

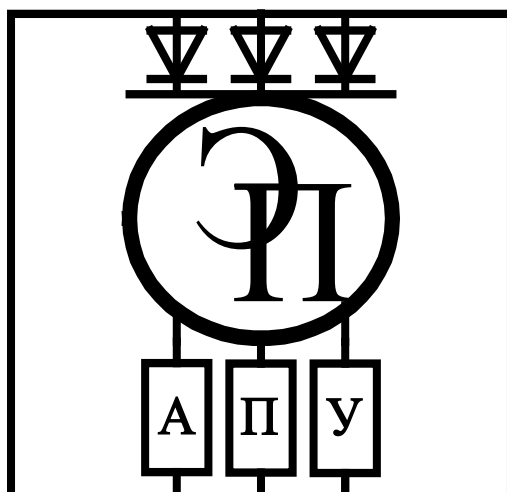
ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и АПУ»

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ АВТОМОБИЛЕЙ И ТРАКТОРОВ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки
13.03.02 «Электроэнергетика и электротехника»
дневной формы обучения*

Часть 2



Могилев 2018

УДК 629.113
ББК 39.33
М 59

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Электропривод и АПУ» «б» февраля 2018 г.,
протокол № 7

Составитель ст. преподаватель В. Т. Вишнеревский

Рецензент канд. техн. наук, доц. Б. Б. Скарыно

Методические рекомендации к лабораторным работам по дисциплине
«Микропроцессорные системы автомобилей и тракторов» предназначены для
студентов направления подготовки 13.03.02 «Электроэнергетика и
электротехника» дневной формы обучения.

Учебно-методическое издание

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ АВТОМОБИЛЕЙ И ТРАКТОРОВ

Часть 2

Ответственный за выпуск Г. С. Леневский

Технический редактор А. А. Подошевка

Компьютерная верстка М. М. Дударева

Подписано в печать. . Формат 60x84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 46 экз. Заказ №

Издатель и полиграфическое исполнение:

Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».

Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.

Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

1 Лабораторная работа № 6. Изучение организации жидкокристаллической индикации в микропроцессорных устройствах.....	4
2 Лабораторная работа № 7. Изучение организации широтно-импульсной модуляции в микропроцессорных устройствах.....	8
3 Лабораторная работа № 8. Изучение организации клавиатуры в микропроцессорных устройствах.....	16
4 Лабораторная работа № 9. Изучение организации устройства ввода аналоговой информации в микропроцессорных устройствах	18
5 Лабораторная работа № 10. Изучение организации устройства вывода аналоговой информации в микропроцессорных устройствах.....	22
6 Лабораторная работа № 11. Изучение организации последовательных портов в микропроцессорных устройствах.....	28
7 Лабораторная работа № 12. Разработка и отладка программ управления технологическими объектами в реальном времени.....	37
Список литературы	43



1 Лабораторная работа № 6. Изучение организации жидкокристаллической индикации в микропроцессорных устройствах

1.1 Ход работы

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.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- применение жидкокристаллических знаковосинтезирующих индикаторов;
- инициализация жидкокристаллических знаковосинтезирующих индикаторов;
- вывод информации на жидкокристаллические знаковосинтезирующие индикаторы.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде uVision;
- загрузить программу в учебный стенд;
- исследовать программную и аппаратную реализацию жидкокристаллической индикации, режимы работы индикаторов и порядок программирования микроконтроллера в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

1.2 Основные теоретические сведения

В работе используется жидкокристаллический индикатор Volymin BC1604A-GPLCW.

Жидкокристаллический модуль BC1604A представляет собой алфавитно-цифровой дисплей. Тип сегментов – матричный. Алфавитно-цифровые (или символьные) индикаторы в своем составе обязательно имеют знаковогенератор, 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).

Перед началом работы с индикатором он должен быть проинициализирован.



1.3 Схема электрическая принципиальная к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

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...PG.10 (R/W, RS и E соответственно).

1.4 Написание программы для микроконтроллера STM32 с использованием жидкокристаллического индикатора

После создания проекта в среде STM32 CubeMX его следует открыть в среде разработки программ KeilVision5 и произвести следующие действия.

В директории проекта создать следующие файлы:

- 1) ..\Inc\main.h
- 2) ..\Src\lcd.c
- 3) ..\Src\lcd.h

Содержимое файла main.h должно быть следующим:

```
#ifndef MAIN_H_
#define MAIN_H_

#include "stm32f4xx.h"

#include "main.h"

#endif /* MAIN_H_ */
```

Содержимое файла lcd.h должно быть следующим:

```
#ifndef LCD_H_
#define LCD_H_
//-----
#include "stm32f4xx_hal.h"
//-----
#define d0_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_0, GPIO_PIN_SET)
#define d1_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_1, GPIO_PIN_SET)
#define d2_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_2, GPIO_PIN_SET)
#define d3_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_3, GPIO_PIN_SET)
#define d4_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_4, GPIO_PIN_SET)
#define d5_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_5, GPIO_PIN_SET)
#define d6_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_6, GPIO_PIN_SET)
#define d7_set() HAL_GPIO_WritePin(GPIOD, GPIO_PIN_7, GPIO_PIN_SET)
```



```

#define d0_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_0, GPIO_PIN_RESET)
#define d1_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_1, GPIO_PIN_RESET)
#define d2_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_2, GPIO_PIN_RESET)
#define d3_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_3, GPIO_PIN_RESET)
#define d4_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_4, GPIO_PIN_RESET)
#define d5_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_5, GPIO_PIN_RESET)
#define d6_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_6, GPIO_PIN_RESET)
#define d7_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_7, GPIO_PIN_RESET)

#define RW_set() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_8, GPIO_PIN_SET)
#define RS_set() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_9, GPIO_PIN_SET)
#define EN_set() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_SET)

#define RW_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_8, GPIO_PIN_RESET)
#define RS_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_9, GPIO_PIN_RESET)
#define EN_reset() HAL_GPIO_WritePin(GPIOG, GPIO_PIN_10, GPIO_PIN_RESET)
//-----
voidlcd_init(void);
voidlcd_write_data(uint8_t data);
voidlcd_command(uint8_t data);
#endif /* LCD_H_ */

```

Содержимое файла lcd.c должно быть следующим:

```

#include "lcd.h"
//-----
voidlcd_write_data(uint8_t data)
{
    if(((data>>7)&0x01)==1) {d7_set();} else {d7_reset();}
    if(((data>>6)&0x01)==1) {d6_set();} else {d6_reset();}
    if(((data>>5)&0x01)==1) {d5_set();} else {d5_reset();}
    if(((data>>4)&0x01)==1) {d4_set();} else {d4_reset();}
    if(((data>>3)&0x01)==1) {d3_set();} else {d3_reset();}
    if(((data>>2)&0x01)==1) {d2_set();} else {d2_reset();}
    if(((data>>1)&0x01)==1) {d1_set();} else {d1_reset();}
    if(((data)&0x01)==1) {d0_set();} else {d0_reset();}
}
//-----
voidlcd_command(uint8_t data)
{
    RS_reset();
    lcd_write_data(data);
    EN_set();
    HAL_Delay(1);
    EN_reset();
}
//-----
voidlcd_data(uint8_t data)
{
    RS_set();
    lcd_write_data(data);
}

```

```

EN_set();
HAL_Delay(1);
EN_reset();
}
//-----
void lcd_init(void)
{
    HAL_Delay(41);
    RS_reset();
    lcd_write_data(48);
    EN_set();
    HAL_Delay(1);
    EN_reset();
    HAL_Delay(5);
    lcd_write_data(48);
    EN_set();
    HAL_Delay(1);
    EN_reset();
    HAL_Delay(1);
    EN_reset();
    HAL_Delay(1);
    lcd_command(56);
    HAL_Delay(1);
    lcd_command(0x0F);
    HAL_Delay(1);
    lcd_command(0x01);
    HAL_Delay(2);
    lcd_command(0x06);
    HAL_Delay(1);
    lcd_command(0x02);
    HAL_Delay(1);
}

```



После этого в созданном проекте становятся доступными для использования функции инициализации индикатора, а также функции установки курсора и вывода символа, которых достаточно для выполнения лабораторной работы.

Содержание отчета

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Схема электрическая принципиальная к лабораторной работе.
- 4 Блок-схема алгоритма пользовательской программы.
- 5 Текст пользовательской программы.

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?

2 Лабораторная работа № 7. Изучение организации широтно-импульсной модуляции в микропроцессорных устройствах

2.1 Ход работы

Цель работы: изучить особенности программной и аппаратной реализации широтно-импульсной модуляции с помощью языка программирования С, составить программу, перевести ее в машинные коды, записать в память программ микроконтроллера и выполнить.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- применение широтно-импульсной модуляции;
- организация широтно-импульсной модуляции;
- режимы работы таймеров микроконтроллеров STM32.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде uVision;
- загрузить программу в учебный стенд;
- исследовать программную и аппаратную реализацию ввода информации с помощью клавиатуры и порядок программирования микроконтроллера в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

2.2 Широтно-импульсная модуляция

Широтно-импульсная модуляция (ШИМ) – один из способов снижения воспринимаемой яркости в дисплеях, работающий путём быстрого циклического включения и выключения подсветки. Такая периодическая подача импульсов обычно происходит на постоянной частоте, а отношение длительности части каждого цикла, в течение которой подсветка включена, к общей длительности цикла называется коэффициентом заполнения (величина, обратная скважности). Изменением скважности достигается изменение общей

светоотдачи подсветки. На зрительном уровне этот механизм работает благодаря тому, что чередование включённого и выключенного состояний подсветки происходит достаточно быстро, и пользователь не замечает мерцания, поскольку оно находится за пределами порога слияния мельканий.

Основными причинами применения ШИМ являются лёгкость её реализации, для которой от подсветки нужна лишь способность часто включаться и выключаться, а также обеспечиваемый с её помощью широкий диапазон возможных значений яркости.

Снизить яркость CCFL-подсветки можно путём снижения тока, протекающего через лампу, но лишь примерно вдвое ввиду их строгих требований к току и напряжению. Это делает ШИМ единственным простым способом достижения широкого диапазона регулирования яркости. CCFL-лампа обычно управляется инвертором, включающимся и выключающимся с частотой в десятки килогерц, что находится за пределами мерцания, заметного для человека. Однако ШИМ обычно работает на гораздо более низкой частоте, около 175 Гц, что может приводить к заметным дефектам изображения.

Яркость светодиодной подсветки можно регулировать в широких пределах путём изменения проходящего через них тока, однако в результате несколько изменяется цветовая температура. Данный аналоговый подход к изменению яркости светодиодов также нежелателен ввиду того, что вспомогательные цепи обязаны учитывать тепло, выделяемое светодиодами. Светодиоды во включённом состоянии нагреваются, что уменьшает их сопротивление и дополнительно увеличивает протекающий через них ток. Это может привести к быстрому росту тока в сверхъярких светодиодах и послужить причиной выхода их из строя. При использовании ШИМ ток можно принудительно удерживать на постоянном уровне в течение рабочего цикла, в результате чего цветовая температура всегда одинакова и перегрузок потоку не возникает.

2.3 Схема электрическая принципиальная к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

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. При этом датчик ДД1 подключен к PF.15, ДД2 – к PF.14 и т. д.

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

Семисегментный код подается на индикатор через усилитель на базе транзисторов VT2, VT5, VT9, VT11, VT13, VT16, VT17. Выдача кода осуществляется через порт E (выводы PE.0...PE.7).

Для выбора активного индикатора используется усилитель на базе транзисторов VT3, VT4, VT8, VT10, VT12, VT15, VT18, VT19. Выбор индикатора осуществляется через порт B (выводы PB.4...PB.7 и PB.14).



В данной схеме, чтобы засветить соответствующий сегмент на индикаторе, необходимо вывести логическую 1 в соответствующий разряд порта и, соответственно, 0, чтобы погасить сегмент.

2.4 Пример реализации широтно-импульсной модуляции на микроконтроллере семейства STM32

```

/* Includes -----*/
#include "stm32f4xx_hal.h"

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private variables -----*/
TIM_HandleTypeDef htim3;
TIM_HandleTypeDef htim4;

/* USER CODE BEGIN PV */
/* Private variables -----*/

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM3_Init(void);
static void MX_TIM4_Init(void);

/* USER CODE BEGIN PFP */
/* Private function prototypes -----*/

/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

int main(void)
{
    uint32_t i = 0;
    uint32_t d = 0;
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

```



```

/* Configure the system clock */
SystemClock_Config();

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_TIM3_Init();
MX_TIM4_Init();

/* USER CODE BEGIN 2 */
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_2);
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_2);
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_SET);
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_SET);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
    for (i=65535;i>0;i--)
    {
        TIM3->CCR1 = 0;
        TIM3->CCR2 = 0;
        TIM4->CCR1 = 65535 - i;
        TIM4->CCR2 = i;
        for(d=0;d<200;d++)
        {}
    }
    for (i=65535;i>0;i--)
    {
        TIM3->CCR1 = 0;
        TIM3->CCR2 = 65535 - i;
        TIM4->CCR1 = i;
        TIM4->CCR2 = 0;
        for(d=0;d<200;d++)
        {}
    }
    for (i=65535;i>0;i--)
    {
        TIM3->CCR1 = 65535 - i;

```



```

        TIM3->CCR2 = i;

        TIM4->CCR1 = 0;
        TIM4->CCR2 = 0;
        for(d=0;d<200;d++)
        {}
    }
    for (i=0;i<=65535;i++)
    {
        TIM3->CCR1 = i;
        TIM3->CCR2 = 0;
        TIM4->CCR1 = 0;
        TIM4->CCR2 = 65535 - i;
        for(d=0;d<200;d++)
        {}
    }
}
/* USER CODE END 3 */

```

```

}

```

```

/** System Clock Configuration
*/

```

```

void SystemClock_Config(void)
{

```

```

    RCC_OscInitTypeDef RCC_OscInitStruct;
    RCC_ClkInitTypeDef RCC_ClkInitStruct;

```

```

    __PWR_CLK_ENABLE();

```

```

    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);

```

```

    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 25;
    RCC_OscInitStruct.PLL.PLLN = 336;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV8;
    RCC_OscInitStruct.PLL.PLLQ = 4;
    HAL_RCC_OscConfig(&RCC_OscInitStruct);

```

```

    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1);

```

```

HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);

HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

}

/* TIM3 init function */
void MX_TIM3_Init(void)
{

    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 0;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 65535;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    HAL_TIM_Base_Init(&htim3);

    sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
    HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig);

    HAL_TIM_PWM_Init(&htim3);

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig);

    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1);

    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_2);

}

/* TIM4 init function */
void MX_TIM4_Init(void)
{

    TIM_ClockConfigTypeDef sClockSourceConfig;
    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim4.Instance = TIM4;
    htim4.Init.Prescaler = 0;
    htim4.Init.CounterMode = TIM_COUNTERMODE_UP;

```



```

htim4.Init.Period = 65535;
htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
HAL_TIM_Base_Init(&htim4);

sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
HAL_TIM_ConfigClockSource(&htim4, &sClockSourceConfig);

HAL_TIM_PWM_Init(&htim4);
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig);

sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_1);

HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC, TIM_CHANNEL_2);

}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
*/
void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct;

    /* GPIO Ports Clock Enable */
    __GPIOE_CLK_ENABLE();
    __GPIOH_CLK_ENABLE();
    __GPIOB_CLK_ENABLE();

    /*Configure GPIO pins : PE2 PE3 PE4 PE5
        PE6 PE7 PE0 PE1 */
    GPIO_InitStruct.Pin = GPIO_PIN_2|GPIO_PIN_3|GPIO_PIN_4|GPIO_PIN_5
        |GPIO_PIN_6|GPIO_PIN_7|GPIO_PIN_0|GPIO_PIN_1;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_LOW;
    HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

```



```

/* USER CODE END 4 */

#ifdef USE_FULL_ASSERT

/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t* file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}

#endif

```

Содержание отчета

- 1 Цель работы.
- 2 Постановка задачи.
- 3 Схема электрическая принципиальная к лабораторной работе.
- 4 Блок-схема алгоритма пользовательской программы.
- 5 Текст пользовательской программы.

Контрольные вопросы

- 1 Что такое широтно-импульсная модуляция?
- 2 Для чего используется широтно-импульсная модуляция?
- 3 Как реализуется широтно-импульсная модуляция?
- 4 Какие бывают разновидности широтно-импульсной модуляции?



3 Лабораторная работа № 8. Изучение организации клавиатуры в микропроцессорных устройствах

3.1 Ход работы

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

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- применение матричной клавиатуры;
- организация ввода информации;
- вывод информации на жидкокристаллические знакосинтезирующие индикаторы.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде uVision;
- загрузить программу в учебный стенд;
- исследовать программную и аппаратную реализацию ввода информации с помощью клавиатуры и порядок программирования микроконтроллера в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

3.2 Основные теоретические сведения о клавиатурах

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. 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.



При использовании матричной клавиатуры следует учитывать, что 16-клавишную клавиатуру можно подключить, используя только восемь линий параллельного порта.

Работает матричная клавиатура следующим образом. На одну из линий подаются сигналы, т. е. происходит сканирование одного из рядов клавиатуры. Если ни одна клавиша не нажата, то сигнал на выходных линиях отсутствует. Если клавиша в сканирующем ряду нажата, то на соответствующей выходной линии появляется поданный на линию R_i сигнал. Таким образом, зная, какой ряд в данный момент сканируется и на какой из выходных линий появится сигнал, можно определить, какая клавиша нажата в данный момент.

3.3 Электрическая принципиальная схема к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

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

В качестве матричного жидкокристаллического индикатора А1 используется индикатор ВС1604А-GPLCW. Индикатор подключен к портам микроконтроллера: 8-битная шина данных к выводам порта G (PG.0...PG.7), сигналы управления – к выводам PG.8...PG.10 (R/W, RS и En соответственно).

3.4 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).

Задания

1 Ввести последовательно два трехзначных числа, найти их разность и вывести результат на матричный жидкокристаллический индикатор (МЖКИ).

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

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

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

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

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



Пример проекта с использованием матричной клавиатуры можно получить у преподавателя.

Контрольные вопросы

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

4 Лабораторная работа № 9. Изучение организации устройства ввода аналоговой информации в микропроцессорных устройствах

4.1 Ход работы

Цель работы: изучить особенности организации устройств ввода аналоговой информации в микропроцессорных устройствах с помощью языка программирования С, составить программу, перевести ее в машинные коды, записать в память программ микроконтроллера и выполнить.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- применение аналого-цифровых преобразователей;
- организация аналого-цифрового преобразования;
- вывод информации на жидкокристаллические знакосинтезирующие индикаторы.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде uVision;
- загрузить программу в учебный стенд;
- исследовать программную и аппаратную реализацию ввода аналоговой информации и порядок программирования микроконтроллера в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

4.2 Аналого-цифровое преобразование

Аналого-цифровое преобразование (АЦП) в STM32 обладает следующими особенностями:

- разрядность – 12 бит;
- опорное напряжение – от 2,4 до 3,6 В;
- скорость оцифровки – до 1 MSPS;



- 18 каналов (16 внешних и два внутренних – опорное напряжение и температурный датчик);
- прерывание по окончании регулярных и инжектированных преобразований;
- прерывание от оконного компаратора (Analogwatchdog);
- отправка данных по DMA для регулярных преобразований;
- одиночное и непрерывное преобразование;
- режим сканирования каналов по заданному списку;
- самокалибровка;
- выравнивание результата по выбранному краю;
- время преобразования – 12,5 цикла, время захвата – программируемое.

Режим регулярных преобразований. В этом режиме задаётся группа от 1 до 16 каналов, обработка которых производится последовательно, а результат помещается в единственный 12-разрядный регистр, подразумевается, что данные будут сохраняться с помощью DMA. Следует отметить, что нельзя просто выбрать один канал, надо задать группу, а в ней выбрать количество каналов. Преобразования могут быть циклическими и периодическими, запускаться по таймеру или внешнему прерыванию.

Режим инжектированных преобразований. В последовательность инжектированных преобразований может входить до четырех каналов, они имеют собственные регистры для сохранения результата. Сразу после запуска инжектированных преобразований приостанавливается оцифровка регулярных, затем по окончании инжектированных она возобновляется. В регистры компенсации можно записать число, которое будет автоматически вычитаться из результата преобразования; если в результате вычитания получается отрицательное число, регистр инжектированного преобразования дополняется знаком.

4.3 Электрическая принципиальная схема к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

В качестве датчиков аналоговых сигналов выступают переменные резисторы R23, R26, R29 сопротивлением 10 кОм. Выходы датчиков АД1, АД2, АД3 подключены к выводам PF.5, PF.4, PF.3 соответственно, альтернативной функцией которых являются входы 15, 14 и 9 АЦП3 микроконтроллера. Величина напряжения V_a , питающего АЦП, составляет 3,3 В. В качестве источника опорного напряжения для АЦП используется микросхема стабилизатора напряжения DA5 AD1582CRT, на выходе которой формируется напряжение 2,5 В.

4.4 Настройка проекта в среде CubeMX

Сначала требуется найти АЦП в списке периферийных устройств микроконтроллера, как показано на рисунке 4.1.



Затем нужно произвести настройку параметров АЦП, как показано на рисунке 4.2. Также нужно разрешить прерывания АЦП, как показано на рисунке 4.3.

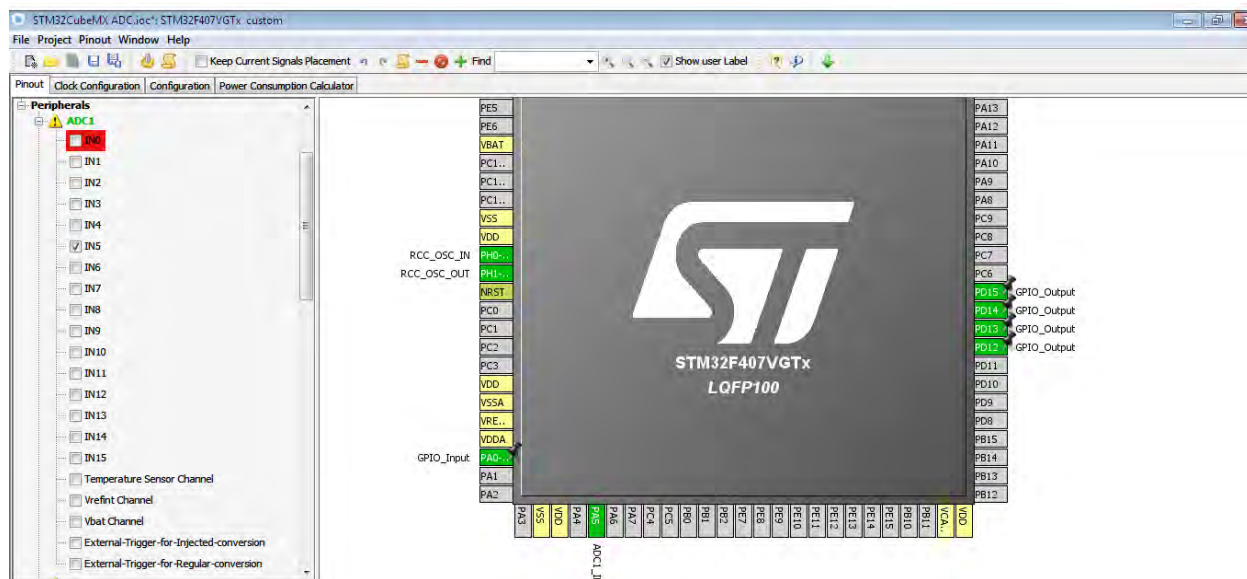


Рисунок 4.1 – Выбор АЦП микроконтроллера

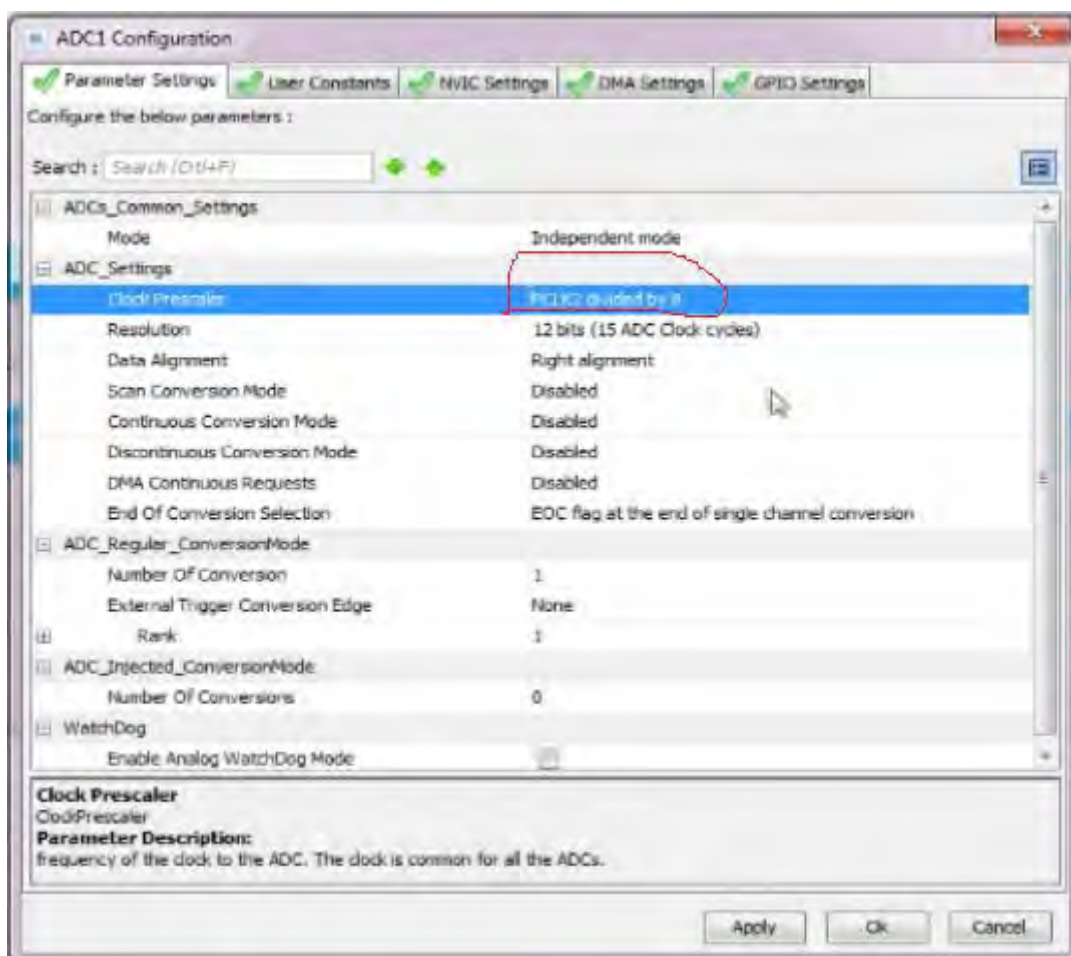


Рисунок 4.2 – Настройка АЦП микроконтроллера

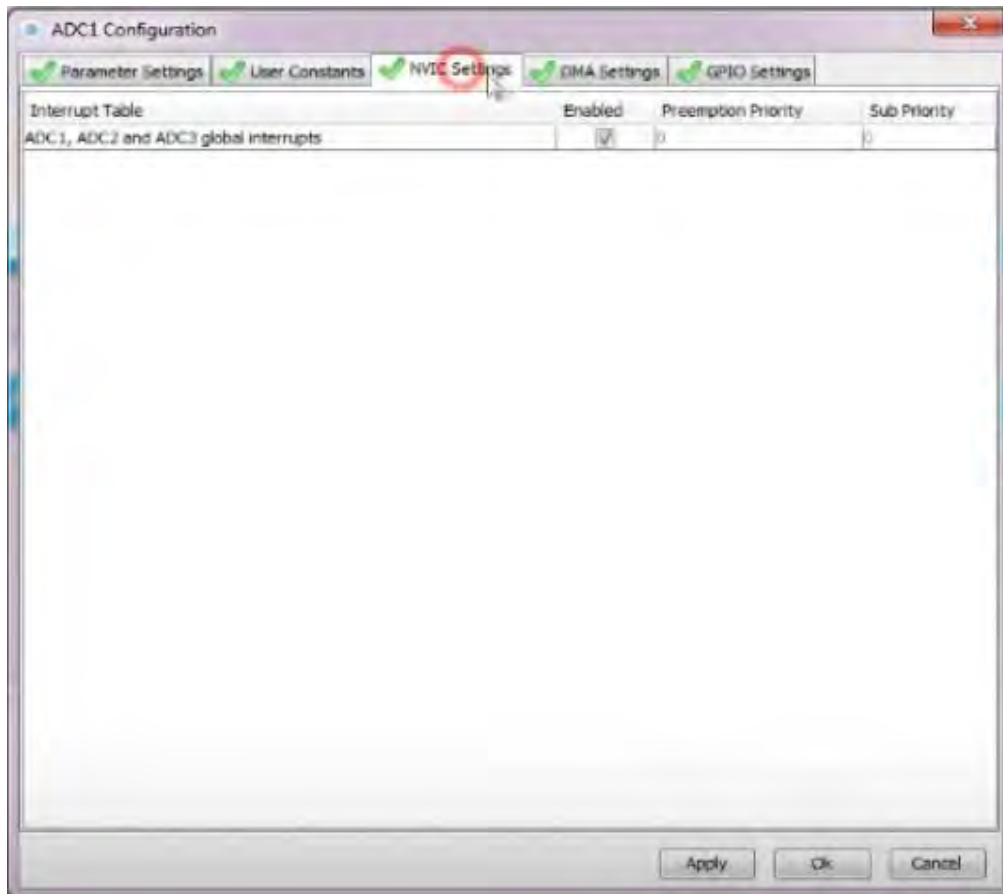


Рисунок 4.3 – Разрешение прерываний АЦП микроконтроллера

4.5 Пример программы с использованием АЦП

Пример проекта для микроконтроллера STM32F407ZET6 с использованием АЦП можно получить у преподавателя.

В указанном примере присутствует main.c, где вызывается функция инициализации АЦП, а также осуществляется преобразование в бесконечном цикле. По результату преобразования производится индикация с помощью дискретных выходов микроконтроллера.

Контрольные вопросы

- 1 Для чего используются АЦП?
- 2 По какому принципу чаще всего работают АЦП?
- 3 Как реализуется опрос АЦП?

5 Лабораторная работа № 10. Изучение организации устройства вывода аналоговой информации в микропроцессорных устройствах

5.1 Ход работы

Цель работы: ознакомиться с техническими характеристиками и принципом действия цифроаналогового преобразователя (ЦАП), изучить схему вывода аналоговых сигналов в микропроцессорной системе, разработать и отладить программу для вывода аналоговой информации.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- основные типы и принцип действия ЦАП;
- основные технические характеристики ЦАП.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде μ Vision;
- загрузить программу в учебный стенд;
- исследовать работу цифроаналогового преобразователя в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

5.2 Цифроаналоговое преобразование

Цифроаналоговый преобразователь (ЦАП) – это устройство для преобразования цифрового кода в аналоговый сигнал по величине, пропорциональной значению кода.

ЦАП применяются для связи цифровых управляющих систем с устройствами, которые управляются уровнем аналогового сигнала. Также ЦАП является составной частью во многих структурах аналого-цифровых устройств и преобразователей.

ЦАП характеризуется функцией преобразования. Она связывает изменение цифрового кода с изменением напряжения или тока.

Принцип работы большинства ЦАП – это суммирование долей аналоговых сигналов (веса разряда) в зависимости от входного кода.

ЦАП можно реализовать с помощью суммирования токов, суммирования напряжений и деления напряжений. В первом и втором случае в соответствии со значениями разрядов входного кода суммируются сигналы генераторов токов и источников ЭДС. Последний способ представляет собой управляемый кодом делитель напряжения. Два последних способа не нашли широкого распространения в связи с практическими трудностями их реализации.

Для организации выдачи аналоговых сигналов в цифровых системах часто используются встроенные цифроаналоговые преобразователи. В состав



микроконтроллера STM32F407 входит модуль ЦАП, представляющий собой 12-битный цифроаналоговый преобразователь с напряжением на выходе. ЦАП могут быть сконфигурированы в 8- или 12-битном режиме и могут использоваться вместе с контроллером ПДП. В 12-битном режиме данные могут быть выравнены по левому или по правому краю регистра. Модуль ЦАП имеет два выходных канала, каждый из которых управляется своим преобразователем. В «сдвоенном» режиме преобразования могут быть выполнены независимо или одновременно, когда оба канала сгруппированы вместе для синхронного обновления выхода. В качестве опорного напряжения может использоваться сигнал на входе V_{REF+} (совместно с АЦП).

Основные особенности модуля ЦАП:

- два преобразователя: у каждого один выходной канал;
- правое или левое выравнивание данных в 12-битном режиме;
- возможность синхронного обновления;
- генератор шума;
- генератор треугольного сигнала;
- режим «сдвоенных» каналов для независимых или одновременных преобразований;
- возможность ПДП для каждого канала;
- обнаружение ошибок опустошения ПДП;
- возможность запуска преобразования от внешнего сигнала;
- вход опорного напряжения V_{REF+} ;
- время установления сигнала на выходе не превышает 6 мкс;
- максимальная скорость преобразований может достигать 1 миллиона преобразований в секунду.

Микросхема светодиодного индикатора уровня LM3914 представляет собой схему, чувствительную ко входному уровню напряжения и управляющую 10 светодиодами, т. е. являющуюся линейным аналоговым индикатором. Состояние индикатора изменяется либо путем перемещения точки, либо путем изменения величины столбца светодиодов.

Микросхема содержит внутренний источник опорного напряжения и 10-уровневый делитель напряжения. Входной сигнал подается на входной буфер и на 10 внутренних компараторов, вырабатывающих сигнал, управляющий светодиодами.

Условное обозначение микросхемы приведено на рисунке 5.1.

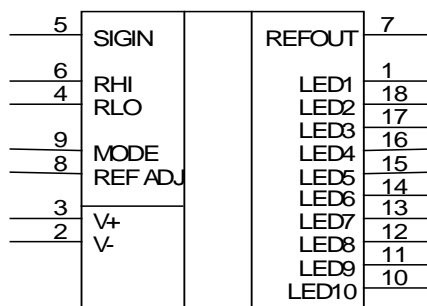


Рисунок 5.1 – Условное обозначение микросхемы LM3914

Микросхема LM3914 размещена в 18-контактном пластиковом корпусе типа DIP. Назначение выводов микросхемы LM3914 приведены в таблице 5.1.

Таблица 5.1 – Назначение выводов микросхемы LM3914

Обозначение выводов	Номер	Назначение
V-	2	Отрицательное питание
V+	3	Положительное питание
SigIn	5	Аналоговый вход
Rhi	6	Верхний конец делителя
Rlo	4	Нижний конец делителя
Mode	9	Режим (точка / столбец)
RefAdj	8	Регулировка опорного напряжения
RefOut	7	Выход опорного напряжения
Led1...Led10	1, 18...10	Выходы для подключения светодиодов

Основные параметры микросхемы приведены в таблице 5.2.

Таблица 5.2 – Параметры микросхемы LM3914

Параметры	Единица измерения	Значение
Входной ток аналогового входа напряжения	нА	25...100
Максимальное перенапряжение на аналоговом входе	В	-35...+35
Сопротивление делителя	кОм	(12±0,5) %
Напряжение внутреннего источника опорного напряжения	В	1,28
Ток светодиодов ($V_{LCD} = 5 \text{ В}$)	мА	7...13

Вывод MODE (режим) предназначен для переключения режима индикации. Если этот вывод подключен к выводу V+, то используется режим индикации столбцом светодиодов. Если вывод не подключен, то используется индикация светодиодной точкой.

5.3 Электрическая принципиальная схема к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

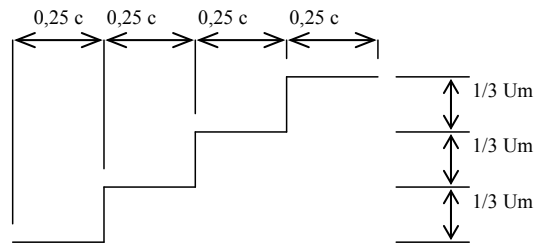
В схеме дискретные датчики ДД1...ДД8 оформлены в виде набора переключателей SA1...SA8, подключенных к старшим восьми разрядам порта F (PF.8...PF.15) микроконтроллера. При этом датчик ДД1 подключен к PF.15, ДД2 – к PF.14 и т. д.

Выход ЦАП (вывод PA.4) подключен ко входу микросхемы светодиодного индикатора уровня LM3914 DA1. Выбор режима индикации микросхемы осуществляется инвертором, собранным на транзисторе VT22, управлять которым можно, используя вывод порта PB.15. К выходу LM3914 подключена линейка светодиодов A7, LTA-1000HR.

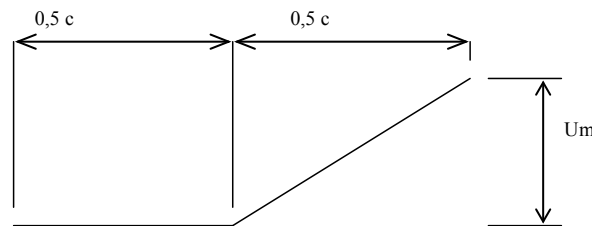
5.4 Варианты индивидуальных заданий к лабораторной работе

Разработать программу для учебного стенда, позволяющую вводить данные с дискретных датчиков (ДД1...ДД8), выполнять определенные действия над данными и выводить аналоговый сигнал с помощью цифроаналогового преобразователя (ЦАП).

1 Если ДД1 = 1 и ДД2 = 0, то вывести на ЦАП периодически повторяющийся сигнал. Если ДД2 = 1, то сигнал инвертировать.

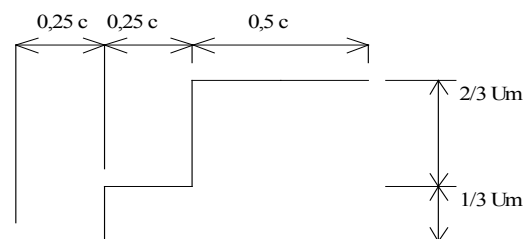


2 Если ДД4 = 1, то вывести на ЦАП периодически повторяющийся сигнал. U_m – максимальное напряжение на выходе ЦАП.

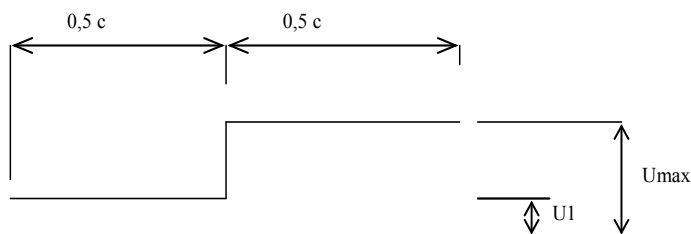


3 Рассматривая состояние дискретных датчиков ДД1...ДД4 и ДД5...ДД8 как два четырехразрядных двоичных числа, найти их сумму и результат вывести на ЦАП.

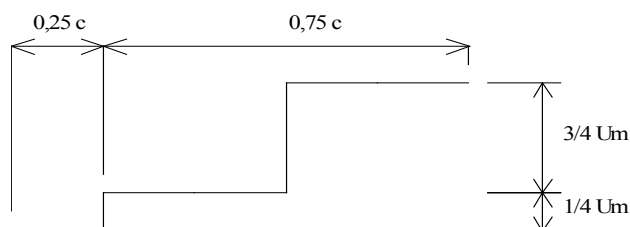
4 Если ДД1 = 1 и ДД6 = 0, то вывести на ЦАП периодически повторяющийся сигнал. Если ДД6 = 1, то сигнал инвертировать. U_m – максимальное напряжение на выходе ЦАП.



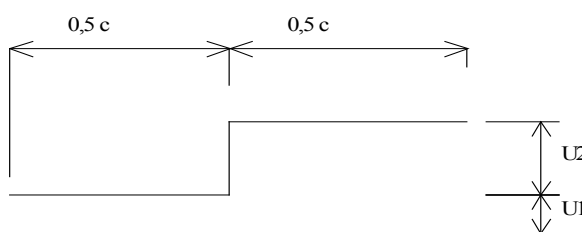
Вывести на ЦАП сигнал, где U_1 прямо пропорционален состоянию датчиков ДД1...ДД8. Причем при состоянии датчиков «00000000» $U = 0$ В, при состоянии датчиков «11111111» $U = U_{\max}$.



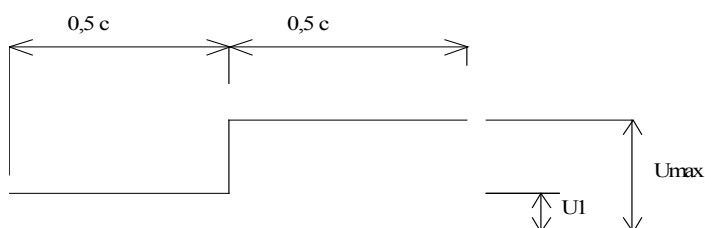
5 Если ДД7 = 1 и ДД8 = 0, то вывести на ЦАП периодически повторяющийся сигнал. Если ДД8 = 1, то сигнал инвертировать. U_m – максимальное напряжение на выходе ЦАП.



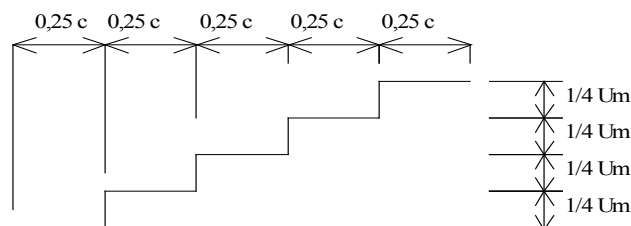
6 Вывести на ЦАП сигнал, где U_1 прямо пропорционален состоянию датчиков ДД1...ДД4, U_2 – состоянию датчиков ДД5... ДД8. Причем при состоянии датчиков «1111» $U = 0$ В, при состоянии датчиков «0000» $U = U_{\max}$.



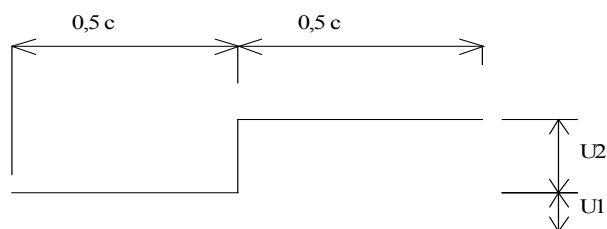
7 Вывести на ЦАП сигнал, где U_1 прямо пропорционален состоянию датчиков ДД1...ДД8. Причем при состоянии датчиков «11111111» $U = 0$ В, при состоянии датчиков «00000000» $U = U_{\max}$.



8 Если $ДД3 = 1$ и $ДД4 = 0$, то вывести на ЦАП периодически повторяющийся сигнал. Если $ДД4 = 1$, то сигнал инвертировать. U_m – максимальное напряжение на выходе ЦАП.



9 Вывести на ЦАП сигнал, где U_1 прямо пропорционален состоянию датчиков $ДД1...ДД4$, U_2 – состоянию датчиков $ДД5...ДД8$. Причем при состоянии датчиков «0000» $U = 0$ В, при состоянии датчиков «1111» $U = U_{max}$.



10 Рассматривая состояние дискретных датчиков $ДД1...ДД4$ и $ДД5...ДД8$ как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ и вывести на ЦАП.

11 Рассматривая состояние дискретных датчиков $ДД1...ДД4$ и $ДД5...ДД8$ как два четырехразрядных двоичных числа, найти результат поразрядной логической операции И и вывести на ЦАП.

12 Рассматривая состояние дискретных датчиков $ДД1...ДД4$ и $ДД5...ДД8$ как два четырехразрядных двоичных числа, найти их произведение и результат вывести на ЦАП.

13 Рассматривая состояние дискретных датчиков $ДД1...ДД4$ и $ДД5...ДД8$ как два четырехразрядных двоичных числа, найти результат поразрядной логической операции ИЛИ и вывести на ЦАП.

14 Рассматривая состояние дискретных датчиков $ДД1...ДД4$ и $ДД5...ДД8$ как два четырехразрядных двоичных числа, найти их разность и результат вывести на ЦАП.

Пример программы с использованием ЦАП можно взять у преподавателя.

Контрольные вопросы

- 1 Какую функцию выполняют цифроаналоговые преобразователи?
- 2 Какие методы построения ЦАП применяют?



- 3 Как классифицируются ЦАП?
- 4 При помощи каких структур реализуются ЦАП?
- 5 Назовите основные особенности ЦАП микроконтроллера STM32F407.
- 6 Из каких структурных элементов состоит ЦАП микроконтроллера STM32F407?
- 7 Как осуществляется управление ЦАП микроконтроллера STM32F407?

6 Лабораторная работа № 11. Изучение организации последовательных портов в микропроцессорных устройствах

6.1 Ход работы

Цель работы: изучить структуру и особенности работы последовательного интерфейса в различных режимах, программу передачи и приема информации по последовательному каналу на языке программирования микроконтроллера; составить программу передачи или приема информации по последовательному каналу в заданном режиме, используя язык программирования микроконтроллера, перевести ее в машинные коды, записать в память программ микроконтроллера, соединив приемник и передатчик, проверить работу интерфейса.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- основные виды последовательных интерфейсов;
- основные характеристики последовательных интерфейсов;

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде μ Vision;
- загрузить программу в учебный стенд;
- исследовать работу цифроаналогового преобразователя в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

6.2 Интерфейс RS-232-C

Интерфейс RS-232-C разработан ассоциацией электронной промышленности (Electronic Industries Association – EIA) как стандарт для соединения компьютеров и различных последовательных периферийных устройств. Широко известен как последовательный порт персональных компьютеров. Исторически имел широкое распространение в телекоммуникационном оборудовании. В настоящее время всё ещё используется для подключения всевозможного специального или



устаревшего оборудования к компьютерам, однако в основном уже вытеснен интерфейсом USB.

Последовательная передача данных означает, что данные передаются по единственной линии. При этом биты байта данных передаются по очереди с использованием одного провода. Для синхронизации группе битов данных обычно предшествует специальный стартовый бит, после группы битов следуют бит проверки на четность и один или два стоповых бита. Иногда бит проверки на четность может отсутствовать (рисунок 6.1).



Рисунок 6.1 – Формат передачи данных интерфейса RS-232

Из рисунка видно, что исходное состояние линии последовательной передачи данных – уровень логической 1. Это состояние линии называют отмеченным – MARK. Когда начинается передача данных, уровень линии переходит в 0. Это состояние линии называют пустым – SPACE. Если линия находится в таком состоянии больше определенного времени, считается, что линия перешла в состояние разрыва связи – BREAK.

Стартовый бит START сигнализирует о начале передачи данных. Далее передаются биты данных, вначале младшие, затем старшие.

Если используется бит четности P, то передается и он. Бит четности имеет такое значение, чтобы в пакете битов общее количество единиц (или нулей) было четно или нечетно, в зависимости от установки регистров порта. Этот бит служит для обнаружения ошибок, которые могут возникнуть при передаче данных из-за помех на линии. Приемное устройство заново вычисляет четность данных и сравнивает результат с принятым битом четности. Если четность не совпала, то считается, что данные переданы с ошибкой.

В самом конце передаются один или два стоповых бита STOP, завершающих передачу байта. Затем до прихода следующего стартового бита линия снова переходит в состояние MARK.

Использование бита четности, стартовых и стоповых битов определяет формат передачи данных. Очевидно, что передатчик и приемник должны использовать один и тот же формат данных, иначе обмен будет невозможен.

Другая важная характеристика – скорость передачи данных. Она также должна быть одинаковой для передатчика и приемника.

Скорость передачи данных обычно измеряется в бодах (по фамилии французского изобретателя телеграфного аппарата Э. Бодо (Emile Baudot)). Боды определяют количество передаваемых битов в секунду. При этом учитываются и старт/стопные биты, а также бит четности.

Иногда используется другой термин – биты в секунду (bps). Здесь имеется в виду эффективная скорость передачи данных, без учета служебных битов.

Очень важно, чтобы тактовые частоты приемника и передатчика были одинаковыми, допустимое расхождение – не более 10 %. Скорость передачи по RS-232C может выбираться из ряда 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с.

Интерфейс RS-232-C определяет обмен между устройствами двух типов: DTE (Data Terminal Equipment – терминальное устройство) и DCE (Data Communication Equipment – устройство связи). В большинстве случаев, но не всегда, компьютер является терминальным устройством. Модемы, принтеры, графопостроители всегда являются устройствами связи.

В компьютерах типа IBM PC внешний последовательный порт оформлен в виде разъемов типа DB25 или DB9 (таблица 6.1).

Таблица 6.1 – Разводка разъема последовательной передачи данных DB25

Номер контакта DB25	Номер контакта DB9	Назначение контакта	Вход или выход компьютера
1	–	Защитное заземление (Frame Ground, FG)	–
2	3	Передаваемые данные (Transmitted Data, TD)	Выход
3	2	Принимаемые данные (Received Data, RD)	Вход
4	7	Запрос для передачи (Request to send, RTS)	Выход
5	8	Сброс для передачи (Clear to Send, CTS)	Вход
6	6	Готовность данных (Data Set Ready, DSR)	Вход
7	5	Сигнальное заземление (Signal Ground, SG)	–
8	1	Детектор принимаемого с линии сигнала (Data Carrier Detect, DCD)	Вход
9...19		Не используются	
20	4	Готовность выходных данных (Data Terminal Ready, DTR)	Выход
21		Не используется	
22	9	Индикаторвызова (Ring Indicator, RI)	Вход
23...25		Не используются	

Входы TD и RD используются устройствами DTE и DCE по-разному. Устройство DTE использует вход TD для передачи данных, а вход RD – для приема данных. И наоборот, устройство DCE использует вход TD для приема, а вход RD – для передачи данных.

Поэтому для соединения терминального устройства и устройства связи выводы их разъемов необходимо соединить напрямую (рисунки 6.2).

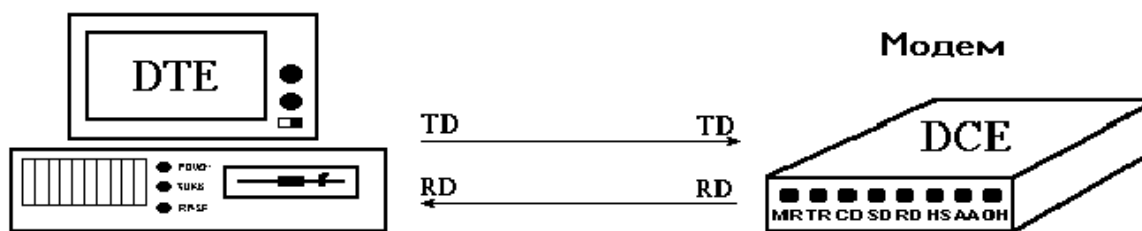


Рисунок 6.2 – Соединение устройств DTE и DCE

Рассмотрим процесс подтверждения связи между компьютером и модемом. В начале сеанса связи компьютер должен удостовериться, что модем может произвести вызов (находится в рабочем состоянии). Затем, после вызова абонента, модем должен сообщить компьютеру, что он произвел соединение с удаленной системой. Подробнее это происходит следующим образом.

Компьютер подает сигнал по линии DTR, чтобы показать модему, что он готов к проведению сеанса связи. В ответ модем подает сигнал по линии DSR. Когда модем произвел соединение с другим, удаленным модемом, он подает сигнал по линии DCD, чтобы сообщить об этом компьютеру.

Падение напряжения на линии DTR сообщает модему, что компьютер не может далее продолжать сеанс связи, например, из-за того, что выключено питание компьютера. В этом случае модем прервет связь. Падение напряжения на линии DCD сообщает компьютеру, что модем потерял связь и не может больше продолжать соединение. В обоих случаях эти сигналы дают ответ на наличие связи между модемом и компьютером.

Рассмотренный уровень управлением связи – подтверждение связи – является самым низким. Существует более высокий уровень, который используется для управления скоростью обмена данными, но он также реализуется аппаратно. Практически управление скоростью обмена данными (управление потоком) необходимо, если производится передача больших объемов данных с высокой скоростью. Когда одна система пытается передать данные с большей скоростью, чем они могут быть обработаны принимающей системой, результатом может стать потеря части передаваемых данных. Чтобы предотвратить передачу большего числа данных, чем то, которое может быть обработано, используют управление связью, называемое «управление потоком» (flow-controll handshake).

Стандарт RS-232-C определяет возможность управления потоком только для полудуплексного соединения. Полудуплексным называется соединение, при котором в каждый момент времени данные могут передаваться только в одну сторону.

Однако фактически этот механизм используется и для дуплексных соединений, когда данные передаются по линии связи одновременно в двух направлениях.

В полудуплексных соединениях устройство DTE подает сигнал RTS, когда оно желает передать данные. DCE отвечает сигналом по линии CTS, когда оно

готово, и DTE начинает передачу данных. До тех пор, пока оба сигнала RTS и CTS не примут активное состояние, только DCE может передавать данные.

При дуплексных соединениях сигналы RTS/CTS имеют противоположные значения по сравнению с теми, которые они имели для полудуплексных соединений.

Когда DTE может принять данные, он подает сигнал по линии RTS. Если при этом DCE готово для принятия данных, оно возвращает сигнал CTS. Падение напряжения на линиях RTS или CTS сообщает передающей системе, что получающая система не готова для приема данных.

При передаче данных на большие расстояния без использования специальной аппаратуры из-за помех, наводимых электромагнитными полями, возможно возникновение ошибок. Вследствие этого накладываются ограничения на длину соединительного кабеля между устройствами DTR-DTR и DTR-DCE.

Официальное ограничение по длине для соединительного кабеля по стандарту RS-232-C составляет 15,24 м.

Уровни напряжения на линиях разъема составляют для логического нуля $-15 \dots -3$ В, для логической единицы $+3 \dots +15$ В. Промежуток от -3 до $+3$ В соответствует неопределенному значению (рисунок 6.3).



Рисунок 6.3 – Уровни сигналов

Если подключить внешние устройства к разъему интерфейса RS-232-C (а также при соединении двух компьютеров нуль-модемом), предварительно следует выключить его и компьютер, а также снять статический заряд (подсоединив заземление). В противном случае можно вывести из строя асинхронный адаптер. Земля компьютера и земля внешнего устройства должны быть соединены вместе.

6.3 Последовательный порт микроконтроллеров STM32F407

Наряду с портами ввода-вывода GPIO, универсальный синхронно-асинхронный последовательный приёмопередатчик USART (Universal Synchronous Asynchronous Receiver Transmitter) является одним из самых востребованных портов для связи любого микроконтроллера с внешними

устройствами. С помощью порта USART можно легко организовать связь микроконтроллера с компьютером, провести его программирование, а также связать микроконтроллеры между собой по интерфейсам RS-232, RS-485, S-422 и т. д.

Преимуществами порта USART являются простота в использовании, а также, в отличие от интерфейса USB, гибкость в настройке и надёжность работы.

Микроконтроллеры STM32 имеют обычно несколько портов USART. Например, STM32F407 имеет четыре синхронно-асинхронных последовательных приёмопередатчика USART (USART1, USART2, USART3 и USART6) и два асинхронных последовательных приёмопередатчика UART (UART4 и UART5).

Кроме обычного асинхронного режима работы с использованием сигналов RXD и TXD, порты USART STM32 поддерживают несколько расширенных режимов работы. В отличие от стандартных портов USART они могут:

- работать с однопроводной полудуплексной линией связи;
- поддерживать интерфейсы Smart-карт стандарта ISO7618-3, LIN (local interconnection network) и IrDA (infrared data association);
- связываться с внешними устройствами, оснащёнными SPI-совместимым интерфейсом, по трехпроводной линии.

К особенностям порта USART микроконтроллеров STM32 также относится возможность дробного деления тактовой частоты для формирования заданной скорости работы. Благодаря этому можно получить стандартные скорости связи порта при любой частоте тактового сигнала.

Кроме того, с помощью блока DMA (Direct Memory Access) для любого порта USART может быть организован прямой доступ к памяти как для приёма, так и для передачи данных.

Порты USART STM32 способны поддерживать скорость обмена до 4,5 Мбит/с. Формат слова USART может составлять 8 или 9 бит данных и 0,5; 1; 1,5 или 2 стоповых бит. Дробные значения стоповых бит применяются в режиме порта Smartcard. Первым передаётся и принимается младший бит данных.

Приёмопередатчики оснащены выходом синхронизации передатчика для обеспечения синхронной передачи.

Некоторые сигналы USART STM32 можно программно переназначать на другие выводы микроконтроллера.

При работе приёмопередатчика возможно формирование десяти флагов прерывания:

- по изменению сигнала CTS;
- при разрыве связи LIN;
- при очистке сдвигового регистра передачи данных;
- при завершении передачи очередного байта данных;
- при приеме данных в регистр приемника;
- при обнаружении кадра ожидания;
- при обнаружении ошибки переполнения приёмного буфера;
- при обнаружении ошибки кадра при приёме байта;



- при наличии шума в кадре;
- при обнаружении ошибки паритета при приёме байта.

6.4 Микросхема MAX202

MAX202 представляет собой микросхему приемопередатчика для согласования уровней сигналов TTL/CMOS с уровнями сигналов стандарта EIA/TIA-232E (RS-232-C). Микросхема предназначена для построения интерфейсов RS-232 и V.28 в схемах, в которых отсутствует источник питания ± 12 В. Микросхема имеет встроенный преобразователь напряжения $+5$ В/ ± 12 В.

6.5 Электрическая принципиальная схема к лабораторной работе

Электрическая принципиальная схема к лабораторной работе находится у преподавателя.

Связь платы стенда с персональным компьютером осуществляется по интерфейсу RS-232 (разъем XP2). В качестве устройства согласования уровней сигналов с интерфейсом RS-232 используется микросхема DA1 MAX202.

В качестве семисегментного символьного индикатора ССИ А4 используется индикатор LTC-4627JS. Индикатор представляет собой модуль из четырех семисегментных светодиодных индикаторов с общим анодом с дополнительным служебным разрядом.

Семисегментный код подается на индикатор через усилитель на базе транзисторов VT2, VT5, VT7, VT9, VT11, VT13, VT16, VT17. Выдача кода осуществляется через порт E выводы PE.0...PE.7.

Для выбора активного индикатора используется усилитель на базе транзисторов VT3, VT4, VT6, VT8, VT10, VT12, VT14, VT15, VT18, VT19. Выбор индикатора осуществляется через порт B выводы PB.4...PB.7 и PB.14. В данной схеме, чтобы засветить соответствующий сегмент на индикаторе, необходимо вывести логический 1 в соответствующий разряд порта и, соответственно, 0, чтобы погасить сегмент.

6.6 Варианты индивидуальных заданий к лабораторной работе

Разработать программу для учебного стенда, позволяющую вводить данные с персонального компьютера через последовательный интерфейс RS-232, выполнять определенные действия над данными и выводить поясняющие надписи и результат вычислений на персональный компьютер через последовательный интерфейс RS-232 и на матричный жидкокристаллический индикатор или на символьный светодиодный индикатор (ССИ). Для работы на персональном компьютере использовать программу «Загрузчик СУ-МК» в режиме терминала.



Задания

1 Ввести три трехзначных числа, найти максимальное из них и вывести его на МЖКИ.

2 Ввести три трехзначных числа, найти максимальное из них и вывести его на ССИ.

3 Ввести последовательно два трехзначных числа, найти их произведение и результат вывести на ССИ.

4 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции ИЛИ и вывести на ССИ в десятичной форме.

5 Ввести последовательно два трехзначных числа, найти их сумму и результат вывести на ССИ.

6 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ и вывести на ССИ в десятичной форме.

7 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции И и вывести на МЖКИ в десятичной форме.

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

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

10 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции ИЛИ и вывести на МЖКИ в десятичной форме.

11 Ввести три трехзначных числа, найти минимальное из них и вывести его на ССИ.

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

13 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ и вывести на МЖКИ в десятичной форме.

14 Ввести последовательно два трехзначных числа, найти их разность и результат вывести на ССИ.

15 Ввести последовательно два трехзначных числа, найти результат поразрядной логической операции И и вывести на ССИ в десятичной форме.

Пример проекта для микроконтроллера STM32F407ZET6 с использованием UART можно взять у преподавателя.

Контрольные вопросы

- 1 Для чего предназначен интерфейс RS-232?
- 2 Что такое последовательная передача данных?
- 3 Формат передачи данных интерфейса RS-232.
- 4 Какие соглашения по скоростям передачи данных приняты в RS-232?
- 5 Что такое DTE и DTR устройства?



- 6 Опишите низкоуровневый интерфейс подтверждения связи.
- 7 Опишите процесс управления потоком.
- 8 Каковы технические параметры интерфейса RS-232?
- 9 Охарактеризуйте режимы работы последовательного порта в микроконтроллере STM32F407.
- 10 Как изменить скорость передачи данных через последовательный порт?

7 Лабораторная работа № 12. Разработка и отладка программ управления технологическими объектами в реальном времени

7.1 Ход работы

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.

Перед началом лабораторных занятий студентам необходимо изучить следующие вопросы:

- разностные уравнения;
- основные технические характеристики ЦАП.

В ходе выполнения работы нужно:

- изучить электрическую принципиальную схему к лабораторной работе;
- разработать программу в соответствии с индивидуальным заданием;
- отладить программу в среде μ Vision;
- загрузить программу в учебный стенд;
- исследовать работу цифроаналогового преобразователя в соответствии с индивидуальным заданием;
- оформить отчет по лабораторной работе.

После выполнения работы следует ответить на контрольные вопросы.

7.2 Цифровые системы управления

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 σ - Δ 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 или самого объекта.

Если ЦВМ универсальная (ЭВМ), то возможно построение многофункциональных САУ, когда одна ЦВМ обслуживает комплекс составляющих объект устройств. Например, в автомобиле это система навигации, система бортового электропитания, АБС, электронная подвеска, управление топливopодачей. В подобных случаях в состав системы ЦУ должны входить аналоговые или цифровые мультиплексоры и демультимплексоры.

Во всех случаях ЦВМ предоставляет легкодоступные информационные потоки, позволяющие, кроме прямого управления, осуществлять функции контроля, оптимизации, координации и организации всех процессов.

7.3 Цифровые регуляторы

В непрерывных системах широко используются ПИД-регуляторы (пропорционально-интегрально-дифференциальные регуляторы), которые представляются идеализированным уравнением

$$u(t) = K_p \cdot \left[x(t) + \frac{1}{T_I^x} \int_0^t x(\tau) d\tau + T_D^x \frac{dx(t)}{dt} \right],$$

где $x(t)$ – сигнал разогласования между управляющим воздействием и реакцией объекта управления;

K_p – коэффициент усиления пропорционального канала;

T_I^x – постоянная времени сопрягающего полюса интегрального канала;

T_D^x – постоянная времени сопрягающего полюса дифференциального канала.

Для малых периодов дискретизации $T_{Ц}$ уравнение может быть преобразовано в разностное без существенной потери в точности. Непрерывное интегрирование может быть представлено с помощью метода прямоугольников или метода трапеций.

Используем метод прямоугольников для аппроксимации непрерывного интеграла и запишем ПИД-закон в дискретном виде:



$$u[n] = K_p \cdot \left[x[n] + \frac{T_{\Pi}}{T_I^x} \sum_{i=0}^{n-1} x[i] + \frac{T_D^x}{T_{\Pi}} (x[n] - x[n-1]) \right],$$

где n – номер дискретного отсчета.

В результате получен нерекуррентный (позиционный) алгоритм управления, который требует сохранения всех предыдущих значений сигнала ошибки $x[i]$ и в котором каждый раз заново вычисляется управляющий сигнал $u[n]$.

Для реализации программ закона регулирования на ЦВМ более удобным является рекуррентный алгоритм. Он характеризуется тем, что для вычисления текущего значения сигнала $u[n]$ используется его предыдущее значение $u[n-1]$ и поправочный коэффициент, не требующий существенных вычислительных затрат.

$$u[n] = u[n-1] + b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2], \quad (7.1)$$

где

$$b_0 = K_p \cdot \left(1 + \frac{T_D^x}{T_{\Pi}} \right);$$

$$b_1 = K_p \cdot \left(\frac{T_{\Pi}}{T_I^x} - 1 - 2 \cdot \frac{T_D^x}{T_{\Pi}} \right);$$

$$b_2 = K_p \cdot \frac{T_D^x}{T_{\Pi}}.$$

Если для аппроксимации непрерывного интеграла использовать метод трапеций, то разностное уравнение будет иметь вид:

$$u[n] = K_p \cdot \left[x[n] + \frac{T_{\Pi}}{T_I^x} \cdot \left(\frac{x[n] - x[0]}{2} + \sum_{i=0}^{n-1} x[i] \right) + \frac{T_D^x}{T_{\Pi}} (x[n] - x[n-1]) \right].$$

Для рекуррентного соотношения выявляются отличия только для коэффициента b_0 :

$$b_0 = K_p \cdot \left(1 + \frac{T_D^x}{T_{\Pi}} + \frac{T_{\Pi}}{2 \cdot T_I^x} \right)$$

Запишем рекуррентное соотношение (7.1) для изображений в z -домене:

$$U[z] (1 - z^{-1}) = (b_0 + b_1 z^{-1} + b_2 z^{-2}) X[z],$$

и представим его в виде дискретной передаточной функции



$$W_{PID}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - z^{-1}}. \quad (7.2)$$

Анализ коэффициентов показывает, что если $b_2 = 0$, то получим ПИ-регулятор.

Если $b_0 = 0$, а $b_1 = (1 + b_2)$, то получим ПД-регулятор.

7.4 Электрическая принципиальная схема к лабораторной работе

Схема электрическая принципиальная к лабораторной работе находится у преподавателя.

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. При этом датчик ДД1 подключен к PF.15, ДД2 – к PF.14 и т. д.

В качестве датчиков аналоговых сигналов выступают переменные резисторы R23, R26, R29 сопротивлением 10 кОм. Выходы датчиков АД1, АД2, АД3 подключены к выводам PF.5, PF.4, PF.3 соответственно, одной альтернативной функций которых являются входы 15, 14 и 9 АЦП3 микроконтроллера. Величина напряжения V_a , питающего АЦП, составляет 3,3 В. В качестве источника опорного напряжения для АЦП используется микросхема стабилизатора напряжения DA5 AD1582CRT, на выходе которой формируется напряжение 2,5 В.

В качестве семисегментного символьного индикатора ССИ А4 используется индикатор LTC-4627JS. Индикатор представляет собой модуль из четырех семисегментных светодиодных индикаторов с общим анодом с дополнительным служебным разрядом.

Семисегментный код подается на индикатор через усилитель на базе транзисторов VT2, VT5, VT7, VT9, VT11, VT13, VT16, VT17. Выдача кода осуществляется через порт E (выводы PE.0...PE.7).

Для выбора активного индикатора используется усилитель на базе транзисторов VT3, VT4, VT6, VT8, VT10, VT12, VT14, VT15, VT18, VT19. Выбор индикатора осуществляется через порт B (выводы PB.4...PB.7 и PB.14). В данной схеме, чтобы засветить соответствующий сегмент на индикаторе, необходимо вывести логический 1 в соответствующий разряд порта и, соответственно, 0, чтобы погасить сегмент.

В качестве матричного жидкокристаллического индикатора А6 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...PG.10 (R/W, RS и EN соответственно).

Выход ЦАП микроконтроллера (вывод PA.4) подключен ко входу микросхемы светодиодного индикатора уровня LM3914 DA1. Выбор режима индикации микросхемы осуществляется инвертором, собранным на транзисторе VT22, управлять которым можно, используя вывод порта PB.15. К выходу LM3914 подключена линейка светодиодов А7, LTA-1000HR.



7.5 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.

Задания

1 ПИД-регулятор со следующими параметрами: $K_P = 0,4$; $T_U = 0,2$ 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.

2 ПИ-регулятор со следующими параметрами: $K_P = 0,3$; $T_U = 0,4$ с; $T_I = 28$ с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие – треугольный сигнал с уровнем сигнала от 0 до 0,5 максимального – сгенерировать при ДД1 = 1. Значение управляющего воздействия вывести на ССИ.

3 ПИД-регулятор со следующими параметрами: $K_P = 0,8$; $T_U = 0,1$ 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.

4 ПД-регулятор со следующими параметрами: $K_P = 1$; $T_U = 0,4$ с; $T_D = 1,2$ с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие – пилообразный сигнал с уровнем сигнала от 0,5 до 0 максимального – сгенерировать при ДД7 = 1. Значение управляющего воздействия вывести на ССИ.

5 ПИ-регулятор со следующими параметрами: $K_P = 1$; $T_U = 0,5$ 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.

6 ПИД-регулятор со следующими параметрами: $K_P = 1$; $T_U = 0,4$ с; $T_D = 1$ с; $T_I = 10$ с. Входное воздействие снимается с аналогового датчика АД3. Тестовое воздействие – пилообразный сигнал с уровнем сигнала от 0 до 0,5 максимального – сгенерировать при ДД4 = 0. Значение управляющего воздействия вывести на МЖКИ.

7 ПД-регулятор со следующими параметрами: $K_P = 0,8$; $T_U = 0,5$ с; $T_D = 0,8$ 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 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.

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.

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.

11 ПИ-регулятор со следующими параметрами: $K_P = 0,5$; $T_C = 0,6$ с; $T_I = 60$ 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.

12 ПД-регулятор со следующими параметрами: $K_P = 0,5$; $T_C = 0,6$ с; $T_D = 0,9$ с. Входное воздействие снимается с аналогового датчика АД3. Тестовое воздействие – ступенчатый сигнал с уровнем сигнала от 0 до 0,5 максимального – сгенерировать при ДД5 = 0. Значение управляющего воздействия вывести на ССИ.

13 ПИ-регулятор со следующими параметрами: $K_P = 0,7$; $T_C = 0,2$ с; $T_I = 12$ с. Входное воздействие снимается с аналогового датчика АД2. Тестовое воздействие – пилообразный сигнал с уровнем сигнала от $-0,5$ до $+0,5$ максимального – сгенерировать при ДД3 = 1. Значение управляющего воздействия вывести на МЖКИ.

14 ПИД-регулятор со следующими параметрами: $K_P = 1$; $T_C = 0,6$ с; $T_D = 0,9$ с; $T_I = 45$ 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 ССИ.

15 ПД-регулятор со следующими параметрами: $K_P = 0,2$; $T_C = 0,1$ с; $T_D = 0,4$ с. Входное воздействие снимается с аналогового датчика АД1. Тестовое воздействие – треугольный сигнал с уровнем сигнала от 0,5 до 0 максимального – сгенерировать при ДД4 = 0. Значение управляющего воздействия вывести на МЖКИ.

Пример проекта для микроконтроллера STM32F407ZET6 с реализацией ПИД-регулятора можно взять у преподавателя.

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 **Опейко, О. Ф.** Микропроцессорные средства в автоматизированном электроприводе: учебное пособие / О. Ф. Опейко, Ю. В. Петренко. – Минск : Амалфея, 2008. – 340 с.
- 2 **Джозеф, Ю.** Ядро Cortex M3 компании ARM. Полное руководство: пер. с англ. / Ю. Джозеф. – Москва : Додэка XXI, 2012. – 552 с.
- 3 **Иванов, В. Н.** Электроника и микропроцессорная техника : учебник / В. Н. Иванов, И. О. Мартынова. – Москва : Академия, 2016. – 288 с.
- 4 **Гуров, В. В.** Микропроцессорные системы : учебное пособие / В. В. Гуров. – Москва : ИНФРА-М, 2017. – 336 с.

