



ThingSpace

ST Micro – NimbeLink Cellular IoT Development Kit

Getting Started Guide

Version 1

Important – Please Read

© 2018 Verizon. All rights reserved.

Restricted and Controlled Distribution. Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of Verizon.

All information herein is subject to change without notice. The information provided was considered accurate at the time the document(s) were developed, and Verizon disclaims and makes no guaranty or warranty, express or implied, as to the accuracy or completeness of any information contained or referenced herein.

VERIZON DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

Verizon does not guarantee or warrant the availability of the network nor the compatibility of a network with any device, service or product. Verizon disclaims liability for any damages or losses of any nature whatsoever whether direct, indirect, special or consequential resulting from the use of or reliance on any information contained or referenced herein.

Technical data contained in this document may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

Verizon and Verizon logos are trademarks of Verizon. Other product and brand names may be trademarks or registered trademarks of their respective owners.

Contents

Overview	3
STM32K4 MCU	3
Skywire 4G LTE M Embedded Modem	3
ThingSpace	3
Getting Started	3
Connecting to the Verizon LTE CAT-M1 Network	4
Methods	4
Integrating the Modem Sample Code into your Application.....	5
Issues connecting to the Verizon Network	5
For More Information	5

Overview

The ThingSpace ST Micro – NimbeLink Cellular IoT Development Kit combines the [ST Micro NUCLEO-L476RG](#) microcontroller (MCU) development board and the [NimbeLink Skywire NL-M1DK](#) modem board development kit to enable IoT device developers to evaluate the capabilities of the STM32L4 MCU on the Verizon Wireless Cat-M1 network at a low cost. The kit delivers direct connection to ThingSpace and other cloud servers together with a variety of sensing features.

STM32L4 MCU

The STM32L4 MCU enables a wide diversity of applications by exploiting low-power communication, multiway sensing, and ARM Cortex -M4 core-based STM32L4 Series features. The support for Arduino Uno V3 connectivity provides unlimited expansion capabilities with a large choice of specialized add-on boards.

Skywire 4G LTE M Embedded Modem

The NimbeLink Skywire M1 development kit is specifically designed to help minimize the power requirement of the finished prototype. This makes the development kit ideal for proof-of-concept, field-testable prototyping of low-power M1 IoT devices. The development kit includes a Skywire 4G LTE M Embedded Modem and an adapter board that, among other things, combines the modem's separate cellular UART¹ and GPS UART into a single USB port for easy computer access to the modem. Skywire Cat M LTE modems are among the smallest in the industry.

ThingSpace

You can manage the ST Micro – NimbeLink Cellular IoT Development Kit using the ThingSpace device management platform whose features include:

- Diagnostics
- Security
- Cloud messaging capability

Getting Started

1. Purchase the ThingSpace ST Micro – NimbeLink Cellular IoT Development Kit from [Arrow.com](#)
2. Download and install the following tools from ST Micro onto your computer:

¹ Universal Asynchronous Receiver/Transmitter for serial communications

Item Description	Version	Source vendor	Purpose	Link
Atollic TrueSTUDIO	7.1.2	Atollic	Recommended integrated development environment (IDE) that includes the ST-LINK drivers	https://atollic.com/resources/download/windows/
STM32Cube MX	any	Atollic	Tools for adding the STM32CubeMX	http://gotland.atollic.com/resources/applicationnotes/CubeMX_installation_in_TrueSTUDIO.pdf

3. Install the appropriate IDE for the Nucleo-L476RG MCU board (we recommend Atollic TrueSTUDIO) specified in [Getting started with STM32 MCU Discovery Kits software development tools](#).
4. Set up the SkyWire modem board as specified in the [Skywire M1 Development Kit User Manual](#).

Connecting to the Verizon LTE CAT-M1 Network

Verizon provides sample code for you to connect the NUCLEO-L476RG and the SkyWire CAT-M1 module to the Verizon Network.

[Click here](#) to view the GitHub with the sample code file.

Methods

The file has 4 public routines:

```
bool modem_port_init(rw_port_tx_cb tx_cb, rw_port_rx_cb rx_cb)
```

Initializes the port used to talk to the modem (UART in this case) by setting the baud rate, flow control mechanisms etc. It accepts and stores two callbacks: transmit and receive.

```
void modem_port_tx(size_t sz, const uint8_t data[])
```

Transmit "sz" bytes from the buffer "data". AT commands can be passed through this routine

```
void modem_hw_init(void)
```

Initialize the modem hardware. In the NimbeLink case, it sets up a GPIO² on the MCU that's connected to the modem's reset line.

```
void modem_hw_reset(void)
```

Reset the modem through hardware means.

² General Purpose Input/Output

Callbacks:

Transmit callback:

```
typedef void (*rw_port_tx_cb)(size_t sz, bool err);
```

If "err" is unset, "sz" represents the number of bytes transmitted. "err" is set if the hardware port faced an error while transmitting the bytes.

Receive callback:

```
typedef void (*rw_port_rx_cb)(size_t sz, const uint8_t data[]);
```

Port hardware received "sz" bytes. The received bytes are stored in the buffer "data". The AT interpreter framework implements this callback, usually to store & parse the received bytes.

The function "uart_pin_init" is a helper function used during initialization and is private to the file, whereas "USART1_IRQHandler" is an L476-specific UART interrupt handler.

At the top of the file are macros that define the mapping of NimbeLink pins to the MCU's GPIOs and certain timing constants.

Integrating the Modem Sample Code into your Application

Use the following code in your application to call the script in the sample code that turns on the LTE modem and connects it to the network.

```
modem_hw_init();
modem_port_init(tx_cb, rx_cb);
modem_hw_reset();           // To force the modem into a known state
modem_tx(8, "at+cimi\r");   // Retrieve IMSI
// Ensure command was transmitted without errors using the transmit callback (tx_cb).
// Response bytes will be received through the receive callback (rx_cb).
// Parse received bytes.
```

Issues connecting to the Verizon Network

If you're having issues connecting to the Verizon network please see our [frequently asked questions page](#).

For More Information

Please contact your distributor for information or assistance.