Accelerate your ZigBee design using the new TI ZigBee Processor

Abstract

- To simplify the addition of ZigBee to an application, TI has created a new concept called Z-Accel.
- Z-Accel, as the name implies, is intended to accelerate the development of a ZigBee solution.
- Z-Accel consists of two parts: a ZigBee processor, CC2480, and an application MCU, any MSP430.
- This presentation will explain the concepts behind Z-Accel, describe the features of the CC2480 and ends with a demo experience using eZ430-RF2480.
What is Z-Accel and CC2480?

- Z-Accel is a ZigBee Network Processor that communicates with any MCU via an SPI or UART interface
- CC2480 is the first-generation ZigBee-compliant network processor in the Z-Accel family
- Z-Accel allows customers to work with their favorite MSP430
- Z-Accel provides complete ZigBee functionality without having to learn the complexities of a full ZigBee stack

---

CC2480 Advantages

- **Wireless Network Processor, “Smart Transceiver”**
  - Z-Accel: ZigBee Network Processor
- **Accelerate your ZigBee Connectivity**
  - Easy Add-On to existing solution
  - ZigBee Connectivity made easy to design and develop
- **Reduce development time and time to market**
- **Enable ZigBee “out-of-the-box” and “black box” network capabilities**
- **Enable interoperable devices where desired by manufacturers using private profiles**
Z-Accel – Key Features

• **Reduce Development Complexity**
  - Preprogrammed / preconfigured ZigBee Node

• **Easy to Deploy**
  - Accelerate ZigBee Connectivity for existing solution
  - Easily added as serial peripheral
  - Simple API applied for network commissioning

• **Easy to Use**
  - Split Application and Network processor
  - Application Processor running Application Framework & Profile
  - Physical Layer
    - Serial Transport / Full Duplex
  - Remote Procedure Call (RPC) Layer
    - Layered Interface Messages: System, Configuration and Data

• **Standardized protocol**
  - Network Infrastructure reuse by multiple applications
  - Manufacturer Specific Profiles to enable interoperability whenever possible

---

**ZigBee® GLOSSARY**

• **A Application Profile** is an agreement on a series of messages defining an application space
  - **Public Application Profile**
    - Interoperable application profile developed by the ZigBee Alliance
  - **Manufacturer Specific Profile**
    - Private application profile developed by a company to operate a ZigBee Device
    - Limits application interoperability to devices that share this profile
  - Profile ID a unique identifier assigned by the ZigBee Alliance

• **A Cluster** is a message identifier for exchange of information within an application profile
  - Endpoint a communication entity within a device which permits support for a specific application
  - Each binding supporting a specific application profile
  - Each message type is represented by a cluster within the profile

• **Binding/s** is a way of connecting devices between two endpoints
  - Commissioning the task of configuring devices and networks to achieve the needs of a specific installation

• **Application Object** is software at an endpoint that controls the ZigBee Device
  - A single ZigBee Device node supports up to 240 application objects
  - Each application objects supports endpoints numbered between 1 and 240
  - Endpoint 0 reserved for the ZigBee Device Object (ZDO)
ZigBee® GLOSSARY

- **ZigBee Device Object (ZDO)** defines the logical role of a device within the network:
  - Device type: Coordinator, Router, End device
  - Initiates and/or responds to binding and discovery
  - CC2480 is running the ZDO endpoint
- **PAN ID** – a unique Personal Area Network Identifier; Controlled by User and Application development
- **Device Description** is a description of a specific device within a profile
- **Device Discovery** can find the identity of devices on active channels within the PAN
- In a Mesh network the routing of messages is performed as a decentralized, cooperative process involving many peer devices routing on each other's behalf
- **Service Discovery** is the ability to determine supported services on given devices within the PAN (e.g., USB Enumeration)

---

**ZigBee-2006**

- **Benefits to Consider**
  - Extremely well tested by a variety of companies
  - Base of products and networks on market and in use today!
  - Many certified stacks and silicon providers available
  - Simple = less code & less overhead

![ZigBee network diagram](image-url)
ZigBee-2007 Feature Set

**ZigBee**
- Tree Addressing
- AODV Routing
- Backup Tree Routing
- Fragmentation
- Frequency Agility
- Basic Group Addressing
- Security

**ZigBee PRO**
- Stochastic Addressing
- AODV Routing
- Many to One / Source Routing
- Asymmetric Link Handling
- Fragmentation
- Frequency Agility
- Basic Group Addressing
- Limited Broadcast Addressing
- Security
- High Security

---

**CC2480 Application Design Process**

- Customer has a new application with no defined ZigBee Profile
  - Define devices in the system
  - Map to Logical Devices
  - Define the Profile by partitioning the devices
  - Define the Device Descriptions within each Profile
  - Define the Clusters and indicate which are input and output from each Device Description
  - Define the Security solution
  - Define the Commissioning Process
  - Package the solution:
    - Define the descriptors, deploy the applications over endpoints
    - Integrate and test the solution.
    - Solve Deployment issues.

Profiles

CC2480 only supports Manufacturer Specific Profiles (Private Profile)
- Privately developed by individual manufacturers
- CC2480 only supports Stack Profile ZigBee 2006
- A set of devices required in the application area
- A set of clusters to implement the functionality
  » A set of attributes to represent device state
  » A set of commands to enable the communication
- A profile ID must be unique and use a ZigBee allocated profile identifier
- Commercial products developed using this profile must undergo network capable testing

Endpoints

- An Endpoint (EP) is the address of your application running on Host Client
  - EP 0 is ZDO application running on CC2480 Up to 240 additional applications can be supported on a device
  - Each application on an endpoint is identified by its own Endpoint Structure containing the Cluster list

- This device contain one EP (ZDO EP is on CC2480)
- Each EP contains an application
Clusters

- Commands sent or received over the network are called Clusters (data types)
- Cluster may be inputs or outputs from a device
- No ZigBee Cluster Library (ZCL) support with CC2480
- Cluster List must be registered through communication entities called EndPoint

CC2480 Target Application Example
Hardware Flexibility

<table>
<thead>
<tr>
<th>Path 1</th>
<th>Path 2</th>
<th>Path 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small footprint</td>
<td>Flexible</td>
<td>Ultra-low power</td>
</tr>
<tr>
<td>High integration</td>
<td>Easy to use</td>
<td>Well-known radio</td>
</tr>
<tr>
<td>Location Engine optional</td>
<td>Reduces time to market</td>
<td></td>
</tr>
</tbody>
</table>

**Customer Application**
- CC2430/1ZDK
- Any MSP430
- CC2480
- CC2420/CC2520

**Z-Stack™ ZigBee stack**

**Radio**

**Development Kits**
- CC2430/1ZDK
- eZ430-RF2480
- CC2420/CC2520 + MSP430/2ZDK

---

Z-Accel: (ZigBee Processor)
eZ430 -RF2480 = MSP430 + CC2480

---

ZigBee System Architectures

- **Layer Solution**
  - Application Profile: ZigBee 2006, ZigBee 2007, ZigBee 2007 PRO
  - Stack Profile: ZigBee 2006, ZigBee 2007, ZigBee 2007 PRO
  - MAC Layer: IEEE 802.15.4
  - Physical Layer*: CC2430/CC2420, CC2520

- **PP – Public Profile**
  - MSP – Manufacture Specific Profile

**ATC 2008**

MSP430 Advanced Technical Conference
• ZigBee products are a combination of Application, Logical, and Physical device types.
• The communication between applications is based on either a Public Profile (PP) or a Manufacturer Specific Profile (MSP).
• The public Home Automation profile (HA) provides the support for several device types (e.g. lamp, switch, thermostat, etc.)
CC2480 – Mesh Network Devices

- ZigBee Coordinator
  - Starts the Network
  - Routes packets
  - Manages security
  - Associates Routers and End Devices
  - Example: Sink

- ZigBee Router
  - Routes packets
  - Associates Routers and End Devices
  - Example: Source

- ZigBee End Device
  - Sleeps most of the time
  - Can be battery powered
  - Does not route
  - Example: Source

- Devices are configured for their network function
- Coordinator can be removed

**Remark:**
- Temperature Sensor is not calibrated

---

### ez430- RF2480 Sensor Monitor
PC application running on PC connected to sink via Virtual COM Port (VCP)

- ZASA:
  - ZigBee Accelerator Sample Application

- Remark:
  - Temperature Sensor is not calibrated
**Z-Accel - Application Model**

- Configure Application Device as ZigBee Logical Device
  - Register your Application Devices and End Points
- Join Network as a configured ZigBee Logical Device mapped as an Application Devices Types:
  - Source (Router or End Device)
    - Creator for periodic sensor reports (Temperature, Battery). Transmit data to one or more Sink Devices to which it is bound
  - Sink (Coordinator)
    - Destination for Data. Receive data from one or more Source Devices

**Z-Accel Architecture**

- Host processor (MSP430)
  - HAL
  - Application
  - Simple API
    - ZACC RPC

- ZigBee Processor (CC2480)
  - ZigBee Processor Server
    - Simple API
    - Z-Stack API
  - Z-Stack
  - Radio
Simple API - SAPI

Z-Accel support TI's Simple API. Simple API has only 10 API calls to learn, drastically simplifying the ZigBee application development.

- Simple API simplifies:
  - Device Configuration
  - Commissioning of Networks
  - Binding of Devices
  - Data Transfer
### SAPI Overview Functions and Callbacks

<table>
<thead>
<tr>
<th>API</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>zb_SystemReset</td>
<td>Resets network</td>
</tr>
<tr>
<td>zb_StartRequest</td>
<td>Starts network</td>
</tr>
<tr>
<td>zb_PermitJoiningRequest</td>
<td>Allow nodes to join network</td>
</tr>
<tr>
<td>zb_BindDevice</td>
<td>Establish a binding (connection)</td>
</tr>
<tr>
<td>zb_AllowBind</td>
<td>Allow binding request</td>
</tr>
<tr>
<td>zb_SendDataRequest</td>
<td>Send data</td>
</tr>
<tr>
<td>zb_ReadConfiguration</td>
<td>Read configuration parameters</td>
</tr>
<tr>
<td>zb_WriteConfiguration</td>
<td>Write configuration parameters</td>
</tr>
<tr>
<td>zb_GetDeviceInfo</td>
<td>Get current address, PAN ID etc</td>
</tr>
<tr>
<td>zb_FindDeviceRequest</td>
<td>Search for a device on the network</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Callbacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>zb_StartConfirm</td>
</tr>
<tr>
<td>zb_AllowBindConfirm</td>
</tr>
<tr>
<td>zb_SendDataConfirm</td>
</tr>
<tr>
<td>zb_ReceiveDataIndication</td>
</tr>
<tr>
<td>zb_FindDeviceConfirm</td>
</tr>
<tr>
<td>zb_HandleKeys</td>
</tr>
<tr>
<td>zb_HandleOsalEvent</td>
</tr>
</tbody>
</table>

For more details about the SimpleAPI see ZACC documentation.

### CC2480 Digital IF SPI Transport to MSP430F2274

- **Sensor IF**
  - Analog Input 1
  - Analog Input 2
- **Reset (RESETn)**
- **4 Wire GPIO IF**
  - GPIO[0..3]
- **3 Wire SPI**
  - MRDY
  - Optional with 3-Wire
- **Configuration GPIO IF**
  - Option 1: Dynamical Configuration
    - CFG0
    - CFG1
  - Option 2: Static Configuration
    - CFG0: 32 kHz XTAL
      - Pulled to VCC
      - No 32 kHz XTAL
      - Pulled to GND
    - CFG1: SPI Transport
      - Pulled to VCC
      - UART Transport
      - Pull to GND
In the sample_app.cfg file, you’ll need to select the channel your PAN will operate on.

Remark: Selected Channel
PAN ID Configuration

While we’re at it, let’s also select the PAN ID and Profile ID in the same file...

If PAN_ID = 0xFFFF and device = Coordinator:
- Device uses IEEE address to choose a PAN_ID (last 2 bytes)

If PAN_ID = 0xFFFF and device = Router or End Device:
- Device will join any available PAN

If the PAN_ID ≠ 0xFFFF and device = Coordinator:
- Device will use the set value for the PAN_ID

If the PAN_ID ≠ 0xFFFF and device = Router or End Device:
- Device will ONLY join a PAN that has this PAN_ID

Packet Sniffer

- Displays RF data packets on the air
- Advanced filters
- Timeline display - Shows all nodes in a network
- Can be connected to all nodes in a network
- Supports IEEE 802.15.4/Zigbee packets
ZigBee Demonstration Kit

- eZ430-RF2480
- Based on the CC2480 – A ZigBee Network Processor
- 3x Target Boards
- 2x Battery Boards
- 1x USB debug dongle
- Comes with a simple application that demonstrates:
  - Command interface
  - Chip configuration
  - Simple API
  - Basic network operations
- A PC application shows the network topology
- Price $99

Demo – Starting a PAN, eZ430 – RF2480

- Start PC Demo Application
- Start Sink, then Source Router then Source End Device
  - Observe PAN traffic
Appendix

Detailed Information
- Profiles/Clusters/Endpoints
- RPC –Remote Procedure Call
- UART Transport Interface

Descriptor

- Assign a Device ID & Version
- Specify Profile ID
- Specify Endpoint ID
- Initialized in AppReset()
- Must be registered with CC2480

• Endpoint Descriptor provide information about the application like:
  - Number of defined Clusters
  - Number of defined Endpoints
• zb_AppRegisterRequest(srcEP)
  - Allows CC2480 to know the cluster information for each endpoint
  - Simplifies handling of messages to the node by differentiating by endpoint
**PAN Formation (Sink: Coordinator)**

- Application processor – MSP430

- Register Device
  - (Coordinator)

- ZASA
- SAPI
- Register Device
- Network Form
- SREQ/SRSP
- AF Register
- Start Confirm
- SREQ/SRSP
- AREQ

- A unique PAN ID is determined either from the IEEE address dynamically or statically
- One channel is selected
- Process fully automated once started

**ATC 2008**

**Pan Discovery/Join/Bind**

- (Source: Router or End Device)

- Application processor – MSP430

- Register Device
  - (Router/EndDevice)

- ZASA
- SAPI
- Register Device
- Register Device confirmed
- SREQ/SRSP
- App & Start Req
- SREQ/SRSP
- Start Confirm
- AREQ
- SREQ/SRSP
- Bind Request
- Bind Confirm
- AREQ

- A unique PAN ID is determined either from the IEEE address dynamically or statically
- One channel is selected
- Process fully automated once started

**ATC 2008**
Z-Accel Transport Overview:
Remote Procedure Calls (RPC)

CC2480's 10 Simple API calls are abstracted to
4 Physical Transport Layer Command Types

- **Z-Accel RPC Command Types**
  - **AREQ** – Asynchronous Request
    - Callback Events
    - Function call with return value
  - **POLL** – Polling for Data
    - Retrieve Queued Data
  - **SREQ** – Synchronous Request
    - Immediate Response
  - **SRSP** – Synchronous Response
    - Response to a SREQ command

General Frame Format (GFF) - SPI

<table>
<thead>
<tr>
<th>Frame Format Transport Packet</th>
<th>CMD0</th>
<th>CMD1</th>
<th>Data Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Data Field</td>
<td>0-252</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame Format Transport Packet</th>
<th>CMD0</th>
<th>CMD1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>1</td>
<td>0-7</td>
</tr>
<tr>
<td>CMD0</td>
<td>0-31</td>
<td>0-127</td>
</tr>
</tbody>
</table>

- **CMD 0 – Command Type**
  - [7:5] – Type
    - (AREQ, POLL, SREQ, or SRSP)
  - [4:0] - SubType
    - (SAPI, SYS, AF, ZDO, Reserved)
- **CMD 1 – Command ID**
  - CC2480 Application Interface Commands

**Note:** Length byte is the length of the data field excluding itself and commands
General Frame Format (GFF) - UART

Frame Format Transport Packet

<table>
<thead>
<tr>
<th>SOF</th>
<th>GFF</th>
<th>FCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3-253</td>
<td>1</td>
</tr>
</tbody>
</table>

- **CMD0:**
- **Type & Sub Type:**
- **CMD1:**
- **Command ID**
- **Note:** Length byte is the length of the data field excluding itself and commands

**Remark:** UART Transport not supported on eZ430 – RF2480

CC2480 Digital IF UART Transport to MSP430F2274

- UART Transport
  - Not supported on eZ430 – RF2480
  - Reference Design: How To
  - VCP – Virtual COM Port
  - Used with PC Demo tool

**Remark:** UART Transport not supported on eZ430 – RF2480

- VCP – Virtual COM Port
- Used with PC Demo tool
CC2480 Digital IF for Multi Drop SPI

- CC2480 Errata
  - Multi Drop SPI will require external 3-State Buffer for MOSI

Thank you