```
/*****************************************************************************

  FileName:
        SM3.h
  Version:
        SM3_V1.1
  Date:
        Sep 18,2016
  Description:
        This headfile provide macro defination, parameter definition
          and function declaration needed in SM3 algorithm implement
  Function List:
    1.SM3_256         //calls SM3_init, SM3_process and SM3_done to calculate hash value
    2.SM3_init        //init the SM3 state
    3.SM3_process     //compress the the first len/64 blocks of the message
    4.SM3_done        //compress the rest message and output the hash value
    5.SM3_compress    //called by SM3_process and SM3_done, compress a single block of message
    6.BiToW           //called by SM3_compress,to calculate W from Bi
    7.WToW1           //called by SM3_compress, calculate W' from W
    8.CF              //called by SM3_compress, to calculate CF function.
    9.BigEndian       //called by SM3_compress and SM3_done.GM/T 0004-2012 requires to use
big-endian.
                      //if CPU uses little-endian, BigEndian function is a necessary call to
change the
                      //little-endian format into big-endian format.
    10.SM3_SelfTest   //test whether the SM3 calculation is correct by comparing the hash result
with the standard data
  History:
    1. Date:   Sep 18,2016
       Author: Mao Yingying, Huo Lili
       Modification: 1)add notes to all the functions
                     2)add SM3_SelfTest function
*****************************************************************************/


#include <string.h>

#define SM3_len 256
#define SM3_T1 0x79CC4519
#define SM3_T2 0x7A879D8A
#define SM3_IVA 0x7380166f
#define SM3_IVB 0x4914b2b9
#define SM3_IVC 0x172442d7
#define SM3_IVD 0xda8a0600
```

```
#define SM3_IVE 0xa96f30bc
#define SM3_IVF 0x163138aa
#define SM3_IVG 0xe38dee4d
#define SM3_IVH 0xb0fb0e4e

/* Various logical functions */
#define SM3_p1(x)         (x^SM3_rotl32(x,15)^SM3_rotl32(x,23))
#define SM3_p0(x)         (x^SM3_rotl32(x,9)^SM3_rotl32(x,17))
#define SM3_ff0(a,b,c)    (a^b^c)
#define SM3_ff1(a,b,c)    ((a&b)|(a&c)|(b&c))
#define SM3_gg0(e,f,g)    (e^f^g)
#define SM3_gg1(e,f,g)    ((e&f)|((~e)&g))
#define SM3_rotl32(x,n)   ((((unsigned int) x) << n) | (((unsigned int) x) >> (32 - n)))
#define SM3_rotr32(x,n)   ((((unsigned int) x) >> n) | (((unsigned int) x) << (32 - n)))


typedef struct {
    unsigned int  state[8];
    unsigned int  length;
    unsigned int  curlen;
    unsigned char  buf[64];
} SM3_STATE;


void BiToWj(unsigned int Bi[], unsigned int Wj[]);
void WjToWj1(unsigned int Wj[], unsigned int Wj1[]);
void CF(unsigned int Wj[], unsigned int Wj1[], unsigned int V[]);
void BigEndian(unsigned char src[], unsigned int bytelen, unsigned char des[]);
void SM3_init(SM3_STATE *md);
void SM3_compress(SM3_STATE * md);
void SM3_process(SM3_STATE * md, unsigned char buf[], int len);
void SM3_done(SM3_STATE *md, unsigned char *hash);
void SM3_256(unsigned char buf[], int len, unsigned char hash[]);
int SM3_SelfTest();
```