

中嵌凌云 AMKN880X 快速体验手册 V1.0

(2025 年 2 月 26 日修订版)

版权声明

本产品使用手册包含的所有内容均受版权法的保护，未经北京中嵌凌云电子有限公司的书面授权，任何组织和个人不得以任何形式或手段对整个手册和部分内容进行复制和转载。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我司概不承担其他责任。并且我司对本产品的销售和使用不作任何明示或暗示的担保，包括对产品特定用途的适用性，适销性或对任何专利权、版权或其他知识产权的侵权责任等均不作担保。我司对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，我司可能随时会对产品描述和相关的功能调整或技术改进，保留修改文档中任何内容的权利，恕不另行通知。

商标声明

AMKN 系北京中嵌凌云电子有限公司注册商标，未经书面授权，任何人不得以任何方式使用该商标、标记。

销售及服务网络

北京

销售电话：185 0042 1002

地址：北京市海淀区吴家场路 1 号院 2 号楼

邮箱：sales@embedarm.com

西安

销售电话：029-6888 8268（工作日）

手机：189 9285 2102

地址：西安市曲江新区旺座曲江 H 座 3003 室

邮箱：sales@embedarm.com

技术支持：

电话：029-8877 2044（工作日）

手机：188 0108 0298

微信：133 9928 8868

邮箱：embedarm@126.com

网址：www.zqlymcu.com

版本变更

下面表格显示本产品使用手册在不同时期的修订版本及修订原因说明：

版本	修改内容	完成日期	修订部门
V1.00	初稿	2025. 2. 26	研发部

订货型号：

序号	订货型号	说明	备注
1	AMKN8801		工业级：-30~70° (不结露)
2	AMKN8802		

目录

一、 准备工作	5
二、 硬件资源	6
三、 出厂固件烧录	8
四、 AMKN880X 功能测试	12
4.1 串口测试	13
4.2 网络测试	16
4.3 PWM 测试	18
4.4 IO 测试	21
4.5 CAN 测试	23
4.6 ADC 测试	25
4.7 DAC 测试	25
4.8 WIFI 测试	26

一、准备工作

本文主要讲述如何快速体验出厂镜像的部分功能对接口进行测试，在测试网口之前请运行 /root/start_network.sh 脚本，该脚本用来配置以太网的 IP 地址、MAC 地址等信息。运行后终端会显示目前已经打开的网络信息，用户可以根据需求修改 start_network.sh 文件的内容

```
[root@embedarm-stm32mp1]~$ ./start_network.sh
[ 12.722372] stm32-dwmac 5800a000.ethernet eth0: PHY [stmmac-0:00] driver [RTL8211F Gigabit Ethernet]
[ 12.741187] dwmac4: Master AXI performs any burst length
[ 12.745094] stm32-dwmac 5800a000.ethernet eth0: No Safety Features support found
[ 12.752604] stm32-dwmac 5800a000.ethernet eth0: IEEE 1588-2008 Advanced Timestamp supported
[ 12.761674] stm32-dwmac 5800a000.ethernet eth0: registered PTP clock
[ 12.767456] stm32-dwmac 5800a000.ethernet eth0: configuring for phy/rgmii-id link mode
ifconfig: SIOCSIFHWADDR: Device or resource busy
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:53
can1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:55
eth0      Link encap:Ethernet  HWaddr B8:AE:1D:01:01:00
          inet addr:192.168.1.63  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:50
```

图 1.1 start_network.sh 脚本运行后

二、 硬件资源

AMKN8801 硬件资源如下表所示：

标号	功能说明	连接对象
JP1	4路模拟信号输入，0~10V或4~20mA电流环	各种电压或电流环型传感器
	2路模拟信号输出，范围：0~10V	各种电压控制型设备
JP2	8路DI隔离输入(DI1~DI8)	开关量输入
JP3	4路DO隔离输出(DO1~DO4)	开关量输出
	4路继电器干接点输出(N01~N04)	开关量输出
JP4	下载器接口(SWD)，调试下载程序	ARM下载器
JP5	脉冲输出端口4	各种电机驱动器
	脉冲输入端口2	各种脉冲信号及正交编码器
JP6	2路光电或数字隔离CAN通信端口	外部通信设备
	4路光电或数字隔离RS485通信端口	外部通信设备
	2路RS232通信端口	外部通信设备
JP7	MIPI DSI屏接口(0.5mmFPC 18P)	MIPI DSI屏
JP8	喇叭与耳机功能切换(短接：耳机；不短接：喇叭)	
JP10	RGBLCD屏接口(0.5mmFPC 40P)	RGBLCD屏
JP11	HDMI 接口	HDMI接口屏
JP14	立体声输出接口	耳机或功放输入
SP1	PH2.0喇叭接口	1W/8欧
SP2		1W/8欧
JP15	2.4G天线接口(WIFI或蓝牙)	SMA内孔
JP18	+22V~+26V直流电源输入接口	电源
JP19	TF SD卡座	
JP23	1000M以太网接口	以太网接口设备
USB1	TYPE C 接口	USB设备
USB2	TYPE A 双接口	USB设备
USB3	TYPE A 双接口	USB设备
SW1	拨码开关	B00T选择
K1	复位按键	系统复位
LED1	双LED	电源/运行指示

表 2-1 硬件资源展示表

AMKN8802 硬件资源表如下所示：

标号	功能说明	连接对象
JP1	MIPI DSI屏接口 (0.5mmFPC 18P)	MIPI DSI屏
JP2	TF SD卡座	
JP3	下载器接口 (SWD), 调试下载程序	ARM下载器
JP4	8路DI隔离输入 (DI1~DI8)	开关量输入
JP5	8路DI隔离输入 (DI9~DI16)	开关量输入
JP6	4路继电器干接点输出 (N01~N04)	开关量输出
JP18	4路继电器干接点输出 (N05~N08)	开关量输出
JP7	脉冲输出端口1-3	各种电机驱动器
JP14	脉冲输出端口4-6	各种电机驱动器
JP8	光电或数字隔离RS485通信端口1-2	外部通信设备
JP15	光电或数字隔离RS485通信端口3-4	外部通信设备
JP9	2. 4G天线接口 (WIFI或蓝牙)	SMA内孔
JP10	1000M以太网接口	以太网接口设备
JP11	+22V~+26V直流电源输入接口	电源
JP12	4路0~10V和4路0~20mA模拟信号输入	各种电压或电流环型传感器
JP13	8路NPN MOS输出	开关量输出
JP16	2路RS232通信端口	外部通信设备
JP17	2路光电或数字隔离CAN通信端口	外部通信设备
USB1	TYPE C 接口	USB设备
USB2	TYPE A 双接口	USB设备
USB3	TYPE A 双接口	USB设备
USB4	TYPE C 转串口	外部通信
SW1	拨码开关	B00T选择
K1	复位按键	系统复位
K2-K6	功能按键	自定义

表 2-2 硬件资源展示表

三、 出厂固件烧录

首先确保启动选项 BT0-BT2 的电平为 000，也就是均为低电平。按照图中所示连接下载软件和开发板，点击刷新按钮如果连接没有错误则会弹出一个 USB 设备，搜索到 USB 后直接点击 connect 按钮。出厂镜像需要根据工控板的型号（AMKN8801、AMKN8802）来选择对应镜像下载，下面以 AMKN8801 为例。

出厂镜像在开发资料 ->04_系统镜像->AMKN8801->04_Windows 下载文件夹，打开 STM32CubeProgrammer 软件打开文件选择开发资料 04_系统镜像->04_Windows 下载文件夹 tf-a. tsv。连接和打开系统镜像所存储的文件后直接点击 Download 按键开始下载镜像，由于镜像比较大下载时间需要 5-15 分钟，请耐心等待。



图 3.1 拨码开关设置 USB 启动

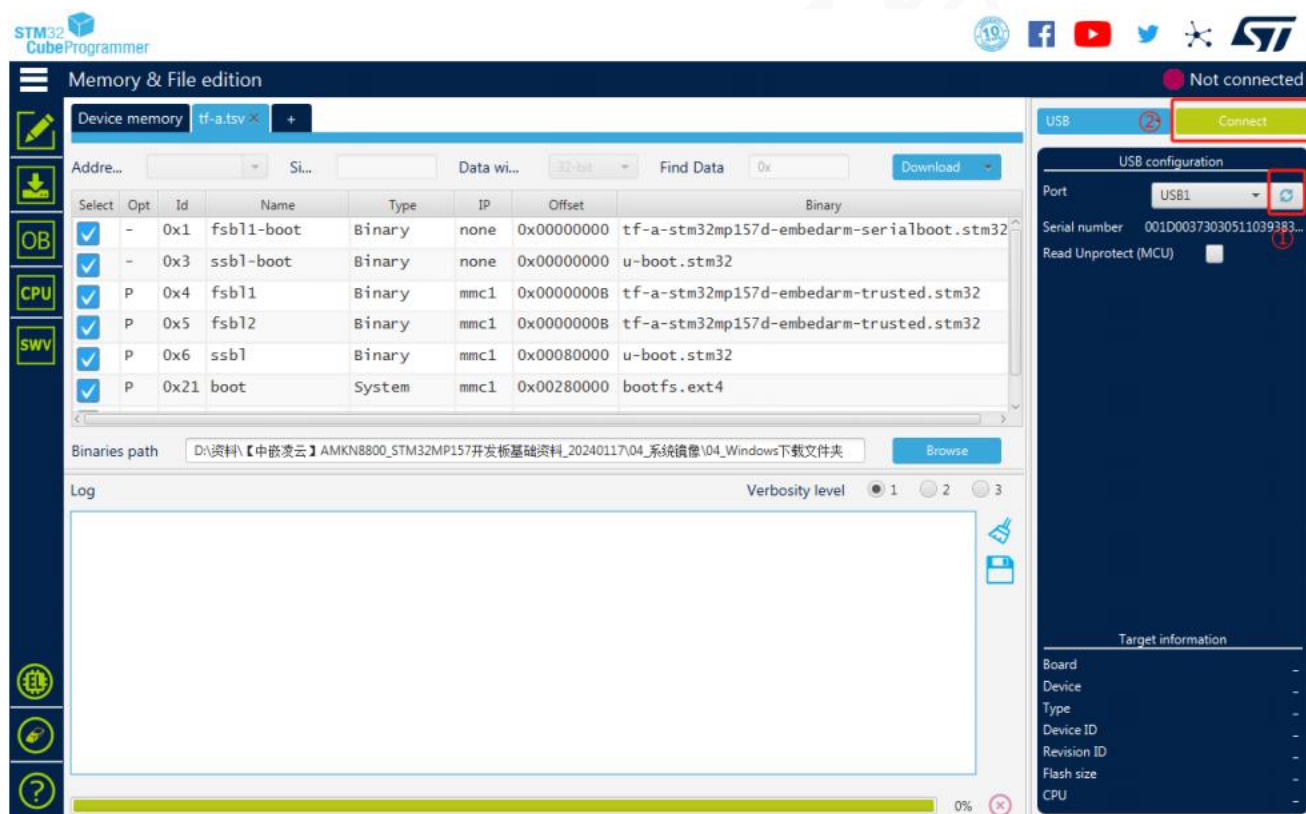


图 3.2 下载软件连接开发板

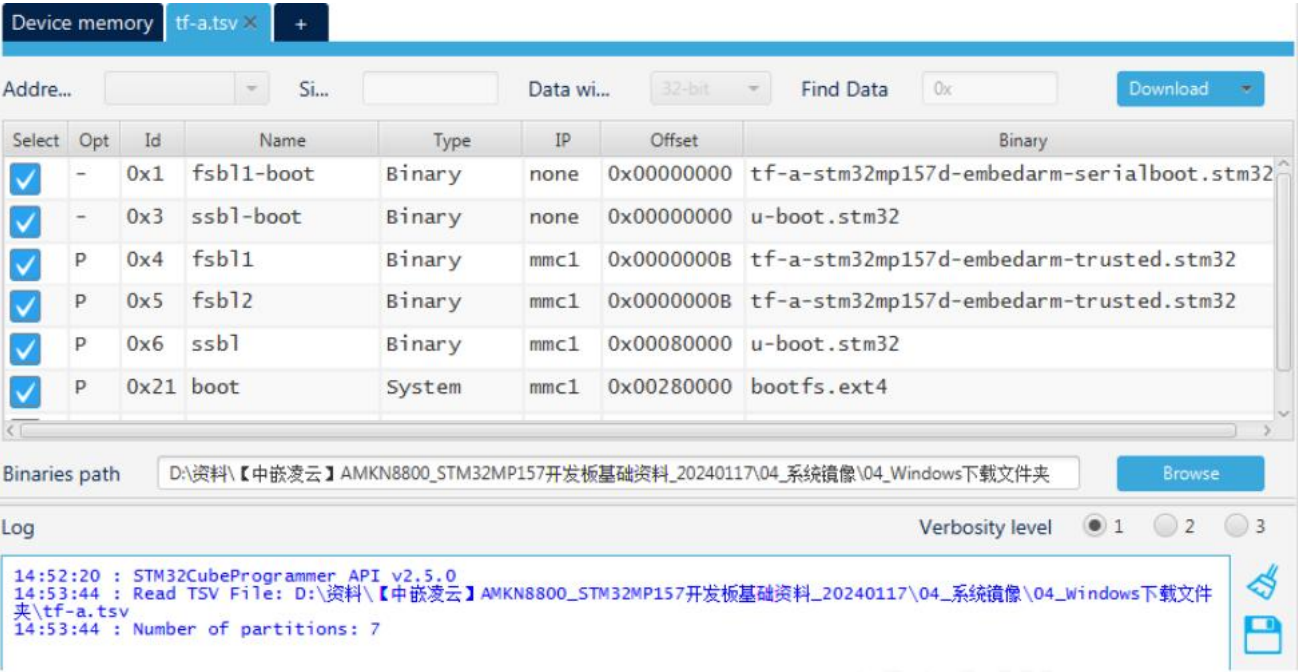


图 3.3 烧录软件选项设置

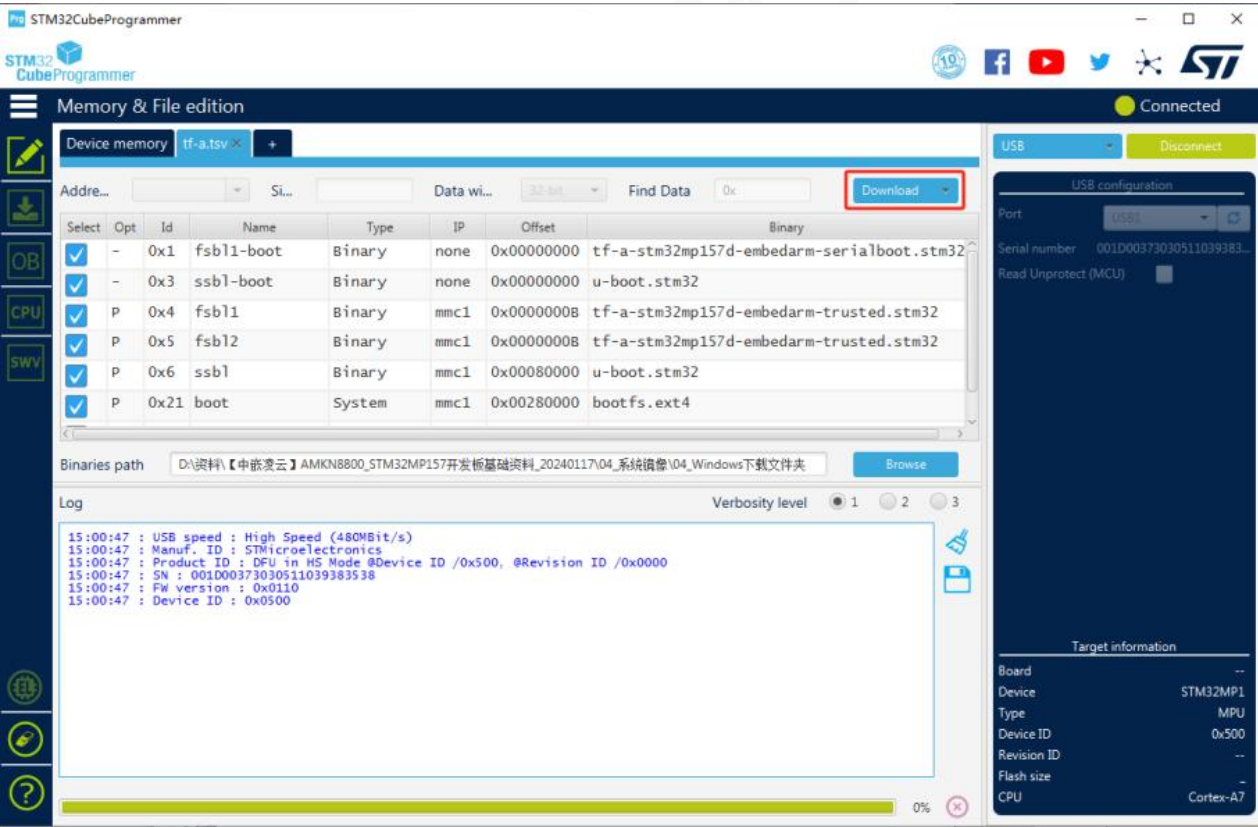


图 3.4 下载镜像到开发板

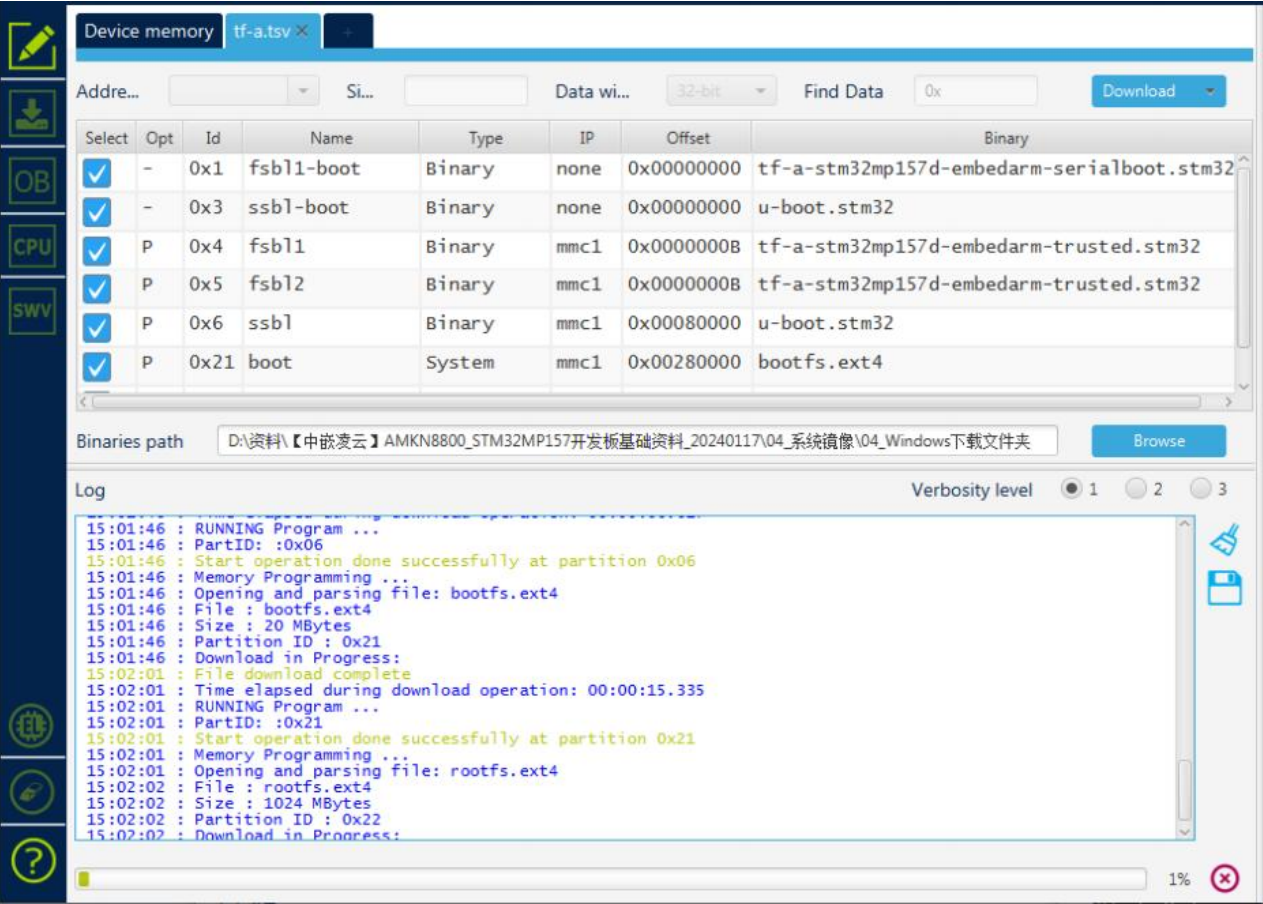


图 3.5 等待镜像下载

下载完成后，将启动方式选择为 EMMC 启动，按下复位按钮即可



图 3.6 EMMC 启动

```
[ 5.945995] fusb302 0-0022: PD disabled
[ 9.208816] stm32-dwmac 5800a000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[ 9.256806] IP-Config: Complete:
[ 9.258601] device=eth0, hwaddr=b8:ae:1d:01:01:00, ipaddr=192.168.1.63, mask=255.255.255.0, gw=192.168.1.1
[ 9.268858] host=192.168.1.63, domain=, nis-domain=(none)
[ 9.274630] bootserver=192.168.1.229, rootserver=192.168.1.229, rootpath=
[ 9.282972] ALSA device list:
[ 9.284882] No soundcards found.
[ 9.301481] VFS: Mounted root (nfs filesystem) on device 0:18.
[ 9.306485] devtmpfs: mounted
[ 9.310994] Freeing unused kernel memory: 1024K
[ 9.367463] Run /sbin/init as init process
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting mdev... OK
[ 11.609528] cfg80211: module verification failed: signature and/or required key missing - tainting kernel
[ 11.640849] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[ 11.649967] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
Initializing random number generator: OK
Saving random seed: OK
Starting system message bus: done
Starting network: RTNETLINK answers: File exists
FAIL
Starting gpsd: OK
Starting sshd: /var/empty must be owned by root and not group or world-writable.
OK
Starting vsftpd: OK

Welcome to EmbedArm STM32MP157
embedarm-stm32mp1 login: █
```

图 3.7 开发板正常启动

板卡启动后在串口终端（默认串口一）会打印图 所示内容，要进入根文件系统需要输入 root 用户，输入后的打印信息是挂载本司编写的 DI 和 DO 测试驱动。

```
Welcome to EmbedArm STM32MP157
embedarm-stm32mp1 login: root
[ 166.222084] Do_Device major=241,minor=0
[ 166.252807] DI_Device major=240,minor=0
```

图 3.8 输入 root 用户进入根文件系统



图 3.9 开发板运行展示图

四、 AMKN880X 功能测试

如购买 LCD 屏幕或者有 HDMI 接口的可以通过本司编写的 Qt 应用软件对部分功能进项测试，进入根文件目录后输入如下命令运行测试程序：

hellow



图 4.1 测试软件运行结果展示

4.1 串口测试

图形化界面测试方法

点击屏幕的串口测试按钮进入串口测试页面，如图所示：

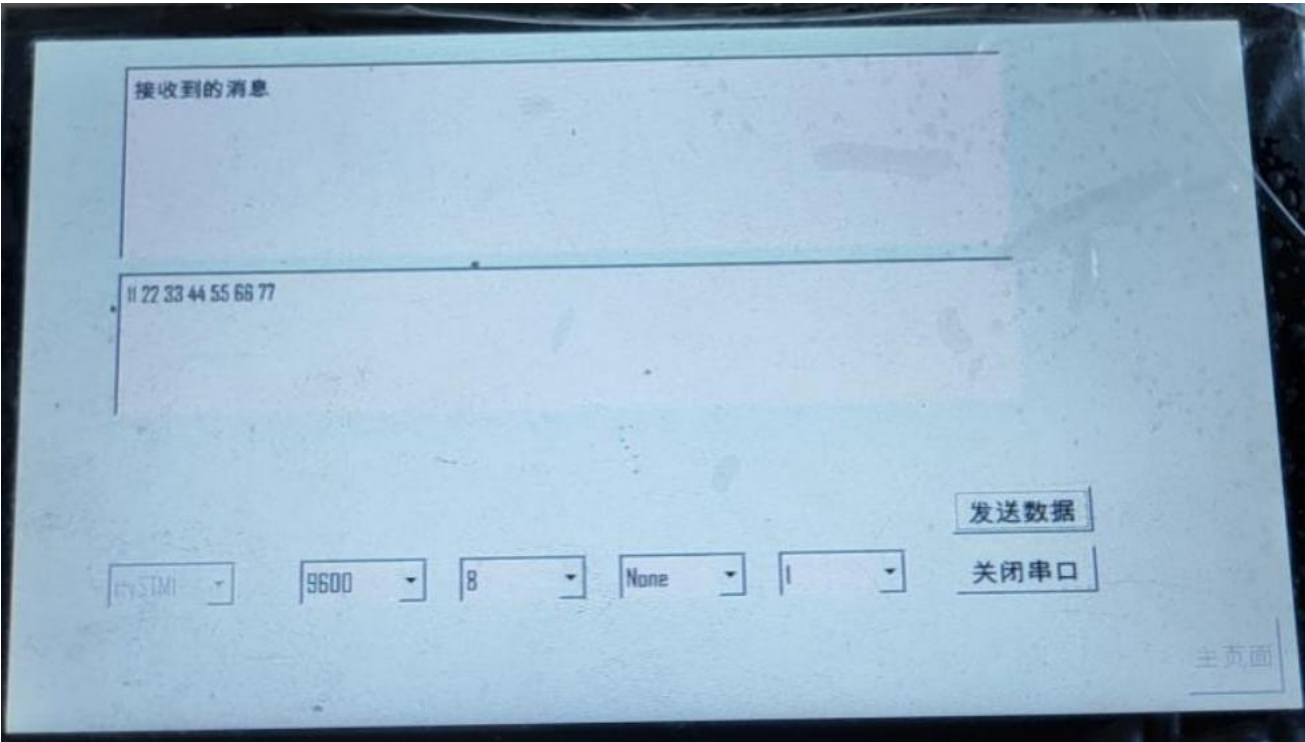


图 4.1.1 串口测试页面展示

选择串口端口号，端口号和对应串口的关系如下表所示：

端口	串口序号
ttySTM0	串口一
ttySTM1	串口二
ttySTM2	串口三
ttySTM3	串口四
ttySTM4	串口五
ttySTM5	串口六
ttySTM6	串口七

表 4-1-1 AMKN8801 端口号与对应串口的关系

端口	串口序号
ttySTM0	串口一
ttySTM1	串口三
ttySTM2	串口四
ttySTM3	串口五
ttySTM4	串口六
ttySTM5	串口七
ttySTM6	串口八

表 4-1-2 AMKN8802 端口号与对应串口的关系

选择串口的端口号后，确认波特率和数据位后点击发送数据，在串口助手上就可以显示发送的数据信息。串口助手点击发送数据在测试软件的接受也可以看到接受的串口助手数据。



图 4.1.2 串口助手显示接受数据

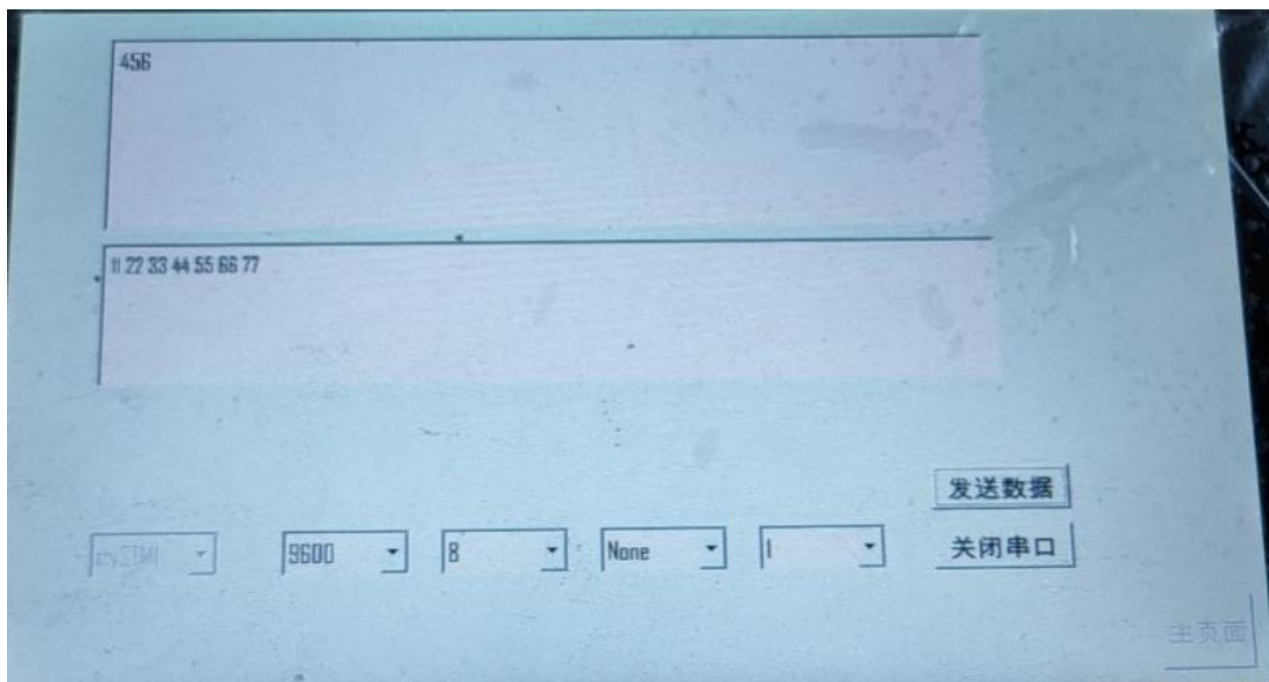


图 4.1.3 串口测试界面接受数据展示

非图形化界面测试方法

测试串口使用 `uartapp` 命令，默认默认的波特率是 115200、数据位数 8、无奇偶校验、停止位 1。具体命令如图 4.1.5 所示：

```
[root@embedarm-stm32mp1]:/dev$ uartapp -h
Usage: uartapp [选项]

可选选项:
  --dev=DEVICE      指定串口终端设备名称, 譬如--dev=/dev/ttySTM1
  --type=TYPE       指定操作类型, 读串口还是写串口, 譬如--type=read(read表示读、write表示写、其它值无效)

可选选项:
  --brate=SPEED     指定串口波特率, 譬如--brate=115200
  --dbit=SIZE       指定串口数据位个数, 譬如--dbit=8(可取值为: 5/6/7/8)
  --parity=PARITY   指定串口奇偶校验方式, 譬如--parity=N(N表示无校验、0表示奇校验、E表示偶校验)
  --sbit=SIZE       指定串口停止位个数, 譬如--sbit=1(可取值为: 1/2)
  -h               查看本程序使用帮助信息
```

图 4.1.4 uartapp 命令帮助

```
[root@embedarm-stm32mp1]:/dev$ uartapp --dev=ttySTM1 --type=write --brate=9600
default data bit size: 8
default parity: N
default stop bit size: 1
```

图 4.1.5 uartapp 命令测试

```
[10:52:32.456] 收←◆11 22 33 44 55 66 77 88
[10:52:33.456] 收←◆11 22 33 44 55 66 77 88
[10:52:34.457] 收←◆11 22 33 44 55 66 77 88
[10:52:35.457] 收←◆11 22 33 44 55 66 77 88
[10:52:36.457] 收←◆11 22 33 44 55 66 77 88
[10:52:37.456] 收←◆11 22 33 44 55 66 77 88
[10:52:38.456] 收←◆11 22 33 44 55 66 77 88
[10:52:39.456] 收←◆11 22 33 44 55 66 77 88
[10:52:40.456] 收←◆11 22 33 44 55 66 77 88
[10:52:41.457] 收←◆11 22 33 44 55 66 77 88
[10:52:42.457] 收←◆11 22 33 44 55 66 77 88
[10:52:43.457] 收←◆11 22 33 44 55 66 77 88
[10:52:44.458] 收←◆11 22 33 44 55 66 77 88
[10:52:45.458] 收←◆11 22 33 44 55 66 77 88
[10:52:46.458] 收←◆11 22 33 44 55 66 77 88
[10:52:47.457] 收←◆11 22 33 44 55 66 77 88
[10:52:48.457] 收←◆11 22 33 44 55 66 77 88
[10:52:49.457] 收←◆11 22 33 44 55 66 77 88
[10:52:50.457] 收←◆11 22 33 44 55 66 77 88
[10:52:51.458] 收←◆11 22 33 44 55 66 77 88
[10:52:52.458] 收←◆11 22 33 44 55 66 77 88
[10:52:53.458] 收←◆11 22 33 44 55 66 77 88
[10:52:54.459] 收←◆11 22 33 44 55 66 77 88 |
```

图 4.1.6 串口调试助手接受数据

4.2 网络测试

图形化界面测试方法

回到主页面后，点击“网络测试”按钮进入网络测试页面，页面展示如图所示

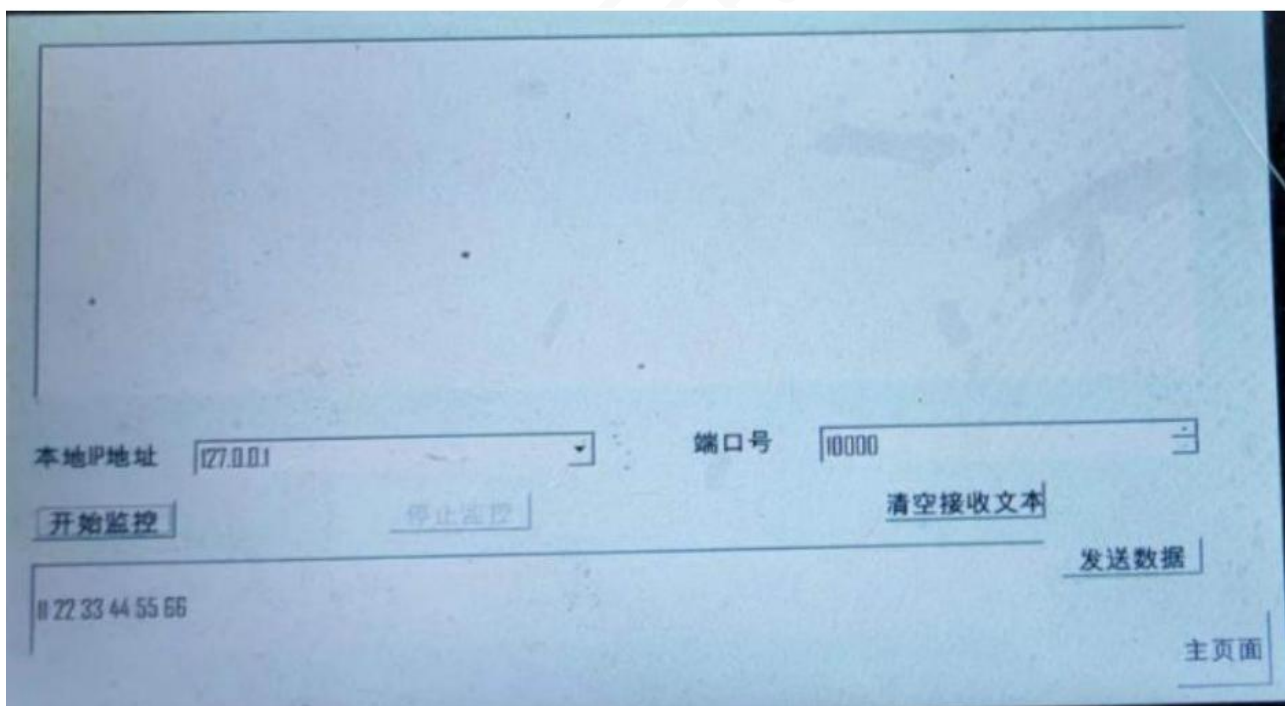


图 4.2.1 网络测试页面展示

网络测试页面是使用 TCP 协议栈，板卡是作为 TCPServer，接好网线选择监控的本机 IP 地址和端口号，点击“开始监控”按钮，在电脑上打开 TCPClient，输入板卡 IP 地址和端口

号点击连接。

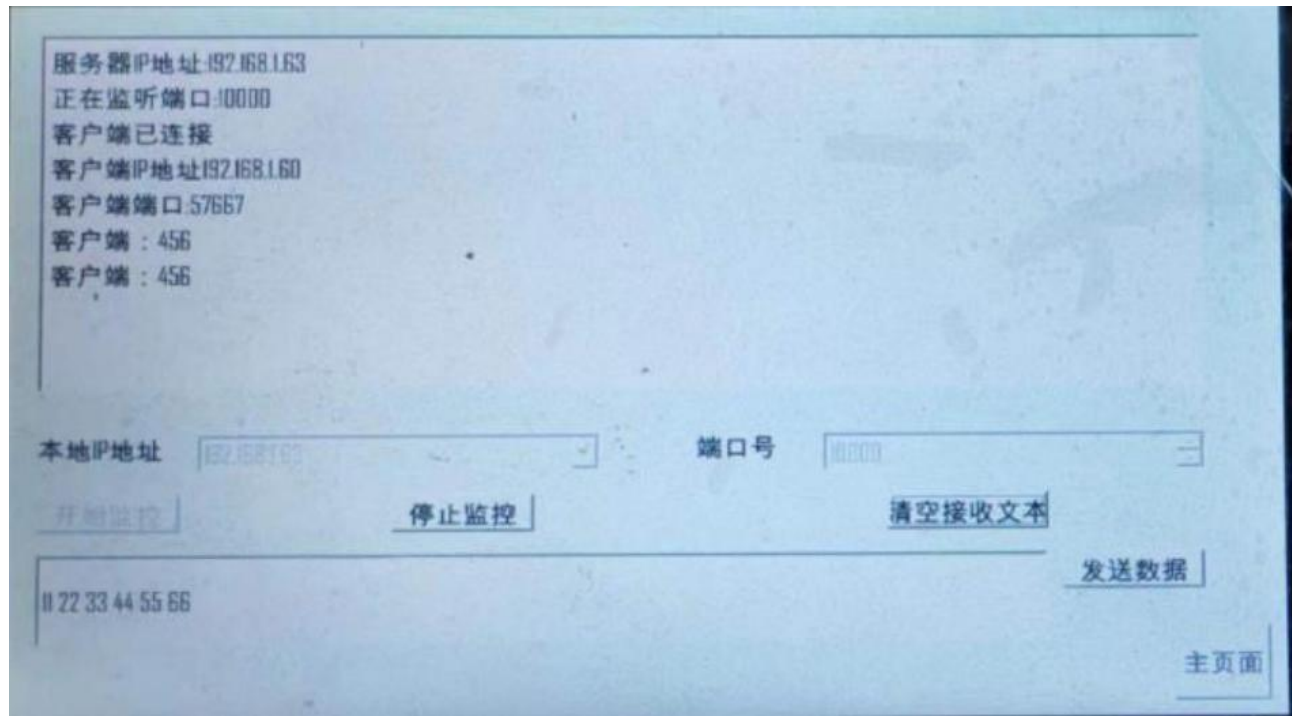


图 4.2.2 板卡接受数据展示

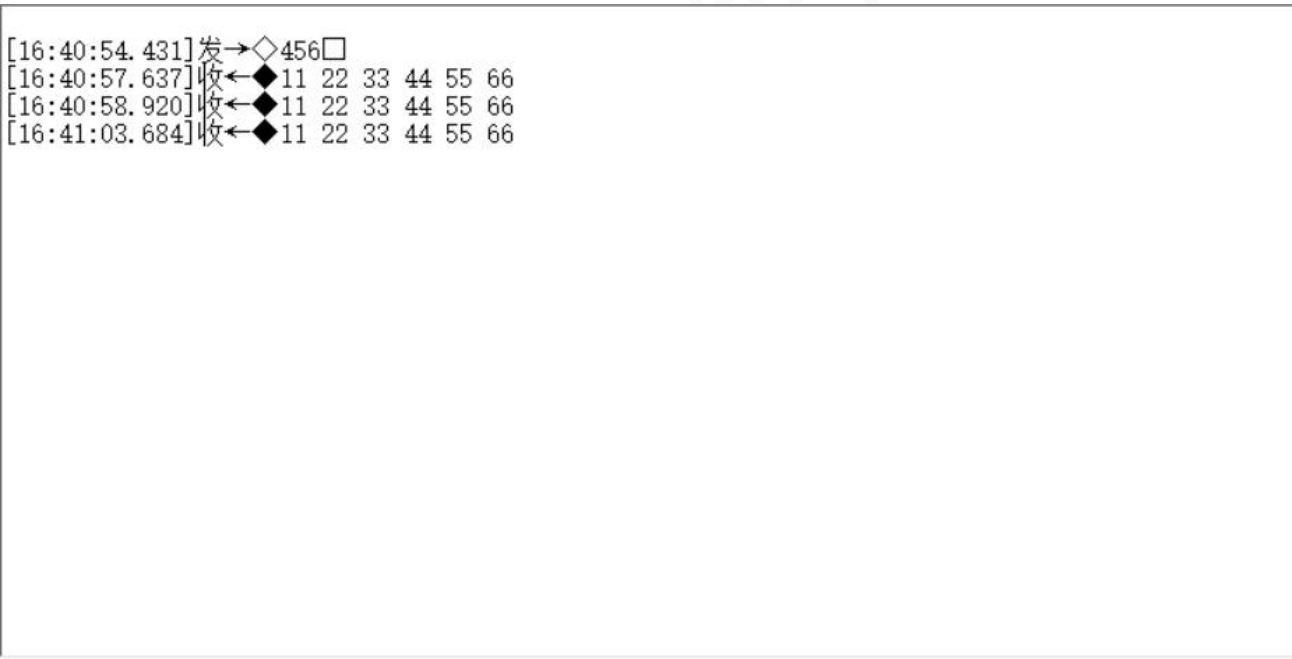


图 4.2.3 TCPClient 接受数据页面展示

非图形化界面测试方法

使用 netserver 命令进行测试，直接运行该命令即可。然后打开 TCPClient 连接板卡端口号为 8888。连接成功后会打印连接的客户端 IP 地址，然后客户端发数据即可。

```
[root@embedarm-stm32mp1]:/dev$ netserver  
有客户端接入...  
客户端主机的IP地址: 192.168.1.60  
客户端进程的端口号: 48078
```

图 4.2.4 运行 netserver 测试命令

```
[root@embedarm-stm32mp1]:/dev$ netserver  
有客户端接入...  
客户端主机的IP地址: 192.168.1.60  
客户端进程的端口号: 48078  
from client: 456  
from client: 456
```

图 4.2.5 板卡 TCP 数据

```
[11:00:26.067]发→◇456□  
[11:00:28.781]发→◇456□
```

图 4.2.6 TCPClient 发数据

4.3 PWM 测试

图形化测试方法

回到主页面后点击“PWM 测试”按钮，进入 PWM 测试页面，如下图所示：

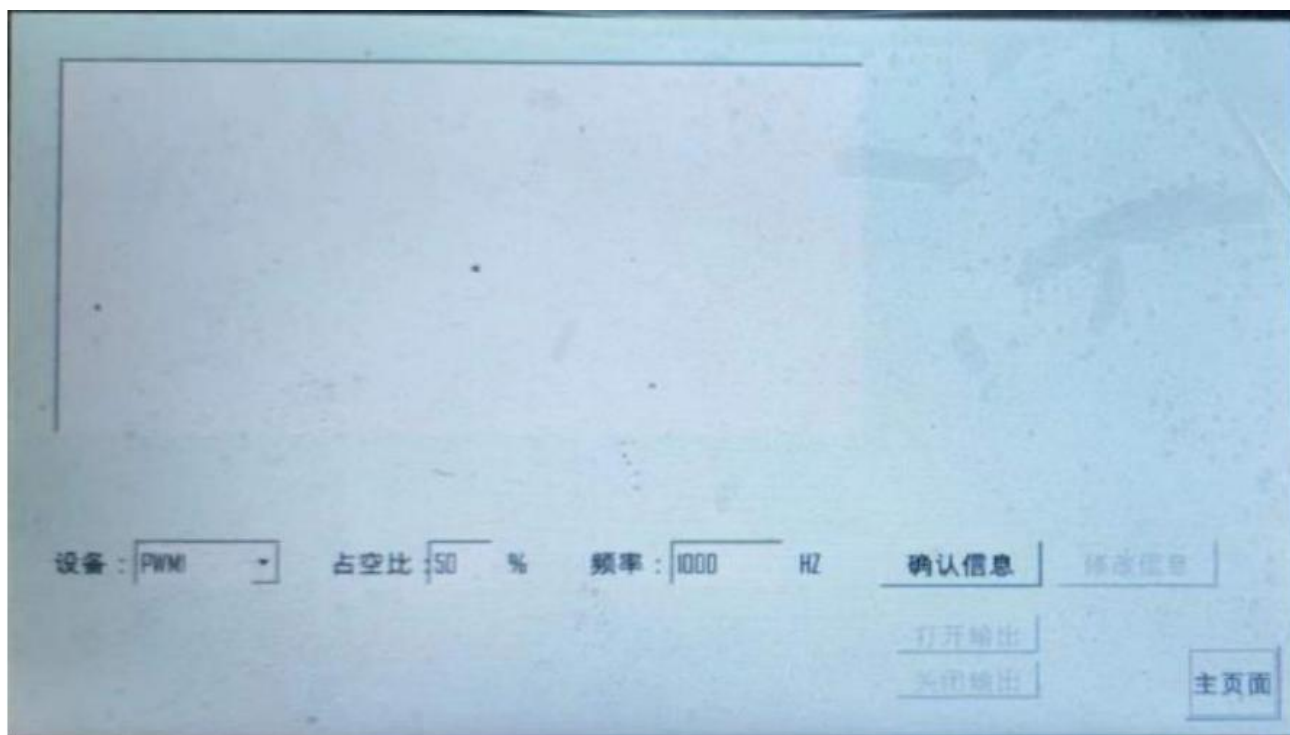


图 4.3.1 PWM 测试页面展示图

进入页面后选择 PWM 设备，输入占空比，输入频率（数据输入可以在 USB 口上接入 USB2.0 的键盘作为输入）。确认后点击“确认信息”按钮，然后点击“打开输出”。对应的

PWM 设备就开始输出方波信号

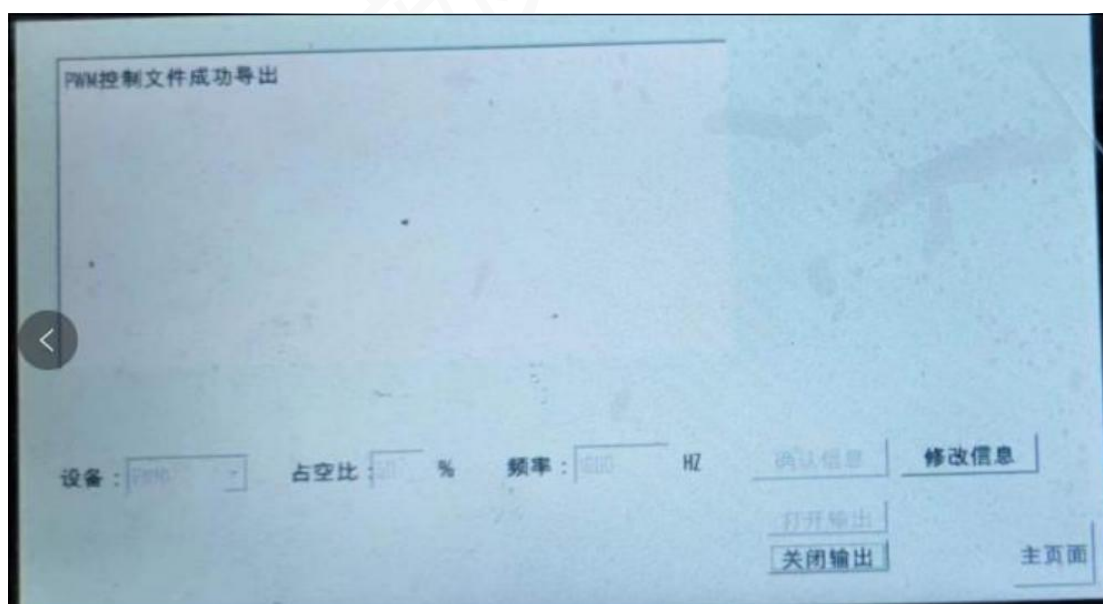


图 4.3.2 PWM 开始输出界面

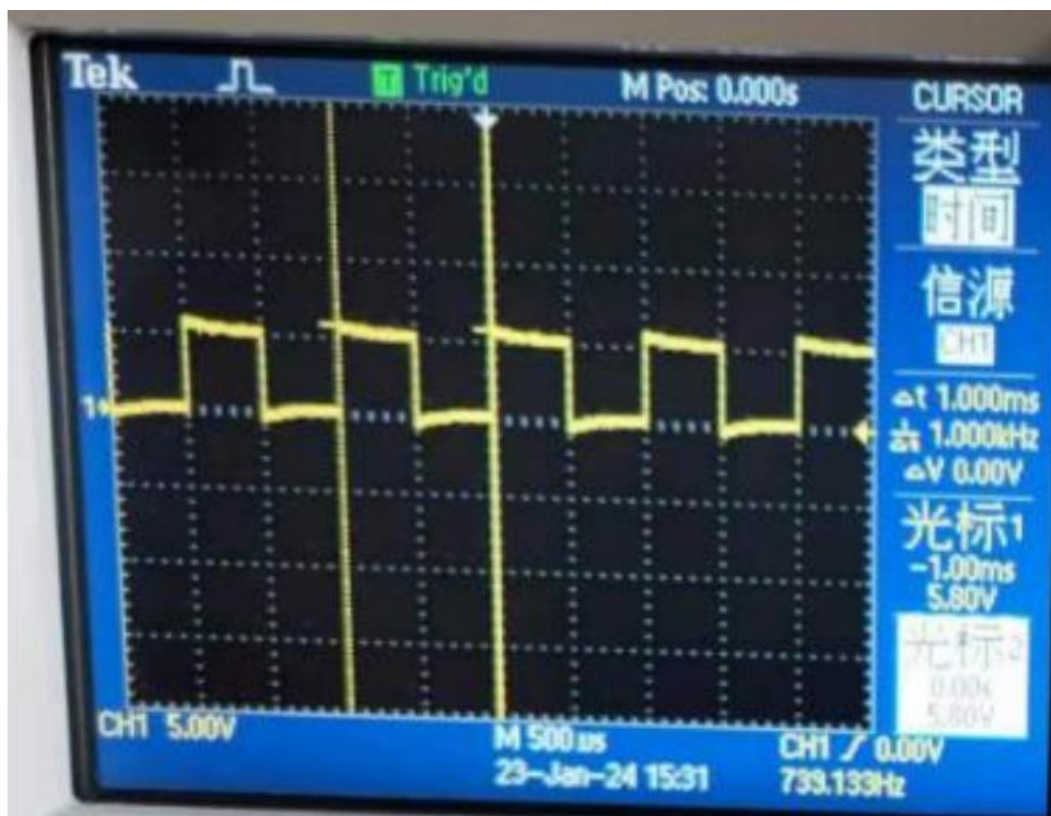


图 4.3.3 示波器展示图

非图形化测试方法

使命 `pwmapp` 命令对 PWM 接口进项测试，如下图所示展示该命令的具体用法

```
[root@embedarm-stm32mp1]:/dev$ pwmapp -h
Usage: pwmapp

必填项目：
--pwm=DEVICE_NAME 具体PWM设备文件名称，例如：--pwm=pwmchip8
--ch=CHANNEL       定时器通道，例如--ch=2

可选项目：
--period=PERIOD    输出PWM周期 单位：Hz 例如：--period=2
--duty=DUTY_CYCLE  输出PWM占空比时间 1-100% 例如：--duty=50
--disen=ENABLE     PWM是否使能 例如：--disen=0
--help或-h        查看本程序使用帮助信息
```

图 4.3.4 pwmapp 命令帮助展示

具体 AMKN8801 PWM 设备文件名称可选项如下表所示：

原理图备注	引脚名称	引脚功能备注	驱动文件名称
PWM1	PA10	TIM1 (0x 44000000) _CH3	/sys/class/pwm/pwmchip8
PWM2	PB10	TIM2 (0x 40000000) _CH3	/sys/class/pwm/pwmchip0
PWM3	PB8	TIM16 (0x 44007000) _CH1	/sys/class/pwm/pwmchip12
PWM4	PB9	TIM17 (0x 44008000) CH1	/sys/class/pwm/pwmchip13

表 4-3-1 AMKN8801 PWM 设备对应驱动文件名称

原理图备注	引脚名称	引脚功能备注	驱动文件名称
PWM1	PE11	TIM1 (0x44000000) CH2	/sys/class/pwm/pwmchip12
PWM2	PA5	TIM2 (0x40000000) CH1	/sys/class/pwm/pwmchip0
PWM3	PB7	TIM4 (0x40002000) CH2	/sys/class/pwm/pwmchip4
PWM4	PH10	TIM5 (0x40003000) CH1	/sys/class/pwm/pwmchip8
PWM5	PI5	TIM8 (0x44001000) CH1	/sys/class/pwm/pwmchip16
PWM6	PE6	TIM15 (0x44006000) CH2	/sys/class/pwm/pwmchip20

表 4-3-2 AMKN8802 PWM 设备对应驱动文件名称

4.4 I/O 测试

图形化测试方法

回到主页面点击 “I/O 测试” 按钮，进入 I/O 测试页面，如图下所示：

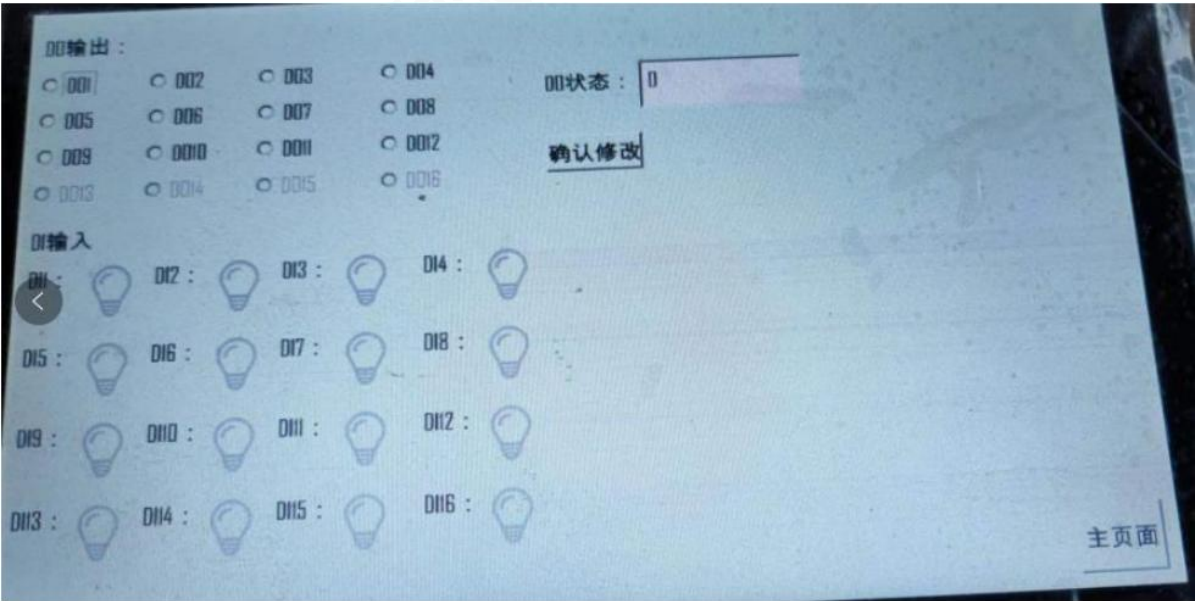


图 4.4.1 I/O 测试页面展示图

进入页面后下半部分是 DI 输入，有输入信号时对应的图标会变为红色。举例当 DI5 有输入信号，则界面如图所示：

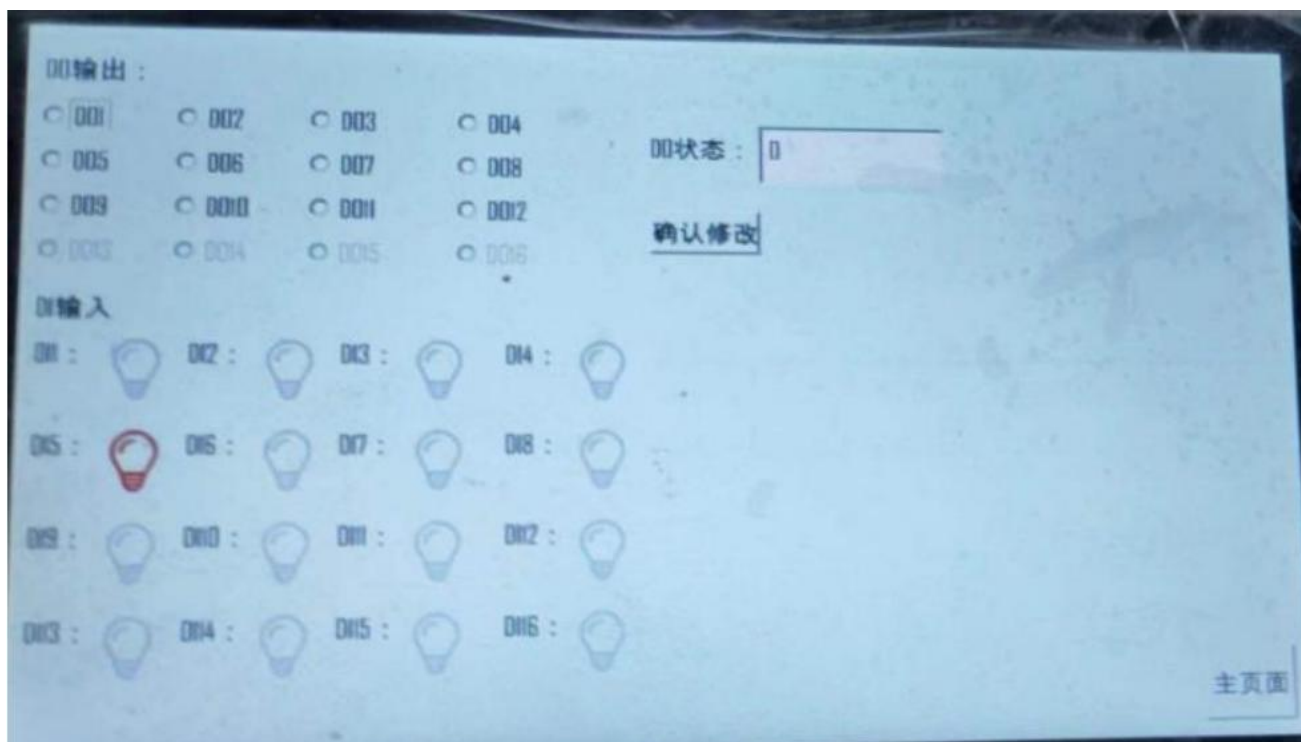


图 4.4.2 DI5 输入信号界面展示

举例当需要 DO1、DO2、DO3 输出时，点击 DO1-DO3 前面的原点之后，点击“确认修改”。当 DO 状态后面的数据改变后表示成功。

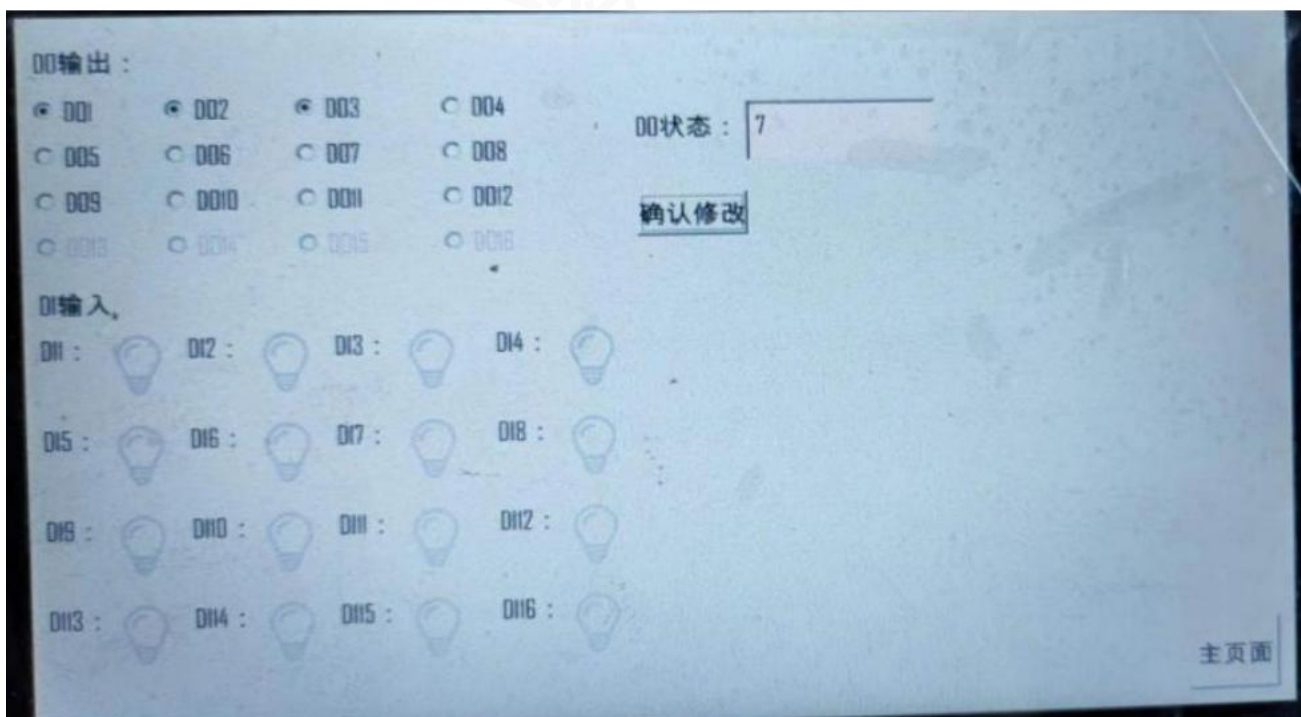


图 4.4.3 输出界面展示

非图形化测试方法

DI 测试使用命令 `inapp`，读取的是一个 16 进制的字符串数据。

```
[root@embedarm-stm32mp1]:/dev$ inapp /dev/embedarm_in
read:ffff
[root@embedarm-stm32mp1]:/dev$
```

图 4.4.4 DI 命令测试

DO 测试使用命令 `outapp`，最后一个选项请填入一个 16 进制数据，例如要让 DI1、DI2、DI3 同时输出最后一个选项就填入 7。DI1、DI2 同时输出就填入 3。

```
[root@embedarm-stm32mp1]:/dev$ outapp /dev/embedarm_out 7
read:7
[root@embedarm-stm32mp1]:/dev$
```

图 4.4.5 DO 命令测试

4.5 CAN 测试

图形化测试方法

回到主页面点击“CAN 测试”按钮，进入 CAN 通信测试，如图下图所示：

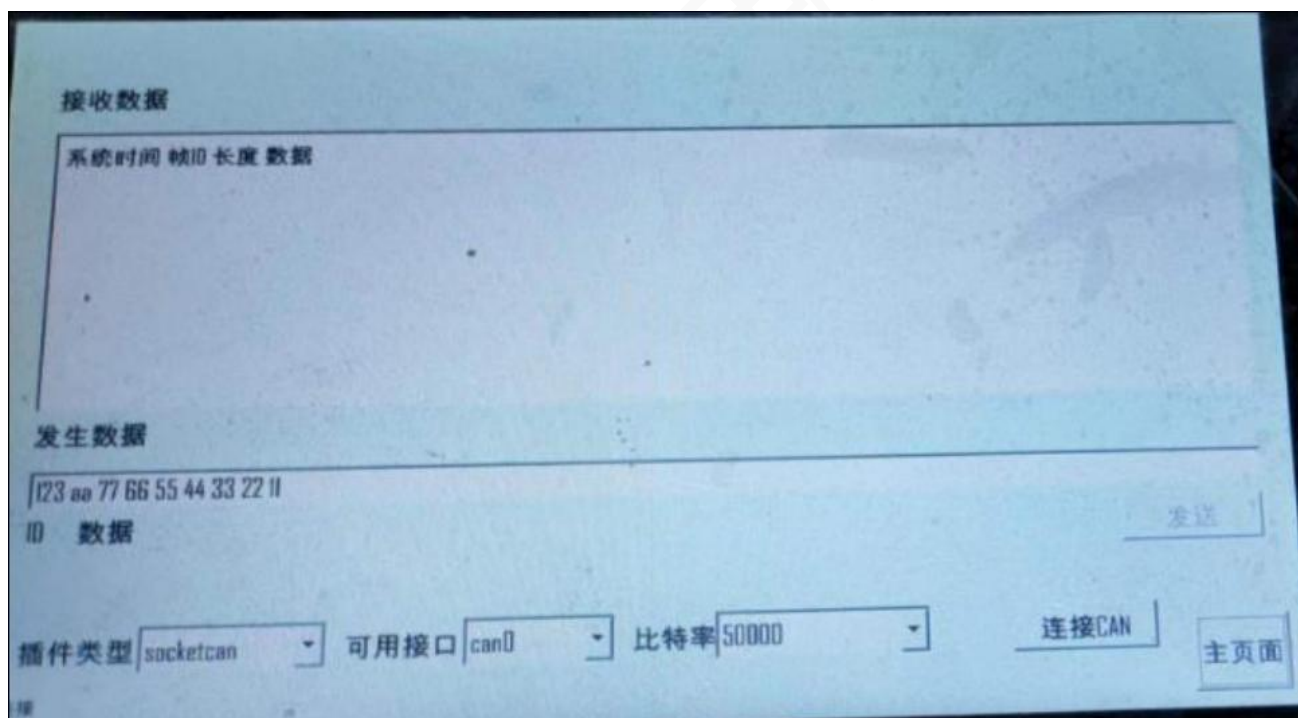


图 4.4.6 CAN 测试页面展示图

选择插件类型为“socketcan”，选择测试接口、比特率然后直接点击“连接 CAN”按钮。在电脑上打开 USB Can 测试软件准备接受数据。



图 4.4.7 USB CAN 接受数据

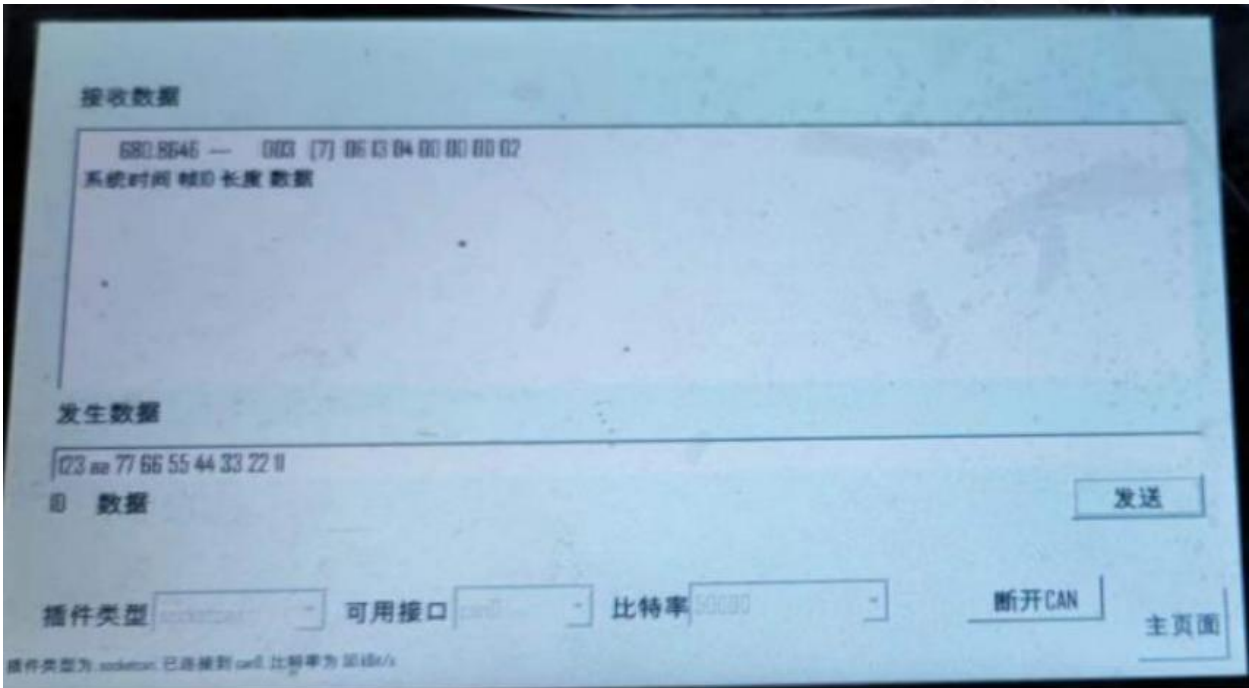


图 4.4.8 CAN 测试界面接受数据

非图形化测试方法

默认数据比特率为 50K，使用 `can_readapp1`、`can_readapp2` 命令获取 CAN1 和 CAN2 接受的数据，使用 `can_writeapp1`、`can_writeapp2` 命令发送指定数据

```
[root@embedarm-stm32mp1]:/dev$ can_readapp1
标准帧 <0x003> [7] 06 13 04 00 00 00 02
标准帧 <0x003> [7] 06 13 04 00 00 00 02
标准帧 <0x003> [7] 06 13 04 00 00 00 03
```

图 4.4.9 使用测试命令测试 CAN1 接受数据

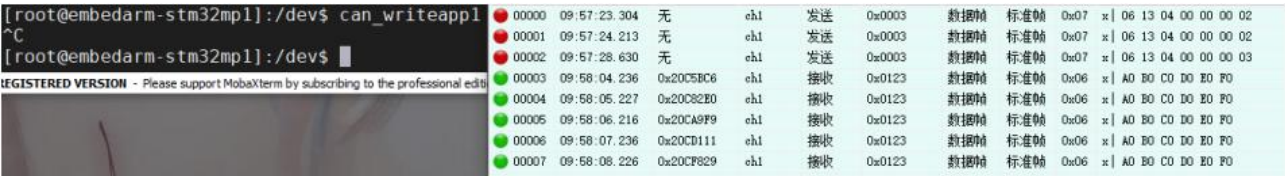


图 4.4.10 使用测试命令测试 CAN1 发送数据

4.6 ADC 测试

非图形化测试方法

使用 `adcapp` 命令读取 AI1-AI4 数据，AI 输入电压范围 0-10V。举例读取端口 AI2 输入电压数值。设备对应表如下所示：

表 4.4.11 adc 设备与端口对应表

设备名称	端口
in_voltage0_raw	AI1
in_voltage1_raw	AI2
in_voltage16_raw	AI3
in_voltage2_raw	AI4

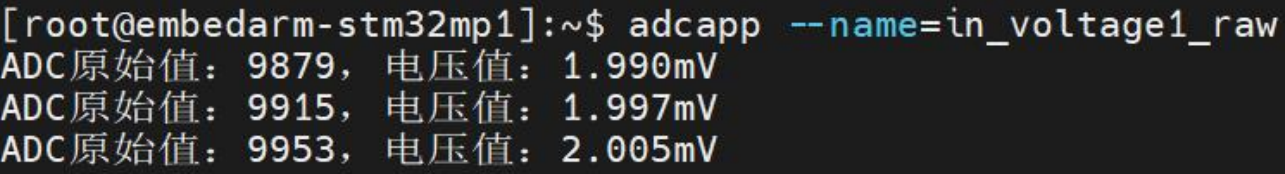


图 4.4.12 使用读取 AI2 输入电压数值

注：上述为 AMKN8801 的测试过程，AMKN8802 的 AD 是通过 SPI 挂载到 M4 内核上的

4.7 DAC 测试

非图形化测试方法

使用 `dacapp` 命令设置输出电压值，A0 输出范围 0-10V。举例设置 AI1 输出 8V。设备对应表如下所示：

设备名称	端口
iio:device1	A01
iio:device2	A02

表 4.4.13 AMKN8801 dac 设备与端口对应表

```
[root@embedarm-stm32mp1]:~$ dacapp -h
Usage: dacapp

必填项目:
--id=1          指定需要驱动的DAC驱动设备,例如--id=1或--id=2
--val=8000      设置填入电压 单位:mV,例如--Val=8000

可选项目:
--help或-help或-h  查看本程序使用帮助信息
[root@embedarm-stm32mp1]:~$ dacapp --id=1 --val=8000
```

图 4.4.14 dacapp 帮助打印和设置 AI1 端口输出 8V 电压



图 4.4.15 万用表测量 AI1 端口电压

4.8 WIFI 测试

首先可以使用下面的命令查看一下 wifi 设备的驱动是否正常挂载，图中红色方框选中的设备 wlan0 就是 wifi 网卡。

```
ifconfig -a
```

```
[root@embedarm-stm32mp1]:~$ ifconfig -a
can0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:48

can1      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING NOARP  MTU:16  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:10
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:50

eth0      Link encap:Ethernet  HWaddr B8:AE:1D:01:00:00
          inet addr:192.168.1.63  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:16441 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11400 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:21914510 (20.8 MiB)  TX bytes:1475712 (1.4 MiB)
          Interrupt:46 Base address:0x4000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

wlan0     Link encap:Ethernet  HWaddr BC:FD:0C:BC:12:AC
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

图 4.8.1 WIFI 网卡

本公司的自带镜像中有关于 wifi 的调试工具，首先使用如下命令打开 wifi 网卡：

```
ifconfig wlan0 up
```

打开网卡后使用 `iwlist` 命令扫描网络（请接天线），输入如下命令：

```
iwlist wlan0 scanning
```

```

[root@embedarm-stm32mp1]:~$ iwlist wlan0 scanning
wlan0 Scan completed :
Cell 01 - Address: E4:DA:DF:06:E7:A8
ESSID:"ChinaNet-uMpt"
Protocol:IEEE 802.11bgn
Mode:Master
Frequency:2.412 GHz (Channel 1)
Encryption key:on
Bit Rates:144 Mb/s
Extra:wpa_ie=dd1a0050f20101000050f20202000050f2020050f20401000050f202
IE: WPA Version 1
Group Cipher : TKIP
Pairwise Ciphers (2) : TKIP CCMP
Authentication Suites (1) : PSK
Extra:
IE: IEEE 802.11i/WPA2 Version 1
Group Cipher : TKIP
Pairwise Ciphers (2) : TKIP CCMP
Authentication Suites (1) : PSK
Quality=0/100 Signal level=18/100
Extra:fm=0003
Cell 02 - Address: 82:B8:4B:34:FF:4E
ESSID:"CSTX"
Protocol:IEEE 802.11bgn
Mode:Master
Frequency:2.412 GHz (Channel 1)
Encryption key:on
Bit Rates:144 Mb/s
Extra:rsn_ie=30140100000fac040100000fac040100000fac020c00
IE: IEEE 802.11i/WPA2 Version 1
Group Cipher : CCMP
Pairwise Ciphers (1) : CCMP
Authentication Suites (1) : PSK
IE: Unknown: DD230050F204104A0001101044000102100800020284103C0001011049000600372A000120
Quality=0/100 Signal level=19/100
Extra:fm=0001
Cell 03 - Address: 70:C6:DD:3E:61:A0
ESSID:"ZQLY"
Protocol:IEEE 802.11bgn
Mode:Master
Frequency:2.462 GHz (Channel 11)
Encryption key:on
Bit Rates:144 Mb/s
Extra:wpa_ie=dd160050f20101000050f20401000050f20401000050f202
IE: WPA Version 1

```

图 4.8.2 wifi 扫描结果

能扫描到设备后就可以对 wifi 配置连接指定网络，在 /etc 目录下有一个名为“wpa_supplicant.conf”的配置文件（没有请手动创建）。举例我现在需要连接“ZQLY”这个 AP 上，修改后的 wpa_supplicant.conf 文件如下：

```

ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1
#ap_scan=1

network={
    ssid="ZQLY"           // wifi 名称
    psk="xxxxxx"         // 密码
}

```

wpa_supplicant.conf 文件编写好以后需要创建 “/var/run/wpa_supplicant” 目录，命令如下

```
mkdir /var/run/wpa_supplicant -p
```

准备就绪后使用 wpa_supplicant 命令连接网络，命令如下：

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

```
[root@embedarm-stm32mp1]:~$ wpa_supplicant -Dnl80211 -c /etc/wpa_supplicant.conf -i wlan0 &
[root@embedarm-stm32mp1]:~$ Successfully initialized wpa_supplicant
wlan0: Trying to associate with c4:c0:63:f0:b1:5e (SSID='ZQLY' freq=2462 MHz)
wlan0: Associated with c4:c0:63:f0:b1:5e
wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan0: CTRL-EVENT-REGDOM-CHANGE init=COUNTRY_IE type=COUNTRY alpha2=CN
wlan0: WPA: Key negotiation completed with c4:c0:63:f0:b1:5e [PTK=CCMP GTK=CCMP]
wlan0: CTRL-EVENT-CONNECTED - Connection to c4:c0:63:f0:b1:5e completed [id=0 id_str=]
```

图 4.8.3 wifi 连接成功

连接成功后使用 udhcpc 命令从路由器申请 IP 地址

```
udhcpc -i wlan0
```

```
[root@embedarm-stm32mp1]:~$ udhcpc -i wlan0
udhcpc: started, v1.31.1
udhcpc: sending discover
udhcpc: sending select for 192.168.1.240
udhcpc: lease of 192.168.1.240 obtained, lease time 86400
deleting routers
adding dns 192.168.1.1
```

图 4.8.4 申请 IP 地址