



**Некоммерческое  
акционерное  
общество**

**АЛМАТИНСКИЙ  
УНИВЕРСИТЕТ  
ЭНЕРГЕТИКИ И  
СВЯЗИ**

Кафедра инженерной  
кибернетики

## **ЭЛЕМЕНТЫ И УСТРОЙСТВА АВТОМАТИКИ**

Методические указания по выполнению лабораторных работ  
для студентов специальности 5В070200 - Автоматизация и управление

Алматы 2017

СОСТАВИТЕЛЬ: Л.Н.Рудакова. Элементы и устройства автоматики. Методические указания по выполнению лабораторных работ для студентов специальности 5В070200 - Автоматизация и управление. – Алматы: АУЭС, 2017. – 70 с.

Настоящие методические указания включают 11 лабораторных работ по дисциплине «Элементы и устройства автоматики», которые позволят студентам получить практические навыки использования и настройки аналоговых блоков и исполнительных механизмов, а также освоить технологию разработки программ и применения микропроцессоров типа PIC16F877. Контрольные вопросы, приведенные в конце каждой лабораторной работы, помогут закрепить полученные теоретические знания.

В приложении содержится необходимый справочный материал.

Методические указания предназначены для студентов всех форм обучения специальности 5В070200 - Автоматизация и управление.

Ил.38, табл.15, библиогр. – 7 назв.

Рецензент: канд. техн. наук, доцент АУЭС Е.О. Елеукулов

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2017 г.

© НАО «Алматинский университет энергетики и связи», 2017 г.

## Введение

Современное состояние технологических процессов и их автоматизация требует от специалистов знания микроконтроллеров и аналоговых регуляторов.

В настоящее время в составе выпускаемых изделий многих фирм в дальнем и ближнем зарубежье содержатся микроконтроллеры, область применения которых постоянно увеличивается. Микроконтроллеры широкого назначения выпускаются многочисленными зарубежными фирмами: Motorola, NEC Corporation, Siemens, Microchip и другими.

Применение методов и технических средств обработки информации цифровой вычислительной техникой привело к созданию интегрированных комплексов, выполняющих все функции традиционных устройств. Они обладают широкими информационными свойствами и сервисными возможностями, существенно повышающими надежность и эффективность функционирования технических средств автоматического управления электроэнергетическими установками.

Для обучения студентов на кафедре имеется учебный микропроцессорный комплект (УМК-7). Помимо изучения языка Ассемблер на примере микроконтроллера PIC16F877, студенты знакомятся с внутренней и внешней структурой современных микроконтроллеров, применяемых в оборудовании.

Лабораторные работы данных методических указаний посвящены также и изучению аналоговых средств регулирования и управления. Работы выполняются на следующих стендах: стенд блока динамических преобразований БДП-П; стенд системы автоматического регулирования с блоком релейно-импульсного регулятора РП4-У; стенд современных исполнительных механизмов; стенд логического преобразования дискретных сигналов.

При выполнении лабораторных работ необходимо пользоваться принципиальными схемами, приведенными на панели лабораторных установок и в данном описании.

До дня проведения работы студенты должны к ней подготовиться: прочитать описание лабораторной работы и составить программы для своего варианта задания.

Отчет о лабораторной работе должен содержать титульный лист, задание, текст программы, заполненную таблицу результатов, рисунки с экрана и контрольные вопросы.

Лабораторная работа должна быть защищена студентом.

Оборудование и программное обеспечение для проведения лабораторных работ, связанных с МК: Windows 98 или выше, среда MPLAB, комплект УМК-7.

# **1 Лабораторная работа №1. Проверка блока динамических преобразований БДП-П**

**Цель:** приобрести навыки проверки и настройки аналогового блока динамических преобразований БДП-П.

Оборудование и программное обеспечение: стенд блока динамических преобразований БДП-П.

## **1.1 Порядок выполнения**

В ходе выполнения лабораторной работы студент должен выполнить следующие задания:

- задание 1. Знакомство с назначением и принципом действия блока БДП-П;
- задание 2. Определение коэффициента передачи не масштабируемого входа сумматора;
- задание 3. Установка и определение коэффициента масштабирования  $\alpha_{12}$  ( $\alpha_{13}$ );
- задание 4. Определение коэффициента усиления сумматора  $K_c$ ;
- задание 5. Проверка операции «разрешение интегрирования» и измерение постоянной времени интегрирования.

1.1.1 Блоки динамических преобразований типа БДП входят в состав агрегатированного комплекса АКЭСР и предназначены для интегрирования, демпфирования или дифференцирования линейной комбинации входных сигналов с логическим управлением, двухпредельными сигнализацией и ограничением в системах автоматического контроля и регулирования. Интегратор имеет входы логического управления РБ и РМ.

На стенде установлены:

- панель, на которую выведены основные контролируемые величины. Причем входные сигналы выведены на правое наборное поле, выходные выведены на левое поле. Посредине находятся линии с гнездами, обозначенные буквами А, В, С, D, предназначенные для разmultiplication точек ввода или вывода. Общая точка электрической схемы выведена на линию, обозначенную цифрой 4;

- блок БДП-П;
- контролирующие вольтметры со шкалами 0-10 В и 0-1 В.

Блок динамических преобразований содержит сумматор с тремя входами, два из которых масштабируются, интегратор аналогового сигнала, логическую схему управления интегратором и узел двухпредельной сигнализации.

Функциональная схема блока приведена на рисунке 1.1. Сумматор блока имеет три входа:  $X_{11}$  без масштабирования,  $X_{12}$  и  $X_{13}$  с масштабированием. Коэффициент усиления  $S$  сумматора устанавливается ручкой на панели управления блока. Интегратор блока имеет логическое

управление, двухпредельную сигнализацию и ограничитель выходного сигнала.

В приложении А на рисунке А.1 приведена структурная схема блока с указанием назначения входных и выходных зажимов и номеров клемм, а также показаны входы логического управления интегратором.

Интегратор аналогового сигнала выполнен в виде преобразователя (ПНЧ) входного сигнала напряжения  $X_3$  в частоту импульсов, поступающих на вход реверсивного счетчика, цифро-аналоговый преобразователь которого формирует аналоговый выходной сигнал блока интегрирования  $Y_{21}$ . Постоянная времени интегрирования  $T_i$  устанавливается в ПНЧ. Аналоговый сигнал поступает в интегратор через бесконтактные ключи, обеспечивающие запираение и реверс интегратора, и управление от логических схем совпадения «И». Сумматор вместе с интегратором, охваченные различными обратными связями, позволяет осуществлять операции дифференцирования, интегрирования и демпфирования. Узел сигнализации выполнен в виде двух нуль - органов с объединенным входом  $X_2$ , пороги срабатывания которых задаются с помощью элементов масштабирования  $\alpha_B$  и  $\alpha_H$ . Выходные сигналы нуль - органов управляются двумя реле  $P_B$  и  $P_M$  с переключающими контактами.

Различают прямое и обратное интегрирование. При прямом интегрировании изменение выходного сигнала  $Y_{12}$  в процессе интегрирования имеет знак, совпадающий со знаком входного сигнала  $X_3$ . При обратном интегрировании изменение выходного сигнала  $Y_{12}$  в процессе интегрирования имеет знак, противоположный знаку входного сигнала  $X_3$ .

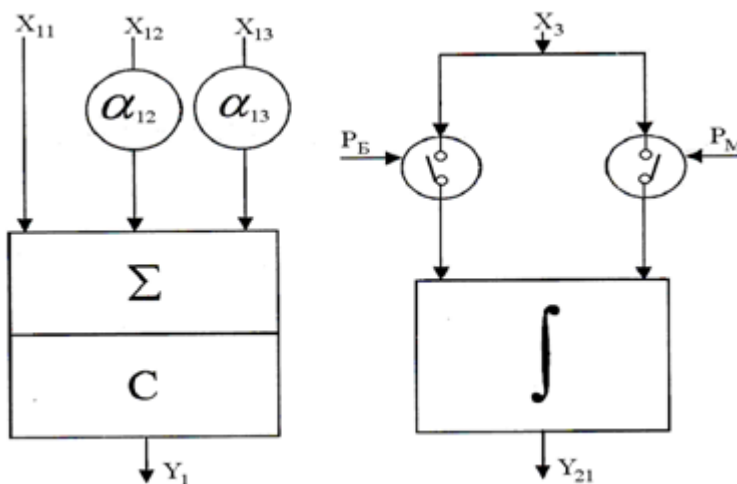


Рисунок 1.1 – Функциональная схема БДП

1.1.2 Включить стенд и приступить к работе с блоком ПДП-П после 3-5 минут прогрева.

Внимание! Общая точка электрических соединений блока обозначена цифрой 4 и выведена на среднее наборное поле.

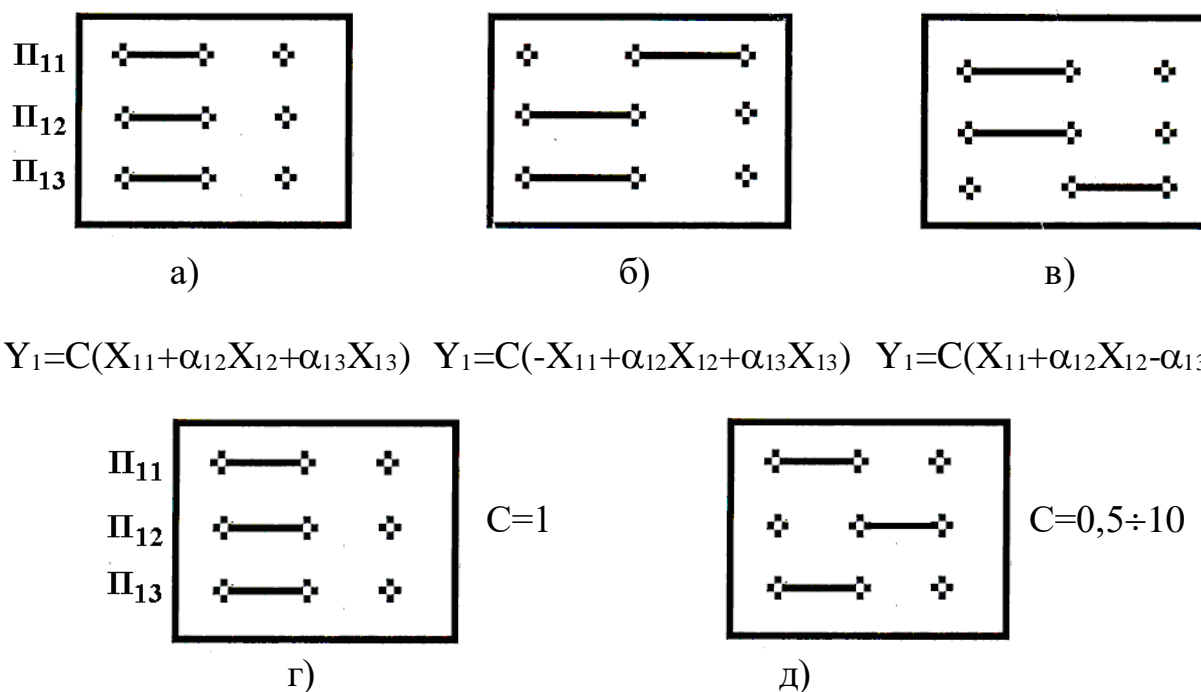


Рисунок 1.2 – Схемы переключателей сумматора

Для определения коэффициента преобразования не масштабируемого входа  $X_{11}$  в сигнал  $Y_1$  следует:

- установить переключки блока в соответствии с рисунком 1.2, а;
- подать опорное напряжение  $+U_{оп}$  (клемма 29) на вход  $X_{11}$  (клемма 21) и измерить вольтметром напряжение на выходе  $Y_1$  (клемма 28). Вычислить коэффициент передачи по формуле

$$K_c = Y_1 / X_{11}; \quad (1.1)$$

- подать опорное напряжение  $-U_{оп}$  с клеммы 27 на  $X_{11}$  (клемма 21) и измерить напряжение на выходе  $Y_1$  (клемма 28). Вычислить коэффициент передачи по формуле

$$K_c = Y_1 / X_{11}; \quad (1.2)$$

- занесите полученные результаты в таблицу 1.1.

Таблица 1.1 – Результаты опытов

Задание 2. Проверка входа $X_{11}$	Задание 3. Проверка входа $X_{12}$ или $X_{13}$		Задание 4. Проверка $K_c$		Задание 5. Интегрирование	
	Установ- лено	Значе- ние $Y_1$	Установ- лено	Получе- но	Задано	Полу- чено
$Y_1 =$	$\alpha =$		$\alpha, C$	$K_c$	$T_i$	$T_i$

1.1.3 Для установки и проверки коэффициента масштабирования  $\alpha_{12}$  необходимо выполнить следующее:

- установить переключки блока в соответствии с рисунком 1.2, а;

– подать опорное напряжение  $+U_{оп}$  на вход  $X_{12}$  (соединить клемму 29 с клеммой 19);

– установить коэффициент  $\alpha_{12}=100\%$  и измерить максимальное значение  $U_M=Y_1$  на клемме 28;

– установить коэффициент  $\alpha_{12}$  согласно варианту по таблице 3 и измерить напряжение  $U_{ср}=Y_1$ ;

– установить коэффициент  $\alpha_{12}=0$  и измерить напряжение  $U_0=Y_1$  вольтметром со шкалой 0-1 В на клемме 28. Занесите полученные результаты в таблицу 2.

Для определения коэффициента масштабирования  $\alpha_{13}$  необходимо выполнить следующее:

– установить переключки блока в соответствии с рисунком 1.2, а;

– подключить опорное напряжение  $+U_{оп}$  на вход  $X_{13}$  (клемму 29 соединить с клеммой 17);

– установить коэффициент  $\alpha_{13}=100\%$  и измерить максимальное значение  $U_M=Y_1$  на клемме 28;

– установить коэффициент  $\alpha_{13}$  согласно варианту по таблице 3 и измерить напряжение  $U_{ср}=Y_1$ ;

– выставить коэффициент  $\alpha_{13}=0$  и измерить напряжение  $U_0=Y_1$  вольтметром со шкалой 0-1 В на клемме 28. Занесите полученные результаты в таблицу 1.1.

Таблица 1.2 – Варианты заданий

Вариант	Задания				
	3		4	5	
	$\alpha_{12}$	$\alpha_{13}$	С, $\alpha_{12}$	Интегрирование	
				прямое	обратное
1	30	-	3,5; 30%	10	-
2	-	30	3; 30%	-	10
3	40	-	4,5; 25%	15	-
4	-	40	4; 20%	-	15
5	50	-	5,5; 20%	20	-
6	-	50	5; 15%	-	20
7	60	-	6,5; 15%	25	-
8	-	60	6; 15%	-	25
9	70	-	7,5; 10%	35	-
10	-	70	7; 10%	-	35

1.1.4 Для проверки коэффициента усиления сумматора  $K_c$  требуется выполнить следующие действия:

– установить переключки блока в соответствии с рисунком 1.2, д;

– при отключенных входах  $X_{11}$ ,  $X_{12}$  и  $X_{13}$  замерить выходное напряжение  $Y_1$  (клемма 28), если оно не равно нулю, откорректировать

вольтметр;

- подать опорное напряжение  $+U_{оп}$  (клемма 29) на вход  $X_{12}$  (клемма 19);

- установить  $\alpha_{12}=10\%$ ,  $C=10$ . По вольтметру зафиксировать значение выходного напряжения сумматора  $Y_1$  (клемма 28). Вычислите  $K_C$  по формуле

$$K_C = Y_1 / X_{12}; \quad (1.3)$$

- плавно поверните ось ручки установки коэффициента «С» против часовой стрелки до значения, указанного в вашем варианте в таблице 1.2. Запишите значение  $Y_1$  и рассчитайте значение  $K_C$ ;

- занесите полученные результаты в таблицу 1.1.

1.1.5 Для проверки операции «разрешение интегрирования» необходимо:

- разобрать предыдущую схему;

- входы логического управления интегратором:  $P_B$  (разрешение интегрирования в прямом направлении),  $P_V$  (разрешения перегона вверх),  $P_H$  (разрешение перегона вниз), то есть клеммы 5, 13, 15 замкнуть на общую клемму 4. Установить постоянную интегрирования  $T$  равную 10 сек;

- подключить вольтметр к выходу интегратора (клемма 26);

- произвести сброс напряжения, кратковременно замкнув вход  $P_C$  (перегон вниз), то есть клемму 8 на общую клемму 4. Вход  $X_3$  (клемма 24) и  $+U_{оп}$  (клемма 29) должны быть при этом разомкнуты;

- подать  $+U_{оп}$  (клемма 29) на вход  $X_3$  интегратора (клемма 24). На выходе интегратора после окончания процесса интегрирования должно установиться напряжение  $+10$  В.

Для измерения постоянной времени при  $+U_{оп}$  необходимо:

- отключить входное напряжение  $+U_{оп}$  с входа интегратора  $X_3$ , разомкнув клеммы 29 и 24;

- установить значение постоянной времени интегрирования левым и правым переключателями  $T$ , согласно варианту по таблице 1.2;

- произвести сброс напряжения, кратковременно замкнув вход  $P_C$  (перегон вниз), то есть клемму 8 на общую клемму 4. Вход  $X_3$  (клемму 24) и  $+U_{оп}$  (клемма 29) должны быть при этом разомкнуты;

- установить значение постоянной времени интегрирования левым и правым переключателями  $T$ , согласно варианту по таблице 1.2;

- подать напряжение  $+10$  В на вход интегратора  $X_3$ , замкнув клемму 29 на клемму 24;

- по секундомеру засечь время совпадения величин напряжения на входе и выходе и в соответствии с рисунком 1.3.

Значение времени с начала подачи напряжения на вход до совпадения величин входного и выходного напряжения даёт с достаточной точностью действительное значение времени интегрирования. Записать экспериментальные значения постоянной времени интегрирования  $T_i$  при интегрировании в сторону «больше» в таблицу 1.1;



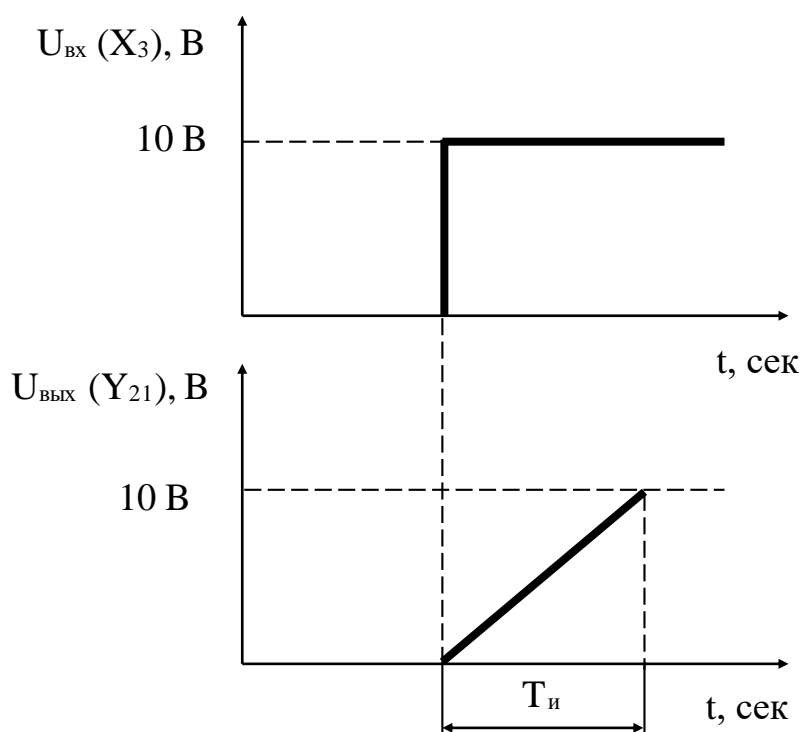


Рисунок 1.3 – Переходная характеристика интегратора

Для проверки операции «разрешение интегрирования» в обратном направлении необходимо:

- разобрать предыдущую схему;
- входы логического управления интегратором:  $P_M$  (разрешение интегрирования в обратном направлении),  $P_V$  (разрешение перегона вверх) и  $P_H$  (разрешение перегона вниз), то есть клеммы 7, 13, 15 замкнуть на клемму 4 (корпус) и установить постоянную интегрирования  $T$  равную 10 сек:
- подключить вольтметр к выходу интегратора;
- произвести сброс напряжения, кратковременно замкнув вход  $P_C^-$  (перегон вниз), то есть клемму 8 на общую клемму 4. Вход интегратора  $X_3$  (клемма 24) и  $+U_{оп}$  (клемма 29) должны быть при этом разомкнуты;
- подать  $-U_{оп}$  (клемма 27) на вход интегратора  $X_3$  (клемма 24).

На выходе интегратора после окончания процесса интегрирования должно установиться напряжение  $+10$  В.

Для проверки установки постоянной времени необходимо:

- отключить входное напряжение  $-U_{оп}$  с входа интегратора  $X_3$ , разомкнув клеммы 27 и 24;
- установить значение постоянной времени интегрирования левым и правым переключателями  $T$ , согласно варианту по таблице 3;
- кратковременным замыканием входа  $P_C^-$  (перегон вниз), то есть клеммы 8 на клемму 4, произвести операцию «перегон» вниз;
- подать опорное напряжение  $-10$  В на вход интегратора  $X_3$ , замкнув клеммы 27 на клемму 24;

– по секундомеру засечь время совпадения величин напряжения на входе и выходе и в соответствии с рисунком 1.3.

Значение времени с начала подачи напряжения на вход до совпадения величин входного и выходного напряжения даёт с достаточной точностью действительное значение времени интегрирования. Записать экспериментальные значения постоянной времени интегрирования  $T_i$  при интегрировании в сторону «меньше» в таблицу 1.1.

## **1.2 Контрольные вопросы**

1.2.1 Назначение блока БДП-П.

1.2.2 Объясните назначение логических входов.

1.2.3 Какие динамические преобразования возможны при помощи блока БДП-П?

1.2.4 Расскажите о сумматоре.

1.2.5 Расскажите об интеграторе.

1.2.6 Как на практике используются полученные в работе сведения о приборе?

## **2 Лабораторная работа №2. Исследование аналогового блока РП4-У и исполнительного механизма постоянной скорости**

**Цель:** получить навыки регулирования на основе релейно-импульсного регулирующего блока и исполнительного механизма постоянной скорости.

Оборудование и программное обеспечение: стенд автоматической системы регулирования (АСР) с блоком релейно-импульсного регулятора РП4-У.

### **2.1 Порядок выполнения**

В ходе выполнения лабораторной работы студент должен выполнить следующие задания:

- задание 1. Изучение функциональных возможностей блока РП4-У;
- задание 2. Определение средней скорости ИМ;
- задание 3. Определение скорости связи прибора РП4-У.

2.1.1 Приборы типа РП4-У используются в автоматической системе регулирования с исполнительными электрическими механизмами в качестве регулирующих элементов. Существуют различные модификации этих приборов, различающиеся специализацией их применения в производстве.

Функциональная схема прибора приведена на рисунке 2.1.

Регулятор имеет вход по напряжению  $U_1$  и четыре входа по току:  $I_1$ ,  $I_2$ ,  $I_3$ ,  $I_4$ , из которых три можно масштабировать с помощью ручек  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ . На выходе регулятор вырабатывает импульсные сигналы. Скважность импульсов

регулируется настройкой коэффициента пропорциональности обратной связи  $\alpha_{\Pi}$  и постоянной времени функциональной обратной связи (ФОС)  $\tau_{\text{И}}$ .

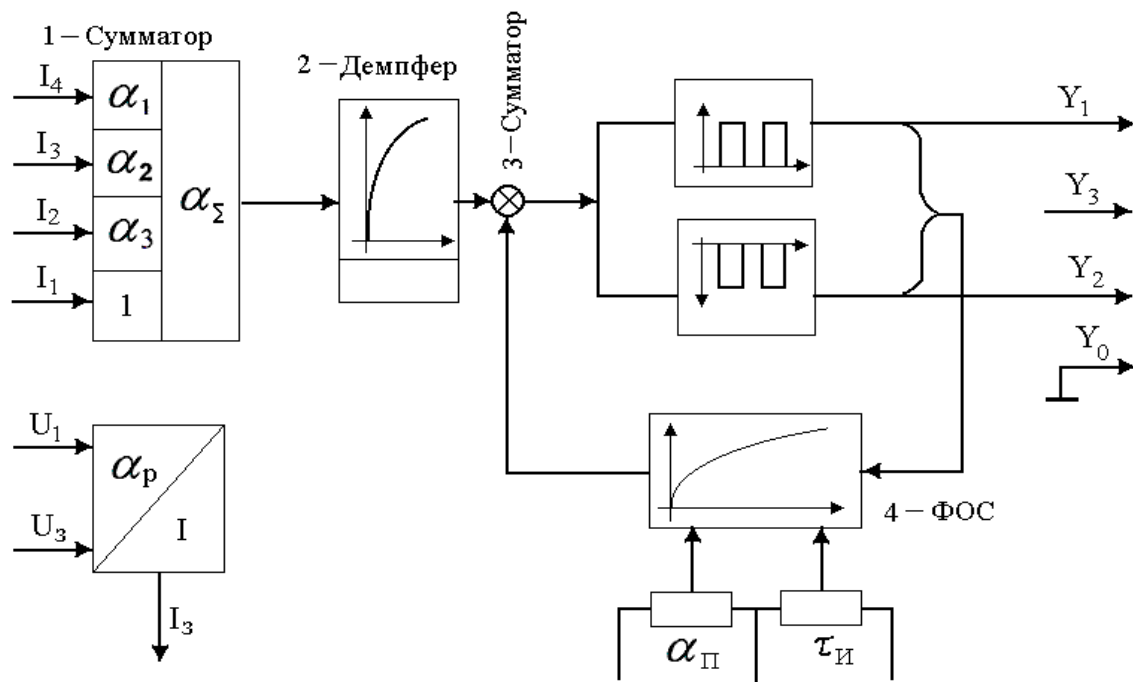


Рисунок 2.1 – Функциональная схема прибора РП4-У

Функциональной обратной связью определяется динамика релейно-импульсного регулятора как замкнутой системы. Эта обратная связь осуществляется по выходу релейно-импульсного регулятора.

Четыре входных сигнала (рисунок 2.1) поступают на вход сумматора 1. Далее сигнал рассогласования через демпфер 2 поступает на вход сумматора 3. На другой вход сумматора поступает сигнал функциональной обратной связи ФОС 4.

Релейный элемент, представленный на рисунке 2.2 с зоной нечувствительности  $\Delta_n$  и зоной возврата  $\Delta_v$ , формирует импульсные сигналы различной полярности с длительностью импульсов  $t_i$  и паузой  $t_p$ . На сигнал условно положительной полярности срабатывает индикатор М, на сигнал условной отрицательной полярности срабатывает индикатор Б. Ручки настройки параметров  $\Delta_v$ ,  $t_i$ ,  $t_p$ ,  $\alpha_{\Pi}$  выведены на боковую панель прибора РП4-У.

Динамика автоматической системы регулирования, имитируемой на данном стенде, определяется двумя основными факторами: регулирующим релейным блоком РБ и исполнительным механизмом ИМ постоянной скорости (рисунок 2.3).

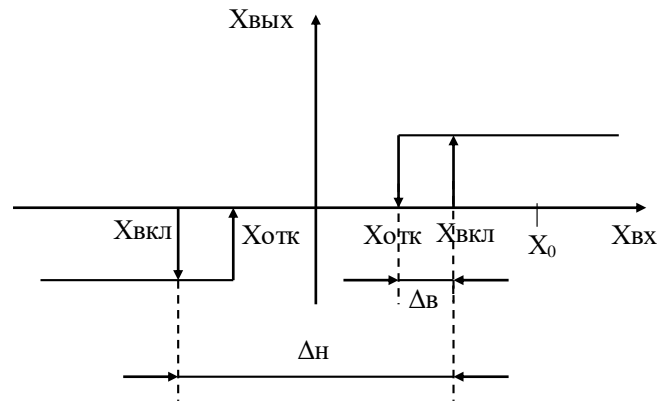


Рисунок 2.2- Релейный элемент

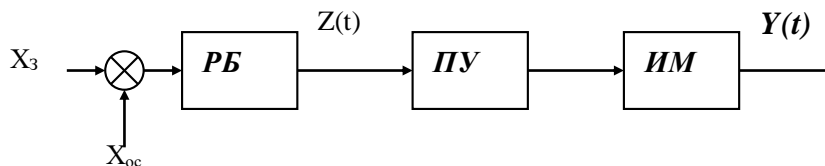


Рисунок 2.3 - Система регулирования

Исполнительный механизм выполнен с редуктором и включается с помощью пускового устройства ПУ. Особенностью ИМ является то, что он рассчитан на определенное напряжение питания и не может управляться за счет изменения его амплитуды. Электродвигатель производит перемещение регулирующего органа  $Y(t)$  с постоянной скоростью  $S_0 = \text{const}$ , не зависящей от амплитуды управляющего сигнала  $Z(t)$ .

При каждом импульсе ИМ регулирующий орган перемещается на величину  $Y = S_0 \cdot t_i$ , а во время пауз положение регулирующего органа не меняется. Под действием управляющих импульсов  $Z(t)$  исполнительный механизм постоянной скорости будет перемещать рабочий орган. График этого перемещения представляется ломаной линией  $Y(t)$ , показанной на рисунке 2.4. Прерывистое перемещение исполнительного механизма можно представить как плавное, происходящее с некоторой усредненной скоростью  $S_{\text{рег}}$ , называемой скоростью регулирования. Она представлена на рисунке 2.4 как аппроксимирующая прямая  $Y_{\text{ид}}$ .

Наклон аппроксимирующей прямой линии можно вычислить по формуле:

$$\text{tg} \beta = S_0 \cdot \frac{t_e}{t_e + t_i} = S_0 \cdot \gamma. \quad (2.1)$$

Параметр  $V_{\text{на}} = |\text{tg} \alpha| = \left| \frac{d\tilde{O}_{\text{ин}}(t)}{dt} \right|_{t=t_{\text{ил}}} = C \cdot \frac{\hat{E}_{\text{ин}}}{\dot{O}_{\zeta}}$  называют скоростью связи

или скоростью изменения сигнала обратной связи под действием  $z = \pm C$  при включении релейного элемента, вычисленное за время заряда  $T_3$ .

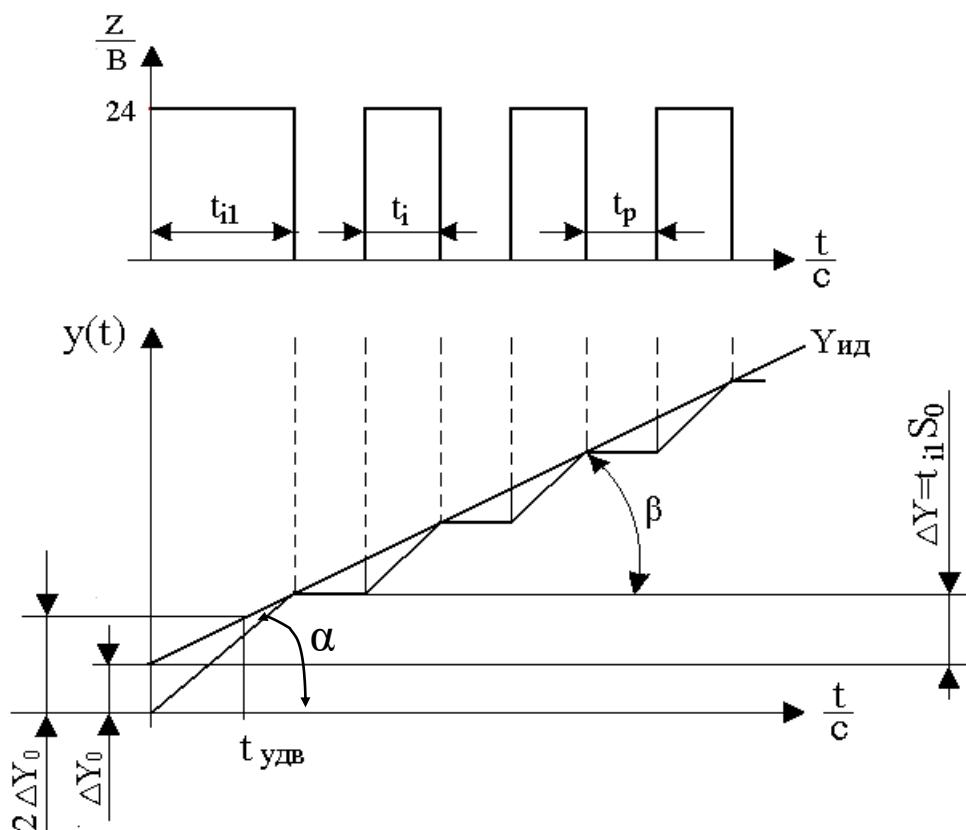


Рисунок 2.4 – Реакция ИМ на ступенчатый входной сигнал

На приборе РП4-У настроечным параметром скорости связи является коэффициент:

$$\alpha_i = \frac{1}{V_{\text{н.д.}}} \cdot \left[ \frac{c}{\%} \right]. \quad (2.2)$$

Блок РП4-У позволяет устанавливать настроечные параметры независимо друг от друга. Длительность импульсов  $t_i$  практически не зависит от значения входного сигнала  $X_3$ , тогда как длительность паузы  $t_p$  обратно пропорциональна  $X_3$ . При этом скважность  $\gamma(t)$  управляющих импульсов  $Z(t)$  зависит от  $X_3$  практически линейно.

2.1.2 Для определения средней скорости исполнительного механизма (ИМ) следует:

1) На блоке БУ21 установить переключатель в положение «Р» (ручной режим). Кнопками «Больше» и «Меньше» блока БУ21 опробовать перемещение ИМ. Затем переместить его в среднее положение, контролируя его положение по указателю положения УП (УП=50%).

2) Измерить скорость ИМ в положительном направлении (в сторону «Больше» на РП4-У). Для этого в ручном режиме кнопкой «Больше» на БУ21 переместить ИМ от среднего положения до положения 70% от возможно хода (УП=70%) и зафиксировать время его движения. Вычислить скорость ИМ в сторону «Больше» по формуле:

$$S_{им i}^{+} = \frac{\Delta УП_i [\%]}{\Delta t_i [c]} = \frac{20\%}{\Delta t_i [c]}, \quad (2.3)$$

где  $\Delta t$  – время перемещения исполнительного механизма от одного положения до другого, указываемого с помощью указателя положения УП;

$\Delta УП\% = 70\% - 50\% = 20\%$  изменение положения механизма за время  $\Delta t$ .

3) Измерить скорость ИМ в отрицательном направлении (в сторону «Меньше» на РП4-У). Для этого повторить опыт пункта 2, управляя исполнительным механизмом кнопкой «Меньше» на БУ21. Вычислить скорость ИМ в сторону «меньше» по формуле:

$$S_{им i}^{-} = \frac{\Delta УП_i [\%]}{\Delta t_i [c]} = \frac{20\%}{\Delta t_i [c]}, \quad (2.4)$$

где  $\Delta УП\% = 50\% - 30\% = 20\%$  изменение положения механизма за время  $\Delta t$ .

4) Измерения по пунктам 2 и 3 повторить три раза и вычислить среднее значение скорости в обоих направлениях

5) Затем вычислить окончательно среднюю скорость ИМ:

$$S_{cp} = \frac{S_{им cp}^{+} + S_{им cp}^{-}}{2}. \quad (2.5)$$

2.1.3 Скорость связи прибора РП4-У - это скорость изменения сигнала обратной связи под действием импульса при включении релейного элемента.

Для определения скорости связи необходимо:

1) На блоке БУ21 установить переключатель в положение «Р» (ручной режим). Установить на панели РП4-У коэффициент передачи сумматора  $\alpha_{\Sigma} = 1$ . Установить остальные параметры: коэффициент пропорциональности функциональной обратной связи  $\alpha_{п}$ , постоянную времени ФОС (она же постоянная времени интегрирования)  $\tau_{и}$ , минимальную величину продолжительности импульса  $t_i$  в положения согласно варианту по таблице 2.1. Ручки входного задатчика «Задание» на панели РП4-У установить в нулевое положение.

Таблица 2.1 – Варианты заданий

Вариант	Установленное значение коэффициента пропорциональности $\alpha_{п}$	Постоянная интегрирования $\tau_{и}$	Величина продолжительности импульса $t_i$
1	0.65	5	0.3
2	0.8	5	0.4
3	0.8	10	0.5
4	1.1	5	0.3
5	1.1	5	0.4
6	1.1	10	0.5
7	1.1	10	0.4
8	1.4	5	0.5
9	1.4	10	0.4
10	1.4	10	0.3

2) Сбалансировать блок РП4-У. Для этого на панели «Задание» ручку «Грубо» установите на ноль. Ручку «Точно» поворачивайте в разные стороны до тех пор, пока не погаснут оба индикатора «Б» и «М».

3) Чтобы снять зависимость скорости связи  $V_{св}(\alpha_{дп})$  от установочного параметра  $\alpha_{п.уст}$ , заметьте положение точной ручки «Задание» и передвиньте ее на 2%. К выходу РП4-У (к клеммам  $Y_0$  и  $Y_2$ ) подключить вольтметр на 10 вольт. Затем следует отключить питание на тренажер тумблером на панели блока.

4) Подать напряжение питания на РП4-У (включить питание) и измерить секундомером длительность первого импульса  $t_{и1}$ . Скорость связи и действительный параметр  $\alpha_{дп}$  определяются по формулам:

$$V_{\dot{n}\dot{a}} = \frac{\Delta U_{\dot{a}\dot{o}}}{t_{и1}} = \frac{2\%}{t_{и1}}, \quad \alpha_{\dot{a}\dot{i}} = \frac{t_{и1}}{\Delta U_{\dot{a}\dot{o}}}, \quad (2.6)$$

где  $\Delta U_{вх}$  - напряжение на входе РП-4У;

$t_{и1}$  – длительность первого импульса при включении РП4-У.

Опыты повторите для всех значений  $\alpha_{п.уст}$  по шкале прибора: (0,5; 0,65; 0,8; 1,1; 1,4; 1,8 2.2). Результаты занести в таблицу 2.2.

Таблица 2.2 – Результаты выполнения задания 3

Номер опыта	$\Delta U_{вх}, \%$	$\alpha_{п.уст}, \text{с}/\%$	$t_{и1}, \text{с}$	$\alpha_{дп}, \text{с}/\%$	$V_{св}, \%/с$
1	2				
2	2				
3	2				

По результатам построить градировочную характеристику  $\alpha_{дп}=f(\alpha_{п.уст})$ .

## 2.2 Контрольные вопросы

2.2.1 Приведите структурную схему регулирования.

2.2.2 Какими настроечными параметрами блока РП4-У можно изменить коэффициент пропорциональности и постоянную времени интегратора?

2.2.3 Почему для регулирования от ИМ постоянной скорости используется релейно-импульсные блоки?

2.2.4 Поясните физический смысл скорости связи.

2.2.5 Как изменить среднюю скорость передвижения ИМ?

2.2.6 Какие физические параметры характеризуют  $\text{tg}\alpha$  и  $\text{tg}\beta$  на рисунке 2.4?

2.2.7 Что значит «Больше» и «Меньше»?

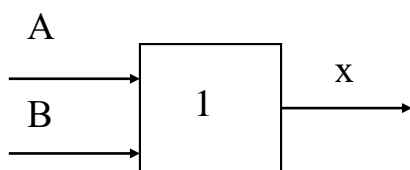
### 3 Лабораторная работа №3. Логическое преобразование дискретных сигналов

**Цель:** отработать реализацию алгоритмов логического преобразования дискретных сигналов на электромагнитных реле и приобрести практические навыки в составлении типовых фрагментов принципиально электрических схем устройств дискретного логического управления.

#### 3.1 Общие сведения

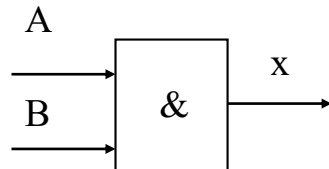
3.1.1 Управляющая часть систем дистанционного и дискретного управления операциями пуска, останова технологического процесса синтезируется с использованием элементарных алгоритмов логического преобразования дискретных сигналов (например, А и В ступенчатой или импульсной формы):

«ИЛИ»



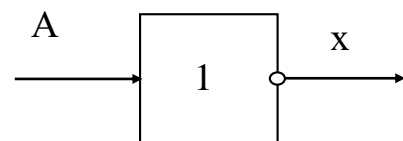
$$\hat{A} + \hat{A} = \tilde{o};$$

«И»



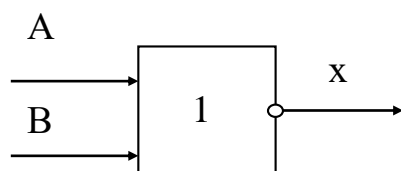
$$\hat{A} \times \hat{A} = \tilde{o};$$

«НЕ»



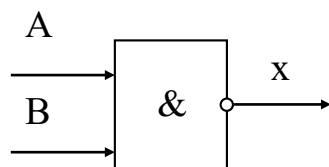
$$\hat{A} = \tilde{o};$$

«ИЛИ-НЕ»



$$\hat{A} + \hat{A} = \tilde{o};$$

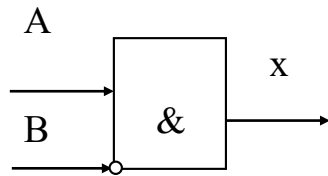
«И-НЕ»



$$\hat{A} \times \hat{A} = \tilde{o};$$

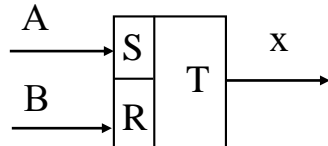


«Запрет»



$$A \times \bar{B} = x;$$

«Память»



$X = \langle 1 \rangle$  при  $A = \langle 1 \rangle$

$A^*$  - импульсная форма;

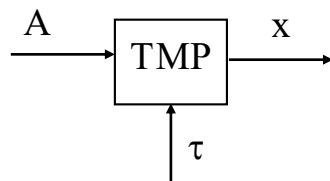
$B^*$  - импульсная форма;

$x$  – ступенчатая форма.

В этом случае возможны два состояния:

$$x = A^* \times \bar{B} \text{ и } \bar{x} = A \times B^*$$

«Память»



$$x = A \times (t + \tau),$$

где  $t$  – момент времени появления дискретного сигнала  $A$ ;

$\tau$  - время задержки формируемое таймером.

«Одновибратор» (выделение переднего сигнала  $A$ )



$$x^* = A, \text{ где } x^* - \text{импульсный сигнал определенной длительности } \Delta t.$$

Существуют и другие виды более сложных алгоритмов логического преобразования дискретных сигналов такие, как выделение заднего фронта сигнала  $A$ ; счёта импульсов, генерации импульсных сигналов и т.д.

Выходные сигналы  $x$  элементарных алгоритмов либо непосредственно являются выходными командами на различные исполнительные механизмы, либо являются промежуточными командами для реализации более сложных алгоритмов управления.

Указанные алгоритмы реализуются на различной элементной базе (интегральные мнемосхемы, электромагнитные реле). Кроме того, указанные

алгоритмы могут быть реализованы с помощью программного обеспечения и содержаться в библиотеках алгоритмов различных модификаций микропроцессорных контроллеров с определенными буквенными или цифровыми кодами.

Источниками дискретных входных сигналов являются:

а) аналогово-дискретные преобразователи датчиков технологических параметров (расход, температура, уровень и т.д.) и электрических величин (ток, напряжение и т.д.);

б) дискретные датчики понижения запорной и регулирующей арматуры (концевые выключатели и т.д.);

в) органы формирования команд ступенчатой и импульсной формы (ключи, кнопки, тумблеры и т.д.).

### 3.2 Описание лабораторного стенда

Общий вид стенда «Логическое преобразование дискретных сигналов» представлен на рисунке 3.1. Стенд представляет собой набор модулей, на верхней части корпуса расположен исполнительный механизм с концевыми выключателями датчиком положения.

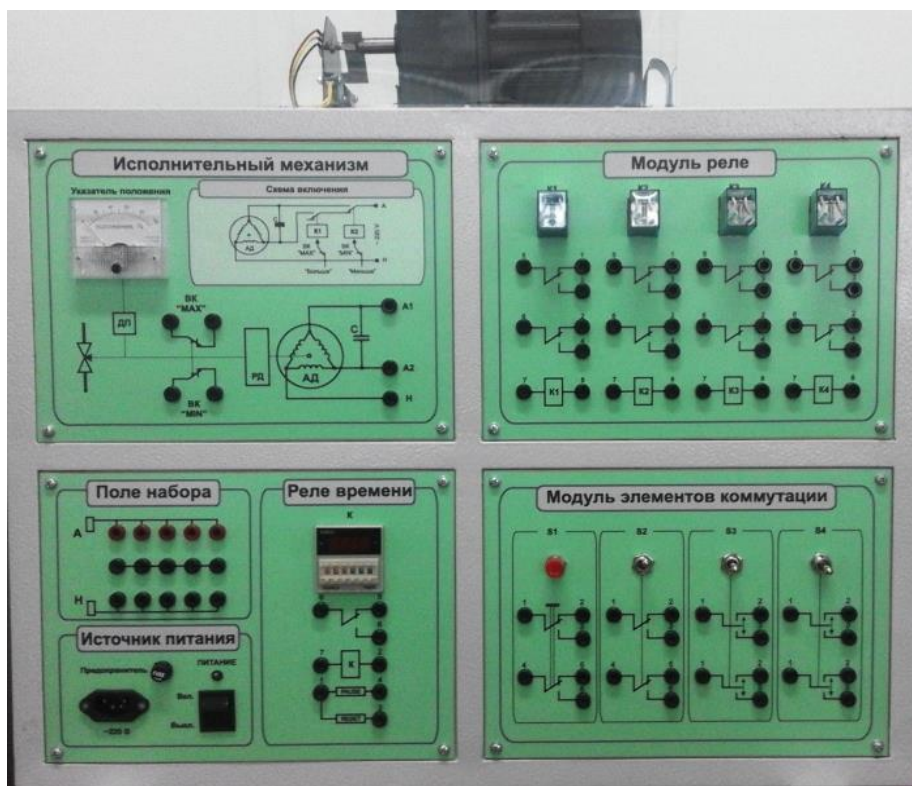


Рисунок 3.1 – Общий вид стенда

В левом нижнем углу стенда расположены:

- модуль «Источник питания»;
- модуль «Поле набора»;
- модуль «Реле времени».

3.2.1 На модуле «Поле набора» располагаются клеммы соединительной шины фазы, обозначенные «А» красного цвета, клеммы соединительной шины нейтрали, обозначенные «Н» черного цвета, а также клеммы промежуточной соединительной шины, располагающейся между шинами фазы и нейтрали.

Внимание! Все соединения должны выполняться при отключенном источнике питания стенда. Питание на шины фазы и нейтрали подается при включении источника питания стенда.

На модуле «Реле времени» располагается таймер типа DH48S-S, принципиально-электрическая схема которого представлена на рисунке 3.2. Схема состоит из катушки реле (контакты 7 и 2 на схеме), при подаче питания на которую начинается отсчет времени. Отсчет производится отдельно для времени во включенном состоянии реле и отдельно для выключенного состояния. Катушка реле управляет переключающей группой трёх контактов, один из которых (согласно изображенной на модуле схеме) нормально-замкнутый (контакт 5 на схеме), второй - нормально-разомкнутый (контакт 6 на схеме).

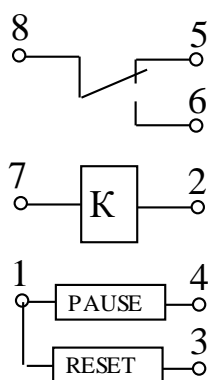


Рисунок 3.2 - Принципиально-электрическая схема таймера

Питание на реле времени подается при помощи внешних соединительных проводников с модуля «Поле набора». При подаче питания автоматически начинается отсчет времени. Отсчет начинается с включенного состояния реле времени, затем производится отсчет времени выключенного состояния, далее, если питание с катушки реле не снимается, цикл отсчетов времени повторяется.

Контакты 1, 3 и 4 на схеме являются зарезервированными для служебного использования и не должны соединяться в процессе выполнения работ на стенде.

Основные элементы управления реле времени показаны на рисунке 3.3.

Индикаторы 1 и 2 показывают количество отсчитанных интервалов времени. Когда идет отсчет включенного состояния реле – работает индикатор 1, для отсчета выключенного состояния – работает индикатор 2.

Кнопки 4 и 7 позволяют выбирать числовое значение (от 0 до 99) интервалов времени и единицы измерения выбранных интервалов времени (0,01 сек; 0,1 сек; 1 сек; 0,1 мин; 1 мин; 0,1 ч; 1 ч; 10 ч). Выбор интервала времени производится обязательно до подачи питания на катушку реле.



- 1 - индикатор отсчета времени включенного состояния реле;
- 2 - индикатор отсчета времени выключенного состояния реле;
- 3 - индикатор единиц отсчета интервала времени для включенного состояния;
- 4 - кнопки уменьшения значений;
- 5 - индикация выбранного числового значения интервалов времени;
- 6 - индикатор единиц отсчета интервала времени для выключенного состояния;
- 7 - кнопки увеличения значений.

Рисунок 3.3 - Элементы управления реле

Индикатор 5 показывает числовое значение выбранных интервалов времени. Индикатор 3 показывает выбранные единицы измерения интервала времени для включенного состояния реле. Индикатор 6 показывает выбранные единицы измерения интервала времени для выключенного состояния реле.

3.2.2 Внешний вид модуля «Исполнительный механизм» приведен на рисунке 3.4. В нижней части модуля приведена технологическая схема исполнительного механизма, состоящая из асинхронного трехфазного электродвигателя (АД), который через редуктор (РД) управляется регулирующей арматурой. К оси редуктора на выходе присоединены механически концевые выключатели (ВК«МАХ» и ВК«MIN»), которые обеспечивают защиту регулирующей арматуры, имеющей ограничения по

углу вращения, от повреждения. Также к оси редуктора механически присоединен электрический датчик положения.



Рисунок 3.4 - Модуль «Исполнительный механизм»

На входе технологической схемы расположены входные клеммы для подключения цепей управления исполнительным механизмом, обозначенные А1, А2, Н.

В верхней левой части модуля расположен указатель положения, отображающий положение вала исполнительного механизма в текущий момент времени и получающий сигнал о положении с датчика положения.

В верхней правой части модуля расположена принципиальная схема управления исполнительным механизмом на основе асинхронного двигателя. Данную схему студенты должны собрать самостоятельно, используя необходимые элементы, расположенные на различных модулях стенда.

3.2.3 В правом верхнем углу стенда расположены 4 реле К1 – К4 типа OMRON MY2NJ. Реле имеют катушку управления и 2 группы переключающих контактов. Принципиально-электрическая схема реле представлена на рисунке 3.5.

3.2.4 В правом нижнем углу стенда расположены 4 переключателя S1-S4, принципиально-электрические схемы которых представлены на рисунке 3.6.

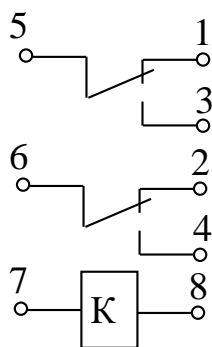


Рисунок 3.5 - Принципиально-электрическая схема реле K1

Переключатель S1 (рисунок 3.6, а) представляет собой возвратную кнопку (орган формирования дискретных команд импульсной формы), имеет 2 группы переключающих контактов.

Переключатель S2 (рисунок 3.6, б) представляет собой тумблер на 2 положения с фиксацией в каждом из положений (орган формирования дискретных команд ступенчатой формы), имеет 2 группы переключающих контактов.

Переключатель S3 (рисунок 3.6, в) представляет собой возвратный ключ на 3 положения с возвратом из крайних положений в среднее (орган формирования двух импульсных команд типа «открыть-закрыть»), имеет 2 группы замыкающих контактов.

Переключатель S4 (рисунок 3.6, г) представляет собой ключ на 3 положения с фиксацией в каждом из трех положений (орган формирования двухступенчатых команд типа «ручное-отключено-автоматическое»), имеет 2 группы замыкающих контактов.

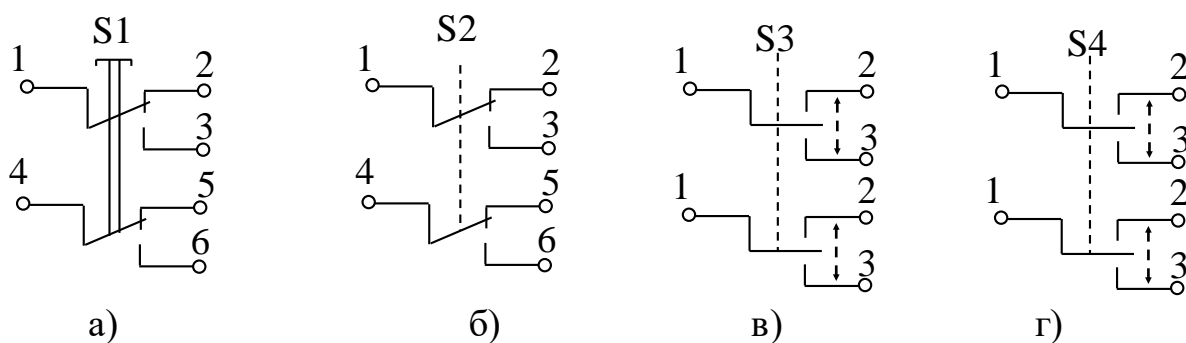


Рисунок 3.6 - Принципиально-электрические схемы переключателей S1- S4

### 3.3 Порядок выполнения

3.3.1 Составить принципиально-электрические схемы (ПЭС) управления отдельными релейными элементами от команд, формируемых органами S1, S2, S3 и S4 согласно заданным вариантам.

Вариант 1. Составить ПЭС управления АД посредством S1 и S2 ( $\overline{S1}$  и S2).

Вариант 2. Составить ПЭС управления реле K2 посредством S1 или S2 ( $\overline{S1}$  или S2).



Вариант 3. Составить ПЭС управления реле К4 посредством S1 и S2 ( $\overline{S1}$  и  $\overline{S2}$ ).

Вариант 4. Составить ПЭС управления АД посредством S1 или S2 ( $\overline{S1}$  или S2).

Вариант 5. Составить ПЭС управления реле К1 посредством S1 и S3 ( $\overline{S1}$  и S3).

Вариант 6. Составить ПЭС управления реле К3 посредством S3 или S4 (S3 или  $\overline{S4}$ ).

Вариант 7. Составить ПЭС управления реле К4 посредством ключа S3.

Вариант 8. Составить ПЭС управления реле К2 посредством переключателя S4.

Примечание - Каждый студент выполняет три варианта ПЭС по указанию преподавателя.

3.3.2 Выполнить анализ составленных ПЭС:

- представить таблицу состояния входных сигналов (S1, S2, S3 и S4) и выходных сигналов (состояние электромагнитной системы реле - в рабочем или не рабочем положении, т.е. наличие или отсутствие напряжения на электромагните);

- составить временную диаграмму работы устройств в соответствии с составленной ПЭС.

3.3.3 Используя соединительные проводники, собрать ПЭС на наборном поле стенда; затем представить собранную схему преподавателю для проверки.

3.3.4 Получив разрешение преподавателя, выполнить следующие действия:

- с соблюдением мер предосторожности подать питание на шины А и Н;
- изменяя состояние входных сигналов (S1, S2, S3 и S4), проверить работу устройства в соответствии с таблицей состояния.

3.4.5 Составить отчёт по выполненным вариантам ПЭС и произвести защиту лабораторной работы.

Пример 1. Составить ПЭС управления реле К1 посредством S1, обеспечив с помощью переключающей контактной группы реле К1 подачу напряжения на электромагнитное реле К1 и К2 (рисунок 3.7).

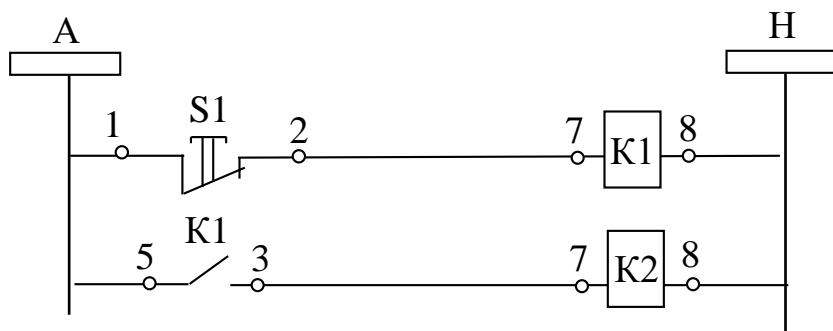


Рисунок 3.7 - Принципиально-электрическая схема управления реле К1

Представить таблицу состояния входных сигналов и выходных сигналов (таблица 3.1). Составить временную диаграмму работы устройств в соответствии с составленной ПЭС (рисунок 3.8).

Таблица 3.1 - Таблица состояния

S1	K1	K2
0	0	0
0	1	1
1	0	0

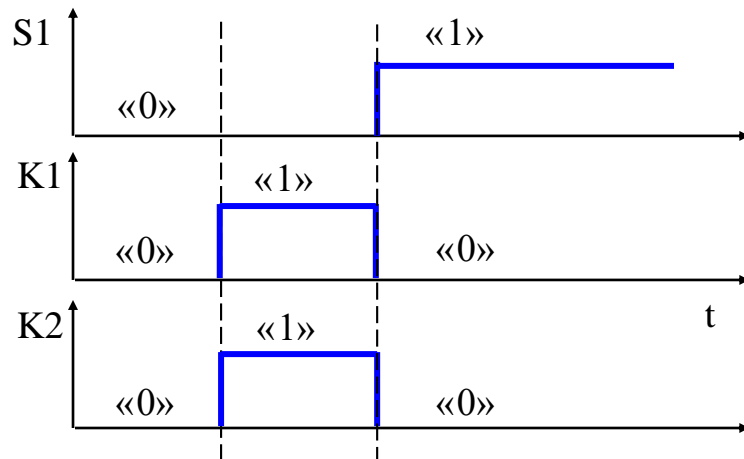


Рисунок 3.8 - Временная диаграмма работы устройств

Пример 2. Составить ПЭС управления асинхронным трехфазным электродвигателем АД с помощью реле времени (рисунок 3.9).

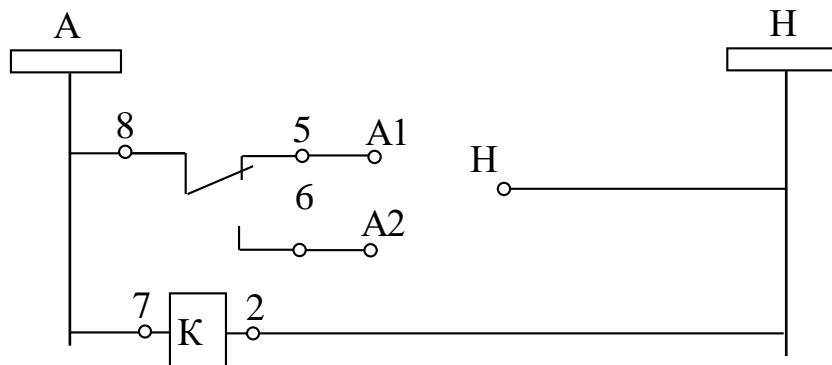


Рисунок 3.9 - Принципиально-электрическая схема управления асинхронным трехфазным электродвигателем АД

### 3.5 Контрольные вопросы

3.5.1 Каким способом управляется электромагнитное реле (ЭМР)?

3.5.2 Что является входным и выходным сигналом для ЭМР?

3.5.3 Какие элементарные алгоритмы логического преобразования дискретных сигналов S1, S2, S3 и S4 реализуются устройствами в соответствии с составленными вами ПЭС?



## 4 Лабораторная работа №4. Двигатель постоянного тока: параметры и характеристики

**Цель:** изучить параметры и характеристики работы электрического двигателя постоянного тока в различных режимах.

### 4.1 Общие сведения

Двигатели постоянного тока (ДПТ) предназначены для превращения энергии постоянного тока в механическую работу. Они реже используются, нежели двигатели переменного тока. Это связано со сравнительной дороговизной, более сложным устройством, трудностями в обеспечении питания. Однако ДПТ отлично регулируются различными способами, имеют более жесткие механические характеристики и позволяют обеспечить большой пусковой момент.

Электродвигатели постоянного тока применяются в качестве тяговых двигателей, в электротранспорте и в качестве различных исполнительных устройств.

Лабораторный комплекс состоит из:

- модуля питания (МП) с встроенным универсальным измерительным прибором (УИП) (рисунок 4.1);
- модуля универсального драйвера (МД) (рисунок 4.2);
- модуля внешних соединений шагового двигателя (рисунок 4.3);
- модуля внешних соединений ДПТ (МДТП) (рисунок 4.4).

На рисунках 4.1 – 4.4 приведены названия и назначения - клемм, разъемов, тумблеров, кнопок и индикаторов лабораторного стенда.

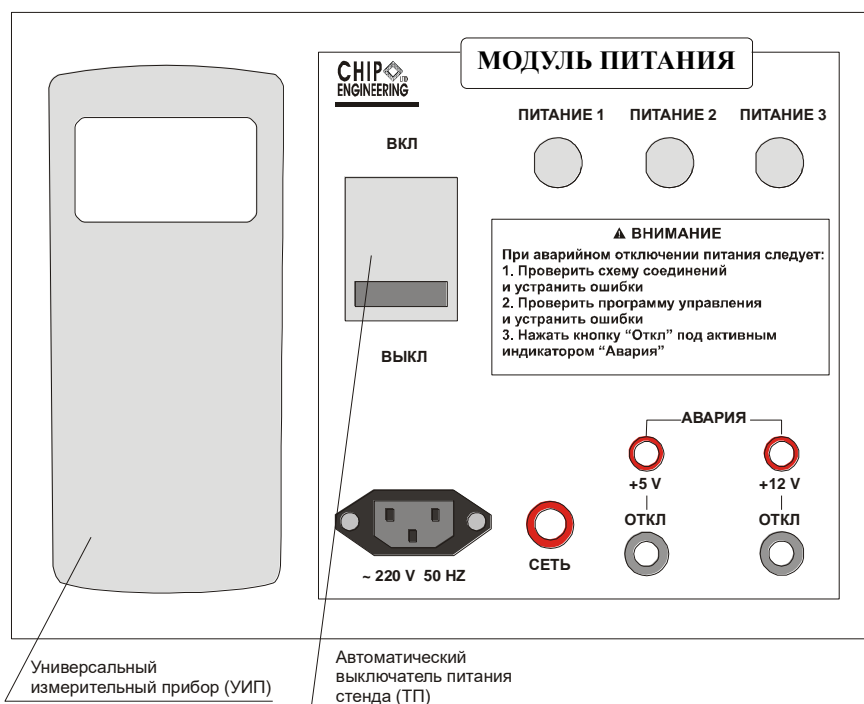


Рисунок 4.1 – Модуль питания с встроенным универсальным измерительным прибором

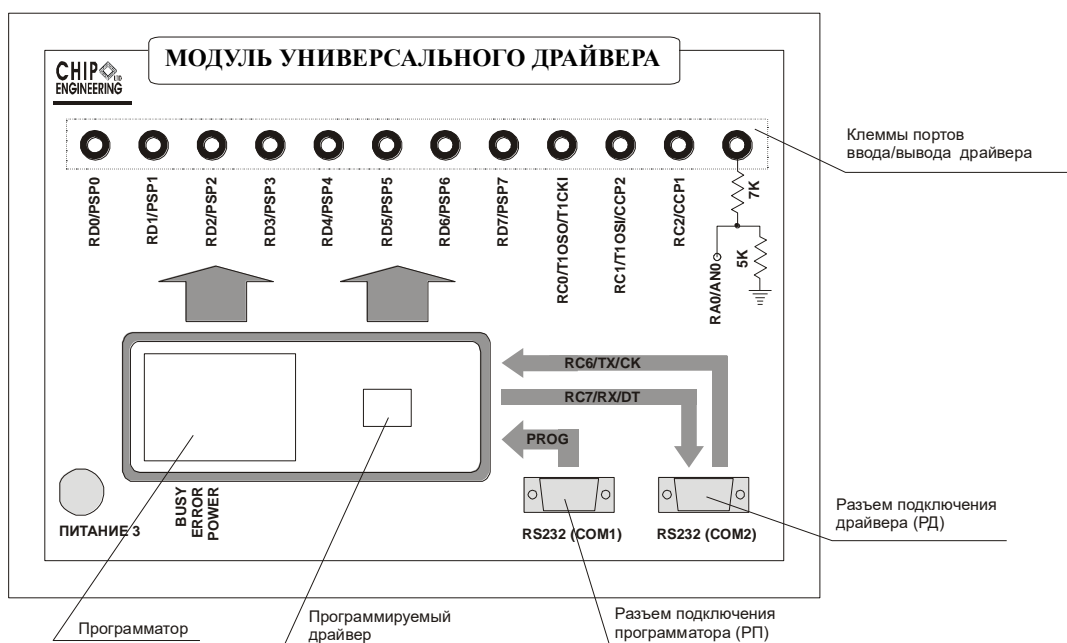


Рисунок 4.2 – Модуль универсального драйвера

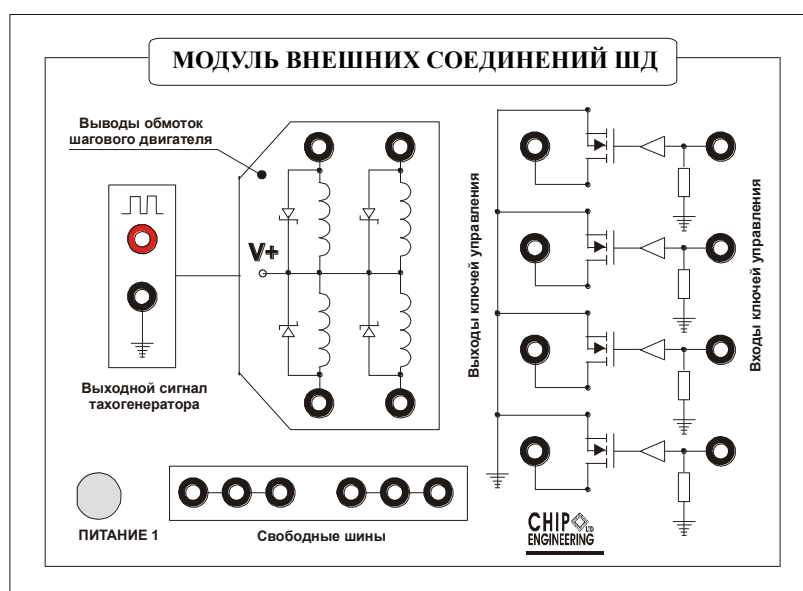


Рисунок 4.3 – Модуль внешних соединений ШД

На валах ДПТ и ШД установлены оптические тахогенераторы, на выходных клеммах которых можно измерить частотные сигналы  $F1t$  и  $F2t$  пропорциональные частоте вращения валов двигателей:

- для ДПТ  $F1t=8 \times F1m$ , где  $F1m$  – частота вращения вала ДПТ;
- для ШД  $F2t=24 \times F2m$ , где  $F2m$  – частота вращения вала ШД.

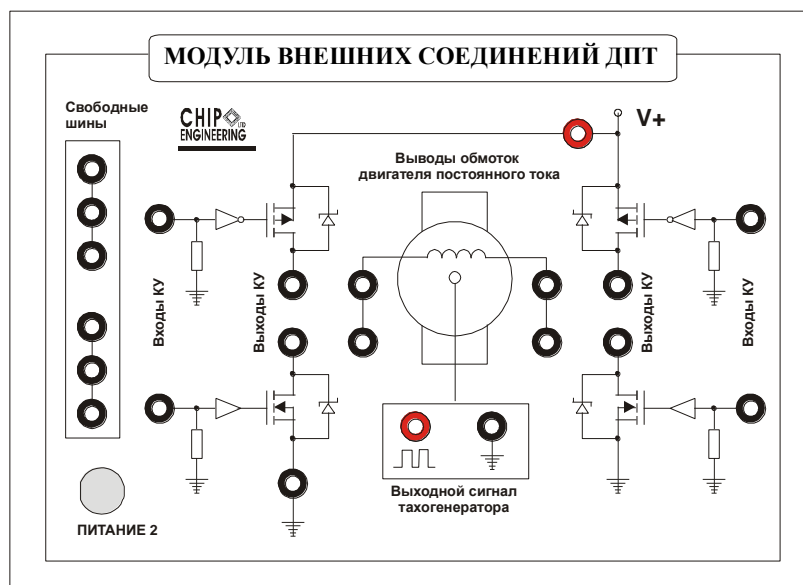


Рисунок 4.4 – Модуль внешних соединений ДПТ

## 4.2 Порядок выполнения

4.2.1 Подготовка стенда к выполнению лабораторной работы включает следующие действия:

- отсоединить все подключенные клеммы модулей стенда (если таковые имеются);
- проверить подключение шнура питания стенда и соединение разъемов ПИТАНИЕ с соответствующими номерами;
- проверить соединение разъемов РП и РД с разъемами ПК COM1 и COM2 соответственно;
- включить питание стенда;
- проверить состояние индикаторов СЕТЬ – горит, АВАРИЯ – не горят:
  - а) если один или оба индикатора АВАРИЯ горят, то нажать кнопку «Откл» под активным индикатором «Авария»;
  - б) если индикаторы продолжают гореть, то проверить схему соединений и устранить ошибки;
- нажать кнопку «Откл» под активным индикатором «Авария»;
- проверить работоспособность УИП.

4.2.2 При аварийном отключении питания следует:

- проверить схему соединений и устранить ошибки;
- проверить программу управления и устранить ошибки;
- нажать кнопку «Откл» под активным индикатором «Авария».

4.2.2 Для определения параметров ДПТ необходимо:

- выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 4.5;
- включить питание стенда. Двигатель должен вращаться с постоянной скоростью. Запомнить направление вращения двигателя;
- выключить питание стенда и изменить полярность подключения;

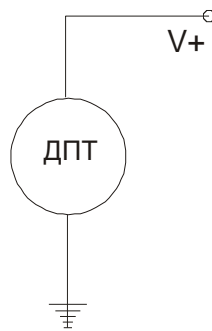


Рисунок 4.5 – Схема соединения двигателя постоянного тока

- включить питание стенда, наблюдать смену направления вращения;
- выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 4.6, а;

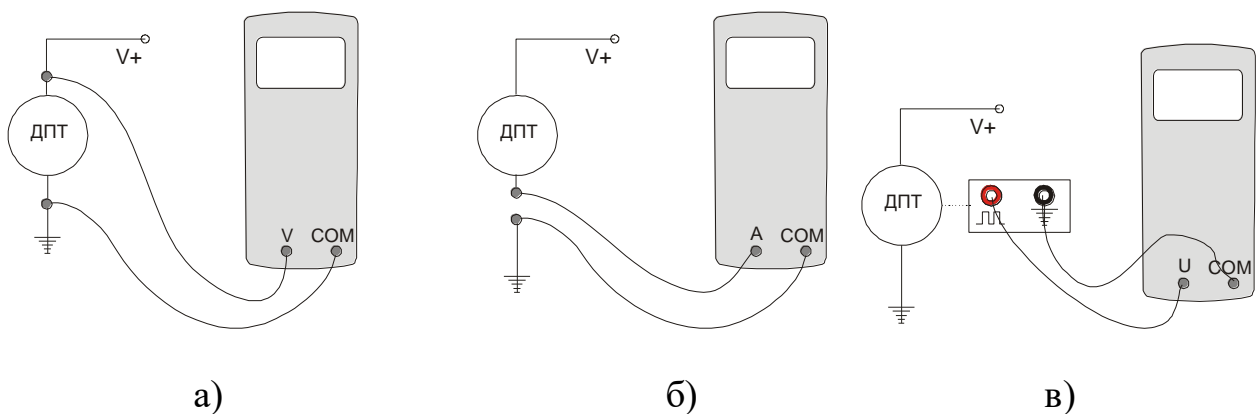


Рисунок 4.6 – Схемы соединения двигателя постоянного тока

- установить УИП в режим измерений постоянного напряжения, включить питание стенда, произвести измерение и записать значение питающего напряжения двигателя  $U$ ;
  - выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 4.6, б;
  - установить УИП в режим измерений постоянного тока, включить питание стенда, произвести измерение и записать значение тока холостого хода  $I$ ;
  - выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 4.6, в;
  - установить УИП в режим измерений частоты, включить питание стенда, произвести измерение и записать значение частоты вращения ротора  $f$ .
  - выключить питание стенда, произвести вычисление мощности  $W=UI$ .
- Сравнить полученные значения  $U$ ,  $I$ ,  $f$ ,  $W$  с паспортными данными двигателя (паспортные данные нормализовать для измеренного значения напряжения  $U$ ).

### 4.3 Контрольные вопросы

4.3.1 Каковы достоинства и недостатки двигателей постоянного тока?

4.3.2 Объяснить смену направления вращения ДПТ при изменении полярности питающего напряжения.

4.3.3 Как рассчитать диапазон измерений тока для УИП при следующих характеристиках ДПТ:  $U=12\text{ В}$ ,  $W_{\max}=25\text{ Вт}$ ?

## 5 Лабораторная работа №5. Изучение среды MPLAB. Управление оборудованием

**Цель:** изучить среду MPLAB, принципы подключения технологического оборудования к микропроцессору (МК), изучить технику управления регистром PORTC и инструкции установки отдельных бит в регистрах.

Задания для внеаудиторной подготовки:

- ознакомиться с описанием УМК-7 и MPLAB (приложение Б);
- изучить биты RP1 и RP0 (5 и 6 разряды) регистра STATUS (приложение Г);
- ознакомиться с инструкциями, примененными в приведенной ниже программе: CLRF, BSF, BCF, MOVLW, MOVWF, GOTO (приложение Д);
- определить в каких банках находятся регистры PORTC и TRISC (приложение В);
- изучить приведенный ниже образец программы.

### 5.1 Общие сведения

Если в регистре TRISC записаны все нули (регистр очищен), то PORTC работает только на вывод. В этом случае подаваемый на внутренние клеммы PORTC сигнал «1» с ЦПУ, поступит на выходные клеммы, и присоединенное оборудование включится (имитирующие светодиоды начнут светиться).

В любой регистр нельзя непосредственно записать какое - либо число (константу). Сначала константу по инструкции MOVLW записывают в регистр-аккумулятор W, затем по инструкции MOVWF пересылают ее в требуемый регистр.

При включении или сбросе МК выполнение программы начинается с адреса h'00'.

При выборе реального подключаемого оборудования необходимо помнить, что выходное напряжение МК от 3.0 V до 5.5 V постоянного тока.

Задача. Создать программу для ручного управления окрасочной камерой. При подаче питания 220 вольт включается сигнальная лампа на пульте оператора L2. Окрашиваемая деталь поступает в камеру по конвейеру, после чего включается звуковая предупреждающая сигнализация S1. Затем

включается двигатель краскораспылителя D1, вентилятора D2, сигнал на пульт оператора о работе камеры L1. После окрашивания двигатель D1 выключается, после проветривания выключается двигатель вентилятора D2 и лампочка L1. После поступления новой детали процесс окраски повторяется. Указанное оборудование подключается к отдельным битам PORTC (рисунок 5.1). На УМК-7 оборудование имитируется светодиодами.

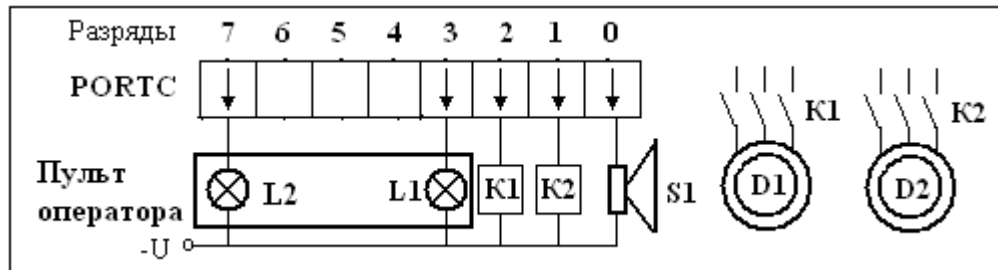


Рисунок 5.1 – Схема подключения оборудования

Программа 5.1 - Ручное управление окрасочной камерой.

`include <p16F877.inc>` ; подключение библиотеки символьных имен РСН  
`org h'00'` ; это указатель для ассемблера, что код следующий за этим выражением, начинается с нулевого адреса ЭППЗУ, т.е. следующая инструкция (por) будет записана по адресу h'00'

`por` ; инструкция, для работы отладчика будет записана по адресу h'00'

`por` ; будет записана по адресу h'01' памяти МК

`por` ; будет записана по адресу h'02' памяти МК

`org h'05'` ; код программы CLRf STATUS будет записан по адресу h'05'

START

*Ниже приведены инструкции для настройки МК.*

`CLRf STATUS` ; очищаем регистр Status, выбираем нулевой банк

`BSF STATUS, 5` ; записав в 5-й разряд '1', выбираем первый банк

`CLRf TRISC` ; обнуляем TRISC и настраиваем PORTC на вывод

`BCF STATUS, 5` ; возвращаемся в нулевой банк

*Рабочая часть программы.*

`MOVLW B'10000000'` ; пересылаем в регистр W число B'10000000'

`MOVWF PORTC` ; включаем сигнализацию L2 на пульт оператора

`Sled_det BSF PORTC, 0` ; включаем предупреждающую сигнализацию S1

`MOVLW B'10001110'` ; пересылаем в регистр W число B'10001110'

`MOVWF PORTC` ; включаем двигатели D1 и D2, сигнализацию L1

`BCF PORTC, 2` ; выключаем двигатель D1 (D2, L1 и L2 работают)

`MOVLW B'10000000'` ;

`MOVWF PORTC` ; выключаем вентилятор D2, сигнализацию L1

`GOTO Sled_det` ; переход на метку Sled\_det для повторения

END ; конец программы.

После точки с запятой в программе напечатаны комментарии, которые пропускаются компилятором.

## 5.2 Порядок выполнения

Выбрать вариант по таблице 5.1 и составить программу.

Таблица 5.1 – Варианты заданий

Вариант	Биты подключения S1, D1, D2, L1, L2.	Вариант	Биты подключения S1, D1, D2, L1, L2.
1	1, 2, 3, 4, 5	6	6, 7, 0, 1, 2
2	2, 3, 4, 5, 6	7	7, 0, 1, 2, 3
3	3, 4, 5, 6, 7	8	0, 1, 2, 3, 4
4	4, 5, 6, 7, 0	9	1, 3, 5, 7, 0
5	5, 6, 7, 0, 1	10	2, 4, 6, 0, 1

Создание проекта описано в приложении Е. После исправления всех ошибок запустить программу на выполнение в пошаговом режиме командами Debug – Run – Step или кнопкой F7. Следить за результатами выполнения инструкций с помощью окна наблюдения.

Остановить выполнение программы командами Run - Reset. Команда Halt просто приостанавливает выполнение программы, и после команд Debug – Run- Step выполнение программы продолжится с места останова.

Включить в отчет схему подключений, программу и результаты. Рекомендуется скопировать текст программы в Блокнот и разместить в папке Мои документы.

Повторно запустить программу в автоматическом режиме, кнопкой F9.

Вывод. Можно управлять выводом информации в PORTC цифровым методом. Выводимая информация может управлять технологическим процессом, включая (выключая) разные механизмы, подключенные к PORTC.

## 5.3 Контрольные вопросы

5.3.1 Нарисуйте схему подключения оборудования к PORTC.

5.3.2 Что выполняет инструкция CLRF STATUS?

5.3.3 Что выполняют инструкции MOVLW B'01010111' и MOVWF PORTC?

5.3.4 Что выполняют инструкции BSF STATUS, 5 и BCF STATUS, 5?

5.3.5 Что выполняет команда GOTO МЕТКА?

5.3.6 Какие внешние устройства расположены на панели стенда?

5.3.7 Почему светятся все светодиоды одновременно в режиме RUN?

5.3.8 В чем разница между адресом регистра и его содержимым?

5.3.9 Перечислите преимущества МК по сравнению с релейной аппаратурой. Назовите устройства, в которых применяются МК.

5.3.10 Назначение символа «;» в тексте программы?

5.3.11 Назначение регистров PORTC и TRISC. В каких банках они находятся?

5.3.12 На какие части делится программа?

## 6 Лабораторная работа №6. Сложение, вычитание. Логика

**Цель:** изучить действие инструкций ADDWF, ANDWF, IORWF, XORWF, SUBWF. Выполнить сложение, вычитание, применить логические функции.

Задание для внеаудиторной подготовки:

Изучить формат инструкций ADDWF, SUBWF, ANDWF, IORWF, XORWF по приложению Д.

### 6.1 Общие сведения

Операции сложения и вычитания применяются, например, для автоматизации учета количества автомашин на объекте, которые могут свободно въезжать и выезжать с территории через ворота.

В таблице 6.1 представлены логические функции, их таблицы истинности, условные обозначения и интерпретация на электрической схеме. Все логические операции выполняются поразрядно.

Таблица 6.1 – Логические функции  $Y=f(X1, X2)$

Инструкция	Таблица истинности			Условное обозначение	Интерпретация на электрической схеме
	X1	X2	Y		
IORWF Функция OR (ИЛИ)	0	0	0		 При замыкании любого контакта в цепи будет ток
	0	1	1		
	1	0	1		
	1	1	1		
ANDWF Функция AND (И)	0	0	0		 Ток в цепи будет только при замыкании обоих контактов
	0	1	0		
	1	0	0		
	1	1	1		
XORWF Функция XOR (Исключительно ИЛИ)	0	0	0		 Ток в цепи будет только тогда, когда переключатели X1 и X2 будут в разном положении
	0	1	1		
	1	0	1		
	1	1	0		



Логическая функция «ИЛИ» («OR») может применяться для включения оборудования любой из двух кнопок, расположенными в разных местах помещения.

Логическая функция «И» («AND») может применяться в том случае, если оборудование должно включаться, при срабатывании (включении) двух контактов. Например, освещение в помещении автоматически включается при низкой освещенности и наличия в нем людей.

Логическая функция «Исключительно ИЛИ» («XOR») применяется, например, для управления освещением длинного тоннеля. С любого конца тоннеля можно включить или выключить освещение.

Изучить программу 6.1, которая выполняет различные действия с двумя константами и выводит результаты работы в регистр REZ.

Программа 6.1 – Сложение и вычитание данных.

```
include <p16F877.inc>
```

```
R1 EQU h'21' ; присваиваем регистру по адресу h'21' имя R1
```

```
R2 EQU h'22' ; присваиваем регистру по адресу h'22' имя R2
```

```
R3 EQU h'23' ; R3 – имя регистра по адресу h'23'
```

```
REZ EQU h'24' ; REZ - имя регистра по адресу h'24' для записи результатов
```

```
org h'00' ; инструкция (nop) будет записана по адресу h'00'
```

```
nop ; пустая инструкция будет записана по адресу h'00'
```

```
nop ; пустая инструкция будет записана по адресу h'01'
```

```
nop
```

```
org h'05' ; следующая инструкция будет записана по адресу h'05'
```

START

```
CLRF STATUS ; очищаем регистр STATUS и выбираем нулевой банк
```

```
BSF STATUS, 5; переходим в первый банк
```

```
CLRF TRISC ; настраиваем биты PORTC на выход
```

```
BCF STATUS, 5; возвращаемся в нулевой банк
```

```
CLRF REZ ; записать в регистр REZ нули
```

```
MOVLW D'5' ; запись константы в W
```

```
MOVWF R3 ; запись содержимого W в регистр R3
```

```
MOVLW D'130'; запись константы в W
```

```
MOVWF R2 ; запись содержимого W в регистр R2
```

```
MOVLW D'240'; запись константы в W
```

```
MOVWF R1 ; запись содержимого W в регистр R1
```

```
MOVF R1, W ; копирование константы из R1 в регистр W
```

```
ADDWF R2, W ; сложение W=R2+ W=R2+R1
```

```
MOVWF REZ ; загрузить содержимое W в регистр REZ
```

```
MOVF R1, W ; копирование константы из R1 в регистр W
```

```
SUBWF R2, W ; вычитание W=R2-W=R2- R1
```

```
MOVWF REZ ; переслать содержимое W в регистр REZ
```

```
MOVF R2, W ; копирование константы из R2 в регистр W
```

```
SUBWF R1, W ; W=R1-W=R1-R2
```

```

MOVWF REZ ;
MOVF R2, W ;
SUBWF R1, W ; вычитание W=R1-W=R1-R2. Результат оставляем в W
MOVWF REZ ; переслать содержимое W в регистр REZ
MOVF R2, W ; запись константы в регистр W из R2
ANDWF R1, W ; W=R1 AND W= R1 AND R2. Результат заносим в W
MOVWF REZ ; копируем аккумулятор в регистр REZ.
END

```

## 6.2 Порядок выполнения

Ознакомиться с предложенной выше программой и самостоятельно продолжить программу командами для выполнения операций R2-R2, R2+R3, R1 OR R2, R1 XOR R2.

Выбрать вариант по таблице 6.2 и составить программу.

Таблица 6.2 – Варианты задания

Вариант	Числа R1, R2, R3	Вариант	Числа R1, R2, R3
1	81h, 12h, 1h	6	86h, 67h, 6h
2	82h, 23h, 2h	7	87h, 68h, 7h
3	83h, 34h, 3h	8	88h, 69h, 8h
4	84h, 45h, 4h	9	89h, 6Ah, 9h
5	85h, 56h, 5h	10	8Ah, 6Bh, Ah

Создать окно наблюдения для всех применяемых в программе регистров в необходимом формате чисел. Результаты работы всех операций с числами записать в таблицу 6.3.

Таблица 6.3 - Пример записи результатов работы команд

Числа на входе		Действие	Результат в REZ		Значение в Status
Форма В или D	Форма В или D		Форма В	Форма D	Форма В
R1=D'240'	R2=D'130'	R1+R2	-	114	00011001
R1=D'240'	R2=D'130'	R1-R2	-	110	00011001
R1=D'240'	R2=D'130'	R2-R1	-	146	00011010
R2=D'130'	R2=D'130'	R2-R2	-	0	00011111
R2=D'130'	R3=D'5'	R2+R3	-	135	00011000
R1=b'11110000'	R2=b'10000010'	R1 AND R2	10000000	-	00011011
R1=b'11110000'	R2=b'10000010'	R1 OR R2	11110010	-	00011011
R1=b'11110000'	R2=b'10000010'	R1 XOR R2	01110010	-	00011011

### 6.3 Контрольные вопросы

- 6.3.1 Как определить с регистрами какого банка работает программа?
- 6.3.2 Что выполняют инструкции:  
ADDWF R1, f; SUBWF R2, f; ?
- 6.3.3 Что выполняют инструкции:  
ANDWF R1, f; IORWF R1, W; XORWF R2, W; ?
- 6.3.4 Какое назначение директивы ORG h'05'?
- 6.3.5 Приведите таблицы истинности логических операций.
- 6.3.6 В каком банке находятся регистры R1, R2 и REZ?
- 6.3.7 Какой результат мы увидим при сложении 230+60 в МК?
- 6.3.8 Какой результат мы увидим при вычитании 130-135 в МК?

## 7 Лабораторная работа №7. Анализ информации и автоматизация

**Цель:** научиться использовать флаги регистра STATUS для анализа информации. Изучить инструкции условных переходов BTFSC и BTFSS, применить их для ветвления алгоритма автоматизации.

### 7.1 Общие сведения

Инструкции условных переходов BTFSC и BTFSS проверяют состояние заданного бита в любом регистре и, в зависимости от результата, пропускают или не пропускают следующую инструкцию программы. Инструкция BTFSC пропускает следующую инструкцию, если заданный бит сброшен. Инструкция BTFSS пропускает следующую инструкцию, если заданный бит установлен. На основе этого создается передача управления в другое место программы и организация циклов (таблица 7.1).

Таблица 7.1 – Ветвление алгоритма

Использование бита CARRY (C)	
BTFSS STATUS, C GOTO METKA MOVWF R2	Если в бите Carry установлена единица, то пропускается следующая инструкция программы GOTO METKA и сразу выполняется инструкция MOVWF R2
	Если бит Carry сброшен, то выполняется следующая инструкция программы GOTO METKA
Использование бита ZERO (Z)	
BTFSS STATUS, Z MOVF R2, W MOVWF PORTC	Если в бите Z установлена единица, то пропускается следующая инструкция программы MOVF R2 и сразу выполняется инструкция MOVWF PORTC
	Если бит Z сброшен, то выполняется следующая инструкция программы MOVF R2, W и затем MOVWF PORTC

В реальных программах вместо указанных инструкций пишутся инструкции, соответствующие разработанному алгоритму.

В ассемблере отсутствуют инструкции сравнения чисел, поэтому, чтобы определить, какое из чисел больше (меньше), применяют следующий алгоритм:

1) Выполняют вычитание чисел. При этом в зависимости от результата, устанавливаются флаги C и Z регистра STATUS (таблица 7.2).

2) Далее применяют инструкции условных переходов.

В таблице 7.2 показаны значения флагов регистра STATUS при вычитании по команде SUBWF UMENSH, W, т.е.  $W = UMENSH - W$ .

Таблица 7.2 – Регистрация событий в регистре STATUS

Соотношения значений в W и POH UMENSH	Значения флагов регистра STATUS после выполнения вычитания	
	Флаг Z	Флаг C
$UMENSH - W > 0$	0 - Нулевого результата не было	1 - был перенос из 7 бита W в бит C (при сложении в дополнительном коде)
$UMENSH - W < 0$	0 - Нулевого результата не было	0 - не было переноса из 7 разряда W в бит C (при сложении в дополнительном коде)
$UMENSH - W = 0$	1 - Был нулевой результат операции	1 - был перенос из 7 разряда W в бит C (при сложении в дополнительном коде)
Примечание – В регистр UMENSH помещается уменьшаемое число		

Составим программу для автоматизации теплового режима.

Технологический процесс должен выполняться при определенной требуемой температуре  $T_{treb}$ . Температура окружающей среды  $T_{sredy}$  измеряется аналоговым датчиком. Его показания поступают на АЦП микроконтроллера и преобразуются в двоичный код. Измеренная температура среды сравнивается с требуемой температурой, если она выше - включают охлаждение, если ниже - включают подогрев. Охлаждающая установка подключается к нулевому биту PORTC, нагревающая установка подключается к первому биту. Работа такой установки аналогична на работе кондиционера «Зима-Лето».

Программа 7.1- Регулирование температуры

include<p16f877.inc>

T\_sredy EQU h'40' ; регистр для записи температуры окружающей среды

T\_treb EQU h'41' ; регистр для записи требуемой температуры установки

UMENSH EQU h'43' ; рабочий регистр для хранения данных.

*Пропущенные инструкции настройки МК напишите самостоятельно.*

```

        MOVLW D'22'          ;
        MOVWF T_treb         ; требуемая температура
        MOVLW b'00011000'    ; первое значение T_sredy, данные после АЦП
M1      MOVWF T_sredy        ; температура по показаниям датчика
        CLRF PORTC
        MOVF T_treb, W       ; устанавливаем UMENSH = T_treb
        MOVWF UMENSH        ; чтобы не портить данные в регистре T_treb
        MOVF T_sredy, W      ; температура --> W
        SUBWF UMENSH, F      ; UMENSH=T_Treb -T_Sredy
        MOVLW b'00000010'    ; предполагаем, что надо включить подогрев
        (Запомните, что команда MOVLW не влияет на регистр Status)
        BTFSS STATUS, C      ; C = 1? (разность положительна?)
        MOVLW b'00000001'    ; (только при C=0) надо включить охлаждение
        MOVWF PORTC         ; пересылаем информацию на установку из W
        MOVLW b'00010100'    ; второе значение T_sredy.
        GOTO M1;
        END

```

На рисунках 7.1 и 7.2 показаны окна наблюдения, из которых видно, как переключается нагрев и охлаждение при изменении окружающей температуры.

Address	Symbol	Value
03	STATUS	B'00011011'
200	W	B'00000010'
87	TRISC	B'00000000'
07	PORTC	B'00000010'
40	T_sredy	D'20 '
41	T_treb	D'22 '
43	UMENSH	D'2 '

Рисунок 7.1 – Нагрев включился

Address	Symbol	Value
03	STATUS	B'00011000'
200	W	B'00000001'
87	TRISC	B'00000000'
07	PORTC	B'00000001'
40	T_sredy	D'24 '
41	T_treb	D'22 '
43	UMENSH	D'254'

Рисунок 7.2 – Охлаждение включилось

## 7.2 Порядок выполнения

Выбрать вариант по таблице 7.3 и составить программу. При написании программы используйте блок-схему на рисунке 7.3.

Таблица 7.3 – Варианты заданий

Вариант	T_sredy 1	T_treb	T_sredy 2	Бит подключения системы нагрева	Бит подключения системы охлаждения
1	00001010	11	00001101	3	1
2	00010101	22	00011001	4	2
3	00011110	33	00100011	5	3
4	00101001	44	00101111	6	4
5	00110101	55	00111001	7	5
6	00111110	66	01000101	3	0

7	01001011	77	01001111	4	1
8	01010100	88	01011001	5	2
9	01011111	99	11010000	6	3
10	01100000	110	01101111	7	4

Примечание - Значения T\_sredy 1 и T\_sredy 2 заданы в двоичной системе, T\_treb в десятичной.

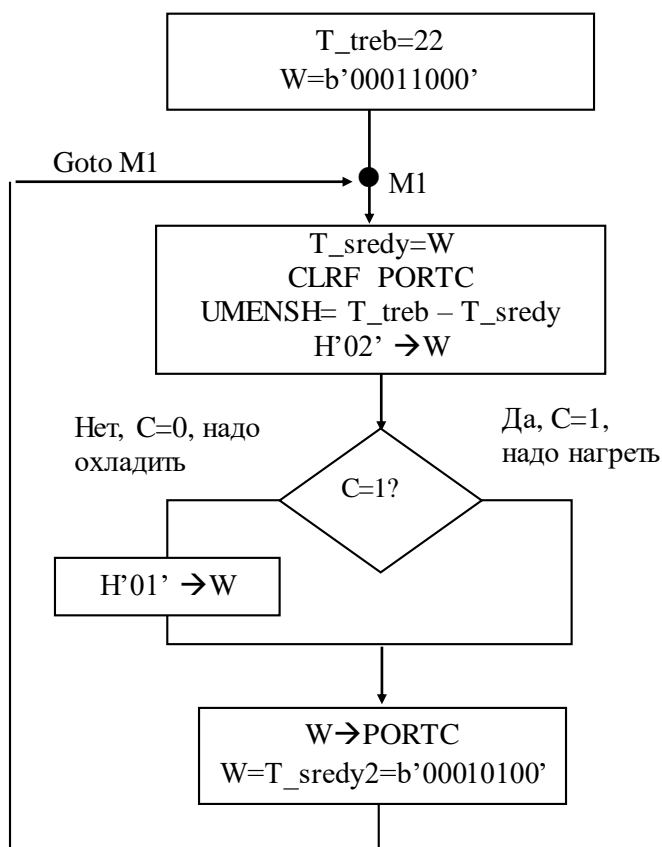


Рисунок 7.3 – Блок-схема (фрагмент)

### 7.3 Контрольные вопросы

7.3.1 Что происходит с регистром STATUS при логических и математических операциях?

7.3.2 Расскажите, какую реальную задачу решает ваша программа.

7.3.3 Поясните инструкцию условного перехода BTFSC STATUS, Z.

7.3.4 Поясните инструкцию условного перехода BTFSS STATUS, Z.

7.3.5 Какой операцией можно определить равенство чисел?

7.3.6 В каких случаях флаг Z равен единице при вычитании?

7.3.7 В каких случаях флаг C равен единице при вычитании?

7.3.8 Когда флаги Z и C регистра Status одновременно равны нулю?

7.3.9 Как определить более мощную установку из двух потребителей?

7.3.10 Как выполняется сравнение чисел?

7.3.11 В каких банках находятся, регистры применяемые в программе?

7.3.12 Как переключается нагрев и охлаждение установки?

## 8 Лабораторная работа №8. Подпрограммы

**Цель:** освоить навыки составления подпрограмм для задержки времени.

### 8.1 Общие сведения

Подпрограммы применяют в тех случаях, когда в нескольких местах алгоритма требуется выполнить одинаковые действия. Отсчет времени можно организовать с помощью подпрограмм со счетчиками, что и реализовано в алгоритме, показанном на рисунке 8.1.

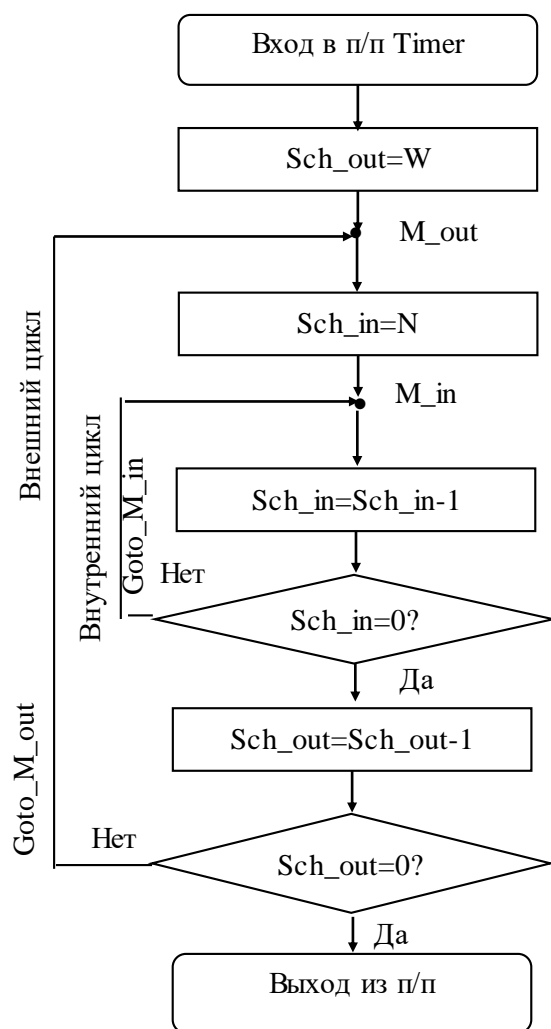


Рисунок 8.1 – Блок-схема

Часто программу разбивают на подпрограммы, чтобы упростить её чтение.

Рассмотрим сначала работу внутреннего цикла подпрограммы Timer. Сначала счетчику внутреннего цикла Sch\_in задается какое-то значение N. Затем значение счетчика Sch\_in уменьшается на 1 и проверяется: не стало ли оно нулевым. Если значение счетчика не равно нулю, то выполняется

инструкция перехода GOTO M\_in, повторяющая цикл. Если счетчик обнулился, программа выходит из цикла. Длительность задержки времени определяется начальной величиной счетчика Sch\_in и временем выполнения инструкций в МК. Поскольку микропроцессор работает с высокой частотой, приходится создавать внешний цикл со счетчиком Sch\_out для увеличения времени задержки. Аналогично, сначала задается значение Sch\_out=W. На каждый цикл уменьшения Sch\_out на единицу, внутренний цикл сработает N раз. Время задержки будет определяться произведением Sch\_out\*Sch\_in и частотой работы генератора МК.

Подпрограммы пишут после инструкции GOTO\$ основной программы. Подпрограмма всегда начинается с имени, а заканчивается - инструкцией возврата в основную программу RETURN. Подпрограмма из основной программы вызывается по имени инструкцией CALL ИМЯ, например, CALL Timer. Значение аккумулятора W является аргументом для подпрограммы Timer. После завершения ее работы управление передается в главную программу на следующую инструкцию после инструкции CALL Timer.

Задание на проектирование. Создать программу для автоматической окрасочной камеры, описанной в лабораторной работе 5, выполняющую следующую работу:

- при подаче питания включить световую сигнализацию на пульт оператора L2 и звуковую предупреждающую сигнализацию S1 на 2 секунды;
- включить двигатель краскораспылителя D1 на 3 секунды, а также двигатель вентилятора D2 и лампочку L1;
- выключить краскораспылитель D1, вентилятор D2 и лампочка L1 должны работать еще 2 секунды;
- выключить D2 и L1 на 1 секунду для замены заготовки в камере;
- повторить процесс окрашивания новой заготовки.

Временная диаграмма работы камеры показана на рисунке 8.2.

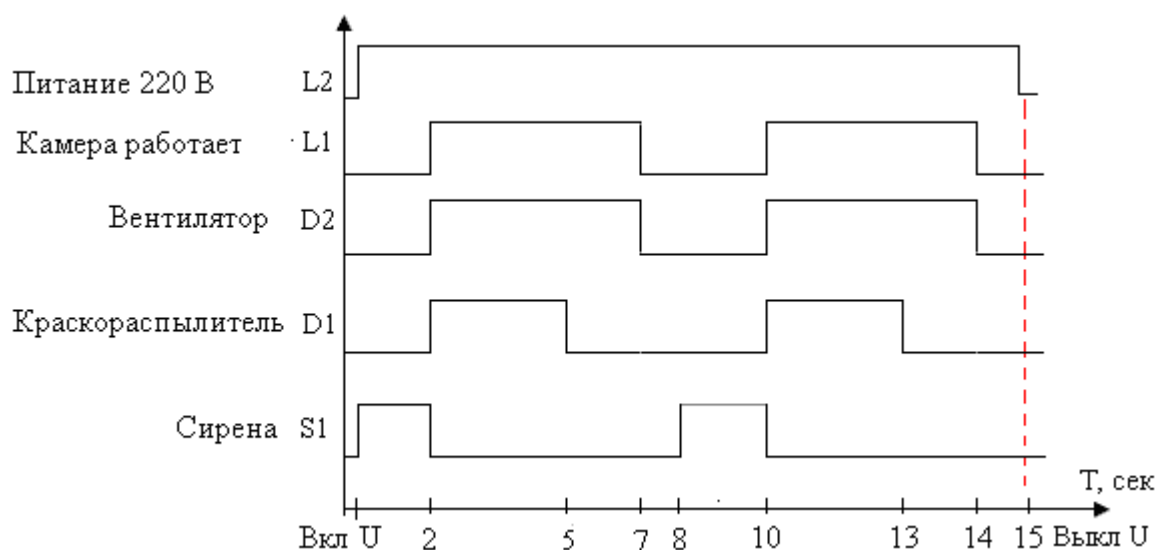


Рисунок 8.2 – Временная диаграмма работы окрасочной камеры



```

Программа 8.1
include<p16F877.inc>
Sch_in    EQU  H'22' ; счетчик внутреннего цикла задержки
Sch_out    EQU  H'24' ; счетчик внешнего цикла задержки
Sch3      EQU  H'23' ; счетчик третьего цикла задержки
                ; инструкции настройки МК опущены
                ; рабочая часть программы

    MOVLW  B'10000000'
    MOVWF  PORTC      ; включаем сигнализацию L2 на пульт оператора
Sled_det BSF  PORTC, 0 ; включаем звуковую сигнализацию S1
    MOVLW  d'40'      ; аргумент W задает время таймера. При отладке d'2'
    CALL  Timer
    MOVLW  B'10001110'
    MOVWF  PORTC      ; выключаем S1, включаем D1, D2 и L1
    MOVLW  d'60'      ; задаем время окраски
    CALL  Timer
    BCF  PORTC, 2      ; выключаем D1, продолжают работать D2, L1 и L2
    MOVLW  d'40'      ; задаем время работы D2 и L1
    CALL  Timer
    MOVLW  B'10000000'
    MOVWF  PORTC      ; окраска закончена, сигнализация на пульте L2 работает
    MOVLW  d'20'      ; задаем время для замены заготовки
    CALL  Timer
    GOTO  Sled_det      ; на метку Sled_det для нового включения камеры
    GOTO  $              ; разделитель основной программы и подпрограмм
Timer      ; Подпрограмма Timer. Применено два вложенных цикла.
    MOVWF  Sch3          ; значение W является аргументом для таймера
M3  MOVLW  D'150'
    MOVWF  Sch_out      ; устанавливаем значение внешнего счетчика
M_out      ; метка внешнего счетчика
    MOVLW  D'255'      ;
    MOVWF  Sch_in      ; устанавливаем значение внутреннего счетчика
M_in      ; метка внутреннего счетчика
    DECF  Sch_in, F      ; уменьшаем значение счетчика Sch_in на 1
    BTFSS STATUS, Z      ; если оно равно нулю (Z=1), пропускаем GOTO
    GOTO  M_in          ; срабатывает только при Z=0
    DECF  Sch_out, F     ; уменьшаем значение счетчика Sch_out на 1,
    BTFSS STATUS, Z      ; если оно равно нулю (Z=1), пропускаем GOTO
    GOTO  M_out         ; срабатывает только при Z=0
    DECF  Sch3, F        ; уменьшаем значение счетчика Sch3 на 1,
    BTFSS STATUS, Z      ; если оно равно нулю (Z=1), пропускаем GOTO
    GOTO  M3            ; срабатывает только при Z=0
    RETURN              ; конец подпрограммы Timer
    END                ; конец текста всей программы.

```

## 8.2 Порядок выполнения

Разработайте программу, включающую и выключающую оборудование на заданное время, согласно варианту задания из таблицы 8.1.

Отладка программы. Оставьте только первый вызов подпрограммы Timer, установив аргумент W=2. Остальные вызовы выключите, напечатав перед инструкцией Call символ «;». Установите значения для счетчиков Sch\_in и Sch\_out равные 2. Введите разработанную программу в микроконтроллер. Создайте окно наблюдения. После отладки программы в пошаговом режиме, установите первоначальные значения аргумента W, счетчиков Sch\_in и Sch\_out и запустите ее в автоматическом режиме. Рассчитайте W для заданного времени работы оборудования по варианту методом пропорций. Для уточнения времени задержки можно добавить в циклы дополнительные бесполезные инструкции, например, пор.

Таблица 8.1 – Варианты заданий

Вариант	Биты подключения S1, D1, D2 L1, L2	Время работы, сек			Вариант	Биты подключения S1, D1, D2 L1, L2	Время работы, сек		
		S1	D1	D2			S1	D1	D2
1	1, 2, 3, 4, 5	5	6	3	6	6, 7, 0, 1, 2	6	6	4
2	2, 3, 4, 5, 6	6	7	4	7	7, 0, 1, 2, 3	7	7	5
3	3, 4, 5, 6, 7	7	6	5	8	0, 1, 2, 3, 4	8	6	6
4	4, 5, 6, 7, 0	8	5	6	9	1, 3, 5, 7, 0	9	5	7
5	5, 6, 7, 0, 1	9	7	7	10	2, 4, 6, 0, 1	10	6	3

Результаты занесите в таблицу 8.2. Продемонстрируйте работу программы преподавателю.

Таблица 8.2– Результаты настройки циклов задержки

Вариант	Оборудование	Параметры подпрограммы Timer			
		Время работы, с	Значение счетчиков		
			Sch_in	Sch_out	Sch3
	S1				
	D1, D2, L1				
	D2, L1				

## 8.3 Контрольные вопросы

8.3.1 Почему создан третий цикл в подпрограмме Timer?

8.3.2 Почему в счетчики циклов записаны указанные значения?

8.3.3 Назначение подпрограмм.

8.3.4 Цикл, вложенный цикл.

8.3.5 Почему в Timer выбраны указанные значения Sch\_in и Sch\_out?

8.3.6 Что такое машинный цикл? За сколько машинных циклов выполняются команды в Timer?

8.3.7 Как вызывается подпрограмма на выполнение?

8.3.8 Опишите работу инструкции DECF Sch\_out, F.

8.3.9 Как разделяются главная программа и подпрограммы?

8.3.10 Какими операторами начинается и заканчивается подпрограмма?

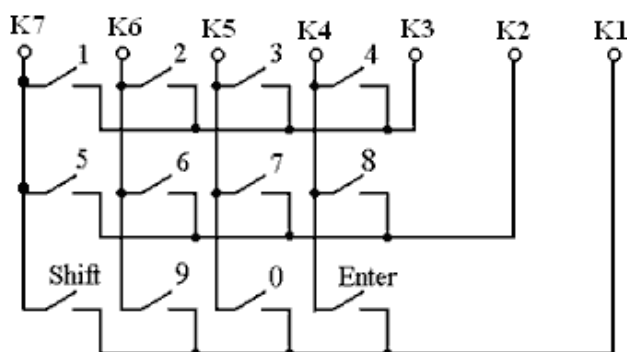
8.3.11 Почему в Timer применено два вложенных цикла?

## 9 Лабораторная работа №9. Подпрограммы

**Цель:** освоить навыки составления программ с вводом данных клавиатурой.

### 9.1 Общие сведения

Клавиатура используется в реальных программах для ввода данных, инструкций, изменения алгоритма вычислений. В лабораторной работе с клавиатуры будем подавать сигнал «1» в PORTD, без применения дополнительного источника питания. Сигнал будет обработан микропроцессором и в PORTC будет выведен результат работы программы.



Роль кнопок клавиатуры в программе и их расположение.

«5» - Кнопка Pusk1 на пульте;

«6» - Кнопка Pusk2 по месту;

«Shift»- Кнопка Stop1 на пульте;

«9» - Кнопка Stop2 по месту.

К биту 1 PORTD подключена кнопка K1, к биту 7 кнопка K7.

Рисунок 9.1 – Схема клавиатуры

Изучите принципиальную схему пленочной клавиатуры (рисунок 9.1), изображенную на лицевой панели стенда. Такая схема продиктована тем, что все кнопки подключаются к одному регистру, имеющему восемь бит.

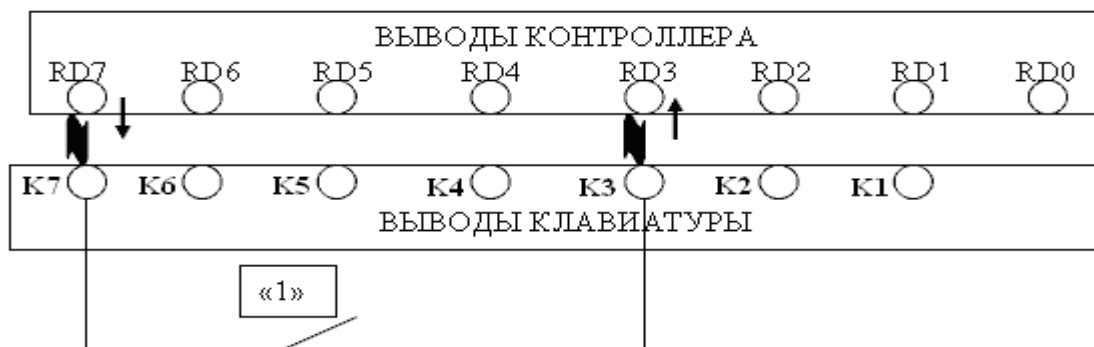


Рисунок 9.2 – Монтажная схема подключения кнопки «1»

На рисунке 9.2 приведена, в качестве примера, монтажная схема подключения кнопки с цифрой «1» к PORTD. Разряды RD7, RD6, RD5, RD4 предназначены для подачи сигнала «1». Остальные для приема сигнала. В разряде RD0 по заводской схеме всегда записана «1». Одна клемма кнопки «1» подключена к выводу RD7, а вторая клемма кнопки подключена к выводу RD3. Бит RD7 должен быть запрограммирован на вывод, а бит RD3 - на ввод. Когда кнопка нажата, с бита RD7 проходит сигнал через кнопку «1» на клемму RD3.

Задание на проектирование. Создать программу автоматизации окрасочной камеры по заданию лабораторной работы «Подпрограммы», изменив её. Предусмотреть запуск и останов камеры кнопками по месту и с пульта оператора.

Дребезг контактов клавиатуры убирается включением подпрограммы Timer. Кнопка Stop в данной программе находится в подпрограмме Timer. Выключение оборудования кнопкой здесь возможно, потому что при выполнении данной программы, МК практически все время находится во внутреннем цикле подпрограммы Timer. При других алгоритмах такой вариант выключения невозможен, необходимо предусмотреть прерывание.

Фрагмент 9.1 - Программирование клавиатуры

Добавленные команды по сравнению с лабораторной работой 8, выделены курсивом.

*Пропущенные инструкции настройки МК напишите самостоятельно.*

CLRF TRISC

MOVLW b'00001111'

MOVWF TRISD

; настраиваем PORTD: биты 0-3 на ввод, 4-7 на вывод

BCF STATUS, 5

CLRF PORTC

*Рабочая часть программы.*

MOVLW B'11110000'

MOVWF PORTD; подаем питание на выводы клавиатуры

MOVLW B'10000000'

MOVWF PORTC

; включаем сигнализацию наличия питания на пульт

Sled\_det BTFSS PORTD, 2 ; проверяем нажатие кнопки Push

GOTO Sled\_det

BSF PORTC, 0

; включаем звуковую предупреждающую сигнализацию

MOVLW d'40'

; задаем время работы сигнализации S1

CALL Timer

MOVLW B'10001110'

MOVWF PORTC

; включаем двигатели D1 и D2, сигнализацию L1

MOVLW d'60'

; задаем время работы D1

CALL Timer

BCF PORTC, 2

; выключаем D1, продолжают работать D2, L1 и L2

MOVLW d'20'

; задаем время работы вентилятора и L1

CALL Timer

```

    MOVLW B'10000000'
    MOVWF PORTC           ; выключаем вентилятор D2 и сигнализацию L1
    MOVLW d'20'           ; задаем время для замены заготовки
    CALL Timer
    GOTO Sled_det          ; на метку Sled_det для нового запуска установки
    GOTO $                 ; разделитель основной программы и подпрограмм
Timer                      ; в подпрограмме показан только внутренний цикл с кнопками Stop
M_in                      ; метка внутреннего счетчика
    BTFSC PORTD, 1        ; проверяем нажатие кнопки Stop1 или Stop2
    GOTO Stop             ; если нажата, переход к инструкциям останова камеры
    DECF Sch_in, F         ; уменьшаем значение счетчика Sch_in на 1
    BTFSS STATUS, Z        ; если оно равно нулю (Z=1), пропускаем GOTO
    GOTO M_in              ; срабатывает только при Z=0
    . . . .
    RETURN
Stop CLRF PORTC           ; окончание работы окрасочной камеры
    END;                  ; Конец текста всей программы.

```

## 9.2 Порядок выполнения

Разработайте программу, автоматизации окрасочной камеры, согласно варианту задания из таблицы 9.1.

Таблица 9.1 – Варианты заданий

Вариант	Биты подключения S1, D1, D2 L1, L2	Кнопка Pusk	Кнопка Stop	Вариант	Биты подключения S1, D1, D2 L1, L2	Кнопка Pusk	Кнопка Stop
1	1, 2, 3, 4, 5	1 или 2	5 или 6	6	6, 7, 0, 1, 2	6 или 7	9 или 0
2	2, 3, 4, 5, 6	2 или 3	6 или 7	7	7, 0, 1, 2, 3	7 или 8	3 или 4
3	3, 4, 5, 6, 7	3 или 4	7 или 8	8	0, 1, 2, 3, 4	8 или 6	4 или 2
4	4, 5, 6, 7, 0	4 или 2	6 или 8	9	1, 3, 5, 7, 0	9 или 0	2 или 3
5	5, 6, 7, 0, 1	5 или 6	1 или 2	10	2, 4, 6, 0, 1	0 или 9	7 или 6

Время работы S1, D1, D2, L1 и L2 и биты их подключения к PORTC принять как в лабораторной работе 8.

Продемонстрируйте работу программы преподавателю.

## 9.3 Контрольные вопросы

9.3.1 Каково назначение клавиатуры?

9.3.2 Зачем разряды PORTD настраивают на ввод и на вывод?

9.3.3 Почему каждая кнопка клавиатуры не имеет отдельной клеммы?

9.3.4 Как проверяется нажатие кнопки клавиатуры?

- 9.3.5 Что выполняет инструкция BTFSC PORTD, 2?
- 9.3.6 Что выполняет инструкция BTFSS PORTD, 2?
- 9.3.7 Назначение сигнализации в программе.
- 9.3.8 Что записано в файле p16F877.inc?
- 9.3.9 Можно ли кнопку Stop разместить вне подпрограммы Timer?

## **10 Лабораторная работа №10. Управление двигателем постоянного тока**

**Цель:** изучить схемы включения драйвера управления ДПТ и получить навыки программирования драйвера ДПТ; приобрести навыки управления частотой вращения ДПТ; приобрести навыки реверсивного управления ДПТ с использованием ШИМ.

### **10.1 Общие сведения**

Изучить мостовую схему включения ДПТ (рисунок 10.1).

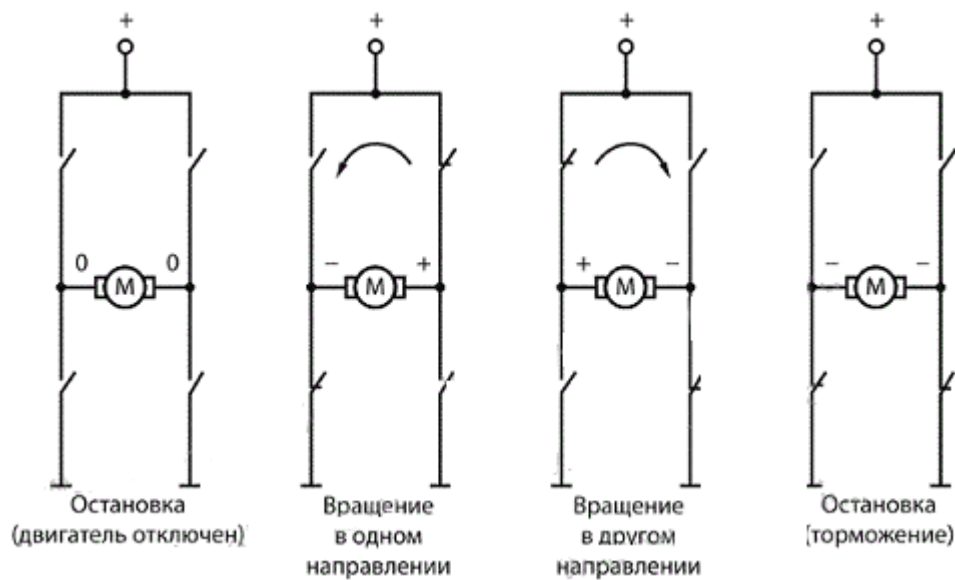
Полная мостовая схема (H-образный мост) содержит четыре выключателя, соединенных последовательно-параллельно. Широко распространен электронный вариант моста, где обычно используются транзисторы, работающие в режиме переключения. Такая схема часто служит для управления двигателем постоянного тока и позволяет изменять скорость и направление вращения.

Схема, приведенная ниже (рисунок 10.1, а), иллюстрирует управление двигателем, который можно привести в одно из четырех различных состояний: вращения в одном или в другом направлении, отключения и принудительной остановки (торможения). Последний вариант осуществляется путем одновременного замыкания двух нижних выключателей. В результате происходит закорачивание обмотки двигателя. Схема часто используется для управления двигателями в радиоуправляемых моделях. Последовательность сигналов должна быть достаточно точной: нужно избежать одновременного замыкания двух переключателей в одной ветви, что привело бы к закорачиванию источника питания. Чтобы выполнить это условие, для формирования управляющих сигналов обычно применяется специальное устройство. Путем периодического прерывания тока в ветвях моста можно изменять среднее значение тока, протекающего через двигатель, а, следовательно, и скорость его вращения.

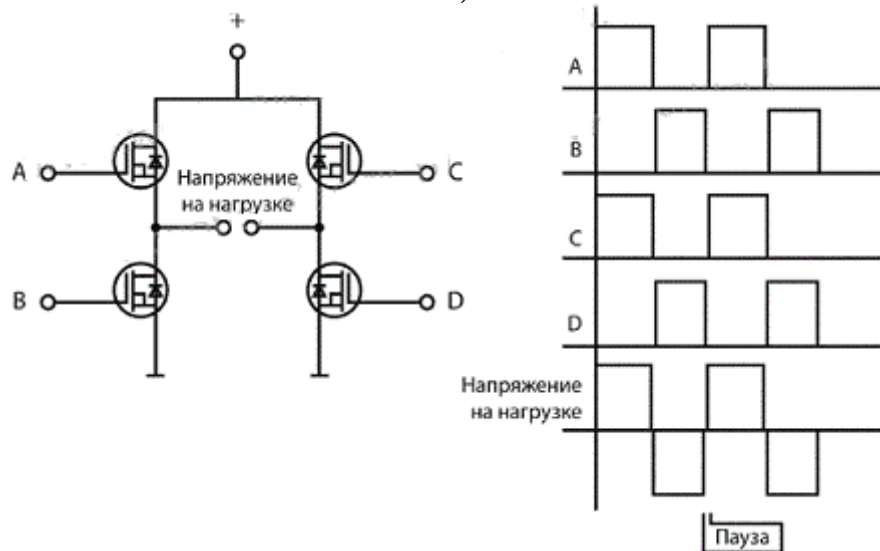
### **10.2 Порядок выполнения**

Лабораторная работа выполняется на стенде, описанном в лабораторной работе 4.

- 10.2.1 Для программирования драйвера управления ДПТ необходимо:
  - выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 10.2;



а)



б)

Рисунок 10.1 – Мостовая схема включения ДПТ

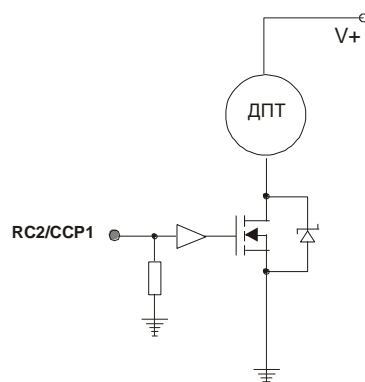


Рисунок 10.2 – Схема соединения ДПТ

- включить питание стенда, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;

- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «DCM\_test1.asm», пример которой представлен в Приложении Ж;

- проверить результат ее выполнения: двигатель должен включаться и отключаться через равные промежутки времени (примерно 10 с);

- проанализировать подпрограмму задержки «Delay» - изменить программу «DCM\_test1.asm» так, чтобы время работы двигателя составило 5 с, а время отключения 10 с (Тактовая частота МК=10 MHz). Новую программу сохранить под именем «DCM\_test2.asm». Проверить результат ее выполнения.

10.2.2 Для управления частотой вращения ДПТ с использованием широтно импульсного модулятора необходимо:

- выключить питание стенда. Выполнить соединение ДПТ по схеме на рисунке 10.3, а;

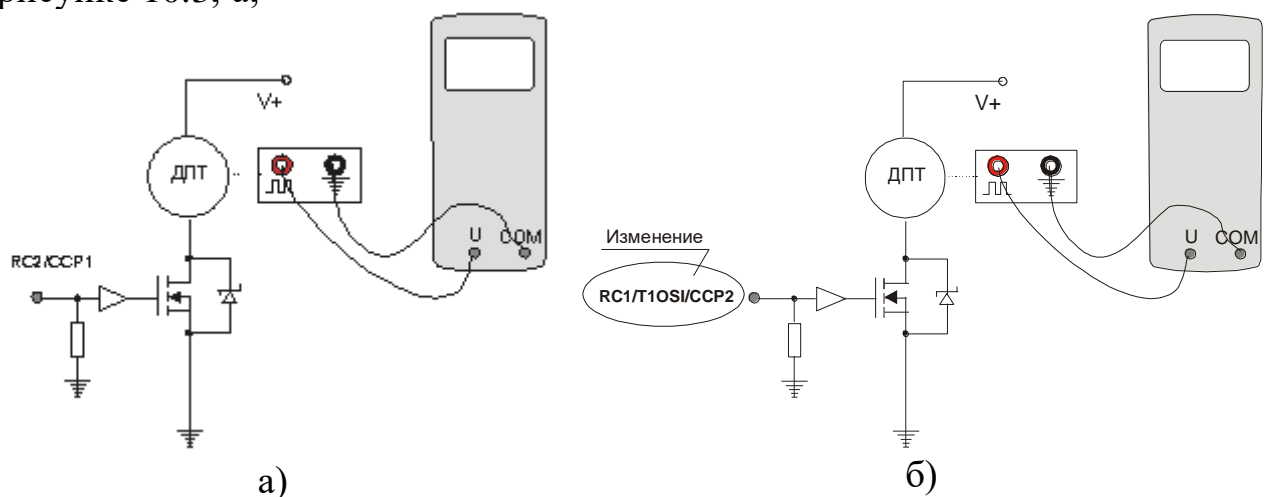


Рисунок 10.3 – Схемы соединения ДПТ

- включить питание стенда, установить УИП в режим измерений частоты, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;

- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «DCM\_test3.asm»;

- проверить результат ее выполнения: двигатель должен наращивать частоту вращения через равные промежутки времени и, дойдя до максимального значения, отключаться и потом снова повторять весь цикл;

- не останавливая двигатель, снять показания частоты вращения с УИП после каждого приращения. Построить график зависимости  $F$  (частота вращения ДПТ) от  $T_{и}$  (время импульса ШИМ – предварительно рассчитать, ( $F_{osc}=10\text{ MHz}$ , коэффициент делителя  $TMR2=16$ ));

- проанализировать и изменить программу «DCM\_test3.asm» так, чтобы цикл приращения частоты двигателя состоял из 8 ступеней. Новую программу сохранить под именем «DCM\_test4.asm». Проверить результат ее выполнения.



Построить график зависимости  $F$  от  $T$  для этой программы ( $F_{osc}=10\text{ MHz}$ , коэффициент делителя  $TMR2=16$ );

- выключить питание стенда, изменить схему соединений ДПТ в соответствии с рисунком 10.3, б;

- проанализировать и изменить программу «DCM\_test3.asm» так, чтобы для регулирования частоты вращения двигателя использовался модуль CCP2. Новую программу сохранить под именем «DCM\_test3\_1.asm». Проверить результат ее выполнения.

10.2.3 Для реверсивного управления ДПТ необходимо:

- выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 10.4;

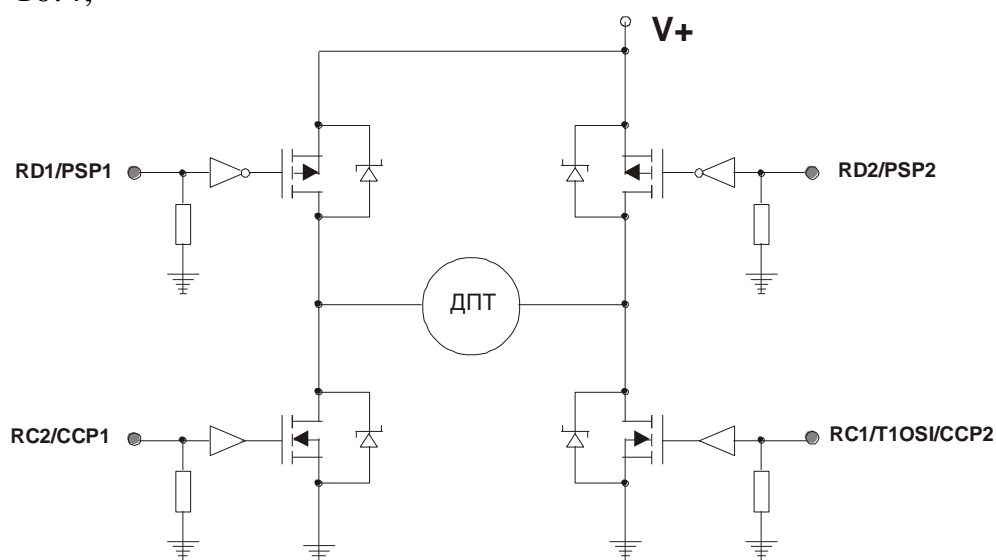


Рисунок 10.4 – Схема соединения ДПТ

- включить питание стенда, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;

- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «DCM\_test5.asm»;

- проверить результат ее выполнения: двигатель должен раскручиваться в одну сторону, потом медленно останавливаться и раскручиваться в другую сторону, повторяя этот цикл через равные промежутки времени;

- проанализировать и изменить программу «DCM\_test5.asm» так, чтобы остановка двигателя производилась режимом торможения. Новую программу сохранить под именем «DCM\_test6.asm». Проверить результат ее выполнения.

10.2.4 Для реверсивного управления ДПТ с использованием ШИМ необходимо:

- выключить питание стенда, выполнить соединение ДПТ по схеме на рисунке 10.5;

- включить питание стенда, установить УИП в режим измерений частоты, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;

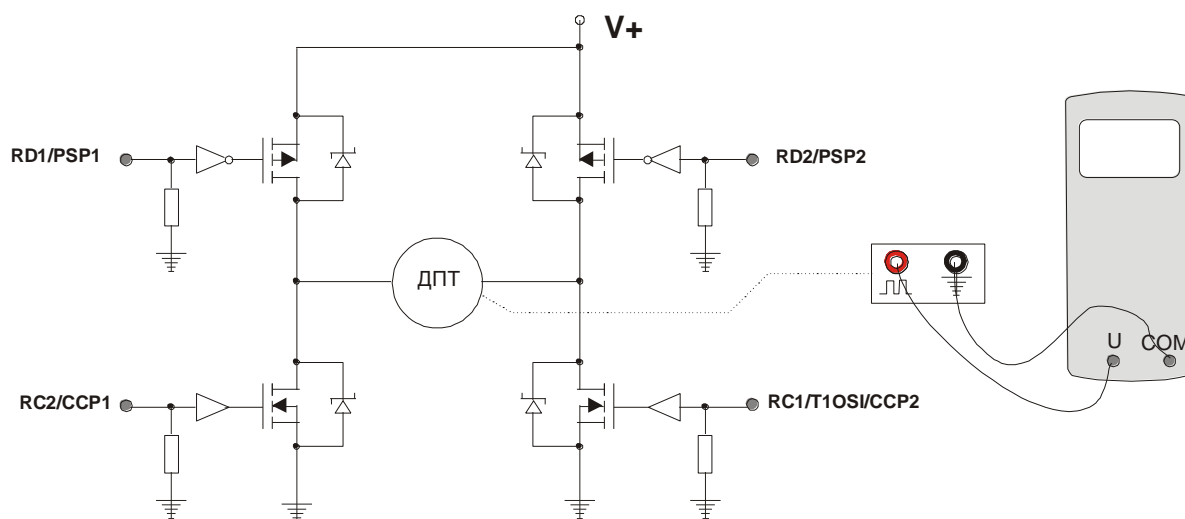


Рисунок 10.5 – Схема соединения ДПТ

- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «DCM\_test7.asm»;
- проверить результат ее выполнения: при движении в одну сторону, двигатель должен иметь частоту вращения в 2 раза меньше, чем в другую сторону;
- проанализировать и изменить программу «DCM\_test7.asm», так чтобы при движении в одну сторону двигатель имел частоту вращения в 4 раза меньше, чем в другую сторону. Новую программу сохранить под именем «DCM\_test8.asm». Проверить результат ее выполнения.

### 10.3 Контрольные вопросы

10.3.1 Вывести и объяснить формулу для расчета времени задержки подпрограммы «Delay» (Тактовая частота МК =10 MHz).

10.3.2 Рассчитать значение регистра PR2 для получения частоты ШИМ = 100 kHz при ( $F_{osc}=10\text{ MHz}$ , коэффициент делителя TMR2 =1).

10.3.3 Объясните зависимость  $F$  от  $T_i$  для полученных графиков.

10.3.4 Построить диаграммы токов, протекающих через обмотки ДПТ для всех рассмотренных режимов.

10.3.5 Сколько разрядов ШИМ используется в программе «DCM\_test7.asm»?

10.3.6 Написать аналогичную программу с использованием 10-ти разрядного ШИМ.

10.3.7 Сколько разрядов ШИМ используется в программе «DCM\_test7.asm»?

10.3.8 Написать аналогичную программу с использованием 10-ти разрядного ШИМ.

## 11 Лабораторная работа №11. Управление шаговым двигателем

**Цель:** приобрести навыки подключения и способы управления шаговым двигателем (ШД), определить количество шагов.

### 11.1 Общие сведения

Асинхронной машиной (АМ) называется машина переменного тока, у которой угловая скорость ротора не равна угловой скорости магнитного поля статора. Принцип работы АМ связан с понятием вращающегося магнитного поля.

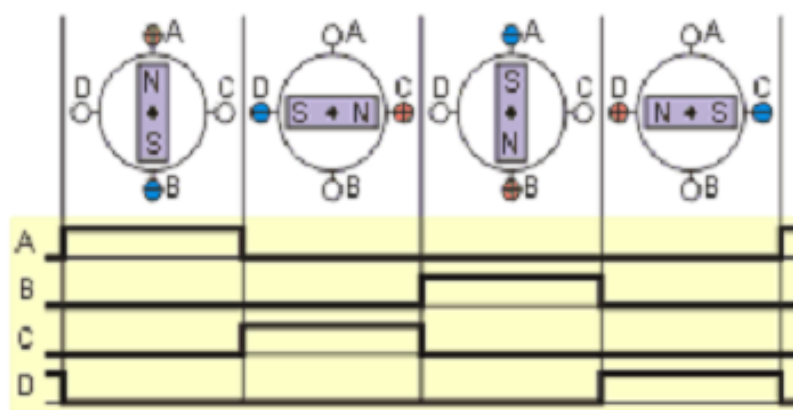
Синхронными называются электрические машины переменного тока, у которых в рабочем режиме угловая скорость ротора равна угловой скорости магнитного поля статора и не зависит от нагрузки. В отдельных случаях скорость ротора кратна скорости поля статора.

Шаговыми называют синхронные двигатели, преобразующие команду, заданную в виде импульсов, в фиксированный угол поворота вала или фиксированное перемещение без датчиков обратной связи. Это упрощает систему привода и заменяет систему следящего замкнутого привода разомкнутой системой (удешевление устройства (меньше элементов), увеличение точности в связи с фиксацией ротора ШД при отсутствии импульсов сигнала). Недостаток: при сбое импульса дальнейшее слежение идет с ошибкой в угле поворота ротора пропорционально числу пропущенных импульсов.

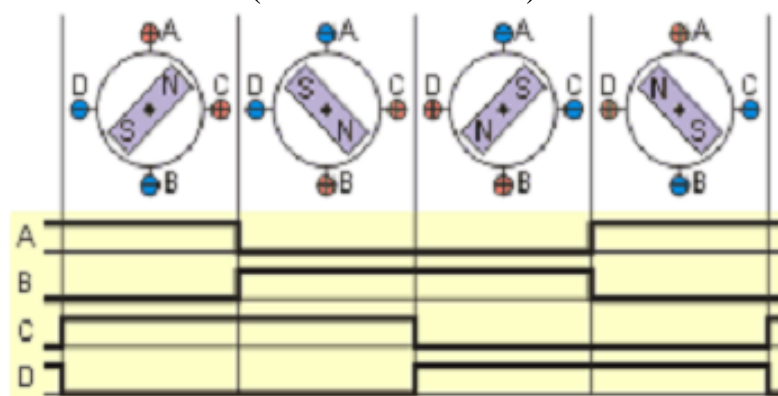
Существует несколько способов управления фазами шагового двигателя.

Первый способ обеспечивается попеременной коммутации фаз, при этом они не перекрываются: в один момент времени включена только одна фаза (рисунок 11.1, а). Этот способ называют «one phase on» full step или wave drive mode. Точки равновесия ротора для каждого шага совпадают с «естественными» точками равновесия ротора у незапитанного двигателя. Недостатком этого способа управления является то, что для биполярного двигателя в один и тот же момент времени используется 50% обмоток, а для униполярного – только 25%. Это означает, что в таком режиме не может быть получен полный момент.

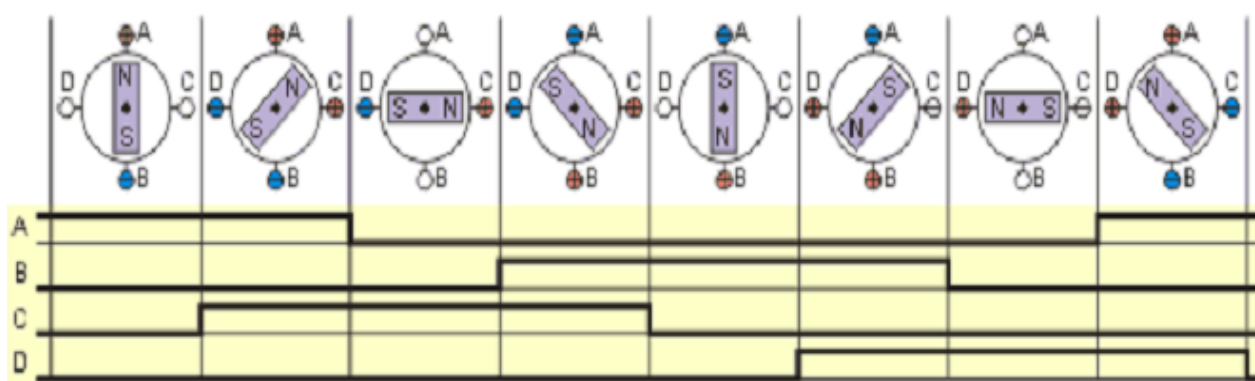
Второй способ - управление фазами с перекрытием: две фазы включены в одно и то же время. Его называют «two-phase-on» full step или просто full step mode. При этом способе управления ротор фиксируется в промежуточных позициях между полюсами статора (рисунок 11.1, б), и обеспечивается примерно на 40% больший момент, чем в случае одной включенной фазы. Этот способ управления обеспечивает такой же угол шага, как и первый способ, но положение точек равновесия ротора смещено на полшага.



а) пошаговый режим, включена одна фаза, «one phase on» full step (wave drive mode)



б) пошаговый режим, включены две фазы, «two-phase-on» full step (full step mode)



в) полушаговый режим, «one and two-phase-on» half step (half step mode)

Рисунок 11.1 - Различные способы управления фазами шагового двигателя

Третий способ является комбинацией первых двух и называется полушаговым режимом, «one and two-phase-on» half step или просто half step mode, когда двигатель делает шаг в половину основного. Этот метод управления достаточно распространен, так как двигатель с меньшим шагом стоит дороже и очень заманчиво получить от 100-шагового двигателя 200 шагов на оборот. Каждый второй шаг - запитана лишь одна фаза, а в остальных случаях запитаны две (рисунок 11.1, в). В результате угловое

перемещение ротора составляет половину угла шага для первых двух способов управления. Кроме уменьшения размера шага, этот способ управления позволяет частично избавиться от явления резонанса. Полушаговый режим обычно не позволяет получить полный момент, хотя наиболее совершенные драйверы реализуют модифицированный полушаговый режим, в котором двигатель обеспечивает практически полный момент, при этом рассеиваемая мощность не превышает номинальной.

Еще один способ управления называется микрошаговым режимом или *micro stepping mode*. При этом способе управления ток в фазах нужно менять небольшими шагами, обеспечивая, таким образом, дробление половинного шага на еще меньшие микрошаги. Когда одновременно включены две фазы, но их токи не равны, то положение равновесия ротора будет лежать не в середине шага, а в другом месте, определяемом соотношением токов фаз. Меняя это соотношение, можно обеспечить некоторое количество микрошагов внутри одного шага. Кроме увеличения разрешающей способности, микрошаговый режим имеет и другие преимущества, которые будут описаны ниже. Вместе с тем, для реализации микрошагового режима требуются значительно более сложные драйверы, позволяющие задавать ток в обмотках с необходимой дискретностью. Полушаговый режим является частным случаем микрошагового режима, но он не требует формирования ступенчатого тока питания катушек, поэтому часто реализуется.

## 11.2 Порядок выполнения

11.2.1 Для подключения и управления шаговым двигателем необходимо:

- выключить питание стенда, выполнить соединение ШД по схеме на рисунке 11.2;

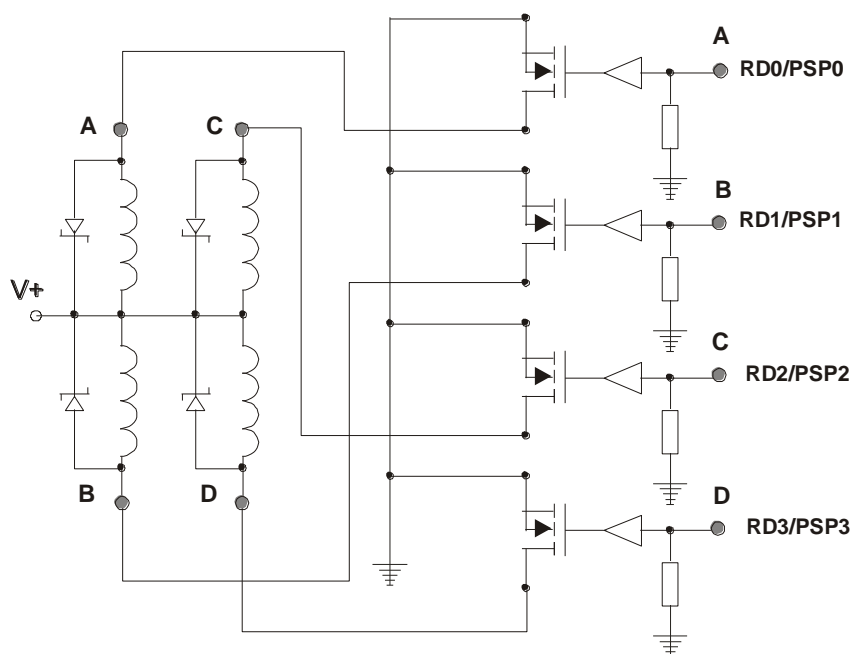


Рисунок 11.2 – Схема соединения ШД

- включить питание стенда, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;
- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «SM\_test1.asm», пример которой представлен в Приложении Ж;
- проверить результат ее выполнения: двигатель должен вращаться в одном направлении с постоянной скоростью. В данной программе используется полношаговый режим с временем шага  $T_{st} = 10$  мс;
- проанализировать и изменить программу «SM\_test1.asm» так, чтобы время шага  $T_{st} = 7$  мс. Новую программу сохранить под именем «SM\_test2.asm». Проверить результат ее выполнения.

11.2.2 Для определения количества шагов ШД необходимо:

- выключить питание стенда, выполнить соединение ШД по схеме на рисунке 11.3;

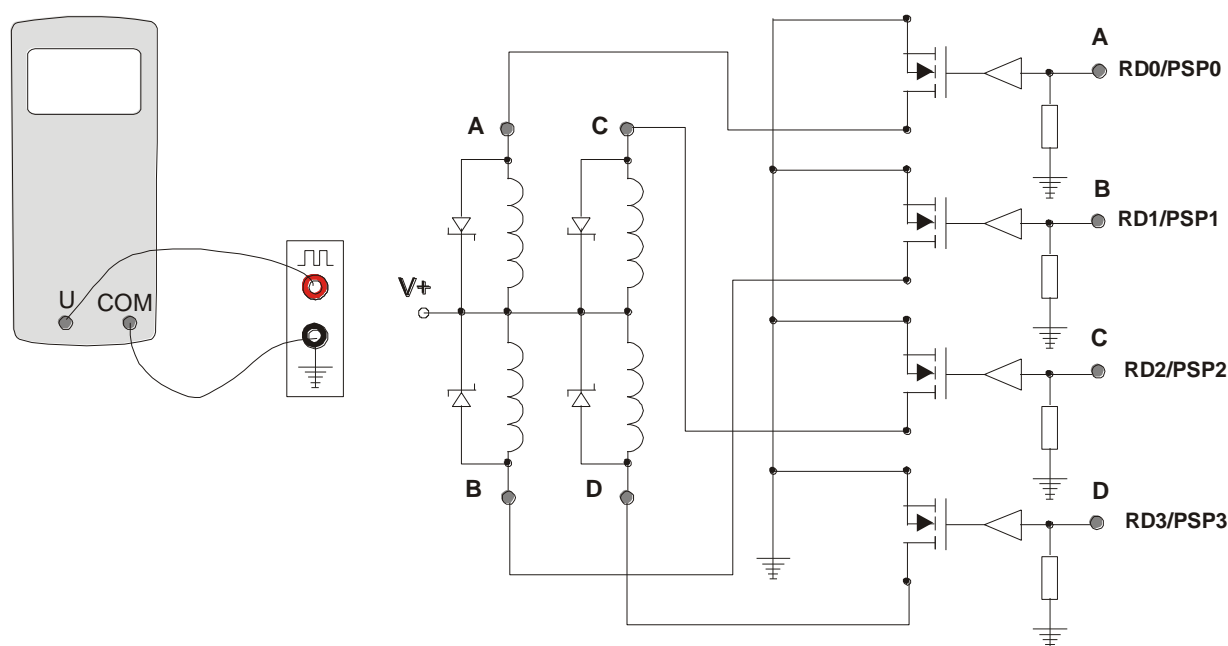


Рисунок 11.3 – Схема соединения ШД

- включить питание стенда, установить УИП в режим измерений частоты, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;
- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «SM\_test1.asm»;
- проверить результат ее выполнения: двигатель должен вращаться в одном направлении с постоянной скоростью.

Снять показания частоты  $F_t$  с УИП. Вычислить количество шагов  $S_{st}$  ШД на один полный оборот, используя следующие данные:

*Время 1 шага  $T_{st} = 10$  мс;*

*$F_m = F_t / 24$  (частота вращения вала ШД);*

- используя программную среду MPLAB IDE, записать в МК и выполнить программу из файла «SM\_test2.asm»;

- проверить результат ее выполнения: двигатель должен вращаться в одном направлении с постоянной скоростью. Снять показания частоты Ft с УИП. Вычислить количество шагов Sst ШД на один полный оборот, используя следующие данные:

*Время 1 шага  $T_{st} = 7$  мс;*

*$F_m = F_t / 24$  (частота вращения вала ШД);*

- сравнить полученные значения Sst.

11.2.3 Для управления ШД различными способами необходимо:

- выключить питание стенда, выполнить соединение ШД по схеме на рисунке 11.3;

- включить питание стенда, установить УИП в режим измерений частоты, включить ПК, запустить программу MPLAB IDE, установить соединение программы со стендом;

- используя программную среду MPLAB IDE, записать в МК и выполнить программу управления ШД с одной рабочей фазой, файл «SM\_test3.asm»;

- проверить результат ее выполнения: двигатель должен вращаться в одном направлении с постоянной скоростью. Снять показания частоты Ft с УИП. Вычислить частоту вращения вала ШД  $F_m$ .

*$F_m = F_t / 24$  (частота вращения вала ШД);*

- используя программную среду MPLAB IDE, записать в МК и выполнить программу управления ШД с двумя рабочими фазами, файл «SM\_test4.asm»;

- проверить результат ее выполнения: двигатель должен вращаться в одном направлении с постоянной скоростью. Снять показания частоты Ft с УИП. Вычислить частоту вращения вала ШД  $F_m$ .

*$F_m = F_t / 24$  (частота вращения вала ШД);*

- сравнить используемые подпрограммы управления ШД и полученные значения  $F_m$ .

## **11.3 Контрольные вопросы**

11.3.1 Какие существуют типы ШД и как определить тип ШД?

11.3.2 Сколько шагов на один полный оборот имеют стандартные ШД?

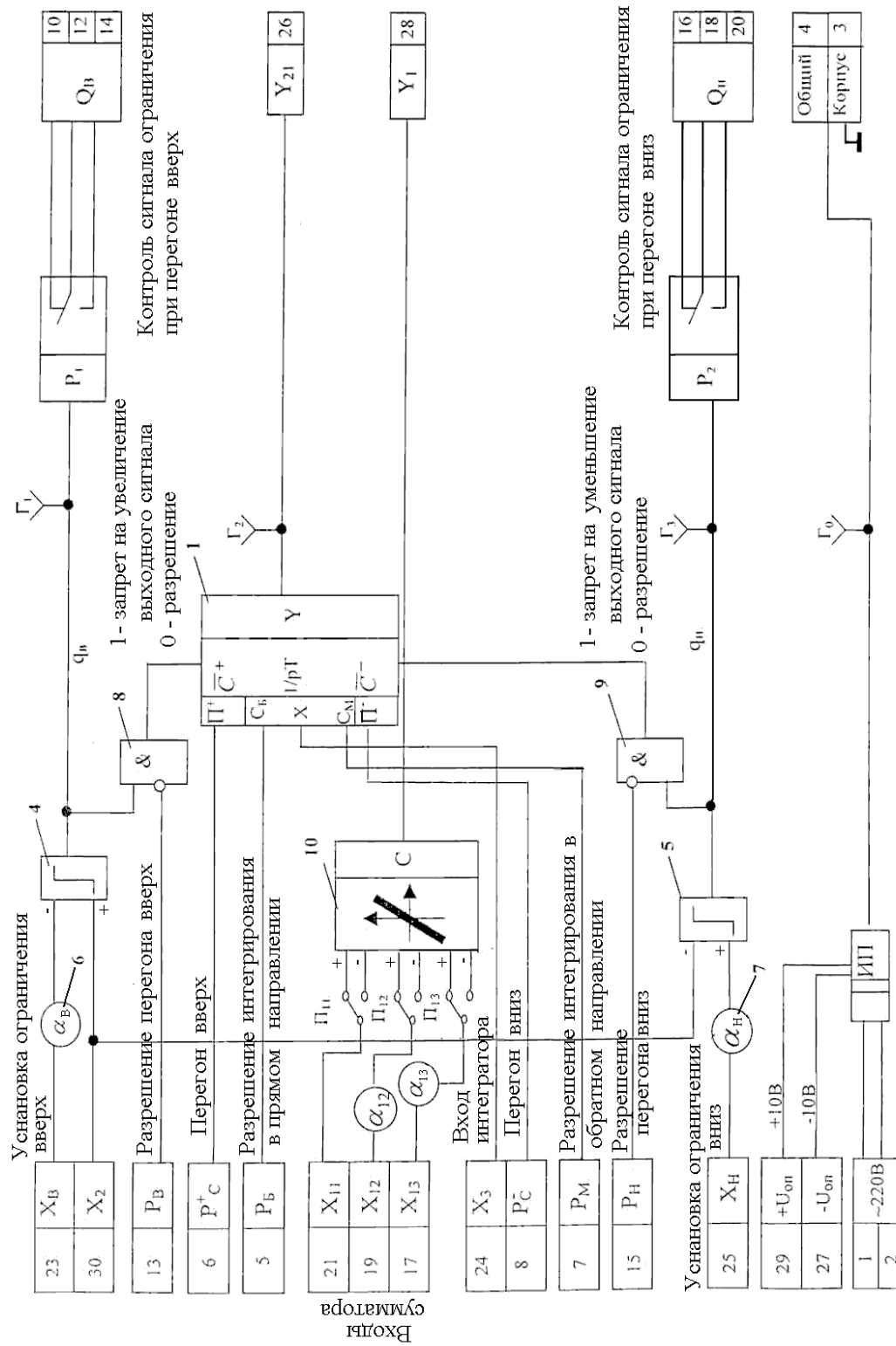
11.3.3 Вывести формулу для вычисления количества шагов ШД на один полный оборот, если известны: частота вращения ШД и время одного шага.

11.3.4 Объяснить разницу в 2-х используемых способах управления ШД.

11.3.5 Какие еще есть способы управления ШД?

## Приложение А

### Структурная схема блока БДП-П



1 - интегратор; 4,5 - нуль-органы; 6,7 - блоки масштабирования; 8,9 - логические схемы; 10 - входной сумматор; P1, P2 - выходные реле; ИП - источник питания; C<sub>Б</sub>, C<sub>М</sub>, П<sup>+</sup>, П<sup>-</sup>, C<sup>+</sup>, C<sup>-</sup> - логические входные сигналы интегратора 1.

Рисунок А.1 – Схема блока БДП-П



На рисунке А.1 приведена структурная схема блока с указанием назначения входных и выходных зажимов и номеров клемм, а также показаны входы логического управления интегратором:

- $+U_{оп}$  – положительное опорное напряжение клемма 29;
- $-U_{оп}$  – отрицательное опорное напряжение клемма 27;
- $X_3$  - основной свободный вход интегратора (клемма 24);
- вход сумматора  $X_{11}$  (клемма 21) не масштабируемый;
- входы сумматора  $X_{12}$  (клемма 19) и  $X_{13}$  (клемма 17) масштабируемые входы.

Входы логического управления интегратором:

- вход  $P_B$  (клемма 5) - вход разрешения интегрирования в прямом направлении. При прямом направлении изменение выходного сигнала  $Y_{12}$  в процессе интегрирования имеет знак, совпадающий со знаком входного сигнала  $X_3$ ;

- вход  $P_M$  (клемма 7) - вход разрешения интегрирования в обратном направлении. При обратном направлении изменение выходного сигнала  $Y_{12}$  в процессе интегрирования имеет знак, противоположный знаку входного сигнала  $X_3$ ;

- $C^+$  - вход запрета увеличения выходного сигнала  $Y_{21}$ ;
- $C^-$  - вход запрета уменьшения выходного сигнала  $Y_{21}$ ;
- $P^+_C$  (клемма 6) - вход запуска ускоренного увеличения (перегона вверх) выходного сигнала  $Y_{21}$ ;
- $P^-_C$  (клемма 8) - вход запуска ускоренного увеличения (перегона вниз) выходного сигнала  $Y_{21}$ .

Выходные сигналы блока:

- $Y_{21}$  (клемма 26) - основной выходной сигнал интегратора;
- $Y_1$  (клемма 28) - выходной сигнал сумматора;
- $Q_B$  (клеммы 10, 12, 14) - контроль сигнала ограничения при перегоне вверх;
- $Q_N$  (клеммы 16, 18, 20) - контроль сигнала ограничения при перегоне вниз.

Запрет на увеличение или уменьшение выходного сигнала осуществляется с помощью логических элементов «И». Единица на выходе логического элемента означает «запрет», а ноль - «разрешение». «Разрешение» поступает на вход  $C^+$  только в том случае, если одновременно есть сигнал «установка ограничения напряжения вверх» на входе  $X_B$  (клемма 23) и нет сигнала на входе «разрешение перегона вверх»  $P_B$  (клемма 13). Аналогично работает канал по запрету перегона вниз.

## Приложение Б

### Описание УМК-7 и MPLAB

На передней панели стенда (рисунок Б.1) расположены:

- 1 - микроконтроллер PIC16F877;
- 2 - ряд клемм, соединенных с выводами PIC16F877;
- 3 - ряд клемм, соединенных с выводами внешних устройств;
- 4 –источник звука;
- 5 –источник аналогового сигнала;
- 6 –панель светодиодов;
- 7 - жидкокристаллический индикатор;
- 8 - клавиатура.

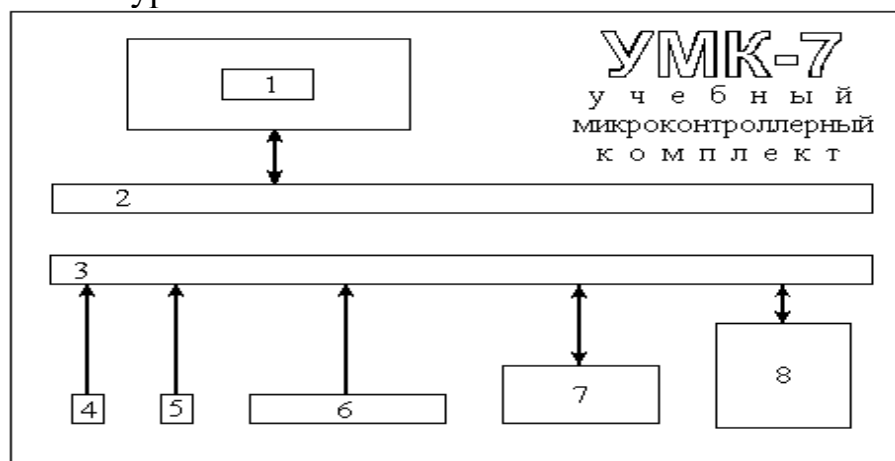


Рисунок Б.1- Передняя панель УМК-7

MPLAB-ICD – оценочный комплект для микроконтроллеров серии PIC16F87X. В нем используется возможность внутрисхемной отладки (ICD), встроенной в кристаллы PIC16F87X, и протокол внутрисхемного программирования фирмы Microchip. MPLAB-ICD одновременно является программатором и внутрисхемным отладчиком, поскольку он подключается к отлаживаемому устройству и работает как микроконтроллер PIC16F87X. Комплект MPLAB-ICD предназначен для отладки кода в лабораторных условиях и поддерживает:

- пошаговое выполнение кода в реальном масштабе времени;
- точки останова (break points);
- внутрисхемную отладку;
- встроенное программирование;
- отладку по исходному коду и символическим именам;
- RS-232 интерфейс.

Технические данные микроконтроллера:

- диапазон выходных рабочих напряжений от 3.0 V до 5.5 V постоянного тока;
- рабочие частоты от 32 кГц до 20 МГц.

## Приложение В

### Карта памяти МК PIC16877

				Адрес			
Регистр косвенной адресации	00h	Регистр косвенной адресации	80h	Регистр косвенной адресации	100h	Регистр косвенной адресации	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h		105h		185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h		107h		187h
PORTD <sup>(1)</sup>	08h	TRISD <sup>(1)</sup>	88h		108h		188h
PORTE <sup>(1)</sup>	09h	TRISE <sup>(1)</sup>	89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Резерв <sup>(2)</sup>	18Eh
TMR1H	0Fh		8Fh	EEADRH	10Fh	Резерв <sup>(2)</sup>	18Fh
T1CON	10h		90h	Регистры общего назначения  16 байт	110h	Регистры общего назначения  16 байт	190h
TMR2	11h	SSPCON2	91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h		95h		115h		195h
CCPR1H	16h		96h		116h		196h
CCP1CON	17h		97h		117h		197h
RCSTA	18h	TXSTA	98h		118h		198h
TXREG	19h	SPBRG	99h		119h		199h
RCREG	1Ah		9Ah		11Ah		19Ah
CCPR2L	1Bh		9Bh		11Bh		19Bh
CCPR2H	1Ch		9Ch		11Ch		19Ch
CCP2CON	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh
Регистры общего назначения  96 байт	20h	Регистры общего назначения  80 байт	A0h	Регистры общего назначения  80 байт	120h	Регистры общего назначения  80 байт	1A0h
			EFh		16Fh		1EFh
		Доступ к 70h-7Fh	F0h	Доступ к 70h-7Fh	170h	Доступ к 70h-7Fh	1F0h
Банк 0	7Fh	Банк 1	FFh	Банк 2	17Fh	Банк 3	1FFh

\* - не физический регистр  
Закрашенные участки памяти данных не реализованы, значение при чтении 00h

Адрес аккумулятора W равен 200h

## Приложение Г

### Регистр STATUS

В таблице Г.1 показаны расположение и имена битов регистра STATUS, которые используются для контроля результатов арифметических и логических операций, для выбора банков и других целей.

Таблица Г.1 – Биты регистра STATUS

Номера бит	7	6	5	4	3	2	1	0
Имена бит	IRP	RP1	RP0	-TO	-PD	Z	DC	C

В таблице Г.1 приняты следующие имена:

а) IRP - бит выбора банка при косвенной адресации (IRP=1 - выбор банка 2 или банка 3, IRP=0 - выбор банка 0 или 1);

б) RP1, RP0 - биты выбора банка при непосредственной адресации (т.е. разряды 5 и 6). Их значения для выбора банков показаны в таблице Г.2;

в) TO - флаг переполнения сторожевого таймера;

г) PD - флаг включения питания;

д) Z - флаг нулевого результата. Записывается «1» при нулевом результате арифметической или логической операции;

е) DC - флаг десятичного переноса или заема. Записывается «1», если был перенос из младшего полубайта регистра в старший полубайт;

ж) C - флаг переноса или заема. Записывается «1», если был перенос из старшего бита регистра для команд сложения и вычитания.

Таблица Г.2 – Выбор номера банка при непосредственной адресации

RP1 (первый бит номера банка)	RP0 (нулевой бит номера банка)	Номер банка (десятичная система)
0	0	0
0	1	1
1	0	2
1	1	3
Примечание - RP1 и RP0 образуют разряды двоичного числа		

К отдельным битам регистра можно обращаться по имени или по номеру. Инструкция BTFSC STATUS, Z эквивалентна команде BTFSC STATUS, 2.

Внимание! Символьное имя S обозначает регистр STATUS.

Таблица Г.3 – Значения регистра STATUS после команды CLRF STATUS

IRP	RP1	RP0	-TO	-PD	Z	DC	C
0	0	0	1	1	1	1 или 0	1 или 0

## Приложение Д

### Описание инструкций МК PIC

Константа в инструкциях представлена символом *k*. В описании инструкций указаны флаги, которые могут измениться при ее выполнении. Параметр *d* (*dest*) в инструкциях определяет, в какой регистр записывается результат. Если *d*=0 или отсутствует, результат сохраняется в регистре *W*. Если *d*=1, результат сохраняется в регистре *f*.

Вместо адреса регистра в программе более удобно писать его символьное имя. Например, если регистр по адресу *h'21* имеет имя *R1*, тогда инструкция, описанная как *ADDWF f, d*, в программе может быть записана двумя способами: *ADDWF h'21, W* или *ADDWF R1, W*.

Директива *ORG h'00* – это указатель для ассемблера, что код следующий за этим выражением, начинается с нулевого адреса ЭППЗУ.

**ADDLW** - Сложить *k* с *W*  
Синтаксис: *[label] ADDLW k*  
Операнды:  $0 \leq k \leq 255$   
Операция:  $(W) + k \rightarrow (W)$   
Изменяемые флаги: *C, DC, Z*

**ADDWF** - Сложение *W* и *f*  
Синтаксис: *[label] ADDWF f, d*  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(W) + (f) \rightarrow (dest)$   
Изменяемые флаги: *C, DC, Z*

**ANDLW** - Побитное 'И' *k* и *W*  
Синтаксис: *[label] ANDLW k*  
Операнды:  $0 \leq k \leq 255$   
Операция:  $(W) .AND. k \rightarrow (W)$   
Изменяемые флаги: *Z*

**ANDWF** - Побитное 'И' *W* и *f*  
Синтаксис: *[label] ANDWF f, d*  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(W) .AND. (f) \rightarrow (dest)$   
Изменяемые флаги: *Z*

**BCF** - Очистить бит *b* в регистре *f*  
Синтаксис: *[label] BCF f, b*  
Операнды:  $0 \leq f \leq 127; 0 \leq b \leq 7$

Операция:  $0 \rightarrow (f < b >)$   
Изменяемые флаги: нет

**BSF** - Установить бит *b* в регистре *f*  
Синтаксис: *[label] BSF f, b*  
Операнды:  $0 \leq f \leq 127; 0 \leq b \leq 7$   
Операция:  $1 \rightarrow f < b >$   
Изменяемые флаги: нет

**BTFSC** - Проверить бит *b* в регистре *f*, пропустить следующую инструкцию, если *b*=0  
Синтаксис: *[label] BTFSC f, b*  
Операнды:  $0 \leq f \leq 127; 0 \leq b \leq 7$   
Изменяемые флаги: нет  
Описание: Если бит *b* в регистре *f* равен 0, то следующая инструкция программы пропускается.

**BTFSS** - Проверить бит *b* в регистре *f*, пропустить, если *b* = 1  
Синтаксис: *[label] BTFSS f, b*  
Операнды:  $0 \leq f \leq 127; 0 \leq b \leq 7$   
Изменяемые флаги: нет  
Описание: Если бит *b* в регистре *f* равен '1', то следующая инструкция программы пропускается.

CALL Вызов подпрограммы  
Синтаксис: *[label]* CALL f

CLRF - Очистить f  
Синтаксис: *[label]* CLRF f  
Операнды:  $0 \leq f \leq 127$   
Операция:  $00h \rightarrow (f); 1 \rightarrow Z$   
Изменяемые флаги: Z  
Описание: Очистить содержимое регистра f и установить флаг Z=1.

COMF - Инвертировать f  
Синтаксис: *[label]* COMF f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(-f) \rightarrow (dest)$   
Изменяемые флаги: Z  
Описание: Инвертировать все биты в регистре f.

DECF - Вычесть 1 из регистра f  
Синтаксис: *[label]* DECF f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(f) - 1 \rightarrow (dest)$   
Изменяемые флаги: Z

DECFSZ - Вычесть 1 из f и пропустить, если f=0  
Синтаксис: *[label]* DECFSZ f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(f) - 1 \rightarrow (dest)$   
Изменяемые флаги: нет  
Описание: Декрементировать содержимое регистра f. Если результат равен «0», то следующая инструкция программы пропускается.

GOTO - Безусловный переход  
Синтаксис: GOTO k  
Операнды:  $0 \leq k \leq 2047$   
Операция: Переход на метку  
Изменяемые флаги: нет

INCF - Прибавить 1 к f

Синтаксис: *[label]* INCF f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(f) + 1 \rightarrow (dest)$   
Изменяемые флаги: Z

IORLW - Побитное 'ИЛИ' K и W  
Синтаксис: *[label]* IORLW k  
Операнды:  $0 \leq k \leq 255$   
Операция:  $(W).OR.(k) \rightarrow (W)$   
Изменяемые флаги: Z

IORWF - Побитное 'ИЛИ' W и f  
Синтаксис: *[label]* IORWF f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(W).OR.(f) \rightarrow (dest)$   
Изменяемые флаги: Z

MOVF - Переслать f  
Синтаксис: *[label]* MOVF f, d  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(f) \rightarrow (dest)$   
Изменяемые флаги: Z  
Описание: Значение d=1 используется для проверки содержимого регистра f на ноль, которое регистрируется флагом Z.

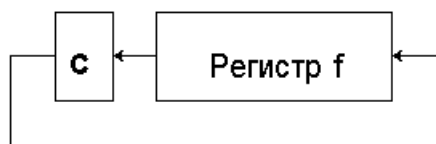
MOVLW - Переслать K в W  
Синтаксис: *[label]* MOVLW k  
Операнды:  $0 \leq k \leq 255$   
Операция:  $k \rightarrow (W)$   
Изменяемые флаги: нет  
Описание: В неиспользуемых битах ассемблер устанавливает '0'.

MOVWF - Переслать W в f  
Синтаксис: *[label]* MOVWF f  
Операнды:  $0 \leq f \leq 127$   
Операция:  $(W) \rightarrow (f)$   
Изменяемые флаги: нет

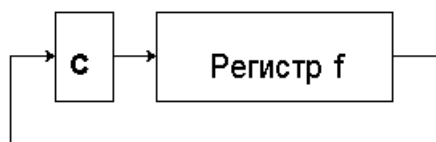
NOP - Нет операции

**RETURN** Возврат из подпрограммы  
 Синтаксис: *[label]* RETURN  
 Операнды: Нет  
 Операция: TOS  $\rightarrow$  PC  
 Изменяемые флаги: нет  
 Описание: возврат из подпрограммы. Вершина стека TOS загружается в счетчик инструкций PC.

**RLF** - Циклический сдвиг регистра f влево через бит C регистра STATUS  
 Синтаксис: *[label]* RLF f, d  
 Операнды:  $0 \leq f \leq 127$   
 Изменяемые флаги: C  
 Описание: выполняется циклический сдвиг влево содержимого регистра f через бит C регистра status.



**RRF** - Циклический сдвиг регистра f вправо через бит C регистра status  
 Синтаксис: *[label]* RRF f, d  
 Операнды:  $0 \leq f \leq 127$   
 Изменяемые флаги: C  
 Описание: выполняется циклический сдвиг вправо содержимого регистра f через бит C регистра status.



**SLEEP** Перейти в режим SLEEP  
 Синтаксис: *[label]* SLEEP  
 Операнды: нет  
 Операция: 00h  $\rightarrow$  WDT;  
 00h  $\rightarrow$  предделитель WDT;

1  $\rightarrow$  -TO; 0  $\rightarrow$  PD  
 Изменяемые флаги: -TO, -PD  
 Описание: сбросить флаг включения питания -PD в «0».  
 Установить флаг -TO переполнения WDT в «1». Очистить таймер WDT и его предделитель. Перевести микроконтроллер в режим SLEEP и выключить тактовый генератор.

**SUBLW** - Вычесть W из K  
 Синтаксис: *[label]* SUBLW k  
 Операнды:  $0 \leq k \leq 255$   
 Операция: k - (W)  $\rightarrow$  (W)  
 Изменяемые флаги: C, DC, Z

**SUBWF** - Вычесть W из f  
 Синтаксис: *[label]* SUBWF f, d  
 Операнды:  $0 \leq f \leq 127$ ; d  $\in$  [0,1]  
 Операция: (f) - (W)  $\rightarrow$  (dest)  
 Изменяемые флаги: C, DC, Z

**SWAPF** - Поменять местами полубайты в регистре f  
 Синтаксис: *[label]* SWAPF f, d  
 Операнды:  $0 \leq f \leq 127$   
 Операция: (f<3:0>)  $\rightarrow$  (dest<7:4>)  
 Изменяемые флаги: нет

**XORLW** - Побитное 'исключающее ИЛИ' K и W  
 Синтаксис: *[label]* XORLW k  
 Операнды:  $0 \leq k \leq 255$   
 Операция: (W).XOR.k  $\rightarrow$  (W)  
 Изменяемые флаги: Z

**XORWF** - Побитное 'исключающее ИЛИ' W и f  
 Синтаксис: *[label]* XORWF f, d  
 Операнды:  $0 \leq f \leq 127$   
 Операция: (W).XOR.(f)  $\rightarrow$  (dest)  
 Изменяемые флаги: Z

## Приложение Е

### Создание проекта и выполнение программы на УМК-7

Запустите MPLAB. Если появится открытый проект, закройте его командами Project – Close Project в основном меню MPLAB.

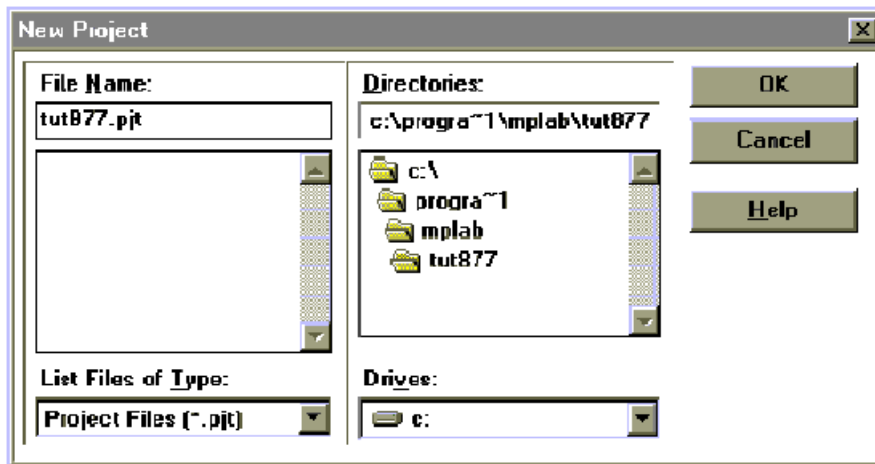


Рисунок Е.1- Окно создания проекта

Откройте новый лист, не создавая проекта, и напечатайте в окне редактора программу. При наборе текста программы применяйте регистр букв такой же, как в образце. С первой позиции печатаются метки и символьные описания регистров, с шестой позиции печатаются инструкции.

Сохраните программу с именем, имеющим для вас смысл, с расширением .asm (в примере tut877.asm) в папке program. Далее, чтобы заставить работать вашу программу, нужно её скомпилировать. То есть перевести текст программы в инструкции микроконтроллера в двоичной форме. Для этого необходимо:

- создать новый проект командами Project – New Project;
- ввести имя проекта, такое же, как у программы и нажать кнопку ОК.

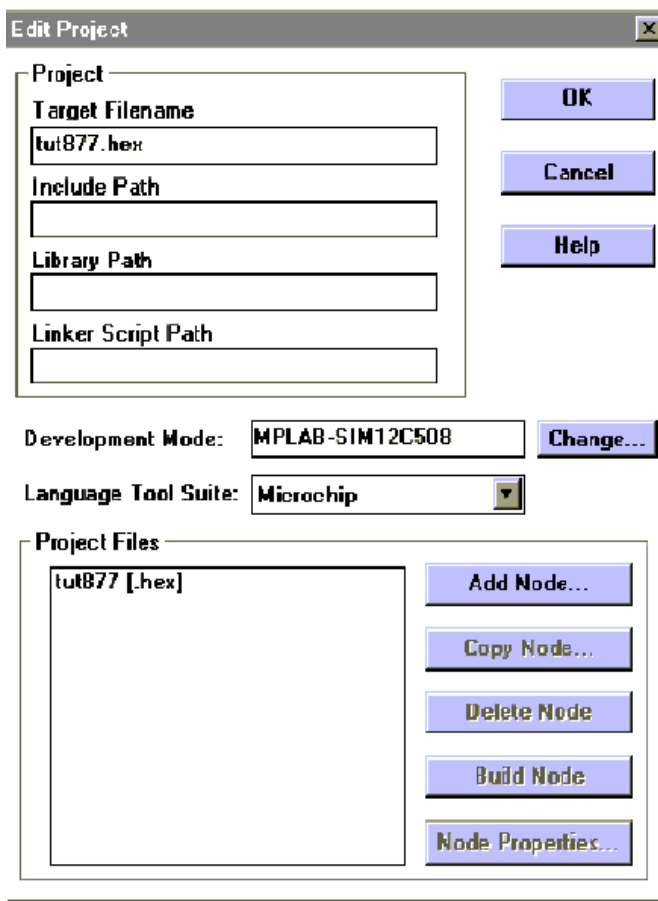


Рисунок Е.2 - Окно Edit Project



Выполняемые далее действия приводятся для проекта с именем tut877.pjt. Должно открыться диалоговое окно редактирования проекта Edit Project, показанное на рисунке Е.2.

Обратите внимание на строку Development Mode. Для нас там должна быть строка «MPLAB-ICD Debugger». Если необходимо изменить этот параметр, нажмите кнопку Change. Откроется диалоговое окно Development Mode, показанное на рисунке Е.3. Выберите строку MPLAB-ICD Debugger. Далее нажмите кнопки Apply и OK и вы вернетесь в окно Edit Project.

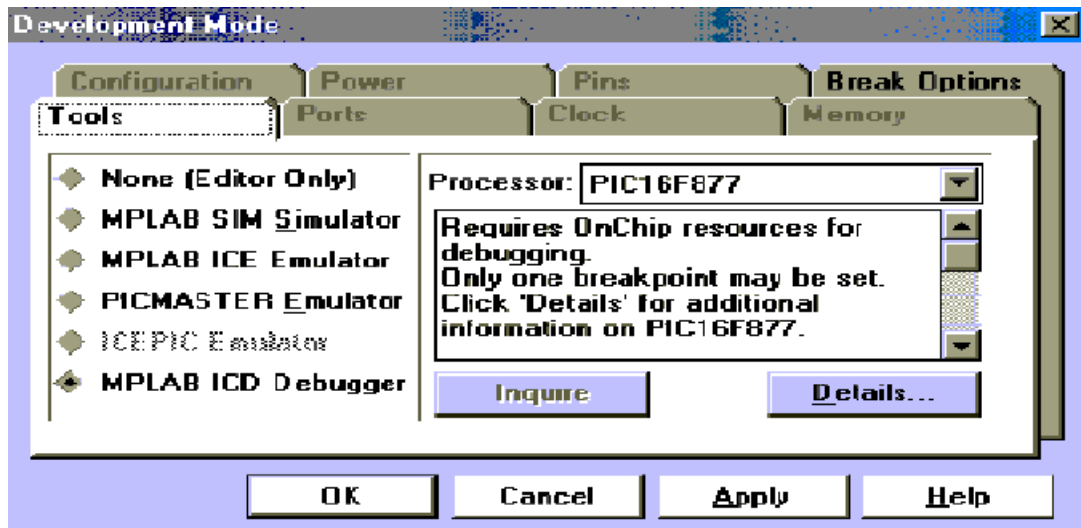


Рисунок Е.3 - Установки Development Mode

Выделите строку tut877 [.hex] в области Project Files в диалоговом окне Edit Project и затем нажмите кнопку Add Node. В диалоговом окне Add Node выберите файл tut877.asm и нажмите кнопку OK. Вы вновь вернетесь в окно Edit Project.

Ассемблер MPASM создает файл с расширением .hex, скомпилированный из asm-файла с тем же именем, что и исходный asm-файл. Нажав OK, закройте окно Edit Project.



Рисунок Е.4 – Результаты сборки проекта

Теперь командами Project - Make Project выполните компиляцию приложения. Окно Build Results (рисунок Е.4) покажет результат компиляции и отметит ошибки, если они есть. Фраза «Build completed successfully» указывает, что компиляция выполнена успешно. В случае неудачи щелкните 2 раза в окне Build Results на строку, в которой описана ошибка, и вы вернетесь в текст программы на строку, содержащую ошибку (error). Исправьте текст

программы. Устраните таким образом все ошибки. Повторно выполните команды Project - Make Project для компиляции и Project - Save Project.

Устранив все ошибки, перешлите программу в МК, нажав в окне MPLAB-ICD кнопку Program (рисунок Е.5). Если это окно отсутствует, закройте и снова откройте созданный вами проект. Если эти действия не помогут, переустановите значения Development Mode командами Options - Development Mode – MPLAB ICD Debugger.

Программирование кристалла может занять пару минут, в течение которых в поле Status показывается процесс выполнения. Завершение пересылки в поле Status отображается сообщением «Waiting for user command» - «Ожидание команды пользователя».

Создайте окно наблюдения командами Window – Watch Window – New Watch Window. Напечатайте в поле редактирования символы «PO» и из предложенного программой списка выберите строку PORTC. Нажмите кнопку Properties и выберите двоичный формат вывода чисел. Далее нажмите кнопку Add. В результате будет создано новое окно наблюдения с именем Watch1. Теперь напечатайте в поле редактирования «W» и из предложенного списка выберите W, нажмите на кнопку Add. Повторите это для регистров TRISC, Status и всех регистров программы, затем закройте окно кнопкой Close. Сохраните окно наблюдения с тем же именем, как у программы.



Рисунок Е.5 – Окно MPLAB-ICD

Запустите на выполнение программу в пошаговом режиме командами Debug – Run – Step или кнопкой F7. Следите за результатами выполнения инструкций по окну наблюдения.

Если пошаговое выполнение программы застряло на какой-либо строке и не продолжается дальше, выполните команду Debug – Clear all points, убрав точки останова.

Остановите выполнение программы командами Run - Reset. Команда Halt просто приостанавливает выполнение программы, и после команд Debug – Run- Step выполнение программы продолжится с места останова.

Внимание! При любом изменении проекта необходимо остановить программу и вновь повторить описанные выше действия по созданию проекта.

При необходимости исправьте программу и включите ее в отчет по лабораторной работе. Рекомендуется скопировать текст программы в Блокнот и разместить в папке Мои документы.

Повторно запустите программу в автоматическом режиме кнопкой F9.

## Приложение Ж

DCM\_test1.asm - тестовая программа ДПТ, один канал, скорость вращения не меняется:

```
include <p16f877a.inc>
SIMVOL      EQU    H'0020' ; Bank 0
SIMVOLL      EQU    H'0021' ; Bank 0
SIM          EQU    H'0022' ; Bank 0
org          H'0000'
RESET
GOTO START
org          H'0010'
START
    CLRF      STATUS                ; Банк 0
    CLRF      INTCON                ; Запретить все прерывания
    CLRF      PORTC                 ; Очистить PORTC
    BSF       STATUS,RP0            ; Bank 1
    MOVLW     B'10000000'           ;
    MOVWF     TRISC                 ;
    BCF       STATUS,RP0            ; Bank 0
LOOP_GO BSF      PORTC,2             ;
    MOVLW     D'0200'               ; Задержка
    CALL Delay                       ;
    BCF       PORTC,2               ;
    MOVLW     D'0200'               ; Задержка
    CALL DELAY                       ;
    GOTO      LOOP_GO               ;
Delay
    MOVWF     SIMVOLL               ;
PAU  MOVWF     SIMVOL               ;
PAUZ MOVWF     SIM                  ;
PAUZA DECFSZ   SIM,F                ;
    GOTO      PAUZA                 ;
    DECFSZ    SIMVOL,F              ;
    GOTO      PAUZ                  ;
    DECFSZ    SIMVOLL,F             ;
    GOTO      PAU                   ;
    RETURN                                ;
end
```

SM\_test1.asm - тестовая программа ШД, 4 канала

```
include <p16f877a.inc>
SIMVOL      EQU    H'0020' ; Bank 0
SIMVOLL      EQU    H'0021' ; Bank 0
SIM          EQU    H'0022' ; Bank 0
org          H'0000'
RESET
GOTO START
org          H'0010'
START
    CLRF      STATUS                ; Банк 0
    CLRF      INTCON                ; Запретить все прерывания
    BSF       STATUS,RP0            ; Bank 1
    CLRF      PIE1                  ; Выключить периферийные прерывания
    BCF       STATUS,RP0            ; Bank 0
    CLRF      PORTC                 ; Очистить PORTC
    CLRF      PORTD                 ; Очистить PORTD
    BSF       STATUS,RP0            ; Bank 1
    MOVLW     B'10000000'           ;
```

```

MOVWF    TRISC
MOVLW    B'00000000'
MOVWF    TRISD
BCF      STATUS,RP0    ; Bank 0
CALL     PP_TMR1_INI    ; Инициализация TMR1

LOOP_GO
CALL     RIGHT_SM
GOTO     LOOP_GO

; Движение ШД вправо

RIGHT_SM
MOVLW    B'00000001'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00000100'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00000010'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00001000'
MOVWF    PORTD
CALL     TMR1_10ms
RETURN

; Движение ШД вправо

RIGHT_SM1
MOVLW    B'00000101'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00000110'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00001010'
MOVWF    PORTD
CALL     TMR1_10ms
MOVLW    B'00001001'
MOVWF    PORTD
CALL     TMR1_10ms
RETURN

; ИНИЦИАЛИЗАЦИЯ TMR1

PP_TMR1_INI
MOVLW    B'00000000'
MOVWF    T1CON    ; Stop Timer1, Internal Clock Source, T1 oscillator disabled, prescaler = 1:1
CLRF     TMR1H    ; Clear Timer1 High byte register
CLRF     TMR1L    ; Clear Timer1 Low byte register
; Подпрограмма TMR1 производит отсчет частоты 50 Hz, нужно получить 20ms
; вх. частота 10MHz/4/ = 2.5 MHz, 2.5MHz/50Hz = 50000, D'50000' = H'C350'
; FFFF-C350 = 3CAF

TMR1_20ms
BCF      T1CON,TMR1ON    ; Timer1 OFF
MOVLW    H'003C'
MOVWF    TMR1H
MOVLW    H'00AF'
MOVWF    TMR1L
BCF      PIR1,TMR1IF
BSF      T1CON,TMR1ON    ; Timer1 ON
CALL     PP_TMR1_IF    ; Ждем окончания отсчета
RETURN
; Подпрограмма TMR1 производит отсчет частоты 100 Hz, нужно получить 10ms
; вх. частота 10MHz/4/ = 2.5 MHz, 2.5MHz/100Hz = 25000, D'25000' = H'61A8'
; FFFF-61A8 = 9E57

TMR1_10ms
BCF      T1CON,TMR1ON    ; Timer1 OFF

```

```

    MOVLW      H'009E'
    MOVWF      TMR1H
    MOVLW      H'0057'
    MOVWF      TMR1L
    BCF        PIR1,TMR1IF
    BSF        T1CON,TMR1ON      ; Timer1 ON
    CALL PP_TMR1_IF              ; Ждем окончания отсчета
    RETURN

; Подпрограмма TMR1 производит отсчет частоты 200 Hz, нужно получить 5ms
; вх. частота 10MHz/4/ = 2.5 MHz, 2.5MHz/200Hz = 12500, D'12500' = H'30D4'
; FFFF-30D4 = CF2B

TMR1_5ms
    BCF        T1CON,TMR1ON      ; Timer1 OFF
    MOVLW      H'00CF'
    MOVWF      TMR1H
    MOVLW      H'002B'
    MOVWF      TMR1L
    BCF        PIR1,TMR1IF
    BSF        T1CON,TMR1ON      ; Timer1 ON
    CALL PP_TMR1_IF              ; Ждем окончания отсчета
    RETURN

; Подпрограмма TMR1 производит отсчет частоты 500 Hz, нужно получить 2ms
; вх. частота 10MHz/4/ = 2.5 MHz, 2.5MHz/500Hz = 5000, D'5000' = H'1388'
; FFFF-1388 = EC77

TMR1_2ms
    BCF        T1CON,TMR1ON      ; Timer1 OFF
    MOVLW      H'00EC'
    MOVWF      TMR1H
    MOVLW      H'0077'
    MOVWF      TMR1L
    BCF        PIR1,TMR1IF
    BSF        T1CON,TMR1ON      ; Timer1 ON
    CALL PP_TMR1_IF              ; Ждем окончания отсчета
    RETURN

PP_TMR1_IF
    BTFSS     PIR1,TMR1IF
    GOTO      PP_TMR1_IF
    RETURN

Delay
    MOVWF     SIMVOLL
PAU  MOVWF     SIMVOL
PAUZ MOVWF     SIM
PAUZA DECFSZ   SIM,F
      GOTO     PAUZA
      DECFSZ   SIMVOL,F
      GOTO     PAUZ
      DECFSZ   SIMVOLL,F
      GOTO     PAU
    RETURN
end

```

## Список литературы

- 1 Шандров Б.В. Технические средства автоматизации. М.: ИЦ "Академия", 2010.
- 2 Исембергенов Н.Т. Элементы и устройства автоматики. - А., «Бастау», 2009.
- 3 Беляев Г.Б., Кузищин В.Ф. и др. Технические средства автоматизации в теплоэнергетике. М., Энергоатомиздат, 1982.
- 4 Тарасов В.М. Программирование цифровой техники и микроконтроллеров управления. Конспект лекций для студентов всех форм обучения специальности 5В070200 «Автоматизация и управление».- Алматы: АУЭС, 2011.
- 5 Катцен С. PIC микроконтроллеры: полное руководство.- М.: «Додека», 2010.
- 6 Техническая документация МК PIC16F87X.
- 7 <http://www.microchip.ru/>

## Содержание

Введение.....	3
1 Лабораторная работа №1. Проверка блока динамических преобразований БДП-П .....	4
2 Лабораторная работа №2. Исследование аналогового блока РП4-У и исполнительного механизма постоянной скорости.....	10
3 Лабораторная работа №3. Логическое преобразование дискретных сигналов.....	16
4 Лабораторная работа №4. Двигатель постоянного тока: параметры и характеристики.....	25
5 Лабораторная работа №5. Изучение среды MPLAB. Управление оборудованием.....	29
6 Лабораторная работа №6. Сложение, вычитание. Логика.....	32
7 Лабораторная работа №7. Анализ информации и автоматизация.....	35
8 Лабораторная работа №8. Подпрограммы .....	39
9 Лабораторная работа №9. Программирование клавиатуры.....	43
10 Лабораторная работа №10. Управление двигателем постоянного тока....	46
11 Лабораторная работа №11. Управление шаговым двигателем .....	51
Приложение А. Структурная схема блока БДП-П.....	56
Приложение Б. Описание УМК-7 и MPLAB .....	58
Приложение В. Карта памяти микроконтроллера PIC16877.....	59
Приложение Г. Регистр STATUS .....	60
Приложение Д. Описание инструкций микроконтроллера PIC .....	61
Приложение Е. Создание и выполнение программы на УМК-7.....	64
Приложение Ж. Примеры программ.....	67
Список литературы .....	70

Лариса Николаевна Рудакова

## ЭЛЕМЕНТЫ И УСТРОЙСТВА АВТОМАТИКИ

Методические указания по выполнению лабораторных работ  
для студентов специальности 5В070200 – Автоматизация и управление

Редактор Л.Т. Сластихина

Специалист по стандартизации Н.К. Молдабекова

Подписано в печать \_\_\_\_ . \_\_\_\_ . \_\_\_\_ .

Тираж 60 экз.

Объем 4.4 уч.-изд. л.

Формат 60х84 1/16

Бумага типографская №1

Заказ \_\_\_\_ . Цена 2180 тг.

Копировально-множительное бюро  
некоммерческого акционерного общества  
«Алматинский университет энергетики и связи»  
050013 Алматы, ул. Байтурсынова, 126