
Smart IoT Wireless Sensor Node using XMEGA® AU MCU and LoRa Technology

Introduction

“When we talk about the Internet of Things, it’s all about embedding intelligence, so things become smarter and do more than they were proposed to do.” – Nicholas Negroponte

Internet of Things (IoT) has become very popular in today’s embedded world. The Internet of Things is a confluence of powerful and cost-effective microcontrollers, efficient wireless protocols, and enriched sensors. The IoT network comprises embedded technology to sense and communicate with their internal states or the external cloud environment.

The IoT applications demand lower power consumption, secure communication, and extended wireless communication range of connected devices. This enables the IoT systems to cover a wider area while being very power efficient, which in turn helps to achieve longer battery life.

The AVR® XMEGA® AU family of MCUs are peripheral rich, high performance, and low-power 8-bit microcontroller devices. The picoPower® technology and hardware crypto module of XMEGA AU MCUs enable the systems to achieve low-power consumption, encrypted communication, and secure data storage, which is crucial for several IoT applications.

The LoRa® technology is a Low Power Wide Area Network (LPWAN), with key characteristics like wider coverage, lower bandwidth, small packet, and application layer data sizes and long battery life operation. The LoRaWAN network protocol targets key requirements of IoT such as secure bi-directional communication, mobility and localization services.

The combination of the Microchip AVR XMEGA AU family of MCU devices and LoRa wireless transceiver modules offers key features such as low data rate, longer communication range, low power consumption, and a secure and efficient network which are ideal for IoT applications.

This application note discusses the smart IoT wireless sensor node realized using XMEGA AU MCU and LoRa wireless technology.

The application note covers the most vital features of the AVR XMEGA AU product family of MCUs for IoT applications and an overview of the LoRaWAN communication protocol. The supplemented IoT wireless sensor node firmware is developed using the Microchip ATxmega256A3BU MCU. The firmware offers a LoRa sensor node with various methods to join (i.e. ABP, OTAA) to a LoRa network. After successful connection to the LoRa network, the firmware periodically monitors sensors and transmit the acquired data to a cloud-based network server through a gateway. The application note also provides power optimization techniques for XMEGA AU MCUs to reduce the overall system power consumption.

Features

This application note features the following contents:

- An overview of the LoRaWAN protocol

- Highlighting features of XMEGA AU MCUs for IoT applications
- An overview of the LoRa application
- Demo firmware for the LoRa sensor node
- Low-power considerations

Table of Contents

Introduction.....	1
Features.....	1
1. Definitions.....	5
2. Overview of LoRaWAN Protocol.....	6
2.1. LoRaWAN Security.....	8
2.2. End Device Personalization and Activation.....	9
2.2.1. Over the Air Activation (OTAA).....	9
2.2.2. Activation by Personalization (ABP).....	9
2.3. LoRaWAN Classes.....	9
2.3.1. Class A End-Devices.....	10
2.3.2. Class B End-Devices.....	10
2.3.3. Class C End-Devices.....	10
2.4. LoRa Technology for IoT Applications.....	11
3. Features of XMEGA AU Devices.....	12
3.1. AVR CPU and Instruction Set.....	12
3.2. Clock System.....	12
3.3. ADC.....	13
3.4. Event System.....	13
3.5. DMA Controller.....	14
3.6. Crypto Engine.....	14
3.7. USB.....	14
3.8. picoPower® Technology.....	14
3.8.1. True 1.6V Operation.....	15
3.8.2. Minimized Leakage Current.....	15
3.8.3. Ultra-Low Power 32 kHz Crystal Oscillator.....	15
4. Application Overview.....	16
4.1. Hardware Overview.....	17
4.1.1. XMEGA-A3BU Xplained.....	17
4.1.2. RN2483 LoRa Technology Mote.....	18
4.1.3. LoRa Gateway Board.....	19
4.2. Demo Firmware Overview.....	19
4.2.1. MCU Clock and Peripheral Configuration.....	19
4.2.2. Firmware Operation.....	20
4.3. Application Flow.....	23
4.4. Demo Setup.....	24
4.4.1. Sensor Node Arrangement.....	24
4.4.2. LoRa Gateway Core Board Arrangement.....	25
4.4.3. LoRa Functional Parameters Configuration.....	25
4.4.4. Transmit Data from Sensor Node to TTN Server.....	26
4.5. Power Consumption Results.....	28

4.6.	Program and Data Memory Requirements.....	29
4.7.	Application Results.....	29
5.	Low Power Considerations.....	30
5.1.	Optimizing MCU Power Consumption.....	30
5.1.1.	Operating Voltage.....	30
5.1.2.	Operating Frequency.....	30
5.1.3.	Clock Source Selection.....	30
5.1.4.	Clock Prescaling.....	31
5.1.5.	Active Mode Operation.....	31
5.1.6.	Sleep Modes.....	31
5.1.7.	Unused Peripherals.....	31
5.1.8.	Unused GPIOs.....	32
5.1.9.	Analog I/Os.....	32
5.1.10.	Brown-Out Detector (BOD).....	32
5.1.11.	RTC Clock Source.....	32
5.1.12.	Watchdog.....	32
5.1.13.	JTAG Interface and On-Chip Debugging.....	33
5.1.14.	Wake-Up Delays.....	33
5.1.15.	Virtual Port Registers.....	33
5.2.	Code Optimization.....	33
6.	Conclusion.....	35
7.	Reference.....	36
8.	Revision History.....	37
	The Microchip Web Site.....	38
	Customer Change Notification Service.....	38
	Customer Support.....	38
	Microchip Devices Code Protection Feature.....	38
	Legal Notice.....	39
	Trademarks.....	39
	Quality Management System Certified by DNV.....	40
	Worldwide Sales and Service.....	41

1. Definitions

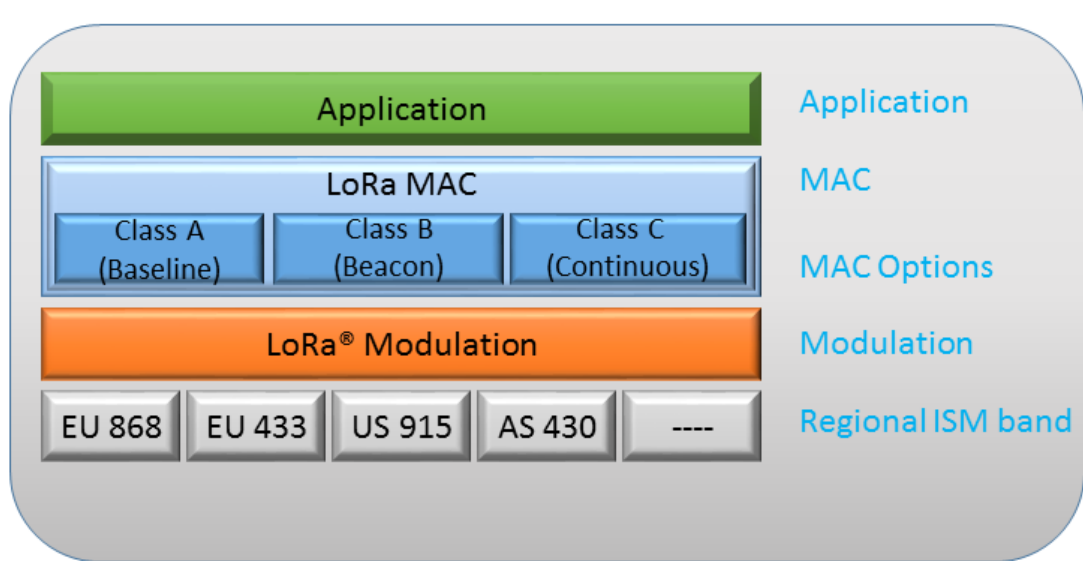
- **LoRaWAN: Long Range Wide Area Network**
- **End-devices:** End-devices are the sensor nodes in a LoRa network
- **Gateway:** The gateway is a transparent bridge relaying messages between end devices and a network server
- **Network Server:** The network server is responsible for managing the state of active nodes within the network
- **Application Server:** The application server is the user-facing service, responsible for managing the inventory of nodes. It provides an easy user web-interface for managing users, applications, and nodes
- **Up-link Transmission:** The messages sent by end-devices to a network server
- **Down-link Transmission:** The messages sent by network server to an end-device
- **End-device address (DevAddr):** This is useful to identify an end-device within a LoRaWAN network
- **Application Identifier (AppEUI):** This is a global application ID in IEEE EUI64 address space that uniquely identifies the application during join request
- **Network session key (NwkSKey):** The network session key is specific for an end-device. It is used by both the network server and the end-device to prove/verify the data packets authenticity and integrity
- **Application session key (AppSKey):** The application session key is specific for an end-device. It is used by both the network server and the end-device to encrypt, decrypt the application payload
- **End-device Identifier (DevEUI):** This is a global end-device ID in IEEE EUI64 address space that uniquely identifies the end-device
- **Application key (AppKey):** The AppKey is an AES 128 root-key specific to the end-device. If an end-device joins in a network via OTAA activation, the AppKey is used to derive both NwkSKey and AppSKey specific to that end-device to encrypt and verify network communication and application data.

2. Overview of LoRaWAN Protocol

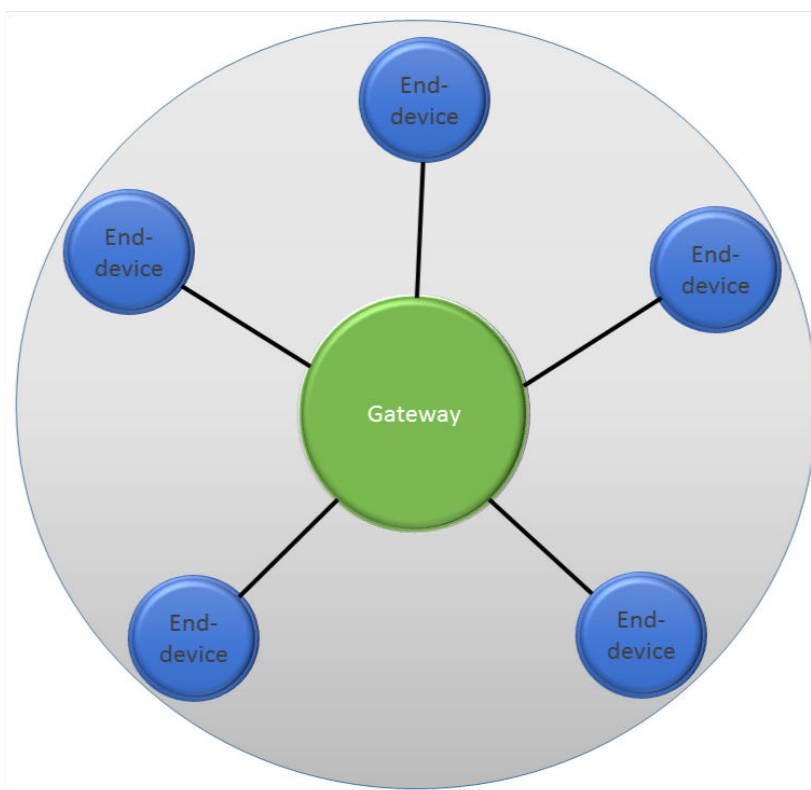
The LoRa [Long Range Radio] is a sub-GHz wireless RF technology. LoRa is the physical (PHY) layer or the wireless modulation at the hardware level, used to create long-range wireless communication links. The LoRa technology uses chirp spread spectrum modulation to enable long range, robust communication, and low-power consumption.

LoRaWAN is a Wide Area Network (WAN) communication protocol specification built on top of the LoRa technology. The LoRaWAN protocol is a media access control (MAC) layer, which extends the LoRa physical communication layer onto internet networks. The LoRaWAN defines communication protocol and system architecture for the network. [Figure 2-1](#) shows the different layers of a LoRa network. The LoRaWAN protocol and network architecture have the most influence in determining battery lifetime of a node, security, network capacity, quality of service, and the variety of applications served by the network.

Figure 2-1. LoRa Network Architecture



The LoRaWAN network architecture is typically laid out in a star-of-stars topology. [Figure 2-2](#) shows block diagram view of a star topology. Hence, the LoRaWAN network does not enable device-to-device communications. Data packets can only be transmitted from an end-device to the network server or vice versa.

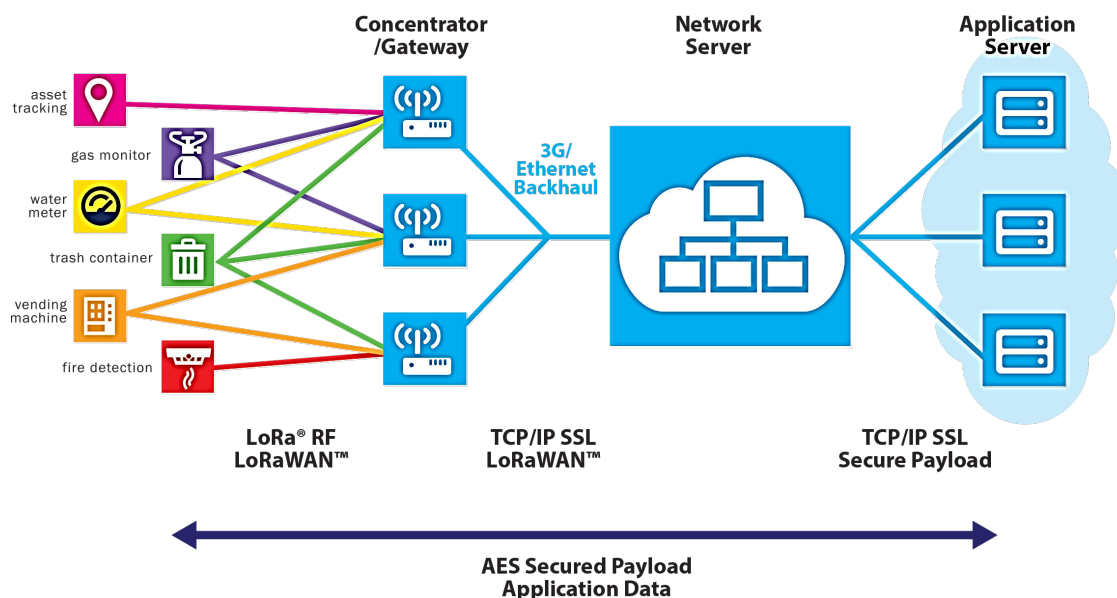
Figure 2-2. Star Topology

The LoRa network operates in the license-free 433 MHz, 868 MHz, and 915 MHz Industrial, Scientific, and Medical (ISM) bands, with data rates of 0.3 kbps to 50 kbps for a 125 kHz bandwidth. The LoRaWAN network protocol provides a media access control mechanism, enabling many end-devices to communicate with a gateway using the LoRa modulation.

A typical LoRa network consists of the below elements:

- End Devices
- Central Gateway
- Network Server

Figure 2-3. LoRa Network Topology



End-devices are the low-power sensor nodes which usually sense data in the field. These nodes are often placed remotely. A LoRa network can have multiple gateways, and the same data packet can be received (and forwarded) by more than one gateway.

The network server is an intelligent and core element in a LoRa network. It manages the network, perform security checks, filters redundant received packets, decoding the packets, schedule acknowledgments through the gateway, and also performs adaptive data rate. If the packet is intended for an application server, the network server sends the packet to the specific application server. Using application server, the user receives data from end-devices through a network server for further analysis. Additionally, the application server determines end-device actions. Figure 2-3 shows the typical LoRa network topology.

The end-devices use the single-hop LoRa communication to communicate with the central gateway. The gateway is connected to the network server via standard IP connections (for example: Ethernet, Wi-Fi®, 3G).

The end-point communication is bi-directional. Communication between end-devices and gateway is spread out on different frequency channels and data rates. The selection of data rate is a trade-off between communication range and message duration. Due to the spread spectrum technology, communications with different data rates do not interfere with each other and create a set of virtual channels increasing the capacity of the gateway. To maximize both battery life of the end-devices and overall network capacity, the LoRaWAN network server is managing the data rate and RF output for each end-device individually by means of an adaptive data rate (ADR) scheme.

2.1 LoRaWAN Security

Robust security is a key element for LoRa network. The LoRaWAN security design adheres the fundamental properties such as mutual authentication, integrity protection, and confidentiality. The LoRaWAN security uses AES 128-bit cryptographic encryption algorithm. The LoRaWAN security guarantees end-to-end encryption for application payloads exchanged between end-devices and application servers.

The LoRaWAN protocol includes a number of different layers of security at the network and application level for secure communications. The network security ensures authenticity of the node in the network while the application layer of security ensures the network operator does not have access to the end users application data.

2.2 End Device Personalization and Activation

To participate in a LoRaWAN network, each end-device has to be personalized and activated. After activation, the device address (DevAddr), an application identifier (AppEUI), a network session key (NwkSKey), and an application session key (AppSKey) is stored in the end-device. In a LoRaWAN network, the end-device personalization and activation can be achieved in two methods.

- Over the Air Activation (OTAA)
- Activation by Personalization (ABP)

2.2.1 Over the Air Activation (OTAA)

In the over-the-air activation method, end-devices must follow a join procedure prior to perform data exchanges with the network server. An end-device has to go through a new join procedure every time it has lost the session context information.

The join procedure requires the end-device to be personalized with a unique end-device identifier (DevEUI), an application identifier (AppEUI), and Application Key (AppKey) information before it starts the join procedure (join request - join accept procedure).

2.2.2 Activation by Personalization (ABP)

The ABP method ties an end-device to a specific network bypassing the join request, join accept procedure. Activating an end-device by personalization means, the device address (DevAddr) and the two unique session keys (NwkSKey and AppSKey) are pre-configured directly into the end-device. The end-device is equipped with the above-required information for participating in a specific LoRaWAN network when started.

The OTAA method is a more preferred method as the keys are not pre-determined and can be regenerated.

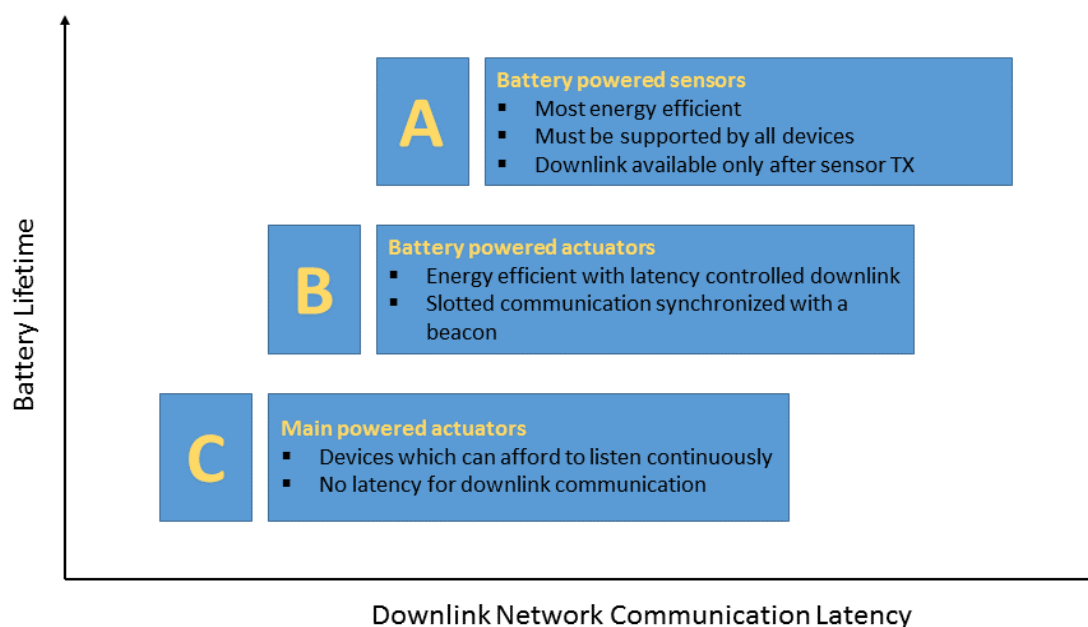
2.3 LoRaWAN Classes

To address a variety of end application profiles, LoRaWAN network utilizes end-devices from the following device classes.

- Class A: Bi-directional end-devices
- Class B: Bi-directional end-devices with scheduled receive slots
- Class C: Bi-directional end-devices with maximal receive slots

All three classes allow bi-directional communication and can initiate an up-link to the server via the gateway. The key difference between these device classes is the trade-off made between network down-link communication latency and power consumption. Each class serves different application needs and has optimized requirements for specific purposes. [Figure 2-4](#) shows the down-link network communication latency versus battery lifetime for the end-devices of different classes.

Figure 2-4. End-Device Classes



2.3.1 Class A End-Devices

The class A end-devices allow for bi-directional communications whereby each end-device's up-link transmission is followed by two short down-link receive windows. The transmission slot scheduled by the end-device is based on its own communication needs with a small variation based on a random time basis (ALOHA-type of the protocol).

This Class A operation is the lowest power end-device system for applications that only require down-link communication from the server shortly after the end-device has sent an up-link transmission. Down-link communications from the server at any other time will have to wait until the next scheduled up-link.

2.3.2 Class B End-Devices

The class B end-devices allow for more receive slots. In addition to the Class A random receive windows, Class B devices open extra receive windows at scheduled times. In order for the end-device to open its receive window at the scheduled time, it receives a time synchronized beacon from the gateway. This allows the server to know when the end-device is listening.

2.3.3 Class C End-Devices

The class C end-devices have nearly continuously open receive windows, only closed when transmitting. Class C end-device will use more power to operate than Class A or Class B, but they offer the lowest latency for the server to end-device communication.

For more detailed information on LoRaWAN protocol and LoRa technology, refer the LoRaWAN specification documents available with [LoRa Alliance](#).

2.4 LoRa Technology for IoT Applications

The embedded developers have a wide variety of wireless technologies to connect a product to the Internet of Things (IoT). Each technology suits different applications, requiring developers to carefully consider factors such as range, data rate, cost, power consumption, and wireless network capacity. Each wireless technology has both strong and weak points. For example, standard Wi-Fi can transmit large amounts of data at high speeds, but it has a limited range. A cellular network combines high speed and long range, but it is power hungry. The low power wireless networks are a key enabler for IoT.

For IoT applications, the LoRa technology's critical advantage over the established wireless protocols such as Wi-Fi and Bluetooth® technologies is long range, low data rate, and low power consumption of the connected end-devices.

These unique features of LoRa makes it an interesting candidate for smart sensing technology in applications such as health monitoring, smart metering, urban light control, and environment monitoring, etc., as well as in industrial applications. Thus, the LoRa wireless technology is essential for battery-operated wireless applications in the IoT, Machine-to-Machine (M2M) market segments.

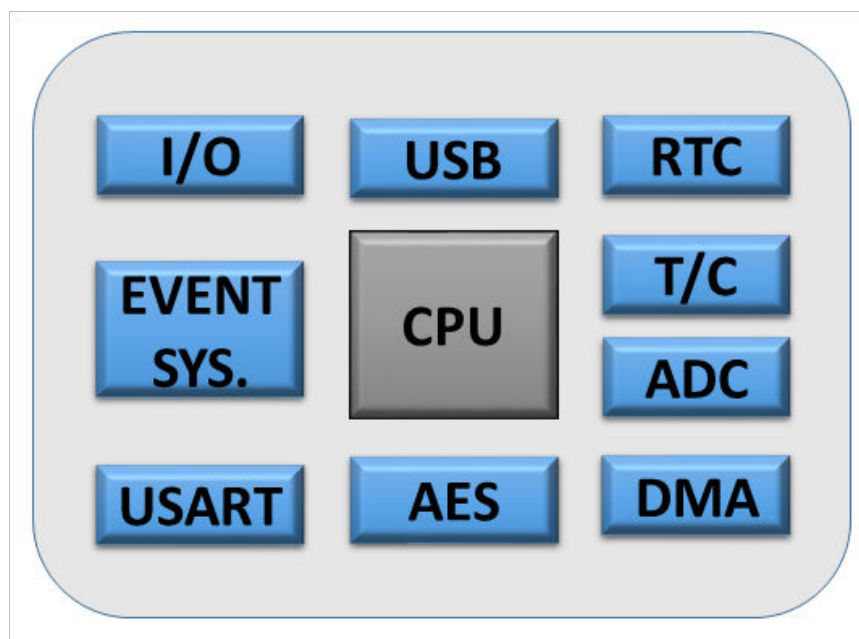
Additionally, the LoRaWAN network protocol is an open standard, supported by the LoRa Alliance. This enables the developers to develop cost-effective and customized protocol for any specific needs of the IoT applications.

3. Features of XMEGA AU Devices

For embedded applications, system performance is much more than a good MIPS number. It is important to have powerful peripherals and features for an MCU that allow the application to run smoothly with minimum power consumption.

The XMEGA AU is a family of low-power, high performance, and peripheral-rich CMOS 8-bit MCUs based on the AVR enhanced RISC architecture. All the AVR XMEGA AU are 100% pin- and code-compatible across all devices from the smallest to the largest device. It is possible to develop with any XMEGA AU device, and switch to any other XMEGA AU device later without having to change any code. [Figure 3-1](#) shows the important peripherals of an XMEGA AU MCU usually required for the IoT applications.

Figure 3-1. XMEGA AU MCU Peripherals



In this section, the most highlighting features of XMEGA AU MCU devices are described, which are essential for IoT applications.

3.1 AVR CPU and Instruction Set

The CPU design and instruction set are tuned for minimum code size and maximum execution speed. The AVR CPU executes powerful arithmetic and logical operations in a single clock cycle. Hence, the XMEGA AU devices achieve throughput's approaching one million instructions per second (MIPS) per megahertz, allowing the designer to optimize power consumption versus processing speed.

3.2 Clock System

AVR XMEGA's clock system allows flexible change of frequency. Dynamic clock switching allows the embedded designer to tune performance and power consumption to fit the application. The internal PLL and prescaler can be used to scale the clock sources dynamically up or down to further match application requirements.

With a built-in external oscillator failure detection and internal RC oscillator with $\pm 1\%$ accuracy over temperature and voltage, XMEGA AU MCUs offer the safest and most reliable and flexible clock system.

3.3 ADC

The AVR XMEGA AU MCUs are highly integrated with dual-sixteen channel pipeline ADC modules for high analog accuracy. The high-performance 12-bit pipeline ADC is capable of conversion rates up to 2 million samples per second (MSPS).

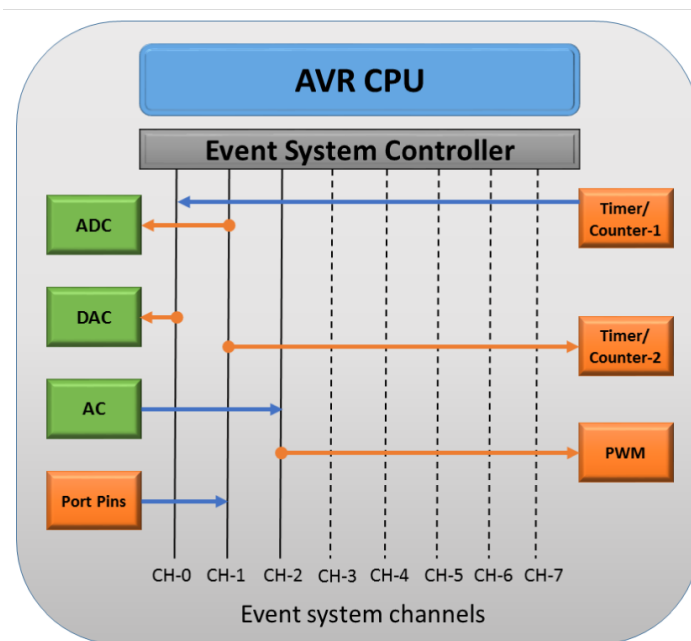
The ADC conversion block has a 12-stage pipelined architecture. All the channels of an ADC module use the same ADC pipeline for the conversions, and the pipeline enables a new conversion to be started for each ADC clock cycle. Thus, multiple ADC measurements from different channels can be converted simultaneously and independently.

A wide range of multiplexer (MUX) settings, integrated gain stage, and four virtual input channels make this a flexible ADC module suitable for a wide range of applications such as data acquisition, embedded control, and general signal processing.

3.4 Event System

The innovative AVR XMEGA event system allows peripherals to send signals (events) directly to other peripherals without involving the CPU. This ensures a short and 100% predictable response time and at the same time offloads the CPU. [Figure 3-2](#) shows the example usage of the event system controller in XMEGA AU MCUs.

Figure 3-2. Event Signals to Peripherals



In [Figure 3-2](#), the Timer/Counter, PWM, and Port pins are the event generators and the ADC, DAC, and AC modules are event users.

The event system is operative in both active and idle sleep modes. In all other sleep modes, the event system is not available for the inter-peripheral communication.

3.5 DMA Controller

The AVR XMEGA AU MCUs 4-channel Direct Memory Access (DMA) controller enables fast, CPU independent data transfer between any locations in the data memory space and peripherals.

The DMA controller supports four different types of data transfers:

- From data memory to data memory
- From data memory to peripheral
- From peripheral to data memory
- From peripheral to peripheral

The DMA controller supports flexible channel priority selection, several addressing modes, and double buffering capabilities. Each DMA transfer can range from 1B to 16 MB. The usage of DMA controller for data transfers increases CPU computing performance and lowering power consumption by allowing the CPU to spend more time in sleep mode. The DMA controller is a powerful module for all data-oriented applications such as signal processing and industrial control.

3.6 Crypto Engine

AVR XMEGA AU family of devices includes a crypto engine for 64-bit DES and 128-bit AES encryption and decryption. These are supported through a DES CPU instruction and an AES peripheral module respectively. The crypto engine in XMEGA AU devices supports secure high bandwidth transmissions with additional low-power requirements.

In today's interconnected world of IoT, security of electronic devices is essential for secure data communication, processing, or data storage. The authentication ensures that end-devices are interacting with authorized gateways and cloud services and they, in turn, verify they are working with authentic IoT end devices. Hence, the crypto engine available in AVR XMEGA AU MCUs is a very useful feature for the IoT applications.

3.7 USB

The USB module in AVR XMEGA AU is a USB 2.0 compliant interface and offers an efficient USB solution to integrate USB in most applications. The USB module supports full speed (12 Mbps) and low speed (1.5 Mbps) operations.

The USB module has become a standard and is an easy-to-use communication interface. In IoT applications, the USB becomes more attractive for designers to integrate as the primary interface, both internally and externally. In IoT wearable marketing segment, the USB interface is useful for data transfer and charge purposes.

3.8 picoPower[®] Technology

The increasing usage of battery and signal line powered applications demand low-power solutions. The most innovative picoPower[®] technology being used in the ultra-low power 8-bit AVR XMEGA microcontroller's architecture achieves the industry's lowest power consumption. Sleep is often considered the most important low-power mode. The picoPower technology employs the below important techniques which result in the lowest sleep mode power consumption.

3.8.1 True 1.6V Operation

The 1.6V operation is a true 1.6V operation. This means all the analog peripherals, Flash, EEPROM, and RAM run at 1.6V. This allows safe operation directly from a $1.8V \pm 10\%$ power supply. It also enables deeper battery discharge to increase battery life.

3.8.2 Minimized Leakage Current

The temperature, the supply voltage, and the process affect the leakage current. The proprietary processes that have been specifically developed for low-power operation, has reduced the leakage current of picoPower AVR XMEGA MCUs to less than 100 nA.

3.8.3 Ultra-Low Power 32 kHz Crystal Oscillator

Since the time spent in active mode can be insignificant compared to the time spent in power save mode, power save mode is often the most important power consumption characteristic of an MCU. The latest 32 kHz crystal oscillator design utilized in the AVR XMEGA MCUs reduces the current consumption in power save mode to a level comparable to power down mode.

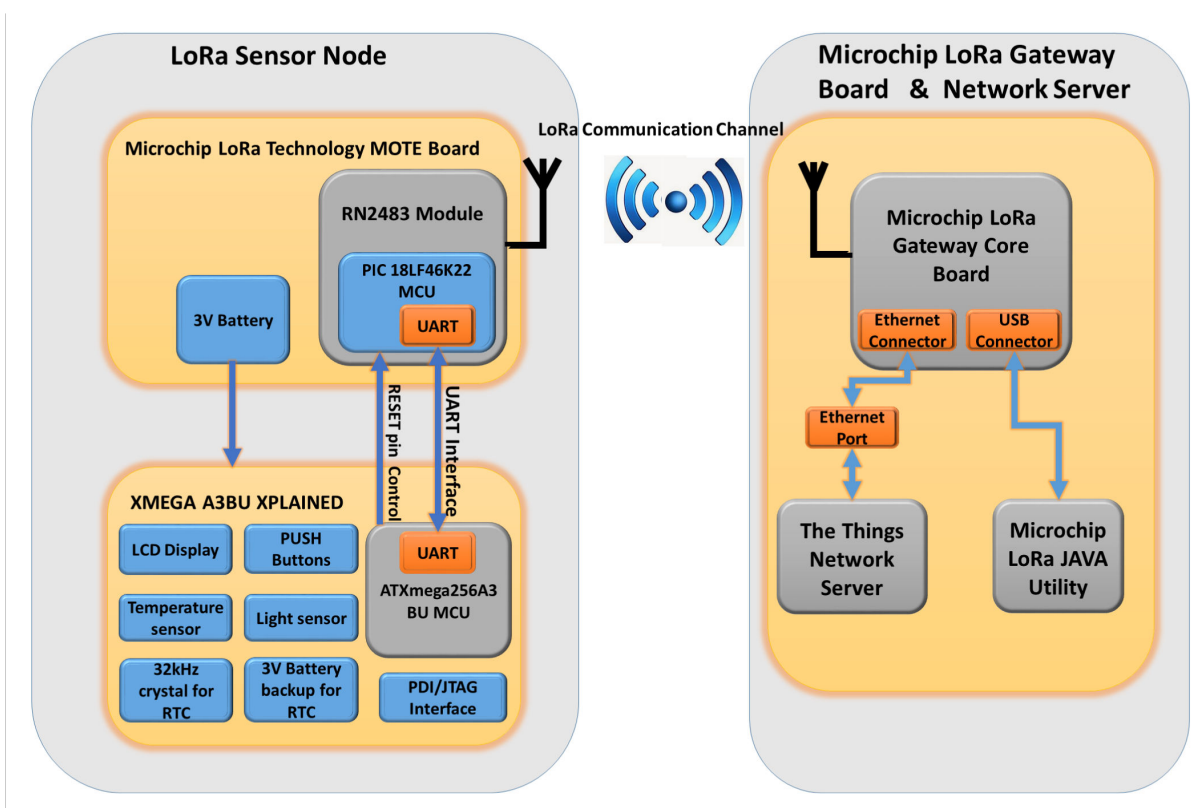
With a supply voltage of 1.6V, the AVR picoPower technology achieves the industry's lowest power save current consumption of 650 nA with the 32 kHz oscillator running and a sleeping BOD.

4. Application Overview

This application note discusses the LoRa based IoT wireless sensor node, developed using XMEGA AU MCU and RN2483 LoRa transceiver module.

The scope of this application is only limited to development of a LoRa sensor node. The LoRa gateway core board available with Microchip LoRa evaluation kit is used as a gateway for the necessary communication between the sensor node and a network server. The things network (TTN), an open standard network server, is used to receive the data packets from the sensor node. The Microchip LoRa Development Utility (GUI) is used to configure essential parameters of the gateway and RN2483 module. [Figure 4-1](#) shows the block diagram view of the LoRa network modeled using XMEGA AU based sensor node, LoRa gateway core board, and TTN network server.

Figure 4-1. Block Diagram View of LoRa Network - XMEGA AU MCU Based Sensor Node



After power up, the LoRa sensor node initiates a request to join the LoRa network. The sensor node can join the LoRa network by personalization and activation. In the LoRa network, the sensor node can be personalized and activated either by OTAA or ABP method. Once the sensor node is successfully joined in the LoRa network, the sensor node scans on-board temperature and light sensors either periodically or to an arbitrary button press event. The processed sensor's data is transmitted to the LoRa gateway, which will be accessed by the things network (TTN) server. The application server configured on the things network displays the received sensors data. Upon successful reception of confirmed up-link transmission data, the network server sends an acknowledgment to the sensor node along with a down-link message if present. Then after, the sensor node goes into sleep mode until the next turn for sensors

data measurement. When the sleep period is elapsed, the sensor node wakes up from sleep and starts scanning the sensors and transmit the processed sensors data to the network server through a gateway.

4.1 Hardware Overview

In this application, the elements of the LoRa network (i.e. sensor node and gateway) are realized using the existing evaluation boards instead of developing a custom hardware.

The LoRa sensor node is a standalone battery-powered end-device and belongs to Class-A category of end-devices. The sensor node is modeled using the existing XMEGA-A3BU Xplained and RN2483 LoRa Technology Mote evaluation boards.

Note:

- *Since the XMEGA-A3BU Xplained board comes with the ATxmega256A3BU MCU, the application demonstrates a sensor node with an ATxmega256A3BU MCU. The developers may consider an XMEGA AU MCU with lower memory footprint for their applications depending upon the program and data memory requirements.*

As shown in [Figure 4-1](#), the XMEGA-A3BU Xplained and RN2483 LoRa Technology Mote evaluation boards are connected through a UART interface. The 3V battery source on the LoRa mote board powers both the evaluation boards.

The onboard temperature and light sensors of XMEGA-A3BU Xplained are used to monitor the ambient light and temperature. The LCD display on XMEGA-A3BU Xplained provides feedback on connection status, sensors data, and down-link transmission data or acknowledgments. The RTC32 works with battery backup feature. The 3V battery source and external 32 kHz crystal oscillator is to power and enable the clock for the RTC32 module in power down mode.

The RN2483 LoRa transceiver module alone is enabled on the LoRa mote evaluation board. The onboard LCD display, sensors, and PIC MCU are isolated from the sensor node circuitry.

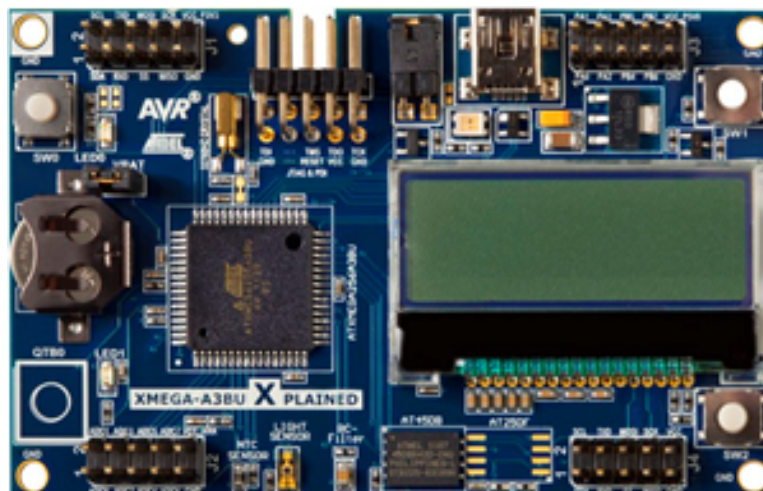
The LoRa gateway core board available with the Microchip LoRa technology evaluation kit is used as a gateway for the LoRa network. The gateway is connected to the TTN server via an Ethernet LAN port. The following subsections give a brief overview of the evaluation boards used to realize a LoRa network.

4.1.1 XMEGA-A3BU Xplained

The XMEGA-A3BU Xplained consists of an ATxmega256A3BU MCU device and works with a 3.3V supply voltage. [Figure 4-2](#) shows an image of the XMEGA-A3BU Xplained. The main highlighting features of the Xplained kit are:

- Light and Temperature sensors
- Three mechanical buttons
- Four status indication LEDs
- USB port
- 128x32 pixel FSTN LCD display

Figure 4-2. XMEGA A3BU Xplained



4.1.2 RN2483 LoRa Technology Mote

The RN2483 LoRa Technology Mote is a Class A end-device based on the Microchip's RN2483 LoRa modem. It is a standalone battery-powered node. Figure 4-3 shows the image of a Microchip RN2483 LoRa Technology mote. The mote provides a convenient platform to quickly demonstrate the long-range capabilities of the modem, as well as to verify inter-operability when connecting to LoRaWAN v1.0 compliant gateways and infrastructure.

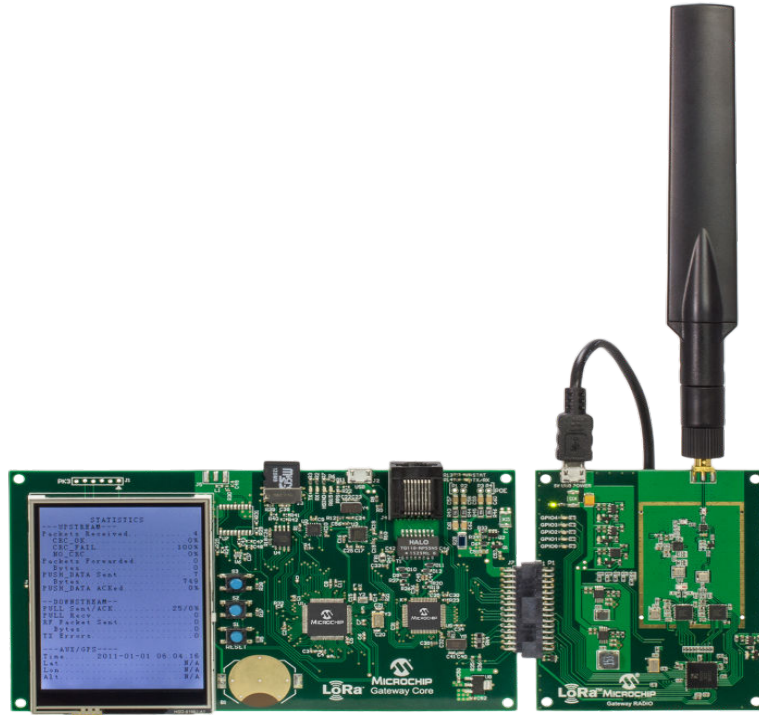
Figure 4-3. Microchip RN2483 LoRa Technology Mote



A standard USB interface is provided for connection to a host computer, providing a bridge to the UART interface of the RN2483 modem. As with all Microchip RN family of products, this enables rapid setup and control of the onboard LoRaWAN protocol stack using the high-level ASCII command set.

LoRa Gateway Board

Figure 4-4. Microchip LoRa Gateway Board



Demo Firmware Overview

The demo firmware is based on an ATxmega256A3BU MCU and generated using Atmel Software Framework (ASF) and Atmel Studio7.0 IDE. The firmware is tested with XMEGA-A3BU Xplained board.

MCU Clock and Peripheral Configuration

In the demo firmware of the LoRa sensor node, the internal 32MHz RC Oscillator is calibrated using DFLL to generate 48 MHz clock. The USB Start of Frame (SOF) is used as a reference for DFLL. The 48 MHz DFLL clock output is used for full speed USB operation. The DFLL clock is prescaled to operate the CPU at a 12 MHz frequency.

The following peripherals of the ATXmega256A3BU MCU are configured to perform the desired sensor node operation.

- Analog-to-Digital Converter (ADC)
- Event System Controller
- Direct Memory Access (DMA) Controller
- Timer
- Real Time Clock (RTC)

- Universal Serial Bus (USB) Interface
- UART Interface

4.2.2 Firmware Operation

The demo firmware implementation ensures the application operates in either of the below two modes depending upon USB cable connection status and the push buttons SW0 and SW1 state.

- USB mode
- Sensor Node mode

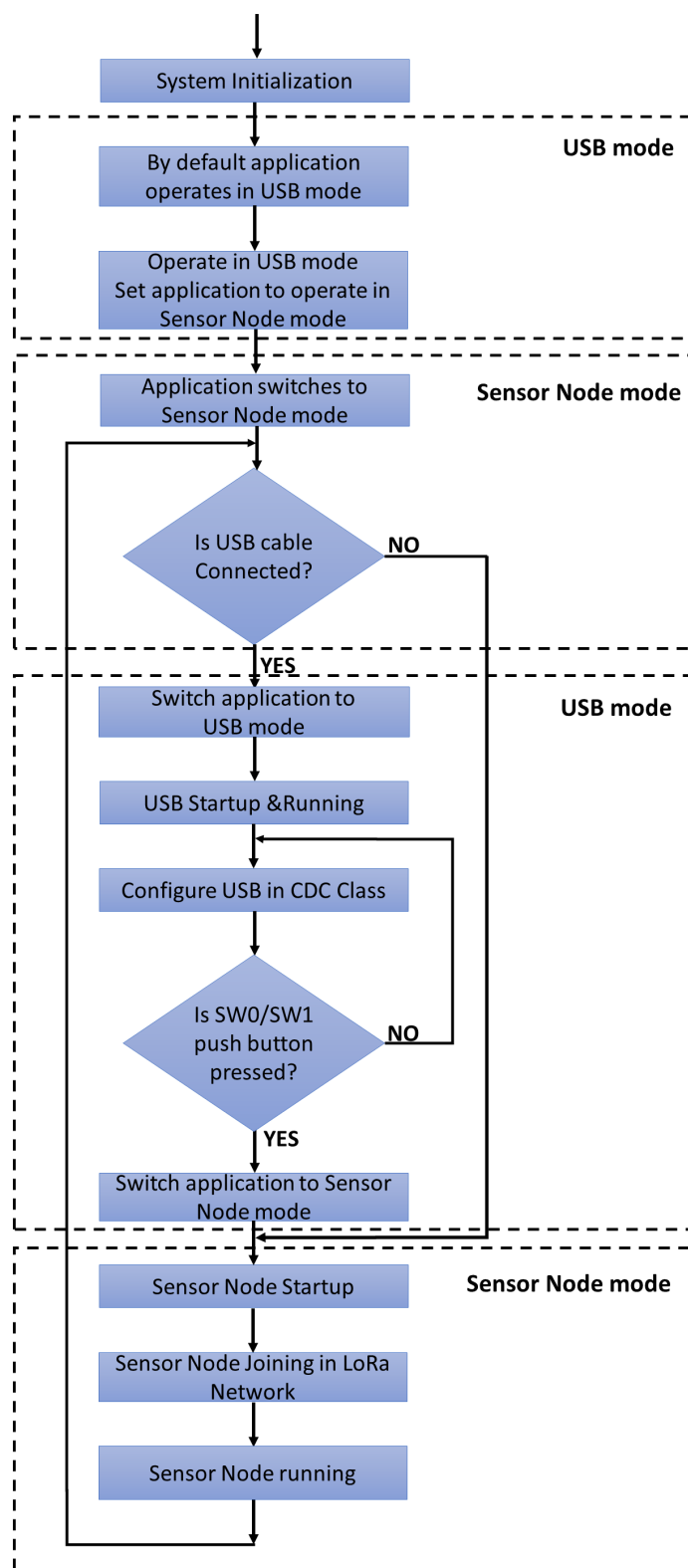
In the USB mode of operation, the sensor node works on USB power. Also, the MCU communicates with the USB host through a USB-UART bridge interface to transmit/receive data.

Whereas in the sensor node mode, the sensor node works on battery power. In this mode, the sensor node joins in a LoRa network and periodically transmit the captured sensors data to The Things Network server through LoRa gateway. Also, in sensor node mode of operation, the ATxmega256A3BU MCU power down mode is enabled to reduce the power consumption.

Figure 4-5 shows a flow diagram of the two different operating modes of sensor node application firmware. After MCU power-up, by default, the application operates in USB mode. From USB mode, the application switches into sensor node mode operation. As long as the USB cable is not plugged in, the application operates in the sensor node mode.

If the USB cable is plugged in, the application comes out of sensor node mode and switches to USB mode. In this mode, the USB is configured to operate in CDC class. To an arbitrary push-button (either SW0 or SW1) press event, the application switch back to the sensor node mode, perform the desired operation. Then after, the application checks for the USB connection status. Depending upon the USB cable connection status, the application continues in either of the two operating modes.

Figure 4-5. Operating Modes in Sensor Node Application Firmware



The following sub-sections describe the firmware implementation of USB and sensor node modes.

4.2.2.1 USB Mode

In this mode, the USB interface of ATxmega256A3BU MCU acts as a bridge between the PC (i.e. USB Host) and the sensor node. The USB interface accepts the data coming from the Microchip LoRa Development Utility (GUI) and converts it to UART commands through the USB-UART bridge, which is then transmitted to the RN2483 LoRa transceiver module over the UART interface.

4.2.2.2 Sensor Node Mode

In the demo firmware, the sensor node mode delivers sensor node functionality. The timer module provides time reference in the application. The timer is configured to generate overflow interrupt for every 1 msec. In sensor node mode, the sensor node goes through the following three different phases for successful operation in the LoRa network.

Sensor Node Start-Up:

The sensor node start-up is the first phase to start with, as soon as the sensor node enters in sensor node mode of operation. In start-up stage, the LCD screen of the XMEGA-A3BU Xplained displays the various methods of linking a sensor node to a LoRa network. Then after, the application waits for the user response on suitable method.

Sensor Node Joining:

In sensor node joining phase, the sensor node joins in a LoRa network through a personalization and activation process. The application firmware allows the users to manually choose between ABP and OTAA methods for sensor node activation, using push buttons SW0 and SW1 respectively.

Sensor Node Running:

Once the sensor node is successfully joined in a LoRa network, it continues to operate in this phase. In the sensor node running phase, the LCD screen displays the menu of various user configurable sensor node parameters. These parameters are listed in [Table 4-2](#). The application monitors for a button press event on SW0/SW1. To an arbitrary button press event on either SW0 or SW1, the application enters into the sensor node parameters configuration mode. After the sensor node parameters configuration, the application monitors for any button press event until the inactivity counter gets elapsed.

If no button press event is reported during the inactivity period, the application checks for any periodic data transmission trigger event. After the data transmission event is triggered, the application enables the ADC module. The ADC module performs parallel data acquisition of the temperature and light sensors for sixteen successive iterations. The DMA controller stores the averaged sensors data in data buffers without any CPU intervention. Then MCU disables ADC, Event System, DMA controller and transmits the data to the RN2483 module through the UART interface.

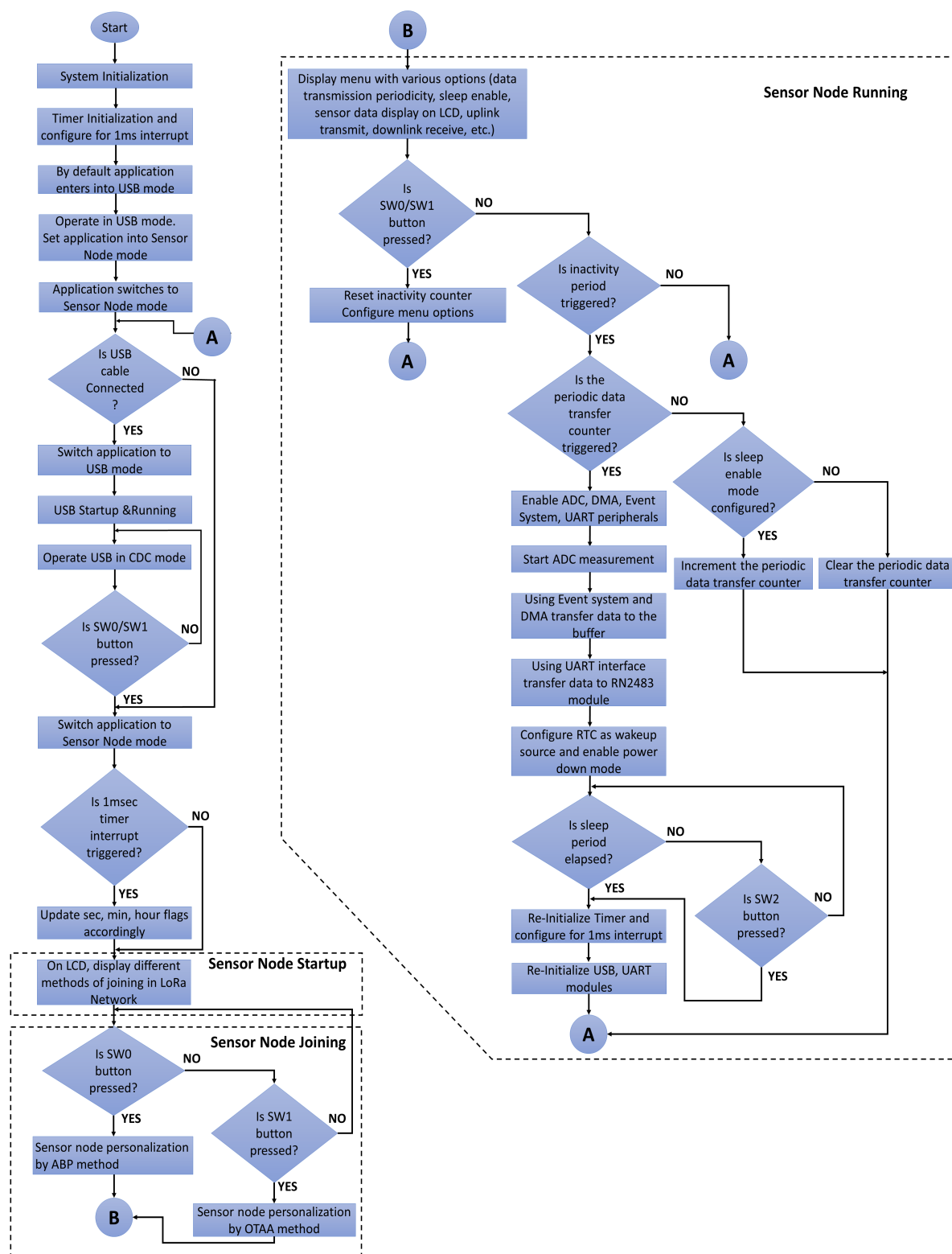
The RN2483 module transmits data to the TTN server through the LoRa gateway. Upon reception of acknowledgment from the TTN server, the MCU enables the RTC module, disable all other peripherals and goes into power down mode.

The MCU wakes up from sleep either to an SW2 button press event or to an RTC interrupt after the sleep period is elapsed. Then after, the application re-enables the USB and UART modules and continues with the sensor node running phase.

[Figure 4-6](#) shows the application flow of the sensor node firmware.

4.3 Application Flow

Figure 4-6. Application Flow of the Sensor Node Firmware



4.4 Demo Setup

This section describes the sequence of steps to be followed for the LoRa hardware setup and also the necessary LoRa functional parameters configuration through The Things Network (TTN) server and Microchip LoRa Development Utility GUI.

The LoRa network hardware setup comprises of:

- XMEGA-A3BU Xplained
- Microchip RN2483 LoRa Technology Mote board
- Microchip LoRa Gateway Core board

4.4.1 Sensor Node Arrangement

As shown in [Figure 4-1](#), the ATxmega256A3BU based sensor node is realized using XMEGA-A3BU Xplained and Microchip RN2483 LoRa technology mote board.

On the RN2483 LoRa technology mote board, only the 3V battery source and the RN2483 LoRa transceiver module are being used by the sensor node.

Note: *To prevent the rest of the circuitry on the RN2483 LoRa technology mote board from consuming the battery power, the on-board PIC18LF45K50 MCU is put in power down mode and the LCD module is turned off.*

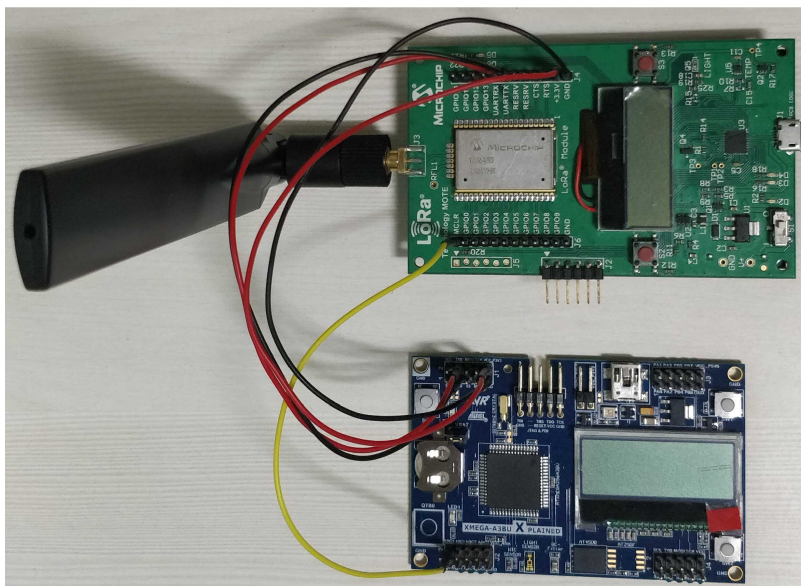
[Table 4-1](#) shows the essential connections required between the RN2483 LoRa technology mote board and the XMEGA-A3BU Xplained board.

Table 4-1. Connections Between RN2483 LoRa Technology Mote Board and XMEGA-A3BU Xplained

RN2483 LoRa Technology Mote	XMEGA-A3BU Xplained
UARTRX (PIN5 of Header J4)	TXD (PIN4 of Header J1)
UARTTX (PIN6 of Header J4)	RXD (PIN3 of Header J1)
+3.3V (PIN11 of Header J4)	P3V3 (PIN10 of Header J1)
GND (PIN12 of Header J4)	GND (PIN9 of Header J1)
MCLR (PIN1 of Header J6)	PIN2 of Header J2

[Figure 4-7](#) shows the sensor node arrangement using XMEGA-A3BU Xplained and RN2483 LoRa technology mote board.

Figure 4-7. Sensor Node Arrangement



4.4.2 LoRa Gateway Core Board Arrangement

The LoRa gateway acts as a bridge between the sensor node and the cloud-based The Things Network (TTN) server. To set up the LoRa gateway and connect to the cloud-based TTN server, the procedure given [here](https://www.thethingsnetwork.org/labs/story/setting-up-your-own-gateway-and-endpoint-with-microchips-lora-technology-evaluation-kit) (<https://www.thethingsnetwork.org/labs/story/setting-up-your-own-gateway-and-endpoint-with-microchips-lora-technology-evaluation-kit>) is very useful. Then after, ensure the gateway is “online” and successfully connected to the TTN cloud server.

4.4.3 LoRa Functional Parameters Configuration

Once the gateway is online and the sensor node is joined in the LoRa network, the sensor node transmits the data to the TTN server through the LoRa gateway. The up-link transmission data appears on an application server of the TTN dashboard. The following steps should be followed to successfully configure and link the sensor node to a LoRa network, and also to view sensor data on the application.

Connect a Mini-USB cable from the XMEGA-A3BU Xplained board to the PC running the Microchip LoRa Development Utility software. Hereafter, the LoRa sensor node works in USB CDC mode.

In the LoRa Development Utility software, click on “Find Devices” option. If the gateway board is already connected to the PC through the Micro-USB cable, both the gateway and the “RN module” appears in the device list. Select the RN module in order to configure it.

By default, the sensor node register to the network through **OTAA** method, by following the instructions mentioned below.

1. In the TTN console, from the Application or an application’s Devices screen, click on **register device** option.
 - For **Device ID**, enter appropriate device ID. For example “loramote1”
 - For **Device EUI**, enter an 8-byte hexadecimal value or click the generate icon at the left to auto-generate the Device EUI
 - Leave the **App Key** to be generated automatically
 - For **App EUI**, select the generated EUI from the list
 - Click **Register** to finish

- The control redirects to the newly registered device, where the generated **App Key** details are needed to activate the device
- 2. In the Server Authentication Keys section of the utility, select **OTAA** (over-the-air activation).
- 3. Select "Use Entered (DevEUI):" and copy the **Device EUI** from the TTN Console.
- 4. Copy the **App Key** and **Application EUI** from the TTN console to the appropriate boxes in the LoRa Development Utility software.
- 5. After each value is entered, press the "Enter" button on the PC keyboard to ensure that information gets sent to the RN module.
- 6. Click on "Save" option.
- 7. Click on "Join" option.
- 8. Then after, "ok" and "accepted" response appears in the RN module console.

After registering the sensor node by the **OTAA** method, in case of switching over to the **ABP** joining method, the following changes are necessary.

1. In the TTN console, go to the device that needs to be personalized.
 - From the top right menu, select the **Settings** option
 - For **Activation method**, click on the **ABP** option
 - To generate the **Network Session Key** and **App Session Key** automatically by the console, do not enter any information. To manually enter the customized keys information, click on the **customize it** option and enter the details
 - Click **Save** to finish
 - The control redirects back to the device, where the **Device Address** and session keys information is needed to activate the device
2. In the Server Authentication Keys section of the utility, select **ABP** (activation by personalization).
3. Select "Device Address (DevAddr)" and copy the **Device Address** from the TTN console.
4. Copy the **NetworkSessionKey** and **ApplicationSessionKey** from the TTN console to the appropriate boxes in the LoRa Development Utility software.
5. Follow the steps from 5 to 8 as mentioned above.

To switch the mode back to **OTAA** from **ABP**, the following changes are necessary.

1. In the TTN console, go to the device that needs to be personalized.
 - From the top right menu, select the **Settings** option
 - For **Activation method**, click on the **OTAA** option
 - Leave the App Key to be generated automatically
 - Click **Save** to finish
 - The control redirects back to the device, where the **App Key** details are needed to activate the device
2. In the Server Authentication Keys section of the utility, select **OTAA** (over-the-air activation).
3. Select "Use Entered (DevEUI):" and copy the Device EUI from the TTN console.
4. Copy the **App Key** and **Application EUI** from the TTN console to the appropriate boxes in the LoRa Development Utility software.
5. Follow the steps from 5 to 8 as mentioned above.

Note: If the joining method is changed in the TTN server, the device should use the same join procedure each time the device tries to join the network.

4.4.4 Transmit Data from Sensor Node to TTN Server

The below steps are useful to follow to transmit data from the sensor node to the TTN server.

1. Disconnect the Mini-USB cable from the XMEGA-A3BUXplained board, which is used to connect to the PC.
2. Insert two AAA batteries in the battery holder of the RN2483 LoRa technology mote board and turn switch S1 ON to power the board. Hereafter, the sensor node works in sensor node mode.
3. The initial message that gets displayed on the LCD of XMEGA-A3BUXplained board is XMEGA LoRa, Sensor Node Demo, V 1.0.3 Code.
4. Then after, it waits for the user inputs to choose the appropriate joining method for the sensor node to join the LoRa network.
5. Press button SW0 on XMEGA-A3BU Xplained to select the Activation By Personalization (ABP) method of joining the LoRa network.
Note: Press button SW1 to select OTAA method of joining.
6. After the sensor node has successfully joined the network, the ABP join successful message gets displayed on the LCD screen.
7. Use switches SW0 and SW1 to navigate between the different states of the sensor node.
8. If there isn't any button press activity for 15 seconds, the sensor node goes into sleep mode.
9. The sensor node periodically wakes up from the sleep mode to scan the light and temperature sensors and transmit the data to TTN server. The periodicity is a user-configurable parameter. By default, the periodic data transmission interval is set to one minute. The periodicity can be configured as 5 minutes, 10 minutes, 15 minutes, 30 minutes, or 60 minutes.

Table 4-2 shows the different states supported by the sensor node mode in demo application firmware.

Table 4-2. The Different States Supported by Sensor Node Mode

State	Description
Issue up-link message	Transmit light and temperature sensors data to the TTN server either as unconfirmed or confirmed messages. For the confirmed messages, the TTN server responds with an acknowledgment after each message reception.
View down-link message	View last received down-link data along with down-link message if present
Sensor data display	View light and temperature sensor data, which is updated every 1 second
Menu timeout	Enable or disable menu timeout of 15 sec. If disabled, the LoRa sensor node will not go into sleep mode.
Periodic up-link timing configuration	Configure periodic up-link message timing for sensor data to the TTN server. The time interval can be 1, 5, 10, 15, 30, or 60 minutes
Select data rate	The data rate can be configured as 0, 1, 2, 3, 4, 5, or ADR

4.5 Power Consumption Results

The power consumption data of ATxmega256A3BU MCU is captured while the sensor node is operating in active mode as well as in power down mode. The measurement setup for power consumption consists of:

- XMEGA-A3BU Xplained
- Multimeter for current measurement

Table 4-3. Operating Conditions and Peripherals Configured

MCU	ATxmega256A3BU
Operating Voltage	3.3V
Operating Frequency	48 MHz/4 = 12 MHz
CPU clock configuration	32 MHz internal oscillator calibrated to DFLL 48 MHz, prescaled to 12 MHz
Sleep mode	Power down
Peripherals enabled	ADC, RTC32, Timer0, UART, USB, Event System, DMA
ADC configuration	Operates with 12 MHz system clock, signed mode, 12-bit resolution, 1 MHz sampling rate, $V_{CC}/1.6$ reference voltage, timer triggers over event system
ADC channels	Channel 0: Temperature sensor, Channel 1: Light sensor
RTC32 configuration	An external 32 kHz crystal is the clock source. RTC ticks for every one minute, interrupt on overflow
Timer0 configuration	The clock source is a 12 MHz system clock, timer overflow interrupt occurs for every 1 ms
UART configuration	Asynchronous mode, 9600 baud rate, 8 data bits, 1 stop bit
USB configuration	48 MHz DFLL is clock source. USB 2.0 full speed mode, CDC Class
Event System configuration	Clock source is 12 MHz system clock, Ch-0 of the event system is used for timer0 interrupt
DMA configuration	Two DMA channels to transfer data from the two ADC channels to data buffer

Table 4-3 shows the operating conditions of the ATxmega256A3BU MCU and the list of peripherals configured for sensor node operation.

Table 4-4 shows the power consumption values of the ATxmega256A3BU MCU with different sleep period values.

Table 4-4. Power Consumption Values of ATxmega256A3BU with Different Sleep Periods

S. No.	Active Period [sec]	Active Current [mA]	Sleep Period [sec]	Sleep Current [μ A]	Average Current [μ A]
1	4	9.58	60	1.1	599.7
2	4	9.58	300	1.1	127.1
3	4	9.58	600	1.1	64.5
4	4	9.58	900	1.1	43.5
5	4	9.58	1800	1.1	22.3
6	4	9.58	3600	1.1	11.7

The power consumption of the RN2483 LoRa technology mote in active mode is 3.76 mA and in sleep mode it is 0.87 mA.

Note:

- In this application, the RN2483 LoRa transceiver module alone is enabled on the RN2483 LoRa technology mote board. The on-board LCD display, sensors, and PIC MCU are isolated from the sensor node circuitry.

4.6 Program and Data Memory Requirements

The application firmware program and data memory requirements for various compiler optimization levels are as given below.

Table 4-5. Application Memory Requirements with Various Compiler Optimizations

Optimization	Program Memory [in bytes]	Data Memory [in bytes]
-O0	67668	3309
-O1	34700	3292
-O2	34384	3308
-Os	30604	3308

4.7 Application Results

The LoRaWAN network with the XMEGA AU based sensor node, Microchip LoRa Gateway board, and The Things Network server has successfully demonstrated secure, bi-directional LoRa communication.

All the experiments with the wireless sensor node are carried out in Bengaluru, India. In highly dense urban areas that are surrounded by high rise buildings, the sensor node reliably communicated with LoRa gateway within a radius of approximately 1 km, where the gateway is positioned inside the building.

The sensor node realized using ATxmega256A3BU MCU and RN2483 LoRa transceiver module has demonstrated effective utilization of ADC, Event System controller, DMA controller, Timer, RTC, UART, and USB peripherals of ATxmega256A3BU MCU for IoT applications.

The sensor node has attained optimized power consumption due to the lower power consumption of ATxmega256A3BU MCU and RN2483 LoRa transceiver module, both in active and power down modes.

5. Low Power Considerations

For IoT applications, power consumption of an MCU is very crucial. Power consumption is greatly influenced by several factors like operating voltage, the frequency of operation, CPU active period, sleep mode, operating temperature, etc.

In practical applications, usually, the MCU is put into sleep mode while CPU is idle and not performing any active computations. This minimizes the MCU active period to a larger extent. Thereby, the MCU average power consumption reduces, which benefits to attain longer battery life.

Theoretically, average power consumption is calculated by the following formula:

$$\text{Average Power Consumption} = \frac{(\text{Active Current} \times \text{Active Period}) + (\text{Sleep Current} \times \text{Sleep Period})}{(\text{Active Period} + \text{Sleep Period})}$$

Here,

- Active current is the current consumed by the MCU during active period
- Active period is the duration for which the MCU is in active period
- Sleep current is the current consumed by the MCU in sleep mode
- Sleep period is the duration for which the MCU is in sleep mode

5.1 Optimizing MCU Power Consumption

The various factors which significantly influence the MCU power consumption are concisely discussed below along with appropriate recommendations and considerations to reduce the power consumption.

5.1.1 Operating Voltage

The power consumption is proportional to the square of the MCU's supply voltage. Increase in operating voltage increases device power consumption. Thus, the supply voltage should, therefore, be kept as low as possible.

Minimize power consumption by using supply voltage as low as possible.

5.1.2 Operating Frequency

The operating frequency of the MCU affects its power consumption. Increase in MCU operating frequency increases the device power consumption. Also, at lower operating frequencies the MCU active period increases and vice-versa.

The MCU power consumption will be a trade-off between active period and active current across different frequencies.

5.1.3 Clock Source Selection

For optimal power consumption, selecting appropriate system clock for MCUs is very much essential. Instead of using higher system clock and prescaling, switch to a different and slower clock source.

For example, to generate a 16 MHz system clock, using PLL and the 2 MHz RC oscillator as a reference is preferable over the 32 MHz RC oscillator with prescaling to 16 MHz. Usage of external clock sources may also be a good choice, especially if they are already available in the system.

The wake-up delay for the MCU depends on which clock source is used for the system clock. One way to reduce this delay is to switch between clock sources so that the MCU goes to sleep and wakes up with a fast responding clock source.

Minimize power consumption by switching clock sources rather than relying on prescaling alone for reducing clock rates.

5.1.4 Clock Prescaling

Many applications require the CPU operating as fast as possible to minimize the time spent in active mode. But, for certain applications, the CPU has to wait in Active or Idle mode for a fixed amount of time (e.g. serial communication), where it is better to reduce the system clock rate. In these cases, it is better to configure the appropriate CPU and peripheral clock frequencies using clock prescaling. If prescaling is done internally in several peripherals, power can be conserved by prescaling with the largest common factor as early as possible in the clock distribution chain.

Note: Since the prescaling also affects the CPU clock, it might not always be desirable to perform this common prescaling in Active mode because computations will take longer.

Minimize power consumption by effectively using prescaler especially when waiting in Active or Idle mode.

5.1.5 Active Mode Operation

In Active mode, i.e. when sleep modes are not used, the power consumption is proportional to the system clock frequency. This means that if sleep modes are not used, the device should be run at the lowest possible system clock frequency to minimize the power consumption.

Minimize power consumption by keeping the clock frequency as low as possible if sleep modes are not used.

5.1.6 Sleep Modes

If the MCU is continuously active, its power consumption increases. In most applications, it is desirable to minimize the power consumption, but not to reduce the system clock frequency. This is mainly to ensure fast processing and quick response of the system. In such applications, the use of sleep modes of the AVR XMEGA devices are to keep the device in a low-power state when there is nothing to process. The main principle is to run as fast as possible and sleep as much as possible. Running as fast as possible reduces the effect of static power consumption (i.e. independent of clock frequency), e.g.; due to non-volatile memory being enabled in Active mode.

Most commonly available sleep modes are Idle, Power-down, Power-save, Standby, and Extended Standby. Each sleep mode utilizes a different set of peripherals. Among these sleep modes, the Power-down mode is the best sleep mode since this mode shuts down almost all the peripherals resulting in good power reduction. Based on application power requirements, active and sleep duration of the device can be configured.

Keep the device in active mode for a defined shorter duration and in sleep mode for a longer duration.

5.1.7 Unused Peripherals

Enabling the unused peripherals in the active period of the MCU increases its power consumption. In the applications, it is recommended to shut down unused peripherals by disabling its clock signal. This is possible by using appropriate power reduction register (PRR) available in the device, which disconnects the individual peripherals from peripheral clock domain. In Power-save and Power-down modes the modules are stopped regardless of the state of the PRRs since the peripheral clock domain is disabled.

Disable unused peripherals in the active period of the MCU to reduce the power consumed.

5.1.8 Unused GPIOs

It is recommended to ensure that the unused general purpose I/O-pins have a defined level. Though most of the digital inputs are disabled in deep sleep modes, floating inputs should be avoided in other modes like Active mode, Power-save mode, and Idle mode to reduce the power consumption.

To minimize the power consumption, configure the unused general purpose I/O-pins as Input with internal pull-up enabled to ensure a defined level and for reduced power consumption.

5.1.9 Analog I/Os

If an I/O pin is connected to an analog source and is used for analog operations, then the digital input from these pins is not required. Disable the digital input buffer by use of the PINnCTRL registers for the individual ports to reduce the power consumption.

Disable digital input buffer on the pins that are connected to analog sources.

5.1.10 Brown-Out Detector (BOD)

The purpose of the Brown-Out Detector (BOD) is to actively monitor power supply voltage and to ensure that the MCU is not operating at a too low voltage. BOD is configured by BODLEVEL fuses. BOD can be configured separately for Active/Idle and sleep modes to track for indeterminate voltage levels. This allows for the BOD to be enabled only in Active and Idle mode. In AVR XMEGA devices, to reduce power it is recommended to disable BOD in Power-down sleep mode.

When BOD is disabled in software, it is turned OFF immediately after entering the sleep mode and automatically turned ON upon wake-up for monitoring supply voltage.

Disable BOD in sleep modes through application software to conserve power.

5.1.11 RTC Clock Source

The RTC is commonly used to wake up the device from sleep at periodic intervals. The RTC and its clock are active in Idle, Power-save and Extended standby sleep modes. Hence, RTC can be configured as wake-up source for these sleep modes.

For XMEGA family of devices, three different oscillators can be used to clock the RTC:

- External 32 kHz crystal
- Internal 32 kHz RC oscillator
- Internal 32 kHz Ultra Low Power (ULP) oscillator

In all these cases, a prescaled 1 kHz clock signal is available and should be used for optimized power consumption.

For the external 32 kHz crystal oscillator, a special low-power mode is also available (X32KLPM). It is recommended to use an external 32 kHz crystal with X32KLPM enabled. This gives lower power consumption than the ULP, yet greater accuracy than the internal RC oscillator.

For AVR XMEGA families with the battery backup module and 32-bit RTC, only the 32 kHz crystal oscillator may be used as the clock source. In these devices, the RTC is left running regardless of sleep.

Minimize power consumption by clocking the RTC at 1 kHz with an external crystal in low-power mode.

5.1.12 Watchdog

The watchdog is basically a timer with a separate clock source. It will, if enabled, contribute to the power consumption in sleep. The watchdog can only be clocked by the internal 32 kHz Ultra Low Power (ULP) oscillator, prescaled to 1 kHz.

To minimize the power consumption, disable the watchdog.

5.1.13 JTAG Interface and On-Chip Debugging

The JTAG interface is used for programming and debugging, but has no function during operation of an end-product. It is clocked and active during sleep if the On-chip Debugging (OCD) feature is enabled. The OCD and JTAG interface should therefore be disabled if it is not needed.

The OCD can be disabled in fuses, while the JTAG interface can be disabled both in fuses and in software. Disabling the JTAG in software ensures that the MCU can be reprogrammed because the JTAG interface is re-enabled upon RESET.

Alternatively, the PDI interface can be used for programming and debugging. In this case the JTAG interface may not be required at all, and may be disabled by fuses. The PDI interface also works in all sleep modes.

To minimize power consumption, disable the OCD and the JTAG interface.

5.1.14 Wake-Up Delays

When the MCU wakes up from sleep modes deeper than Idle (with the exception of the two standby modes), the system clock source must stabilize before the CPU starts to operate. This introduces a short delay, which depends on the selected clock source. If an internal RC oscillator or external clock is used, the start-up delay is 6 cycles. This is in addition to the RC oscillator start-up time. If the XTAL oscillator is used, the start-up delay is configurable. If frequency stability is wanted, it is recommended with start-up delays of 1,000 cycles for ceramic resonators and 16,000 cycles for quartz crystals respectively. This is in addition to the oscillator start-up time, which will depend on the resonator and load capacitances. In addition, there is a 13 cycle minimum delay before an Interrupt Service Routine (ISR) starts executing after wake-up. This is due to, e.g. the program counter being pushed on the stack and the jump to the ISR.

During the start-up delay, the power consumption is close to the power consumption in Idle, and thus represents “inefficient” power. If possible, it is therefore recommended to wake up as seldom as possible and rather “do more” every time the device wakes up.

To minimize the wake-up delay and conserve power, use an RC oscillator or external clock source, and wake up as seldom as possible.

5.1.15 Virtual Port Registers

To minimize the time spent in active mode, virtual port registers can be used. This allows for single-cycle access with I/O memory specific instructions like IN, OUT, and bit manipulation for the registers DIR, IN, OUT, and INTFLAGS for up to four I/O ports.

Use the Virtual Port registers for I/O port access to minimize power consumption.

For more detailed information on low-power consumption techniques to be followed, refer application note [AVR1010: Minimizing the power consumption of Atmel AVR XMEGA devices](#).

5.2 Code Optimization

To reduce power consumption, optimize the studio-based firmware for speed. The Atmel Studio7.0 IDE supports various levels of speed optimization. The highest level of speed optimization is possible with optimization level -O3.

Among the available GCC and IAR™ compiler tool-chains for AVR XMEGA devices, the IAR compiler provides the best code optimization compared to GCC. Hence, for power critical applications using IAR compiler is a more preferable option.

- *Select highest optimization level for Speed (-O3)*

- *Use the IAR compiler tool-chain for critical power requirements*

6. Conclusion

This application note illustrates the important features of AVR XMEGA AU MCUs, ideal for IoT applications. The combination of ADC, Event System, DMA, and Crypto engine peripherals along with other generic peripherals and picoPower technology of XMEGA AU MCUs offers the lowest system power consumption and secure communications. These highlighting features of XMEGA AU MCUs make it a choice of MCU for several battery-powered IoT applications.

The combination of the Microchip XMEGA AU MCU and RN LoRa transceiver module best suits for IoT, Machine-to-Machine (M2M) applications. It is also appropriate for applications such as health monitoring, smart metering, smart city, urban light control, object/environment monitoring, etc.

7. Reference

1. A Technical overview of LoRa and LoRaWAN - https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf
2. LoRaWAN Security - https://docs.wixstatic.com/ugd/eccc1a_cc44304714c14f80a6ce50fcf9fcee2a.pdf
3. Microchip RN2483 LoRa Technology Transceiver Module - <http://ww1.microchip.com/downloads/en/DeviceDoc/50002346C.pdf>
4. Microchip RN2483 LoRa Technology Mote - <http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=DM164138>
5. Microchip LoRa Technology Evaluation Kit - <http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=DV164140-1>
6. AVR1010: Minimizing the Power Consumption of Atmel AVR XMEGA Devices - <http://ww1.microchip.com/downloads/en/AppNotes/doc8267.pdf>

8. Revision History

Doc Rev.	Date	Comments
A	10/2017	Initial document revision.

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BeaconThings, BitCloud, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, KeeLoq logo, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, RightTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, chipKIT, chipKIT logo, CodeGuard, CryptoAuthentication, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PureSilicon, QMatrix, RightTouch logo, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2017, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-2228-0

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com	Asia Pacific Office Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon Hong Kong Tel: 852-2943-5100 Fax: 852-2401-3431 Australia - Sydney Tel: 61-2-9868-6733 Fax: 61-2-9868-6755 China - Beijing Tel: 86-10-8569-7000 Fax: 86-10-8528-2104 China - Chengdu Tel: 86-28-8665-5511 Fax: 86-28-8665-7889 China - Chongqing Tel: 86-23-8980-9588 Fax: 86-23-8980-9500 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 Fax: 86-571-8792-8116 China - Hong Kong SAR Tel: 852-2943-5100 Fax: 852-2401-3431 China - Nanjing Tel: 86-25-8473-2460 Fax: 86-25-8473-2470 China - Qingdao Tel: 86-532-8502-7355 Fax: 86-532-8502-7205 China - Shanghai Tel: 86-21-3326-8000 Fax: 86-21-3326-8021 China - Shenyang Tel: 86-24-2334-2829 Fax: 86-24-2334-2393 China - Shenzhen Tel: 86-755-8864-2200 Fax: 86-755-8203-1760 China - Wuhan Tel: 86-27-5980-5300 Fax: 86-27-5980-5118 China - Xian Tel: 86-29-8833-7252 Fax: 86-29-8833-7256	China - Xiamen Tel: 86-592-2388138 Fax: 86-592-2388130 China - Zhuhai Tel: 86-756-3210040 Fax: 86-756-3210049 India - Bangalore Tel: 91-80-3090-4444 Fax: 91-80-3090-4123 India - New Delhi Tel: 91-11-4160-8631 Fax: 91-11-4160-8632 India - Pune Tel: 91-20-3019-1500 Japan - Osaka Tel: 81-6-6152-7160 Fax: 81-6-6152-9310 Japan - Tokyo Tel: 81-3-6880-3770 Fax: 81-3-6880-3771 Korea - Daegu Tel: 82-53-744-4301 Fax: 82-53-744-4302 Korea - Seoul Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934 Malaysia - Kuala Lumpur Tel: 60-3-6201-9857 Fax: 60-3-6201-9859 Malaysia - Penang Tel: 60-4-227-8870 Fax: 60-4-227-4068 Philippines - Manila Tel: 63-2-634-9065 Fax: 63-2-634-9069 Singapore Tel: 65-6334-8870 Fax: 65-6334-8850 Taiwan - Hsin Chu Tel: 886-3-5778-366 Fax: 886-3-5770-955 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Fax: 886-2-2508-0102 Thailand - Bangkok Tel: 66-2-694-1351 Fax: 66-2-694-1350	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-7289-7561 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820