Which solution should you use to enable Long Range IoT?

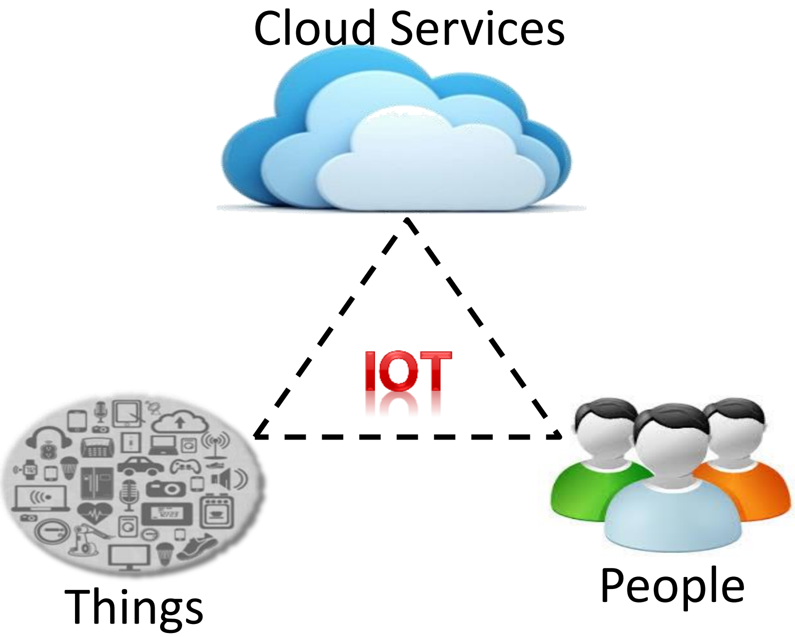
Tony Cave
Low Power Connectivity
Texas Instruments

June 2016

Sub-1GHz Customizable · RF4CE · Bluetooth SMART · 6LoWPAN · ZigBee · M-Bus wireless · SIGFOX One network A billion dreams · 802.15.4g · WiFi

What is the IoT ?

Things, People and Cloud services get connected using the Internet



TI Wireless Connectivity Portfolio

Largest wireless selection

Support for all key technologies and standards for industrial, automotive and consumer

A solution for any application.
Future proof.
Leverage your investment

Lowest power consumption

Use a coin cell or for multi-year, always-on operation or go battery-less with energy harvesting

Ultra-low power by design

Easiest to design with

Quickest learning-curve and development time with full broad market ecosystem

Software, tools, E2E, certified TI modules, TI Designs, SensorTag, online trainings, Cloud

SimpleLink™ Ultra-Low Power Platform

SimpleLink™ 6LoWPAN/ZigBee® CC2630 Wireless MCU
Texas Instruments
SimpleLink™ Bluetooth® Smart CC2640 Wireless MCU
EASY
CR2032
Texas Instruments
SimpleLink™ 2.4 GHz Multi-Standard CC2650 Wireless MCU
EASY
Multi-standard
Texas Instruments
SimpleLink™ Sub-1 GHz CC1310 Wireless MCU
Texas Instruments

CC2630
6LoWPAN/ZigBee

Power a cloud-connected light switch for 10 years with a coin cell battery

CC2640
***Bluetooth*® Smart**

Easy multi-year support for IoT in a tiny package

CC2650
Multi-protocol

Future-proof: Switch between multiple 2.4 GHz technologies with only one design

CC1310
Sub-1 GHz

Combining low-power with high RF performance in a tiny package for long-range connectivity

Ultra Low Power Wireless MCU's

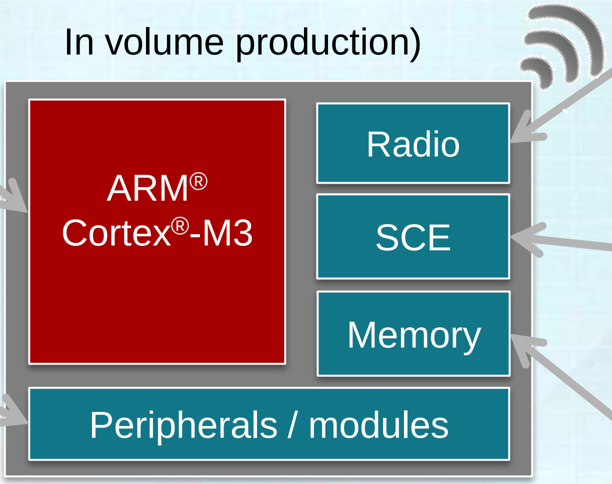
Application MCU

- Application
- Profiles / services
- TI RTOS
- Peripheral drivers and libraries
- Royalty free protocol stacks

Peripherals / modules

- DC/DC converter
- Temp/battery monitor
- AES
- GPIO
- Timers
- UART / SPI
- I2C / I2S
- DMA

In volume production)



QFN package options:
4x4mm, 5x5mm, 7x7mm

Radio

- Extremely flexible ultra-low power radio
- Built in autonomous packet engine
- IDE: SmartRF Studio

Sensor Controller Engine

- Ultra-low power sensor interface
- ADC readings
- Sensor readings via I2C/SPI
- Simple processing
- IDE: Sensor Controller Studio

Memory

- 128 KB Flash
- 8 KB cache
- 20 KB SRAM

Bluetooth SMART

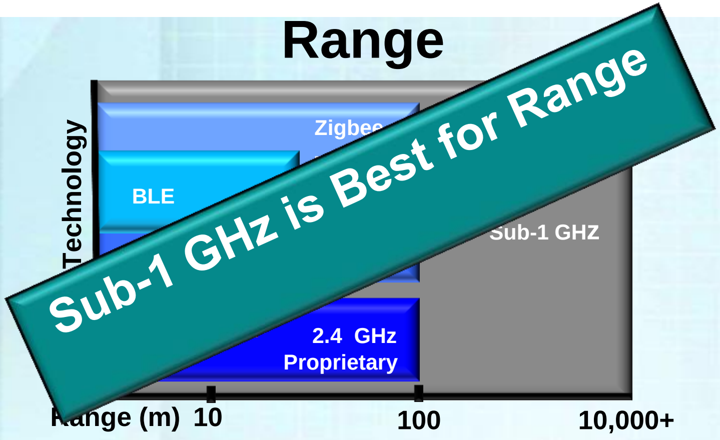
6LoWPAN

ZigBee

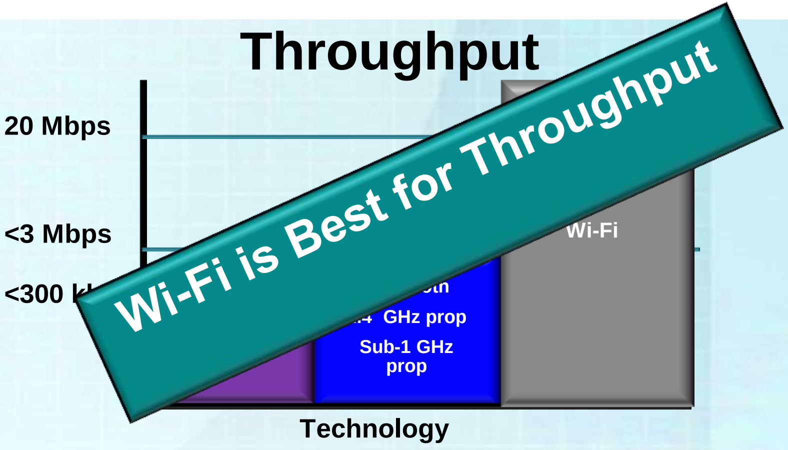
Sub-1GHz

RF4CE

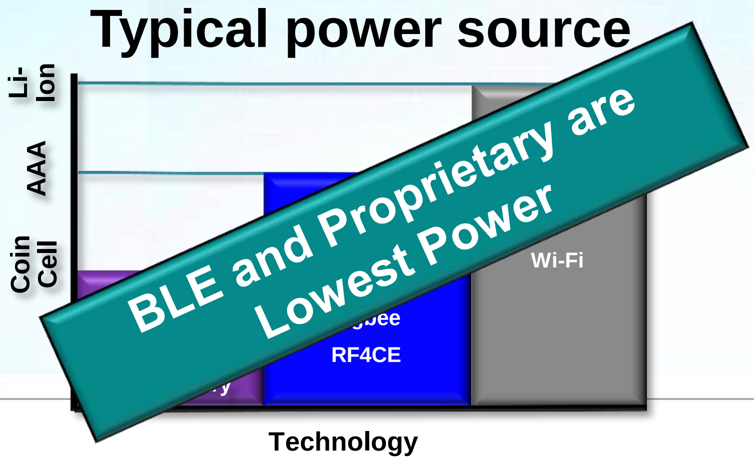
Choosing the Right Technology


Range

Sub-1 GHz is Best for Range


Throughput

Wi-Fi is Best for Throughput


Typical power source

BLE and Proprietary are Lowest Power

CC1310 Why was this part developed?

Improving the three key challenges for a *Sub-1 GHz Wireless MCU*

Lowest Power

- 5.5 mA Radio RX current
- 22.6 mA @ +14 dBm , 12.9 mA @ +10 dBm, Radio TX current
- 51 µA / MHz ARM Cortex M3
- 0.6 µA sleep current w/RTC + full memory retention

Up to 20 year battery life for sensor nodes and flow meters

Long Range

- High sensitivity
  - -110 dBm @ 50 kbps
  - -124 dBm @ 0.625 kbps
- Strong co-existence
  - Up to 90 dB blocking
- +14 dBm output power

Full building to city-wide RF coverage

Most Integrated

- Sensor Controller Engine (SCE)
- 4x4 QFN
- Integrated DCDC
- On-Chip Flash
- TI-RTOS + RF Driver

Complete 315 / 433 / 490 / 779 / 868 / 915 / 920 MHz wireless MCU in a finger-tip size

Texas Instruments

Sensor Controller Engine *(SCE)*

Ultra low power controller to offload the M3

Key features

- Handles sensor polling and performs simple processing
- Operates while the rest of the system is powered down

Examples of sensors that will greatly benefit from using the Sensor Controller:

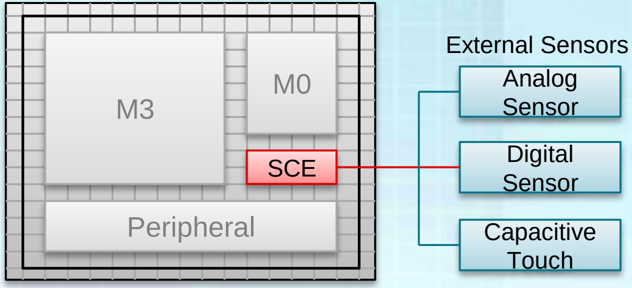
- PIR (motion detector)
- Capacitive touch keys
- Proximity sensors
- Accelerometers
- ADC measurements
- Pulse counting

- Use Sensor Controller Studio for configuration

Data Sheet – Key Features

- 2 KB SRAM (code + data)
- 8.2 uA/MHz

Example: 1 Hz ADC sampling: 0.85 uA

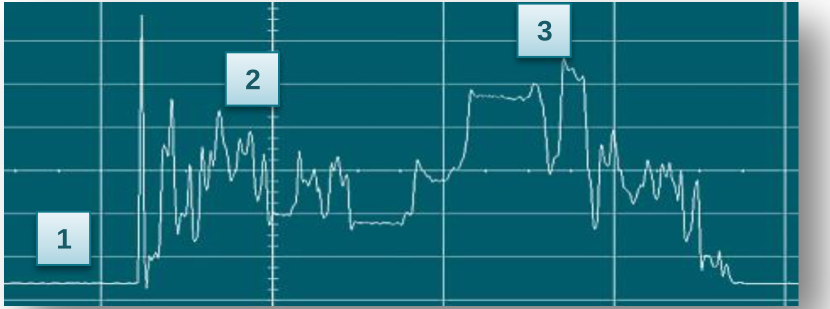


Sensor Controller Studio 1.0.0.34007 - Texas Instruments
File View Help
Start Page
Code Generator
Start Page
Sensor Controller Studio
Texas Instruments
Projects
Create New Project
Open Existing Project
Light Sensor Demo
Clear history...
Tool Documentation
Quick Start Guide
Revision History
Sensor Controller Studio Help
Sensor Controller Task Code Language Reference
Sensor Controller Interface Driver Sample Documentation
Sensor Controller Studio Software License Agreement
Sensor Controller Studio Software Manifest
Examples
SmartRF06EB
Analog Light Sensor
LED Blinker
UART Emulator
SensorTag
Capacitive Touch Data Logger
I2C Light Sensor
General
ADC Data Logger
ADC Window Monitor
Restore original example files...
Online Documentation and Resources
Software
Sensor Controller Studio Product Web Page

Power Consumption

Low average power consumption

1. When in Standby (with RTC and RAM retention)
2. When processing with MCU
3. When radio is in Receive or Transmit
4. When peripheral is polled for data

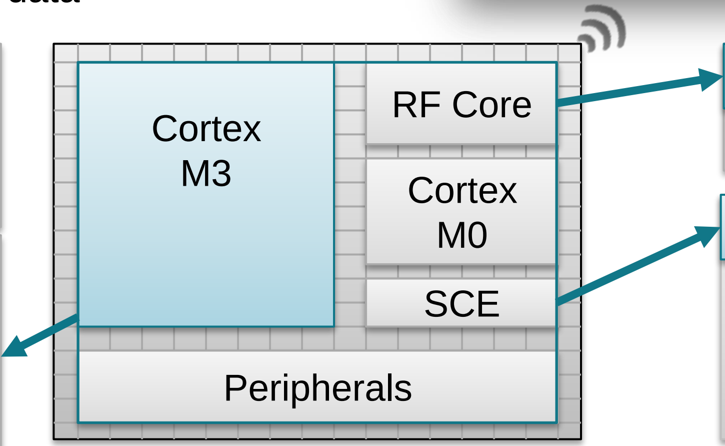


1 Standby Current

- 0.6 µA Standby with RTC and full retention
- Less than 0.2 µA in Shutdown

2 ARM Cortex M3

- Fast processing using 2.5 mA @ 48MHz
- Less time used for stack and application processing and BLE connection events


Cortex M3

RF Core

Cortex M0

SCE

Peripherals

3 Radio

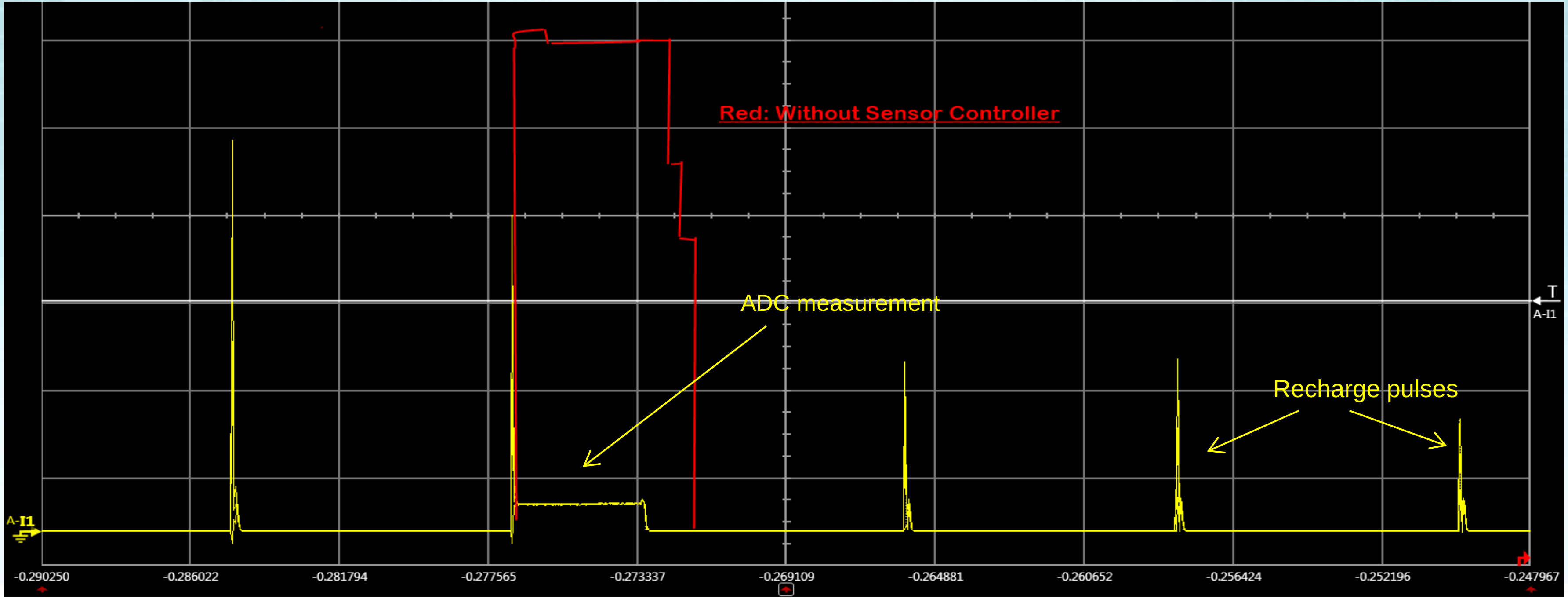
- ~5.5 mA RX / ~13 mA TX current@ 10 dBm

4 Sensor Controller Engine

- Sensor controller and its peripherals can be powered while rest of system is power off.
- Run Sensor Reading with < 5 µA current consumption

ADC Reading: SCE vs CM3

- Current profile, ADC sampling 4 samples / second
  - Sensor Controller vs Waking up the full system



CC1310/50, Software & Tools Overview

- **CC13xx TIRTOS SimpleLink SDK**
  - Starting point examples showing how to use radio, pin, LCD, UART
  - RF examples for peer-to-peer type of applications, great tools integr.
  - Provided in the TI-RTOS SimpleLink bundle, http://www.ti.com/tool/TI-RTOS
- **TI-MAC 2.0 SDK**
  - Sub-1 GHz (IEEE 802.15.4g, ETSI & FCC, including FHSS)
    - Star network applications
- **Contiki OS**
  - Open Source IP/6LoWPAN IEEE 802.15.4g mesh stack
  - CC1310, CC2630, CC2650, and CC2538 supported, CC1350 (1Q16)
  - Available at Github: https://github.com/contiki-os/contiki
- **Software Tools**
  - SmartRF Studio
  - SmartRF Flash Programmer 2.0
  - Sensor Controller Studio
  - CCS and IAR Embedded Workbench

CC1310/50, Software & Tools Overview

- **CC13xx TIRTOS SimpleLink SDK**
  - Starting point examples showing how to use radio, pin, LCD, UART
  - RF examples for peer-to-peer type of applications, great tools integr.
  - Provided in the TI-RTOS SimpleLink bundle, http://www.ti.com/tool/TI-RTOS
- **TI-MAC 2.0 SDK**
  - Sub-1 GHz (IEEE 802.15.4g, ETSI & FCC, including FHSS)
    - Star network applications
- **Contiki OS**
  - Open Source IP/6LoWPAN IEEE 802.15.4g mesh stack
  - CC1310, CC2630, CC2650, and CC2538 supported, CC1350 (1Q16)
  - Available at Github: https://github.com/contiki-os/contiki
- **Software Tools**
  - SmartRF Studio
  - SmartRF Flash Programmer 2.0
  - Sensor Controller Studio
  - CCS and IAR Embedded Workbench

This Training Session will focus on the CC13xx SDK's developed by TI

What is CC13xx TIRTOS SimpleLink SDK?

Proprietary RF Example Tool Box

Flexible

- Supports a large range of Phy and modulation settings
  - OOK
  - GFSK
  - High Speed Mode
  - Long Range Mode
  - Narrow Band
- Generate custom Phy settings from SmartRF Studio dirrectly into the examples

Extreme flexibility

Easy

- Get going quickly with feature rich Out of Box Example Applications
- Fully integrated in to TIRTOS
- Tool Box of examples for building a propreitory RF protocol
- AT Network Processor Example for ultra easy integration with Host MCU/MPU

Accelerate your time to market

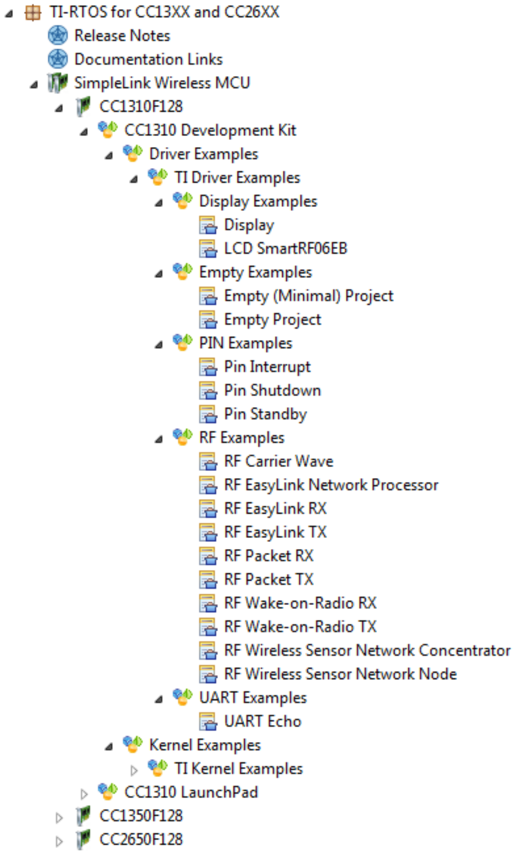
Ultra-low power

- Best-in-class CC1310 platform
  - <6 mA peak current
  - 0.6 uA sleep current (RTC)
  - Autonomous sensor controller
- Integrate Power Manager Driver

Very long battery life or energy harvesting

TI-RTOS Getting Started Examples

- Bundled in TI-RTOS Simplelink
  - IAR & CCS support
  - CCS Cloud (dev.ti.com)
- Platforms supported: SRF06EB/CC1310EM, CC1310 LaunchPad
- RF examples in TI-RTOS SimpleLink bundle:
  - Packet RX
  - Packet TX
  - Carrier Wave (mod, unmod)
  - Wake On Radio
  - ETSI compliant Listen Before Talk
  - PER test supporting IEEE802.15.g FSK, LRM, OOK, HS
  - EasyLink RX
  - EasyLink TX
  - EasyLink Network Processor
  - Wireless Sensor Network, based on EasyLink,
    - Node + Concentrator



- TI-RTOS for CC13XX and CC26XX
  - Release Notes
  - Documentation Links
  - SimpleLink Wireless MCU
    - CC1310F128
      - CC1310 Development Kit
        - Driver Examples
          - TI Driver Examples
            - Display Examples
              - Display
              - LCD SmartRF06EB
            - Empty Examples
              - Empty (Minimal) Project
              - Empty Project
            - PIN Examples
              - Pin Interrupt
              - Pin Shutdown
              - Pin Standby
            - RF Examples
              - RF Carrier Wave
              - RF EasyLink Network Processor
              - RF EasyLink RX
              - RF EasyLink TX
              - RF Packet RX
              - RF Packet TX
              - RF Wake-on-Radio RX
              - RF Wake-on-Radio TX
              - RF Wireless Sensor Network Concentrator
              - RF Wireless Sensor Network Node
            - UART Examples
              - UART Echo
        - Kernel Examples
          - TI Kernel Examples
      - CC1310 LaunchPad
    - CC1350F128
    - CC2650F128

What is TIMAC-2.0.0?

IEEE 802.15.4e/g standard based star networking solution

Robust

- Provides reliable communication: Built-in acknowledgments and retries
- Supports sub-1GHz band: Avoid crowded 2.4GHz spectrum + Long Range
- Provides resistance against interferences: Supports Frequency Hopping configuration
- Provides secure operation: supports AES encryption

Uncompromised robustness

Easy

- Get going quickly with feature rich Out of Box Example Applications
- End to end solution Out of box
  - Local Gateway
  - End nodes
- Compliant with regional regulations
- Scalable network size

Accelerate your time to market

Ultra-low power

- Best-in-class CC1310 platform
  - <6 mA peak current
  - 0.6 uA sleep current (RTC)
  - Autonomous sensor controller
- Low-overhead protocol
- Supports Sleepy devices

Very long battery life or energy harvesting

Choosing the right SDK

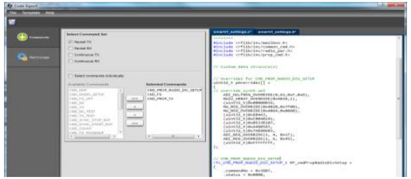
- The Following customers should find the TI RTOS SimpleLink SDK a better match to there needs:
  - Customers that already have a good knowledge or RF and their own proprietary protocols
  - Customers wanting low level access to the RF commands
  - Customer that want to support Phy configurations such as GFSK, On Off keying, long range and High Speed mode
  - Customers that need to control over Phy settings such as BW and Bit Rate.
- The following customers should find TI MAC 2 SDK a better match to there needs :
  - Customers with little knowledge of RF, who need a networking solution
  - Customers who needs to deploy networks scalable in size
  - Customers wanting to support the 802.15.4g standard.
  - Customers that want to bridge IP to RF using a Linux based Gateway.

Backup Slides

CC13xx TIRTOS Simple Link SDK

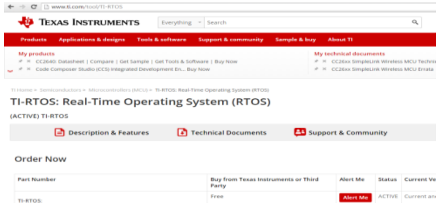
TI-RTOS – Proprietary tools example

SmartRF™ Studio 7



Radio Settings

TI-RTOS



www.ti.com/tool/TI-RTOS

Texas Instruments

My products

- CC2640: Datasheet | Compare | Get Sample | Get Tools & Software | Buy Now
- Code Composer Studio (CCS) Integrated Development En... Buy Now

My technical documents

- CC26xx SimpleLink Wireless MCU Techni
- CC26xx SimpleLink Wireless MCU Errata

TI Home » Semiconductors » Microcontrollers (MCU) » TI-RTOS: Real-Time Operating System (RTOS)

TI-RTOS: Real-Time Operating System (RTOS)

(ACTIVE) TI-RTOS

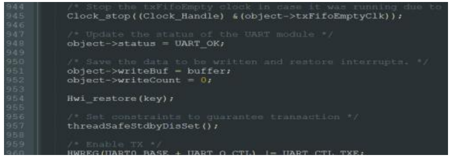
Order Now

Part Number	Buy from Texas Instruments or Third Party	Alert Me	Status	Current Ve
TI-RTOS:	Free	Alert Me	ACTIVE	Current an

RTOS

Drivers

Application Code



```
    /* Stop the txFifoEmpty clock in case it was running due to
    Clock_stop((Clock_Handle) &(object->txFifoEmptyClk));

    /* Update the status of the UART module */
    object->status = UART_OK;

    /* Save the data to be written and restore interrupts. */
    object->writeBuf = buffer;
    object->writeCount = 0;

    Hwi_restore(key);

    /* Set constraints to guarantee transaction */
    threadSafeStdbyDisSet();

    /* Enable TX */
    HWREG(UART0_BASE + UART_O_CTL) |= UART_CTL_TXE;
```

Driver source Code

IDE & Compiler

CCS

IAR SYSTEMS

Sensor Controller Studio

Start Page

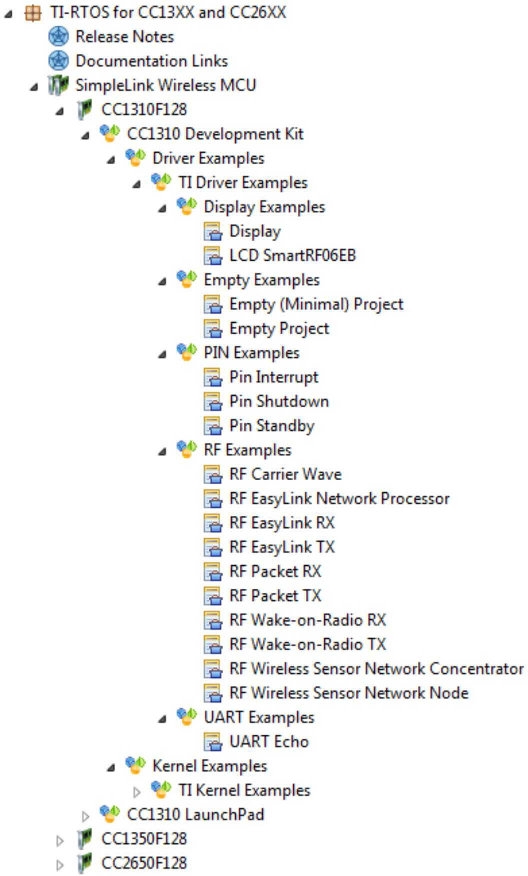
Sensor Controller Studio

Texas Instruments

CC1310 LaunchPad

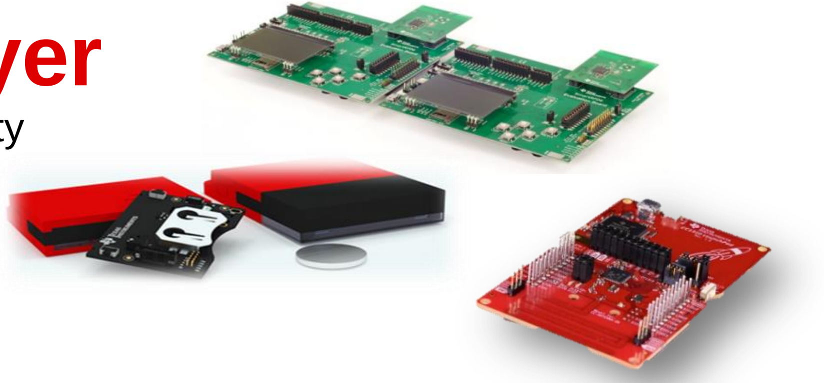
TI-RTOS Getting Started Examples

- Bundled in TI-RTOS Simplelink
  - IAR & CCS support
- Platforms supported: SRF06EB/CC1310EM, CC1310 LaunchPad
- RF examples in TI-RTOS CC13XX and CC26XX bundle:
  - Packet RX
  - Packet TX
  - Carrier Wave (mod, unmod)
  - EasyLink RX
  - EasyLink TX
  - EasyLink Network Processor
  - Wireless Sensor Network, based on EasyLink,
    - Node + Concentrator
  - Wake On Radio
- New in TI-RTOS 2.20.x
  - ETSI compliant LBT
  - PER test supporting IEEE802.15.g FSK, LRM, OOK, HS
  - CC1350 examples (BLE beacons)



- TI-RTOS for CC13XX and CC26XX
  - Release Notes
  - Documentation Links
  - SimpleLink Wireless MCU
    - CC1310F128
      - CC1310 Development Kit
        - Driver Examples
          - TI Driver Examples
            - Display Examples
              - Display
              - LCD SmartRF06EB
            - Empty Examples
              - Empty (Minimal) Project
              - Empty Project
            - PIN Examples
              - Pin Interrupt
              - Pin Shutdown
              - Pin Standby
            - RF Examples
              - RF Carrier Wave
              - RF EasyLink Network Processor
              - RF EasyLink RX
              - RF EasyLink TX
              - RF Packet RX
              - RF Packet TX
              - RF Wake-on-Radio RX
              - RF Wake-on-Radio TX
              - RF Wireless Sensor Network Concentrator
              - RF Wireless Sensor Network Node
            - UART Examples
              - UART Echo
        - Kernel Examples
          - TI Kernel Examples
      - CC1310 LaunchPad
    - CC1350F128
    - CC2650F128

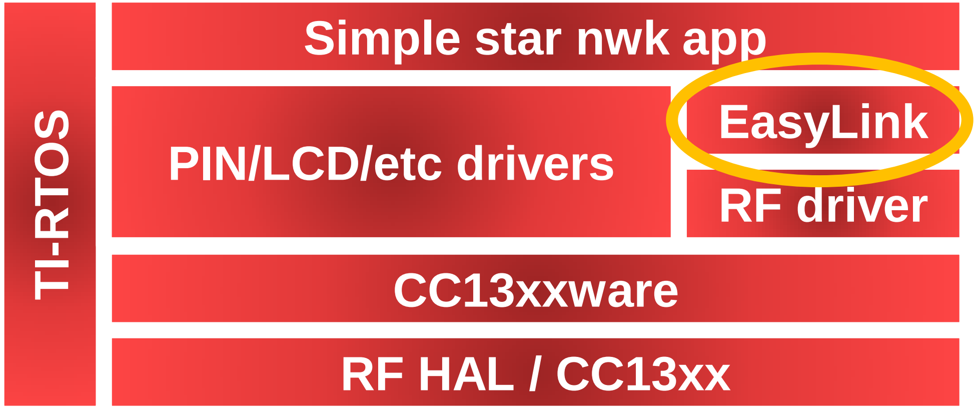
EasyLink Abstraction Layer



- Simple to use abstraction layer, abstract RF complexity

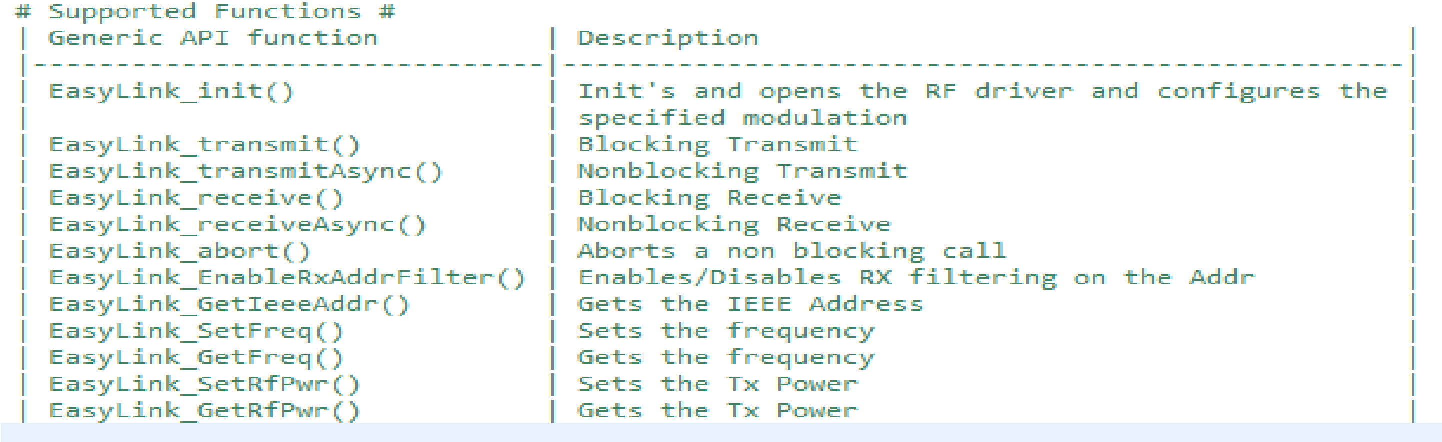
```
EasyLink_init()
EasyLink_transmit(), EasyLink_receive()
```

- Distributed in TI-RTOS Simplelink bundle
- Support for different PHY settings
  - IEEE802.15.4g (GFSK 50kbps), Long Range Mode (LRM) 625bps
  - Custom settings exported from SmartRF Studio
- Muli purpose: 1) abs layer example/start 2) building block
- Platforms:
  - **srf06/cc13xx**,
  - cc1310/50 launchpad,
  - cc1350 sensortag
- CCS cloud & TI-RTOS based



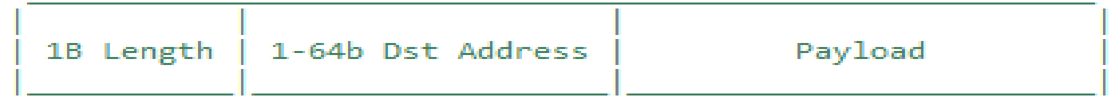
EasyLink API and Packet format

- EasyLink API:



```
# Supported Functions #
| Generic API function            | Description                                       |
|---------------------------------|---------------------------------------------------|
| EasyLink_init()                 | Init's and opens the RF driver and configures the |
|                                 | specified modulation                              |
| EasyLink_transmit()             | Blocking Transmit                                 |
| EasyLink_transmitAsync()        | Nonblocking Transmit                              |
| EasyLink_receive()              | Blocking Receive                                  |
| EasyLink_receiveAsync()         | Nonblocking Receive                               |
| EasyLink_abort()                | Aborts a non blocking call                        |
| EasyLink_EnableRxAddrFilter()   | Enables/Disables RX filtering on the Addr         |
| EasyLink_GetIeeeAddr()          | Gets the IEEE Address                             |
| EasyLink_SetFreq()              | Sets the frequency                                |
| EasyLink_GetFreq()              | Gets the frequency                                |
| EasyLink_SetRfPwr()             | Sets the Tx Power                                 |
| EasyLink_GetRfPwr()             | Gets the Tx Power                                 |
```

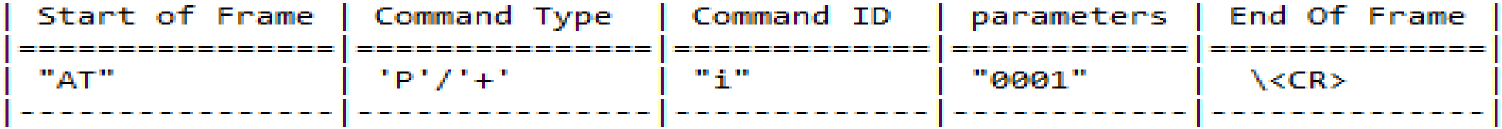
- Frame format
  - The EasyLink implements a basic header for transmitting and receiving data. This header supports addressing for a star or point-to-point network.



```
 ____________________________________________________________________________
|           |                         |                                      |
| 1B Length | 1-64b Dst Address       |              Payload                 |
|___________|_________________________|______________________________________|
```

EasyLink Network Processor

- An EasyLink AT Command Network Processor example
- The EasyLink API has been exposed over an AT Command UART Interface such that it can be exercised by Host SW (running on an PC, MPU or MCU) or by a human using a serial terminal emulator.
- The AT Command Interface uses ASCII characters so that a terminal emulator can send the commands, but also uses framing so that SW can format and parse the AT commands.
- AT Frame format



```
| Start of Frame  | Command Type    | Command ID    | parameters   | End Of Frame    |
|=================|=================|===============|==============|=================|
| "AT"            | 'P'/'+'         | "i"           | "0001"       |  \<CR>          |
|-----------------|-----------------|---------------|--------------|-----------------|
```

- An example, to initialize the radio for IEEE802.15.4g 50kbps, the command is:

  `AT+I 0000<CR>`

- The EasyLink AT Command Interface uses 2 command types:
  - Pxx: Parameters
  - +x: Control Commands

TIMAC-2.0.0

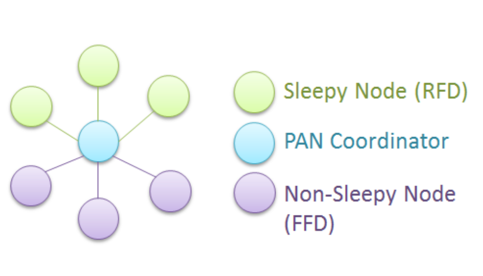
TIMAC-2.0.0 Example Applications

Collector Application

- Creates a TIMAC Low Power Network by starting the device as a PAN-Coordinator.
- Allows new devices to join the network.
- Configures the joining devices for how often to report the sensor data. For sleepy devices it also configures the sleepy node on how often to poll for data.
- Tracks if the devices are active/inactive in the network. It achieves this by sending the tracking request message in a round robin fashion to the connected devices. A device which responds to the command is marked as active while that does not is marked as inactive.

Sensor Application

- Joins a TIMAC Network
- Reports sensor data at configured interval
- If sleepy, polls for any buffered data from the PAN-Coordinator at configured interval
- Responds to the tracking request messages
- Can be battery powered

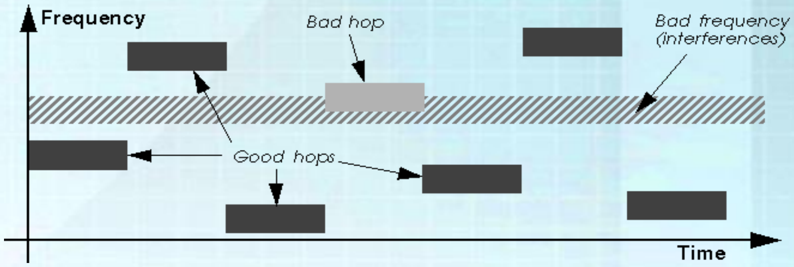


TIMAC-2.0.0

Uncompromised robustness

Designed for range & resilience

Feature	Benefits
802.15.4g sub-1GHz PHY	Longer range Better penetration Less crowded spectrum IEEE-defined Field-proven
Frequency Hopping Configuration	Protection against interferences Longer range in FCC Band
CSMA/CA	Effective medium management
Acks & Retries	Automatic MAC error recovery
Strong link budget	Long range
MAC beacon mode configuration	Reduces collisions Enables very large networks



TIMAC-2.0.0 Network Configurations

Non – Hopping Configurations

Beacon Enabled Mode

- PAN Coordinator sends out periodic messages to which network
- Suitable for L Depl
- , Industrial Sensor any more

Large Networks

Synchronous Network Operation

Non Beacon Mode

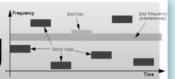
- Network devices request parent bea procedures such as associa
- Network device (using
- eds) of node s
- et Applications: Home Automation, Industrial Automation, and many more

Lowest Power Nodes

Asynchronous Network Operation

Hopping Configuration

Frequency Hopping



- Compliant to frequency hopping schem defined by WiSUN FAN v83 s
- Devices hop on di
- Long t power in
- ons: Home Automation, al Automation, and many more

Longest Range

Asynchronous Network Operation

Q & A