```
#ifndef SPI_HARDWARE_H
#define SPI_HARDWARE_H

#include "n32g45x.h"                    // Device header

#define DUMMY_BIT    0xFF

//定义接收的长度
#define Sector_Length          512 //仅单一扇区的字节数
#define Total_Sector_Length    516 //R1 + 正文 + CRC16 + CRC7
#define CID_CSD_Length         16
#define Total_CID_CSD_Length   20  //R1 + 正文 + crc16 + crc7
#define Normal_Length          5
#define No_Meaning_Length      0

//定义CID/CSD的TOKEN
/*!< Data token start byte, 单块读起始Token */
#define Single_Block_Read_Token          0xFE
/*!< Data token start byte, 多块读起始Token */
#define Mult_Block_Read_Token            0xFE
/*!< Data token start byte, 单块写起始Token */
#define Single_Block_Write_Token         0xFE
/*!< Data token start byte, 多块写起始Token */
#define Mult_Block_Write_Start_Token     0xFC
/*!< Data toke stop byte, 多块写停止Token */
#define Mult_Block_Write_End_Token       0xFD

//手动定义CSC
#define CMD0_CSC    0x95
#define CMD8_CSC    0x87
#define ACMD41_CSC  0x01
#define OTHER_CSC   0xFF

//定义CMD的命令参数(格式: 0x40 | (uint8_t)Cmd_Number)
typedef enum{
        CMD0 = 0x40,
        CMD1 = 0x41,
        CMD2 = 0x42,
        CMD3 = 0x43,
        CMD7 = 0x47,
        CMD8 = 0x48,
        CMD9 = 0x49,
        CMD10 = 0x4A,
        CMD16 = 0x50,
        CMD17 = 0x51,
        CMD24 = 0x58,
        CMD55 = 0x77,
        CMD58 = 0x7A,
        ACMD6 = 0x46,
        ACMD41 = 0x69,
}CMD_Type;

//定义参数返回状态
typedef enum{
        Failed = 0,
        Complete = !Failed,
}SPI_Status;

//定义CMD参数
#define CMD0_Index    0x00000000
#define CMD8_Index    0x000001AA
#define CMD55_Index   0x00000000
#define CMD58_Index   0x00000000
//ACMD41发送参数(需依照不同的环境下使用)
#define ACMD41_Index_L  0x00000000  //支持SC标准容量SD卡
#define ACMD41_Index_H  0x40000000  //支持HC/XC高容量SD卡

//定义引脚
#define Enable_CS(void)    GPIO_WriteBit(GPIOA, GPIO_PIN_4, (Bit_OperateType)0)
```

```
#define Disable_CS(void)   GPIO_WriteBit(GPIOA, GPIO_PIN_4, (Bit_OperateType)1)

//定义正常R1响应和错误位
#define Card_Ready      0x00
#define In_Idle_State   0x01
#define Erase_Reset     0x02
#define Illegal_CMD     0x04
#define Com_CRC_ERR     0x08
#define Erase_Seq_ERR   0x10
#define Address_ERR     0x20
#define Parameter_ERR   0x40
#define Invalid         0x80

//数据响应类型
#define SD_DATA_OK                (0x05)
#define SD_DATA_CRC_ERROR         (0x0B)
#define SD_DATA_WRITE_ERROR       (0x0D)
#define SD_DATA_OTHER_ERROR       (0xFF)

//数据响应成功与否
#define SD_WriteData_Success    0x01
#define SD_WriteData_Failed     0x00


//有效电压范围
#define Valid_Voltage_Range                0x00FF8000

void SPI_Hardware_Init(SPI_InitType SPI_InitStructure);

void SPI_Hardware_SendPulse(void);

void SPI_Hardware_SendCMD(uint8_t CMD, uint32_t Coefficient, uint8_t CMD_CRC, uint8_t *Datas_IN,
uint8_t *Datas_OUT);

uint8_t SPI_Hardware_SendACMD(uint8_t SD_Card_Type);

void SPI_Hardware_SendSinglePulse(void);

#endif
```