多应用日志库AppLog

编程指南

V1.0.0

惠尔丰电子（北京）有限公司

2010年9月4日

修订历史记录

**A**-添加，**M**-修改，**D**-删除

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本** | **日期** | **AMD** | **修订者** | **说明** |
| 1.0 | 2010-9-4 | A | 于海涛 | 创建 |
|  |  |  |  |  |
|  |  |  |  |  |

[第一章 前言 4](#_Toc272053989)

[第二章 动手之前的要知道的 5](#_Toc272053990)

[1. 了解一下多应用日志库AppLog实现哪些功能和特点？ 5](#_Toc272053991)

[第三章 函数接口API 6](#_Toc272053992)

[1. 标准日志输出接口 6](#_Toc272053993)

[2. 16进制HEX数据输出接口 6](#_Toc272053994)

[第四章 使用方法和示例 8](#_Toc272053995)

[1. 日志输出开关和指向 8](#_Toc272053996)

[2. 日志过滤 8](#_Toc272053997)

[3. 日志过滤示例 8](#_Toc272053998)

[4. 标准日志示例 9](#_Toc272053999)

[5. 16进制HEX日志输出示例 10](#_Toc272054000)

# 前言

此文档是指导开发人员在Vx系列的POS上如何使用多应用日志库调试应用程序的指南。

此文档的读者需要具备以下能力：

* 对POS系统有所了解
* 对多应用基本概念有所了解
* Vx平台开发和调试经验

如果您开发过VX的程序并使用过LOG\_PRINTF将对深入了解此文档有很大的帮助。

如果使用多应用日志库调试应用程序，您至少还需要：

* 一台带串口的电脑
* 一台Vx系列POS机
* 串口工具软件（如debugview）

# 动手之前的要知道的

### 了解一下多应用日志库AppLog实现哪些功能和特点？

* 通过串口输出跟踪日志：

通过简单的调用ApplLog\_Debug函数，就实现了输出应用日志。

* 通过串口输出HEX日志：

通过调用ApplLog\_DebugHex，实现了HEX码的日志输出。

* 查看库运行日志：

部分库（ApplMngr, AppTools）也采用了AppLog输出日志。

* 日志开关功能：

通过在GROUP15中设置#LOGPORT与否，决定日志是否被输出。

* 日志无延时特性：

ApplLog库输出日志通过ApplLog进程进行，对应用执行来说基本没有延时。

* 日志输出信息丰富：

ApplLog库输出日志包括了应用名、模块名、时间（tick）、源文件名、行号和应用指定输出信息。

* 日志过滤：

通过GROUP15设置#LOGAPP的参数不同，可以输出指定的应用或库的日志。

* Rlease版本和Debug版本的切换：

如果机器中没有下载AppLog的进程，则日志不起作用，应用执行也不会受到影响。如果需要对应用或库进行跟踪调试，则下载AppLog进程到机器中，即可输出日志。

# 函数接口API

您也可以参考《AppLog.h》中的API定义。

### 标准日志输出接口

/\* --------------------------------------------------------------------------

\* FUNCTION NAME: ApplLog\_Debug

\* DESCRIPTION: 普通日志输出

\* PARAMETERS

\* ModuleName - (Input) 指定的模块名称，用于日志过滤，需大写

fmt, args... - (Input) 可变参数的输出数据

\* RETURN: none

\* NOTES: none.

\* ------------------------------------------------------------------------ \*/

ApplLog\_Debug(ModuleName,fmt, args...)

**说明：**ApplLog\_Debug是一个可变参数的函数接口定义，支持标准C语言的变量类型。

* ApplLog\_Debug虽然是个宏定义，但是调用时，其后要加“；”;
* 可变参数的使用方式类似sprintf。
* 模块名称(ModuleName)可以按照需要自定义，作为日志过滤输出的条件(大写)。

### 16进制HEX数据输出接口

/\* --------------------------------------------------------------------------

\* FUNCTION NAME: ApplLog\_DebugHex

\* DESCRIPTION: 16进制数据输出

\* PARAMETERS

\* ModuleName - (Input) 指定的模块名称，用于日志过滤，需大写

title - (Input) 数据标题

LogBuf - (Input) 16进制数据

Loglen - (Input) 16进制数据的字节数

\* RETURN: none

\* NOTES: none.

\* ------------------------------------------------------------------------ \*/

ApplLog\_DebugHex(ModuleName,title,LogBuf,Loglen)

**说明：**

* ApplLog\_DebugHex虽然是个宏定义，但是调用时，其后要加“；”
* 当Loglen小于等于0时，则只会看到标题输出；
* 模块名称(ModuleName)可以按照需要自定义，作为日志过滤输出的条件(大写)。

# 使用方法和示例

### 日志输出开关和指向

通过在GROUP15中设置#LOGPORT系统变量来控制日志的输出与否及输出串口。

**说明：**

* 如果需要关闭日志输出，有2种方法；
  + 清除GROUP15中的AppLog.out;
  + 清空GROUP15中的#LOGPORT；
* 修改日志输出的设备（目前只支持串口）；

设置#LOGPORT=COM1，则日志输出到COM1口；

根据机型不同，可用于日志输出的串口有：

* + Vx510/520: COM1、COM2（PINPAD口）
  + Vx670/680: COM1（CABLE）、COM6（BASE）

### 日志过滤

通过在GROUP15中设置#LOGAPP系统变量来控制日志的过滤输出。

**说明：**

* 实现日志输出过滤的前提是允许日志输出，输出方法见上述描述；
* 日志可按照应用名称和模块名称分别进行过滤，用“/”分割。如：

#LOGAPP=APPNAME/MODULENAME

* 日志过滤同时支持多个应用名称和模块名称的组合，用“，”分割；如:

#LOGAPP=APPNAME1/MODULENAME1,APPNAME2/MODULENAME2

* #LOGAPP不设置或设置为“\*”，则输出所有日志；
* 日志过滤支持”\*”通配符模式；如需要输出所有应用的ATOOL库日志，则：

#LOGAPP=\*/ATOOL

### 日志过滤示例

假设机器中装用MAINAPP、CCBEMV、DEMOA 3个应用，都使用了ATOOL、APPLMNGR、EMVTOOL库：

示例1：只输出CCBEMV应用的ATOOL库日志：

#LOGAPP=CCBEMV/ATOOL

示例2：输出CCBEMV应用的ATOOL库和AppLMngr库日志：

#LOGAPP= CCBEMV/ATOOL,CCBEMV/APPLMNGR

示例3：输出CCBEMV应用的所有日志（含其使用的库日志）：

#LOGAPP=CCBEMV 或

#LOGAPP=CCBEMV/\*

示例4：输出CCBEMV应用的ATOOL库日志和MAINAPP应用的AppLMngr库日志：

#LOGAPP=CCBEMV/ATOOL,MAINAPP/ APPLMNGR

示例5：输出所有应用的AppLMngr库日志：

#LOGAPP=\*/APPLMNGR 或

#LOGAPP=APPLMNGR

示例6：输出所有应用的AppLMngr库和ATOOL库日志：

#LOGAPP=\*/APPLMNGR,\*/ATOOL 或

#LOGAPP=APPLMNGR,\*/ATOOL 或

#LOGAPP=\*/APPLMNGR, ATOOL 或

#LOGAPP=APPLMNGR,ATOOL

示例7：输出MAINAPP和DEMOA 2个应用的所有日志（含其使用的库日志）：

#LOGAPP=MAINAPP,DEMOA

示例8：输出所有日志：

#LOGAPP=\* 或

不设置#LOGAPP

### 标准日志示例

如在ApplMngr库中AppPipe.c文件第353行增加：

ApplLog\_Debug("APPLMNGR","[%s]:Read Pipe Msg [Ret=%d][MsgCmd=%d][FromTaskID=%d][FromAppName=%s][FrmPipe=%d]....",szCurrAppName,rlen,MsgCmd,MsgFromTaskID,MsgFromAppName,MsgFromPipeHandle);

并在CCBEMV应用中使用ApplMngr库；

则在串口工具软件中能看到如下日志信息：

A:CCBEMV/APPLMNGR|T:19546|F:AppPipe.c|L:353:[CCBEMV]:Read Pipe Msg [Ret=0][MsgCmd=7][FromTaskID=3][FromAppName=MAINAPP][FrmPipe=258]....

其中 “|”是分隔符：

A:表示应用信息；

CCBEMV是应用名称；

APPLMNGR是模块名称；

T:19546表示从机器开机到执行此日志的时刻是19456（tick，1 tick约等于1 msec）；

F:AppPipe.c 和 L:353 表示在此日志在AppPipe.c文件中的第353行；

后续则是自定义的输出信息；

### 16进制HEX日志输出示例

如在CCBEMV应用中debug.c文件第37行增加：

ApplLog\_DebugHex("CCBEMV",”Send”, MessageBuf, MessageBufLen);

则在串口工具软件中能看到如下日志信息：

A:CCBEMV/CCBEMV|T:37757|F:debug.c|L:37:Send[LEN=97]

60 00 03 00 00 60 01 00 03 02 01 08 00 00 20 00 | `...`[. ][.. .

00 00 C0 00 12 00 00 02 31 31 31 31 31 31 31 31 | ..?..] 11111111

31 32 33 34 35 36 37 38 39 31 32 33 34 35 33 00 | 12345678 9123453.

11 00 00 00 01 00 30 00 40 30 31 20 30 33 30 31 | ...[.0. @01 0301

30 30 30 30 30 33 30 31 31 30 31 30 30 31 20 20 | 00000301 101001

20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 |

01 | [