



## Web 服务器终端接入接口 1.2

(比较 v1.0 版差别: 修改条目 5.6)

(比较 v1.1 版差别: 修改条目 5.6)

## 一、计算机软件接口协议

## 1. 发送用户登录 (计算机→数据中心)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e	NULL	NULL	0x50		24	用户 16bytes 密码 8bytes	

和校验: 从 IMEI 开始到正文结束所有字节的和取反 (包括命令字节)

## 2. 用户修改密码并登录 (计算机→数据中心)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e	NULL	NULL	0x57		32	用户 16bytes 旧密码 8bytes 新密码 8bytes	

接收应答 (数据中心→计算机)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e			0x51		1	返回值	

返回值: -1: 认证失败, 套接字错误需要重新建立连接 0: 认证失败; 2: 用户不存在; 3: 成功, 4: 密码错误;

5: 旧密码错误; 6: 新密码修改错误 7: 此用户在其他地方登录, 强制其他地方下线, 登录成功; 8: 系统正在维护中

## 3. 获取 DTU 信息

方式一: 命令获取 (计算机→数据中心) (IMEI 号为空则为查询所有 DTU 信息)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e			0x03		0	空	

应答 (数据中心→计算机)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e			0x05		2	状态 1byte 接入方式	

状态: 0x00: 无此DTU; 0x01: DTU在线; 0x02: DTU下线; 0x03: DTU异常

接入方式: enum \_INTERNET\_COMMUNICATION\_TYPE

```
{
    INTERNET_COMMUNICATION_TYPE_NONE=0, //未知
    INTERNET_COMMUNICATION_TYPE_NET_WIRE, //有线
    INTERNET_COMMUNICATION_TYPE_NET_WIFI, //wifi
    INTERNET_COMMUNICATION_TYPE_MOBILE_3G, //3G
    INTERNET_COMMUNICATION_TYPE_MOBILE_2G, //GPRS
    INTERNET_COMMUNICATION_TYPE_ADSL, //ADSL
}
```



INTERNET\_COMMUNICATION\_TYPE\_MODEM, //MODEL

INTERNET\_COMMUNICATION\_TYPE\_COM, //串口

};

方式二：从数据库读取方式：（数据中心可绑定该数据库，实时动态修改所有DTU的上下线状态）

数据库名:UserCenter(可根据配置文件更改)

表名:Cards

结构:

列名	数据库类型	允许 Null 值	说明
IMEI	Nvarchar(128)	否	主键，唯一标示
CardName	Int	是	别名
State	Int	否	1：上线，2：下线
AccessMethod	Int	否	
Del	Bit	否	1:已删除，2：未删除
Width	Int	是	
Height	Int	是	
DiskSize	Int	是	
MainVer	Int	是	
ChildVer	Int	是	
SoftwareVer	Int	是	
HardwareVer	Int	是	
Brightness	Int	是	
Company	Nvarchar(MAX)	是	
CardType_CardTypeId	Int	是	外键，卡的类别
CardGroup_CardGroupId	Int	是	外键，卡的分组

更新数据库使用的存储过程：

```
CREATE PROCEDURE update_dtu_status
```

```
@dtu_imei varchar(64),
```

```
/*对应字段：IMEI*/
```

```
@dtu_cardname varchar(64),
```

```
/*对应字段：CardName*/
```

```
@dtu_master varchar(64),
```

```
/*对应字段：Company*/
```

```
@dtu_status int,
```

```
/*对应字段：State*/
```

```
@dtu_accessmode int
```

```
/*对应字段：AccessMethod*/
```

```
AS
```

```
DECLARE @type_id int
```

```
IF @dtu_status=2
```

```
BEGIN
```

```
UPDATE cards SET state=@dtu_status WHERE imei=@dtu_imei
```

```
END
```

```
ELSE
```



```

BEGIN
    SELECT @type_id=CardTypeId FROM cardtypes WHERE
cardtypename=substring(@dtu_imei, 1, 3)
    IF @@ROWCOUNT=0
        BEGIN
            INSERT INTO cardtypes(cardtypename) VALUES (substring(@dtu_imei, 1, 3))
            SET @type_id=@@IDENTITY
        END
    SELECT * FROM cards where imei=@dtu_imei
    IF @@ROWCOUNT=0
        BEGIN
            INSERT INTO cards (imei,cardname,Company,state,accessmethod,
CardGroup_CardGroupId, CardType_CardTypeId) VALUES
(@dtu_imei,@dtu_cardname,@dtu_master,@dtu_status,@dtu_accessmode, 1, @type_id)
        END
    ELSE
        BEGIN
            UPDATE cards SET cardname=@dtu_cardname,state=@dtu_status,Company=
@dtu_master ,accessmethod=@dtu_accessmode, Del=0 WHERE imei=@dtu_imei
        END
END

```

删除 DTU 使用的存储过程:

```

CREATE PROCEDURE del_dtu
@dtu_imei varchar(64)
AS
BEGIN
UPDATE cards SET [State]=2, Del=1 WHERE imei=@dtu_imei
END

```

#### 4. 用户退出 (计算机→数据中心)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长 度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e	NULL	NULL	0x60		16	用户 16bytes	

接收应答 (数据中心→计算机)

协议头 4bytes	IMEI 16bytes	别名 16 bytes	命令 1byte	保留 1byte	正文长 度 2bytes	正文	校验 1byte
0x7e0x7e0x7e0x7e			0x61		1	0 成功 1 失败	

#### 5. PC 端收发数据协议 (数据中心↔计算机)

协议头 4bytes	IMEI	别名	命令	保留	正文长	正文	校验
------------	------	----	----	----	-----	----	----



	16bytes	16 bytes	1byte	1byte	度 2bytes		1byte
0x7e0x7e0x7e0x7e			0x00			5.1~5.6 蓝色字标注的内容	

### 5.1 获取终端信息

(web 服务器→终端)

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x17	0x00000000	sizeof(SYSTIMER)	SYSTIMER	

#### 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x57	0x00000000	sizeof(STRUCT_SYSTEM_STATUS)	STRUCT_SYSTEM_STATUS	

### 5.2 开关大屏幕

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x10	0x00000000	Sizeof (UINT)	0 关 1 开 UINT	

#### 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x50	0x00000000	Sizeof (UINT)	0 成功 1 失败 UINT	

### 5.3 校准时钟

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x11	0x00000000	sizeof(SYSTIMER)	SYSTIMER	

#### 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x51	0x00000000	Sizeof (UINT)	0 成功 1 失败 UINT	

### 5.4 调整亮度

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x12	0x00000000	Sizeof (UINT)	亮度值 UINT	

#### 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x52	0x00000000	Sizeof (UINT)	0 成功	



				1 失败 UINT	
--	--	--	--	--------------	--

## 5.5 点播控制卡内的节目

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0x97	0x00000000	Sizeof (UINT)	节目号 UINT	

## 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0XD7	0x00000000	正文字符串的长度	"Success" 成功 其他失败	

## 5.6 通知控制卡从指定 ftp 下载节目

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文										校验 1 bytes
0x7e0x7e0x55	0x05	0x00000000		Web 应答方式 0x01	域名字符串 (需'\0'结尾)64 字节	端口 INT	方法 网页字符串 (需'\0'结尾) 256 字节	节目大小	主版本	子版本	宽	高	节目名称字符串 (需'\0'结尾)64 字节	
				APP 应用程序应答方式 0x00										

## 应答

协议头 3bytes	命令 1byte	保留 4bytes	正文长度 4bytes	正文	校验 1bytes
0x7e0x7e0x55	0X45	0x00000000		0 成功 1 失败	

## 注释: Web 应答方式

0x01

域名: www.ledm2m.net

端口:8083

方法页面: GET \_/Task/UpdateState?imei=M10-550-00001&amp;taskId=1&amp;statId=



方法:

获取控制卡下载 ftp 节目的状态

服务器启动一个 web 服务响应控制卡的 web 请求, 根据 web 请求中的参数, 写任务状态数据库;

**控制卡端:**

控制卡终端卡返回节目下载状态 (更新 WEB 数据库)

1. 建立与网站的 socket, IP: WEB 的 IP 地址, Port: Web 的 Port;
2. 发送字符串:

"GET\_/Task/UpdateState?imei=M10-550-00001&taskId=1&stateId=2\_HTTP/1.1\r\nHost:www.ledm2m.net:8083\r\n\r\n", 其中的 "\_" 在实际字符串中为一个空格。

Host:www.ledm2m.net:8083 为网站的域名加端口号

**服务器端:**

接收到字符串说明:

- 此字符串为 HTTP 协议头,
- GET 意为请求方式,
- "/Task/UpdateState" 为网站根目录下的网址, 在此页面内写入更新数据库的方法, 接收3个参数imei (终端卡的序列号), taskId (任务的id), stateId (任务的状态ID), 例如: Update\_TaskState(imei, taskId, stateId);
- 带参数的 Get 请求格式为: WebPath?parameter1=value1&parameter2=value2& parameter3=value3&....
- "\r\n" 意为换行光标置首;
- 注意: 此字符串最后两个 "\r\n" 必不可少 (因为 HTTP 协议的规范要求)

注释: APP 应用程序应答方式

直接将状态通过接收到命令的套接字发送;

## 附注: 相关结构定义

```
typedef struct _SYSTIMER  
{
```

```
    int y; //获取年份  
    int m; //获取当前月份  
    int d; //获得几号  
    int h; //获取当前为几时  
    int mm; //获取分钟  
    int s; //获取秒
```



```
int w; //获取星期几, 注意1为星期天, 7为星期六  
}SYSTIMER,*P_SYSTIMER;
```

```
typedef struct _ST_VERSION  
{  
    int iMainVer;  
    int iChildVer;  
}ST_VERSION;  
typedef struct _ST_CARD_VERSION  
{  
    int iSoftwareVer;  
    int iHardwareVer;  
}ST_CARD_VERSION;  
typedef struct _STRUCT_SYSTEM_STATUS  
{  
    int iWidth;  
    int iHeight;  
    int iStatus;  
    int iDiskSize;  
    ST_VERSION stVersion;  
    ST_CARD_VERSION stCardVersion;  
    int iBrightValue;  
    int iPasswordId;//20100324  
    int iPassword;//20100324  
}STRUCT_SYSTEM_STATUS;
```

校验码计算函数

```
unsigned char GetCheckCodeIn8(unsigned char * pBuffer,unsigned int uiSize)  
{  
    unsigned int i=0;  
    unsigned char ucCheckCode=0;  
    for (i=0;i<uiSize;i++) ucCheckCode+=pBuffer[i];  
    return (~ucCheckCode)&0xff;  
}
```