

```

;*****
;
;   Filename:      Stepper Gauge_16F1829_v.1.6.asm
;   Date:         07.21.15
;   File Version:
;
;   Author:       John Anhalt
;*****
;
;   Files Required: P16F1829.INC
;
;*****
;
;   Notes:
;   1) Tabs=5, Gutter=4, Window =80
;
;*****
;
;   Revision History:
;   07.19.15 Added Mode3: quarter step
;   07.21.15 Added start-up zero/mid-range
;
;*****

list      p=16f1829      ; list directive to define processor
#include <p16f1829.inc> ; processor specific variable definitions
errorlevel -302,-305
RADIX     DEC

__CONFIG __CONFIG1, _FOSC_INTOSC & _WDTE_OFF & _PWRTE_OFF & _MCLRE_ON & _CP_OFF & _CPD_OFF & _BOREN_OFF
& _CLKOUTEN_ON & _IESO_OFF & _FCMEN_OFF
__CONFIG __CONFIG2, _WRT_OFF & _PLLEN_OFF & _STVREN_OFF & _BORV_19 & _LVP_OFF

;
;
;           VDD|      |VSS
;           RA5|      |RA0/ICSPDAT
;           RA4|      |RA1/ICSPCLK
;           MCLR/RA3|  |RA2
;           RC5|      |RC0
;           RC4|      |RC1
;           RC3|      |RC2
;           RC6|      |RB4
;           RC7|      |RB5
;           RB7|      |RB6
;
;
;   CBLOCK 0x20      ;General Purpose RAM
;
;   Mode          ;AN906B
;   State         ;AN906B
;   Delay         ;AN906B
;   CountL        ;AN906B
;   CountH        ;AN906B
;   Index         ;AN906B
;   ENDC
;   CBLOCK 0x70      ;Common RAM
;   delayhi       ;used in delay macro
;   Flag0         ;generic flag register
;   ENDC
;
;   #define SW1     PORTA,2      ;N.O. switch, open = high
;   #define A1      PORTB,4      ;coil A, pin 1
;   #define A2      PORTB,6      ;coil A, pin 2
;   #define B1      PORTB,5      ;coil B, pin 1
;   #define B2      PORTB,7      ;coil B, pin 2
;   #define FWD_REV Flag0,0

```

```

;*****
;
;           DELAY MACRO
;*****
;
; 1) DelayCy() subsystem macro generates four instructions
; 2) This is the Macro for 14-bit cores (bra instruction)
; 3) Required register(s): delayhi
; 4) Provided by Mike K8LH

```

```

; 5) Use: DelayCy      (nnn*msecs)
;*****
clock    equ      8          ;4, 8, 12, 16, or 20 MHz clock
usecs    equ      clock/4    ;cycles per microsecond multiplier
msecs    equ      clock/4*1000 ;cycles per millisecond multiplier
DelayCy  macro      delay      ;12..327690 cycle range
    movlw    high((delay-11)/5)+1
    movwf    delayhi
    movlw    low((delay-11)/5)
    call     uDelay-((delay-11)%5)
endm

;*****

ORG      0x0000          ;processor reset vector
PAGESEL  START
GOTO     START          ;According to Microchip, when using debug
                        ;header, first inst. may be passed over by ICD2.

;*****
; INTERRUPT SERVICE ROUTINE
;*****
ORG      0x0004
;*****
; USER INTERRUPT SERVICE ROUTINE GOES HERE
;*****
; Note the 16F1827 family automatically handles context restoration for
; W, STATUS, BSR, FSR, and PCLATH where previous templates for 16F families
; required manual restoration. Shadow registers store these SFR values, and
; shadow registers may be modified since they are readable and writable for
; modification to the context restoration.
;*****
    RETFIE              ; return from interrupt
;*****
START
    MOVLB    1          ;Bank1
    MOVLW    b'01110000' ;
    MOVWF    OSCCON      ;4xPLL disabled, 8MHz
    BTFS    OSCSTAT,HFIOFR ;HF internal oscillator ready
    GOTO     $-1
    MOVLW    b'00001100' ;
    MOVWF    TRISA       ;PORTA 2&3 inputs
    CLRF     TRISB       ;all outputs
    CLRF     TRISC       ;all outputs
    CLRF     ADCON0      ;ensure ADC off (necessary?)
    CLRF     ADCON1      ;ensure ADC off (necessary?)
    BCF      OPTION_REG,7 ;enable WPU by individual latches
    MOVLB    3          ;Bank3
    CLRF     ANSELA      ;Turn off analog inputs
    CLRF     ANSELB      ;NB:comparators are off on reset
    CLRF     ANSEL      ;
    MOVLB    4          ;
    BSF      WPUA,2      ;enable WPUA for SW1
    MOVLB    0          ;
MAIN
    CLRF     Mode
;*****
;AN906B: Mode 1 (OFF), 2 (single-step), and 3 (half-step) ONLY
;Changed to: Mode 0 (OFF), 1 (single-step), 2 (half-step), and 3 (quarter-step)
;*****
ModeSelect
    clrf     State      ;"DEBOUNCE" returns here
    movf     Mode,w      ;initialize motor state to zero
    andlw    0x07        ;move current mode to w
    movwf    Mode        ;converts 0x08 to 0x00
    addlw    low ModeSelectTable
    movwf    PCL

ModeSelectTable
    goto     Mode0       ;motor off
    goto     Mode1F      ;single-step forward
    goto     Mode1R      ;single-step reverse
    goto     Mode2F      ;half-step forward

```

```

goto Mode2R      ;half-step reverse
goto Mode3F      ;quarter-step forward
goto Mode3R      ;quarter-step reverse
nop              ;07.19.15
clrf Mode
goto Mode0       ; start back at Mode0

;*****
; Mode0 - Motor off
; Wait here with the motor off until switch 1 is pressed.
;*****
Mode0
    movlb    2
    clrf     LATB
    movlb    0
    btfss    SW1      ;SW1 pressed (=low), goto debounce routine
    goto     Debounce ;debounce increments mode# and returns to Mode Select
    goto     Mode0

;*****
; Model - Single-Step Mode
;*****
ModelF
    bcf      FWD_REV
    decf     State,f      ;adjusts for initial incf
    movlw    high(620)
    movwf    CountH
    movlw    low(620)
    movwf    CountL
    goto     Model

ModelR
    bsf      FWD_REV
    incf     State,f      ;adjusts for initial decf
    movlw    high(300)
    movwf    CountH
    movlw    low(300)
    movwf    CountL
    goto     Model

Model
    movlb    1      ;Bank1
    movlw    B'00000100' ;TMR0 pre-scale = 1:32, 1:8(010) does not work, enable WPU
    movwf    OPTION_REG
    movlb    0      ; Bank 0

ModelStart
    movf     CountL,w      ;16-bit decrement counter
    skpnz
    decf     CountH,f
    decf     CountL,f
    skpnz
    movf     CountH,f
    skpnz
    goto     Mode0         ;stop, wait for switch
    btfss    INTCON,T0IF   ;timer 0 interrupt flag set?
    goto     $-1           ;ModelStateMachine ;do not change state
    bcf      INTCON,T0IF   ;yes, then clear interrupt flag,change state
    btfss    FWD_REV
    incf     State,f      ;increment motor state
    btfsc    FWD_REV
    decf     State,f      ;decrement motor state
    goto     ModelStateMachine

ModelStateMachine
    movf     State,w
    andlw    0x03         ;converts 0x04 to 0x00 or 0xFF to 0x03
    movwf    State
    call     ModelJumpTable
    movlb    2
    movwf    LATB
    movlb    0
    goto     ModelStart

ModelJumpTable
    brw

```

```

    dt 0x10,0x20,0x40,0x80
;*****
;NB: PortB is just the 4 high bits.  State0 = 0001(0x10), State1 = 0010(0x20),
;State2 = 0100(0x40),and State3 = 1000(0x80)
;*****
; Mode2 - Half-Step Mode
;*****
Mode2F
    bcf      FWD_REV
    decf     State,f
    movlw    high(300)
    movwf    CountH
    movlw    low(300)
    movwf    CountL
    goto     Mode2

Mode2R
    bsf      FWD_REV
    incf     State,f
    movlw    high(300)
    movwf    CountH
    movlw    low(300)
    movwf    CountL
    goto     Mode2

Mode2
    movlb    1                ;Bank 1
    movlw    B'00000011'      ;1:16
    movwf    OPTION_REG
    movlb    0                ;Bank 0

Mode2Start
    movf     CountL,w          ;16-bit decrement counter
    skpnz
    decf     CountH,f
    decf     CountL,f
    skpnz
    movf     CountH,f
    skpnz
    goto     Mode0             ;stop, wait for switch
    btfss    INTCON,T0IF       ;timer 0 interrupt?
    goto     $-1
    bcf      INTCON,T0IF       ;yes, then clear interrupt flag
    btfss    FWD_REV
    incf     State,f           ;increment motor state
    btfsc    FWD_REV
    decf     State,f           ;decrement motor state
    goto     Mode2StateMachine

Mode2StateMachine
    movf     State,w
    andlw    0x07
    movwf    State
    call     Mode2JumpTable
    movlb    2
    movwf    LATB
    movlb    0
    goto     Mode2Start

Mode2JumpTable
    brw
    dt 0x10,0x30,0x20,0x60
    dt 0x40,0xC0,0x80,0x90
;*****
;NB: PortB is just the 4 high bits.  State0 = 0001(0x10), State1 = 0010(0x20),
;State2 = 0100(0x40),and State3 = 1000(0x80)
;*****
; Mode3 - Quarter-Step Mode
;*****
Mode3F
    bcf      FWD_REV
    decf     State,f
    movlw    high(300)
    movwf    CountH

```

```

    movlw    low(300)
    movwf    CountL
    goto     Mode3
Mode3R
    bsf      FWD_REV
    incf     State,f
    movlw    high(300)
    movwf    CountH
    movlw    low(300)
    movwf    CountL
    goto     Mode3
Mode3
    movlb    1                ;Bank 1
    movlw    B'00000010'      ;1:8 was 1:32 prescaler for TMR0
    movwf    OPTION_REG
    movlb    0                ;Bank 0

Mode3Start
    movf     CountL,w          ;16-bit decrement counter
    skpnz
    decf     CountH,f
    decf     CountL,f
    skpnz
    movf     CountH,f
    skpnz
    goto     Mode0             ;stop, wait for switch
    btfss    INTCON,T0IF       ;timer 0 interrupt?
    goto     $-1
    bcf      INTCON,T0IF       ;yes, then clear interrupt flag
    btfss    FWD_REV
    incf     State,f           ;increment motor state
    btfsc    FWD_REV
    decf     State,f           ;decrement motor state
    goto     Mode3StateMachine
Mode3StateMachine
    movf     State,w
    andlw    0x0F
    movwf    State
    call     Mode3JumpTable
    movlb    2
    movwf    LATB
    movlb    0
    goto     Mode3Start

Mode3JumpTable
    brw
    dt 0x90,0x10,0x10,0x10,0x30,0x20,0x20,0x20
    dt 0x60,0x40,0x40,0x40,0xD0,0x80,0x80,0x80

;*****
;                               SUBROUTINES
;*****
;                               DEBOUNCE
;*****
Debounce
    clrf     PORTB             ;turn off all outputs do with LATC,Bank2?
    incf     Mode,f           ;goes to next mode on return -- 0x08 -> 0x00
    _bounce
    DelayCy   (10*msecs)
    btfss    SW1
    goto     _bounce
    goto     ModeSelect        ;goto mode select state machine
;*****
;                               DELAY SUBROUTINE
;*****
    nop
    nop                       ;entry for (delay-11)%5 == 4
    nop                       ;entry for (delay-11)%5 == 3
    nop                       ;entry for (delay-11)%5 == 2
    nop                       ;entry for (delay-11)%5 == 1
uDelay    addlw    -1          ;subtract 5 cycle loop time
    skpc
    decfsz    delayhi,f        ;borrow? no,skip,else
    bra       uDelay           ;done? yes,skip,else
    return                    ;do another loop
                                ;return with C = Z = 0

```

D:\eProject\MPLAB\MPLab Projects\Stepper Gauge\_16F1829\Stepper Gauge\_16F1829\_v.1.6.ASM

;\*\*\*\*\*

END