Application Report
SLRA004–March 2015

RF430F5978EVM Optimized Power Consumption

Claus Kuch and Dominik Gerl

ABSTRACT

The RF430F5978EVM bundle helps system developers evaluate the key features of the RF430F5978 MCU. The key features include a sub-1-GHz RF transceiver, a 3-D low-frequency (LF) wake-up and trigger function, passive battery-less transponder operation, and resonant trimming. A plug-in LF trigger module (MRD2EVM micro reader), an RF430F5978 MCU evaluation board (RF430F5978EVM), and an AP434R01 RF access point are included in the RF430F5978EVM kit.

A typical RF430F5978 application captures data from multiple sensors over a period of time, waits for an LF wake-up trigger from the MRD2EVM reader, and transmits the collected data to the access point using the sub-1-GHz RF interface. This data is then sent to the included host GUI to visualize the data and 3-D position of the RF430F5978.

This application note uses the RF430F5978EVM as an example system and describes methods to reduce power consumption. The information here can be used as a guideline to the most important steps to write energy-efficient code. In addition, this application note explains how to optimize the firmware for very low current consumption using EnergyTrace™ technology.

Contents

List of Figures

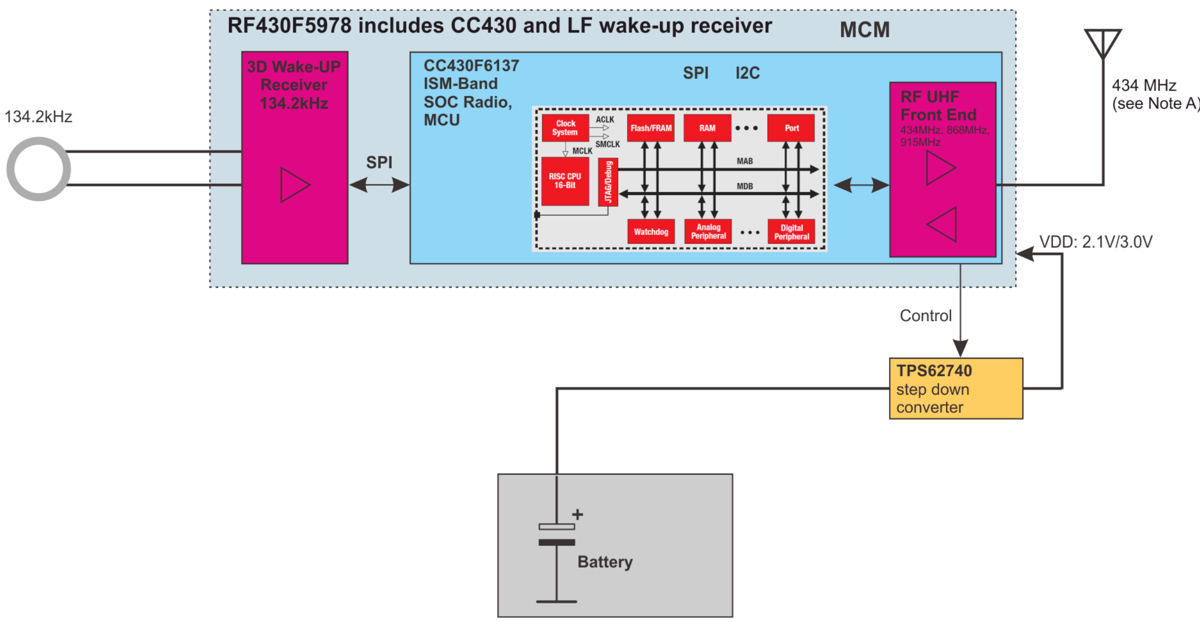
List of Tables

1 Introduction to the RF430F5978EVM

The Texas Instruments RF430F5978 system in package is built on the CC430F6137, which integrates a sub-1-GHz RF transceiver (CC1101) with an MSP430™ core, and extends its functionality by adding a 3-D LF wake-up trigger and transponder interface. The LF module works in transponder mode even without an active power supply. The 128-bit AES security encryption and decryption coprocessor adds advanced security for data protection. With its innovative wake-up and localization RF connectivity, this highly integrated SOC solution enables very small compact designs with small batteries.

The device architecture, combined with five low-power modes, is optimized to maximize battery life and to enable wireless connectivity in battery-powered applications. The MSP430 core offers performance, 16-bit RISC registers, and constant generators that contribute to maximum code efficiency. All of these features are supplemented by a variety of analog and digital peripherals.

The RF430F5978 EVM demonstrates this functionality and provides an easy-to-use development platform (see Figure 1). This bundle helps to evaluate the key features of the RF430F5978 MCU: sub-1-GHz RF transceiver, 3-D LF wake-up and trigger function, passive battery-less transponder operation, AES security, and resonant trimming.



A 434-MHz worldwide frequency depends on the RF430F5978EVM RF filter and the selected UHF frequency condition. With different hardware, the kit can use the 315-, 868-, or 915-MHz bands.

Figure 1. Block Diagram RF430F5978EVM

The bundle includes a USB plug-in LF trigger module (MRD2EVM micro reader), an RF430F5978 MCU evaluation board (RF430F5978EVM), an AP434R01 RF access point, and a 3-D LF antenna (see Figure 2).

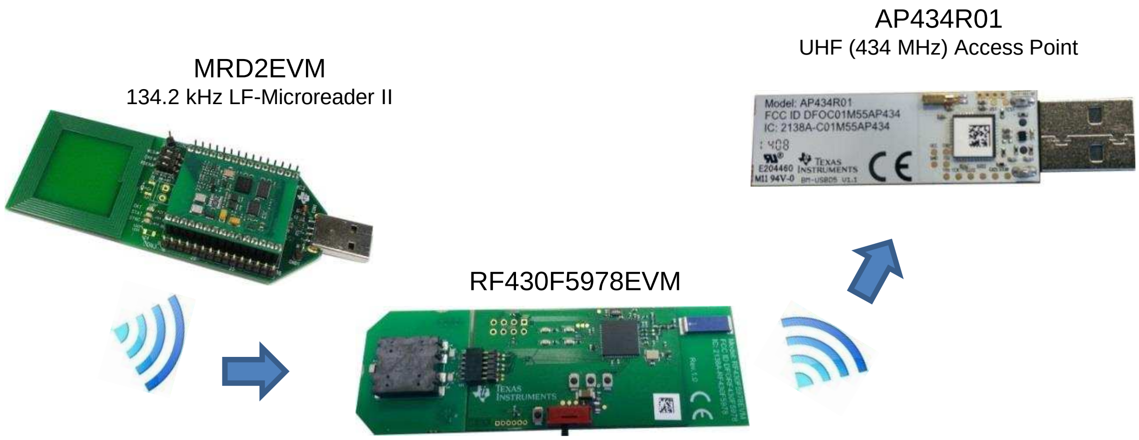


Figure 2. RF430F5978EVM SoC Evaluation Kit

This EVM bundle can be used to set up a typical application in which the RF430F5978 captures data from multiple sensors over a period of time, waits for an LF wake-up trigger from the reader, and transmits the collected data to the access point over the sub-1-GHz RF link. The access point sends this data to the included host GUI to show the data and position of the RF430F5978 (see Figure 3).

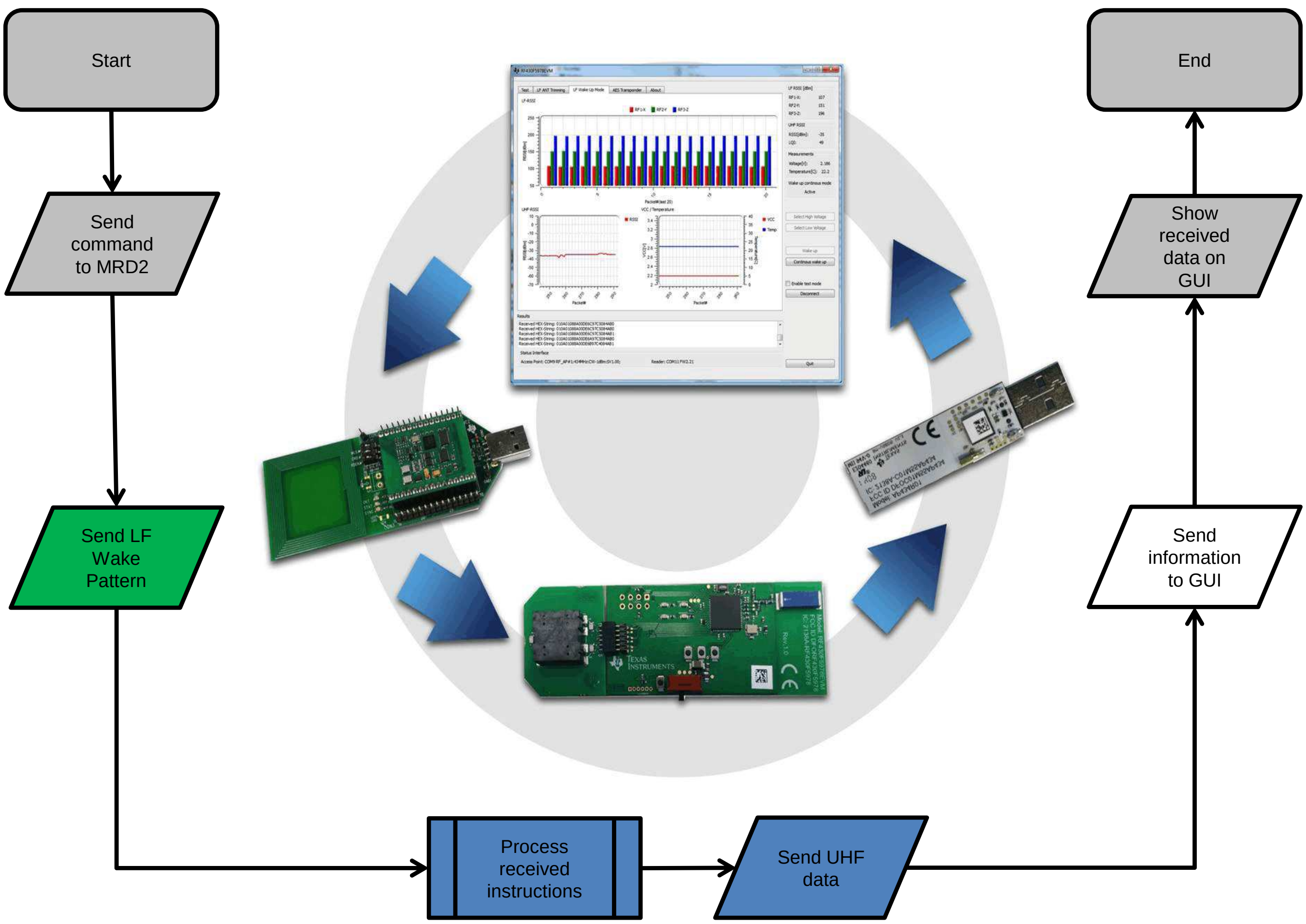


Figure 3. RF430F5978EVM Flowchart

For more detailed information, see the *RF430F5978EVM User's Guide* (SLRU007).

Before starting the evaluation, the EVM must be set up as described in the *RF430F5978EVM User's Guide*. When the RF430F5978 power switch is turned on, the EVM executes the code that is described in Section 2.

2 Code Explanation of the RF430F5978EVM

2.1 Module Initialization

The flowcharts and explanations in this section are based on RF430F5978EVM code version 1.10.0.0.

The startup process (see Figure 4) initializes the radio core, the GPIO ports, the ADC, the internal SPI bus, and the external oscillator. After the initialization process, the RF430F5978 enters LPM4 to save power (standby current is approximately 7 µA).

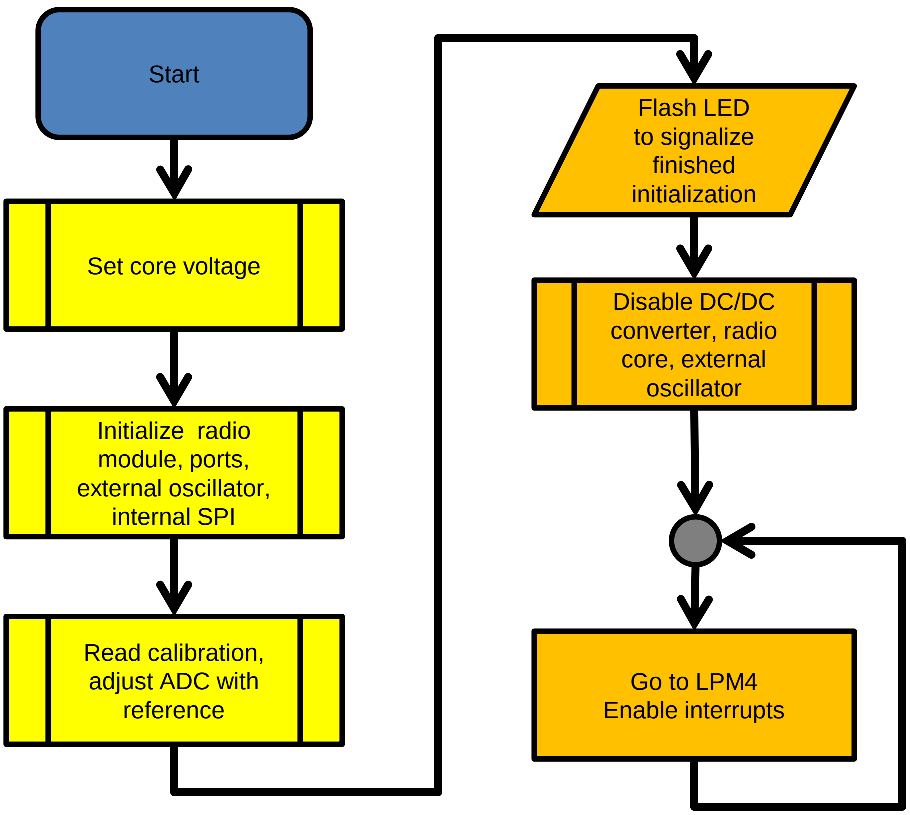


Figure 4. RF430F5978EVM Main Function Flowchart

2.2 LF Wake-up Interrupt

For LF wakeup, the MRD2EVM is controlled by the PC software "RF430F5978EVM". This application has various functions including reading the RF430F5978 as a passive LF transponder and sending wake-up signals. In a typical application, the application sends 64-bit wake patterns. When the RF430F5978EVM board is in range (typically 2 m but can be extended up to 10 m with the LF high-power transmitter RI-RFM-007B), it detects the LF trigger signal and activates the microcontroller using an external interrupt (see Figure 5).

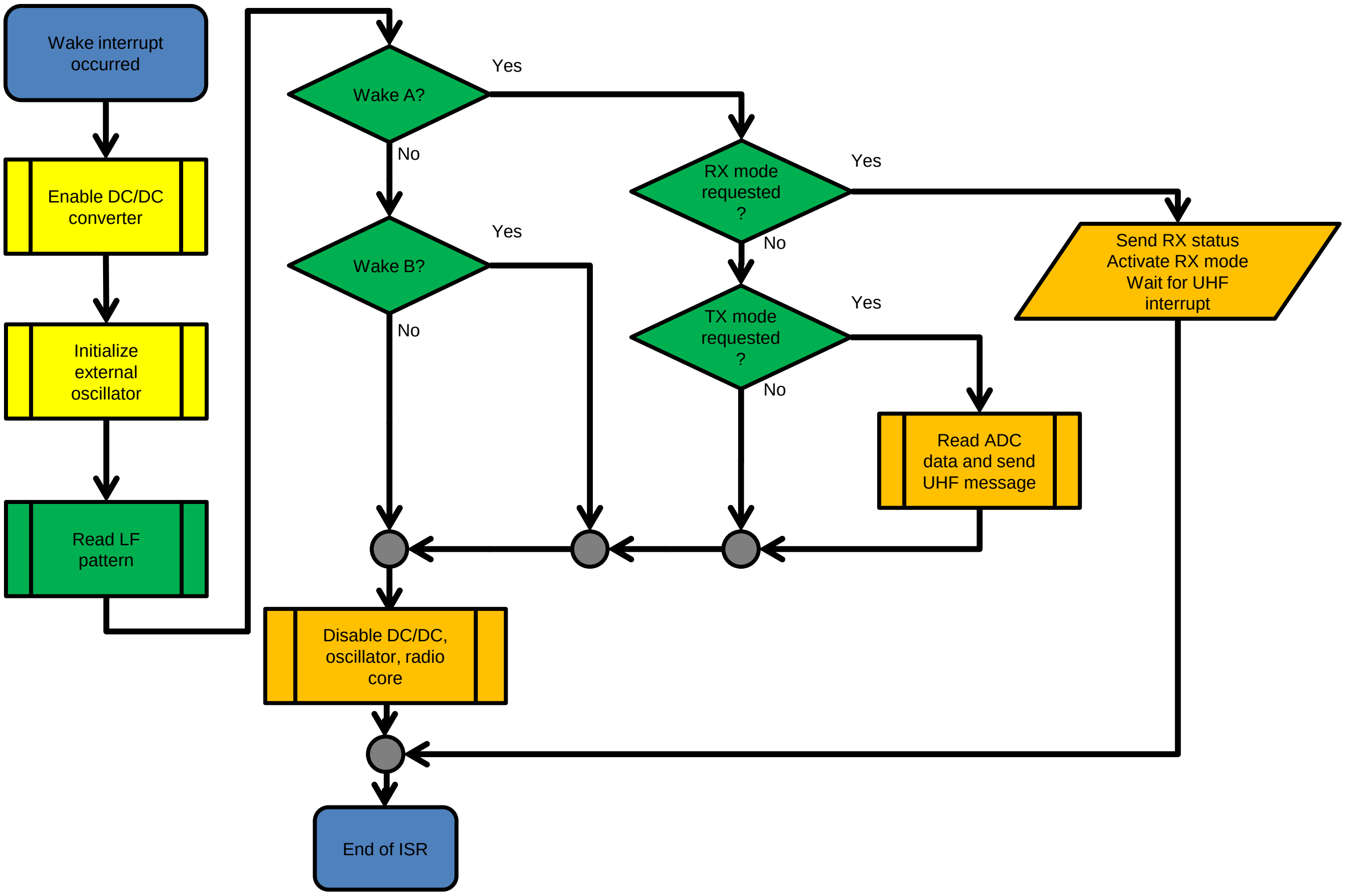


Figure 5. RF430F5978EVM LF ISR Flowchart

The module is in low-power mode most of the time. When a wake-up signal is detected, the CPU wakes from LPM and powers up the oscillator and the radio core.

	Device Status								PE Status								SWITCHES								PE_DATA							
	3								2								1								0							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Status Read	LVL_B_DET			IMMO_ACTIVE	MSP_ACCESS	NEW_SWITCH	WAKE_B	WAKE_A	PE_BIT_CNT				EOF_DET	DATA_PENDING	ERROR_DET	START_DET	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0	RECEIVED BITS (Wake Receiver)							

Figure 6. LF SPI Data

The 64-bit wake signal that is sent to the RF430F5978 includes the type of wake pattern, which is stored in the Device Status byte (see Figure 6). Optional user defined information given by the wake signal can be stored in the Switches byte.

These bytes are read from the LF front end and sent to the controller through the internal SPI connection. For details, see the "Low-Frequency Wake-Up Receiver" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

RX Mode

When the EVM enters RX mode, it sends a short status message to the access point and then stays on and waits for further instructions over the UHF interface. Received commands are processed by the parser (see Section 2.3).

TX Mode

The typical process for this mode is for the RF430F5978 to collect data (for example, temperature measurements) and send the data to the UHF access point.

Table 1. UHF Data Format(1)

Content	Start	Length	Command	Voltage (mV)	Temper-ature (ºC)	LF RSSI X (dBm)	LF RSSI Y (dBm)	LF RSSI Z (dBm)	LF Wake	UHF RSSI	UHF LQI	EOL
Example:	01	0A	01	087E	00EA	A3	AD	CA	08	24	B0	0D
Explanation:	Start of telegram	Length without Start, Length, Command, and EOL bytes	Response always null byte	Voltage: 0x087E = 2174 mV	Temperature: 0x00EA = 23.4ºC	LF RSSI X: 0xA3 = 163 dBm	LF RSSI Y: 0xAD = 173 dBm	LF RSSI Z: 0xCA = 202 dBm	Which wakeup occurred: 8 = Wake A	UHF RSSI: 0x24 = -54 dBm	UHF LQI EST: 0xB0 and 0x7F = 48	EOL end of messag e
Size:	1 byte	1 byte	1 byte	2 byte	2 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1 byte	1byte

(1) For more information, see the *RF430F5978EVM User's Guide* (SLRU007).

When the module is in RX mode, the oscillators, DC/DC converter, and radio core need to stay ready for incoming messages from the UHF access point.

```
if(OPER != 0x01)       // do not shut down XT2 when RX Mode is enabled
{
  Strobe(RF_SIDLE);    // set UHF transceiver in idle state
  Strobe(RF_SPWD);     // switch uhf transceiver off
  XT2_OFF();           // disable XT2 to save energy in LPM Mode
  BYPASS_ON;           // disable TPS62730 for LPM Mode
}
else                   // missing uhf signal in RX mode avoids shutting down xt2
{
  Timer1_Init();       // init timer for missing uhf timeout
  Timer1_Start();      // start timer
}
```

A timeout mechanism is included to prevent the RF430F5978 from remaining in active mode for a long time. Otherwise, if the RF430F5978 were to miss messages from the UHF transceiver while in RX mode, it would stay in active mode with active oscillator and radio core. This would cause high current consumption. To prevent this, a simple timer works as timeout detection. When RX mode is enabled, the timer is started and is reset inside the UHF interrupt service routine.

The timer is sourced by ACLK, which is turned off in LPM4. As long as XT2 is enabled, the MCU does not go into LPM4, and ACLK stays active. The timer works in continuous mode and overflows after 1 second.

In Figure 7, the first peak represents a working IDN command. For the second peak, the AP434R01 UHF access point has been unplugged from the computer. The RF430F5978 stays active while it is waiting for the UHF interrupt. After 1 second, it transitions into LPM4 due to the timeout detection.

A timeout function could also be implemented using the internal real-time clock.

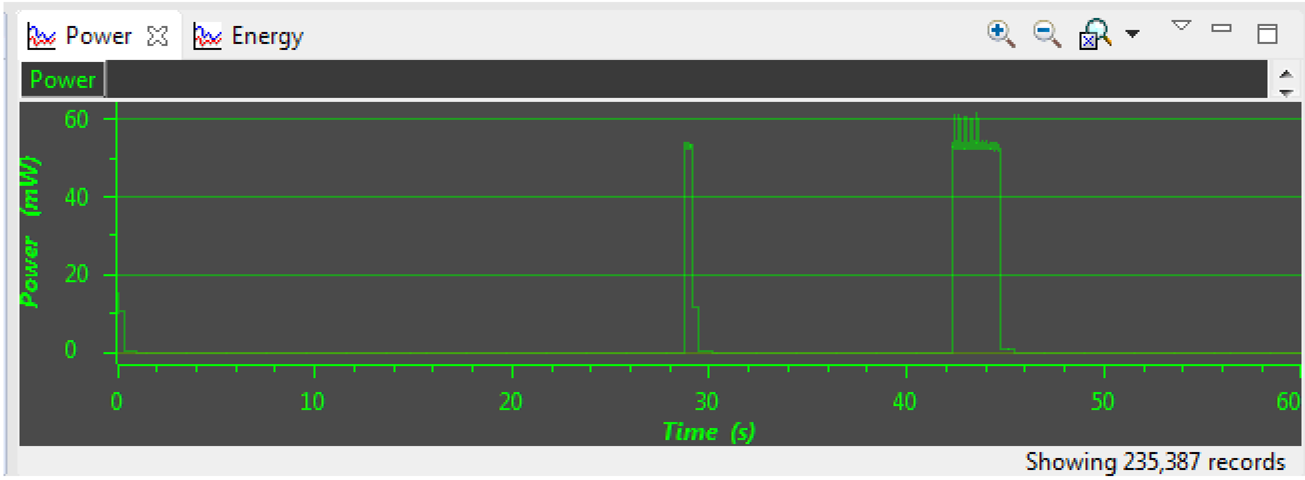


Figure 7. UHF Timeout

2.3 UHF Interrupt

Module control is performed over the UHF link. This includes switching the voltage regulator on or off, performing antenna trims, and many other functions. To achieve an effective and simple program architecture for control, a parser is used in the RF430F5978 example code. This parser is called when a UHF interrupt occurs and the end of a packet is received (see Figure 8).

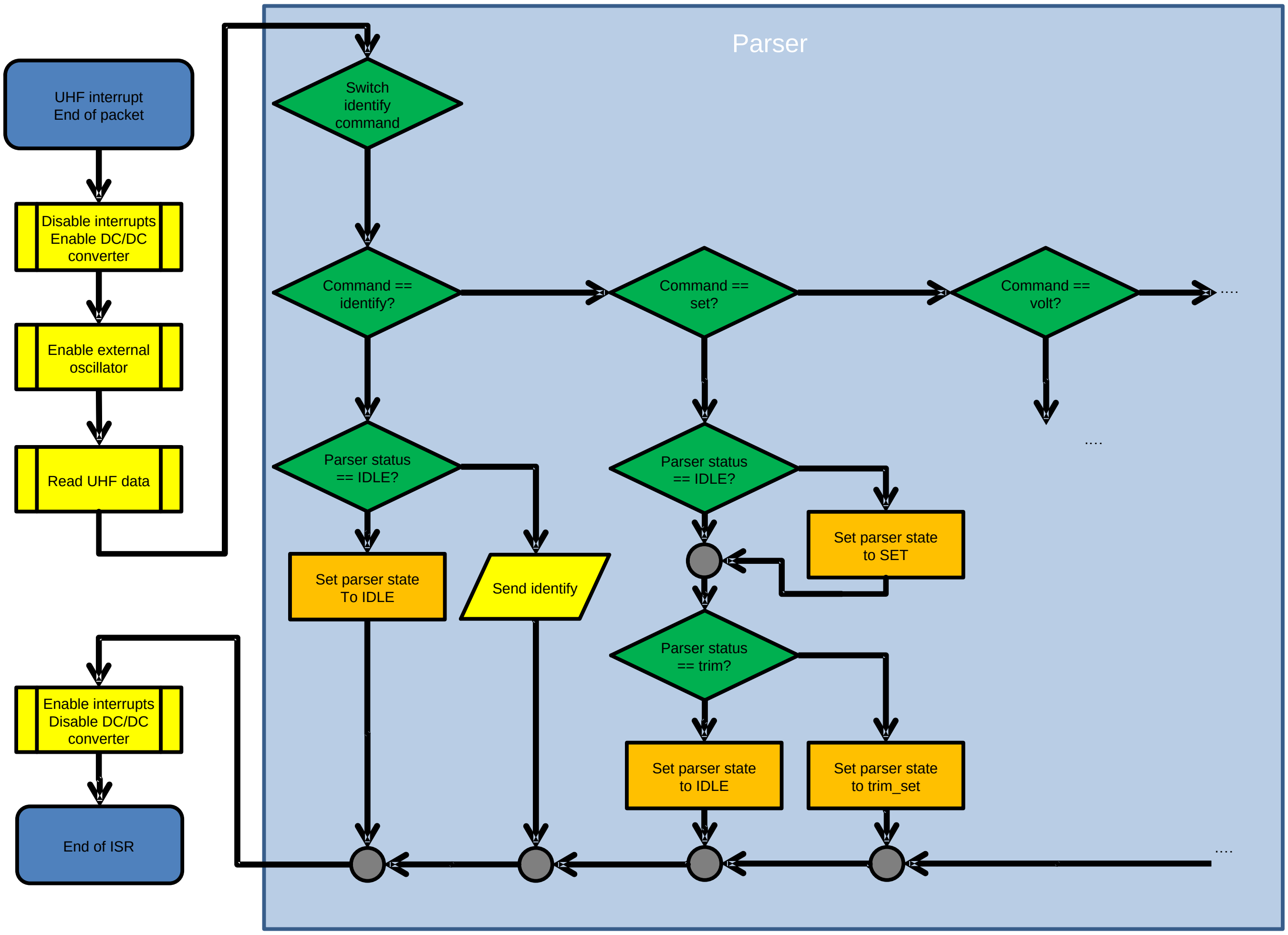


Figure 8. RF430F5978EVM UHF ISR Flowchart

The next task for the module is controlled by the current state of the parser combined with the received message. For example, to set the voltage high, the parser must be in the initial condition (IDLE). A sequence of "set" and then "volt" commands is required to first enter the STATE_SET and then enter the STATE_SET_VOLT condition. Sending the "high" command next executes the necessary code to switch to high voltage. If an invalid command is received when the current sequence of commands has not been finished, the parser returns to the initial IDLE condition (see Figure 9).

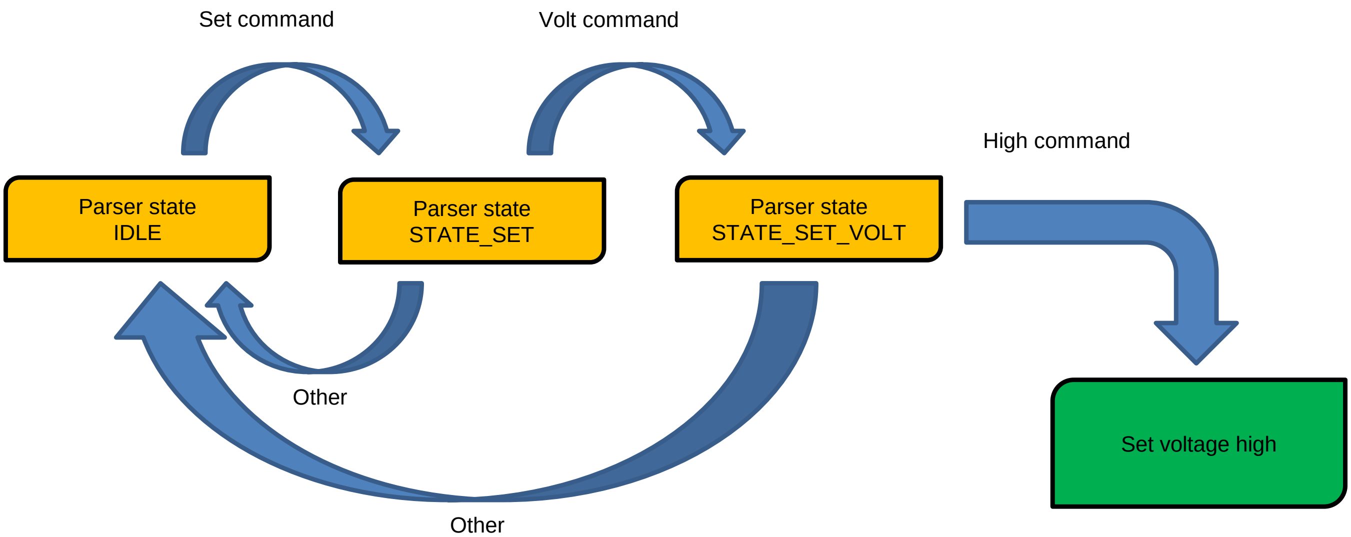


Figure 9. Parser Example Flowchart

As shown in the following code example, many of the commands depend on the current parser state. This extends the variety of usable processes with a limited number of commands. This parser architecture is designed for both high code clarity and high processing speed.

Code Example

```
for(;;)                          // Parser loop.
{
    unsigned int cmdID;
    switch(cmdID = GetCmdID())   // Identify command received.
    {
        case SET:                // Trim set command received.
        {
            if(parserState == STATE_IDLE)
            {
                parserState = STATE_SET;
            }
            else if(parserState == STATE_TRIM)
            {
                parserState = STATE_TRIM_SET;
            }
            else
            {
                parserState = STATE_IDLE;
            }
            break;
        }
        case VOLT:                // Set Voltage command received.
        {
            if(parserState == STATE_SET)
            {
                parserState = STATE_SET_VOLT;
            }
            else
            {
                parserState = STATE_IDLE;
            }
            break;
        }
        case HIGH:                // Switch to high voltage.
        {
            if(parserState == STATE_SET_VOLT)
            {
```

```
                BYPASS_ON;        // disable TPS62730
                voltage = VOLTAGE_HIGH;
            }
            parserState = STATE_IDLE;
            break;
        }
        case LOW:                 // Switch to low voltage.
        {
            if(parserState == STATE_SET_VOLT)
            {
                BYPASS_OFF;       // enable TPS6273
                voltage = VOLTAGE_LOW;
            }
            parserState = STATE_IDLE;
            break;
        }
        ....................      // further program code
    }
}
```

3 Steps To Decrease Power Consumption

3.1 GPIO Ports

The RF430 device provides six GPIOs that are mapped to external port pins. Unused pins that are improperly set result in high leakage currents. To avoid these leakage currents and decrease power consumption, set unused pins as output and high state. For details, see the "Digital I/O" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

```
P2OUT = 0xFF;       // set high to save current
P2DIR = 0xFF;       // set port to output
```

Improperly configured internal pullup or pulldown resistors also lead to increased power consumption. Make sure to configure them according to the external wiring of the pin. If a pullup or pulldown is not required, disable it.

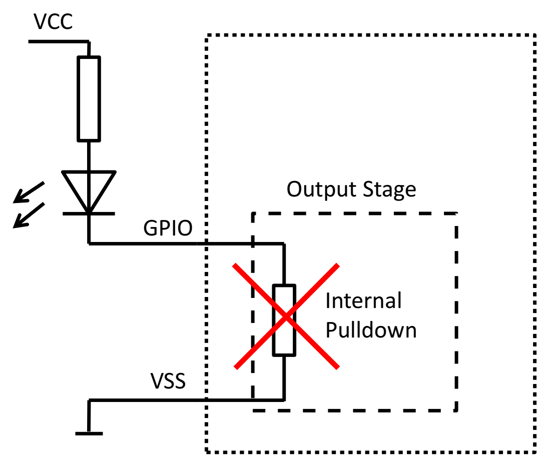


Figure 10. Pulldown Example

3.2 PMM and V_{core}

Higher clock frequencies require higher core voltage, but power consumption and power losses rise with the core voltage squared, as shown in Equation 1.

$$P = \frac{V_{core}^{2}}{R} \tag{1}$$

To minimize power consumption, the core voltage should set as high as necessary to support the selected frequency but no higher. For correct selection of V_{core}, see the "Power Management Module and Supply Voltage Supervisor" chapter of the *RF430F59xx Family User's Guide* (SLAU378). The example code includes an example of how to properly set the V_{core}.

The lowest possible V_{core} setting for the RF430F5978EVM is mode 2. A lower V_{core} leads to a power-on reset (POR), which permanently resets the device. A fault of the V_{core} can be detected by the low-side supply voltage monitor (SVML) and the low-side supply voltage supervisor (SVSL), which can be turned off if they are not needed.

3.3 Oscillators

For radio functionality, an external high-frequency oscillator (XT2) must be provided on the RF_XIN and RF_XOUT pins. This oscillator also supplies the MSP430 core through an internal clock connection.

When the radio core is disabled, the high-frequency oscillator also should be disabled, and the MSP430 core must be supplied by a lower-frequency oscillator, such as the DCO or the REFOCLK. For details of clock settings, see the "Unified Clock System (UCS)" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

The MSP core can be supplied by five different clock sources:

- XT1CLK: external low-frequency oscillator
- VLOCLK: internal very-low-power oscillator (10 kHz)
- REFOCLK: internal low-frequency oscillator (32768 Hz)
- DCOCLK: internal digitally controlled oscillator
- XT2CLK: external radio oscillator, essential for radio core functionality

These clock sources supply the three system clocks:

- ACLK: auxiliary clock
- MCLK: master clock
- SMCLK: subsystem master clock

For details on oscillator configuration, see Figure 11 and the block diagram and description in the "Unified Clock System (UCS)" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

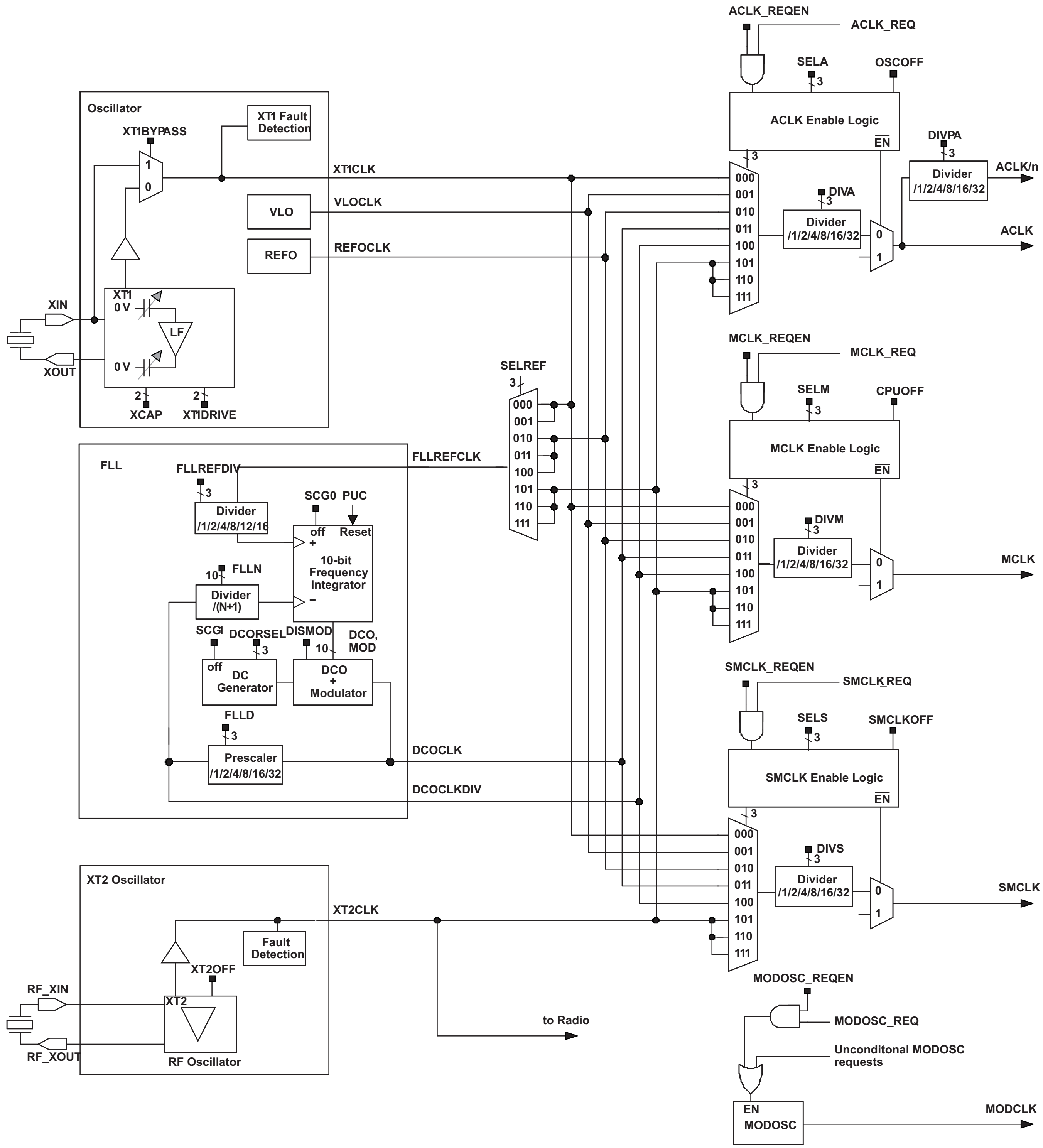


Figure 11. RF430F5978 Unified Clock System

In the example code, MCLK and SMCLK are supplied by XT2CLK divided by two when XT2CLK is active. When XT2 is deactivated, MCLK and SMCLK are switched to the DCOCLKDIV, which is the stabilized oscillation from the FLL module. The FLL reference is sourced by the low-current REFOCLK clock.

ACLK is directly supplied by REFOCLK. It should not be supplied from DCOCLKDIV, because the FLL is deactivated in LPM1 and higher, while REFOCLK remains active in LPM0 to LPM3. Therefore REFOCLK can supply ACLK for any timers and counters that remain active in these low-power modes.

```
void InitXT2Osc(void)                          // init oscillator with external 26MHz
{
  UCSCTL6 &= ~XT2OFF;                          // Enable XT2
  UCSCTL3 |= SELREF_2;                         // FLLref = REFO
  UCSCTL5 = DIVM__2+ DIVS__2;                  // SMCLK and MCLK = XT2/2
  UCSCTL4 |= SELA__REFOCLK + SELS__XT2CLK      // ACLK = REFOCLK
            + SELM__XT2CLK;                    // SMCLK, MCLK = XT2CLK when available,
                                               // otherwise DCOCLKDIV
  do                                           // Loop until XT2, XT1 & DCO stabilizes
  {
    UCSCTL7 &= ~(XT2OFFG + XT1LFOFFG + DCOFFG);  // Clear XT2,DCO fault flags
    SFRIFG1 &= ~OFIFG;                           // Clear fault flags
  }while (SFRIFG1&OFIFG);                        // Test oscillator fault flag
}
```

Regardless of the configuration shown above, all oscillators are deactivated in LPM4. Before entering low-power modes, the external oscillator should be switched off. To deactivate XT2 for low-power mode, the XT2OFF bit in the UCSCTL6 register must be set.

```
void XT2_OFF()
{
     UCSCTL6 |= XT2OFF;    // disable XT2
}
```

Because XT2 is required by the radio module, XT2 cannot be deactivated when it is needed by the radio core.

3.4 Radio Module

The RF430 in UHF transmit or receive mode consumes 15 mA to 35 mA, depending on the data size and the RF output power. The radio module should be activated only when actively sending or receiving messages.

Before the radio core can enter sleep mode, it must transition into IDLE mode first. IDLE mode is the initial condition of the radio module. For details, see the radio-control state diagram in the "CC1101-Based Radio Module (RF1A)" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

```
  Strobe(RF_SIDLE);   // set UHF transceiver in idle state
  Strobe(RF_SPWD);    // switch uhf transceiver off
  XT2_OFF();          // disable XT2 to save energy in LPM
```

The RF_SPWD command should be used instead of the RF_SXOFF command. RF_SPWD completely disables the radio core, and RF_SXOFF only turns off the radio crystal oscillator path. For details, see the radio core instruction set in the "CC1101-Based Radio Module (RF1A)" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

After the radio core is turned off, the XT2 oscillator can also be disabled as shown in Section 3.3.

3.5 Low-Power Modes

The RF430F5978 offers five low-power modes. The amount of saved energy increases from LPM0 to LPM4. As described in the operating modes section in the "System Resets, Interrupts, and Operating Modes, System Control Module (SYS)" chapter of the *RF430F59xx Family User's Guide* (SLAU378), different features and functions are enabled or disabled in each mode, and the process to return to active mode can also vary.

In the RF430F5978EVM, LPM4 is used, because it offers high power savings. LPM4 not only deactivates the CPU, it also deactivates all oscillators. Even when the device is in LPM4, an external interrupt from the LF receiver can wake it.

There are two additional modes (LPMx.5) which would result in even higher power savings. These modes deactivate the V_{core}, so all memory content is lost, and the MCU must be reinitialized on every power up. The LPMx.5 modes should not be used on the RF430F5978 devices (see the CC430F6137 errata SLAZ102).

To enter LPM4 and enable interrupts, a preprocessor directive can be used inside the main function.

```
while (1)
{
    __bis_SR_register(LPM4_bits + GIE);     // Enter LPM4, enable interrupts
    __no_operation();
}
```

Entering a low-power mode can also be done manually by setting the CPUOFF, OSCOFF, SCG0, and SCG1 bits in the status register. Before entering a low-power mode, make sure to disable unused peripheral devices such as timers or the USCI interface. For details, see the "Power Management Module and Supply Voltage Supervisor" chapter of the *RF430F59xx Family User's Guide* (SLAU378).

3.6 TPS62730 Buck Converter

The RF430F5978 supports operation over a wide supply voltage range from 1.8 V to 3.6 V for the microcontroller core and from 2.0 V to 3.6 V for radio operation. Internally, this supply is typically regulated down to 2.0 V (in RF mode) using LDOs.

The RF430F5978 can be powered directly from a battery (see Figure 12). However, the battery load is high in active transmit or receive mode. A high load current causes a voltage drop due to the voltage divider created by the internal battery cell impedance and the load impedance of the RF430F5978.

The expected voltage drop depends on the internal impedance of the battery and the current drawn by the radio core. Because of the high peak currents, the effective battery life decreases, and the system runs the risk of an unstable voltage supply to the VCC pins. This voltage drop can negatively affect the performance of the RF430F5978 system. Small button cell batteries, which are usually preferred for such applications, have high internal impedance, which intensifies this negative effect.

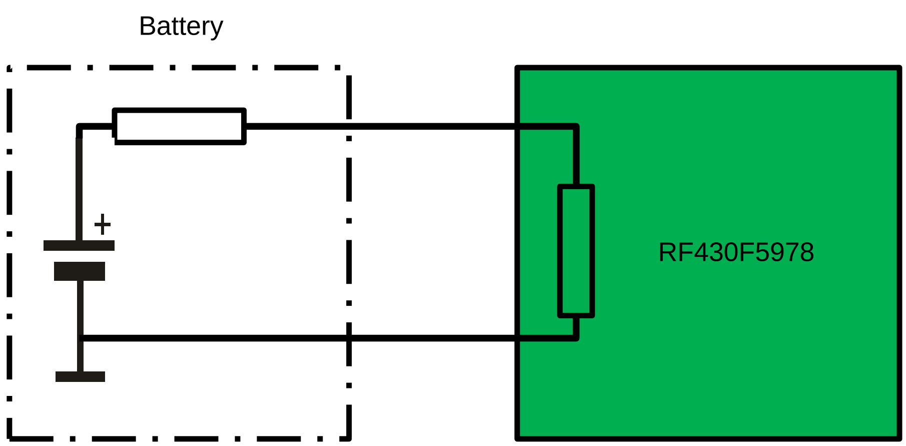


Figure 12. RF430F5978 Battery Supply

One remedy to this problem is to use an external ultra-low-power voltage regulator. Figure 13 shows a proposal that uses the TPS62730 step-down converter between the battery and the RF430F5978.

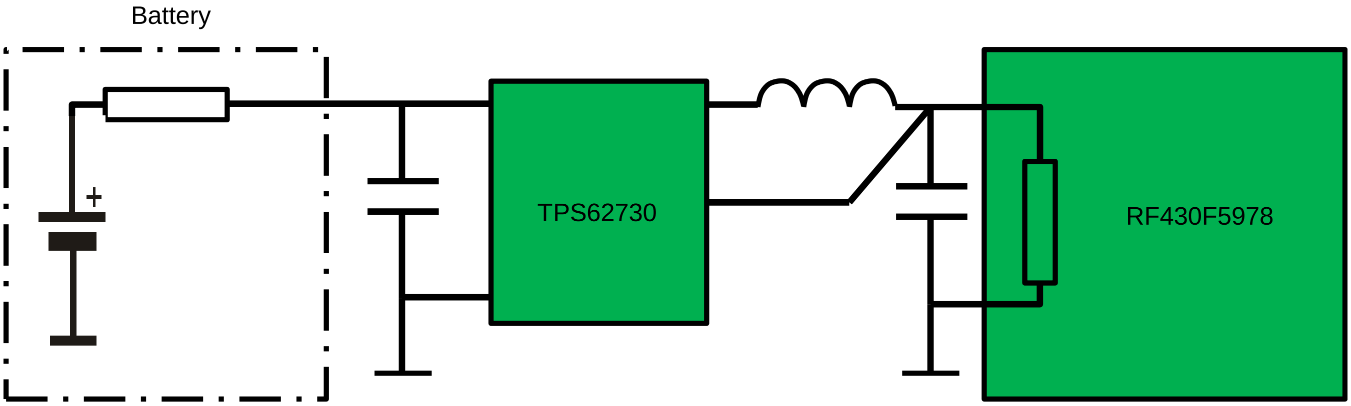


Figure 13. RF430F5978 TPS62730 Supply

The TPS62730 step-down converter is designed for efficiency and simplicity. For its intended use, only three external components are required, as shown in Figure 13. This not only saves costs, it also results in a simple PCB layout. Due to the high switching frequency (up to 3 MHz) of the converter, the external circuit requires low inductance and capacitance. This high frequency enables low output ripple voltage and low noise even with a small 2.2-µF output capacitor.

The converter automatically enters bypass mode when the battery voltage falls below the automatic bypass switch transition threshold. The automatic transition into bypass mode during DC/DC operation prevents an increase of output ripple voltage and noise when the DC/DC converter operates close to 100% duty cycle.

The TPS62730 has two pins (ON/BYP and STAT) that can be controlled or evaluated by the RF430F5978.

The STAT pin is high impedance when the TPS62730 is in ultra-low-power bypass mode. The STAT pin is active low when the TPS62730 is operating in DC/DC mode.

When the ON/BYP is externally set low, the TPS62730 goes into ultra-low-power bypass mode. The battery voltage (supplied to the VIN pin of the regulator) is directly fed to the VOUT pin of the regulator. In bypass mode, the current consumption of the TPS62730 is lowered to 30 nA. When the ON/BYP is externally set high, the TPS62730 becomes active.

3.6.1 Implementation

In the configuration described here, the TPS62730 acts as a preregulator before the internal LDOs on the RF430F5978. With this improved power management circuit, the RF430F5978 always sees a stable supply voltage of 2.1 V, regardless of the battery voltage. This not only increases the battery lifetime significantly by reducing current consumption up to 30% or more during radio transmit or receive modes, but it also ensures a stable and more reliable operation of the system.

As previously mentioned, the RF430F5978 is designed to be powered directly from a battery. Internal to the RF430F5978, this supply is regulated down to 2.0 V (RF mode) using LDOs. Note that the input current to and output current from the LDO are always the same. The efficiency is determined by the ratio between output and input voltage (see Figure 14). This means that if the supply voltage from the battery is higher (up to the 3.6-V maximum supported by the RF430F5978), efficiency is lower and some energy is lost in the regulation by the LDOs.

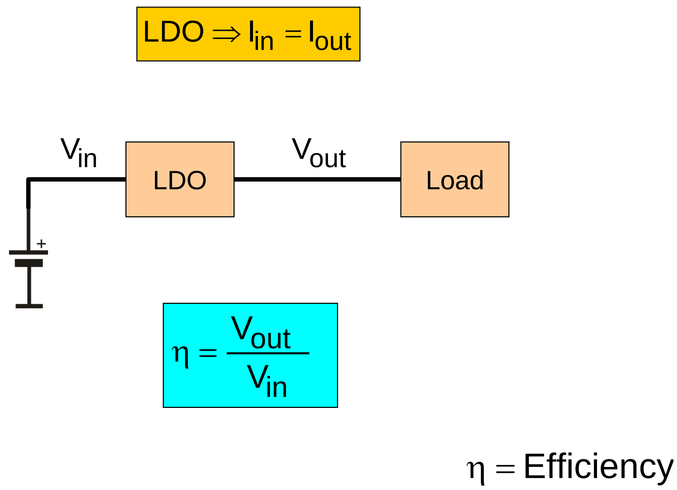


Figure 14. LDO Scheme

3.6.2 Power Reduction Using a Step-Down Converter

By using the DC/DC step-down converter between the battery and the RF430F5978, the supply voltage to the RF430F5978 is always regulated to approximately 2.1 V. This increases the overall efficiency and robustness of the system. The key point to note is that, unlike with the LDO, the input power to the DC/DC converter is equal to the output power (and not the current). Therefore, efficiency is determined by the ratio between output and input power (see Figure 15).

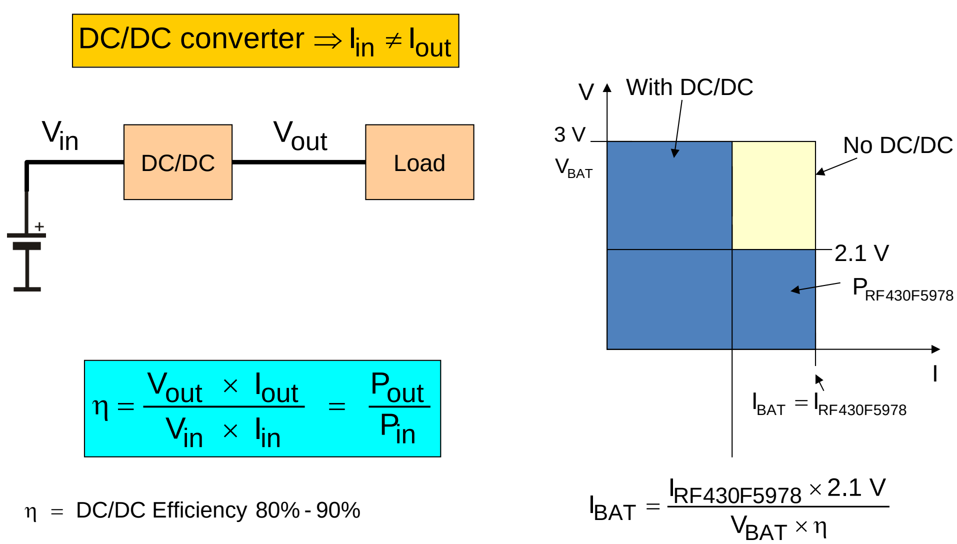


Figure 15. DC/DC Converter Scheme

The voltage before the internal LDOs is always regulated to this voltage level, which results in minimal losses (see Figure 16).

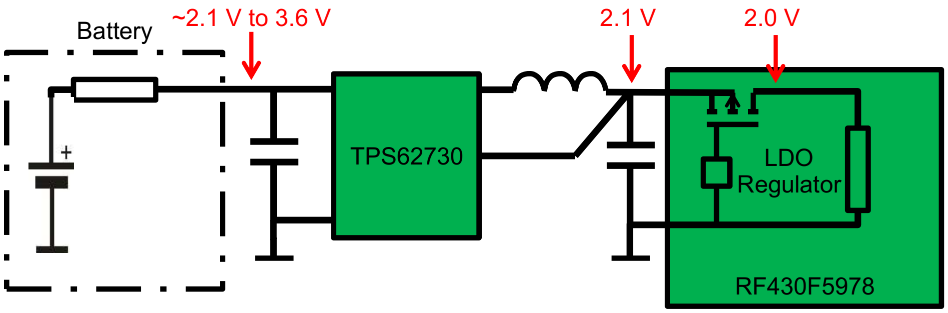


Figure 16. RF430F5978 TPS62730 Supply With Voltage Levels

3.6.3 Benefits of Using the TPS62730

- Stabilized voltage and fewer losses on internal LDO of the RF430F5978
- Lower peak currents due to lower supply voltage
- Easy layout
- Cost effective solution
- Higher efficiency, which leads to an increased battery lifetime
- Fewer stresses to battery
- Regulator is controllable by the RF430F5978 device

3.6.4 Code Example

To achieve maximum efficiency, the TPS62730 should be activated as soon as the RF430F5978 is in active mode. Before the RF430F5978 enters LPM4, the TPS62730 should be set to bypass mode. In LPM4, the internal losses of the LDO are negligible. In this case, it is much more important to reduce the energy consumption of the TPS62730 converter.

```
#pragma vector=PORT1_VECTOR
__interrupt void Port_1(void)
{
    __disable_interrupt();
    BYPASS_OFF;             // enable TPS62730 to power device with 2.1V

    ..............          // ISR actions
    BYPASS_ON;              // disable TPS62730 for LPM Mode
    __enable_interrupt();
}
```

The functions BYPASS_ON and BYPASS_OFF only set or clear the output pin on the RF430F5978 device that drives the ON/BYP pin of the TPS62730.

4 Energy Measurements

4.1 Hardware Setup

The following measurements were taken with EnergyTrace™ technology. EnergyTrace technology is a software function available in the Code Composer Studio™ (CCS) IDE version 6 or newer and the latest version of the IAR Embedded Workbench™ IDE. For more information on EnergyTrace technology, see the *Code Composer Studio for MSP430 User's Guide* (SLAU157) or the *IAR Embedded Workbench Version 3+ for MSP430 User's Guide* (SLAU138).

EnergyTrace technology can measure the power and energy consumption of MSP430-based modules using only the JTAG debug interface. It does not need any other external equipment. Systems other than some MSP430 evaluation boards with an integrated EZ-FET flash tool require an external MSP-FET JTAG emulator with EnergyTrace technology support to flash the code and monitor the power consumption.

To use EnergyTrace technology, the RF430F5978EVM must be connected to the MSP-FET, as shown in Figure 17.

Figure 17. RF430F5978EVM Connection to MSP-FET

The values shown in Figure 18 were measured with CCS version 6.0.1.00040 and a TI MSP-FET flash emulation tool. The upper window shows a summary of the measured values including minimum, maximum, and average current and power. The lower window shows a graph that traces power consumption. The profile shown here was measured with a voltage of 3.0 V supplied by the JTAG emulator to the RF430F5978 module (TX power: –20 dBm).

The first peak shows the module going into RX mode. After a wakeup, the UHF transmission starts. The RF430F5978 module received an IDN command and returned its IDN string (board name and software version) to the access point.

The second peak represents a TX command that takes much less time. The RF430F5978 wakes from LPM4, measures its input voltage and temperature, and sends this information to the access point.

NOTE: When the MSP-FET supplies the RF430F5978EVM over the 4-wire debug interface, the voltage regulator on the EVM is bypassed and the module is directly supplied by the JTAG emulator. Therefore, the power consumption of the voltage regulator is not included in this data.

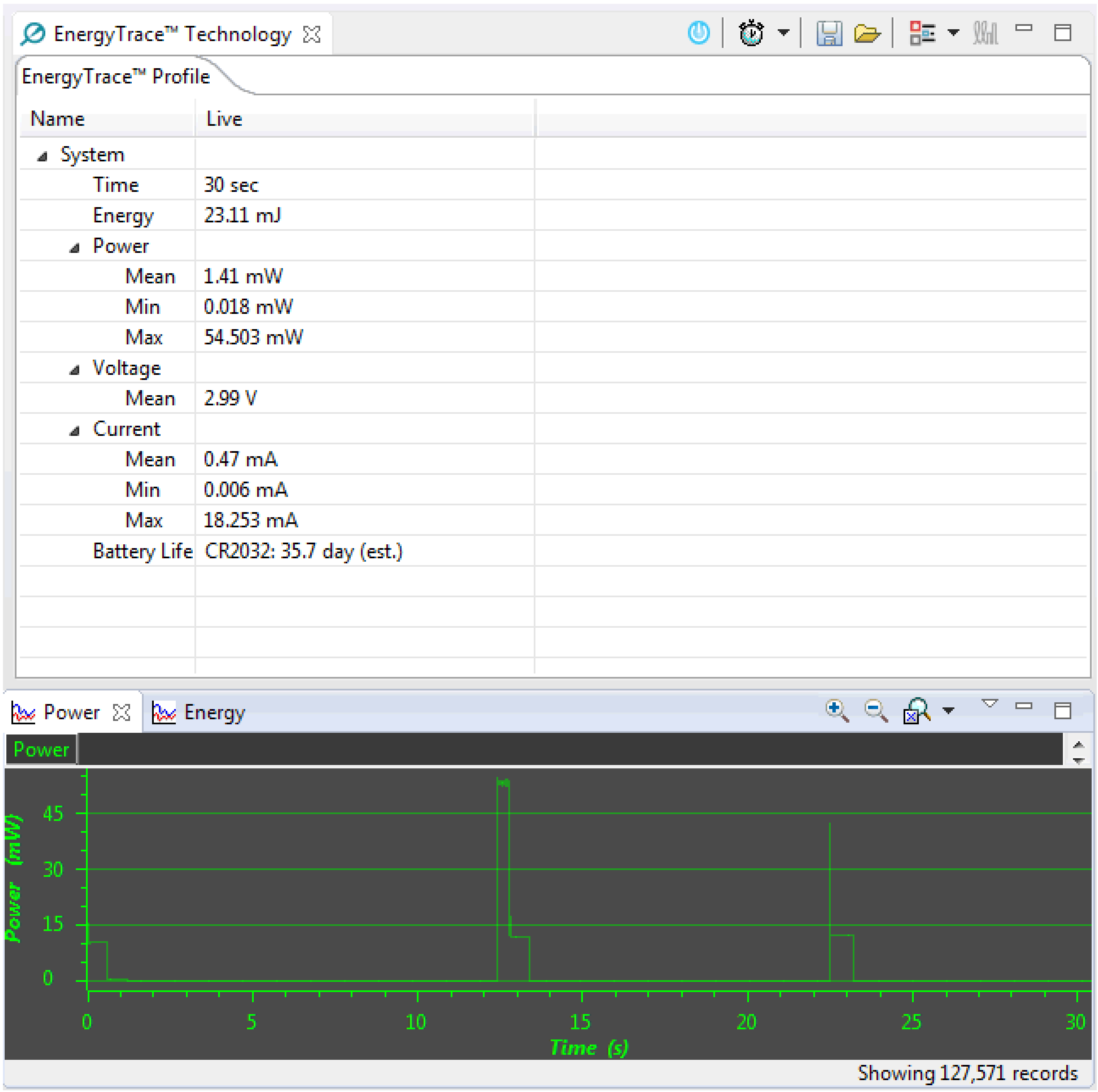


Figure 18. EnergyTrace™ Profile in RX Mode and TX Mode

The power consumption can be measured in both debug mode and free run mode. The free run mode disables the control of the RF430F5978 device by deactivating the Spy-Bi-Wire connection. Due to this, the power consumption by Spy-Bi-Wire pins are excluded from the measurement. In debug more, a higher current (approximately 130 µA) is measured by the tool because of the Spy-Bi-Wire connection.

5 Summary

This application note is an introduction to how to create energy-efficient code and what tools can be used to measure power consumption. As the number of devices in use rises, the use of low-power devices becomes more important. As can be seen in the following sections, the RF430F5978 fulfills this demand and makes it possible to achieve a long lifetime in battery-powered devices.

5.1 Measured Values

In Figure 19, the RF430F5978EVM module stays in LPM4 and bypasses the regulator, which results in a current consumption of approximately 7 µA. During continuous mode with one wake signal received every minute, which represents typical operation, a battery lifetime of over 900 days (estimated by EnergyTrace technology measurement) can be achieved. (Battery type: CR2032, transmission power: –20 dBm, V_{CC}: 3 V, measured with active regulator.)

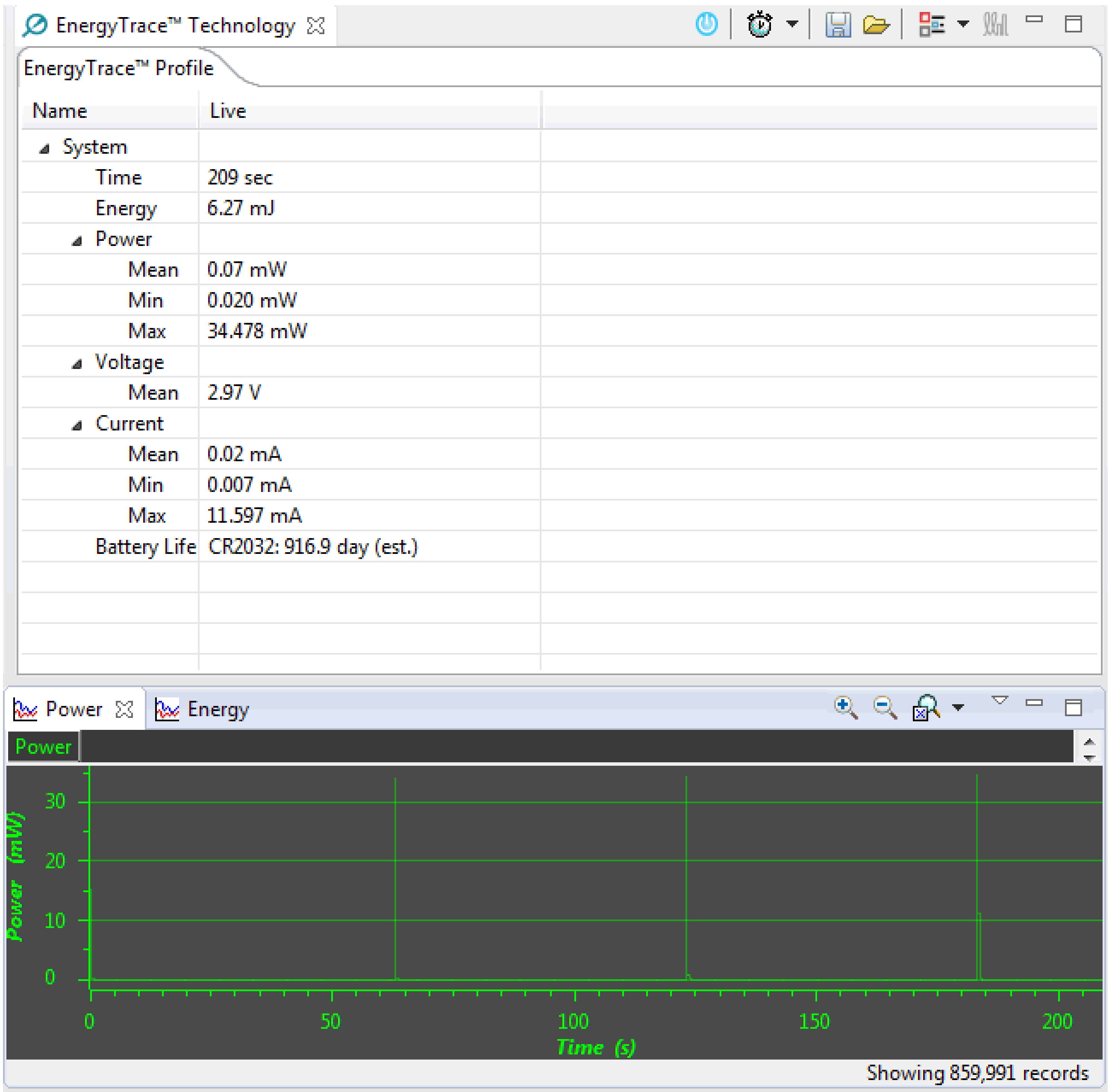


Figure 19. EnergyTrace™ Profile With Continuous Measurement

Working with ultra-low-power circuits often needs complex measurement equipment to achieve the necessary resolution. This issue can be eliminated by using EnergyTrace technology. Because EnergyTrace technology combines all of the essential functions for power measurement during code development, it represents an enormous help in the process of developing low-power circuits.

5.2 Highlights

- Module standby power consumption is approximately 20 mW.
- Average power consumption is approximately 70 mW.
- The battery (CR2032) lifetime with new firmware is approximately 2.5 years with one wake signal every minute and transmit power of –20 dBm.

6 References

1. *RF430F5978EVM User's Guide* (SLRU007)
2. *RF430F59xx Family User's Guide* (SLAU378)
3. *RF430F5978 MSP430 System-in-Package With Sub 1-GHz Transceiver* (SLAS740)
4. *TPS62730 Step Down Converter With Bypass Mode for Ultra Low Power Wireless Applications* (SLVSAC3)
5. EnergyTrace™ Technology (http://www.ti.com/tool/energytrace)
6. ULP Advisor TI Wiki (http://processors.wiki.ti.com/index.php/ULP_Advisor)