

使用 TI Real Time Localization System - Connection Monitor 方案设计无钥匙进入 (PEPS) 系统



作者：Island Wei

摘要

本文以汽车感知应用中广泛使用的 [CC2642-Q1](#) 和 [CC2340-Q1](#) 为平台展示 TI 的 RTLS Toolbox 以及基于连接获取 RSSI 的 Connection Monitor 解决方案，该方案广泛应用于智能汽车的无钥匙进入系统设计。本文有以下内容：

1. RTLS-Connection Monitor 方案介绍
2. 使用 TI 开发板搭建 RTLS-CM 示例
3. 如何基于 RTLS-CM 开发自己的示例以及常用架构设计

RTLS-CM 方案介绍

RTLS (Real Time Localization System) 是以 SimpleLink 低功耗蓝牙无线设备平台为基础的实时定位测距技术集合，RTLS Toolbox 通过 SimpleLink SDK 开放给客户。RTLS 为定位算法提供了原始数据，接收信号强度 (RSSI) 和到达角检测 (AoA , Angel of Arrival) 。本文主要对基于 RSSI 的连接监听技术进行说明，并不涉及 AoA 方案 (有兴趣的同学可以自行查看 SDK 文档对于 AoA 的说明，或者 BLE Core Specification Version 5.1 对于 AoA over BLE Radio 规范) 。

使用 RSSI 和空间关系来进行定位的方案有很多，连接监听 (Connection Monitor , CM) 是为了同步得到可靠 RSSI 的一种方案名称，同时 CM 也是 CM 方案利用 RSSI 进行定位的一个逻辑角色，在 CM 方案中与 CM 相对应的另外两个逻辑角色是 RTLS Coordinator 和 RTLS Responder。总的来说，Connection Monitor 通过监听 Coordinator 和 Responder 之间的 BLE 连接获取 RSSI 来辅助定位算法从而对车主位置进行感知。

因为 Coordinator 和 Responder 间的 BLE 连接信息已知且唯一，CM 方案解决了基于非连接多锚点 RSSI 不统一的问题并且规避了因为部署结构和射频设计生产等环节引入的误差，极大程度的提升了基于 RSSI 定位方案的鲁棒性。

由于 BLE 核心规范并没有支持对 BLE 连接 (ACL Connection) 的监听做出任何的规范和建议，所以虽然 CM 会使用 BLE 射频部分，但在 TI 的软件中，CM 的实现并非基于 TI BLE5Stack 标准蓝牙协议栈实现，而是基于 Micro BLE 协议栈。

通常，一个比较基础的部署方案是这样的：RTLS Responder 为无线车钥匙或者手机，RTLS Coordinator 为车身中的主节点 (因为主节点一般都还承担着其他的一些任务，所以在现实中为资源能力相对较高的平台，如 [CC2642-Q1](#)) ，CM 为车身四周锚点 (可以选择与主节点相同的方案 [CC2642-Q1](#) 作为 CM，也可以选用高性价比平台 [CC2340-Q1](#)) ，一般位于车前大灯附近以及车尾两侧。通过各个点位得到的 RSSI 值和点位之间既定的三角关系，感知钥匙 (即车主) 与车身的距离和位置关系，从而提供一些智能服务，比如当车主接近时自动打开车灯，空调等智能应用。

使用 TI 开发板搭建 RTLS-CM 示例

本文实验以 [CC2642R2](#) 为 Coordinator，[CC2340R5](#) 为 Connection Monitor 为例来搭建 CM demo。本文并不准备赘述 CCS 的使用方法，专注于 RTLS-CM 方案示例的搭建上。如果您对 TI 的开发环境不熟悉，请参考：[Code Composer Studio 用户指南](#)。

硬件需求：

- 1x [LP-CC2642R2](#)
- 1x [LP-CC2340R5](#)
- 1x [LP-XDS110](#)
- 支持 BLE 功能的手机

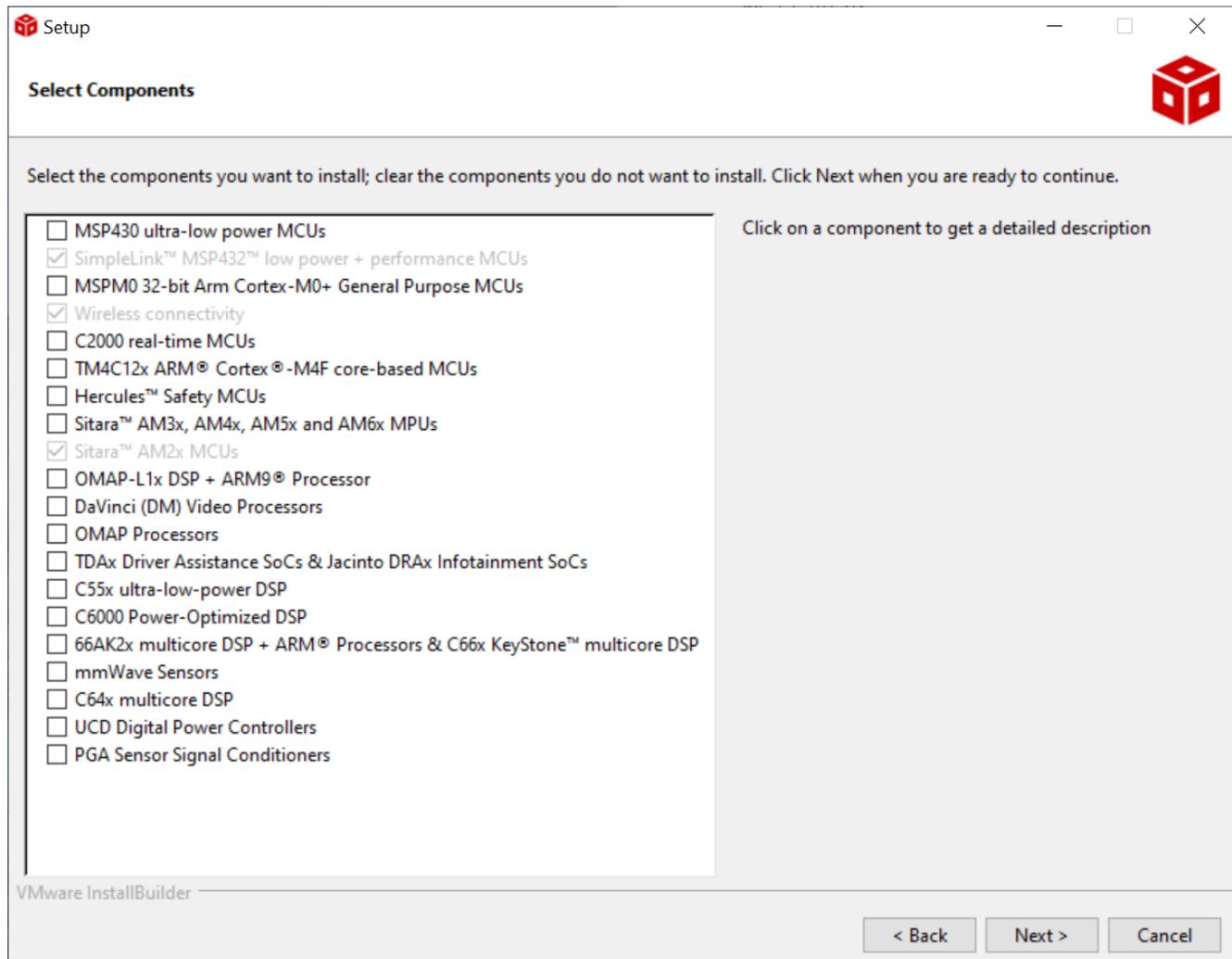
软件需求：

- CCS 集成开发环境：[\(CCSTUDIO IDE, configuration, compiler or debugger | TI.com\)](#)
- 编译工具链：[ARM-CGT-CLANG IDE, configuration, compiler or debugger | TI.com](#)
- SDK for CC2642: [SIMPLELINK-LOWPOWER-F2-SDK Software development kit \(SDK\) | TI.com](#)
- SDK for CC2340: [SIMPLELINK-LOWPOWER-F3-SDK Software development kit \(SDK\) | TI.com](#)
- Google Chrome 浏览器

RTLS-CM 使用方法：

安装软件

按提示安装 CCS。在 Setup Type 页面选择 Custom Installation，在 Device Support 页面勾选 Wireless connectivity 进行安装，Debug Probes 页面选择 Spectrum Digital Debug Probes and Boards。成功安装 CCS 后，以默认路径和选项安装 SDK 和 编译器。



工作空间和导入工程

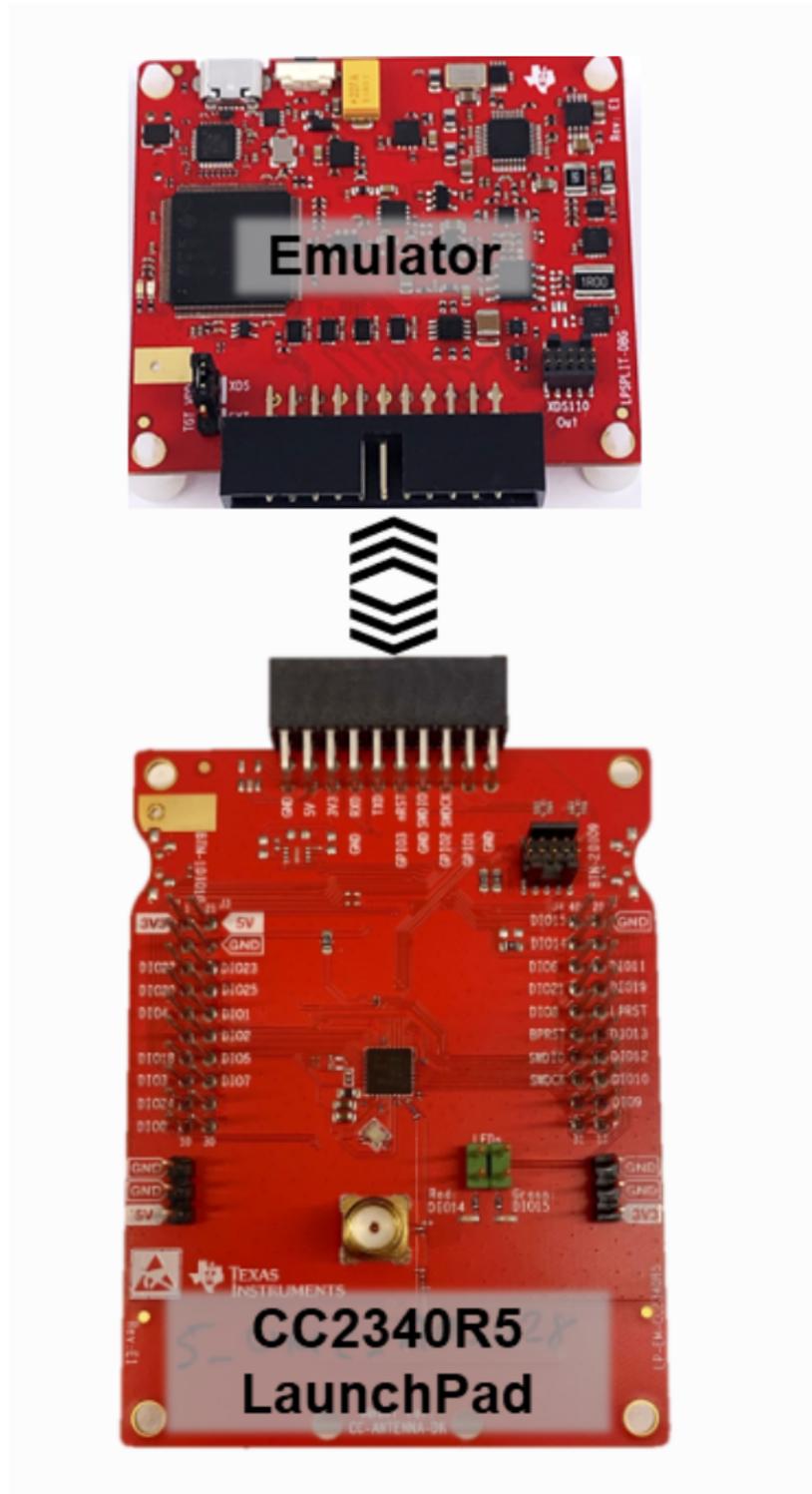
打开 CCS 创建 workspace，并从 F2 SDK 导入 *rtls_coordinator*，从 F3 SDK 导入 *connection_monitor* 例程。例程路径分别为：

`${F2_SDK_PATH}\examples\rtos\CC26X2R1_LAUNCHXL\ble5stack\rtls_coordinator`

`${F3_SDK_PATH}\examples\rtos\LP_EM_CC2340R5\ble5stack\connection_monitor`

编译和烧录

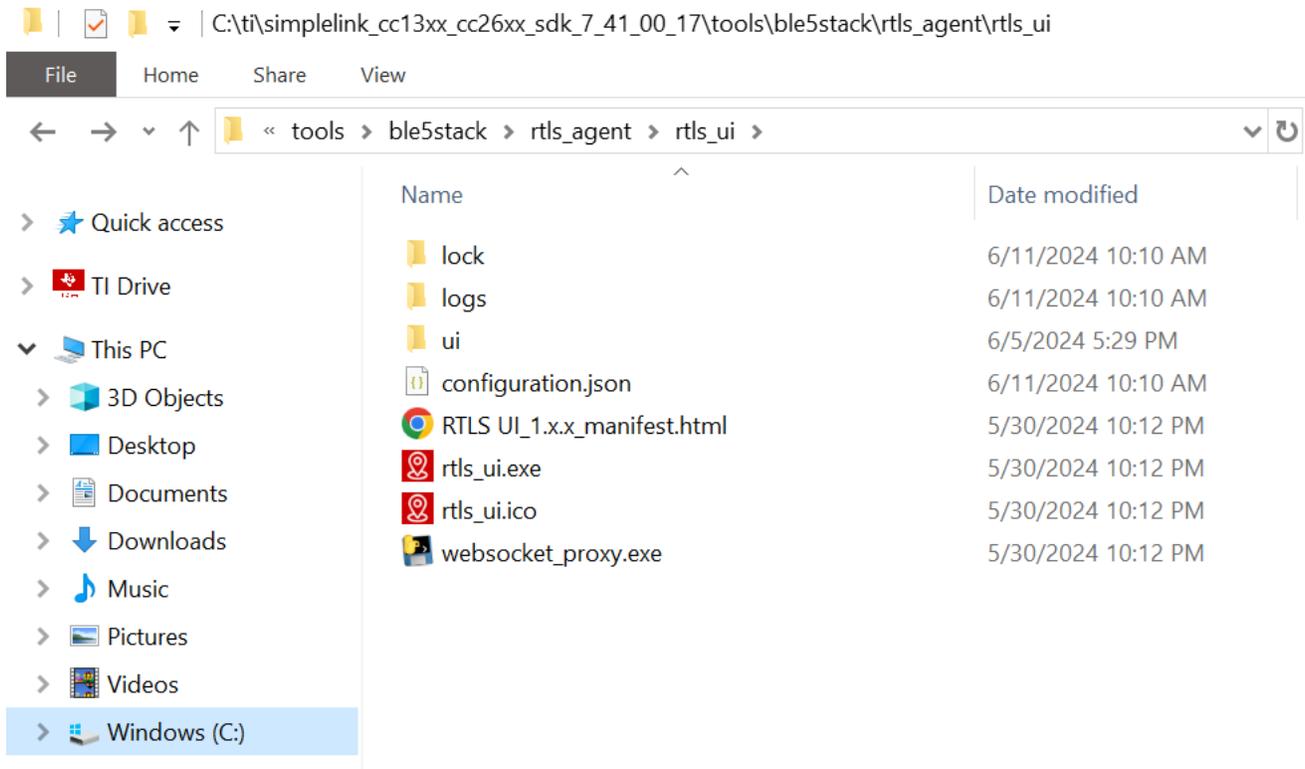
导入完成后编译两个例程。并将 CC26x2R2 开发板通过 USB 口连接电脑，按图示连接 LP-XDS110 和 CC2340 开发板，然后连接 USB 口到电脑。



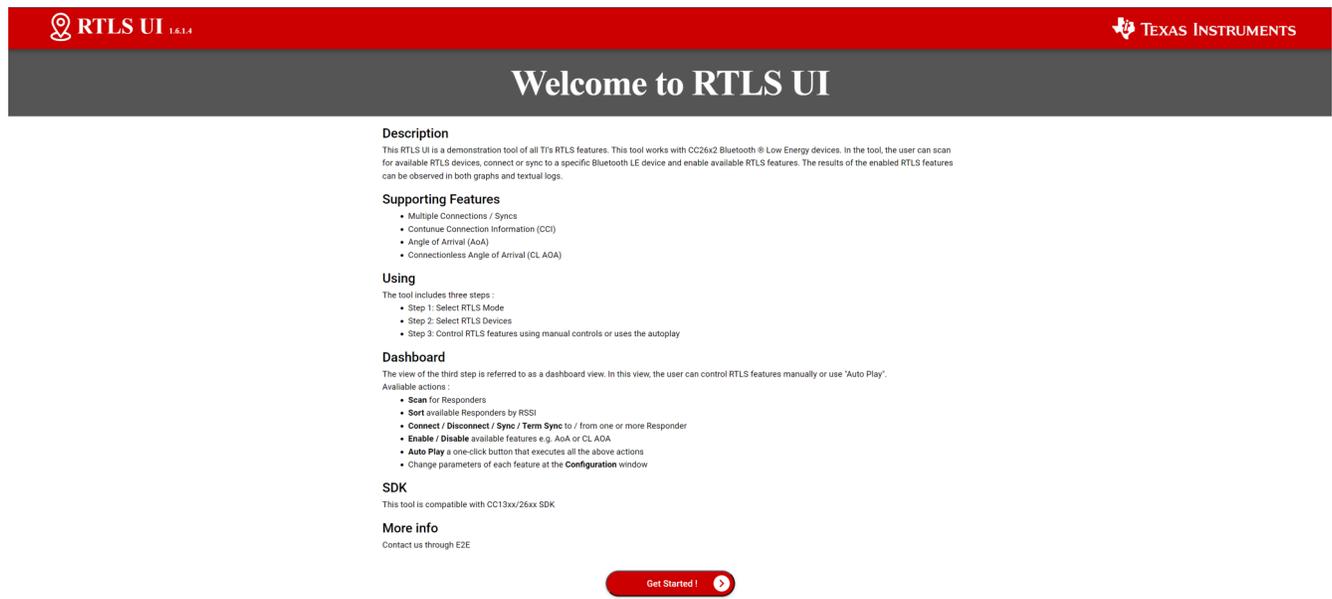
给 CC2642 开发板烧录 `rtls_coordinator` 例程，给 CC2340 烧录 `connection_monitor`。烧录可以使用 [Uniflash](#) 软件或直接使用 [CCS](#)，请参考 [Uniflash 使用指南](#)和 [CCS 使用指南](#)。烧录成功后请按一下开发板上的 `reset` 按键启动程序。

打开 RTLS 工具箱

在 `#{F2_SDK_PATH}\tools\ble5stack\rtls_agent\rtls_ui` 文件夹中，双击 `rtls_ui.exe` 打开软件：



该软件会调用 **Google Chrome** 来进行演示，请提前下载 **Google Chrome 浏览器**。启动成功后，您将会看到对 RTLS UI 工具的介绍，点击 **Get Started!** 进入模式选择：



前面我们已经介绍过 **CM** 方案是基于连接的，所以这里选择 **RTLS Connected** 并点击 **CONTINUE**。

Select RTLS Mode

RTLS Connected

RTLS Connectionless

CONTINUE

在这里您应该看到连接电脑的 CC2642 和 CC2340，点击 *Select All -> Continue* 进入页面。

Select Devices

Select All Select None

Port:	COM25
Bluetooth Address:	80:6F:80:EE:BB:F7
Device:	CC26X2
RTLS:	COORDINATOR
AoA Support:	No
CL AoA Support:	No

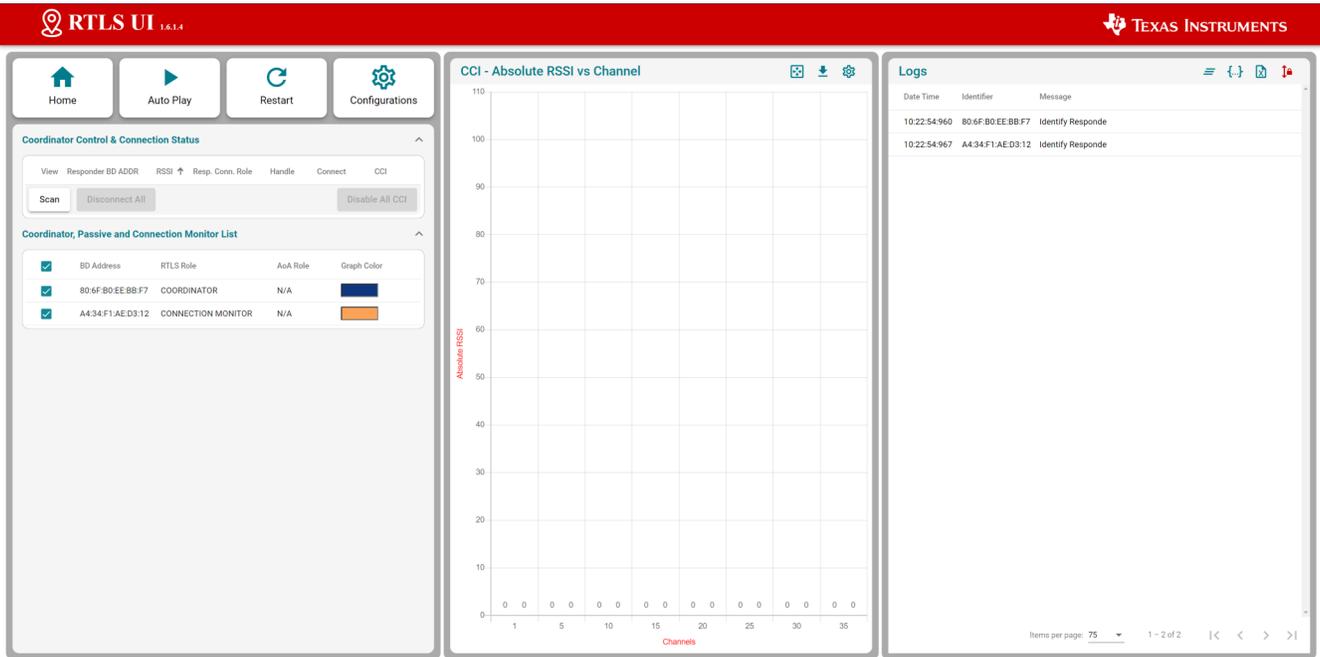
Port:	COM71
Bluetooth Address:	A4:34:F1:AE:D3:12
Device:	CC23X0R5
RTLS:	CONNECTION MONITOR
AoA Support:	No
CL AoA Support:	No

Refresh

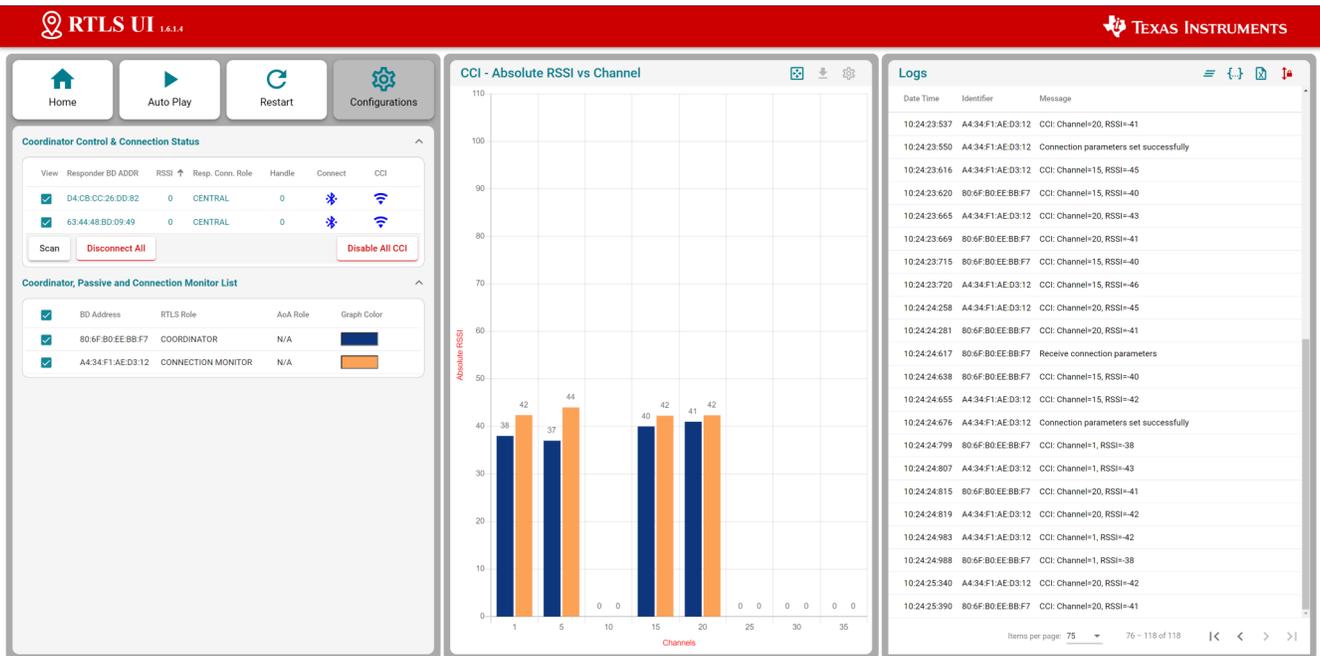
Identify

CONTINUE

在页面中您将看到左侧的设备列表 (*Coordinator* 和 *Connection Monitor*)，中间的 RSSI 显示区域以及右侧的 log 输出部分。

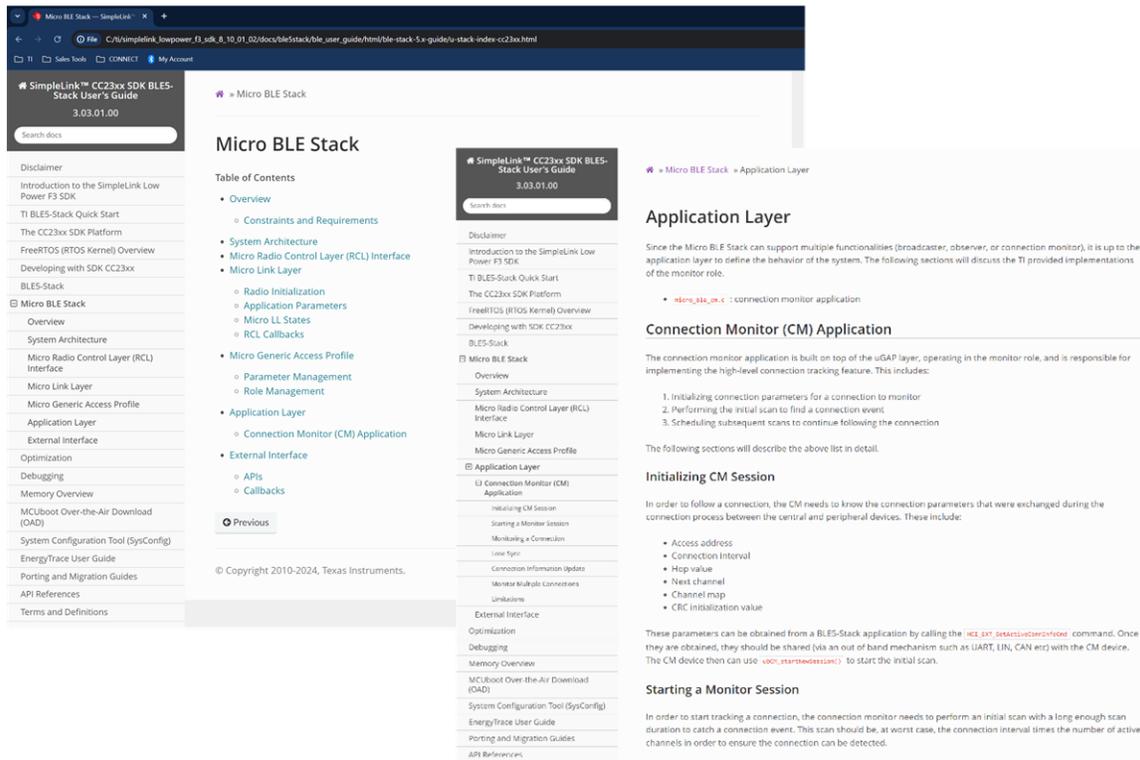


此时，请使用手机，打开蓝牙连接界面搜索 *RTLSCoordinator* 设备，连接并配对，默认配对码为 *123456*。可以看到界面左侧的 *RTLS_Coordinator* 项目的蓝牙标志从灰色变为蓝色，连接之后的任意时刻，都可以点击 *connection_monitor* 设备条目最右侧的 *CCI* 按钮开始监听某个 *Responder* 和 *Coordinator* 之间的连接，此时，RSSI 显示区域就应该有两组 RSSI (蓝色和橙色，蓝色为 *Coordinator* 得到的 RSSI，橙色为 *Connection Monitor* 得到的 RSSI) 在不同的 *channel* 显示并动态变化。



客户基于 RTLS 开发自己的示例

如果您希望基于 TI RTLS-CM 开发真实场景下基于 CAN 总线通信的样机，那么需要您在代码上添加 CAN 总线相关逻辑时，参考 SDK 文档中的 [MicroBLE](#) 和 [Connection Monitor](#) 部分，另外还有[无钥匙进入及启动 \(PEPS\) 系统设计资源](#) 可以帮助大家快速了解 PEPS 的方案介绍。



另外，根据真实的开发过程，本文对开发者进行以下提示：

1. 匹配真实的总线延时

Connection Monitor 的监听原理：CM 节点通过得到连接信息和时间戳，控制自己的 BLE 射频模块在下一个最接近的 BLE 连接事件对应的信道 (Channel) 上扫描 $(N+1) * \text{连接间隔}$ ，来确保一定能在该信道“听”到 coordinator 和 responder 之间的 BLE 连接包。我们知道真实的环境下是没有电脑和 RTLS log 界面通知你 CM 节点已经拿到连接信息了，而是 CM 节点通过私有 CAN 总线和 Coordinator 连接，通过 CAN 总线得到 BLE 连接信息，并产生对应事件来通知应用层可以开始监听。通过前面的介绍可以知道，Connection Monitor 对时间非常敏感，所以通过 CAN 总线的信息传输延时必须要考虑在内。在 `micro_ble_cm.c` 文件中定义了总线延迟：

```
// Bus latency - used to calculate number of connection events to skip
#define BUS_LATENCY_IN_MS 320
```

在 RTLS 的示例中，有一个 BUS LATENCY 的值为 320 (ms)，这个值对应的延时路径是 `rtls-coordinator application--> UART(coordinator 转 USB 连接电脑)--> Python--> UART (connection_monitor 转 USB 连接电脑)--> CM Application`，那么客户在搭建自己的真实 demo 时要将延时重新匹配自己的延时路径。举个例子，`CC2642 (Coordinator, BLE connection Info-> SPI) -> TCAN4550 (SPI -> CAN -> SPI) -> CC2642(SPI -> Connection Monitor)`，在我的实验中 BUS LATENCY 应该设置为 700ms，客户应按照自己的总线传输延时来评估 BUS_LATENCY 的数值，且这个数值应该在方案定下之后作为系统固定值，不应更改。

2. 物理层要求

CM 方案对 Coordinator 和 Responder 间的 BLE 连接有一些要求。如果您想要基于 `basic_ble` 例程代替 `rtls_coordinator` 进行软件方案设计（不建议这么做），那么请遵照以下限制利用 `sysconfig` 对 `basic_ble` 进行配置，具体如下：

- 必须使用 LE Channel Selection Algorithm #1
- 必须使用 LE 1M PHY
- 必须拥有 LE Data channel

3. 扩展广播类型

如果您的示例中需要使用 BLE 5 定义的 Ext ADV，请将 Ext ADV 设置为 Scannable，不要将其设置为 Connectable 属性。避免 Responder 和 Coordinator 使用 AUX_CONN_REQ 流程连接，这会使得 CM 无法监听到可解析的 BLE 射频数据交互。

车型匹配总结

目前 TI RTLS-CM 方案已经广泛量产于新能源汽车、插电混动以及增程式车型中，目前常用架构都是 1+N 或者 2+N，也就是一主多从，或者二主多从的结构。方案和点位数量选择主要取决于车身结构和大小。

在这里给出几个参考架构：

- A 级轿车或两厢车：1*主节点 CC2642-Q1 + 3*锚点 CC2642-Q1
 - 主节点位于两个 前排中心位置，略靠近驾驶员；
 - 前锚点位于左大灯附近，后锚点位于两个尾灯。
- 小型 SUV：1*主节点 CC2642-Q1 + 4* 锚点 CC2340-Q1
 - 主节点位于前排中心位置，略微靠近驾驶员位；
 - 前锚点位于左大灯附近，后锚点位于两个尾灯。
- B 级轿车或中大型 SUV：2*主节点 CC2642-Q1 + 4* 锚点 CC2340-Q1
 - 前主节点位于前排中心位置，后主节点位于后排中心位置；
 - 前锚点位于前大灯附近，后锚点位于尾灯附近。

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 [TI 的销售条款](#) 或 [ti.com](#) 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

Copyright © 2024，德州仪器 (TI) 公司