Министерство образования РБ

Белорусский государственный университет информатики и радиоэлектроники

**Кафедра ЭВС**

К защите допускаю

“ “ \_\_\_\_\_\_\_\_\_ 2001 г.

Руководитель работы

Давыдов А.Б.

Пояснительная записка

к курсовому проекту на тему:

“Цифровой тахометр”

Выполнил: Проверил:

студент гр. 810702

Трифонов А.А. Давыдов А.Б.

Минск 2001 г.

**Содержание**

Введение

1. Анализ задачи

1.1 Функциональная спецификация системы

1.2 Список функций, выполняемых системой

1.3 Описание интерфейса между системой и пользователем

2. Предварительное проектирование системы

2.1 Разбиение системы на модули

2.2 Выбор соотношения между аппаратными и программными средствами

3. Проектирование аппаратных средств системы

3.1 Разработка принципиальной схемы системы

3.2 Описание работы тахометра

4. Проектирование программного обеспечения

4.1 Разработка схемы алгоритма работы системы и программы

4.2 Описание алгоритма работы системы и программы

Заключение

Литература

Приложение А — Листинг управляющей программы

**Введение**

Современный этап развития научно-технического прогресса характеризуется широким применением электроники и микроэлектроники во всех сферах жизни и деятельности человека. Важную роль при этом сыграло появление и быстрое совершенствование интегральных микросхем – основной элементной базы современной электроники. Интегральные микросхемы применяются в вычислительных машинах и комплексах, в электронных устройствах автоматики, цифровых измерительных приборах, аппаратуре связи и передачи данных, медицинской и бытовой аппаратуре, в приборах и оборудовании для научных исследований и т.д.

Микрокомпьютер, в отличие от других компонентов, не обладает фиксированным набором функциональных характеристик. Его характеристики определяются в процессе проектирования системы с помощью программирования. Практически неограниченный диапазон программируемых функциональных возможностей микрокомпьютера придает этому компоненту особое значение.

В данном курсовом проекте необходимо разработать цифровой тахометр на базе микропроцессора 8086. Цифровой тахометр должен выполнять следующие действия:

1) Обеспечить измерение частоты вращения ротора двигателя, которая может изменятся от 100 до 5000 об/мин.;

2) Допустимая погрешность измерения в заданном диапазоне не более 2%;

3) Импульсный датчик выбрасывает 10 импульсов на каждый оборот.

4) Быстродействие системы должно быть таково, чтобы время измерения не превышало двух оборотов.

5) Измеренное число оборотов выводится на индикатор.

Эти требования должны лечь в основу проектируемого устройства.

**1. Анали****з задачи**

**1.1 Функциональная** **спецификация системы**

Функциональная спецификация фиксирует, что система должна сделать и как она взаимодействует с окружением.

Функциональная спецификация включает два основных компонента:

1)список функций выполняемых системой;

2)описание интерфейса между системой и пользователем.

Список функций выполняемых системой и интерфейс между системой и пользователем определяются требованиями, которые предъявляются к проектируемой системе потребителем или пользователем, будущих эксплуатировать устройство. Исходя из заданных данных на курсовой проект можно выделить следующие основные функциональные действия, которые должна выполнять проектируемая компьютерная система:

**Обеспечить измерение частоты вращения ротора двигателя, которая может изменятся от 100 до 5000 об/мин ;**

Данную функцию можно реализовать путем преобразования импульсов за каждый оборот, которые снимаются с помощью импульсного датчика, в цифровой код с помощью биполярного аналого-цифрового преобразователя (АЦП) с дальнейшей программной обработкой в микропроцессорной системе и передачей полученных данных на цифровой индикатор.

**обеспечивать точность измерения напряжений 2% на заданном диапазоне;**Данное требование в проектируемом устройстве обеспечивается путем применения радиоэлементов с отклонением от номинала не более 0.5% и АЦП соответствующей разрядности.

**быть несложным в управлении;**

Несложность в управлении достигается минимизацией числа органов управления и наглядным отображением измеряемой величины. В проектируемом устройстве в качестве органов управления применяются только кнопка «POWER». Отображения информации осуществляется при помощи четырех индикаторов, на которых выводится значение измеряемой величины.

**1.2 Список функций, выполняемых системой**

В результате анализа функциональных действий можно определить основные функции, выполняемые системой:

преобразование аналогового сигнала в цифровой код;

считывание и запись данных в порты;

обработка полученных данных (определения порядка, знака и размерности измеряемой величины);

преобразование вводимого двоичного кода в двоично-десятичный;

преобразование цифрового кода в символьное значение для индикации;

вывод проанализированных данных на индикатор.

**1.3 Описание интерфейса между системой и пользователем**

На основании списка функций выполняемых системой определим интерфейс между системой и пользователем.

Внешний вид передней панели цифрового тахометра представлен на рисунке 1.

POW

ER

**Рисунок 1** - Внешний вид передней панели тахометра

На передней панели расположены четыре цифровых индикатора, на которых выводится значения измеряемой величины.

Включения тахометра производят клавишей «POWER»

**2.** **Предварительное проектирование системы**

**2.1 Разбиение системы на модули**

Из выше перечисленных функций следует, что в состав проектируемого цифрового тахометра должны войти следующие модули.

***Модуль аналого-цифрового преобразователя***, который будет выполнять функцию преобразования импульсов в цифровой код.

Преобразования двоичного кода в двоично-десятичный будет, осуществляется при помощи ***модуля преобразования в двоично-десятичный код***.

***Модуль индикации*** необходим для отображения значения измеряемой частоты.

Обработка данных вводимых из АЦП (выделение знака, определение размерности) осуществляется ***модулем обработки***.

Преобразование значений импульсов в символьное значение для знакоразрядного индикатора осуществляется ***модулем преобразования в семисегментный код***.

Управление работой тахометра в целом будет, осуществляется при помощи **микрокомпьютера**. Так как микрокомпьютер, является последовательным устройством, которое выполняет одну команду за другой, в модульный состав необходимо включить ***исполнительный модуль***, который будет обеспечивать последовательное исполнение системой функций.

**2.2 Выбор соотношения между аппаратными и программными средствами**

В результате анализа функций, выполняемых системой, и на основании модульной структуры произведем разбиение системы на аппаратные и программные модулями. Полученная модульная структура цифрового тахометра представлена на рисунке 2.

### Программные средства цифрового тахометра

***Исполнительный модуль***

***Модуль выбора диапазона***

модуль преобразования в двоично-десятичный код

***Модуль обработки***

***Модуль преобразования в семи-сегментный код***

***Модуль аналого-цифрового преобразователя***

***Микрокомпьютер***

***Модуль индикатора***

### Аппаратные средства цифрового тахометра

**Рисунок 2** – Соотношения между программными аппаратными модулями в цифровом тахометре

Связь между программными модулями и модулем микрокомпьютера показана двойной стрелкой, так как программные модули реализуются как процедуры микрокомпьютера.

Таким образом, программным методом реализована следующая группа функций: преобразование двоичного кода в двоично-десятичный, обработка данных.

Из выше перечисленных, реализованных программным методом, функций две (преобразование двоичного кода в двоично-десятичный), возможно, было реализовать и аппаратно. Однако они реализованы программно по следующим причинам:

решение этих задач не критично по времени;

реализация данных функции аппаратно сложная задача и требует включения в состав устройства большого количества логических ИМС.

Аппаратным методом реализованы модули аналого-цифрового преобразователя, масштабирующего устройства, модуль переключателя, модуль индикации и преобразователь в семисегментный код.

Модуль преобразователя в семисегментный код строится аппаратно так как его программная реализация потребовало бы вывода двадцатичетырех разрядного слова вместо шестнадцати, что не приведет к уменьшению аппаратных затрат.

Также в состав аппаратных средств вошел микрокомпьютер, который содержит в себе необходимые элементы, позволяющие реализовать все необходимые программные действия системы.

**3. Проектирование аппаратных средств системы**

**3.1 Разработка принципиальной схемы системы**

Основу проектируемого устройства составляет центральный процессор (ЦП). В разрабатываемом цифровом тахометре ЦП строится на основе микропроцессора 8086. При разработке структуры блока ЦП возникают задачи разделения (демультиплексирования) шины адреса/данных буферирования шин адреса и шин данных, а также выработка тактовых импульсов и синхронизация сигналов REDY и RESET с тактовыми сигналами микропроцессора.

Первая задача решается с помощью БИС КР580ИР82, выполняющих функции адресной защелки. Поэтому на принципиальной схеме два 8 – битовых регистра КР580ИР82 (DD12 и DD13) обеспечивают запоминание 11 разрядов адреса.

Вторая задача решается с помощью двух 8 – битовых шинных формирователей КР580ВА86 (DD14 и DD15), которые усиливают сигналы системной шины.

Выработка тактовых импульсов и синхронизация сигналов REDY и RESET с тактовыми сигналами микропроцессора выполняется генератором тактовых импульсов (ГТИ) КР1810ГФ84. Ко входом X1 и X2 БИС КР1810ГФ84 (DD6) подключается кварцевый резонатор РК374 (ZQ1) с частотой колебаний – 15 МГц.

На вход RDY БИС КР1810ГФ84 поступает сигнал готовности от блока АЦП.

Сброс МП осуществляется сигналом RESET, который вырабатывается в момент включения питания. Выработка сигнала RESET осуществляется RC цепочкой построенной на элементах C1 и R8. Выбор номиналов элементов C1 и R8 производят исходя из того, что минимальная продолжительность сигнала RESET, при первом включении МП должна составлять не менее 50 мкс. При номиналах C1 = 1мкф и R8 = 200кОм получим длительность сигнала сброса равную:

( 1 )



где *Ucc -* напряжение источника питания;

*Uh -* напряжение логической единицы на входе RES.

Из формулы ( 1 ) видно, что при выбранных номиналах элементов достигается необходимая длительность сигнала сброса.

В разрабатываемой системе блок ПЗУ построен из двух включенных параллельно БИС КР556РТ6 (DD16 и DD19) емкостью 2 Кбайта каждая. Адресные входы А0 – А10 каждой бис соединены параллельно и подключены к адресным линиям А0 – А10. Процесс обращения к ПЗУ стробируется сигналами M/IO и RD подаваемыми на входы CS БИС. Выдача данных на шину микропроцессора после установки адресов микросхемой памяти осуществится не позднее чем, через 80 нс. Так как в МП 8086 длительность цикла чтения длится дольше, то выбранные для построения памяти микросхемы смогут передавать информацию с максимальной пропускной способностью шины.

Для преобразования импульса в код используется микросхема АЦП КР1113ПВ1 (DD3). Микросхема КР1113ПВ1 – это биполярный, десяти разрядный АЦП, с динамическим диапазонам от –5.5В до +5.5В и выдачей данных в прямом коде.

Данная микросхема имеет внутренний буфер, что позволяет подключать ее непосредственно к шине данных. Управления началом преобразования и выдачей данных на шину осуществляется логической схемой собранной на элементах DD4, DD5, DD8 и DD11. Если микропроцессор не обращается к АЦП, то низким уровнем на входе логического элемента *И* (DD8) триггер готовности (DD11) удерживается в единичном состоянии. При этом на входе RDY БИС КР1810ГФ84 присутствует высокий уровень напряжения (устройство готово), а выходы данных АЦП находятся в третьем состоянии. При обращении к АЦП на одном входе элемента *И* будет присутствовать высокий уровень напряжения, а состояния на другом будет определятся сигналом готовности АЦП. Если данные в АЦП не готовы, то высоким уровнем на входе логического элемента *И* триггер готовности устанавливается в нулевое состоянии. При этом на входе RDY БИС КР1810ГФ84 присутствует низкий уровень напряжения (устройство не готово), а выходы данных АЦП по прежнему находятся в третьем состоянии. Если же данные в АЦП готовы, то низким уровнем на входе логического элемента *И* триггер готовности устанавливается в единичное состояние. При этом на входе RDY БИС КР1810ГФ84 появляется высокий уровень напряжения (устройство готово), а на шину АЦП помещает данные.

В микросхеме КР1113ПВ1 младший значащий разряд (МЗР) соответствует напряжению:

( 2 )



При обработке данных использовать данное значение неудобно, поэтому для того чтобы довести это значение до величины 10мВ напряжение на вход АЦП подается через усилитель постоянного тока с коэффициентом передачи равным:

( 3 )



Точное значение коэффициента передачи устанавливается при помощи построечного резистора R15 в цепи обратной связи.

Коэффициент передачи инвертирующего операционного устройства определяется следующим образом:

( 4 )



Из формулы ( 4 ) видно, что изменения коэффициента передачи может быть осуществлено изменением сопротивления в цепи обратной связи. Его, легко изменять параллельным подключением различного числа резисторов в цепи обратной связи.

Подключение и отключение резисторов осуществляется, ключами DD2.1- DD2.3. Управление состоянием ключа (открыт, закрыт), осуществляется уровнем напряжения на выходе регистра DD1. При высоком уровне напряжения на одном из выходов регистра отпирается ключ, подключенный к этому выводу. Данные в регистр заносятся микропроцессором, командой вывода в порт по адресу 01h. Таким образом, значения коэффициента передачи будет определятся выражением:

( 5 )



Так как в разрабатываемом вольтметре необходимы следующие значения коэффициента передачи 1, то принимая значения R0 = 90 кОм по формуле (5) получим следующие значения номиналов резисторов:

R1 = 1 МОм.

Информация об измеряемой частоте (значение) отображается на четырех индикаторах (HG1, HG2, HG3 и HG4). Для хранения отображаемых данных до поступления новых используются два регистра КР1533ИР27 (DD17 и DD18). Процесс записи данных в регистры КР1533ИР27 стробируется сигналами M/IO и WR, подаваемые на входы PE и C. В разряды с нулевого по двенадцатый регистров DD17 и DD18 в двоично-десятичном коде микропроцессором записывается значение измеренного напряжения. Эти данные с выходов регистров поступают на дешифраторы двоично-десятичного кода в семисегментный.

Индикаторы HG1 - HG4 подключаются к выводам микросхем через ограничительные резисторы. Номинал резисторов рассчитываются из выражения:

( 6 )



где *Ucc* – напряжения источника питания;

*Uпр* – напряжение на светодиоде матричного индикатора;

*U0вых* – напряжение логического нуля на выходе ИМС;

*I* – ток протекающий через светодиод матричного индикатора.

**3.2 Описание работы тахометра**

После включения питания микропроцессор считывает поступающие импульсы, устанавливая на выходах логические уровни. В регистр DD1 записывается соответствующее значение. И на вход АЦП поступает некоторое количество импульсов. После окончания преобразования импульсов в код, данные из АЦП по шине данных поступают в микропроцессор, где они преобразуются из двоичного кода в двоично-десятичный. Полученный код МП записывает в регистры DD17 и DD18.

Значение измеренной частоты после преобразование в семисегментный код дешифраторами DD20 - DD22 отображается на индикаторах HG1-HG4.

**4. Проектирования программного обеспечения**

**4.1 Разработка схемы алгоритма работы системы и программы**

С точки зрения программиста разработанный цифровой тахометр, который состоит из программно доступных регистров микропроцессора и двух портов: порт АЦП и порт индикатора. И вся работа системы заключается в вводе данных из АЦП и их обработка (перевод в двоично-десятичный код), и выводе данных на индикатор.

Прежде чем приступить к разработке программы выполняющей перечисленные действия необходимо определить форматы данных вводимых и выводимых в порты.

**4.2 Описание алгоритма работы системы и программы**

Разработанный алгоритм работы системы представлен на чертеже БГУИ.ХХХХХХ.002.

**000h**

**089h**

##### УПРАВЛЯЮЩАЯ

##### ПРОГРАММА

##### РЕЗЕРВ

##### УПРАВЛЯЮЩАЯ

##### ПРОГРАММА

**7F0h**

**7FFh**

Рисунок 3 – Размещение программы в адресном пространстве микропроцессорной системы

При включении тахометра выполняет начальные установки режимов работы и регистров, ждет прерывания от таймера и сигнального входа. Установки портов - индикатор погашен. Обнуляет счетчик оборотов и таймер 1 секунда. Снимает показания счетчика оборотов. Увеличивается содержимое таймера на 1, следит за реальным временем и записывает результат в буфер индикации. Производит счет входных импульсов + нормировку для обеспечения показаний в оборотах/мин, счет ведется в двоично-десятичном коде, обнуление счетчика при включении и по таймеру после считывания результата.

**Заключение**

Входе выполнения курсовой работы был разработан цифровой тахметр с автоматическим выбором диапазона измеряемого напряжения.

В пояснительной записке проработаны вопросы соотношения аппаратных и программных средств разрабатываемой системы, разработана и описана структурная и принципиальная схема устройства. Бес внимания не остались вопросы разработки программного обеспечения. В курсовой работе разработан и описан алгоритм функционирования цифрового тахометра. По разработанному алгоритму составлена программа.

Конечным результатом проведенной работы, явилась структурная и принципиальная схема цифрового тахометра, а также алгоритм функционирования системы и листинг программы.

**Литература**

1. Абрайтис В. -Б. Б., Аверьянов Н. Н., Белоус А. И. и др. Под ред. Шахнова В. А. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: справочник. В 2 т. – М.: Радио и связь, 1988.
2. Богданович М. И., Грель И. Н., Прохоренко В. А., Шалимо В. В. Цифровые интегральные микросхемы: справочник. – Мн.: Беларусь, 1991.
3. Каган Б. М., Сташин В. В. Основы проектирования микропроцессорных устройств автоматики. – М.: Энергоатомиздат, 1987.
4. Тарабрин Б. В., Лунин Л. Ф., Смирнов Ю. Н и др. Под ред. Тарабрина Б. В. Интегральные микросхемы: справочник. – М.: Радио и связь, 1984.
5. Фридмен М., Ивенс Л. Проектирование систем с микрокомпьютерами: пер. с англ. – М.: Мир, 1986.

**Приложение А**

Листинг управляющей программы

;\*\*\*\*\*\*Стандартная шапка\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

list p=16f84, f=inhx8m; \*

#include p16f84.inc; \*

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;-------буфер индикатора-----------

INR EQU H'0C' ; номер текущего разряда

IKANB EQU H'0D' ; косвен. адрес начала буфера

IR0 EQU H'0D' ; разряд 0

IR1 EQU H'0E' ; разряд 1

IR2 EQU H'0F' ; разряд 2

IR3 EQU H'10' ; разряд 3

;--------таймер---------------------------------

TIM0 EQU H'11' ; младший байт

TIM1 EQU H'12' ; старший байт

TIMF EQU H'00' ; флаг переполнения таймера

;-счетчик входных импульсов - работа по прерыванию---

OB0 EQU H'13' ;

OB1 EQU H'14' ;

DOB EQU H'15' ; приращение счетчика импусов

;---------------------------------------------

org 0

goto Begin

org 4

goto Int

;----------------подпрограммы------------------------

Table1

;-перекодировка двоично-десятичного числа

; в семисегментный код

; перед вызовом программы в W загрузить преобразуемое число

; результат после выполнения программы - в W

addwf PCL

retlw H'7F' ; 0

retlw H'0D' ; 1

retlw H'B7' ; 2

retlw H'9F' ; 3

retlw H'CD' ; 4

retlw H'DB' ; 5

retlw H'FB' ; 6

retlw H'0F' ; 7

retlw H'FF' ; 8

retlw H'DF' ; 9

retlw H'01' ; пусто

;-----------------------------------------------

Table2

;-сканирование разряда индикатора

; перед вызовом программы в W загрузить текущий номер раз-да

; результат после выполнения программы - в W

addwf PCL

retlw H'0E' ; 0

retlw H'0D' ; 1

retlw H'0B' ; 2

retlw H'07' ; 3

;----------------------------------------------------

Indic

;-обслуживание индикатора------

; коррекция TMR0

; при обращении преобразует число текущего разряда

; в семисегментный код, и производит его подсветку

; затем производит смещение указателя

; текущего разряда на 1 в пределах 0...3

movlw 0xF1

movwf TMR0 ; коррекция TMR0

bcf STATUS,RP0 ; банк 0

movlw 0xFF ; погасить индикатор

movwf PORTA ;

movlw IKANB

addwf INR,0 ; вычисление косвенного адреса разряда

movwf FSR

movf INDF,0 ; код числа разряда - в W

call Table1

movwf PORTB ; семисегментный код - на индикатор

movf INR,0

call Table2

movwf PORTA ; подсветка разряда

incf INR,1 ;

btfsc INR,H'02' ;

clrf INR ; INR=INR+1

return

;------------------------------------------------------------

Sum

;-производит счет входных импульсов + нормировку для

; обеспечения показаний в оборотах/мин.

; счет ведется в двоично-десятичном коде

; обнуление счетчика при включении и по таймеру после

; считывания результата

bcf STATUS,RP0 ; банк 0

; задание приращения счетчика импульсов

movlw 0x1 ; \*!!! в десятичном виде !!!!!!

movwf DOB

; ----------

movlw 0x0F ;

andwf DOB,0 ;

addwf OB0,1 ; добавить младшую тетраду приращения

btfss STATUS,DC ;

goto Metka2 ;--------------------------

movlw 0x06 ;

addwf OB0,1 ; коррекция при переносе из 3 в 4

goto Metka3 ;--------------------

Metka2 movlw 0x06 ; проверка и коррекция

addwf OB0,0 ; недопустимой комбинации

btfss STATUS,DC ; первой тетрады

goto Metka4 ; выход в конец процедуры !!!!!!!

movwf OB0 ;--------------------------

Metka3 ;проверка и коррекция недопустимой комбинации II тетр.

movlw 0x1 ;\*

addwf OB0,0 ;

btfss STATUS,C ;

goto Metka4 ; !!!!!!!!!!

movwf OB0

incf OB1,1 ;

; проверка и коррекция I тетрады II байта

movlw 0x06

addwf OB1,0

btfss STATUS,DC

goto Metka4

movwf OB1

; проверка II тетрады

movlw 0x1 ;\*

addwf OB1,0

btfss STATUS,C

goto Metka4

movwf OB1

Metka4 ; Добавить старшую тетраду приращения

movlw 0xF0

andwf DOB,0

addwf OB0,1

btfss STATUS,C

goto Metka5

movlw 0x1 ;\*

addwf OB0,1

goto Metka6

Metka5

movlw 0x1 ;\*

addwf OB0,0

btfss STATUS,C

goto Metka7

movwf OB0

Metka6

incf OB1

movlw 0x06

addwf OB1,0

btfss STATUS,DC

goto Metka7

movwf OB1

movlw 0x1 ;\*

addwf OB1,0

btfss STATUS,C

goto Metka7

movwf OB1

Metka7

bcf INTCON,INTF ; сброс флага прерывания

return

;--------------------------------------------------

Taim

; вызывает программу Indic

;-увеличивает содержимое таймера на 1

; следит за реальным временем:

; и вызывает программу Result

call Indic

bcf STATUS,RP0 ; банк 0

incfsz TIM0,1

goto Metka0

incf TIM1,1

Metka0 btfsc TIM1,TIMF

call Result

bcf INTCON,T0IF ; сброс флага прерывания

return

;---------------------------------------------------

Result

;-снимает показания счетчика оборотов,

; записывает результат в буфер индикации

bcf STATUS,RP0 ; банк 0

movlw H'0F'

andwf OB0,0

movwf IR0

swapf OB0,0

andlw 0x0F

movwf IR1

movlw H'0F'

andwf OB1,0

movwf IR2

swapf OB1,0

andlw 0x0F

movwf IR3

; гасит незначащие нули на индикаторе

movf IR3,1

btfss STATUS,Z

goto Metka1

movlw 0x0A

movwf IR3

movf IR2,1

btfss STATUS,Z

goto Metka1

movlw 0x0A

movwf IR2

movf IR1,1

btfss STATUS,Z

goto Metka1

movlw 0x0A

movwf IR1

; обнуляет счетчик оборотов и таймер 1 секунда

Metka1 clrf OB0

clrf OB1

clrf TIM0

clrf TIM1

return

;-----------------------------------------------------

Int

;-начало подпрограммы обработки прерываний-----

;-прерывания возможны от таймера и от входа INT0

;-задача - определить причину прерывания и вызвать

;-соответствующую подпрограмму - Taim или Sum

btfsc INTCON,INTF

call Sum

btfsc INTCON,T0IF

call Taim

retfie

;-------------------------------------------------

Begin

;-----начало основной программы-----------------

; выпоняет начальные установки режимов работы и регистров

; ждет прерывания от таймера и сигнального входа

bcf STATUS,RP0 ; банк 0

; установки портов - индикатор погашен

clrf PORTB

movlw 0xFF

movwf PORTA

; установки режимов работы

bsf STATUS,RP0 ; банк 1

movlw 0x10

movwf TRISA

movlw 0x01

movwf TRISB

; bcf OPTION\_REG,PS0 ;

bcf OPTION\_REG,PS2 ; коэффициет деления 1:16

bcf OPTION\_REG,PSA ; предв. делитель перед TMRO

bcf OPTION\_REG,T0CS ; внутреннее тактирование TMRO

;!!!!!!!bcf OPTION\_REG,NOT\_RBPU ; включить подтяг. рез-ры

bcf STATUS,RP0 ; банк 0

; сброс таймера

clrf TMR0

clrf TIM0

clrf TIM1

; сброс счетчика входных импульсов

clrf OB0

clrf OB1

; сброс буфера индикации

clrf INR

clrf IR0

clrf IR1

clrf IR2

clrf IR3

movlw 0xB0 ; сброс всех флагов прерывания

movwf INTCON ; разрешить прер-я от таймера и входа

;----------------------------------------

Wait goto Wait ; ожидание прерывания

;------------------------------------------------------------

end