

```

; filename: i2s_500.asm

; This is a I2S test signal generator which generates a 500 Hz (441 sample) sine wave
; with 44.1 kHz sample rate.
; PORTB puts out data to a 74HCT165 shift register
; Code clock with 16*44.1 kHz=705.6 kHz, PIC clock is 8*705.6=5.6448 MHz (on RA6)

; v0.0 260512 Copied from i2s_1002.asm
; v0.1 260517 Ditch CLK2 for 74HC74 (sync issues CLK2-0)

LIST    P=16F1827, F=INHX32
#include <p16f1827.inc>
ERRORLEVEL -302                                ; sod message about using proper bank

__CONFIG __CONFIG1, _WDTE_SWDTEN & _BOREN_OFF & _CLKOUTEN_ON ;_FOSC_ECH
__CONFIG __CONFIG2, _PLLEN_OFF

; Equates
RESET_V    EQU    0x00                        ; Address of RESET Vector

; Registers
FLAGS    EQU    0x20                        ; Various flags
I2CBUF    EQU    0x21                        ; Used for I2C data and 2 ms delay
I2CCNT    EQU    0x22                        ; Used for I2C bitbanger and 2 ms delay
INDX     EQU    0x70                        ; Index for I2S sample (in all banks)

; Defines
#define SCL    PORTB,0                        ; I2C Clock
#define RED    PORTA,0                        ; ERR LED (red), TEST
#define GRN    PORTA,1                        ; Success LED (green)
#define SDA    PORTA,3                        ; I2C Data
#define TEST   LATA,0                          ; Start of the sequence (red LED)
#define LD     LATA,1                          ; Load 74HCT165 (green LED)
#define WS     LATA,2                          ; Word Sync
#define SYNCC  LATA,3                          ; Sync 74HCT74 clock with PIC
#define BLANK  LATA,4                          ; Blank 74HCT74 clock to '165 to prevent glitch
#define SYSINIT  FLAGS,0                       ; Si5351A needs this
#define NAK    FLAGS,1                        ; Not Acknowledge after I2C byte

    ORG    0
    GOTO   INIT
    ORG    4

; Interrupts
    RETFIE                                ; There are no interrupts

INIT
; Init stuff
    CLRF   STATUS                          ; Do initialization
    CLRF   BSR                              ; Select Bank 0
    CLRF   INTCON                            ; Clear int-flags, disable interrupts
    CLRF   PCLATH                             ; Keep in lower 2KByte
    CLRF   CCP1CON
    BSF    BSR,0                              ; Select Bank 1
    CLRF   TRISA                              ; RA7-0 Outputs
    CLRF   TRISB                              ; RB7-0 Outputs
    MOVLW  B'00000110'                        ; Timer0, prescaler 1:128
    MOVWF  OPTION_REG
    MOVLW  B'01101010'
    MOVWF  OSCCON                              ; 4 MHz intosc
    MOVLW  B'00000100'
    MOVWF  WDTCON                              ; WDT 4 ms, disabled
    CLRF   BSR                              ; Select Bank 0
    MOVLW  0xFF
    MOVWF  PORTB
    CLRF   PORTA
    BSF    SDA
    CLRF   FLAGS
    BSF    RED

```

```

CALL    STARTUP

; Initialize Si5351A
CALL    I2CSTART
MOVLW  0xC0           ; Address
CALL    I2CWRITE
CLRW    0             ; Prepare register 0 for reading
CALL    I2CWRITE
CALL    I2CSTOP

Readsysinit
CALL    I2CSTART
MOVLW  0xC1           ; Read address
CALL    I2CWRITE
CLRF    I2CCNT        ; Used for ACK timeout
BSF    BSR, 0         ; Select Bank 1
BSF    TRISA, 3       ; Set SDA as input to float SDA line
CLRF    BSR           ; Select Bank 0
CALL    I2CDELAY
BSF    SCL            ; Read SYS_INIT bit
CALL    I2CDELAY
BSF    SYSINIT
BTFSS  SDA
BCF    SYSINIT
BCF    SCL
MOVLW  0x07
MOVWF  I2CCNT

DUMMYREAD
CALL    I2CDELAY
BSF    SCL
CALL    I2CDELAY
BCF    SCL
DECFSZ I2CCNT, F
GOTO   DUMMYREAD
BSF    BSR, 0         ; Select Bank 1
BCF    TRISA, 3       ; Set SDA as output to assert SDA line
CLRF    BSR           ; Select Bank 0
BCF    SDA            ; Present NAK
CALL    I2CDELAY
BSF    SCL
CALL    I2CDELAY
BCF    SCL
CALL    I2CSTOP
BTFSC  SYSINIT        ; System init done?
GOTO   Readsysinit   ; No
BCF    RED            ; Done system init

; Start configure
CALL    I2CSTART
MOVLW  0xC0           ; Address
CALL    I2CWRITE
MOVLW  0x03           ; Register 3
CALL    I2CWRITE
MOVLW  0xFF           ; Register 3 data (Output Enable Control)
CALL    I2CWRITE
CALL    I2CSTOP

CALL    I2CSTART
MOVLW  0xC0           ; Address
CALL    I2CWRITE
MOVLW  0x10           ; Register 16
CALL    I2CWRITE
MOVLW  0x0F           ; Register 16 data: CLK0 control (On)
CALL    I2CWRITE
MOVLW  0x80           ; Register 17 data: CLK1 control (Off)
CALL    I2CWRITE
MOVLW  0x80           ; Register 18 data: CLK2 control (Off)
CALL    I2CWRITE

```

```

    MOVLW 0x80 ; Register 19 data: CLK3 control (Off)
    CALL I2CWRITE
    MOVLW 0x80 ; Register 20 data: CLK4 control (Off)
    CALL I2CWRITE
    MOVLW 0x80 ; Register 21 data: CLK5 control (Off)
    CALL I2CWRITE
    MOVLW 0x80 ; Register 22 data: CLK6 control (Off)
    CALL I2CWRITE
    MOVLW 0x80 ; Register 23 data: CLK7 control (Off)
    CALL I2CWRITE
    CALL I2CSTOP

    CALL I2CSTART
    MOVLW 0xC0 ; Address
    CALL I2CWRITE
    MOVLW 0xB7 ; Register 183
    CALL I2CWRITE
    MOVLW 0xD2 ; Register 183 data: XTAL load capacitance (10 pF)
and mandatory bit pattern
    CALL I2CWRITE
    CALL I2CSTOP

; PLLA values (745.1136 MHz)
    CALL I2CSTART
    MOVLW 0xC0 ; Address
    CALL I2CWRITE
    MOVLW 0x1A ; Register 26
    CALL I2CWRITE
    MOVLW 0x3D ; Register 26 data: MSNA_P3[15:8]
    CALL I2CWRITE
    MOVLW 0x09 ; Register 27 data: MSNA_P3[7:0]
    CALL I2CWRITE
    CLRW ; Register 28 data: MSNA_P1[17:16]
    CALL I2CWRITE
    MOVLW 0x0C ; Register 29 data: MSNA_P1[15:8]
    CALL I2CWRITE
    MOVLW 0xE6 ; Register 30 data: MSNA_P1[7:0]
    CALL I2CWRITE
    CLRW ; Register 31 data: MSNA_P3/2[19:16]
    CALL I2CWRITE
    MOVLW 0x3B ; Register 32 data: MSNA_P2[15:8]
    CALL I2CWRITE
    MOVLW 0xEA ; Register 33 data: MSNA_P2[7:0]
    CALL I2CWRITE
    CALL I2CSTOP

; MS0 values for 22.5792 MHz (CLK0)
    CALL I2CSTART
    MOVLW 0xC0 ; Address
    CALL I2CWRITE
    MOVLW 0x2B ; Register 43
    CALL I2CWRITE
    MOVLW 0x01 ; Register 43 data: MS0_P3[7:0]
    CALL I2CWRITE
    CLRW ; Register 44 data: MS0 misc
    CALL I2CWRITE
    MOVLW 0x0E ; Register 45 data: MS0_P1[15:8]
    CALL I2CWRITE
    MOVLW 0x80 ; Register 46 data: MS0_P1[7:0]
    CALL I2CWRITE
    CALL I2CSTOP

    CALL I2CSTART
    MOVLW 0xC0 ; Address
    CALL I2CWRITE
    MOVLW 0xB1 ; Register 177
    CALL I2CWRITE
    MOVLW 0x20 ; Register 177 data: PLLA reset
    CALL I2CWRITE

```



```

NOP
INCF     INDX, F
MOVFW   INDX
CALL    SAMPLE1
MOVWF   LATB
NOP
NOP
NOP
NOP
NOP
NOP
BSF     BLANK           ; Blank clock pulse to 74HCT165
BCF     LD              ; Load shift register
BSF     LD              ; Load shift register
NOP
BCF     BLANK           ; Unblank clock pulse to 74HCT165
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
DECF    INDX, F
MOVFW   INDX
CALL    SAMPLE1
MOVWF   LATB
NOP
NOP
BSF     WS              ; Set WS (R_Ch)
NOP
NOP
BSF     BLANK           ; Blank clock pulse to 74HCT165
BCF     LD              ; Load shift register
BSF     LD              ; Load shift register
NOP
BCF     BLANK           ; Unblank clock pulse to 74HCT165
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
INCF    INDX, F
MOVFW   INDX
CALL    SAMPLE1
MOVWF   LATB
NOP
NOP
NOP
NOP
NOP
NOP
DECF    INDX, F
BSF     BLANK           ; Blank clock pulse to 74HCT165
BCF     LD              ; Load shift register
BSF     LD              ; Load shift register
NOP

```



```

NOP
NOP
NOP
NOP
NOP
NOP
NOP
INCF    INDX, F
MOVFW   INDX
CALL    SAMPLE3
MOVWF   LATB
NOP
NOP
NOP
NOP
NOP
DECF    INDX, F
BSF     BLANK           ; Blank clock pulse to 74HCT165
BCF     LD
BSF     LD              ; Load shift register
NOP
BCF     BLANK           ; Unblank clock pulse to 74HCT165
DECF    INDX, F
DECFSZ  INDX, F
GOTO    LOOP3

CLRWDT
MOVLW   0xDC           ; 110 samples (4)
MOVWF   INDX
GOTO    $+5

```

LOOP4

```

NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
MOVFW   INDX
CALL    SAMPLE4
MOVWF   LATB
NOP
NOP
BCF     WS              ; Clear WS (L_Ch)
NOP
NOP
BSF     BLANK           ; Blank clock pulse to 74HCT165
BCF     LD
BSF     LD              ; Load shift register
NOP
BCF     BLANK           ; Unblank clock pulse to 74HCT165
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
INCF    INDX, F
MOVFW   INDX
CALL    SAMPLE4

```


GOTO LOOP

SAMPLE1

BRW

RETLW 0x00
RETLW 0x00
RETLW 0x3F
RETLW 0xFD
RETLW 0x3F
RETLW 0xBF
RETLW 0x3F
RETLW 0x2D
RETLW 0x3E
RETLW 0x4A
RETLW 0x3D
RETLW 0x16
RETLW 0x3B
RETLW 0x93
RETLW 0x39
RETLW 0xC2
RETLW 0x37
RETLW 0xA6
RETLW 0x35
RETLW 0x42
RETLW 0x32
RETLW 0x99
RETLW 0x2F
RETLW 0xAE
RETLW 0x2C
RETLW 0x86
RETLW 0x29
RETLW 0x23
RETLW 0x25
RETLW 0x8B
RETLW 0x21
RETLW 0xC2
RETLW 0x1D
RETLW 0xCD
RETLW 0x19
RETLW 0xB2
RETLW 0x15
RETLW 0x75
RETLW 0x11
RETLW 0x1D
RETLW 0x0C
RETLW 0xAE
RETLW 0x08
RETLW 0x2F
RETLW 0x03
RETLW 0xA5
RETLW 0xFF
RETLW 0x16
RETLW 0xFA
RETLW 0x89
RETLW 0xF6
RETLW 0x02
RETLW 0xF1
RETLW 0x89
RETLW 0xED
RETLW 0x22
RETLW 0xE8
RETLW 0xD4
RETLW 0xE4
RETLW 0xA4
RETLW 0xE0
RETLW 0x98
RETLW 0xDC
RETLW 0xB4
RETLW 0xD8

; Bytes 1-222

; The first two bytes always get skipped

RETLW 0xFE
RETLW 0xD5
RETLW 0x7B
RETLW 0xD2
RETLW 0x2F
RETLW 0xCF
RETLW 0x1E
RETLW 0xCC
RETLW 0x4D
RETLW 0xC9
RETLW 0xBF
RETLW 0xC7
RETLW 0x78
RETLW 0xC5
RETLW 0x7A
RETLW 0xC3
RETLW 0xC8
RETLW 0xC2
RETLW 0x64
RETLW 0xC1
RETLW 0x50
RETLW 0xC0
RETLW 0x8E
RETLW 0xC0
RETLW 0x1E
RETLW 0xC0
RETLW 0x00
RETLW 0xC0
RETLW 0x36
RETLW 0xC0
RETLW 0xBF
RETLW 0xC1
RETLW 0x9A
RETLW 0xC2
RETLW 0xC6
RETLW 0xC4
RETLW 0x42
RETLW 0xC6
RETLW 0x0B
RETLW 0xC8
RETLW 0x1F
RETLW 0xCA
RETLW 0x7C
RETLW 0xCD
RETLW 0x1F
RETLW 0xD0
RETLW 0x03
RETLW 0xD3
RETLW 0x26
RETLW 0xD6
RETLW 0x83
RETLW 0xDA
RETLW 0x16
RETLW 0xDD
RETLW 0xDA
RETLW 0xE1
RETLW 0xCA
RETLW 0xE5
RETLW 0xE2
RETLW 0xEA
RETLW 0x1C
RETLW 0xEE
RETLW 0x72
RETLW 0xF2
RETLW 0xDF
RETLW 0xF7
RETLW 0x5D
RETLW 0xFB
RETLW 0xE6

RETLW 0x00
RETLW 0x74
RETLW 0x05
RETLW 0x02
RETLW 0x09
RETLW 0x89
RETLW 0x0E
RETLW 0x04
RETLW 0x12
RETLW 0x6D
RETLW 0x16
RETLW 0xBE
RETLW 0x1A
RETLW 0xF1
RETLW 0x1F
RETLW 0x02
RETLW 0x22
RETLW 0xEA
RETLW 0x26
RETLW 0xA4
RETLW 0x2A
RETLW 0x2D
RETLW 0x2D
RETLW 0x7E
RETLW 0x30
RETLW 0x95
RETLW 0x33
RETLW 0x6D
RETLW 0x36
RETLW 0x01
RETLW 0x38
RETLW 0x50
RETLW 0x3A
RETLW 0x56
RETLW 0x3C
RETLW 0x0F
RETLW 0x3D
RETLW 0x7B
RETLW 0x3E
RETLW 0x97
RETLW 0x3F
RETLW 0x62
RETLW 0x3F
RETLW 0xDA
RETLW 0x3F
RETLW 0xFF
RETLW 0x3F
RETLW 0xD2
RETLW 0x3F
RETLW 0x51
RETLW 0x3E
RETLW 0x7E
RETLW 0x3D
RETLW 0x5A
RETLW 0x3B
RETLW 0xE7
RETLW 0x3A
RETLW 0x25
RETLW 0x38
RETLW 0x18
RETLW 0x35
RETLW 0xC2
RETLW 0x33
RETLW 0x27
RETLW 0x30
RETLW 0x49
RETLW 0x2D
RETLW 0x2C
RETLW 0x29

```
RETLW 0xD5
RETLW 0x26
RETLW 0x47
RETLW 0x22
RETLW 0x87
RETLW 0x1E
RETLW 0x9B
RETLW 0x1A
RETLW 0x87
RETLW 0x16
RETLW 0x51
RETLW 0x11
RETLW 0xFD
RETLW 0x0D
RETLW 0x92
RETLW 0x09
RETLW 0x16
RETLW 0x04
RETLW 0x8E
RETLW 0x00
RETLW 0x00
```

SAMPLE2

BRW

```
RETLW 0x00
RETLW 0x00
RETLW 0x02
RETLW 0x47
RETLW 0x06
RETLW 0xD3
RETLW 0x0B
RETLW 0x56
RETLW 0x0F
RETLW 0xCA
RETLW 0x14
RETLW 0x2A
RETLW 0x18
RETLW 0x70
RETLW 0x1C
RETLW 0x96
RETLW 0x20
RETLW 0x97
RETLW 0x24
RETLW 0x6D
RETLW 0x28
RETLW 0x14
RETLW 0x2B
RETLW 0x87
RETLW 0x2E
RETLW 0xC2
RETLW 0x31
RETLW 0xC0
RETLW 0x34
RETLW 0x7D
RETLW 0x36
RETLW 0xF6
RETLW 0x39
RETLW 0x28
RETLW 0x3B
RETLW 0x10
RETLW 0x3C
RETLW 0xAA
RETLW 0x3D
RETLW 0xF6
RETLW 0x3E
RETLW 0xF2
RETLW 0x3F
RETLW 0x9C
RETLW 0x3F
```

; Bytes 223-442

; The first two bytes always get skipped

RETLW 0xF3
RETLW 0x3F
RETLW 0xF7
RETLW 0x3F
RETLW 0xA8
RETLW 0x3F
RETLW 0x07
RETLW 0x3E
RETLW 0x13
RETLW 0x3C
RETLW 0xCF
RETLW 0x3B
RETLW 0x3C
RETLW 0x39
RETLW 0x5C
RETLW 0x37
RETLW 0x32
RETLW 0x34
RETLW 0xC0
RETLW 0x32
RETLW 0x09
RETLW 0x2F
RETLW 0x11
RETLW 0x2B
RETLW 0xDD
RETLW 0x28
RETLW 0x6F
RETLW 0x24
RETLW 0xCD
RETLW 0x20
RETLW 0xFB
RETLW 0x1C
RETLW 0xFE
RETLW 0x18
RETLW 0xDC
RETLW 0x14
RETLW 0x99
RETLW 0x10
RETLW 0x3B
RETLW 0x0B
RETLW 0xC9
RETLW 0x07
RETLW 0x47
RETLW 0x02
RETLW 0xBC
RETLW 0xFE
RETLW 0x2D
RETLW 0xF9
RETLW 0xA0
RETLW 0xF5
RETLW 0x1C
RETLW 0xF0
RETLW 0xA6
RETLW 0xEC
RETLW 0x44
RETLW 0xE7
RETLW 0xFB
RETLW 0xE3
RETLW 0xD2
RETLW 0xDF
RETLW 0xCD
RETLW 0xDB
RETLW 0xF2
RETLW 0xD8
RETLW 0x46
RETLW 0xD4
RETLW 0xCD
RETLW 0xD1
RETLW 0x8D

RETLW 0xCE
RETLW 0x89
RETLW 0xCB
RETLW 0xC5
RETLW 0xC9
RETLW 0x45
RETLW 0xC7
RETLW 0x0C
RETLW 0xC5
RETLW 0x1D
RETLW 0xC3
RETLW 0x7A
RETLW 0xC2
RETLW 0x26
RETLW 0xC1
RETLW 0x23
RETLW 0xC0
RETLW 0x71
RETLW 0xC0
RETLW 0x11
RETLW 0xC0
RETLW 0x05
RETLW 0xC0
RETLW 0x4B
RETLW 0xC0
RETLW 0xE5
RETLW 0xC1
RETLW 0xD0
RETLW 0xC3
RETLW 0x0C
RETLW 0xC4
RETLW 0x97
RETLW 0xC6
RETLW 0x6F
RETLW 0xC8
RETLW 0x93
RETLW 0xCA
RETLW 0xFE
RETLW 0xCD
RETLW 0xAE
RETLW 0xD0
RETLW 0x9F
RETLW 0xD3
RETLW 0xCE
RETLW 0xD7
RETLW 0x36
RETLW 0xDA
RETLW 0xD3
RETLW 0xDE
RETLW 0xA0
RETLW 0xE2
RETLW 0x99
RETLW 0xE6
RETLW 0xB8
RETLW 0xEA
RETLW 0xF8
RETLW 0xEF
RETLW 0x53
RETLW 0xF3
RETLW 0xC4
RETLW 0xF8
RETLW 0x44
RETLW 0xFC
RETLW 0xCF
RETLW 0x01
RETLW 0x5E
RETLW 0x05
RETLW 0xEB
RETLW 0x0A


```
RETLW 0x70
RETLW 0x0E
RETLW 0xE8
RETLW 0x13
RETLW 0x4C
RETLW 0x17
RETLW 0x98
RETLW 0x1B
RETLW 0xC4
RETLW 0x1F
RETLW 0xCD
RETLW 0x23
RETLW 0xAC
RETLW 0x27
RETLW 0x5D
RETLW 0x2A
RETLW 0xDB
RETLW 0x2E
RETLW 0x21
RETLW 0x31
RETLW 0x2C
RETLW 0x33
RETLW 0xF6
RETLW 0x36
RETLW 0x7D
RETLW 0x38
RETLW 0xBE
RETLW 0x3A
RETLW 0xB4
RETLW 0x3C
RETLW 0x5E
RETLW 0x3D
RETLW 0xBA
RETLW 0x3E
RETLW 0xC6
RETLW 0x3F
RETLW 0x80
RETLW 0x3F
RETLW 0xE8
```

SAMPLE3

BRW

```
RETLW 0x00
RETLW 0x00
RETLW 0xC0
RETLW 0x17
RETLW 0xC0
RETLW 0x7F
RETLW 0xC1
RETLW 0x39
RETLW 0xC2
RETLW 0x45
RETLW 0xC3
RETLW 0xA1
RETLW 0xC5
RETLW 0x4B
RETLW 0xC7
RETLW 0x41
RETLW 0xC9
RETLW 0x82
RETLW 0xCC
RETLW 0x09
RETLW 0xCE
RETLW 0xD3
RETLW 0xD1
RETLW 0xDE
RETLW 0xD5
RETLW 0x24
RETLW 0xD8
```

; Bytes 443-662

; The first two bytes always get skipped

RETLW 0xA2
RETLW 0xDC
RETLW 0x53
RETLW 0xE0
RETLW 0x32
RETLW 0xE4
RETLW 0x3B
RETLW 0xE8
RETLW 0x67
RETLW 0xEC
RETLW 0xB3
RETLW 0xF1
RETLW 0x17
RETLW 0xF5
RETLW 0x8F
RETLW 0xFA
RETLW 0x14
RETLW 0xFE
RETLW 0xA1
RETLW 0x03
RETLW 0x30
RETLW 0x07
RETLW 0xBB
RETLW 0x0C
RETLW 0x3B
RETLW 0x10
RETLW 0xAC
RETLW 0x15
RETLW 0x07
RETLW 0x19
RETLW 0x47
RETLW 0x1D
RETLW 0x66
RETLW 0x21
RETLW 0x5F
RETLW 0x25
RETLW 0x2C
RETLW 0x28
RETLW 0xC9
RETLW 0x2C
RETLW 0x31
RETLW 0x2F
RETLW 0x60
RETLW 0x32
RETLW 0x51
RETLW 0x35
RETLW 0x01
RETLW 0x37
RETLW 0x6C
RETLW 0x39
RETLW 0x90
RETLW 0x3B
RETLW 0x68
RETLW 0x3C
RETLW 0xF3
RETLW 0x3E
RETLW 0x2F
RETLW 0x3F
RETLW 0x1A
RETLW 0x3F
RETLW 0xB4
RETLW 0x3F
RETLW 0xFA
RETLW 0x3F
RETLW 0xEE
RETLW 0x3F
RETLW 0x8E
RETLW 0x3E
RETLW 0xDC

RETLW 0x3D
RETLW 0xD9
RETLW 0x3C
RETLW 0x85
RETLW 0x3A
RETLW 0xE2
RETLW 0x38
RETLW 0xF3
RETLW 0x36
RETLW 0xBA
RETLW 0x34
RETLW 0x3A
RETLW 0x31
RETLW 0x76
RETLW 0x2E
RETLW 0x72
RETLW 0x2B
RETLW 0x32
RETLW 0x27
RETLW 0xB9
RETLW 0x24
RETLW 0x0D
RETLW 0x20
RETLW 0x32
RETLW 0x1C
RETLW 0x2D
RETLW 0x18
RETLW 0x04
RETLW 0x13
RETLW 0xBB
RETLW 0x0F
RETLW 0x59
RETLW 0x0A
RETLW 0xE3
RETLW 0x06
RETLW 0x5F
RETLW 0x01
RETLW 0xD2
RETLW 0xFD
RETLW 0x43
RETLW 0xF8
RETLW 0xB8
RETLW 0xF4
RETLW 0x36
RETLW 0xEF
RETLW 0xC4
RETLW 0xEB
RETLW 0x66
RETLW 0xE7
RETLW 0x23
RETLW 0xE3
RETLW 0x01
RETLW 0xDF
RETLW 0x04
RETLW 0xDB
RETLW 0x32
RETLW 0xD7
RETLW 0x90
RETLW 0xD4
RETLW 0x22
RETLW 0xD0
RETLW 0xEE
RETLW 0xCD
RETLW 0xF6
RETLW 0xCB
RETLW 0x3F
RETLW 0xC8
RETLW 0xCD
RETLW 0xC6

```
RETLW 0xA3
RETLW 0xC4
RETLW 0xC3
RETLW 0xC3
RETLW 0x30
RETLW 0xC1
RETLW 0xEC
RETLW 0xC0
RETLW 0xF8
RETLW 0xC0
RETLW 0x57
RETLW 0xC0
RETLW 0x08
RETLW 0xC0
RETLW 0x0C
RETLW 0xC0
RETLW 0x63
RETLW 0xC1
RETLW 0x0D
RETLW 0xC2
RETLW 0x09
RETLW 0xC3
RETLW 0x55
RETLW 0xC4
RETLW 0xEF
RETLW 0xC6
RETLW 0xD7
RETLW 0xC9
RETLW 0x09
RETLW 0xCB
RETLW 0x82
RETLW 0xCE
RETLW 0x3F
RETLW 0xD1
RETLW 0x3D
RETLW 0xD4
RETLW 0x78
RETLW 0xD7
RETLW 0xEB
RETLW 0xDB
RETLW 0x92
RETLW 0xDF
RETLW 0x68
RETLW 0xE3
RETLW 0x69
RETLW 0xE7
RETLW 0x8F
RETLW 0xEB
RETLW 0xD5
RETLW 0xF0
RETLW 0x35
RETLW 0xF4
RETLW 0xA9
RETLW 0xF9
RETLW 0x2C
RETLW 0xFD
RETLW 0xB8
```

SAMPLE4

```
BRW
RETLW 0x00
RETLW 0x00
RETLW 0xFB
RETLW 0x71
RETLW 0xF6
RETLW 0xE9
RETLW 0xF2
RETLW 0x6D
RETLW 0xEE
```

; Bytes 663-882

; The first two bytes always get skipped

RETLW 0x02
RETLW 0xE9
RETLW 0xAE
RETLW 0xE5
RETLW 0x78
RETLW 0xE1
RETLW 0x64
RETLW 0xDD
RETLW 0x78
RETLW 0xD9
RETLW 0xB8
RETLW 0xD6
RETLW 0x2A
RETLW 0xD2
RETLW 0xD3
RETLW 0xCF
RETLW 0xB6
RETLW 0xCC
RETLW 0xD8
RETLW 0xCA
RETLW 0x3D
RETLW 0xC7
RETLW 0xE7
RETLW 0xC5
RETLW 0xDA
RETLW 0xC4
RETLW 0x18
RETLW 0xC2
RETLW 0xA5
RETLW 0xC1
RETLW 0x81
RETLW 0xC0
RETLW 0xAE
RETLW 0xC0
RETLW 0x2D
RETLW 0xC0
RETLW 0x00
RETLW 0xC0
RETLW 0x25
RETLW 0xC0
RETLW 0x9D
RETLW 0xC1
RETLW 0x68
RETLW 0xC2
RETLW 0x84
RETLW 0xC3
RETLW 0xF0
RETLW 0xC5
RETLW 0xA9
RETLW 0xC7
RETLW 0xAF
RETLW 0xC9
RETLW 0xFE
RETLW 0xCC
RETLW 0x92
RETLW 0xCF
RETLW 0x6A
RETLW 0xD2
RETLW 0x81
RETLW 0xD5
RETLW 0xD2
RETLW 0xD9
RETLW 0x5B
RETLW 0xDD
RETLW 0x15
RETLW 0xE0
RETLW 0xFD
RETLW 0xE5
RETLW 0x0E

RETLW 0xE9
RETLW 0x41
RETLW 0xED
RETLW 0x92
RETLW 0xF1
RETLW 0xFB
RETLW 0xF6
RETLW 0x76
RETLW 0xFA
RETLW 0xFD
RETLW 0xFF
RETLW 0x8B
RETLW 0x04
RETLW 0x19
RETLW 0x08
RETLW 0xA2
RETLW 0x0D
RETLW 0x20
RETLW 0x11
RETLW 0x8D
RETLW 0x15
RETLW 0xE3
RETLW 0x1A
RETLW 0x1D
RETLW 0x1E
RETLW 0x34
RETLW 0x22
RETLW 0x25
RETLW 0x25
RETLW 0xE9
RETLW 0x29
RETLW 0x7C
RETLW 0x2C
RETLW 0xD9
RETLW 0x2F
RETLW 0xFC
RETLW 0x32
RETLW 0xE0
RETLW 0x35
RETLW 0x83
RETLW 0x37
RETLW 0xE0
RETLW 0x39
RETLW 0xF4
RETLW 0x3B
RETLW 0xBD
RETLW 0x3D
RETLW 0x39
RETLW 0x3E
RETLW 0x65
RETLW 0x3F
RETLW 0x40
RETLW 0x3F
RETLW 0xC9
RETLW 0x3F
RETLW 0xFF
RETLW 0x3F
RETLW 0xE1
RETLW 0x3F
RETLW 0x71
RETLW 0x3E
RETLW 0xAF
RETLW 0x3D
RETLW 0x9B
RETLW 0x3C
RETLW 0x37
RETLW 0x3A
RETLW 0x85
RETLW 0x38

RETLW 0x87
RETLW 0x36
RETLW 0x40
RETLW 0x33
RETLW 0xB2
RETLW 0x30
RETLW 0xE1
RETLW 0x2D
RETLW 0xD0
RETLW 0x2A
RETLW 0x84
RETLW 0x27
RETLW 0x01
RETLW 0x23
RETLW 0x4B
RETLW 0x1F
RETLW 0x67
RETLW 0x1B
RETLW 0x5B
RETLW 0x17
RETLW 0x2B
RETLW 0x12
RETLW 0xDD
RETLW 0x0E
RETLW 0x76
RETLW 0x09
RETLW 0xFD
RETLW 0x05
RETLW 0x76
RETLW 0x00
RETLW 0xE9
RETLW 0xFC
RETLW 0x5A
RETLW 0xF7
RETLW 0xD0
RETLW 0xF3
RETLW 0x51
RETLW 0xEE
RETLW 0xE2
RETLW 0xEA
RETLW 0x8A
RETLW 0xE6
RETLW 0x4D
RETLW 0xE2
RETLW 0x32
RETLW 0xDE
RETLW 0x3D
RETLW 0xDA
RETLW 0x74
RETLW 0xD6
RETLW 0xDC
RETLW 0xD3
RETLW 0x79
RETLW 0xD0
RETLW 0x51
RETLW 0xCD
RETLW 0x66
RETLW 0xCA
RETLW 0xBD
RETLW 0xC8
RETLW 0x59
RETLW 0xC6
RETLW 0x3D
RETLW 0xC4
RETLW 0x6C
RETLW 0xC2
RETLW 0xE9
RETLW 0xC1
RETLW 0xB5

```

RETLW 0xC0
RETLW 0xD2
RETLW 0xC0
RETLW 0x40
RETLW 0xC0
RETLW 0x02

```

```

; Subroutines (I2C)

```

```

I2CSTART

```

```

BCF SDA
CALL I2CDELAY
BCF SCL
NOP
RETURN

```

```

I2CWRITE

```

```

; Writes byte in W

```

```

MOVWF I2CBUF
MOVLW 0x08
MOVWF I2CCNT

```

```

NEXTBIT

```

```

BCF STATUS,C ; Clear carry
RLF I2CBUF,F ; Rotate MSB into carry
BTFSC STATUS,C ; Carry clear?
BSF SDA ; No, set SDA
BTFSS STATUS,C ; Carry set?
BCF SDA ; No, clear SDA
BSF SCL ; Send clock
CALL I2CDELAY
BCF SCL ; Done clock
DECFSZ I2CCNT,F
GOTO NEXTBIT

```

```

; ACK handler

```

```

BSF SDA
BSF BSR,0 ; Select Bank 1
BSF TRISA,3 ; Set SDA as input to float SDA line
CLRF BSR ; Select Bank 0
BSF NAK
CALL I2CDELAY
BSF SCL ; Solicit ACK
CLRF I2CCNT

```

```

TESTACK

```

```

BTFSC SDA ; SDA pulled low?
GOTO CONTACK ; No, continue testing for ACK until timeout (~20 ms)
BCF NAK ; Clear NAK
GOTO EXITACK ; Exit loop

```

```

CONTACK

```

```

DECFSZ I2CCNT,F
GOTO TESTACK

```

```

EXITACK

```

```

BSF BSR,0 ; Select Bank 1
BCF TRISA,3 ; Set SDA as output
CLRF BSR ; Select Bank 0
BTFSC NAK ; NAK handler
GOTO FAIL ; Timeout, no ACK, illuminate ERR LED
BCF SCL
BSF SDA
CALL I2CDELAY
RETURN

```

```

FAIL

```

```

BSF RED
SLEEP ; Red LED stays on

```

```

I2CREAD

```

```

; Reads byte to I2CBUF

```



```

    BSF      SDA                ; Send ACK
    MOVLW   0x08
    MOVWF   I2CCNT
    BSF     BSR, 0              ; Select Bank 1
    BSF     TRISA, 3            ; Set SDA as input to float SDA line
    CLRF    BSR                ; Select Bank 0

NEXTREAD
    BSF     SCL                ; Send clock
    NOP
    NOP
    NOP
    BTFSC   SDA                ; SDA low?
    BSF     STATUS, C          ; No, set carry
    BTFSS   SDA                ; SDA high?
    BCF     STATUS, C          ; No, clear carry
    RLF     I2CBUF, F          ; Rotate bit into buffer
    BCF     SCL                ; Done clock
    CALL    I2CDELAY
    DECFSZ  I2CCNT, F
    GOTO    NEXTREAD

; ACK handler
    BSF     BSR, 0              ; Select Bank 1
    BCF     TRISA, 3            ; Set SDA as output
    CLRF    BSR                ; Select Bank 0
    BSF     SCL
    CALL    I2CDELAY
    BCF     SCL
    BCF     SDA                ; Clock out ACK
    CALL    I2CDELAY
    BSF     SCL
    CALL    I2CDELAY
    BSF     SDA
    RETURN

I2CSTOP
    BCF     SDA
    NOP
    NOP
    BSF     SCL
    CALL    I2CDELAY
    BSF     SDA
    RETURN

I2CDELAY
    NOP                ; 5 clocks delay
    RETURN

STARTUP
                                ; 2 ms delay at startup to let things settle
    DECFSZ  I2CBUF, F
    GOTO    $-1
    DECFSZ  I2CCNT, F
    GOTO    $-3
    RETURN

; End Subroutines

END

```