AN4500
应用笔记

如何在 4 灰度级 E-Paper 上显示 STM32 嵌入式存储器中尺寸优化的图片

前言

本应用笔记描述了如何优化黑白图像的大小以将其存入 STM32 微控制器的嵌入式闪存，以及如何在 E-Paper 显示器上显示它们。

本应用笔记中说明了如何准备和编码黑白图像，并给出了对图像解压缩、将其在 4 灰度级的 E-Paper 显示器上显示出来的软件解决方案。

STM32 微控制器可连接 E-Paper 显示器，使用特定外设向 E-Paper 显示器控制器发送数据 / 命令，并驱动特定 GPIO 来管理 E-Paper 控制引脚。

本应用笔记和相关软件 （STSW-STM32152）都是基于 STM32L053 探索套件（32L0538DISCOVERY），提供了嵌入式 E-Paper 显示器。对于任意一款 STM32 微控制器客户板，经过微小改动 （时钟配置， GPIO 定义），可以很容易地重复利用。

E-Paper 显示器的显示区域为 2.1 英寸，包含 172x72 个像素，具有 2 比特完全显示能力。关于本应用笔记中未说明的 E-Paper 功能，更多详细信息请参考 ST 网站上的 GDE021A1 规范。

表 1. 可用产品、工具 & 软件

类型	参考产品
STM32 嵌入式软件	STSW-STM32152
STM32 MCU 评估工具	32L0538DISCOVERY

目录

1 实现示例 5

1.1 概述 5

1.2 STM32 配置 6

1.2.1 SPI 外设 6

1.2.2 系统时钟 6

1.2.3 使用特定的 GPIO 控制 E-Paper 显示器。 6

1.3 E-Paper 显示器配置 6

1.4 图片创建和尺寸压缩 8

1.4.1 图片帧 9

1.5 图片数据扩展，以载入嵌入式 E-Paper RAM 11

2 固件说明 12

2.1 系统配置 12

2.2 中断源 12

2.3 E-Paper 电源 12

2.4 主要软件函数说明 12

3 可能的固件优化 15

3.1 E-Paper 功耗管理 15

3.2 E-Paper 显示 RAM 的部分更新 15

4 版本历史 16

表格索引

表 1. 可用产品、工具 & 软件 1
表 2. Ram 地址映射 8
表 3. 高层软件函数 13
表 4. EPD_IO_WriteReg 函数 13
表 5. EPD_IO_WriteData 函数 14
表 6. Gde021a1_DrawImage 函数 14
表 7. E-Paper 显示模块低功耗模式 15
表 8. 文档版本历史 16
表 9. 中文文档版本历史 16

图片索引

图 1. 典型的实现设计说明 5
图 2. 数据输入模式和 RAM 配置 7
图 3. 波形查阅表 8
图 4. 图片帧 9
图 5. 4 张图片显示在 E-Paper 显示器上 9
图 6. 图片向右 90° 旋转 10
图 7. C 常数编码图片帧 10
图 8. 数据扩展，以加载 E-Paper RAM 11

1 实现示例

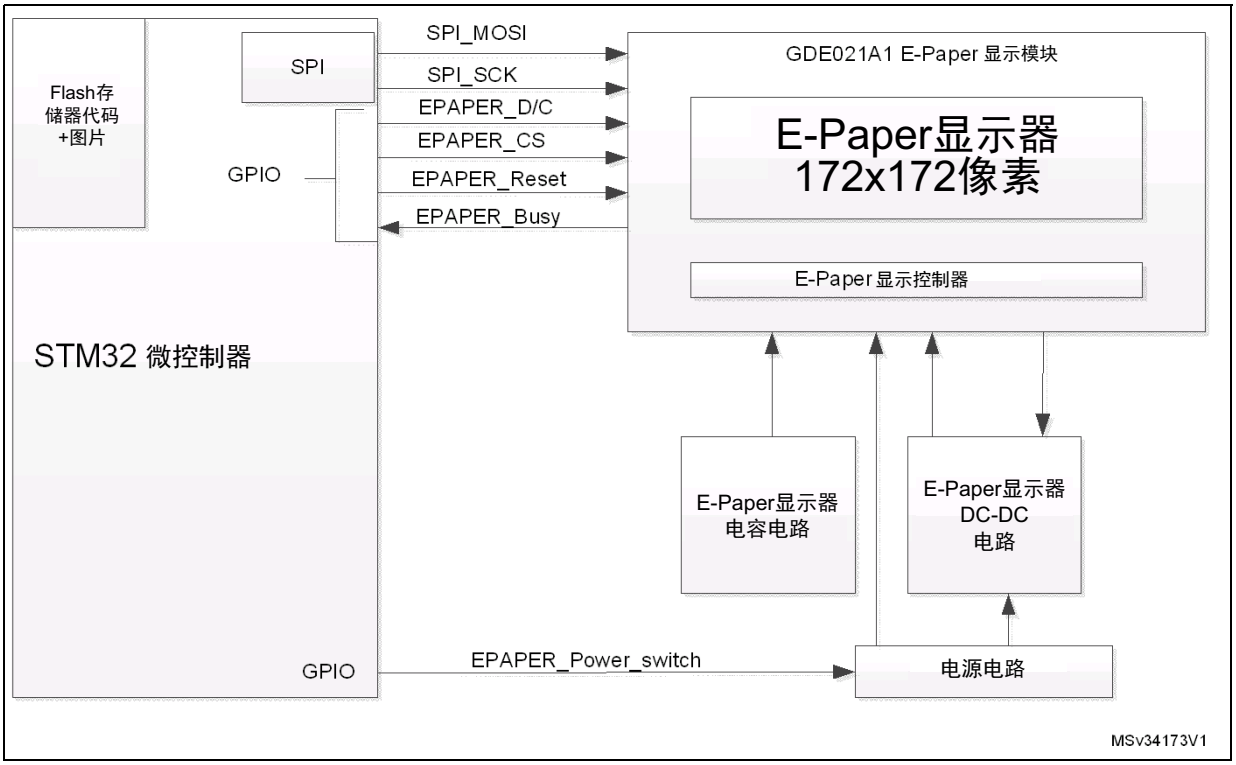
1.1 概述

本应用笔记中的示例提供了连接某个 E-Paper 与 STM32 微控制器的典型硬件和软件实现基本知识。

一般来说，系统包括：

- 一个 STM32 微控制器
- 一个 E-Paper 显示器，其外部元件用于 E-Paper 显示器驱动器 （嵌入到 GDE021A1 E-Paper 显示模块中）的电荷泵。

图 1. 典型的实现设计说明



E-Paper 显示模块经由 SPI 接口连接到 STM32 MCU 来接收数据和命令，配置显示器并将图片传输到 E-Paper 模块内部 RAM 缓冲器中。

图片存储到内部 Flash 程序存储器中，以减少外部资源。包括 4 张图片以演示 STM32L053 的主要特性。图片会循环显示，当第 4 张图片已经在 E-Paper 模块上显示出来时，又返回第一张图片。

注： *如果嵌入式闪存的空间很小，不能满足应用代码和图片库的需求，那么图片可存储在外部存储器中 （比如 SD 卡或外部闪存）。这种情况下，通常没有必要缩小图片尺寸，也不需要使用本应用笔记中给出的扩展算法。由于图片无任何预处理就被发送到 EPD 的缓冲区，因此缩短了数据处理时间。*

1.2 STM32 配置

一般要求

样例主要基于 STM32L053 探索套件，但其功能和结构说明与大多数应用和平台类似。

1.2.1 SPI 外设

MCU 和 E-Paper 显示器之间的通信使用 SPI 协议。 MCU 将 SPI 配置为主 8 位模式， NSS 由软件管理。这里不需要 CRC。 E-Paper 显示模块可只通过 SPI 通道写入。 这是为什么定义 MOSI 线而不是定义 MISO 线的原因。

用来通信的频率是 2 MHz，开始 HSI 设为 16 MHz，在 SPI 波特率发生器中应用系数为 8 的预分频器 （将其分频为 2 MHz）。

1.2.2 系统时钟

应用笔记中将高速内部振荡器设置为系统时钟。时钟路径上没有分频器， APB 和 AHB 总线频率为 16MHz。

1.2.3 使用特定的 GPIO 控制 E-Paper 显示器。

利用一些特定信号来控制 E-Paper 显示器：

- EPAPER_Reset：此信号由 MCU 产生，用来复位 E-Paper 寄存器，并清除任何正在进行的刷新。
- EPAPER_D/C：数据 / 命令线。此输出由 MCU 生成，使 E-Paper 显示模块能够识别 SPI 发送的数值是命令还是数据。
- EPAPER_CS：这是芯片选择引脚。此输出由 MCU 生成，用来使能嵌入到 E-Paper 显示模块中的 SPI 从设备。
- EPAPER_Busy：此信号来自 E-Paper 显示模块，用来向 MCU 报告该模块的状态。当软件启动刷新时， BUSY 位将被置位， E-Paper 显示器上不能有更多操作（不再有命令或数据），以避免显示错误。
- EPAPER_Power_switch：此 GPIO 用来控制为 E-Paper 显示模块上电 / 掉电的模拟开关，以节约应用的功耗。

1.3 E-Paper 显示器配置

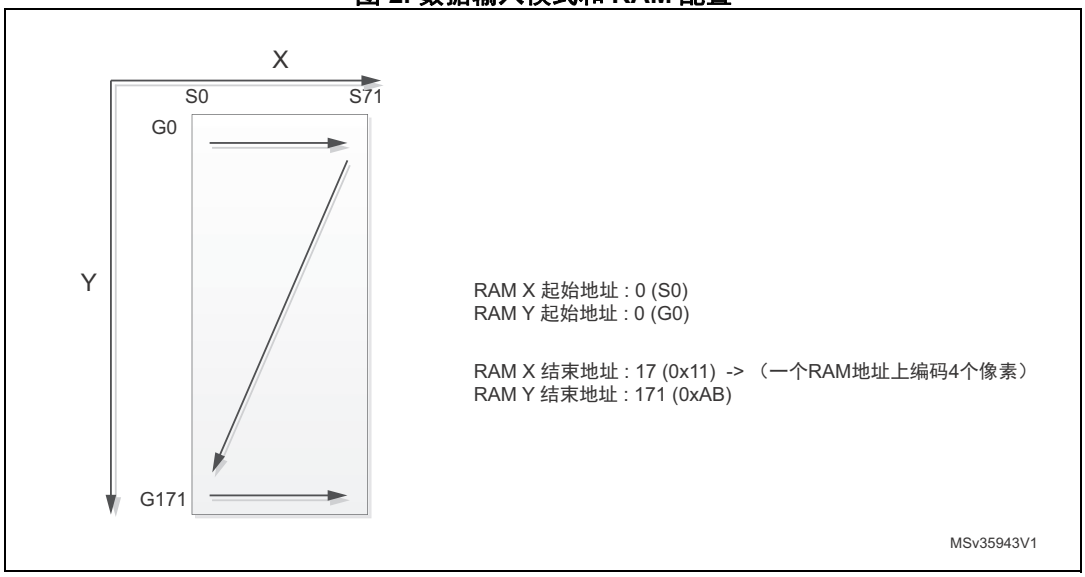
用于 STM32L053 探索套件的 E-Paper 显示器是可配置的。建议参考 E-Paper 说明 （型号 GDE021A1），以便能更好地理解此应用笔记处理 E-Paper 模块所采用的方法。

编码图片的数据通过 MCU 的 SPI 外设被填充到 E-Paper 模块的内部 RAM （请参考*表 2：Ram 地址映射*）。

每次写操作后，根据 E-Paper 模块的配置， RAM 地址递增或递减。地址计数器可沿 X 或 Y 方向更新。对于 X、 Y 坐标系，可独立配置每个轴的开始 / 结束地址。

对于本应用笔记，模块配置为沿 X 方向从 0 至 71 递增地址计数器，然后沿 Y 方向从 0 递增至 171。

图 2. 数据输入模式和 RAM 配置



此应用笔记中所采用的配置在更新操作后无需使 E-Paper 显示模块进入深度睡眠模式。这意味着 RAM 数据保持在两个更新刷新周期之间。

当没有正在进行的更新时，如果应用需要降低功耗，可以配置 E-Paper 使其处于深度睡眠模式。于是功耗可以降低为 1/10 （降至 2 µA），但是这种情况下无法保持 RAM 内容。这种情况的缺点是部分图片不能刷新显示，在刷新显示之前，整个图片的所有像素必须通过 SPI 重新加载到 RAM 中。

这里使用的波形查阅表 （Waveform Look Up Table， LUT）（默认的，无温度范围）。详情请参见 GDE021A1 规范。

图 3. 波形查阅表

```
/* Look-up table for the epaper (90 bytes) */
const unsigned char init_data[]={
0x82,0x00,0x00,0x00,0xAA,0x00,0x00,0x00,
0xAA,0xAA,0x00,0x00,0xAA,0xAA,0xAA,0x00,
0x55,0xAA,0xAA,0x00,0x55,0x55,0x55,0x55,
0xAA,0xAA,0xAA,0xAA,0x55,0x55,0x55,0x55,
0xAA,0xAA,0xAA,0xAA,0x15,0x15,0x15,0x15,
0x05,0x05,0x05,0x05,0x01,0x01,0x01,0x01,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x41,0x45,0xF1,0xFF,0x5F,0x55,0x01,0x00,
0x00,0x00,};//waveform
```

本应用笔记中的 E-Paper RAM 映射如*表 2：Ram 地址映射*中所示进行组织。

表 2. Ram 地址映射

			X-ADDR （SOURCE）													
			S0	S1：	S2：	S3	S4	S5	S6	S7			S68	S69	S70	S71
			00h				01h				...		11h			
Y-ADDR （Gate）	G0	00h	DB0 [7.6]	DB0 [5.4]	DB0 [3.2]	DB0 [1.0]	DB1 [7.6]	DB1 [5.4]	DB1 [3.2]	DB1 [1.0]	...	...	DB17 [7.6]	DB17 [5.4]	DB17 [3.2]	DB17 [1.0]
	G1	01h	DB18 [7.6]	DB18 [5.4]	DB18 [3.2]	DB18 [1.0]	DB19 [7.6]	DB19 [5.4]	DB19 [3.2]	DB19 [1.0]	...	...	DB35 [7.6]	DB35 [5.4]	DB35 [3.2]	DB35 [1.0]
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	G170	AAh	DB3060 [7.6]	DB3060 [5.4]	DB3060 [3.2]	DB3060 [1.0]	DB3061 [7.6]	DB3061 [5.4]	DB3061 [3.2]	DB3061 [1.0]	...	...	DB3077 [7.6]	DB3077 [5.4]	DB3077 [3.2]	DB3077 [1.0]
	G171	ABh	DB3078 [7.6]	DB3078 [5.4]	DB3078 [3.2]	DB3078 [1.0]	DB3079 1[7.6]	DB3079 [5.4]	DB3079 [3.2]	DB3079 [1.0]	...	...	DB3095 [7.6]	DB3095 [5.4]	DB3095 [3.2]	DB3095 [1.0]

每个 RAM 地址存储 4 个像素，每个像素以 4 灰度级进行编码。 STM32 的 SPI 配置为以高位在前的方式来设置传输，如 E-Paper 显示模块所要求的。

图片需要按这些限制进行编码。详情请参见*第 1.4 节：图片创建和尺寸压缩*。

1.4 图片创建和尺寸压缩

要在 E-Paper 模块上显示的图片须以巧妙的方式创建，以得到适当格式的相应软件常数。它应该易于处理并在 E-Paper 模块上显示。

此图片存储在 STM32L053 的内部非易失性 Flash 程序区。因此图片必须以特定格式进行压缩，编码为 1 比特每像素，以节省存储空间。这样，图片将被编码为黑白的而不再是 4 灰度级的图片。

由于 E-Paper 显示器 RAM 模块需要接收 2 比特每像素编码模式，因此需要从图片存入非易失性 Flash 程序存储器开始进行扩展处理。此操作将由软件管理 （请参考章节*第 1.5 节：图片数据扩展，以载入嵌入式 E-Paper RAM*）

1.4.1 图片帧

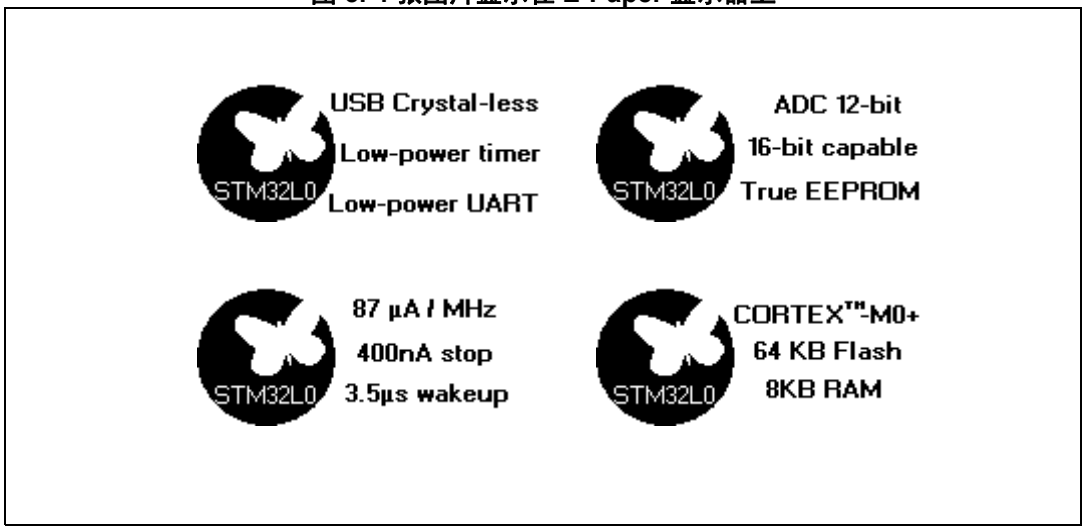
图片帧可进行编辑，例如采用 Windows 中有名的 “ 画板 ” 软件。首先须编辑空白图片 （如*图 4：图片帧*中）。空白帧为 172x72 像素。

图 4. 图片帧

然后必须将要显示的图片放入此帧中。如果要在图片中出现文本，可使用专门字体 “ 小字体 ”，它提供了简便的文本显示。

本应用笔记的示意图片显示在*图 5：4 张图片显示在 E-Paper 显示器上*中。

图 5. 4 张图片显示在 E-Paper 显示器上



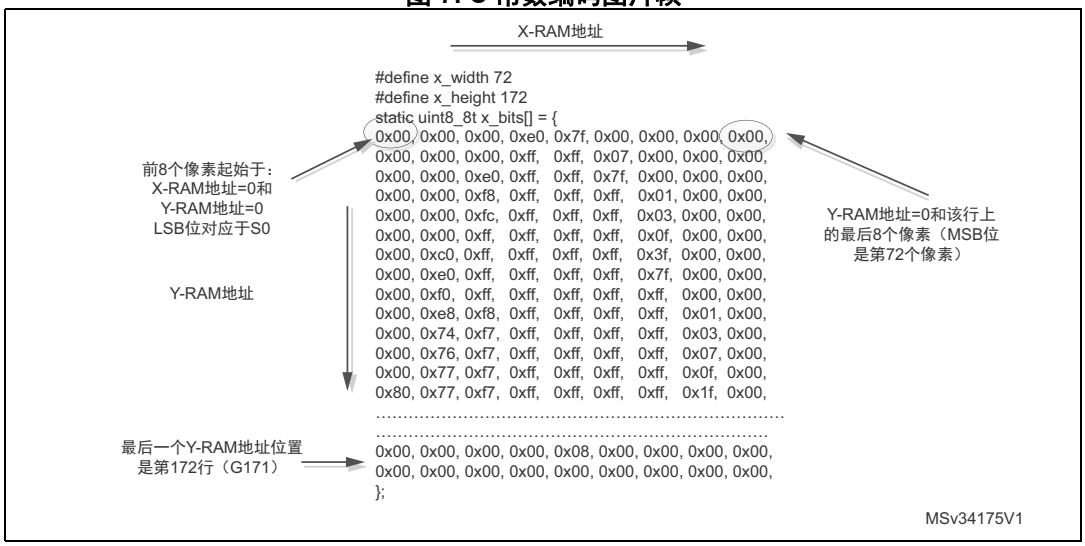
为了在编码图片 （以一种简便方式，可最小化数据处理）时得到常数，可将图片向右旋转 90°（*图 6：图片向右 90° 旋转*）。这样，如*图 7* 中所示，编码成常数的字节数量将对所有像素进行编码，且为全帧像素总数的整数倍。

图 6. 图片向右 90° 旋转

然后，必须利用 PAINT 将每张图片存储为 .bmp 图片格式和单色位图类型。

为了直接得到编码要显示图片的 C-constant，可将图片存储为 XBM-X11 格式，例如可使用免费软件 Xnview。这里是进行这种操作后得到的文件格式 （*图 7：C 常数编码图片帧*）。此常数可在 C 项目中直接使用和声明。

图 7. C 常数编码图片帧



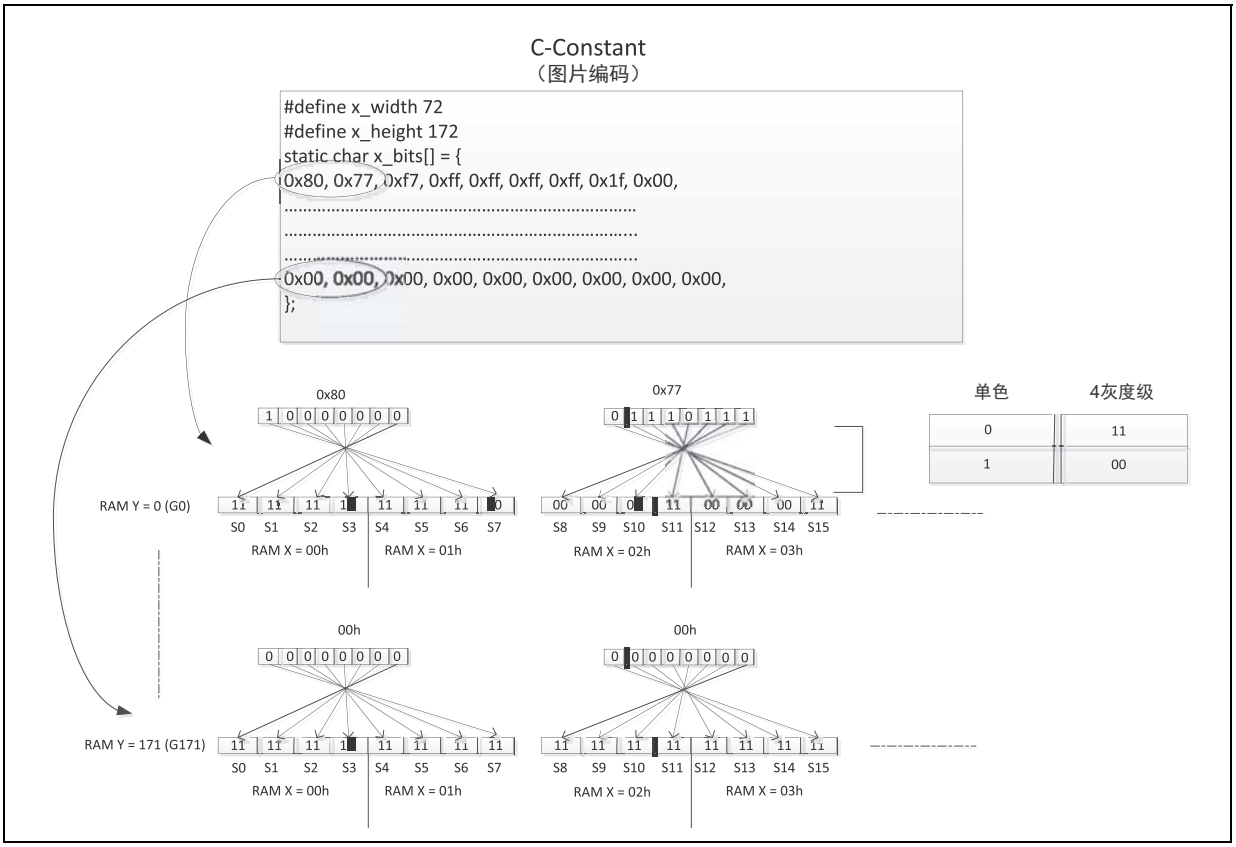
利用此图片格式可将每张图片的大小减少 1/2，意味着图片大小为 1.5KB，而不是 4 灰度级模式下的 3KB。

1.5 图片数据扩展，以载入嵌入式 E-Paper RAM

该软件可高效读取生成在 xbm 文件中的常数并显示它。根据应用，软件和 E-Paper 显示器配置具有不同的最优化轴。其中一些如*第 2 节：固件说明*中所示。

图 8：数据扩展，以加载 E-Paper RAM 说明了被载入 E-Paper RAM 模块的 C-constant（将图片编码为单色的 1 比特格式）是处理方式。

图 8. 数据扩展，以加载 E-Paper RAM



此软件使实现此数据扩展操作的数据处理时间最短。如果从 C-Constant 中读取的数据等于 0，则软件直接在 X 和 Y RAM 地址指针指向的相应 RAM 地址写入 2 字节 FF 数据。实际上，在 RAM 中白度在 4 灰度级下编码为 11b，而在 C-constant 下编码为 0b。以 C-Constant 编码的每个字节表示 8 个像素和 2 个 RAM 地址，而 4 灰度级下它们每个则编码 4 像素。

如果至少有一个像素由 C-Constant 值定义为黑像素，则启动软件数据处理，将 1 比特单色编码扩展为 4 灰度级编码，如*图 8：数据扩展，以加载 E-Paper RAM* 中所示。

当其处于数据输入模式时，每次数据以 SPI 模式写入时， X 和 Y RAM 地址指针有 E-Paper 显示模块的硬件管理 （更多详细信息，请参考 GDE021A1 E-Paper 显示器说明）。

2 固件说明

本应用笔记基于一个软件 （参考 STSW-STM32152），此软件在 STM32L053 探索套件上运行。本章描述取址 E-Paper 显示模块和管理图片尺寸压缩 / 扩展以节省图片存储空间的主要函数和软件功能。

软件以循环模式显示 4 张图片，每次显示刷新周期之间有 5s 的延迟。

2.1 系统配置

STM32L053 探索套件被配置为在 16MHz （来自内部 HSI16 RC- 振荡器）下运行。

SPI 在 2MHz 下运行，与 E-Paper 显示模块之间进行通信，来发送命令和数据。

systick 用来控制每张图片显示刷新之间的延迟，并管理 E-Paper 初始化。

2.2 中断源

软件中唯一使用的中断源是 systick 中断，它能够使计数器递增，对 HAL_Delay() 函数进行管理。

2.3 E-Paper 电源

main.c 文件中的 GPIO PB10 用于将 E-Paper 电源切换为 ON。此后本应用中它始终保持通电。

2.4 主要软件函数说明

本应用笔记中使用的高层软件函数在*表 3：高层软件函数*中显示和描述。

每次启动显示处理过程时， E-Paper 的 CS （芯片选择）引脚由 MCU GPIO 控制。 HAL 宏可直接控制它：

EPD_CS_LOW() 或 EPD_CS_HIGH()

表 3. 高层软件函数

函数名称	说明
BSP_EPD_DrawImage(uint16_t Xpos, uint16_t Ypos, uint16_t Xsize, uint16_t Ysize, uint8_t *pdata)	此函数调用用于显示特定图片到 E-Paper 显示器的其他高层函数。
BSP_EPD_RefreshDisplay(void)	此函数用来启动 E-Paper 刷新命令。它等待由 E-Paper 模块声明的 BUSY 信号。
EPD_RESET_HIGH() EPD_RESET_LOW()	由于有 MCU GPIO，此宏可用来为 E-Paper 显示器生成复位信号。
BSP_EPD_Init()	配置 E-Paper： - 睡眠模式 - RAM X 起始 / 结束地址：00h/11h - RAM Y 起始 / 结束地址：00h/ABh - RAM X 计数器：00h - RAM Y 计数器：00h - 禁用 RAM 旁路和 GS0 到 GS3 （通路），用于显示转换 - 显示更新：CLK ON + Charge Pump ON + 图形显示

下面只详细介绍了最相关的低层函数。

- **EPD_IO_WriteReg 函数**

  此函数实现发送命令到 E-Paper 显示模块所需的全部操作。

表 4. EPD_IO_WriteReg 函数

函数名称	EPD_Write_Com
原型	void EPD_IO_WriteReg(uint8_t Reg)
函数说明	由于具有 SPI MOSI 线，可发送命令到 E-Paper 显示模块
输入参数	发送到 E-Paper 显示模块的命令。
输出参数	无
返回值	无
规定的前提条件	无
调用的函数	SPIx_Write(), EPD_CS_LOW(), EPD_CS_HIGH(), EPD_DC_LOW()

此函数将传输识别为命令的单个字节 （关于命令值及其意义的更多详细信息，请参考 GDE021A1 E-Paper 显示器说明）

- **EPD_IO_WriteData 函数**

  此函数实现发送数据到 E-Paper 显示模块所需的全部操作。

表 5. EPD_IO_WriteData 函数

函数名称	EPD_Write_Com
原型	void EPD_IO_WriteData(uint8_t RegValue)
函数说明	由于具有 SPI MOSI 线，可发送数据到 E-Paper 显示模块
输入参数	发送用于显示器配置的数据，或载入 RAM 模块的要显示图片的数据到 E-Paper 显示模块。
输出参数	无
返回值	无
规定的前提条件	无
调用的函数	SPIx_Write(), EPD_CS_LOW(), EPD_CS_HIGH(), EPD_DC_HIGH()

此函数将传输单个数据字节，该数据可能是用于配置 E-Paper 寄存器的数据，也可能是要显示图片对应的数据 （关于命令值及其意义的更多详细信息，请参考 GDE021A1 E-Paper 显示器说明）。

- **gde021a1_DrawImage 函数**

此函数准备 2 个扩展字节的数据，对从 1 比特 XBM 文件中读取的 8 比特数据进行扩展。它采用适当的方式将 2 比特像素重排为字节，填充到 RAM 中 （参考*图 8：数据扩展，以加载 E-Paper RAM*）。

表 6. Gde021a1_DrawImage 函数

函数名称	Processing_8_pixels
原型	gde021a1_DrawImage(uint16_t Xpos, uint16_t Ypos, uint16_t Xsize, uint16_t Ysize, uint8_t *pdata)
函数说明	进行双字节扩展和字节重新排序，并将处理过的数据依像素信息填充到 E-Paper 显示的 RAM 的正确位置
输入参数	发送到 E-Paper 显示模块的数据，矩阵 X/Y 的起始和结束地址及其大小。
输出参数	无
返回值	无
规定的前提条件	无
调用的函数	EPD_IO_WriteReg(uint8_t Reg), EPD_IO_WriteData(uint8_t RegValue)

3 可能的固件优化

优化代码和 / 或降低功耗。

3.1 E-Paper 功耗管理

本应用笔记中，软件将 E-Paper 的电源永久地切换为 ON。对于不同应用，功耗可能很重要，本软件希望能够显著降低功耗。 E-Paper 提供了 2 种模式如*表 7：E-Paper 显示模块低功耗模式*中所示。

表 7. E-Paper 显示模块低功耗模式

E-Paper 低功耗模式	功耗 （典型值） @3,3V	贡献	缺点
睡眠模式	35 uA	DC/DC 转换器关闭 无时钟，无输出负载，确保 RAM 保存	消耗
深度睡眠模式	2 uA	DC/DC 转换器关闭 无时钟，无输出负载，无 RAM 保存	不再保存 RAM，这意味着唤醒后部分图片不可能刷新。对于要显示的新图片，RAM 内容需要完全重载。

通常大部分应用中， E-Paper 刷新速率应当足够长，不忙时最好使 E-Paper 进入深度睡眠模式。从功耗角度来说，通过 SPI 将要显示的图片 （3096 字节）重载到 RAM 中的开销可能更小。相反，如果刷新速率很快，最好进入睡眠模式而不是深度睡眠模式，以保持 RAM 内容，允许 MCU 在较短的运行时间内进行部分显示区域刷新。用户须根据其目标、从功耗和刷新图片的时间方面来评估最佳解决方案。

用来运行本应用笔记中软件的 STM32L053 探索套件允许试用一个 MOSFET，从特定 GPIO （PB10）彻底切断功耗，与 E-Paper 主电源不连接。这种情形下，屏幕确实仍然在显示图片，但模块不消耗任何电流。它是一种更高级的深度睡眠模式。这种情况下，当应用切换为 ON 以及重新激活电荷泵时，模块可能消耗更多 （电流）。

3.2 E-Paper 显示 RAM 的部分更新

应用可能需要不时地部分刷新显示。这种情况下， E-Paper 显示模块的数字接口可配置 X 和 Y RAM 地址指针和相应的计数器，仅加载发生改变的那部分显示。

如果在 E-Paper 初始化阶段设置为睡眠模式而不是深度睡眠模式 （并且 E-Paper 上保持供电），那么其余显示的数据仍然在 E-Paper RAM 中。它有两个作用，可以减少 MCU 传输到 E-Paper RAM 的字节数量，并因此减少 MCU/CPU 执行此操作的时间。

4 版本历史

表 8. 文档版本历史

日期	版本	变更
2014 年 10 月 15 日	1.0	初始版本

表 9. 中文文档版本历史

日期	版本	变更
2015 年 11 月 30 日	1.0	中文初始版本

重要通知 - 请仔细阅读

意法半导体公司及其子公司 （“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。 ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 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。

本文档中的信息取代本文档所有早期版本中提供的信息。