

```

*****
'* Name      : Epson16F72_01.PBP                                     *
'* Author    : Anatoly A. Tokarev                                   *
'* Notice    : Copyright (c) 2007 Anatoly A. Tokarev               *
'*           : All Rights Reserved                                  *
'* Date      : 09.12.2007                                           *
'* Version   : 0.01                                                 *
'* Notes     : Контроллер для управления биполярным двигателем    *
'*           : в Epson на PIC16F72 - 2 канала                      *
*****

' Подключение железа:
'
' RA0 - Ph1A - выход
' RA1 - I01A - выход
' RA2 - I11A - выход
' RA3 - Ph2A - выход
' RA4 - I02A - выход
' RA5 - I02A - выход

' Включены резисторы подтяжки PORTB
' RB0 - STEPA - вход
' RB1 - DIRA  - вход
' RB2 -      - выход
' RB3 - DIRB  - вход
' RB4 - STEPB - вход
' RB5 -      - выход
' RB6 - UrefA - выход
' RB7 - UrefB - выход
'
' RC0 - Ph1B - выход
' RC1 - I01B - выход
' RC2 - I11B - выход
' RC3 - Ph2B - выход
' RC4 - I02B - выход
' RC5 - I02B - выход
' RC6 -      - выход
' RC7 -      - выход

DEFINE OSC 4           ' 20 MHz OSC
DEFINE NO_CLRWDT 1     ' убрали из программы вставки CLRWDT

'-- Задаем для удобства обозначения портам и выводам -----
STEPA  var PORTB.0      ' вход STEP A
DIRA   var PORTB.1      '
DIRB   var PORTB.3      '
STEPB  var PORTB.4      ' вход STEP A
RefA   var PORTB.7      '
RefB   var PORTB.6      '

'-- Переменные -----

CntrA  var byte         ' реверсивный счетчик канала A
CntrB  var byte         ' реверсивный счетчик канала B
IdleCntr var byte       ' Время простоя до снижения тока в обмотках
A      var Byte[17]     ' массив с управляющей последовательностью

GoTo Begin              '

' Собственно само начало программы
Begin:
' Проводим инициализацию контроллера
' bank 1
ADCON1 = %00000110      ' все входы - цифровые, опорное - от питания
TRISA = %00000000       ' все выходы
TRISB = %00011011       ' RB0-вход StepA, RB1- DirA, RB4-вход StepB, RB3- DirB,
TRISC = %00000000       ' все выходы
SSPSTAT = %01000000     ' CKE=1

```

```

OPTION_REG = %00000000      ' включили подтягивающие резисторы в PORTB,
                              ' прерывания INT по падающему 1->0 переходу

' bank 0
ADCON0 = 0                   ' АЦП выключен
SSPCON = %00000000          ' запрещаем SSP
T1CON = $31                  ' включаем Timer1, prescaler = 1:8, переполнение через 524,288 мс
INTCON = 0                   ' все запрещено
PIR1 = 0                     ' сбрасываем все флажки прерываний от периферии

'-- Читаем таблицу в память для ускорения работы -----
For IdleCntr = 0 To 15
    lookup IdleCntr,[$03,$02,$09,$10,$38,$30,$29,$22,$27,$26,$2D,$34,$1C,$14,$0D,$06],CntrA
    A[IdleCntr] = CntrA
Next

'-- Задаем начальные значения -----
CntrA = 0
CntrB = 0
TMR1H = 0
TMR1L = 0
IdleCntr = 0
RefA = 1
RefB = 1
PORTA = $03                  ' начальное значение
PORTC = $03                  ' начальное значение

'== Бесконечный цикл опроса =====
Loop:
'-- обработка сигнала STEPA -----
if (INTCON.1 = 1) then      ' ждем прерывания от INT - сигнала STEP
    RefA = 1
    INTCON.1 = 0           ' было прерывание, сбрасываем флажок
    If DIRA = 0 Then       ' в зависимости от сигнала DIR
        CntrA = CntrA-1    ' уменьшаем
    Else                   ' или
        CntrA = CntrA+1    ' увеличиваем счетчик
    EndIf
    CntrA = CntrA & %00001111 ' ограничиваем счетчик 4 разрядами
    PORTA = A[CntrA]
    IdleCntr = 0
endif

'-- обработка сигнала STEPВ -----
if (INTCON.0 = 1) then      ' ждем прерывание по изменению состояния вывода PortB
    RefB = 1
    INTCON.0 = 0           ' дождались
    if StepB = 0 then      ' это переход 1->0? Тогда обрабатываем сигнал STEP
        If DIRB = 0 Then  ' в зависимости от сигнала DIR
            CntrB = CntrB-1 ' уменьшаем
        Else              ' или
            CntrB = CntrB+1 ' увеличиваем счетчик в канале В
        EndIf
        CntrB = CntrB & %00001111 ' ограничиваем счетчик 4 разрядами
        PORTC = A[CntrB]
    endif
    IdleCntr = 0
endif

'-- обработка переполнения TMR1 -----
if PIR1.0 = 1 then         ' прерывание по переполнению TMR1
    PIR1.0 = 0             ' сбрасываем флажок
    IdleCntr = IdleCntr + 1
    if IdleCntr => 64 then  ' примерно 33 сек
        IdleCntr = 64      ' ограничиваем счетчик по максимальному значению
        RefA = 0           ' уменьшаем опорное напряжение
        RefB = 0           ' в каналах
    endif

```

```
endif
```

```
goto Loop
```

```
,
```

```
'=== Здесь окончание цикла опроса ====='
```