# Low Power RF Software System Design

**How to write low-power RF software**

# Abstract

## How to write low-power RF software


**This presentation explores how to design a RF protocol from scratch with the main focus being on small point-to-point or point-to-multipoint networks.**

**If discusses the different aspects of the protocol design with defining the system details, network topology, network access, device pairing, power budget and saving, and interference. A brief overview of development tools and other available resources are mentioned as well.**

**Some of the TI LPRF specific low-power features are explored like Wake-On-Radio, fast startup from sleep and low power modes.**

# Content

Software design considerations for a low power RF system

- Network Topology
- Network Access
- Device Pairing
- Power Budget & Power Saving
- Interference
- Development Tools
- Debug & Test
- Resources

# Basic Protocol Design

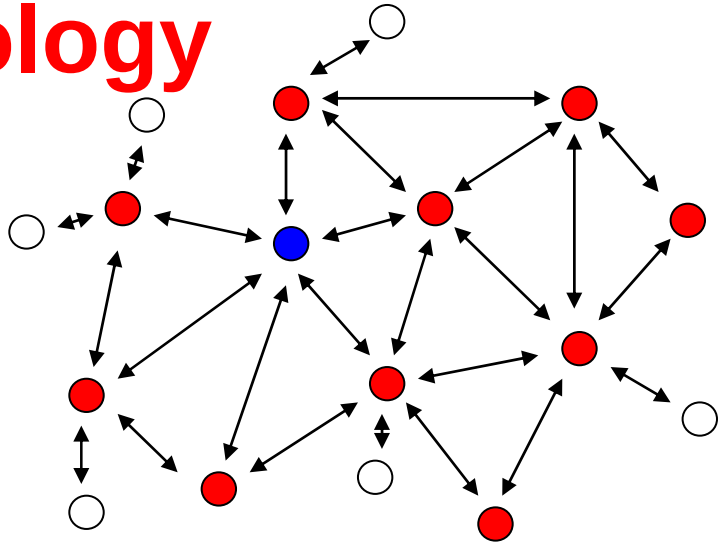
When designing the protocol it is important to know all the details of the system.

- How many nodes in the system?
- How much data will be transmitted and at what rate?
- Which nodes are transmitting and which are replying?
- What kind of latency can be accepted?
- If nodes are battery powered, what kind of battery lifetime is accepted?
- What is the required range?
- What kind of “Quality of Service” is required
- Is precise timing required?
- Does the system have strict memory requirements?
- Is encryption required?

# Network Topology

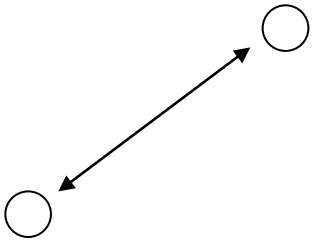
# Network Topology

- Point to point
- Point to multi-point (Star)
- Mesh

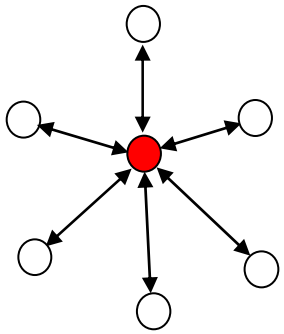


**Mesh network**

For networks consisting of more than 5 nodes, one should consider going for a standard or available proprietary solution.



**Point-To-Point**



**Star Network**

# Network Topology

## Low Power RF protocol overview

| Layer \ Solution | Proprietary | SimpliciTI | TI-MAC | RemoTI | ZigBee |
|---|---|---|---|---|---|
| Application | Design freedom | Design freedom | Design freedom | Design freedom* | Design freedom* |
| Higher Layer Protocol | Design freedom | Design freedom | Design freedom | RF4CE | ZigBee |
| Lower Layer Protocol | Design freedom | SimpliciTI | IEEE 802.15.4 | IEEE 802.15.4 | IEEE 802.15.4 |
| Physical Layer | TI HW | TI HW | IEEE 802.15.4 | IEEE 802.15.4 | IEEE 802.15.4 |
| Frequency | Sub 1 GHz<br>2.4 GHz | Sub 1 GHz<br>2.4 GHz | 2.4 GHz | 2.4 GHz | 2.4 GHz |

Increasing interoperability

* RF4CE and ZigBee (HA, CBA, SE) define application user profiles

# Network Access

# Network access

Techniques to organize the channel access

- Listen before talk (LBT)
- Time division multiple access (TDMA)
- Code division multiple access (CDMA)
- Frequency division multiple access (FDMA)

# Network access - TDMA

| | | | | | |
|---|---|---|---|---|---|
| Time slot 0 (Beacon) | Time slot 1 (Keyboard) | Time slot 2 (Mouse) | Time slot 3 (Remote control) | Time slot 4 (Frequency scan) | Time slot 0 (Beacon) |

Frame

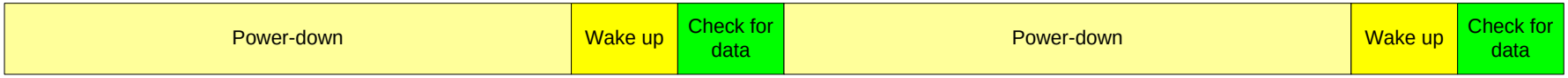
## Advantages:

- Timeslot can be allocated to each slave and thus guarantee data throughput
- No collisions between devices in the system – the channel can be fully utilized
- Provides infinite attenuation between the various devices

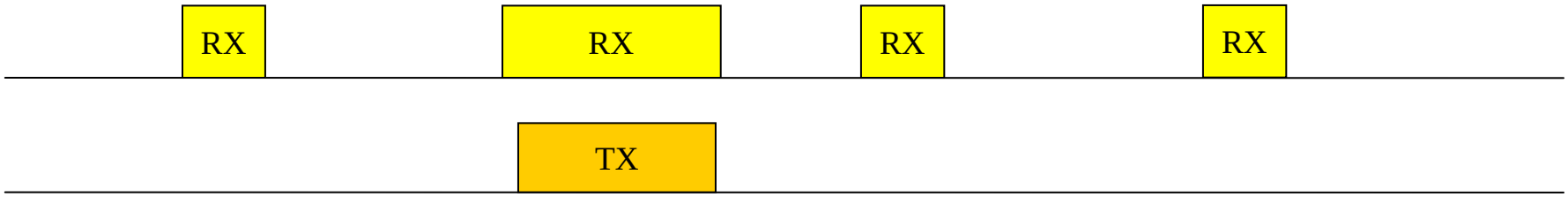
## Disadvantages:

- Need accurate clock on devices or frequent time alignement on start-of-frame

# Network access – Polling receiver




- The receiver typically sleeps and periodically wakes up to listen for a incoming packet.




- The system needs synchronization, so that the polling receiver can wake up at the correct time to receive the transmission.

# Device Pairing

# Device pairing mechanism

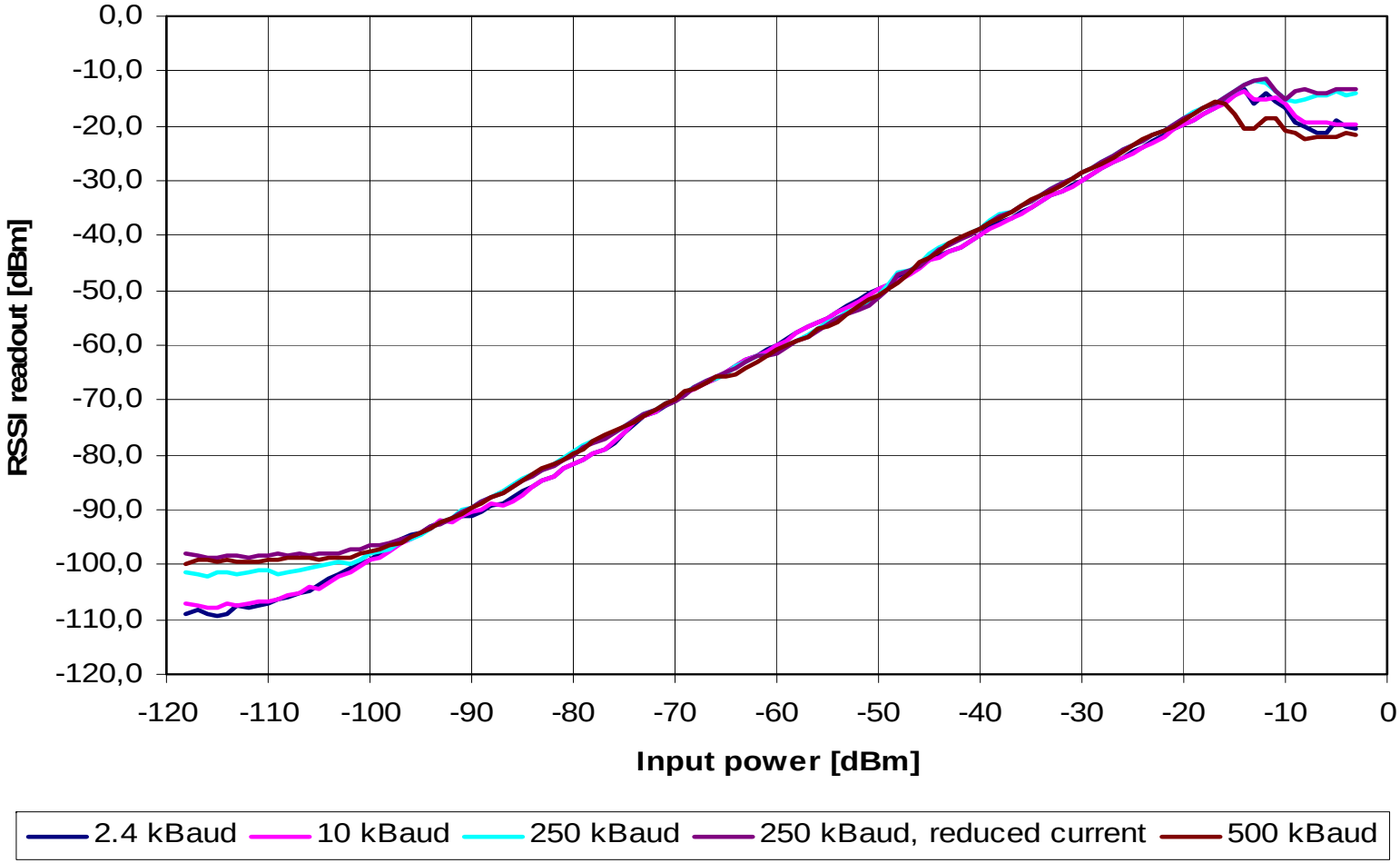
Pairing is needed if there are more networks in range and not all messaging are broadcasts.

How to implement a pairing mechanism?

- Proximity pairing
- Fixed pairing
- User intervention
- Commisioning

For more robust pairing, the above can be combined.

# Device pairing mechanism - proximity




## Typical RSSI Value vs. Input Power Level

# Power Budget & Power Saving

# Power Budget

The system's available power might influence the choices one have when designing the radio protocol.

- Mains powered vs. Battery powered
- Output power and system peak power
- Average current consumption
- Radio Duty-Cycle

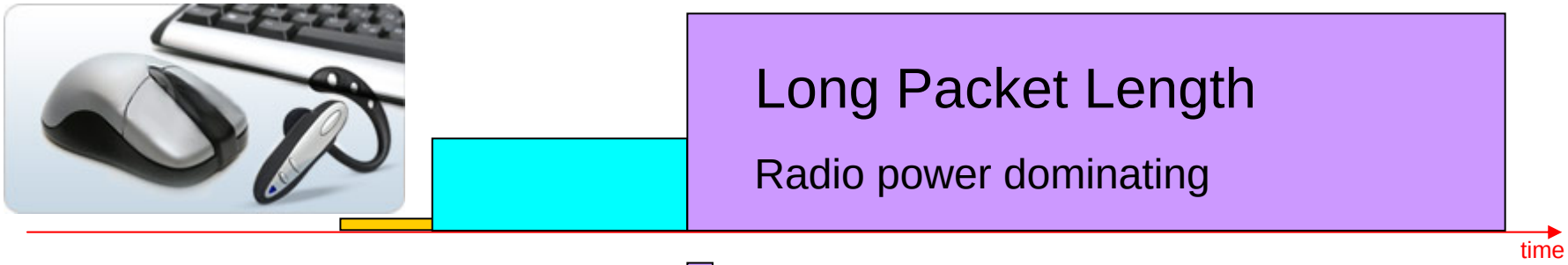
# Power saving

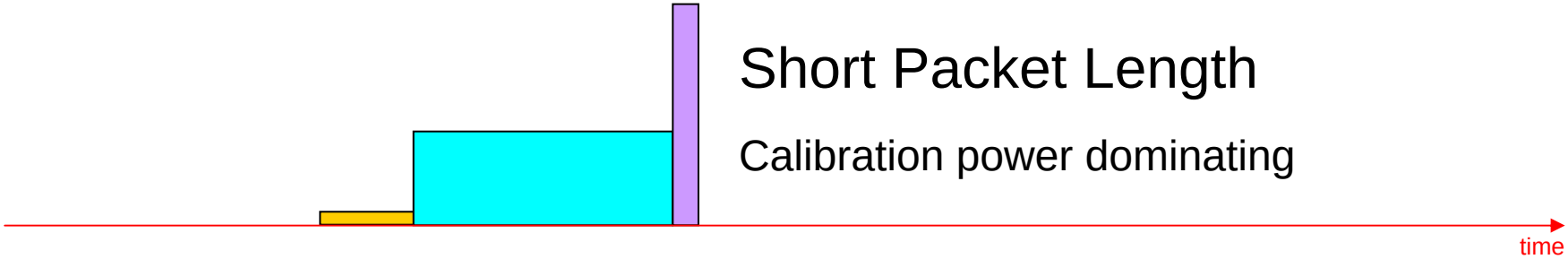
LPRF power saving features:

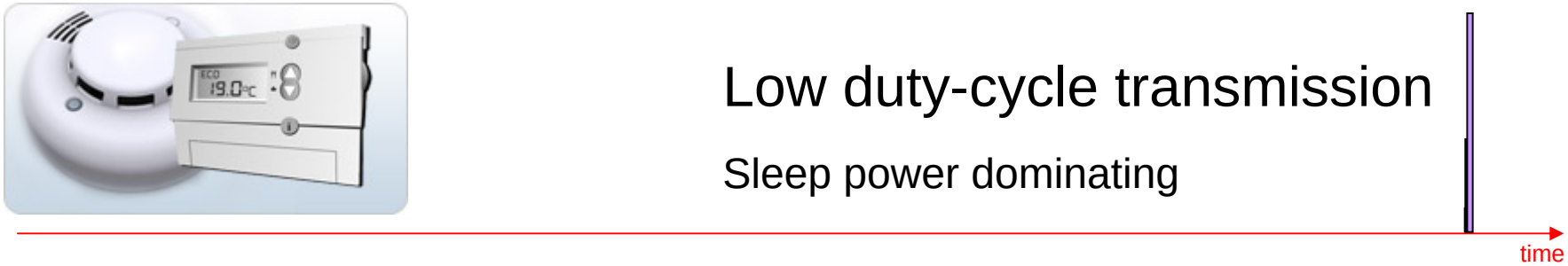
- Low sleep current (<1 uA)
- Wake on radio / sleep timer
- Fast RX/TX turn around time
- Adaptive output power using RSSI
- Fast crystal start-up time
- Fast PLL calibration (and settling)
- Terminate RX if there are no signal at the antenna: carrier sense
- Low RX peak current
- Automatic packet handling
- FIFO (for burst transmission modes)

# Power saving

- Crystal Oscilator Start-up
- Calibration
- RX/TX mode

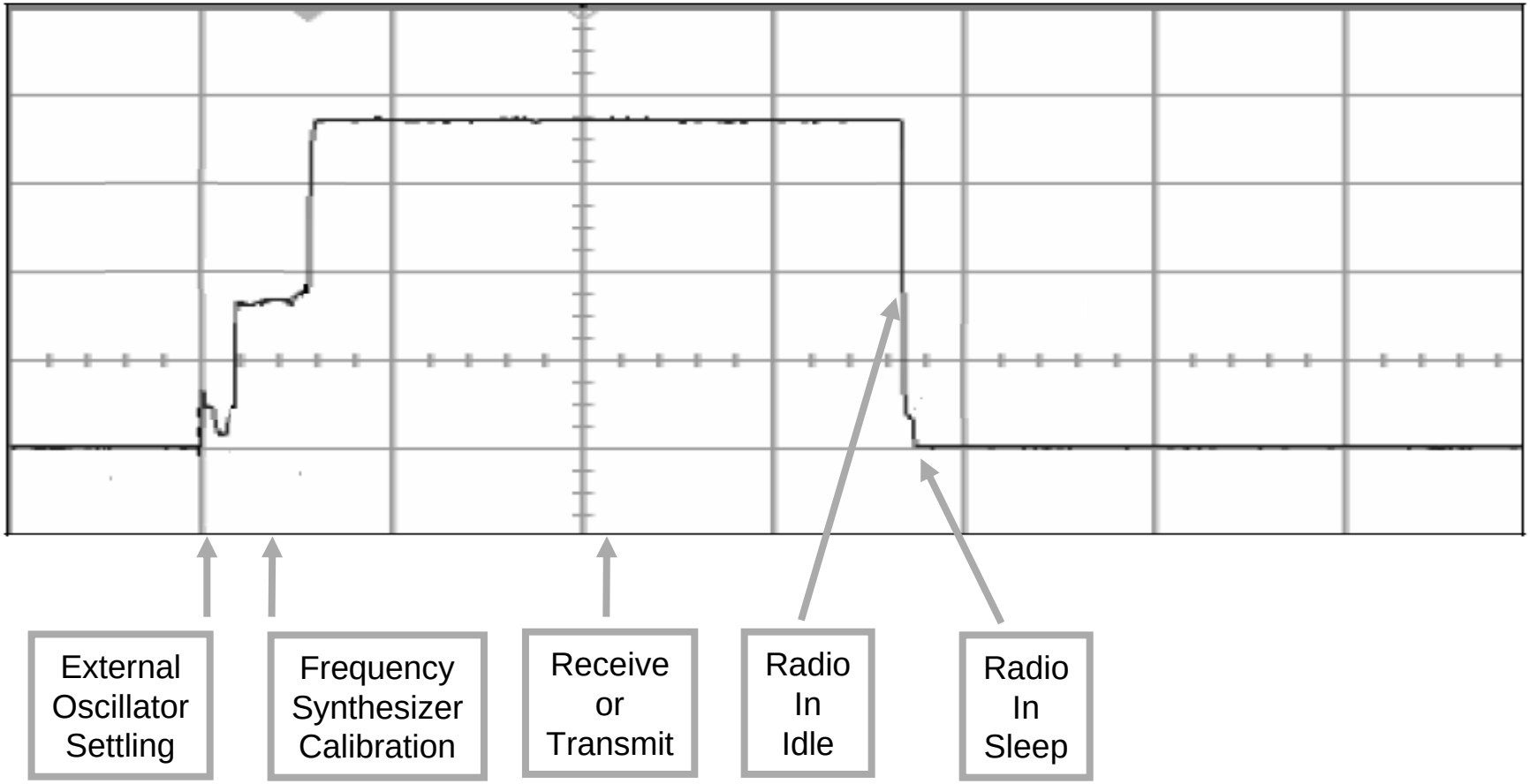


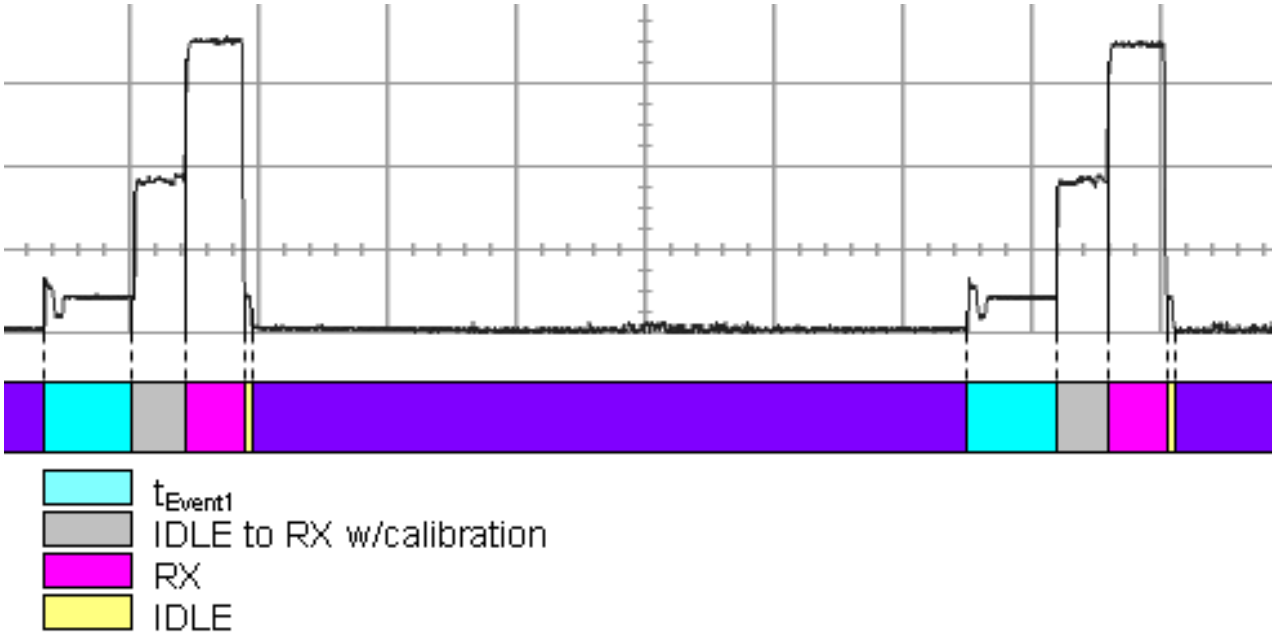
# Power saving

**Typical Power Profile of a LPRF System**



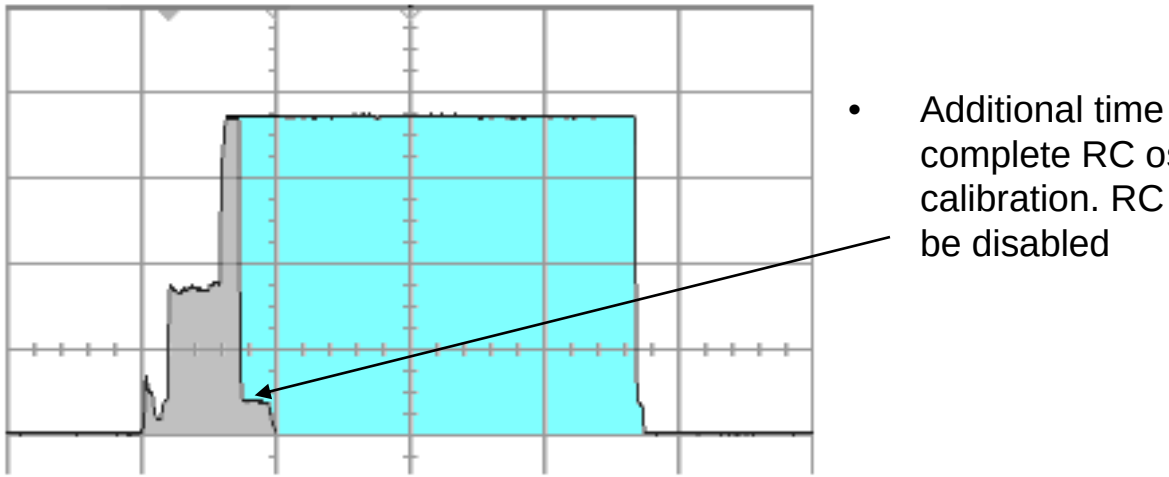
# Power saving - Wake on radio

- Wake-on-radio function allows very low average power consumption in a polling receiver (a few tens of µA)




- If data is received
  - Go to IDLE, RX (new sync search), TX or FSTXON

# Power saving - Wake on radio



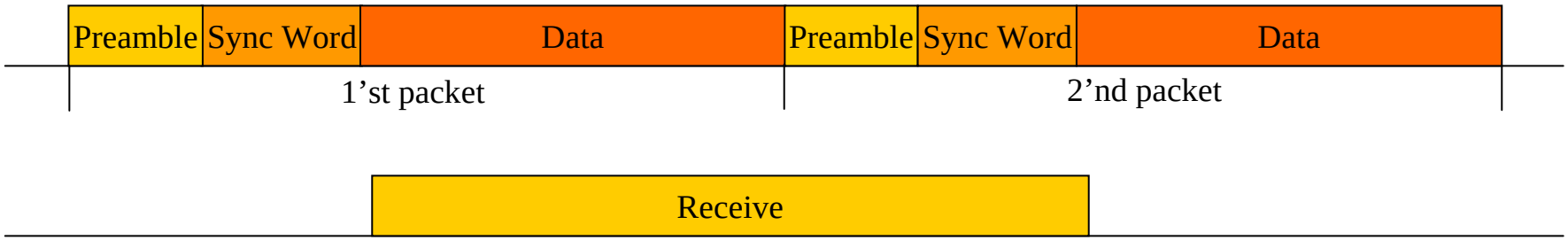
- Additional time in IDLE to complete RC oscillator calibration. RC calibration can be disabled

MCSM2.RX_TIME_RSSI = 0 or
MCSM2.RX_TIME_RSSI = 1 and RSSI is above threshold
MCSM2.RX_TIME_RSSI = 1 and RSSI is below threshold

- This diagram shows the difference in consumption depending if there are received signals above RSSI threshold or not after T_event1.

# Power saving - Polling receiver

The transmitter is sending its packet back-to-back.




The receivers timeout should be set to the time it takes to send:

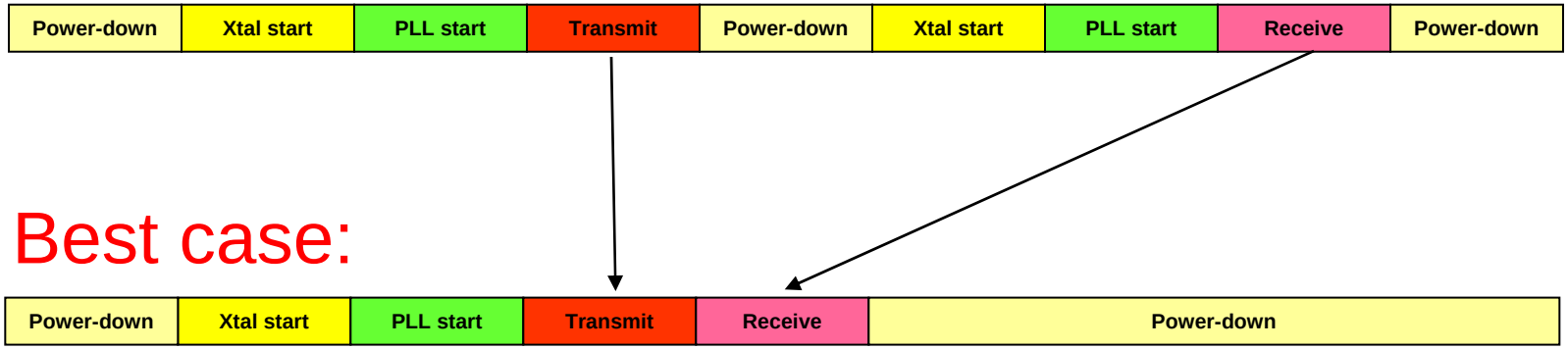
- (X bytes preamble and Y bytes sync word) <u>times two</u>.
- Z bytes data

This will ensure that if the receiver briefly misses the first sync word it will stay in RX until the next one.

# Power saving - RX/TX turn around

For 2-way protocols, go as quickly as possible from transmit to receive mode or vice versa

Worst case:




Best case:

# Power saving – packet handling

- Minimize time in RX processing false packets
  - Check carrier sense
  - Check for valid preamble
  - Check for valid sync word
  - Check length byte
  - Check for valid address

- Only notify MCU when a valid packet has been received
  - Automatic CRC check
    - Packet discarded if CRC fails
    - Interrupt to MCU if CRC OK

# Power saving – output power

- Do not transmit more power than needed!

- The output power can be lowered when operating at close range to reduce TX current consumption

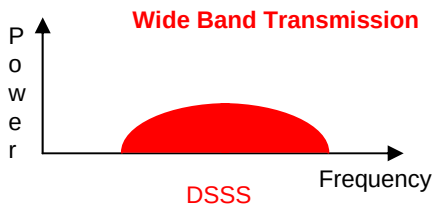
# Interference

# Interference

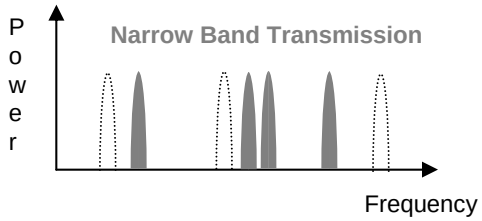
**Depending on the frequency chosen one has to cope with different types of interference.**

**For wireless systems there are different methods to handle radio interference from WLAN, Bluetooth® devices, analog video, microwave ovens, other ISM systems, Cordless phones etc.:**

- **DSSS** (Direct Sequence Spread Spectrum)
  - Increased robustness against multi-path fading
  - Increased sensitivity / robustness to narrow-band interference
  - Wider bandwidth required



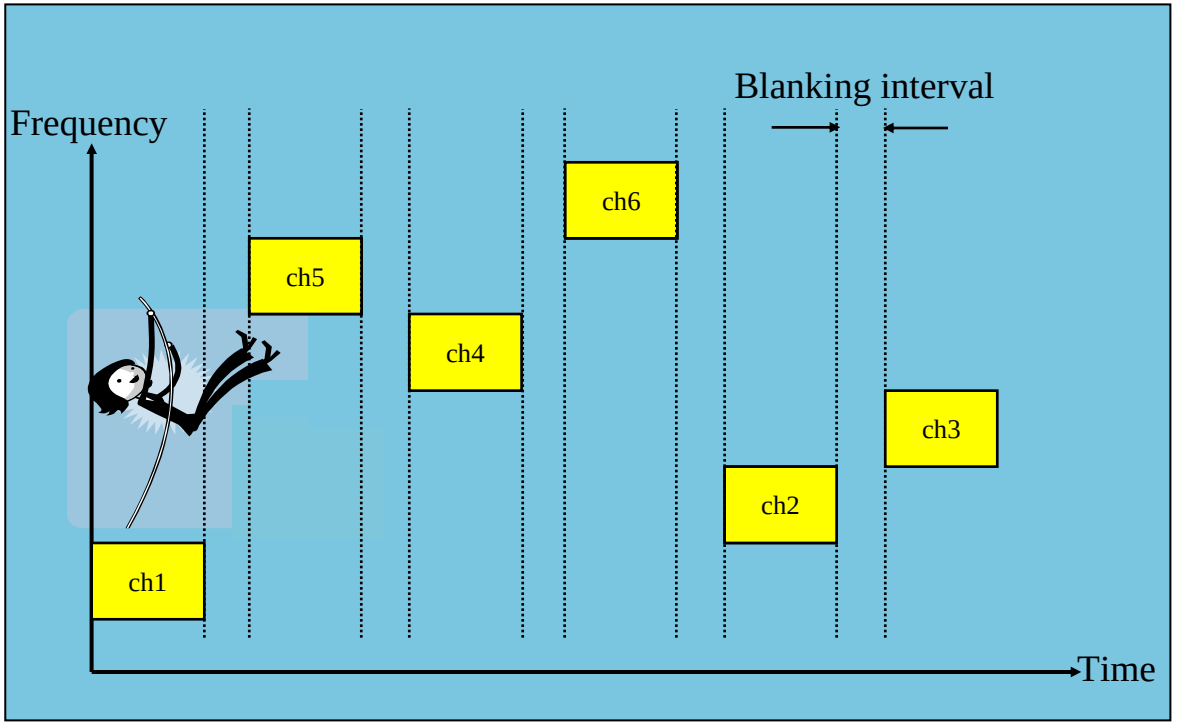

- **Frequency agility**
  - Automatic or manual channel migration
  - The implementation has to account for sleeping devices as they might miss the control signal to switch channel while being asleep
- **Frequency hopping**
  - Synchronization required
  - The implementation has to assure that sleeping devices do not get out of sync




FHSS (Frequency Hopping Spread Spectrum)

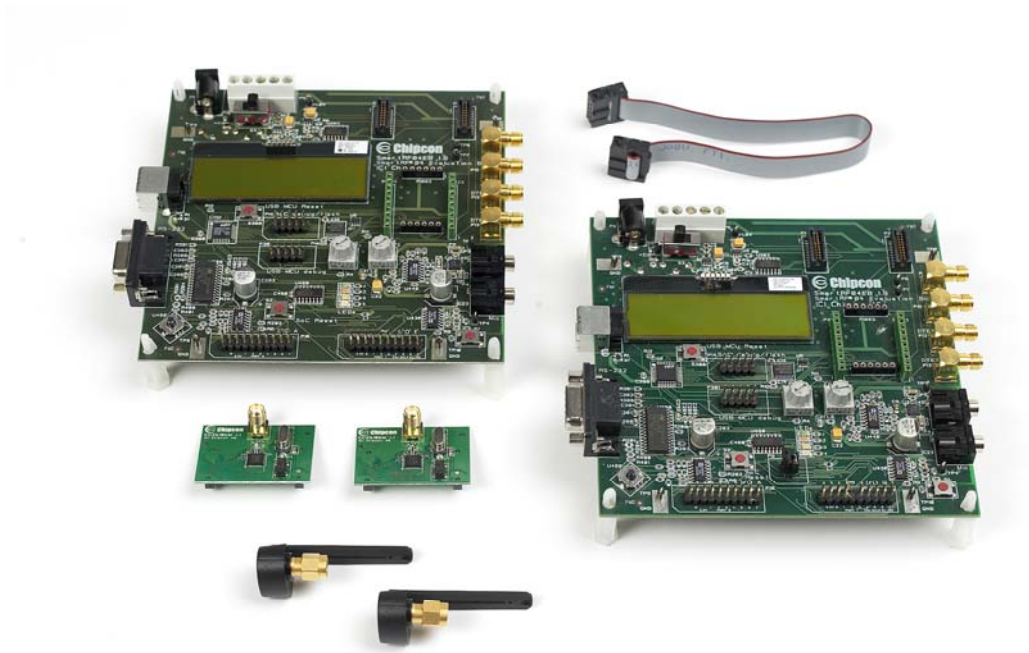
# Interference – frequency hopping

- Fast settling PLL is important to minimize blanking interval
- Minimize synchronization time



# Development Tools

# Development tools



- SmartRF Studio
- Packet Sniffer
- IAR embedded workbench
- Development Kits

# Development tools

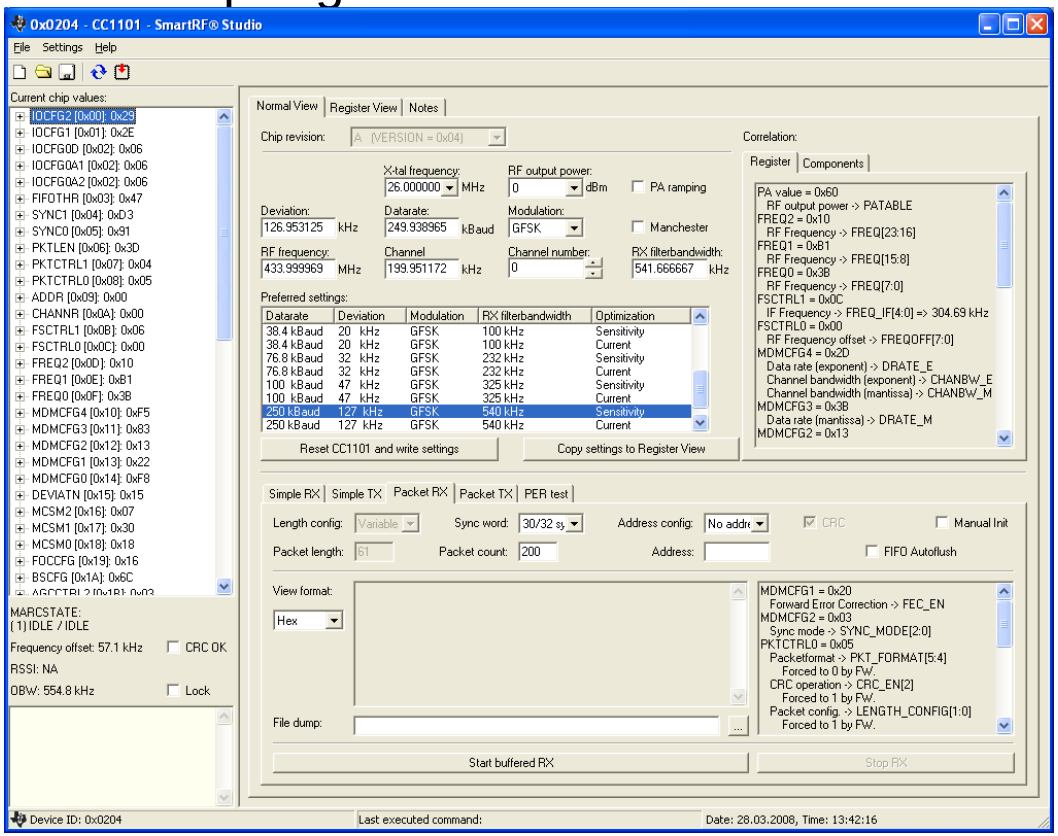
## SmartRF Studio

- SmartRF Studio is a PC application to be used together with TI's development kits for ***ALL*** CCxxxx RF-ICs.

- Converts user input to associated chip register values
  - *RF frequency*
  - *Data rate*
  - *Output power*

- Allows remote control/ configuration of the RF chip when connected to a DK

- Supports quick and simple performance testing
  - *Simple RX/TX*
  - *Packet RX/TX*
  - *Packet Error Rate (PER)*



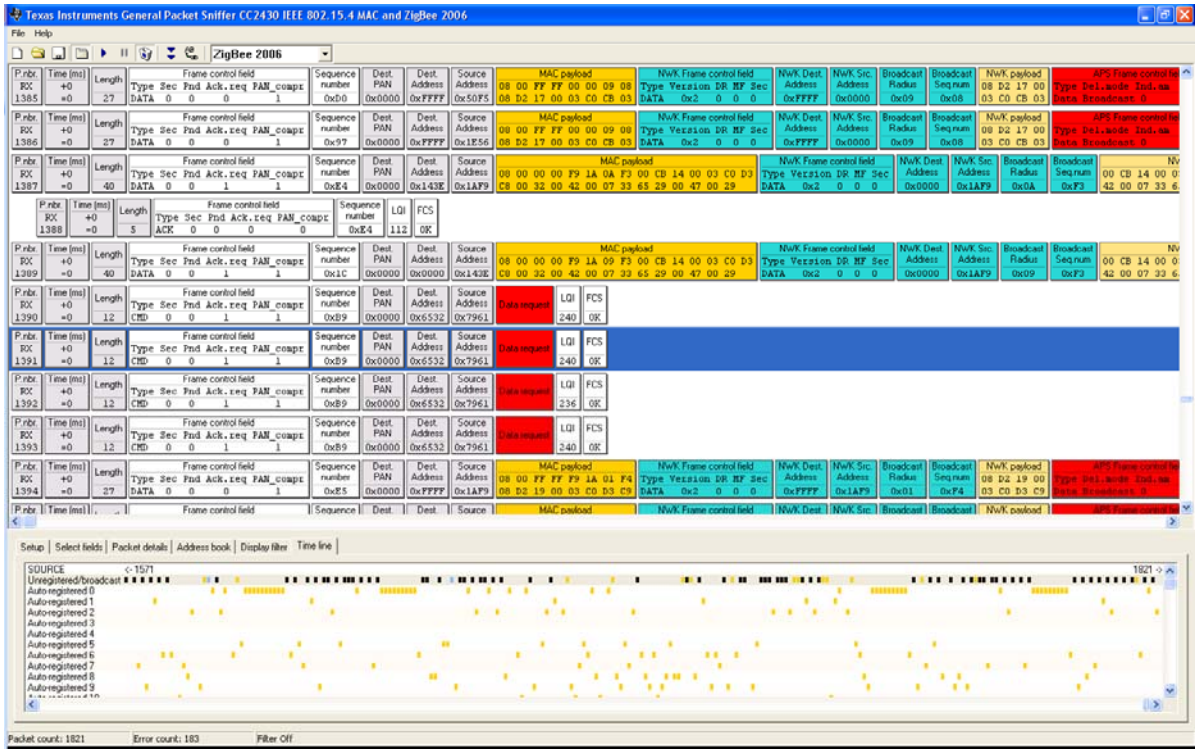
# Development tools

## Packet Sniffer

Packet sniffer captures packets going over the air

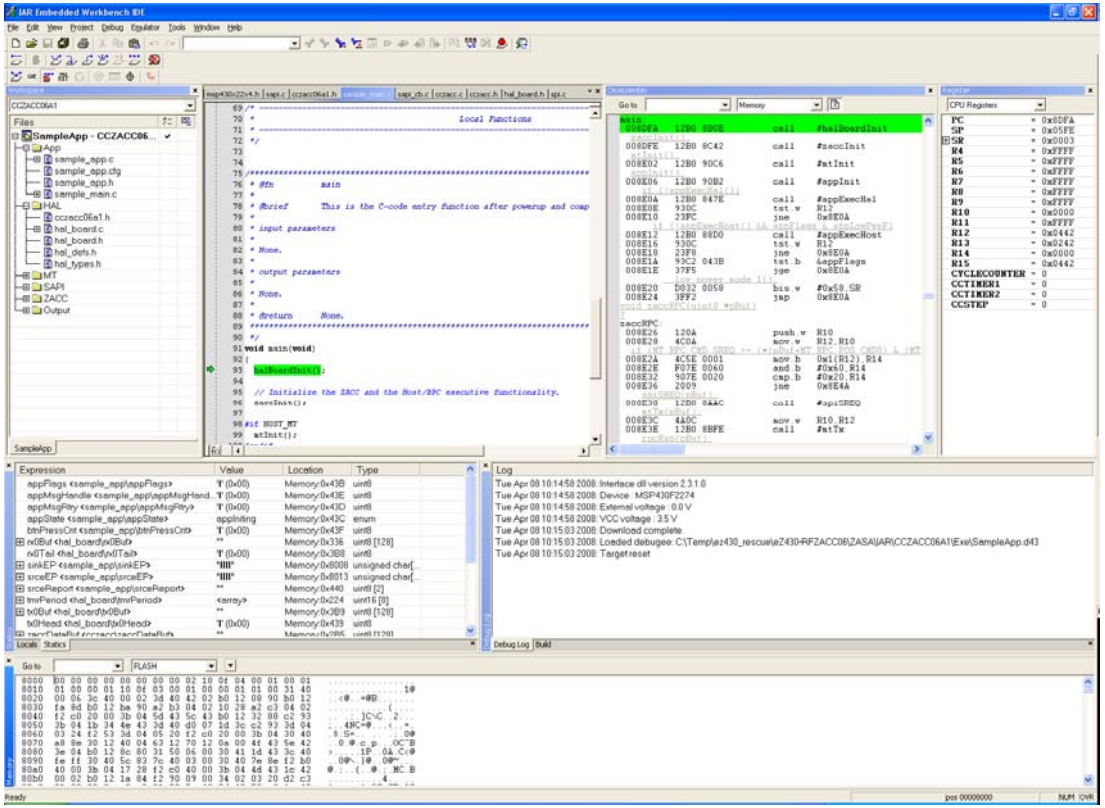
Protocols:

- SimpliciTI
- TIMAC
- RF4CE
- ZigBee
- Proprietary protocols



# Development tools

## IAR Embedded Workbench

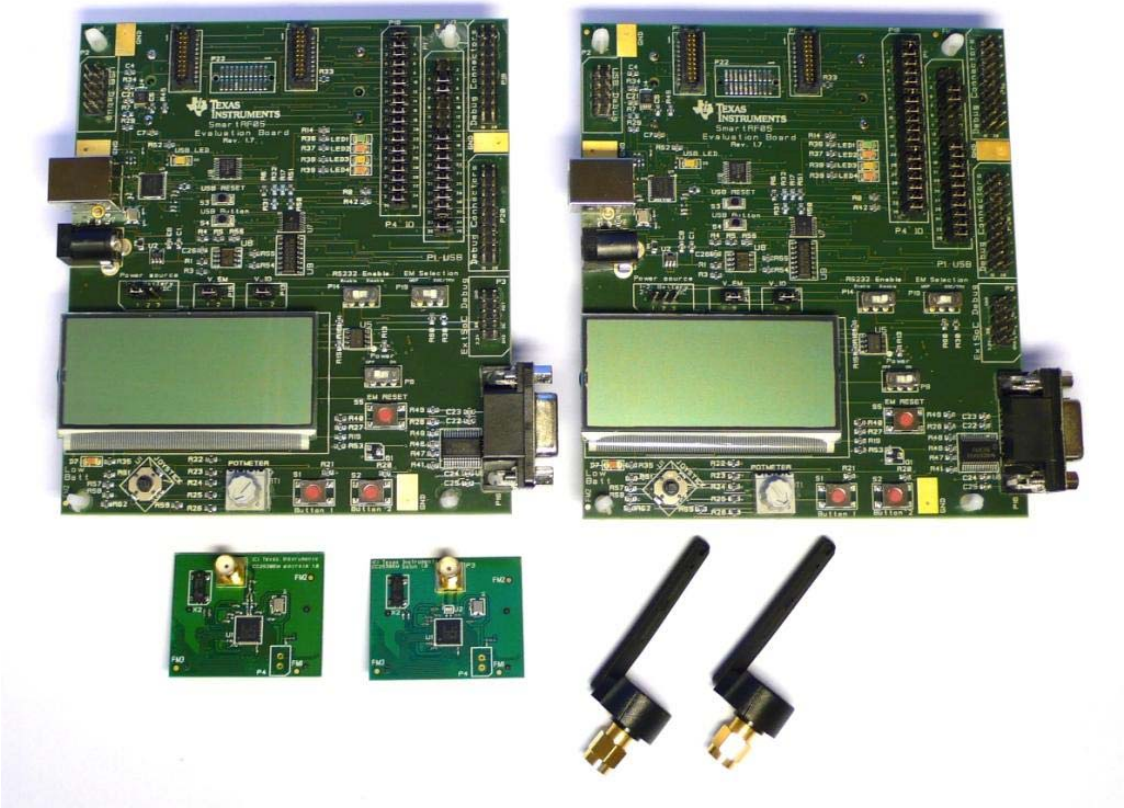


www.IAR.com

- IDE for software development and debugging
- Supports
  - All LPRF SoCs
  - All MSP430s
- 30 day full-feature evaluation version
  - Extended evaluation time when buying a SoC DK or ZDK
- Free code-size limited version (16kB)

# Development tools

## Development Kits



The typical development kit contains

2x RF EMs

2x SmartRF Boards

2x Antennas

Cables

Documentation

# Debug & Test

# Debug & Test

- Always test your Software with a known working HW
  - The Development Kits from TI is a good candidate, but your own developed hardware can of course be used as well if it has been verified to work.
- Use code or code snippets known to be working
  - TI LPRF provides code examples for all our devices.
  - This will speed up the developments process and make it easier for TI to provide efficient support.
- Test different software functions during the development phase
  - This will make it easier to find and correct bugs.

# Resources

# Software Examples

- Software available on the product folder on the web
- Software examples in Application Notes and Design Notes
- Free download – Full source code available

- Examples
  - PER Test software for the SoCs
  - USB Firmware Library and Basic peripheral examples for CC111x and CC251x
  - For basic software for MSP430 and transceivers, see www.ti.com/ccmsplib
  - Temp.sensor demo with PC GUI for the eZ430-RF2500 tool, see http://focus.ti.com/docs/toolsw/folders/print/ez430-rf2500-seh.html

# Other Resources

- Low Power RF Developer Network
- Engineer-To-Engineer Community
- FAQ, Design And Application Notes

http://community.ti.com

# Summary

## Software design considerations for a low power RF system

- System Details
  - How Many Nodes, How Much Data At What Rate, Battery vs. Mains Powered Devices, Latency, Range, Quality Of Service, Encryption
- Network Topology
  - Point-to-point, Point-to-Multipoint, Mesh
- Network Access
  - Listen Before Talk, CDMA, TDMA, FDMA, Polling Receiver
- Device Pairing Mechanism
  - Proximity, Fixed, User intervention, Commisioning
- Power Budget & Power Saving
  - Sleep Current, Start-/Wakeup Time, RSSI, Packet Handling, PLL
- Interference
  - DSSS, Frequency Agility, Frequency Hopping

# QA

**Thank you for your attention!**
**Any questions?**