AN12977

KW38/KW36 Localization base on RSSI ranging

Rev. 0 — 09/2020 Application Note

by: NXP Semiconductors

Contents

1 Introduction

The Received Signal Strength Indication (RSSI) is used to determine the quality of a wireless signal. It can be used in a localization system. The relationship between the signal space path loss and the distance can be obtained by establishing the signal-attenuation model. The distance can be calculated using the link-loss equation and the position can be evaluated by combining the distance with the software algorithm.

This document provides an introduction to localization based on the Bluetooth LE RSSI ranging. It implements a simple trilateral localization system using the Kinetis KW38 wireless MCU.

2 RSSI ranging

2.1 Prior definition

Tag - a target in an unknown location that must be located. In this application note, it is a KW38 board.

Anchor – a device at a fixed location. It collects the RSSI values and uploads them to the location computing device. Usually, a minimum of three Anchors are required for the localization.

2.2 Wireless signal path loss models

In the location system, based on RSSI ranging, the Tag node applies the RSSI measurements to estimate its distances from the Anchor node using a known signal propagation model. The model of the wireless signal path loss in space is shown in Figure 1 (a1c1, a1c2, and so on are test device numbers).

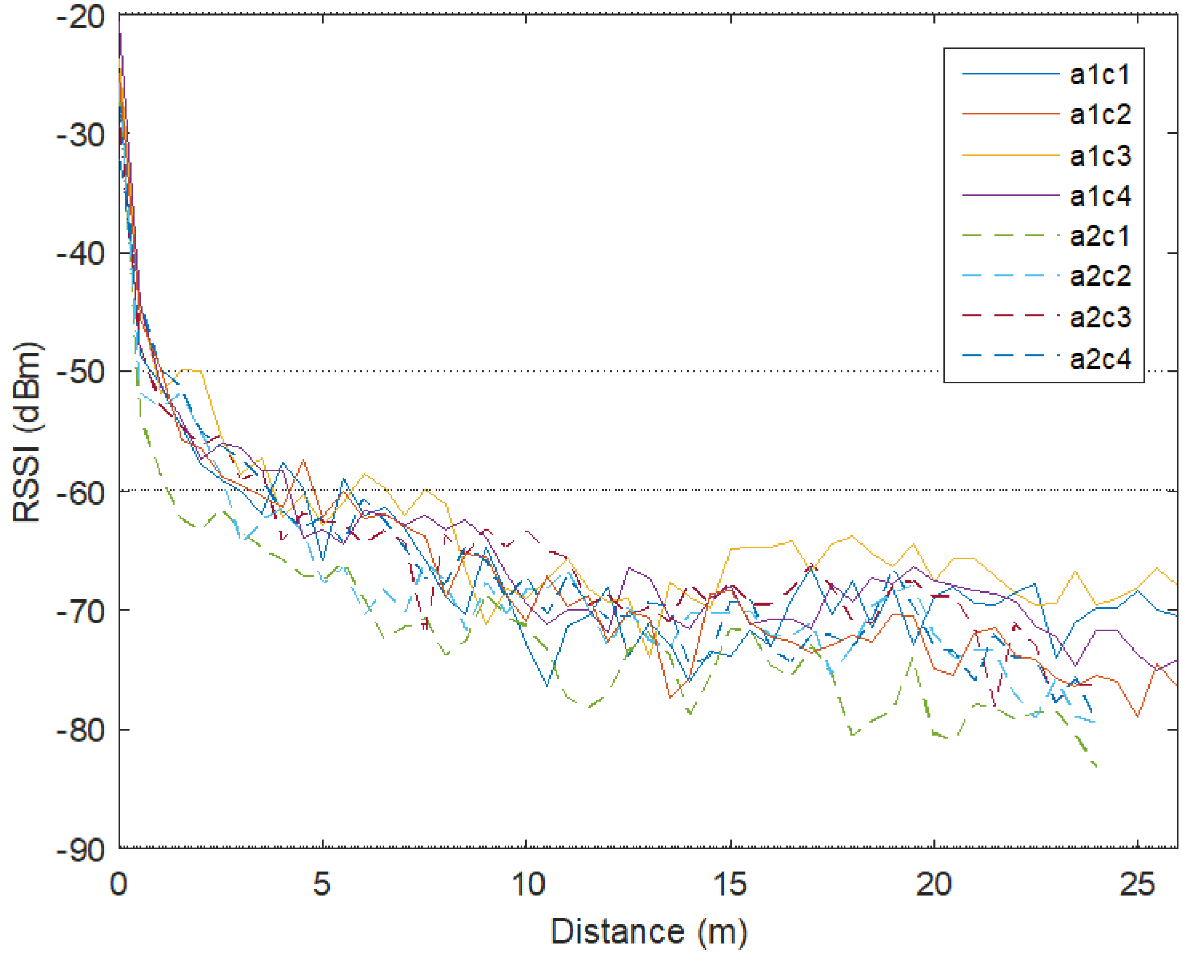

Figure 1. Wireless signal path loss

The shadowing model is widely used to model the wireless signal propagation loss, which is expressed as follows:

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

- d and d_0 denote the real distance and the reference distance.
- $P_r(d)$ and $P_r(d_0)$ denote the RSSI (dBm) received at the real distance and the reference distance, respectively.
- n is the path loss exponent.
- a_δ is random noise. A gaussian distributed random variable with zero mean and variance is always assumed. As shown in Figure 1, the greater the distance, the greater the fluctuation in the received RSSI value.

For most applications, d_0 is 1 meter and $P_r(d_0)$ is calculated by a free space path loss formula. Thus, the simplified shadowing model is used as follows:

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

- d is distance.
- n is determined by the test environment and it is usually an empirical constant that is gathered through testing.
- RSSI is the received RSSI value at the distance d.

- *A* is the received RSSI of the receiver from a transmitter one meter away.

This formula is used to estimate distances based on RSSI.

2.3 Influence on RSSI receiving accuracy

2.3.1 Environment

Because of the existence of Multipath, obstacles, electromagnetic interference, and other unstable factors in the environment, the RSSI value on the receiving nodes is sometimes unstable and it has a constructive and destructive superposition.

The filtering algorithms can be used to filter inaccurate RSSI values, as described in Filtering in RSSI.

2.3.2 Antenna

Different antennas have different radiation modes, polarization directions, and gain for signals. In the same conditions, the use of different antennas at the receiver and transmitter also affects the RSSI.

3 Trilateral localization based on RSSI ranging

3.1 Trilateral localization

At least three anchor nodes are enough to locate a Tag node. The trilateral localization is shown in Figure 2.

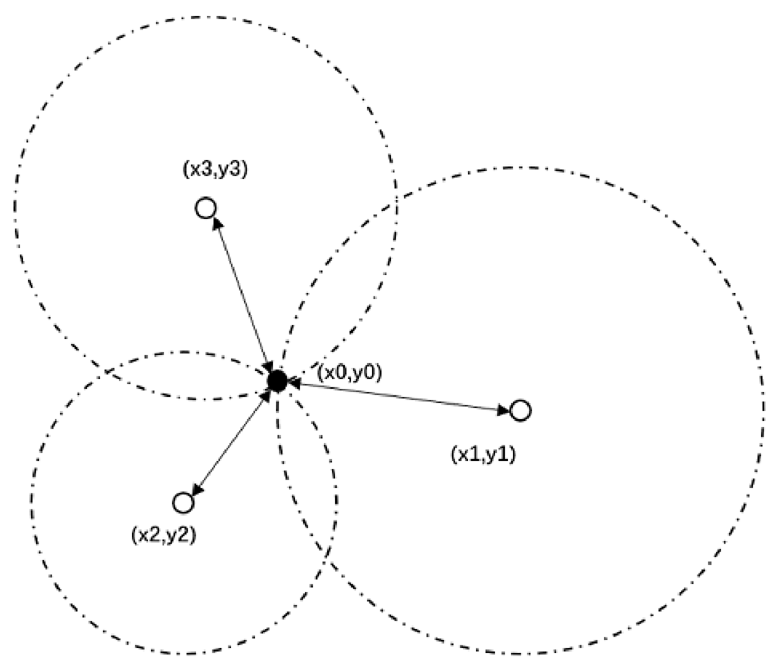

Figure 2. Trilateral localization in ideal condition

- (x_0, y_0) is the Tag board coordinate to be determined.
- (x_1, y_1), (x_2, y_2), (x_3, y_3) are three known Anchor board's coordinates.

- The radius of the three circles is the distance from the Tag to each Anchor. It can be calculated using this formula. The coordinates of the Tag can be calculated by the following system of equations (D is the distance from the Tag board to the Anchor board).

$$(x_1 - x_0)^2 + (y_1 - y_0)^2 = D_1^2$$
$$(x_2 - x_0)^2 + (y_2 - y_0)^2 = D_2^2$$
$$(x_3 - x_0)^2 + (y_3 - y_0)^2 = D_3^2$$

Due to measurement errors, the three circles rarely intersect at one point, as shown in Figure 3. The least square method is considered the best way to get an optimal solution. The equation set is converted to a matrix form:

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

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

The Tag board coordinates are calculated using the least square method equation:

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

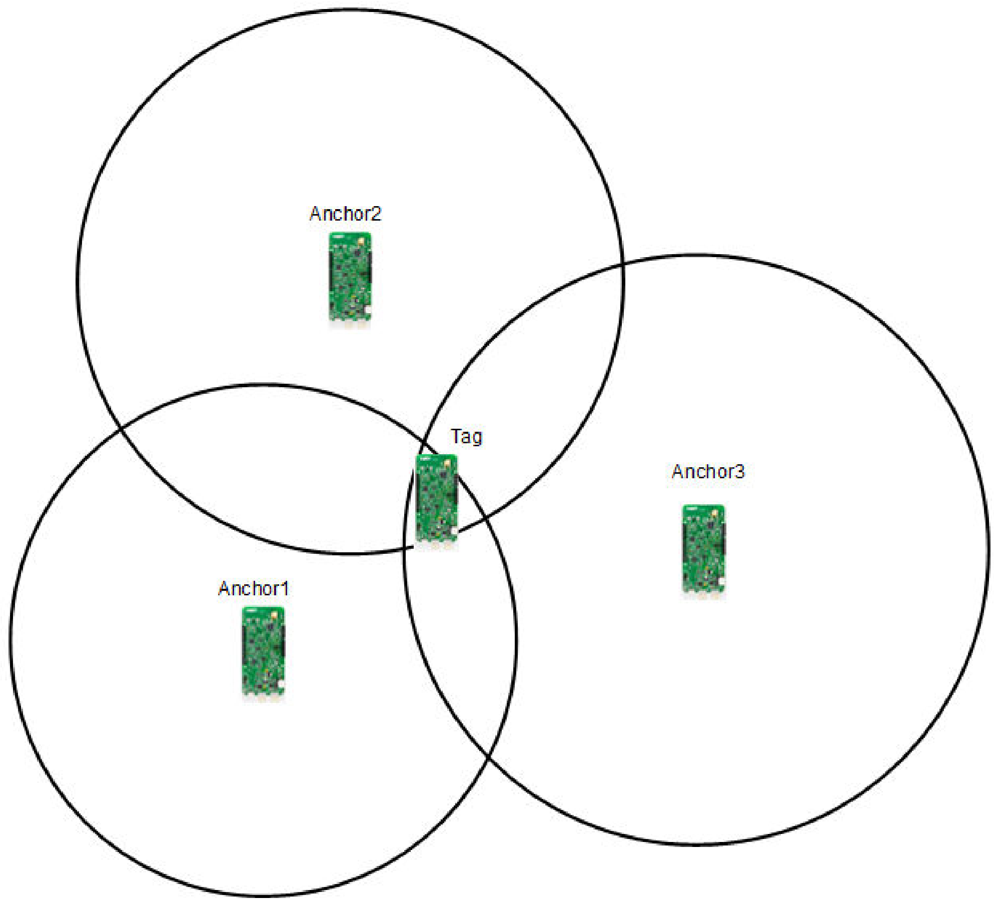

Figure 3. Tag board coordinates

4 Localization system implementation considerations

4.1 Localization scheme and hardware requirements

There are two localization schemes, as described in the following sections.

4.1.1 Localization scheme - advertising

The localization scheme is shown in Figure 4. For the RSSI acquirement, see Acquiring RSSI from advertising.

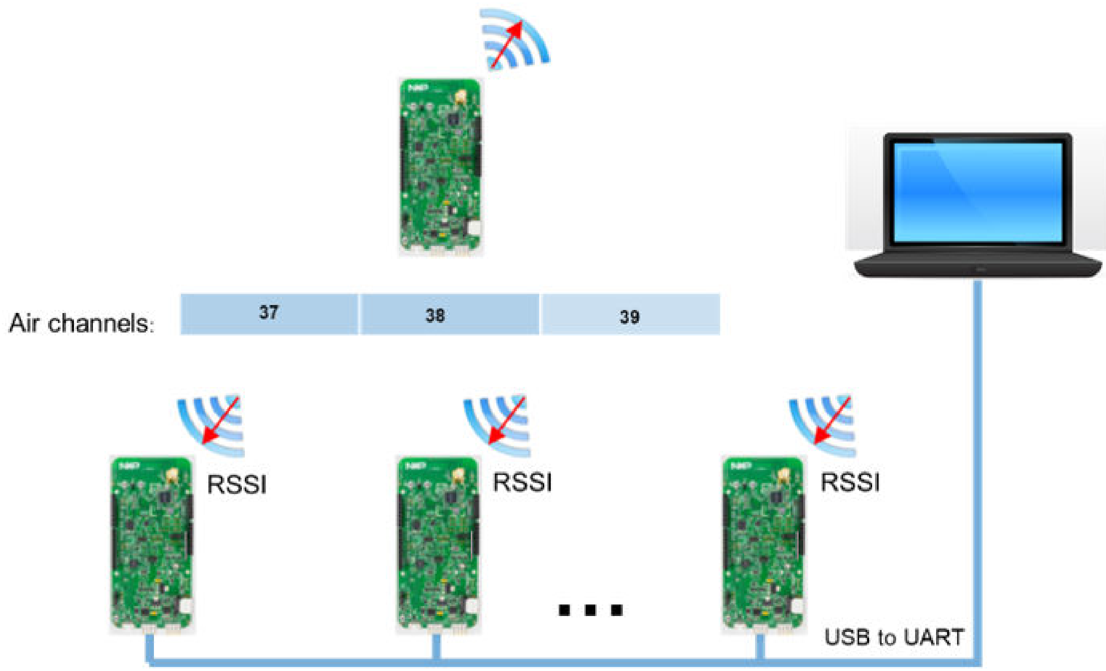

Figure 4. Advertising

The hardware prerequisites are as follows:

- Three FRDM-KW38 boards as the Anchor boards to scan
- One FRDM-KW38 board as the Tag board to advertise
- Four Micro-USB cables
- Four Monopole antennas
- One personal computer with the GUI tool installed

4.1.2 Localization scheme – connection

The localization scheme is shown in Figure 5. For the RSSI acquirement, see Acquiring RSSI from connection.

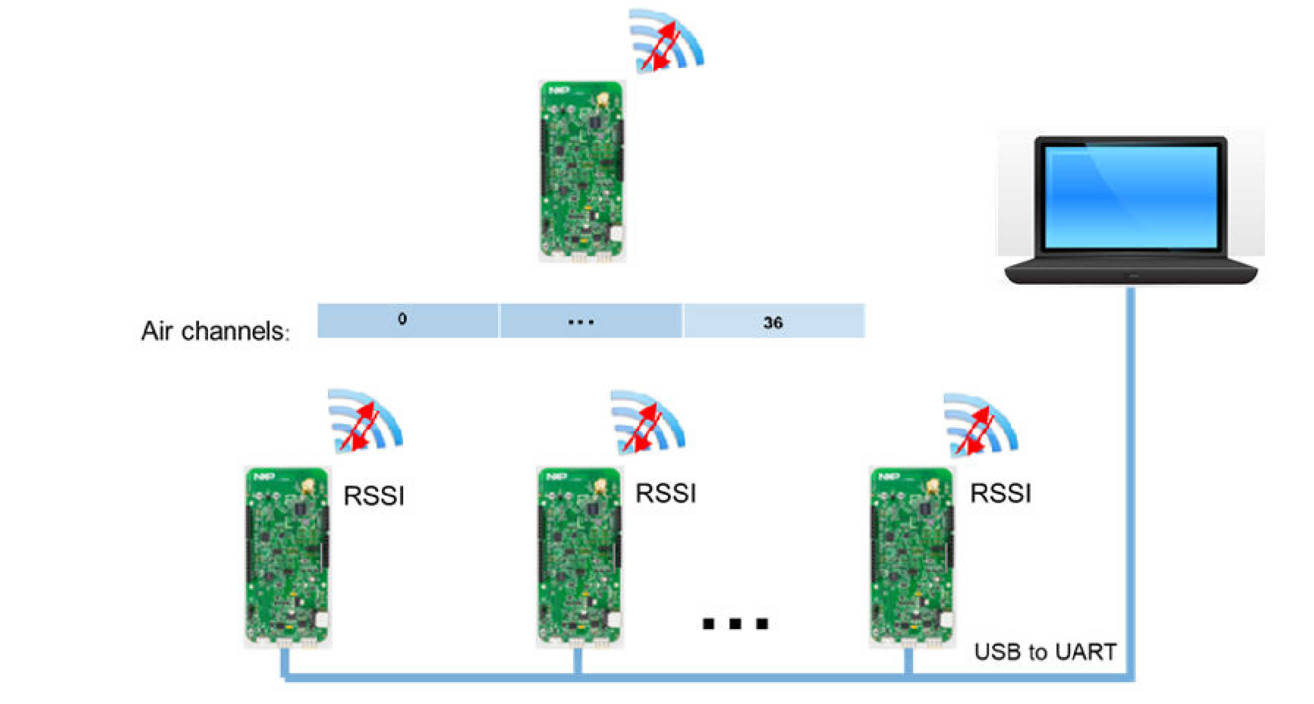

Figure 5. Connection

- Three FRDM-KW38 boards as the Anchor boards to wait for the connection
- One FRDM-KW38 board as the Tag board to connect with the Anchors
- Four Micro USB cables
- Four Monopole antennas
- One personal computer with the GUI tool installed

4.2 Embedded software consideration

4.2.1 Bluetooth LE project

The KW38 SDK provides the Wireless UART project, which implements the RSSI notification by the Bluetooth LE controller. The KW38 Bluetooth LE controller supports the RSSI notification associated with the channel number in the scanning event and the RSSI reporting in the data packet in the connection event.

The project path is *SDK\boards\frdmkw38\wireless_examples\bluetooth\ w_uart*

You only have to enable the *gUseControllerNotifications_c* macro. The *gControllerNotificationEvent_c* events are reported at *BleApp_GenericCallback()*. Select which type of radio packets should be notified to the RSSI application (when received) by the *Gap_ControllerEnhancedNotification()* API function.

NOTE

For more information about the Wireless UART, see the *Bluetooth Low Energy Demo Applications User Guide* (document BLEDAUG)

4.2.2 Acquiring RSSI from advertising

Both Anchor and Tag boards are programmed with Wireless UART. The difference is that the Anchor boards are used as the central device to scan advertising and the Tag board is used as the peripheral which sends advertising.

4.2.2.1 Configure notification type

On the central side, configure only the send notification of the Scanning ADV PKT Rx by the *Gap_ControllerEnhancedNotification()* API function.

4.2.2.2 Continuous scan and address filtering

By default, the Wireless UART behaves as a central device. It searches for other Wireless UART devices to connect. Before finding a device, it scans all advertisements and notifies the application about the RSSI value.

To continuously scan the expected device's advertising, do not connect the expected device, add the device address to the whitelist, and scan with the whitelist.

```
 ****************************************************************************** */
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(mAppAlreadyScanedDevice == FALSE)
                {
                    Gap_StopScanning();
                    Gap_AddDeviceToWhiteList(gConnReqParams.peerAddressType, gConnReqParams.peerAddress); // add device address to white list
                    mAppAlreadyScanedDevice = TRUE;

                    gScanParams.filterPolicy = gScanWithWhiteList_c; // only scan addres 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
```

Figure 6. Code to make Wireless UART continuously scan the expected device

In addition, the scanning is stopped after 10 seconds by a scan timer. To scan continously, disable the timer *ScanningTimerCallback()* function.

4.2.3 Acquiring RSSI from connection

Both Anchor and Tag boards are programmed with Wireless UART. The difference is that Anchor boards are used as the peripheral device to start advertising and wait for the central device connection and the Tag boards are used as the central device to connect all peripheral devices.

4.2.3.1 Establishing multiple connections quickly

To quickly establish a connection between the central device and multiple peripherals, add all target peripherals' public addresses to the whitelist on the central device side before starting the connection. Then the central device connects the peripherals in the whitelist, as shown in the figures below.

Figure 7. Establishing multiple connections quickly

```
/************************************************************************************ */
void 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)
    {
        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);
                }
```

Figure 8. Establishing multiple connections quickly

4.2.3.2 Peripherals delay start advertising

In the process of establishing multiple connections, if multiple peripherals start advertising at the same time, it may sometimes cause the central device to not establish connection with all devices, but to establish connection with some of them. The safe way to solve this problem is to delay the start of the advertising by the peripherals in the sequence. This delay time is set to 100 ms in the GUI tool.

4.2.3.3 Configure notifications type

On the peripheral side, when connected with the central device, configure only the send notification of the Connection Rx PDU and Connection event.

4.2.3.4 Connection parameter

If the connection link is lost, the device continues to send the RSSI value of the last received connection event until the supervision timeout disconnects. Therefore, the supervision timeout value should be within a reasonable range to reduce false values caused by a link loss.

4.2.4 Filtering in RSSI

In practice, because the RSSI is sensitive to the environment, multiple measured RSSI values must be filtered and optimized to get the optimized RSSI value, and then calculate the location. The filtering algorithm can be implemented on KW38 MCUs. Figure 9 shows the compares commonly used by RSSI filtering algorithms.

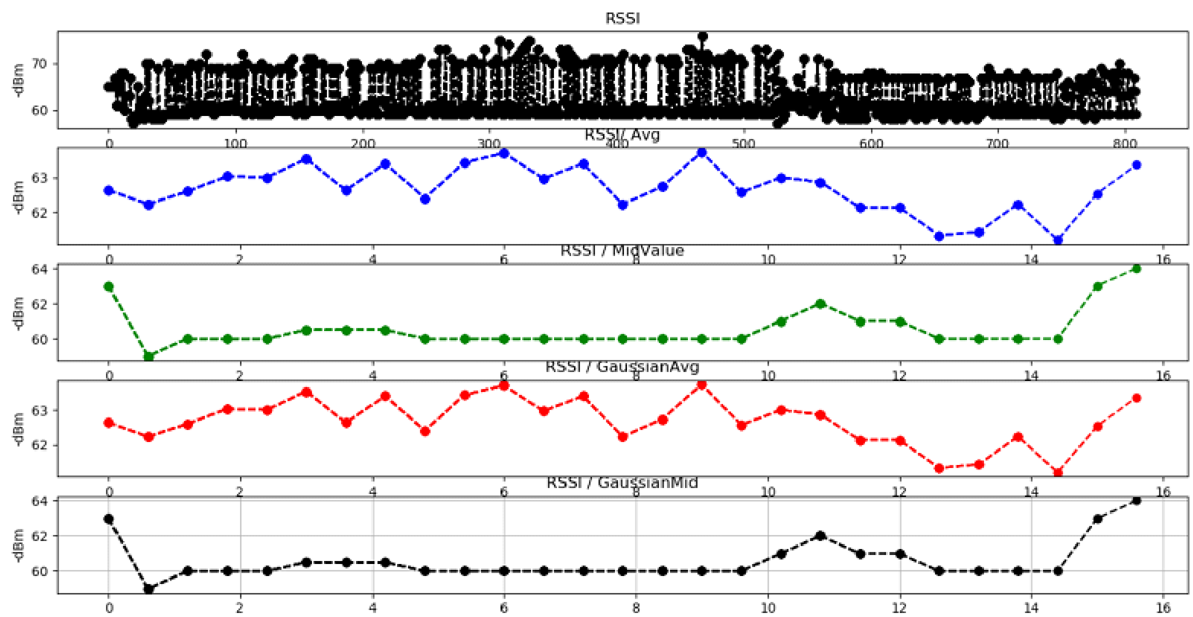

Figure 9. RSSI filtering algorithm compare

In the test shown in Figure 9, the Tag and Anchor devices are placed in a fixed position. The Anchor device collects and filters the RSSI. The first row is the RSSI measurement and the other rows are the optimized RSSI values of the filtered output. In Figure 9, the filter collects 30 samples for filtering once.

There are four types of filtering: mean filtering, median filtering, Gaussian filtering, and Gaussian filtering combined with median filtering.

Mean filtering: The mean value of multiple RSSI values is calculated as the test result. When the RSSI value fluctuates greatly, the reliability of results is decreased.

Median filtering: It arranges the multiple RSSI according to the values. The RSSI value in the middle is taken as the filtering output. It can effectively overcome the wave interference caused by accidental factors, but the filtering is not ideal in the case of a strong pulse interference and small samples.

Gaussian filtering: The RSSI value in the high-probability region is selected as the effective value through the Gaussian model and then its mean value is calculated. It can effectively reduce the influence of small probability and big interference on the overall measurement data.

Gaussian combined with median filtering: Combines the means of Gaussian filtering and median filtering.

In the case of more inaccurate RSSI values, a relatively stable RSSI value can be obtained using mean filtering combined with Gaussian filtering (the last row). All the measurements in the rest of the application note are based on the Gaussian and median filtering.

The RSSI value filtering can be implemented on the Anchor side to reduce the amount of calculation on the PC side.

4.2.5 Communicate with PC

The communication format is as follows:

Because the GUI only has to collect the RSSI received by the Anchor board, it transmits the RSSI from the KW38 MCU to the PC as a string, as shown in Figure 10.

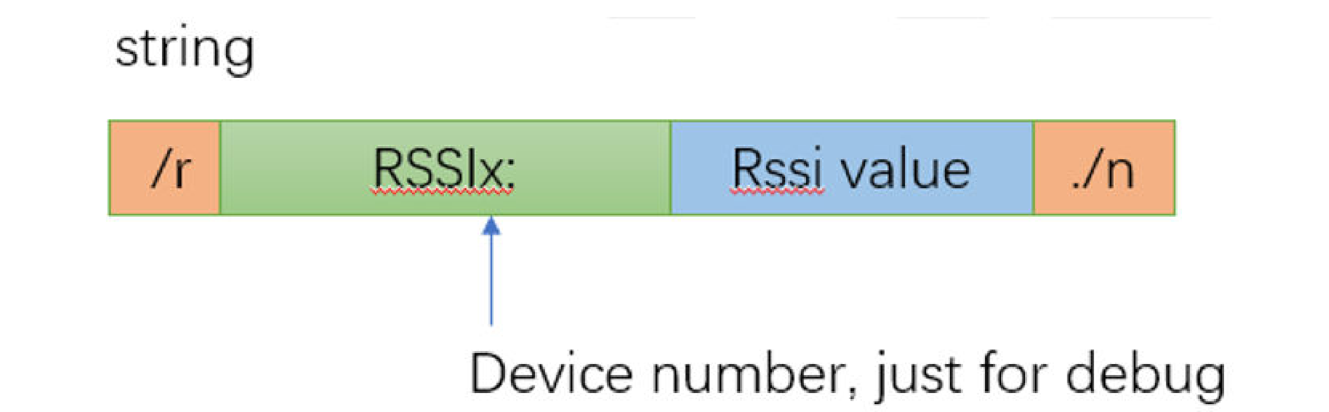

Figure 10. RSSI transmission

For example, if the RSSI value that the KW38 MCU has to transmit is -40 dbm, then it will print the string "/ rRSSI: -40./n" from the UART port.

The Anchor scan is as follows:

The GUI software can send a "start" string to each Anchor COM port when starting the test. A “stop” string is sent when the GUI stops the test. This simplifies the control method for the test. The corresponding code on the Anchor side is shown in Figure 11.

NOTE

This is not necessary, it just simplifies the control method of the test so that you do not have to press the button to start or stop the scan on the Anchor side.

Figure 11. Anchor code controlled by GUI to start scan

4.3 GUI software consideration

The GUI software on the PC is implemented by Python. It interacts with the FRDM-KW38 board via UART, collects the RSSI data from the Anchor devices, calculates the location, and displays the tag position on the UI.

4.3.1 GUI prerequisite

The GUI source code is provided in the AN source code tool packet, but no executable file is provided. To run the source code, install Python 3.x and the Python module required in the source code.

The package of modules required for the GUI tool is as follows:

- PyQt5
- numpy
- matplotlib
- pyserial
- serial

4.3.2 Launch GUI

After the package of modules required for the GUI tool is installed, the GUI tools can be easily launched as follows:

1. Launch the Command Prompt application on the PC.
2. Switch to the source code storage path using the following command: `cd /path of source code`.
3. Launch the GUI by typing the following: `mainwindow.py`

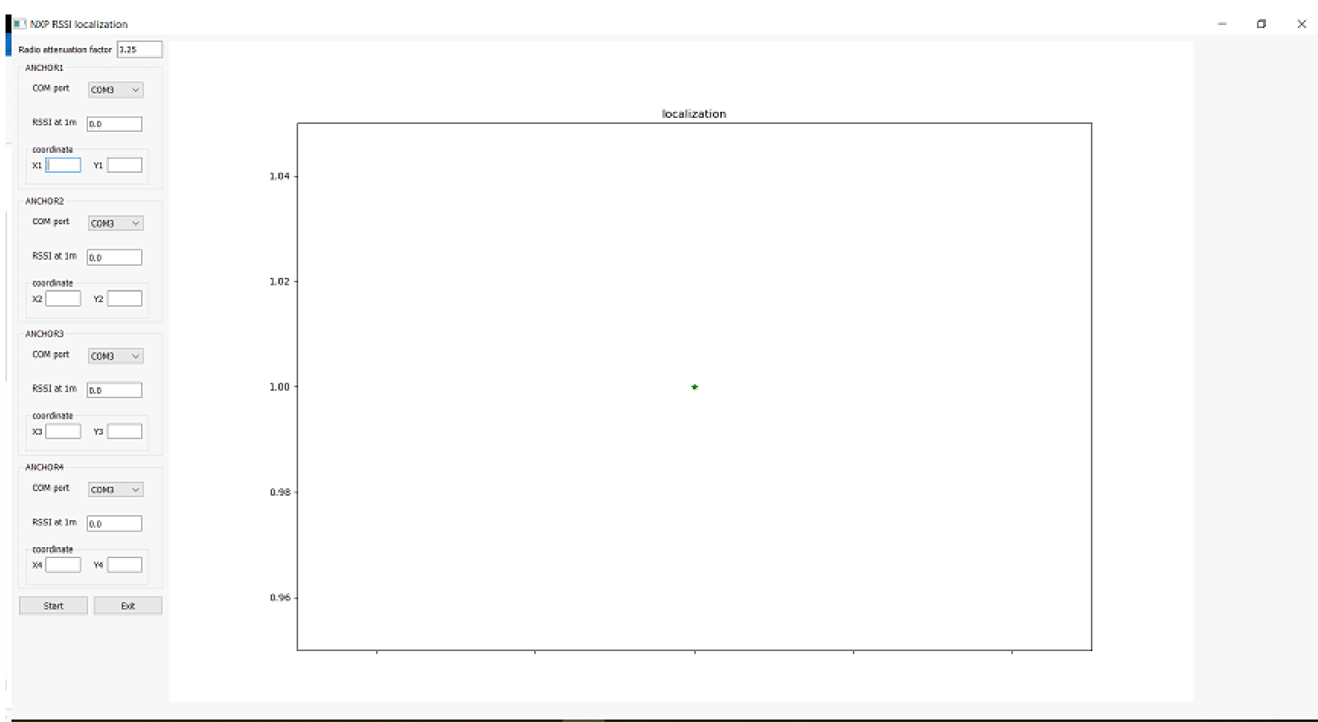

Figure 12. GUI prerequisite

4.4 Calibration

Due to the antenna sensitivity and environmental sensitivity of the RSSI and according to this formula, the received RSSI value at 1 meter and the path loss exponent should be measured to calibrate the system.

- The “RSSI at 1 meter” should use mean filtering combined with Gaussian filtering of the RSSI measure at a 1-meter distance.
- The path loss exponent can be derived from the relationship between multiple measured RSSIs and the distance.

5 Setup and test result

5.1 Hardware setup

For the FRDM-KW38 board, do the following:

- Use the SMA connector instead of the printed “F” antenna (the RF path must be routed to the SMA connector).
- Connect the monopole antenna to the SMA connector.

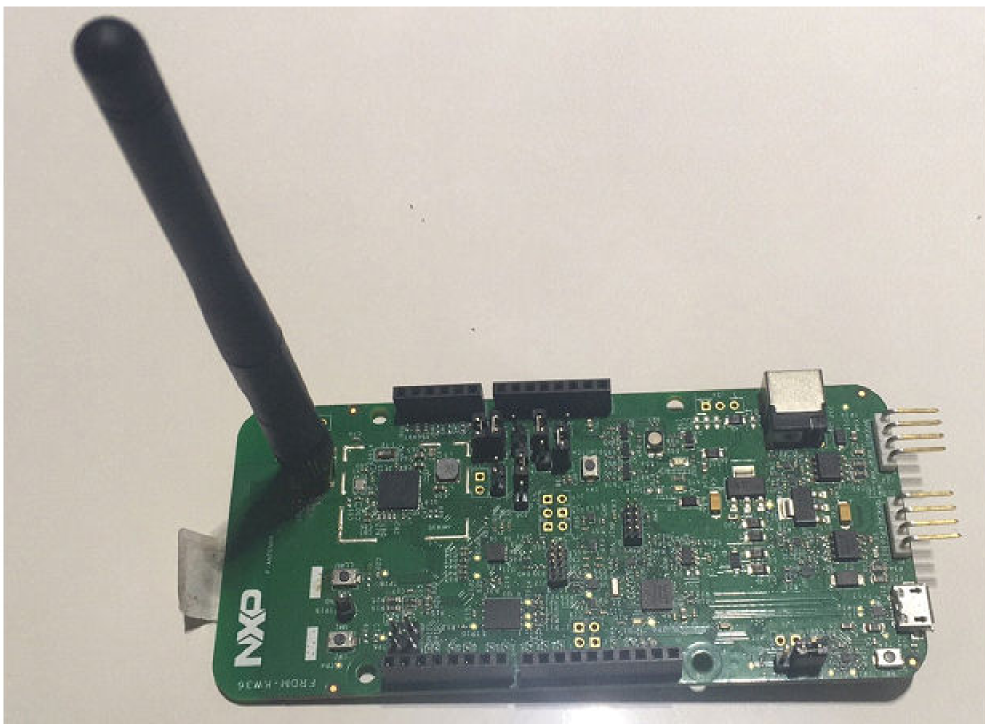

Figure 13. KW38 FREEDOM board with monopole antenna

- Serial Com port settings:
  - — 115200 baud rate
  - — 8 data bits
  - — No parity
  - — One stop bit
  - — No flow control

5.2 Test environment and software configuration

5.2.1 In case of acquiring RSSI from advertising

Environment:

- Outdoors, the test surrounding is open.
- The influence of Bluetooth LE or WIFI or any 2.4-GHz signal is almost negligible.
- Fixed position.

Advertising parameter:

- Test only the single-channel advertising.
- The peripheral device send advertising has an advertising interval of 30 ms, that means about 30 advertising packets per second.
- In the *app_preinclude.n* file, set TEST_FOR_RSSI_ADV to 1 and disable TEST_FOR_RSSI_CON.

5.2.2 In case of acquiring RSSI from connection

Environment:

- Outdoors, the test surrounding is open.
- The influence of Bluetooth LE or WIFI or any 2.4-GHz signal is almost negligible.
- Fixed position.

Connection parameter:

- All channels of connection are tested.
- The connection interval is 20 ms, the Supervision is 300 ms.
- In the *app_preinclude.n* file, set TEST_FOR_RSSI_CON to 1 and disable TEST_FOR_RSSI_ADV. Set gTestSlave to 1 when downloading the Anchor firmware and set gTestSlave to 0 when downloading the Tag firmware.

5.3 Test procedure

1. Place the three Anchor boards in a fixed position in advance and record the relative coordinates of each Anchor.
2. Connect Anchor boards to a personal computer.
3. Use a 5-V mobile power supply for the Tag board.
4. Launch the GUI tool and enter the Anchor parameter (such as the COM port, RSSI value received on one meter, and Anchor's coordinate).
5. Click the start button to start the test. The tag dots are displayed in the base diagram after the tool calculates the tag coordinates.
6. For Advertising, press SW3 once to switch the Tag board to the peripheral role and press SW2 on the Tag board to start advertising.
7. For Connection, press SW2 once on the Tag board to start the connection with the Anchor board.

5.4 Test result

In this test, three Anchors are placed at three different coordinates. The Anchor1 coordinate is the origin (the coordinate ratio is 1:1 millimeter), the Anchor2 coordinate is (0:2000), and the Anchor3 coordinate is (-2000:0). The legend in the figure is as follows:

- The red triangle is the Anchor.
- The green star is the Tag's coordinate point calculated by the RSSI.
- The blue circle is the Tag's physical location.

5.4.1 Advertising test 1-1

The Tag is 1 meter away from Anchor1, the tag physical coordinate is in (710, 710).

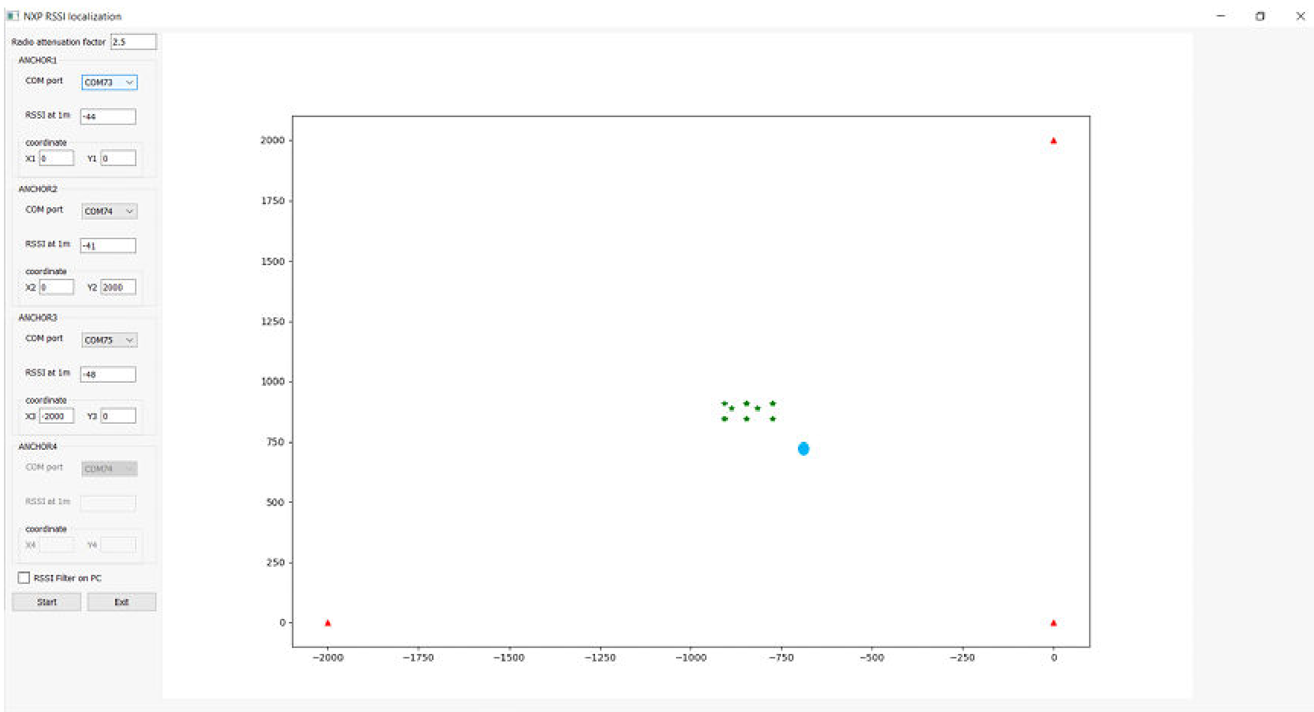

Figure 14. Advertising test 1-1

Table 1. Test 1-1 data

	Anchor1	Anchor2	Anchor3	coordinate	
				x	y
Distance(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

Table continues on the next page...

Table 1. Test 1-1 data (continued)

	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 Advertising test 1-2

The Tag is 2 meters away from Anchor1, the tag physical coordinate is in (1750, 1000).

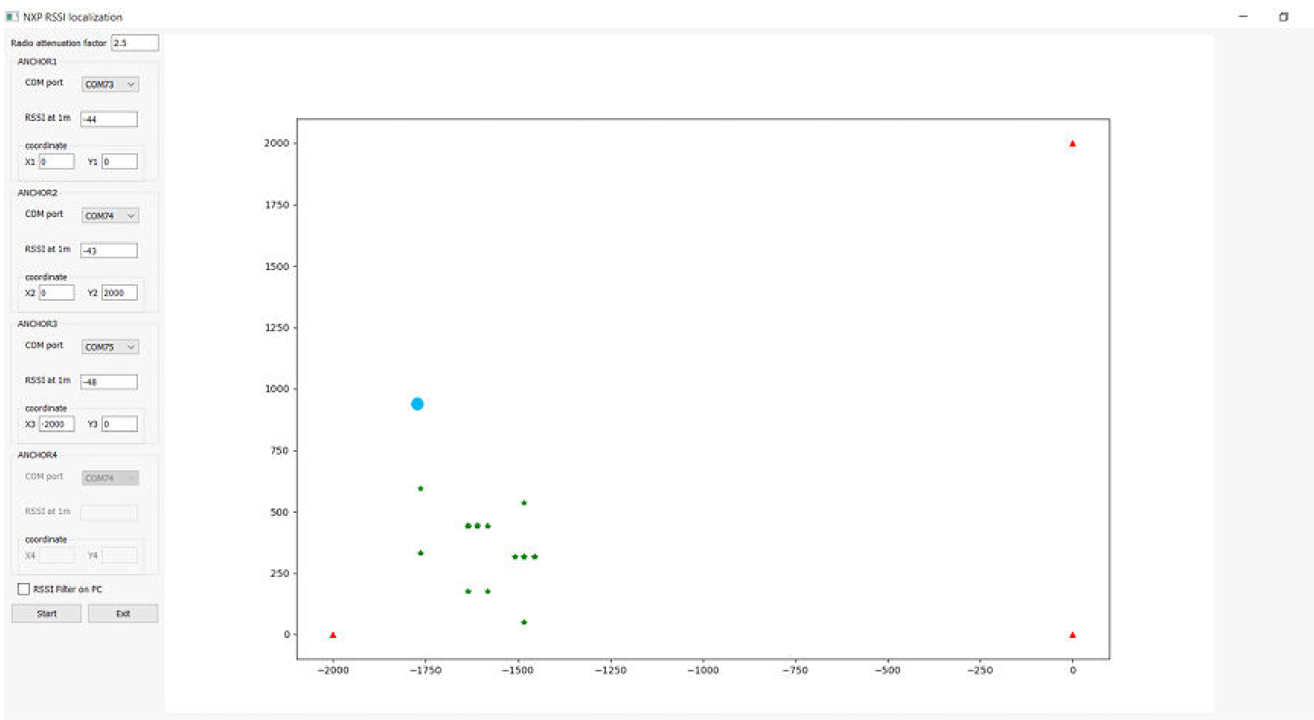

Figure 15. Advertising test 1-2

Table 2. Test 1-2 data

	Anchor1	Anchor2	Anchor3	coordinate	
				x	y
Distance(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

Table continues on the next page...

Table 2. Test 1-2 data (continued)

	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 Advertising test 1-3

The Tag is 3 meters away from Anchor1, the tag physical coordinate is in (1750, 1000).

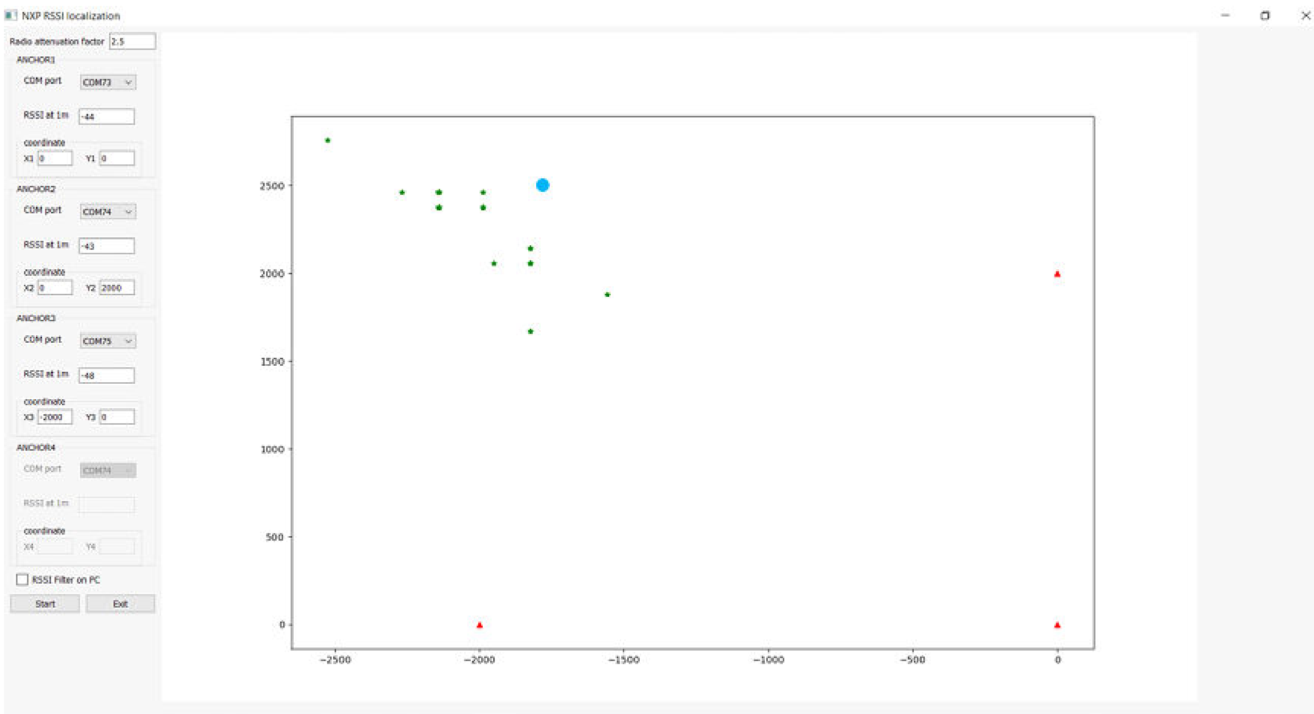

Figure 16. Advertising test 1-3

Table 3. Test 1-3 data

	Anchor1	Anchor2	Anchor3	coordinate	
				x	y
Distance(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
	2.5119	1.4454	1.7378	-1822.42	2055.115
	2.7542	1.3183	1.7378	-2141.42	2461.926

Table continues on the next page...

Table 3. Test 1-3 data (continued)

	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 Conclusion

In this application note, a simple three-point localization based on the Bluetooth LE RSSI ranging is implemented. The test results show that the accuracy is relatively high when the Tag node is close to the Anchor, but the accuracy decreases when the distance increases. This feature can be used in some applications. For example, when the distance is long (when the RSSI value is low), it can be used to identify the Tag entering the location region. When the distance is short (when the RSSI value is high), it can be used for localization. This can be realized by the KW38 MCU feasibly and effectively.

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

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