ZIC2410 ZigBee Application Note
Serial Communication (COM) Device Implementation

1. Purpose
ZigBee2006_DK_COM provided by California Eastern Laboratories is an example of how to set up the software to configure a ZigBee ZIC2410 for a Serial Communication application. In this application, when the device receives the data as much as defined length from serial port, it sends this data to Coordinator.

This document describes ZigBee2006_DK_COM.

2. Device Configuration Setting
The first step is to set the device configuration in the Profile-Builder. The configuration values that are set here, are generated as a ZDO_CONF.h file through the running of the 'build' function. In addition, ZZDO_INIT_CONFIGURATION() function in ZZDO .c, which is generated by the 'build', refers these configuration values to ____ and determines the characteristic of the device.

Major setting values are as follows.

![Figure 1 Configuration Setting Window of the Profile-Builder]
Please refer to Figure 1 for the next set of instructions.

2.1 Logical Type Setting
Selects the type, i.e. how the device is to be operated; choose from among Coordinator, Router or EndDevice. In the case of either Router or EndDevice, it is joined to the network by using ZNWK_JOIN_REQ() function in ZUSER_MAIN() function; the value of JoinAsRouter parameter is set to 1 (for Router) or 0 (for EndDevice).

2.2 Device Type Setting
Sets the Device Type as FFD (Full Function Device) or RFD (Reduced Function Device). When setting the Logical Type to Router or Coordinator, the Device Type should be set to FFD.

2.3 RxOnIdle Setting
To choose between the RXOnIdle state and the Power-Down state, check the RxOnIdle box for the RXOnIdle state and leave the box empty for the Power-Down state.

2.4 Channel List Setting
Sets the potential channels in which the device can operate. When each bit is ‘1’, corresponding channel is activated; multiple channels can be activated for each device. This value is referred when the Coordinator initiates Network formation or Routers/EndDevices are joined to the Network.

2.5 Network Tree Information(MaxDepth, MaxRouter, MaxChildren)Setting
Sets the nwkMaxDepth, nwkMaxChildren, nwkMaxRouter values to determine the Network Tree structure.

CAUTION: All Coordinators, Routers and EndDevices in the same network should have the same values in the Network Tree Information.
3. Profile Configuration

Generate profile to use by Profile-Builder. Following example shows Home Automation (HA) Profile and uses SerialComm device (SerialComm device is not defined in the standard, but rather by California Eastern Laboratories.)

There are one input cluster and one output cluster in SerialComm device.

![Diagram of Profile Configuration Window of the Profile-Builder]

**Figure 2 Profile Configuration Window of the Profile-Builder**

### 3.1 ComCtrl Cluster

It an input cluster (Cluster ID = 0x0800) defined in SerialComm device and it determines the length of string data, which is included per a packet when performing serial communication. It includes “Len” attribute.

“Len” Attribute is as follows.

- Attribute ID: 0x0000
- Attribute Data Type: UINT8 (8-bit Unsigned Integer)
- Attribute Data: The length of string data. When ‘0’, it means serial communication does not work.
3.2 **ComStr Cluster**

It an output cluster (Cluster ID = 0x0801) defined in SerialComm and it sends the string data inputted by serial port to coordinator. It includes "Content" attribute.

4. **ZUSER_INIT()**

It initializes to run application. For initialization in detail, refer to the ZIC2410 User's Guide Application Development Guide (Doc #0005-05-08-02-001) document.

5. **ZUSER_MAIN()**

A user can modify this function. The application function is implemented in this function.

This example code implements the following functions;

- Searches network and then join to the network.
- Initializes the flash area which stores data.

5.1 **Initialize Flash Area**

This design example uses some parts of flash area to store data. Basic network information is stored in this area and is therefore protected from data loss when Power-Off. When Power-On, the program needs to determine whether to recall the stored data or not as considering the device's specific mode of operation. The NIB.DEVTYPE field to shows the current mode of operation for the device and has the following value.

**NIB.DEVTYPE**

- C: Operates as Coordinator.
- R: Operates as Router.
- E: Operates as EndDevice.
- N: Default state.

When there is an external interrupt transition from ‘0’ to ‘1’ on GPIO P0_7 (GP07) (this can be done by pressing external interrupt 1 switch on the Evaluation board), the ZSYS_DATABASE_CLEAR() function is called. This function initializes flash area by writing NIB.DEVTYPE as ‘N’.

```c
if(INT_EXT1)
{
    INT_EXT1 = 0;
    if(GP07 == 0) ZSYS_DATABASE_CLEAR();
}
```

5.2 **Join To Network**

When a device is in the default status (NIB.DEVTYPE==‘N’), the device is trying to join to Coordinator. At this moment, if it is a Router, JoinAsRoute is set to ‘1’ and if it is an EndDevice, JoinAsRoute is set to ‘0’.

When it is joined successfully, send END_DEVICE_ANNOUNCE command and store the network information in the flash area by calling ZSYS_DATABASE_SAVE() function.

In case of Router, let it operate as Router by calling ZNWK_START_ROUTE_REQ() and ZNWK_PERMIT_JOIN_REQ() function. When join fails, it tries again until it succeeds.
if(NIB.DEVTYPE == 'N')
{
  ...  
  ZNWK_JOIN_REQ();    // Perform JOIN  
  ...  
  // Send END_DEVICE_ANNOUNCE  
  ...  
  ZSYS_DATABASE_SAVE();  // Store data in the flash area.  
}

### 5.3 Send Data Inputted by Serial Port to Coordinator

If a device is successfully joined to Coordinator or Router and "Len" attribute of ComCtrl is not '0', it sends the data to the Coordinator with a length, indicated by its attributes.

```c
if(ZSYS_UART1_GET(&KeyIn))
{
  if( (NIB.DEVTYPE != 'N') && (ATT_CtrlCom.Value) )
  {
    ATT_StrCom.pStr[StrIdx] = KeyIn;
    ...  
  }
}
```

### 5.4 User Function

The function which a user needs can be added in this example.

### 6. ZUSER_APS_DATA_IND()

When stack library receives APS data packet, it informs the status to the user by ZUSER_APS_DATA_IND function. A user performs the process for packet in this function. Example code extracts the information of Profile ID, Cluster ID and Attribute Data Type by analyzing the packet. Extracted Cluster ID means "ComCtrl" Cluster and when other information is matched to the format defined in the profile, String Data is determined according to the attribute data.

The operation based on the attribute data
- 0 : Serial Communication does not work.
- 0x01 ~ 0x3F : The length of data string which is included to one packet.

### 7. ZUSER_OOB_RF_ISR()

When received data is OOB (Out-Of-Band), RF Interrupt Service Routine (ISR) of stack library calls ZUSER_OOB_RF_ISR() function.

The received packet should be processed quickly by the application, and done by utilizing the ZUSER_OOB_RF_ISR() function.

When returned value is 1, it indicates that the received packet has already been processed successfully and the packet is ignored without processing in stack library. When returned value is 0, it indicates that the received packet has not yet been processed. The stack library processes the packet again and then calls ZUSER_OOB_DATA_IND() function.
8. **ZUSER_OOB_DATA_IND()**
   When received data is OOB (Out-Of-Band) packet and the return value of ZUSER_OOB_RF_ISR() function is 0, stack library calls ZUSER_OOB_DATA_IND() function.

9. **ZUSER_NWK_CON()**
   When NWK Request primitive is called, there is primitive which is not returned immediately (e.g. NWK-ROUTE-DISCOVERY.req). This primitive causes the confirmation after corresponding operation for the defined time and then it informs the status to the user by ZUSER_NWK_CON() function.

10. **ZUSER_NWK_JOIN_IND()**
    When a device is joined, stack library calls ZUSER_NWK_JOIN_IND() function.

11. **ZUSER_NWK_ROUTE_ERR_IND()**
    When routing error is caused, stack library informs this status to the user by ZUSER_NWK_ROUTE_ERROR_IND() function.

12. **ZUSER_SYNC_IND()**
    When Sync is caused, stack library informs this status to the user by ZUSER_SYNC_IND() function.

13. **ZUSER_LEAVE_IND()**
    When leave is caused, stack library informs this status to the user by ZUSER_LEAVE_IND() function.

14. **REVISION HISTORY**

<table>
<thead>
<tr>
<th>VERSION</th>
<th>DATE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>20June08</td>
<td>Released</td>
</tr>
</tbody>
</table>

Information and data presented here is subject to change without notice. California Eastern Laboratories assumes no responsibility for the use of any circuits described herein and makes no representations or warranties, expressed or implied, that such circuits are free from patent infringement.

© California Eastern Laboratories 6/08 Version 1.0