

eMMC : Kingston KE4CN3H5A , 8GB , eMMC45 , www.kingston.com

```
Protocol analyzer data on function : bool _io_emmc_esdhc_init ( MQX_FILE_PTR fd_ptr, uint32_t
baudrate )
{
```

```
/* CMD0 - Go to idle - reset card */
command.COMMAND = ESDHC_CREATE_CMD(0, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_NO, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = 0;
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}
```

Packet #	Name	TimeStamp	Start	Host	CMD0(GO_IDLE_STATE)	Content32	CRC7	Stop
1	SD(eMMC)	-1.25us	Start	Host	00	00000000	4A	Stop

```
/* CMD1 - Operating voltage range validation (MMC will respond to nothing before seeing this) */
do {
command.COMMAND = ESDHC_CREATE_CMD(1, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R4, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = 0x40FF8000; /* specifies voltage range and sector access mode - see
JEDEC 84-A44 Section 8.1 */
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}
} while( (command.RESPONSE[0] & 0x80000000) == 0 );
```

1st capture :

Packet #	Name	TimeStamp	Start	Host	CMD1(SEND_OF_COND)	Content32	CRC7	Stop						
1	SD(eMMC)	-1.25us	Start	Host	01	40FF8000	05	Stop						
2	SD(eMMC)	131.46us	Start	Device	3F	0	0	00	1FF	00	1	00	7F	Stop

2nd capture :

Packet #	Name	TimeStamp	Start	Host	CMD1(SEND_OF_COND)	Content32	CRC7	Stop						
1	SD(eMMC)	-1.25us	Start	Host	01	40FF8000	05	Stop						
2	SD(eMMC)	131.46us	Start	Device	3F	1	2	00	1FF	00	1	00	7F	Stop

Note : Busy[31] change from 0 to 1

```

/* Card identify */
command.COMMAND = ESDHC_CREATE_CMD(2, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R2, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = 0;
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}

```

Packet #	Name	TimeStamp	Start	Host	Host	Cmd02(ALL_SEND_CMD)	Content32	CRC7	Stop
1	SD(eMMC)	-1.255us	Start	Host	Host	02	00000000	26	Stop

Packet #	Name	TimeStamp	Start	Device	Check(R2)	MID[127:120]	Reserved[119:114]	CBX[113:112]	OID[111:104]	PNM[103:56]	PRV[55:48]	PSN[47:16]	MDT[15:8]	CRC[7:1]	Stop
2	SD(eMMC)	131.455us	Start	Device	3F	70	00	BGA	00	MMC08G	5.8	017800AF	1998.7	10	Stop

Note : PNM MMC08G (Kingston eMMC Part Number for KE4CN3H5A)

```

/* Set relative card address */
sdcard_ptr->ADDRESS = 0x00010000; //Arbitrarily select address 1
command.COMMAND = ESDHC_CREATE_CMD(3, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = sdcard_ptr->ADDRESS;
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}

```

Packet #	Name	TimeStamp	Start	Host	Host	Cmd03(SET_RELATIVE_ADDR)	Content32	CRC7	Stop												
1	SD(eMMC)	-1.25us	Start	Host	Host	03	00010000	3F	Stop												
2	SD(eMMC)	131.46us	Start	Device	03	0	0	0	0	0	0	0	0	0	0	0					
										CC_ERR[28]	ERR[19]	Reserved[18:17]	CS0/CD_OV[16]	WP_ERR_SK[15]	Reserved[14]	ERR_RES[13]	CUR_STA[12:9]	RD_FOR_DAT[8]	SW_ERR[7]	EXC_EVENT[6]	
										0	0	0	0	0	0	0	0	Ident	1	0	0
APP_CMD[5]	Reserved[4:0]	CRC7	Stop																		
0	00	70	Stop																		

```

/* Select card */
command.COMMAND = ESDHC_CREATE_CMD(7, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1b, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = sdcard_ptr->ADDRESS;
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}

```

Packet #	Name	TimeStamp	Start	Host	CMD7(SELECT/DESELECT_CARD)	Content32	CRC7	Stop								
1	SD(eMMC)	-1.255us	Start	Host	07	00010000	6E	Stop								
Packet #	Name	TimeStamp	Start	Host	CMD7(SELECT/DESELECT_CARD)	ADD_OUT_OF_RA[31]	ADDR_MIS[30]	BL_LEA_ERR[29]	ERR_SEQ_ERR[28]	ERR_PAR[27]	WP_VIO[26]	DEV_IS_LOCK[25]	LO_UNFAC[24]			
2	SD(eMMC)	131.455us	Start	Device	07	0	0	0	0	0	0	0	0			
					CMD16(SET_BLOCKLEN)	ADD_OUT_OF_RA[31]	ADDR_MIS[30]	BL_LEA_ERR[29]	ERR_SEQ_ERR[28]	ERR_PAR[27]	WP_VIO[26]	DEV_IS_LOCK[25]	LO_UNFAC[24]			
					10	0	0	0	0	0	0	0	0			
						CC_ERR[20]	ERR[19]	Reserved[18:17]	CS0_CD_OVE[16]	WP_ERR_3X[15]	Reserved[14]	ERR_RES[13]	CUR_STAT[12:9]	NO_FOR_DATA[8]	SW_ERR[7]	EXC_EVENT[6]
						0	0	0	0	0	0	0	Stby	1	0	0
					APP_CMD[5]	Reserved[4:0]	CRC7	Stop								
					0	00	3A	Stop								

```

/* Set the block size */
command.COMMAND = ESDHC_CREATE_CMD(16, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = IO_SDCARD_BLOCK_SIZE;
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdcard_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}

```

Packet #	Name	TimeStamp	Start	Host	CMD16(SET_BLOCKLEN)	Content32	CRC7	Stop								
1	SD(eMMC)	-1.255us	Start	Host	10	00000200	0A	Stop								
Packet #	Name	TimeStamp	Start	Device	CMD16(SET_BLOCKLEN)	ADD_OUT_OF_RA[31]	ADDR_MIS[30]	BL_LEA_ERR[29]	ERR_SEQ_ERR[28]	ERR_PAR[27]	WP_VIO[26]	DEV_IS_LOCK[25]	LO_UNFAC[24]			
2	SD(eMMC)	131.455us	Start	Device	10	0	0	0	0	0	0	0	0			
						CC_ERR[20]	ERR[19]	Reserved[18:17]	CS0_CD_OVE[16]	WP_ERR_3X[15]	Reserved[14]	ERR_RES[13]	CUR_STAT[12:9]	NO_FOR_DATA[8]	SW_ERR[7]	EXC_EVENT[6]
						0	0	0	0	0	0	0	Tran	1	0	0
					APP_CMD[5]	Reserved[4:0]	CRC7	Stop								
					0	00	05	Stop								

Enlarge :

Packet #	Name	TimeStamp	Start	Host	CMD16(SET_BLOCKLEN)	Content32	CRC7	Stop
1	SD(eMMC)	-1.255us	Start	Host	10	00000200	0A	Stop

Content32 : 0x00000200 - > 512

```

/* Get the EXT_CSD data, (responds with R1, the EXT_CSD is sent as a block of data */
/*command.COMMAND = ESDHC_CREATE_CMD(8, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_DATA_READ_FLAG ); */
command.COMMAND = ESDHC_CREATE_CMD(8, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_DATA_CMD_FLAG |
ESDHC_COMMAND_DATA_READ_FLAG );
command.ARGUMENT = 0;
command.BLOCKS = 1;
/* command.BLOCKSIZE = IO_SDCARD_BLOCK_SIZE; */
command.BLOCKSIZE = 512; /* hard code 512 to IO_SDCARD_BLOCK_SIZE */
int32_t res = ioctl (sdc_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command);
if (ESDHC_OK != res)
{
return FALSE;
}
if (512 != read (sdc_ptr->COM_DEVICE, buffer, IO_SDCARD_BLOCK_SIZE))
{
return FALSE;
}

sdc_ptr->NUM_BLOCKS = *((uint32_t *)&buffer[212]);

```

Packet #	Name	TimeStamp	Start	Host	CMD6(SEN1_EXT_CSD)	Command32	CRC7	Stop							
1	SD(eMMC)	-1.25us	Start	Host	08	00000000	61	Stop							
Packet #	Name	TimeStamp	Start	Device	CMD6(SEN1_EXT_CSD)	ADD_OUT_OP_RA(31)	ADDR_MIS(30)	BL_LEN_ERR(29)	ERA_SEQ_ERR(28)	ERA_PAR(27)	WP_VID(26)	DEV_IS_LOCK(25)	IO_UN_FAI(24)		
2	SD(eMMC)	131.46us	Start	Device	08	0	0	0	0	0	0	0	0		
		COM_CRC_ERR(23)	BL_CMD(22)	DEV_ECC_FAI(21)	CC_ERR(20)	ERR(19)	Reserved(18-17)	CSD/CD_OVE(16)	WP_ERA_SK(15)	Reserved(14)	ERA_RES(13)	CLR_STA(12-9)	RD_FDR_DAT(8)	SW_ERR(7)	EXC_EVENT(6)
		0	0	0	0	0	0	0	0	0	0	Tran	1	0	0
		APP_CMD(5)	Reserved(4:0)	CRC7	Stop										
		0	00	7a	Stop										

```

/* CMD6 - SWITCH to high speed mode */
command.COMMAND = ESDHC_CREATE_CMD(6, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_NONE_FLAG);
command.ARGUMENT = 0x03B90100; /* set the HIGH SPEED mode byte*/
command.BLOCKS = 0;
if (ESDHC_OK != ioctl (sdc_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}

```

Packet #	Name	TimeStamp	Start	Host	CMD6(SW(TCP))	Command32	CRC7	Stop							
1	SD(eMMC)	-1.25us	Start	Host	06	03B90100	17	Stop							
Packet #	Name	TimeStamp	Start	Device	CMD6(SW(TCP))	ADD_OUT_OP_RA(31)	ADDR_MIS(30)	BL_LEN_ERR(29)	ERA_SEQ_ERR(28)	ERA_PAR(27)	WP_VID(26)	DEV_IS_LOCK(25)	IO_UN_FAI(24)	COM_CRC_ERR(23)	
2	SD(eMMC)	131.46us	Start	Device	06	0	0	0	0	0	0	0	0	0	
		BL_CMD(22)	DEV_ECC_FAI(21)	CC_ERR(20)	ERR(19)	Reserved(18-17)	CSD/CD_OVE(16)	WP_ERA_SK(15)	Reserved(14)	ERA_RES(13)	CLR_STA(12-9)	RD_FDR_DAT(8)	SW_ERR(7)	EXC_EVENT(6)	APP_CMD(5)
		0	0	0	0	0	0	0	0	0	Tran	0	0	0	0
		Reserved(4:0)	CRC7	Stop											
		00	65	Stop											

```

// Adjust bus width
command.ARGUMENT = 0;
switch( sdc_card_ptr->INIT->SIGNALS ) {
default:
case ESDHC_BUS_WIDTH_1BIT:
/* SDCARD_DEBUG( "1 bit SDIO bus selected\n" ); */
/* nothing to do */
break;

case ESDHC_BUS_WIDTH_4BIT:
/*SDCARD_DEBUG( "4 bit SDIO bus selected\n" ); */
param = ESDHC_BUS_WIDTH_4BIT;
command.ARGUMENT = 0x03B70100;
break;

case ESDHC_BUS_WIDTH_8BIT:
/*SDCARD_DEBUG( "8 bit SDIO bus selected\n" ); */
param = ESDHC_BUS_WIDTH_8BIT;
command.ARGUMENT = 0x03B70200;
break;
}

if( command.ARGUMENT != 0 ) {
/* CMD6 - SWITCH bus width */
command.COMMAND = ESDHC_CREATE_CMD(6, ESDHC_COMMAND_TYPE_NORMAL,
ESDHC_COMMAND_RESPONSE_R1, ESDHC_COMMAND_NONE_FLAG);
command.BLOCKS = 0;
if (ESDHC_OK != ioctl( sdc_card_ptr->COM_DEVICE, IO_IOCTL_ESDHC_SEND_COMMAND,
&command))
{
return FALSE;
}
}

```

Packet #	Name	TimeStamp	Start	Host	CMD6(SWITCH)	Content32	CRC7	Stop
1	SD(eMMC)	-10.015us	Start	Host	06	03B70100	16	Stop
Packet #	Name	TimeStamp	Start	Device				
2	SD(eMMC)	1.051675ms	Start	Device				

