# 第二章：使用WICED SDK连接输入和输出

## 目的

学完本章，您应能够为MCU外设（GPIO、PWM、ADC、UART和I2C）编写固件，以便与包括LED、开关、热敏电阻、模拟协处理器（Co-processor）（通过I2C）和KitProg UART等在内的子板连接。此外，您还将了解与套件硬件平台相关的关键文件（platform.h和platform.c）的作用。

## 时间：2小时

## 基础

### 设置新的WICED板支持包

WICED SDK提供的文件能简化与给定套件上的外设的工作。在我们的案例中，我们使用的套件被称为“BCM94343W\_AVN”。每个电路板都受SDK Workspace文件夹中的文件夹支持。我们套件的文件夹并非默认安装在SDK中，因此我们需要将平台文件拷贝到SDK Workspace中。用于该电路板的文件夹被命名为“BCM94343W\_AVN”并随本课程资料一起提供。从本课程资料中将整个文件夹复制到SDK Workspace中的“平台（platforms）”目录下。BCM94343W\_AVN包含：



两个关键文件platform.c和platform.h。platform.h文件包括用于设置和访问各种套件外设的#define和类型定义。例如套件电路板包含两个LED和一个按键。这些在platform.h中使用WICED\_LED1、 WICED\_LED2和WICED\_BUTTON1不同名称来识别。



platform.c文件含多个用于配置外设的常量数组和结构。该文件还包含用于初始化和控制外设的函数。例如LED引脚初始化为输出，按键初始化为按键的电阻上拉输入。

在platform.h中您将找到全部有效外设的清单。例如该套件的4个ADC通道包括：



在platform.c可找到用于每个ADC通道的引脚：



注意ADC名称必须在ADC API函数调用中使用。这就是说您必须使用*WICED\_ADC\_1来使用ADC通道1。*您不能在ADC API函数调用中使用*WICED\_GPIO\_2*。

如果您开发自己的硬件，最好给SDK Workspace平台文件夹添加一个新文件，以便存放您的硬件所用的合适文件。对任何不同的硬件连接，通常最简便的方法是复制现有平台并根据需要修改。

### 文档

文档可在SDK Workspace doc文件夹中找到。API.html文件包含我们要使用的API文档。用浏览器打开该文件，展开“组件（Components）”和“平台函数（Platform Functions）”，查看支持的组件清单（您也可以从WICED Studio中打开该文件）。我们将使用GPIO、PWM、ADC、UART和I2C。



点击GPIO查看GPIO API清单，然后点击*wiced\_gpio\_init函数查看描述。*



函数描述会告诉您函数的作用，但未说明配置结构相关信息。要找到该信息，如果您在WICED Studio内，您可以在C代码中高亮选定该参数，右键点击，并选择“打开声明（Open Declaration）”（您可以在后面练习中尝试这个操作）。如果您没有提供的有效参数，您可以在函数名称上使用“打开声明（Open Declaration）”，然后选择参数类型，接着类型名称来获得有效参数。这样操作会向您显示数据类型以及对允许的选择的说明：



### 创建新的WICED Studio项目

#### 目录结构

WICED Studio项目可以位于SDK Workspace的应用文件夹的任何位置。为方便起见，把现有的示例项目拷贝到新建名称的文件夹中比从头开始更加简便。项目的主要部分有：

用项目名称命名的文件夹。

位于项目文件夹内的名为 <project>.c的C源文件。

位于项目文件夹内的名为<project>.mk的生成文件（makefile）。

***重要提示：<project>名称必须与文件夹名称、C文件名称和生成文件名称相同。***

生成文件包含应用名称（任何唯一的字符串）和全部源文件清单（包括 <project>.c）。生成文件可能还包括给定项目的有效和/或无效平台清单、用于访问库的生成文件宏，以及其他资源（例如图像、网页等）。**生成文件中的应用名称必须是唯一字符串**。如果整个工作空间中任意两个项目有相同的应用名称，那么新建项目可能不会工作。在某些情况下新建项目可能进入无限循环，因为生成目标会从错误项目建立文件，随即无法找到正确的对象文件，迫使其不断要求重建（错误的）文件。因此建议在应用名称中使用包括文件夹路径在内的完整项目名称。

#### 生成目标

为下载项目到您的电路板上，您需要以下列形式创建新的生成目标：

*<folder1>.[<folder2>…].<project>-<platform> download run*

* <folder1>是应用文件夹下的文件夹名称。
* <folder2>、<folder3>等是项目名称路径的其余部分。您可以根据需要使用或多或少的额外文件夹名称。使用句点来分隔文件夹名称。
* <project>是项目名称。文件夹、主C文件和生成文件必须使用相同名称。
* <platform>是硬件平台的名称（即套件）。在平台目录中必须有一条与这里提供的名称相匹配的条目。

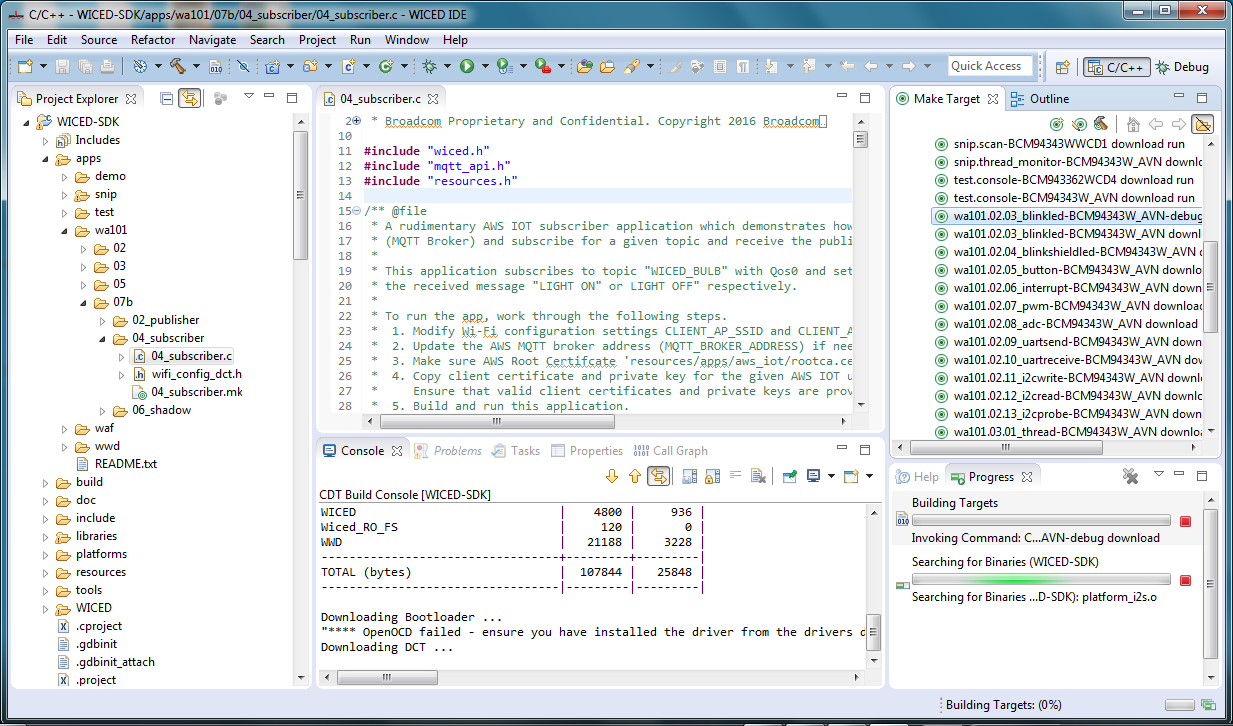
例如如果我们为我们的课程项目创建一个名为“wa101”的文件夹，为第二章的项目创建一个名为“02”的子文件夹并将第一个项目称为“01\_blinkled”，我们的电路板的新建项目目标将为：

*wa101.02.01\_blinkled-BCM94343W\_AVN download run*

已经定义的生成目标可以在WIKED Studio右侧的“生成目标（Make Target）”窗口中看到。展开“43xxx\_Wi-Fi”查看现有的生成目标。

要创建新的生成目标，您可以右键点击与您想要创建的生成目标相似的现有生成目标并选择*“新（New）…*。这样就能得到该生成目标的一个名称前带“副本（Copy of）...”的副本。删除“副本（Copy of）...”（切记删除空格！）并将修改名称用作您的新生成目标。

在有了生成目标后，您就可以双击生成目标，创建项目并为套件编程。您可以在“控制台（Console）”窗口中查看创建进展情况。如果您要中止一个正在新建的项目，您可以点击IDE右下角打开“进展”窗口，然后点击新建项目旁边的红框，如下图所示。



对BCM94343W\_AVN套件，该模块包含两个芯片：微控制器和Wi-Fi/蓝牙射频。生成目标选项“下载（download）”只用于下载固件到微控制器，但不会影响射频固件。在大多数情况下这样做已经足够，因为默认的射频固件不会改变。但是在某些情况下您可能收到有定制射频固件的套件，使它不能按预期的那样工作。在这样的情况下，您可以添加“download\_apps”到生成目标，下载一次无线电固件。也就是说您要使用：

*<folder1>.[<folder2>…].<project>-<platform> download download\_apps run*

#### C文件

您必须在主C文件顶层包含“wiced.h”。您还必须在主C文件的初始化段调用wiced\_init();函数。该函数可执行让其他WICED API正常工作所需的全部初始化工作并调用用于为套件初始化外设的函数。

### 外设

#### GPIO

如前文所述，GPIO必须在使用前初始化。套件上连接到LED和按键等专用外设的IO往往作为平台文件的一部分会自动初始化。

初始化完成后，使用*wiced\_gpio\_input\_get()就能读取输入引脚（pin），使用wiced\_gpio\_output\_high()和 wiced\_gpio\_output\_low()就能驱动输出。*这些函数的参数就是WICED引脚的名称，例如WICED\_GPIO\_1。或是您的平台的外设名称，例如WICED\_LED1。

GPIO中断使用*wiced\_gpio\_input\_irq\_enable()*和 *wiced\_gpio\_input\_irq\_disable()来控制。*

#### PWM

PWM提供的API函数可用于选择引脚、设置频率（Hz）和占空比（百分比）。该函数用于初始化和在PWM运作中修改频率或占空比。PWM还提供启动和停止输出的函数。请参阅API文档了解详细信息。

**请注意初始化函数不会启动PWM，因此您必须在第一次初始化PWM之后使用启动函数。**

如果输入的频率小于375Hz，将得到意想不到的频率。

#### ADC

ADC提供的API函数可用于选择通道（即引脚）和设置采样周期。它还拥有进行单采样或一系列采样的函数。样本为16位无符号值。请参阅API文档了解详细信息。

#### 调试打印

该SDK提供内置的调试打印功能，可用于通过内置在套件内的USB-UART桥接器显示消息。文件“wwd\_debug.h”可用于定于所有的不同消息类型。我们要使用的消息为“WPRINT\_APP\_INFO”，意即打印应用信息。这是一个使用标准*printf()格式化的宏。*它在SDK中是默认启用的（“wiced\_defaults.h”）。例如要打印“测试（test）”变量，您可以使用下列方式：

WPRINT\_APP\_INFO( (“测试值为:%d\n”, 测试) );

**注意圆括号内的额外部分是宏的定义方式要求的。**

#### UART

除了USB-UART调试打印功能，该设备还能使用“platform.h”文件中定义的STDIO\_UART并通过 Arduino UART引脚（D0和D1）传输标准UART数据。这些引脚也与板载的USB-UART桥接器相连，这样用于调试消息的终端窗口也同样能用于标准的UART通信。在BCM94343W\_AVN套件上，还有与Arduino引脚D8和D9相连的另一个UART（称为WICED\_UART\_2）。

有用于UART初始化、发送和接收的API函数。请参阅API文档了解这些函数的详细信息。

如果您在使用平台中定义的STDIO\_UART，那么您无需调用初始化函数，您无需按下文描述的设置环形缓冲区，因为这些函数已经从*platform\_stdio\_init()调用，而后者相应地从“platform.c”调用。*只有您在使用不同的UART接口或不同的UART设置时才需要调用这些初始化函数。STDIO\_UART默认设置为115200波特、8位带宽、无奇偶校验、无流程控制。

如果您想禁用STDIO\_UART功能或以不同的设置使用该接口，请将下列添加到项目的生成文件中：

GLOBAL\_DEFINES := WICED\_DISABLE\_STDIO

一旦进行这样的操作，您在终端上将不再看到显示的标准引导时间信息。

UART初始化函数需要带有下列元素的类型*wiced\_uart\_config\_t的配置结构。*其定义见“platform\_peripheral.h”。如上文所述，关于配置结构您可以在WICED Studio内高亮显示函数名称、参数类型、类型名称，右键点击并选择“打开声明（Open Declaration）”。



您还可以使用结构内每种类型上的“打开声明（Open Declaration）”找到有效选择。例如对数据宽度，可能的选择有：



如果您使用UART接收，您必须提供类型为*wiced\_ring\_buffer\_t的缓存。*该缓存必须使用*ring\_buffer\_init()* 函数初始化，这个函数需要指向环形缓冲区的指针、指向用于保持数据的数组的指针和缓存大小。例如下列可以用于创建名为rx\_buffer的10字节环形缓冲区：

#define RX\_BUFFER\_SIZE (10)

wiced\_ring\_buffer\_t rx\_buffer;

uint8\_t rx\_data[RX\_BUFFER\_SIZE];

ring\_buffer\_init(&rx\_buffer, rx\_data, RX\_BUFFER\_SIZE ); /\* Initialize ring buffer to hold receive data \*/

#### I2C

该设备包含WICED\_I2C\_1和WICED\_I2C\_2两个I2C主设备。

和其他外设一样，您需要使用初始化函数初始化该模块。但是在这种情况下您传递的参数不是模块的名称，而是类型*wiced\_i2c\_device\_t的结构。*该结构内含您要与之通信的I2C从设备的信息。例如下列可用于初始化I2C模块1以100KHz速度（标准速度）与地址0x08的从设备连接。



在发送数据之前，您需要设置类型*wiced\_i2c\_message\_t的消息结构。*有三个函数可用于这个目的：*wiced\_i2c\_init\_tx\_message()*、*wiced\_i2c\_init\_rx\_message()*、 *wiced\_i2c\_init\_combined\_message()*。请参阅API文档了解这些函数的详细信息。请注意“重试（retries）”参数必须设置为非零值（例如1）。0值的意思是一次消息发送尝试都不做。

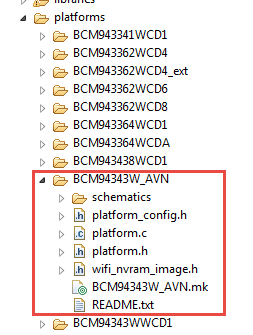
在该结构设置完成后，使用函数*wiced\_i2c\_transfer()* 发送或接收消息。

您还可以使用*wiced\_i2c\_probe\_device()*检查在给定地址是否存在I2C从设备。如果找到设备，该函数返回WICED\_TRUE；如果未找到，该函数返回WICED\_FALSE。注意在使用*wiced\_i2c\_probe\_device()之前必须仍然使用wiced\_i2c\_init()初始化设备。*

## 练习

### 01 (平台) 安装BCM94343W\_AVN到平台目录

1. 使用您在基础课程中学习的知识将BCM94343W\_AVN套件的文件安装到您的SDK Workspace。
2. 完成平台文件安装后，从WICED Studio内部右键点击平台文件夹，选择“刷新（Refresh）”。完成这一操作后，您应该看到BCM94343\_AVN文件夹和文件。如果您没有看到，请寻求帮助——勿进行下一步操作直到平台正确安装完毕。



回答下列问题：

您需要查看“platform.h”、“platform.c”和套件原理图，以回答下列问题。

WICED\_PWM\_1连接到哪个WICED GPIO引脚？

WICED\_PWM\_1连接到哪个模块？

WICED\_PWM\_1连接到哪个Arduino插头引脚？

### 02 (项目) 从模板设置新的项目

1. 在SDK Workspace内创建*43xxx\_Wi-Fi/apps文件夹，命名为“wa101”和命名为“02”的子文件夹。*
2. 从*WA-101 Files/projects/WICED/wa101key/02拷贝名为“02\_template”的项目。*
3. 将模板项目重命名为“03\_blinkled”。
   1. 提示：记得修改C文件和生成文件的名称，使之与项目名称匹配。
4. 必要时修改生成文件。
5. 为您的新项目创建生成目标。
   1. 提示：如果您右键点击现有的生成目标并选择“新（New）”，目标名称将以“副本...(Copy of)”开始，后接现有的目标名称。这样便于从熟悉的现有目标设置新目标。务必从新目标名称的开头删除“副本（Copy of）”（包括“of”后的空格）。

其余的项目要么使用套件，要么使用带有附加子板的套件。对需要子板的套件，会在项目标题中予以提示。对其他项目，如果您想更加方便地操作基板（即按键、环境光传感器、LED），可以移除子板（shield board）。

子板按设计从基板供电。无需将USB mini-B电缆连接到子板，除非您想重新为子板上的**PSoC 4编程。**实际上如果子板连接到USB mini-B电缆，基板就无法编程。

### 03 (GPIO) LED闪光

1. 修改03\_blinkled.c，让LED 1每250毫秒闪烁一次。
   1. 提示：请参阅API文档，了解用于驱动LED高/低电平的GPIO函数。
   2. 提示：使用*wiced\_rtos\_delay\_milliseconds()函数实现延迟。*
2. 编程您的项目到板上。
   1. 提示：一定在创建项目前保存文件，否则您会创建旧项目。如果您想让WICED Studio在每一次创建前自动保存任何修改过的文件，您可以设置“Window > Preferences（参考选项） > General（常规） > Workspace（工作空间） > Save automatically before build（创建前自动保存）”。

回答下列问题：

为什么不用*wiced\_gpio\_input\_get()* 函数而是用变量记忆状态来读出LED的值？

在哪个文件和在哪一行上WICED\_LED1被分配给套件里正确的引脚？

在哪个文件和在哪一行上该引脚被连接到设置为输出的LED上？

### 04 (高级) (GPIO) 触发未被平台文件预先初始化的引脚（需要子板）

1. 把您的项目从03\_blinkled复制到04\_blinkshieldled。根据需要修改生成文件并创建生成目标。
   1. 提示：完成此项操作或可从Windows的Explorer，或可使用右键点击、复制、粘贴和重命名从WICED Studio内部完成。
2. 连接模拟协处理器子板到套件。
   1. 提示：电路板上的USB连接器应位于相同侧。
   2. 提示：对任何此类练习都无需使用到模拟协处理器子板的USB连接器。子板从基板供电。如果到模拟协处理器子板的USB连接器被插上，基板就无法编程。
3. 修改项目，让子板上的绿色LED闪烁。这需要连接到Arduino引脚A1，A1又与SDK引脚WICED\_GPIO\_3相连。
   1. 提示：切记将该IO初始化为输出。请参阅API文档，了解该API函数所需的GPIO。
4. 编程您的项目到板上。

### 05 (GPIO) 读取按键状态

1. 本练习不需要子板。您现在可以先将其移除。
2. 将03\_blinkled项目复制到05\_button，更新生成文件，并创建生成目标。
3. 在C文件中，检查套件的按键输入状态（使用WICED\_BUTTON1）。让按下按键时，LED打开，松开按键时，LED熄灭。
4. 编程您的项目到板上。

### 06 (GPIO) 使用中断触发LED状态

1. 将05\_button项目复制到06\_interrupt，更新生成文件并创建生成目标。
2. 在C文件中，为连接到按键的GPIO设置下降沿中断。
   1. 提示：查看文档了解 *wiced\_gpio\_input\_irq\_enable()*。
   2. 提示：在您的C代码中进行下列操作：
      1. 在您的代码中输入*wiced\_gpio\_input\_irq\_enable()*。
      2. 高亮选中*wiced\_gpio\_input\_irq\_enable()*，右键点击并能选择“打开声明（Open Declaration）”。这样就会显示该函数所需的参数。
      3. 高亮选中*wiced\_gpio\_irq\_trigger\_t*，右键点击并选择“打开声明（Open Declaration）”。
      4. 高亮选中*platform\_gpio\_irq\_trigger\_t*，右键点击并选择“打开声明（Open Declaration）”。
      5. 找出用于下降沿中断的正确值。
   3. 提示：使用NULL把该参数传递给中断处理程序。
   4. 提示：对中断处理程序函数声明，使用(void\* arg)表达变元表。
3. 创建中断服务例程（ISR），让它在每次按键按下时触发LED状态。
4. 提示：您可以在ISR中使用静态布尔变量类型来记住LED状态：
   * 1. *static wiced\_bool\_t led1 = WICED\_FALSE;*
5. 编程您的项目到板上。

### 07 (高级) (PWM) LED亮度 (需要子板)

1. 将 04\_blinkshieldled 项目复制到07\_pwm，更新生成文件并创建生成目标。
2. 在C文件中，通过配置PWM（而不使用GPIO函数）来驱动子板上的绿色LED。
   1. 提示：绿色LED连接到WICED\_GPIO\_3，所以您需要找出哪个PWM与该引脚相连（查看平台文件）。
   2. 提示：切记在配置后启动PWM。
3. 修改主循环中的PWM占空比，这样LED会逐渐改变亮度。
   1. 提示：使用延迟，让亮度在一秒钟内从0%上升到100%。
4. 该项目需要子板。如果没有连接子板，请将它连接到套件。编程您的项目到板上。

### 08 (ADC/调试打印)读取环境光传感器并在电脑上打印值。

1. 创建名为08\_adc的新项目（或拷贝02\_template项目）。修改生成文件并创建生成目标。
2. 在C文件中：
   1. 初始化ADC，读取连接到环境光传感器的GPIO。
      1. 提示：查看“platform.h”文件，找出环境光传感器的#define。
   2. 在主循环中，从ADC读取值并使用WPRINT\_APP\_INFO宏打印值到屏幕上。
   3. 等一会儿（比如250毫秒）再获取下一个ADC样本。
3. 编程您的项目到板上。
4. 用波特率115200打开终端窗口，查看环境光传感器读数。
   1. 提示：该套件将出现在“端口(COM和LPT)（Ports (COM & LPT)）”下的设备管理器中，名称为*“WICED USB 串行端口（WICED USB Serial Port）”*。
   2. 提示：环境光传感器紧挨Arduino插头引脚D0。检测到的光越强，值的读数越小。使用您的手机手电筒即可看到明显变化。
5. 复位该套件，查看WICED固件显示的启动消息。

### 09 (高级)(UART)使用标准UART函数写入值

1. 将06\_interrupt项目复制到09\_uartsend。修改生成文件并创建生成目标。
2. 修改C文件，只要按下按键，按键按下的次数就会通过UART接口发送出去。为简单起见，只计数0到9然后归零，这样每次您只需要发送一个字符。
   1. 提示：加入下一行禁用生成文件中的STDIO\_UART：
      1. GLOBAL\_DEFINES := WICED\_DISABLE\_STDIO
   2. 提示：设置UART配置结构为波特率9600、数据宽度为8、无奇偶校验、1个停止位、无流程控制并初始化UART。
   3. 提示：在ISR内设置一个标志变量，然后在主应用循环中运行UART发送函数。确定把该标志变量定义为易失性全局变量。
   4. 提示：对读取缓冲使用NULL，因为我们只发送值。
3. 编程您的项目到板上。
4. 用波特率9600打开终端窗口。
   1. 提示：该套件将出现在“端口(COM和 LPT)（Ports (COM & LPT)）”下的设备管理器中，名称为*“WICED USB 串行端口（WICED USB Serial Port）”*。
5. 按下按键，观察终端上显示的值。

### 10 (高级) (UART)使用标准UART函数读取值

1. 将09\_uartsend复制到10\_uartreceive。更新生成文件并创建生成目标
2. 更新代码，让它查找来自UART的字符。如果接收到一个1，点亮一个LED。如果接收到一个0，熄灭一个LED。忽略任何其他字符。
   1. 提示：您需要设置一个环形缓冲区来接收UART字符。
   2. 提示：移除用于按键按压及其中断的代码。
3. 编程您的项目到板上。
4. 用波特率9600打开终端窗口。
   1. 提示：该套件将出现在“端口(COM和LPT)（Ports (COM & LPT)）”下的设备管理器中，名称为*“WICED USB 串行端口（WICED USB Serial Port）”*。
5. 按下键盘上的1和0键，观察LED打开/熄灭。

### 11 (高级) (I2C写入)修改子板上的LED状态（需要子板）

1. 复制06\_interrupt到11\_i2cwrite。更新生成文件并创建生成目标
2. 更新代码，使得当按键按下时会触发受子板上的模拟协处理器控制的红光和蓝光LED。模拟协处理器子板内置带下列属性的I2C从设备：
   1. 连接到Arduino引脚D14和D15（WICED\_I2C\_1）
   2. 7位地址=0x08
   3. 标准速度（100KHz）
   4. EZI2C寄存器访问
      1. 第一个写入的字节是寄存器偏移量。
      2. 所有读取从之前的写入偏移量开始。
   5. 寄存器映射如下：

|  |  |  |
| --- | --- | --- |
| 偏移 | 描述 | 详细信息 |
| 00 | 红光LED | 0 = 熄灭, 非零 = 打开 |
| 01 | 蓝光LED | 0 = 熄灭, 非零 = 打开 |
| 02 | 温度 | 温度的最低有效位 \* 100 |
| 03 | 温度 | 温度的最高有效位 \* 100 |
| 04 | 湿度传感器 | 湿度的最低有效位 \* 10 |
| 05 | 湿度 | 湿度的最低有效位 \* 10 |

* 1. 提示：要点亮红光LED并熄灭蓝光LED，您需要发送三个字节：0x00 (偏移)、0x01 (红光LED打开)、0x00 (蓝光LED熄灭)。
  2. 提示：在ISR中，只发送一个标志强制I2C更新。只有当设置标志后才在主应用循环中进行I2C处理。确定把该标志变量定义为易失性全局变量。
  3. 提示：确保“重试（re-tries）”的数量设置为1。值为0时，意味着根本无需发送消息！

### 12 (高级) (I2C读取) 读取传感器值 (需要子板)

1. 复制11\_i2cwrite到12\_i2cread。更新生成文件并创建生成目标
2. 更新代码，让每次按下按键时从I2C从设备读取温度和湿度数据。使用WPRINT\_APP\_INFO将值打印到终端。
   1. 提示：记得要将偏移量设置为02，以便读取温度。您只需要操作一次，以后的读取都会保持这个设置。在偏移量设置为02时，您可以读取4个字节来获取温度和湿度值。

### 13 (高级) (I2C探测)探测I2C设备(需要子板)

1. 复制11\_i2cwrite到13\_i2cprobe。更新生成文件并创建生成目标
2. 更新代码，这样每一次按键按下时就会对每个可能的I2C地址进行一次扫描。使用WPRINT\_APP\_INFO打印找到的任何设备的地址到终端（六进制）。
   1. 提示：I2C地址为7位。0x00是专门的“全部调用（All Call）”地址，所有高于0x7C的值都保留供以后使用途，因此有效地址仅有0x01-0x7B。
3. 在子板中找到什么地址呢？
   1. 提示：应该有3个地址，一个用于PSoC模拟协处理器，两个用于板载FRAM。

### 14 (高级) (平台) 生成/修改用于子板的平台文件(需要子板)

1. 我们没有创建完整的定制电路板，但我们在使用连接到基板的PSoC模拟协处理器子板。让我们为这个硬件配置生成一个平台，让我们的项目更便于读取。
2. 首先在平台文件夹中，将“BCM94343W\_AVN”复制到名为“WA101”的新文件夹。您可以直接在WICED Studio内完成这一操作。
3. 在新文件夹中，重新命名生成文件WA101.mk并删除原理图目录。
4. 更新README.txt文件。
5. 在“platform.c”和“platform.h”中添加下列新项目：
   1. SHIELD\_I2C -> 这将指向 WICED\_I2C\_1
   2. HIELD\_LED\_GREEN -> 这将指向 WICED\_GPIO\_3
      1. 提示：同时更新初始化功能，让WICED\_GPIO\_3初始化为推挽输出，如同LED1和LED2驱动为低电平。
6. 因为我们的基板和CPU没有变化，我们不需要在“platform.h”、“platform.c”、“platform\_config.h”或“Wi-Fi\_nvram\_image.h”内做任何其他修改。如果您要生成全新的电路板，比如使用不同的Wi-Fi设备，这些文件也需要更新。
7. 完成这些操作后，您返回并修改项目04和11，以使用新的定义。
   1. 提示：您将需要修改生成目标这样它们就能使用新的平台名称。
   2. 提示：在项目04上，您无需对绿色LED的GPIO进行初始化。

## 相关示例“应用（App）”

|  |  |
| --- | --- |
| **应用名称** | **功能** |
| snip.gpio | 演示如何读取连接到按键的输入，以及如何触发驱动LED的输出。 |
| snip.uart | 演示如何使用普通的WICED UART发送和接收字符。 |
| snip.stdio | 演示如何STDIO运算使用UART。 |

## 已知勘误 + 增强 + 注释

如果您更新到新版WICED，您的设置、项目和生成目标不会自动转移，您必须手动完成。