

EC2X&AG35-QuecOpen SGMII API 说明

LTE 系列

版本: EC2X&AG35-QuecOpen_SGMII_API_说明_V1.0

日期: 2018-04-07

状态: 临时文件



上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2018，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2018.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2018-04-07	Mike	初始版本

目录

文档历史	2
目录	3
1 引言	4
2 SGMII 介绍	5
3 API 接口说明	6
3.1 ql_sgmii_enable	6
3.2 ql_sgmii_disable	7
3.3 ql_sgmii_speed_set	7
3.4 ql_sgmii_speed_get	8
3.5 ql_sgmii_duplex_set	8
3.6 ql_sgmii_duplex_get	9
3.7 ql_sgmii_autoneg_set	9
3.8 ql_sgmii_autoneg_get	10
3.9 ql_sgmii_info_get	10
3.10 QL_Ethernet_Mode_Set	10
3.11 QL_Ethernet_Mode_Get	11
4 实例	12

1 引言

此文档是关于 SGMII 功能介绍和 API 的使用说明。

2 SGMII 介绍

SGMII 是 PHY 与 MAC 之间的接口，类似与 GMII 和 RGMII，只不过 GMII 和 RGMII 都是并行的，而且需要随路时钟，PCB 布线相对麻烦，而且不适应背板应用。而 SGMII 是串行的，不需要提供另外的时钟，MAC 和 PHY 都需要 CDR 去恢复时钟。另外 SGMII 是有 8B/10b 编码的，速率是 1.25G。

EC20 和 AG35 目前仅支持 AR8033 PHY 芯片，10BASE-T_e/100BASE-T_e/1000BASE-T_e IEEE802.3 compliant。

3 API 接口说明

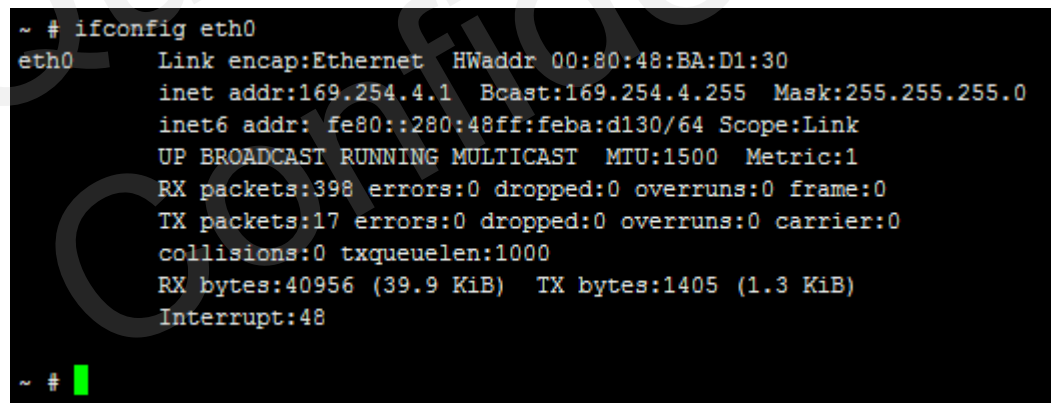
Linux 默认开启不会启动 SGMII 功能，提供接口如下：

3.1 ql_sgmmi_enable

```
/**
 * Enables the SGMII ethernet module.
 *
 * @param None
 *
 * @return
 *   On success, 0 is returned. On error, -1 is returned.
 */
```

extern int ql_sgmmi_enable(void);

接口说明：启动 SGMII 功能，调用该函数会加载 SGMII 驱动；驱动加载成功后，在控制台可以看到 eth0 网口，如图 1：

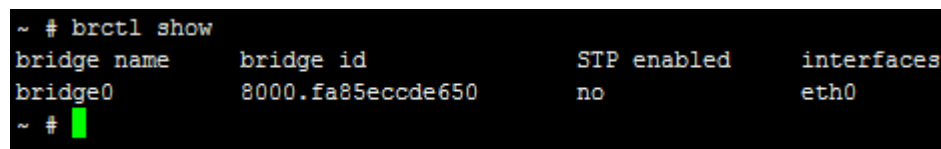


```
~ # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:80:48:BA:D1:30
          inet addr:169.254.4.1  Bcast:169.254.4.255  Mask:255.255.255.0
          inet6 addr: fe80::280:48ff:feba:d130/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:398 errors:0 dropped:0 overruns:0 frame:0
          TX packets:17 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40956 (39.9 KiB)  TX bytes:1405 (1.3 KiB)
          Interrupt:48

~ #
```

图 1

该网口启动成功后，qti 程序会捕获到启动事件，并通知 QCMAP_ConnectionManager 进程把 eth0 设备加载到 bridge0 桥上，如图 2：



```
~ # brctl show
bridge name      bridge id                STP enabled    interfaces
bridge0          8000.fa85eccde650        no             eth0

~ #
```

图 2

注意事项：

eth0 接口默认配置的 MAC 地址为 00:80:48:BA:D1:30，如果客户需要修改这个 MAC 地址，可以通过命令行设置(例如：ifconfig eth0 hw ether 00:80:48:BA:d1:30)，也可以修改 SGMII 启

动脚本(/etc/init.d/start_emac_le)里面的地址设置。

3.2 ql_sgmii_disable

```
/**
 * Disable the SGMII ethernet module.
 *
 * @param None
 *
 * @return
 *   On success, 0 is returned. On error, -1 is returned.
 */
```

extern int ql_sgmii_disable(void);

接口说明：关闭 SGMII 功能，调用该函数，会把 eth0 网口从 bridge0 桥移走，并把启动加载的驱动都卸载掉。

3.3 ql_sgmii_speed_set

```
/**
 * Set the SGMII ethernet configuration: the ethernet auto negotiation configuration.
 *
 * @param [in] auto negotiation
 *
 * @return
 *   On success, 0 is returned. On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
```

extern int ql_sgmii_autoneg_set(ql_sgmii_autoneg_e autoneg);

接口说明：设置网口速率，支持设置 10MHZ/100MHZ/1000MHZ 三种速率集，默认为自适应，其宏定义如下：

```
typedef enum {
    QL_SGMII_SPEED_10MHZ,      /* 10MHZ */
    QL_SGMII_SPEED_100MHZ,    /* 100MHZ */
    QL_SGMII_SPEED_1000MHZ    /* 1000MHZ */
} ql_sgmii_speed_e;
```


3.4 ql_sgmii_speed_get

```
/**
 * Get the SGMII ethernet configuration: the ethernet speed configuration.
 *
 * @param [out]          the ethernet current speed
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
extern int ql_sgmii_speed_get(ql_sgmii_speed_e *speed);
```

接口说明：获取网口当前设置的速率。

3.5 ql_sgmii_duplex_set

```
/**
 * Set the SGMII ethernet configuration: the ethernet duplex configuration.
 *
 * @param [in]          ethernet duplex
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
extern int ql_sgmii_duplex_set(ql_sgmii_duplex_e duplex);
```

接口说明：设置网口的双工模式，支持设置半双工和全双工，其宏定义如下：

```
typedef enum {
    QL_SGMII_DUPLEX_FULL = 0,          /* 全双工 */
    QL_SGMII_DUPLEX_HALF          /* 半双工 */
} ql_sgmii_duplex_e;
```

注意：调用该函数，不能设置网口速率为 **QL_SGMII_SPEED_AUTO**，对于千兆速率 **AR8033** 芯片仅支持全双工。

3.6 ql_sgmii_duplex_get

```
/**
 * Get the SGMII ethernet configuration: the ethernet duplex configuration.
 *
 * @param [out]          the ethernet current duplex
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
extern int ql_sgmii_duplex_get(ql_sgmii_duplex_e *duplex);
```

接口说明：获取网口设置的双工模式

3.7 ql_sgmii_autoneg_set

```
/**
 * Set the SGMII ethernet configuration: the ethernet auto negotiation configuration.
 *
 * @param [in]          auto negotiation
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
extern int ql_sgmii_autoneg_set(ql_sgmii_autoneg_e autoneg);
```

接口说明：设置以太网卡是否是自适应模式，其宏定义如下：

```
typedef enum {
    QL_SGMII_AUTONEG_OFF = 0,
    QL_SGMII_AUTONEG_ON
} ql_sgmii_autoneg_e;
```

3.8 ql_sgmii_autoneg_get

```
/**
 * Get the SGMII ethernet configuration: the ethernet auto negotiation configuration.
 *
 * @param [out]      auto negotiation
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
```

```
extern int ql_sgmii_autoneg_get(ql_sgmii_autoneg_e *autoneg);
```

接口说明：获取以太网卡是否是自适应模式

3.9 ql_smgii_info_get

```
/**
 * Get the smgii ethernet information: transmit and receive bytes, transmit and receive packets,
 * speed, duplex.
 *
 * @param [out]      the ethernet information
 *
 * @return
 *   On success, 0 is returned.  On error, -1 is returned.
 *
 * @dependencies
 *   the ql_sgmii_enable must be set enable
 */
```

```
extern int ql_smgii_info_get(struct ql_sgmii_info *info);
```

接口说明：获取网口当前状态，包括：接收发送数据包个数以及数据大小、当前运行速率、当前运行双工模式。

注意：3~9 接口调用之前，需要调用 ql_sgmii_enable();

3.10 QL_Ethernet_Mode_Set

```
/**
```

```

* Sets Ethernet Tethering Mode.
*
* @param [in]   ethernet tethering mode
*
* @return
*   On success, 0 is returned.  On error, -1 is returned.
*
*/

```

```
extern int QL_Ethernet_Mode_Set(ql_ethernet_mode_e mode);
```

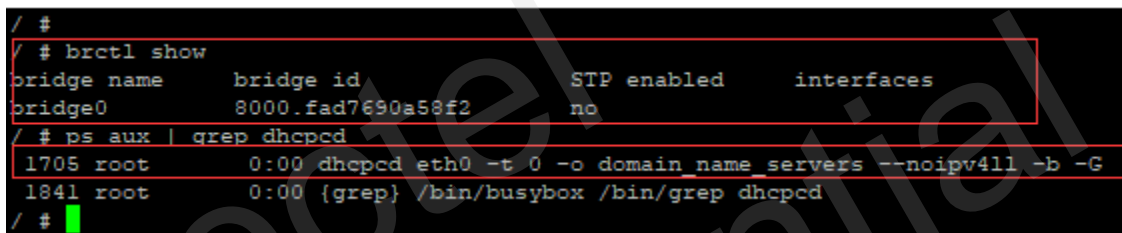
接口说明：设置以太网的工作模式，默认为 LAN Route 模式

以太网工作模式有两种工作模式：WAN Route 模式和 LAN Route 模式

1) LAN Route 模式：以太网卡作为局域网模式为对端设备（如：PC）提供网络服务。

2) WAN Route 模式：一台网卡作为上网口，通过对端的设备（如：交换机）为模块提供网络服务。

该模式下 eth0 端口不会加入到 bridge0 桥接下面，并且通过 udhcpd 来获取 IP 地址。如图 4：



```

/ #
/ # brctl show
bridge name      bridge id        STP enabled      interfaces
bridge0          8000.fad7690a58f2 no
/ # ps aux | grep dhcpcd
1705 root        0:00 dhcpcd eth0 -t 0 -o domain_name_servers --noipv4ll -b -G
1841 root        0:00 {grep} /bin/busybox /bin/grep dhcpcd
/ #

```

图 4

注意：该接口仅支持 AG35 系列的模块

3.11 QL_Ethernet_Mode_Get

```

/**
* Gets Ethernet Tethering Mode.
*
* @param [out]   ethernet tethering mode
*
* @return
*   On success, 0 is returned.  On error, -1 is returned.
*
*/

```

```
extern int QL_Ethernet_Mode_Get(ql_ethernet_mode_e *mode);
```

接口说明：获取当前以太网的工作模式

4 实例

可以参考 example/sgmii/example_sgmii.c

```
int main(int argc, char **argv)
{
    ql_sgmii_enable();
    ql_sgmii_speed_duplex_set(QL_SGMII_SPEED_100MHZ, QL_SGMII_DUPLEX_FULL);
    return 0;
}
```

说明：当前未提供配置保存，后续会提供配置保存。