# EC2x&AG35-Quecopen SPI FLASH User Guide

**LTE Standard/Automotive Module Series**

Rev. EC2x&AG35-Quecopen_SPIFLASH_User_Guide_V1.0

Date: 2018-10-22

Status: Preliminary

**Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:**

**Quectel Wireless Solutions Co., Ltd.**

7<sup>th</sup> Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

**Or our local office. For more information, please visit:**

http://www.quectel.com/support/sales.htm

**For technical support, or to report documentation errors, please visit:**

http://www.quectel.com/support/technical.htm

Or email to: support@quectel.com

**GENERAL NOTES**

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

# About the Document

## History

| Revision | Date | Author | Description |
|----------|------|--------|-------------|
| 1.0 | 2018-10-22 | Matthew MA | Initial |

# Contents

# 1 Introduction

This document mainly provides adding methods and instructions for customers who need to add Spi Flash in Open Linux platform. The purpose is to support the flash of the spi interface in the Linux system, and to run stably and use the file system based on the flash. This document will, take a 512M size Toshiba flash which model No.is TC58CVG2S0HxS0HxAIx, as an example, the use of file system is based on ubifs.

This document mainly applies to overseas market. The Automotive module currently supporting this includes:

● EC2x: EC20 R2.1/EC25/EC21
● AG35

# 2 Method to Add SPI Flash Driver

## 2.1. Add SPI Interface

Configure macro definition of *CONFIG_MTD_SERIAL_NAND* to *y* in *arch/armconfigs/mdm9607-perf_defconfig* and *arch/arm/configs/mdm9607_defconfig*. This option will open support function of SPI Nand.

## 2.2. Add TSB Serial-Nand Driver

Add Toshiba spi flash driver under the premise of opening spi nand, this driver is provided by Toshiba (TC58CVG2S0HxS0HxAlx) manufacturer, and then integrated it into the Linux system to make Linux support this spi flash. Drivers of different spi flash models are different, but adding process keeps the same, perform the same adaptation according to different models.

### 2.2.1. Device Tree Configuration

Add the related attributes of the spi flash sub-node to the device tree. Different types of spi flash have different attributes. User can modify the corresponding attributes according to the chip manual, as shown in the following figure.
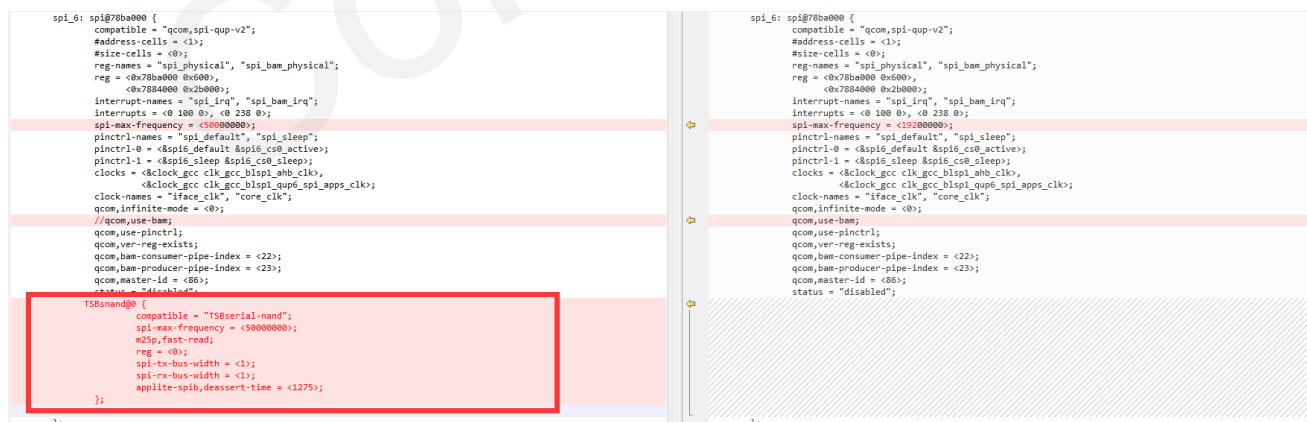
## 2.2.2. Add Spi Flash Driver

Add compile option to *drivers/mtd/devices/Makefile* and configure *CONFIG_MTD_SERIAL_NAND* to *y* or m according to user needs, as shown in the following figure:





serial-nand.c     serial-nand.h     serial-nand_bbt.c
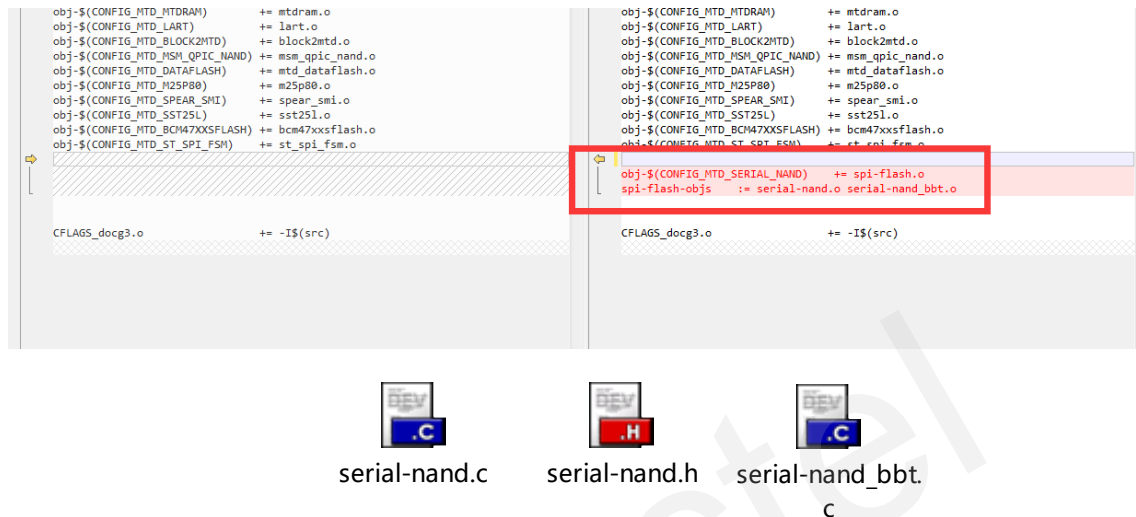
And put three driver files provided under the directory of *drivers/mtd/devices/(mv serial-nand\* drivers/mtd/devices/)*.

# 3 Support Based on UBIFS File System

It is recommended to use the file system of ubifs, because the factory default flash chip is empty flash (that is, all the content is 0xff data). Therefore, when performing the process of leaving the factory, the film must first perform the ubi data format operation, and then support the mount and use of ubifs (first format, then mount is standard operation), the operation only needs execute once when leaving the factory. To avoid data loss, if the file system data format of ubifs is not wrong, please do not perform ubiformat operation (Note: This command will cause all flash data content to lost).

The following example simply shows mount process:

## 3.1. Ubi Data Format

Method one: *ubiformat -y /dev/mtd25*

Note: 25 is the mtd number, which can be modified according to the actual situation.*cat /proc/mtd* can be viewed. Executing this command when leaving the factory or the ubi file system is damaged. It will cause data loss and please be used with caution.

Method two: By downloading an empty ubifs file system ubi to the internal of the flash.

Note: Executing the command *mtding –f "mtd name" "data.ubi"*. This method takes much less time than the first method. Different sizes of flash time consumption reduction values are not the same, but this method needs to make ubi in advance.

## 3.2. ubattach -p /dev/mtd25

## 3.3. ubimkvol /dev/ubi${nub} -m -N ${device name}

Note: Create a volume name, nub number value of ubi can be modified according to the actual situation, and devicename is the volume name. If the volume already exists, users do not need to execute this command. Generally, it will only be executed after ubi format is performed.

## 3.4. mount -t ubifs /dev/ubi${nub}_0 ${dir}

Note: Mount flash, nub of ubi can be modified according to the actual situation, dir is the directory that mounts