SILICON LABS FG23-DK2600A 开发套件试用报告

2022 年 05 月 23 日

1 背景

4 月下旬收到了中电网工作人员的中选通知，中间由于疫情原因 5 月上旬才收到板卡。前后花了 3 个周末认真学习了一下，发现 FG23 还是挺强大的。据官网介绍，EFR32FG23 是适用于智能家居、安防、照明、楼宇自动化和计量的 1 GHz 以下物联网无线连接的理想解决方案。高性能的 1 GHz 以下无线电可进行长距离传输，并且不容易受到其他技术造成的 2.4 GHz 干扰。单芯片、多核解决方案可提供多种行业前沿的功能，包括出色的安全性、低功耗、快速唤醒时间以及集成的功率放大器，可为物联网设备实现下一代安全连接。

2 摘要

本文将通过“点亮一颗 LED”和“用 LCD 显示温度”两个项目来循序渐进地介绍 FG23 开发套件的一些基础功能以及调试软件的使用，旨在体现 FG23 的开箱即用的特点，以及 SILICON 完善、方便的生态和工具链。

3 开箱

收到快递的那一刻我还在想怎么这么小个盒子，会不会很鸡肋，但深入学习之后我发现我错了，麻雀虽小，五脏俱全。正式开箱前贴一个官网的配置列表：

- +14 dBm xG23 开发板，基于 xG23 512kB 闪存 QFN48 SoC
- SMC 连接器
- 数据包追踪接口
- 集成调试和数据包追踪
- 虚拟 COM 端口
- J-Link 板载调试器，支持 SWD
- 外部设备调试
- USB 连接
- 段式液晶
- 用户 LED / 按钮
- Si7021 RHT 传感器

- CR2032 钮扣电池支持

外观

四四方方的一个小盒子，包装简洁精致，值得一提的是盒子上的小动物挺可爱的。

箱体外观图

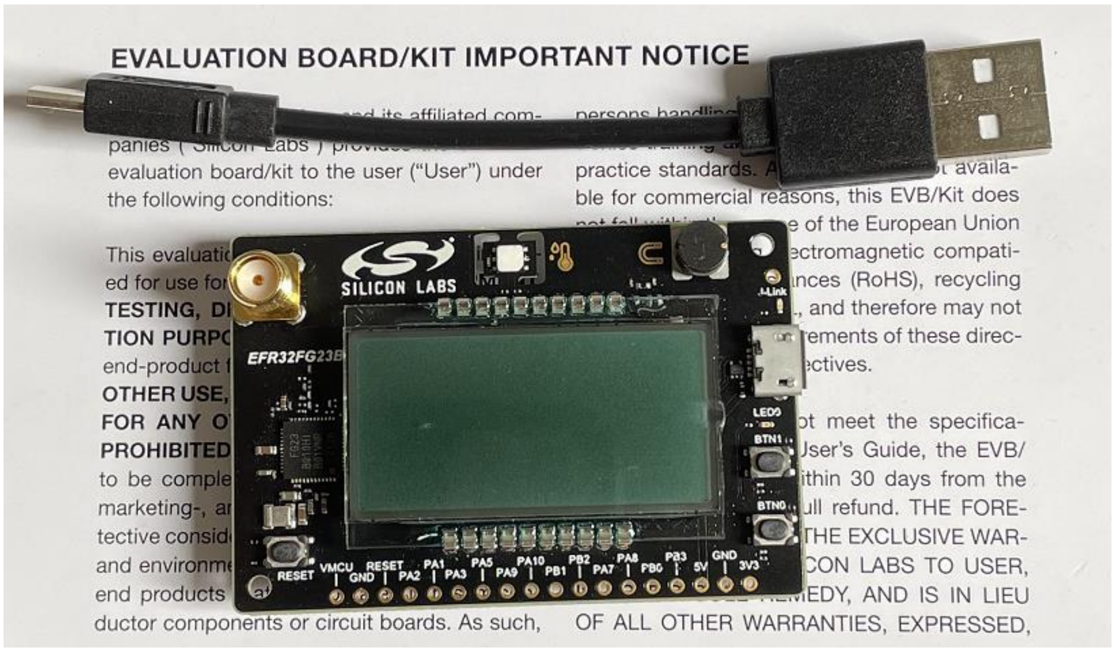
拆开里面带有一张 SILICON LABS 的小卡片，背面印有开发工具下载网址；

卡片图

还有 1 根 868 MHz 天线和 1 根 915 MHz 天线；

天线图

最后就是我们的“主人公”板卡本体了（附带一个调试用的 USB 连接线）；



板卡和调试 USB 线

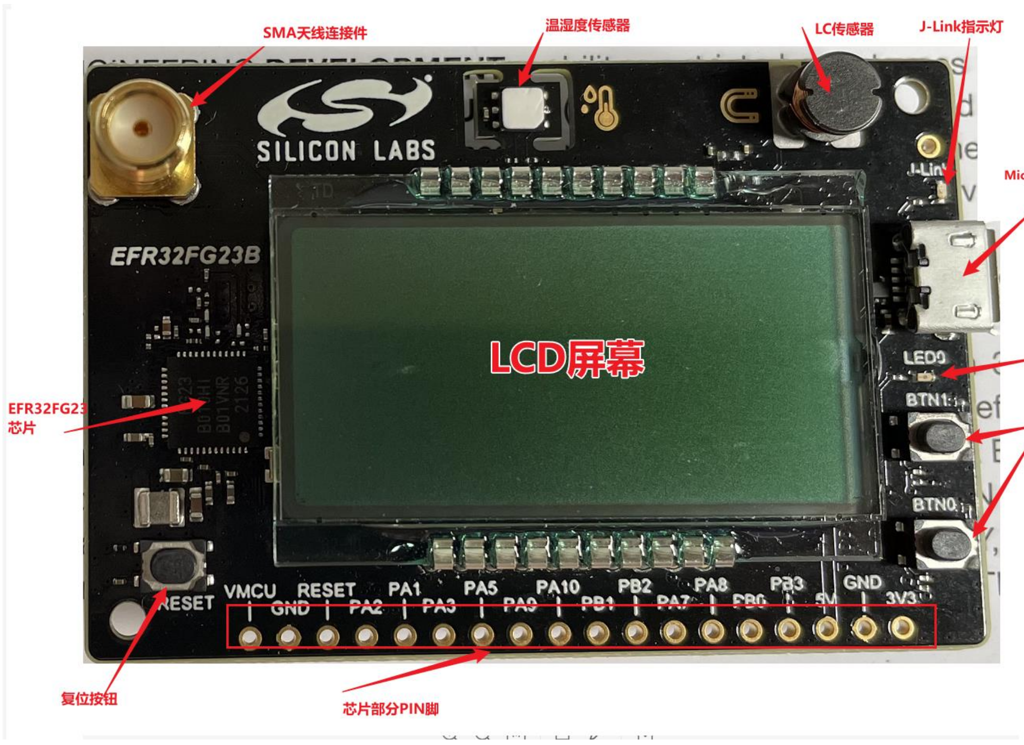
最后来张“全家福”;

915MHz
868MHz
SILICON LABS
W7230F1 EVB/Kit Disclaimer | Rev C
EVALUATION BOARD/KIT IMPORTANT NOTICE
EFR32FG23
SILICON LABS

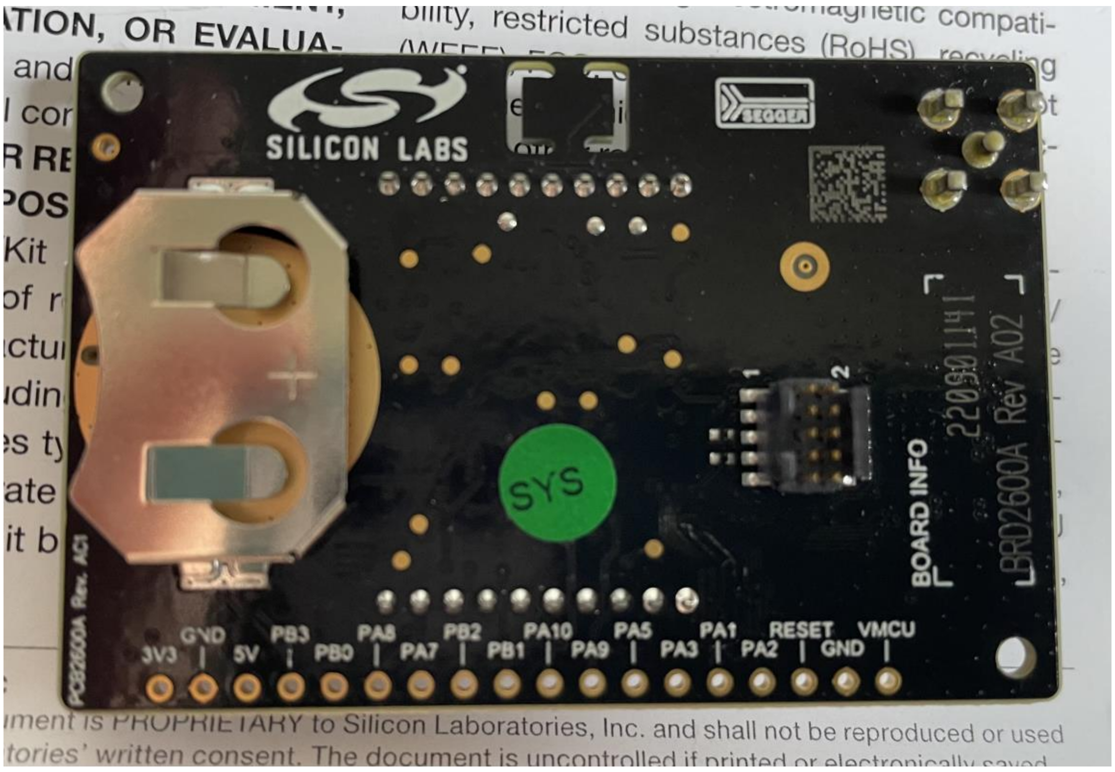
全家福

板卡介绍

板卡正面包含一个 SMA 天线接口、一个温湿度传感器、一个 LC 传感器、一个 J-Link 调试 LED 灯、一个 MicroUSB 接口、一个用户 LED 灯、两个用户按钮、一排外引的 IO 口和供电 PIN 脚、一个外部 Reset 按钮、一块 LCD 屏幕以及 EFR32FG23 芯片；背面比较简单，只有一个 CR2032 钮扣电池座和一个 10PIN 的 SWD 调试接口。


SMA天线连接件
温湿度传感器
LC传感器
J-Link指示灯
Micro USB接口
LCD屏幕
用户LED
用户按钮
EFR32FG23芯片
复位按钮
芯片部分PIN脚
EFR32FG23B
SILICON LABS
VMCU
GND
RESET
PA2
PA1
PA3
PA5
PA9
PA10
PB1
PB2
PA7
PA8
PB0
PB3
5V
GND
3V3
LED0
BTN1
BTN0
RESET

板卡正面


SILICON LABS
SEGGER
SYS
BOARD INFO
BRD2600A Rev A02
PCB2600A Rev. A01
3V3
GND
5V
PB3
PB0
PA8
PA7
PB2
PB1
PA10
PA9
PA5
PA3
PA1
PA2
RESET
GND
VMCU

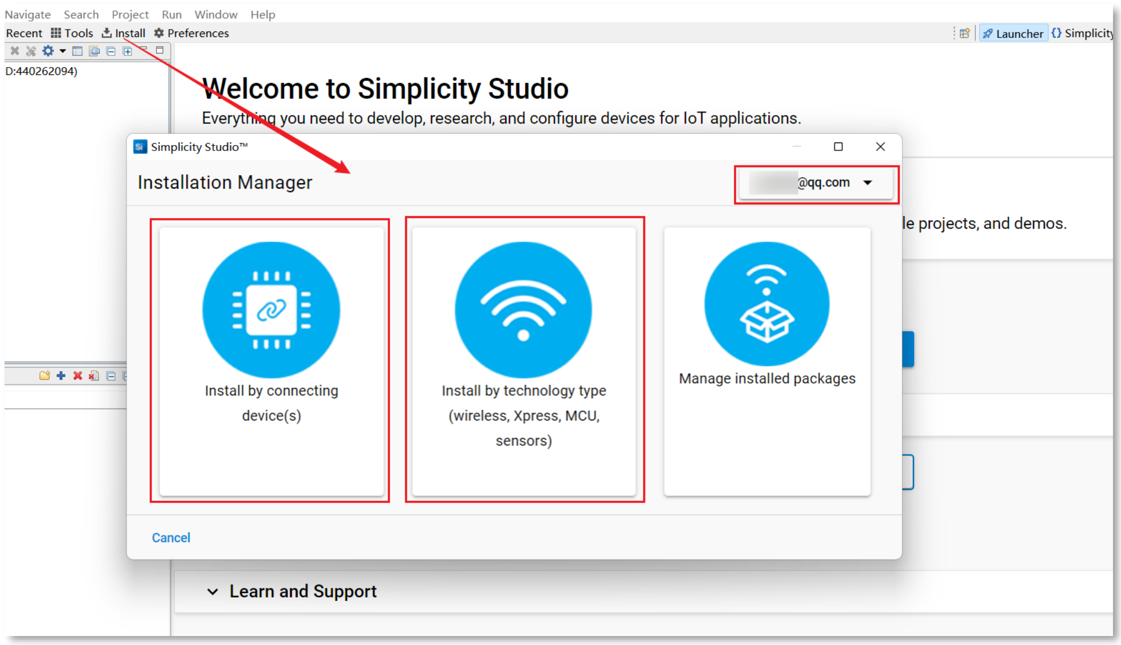
板卡背面

4　试用

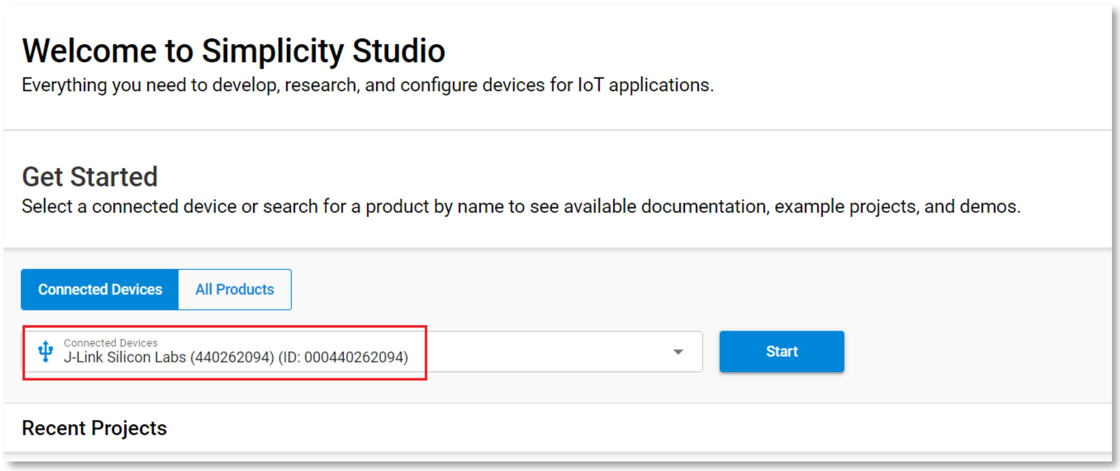
点亮 LED

前期准备

正如我们在学习一门语言是以打印一行“Hello World!”开始的，学习一个开发板通常以点亮一颗 LED 为起点。根据官网的引导，下载安装完软件就可以开始使用了。不过在开始之前我们需要先安装驱动和 SDK，打开软件之后先登录账号，然后按需下载即可。


Navigate Search Project Run Window Help
Recent Tools Install Preferences
Launcher Simplicity
D:440262094)
Welcome to Simplicity Studio
Everything you need to develop, research, and configure devices for IoT applications.
Simplicity Studio™
Installation Manager
@qq.com
le projects, and demos.
Install by connecting device(s)
Install by technology type (wireless, Xpress, MCU, sensors)
Manage installed packages
Cancel
Learn and Support

一切准备就绪后，连接电脑和开发板，此时板卡上的 J-Link LED 点亮，同时电脑软件中会自动显示出设备型号，如无法显示可能是连接驱动未安装成功，可重启电脑或者进入设备管理器查看是否有相应设备。



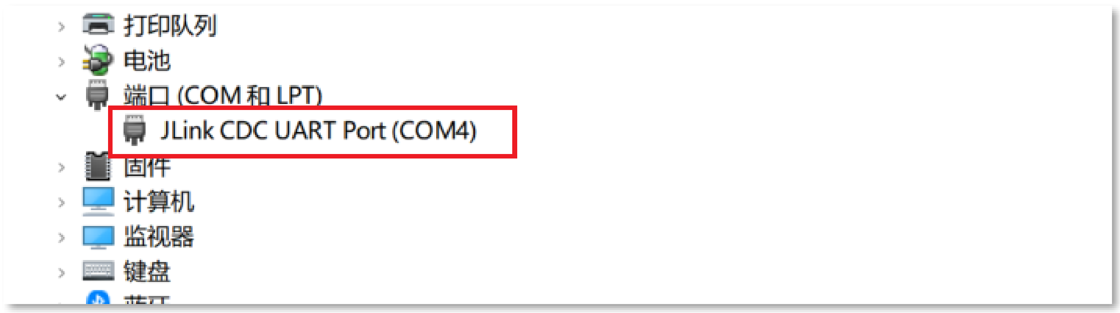
Welcome to Simplicity Studio

Everything you need to develop, research, and configure devices for IoT applications.

Get Started

Select a connected device or search for a product by name to see available documentation, example projects, and demos.

连接成功图

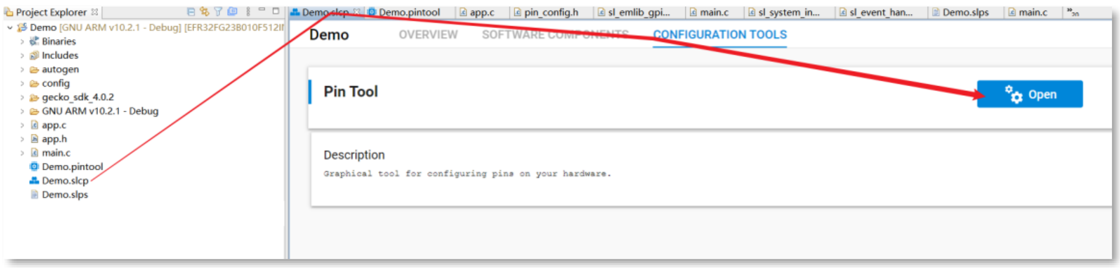


电脑设备管理器界面

连接成功后即可根据指引新建工程，选择相应的 SDK 和 IDE 即可开始开发。

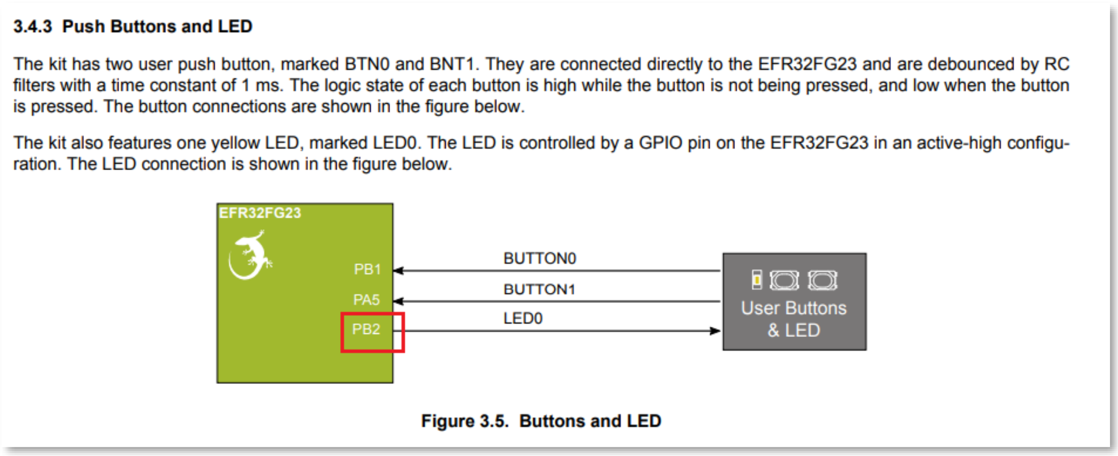
端口配置

双击工程中的 slcp 文件即可打开配置页面，点击配置工具 – Pin Tool；



Graphical tool for configuring pins on your hardware.

通过查看用户手册可知 LED 的控制引脚为 PB2；



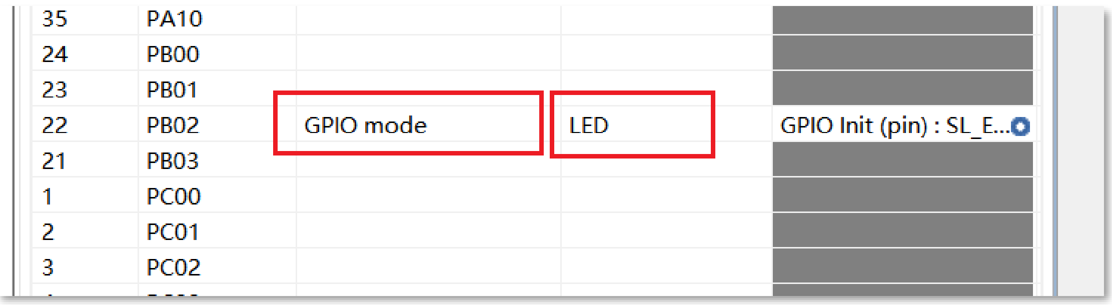
3.4.3 Push Buttons and LED

The kit has two user push button, marked BTN0 and BNT1. They are connected directly to the EFR32FG23 and are debounced by RC filters with a time constant of 1 ms. The logic state of each button is high while the button is not being pressed, and low when the button is pressed. The button connections are shown in the figure below.

The kit also features one yellow LED, marked LED0. The LED is controlled by a GPIO pin on the EFR32FG23 in an active-high configuration. The LED connection is shown in the figure below.

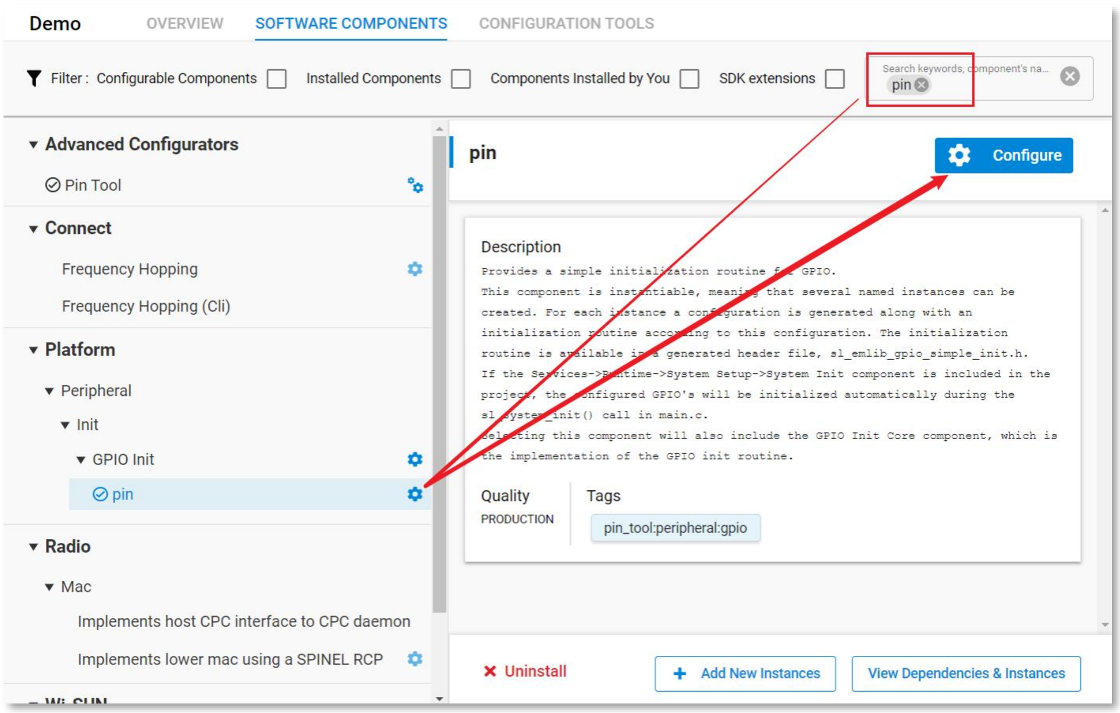
Figure 3.5. Buttons and LED

由此我们可以配置 PB2 为 GPIO 模式，修改 PIN 脚名字为 LED；



35	PA10			
24	PB00			
23	PB01			
22	PB02	GPIO mode	LED	GPIO Init (pin) : SL_E...
21	PB03			
1	PC00			
2	PC01			
3	PC02			

再回到软件组件页面搜索 PIN，配置 PIN 脚的状态；



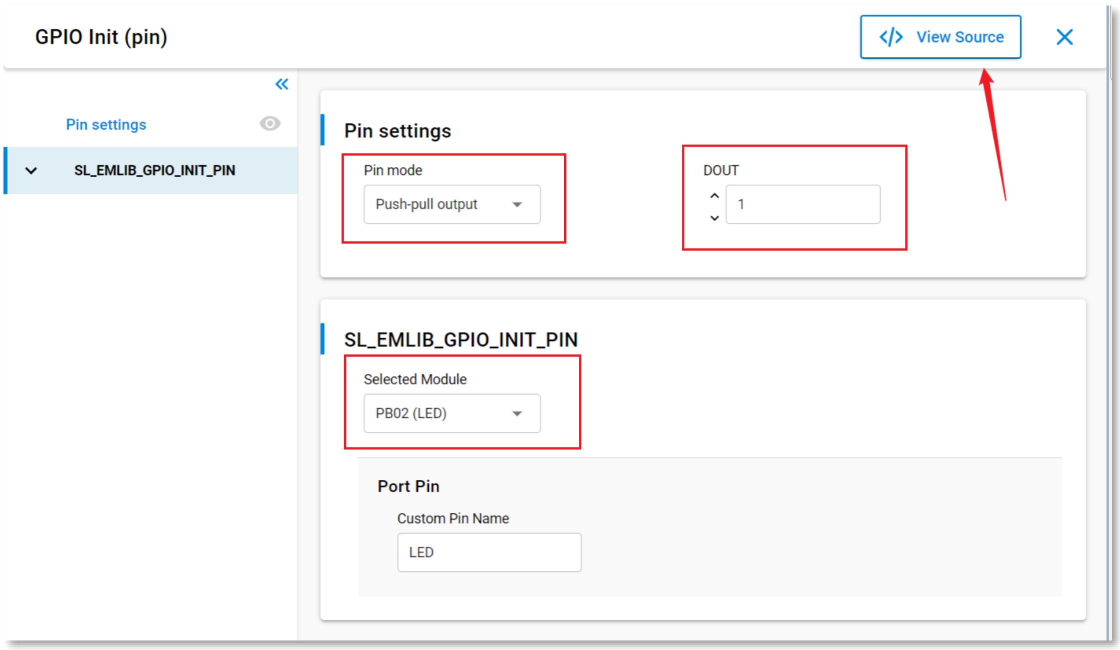
Description

```
Provides a simple initialization routine for GPIO.
This component is instantiable, meaning that several named instances can be
created. For each instance a configuration is generated along with an
initialization routine according to this configuration. The initialization
routine is available in a generated header file, sl_emlib_gpio_simple_init.h.
If the Services->Runtime->System Setup->System Init component is included in the
project, the configured GPIO's will be initialized automatically during the
sl_system_init() call in main.c.
Selecting this component will also include the GPIO Init Core component, which is
the implementation of the GPIO init routine.
```

Quality: PRODUCTION

Tags: pin_tool:peripheral:gpio

然后选择 PB2，模式为 Push-pull 输出，输出状态为“1”；



Pin settings

Pin mode: Push-pull output

DOUT: 1

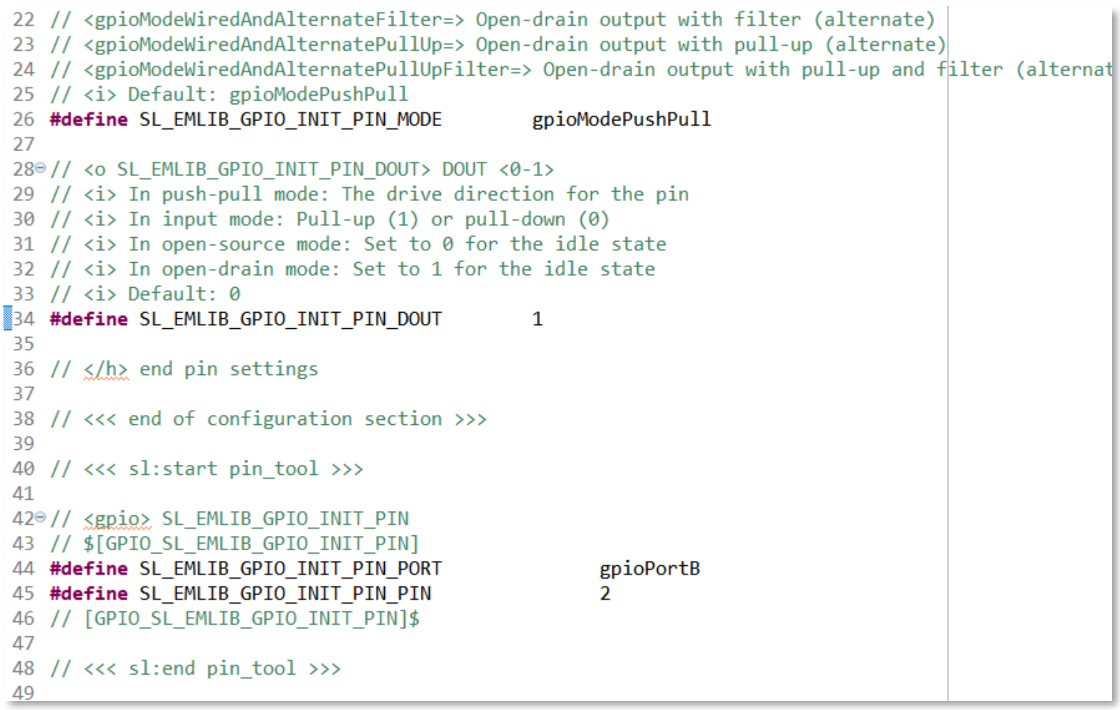
SL_EMLIB_GPIO_INIT_PIN

Selected Module: PB02 (LED)

Port Pin

Custom Pin Name: LED

点击 view source 即可看到生产的代码；

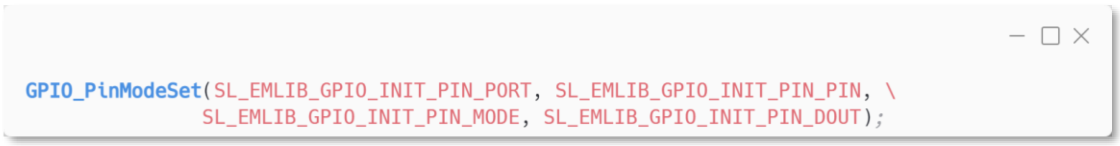


```
// <gpioModeWiredAndAlternateFilter=> Open-drain output with filter (alternate)
// <gpioModeWiredAndAlternatePullUp=> Open-drain output with pull-up (alternate)
// <gpioModeWiredAndAlternatePullUpFilter=> Open-drain output with pull-up and filter (alternat
// <i> Default: gpioModePushPull
#define SL_EMLIB_GPIO_INIT_PIN_MODE        gpioModePushPull

// <o SL_EMLIB_GPIO_INIT_PIN_DOUT> DOUT <0-1>
// <i> In push-pull mode: The drive direction for the pin
// <i> In input mode: Pull-up (1) or pull-down (0)
// <i> In open-source mode: Set to 0 for the idle state
// <i> In open-drain mode: Set to 1 for the idle state
// <i> Default: 0
#define SL_EMLIB_GPIO_INIT_PIN_DOUT        1

// </h> end pin settings

// <<< end of configuration section >>>

// <<< sl:start pin_tool >>>

// <gpio> SL_EMLIB_GPIO_INIT_PIN
// $[GPIO_SL_EMLIB_GPIO_INIT_PIN]
#define SL_EMLIB_GPIO_INIT_PIN_PORT              gpioPortB
#define SL_EMLIB_GPIO_INIT_PIN_PIN               2
// [GPIO_SL_EMLIB_GPIO_INIT_PIN]$

// <<< sl:end pin_tool >>>

```

生成代码

调试

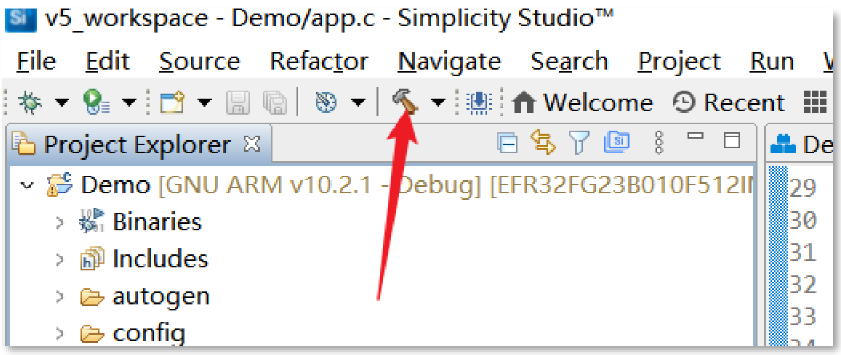
点击工程文件中的 app.c，在 app_init()函数中输入下面代码：



```
GPIO_PinModeSet(SL_EMLIB_GPIO_INIT_PIN_PORT, SL_EMLIB_GPIO_INIT_PIN_PIN, \
               SL_EMLIB_GPIO_INIT_PIN_MODE, SL_EMLIB_GPIO_INIT_PIN_DOUT);
```

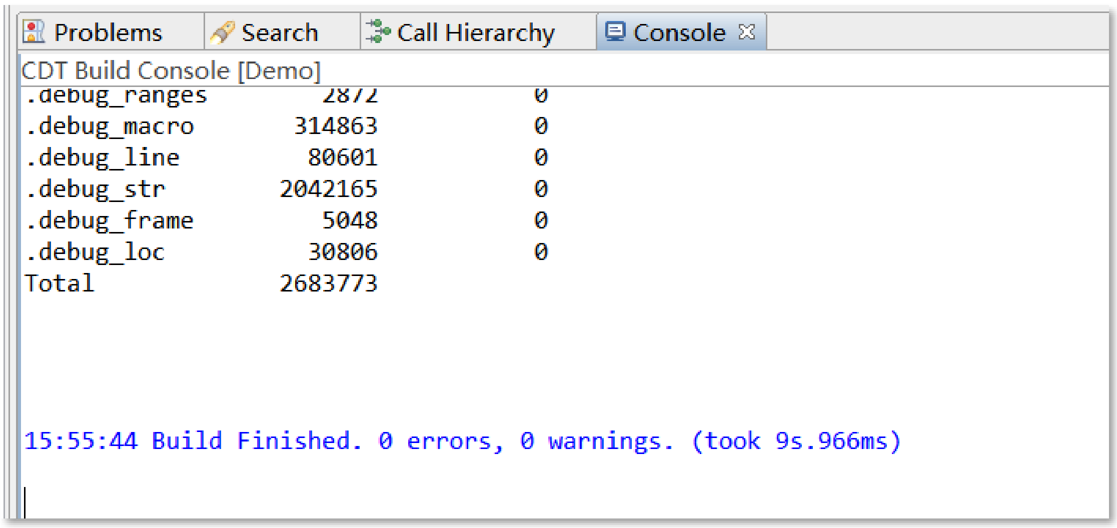
代码截图

然后点击工具栏中的构建按钮；



构建按钮

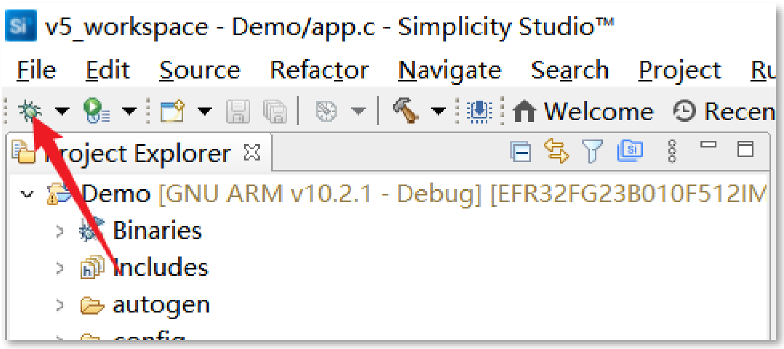
这时我们可以在底部控制台中看到构建成功的信息；



```
CDT Build Console [Demo]
.debug_ranges          2872          0
.debug_macro         314863          0
.debug_line           80601          0
.debug_str          2042165          0
.debug_frame           5048          0
.debug_loc            30806          0
Total               2683773



15:55:44 Build Finished. 0 errors, 0 warnings. (took 9s.966ms)
```

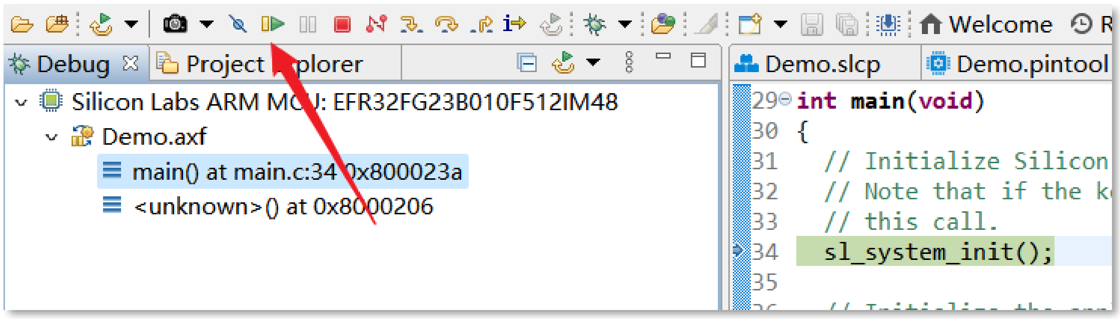
控制台

然后点击工具栏中的调试按钮；


v5_workspace - Demo/app.c - Simplicity Studio™
File
Edit
Source
Refactor
Navigate
Search
Project
Welcome
Project Explorer
Demo [GNU ARM v10.2.1 - Debug] [EFR32FG23B010F512IM
Binaries
Includes
autogen

调试按钮

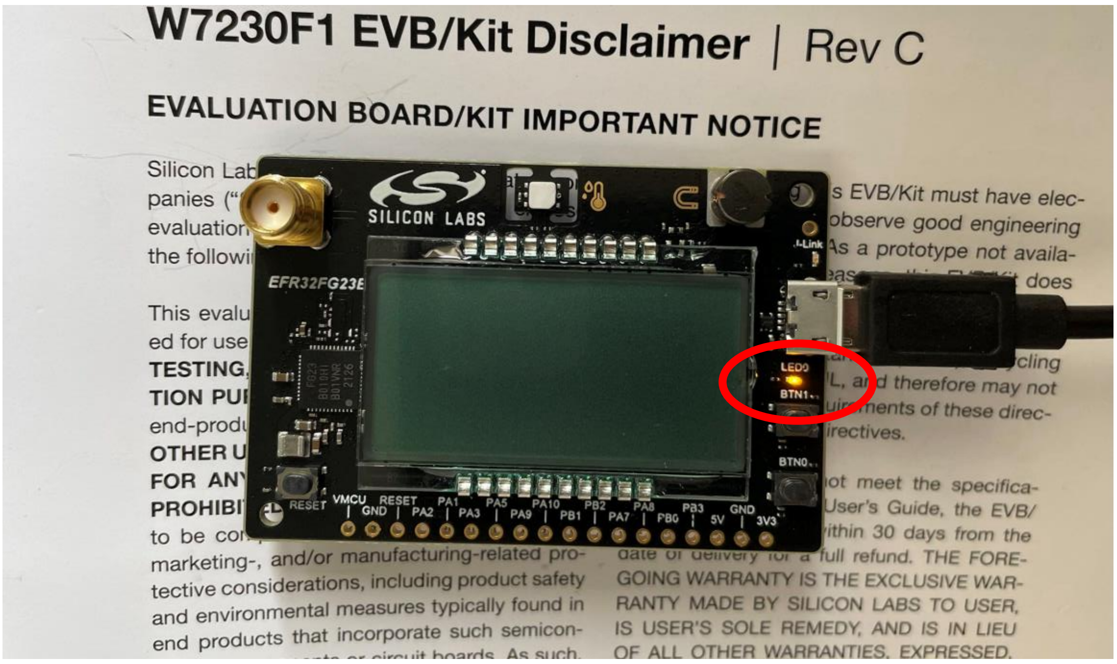
等到进程完成后，点击工具栏中的运行按钮，即可看到 LED 灯被点亮；



Silicon Labs ARM MCU: EFR32FG23B010F512IM48
Demo.axf
main() at main.c:34 0x800023a
<unknown>() at 0x8000206

```
int main(void)
{
  // Initialize Silicon
  // Note that if the k
  // this call.
  sl_system_init();
```

运行按钮


W7230F1 EVB/Kit Disclaimer | Rev C
EVALUATION BOARD/KIT IMPORTANT NOTICE
SILICON LABS
EFR32FG23B
LED0
BTN1
BTN0
RESET

LED 被点亮

显示温度

通过查询，发现官方在 GitHub 有开源的 Demo 代码库，下载下来后决定做一个进阶一点的功能，用 LCD 屏幕显示当前温度。

前期准备

下载 Demo 工程后导入 Simplicity Studio 中。通过用户手册可以得知 LCD 和温湿度传感器与芯片的硬件连接信息；

3.4.1 Segment LCD

A 20-pin segment LCD is connected to the EFR32FG23's LCD peripheral. The LCD has 4 common lines and 10 segment lines, giving a total of 40 segments. These lines are not shared on the breakout pads. Refer to the kit schematic for information on signals to segments mapping.

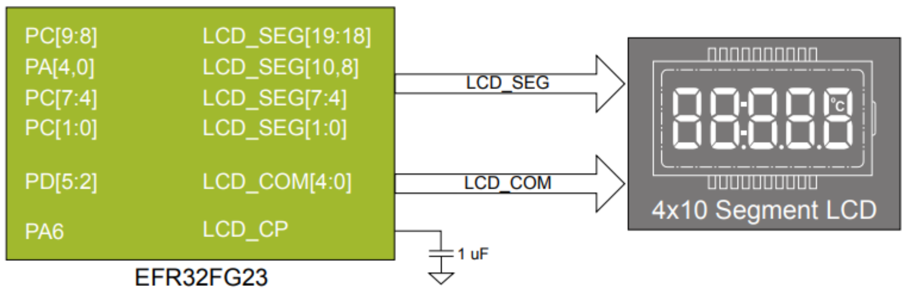


Figure 3.3. Segment LCD

LCD 硬件连接

3.4.2 Si7021 Relative Humidity and Temperature Sensor

The Si7021 I^{2}C relative humidity and temperature sensor is a monolithic CMOS IC integrating humidity and temperature sensor elements, an analog-to-digital converter, signal processing, calibration data, and an I^{2}C interface. The patented use of industry-standard, low-K polymeric dielectrics for sensing humidity enables the construction of low-power, monolithic CMOS Sensor ICs with low drift and hysteresis, and excellent long term stability. The Si7021 offers an accurate, low-power, factory-calibrated digital solution ideal for measuring humidity, dew-point, and temperature, in applications ranging from HVAC/R and asset tracking to industrial and consumer platforms.

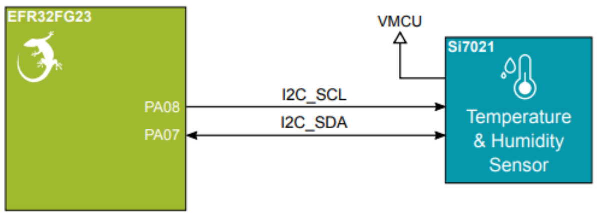


Figure 3.4. Si7021 Relative Humidity and Temperature Sensor

Although measures have been taken to thermally isolate the sensor from the board, temperature readings will be influenced when power is dissipated on the board. More accurate temperature measurements are achieved when powering the board with a battery or through the Mini Simplicity connector as self-heating from the on-board LDO is eliminated and the on-board debugger is put in a low power state.

温湿度传感器硬件连接

由此可以得知温湿度传感器的型号为 Si7021，通信方式为 I2C，通过查阅数据手册得到从机地址为 0x40，以及读取温度所需的指令；

5. I^{2}C Interface

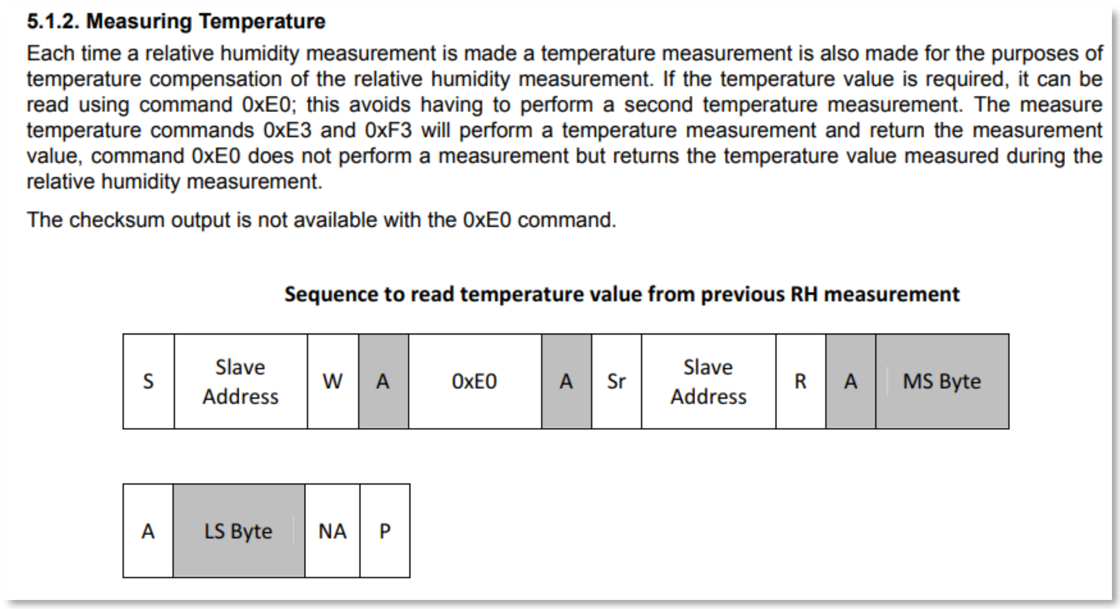
The Si7021 communicates with the host controller over a digital I^{2}C interface. The 7-bit base slave address is 0x40.

Table 10. I^{2}C Slave Address Byte

A6	A5	A4	A3	A2	A1	A0	R/W
1	0	0	0	0	0	0	0

Master I^{2}C devices communicate with the Si7021 using a command structure. The commands are listed in the I^{2}C command table. Commands other than those documented below are undefined and should not be sent to the device.

传感器从机地址



5.1.2. Measuring Temperature

Each time a relative humidity measurement is made a temperature measurement is also made for the purposes of temperature compensation of the relative humidity measurement. If the temperature value is required, it can be read using command 0xE0; this avoids having to perform a second temperature measurement. The measure temperature commands 0xE3 and 0xF3 will perform a temperature measurement and return the measurement value, command 0xE0 does not perform a measurement but returns the temperature value measured during the relative humidity measurement.

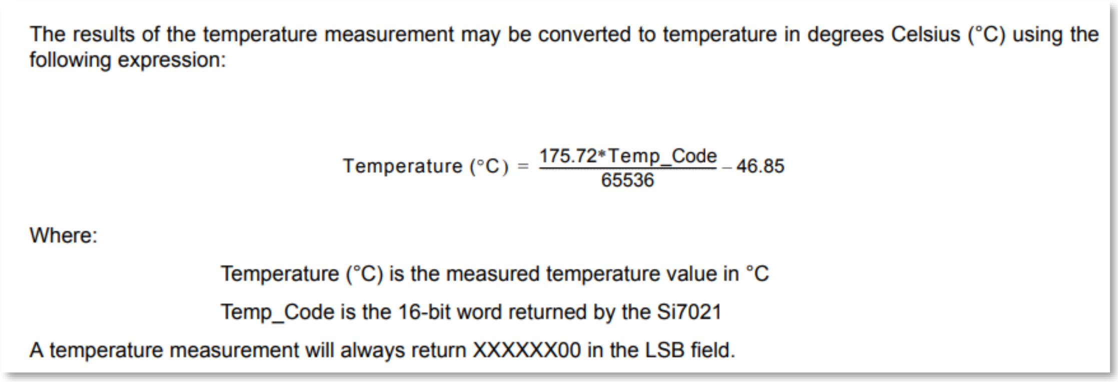
The checksum output is not available with the 0xE0 command.

Sequence to read temperature value from previous RH measurement

S	Slave Address	W	A	0xE0	A	Sr	Slave Address	R	A	MS Byte

A	LS Byte	NA	P

温度测量指令



The results of the temperature measurement may be converted to temperature in degrees Celsius (°C) using the following expression:

$$\text{Temperature (°C)} = \frac{175.72 * \text{Temp\_Code}}{65536} - 46.85$$

Where:

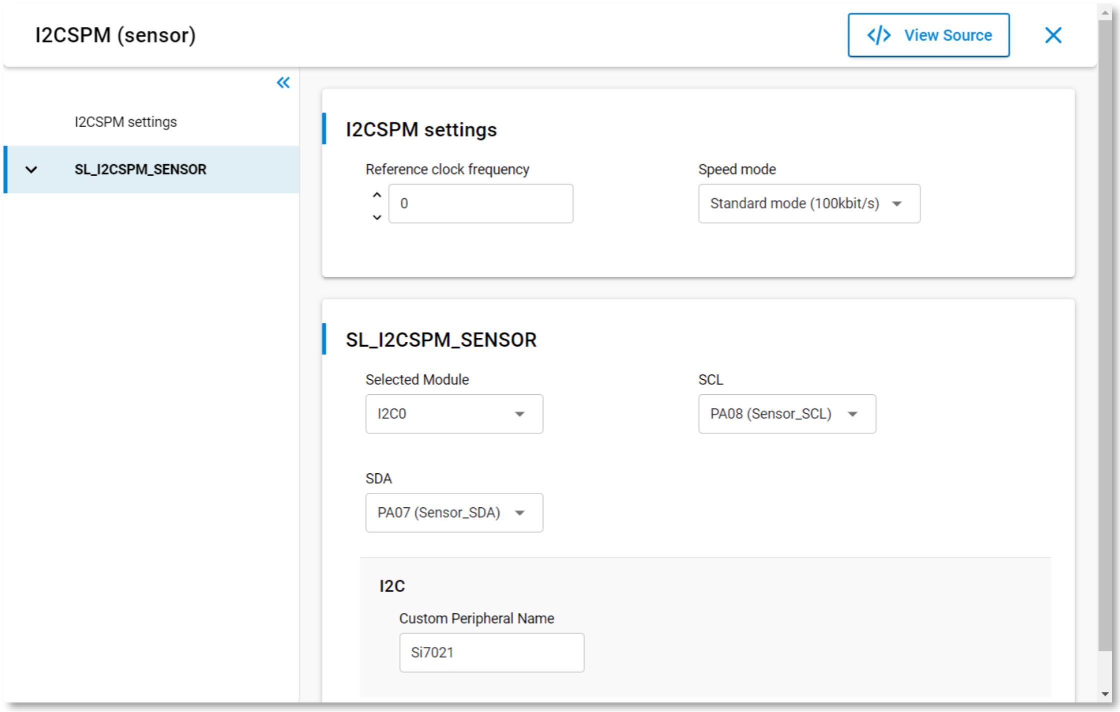
Temperature (°C) is the measured temperature value in °C

Temp_Code is the 16-bit word returned by the Si7021

A temperature measurement will always return XXXXXX00 in the LSB field.

温度转换公式

I2C PIN 脚配置；


I2CSPM (sensor)

I2CSPM settings

Reference clock frequency: 0

Speed mode: Standard mode (100kbit/s)

SL_I2CSPM_SENSOR

Selected Module: I2C0

SCL: PA08 (Sensor_SCL)

SDA: PA07 (Sensor_SDA)

I2C

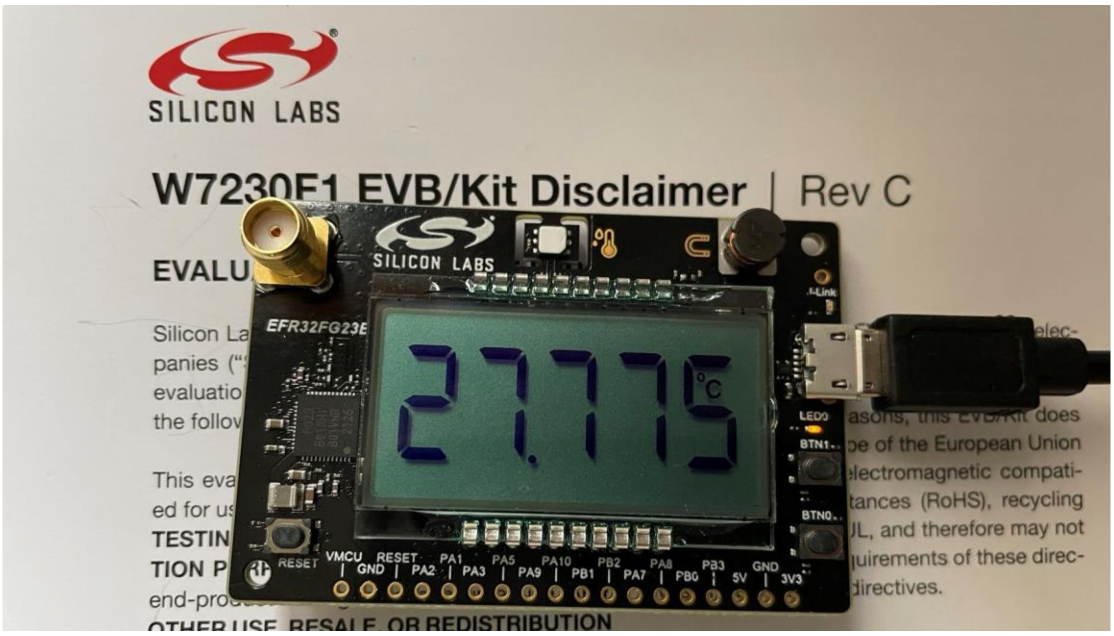
Custom Peripheral Name: Si7021

I2C 配置

调试

由于 demo 代码功能已经比较完善，通过调节部分参数即可满足需求。

代码调试过程和烧录过程此处与“点亮 LED”中的流程并无差异，此处不再赘述。下面是成品展示：


SILICON LABS
EFR32FG23E
27.775 °C
LED0
BTN1
BTN0
RESET
VMCU GND RESET PA2 PA1 PA3 PA5 PA9 PA10 PB1 PB2 PA7 PA8 PB0 PB3 5V GND 3V3

温度显示

5 总结

由于试用时间较短，还有很多很强大的功能有待学习。在试用过程中我也发现了一些小问题，对新用户来说可能不太友好。一个是开箱即用的 Demo 代码缺少维护，SDK 版本为 3.2.2，与最新 4.0 的不太兼容；另一方面是资料太少，不知道是不是由于保密原因，网上并未找到公开 EFR32FG23 芯片的数据手册，这样就无法查看每个寄存器的具体功能，不太利于深入学习。不过瑕不掩瑜，EFR32FG23 总体来说是一块非常优秀的物联网芯片，个人决定还是很具有竞争力的。好了，试用报告到此结束，但对开发板的学习才刚刚开始，期待它后期的表现！