

FG23 评估报告

前言：

FG23-DK2600A EFR32FG2 这块开发板功能还是比较丰富的，里面有 1 个 BRD2600A xG23 868-915 MHz +14 开发板，1 根 868 MHz 天线，1 根 915 MHz 天线，2 个 5 号电池座，USB 连接线。

目录

前言:	0
基本组成:	3
目标设备	3
设备功能	3
虚拟串口	4
Simplicity Studio 5	4
程序示例:	5

基本组成:

EFR32FG23 开发套件是用于 EFR32FG23 无线片上系统的低成本、小尺寸原型设计和开发平台。该板体积小、成本效益高且功能丰富。它是开发基于 EFR32FG23 的节能物联网设备的理想平台。内置的 SEGGER J-Link 调试器确保通过 USB Micro-B 连接器轻松调试。



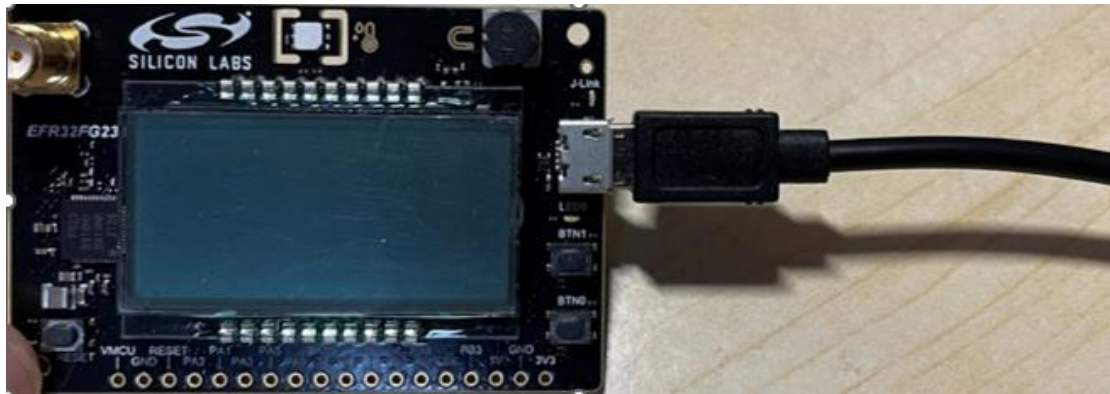
目标设备

EFR32FG23 无线片上系统

- EFR32FG23 无线片上系统 (EFR32FG23B010F512IM48) • 32 位 ARM@Cortex@-M33, 最大工作频率 78 MHz • 512kb 闪存和 64kb RAM • 低活性和睡眠电流的节能无线电核心 • 集成 PA, TX 功率高达 20 dBm (subGHz) • 强大的外设和 31 GPIO。

设备功能

SMA 连接器用于天线连接 5 位 7 段液晶显示器相对湿度和温度传感器 LC 传感器用于金属检测用户 LED 和 2 个按钮 2.54 mm 断开板用于 GPIO 访问 SEGGER J-Link 机载调试器虚拟 COM 端口报文跟踪接口(PTI)使用外部 Silicon Labs 调试器的 AEM 和包跟踪的迷你简单连接器 • USB 或硬币电池供电。



虚拟串口

虚拟 COM 端口(VCOM)连接到 EFR32FG23 上的 UART，并允许从设备发送和接收串行数据。板载调试器将此显示为主机计算机上的虚拟 COM 端口，插入 USB 电缆时会显示该端

口。数据通过 USB 连接在主机和调试器之间传输，该连接使用 USB 通信设备类(CDC)模拟串行端口。数据从调试器通过物理 UART 连接传递到目标设备。串行格式默认为 115200bps、8 位、无奇偶校验和 1 个停止位。

注意：更改 PC 端 COM 端口的波特率不会影响调试器和目标设备之间的 UART 波特率。但是，可以通过 SimplicityStudio 提供的套件管理控制台更改 VCOM 波特率。或者，VCOM 端口也可以通过 MiniSimplicity 连接器与外部无线 STK 一起使用。通过 MiniSimplicity 连接器与外部无线 STK 使用 VCOM 端口的工作方式类似，但需要拔下连接

板载调试器的 USB 电缆。然后，无线 STK 上的板控制器通过 USB(CDC)或 IP 插座提供数据。

MiniSimplicity 连接器不提供流量控制。

Simplicity Studio 5

Simplicity Studio 5 是 Silicon Labs 推出的针对 8 位、32 位 MCU 和无线 SOC 的开发平台。Simplicity Studio 5 集成丰富的开发资料和工具，可以简化 Silicon Labs 产品资料查询和程序开发，提高开发效率。



程序示例：

厂商给的程序示例还是很多的，可以让使用者快速上手。

.pdm	2022/5/14 20:46	文件夹	
.settings	2022/5/14 20:47	文件夹	
autogen	2022/5/14 20:46	文件夹	
blink_baremetal	2022/5/23 19:25	文件夹	
config	2022/5/14 20:47	文件夹	
gecko_sdk_3.2.3	2022/5/14 20:46	文件夹	
.cproject	2022/5/23 19:11	CPROJECT 文件	118 KB
.project	2022/5/14 20:46	PROJECT 文件	26 KB
app.c	2022/5/14 20:46	sourceinsight.c_f...	2 KB
app.h	2022/5/14 20:46	H 文件	2 KB
FG23.slcp	2022/5/14 20:46	SLCP 文件	1 KB
FG23.slps	2022/5/14 20:46	SLPS 文件	1 KB
FG32.pintool	2022/5/23 19:11	PINTOOL 文件	1 KB
main.c	2022/5/14 20:46	sourceinsight.c_f...	3 KB

```
* software is distributed to you in Source Code format and is governed by the
* sections of the MSLA applicable to Source Code.
*
*****/
#include "sl_component_catalog.h"
#include "sl_system_init.h"
#include "app.h"
#if defined(SL_CATALOG_POWER_MANAGER_PRESENT)
#include "sl_power_manager.h"
#endif
#if defined(SL_CATALOG_KERNEL_PRESENT)
#include "sl_system_kernel.h"
#else // SL_CATALOG_KERNEL_PRESENT
#include "sl_system_process_action.h"
#endif // SL_CATALOG_KERNEL_PRESENT

int main(void)
{
    // Initialize Silicon Labs device, system, service(s) and protocol stack(s).
    // Note that if the kernel is present, processing task(s) will be created by
    // this call.
    sl_system_init();

    // Initialize the application. For example, create periodic timer(s) or
    // task(s) if the kernel is present.
    app_init();
#if defined(SL_CATALOG_KERNEL_PRESENT)
```