

## 1 介绍

本篇应用笔记描述了如何在 K32L2A 上使用 FlexIO 来驱动 8080 总线接口 LCD 模块。

FlexIO 模块是一个高度可配置的外设，允许用户实现各种功能。根据不同的模块版本，它可以执行以下操作：

- 模拟串行通信协议，如 UART, SPI, I2C, I2S。
- 模拟并行通信协议，如摄像头接口, Motorola 68 K 总线, Intel 8080 总线。
- 产生 PWM 波形。
- 实现逻辑功能。
- 实现状态机功能。

图形 TFT LCD 模块广泛用于需要 GUI 功能的嵌入式应用中。8080 并行总线是 TFT LCD 模块的常用接口。本文介绍了如何使用 FlexIO 模块来模拟 8080 并行总线并通过 8080 总线接口驱动图形 TFT LCD。

该应用程序基于最近发布的 MCU K32L2A。该 Kinetis 子系列是基于 Arm® Cortex®-M0+ 的微控制器，支持超低功耗，并集成了无晶体 USB、大容量存储器（512 KB flash 和 128 KB RAM）、演进的低功耗外设、安全功能等。

## 2 FlexIO 概述

### 2.1 功能和模块框图

K32L2A 的 FlexIO 模块版本提供以下功能：

- 具有发送、接收和数据匹配模式的 32 位移位寄存器（也称为移位器）阵列，用于连续传输的双缓冲结构以及支持大传输量的级联机制。
- 高度灵活的 16 位定时器，支持各种内部或外部触发、复位、启用和禁用条件。
- 自动开始/停止位生成和检查。
- 4、8、16 或 32 多位移位宽度，用于并行接口支持。
- 中断、DMA 或轮询的发送/接收操作。
- 可编程波特率，在停止模式下支持异步操作。
- 可编程逻辑模式，用于在片上集成外部数字逻辑功能或组合引脚/移位器/定时器功能以生成复杂的输出。
- 可编程状态机，用于从 CPU 卸载基本的系统控制功能。

图 1 显示了该模块的高级概览：

### 目录

1	介绍.....	1
2	FlexIO 概述.....	1
2.1	功能和模块框图.....	1
2.2	内部逻辑连接.....	2
2.3	移位器和定时器.....	2
2.4	常规配置和操作.....	3
2.5	FlexIO 并行传输.....	3
3	LCD 模块的 8080 并行总线序列.....	4
4	模拟 8080 总线和驱动 LCD 模块.....	5
4.1	配置 FlexIO 以模拟 8080 总线.....	5
4.2	硬件平台.....	9
4.3	FlexIO 配置与硬件连接.....	10
4.4	运行演示.....	13
5	结论.....	14



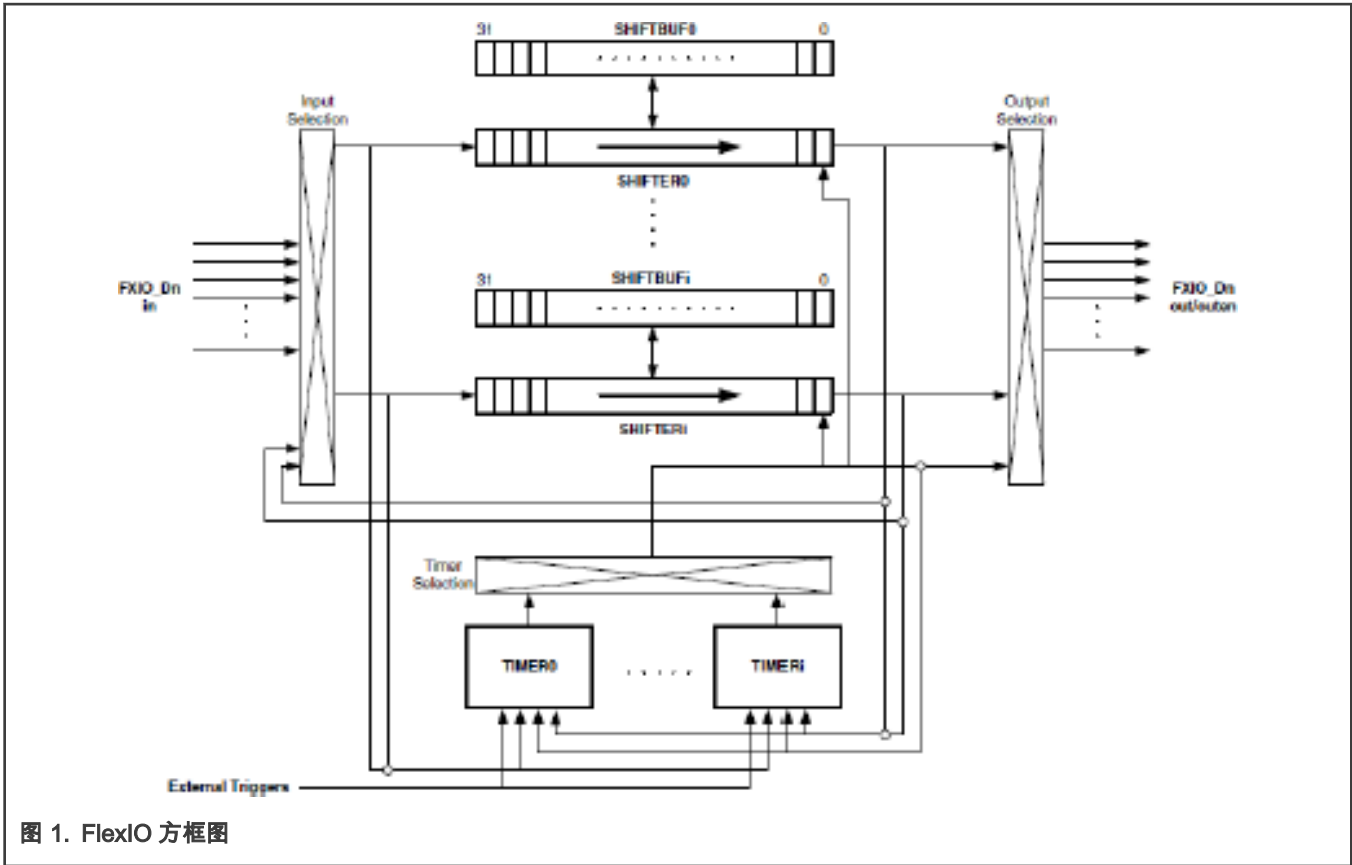


图 1. FlexIO 方框图

## 2.2 内部逻辑连接

为了满足各种要求，内部逻辑连接非常灵活和复杂。以下是一些功能。

- 可以将任何引脚分配给移位器进行输入或输出。
- 可以将任何定时器分配给移位器进行移位控制。
- 可以将任何引脚分配给定时器以进行定时器输出或输入。
- 定时器触发可以源自移位器标志、引脚或 FlexIO 模块外部。
- 移位可以发生在移位时钟的上升沿或下降沿。
- 引脚方向和极性是可配置的。
- 触发极性是可配置的。
- 定时器的启用、禁用、递减和复位条件可能源自触发器、引脚、相邻定时器。

有关详细信息，请参见相应的 MCU 参考手册。

## 2.3 移位器和定时器

以上描述显示了 FlexIO 硬件资源由移位器、定时器和引脚组成。可以从 PARAM 寄存器中读取给定 MCU 的这些资源数量。例如，K32L2A 的 FlexIO 模块具有 8 个移位器，8 个定时器和 32 个引脚。

发送和接收是移位器的两个主要模式。当移位器处于发送模式，它从缓冲寄存器装载数据，然后将数据输出至分配的一个或多个引脚。当移位器处于接收模式，它从分配好的一个或多个引脚中输入数据，然后将数据储存进缓冲寄存器。移位器分配的定时器控制着装载、储存以及移位操作。

还可根据需要将定时器设置为不同的工作模式，包括双 8 位计数器波特率/位模式，双 8 位计数器 PWM 模式和单 16 位计数器模式。双 8 位计数器波特率/位模式通常实现一个数据发送器。在这种模式下，16 位定时器的低 8 位对模块时钟源进行分频，以生

成所需的波特率，而高 8 位对帧的移位位进行计数。使能后，定时器会从比较寄存器中加载初始值，并开始递减计数。当低 8 位递减为零时，定时器的移位时钟及其输出信号将被翻转以产生上升沿或下降沿。高 8 位的减量计数为 1。移位时钟驱动移位器。定时器输出信号通常驱动时钟输出引脚，例如 SPI 主设备的 SCK 和 8080 总线的 WR。之后，低 8 位重新加载初始值以开始另一个递减周期。两个递减周期组成一个移位周期，它驱动移位器移动一个节拍。当整个 16 位的减量计数为零时，移位中的所有数据位都将移出。然后，在另一个传输帧之前禁用定时器。

## 2.4 常规配置和操作

FlexIO 可以模拟各种通信协议。但是，为了模拟专用外设并处理发送和接收过程，必须使用软件配置 FlexIO。

通常，为了实现主发送器，将移位器配置为发送模式，将所分配的定时器配置为双 8 位计数器波特率/位模式。定时器递减时钟源于模块时钟。定时器触发来自极性相反的移位器标志。通过轮询/中断/ DMA 填充移位器的缓冲区会清除移位器标志，从而使定时器可以开始递减计数。定时器的减少驱动移位器将数据移出并生成时钟输出信号。

为了实现接收器，将移位器配置为接收模式。定时器配置为双 8 位计数器波特率/位模式，用于同步主接收器，例如 8080 总线读取实现。此定时器模式还用于异步接收器，例如 UART 接收器。接收过程类似于主发送器，但数据被移入移位器而不是移出。对于同步接收器，分配的定时器以单个 16 位计数器模式配置，例如 SPI 从接收器。递减时钟源于引脚输入，例如 SPI SCK 信号。定时器触发来自另一个引脚，例如 SPI CS 信号。主设备使能定时器并通过引脚控制减量。同样，定时器的减少会驱动移位器移入数据。

为了易于使用，恩智浦在 KSDK ( Kinetis 软件开发套件 ) 中提供了 API 驱动程序和驱动程序示例。用户还可以在应用笔记中找到这些实现的详细说明和类似的 API。

## 2.5 FlexIO 并行传输

早期的 FlexIO IP 版本仅支持串行传输，例如 KL17 / KL27 / KL33 / KL43。而更高版本同时支持串行和并行传输，例如 K32L2A。

数据始终在移位器中从 MSB 移到 LSB。在串行传输模式下，数据从 LSB ( 位 0 ) 逐位移出，从 MSB ( 位 31 ) 逐位移入。此过程如图 2 所示。

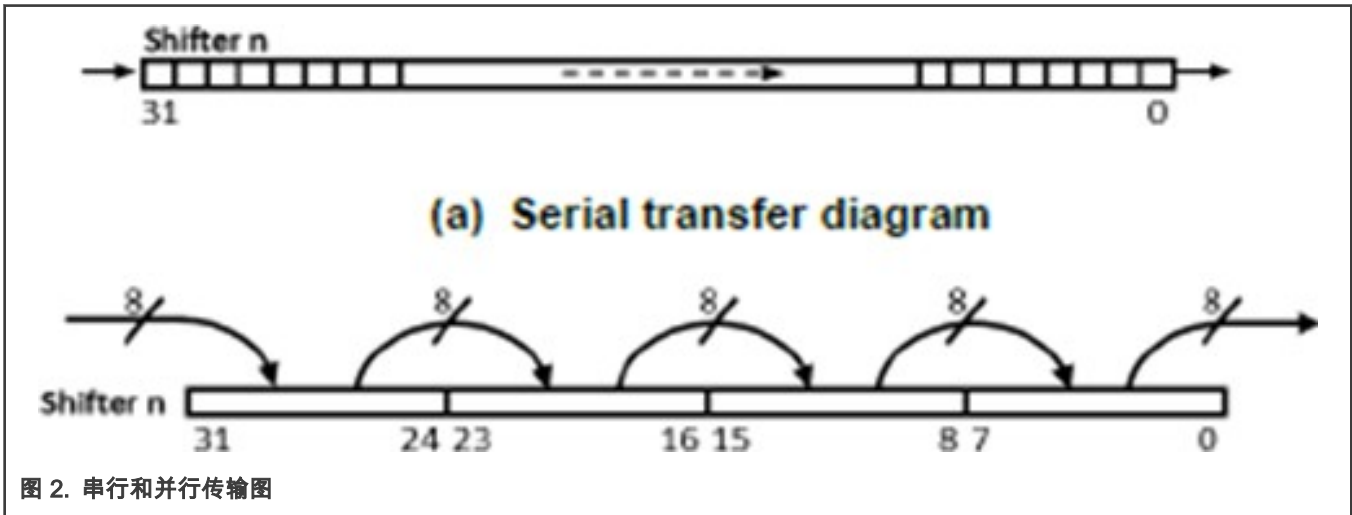


图 2. 串行和并行传输图

在并行传输模式下，数据从移位器的  $n$  个 LSB 移出，并从  $n$  个 MSB 移入，其中  $n$  为并行总线宽度。图 3 显示了  $n=8$  的用例。下面描述并行传输模式：

1. 在每个移位时钟上将数据移位  $n$  位，其中  $n$  是配置的总线宽度。
2. 支持 4、8、16 或 32 总线宽度。
3. 将多个移位器组合在一起进行串联以支持较大的传输大小，并使用 DMA 方法访问移位缓冲寄存器以进行高速传输和低功耗操作。图 3 显示了移位器串联图，其中其他移位器用作 FIFO。

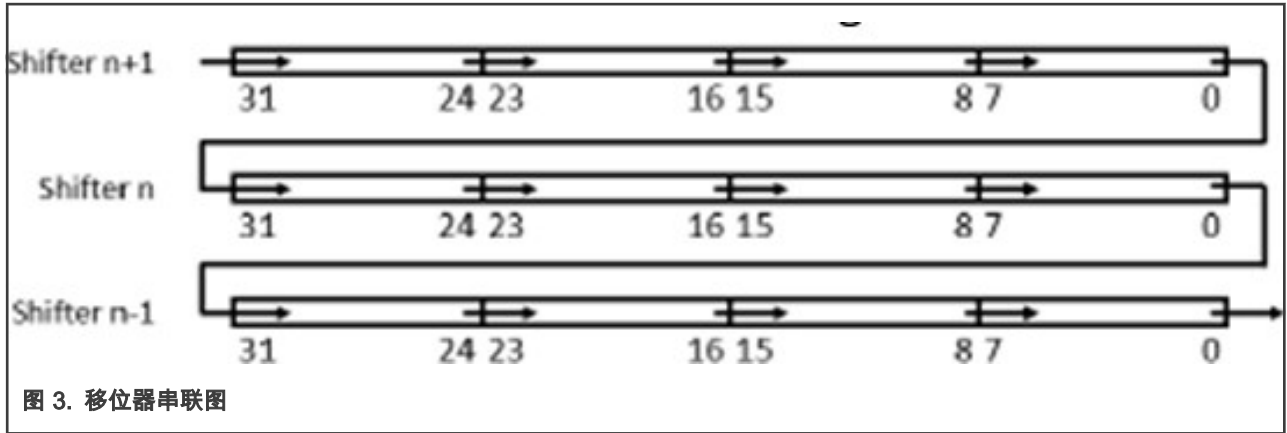


图 3. 移位器串联图

4. 仅特定的移位器 ( SHIFTER0 和 SHIFTER4 ) 支持并行输出到 FlexIO 引脚。但是，除 SHIFTER0 以外的所有移位器均支持输出到相邻的低阶移位器。
5. 类似地，仅特定的移位器 ( SHIFTER3 和 SHIFTER7 ) 支持来自 FlexIO 引脚的并行输入。但是，除 SHIFTER7 以外的所有移位器均支持来自相邻高阶移位器的输入。
6. 任何 FlexIO 引脚都可以是并行输出/输入引脚。但是，对于特定用途，引脚索引必须是连续的，例如，对于 8 位总线，引脚 0 至引脚 7，或引脚 1 至引脚 8，依此类推。

### 3 LCD 模块的 8080 并行总线序列

通常，用于 LCD 模块的 8080 并行接口包括 8 或 16 条双向数据线（数据总线），一条片选线（CS），一条写锁存线（WR），一条读锁存线（RD），一条数据/命令选择线（RS）。

CS，WR，RD 和 RS 均为低电平有效。CS 的低电平选择从设备。WR 线的上升沿是数据写锁存信号（时钟）。RD 线的上升沿是数据读取锁存信号（时钟）。正在进行写入序列时，RD 应为高电平。同样，在进行读取序列时，WR 应为高电平。RS 是数据/命令选择信号。RS 的低电平表示命令（或地址）传输。RS 的高电平表示数据传输。RS 也称为 DC 或 ALE（地址锁存启用）。

在写入/读取传输开始时，命令/地址写入序列指定目标地址。数据传输可以是一个或多个节拍。

图 4 显示了写入时序。在 0 节拍命令类型下，在写入序列期间发生数据传输。

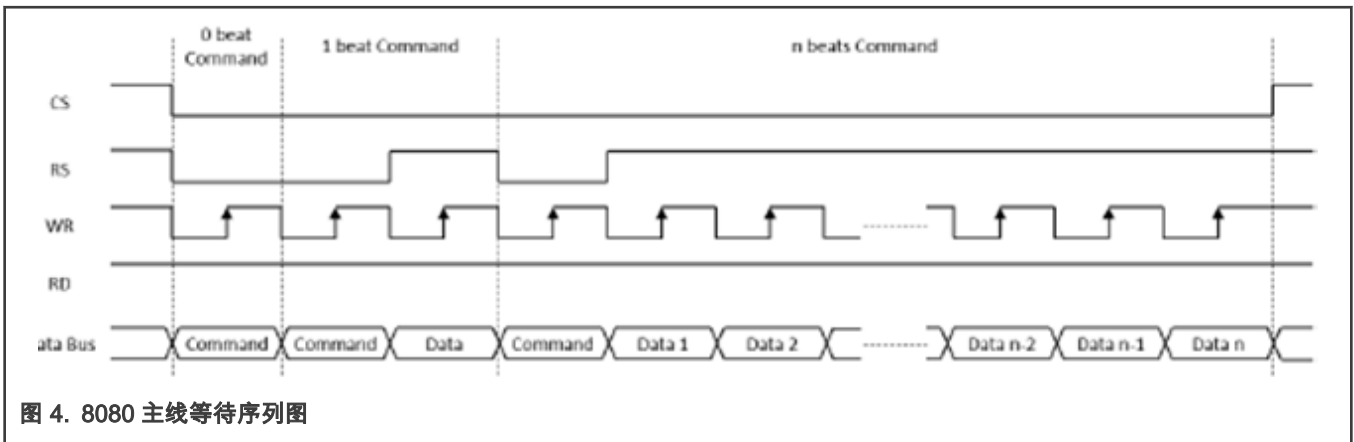
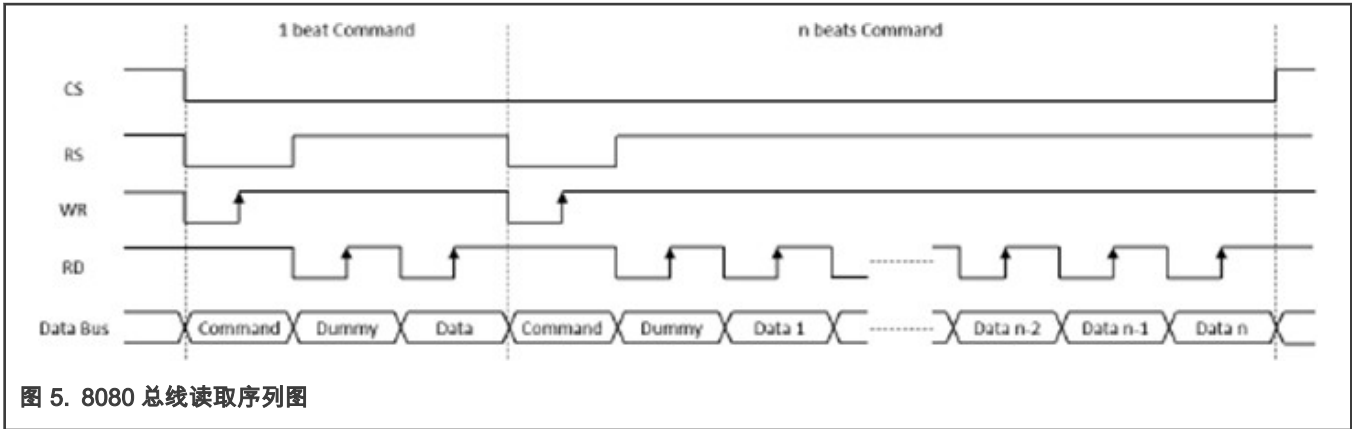


图 4. 8080 总线等待序列图

图 5 显示了读取时序。根据总线从机，在命令写入节拍和第一个数据读取节拍之间可能会出现虚读取节拍。



## 4 模拟 8080 总线和驱动 LCD 模块

### 4.1 配置 FlexIO 以模拟 8080 总线

用户可以配置 FlexIO 以不同的方式模拟 8080 总线，例如不同的总线宽度、波特率、串联移位器的数量、特定的移位器、引脚和要使用的定时器。根据应用程序要求设置适当的配置。

此应用实现了 8 位和 16 位宽度的总线。由于 8 位和 16 位总线实现之间的模块配置和软件驱动程序非常相似，因此以下各节仅介绍 8 位实现。

读写功能均已实现。通过将移位器配置为发送模式来实现写入功能。通过将移位器配置为接收模式来实现读取功能。

1 拍传输驱动程序和多拍传输驱动程序均已实现。1 拍传输传输小尺寸数据，例如配置 LCD 驱动器 IC 寄存器。由于仅使用一个移位器，所以移位器级联不用于 1 拍传输。一个传输序列需要定时器仅生成一个移位时钟。同时传送 8 位进/出。在此应用程序中，轮询方法访问移位缓冲区以进行 1 拍传输。

多拍传输支持较大的传输大小，例如将帧数据传输到 LCD 模块。一个传输序列需要定时器生成多个移位时钟。每一个传输序列的拍数与串联的移位器数和总线宽度有关。所有移位器均为 32 位大小。一个移位器最多支持 8 位宽度总线的四拍传输。两个移位器最多支持八拍，依此类推。在此应用中，使用了所有八个移位器，并且 8 位宽度的总线支持 32 个节拍。DMA 方法在此应用程序中访问移位缓冲区以进行多拍传输。

在实际的应用程序中，写入和读取不会同时发生。一拍和多节拍传输也永远不会同时运行。此外，移位器和引脚在不同功能之间共享。因此，用户应在每次功能切换时重新配置 FlexIO 模块。

读写有四种实现方式，即 1 拍和多拍传输，如下所示：

1. 1 拍写入
2. 1 拍读取
3. 多拍写入
4. 多拍读取

在此应用中，1 拍写入被广泛使用，例如配置 LCD 驱动器 IC 的寄存器，传输较小的帧数据以及所有四种实现方式中传输命令节拍。多拍写入主要将较大的帧数据传输到 LCD。在此应用程序中不使用 1 拍读取和多拍读取。

由于 1 拍传输是多拍传输的简单版本，因此，以下各节简要介绍了多拍写入和多拍读取。

#### 4.1.1 多拍写入

图 6 显示了用于多拍写传输的 FlexIO 模块配置。

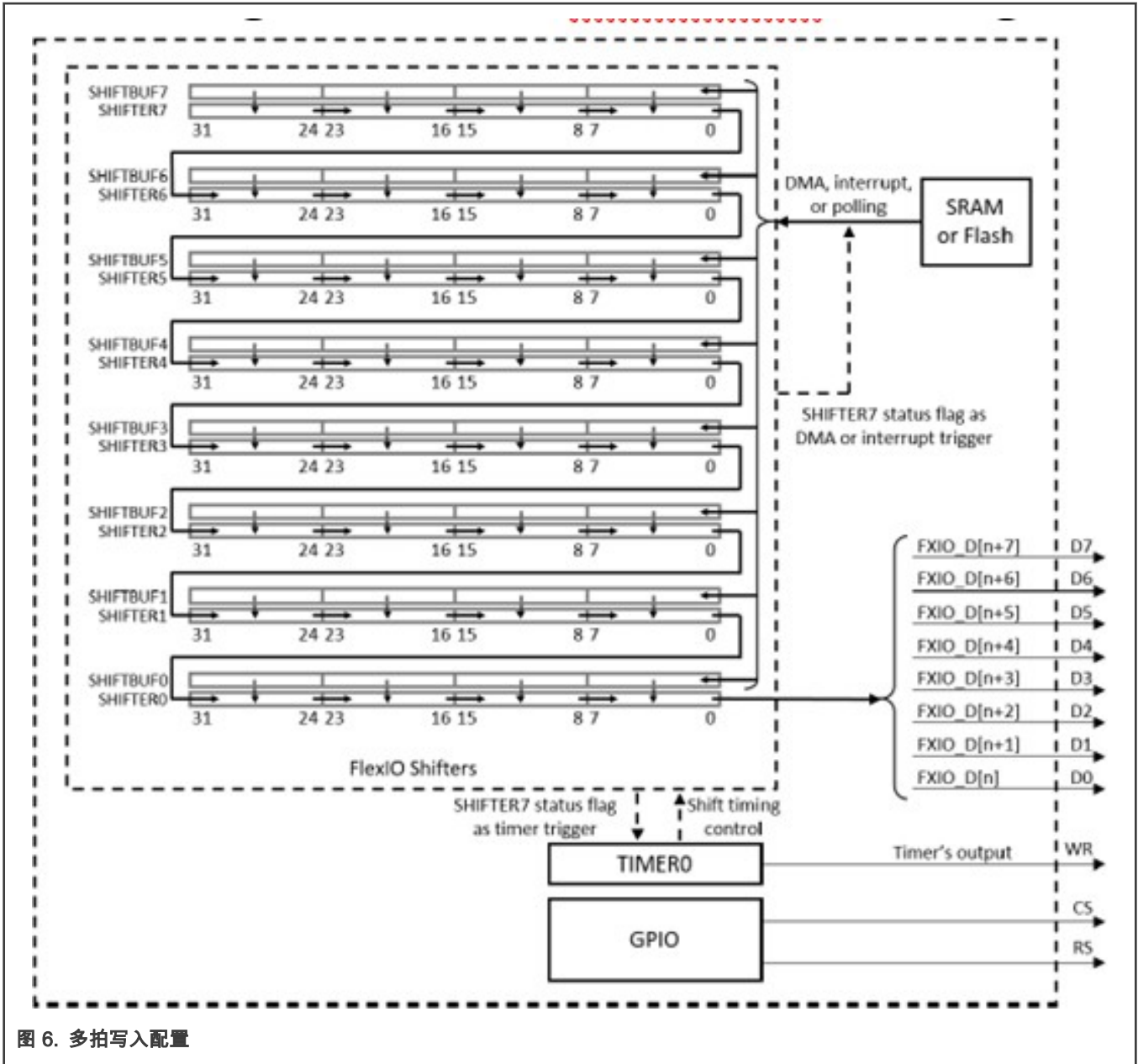


图 6. 多拍写入配置

在该配置中，所有八个移位器串联在一起。TIMER0 用于产生移位时钟和 RD 信号。D0-D7 和 RD 基于 FlexIO 引脚。其他的 GPIO 引脚用于驱动 CS 和 RS 信号。SHIFTER0 状态标志用于触发 TIMER0 并生成 DMA 请求。下面的步骤描述了使用 DMA 方法的多拍写入。

1. 配置 DMA、FlexIO 模块和 GPIO。启用 SHIFTER7 状态标志的 DMA 请求。
2. 启用请求后，DMA 请求立即响应。DMA 将数据从 SRAM 或 flash 复制到移位缓冲区 SHIFTBUF0~SHIFTBUF7。每一个 DMA 请求总共复制 32 个字节。

注意

SHIFTBUF0~SHIFTBUF7 的地址是连的。

3. 填充移位缓冲区会清除移位器状态标志，从而使能所选的 TIMER0。
4. TIMER0 发出加载事件信号，数据从移位缓冲器 SHIFTBUF0~SHIFTBUF7 加载到移位器 SHIFTER0~SHIFTER7。

5. 加载事件清空移位器缓冲区，移位器缓冲区设置移位器状态标志并触发另一个 DMA 请求。DMA 将新数据填充到 SHIFTBUF0~SHIFTBUF7 中，并再次清除移位器状态标志。
6. 加载事件后，TIMER0 开始递减计数。它产生定时器移位时钟以及减量，以控制移位器将数据移出，并产生定时器输出以驱动 WR 信号。
7. 配置了 32 个移位时钟后，TIMER0 递减计数为零，发生比较事件。然后短时间内禁用 TIMER0。
8. 由于移位缓冲区包含有效数据，并且此时 SHIFTER7 状态标志为零，因此 TIMER0 再次使能。然后，重复步骤 3-8。
9. 将所有数据复制到移位缓冲区后，DMA 完成主循环。在上一个比较事件之后，不再复制任何数据，并且不会触发 TIMER0。产生 DMA 中断以指示主循环已完成。
10. 如果总传输大小不能被 32 整除，则使用轮询方法以 1 拍模式传输其他字节。

另外，CS 在传输序列之前被软件拉低，在传输序列之后被拉高。RS 在命令发送之前拉高，此后再次拉高。

#### 4.1.2 多拍读取

显示了用于多拍读取传输的 FlexIO 模块配置。

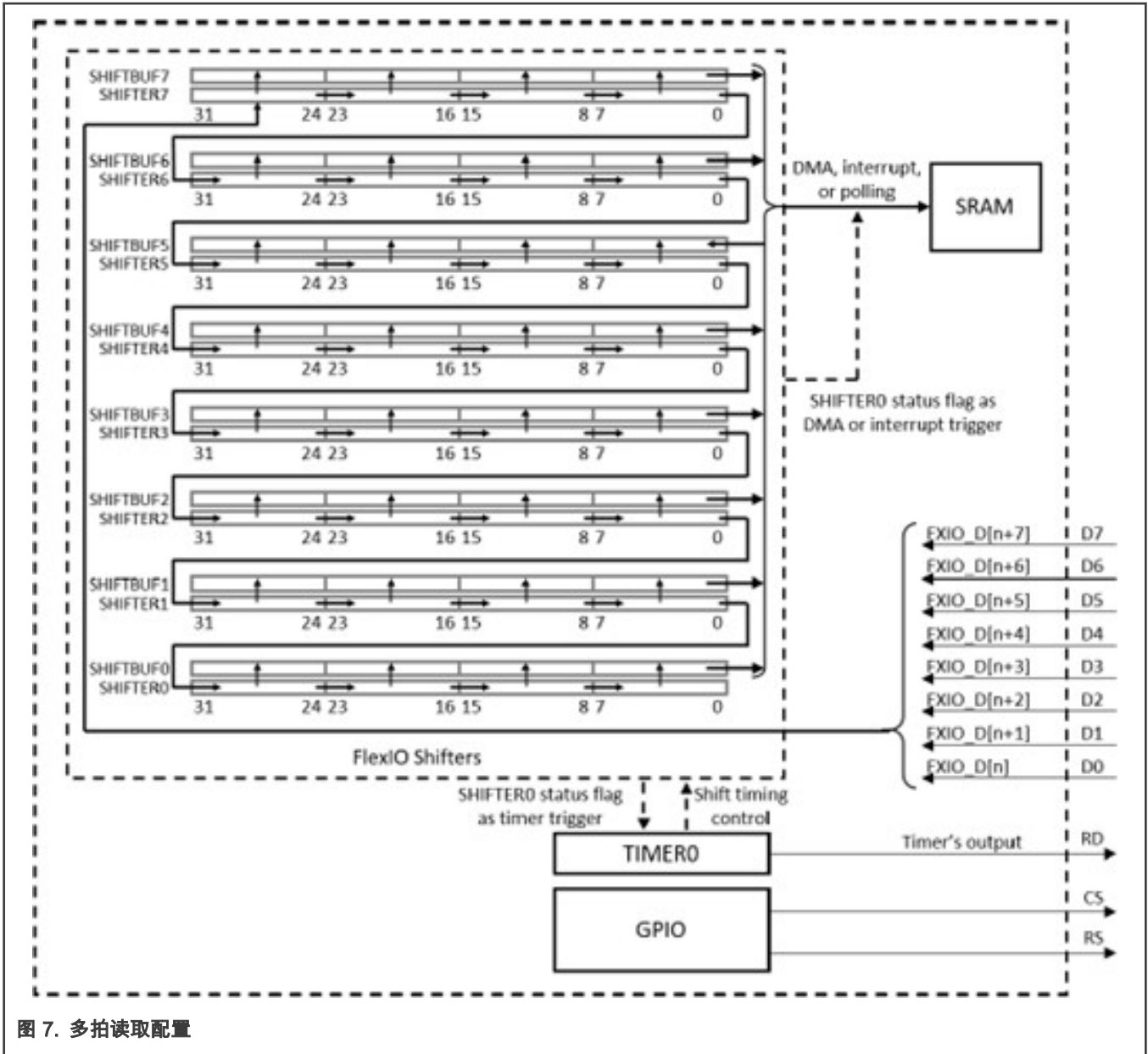


图 7. 多拍读取配置

在该配置中，所有八个移位器串联在一起。TIMER0 用于产生移位时钟和 RD 信号。D0-D7 和 RD 基于 FlexIO 引脚。其他的 GPIO 引脚用于驱动 CS 和 RS 信号。SHIFTER0 状态标志用于触发 TIMER0 并生成 DMA 请求。

以下各项描述了使用 DMA 方法的多拍读取传输过程。

1. 配置 DMA、FlexIO 模块和 GPIO。启用 SHIFTER0 状态标志的 DMA 请求。
2. TIMER0 在配置完成后立即启用，并开始递减计数。它产生定时器移位时钟以及减量以控制移位器将数据移入，并产生定时器输出以驱动 RD 信号。
3. TIMER0 的递减计数为零，并且在 32 个移位时钟后发生比较事件。然后，TIMER0 被禁用。
4. 在比较事件之后，TIMER0 发出存储事件信号。数据从移位器 SHIFTER0-SHIFTER7 存储到移位器缓冲区 SHIFTBUF0-SHIFTBUF7。
5. 存储事件填充移位器缓冲区，移位器缓冲区设置移位器状态标志并触发 DMA 请求。
6. DMA 请求得到响应。DMA 将数据从移位缓冲区 SHIFTBUF0-SHIFTBUF7 复制到 SRAM。每 1 个 DMA 请求复制 32 个字节。



7. DMA 读取操作清除移位器标志，从而再次使能 TIMER0。然后重复步骤 2 到步骤 7。
8. 主循环完成后，将再次触发 TIMER0 以移入最后 32 个字节。产生 DMA 中断以指示 DMA 主循环的完成。

**注意**

TDMA 主循环大小配置为不包括最后一个次循环，以避免在完成所有传输后启用 TIMER0。

9. 如果总传输大小不能被 32 整除，则使用轮询方法以 1 拍模式传输其他字节。

另外，CS 在传输序列之前被软件拉低，而在传输序列之后被拉高。RS 在命令发送之前拉高，此后再次拉高。

## 4.2 硬件平台

本应用以 FRDM-K32L2A 开发模块为例，如 图 8 所示。Arduino 插槽连接到 FlexIO 引脚，这使得该应用程序的硬件连接更加方便。

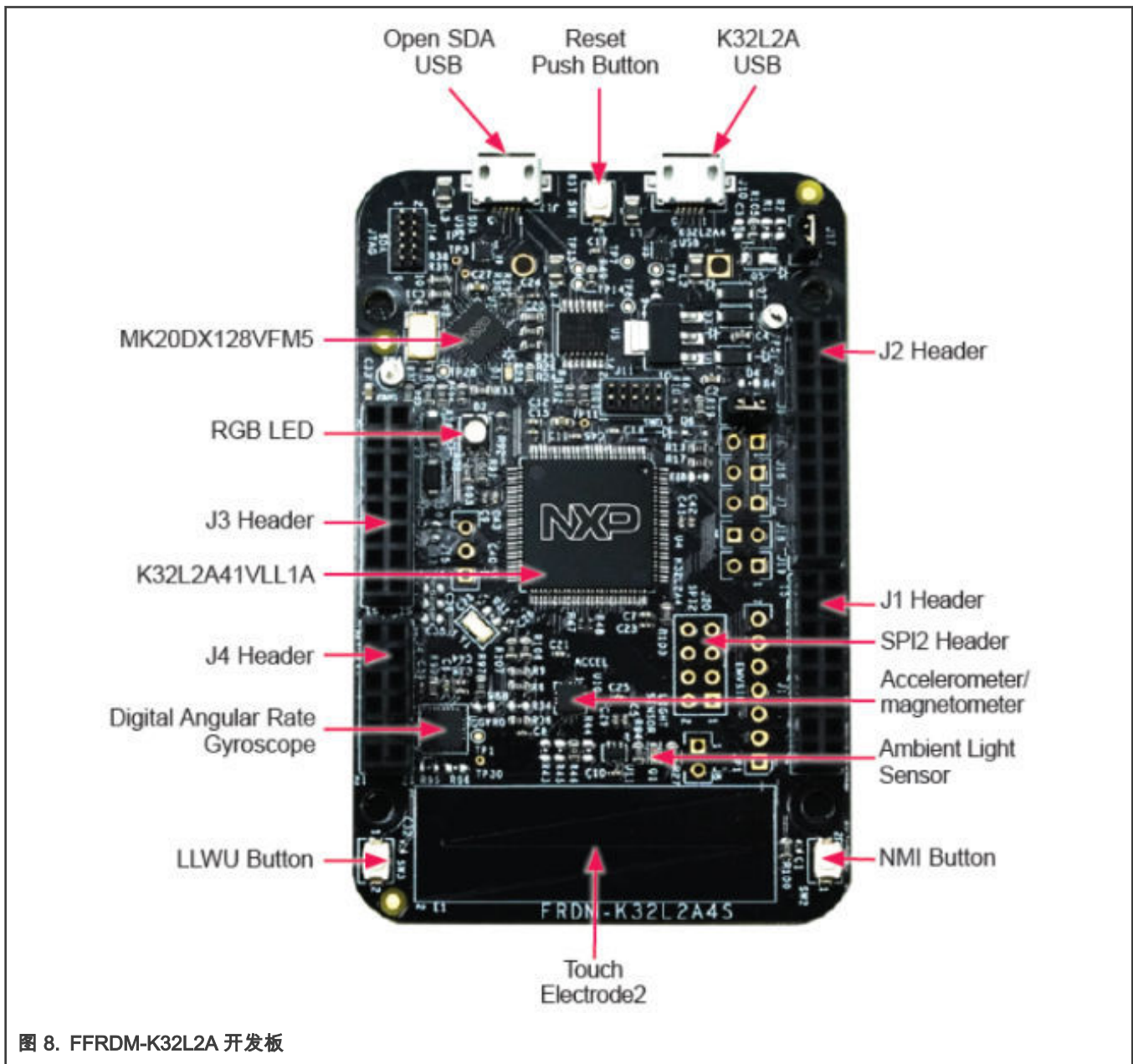


图 8. FFRDM-K32L2A 开发板

该应用中使用的 LCD 模块集成了 Himax LCD 驱动器 IC HX8357。它支持许多数据传输接口，包括 MIPI-DBI 8-/16-/18-/24 位并行接口（兼容 8080）、MIPI-DBI 串行接口、MIPI-DPI 8-/9-/16-/18-/24-数据线并行（RGB）接口和 MIPI-DSI 接口。在此应用中使用 8080 兼容的 8 位和 16 位 MIPI-DBI 并行接口。该应用程序使用引导捆扎方法，在模块 PCB 上使用多个电阻器来上拉或下拉 HX8357 接口选择引脚，以选择特定的接口。

此应用程序演示的系统框图如图 9 所示。

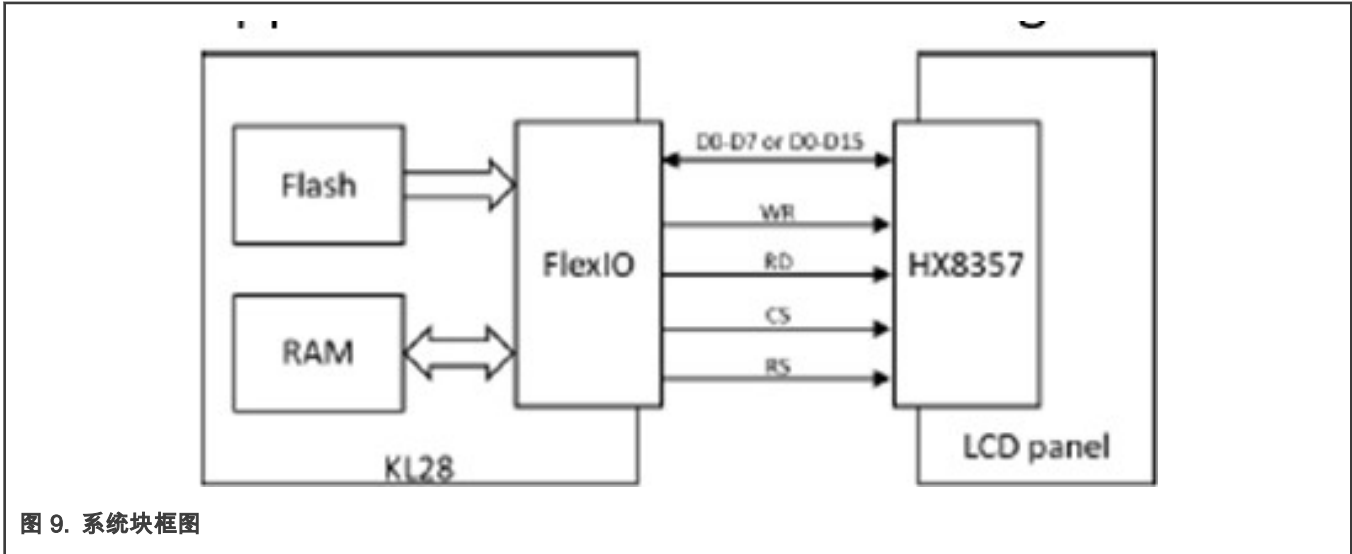


图 9. 系统块框图

### 4.3 FlexIO 配置与硬件连接

本节提供了 FlexIO 寄存器的详细配置以及 FRDM-K32L2A 模块和 LCD 模块之间的硬件连接。

#### 4.3.1 FlexIO 配置

表 1 至 表 4 提供了四种操作模式下 FlexIO 的主要寄存器配置。

表 1. 1 拍写入

寄存器	值	说明
SHIFTCFG0	0x0007_0100	配置移位器停止位禁用，启动位禁用，并启用 8 位并行宽度来加载数据。
SHIFTCTL0	0x0003_0002	从 TIMER0 配置移位时钟，在移位时钟的上升沿移位。将移位器的引脚配置为输出，引脚索引从 0 开始，并且引脚极性为高电平有效。将移位器模式配置为发送。
TIMCMP0	0x0000_0101	设置 TIMCMP [15:8] = 节拍数 $x2-1=1x2-1=1$ (对于节拍为 1)。 设置 TIMCMP [7:0] = 波特率分频器 $/2-1=4/2-1=1$ 。
TIMCFG0	0x0000_2200	定时器启用且不受复位影响时、FlexIO 时钟递减时、移位时钟等于定时器输出时、永不复位时、定时器比较时禁用时、触发高电平使能时、停止位禁用时、启动位禁用时，配置定时器输出逻辑为 1。
TIMCTL0	0x01C3_1081	将 SHIFT0 状态标志配置为定时器触发，并且触发极性为低电平有效。配置计时器的引脚为输出，引脚索引为 16 (WR)，引脚极性为低电平有效。将定时器模式配置为 8 位计数器波特/位。

表 2. 1 拍读取

寄存器	值	说明
SHIFTCFG7	0x0007_0000	从引脚配置移位器输入，禁用停止位，禁用开始位，并在使能的 8 位并行宽度上加载数据。
SHIFTCTL7	0x0080_0001	从 TIMER0 配置移位时钟，移位时钟下降沿进行移位。将移位器的引脚配置为输入，引脚序号从 0 开始，并且引脚极性为高电平有效。将移位器模式配置为接收。
TIMCMP0	0x0000_0101	设置 TIMCMP [15:8] = 节拍数 $x2-1=1x2-1=1$ (对于一拍时) 设置 TIMCMP [7:0] = 波特率分频器 $/2-1=4/2-1=1$ 。
TIMCFG0	0x0000_2220	在启用且不受复位影响、FlexIO 时钟递减、移位时钟等于定时器输出、永不复位、在定时器比较时禁用、在触发高电平时启用、在定时器禁用时启用停止位的情况下，配置定时器输出逻辑为 1 (用于禁用内部信号同步计时器)，并且禁用启动位。
TIMCTL0	0x1DC3_1181	将 SHIFT7 状态标志配置为定时器触发，并且触发极性为低电平有效。配置定时器的引脚为输出，引脚索引为 17 (RD)，引脚极性为低电平有效。将定时器模式配置为 8 位计数器波特/位。

表 3. 多拍写入

寄存器	值	说明
SHIFTCFG0...7	0x0007_0100	从下一个移位器的输出配置移位器输入，禁用停止位，禁用开始位，并在使能的 8 位并行宽度上加载数据。
SHIFTCTL0	0x0003_0002	从 TIMER0 配置移位时钟，在移位时钟的上升沿触发移位。将移位器的引脚配置为输出，引脚索引从 0 开始，并且引脚极性为高电平有效。将移位器模式配置为发送。
SHIFTCTL1...7	0x0000_0002	使用 TIMER0 配置发送，在移位时钟的上升沿触发移位。将移位器的引脚配置为禁止输出，引脚索引从 0 开始，并且引脚极性为高电平有效。将移位器模式配置为发送。
TIMCMP0	0x0000_3F01	设置 TIMCMP [15:8] = 节拍数 $x2-1=32x2-1=63$ (对于 32 拍而言) 设置 TIMCMP [7:0] = 波特率分频器 $/2-1=4/2-1=1$ 。
TIMCFG0	0x0000_2200	在配置定时器输出逻辑为启用时且不受复位影响、FlexIO 时钟递减、移位时钟等于定时器输出、永不复位、在定时器比较时禁用、在触发高电平时使能、停止位禁用、启动位禁用的情况下，配置定时器输出逻辑为 1。
TIMCTL0	0x1DC3_1081	将 SHIFT7 状态标志配置为定时器触发，并且触发极性为低电平有效。配置定时器的引脚为输出，引脚索引为 16 (WR)，引脚极性为低电平有效。将定时器模式配置为 8 位计数器波特/位。

表 4. 多拍读取

寄存器	值	说明
SHIFTCFG0...6	0x0007_0100	从下一个移位器的输出配置移位器输入，禁用停止位，禁用开始位，并在使能的 8 位并行宽度上加载数据。
SHIFTCFG7	0x0007_0000	从引脚配置移位器输入，禁用停止位，禁用开始位，并在使能的 8 位并行宽度上加载数据。
SHIFTCTL0...7	0x0080_0001	从 TIMER0 配置移位时钟，在移位时钟的下降沿触发移位。将移位器的引脚配置为输入，引脚索引从 0 开始，并且引脚极性为高电平有效。将移位器模式配置为接收。
TIMCMP0	0x0000_3F01	设置 TIMCMP [15:8] = 节拍数 $x2-1=32x2-1=63$ (对 32 位而言) 设置 TIMCMP [7:0] = 波特率分频器 $/2-1=4/2-1=1$ 。
TIMCFG0	0x0000_2220	在定时器启用且不受复位影响、FlexIO 时钟递减、移位时钟等于定时器输出、永不复位、在定时器比较时禁用、在触发高电平时启用、在定时器禁用时启用停止位的情况下，配置定时器输出逻辑为 1 (用于禁用内部信号同步计时器)，并且禁用启动位。
TIMCTL0	0x01C3_1181	将 SHIFT0 状态标志配置为定时器触发，并且触发极性为低电平有效。配置定时器的引脚为输出，引脚索引为 17 (RD)，引脚极性为低电平有效。将定时器模式配置为 8 位计数器波特/位。

### 4.3.2 硬件连接

表 5 至 表 7 提供了 FRDM-K32L2A 板与 LCD 模块之间的硬件连接。使用外部电线建立这些连接。

表 5 列出了 FlexIO 引脚与 LCD 模块的连接。

#### 注意

FXIO\_D8 至 FXIO\_D15 引脚专用于 16 位总线接口。

表 5. FlexIO 引脚分配和连接

FlexIO 引脚	端口引脚	板连接器	LCD 信号
FXIO_D0	PTD0	J2-6	D0
FXIO_D1	PTD1	J2-12	D1
FXIO_D2	PTD2	J2-8	D2
FXIO_D3	PTD3	J2-10	D3
FXIO_D4	PTD4	R44-1	D4
FXIO_D5	PTD5	J8-1	D5
FXIO_D6	PTD6	J2-2	D6
FXIO_D7	PTD7	J2-4	D7

下页继续...

表 5. FlexIO 引脚分配和连接 (续上页)

FlexIO 引脚	端口引脚	板连接器	LCD 信号
FXIO_D8	PTB0	J4-2	D8
FXIO_D9	PTB1	J4-4	D9
FXIO_D10	PTB2	J1-6	D10
FXIO_D11	PTB3	J4-8	D11
FXIO_D12	PTB8	J3-15	D12
FXIO_D13	PTB9	J1-8	D13
FXIO_D14	PTB10	J1-10	D14
FXIO_D15	PTB11	J1-12	D15
FXIO_D16	PTB16	J1-2	WR
FXIO_D17	PTB17	J1-4	RD

表 6 列出了 GPIO 引脚到 LCD 模块的连接。

表 6. GPIO 引脚分配和连接

GPIO 引脚	端口引脚	板连接器	LCD 信号
GPIOB18	PTB18	J1-1	CS
GPIOB19	PTB19	J1-3	RS

表 7 列出了 LCD 模块的电源连接。

表 7. 电源连接

电源	板连接器	LCD 信号
VCC	J3-4	3V3
GND	J3-12	GND

## 4.4 运行演示

下载并解压缩软件包。生成、下载并运行该演示。您可以在 LCD 屏幕上看到 NXP 标志的图片，如 图 10 所示。标志图像在屏幕周围移动，到达边缘时会回显。

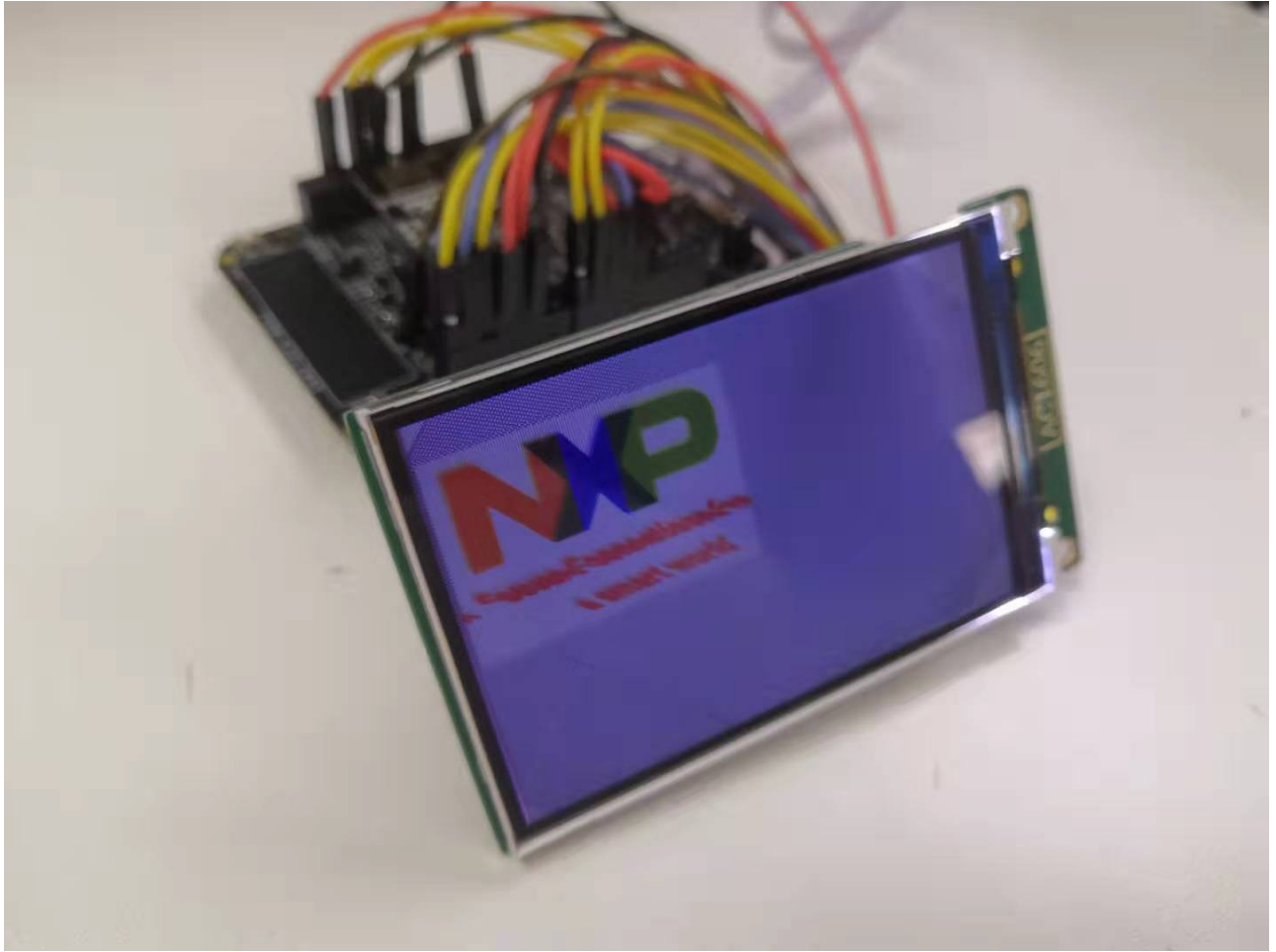


图 10. 运行演示

## 5 结论

该应用程序通过使用 FlexIO 实现了一个模拟 8080 总线的演示。实现了几种驱动器功能，用于写入和读取驱动 TFT LCD 模块。这些测量表明，刷新 320x480x2 尺寸的 LCD 模块所需的 39 fps 的性能令人满意。

**How To Reach Us**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://nxp.com/support)

**Limited warranty and liability** — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

**Right to make changes** - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2020-2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 2020 年 1 月  
Document identifier: AN12685

