



# OK系列智能摄像头使用手册

Version 2012.11

**北京嘉恒中自图像技术有限公司**  
Beijing JoinHope Image Technology Ltd.

## 注意事项

- 1、不要在超出摄像头要求的温度条件和湿度条件下使用摄像头，保持摄像头干燥，禁止淋雨或放在潮湿的地方。室外应用场合，摄像头必须安装在摄像头防护罩中。如果被摄像头淋雨或沾湿，应关闭系统电源，交给生产厂家进行维修。
  - 2、摄像头安装在室外高处时，为避免遭遇雷击而导致摄像头损坏，摄像头外壳必须通过防护罩良好接地。
  - 3、请勿使用不合格电源给摄像头供电，摄像头供电电压应为 12 伏（最大不能超过 15 伏），输出电流应不小于 1.2 安。为避免引入电源干扰，电源不要与其他设备共用。
  - 4、请勿将摄像头长时间的对准强光（例如太阳光），这样可能会导致 CCD 传感器过度饱和造成永久性损坏。
  - 5、请勿擅自拆卸、修理和组装摄像头，有任何问题请联系北京嘉恒中自图像技术有限公司技术支持。
  - 6、请勿使用手指或者坚硬物体接触 CCD 的保护玻璃。
  - 7、摄像头不使用时，应将镜头取下，盖上摄像头盖放好。
- 由以上原因造成的摄像头损坏将不在保修范围之内。

## 配件信息

名称	数量
OK智能摄像头	1
12V电源	1
电源线	1
网线	1
开发扩展板（LINK板）	1
《OK系列智能摄像头使用手册》	1

## 版本记录

文档版本	修改说明	发布日期	相关部门
V1.0	文档建立	2012. 11. 02	研发/技术/市场

# 前 言

北京嘉恒中自图像技术有限公司是国内领先的数字图像产品供应商，总部位于中关村中科院自动化研究所，是一家聚集了大批业内技术精英，以自主研发为核心竞争力的股份制高新技术企业。我们的前身是中科院自动化所图像部及后来成立的科技嘉仪器仪表有限公司。公司研发骨干主要来自中科院研究所和重点高校，具有扎实的技术实力，丰富的产品开发经验和良好的用户服务信誉。

嘉恒图像是国内最早的专业图像卡生产商，也是国内为数不多的能够自主研发各种高性能 CCD 和 CMOS 摄像头产品及 DSP, FPGA 图像处理和采集产品的公司之一。目前，我们的主要产品系列有图像采集卡、工业摄像头、嵌入式专用图像采集处理器及基于 DSP 技术的图像采集处理产品等，广范应用于医学影像，生物技术，工业检测，智能交通，保安监控，金融票证，动态分析等领域。我们根据客户的应用需求，提供各种普及档、中档和高档的图像产品，同时提供强大的技术支持和研发定制服务。

了解更多嘉恒图像 OK 系列摄像头的详细信息及使用指南，请阅读本手册，也可登陆嘉恒网站 [www.jhi.com.cn](http://www.jhi.com.cn) 或致电 010-51665596 进行咨询。



# 目 录

前言	
<b>第1章 产品概述</b>	<b>01</b>
1.1 产品性能参数	02
1.2 硬件资源	03
1.3 内嵌算法和功能	03
1.4 开发环境	04
<b>第2章 硬件介绍</b>	<b>03</b>
2.1 硬件系统方框图	03
2.2 外部接口描述	06
2.2.1 以太网接口	06
2.2.2 综合接口	06
2.2.3 36芯数字输入输出接口	07
2.2.4 开发扩展板 (LINK板) 接口	10
<b>第3章 OK系列智能摄像头安装与使用说明</b>	<b>17</b>
3.1 OK系列智能摄像头的安装与固定	17
3.1.1 固定摄像头	17
3.1.2 镜头连接	19
3.1.3 调节后截距	19
3.1.4 连接电源和网线	19
3.2 OK系列智能摄像头上位机软件操作指南	20
3.2.1 打开摄像头	20
3.2.2 采集和停止	20
3.2.3 采集压缩图像	21
3.2.4 设置参数	22
3.2.5 设置查找表	22
3.2.6 自动白平衡	23
3.2.7 设置IP地址	24
3.2.8 同步实时时钟	25
3.2.9 软件/硬件外触发	25
<b>第4章 OK系列智能摄像头嵌入式编程指南</b>	<b>27</b>
4.1 OK_IM系列智能摄像头嵌入式编程指南	27
4.1.1 OK_IM系列智能摄像头DEMO软件简介	27
4.1.2 OK_IM系列智能摄像头提供的存储器资源	30
4.1.3 DEMO工程存储器使用分配	31
4.1.4 OK_IM系列智能摄像头嵌入式编程API函数说明	32
4.1.5 DSP程序固化说明	51
4.1.6 DSP程序远程更新说明	53
4.1.7 用户需了解的其他内容	55
4.2 OK_IC系列智能摄像头嵌入式编程指南	58
4.2.1 OK_IC系列智能摄像头DEMO软件介绍	58
4.2.2 OK_IC系列智能摄像头提供的存储器资源	61
4.2.3 DEMO工程存储器使用分配	62
4.2.4 OK_IC系列智能摄像头嵌入式编程API函数说明	63
4.2.5 DSP程序固化说明	81
4.2.6 DSP程序远程更新说明	83
4.2.7 用户需了解的其他内容	86
<b>第5章 OK系列智能摄像头上位机应用程序编程指南</b>	<b>89</b>
5.1 接口参数结构定义说明	89
5.1.1 图像设备信息结构	89
5.1.2 网络设备参数结构	90
5.1.3 图像块信息结构	90
5.1.4 序列文件信息结构	91
5.1.5 MJPG信息头结构	91
5.1.6 解压的扩展信息结构	92
5.2 DLL动态库接口函数	92
5.3 其他支持函数	104
<b>第6章 故障诊断及排除</b>	<b>107</b>
6.1 故障现象：没有图像	107
<b>第7章 获得支持</b>	<b>108</b>
<b>第8章 附录</b>	<b>109</b>
8.1 摄像头的维护	109
8.2 图表附录	109

# 第1章 产品概述

OK系列智能摄像头是嘉恒图像开发的基于CCD图像传感器和高性能DSP的全数字化高性能、高清、高帧率智能摄像头，具有高性价比、低功耗、低噪声等特点。

OK系列智能摄像头内部集成了逐行扫描CCD、高速DSP、大规模FPGA、大容量SDRAM / SD卡、10/100Mb以太网，可完成多种图像处理功能。摄像头自带的 DEMO 给用户提供了一个基于 DSP/BIOS 实时操作系统的最基本的软件开发平台，采用标准C语言编程，网络通信使用了TI公司的NDK，用户可参考DEMO建立自己的应用程序，实现图像采集和处理功能。OK系列智能摄像头通过10/100Mb以太网连接PC机，可以通过以太网远程下载程序到摄像头内部运行，快速方便的实现程序更新。

OK系列智能摄像头现有型号有：高帧率黑白智能摄像头OK\_IM1160、OK\_IM1161、200万像素彩色智能摄像头OK\_IC1200。我们为OK系列智能摄像头提供完备的技术资料以及技术支持，可以帮助用户进行二次开发。

## 1.1 产品性能参数

产品型号	OK_IM1160	OK_IM1161	OK_IC1200
产品描述	高帧率黑白智能摄像头	高帧率黑白智能摄像头	200万像素彩色智能摄像头
帧频 (fps)	61	100	15
传感器类型	40万像素黑白逐行CCD	30万像素黑白逐行CCD	200万像素彩色逐行CCD
CCD传感器尺寸	1/2inch	1/3inch	1/1.8inch
像素尺寸(μm)	8.3×8.3	7.4×7.4	4.4×4.4
有效像素数	768×576	640×480	1600×1200
AD采样精度	10bit	10bit	12bit
曝光时间	26μs~16.2ms (触发模式可达1s)	20μs~10ms (触发模式可达1s)	53μs~66ms (触发模式可达1s)
增益调节	可编程调节(0~36db)和自动增益可选	可编程调节(0~36db)和自动增益可选	可编程调节(0~36db)和自动增益可选
供电电源	12V DC	12V DC	12V DC
功耗	≤6W	≤6W	≤6W
传输方式	10/100Mb自适应以太网接口	10/100Mb自适应以太网接口	10/100Mb自适应以太网接口
支持网络协议	TCP/IP、UDP	TCP/IP、UDP	TCP/IP、UDP
支持图像格式	JPEG, 压缩因子可调	JPEG, 压缩因子可调	JPEG, 压缩因子可调
支持视频格式	M-JPEG	M-JPEG	M-JPEG
远程升级功能	有	有	有
网络自动续连功能	有	有	有
工作环境温度	-20℃~+75℃	-20℃~+75℃	-20℃~+75℃
工作环境湿度	20%~80%	20%~80%	20%~80%
镜头接口	C/CS	C/CS	C/CS
净重	约等于411克	约等于411克	约等于411克
应用领域	适用于工业检测等领域	适用于工业检测等领域	适用于电子警察、高清卡口、视频监控等领域

表格1-1 OK系列智能摄像头性能参数表

## 1.2 硬件资源

DSP处理器	TI TMS320DM642, 主频600MHz
数据存储器	128M字节SDRAM
程序存储器	8M字节串行FLASH, 2M字节并行FLASH
参数存储器	32K字节EEPROM
海量存储器	SD卡(非标配)
查找表	10bit LUT
内部硬件加密	可定制
实时时钟	高精度RTC时钟, 保证抓拍时间准确
温度监控	内嵌温度传感器, 实时提供设备温度信息
外部光照传感器	可定制
看门狗	时间可设
以太网接口	10/100Mb
串口	RS232串口1组, RS485串口1组
GPIO接口	10个可编程I/O, 5V TTL电平
外触发接口	一个
光源控制接口	一个闪光灯控制信号输出
DSP的JTAG口	有

表格1-2 OK系列智能摄像头硬件资源表

## 1.3 内嵌算法和功能

摄像头内部的高速32位DSP系统, 出厂时已经嵌入了如下图像算法和软件功能, 随摄像头提供给用户。用户可以根据现场需要, 通过嵌入式编程调用相应的API函数启用这些功能, 从而大大减少上位机的工作量。

- 1) 硬件实现色彩空间变换;
- 2) 硬件实现自动白平衡;
- 3) JPEG 压缩算法;
- 4) 用户可配置的图像数据LUT表, 实现任意函数变换;
- 5) 丰富的TCP/IP 控制命令, 实现了包括抓拍图像、自动白平衡、曝光时间设定、

设定IP地址、程序远程升级等功能；

6) 丰富的底层控制API，用户可以精确控制CCD的增益、曝光时间、可编程IO等资源。

这些算法和功能的详尽描述，请参见本手册第四章——OK系列智能摄像头嵌入式编程指南。

## 1.4 开发环境

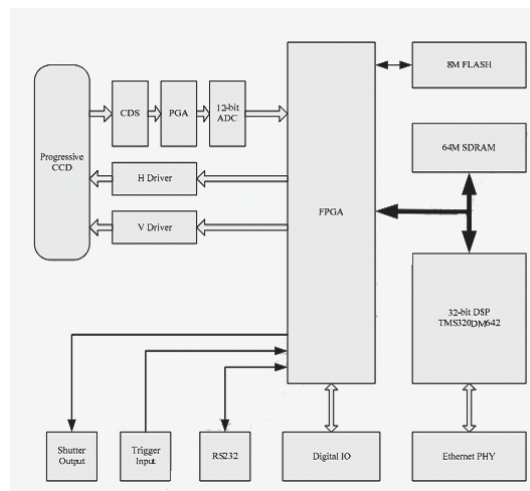
**DSP编程环境：**在TI公司的集成开发环境（IDE）CCS2.20.18下采用标准C语言编程。提供良好移植测试过的基于DSP/BIOS的实时操作系统（RTOS）软件平台源码，以及底层驱动API函数。API函数的说明及调用方法详见本手册第四章——OK系列智能摄像头嵌入式编程指南。

**PC编程环境：**标准的TCP/IP socket命令接口，兼容所有操作系统，例如Windows、Linux等。提供在Visual C++6.0下编写的DEMO程序。各种TCP/IP控制命令和编程方法详见本手册第五章——OK系列智能摄像头上位机应用程序编程指南。

# 第2章 硬件介绍

## 2.1 硬件系统方框图

OK系列智能摄像头的硬件系统方框图如下：



图表2-1 OK系列智能摄像头的硬件系统方框图

## 2.2 外部接口描述

OK系列智能摄像头为嵌入式应用提供了丰富的用户输入输出接口，用以连接和控制各种外部设备，包括：一个以太网接口、一个综合控制接口、一个36芯数字输入输出接口。摄像头后面板外部接口图如下：



图表2-2 摄像头后面板外部接口图

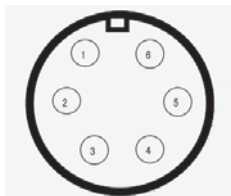
### 2.2.1 以太网接口

摄像头的网络接口为10/100Mb自适应以太网接口，用于与上位机通信，传送图像数据和各种命令。以太网接口采用RJ45标准接头，工作范围达到工业级要求，自带变压器。RJ45接头带有固定孔位，使连接更加牢固。

OK系列智能摄像头的出厂默认IP地址是192.168.42.150，出厂时已经设置MAC地址各不相同。

用户可以通过PC端控制程序任意修改IP地址和MAC地址，以符合自己系统的需要，具体请参照本手册第三章《OK系列智能摄像头安装与使用说明》。

### 2.2.2 综合接口



图表2-3 综合接口示意图

Pin1: 12V 电源正，红色

Pin2: 电源地/信号地，黑色

Pin3: 外触发输入，5V\_TTL 电平，下降沿触发，黄色

Pin4: 闪光灯输出，5V\_TTL 电平，高电平有效，白色

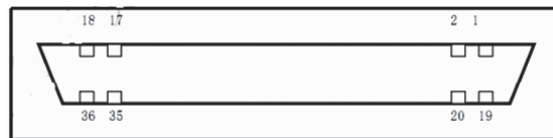
Pin5: 串口RS232, RXA, 接受端，绿色

Pin6: 串口RS232, TXA, 发送端，蓝色

在摄像头内部，对电源输入设计了过压保护、反极性保护、浪涌保护。

### 2.2.3 36芯数字输入输出接口

在智能摄像头的后面板上，有一个36芯的数字输入输出接口，如下图：



图表2-4 36芯数字输入输出接口示意图

该接口供用户通过连接开发扩展板（LINK板）来连接DSP仿真器从而实现对DSP的在线编程、调试与程序烧写，或者使用其中的各种输入输出控制端来控制外部设备。接口的每个控制引脚的定义如下：

端子序号	信号名称	信号方向	功能描述
PIN1	DSP_JTAG_EMU1	NA	DSP仿真信号
PIN2	DSP_JTAG_TCK	NA	DSP仿真信号
PIN3	DSP_JTAG_TDI	NA	DSP仿真信号
PIN4	DSP_JTAG_RTCK	NA	DSP仿真信号
PIN5	+3.3V	POWER	+3.3V输出（最大输出500mA）
PIN6	GND	POWER	电源地
PIN7	RS422R+	IN	串口RS422接收正端
PIN8	GND	POWER	电源地
PIN9	WATCHDOG_CLOSE	IN	看门狗外部禁止信号



端子序号	信号名称	信号方向	功能描述
PIN10	+5V	POWER	+5V输出（最大输出500mA）
PIN11	TX232	OUT	串口RS232发送端
PIN12	SHUTTER	OUT	闪光灯信号输出
PIN13	GPIO1	IN/OUT	通用IO1
PIN14	GPIO3	IN/OUT	通用IO3
PIN15	GPIO5	IN/OUT	通用IO5
PIN16	GPIO7	IN/OUT	通用IO7
PIN17	GPIO9	IN/OUT	通用IO9
PIN18	RS422T+	OUT	串口RS422发送正端
PIN19	DSP_JTAG_TRST	NA	DSP仿真信号
PIN20	GND	POWER	电源地
PIN21	DSP_JTAG_EMU0	NA	DSP仿真信号
PIN22	DSP_JTAG_TDO	NA	DSP仿真信号
PIN23	DSP_JTAG_TMS	NA	DSP仿真信号
PIN24	GND	POWER	电源地
PIN25	RS422R-	IN	串口RS422接收负端
PIN26	GND	POWER	电源地
PIN27	EXT_RESET	IN	外部复位信号
PIN28	GND	POWER	电源地
PIN29	RX232	IN	串口RS232接收端
PIN30	TRIGIN	IN	外触发信号输入
PIN31	GPIO2	IN/OUT	通用IO2
PIN32	GPIO4	IN/OUT	通用IO4
PIN33	GPIO6	IN/OUT	通用IO6
PIN34	GPIO8	IN/OUT	通用IO8
PIN35	GPIO10	IN/OUT	通用IO10
PIN36	RS422T-	OUT	串口RS422发送负端

表格2-1 OK系列智能摄像头36芯数字输入输出接口引脚定义

摄像头接口（标记为电源）和开发扩展板（标记为LINK板）接口连接关系定义如下：

36舌头（标记为：电源）	36舌头（标记为：LINK板）	信号
1	36	DSP_JTAG_EMU1
6	35	GND
2	34	DSP_JTAG_TCK
21	33	DSP_JTAG_EMU0
22	32	DSP_JTAG_TDO
7	31	GND
23	30	DSP_JTAG_TMS
19	29	DSP_JTAG_TRST
3	28	DSP_JTAG_TDI
10	27	+5V
11	26	UART232 TXDA
12	25	SHUTTER
13	24	IO1
14	23	IO3
15	22	IO5
16	21	IO7
17	20	IO9
18	19	UART485 A485
5	18	+3.3V
20	17	DSP_JTAG_TMS
8	16	GND
9	14	WATCH_DOG_EN
24	13	GND
25	12	GND

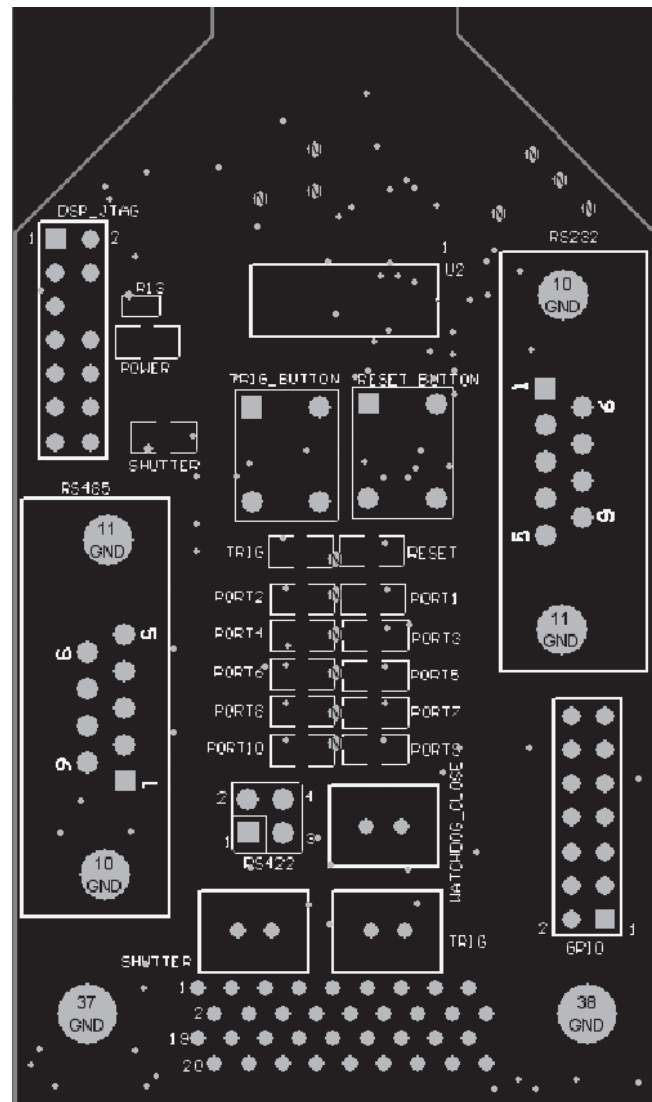
36舌头 (标记为: 电源)	36舌头 (标记为: LINK板)	信号
26	11	GND
27	10	RESET
28	9	GND
29	8	UART232 RXDA
30	7	EXT_TRIG_IN
31	6	IO2
32	5	IO4
33	4	IO6
34	3	IO8
35	2	IO10
36	1	UART485 B485

表格2-2 OK系列智能摄像头接口与开发扩展板接口连接关系

#### 2.2.4 开发扩展板 (LINK板) 接口

插针或按钮	功能描述
JP1	DSP的仿真接口
J2	UART485接口
J1	UART232接口
P7	看门狗硬件禁止, 如果P7接跳线帽, 禁止摄像头的看门狗功能
TRIG按钮	外触发信号输入
P9_RESET按钮	DSP复位信号输入
LED灯	D1到D10, 表示IO的高点状态

表格2-3 OK系列智能摄像头开发扩展板 (LINK板) 接口引脚定义



图表2-5 LINK板示意图

## 2.2.4.1 DSP\_JTAG

DSP\_JTAG接口（图表2-5左上）用于连接DSP仿真器，实现对摄像头的在线仿真调试，接口定义如下：

PIN1	TMS
PIN2	TRST
PIN3	TDI
PIN4	GND
PIN5	VCC (+3.3V)
PIN6	NC
PIN7	TDO
PIN8	GND
PIN9	RTCK
PIN10	GND
PIN11	TCK
PIN12	GND
PIN13	EMU0
PIN14	EMU1

表格2-4 LINK板DSP\_JTAG接口引脚定义



图表2-6 LINK板DSP\_JTAG接口示意图

## 2.2.4.2 RS485

串口RS485接口（图表2-5左下），DB9孔，接口定义如下：

PIN1	PIN2	PIN5
A485	B485	GND

表格2-5 LINK板RS485接口引脚定义



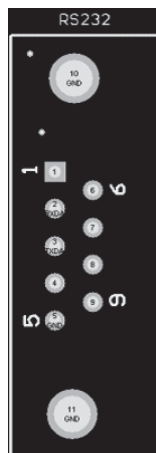
图表2-7 LINK板RS485接口示意图

## 2.2.4.3 RS232

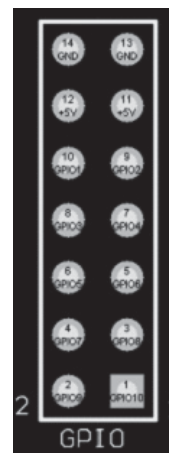
串口RS232 接口（图表2-5右上），DB9孔，接口定义如下（接收/发送以摄像头为参照物）：

PIN2	PIN3	PIN5
RXDA	TXDA	GND

表格2-6 LINK板RS232接口引脚定义



图表2-8 LINK板RS232接口示意图



图表2-9 LINK板GPIO接口示意图

#### 2.2.4.4 GPIO

GPIO 输入输出接口（图表2-5右下），5VTTL，接口定义如下：

PIN1	GPIO10
PIN2	GPIO9
PIN3	GPIO8
PIN4	GPIO7
PIN5	GPIO6
PIN6	GPIO5
PIN7	GPIO4
PIN8	GPIO3
PIN9	GPIO2
PIN10	GPIO1
PIN11	+5V
PIN12	+5V
PIN13	GND
PIN14	GND

表格2-7 LINK板GPIO接口引脚定义

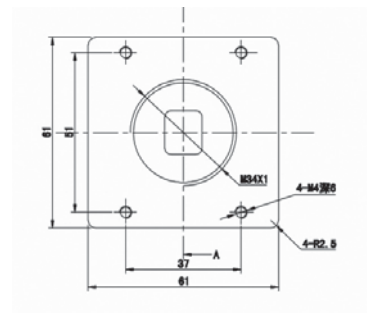
## 第3章

# OK系列智能摄像头 安装与使用说明

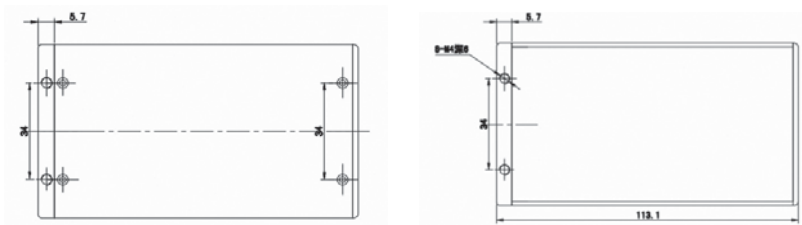
### 3.1 OK系列智能摄像头的安装与固定

#### 3.1.1 固定摄像头

OK系列智能摄像头采用整体成型的金属外壳，用户现场安装非常方便。摄像头外壳的上下各有一个固定螺孔，用以底座安装或者吊顶安装。安装孔的具体机械尺寸，请参看图3-1 OK系列智能摄像头外观尺寸图。

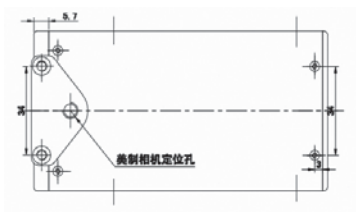


摄像头正面

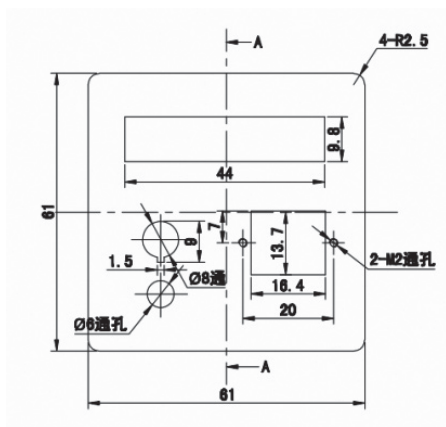


摄像头上表面

摄像头侧面



摄像头下表面



摄像头后面板

图表3-1 OK系列智能摄像头外观尺寸图

### 3.1.2 镜头连接

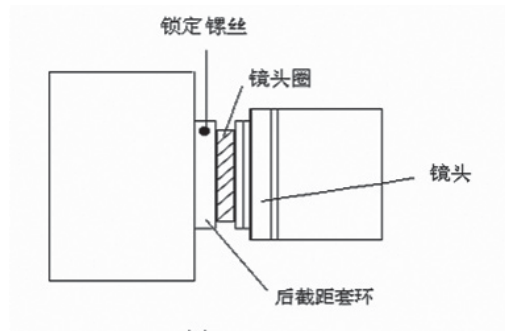
OK系列智能摄像头采用C/CS镜头接口，使用C接口镜头时需加一个5mm的截圈。

### 3.1.3 调节后截距

后截距是指镜头安装面和摄像头CCD靶面之间的距离，图3-2为摄像头后截距图。

调节后截距的步骤如下：

- 1、顺时针拧动镜头，确认镜头是否已拧紧；
- 2、将镜头调焦环调整至全部行程中间的位置，并锁紧顶丝；
- 3、将锁紧圈松开；
- 4、调节镜头直至调出清晰图像；
- 5、锁紧圈锁紧；
- 6、可视情况松开镜头调焦环顶丝进行微调，达到最佳成像效果；
- 7、调整结束后，锁紧调焦环顶丝。



图表3-2 摄像头后截距

### 3.1.4 连接电源和网线

按照2.2章节的外部接口描述正确连接电源线和网线。

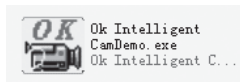
OK系列摄像头的后面板上有一个红色指示灯，用于电源指示，正常状态是上电后常亮，当电源连接出现问题时会熄灭，此时请检查电源连接。

OK系列摄像头的RJ45连接件上有两个指示灯，橙色和绿色。橙色灯和绿色灯同时慢速闪烁说明网络连接不正常。绿色灯快速闪烁说明网络连接正常并且有数据传输。

## 3.2 OK系列智能摄像头上位机软件操作指南

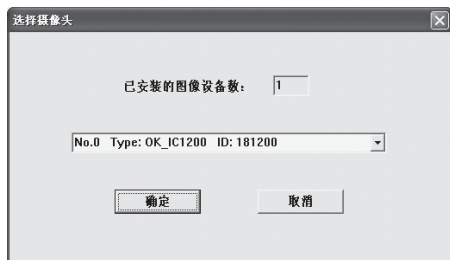
### 3.2.1 打开摄像头

1、运行“Ok Intelligent CamDemo.exe” OK系列智能摄像头演示程序（如图表3-3）。



图表3-3

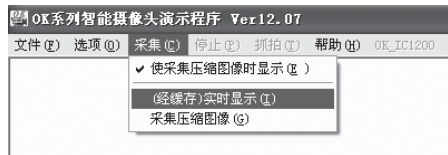
2、选择“选项”->“选择摄像头”，弹出“选择摄像头”对话框（如图表3-4），选择要使用的摄像头后，点击“确定”按钮。



图表3-4

### 3.2.2 采集和停止

1、采集：选择“采集”->“(经缓存)实时显示”（如图表3-5）开始采集并显示图像。



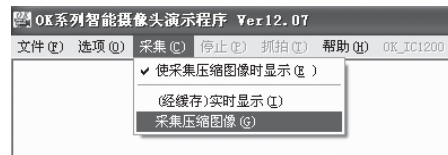
图表3-5

2、停止：选择“停止”停止采集

### 3.2.3 采集压缩图像

需要采集压缩图像到文件的时候，选择“采集”->“采集压缩图像”（如图表3-6），然后在弹出的对话框中指定文件名和存储路径，点击确定（如图表3-7）。需要停止时选择“停止”，完成采集压缩图像到文件的过程。

如果需要采集压缩图像时实时显示图像，可以在此过程之前将“采集”->“使采集压缩图像时显示”选项打勾（如图表3-6）；如果不需要采集压缩图像时实时显示图像，可以在此过程之前将“采集”->“使采集压缩图像时显示”选项对勾去掉。默认是打勾状态。



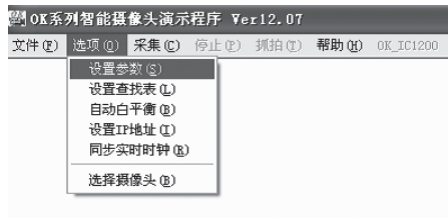
图表3-6



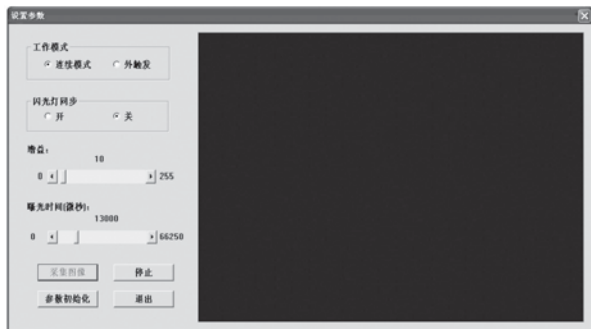
图表3-7

### 3.2.4 设置参数

如果需要对摄像头的参数进行设置，可以选择“选项”->“设置参数”（如图表3-8）。在弹出的“设置参数”对话框中（如图表3-9）可以设置摄像头的工作模式、闪光灯同步、增益、曝光时间，将摄像头参数初始化。



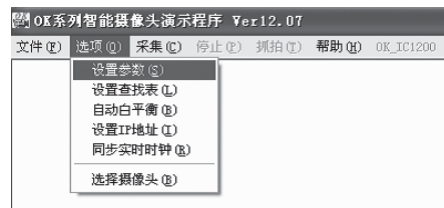
图表3-8



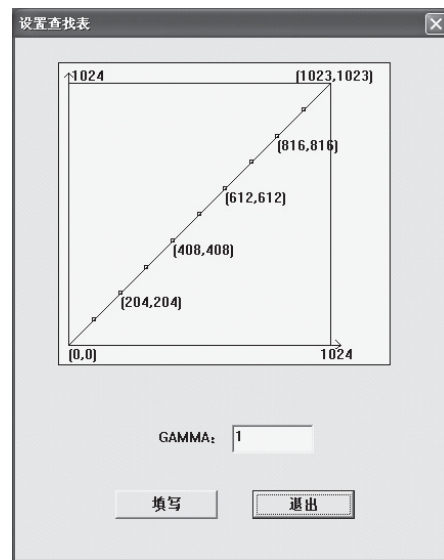
图表3-9

### 3.2.5 设置查找表

如果需要设置摄像头的查找表，可以选择“选项”->“设置查找表”（如图表3-10）。在弹出的“设置查找表”对话框（如图表3-11）中，设置GAMMA校正的数值，数值范围是0-1之间的小数。点击“填写”后，可以看到设置的曲线，以及图像的相应变化。



图表3-10



图表3-11

### 3.2.6 自动白平衡

如果彩色摄像头的颜色有偏色，可以通过自动白平衡来解决。设置方法为：将摄像头对准一张白纸拍摄，采集并显示图像，然后选择“选项”->“自动白平衡”（如图表3-12）。



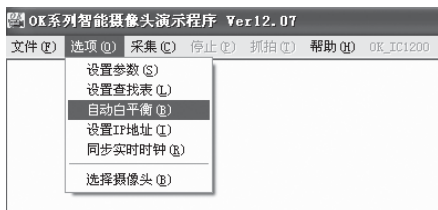


图 表 3-12

### 3.2.7 设置IP地址

选择“选项”->“设置IP地址”（如图表3-13），在弹出的“设置IP地址”对话框（如图表3-14）中，默认显示摄像头的当前IP。用户可以填写需要设置的IP地址，然后点击“确定”。

设置IP地址后，摄像头需要先断电再上电，新设置的IP地址才能生效。

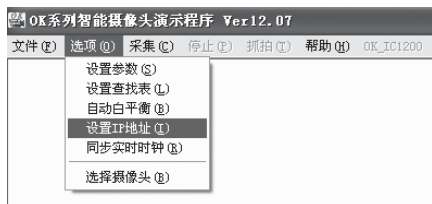


图 表 3-13

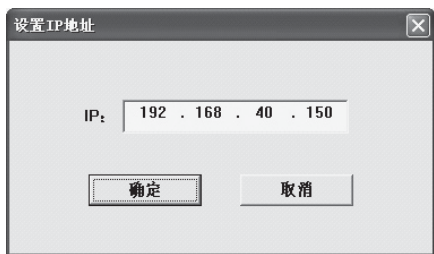


图 表 3-14

### 3.2.8 同步实时时钟

选择“选项”->“同步实时时钟”（如图表3-15），可以将摄像头时钟与电脑的系统时间同步。在界面右下角的状态栏处，显示摄像头内部的温度和实时时钟（如图表3-16）。

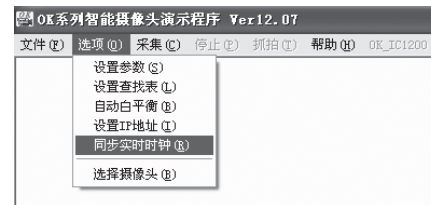


图 表 3-15



图 表 3-16

### 3.2.9 软件/硬件外触发

1、软件外触发：选择“选项”->“设置参数”将摄像头设置成外触发工作模式，然后选择“采集”->“（经缓存）实时显示”开始采集图像。此时，每点击一次“抓拍”则进行一次软件外触发。

2、硬件外触发：选择“选项”->“设置参数”将摄像头设置成外触发工作模式，然后选择“采集”->“（经缓存）实时显示”开始采集图像。此时，每收到一次硬件外触发信号则进行一次硬件外触发。

# 第4章

## OK系列智能摄像头 嵌入式编程指南

### 4.1 OK\_IM系列智能摄像头嵌入式编程指南

#### 4.1.1 OK\_IM系列智能摄像头DEMO软件简介

##### 4.1.1.1 OK\_IM系列智能摄像头工作模式介绍

OK\_IM系列智能摄像头是采用高分辨率工业级逐行扫描黑白CCD图像传感器的高端摄像头，内部具有宽增益范围可编程放大器(0~36dB逐dB可调)和10位ADC。摄像头附带的DEMO给用户提供了一个基于DSP/BIOS 实时操作系统的最基本的软件开发平台，网络通信使用了TI公司的NDK，用户可在其上参考DEMO建立自己的应用程序。

OK\_IM系列智能摄像头具有四种运行模式，用户可任意在其中无级切换。

- 全分辨率触发(抓拍)模式(Trigger Mode --- Trg Mode)
- 全分辨率连续模式(Fine Continuous Mode --- FCon Mode)
- 局部分辨率连续模式(Part Continuous Mode --- Part Mode)——OK\_IM1160特有
- 局部分辨率触发模式(Part Trigger Mode --- Part Trg Mode)——OK\_IM1160特有

在全分辨率触发模式下，用户可通过外部 TTL 触发信号(上升沿) 抓取一帧图像，或通过上位机软件设置抓拍次数和抓拍间隔来抓取一帧或多帧图像；在全分辨率连续模式和局部分辨率连续模式下，用户可以连续抓取图像。

#### 4.1.1.2 DEMO工程软件主体架构介绍

DEMO工程软件主体架构如下图所示：

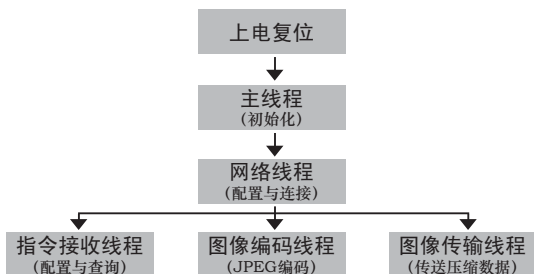


图4-1 OK\_IM系列智能摄像头DEMO工程软件主体架构

##### ①主线程

DEMO工程入口函数为“main( )”。在此函数中，主要完成摄像头运行所需的基本参数配置，并开放中断使能。用户可在“OkIM\_Init( )”函数后添加自己的初始化函数。

##### ②网络线程

DEMO工程的网络通信部分使用了TI公司的NDK，使用标准网络套接字模式进行数据传递。

网络线程对应函数入口为“tskTCPIP( )”，在DSP/BIOS系统中被设定为优先级3，主要完成摄像头MAC配置、网络参数配置、指令接收线程及图像数据传输线程挂起、启动摄像头图像采集通路四项工作。

##### ③图像编码线程

图像编码线程对应函数入口为“tskMJPEGEncode( )”，在DSP/BIOS系统中被设定为优先级2，主要完成摄像头图像数据的JPEG编码功能。

##### ④指令接收线程

指令接收线程对应函数入口为“tskTCPIPTxCmd( )”，主要负责PC机与摄像头间的指令及数据的传递。

##### ⑤图像传输线程

图像传输线程对应函数入口为“tskTCPIPTxImage( )”，主要负责向PC机传送编码后的JPEG图像数据。

#### 4.1.1.3 中断服务程序介绍

##### ①帧传输触发中断

中断源：GPIO4。

函数入口：ISR\_CCDReady ( )。

作用：此中断表示摄像头CCD已经准备好一帧图像，请求进行传输操作。

中断服务程序说明：

关闭上帧数据传输使用的EDMA通路；在四个循环缓冲中，选择存储当前帧所需的缓存目标地址；配置当前帧数据传输使用的EDMA通路。

##### ②数据传输中断

中断源：EDMA通路9。

函数入口：isr\_CaptureFinished ( )。

作用：此中断表示已经将采集到的图像数据送至目标缓存。

中断服务程序说明：

清除中断标志；关闭当前帧数据传输使用的EDMA通路；为编码配置图像数据缓存地址；发送编码信号。

全分辨率连续模式下，中断服务程序只负责发送第一帧图像数据的编码信号量，后续编码信号量的发送在编码数据传输完毕后发送。

全分辨率触发(抓拍)模式下，中断服务程序通过判断当前的触发抓拍状态来发送各触发帧的编码信号量。

##### ③外部触发中断

中断源：GPIO5。

函数入口：isr\_ExternTriggerInt ( )。

作用：此中断表示有外触发抓拍信号产生。

中断服务程序说明：

判断摄像头当前运行状态来向CCD发送触发抓拍指令。

在这个中断函数中通过判断上一帧图像是否传输完毕来决定是否启动采集新一帧图像。

##### ④定时中断

中断源：TIMER2。

函数入口：isr\_TimerInt ( )。

作用：此中断表示设定时间点已经到达。

中断服务程序说明：

判断摄像头当前运行状态来向CCD发送触发抓拍指令；判断定时任务是否结束。

摄像头使用Timer2控制抓拍间隔，每次抓拍间隔到了就会进入该中断函数。中断服务程序通过判断上一帧图像是否传输完毕来决定是否启动采集新一帧图像，直到达到设定的抓拍帧数定时器才暂停工作，计数归零。

◎串口中断

中断源：GPIO6。

函数入口：isr\_UARTInt ( )。

作用：接收串口数据。

中断服务程序说明：

RS232与RS485共享一个中断源。进入中断服务程序后，需要先判断是哪条通路发起的中断申请，然后再根据具体需要进行处理。

◎通用定时器中断

中断源：TIMER1。

函数入口：isr\_TimerInt(void)。

作用：无。

中断服务程序说明：

可根据用户具体需求开启TIMER1定时功能。

#### 4.1.1.4 图像采集格式及存储架构介绍

摄像头的图像采集格式为8位灰度点阵格式。用户可以编写相关的处理函数对其进行格式转换。

原始图像缓存为八帧，轮换使用。图像数据传输中断服务程序中的g\_ui32ImageSrcAddr变量用于反映当前帧图像数据的存放地址。

用户可根据自己的需要修改数据存储架构以提升摄像头运行效率。

### 4.1.2 OK\_IM系列智能摄像头提供的存储器资源

#### 4.1.2.1 SDRAM (128MB)

SDRAM用于存储DSP运行所需要的代码和数据。用户定义的数据段，如不特殊指

定存放在片内，则将全部存储于SDRAM中。

SDRAM段中提供了16MB的堆空间，堆标识符定义为“extHeap”，用户可使用MEM\_malloc函数从其中动态申请存储空间。例如：

```
g_pcUserDataBuf = MEM_malloc(extHeap, 0x100000, 256);
if(g_pcUserDataBuf == NULL)
{
return FALSE;
}
```

#### 4.1.2.2 EEPROM (32KB)

EEPROM的前16KB保留给摄像头使用，其余存储空间开放给用户自由使用。用户不可对前16KB进行写操作，否则会破坏摄像头上电初始化必须的一些参数，导致摄像头工作不正常。

#### 4.1.2.3 代码FLASH (1MB)

用于存储出厂演示程序。其中的数据内容不可更改。

#### 4.1.2.4 数据FLASH (8MB)

用于存储用户代码和数据。前4MB空间用于存放用户代码，后4MB空间用于存放用户参数。

### 4.1.3 DEMO工程存储器使用分配

数据和代码段的分配，在工程“DSP/BIOS内存分配”和“IM1161.cmd”文件中完成。

① .user\_data\_ext

片外数据段，分配于SDRAM中。用户可以使用#pragma DATA\_SECTION来从SDRAM中申请空间。例如：

```
#pragma DATA_SECTION(stream, ".user_data_ext ")
unsigned char stream[1024*1024];
```

② .InternalCode

片内代码段。如果需要将某个函数放到片内RAM，则需要将该函数分配到.InternalCode段内。例如：

```
#pragma CODE_SECTION(WatchDog, ".InternalCode")
void WatchDog( )
```

```
{
    OkIM_KickWatchdog( );
    return;
}
```

© PACKETMEM、MMBUFFER、OBJMEM

NDK使用的数据段。需要保留。

④jpgenc\_tbl、.jpgdec\_tbl、.user\_data\_int、cache\_align、.jpgtbls

JPEG编码库使用的数据和代码段。需要保留。

#### 4.1.4 OK\_IM系列智能摄像头嵌入式编程API函数说明

##### 1、OkIM\_Init

原 型：void OkIM\_Init( )

参 数：无。

返回值：无。

说 明：初始化摄像头硬件，负责CACHE、定时器、IIC模块、GPIO模块等摄像头参数的初始化配置。该函数中调用了OkIM\_Set\_InitParameter函数进行曝光、增益等参数配置。

##### 2、OkIM\_Set\_InitParameter

原 型：void OkIM\_Set\_InitParameter( )

参 数：无。

返回值：无。

说 明：初始化摄像头参数，包括摄像头增益、RGB通道增益、两种模式的曝光时间等。

##### 3、OkIM\_Get\_Hardware\_Version

原 型：Uint32 OkIM\_Get\_Hardware\_Version( )

参 数：无。

返回值：硬件版本号。

说 明：获得硬件版本号。

##### 4、OkIM\_Get\_PIO\_Direction

原 型：Uint32 OkIM\_Get\_PIO\_Direction( )

参 数：无。

返回值：十个可编程 I/O 对应的输入/输出方向，数据低位对应I/O低位，数据高位对应I/O高位。

说 明：获得十个可编程 I/O 对应的输入/输出方向。

##### 5、OkIM\_Get\_Afe\_Pga

原 型：float OkIM\_Get\_Afe\_Pga( )

参 数：无。

返回值：传感器输出信号的增益，单位为dB，范围为0~36dB。

说 明：获得传感器输出信号的增益。

##### 6、OkIM\_Get\_FlashSyncStatus

原 型：Uint8 OkIM\_Get\_FlashSyncStatus( )

参 数：无。

返回值：闪光灯同步输出状态，1为同步闪烁使能，0为禁止。

说 明：获得闪光灯同步输出状态。

##### 7、OkIM\_Get\_TriggerShutter\_Time

原 型：Uint32 OkIM\_Get\_TriggerShutter\_Time( )

参 数：无。

返回值：触发模式下的曝光时间。

说 明：获得触发模式下的曝光时间。

##### 8、OkIM\_Get\_FConShutter\_Time

原 型：Uint32 OkIM\_Get\_FConShutter\_Time( )

参 数：无。

返回值：全分辨率连续模式下的曝光时间。

说 明：获得全分辨率连续模式下的曝光时间。

##### 9、OkIM\_GAMA

原 型：void OkIM\_GAMA(float fLUTGama)

参 数：fLUTGama，GAMA校正系数。

返回值：无。

说 明：设定摄像头GAMA校正系数，通过LUTGama变量进行GAMA校正系数的设定。

##### 10、OkIM\_Get\_DSPFirmware\_Version

原 型：Uint32 OkIM\_Get\_DSPFirmware\_Version( )

参 数：无。

返回值：DSP固件程序版本号。

说明：获得DSP固件程序版本号。

#### 11、OkIM\_TransBlock

原型：void OkIM\_TransBlock (Uint32 ui32SrcAddr,  
          Uint32 ui32YSrcAddr, Uint32 ui32USrcAddr, Uint32 ui32VSrcAddr,  
          Uint32 ui8ImageMode, Uint32 ImageWidth, Uint32 ImageHeight)

参数：ui32SrcAddr, 图像源地址;  
        ui32YSrcAddr, 搬移后图像的Y分量缓存首地址;  
        ui32USrcAddr, 搬移后图像的U分量缓存首地址;  
        ui32VSrcAddr, 搬移后图像的V分量缓存首地址;  
        ui8ImageMode, 图像格式, 此处应当为2;  
        ImageWidth, 图像宽度;  
        ImageHeight, 图像高度。

返回值：无。

说明：将采集到的Bayer图像数据转换为YUV420图像数据, 用于JPEG编码输入。

#### 12、OkIM\_Reset\_TrgEnableFlag

原型：void OkIM\_Reset\_TrgEnableFlag( )

参数：无。

返回值：无。

说明：复位允许触发抓拍标志。该标志在触发模式下设置, 表示上一帧触发图片没有传输完毕, 禁止下一次触发。

#### 13、OkIM\_Set\_TrgEnableFlag

原型：void OkIM\_Set\_TrgEnableFlag( )

参数：无。

返回值：无。

说明：置位允许触发抓拍标志。该标志触发模式下设置, 表示上一帧触发图片已经传输完毕, 允许下一次触发。

#### 14、OkIM\_Save\_Parameter

原型：void OkIM\_Save\_Parameter( )

参数：无。

返回值：无。

说明：保存摄像头参数, 包括摄像头增益、各种采集模式的曝光时间和GAMA系数。

#### 15、OkIM\_KickWatchdog

原型：void OkIM\_KickWatchdog( )

参数：无。

返回值：无。

说明：喂狗操作。看门狗上电默认为关闭状态, 使能后此函数生效。

#### 16、OkIM\_Waitusec

原型：void OkIM\_Waitusec(Uint32 ui32Delay)

参数：ui32Delay, 延迟计数, 单位为微秒。

返回值：无。

说明：时间延迟函数。

#### 17、OkIM\_RS232\_Open

原型：void OkIM\_RS232\_Open( )

参数：无。

返回值：无。

说明：RS232串口使能。缓存大小为512KB。

#### 18、OkIM\_RS232\_Close

原型：void OkIM\_RS232\_Close( )

参数：无。

返回值：无。

说明：RS232串口关闭操作。

#### 19、OkIM\_RS232\_SetBaudrate

原型：Bool OkIM\_RS232\_SetBaudrate(Uint32 ui32BaudRate)

参数：ui32BaudRate, RS232串口波特率。

返回值：波特率设置结果, 1表示设置成功, 0表示设置失败。

说明：设置RS232串口波特率 (4800、9600、19200、38400、57600、115200) 。

#### 20、OkIM\_RS232\_WriteData

原型：void OkIM\_RS232\_WriteData(char \*pcWriteBuf, Uint32 ui32Count)

参数：pcWriteBuf, RS232串口输出数据缓存指针;

        ui32Count, RS232串口输出数据量。

返回值：无。

说明：RS232串口写数据块操作。

#### 21、OkIM\_RS232\_ReadBuf

原型：void OkIM\_RS232\_ReadBuf(char \*pcReadBuf, Uint32 ui32Count)

参数：pcReadBuf, RS232串口读取数据缓存指针；  
ui32Count, RS232串口读取数据量。

返回值：无。

说明：RS232串口读数据块操作。

#### 22、OkIM\_IsRS232RcvReady

原型：Bool OkIM\_IsRS232RcvReady( )

参数：无。

返回值：RS232串口数据接收状态，1表示接收到数据，0表示没有接收到数据。

说明：RS232串口数据接收状态查询。

#### 23、OkIM\_GetRS232\_RecvCnt

原型：Uint32 OkIM\_GetRS232\_RecvCnt( )

参数：无。

返回值：RS232串口接收到的数据量值。

说明：RS232串口接收数据量查询。

#### 24、OkIM\_Read\_CurTemperature

原型：float OkIM\_Read\_CurTemperature( )

参数：无。

返回值：当前温度值。

说明：温度查询函数，返回温度值。

#### 25、OkIM\_RTC\_SetTime

原型：Bool OkIM\_RTC\_SetTime(Uint8 \*pui8Time)

参数：pui8Time, 实时时钟数据指针。

返回值：配置状态，1表示配置成功，0表示配置失败。

说明：配置实时时钟。

#### 26、OkIM\_RTC\_ReadTime

原型：void OkIM\_RTC\_ReadTime(Uint8 \*pui8Time)

参数：pui8Time, 实时时钟数据指针。

返回值：无。

说明：实时时钟读取操作。

#### 27、OkIM\_Save\_MAC\_Address

原型：void OkIM\_Save\_MAC\_Address(Uint8\* pui8MACAdd)

参数：pui8MACAdd, MAC地址指针。

返回值：无。

说明：保存目标摄像头MAC地址到EEPROM中。

#### 28、OkIM\_Read\_MAC\_Address

原型：void OkIM\_Read\_MAC\_Address(Uint8\* pui8MACAdd)

参数：pui8MACAdd, MAC地址指针。

返回值：无。

说明：读取目标摄像头MAC地址。

#### 29、OkIM\_Save\_IP\_Address

原型：void OkIM\_Save\_IP\_Address(Uint32 ui32IPAddr)

参数：ui32IPAddr, IP地址。

返回值：无。

说明：保存目标摄像头IP地址到EEPROM中。

#### 30、OkIM\_Read\_IP\_Address

原型：Uint32 OkIM\_Read\_IP\_Address( )

参数：无。

返回值：目标摄像头的IP地址。

说明：读取目标摄像头IP地址，返回摄像头IP地址数据。

#### 31、OkIM\_EEPROM\_Write

原型：Bool OkIM\_EEPROM\_Write(Uint16 ui16SubAddress,  
Uint8\* pui8DataWriteBuff, Uint16 ui16WriteByteNumber)

参数：ui16SubAddress, EEPROM的物理地址；  
pui8DataWriteBuff, 输出数据缓冲区指针；  
ui16WriteByteNumber, 输出数据字节数。

返回值：写入状态，1表示写入成功，0表示写入失败。

说明：用户可以通过此函数向EEPROM的后16KB数据空间中写入数据。

例程：

向EEPROM的16K字节地址处烧写100字节数据，100字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIM_EEPROM_Write(0x4000, (UInt8*)0x82000000, 100);
```

### 32、OkIM\_EEPROM\_Read

原 型: Bool OkIM\_EEPROM\_Read(UInt16 ui16SubAddress, UInt8\* puiDataReadBuff, UInt16 ui16ReadByteNumber)

参 数: ui16SubAddress, EEPROM的物理地址;  
puiDataReadBuff, 数据读取缓冲区指针;  
ui16ReadByteNumber, 读取数据字节数。

返回值: 读取状态, 1表示读取成功, 0表示读取失败。

说 明: 用户可以通过此函数从EEPROM的后16KB数据空间中读取数据。

例 程:

从EEPROM的16K字节地址处, 读取100字节数据, 将100字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIM_EEPROM_Read(0x4000, (UInt8*)0x82000000, 100);
```

### 33、OkIM\_Enable\_TrgFlashlamp\_Syn\_Out

原 型: void OkIM\_Enable\_TrgFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 触发模式下, 闪光灯同步输出使能。

### 34、OkIM\_Enable\_FconFlashlamp\_Syn\_Out

原 型: void OkIM\_Enable\_FconFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 连续模式下, 闪光灯同步输出使能。

### 35、OkIM\_Disable\_TrgFlashlamp\_Syn\_Out

原 型: void OkIM\_Disable\_TrgFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 禁止触发模式闪光灯同步输出。

### 36、OkIM\_Disable\_FconFlashlamp\_Syn\_Out

原 型: void OkIM\_Disable\_FconFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 禁止连续模式闪光灯同步输出。

### 37、OkIM\_Set\_FlashSyncSignalTime

原 型: void okIM\_Set\_FlashSyncSignalTime(UInt32 i32FlashSyncSignalTime\_us)

参 数: ui32FlashSyncSignalTime\_us, 闪光灯同步信号保持时间, 单位为微秒。

返回值: 无。

说 明: 设置闪光灯同步输出信号(高电平)保持时间。

### 38、OkIM\_GetPIOState

原 型: UInt32 OkIM\_GetPIOState( )

参 数: 无。

返回值: 获取当前I/O口的输入输出值, I/O低位对应数据低位。

说 明: 获取10个可编程I/O的当前状态值。I/O状态没有进行锁存。

### 39、OkIM\_Set\_ImgAcquisition\_Format

原 型: void OkIM\_Set\_ImgAcquisition\_Format(  
enum OkIM\_ImgAcquisition\_Format eunmImgFormat)

参 数: eunmImgFormat, 图像采集格式。当前摄像头只支持Bayer数据格式, 输入值必须为2。

返回值: 无。

说 明: 设置图像采集格式。

### 40、OkIM\_Get\_CurImgFormat

原 型: enum OkIM\_ImgAcquisition\_Format OkIM\_Get\_CurImgFormat( )

参 数: 无。

返回值: 图像格式类型。

说 明: 获取当前摄像头图像采集格式。

### 41、OkIM\_Get\_CurRunMode

原 型: enum OkIM\_SensorRunMode OkIM\_Get\_CurRunMode( )

参 数: 无。



返回值：摄像头运行模式类型，0表示触发模式，1表示连续模式。

说明：获取摄像头当前运行模式。

#### 42, OkIM\_Mode\_Switch\_Ctrl

原型：void OkIM\_Mode\_Switch\_Ctrl(  
enum OkIM\_SensorRunMode enumRunMode)

参数：enumRunMode，摄像头运行模式，0为触发模式，1为连续模式。

返回值：无。

说明：切换摄像头运行模式。

#### 43, OkIM\_Enable\_RunModeSwitch\_AutoTrg

原型：void OkIM\_Enable\_RunModeSwitch\_AutoTrg(Bool blsEnable)

参数：blsEnable，自动抓取一帧图像使能。

返回值：无。

说明：设置摄像头是否在切换为触发模式时自动触发一帧图像。

#### 44, OkIM\_Get\_TrgEnableFlag

原型：Bool OkIM\_Get\_TrgEnableFlag( )

参数：无。

返回值：触发允许标志，1表示允许触发，0表示禁止触发。

说明：获取摄像头触发允许标志。

#### 45, OkIM\_Set\_Afe\_Pga

原型：void OkIM\_Set\_Afe\_Pga(float fPgaValue\_db)

参数：fPgaValue\_db，摄像头增益值，为0-36之间的float型数据。

返回值：无。

说明：设置摄像头增益。

#### 46, OkIM\_Grab\_Single\_Frame

原型：void OkIM\_Grab\_Single\_Frame( )

参数：无。

返回值：无。

说明：软触发抓拍图像。外触发源以外的其他触发源可以调用此函数实现触发抓拍。

#### 47, OkIM\_Set\_TrgShutter\_Time

原型：Bool OkIM\_Set\_TrgShutter\_Time(UINT32 ui32TrgShutterTime\_us)

参数：ui32TrgShutterTime\_us，触发模式曝光时间。

返回值：配置状态，1表示配置完成，0表示配置未完成。

说明：设置触发模式的曝光时间。

#### 48, OkIM\_Set\_FConShutter\_Time

原型：Bool OkIM\_Set\_FConShutter\_Time(UINT32 ui32FConShutterTime\_us)

参数：ui32FConShutterTime\_us，连续模式曝光时间。

返回值：配置状态，1表示配置完成，0表示配置未完成。

说明：设置连续模式的曝光时间。

#### 49, OkIM\_Open\_Imagechannel

原型：void OkIM\_Open\_Imagechannel( )

参数：无。

返回值：无。

说明：开启图像采集通道，开启后摄像头可进行图像采集。

#### 50, OkIM\_Close\_Imagechannel

原型：void OkIM\_Close\_Imagechannel( )

参数：无。

返回值：无。

说明：关闭图像采集通道。

#### 51, OkIM\_Set\_Image\_LUT10bit

原型：void OkIM\_Set\_Image\_LUT10bit(UINT8\* pui8ImgLUTBuf)

参数：pui8ImgLUTBuf，LUT数据指针，此数据应当为1024个0-255大小范围的整型数据。

返回值：无。

说明：将输入的LUT数据存储于EEPROM中，并按照查找表内容使能LUT。

#### 52, OkIM\_Disable\_Image\_LUT

原型：void OkIM\_Disable\_Image\_LUT(Bool blsSetEEPROM)

参数：blsSetEEPROM，LUT数据恢复标识。

返回值：无。

说明：禁止LUT功能。若LUT数据恢复标识为1，则改写EEPROM中的数据到出厂状态。

**53, OkIM\_Enable\_Image\_LUT**

原 型: void OkIM\_Enable\_Image\_LUT( )

参 数: 无。

返回值: 无。

说 明: 使能摄像头LUT功能, EEPROM中的LUT数据将被启用。

**54, OkIM\_Resume\_Image\_LUT**

原 型: void OkIM\_Resume\_Image\_LUT( )

参 数: 无。

返回值: 无。

说 明: 恢复EEPROM中的LUT为出厂状态。出厂默认LUT状态为四段直线, 五个拐点分别为: (0, 0), (255, 84), (512, 151), (768, 210), (1023, 255)。

**55, OkIM\_Set\_Pio\_Direction**

原 型: void OkIM\_Set\_Pio\_Direction(Uint32 ui32PioDirList)

参 数: ui32PioDirList, I/O方向变量, 0为输入, 1为输出, I/O方向变量低位对应I/O低位。

返回值: 无。

说 明: 设置10个I/O数据流方向。

**56, OkIM\_Set\_Pio\_Outvalue**

原 型: void OkIM\_Set\_Pio\_Outvalue(Uint32 ui32PioOutvalueList)

参 数: ui32PioOutvalueList, I/O管脚输出值, I/O管脚输出值低位对应I/O低位。

返回值: 无。

说 明: 设置10个I/O管脚上的输出值。

**57, OkIM\_GetPIOIntTime**

原 型: Uint32 OkIM\_GetPIOIntTime(Uint32 ui32PioID)

参 数: ui32PioID, I/O管脚ID, 范围为0-9, 0-9分别对应I/O的1-10号管脚。

返回值: 中断产生时间。

说 明: 如果允许中断产生, 则在中断发生后, 可通过此函数查询对应I/O的中断时间。中断时间产生于系统计数器, 返回值为32位计数值。系统计数器跳变时间为1微秒。

**58, OkIM\_Flash\_Burn**

原 型: Bool OkIM\_Flash\_Burn(Uint8\* pui8Source, Uint32 ui32DestAddr, Uint32 ui32Length)

参 数: pui8Source, 输出数据缓存指针;

ui32DestAddr, 写入数据的FLASH目标地址, 此处地址为FLASH的物理地址, 不是DSP中的相对地址;

ui32Length, 写入数据的字节数。

返回值: 写入状态, 1表示写入成功, 0表示写入失败。

说 明: 用户可通过此函数向FLASH的后4MB数据空间中写入数据。

例 程:

向FLASH的4M字节地址处烧写3000字节数据, 3000字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIM_Flash_Burn( (Uint8*)0x82000000, 0x400000, 3000 );
```

**59, OkIM\_Flash\_Read**

原 型: Bool OkIM\_Flash\_Read(Uint8\* pui8DataBuf, Uint32 ui32FlashAddr, Uint32 ui32Length)

参 数: pui8DataBuf, 读取数据缓存指针;

ui32FlashAddr, 读取数据的FLASH目标地址, 此处地址为FLASH的物理地址, 不是DSP中的相对地址;

ui32Length, 读取数据字节数。

返回值: 读取状态, 1表示读取成功, 0表示读取失败。

说 明: 用户可通过此函数从FLASH的后4MB空间中读取数据。

例 程:

从FLASH的4M字节地址处读取3000字节数据, 将3000字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIM_Flash_Read( (Uint8*)0x82000000, 0x400000, 3000 );
```

**60, OkIM\_RS485\_Open**

原 型: void OkIM\_RS485\_Open( )

参 数: 无。

返回值: 无。

说明: RS485串口使能。

#### 61、OkIM\_RS485\_Close

原型: void OkIM\_RS485\_Close( )

参数: 无。

返回值: 无。

说明: RS485串口关闭。

#### 62、OkIM\_RS485\_SetBaudrate

原型: Bool OkIM\_RS485\_SetBaudrate(Uint32 ui32BaudRate)

参数: ui32BaudRate, RS485串口波特率。

返回值: 波特率设置结果, 1表示设置成功, 0表示设置失败。

说明: 设置RS485串口波特率 (4800、9600、19200、38400、57600、115200)。

#### 63、OkIM\_RS485\_WriteData

原型: void OkIM\_RS485\_WriteData(char \*pcWriteBuf, Uint32 ui32Count)

参数: pcWriteBuf, RS485串口输出数据缓存指针;  
ui32Count, RS485串口输出数据量。

返回值: 无。

说明: RS485串口数据写数据块操作。

#### 64、OkIM\_RS485\_ReadBuf

原型: void OkIM\_RS485\_ReadBuf(char \*pcReadBuf, Uint32 ui32Count)

参数: pcReadBuf, RS485串口读取数据缓存指针;  
ui32Count, RS485串口读取数据量。

返回值: 无。

说明: RS485串口数据读数据块操作。

#### 65、OkIM\_IsRS485RcvReady

原型: Bool OkIM\_IsRS485RcvReady( )

参数: 无。

返回值: RS485串口数据接收状态, 1表示接收到数据, 0表示没有接收到数据。

说明: RS485串口数据接收状态查询。

#### 66、OkIM\_GetRS485\_RecvCnt

原型: Uint32 OkIM\_GetRS485\_RecvCnt( )

参数: 无。

返回值: RS485串口接收到的数据量值。

说明: RS485串口接收数据量查询。

#### 67、OkIM\_GetTickCount

原型: Uint32 OkIM\_GetTickCount( )

参数: 无。

返回值: 系统计数器值。

说明: 系统计数器查询函数, 返回一个32位的系统计数值。系统计数器每间隔1微秒跳变一次。

#### 68、OkIM\_Reset

原型: void OkIM\_Reset( )

参数: 无。

返回值: 无。

说明: DSP系统复位。

#### 69、OkIM\_Watchdog\_Open

原型: void OkIM\_Watchdog\_Open( )

参数: 无。

返回值: 无。

说明: 看门狗使能。

#### 70、OkIM\_Watchdog\_Close

原型: void OkIM\_Watchdog\_Close( )

参数: 无。

返回值: 无。

说明: 关闭看门狗。

#### 71、OkIM\_Set\_WatchdogTimer

原型: void OkIM\_Set\_WatchdogTimer(Uint32 ui32WDDTTime\_S)

参数: ui32WDDTTime\_S, 看门狗复位时间, 单位为秒, 设定范围为1-255。

返回值: 无。

说明: 设定看门狗复位时间。

#### 72、OkIM\_JPEGENCINIT

原型: void OkIM\_JPEGENCINIT( )

参数: 无。

返回值：无。

说明：JPEG编库初始化。如使用编库，则需要在main( )函数中对其初始化。

### 73、OkIM\_MoveImageData\_and\_wait

原型：void OkIM\_MoveImageData\_and\_wait (UInt32 SrcDataAddr,  
 UInt32 DstDataAddr, UInt8 ImageFormat, UInt16 clmageWidth,  
 UInt16 clmageHeight)

参数：SrcDataAddr, 数据源地址;  
 DstDataAddr, 数据目的地址;  
 ImageFormat, 图像格式, 此处应当为2;  
 clmageWidth, 图像宽度;  
 clmageHeight, 图像高度。

返回值：无。

说明：将循环缓冲中的数据搬运到指定目标地址。该函数使用32位QDMA实现其功能。

### 74、OkIM\_JPGENC\_Parameter\_Config

原型：void OkIM\_JPGENC\_Parameter\_Config (UInt32 ui32YUVFormat,  
 UInt32 ui32quality, UInt32 ui32HResolution, UInt32 ui32VResolution)

参数：ui32YUVFormat, 图像格式, 此处为2;  
 ui32quality, JPEG压缩因子, 目前只支持30-95;  
 ui32HResolution, 编码图像宽度;  
 ui32VResolution, 编码图像高度。

返回值：无。

说明：设置JPEG参数。

### 75、OkIM\_JPGEncoder

原型：void OkIM\_JPGEncoder ( UInt8\* pui8YBuf, UInt8\* pui8CrBuf,  
 UInt8\* pui8CbBuf, UInt8\* pui8JpgOutBuf[] )

参数：pui8YBuf, 编码的YUV420数据Y分量数据指针;  
 pui8CrBuf, 编码的YUV420数据U分量数据指针;  
 pui8CbBuf, 编码的YUV420数据V分量数据指针;  
 pui8JpgOutBuf[], 输出数据指针数组, 第一个指针地址返回JPEG编码图像的大小, 第二个指针地址返回JPEG编码的图像数据。

返回值：无。

说明：JPEG编码操作。输入为YUV420格式的图像数据, 输出为JPEG编码数据。

### 76、OkIM\_GetJPEGCompressQuality

原型：int OkIM\_GetJPEGCompressQuality (UInt8 RunMode)

参数：RunMode, 摄像头运行模式。

返回值：压缩因子。

说明：查询当前的JPEG编码压缩因子。

### 77、OkIM\_SetSnapIntervaltime

原型：void OkIM\_SetSnapIntervaltime (UInt32 Intervaltime)

参数：Intervaltime, 抓拍间隔时间, 单位为毫秒。

返回值：无。

说明：如使用定时器进行间隔抓拍, 则使用此函数设定抓拍间隔。抓拍间隔默认为1秒。

### 78、OkIM\_GetSnapIntervaltime

原型：UInt32 OkIM\_GetSnapIntervaltime ( )

参数：无。

返回值：抓拍间隔。

说明：如使用定时器进行间隔抓拍, 则使用此函数查询抓拍间隔。

### 79、OkIM\_SetSnap

原型：void OkIM\_SetSnap ( )

参数：无。

返回值：无。

说明：如使用定时器进行间隔抓拍, 则使用此函数启动定时抓拍。

### 80、OkIM\_StopTimeSnap

原型：void OkIM\_StopTimeSnap ( )

参数：无。

返回值：无。

说明：如使用定时器进行间隔抓拍, 则使用此函数停止定时抓拍操作。

### 81、OkIM\_EDMAClose

原型：void OkIM\_EDMAClose ( )

参数：无。

返回值：无。

说明：关闭VP的EDMA数据搬移通路。

### 82、OkIM\_EDMAOpen

原型：void OkIM\_EDMAOpen(Uint8 ui8ImageMode,  
                          Uint32 ImgDstAddr)

参数：ui8ImageMode, 图像格式, 此处为2;  
          ImgDstAddr, 目标地址, 必须四字节对齐。

返回值：无。

说明：开启VP的EDMA数据搬移通路。

### 83、OkIM\_Updata

原型：Bool OkIM\_Updata(Uint8 \*pBurndata, Uint16 RebootFlag)

参数：pBurndata, 烧写数据缓存指针;  
          RebootFlag, 烧写标识, 为1则烧写数据, 为0则擦除代码空间。

返回值：烧写状态, 1表示烧写成功, 0表示烧写失败。

说明：用于远程更新模块, 向代码8M串行FLASH的前4MB空间烧写数据。

### 84、OkIM\_SetYUV420Data\_UV

原型：void okIM\_SetYUV420Data\_UV(unsigned char \*ui32USrcAddr, unsigned  
                                  char \*ui32VSrcAddr, Uint32 ImageWidth, Uint32 ImageHeight)

参数：ui32USrcAddr, 预编码图像U分量数据存储空间;  
          ui32VSrcAddr, 预编码图像V分量数据存储空间;  
          ImageWidth, 图像宽度;  
          ImageHeight, 图像高度。

返回值：无。

说明：在摄像头主函数中, 负责初始化预编码图像的UV数据。

### 85、okIM\_GetGAMA

原型：float okIM\_GetGAMA( )

参数：无。

返回值：Gama系数。

说明：得到摄像头Gama系数。

### 86、okIM\_SetPartImgSize

原型：void okIM\_SetPartImgSize(Uint16 PartImg\_w, Uint16 PartImg\_h)

参数：PartImg\_w, 局部模式图像宽度;

          PartImg\_h, 局部模式图像高度。

返回值：无。

说明：OK\_IM1160特有, 设定局部连续模式和局部触发模式的图像宽高。

### 87、okIM\_Enable\_PartFlashlamp\_Syn\_Out

原型：void okIM\_Enable\_PartFlashlamp\_Syn\_Out( )

参数：无。

返回值：无。

说明：OK\_IM1160特有, 局部连续模式闪光灯同步输出使能。

### 88、okIM\_Enable\_PartTrgFlashlamp\_Syn\_Out

原型：void okIM\_Enable\_PartTrgFlashlamp\_Syn\_Out( )

参数：无。

返回值：无。

说明：OK\_IM1160特有, 局部触发模式闪光灯同步输出使能。

### 89、okIM\_Disable\_PartFlashlamp\_Syn\_Out

原型：void okIM\_Disable\_PartFlashlamp\_Syn\_Out( )

参数：无。

返回值：无。

说明：OK\_IM1160特有, 局部连续模式闪光灯同步输出禁止。

### 90、okIM\_Disable\_PartTrgFlashlamp\_Syn\_Out

原型：void okIM\_Disable\_PartTrgFlashlamp\_Syn\_Out( )

参数：无。

返回值：无。

说明：OK\_IM1160特有, 局部触发模式闪光灯同步输出禁止。

### 91、okIM\_Set\_PartShutter\_Time

原型：Bool okIM\_Set\_PartShutter\_Time(Uint32 ui32PartShutterTime\_us)

参数：ui32PartShutterTime\_us, 局部连续模式的曝光时间, 单位为微秒。

返回值：曝光时间配置是否成功, 1表示配置成功, 0表示配置失败。

说明：OK\_IM1160特有, 设定局部连续模式的CCD曝光时间。

### 92、okIM\_Set\_PartTrgShutter\_Time

原型：Bool okIM\_Set\_PartTrgShutter\_Time(Uint32 ui32PartTrgShutterTime\_us)

参 数：ui32PartTrgShutterTime\_us，局部触发模式的曝光时间，单位为微秒。

返回值：曝光时间配置是否成功，1表示配置成功，0表示配置失败。

说 明：OK\_IM1160特有，设定局部触发模式的CCD曝光时间。

#### 93. okIM\_Get\_PartShutter\_Time

原 型：Uint32 okIM\_Get\_PartShutter\_Time( )

参 数：无。

返回值：局部连续模式的曝光时间，单位为微秒。

说 明：OK\_IM1160特有，获取局部连续模式的CCD曝光时间。

#### 94. okIM\_Get\_PartTrgShutter\_Time

原 型：Uint32 okIM\_Get\_PartTrgShutter\_Time( )

参 数：无。

返回值：局部触发模式的曝光时间，单位为微秒。

说 明：OK\_IM1160特有，获取局部触发模式的CCD曝光时间。

#### 95. OkIM\_SetTimerCounter

原 型：void OkIM\_SetTimerCounter(Uint32 Counter)。

参 数：Counter，通用定时器定时周期。

返回值：无。

说 明：用于75MHz通用定时器定时周期设置。

#### 96. OkIM\_TimerStart

原 型：void OkIM\_TimerStart( )。

参 数：无。

返回值：无。

说 明：启动通用定时器。启动后，通用定时器重复运行，不自动停止。

#### 97. OkIM\_TimerStop( )

原 型：void OkIM\_TimerStop( )。

参 数：无。

返回值：无。

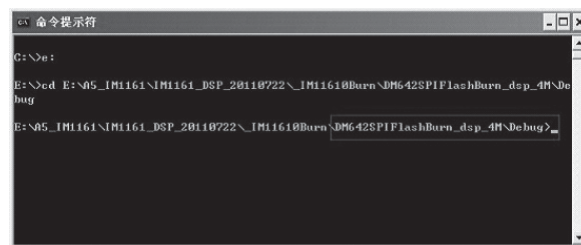
说 明：停止通用定时器。

### 4.1.5 DSP程序固化说明

步骤1：编译预加载的“XXX”工程文件，在工程文件夹中的“Debug”文件夹下找到生成“XXX.out”文件。

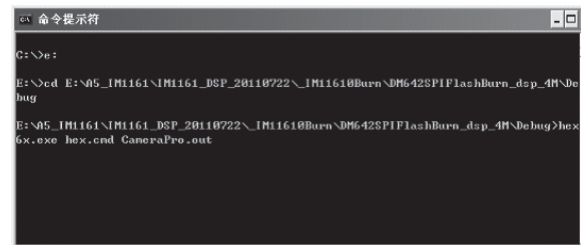
步骤2：将“XXX.out”文件复制到烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹下。

步骤3：开启WINDOWS命令提示符。将文件目录设置到烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹。



图表4-2

步骤4：键入命令“hex6x.exe hex.cmd XXX.out”（见图表4-3），回车执行命令，并等待程序执行完成，出现图表4-4所示的界面。其中“XXX.out”是目标烧写文件。



图表4-3

```

C:\>命令提示符
C:\>E:
E:\>cd E:\N5_IM1161\IM1161_DSP_20110722\IM11610Burn\DM642SPIFlashBurn_dsp_4M\De
bug
E:\>N5_IM1161\IM1161_DSP_20110722\IM11610Burn\DM642SPIFlashBurn_dsp_4M\Debug>hex
6x.exe hex.cmd CameraPro.out
>> WARNING: section .boot not found in CameraPro.out
Translating CameraPro.out to ASCII-Hex format...
"CameraPro.out" ==> .stc (BOOT LOAD)
"CameraPro.out" ==> .log (BOOT LOAD)
"CameraPro.out" ==> .trace (BOOT LOAD)
"CameraPro.out" ==> .abi_vec (BOOT LOAD)
"CameraPro.out" ==> .data (BOOT LOAD)
"CameraPro.out" ==> .stack (BOOT LOAD)
"CameraPro.out" ==> .text (BOOT LOAD)
"CameraPro.out" ==> .cinit (BOOT LOAD)
"CameraPro.out" ==> .switch (BOOT LOAD)
"CameraPro.out" ==> .args (BOOT LOAD)
"CameraPro.out" ==> .sys_init (BOOT LOAD)
"CameraPro.out" ==> .gbl_init (BOOT LOAD)
"CameraPro.out" ==> .LOG_system$buf (BOOT LOAD)
"CameraPro.out" ==> .trcdata (BOOT LOAD)
"CameraPro.out" ==> .const (BOOT LOAD)
"CameraPro.out" ==> .bios (BOOT LOAD)
"CameraPro.out" ==> .jpgenc_th1 (BOOT LOAD)
"CameraPro.out" ==> .jpgdec_th1 (BOOT LOAD)
"CameraPro.out" ==> .cache_align (BOOT LOAD)
"CameraPro.out" ==> .text: DAI_copy (BOOT LOAD)
"CameraPro.out" ==> .text: DAI_copy2d (BOOT LOAD)
"CameraPro.out" ==> .text: DAI_wait (BOOT LOAD)
"CameraPro.out" ==> .text: EDMA_intalloc (BOOT LOAD)
"CameraPro.out" ==> .text: DAI_open (BOOT LOAD)
"CameraPro.out" ==> .InternalCode (BOOT LOAD)
E:\>N5_IM1161\IM1161_DSP_20110722\IM11610Burn\DM642SPIFlashBurn_dsp_4M\Debug>

```

图4-4

步骤5: 键入命令“flash.exe”, 回车执行命令, 并等待程序执行完成。

```

C:\>命令提示符
"CameraPro.out" ==> .text: EDMA_intalloc (BOOT LOAD)
"CameraPro.out" ==> .text: DAI_open (BOOT LOAD)
"CameraPro.out" ==> .InternalCode (BOOT LOAD)
E:\>N5_IM1161\IM1161_DSP_20110722\IM11610Burn\DM642SPIFlashBurn_dsp_4M\Debug>la
sh.exe

```

图4-5

此时, 在烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹下会生成“flashburn.bin”文件。

步骤6: 使用仿真器连接待烧写的OK\_IM系列摄像头, 在CCS中打开“DM642SPIFlashBurn\_dsp\_4M”文件夹下的“DM642FlashBurn.pjt”烧写工程, 并加载程序, 运行。

步骤7: 等待CCS中显示“Demo Burn Completed!”提示后(见图4-6), 说明已经完成烧写。断开摄像头的LINK板, 对摄像头重新上电后即可执行刚才烧入摄像头的“XXX.out”文件。



图4-6

#### 4.1.6 DSP程序远程更新说明

步骤1: 将JHL\_IM1161\_Update文件夹及其中文件(包括RemoteUpdate.exe、HEX6X.EXE、hex.cmd、flash.exe)存储于计算机根目录下。

步骤2: 将CCS生成的“XXX.out”文件放入JHL\_IM1161\_Update文件夹中。

步骤3: 运行RemoteUpdate.exe。

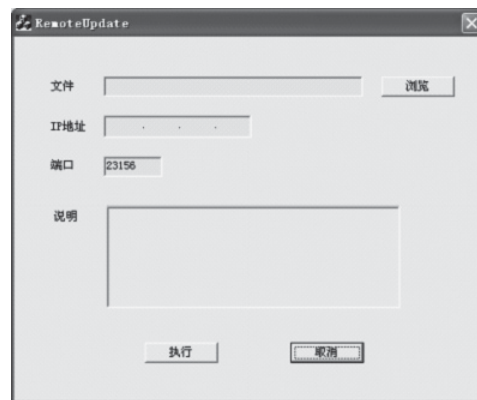
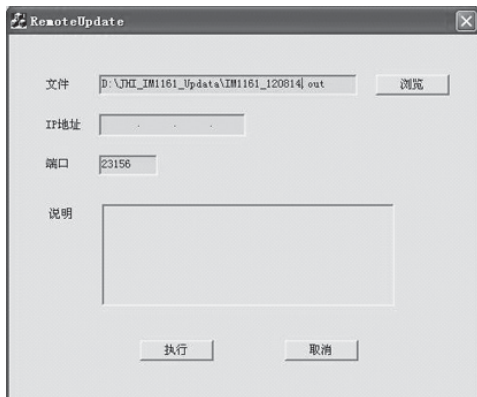


图4-7



步骤4: 点击“浏览”，选择JHI\_IM1161\_Updata文件夹中的“XXX.out”文件。



图表4-8

步骤5: 输入IP地址和对应的端口号（端口号在DEMO中默认为23156）。



图表4-9

步骤6: 点击执行开始更新程序。等待下图所示的四个状态执行完毕，即完成了程序固化，DSP将重启。



图表4-10

注意事项:

1. 如果最后一步出现了烧写失败的情况，则程序将擦除用户程序空间。摄像头重新上电15秒后，会进入DEMO程序。用户可在DEMO程序基础上重新烧写。
2. JHI\_IM1161\_Updata文件夹应存储于计算硬盘根目录下。
3. 如出现无法打开文件的情况，请尝试将要烧入摄像头的“XXX.out”文件名改为英文字符，并缩短文件名称。

#### 4.1.7 用户需了解的其他内容

##### 4.1.7.1 关于开发工具的说明

摄像头DEMO工程基于TI CCS2.20.18版本集成开发环境进行开发与测试。

##### 4.1.7.2 EDMA通道的使用情况

当前摄像头中使用了VPO端口对应的DMA通道，占用了EDMA中断标志TCC1至TCC9和TCC14至TCC15。

##### 4.1.7.3 关于局部图像采集模式

局部图像采集模式与全分辨率图像采集模式相同，分为连续模式和触发模式。每次配置完局部图像采集模式的图像宽高和图像采集位置后，需要进行模式切换才能使配置参数生效。具体实现方法可参考DEMO程序0x01指令和0xa7指令内容。

##### 4.1.7.4 TCP/IP 协议应用编程说明



DEMO程序演示了如何在摄像头平台上进行TCP/IP编程。网络运行环境初始化和服务线程的建立，用户可仔细阅读DEMO程序中的SETUPTCPINET.c文件中提供的例程，同时参阅TI相关文档。在用户建立的服务线程中，与上位机的通讯程序可由标准的Socket套接字编程来实现。DEMO提供的网络服务函数的工作模式均为迭代服务器模型，因此要求上位机端的网络为客户端模型，这种结构对于一个利用多台摄像头组网的网络结构来说是非常有利的，具体数据交互过程请用户参见摄像头端和上位机端的例程。

#### 4.1.7.5 RTC数据格式说明

Addr									Function/Range BCD Format	
	D7	D6	D5	D4	D3	D2	D1	D0		
00h	0.1 Seconds				0.01 Seconds				Seconds	00-99
01h	ST	10 Seconds			Seconds				Seconds	00-59
02h	0	10 Minutes			Minutes				Minutes	00-59
03h	CEB	CB	10 Hours		Hours (24 Hour Format)			Century/ Hours	0-1/00-23	
04h	0	0	0	0	0	Day of Week		Day	01-7	
05h	0	0	10 Date		Date: Day of Month			Date	01-31	
06h	0	0	0	10M	Month			Month	01-12	
07h	10 Years				Year			Year	00-99	

图表4-11 OK\_IM系列智能摄像头RTC寄存器映射表

上层软件如需使用RTC，则需要在每次摄像头上电后，同步一次RTC时钟。

PC端RTC配置代码示例：

```

cmdbuf[0] = 0x18;
cmdbuf[1] = 0x00;
*(UINT*)(cmdbuf+2) = 0x07+1;
cmdbuf[6] = m_RTCConfigDlg.m_iYear%100;
cmdbuf[7] = m_RTCConfigDlg.m_iMonth;
cmdbuf[8] = m_RTCConfigDlg.m_iDay;
cmdbuf[9] = m_RTCConfigDlg.m_iWeek;
cmdbuf[10] = m_RTCConfigDlg.m_iHour;
cmdbuf[11] = m_RTCConfigDlg.m_iMinute;
cmdbuf[12] = m_RTCConfigDlg.m_iSecond;

```

PC端RTC读取代码示例：

```

cmdbuf[0] = 0x19;
cmdbuf[1] = 0x01;
*(UINT*)(cmdbuf+2) = 0x04+1;
*(UINT*)(cmdbuf+6) = 0x00;
*(cmdbuf+10) = GetXorCheck(cmdbuf+6,4);
Camerahour = cRcvBuf[4];
Camera minute = cRcvBuf[5];
Camera second = cRcvBuf[6];

```

摄像头返回数据内容：

摄像头返回7个字节无符号整型数据，从低位到高位分别表示年、月、日、星期、时、分、秒。

#### 4.1.7.6 定时抓拍操作的拓展应用

定时抓拍操作利用DSP的定时器实现精准计时，在定时器的中断服务程序中实现定时抓拍操作。用户也可以对定时器的中断服务函数进行修改，进而实现其它精准定时任务。

相关API操作：

```

C64_enableIER(0x0400); //在主函数中开启定时器中断。
OkIM_SetSnapIntervaltime(1000); //设定定时器运行时间，单位毫秒。
OkIM_SetSnap(); //启动定时器。
OkIM_StopTimeSnap(); //在中断服务程序中关闭定时器。

```

#### 4.1.7.7 串口演示程序说明

DEMO程序中，串口波特率为38400bit/s。串口的最大数据缓存为512字节。向RS232发送数据，如果数据的第一个字节内容为1，则LINK板的LED灯将会被点亮；如果数据的第一个字节内容为2，则LINK板的LED灯将会熄灭；如果数据的第一个字节为其他数据，则回传摄像头接收到的串口数据。

配置串口波特率时，请使用okdsplib.h文件中的宏定义设定串口波特率。

#### 4.1.7.8 关于看门狗操作的说明

系统在PRD周期函数“prd\_WatchDog ( )”中进行喂狗。系统看门狗定时器以秒为基本单位，设置范围为1—255秒。用户可通过修改dm642.cdb文件中的PRD1对象的周期

来修改喂狗时间间隔，单位是ms。

DEMO程序中通过“RunFlg”标志变量判断当前摄像头运行状态是否正常。用户可根据自己的需要，调整此标志变量所在位置。

## 4.2 OK\_IC系列智能摄像头嵌入式编程指南

### 4.2.1 OK\_IC系列智能摄像头DEMO软件介绍

#### 4.2.1.1 OK\_IC1200智能摄像头工作模式介绍

OK\_IC1200智能摄像头是一款采用高分辨率(1600×1200)工业级逐行扫描彩色CCD图像传感器的高端摄像头，内部具有宽增益范围可编程放大器(0~36dB 逐dB可调)和12位ADC。摄像头附带的DEMO给用户提供了一个基于DSP/BIOS实时操作系统的最基本的软件开发平台，网络通信使用了TI公司的NDK，用户可在其上参考DEMO建立自己的应用程序。

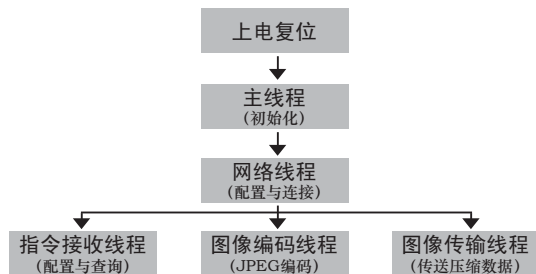
DEMO程序具有两种运行模式，用户可任意在其中无级切换。

- 全分辨率触发(抓拍)模式(Trigger Mode --- Trg Mode)
- 全分辨率连续模式(Fine Continuous Mode --- FCon Mode)

在全分辨率触发模式下，用户可通过外部TTL触发信号(上升沿)抓取一帧图像(分辨率为1600×1200)，或通过上位机软件设置抓拍次数和抓拍间隔来抓取一帧或多帧图像；在全分辨率连续模式下，用户可以连续抓取图像(分辨率为1600×1200)。

#### 4.2.1.2 DEMO工程软件主体架构介绍

DEMO工程软件主体架构如下图所示：



图表4-12 OK\_IC系列智能摄像头DEMO工程软件主体架构

#### ①主线程

DEMO工程入口函数为“main( )”。在此函数中，主要完成摄像头运行所需的基本参数配置，并开放中断使能。用户可在“OkIC\_Init( )”函数后添加自己的初始化函数。

#### ②网络线程

DEMO工程的网络通信部分使用了TI公司的NDK，使用标准网络套接字模式进行数据传输。

网络线程对应函数入口为“tskTCPIP( )”，在DSP/BIOS系统中被设定为优先级3，主要完成摄像头MAC配置、网络参数配置、指令接收线程及图像数据传输线程挂起、启动摄像头图像采集通路四项工作。

#### ③图像编码线程

图像编码线程对应函数入口为“tskMJPEGEncode( )”，在DSP/BIOS系统中被设定为优先级2，主要完成摄像头白平衡操作、图像数据的JPEG编码两项功能。

#### ④指令接收线程

指令接收线程对应函数入口为“tskTCPIPTxCmd( )”，主要负责PC机与摄像头间的指令及数据的传递。

#### ⑤图像传输线程

图像传输线程对应函数入口为“tskTCPIPTxImage( )”，主要负责向PC机传送编码后的JPEG图像数据。

### 4.2.1.3 中断服务程序介绍

#### ①帧传输触发中断

中断源：GPIO4。

函数入口：ISR\_CCDReady( )。

作用：此中断表示摄像头CCD已经准备好一帧图像，请求进行传输操作。

中断服务程序说明：

关闭上帧数据传输使用的DMA通路；在四个循环缓冲中，选择存储当前帧所需的缓存目标地址；配置当前帧数据传输使用的DMA通路。

#### ②数据传输中断

中断源：EDMA通路9。

函数入口：isr\_CaptureFinished( )。

作用：此中断表示已经将采集到的图像数据送至目标缓存。

中断服务程序说明：

清除中断标志；关闭当前帧数据传输使用的DMA通路；为编码配置图像数据缓存地址；发送编码信号。

全分辨率连续模式下，中断服务程序只负责发送第一帧图像数据的编码信号量，后续编码信号量的发送在编码数据传输完毕后发送。

全分辨率触发(抓拍)模式下，中断服务程序通过判断当前的触发抓拍状态来发送各触发帧的编码信号量。

③外部触发中断

中断源：GPIO5。

函数入口：isr\_ExternTriggerInt ( )。

作用：此中断表示有外触发抓拍信号产生。

中断服务程序说明：

判断摄像头当前运行状态来向CCD发送触发抓拍指令。

在这个中断函数中通过判断上一帧图像是否传输完毕来决定是否启动采集新一帧图像。

④定时中断

中断源：TIMER2。

函数入口：isr\_TimerInt ( )。

作用：此中断表示设定时间点已经到达。

中断服务程序说明：

判断摄像头当前运行状态来向CCD发送触发抓拍指令；判断定时任务是否结束。

摄像头使用Timer2控制抓拍间隔，每次抓拍间隔到了就会进入该中断函数。中断服务程序通过判断上一帧图像是否传输完毕来决定是否启动采集新一帧图像，直到达到设定的抓拍帧数定时器才暂停工作，计数归零。

⑤串口中断

中断源：GPIO6。

函数入口：isr\_UARTInt ( )。

作用：接收串口数据。

中断服务程序说明：

RS232与RS485共享一个中断源。进入中断服务程序后，需要先判断是哪条通路发

起的中断申请，然后再根据具体需要进行处理。

◎通用定时器中断

中断源：TIMER1。

函数入口：isr\_TimerInt (void)。

作用：无。

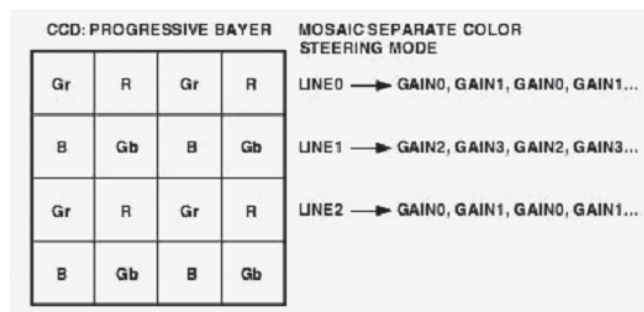
中断服务程序说明：

可根据用户具体需求开启TIMER1定时功能。

#### 4.2.1.4 图像采集格式及存储架构介绍

摄像头的图像采集格式为Bayer点阵格式。用户可以编写相关的处理函数对其进行格式转换。

图像缓存为四帧，轮换使用。图像数据传输中断服务程序中的g\_ui32ImageSrcAddr变量用于反映当前帧图像数据的存放地址。每帧Bayer点阵的格式如下图所示：



图表4-13 Bayer点阵格式

用户可根据自己的需要修改数据存储架构以提升摄像头运行效率。

#### 4.2.2 OK\_IC系列智能摄像头提供的存储器资源

##### 4.2.2.1 SDRAM (128MB)

SDRAM用于存储DSP运行所需要的代码和数据。用户定义的数据段，如不特殊指定存放在片内，则将全部存储于SDRAM中。

SDRAM段中提供了16MB的堆空间，堆标识符定义为“extHeap”，用户可使用MEM\_malloc函数从其中动态申请存储空间。例如：

```
g_pcUserDataBuf = MEM_malloc(extHeap, 0x100000, 256);
if(g_pcUserDataBuf == NULL)
{
return FALSE;
}
```

#### 4.2.2.2 EEPROM (32KB)

EEPROM的前16KB保留给摄像头使用，其余存储空间开放给用户自由使用。用户不可对前16KB进行写操作，否则会破坏摄像头上电初始化必须的一些参数，导致摄像头工作不正常。

#### 4.2.2.3 代码FLASH (1MB)

用于存储出厂演示程序。其中的数据内容不可更改。

#### 4.2.2.4 数据FLASH (8MB)

用于存储用户代码和数据。前4MB空间用于存放用户代码，后4MB空间用于存放用户参数。

### 4.2.3 DEMO工程存储器使用分配

数据和代码段的分配，在工程“DSP/BIOS内存分配”和“IC1200.cmd”文件中完成。

#### ① .user\_data\_ext

片外数据段，分配于SDRAM中。用户可以使用#pragma DATA\_SECTION来从SDRAM中申请空间。例如：

```
#pragma DATA_SECTION(stream, ".user_data_ext ")
unsigned char stream[1024*1024];
```

#### ② .InternalCode

片内代码段。如果需要将某个函数放到片内RAM，则需要将该函数分配到.InternalCode段内。例如：

```
#pragma CODE_SECTION(WatchDog, ".InternalCode")
void WatchDog( )
{
OkIC_KickWatchdog( );
return;
```

```
}
```

③ PACKETMEM、MMBUFFER、OBJMEM

NDK使用的数据段。需要保留。

④ jpgenc\_tbl、.jpgdec\_tbl、.user\_data\_int、cache\_align、.jpgtbls

JPEG编码库使用的数据和代码段。需要保留。

### 4.2.4 OK\_IC系列智能摄像头嵌入式编程API函数说明

#### 1、OkIC\_Init

原 型：void OkIC\_Init( )

参 数：无。

返回值：无。

说 明：初始化摄像头硬件，负责CACHE、定时器、IIC模块、GPIO模块等摄像头参数的初始化配置。该函数中调用了OkIC\_Set\_InitParameter函数进行曝光、增益等参数配置。

#### 2、OkIC\_Set\_InitParameter

原 型：void OkIC\_Set\_InitParameter( )

参 数：无。

返回值：无。

说 明：初始化摄像头参数，包括摄像头增益、RGB通道增益、两种模式的曝光时间等。

#### 3、OkIC\_Get\_Hardware\_Version

原 型：uint32 OkIC\_Get\_Hardware\_Version( )

参 数：无。

返回值：硬件版本号。

说 明：获得硬件版本号。

#### 4、OkIC\_Get\_PIO\_Direction

原 型：uint32 OkIC\_Get\_PIO\_Direction( )

参 数：无。

返回值：十个可编程 I/O 对应的输入/输出方向，数据低位对应I/O低位，数据高位对应I/O高位。

说 明：获得十个可编程 I/O 对应的输入/输出方向。

### 5、OkIC\_Get\_Afe\_Pga

原 型：float OkIC\_Get\_Afe\_Pga( )

参 数：无。

返回值：传感器输出信号的增益，单位为dB，范围为0~36dB。

说 明：获得传感器输出信号的增益。

### 6、OkIC\_Get\_FlashSyncStatus

原 型：Uin8 OkIC\_Get\_FlashSyncStatus( )

参 数：无。

返回值：闪光灯同步输出状态，1为同步闪烁使能，0为禁止。

说 明：获得闪光灯同步输出状态。

### 7、OkIC\_Get\_TrgShutter\_Time

原 型：Uin32 OkIC\_Get\_TrgShutter\_Time( )

参 数：无。

返回值：触发模式下的曝光时间。

说 明：获得触发模式下的曝光时间。

### 8、OkIC\_Get\_FConShutter\_Time

原 型：Uin32 OkIC\_Get\_FConShutter\_Time( )

参 数：无

返回值：全分辨率连续模式下的曝光时间。

说 明：获得全分辨率连续模式下的曝光时间。

### 9、OkIC\_GAMA

原 型：void OkIC\_GAMA(float fLUTGama)

参 数：fLUTGama，GAMA校正系数。

返回值：无。

说 明：设定摄像头GAMA校正系数，通过LUTGama变量进行GAMA校正系数的设定。

### 10、OkIC\_Get\_DSPFirmware\_Version

原 型：Uin32 OkIC\_Get\_DSPFirmware\_Version( )

参 数：无。

返回值：DSP固件程序版本号。

说 明：获得DSP固件程序版本号。

### 11、OkIC\_TransBlock

原 型：void OkIC\_TransBlock(Uin32 ui32SrcAddr,  
Uin32 ui32YSrcAddr, Uin32 ui32USrcAddr, Uin32 ui32VSrcAddr,  
Uin32 ui8ImageMode, Uin32 ImageWidth, Uin32 ImageHeight)

参 数：ui32SrcAddr，图像源地址；

ui32YSrcAddr，搬移后图像的Y分量缓存首地址；

ui32USrcAddr，搬移后图像的U分量缓存首地址；

ui32VSrcAddr，搬移后图像的V分量缓存首地址；

ui8ImageMode，图像格式，此处应当为2；

ImageWidth，图像宽度；

ImageHeight，图像高度。

返回值：无。

说 明：将采集到的Bayer图像数据转换为YUV420图像数据，用于JPEG编码输入。

### 12、OkIC\_Reset\_TrgEnableFlag

原 型：void OkIC\_Reset\_TrgEnableFlag( )

参 数：无。

返回值：无。

说 明：复位允许触发抓拍标志。该标志在触发模式下设置，表示上一帧触发图片没有传输完毕，禁止下一次触发。与OkIC\_Get\_TrgEnableFlag( )函数和OkIC\_Set\_TrgEnableFlag( )函数配套使用。

### 13、OkIC\_Set\_TrgEnableFlag

原 型：void OkIC\_Set\_TrgEnableFlag( )

参 数：无。

返回值：无。

说 明：置位允许触发抓拍标志。该标志触发模式下设置，表示上一帧触发图片已经传输完毕，允许下一次触发。与OkIC\_Get\_TrgEnableFlag( )函数和OkIC\_Reset\_TrgEnableFlag( )函数配套使用。

### 14、OkIC\_Save\_Parameter

原 型：void OkIC\_Save\_Parameter( )

参 数：无。

返回值：无。

说明：保存摄像头参数，包括摄像头增益、RGB通路增益、各种采集模式的光时间和GAMA系数。

#### 15、OkIC\_SetRGain

原型：void OkIC\_SetRGain(int fRGain)

参数：fRGain, R色彩增益值，为0-63范围内的整型数据。

返回值：无。

说明：设置R色彩通路增益。

#### 16、OkIC\_SetGGain

原型：void OkIC\_SetGGain(int fGGain)

参数：fGGain, G色彩增益值，为0-63范围内的整型数据。

返回值：无。

说明：设置G色彩通路增益。

#### 17、OkIC\_SetBGain

原型：void OkIC\_SetBGain(int fBGain)

参数：fBGain, B色彩增益值，为0-63范围内的整型数据。

返回值：无。

说明：设置B色彩通路增益。

#### 18、OkIC\_GetRGain

原型：int OkIC\_GetRGain( )

参数：无。

返回值：R色彩通路增益值，为0-63范围内的整型数据。

说明：获取R色彩通路的增益值。

#### 19、OkIC\_GetGGain

原型：int OkIC\_GetGGain( )

参数：无。

返回值：G色彩通路增益值，为0-63范围内的整型数据。

说明：获取G色彩通路的增益值。

#### 20、OkIC\_GetBGain

原型：int OkIC\_GetBGain( )

参数：无。

返回值：B色彩通路增益值，为0-63范围内的整型数据。

说明：获取B色彩通路增益值。

#### 21、OkIC\_KickWatchdog

原型：void OkIC\_KickWatchdog( )

参数：无。

返回值：无。

说明：喂狗操作。看门狗上电默认为关闭状态，使能后此函数生效。

#### 22、OkIC\_Waitusec

原型：void OkIC\_Waitusec(Uint32 ui32Delay)

参数：ui32Delay, 延迟计数，单位为微秒。

返回值：无。

说明：时间延迟函数。

#### 23、OkIC\_RS232\_Open

原型：void OkIC\_RS232\_Open( )

参数：无。

返回值：无。

说明：RS232串口使能。缓存大小为512KB。

#### 24、OkIC\_RS232\_Close

原型：void OkIC\_RS232\_Close( )

参数：无。

返回值：无。

说明：RS232串口关闭操作。

#### 25、OkIC\_RS232\_SetBaudrate

原型：Bool OkIC\_RS232\_SetBaudrate(Uint32 ui32BaudRate)

参数：ui32BaudRate, RS232串口波特率。

返回值：波特率设置结果，1表示设置成功，0表示设置失败。

说明：设置RS232串口波特率（4800、9600、19200、38400、57600、115200）。

#### 26、OkIC\_RS232\_WriteData

原型：void OkIC\_RS232\_WriteData(char \*pcWriteBuf, Uint32 ui32Count)

参数：pcWriteBuf, RS232串口输出数据缓存指针；

ui32Count, RS232串口输出数据量。

返回值：无。



说明：RS232串口写数据块操作。

#### 27、OkIC\_RS232\_ReadBuf

原型：void OkIC\_RS232\_ReadBuf(char \*pcReadBuf, Uint32 ui32Count)

参数：pcReadBuf, RS232串口读取数据缓存指针；  
ui32Count, 数据读取数据量。

返回值：无。

说明：RS232串口读数据块操作。

#### 28、OkIC\_IsRS232RcvReady

原型：Bool OkIC\_IsRS232RcvReady( )

参数：无。

返回值：RS232串口数据接收状态，1表示接收到数据，0表示没有接收到数据。

说明：RS232串口数据接收状态查询。

#### 29、OkIC\_GetRS232\_Recvcnt

原型：Uint32 OkIC\_GetRS232\_Recvcnt( )

参数：无。

返回值：RS232串口接收到的数据量值。

说明：RS232串口接收数据量查询。

#### 30、OkIC\_Read\_CurTemperature

原型：float OkIC\_Read\_CurTemperature( )

参数：无。

返回值：当前温度值。

说明：温度查询函数，返回温度值。

#### 31、OkIC\_RTC\_SetTime

原型：Bool OkIC\_RTC\_SetTime(Uint8 \*pui8Time)

参数：pui8Time, 实时时钟数据指针。

返回值：配置状态，1表示配置成功，0表示配置失败。

说明：配置实时时钟。

#### 32、OkIC\_RTC\_ReadTime

原型：void OkIC\_RTC\_ReadTime(Uint8 \*pui8Time)

参数：pui8Time, 实时时钟数据指针。

返回值：无。

说明：实时时钟读取操作。

#### 33、OkIC\_Save\_MAC\_Address

原型：void OkIC\_Save\_MAC\_Address(Uint8\* pui8MACAdd)

参数：pui8MACAdd, MAC地址指针。

返回值：无。

说明：保存目标摄像头MAC地址到EEPROM中。

#### 34、OkIC\_Read\_MAC\_Address

原型：void OkIC\_Read\_MAC\_Address(Uint8\* pui8MACAdd)

参数：pui8MACAdd, MAC地址指针。

返回值：无。

说明：读取目标摄像头MAC地址。

#### 35、OkIC\_Save\_IP\_Address

原型：void OkIC\_Save\_IP\_Address(Uint32 ui32IPAddr)

参数：ui32IPAddr, IP地址。

返回值：无。

说明：保存目标摄像头IP地址到EEPROM中。

#### 36、OkIC\_Read\_IP\_Address

原型：Uint32 OkIC\_Read\_IP\_Address( )

参数：无。

返回值：目标摄像头的IP地址。

说明：读取目标摄像头IP地址，返回摄像头IP地址数据。

#### 37、OkIC\_EEPROM\_Write

原型：Bool OkIC\_EEPROM\_Write(Uint16 ui16SubAddress,  
Uint8\* pui8DataWriteBuff, Uint16 ui16WriteByteNumber)

参数：ui16SubAddress, EEPROM的物理地址；  
pui8DataWriteBuff, 输出数据缓冲区指针；  
ui16WriteByteNumber, 写入数据字节数。

返回值：写入状态，1表示写入成功，0表示写入失败。

说明：用户可以通过此函数向EEPROM的后16KB数据空间中写入数据。

例程：

向EEPROM的16K字节地址处烧写100字节数据，100字节数据存储于DSP

SDRAM的0x82000000地址处。

```
OkIC_EEPROM_Write(0x4000, (UInt8*)0x82000000, 100);
```

### 38、OkIC\_EEPROM\_Read

原 型: Bool OkIC\_EEPROM\_Read(UInt16 ui16SubAddress,  
 UInt8\* puiDataReadBuff, UInt16 ui16ReadByteNumber)

参 数: ui16SubAddress, EEPROM的物理地址;  
 puiDataReadBuff, 数据输出缓冲区指针;  
 ui16ReadByteNumber, 读取数据字节数。

返回值: 读取状态, 1表示读取成功, 0表示读取失败。

说 明: 用户可以通过此函数从EEPROM的后16KB数据空间中读取数据。

例 程:

从EEPROM的16K字节地址处, 读取100字节数据, 将100字节数据存储于  
 DSP SDRAM的0x82000000地址处。

```
OkIC_EEPROM_Read(0x4000, (UInt8*)0x82000000, 100);
```

### 39、OkIC\_Enable\_TrgFlashlamp\_Syn\_Out

原 型: void OkIC\_Enable\_TrgFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 触发模式下, 闪光灯同步输出使能。

### 40、OkIC\_Enable\_FconFlashlamp\_Syn\_Out

原 型: void OkIC\_Enable\_FconFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 连续模式下, 闪光灯同步输出使能。

### 41、OkIC\_Disable\_TrgFlashlamp\_Syn\_Out

原 型: void OkIC\_Disable\_TrgFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 禁止触发模式闪光灯同步输出。

### 42、OkIC\_Disable\_FconFlashlamp\_Syn\_Out

原 型: void OkIC\_Disable\_FconFlashlamp\_Syn\_Out( )

参 数: 无。

返回值: 无。

说 明: 禁止连续模式闪光灯同步输出。

### 43、OkIC\_Set\_FlashSyncSignalTime

原 型: void okIC\_Set\_FlashSyncSignalTime(UInt32 ui32FlashSyncSignalTime\_us)

参 数: ui32FlashSyncSignalTime\_us, 闪光灯同步信号保持时间, 单位为微秒。

返回值: 无。

说 明: 设置闪光灯同步输出信号(高电平)保持时间。

### 44、OkIC\_WhiteBalance

原 型: Bool OkIC\_WhiteBalance( )

参 数: 无。

返回值: 白平衡操作状态, 1表示白平衡操作完成, 0表示还需要继续进行白平衡操作。

说 明: 进行白平衡操作。

### 45、OkIC\_Set\_ImgAcquisition\_Format

原 型: void OkIC\_Set\_ImgAcquisition\_Format(  
 enum OkIC\_ImgAcquisition\_Format eunmImgFormat)

参 数: eunmImgFormat, 图像采集格式。当前摄像头只支持Bayer数据格式。输入  
 值必须为2。

返回值: 无。

说 明: 设置图像采集格式。

### 46、OkIC\_Get\_CurImgFormat

原 型: enum OkIC\_ImgAcquisition\_Format OkIC\_Get\_CurImgFormat( )

参 数: 无。

返回值: 图像格式类型。

说 明: 获取当前摄像头图像采集格式。

### 47、OkIC\_Get\_CurRunMode

原 型: enum OkIC\_SensorRunMode OkIC\_Get\_CurRunMode( )

参 数: 无。

返回值: 摄像头运行模式类型, 0表示触发模式, 1表示连续模式。

说 明: 获取摄像头当前运行模式。



#### 48、OkIC\_Mode\_Switch\_Ctrl

原 型: void OkIC\_Mode\_Switch\_Ctrl(  
enum OkIC\_SensorRunMode enumRunMode)。

参 数: enumRunMode, 摄像头运行模式, 0为触发模式, 1为连续模式。

返回值: 无。

说 明: 切换摄像头运行模式。

#### 49、OkIC\_Get\_TriggerEnableFlag

原 型: Bool OkIC\_Get\_TriggerEnableFlag( )

参 数: 无。

返回值: 触发允许标志, 1表示允许触发, 0表示禁止触发。

说 明: 获取摄像头触发允许标志。

#### 50、OkIC\_Set\_AFE\_PGA

原 型: void OkIC\_Set\_AFE\_PGA(float fPgaValue\_db)

参 数: fPgaValue\_db, 摄像头增益值, 为0-36之间的float型数据。

返回值: 无。

说 明: 设置摄像头增益。

#### 51、OkIC\_Grab\_Single\_Frame

原 型: void OkIC\_Grab\_Single\_Frame( )

参 数: 无。

返回值: 无。

说 明: 软触发抓拍图像。外触发源以外的其他触发源可以调用此函数实现触发抓拍。

#### 52、OkIC\_Set\_TriggerShutter\_Time

原 型: Bool OkIC\_Set\_TriggerShutter\_Time(UINT32 ui32TrgShutterTime\_us)

参 数: ui32TrgShutterTime\_us, 触发模式曝光时间。

返回值: 配置状态, 1表示配置完成, 0表示配置未完成。

说 明: 设置触发模式的曝光时间。

#### 53、OkIC\_Set\_FConShutter\_Time

原 型: Bool OkIC\_Set\_FConShutter\_Time(UINT32 ui32FConShutterTime\_us)

参 数: ui32FConShutterTime\_us, 连续模式曝光时间。

返回值: 配置状态, 1表示配置完成, 0表示配置未完成。

说 明: 设置连续模式的曝光时间。

#### 54、OkIC\_Open\_Imagechannel

原 型: void OkIC\_Open\_Imagechannel( )

参 数: 无。

返回值: 无。

说 明: 开启图像采集通道, 开启后摄像头可进行图像采集。

#### 55、OkIC\_Close\_Imagechannel

原 型: void OkIC\_Close\_Imagechannel( )

参 数: 无。

返回值: 无。

说 明: 关闭图像采集通道。

#### 56、OkIC\_Set\_WBCof\_Value

原 型: void OkIC\_Set\_WBCof\_Value (float fRGain, float fGGain, float fBGain)

参 数: fRGain, R路增益, 为0-64之间的float型数据;

fGGain, G路增益, 为0-64之间的float型数据;

fBGain, B路增益, 为0-64之间的float型数据。

返回值: 无。

说 明: 调整白平衡参数。增益值设置, 请参照后文“白平衡增益设置范围说明”相关内容。

#### 57、OkIC\_Set\_Image\_LUT10bit

原 型: void OkIC\_Set\_Image\_LUT10bit(UINT8\* pui8ImgLUTBuf)

参 数: pui8ImgLUTBuf, LUT数据指针, 此数据应当为1024个0-255大小范围的整型数据。

返回值: 无。

说 明: 将输入的LUT数据存储于EEPROM中, 并按照查找表内容使能LUT。

#### 58、OkIC\_Disable\_Image\_LUT

原 型: void OkIC\_Disable\_Image\_LUT (Bool blsSetEEPROM)。

参 数: blsSetEEPROM, LUT数据恢复标识。

返回值: 无。

说 明: 禁止LUT功能。若LUT数据恢复标识为1, 则改写EEPROM中的数据到出厂状态。

**59, OkIC\_Enable\_Image\_LUT**

原 型: void OkIC\_Enable\_Image\_LUT( )。

参 数: 无。

返回值: 无。

说 明: 使能摄像头LUT功能。EEPROM中的LUT数据将被启用。

**60, OkIC\_Resume\_Image\_LUT**

原 型: void OkIC\_Resume\_Image\_LUT( )。

参 数: 无。

返回值: 无。

说 明: 恢复EEPROM中的LUT为出厂状态。出厂默认LUT状态为四段直线，五个拐点分别为：(0, 0) , (255, 84) , (512, 151) , (768, 210) , (1023, 255) 。

**61, OkIC\_Set\_MIBlockParam**

原 型: Bool OkIC\_Set\_MIBlockParam(Uint32 ui32ID, Uint32 ui32StartX, Uint32 ui32StartY, Uint32 ui32Height, Uint32 ui32Width)。

参 数: ui32ID, 区域测光区块ID, 范围1-5;  
ui32StartX, 区域测光范围左上角点X坐标;  
ui32StartY, 区域测光范围左上角点Y坐标;  
ui32Height, 区域测光范围高度;  
ui32Width, 区域测光宽度。

返回值: 设置是否成功, 1表示设置成功, 0表示设置失败。

说 明: 设置区域测光范围。注意在使用过程中要保证区域测光范围不要超过图像有效范围。

**62, OkIC\_Get\_MIBlockYRGBMean**

原 型: Bool OkIC\_Get\_MIBlockYRGBMean(Uint32 ui32ID, Uint32 ui32MeanArray[4])。

参 数: ui32ID, 区域测光区块ID;  
ui32MeanArray, 测光结果反馈缓存, 内容依次为Y均值, R均值, G均值, B均值。

返回值: 数据读取结果, 1表示读取成功, 0表示读取失败。

说 明: 返回区域测光结果。

**63, OkIC\_Set\_Pio\_Direction**

原 型: void OkIC\_Set\_Pio\_Direction(Uint32 ui32PioDirList)

参 数: ui32PioDirList, I/O方向变量, 0为输入, 1为输出, I/O方向变量低位对应I/O低位。

返回值: 无。

说 明: 设置10个I/O数据流方向。

**64, OkIC\_Set\_Pio\_Outvalue**

原 型: void OkIC\_Set\_Pio\_Outvalue(Uint32 ui32PioOutvalueList)

参 数: ui32PioOutvalueList, I/O管脚输出值, I/O管脚输出值低位对应I/O低位。

返回值: 无。

说 明: 设置10个I/O管脚上的输出值。

**65, OkIC\_GetPIOIntTime**

原 型: Uint32 OkIC\_GetPIOIntTime(Uint32 ui32PioID)

参 数: ui32PioID, I/O管脚ID, 范围为0-9, 0-9分别对应I/O的1-10号管脚。

返回值: 中断产生时间。

说 明: 如果允许中断产生, 则在中断发生后, 可通过此函数查询对应I/O的中断时间。中断时间产生于系统计数器, 返回值为32位计数值。系统计数器跳变时间为1微秒。

**66, OkIC\_Flash\_Burn**

原 型: Bool OkIC\_Flash\_Burn(Uint8\* pui8Source, Uint32 ui32DestAddr, Uint32 ui32Length)

参 数: pui8Source, 输出数据缓存指针;  
ui32DestAddr, 输出数据的FLASH目标地址, 此处地址为FLASH的物理地址, 不是DSP中的相对地址;  
ui32Length, 输出数据的字节数。

返回值: 写入状态, 1表示写入成功, 0表示写入失败。

说 明: 用户可通过此函数向FLASH的后4MB数据空间中写入数据。

例 程:

向FLASH的4M字节地址处烧写3000字节数据, 3000字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIC_Flash_Burn( (UInt8*)0x82000000, 0x400000, 3000 );
```

#### 67、OkIC\_Flash\_Read

原 型: Bool OkIC\_Flash\_Read(UInt8\* pui8DataBuf,  
 UInt32 ui32FlashAddr, UInt32 ui32Length)

参 数: pui8DataBuf, 读取数据缓存指针;  
 ui32FlashAddr, 读取数据的FLASH目标地址, 此处地址为FLASH的物理地址, 不是DSP中的相对地址;  
 ui32Length, 读取数据字节数。

返回值: 读取状态, 1表示读取成功, 0表示读取失败。

说 明: 用户可通过此函数从FLASH的后4MB空间中读取数据。

例 程:

从FLASH的4M字节地址处读取3000字节数据, 将3000字节数据存储于DSP SDRAM的0x82000000地址处。

```
OkIC_Flash_Read( (UInt8*)0x82000000, 0x400000, 3000 );
```

#### 68、OkIC\_RS485\_Open

原 型: void OkIC\_RS485\_Open( )

参 数: 无。

返回值: 无。

说 明: RS485串口使能。

#### 69、OkIC\_RS485\_Close

原 型: void OkIC\_RS485\_Close( )

参 数: 无。

返回值: 无。

说 明: RS485串口关闭。

#### 70、OkIC\_RS485\_SetBaudrate

原 型: Bool OkIC\_RS485\_SetBaudrate(UInt32 ui32BaudRate)

参 数: ui32BaudRate, RS485串口波特率。

返回值: 波特率设置结果, 1表示设置成功, 0表示设置失败。

说 明: 设置RS485波特率 (4800、9600、19200、38400、57600、115200)。

#### 71、OkIC\_RS485\_WriteData

原 型: void OkIC\_RS485\_WriteData(char \*pcWriteBuf, UInt32 ui32Count)

参 数: pcWriteBuf, RS485串口输出数据缓存指针;

ui32Count, RS485串口输出数据量。

返回值: 无。

说 明: RS485串口写数据块操作。

#### 72、OkIC\_RS485\_ReadBuf

原 型: void OkIC\_RS485\_ReadBuf(char \*pcReadBuf, UInt32 ui32Count)

参 数: pcReadBuf, RS485串口读取数据缓存指针;

ui32Count, RS485串口读取数据量。

返回值: 无。

说 明: RS485串口读数据块操作。

#### 73、OkIC\_IsRS485RcvReady

原 型: Bool OkIC\_IsRS485RcvReady( )

参 数: 无。

返回值: RS485串口数据接收状态, 1表示接收到数据, 0表示没有接收到数据。

说 明: RS485串口数据接收状态查询。

#### 74、OkIC\_GetRS485\_Recvcnt

原 型: UInt32 OkIC\_GetRS485\_Recvcnt( )

参 数: 无。

返回值: RS485串口接收到的数据量值。

说 明: RS485串口接收数据量查询。

#### 75、OkIC\_GetTickCount

原 型: UInt32 OkIC\_GetTickCount( )

参 数: 无。

返回值: 系统计数数值。

说 明: 系统计数器查询函数, 返回一个32位的系统计数值。系统计数器每间隔1微秒跳变一次。

#### 76、OkIC\_Reset

原 型: void OkIC\_Reset( )

参 数: 无。

返回值: 无。

说 明: DSP系统复位。

**77、OkIC\_Watchdog\_Open**

原 型: void OkIC\_Watchdog\_Open( )

参 数: 无。

返回值: 无。

说 明: 看门狗使能。

**78、OkIC\_Watchdog\_Close**

原 型: void OkIC\_Watchdog\_Close( )

参 数: 无。

返回值: 无。

说 明: 关闭看门狗。

**79、OkIC\_Set\_WatchdogTimer**

原 型: void OkIC\_Set\_WatchdogTimer(Uint32 ui32WDTTime\_S)

参 数: ui32WDTTime\_S, 看门狗复位时间, 单位为秒, 设定范围为1-255。

返回值: 无。

说 明: 设定看门狗复位时间。

**80、OkIC\_JPEGENCINIT**

原 型: void OkIC\_JPEGENCINIT( )

参 数: 无。

返回值: 无。

说 明: JPEG编码库初始化。如使用编码库, 则需要在main( )函数中对其初始化。

**81、OkIC\_MoveImageData\_and\_wait**

原 型: void OkIC\_MoveImageData\_and\_wait(Uint32 SrcDataAddr,  
 Uint32 DstDataAddr, Uint8 ImageFormat, Uint16 cImageWidth,  
 Uint16 cImageHeight)

参 数: SrcDataAddr, 数据源地址;

DstDataAddr, 数据目的地址;

ImageFormat, 图像格式, 此处应当为2;

cImageWidth, 图像宽度;

cImageHeight, 图像高度。

返回值: 无。

说 明: 将循环缓冲中的数据搬运到指定目标地址。该函数使用32位QDMA实现

其功能。

**82、OkIC\_JPGENC\_Parameter\_Config**

原 型: void OkIC\_JPGENC\_Parameter\_Config(Uint32 ui32YUVFormat,  
 Uint32 ui32quality, Uint32 ui32HResolution, Uint32 ui32VResolution)

参 数: ui32YUVFormat, 图像格式, 此处为2;

ui32quality, JPEG压缩因子, 目前只支30-95; ;

ui32HResolution, 编码图像宽度;

ui32VResolution, 编码图像高度。

返回值: 无。

说 明: 设置JPEG参数。

**83、OkIC\_JPGEncoder**

原 型: void OkIC\_JPGEncoder( Uint8\* pui8YBuf, Uint8\* pui8CrBuf,  
 Uint8\* pui8CbBuf, Uint8\* pui8JpgOutBuf[] )

参 数: pui8YBuf, 编码的YUV420数据Y分量数据指针;

pui8CrBuf, 编码的YUV420数据U分量数据指针;

pui8CbBuf, 编码的YUV420数据V分量数据指针;

pui8JpgOutBuf[], 输出数据指针数组, 第一个指针地址返回JPEG编码图像的大小, 第二个指针地址返回JPEG编码的图像数据。

返回值: 无。

说 明: JPEG编码操作。输入为YUV420格式的图像数据, 输出为JPEG编码数据。

**84、OkIC\_GetJPEGCompressQuality**

原 型: int OkIC\_GetJPEGCompressQuality(Uint8 RunMode)

参 数: RunMode, 摄像头运行模式。

返回值: 压缩因子。

说 明: 查询当前的JPEG编码压缩因子。

**85、OkIC\_SetSnapIntervaltime**

原 型: void OkIC\_SetSnapIntervaltime(Uint32 Intervaltime)

参 数: Intervaltime, 抓拍间隔时间, 单位为毫秒。

返回值: 无。

说 明: 如使用定时器进行间隔抓拍, 则使用此函数设定抓拍间隔。抓拍间隔默认为1秒。

#### 86、OkIC\_GetSnapIntervaltime

原 型: Uint32 OkIC\_GetSnapIntervaltime( )

参 数: 无。

返回值: 抓拍间隔。

说 明: 如使用定时器进行间隔抓拍, 则使用此函数查询抓拍间隔。

#### 87、OkIC\_SetSnap

原 型: void OkIC\_SetSnap( )

参 数: 无。

返回值: 无。

说 明: 启动定时抓拍操作。

#### 88、OkIC\_StopTimeSnap

原 型: void OkIC\_StopTimeSnap( )

参 数: 无。

返回值: 无。

说 明: 如使用定时器进行间隔抓拍, 则使用此函数停止定时抓拍操作。

#### 89、OkIC\_EDMAClose

原 型: void OkIC\_EDMAClose( )

参 数: 无。

返回值: 无。

说 明: 关闭VP的EDMA数据搬移通路。

#### 90、OkIC\_EDMAOpen

原 型: void OkIC\_EDMAOpen(Uint8 ui8ImageMode, Uint32 ImgDstAddr)

参 数: ui8ImageMode, 图像格式, 此处为2;  
ImgDstAddr, 目标地址, 必须四字节对齐。

返回值: 无。

说 明: 开启VP的EDMA数据搬移通路。

#### 91、OkIC\_Update

原 型: Bool OkIC\_Update(Uint8 \*pBurndata, Uint16 RebootFlag)

参 数: pBurndata, 烧写数据缓存指针;

RebootFlag, 烧写标识, 为1则烧写数据, 为0则擦除代码空间。

返回值: 烧写状态, 1表示烧写成功, 0表示烧写失败。

说 明: 用于远程更新模块, 向代码8M串行FLASH的前4MB空间烧写数据。

#### 92、OkIC\_SetTimerCounter

原 型: void OkIC\_SetTimerCounter(Uint32 Counter)。

参 数: Counter, 通用定时器定时周期。

返回值: 无。

说 明: 用于75MHz通用定时器定时周期设置。

#### 93、OkIC\_TimerStart

原 型: void OkIC\_TimerStart( )。

参 数: 无。

返回值: 无。

说 明: 启动通用定时器。启动后, 通用定时器重复运行, 不自动停止。

#### 94、OkIC\_TimerStop( )

原 型: void OkIC\_TimerStop( )。

参 数: 无。

返回值: 无。

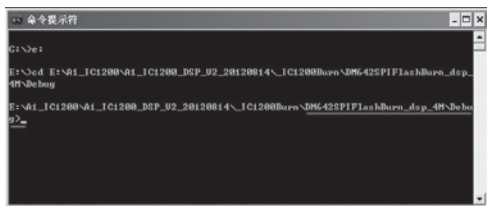
说 明: 停止通用定时器。

### 4.2.5 DSP程序固化说明

步骤1: 编译预加载的“XXX”工程文件, 在工程文件夹中的“Debug”文件夹下找到生成“XXX.out”文件。

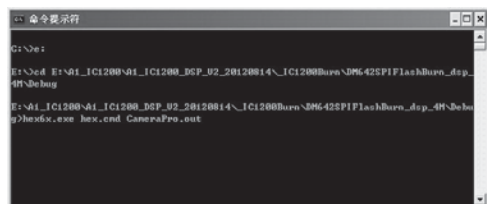
步骤2: 将“XXX.out”文件复制到烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹下。

步骤3: 开启WINDOWS命令提示符。将文件目录设置到烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹。

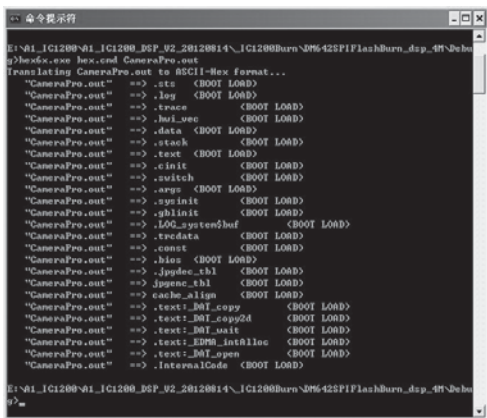


图表4-14

步骤4: 键入命令“hex6x.exe hex.cmd XXX.out” (见图表4-15), 回车执行命令, 并等待程序执行完成, 出现图表4-16所示的界面。其中“XXX.out”是目标烧写文件。

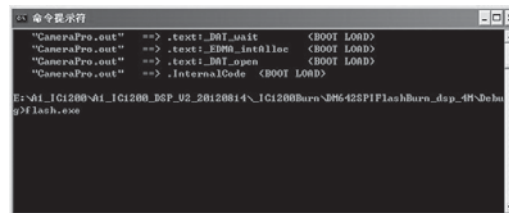


图表4-15



图表4-16

步骤5: 键入命令“flash.exe”, 回车执行命令, 并等待程序执行完成。

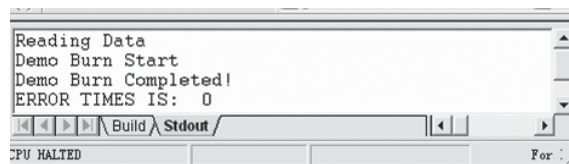


图表4-17

此时, 在烧写工程“DM642SPIFlashBurn\_dsp\_4M”文件夹中的“Debug”文件夹下会生成“flashburn.bin”文件。

步骤6: 使用仿真器连接待烧写的OK\_IC系列摄像头, 在CCS中打开“DM642SPIFlashBurn\_dsp\_4M”文件夹下的“DM642FlashBurn.pjt”烧写工程, 并加载程序, 运行。

步骤7: 等待CCS中显示“Demo Burn Completed!”提示后 (见图表4-18), 说明已经完成烧写。断开摄像头的LINK板, 对摄像头重新上电后即可执行刚才烧入摄像头的“XXX.out”文件。



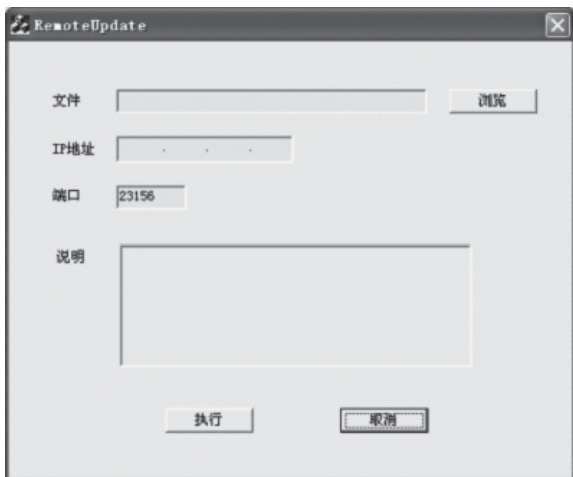
图表4-18

#### 4.2.6 DSP程序远程更新说明

步骤1: 将JHI\_IC1200\_Updatasoft文件夹及其中文件 (包括RemoteUpdate.exe, HEX6X.EXE, hex.cmd, flash.exe) 存储于计算机根目录下。

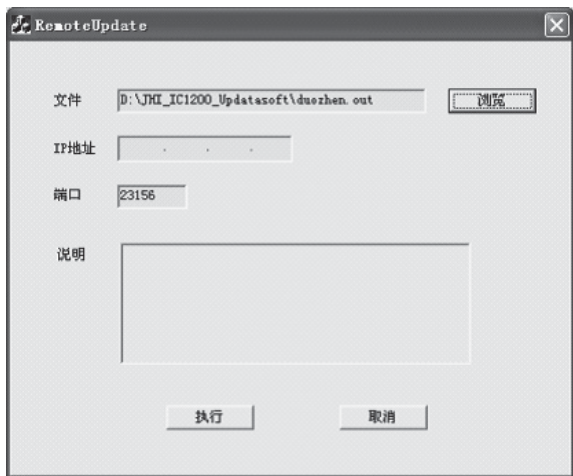
步骤2: 将CCS生成的“XXX.out”文件放入JHI\_IC1200\_Updatasoft文件夹中。

步骤3: 运行RemoteUpdate.exe。



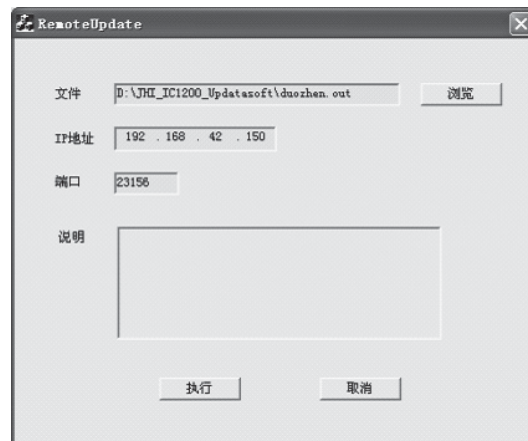
图表4-19

步骤4: 点击“浏览”，选择JHI\_IC1200\_Updatasoft文件夹中的“XXX.out”文件。



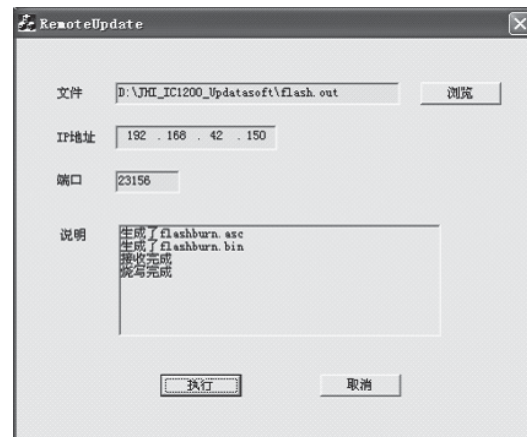
图表4-20

步骤5: 输入IP地址和对应的端口号（端口号在DEMO中默认为23156）。



图表4-21

步骤6: 点击执行开始更新程序。等待下图所示的四个状态执行完毕，即完成了程序固化，DSP将重启。



图表4-22



注意事项:

1. 如果最后一步出现了烧写失败的情况, 则程序将擦除用户程序空间。摄像头重新上电15秒后, 会进入DEMO程序。用户可在DEMO程序基础上重新烧写。
2. JHL\_IC1200\_Updatesoft文件夹应存储于计算硬盘根目录下。
3. 如出现无法打开文件的情况, 请尝试将要烧入摄像头的“XXX.out”文件名改为英文字符, 并缩短文件名称。

#### 4.2.7 用户需了解的其他内容

##### 4.2.7.1 关于开发工具的说明

摄像头DEMO工程基于TI CCS2.20.18版本集成开发环境, 进行开发与测试。

##### 4.2.7.2 EDMA通道的使用情况

当前摄像头中使用了VPO端口对应的DMA通道, 占用了EDMA中断标志TCC1至TCC9和TCC14至TCC15。

##### 4.2.7.3 TCP/IP 协议应用编程说明

DEMO程序演示了如何在摄像头平台上进行TCP/IP编程。网络运行环境初始化和服务线程的建立, 用户可仔细阅读DEMO程序中的SETUPTCPINET.c文件中提供的例程, 同时参阅TI相关文档。在用户建立的服务线程中, 与上位机的通讯程序可由标准的Socket套接字编程来实现。DEMO提供的网络服务函数的工作模式均为迭代服务器模型, 因此要求上位机端的网络为客户端模型。这种结构对于一个利用多台摄像头组网的网络结构来说是非常有利的, 具体数据交互过程请用户参见摄像头端和上位机端的例程。

##### 4.2.7.4 RTC数据格式说明

Addr									Function/Range BCD Format	
	D7	D6	D5	D4	D3	D2	D1	D0		
00h	0.1 Seconds				0.01 Seconds				Seconds	00-99
01h	ST	10 Seconds			Seconds				Seconds	00-59
02h	0	10 Minutes			Minutes				Minutes	00-59
03h	CEB	CB	10 Hours		Hours (24 Hour Format)				Century/ Hours	0-1/00-23
04h	0	0	0	0	0	Day of Week			Day	01-7
05h	0	0	10 Date		Date: Day of Month				Date	01-31
06h	0	0	0	10M	Month				Month	01-12
07h	10 Years				Year				Year	00-99

图表4-23 OK\_IC系列智能摄像头RTC寄存器映射表

PC端RTC配置代码示例:

```
cmdbuf[0] = 0x18;
cmdbuf[1] = 0x00;
*(UINT*)(cmdbuf+2) = 0x07+1;
cmdbuf[6] = m_RTCConfigDlg.m_iYear%100;
cmdbuf[7] = m_RTCConfigDlg.m_iMonth;
cmdbuf[8] = m_RTCConfigDlg.m_iDay;
cmdbuf[9] = m_RTCConfigDlg.m_iWeek;
cmdbuf[10] = m_RTCConfigDlg.m_iHour;
cmdbuf[11] = m_RTCConfigDlg.m_iMinute;
cmdbuf[12] = m_RTCConfigDlg.m_iSecond;
```

PC端RTC读取代码示例:

```
cmdbuf[0] = 0x19;
cmdbuf[1] = 0x01;
*(UINT*)(cmdbuf+2) = 0x04+1;
*(UINT*)(cmdbuf+6) = 0x00;
*(cmdbuf+10) = GetXorCheck(cmdbuf+6,4);
Camerahour = cRcvBuf[4];
Cameramminute = cRcvBuf[5];
Camerasecond = cRcvBuf[6];
```

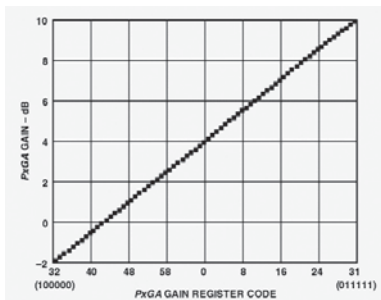
摄像头返回数据内容:

摄像头返回7个字节无符号整型数据, 从低位到高位分别表示年、月、日、星期、时、分、秒。

##### 4.2.7.5 白平衡增益设置范围说明

如图表4-24, 横坐标为写入RGB增益寄存器的值, 十进制表示; 纵坐标为对应的增益值。





图表4-24 白平衡增益寄存器与对应的增益值关系

#### 4.2.7.6 定时抓拍操作的拓展应用

定时抓拍操作利用DSP的定时器实现精准计时，在定时器的中断服务程序中实现定时抓拍操作。用户也可以对定时器的中断服务函数进行修改，进而实现其它精准定时任务。

相关API操作：

```
C64_enableER(0x0400);           //在主函数中开启定时器中断。
OkIC_SetSnapIntervaltime(1000); //设定定时器运行时间，单位毫秒。
OkIC_SetSnap();                 //启动定时器。
OkIC_StopTimeSnap();           //在中断服务程序中关闭定时器。
```

#### 4.2.7.7 串口演示程序说明

DEMO程序中，串口波特率为38400bit/s。串口的最大数据缓存为512字节。向RS232发送数据，如果数据的第一个字节内容为1，则LINK板的LED灯将会被点亮；如果数据的第一个字节内容为2，则LINK板的LED灯将会熄灭；如果数据的第一个字节为其他数据，则回传摄像头接收到的串口数据。

配置串口波特率时，请使用okdspilib.h文件中的宏定义设定串口波特率。

#### 4.2.7.8 关于看门狗操作的说明

系统在PRD周期函数“prd\_WatchDog ( )”中进行喂狗。系统看门狗定时器以秒为基本单位，设置范围为1—255秒。用户可通过修改dm642.cdb文件中的PRD1对象的周期来修改喂狗时间间隔，单位是ms。

DEMO程序中通过“g\_ui8RunFlg”标志变量判断当前摄像头运行状态是否正常。用户可根据自己的需要，调整此标志变量所在位置。

## 第5章 OK系列智能摄像头 上位机应用程序编程指南

### 5.1 接口参数结构定义说明

#### 5.1.1 图像设备信息结构

```
typedef struct _okdevtype
{
    INT32    iBoardTypeCode;    //OK系列图像设备类型码
    INT32    iBoardIdentCode;  //OK系列智能摄像头IP地址
    INT32    iBoardRankCode;   //小类型编码 0或1
    CHAR     szBoardName[116]; //OK系列图像设备名称（例如"OK_IC1200"）
} OKDEVTYPE, *LPOKDEVTYPE;    //共128字节
```

### 5.1.2 网络设备参数结构

```
typedef struct _net_cfg_param
{
    WORD    dwType;           //所要设置的参数: IP地址或MAC地址
    BYTE    bRWFlag;         //读写标志位, =0:读; =1:写
    BYTE    bReset;          //是否写入后重启摄像头, =1: 重启; =0: 不重启
    CHAR    szAddress[120];  //字符串格式的IP地址或MAC地址
                                //IP地址格式为: 192.168.42.234 (10进制)
                                //MAC地址格式为: 60-50-40-30-20-10 (16进制)
                                //两种输入都是本机顺序, 高位在前
    DWORD   dwLowAddress;    // MAC地址的低字 "40-30-20-10"或IP地址
    DWORD   dwHiAddress;     // MAC地址的高字 "60-50"
    DWORD   lRes;            //保留
} NETCFGPARAM, LPNETCFGPARAM; // 此结构体在调用之前须初始化为0
```

### 5.1.3 图像块信息结构

```
typedef struct _blockinfo
{
    SHORT    iType;           //图像类型 (如BK、BM)
    SHORT    iWidth;          //图像宽度
    SHORT    iHeight;         //图像高度
    SHORT    iBitCount;       //象元位数
    SHORT    iFormType;       //图像格式
    SHORT    lBlockStep;      //图像块跨距的低字
    SHORT    iHiStep;         //图像块跨距的高字
    SHORT    lTotal;          //图像总帧数的低字
    SHORT    iHiTotal;        //图像总帧数的高字
    SHORT    iInterval;       //图像帧间隔数
    LPBYTE   lpBits;          //图像数据指针/文件路径名
    LPBYTE   lpExtra;         //额外数据 (如调色板等) 指针
} BLOCKINFO, *LPBLOCKINFO;
```

### 5.1.4 序列文件信息结构

```
typedef struct
{
    SHORT    iType;           //文件类型 (SQ、JP)
    SHORT    iWidth;          //图像宽度
    SHORT    iHeight;         //图像高度
    SHORT    iBitCount;       //象元位数
    SHORT    iFormType;       //图像格式
    SHORT    lBlockStep;      //图像块跨距的低字
    SHORT    iHiStep;         //图像块跨距的高字
    SHORT    lTotal;          //图像总帧数的低字
    SHORT    iHiTotal;        //图像总帧数的高字
    SHORT    iInterval;       //图像帧间隔数
} SEQINFO;
```

### 5.1.5 MJPG信息头结构

```
typedef struct _mjpg_header_info
{
    INT32    lHeadSize;       //结构大小
    INT32    lHorzSize;       //宽度
    INT32    lVertSize;       //高度
    INT32    lBitCount;       //象元位数(缺省值为24)
    INT32    lTotalFrame;     //总帧数
    INT32    lQuality;        //质量因子(JPEG缺省值为75)
    WORD     bHardEncode;     //硬件压缩标志
    WORD     wMakerFlag;
    DWORD    dwVersion;       //版本
    INT32    res0[4];         //保留
    double   dbFrameRate;     //帧率(缺省值为25)
    DWORD    dwGetFrmInfo;    //如果是"GINF"得到当前帧的信息
    void     *lpExtraInfo;    //指向存储额外信息的内存指针
                                //通过okGetCurrImagelInfo返回
```

```

//否则必须为空
INT32 res1[14]; //保留
} MJPGHEADER; //共120字节

```

### 5.1.6 解压的扩展信息结构

```
typedef struct _extra_frame_info
```

```

{
    DWORD dwFlgSkip; //如果是"SKIP" 跳过解码数据, 只得到参
    DWORD dwFrmType; //关键帧类型
    DWORD dwFrmNumber; //视频源帧号
    DWORD dwFrmRatio; //当前帧率
    DWORD dwTimeStamp; //时间戳 (毫秒)
    union {
        struct {
            DWORD Day;5; //日
            DWORD Mon;4; //月
            DWORD Year;14; //年
        };
        DWORD dwDateStamp; //日期戳
    };
    DWORD lQuality; //质量因子
    WORD wres;
    WORD wMakerTag; //JH
    DWORD dwVersion; //版本
    DWORD reserved[5]; //保留
} EXTRAFRMINFO, *LPEXTRAFRMINFO; //用于解压时得到当前帧的扩展信息

```

## 5.2 DLL动态库接口函数

本节只列举了OK系列智能摄像头特有的功能以及最常用的功能函数, 如果需要使用其他函数可以查看《OK系列图像卡编程手册》。

### INT32 WINAPI okGetImageDevice(OKDEVTYPE \*\*lpOkDevInfo, MLONG IParam);

**功能:** 搜寻已正确安装、连接的 OK 系列图像设备的信息, 包括: PCI、PCI-E、PCI-X等总线, USB接口和GigE接口。

**参数:** lpOkDevInfo: 输入指向 OK 系列图像设备信息结构指针的地址。调用返回后该指针指向当前 OK 系列图像设备信息结构数组。图像设备信息结构OKDEVTYPE含有设备的类型码和类型名称, 详见 OKAPI32.H 中的定义。

IParam: 是输入、输出参数的指针。输入时: b0=1, 重新检测; b1=1, 包含网络设备。输出时: 返回Ok Image Manager中是否选择了包含网络设备的选项。

**返回值:** 返回已正确安装的OK系列图像设备数量。如果无, 返回零。

**说明:** 本函数可以查到所有OK系列图像设备。当Ok Image Manager中选中了包含网络设备的选项或者IParam指定搜索时包含网络设备的时候返回包含网络设备在内的所有OK设备的总数; 如果没有指定搜索时包含网络设备, 则函数返回不包含网络设备在内的设备总数。如仅需要获得当前摄像头数, 可设 lpOkInfo=NULL, IParam=0。

**相关函数:** okGetSlotBoard, okGetBoardIndex, okGetTypeCode, okGetBoardName。

### HANDLE WINAPI okOpenBoard(MLONG \*iIndex);

**功能:** 打开指定智能摄像头。从驱动程序自动生成的初始化文件 (在 WINDOWS 目录下的OKADRV.INI) 中读出上次关闭前设置的参数 (如是首次则用系统缺省值) 进行初始化, 并获得所需系统设置, 创建该摄像头的引用句柄, 以供各功能函数引用。

**参数:** iIndex: 1. 要打开的摄像头在已安装连接在计算机中的所有OK系列图像设备索引号, 该索引号从0 开始, =-1时, 即打开缺省选择的设备。  
2. 要打开摄像头的类型码, 这时其低三字节为类型码, 最高字节为同种类型摄像头顺序号。  
3. 一个指向IP地址的32位指针 (例如 char ipstring[] = "192.168.1.1" )。  
另外要注意该变量是以地址方式传递的, 原因见下面的说明。

对于初次使用的用户可以简单设其值为 -1，这样就可以通过我们提供的设备管理器来随时改变用户程序默认打开的摄像头，我们推荐用户这样使用（将iIndex赋值为-1）。

**返回值:** 如果调用成功，返回指定摄像头的引用句柄；如果未发现可用摄像头，返回NULL，详细错误码可通过调用 okGetLastError() 获得。

**说明:** 如果要打开的摄像头索引号/类型码是用户在“OK设备管理器”所设置的缺省索引号/类型码（如未设置，缺省值为0），可以设定 iIndex = -1。如iIndex 输入 -1 或类型码，则调用结束后iIndex 被赋值为该摄像头在当前计算机中实际的索引（顺序）号，这就是要传递地址的原因。类型码的定义参见头文件 OKAPI32.H。

**相关函数:** okCloseBoard, okGetLastError。

#### **BOOL WINAPI okCloseBoard(HANDLE hBoard);**

**功能:** 关闭指定的智能摄像头，然后释放该句柄所用资源。

**参数:** hBoard: 指定要关闭摄像头的引用句柄。

**返回值:** 调用成功，返回 1 (TRUE)；调用失败，返回0 (FALSE)。

**说明:** 不再使用某摄像头时，一定要调用该函数关闭该摄像头。

**相关函数:** okOpenBoard。

#### **HANDLE WINAPI okOpenStream(HANDLE hBoard, DWORD IParam);**

**功能:** 打开指定的视频流压缩设备。

**参数:** hBoard: 输入智能摄像头的句柄。

IParam: 保留参数，目前必须设为 0。

**返回值:** 如成功，返回视频流压缩设备句柄，否则返回 0。

**相关函数:** okCloseStream, okCaptureStream, okOpenBoard。

#### **BOOL WINAPI okCloseStream(HANDLE hStream);**

**功能:** 关闭指定的视频流压缩设备。

**参数:** hStream: 输入智能摄像头的句柄。

**返回值:** 调用成功，返回 1 (TRUE)；调用失败，返回0 (FALSE)。

**说明:** 不再进行视频流压缩时调用此函数关闭视频流压缩设备。

**相关函数:** okOpenStream, okCaptureStream, okCloseBoard。

#### **HANDLE WINAPI okCaptureStream(HANDLE hStream, TARGET target, FARPROC lpfnUserProc, MLONG IMiliSeconds);**

**功能:** 开始视频流压缩。

**参数:** hStream: 输入视频流压缩设备句柄。

target: 采集目标可以为 NONE(无)，BUFFER(缓存)，或文件名，目前只可为AVI(MJPEG)文件格式。

lpfnUserProc: 进行视频流压缩时，用户读取视频流压缩数据的回调函数，当用户不使用回调函数时必须设为 NULL。

IMiliSeconds: 指定要采集压缩的时间，以毫秒为单位，如果是0，则为无限长采集压缩，直到通过调用 okStopCaptureStream 停止为止。

**返回值:** 返回0: 失败；返回非0: 成功开始压缩。

**说明:** 如果采集目标是 NONE(无)，或BUFFER(缓存)，则应在一定期限内，或通过回调函数，或直接用函数 okReadStreamData 将已完成的视频流压缩数据读走，否则后面的数据就会将前面的数据覆盖掉。读取视频流压缩数据的回调函数要按如下方式定义：

BOOL CALLBACK lpfnUserProc(HANDLE hStream, LPBYTE lpStreamBuf, long length);

其中 hStream: 输入视频流压缩设备句柄，lpStreamBuf: 指向可存放读取的视频流压缩数据内存区，length: 为要读取的视频流压缩数据长度，以字节为单位。每当采集到一定数量的数据后，就会调用本回调函数，送出已采集到的数据，length 为数据长度。采集停止后，会自动再调用一次本回调函数，这时 length 的值为0，表示采集结束。

**相关函数:** okOpenStream, okCloseStream, WriteStreamProc。

#### **BOOL WINAPI okStopCaptureStream(HANDLE hStream, DWORD dwPause);**

**功能:** 停止视频流压缩。

**参数:** hStream: 输入视频流压缩设备句柄。

dwPause: =0: 停止采集压缩；=1: 暂停压缩；=2: 恢复压缩。

**返回值:** 如果正在采集，则返回视频流压缩过程中已读出的数据总长度。

说明: 无。

相关函数: okOpenStream, okCloseStream。

### INT32 WINAPI okReadStreamData (HANDLE hStream, LPBYTE lpStreamBuf, INT32 lReadSize);

功能: 读取采集到视频流压缩数据。

参数: hStream: 输入视频流压缩设备句柄。

lpStreamBuf: 用户用于保存视频流压缩数据所申请的内存的指针。

lReadSize: 用户所要读取的视频数据的长度, 以字节为单位。

返回值: 返回实际读取的视频流压缩数据长度, 以字节为单位。

说明: 用户在调用视频流压缩数据函数后, 就需要每隔一定时间, 通过本函数把已完成的视频流压缩数据读出来。

相关函数: okOpenStream, okCloseStream, okCaptureStream。

### MLONG WINAPI okSetTargetRect(HANDLE hBoard, TARGET target, LPRECT lpTgtRect);

功能: 设置采集/回显的目标体的窗口 (兴趣区AOI)。视频源目标体 (VIDEO) 的源窗口, 与采集目标体 (VGA屏幕 (SCREEN), 缓存 (BUFFER)) 的目的窗口均需用此函数来设置。

参数: hBoard: 输入摄像头句柄。

target: 要设置的目标体, 可以是源目标体VIDEO, 也可以是目的目标体SCREEN、BUFFER及WINDOWS窗口句柄其中之一。目标体的宏定义参与OKAPI32.H。窗口句柄的坐标使用的是用户区坐标, 而非绝对坐标, 如果从未设置过窗口句柄的坐标或设置使其坐标的left=right, 则采集到窗口句柄时使用与SCREEN相同的位置。

lpTgtRect: 要设置目标体的窗口坐标。如果调用成功且其坐标值不完全正确 (X 宽度坐标要求4字节对齐), 将会被自动调整为正确值。窗口坐标按 WINDOW 习惯, 定义左上为闭坐标, 右下为开坐标, 即设定的图像窗口是以 (left, top) 点为起始, 宽度是right-left, 高度是bottom-top的一块区域。设置SCREEN 的窗口坐标时, 如果是要设定到用户所打开的某一 WINDOWS 窗口, 则要把该窗

口的坐标转换成屏幕绝对坐标值。设置 WINDOW 窗口句柄的坐标时, 直接使用该窗口句柄的用户区坐标。

注意: 所设置目标体窗口的大小可以比当前摄像头所允许的大, 但如果调用采集函数时, 将自动调整窗口为允许的大小。如果想要没有调用采集函数之前, 就保证设置的窗口符合当前摄像头所允许的大小。可以通过调用本函数, 使 lpTgtRect=NULL, 来实现要设置的目标体窗口与摄像头的自动匹配。但当目标体是 WINDOW 窗口句柄时除外, 当 lpTgtRect=NULL时, 意味着清除对 WINDOW 窗口句柄的采集坐标设置, 而使用 SCREEN 设置的坐标采集。

返回值: 如果调用成功, 返回该目标体在当前窗口设置下可以支持的图像帧数, 否则返回0 (FALSE)。

说明: 如果调用前设置lpTgtRect结构中的.right或.bottom为-1, 则意味着要获得该目标体在本系统中当前的窗口坐标设置, 调用返回后, lpTgtRect被填写为指定目标体的当前窗口坐标值。

相关函数: okSetVideoParam、okSetCaptureParam、okSetToWndRect、okOpenBoard、okCloseBoard。

### MLONG WINAPI okSetCaptureParam(HANDLE hBoard, INT32 wParam, MLONG lParam);

功能: 设置并获得采集控制参数 (如采集格式、方式等)。

参数: hBoard: 输入摄像头句柄。

wParam: 指定设置采集项目。智能摄像头所支持的常用项目如下 (其他项目可以查看《OK系列图像卡编程手册》或OKAPI32.H 中的宏定义)。

- CAPTURE\_EXPOSTIME: 设置摄像头的曝光时间, 以微秒为单位, 最小曝光是摄像头的1个行周期。
- CAPTURE\_TRIGCAPTURE: 设置外触发硬件控制采集的方式 (即摄像头的工作模式)。1、lParam的低字=0: 立即采集, 即正常方式 (连续模式); =1: 等到外触发后采集1场图像 (外触发模式)。2、当 lParam的低字不等于零时, lParam的高字中的低字节 =0, 外触发到来后立即采集; =1延迟1幅图像再采集, 以此类推; lParam的高字中的高字节 表示信号源是触发状

态下驱动查询等待时间，=0 为10ms（默认），可以为1~255，对应1~255ms，在序列采集的时候起效，表示驱动等待多久查询一次有没有新的图像来，如果时间间隔长有可能会丢帧，时间间隔短CPU占用率会高。

- CAPTURE\_WHITEBALANCE：设置摄像头白平衡功能。IParam =1：自动白平衡一次。
- CAPTURE\_MISCCONTROL：设置摄像头区域模式。IParam的bit6：=0摄像头全局模式；=1：摄像头局部模式。
- CAPTURE\_SETTEMPER：获得摄像头当前的实际温度。IParam=-2。
- CAPTURE \_RESETALL：重置所有采集项的参数值为系统缺省值，IParam 为0。
- CAPTURE\_SETCOMPPARAM：设置智能摄像头压缩参数。IParam的低字：设置jpg压缩的质量因子（1~100），决定图像的压缩比，该值越大压缩比越小，图像保质越好，0是默认参数，等价于值50。IParam的高字：设置压缩格式，=1为JPG，=2为PNG，=3为TIFF。目前支持的是jpg。

IParam：对应wParam的参数值。意义详见上述。对于有些IParam，他的高字和低字对应了不同的参数设置，应该先把整个参数读出，然后只改变需要修改的高字或低字部分然后再进行参数设置。

**返回值：**如果指定的项目或参数不被支持，返回 -1；如果失败，返回 -2；如果成功，返回该项目之前的参数值。

**说明：**如果 IParam = -1 (GETCURRPARAM)，则仅返回该项目当前的参数值。通过此函数设置的采集参数，在应用程序调用 okCloseBoard 之后自动存入该摄像头的初始化配置文件中，在应用程序调用 okOpenBoard 之后，此函数将从对应摄像头的初始化配置文件中取出采集参数来进行摄像头的初始视频采集参数设置。已设置的参数如果没有改变将一直有效。所以当需要不同的设置时就要重新设置相应的参数。

**相关函数：**okSetVideoParam、okSetTargetRect、okCloseBoard、okOpenBoard。

#### MLONG WINAPI okSetVideoParam(HANDLE hBoard, INT32 wParam, MLONG IParam);

**功能：**设置并获得视频输入信号的调节参数（如对比度、亮度等）。

**参数：**hBoard：输入摄像头句柄。

wParam：指定设置视频项目，智能摄像头所支持的项目如下（其他项目可

以查看《OK系列图像卡编程手册》或OKAPI32.H 中的宏定义）：

- VIDEO\_CONTRAST：增益（对比度）调节，LOWORD(IParam) 为对比度调节值，范围0~255。
- VIDEO\_BRIGHTNESS：黑电平（亮度）调节，LOWORD(IParam) 为亮度调节值，范围0~255。
- VIDEO\_TAGFRMCOUNT：帧计数器开关。LOWORD(IParam)=1：打开帧计数功能；=0：关闭帧计数功能。HIWORD(IParam)=1：计数器清零。
- VIDEO\_SETDATETIME：同步实时时钟。IParam=-1：获得摄像头内部实时时钟。IParam等于其他任何值：将摄像头内部时钟设置成当前计算机系统时间。
- VIDEO\_RESETALL(0)：重置所有项的参数值为系统缺省值，IParam为0。

IParam：对应 wParam 的参数值。意义详见上述。

**返回值：**如果指定的项目或参数不被支持，返回 -1；如果失败则返回 -2。如果成功则返回该项目之前的参数值。

**注意：**如果 IParam = -1 (GETCURRPARAM)，则仅返回该项目当前的参数值。

通过此函数设置的视频输入参数，在应用程序调用 okCloseBoard 之后自动存入该摄像头的初始化配置文件中，在应用程序调用 okOpenBoard 之后，此函数将从对应摄像头的初始化配置文件中取出视频输入参数来进行图像摄像头的初始视频输入参数设置，已设置的参数如果没有改变将一直有效，所以当需要不同的设置时就要重新设置相应的参数。

**相关函数：**okSetCaptureParam、okSetTargetRect、okCloseBoard、okOpenBoard

#### MLONG WINAPI okPutSignalParam(HANDLE hBoard, INT32 wParam, MLONG IParam);

**功能：**设置要输出到其他设备上的信号（触发等）的参数。

**参数：**hBoard：输入摄像头句柄。

wParam：指定要输出的信号选项，智能摄像头所支持的项目如下（其他项目可以查看《OK系列图像卡编程手册》或OKAPI32.H 中的宏定义）：

- PUTSIGNAL\_TRIGGER：触发信号。IParam =1：软件外触发1次。
- PUTSIGNAL\_SETFLASHMODE：设置闪光灯同步。LOWORD(Iparam) = 0：关闭；=1：打开。

IParam：对应 wParam 的参数值。



**返回值:** 如果指定的项目或参数不被支持, 返回 -1。

**相关函数:** okWaitSignalEvent, okSetVideoParam, okSetCaptureParam

**MLONG WINAPI okSetDeviceParam(HANDLE hBoard, INT32 wParam, MLONG iParam);**

**功能:** 设置、获得网络设备的调节参数。

**参数:** hBoard: 输入摄像头句柄。

wParam: 指定设置项目。智能摄像头所支持的常用项目如下 (其他项目可以查看《OK系列图像卡编程手册》或OKAPI32.H中的宏定义)。

- DEVICE\_SETIPADDRESS: 设置或获得网络设备的IP地址。iParam 是 NETCFGPARAM 结构体的指针, 具体定义详见OKAPI32.H中的宏定义。
- DEVICE\_GETMACADDRESS: 获得网络设备的mac地址。iParam 是 NETCFGPARAM 结构体的指针, 具体定义详见OKAPI32.H中的宏定义。

iParam: 对应 wParam 的参数值。意义详见上述。

**返回值:** 如果指定的项目或参数不被支持, 返回 -1; 如果失败则返回 0; 如果成功则返回 1 (TRUE) 或是预期的参数值。

**说明:** NETCFGPARAM结构体的bRWFlag为0的时候是读当前参数的值, 为1的时候是写当前参数的值。

**相关函数:** okSetCaptureParam, okSetVideoParam, okCloseBoard, okOpenBoard, okOpenBoardEx。

**MLONG WINAPI okSaveImageFile(HANDLE hBoard, LPSTR szFileName, MLONG first, TARGET target, MLONG start, MLONG num);**

**功能:** 从源目标体存图像窗口 (RECT) 到硬盘, 等存盘完成后返回。这里的源目标体可以是 VGA 屏幕 (SCREEN)、帧缓存 (BUFFER), 以及带有用户内存 (MEMORY) 地址的 BLOCKINFO指针。

**参数:** hBoard: 输入摄像头句柄。

szFileName: 存盘文件名。

如果存不压缩的文件, 文件名可以是 ".SEQ"、".AVI"、".BMP"、".JPG" 或 ".RAW" 为后缀的文件名, 例如: "AA.SEQ"。

如果存SEQ文件, 但是希望图像数据是JPG压缩的, 可以按如下格式输入文件名 "AA.SEQ,JPG,60"。其中 ",JPG,60" 表示图像数据是JPG压缩的, 60是JPG的质量控制因子, 所存图像的文件名仍为 "AA.SEQ"。

如果存AVI文件, 但是希望图像数据是压缩的, 目前支持 MotionJPEG 和 MPEG4, 如需要存 MJPEG, 则输入文件名后跟 ",MJPG", 缺省质量因子为60, 如需自己指定可再后跟系数, 如 "AA.AVI,MJPG,70"。如果需要存 MPEG4, 输入文件名后跟 ",MPG4", 如 "AA.AVI,MPG4"。

**first:** 对于 SEQ、AVI 文件, 该参数是从序列文件中的第几幅开始存 (起始为0)。对于BMP文件, first=0, 按标准BMP格式存放, 即把源目标体的格式转换为8位 (黑白) 或24位 (彩色) 来存放; first=1, 则按支持扩展格式的BMP存放, 即按源目标体当前的格式存放而不论它是否8位或24位, 注意: 如此时源目标体的格式是16位或32位, 生成的BMP图像文件, 一般的应用软件 (如PaintBrush等) 因不支持这些扩展格式而不能读出。

对于JPG文件, 该参数也是质量控制因子 (1-100)。(但优先于通过文件名字符串输入方法, 只有这里设置为0时, 文件字符串的输入才有效。) =1: 保质最差 (但压缩比最大); =100: 保质最好 (压缩比最小); =0: 为缺省设置, 等价于质量控制因子是50。当源目标体的格式为非8位 (黑白和24位 (彩色)) 时, 该函数将自动转换成这两种格式再进行压缩。

**target:** 源目标体。

**start:** 源目标体开始读取的帧序号 (起始为 0)。

**num:** 要存盘图像幅数, 应该大于 0。

**返回值:** 如果是 ".SEQ" 文件, 返回文件的总长度 (以字节为单位); 如果是 ".BMP"、".JPG" 或 ".RAW" 文件, 返回单个文件的长度 (以字节为单位); 如果失败返回0。

**说明:** 存 SEQ 文件不进行格式转换, 因而文件将按目标体的数据位数格式存放。SEQ 序列图像文件的格式为头 20 字节 (即结构变量 SEQINFO, 详细参见头文件 OKAPI32.H) 是序列图像的格式信息, 紧跟其后按行顺序存放各帧

原始格式图像数据。

如果是“.SEQ”文件，函数将从序列文件的第first幅开始，存num幅图像。如果输入的文件名已存在，原有文件将不会被删除和覆盖。因此，应在存储文件前自行删除原有同名文件才能正确保存文件。

RAW格式的图像文件，是按行顺序存原始格式的图像数据。RAW格式仅限于8位格式。

如果是“.BMP”、“.JPG”或“.RAW”单幅格式文件，并且num=1，则只存一幅单幅格式（如BMP）的图像文件。如果是存某一单幅格式文件且num>1，则存多幅单幅格式（如BMP）的图像文件。多幅的单幅格式（如BMP）图像文件的命名规则如下：（1）如果给定的文件名已含有数字（如：“OK1000.BMP”），则函数以第一个出现的数字串，1000开始顺序加1，来命名各文件名，最多可到“OK9999.BMP”；（2）如果给定的文件名不含有数字（如：“OK.BMP”），则函数在给定的主名最后自动附加三位数字，从000开始顺序加1，来命名之后的各文件名，最多可到“OK999.BMP”。注意：不要只用数字来定义文件名，应至少含有1个字母。

对于“.SEQ”、“.AVI”、“.M2V”等多图像文件，要存储序列格式的图像，包括压缩格式的，如JPEG、MJPEG、MPEG4等，可以通过采用“开始、结束”的头尾结构来提高存入的效率。方法是：①先调用本函数一次，给入的文件名后要加上“.beg”，这时该函数仅创建序列文件，并不实际存图像；②然后每实际存入一幅图像时，调用一次本函数，这时每次给入的文件名后都要加上“.con”，如此连续调用即可实现顺序存图像；③在结束序列存入图像后，必须再调用本函数一次，此时给入的文件名后要加上“.end”，此后程序才生成并关闭文件。

**相关函数：**okLoadImageFile、okCaptureByBuffer。

**HANDLE WINAPI okBeginDecode(HANDLE hBoard, DWORD dwCodeWay, LPBYTE lpData, MLONG lpImageInfo);**

**功能：**启动指定编码方式的解码，并获得编码数据的参数。

**参数：**hBoard：输入摄像头句柄。

dwCodeWay：输入指定的编码方式，参见头文件OKAPI32.H。

lpData：输入含有编码头的编码数据。

lpImageInfo：如果其不为NULL，输出编码数据的原图像参数。结构定义参见头文件OKAPI32.H。

**返回值：**如果成功，返回解码器句柄，否则返回0（NULL）。

**说明：**目前该函数支持JPEG、MJPEG、MPEG、MPEG2、MPEG4、JPEG2000等解码方法。

**相关函数：**okDecodeImage、okEndDecode、okEncodeImage。

**INT32 WINAPI okDecodeImage(HANDLE hCoder, LPBYTE lpData, MLONG \*length, TARGET target, MLONG start);**

**功能：**对给定的编码数据进行解码。

**参数：**hCoder：输入解码器句柄。

lpData：输入存放编码数据的内存指针。

Length：输入编码数据的长度，输出实际用的长度。

target：指定输出目的体，可以是SCREEN、BUFFER，或含有用户内存指针的BLOCKINFO结构变量指针。BLOCKINFO结构的定义参见OKAPI32.H。

Start：指定输出到目的体的第几帧。

**返回值：**如果对已输入的编码数据，解出了一幅图，则返回1，如果还不够一整幅图，则返回0。

**说明：**目前该函数支持JPEG、MJPEG、MPEG、MPEG2、MPEG4、JPEG2000等解码方法。

**相关函数：**okBeginDecode、okEndDecode、okEncodeImage。

**INT32 WINAPI okEndDecode (HANDLE hCoder);**

**功能：**结束解码，并释放解码器资源。

**参数：**hCoder：输入解码器句柄。

**返回值：**如果成功，返回1。

**说明：**目前该函数支持JPEG、MJPEG、MPEG、MPEG2、MPEG4、JPEG2000等解码方法。

**相关函数：**okBeginDecode、okDecodeImage、okEndEncode。



**BOOL WINAPI okGetCurrImageInfo(HANDLE hBoard, MLONG lpImgFrmlInfo, MLONG lSize);**

**功能:** 获得图像文件的额外信息。

**参数:** hBoard : 输入摄像头句柄。

lpImgFrmlInfo: 是EXTRAFRMINFO结构体的指针, 返回的额外的信息放在EXTRAFRMINFO结构体里面, 结构体的定义可参见 OKAPI32.H。  
lSize: 是EXTRAFRMINFO结构体的的大小。

**返回值:** 如果 lpImgFrmlInfo非0, 成功返回1, 不成功返回0。

**说明:** 对于mjpg压缩的avi文件(硬件压缩的或者是驱动是2011年11月17日以后的用ok函数保存的avi文件), 其中会包含额外的压缩该文件时的年月日等信息, 如果需要的话, 可以在解压文件的时候将其读取出来。先用okLoadImageFile函数解压一帧, 然后用okGetCurrImageInfo得到这一帧的额外信息, 用来记录或显示。

### 5.3 其他支持函数

以下函数和ok系列采集卡通用, 不再另写说明文档。具体使用请查看《OK系列图像卡编程手册》。

- 1) INT32 WINAPI okGetLastError();
- 2) MLONG WINAPI okGetBufferSize(HANDLE hBoard, void \*\*lpLinear, MWORD \*dwSize);
- 3) LPVOID WINAPI okGetBufferAddr(HANDLE hBoard, MLONG lNoFrame);
- 4) LPVOID WINAPI okGetTargetInfo(HANDLE hBoard, TARGET target, MLONG lNoFrame, SHORT \* wid, SHORT \* ht, MLONG \*stride);
- 5) INT32 WINAPI okGetTypeCode(HANDLE hBoard, LPSTR lpBoardName);
- 6) INT32 WINAPI okSetToWndRect(HANDLE hBoard, HWND hWnd);
- 7) INT32 WINAPI okReadPixel(HANDLE hBoard, TARGET src, MLONG start, SHORT x, SHORT y);
- 8) INT32 WINAPI okWritePixel(HANDLE hBoard, TARGET tgt, MLONG start, SHORT x, SHORT y, INT32 lValue);
- 9) MLONG WINAPI okSetConvertParam(HANDLE hBoard, INT32 wParam, MLONG lParam);

- 10) INT32 WINAPI okTransferRect(HANDLE hBoard, TARGET dest, MLONG iFirst, TARGET src, MLONG iStart, MLONG lNum);
- 11) INT32 WINAPI okConvertRect(HANDLE hBoard, TARGET dst, MLONG first, TARGET src, MLONG start, MLONG lParam);
- 12) INT32 WINAPI okConvertRectEx(HANDLE hDstBoard, TARGET dst, MLONG first, HANDLE hSrcBoard, TARGET src, MLONG start, MLONG no);
- 13) INT32 WINAPI okReadRect(HANDLE hBoard, TARGET src, MLONG start, LPBYTE lpBuf);
- 14) INT32 WINAPI okWriteRect(HANDLE hBoard, TARGET dst, MLONG start, LPBYTE lpBuf);
- 15) INT32 WINAPI okReadRectEx(HANDLE hBoard, TARGET src, MLONG start, LPBYTE lpBuf, INT32 lParam);
- 16) INT32 WINAPI okWriteRectEx(HANDLE hBoard, TARGET dst, MLONG start, LPBYTE lpBuf, INT32 lParam);
- 17) MLONG WINAPI okLoadImageFile(HANDLE hBoard, LPSTR szFileName, MLONG first, TARGET target, MLONG start, MLONG num);
- 18) BOOL WINAPI okSaveConfigFile(HANDLE hBoard, LPSTR szFileName);
- 19) BOOL WINAPI okLoadConfigFile(HANDLE hBoard, LPSTR szFileName);
- 20) INT32 WINAPI okGetSignalParam(HANDLE hBoard, INT32 wParam);
- 21) INT32 WINAPI okWaitSignalEvent(HANDLE hBoard, INT32 wParam, MLONG lMilliSecond);
- 22) MLONG WINAPI okPutSignalParam(HANDLE hBoard, INT32 wParam, MLONG lParam);
- 23) MLONG WINAPI okSetAllocBuffer(MLONG lSize);
- 24) BOOL WINAPI okSetStaticVxD(INT32 lMode);
- 25) BOOL WINAPI okSetNTDriver(BOOL bCmd);
- 26) SHORT WINAPI okGetSlotBoard(BOARDTYPE \*\*lpOkInfo);
- 27) SHORT WINAPI okGetBoardIndex(CHAR \*szBoardName, SHORT iNo);
- 28) INT32 WINAPI okGetBoardName(SHORT lIndex, LPSTR szBoardName);
- 29) MLONG WINAPI okGetAvailBuffer(void \*\*lpLinear, MWORD \*dwSize);
- 30) MLONG WINAPI okLockBuffer(HANDLE hBoard, MWORD dwSizeByte, void \*\*lpBasLinear);

```

31) BOOL WINAPI okUnlockAllBuffer( void );
32) BOOL WINAPI okLoadInitParam( HANDLE hBoard, SHORT iChannNo);
33) BOOL WINAPI okSaveInitParam( HANDLE hBoard, SHORT iChannNo);
34) INT32 WINAPI okApplyMemBlock(DWORD dwBlockSize,DWORD dwBlockNo);
35) BOOL WINAPI okFreeMemBlock();
36) INT32 WINAPI okGetMemBlock(HANDLE hBoard, DWORD *dwEachSize,
    DWORD *dwBlockNo);
37) BOOL WINAPI okOpenSetParamDlg( HANDLE hBoard, HWND hParentWnd);
38) BOOL WINAPI okOpenSeqCaptureDlg( HANDLE hBoard, HWND hParentWnd);
39) LPDIBINFO WINAPI okOpenReplayDlg( HANDLE hBoard, HWND hWnd,
    TARGET src, MLONG total);
40) HWND WINAPI okOpenReplayDlgEx( HANDLE hBoard, HWND hWnd, TARGET
    src, MLONG total, LPBITMAPINFOHEADER lpbi, LPBYTE lpdib);
41) BOOL WINAPI okSetTextTo(HANDLE hBoard, TARGET target, LPRECT lpRect,
    LOGFONT *lpLogFont, SETTEXTMODE *textMode, LPSTR lpString, INT32
    lLength);
42) INT32 WINAPI okDrawLineTo(HANDLE hBoard, TARGET target, MLONG
    lStart, POINT ptS, POINT ptE, INT32 iForeColor);
43) INT32 WINAPI okDrawEllipsTo(HANDLE hBoard, TARGET target, MLONG
    lStart, LPRECT lpRect, INT32 iForeColor);
44) HDC WINAPI okCreateDCBitmap(HANDLE hBoard, TARGET target, HANDLE
    *hDCBitmap);
45) BOOL WINAPI okMapDCBitmapTo(HANDLE hDCBitmap, MLONG lStart);
46) BOOL WINAPI okFreeDCBitmap(HANDLE hDCBitmap);
47) HANDLE WINAPI okBeginEncode(HANDLE hBoard, DWORD dwCodeWay,
    MLONG lpImagelInfo);
48) INT32 WINAPI okEncodeImage(HANDLE hCoder, TARGET src, MLONG start,
    LPBYTE lpData, MLONG maxlen);
49) INT32 WINAPI okEndEncode(HANDLE hCoder);
50) MLONG WINAPI okSetStreamParam(HANDLE hStream, INT32 wParam, MLONG
    lParam);

```

## 第6章 故障诊断及排除

### 6.1 故障现象：没有图像

#### 1. 确认摄像头是否已经上电（12V 1.5A电源）：

智能摄像头红色指示灯表示电源，供电正常情况下，红色指示灯亮。

如果红色指示灯不亮，请检查电源线缆是否正常通电，或是否有接触不良的情况。

#### 2. 确认摄像头网络是否通畅：

1) 确认网口指示灯是否亮，如果不亮，请检查网线是否连通正常。

2) 确认摄像头IP地址和与其连接的计算机IP地址在同一网段（摄像头IP默认为192.168.42.150）。

3) 在命令提示符中确认是否能ping通摄像头。如果无法ping通，可能有以下几个原因：

- ①网卡没有正常工作，请重新安装网卡驱动，直到提示网卡设备工作正常；
- ②网线没有连通，请更换网线到别的电脑，看是否可以连通局域网（广域）网络；
- ③摄像头没有正常工作，请断电上电后，观察摄像头指示灯状态。

## 第7章 获得支持

您可以通过一下途径获得我们的技术支持：

1. 访问我们的网站www.jhi.com.cn获得在线技术支持的帮助；
2. 其他支持-----通过电话（010-51665596）电子邮件（info@jhi.com.cn）或者传真（010-82629477）联系我们；
3. 现场服务-----我们为客户提供现场调试、定期维修维护、故障排除等现场服务；
4. 产品维修-----嘉恒图像OK系列产品均实行一年免费保质，终身维护的方针。

## 第8章 附录

### 8.1 摄像头的维护

摄像头内置清洁的CCD传感器，用户应尽可能防止灰尘进入“C/CS”安装口，在更换镜头时，“C/CS”安装头应罩上，摄像头在不使用且没有镜头的情况下应盖上摄像头盖。

CCD传感器在清洁时，需特别注意以下各点：

- 1、在清洁前务必切断电源；
- 2、摄像头和传输线如有污垢，应用干的软布擦拭，如污垢较严重，请用中性清洁剂轻轻擦拭，绝不可使用干刷子刷；
- 3、切勿使用带有腐蚀性或研磨性的清洁剂清洁CCD传感器表面和彩色滤光片，应使用沾有酒精或乙烷的棉签来清洁CCD传感器表面，且不要来回擦拭，使用乙烷时需注意通风；
- 4、可用吸耳球吹除浮尘

### 8.2 图表附录

表格1-1 OK系列智能摄像头性能参数表 .....	2
表格1-2 OK系列智能摄像头硬件资源表 .....	3
表格2-1 OK系列智能摄像头36芯数字输入输出接口引脚定义 .....	8
表格2-2 OK系列智能摄像头接口与开发扩展板接口连接关系 .....	10
表格2-3 OK系列智能摄像头开发扩展板（LINK板）接口引脚定义 .....	10

表格2-4 LINK板DSP_JTAG接口引脚定义 .....	12	图表4-3 .....	51
表格2-5 LINK板RS485接口引脚定义 .....	13	图表4-4 .....	52
表格2-6 LINK板RS232接口引脚定义 .....	13	图表4-5 .....	52
表格2-7 LINK板GPIO接口引脚定义 .....	14	图表4-6 .....	53
图表2-1 OK系列智能摄像头的硬件系统方框图 .....	5	图表4-7 .....	53
图表2-2 摄像头后面板外部接口图 .....	6	图表4-8 .....	54
图表2-3 综合接口示意图 .....	6	图表4-9 .....	54
图表2-4 36芯数字输入输出接口示意图 .....	7	图表4-10 .....	55
图表2-5 LINK板示意图 .....	11	图表4-11 OK_IM系列智能摄像头RTC寄存器映射表 .....	56
图表2-6 LINK板DSP_JTAG接口示意图 .....	12	图表4-12 OK_IC系列智能摄像头DEMO工程软件主体架构 .....	58
图表2-7 LINK板RS485接口示意图 .....	13	图表4-13 Bayer点阵格式 .....	61
图表2-8 LINK板RS232接口示意图 .....	14	图表4-14 .....	82
图表2-9 LINK板GPIO接口示意图 .....	15	图表4-15 .....	82
图表3-1 OK系列智能摄像头外观尺寸图 .....	18	图表4-16 .....	82
图表3-2 摄像头后截距 .....	19	图表4-17 .....	83
图表3-3 .....	20	图表4-18 .....	83
图表3-4 .....	20	图表4-19 .....	84
图表3-5 .....	20	图表4-20 .....	84
图表3-6 .....	21	图表4-21 .....	85
图表3-7 .....	21	图表4-22 .....	85
图表3-8 .....	22	图表4-23 OK_IC系列智能摄像头RTC寄存器映射表 .....	86
图表3-9 .....	22	图表4-24 白平衡增益寄存器与对应的增益值关系 .....	88
图表3-10 .....	23		
图表3-11 .....	23		
图表3-12 .....	24		
图表3-13 .....	24		
图表3-14 .....	24		
图表3-15 .....	25		
图表3-16 .....	25		
图表4-1 OK_IM系列智能摄像头DEMO工程软件主体架构 .....	28		
图表4-2 .....	51		

