

广东建邦计算机软件有限公司

HD6P 底层开发接口 API

| 文档名称 | 版本 | 时间 | 修改人 | 说明 |
|--------------------|--------|----------|-----|--------------------|
| HD6P 底层开发接口 API | V1.0.0 | 2014-5-5 | | 提供 6P 设备硬件主要底层驱动接口 |
| | V1.0.1 | 2014-5-6 | | 增加 IAP 接口 |
| | | | | |

| | |
|----------------------|----|
| 1. 非接卡片接口..... | 2 |
| 2. LCD 接口..... | 6 |
| 3. PSAM 卡接口..... | 8 |
| 4. RTC 实时时钟接口..... | 9 |
| 5. 键盘接口..... | 10 |
| 6. SPI Flash 接口..... | 11 |
| 7. 定时器接口..... | 13 |
| 8. IAP 接口..... | 14 |

1. 非接卡片接口

M1 卡操作接口

```
/*
*****
** 函数名称   : HL_Active
** 函数功能   : 卡片激活(含寻卡\防冲突\选卡\密码验证等操作)
** 入口参数   : Block_Adr :块地址
                  hKey: KEYA(0x10) 或 KEYB(0x20)认证
                  cKey:密码值数组
** 出口参数   : 无
** 返    回   : 见返回值定义
*****
*/
unsigned char HL_Active(
    unsigned char Block_Adr,
    unsigned char hKey,
    unsigned char const *cKey);

/*
*****
** 函数名称   : MIF_ReadBlock
** 函数功能   : 读块数据
** 入口参数   : Adr:块号
** 出口参数   : buff:读出的数据
** 返    回   : 见返回值定义
*****
*/
unsigned char MIF_ReadBlock(
    unsigned char *buff,
    unsigned char Block_Adr);

/*
*****
** 函数名称   : MIF_WriteBlock
** 函数功能   : 写块数据
** 入口参数   : buff:要写入的数据  Adr:块号
** 出口参数   : 无
** 返    回   : 见返回值定义
*****
*/
unsigned char MIF_WriteBlock(
    unsigned char *buff,
    unsigned char Block_Adr);
```

```
/*
*****

** 函数名称   : MIF_ReadCard
** 函数功能   : 读块数据(已含激活功能)
** 入口参数   : cBlock: 读块数据,
                  : hKey 读块数据认证方式, KEYA_MODE 表示 Keya,
                  KEYB_MODE 表示          Keyb
                  : cKey 此块对应扇区的密钥
                  : pData 读出的块数据
                  : cTry 读卡失败后重复次数
** 出口参数   : 无
** 返    回   : 0: 见返回值定义
*****

*/

bool MIF_ReadCard(
                unsigned char cBlock,
                unsigned char hKey,
                unsigned char const *cKey,
                unsigned char *pData,
                unsigned char cTry);

/*
*****

** 函数名称   : MIF_ReadCard2
** 函数功能   : 读同一扇区前二块数据(已含激活功能)
** 入口参数   : cBlock: 块号
                  : hKey 读块数据认证方式, KEYA_MODE 表示 Keya,
                  KEYB_MODE 表示 Keyb
                  : cKey 此块对应扇区的密钥
                  : pData 读出的块数据
                  : cTry 读卡失败后重复次数
** 出口参数   : 无
** 返    回   : 0: 见返回值定义
*****

*/

bool MIF_ReadCard2(
                unsigned char cBlock,
                unsigned char hKey,
                unsigned char const *cKey,
                unsigned char *pData,
                unsigned char cTry);

/*
```

** 函数名称 : MIF_ReadCard3
** 函数功能 : 读同一扇区三块数据(已含激活功能)
** 入口参数 : cBlock : 块耗
 : hKey 读块数据认证方式, KEYA_MODE 表示 Keya,
 KEYB_MODE 表示 Keyb
 : cKey 此块对应扇区的密钥
 : pData 读出的块数据
 : cTry 读卡失败后重复次数
** 出口参数 : 无
** 返 回 : 见返回值定义

*/

```
bool MIF_ReadCard3(  
                    unsigned char cBlock,  
                    unsigned char hKey,  
                    unsigned char const *cKey,  
                    unsigned char *pData,  
                    unsigned char cTry);
```

/*

** 函数名称 : MIF_WriteCard
** 函数功能 : 写块数据(已含激活功能)
** 入口参数 : cBlock : 块号
 : hKey 写块数据认证方式, KEYA_MODE 表示 Keya,
 KEYB_MODE 表示 Keyb
 : cKey 此块对应扇区的密钥
 : pData 写入的块数据
 : cTry 写卡失败后重复次数
** 出口参数 : 无
** 返 回 : 见返回值定义

*/

```
bool MIF_WriteCard(  
                   unsigned char cBlock,  
                   unsigned char hKey,  
                   unsigned char const *cKey,  
                   unsigned char *pData,  
                   unsigned char cTry);
```

/*

广东建邦计算机软件有限公司

```
** 函数名称 : ReadICCardUID
** 函数功能 : 读卡序列号 4 个字节
** 入口参数 : cUID 成功则表示卡片序列号
** 出口参数 : 无
** 返 回 : 见返回值定义
```

```
*****
```

```
*/
```

```
bool ReadICCardUID(unsigned char *cUID);
```

非接 CPU 卡操作接口

```
/*
```

```
*****
```

```
** 函数名称 : ResetCard_A
** 函数功能 : 复位非接 CPU 卡
** 入口参数 : 无
** 出口参数 : 无
** 返 回 : TRUE:成功 False:失败
```

```
*****
```

```
*/
```

```
uint8 ResetCard_A(void);
```

```
/*
```

```
*****
```

```
Name:      CpuCard_Apdu
Function: 应用协议数据单元信息交换函数
InPara:    cCID:固定为 0x0F 不使用 CID
           cSendLen:发送数据长度
           cSendBuf:发送数据内容
```

```
OutPara:    cRcvLen: 接收数据长度
           cRcvBuf: 接收数据内容
```

```
return:    命令执行成功 0x9000
           发送命令失败 0x0001
           接收长度错误 0x0002
           协议错误      0x0003
           CID 超过最大值 0x0004
           其它的值错误代码
```

```
*****
```

```
*/
```

```
uint16 CpuCard_Apdu(uint8 cCID, uint8 cSendLen, uint8 *pSendBuf, uint8 *pRcvLen, uint8
*pRcvBuf);
```

```
/*
```

```
*****
```

```
** 函数名称 : Deselect_A
```

```
** 函数功能 : 取消被选择的卡片
** 入口参数 : 无
** 出口参数 : 无
** 返 回 : FM1715_OK:成功 FM1715_NOTAGERR:失败
*****

*/
uint8 Deselect_A(void);

/*
*****

** 函数名称 : Reset_Reader_TimeOut
** 函数功能 : 重设读头指令执行超时时间
** 入口参数 : 无
** 出口参数 : 无
** 返 回 : 无
*****

*/
void Reset_Reader_TimeOut(void);
```

2. LCD 接口

```
/*
*****

**函数名称:SetLcdContrast()
**函数功能:设置液晶对比度
**入口参数:cContrast 对比度:范围 0x20~0x3f,默认为 0x28
**出口参数:无
**返回:无
*****

*/
void SetLcdContrast(uint8 cContrast);

/*
*****

**函数名称:CLS()
**函数功能:清屏
**入口参数: CRow: 0~3, 0xFF 为清全屏
**出口参数:无
**返回:无
*****
```

```

*/
void LcdCLS(uint8 cRow);                                //LCD 清屏

/*
*****
**函数名称:LcdInit()
**函数功能:液晶初始化
**入口参数:无
**出口参数:无
**返回:无
*****
*/
void LcdInit(void);                                    // LCD 显示初始化

/*
*****
**函数名称:LcdPrint()
**函数功能:显示
**入口参数:cArray:列(0-15),cRow:行(0-3),String:字符串指针, cFlashnum:不起用此参数
**出口参数:无
**返回:无
*****
*/
void LcdPrint(uint8 cRow, uint8 cArray, char *String,uint8 cFlashnum);//LCD 输出显示

#define LCDLightON()      IO1SET = LCD_BAK // 开 LCD 背光
#define LCDLightOFF()     IO1CLR = LCD_BAK // 关 LCD 背光
#define LEDON()           IO0CLR = LED     // LED 开
#define LEDOFF()          IO0SET = LED     // LED 关

//字库起始地址    //each page 256B  ascii 码共 6Page,其它为汉字
#define FPA_ASCII_LIB_ADDR_START      (uint16)7122 // ASCII 码字库起始页面
#define FPA_FONT_ADDR_START           (uint16)7128 // 汉字库起始页面
#define FONTLIB_ADDR_END              (uint16)8192 // 汉字库结束页面
注:因 LCD 为 COG 结构,不带字库,字库需存放 Flash 空间

1. 串口通信接口
#define COM0    0    //串口 0
#define COM1    1    //串口 1
#define COMB    2    //两个串口
// 接收缓冲区大小
#define RS232_MAXBUF      (uint16)1024    //暂不能由用户修改

typedef void (*pREVFUNC)(void *pBuff,uint16 nlen);//串口回调函数

```

```
typedef struct tagDCB
{
    uint32          baudrate;          // 波特率
    char            datab;             // 字长度, 5/6/7/8 可选
    char            stopb;             // 停止位, 1/2 可选
    char            parity;            // 奇偶校验位, 0-无校验,1-奇校验,2-偶校验
    char            ruf;               // 保留
    uint16          limitlen;          // 长度阈值
    uint32          timeout;           // 接收超时时限
    pREVFUNC        pRecvfuc;         // 接收回调函数
}DCB,*pDCB;

/*
*****
** 函数名称 : StartComm()
** 函数功能 : 串口初始化,设置串口 0、1 的速率、位数、停止位、奇偶校验位
** 入口参数 : pComData :设置串口 0、1, lBaudBate :速率, cBits :数据位数, cStop :停止
            位数, cParty :奇校验位数等
** 出口参数 : 无
** 返    回 : 无
*****
*/
void StartComm(uint8 COM,pDCB pComData);

/*
*****
** 函数名称 : COM_SendBuf()
** 函数功能 : 向串口发送 n 字节数据
** 入口参数 : dat 要发送的数据
** 出口参数 : 无
** 返    回 : 0: 表示不成功, 1: 表示成功
*****
*/
uint8 COM_SendBuf(uint8 uComm,uint8 *pcSendBuf, int iLen);
```

3. PSAM 卡接口

```
/*
*****
** 函数名称 : Init_Psam
** 函数功能 : 初始化 psam 卡
** 入口参数 : 无
** 出口参数 : 无
*****
*/
```



```

*****
*/
void Init_Psam(void);
/*
*****
** 函数名称:Rst_Psam_Card
** 函数功能: 复位 PSAM_Card
** 入口参数:
** 出口参数:cData:复位信息 cLen:复位信息长度 s
** 返回 :1 表示成功,0 表示失败
*****
*/
uint8 Rst_Psam_Card(uint8 *cData, uint8 *cLen);

/*
*****
**函数名称：CosCommand
**函数功能：PSAM 卡 Apdu 指令操作函数
**入口参数：
**pSData :发送数据指针，cSNum：发送数据长度
**出口参数:pRData :接收数据指针，cRNum：接收数据长度指针
**返回：COS 指令返回码
*****
*/
uint16 CosCommand(unsigned char *pSData,unsigned char,unsigned char *pRData,unsigned char
*cRNum);

```

4. RTC 实时时钟接口

```

//系统时间结构
typedef struct
{
    uint16 chYear;           // 年
    uint8 chMonth;          // 月
    uint8 chDay;            // 日
    uint8 chHour;           // 时
    uint8 chMinute;         // 分
    uint8 chSecond;         // 秒
    uint8 chWeek;           // 周
}SYSTIME;
/*

```

```

*****
** 函数名称   : SetTime()
** 函数功能   : 设置时钟
** 入口参数   : sTime 要设置的时间
** 出口参数   :
*****
*/
uint8 SetTime (SYSTIME *sTime);

/*
*****
** 函数名称   : GetTime()
** 函数功能   : 获取时钟
** 入口参数   : sTime 获得的时间
** 出口参数   :
*****
*/
uint8 GetTime (SYSTIME *sTime);

/*
*****
** 函数名称   : IntiWatchDog()
** 函数功能   : 初始化看门狗
** 入口参数   : 无
** 出口参数   : 无
*****
*/
void IntiWatchDog (void);

//喂狗宏定义 在溢出周期内必须喂狗，否则程序重启
#define KillWatchDog() {IRQDisable();WDFEED = 0XAA;WDFEED = 0X55; IRQEnable();}

```

5. 键盘接口

```

/*
*****
** 函数名称   : SCKEYInit()
** 函数功能   : SC 设备按键初始化
** 入口参数   : bIsBeep:按键是否鸣叫 非 0 按键鸣叫
** 出口参数   : 无
*****
*/

```

```
void SCKEYInit(bool bIsBeep);    //7279 初始化
/*
*****
** 函数名称   : GetSCKEY()
** 函数功能   : 获取按键
** 入口参数   : 无
** 出口参数   : 无
** 返回       : 0 : 无键值
*****
*/
uint8 GetSCKEY(void)

/*
*****
函数名称 : LedPrint()
功能: LED 显示屏 显示字符串 ---单排
入口: row      ---   1   : 选中第 1 片 7279 ;
      非 1 : 选中第 2 片 7279 ;
      cWay   ---   0   : 方式 0 译码 ;
      非 0 : 方式 1 译码;
      *cBuf   ---  要显示数据的指针;
      cFlashNum ---  闪动显示的对应位
      cFlashNum : 0---无闪烁 1111 1111 : 分别对应 8 位闪烁
出口: 无
返回: 无
*****
*/
void LedPrint(uint8 cRow, uint8 cArray, char *String, uint8 cFlashNum);

/*
*****
** 函数名称   : GetS2KEY()
** 函数功能   : 获取按键
** 入口参数   : 无
** 出口参数   : 无
** 返回       : 0 : 见头文件定义
*****
*/
uint8 GetS2KEY(void)
```

6. SPI FLash 接口

Flash 相关参数:

广东建邦计算机软件有限公司

```
#ifndef USE_M25PE16                                /* The M25PE16 device */
#define EXPECTED_DEVICE (0x8015)                    /* Device code for the M25PE16 */
#define FLASH_SIZE (0x200000)                       /* Total device size in Bytes */
#define FLASH_PAGE_COUNT (0x2000)                  /* Total device size in Pages */
#define FLASH_SECTOR_COUNT (0x20)                  /* Total device size in Sectors */
#define FLASH_WRITE_BUFFER_SIZE 0x100 /* Write Buffer = 256 bytes */
#define FLASH_MWA 1                                 /* Minimum Write Access */

#define PW_TIMEOUT (0x01) /* Timeout in seconds suggested for Page Write Operation*/
#define PP_TIMEOUT (0x01) /*Timeout in seconds suggested for Page Program Operation*/
#define PE_TIMEOUT (0x01) /* Timeout in seconds suggested for Page Erase Operation*/
#define SE_TIMEOUT (0x06) /* Timeout in seconds suggested for Sector Erase Operation*/
#define BE_TIMEOUT (0x3D)/* Timeout in seconds suggested for Bulk Erase Operation*/

#endif /* USE_M25PE16 */
Flash 操作返回码
typedef enum {
    Flash_AddressInvalid,
    Flash_MemoryOverflow,
    Flash_PageEraseFailed,
    Flash_PageNrInvalid,
    Flash_SectorNrInvalid,
    Flash_FunctionNotSupported,
    Flash_NoInformationAvailable,
    Flash_OperationOngoing,
    Flash_OperationTimeOut,
    Flash_ProgramFailed,
    Flash_WrongType,
    Flash_Success
} Return_Type;
/*
*****
** 函数名称   : SF_Read()
** 函数功能   : 读 Flash 数据
** 入口参数   : page_addr:页面地址
                  in_page_byte_addr:页内地址
                  array:数据缓冲
                  counter:读取个数
** 出口参数   : 无
** 返回       : 见头文件定义
*****
*/
uint8 SF_Read(uint16 page_addr, uint16 in_page_byte_addr, uint8 *array, uint16 counter);
/*
```

```
*****
** 函数名称   : SF_Write()
** 函数功能   : 读 Flash 数据
** 入口参数   : page_addr:页面地址
                  in_page_byte_addr:页内地址
                  array:数据缓冲
                  coutner:写的个数
                  uIsErase:是否擦除写
** 出口参数   : 无
** 返回       : 见头文件定义
*****
*/

uint8 SF_Write(uint16 page_addr, uint16 in_page_begin_addr, uint8 *array, uint16 counter,uint8
uIsErase);
```

7. 定时器接口

```
typedef void (*OnTimer)(uint8 cTimerID);
/*
*****
** 函数名称   : SetTimer()
** 函数功能   : 设置定时器处理服务程序
** 入口参数   : uint8 nID 定时器 id 号  0~3
                  : uint32 lMillisecond 定时器定时间 ms 单位最少 10ms
                  : bRepeat 是否周期性执行还是一次性执行 ,true 为周期性执行 ,
                    false 为一次性执行
** 出口参数   : 无
*****
*/

void SetTimer(uint8 cID,uint32 lMillisecond,uint8 bRepeat,OnTimer pfunc);

/*
*****
** 函数名称   : KillTimer()
** 函数功能   : 关闭定时器
** 入口参数   : uint8 nID 定时器 id 号
** 出口参数   : 无
*****
*/

void KillTimer(uint8 nID);
```

8. IAP 接口

```
/*
*****
** 函数名称   : ReIAPFlash()
** 函数功能   : 在线编程 Flash 执行完毕程序重启 需要将所有程序写到 data falsh 后才能调用此函数
** 入口参数   : 无
** 出口参数   : 无
*****
*/
void ReIAPFlash(void);

//因 POS 程序过大，需要占用 data falsh 空间暂存
#define FPA_IAP_CONFIG          (uint16)6467          //存储程序大小
值为程序文件大小\512+程序文件大小%512 每次通信 512B
IAP 编程每次为 512B
#define FPA_IAP_START           (uint16)6468          //IAP 开始地址 511 页
#define FPA_IAP_END             (uint16)6978          //IAP 结束地址
```