



联发科技 LinkIt™ Assist 2502 开发者指南

版本: 1.1

发行日期: 2015 年 6 月 17 日

本指南内容及产品规范如有变更，恕不另行通知。

© 2014, 2015 MediaTek Inc.

本文件包含的内容为联发科技有限公司所有。
严禁未经授权擅自复制全部或部分本文信息。

文件修改历史

版本	日期	说明
1.1	2015 年 6 月 15 日	

目录

1. 介绍	7
1.1. 什么是联发科技 LinkIt?	7
1.2. 联发科技 LinkIt Assist 2502 开发平台	8
1.3. 相关的开发平台	20
1.4. 加入联发科技创意实验室生态	20
2. 开始使用	21
2.1. 环境	21
2.2. 安装 Eclipse IDE C/C++	21
2.3. 安装 LinkIt Assist 2502 SDK 2.0	21
2.4. 更新固件	25
2.5. 创建您的第一个项目	27
2.6. 编译与上传硬件	31
2.7. 运行您的项目	33
2.8. 使用应用范例	33
2.9. 使用 LinkIt 工具栏	36
3. 故障排除	44
3.1. LinkIt Assist 2502 开发板插入 USB 线之后仍然无法启动	44
3.2. VXP 上传失败	44
3.3. 扫描 Wi-Fi AP 无法显示	44
3.4. 无法使用 GSM 功能	44
3.5. 无法使用 GPRS 功能	45
4. 编程指南	46
4.1. 驱动程序	46
4.2. LinkIt Assist 2502 应用的进入与退出点	47
4.3. 事件驱动编程模式	48
4.4. 线程	49
4.5. LinkIt Assist 2502 应用存储布局	49
4.6. 记录与命令	51
4.7. 周边硬件与驱动功能	51
4.8. 应用资源	53
4.9. 电源管理	57
4.10. 移植 Arduino sketche 和 驱动程序	58
5. API 指南	63
5.1. 基本类型	63

5.2. 返回值	63
5.3. Log	64
5.4. 命令	65
5.5. 标准库	65
5.6. 标签	70
5.7. 资源	71
5.8. 过程控制	72
5.9. 固件	72
5.10. 驱动 API	72
5.11. 网络	76
5.12. Wi-Fi	81
5.13. GSM	82
5.14. 蓝牙	83
5.15. 音频	86
5.16. GNSS	86
5.17. 图像	88
5.18. VXP 与固件更新	89
5.19. 智能手机通知	90

图表列表

表 1 LinkIt Assist 2502 开发板规格	13
表 2 LinkIt ONE 和 Assist 2502 开发板对照表	15
表 3 LinkIt 2502 API 函数概要	16
表 4 启用与关闭硬件组件功能	58
表 5 HTTPS 回调功能	81
图 1 可穿戴与物联网产品区隔	7
图 2 联发科技 LinkIt Assist 2502 开发平台组件	8
图 3 LinkIt Assist 2502 开发板 (背面)	10
图 4 LinkIt Assist 2502 开发板引脚图	12
图 5 LinkIt Assist 2502 SDK 2.0 设定应用	22
图 6 SDK 启动向导里的 Welcome 页面	22
图 7 选择与 Eclipse IDE 安装所在相同的文件夹	23
图 8 setup 向导的 Ready to Install 页面	24
图 9 SDK 安装完成页面	24
图 10 LinkIt ONE 大容量存储模式	25
图 11 LinkIt 固件更新开启页面	25
图 12 固件更新中	26
图 13 固件更新下载完成	26
图 14 创建新 LinkIt Assist 2502 应用	27
图 15 点选 LinkIt 2.0 应用	28
图 16 选择正确的硬件平台	29
图 17 LinkIt Assist 2502 函数库	30
图 18 您的第一个 LinkIt Assist 2502 应用	31
图 19 Eclipse IDE 内的 LinkIt 工具栏	31
图 20 LinkIt 工具栏的构建应用工具按钮	32
图 21 Eclipse 内的 LinkIt 控制台	32
图 22 LinkIt Assist 2502 代码范例	34
图 23 创建新的 LinkIt 应用	35
图 24 LinkIt 应用范例	36

图 25 应用设置工具按钮37

图 26 项目信息.....37

图 27 构建应用工具按钮 38

图 28 资源编辑工具按钮..... 39

图 29 显示工具按钮..... 39

图 30 显示工具..... 40

图 31 设备管理器 COM 端口..... 40

图 32 COM 端口连接..... 41

图 33 硬件连接记录..... 41

图 34 过滤工具按钮..... 42

图 35 发送 AT 命令窗口 42

图 36 API 参考工具按钮..... 43

图 37 应用存储布局..... 50

图 38 应用设置..... 66

图 39 应用设定里的标签设定..... 70

图 40 资源标志符范例..... 71

图 41 引脚名称与引脚号码映照图.....73

图 42 GATT 回调顺序图..... 85

图 43 启动流程顺序..... **Error! Bookmark not defined.**

1. 介绍

1.1. 什么是联发科技 LinkIt?

联发科技 LinkIt™ 是一组专门为可穿戴与物联网 (IoT) 原型创建所设计的开发平台。每个开发平台提供了完整的工具包，包含硬件和相关资源，让开发者应对三种可穿戴和物联网 (IoT) 领域的可能性，如图 1。

	单一应用 (OAU)	简单应用 (SAU)	丰富应用 (RAU)
范例	健身手环 健康手环 单功能蓝牙装置 智能灯泡 智能电器	智能手环 智能手表 孩童/老人安全手表/腕带/手环	高端智能手表 智能眼镜
硬件 (选配)	MCU (<100 MHz) 无线或蓝牙 传感器 LED 显示	MCU (100~300 MHz) 蓝牙 传感器 LED 或 TFT 显示 GSM/GPRS GPS Wi-Fi	AP (> 1GHz 的多核) 蓝牙 传感器 TFT 显示或透视显示器 GSM/GPRS GPS Wi-Fi
操作系统		大部分使用 RTOS 的核心	大部分使用带品牌的 Linux 为核心
价位	最低	普通	最高
电池寿命	久 >7 天	中等 5-7 天	短 2-3 天
特色	有限的运算能力，集中于单功能 (如运动、健康、搜寻装置，切换，设置) 大部分无显示器或仅有非常基本的 LED 显示	也许有多种功能并可以更新应用程序 也需要室外/室内定位为联发科技	多应用程序与功能 复杂的介面与更强大的图像与多媒体功能
			
联发科技 LinkIt Connect 开发平台和 MT7681 芯片组所专注的市场		联发科技 LinkIt ONE 和 LinkIt Assist 2502 开发平台与 MT2502 芯片组所专注的市场	

图 1 可穿戴与物联网产品区隔

每个开发平台还提供单一或多个芯片组和多种 API 变化，以满足特定的开发和设备的要求。为了让您创建原型设备，每个开发平台包括以下工具：

- 开发板和多种芯片模块的单一或多种 HDKs，让您的概念原型化。
- 创建设备固件或软件的 SDK。
- 硬件参考设计，包含最终产品电路板布局。
- 全面的文档，如 API 参考，开发者指南，芯片组描述和引脚图。
- 支持论坛。

1.2. 联发科技 LinkIt Assist 2502 开发平台

联发科技 LinkIt Assist 2502 开发平台专注于简单应用（SAU）可穿戴和物联网设备市场。这些设备包含智能手表，幼儿和老年人跟踪器，智能腕带和医疗可穿戴设备。

如图 2，该开发平台包含了一个系统级芯片组（SOC），API，硬件开发包（HDK），软件开发包（SDK），和全面的文档支持。

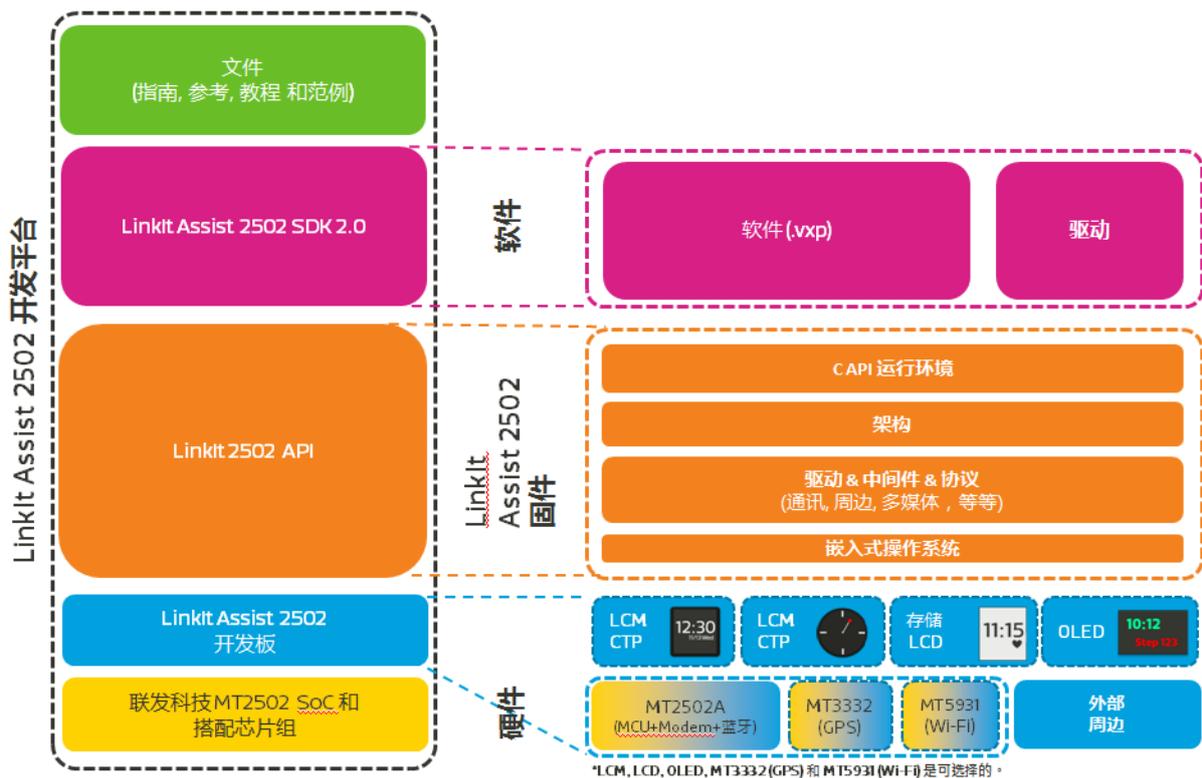


图 2 联发科技 LinkIt Assist 2502 开发平台组件

1.2.1. 系统级芯片 (SOC)

LinkIt 开发平台合并了世界上最小的可穿戴商用系统级芯片 (SOC)，联发科技 MT2502 (Aster)。该款 SoC 也可与联发科技的低功耗 Wi-Fi (联发科技 MT5931) 和全球导航卫星系统 (联发科技 MT3332) 的芯片组搭配。

1.2.2. LinkIt Assist 2502 硬件开发包 (HDK)

LinkIt Assist 2502 HDK 包含:

- 由 Seeed 制造的 LinkIt Assist 2502 开发板。该开发板提供了访问联发科技 MT2502 SoC 的硬件 LCD 引擎，使它适用于可穿戴设备原型。
- 由 AcSiP 制造的 LinkIt Assist 2502 模块包含:
 - LinkIt Assist 2502 模块 (AcSiP 模块型号 AI2502S05)
 - LinkIt Assist 2502 GNSS 模块 (AcSiP 模块型号 CW03S)
 - LinkIt Assist 2502 Wi-Fi 模块 (AcSiP 模块型号 CW01S)

1.2.2.1. LinkIt Assist 2502 开发板

该 LinkIt Assist 2502 开发板采用了三个 LinkIt Assist 2502 模块，如图 3。

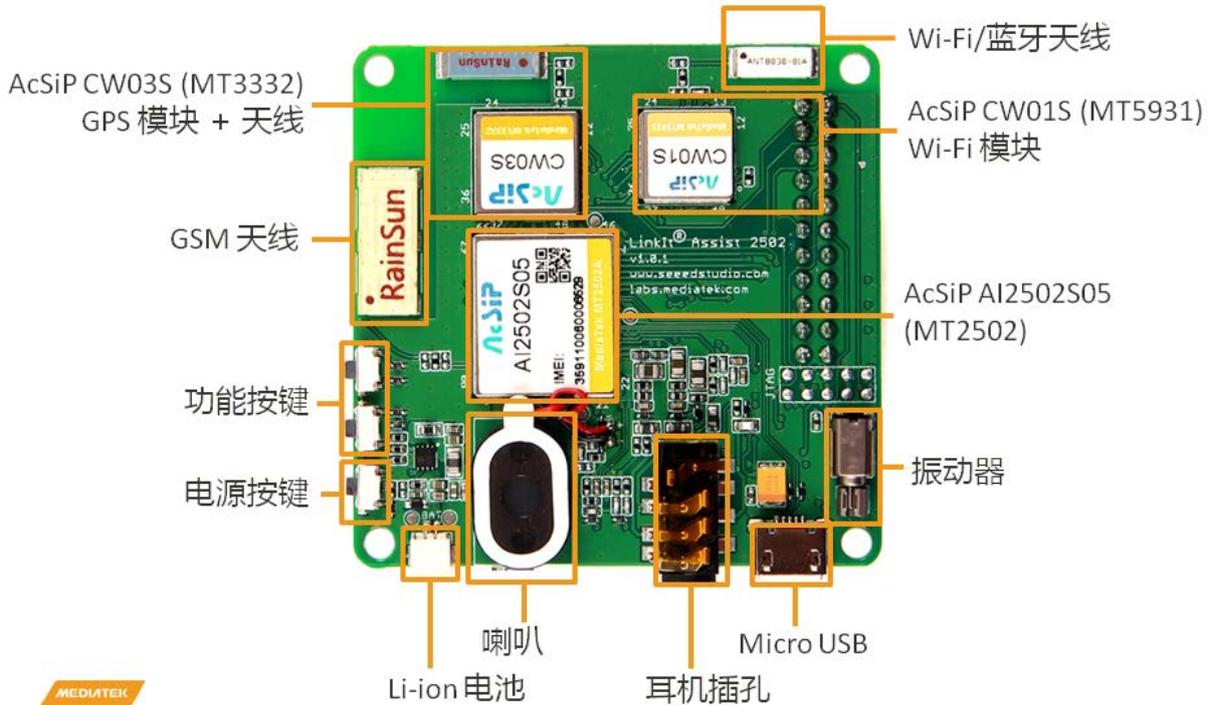


图 3 LinkIt Assist 2502 开发板 (背面)

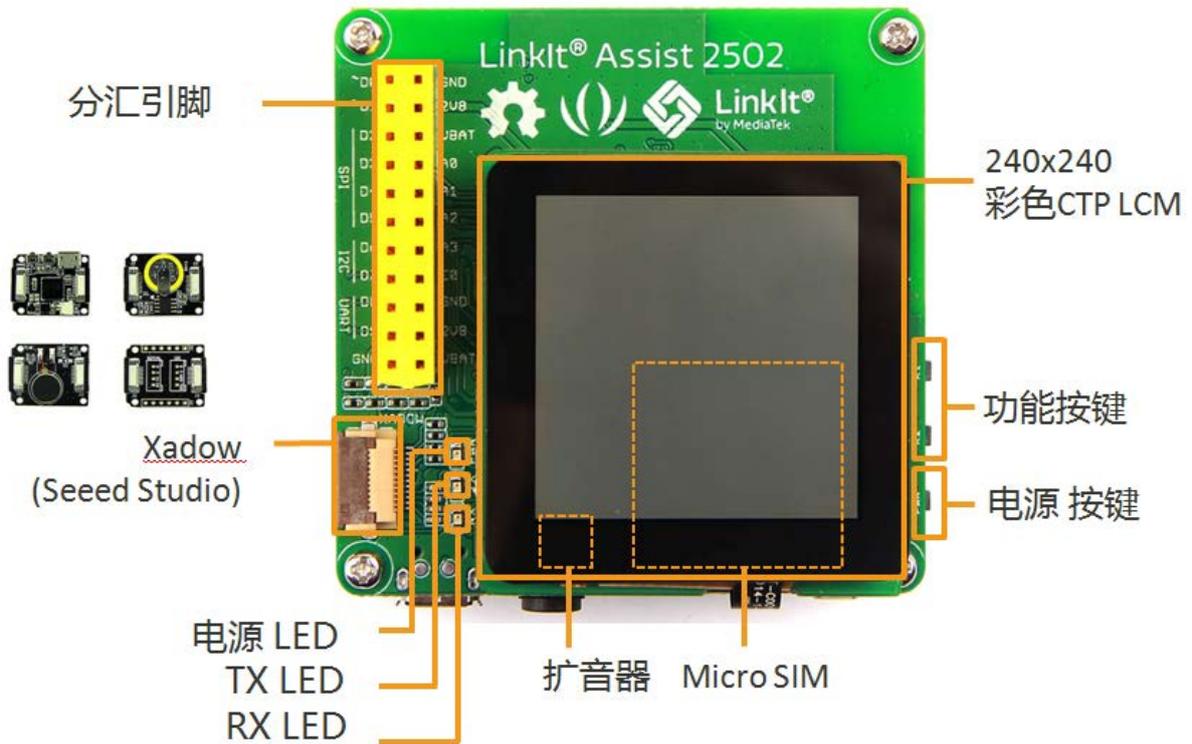


图4 LinkIt Assist 2502 开发板 (正面)

每个联发科技 LinkIt Assist 2502 模块包含一个联发科技的 SoC 及相关射频组件以提供一个易于使用的硬件接口。每个模块都提供原理图，布局和物料清单让您设计您的 MT2502 产品。硬件开发周期可因通过本模块的参考设计而被缩短，并让您的产品更快的市场化。

该平台提供了一个全方位的通信，定位和多媒体功能。这使其适合于设计和开发多元化可穿戴设备，如手表，手环，跟踪器，和其他移动物联网设备。通信协议支持 GSM，GPRS，蓝牙 2.1 和 4.0 以及 Wi-Fi。GNSS 定位包括 GPS，GLONASS 和北斗系统。

该 LinkIt Assist 2502 开发板的完整规格如下表 1 提供。

表 1 LinkIt Assist 2502 开发板规格

类别	功能	规格
微控制器模块	模块名称	AcSiP AI2502S05
	芯片组	MT2502A (Aster)
	核心	ARM7 EJ-S™
	主频	260MHz
Wi-Fi 模块	模块名称	AcSiP CW01S
	芯片组	MT5931
GPS 模块	模块名称	AcSiP CW03S
	芯片组	MT3332
PCB 大小	尺寸	5.33 cm x 5.33 cm
存储	闪存	16MB
	RAM	4MB
电源	电池插孔	3.7~4.2V Li-ion battery
	充电	Micro USB
数字 IO 引脚	引脚数量	14
	电压	2.8V
Analog 输入引脚	引脚数量	4
	电压	0~2.8V ⁽¹⁾
PWM 输出引脚	引脚数量	2
	电压	2.8V
外部中断	引脚数量	2
	电压	2.8V
I2C (master only)	组数	1 (SDA, SCL)
	速度	100Kbps, 400Kbps, 3.4Mbps
SPI (master only)	组数	1 (MOSI, MISO, SCK)
	速度	104Kbps~26Mbps
UART	组数	1 (Rx, Tx)
UART on USB ⁽²⁾	组数	1

类别	功能	规格
Xadow (Seeed Studio)	组数	1
	Voltage	Signal 2.8V, Power 3.3V
通讯	GSM	850/900/1800/1900 MHz
	GPRS	Class 12
	蓝牙	2.1 SPP 和 4.0 GATT (Dual Mode)
	Wi-Fi	802.11 b/g/n
定位	GNSS	GPS/GLONASS/北斗
屏幕 (可换)模块	LCD	Truer 240x240 半透
	驱动 IC	ST7789S

(1) 注意:电压超过最大允许时将损坏引脚

(2) SDK 包含 UART USB 所使用的监视工具以显示应用记录和系统记录 .

1.2.2.2. 相关的 HDK

LinkIt ONE 开发板也是基于联发科技 MT2502 的 SoC。它可以运行联发科技 LinkIt Assist 2502 的应用程序，但具有一定的局限性。这两个开发板的比较，如表 2 规定。

	LinkIt ONE	LinkIt Assist 2502
		
用途	概念验证	原型化
支持 LinkIt SDK 1.0 (针对 Arduino)	支持	不支持
支持 LinkIt Assist 2502 SDK	支持	支持
板输出引脚设计	Arduino-类型 引脚布局	专有设计
大小	84 x 53mm	53 x 53mm
引脚电压	3.3V	2.8V
额外周边接口	Grove (Seeed Studio)	Xadow (Seeed Studio)
内建按键	0	2
SD 卡	有	无
SIM	SIM	Micro SIM
电源	Micro USB (5V) 电池 (可选择)	Micro USB (5V) 电池 (需要)
音频	耳机 (包含麦克风)	耳机, 扩音器和麦克风
振动器	无	有
天线	外部	内置 陶瓷天线

表 2 LinkIt ONE 和 Assist 2502 开发板对照表

关于更多 LinkIt ONE 开发板详细信息, 请阅读参考 [联发科技 LinkIt™ ONE 开发者指南](#).



为了符合 Arduino, LinkIt ONE 开发板设计包含专用电路, 例如把 2.8V 引脚电压转为 3.3V, 而导致最终产品增加成本。因此, 这些电路没有包含在 LinkIt Assist 2502 开发板内。

1.2.3. LinkIt Assist 2502 API

LinkIt Assist 2502 API 提供基于 C 语言的模数，使编程访问该平台系统芯片的特点，搭配芯片组，显示器和连接周边设备。更多 API 细节在第 5 章节“API 指南”提供。

LinkIt Assist 2502 API 是用于创建 LinkIt Assist 2502 应用在 LinkIt Assist 2502 SDK，如图 2。它是个用 C 语言编程基础的 API 函数，以及第三方库和设备驱动程序进行。该应用程序被编译成 LinkIt Assist 2502 可执行文件（VXP 文件），并由 LinkIt Assist 2502 开发板固件所提供的运行时环境执行并支持动态负载从闪存运行应用。

使用 C++ 开发 LinkIt Assist 2502 应用程序是可行的因为软件工具链支持 C 和 C++ 编译的文件。如果您使用 C++ 编程代码，请产生项目之后将应用源代码文件从 *.c 改成 *.cpp，但是请注意，LinkIt Assist 2502 API 提供 C 函数，而不是 C++ 函数，并且某些 C++ 库有可能依赖执行环境函数，但是这些可能 LinkIt Assist 2502 开发平台是不支持的。

LinkIt Assist 2502 应用程序在多线程环境中执行。在这种环境下，有系统管理线程如驱动线程负责管理集成的硬件，如 GSM 调制解调器和蓝牙协议栈。LinkIt Assist 2502 应用运行是在一个特定的线程称 *main thread*。LinkIt Assist 2502 应用还可以创建新的线程。然而，大多数 LinkIt Assist 2502 SDK2.0 API 函数只能从 *main thread* 调用。关于被已创建的线程所叫的函数细节，请参阅 API 参考文件。

由于多线程，多任务环境的性质，LinkIt Assist 2502 应用采用基于事件的编程模型。这种编程模型的更多信息将在章节第 5.5.5 “线程”提供。

1.2.3.1. API 概要

LinkIt 2502 API 函数概要由以下表 3 陈述。

表 3 LinkIt 2502 API 函数概要

函数	功能
核心	
存储	提供存储分配，释放和类似函数。
定时器	提供三种定时器：精确定时，不精确定时和 HISR 定时。
线程	提供线程，信号和互斥量编程功能。
字符转换	定义 ASCII，USC-2，UTF-8 和 UTF-16BE 的字符转换功能。
文件系统	提供文件和和文件夹相关操作。
日期 & 时间	设置并检索系统时间，或获取微秒计。
记录	由 SDK 提供的监视工具输出记录并显示严重程度。
资源	VXP 文件资源访问，例如字符串，影像和音频数据。

函数	功能
固件	提供 API 查询固件信息，并触发固件更新过程。
电源管理	启用关闭并重新启动系统，还提供电池和充电信息。
用户互动	
按键	注册按键事件处理程序，如果 HDK 提供按键硬件。
触控	注册触控事件处理程序，如果 HDK 和固件能支持。
界面与网络	
驱动共享层	提供控制和访问硬件设备的（读取和写入）功能，硬件包括 ADC，GPIO，I2C，SPI，UART，EINT，PMU 和 PWM。
网络协议	支持 TCP, UDP, SSL, HTTPS 和 HTTP。
电话与移动数据	
SIM	提供从设备中的 SIM 卡访问信息功能。
电话服务	提供控制语音通话功能。
SMS	提供发送，阅读，删除和取消短信功能。
蜂窝通信信息	提供获得当前或相邻小区的信息的功能。
GPRS	提供配置 GPRS 连接功能。
Wi-Fi	
WLAN	提供配置 Wireless LAN 无线连接功能。
Location	
GPS	提供 GPS 和 EPO（离线辅助定位）功能。
蓝牙	
蓝牙	提供应用进行配置，监控和搜索蓝牙连接功能。蓝牙 SPP API 支持客户端和服务端一对一的连接功能。
通知服务	提供从 iOS 和 Android 设备接收通知功能。
蓝牙 Low Energy	提供蓝牙 4.0 的通用属性配置文件（GATT）支持。
图像	
图像	执行像素操作 绘制点，线，矩形和椭圆图 解码和画图像 采用矢量字体输出文本 在支持的硬件上执行 LCD 操作

函数	功能
音频	
音频	播放和录制音频文件功能

1.2.3.2. 相关 API

除了核心 API 之外，LinkIt Assist 2502 开发板还提供以下相关 API 函数：

- LinkIt Assist 2502 智能手机通知 API，它通过蓝牙连接提供接收通知功能，包括来自苹果通知中心服务（ANCS）的通知。请参阅 [LinkIt Assist 2502 智能手机通知开发者指南](#) 了解更多信息。
- Android 和 iOS 适用的 LinkIt Smart Connection 库，该 API 提供从智能手机或平板广播无线 AP 设置至 LinkIt 2502 开发板或设备。请参阅 [LinkIt Smart Connection 开发者指南](#) 了解更多信息。

1.2.4. LinkIt Assist 2502 软件开发包 (SDK)

联发科技 LinkIt Assist 2502 SDK 2.0 让您为开发平台创建应用。SDK 使用指南细节请参考章节 2 “开始使用”。

1.2.4.1. 功能概要

SDK 包含以下功能：

- **固件更新**替开发板安装对应固件。该固件提供二进制格式。
- **Eclipse IDE 插件 (CDT 版)** 包含：
 - **Eclipse 项目启动向导**，为您开始编程 LinkIt Assist 2502 应用程序创建一个项目框架。
 - **Eclipse 工具栏**定义适当的应用程序设置和编译编译器选项来为您 LinkIt Assist 2502 应用程序构建 VXP 执行档。
 - **资源编辑器**，使您能够为 VXP 执行档嵌入图像和字符串并可以通过 SDK 的资源模块复取。
- **上传工具**让您通过 USB 接口上传 VXP 执行档来连接开发板。

1.2.4.2. 开发过程概观

您首先将使用 Eclipse 项目启动向导来创建一个项目架构或打开一个范例项目并开始在 Eclipse CDT 里编程 LinkIt Assist 2502 应用程序，然后从 Eclipse 的工具栏选项执行代码编译和与开发板链接。

生成的可执行文件（VXP）将经由上传工具被上传到开发板的内部存储。接下来，开发板固件重新启动，并执行上传的应用程序，使访问底层框架和内置驱动程序的功能。

1.2.5. 文件与范例

为协助 LinkIt Assist 2502 原型软件开发，一共有五个参考文件供您使用：

- 本[开发者指南](#)，在联发科技创意实验室网站查询[最新](#)的版本。
- [LinkIt Smart Connection 开发者指南](#)。本文件提供了有关如何使用 LinkIt Smart Connection 来规定 LinkIt 开发板和 Wi-Fi AP 之间的 Wi-Fi 连接。它还包括如何使用安卓和 iOS 库来构建一个智能手机应用程序，用于定义一个 AP，并发送信息到开发板的详细信息。
- [LinkIt Assist 2502 智能手机通知](#)。本文件提供了在 LinkIt Assist 2502 设备上可接收智能手机通知的 API 详细信息。
- [LinkIt Assist 2502 开发板输出引脚图](#)：该图提供开发板上的分汇引脚细节。
- [LinkIt Assist 2502 API 参考](#)。本参考提供多种模块功能，参数细节以供开发平台软件编程。
- 代码范例。这些范例是 SDK 的一部分并可由 Eclipse 项目向导访问。

除了以上文件，以下还有更多资源来协助您创建硬件产品设备：

- [LinkIt Assist 2502 硬件参考设计](#)：该文件包含：
 - LinkIt Assist 2502 开发平台原理图与布局图
 - LinkIt Assist 2502 开发板引脚图
 - LinkIt Assist 2502 引脚复用表
 - LinkIt Assist 2502 模块规格书 (AcSiP)
 - 芯片组规格书

请注意：本 PCB 布局图与原理图由 [Eagle](#) 开发。您可以查寻[免费的 Eagle Light 版本](#)。

- [MT2502 \(Aster\) 芯片组技术简介](#)
- [MT3332 \(GNSS\) 芯片组技术简介](#)
- [MT5931 \(Wi-Fi\) 芯片组技术简介](#)
- [LinkIt Assist 2502 模块规格书 \(AcSiP AI2502S05\)](#)
- [LinkIt Assist 2502 GNSS 模块规格书 \(AcSiP CW03S\)](#)
- [LinkIt Assist 2502 Wi-Fi 模块规格书 \(AcSiP CW01S\)](#)

更多其他文档会不定期的在联发科技创意实验室开发平台[文档](#)网页提供。

1.3. 相关的开发平台

联发科技 LinkIt ONE 开发平台也是基于联发科技 MT2502 芯片组。然而，LinkIt ONE 开发平台提供了一组类似 Arduino 开发板使用的 API，通过 Arduino 的 API 和运行环境之间的移植层实现。而软件开发是由一个 SDK 提供一个插件到 Arduino 软件中启用。

另外 LinkIt ONE 应用和 LinkIt Assist 2502 应用有着软件架构上的根本区别。LinkIt ONE 应用是由 setup() 和 loop() 函数执行，而 loop() 函数会使系统维持在一个无尽的轮询状态。这种编程风格是非常简单的，但它会导致功耗增加的缺点。然而，LinkIt Assist 2502 的编程风格，是使用了低功耗基于事件的编程方法。请参阅章节 4.2，“事件驱动编程模型”，关于事件编程模型的细节。

LinkIt ONE 开发板提供的功能与 LinkIt ONE API 兼容。为 LinkIt Assist 2502 开发板创建的软件可以在 LinkIt ONE 开发板上运行，但某些功能可能不能运作因硬件能力的不同。

请参阅[联发科技 LinkIt™ ONE 开发者指南](#)关于 LinkIt ONE 开发板信息。

1.4. 加入联发科技创意实验室生态

可穿戴与物联网是消费小装置革命的下一波浪潮。联发科技创意实验室在该领域是个关键者，我们结合了两个最好的世界 - 现有的联发科技手机生态系统制造商，电子设备，电信运营商与一个开放的，充满活力的创客和开发者社区。

无论您是一个制造商，设备制造商，学生，DIY 爱好者，还是程序员，您都会在联发科技提供的开发平台找到满足您创建新概念的需求。然后利用[联发科技创意实验室合作伙伴服务](#)的优势将您的创意推向市场。

请在 labs.mediatek.com 注册并加入联发科技创意实验室生态。

2. 开始使用

本章节提供开始使用联发科技 LinkIt Assist 2502 开发平台指南并包含以下介绍：

- 开发平台所支持的环境。
- 安装 Eclipse IDE C/C++。
- 安装与配置 LinkIt Assist 2502 SDK 2.0。
- 创建您的第一个项目并在开发板上运。
- 探索代码范例，以完成特定任务。
- 使用 SDK 的其他工具。

2.1. 环境

在使用联发科技 LinkIt Assist 2502 SDK 和开发板之前，请确保您的计算机已安装好以下软件：

- 操作系统：微软 Windows XP, Vista, 7 或 8。
- Eclipse IDE: Indigo (3.7) 和 CDT 8.0.2。

2.2. 安装 Eclipse IDE C/C++

该 LinkIt Assist 2502 SDK2.0 包含一个 Eclipse IDE 插件，并与 Eclipse IDE Indigo 或更高版本兼容。请从 [Eclipse 官方网站](#) 下载和安装 Eclipse IDE 并确保它具有 C / C++ 开发 (CDT) 插件。在安装 IDE 时可能需要适当的 Java SE 版本安装在您的电脑上。

该 LinkIt Assist 2502 Eclipse 插件必须安装在与 Eclipse IDE 同一个文件夹中，同等于其 eclipse.exe 档所在。

2.3. 安装 LinkIt Assist 2502 SDK 2.0

安装 LinkIt Assist 2502 SDK 2.0 之前您需要做一些准备：

- 1) 下载 [LinkIt Assist 2502 SDK 2.0](#)
- 2) 解压 LinkIt Assist 2502 SDK2.0 的压缩文件。另外，除了 Eclipse 插件之外，SDK 里还有其他工具，所以请您考虑解压该 SDK 内容至一个固定位置。如果您解压内容到一个临时位置，安装程序会提供移动到固定位置的选项。
- 3) 确认 Eclipse IDE 没有在运行。
- 4) 电击执行 LinkIt Assist 2502 SDK 2.0 InstallPlugins 安装程序，如图 6。

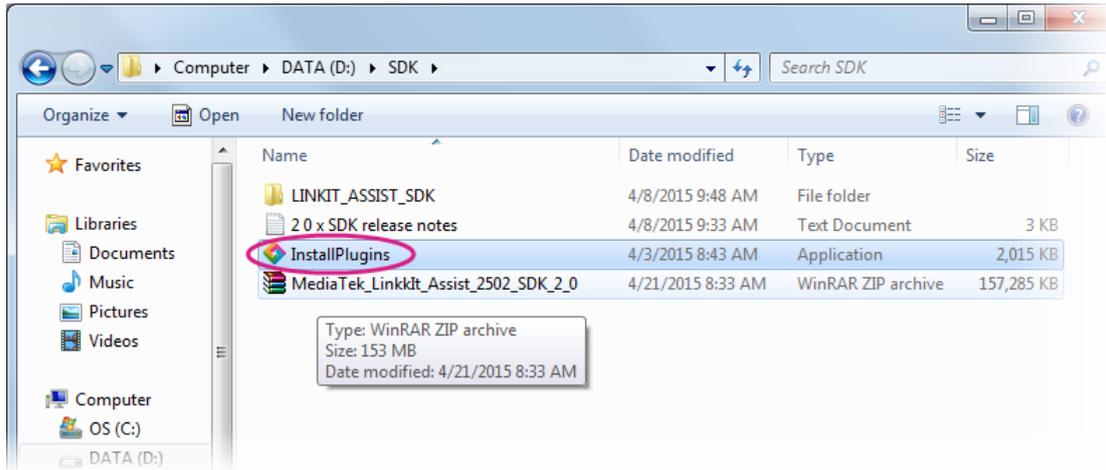


图 6 LinkIt Assist 2502 SDK 2.0 设定应用

5) 在 Welcome 页面, 如图 7, 电击 **Next**.

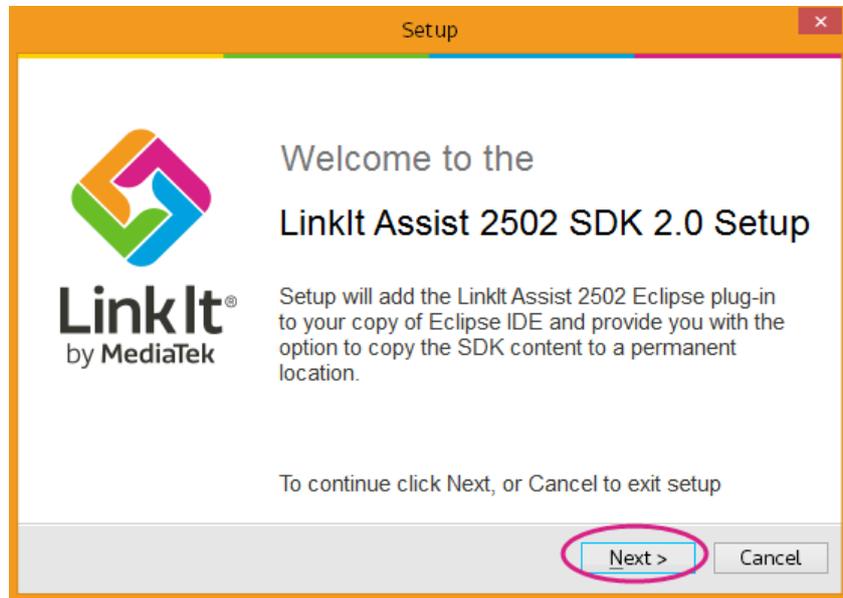


图 7 SDK 启动向导里的 Welcome 页面

- 6) 在 **Select Eclipse IDE Location** 电击 **Browse** 并指定您安装 Eclipse IDE 的文件夹 (如章节 2.2 “安装 Eclipse IDE C/C++ ” 提到的文件夹), 并电击 **Next**。如图 8。

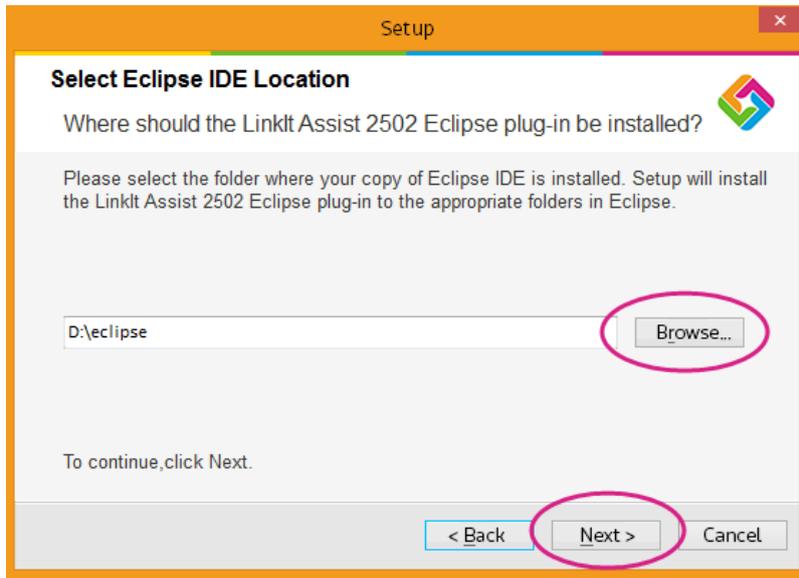


图 8 选择与 Eclipse IDE 安装所在相同的文件夹

- 7) 如果您将 SDK 解压至临时位置但需要将它移动至一个固定的位置, 请在 **Move SDK to permanent location** 页面中的 **Move LinkIt SDK 2.0 package to new folder** 复选框打勾, 再点击 **Browse** 以选择文件固定的位置, 最后点击 **Next** 如图 9。

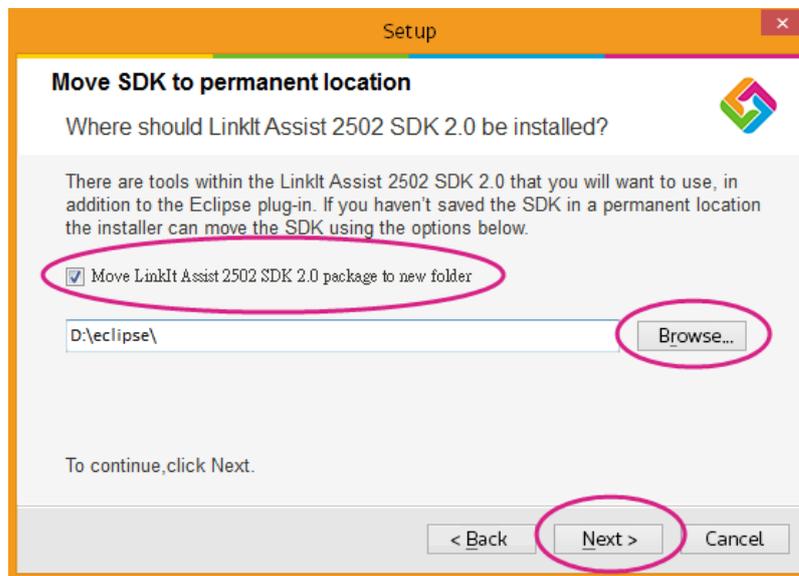


图 9 将 SDK 移动至固定位置

8) Ready to Install 页面出现后，检查文件路径，如果 OK 请电击 install 如图 10。

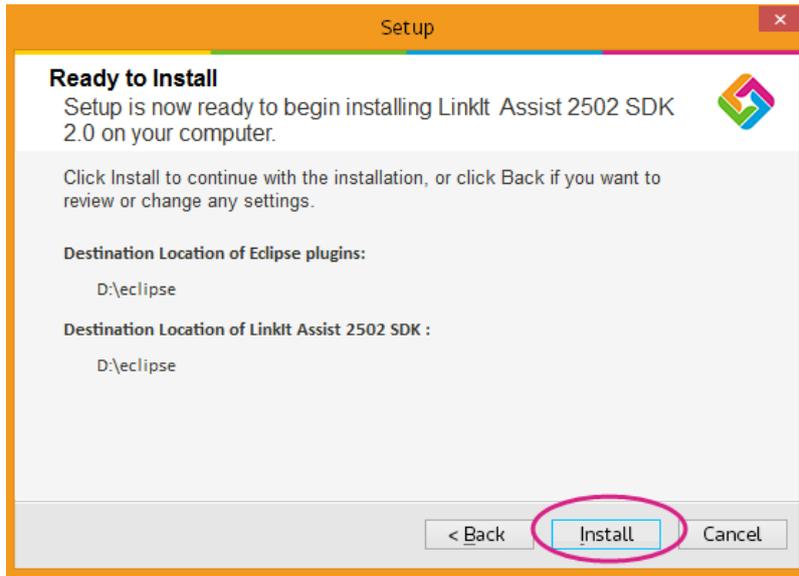


图 10 setup 向导的 Ready to Install 页面

9) 一旦安装了插件，并且您选择了新路径，那么 SDK 将会安装在那里。在完成页面，如图 11 中，确保 Install the MediaTek USB driver (安装联发科 USB 驱动程序) 被打勾，然后单击 Finish (完成)。

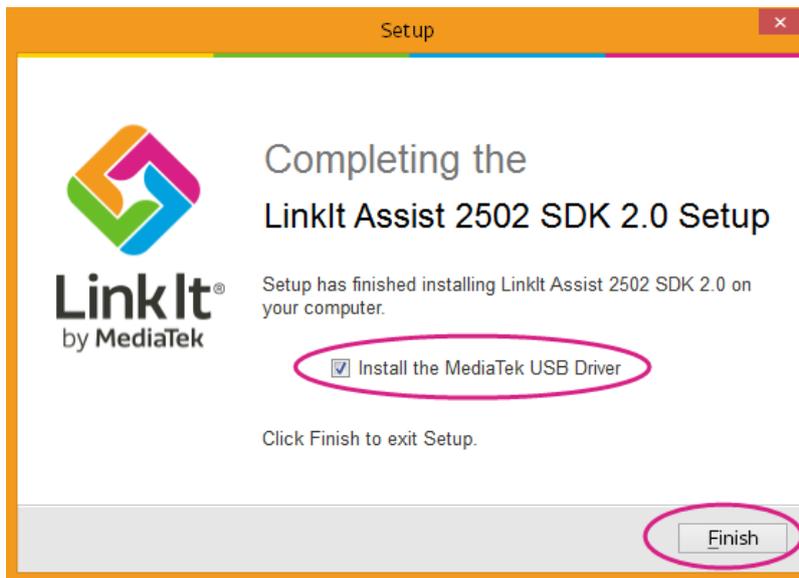


图 11 SDK 安装完成页面

10) SDK 安装已完成，联发科技 USB 驱动程序也已安装。



下一次您更新 SDK 时，不需要再安装联发科技 USB 驱动程序。

2.4. 更新固件

在使用 Eclipse IDE 环境编程应用程序之前，请确保您的开发板固件是对应于您刚安装的 SDK。在 SDK 文件夹里有个 ‘固件更新’ 工具。那么在开始更新固件之前，请确保以下已准备好：

- 1) 如果您的开发板是：
 - a) LinkIt Assist 2502 开发板，请看步骤二。
 - b) LinkIt ONE 开发板，请确认开关切换成大容量存储模式 (mass storage) ，(如果电源是 USB，请确认电源开关已切换成 USB) 如图 12.



图 12 LinkIt ONE 大容量存储模式

- 2) 断开 LinkIt 开发板与您电脑的连线
- 3) 启动 LinkIt Firmware Updater.exe, 其位于 SDK 文件夹下的工具：
{SDK_path}\tools\FirmwareUpdater\
- 4) 在 LinkIt 固件更新程序窗口里选择对应您的开发板平台。然后单击绿色更新按钮，如图 13。

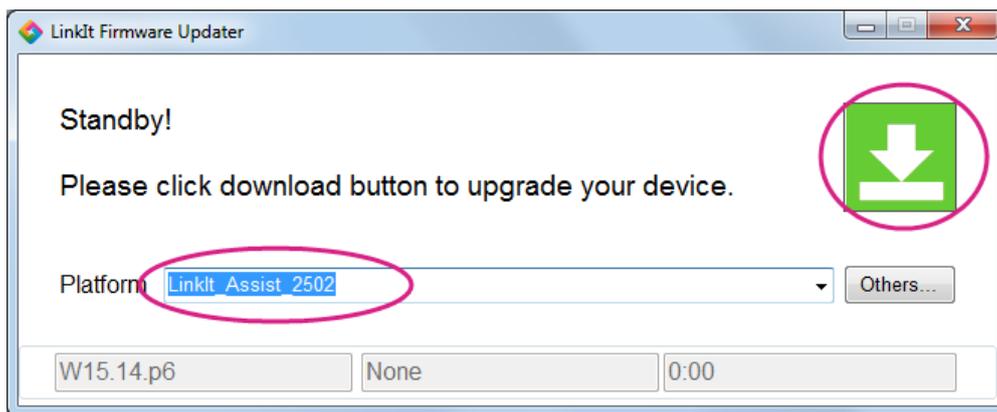


图 13 LinkIt 固件更新开启页面

- 5) 依照**固件更新程序**窗口里的指令执行如图 图 14 .固件正在下载中，这时请不要切断开发板和电脑连接。

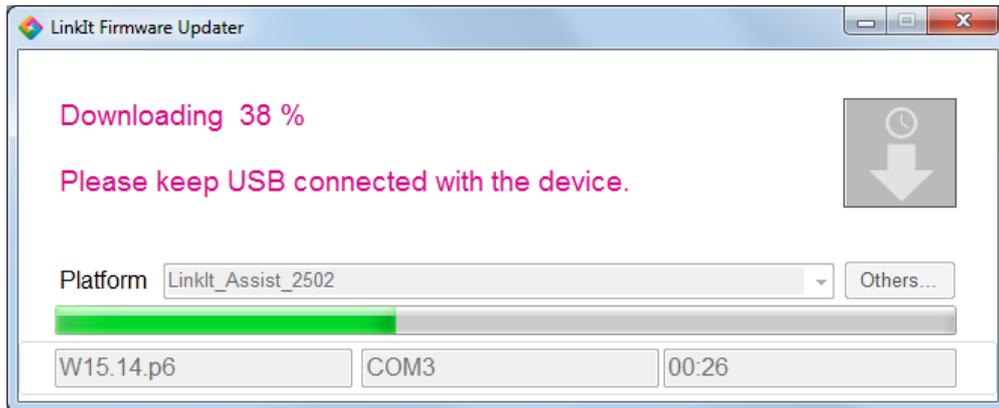


图 14 固件更新中

- 6) 当固件更新完成后，下载完成网页将会确认，如图 15 。

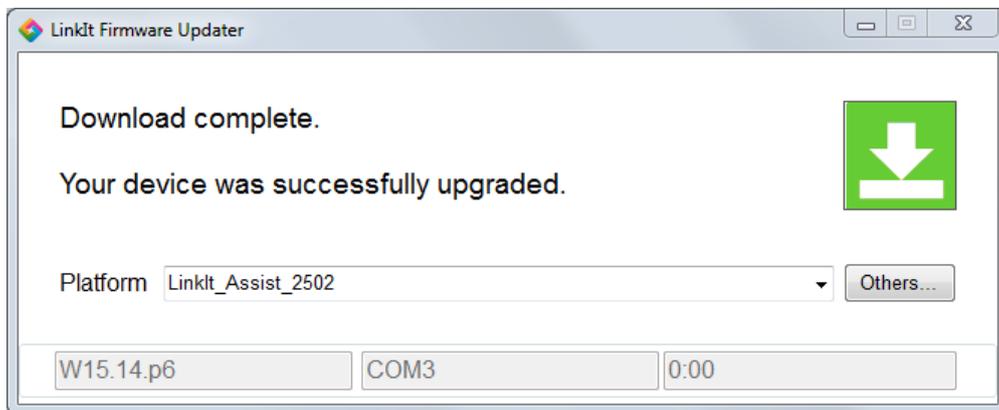


图 15 固件更新下载完成

- 7) 断开 LinkIt Assist 2502 开发板（如果是 LinkIt ONE 开发板，请把它切换回 UART 模式），再重新连接到您的电脑。

您已安装了最新的 SDK 固件，并可进行至 Eclipse IDE 中创建第一个应用。

2.5. 创建您的第一个项目

在本章节，为 LinkIt ONE 或 LinkIt Assist 2502 开发板创建项目和软件开发的步骤是相同的，但结果是不一样的。

- 1) 首先请创建一个新的 LinkIt Assist 2502 应用程序。在 Eclipse IDE 文件菜单，指向**新建**，然后单击**其他**，如图 16，或使用 CTRL+ N 的键盘快捷键。

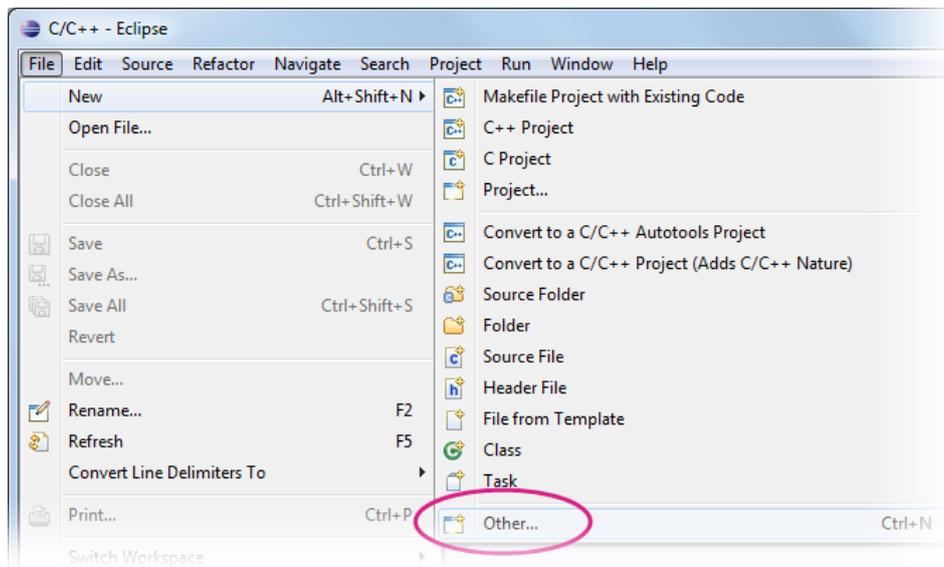


图 16 创建新 LinkIt Assist 2502 应用

- 2) 新的窗口显示如下。展开 LinkIt2.0 文件夹，选择应用 Application (*.vxp)，并单击 Next，如图 17 所示。

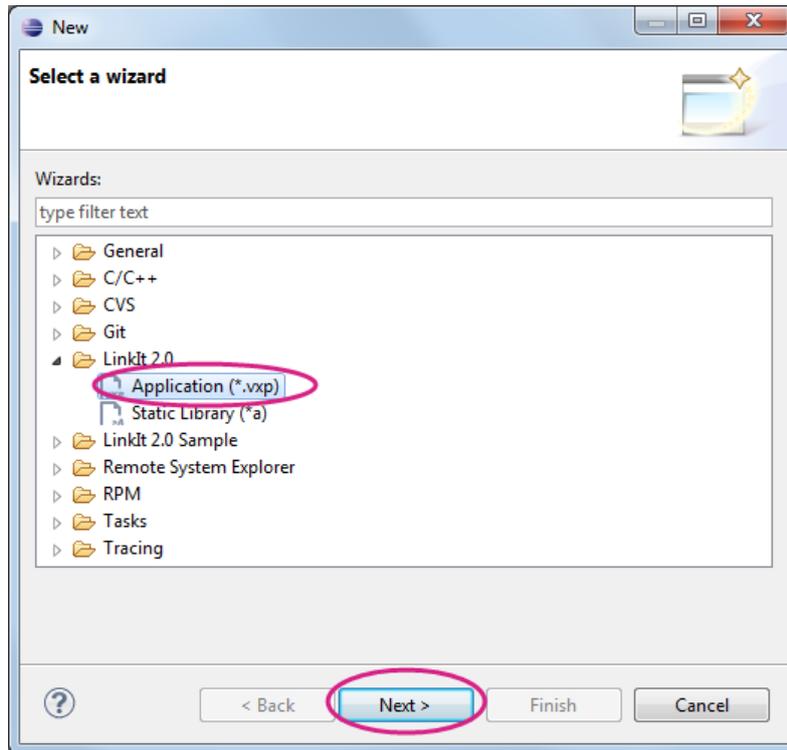


图 17 点选 LinkIt 2.0 应用

- 3) LinkIt Assist 2502 SDK 2.0 向导窗口显示如下。请更改项目名称，并确保**硬件平台**与您的开发板型匹配，如图 18.单击 'Next' 。

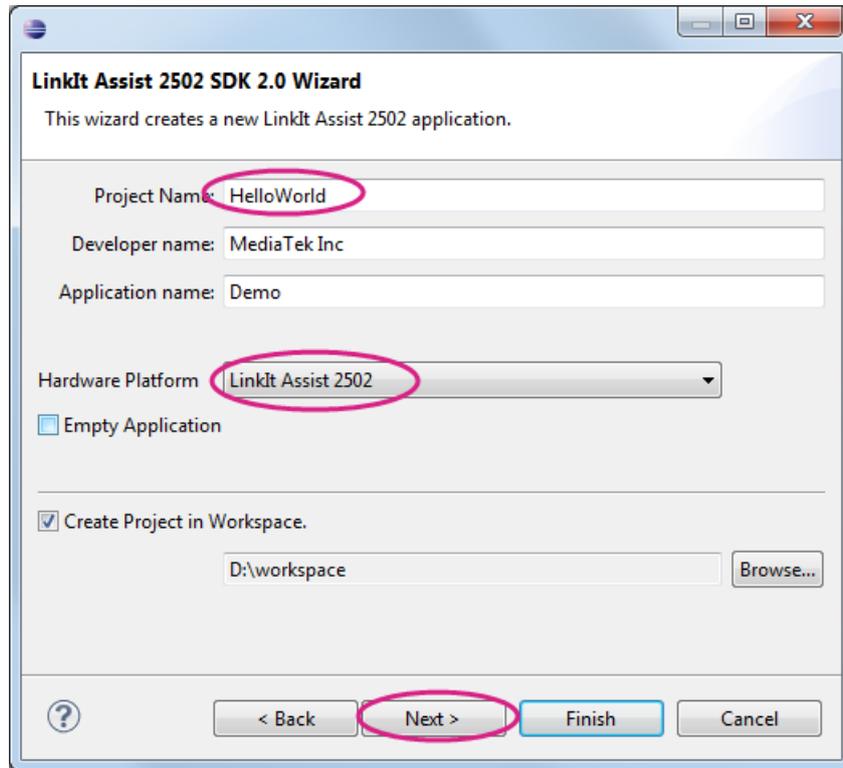


图 18 选择正确的硬件平台

- 4) 函数库选择页面将显示，如图 19，该页面帮助您加头文件库 (header) 至您的项目。您也可以之后使用 #include 命令手动添加头文件。在这第一个项目您可以使用默认库，然后单击 **Finish**。

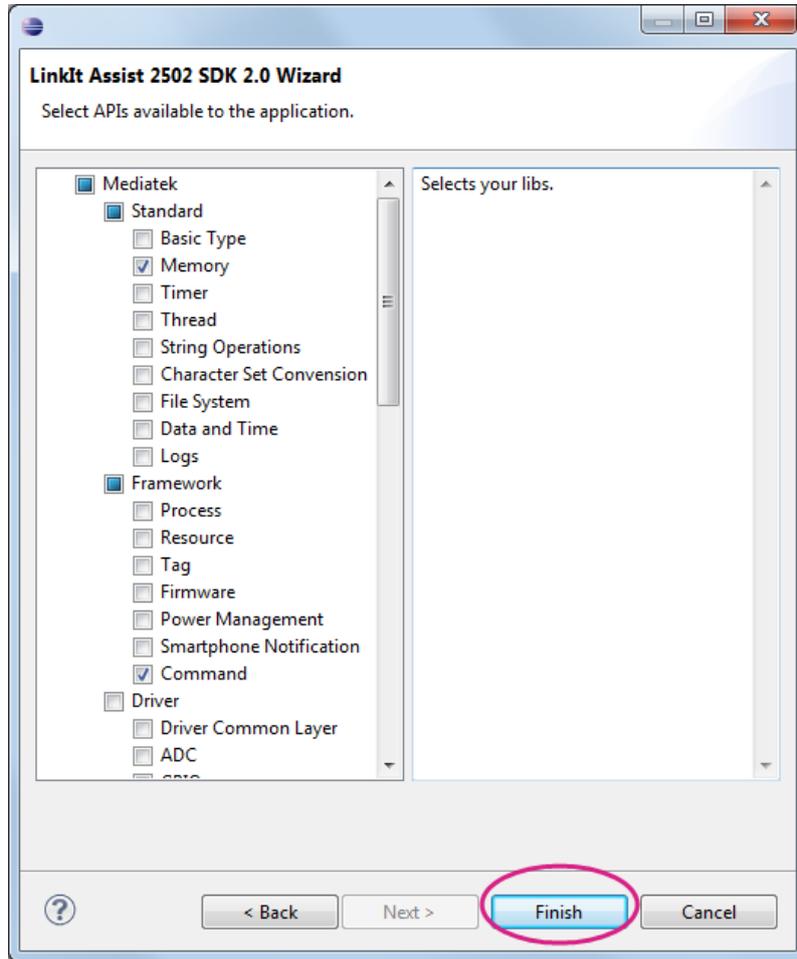


图 19 LinkIt Assist 2502 函数库

现在，您已经创建了第一个 LinkIt Assist 2502 应用程序。从 Project Explorer 窗格中打开您的项目，然后双击 *.c 文件，如图 20。

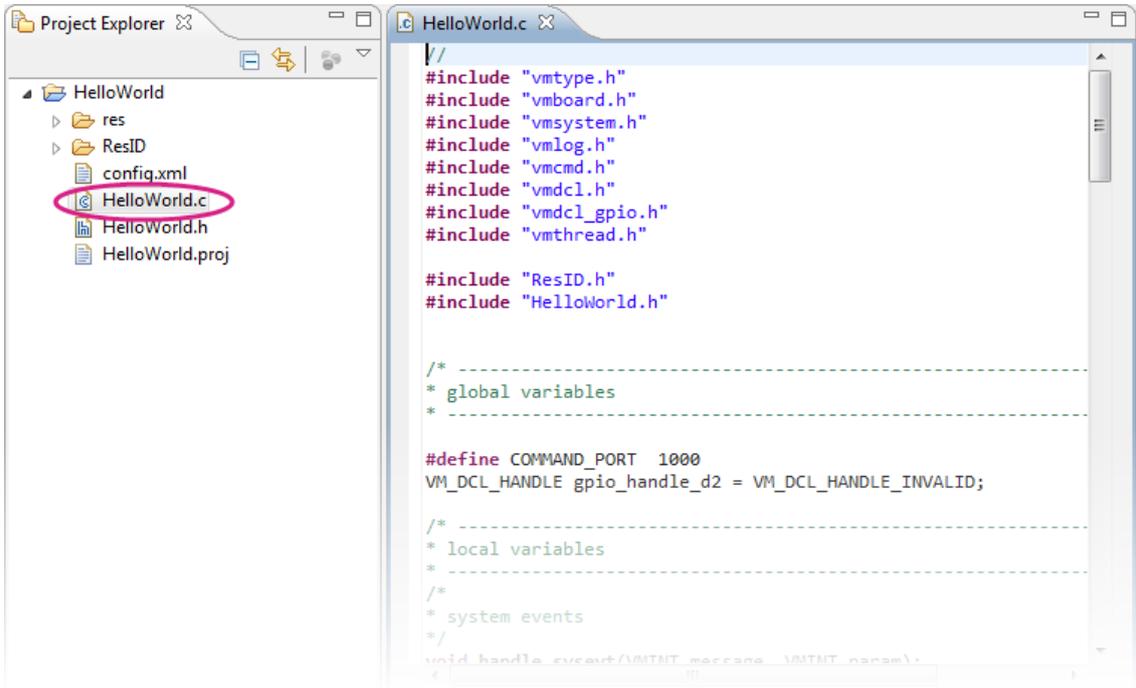


图 20 您的第一个 LinkIt Assist 2502 应用

2.6. 编译与上传硬件

在开始本章节之前请注意一件重要的事情，就是在 Eclipse IDE 的**构建** (Build) 和**调试** (Debug) 命令和相关按钮工具是不能用于 LinkIt Assist 2502 应用的，如图 21。这些按钮工具适合在 PC 上的 C / C++应用运行。请使用插件的工具栏代替来执行必要的开发任务。



图 21 Eclipse IDE 内的 LinkIt 工具栏

有关 LinkIt 工具栏功能的详细信息请参阅章节 2.9，“使用 LinkIt 工具栏”。

创建和上传您的应用程序：

- 1) 确认您的 LinkIt Assist 2502 开发板与您的计算机连接上，然后点击构建应用工具按钮，如图 22。

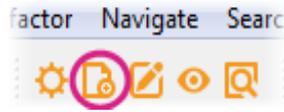


图 22 LinkIt 工具栏的构建应用工具按钮

- 2) Eclipse 的 IDE 将构建并上传应用程序。完成之后，结果将显示在控制台（console）面板上。如果您要查看构建和上传记录，请更改控制台视图至 LinkIt 控制台，如图 23。

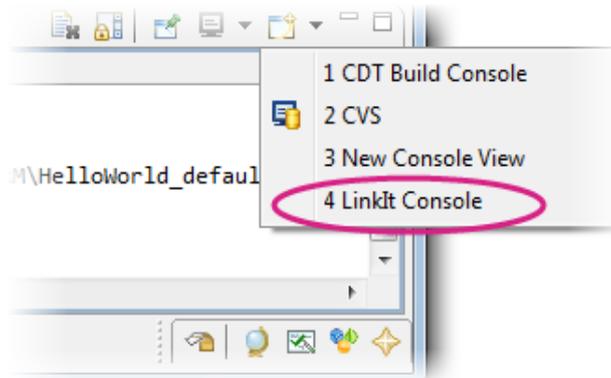


图 23 Eclipse 内的 LinkIt 控制台

2.7. 运行您的项目

当 Eclipse 已构建和安装了您的应用时您将看见：

- LinkIt ONE 开发板: LED 开始闪烁.
- LinkIt Assist 2502 开发板: 显示面板上示出 **HelloWorld** 信息.

恭喜您，您已成功完成您第一个 LinkIt Assist2502 SDK2.0 项目。

You can now explore the example applications provided in the SDK.

2.8. 使用应用范例

该 LinkIt Assist 2502 SDK2.0 提供了可用于进一步了解 API 或作为您的原型基础的范例应用项目。范例应用程序分组于以下文件夹：{SDK_installation_path}\examples\API。

范例应用程序是由功能模块和子模块而命名并按字母顺序排列。例如，DCL_GPIO_Output 范例展示了如何使用 GPIO 为您的硬件作输出功能。

您可以在 Eclipse 直接引入范例如下：

- 1) 在 Eclipse IDE **文件**菜单里，指向**新建**，然后单击**其他**，或使用 CTRL+ N 的键盘快捷键。**新的窗口**将显示。这时展开 **LinkIt Assist 2502 SDK 2.0 Examples** 文件夹，选择一个范例应用 **Example Application (*.vxp)**，并单击 **Next**，如图 24 所示。

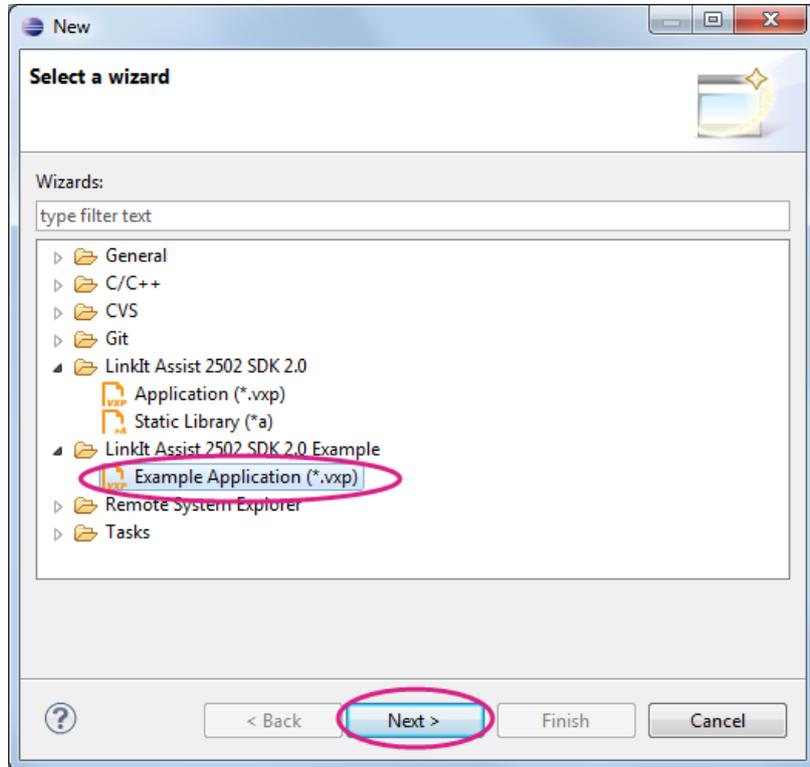


图 24 LinkIt Assist 2502 代码范例

- 2) 在 **LinkIt Assist 2502 SDK 2.0 范例向导**里，给您的项目一个新名称，并选择合适的硬件平台，如图 25 所示。单击 **Next**。

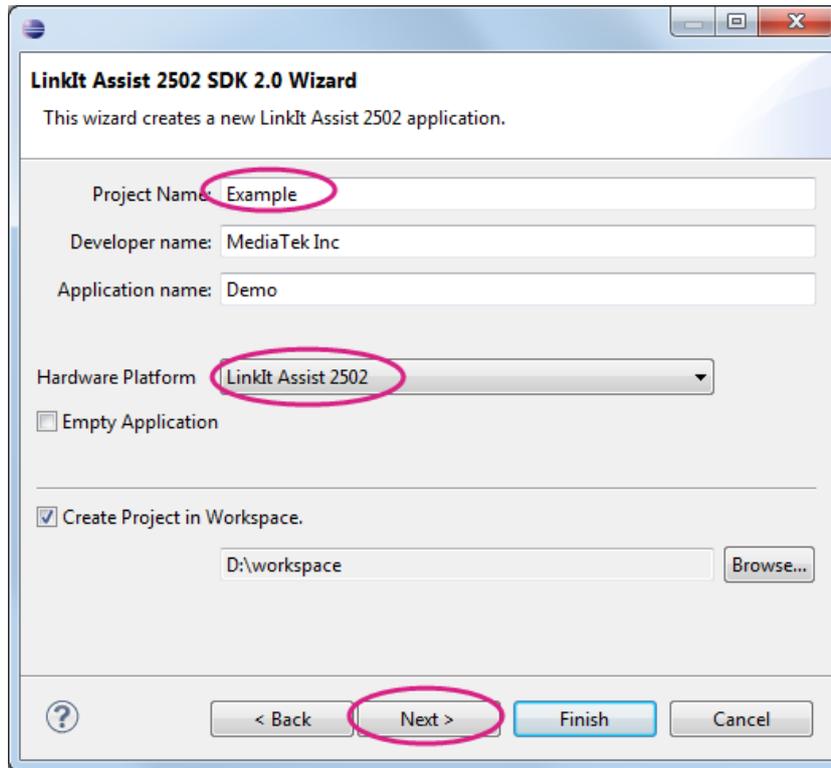


图 25 创建新的 LinkIt Assist 2502 应用

- 3) 您将看见应用程序范例列表（左边）和它们的描述（右边）显示在 **LinkIt Assist 2502 SDK 2.0 范例向导窗口**，如图 26。

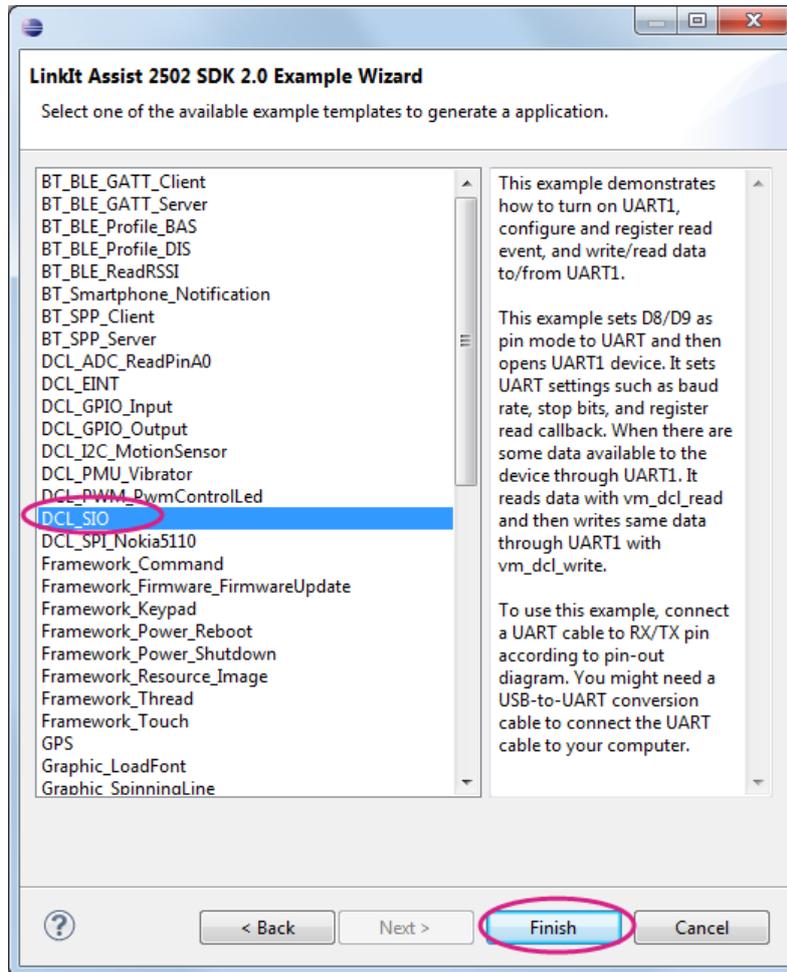


图 26 LinkIt 应用范例

- 4) 选择您要加载到应用程序的范例，然后单击 **Finish**。

Eclipse 会加载所选范例至应用程序中。现在，您可以探索该范例和依您的需求而更新代码。

2.9. 使用 LinkIt 工具栏

LinkIt 工具栏让您访问 LinkIt Assist2502 应用或项目设置。LinkIt 工具栏的功能包含：

- 应用设置 (⚙️)
- 构建应用 (🏗️)
- 资源编辑 (📄)
- 显示工具 (🔍)
- API 参考 (📖)

这些功能将在以下章节详细介绍。

2.9.1. 应用设置

要为一个应用程序设置作定义，首先在 **Project Explorer** 中选择一个项目，然后单击应用程序设置工具按钮，如图 27。



图 27 应用设置工具按钮

接下来，一个项目窗口将显示，如图 28：

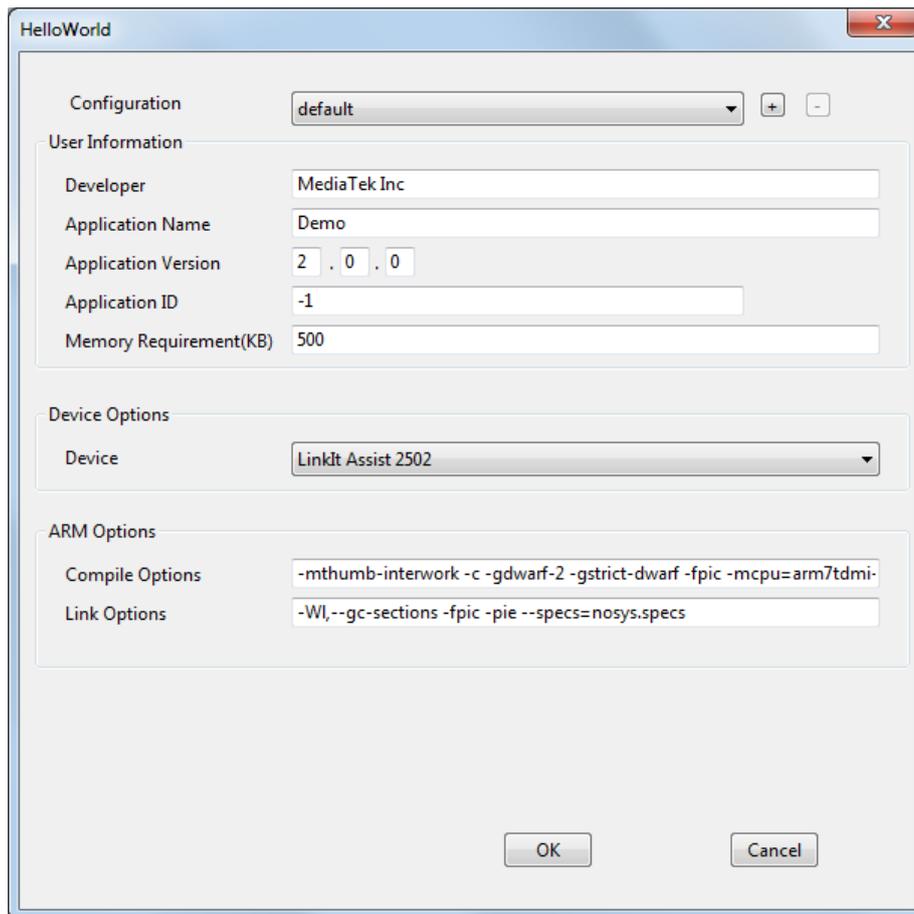


图 28 项目信息

应用设置包含：

- **用户信息**

在**用户信息**下的资料栏是您的应用程序的性能，这些资料栏可以由 API 标签进行查询（请参阅章节 5.6 “标签” 了解更多细节）。**内存要求**是资料栏里最重要的信息，因为它会影响您的应用程序执行。如果它被设置得太高或太低，则应用程序可能无法运行。请参见章节 4.5，“LinkIt Assist 2502 应用存储布局”，以了解有关如何设置此值的详细信息。

- **设备选项**

选择您的硬件开发平台。某些 API 的正确操作取决于设备上所提供的功能，还有硬件的引脚配置。

- **ARM 选项**

这些选项是在构建过程中使用的编译器和连接器选项。您可以在这一栏添加其他选项，但如果您去除掉重要选项，这可能会导致编译或执行错误。

2.9.2. 构建应用

构建应用工具按钮（见图 29）发起构建并上传应用至硬件的过程。这些过程所生成的记录将显示在 **LinkIt 控制台** 面板上。



图 29 构建应用工具按钮

2.9.3. 资源编辑

资源编辑工具按钮（如图 30）启动资源编辑器工具。该工具使音频，视频，二进制，图像和字符串内容嵌入至应用程序中。请参阅章节 4.8，“应用资源”和章节 5.7“资源”，了解如何使用该工具，以及如何使用资源 API 访问它的更多细节。



图 30 资源编辑工具按钮

2.9.4. 显示屏工具

显示屏工具按钮（如图 31）启动显示屏工具（见图 32），它可以让你捕捉应用程序记录，并发送指令至开发板上。



图 31 显示屏工具按钮

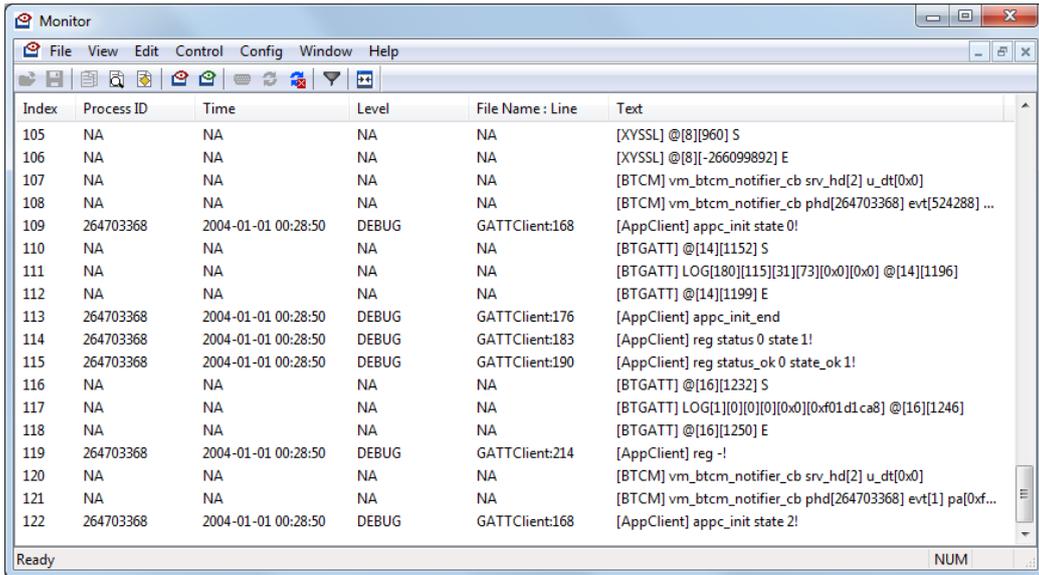


图 32 显示屏工具

如何使用显示工具:

- 1) 选择一个数据库文件，请记住因硬件和固件版本不同的关系，而数据库文件也会不同。这些数据库包含在 SDK 中，您可以在该路径找到它们：

```
{SDK_installation_path}\tools\FirmwareUpdater\firmware\LinkIt_Device\{Hardware}\{Firmware_version}\database.db
```

如果您选择的硬件与数据库文件不匹配，这可能导致连接错误。

- 2) 接下来，选择 **COM** 接口如图 33，这将是联发科技 USB debug 端口，每个计算机的 **COM** 端口号各不相同，选完之后您会在**设备管理器**中看见该调试端口。点击 **OK**。

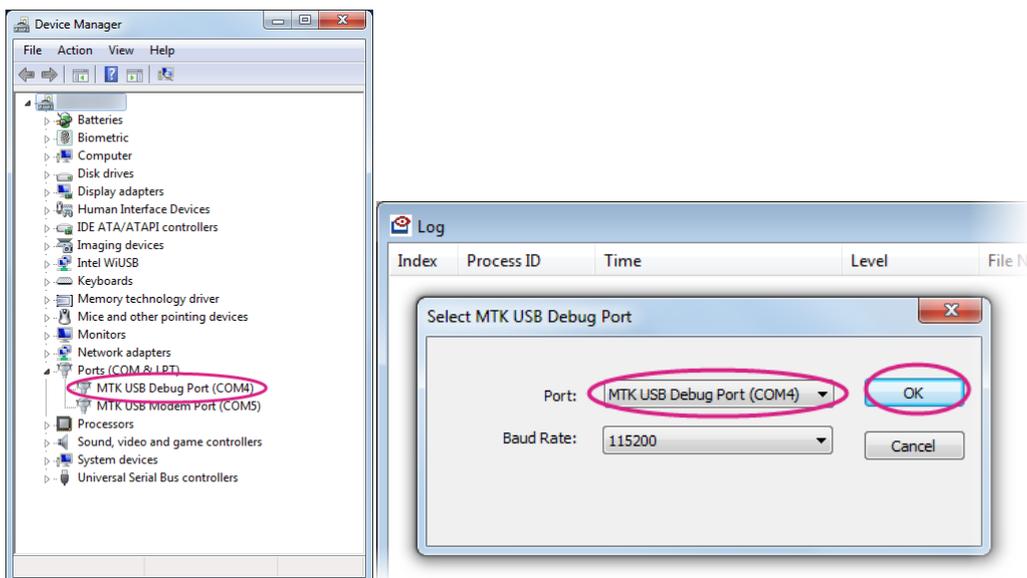


图 33 设备管理器 COM 端口

- 3) 选择 COM 端口之后，请点击连接工具按钮以便与开发板建立连接，如图 34。

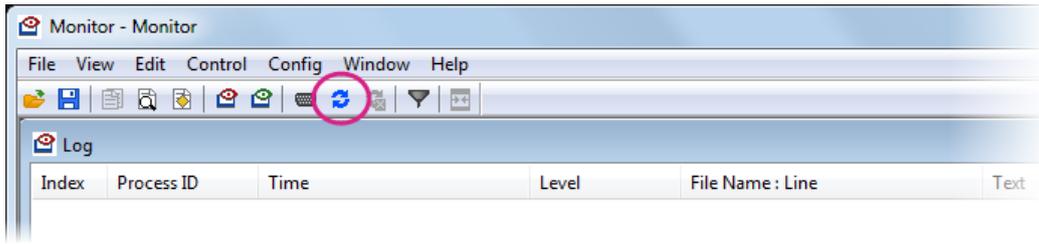


图 34 COM 端口连接

- 4) 一旦开发板连接已建立，您将可以看到记录时间，记录等级，文件名称和记录内容如图 35 中的记录显示。

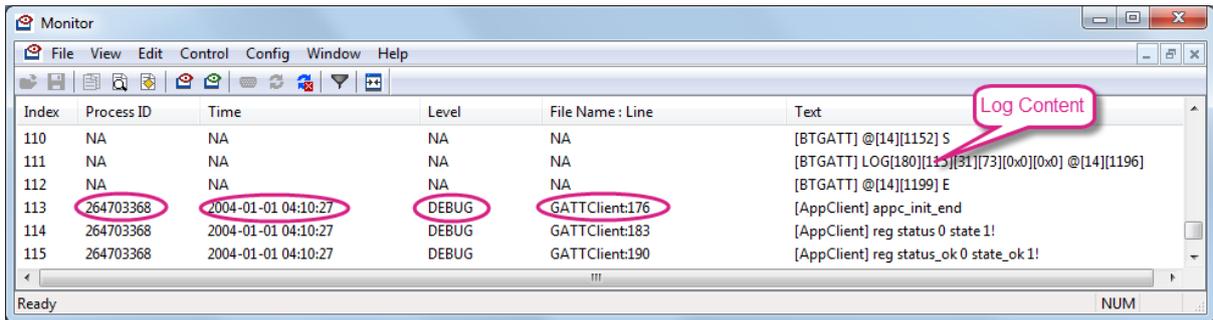


图 35 硬件连接记录

5) 您还可以使用过滤工具按钮创建不同的记录级别，如图 36 所示，单击 **Apply** 调用过滤器设置。

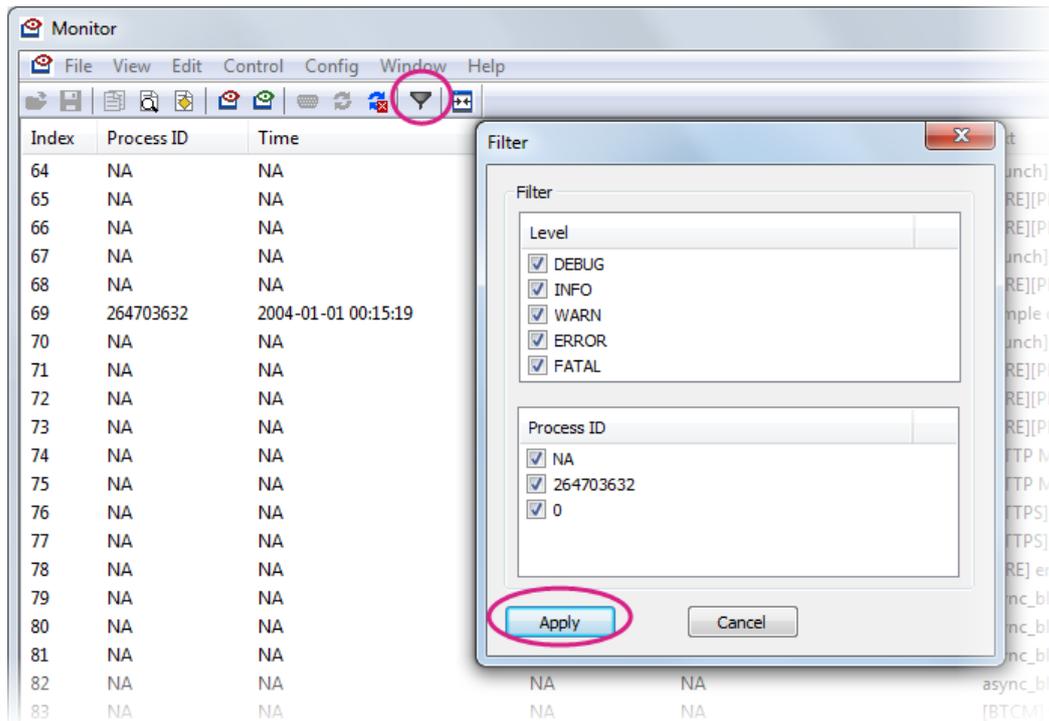


图 36 过滤工具按钮

6) 您还可以点击发送命令按钮工具，来发送命令，如图 37，请参见章节 5.4，“命令”了解命令字符串格式。

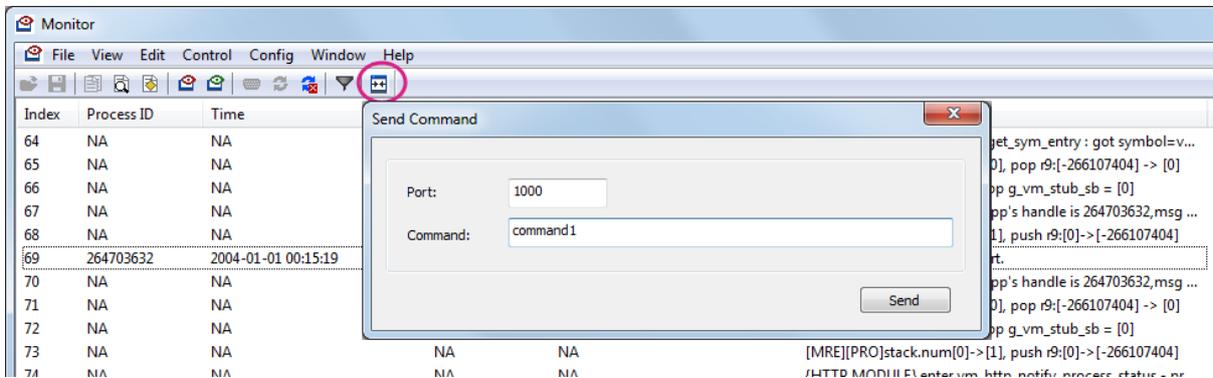


图 37 发送 AT 命令窗口

- 7) 单击切断工具按钮并关闭窗口就完成了。请注意，此时 COM 端口仍然有效的，一直到显示屏工具被切断才无效。

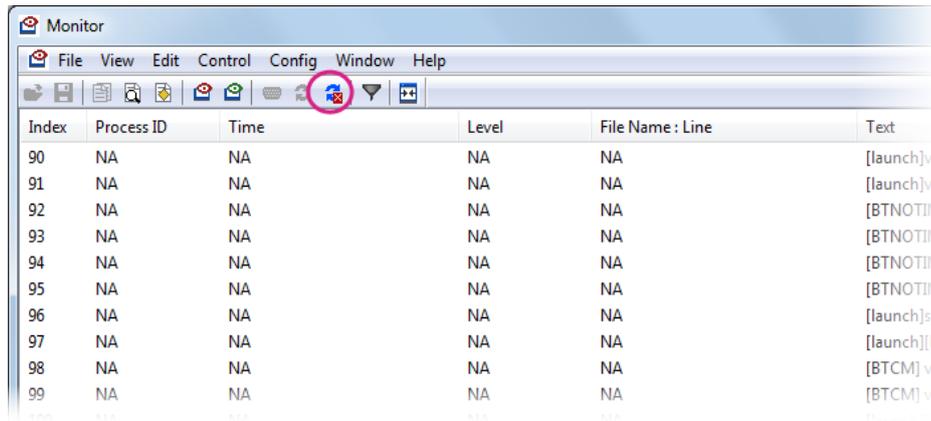


图 38 切断工具按钮

2.9.5. API 参考

该 API 参考工具按钮（见图 39）打开您的默认浏览器，并带您到在线 API 参考文档。



图 39 API 参考工具按钮

3. 故障排除

本章节提供当您在使用 LinkIt Assist 2502 开发板时最常见的问题。

3.1. LinkIt Assist 2502 开发板插入 USB 线之后仍然无法启动

如果您的 LinkIt Assist 2502 开发板启动失败，请确认开发板有接上电池（LinkIt Assist 2502 开发板必须要接上一个锂电池才能启动，开发板包里面包含锂电池）然后用一条 micro USB 线将板与电脑连接，并查看电源 LED 是否亮起。如果 LED 不亮，那代表开发板没有 USB 电源，这时请检查该 USB 线是否连接上合适的电源。请参见图 4 LinkIt Assist 2502 电源 LED 的位置。

如果电源 LED 亮起来而且开发板与电脑是连接上的--由 USB 线，那么您应该在您的 Computer 槽里看见一个移动大量存储设备，如果还是没看见，请与 Seeed Studio 硬件技术支持联系。

如果您看见了大量存储设备，请按位于开发板旁边的电源按键（见图 4 第 11 页）1 秒重启设备，这时大量存储设备将从 Computer 槽里消失，而您将在 DeviceManager 里看见两个 COM 接口。

3.2. VXP 上传失败

如果您的 VXP 上传失败，请确认开发板驱动程序是正确的。参见 3.1 “LinkIt Assist 2502 开发板无法启动” 不同于 LinkIt ONE 开发板接上电池或 USB 线之后会自动启动，LinkIt Assist 2502 开发板要按电源按键约 3 秒钟之后才会进入正常驱动程序（见图 4 第 11 页）若驱动成功，您将在 Device Manager 看见两个 COM 接口，如果没有，请尝试上传开发板的固件，如第 2.4 章节“更新固件”所描述。

如果 COM 接口有显示但是 VXP 还是不能上传，请确认只有一个 LinkIt Assist 2502 设备是与您的电脑连接上的。如果有另一个板子，请将它断开并检查 IDE 里设定的 COM 接口是正确的。如果 VXP 还是不能上传，请将 LinkIt 控制台记录信息复印下来并发个帖子至[联发科技创意实验室论坛](#)以获得支持。

3.3. 扫描 Wi-Fi AP 无法显示

LinkIt Assist 2502 支持的 Wi-Fi 频带是 2.4 GHz，而有一些 Wi-Fi AP 默认配置是 5 GHz，因此请检查并调整该 Wi-Fi AP 的频带。

3.4. 无法使用 GSM 功能

请确认您的电信公司和 SIM 卡提供 GSM（2G）网络。另外，某些 SIM 卡初次使用时需要激活，在这种情况下，请用您的手机激活 SIM 卡然后插入 LinkIt Assist2502 开发板。

3.5. 无法使用 GPRS 功能

某些电信公司要求特定的 APN 设定来访问 GPRS 网络。这种情况下，请与您的电信公司查询正确的 APN 设定并调用 [vm_gsm_gprs_set_customized_apn_info\(\)](#) 来为您的 VXP 软件定义正确的设定。

4. 编程指南

本章介绍驱动程序和使用由 LinkIt Assist 2502 SDK 2.0 提供的工具和编辑运行应用程序的过程与编程概念。

4.1. 驱动程序

虽然联发科技 MT2502 芯片组的驱动程序无法更改，但是了解它的操作方式还是有帮助的：开发板固件更新，软件上传的流程和功能等。

根据不同的硬件配置有两种驱动程序：一个是通过插入 USB 电缆和另一个是通过按下硬件的电源键

Boot loader 起初是系统启动之后才开动的，如果您的 USB 电线连接着您的电脑，那么[MTK USB Port] 将显示在 **Device Manager** 里。固件更新工具是用该接口来上传新的固件（若 COM 接口是上传模式），如图 40 流程[2-1]。不然，如果 bootup 是由插入 USB 电线的话，那系统将进入充电模式，如图 40 流程[2-2]。

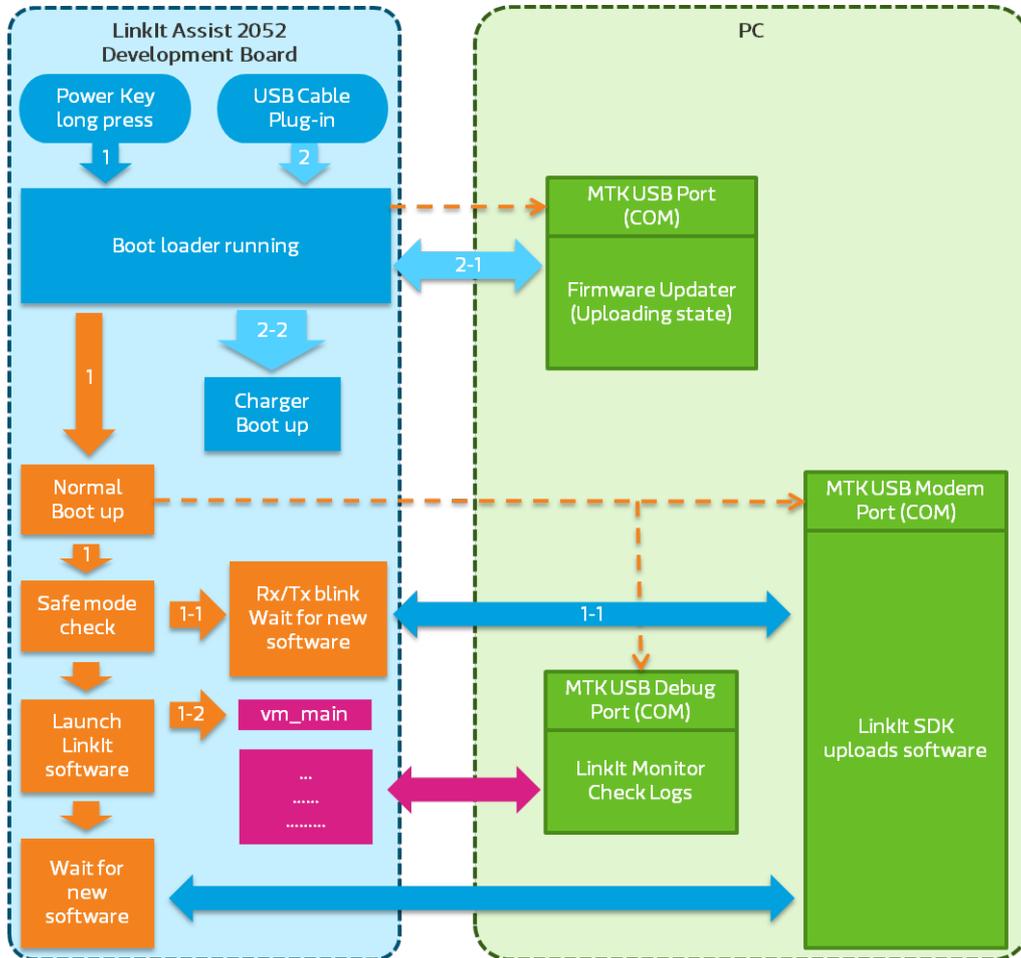


图 40 驱动流程顺序

如果启动是由电源键触发时，系统将进入正常的引导过程，开始系统模块例如驱动，modem 协议，架构等等。两个 COM 接口将出现在 **DeviceManager** 出现，一个为记录，另一个为上传应用。

之后系统会检查是否之前有系统崩溃过。若有，系统将进入安全模式。Rx/Tx LED 将闪烁，系统将等待新应用上传，如流程[1-1]。如果没有系统崩溃过，那软件将被 vm_main() 调用如流程[1-2]，请参见第 4.2 “LinkIt Assist 2502 应用的进入与退出点” 如果软件包含任何串行接口输出那么它将被送到串口让 LinkIt Assist 2502 显示屏读取。

在这种情况下，SDK 可以随时上传一个新的应用。

4.2. LinkIt Assist 2502 应用的进入与退出点

当固件完成驱动过程后，运行环境将启动 VXP 可执行文件。

当使用上传工具 Uploader 上传 LinkIt Assist 2502 应用程序后，在开发板的系统分区中将创建一个新的文件，称为 autostart.txt。该文件的内容是 LinkIt Assist 2502 应用文件名（VXP 文件名）。一旦固件启动完成后，LinkIt API runtime 环境将读取 VXP 文件，并启动 LinkIt Assist 2502 应用程序。然后该 VXP 文件将加载到内存中，并查找 LinkIt Assist 2502 在应用中使用的 API 的实际地址。最后，运行环境将执行您应用定义的 vm_main（）函数。vm_main（）函数包含注册事件处理程序：可以由 LinkIt Assist 2502 应用默认指定的，或者由开发者定义的注册事件处理。系统将调用事件处理程序中定义的对处理或事件。此外，应用程序也可以定期性的获得该程序的操作状态，通过设置一个计时器来相应地处理不同的状态。因此，vm_main（）的默认执行是调用注册系统事件处理程序回调函数。

```
void vm_main(void) {  
    vm_pmng_register_system_event_callback (app_system_event_handler);  
}
```

vm_main（）是 LinkIt Assist 2502 应用程序的主要入口点。它是由 LinkIt Assist 2502 应用执行的，在一般情况下输入事件例如系统事件和回调函数，如果有的话，是由该应用程序注册的。需要注意的是不同于 C 开发环境，LinkIt Assist 2502 应用程序在 vm_main（）返回后仍然不断运行。

LinkIt Assist 2502 应用程序的 vm_exit_app（）提供应用退出的功能。在此函数中，LinkIt Assist 2502 将卸载应用程序，并释放所有占用的资源，并且它通常是最后一个 LinkIt Assist 2502 程序语句，之后没有其他 LinkIt Assist 应用程序可以被执行。

在 vm_main（）函数的初始呼叫期间，LinkIt Assist 2502 将发送 VM_EVENT_CREATE 消息至应用程序。

4.3. 事件驱动编程模式

LinkIt Assist 2502 SDK 2.0 采用的事件驱动编程模式，在当没有进入事件时，能使系统停留在一个低功耗的空闲模式。您编程的事件处理程序是被呼叫相应的系统事件所触发的，而不是通过每个软件和硬件组件遍历检查它们的状态。由于事件驱动编程模式的性质，事件处理例行程序应该不会占用执行器很长的时间。如果不是如此，这将导致一个拥挤的事件处理队列，并导致系统响应缓慢，甚至系统崩溃。因此，我们建议，您应该为系统事件注册回调函数而不是依靠轮询软件和硬件的状态。对于定期活动，例如使用 GPIO 引脚的闪烁 LED，您应该设定一个计时器 API 来注册一个回调函数，而不是制造一个在改变 GPIO 引脚状态之前迫使整个系统休眠一段时间的循环函数。定时器将在指定时间内被调用，并在回调函数里改变 GPIO 引脚状态。

当应用在执行中，应用信息队列将存储信息处理程序。只有将在队列中处理完当前信息的代码完成后，下一个信息才会被恢复处理。因此，如果处理信息时间过长，队列中的其他信息可能无法及时接收响应。这可能不仅影响用户体验，还可能导致错误（exception）。因此我们建议，分解需要长时间处理的操作成多个步骤，并使用定时器来控制这些程序。例如，如果有必要加载大量的资源，您可以创建一个计时器，以它暂停时段，将资源部分化来分段上传。

由于 LinkIt Assist 2502 应用程序是信息驱动的，在执行应用时会需要处理至少两个系统信息，它们是 VM_EVENT_CREATE 和 VM_EVENT_QUIT。在启动应用过程中，运行架构将发送 VM_EVENT_CREATE 信

息给应用程序。首先，在大多数情况下，应用将 VM_EVENT_CREATE 信息处理程序执行资源分配。例如呼叫 `vm_exit_app()` 之后，一个应用程序被杀掉，LinkIt Assist 2502 将会触发系统信息 VM_EVENT_QUIT。而在 VM_EVENT_QUIT 的信息处理程序中，应释放应用程序使用的所有资源。如果仍然还有运行中的线程或计时器访问这些资源，程序将会崩溃。因此，您必须确保在呼叫 `vm_exit_app()` 之后，无任何执行代码事件发生。

应用程序可以接收其它不需要处理的系统信息，如 VM_EVENT_LOW_BATTERY, VM_EVENT_CARD_PLUG_OUT. 详细的系统事件描述请参阅 API 参考。有些模块需要您来处理特定的事件。例如，一旦图形子系统已准备好，VM_EVENT_PAINT 信息将被发送至应用并指示图形模块已准备好使用。

4.4. 线程

如在第 2.1 章节“环境”所描述，应用是在一个多线程环境中执行而应用本身在主线程中运行，并且大多数的 API 仅可用于主线程。该应用可使用 `vm_thread` 函数创建并控制新的和一些限数量的线程。还有基本的线程间通信信息，机制和同步机制。请参阅章节 5.5.5，“线程”了解更多详细信息。

4.5. LinkIt Assist 2502 应用存储布局

本章节介绍应用程序的存储布局，应用项目设置配置的存储要求（请参阅章节 2.9.1，“应用程序设置”），并解释如何估算存储的需求。

应用程序运行时存储器可以分为三个部分：

- 1) 由应用代码本身所占用的存储器。这部分在程序启动将被加载到存储，并会留在存储中，直到它终止。
- 2) 应用运行时分配的单独堆动态存储。
- 3) 函数调用和局部变量的堆栈空间。



注：使用矢量字体需要缓存，大约 100KB 的缓存，位图字体需要 50KB 的缓存，为了让软件有个好的表现，最好将应用最大可用的存储值减掉缓存来确保应有足够的存储来运行字体需求。

应用程序必须配置该程序代码与其堆的大小。但是存储堆栈是由系统固件预先分配，而不可令行配置的。在一个函数里请不要试图定义太大的局部变量因为堆栈存储有限而不能配置。因此建议使用 `vm_malloc()` 在堆上动态分配存储器。否则，当调用函数级别变得过深，可能因由堆栈溢出而导致致命错误。

为应用代码和堆的正确存储大小配置，有两个考虑因素：一个是在系统上可用存储的最大极限，另一个是应用代码和堆所需要的存储。

最大的可用存储将取决于硬件和系统固件。此值可能由不同的 SDK 而不一样。您可以通过 `vm_firmware_get_info()` API 查询。

应用程序所需要的存储大小取决于该应用的行为。应用被加载到存储器中后，该存储布局显示于图 41。

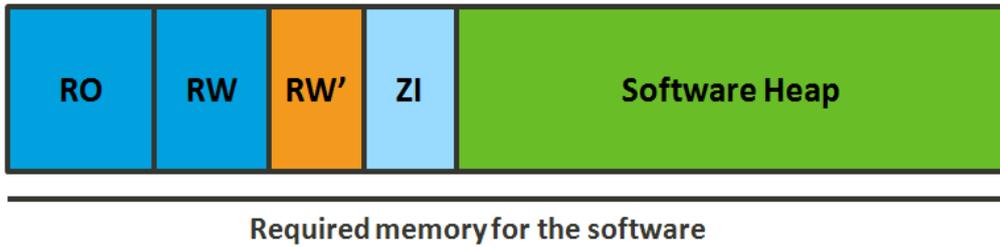


图 41 应用存储布局

只读 (RO)，读写 (RW，RW')，零初始化 (ZI) 区是从执行文件而复制。LinkIt Assist 2502 应用里的软件堆是调用 `vm_malloc ()` 时提供的存储分区。

运行中的应用存储动态分配，可用以下计算公式取得：

能动态分配的存储量 = 应用存储要求量 - 应用占有存储量

应用总存储量要求将不同取决于硬件平台和系统固件。

假设 LinkIt Assist 2502 的总存储量是 800 KB，当 VXP 产生时，宏 (macro) 窗口将显示以下信息：

总 RO 量(代码 + RO 数据)	13616 (13.30KB)	总
RW 量(RW Data + ZI 数据)	1956 (1.91KB)	
总ROM 量(Code + RO 数据 + RW 数据)	13620 (13.30KB)	

那么，可动态分配的存储量应是 800KB-1.91KB-13.30KB=784.79KB

在这种情况下，建议应用存储要求设定在 800KB。请参阅章节 2.9.1 "应用设置 " 了解关于应用存储设定要求。

请注意 ARM 处理器对字节对齐有严格的要求。如果它需要在一个步骤中访问 4 个字节内容，则该内容的起始地址必须位于 4 个字节的边界上。同样地，如果有必要访问 2 个字节的内容在单一的步骤中，则该内容的起始地址必须位于 2 字节的边界上等等，或将发生异常 (exception)。

当发生字节对齐问题时主要与不正确的指针转换有关，例如一个字节的指针转换成一个结构指针。您应该尽量避免这种类型的指针转换。另外，由 `vm_malloc` 返回的指针必须被强制转换并通过 `vm_malloc` 指针返回，而不管其分配的空间的大小，必须正确对齐。

4.6. 记录与命令

一旦您将 HDK 与电脑连接上，您将拥有两个 MTK USB COM 插口，一个是做调试用，另一个是将设备当 Modem 用。Modem 插口可用来上传代码，调试插口可供显示工具显示应用记录和发送命令至应用。章节 5.3 “Log” 描述应用输出记录，还有章节 5.4 “命令”。

请依章节 2.9.4 “显示屏工具” 启动显示工具，若要发送命令，请先按连接 ()，再选择发送命令 () 工具按钮，如图 42。

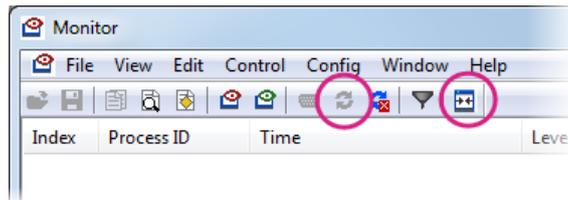


图 42 显示屏里发送命令

发送命令窗口 (Send Command) 对话框里输入 Port 口号 (500 至 65535 有效) 还有命令 (Command)输入您定义的字串 (最大 512 byte)，如图 43 并将它送出。

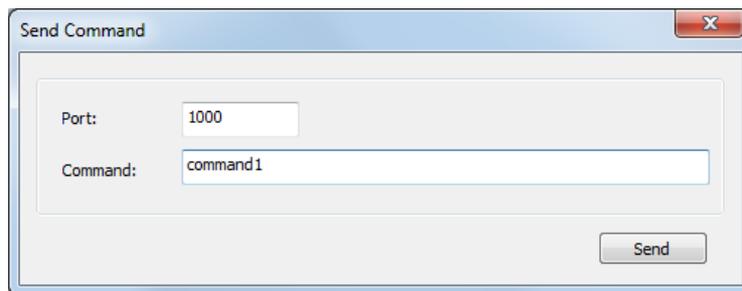


图 43 发送命令

命令格式是 `command_string`。当应用注册命令 API 程序时，才会指定插口号码。请参阅章节 5.4 “命令”。

4.7. 周边硬件与驱动功能

LinkIt Assist 2502 HDKs 有一组分汇引脚来连接多种周边硬件。例如接收开关输入和 I2C 接口的 GPIO 来连接支持 I2C 接口的传感器。请用一般驱动层 (DCL) 如第 5.10 “驱动 API” 描述来连接周边设备。

以下章节说明所支持的硬件界面。



注：虽然 SDK 支持以下硬件界面，但是并非所有开发板都拥有这些界面，请查看开发板规格书详细细节。

4.7.1. GPIO

GPIO 模块是最基本的硬件模块之一。它通常用来传送数字信号，数字信号有两种状态，高与低，并和硬件的电压关连。电压值取决于 I/O 引脚的硬件设置，例如 LinkIt ONE 引脚全是 3.3 V 并与 Arduino 设备兼容，但 LinkIt Assist 2502 引脚是 2.8 V，因此省掉多余的外部电压转换电路。细节请参见 5.10.2 “GPIO” 章节。

有一点要注意的是，在多线程环境中，线程的优先级和 GPIO 引脚信号时序会因系统调度行为而不同。因此我们建议使用相应硬件支持的模块，如 I2C，SPI 和 PWM 模块而不是模拟信号模式与 GPIO 模块。

4.7.2. 模数转换器 (ADC)

模数转换器模块读取所分配的模拟输入引脚的输入电压值，然后将所输入的电压值 0~参考电压至离散整数 0~1023。请注意，输入电压取决于硬件配置。例如，在 LinkIt Assist 2502，该参考电压为 2.8V 但在 LinkIt ONE 因输入模拟引脚与外部电路被分成一半，所以它实际上只有 5.6V 参考电压。

4.7.3. I2C

I2C (内部集成电路) 协议有两个引脚; SDA 和 SCL，它用于连接 I2C 总线与 I2C 设备之间的通信。请参见第 5.10.3 “I2C”

4.7.4. SPI

SPI (串行外部接口) 是一个同步串行外部端口，为用户进行数据通信。在 LinkIt Assist 2502 平台，SPI 需要以下引脚来完成通信：

- MISO (Master In Slave Out): 从仆到主的数据传送
- MOSI (Master Out Slave In): 从主到仆的数据传送
- SCK (Serial Clock): 从主串行时钟输出，用于同步主机和仆机之间的信号时序

许多仆站同时连接到一个主站是可能的。有一个挑选仆站特定的引脚：

- SS (仆选择)：当 SS 的信号为低电压时，主由该引脚选择仆进行沟通。

该 SPI 仅在主模式，并在目前的版本只支持一组 SPI 接口。

4.7.5. 串行 IO (UART)

串行 IO (SIO) 模块支持 UART (通用异步接收器/发送器) 协议。这个模块是能够与电脑或其他设备交换数据，在 COM 端口建立与远程设备相同的波特率以交换数据，SDK 有两种 SIO 设备：一个是硬件 UART 端口，这是通过 DCL SIO 模块可访问的，和另一个是 USB UART 端口，这是模拟当 HDK 被连接到一个计算机中出现的 COM 端口。请注意 USB UART 端口是 LinkIt Assist 2502 SDK 2.0 的特别用途端口，例如，执行上载和记录，因此，它不能用于其他用途。

4.7.6. PWM

PWM (脉冲宽度调制) 为写指定的引脚而转换离散数字信号至连续模拟信号。该应用可以使用 PWM 以不同亮度来点亮 LED，或控制电机。不同的硬件支持不同数目的 PWM 引脚，例如 LinkIt ONE 和 LinkIt Assist 2502 开发板配有 2 套 PWM 引脚，但是 LinkIt Assist 2502 的其中一个 PWM 引脚是用于控制显示模块的背光源亮度。

4.7.7. EINT

外部中断 (EINT) 模块允许应用处理由电压电平或电压变化而触发的事件。有一点要注意的是，在 SDK，中断处理程序是在主线程处理。在硬件时，当一个中断被触发，中断服务处理程序会在系统中创建一个事件至主线程，然后应用回调将在主线程调用。因此，应用的回调处理不会中断彼此的执行，并且总是在一个序列中执行的。

4.7.8. PMU

功率管理单元 (PMU) 允许应用控制某些 GPIO 引脚和系统的某些部分功率，例如 VIBR LDO 输出，它连接到 LinkIt Assist 2502 的振动器。因此，很明显地，VIBR LDO 在 LinkIt ONE 板无法用，因为它没有振动器而因此也不会出现在分汇引脚里。

4.8. 应用资源

该 SDK 允许开发者嵌入资源，如图像，字符串和文件 (二进制大对象如音频) 到应用程序的可执行文件。对于字符串，资源编辑能保存不同语言的字符串，这让能切换语言的软件很方便。

这些外部资源在资源编辑器中创建并纳入在应用构建过程中的应用 (VXP 文件)，在 Runtime 环境时，资源函数搜索资源 ID 的地址，上传资源的内容到 RAM 并向应用报告 RAM 的地址。

请参阅章节 5.7，“资源”了解如何复取嵌入资源的资源 API。一旦我们启动资源编辑工具如第 2.9.3 章节“资源编辑”，它能够添加不同的资源至应用的项目文件。

4.8.1. 添加文件和图影资源

音频，图像和独断的二进制文件资源，可被嵌入通过资源类型面板选择，如图 44。

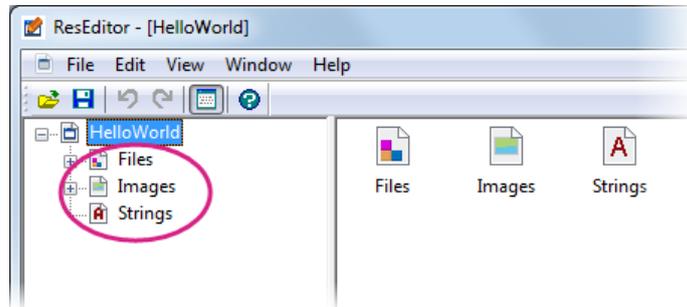


图 44 资源种类

要添加图像，请先单击图像，再选 **Default**，如图 45。

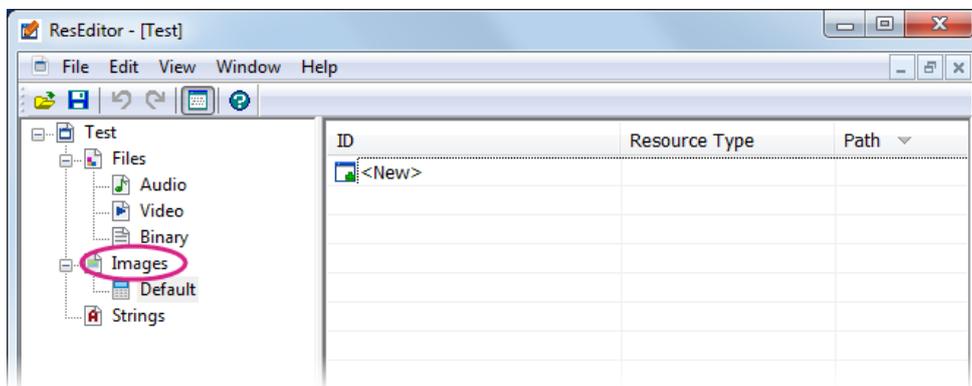


图 45 选择图像资源种类

然后在<NEW> ID 双击指定的标识符至新的资源项目，如图 46。该标识符将被资源 API 来加载资源。

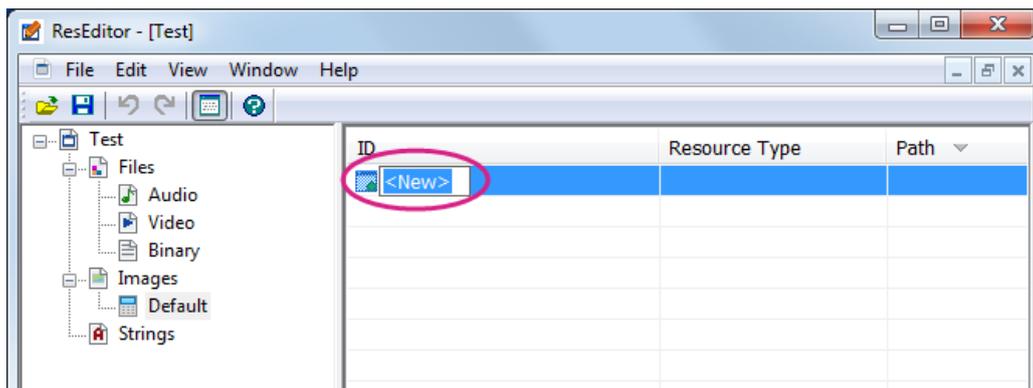


图 46 图像资源范例

注：该资源标识符有命名限制：



- ID 长度不能超过 64 字符
- 第一个字符不可是 0-9
- 字符必须是 A - Z, a - z, 0 - 9, or ‘_’

在这个例子中，我们将其命名图像资源为 APP_IMG_1。电击 Resource Type 下面并选择 **Normal** 和电击 **Browse** 选您要嵌入应用的图像文件，如图 47。

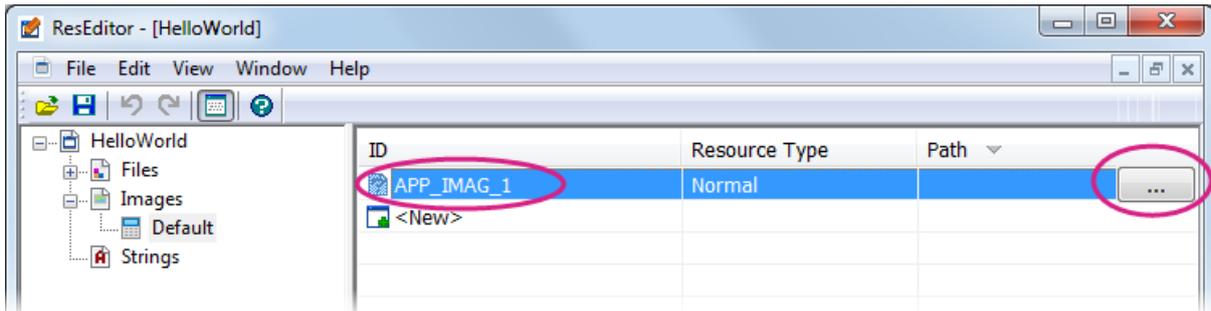


图 47 点选图像文件

最后，点击“保存”将设置保存到项目中，如图 48。当你构建应用后，您选择的文件将会被嵌入到应用程序，并且您可以用资源 API 来访问它。请参阅章节 5.7，“资源”了解更多。

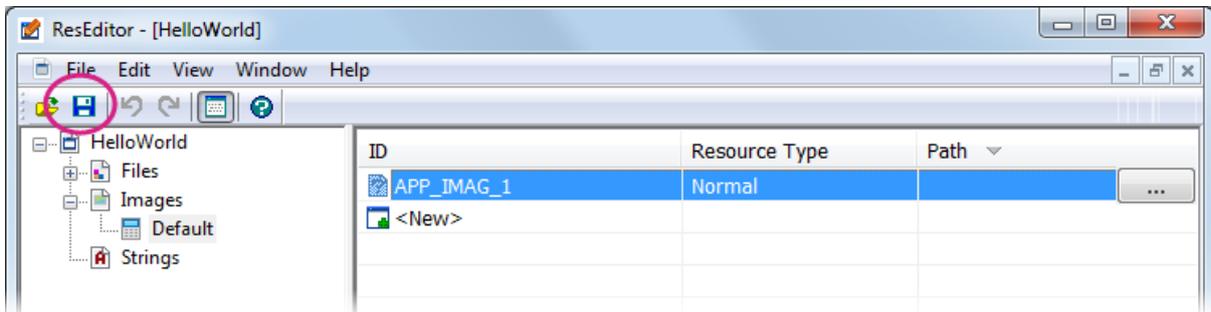


图 48 保存 (Save) 工具按钮

音频和二进制文件可以类似的方法作嵌入。

4.8.2. 添加字符串资源

字符串可以文件相同的方式添加，但由于多国语言支持，首先应定义我们需要的语言并电击 **Strings** 文件夹如图 49。

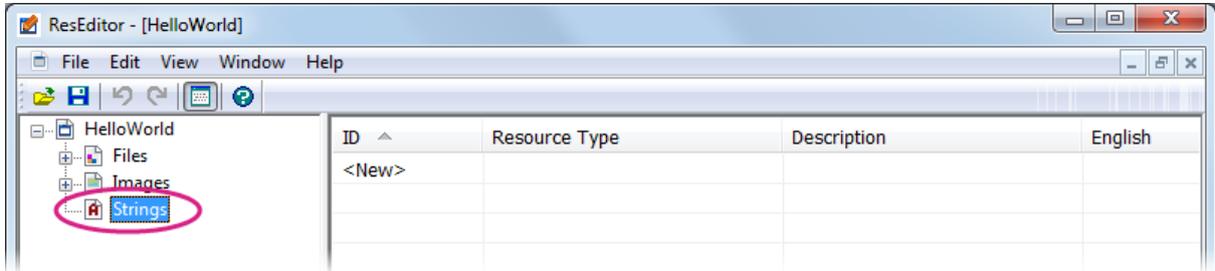


图 49 选择字符串资源种类

右键单击 **New**，并选择语言如图 50。

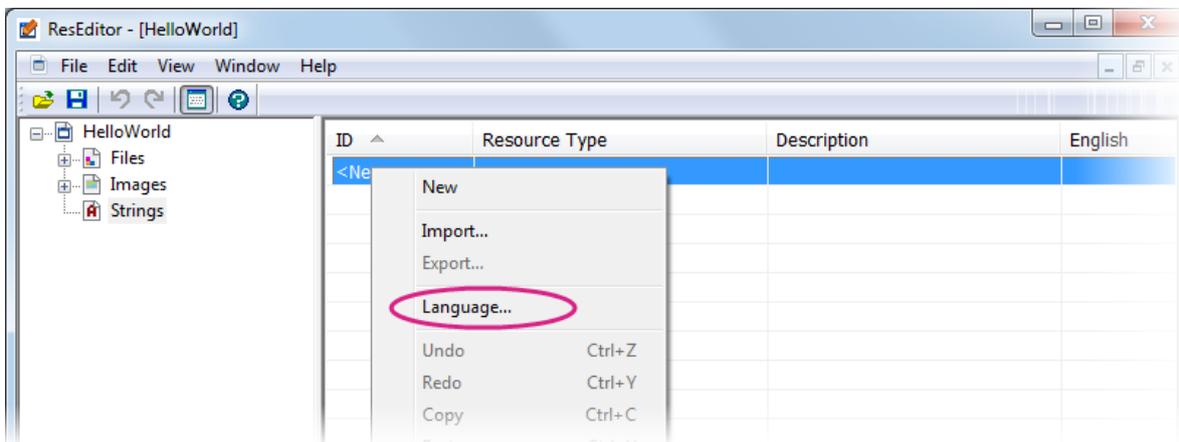


图 50 选择字符串资源语言

选择您想加在资源编辑器的语言，如图 51。语言 ID 被资源 API 用于定义不同的语言。单击 **OK**

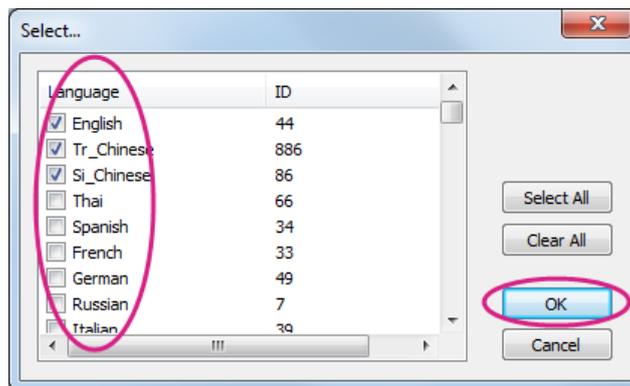


图 51 选加入字符串包含的语言

接下来为每种语言编辑文本，如图 52。

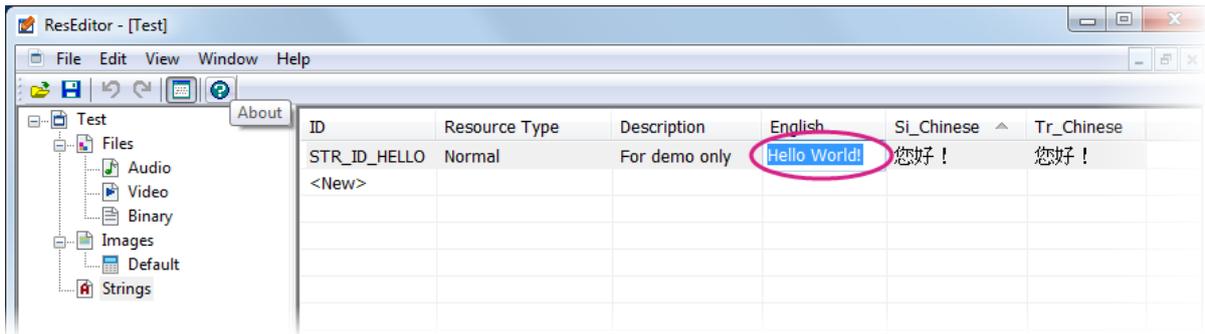


图 52 编辑本文

您也可以从右键中选择出口或进口来导出和导入字符串表至 Excel 文件，如图 53。

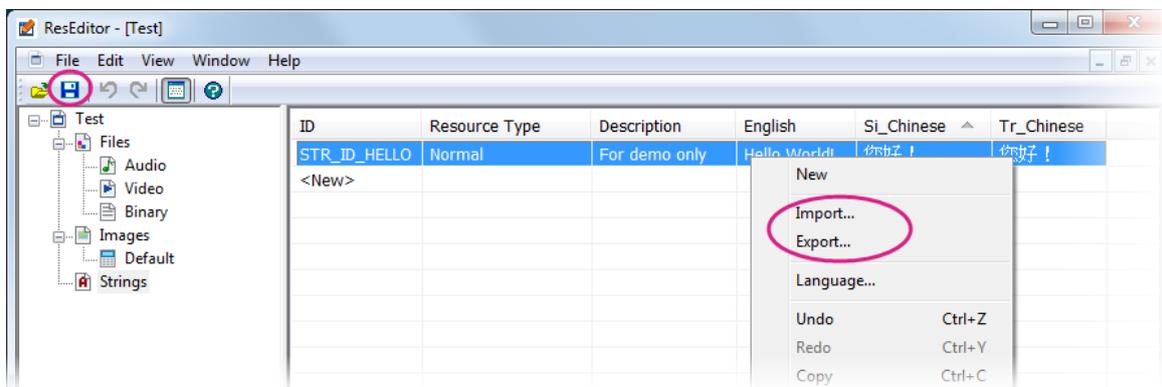


图 53 Import/Export 字符串资源

点击“Save”将最终字符串表保存至项目中。参见第 5.7 “资源”了解如何上传这些资源到软件上。

4.9. 电源管理

SDK 不提供一个明确的电源管理控制器，因为应用采取事件驱动编程方法，使系统在没有事件处理运行状况之下会自动进入休眠模式，而不是让应用处理，这有助于系统保存功率。

一些主要的硬件连接模块可以明确地启用和禁用，以确保硬件完全关机，以进一步节省功耗。这些模块提供单独的 API 来启用和禁用其硬件组件。当该装置启动时，大部分的硬件模块都默认禁用式，应用必须明确启动这些硬件模块才能访问它们的功能。

表 4 表 4 显示需要明确启用和关闭的硬件组件：

模块	启动 API	关闭 API
蓝牙	vm_bt_cm_switch_on	vm_bt_cm_switch_off
WiFi	vm_wlan_mode_set with VM_WLAN_MODE_STA	vm_wlan_mode_set with VM_WLAN_MODE_OFF
GSM/GPRS	vm_gsm_switch_mode	vm_gsm_switch_mode
GPS	vm_gps_open	vm_gps_close
LCM	下一版	下一版

表 4 启用与关闭硬件组件功能

4.10. 移植 Arduino sketch 和 驱动程序

在 LinkIt Assist 2502 上执行 Arduino sketch 是可行的，您只要移植它们和一些外部设备驱动就行了。移植机制于 LinkIt ONE 开发平台类似，其 LinkIt runtime 环境中提供一个移植层来执行 Arduino sketch 和相关驱动程序。

唯一不同的是该移植层在 LinkIt ONE 开发平台是内置的，而 LinkIt Assist 2502 则是提供一个范例项目当作移植层，让您导入至 SDK 2.0 里并与 Arduino sketch 集合。

该移植范例仅支持核心 Arduino 功能包含：

- Arduino [pinMode](#), [delay](#) and [digitalWrite](#)。
- Serial1 对应硬件发送与接收端口，但是 Serial1 已被 USB UART 端口显示屏工具占用。
- [Wire \(I2C\) 函数](#)。
- [SPI 函数](#)。

请依照以下步骤移植您的 sketch:

- 1) 导入 LinkIt Assist 2502 SDK 里文件夹 custom\Arduino 的移植范例：
 - a) 在 eclipse 的 **File** 菜单电击**导入 (Import)**。在 **Select** 窗口打开 **General** 文件夹并电击 **ExistingProjects into Workspace**, 如图 54。

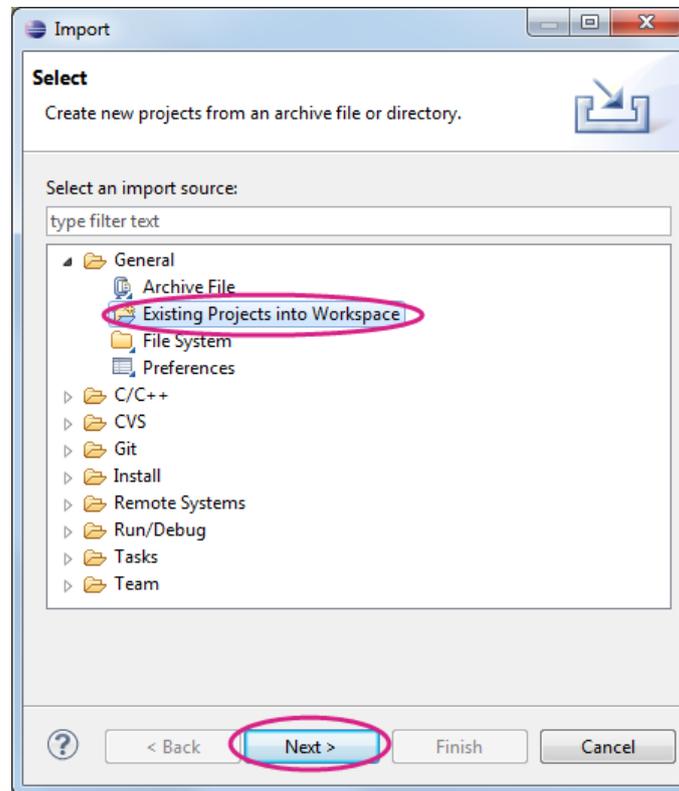


图 54 导入源件

在 **Import Projects** 窗口，**Select root directory** 旁电击 **Browse** 并在 **LinkIt Assist SDK** 文件夹里找到 **custom\Arduino** 文件夹。确认 **Copy projects into workspace** 复选框打钩并电击 **Finish**，如

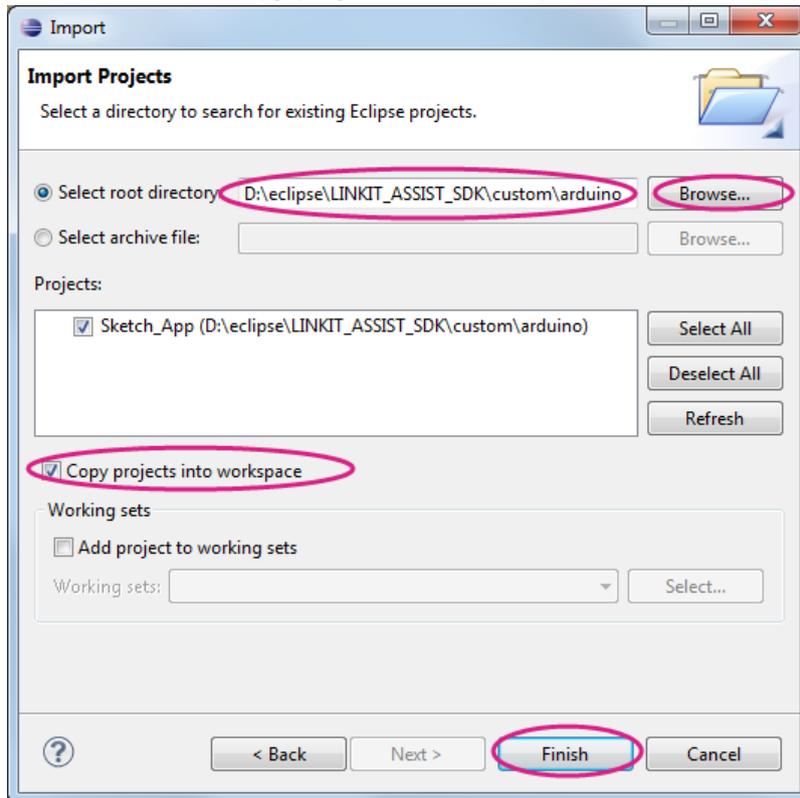


图 55 设定 root directory custom/arduino

2) 导入您的 Arduino sketch 如下：

a) 在 **Project explorer** 里电击 **Sketch_App** 文件夹并打开 **Sketch_App.cpp**

- b) 复制与粘贴您的 Arduino sketch 至 **Sketch_App.cpp**。您必须在 .cpp 顶部插入 #include <Arduino.h> 加 Arduino 放头文件，因为 eclipse IDE 不会自动加此放头文件，Arduino IDE 是会的。以下范例显示 Ardino Blink 代码复制与粘贴至 .cpp 文件。您也可以删除 **Sketch_App.cpp** 而用您自己的 Sketch 文件，只要将它字尾从 *.ino 改成 *.cpp 就行了。

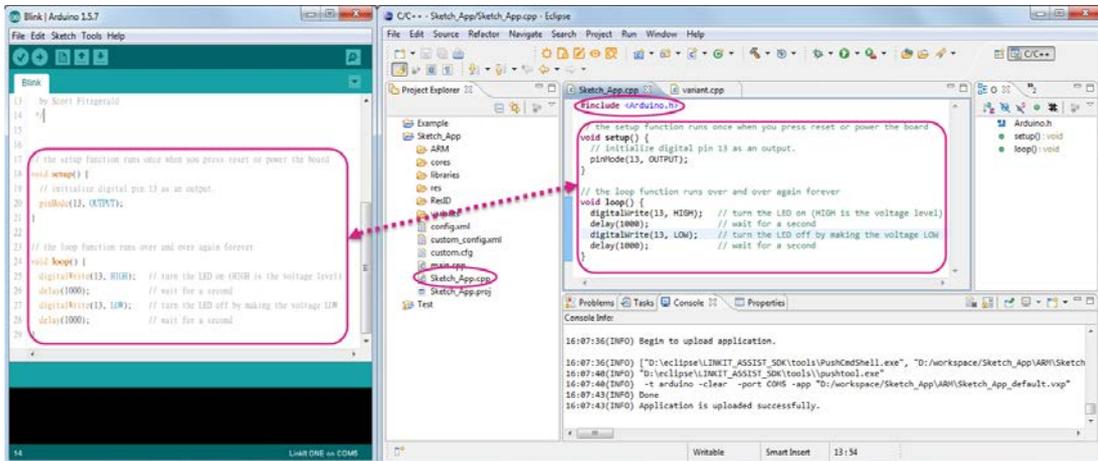


图 56 导入和粘贴 Arduino sketch

- c) 您可能需要修改从 LinkIt ONE 来的引脚配置和对应，请参见 variants\linkit_one\variant.cpp 文件，该文件里面包含引脚对应定义，然后更新和导入您的 sketch 代码。改好的文件将对应 Arduino 和 LinkIt Assist 2502 的 DCL 引脚。
 - d) 电击 Build Application 工具按钮并上传您的 VXP 文件。
- 3) 如果您的 Sketch 需要附带 Arduino 外部驱动，请依以下步骤导入：
- a) 打开 **Windows explorer**，复制 Arduino 驱动库至 **Sketch_App** 文件夹 \workspace\Sketch_App. 然后在 **Project explorer** 里右键电击 **Sketch_App** 电击 **Refresh** 来更新项目，如图 57，图内范例导入一个 ADXL345 驱动程序。

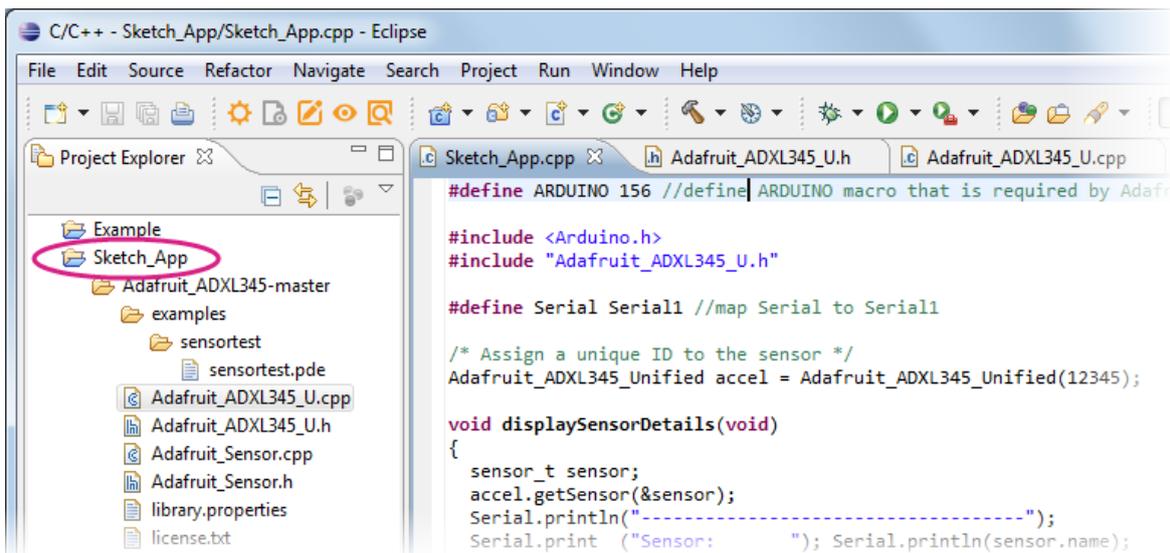


图 57 导入外部驱动程序

- b) 根据不同的驱动程序，您需要加或修改 #define 规定。ADXL345 范例需要一个 Arduino macro 在 LinkItAssist 2502 没有的，因此您必须手动将它加入 Sketch_App.cpp 里，如图 58。

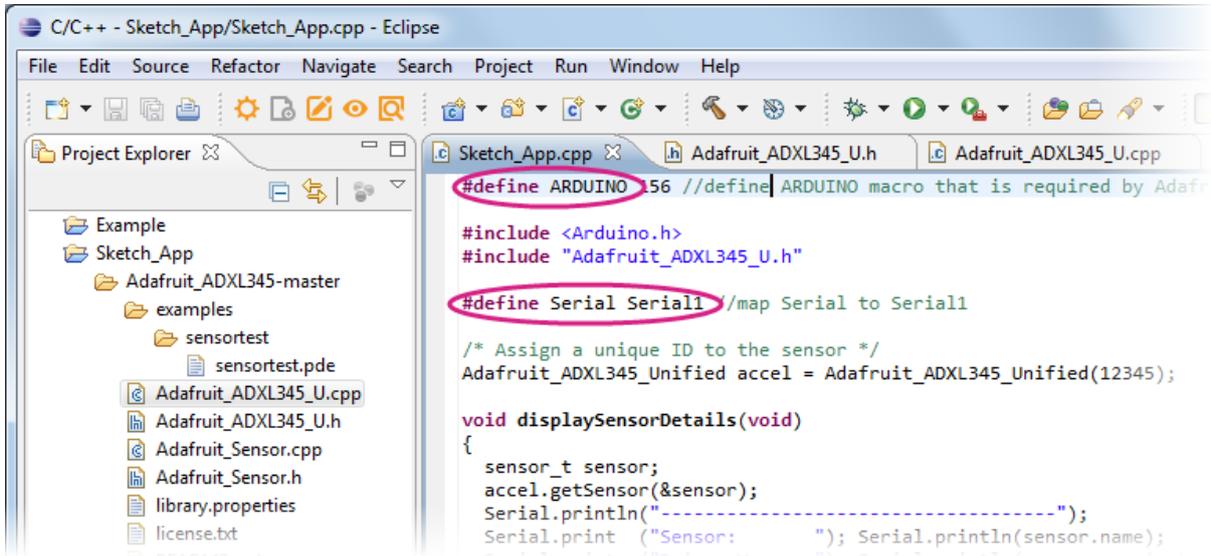


图 58 加入需要的 #define statement

注：Arduino sketch 是在 vm_main 线程里创建的线而执行的，细节请参见 Sketch_App 文件夹里的 main.cpp



另外，因为 Arduino sketch 不是主线程，您不该在那里调用 LinkIt Assist 2502 API，您应该根据软件的要求贯彻落实 Arduino 的线程和主线程之间的同步机制。

5. API 指南

本章介绍 LinkIt Assist2502 API，以各个模块呈现，它简单地覆盖每个模块的功能和限制。有关详细的 API 描述，请参考联发科技创意实验室网站上的 [API 参考](#)。

API 函数是由 C 语言出口的头文件并有个 VM 词头，例如：vm_malloc。类似地，模块名称也前有 VM 字首，例如：vm_audio_get_volume。

5.1. 基本类型

LinkIt Assist 2502 SDK2.0 运行在基于 ARM 的硬件，因此潜在的机器字是低字节序，而默认整数位宽度为 32 位。

大部分一般类型已定义并且也有 VM 词头，例如：VMINT，VMCHAR 和 VMSTR。其他包含 char，short，int，long，和 long long。Float 和 double 类型也在软件浮动指针计算库中支持，空值终止字符串被定义为 VMSTR 和 VMWSTR。



注：浮点数计算通常是在软件中实现，其性能将不如如果是在基于硬件中实现。因此建议您尽量避免浮点数运算。

5.2. 返回值

许多 LinkIt Assist 2502 函数返回 VM_RESULT 作为返回类型。

为了您的方便，LinkIt Assist 2502 SDK2.0 提供了 2 个宏：VM_IS_SUCCEEDED 和 VM_IS_FAILED，它可以用来确定一个函数调用成功还是失败。

当您看见一个函数返回类型 VM_RESULT，你可以使用 VM_IS_SUCCEEDED(x) 或 VM_IS_FAILED(x)，x 是函数的返回值。

请记住，虽然一般成功值，如 VM_SUCCESS 和 VM_OK 已定义，您应该避免直接比较这些成功值。这是因为某些 API 可能会返回代表不同的成功场景而产生不同的值。因此，与其编程以下正确逻辑功能的代码：

```
if(VM_OK == vm_api_function())
    do_success_case();
```

您应该这样编程：

```
if(VM_IS_SUCCEEDED(vm_api_function()))
    do_success_case();
```



注：这仅适用于 VM_RESULT 类型的返回值。对于其他类型的返回，如 handles 或存储 pointers，请参阅关于错误处理的 API 参考。

对于 VM_RESULT 返回值，详细的错误代码和其它的成功代码分别定义为每个模块中的枚举值。请参考 API 文档，如果你需要处理特定的错误代码或成功的方案。

5.3. Log

LinkIt Assist 2502 SDK2.0 提供一个记录模块并将应用所产生的调试和错误信息发送至该 SDK 的显示工具（关于如何使用显示工具的更多信息，请参阅第 2.9.4 章节，“显示屏工具”调试和错误信息的记录模块）。

还有，针对不同等级的错误严重程度，例如 FATAL, ERROR, WARN, INFO 和 DEBUG 功能，SDK 提供不同的功能。错误严重程度等级从最高到最低的顺序如：FATAL > ERROR > WARN > INFO > DEBUG，您还能在显示工具中控制严重筛选。

您可以使用以下 API 来产生应用记录：

- 关于调试信息：
`void vm_log_debug(char* fmt, ...);`
- 关于应用执行进入功能信息
`void vm_log_info(char* fmt, ...);`
- 关于应用可解决的错误警告信息。例如程序因无信号或欠缴电话费而无法连接网络。一旦帐单缴了或网络信号好转应用就会将问题删除并可正常运行。
`void vm_log_warn(char* fmt, ...);`
- 关于功能失败信息，但应用还未关闭。例如英文教学应用，教师发音功能坏了，但其他功能还正常运作。
`void vm_log_error(char* fmt, ...);`
- 关于致命错误导致应用终止。例如应用才启动时未足够分配存储而导致系统无法产生用户界面和终止。
`void vm_log_fatal(char* fmt, ...);`



注：每个输出字符串长度不得超过 255 个字符，并该系统具有个有限的缓冲器给记录，如果记录功能被调用过于频繁，它有可能失去记录字符串。

5.4. 命令

[命令模块](#)接收来自显示屏工具输入的命令。您可以在显示屏工具中选择“发送命令”功能（细节请参见 4.6 “记录与命令”。该命令格式是 `command_string`。

为处理来自不同端口的输入，您可以调用 [vm_cmd_open_port\(\)](#) 注册一个特定的端口。然后命令串将被传递并提供给 `vm_cmd_open_port()` 回调函数。该命令串最长为 512 bytes。SDK 提供许多的范例代码，它们是使用命令模块来触发范例程序的，请参考命令模块的使用范例。

标准库涵盖了核心编程元素，包括：内存分配，线程创建，计时器回调，日期时间服务，字符串操作，文件系统和应用程序数据和字符集转换功能。

5.5. 标准库

标准库涵盖了核心编程元素，包括：内存分配，线程创建，计时器回调，日期时间服务，字符串操作，文件系统和应用程序数据，记录服务和字符集转换功能。

5.5.1. 存储

[存储模块](#)提供内存分配的功能，使您能够为应用程序堆分配内存。LinkIt Assist 2502 拥有专有的存储管理界面而不是采用标准 C 函数例如 `malloc` 和 `realloc` 分配和解放存储。请使用 [vm_malloc/vm_realloc](#) 来询求存储和 [vm_free](#)，来解放存储。



注：ARM 处理器对字节对齐有严格的要求。如果它需要在一步中访问 4 个字节内容，则该内容的起始地址必须位于 4 个字节的边界上。同样地，如果有必要访问 2 个字节的内容在单一的步骤中，则该内容的起始地址必须位于 2 字节的边界上等等，或将发生异常（exception）。

当发生字节对齐问题时主要与不正确的指针转换有关，例如一个字节的指针转换成一个结构指针。您应该尽量避免这种类型的指针转换。另外，由 `vm_malloc` 返回的指针必须被强制转换并通过 `vm_malloc` 指针返回，而不管其分配的的空间的大小，必须正确对齐。

存储最大量分配是由堆的大小取决，如图 59。当构建应用程序时，软件的存储堆大小是静态配置，如第 5.5.1 章节，“存储”中所述。您应在应用程序设置里配置总存储要求。关于应用程序设置界面请参见第 2.9.1 章节，“应用设置”。软件的总存储需求由两部分组成，负载和堆的大小。如果总存储大小超过了系统可用存储，应用加载将失败，并会在显视工具出现错误记录。

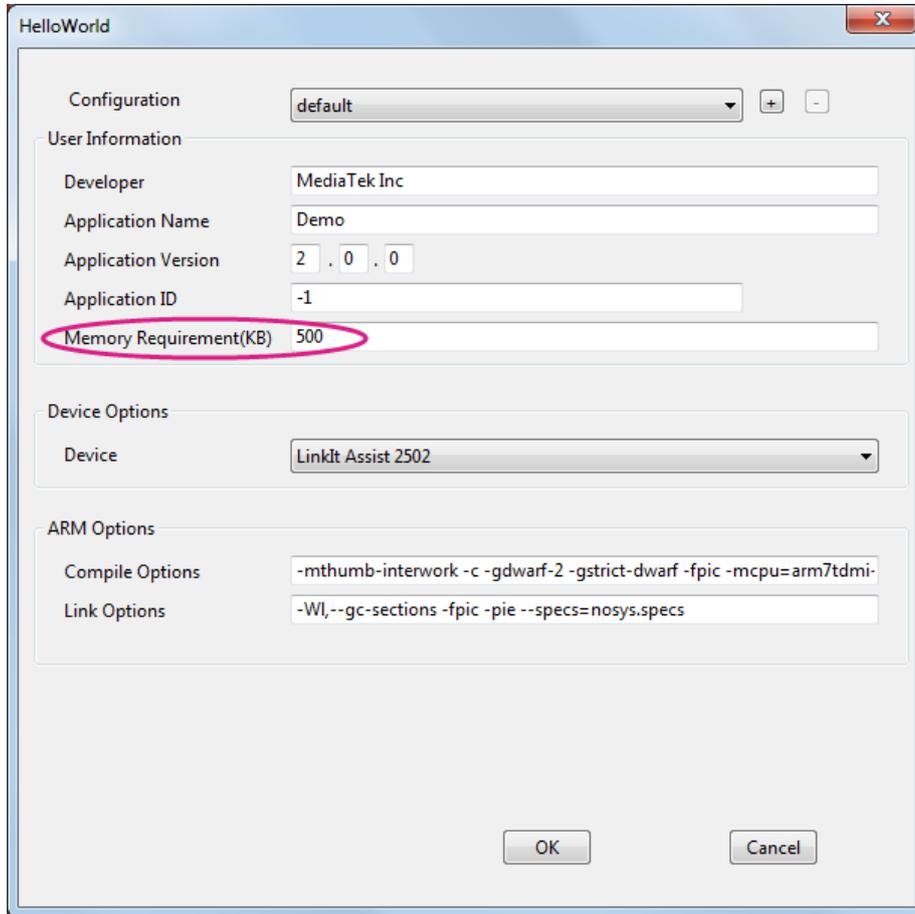


图 59 应用设置

LinkIt Assist 2502 SDK 2.0 版本为应用的系统固件储备预留约 1.7 MB 的 RAM。然而，确切的值将不同硬件和固件而不同，并且当固件版本改变或更新也将不同。因此，建议您使用固件 API `vm_firmware_get_info` 查询 `VM_FIRMWARE_HOST_MAX_MEM`，以确定该应用合适的总存储大小。

若系统在存储分配时无法提供足够的堆存储，分配错误将发生并需要妥当处理。应用必须立即反应和决定 `vm_malloc` 回返回值。当使用 `vm_free` 解放存储时，务必确认存储地址是正确的，也必须由应用来解放，要不然的话，存储将有泄漏问题并产生错误。

请记住若应用调用太多次 `vm_malloc` 或 `vm_free` 而每次都是小块存储，那么存储断片可能发生。当存储断片发生时，就算总存储是足够的，分配还是可能会产生错误，原因已无足够的连续存储供使用。

LinkIt Assist 2502 SDK 提供一个 API 让硬件访问存储，[vm_malloc_dma](#) 分配无缓存存储，DMA 是指直接内存存取。有了无缓存存储，硬件和软件就不必要更新或冲洗 CPU 的高速缓存。因此您应调用 `vm_malloc_dma` 将无缓存存储发给硬件驱动，例如图像模块用的缓存 5.17 “图像”。

5.5.2. 字符串和字符集

虽然处理 ASCII 字符串很常规，但对于具有本地化/全球化规范的应用程序，处理 Unicode 字符串是很重要的。在 SDK 中，VMSTR 类型代表 ASCII 编码的字符串；而 VMWSTR 类型代表 UCS2 编码字符串。该 UCS2 编码字符串在 SDK 非常普遍，尤其是在文件路径。所有字符串参数代表的文件夹/文件路径必须是 UCS2。

LinkIt Assist 2502 提供基本的字符串 UTF8，UTF16，UCS2 和 ASCII 编码格式之间的[转换功能](#)。需要转换的字符串必须是空终止，并注意 UCS2 结尾的空终止是 2 字节长。因存储由调用者分配，所以无内部存储分配，解放，但是您应记住字符串包含空中指不该超过 256 字符。

5.5.3. 定时器

[定时器功能](#)接受一个时间参数和一个回调函数。在指定的时间过后，回调函数将被调用。

SDK 提供三种计时器：精确的，非精确，和 HISR 定时器。当系统进入省电模式时，精确计时器将停止，而非精确定时可从省电模式中唤醒系统，但它可能在定时器回调间隔引入方差。最大可能的方差为 254 系统滴答，其中 1 滴答等于 4.615 毫秒。

LinkIt ONE 固件的设置为永不进入休眠模式，因此您不需要防止系统进入休眠来使用精确计时器。

[vm_timer_create_precise](#) 可用于建立精确计时器，而 [vm_timer_create_non_precise](#)，建立非精确计时器，定时器的最小精确度为 10ms。当定时创建时，该系统将定期调用计时处理的回调函数，直到定时器被应用删除。在一个定时处理函数未完成之前，下一轮的计时操作不会执行，而若当前计时功能未完成，其他计时处理也不能开始，因此，计时回调会涉及一定的时间延迟。定时器越多，计时器的处理功能越复杂，并且还会累积时间延迟。还要注意的，当程序退出时，应用必须删除所有的计时器。当系统进入省电模式，而您需要应用继续运行，请用 [vm_timer_create_non_precise](#) 创建定时器。

[HISR 计时器](#)非常精确并且不收系统休眠状态影响。然而，因它的环境执行性质，HISR 定时器的回调函数不能调用任何 LinkIt Assist 2502 的 API，因此它只能被用来进行数值的快速更新。不然，关键系统故障例如 GSM modem 故障因它的驱动需要实时更新。

5.5.4. 日期与时间

该[程序模块](#)提供查询和设置当前系统时间精准度至秒，甚至微秒的功能。

一共有三种日期和时间函数。第一个是日期时间 API，它提供描述年，月，日和时结构。您应该使用这个 API 来设置和获取当前的系统时间，如 [vm_time_set_date_time](#)。

第二个是 UNIX 时间的 API，根据 UNIX 时代返回当前系统时间秒数，例如：[vm_time_get_unix_time](#)。

若想测量高精度持续时间，例如图形动画，您可用微秒定时器（美国）[vm_time_ust_get_count](#)，然后传递两个计数器值 `vm_time_ust_get_duration` 计算两个之间的时间差异。提醒您，此函数处理计数器的值溢出，因此，它比直接地减计数器值更妥当。



注：[vm_time_ust_get_count](#) 能处理定时器值溢出(overflow),因此它比减定时器值安全。

5.5.5. 线程

LinkIt Assist 2502 开发平台是一个多线程的环境。在 LinkIt Assist 2502 平台里，应用是由主线程上运行。应用可以调用 [vm_thread_create\(\)](#) 来创建一个子线程，并指定优先级。系统允许最多十个子线程。该指定优先级只是一个建议值。根据当前的系统资源的子线程，该系统将相应地分配一个实际的优先级（基于指定优先级）。

如果建议的优先级等于或小于 128，那么系统指定的实际优先将比主线程还低。而若建议优先级等于或大于 129，系统实际优先将比主线程还高。如果建议优先级为零，则系统将分配默认优先级给线程，这比主线程的优先级低。

一个线程可以通过调用 `vm_signal_create()` 创建一个信号，然后调用 `vm_signal_post()` 传播信号来达成线程同步。该[程序模块](#)还提供互斥锁功能来保证共享数据操作的安全。

一个线程可以通过 `vm_thread_send_message()` 发送异步信息到另一个线程，或从另一个线程通过 `vm_thread_get_message()` 接收发送的信息。发送消息前，您必须定义一个信息 ID，并确保信息的 ID 是在线程之间唯一的。消息 ID 的有效范围是 1000 到 60000。然后信息 ID 和相应的用户数据可以被发送到另一个线程通过查询线程句柄 `vm_thread_get_current_handle()`。还有个专用的 API 为创建的线程检索句柄至主线程，它就是 [vm_thread_get_main_handle\(\)](#)。

有个重要的例外，当发送一个信息至主线时，`vm_pmng_register_system_event_callback` 回调注册的信息是重新定向至系统事件句柄的。在这种情况下，就没有必要在主线程调用 `vm_thread_get_message`。而您应处理系统事件句柄的信息 ID。

5.5.6. 使用 C 标准库函数

SDK 内的应用不需要 C 标准库来执行。但您编程时可以选择链接至 C 标准库。然而，不是所有的 C 标准库功能都支持，若采用了不支持的功能，应用将失败。以下功能不支持，包含操作功能，主张功能，和存储功能。

- 文件操作界面, 例如 `fopen / fread / fwrite`.
- 中止与退出界面, 例如 `abort / exit`.
- 字符访问界面, 例如 `getchar / getc / putchar / putc`.
- 输入显示界面, 例如 `scanf / printf`.
- 主张与调试界面, 例如 `ASSERT / TRACE`.

其他平台中立函数可以是编译，链接，并执行，例如 `rand ()` 和 `srand ()` 函数。

5.5.7. 文件系统

该 SDK 支持 FAT32 文件系统。路径的最大长度为 256 个字符，包括终止 NULL。路径名的编码格式是 UCS2。

SDK 定义了几个分区驱动器。这些驱动器是由文件路径驱动器号访问。对于内部存储，[vm_fs_get_internal_drive_letter](#) 会查询该驱动器的盘符；对于外部存储，例如 SD 卡，[vm_fs_get_removable_drive_letter](#) 将查询驱动器的盘符。请注意，有些 HDK 不附带 SDK 卡插槽。在这种情况下，固件不支持外部（移动）驱动器。

在 SDK 没有“当前”目录中的概念。当读取和写入文件时，必须使用绝对路径。而 SDK 不支持同时打开多个文件，因为内存资源约束而不建议。应及时关闭不再使用的文件，不然它可能因缺乏可用的文件句柄资源而导致失败。此外，不要从 `VM_EVENT_QUIT` 的信息处理逻辑中打开一个文件，否则操作将失败。

当读或写一个文件时，您必须对可能发生的任何错误进行适当的错误处理，否则它会导致异常。请参阅 [VM_FS_RESULT](#) 了解错误代码。

5.5.8. 应用数据

虽然存储应用数据至文件系统很方便，但有时候您需要存储数据在 HDK 大量存储模式无法访问的地方。应用数据是一组文件系统 API，它可以操作一些用户看不见，存在隐藏分区（系统驱动分区）的数据，该数据最大可达 2KB。该分区无是加密的，一般来说，应用数据存储要越少并且加密，而且是靠应用 ID 在分区里排列，因此设定特定应用 ID 来访问应用数据很重要。应用 ID 能由标签模块而检索，而标签在下一章有更多说明。

5.6. 标签

[标签模块](#) 让您可以嵌入特定的字符串和整数信息至您的应用程序，包括开发者，应用程序名称，应用程序版本和应用程序 ID。这些嵌入式的 VXP 可执行字符串和整数称为标签。

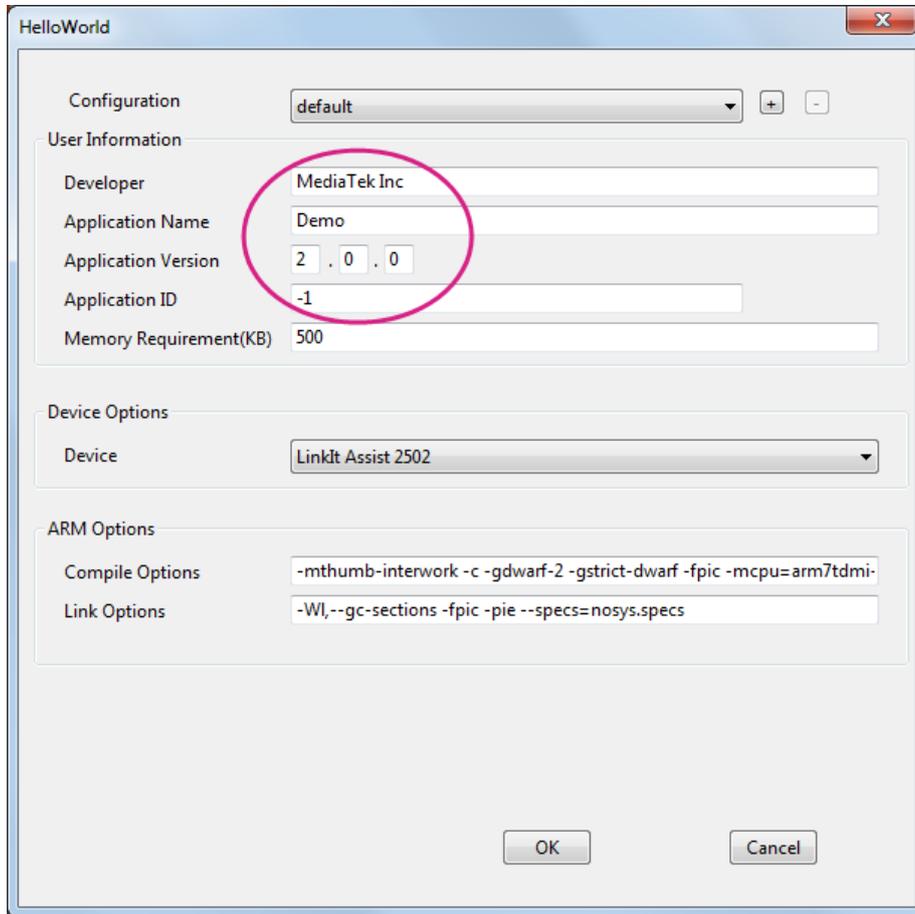


图 60 应用设定里的标签设定

要定义标签信息，请在 Eclipse 插件使用项目设置，如图 60。您可以调用 [vm_tag_get_tag](#) API 来检索应用标签信息。该 API 允许 NULL 缓冲参数，而功能将输出所需的缓冲长度为给应用 ID，包括终止 NULL，因为标签并无特定长度要求，如下代码：

```
VMWSTR app_name = NULL;
// passing NULL to retrieve tag size.
vm_tag_get_tag(NULL, VM_TAG_ID_APP_NAME, (void*)NULL, &tag_size);
// allocate buffer according to tag size
app_name = vm_malloc(tag_size);
// copy the tag string into the buffer.
vm_tag_get_tag(NULL, VM_TAG_ID_APP_NAME, app_name, &tag_size);
```

5.7. 资源

应用可以调用[资源 API](#) 来加载资源至资源编辑。每一个资源的识别于访问是由一个标识符实现的，如图 61。

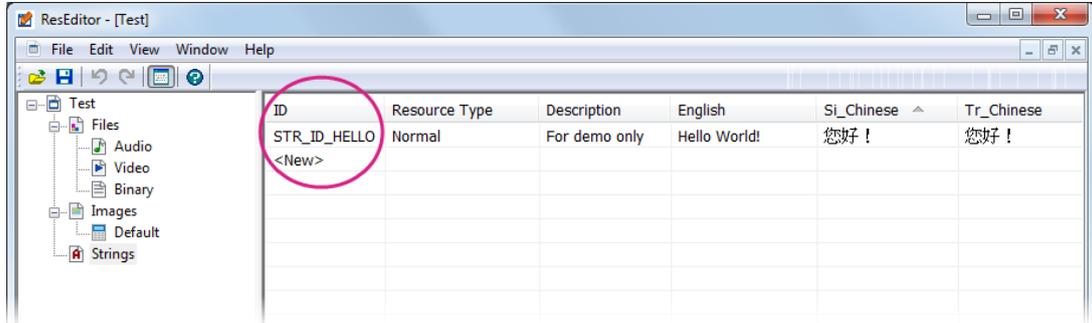


图 61 资源标志符范例

注：资源标志符有以下限制：

- ID 长度不得超过 64
- 首字不得由 0-9 开始

所有的字符必须是 A - Z, a - z, 0 - 9, or ‘_’

应用程序必须首先初始化资源与所需的语言 ID 用 [vm_res_init\(\)](#)。语言 ID 映射到[资源编辑器语言窗口](#)中的语言 ID：

```
/* load the "Tr_Chinese" language strings */
vm_res_init(886);
```

而 [vm_res_get_string](#) and [vm_res_get_image](#) API 可以用于检索资源。

应用可用 [vm_res_delete](#) 来解放资源数据内存，除了字符串资源之外，因为它在资源启动时被加载至内存并会被资源模块解放。

```
/* release the resource occupied by the image resource */
vm_res_delete(IMG_ID);
/* unload all resources from memory, including string resources */
vm_res_deinit();
```

资源加载失败可能是因为某一些原因，如下：

- 1) LinkIt Assist 2502 资源编辑路径不对，因此未能将资源包在应用内。
- 2) 启动前查询资源。请确认您在查询资源之前调用 `vm_res_init`，并调用 `vm_res_delete()`，当您不需要再用资源。
- 3) 内存或文件句柄不足。原因可能是应用占用太多的内存，或同时打开太多的文件。

4) 在 VM_EVENT_QUIT 信息句柄逻辑加载资源。当应用在执行关闭时是无法加载资源地。

5.8. 过程控制

[系统模块](#)提供核心功能至 LinkIt Assist 2502 应用。当执行文件加载至内存时，将变成一个过程。之后过程将注册系统事件句柄以便回应系统事件例如进入或退出应用。还有检索过程信息的 API 例如执行文件路径和过程状态。

另外还有 API 以协助更新应用执行文件。环境架构将终止应用，检查和更新应用执行文件和一个 API 发动程序。请参阅 API 参考 [vm_pmng_exit_and_update_application](#) 了解更多。

5.9. 固件

[固件模块](#)提供固件相关的 API 以询问固件信息例如版本和创建时间。

还提供 API 来询问特定功能例如 Wi-Fi 为当前运行固件。

最后，[vm_firmware_trigger_update](#) API 为 LinkIt Assist 2502 应用提供固件更新触发。固件更新包与 SDK 一同发布，应用应该实施传输机制来传输固件更新包至硬件的内存储然后触发更新过程。请参阅 LinkIt Assist 2502 SDK 里的 Framework_Firmware_FirmwareUpdate [参考范例](#)。

5.10. 驱动 API

[驱动 API](#) 提供了连接周边的功能。周边设备是经由硬件引脚连接至 LinkIt ONE 或 LinkIt Assist 2502 HDK，而不同的 HDK 有不同的分汇引脚布局和功能。例如 LinkIt ONE 数字引脚为 D0, D1, D2,...等，LinkIt Assist 2502 引脚为 P0, P1, P2, ...等。

另外，同个分汇引脚可以支持多种功能，例如 LinkIt ONE 的 D0 引脚 也可用于 UART 串口。

为了支持不同硬件与不同引脚名称，我们提供了一般装置层（DCL），让您能在一般 API 访问周边功能。这使所有的硬件端口例如 UART，GPIO，ADC，I2C，SPI 和 EINT 可由同个功能而不同的参数访问，例如 `vm_dcl_open`, `vm_dcl_close`, `vm_dcl_control`, `vm_dcl_read` 和 `vm_dcl_write`。

该 [DCL 模块](#) 是以引脚号码来识别每个引脚，而 SDK 提供了一个头文件 `vmboard.h` 将开发板引脚与 DCL 的引脚名称作映照。这样，您只需要加该头文件，就可以依照用开发板的引脚 macro 作参考，例如，您可以映照以下引脚输出图 62，GPIO1 引脚对应到第 12 引脚。

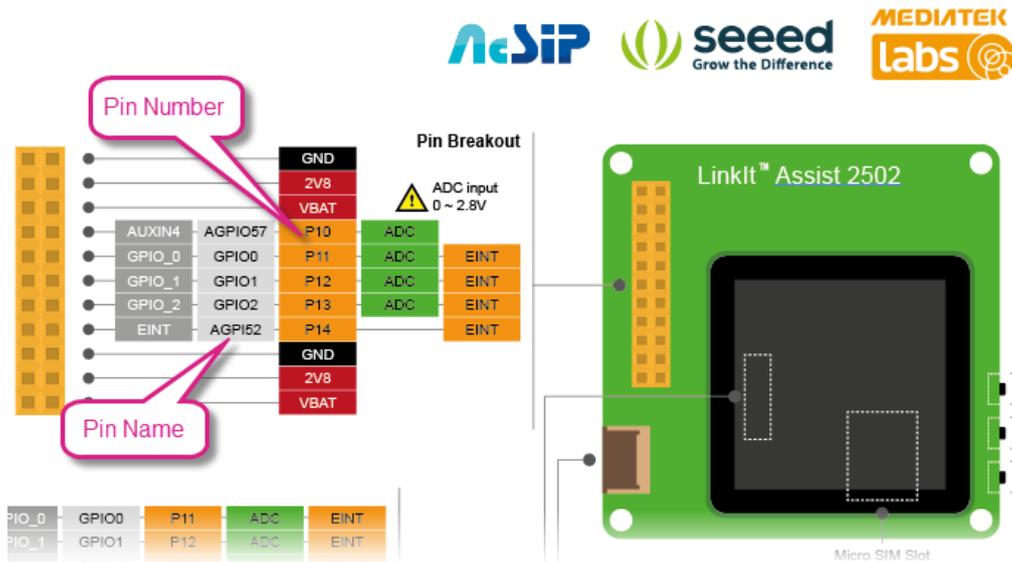


图 62 引脚名称与引脚号码映照图

DCL 模块提供了 `vm_dcl_config_pin_mode` API 来访问一个引脚所有的不同的功能。该 API 能设置一个引脚号的功能，例如设置 LinkIt Assist 2502P8 引脚为 GPIO 引脚，如下代码。请参见图 5 引脚图查看 P8 的位置。

```
vm_dcl_config_pin_mode(D0, VM_DCL_PIN_MODE_GPIO);
```

如果要将引脚设置成 UART，请注意 UART 协议必须使用两个引脚（D0/D1），因此这两个引脚都应该要设置，如下：

```
vm_dcl_config_pin_mode(D0, VM_DCL_PIN_MODE_UART);
vm_dcl_config_pin_mode(D1, VM_DCL_PIN_MODE_UART);
```

在 HDK 文件内有引脚所支持的功能细节。有一些 API 是用指数来映照引脚，例如 PWM ID 和 ADC channel ID。在这种情况下，您可以用宏来转换引脚号至功能指数。如下：

```
PIN2PWM(D3) /* Get PWM ID of pin D3 */
PIN2CHANNEL(A0) /* Get ADC channel ID of pin A0 */
PIN2EINT(D2) /* Get EINT ID of pin D2*/
```

之后，您能使用 `vm_dcl_open` 来打开设备和复取相应的句柄。而该句柄能让 `vm_dcl_read` 和 `vm_dcl_write` 使用来操作一般读与写，或者让 `vm_dcl_control` 使用来发送设备命令结构至底层硬件。请参阅各个子模块以了解该有的命令结构。

有一些硬件操作能让您注册硬件事件回调，请注意这些回调不是在硬件中断环境执行的，而是在主线执行的。这能确保正确的事件顺序但是会牺牲即时的约束。接下来我们将简要地解释每个子模块。

5.10.1. ADC

[模拟数字转换器](#) (ADC) 模块是用于读取模拟输入引脚数值，它可以将输入电压转换成离散整数 0~1023。不同的 HDK 电路设计会有不同的参考电压。请参阅 [HDK 文件](#) 了解细节。

5.10.2. GPIO

[GPIO 模块](#) 能让您设定引脚电压为高电压或者低电压，并可以当输入引脚或者输出引脚。但是在使用某个 GPIO 引脚之前，您需要为引脚设定引脚模式。引脚的默认值是输入，在大多数情况下，当您使用 GPIO 引脚时，您需要结合多个引脚，建造一个端口与周边设备进行通信。

5.10.3. I2C

[I2C 模块](#) 支持 I2C 协议来连接周边设备，该 SDK 作为 I2C 主机。设定引脚模式之后，打开 I2C 句柄，并用以下命令调用 `vm_dcl_control()`。

- 发送 `VM_DCL_I2C_CMD_CONFIG` 命令来设定从机地址和交易模式和模式速度。
- 发送 `VM_DCL_I2C_CMD_SINGLE_WRITE/READ` 或 `VM_DCL_I2C_CMD_CONT_WRITE/READ`，或 `VM_DCL_I2C_CMD_WRITE_和_READ` 来与从机通讯。

5.10.4. SPI

LinkIt Assist 2502 HDK 是以 SPI 主机运作的，并可以连接 SPI 从机设备。只有当相关的参数设置符合从机的需求时，主机和从机才可以进行正常的通信。因此，在开始前请务必注意以下设定：

- 使用 `VM_DCL_SPI_PORT1` 访问 SPI 端口。
- 确认 `VM_DCL_SPI_CONTROL_SET_CONFIG_PARAMETER` 设定如下：
- `rx_msb` 和 `tx_msb` 数据是 "MSB first" 或者 "LSB first"。
- 时钟的极性和相位设置 `clock_polarity` 和 `clock_phase` 是正确的。
- 符合从机传送速度。
- [SPI 模块](#) 是以 DMA 模式运作的，而交换数据缓冲必须是不可缓存的。因此，`vm_malloc_dma` API 应该使用来分配 SPI 缓冲结构 `vm_dcl_spi_control_read_write_t`。

5.10.5. 串行 IO (UART)

[串行输入和输出](#) (SIO) 模块支持 UART 协议。而 SDK 支持 USB 仿真的 UART 端口和 HDK 的硬件串行设备。您应该只使用硬件串行设备，因为 USB UART 端口已被 SDK 使用来当作显示工具和上传工具。请先设置 UART 模式的引脚来访问硬件串行设备，如下代码：

```
vm_dcl_config_pin_mode(D0, VM_DCL_PIN_MODE_UART)
vm_dcl_config_pin_mode(D1, VM_DCL_PIN_MODE_UART)
```

然后打开硬件串行设备 VM_DCL_SIO_UART_PORT1,如下：

```
handle = vm_dcl_open(VM_DCL_SIO_UART_PORT1, vm_dcl_get_ownerid());
```

接下来请调用 vm_dcl_control 来设置 UART 端口，并使用命令

VM_DCL_SIO_COMMAND_SET_DCB_CONFIG，和参数 vm_dcl_sio_config_t。请参阅 API [参考](#) 细节。

最后，您应该使用 vm_dcl_register_callback 来注册当读写缓冲可用时调用的回调，和使用 VM_DCL_SIO_UART_READY_TO_WRITE 事件，和 VM_DCL_SIO_UART_READY_TO_READ 事件并调用 vm_dcl_write / vm_dcl_read，当缓冲可以使用时。

5.10.6. PWM

[脉宽调制](#)（PWM）是用来转换离散数字信号到连续模拟信号。它可以用来电亮一个 LED 让它有不同的亮度。LinkIt ONE 和 LinkIt Assist 2502 都支持 PWM VM_DCL_PWM_1 and VM_DCL_PWM_4。

控制 PWM 的步骤如下范例。例如 LinkIt Assist 2502 引脚 P0 和 P1 支持 PWM。首先将这两个引脚设置成 PWM。

```
vm_dcl_config_pin_mode(D3, VM_DCL_PIN_MODE_PWM);
```

然后打开 PWM 设备柄。

```
pwm_handle = vm_dcl_open(VM_DCL_PWM_1, 0);
```

调用 vm_dcl_control 来设置 PWM 模块，并发送以下命令：

- VM_PWM_CMD_START
- VM_PWM_CMD_SET_CLOCK
- VM_PWM_CMD_SET_COUNTER_AND_THRESHOLD

最后，用 vm_dcl_close 将柄关闭若不再需要用它。

PWM 模块里有个值得一提的应用，它就是在 LinkIt Assist 2502 HDK 显示模块的背光源控制。该背光源控制事由 VM_DCL_PWM_1 来控制，它可以由 VM_PWM_CMD_SET_COUNTER_AND_THRESHOLD 来控制背光的亮度，0 是最低并无亮度门槛，门槛越高，亮度也越高，请注意，该门槛应该比计数器还小。



注：门槛值应该比计数器值要小。

5.10.7. 外部中断 (EINT)

该 [EINT 模块](#) 允许应用注册回调到外部中断。该处理回调在主线程执行，而不是在中断执行，所以没有再进来之考虑。外部中断是由电压的变化而触发，包括 RAISING，FALLING 或 EDGE 触发。由于回调

处理是在主线执行的，因此，事件产生比处理功能还要快是有可能的。若是这样，多余事件可能积累主线事件队列而导致系统错误。我们建议您设置硬件去抖，以防止过度的 EINT 事件。去抖的设置最短期限为 1 毫秒。请参考 LinkIt Assist 2502 SDK 里的 DCL_EINT 范例代码来了解使用流程。

5.10.8. PMU

[功率管理单元](#) (PMU) 模块可以用来控制振动器引脚的功率。要控制振动器引脚，先打开 PMU 模块如下：

```
pmu_handle = vm_dcl_open(VM_DCL_PMU, 0);
```

之后发送 VM_DCL_PMU_CONTROL_LDO_BUCK_SET_ENABLE 命令和 vm_dcl_control, 如下：

```
vm_dcl_pmu_ld0_buck_enable_t val;  
val.enable = TRUE;  
val.module = VM_DCL_PMU_VIBR;  
vm_dcl_control(pmu_handle, VM_DCL_PMU_CONTROL_LDO_BUCK_SET_ENABLE, (void *)&val);
```

请参考 DCL_PMU_Vibrator 范例和使用。

5.11. 网络

SDK 支持 Wi-Fi 和 GPRS 互联网连接并提供一组[共用 API](#)。您应该使用 Wi-Fi 和 GSM-GPRS 模块设置硬件的连接，然后用网络 API 创建远端伺服器连接。以下子模块供您使用：

- **BSD sockets:** 提供兼容 BSD socket
- **TCP and UDP 模块:** 提供同步和异步 socket 连接
- **SSL 模块:** 提供 SSL socket 连接
- **HTTPS 和 HTTP 模块:** 提供 HTTPS/HTTP 连接

5.11.1. BSD Sockets

BSD socket 是用于处理间通信 (IPC)，它有 internet sockets 和 UNIX domain sockets 的 API 计算库。LinkIt Assist 2502 实施了大部分的 [BSD socket API](#)。您可以创建伺服或客户 socket，但是您必须先设置 Wi-Fi 或 GPRS。细节请参见 [Wi-Fi 模块](#)和 [GSM-GPRS 模块](#)。以下步骤描述如何使用 BSD socket API 创建一个伺服 socket。

1) 打开 Bearer

GPRS 或 Wi-Fi bearer 在创建 socket 连接之前应先打开，让 BSD socket 知道使用哪一个底层连接硬件。使用 `vm_open_bearer` 和 APN 参数打开 bearer，APN 将指示使用哪种 bearer，和确认一下设定：

- a) `VM_BEARER_DATA_ACCOUNT_TYPE_GPRS_NONE_PROXY_APN`:使用 GPRS bearer，无 proxy。
- b) `VM_BEARER_DATA_ACCOUNT_TYPE_GPRS_PROXY_APN`:使用 GPRS bearer，有 proxy。
- c) `VM_BEARER_DATA_ACCOUNT_TYPE_WLAN`:使用 Wi-Fi bearer。
- d) `VM_BEARER_DATA_ACCOUNT_TYPE_GPRS_CUSTOMIZED_APN`:使用定制 APN。

如果 bearer 有成功打开，并 `VM_BEARER_OK` 将返回，然后应用就会创建 socket 连接。如果返回是 `VM_BEARER_WOULDBLOCK`，那代表应用正在打开 bearer，并在接受到 `VM_BEARER_ACTIVATED` 事件回调之前不应该创建 socket。

2) 保留 bearer (可选择)

对于 GPRS bearer，平台将在 socket 连接关闭时自动释放 bearer。如果应用想保留 bearer 为了要创建另一个 socket，您可以调用 `vm_gsm_gprs_hold_bearer()`。当应用不再需要使用 bearer 时应调用 `vm_gsm_gprs_release_bearer()`，来释放它。

3) 创建 socket 连接

`socket()`是用于创建 socket 连接的。BSD socket API 与创建 API 的 socket 连接是同样地。与 BSD socket 比较起来，LinkIt Assist 2502 只支持 `PF_INET / PF_INET6` 协议和 `SOCK_STREAM / SOCK_DGRAM` socket 类型。

4) Bind socket

当使用 `socket()` 来创建 socket 时，它仅给出一个协议族，但是没有分配一个地址。因此您必需使用 `bind()`来分配地址，才能这使 socket 来接收其他主机设备。

5) 聆听 socket

当一个 socket 已与地址联结，`listen()` 会为将进入连接作准备。然而，这仅对于面向连接数据模式 (`SOCK_STREAM`) 才需要。

6) 经由连接发送或接收数据

当收到一个客户端连接要求时，您能使用 `send()` / `sendto()` / `recv()` / `recvfrom()` 功能而经由连接发送或接收数据。

7) 关闭 socket 连接

当连接已不再需要时，`closesocket()` 调用将关闭连接并释放 socket 资源。

8) 释放 bearer

Bearer 将被平台自动释放，若 `vm_gsm_grps_hold_bearer` 没有被调用。这时，您需要调用 `vm_gsm_gprs_release_bearer` 才能释放硬件连接。

用户 socket 创建步骤如下：

1) 打开 bearer

与伺服模式相同。

2) 保留 bearer (可选择)

与伺服模式相同。

3) 创建 socket 连接

与伺服模式相同。

4) 设定 socket 连接

Socket 创建之后，应用可调用 `setsockopt()` 来设定 socket，socket 可设置为异步模式，这代表即使无数据，应用将立刻返回。当应用没收到完整数据时，它就会使用轮训。

应用程序使用 `connect()` 以启动由参数列表主机地址指定的远程主机连接。

5) 经由连接发送或接收数据

与伺服模式相同。

6) 关闭 socket 连接

与伺服模式相同。

7) 释放 bearer

与伺服模式相同。

所有的应用共用 socket 资源，socket 最多可达 12 个，还可能被其他资源限制。如果没有 socket 可用时，`vm_soc_socket()` API 将返回错误代码 `VM_SOC_LIMIT_RESOURCE`。

Socket API 可以被 API 线程调用，因为一部分的 socket API 是同步线程，我们建议用与主线程不同的线程调用阻挡 API 以免造成主线程事件队列堵塞。

5.11.2. TCP

在头文件里定义的 TCP 模块 [vmtcp.h](#) 是一组创建同步与异步 TCP 用户和伺服器调用的 API。TCP 最多能有三个并发连接。

以下范例 API 让您由账户参数分配底层硬件连接：

```
vm_tcp_connect_sync(URL, Port, VM_BEARER_DATA_ACCOUNT_TYPE_WLAN)
```

这样您就不需要在 BSDsocket 模块调用另一个 bearer API。

在使用同步 API 时，就算操作很慢，应用还是被挡住，直到收到结果为止。这组 API 是线程安全的并可以调用来创建线程。请参阅 Network_TCP_Sync 范例。

在使用异步 API 时，应用将注册回调功能，但不会因为还未收到结果而被阻挡。当结果收到时，将调用指定事件的回调。请参阅 Network_TCP_Async 范例。



注:异步 API 不是很线程安全，它只能在主线调用。请不要在已创建线程里调用异步 API。

5.11.3. UDP

[UDP 模块](#)只拥有异步 API。这些 API 只能在主线调用。应用能经由这些 API 来发送以及传送数据。使用 UDP 传或送数据涉及以下：

- 使用 `vm_udp_create()` 创建 UDP 连接。应该注册一个回调功能。
- 在回调功能处理 UDP 事件。
- 收到 `VM_UDP_EVENT_WRITE` 之后，调用 `vm_udp_send()` 来发送数据。
- 收到 `VM_UDP_EVENT_READ` 之后，调用 `vm_udp_receive()` 来发送数据。
- 当连接断开时，收到 `VM_UDP_EVENT_PIPE_BROKEN`。
- 当连接关闭时，收到 `VM_UDP_EVENT_PIPE_CLOSED`。
- 使用 `vm_udp_close()` 当不再需要连接时。

5.11.4. SSL

SDK 有一组 [SSL API](#) 基于 XySSL SSL 库界面。以下是 SSL API 使用步骤和范例。

- 准备证书。SDK 没有预先的安装证书。您应提供 X.509 格式的证书，它们可以从任何浏览器出口。
- 使用 [vm_ssl_connect\(\)](#) 来连接主机。这是一个异步 API 而事件结果将传至注册回调。
- 用 [vm_ssl_load_ca_chain_certificate\(\)](#) 加载证书。
- 安全连接建立后，使用 [vm_ssl_read\(\)](#) 来读入或 [vm_ssl_write\(\)](#) 来写出。
- 使用 [vm_ssl_close\(\)](#) 来关闭连接。



注：SSL API 只能在主线执行。

5.11.5. HTTPS

SDK 支持 HTTPS 协议，而该协议定义在 [vmhttps.h](#)。一般地 HTTP 功能例如 GET/HEAD/POST/PUT/DELETE 也支持，请调用 VM_HTTPS_PROTOCOL_HTTP 来设定 HTTP。

HTTPS 的关键步骤有：

- 1) 注册 [vm_https_register_context_and_callback\(\)](#) 回调功能。
- 2) 调用 [vm_https_set_channel\(\)](#) 来设置专有的 HTTP 渠道要求。
- 3) 使用 [vm_https_send_request\(\)](#) 来发送 HTTP 请求。
- 4) 使用 [vm_https_read_content\(\)](#) 来读取内容。

每一个 API 的回调功能细节如表 5.

回调功能	API	说明
vm_https_set_channel_response_callback	vm_https_set_channel	当渠道打开时，该回调将被调用。
vm_https_unset_channel_response_callback	vm_https_unset_channel	当渠道关闭时，该回调将被调用。
vm_https_release_all_request_response_callback	NA	
vm_https_wps_termination_callback	vm_https_set_channel	当打开渠道时有错误，该回调将被调用。
vm_https_send_response_callback	vm_https_send_request	当有数据可以写出时，该回调将被调用。

回调功能	API	说明
vm_https_read_content_response_callback	vm_https_read_content	当有数据可以读入时，该回调将被调用
vm_https_cancel_response_callback	vm_https_cancel	当请求被取消时，该回调将被调用
vm_https_status_query_response_callback	NA	

表 5 HTTPS 回调功能

5.12. Wi-Fi

LinkIt Assist 2502 提供了一系列 API 来支持 Wi-Fi 功能，有这些 API，应用程序可以执行以下的 Wi-Fi 无线操作：

- 无线网络控制：扫描并连接到 Wi-Fi 接入点
- 智能连接能聆听智能连接配置广播。这些配置广播可以与 iOS 和 Android 的智能连接库的智能手机广播。智能连接位于 SDK 文件夹中的客至文件夹。

5.12.1. Wi-Fi 控制

您可以调用 [vm wlan mode set\(\)](#) 让应用设定为无线台模式来连接接入点，或者 VM_WLAN_MODE_OFF 来关闭 Wi-Fi。当应用是在无线站模式，它能操作以下：

- 1) 扫描 AP 列表：多达 16 个 AP 可以被扫描。确切的限制可能是在不同的硬件平台上而不同。
- 2) 连接到 AP：应用程序可以与 SSID/验证模式/密码进行连接的 AP
- 3) 获取信息，如 MAC 地址/ IP 地址/子网掩码设置。
- 4) 启用 WLAN 漫游。WLAN 漫游使站 AP 组里 AP 之间漫游。

5.12.2. 智能连接

智能连接功能让用户从智能手机以动态模式设定无线接入点。在智能手机里这个应用程序将使用广播的智能连接协议来广播 AP 的信息。您可以找到应用范例库组在 SDK 里的 **Custom** 文件夹。

要收听智能连接配置，应用程序应该设置 Wi-Fi 为 STA 模式，并通过调用 [vm wlan sniffer on](#) 进入嗅探模式。然后，应用将接收到 cb_type 作为 VM_WLAN_SNIFFER_ON_AP_INFO 回调函数的配置信息。接下来应用就可以运用这个信息来连接相应的 AP。

5.13. GSM

GSM 模块让开发者访问 2G 连接功能，其中包括：

- SIM 卡信息
- 蜂窝信息
- 高层电信操作
- GPRS 数据连接

这些 APIs 在 [vmgsm *.h](#) 定义，例如，SIM 卡信息是定义在 `vmgsm_sim.h`。您需要准备有效的 SIM 卡来操作有效的 GPRS 和 SMS 计划。请先插入 SIM 卡至 HDK 在启动设备之前。



注: GSM modem 的默认值是打开的，请调用 `vm_gsm_switch_mode` 函数来切换于打开或关闭。细节请参见 `vmgsm.h`。

5.13.1. SIM 卡

该 SDK 提供了许多的 SIM 卡相关的 API 调用，以获得相关的 SIM 卡的状态和信息，例如，IMEI 可以通过调用 [vm_gsm_sim_get_imei](#) 进行检索。有一点要注意的是，该模块与具有多个 SIM 卡的硬件兼容，并且 SIM 卡的标识符是从 `VM_GSM_SIM_SIM1` 开始而不是 0。

5.13.2. 电信

该 SDK 提供了高层次的 API 应用程序来拨打或接听电话，该电信在 [vmgsm_tel.h](#) 里定义。附注，当拨打或接收电话呼叫时音频硬件是被占用的，因此音频 API 将被电话 API 而中断。

5.13.3. 蜂窝信息

GSM 蜂窝信息模块让开发者能够检索有关基站的信息，其中包括：

- ARFCN - Absolute Radio Frequency Channel Number
- BSIC - Base Station Identity Code
- MCC - Mobile Country Code
- MNC - Mobile Network Code
- LAC - Location Area Code
- CI - Cell ID Received signal levels

请参阅 [vmgsm_cell.h](#) 了解更多细节。调用 `vm_gsm_cell_open` 来分配模块需要的资源。然后：

- 调用 `vm_gsm_cell_get_current_cell_info` 检索有关当前蜂窝信息
- 调用 `vm_gsm_cell_get_neighbor_cell_info` 检索有关邻居蜂窝信息。

由于小单元信息可被更新，当系统事件 `VM_EVENT_CELL_INFO_CHANGE` 有细胞信息变化时，将被分派到系统事件处理程序。

请调用 `vm_gsm_cell_close` 来停止查询小区信息。

5.13.4. SMS

该 SDK 提供了一些高级功能的短信，其中包括：读/发送/取消发送/创建并获得短信中心号码。在系统启动后短信模块并不可立即使用。您可以使用 API [vm_sms_is_sms_ready](#) 来检索服务的可用性。等模块准备好了之后，开发者可以发送信息或保存邮件入库 - SIM 卡或 T-卡。它可以删除指定的信息，或者删除 API 指定的框消息。这些消息是在重启后存在的。



注:取消发送 API (`vm_sms_cancel_send()`) 只能在调用发送 API (`vm_sms_send`) 之后和发送回调之前使用。

5.13.5. GPRS

GPRS 模块是 GSM 模块的子模块并提供多种 API 来设置 GPRS 功能：

- 保留/释放 bearer

如第 5.11.1 章节“BSD Sockets”描述，通过保留 bearer，bearer 可再用于不同的 socket 连接。如果该应用程序必须在很短的时间周期之后发出多个连接，保留 bearer 是很有效率的方法。

- 设定定制 APN

LinkIt Assist 2502 平台为不同的运营商预载了许多默认的 APN，但固件不能涵盖所有的运营商，因此它允许应用程序中设置了自己的 APN 通过提供 [vm_gprs_set_custom_apn_info\(\)](#) API，`data_account` 参数应该传递给该 API，并通过调用 `vm_gprs_get_encoded_data_account_id()` 而产生。

5.14. 蓝牙

SDK 支持蓝牙 2.1 和 4.0，以下子模块支持：

- 连接管理：该模块控制蓝牙开关，设备发现和蓝牙 2.1 设备配对
- SPP 协议：该模块支持蓝牙 SPP 协议 2.1。
- GATT 协议：该模块支持蓝牙 GATT 协议 4.0。

5.14.1. 连接管理

在 [vmbt_cm.h](#) 里定义的连接管理是您需要蓝牙服务的第一个 API。该模块需要调用：

- `vm_bt_cm_init`

以下 API 让您打开和关闭蓝牙模块：

- `vm_bt_cm_switch_on`
- `vm_bt_cm_switch_off`

一旦设备准备好了之后，应用可以设定主机和能见度如下：

- `vm_bt_cm_set_host_name`
- `vm_bt_cm_set_visibility`

该模块还管理蓝牙 2.x 设备发现和配对。请注意蓝牙 4.0 设备不是通过该模块发现和配对 – 您应使用 GATT 协议模块来代替。

5.14.2. SPP 协议

蓝牙 SPP 协议支持一对一的连接，它的 API 在 [vmbt_spp.h](#) 定义。该模块让应用与其他支持蓝牙 2.1 SPP 的设备连接并交换数据。应用可以设定硬件成为伺服或客户端。一旦蓝牙设备设定好连接管理，就可以作以下的动作：

- 调用 `vm_bt_spp_open` 访问 SPP 模块并分配内存给模块。
- 调用 `vm_bt_spp_bind` 设定伺服端。
- 或者使用连接管理 API `vm_bt_cm_search` 来发现附近的 SPP 伺服并用 `vm_bt_spp_connect` 连接上。

请参阅 `BT_SPP_Server` 和 `BT_SPP_Client` 范例使用。

5.14.3. GATT 协议

SDK 提供蓝牙 GATT 4.0 协议。与 SPP 不同的是，GATT 协议是个一般的规格并让您在蓝牙 4.0 连接时发送和接收短的数据名称“属性”。因此您的应用应该在 GATT 协议服务里定义其他蓝牙 4.0 协议以确保一个明确的用意。

以下是 LinkIt Assist 2502 设备和蓝牙设备的角色分配和职责：

- 中央和外部。在中央的设备将扫描广播，而外部的设备发送广播。
- GATT 伺服端和客户端，这将决定两个设备连接上后的沟通方式。

所有的 GATT 模块中的 API 是异步的。这代表所有的功能是由个请求传递至基础框架内而执行。执行结果将从回调传回。因此，开发者通过注册一组回调函数来初始化 GATT 服务和客户模块，每个回调代表 GATT 服务器和客户端模式的事件。

调用 `vm_bt_gatt_server_register` 来开启一个 GATT 外部设备，伺服器，结构 `vm_bt_gatt_server_callback_t`。在结构里应用必须定义回调给不同的 GATT 伺服事件。例如启动服务，特征描述和定义，以及传入的连接。请参见图 63 范例了解 API 回调顺序。

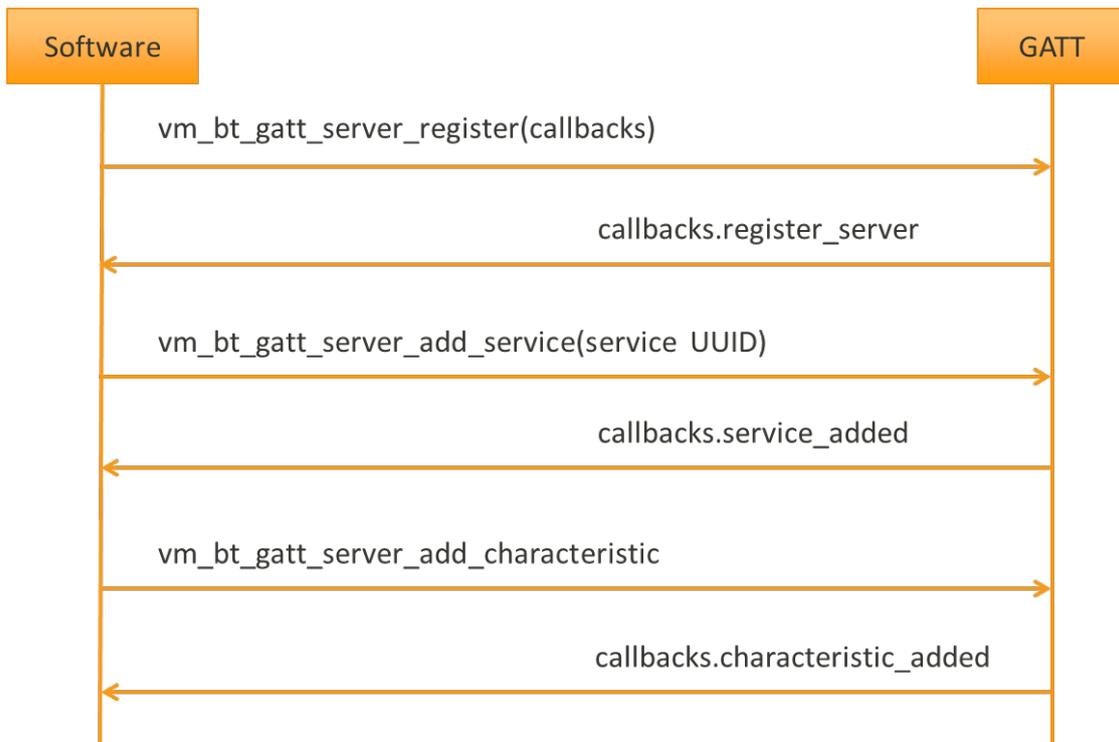


图 63 GATT 回调顺序图

同样地，应用应该调用 `vm_bt_gatt_client_register` 开启一个 GATT 中央/客户并有一个 GATT 客户事件回调表包含注册，扫描，连接，读/写特征为相对的服务，读/写特征。

请参考以下 SDK 范例了解 GATT API 调用步骤：

- BT_BLE_GATT_Client : GATT 客户端范例。
- BT_BLE_GATT_Server: GATT 伺服器端范例。
- BT_BLE_Profile_BAS : 创建 蓝牙 4.0 电池 服务 范例 (<https://developer.bluetooth.org/TechnologyOverview/Pages/BAS.aspx>) GATT 伺服器 API。
- 创建蓝牙信息服务设备范例，这是一个常用的蓝牙 4.0 协议，GATT 伺服器 API。
- BT_BLE_ReadRSSI: 扫描附近的蓝牙 4.0 设备并发表其 RSSI 强度。

5.15. 音频

音频模块提供以下子模块：

- 音频：一般音频功能例如控制音量和注册事件回调功能。 [vmaudio.h](#) 内定义。
- 音频播放：播放 MP3, AAC, WAV, AMR, 和 MIDI 格式文件. 定义于 [vmaudio_play.h](#)
- 音频录音：录音扩音器输入至 WAV 和 AMR 格式文件。在 [vmaudio_record.h](#) 定义。
- 音频流播放：允许应用播放内存数据 PCM, ARM, AAC, MP3 格式。在 [vmaudio_stream_play.h](#) 定义。

音频播放模块是相当简单的使用，请参阅 [vmaudio_play.h](#) 的接口定义。只有一个文件可以同时播放，音频资源是音频模块和电话模块之间共享，因此音频播放将可能由电话呼叫被中断。

音频记录模块记录麦克风输入到一个单一的文件。并支持 WAV 和 AMR 文件格式。需要注意的是，记录过程中访问原始音频数据是没有界面的。

音频流播放模块只支持播放。应用程序提供流在内存缓存数据至模块以 PCM/ AMC/ AAC/ MP3 格式。该模块将消耗数据和发送 `VM_AUDIO_STREAM_PLAY_EVENT_DATA_REQUEST` 由 `vm_audio_register_interrupt_callback` 注册的回调函数。然后应用程序调用 `vm_audio_stream_play_put_data` 填补后续的数据流。如果应用程序无法提供数据，音频播放将失败。

5.16. GNSS

硬件中包含联发科技 MT3332 芯片组，例如 LinkIt Assist 2502 开发板，您就可以用 GPS API 来访问 GPS NMEA 数据。如果硬件支持多种卫星系统并指定使用多种卫星系统，驱动程序将把这些不同系统的结果合并成同个 NMEA 输出，基于以下规则：

- 不同卫星系统的 GGA 信息将被合并成 GPGGA 字串。
- GSA 和 GSV 信息是分开保存的例如：
 - GPS GSV 数据为 GPGSV
 - GLONASS GSV 数据为 GLGSV
 - GSV 数据为 BDGSV 北斗

调用 [vm_gps_open\(\)](#) 来指定您要的卫星系统和回调收到 NMEA 事件字串和参数更新。请参见 `VM_GPS_TYPE` 了解支持的卫星系统和 `VM_GPS_MESSAGE` 了解回调的信息种类细节。

一旦 GPS 系统已为当前位置定位，在选择性设定 [vm_gps_set_parameter\(\)](#) 和参数之后，在 [vm_gps_open\(\)](#) 注册的回调功能将开始收到 [VM_GPS_SENTENCE_DATA](#) 类型的数据。这时您应该解析 NMEA 字串和检索所需要的信息。

请调用 `vm_gps_close()`，来关闭 GPS 模块。

5.16.1. EPO (辅助定位)

该 LinkIt Assist 2502 开发平台里的联发科技 MT3332 芯片组支持离线 GPS 辅助定位技术 (EPO)。此功能增强并加快 GPS 定位，减少全球定位系统首次定位时间 (TTFF) 并提升用户体验。联发科技 EPO 通过 Wi-Fi 或 GPRS 设备下载卫星定位日历数据。下载的文件很小 (约 10 KB)，您也可以关闭 EPO，通过编辑 `vm_gps_set_parameters` 函数的 `VM_GPS_SET_EPO_AUTO_DOWNLOAD` 参数为 0，例如：

```
vm_gps_set_parameters(VM_GPS_SET_EPO_AUTO_DOWNLOAD, 0, user_data)
```



注：VM_GPS_SET_EPO_AUTO_DOWNLOAD 的默认值是 1，代表打开。

5.17. 图像

图像模块提供以下功能：

- 执行像素操作
- 绘制点，线，矩形和椭圆形
- 图像文件解码
- 支持联发科技串行 LCD 设备
- 使用矢量字体

图像 API 应该要在 VM_EVENT_PAINT 到系统事件柄之后再调用。并且只能在主线调用。

5.17.1. 图像和显示器

LinkIt Assist 2502 SDK 2.0 支持 JPEG / GIF / PNG / BMP 这四种图像格式。输入可由文件或者内存。图像能从嵌入文件或者资源解码。请参阅第 4.8.1 了解如何嵌入图像到应用。

LinkIt Assist2502 将画的形状，图像，字符放在框里。框里面有以下元素：

- 宽度：框里的像素宽度
- 高度：框里的像素高度
- 颜色格式：框的颜色格式，这指示一个像素需要多少 bit 和它是否有个 alpha 渠道。
- 缓存：拥有图像数据的内存。这需要由 `vm_malloc_dma()` 分配。
- 缓存长度：缓存大小，取决于框宽度，高度和颜色格式

使用图形模块之前，应用应该先设置框的正确使用。如果缓冲器的大小比显示装置还要大，它将会被裁剪。要在显示器上显示框，请先确认它的大小和颜色格式与显示设备互相匹配，并调用

[vm_graphic_blt_frame\(\)](#)，它取多达 4 个框的一排框指针数组，将它们复合，然后将传送结果到显示装置。请注意只有支持联发科技串行 LCM 界面的显示器才可以用。SPI 界面的显示器请用 DCL API 根据设备本身定义的协议来传输缓冲器。**矢量字符**

要绘制矢量字体到一个框，请调用 [vm_graphic_draw_text](#)。还有各种变异的绘图 API 而允许应用程序绘制文本根据不同的约束，例如变化基准线和边界宽度。请参阅 API 参考细节。

文本绘制是由系统默认字体，并且只支持拉丁字符。该 SDK 附带了一组字体文件

在 `custom\VectorFont` 文件夹定义，支持不同的语言。

要更改绘制文本的字体，应用程序应该分配内存资源给字符文件，再将它加载与 `vm_graphic_get_font_pool_size` 和 `vm_graphic_init_font_pool` 之后

`vm_graphic_get_font_pool_size` 将使用 WHAT TOOL ? 在 `custom\VectorFont` 里来评估字符所需要用的内存。

接下来, 应用应该调用 `vm_graphic_set_font` API, 它将把字符路径排列当输入和逐个加载。字符排列的字符有优先, 这代表系统将先尝试找到一个字的字符数据, 如果它找不到, 它将于第二顺序去寻找字符数据。这允许应用回归到不同语言的字符。

加载字符操作需要时间, 但是只有在转换字符时才需要。如果应用需要除掉所有已安装的字符并还原于内建字符, 请调用 `vm_graphic_reset_font()`。



注: 使用矢量字体时软件需要更多存储, 矢量字体需要 100KB 高速缓存, 为确保您的软件最佳状态, 请将您应用最大存储允许值减掉该高速缓存。(细节请参见第 4.5 “LinkIt Assist 2502 应用存储布局”)

5.17.2. 计算文字宽度

我们建议调用 `vm_graphic_get_text_width` 来获取字串宽度, `vm_graphic_draw_text` 来画整个字串。`vm_graphic_measure_character` 或者 `vm_graphic_get_character_width` 的返回包含无充填的个体。因此如果用了矢量字体的话, 计算单个字符的合计长度串的长度将会于不同的手机型号和版本产生不可预知和不一致的结果。

您可以使用 `vm_graphic_measure_character` 接口来同时访问字符的高度和宽度。输入字符参数可以是 ASCII 或中文的字符格式。您还可以使用 `vm_graphic_get_character_width` 接口访问 ASCII 和中文字和 `vm_graphic_get_character_height` 的宽度, 以获得的 ASCII 字符的高度。

5.18. VXP 与固件更新

LinkIt Assist 2502 提供 VXP 更新和固件更新方法, VXP 更新能使用在替代旧的 VXP, 固件更新顾名思义, 我们支持整包更新。

VXP 更新: 请在 `vmsystem.h`. 参阅 [vm pmng exit and update application](#)

文件名称就是 VXP 的完整路径, 我们只支持 DES 加密, 请不要充填 DES 并确认加过密的文件是与原先的文件一样大, 不然的话, 错误将产生或者 VXP 不会运行。如果 VXP 文件没有加密, 请设定三个参数: `key` 是 NULL, `key_length` 是 0, `encryption` 是 `VM_PMNG_ENCRYPTION_NONE`。

固件更新请参阅 `vm_firmware_trigger_update()` in `vmfirmware.h`，二进制必须是根据系统文件路径 `C:\image.bin`，而这个路径不可更改。如果更新是停电，下次更新程序会自动触发。如果固件失败，有可能 `image.bin` 有错误。`C:\update_status` 错误代码，如何读错误代码，请参考固件范例。以下五个错误供您参考：

- 1) `image.bin` 太大，不能更新，应该从新产生适合大小的图像，错误代码可能是 1107,-1106,-1108,-321。
- 2) `image.bin` 署名有错误，请将正确的 `image.bin` 上传。错误代码可能是：303,-304。
- 3) 找不到 `image.bin`, 请确认 `image.bin` 是在 `C:\ root dir` 里。错误代码可能是：701,-702。
- 4) `image.bin` MBA 与目标不匹配，请确认 MBA 是匹配的，错误代码可能是 412 至 416, 324 或 325。
- 5) `image.bin` 的 UBIN 与目标不匹配，请确认 UBIN 是匹配的，错误代码可能是：417 至 -420。

5.19. 智能手机通知

智能手机通知模块可以接收安卓和 iOS 设备的通知，从一个高层观点来看，而不用实施底层协议。

针对 iOS，ANCS 协议有支持。

针对安卓，SDK 的客制文件夹里包含智能手机通知库。还有一个安卓应用“智能设备”让您快速测试。库连接至 LinkIt Assist 2502 设备经由蓝牙 SPP 或 GATT，取决于不同的用途。

使用智能手机通知 API 之前，应用应该要允许蓝牙连接管理并设定让智能手机扫描和连接至 LinkIt Assist 2502 设备。然后 LinkIt Assist 2502 设备就能接收以下通知：

- 信息：如果智能设备收到新短信，而该短信包含发送者号码和短信内容，它将被推向 LinkIt Assist 2502 设备上。
- 错过呼叫：如果智能设备错过一个呼叫，它将会把信息传送给 LinkIt Assist2502 设备。
- 电池电量不足警告：智能设备将发送一个警告讯息至 LinkIt Assist2502。
- 其他通知：包含内容和发送者（应用名称）。



注:该模块依赖于 iOS 和 Android 设备发送通知，有的通知类型可能不提供在某些平台上。

请参阅更多信息在[联发科技 LinkIt 智能通知开发者指南](#)。