

作者：恩智浦半导体

1 介绍

接收信号强度指示 (RSSI) 用于确定无线信号的质量。它可以用于定位系统。通过建立信号衰减模型，可以得到信号空间路径损耗与距离的关系。可以使用链路损耗方程计算距离，并且可以通过将距离与软件算法相结合来评估位置。

本文档介绍了基于低功耗蓝牙RSSI测距的定位。它使用 Kinetis KW38 无线 MCU 实现了一个简单的三边定位系统。

2 RSSI测距

2.1 事先定义

标签 - 必须定位的未知位置的目标。在本应用笔记中，它是一块 KW38 板。

锚点 - 固定位置的设备。它收集 RSSI 值并将它们上传到位置计算设备。通常，定位至少需要三个锚点。

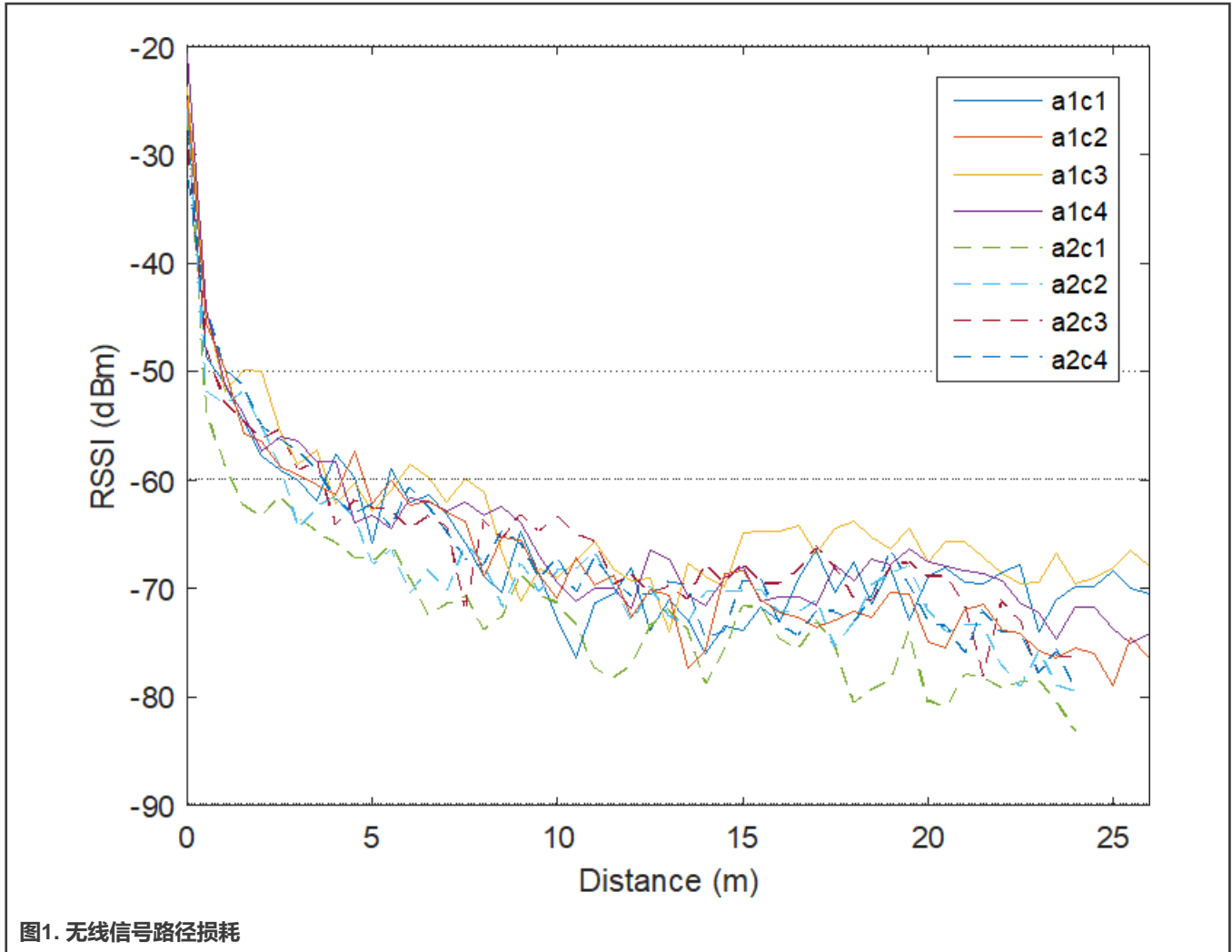
2.2 无线信号路径损耗模型

在定位系统，基于RSSI测距，标签节点应用RSSI测量使用已知的信号传播模型来估计从锚节点其距离。空间无线信号路径损耗模型如图 1所示 (a1c1, a1c2等为测试设备编号)。

目录

1 介绍	1
2 RSSI测距	1
3 基于RSSI测距的三边定位	3
4 本地化系统实施注意事项	5
5 设置和测试结果	14
6 结论	21





渐变模型广泛用于对无线信号传播损耗进行建模，其表达式如下：

$$P_r(d) = P_r(d_0) - 10n \log_{10} \left(\frac{d}{d_0} \right) + \alpha_\sigma$$

- d 和 d_0 表示实际距离和参考距离。
- $P_r(d)$ 和 $P_r(d_0)$ 分别表示在实际距离和参考距离处接收到的RSSI (dBm)。
- n 是路径损耗指数。
- α_σ 是随机噪声。始终假定具有零均值和方差的高斯分布随机变量。如图 1 所示，距离越大，接收到的RSSI值的波动越大。

对于大多数应用， d_0 为1米， $P_r(d_0)$ 由自由空间路径损耗公式计算。所以简化的渐变模型使用如下：

$$\text{RSSI} = A - 10 n \lg d,$$

- d 是距离。
- n 由测试环境决定，通常是通过测试收集的经验常数。
- RSSI 是在距离 d 处接收到的 RSSI 值。

- A 是接收器从一米外的发射器接收到的RSSI。

该公式用于基于RSSI的估计距离。

2.3 对RSSI接收精度的影响

2.3.1 环境

由于环境中存在多径、障碍物、电磁干扰等不稳定因素，接收节点上的RSSI值有时不稳定，有建设性和破坏性叠加。过滤算法可用于过滤不准确的RSSI值，如[过滤RSSI](#)中所述。

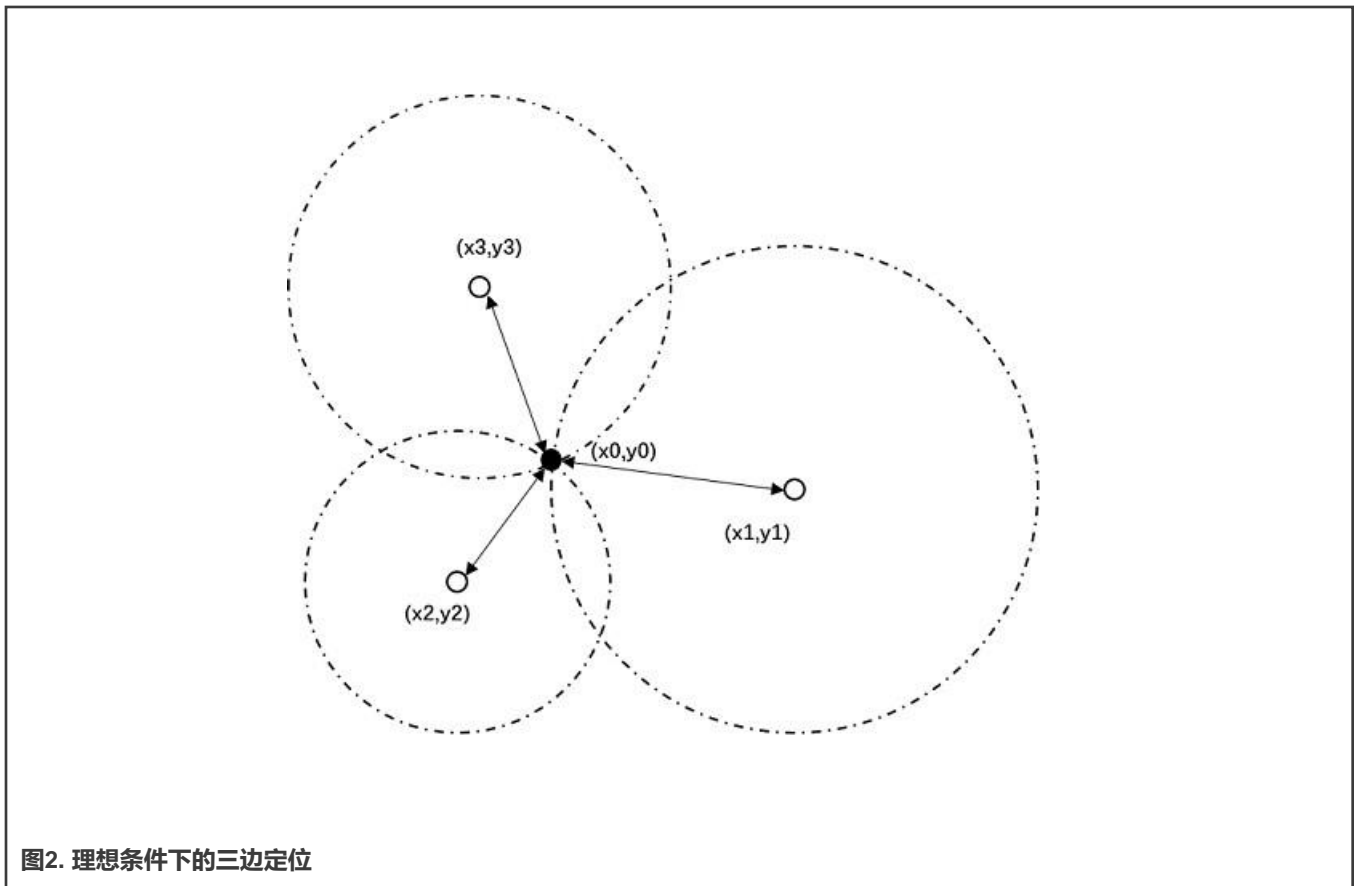
2.3.2 天线

不同的天线有不同的辐射模式、极化方向和信号增益。在相同条件下，接收端和发射端使用不同的天线也会影响RSSI。

3 基于RSSI测距的三边定位

3.1 三边定位

最少三个锚节点就足以定位一个标签节点。三边定位如图2所示。



- (x_0, y_0) 是要确定的标签板坐标。
- (x_1, y_1) , (x_2, y_2) , (x_3, y_3) 是三个已知的锚板坐标。

- 三个圆的半径是从标签到每个锚点的距离。可以用[这个公式](#)来计算。Tag 的坐标可以通过以下方程组计算（D 是 Tag 板到 Anchor 板的距离）。

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = D_1^2$$

$$(x_2 - x_0)^2 + (y_2 - y_0)^2 = D_2^2$$

$$(x_3 - x_0)^2 + (y_3 - y_0)^2 = D_3^2$$

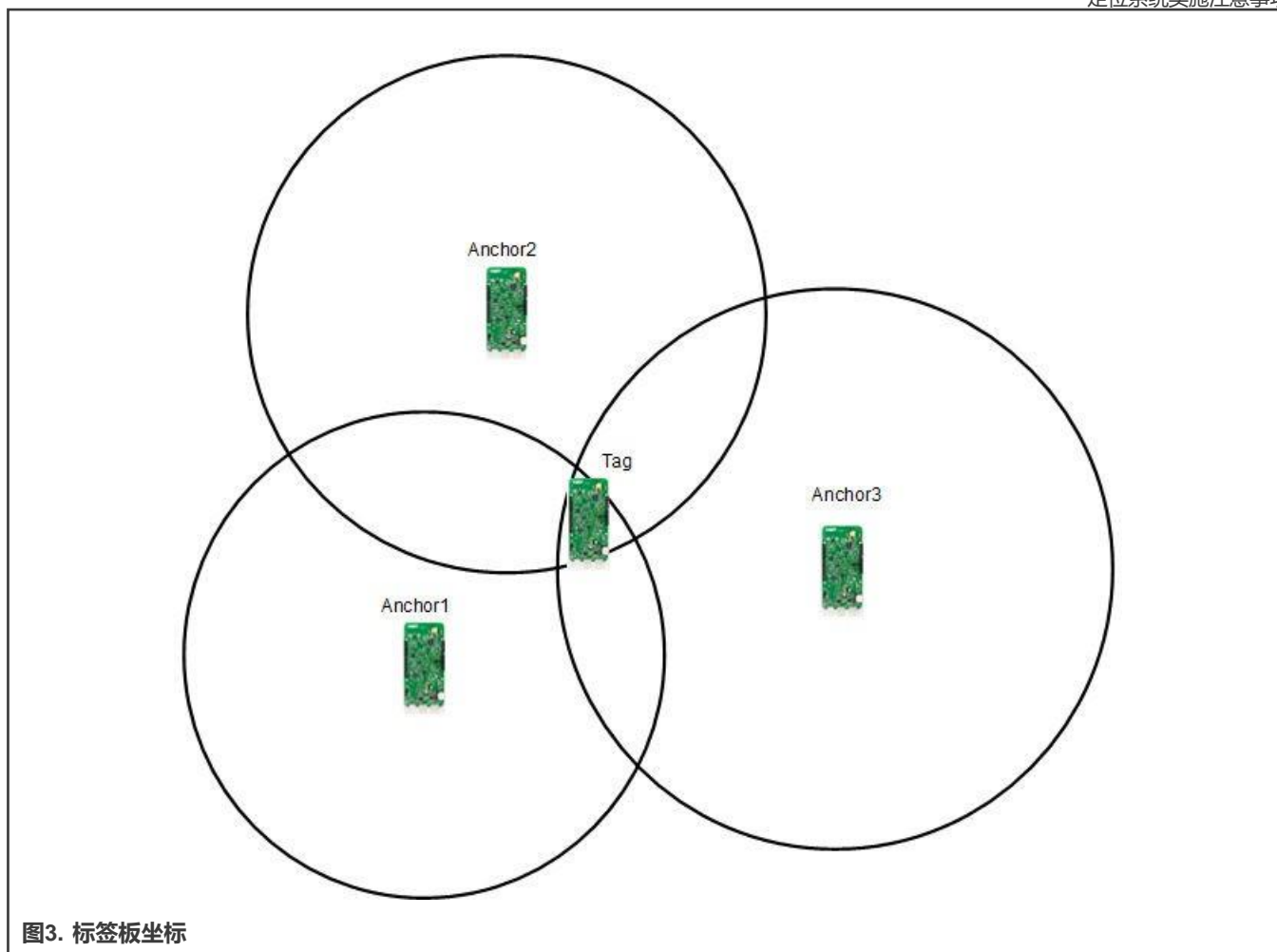
由于测量误差，三个圆很少在一点相交，如图 3 所示。最小二乘法被认为是获得最优解的最佳方法。方程组转换为矩阵形式：

$$B = \begin{pmatrix} x_1^2 + y_1^2 - x_n^2 - y_n^2 - d_1^2 + d_n^2 \\ x_2^2 + y_2^2 - x_n^2 - y_n^2 - d_2^2 + d_n^2 \\ \vdots \\ x_{n-1}^2 + y_{n-1}^2 - x_n^2 - y_n^2 - d_{n-1}^2 + d_n^2 \end{pmatrix};$$

$$X = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}; \quad A = \begin{pmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \vdots & \vdots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{pmatrix}$$

使用最小二乘法方程计算标签板坐标：

$$X = A^T B / (A^T A)$$



4 定位系统实施注意事项

4.1 定位方案及硬件要求

有两种定位方案，如以下部分所述。

4.1.1 定位方案——广告

定位方案如图 4 所示。RSSI 的获取参见[从广告中获取 RSSI](#)。

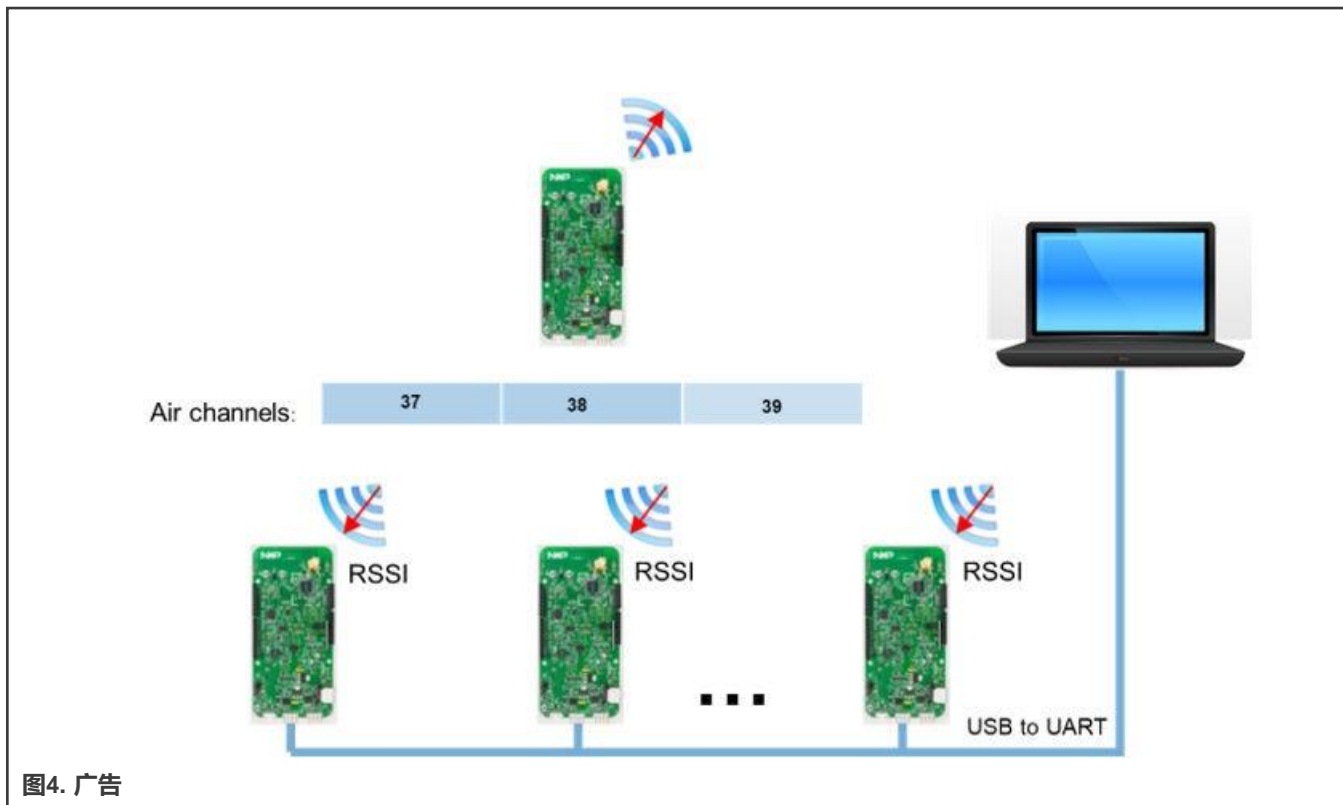


图4. 广告

硬件先决条件如下：

- 三块FRDM-KW38板作为要扫描的锚板
- 一块FRDM-KW38板作为标签板做广告
- 四根Micro-USB电缆
- 四个单极天线
- 一台安装了GUI工具的个人计算机

4.1.2 定位方案——连接

定位方案如图 5所示。RSSI的获取参见[从连接中获取RSSI](#)。

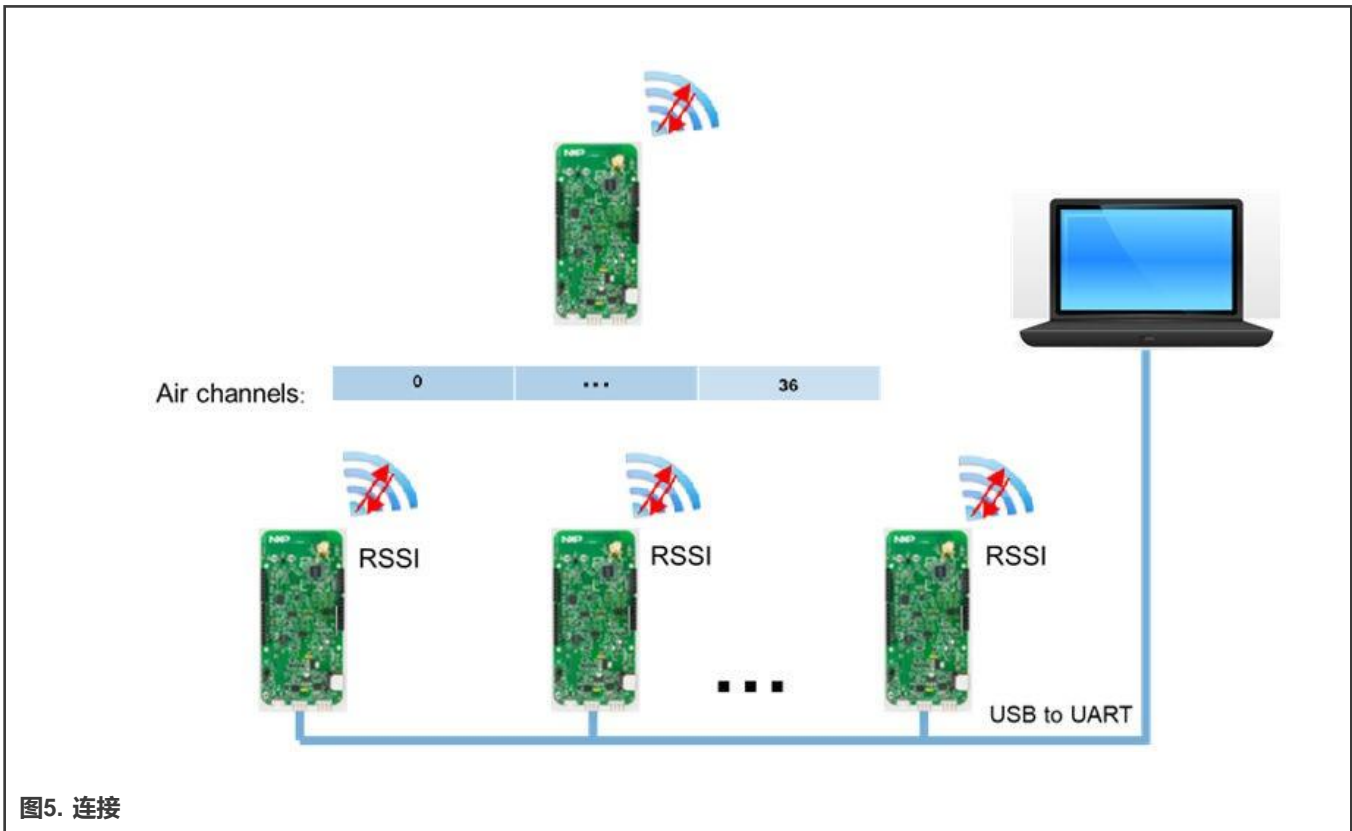


图5. 连接

- 三块FRDM-KW38板作为锚板等待连接
- 一块FRDM-KW38板作为标签板与锚点连接
- 四条Micro-USB电缆
- 四个单极天线
- 一台安装了GUI工具的个人计算机

4.2 考虑嵌入式软件

4.2.1 低功耗蓝牙项目

KW38 SDK 提供了无线 UART 项目，它通过蓝牙 LE 控制器实现 RSSI 通知。KW38 蓝牙 LE 控制器支持与扫描事件中的通道号相关联的 RSSI 通知和连接事件中数据包中的 RSSI 报告。

项目路径为 `SDK\boards\frdmkw38\wireless_examples\bluetooth\w_uart`

您只需启用 `gUseControllerNotifications_c` 宏。`gControllerNotificationEvent_c` 事件在 `BleApp_GenericCallback()` 处报告。选择哪个无线数据包的类型，应该通过 `Gap_ControllerEnhancedNotification()` API 函数通知给应用 RSSI（接收时）。

注意

有关无线 UART 的更多信息，请参阅低功耗蓝牙演示应用程序用户指南（文档 [BLEDAUG](#)）。

4.2.2 从广告中获取RSSI

锚板和标签板均使用无线 UART 进行编程。区别在于锚板作为中央设备扫描广告，标签板作为发送广告的外围设备。

4.2.2.1 配置通知类型

在中心侧，仅通过 `Gap_ControllerEnhancedNotification()` API 函数配置 Scanning ADV PKT Rx 发送的通知。

4.2.2.2 连续扫描和地址过滤

默认情况下，无线 UART 充当中央设备。它搜索要连接的其他无线 UART 设备。在找到设备之前，它会扫描所有广告并将 RSSI 值通知给应用程序。

持续扫描预期设备的广告，不连接预期设备，将设备地址添加到白名单，用白名单扫描。

```

***** */
static void BleApp_ScanningCallback(gapScanningEvent_t *pScanningEvent)
{
    switch (pScanningEvent->eventType)
    {
        case gDeviceScanned_c:
        {
            if (BleApp_CheckScanEvent(&pScanningEvent->eventData.scannedDevice))
            {
                gConnReqParams.peerAddressType = pScanningEvent->eventData.scannedDevice.addressType;
                FLib_MemCpy(gConnReqParams.peerAddress,
                    pScanningEvent->eventData.scannedDevice.aAddress,
                    sizeof(bleDeviceAddress_t));
            }

            #if (JIA_TEST == 1)
            if (mAppAlreadyScannedDevice == FALSE)
            {
                Gap_StopScanning();
                Gap_AddDeviceToWhiteList(gConnReqParams.peerAddressType, gConnReqParams.peerAddress); // add device address to white list
                mAppAlreadyScannedDevice = TRUE;

                gScanParams.filterPolicy = gScanWithWhiteList_c; // only scan address in white list.
                (void)App_StartScanning(&gScanParams, BleApp_ScanningCallback,
                    gGapDuplicateFilteringDisable_c,
                    gGapScanContinuously_d,
                    gGapScanPeriodicDisabled_d);
            }
            #else
            (void)Gap_StopScanning();
            #if gAppUsePrivacy_d
            gConnReqParams.usePeerIdentityAddress = pScanningEvent->eventData.scannedDevice.advertisingAddressResolved;
            #endif
            (void)App_Connect(&gConnReqParams, BleApp_ConnectionCallback);
            #endif
        }
    }
}

```

图6. 使无线 UART 连续扫描预期设备的代码

此外，扫描定时器会在10秒后停止扫描。要连续扫描，请禁用计时器 `ScanningTimerCallback()` 函数。

4.2.3 从连接中获取RSSI

锚板和标签板均使用无线 UART 进行编程。区别在于锚板作为外围设备开始广播等待中心设备连接，标签板作为中心设备连接所有外围设备。

4.2.3.1 快速建立多条连接

为了在中心设备和多个外设之间快速建立连接，要在开始连接之前，将所有目标外设的公共地址添加到中心设备侧的白名单中。中心设备连接白名单中的外设，如下图所示。


```

test_peer_inf_t TEST_peer_inf[gAppMaxConnections_c] =
{
  {{0xE0,0x28,0xB9,0x37,0x60,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x8F,0x34,0x23,0x37,0x60,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x5C,0x9C,0x4A,0x37,0x60,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x00,0x00,0x00,0x00,0x00,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x00,0x00,0x00,0x00,0x00,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x00,0x00,0x00,0x00,0x00,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x00,0x00,0x00,0x00,0x00,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
  {{0x00,0x00,0x00,0x00,0x00,0x00},gInvalidDeviceId_c,TEST_PEER_DISCONN},
};

uint32_t peer_device_counter = 0;
uint32_t valid_connections = 3;

# if (gTestSlave == 1)
  mGapRole = gGapPeripheral_c;
  (void)Serial_Print(gAppSerMgrIf, "\n\rWireless UART starting as GAP Peripheral, press the role switch to change it.\n\r", gAllowToBlock_d);
#else
  mGapRole = gGapCentral_c;
  (void)Serial_Print(gAppSerMgrIf, "\n\rWireless UART starting as GAP Central, press the role switch to change it.\n\r", gAllowToBlock_d);
  uint8_t all_zero[6] = {0,0,0,0,0,0};
  for (uint8_t i = 0; i < gAppMaxConnections_c; i++)
  {
    if(FlLib_MemCmp((void *)&TEST_peer_inf[i].peer_address, &all_zero, sizeof(all_zero)))
    {
    }
    else
    {
      if(gBleSuccess_c != Gap_AddDeviceToWhiteList(gBleAddrTypePublic_c, TEST_peer_inf[i].peer_address))
      {
        Serial_Print(gAppSerMgrIf, "add to wl failed\r\n", gNoBlock_d);
      }
    }
  }
}

```

① BleApp_Config()

图7. 快速建立多个连接

```

***** */
v2d BleApp_Start(gapRole_t gapRole)
{
    switch (gapRole)
    {
        case gGapCentral_c:
        {
            (void)Serial_Print(gAppSerMgrIf, "\n\rScanning...\n\r", gAllowToBlock_d);
            mAppUartNewLine = TRUE;
        }
        #if defined(gUseControllerNotifications_c) && (gUseControllerNotifications_c)
        #if (JIA_TEST_FOR_RSSI_ADV == 1)
            //in central side, only send notification of Scanning ADV PKT RX
            Gap_ControllerEnhancedNotification(gNotifScanAdvPktRx_c , 0);
        #elif (JIA_TEST_FOR_RSSI_CON == 1)
            //in central side, try to connect directly with peers which are in whitelist
            gConnReqParams.filterPolicy = gUseWhiteList_c;
            (void)App_Connect(&gConnReqParams, BleApp_ConnectionCallback);
        #else
            Gap_ControllerEnhancedNotification(gNotifScanEventOver_c | gNotifScanAdvPktRx_c |
                gNotifScanRspRx_c | gNotifScanReqTx_c, 0);
        #endif
        #endif
        static void BleApp_ConnectionCallback(deviceId_t peerDeviceId, gapConnectionEvent_t *pConnectionEvent)
        {
            switch (pConnectionEvent->eventType) ③
            {
                case gConnEvtConnected_c:
                {
                    /* Save peer device ID */
                    maPeerInformation[peerDeviceId].deviceId = peerDeviceId;
                }
                #if (JIA_TEST_FOR_RSSI_ADV == 1)
                #elif (JIA_TEST_FOR_RSSI_CON == 1)
                if(mGapRole == gGapCentral_c)
                {
                    peer_device_counter++;
                    if(peer_device_counter == valid_connections)
                    {
                        Serial_Print(gAppSerMgrIf, "\nConnection with all slaves\r\n", gNoBlock_d);
                    }
                }
                else
                {
                    //if not connected with all peripherals device, start connect process
                    App_Connect(&gConnReqParams, BleApp_ConnectionCallback);
                }
            }
        }
    }
}

```

图8. 快速建立多个连接

4.2.3.2 外设延迟广告开始

在建立多个连接的过程中，如果多个外设同时开始广播，有时可能会导致中心设备没有与所有设备建立连接，而是与其中一些设备建立连接。解决这个问题的安全方法是延迟序列中外设的广告开始。延迟时间在 GUI 工具中设置为 100ms。

4.2.3.3 配置通知类型

在外设端，当与中心设备连接时，只配置Connection Rx PDU和Connection事件发送的通知。

4.2.3.4 连接参数

如果连接链路丢失，设备会继续发送最后接收到的连接事件的RSSI值直到监督超时断开连接。因此，监督超时值应该在一个合理的范围内，这样可以减少链路丢失导致的错误值。

4.2.4 RSSI中的过滤

在实际应用中，由于RSSI对环境敏感，需要对多个实测RSSI值进行过滤优化，得到优化后的RSSI值，然后计算位置。过滤算法可以在 KW38 MCU 上实现。图 9 比较了常用的 RSSI 过滤算法。

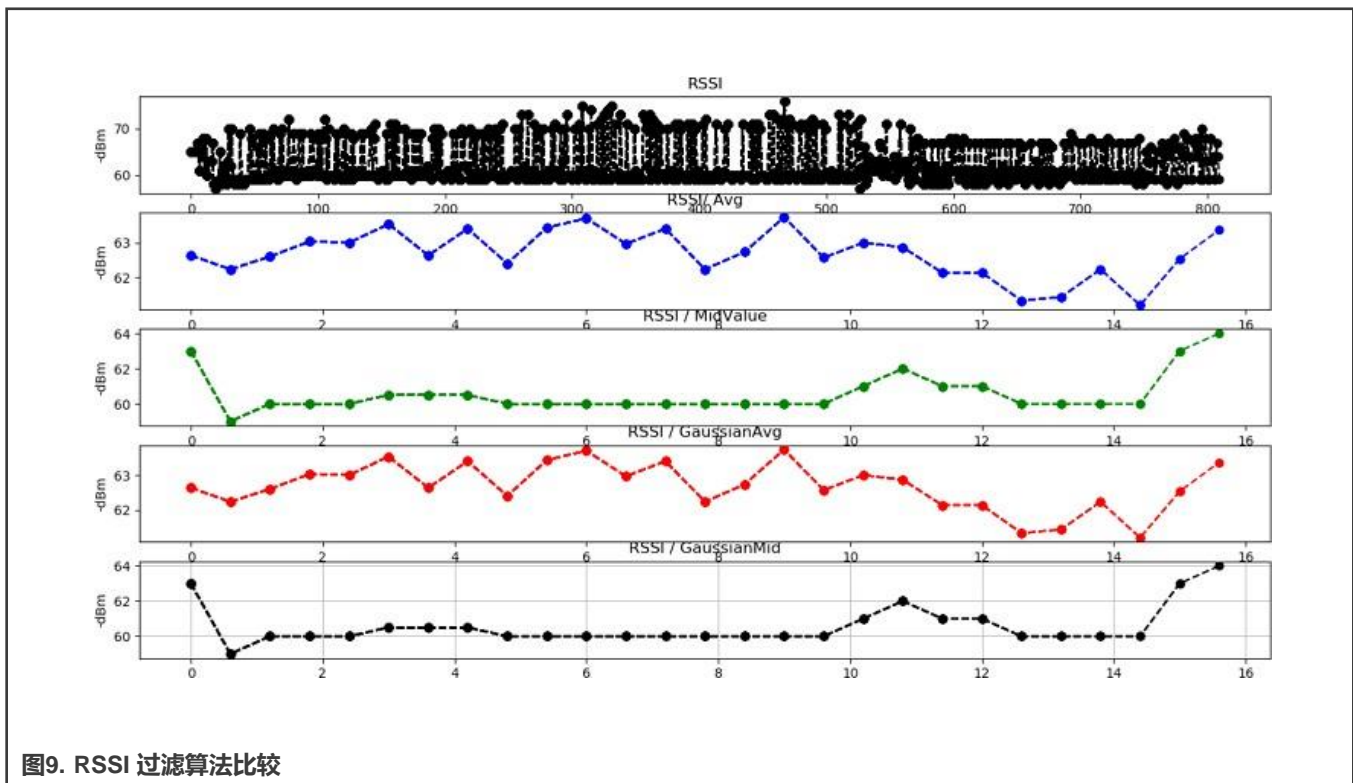


图9. RSSI 过滤算法比较

在图 9 所示的测试中，标签和锚设备放置在固定位置。锚设备收集，过滤的RSSI。第一行是RSSI 测量值，其他行是过滤输出的优化RSSI 值。在图 9 中，过滤器收集30个样本进行一次过滤。

滤波有四种类型：均值滤波、中值滤波、高斯滤波和高斯中值结合滤波。

均值滤波：计算多个 RSSI 值的平均值作为测试结果。当RSSI值波动较大时，结果的可靠性降低。

中值滤波：它根据值排列多个RSSI。中间的RSSI值作为过滤输出。能有效克服偶然因素引起的波干扰，但在脉冲干扰强、样本小的情况下滤波效果不理想。

高斯滤波：通过高斯模型选取高概率区域的RSSI值作为有效值，然后计算其平均值。可有效降低小概率、大干扰对整体测量数据的影响。

高斯中值结合滤波：结合了高斯滤波和中值滤波的均值。

在RSSI值比较不准确的情况下，使用均值滤波结合高斯滤波（最后一行）可以得到一个相对稳定的RSSI值。本应用笔记其余部分中的所有测量均基于高斯和中值滤波。

可以在锚端实现RSSI值过滤，减少PC端的计算量。

4.2.5 与PC通讯

通讯格式如下：

由于GUI只需要收集锚板接收到的RSSI，它会将来自KW38 MCU的RSSI以字符串的形式传输到PC，如图10所示。

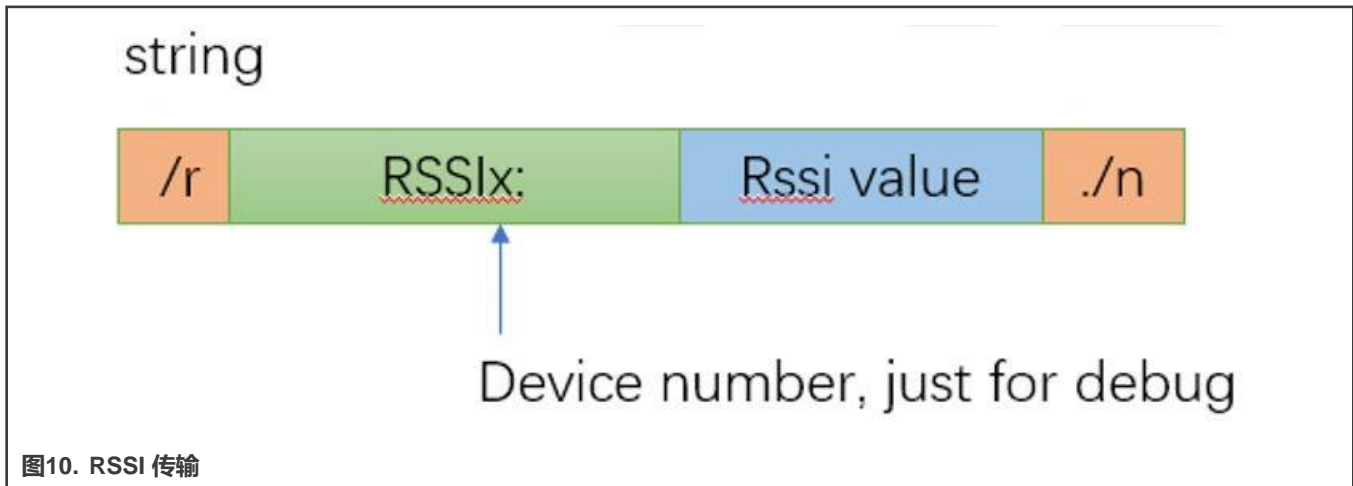


图10. RSSI 传输

例如，如果 KW38 MCU 必须传输的 RSSI 值为 -40dbm，那么它将从 UART 端口打印字符串 “/rRSSI:-40./n”。

锚点扫描如下：

GUI 软件可以在开始测试时向每个锚的 COM 端口发送一个“开始”字符串。当 GUI 停止测试时发送“停止”字符串。这简化了测试的控制方法。锚端对应的代码如图11所示。

注意

这不是必需的，它只是简化了测试的控制方法，这样您就不必在锚侧按下按钮开始或停止扫描。

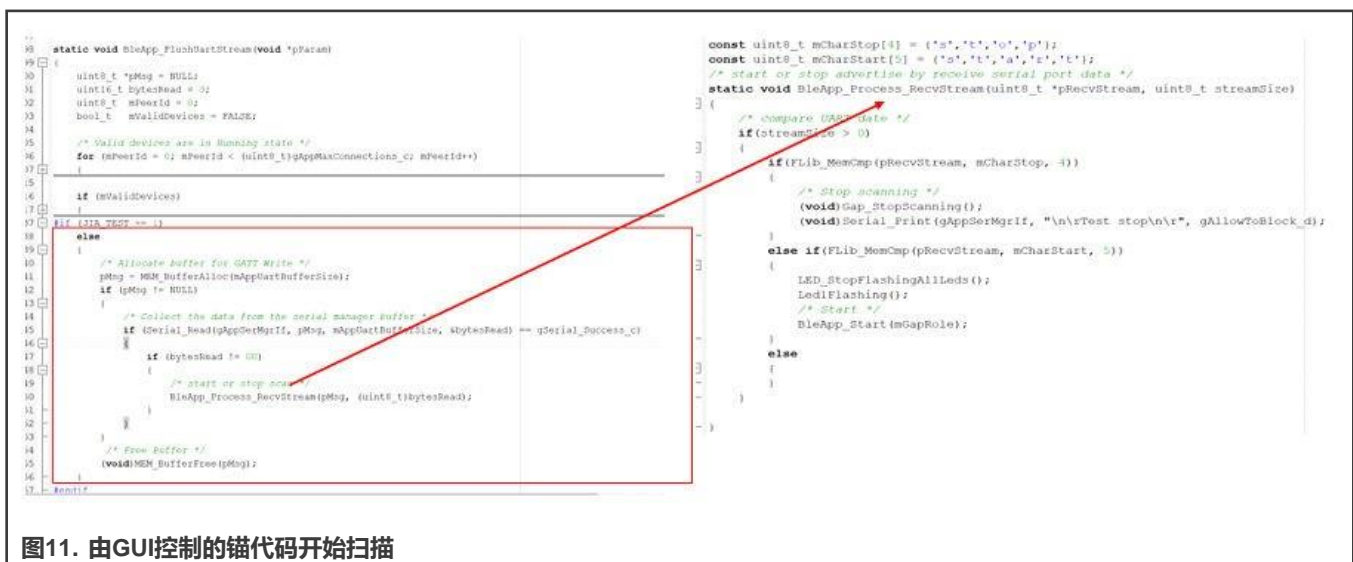


图11. 由GUI控制的锚代码开始扫描

4.3 GUI 软件考虑

PC上的GUI软件由Python实现。它通过 UART 与 FRDM-KW38 板交互，从锚设备收集 RSSI 数据，计算位置，并在 UI 上显示标签位置。

4.3.1 GUI 先决条件

AN源代码工具包中提供了GUI源代码，但没有提供可执行文件。要运行源代码，请安装 Python 3.x 和源代码中所需的 Python 组件。

GUI工具所需的组件包如下：

- PyQt5
- numpy
- matplotlib
- pyserial
- serial

4.3.2 启动GUI

安装GUI工具所需的组件包后，可以轻松启动GUI工具，如下所示：

1. 在 PC 上启动命令提示符应用程序。
2. 切换到源代码储存路径，使用命令：`cd /path of source code`
3. 通过键入以下内容启动 GUI：`mainwindow.py`

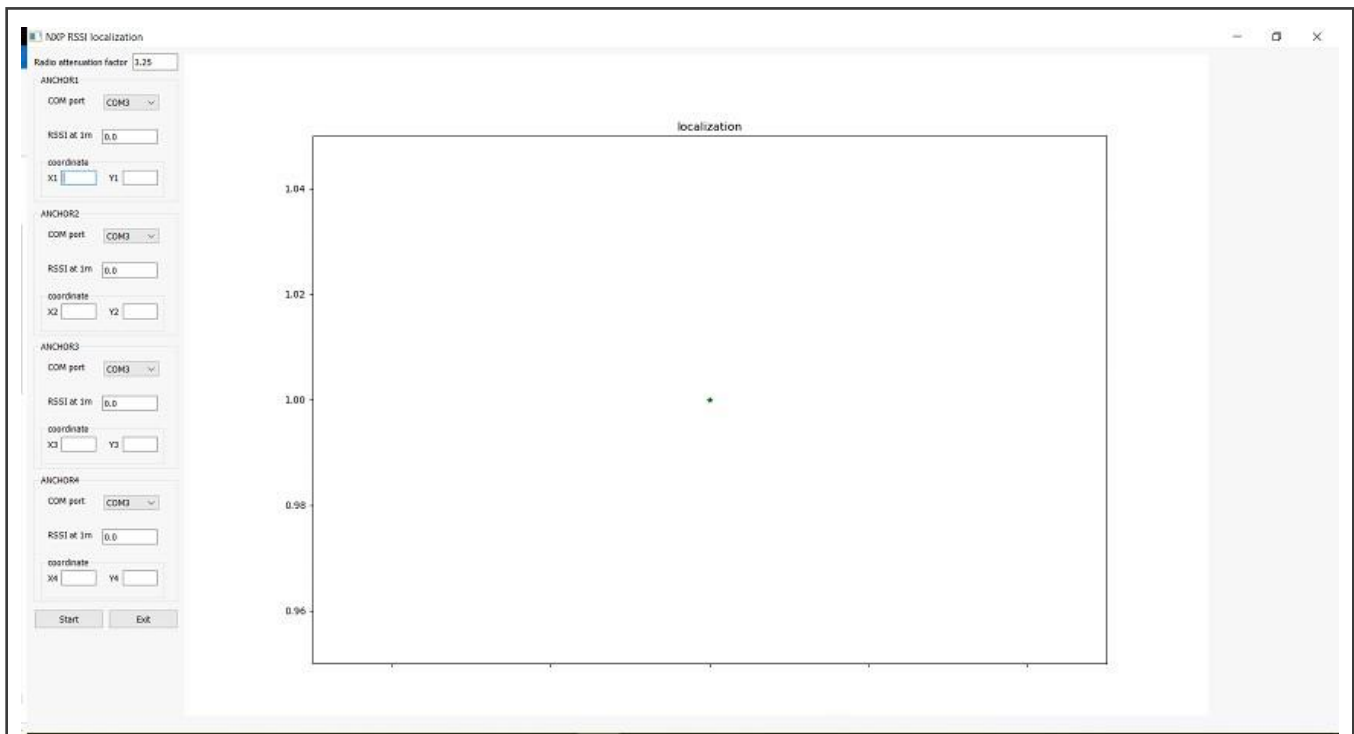


图12. GUI 先决条件

4.4 校准

由于天线灵敏度和RSSI的环境敏感性，并根据[这个公式](#)，在1米的接收 RSSI 值和路径损耗指数应当被测量以校准该系统。

- “1米处的RSSI” 应使用均值滤波与1米距离处 RSSI 测量的高斯滤波相结合。
- 路径损耗指数可以从多个测量的 RSSI 与距离之间的关系推导出来。

5 设置和测试结果

5.1 硬件设置

对于 FRDM-KW38 板，请执行以下操作：

- 使用 SMA 连接器而不是印刷的“F”天线（RF路径必须路由到 SMA 连接器）。
- 将单极天线连接到 SMA 连接器。

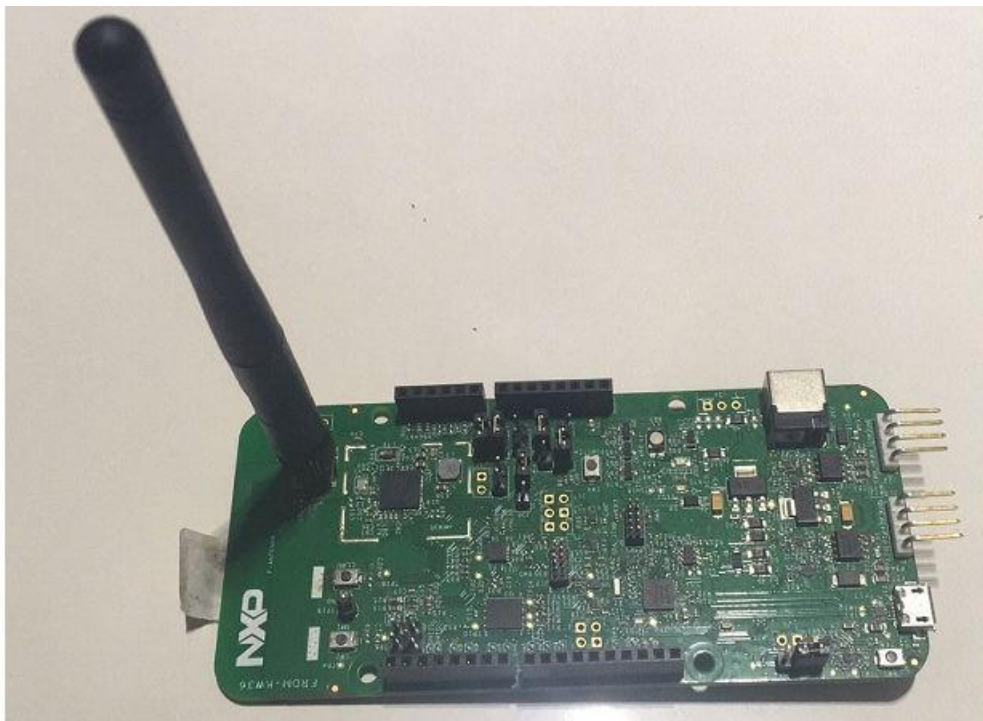


图13. KW38 FREEDOM单极天线板

- 串口设置：
 - 115200波特率
 - 8个数据位
 - 无奇偶
 - 一个停止位
 - 无流量控制

5.2 测试环境及软件配置

5.2.1 从广告中获取RSSI的情况

环境:

- 在室外，测试环境是开放的。
- 蓝牙 LE 或 WIFI 或任何 2.4GHz 信号的影响几乎可以忽略不计。
- 固定位置。

广告参数:

- 仅测试单通道广告。
- 外围设备发送广告的广告间隔为30毫秒，这意味着每秒大约有30个广告数据包。
- 在 *app_preinclude.n* 文件中，将 TEST_FOR_RSSI_ADV 设置为 1 并禁用 TEST_FOR_RSSI_CON。

5.2.2 从连接获取RSSI的情况

环境:

- 在室外，测试环境是开放的。
- 蓝牙 LE 或 WIFI 或任何 2.4GHz 信号的影响几乎可以忽略不计。
- 固定位置。

连接参数:

- 所有连接通道都经过测试。
- 连接间隔为20毫秒，监督是300毫秒。
- 在 *app_preinclude.n* 文件中，将 TEST_FOR_RSSI_CON 设置为 1 并禁用 TEST_FOR_RSSI_ADV。下载锚固件时设置 gTestSlave 为 1，下载标签固件时设置 gTestSlave 为 0。

5.3 测试程序

1. 提前将三块锚板放置在固定位置，记录每个锚的相对坐标。
2. 将锚板连接到个人计算机。
3. 标签板使用 5V 移动电源。
4. 启动GUI工具，输入锚参数（如COM口、一米接收到的RSSI值、锚坐标等）。
5. 点击开始按钮开始测试。工具计算标签坐标后，标签点显示在基础图中。
6. 对于广告，按一下SW3，标签板切换到外设角色，按标签板上的SW2开始广告。
7. 对于连接，在 标签板上按一次 SW2 开始与 锚板的连接。

5.4 测试结果

在这个测试中，三个锚被放置在三个不同的坐标上。锚1坐标为原点（坐标比例为1:1毫米），锚2坐标为（0:2000），锚3坐标为（-2000:0）。图中的图例如下：

- 红色三角形是锚。
- 绿色星号是由 RSSI 计算出的标签坐标点。
- 蓝色圆圈是标签的物理位置。

5.4.1 广告测试1-1

标签距离锚1(Anchor1) 1米， 标签物理坐标在 (710, 710) 。

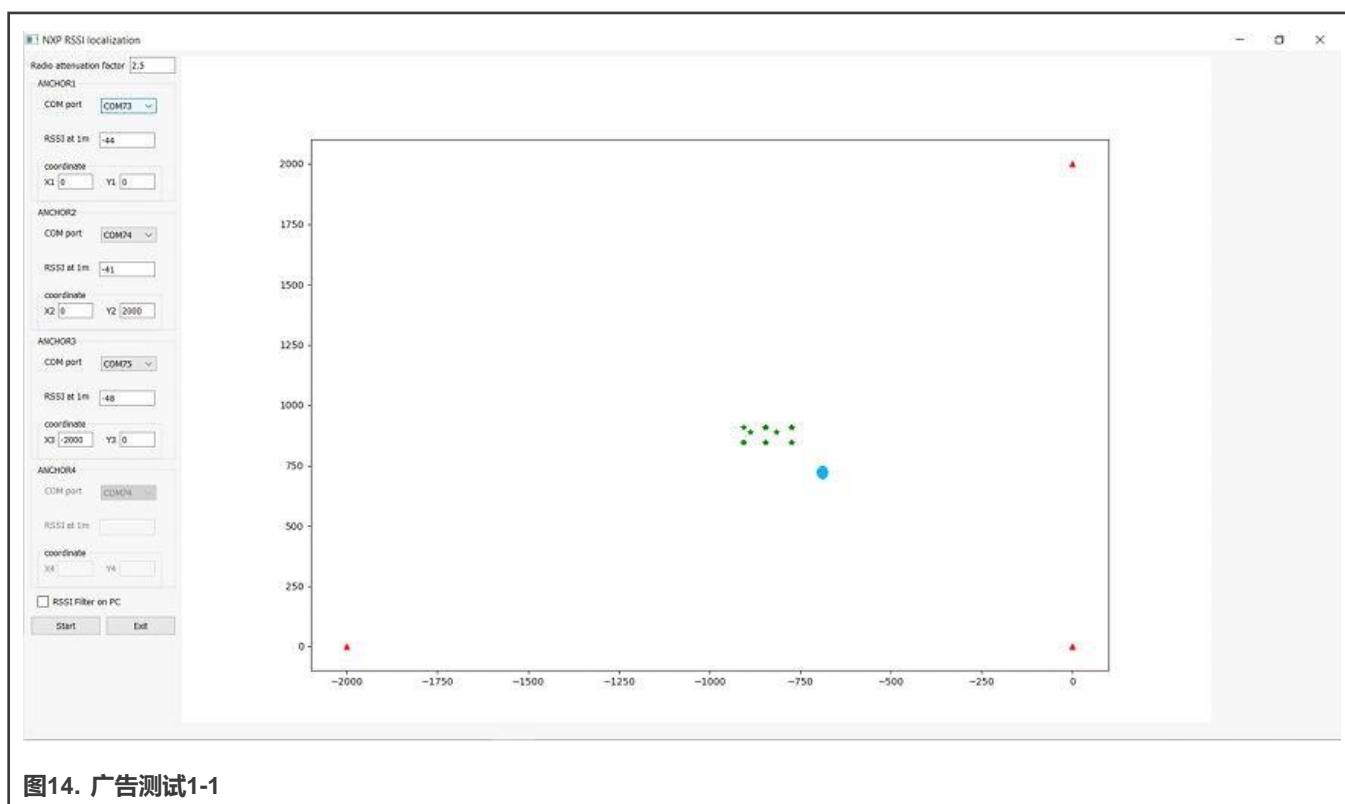


图14. 广告测试1-1

表1. 测试1-1数据

	锚1	锚2	锚3	坐标	
				x	y
距离(m)	0.912	1.0965	1.3183	-773.457	907.3579
	0.912	1.2023	1.2023	-846.555	846.5547
	0.912	1.2023	1.2023	-846.555	846.5547
	0.912	1.2023	1.0965	-907.358	846.5547
	0.912	1.2023	1.0965	-907.358	846.5547
	0.912	1.2023	1.2023	-846.555	846.5547
	0.912	1.0965	1.2023	-846.555	907.3579
	0.912	1.0965	1.2023	-846.555	907.3579
	0.912	1.2023	1.0965	-907.358	846.5547
	0.912	1.0965	1.0965	-907.358	907.3579
	0.912	1.2023	1.0965	-907.358	846.5547
	0.912	1.2023	1.0965	-907.358	846.5547
0.912	1.2023	1.3183	-773.457	846.5547	

表格在下一页继续.....

表1. 测试1-1数据 (续)

0.912	1.0965	1.3183	-773.457	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.0965	1.3183	-773.457	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.2023	1.3183	-773.457	846.5547
0.912	1.0965	1.3183	-773.457	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.2023	1.2023	-846.555	846.5547
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.2023	1.0965	-907.358	846.5547
0.912	1.0965	1.3183	-773.457	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
0.912	1.0965	1.2023	-846.555	907.3579
1	1.2023	1.2023	-888.619	888.6187
0.912	1.2023	1.3183	-773.457	846.5547
1	1.2023	1.3183	-815.521	888.6187

5.4.2 广告测试1-2

标签距离锚1(Anchor1) 2米, 标签物理坐标在 (1750, 1000) 。

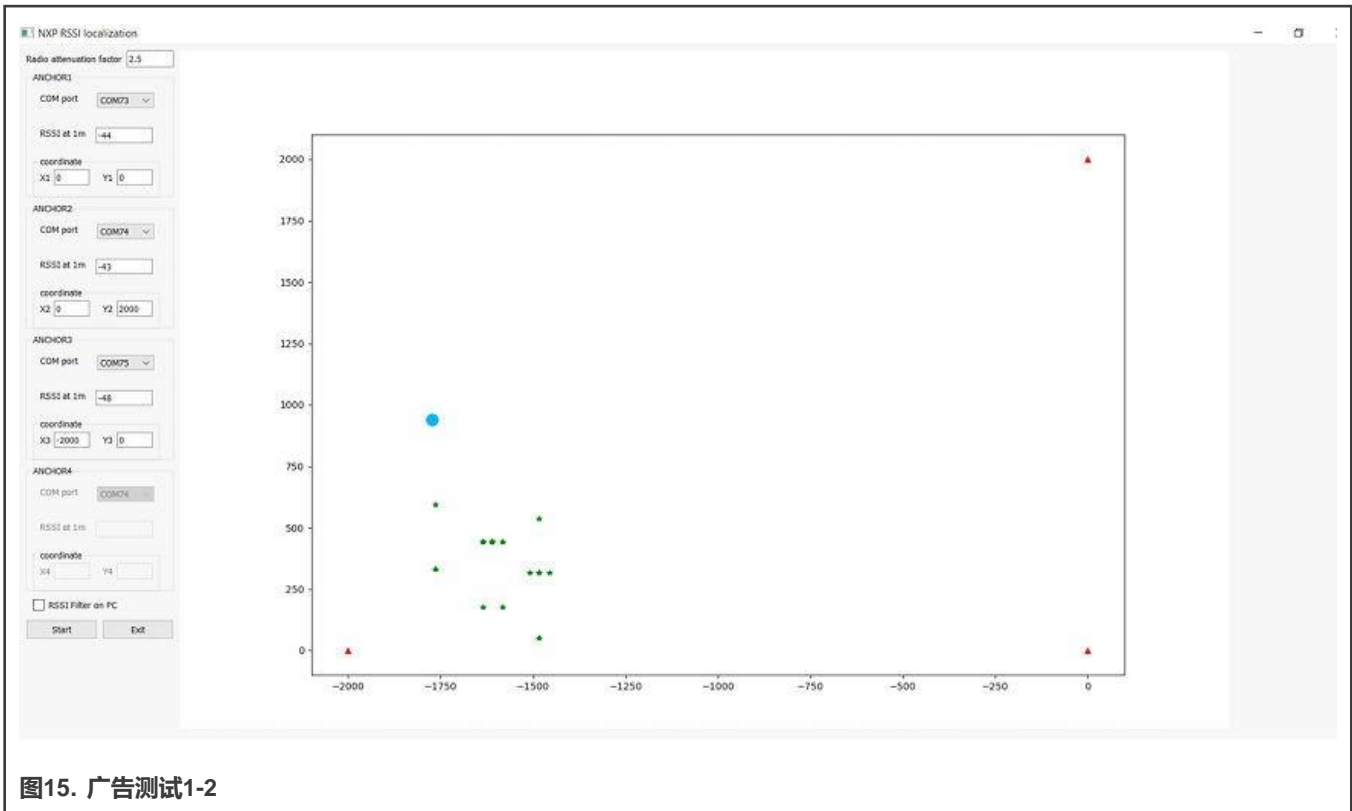


图15. 广告测试1-2

表2. 测试1-2数据

	锚1	锚2	锚3	坐标	
				x	y
距离(m)	1.7378	2.2909	0.7586	-1611.12	442.9315
	1.9055	2.5119	0.7586	-1763.86	330.3222
	1.5849	2.0893	0.7586	-1484.11	536.6834
	1.5849	2.5119	0.7586	-1484.11	50.5666
	1.7378	2.2909	0.8318	-1582.01	442.9315
	1.7378	2.2909	0.7586	-1611.12	442.9315
	1.9055	2.2909	0.7586	-1763.86	595.6769
	1.9055	2.5119	0.7586	-1763.86	330.3222
	1.9055	2.5119	0.7586	-1763.86	330.3222
	1.7378	2.5119	0.6918	-1635.34	177.5768
	1.7378	2.2909	0.7586	-1611.12	442.9315
	1.7378	2.2909	0.7586	-1611.12	442.9315
	1.7378	2.2909	0.8318	-1582.01	442.9315
	1.7378	2.5119	0.8318	-1582.01	177.5768
1.5849	2.2909	0.7586	-1484.11	315.9213	

表格在下一页继续.....

表2. 测试1-2数据 (续)

1.5849	2.2909	0.8318	-1484.11	315.9213
1.7378	2.2909	0.7586	-1611.12	442.9315
1.5849	2.2909	0.7586	-1484.11	315.9213
1.5849	2.2909	0.7586	-1484.11	315.9213
1.5849	2.5119	0.7586	-1484.11	50.5666
1.7378	2.2909	0.6918	-1635.34	442.9315
1.5849	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.6918	-1635.34	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.6918	-1635.34	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.6918	-1635.34	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315
1.5849	2.2909	0.6918	-1508.33	315.9213
1.7378	2.2909	0.6918	-1635.34	442.9315
1.5849	2.2909	0.7586	-1484.11	315.9213
1.7378	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.6918	-1635.34	442.9315
1.7378	2.2909	0.6918	-1635.34	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315
1.7378	2.2909	0.7586	-1611.12	442.9315

5.4.3 广告测试1-3

标签距离锚1(Anchor1) 3米, 标签物理坐标在 (1750, 1000) 。

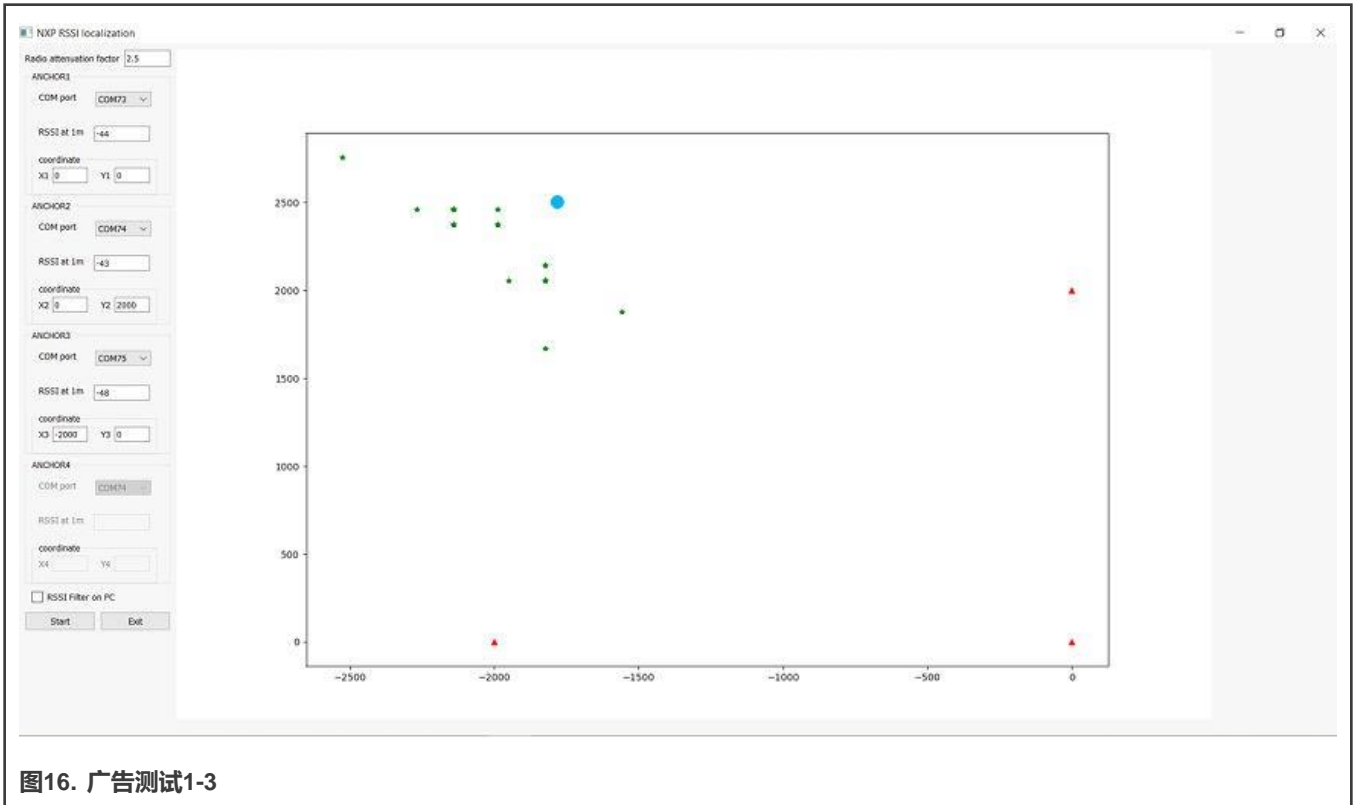


图16. 广告测试1-3

表3. 测试1-3数据

	锚1	锚2	锚3	坐标	
				x	y
距离(m)	2.5119	1.4454	1.7378	-1822.42	2055.115
	2.7542	1.3183	1.5849	-2268.43	2461.926
	2.2909	1.3183	1.7378	-1557.07	1877.577
	2.5119	1.3183	1.7378	-1822.42	2142.932
	2.7542	1.4454	1.7378	-2141.42	2374.109
	2.7542	1.3183	1.7378	-2141.42	2461.926
	2.5119	1.4454	1.7378	-1822.42	2055.115
	2.7542	1.4454	1.7378	-2141.42	2374.109
	2.7542	1.3183	1.7378	-2141.42	2461.926
	3.02	1.4454	1.7378	-2525.11	2757.805
	2.5119	1.3183	1.5849	-1822.42	2142.932
	2.5119	1.4454	1.7378	-1822.42	2055.115
	2.5119	1.4454	1.7378	-1822.42	2055.115
2.7542	1.3183	1.7378	-2141.42	2461.926	

表格在下一页继续.....

表3. 测试1-3数据 (续)

	2.7542	1.4454	1.7378	-2141.42	2374.109
	2.5119	1.3183	1.7378	-1822.42	2142.932
	2.5119	1.3183	1.7378	-1822.42	2142.932
	2.7542	1.3183	1.7378	-2141.42	2461.926
	2.7542	1.3183	1.7378	-2141.42	2461.926
	2.7542	1.4454	1.7378	-2141.42	2374.109
	2.7542	1.9055	1.7378	-1822.42	1669.678
	2.5119	1.3183	1.7378	-2141.42	2461.926
	2.7542	1.3183	1.9055	-2141.42	2461.926
	2.7542	1.4454	1.9055	-1988.67	2374.109
	2.7542	1.4454	1.7378	-1988.67	2374.109
	2.7542	1.4454	1.9055	-1988.67	2374.109
	2.5119	1.4454	1.7378	-1822.42	2055.115
	2.7542	1.3183	1.9055	-1988.67	2461.926

6 总结

在本应用笔记中，实现了基于低功耗蓝牙 RSSI 测距的简单三点定位。测试结果表明，标签节点靠近锚时准确率较高，但随着距离增加，准确率下降。此功能可用于某些应用程序。例如，当距离较远时（RSSI值较低时），可用于识别进入位置区域的标签。当距离较短时（RSSI值高时），可用于定位。这可以通过 KW38 MCU 切实有效地实现。

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QoriQ, QoriQ Converge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, μ Vision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© NXP B.V. 2020.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 09/2020

Document identifier: AN12977

