TI Designs: TIDA-01531
Low-Power wM-Bus Communications Module Reference Design

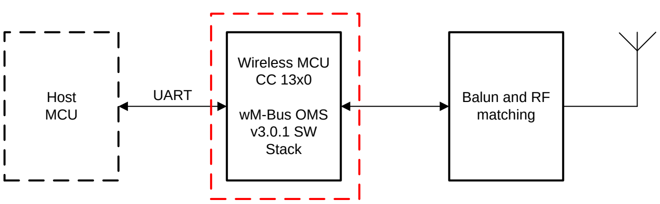
Description

This reference design explains how to use the TI wireless metering-bus (M-Bus) stack for CC1310 and CC1350 wireless MCUs and integrate it into a smart meter or data collector product. This software stack is compatible with the Open Metering System (OMS) v3.0.1 specification. EN13757-1 through EN13757-7 are European standards for meter reading and include both wired and wireless M-Bus; these together are very popular in ultra-low-power metering and sub-metering applications. This design offers ready-to-use binary images for any of the wireless M-Bus S-, T-, or C-modes at 868 MHz with uni- (meter) or bidirectional configurations (both meter and data collector). Multiple pre-compiled binary images are provided that cover metering applications, including but not limited to heat cost allocators (HCAs), gas, water, and heat meters, or e-meters with an external host MCU.

Resources

TIDA-01531	Design Folder
CC1350 LaunchPad™	Product Folder
CC1310 LaunchPad	Product Folder
CC1350 Wireless MCU	Product Folder
CC1310 Wireless MCU	Product Folder

TI E2E™ Community — ASK Our E2E Experts

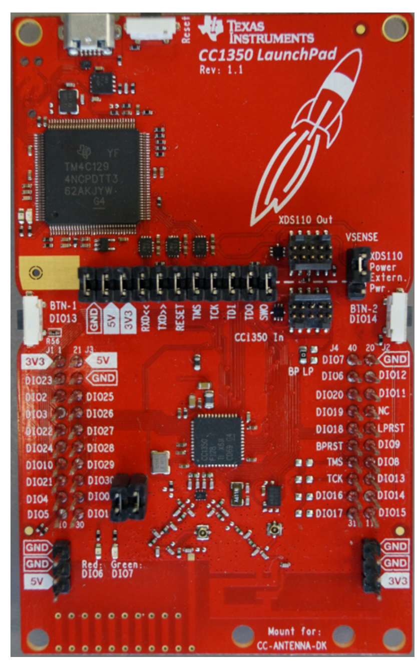


Features

- Meets EN13757-4 Class H_R Requirements for Sensitivity and Selectivity and Class H_T for Transmit Power in S-, T-, and C-Modes
- Complete Single-Chip Implementation With Serial Interface to Host MCU
- Consumes Only 0.7 µA at 3.6 V in Shutdown Mode
- Embedded (API Level) Interface for Combining wM-Bus Stack and Meter Application
- wM-Bus OMSv3.0.1 Compliant S- and T-Modes (S1, S2, T1, T2) With C1- and C2-Modes Added
- Supports Meter and Data Collector (Also Called "Other") Functionality

Applications

- Metering and Sub-Metering Systems With wM-Bus at 868-MHz Communication
- Water Meters, Heat Meters, Gas Meters, and Heat Cost Allocators
- E-Meters (as Meter Device)
- Data Collectors and E-Meters Collecting Data From Flow Meters
- In-Home Displays, Handheld Readers



An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Description

The European wM-Bus standard EN13757-4 was published in 2005 and defines a "star network" system architecture for wireless meter read-out and uses the 863- to 870-MHz ISM band. This standard was updated in 2013, when the 433- and 169-MHz narrow-band support were added. By 2017, the wM-Bus protocol has been selected as the communication interface for e-meters, gas meters, and water meters in countries like France, Netherlands, Italy, Germany, and others. Many of these countries are already deploying smart meters in significant volumes.

This reference design focuses on the 868-MHz ISM band and describes a wireless module compatible with S-, T-, and C-mode wM-Bus based on a single-chip CC13x0 Wireless MCU, which runs the TI OMSv3.0.1 software stack. This combined solution of the CC13x0 and software stack is applicable for any metering applications such as mains-powered e-meters or battery-powered gas or water meters. Other widely deployed metering devices that can be used are heat cost allocators (HCAs), heat meters, data collectors, and in-home displays.

Statements in this design guide apply also to the CC1310, even if only CC1350 is mainly mentioned.

1.1 Key System Specifications

The CC1350 Wireless MCU operates from 1.8 to 3.8 V (4.1 V is the absolute maximum value allowed). This voltage range fits well with the two popular types of primary battery cells, either 3.6 V (LiSoCl2) or 3.0 V (LiMnO2) ones. The data rate of the UART link is set to 115.2 kbps; this is higher than the 100-kbps data rate over the air and sufficient to avoid data overflow while moving data to and from the CC1350.

The wM-Bus Stack supports:

- Unidirectional S1-, T1-, and C1-mode meter configuration at 868 MHz
- Bidirectional S2-, T2-, and C2-mode meter configuration at 868 MHz
- Bidirectional S2-, T2-, and C2-mode data collector configuration at 868 MHz
- 115.2-kbps UART interface with a serial command interpreter (to the external host MCU)
- API interface to integrate the wM-Bus stack into own application
- OMSv3.0.1 compatibility

The operating ambient temperature for the CC1350 and CC1310 is T_A = –40°C to 85°C.

2 System Overview

2.1 Block Diagram

This reference design describes the wM-Bus stack developed for the CC1350 and CC1310 devices. The combination of a CC13x0 Wireless MCU and a field proven wM-Bus OMSv3.0.1 software stack delivers a true *single-chip wireless M-Bus communication subsystem* with excellent RF parameters and extremely low-power.

The design implements either uni- or bidirectional RF communication subsystem, supporting the popular S-, T-, and C-modes in the 868-MHz band. Due to its small solution size and ultra-low-power consumption, it is a perfect fit for all battery-powered sub-meters such as gas, water, and heat meters or HCAs.

The main components of this reference design are:

- CC1350 Wireless SoC (on the CC1350 LaunchPad EVM), using discrete balun and RF filtering
- TI wM-Bus Stack

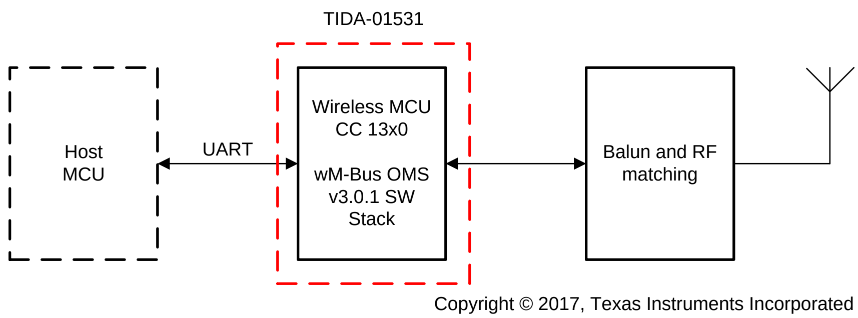


Figure 1. TIDA-01531 Block Diagram

2.2 Design Considerations

The first commercially available off-the-shelf wireless M-Bus modules were developed around 2008 to 2009 and have become quite popular. A wM-Bus module consists of an ultra-low-power MCU, executing the software stack and a Sub-1 GHz ISM band RF transceiver. There are modules with a single-chip or two-chip approach. In some cases, a third chip, such as an external PA or LNA, is added to boost the transmit power up to 27 or 30 dBm (conducted measurement) or improve the receive sensitivity. Wireless M-Bus modules are easy to integrate and shorten the development time because they handle the protocol sack and separate the time-critical metrology tasks from the wireless communication and are usually pre-certified for R&TTE [now radio equipment directive (RED)] compliance.

Using off-the-shelf wM-Bus modules like the MBUS3 and MBUS4 Radiocrafts families[9] shortens the time to market and brings many advantages over designing, testing, and manufacturing an own RF solution, especially for less experienced users. This reference design represents a *single-chip wM-Bus RF subsystem*, which is functionally equivalent to a "standard" wM-Bus module and allows full software and hardware integration into a customer product on the chip level, which is often preferred for high-volume projects.

Users have the choice of developing their own wM-Bus modules as a dedicated hardware by combining the wM-Bus software stack with one of the multiple CC1350 (or CC1310) hardware reference designs on TI.com. Users can also embed both the stack and the preferred TI hardware reference design into their own product for the most cost-optimized solution.

2.3 Highlighted Products

The wM-Bus Communications Module subsystem comprises three building elements: the CC1350 device itself, the associated CC1350 hardware design (for example, single-band or dual-band), and the TI wM-Bus OMS3.0.1 stack.

2.3.1 CC1350 SimpleLink™ Ultra-Low-Power Dual-Band Wireless MCU

The CC1350 is the first device in the CC13xx and CC26xx family of cost-effective, ultra-low-power wireless MCUs capable of handling both Sub-1 GHz and 2.4 GHz RF frequencies. The CC1350 device combines a flexible, very low-power RF transceiver with a powerful 48-MHz Cortex® M3 MCU in a platform supporting multiple physical layers and RF standards. A dedicated radio controller (Cortex-M0) handles low-level RF protocol commands that are stored in ROM or RAM, thus ensuring ultra-low power and flexibility to handle both Sub-1 GHz and 2.4 GHz protocols (for example, *Bluetooth* ® low energy; see Figure 2).

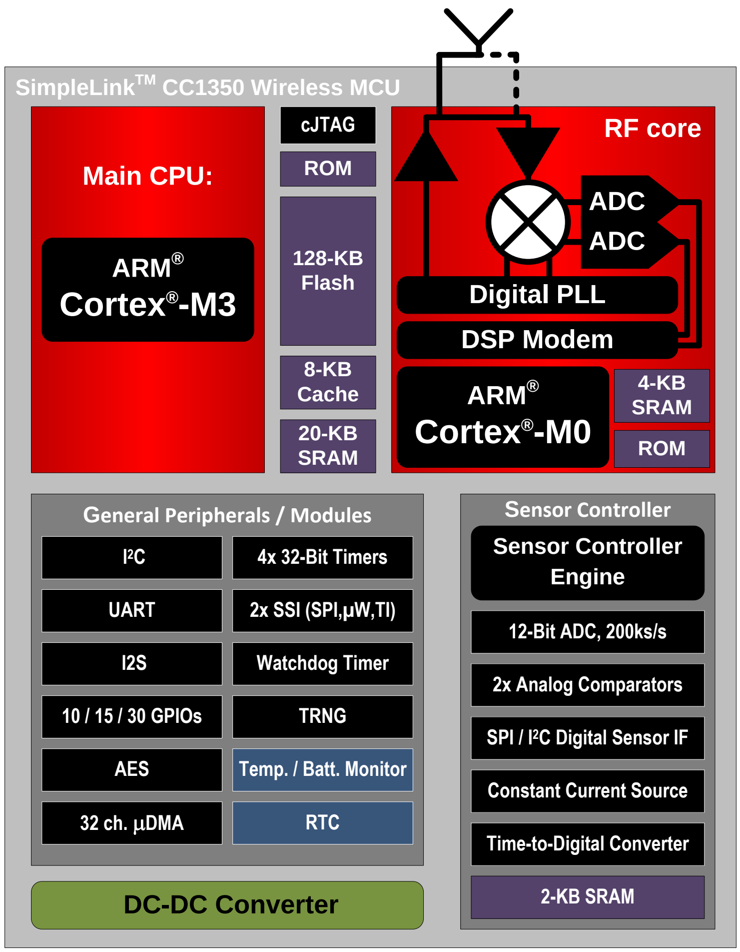


Figure 2. CC1350 Block Diagram

The ARM® Cortex-M3 runs with up to 48 MHz, and there is also 128kB of in-system programmable Flash, 8kB of SRAM for cache (or as general-purpose RAM), and 20-kB of ultra-low leakage SRAM in the device. The sensor controller has been optimized for ultra-low power and can run autonomously from the rest of the system at 24 MHz, using only 0.4 mA + 8.2 µA/MHz.

The sensor controller has a 16-bit architecture, controls the 12-bit ADC hardware block, and has its dedicated 2kB of ultra-low leakage SRAM for code and data. The CC1350 standby current is typically 0.7 µA (with RTC running and RAM and data CPU retention). The sensor controller seamlessly interfaces to the IR LED and uses its internal TDC and COMPA (Comparator A) hardware blocks to deliver outstanding low-power consumption while executing various tasks.

2.3.2 CC1350 LaunchPad (EU Version)

The SimpleLink CC1350 wireless MCU LaunchPad development kit combines a Sub-1 GHz with a Bluetooth low energy radio for the ultimate combination of easy mobile phone integration with long-range connectivity including a 32-bit ARM Cortex-M3 processor on a single chip.

For the wM-Bus communication system, the LAUNCHXL-CC1350EU kit is recommended because it is optimized for 868-MHz operation under ETSI and has been CE certified for operation in the EU. This is the only hardware for which the OMS3.0.1 stack has been developed and tested on. Other CC1350 hardware reference designs can be also used but they might require UART pins reconfiguration (depending on the hardware used).

2.3.3 TI wM-Bus Stack Software

TI is providing a free-of-charge and royalty-free wM-Bus OMSv3.0.1 compatible software stack to use with the CC1350 and CC1310. The installer package is available for download at the wM-Bus product page and contains portions of object code and API code with example applications in source.

The IDE for the wM-Bus software stack is Code Composer Studio™ v7.2.0.00013, the free integrated development environment (IDE) provided by TI, which includes a suite of tools for developing and debugging embedded applications.

The TI wM-Bus stack supports two different configurations:

1. Serial or network processor type
2. Embedded or API type

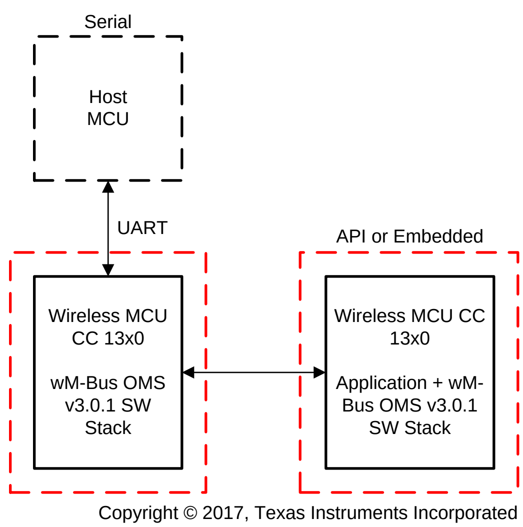


Figure 3. Two TI wM-Bus Stack Alternatives (Serial and Embedded or API)

Figure 3 shows the difference: on the left, the stack option "Serial" is shown. These are the binary images (named "*Serial*.out"), which include support for the serial commands that can be generated or received and processed by an external host MCU. This is particularly useful when the wM-Bus communications subsystem (here OMSv3.0.1) has to be separated from the main application or the metrology portion inside a smart meter. Such system architecture is typical for gas and e-meters and many feature-rich water and heat meters.

The host MCU uses the serial commands, documented in HTML pages inside the software stack installer, to interface and control the CC13x0 wM-Bus device. This serial mode of operation can be also called a wM-Bus "network processor" or "RF module", as the entire wM-Bus stack is self-contained and no stack code is executed on the host MCU (except the serial command handling). This separation of the stack and the application in two different devices avoids any critical timing between the wM-Bus communication and the host application as they run in parallel and independent from each other.

NOTE: The serial stack type together with any CC1350 hardware reference can replace any off-the-shelf wM-Bus RF module with OMSv3.0.1 functionality. This reference design focuses exclusively on the serial approach; the API solution is not reviewed.

On the right of Figure 3 is the embedded- or API-type implementation—both the user application and the wM-Bus stack run together inside the CC13x0 device. For this case, the CCS IDE tool and the API documentation of the stack function calls are used. Because the serial commands parser code is not needed, the latter is not included into the stack to reduce the flash memory footprint on the CC13x0.

The wM-Bus API-type stack version is a perfect fit for HCAs (see the TIDA-00848 and TIDA-00838 reference designs), low-cost water meters, or electronic RF add-on modules (see the TIDA-01228 reference design), where the complete product application including the wM-Bus stack will fit into the available 128kB Flash with 20-kB SRAM + 8kB cache RAM.

The ultra-low-power Sensor Controller Engine (SCE) of the CC1350 is not used in the wM-Bus stack and, if desired, can be programmed with user code. Examples for ultra-low-power tasks for the SCE can be found in multiple TI reference designs: the single-chip HCA reference design (TIDA-00848) as well as the two-chip HCA reference designs (TIDA-00646 and TIDA-00838). Furthermore, the TIDA-01212 reference design shows a half-duplex, bidirectional, single IR LED communication while the TIDA-01228 reference design introduces an inductive (or LC-sensing) approach of a small rotating disc, commonly used in mechanical water and gas meters.

NOTE: The open source example code for wM-Bus, provided together within all of the aforementioned reference designs, is *not* the same as the wM-Bus stack presented here. These reference designs have coding examples under an "open source" license and, even if they do support the S-, T-, C-, F-, or all N-modes, they do not implement the specific OMS timing requirements nor the full feature set of the OMSv3.0.1 specification.

2.4 System Design Theory

The EN13757-4 includes the implementation of a wM-Bus at a 868-MHz band, where a maximum of 25-mW effective radiated power (ERP) is allowed, except for the high power sub-band around 869.525 MHz with 500 mW, where a C2-mode data collector transmits. Due to this limited transmit power, S-, T-, and C-mode meter devices achieve the typical home area network range inside residential or office buildings. This solution has been chosen by Germany and The Netherlands as the wireless communication standard for their smart meter rollouts.

The seven narrowband N-modes (Na through Ng) in the 169-MHz ISM band, where 500-mW ERP is possible as per ERC7003, are also part of EN13757-4. This enables deployment of neighbor area networks (NANs) or even metropolitan area networks (MANs), due to the wireless link coverage of up to 2 km in dense urban areas. This solution has been chosen as the technology for gas and water meters in France as well as for gas meters in Italy. The 169-MHz solution is supported by TI through the high-performance CC1120 family and multiple MSP43x MCUs, but it is not possible on the CC1350 (nor on CC1310) due to a lack of 169-MHz band hardware support, which is outside the scope of this reference design.

2.4.1 CC1350 Hardware Solutions

Besides the CC1350 LaunchPad, there are many other TI hardware reference designs for CC13x0 that accommodate all three different CC13x0 device packages (4×4, 5×5, or 7×7 mm). Any CC13x0 package can be combined with the so-called Integrated Passive Component (IPC), which replaces the discrete Balun and RF filter circuitry for the radio, as described in the CC1310 application report[3].

The IPC solution is an excellent way to achieve the smallest possible PCB size, much simpler layout and less variation in RF performance in the 868-MHz band. TI has partnered with two vendors who offer matched balun filter components for CC13x0 devices for the 779- to 928-MHz band. The Murata LFB18868MBG9E212 and Johanson 0850BM14E0016 are pin-compatible parts that reduce the component count significantly while still obtaining high radio performance. The IPC reference design matches the performance of the discrete *multi-layer* inductor reference design with a single component.

If best-in-class RF performance is desired, then the discrete *wire-wound* inductors solution is recommended, as used on the CC1350 or CC1310 LaunchPad.

2.4.2 wM-Bus Software Stack Customization

The serial-type stack version needs access to the two UART pins. Depending on the CC13x0 package and hardware reference design used, this functionality may be assigned to different pins, which requires a minor change in the hardware abstraction layer (HAL) of the software stack. This reference design has been tested only with the CC1350 LaunchPads and requires a CC1350 PG2.1 (Rev. B) chip device to support the dedicated T- or C-mode and S-mode patches.

3 Hardware, Software, Testing Requirements, and Test Results

3.1 Required Hardware and Software

All these software tools are free and can be downloaded from TI.com (except the PC GUI tool).

3.1.1 Hardware

Use either of the following choices:

- CC1350 LaunchPad (two boards)
- CC1310 LaunchPad (two boards)

Both LaunchPad types are available for purchase through the TI online store or authorized distributors. There are no royalties pay when using the TI wM-Bus OMSv3.0.1 stack with CC13x0 devices in a consumer product.

3.1.2 Software

- FlashProgrammer2 or UniFlash4.2.x
- wM-Bus Suite (PC GUI) for configuration of wM-Bus devices
- CCSv7.2.0.00013 or later (the IDE for CC13xx wireless MCUs) is required only when using the embedded or API version of the stack to combine the stack and user application into a single device. If necessary, customize the serial interface pins to the host MCU in the serial stack version.

3.2 Testing and Results

3.2.1 Test Setup

3.2.1.1 Programming CC1350 LaunchPads

The provided binary images (also contained in the TI wM-Bus stack) have to be flashed into the CC1350 devices. This reference design uses two LaunchPads: one configured as a "meter" and the other as a "data collector" device.

The first step is to select the desired wM-Bus mode of operation(for example, T2-mode_ and program the CC1350 LaunchPads with "Serial_CC13xx_Collector_T2.out" and "Serial_CC13xx_Meter_T2.out", respectively. The "Serial_CC13xx_Collector_T2.out"binary image has been flashed with the FlashProgrammer2 Tool onto the CC1350 LaunchPad (see Figure 4).

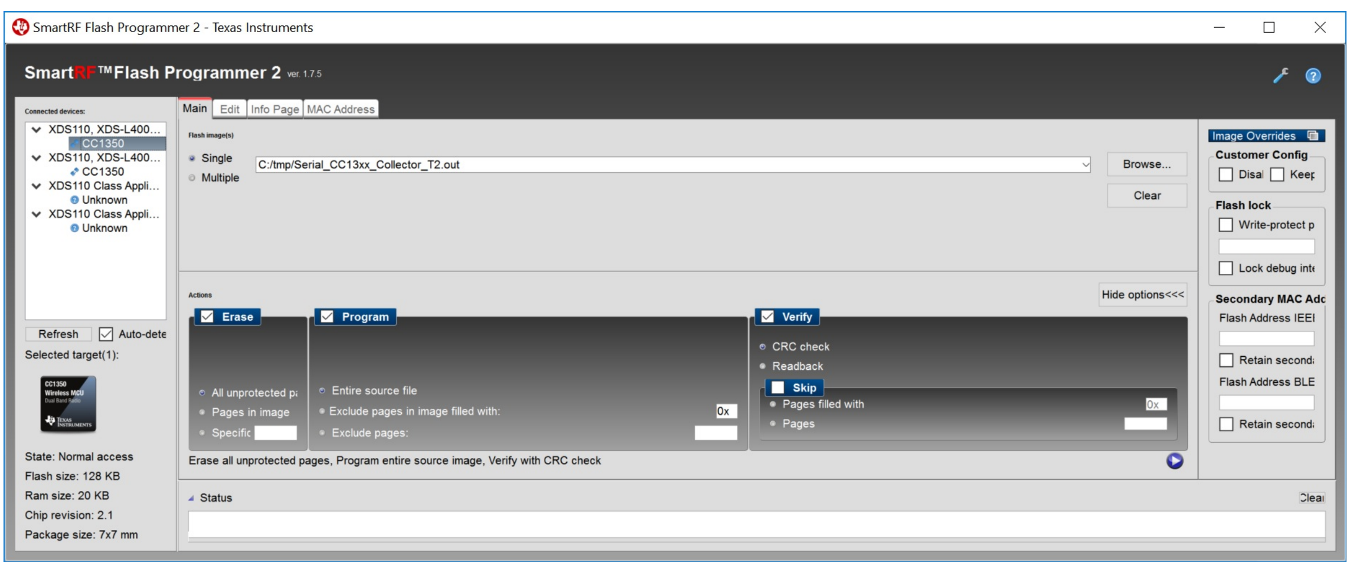


Figure 4. TI wM-Bus Stack (Data Collector Binary) Programming of CC1350 LaunchPad

Alternatively, the UniFlash v.4.2.0.1435 or later can be used as well (see Figure 5). The first step is again to select the serial image (for the desired S-, T-, or C-mode), and then load the meter binary image into the first CC1350 LaunchPad and the data collector binary image into the second.

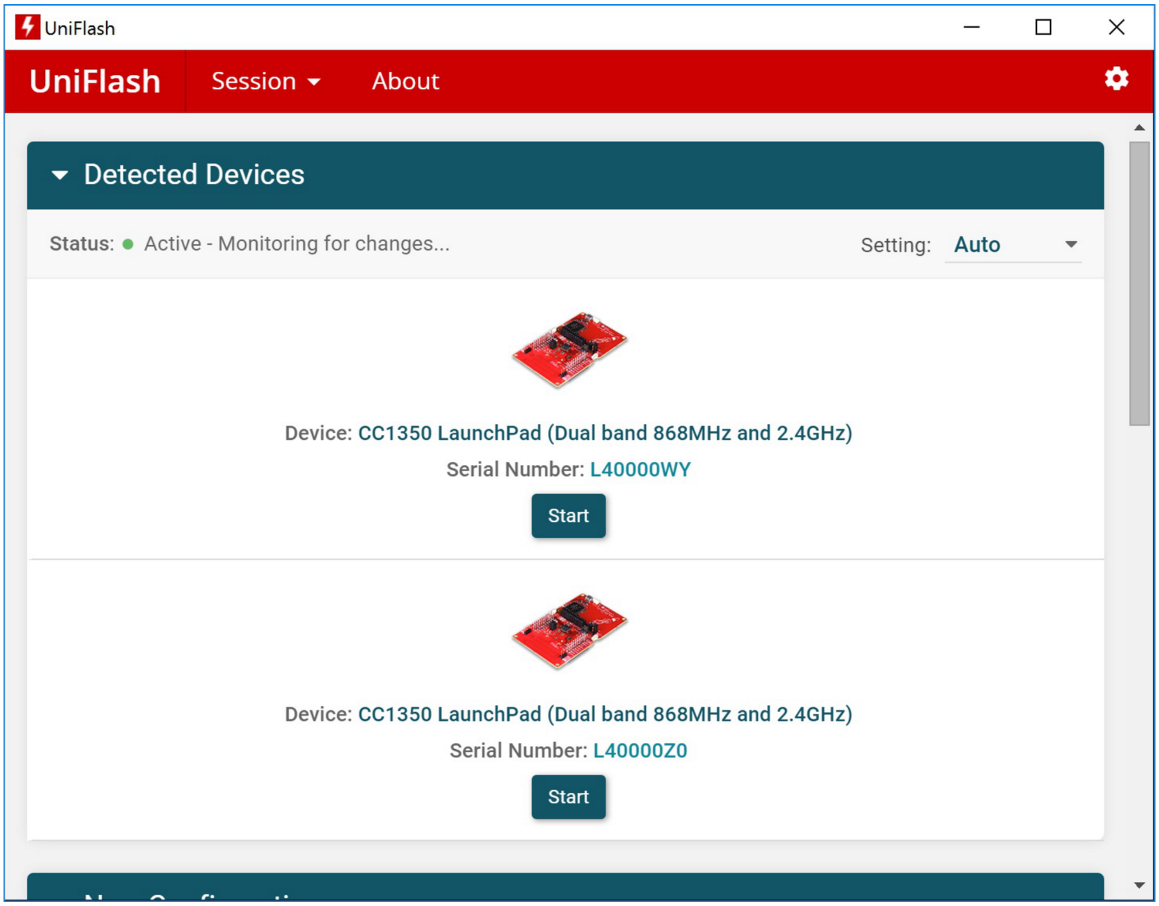


Figure 5. UniFlash v4.2 With Two CC1350LP Boards (for Meter and Data Collector)

The two freshly programmed CC1350 LaunchPads must stay connected (using the USB cable) to the PC for the next steps.

3.2.1.2 Starting the PC GUI Configuration Tool

A special Java-based tool, named Wireless M-Bus Suite by STACKFORCE GmbH (formerly Steinbeis or STZEDN), is used to configure and monitor the wM-Bus meter and data collector devices. There is a Flash video example at the "Watch the tutorial" button to the left, which quickly runs through the main functions of the PC GUI tool (see Figure 6). The tool itself implements many of the serial commands into an easy-to-use graphical interface. As described in the stack online documentation, the serial commands are visible to the user in the so-called console window, where the command flow to and from the GUI are logged.

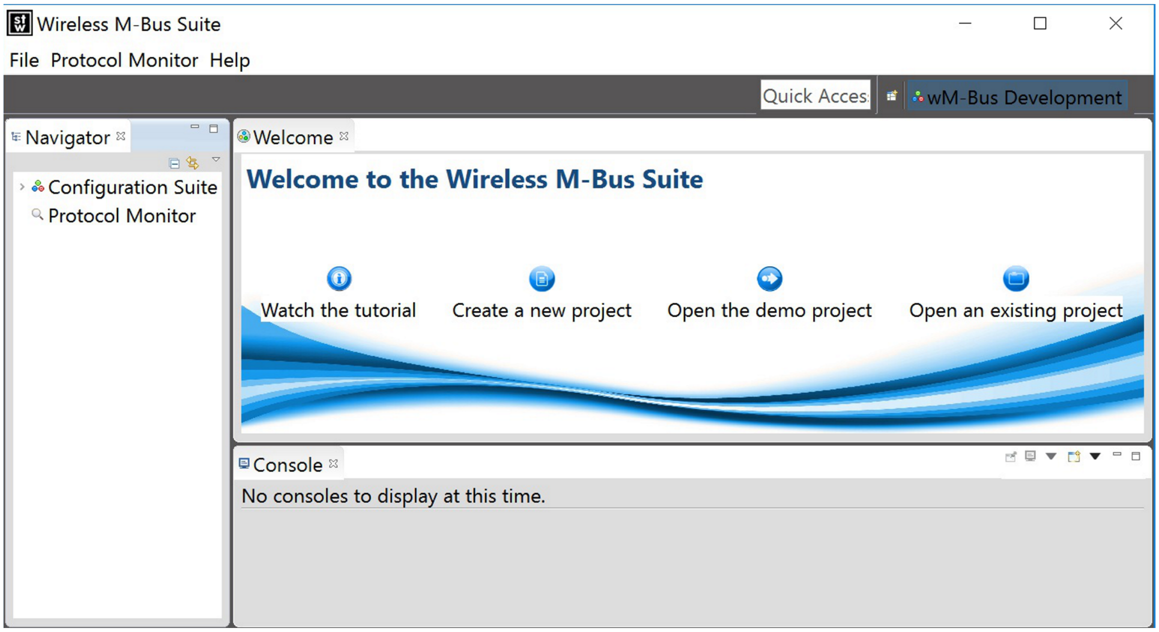


Figure 6. Wireless M-Bus Suite Start Screen

Start with "Open the demo project" and then select the "Smart Meter" entry on the left side (see Figure 7). Then click on the blue and white circle "Start wireless M-Bus demo" to run a scan of the serial COM ports.

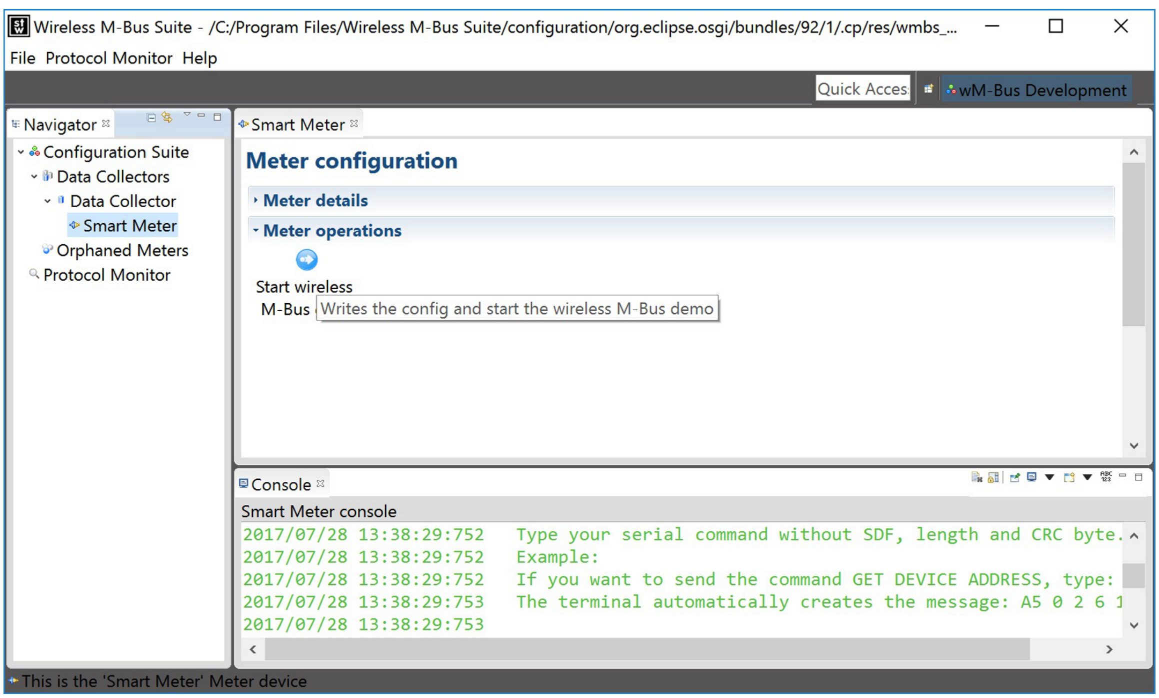


Figure 7. Start Auto-Detection of Connected wM-Bus Devices (UART Over USB)

All connected wM-Bus devices (through USB) are auto-detected; in this case, two are listed (meter and data collector). Next, select the "Meter" device and "Finish" the initialization process for the meter.

3.2.1.3 Binding the Meter to the Data Collector

The next step is to select the data collector entry in the Navigator pane (just above "Smart Meter"). Choose the already detected data collector (by the COM port scan) and then check the box "Smart Meter". This check box replaces the "Installation" procedure and the two devices are now bound to each other (meaning these devices recognize each other's encryption key and device address) and can exchange data.

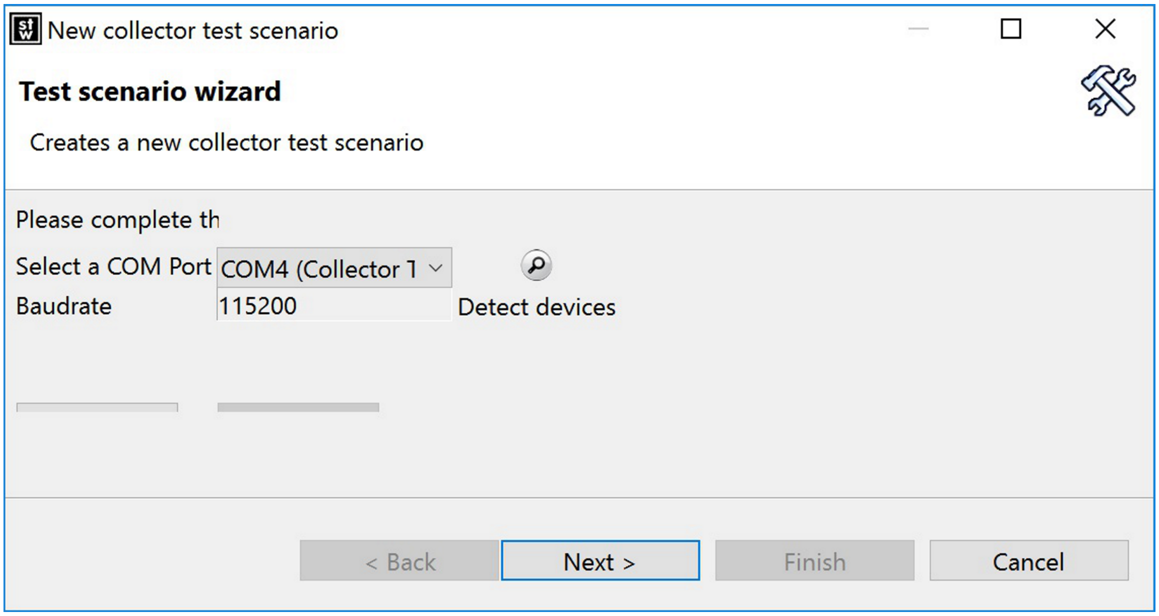


Figure 8. Select Data Collector Entry

This binding is equivalent to the installation of the meter to the data collector. The PC GUI tool enables the monitoring and control of the wM-Bus devices with Serial support; in fact, the PC GUI tool can be viewed as a "Host MCU", which sends serial commands to and receives indications from the meter and data collector units.

Using the PC GUI tool, bind or "install" as shown in Figure 9.

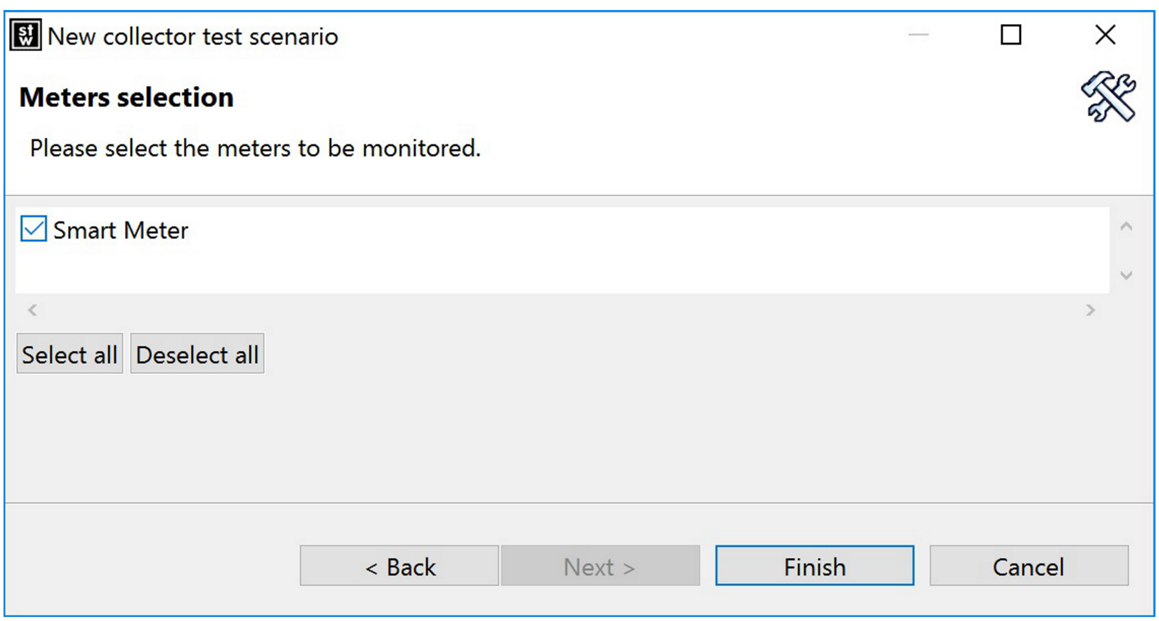


Figure 9. Binding of Meter and Data Collector With PC Tool

Finally, when both the meter and the data collector are correctly configured and launched, the PC tool resembles Figure 10.

NOTE: The Data Collector window tabs were arranged by moving them to the right through clicking and dragging.

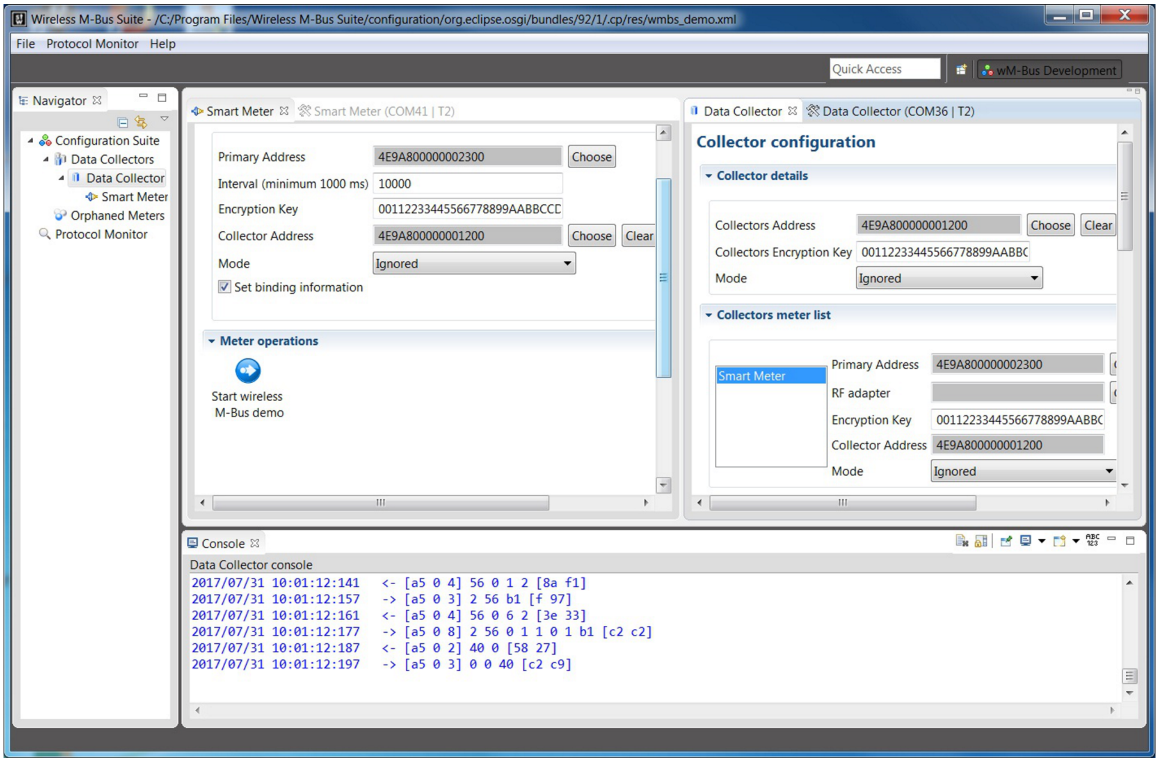


Figure 10. Fully Configured Meter and Data Collector Device in PC Tool

It is important to know that a new console can be started using the symbols on the console tab to the right. The currently active console can be pinned (see symbol in the console tab) to the current device. Then a new (second) console window can be started and assigned to the data collector device (not shown in Figure 10).

3.2.1.4 *Configuring the Data Collector*

The data collector unit needs its own configuration as well; the default values get automatically configured by the wM-Bus demo project inside the PC GUI tool. The data collector configuration on the right top in Figure 10 shows the two parameters that can be configured: the address and the encryption key. Also the meter list with already associated meters is displayed; here the default meter device is listed.

The PC GUI tool supports up to three meters in parallel by using additional CC13x0 hardware and programming with a meter binary. Next, each additional meter must get configured manually as a smart meter under the Device list in the Navigator tab. Finally, the data collector must be restarted, and the pop-up window will ask which of the configured meters must be connected (or "installed").

If all meters are selected in the Configuration window, then the data collector can receive their data packets and show them in the "Received data content" window; otherwise, the data packets get automatically discarded by the stack.

3.2.1.5 Customizing the Meter and Data Collector Parameters

According to the wM-Bus protocol, the meters periodically send out reading data, which can be collected in a data collector (stationary setup) or a handheld unit (for a drive-by read-out). The interval (10,000 ms shown) between these periodic data packets can be modified using the PC GUI, and the data transmission is encrypted with encryption mode 5 using the 16-byte key "001122...FF" as shown in Figure 11.

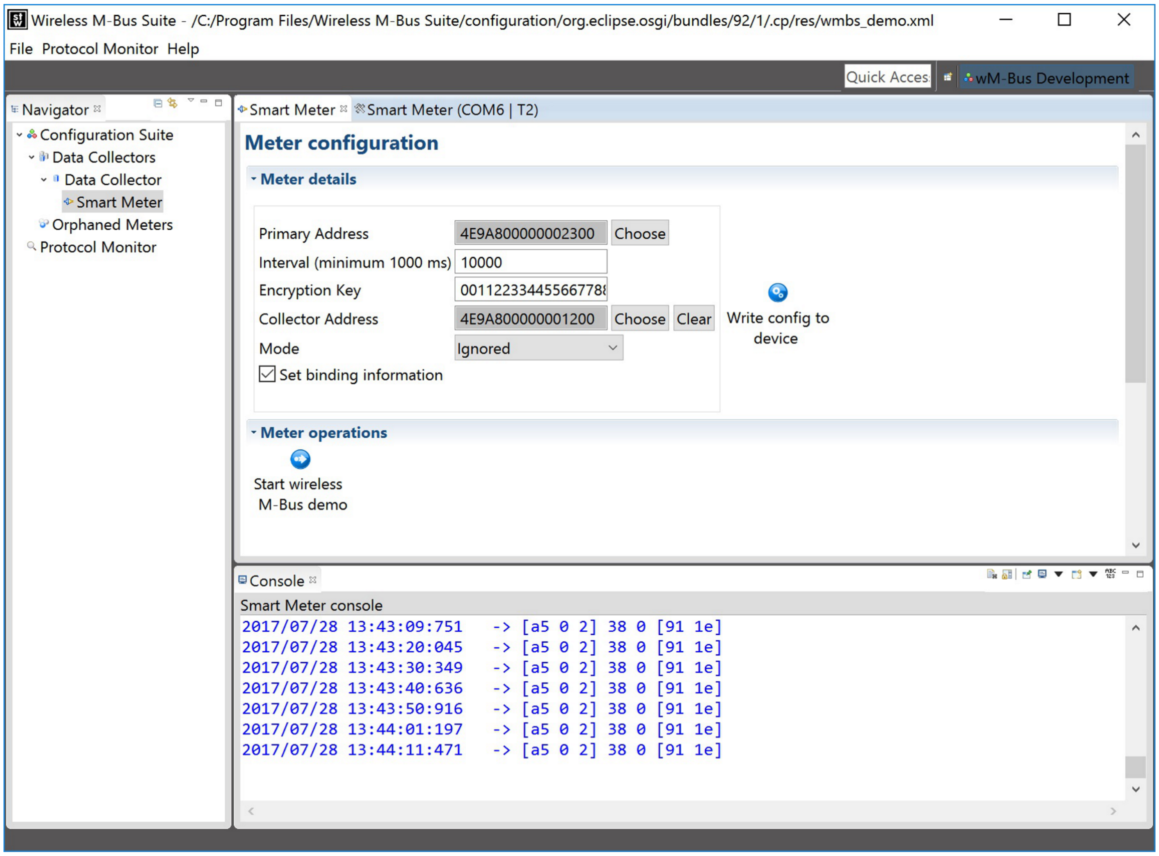


Figure 11. Meter Parameters Setup

3.2.1.5.1 Meter Address

The meter address must be unique per meter, and that is the responsibility of the smart meter vendor. The address format is: first 2 bytes for the vendor name (or manufacturer identification as per DLMS association), 4 bytes for BCD-coded identification or serial number, 1 byte for the version, and 1 byte for device type identification.

Texas Instruments has its own vendor code, TIS, as listed here.

The conversion of the string "TIS" to a 2-byte number is done according to EN13757-3 using the ASCII table value and the following:

- First letter = "T" = (54h-40h) = 0x14 =**20**d
- Second letter = "I" =**9**d
- Third letter = "S" = **19**d; where (40h = 64d)
- or "TIS" = 32 × 32 × **20** + 32 × **9** + **19** = 20787d = 5133h

All address subfield parameters can be modified or set manually and then written to the device using the "Write config to device" button on the right. If additional meter devices need to be added to the test setup, then perform the following:

- Add the new meter device under the "Data Collector" entry in the Configuration suite (left column in the PC GUI tool)
- Configure the new meter manually by adding its address and encryption key with the address of the data collector.
- Close both data collector tabs in the PC GUI tool and restart the demo to detect the data collector device. Then bind it to the two meters.

NOTE: The PC GUI tool is limited to three meters—this is a software limitation only. The TI wM-Bus OMSv3.0.1 software stack for the data collector itself supports up to 16 meter devices. This number can be further increased (requires stack customization and verification), based on free memory resources and the processing time at the data collector side, where the full meter list has to be parsed very quickly during the reception of a data packet to identify the meter and retrieve its encryption key. For example, in T2-mode, the data collector has to start transmission of the packet (preamble bits) in the time slot of 2 to 3 ms after the last transmitted bit of the meter packet (see EN13757-4).

3.2.1.5.2 Encryption Key

The encryption key for Mode 5 (mandatory for OMSv3.0.1) has 16 bytes, where at least 8 bytes of this key are different for each meter. The key must be assigned by the manufacturer and transferred safely together with the meter to the customers. This key has to be added to the data collector during the installation (by "out of band" means) to enable decryption of the meter data. At the same time, the meter device must also recognize the key of the data collector to decrypt its data packets (typically through a configuration tool during meter installation). In the PC tool, both devices use the same key (by default), but this can be modified if necessary.

3.2.1.6 Data Packet Types in wM-Bus

There four message types for the data exchange in OMSv3.0.1, which are explained in the following subsections. All of them apply to bidirectional meters, while unidirectional meters can only support some of these types. The C-field (Control) declares the message types and conforms to the unbalanced C-fields of EN60870-5-2 (source OMSv3.0.1). The C-field is 1 byte and comprises four separate bit-fields and one 4-bit function field (LSB), full details are found in the telecontrol equipment and systems[2] specification.

3.2.1.6.1 Synchronous Transmission (Spontaneous Messages Without Reply)

Unidirectional meters such as HCAs or electronic water meter add-on modules periodically transmit their consumption data. All meters must implement a mode of operation, called synchronous transmission, that is required to support battery-powered data collectors or repeaters, which only switch on their receivers for predicted short time windows.

The spontaneous messages without replies are the most popular data packets, which are used by all meters, including unidirectional ones. The interval parameter, which is set for each meter, defines the period between two transmitted packets. Table 1 in the OMS3.0.1 "Volume 2 Primary communication" document defines a maximum of 90 min for a S-mode and 15 min for a T-mode meter. Based on the defined formula, the data collector must receive at least two synchronous data packets before it can calculate the next transmission. To do that with a reasonable failure rate, it is recommended to have a continuous reception of six intervals. Thus for T-mode, 6 × 15 min = 90 min of receive mode operation for the data collector is sufficient after the meter installation. Later, the data collector goes in receive mode just a few milliseconds before the expected transmission time of each associated Meter.

The synchronous date packets must set the Bit S in the Configuration word and increase the access number (ACC-field) for each successive data packet.

As per OMS, only message types SND-NR (C=0x44 = Send/No Reply), ACC_DMD (C=0x48 = Access Demand) or ACC-NR (C=0x47 =Access/No Reply) can be synchronous.

3.2.1.6.2 Clock Synchronization (Commands to the Meter With Acknowledge)

There are also certain limits for the timing deviation for the meters in the OMS 3.0.1 so for bidirectional meters; for example, they need to use Annex F of OMS3.0.1. The clock synchronization is only applied to meters with a valid encryption key, to avoid any risk of unauthorized manipulation. There are two types of data packets defined: first with CI=0x6C "set new date and time", and the CI=0x6D "Add/Subtract Time offset", which the data collector can transmit. The bidirectional meter has to acknowledge the reception of such a clock synchronization packet, even if the timing correction is not applied. The meter is not obliged to use the encrypted time data; this is an optional feature.

3.2.1.6.3 Data Requests With Response From Slave to Master

According to OMSv3.0.1, the data collector may generate multiple C-fields, which must be accepted or processed by the meter (or actuator) and answered with its own telegram. The "SND-UD" with a C-field of 0x53 or 0x73 means "Send User Data", and the meter responds with an "ACK" data packet. The "REQ-UD1" with C-field = 0x5A or 0x7A means "Request User Data Class 1" or "Alarm Request" and shall be answered by the meter with ACK or RSP-UD data packets. The data request "REQ-UD2" with C-field=0x5B or 0x7B means "Request User Data Class 2" and requires an answer of type "RSP-UD".

3.2.1.6.4 Special Messages for Installation or Alarm

As mentioned in Section 3.2.1.3, the OMS 3.0.1 specification describes an installation process where the installation type of data packets (with C = 0x46) are sent out by the meter periodically (only after a manual event, for example, push installation button) until a timeout occurs or the installation procedure executes as expected. This means that the data collector replies with C = 0x06 data packet, confirming the correct installation of the meter.

For an alarm condition, the meter has 1 status byte for the coding of various alarm conditions, conformant to EN13757-3. This byte is part of the application layer and can be used with specific CI-field, the latter indicates the main data packet function and the type of coding (such as application protocol) used for the rest of the data packet.

3.2.1.6.5 Data Exchange Between Meter and Data Collector

Figure 12 shows a list of the received and decoded data in the Data Collector tab (COM4 | T2). See that CI field = 0x7A, 8-byte packet length, and the receive time stamp; the data packet contents can be viewed by clicking on the "Details" field to the right.

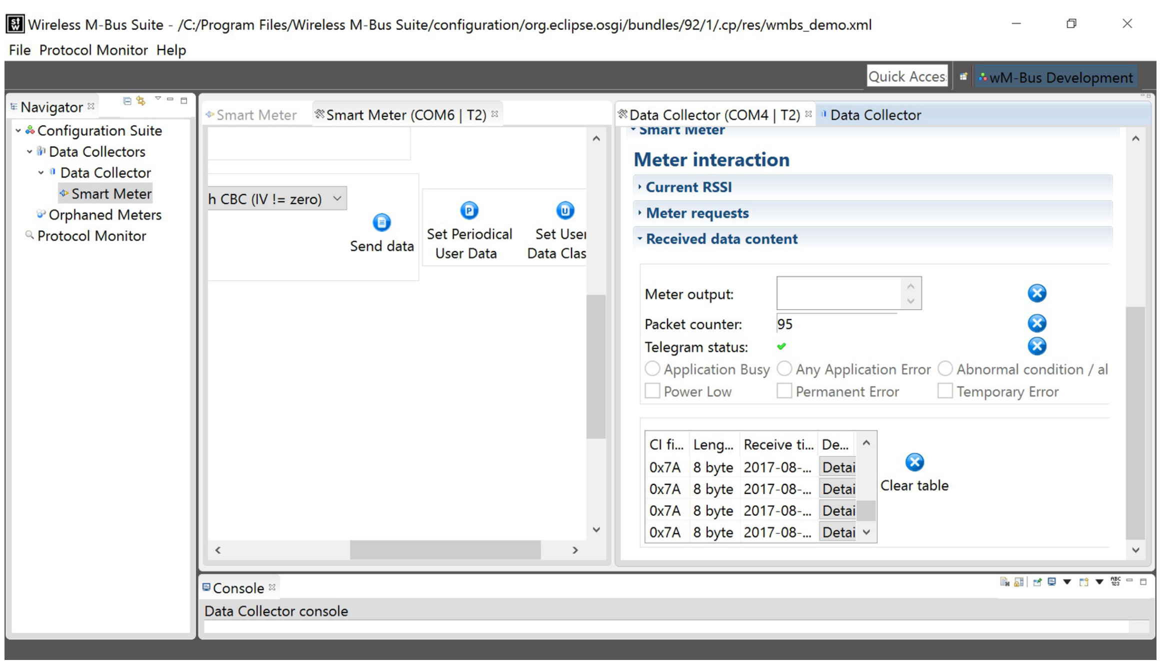


Figure 12. Unidirectional Periodic Data From Meter

The default packet content is at "0" and has a time stamp of "01-01-2000" because the time of the meter device has not been updated by the data collector yet.

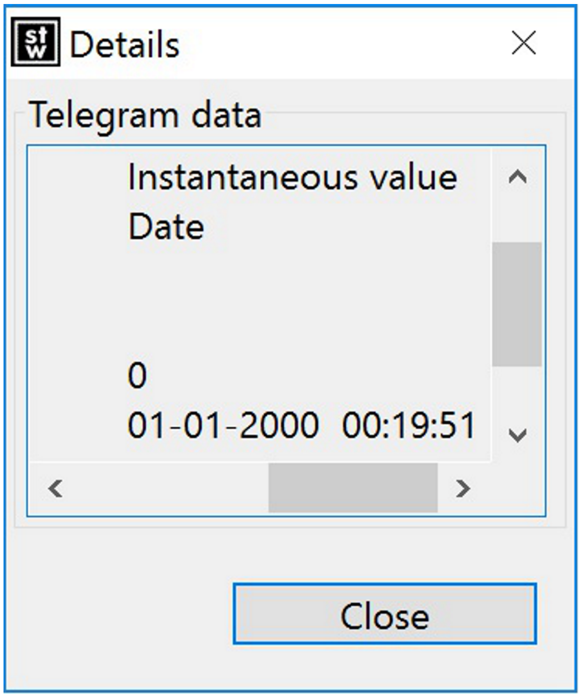


Figure 13. Default "0" Value Meter Data Packet

Now, to transmit the real-world data, the user has to adjust the value in the data packet, which is done by selecting the "Set Periodical user Data" under the Smart Meter (COM 6| T2) tab. After making the changes to the Data Field, Function, Unit, Multiplier, and Value parameters, press "Set Telegram Data" to save these values. In Figure 14, the value of 123 Wh in a 24-bit Integer representation has been chosen.

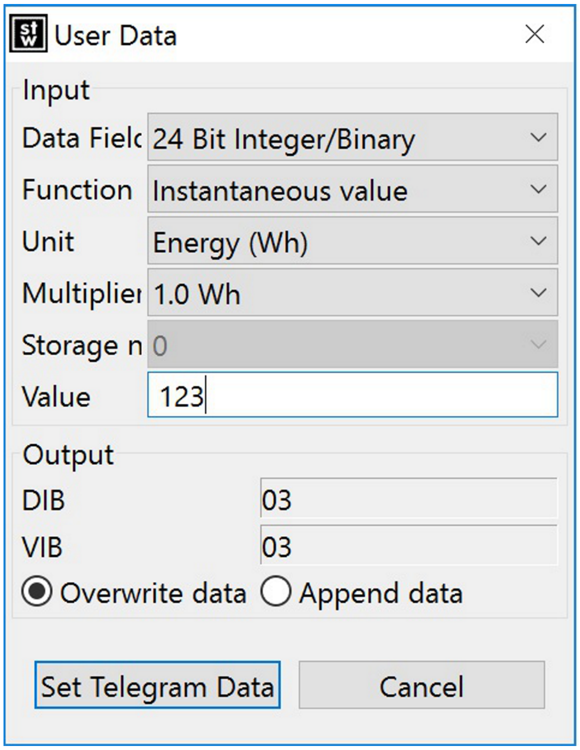


Figure 14. Modified "123" Energy Value for Meter Data Packet

This new value is used in the next periodic (or synchronous) data packet of the meter, which is received and decoded by the data collector (see Figure 15). Note that the time has been synched, as the decoded packet shows the actual time, starting with the date "31-07-2017".

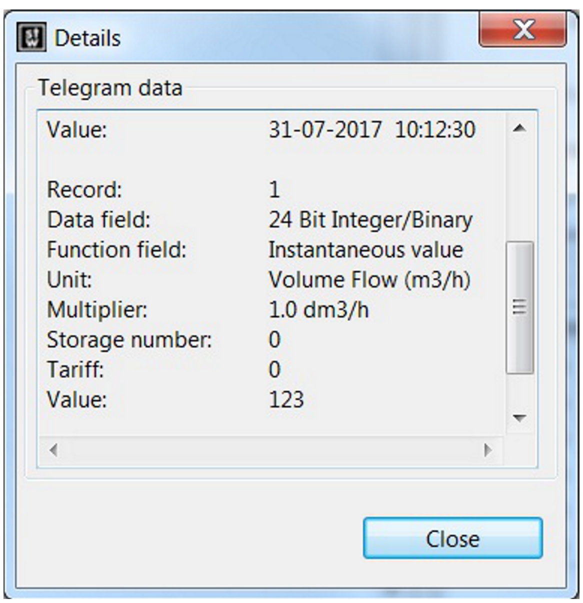


Figure 15. Data Packet With Synchronized Time and User Defined Content

3.2.1.6.6 Time Synchronization by the Data Collector

After adapting the telegram content, run the time synchronization procedure. This is initiated in the Data Collector tab (COM4 | T2) under "Meter requests" (see Figure 16):

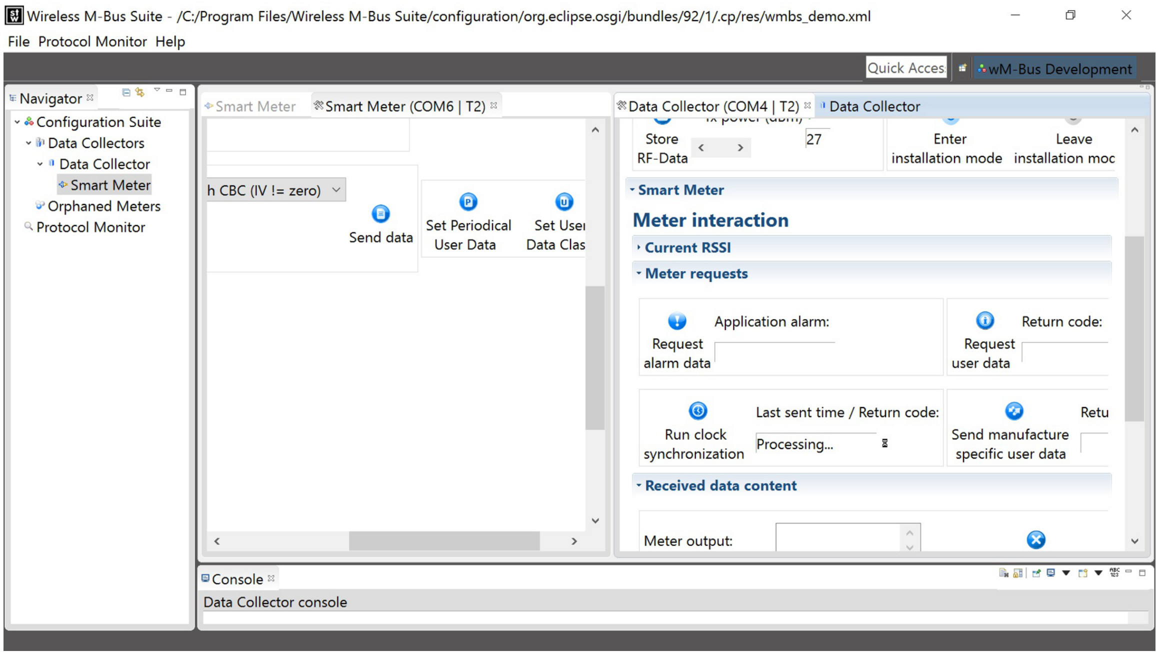


Figure 16. Time Synchronization in Progress

Figure 16 shows that this does not happen immediately (see "Processing...") but is delayed until the next data packet of the meter is received. The data collector in T2-mode has to send the Time Synchronization request in the very short receive window of 2 to 3 ms after the T-mode transmission frame, as this is the only period where the T2-mode meter is listening to receive data. If this request from the data collector is received by the meter, then a short exchange of a few telegrams takes place. At the end, the GUI shows "Passed" with a green check sign behind.

Beginning with the next telegram, the meter uses the actual date and time, which is in fact the PC time where the PC GUI is running. This is also visible under "Details" for the newly received data packets from the meter (see Figure 15).

3.2.1.7 Serial Interface Overview

The wM-Bus Stack has been developed with the idea that a serial communication is always started by the host controller (for example, sending a request). However, events can be called by the radio module to indicate, for example, when a data packet has been received. Furthermore, all commands transmitted by the host controller over UART must be replied by the radio module; otherwise, this could indicate a hardware or software problem.

3.2.1.7.1 Binary Command Format

Each commando comprises three parts: header, data part, and a footer, where the header has always the value of 0xA5 and is used as a Start Frame Delimiter (SFD), followed by a 2-byte length (MSB) that sums all bytes following the length field. The 1-byte type field is the last part of the header and describes the data after the header (the second part). The CRC16 is the third part and is built as per EN13757-4; to calculate the cyclic redundancy check (CRC), the first 3 bytes (SFD + 2-byte length) are excluded.

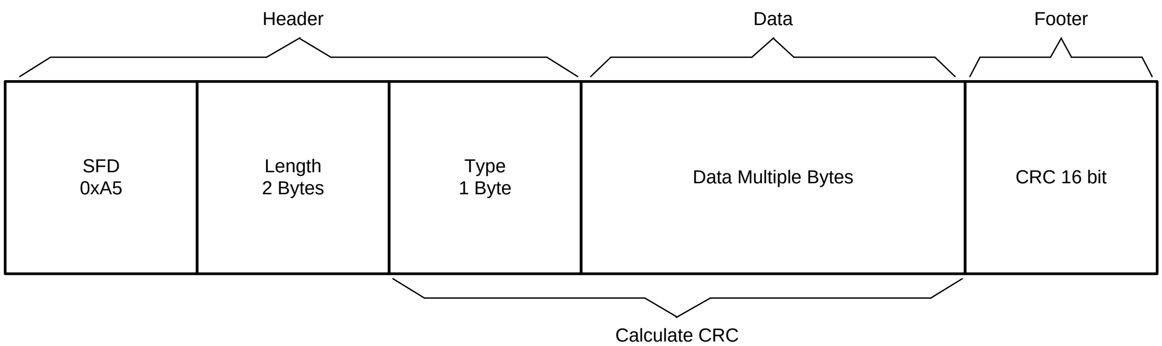


Figure 17. Commands Format Used in Serial Version of TI wM-Bus Stack (by STACKFORCE GmbH)

The command data is transferred over the UART connection to or from the CC13x0 using the format of 115.2kbps, no flow control, and 8N1 (8 data bits, no parity, 1 stop bit).

Note that the UART connection on the CC1350 LaunchPad runs over a USB cable as a COM port and can be accessed with any PC terminal program.

3.2.1.7.2 CRC Field

To protect the serial commands from manipulation or data corruption, a checksum field of 2 bytes is used. This CRC field is computed over the Type and the Data bytes only (SFD and the following two length bytes are excluded).

The formula or the CRC polynomial is:

$$x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^{8} + x^{6} + x^{5} + x^{2} + 1 \quad (1)$$

The initial value is 0, and the final CRC is complemented.

An open source C-code example implementation for such a CRC calculation is found in the firmware of the TIDA-00848 or TIDA-01228 reference designs:

```
/*
 * Filename: crc.c
 *
 * Description:    calculates CRC16 as per EN13757-4
 *
 * Copyright (C) 2008 Texas Instruments Incorporated - http://www.ti.com/
 *
 *
 *  Redistribution and use in source and binary forms, with or without
 *  modification, are permitted provided that the following conditions
 *  are met:
 *
 *    Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 *
 *    Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the
 *    distribution.
 *
 *    Neither the name of Texas Instruments Incorporated nor the names of
 *    its contributors may be used to endorse or promote products derived
 *    from this software without specific prior written permission.
```

```
 *
 *  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
 *  "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
 *  LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
 *  A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
 *  OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
 *  SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
 *  LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 *  DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
 *  THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 *  (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
 *  OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 *
*/

#include "crc.h"
#include <stdint.h>

//---------------------------------------------------------------------------------------------------
-------
//  uint16 crcCalc(uint16 crcReg, uint8 crcData)
//
//  DESCRIPTION:
//     Calculates the 16-bit CRC. The function requires that the CRC_POLYNOM
//     is defined, which gives the wanted CRC polynom.
//
//  ARGUMENTS:
//      uint8  crcData  - Data to perform the CRC-16 operation on.
//      uint16 crcReg   - Current or initial value of the CRC calculation
//
//  RETURN:
//      The value returned is the 16-bit CRC (of the data supplied so far).
//---------------------------------------------------------------------------------------------------
-------

uint16_t crcCalc(uint16_t crcReg, uint8_t crcData)
{
    uint8_t i;

 for (i = 0; i < 8; i++)
  {
    // If upper most bit is 1
    if (((crcReg & 0x8000) >> 8) ^ (crcData & 0x80))
      crcReg = (crcReg << 1)  ^ CRC_POLYNOM;
    else
      crcReg = (crcReg << 1);

    crcData <<= 1;
  }

  return crcReg;
}
//  CRC_POLYNOM         0x3D65
```

3.2.2 Test Results

A basic test of the stack binary images has been done against an RC1180-MBUS3 kit from Radiocrafts. The synchronous data packet transmissions in T- and S-mode are captured successfully, and the data content is displayed correctly when using the MBUS-DEMO software by Radiocrafts.

In addition, by using the TI Smart RF Studio 7 tool with XML configuration file for TC-modes reception, it is possible to receive the "over the air" data as well. Here, a fixed length setting longer than the expected wM-Bus packet must be used because SRF7 Studio is not able to decode the wM-Bus length byte "on the fly" and calculate dynamically the length of the wM-Bus data packet.

Additional, extensive tests with varying packet lengths for all modes are done by STACKFORCE GmbH during the stack validation process. Also, the timing requirements are measured and verified, such as the time critical 2- to 3-ms response window of the T-mode data collector with 16 associated or installed meters.

One difference that has been implemented and verified is the long header requirement for S1- and S2-mode in the OMSv3.0.1 document. According to EN13757-4, both short and long headers in S2-mode are possible; the TI stack follows the OMS specification while the latter requires long headers.

The H_T class for transmitter performance in EN13757-4 requires a minimum ERP of 5 dB for a meter and 8 dBm for others (or a data collector) in S-, T-, or C-modes (R-mode is not supported by the stack). Both CC1350 and CC1310 meet this requirement, with the typical 14 dBm at the antenna input with combined TX and RX paths in any of the S-, T-, and C-modes.

The H_R class for receiver performance is more challenging to meet because both the minimum RX sensitivity and RX selectivity criteria must meet.

In S-mode, the sensitivity is –111 dBm for an 80% PER, 20-byte payload and thus better than the required –105 dBm typical value. The measured S-mode selectivity (using a 196-kHz RX filter bandwidth) is better than 40 dB at a ±200-kHz offset and above (see Table 5 and Table 6 in the application report *CC13xx wM-Bus S-Mode*[7]).

For T-mode, the sensitivity is –105 dBm for 1% BER; for C-mode, it is –106.3 dBm at 1% BER (see Table 7 and Table 9 in the application report *CC13xx Combined wM-Bus C-Mode and T-Mode*[6]). Both values meet the common target of –105 dBm typical sensitivity at 1% BER for a T- or C-mode receiver.

The measured selectivity for the T- and C-mode patch is > 40 dB at a ±200-kHz offset and above as shown in Table 1:

Table 1. Measurement Results of T- and C-Mode Selectivity (Selectivity in dB)

OFFSET (kHz)	T-MODE	C-MODE
–400	40.5	40.8
–300	40.8	41.0
–200	40.8	41.0
200	43.3	43.4
300	44.8	45.0
400	46.0	46.1

Thus CC1350 and CC1310 fulfill the H_R receiver class requirements as per EN13757-4 for sensitivity and selectivity. Note that the immunity requirements as per EN 301 489 are measured at the full system level and thus no statement can be made on the CC13x0 subsystem level.

Both S-mode and CT-mode patches have been integrated into the TI stack and are being automatically involved, depending on the wM-Bus mode used. For example, in a T2-mode meter binary, the TI Stack uses the CT-mode patch for transmitting data and then change to the S-mode patch in receive direction. The switch between the patches happens fast enough to enable S-mode data reception in 2 ms after the last bit of the T-mode data packet has been transmitted.

4 Design Files

4.1 Schematics

To download the schematics, see the design files at CC1350 LaunchPad.

4.2 Bill of Materials

To download the bill of materials (BOM), see the design files at CC1350 LaunchPad and TIDA-01531.

4.3 PCB Layout Recommendations

4.3.1 Layout Prints

To download the layer plots, see the design files at CC1350 LaunchPad and TIDA-01531.

4.4 CAD Project

To download the CAD project files, see the design files at CC1350 LaunchPad and TIDA-01531.

4.5 Gerber Files

To download the Gerber files, see the design files at CC1350 LaunchPad and TIDA-01531.

4.6 Assembly Drawings

To download the assembly drawings, see the design files at CC1350 LaunchPad and TIDA-01531.

5 Software Files

To download the software files, see the design files at CC1350 LaunchPad and TIDA-01531.

6 Related Documentation

1. Beuth, *Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio meter reading for operation in SRD bands)*
2. Beuth, *Telecontrol equipment and systems - Part 5: Transmission protocols - Section 2: Link transmission procedures (EN60870-5-2 or IEC 870-5-2)*
3. Texas Instruments, *CC1310 Integrated Passive Component for 779-928 MHz*, Application Report (SWRA524)
4. Texas Instruments, *CC1350 SimpleLink™ Ultra-Low-Power Dual-Band Wireless MCU*, Datasheet (SWRS183)
5. Texas Instruments, *CC1310 SimpleLink™ Ultra-Low-Power Sub-1 GHz Wireless MCU*, Datasheet (SWRS181)
6. Texas Instruments, *CC13xx Combined wM-Bus C-Mode and T-Mode*, Application Report (SWRA522)
7. Texas Instruments, *CC13xx wM-Bus S-Mode*, Application Report (SWRA512)
8. Texas Instruments, *CC-Antenna-DK2 and Antenna Measurements Summary*, Application Report (SWRA496)
9. Radiocrafts, wM-Bus Products (https://radiocrafts.com/products/wirelessmbus/)

6.1 Trademarks

7 About the Author

MILEN STEFANOV (M.Sc.E.E) is a system engineer at TI, working in the field of Grid Infrastructure and an expert in RF communication technologies and sub-metering applications. After graduating, he spent 5 years as a research assistant at the Chemnitz University of Technology (TUC) and 3.5 years in the semiconductor industry in high-speed optical and wired communications as a system engineer. He joined TI in 2003 to become a Wi-Fi® expert and support TI's Wi-Fi products at major OEMs; since 2010, he has focused on metering and Sub-1 GHz RF solutions for the European Grid Infrastructure market. Milen Stefanov has published multiple articles on wM-Bus technology in Europe and presented technical papers at the Wireless Congress and Smart Home & Metering summits in Munich.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES