

```

#define _I2C_H_

/*
 * Hi-TECH社提供のサンプルプログラムを参考にしました。商用利用禁止。
 *
 * 8MHz対応になってます
 *
 * 特に注意
 * もしもクロックとデータが同じポート群にあるときはPICにおけるポート
 * コマンドのリードモディファイライト動作を良く認識してください。
 * というのは1ビットだけセットしたりクリアしても他の全ビットに対して
 * も読み出し書き込みが行われます。
 */

#ifndef _XTAL_FREQ
/* 4MHz専用なので、それ以外の数値書いても4MHzにしてしまう */
#define _XTAL_FREQ 4000000
#endif

/* PIC内蔵のSSPモジュールを使わないソフト方式のマスターです */
#define LCD_AD 0x7C //

/* GPIO4:SCL, GPIO5:SDAを使用します */
#define SCL RA4 /* clock on port B bit 3 */
#define SCL_DIR TRISA4

#define SDA RA5 /* data on port B bit 1 */
#define SDA_DIR TRISA5

#define I2CTRIS TRISA

#define M_SDA_INP 0x02
#define M_SDA_OUT 0xFD
#define M_SCL_INP 0x04
#define M_SCL_OUT 0xFB

#define I2C_INPUT 1 /* data direction input */
#define I2C_OUTPUT 0 /* data direction output */

#define I2C_READ 0x01 /* read bit used with address */
#define I2C_WRITE 0x00 /* write bit used with address */

#define FALSE 0
#define TRUE !FALSE

#define I2C_ERROR (-1)
#define I2C_LAST FALSE /* SendAck: no more bytes to send */
#define I2C_MORE TRUE /* SendAck: more bytes to send */

#define i2c_start() i2c_restart()
#define i2c_writeto(address) i2c_open((address), I2C_WRITE)
#define i2c_readfrom(address) i2c_open((address), I2C_READ)

#define SCL_HIGH() SCL = 1; SCL_DIR = I2C_OUTPUT
#define SCL_LOW() SCL = 0; SCL_DIR = I2C_OUTPUT
#define SDA_HIGH() SDA = 1; SDA_DIR = I2C_OUTPUT
#define SDA_LOW() SDA = 0; SDA_DIR = I2C_OUTPUT

/*
 * i2cバス制御用のタイミング値です。およそのμsを示す。
 */

#define I2C_TM_BUS_FREE 14 // 5→7
#define I2C_TM_START_SU 14 // 7
#define I2C_TM_START_HD 12 // 6
#define I2C_TM_SCL_LOW 14 // 7
#define I2C_TM_SCL_HIGH 16 // 8
#define I2C_TM_DATA_SU 10 // 2→5
#define I2C_TM_DATA_HD 4 // 0→2
#define I2C_TM_SCL_TO_DATA 16 // 8
/* SCL low to data valid 4→8 */
#define I2C_TM_STOP_SU 12 // 6
#define I2C_TM_SCL_TMO 30 // 15

```

```

                                i2c
                                /* clock time out */

extern signed char      i2c_readack(void);
extern unsigned char    i2c_sendaddress(unsigned char, unsigned char);
extern unsigned char    i2c_sendbyte(unsigned char);
extern int              i2c_readbyte(void);
extern void             i2c_restart(void);
extern void             i2c_stop(void);
extern void             i2c_sendack(unsigned char);
extern signed char      i2c_putbyte(unsigned char);
extern int              i2c_getbyte(unsigned char);
extern unsigned char    i2c_open(unsigned char, unsigned char);
extern unsigned char    i2c_getstring(unsigned char *, unsigned char);
extern int              i2c_putstring(const unsigned char *, unsigned char);
extern unsigned char    i2c_waitforSCL(void);
extern void             i2c_free(void);
extern unsigned char    i2c_read(unsigned char);

```