# 0011-00-16-10-000 ZICM35xSPx Software Design Guidelines



### INTRODUCTION

This Application Note describes how to configure the Ember Desktop AppBuilder application to create software to optimize the performance of the ZICM35xSPx family of modules from California Eastern Laboratories (CEL). There are two types of modules: ZICM35xSP0 and ZICM35xSP2. Configurations for both modules are discussed. Each module has different regulatory restrictions. A general sample AppBuilder configuration is provided with a discussion on how to implement the restrictions within the generated software from AppBuilder.

#### REQUIREMENTS

- 1. Ember Desktop Software, Version 3.0 b612 or newer
- 2. IAR ARM Workbench, Version 6.40
- 3. Ember Stack, any version



# TABLE OF CONTENTS

Introduction	1
Requirements	1
Chapter 1 – Radio Configuration	3
Chapter 2 – GPIO Configuration	4
ZICM35xSP0	4
ZICM35xSP2-1x	4
ZICM35xSP2-2x	5
Chapter 3 – Regulatory Power Restrictions	8
Implementing FCC Power Restrictions For Ember Stack 4.7 GA And Later	8
Implementing Regulatory Power Restrictions For Pre-Ember Stack 4.7 GA	9
FCC Restrictions Software Design	10
Implementing FCC Design into AppBuilder Generated Code	11
emberAfStartSearchForJoinableNetworkCallback	12
emberAfScanErrorCallback	12
emberAfMainTickCallback	13
emberAfJoinableNetworkFoundCallback	13
emberAfStackStatusCallback	14
Choosing To Use Co-Located Transmitters	14
References	15
Revision History	15



### **CHAPTER 1 – RADIO CONFIGURATION**

The radio configuration for the ZICM35xSPx family of modules is preconfigured during manufacturing through the use of tokens. In case they are inadvertently erased, the factory-default token values required for proper radio configuration are as follows:

Token	ZICM35xSP0	ZICM35xSP2
TOKEN_MFG_PHY_CONFIG	0xFF26	OxFFFD

If the end application requires a common software image to be used for both the ZICM35xSP0 and ZICM35xSP2 Modules, it is possible to use <code>TOKEN\_MFG\_PHY\_CONFIG</code> to programmatically determine whether the device is a ZICM35xSP0 or ZICM35xSP2 Module.

txPowerModes for emberSetTxPowerMode and mfglibSetPower (used in the EmberZNet API):

Value	ZICM35xSP0	ZICM35xSP2
Numerical value (used in nodetest and mfglib)	1	2
Constant value (used in EmberZNet stack)	EMBER_TX_POWER_MODE_BOOST	EMBER_TX_POWER_MODE_ALTERNATE

For the ZICM35xSP0 Module, TOKEN\_MFG\_PHY\_CONFIG will automatically enable boost mode if power steps 4 - 8 are selected.

When configuring AppBuilder, make sure the *Use Token* option for the *Power Mode* is selected on the Stack configuration tab. This setting tells the Ember software to use the manufacturing settings for the power mode when configuring the radio. See example in Figure 1.

AppBuilder					
e Help					
New Configuration	n 🖾				
tack: EmberZNet 5.	.0 GA EM35X, app:	ZCL Application Framework V2 De	vice name: ZigBee		< Preview
irectory for generate	ed files: 👸 C:\L	Jsers\emar\Ember\EmberZNet5.0.0-G	A\EM35x\.		🕞 🕨 Generate
		configuration I HAL configurat	ion 🐟 Plugins 🥵 Callback con	figuration & Includes	
<ul> <li>Network config</li> </ul>					
Network cont	-				
Name		ZigBee Device Type	Security Type		Make Default
Primary (de	fault)	Coordinator or Router	НА		
	BA: Latest	•			
	AINTREE: Latest	-			
	A: Latest	•			
HC: HC	C: Latest	-			
SE: SE	: Latest	•			
ZLL: ZL	L: Latest	•			
• Radio configura	ation				
Power mode:	Use token				
(	🖱 Use API 🗌 Er	nable boost power mode 🔲 Enable t	he alternate transmitter output		
End Device Pol	I Configuration				
Debug printing	1				
Auto-generate upo					

Figure 1. Recommended AppBuilder Radio Configuration Setup



It is recommended that Ember Stack Version 4.7.2 or newer be used in order to be compatible with the latest versions of the Ember Desktop AppBuilder application in regards to the tokens.

# **CHAPTER 2 – GPIO CONFIGURATION**

The GPIO PC5 needs to be configured for the ZICM35xSP2 Module only.

#### ZICM35xSP0

For the ZICM35xSP0 Module, PC5 is not required for radio operation. Therefore, there are no restrictions on the configuration.

#### ZICM35xSP2-1x

For the ZICM35xSP2-1x Module, PC5 should be set to *Alt out (PP)*. See Figure 2 below. Internally within the module, it serves as the logic-level control for the PA. PC5 is not connected to any module castellation pin.

AppBuilder	• ×									
File Help										
() *New Configuration (3)	- 0									
Stack: EmberZNet 5.0 GA EM35X, app: ZCL Application Framework V2 Device name: ZigBee										
Directory for generated files: 🚨 C:\Users\emar\Ember/EmberZNet5.0.0-GA\EM35x\.										
🔥 ZCL cluster configuration 🔝 Stack configuration (Internation Application A										
✓ Platform configuration										
Platform: EM357  Host: Unknown  Debug level: Normal										
▼ Bootloader										
Bootloader: Default: Application 💌										
✓ Peripherals configuration										
Heartbeat LED: Use LED 1 -										
▼ 10 configuration										
Command line interface: Full										
Application serial port: Default: SC1 UART (port 1)  Baud rate (bps): Default: 115.2k										
Virtual UART (port 0) RX mode: Default: FIFO (bytes) • RX size: 128 TX mode: Default: FIFO (bytes) • TX size: 128										
SC1 UART (port 1) RX mode: Default: FIFO (bytes) • RX size: 128 TX mode: Default: FIFO (bytes) • TX size: 128										
▼ Board header										
Default (use GPIO editor)										
Port Pin 0 Pin 1 Pin 2 Pin 3 Pin 4 Pin 5 Pin 6 Pin 7										
PA Alt out (PP) 🔹 Input (float) 👻 Alt out (PP) 👻 Output (PP) 💌 Alt out (PP) 👻 Alt out (PP) 👻 Output (PP) 👻 Output (PP) 👻	1									
PB Output (PP) V Alt out (PP) V Input (float) V Input (float) V Alt out (PP) V Analog V Input (PU/PD) V Alt out (PP) V	j 📗									
PC Alt out (PP) V Alt out (PP) V Alt out (PP) V Input (float) V Input (float) V Alt out (PP) V Input (PU/PD) V Output (PP) V										
HAL options										
Auto-generate upon save										





#### ZICM35xSP2-2x

```
EM357 Appbuilder Setup ZICM35xSP2-2x
```

Set up PC5 as Alt Out (PP) and PC6 as Output (PP) as shown below in the Board header section of the HAL configuration tab. More modifications are needed in the board.h file after project is generated.

Ember Desi	ktop				-	A 1.4			
File Edit Fi	ilters Window H	lelp							
6 🖬 🕲	2900	≥ 88 S <sub>10</sub> L <sub>10</sub>	Pio 6 11 4 6	D 227	т 3+ 8, ≣+	// ± ▶ 🕾	🖿 🗦 🐤 🛛 o	+11 🔗 🎀 🐶 🚱	
(2) *New Con	nfiguration 🕄								- D
ZCL Applicat	tion Framework V2								Generate Generate
General	🖧 ZCL cluster con	figuration 🔛 Sta	ck configuration 🐗	HAL configuration	Plugins S Ca	Ilback configuration	n 🛃 Includes		
· Platfor	m configuration								
Platfo	orm: EM357 +	Host Unknown	<ul> <li>Debug level:</li> </ul>	Normal •					
> Bootlos	ader								
> Periphe	erals configuration								
► IO conf	figuration								
- Board I	header								
Defa	ult (use GPIO editor)	)							
Port	Pin 0	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	
PA	Alt out (PP) -	Input (float)	Alt out (PP)	Output (PP)	Alt out (PP) 🔹	Alt out (PP)	Output (PP)	Output (PP) 🔹	
PB	Output (PP) 🔹	Alt out (PP)	<ul> <li>Input (float)</li> </ul>	▼ Input (float) ▼	Alt out (PP) 🔹	Analog	Input (PU/PD)	Alt out (PP)	
PC	Alt out (PP) 🔹	Alt out (PP)	Alt out (PP)	▼ Input (float) ▼	Input (float) 🔹	Alt out (PP)	Output (PP)	Output (PP)	
RAM F	HAL options      RAM Retention All RAM retained								
	erate upon save							1	
1 0° 1	Datagram send error	r.							

Figure 3. ZICM35xSP2-2x Appbuilder Setup

#### xxxxBoard.h File Changes

Edit generated board.h file with the following changes to support SP2-2 and SP2-2C modules. There are 3 areas that need to change: gpioOutPowerUp, gpioCfgPowerDown, and gpioOutPowerDown. See highlights in red.

```
int8u gpioOutPowerUp[3] = {
                                                                                           \
                                                                                           \backslash
                                 ((0
                                                           <<PA0 BIT) |
                                                                                           \setminus
                                  (0
                                                           <<PA1 BIT) |
                                                           <<PA2 BIT) |
                                  (0
                                                                                           \backslash
                                  /* nSSEL is default idle high */
                                                                                           \
                                                           <<PA3 BIT) |
                                  (1
                                                                                           /
                                  (PWRUP OUT PTI EN
                                                          <<PA4 BIT) |
                                                                                           \
                                  (PWRUP OUT PTI DATA <<PA5 BIT) |
                                                                                           \
                                  (PWRUP OUT LED RHO
                                                          <<PA6 BIT)|
                                                                                           \
                                  /* LED default off */
                                                                                           \
                                  (1
                                                          <<PA7 BIT)),
                                 ((1
                                                           <<PB0 BIT) |
                                  (1
                                                           <<PB1 BIT)|
                                                                           /* SC1TXD
                                                                                        */\
                                  (1
                                                           <<PB2 BIT)|
                                                                           /* SC1RXD
                                                                                        */ \setminus
                                                           <<PB3 BIT)|
                                  (1
                                                                           /* SC1nCTS */\
```



(0 <<PB4 BIT)| /\* SC1nRTS \*/\ <<PB5 BIT)| (0 /\* PB6 has button needing a pullup \*/ / (GPIOOUT PULLUP <<PB6 BIT) | (0 <<PB7 BIT)), <<PC0 BIT)| ((1 (0 <<PC1 BIT) | (1 <<PC2 BIT) | (0 <<PC3 BIT) | <<PC4 BIT)| (0 /\* Set PC5=1 at pwr up for SP2-2 or SP2-2C \*/ (1 <<PC5 BIT) | /\* Set PC6=1 at pwr up for SP2-2 or SP2-2C \*/ (1 <<PC6 BIT) | <<PC7 BIT)) (1 } int16u gpioCfgPowerDown[6] = { ((GPIOCFG IN PUD <<PA0 CFG BIT) | <<PA1 CFG\_BIT) | (GPIOCFG IN PUD (GPIOCFG IN PUD <<PA2 CFG BIT) | (GPIOCFG OUT <<PA3 CFG BIT)), ((PWRDN CFG PTI EN <<PA4 CFG BIT) | (PWRDN CFG PTI DATA <<PA5 CFG BIT) | (PWRDN CFG LED RHO <<PA6 CFG BIT) | (GPIOCFG OUT <<PA7 CFG BIT)), ((GPIOCFG OUT <<PB0 CFG BIT) | (GPIOCFG OUT <<PB1 CFG BIT) | /\* SC1TXD (GPIOCFG IN PUD <<PB2 CFG BIT) | /\* SC1RXD \*/\ (GPIOCFG IN PUD <<PB3 CFG BIT)),/\* SC1nCTS \*/\ <<PB4 CFG BIT) | /\* SC1nRTS \*/\ ((GPIOCFG OUT /\* disable analog for sleep \*/ (GPIOCFG IN PUD <<PB5 CFG BIT) | (GPIOCFG IN PUD <<PB6 CFG BIT)| /\* need to use pulldown for sleep \*/ (GPIOCFG IN PUD <<PB7 CFG BIT)), ((GPIOCFG IN PUD <<PC0 CFG\_BIT)| (GPIOCFG OUT <<PC1 CFG BIT) | (GPIOCFG OUT <<PC2 CFG BIT) | (GPIOCFG IN PUD <<PC3 CFG BIT)), ((GPIOCFG IN PUD <<PC4 CFG BIT) | /\* Set PC5 Cfg to OUTPUT for SP2 and SP2-2C \*/ (GPIOCFG OUT <<PC5 CFG BIT) | /\* Set PC6 Cfg to OUTPUT for SP2 and SP2-2C \*/ (GPIOCFG OUT <<PC6 CFG BIT) | (CFG TEMPEN <<PC7 CFG BIT)) int8u gpioOutPowerDown[3] = { / ((GPIOOUT PULLUP <<PA0 BIT) | / (GPIOOUT PULLUP <<PA1 BIT)|





```
(GPIOOUT PULLUP
                       <<PA2 BIT) |
                                                   /
  /* nSSEL is idle high */
                                                   \
  (1
                       <<PA3 BIT) |
  /* enable is idle low */
  (PWRDN OUT PTI EN
                       <<PA4 BIT) |
  /* data is idle high */
  (PWRDN OUT PTI DATA <<PA5 BIT) |
  (PWRDN OUT LED RHO <<PA6 BIT) |
  /* LED off */
  (1
                       <<PA7 BIT)),
 ((0
                       <<PB0 BIT)|
  (GPIOOUT PULLUP
                       <<PB1 BIT) | /* SC1TXD
                                                 */\
  (GPIOOUT PULLUP
                       <<PB2 BIT) | /* SC1RXD
                                                */\
                       <<PB3 BIT) | /* SC1nCTS */\
  (GPIOOUT PULLDOWN
  (GPIOOUT PULLUP
                       <<PB4 BIT)|
                                     /* SC1nRTS */\
  /* tempsense needs pulldown */
  (GPIOOUT_PULLDOWN
                       <<PB5 BIT)|
  /* PB6 has button needing a pullup */
  (GPIOOUT PULLUP
                       <<PB6 BIT)|
  /* buzzer needs pulldown for sleep */
  (GPIOOUT PULLDOWN
                       <<PB7 BIT)),
 ((GPIOOUT PULLUP
                       <<PC0 BIT)|
  (0
                       <<PC1 BIT) |
  (1
                       <<PC2 BIT) |
  (GPIOOUT PULLDOWN
                       <<PC3 BIT) |
  (GPIOOUT PULLDOWN
                       <<PC4 BIT)|
  /* Set PC5 value to 0 for SP2 and SP2-2C */
  (0
                       <<PC5 BIT)|
  /* Set PC6 value to 0 for SP2 and SP2-2C */
  (0
                       <<PC6 BIT) |
  /* Temp Sensor off */
                                                   \setminus
                                                   \
  (0
                       <<PC7 BIT))
}
```



### **CHAPTER 3 – REGULATORY POWER RESTRICTIONS**

This chapter describes how to implement CEL's regulatory restrictions in applications created by AppBuilder for a device that joins a network. FCC restrictions for the ZICM35xSP2 Module are shown as an example. To implement ETSI or FCC restrictions for the ZICM35xSP0 Module, change the power steps according to the *Software Compliance* table in the *ZICM35xSPx-1 Datasheet*. Implementations are described for the following stack versions:

- 1. Ember Stack 4.7 GA and later versions
- 2. Pre-Ember Stack 4.7 GA versions

In addition to being required for regulatory reasons, the implementation of these restrictions is also required for the ZICM35xSP2 because operating at TX steps higher than -2 (i.e., -1 to 8) could potentially damage the module's internal Power Amplifier (PA).

#### IMPLEMENTING FCC POWER RESTRICTIONS FOR EMBER STACK 4.7 GA AND LATER

The Ember Stack 4.7 GA supplies the callback <code>emberAfPluginNetworkFindGetRadioPowerForChannelCallback</code> to set the power level as part of its Network Find Plugin option. Select the plugin and check the *Get radio output power from callback* box to tell the stack to use this callback. If the plugin is used and the *Get radio output power from callback* box is not checked, the radio output power value will be hard-coded with the value in the *Radio Output Power* pull down menu. See the example shown in Figure 3.

New Configuration			
ck: EmberZNet 5.0 GA EM35X, app: ZCL Application Fr	amework	V2 Device name: ZigBee	« Previ
ctory for generated files: 👸 C:\Users\emar\Ember\/	EmberZNe	t5.0.0-GA\EM35x\.	🕞 🕨 Gener
ZCL cluster configuration 🔛 Stack configuration 🥔	HAL confi	iguration 🚸 Plugins 🛛 😴 Callback con	figuration 🔬 Includes
lugins are Ember implementations of callbacks that you	can option	ally include.	
Name	Use' ^	Name: Network Find	
Trust Center Keepalive		Quality: V Production Ready	
Tunneling client cluster		Description:	
Tunneling server cluster			
Standalone Bootloader			finding and joining any viable network via scanning, rather than joining a deploy your application in an environment where there are potentially
Standalone Bootloader Client			o implement a mechanism for network blacklisting. Network blacklisting
Standalone Bootloader Common	T		g the emberAfPluginNetworkFindJoinCallback.
Standalone Bootloader Server	-		
> Trust Center			
Trust Center Network Key Update Broadcast		0.1	
Trust Center Network Key Update Periodic	<b>H</b>	Options:	
Trust Center Network Key Update Unicast			
🕞 Unknown	_		V11 12 13 V14 V15 16 17 18
Window Covering server cluster		Channel mask:	V 19 V 20 21 22 23 V 24 V 25 26
😂 Utility			
Address Table	$\checkmark$	Radio output power:	3 dBm 🔹
Button Form/Join Code		Get radio output power from callba	ck
Concentrator Support	$\checkmark$	Extended PAN ID:	00 00 00 00 00 00 00 00 00
EEPROM		Extended PAINED.	
End Device Support		Scan duration exponent:[0-14]	5
Fragmentation		Joinable scan timeout (minutes):[0-5]	1
Gateway Support			( <sup>7</sup>
General response commands		Plugin information and dependencies:	
- Interpan Plugin		Source files (1)	
Network Find	$\checkmark$	S Implemented callbacks (7)	
Simple Clock		Defined callbacks (3)	
Stack Diagnostics		Required plugins (0)	
Test Harness		<ul> <li>Required server clusters (0)</li> </ul>	
🗁 ZigBee OTA Bootloading		<ul> <li>Required client clusters (0)</li> </ul>	
OTA Cluster Platform Bootloader			
< III			

Figure 4. AppBuilder Plugin Tab - Stack Version 4.7 GA



Fill in the function <code>emberAfPluginNetworkFindGetRadioPowerForChannelCallback()</code> in the generated <code>callback.c</code> file with the appropriate restrictions. The code will return the appropriate power step for the channel specified. The current channel being used is passed into the function call. Sample code is as follows:

```
// set power level for SP2 per FCC restrictions here
if(channel > 10 && channel < 25)
{
    return -2;
}
else if(channel == 25)
{
    return -6;
}
else if(channel == 26)
{
    return -26;
}
return EMBER_AF_PLUGIN_NETWORK_FIND_RADIO_TX_POWER;</pre>
```

#### **IMPLEMENTING REGULATORY POWER RESTRICTIONS FOR PRE-EMBER STACK 4.7 GA**

For Ember Stacks older than Version 4.7, no callback is supplied. The Network Find plugin cannot be used "as is". A new Network Find plugin must be written that incorporates CEL's FCC or ETSI restrictions. To do this, the callbacks that are used by the Network Find plugin must be filled in by the developer in the callbacks.c file. Figure 4 shows an example of enabling the Network Find plugin callbacks into the callbacks.c file in AppBuilder.

App8uilder			- • ×
File Help			
(3 *New Configuration 🕄			° 0
Stack: EmberZNet 4.6.5 GA EM35X with Cont	rolScope e	xtensi	ions, app: ZCL Application Framework V2 Device name: ZigBee
Directory for generated files:	mar\Ember	Embe	vZNet4.6.5.1-GA\em35x\.
			configuration ( Plugins S Callback configuration ) & Includes
s are Ember implementations of callbacks that y			
10	Use?		Name: Network Find
Fragmentation			Quality: V Production Ready
General response commands	H		Description
Generic tunnel cluster	-		
Groups client cluster			Ember implementation of routines for finding and joining any viable network via scanning, rather than joining a specific network.
Groups server cluster			
IAS Ace Server and Zone Client			
IAS WD Server		-18	
IAS Zone Server		- 11	
Identify cluster	~		1 m
Interpan Plugin			Options:
Key Establishment	_		
Level Control server cluster		1	Scan duration exponent[0-14] 5
Messaging client cluster			And the second development of the
Messaging server cluster			Plugin information and dependencies:
Network Find			a 😓 Source files (1)
On/Off server cluster			network-find.c
OTA Cluster Platform Bootloader			A 😴 Implemented callbacks (7)
OTA Bootload Cluster Client		-	emberAfUnusedPanIdFoundCallback
OTA Bootload Cluster Client Policy			emberAljoinableNetworkFoundCallback
OTA Bootload Cluster Common Code			emberAfScanErrorCallback
OTA Bootload Cluster Server			emberAfFindUnusedPanldAndFormCallback
OTA Bootload Cluster Server Policy			emberAfStartSearchForJoinableNetworkCallback
OTA Bootload Cluster Storage Common Coo	de		emberAfContinueJoinableNetworkSearchCallback
OTA POSIX Filesystem Storage Module			emberAfNetworkFindOperationCompleteCallback
OTA Simple Storage Module			Q Defined callbacks (1)
OTA Simple Storage EEPROM Driver			emberAfPluginNetworkFindFinishedCallback
OTA Simple Storage RAM Driver			Required plugins (0)     Required server clusters (0)
Partner Link Key Exchange			Required client clusters (0)
Poll Control server cluster	_	٣	• under on entrie entries (a)
		•	
		-	
Auto-generate upon save			

Figure 5. AppBuilder Plugin Tab - Pre-Stack Version 4.7 GA



Do not check the *Network Find* box. Expanding the 'Implemented callbacks' item details a list of callbacks that need to be filled in. Those callbacks now appear in the Callback Configuration tab in Figure 5.

w Configuration 😂		er ver 70 Andertine fo		ND Designations Talker	
EmberZNet 4.6.5 GA EM35X with ControlScope	extensio	ns, app: ZCL Application Fra	amewor	k VZ Device name: Zigbee	«« Pre
ry for generated files: 🚨 C:\Users\emar\Embe	r\Emberi	ZNet4.6.5.1-GA\em35x\.			🕞 🕒 Ger
L cluster configuration	HAL C	onfiguration 🐟 Plugins 🖇	S Callb	ack configuration	
nabling callbacks, you will have to implement them	in your o	ode otherwise, you will get l	inker en	ors when building your device.	
H 🔀				-	
me	Use	Туре		*	
S Ota Client Incoming Message Raw		Custom callback			
S Ota Client Start		Custom callback			
S Ota Client Version Info		Custom callback			
S Ota Client Custom Verify		Custom callback			
S Ota Client Download Complete		Custom callback			
S Ota Client Bootload		Custom callback			
S Ota Server Query	Ц	Custom callback			
S Ota Server Block Size		Custom callback			
S Ota Server Image Block Request		Custom callback			
S Ota Server Upgrade End Request		Custom callback			
S Ota Page Request Server Policy	Ц.	Custom callback			
S Interpan Send Message		Custom callback			
S Unused Pan Id Found	$\checkmark$	Custom callback			
S Joinable Network Found	$\checkmark$	Custom callback		a	
S Scan Error		Custom callback			
S Find Unused Pan Id And Form	~	Custom callback			
S Start Search For Joinable Network	$\checkmark$	Custom callback		E	
S Continue Joinable Network Search		Custom callback			
S Network Find Operation Complete	$\checkmark$	Custom callback			
S Cluster Security Custom	Ц	Custom callback		-	
S Configure Reporting Command	<u> </u>	Custom callback			
S Read Reporting Configuration Command	<u> </u>	Custom callback			
S Clear Report Table	Ц	Custom callback			
S Reporting Attribute Change	Ц	Custom callback			
S Network Key Update Complete		Custom callback			
Plugin-specific callbacks					
🔥 Handlers defined by stack	_				
S emberPollCompleteHandler	Ц.	Stack handler			
CT 1 11 0 11 111 11 11		C 1 1 1			

Figure 6. Callback Plugin Tab - Pre-Stack Version 4.7 GA

Check the boxes matching the implemented callback. If the device is not a coordinator, the *Unused Pan Id Found* and *Find Unused Pan Id And Form* boxes do not need to be checked and therefore no code is needed for those callbacks. Clicking the Generate button creates the IAR project with the appropriate callback function stubs in the callbacks.c file.

#### FCC Restrictions Software Design

To implement the FCC restrictions, a state machine is needed to keep track of the channels that require a different power step. A separate scan for joinable networks is required for channels requiring a different power step. Example: ZigBee<sup>®</sup> applications use the following channels: 11, 14, 15, 19, 20, 24 and 25.

If using the ZICM35xSP2 Module, the FCC restrictions for the following channels need to have a different max power step (as required):

- Channels 11 24 have a max power step of -2
- Channel 25 has a max power step of -6



If you are implementing the FCC or ETSI restrictions for the ZICM35xSP0 Module, substitute the appropriate values from the *ZICM35xSP2 Datasheet* in the example below.

A state machine is needed to keep track of which channel is being searched. Set the power step accordingly and then initiate a scan for joinable networks. There are two states:

- 1. Channel is in the range of 11 24 (CHANNEL11)
- 2. Channel is at 25 (CHANNEL25).

Initial state is CHANNEL11. Next state is CHANNEL25; then the states alternate between CHANNEL11 and CHANNEL25 during the search for a network to join. Figure 6 shows the simple FCC state machine diagram.

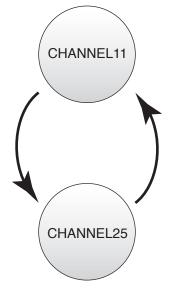


Figure 7. FCC State Machine Diagram

#### Implementing FCC Design into AppBuilder Generated Code

To implement the Network Find operation, copy the code from the Network Find plugin source code into its equivalent callback in the generated callback.c file, then code in the FCC design.

The following callbacks are used to implement the FCC restriction design:

- emberAfJoinableNetworkFoundCallback
- emberAfScanErrorCallback
- emberAfStartSearchForJoinableNetworkCallback



Figure 7 gives an overall flow of how the callbacks are called.

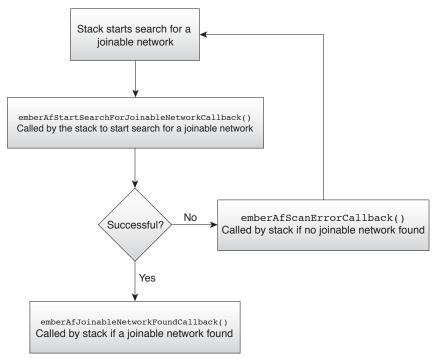


Figure 8. Stack Callback Flowchart

#### ${\tt emberAfStartSearchForJoinableNetworkCallback}$

The callback <code>emberAfStartSearchForJoinableNetworkCallback()</code> is called to start the process of finding a joinable network. Based on the state machine's channel state, set the channel mask for the list of channels to scan and set the power step before the Network Find plugin code calls the API to start the scan. If a joinable network is found, <code>emberAfJoinableNetworkFoundCallback</code> is called by the stack. If no joinable network is found, <code>emberAfScanErrorCallback</code> is called by the stack.

#### Use the following example to merge in:

```
if(scanchannel == 11)
{
    // set tx power level to -2
    emberSetRadioPower(-2);
    // set the mask to scan for 11 - 24 list
    channelMask = 0x0118c800;
}
else
{
    emberSetRadioPower(-6);
    //set channel mast to scan only 25
    channelMask = 0x02000000;
}
Status = emberScanForJoinableNetwork(channelMask, emAfExtendedPanId);
```

#### emberAfScanErrorCallback

The callback <code>emberAfScanErrorCallback()</code> handles the error condition during the search for a joinable network. It is here that code should be added to move the channel state to the next state.



Use the following example:

```
if (status == EMBER_NO_BEACONS) {
   emberAfCorePrintln("%p and join scan done", "Form");
} else {
   emberAfCorePrintln("%p error 0x%x", "Scan", status);
}
emberAfCoreFlush();
// set up for next scan, state machine controlled by scanchannel a global variable
if(scanchannel == 11)
   // set to scan channel 25 as the next state
   scanchannel = 25;
else
   // start from beginning as the next state
   scanchannel = 11;
// set flag to have the emberAfMainTickCallback handler start new search
Startanother_search = 1;
```

#### emberAfMainTickCallback

Check Startanother\_search flag and, if set, call emberAfStartSearchForJoinableNetwork() API to start the process again. The stack will then call emberAfStartSearchForJoinableNetworkCallback().

Use the following example:

```
if(Startanother_search)
{
emberAfStartSearchForJoinableNetwork();
}
```

#### emberAfJoinableNetworkFoundCallback

The emberAfJoinableNetworkFoundCallback is called by the framework when a joinable network is found. Modify the code copied from the Network Find plugin source and set the power level based on the channel before calling the API.

Use the following example:

```
networkParams.radioTxPower = -2; // override default to -2
// FCC restrictions
if(networkFound->channel == 25)
{
    // set power to -6
    networkParams.radioTxPower = -6;
}
networkParams.panId = networkFound->panId;
networkParams.radioChannel = networkFound->channel;
status = emberAfJoinNetwork(&networkParams); // Join this network
```

#### emberAfStackStatusCallback

The emberAfStackStatusCallback is called by the framework when a change occurred in the status of the stack. A change in the channel due to the frequency agility qualifies a status change, so the Ember Stack notifies the application through this callback. The power level may need to be changed due to the FCC restriction. Read in the current channel and set the new power level using the emberSetRadioPower() API. Use the following example for the ZICM35xSP2 Module:

```
if (status == EMBER CHANNEL CHANGED)
{
  channel = emberGetRadioChannel();
  // set power level for SP2 per FCC restrictions here
  if (channel > 10 && channel < 25)
  {
    emberSetRadioPower(-2);
  }
  else if (channel == 25)
  {
    emberSetRadioPower(-6);
  }
  else if (channel == 26)
  {
    emberSetRadioPower(-26);
  }
}
```

#### CHOOSING TO USE CO-LOCATED TRANSMITTERS

There are several options to ensure the CEL EM35x ZigBee module transmitter and a co-located transmitter are not transmitting at the same time which is a requirement for using CEL FCC IDs (See the Agency Certifications section of our datasheets). Here are some options listed most preferred to least:

1) Use the radio-hold-off feature, which suppresses radio activity whenever the radio-hold-off input pin (PA6) is asserted. 2) Use the API called in the ISR of the radio-hold-off pin, (e.g., extern void emRadioHoldOffIsr(boolean active); )

Calling this API with a value of TRUE will prevent the radio from transmitting until it is called again with FALSE.

3) Assign an IRQ to TX\_ACTIVE which will be driven high every time your transmitter is being used and developing an interrupt service handler.

4) Check TX\_ACTIVE pin value anytime you want to transmit on the co-located radio.

Please see 120-3022-000\_EmberZNet\_API\_EM35x.pdf for more information (see "REFERENCES" section).

Note: emRadioHoldOffIsr() is best called within an ATOMIC() block. Options 1 and 2 do different things than 3 and 4. Options 1 and 2 are best for keeping the EM35x radio quiet any time that the other MCU is talking, while 3 and 4 can be used to tell the other MCU that the EM35x device is talking.



# REFERENCES

Reference Documents	Download				
California Eastern Laboratories					
0011-00-07-01-000 ZICM357SP0-1, ZICM357SP2-1 Datasheet	Link				
Silicon Labs					
120-3028-000 Application Framework, Version 2 Developer Guide	Link				
120-3029-000 Application Development Fundamentals					
120-3022-000_EmberZNet_API_EM35x.pdf*					
ZigBee					
IEEE Standard 802.15.4-2003					
ZigBee Home Automation Profile Specification, Version 1.2 Revision 29					
ZigBee Specification, Revision 19					

\*This document is included with the download of the EmberZNet Pro stack, which is available to registered users of Silicon Labs Technical Support

## **REVISION HISTORY**

Previous Versions	Changes to Current Version	Page(s)
0011-00-16-10-000 (Issue A) July 10, 2013	Initial Release	N/A
0011-00-16-10-000 (Issue B) August 21, 2014	Added ZICM35xSP2-2x section (migrating to ZICM35xSP2-2x modules on Appbuilder)	5
0011-00-16-10-000 (Issue C) Februay 3, 2015	Added new section "CHOOSING TO USE CO-LOCATED TRANSMITTERS"	14





#### **Disclaimer**

The information in this document is current as of the published date. The information is subject to change without notice. For actual design-in, refer to the latest publications of CEL Data Sheets or Data Books, etc., for the most up-to-date specifications of CEL products. Not all products and/or types are available in every country. Please check with an CEL sales representative for availability and additional information.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of CEL. CEL assumes no responsibility for any errors that may appear in this document.

CEL does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of CEL products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of CEL or others.

Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. CEL assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

While CEL endeavors to enhance the quality, reliability and safety of CEL products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in CEL products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

#### For More Information

For more information about CEL MeshConnect products and solutions, visit our website at: www.cel.com/MeshConnect.

#### **Technical Assistance**

For Technical Assistance, visit <u>www.cel.com/MeshConnectHelp</u>.

