

感谢您购买英创信息技术有限公司的产品：**ETA9650 CSI接口摄像头模块**。

ETA9650 是一款基于 OmniVision 公司 CMOS 感光元器件 ov9650 的摄像头扩展模块，通过 CSI (COMS Sensor Interface) 接口连接 EM6800V，扩展出一路摄像头，因为 CSI 是一个标准的视频输出接口，视频处理芯片可以直接输出，不需要涉及到 USB 接口摄像头所需的视频压缩芯片以及 USB 接口芯片，所以较市面上普通的 USB 摄像头来说，CSI 接口的摄像头更便宜，配合 ESM6800V 形成了一个低成本的图像处理方案。ETA9650 扩展单元由包括 1 片基于 ov9650 的摄像头和电源转换电路组成，英创公司提供针对 ETA9650 的驱动及应用程序范例。本文将介绍 ETA9650 的使用、各个接口的信号定义等。

本手册详细介绍了 ETA9650 的硬件配置、管脚定义及相关的技术指标，供用户使用时备查。

用户可以访问英创公司网站或直接与英创公司联系以获得 ETA9650 的其他相关资料。

英创信息技术有限公司联系方式如下：

地址：成都市高新区高朋大道 5 号博士创业园 B 座 407# 邮编：610041

联系电话：028-86180660 传真：028-85141028

网址：<http://www.emtronix.com> 电子邮件：support@emtronix.com

注意：本手册的相关技术内容将会不断的完善，请客户适时从公司网站下载最新版本的数据手册，

恕不另行通知。

1. ETA9650简介

ETA9650 是一款基于 OmniVision 公司 CMOS 感光元器件 ov9650 的摄像头扩展模块，支持 130W 像素（最高分辨率 1280×1024）。让需要有摄像头获取图像的应用场合开发的客户能够快速方便的实现硬件设计，该模块可以适用于 ESM6800V 核心板。

为了方便客户在实际设计时灵活固定模块的位置，英创公司将 ETA9650 分为两个部分，主板和转接板，转接板直接插在 ESMARC 通用评估底板上，主板搭载摄像头通过带线和转接板连接，这样主板可以灵活的选择位置和方向。

2. 硬件接口说明

ETA9650 的硬件设计使得用户既能快速方便的对它进行评估，又能很好的融入用户自己的产品设计中。用户对 ETA9650 进行评估时，可以直接将 ETA9650 的转接板部分插入 ESMARC 通用评估底板 ISA 接口上，以方便进行功能评估。在用户自己做应用底板时，ETA9650 的转接板部分可以作为一个“器件”背插在用户的应用底板上，以获得最佳的数据传输性能，而主板部分则可以灵活的固定在需要的位置。我们提供 ETA9650 protel 形式的器件 PCB 封装，以方便用户 Layout。图 1 是 ETA9650 的外形图。

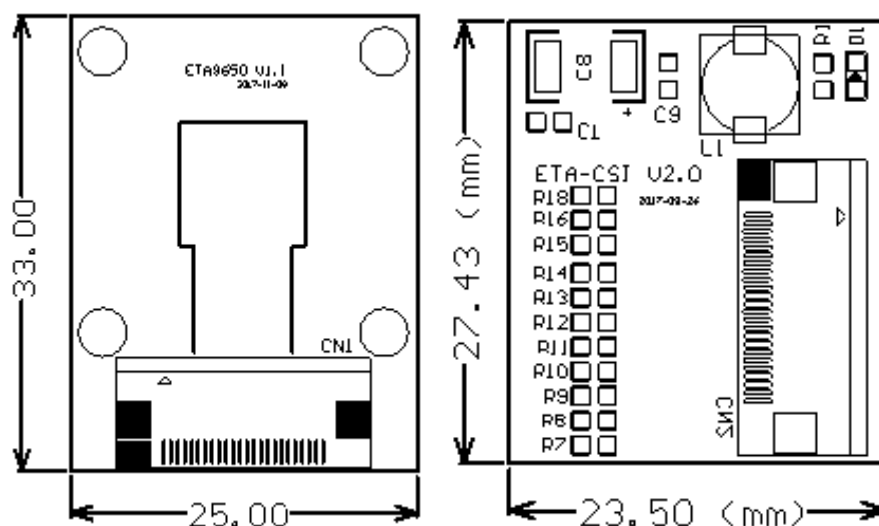


图 1: ETA9650 示意图

主板部分：

CN1 为摄像头的接口，采用 24 芯 ZF 插针，直接摄像头的带线连接。在以下的信号定义表中，信号名称带#尾缀的，表示低电平有效。

CN1 具体信号定义如下：

信号名称及简要描述	CN1		信号名称及简要描述
	PIN#	PIN#	
-	1	2	GND, 公共地
SDA, I2C 信号	3	4	2.8V
SCL, I2C 信号	5	6	Reset, 复位
VSYNC	7	8	GPIO, 电源管理
HREF	9	10	1.5V
3.3V	11	12	CSI_DATA07
MCLK	13	14	CSI_DATA06
GND, 公共地	15	16	CSI_DATA05
PCLK	17	18	CSI_DATA04
CSI_DATA00	19	20	CSI_DATA03
CSI_DATA01	21	22	CSI_DATA02
3.3V	23	24	GND, 公共地

信号说明：

CSI_DATA 0- CSI_DATA 7：CSI 接口信号。

MCLK：摄像头主时钟。

PCLK：像素时钟。

CN2 为调整后为了和转接板连接的接口：

信号名称及简要描述	CN2		信号名称及简要描述
	PIN#	PIN#	
3.3V, 直流电源	1	2	3.3V, 直流电源
SCL, I2C 信号	3	4	SDA, I2C 信号

GPIO, 电源管理	5	6	Reset, 复位
GND, 公共地	7	8	MCLK
GND, 公共地	9	10	FIELD/NC, NC
VSYNC	11	12	HREF
GND, 公共地	13	14	PCLK
GND, 公共地	15	16	CSI_DATA07
CSI_DATA06	17	18	CSI_DATA05
CSI_DATA04	19	20	CSI_DATA03
CSI_DATA02	21	22	CSI_DATA01
CSI_DATA00	23	24	GND, 公共地

转接板部分:

ESM6800V 将 ESMARC 规范中 ISA 接口的管脚定义为 CSI 信号接口的管脚, 所以转接板的 CN1 可以直接与 ESMARC 通用评估底板连接的 ISA 接口, 然后通过 CN2 用带线连接主板部分, CN1 的定义如下:

信号名称及简要描述	CN1		信号名称及简要描述
	PIN#	PIN#	
-	1	2	HREF
CSI_DATA00	3	4	CSI_DATA04
CSI_DATA01	5	6	CSI_DATA05
CSI_DATA02	7	8	CSI_DATA06
CSI_DATA03	9	10	CSI_DATA07
-	11	12	VSYNC
#Reset	13	14	PIXCLK
电源管理	15	16	MCLK
I2C_SCL	17	18	VCC, 5V 直流供电
I2C_SDA	19	20	GND, 公共地

CN2 是与搭载摄像头的主板连接的接口，所以定义与主板的 CN2 是相同的，这里不再赘述。

3. 应用说明

在 ESM6800V 的系统中已经集成了摄像头模块的驱动，并且能够自动识别并加载相应的驱动，加载驱动后会自动生成设备节点：“/dev/video0”，应用程序可以操作该设备节点对摄像头进行图像的采集和控制。

CSI 摄像头都是用了 V4L2 驱动提供的标准 API 来操作的。Video for Linux 2 简称 V4L2，是 V4L 的改进版。接下来就来介绍一下主要的操作，首先打开设备文件：

```
int fd;
fd=open("/dev/video0",O_RDWR);
```

设置视频的制式和帧格式，制式包括 PAL，NTSC，帧的格式个包括宽度和高度等：

```
/*set the form of camera capture data*/
tv_fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;      /*v4l2_buf_typea,camera must
use V4L2_BUF_TYPE_VIDEO_CAPTURE*/
tv_fmt.fmt.pix.width = 680;
tv_fmt.fmt.pix.height = 480;
tv_fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV; /*V4L2_PIX_FMT_YUYV*/
tv_fmt.fmt.pix.field = V4L2_FIELD_NONE;         /*V4L2_FIELD_NONE*/
if (ioctl(fd, VIDIOC_S_FMT, &tv_fmt)< 0)
{
    fprintf(stderr,"VIDIOC_S_FMT set err\n");
    exit(-1);
    close(fd);
}
}
```

向驱动申请帧缓冲，一般不超过五个：

```
struct v4l2_requestbuffers req;
req.count=2;
req.type=V4L2_BUF_TYPE_VIDEO_CAPTURE;
req.memory=V4L2_MEMORY_MMAP;
//申请帧缓冲
ret=ioctl(fd,VIDIOC_REQBUFS,&req);
if(ret<0)
{
    printf("failture VIDIOC_REQBUFS\n");
    return -1;
}
}
```

将申请到的帧缓冲映射到用户空间，这样就能够直接操作帧缓冲了：

```
for (n_buffers = 0; n_buffers < req.count; ++n_buffers)
{
    struct v4l2_buffer buf;
    memset(&buf,0,sizeof(buf));
    buf.type =V4L2_BUF_TYPE_VIDEO_CAPTURE;
    buf.memory =V4L2_MEMORY_MMAP;
    buf.index =n_buffers;
    // 查询序号为n_buffers 的缓冲区，得到其起始物理地址和大小
    if (-1 == ioctl(fd, VIDIOC_QUERYBUF, &buf))
    {
        printf("failture VIDIOC_QUERYBUF\n");
        return -1;
    }
    buffers[n_buffers].length= buf.length;
    // 映射内存
    buffers[n_buffers].start=mmap (NULL,buf.length,PROT_READ |
PROT_WRITE ,MAP_SHARED,fd, buf.m.offset);
    if (MAP_FAILED == buffers[n_buffers].start)
    {
        printf("failture mmap\n");
        return -1;
    }
}
```

开始视频的采集:

```
type =V4L2_BUF_TYPE_VIDEO_CAPTURE;
ioctl (fd,VIDIOC_STREAMON, &type);

struct v4l2_buffer camera_buf;
CLEAR (camera_buf);
camera_buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
camera_buf.memory = V4L2_MEMORY_MMAP;
//取出一个缓冲帧
i1 = ioctl (fd, VIDIOC_DQBUF, &usr_buf);
if(i1<0)
{
    printf("failture\n");
    return -1;
}
```


4. 历史版本

手册版本	简要描述	日期
V1.0	创建文档	2017-11