Chipcon Products from Texas Instruments

Z-Stack Location Profile User's Guide

Document Number: F8W-2006-0008

Texas Instruments, Inc.
San Diego, California USA
(619) 542-1200

Copyright © 2006 Texas Instruments, Inc. All rights reserved.

Version	Description	Date
0.1	Created.	06/28/2006
0.2	Modified to explain blast operation.	11/01/2006

Copyright © 2006 Texas Instruments, Inc. All rights reserved.

TABLE OF CONTENTS

Copyright © 2006 Texas Instruments, Inc. All rights reserved.

1. Location Profile

1.1 Introduction

The Location Profile is a sample application that demonstrates the use of the CC2431 location engine over ZigBee. This system is implemented as a ZigBee profile and can be discovered through ZigBee Device Object Service Discovery messages. This user's guide provides an overview of how the sample application is implemented, and shows the user how they may add the location profile to their existing ZigBee application. This user's guide should be used in conjunction with the Z-Stack Location Profile document.

1.2 Definitions

ZDO – ZigBee Device Object

AF – Application Framework

APS – Application Support Sub-layer

1.3 Blind Node Theory of Operation

The Blind Node is the device that contains the CC2431 location engine. The location engine estimates its position by using the coordinates of all responding reference nodes within direct radio range (up to 16) along with the average RSSI of messages sent to them. Reference Nodes are neighbor devices that are within radio range of the Blind Node. The Blind Node begins a find by broadcasting a sequence of blast message (Cluster ID: 0x0019) with radius one. After waiting the interval specified in its configuration, the blind node then broadcasts a position request (Cluster ID: 0x0011), also with radius one in order to limit the recipients (and thus responders) to the reference nodes in direct radio range.

Each Reference Node that hears the blast broadcasts will begin averaging the RSSI of the series of messages received. Upon receiving the position request, the Reference Node will send back to the Blind Node its position information along with the average received signal strength over the series of blasts.

There are two main modes of operation for the Blind Node in the sample Location Profile:

1) **Polled Mode**

This mode allows any node within the ZigBee network to request the position of the blind node on demand, and thus the Blind Node will only acquire the Reference Node data when it is requested using the Blind Node Request command (Cluster ID: 0x0013). It will then perform the required calculation, and then send a Blind Node Response (Cluster ID: 0x0014) back to the original requester.

When the Blind Node Request command is sent to a Blind Node, the request is processed in the **BlindNode_ProcessEvent()** event handler (which handles all the system and user events for the Blind Node location application). The application is notified of an incoming over-the-air message via the AF_INCOMING_MSG_CMD system event, and then the function **processMSGCmd()** is called to process the incoming message. If the incoming message has a Cluster ID that matches the Blind Node Request command, then it will start the process of acquiring the Reference Node information by calling **startBlast()**.

The first thing the BLINDNODE_FIND_EVT event handler does is to turn on the RxOnWhenIdle flag in the MAC capabilities flag of the device (this information can be pre-configured in the Node Descriptor structure defined in ZDConfig.c). In the case where a Blind Node is an end device, this is done in order for it to receive Reference Node information without having to poll for the actual data packet (using a Data Request packet). This also ensures that end devices can stay awake for the entire period of time when the acquisition of Reference Node information is taking place.

The Blind Node will then send out a series of 1-hop broadcast messages with the option **APS_TX_OPTIONS_SKIP_ROUTING** set to force the message to be sent directly to the intended Reference Node audience. Once the Blind Node finishes blasting it requests the Reference Node audience to send the average RSSI of messages just sent and waits the configured time for reference nodes to respond. Finally, it calls the function **FinishCollection()** to set the RxOnWhenIdle flag to false so that "normal" end device operation is once again adhered to.

The function **sendRsp()** is then called to process the received Reference Node data and calculate the Blind Node's position based on that information. If the Blind Node was able to collect data from at least config.minRefNodes number of nodes, the status **BLINDNODE_RSP_STATUS_SUCCESS** will be returned as part of the message. If not, the status **BLINDNODE_RSP_STATUS_NOT_ENOUGH_REFNODES** will be returned as part of the message. The response is sent to the device that was the originator of the Blind Node Request.

The entire process is started again when another Blind Node Request is received.

2) **Auto Mode**

In Auto Mode, the process for acquiring the Reference Node data is exactly the same as for polled mode, except this mode allows the Blind Node to start collecting Reference Node data automatically in a periodic fashion by timer or user event (such as a button press) without having to be manually asked by the host-system. By default, the Blind Node Response is then sent to the coordinator (short address 0x0000). However, the user can also configure the destination address.

The user can also set or request the configuration parameters for the Reference Nodes and Blind Nodes by using the following Cluster IDs:

Reference Node Configuration (Cluster ID: 0x0015)

Blind Node Configuration (Cluster ID: 0x0016)

Reference Node Configuration Request (Cluster ID: 0x0017)

Blind Node Configuration Request (Cluster ID: 0x0018)

1.4 Reference Node Theory of Operation

When the Reference Node receives a Reference Node Request, it invokes the function **AF_DataRequestMessage()** to send the Reference Node coordinate data (along with the received signal strength) directly to the requester. The parameter **APS_TX_OPTIONS_SKIP_ROUTING** is used to allow the message to be sent direct to the intended destination if the destination is a Blind Node and the logic for mandatory routing this message via the parent of an end device is bypassed.

1.5 Location Dongle Theory of Operation

The location dongle is an application that uses the Monitor Test (MT) System App Msg (**SYS_APP_MSG**) to send and receive Location Profile messages. This is typically the device that is attached to the host-system (e.g. a PC running the Location PC Application) and is used as the sink node for receiving Blind Node Responses and for configuring Reference Node and Blind Node setup parameters. The connection to the PC is via a serial port, and therefore a device built as a Dongle must have a serial port. Currently the only 2430 boards with a readily usable serial port are the EB boards. Therefore, the Sample Application for Dongle will only build as an EB device. Note that although the Dongle will build as an EndDevice, such a configuration is not recommended nor tested.

When the location dongle application receives a Blind Node Response message, it will pass the data to the host-system using the incoming message format defined in section 4.1.2 of the Z-Stack Location Profile document. Section 4.1.1 of that document defines outgoing messages (e.g. Blind Node Configuration Requests, Reference Node Requests, etc.).

Copyright © 2006 Texas Instruments, Inc. All rights reserved.

1.6 Sequence of Events Diagram

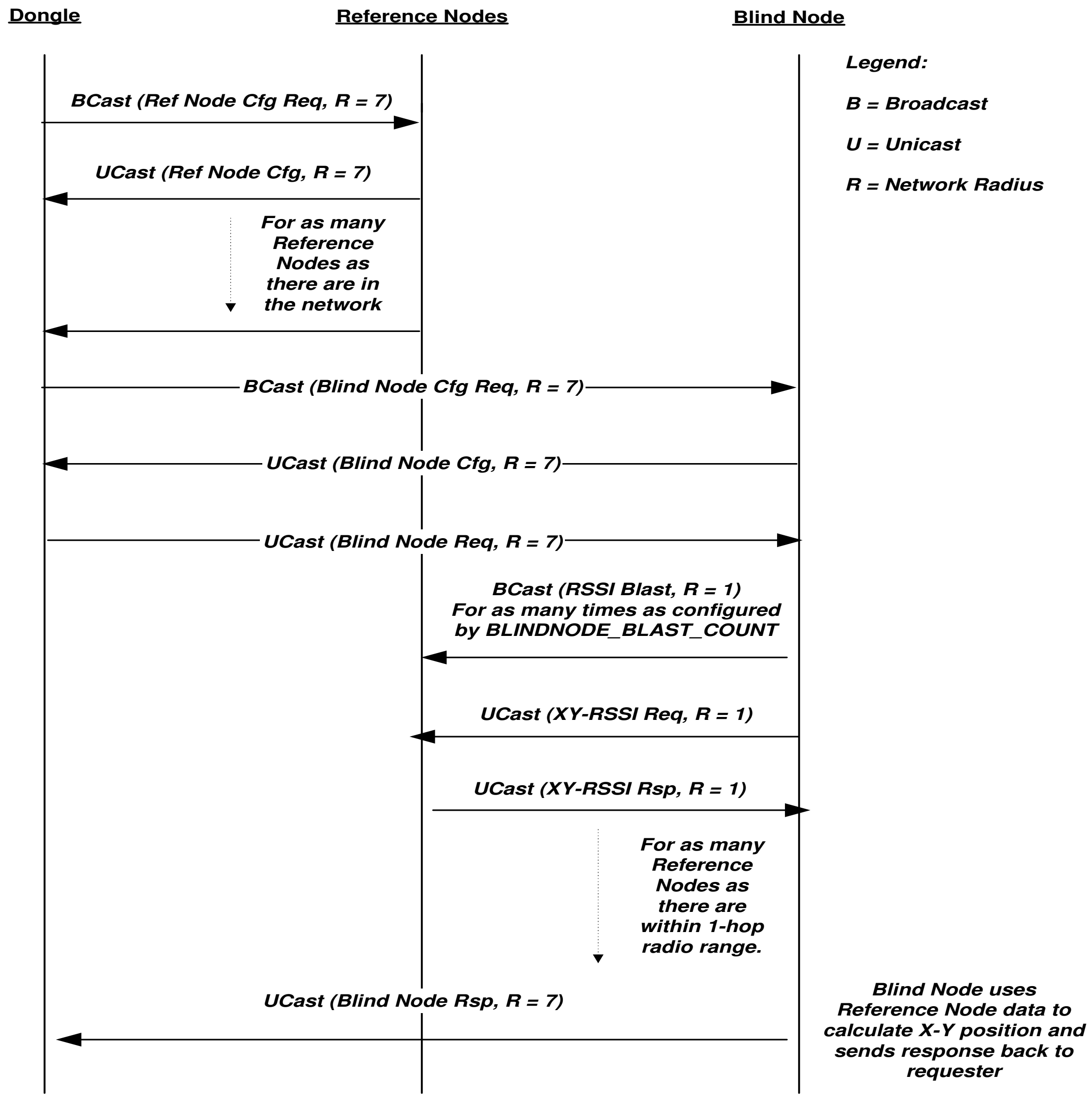
Dongle
Reference Nodes
Blind Node
Legend:
B = Broadcast
U = Unicast
R = Network Radius
BCast (Ref Node Cfg Req, R = 7)
UCast (Ref Node Cfg, R = 7)
For as many Reference Nodes as there are in the network
BCast (Blind Node Cfg Req, R = 7)
UCast (Blind Node Cfg, R = 7)
UCast (Blind Node Req, R = 7)
BCast (RSSI Blast, R = 1) For as many times as configured by BLINDNODE_BLAST_COUNT
UCast (XY-RSSI Req, R = 1)
UCast (XY-RSSI Rsp, R = 1)
For as many Reference Nodes as there are within 1-hop radio range.
UCast (Blind Node Rsp, R = 7)
Blind Node uses Reference Node data to calculate X-Y position and sends response back to requester

Figure 1. Location Profile Algorithm Sequence Diagram for a System with a Single Blind Node

Figure 1 shows acquisition of the Blind Node position being initiated by the Location Dongle. Afterward the Blind Node blasts 5 1-hop broadcast packets to the Reference Nodes and then requests the average RSSI and X-Y position. After the position of the Blind Node is calculated, the Blind Node position data is then sent back to the Location Dongle. The Location Dongle should be connected to a PC or some other equipment with an appropriate user interface in order to visually monitor the position of the Blind Node in relation to the Reference Nodes within the ZigBee network.

2. Key Parameters and Variables

2.1 config.mode

This parameter configures whether the Blind Node should use the polled mode or auto mode of operation. Valid values are NODE_MODE_POLLED or NODE_MODE_AUTO. These are defined in LocationProfile.h. By default, the location profile is set to use NODE_MODE_POLLED, but this can be configured by the user as part of the Blind Node Configuration parameters (e.g. via the Location Dongle application). A user input, such as a key press, can always be used to override the mode and generate a location find and report sequence.

2.2 config.cycle (only valid in Auto Mode)

This parameter configures the time period before the next acquisition of Reference Node data takes place. This is set to BLINDNODE_CYCLE_PERIOD by default. The duration of this is in units of 100 ms, and is derived as a formula: BLINDNODE_CYCLE_PERIOD = BLINDNODE_WAIT_TIMEOUT * BLINDMODE_REQ_RSPS * 2.

It is recommended that the above is the "minimum auto cycle time" – it is only logical that the Blind Node cannot start the next collection cycle before the first is finished. Since it could collect from up to 16 Reference Nodes, and then passing the best 8 to the location engine, the max cycle time (minimum config.cycle setting) would be BLINDNODE_WAIT_TIMEOUT * MAX_REF_NODES, (1600 ms or 1.6 s).

2.3 config.timeout

This parameter configures how long the Blind Node should wait for before the acquisition of Reference Node data is complete. This is set to BLINDNODE_WAIT_TIMEOUT * BN_TIME_INCR by default, where BLINDNODE_WAIT_TIMEOUT is set to 2, and BN_TIME_INCR is 100 ms.

2.4 config.minRefNodes

This parameter configures the minimum number of Reference Nodes required before the Blind Node deems it has enough Reference Node Data in order to calculate its position and send a valid response to the requestor. By default this is set to BLINDNODE_MIN_REF_NODES = 3, which is the minimum number of Reference Nodes that the Blind Node requires in order to estimate its position.

2.5 BLINDNODE_BLAST_COUNT

This parameter configures how many 1-hop blast broadcasts are generated in a row to the Reference Nodes before requesting the average RSSI for the blast sequence. By default this is set to 5. For desired results in auto mode, the user may have to adjust the config.cycle parameter accordingly. A suggested formula is given in Section 2.2.

2.6 BLINDNODE_MAX_REF_NODES

BLINDNODE_MAX_REF_NODES (by default set to 16) represents how many Reference Nodes the Blind Node can receive responses from.

3. Adding Support for the Location Profile

3.1 Overview

The Location Profile in incorporated into the SampleApp IAR-project in such a way that any of the 3 device types (Dongle, Reference Node, or Blind Node) can be built. The SampleApp provides the added functionality of a simple light switch and light controller to show how the user's application would co-exist with the Location Profile.

3.2 Adding Location Profile Source Files

1) Notice that the Location Profile files (LocationProfile.c/h, RefNode.c/h, and BlindNode.c/h) are placed in the SampleApp\Source sub-directory. In the App folder of the SampleApp project, only the LocationProfile.c/h files had to be added (see Figure 2).

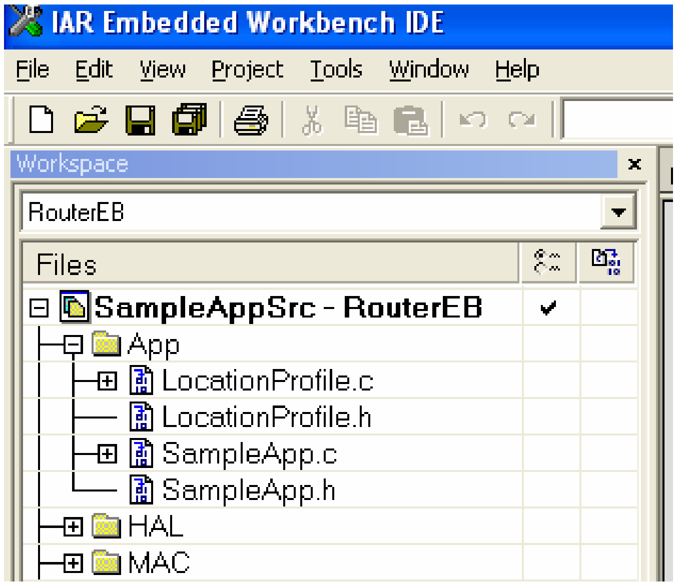

Figure 2. Adding Reference Node Source Files

2) Notice the conditional compilation in SampleApp.c when adding Location Profile tasks:

```
#if defined ( LOCATION_REFNODE )
  osalTaskAdd( RefNode_Init, RefNode_ProcessEvent, OSAL_TASK_PRIORITY_LOW );
#endif
#if defined ( LOCATION_BLINDNODE )
  osalTaskAdd( BlindNode_Init, BlindNode_ProcessEvent, OSAL_TASK_PRIORITY_LOW );
#endif
#if defined ( LOCATION_DONGLE )
  osalTaskAdd( LocDongle_Init, LocDongle_ProcessEvent, OSAL_TASK_PRIORITY_LOW );
#endif
```

3) Choose which device gets built by un-commenting the corresponding include in LocationProfile.h:

```
// Uncomment the definition of the LocationProfile device to build.
//#define LOCATION_REFNODE
#define LOCATION_BLINDNODE
//#define LOCATION_DONGLE
```

Copyright © 2006 Texas Instruments, Inc. All rights reserved.

4. Reference Documents

1. Z-Stack Location Profile (F8W-2006-0002)Z-Stack Device Object API (F8W-2004-0009)
2. Z-Stack Device Object Programmer's Guide (F8W-2004-0008)
3. CC2431ZDK Z-Location Engine User Manual
4. CC2431ZDK Quick Start Guide
5. CC2431ZDK User Manual
6. Taubenheim, David., Kyperountas, Spyros., Neiyer, Correal. *"Distributed Radiolocation Hardware Core for IEEE 802.15.4"*. Motorola Labs. December 8, 2005.

Copyright © 2006 Texas Instruments, Inc. All rights reserved.