}{N}, \quad (1)$$

где n – подсчитанное количество импульсов в секунду, N – количество импульсов за один оборот энкодера (паспортное значение), ω – круговая частота [рад/с].

На микроконтроллере также необходимо реализовать ШИМ выход для управления средним напряжением якорной обмотки двигателя $U_{д.ср.}$.

Идентификация статической характеристики разомкнутой части системы (УПУ + ОУ + ИзПУ) осуществляется путем подачи различных значений скважности γ импульсов с микроконтроллера и определения частоты вращения вала ОУ.

Для идентификации динамической части объекта управления необходимо осуществить обмен информацией по порту USB с персональным компьютером, в котором реализуется регистрация состояния входных и выходных сигналов в текстовой или графической форме, начиная с момента подачи ступенчатого воздействия, регистрируемого при возникновении сигнала начала воздействия на порту PD0 на МК. При этом необходимо фиксировать ток и напряжение якоря генератора G. При наличии реакции на ступенчатое воздействие можно выполнить идентификацию динамической части ОУ различными методами [15-17].

Стабилизация частоты вращения электродвигателя осуществляется при помощи устройства управления на базе микроконтроллера МК.



Рис. 7.2. Функциональная схема цифровой системы автоматической стабилизации скорости вращения:
 ЗД – задатчик; УО – устройство отображения; УУ – устройство управления; Пр – преобразователь;
 ИзПУ – измерительно-преобразовательное устройство; ОУ – объект управления;
 ГПТ – генератор постоянного тока

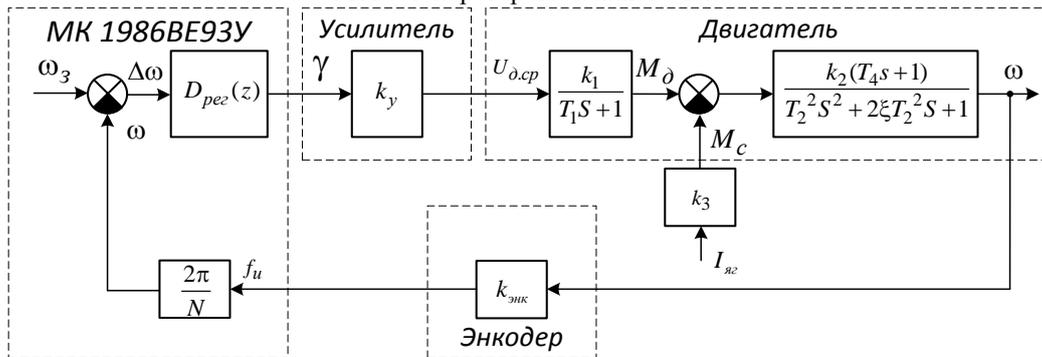


Рис. 7.3. Структурная схема системы

Наиболее распространенными корректирующими алгоритмами в цифровых системах являются [18], как и в непрерывных системах пропорциональный (П), пропорционально-интегральный (ПИ) и пропорционально-интегрально-дифференциальный (ПИД). Покажем способы их реализации. Пропорциональный алгоритм управления имеет вид

$$u[k] = c_0 \varepsilon[k],$$

а передаточная функция $D(z) = c_0$.

Пропорционально-интегральный алгоритм (ПИ) соответствует непрерывному закону управления вида

$$u(t) = \varepsilon(t) + \frac{1}{T_H} \int_0^t \varepsilon(\tau) d\tau, \quad (7.1)$$

где T_H – постоянная времени интегрирования.

При малом периоде дискретизации T_0 и реализации процедуры интегрирования методом прямоугольников уравнение (7.1) можно записать в виде конечно-разностного уравнения:

для k -го такта квантования

$$u[k] = \varepsilon[k] + \frac{T_0}{T_H} \sum_{i=0}^k \varepsilon[i-1]; \quad (7.2)$$

а для $[k-1]$ -го такта квантования

$$u[k-1] = \varepsilon[k-1] + \frac{T_0}{T_H} \sum_{i=0}^{k-1} \varepsilon[i-1]. \quad (7.3)$$

Вычитая из (7.2) выражение (7.3), получим

$$u[k] - u[k-1] = \varepsilon[k] + \left(\frac{T_0}{T_H} - 1 \right) \varepsilon[k-1], \quad (7.4)$$

поскольку разность сумм, отличающихся одним слагаемым при $i=k$ дает значение

$$\frac{T_0}{T_H} \varepsilon[k-1] = \frac{T_0}{T_H} \sum_{i=0}^k \varepsilon[i-1] - \frac{T_0}{T_H} \sum_{i=0}^{k-1} \varepsilon[i-1].$$

Теперь на основании (7.4) можно записать рекуррентное выражение корректирующего цифрового пропорционально-интегрального (изодромного) алгоритма управления

$$u[k] = u[k-1] + \varepsilon[k] + \left(\frac{T_0}{T_H} - 1 \right) \varepsilon[k-1]. \quad (7.5)$$

Передаточная функция изодромного алгоритма управления имеет вид:

$$D_{pez}(z) = \frac{c_0 + c_1 z^{-1}}{1 - z^{-1}}, \quad (7.6)$$

где коэффициенты $c_0=1$, $c_1 = \frac{T_0}{T_H} - 1$.

ПИД-корректирующий алгоритм управления в дискретной форме соответствует непрерывному ПИД-закону управления, описываемому выражением вида

$$u(t) = \varepsilon(t) + \frac{1}{T_H} \int_0^t \varepsilon(\tau) d\tau + T_D \frac{d\varepsilon}{dt}. \quad (7.7)$$

Здесь T_D – постоянная времени дифференцирования. Перейдем при выполнении тех же условий, что и для непрерывного изодромного закона управления, к конечно-разностным уравнениям:

для k -того такта

$$u[k] = \varepsilon[k] + \frac{T_0}{T_H} \sum_{i=0}^k \varepsilon[i-1] + \frac{T_D}{T_0} (\varepsilon[k] - \varepsilon[k-1]); \quad (7.8)$$

а для $[k-1]$ -го такта квантования

$$u[k-1] = \varepsilon[k-1] + \frac{T_0}{T_H} \sum_{i=0}^{k-1} \varepsilon[i-1] + \frac{T_D}{T_0} (\varepsilon[k-1] - \varepsilon[k-2]). \quad (7.9)$$

Из нерекуррентных конечно-разностных уравнений (7.8) и (7.9) получим рекуррентное конечно-разностное уравнение путем вычитания из (7.8) выражения (7.9) и последующего переноса значения управления на предшествующем шаге $u[k-1]$ в правую часть равенства. Тогда запишем рекуррентную формулу для ПИД-корректирующего алгоритма

$$u[k] = u[k-1] + c_0 \varepsilon[k] + c_1 \varepsilon[k-1] + c_2 \varepsilon[k-2], \quad (7.10)$$

где $c_0 = 1 + \frac{T_D}{T_0}$, $c_1 = -\left(1 + 2\frac{T_D}{T_0} - \frac{T_0}{T_H}\right)$, $c_2 = \frac{T_D}{T_0}$.

Применяя к выражению (7.10) Z-преобразование, найдем дискретную передаточную функцию цифрового ПИД-корректирующего алгоритма как отношения $\frac{U(z)}{E(z)}$

$$D_{pec}(z) = \frac{U(z)}{E(z)} = \frac{c_0 + c_1 z^{-1} + c_2 z^{-2}}{1 - z^{-1}}. \quad (7.11)$$

Порядок выполнения лабораторной работы

1. Изучить функциональную и структурную схемы системы автоматической стабилизации и принципиальную схему лабораторной установки ее исследования.

В данном пункте особое внимание уделить схемотехнике подключения модулей к микроконтроллеру и порядку дальнейшей работы по идентификации.

2. Снять и обработать статические характеристики элементов автоматической системы (ИзПУ, УПУ, ОУ) при помощи МК 1986VE93У фирмы Миландр. Задание и отображение информации реализовать с помощью отладочной платы.

3. Учитывая опыт передачи данных, полученные в лаб. раб. №6, снять и обработать кривые разгона элементов автоматической системы (УПУ+ОУ+ИзПУ) при помощи МК 1986VE93У фирмы Миландр;

4. Реализовать цифровые законы управления на микроконтроллере 1986VE93У фирмы Миландр (по заданию преподавателя реализуется П, ПИ, ПИД регуляторы, метод Циглера-Никольса для автонастройки параметров ПИД-регулятора и т.д.).

5. Отладка законов управления, включающая настройку параметров регулятора.

6. Представление результатов работы цифровой системы стабилизации.

Содержание отчета

1. Цель и план работы.
2. Графики и математическое описание статических и динамических характеристики элементов автоматической системы.
3. Код программы, реализующей управление, сбор и обработку данных, и их вывод на дисплей.
4. Определение показателей качества работы системы.
5. Выводы.

Контрольные вопросы

1. Расскажите принцип действия системы и порядок выполнения лабораторной работы.

2. Назовите элементы исследуемой автоматической системы. На основе каких физических принципах действия они работают?

3. К каким портам ввода/вывода можно подключить каналы исследуемой автоматической системы?

4. Зачем нужна гальваническая развязка?

5. Какие математические модели имеют элементы рассматриваемой автоматической системы?

6. Объясните влияние периода повторения дискретного элемента на устойчивость систем.

7. Из каких условий выбирается значение периода повторения квантователя?
8. Как запрограммировать ШИМ-выход микроконтроллера?
9. Для чего нужно использовать подтягивающий резистор?
10. Как реализуются цифровые законы управления на микроконтроллере?

Библиографический список

1. Джозеф Ю. Ядро Cortex-M3 компании ARM: полное руководство : [перевод]. - Додэка-XXI, 2012 — с.535.
2. Серия 1986BE9x, K1986BE9x, MDR32F9Qx, K1986BE91H4, высокопроизводительных 32-х разрядных микроконтроллеров на базе процессорного ядра ARM Cortex-M3. Спецификация микроконтроллеров серии 1986BE9x, K1986BE9x и MDR32F9Qx - © ЗАО «ПКК Миландр» - Версия 3.2.0 от 20.09.2012
URL: "ftp://student:@esau.tusur.ru/_For_Students/MPSSAU/Milandr/Микроконтроллеры и микропроцессоры/1986/spec_seriya_1986BE9x.pdf"
3. Магда Ю.С. Программирование и отладка C/C++ приложений для микроконтроллеров ARM. — М:2012
4. Cortex™-M3 Technical Reference Manual - Revision: r1p1 - ARM© 2005, 2006 - 384 с.
5. Жидкокристаллический модуль МТ–2864J. [Электронный ресурс] — Техническая документация. – Режим доступа: <http://www.melt.com.ru/docs/MT-12864J.pdf>
6. Электромагнитное реле 1393219-3. Паспорт [Электронный ресурс] Режим доступа: <http://lib.chipdip.ru/145/DOC000145173.pdf>
7. Транзисторная оптопара 4N33. Паспорт [Электронный ресурс] Режим доступа: <http://lib.chipdip.ru/043/DOC000043568.pdf>
8. IGBT-транзистор IRGB14C40LPBF. Паспорт [Электронный ресурс] Режим доступа: <http://lib.chipdip.ru/205/DOC000205494.pdf>
9. Справочник по электрическим машинам: В 2 т. Т. 2/С74 Под общ. ред. И.П. Копылова, Б.К. Клокова. – М.: Энергоатомиздат, 1989. – 688 с.: ил. – ISBN 5-283-00531-3 (Т. 2)
10. Брускин, Д.Э. Электрические машины и микромашины: Учеб. для электротехн. спец. вузов / Д.Э. Брускин, А.Е. Зорохович, В.С. Хвостов. – 3-е изд., перераб. и доп. – М.: Высш. шк., 1990. –528 с.: ил. – ISBN 5-06-000725-1
11. Волков, Н.И. Электромашинные устройства автоматики / Н.И. Волков, В.П. Миловзоров; Учеб. для вузов по спец. «Автоматика и телемеханика». – 2-е изд., перераб. и доп. – М.: Высш. шк., 1986. – 335 с.: ил.
12. Подлесный, Н.И. Элементы систем автоматического управления и контроля / Н.И. Подлесный, В.Г. Рубанов – Киев.: Политехника, 1991. – 461 с. - ISBN 5-11-002445-6

13. Янг, Г. Использование ПЛИС в системе управления двигателями с квадратурным энкодером / Глен Янг // Электронные компоненты – 2007.
14. Жданкин, В. Поворотные шифраторы. Основные типы и некоторые особенности применения / В. Жданкин // Компоненты и технологии. – 2001. - №8. – С. 90-96.
15. Магергут, В.З. Выбор промышленных регуляторов и расчет из оптимальных настроек: монография / В.З. Магергут, Д.П. Вент, И.А. Кацер. – Белгород: Изд-во БГТУ, 2009. – 239 с.
16. Копелович, А.П. Инженерные методы расчета при выборе автоматических регуляторов / А. П. Копелович. – М.: Металлургиздат, 1960. – 190 с.
17. Ljung, L. System Identification Toolbox. User Guide, MathWorks, 1997, 274 С
18. Рубанов, В. Г. Теория линейных систем автоматического управления: учеб. пособие, / В. Г. Рубанов. Белгород: Изд-во БГТУ, 2015. – 209 с.