

原文链接: <https://www.nxp.com/docs/en/application-note/AN12110.pdf>

使用i.MX RT系列开发摄像头应用程序

内容

1. 简介

本应用笔记介绍了如何使用NXP i.MX RT1050处理器开发高清摄像头应用程序。

对于此演示应用程序，RT1050从摄像头设备接收分辨率为HD-720（1280x720像素，也称为720P）的视频帧，在LCD面板上显示，并可选择以压缩JPEG格式保存到microSD卡中。

i.MX RT1050是具有单个ARM Cortex-M7内核的处理器，其运行速度高达600 MHz。i.MX RT1050强大的处理能力，实时功能以及丰富的外围设备的集成使它们成为工业计算、电机控制、电源转换、智能消费类产品、高端音频系统和家庭和楼宇自动化等许多高性能应用的理想选择。

本文档的**第2节**介绍了演示应用程序的硬件和软件平台。**第3节**介绍了基于[MCUXpresso 软件开发包](#)。使用i.MX RT1050开发高清摄像头应用程序的过程。

1. 简介.....	1
2. 硬件和软件平台.....	2
2.1. i.MX RT1050 处理器.....	2
2.2. i.MX RT1050 EVK 板.....	2
2.3. MT9M114 图像传感器.....	3
2.4. RK043FN02H-CT TFT LCD 面板.....	3
2.5. MCUXpresso 软件开发包.....	4
3. 开发高清摄像头应用.....	4
3.1. 系统结构分析.....	4
3.2. 创建演示项目.....	5
3.3. 视频捕获.....	8
3.4. LCD显示画面.....	10
3.5. 将帧存储到SD卡.....	12
3.6. 软件结构.....	12
3.7. 运行演示应用程序.....	13
4. 结论.....	13
5. 参考文献.....	13
6. 修订历史.....	13



2. 硬件和软件平台

本节简要介绍了演示应用程序的硬件和软件平台，包括以下各项。

- [i.MX RT 1050 处理器](#)
- [i.MX RT1050 EVK板](#)
- [MT9M114 图像传感器](#)
- [RK043FN02H-CT TFT LCD面板](#)
- [MCUXpresso 软件开发包](#)

2.1. i.MX RT1050 处理器

NXP提供的具有单个ARM Cortex-M7内核的[i.MX RT1050](#)可以在高达600 MHz的速度下运行。它具有512 KB片上RAM，可以灵活配置为核心紧密耦合存储器（TCM）或通用RAM。它提供连接各种外部存储器的各种接口，以及各种串行通信接口，如USB、以太网、SDIO、CAN、UART、I2C和SPI。它还具有丰富的音频和视频功能，包括LCD显示屏、基本2D图形、摄像头接口、SPDIF和I2S音频接口。其他值得注意的功能包括用于安全性、电机控制、模拟信号处理和电源管理的各种模块。

CMOS传感器接口（CSI）模块使芯片可以直接连接到外部CMOS图像传感器。CSI支持CCIR656视频接口以及传统的传感器接口，具有8/10/16位宽度的数据端口。内嵌的DMA控制器具有通过AHB总线将数据从接收FIFO或统计FIFO传输到内存缓冲区的能力。

增强型液晶显示接口（eLCDIF）是一个RGB接口显示控制器，支持8/16/18/24位宽度的数据端口，分辨率高达1366x768 WXGA。为了传输帧数据以进行显示刷新，eLCDIF可以充当总线主机，或作为总线从机与SOC集成的DMA引擎协同工作。在这两种情况下，CPU都不再处理帧数据。

像素管道（PXP）模块集成了多种2D图形处理功能，包括缩放，色彩空间转换（CSC）和旋转。

2.2. i.MX RT1050 EVK 板

[i.MX RT1050 EVK](#)板是一个平台，旨在展示i.MX RT1050处理器最常用的功能。EVK板提供以下功能：

- 存储：256Mbit SDRAM，64Mbit Quad SPI Flash，512Mbit Hyper Flash，TF卡插槽；
- 通信接口：USB 2.0 OTG连接器，USB 2.0主机连接器，10/100 Mbit / s 以太网连接器，CAN总线连接器；
- 多媒体接口：CMOS传感器连接器，LCD连接器；
- 音频接口：3.5毫米立体声耳机插孔，板载麦克风，SPDIF连接器（默认情况下未安装）；

Developing a Camera Application with i.MX RT Series, Application Note, Rev. 0, 12/2017

- 调试接口：板载DAP-Link调试适配器，JTAG 20针连接器；
- Arduino 接口；
- 用户按钮和LED灯。

图 1 展示了 i.MX RT1050 EVK 的图片。

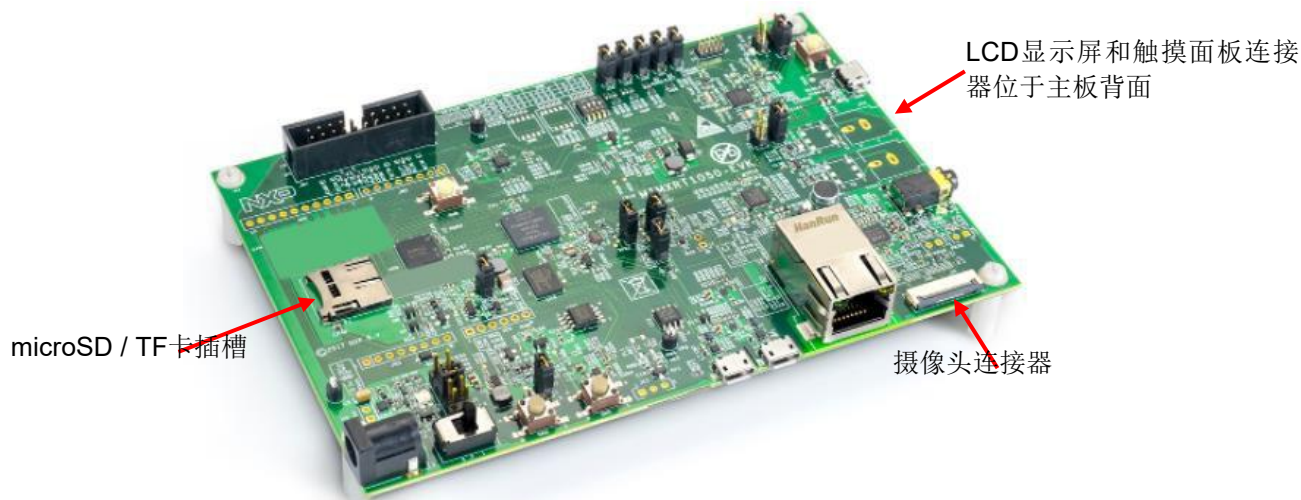


图 1. i.MX RT1050 EVK

2.3. MT9M114 图像传感器

来自安森美半导体(原为Aptina Imaging Corporation)的MT9M114是一款1/6英寸1.26 Mp CMOS数字图像传感器，有效像素阵列为1296x976。它包括高级的摄像头功能，例如自动曝光控制，自动白平衡，黑电平控制，防止闪烁和缺陷校正。

此演示应用程序使用了一个MT9M114摄像头模块。该模块通过一排与i.MX RT1050 EVK板上的摄像头连接器兼容的24线FPC导线引出其信号。

2.4. RK043FN02H-CT TFT LCD 面板

RK043FN02H-CT TFT LCD 面板的特性如下：

- 4.3英寸物理尺寸；
- 480*272 像素 (RGB888)；
- 24位RGB888接口，支持DE或HV模式；
- LED背光灯；
- I2C接口电容式触摸；

LCD面板通过40针FPC线引出其信号，通过6针FPC线引出触摸信号，这些信号与i.MX RT1050 EVK板上的连接器兼容。

i.MX RT1050 EVK不使用所有的24位显示数据信号，而是部分信号，最多支持RGB565。

2.5. MCUXpresso 软件开发包

MCUXpresso 软件开发包为NXP的多个微控制器系列提供了全面的软件支持。软件开发包包含以下组件：

- 一套灵活的外设驱动程序。
- 丰富的示例应用程序。
- 来自NXP或第三方的多种中间件，如FreeRTOS、emWin、FatFs、LIBJPEG、LwIP、mbed TLS、USB协议栈、wolfSSL等。
- SOC头文件，各种工具链的启动文件和链接器配置文件。

3. 开发高清摄像头应用

本节介绍了基于第2节中介绍的硬件和软件平台开发高清摄像头应用程序的过程。

3.1. 系统结构分析

首先，我们分析系统需求，然后找出系统结构，并绘制系统框图。这样我们就可以开发每个软件组件，然后构建整个系统。

图 2 显示了这个演示应用程序的硬件框图，其中有系统的主要组件。

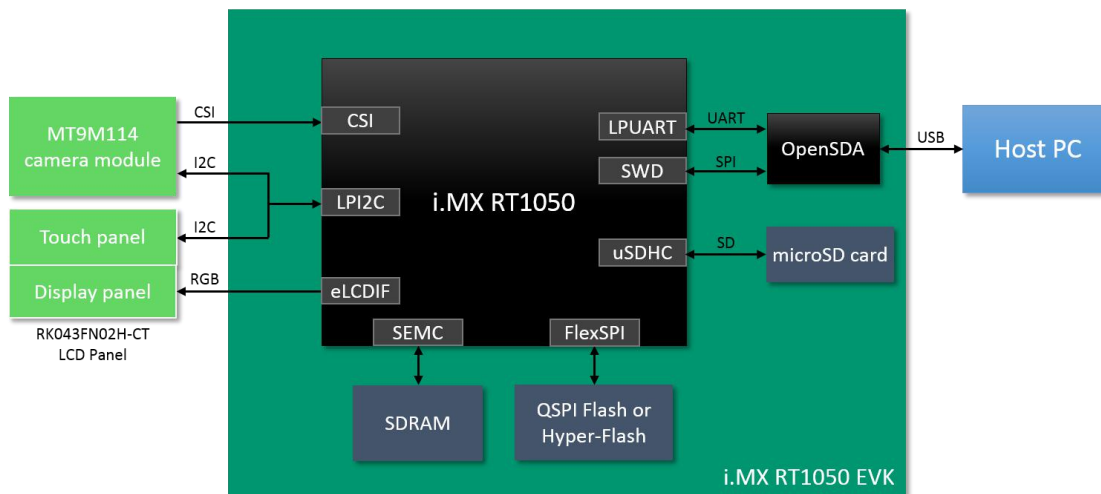


图 2. 系统硬件框图

- RT1050通过I2C总线配置MT9M114，并通过CSI接口捕获视频帧数据。
- RT1050通过RGB接口将帧数据传输到LCD面板，并通过I2C总线接收触摸事件。
- 与RT1050 uSDHC模块连接的microSD卡用于存储压缩视频帧的JPEG文件。

- SDRAM为帧缓存和/或代码空间提供了数据空间。RT1050通过智能外部存储器控制器（SEMC）模块与SDRAM器件连接。
- QSPI flash或hyper flash为非调试运行配置提供代码空间, i.MX RT1050通过FlexSPI控制器连接二者。
- 开放标准串行调试适配器（OpenSDA）为板提供SWD调试访问，调试UART桥接和电源。OpenSDA通过USB端口与主机PC通信。

图 3 显示了此演示应用程序的帧数据流程图。

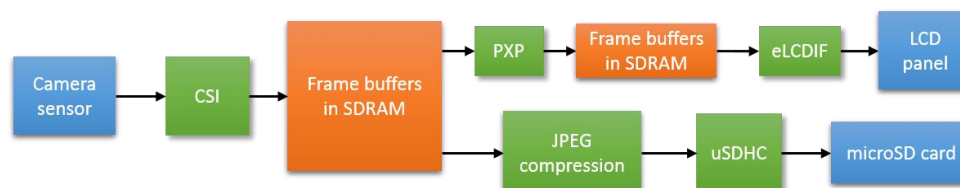


图 3. 帧数据流图

CSI接收帧数据并存储在SDRAM中的帧缓存中。一方面，引入PXP将帧大小从原来的1280x720调整为480x72，然后eLCDIF将调整后的帧传输到LCD显示面板上。另一方面，帧可以可选择性地压缩成JPEG格式并存储在microSD卡中。

3.2. 创建演示项目

如我们在第2节中所述，MCUXpresso 软件开发包为嵌入式应用程序提供了丰富的外围设备驱动程序，中间件，示例应用程序以及其他必要组件的集合。建议基于这些软件开发包组件开发应用程序，以加快并简化开发过程。

您可以通过移植软件开发包示例项目来创建您的应用程序项目，然后删除，添加和自定义组件。或创建一个空项目，然后添加并自定义组件。选择自己喜欢的方式。在此演示中，我们首先使用IAR Embedded Workbench IDE版本8.20为应用程序创建一个空项目。

以下小节介绍了基于MCUXpresso 软件开发包创建应用程序项目的步骤。有关详细信息，请参考软件包以及本应用笔记和IDE帮助文档。

3.2.1. 下载 MCUXpresso 软件开发包

从 <https://mcuxpresso.nxp.com> 下载软件开发包软件包，需要登录。如果您没有NXP帐户，请注册一个。为EVK板下载最新版本的软件开发包软件包软件开发包_2.3.0_EVK-MIMXRT1050.zip，并将存档文件解压缩到本地磁盘。

3.2.2. 创建一个空的 IAR project

按照以下步骤创建一个空的IAR项目。

1. 在 `boards\evkmimxrt1050` 目录中为项目创建新目录 `user_apps\camera_demo\iar`。
2. 启动 IAR Embedded Workbench，并在创建的“iar”目录中创建一个新项目“camera_demo.ewp”。并将工作区另存为“camera_demo.eww”。
3. 由于项目中使用 SDRAM 来保存数据和代码，所以我们创建了一个名为“`sdram_txt`”的配置目标。并删除默认的“调试”和“发布”配置。

本文档主要描述用于调试阶段的 SDRAM 配置。除此配置外，软件包还提供了用于离线运行的 FlexSPI NOR flash 配置。这两种配置的唯一区别在于存储空间的分配。

3.2.3. 导入所需的源文件

按照以下步骤将所需的源文件导入到项目中。

1. 将要使用的驱动文件从“`devices\MIMXRT1052\drivers`”目录导入到项目中，包括 `cache`、`clock`、`CSI`、`eLCDIF`、`GPIO`、`IOMUX`、`LPI2C`、`LPUART`、`PXP`、`uSDHC` 的驱动程序。
2. 从“`devices\MIMXRT1052`”和“`devices\MIMXRT1052\iar`”目录导入启动文件。
3. 从“`devices\MIMXRT1052\utilities`”目录及其子目录导入调试 UART 文件。
4. 从“`middleware\fatfs\src`”目录导入 FatFs 源文件。
5. 从“`middleware\sdmmc`”目录导入 SD/MMC 卡管理文件。
6. 从“`middleware\libjpeg`”目录导入 LIBJPEG 源文件。
7. 从示例目录（例如“`sdcard_fatfs`”）复制板特定文件，然后进入项目目录，并将文件导入到项目中。

除了来自软件开发包的这些现有文件之外，还需要为此演示项目创建一些特定于项目的源文件，以配置各种模块，捕获和显示框架，处理触摸事件等。本文档的其余部分没有详细介绍那些项目专用文件，但说明了实现的方法和原理。有关更多详细信息，请参阅软件包。

3.2.4. 项目配置

在 IDE 中打开“项目选项”对话框，然后配置项目。

1. 将目标设备指定为“`NXP MIMXRT1052xxx6A`”。
2. 添加各种 `include` 路径，包括 CMSIS 的头文件目录、外设驱动程序、启动文件、`debug UART` 实用程序、`FatFs`、`LIBJPEG`、`SDMMC` 以及项目特定的头文件。
3. 为软件开发包驱动程序配置添加必要的项目符号定义。

4. 指定“CISIS-DAP”作为调试器驱动程序，设置SWD作为接口。
5. 将调试器设置宏文件和脚本文件从示例项目复制到项目目录，该目录主要用于SDRAM初始化。然后将文件导入到项目中。

3.2.5. 存储空间分配

将链接器配置文件“MIMXRT1052xxxxx_sdram_txt.icf”和“MIMXRT1052xxxxx_flexspi_nor.icf”从“devices\MIMXRT1052\iar”目录复制到项目目录。编辑此文件，根据应用程序要求为文本、数据、堆栈、堆和无缓存区域指定存储空间（RAM、SDRAM、FlexSPI flash）。

对于此演示应用程序，我们使用如图4所示的方案分配存储空间。

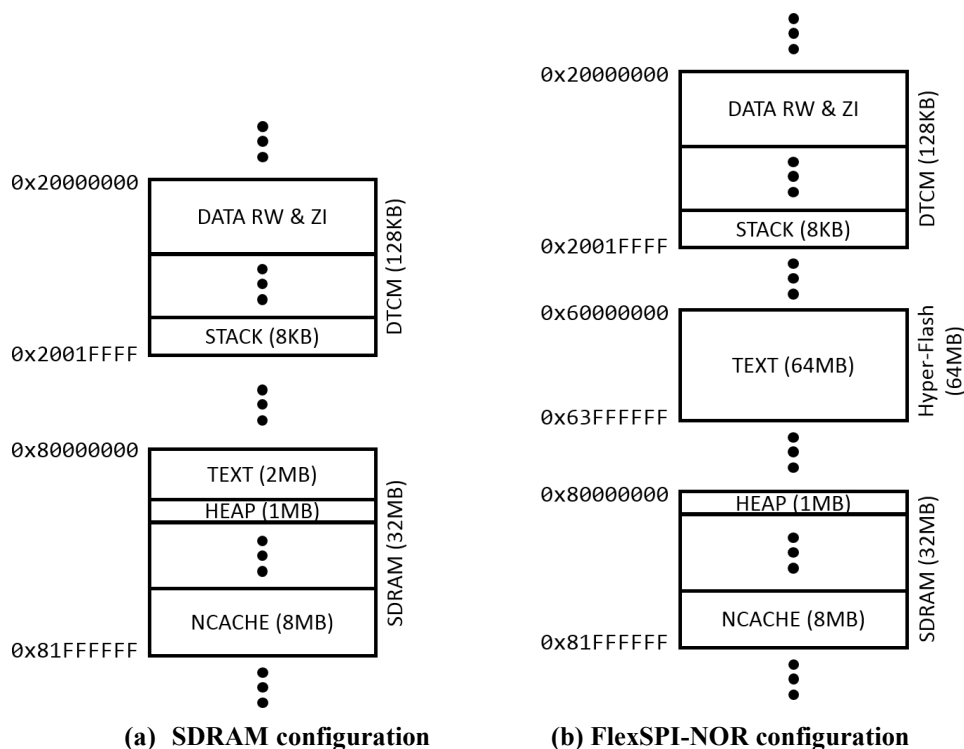


图 4. 内存分配方案

最后，打开project option对话框，将自定义的链接器配置文件导入到项目中进行配置。

3.2.6. 引脚复用器配置

i.MX RT1050的大多数引脚被多个外围模块多路复用。在使用多路复用引脚之前，必须先配置该引脚的多路复用。

对于此演示应用程序，我们需要配置调试UART，SEMC，CSI，LPI2C，eLCDIF和SDHC引脚的引脚复用。

3.3. 视频捕获

对于这个演示应用程序，MT9M114摄像头设备被配置为以30fps帧速率输出RGB565像素格式的720P视频帧。摄像头设备和CSI模块都需要预先配置。

3.3.1. MT9M114 驱动程序

MT9M114驱动程序提供API和参数来配置摄像头设备。注意，MT9M114在工作之前有大量的寄存器（和所谓的变量）需要配置。用户很难确定各种参数。因此，设备供应商提供了一个配置工具注册向导*Register Wizard*，这使得获得特定视频特征集的基本参数变得容易。图5显示了*Register Wizard*工具界面。

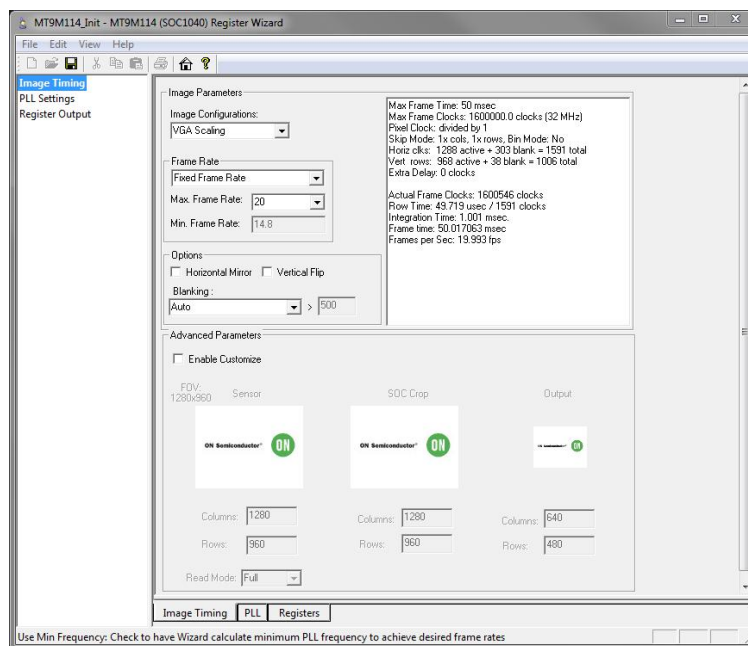


图 5. The Register-Wizard tool

要存档或更新配置，应完成以下步骤：

1. 为需要写入的寄存器和变量准备参数。
2. 通过I2C接口写入寄存器和变量。
3. 按规定时序写入一系列命令来发起更改配置过程。

3.3.2. 通过CSI捕获视频

CSI模块需要针对特定的视频格式进行初始化。CSI驱动程序参考各种视频特性来初始化CSI，包括像素数据格式，每个像素的字节数，帧分辨率，帧速率等。

CSI内嵌的DMA可以交替写入两个帧缓冲地址而不需要停止。为了提高性能，特别是对于复杂的应用程序，软件开发包CSI驱动程序将支持的缓冲区扩展到最多四个，并将帧缓冲区组成环形队列结构。CSI一个接一个地填充缓冲区，这样LCD接口等数据接收器就可以同时读取缓冲区。图6说明了CSI帧缓冲区填充方案。

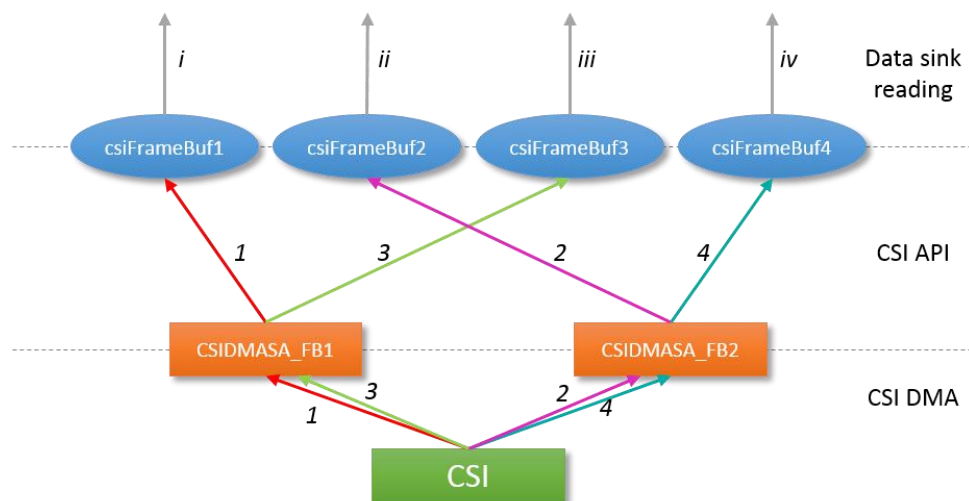


图 6. CSI帧缓冲填充方案

CSI DMA使用两个寄存器CSIDMASA_FB1和CSIDMASA_FB2（用来保存缓冲区地址）来访问四个帧缓冲区。该过程是：

- a. 设置 *CSIDMASA_FB1* 指向帧缓冲区1, *CSIDMASA_FB2*指向帧缓冲区2.
- b. CSI通过*CSIDMASA_FB1*填充帧缓冲区1（图6中的数据流1）。
- c. 帧缓冲区1满后，将产生一个中断。设置*CSIDMASA_FB1*指向帧缓冲区3。
- d. CSI开始通过*CSIDMASA_FB2*填充帧缓冲区2（图6中的数据流2）。同时，数据接收器可以读取帧缓冲区1（图6中的数据流i）。
- e. 帧缓冲区2满后，将产生一个中断。设置*CSIDMASA_FB2*指向帧缓冲区4。
- f. CSI开始通过*CSIDMASA_FB1*填充帧缓冲区3（图6中的数据流3）。同时，数据接收器可以读取帧缓冲器2（图6中的数据流ii）。
- g. 帧缓冲区3满后，将产生一个中断。设置*CSIDMASA_FB1*指向帧缓冲区1。
- h. CSI开始通过*CSIDMASA_FB2*（图6中的数据流4）填充帧缓冲区4。同时，数据接收器可以读取缓冲区3（图6中的数据流iii）。
- i. 帧缓冲区4满后，将产生一个中断。设置*CSIDMASA_FB2*指向帧缓冲区2。
- j. 再次重复步骤2。同时，数据接收器可以读取缓冲区4（图6中的数据流iv）。

同一组帧缓冲区上的填充和排空操作可以同时进行，也可以独立进行，可以使系统性能最大化。

3.4. LCD显示画面

为了排空CSI帧缓冲区，引入了两个缓冲区指针`csiFrameBufPtr1`和`csiFrameBufPtr2`。结合两个软件开发包驱动程序函数`CSI_TransferSubmitEmptyBuffer()`和`CSI_TransferGetFullBuffer()`，顶层应用程序可以连续读取CSI帧缓冲区。图7说明了CSI帧缓冲区读取方案。

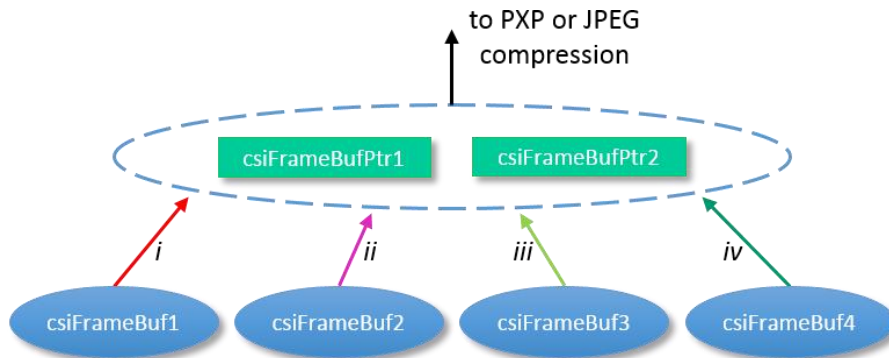


图 7. CSI帧缓冲区排空方案

需要注意的是，两个缓冲区指针被设计为交替指向四个帧缓冲区。此策略的目的是允许帧缓冲区的填充和读取同时连续进行。

由于LCD的显示分辨率与摄像头的输出分辨率不同，因此需要额外的LCD缓冲区，并且PXP模块负责缓冲区间间的缩放。

图8说明了LCD帧缓冲区访问方案。

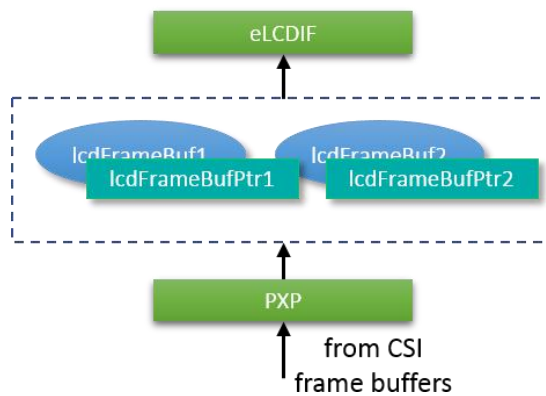


图 8. LCD帧缓存访问方案

引入了两个LCD帧缓冲区和两个缓冲区指针。PXP通过两个指针填充两个缓冲区，而eLCDIF也通过两个指针读取两个缓冲区。类似地，使用两个指针还可以使缓冲区的填充和读取连续同时进行。

帧缓冲区操作流程如图 9 所示，包括 CSI 缓冲区和 LCD 缓冲区的操作。这里我们可以看到如何操作缓冲区指针来实现填充和读取操作的连续性、同步性和独立性。

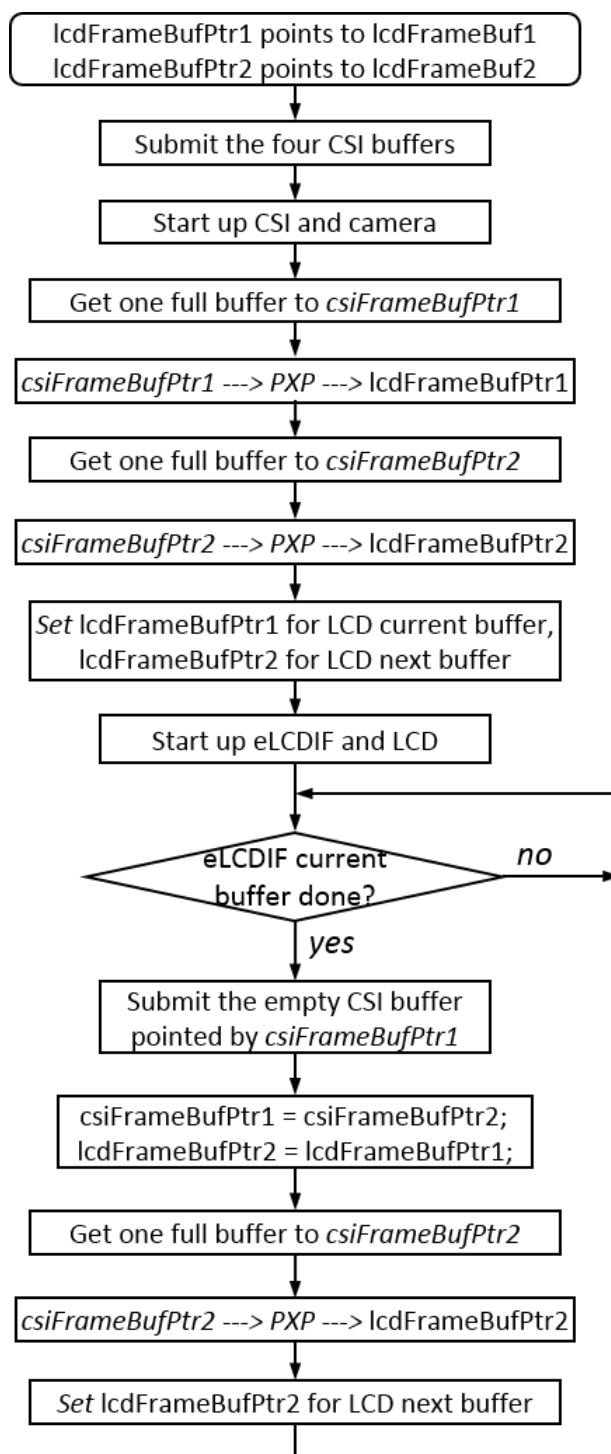


图 9. CSI/LCD 帧缓冲操作流程

3.5. JPEG压缩并存储在SD卡中

对于此演示应用程序，当用户触摸LCD面板时，当前视频帧将被压缩为JPEG格式并存储在microSD卡中。

获取用于JPEG压缩的CSI帧的方法与获取LCD显示的帧的方法类似。实际上，用于获取压缩帧和显示帧的代码是混合在一起的。如图9所示，当接收到一个CSI帧并由*csiFrameBufPtr2*处理时，将轮询触摸事件。

如果检测到触摸事件，将启动JPEG压缩。

开源的LIBJPEG和FatFs包含在软件开发包中。这个演示应用程序利用这两个组件将视频帧压缩为JPEG文件并存储在microSD卡中。

3.6. 软件结构

图10展示了这个演示应用程序的总体软件结构，其中显示了程序和数据流。

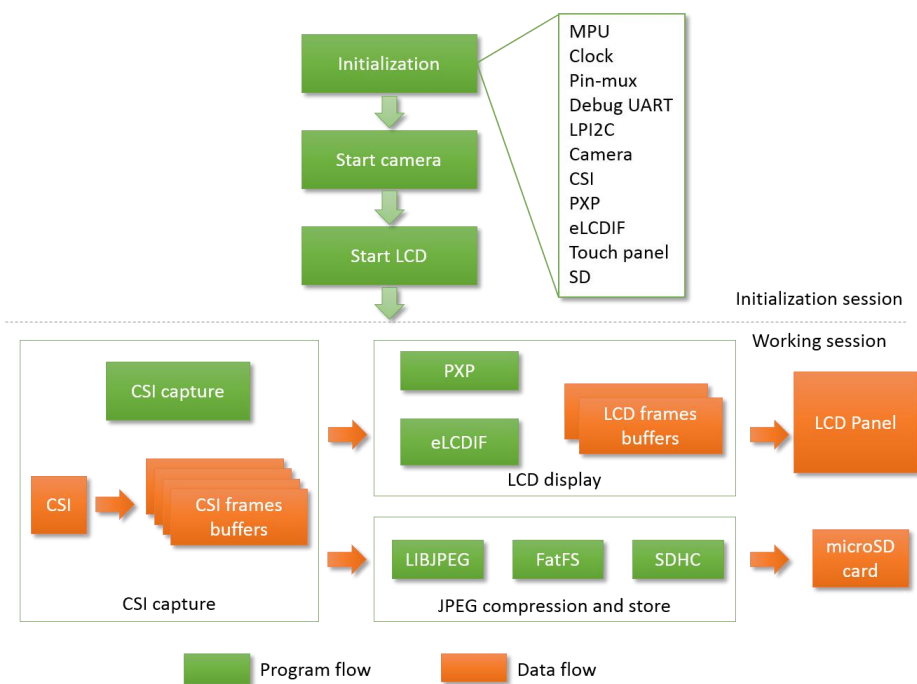


图 10. 总体程序和数据流

在初始化过程中，将初始化各种组件，如系统时钟、引脚复用、摄像头器件、CSI、eLCDIF、触摸面板和SD卡。然后摄像头和LCD显示器开始运行。

在连续运行中，CSI捕获帧，CSI用这些帧填充帧缓冲区。一方面，CSI帧通过PXP缩放到LCD帧大小，并通过eLCDIF传输到LCD面板上。另一方面，CSI帧可以选择性地压缩成JPEG格式并存储到microSD卡中。

3.7. 运行演示应用程序

与本文档一起发布的软件包提供了演示应用程序的全部源文件和项目文件。运行演示需要：

- 使用micro-USB线连接电脑和EVK-MIMXRT1050板上的OpenSDA USB端口J28。
- 根据需要打开一个串行终端工具，设置为115200波特率、8个数据位、无奇偶校验位和1个停止位，用来显示调试日志。
- 打开演示项目，并构建这其中配置：“s dram_txt”或“flexspi_nor”。
- 启动调试或将二进制文件下载到Flash存储器。
- 在IDE中启动调试器，或按重置按钮SW4开始运行演示。
- 然后可以在LCD面板上看到视频帧。您可以用手指触摸LCD面板，并将一帧压缩为JPEG格式并存储在microSD卡中。

4. 结论

本应用笔记介绍了使用基于MCUXpresso 软件开发包的i.MX RT1050处理器开发摄像头应用程序的步骤，从创建一个空白项目到完成该应用程序。软件开发包提供的外围设备驱动程序和各种中间件使整个开发过程更容易。

随本应用笔记一起提供了演示应用程序的源代码，您可以在此基础上开发自己的自定义摄像头应用程序。

5. 参考文献

以下文件可能会提供进一步的参考。

- [i.MX RT1050 Processor Reference Manual, Rev. 0](#)
- [MT9M114 Data Sheet, Rev. 12](#)
- [MT9M114 Register and Variable Reference, Rev. E](#)

6. 修订历史

Revision number	Date	Substantive changes
0	12/2017	Initial release

如何找到我们:

主页:

nxp.com

网页支持:

nxp.com/support

本文档中的信息仅用于使系统和软件实施者能够使用NXP产品。根据本文档中的信息，根据本文档的设计，没有授予任何明示或暗示的版权许可来设计或制造任何集成电路。恩智浦保留更改的权利，恕不另行通知此处的任何产品。

恩智浦不就其产品在特定目的下的适用性提供任何担保，陈述或担保，也不承担因应用或使用任何产品或电路而引起的任何责任，并且恩智浦明确声明不承担任何责任，包括但不限于结果性或偶然性损害。恩智浦数据表和/或规格中可能提供的“典型”参数在不同的应用中可能会发生变化，并且实际性能可能会随时间变化。必须由客户的技术专家针对每个客户应用验证所有操作参数，包括“典型值”。恩智浦不会根据其专利权或他人的权利转让任何许可。恩智浦根据标准销售条款和条件销售产品，可在以下地址找到该产品：nxp.com/SalesTermsandConditions。

恩智浦（NXP），恩智浦（NXP）徽标，恩智浦（NXP）智慧世界的安全连接，飞思卡尔（Freescale），飞思卡尔（Freescale）徽标是NXP B.V.的商标。所有其他产品或服务名称均为其各自所有者的财产。

Arm，Arm徽标和Cortex是ARM Limited（或其子公司）在欧盟和/或其他地方的注册商标。版权所有。

© 2017 NXP B.V.

Document Number: AN12110
Rev. 0
12/2017

The ARM logo consists of the lowercase letters "arm" in a bold, blue, sans-serif font.