

# H3 项目

以太网模块说明书 V1.0

## 文档履历

| 版本号  | 日期         | 制/修订人 | 内容描述 |
|------|------------|-------|------|
| V1.0 | 2015-01-12 |       | 正式版本 |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |
|      |            |       |      |

confidential

# 目 录

|                      |    |
|----------------------|----|
| 1. 功能介绍 .....        | 4  |
| 1.1. 连接模式 .....      | 4  |
| 1.2. 规格说明 .....      | 4  |
| 2. 框架结构说明 .....      | 5  |
| 3. 接口说明 .....        | 6  |
| 3.1. 类结构介绍 .....     | 6  |
| 3.2. API 函数 .....    | 7  |
| 3.3. 广播消息定义 .....    | 8  |
| 3.4. Demo .....      | 10 |
| 4. 配置说明 .....        | 11 |
| 5. Declaration ..... | 13 |

confidential

# 1. 功能介绍

## 1.1. 连接模式

以太网支持三种连接模式：动态 IP 模式(DHCP)、静态 IP 模式和 PPPoE 模式(注：H8 项目使用的阿里云系统 Settings 不支持 PPPoE 模式，框架层是支持的)，此三种模式互斥，每次连接只能选择其中的一种；

## 1.2. 规格说明

- **连接速度**

芯片内置有 100M 的 EPHY，同时支持外置 PHY，支持 10M/100M/1000M 半双工和全双工速度模式；

- **接口**

支持 MII 和 RGMII 两种接口连接；

- **已支持的 PHY 型号**

| 型号           | 速度    |
|--------------|-------|
| RTL8211D/E   | 1000M |
| IP1001       | 1000M |
| RTL8201CP    | 100M  |
| IP101A/GA/GR | 100M  |
| 内置 EPHY      | 100M  |

- **实测吞吐量**

测试样机：H3 原型机

PHY 型号：内置 EPHY

测试工具：iperf

测试环境：H3 原型机通过网线直接与 PC 相连

测试结果：

| 连接模式         | 吞吐量      |
|--------------|----------|
| 100M 全双工--Tx | 94.2Mbps |
| 100M 全双工--Rx | 92.1Mbps |

注：该项测试的测试结果与 PC 端有关系，在 PC 端的以太网的传输性能优于 H3 盒子，测试结果才能反映 H3 盒子以太网真实速度；



### 3. 接口说明

#### 3.1. 类结构介绍

以太网相关的类主要有：  
android.net.ethernet.EthernetManager  
com.android.server.EthernetService  
android.net.ethernet.EthernetDevInfo  
android.net.EthernetDataTracker  
android.net.PPPoEHandler  
android.net.InterfaceObserver  
其关系如图 2-1 所示：

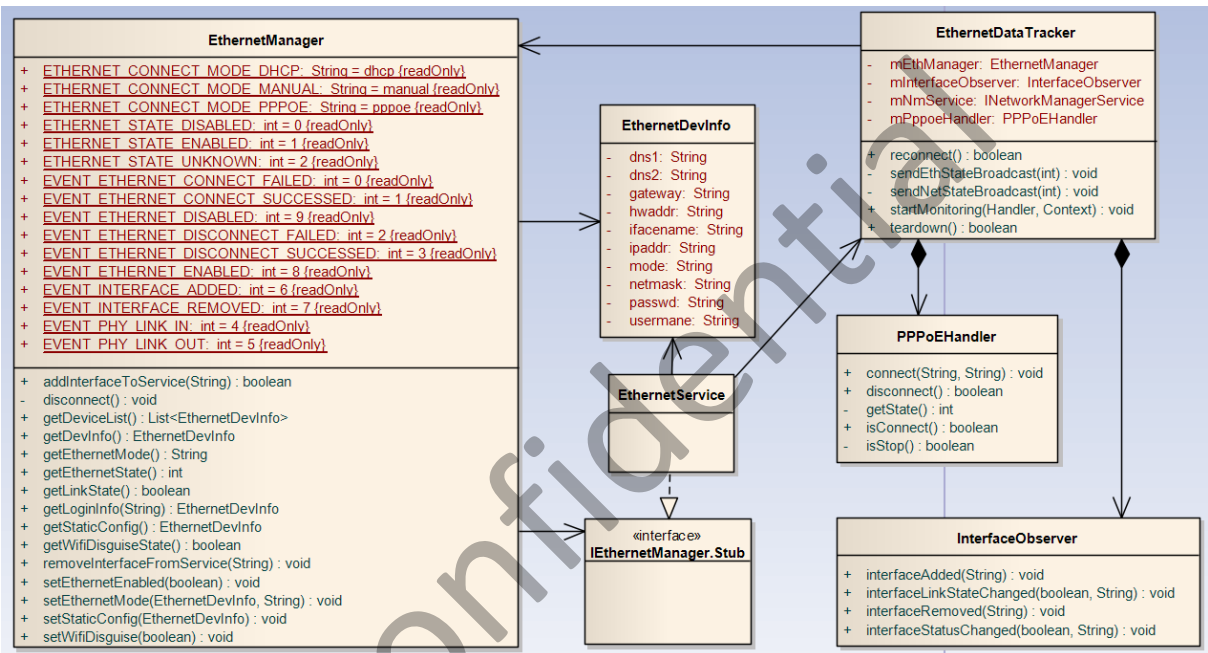


图 2-1

其中，EthernetManager 为应用层提供 API 函数，定义了一些状态和广播常量，具体请参考 3.2 节和 3.3 节；

EthernetDevInfo 包含以太网所有的信息，包括连接信息和设备信息，包括如下内容：

```
public class EthernetDevInfo implements Parcelable

private String ifacename;//网口名称
private String hwaddr;//物理地址
private String ipaddr;//IP 地址
private String gateway;//网关地址
private String netmask;//子网掩码
private String dns1;//DNS1
private String dns2;//DNS2
private String mode;//连接模式
private String passwd;//IPoE 模式或者 PPPoE 模式下的密码
private String username;//IPoE 模式或者 PPPoE 模式下的用户名

... ..
```

EthernetDataTracker 是以太网的核心类，用于以太网的状态管理，PPPoEHandler 专门用于 PPPoE 的状态管理，InterfaceObserver 用于接收来自 kernel 的网口状态变化的广播；

### 3.2. API 函数

(EthernetManager)mContext.getSystemService(Context.ETHERNET\_SERVICE)

作用： 返回一个 EthernetManager 对象；  
 参数： Context.ETHERNET\_SERVICE；  
 返回值： EthernetManager 对象；

public void setEthernetEnabled(boolean enable)

作用： 开启/禁用以太网功能；  
 参数： true： 使能以太网，false： 禁用以太网；  
 返回值： 无；

public int getEthernetState()

作用： 获取当前状态；  
 参数： 无；  
 返回值： ETHERNET\_STATE\_DISABLED： 以太网已禁用；  
 ETHERNET\_STATE\_ENABLED： 以太网已开启；  
 ETHERNET\_STATE\_UNKNOWN： 状态不确定；

public boolean getLinkState()

作用： 获取当前网线连接状态；  
 参数： 无；  
 返回值： true： 网线已连接；false： 网线未连接；

public String getEthernetMode()

作用： 获取当前连接模式；  
 参数： 无；  
 返回值： ETHERNET\_CONNECT\_MODE\_DHCP： 连接模式为 DHCP；  
 ETHERNET\_CONNECT\_MODE\_STATIC： 连接模式为静态 IP 模式；  
 ETHERNET\_CONNECT\_MODE\_PPPOE： 连接模式为 PPPoE 模式；

public void setEthernetMode(String type, EthernetDevInfo info)

作用： 切换连接模式，并尝试使用参数 info 进行连接；  
 参数： type： 连接类型，可能取值为：  
 ETHERNET\_CONNECT\_MODE\_DHCP；  
 ETHERNET\_CONNECT\_MODE\_STATIC；  
 ETHERNET\_CONNECT\_MODE\_PPPOE；

info： 连接需要的配置信息，DHCP 模式时如果使用 IPoE，info 可能包含 IPoE 认证的用户名和密码信息；STATIC 模式时，info 可能包含 ip、netmask、gateway、dns1、dns2；PPPoE 模式时，info 可能包含 PPPoE 认证的用户名和密码信息；

返回值： 无；

```
public EthernetDevInfo getDevInfo()
```

作用： 获取连接信息；

参数： 无；

返回值： 如果为 DHCP 模式时，返回由 DHCP 结果转换而来的 EthernetDevInfo；

如果为 STATIC 模式时，返回结果同 getStaticConfig();

如果为 PPPoE 模式时，返回 PPPoE 协商结果转换而来的 EthernetDevInfo；

```
public EthernetDevInfo getStaticConfig()
```

作用： 获取静态 IP 配置信息；

参数： 无；

返回值： 已保存在 Settings.Global 中的静态 IP 配置，包括 ip、netmask、gateway、dns1、dns2；

```
public EthernetDevInfo getLoginInfo(String mode)
```

作用： 获取 IPoE 获取 PPPoE 模式下的账号密码信息；

参数： ETHERNET\_CONNECT\_MODE\_DHCP：连接模式为 DHCP；

ETHERNET\_CONNECT\_MODE\_PPPOE：连接模式为 PPPoE；

返回值： 已保存在相关文件中的账号和密码信息，以 EthernetDevInfo 的格式返回；

```
public boolean getWifiDisguiseState()
```

作用： 获取 wifi cheat(将以太网连接欺骗成 wifi 连接，以使某些需要检测 wifi 连接的 apk 正常使用)状态；

参数： 无；

返回值： true：wifi cheat 已启用；false：wifi cheat 未启动；

```
public void setWifiDisguise(boolean enable)
```

作用： 开启/禁用 wifi cheat 功能；

参数： true：开启 wifi cheat；false：禁用 wifi cheat；

返回值： 无；

### 3.3. 广播消息定义

以太网除了可以使用上面的 API 函数外，还可以通过接受下面的广播来异步通知相关状态的变化；

以太网已开启：

|        |   |
|--------|---|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION                                   |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.ETHERNET_STATE_ENABLED |

以太网已禁用：

|        |  |
|--------|--|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION                                    |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.ETHERNET_STATE_DISABLED |

插入网线：

|        |   |
|--------|---|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:         |



|  |                                   |
|--|-----------------------------------|
|  | EthernetManager.EVENT_PHY_LINK_IN |
|--|-----------------------------------|

拔出网线:

|        |   |
|--------|---|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION                               |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_PHY_LINK_OUT |

网口加载:

|        |  |
|--------|--|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION                                  |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_INTERFACE_ADDED |

网口删除:

|        |  |
|--------|--|
| Action | EthernetManager.ETHERNET_STATE_CHANGED_ACTION                                    |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_INTERFACE_REMOVED |

连接成功(DHCP、静态 IP 模式):

|        |   |
|--------|---|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION  |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_ETHERNET_CONNECT_SUCCEEDED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

连接成功(PPPoE 模式):

|        |  |
|--------|--|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION   |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_PPPOE_CONNECT_SUCCEEDED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

连接失败(DHCP、静态 IP 模式):

|        |  |
|--------|--|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION   |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_ETHERNET_CONNECT_FAILED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

连接失败(PPPoE 模式):

|        |   |
|--------|---|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION  |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_PPPOE_CONNECT_FAILED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

**断开连接成功(DHCP、静态 IP 模式):**

|        |  |
|--------|--|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION   |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_ETHERNET_DISCONNECT_SUCCEEDED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

**断开连接成功(PPPoE 模式):**

|        |   |
|--------|---|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION  |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_PPPOE_DISCONNECT_SUCCEEDED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

**断开连接失败(DHCP、静态 IP 模式):**

|        |   |
|--------|---|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION  |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_ETHERNET_DISCONNECT_FAILED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

**断开连接失败(PPPoE 模式):**

|        |  |
|--------|--|
| Action | EthernetManager.NETWORK_STATE_CHANGED_ACTION   |
| Extra  | EthernetManager.EXTRA_ETHERNET_STATE:<br>EthernetManager.EVENT_PPPOE_DISCONNECT_FAILED<br>EthernetManager.EXTRA_NETWORK_INFO: mNetworkInfo<br>EthernetManager.EXTRA_LINK_PROPERTIES: mLinkProperties |

**3.4. Demo**

以太网相关的接口使用请参考 TvdsSettings 中的以太网部分，该部分的代码位于 SDK 中：  
android/vendor/tvd/packages/TvdSettings/src/com/android/settings/ethernet/;

## 4. 配置说明

### ● 编译选择

以太网 MAC 控制器驱动在 SDK 中默认已经编译到内核，如果需要去除编译或者更改编译方式，请按如下方式进行：

```
去到 linux-3.4 目录下，输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Ethernet driver support --->
            [*] SUNXI Platform Network devices drivers
                <*> Sunxi platform Gigabit Ethernet driver
```

更改“Sunxi platform Gigabit Ethernet driver”此项，即可以更改编译选择；

### ● PHY 模式选择

PHY 模式选择定义在 sys\_config.fix 中，如下所示：

```
[gmac0]
gmac_used      = 2
gmac_mode      = "MII"
... ..
```

gmac\_used 即表示 PHY 模式，“gmac\_used = 0”表示不使用 gmac，“gmac\_used = 1”表示使用外部 PHY，“gmac\_used = 2”表示使用内置 EPHY。

### ● 接口类型配置

PHY 的接口类型定义在 sys\_config.fix 中，如下所示：

```
[gmac0]
gmac_used      = 2
gmac_mode      = "MII"
... ..
```

gmac\_mode 即表示选择的接口类型，如果选择 MII 接口，则设置为“MII”，如果选择 RGMII 接口，则设置为“RGMII”，不配置情况下，默认选择“RGMII”。

### ● 电源配置

H3 方案没有使用 AXP 进行电源控制，顾如果使用外部 PHY，则在硬件上做好电源设计，如果使用内置 EPHY，则无需供电。

### ● 默认 MAC 地址设置

在未烧录 MAC 地址之前，驱动从 Chip ID 生成一个有效的 MAC 地址，如果需要更改默认 MAC 地址，修改 env.cfg 中的“mac=”这一项即可，如下所示：

```
nand_root=/dev/block/system
mmc_root=/dev/block/mmcblk0p7
init=/init
loglevel=8
mac=98:3b:16:ff:0d:ac
```

**注意：**mac 地址中的字母需小写

Confidential

## 5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

Confidential