

## 1 引言

我们可以发现定制的分段液晶显示器 (SLCD) 技术无处不在。例如：

- 在测量游泳池 PH 值的产品中。
- 用于测量矿井中特定气体的监测器。
- 在温度计里，用来检查孩子是否发烧。

SLCD 是最古老的显示技术之一。由于价格和功耗最低，它仍然是最受欢迎的技术之一。

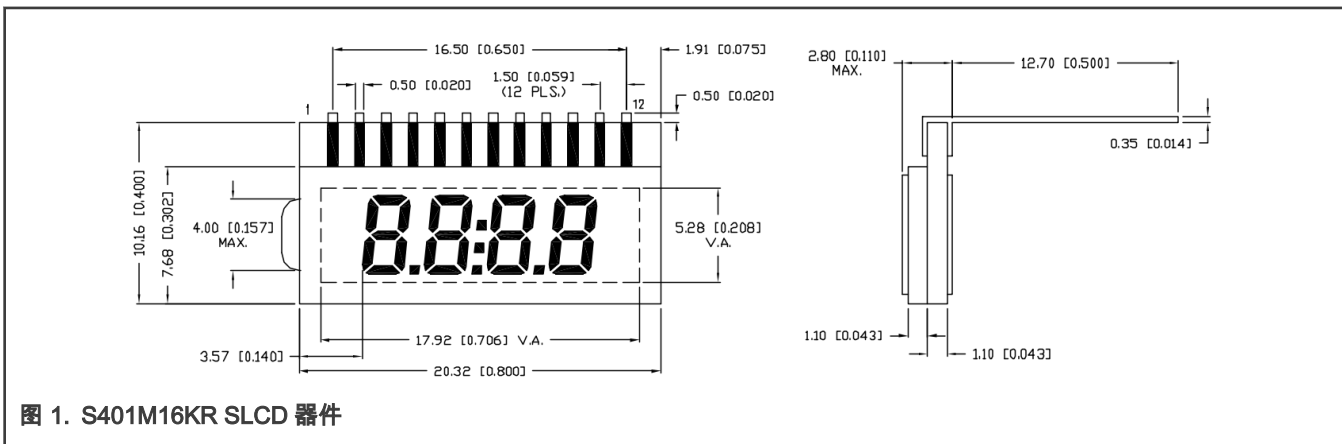
段液晶显示器，也称为静态显示器或纯玻璃显示器，由两块氧化铟锡(ITO)玻璃组成，中间夹有扭曲的向列流体。静态显示是一个段显示器，每个段有一个引脚。段是可以独立打开和关闭的任何线、点或符号。

NXP K32L2B3MCU 集成了一个 SLCD 控制器模块，其最多有八个背板和 47 个前板，如 8×47 个或 4×51。本文档描述了通过启用一个 SLCD 器件 S401M16KR 来使用片上 SLCD 控制器，该设备是一个四位 0.17 英寸七段定制液晶面板。

## 2 硬件

### 2.1 S401M16KR SLCD 器件

S401M16KRSLCD 器件包含显示在面板上的四位数字。每个数字用七段和一个点或冒号显示，如 图 1 所示。



S401M16KR SLCD 器件包含四个 COM 引脚和八个 DATA 引脚作为控制信号。COM 引脚和 DATA 引脚控制一个矩阵，该矩阵指示特定时间段内哪些段为 ON，其他段为 OFF，如 图 2 所示。

### 目录

1	引言.....	1
2	硬件.....	1
2.1	S401M16KR SLCD 器件.....	1
2.2	FRDM-K32L2B3 板.....	3
3	基本用法.....	5
4	低功耗模式下的使用.....	8
5	结论.....	9
6	参考资料.....	10

PIN	1	2	3	4	5	6	7	8	9	10	11	12
COM0	COM0	/	/	/	1D	DP1	2D	DP2	3D	DP3	4D	COL
COM1	/	COM1	/	/	1E	1C	2E	2C	3E	3C	4E	4C
COM2	/	/	COM2	/	1G	1B	2G	2B	3G	3B	4G	4B
COM3	/	/	/	COM3	1F	1A	2F	2A	3F	3A	4F	4A

图 2. S401M16KR SLCD 器件

每个步骤逐个启用 COM 引脚。在每个步骤中，由自己的 COM 引脚激活，八个数据引脚输出控制电平信号来打开和关闭段。每个 COM 的段是逐行打开和关闭的。一旦具有四个步骤的循环快速运行，就可以在整个显示视图中一起看到一些段（即使它们不在矩阵的同一行中）。

将控制信号视为激活矩阵，见 表 1。

表 1. 激活 Matrix 来控制信号

nCS	D0	D1	D2	D3	D4	D5	D6	D7
COM0	1D	1DP	2D	2DP	3D	3DP	4D	4DP
COM1	1E	1C	2E	2C	3E	3C	4E	4C
COM2	1G	1B	2G	2B	3G	3B	4G	4B
COM3	1F	1A	2F	2A	3F	3A	4F	4A

对于每个数字位置，不同的数字由不同的段组装而成。图 3 以直接段的方式显示 0-9 个数字。

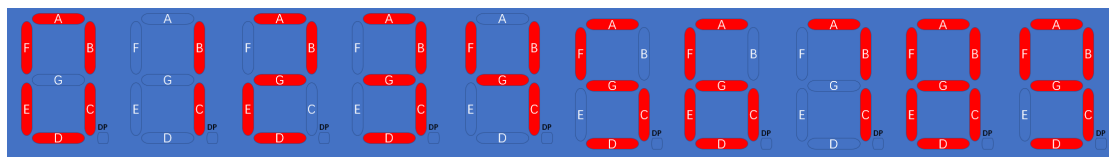


图 3. SLCD 位

表 2 描述了相关信息。

表 2. 液晶数字信息

Number	Segment	COM0 (.D)	COM1 (CE)	COM2 (BG)	COM3 (AF)
0	ABCDEF	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
1	BC	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
2	ABDEG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
3	ABCDG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
4	BCFG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
5	ACDFG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
6	ACDEFG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
7	ABC	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)

下页继续...

表 2. 液晶数字信息 (续上页)

Number	Segment	COM0 (.D)	COM1 (CE)	COM2 (BG)	COM3 (AF)
8	ABCDEFG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
9	ABCDFG	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
None	—	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)
Dot	DP	*D (0 × 1)	CE (0 × 3)	B* (0 × 2)	AF (0 × 3)

如表 2 所示，我们可以得到代码，在每个由指示 COMx 引脚激活的周期显示不同的数字。源

文件中有以下代码数组：

```
#define SLCD_ON_SHOW_COUNT 11u

const uint8_t SLCD_NUMBER_TABLE[][SLCD_COMx_COUNT] =
{
    /* COM0, COM1, COM2, COM3 */
    { 0x1, 0x3, 0x2, 0x3 }, /* SLCD_ON_SHOW_NUMBER_0 */
    { 0x0, 0x2, 0x2, 0x0 }, /* SLCD_ON_SHOW_NUMBER_1 */
    { 0x1, 0x1, 0x3, 0x2 }, /* SLCD_ON_SHOW_NUMBER_2 */
    { 0x1, 0x2, 0x3, 0x2 }, /* SLCD_ON_SHOW_NUMBER_3 */
    { 0x0, 0x2, 0x3, 0x1 }, /* SLCD_ON_SHOW_NUMBER_4 */
    { 0x1, 0x2, 0x1, 0x3 }, /* SLCD_ON_SHOW_NUMBER_5 */
    { 0x1, 0x3, 0x1, 0x3 }, /* SLCD_ON_SHOW_NUMBER_6 */
    { 0x0, 0x2, 0x2, 0x2 }, /* SLCD_ON_SHOW_NUMBER_7 */
    { 0x1, 0x3, 0x3, 0x3 }, /* SLCD_ON_SHOW_NUMBER_8 */
    { 0x1, 0x2, 0x3, 0x3 }, /* SLCD_ON_SHOW_NUMBER_9 */
    { 0x0, 0x0, 0x0, 0x0 }, /* SLCD_ON_SHOW_NONE */
    { 0x2, 0x0, 0x0, 0x0 }, /* SLCD_ON_SHOW_DP */
};
```

#### 注意

数组的每个数字仅包含两个引脚。一个四位数的并行可以用八个引脚重复扩展。下面一节介绍四位数的用法。

## 2.2 FRDM-K32L2B3 板

在 FRDM-K32L2B3 板上，SLCD 器件用引脚连接到 MCU。原理图如图 4 所示。

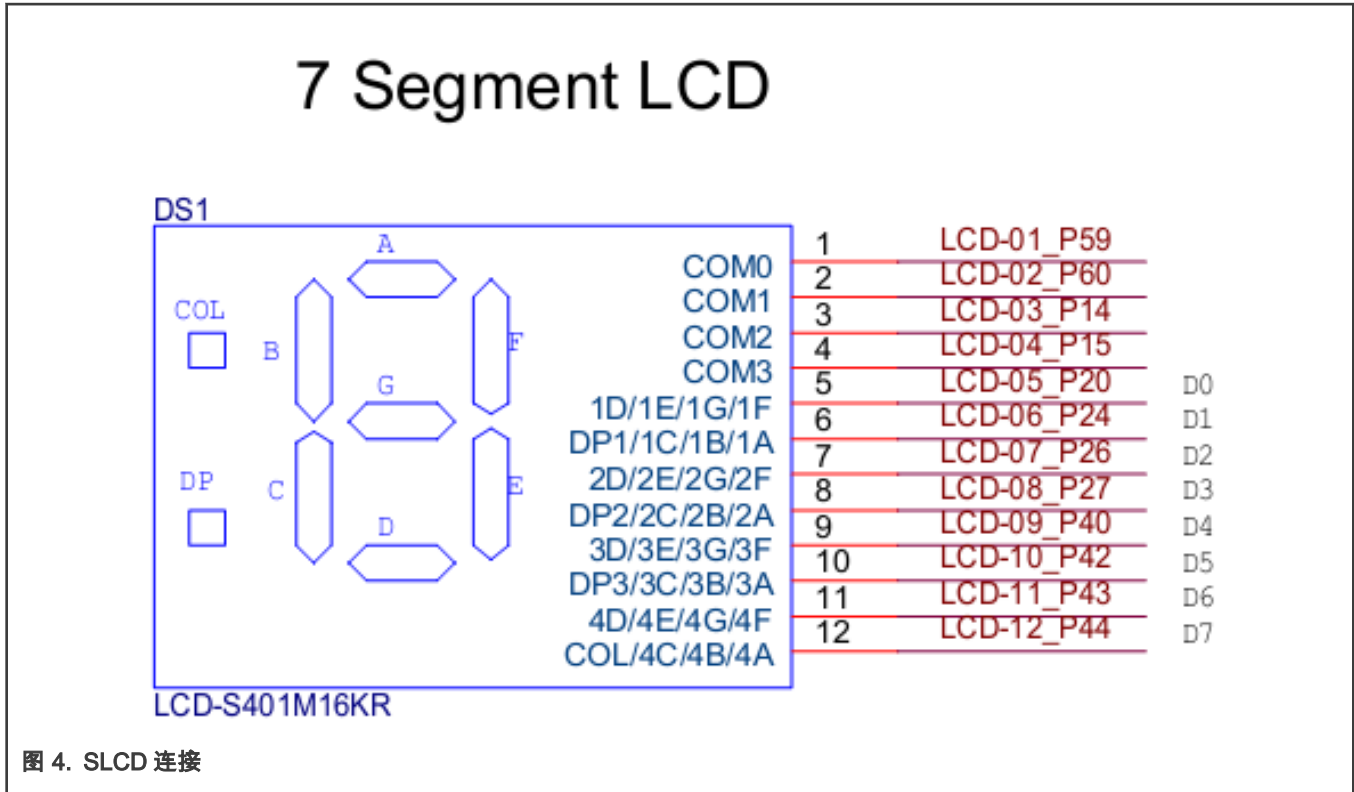


表 3 显示有关引脚设置的功能信息。

表 3. 关于引脚设置的功能信息

Functional ID	SLCD pin	MCU pin	ALT	Comments
LCD-01	P59	PTE20	ALT0	COM0
LCD-02	P60	PTE21	ALT0	COM1
LCD-03	P14	PTB18	ALT0	COM2
LCD-04	P15	PTB19	ALT0	COM3
LCD-05	P20	PTC0	ALT0	D0
LCD-06	P24	PTC4	ALT0	D1
LCD-07	P26	PTC5	ALT0	D2
LCD-08	P27	PTC6	ALT0	D3
LCD-09	P40	PTD0	ALT0	D4
LCD-10	P42	PTD2	ALT0	D5
LCD-11	P43	PTD3	ALT0	D6
LCD-12	P44	PTD4	ALT0	D7

作为总线，数据信号由八个分离的引脚组装，从 D0 到 D7。编码是为了控制这些信号而执行的，来自每个引脚的信号通过总线被看作是一个整体的数据。

为了像总线一样操作所有的控制信号，在源代码中将信号索引排列成两个数组：

```

/* Define the sync bus and the data bus. */
#define SLCD_COMx_COUNT 4u
#define SLCD_DATA_BUS_WIDTH 8u
/* Define the pins for sync bus and data bus. */
const uint8_t SLCD_PIN_COMx[SLCD_COMx_COUNT] =
{
    59, /* COM0. */
    60, /* COM1. */
    14, /* COM2. */
    15 /* COM3. */
};
const uint8_t SLCD_PIN_DATA[SLCD_DATA_BUS_WIDTH] =
{
    20, /* D0. */
    24, /* D1. */
    26, /* D2. */
    27, /* D3. */
    40, /* D4. */
    42, /* D5. */
    43, /* D6. */
    44 /* D7. */
};

```

### 3 基本用法

K32L2B 上的 SLCD 控制器使用方便。在启用时钟和设置引脚复用功能后，对于没有闪烁和故障检测功能的基本用法，只需要一个控制寄存器 LCD 总控制寄存器 ( LCD\_GCR ) 就可以初始化控制器。下面提供一组典型设置：

```

/* Setup slcd controller. */
LCD->GCR = LCD_GCR_DUTY(3) /* Selects the duty cycle of the LCD controller driver. 3: 4 COMx lines.
*/
| LCD_GCR_LCLK(2) /* Clock divider for clock source. 0-7 */
| LCD_GCR_SOURCE(0) /* LCD clock source. 1:use MCGIRCLK. 0:OSC32K */
| LCD_GCR_LCDEN(0) /* Disable the controller during setting. */
| LCD_GCR_LCDSTP(0) /* Keep LCD module alive in STOP modes. */
| LCD_GCR_LCDDOZE(1) /* Keep LCD module alive in DOZE mode. */
| LCD_GCR_FFR(0) /* Select the frame rate mode. 0:standard frame rate. */
| LCD_GCR_ALTSOURCE(0) /* Select the alternate clock source. no available when using default
clock source.*/
| LCD_GCR_ALTDIV(0) /* Clock divider for alternate clock source. no available when using
default clock.*/
| LCD_GCR_FDCIEN(0) /* Enables an LCD interrupt event when fault detection is completed. */
| LCD_GCR_PADSAFE(0) /* Force safe state on LCD pad control, locking all LCD control bits. */
| LCD_GCR_VSUPPLY(0) /* Select the power voltage supply. 0: from internal Vdd. */
| LCD_GCR_LADJ(1) /* Configures SLCD to handle different LCD glass capacitance.*/
| LCD_GCR_CPSEL(1) /* Selects the LCD controller charge pump or a resistor network to
supply the LCD voltages V_LLx. */
| LCD_GCR_RVTRIM(8) /* Regulated Voltage Trim. no available when disabled.*/
| LCD_GCR_RVEN(0) /* Regulated Voltage Enable. disabled. */
;

```

要求将 MCU 上的引脚映射到用于 COMX 信号和 Dx 信号的 SLCD 控制总线上。

- COMX 信号的映射被配置为背板引脚。
- 将 DX 信号的映射配置为前面板引脚。

- 使用的引脚通过 LCD 引脚启用寄存器 ( LCD\_PEN0 , LCD\_PEN1 )、LCD 背板启用寄存器 ( LCD\_BPEN0 , LCD\_BPEN1 ) 初始化，LCD\_PENx 启用所有正在使用的引脚，而 LCD\_BPENx 选择它们为前面板或后面板。
- LCD\_WF8Bx 寄存器用于每个引脚的信号时序。

下面介绍 LCD\_WF8Bx 寄存器的用法：

- LCD\_WF8Bx 阵列中的每个寄存器为一个 LCD 信号引脚。数组的索引也用于 SLCD 模块的功能引脚。例如，LCD\_WF8B[59]是针对 SLCD 模块的信号引脚，LCD\_P59。
- LCD\_WF8Bx 寄存器中的每个位都是步骤 1，而位的索引也是步骤 1。例如，在整个周期的步骤 2 中，LCD\_WF8B [59]的位 2 相应于 LCD\_P59 ( 包括四步或八步 )。

在软件中，控制数据信号与硬件有点不同。软件先对数据进行并行引脚的搜索，然后整理数据时序。然而，硬件首先搜索每个引脚的数据时序，然后将并行引脚组装成 8 位宽度的总线。因此，在源代码项目中设计了一个转换函数。

```
/**
 * @brief Set the data on SLCD control bus
 * @param com_idx The index of step (COMx), 0-3.
 * @param show_dat The display code to the bus for current step.
 */
void slcd_set_bus_data(uint8_t com_idx, uint8_t show_dat)
{
    uint8_t bit_mask = (1u << com_idx);
    for (uint8_t i = 0u; i < SLCD_DATA_BUS_WIDTH; i++)
    {
        if (show_dat & 0x1)
        {
            LCD->WF8B[SLCD_PIN_DATA[i]] |= bit_mask;
        }
        else
        {
            LCD->WF8B[SLCD_PIN_DATA[i]] = ~bit_mask;
        }
        show_dat >>= 1u;
    }
}
```

创建了一个 API 函数，以将段代码组合到四位数字的显示矩阵中。使用此 API，不需要复杂的矩阵转换。您只需告诉 MCU 您想要显示的数字以及您希望将其显示在哪个位置，因为该软件会自动处理所有转换。

```
/* keep the unchanged displaying code in the matrix. */
static uint8_t slcd_on_show_numbers[SLCD_COMx_COUNT];

/**
 * @brief Set the displaying number in the digital position of SLCD device.
 * @param index The index of digital position, 0-3.
 * @param number The value of showing number, 0-10, while 10 is "none".
 * @param en_dp Enable showing the dop in current digital positon, true or false.
 */
void slcd_set_number(uint8_t index, uint8_t number, bool en_dp)
{
    uint8_t tmp8 = 0u;
    for (uint8_t i = 0u; i < SLCD_COMx_COUNT; i++)
    {
        tmp8 = slcd_on_show_numbers[i] & ~(0x3 << (2 * index)); /* clear old setting code.*/
        tmp8 |= (SLCD_NUMBER_TABLE[number][i] << (2 * index)); /* add new setting code. */
        if (en_dp)
        {
            tmp8 |= SLCD_NUMBER_TABLE[SLCD_ON_SHOW_NUMBER_DP][i] << (2 * index); /* add new setting for
```

```
dot point. */
    }
    slcd_on_show_numbers[i] = tmp8;
    slcd_set_bus_data(i, slcd_on_show_numbers[i]);
}
}
```

在应用程序 main()函数中，用于在目标 SLCD 上显示变化的数字的源代码如下：

```
int main(void)
{
    bool en_dp;

    /* init board hardware. */
    BOARD_InitPins();
    BOARD_BootClockRUN();
    BOARD_InitDebugConsole();

    PRINTF("slcd basic example.\r\n");
    /* init the clock and pins for slcd, setup the controller for slcd. */
    slcd_init();

    en_dp = false;
    while (1)
    {
        for (uint8_t i = 0u; i < SLCD_ON_SHOW_COUNT; i++)
        {
            GETCHAR();

            slcd_stop(); /* stop the slcd controller before updating displaying. */
            slcd_set_number(0, i, en_dp);
            slcd_set_number(1, (i+1)%SLCD_ON_SHOW_COUNT, en_dp);
            slcd_set_number(2, (i+2)%SLCD_ON_SHOW_COUNT, en_dp);
            slcd_set_number(3, (i+3)%SLCD_ON_SHOW_COUNT, en_dp);
            slcd_start();
        }
        en_dp = !en_dp;
    }
}
```

下载该工程，并在 FRDM-K32L2B 板上运行。数字显示在 SLCD 上，如 [图 5](#) 所示。

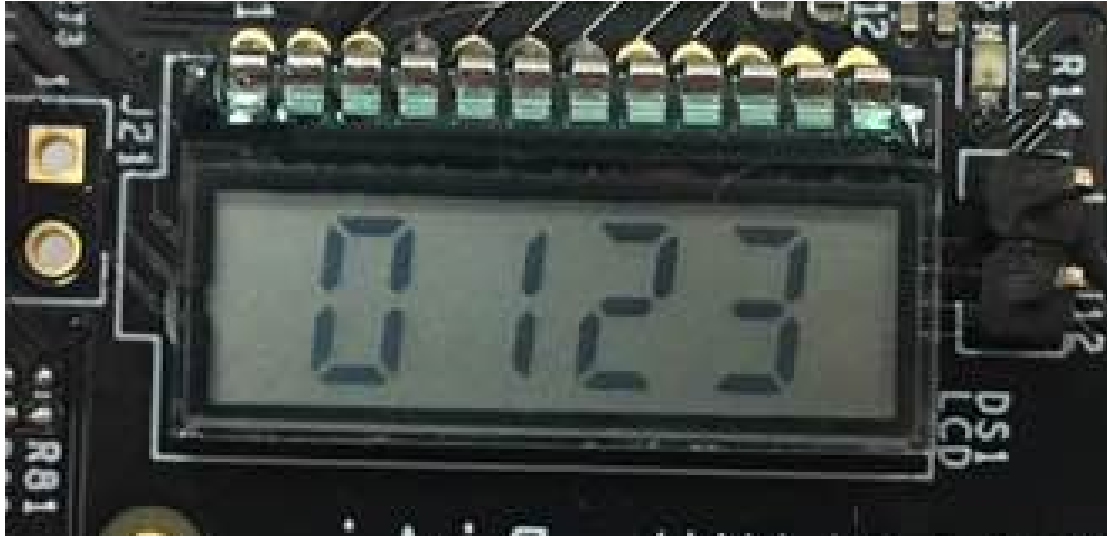


图 5. SLCD

对于整个可运行的源代码项目，slcd\_basic，请参阅 [AN12579SW](#)。

## 4 低功耗模式下的使用

在带有 SLCD 控制器的 MCU 上，SLCD 控制器特别支持一些额外的低功耗停止模式。在这些模式下，几乎所有的硬件都被关闭到相同的能量。SLCD 控制器控制其引脚以刷新 SLCD 器件，以保持其显示在面板上。

除了 VLLS0 模式外，SLCD 几乎可以在所有的功耗模式下工作。有关详细信息，请参阅用户手册中的电源管理章节。

表 4. 低功耗模式下的 SLCD

Modules	VLPR	VLPW	Stop	VLPS	LLS	VLLSx
Segment LCD	FF Async operation in CPO	FF	Async operation FF in PSTOP2	Async operation	Async operation	Async operation, OFF in VLLS0

保持 SLCD 在低功耗模式下工作的关键设置是：

- **时钟源**：确保 SLCD 控制器的时钟源在目标低功耗模式下仍然有效。例如，如果 SLCD 控制器的时钟源为 32KOSC，则在每种模式下都启用此时钟源，即使在 VLLSx 模式下也是低的。
- **引脚复用**：确保 SLCD 控制器使用的引脚被配置为模拟功能 (ALT0)，其他 (数字) 功能的引脚在 VLLSx 模式下被锁定 (电压电平不能改变)。只有当 SLCD 控制器的引脚处于活动状态时，SLCD 控制器才会输出波形，SLCD 面板继续显示数字。
- **SLCD 控制器的低功耗模式**：通过设置 LCD\_GCR [LCDST P]位和 LCD\_GCR[LCD DOZE]位为 0 来启用 SLCD 控制器的低功耗支持，以保持 SLCD 控制器仍在停止和等待模式下工作。

在完成上述参数后，设置用于显示的 SLCD 控制器，然后进入低功耗停止模式。在低功耗停止模式下，在 SLCD 面板上的显示是打开的，因为 SLCD 控制器仍然输出刷新波。

图 6 所示的示例项目中，在 VLLS3 模式下，RTC 和 SLCD 控制器运行，FRDM-K32L2B 板上的测量电流低至 7 uA 左右。



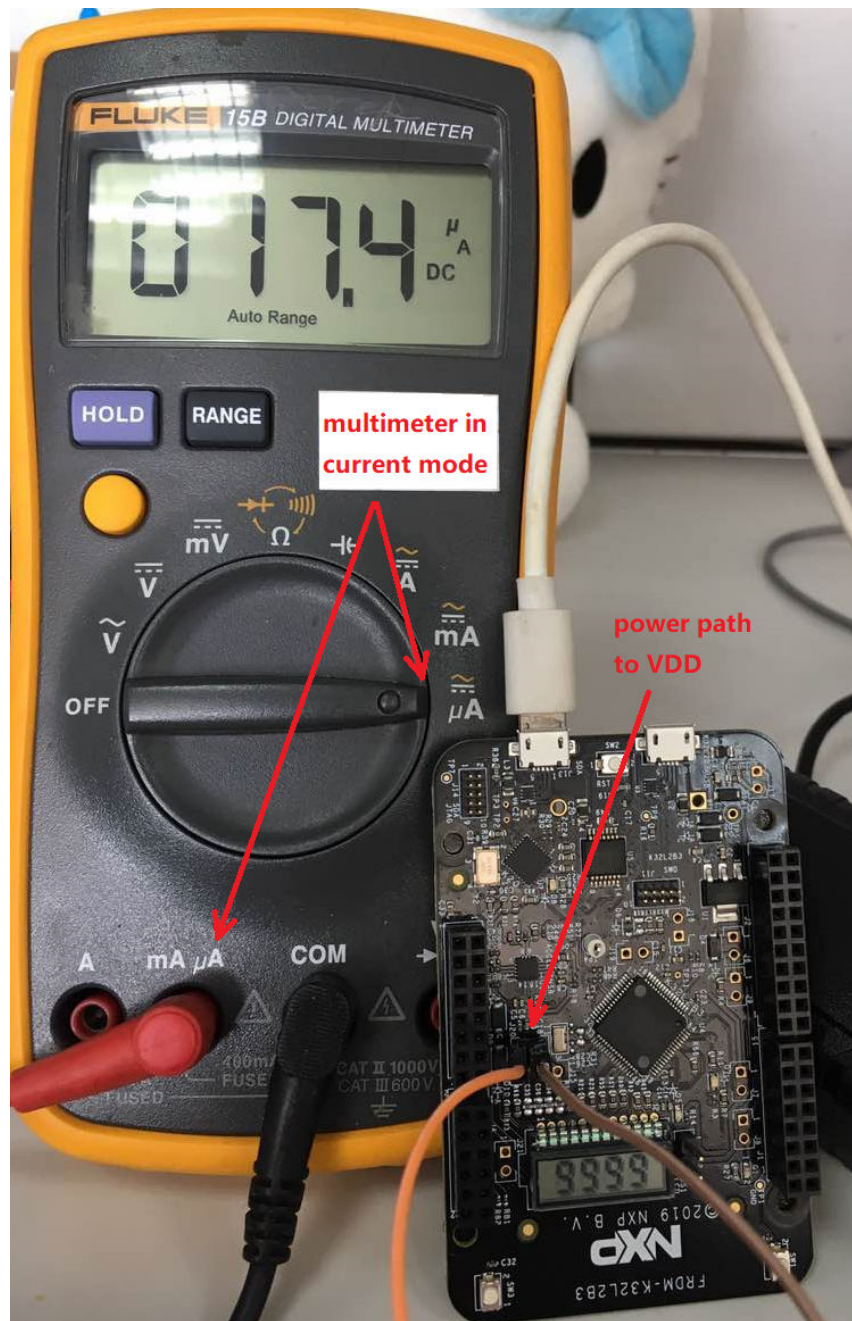


图 6. SLCD 低功耗显示

对于整个可运行的源代码项目，`slcd_low_power`，请参阅 [AN12579SW](#)。

## 5 结论

本文介绍了 K32L2BMCU 片上 SLCD 控制器的基本使用方法，并给出了基于 FRDM-K32L2B 板的实例。SLCD 控制器可以通过适当的硬件配置控制 SLCD 设备在其面板上自动显示内容。即使在低功耗模式下，SLCD 控制器仍然可以用很低的能量工作。这表明带有片上 SLCD 控制器的 K32L2B 可用于能量敏感的应用领域。

## 6 参考资料

- [FOCUS LCDs/](#)

**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 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. 2019-2021.

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: 2019 年 12 月

Document identifier: AN12579

