RemoTI^{TM} Simple Application (SimpleApp)

Keywords

- *RemoTI^{TM}*
- *CC2530*
- *CC2533*
- *Remote Controller Node*
- *Target Node*
- *ZigBee RF4CE*
- *ZRC Profile*
- *SmartRF05*

1 Introduction

The RemoTI development kit contains a remote controller reference design and a target board. The RemoTI software installer includes tools and sample applications for the kit hardware.

This application note complements the RemoTI development kit and software, and provides a simple application (SimpleApp), both remote controller and target node, which runs on the SmartRF05EB+CC2530EM/CC2533EM board. This enables application developers to develop RemoTI applications without the RemoTI development kit. In addition, it provides a simple general starting point for application development.

The goal of SimpleApp is to keep things, as the name indicates, simple and showcase discovery/pairing and message exchange with a button press. SimpleApp is built on top of the RTI interface which implements the ZRC profile. LEDs are used to indicate pairing state and message exchange.

RemoTI is Texas Instrument's implementation of the ZigBee RF4CE network protocol standard. You can find more information about RF4CE at www.zigbee.org/rf4ce and RemoTI at www.ti.com/RemoTI.

Table of Contents

2 Abbreviations

EM	Evaluation module
LED	Light Emitting Diode
MAC	Medium Access Control
RF	Radio Frequency
RF4CE	Radio Frequency for Consumer Electronics
OSAL	Operating System Abstraction Layer
HAL	Hardware Abstraction Layer
ZRC	ZigBee Remote Control Profile
NV	Non-Volatile (memory)

3 Hardware and Software Requirements

Hardware

- 2 x SmartRF05 board (revision 1.8.1)
- 2 x USB cable
- 2 x CC2530EM or CC2533EM

See the CC2530 or CC2533 product page [3] or [4] for technical and ordering information. The CC2530DK or CC2533DK contains the necessary HW for the SimpleApp application.

Software

- Latest RemoTI release (www.ti.com/RemoTI)
- IAR workbench (www.iar.com). The version number is detailed in the resource guide of the RemoTI release.

The SimpleApp application note includes a ZIP file containing the IAR workspace with the controller and target node projects. Unzip the file in the top level directory of your RemoTI installation.

4 Project Organization

The SimpleApp workspace contains both the remote controller node and the target node. Go to the [\Projects\RemoTI\SimpleApp\CC2530EB] folder and open up the SimpleApp.eww with the IAR Workbench; you should see a workspace window similar to Figure 1 below after opening the workspace. The tabs at the bottom of the window are used to activate the remote controller node ("Controller") or the target node ("Target") project respectively.

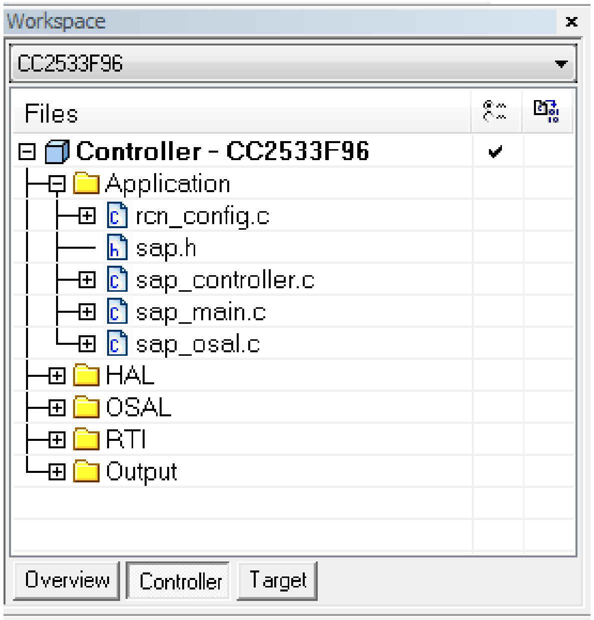

Figure 1 SimpleApp Workspace Window

Note that each project also contains configurations for all CC2530 and CC2533 available RemoTI devices; CC2530F64/F128/F256 and CC2533F64/F96.

4.1 File Descriptions

4.1.1 Common Files

- *sap.h*
  This file declares the SimpleApp external interface that will be referenced in the system setup files sap_main.c and sap_osal.c

- *sap_main.c*
  This file contains the functions for initializing the board, HAL, MAC, OSAL and also registering the callback function for key events.

- *sap_osal.c*
  This file sets up the order of task initialization and event handling.

- *rcn_config.c*
  This file configures the frequency agility parameters (target node) and key seed exchange power level. It is recommended not to change these parameters.

4.1.2 Target node specific project files

- *sap_target.c*
  This file contains the main application code for the target node. It implements the task initialization function (`SAP_Init`), event processing (`SAP_ProcessEvent`) and key callback function (`SAP_KeyCback`) in addition to the RTI interface specific APIs for a target node. It also contains a static function (`sapStartup`) which is used to configure the node capabilities. The node is configured to the default values described in the RTI API guide [1], except from the custom configuration parameters described below.

1. The node is configured as a target node, AC powered, security capable and without channel normalization capability by the following statements:

```
u.Buf[0] = RTI_BUILD_NODE_CAPABILITIES(1, 1, 1, 1);
RTI_WriteItemEx(RTI_PROFILE_RTI, RTI_CP_ITEM_NODE_CAPABILITIES,
1, u.Buf);
```

2. The target node is configured as an TV application device type with the following statements

```
u.Buf[0] = RTI_BUILD_APP_CAPABILITIES(0, 1, 1);
RTI_WriteItemEx(RTI_PROFILE_RTI, RTI_CP_ITEM_APPL_CAPABILITIES,
1, u.Buf);
```

3. The node will store state information (e.g. pairing table, short address, PAN ID) to NV memory and will by default restore this information after a power cycle since `RTI_CP_ITEM_STARTUP_CTRL` is configured to `RESTORE_STATE` in `RTI_InitCnf()`. If ‘Button 1’ (S1) is pressed at power on, the node will clear the state information. This is accomplished with the following statements.

```
  if (HalKeyRead() & SAP_S1_KEY)
  {
    u.Buf[0] = CLEAR_STATE;
    RTI_WriteItemEx(RTI_PROFILE_RTI, RTI_CP_ITEM_STARTUP_CTRL,
    1, u.Buf);
  }
```

4.1.3 Controller project specific files

- *sap_controller.c*
  This file contains the main application code for the remote controller node. It implements the task initialization function (`SAP_Init`), event processing (`SAP_ProcessEvent`) and key callback function (`SAP_KeyCback`) in addition to the RTI interfaces specific APIs for a controller node. It also contains two static functions (`sapStartup` *and* `sapSelectDestIdx`). The `sapStartup` function is used to configure the node capabilities, and the `sapSelectDestIdx` function is used to search the paring table and set the active pairing to the last valid entry (if any) in the pairing table after the stack has initialized. The node is configured to the default values described in the RTI API guide [1], except from the custom configuration parameters described below.

  1. The node is configured as a controller node, battery powered, security capable and channel normalization capable by the following statements

```
    u.Buf[0] = RTI_BUILD_NODE_CAPABILITIES(0, 0, 1, 1);
RTI_WriteItemEx(RTI_PROFILE_RTI, RTI_CP_ITEM_NODE_CAPABILITIES, 1, u.Buf);
```

  2. The node is configured as an remote controller application device type with the following statements. The u.DevList defines the list of devices.

```
u.Buf[0] = RTI_BUILD_APP_CAPABILITIES(0, 1, 1);
RTI_WriteItemEx(RTI_PROFILE_RTI,
RTI_CP_ITEM_APPL_CAPABILITIES, 1, u.Buf);
```

  3. The remote controller node is configured capable of pairing with ZigBee RF4CE TV, Video player/recorder and Set Top Box (STB) application device types with the following statements. The variable TgtTypes entails the devices supported.

```
RTI_WriteItemEx(RTI_PROFILE_RTI,RTI_CP_ITEM_NODE_SUPPORTED_TGT_
TYPES, RTI_MAX_NUM_SUPPORTED_TGT_TYPES, u.TgtTypes);
```

  4. The node will store state information (e.g. pairing table, short address) to NV memory and will by default restore this information after a power cycle since `RTI_CP_ITEM_STARTUP_CTRL` is configured to `RESTORE_STATE.` in `RTI_InitCnf()`. If ‘Button 1’ (S1) is pressed at power on, the node will clear the state information. This is accomplished with the following statements.

```
if (HalKeyRead() & SAP_S1_KEY)
{
  u.Buf[0] = CLEAR_STATE;
  RTI_WriteItemEx(RTI_PROFILE_RTI, RTI_CP_ITEM_STARTUP_CTRL,
  1, u.Buf);
}
```

5 Build and Program the Hardware

Connect a USB cable between your PC and the SmartRF05+CC2530/CC2533 board. If this is the first time you connect a SmartRF05 board to your PC, it will ask you to install the "Chipcon SRF05EB driver. This driver is included in the IAR distribution so you don't want Windows to connect to Windows Update to search for software (select "No, not this time" and hit next in the driver wizard), and let Windows install the software automatically (select "Install software automatically" and hit next in the driver wizard).

1. Start the IAR Workbench and select File->open->workspace.
2. Navigate to the \Projects\RemoTI\SimpleApp\CC2530EB and select the SimpleApp.eww file
3. Click the "Controller" tab in the workspace window, select the project configuration and select Project->Rebuild All
4. Select Project->Debug, ctrl+d or hit the debug symbol to load the executable onto the SmartRF05+CC2530/CC2533EM platform.
5. You can now either hit F5 to run the program in debug mode. If you don't want to run in debug mode, select Debug->stop debugging or hit the stop debugging symbol to disconnect from the target. You need to power cycle the target to execute the program if you elected not to run the program in debug mode.

Repeat step 3, 4 and 5 above for the target receiver node but make sure to click the "target" tab in step 3.

6 Execute the Application

6.1 Buttons and LEDs

The joystick and button 1 (S1) on the SmartRF05 boards is used as input to trigger discovery/pairing and for message exchange. The joystick mapping is shown in Figure 2 with the lines drawn corresponding to the lines around the joystick on the PCB. SW5 is mapped to pressing the joystick down.

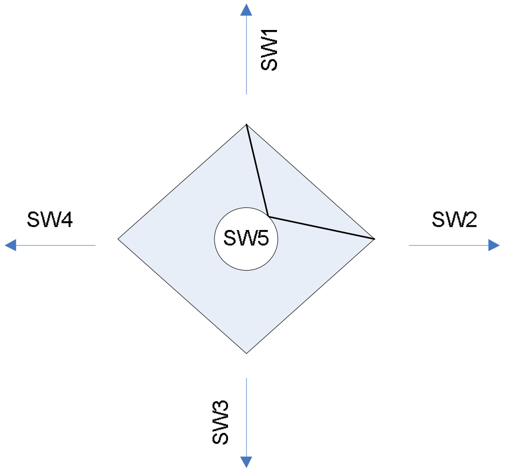

Figure 2. Joystick Switch (SW) Mapping

Table 1 and 2 below shows the key mapping used by SimpleApp. Note that the CC2533 doesn't include the ADC peripheral which is used to read the joystick position. As such, button 1 (S1) is used to send a message.

Node/Switch	SW1	SW2	SW3	SW4	SW5
Remote Controller	Not used	**Send Message**	Not used	**Initiate Pairing**	Not used
Target	Not used	Not used	Not used	**Allow Pairing**	Not used

Table 1. CC2530 Key Mapping

Node/Switch	SW1/2/3/4/5	S1
Remote Controller	**Initiate Pairing**	Send Message
Target	**Allow Pairing**	Not used

Table 2. CC2533 Key Mapping

Three LEDs are used as visual indication of RemoTI initialization, discovery/pairing and message exchange. The 'EM selection' (P19) switch on the SmartRF05 rev.1.7 and 1.8 boards must be put in 'SOC/TRX' position to map the three LEDs correctly.

- Green LED (LED1) is used to indicate message exchange.
  - Controller – The LED is turned on right before the message is sent and turned off when the message is ACKed.
  - Target – The LED is toggled for every message received, independent of which paired remote controller sent the message.

- Red LED (LED2) is used to indicate RemoTI initialization.
  - The LED is turned on when SimpleApp is initializing and turned off when `RTI_InitCnf()` is called indication the node is initialized and ready to process user inputs.

- Yellow LED (LED3) is used to indicate discovery/pairing.
  - Off – no valid pairing in the node paring table
  - On – at least one pairing entry exist in the node paring table
  - Blinking
    - Target node – Indicates the node is in discovery mode and will accept discovery/pairing request from a remote controller node. The target node implements the ZRC auto-discovery pairing mechanism and will as such accept pairing request for 30 sec after key is activated. Note that the pairing request will abort if more than one controller nodes try to pair simultaneously. This is according to ZRC specification.
    - Remote controller node – Indicates the node is sending discovery requests to find a target node for pairing.

NOTE. LED1, LED2 and LED3 are all physically mapped to the Green LED on the SmartRF05 rev1.3 board. As such, the visual indication is slightly impaired on the SmartRF05 rev1.3 boards since the Green LED is used both as RemoTI initialization, discovery/pairing and message exchange indication. See [2] for more details.

6.2 Execution Sequence

1. Apply power to the boards by moving the power-on switch (P8) towards the ‘ON’ position. Make sure you hold ‘Button 1’ (S1) down when applying power to the board if you want to clear the state attributes and the pairing table entries. LED 4 will be on when S1 is pressed. Note that S1 must be pressed until LED2 (Red) is solid on for the clearing of state attributes to take effect.
2. Activate “allow pairing” and “initiate pairing” on the target and controller node respectively to initiate discovery and pairing, and observe the yellow LED blinking.
3. Wait until the yellow LED is solid on for both target and controller to indicate successful pairing. If they are off after blinking, the pairing didn’t succeed. Go back to step 2, or potentially step 1.
4. After successful pairing, activate the “send message” key on the controller node to send a message to the target.
5. Observe the green LED toggle on the target as indication the message was received successfully. The green LED will also be on for the remote controller from the time the message is sent until the ACK is received.

Note that you can also pair a target node with the remote controller included in the RemoTI development kit or pair a controller node with the Target Emulator included in the RemoTI development kit.

7 Limitations

The SimpleApp application is intended to be a very simple starting point for RemoTI developers not using the sample applications included in the RemoTI release. As such, SimpleApp has some limitations:

- Network Processor Interface (NPI) and RTI Surrogate
  The target node does not implement the NPI or the RTI surrogate as the RNP project included in the RemoTI installer. The target node is therefore not capable of connecting to the Target Emulator application running on the PC.

- Controller node LEDs
  This controller node will stay in PM3 until a key is pressed to save power. As such, the LEDs will be turned off. Un-comment the POWER_SAVING preprocessor in the IAR workbench compile options if you don’t want the controller node to enter PM3 sleep mode to see the status of the LEDs.

- ZRC repeat and release messages
  The ZRC profile defines a command repeat and release sequence when a key is held down. This functionality is not implemented in SimpleApp for simplicity.

8 References

[1] RemoTI API, SWRA268, RemoTI installation folder [../Documents/API]

[2] SmartRF05EB User's Guide, www.ti.com/lit/swru210

[3] http://focus.ti.com/docs/prod/folders/print/cc2530.html

[4] http://focus.ti.com/docs/prod/folders/print/cc2533.html

9 General Information

9.1 Document History

Revision	Date	Description/Changes
SWRA286	2009.05.12	Initial release.
SWRA286A	2009.09.30	Updated to comply with RemoTI 1.1
SWRA286B	2010.09.10	Updated to comply with RemoTI 1.2 and misc updates - Changed NV system to snv - Added CLIB for banked project configurations - Added support for CC2533 - Added support for all CC2530 and CC2533 target configurations - Added support for SmartRF05 EB rev. 1.8 - Changed CERC to ZRC
SWRA286C	2012.11.28	Updated to comply with RemoTI 1.3.1 and miscellaneous updates - Changed project configurations - Changed RemoTI API calls for RTI read, and write

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265