

















<http://www.ddrservice.net/f0z/Integrated-circuits/F/fi1216.pdf>

; FI1216 control, ADC and LED bar driver

|  |  |  |  |
| --- | --- | --- | --- |
| ; | V0.0 | 141130 | Initial draft |
| ; | V0.1 | 141201 | Done ADC, LED bar |
| ; | V0.2 | 141202 | Subroutines, TMR0 |
| ; | V0.3 | 141203 | Tetra 4s, MAX HOLD function |
| ; | V0.4 | 141204 | Fixed AD, lose controls, enter buttons |
| ; | V0.5 | 141206 | I2C |
| ; | V0.6 | 141207 | Fixed I2C |
| ; | V0.7 | 141228 | BRT Button |
| ; | V0.8 | 141231 | THR Button |
| ; | V0.9 | 150118 | Cal levels FI1216, f(LO) |
| ; | V0.10 | 150119 | THR setting |
| ; | V0.11 | 150119 | EEPROM R/W |
| ; | V1.0 | 150119  LIST | Release  P**=**16F818**,** F**=**INHX8M |

#include **<**p16f818.inc**>**

CONFIG \_WDT\_OFF **&** \_PWRTE\_OFF **&** \_INTRC\_IO **&** \_MCLR\_ON **&** \_BODEN\_OFF **&** \_LVP\_OFF **&** \_CPD\_OFF **&**

\_WRT\_ENABLE\_OFF **&** \_DEBUG\_OFF **&** \_CCP1\_RB2 **&** \_CP\_OFF

; Equates

RESET\_V EQU 0x00 ; Address of RESET Vector

OSC\_FREQ EQU D'8000000' ; Internal Oscillator Frequency is 8 MHz

; Registers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FLAGS | EQU | 0x20 |  | ; | Various flags |
| THRLVL | EQU | 0x21 |  | ; | Threshold level set via UI |
| MAXH | EQU | 0x22 |  | ; | LEVEL MAX HOLD value to be shown on LED bar |
| TEMPBAR | EQU | 0x23 |  | ; | Pattern on LED bar if it's blanked (BRT) |
| TEMPTHR | EQU | 0x24 |  | ; | Used in threshold set routine |
| BRITE | EQU | 0x25 |  | ; | LED Brightness |
| T4S | EQU | 0x26 |  | ; | Tetra 4s timing |
| I2CDELAY |  | EQU | 0x27 | ; | Used for I2C timing |
| I2CBUF | EQU | 0x28 |  | ; | Used for I2C data |
| I2CCNT | EQU | 0x29 |  | ; | Used for I2C bitbanger |
| DB2 | EQU | 0x30 |  | ; | Stealth byte for prescaler (0x50-0x5F) |
| TEMPW | EQU | 0x40 |  | ; | Context saving for interrupts |

TEMPSTATUS EQU 0x41 ; in all banks

; Defines

#define LEDS PORTB ; 8 LED bar RB0-3 GRN, RB4,5 ORG RB6,7 RED

#define RSSI PORTA**,**0 ; A/D input from AD8307 #define ALARM PORTA**,**1 ; Threshold Alarm Output #define THR PORTA**,**3 ; Alarm threshhold button input #define BRT PORTA**,**4 ; LED brightness button input #define SCL PORTA**,**6

#define SDA PORTA**,**7 ; I2C bus outputs #define BLANKBAR FLAGS**,**0 ; 1: blanked (BRT) #define BRTBTN FLAGS**,**1 ; 1: BRT Button Pressed #define NITE FLAGS**,**3 ; 1: lowest brightness #define DIM FLAGS**,**4 ; 1: low brightness #define DAY FLAGS**,**5 ; 1: mid brightness #define FULL FLAGS**,**6 ; 1: max brightness

ORG 0

**GOTO** START

ORG 4

; Timer0 Interrupt Handler

**BCF** INTCON**,**TMR0IF ; Clear TMR0 interrupt

**BCF** INTCON**,**GIE ; Disable global interrupts

**MOVWF** TEMPW ; Copy W to a Temporary Register (all banks) **SWAPF** STATUS**,**W ; Swap STATUS nibbles and place into W register **MOVWF** TEMPSTATUS ; Save STATUS to a Temporary register (DS31008a) **CALL** BARBRITE ; BRT Button test

**CALL** THRSET ; Threshold Setting Routine

**MOVLW** B'00000001'

**XORWF** FLAGS**,**1 ; XOR BLANKBAR

**BTFSC** BLANKBAR ; Test if clear

**GOTO** BLANKLEDS ; Set, blank leds

**MOVFW** BRITE ; Fetch LED brightness

**SUBLW** 0xFF ; Subtract BRITE from 0xFF

**MOVWF** TMR0 ; And load Timer0

**MOVFW** TEMPBAR ; Fetch LED pattern

**MOVWF** LEDS ; Write to LEDs

**GOTO** ENDINT BLANKLEDS

|  |  |  |  |
| --- | --- | --- | --- |
| **MOVFW** | BRITE | ; | Fetch LED brightness |
| **MOVWF** | TMR0 | ; | And load Timer0 |
| **CLRF** | LEDS | ; | Switch off LEDs |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENDINT | **CALL** | TETRA4S | ; | Check if 4s have elapsed, if so write pattern to TEMPBAR. |
|  | **SWAPF** | TEMPSTATUS**,**W | ; | Swap original STATUS register value |
|  | **MOVWF** | STATUS | ; | Restore STATUS register from W |
|  | **SWAPF** | TEMPW**,**F | ; | Swap W\_Temp nibbles and return value to W\_Temp |
|  | **SWAPF** | TEMPW**,**W | ; | Swap W\_Temp to W to restore original W (DS31008a) |
|  | **BSF** | INTCON**,**GIE | ; | Re-enable global interrupts |
|  | **RETFIE** |  |  |  |
| START |  |  |  |  |
| ; Init | stuff |  |  |  |
|  | **CLRF** | STATUS | ; | Do initialization, Select bank 0 |
|  | **CLRF** | INTCON |  |  |
|  | **BSF** | INTCON**,**GIE | ; | Enable GIE |
|  | **BSF** | INTCON**,**TMR0IE | ; | Enable Timer0 Interrupt |
|  | **CLRF** | PCLATH | ; | Keep in lower 2KByte |
|  | **CLRF** | CCP1CON |  |  |
|  | **MOVLW** | B'00111111' |  |  |
|  | **BANKSEL** | TRISA |  |  |
|  | **MOVWF** | TRISA | ; | RA7,6 Outputs, 5-0 Inputs |
|  | **CLRF** | TRISB | ; | RB7-0 Outputs |
|  | **MOVLW** | B'00000110' | ; | Timer0, prescaler 1:128 |
|  | **MOVWF** | OPTION\_REG |  |  |
|  | **MOVLW** | B'01110000' | ; | 8 MHz clock |
|  | **MOVWF** | OSCCON |  |  |
|  | **MOVLW** | B'00001110' | ; | AN0 Analog input, AN1-4 Digital IO, ADRESH only (ADRESL discarded) |
|  | **MOVWF** | ADCON1 |  |  |
|  | **MOVLW** | B'01000001' | ; | AD conv ON, AN0 Selected, Fosc/8 (T(AD)=1us) |
|  | **BANKSEL** | ADCON0 |  |  |
|  | **MOVWF** | ADCON0 |  |  |
|  | **CLRF** | PORTB | ; | Make all PORT B outputs low |
|  | **CLRF** | TMR0 | ; | Reset Timer 0 |
|  | **MOVLW** | 0x04 | ; | Startup Brightness ("dim") |
|  | **MOVWF** | BRITE |  |  |
|  | **BSF** | DIM |  |  |
| ; Restore Threshold from EEPROM | | | | |
|  | **BANKSEL** | EEADR | ; | Select Bank of EEADR |
|  | **MOVLW** | 0x00 |  |  |
|  | **MOVWF** | EEADR | ; | Data Memory Address to read (0x00) |
|  | **BANKSEL** | EECON1 | ; | Select Bank of EECON1 |
|  | **BCF** | EECON1**,** EEPGD | ; | Point to Data memory |
|  | **BSF** | EECON1**,** RD | ; | EE Read |
|  | **BANKSEL** | EEDATA | ; | Select Bank of EEDATA |
|  | **MOVF** | EEDATA**,** W | ; | W = EEDATA |
|  | **BANKSEL** | THRLVL |  |  |
|  | **MOVWF** | THRLVL | ; | Restored Threshold |
|  | **BSF** | SDA | ; | I2C idle |
|  | **BSF** | SCL |  |  |
|  | **MOVLW** | 0x60 | ; | Set Prescaler second byte (stealth mode) |
|  | **MOVWF** | DB2 |  |  |
|  | **CALL** | DEBOUNCE | ; | Clock Stabilization delay 16 ms |
|  | **CALL** | TUNE | ; | Set tuner frequency |
| MAIN |  |  |  |  |
|  | **BSF** | ADCON0**,**GO | ; | Start A/D (RSSI) |
| CONV0 |  |  |  |  |
|  | **NOP** |  |  |  |
|  | **BTFSC** | ADCON0**,**GO | ; | Test if done |
|  | **GOTO** | CONV0 |  |  |
|  | **MOVFW** | ADRESH | ; | Move 8 bit A/D result to W |
|  | **SUBWF** | MAXH**,**0 | ; | Subtract W from MAXH |
|  | **BTFSC** | STATUS**,C** | ; | result larger than MAXH? |
|  | **GOTO** | MAIN | ; | No |
|  | **MOVFW** | ADRESH |  |  |
|  | **MOVWF** | MAXH | ; | New MAXH |
|  | **GOTO** | MAIN |  |  |

TETRA4S

; Subroutine

; Tests if 4 seconds have elapsed

; Converts MAXH to LED bar

**DECFSZ** T4S**,**1

# RETURN

**MOVLW** 0xF2 ; 0xF2 for 4s

**MOVWF** T4S

**CALL** LEDBAR

**MOVWF** TEMPBAR

**CLRF** MAXH ; Destroy Max Hold value

**CALL** CHECKTHR ; Check threshold, beep if reached

# RETURN

LEDBAR

; MAXH contains RSSI max value

; LED bar pattern is in W

**MOVFW** MAXH

**ADDLW** 0x9A ; LEVEL > -40 dBm

**BTFSS** STATUS**,C GOTO** LEDS7F

LEDS7F

LEDS3F

LEDS1F

LEDS0F

LEDS07

LEDS03

LEDS01

LEDS00

**MOVLW** 0xFF ; Fullscale

# RETURN

**MOVFW** MAXH

**ADDLW** 0xA5 ; LEVEL > -48 dBm

**BTFSS** STATUS**,C GOTO** LEDS3F

**MOVLW** 0x7F

# RETURN

**MOVFW** MAXH

**ADDLW** 0xAF ; LEVEL > -56 dBm

**BTFSS** STATUS**,C GOTO** LEDS1F

**MOVLW** 0x3F

# RETURN

**MOVFW** MAXH

**ADDLW** 0xBA ; LEVEL > -64 dBm

**BTFSS** STATUS**,C GOTO** LEDS0F

**MOVLW** 0x1F

# RETURN

**MOVFW** MAXH

**ADDLW** 0xC5 ; LEVEL > -72 dBm

**BTFSS** STATUS**,C GOTO** LEDS07

**MOVLW** 0x0F

# RETURN

**MOVFW** MAXH

**ADDLW** 0xD0 ; LEVEL > -80 dBm

**BTFSS** STATUS**,C GOTO** LEDS03

**MOVLW** 0x07

# RETURN

**MOVFW** MAXH

**ADDLW** 0xDA ; LEVEL > -88 dBm

**BTFSS** STATUS**,C GOTO** LEDS01

**MOVLW** 0x03

# RETURN

**MOVFW** MAXH

**ADDLW** 0xE3 ; LEVEL > -96 dBm

**BTFSS** STATUS**,C GOTO** LEDS00

**MOVLW** 0x01

# RETURN

**MOVLW** 0x00 ; LEVEL at noise floor -108 dBm

# RETURN

|  |  |  |  |
| --- | --- | --- | --- |
| ; Set frequency  TUNE | of tuner |  | |
| **BCF** | INTCON**,**GIE | ; | Disable global interrupts |
| **CALL** | I2CSTART |  |  |
| **MOVLW** | 0xC0 | ; | Address |
| **CALL** | I2CBYTE |  |  |
| **MOVLW** | 0x34 | ; | Prescaler first byte |
| **CALL** | I2CBYTE |  |  |
| **MOVFW** | DB2 | ; | Get Prescaler second byte |
| **CALL** | I2CBYTE |  |  |
| **MOVLW** | 0xCA | ; | Control Byte |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **CALL** | I2CBYTE |  | |
| **MOVLW** | 0x90 | ; | Port Byte (Mid Band) |
| **CALL** | I2CBYTE |  |  |
| **CALL** | I2CSTOP |  |  |
| **BCF** | INTCON**,**TMR0IF | ; | Clear TMR0 interrupt just in case |
| **BSF** | INTCON**,**GIE | ; | Enable global interrupts |
| **RETURN** |  |  |  |
| I2CSTART |  |  |  |  |
|  | **BCF** | SDA |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **BCF** | SCL |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **RETURN** |  |  |  |
| I2CBYTE |  |  |  |  |
|  | **MOVWF** | I2CBUF |  |  |
|  | **MOVLW** | 0x08 |  |  |
|  | **MOVWF** | I2CCNT |  |  |
| NEXTBIT |  |  |  |  |
|  | **BCF** | STATUS**,C** | ; | Clear carry |
|  | **BCF** | SDA | ; | Glitch if next bit is 1 again UGLY! |
|  | **RLF** | I2CBUF**,**1 |  |  |
|  | **BTFSC** | STATUS**,C** |  |  |
|  | **BSF** | SDA |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **BSF** | SCL | ; | Send clock |
|  | **CALL** | DELAY10US |  |  |
|  | **BCF** | SCL |  |  |
|  | **CALL** | DELAY10US | ; | Done clock |
|  | **DECFSZ** | I2CCNT**,**1 |  |  |
|  | **GOTO** | NEXTBIT |  |  |
|  | **BCF** | SDA | ; | Fake ACK |
|  | **CALL** | DELAY10US |  |  |
|  | **BSF** | SCL |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **BCF** | SCL |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **RETURN** |  |  |  |
| I2CSTOP |  |  |  |  |
|  | **BCF** | SDA |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **BSF** | SCL |  |  |
|  | **CALL** | DELAY10US |  |  |
|  | **BSF** | SDA |  |  |
|  | **RETURN** |  |  |  |
| DELAY10US | | | | |
| **MOVLW** | | 0x14 | ; | 33 us delay |
| **MOVWF** | | I2CDELAY |  |  |

LOOP0

**DECFSZ** I2CDELAY**,**1 **GOTO** LOOP0 **RETURN**

DEBOUNCE

; Subroutine, used in THR setting and INIT

**MOVLW** 0xFF ; 96 ms outer delay

**MOVWF** I2CCNT

LOOP1 LOOP2

**MOVLW** 0xFF ; 0.39 ms inner delay

**MOVWF** I2CDELAY

**DECFSZ** I2CDELAY**,**1 **GOTO** LOOP2 **DECFSZ** I2CCNT**,**1 **GOTO** LOOP1 **RETURN**

BARBRITE

; Subroutine (called from within INT handler)

; Modifies BAR brightness after BRT is pressed

; Debounce (every 16 ms)

|  |  |  |
| --- | --- | --- |
| **BTFSC**  **GOTO** | BRT  NOBRT | ; Test BRT Button |
| **BTFSC RETURN** | BRTBTN | ; BRT Button still pressed? |
| **BSF** | BRTBTN |  |
| **BTFSC** | NITE |  |
| **GOTO** | DIMBRT |  |

|  |  |  |
| --- | --- | --- |
|  | **BTFSC** | DIM |
| **GOTO** | DAYBRT |
| **BTFSC** | DAY |
| **GOTO** | FULLBRT |
| **BCF** | FULL |
| **BSF** | NITE |
| **MOVLW** | 0x01 |
| **MOVWF RETURN** | BRITE |
| DIMBRT | **BCF** | NITE |
|  | **BSF** | DIM |
|  | **MOVLW** | 0x04 |
|  | **MOVWF**  **RETURN** | BRITE |
| DAYBRT | **BCF** | DIM |
|  | **BSF** | DAY |
|  | **MOVLW** | 0x20 |
|  | **MOVWF**  **RETURN** | BRITE |
| FULLBRT | **BCF** | DAY |
|  | **BSF** | FULL |
|  | **MOVLW** | 0xF0 |
|  | **MOVWF RETURN** | BRITE |
| NOBRT | **BCF** | BRTBTN |
|  | **RETURN** |  |

THRSET

; Subroutine (called from within INT handler)

; User interface to change threshold for alarm

; All LEDs are full on because the PWM brightness control uses the interrupt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **BTFSC** | THR | ; | Test THR Button |
| **GOTO** | NOTHR |  |  |
| THR1 | **CLRF** | LEDS | ; | switch off LEDs |
|  | **CALL** | DEBOUNCE | ; | 0.1 s delay |
|  | **BTFSS** | THR | ; | THR Button still pressed? |
|  | **GOTO** | THR1 |  |  |
| THR2 | **MOVFW** | THRLVL |  |  |
|  | **MOVWF** | LEDS | ; | Show threshold |
|  | **MOVWF** | TEMPTHR |  |  |
| THR3 | **BTFSS** | THR | ; | THR button? |
|  | **GOTO** | DECTHR |  |  |
|  | **BTFSS** | BRT | ; | BRT button? |
|  | **GOTO** | INCTHR |  |  |
|  | **GOTO** | THR3 |  |  |

; INC threshold INCTHR

|  |  |  |  |
| --- | --- | --- | --- |
| **CLRF** | LEDS | ; | switch off LEDs |
| **CALL** | DEBOUNCE |  |  |
| **CALL** | DEBOUNCE | ; | 0.2 s delay |
| **MOVFW** | THRLVL |  |  |
| **SUBLW** | 0xFF | ; | FF? |
| **BTFSS** | STATUS**,**Z |  |  |
| **GOTO** | INC7F |  |  |
| **GOTO** | DONETHR | ; | yes, exit INC |

INC7F

INC3F

INC1F

**MOVFW** THRLVL

**SUBLW** 0x7F **BTFSS** STATUS**,**Z **GOTO** INC3F

**MOVLW** 0xFF

**MOVWF** THRLVL

**GOTO** DONETHR

**MOVFW** THRLVL

**SUBLW** 0x3F **BTFSS** STATUS**,**Z **GOTO** INC1F

**MOVLW** 0x7F

**MOVWF** THRLVL

**GOTO** DONETHR

**MOVFW** THRLVL

**SUBLW** 0x1F **BTFSS** STATUS**,**Z **GOTO** INC0F

|  |  |  |
| --- | --- | --- |
|  | **MOVLW** | 0x3F |
| **MOVWF** | THRLVL |
| **GOTO** | DONETHR |
| INC0F | **MOVFW** | THRLVL |
|  | **SUBLW** | 0x0F |
|  | **BTFSS** | STATUS**,**Z |
|  | **GOTO** | INC07 |
|  | **MOVLW** | 0x1F |
|  | **MOVWF** | THRLVL |
|  | **GOTO** | DONETHR |
| INC07 | **MOVFW** | THRLVL |
|  | **SUBLW** | 0x07 |
|  | **BTFSS** | STATUS**,**Z |
|  | **GOTO** | INC03 |
|  | **MOVLW** | 0x0F |
|  | **MOVWF** | THRLVL |
|  | **GOTO** | DONETHR |
| INC03 | **MOVLW** | 0x07 |
|  | **MOVWF** | THRLVL |
|  | **GOTO** | DONETHR |

; DEC threshold DECTHR

|  |  |  |
| --- | --- | --- |
| **CLRF** | LEDS | ; switch off LEDs |
| **CALL** | DEBOUNCE |  |
| **CALL** | DEBOUNCE | ; 0.2 s delay |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0xFF | ; FF? |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DEC7F |  |
| **MOVLW** | 0x7F |  |
| **MOVWF** | THRLVL |  |
| **GOTO** | DONETHR |  |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0x7F | ; 7F? |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DEC3F |  |
| **MOVLW** | 0x3F |  |
| **MOVWF** | THRLVL |  |
| **GOTO** | DONETHR |  |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0x3F |  |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DEC1F |  |
| **MOVLW** | 0x1F |  |
| **MOVWF** | THRLVL |  |
| **GOTO** | DONETHR |  |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0x1F |  |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DEC0F |  |
| **MOVLW** | 0x0F |  |
| **MOVWF** | THRLVL |  |
| **GOTO** | DONETHR |  |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0x0F |  |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DEC07 |  |
| **MOVLW** | 0x07 |  |
| **MOVWF** | THRLVL |  |
| **GOTO** | DONETHR |  |
| **MOVFW** | THRLVL |  |
| **SUBLW** | 0x07 |  |
| **BTFSS** | STATUS**,**Z |  |
| **GOTO** | DONETHR |  |
| **MOVLW** | 0x03 |  |
| **MOVWF** | THRLVL |  |

DEC7F

DEC3F

DEC1F

DEC0F

DEC07

; There's no lower threshold, that'd be silly

; Done manipulating threshold DONETHR

|  |  |  |
| --- | --- | --- |
| **BTFSC** | THR | ; THR button? |
| **GOTO** | THR2 | ; No |
| **MOVFW** | THRLVL |  |
| **CALL** | DEBOUNCE | ; 0.1 s delay |
| **BTFSC** | THR | ; Still THR button? |

|  |  |  |  |
| --- | --- | --- | --- |
| **GOTO** | THR2 | ; | Yes, exit and restore THR |
| **CLRF** | LEDS | ; | switch off LEDs |
| **MOVFW** | TEMPTHR | ; | Restore THR |
| **MOVWF** | THRLVL |  |  |
| THR4  **CALL** | DEBOUNCE | ; | to enable releasing THR button |
| **BTFSS** | THR |  |  |
| **GOTO** | THR4 |  |  |
| **CALL** | DEBOUNCE |  |  |

; Store THRLVL into EEPROM (copied from datasheet) **BANKSEL** EECON1 ; Select Bank of EECON1 **BTFSC** EECON1**,** WR ; Wait for write

**GOTO $-**1 ; to complete

**BANKSEL** EEADR ; Select Bank of EEADR

**MOVLW** 0x00

**MOVWF** EEADR ; Data Memory Address to write

**BANKSEL** THRLVL **MOVFW** THRLVL **BANKSEL** EEDATA

**MOVWF** EEDATA ; Data Memory Value to write

**BANKSEL** EECON1 ; Select Bank of EECON1 **BCF** EECON1**,** EEPGD ; Point to DATA memory **BSF** EECON1**,** WREN ; Enable writes

**MOVLW** 0x55

**MOVWF** EECON2 ; Write 55h

**MOVLW** 0xAA

**MOVWF** EECON2 ; Write AAh

**BSF** EECON1**,** WR ; Set WR bit to begin write

**BCF** EECON1**,** WREN ; Disable writes

**BANKSEL** PORTB ; Select BANK0

NOTHR

# RETURN

CHECKTHR

; Subroutine called from TETRA4S

; Checks if threshold is reached

**MOVFW** TEMPBAR **SUBWF** THRLVL**,**0 **BTFSC** STATUS**,**Z **GOTO** PIP **RETURN**

; Creates "pip" PIP

|  |  |  |  |
| --- | --- | --- | --- |
| **MOVLW**  **BANKSEL** | B'00111101'  TRISA |  | |
| **MOVWF** | TRISA | ; | RA7,6,1 Outputs, 5-2,0 Inputs (ALARM output) |
| **MOVLW BANKSEL MOVWF** | 0xAA I2CCNT I2CCNT | ; | 64 ms outer delay |
| **MOVLW** | 0xFF | ; | 0.39 ms inner delay |
| **MOVWF** | I2CDELAY |  |  |
| **DECFSZ GOTO** | I2CDELAY**,**1 LOOP4 |  |  |
| **MOVLW** | B'00000010' |  |  |
| **XORWF DECFSZ GOTO MOVLW BANKSEL MOVWF** | PORTA**,**1 I2CCNT**,**1 LOOP3 B'00111111' TRISA  TRISA | ;  ; | Toggle ALARM  RA7,6 Outputs, 5-0 Inputs (ALARM Hi Z to minimize pop) |
| **BCF RETURN** | STATUS**,**RP0 | ; | Select bank 0 |
| END |  |  |  |

LOOP3 LOOP4