```
#define _LEGACY_HEADERS

#include        <htc.h>
#include        "mssp_i2c.h"
#define TRUE 0x0;
#define FALSE 0x1;

/**************************************************
 *      MSSP使用マスターモード専用I2C関数
 *      教育・ホビー用、営利目的・商用への利用は禁止
 *      詳しいタイミングはNXPの資料やPICドキュメント
 *      参照のこと
 **************************************************
 */

/*
 *      mssp はSDAとSCL信号をスタート状態にします
 *      衝突有るとMSSPはリセットされる
 */
  void i2c_start(void){
        SEN = 1;                // SSPCON2:0 initiate I2C START condition
        while (SEN == 1);       // wait until START bit finishes
  }

/*
 *      mssp はSDAとSCL信号をRESTART状態にします
 *      衝突有るとBCL1IFが立つ
 */
  void i2c_restart( void) {
        RSEN = 1;               // SSP1CON2:1 Repeated Start
        while (RSEN);
  }

/*      関数の説明
 *      DATAをSSP1BUFに書きスレーブに転送かける
 *      転送中はRWが1になっている
 */
  void i2c_write( unsigned char DataByte ){
        SSP1BUF = DataByte;
        while (R_nW == 1);
  }

/*      関数の説明
 *      受信可にしてDATAを待つ。
 *      RSEN0になったら、SSP1BUFのデータを読み持って返る
 */
  unsigned char i2c_read( void) {
        RCEN = 1;               // SSP1CON2:3 Receive enable
        while (RCEN == 1);
        return SSP1BUF;
 }

/*      関数の説明
 *      mssp はSDAとSCL信号をstop状態にします
 *      PENが0になったら、かえる。衝突あるとBCL1IFが立つ
 */
  void i2c_stop( void) {
        PEN = 1;                // SSP1CON2:2 initiate I2C STOP condition
        while (PEN);
  }

/*      関数の説明
 *      ACK (ACKDT:0)かNACK(ACKDR:1)を送る
 *      ACKENしたらかえる。
 */
  void i2c_sendack(unsigned char status) {
        ACKDT = status;         // SSPCON2:5 set to ACK:0,NACK:1
        ACKEN = 1;              // SSPCON2:4  initiate I2C ACK condition
        while ( ACKEN);         // wait until ACK sequence is over
  }

/*      関数の説明
 *      ACK (ACKDT:0)かNACK(ACKDR:1)をチェックする
 *      ACK (ACKSTAT:0)かNACK(ACKSTAT:1)でかえる。
 */
```

```
 unsigned char i2c_readack(void) {
        if (ACKSTAT == 0) {
        return 1;               // ACK detected
        } else {
        return 0;               // ACK not detected
        }
 }

/*      関数の説明
 *      指定アドレスにWRITEモードでアドレスを書く
 *      ACK有ればTRUEで、無ければFALSEでかえる
 */
 unsigned char i2c_writeto(unsigned char address){
        i2c_start();
        i2c_write(address | 0x0);
        if(i2c_readack()) return TRUE;
        return FALSE;
 }

/*      関数の説明
 *      指定アドレスにREADモードでアドレスを書く
 *      ACK有ればTRUEで、無ければFALSEでかえる
 */
 unsigned char i2c_readfrom(unsigned char address){
        i2c_start();
        i2c_write(address | 0x1);
        if(i2c_readack()) return TRUE;
        return FALSE;
 }
```