异常处理单元说明

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 更新日期 | 更新内容 | 操作者 |
| V1 | 2016/8/13 | 创建本文档 | 陈济斌 |
|  |  |  |  |

目录

[1 打印机硬件状态分类 1](#_Toc458884151)

[1.1 打印机硬件状态分类以及分类依据 1](#_Toc458884152)

[1.1.1 打印机硬件异常情况分类 1](#_Toc458884153)

[1.1.2 打印机硬件状态分类依据 1](#_Toc458884154)

[1.2 打印机长期状态保存 3](#_Toc458884155)

[1.2.1 概要 3](#_Toc458884156)

[1.3 威加打印机协议的函数封装 5](#_Toc458884157)

[1.3.1 概要 5](#_Toc458884158)

[1.3.2 代码文件位置 9](#_Toc458884159)

[1.4 底层硬件配置（串口，DMA相关） 10](#_Toc458884160)

[1.4.1 概要 10](#_Toc458884161)

[1.4.2 代码文件位置 10](#_Toc458884162)

# 打印机硬件状态分类

## 打印机硬件状态分类以及分类依据

### 打印机硬件异常情况分类

硬件状态情况分类如下所示：

#define KNIFE\_ERROR\_STATE 0x01 //切刀错误

#define UNEXPECTED\_OPENED\_STATE 0x02 //打印机头被打开

#define PAPER\_INSUFFICIENT\_STATE 0x03 //缺纸状态

#define FILL\_IN\_PAPER\_STATE 0x04 //进纸状态

#define PAPER\_WILL\_BE\_IN\_INSUFFICIENT\_STATE 0x05 //纸将尽状态

#define NORMAL\_STATE 0x09 //正常状态

代码位于[USER/data\_form.h](../../USER/data_form.h)

### 打印机硬件状态分类依据

上述状态是基于打印机的通信协议向打印机发送请求打印机硬件状态，然后将打印机返回的一系列返回数据进行组合，从而总结出了几种符合现实情况的打印机状态。具体打印机状态返回码组合和打印机处于的硬件状态的组合入下表所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 硬件状态类型 | 打印机状态(n=1) | 脱机状态（n=2） | 错误状态（n=3） | 检纸传感器状态（n=4） |
| 切刀错误 | 00001000  （脱机） | 01000000  （有错误发生） | 00101000  （发生自动切纸错误和出现不可恢复性错误） | 00001100  （纸将尽传感器，纸不足,缺纸传感器显示有纸） |
| 打印机头被打开 | 00001000  （脱机） | 01000100  （有错误发生且机头已经打开） | 00100000  （出现不可恢复错误） | 00001100  （纸将尽传感器，纸不足,缺纸传感器显示有纸） |
| 进纸状态 | 00001000  （脱机） | 01001000  （有错误发生，通过进纸键进纸） | 00101000  （发生自动切纸错误，出现不可恢复错误） | 00001100  （纸将尽传感器，纸不足,缺纸传感器显示有纸） |
| 缺纸状态 | 00001000  （脱机） | 00100000  （打印纸用完） | 00000000  (正常状态) | 01101100  （纸将尽传感器：纸不足和缺纸传感器：无纸） |
| 不放打印纸，打开机头 | 00001000  （脱机） | 01100100（有错误发生，打印机纸用完，机头已打开） | 00100000 | 01101100  （纸将尽传感器：纸不足和缺纸传感器：无纸） |
| 打印时打印机头抬起 | 00001000  （脱机） | 01000100  （有错误发生且机头已经打开） | 00100000  （出现不可以恢复错误） | 00001100  （纸将尽传感器，纸不足,缺纸传感器显示有纸） |
| 正常状态 | 00000000  （正常状态） | 00000000  （正常状态） | 00000000  （正常状态） | 00001100  （纸将尽传感器，纸不足,缺纸传感器显示有纸） |

## 打印机长期状态保存

### 概要

打印机需要离线保存一些长期数据，以便进行实时的打印机硬件模块的健康状态的评估。保存的长期状态被放在指定的EEPROM的指定内存位置，以便后期的维护。

//EEPRROM 存储段的分配（为了方便读写，都用4字节表示）

//设备1的数据空间 （0x00 - 0x1c）

#define DEVICE\_ONE\_START\_ADDR 0x00

#define TOTAL\_WORKING\_TIME\_ADDRESS\_DEVICE\_ONE 0x00 //总打印时间，4字节的长度, 单位：分钟

#define TOTAL\_KNIFE\_CUT\_TIMES\_ADDRESS\_DEVICE\_ONE 0x04 //总切刀次数，4字节的长度，单位：次数

#define TOTAL\_PRINTED\_LENGTH\_DEVICE\_ONE 0x08 //总打印长度，4字节的长度，单位：厘米

//记录设备1各异常类型的发生次数的地址

#define KNIFE\_ERROR\_TIMES\_ADDRESS\_DEVICE\_ONE 0x0c //切刀错误次数，4字节的长度, 单位：次数

#define UNEXPECTED\_OPENED\_TIMES\_ADDRESS\_DEVICE\_ONE 0x10 //机头被打开次数，4字节的长度, 单位：次数

#define PAPER\_INSUFFICIENT\_TIMES\_ADDRESS\_DEVICE\_ONE 0x14 //缺纸次数，4字节的长度, 单位：次数

代码位于[USER/data\_form.h](../../USER/data_form.h)

#define FULL\_IN\_PAPER\_TIMES\_ADDRESS\_DEVICE\_ONE 0x18 //进纸次数，4字节的长度, 单位：次数

//#define HIGH\_TEMPERATURE\_TIMES\_ADDRESS\_DEVICE\_ONE 0x1c //温度过高次数，4字节的长度, 单位：次数

//设备2的数据空间 (0x32 - 0x4e)

#define DEVICE\_TWO\_START\_ADDR 0x32

#define TOTAL\_WORKING\_TIME\_ADDRESS\_DEVICE\_TWO 0x32 //总打印时间，4字节的长度, 单位：分钟

#define TOTAL\_KNIFE\_CUT\_TIMES\_ADDRESS\_DEVICE\_TWO 0x36 //总切刀次数，4字节的长度，单位：次数

#define TOTAL\_PRINTED\_LENGTH\_DEVICE\_TWO 0x3a //总打印长度，4字节的长度，单位：厘米

//记录设备2各异常类型的发生次数的地址

#define KNIFE\_ERROR\_TIMES\_ADDRESS\_DEVICE\_TWO 0x3e //切刀错误次数，4字节的长度, 单位：次数

#define UNEXPECTED\_OPENED\_TIMES\_ADDRESS\_DEVICE\_TWO 0x42 //机头被打开次数，4字节的长度, 单位：次数

#define PAPER\_INSUFFICIENT\_TIMES\_ADDRESS\_DEVICE\_TWO 0x46 //缺纸次数，4字节的长度, 单位：次数

#define FULL\_IN\_PAPER\_TIMES\_ADDRESS\_DEVICE\_TWO 0x4a //进纸次数，4字节的长度, 单位：次数

//#define HIGH\_TEMPERATURE\_TIMES\_ADDRESS\_DEVICE\_TWO 0x4e //温度过高次数，4字节的长度, 单位：次数

## 威加打印机协议的函数封装

### 概要

要让上位机与威加打印机通信需要遵循厂家的相应的通信协议，但是由于通信协议比较散乱，而且使用比较麻烦，所以需要对一系列打印机命令进行组合封装，封装成我们人为需要的工作逻辑。以下是现阶段已经封装的打印机协议。

//API package

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* printQRCode

\*

\* Description: This function is used to printQRCode

\*

\* Arguments : data - the first address of the dataBuff(usually dataBuff must be an address of an URL)

\* size - the size of the data

\*

\* Returns : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printQRCode(u8\_t \*data, u16\_t size, u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* mySendData

\*

\* Description: This function is used to command corrsponding port to Send out data

\*

\* Arguments : dataBuff - the first address of the dataBuff

\* size - the size of the dataBuff

\* deviceNum - portSelect[deviceNum] (the set of usart you gonna use to ouuput data)

\*

\* Returns : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void mySendData(u8\_t \*data, u16\_t size, u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* printImages

\*

\* Description: This function is used to print images

\*

\* Arguments : dataBuff - the first address of the dataBuff

\* size - the size of the dataBuff

\*

\* Returns : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printImages(u8\_t\* data, u16\_t size, u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* printImages\_Alternate

\*

\* Description: This function is used to printImages

\*

\* Arguments : data - the first address of the dataBuff

\* size - the size of the data

\* length - numbers of dots from horizontal direction

\* width - numbers of dots from vertical direction

\*

\* Returns : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printImages\_Alternate(u8\_t \*data, u16\_t size, u16\_t length, u16\_t width, u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* outputData

\*

\* Description: This function is used to Output the data from dataBuff one byte by one byte

\*

\* Arguments : dataBuff - the first address of the dataBuff

\* size - the size of the dataBuff

\*

\* Returns : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char outputData(u8\_t\* data, u16\_t size, u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* cutPaper

\*

\* Description: This function is used to Send paper-cutting command to the printer through USART1

\*

\* Arguments : deviceNum

\*

\* Returns : none

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void cutPaper(u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* getPrinterState

\*

\* Description: This function is used to examine the Real-Time @ General state @ of the printer

\*

\* Arguments : deviceNum

\*

\* Returns : the corresponding state of the printer

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

unsigned char getPrinterState(u8\_t deviceNum);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function Name : ReadEEPROM

\* Description : Reads a block of data from the EEPROM.

\* Input : - data : pointer to the buffer that receives the data read

\* from the EEPROM.

\* - ReadAddr : EEPROM's internal address to read from.

\* - NumByteToRead : number of bytes to read from the EEPROM.

\* Output : None

\* Return : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ReadEEPROM(u8\_t \*data, u8\_t ReadAddr, u8\_t NumByteToWrite);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function Name : ClearEEPROM

\* Description : Clear the first Page of the EEPROM

\* Input : None

\* Output : None

\* Return : None

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void ClearEEPROM(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function Name : specialAdditon

\* Description : To do Addtion operation for a 3-bytes-long data area

\* Input : - mem : pointer to the buffer containing the data to be

\* operated addition operation

\* - numsToAdd : numbers gonna gonna be added

\* Output : None

\* Return : None

\* Note : the API has passed the test in Code::Blocks Platform

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

static void specialAdditon(unsigned char\* mem, unsigned char numsToAdd);

### 代码文件位置

以上接口位于[printerException.h](../../USER/status_mesg.h)。

## 底层硬件配置（串口，DMA相关）

### 概要

为了能够加快传输的速度，对串口的发送波特率调至115200，这样可以加快打印机的打印速度；为了让主控芯片在传输数据的时候CPU不被占用，开启DMA后台传送功能。

底层实现API如下：

### 代码文件位置

代码位于[stm32\_init.c](../../USER/pack_data.c)。