

零死角玩转STM32



SDIO—SD卡读写测试

淘宝：firestm32.taobao.com

论坛：www.chuxue123.com



扫描进入淘宝店铺

主讲内容



01

SDIO协议简介

02

SDIO命令及响应

03

SD卡的操作模式及切换

04

STM32的SDIO接口说明

05

STM32的SDIO结构体

06

实验：SDIO—SD卡读写测试

SDIO—SD卡读写测试



SDIO相关结构体

标准库函数对SDIO外设建立了三个初始化结构体，分别为SDIO初始化结构体SDIO_InitTypeDef、SDIO命令初始化结构体SDIO_CmdInitTypeDef和SDIO数据初始化结构体SDIO_DataInitTypeDef。

- 初始化结构体成员用于设置SDIO工作环境参数，并由SDIO相应初始化配置函数或功能函数调用，这些参数将会被写入到SDIO相应的寄存器，达到配置SDIO工作环境的目的。
- SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。
- SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {  
2     uint32_t SDIO_ClockEdge;           // 时钟沿  
3     uint32_t SDIO_ClockBypass;         // 旁路时钟  
4     uint32_t SDIO_ClockPowerSave;      // 节能模式  
5     uint32_t SDIO_BusWide;             // 数据宽度  
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制  
7     uint8_t  SDIO_ClockDiv;            // 时钟分频  
8 } SDIO_InitTypeDef;
```

- **SDIO_ClockEdge:** 主时钟SDIOCLK产生CLK引脚时钟有效沿选择，可选上升沿或下降沿，它设定SDIO时钟控制寄存器(SDIO_CLKCR)的NEGEDGE位的值，一般选择设置为高电平。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {  
2     uint32_t SDIO_ClockEdge;           // 时钟沿  
3     uint32_t SDIO_ClockBypass;        // 旁路时钟  
4     uint32_t SDIO_ClockPowerSave;     // 节能模式  
5     uint32_t SDIO_BusWide;            // 数据宽度  
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制  
7     uint8_t  SDIO_ClockDiv;           // 时钟分频  
8 } SDIO_InitTypeDef;
```

- **SDIO_ClockBypass:** 时钟分频旁路使用，可选使能或禁用，它设定SDIO_CLKCR寄存器的BYPASS位。如果使能旁路，SDIOCLK直接驱动CLK线输出时钟；如果禁用，使用SDIO_CLKCR寄存器的CLKDIV位值分频SDIOCLK，然后输出到CLK线。一般选择禁用时钟分频旁路。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {
2     uint32_t SDIO_ClockEdge;           // 时钟沿
3     uint32_t SDIO_ClockBypass;         // 旁路时钟
4     uint32_t SDIO_ClockPowerSave;      // 节能模式
5     uint32_t SDIO_BusWide;             // 数据宽度
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制
7     uint8_t  SDIO_ClockDiv;            // 时钟分频
8 } SDIO_InitTypeDef;
```

- **SDIO_ClockPowerSave:** 节能模式选择，可选使能或禁用，它设定SDIO_CLKCR寄存器的PWRSVAV位的值。如果使能节能模式，CLK线只有在总线激活时才有时钟输出；如果禁用节能模式，始终使能CLK线输出时钟。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {  
2     uint32_t SDIO_ClockEdge;           // 时钟沿  
3     uint32_t SDIO_ClockBypass;        // 旁路时钟  
4     uint32_t SDIO_ClockPowerSave;     // 节能模式  
5     uint32_t SDIO_BusWide;            // 数据宽度  
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制  
7     uint8_t  SDIO_ClockDiv;           // 时钟分频  
8 } SDIO_InitTypeDef;
```

- **SDIO_BusWide:** 数据线宽度选择，可选1位数据总线、4位数据总线或8为数据总线，系统默认使用1位数据总线，操作SD卡时在数据传输模式下一般选择4位数据总线。它设定SDIO_CLKCR寄存器的WIDBUS位的值。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {  
2     uint32_t SDIO_ClockEdge;           // 时钟沿  
3     uint32_t SDIO_ClockBypass;        // 旁路时钟  
4     uint32_t SDIO_ClockPowerSave;     // 节能模式  
5     uint32_t SDIO_BusWide;            // 数据宽度  
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制  
7     uint8_t  SDIO_ClockDiv;           // 时钟分频  
8 } SDIO_InitTypeDef;
```

- **SDIO_HardwareFlowControl:** 硬件流控制选择，可选使能或禁用，它设定SDIO_CLKCR寄存器的HWFC_EN位的值。硬件流控制功能可以避免FIFO发送上溢和下溢错误。

SDIO—SD卡读写测试



SDIO初始化结构体

SDIO初始化结构体用于配置SDIO基本工作环境，比如时钟分频、时钟沿、数据宽度等等。

```
1 typedef struct {  
2     uint32_t SDIO_ClockEdge;           // 时钟沿  
3     uint32_t SDIO_ClockBypass;        // 旁路时钟  
4     uint32_t SDIO_ClockPowerSave;     // 节能模式  
5     uint32_t SDIO_BusWide;            // 数据宽度  
6     uint32_t SDIO_HardwareFlowControl; // 硬件流控制  
7     uint8_t  SDIO_ClockDiv;           // 时钟分频  
8 } SDIO_InitTypeDef;
```

- **SDIO_ClockDiv:** 时钟分频系数，它设定SDIO_CLKCR寄存器的CLKDIV位的值，设置SDIOCLK与CLK线输出时钟分频系数：

CLK线时钟频率=SDIOCLK/([CLKDIV+2])。

SDIO—SD卡读写测试



SDIO命令初始化结构体

SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。

```
1 typedef struct {  
2     uint32_t SDIO_Argument; // 命令参数  
3     uint32_t SDIO_CmdIndex; // 命令号  
4     uint32_t SDIO_Response; // 响应类型  
5     uint32_t SDIO_Wait;      // 等待使能  
6     uint32_t SDIO_CPSM;      // 命令路径状态机  
7 } SDIO_CmdInitTypeDef;
```

- **SDIO_Argument:** 作为命令的一部分发送到卡的命令参数，它设定SDIO参数寄存器(SDIO_ARG)的值。

SDIO—SD卡读写测试



SDIO命令初始化结构体

SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。

```
1 typedef struct {  
2     uint32_t SDIO_Argument; // 命令参数  
3     uint32_t SDIO_CmdIndex; // 命令号  
4     uint32_t SDIO_Response; // 响应类型  
5     uint32_t SDIO_Wait;      // 等待使能  
6     uint32_t SDIO_CPSM;      // 命令路径状态机  
7 } SDIO_CmdInitTypeDef;
```

- **SDIO_CmdIndex:** 命令号选择，它设定SDIO命令寄存器(SDIO_CMD)的CMDINDEX位的值。

SDIO—SD卡读写测试



SDIO命令初始化结构体

SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。

```
1 typedef struct {  
2     uint32_t SDIO_Argument; // 命令参数  
3     uint32_t SDIO_CmdIndex; // 命令号  
4     uint32_t SDIO_Response; // 响应类型  
5     uint32_t SDIO_Wait;      // 等待使能  
6     uint32_t SDIO_CPSM;      // 命令路径状态机  
7 } SDIO_CmdInitTypeDef;
```

- **SDIO_Response:** 响应类型，SDIO定义两个响应类型：长响应和短响应。根据命令号选择对应的响应类型。SDIO定义了四个32位的SDIO响应寄存器(SDIO_RESPx,x=1..4)，短响应只用到SDIO_RESP1。

SDIO—SD卡读写测试



SDIO命令初始化结构体

SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。

```
1 typedef struct {  
2     uint32_t SDIO_Argument; // 命令参数  
3     uint32_t SDIO_CmdIndex; // 命令号  
4     uint32_t SDIO_Response; // 响应类型  
5     uint32_t SDIO_Wait;      // 等待使能  
6     uint32_t SDIO_CPSM;      // 命令路径状态机  
7 } SDIO_CmdInitTypeDef;
```

- **SDIO_Wait:** 等待类型选择，有三种状态可选，一种是无等待状态，超时检测功能启动；一种是等待中断，另外一种等待传输完成。它设定SDIO_CMD寄存器的WAITPEND位和WAITINT位的值。

SDIO—SD卡读写测试



SDIO命令初始化结构体

SDIO命令初始化结构体用于设置命令相关内容，比如命令号、命令参数、响应类型等等。

```
1 typedef struct {  
2     uint32_t SDIO_Argument; // 命令参数  
3     uint32_t SDIO_CmdIndex; // 命令号  
4     uint32_t SDIO_Response; // 响应类型  
5     uint32_t SDIO_Wait;      // 等待使能  
6     uint32_t SDIO_CPSM;      // 命令路径状态机  
7 } SDIO_CmdInitTypeDef;
```

- **SDIO_CPSM:** 命令路径状态机控制，可选使能或禁用CPSM。它设定SDIO_CMD寄存器的CPSMEN位的值。

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {  
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时  
3     uint32_t SDIO_DataLength;     // 数据长度  
4     uint32_t SDIO_DataBlockSize;  // 数据块大小  
5     uint32_t SDIO_TransferDir;    // 数据传输方向  
6     uint32_t SDIO_TransferMode;   // 数据传输模式  
7     uint32_t SDIO_DPSM;           // 数据路径状态机  
8 } SDIO_DataInitTypeDef;
```

- **SDIO_DataTimeOut:** 设置数据传输以卡总线时钟周期表示的超时周期，它设定SDIO数据定时器寄存器(SDIO_DTIMER)的值。在DPSM进入Wait_R或繁忙状态后开始递减，直到0还处于以上两种状态则将超时状态标志置1.

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {  
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时  
3     uint32_t SDIO_DataLength;     // 数据长度  
4     uint32_t SDIO_DataBlockSize;  // 数据块大小  
5     uint32_t SDIO_TransferDir;     // 数据传输方向  
6     uint32_t SDIO_TransferMode;    // 数据传输模式  
7     uint32_t SDIO_DPSM;           // 数据路径状态机  
8 } SDIO_DataInitTypeDef;
```

- **SDIO_DataLength:** 设置传输数据长度，它设定SDIO数据长度寄存器(SDIO_DLEN)的值。

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时
3     uint32_t SDIO_DataLength;     // 数据长度
4     uint32_t SDIO_DataBlockSize;  // 数据块大小
5     uint32_t SDIO_TransferDir;    // 数据传输方向
6     uint32_t SDIO_TransferMode;   // 数据传输模式
7     uint32_t SDIO_DPSM;           // 数据路径状态机
8 } SDIO_DataInitTypeDef;
```

- **SDIO_DataBlockSize:** 设置数据块大小，有多种尺寸可选，不同命令要求的数据块可能不同。它设定SDIO数据控制寄存器(SDIO_DCTRL)寄存器的DBLOCKSIZE位的值。

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {  
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时  
3     uint32_t SDIO_DataLength;     // 数据长度  
4     uint32_t SDIO_DataBlockSize;  // 数据块大小  
5     uint32_t SDIO_TransferDir;    // 数据传输方向  
6     uint32_t SDIO_TransferMode;   // 数据传输模式  
7     uint32_t SDIO_DPSM;           // 数据路径状态机  
8 } SDIO_DataInitTypeDef;
```

- **SDIO_TransferDir:** 数据传输方向，可选从主机到卡的写操作，或从卡到主机的读操作。它设定SDIO_DCTRL寄存器的DTDIR位的值。

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {  
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时  
3     uint32_t SDIO_DataLength;     // 数据长度  
4     uint32_t SDIO_DataBlockSize;  // 数据块大小  
5     uint32_t SDIO_TransferDir;    // 数据传输方向  
6     uint32_t SDIO_TransferMode;   // 数据传输模式  
7     uint32_t SDIO_DPSM;           // 数据路径状态机  
8 } SDIO_DataInitTypeDef;
```

- **SDIO_TransferMode:** 数据传输模式，可选数据块或数据流模式。对于SD卡操作使用数据块类型。它设定SDIO_DCTRL寄存器的DTMODE位的值。

SDIO—SD卡读写测试



SDIO数据初始化结构体

SDIO数据初始化结构体用于配置数据发送和接收参数，比如传输超时、数据长度、传输模式等等。它被SDIO_DataConfig函数使用。

```
1 typedef struct {  
2     uint32_t SDIO_DataTimeOut;    // 数据传输超时  
3     uint32_t SDIO_DataLength;     // 数据长度  
4     uint32_t SDIO_DataBlockSize;  // 数据块大小  
5     uint32_t SDIO_TransferDir;    // 数据传输方向  
6     uint32_t SDIO_TransferMode;   // 数据传输模式  
7     uint32_t SDIO_DPSM;           // 数据路径状态机  
8 } SDIO_DataInitTypeDef;
```

- **SDIO_DPSM:** 数据路径状态机控制，可选使能或禁用DPSM。它设定SDIO_DCTRL寄存器的DTEN位的值。要实现数据传输都必须使能SDIO_DPSM。

零死角玩转STM32



THANKS

论坛 : www.firebbs.c

淘宝 : firestm32.taobao.com



扫描进入淘宝店铺