

IDO-QM3568J-V1A 主板接口使用说明

1、固件烧录

- 1.1 安装驱动
- 1.2 打开烧录工具
- 1.3 烧录完整固件
- 1.4 升级boot分区

2、调试

- 2.1 串口调试

3、硬件接口说明

- 3.1 RS485
- 3.2 CAN
- 3.3 12V弱电输出
- 3.4 4G
 - 3.4.1 通信串口
 - 3.4.2 电源和复位控制
 - 3.4.3 IO
- 3.5 PLC
 - 3.5.1 串口通信
 - 3.5.2 复位
- 3.6 蓝牙
 - 3.6.1 串口通信
- 3.7 液晶
 - 3.7.1 SPI通信
 - 3.7.2 复位控制
 - 3.7.3 EN控制
- 3.8 以太网
- 3.9 充电
 - 3.9.1 charge-en控制
- 3.10 按键

3.11 掉电检测

3.11.1 POWER_CHECK输入电平获取

3.12 看门狗

3.12.1 开关控制

3.12.2 喂狗控制

3.13 RN7326

3.14 LED

3.14.1 LED控制

3.15 蜂鸣器

3.15.1 开关控制



IDO-QM3568J-V1A

主板接口使用说明

深圳触觉智能科技有限公司

www.industio.cn

文档修订历史

版本	修订内容	修订	审核	日期
V1.0	创建文档	谭文学		2022/03/31
V1.1	增加【LED】和【蜂鸣器】说明；	谭文学		2022/04/08

1、固件烧录

1.1 安装驱动

windows解压DriverAssitant_v5.11.zip，进入解压目录，双击运行DriverInstall.exe：

名称	修改日期	类型	大小
ADBDriver	2020/11/10 14:13	文件夹	
bin	2020/11/10 14:14	文件夹	
Driver	2022/2/28 14:14	文件夹	
Log	2022/9/7 11:13	文件夹	
config.ini	2014/6/3 15:38	配置设置	1 KB
DriverInstall.exe	2022/2/28 14:11	应用程序	491 KB
Readme.txt	2018/1/31 17:44	文本文档	1 KB
revison.log	2022/2/28 14:14	文本文档	1 KB

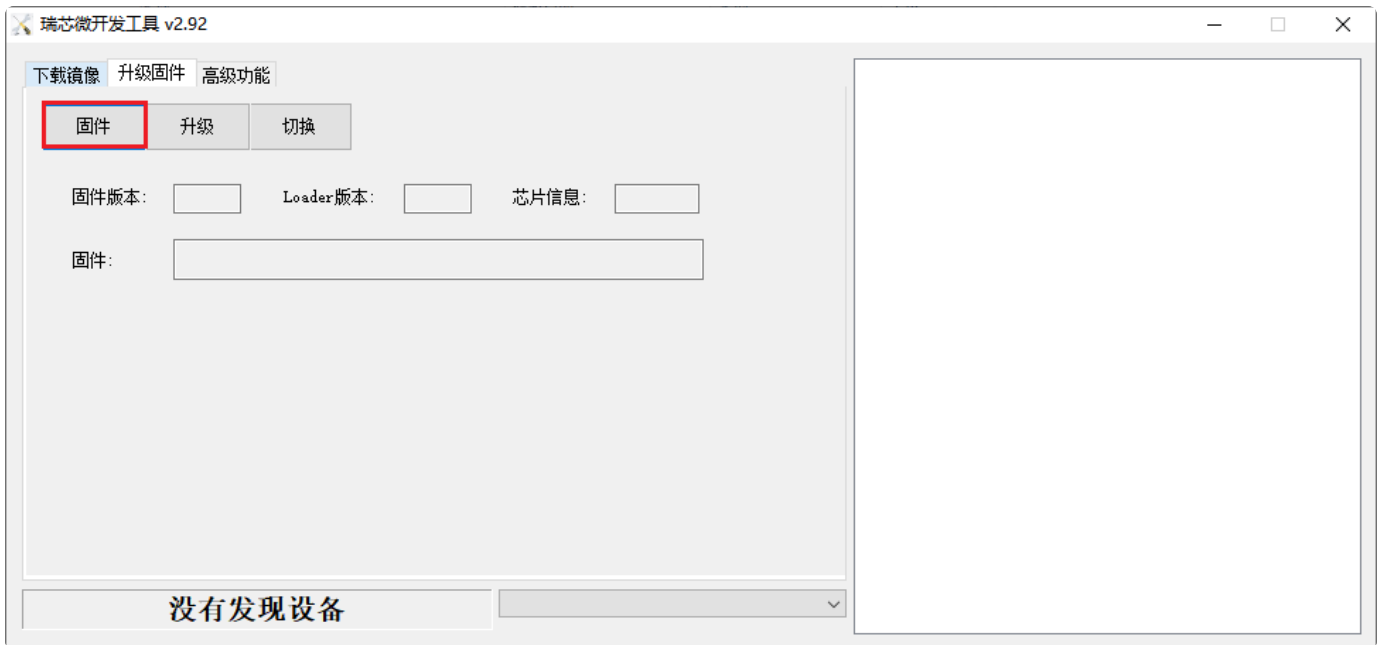
1.2 打开烧录工具

解压RKDevTool_Release_v2.92.zip，并进入解压目录，双击运行RKDevTool.exe：

名称	修改日期	类型	大小
bin	2023/1/11 14:13	文件夹	
Language	2023/1/11 14:13	文件夹	
Log	2023/1/11 14:14	文件夹	
config.cfg	2023/1/11 14:16	CFG 文件	7 KB
config.ini	2021/11/30 11:04	配置设置	2 KB
config_bk.cfg	2022/3/23 9:11	CFG 文件	7 KB
revision.txt	2022/5/19 14:14	文本文档	3 KB
RKDevTool.exe	2022/5/19 14:13	应用程序	1,204 KB
spinor-emmc-burn.cfg	2023/1/7 10:27	CFG 文件	7 KB
开发工具使用文档_v1.0.pdf	2021/8/27 10:28	WPS PDF 文档	450 KB

1.3 烧录完整固件

进入【升级固件】页面，点击【固件】按钮，选择要烧录的固件update.img：



短接烧录触点，使用USB线连接主板和电脑，当烧录软件提示“发现一个MASKROM设备”时，松开烧录触点，然后点击【执行】按钮，开始烧录。

1.4 升级boot分区

进入【下载镜像】页面，Loader分区选择固件rockdev/MiniLoaderAll.bin，地址0x00000000；boot分区选择固件rockdev/boot.img，地址0x00008000。

短接烧录触点，使用USB线连接主板和电脑，当烧录软件提示“发现一个MASKROM设备”时，松开烧录触点，然后点击【执行】按钮，开始烧录。

2、调试

2.1 串口调试

调试串口位于JP4，电平类型为RS232电平。通信参数为1500000 8 N 1。

3、硬件接口说明

3.1 RS485

主板共配置2路RS485接口，位于J1端口的27-30引脚：

13	14	...	27	28	29	30
...	A2	B2	A1	B1
				/dev/ttyS6		/dev/ttyS5

3.2 CAN

主板共配置一路CAN接口，位于端口J1的17和18引脚：

13	14	...	17	18	...	29	30
...	CAN_H	CAN_L
				can0			

3.3 12V弱电输出

12V输出接口位于J1的19-20引脚：

13	14	...	19	20	...	29	30
...	12V	GND

开启12V输出

```

▼  Bash | 复制代码
1 echo 255 > /sys/class/leds/switch-12v-out/brightness

```

关闭12V输出

```

▼  Bash | 复制代码
1 echo 0 > /sys/class/leds/switch-12v-out/brightness

```

3.4 4G

3.4.1 通信串口

串口通信节点为/dev/ttyS9。

3.4.2 电源和复位控制

电源控制

Shell | [复制代码](#)

```
1 //开启电源
2 echo 255 > /sys/class/leds/lte-pwr/brightness
3
4 //关闭电源
5 echo 0 > /sys/class/leds/lte-pwr/brightness
```

复位控制

Shell | [复制代码](#)

```
1 //开启复位
2 echo 255 > /sys/class/leds/lte-rst/brightness
3
4 //关闭复位
5 echo 0 > /sys/class/leds/lte-rst/brightness
```

onoff控制

Shell | [复制代码](#)

```
1 //拉高onoff
2 echo 255 > /sys/class/leds/lte-onoff/brightness
3
4 //拉低onoff
5 echo 0 > /sys/class/leds/lte-onoff/brightness
```

3.4.3 IO

IO1_4G、IO2_4G和IO3_4G的gpio编号见下表:

引脚	gpio number
IO1_4G	73
IO2_4G	19

IO3_4G	122
--------	-----

以上三个gpio均可通过gpio class控制，以IO1_4G为例：

```
Shell | 复制代码
1 echo 73 > /sys/class/gpio/export
2 echo out > /sys/class/gpio/gpio73/direction
3 //输出高电平
4 echo 1 > /sys/class/gpio/gpio73/value
5 //输出低电平
6 echo 0 > /sys/class/gpio/gpio73/value
```

3.5 PLC

3.5.1 串口通信

对应的设备节点为/dev/ttyS1。

3.5.2 复位

```
Shell | 复制代码
1 //拉高复位
2 echo 255 > /sys/class/leds/lte-rst/brightness
3
4 //拉低复位
5 echo 0 > /sys/class/leds/lte-rst/brightness
```

3.6 蓝牙

3.6.1 串口通信

对应的设备节点为/dev/ttyS7。

3.7 液晶

3.7.1 SPI通信

对应的设备节点为/dev/spidev1.0。

3.7.2 复位控制

```
▼ Shell | 复制代码
1 //拉高复位
2 echo 255 > /sys/class/leds/lcd-rst/brightness
3
4 //拉低复位
5 echo 0 > /sys/class/leds/lcd-rst/brightness
```

3.7.3 EN控制

```
▼ Shell | 复制代码
1 //拉高EN
2 echo 255 > /sys/class/leds/lcd_en/brightness
3
4 //拉低EN
5 echo 0 > /sys/class/leds/lcd_en/brightness
```

3.8 以太网

主板共配置2个100M以太网，系统中对应的网络节点分别为eth0（PHY2）和eth1（PHY1）。

3.9 充电

3.9.1 charge-en控制

```

1 //拉高charge1-en
2 echo 255 > /sys/class/leds/charge1-en/brightness
3 //拉低charge1-en
4 echo 0 > /sys/class/leds/charge1-en/brightness
5
6 //拉高charge2-en
7 echo 255 > /sys/class/leds/charge2-en/brightness
8 //拉低charge2-en
9 echo 0 > /sys/class/leds/charge2-en/brightness

```

3.10 按键

主板共配置6个按键，各个按键对应的键值如下表：

名称	键值
KEY_UP	KEY_UP
KEY_DOWN	KEY_DOWN
KEY_LEFT	KEY_LEFT
KEY_RIGHT	KEY_RIGHT
KEY_ENTER	KEY_ENTER
KEY_ESC	KEY_ESC

对应的设备节点为/dev/input/event2。

3.11 掉电检测

3.11.1 POWER_CHECK输入电平获取

掉电检测引脚POWER_CHECK的输入电平可通过以下方法获取：

$$V_{in} = (\text{voltage}_{\text{ex_raw}}/1024)*1.8\text{v}$$

其中voltage_{ex_raw}通过读取设备节点获取：

```
1 cat /sys/bus/iio/devices/iio:device0/in_voltage4_raw
```

假如返回580, 则POWER_CHECK的输入电压 $V = (580/1024)*1.8v=1.02v$

3.12 看门狗

3.12.1 开关控制

```
1 //开启看门狗
2 echo 255 > /sys/class/leds/wd-en/brightness
3 //关闭看门狗
4 echo 0 > /sys/class/leds/wd-en/brightness
```

3.12.2 喂狗控制

```
1 //打开喂狗
2 echo "heartbeat" > /sys/class/leds/wd-en/trigger
3 //关闭喂狗
4 echo "none" > /sys/class/leds/wd-en/trigger
```

3.13 RN7326

主控通过spi接口与RN7326通信, 设备节点为/dev/spidev2.0。

3.14 LED

主板共配置2个LED, SYS_LED和ALARM_LED。SYS_LED默认作为系统指示灯, 系统正常运行时, 会以心跳的形式闪烁。

3.14.1 LED控制

```
1 //打开ALARM_LED
2 echo 255 > /sys/class/leds/alarm-led/brightness
3 //关闭ALARM_LED
4 echo 0 > /sys/class/leds/alarm-led/brightness
```

3.15 蜂鸣器

主板配置了1个有源蜂鸣器。

3.15.1 开关控制

```
1 //打开蜂鸣器
2 echo 255 > /sys/class/leds/bz-en/brightness
3 //关闭蜂鸣器
4 echo 0 > /sys/class/leds/bz-en/brightness
```