

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)  
«МАИ»**

---

Филиал МАИ «Стрела»

Кафедра № С-15

«Цифровые вычислительные комплексы систем управления  
радиоэлектронным оборудованием»

Утверждено на заседании  
кафедры С-15

Протокол № 9  
от «30» 09. 2018 г.

Варианты заданий и методические указания по выполнению  
лабораторных работ по дисциплине:  
**«Информационное обеспечение цикла жизни электронных средств  
(CALS-технологии)»**

для основной образовательной программы  
«Вычислительные машины, комплексы, системы и сети»

по направлению (специальности) подготовки  
«Информатика и вычислительная техника»

Разработано:  
Профессором Башкировым Л.Г.

Жуковский - 2018г.

## ВВЕДЕНИЕ

Лабораторные работы по дисциплине «**Информационное обеспечение цикла жизни электронных средств (CALS-технологии)**» выполняются с использованием программ-эмуляторов микроконтроллеров КМ1816 ВЕ 48 и КМ 1816 ВЕ 51, компилирование и распечатка листингов разработанных и отлаженных программ может осуществляться как программами эмуляторами, так и стандартными кросс системами. Все программы работают в DOS на компьютерах любой конфигурации.

### ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

#### 1. Подготовка и допуск к работе

1.1. К выполнению лабораторной работы допускаются студенты, которые подготовились к работе и имеют не более двух незащищенных работ.

Перед работой каждая бригада должна:

- предъявить преподавателю полностью оформленный отчет о предыдущей работе;
- ответить на вопросы преподавателя.

К работе не допускаются студенты, которые не выполнили одно из перечисленных требований.

1.2. Лабораторные работы, которые студент пропустил, выполняются в конце семестра;

допуск к работе производится в порядке, указанном в п.1.1.

2. Отчёт по лабораторной работе должен содержать следующее:

- Текст задания.
- Электрическую принципиальную схему подключения датчиков
- Блок-схему алгоритма программы, оформленную в соответствии с ГОСТ 19.701-90 (ИСО 5807-85) .
- Текст программы (листинг ассемблера ASM48 или ASM51).
- Комментарии в тексте программы обяза

### ЛАБОРАТОРНАЯ РАБОТА №1

#### РАЗРАБОТКА ПРОГРАММЫ ОПРОСА ДВОИЧНОГО ДАТЧИКА

Цель работы: изучение основных методов опроса датчиков, имеющих два состояния - “включено” и “выключено”.

Задание: датчик, имеющий два состояния подключен к одному из выводов порта P1

микроконтроллера 1816ВЕ48.

Состояние датчика “включено” соответствует значению логического нуля ТТЛ уровня.

Состояние датчика “выключено” соответствует значению логической единицы ТТЛ уровня.

Принцип работы программы: программа должна быть оформлена в виде

подпрограммы, обращение к которой осуществляется командой CALL. Результатом работы подпрограммы должно быть изменение значения одного разряда зарезервированной для этого ячейки памяти данных микроконтроллера.

### Краткие теоретические сведения

В устройствах и системах логического управления объектами события в объекте управления фиксируются с использованием разнообразных датчиков цифрового и аналогового типа. Наибольшее распространение имеют двоичные датчики типа да/нет, например концевые выключатели, которые подключаются к МК так.

Ожидание статического сигнала. Типовая процедура ожидания события (WAIT) состоит из следующих действий: ввода сигнала от датчика, анализа значения сигнала и передачи управления в зависимости от состояния датчика. Блок-схема алгоритма процедуры ожидания события, фиксируемого замыканием контакта двоичного датчика. Конкретная программная реализация процедуры зависит не только от типа МК, но и от того, каким образом датчик подключен к МК. Он может быть подключен к одной из линий портов МК или к специальным тестируемым входам (T0, T1 для МК48).

Например, при подключении датчика к линии бита 3 порта 1 программа реализации процедуры ожидания замыкания контакта будет иметь вид:

При подключении датчика к тестовому входу T0 микроконтроллера МК48 программа будет иметь вид:

; версия для МК48

WAITC: JT0 WAITC ; если контакт разомкнут, то цикл.

Другим частным случаем типовой процедуры ожидания события является процедура ожидания размыкания контакта, которая может быть реализована следующим образом:

; версия для МК48

WAITO: IN A,P1 ; ввод байта

CPL A ; инвертирование

JB3 WAITO ; если контакт замкнут, то цикл

; версия для МК51

WAITC: JB P1.3,WAITC ; ожидание замыкания

; контакта датчика

Наравне с входами T0 и T1 для опроса датчика может использоваться и вход ЗПР. В этом случае надо предварительно запретить прерывания и использовать вход ЗПР как тестовый.

Режим прерывания целесообразно использовать только для опроса особо важных датчиков с целью уменьшения времени реакции на исключительную (аварийную) ситуацию в объекте управления.

## ЛАБОРАТОРНАЯ РАБОТА № 2

### УСТРАНЕНИЕ ДРЕБЕЗГА КОНТАКТА МЕХАНИЧЕСКОГО ДАТЧИКА

Цель работы: изучение основных способов устранения дребезга механического контакта.

Задание: Датчик, имеющий два состояния, подключен к одному из выводов порта P1 микроконтроллера 1816BE48/51. Состояние “включено” соответствует замкнутому состоянию контакта датчика. Состояние “выключено” соответствует разомкнутому состоянию контакта датчика.

Принцип работы программы:

- фиксируется первое срабатывание датчика;
- после выдержки времени, достаточной для окончания процесса дребезга, проверяется состояние датчика;
- если десятикратно зафиксировано “вкл” состояние датчика, то считается, что датчик сработал;
- если однократно было зафиксировано “выкл” состояние датчика, считается, что было ложное срабатывание датчика.

Программа должна быть оформлена в виде подпрограммы, обращение к которой осуществляется командой CALL.

Результатом работы подпрограммы должно быть изменение значения соответствующего разряда зарезервированной для этой ячейки памяти данных.

### Краткие теоретические сведения

При работе МК с датчиками, имеющими механические или электромеханические контакты (кнопки, клавиши, реле и клавиатуры), возникает явление, называемое дребезгом.

Это явление заключается в том, что при замыкании контактов возможно появление отскока (BOUNCE) контактов, которое приводит к переходному процессу. При этом сигнал с контакта может быть прочитан МК как случайная последовательность нулей и единиц.

Подавить это нежелательное явление можно схемотехническими средствами с использованием буферного триггера, но чаще это делается программным путем.

Наибольшее распространение получили два программных способа ожидания установившегося значения: 1) подсчет заданного числа совпадающих значений сигнала;

2) временная задержка. Схемы процедур подавления помех от дребезга контактов (DEBOUNCE) при вводе сигнала 0. Суть первого способа состоит в многократном считывании сигнала с контакта. Подсчет удачных опросов (т.е. опросов, обнаруживших, что контакт устойчиво замкнут) ведется программным счетчиком. Если после серии удачных опросов встречается неудачный, то подсчет начинается сначала.

Контакт считается устойчиво замкнутым (дребезг устранен), если

последовало N удачных опросов. Число N подбирается экспериментально для каждого типа используемых датчиков и лежит в пределах от 5 до 50.

Пример программного подавления дребезга контакта приводится для случая, когда датчик импульсного сигнала подключен к входу T0, счет удачных опросов ведется в

регистре R3, N=20:

; ВЕРСИЯ ДЛЯ МК48

DBNC: MOV R3, # 20 ; инициализация счетчика

DBNC1: JTO DBNC ; если контакт разомкнут, ; то начать отсчет опросов ;

Сначала DJNZ R3, DBNC1 ; декремент счетчика, и ; если содержимое

; счетчика не равно 0, то

; повторить анализ

; состояния контакта

; ВЕРСИЯ ДЛЯ МК51

DBNC: MOV R3, #20 ; инициализация счетчика

DBNC1 JB P3.4, DBNC ; если контакт разомкнут,

; то начать отсчет опросов

; сначала

DJNZ R3, DBNC1 ; повторять, пока R3 не

; станет равным 0

Устранение дребезга контакта путем введения временной задержки заключается в следующем. Программа, обнаружив замыкание контакта К, запрещает опрос состояния этого контакта на время, заведомо большее длительности переходного процесса. Программа, соответствующая блок-схеме алгоритма (БСА), написана для случая подключения датчика к входу T0 и программной реализации временной задержки:

; ВЕРСИЯ ДЛЯ МК48

DBNCDL: JTO DBNCDL ; ожидание нуля на входе T0

CALL DELAY ; вызов подпрограммы задержки

EXIT: ...

Временная задержка (в пределах 1-20 мс) подбирается экспериментально для каждого типа датчиков и реализуется подпрограммой DELAY.

### ЛАБОРАТОРНАЯ РАБОТА № 3

#### РАЗРАБОТКА ПРОГРАММЫ ПОДСЧЁТА ЧИСЛА СРАБАТЫВАНИЙ МЕХАНИЧЕСКОГО ДАТЧИКА

Цель работы: освоить основные методы анализа состояния импульсных механических датчиков.

Задание: Механический датчик, имеющий два состояния, подключён к одному из выводов порта P1 микроконтроллера 1816 ВЕ 51. Состояние “включено” датчика соответствует замкнутому состоянию контакта. Состояние “выключено” соответствует разомкнутому состоянию контакта.

Принцип работы программы:

а) фиксируется первое срабатывание датчика;

- б) после выдержки времени, достаточного для окончания процесса дребезга, проверяется состояние датчика;
- в) если десятикратно зафиксировано нажатое состояние датчика, то считается, что датчик сработал;
- г) если было зафиксировано отжатое состояние датчика, то считается, что было ложное срабатывание датчика;
- д) фиксируется первое разомкнутое состояние датчика;
- е) после выдержки времени, достаточного для окончания процесса дребезга, проверяется состояние датчика;
- ж) если десятикратно зафиксировано отжатое состояние датчика, то считается, что датчик сработал;
- з) если было зафиксировано нажатое состояние датчика, то считается, что было ложное срабатывание датчика;
- и) в случае правильного срабатывания датчика, увеличивается содержимое счетчика числа срабатываний;
- к) время счета задается счетчиком-таймером и составляет 1 сек.;
- л) счетчик-таймер загружается перед началом опроса датчика.

#### Краткие теоретические сведения

Ожидание импульсного сигнала. Схема подключения датчика импульсного сигнала аналогична ранее представленной схеме. Особенность процедуры ожидания импульсного сигнала состоит в том, что МК должен обнаружить не только факт появления, но и факт окончания сигнала.

Оформлять процедуры WAITC и WAITO в виде подпрограмм нецелесообразно, так как это удлиняет программу, а длина и, следовательно, время исполнения программы определяют минимальную длительность импульса, который может быть обнаружен программой.

Последовательность склеивания процедур WAITC и WAITO зависит от формы импульса. Для “отрицательного” импульса (1-0-1) процедура WAITC предшествует процедуре WAITO, для “положительного” (0-1-0) следует за ней.

Ниже приведены примеры программной реализации процедуры ожидания “отрицательного” импульсного сигнала при подключении датчика к биту 3 порта 1 при условии, что начальное состояние входа - единичное:

;Версия для МК48

WAITC: IN A,P1 ; ввод байта

JB3 WAITC ; если P1.3=1, то ждать

WAITO: IN A,P1 ; ввод байта

CPL A ; инверсия

JB3 WAITO ; если P1.3=0, то ждать

;Версия для МК51

WAITC: JB WAITC ; ожидание появления импульса

WAITO: JNTO WAITO ; ожидание окончания импульса

Аналогичным образом строится программа при подключении датчика импульсного “отрицательного” сигнала к тестовому входу МК48:

;Версия для МК48

WAITC: JTO WAITC ; ожидание появления импульса

WAITO: JNTO WAITO ; ожидание окончания импульса

Программная реализация цикла ожидания накладывает ограничения на длительность импульса: импульсы длительностью меньше времени выполнения цикла ожидания могут быть “не замечены” МК. Минимально допустимые длительности импульсов для различных способов подключения импульсного датчика к МК.

Для обнаружения кратковременных импульсов можно использовать способ фиксации импульса на внешнем триггере флага.

На вход МК в этом случае поступает не кратковременный сигнал с датчика, а флаг, формируемый триггером. Триггер устанавливается по фронту импульса, а сбрасывается программным путем - выдачей специального управляющего воздействия. Длительность импульса при этом будет ограничена снизу только быстроедействием триггера.

Текст программы для МК48 приведен ниже:

FLAGIN: ; процедура ввода флага в МК48

ANL P2,#0FЕН ; сброс флага

ORL P2,#01Н

WAITC: JTO WAITC ; ожидание прихода импульса

Часто в управляющих программах возникает необходимость ожидания цепочки событий, представляемой последовательностью импульсных сигналов от датчиков.

Рассмотрим две типовые процедуры: подсчёт числа импульсов между двумя событиями и подсчёт числа импульсов за заданный интервал времени.

Подсчёт числа импульсов между двумя событиями. Эту типовую процедуру удобно проиллюстрировать на конкретном примере. Предположим, что необходимо подсчитать число деталей, сошедших с конвейера от момента его включения до момента выключения. Факт схода детали с конвейера фиксируется фотоэлементом, на выходе которого формируется импульсный сигнал.

Для простоты реализации программы считаем, что общее количество деталей не превышает 99:

; версия для МК48 count: clr

## ЛАБОРАТОРНАЯ РАБОТА № 4

### ФОРМИРОВАНИЕ СИГНАЛОВ УПРАВЛЕНИЯ ИСПОЛНИТЕЛЬНЫМИ МЕХАНИЗМАМИ

Цель работы: изучение основных методов формирования управляющих сигналов для исполнительных механизмов.

Задание: к выводам портов P1 и P2 подключены исполнительные механизмы.

Текущее состояние исполнительных механизмов зафиксировано в текущем слове состояния системы. Внешняя программа может изменить в

произвольные моменты времени эталонное слово состояния системы.

Принцип работы программы следующий:

- периодически, с заданным периодом, программа опрашивает эталонное слово состояния (то, которое нужно установить) системы и сравнивает его с текущим словом состояния системы,

- при обнаружении несовпадения содержимого эталонного и текущего слов состояния, программа изменяет состояние соответствующих линий порта, начиная с младших,

- каждое изменение состояния исполнительных механизмов фиксируется в текущем

слове состояния системы,

- период опроса - 100 ms.

Программа должна быть оформлена в виде подпрограммы, обращение к которой осуществляется командой CALL.

Краткие теоретические сведения Формирование статических сигналов. Для управления исполнительным механизмом,

работающим по принципу включено/выключено, на соответствующей входной линии порта МК необходимо сформировать статический сигнал 0 или 1, что реализуется командами вывода непосредственного операнда, содержащего в требуемом бите значение 0 или 1.

В случае параллельного управления группой автономных исполнительных механизмов, подключенных к выходному порту, формируется не двоичное управляющее воздействие, а управляющее слово (УС), имеющее формат байта, каждому разряду которого ставится в соответствие 1 или 0 в зависимости от того, какие исполнительные механизмы должны быть включены, а какие выключены.

Управляющие слова удобно формировать командами логических операций над содержимым порта. Команда ANL используется для сброса тех бит УС, которые в операнде (маске) заданы нулем. Команда ORL используется для установки бит УС. Командой XRL осуществляется инверсия бит в соответствии с выражением  $x + 1 = \bar{x}$ .

Для формирования сложных последовательностей УС удобно пользоваться табличным способом, при котором все возможные УС упакованы в таблицу, а прикладная программа МК вычисляет адрес требуемого УС выбирает его из таблицы и передает в порт вывода.

Формирование импульсных сигналов. Управляющее воздействие типа "импульс" можно получить последовательной выдачей сигналов включить или отключить с промежуточным вызовом подпрограммы временной задержки:

PULS: ;ВЫДАЧА ИМПУЛЬСА В ЛИНИЮ 3

;ПОРТА 1 МК 48

ON: ANL P1,#11110111B ;ВКЛЮЧЕНИЕ ИСПОЛНИТЕЛЬНОГО

;МЕХАНИЗМА

CALL DELAY ;ВРЕМЕННАЯ ЗАДЕРЖКА

OFF: ORL P1,#00001000B ;

## ОТКЛЮЧЕНИЕ ИСПОЛНИТЕЛЬНОГО ;МЕХАНИЗМА

Длительность импульса определяется временной задержкой, реализуемой подпрограммой DELAY.

### ЛАБОРАТОРНАЯ РАБОТА № 5

#### ФОРМИРОВАНИЕ ИМПУЛЬСНЫХ СИГНАЛОВ

Цель работы: освоить основные методы программной генерации динамических сигналов различного типа.

Задание:

1. Сформировать на одном из выводов порта P2 микроконтроллера 1816BE48 и 1816BE51 одиночный импульсный сигнал, длительность которого задается содержимым одного из регистров Ri

2. Сформировать на одном из выходов порта P2 микроконтроллера 1816BE48 и 1816BE51 периодический импульсный сигнал, период которого и скважность задаются содержимым регистров Ri.

Принцип работы программы:

- программа должна быть оформлена в виде подпрограммы, обращение к которой осуществляется командой CALL.
- внешняя программа в случайные моменты времени заносит в регистры Ri микроконтроллера значения, определяющие длительность импульса, период следования сигнала и его скважность. Разрабатываемая подпрограмма должна использовать эти значения для формирования сигналов на выходах портов микроконтроллеров.

#### Краткие теоретические сведения

Управляющее воздействие типа "импульс" можно получить последовательной выдачей сигналов включить и отключить с промежуточным вызовом подпрограммы временной задержки:

; Версия для МК 48

PULS: ; ВЫДАЧА ИМПУЛЬСА В ЛИНИЮ 3 ПОРТА 1

ON: ANL P1,11110111B ;ВКЛЮЧЕНИЕ ИСПОЛНИТЕЛЬНОГО  
;МЕХАНИЗМА

CALL DELAY ;ВРЕМЕННАЯ ЗАДЕРЖКА

OFF: ORL P1,00001000B ;ОТКЛЮЧЕНИЕ ИСПОЛНИТЕЛЬНОГО  
;МЕХАНИЗМА

...

Длительность импульса определяется временной задержкой, реализуемой подпрограммой DELAY.

Генерация периодического управляющего воздействия (меандра). Для генерации меандра удобно воспользоваться процедурой выдачи импульсного управляющего воздействия (PULS) и подпрограммой реализации временной задержки, равной половине периода сигнала (DLYX).

; Версия для МК 48

MAEANDR:

ON: ANL P1,0F7H  
CALL DLYX  
OFF: ORL P1,04H  
CALL DLYX  
JMP ON

; Версия для МК 51

MEANDR:

XCOR: CPL P1.3

ACALL DLYX

SJMP XCOR

; Версия для МК 48

DLYX: MOV R2,#X ; (R2) Å X

COUNT: DEC R2 ; ДЕКРЕМЕНТ R2 И

DJNZ R2,COUNT ; ЦИКЛ , ЕСЛИ НЕ НУЛЬ

RET ; ВОЗВРАТ

Бесконечный периодический сигнал формируется в линии 3 порта 1; на остальных линиях порта 1 сигналы остаются неизменными.

Формирование аperiodического управляющего сигнала. Последовательность импульсных сигналов с произвольной длительностью и скважностью может быть получена аналогичным образом, т.е. путем чередования процедур выдачи изменяемого значения сигнала (0 или 1) и вызова подпрограмм временных задержек заданных длительностей.

## ЛАБОРАТОРНАЯ РАБОТА № 6

### РАЗРАБОТКА ПРОГРАММЫ УПРАВЛЕНИЯ ГРУППОЙ

#### СЕМИСЕКМЕНТНЫХ ИНДИКАТОРОВ В ДИНАМИЧЕСКОМ РЕЖИМЕ

Задание: Семисегментные индикаторы объединены в группу с общим анодом или общим катодом. Управление индикаторами осуществляется через порты P1 и P2 с соответствующими усилителями.

Принцип работы программы следующий:

- коды отображаемых цифр хранятся в заданной области памяти данных;
- программа периодически, с периодом достаточным для получения приемлемой яркости отображаемых цифр, обновляет информацию на линиях порта;
- коды отображения цифр должны храниться в памяти программ;
- обновление информации на линиях портов по прерываниям от таймера;
- количество отображаемых десятичных цифр - 5.

Программа должна быть оформлена в виде подпрограммы, инициализация которой осуществляется при первом обращении к ней командой CALL.

Должна быть предусмотрена возможность прекращения работы программы с выключением всех индикаторов в произвольный момент времени.

## Краткие теоретические сведения

Индикаторы. Многие МК-устройства требуют наличия только простейшей индикации типа ДА/НЕТ, ВКЛ/ВЫКЛ. Такая индикация реализуется на основе отдельных светодиодов.

Семисегментные индикаторы (ССИ) широко используются для отображения цифровой и буквенной информации. Семь отображающих элементов позволяют высвечивать десятичные и шестнадцатиричные цифры, некоторые буквы русского и латинского алфавитов, а также некоторые специальные знаки. Структура ССИ и способы его подключения к МК показаны на рис. 10. Для засветки одного сегмента большинства типов ССИ необходимо обеспечить протекание через сегмент тока 10-15 мА при напряжении 2,0-2,5 В. Низкая нагрузочная способность МК не допускает прямого соединения с ССИ. В качестве промежуточных усилителей тока могут использоваться логические элементы серии К155 или интегральные схемы преобразователей кодов для управления ССИ.

Преобразование двоичных кодов в коды для ССИ может осуществляться либо программно, либо аппаратно с использованием преобразователей К514ИД1, К514ИД2, 133ПП4, 564ИД5.

Матричные светодиодные индикаторы (МСИ) используются для отображения алфавитно-цифровой информации. Каждый из таких МСИ, выполненный в виде интегральной микросхемы, представляет собой матрицу светодиодов размерностью  $m \times n$ , где  $n$ -число колонок,  $m$ -число строк матрицы. Наибольшее распространение получили МСИ с размерностью матрицы  $7 \times 5$  и  $9 \times 7$ .

Для включения одного светодиода матрицы необходимо обеспечить протекание через него тока 10-15 мА при напряжении 2,0-2,5В. Подключение матричного индикатора к МК осуществляется через управляемые схемы формирования тока колонок и строк.

Для отображения многосимвольной информации используются линейные (однострочные) дисплеи. Такие дисплеи представляют собой “линейку”, смонтированную из отдельных ССИ или МСИ. Число знакомест дисплея определяется в соответствии с требованиями к МК-системе.

Существует два способа организации интерфейса МК с линейным дисплеем: статический и динамический.

Первый требует наличия на входах каждого индикатора специальных буферных регистров для хранения кодов выводимых символов. Естественно, что с увеличением разрядности дисплея возрастает число дополнительных микросхем, а следовательно, и стоимость МК-системы.

Второй способ (динамический) основан на том, что любой световой индикатор является инерционным прибором, а человеческому глазу отображаемая на дисплее информация, если ее обновлять с частотой примерно 20 раз в секунду, представляется неизменяемой.

Динамический способ вывода информации на дисплей требует значительно меньших аппаратных затрат, но более сложного программного обеспечения.

Именно этот способ организации вывода информации получил преимущественное распространение в МК-системах.