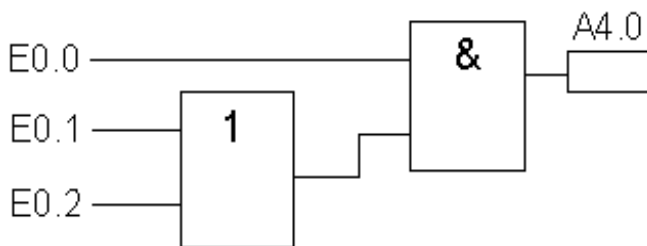


ЗАДАЧА 1

ПРОГРАММИРОВАНИЕ ПРОСТЕЙШИХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ ОПЕРАЦИЯ "И" ПЕРЕД "ИЛИ"

Постановка задачи:

Написать в LAD или FBD следующую схему управления:



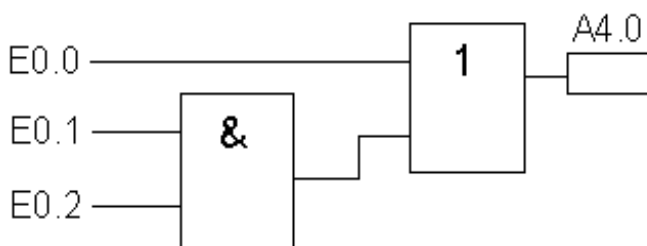
Перевести схему в представление STL, посмотреть работу в отладчике.

ЗАДАЧА 2

ПРОГРАММИРОВАНИЕ ПРОСТЕЙШИХ ЛОГИЧЕСКИХ ОПЕРАЦИЙ ОПЕРАЦИЯ "ИЛИ" ПЕРЕД "И"

Постановка задачи:

Написать в LAD и FBD следующую схему управления:



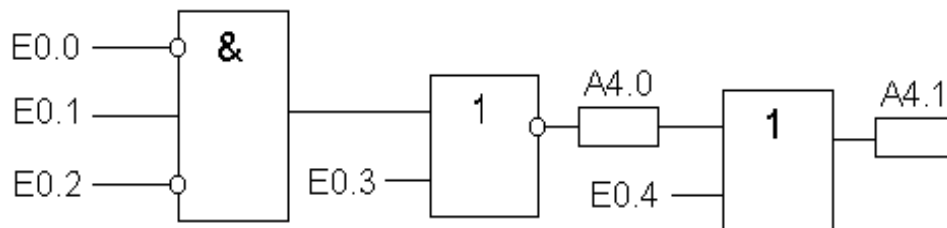
Перевести схему в представление STL, посмотреть работу в отладчике.

ЗАДАЧА 3

ПРЯМОЙ И ИНВЕРСНЫЙ ОПРОС, ОПЕРАЦИЯ ПРИСВАИВАНИЯ, ИНВЕРСИЯ ФЛАГА ЛОГИЧЕСКОГО РЕЗУЛЬТАТА

Постановка задачи:

Написать в FBD и LAD следующую схему управления:

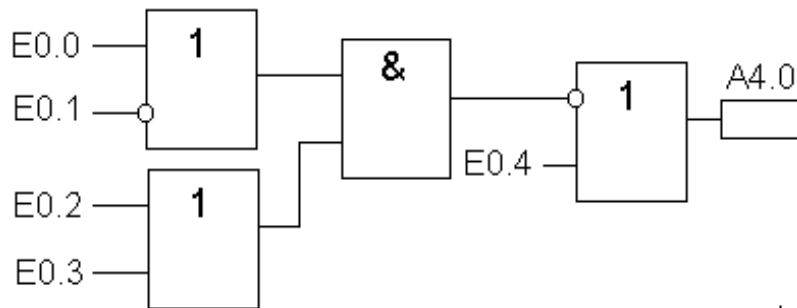


ЗАДАЧА 4

ПРЯМОЙ И ИНВЕРСНЫЙ ОПРОС, ОПЕРАЦИЯ ПРИСВАИВАНИЯ, ИНВЕРСИЯ ФЛАГА ЛОГИЧЕСКОГО РЕЗУЛЬТАТА

Постановка задачи:

Написать в STL следующую схему управления:

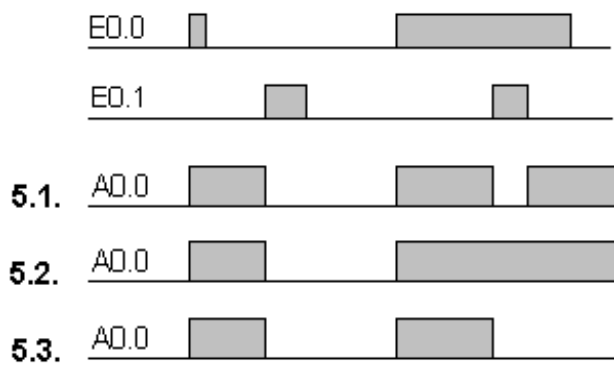


ЗАДАЧА 5

ТРИГГЕРЫ

Постановка задачи:

Написать программу, реализующую следующую диаграмму работы:

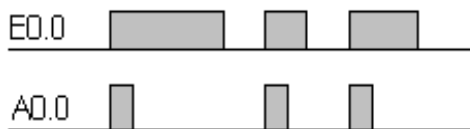


ЗАДАЧА 6

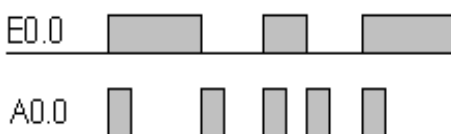
ЛОГИЧЕСКИЕ ОПЕРАЦИИ, ВЫДЕЛЕНИЕ ФРОНТОВ СИГНАЛОВ

Постановка задачи:

1. Написать функцию, осуществляющую выделение положительного фронта сигнала, т.е. функцию, реализующую следующую диаграмму работы (длительность импульсов на выходе A0.0 = длительности 1-го программного цикла):



2. Переписать программу, осуществляющую выделение обоих фронтов сигнала, используя команды логики (также использовать входные и выходные параметры):

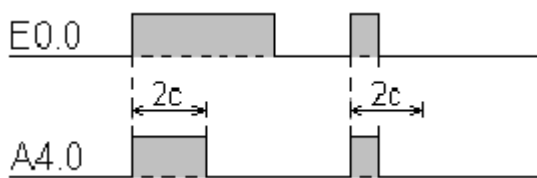


ЗАДАЧА 7

ТАЙМЕРЫ

Постановка задачи:

Необходимо написать программу, реализующую следующую диаграмму работы:

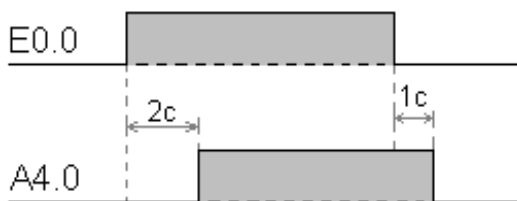


ЗАДАЧА 8

ТАЙМЕРЫ

Постановка задачи:

Необходимо написать программу, реализующую следующую диаграмму работы:



Рекомендации:

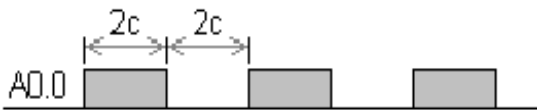
Целесообразно использовать таймеры SE (Задержка включения с подтверждением по входу) и SA (Задержка выключения с подтверждением). На выход выводить сигнал с таймера SA.

ЗАДАЧА 9

ТАЙМЕРЫ

Постановка задачи:

1. Необходимо написать программу, формирующую импульсы со скважностью 2 (меандр) на двух таймерах:



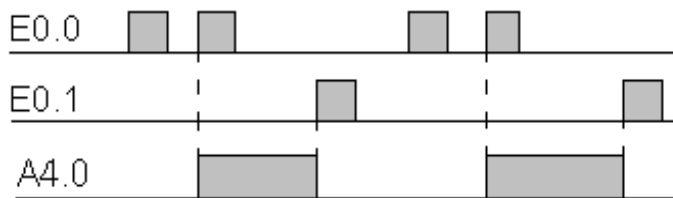
2. Попробовать написать программу, использующую только один таймер.

ЗАДАЧА 10

СЧЕТЧИКИ

Постановка задачи:

Необходимо написать программу, формирующую следующую диаграмму работы:

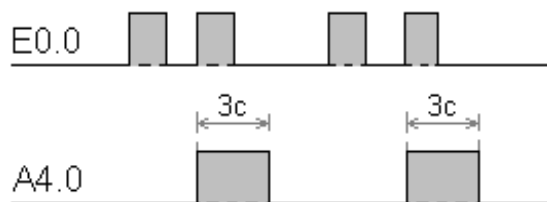


ЗАДАЧА 11

ПРЯМОЙ ОПРОС, СЧЕТЧИКИ, ТАЙМЕРЫ, ОПЕРАЦИЯ ПРИСВАИВАНИЯ

Постановка задачи:

Необходимо написать программу, реализующую следующую диаграмму работы:



Рекомендации:

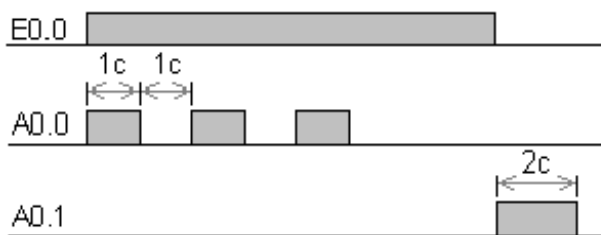
Инициализацию счетчика целесообразно производить не только в начале работы программы, но и после окончания работы таймера (для перезапуска системы).

ЗАДАЧА 12

УПРАВЛЕНИЕ ЗАДВИЖКОЙ (ИСПОЛЬЗОВАНИЕ ТАЙМЕРОВ И СЧЕТЧИКОВ)

Постановка задачи:

Используя вход управления задвижкой E0.0 ("1" - открыть, "0" - закрыть) необходимо сформировать два выходных сигнала управления задвижкой: A0.0 - команда "открыть" и A0.1 - команда "закреть". Команды "открыть" и "закреть" должны устанавливаться на какое-то время, обеспечивающее полное открытие (закрытие) задвижки, после чего сигнал управления должен сбрасываться в "0", причем открытие задвижки должно быть плавным:



Рекомендации:

Формировать сигнала "открыть" можно используя два таймера (на 1 и на 5 с); другой вариант состоит в использовании таймера и счетчика.

ЗАДАЧА 13

НАХОЖДЕНИЕ МАКСИМАЛЬНОГО ИЗ ДВУХ ЧИСЕЛ (ОПЕРАЦИИ СРАВНЕНИЯ, КОМАНДЫ ПЕРЕХОДА)

Постановка задачи:

Написать функцию с двум входными формальными параметрами типа INT и одним выходным типа INT. В выходном параметре функция должна возвращать максимальное из входных чисел.

ЗАДАЧА 14

"БЕГУЩАЯ ЕДИНИЦА" НА ВЫХОДАХ КОНТРОЛЛЕРА (КОМАНДЫ СДВИГА)

Постановка задачи:

Написать функцию, осуществляющую сдвиг "1" на выходах контроллера. Функция должна иметь входной параметр "Разрешение работы", на который заводится сигнал с входа E0.0 контроллера ("1" - формирование бегущей единицы на выходах разрешено, "0" - запрещено).

Рекомендации:

Можно, используя входной параметр функции, вызывать ее по таймеру, например один раз в 2 секунды. Но более предпочтительным является вызов функции из блока OB35.

ЗАДАЧА 15

НОРМИРОВАНИЕ АНАЛОГОВЫХ ВХОДОВ

Постановка задачи:

Имеется в распоряжении стандартный модуль 12-разрядных аналоговых входов, например, SM331 AI 8x12 Bit (331-7KF00-0AB0). Необходимо по измеренному значению входного кода получить вещественное значение измеряемой величины (например, давление в Bar).

Предпосылки:

Будем считать, что к аналоговому входу подключен датчик давления SITRANS P 7MF4000 с диапазоном измерения давления от 0 до 7 Bar. Датчик формирует выходной ток от 4 до 20 mA и отградуирован следующим образом: 4mA соответствует 0 Bar, а 20mA соответствует 7 Bar.

Адрес аналогового входа, к которому подключен датчик давления - PEW288. Аналоговый вход настроен на измерение тока в диапазоне 4 ... 20 mA.

Из описания аналоговых модулей можно найти, что:

1. Коду W#16#0000 соответствует значение минимального входного тока, не меньше нуля, т.е. в нашем примере 4mA.
2. Коду W#16#6C00 соответствует значение максимального входного тока, т.е. 20mA в нашем случае.

Решение:

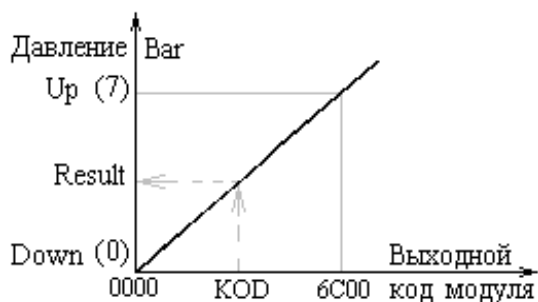
Исход из представленных данных, можно составить следующую пропорцию:

Входной ток аналогового модуля	Код, выдаваемый аналоговым модулем	Давление на входе датчика
4 mA	W#16#0000	0 Bar
20 mA	W#16#6C00	7 Bar

Оформим данную задачу как функцию FC1 с формальными параметрами. Обозначим в таблице объявления переменных функции (declaration table) локальные переменные:

Address	Decl.	Symbol	Data Type	Initial Value	Comment
0.0	in	KOD	INT	0	Измеренное значение аналогового входа (в нашем случае - PEW288)
2.0	in	Up	REAL	0.000000e+000	Давление, соответствующее выходному току датчика 20mA (7.0 Bar)
6.0	in	Down	REAL	0.000000e+000	Давление, соответствующее выходному току датчика 4mA (0.0 Bar)
10.0	out	Result	REAL	0.000000e+000	Результат (вещественное значение измеренного давления в Bar)
	in_out				

Представленную выше пропорцию можно изобразить графически в следующем виде:



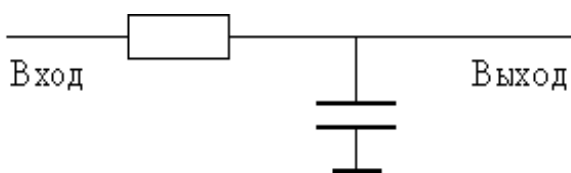
Исход из этой пропорции, можно составить конечную формулу для вычисления результата:

$$\text{Result} = \text{Down} + \frac{\text{KOD} \cdot (\text{Up} - \text{Down})}{27648}$$

ЗАДАЧА 16

РЕКУРСИВНЫЙ ФИЛЬТР ПЕРВОГО ПОРЯДКА

Постановка задачи:



Это, как известно, простейший фильтр или апериодическое звено 1-го порядка. Такой фильтр имеет передаточную функцию следующего вида:

$$W(p) = \frac{y(p)}{x(p)} = \frac{1}{1 + \tau \cdot p},$$

где $t = RC$ - постоянная времени фильтра.

Апериодическое звено можно также представить в виде дифференциального уравнения:

$$y(t) + \tau \cdot \dot{y}(t) = x(t).$$

Перейдя к дискретным интервалам времени, получим:

$$y_k + \tau \cdot \frac{y_k - y_{k-1}}{T} = x_k,$$

где k - текущий момент времени, T - шаг дискретизации по времени. Конечная формула для написания программы:

$$y_k = \frac{T \cdot x_k + \tau \cdot y_{k-1}}{T + \tau}.$$

Чем больше t , тем больше инерционность фильтра, т.е. меньше граничная частота пропускания фильтра. С увеличением t фильтр становится более низкочастотным. Переходные процессы на выходе фильтра заканчиваются примерно за время 3τ с. Т.е. если сигнал на входе фильтра с постоянной времени $t = 2$ с скачкообразно изменится от 0 до 1, то на выходе фильтра сигнал достигнет единицы примерно за $3\tau = 6$ секунд.

Программа...

Необходимо модифицировать функцию, написанную в предыдущем примере, добавив к масштабированию входного аналогового сигнала еще и его фильтрацию от высокочастотных помех, для чего достаточно использовать рекурсивный фильтр 1-го порядка, рассмотренный выше.

Рекомендации

В разработанной ранее функции масштабирования необходимо внести некоторые изменения: добавить вход разрешения фильтрации сигнала, входные параметры T и t , а также определив параметр `Result` как параметр `IN_OUT`, т.к. для фильтрации нам необходимо знать предыдущее значение отфильтрованного сигнала (т.е. `Result`, подсчитанный в прошлом программном цикле).

Внимание!

1. Фактический параметр, сопоставляемый при вызове данной функции формальному параметру `Result` должен содержать предыдущее значение отфильтрованного сигнала, поэтому он не может быть локальной переменной, это может быть только меркерная переменная или переменная блока данных.
2. Функция должна вызываться через равные промежутки времени, равные шагу дискретизации по времени T . Поэтому рекомендуется вызывать ее из блока `ОВ35` и как входной параметр T передавать ей интервал вызова блока `ОВ35`. Шаг дискретизации (равный интервалу вызова блока `ОВ35`) можно взять, например, 0.1с .

ЗАДАЧА 17

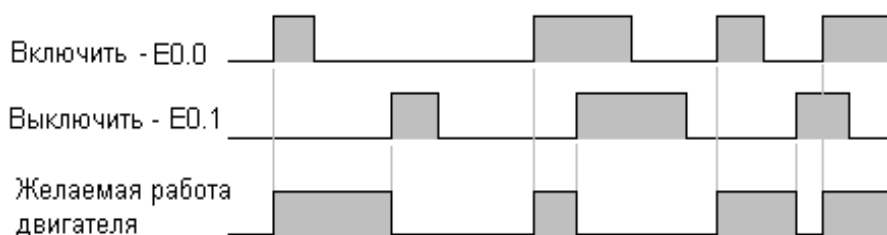
УПРАВЛЕНИЕ ДВИГАТЕЛЕМ

Постановка задачи:

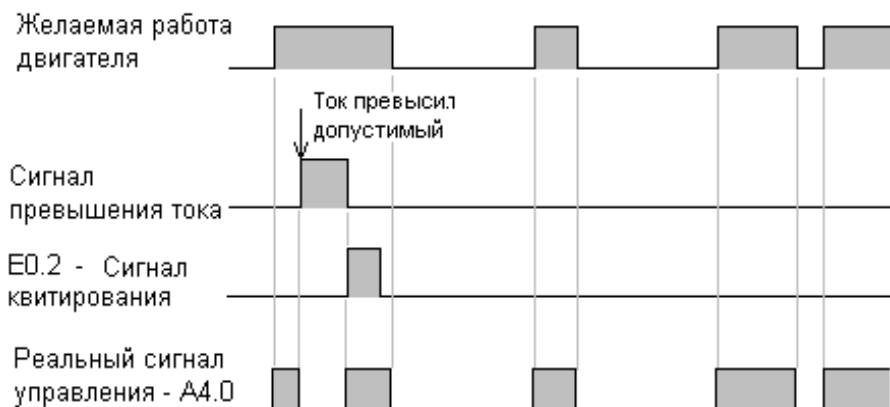
Необходимо написать программу, формирующую сигнал управления двигателем на выходе контроллера А4.0 ("1" - включить, "0" - выключить).

Требования:

1. Необходимо предусмотреть два входа управления двигателем: включение - E0.0 и выключение - E0.1 (обрабатывать фронты этих сигналов).



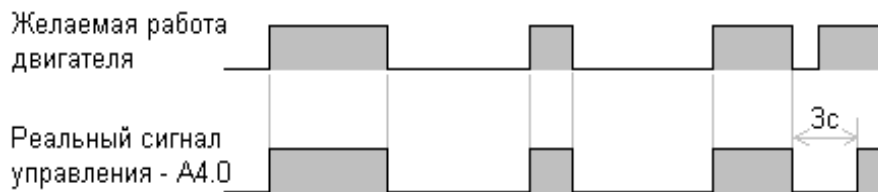
2. Во время работы двигателя необходимо контролировать рабочий ток двигателя и, при превышении допустимого тока, двигатель необходимо выключить (независимо от требуемой команды). Датчик тока подключен к аналоговому входу PEW288.
3. При получении сигнала квитирования превышения тока (вход E0.2), двигатель должен продолжить работу в зависимости от команды со входов.



4. Контролировать рабочий ток двигателя необходимо только после истечения времени переходных процессов. Временем переходных процессов считать

первые 5 секунд после включения двигателя.

5. Необходимо предусмотреть блокировку частых включений двигателя, т.е. после выключения двигатель может вновь включиться не раньше, чем через определенное время, например 3 сек. Если команда на включение двигателя придет до истечения этого времени, сигнал управления двигателем необходимо задержать.



Рекомендации:

Удобнее разрабатывать программу поэтапно, отлаживая работу программы на каждом этапе.

ЗАДАЧА 18

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ЦИФРОВОГО НИЗКОЧАСТОТНОГО ФИЛЬТРА ВТОРОГО ПОРЯДКА

Постановка задачи:

В случаях, если имеет место разделение спектров полезного сигнала и помехи (например, сигнал с датчика-термопары - медленно изменяющийся, а помеха - высокочастотная), то значительное повышение точности измерений может быть достигнуто при использовании цифровых фильтров.

Функционирование фильтра:

Для $f_b \sim 1$ Гц (f_b - верхняя частота среза фильтра) можно рассчитать следующий цифровой фильтр второго порядка (сам расчет фильтра здесь, по понятным причинам, не приводится):

$$y_k = -0.92774 \cdot y_{k-2} + 1.9265 \cdot y_{k-1} + 0.0006 \cdot x_k + 0.00064 \cdot x_{k-1}, \quad (1)$$

где k - текущий момент времени, $k-1$ - предыдущий, и т.д.;
 x - вход фильтра, y - выход фильтра.

Задание:

Написать функциональный блок FB20, работающий по формуле (1) со следующими входными и выходными параметрами:

Inp: // REAL, вход фильтра
Outp: // REAL, выход фильтра

Функциональный блок FB20 вызывать в OB35 с интервалом 50мс.
Для тестирования в OB35 формировать следующий входной сигнал:

$$x_k = 1 + 0.2 \cdot \sin(10 \cdot k \cdot T),$$

где: 1 - входной полезный сигнал, $0.2 \sin(10kT)$ - помеха,
 $T=0.05$ с - шаг дискретизации, k - дискретные моменты времени.

ЗАДАЧА 19

КОСВЕННАЯ АДРЕСАЦИЯ. УКАЗАТЕЛЬ - СЛОВО

Постановка задачи:

Два входа E0.0 и E0.1 задают номер рабочего блока данных (1, 2, 3 или 4). В зависимости от комбинации на входах открыть соответствующий блок данных и вывести на выходы байт DBB0 этого блока данных.

ЗАДАЧА 20

ПРОЦЕСС СМЕШИВАНИЯ

Косвенная адресация. Указатель - двойное слово.

Постановка задачи:

20.1. Промышленный процесс смешивания организует смешивание двух веществ А и В в определенной пропорции. Для организации смешивания имеется 4 рецепта (соотношения веществ А и В). Номер рецепта задается дискретными входами E0.0 и E0.1.

Рецепт №1: Процентное содержание вещества А содержится в переменной DB1.DBD0, вещества В - в DB1.DBD4.

Рецепт №2: Содержание вещества А - в переменной DB1.DBD8, вещества В - в DB1.DBD12.

Для рецептов №2 и 3 - аналогично.

Программа выполняется один раз по фронту сигнала запуска - вход E0.2, и выполняет следующие действия. В зависимости от состояния входов E0.0 и E0.1 считывает требуемое слово блока данных DB1, задающее процентное содержание вещества А и выводит его в DB1.DBD32, а содержание вещества В по этому рецепту - в DB1.DBD36.

Рекомендации

Считанное значение с дискретных входов (EВ0) необходимо сначала преобразовать в формат 32-разрядного указателя.

20.2. Изменить задачу. Пусть соотношения содержания веществ расположены в блоке данных не последовательно, а в случайном порядке, например:

1-й рецепт: А - в DBD0, В - в DBD12

2-й рецепт: А - в DBD8, В - в DBD16

3-й рецепт: А - в DBD24, В - в DBD4

4-й рецепт: А - в DBD28, В - в DBD20

Рекомендации

Можно использовать команду перехода по списку SPL.

ЗАДАЧА 21

ПОДСЧЕТ КОНТРОЛЬНОЙ СУММЫ

Постановка задачи:

Имеется блок данных с какой-либо информацией в первых N байтах ($N/4$ двойных слов). Необходимо подсчитать контрольную сумму всех двойных слов ("Исключающее ИЛИ") и полученное контрольное слово записать в этот же блок данных после N байт информации.

Рекомендации:

Желательно оформить программу в виде функции с входными формальными параметрами: номер блока данных, количество байт полезной информации (N) и выходным параметром `Bool` типа `BOOL`- ошибка (N не кратно двойному слову).

В теле функции рекомендуется использовать косвенную адресацию и оператор цикла.

ЗАДАЧА 22

ЛИНЕАЛИЗАЦИЯ НЕЛИНЕЙНОЙ ХАРАКТЕРИСТИКИ ТЕРМОПАРЫ ХК68

Постановка задачи:

К аналоговому входу контроллера подключена термопара, имеющая нелинейную характеристику. Характеристика термопары задана таблицей соответствий. Требуется написать программу, которая по измеренному коду, выдаваемому аналоговым датчиком, вычисляет температуру в градусах, учитывая нелинейность характеристики.

Рекомендации:

Нелинейную характеристику термопары целесообразно хранить в виде блока данных. Саму же программу следует написать таким образом, чтобы данный блок можно было вызывать с любым блоком данных (с любой характеристикой нелинейности).