

SDMMC controller API 说明:

1. *void FSdmmcPs_initDev(FSdmmcPs_T *pSdmmc,*
*FSdmmcPs_Instance_T *pInstance,*
u8 dev_index)

描述	* This function is initialize the SMC device.
参数	* @param * pSmc is the pointer to a smc device. * pInstance is the pointer to instance structure.
返回值	* @return * NA.

2. *int FSdmmcPs_cardDetect(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the card detect value.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * card detect value.

3. *void FSdmmcPs_resetController(u8 sd_index);*

描述	* Reset the SDMMC controller by SLCR registers.
参数	* @param * sd_index select the SDMMC controller.
返回值	* @return * NA.

4. *void FSdmmcPs_instanceReset(FSdmmcPs_T *pSdmmc, FDmaPs_T *pDmac);*

描述	* This function reset the sdmmc instance.
参数	* @param * pSdmmc is the pointer to a sd card device. * pDmac is the pointer to the DMA controller device.
返回值	* @return * NA.

5. *void FSdmmcPs_setDmaMsize(FSdmmcPs_T *pSdmmc, u32 dma_msize);*

描述	* This function sets the burst size of multipul transaction used with DMA.
参数	* @param * pSdmmc is the pointer to a sd card device. * dma_msize is burst size used with DMA.
返回值	* @return * NA.

6. *void FSdmmcPs_setRXwmark(FSdmmcPs_T *pSdmmc, u16 rx_wmark);*

描述	* This function sets the FIFO threshold watermark level * when receiving data from card.
参数	* @param * pSdmmc is the pointer to a sd card device. * rx_wmark is the watermark level of receiver FIFO threshold.
返回值	* @return * NA.

7. *void FSdmmcPs_setTXwmark(FSdmmcPs_T *pSdmmc, u16 tx_wmark);*

描述	* This function sets the FIFO threshold watermark level * when transmitting data to card.
参数	* @param * pSdmmc is the pointer to a sd card device. * tx_wmark is the watermark level of transmit FIFO threshold.
返回值	* @return * NA.

8. *void FSdmmcPs_powerDown(FSdmmcPs_T *pSdmmc);*

描述	* This function power down the selected card.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

9. *void FSdmmcPs_powerUp(FSdmmcPs_T *pSdmmc);*

描述	* This function power up the selected card.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

10. *int FSdmmcPs_setClock(FSdmmcPs_T *pSdmmc, int rate);*

描述	* This function sets the card's clock frequency to given value.
参数	* @param

	<p>* pSdmmc is the pointer to a sd card device.</p> <p>* rate is clock frequency in Hz.</p>
返回值	<p>* @return</p> <p>* Returns 0 if the operation was successful.</p> <p>* otherwise returns an error code.</p>

11. *void FSdmmcPs_setCtrlBuswidth(FSdmmcPs_T *pSdmmc, unsigned int width);*

描述	* This function configures the bus width in card type register.
参数	<p>* @param</p> <p>* pSdmmc is the pointer to a sd card device.</p> <p>* width is bus width.</p>
返回值	<p>* @return</p> <p>* NA.</p>

12. *int FSdmmcPs_sdControllerReset(FSdmmcPs_T *pSdmmc);*

描述	* This function resets controller.
参数	<p>* @param</p> <p>* pSdmmc is the pointer to a sd card device.</p>
返回值	<p>* @return</p> <p>* Returns 0 if the operation was successful.</p> <p>* otherwise returns an error code.</p>

13. *int FSdmmcPs_sdFifoReset(FSdmmcPs_T *pSdmmc);*

描述	* This function resets FIFO.
参数	<p>* @param</p> <p>* pSdmmc is the pointer to a sd card device.</p>
返回值	* @return

	* Returns 0 if the operation was successful.
	* otherwise returns an error code.

14. *int FSdmmcPs_sdDmaReset(FSdmmcPs_T *pSdmmc);*

描述	* This function resets DMA interface.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

15. *u32 FSdmmcPs_getIntMask(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the interrupt mask value.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * interrupt mask value.

16. *void FSdmmcPs_setIntMask(FSdmmcPs_T *pSdmmc, u32 int_mask);*

描述	* This function sets the interrupt mask value.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

17. *u32 FSdmmcPs_getRawIntStat(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the raw interrupt status value.
----	---------------------------------------------------------

参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * raw interrupt status value.

18. void FSdmmcPs_setRawIntStat(FSdmmcPs_T *pSdmmc, u32 raw_int);

描述	* This function sets the raw interrupt status value.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

19. u32 FSdmmcPs_getMaskIntStat(FSdmmcPs_T *pSdmmc);

描述	* This function returns the masked interrupt status value.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * masked interrupt status value.

20. void FSdmmcPs_enableInt(FSdmmcPs_T *pSdmmc);

描述	* This function enables the global interrupt bit.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

21. *void FSdmmcPs_enableDMA(FSdmmcPs_T *pSdmmc);*

描述	* This function enables DMA transfer mode.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

22. *void FSdmmcPs_disableIntMask(FSdmmcPs_T *pSdmmc);*

描述	* This function masks all interrupt except card detect(CD) interrupt.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * NA.

23. *u8 FSdmmcPs_getTransferMode(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the data transfer mode.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * data transfer mode.

24. *u32 FSdmmcPs_prepareCmd(u32 cardType, char cmd);*

描述	* This function frames the command register for a particular command.
参数	* @param * cardType is the type of card. * cmd is the command to be sent.

返回值	* @return * Command register value.
-----	--------------------------------------------

25. *int FSdmmcPs_sendCmd(FSdmmcPs_T *pSdmmc);*

描述	* This function does card command generation.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

26. *unsigned int FSdmmcPs_getCardSizeKB(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the total card size in Kbytes.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * card size.

27. *unsigned int FSdmmcPs_getBlockNumbers(FSdmmcPs_T *pSdmmc);*

描述	* This function returns the block number.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * block number.

28. *unsigned int FSdmmcPs_getCardStatus(FSdmmcPs_T *pSdmmc);*

描述	* This function returns card status.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * card status.

29. *int FSdmmcPs_setBlockLen(FSdmmcPs_T *pSdmmc, unsigned int len);*

描述	* This function sets the block length to the card.
参数	* @param * pSdmmc is the pointer to a sd card device. * len is block length in bytes.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

30. *int FSdmmcPs_setCardBusWidth(FSdmmcPs_T *pSdmmc, unsigned int bus_width);*

描述	* This function sets the bus width to the card.
参数	* @param * pSdmmc is the pointer to a sd card device. * bus_width is bus width.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

31. *int FSdmmcPs_sdEnumerate(FSdmmcPs_T *pSdmmc);*

描述	<ul style="list-style-type: none"> * This function runs the SD card enumeration sequence. * This function runs after the initialization and identification procedure. * It gets all necessary information from the card.
参数	<ul style="list-style-type: none"> * @param * pSdmmc is the pointer to a sd card device.
返回值	<ul style="list-style-type: none"> * @return * Returns 0 if the operation was successful. * otherwise returns an error code.

32. *int FSdmmcPs_mmcEnumerate(FSdmmcPs_T *pSdmmc);*

描述	<ul style="list-style-type: none"> * This function runs the MMC card and emmc enumeration sequence. * This function runs after the initialization and identification procedure. * It gets all necessary information from the card.
参数	<ul style="list-style-type: none"> * @param * pSdmmc is the pointer to a sd card device.
返回值	<ul style="list-style-type: none"> * @return * Returns 0 if the operation was successful. * otherwise returns an error code.

33. *int FSdmmcPs_cardIdentify(FSdmmcPs_T *pSdmmc);*

描述	<ul style="list-style-type: none"> * This function runs the initialisation procedure and the * identification process, then it sets the card in transfer state to * set the block length and the bus width, at last change the card clock * to high clock.
----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

34. *int FSdmmcPs_doTransfer(FSdmmcPs_T *pSdmmc, unsigned int address, unsigned int len, int rw);*

描述	* This function moves card to transfer state. The buffer size must be at * least 512 byte long.
参数	* @param * pSdmmc is the pointer to a sd card device. * address is address of the block to transfer. * len is number of bytes to be transfered. * rw is 1 for read data and 0 for write data.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

35. *int FSdmmcPs_stopTransfer(FSdmmcPs_T *pSdmmc);*

描述	* This function runs to force the card to stop transmission.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

36. *int FSdmmcPs_readFun(FSdmmcPs_T *pSdmmc, u32 address, char *pBuf, u32 len, enum SDMMC_TransMode mode);*

描述	<p>* This function performs data read transmission.The buffer size must be at</p> <p>* least 512 byte long.</p> <p>* User can select the transfer mode to use with DMA or not.</p>
参数	<p>* @param</p> <p>* pSdmmc is the pointer to a sd card device.</p> <p>* address is address of the block to read.</p> <p>* pBuf is read data buffer, its size at least the block size.</p> <p>* len is number of bytes to be transfered.</p> <p>* trans_mode is enumerated data transfer mode.</p>
返回值	<p>* @return</p> <p>* Returns 0 if the operation was successful.</p> <p>* otherwise returns an error code.</p>

37. *int FSdmmcPs_writeFun(FSdmmcPs_T *pSdmmc, u32 address, char *pBuf, u32 len, enum SDMMC_TransMode mode);*

描述	<p>* This function performs data write transmission.The buffer size must be at</p> <p>* least 512 byte long.</p> <p>* User can select the transfer mode to use with DMA or not.</p>
参数	<p>* @param</p> <p>* pSdmmc is the pointer to a sd card device.</p> <p>* address is address of the block to write.</p> <p>* pBuf is read data buffer, its size at least the block size.</p>

	* len is number of bytes to be transfered. * trans_mode is enumerated data transfer mode.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

38. *int FSdmmcPs_addSdDevice(FSdmmcPs_T *pSdmmc);*

描述	* This function adds a new SD/MMC/EMMC card.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

39. *int FSdmmcPs_removeSdDevice(FSdmmcPs_T *pSdmmc);*

描述	* This function removes a SD/MMC/EMMC card.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

40. *s32 FSdmmcPs_initCard(FSdmmcPs_T *pSdmmc);*

描述	* This function initializes a specific card.
参数	* @param * pSdmmc is the pointer to a sd card device.

返回值	* @return * FMSH_SUCCESS if the operation was successful. * FMSH_FAILURE if the operation was failed.
-----	---------------------------------------------------------------------------------------------------------------------

41. *s32 FSdmmcPs_cfgInitialize(FSdmmcPs_T *pSdmmc, FDmaPs_T *pDmac, enum SDMMC_TransMode trans_mode);*

描述	* This function initializes a specific sdmmc instance * such that the driver is ready to use.
参数	* @param * pSdmmc is the pointer to a sd card device. * pDmac is the pointer to the DMA controller device. * trans_mode is enumerated data transfer mode.
返回值	* @return * FMSH_SUCCESS if the operation was successful. * FMSH_FAILURE if the operation was failed.

42. *int SDMMC_decodeCID(FSdmmcPs_T *pSdmmc);*

描述	* This function decodes the raw CID information.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

43. *int SDMMC_decodeCSD(FSdmmcPs_T *pSdmmc);*

描述	* This function decodes the raw CSD information.
----	--------------------------------------------------

参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

44. *int SDMMC_decodeSCR(FSdmmcPs_T *pSdmmc);*

描述	* This function decodes the raw SCR information.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.

45. *int SDMMC_getCSDInfo(FSdmmcPs_T *pSdmmc);*

描述	* This function decodes the raw SCR information.
参数	* @param * pSdmmc is the pointer to a sd card device.
返回值	* @return * Returns 0 if the operation was successful. * otherwise returns an error code.