Software Application Guide
ECB10-135A5M5M-I Single Board Computer



Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of contents

Disclaimer and Copyright Notice 2
1. Overview 3
1.1. Software Resources 3
2. Preparation before use 3
2.1. Serial port software installation 3
2.2. Setting the DIP switch 4
3. Quick Start 4
3.1. Mirror Configuration 4
3.2. Power on and start 5
4. Functional testing 6
4.1. Core Resources 6
4.2. Peripheral Interface 12
4.3. Network Interface 23
5. References 28
6. Revision Notes 28
7. About Us 29

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Disclaimer and Copyright Notice

The information in this document is subject to change without notice. The document is provided "as is" without any warranty, including any warranty of merchantability, fitness for a particular purpose or non-infringement, and any warranty otherwise provided by any proposal, specification or sample. No liability is assumed for this document, including liability for infringement of any patent rights arising from the use of the information in this document. No license to any intellectual property, whether express or implied, by estoppel or otherwise, is granted by this document.

The test data obtained in this article are all obtained by Ebyte Laboratory testing, and the actual results may be slightly different.

All trade names, trademarks and registered trademarks mentioned herein are the property of their respective owners and are hereby acknowledged.

The final right of interpretation belongs to Chengdu Ebyte Electronic Technology Co., Ltd.

Notice:

Due to product version upgrades or other reasons, the contents of this manual may be changed. Ebyte Electronic Technology Co., Ltd. reserves the right to modify the contents of this manual without any notice or reminder. This manual is only used as a guide. Chengdu Ebyte Electronic Technology Co., Ltd. tries its best to provide accurate information in this manual, but Chengdu Ebyte Electronic Technology Co., Ltd. does not ensure that the contents of the manual are completely error-free, and all statements, information and suggestions in this manual do not constitute any express or implied warranty.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. Overview

This article mainly introduces the complete process of customizing a complete embedded Linux system based on the Ebyte core board, including the preparation of the development environment, code acquisition, and how to transplant Bootloader and Kernel, customize Rootfs suitable for your own application needs, etc. We first introduce how to build a system image suitable for the ECB10-TB13X development board based on the source code we provide, and how to burn the built image to the development board. For those users who develop projects based on the ECK10-135A5M1G-I core board, we focus on the methods and key points of transplanting this system to the user's hardware platform, and through some actual transplantation cases and Rootfs customization cases, users can quickly customize the system image suitable for their own hardware.

1.1. Software Resources

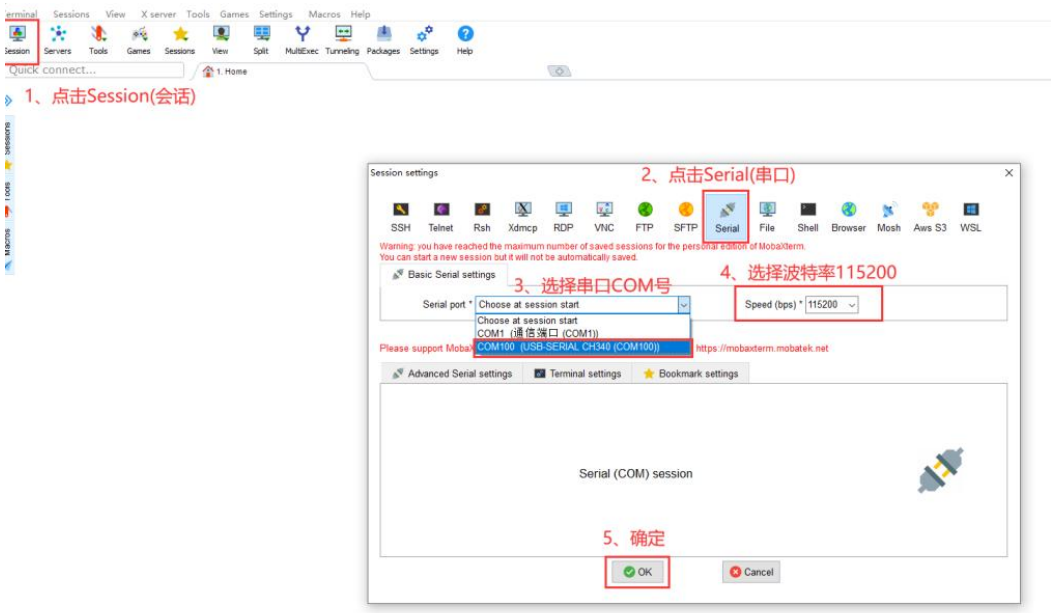
ECB10-TB13X is equipped with an operating system based on the Linux 6.1.28 kernel. The development board comes with a cross-compilation tool chain required for embedded Linux system development, TF-A source code, Optee-os source code, U-boot source code, Linux kernel and driver module source code, as well as various development and debugging tools suitable for Windows desktop environment and Linux desktop environment, application development samples, etc.

2. Preparation before use

2.1. Serial port software installation

Before using the serial port, you need to install the CH340 serial port driver and Moba Xterm serial port terminal. For specific installation methods , see the core board development guide.

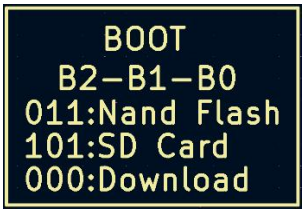
After connecting the serial port, install Moba X term and configure it, and open the serial port as follows.



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2. Setting the DIP switch

The single board computer supports three boot modes, namely Nand Flash , SD card and USB (Download) boot mode.



the SBC is set to boot from N and mode, the serial terminal prints the running information of TF-A, OP-TEE, U-Boot and kernel, as shown below. Note that the SBC has no image by default, refer to Chapter 3 Quick Start to burn the image.

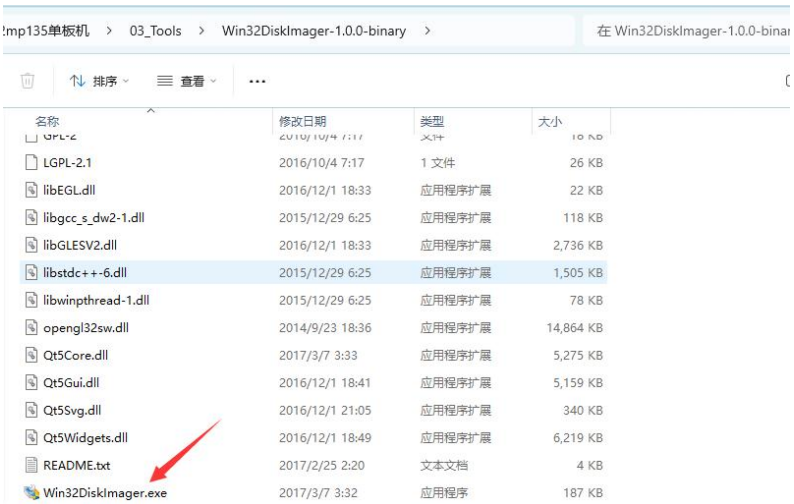
3. Quick Start

3.1. Mirror Configuration

The factory development board does not have a burn-in image. Please refer to Chapter 2 to configure the dial switch to boot from the SD card and install the serial terminal software. ECB10 can boot from the SD card or the onboard N and Flash . For more details, please refer to the Software_Development_Guide .

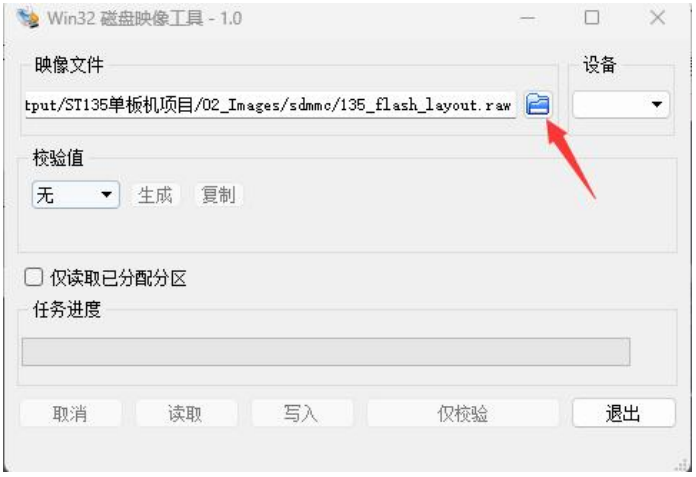
Here is how to make an SD card quick start board:

Use a third-party image burning tool to write the img image file 135_flash_layout.raw in 02_Images/ sd mmc to the SD card. 03_Tools/ already contains a burning tool: Win32DiskImager-1.0.0-binary . Double-click to open it.



Select the image file 135_flash_layout.raw

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



The file type must be selected as * .* to find the burning file



After finding the device with the card reader, click Write and wait for the burning to complete.

3.2. Power on and start

Insert the SD card , find the DEBUG silkscreen on the board , and after connecting correctly, use the USB -TTL serial port to connect to the computer. If there is no serial terminal software, see Chapter 2.

After the single-board computer starts, the serial terminal prints the running information of U-Boot and the kernel.

```
NOTICE:  CPU: STM32MP135D Rev.Y
NOTICE:  Model: EBYTE TECH STM32MP135 Discovery Board
INFO:    PMIC version = 0x21
INFO:    Reset reason (0x35):
INFO:      Power-on Reset (rst_por)
INFO:    FCONF: Reading TB_FW firmware configuration file from: 0x2ffe0000
INFO:    FCONF: Reading firmware configuration information for: stm32mp_io
INFO:    Using FMC NAND
INFO:      Instance 1
INFO:    Boot used partition fsbl1
NOTICE:  BL2: v2.8-stm32mp1-r1.0(debug):()
NOTICE:  BL2: Built : 14:13:43, Jul  5 2024
INFO:    BL2: Doing platform setup
INFO:    RAM: DDR3-1066 bin F 1x4Gb 533MHz v1.53
INFO:    Memory size = 0x20000000 (512 MB)
INFO:    BL2: Loading image id 1
INFO:    Loading image id=1 at address 0x30006000
INFO:    Image id=1 loaded: 0x30006000 - 0x30006236
INFO:    FCONF: Reading FW_CONFIG firmware configuration file from: 0x30006000
INFO:    FCONF: Reading firmware configuration information for: mce_config
INFO:    FCONF: Reading firmware configuration information for: dyn_cfg
INFO:    FCONF: Reading firmware configuration information for: stm32mp1_firewall
INFO:    BL2: Loading image id 4
INFO:    Loading image id=4 at address 0xde000000
INFO:    Image id=4 loaded: 0xde000000 - 0xde00001c
INFO:    OPTEE ep=0xde000000
INFO:    OPTEE header info:
INFO:          magic=0x4554504f
INFO:          version=0x2
INFO:          arch=0x0
INFO:          flags=0x0
INFO:          nb_images=0x1
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4. Functional testing

4.1. Core Resources

In Linux system, the proc virtual file system is provided to query the parameters of various core resources and some general tools are provided to evaluate the performance of resources. The following will specifically read and test the parameters of core resources such as CPU, memory, eMMC, RTC, etc.

4.1.1. CPU

ECB10-135A5MA-I is STM32MP135DAF7, based on high-performance single-core Arm® Cortex®-A7 32-bit RISC core, with an operating frequency of 650 M Hz. The CPU core of the Cortex-A7 processor includes a 32 kbyte L1 instruction cache, a 32 kbyte L1 data cache, and a 128 kbyte L2 cache. The Cortex-A7 processor is a very energy-efficient application processor designed to provide rich performance for high-end wearable devices and other low-power embedded and consumer applications. It provides 20% more single-threaded performance than the Cortex-A5 and provides similar performance to the Cortex-A9.

4.1.1.1. View CPU information

Run cat /proc/cpuinfo to view CPU information .

```
root@ebyte-ubuntu:~# cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 48.00
Features        : half thumb fastmult vfp edsp thumbee neon vfpv3 tls vfpv4 idiva idivt
vfpd32 lpae evtstrm
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

Hardware        : STM32 (Device Tree Support)
Revision        : 0000
Serial          : 000A0010343251163537334
```

processor: The number of the logical processing core in the system. For a multi-core processor, it can be a physical core or a virtualized logical core using hyperthreading technology.

model name: the name and number of the CPU

BogoMIPS: A rough estimate of the number of instructions per second that the CPU runs when the system kernel is booted (MillionInstructions Per Second)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.1.1.2. Check CPU usage

```
top - 13:56:36 up 58 min,  0 users,  load average: 0.00, 0.00, 0.11
Tasks:  64 total,   1 running,  33 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  1.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   449616 total,   370472 free,    30388 used,    48756 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   407640 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  255 root      20   0    4908   2284   1868 R  0.7  0.5   0:00.16 top
    1 root      20   0   28520   5696   4472 S  0.0  1.3   0:03.30 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 slub_flushwq
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 netns
    8 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0+
    9 root      20   0       0      0      0 I  0.0  0.0   0:03.06 kworker/u2:+
   10 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
   11 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_tasks_k+
   12 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_tasks_t+
   13 root      20   0       0      0      0 S  0.0  0.0   0:00.25 ksoftirqd/0
   14 root      20   0       0      0      0 I  0.0  0.0   0:00.34 rcu_preempt
   15 root      20   0       0      0      0 S  0.0  0.0   0:00.02 kdevtmpfs
   16 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 inet_frag_wq
   20 root      20   0       0      0      0 I  0.0  0.0   0:00.77 kworker/0:4+
```

%usr : indicates the CPU usage of user space programs

%sys : indicates the CPU usage of the system space

%idle : idle cpu

%irq : The number of hard interrupts handled by the CPU

%sirq : The number of soft interrupts processed by the CPU

4.1.1.3. View CPU temperature information

Use cat /sys/class/hwmon/hwmon0/temp1_input to check the internal temperature of the CPU .

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:~# cat /sys/class/hwmon/hwmon0/temp1_input
54988
```

The above value divided by 1000 is the correct temperature value in degrees Celsius.

4.1.2. Memory

4.1.2.1. View memory information

You can view memory information through cat /proc/meminfo .

```
root@ebyte-ubuntu:~# cat /proc/meminfo
MemTotal:         449616 kB
MemFree:          370472 kB
MemAvailable:     407640 kB
Buffers:               0 kB
Cached:           39208 kB
SwapCached:            0 kB
Active:           13408 kB
```

MemTotal: Total memory. The value shown here is 449616 kB, which means that the system has a total of about 449 MB of physical memory available.

MemFree: The amount of free memory. The value displayed here is 370472 kB, which means that there are about 370 MB of free memory in the current system and it is not being used.

MemAvailable: The amount of available memory. The value shown here is 407640 kB, which represents the total amount of memory currently available to the system, including cached memory and available memory.

Buffers: Buffer usage. The value shown here is 0 kB, which means that the system is not currently using any memory as a buffer.

Cached: The amount of cached memory. The value shown here is 39208 kB, which indicates the amount of memory currently used by the system for caching.

SwapCached: The amount of memory that is swap cached. The value shown here is 0 kB, which means that there is currently no memory cached in the swap space.

Active: The amount of active memory. The value shown here is 13408 kB, which indicates the amount of memory currently in use.

4.1.2.2. Memory stress test

By specifying the size and number of test memory, the existing memory of the system can be stress tested. You can use the system tool memtester to test, such as specifying a memory size of 10MB and a test number of 10, and the test command is "memtester 10M 10".

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:~# memtester 10M 10
memtester version 4.3.0 (32-bit)
Copyright (C) 2001-2012 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffff000
want 10MB (10485760 bytes)
got  10MB (10485760 bytes), trying mlock ...locked.
Loop 1/10:
  Stuck Address       : ok
  Random Value        : ok
  Compare XOR         : ok
  Compare SUB         : ok
  Compare MUL         : ok
  Compare DIV         : ok
  Compare OR          : ok
  Compare AND         : ok
  Sequential Increment: ok
  Solid Bits          : ok
  Block Sequential    : ok
  Checkerboard        : ok
  Bit Spread          : ok
  Bit Flip            : ok
  Walking Ones        : ok
  Walking Zeroes      : ok
  8-bit Writes        : ok
16-bit Writes         : ok
*****
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.1.3. Nand Flash Test

4.1.3.1. Check Nand capacity and size

```
root@ebyte-ubuntu:~# cat /proc/mtd
dev:        size     erasesize    name
mtd0: 00080000 00020000 "fsbl1"
mtd1: 00080000 00020000 "fsbl2"
mtd2: 00080000 00020000 "metadata1"
mtd3: 00080000 00020000 "metadata2"
mtd4: 00400000 00020000 "fip-a1"
mtd5: 00400000 00020000 "fip-a2"
mtd6: 00400000 00020000 "fip-b1"
mtd7: 00400000 00020000 "fip-b2"
mtd8: 0ee00000 00020000 "UBI"
```

4.1.4. R TC

RTC (Real-time clock) is a clock that records real time. When the software system is shut down, the system time is retained and the timing continues. When the system is restarted, the time is synchronized into the software system.

The STM32MP135 chip contains an RTC clock. If the actual product does not have very high requirements for RTC power consumption and the power-off time is required to be maintained within one month, the internal RTC of the chip can be used directly. Otherwise, a dedicated external RTC chip is required. The RTC test is usually carried out with the hwclock and date commands commonly used in Linux systems. The following test writes the system time to the RTC, reads the RTC time and sets it as the system time, and tests the time power-off retention.

Setting the system time

Set the system time to 2 024.07.12 16:35:15

```
root@ebyte-ubuntu:~# date 071216352024.15
Fri Jul 12 16:35:15 UTC 2024
```

Write system time to R TC

```
root@ebyte-ubuntu:~# hwclock -w
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:~# hwclock -r
2000-01-01 07:07:15.965766+0000
```

Keep RTC time when power off

Turn off the development board and disconnect the power supply, then power it back on after a while. Check the RTC time and system time:

```
root@ebyte-ubuntu:~# hwclock -r
2024-07-12 16:52:42.648561+0000
```

4.1.5. Watch Dog

Use the test routine to test the watchdog. The file has been compiled and placed in the home directory. Run the watchdog application, the timeout is 4s, and the watchdog is fed every 1s:

```
root@ebyte-ubuntu:/home# ./watchdog 4 1 0
Starting wdt_driver (timeout: 4, sleep: 1, test: ioctl)
Trying to set timeout value=4 seconds
The actual timeout was set to 4 seconds
Now reading back -- The timeout is 4 seconds
```

If the above 1s is changed to greater than 4s, the required 4s dog feeding time will be exceeded and the development board will restart.

4.1.6. Power Management

This chapter demonstrates the Suspend function of Linux power management, which allows the development board to sleep and be awakened by external events. The Linux kernel generally provides three types of Suspend: Freeze, Standby and STR (Suspend to RAM). They can be triggered by writing "freeze", "standby" and "mem" to the "/sys/power/state" file in user space. Currently, only the "mem" method is supported.

View currently supported modes

```
root@ebyte-ubuntu:~# cat /sys/power/state
mem
```

Set wakeup source

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:/home#echoenabled >/sys/devices/platform/soc/40010000.serial/tty/ttySTM0/power/wakeup
```

Hibernate to RAM

```
root@ebyte-ubuntu:~# echo "mem" > /sys/power/state
```

4.2. Peripheral Interface

4.2.1. GPIO

The GPIO test is implemented through the file system sysfs interface. The pin number is defined as #define PIN_NO(port, line) (((port) - 'A') * 0x10 + (line)), where port is the gpio port, line is the gpio corresponding pin, and ((port)-'A') represents ASCII code subtraction. For example, the pin number corresponding to P A8 is: PIN_NO(' A ', 8)=(0x4 1 -0x41)*0x10+ 8 =(65 -65)*16+ 8 = 8 .

Export G PIO

```
root@ebyte-ubuntu:~# cd /sys/class/gpio/
root@ebyte-ubuntu:/sys/class/gpio# ls
export       gpiochip112  gpiochip16  gpiochip48  gpiochip80  unexport
gpiochip0    gpiochip128  gpiochip32  gpiochip64  gpiochip96
root@ebyte-ubuntu:/sys/class/gpio# echo 8 > export
```

Set G PIO direction

enter

```
root@ebyte-ubuntu:/sys/class/gpio# echo "in" > PA8/direction
```

Output

```
root@ebyte-ubuntu:/sys/class/gpio# echo "out" > PA8/direction
```

Set G PIO value

```
root@ebyte-ubuntu:/sys/class/gpio# echo 1 > PA8/value
```

Unregister G PIO

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:/sys/class/gpio# echo 8 > unexport
root@ebyte-ubuntu:/sys/class/gpio# ls
export      gpiochip112  gpiochip16  gpiochip48  gpiochip80  unexport
gpiochip0   gpiochip128  gpiochip32  gpiochip64  gpiochip96
```

4.2.2. LED Light

The Linux system provides an independent subsystem to facilitate the operation of LED devices from user space. This subsystem provides operation interfaces for LED devices in the form of files. These interfaces are located in the /sys/class/leds directory. In the hardware resource list, we have listed all the LEDs on the development board. The following commands are used to read and write sysfs to test the LED. The following commands are all general commands and are also general methods for operating LEDs.

```
root@ebyte-ubuntu:/sys/class/leds# ls
green:heartbeat
```

LEDs on and off

```
root@ebyte-ubuntu:/sys/class/leds/green:heartbeat# echo 0 > brightness
root@ebyte-ubuntu:/sys/class/leds/green:heartbeat# echo 1 > brightness
```

View LED Trigger Mode

```
root@ebyte-ubuntu:/sys/class/leds/green:heartbeat# cat trigger
[none] rfkill-any rfkill-none kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock
kbd-shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftllock kbd-shiftrlock kbd-ctrllllock
kbd-ctrlrlock timer oneshot heartbeat backlight gpio cpu cpu0 default-on transient flash torch
mmc0 mmc1
```

4.2.3. Serial Port

This single board computer has two serial ports for use, namely UART5 and USART3 . UART5 is a two-wire system with only RX and TX , and USART3 is a four-wire system with RX , TX, RTS and CTSS .

4.2.3.1. U ART5

This serial port can be directly connected to the PC using a USB - TTL adapter cable for testing. It is registered as tty STM2 in the single board computer , and communication can be performed by opening tty STM2 using minicom .

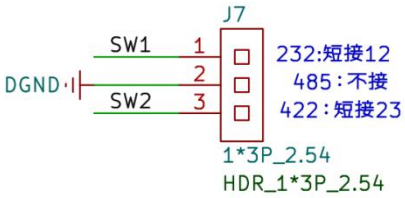
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



4.2.3.2. U SART3

This serial port supports three protocols, RS485 , RS422 and RS232 , which can be selected by jumper cap. It is registered as tty STM1 in the single board computer .

	SW1	SW2
485	NO	NO
422	NO	NC
232	NC	NO



4.2.3.2.1. RS232

Use a jumper to short pins 1 and 2 of J7 , use RS232 to USB to connect to the computer, and use minicom to communicate with the single-board computer.

Supports two communication modes: enabling hardware flow control and disabling it.

4.2.3.2.2. RS422

Use a jumper to short-circuit pin 23 of J7 , use RS422 to USB to connect to the computer, and use minicom to communicate with the single-board computer.

4.2.3.2.3. RS485

You can use the termios library to operate the RTS control line to operate the send enable and receive enable. The sample code is as follows.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
****
#include <termios.h>
int main() {
    int serial_fd = open("/dev/ttySTM1", O_RDWR);
    if (serial_fd == -1) {
        perror("Failed to open serial port");
        return 1;
    }
    int status;
    if (ioctl(serial_fd, TIOCMGET, &status) == -1) {
        perror("Failed to get modem status");
        close(serial_fd);
        return 1;
    }
// Setting the RTS signal

    status |= TIOCM_RTS;    // Set RTS to high (enable sending)//
status &= ~TIOCM_RTS; // Set RTS to low level

    if (ioctl(serial_fd, TIOCMSET, &status) == -1) {
        perror("Failed to set modem status");
        close(serial_fd);
        return 1;
}
    // Other operations or data transfer can be performed here
    while(1);
close(serial_fd);
    return 0;
}
```

using RS485 to USB to connect to the PC , continue to use the Linux system calls write and read to read and write data.

4.2.4. CAN

This section uses the cansend and candump commands commonly used in Linux systems to perform SocketCAN communication tests.

-CAN module used in this test is used to test the CAN interface of the single-board computer . CANH and CANL are connected to CANH and CANL of the USB to CAN module respectively .

Initialize the CAN network interface

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:/ # ifconfig can0 down
root@ebyte-ubuntu:/ # ip link set can0 type can bitrate 1000000
```

The above command sets the can0 speed to 1000Kbit/S. The speeds of the two CAN devices must be set to the same! After the speed is set, turn on the can0 network card with the following command:

```
root@ebyte-ubuntu:/ # ifconfig can0 up
```

After can0 is opened, you can use the gadgets in can-utils to test data transmission and reception. Enter the following commands on the development board to receive and send:

```
root@ebyte-ubuntu:~# candump can0 //receive data
root@ebyte-ubuntu:~# cansend can0 5A1#11.22.33.44.55.66.77.88
```

The cansend command is used to send CAN data. "5A1" is the frame ID, and the "11.22.33.44.55.66.77.88" after the "#" is the data to be sent, in hexadecimal. CAN2.0 can send up to 8 bytes of data at a time, and the 8 bytes of data are separated by ".". If the USB to CAN device works properly, the receiving end will receive the 8 bytes of data sent above, as shown in the figure:

序号	时间标识	源通道	帧ID	CAN类型	方向	长度	数据
				全部	全部		
0	0.000000	0	0x5A1	CAN	Rx	8	11 22 33 44 55 66 77 88

4.2.5. USB

This section verifies the feasibility of the USB Host driver through related commands or hot plugging and USB HUB, and realizes the functions of reading and writing U disk and USB enumeration.

4.2.5.1. 1. Insert the USB drive

Insert the USB drive and use lsblk to find the storage device.

```
root@ebyte-ubuntu:~# lsblk
NAME          MAJ:MIN RM    SIZE RO TYPE MOUNTPOINT
sda             8:0     1 58.6G    0 disk
|-sda1          8:1     1 58.6G    0 part
`-sda2          8:2     1       1M    0 part
mtdblock0    31:0     0    512K    0 disk
mtdblock1    31:1     0    512K    0 disk
mtdblock2    31:2     0    512K    0 disk
mtdblock3    31:3     0    512K    0 disk
mtdblock4    31:4     0      4M    0 disk
mtdblock5    31:5     0      4M    0 disk
mtdblock6    31:6     0      4M    0 disk
mtdblock7    31:7     0      4M    0 disk
mtdblock8    31:8     0    238M    0 disk
```

4.2.5.2. U disk mount read and write

Mount USB drive

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
root@ebyte-ubuntu:~# mount /dev/sda1 /mnt/sda/
```

Reading Files

Create some files in the USB drive in advance

```
root@ebyte-ubuntu:~# mount /dev/sda1 /mnt/sda/
root@ebyte-ubuntu:~# ls /mnt/sda/
MobaXterm_Installer_v12.3.zip    conf
'System Volume Information'      setup.exe
 autorun.inf                      sources
 boot                             support
```

Writing files

```
root@ebyte-ubuntu:~# mount /dev/sda1 /mnt/sda/
root@ebyte-ubuntu:~# ls /mnt/sda/
MobaXterm_Installer_v12.3.zip    conf
'System Volume Information'      setup.exe
 autorun.inf                      sources
 boot                             support
```

4.2.6. OTG

The hardware does not support automatic switching between OTG master and slave modes, so the device tree OTG node can only be configured as master mode or slave mode.

```
/*otg*/
&usbotg_hs {
 phys = <&usbphyc_port1 0>;
 phy-names = "usb2-phy";
 // dr_mode = "peripheral"; //otg Slave
 dr_mode = "host"; // otg master
 vbus-supply = <&vbus_otg>;
 status = "okay";
};
```

After configuring it as the main mode, use the USB to Type-C female to connect the USB device, and it will be automatically mounted, such as a mouse or a USB flash drive.



After configuring it as slave mode, use a USB - TypeC data cable to connect to the PC. Then load the USB mass storage device driver usb_f_mass_storage and libcomposite , and then mount the storage

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

device on the board to the PC. For example, a USB flash drive, /dev/sda1 , or a partition of MMC.

```
depmod
Mount the driver and share the USB drivemodprobe libcomposite
modprobe usb_f_mass_storage
modprobe g_mass_storage file=/dev/sda1 removable=1
Uninstall the driver
modprobe g_mass_storage -r
modprobe usb_f_mass_storage -r
modprobe libcomposite -r
```

4.2.7. SD Card

ECB10-135A5M5M-I uses SDMMC1 to connect Micro SD .

4.2.7.1. Check TF card capacity

The TF card partition information and capacity can be queried through the fdisk -l command:

```
lsblk
NAME            MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
|-mmcblk0p1 179:1    0   256K  0 part
|-mmcblk0p2 179:2    0   256K  0 part
|-mmcblk0p3 179:3    0   256K  0 part
|-mmcblk0p4 179:4    0   256K  0 part
|-mmcblk0p5 179:5    0     4M  0 part
|-mmcblk0p6 179:6    0     4M  0 part
|-mmcblk0p7 179:7    0   512K  0 part
|-mmcblk0p8 179:8    0    64M  0 part
`-mmcblk0p9 179:9    0 29.7G  0 part
```

4.2.7.2. TF card performance test

The performance test mainly tests the reading and writing speed of files on the eMMC under the Linux system, and is generally tested by combining the time and dd commands. Mount the TF card partition to be tested. Here, the last partition /dev/mmcblk0p 9 is taken as an example, and the mount directory is /mnt /rootfs .

Write file test

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
time dd if=/dev/zero of=/mnt/rootfs/home/tempfile bs=1M count=100 conv=fdatasync
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 5.95716 s, 17.6 MB/s

real    0m6.203s
user    0m0.000s
sys     0m2.125s
```

Read file test

Ignore the effect of cache when reading files. This can be done by specifying the parameter iflag=direct,nonblock.

```
time dd if=/mnt/rootfs/home/tempfile of=/dev/null bs=1M count=100 iflag=direct,nonblock
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 4.46744 s, 23.5 MB/s

real    0m4.495s
user    0m0.000s
sys     0m0.059s
```

4.2.8. Display

ECB10-135A5M5M-I supports two display solutions: HDMI and LCD:

HDMI display: Since the STM32MP135CPU does not have an HDMI-related controller, HDMI converts RGB into HDMI signals for output through the display conversion chip SII9022A, supporting a maximum resolution of 1280 x 720@60fps.

LCD display: The LCD test section will demonstrate the operation of Linux 's drm device to achieve LCD output display RGB color and color synthesis test. The LCD uses the RGB888 display mode.

In the uboot stage, the display scheme is switched by selecting the corresponding device tree file.

4.2.8.1. HDMI

3 " in the boot mode during the uboot startup phase , and then press the "Enter" key.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
1:      stm32mp135-ebyte-produce
2:      stm32mp135-discovery-lcd
3:      stm32mp135-discovery-hdmi
Enter choice: 3
3:      stm32mp135-discovery- hdmi
```

4.2.8.2. LCD

2 " under the startup mode in the uboot startup log stage , and then press the "Enter" key.

```
1:      stm32mp135-ebyte-produce
2:      stm32mp135-discovery-lcd
3:      stm32mp135-discovery-hdmi
Enter choice: 2
2:      stm32mp135-discovery- lcd
```

4.2.9. touch

evtest command test

Execute "evtest" in the terminal to enter the test interface. Select the test peripheral as touch screen, here the default is input interrupt 0 , select " 0 " in the test interface and press Enter to start the test.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:          Goodix Capacitive TouchScreen
/dev/input/event1:          wake_up
Select the device event number [0-1]: 0
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x416 product 0x38f version 0x1060
Input device name: "Goodix Capacitive TouchScreen"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 59 (KEY_F1)
    Event code 60 (KEY_F2)
    Event code 61 (KEY_F3)
    Event code 62 (KEY_F4)
    Event code 63 (KEY_F5)
    Event code 64 (KEY_F6)
    Event code 125 (KEY_LEFTMETA)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value          0
      Min            0
      Max         1023
    Event code 1 (ABS_Y)
      Value          0
      Min            0
      Max          599
    Event code 47 (ABS_MT_SLOT)
      Value          0
      Min            0
    Max              4
  Event code 48 (ABS_MT_TOUCH_MAJOR)
    Value          0
    Min            0
    Max          255
  Event code 50 (ABS_MT_WIDTH_MAJOR)
    Value          0
    Min            0
    Max          255
Event code 53 (ABS_MT_POSITION_X)
    Value          0
```

```
              Min             0
              Max          1023
            Event code 54 (ABS_MT_POSITION_Y)
              Value           0
              Min             0
              Max           599
            Event code 57 (ABS_MT_TRACKING_ID)
              Value           0
              Min             0
              Max         65535
        Properties:
          Property type 1 (INPUT_PROP_DIRECT)
        Testing ... (interrupt to exit)
        Event: time 946684926.419206, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID),
value 0
        Event: time 946684926.419206, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X),
value 248
        Event: time 946684926.419206, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y),
value 313
        Event: time 946684926.419206, type 3 (EV_ABS), code 48 (ABS_MT_TOUCH_MAJOR),
value 50
        Event: time 946684926.419206, type 3 (EV_ABS), code 50 (ABS_MT_WIDTH_MAJOR),
value 50
        Event: time 946684926.419206, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
        Event: time 946684926.419206, type 3 (EV_ABS), code 0 (ABS_X), value 248
        Event: time 946684926.419206, type 3 (EV_ABS), code 1 (ABS_Y), value 313
        Event: time 946684926.419206, ---------- SYN_REPORT ------------
        Event: time 946684926.472114, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y),
value 314
        Event: time 946684926.472114, type 3 (EV_ABS), code 1 (ABS_Y), value 314
```

As can be seen above, the main display coordinate values and key values are as follows:

EV_SYN : Synchronization event

EV_KEY : key event, such as BTN_TOUCH , which means it is a touch key

EV_ABS : Absolute coordinates, such as the coordinates reported by the touch screen

BTN_TOUCH : Touch button

ABS_MT_TRACKING_ID indicates the start of collecting information, and the next ABS_MT_TRACKING_ID indicates the end of collecting information

Single-point touch information is carried by ABS and sent in a certain order , such as:

ABS_X : is the absolute coordinate X relative to the screen

ABS_Y: is the absolute coordinate Y relative to the screen , and multi-touch information is carried by ABS_MT and sent in a certain order, such as:

ABS_MT_POSITION_X : Indicates the x- coordinate position of the center point of the screen contact surface .

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

ABS_MT_POSITION_Y : Indicates the Y coordinate position of the center point of the screen contact surface

4.3. Network Interface

4.3.1. Ethernet

Use ifconfig in the net-tools toolkit to manually configure the network. First, view the network device information through the ifconfig command as follows.

```
ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.250  netmask 255.255.255.0  broadcast 192.168.0.255
        ether 00:04:9f:04:d2:35  txqueuelen 1000  (Ethernet)
        RX packets 3524  bytes 3681632 (3.6 MB)
        RX errors 0  dropped 73  overruns 0  frame 0
        TX packets 1262  bytes 218100 (218.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 44  base 0xc000
```

The following is a method to manually configure the IP address 192.168.0.250 for eth0. The command is as follows:

```
ifconfig eth0 192.168.0.250 netmask 255.255.255.0 up
```

After configuring the Ethernet connection, you can use PING to perform a simple test on the network connection and ping the virtual machine on the same network segment.

```
ping 192.168.0.130
PING 192.168.0.130 (192.168.0.130) 56(84) bytes of data.
64 bytes from 192.168.0.130: icmp_seq=1 ttl=64 time=1.53 ms
64 bytes from 192.168.0.130: icmp_seq=2 ttl=64 time=1.34 ms
64 bytes from 192.168.0.130: icmp_seq=3 ttl=64 time=1.37 ms
64 bytes from 192.168.0.130: icmp_seq=4 ttl=64 time=1.30 ms
```

4.3.2. WIFI

This section mainly introduces the configuration and use of Wi-Fi under Linux. Usually, the Wi-Fi module can support two working modes, namely STA mode and AP mode. Some peripheral modules also support STA and AP modes at the same time. STA mode allows the development board to connect to an external Wi-Fi hotspot, and AP mode turns the development board into a Wi-Fi hotspot for other external devices to connect.

The development board has an onboard AP6212 Wi-Fi and Bluetooth two-in-one module. Currently, it does not support STA and AP working at the same time. The driver corresponding to the AP6212 Wi-Fi module is brcmfmac, which is loaded before use.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
modprobe brcmfmac
[ 2250.502879] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[ 2250.650222] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[ 2250.728001] brcmfmac: brcmf_fw_alloc_request: using brcm/brcmfmac43430-sdio for chip BCM43430/1
brcmfmac: brcmf_c_preinit_dcmds: Firmware: BCM43430/1 wl0: Sep 11 2018 09:22:09 version 7.45.98.65 (r707797 CY) FWID 01-b54727f
```

View loaded drivers

```
lsmod | grep brcm
brcmfmac                225280  0
cfg80211                647168  1 brcmfmac
brcmutil                16384  1 brcmfmac
```

During the driver loading process, the Wi-Fi firmware located in /lib/firmware/brcm will be loaded into the module. After the WiFi module driver is loaded successfully, the Wi-Fi network node wlan0 is generated, as shown below

```
ifconfig wlan0
wlan0: flags=4098<BROADCAST,MULTICAST>  mtu 1500
        ether 50:41:1c:b6:9c:32  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

4.3.2.1. STA mode connects to WIFI hotspot

Next, try to manually connect to the nearby Wi-Fi hotspot " TST-2.4G ", which is a Wi-Fi hotspot using WPA2 encryption with a password of 12345678 .

```
wpa_passphrase  TST-2.4G TST12345678 >> /etc/wpa_supplicant.conf
```

Connect to wifi hotspot

```
wpa_supplicant -Dnl80211 -B -c /etc/wpa_supplicant.conf -i wlan0
```

Use the -B parameter to connect to WiFi in the background. If you remove it, you can view the connection process.

After the connection is successful, an IP address needs to be assigned . Use udpchc to assign an IP address

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
udhcpc -i wlan0
udhcpc: started, v1.30.1
udhcpc: sending discover
udhcpc: sending select for 172.20.10.5
udhcpc: lease of 172.20.10.5 obtained, lease time 86400
ip: RTNETLINK answers: File exists
```

View connection status

```
iw wlan0 link
Connected to 4a:fe:e2:56:e0:36 (on wlan0)
        SSID: iPhone
        freq: 2437
        RX: 2652 bytes (17 packets)
        TX: 1806 bytes (13 packets)
        signal: -54 dBm
        tx bitrate: 24.0 MBit/s

        bss flags:      short-slot-time
        dtim period:    1
        beacon int:     100
```

Scan for Wi -Fi hotspots

```
iw dev wlan0 scan | grep SSID
        SSID: iPhone
        SSID: zhendu
        SSID: TP-LINK_\xef\xbc\x88\xe7\xa1\xac\xe4\xbb\xb6\xe7\xbb\x84)
        SSID: \x00\x00\x00\x00\x00\x00
        SSID: TP-LINK_6C35
        SSID: B05_2.4G_\xe6\x89\x93\xe5\x8d\xb0\xe6\x9c\xba
        SSID: MERCURY_2.4G_D02D
        SSID: KingOfSoftware
        SSID: \xe5\xa4\xa7\xe5\x8e\x85\xe7\xbd\x91\xe7\xbb\x9cA-5G
```

4.3.2.2. AP mode turns on W IFI hotspot

Hostapd is a wireless access point program with encryption function. It is a convenient tool for building wireless access points on Linux operating system. It supports IEEE 802.11 protocol and IEEE 802.1X/WPA/WPA2/EAP/RADIUS encryption. As a WiFi hotspot, the board needs to assign IP, routing and other network parameters to each terminal (such as mobile phone) accessing the hotspot.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

For example, create a wireless WiFi hotspot with SSID EBYTE and PASSWD 12345678. The following is a manual configuration:

When using AP mode, you need to activate wlan0 and configure a static IP address. Here, a default IP address is configured: 192.168.10.1.

```
ifconfig wlan0 192.168.10.1 up
```

We have explained above that the current Wi-Fi module does not support STA and AP modes working at the same time, so here we need to clear the STA mode configuration, as follows:

```
killall udhcpc
killall wpa_supplicant
```

wlan0 works in AP mode. When other external devices connect to this AP hotspot, it needs to dynamically allocate IP addresses to other external devices through wlan0, so wlan0 needs to be used to run the DHCP service program udhcpd. The configuration file corresponding to udhcpd is /etc/udhcpd.conf, and the content is as follows:

```
# File: /etc/udhcpd.conf
# the start and end of the IP lease block
start 192.168.10.10
end 192.168.10.254
# the interface that udhcpd will use
interface wlan0
opt         dns         8.8.8.8
option    subnet    255.255.255.0
opt         router    192.168.10.1
option    domain    local
option    lease     864000
```

The most critical step in configuring AP mode is of course to start the hostapd service. Before starting the service, you need to configure the AP mode's ssid, password, encryption algorithm, driver type, working mode, etc. through /etc/hostapd.conf. For complete parameter configuration instructions, see: http://w1.fi/cgit/hostap/plain/hostapd/hostapd.conf. The following is the /etc/hostapd.conf configuration for the current hardware, as follows:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
# File: /etc/hostapd.conf
interface=wlan0
driver=nl80211
# mode Wi-Fi (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g)
hw_mode=g
ssid=EBYTE
channel=7
wmm_enabled=0
macaddr_acl=0
# Wi-Fi closed, need an authentication
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=12345678
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

After the configuration file is prepared, execute the following command to start the hostapd service:

```
hostapd -B /etc/hostapd.conf
Configuration file: /etc/hostapd.conf
wlan0: Could not connect to kernel driver
Using interface wlan0 with hwaddr 50:41:1c:b6:9c:32 and ssid "EBYTE"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

As shown in the log above, the hotspot service can be used normally. If the Ethernet card eth0 is connected to the Internet, then the following configuration is used for IP forwarding, and the external devices connected to EBYTE can also connect to the Internet.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
echo "1" > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -s 192.168.10.1/24 -o eth0 -j MASQUERADE
```

You can use your mobile phone to connect to the above hotspots to test the Internet access.

5. References

- Linux kernel open source community:

  https://www.kernel.org/

- STM32MPU Development Community:

  https://wiki.st.com/stm32mpu/wiki/Development_zone

- STM32MP131 Datasheet
- STM32MP13 5 Data Sheet

6. Revision Notes

Revision Notes Table

Version	Modifications	Modification time	prepared by	Proofreading	Approval
V1.0	First Draft	24-0 9 -19	HSL	Wlq	WFX
V1.1	Added Chapter 3 Quick Start section	2 4-12-14	H SL	Wlq	W FX
V1.2	Modify some issues	2 5-03-11	H SL	Wlq	W FX

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

7. About Us

Technical support: service@cdebyte.com
Documents and RF Setting download link: www.cdebyte.com
Thank you for using Ebyte products! Please contact us with any questions or suggestions: info@cdebyte.com

---

Web: www.cdebyte.com
Address: B5 Mould Park, 199# Xiqu Ave, High-tech District, Sichuan, China

Chengdu Ebyte Electronic Technology Co.,Ltd.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000