

QT251N 系列用户手册

版本号：V2.0

武汉舜通智能科技有限公司

版权所有

版本修订

| 版本号 | 修订日期 | 描述 | 审核 |
|------|----------|------|----|
| V1.0 | 20160907 | 创建文档 | |
| V1.1 | 20171218 | 修改文档 | |
| V2.0 | 20180103 | 修改文档 | |

特别说明

本公司保留在未通知用户的情况下，对产品、文档、服务等内容进行修改、更正等其他一切变更权利。

目录

| | |
|--------------------------|--------|
| 一、产品概述..... | - 3 - |
| 二、产品特性..... | - 3 - |
| 1. 硬件特性..... | - 3 - |
| 2. 软件特性..... | - 5 - |
| 三、接口定义..... | - 6 - |
| 1. 电源接口..... | - 6 - |
| 2. RS485 接口..... | - 6 - |
| 3. RS232、CAN 接口..... | - 7 - |
| 4. 开关量输出输入..... | - 7 - |
| 5. 网络接口..... | - 8 - |
| 6. 调试接口..... | - 9 - |
| 7. RS485/RS232 驱动接口..... | - 9 - |
| 四、驱动实例..... | - 10 - |

一、产品概述

QT251N 是一款基于精简指令集（RISC）架构高性能的 32 位 MPU 的嵌入式计算机。该 CPU 是以 ARM Cortex-A8 为核心的系统级单芯片，内置 NEON 单指令流多数流（SIMD）协处理，带有错误校正码（ECC）的 256KB L2 缓存，最高支持 1GHz 的频率。系统提供 RS458/RS232 通讯，有线网络通讯，CAN 总线，同时也提供可选的无线 GPRS 通讯，具有体积小、功耗低、效率高等特点，适用于电力集中器、HMI、工业控制、网关等场合。

二、产品特性

1. 硬件特性

- AM355x CPU:
 - 32bit ARM Cortex-A8 架构，主频 800MHz，1.6MIPS/MHz，最高主频 1GHz
 - 32KB I-cache，32KB D-cache，NeonSIMD 协处理器
- 内存：
 - 512Mbyte DDR3、64KB 专用 RAM
- FLASH：
 - 512Mbyte NANDFlash，最大支持 8Gbyte
 - 支持 NAND、NOR、SRAM 等 FLASH
- 加密：
 - 支持 PRNG/DES/3DES/AES/SHA/HMAC 加密，最高 256 位加密模式
- 看门狗：
 - 内置 WDT，溢出时间小于 60 秒，支持空闲唤醒和掉电唤醒
- RTC：
 - 高精度实时时钟，内置供电电池
- 调试口：
 - 1 路串口为系统 console 口。波特率：115200，数据位：8，停止位：1，校验位：none，流控：无
- RS485/RS232：
 - 8 路独立 RS485 通讯，内部全隔离保护设计
 - 8 路 RS485/RS232 分时复用通讯，可根据实际选择使用，内部全隔离保护设计

- B 码对时：
 - 1 路 RS485 接收，专用于 B 码对时，连接后自动对时
- CAN：
 - 1 路 CAN 通讯，内置隔离保护设计
- 开关量输出输入：
 - 2 路双刀双掷继电器输出
 - 2 路开关量输入
- 网络：
 - 4 路 10M/100M 自适应工业以太网，标准 RJ45 接口
 - 15KV TVS 保护，内部全隔离保护设计
- 无线功能（可选）：
 - 射频波段 800/900/1800/1900MHz（可选 2/3/4G）
 - 可选 WIFI：可连接 AP，也可做 AP
 - 1 个 SIM 卡接口，1 个天线接口
 - 传输速度：达到相应功能的标准速度
- SD CARD：
 - 内置一个 SD/MMC 卡接口
- 电源：
 - 输入电压：220V，支持交流、直流
 - 单机功耗：< 12W
- 机械特性
 - 外壳金属材质
 - 尺寸：1U
 - 防护等级：IP63
- 工作环境
 - 工作温度：-40℃~+85℃
 - 工作湿度：5%~95%

2. 软件特性

2.1 系统特性

QT251N 预装基于 TI AM335x 的 Linux 操作系统, 版本为 3.2.0。满足 POSIX 标准或类 UNIX 平台的应用程序。针对系统特有的硬件设备, 内核提供了简单、易用的驱动接口, 可加速用户的应用程序开发。

QT251N 系统的软件系统共分为 3 部分, 分别为 Bootloader、linux 内核和 rootfs。Bootloader 是遵循 GPL 条款的开放源码项目, UBoot 主要是引导内核的启动, 支持 NFS 挂载、NAND Flash 启动; linux 内核是整个操作系统的最底层, 负责整个硬件的驱动, 以及提供各种系统所需的核心功能; rootfs 是用于明确磁盘或分区上的文件的方法和数据结构, 即在磁盘上组织文件的方法。

2.2 环境配置

本公司提供的虚拟机系统 ubuntu 10.04, 可直接编译使用。

用户名: work

密码 : 123456

编译命令: arm-linux-gnueabi-gcc -o filename filename.c

编译链: 本公司提供的 arm-linux-gnueabi-4.7.tar.gz

非本公司提供的编译环境下, 把编译链拷贝到 PC 的 LINUX 系统下, 解压编译链后, 把根目录下的 bin 目录添加到系统的环境变量即可。

如解压到/opt/arm-linux-gnu 目录下, 则添加环境变量为:

```
export PATH=$PATH:/opt/arm-linux-gnu/bin
```

编译链下载地址:

<https://pan.baidu.com/s/1nv19D1Z#list/path=%2FWORK%2FCortex-A8>

2.3 管理机登录

IP: eth0: 192.168.1.177

eth1: 192.168.2.177

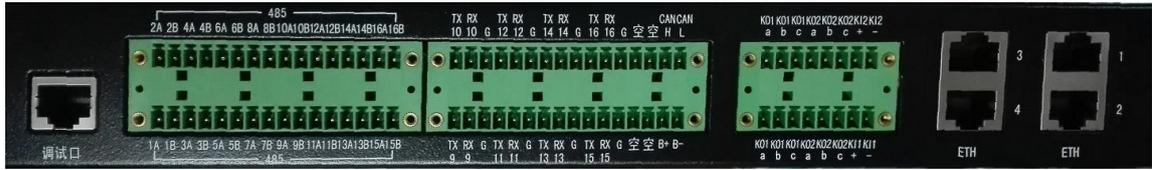
eth2: 192.168.3.177

eth3: 192.168.4.177

用户名: root

密码 : root

三、接口定义

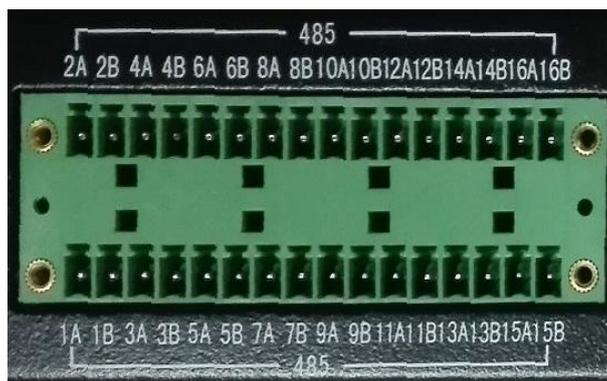


1. 电源接口



| 编号 | 标识符 | 功能说明 |
|----|---|--------------|
| 1 | AC/DC220V | 电源接口，支持交流/直流 |
| 2 |  | 保护地（非零线 N） |

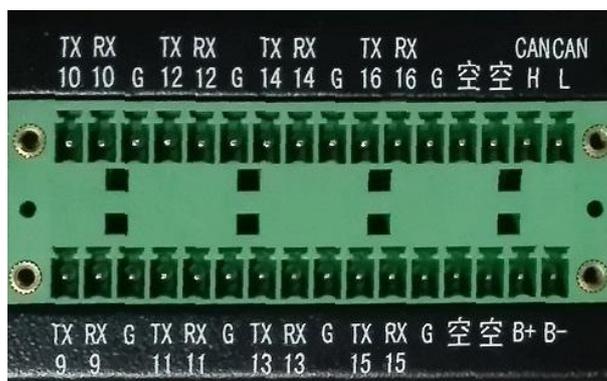
2. RS485 接口



| 编号 | 标识符 | 功能说明 |
|----|-----|----------------------------|
| 1 | nA | 第 n 通道 RS485 端口 A (n=1~16) |
| 2 | nB | 第 n 通道 RS485 端口 B (n=1~16) |

注：RS485 第 9~16 通道与 RS232 的第 9~16 通道为复用通道。

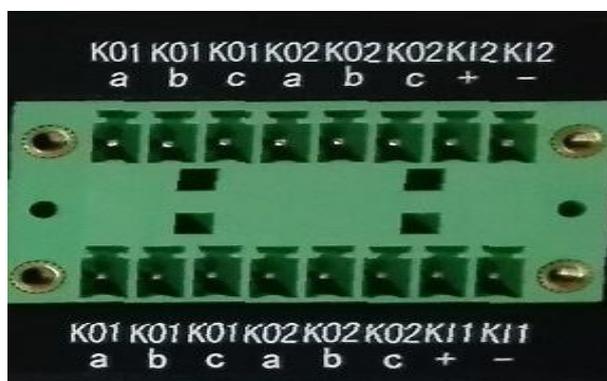
3. RS232、CAN 接口



| 编号 | 标识符 | 功能说明 |
|----|-----------------|-----------------------------|
| 1 | TX _n | 第 n 通道 RS232 端口 TX (n=9~16) |
| 2 | RX _n | 第 n 通道 RS232 端口 RX (n=9~16) |
| 3 | G | GND, 通讯地 |
| 4 | 空 | NC, 无连接 |
| 5 | CAN:H | CAN 通讯 H 端 |
| 6 | CAN:L | CAN 通讯 L 端 |
| 7 | B+ | B 码对时 RS485-A 端 |
| 8 | B- | B 码对时 RS485-B 端 |

注：RS232 第 9~16 通道与 RS485 的第 9~16 通道为复用通道。对应的驱动接口相同，为分时复用通道。

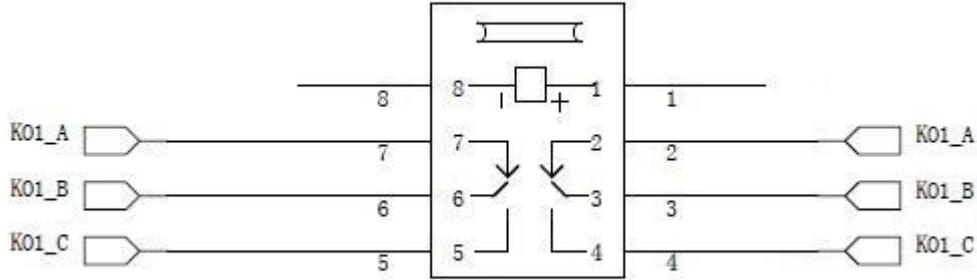
4. 开关量输出输入



| 编号 | 标识符 | 功能说明 |
|----|-----|------|
|----|-----|------|

| | | |
|---|-------|-----------------------|
| 1 | KOn:a | 第 n 路继电器输出常闭端 (n=1,2) |
| 2 | KOn:b | 第 n 路继电器输出公共端 (n=1,2) |
| 3 | KOn:c | 第 n 路继电器输出常开端 (n=1,2) |
| 4 | KIn:+ | 第 n 路开关量输入正端 (n=1,2) |
| 5 | KIn:- | 第 n 路开关量输入负端 (n=1,2) |

注：继电器输出为双刀双掷开关，如下图：



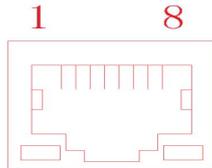
5. 网络接口



| 网口编号 | 编号 | 标识符 | 功能说明 |
|------------|----|----------|----------------|
| ETH1/2/3/4 | 1 | E0/1_TX+ | 以太网 ETH0/1_TX+ |
| | 2 | E0/1_TX- | 以太网 ETH0/1_TX- |
| | 3 | E0/1_RX+ | 以太网 ETH0/1_RX+ |
| | 4 | NC | 未使用 |
| | 5 | NC | 未使用 |
| | 6 | E0/1_RX- | 以太网 ETH0/1_RX- |
| | 7 | NC | 未使用 |
| | 8 | NC | 未使用 |

| | | | |
|----|---|------|---------------|
| IP | 1 | Eth1 | 192.168.1.177 |
| | 2 | Eth2 | 192.168.2.177 |
| | 3 | Eth3 | 192.168.3.177 |
| | 4 | Eth4 | 192.168.4.177 |

6. 调试接口



| 编号 | 标识符 | 功能说明 |
|-----|-----|---------------|
| 1 | TX | RS232 调试串口 TX |
| 2 | RX | RS232 调试串口 RX |
| 3 | GND | 系统通讯地 |
| 4-8 | NC | 未使用 |

调试口配置：波特率：115200，数据位：8，停止位：1，校验位：none，流控：无

7. RS485/RS232 驱动接口

```

/dev/ttyCH0 /dev/ttyCH12 /dev/ttyCH2 /dev/ttyCH6
/dev/ttyCH1 /dev/ttyCH13 /dev/ttyCH3 /dev/ttyCH7
/dev/ttyCH10 /dev/ttyCH14 /dev/ttyCH4 /dev/ttyCH8
/dev/ttyCH11 /dev/ttyCH15 /dev/ttyCH5 /dev/ttyCH9
    
```

```

/dev/ttyS1 /dev/ttyS12 /dev/ttyS15 /dev/ttyS2 /dev/ttyS5 /dev/ttyS8
/dev/ttyS10 /dev/ttyS13 /dev/ttyS16 /dev/ttyS3 /dev/ttyS6 /dev/ttyS9
/dev/ttyS11 /dev/ttyS14 /dev/ttyS17 /dev/ttyS4 /dev/ttyS7
    
```

驱动接口可以在管理机的/dev 目录下查看。

| 编号 | 标识符 | 功能说明 |
|----|--------|--|
| 1 | ttySn | 对应 ttyCH(n-1)，第 n 通道 RS485 驱动接口 (n=1~8) |
| 2 | ttySn | 对应 ttyCH(n-1)，第 n 通道 RS485/RS232 驱动接口 (n=9~16) |
| 3 | TtyS17 | GPRS 通讯串口 (2G 模块才会使用) |

四、驱动实例

在系统的/program 目录下有相应的脚本文件，可以进行一些简单的测试。其中要确保 startup.sh 文件里，端口映射的正确的。文件内容见附录。

附录：

1、startup.sh 文件内容：

```
#!/bin/sh

ln -sf /dev/ttyCH0 /dev/ttyS1
ln -sf /dev/ttyCH1 /dev/ttyS2
ln -sf /dev/ttyCH2 /dev/ttyS3
ln -sf /dev/ttyCH3 /dev/ttyS4
ln -sf /dev/ttyCH4 /dev/ttyS5
ln -sf /dev/ttyCH5 /dev/ttyS6
ln -sf /dev/ttyCH6 /dev/ttyS7
ln -sf /dev/ttyCH7 /dev/ttyS8
ln -sf /dev/ttyCH8 /dev/ttyS9
ln -sf /dev/ttyCH9 /dev/ttyS10
ln -sf /dev/ttyCH10 /dev/ttyS11
ln -sf /dev/ttyCH11 /dev/ttyS12
ln -sf /dev/ttyCH12 /dev/ttyS13
ln -sf /dev/ttyCH13 /dev/ttyS14
ln -sf /dev/ttyCH14 /dev/ttyS15
ln -sf /dev/ttyCH15 /dev/ttyS16
ln -sf /dev/ttyO1 /dev/ttyS17

ip link set can0 type can bitrate 100000

ifconfig can0 up
```

2、serial.c 文件内容：

```
#include <stdio.h>
#include <string.h>
```

```
#include <malloc.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <termios.h>

#define max_buffer_size    100    /* buffer size */
/*****/

int fd1;
int flag_close;

int open_serial(int k,int *fd)
{
    int sfd = -1;
    char str[100];
    sprintf(str,"/dev/ttyS%d",k);
    printf("open %s\n",str);

    sfd = open(str,O_RDWR|O_NOCTTY| O_NONBLOCK);
    if(sfd == -1){
        perror(str);
        return -1;
    }
    else{
        *fd = sfd;
        return 0;
    }
}

/*****/

int main(int argc, char *argv[ ] )
```

```
{
    time_t tNow,tOld;
    int port;
    char
sbuf[]={"12345678901234567890123456789012345678901234567890\n"}; /* 固定
发送的数据 */
    char sbufrec[256]={0};
    int sfd,retv,i,ncount=0,mcount = 0;
    struct termios opt;
    int length=sizeof(sbuf);
    /*****
**/
    if(argc < 2)
    {
        printf("input erro :serial <1~4> \n");
        return 0;
    }
    port = atoi(argv[1]);
    open_serial(port,&fd1);
    /*****
**/
    printf("ready for sending data...\n");
    tcgetattr(fd1,&opt);
    cfmakeraw(&opt);
    /*****
/
    cfsetispeed(&opt,B9600); /*设置波特率为 9600bps*/
    cfsetospeed(&opt,B9600);
    /*****
**/
    tcsetattr(fd1,TCSANOW,&opt);
```

```
while(mcount < 5)
{
    retv=write(fd1,sbuf,length); /* 发送数据 */
    if(retv==-1){
        //perror("write");
        printf("write error .....\\n");
    }
    else{
        printf("the number of char sent is %d\\n",retv);
    }
    ncount=0;
    printf("ready for receiving data...\\n");

    time(&tOld);
    tNow=tOld;
    ncount = 0;
    while(((tNow-tOld) < 2)) /* 设置接收超时 */
    {
        time(&tNow);
        retv=read(fd2,&sbufrec[0],1);
        if(retv==-1){
            //perror("read");
            //printf("error read \\n");
            //printf("tOld=%d;tNow=%d\\n",tOld,tNow);
        }
        else{
            printf("%02x ",sbufrec[0]);
            ncount+=1;
        }
    }
}
```

```
        mcount+=1;
        printf("\n");
    }
    flag_close = close(fd1);
    if(flag_close == -1) /*关闭口端口*/
        printf("Close the Device1 failur!\n");
    return 0;
}
```

QTouch 环境部署

- 1、telnet 登陆设备：用户名：root 密码：root
- 2、进入根目录：cd /
- 3、Tftp 下载 home_QT251N.tar.gz：tftp -r home_QT251N.tar.gz -g 主机 IP
- 4、解压文件：tar -zxvf home_QT251N.tar.gz
- 5、设置启动项，可以直接下载 startup.sh 文件到/program 目录：首先进入 program 目录（cd /program）tftp -r starup.sh -g 主机 IP
- 6、下载授权设备序列号：首先进入 etc 目录(cd /etc)tftp -r name -g 主机 IP
- 7、重启采集器 reboot
- 8、再次进入查看 QTouch 启动状况，如果出现下图所示，即为启动成功

```

Telnet 192.168.1.177
Mem: 14092K used, 500640K free, 0K shrd, 0K buff, 5416K cached
CPU:  2.1% usr  2.3% sys  0.0% nic 95.4% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.14 0.05 0.02 1/46 828
  PID  PPID  USER  STAT  VSZ  %MEM  CPU  %CPU  COMMAND
  814   1  root   S     16932  3.2   0   2.1  /home/ctstor/ctrtdb
  819   1  root   S     8448  1.6   0   1.1  /home/ctstor/ctfiles/modbusRTU
  182   2  root   SW    0  0.0   0   0.2  [sync_supers]
  809   1  root   S     8268  1.6   0   0.0  /home/ctstor/ctserver
  781   1  root   S     3140  0.6   0   0.0  /usr/local/bin/sshd
  822  784  root   S     2628  0.5   0   0.0  telnetd
  784   1  root   S     2564  0.5   0   0.0  /usr/sbin/inetd
  823  822  root   S     2340  0.4   0   0.0  -sh
  828  823  root   R     2340  0.4   0   0.0  top
  810   1  root   S     2340  0.4   0   0.0  /sbin/getty 115200 tty00
    1   0  root   S     2336  0.4   0   0.0  init
  787   1  root   S     2336  0.4   0   0.0  /sbin/syslogd
  334   2  root   SW    0  0.0   0   0.0  [kworker/0:1]
    5   2  root   SW    0  0.0   0   0.0  [kworker/u:0]
  468   2  root   SW    0  0.0   0   0.0  [ubi_bgt0d]
    3   2  root   SW    0  0.0   0   0.0  [ksoftirqd/0]
    2   0  root   SW    0  0.0   0   0.0  [kthreadd]
    6   2  root   SW<   0  0.0   0   0.0  [khelper]
    7   2  root   SW<   0  0.0   0   0.0  [netns]
    8   2  root   SW    0  0.0   0   0.0  [kworker/u:1]
  184   2  root   SW    0  0.0   0   0.0  [bdi-default]
  186   2  root   SW<   0  0.0   0   0.0  [kblockd]
  194   2  root   SW<   0  0.0   0   0.0  [omap2_mcspi]
  206   2  root   SW    0  0.0   0   0.0  [khubd]
  325   2  root   SW<   0  0.0   0   0.0  [musb-hdrc.0]
  330   2  root   SW<   0  0.0   0   0.0  [musb-hdrc.1]

```