```
/*
**
###################################################################
#
**
**     Filename  : accelerometer.h
**
**     Version   : 0.0
**
**     Compiler  : Metrowerks HCS08 C Compiler
**
**     Date      : 21Feb07
**
**
###################################################################
#
*/

#ifndef _accelerometer_H
#define _accelerometer_H
#define DEMOQE

// Declare 'define' constants
//#define Trim_Val  0x8B     //Part A1 trim value for 25 MHz
//#define Trim_Val  0x93     //Part B2 trim value for 25 MHz
//#define Trim_Val  0x9E     //Part 64-2 trim value for 25 MHz
//#define FTrim_Val 1        //Part 64-2 ftrim value for 25 MHz
//#define Trim_Val  0xAD     //Part 64-2 trim value for 24 MHz
#define FTrim_Val 0        //Part 64-2 ftrim value for 24 MHz

#define max    0x10         //Max number of samples to
average/filter
#define dis_base   base16

typedef struct  {
  dword  reading[max];
  dword  result[max];
} ResultStct;

#ifdef DEMOQE
#define SW1 PTAD_PTAD2
#define SW2 PTAD_PTAD3
#define SW3 PTDD_PTDD2
#define SW4 PTDD_PTDD3
#define KBI_SW KBI1PE_KBIPE2_MASK | KBI1PE_KBIPE3_MASK
#define KBI_VAL (PTAD&0x0C)>>2

#define LED_PORT PTCD
#define LED_DD PTCDD
#define LED0 PTCD_PTCD0
#define LED1 PTCD_PTCD1
#define LED2 PTCD_PTCD2
```

```
#define LED_ON 0
#define LED_OFF 1
#else
#define SW1 PTDD_PTDD4
#define SW2 PTDD_PTDD5
#define SW3 PTDD_PTDD6
#define SW4 PTDD_PTDD7
#define KBI_SW KBI2PE_KBIPE4_MASK | KBI1PE_KBIPE5_MASK
#define KBI_VAL (PTDD&0x30)>>4

#define LED_PORT PTED
#define LED_DD PTEDD
#define LED0 PTED_PTED0
#define LED1 PTED_PTED1
#define LED2 PTED_PTED2
#define LED_ON 1
#define LED_OFF 0
#endif

#define Enter_Stop3 asm(jsr RamStop)
#define Enter_Wait asm(wait)
#define Breakpoint asm(bgnd)


#define ICSC1_FEI 0x04
#define ICSC2_FEI 0x06
#define ICSSC_FEI 0x80 | FTrim_Val
#define FEI_Speed 24000000
#define hi_baud 115000
#define fei_baud FEI_Speed/16/hi_baud

#define ICSC1_FEE 0x00
#define ICSC2_FEE 0x07
#define ICSSC_FEE 0x81
#define FEE_Speed 25165824
#define hi_baud 115000
#define fee_baud FEE_Speed/16/hi_baud

#define SCGC1_gates_on  0x81;
#define SCGC2_gates_on  0x30;
#define SCGC1_gates_off 0xFF;
#define SCGC2_gates_off 0xFF;

#define zero  0x30
#define one   0x31
#define two   0x32
#define three 0x33
#define four  0x34
#define five  0x35
#define six   0x36
#define seven 0x37
#define eight 0x38
```

```
#define nine  0x39
#define ten   0x61
#define enter 0x0D
#define space 0x20
#define EXIT  0x45
#define exit  0x65
#define yes   1
#define no    0
#define filter 2
#define avg    1
#define base10 1
#define base16 0

// RAM variables
#pragma DATA_SEG _DATA_ZEROPAGE
static byte samp=0,mode=0,n_str[5];
static word StartCount,StopCount;

static byte last_byte = 0,bytes_to_trans = 0, reading_mma7660_reg
= 0, repeat_start_sent = 0;
static byte count = 0,rec_count = 0,num_to_rec = 0,mma7660[] =
{0,0,0,0};
#pragma DATA_SEG DEFAULT

static ResultStct x;
static ResultStct y;
static ResultStct z;

static word IIC_Rec_Data[ ]={0,0,0,0};
static word IIC_Converted_Data[ ]={0,0,0,0};

void SendChar(char s_char);
void ICS_FEI(void);
void InitKBI(void);
void InitSCI(word baud);
char RecChar(void);
void SendChar(char s_char);
void SendMsg(char msg[]);
word hex2bcd(word hex);
byte asc2byte(char n_asc);
word asc2word(byte n_asc[2]);
char * byte2asc(byte num, byte base);
char * word2asc(word num, byte base);
void StartTPM(byte PS);
word StopTPM(void);
void PeriphInit(void);
void filter_data(void);
void avg_data(void);
void copy_data(void);
void ReadAcceleration(void);
void ShowAcceleration (void);
void Master_Read_and_Store(void);
```

```
void Master_Transmit(byte transbytes, byte recnum);
void Master_Receive(void);
void MMA7660_configuration(void);
void IIC_configuration (void);

#endif  //end #ifndef _accelerometer_H
```