Airgain

SKYWIRE CELLULAR MODEM

Software Developer's Guide

Airgain, Inc provides this documentation in support of its products for the internal use of its current and prospective customers. The publication of this document does not create any other right or license in any party to use and content in or referred to in this document and any modification or redistribution of this document is not permitted.

While efforts are made to ensure accuracy, typographical and other errors may exist in this document. Airgain, Inc reserves the right to modify or discontinue its products and to modify this and any other product documentation at any time.

All Airgain, Inc products are sold subject to its published Terms and Conditions, subject to any separate terms agreed with its customers. No warranty of any type is extended by publication of this documentation, including, but not limited to, implied warranties or merchantability, fitness for a particular purpose and non-infringement.

Table of Contents

1 Introduction

1.1 Scope

This document provides a high-level overview of how you can integrate the Skywire cellular modem into your product and application.

The following sections provide considerations for developing a device with cellular connectivity on different host controller platforms:

- **Getting Started:** Lists resources for getting started with your modem.
- **Microcontroller Platform:** Modem usage with a microcontroller platform.
- **Windows Platform:** Modem usage in a Windows environment.
- **Linux Platform:** Modem usage in a Linux environment.
- **Monitoring and Recovery:** General usage and monitoring principles for developing with the modem.

2 Getting Started

Be sure to read the following resources when you first receive your modem:

- **User Manual:** Each modem model has an associated user manual linked in its product page to help you get started. It shows how to ensure your modem and SIM card (if applicable) are working and registered on the network.

  You must complete all processes outlined in the manual; Specific application notes (described next) are written under the assumption that you've fully read the user manual.

- **Application Notes:** Once you've read the entire user manual and have ensured your modem is working properly, proceed to Airgain's application notes which cover how to:

  - Send SMS messages
  - Send HTTP using the built-in TCP stack
  - Connect the device to a computer to use as a data pipe

  Application notes outline a proof-of-concept and provide a walkthrough for completing a specific modem function.

  For a complete list of configuration options, see the modem's AT Command Manual (described next).

2.1 AT Command Manuals

Each modem has a specific AT command manual with descriptions of AT commands, examples, and capabilities. These manuals are usually provided by the module manufacturer, and are the most complete references for capabilities and functionality.

3 Microcontroller Platform

A key feature of the Skywire family of modems is the ability to connect to multiple host devices for use. The following subsections cover using the modem with a microcontroller platform:

- Connections
- AT Command Data Path Overview
- AT Command Requirements
- Data Protocol Path
- Firmware-Over-the-Air (FOTA) Updates
- SSL/TSL Stacks
- Hardware Flow Control
- The Skywire as a State Machine

3.1 Connections

All Skywire modems have a UART path for AT communications, which is the most common connection path used on a microcontroller platform. Most modems also have a USB communications path that your application can use.

There are two options for data communications on a microcontroller:

- Direct AT commands using socket dials
- Running a data protocol in an RTOS environment

These are covered in the sub sections below.

3.2 AT Command Data Path Overview

The modems adhere to the AT command standard for usage and configuration. Many also include standard commands from the Hayes and 3GPP standard AT command sets. Additionally, many vendors implement their own proprietary AT commands for various functions.

A socket dial is the primary network communication method for a microcontroller. Socket dials allow you to send data over TCP or UDP, and supports higher-level protocols such as HTTP or MQTT.

Many Skywire modems have built-in HTTP stacks. If your modem does not have a protocol that you need, you must implement, run, and manage an external stack for that protocol on your microcontroller or in your RTOS environment. Airgain does not provide support for external protocol stack implementations.

Note: Some modems have FTP clients as well for FTP downloads.

See your modem's AT command manual and application notes for more information on available protocols and functionality using AT commands.

3.3 AT Command Requirements

The following subsections describe processes you must perform when you issue AT commands:

- Handle Command Responses Sequentially
- Insert a Delay Between Commands

3.3.1 Handle Command Responses Sequentially

Your application must handle the response from one AT command before it invokes the next AT command to:

- ensure you don't issue AT commands too quickly.
- helps keep your host processor in sync with the modem.
- alert the host processor to an error response, which needs to be handled.

If your application doesn't handle a response before issuing a subsequent command, this can result in:

- AT commands being issued too quickly.
- an ERROR response being missed and not handled.
- your host processor becoming out of sync and issuing the wrong AT commands.

For more information on handling errors, see Section 6: Monitoring and Recovery.

3.3.2 Insert a Delay Between Commands

When issuing AT commands, you need to:

1. Send the AT command
2. Get the entire response back
3. Send the next AT command

However, if the response in Step 2 takes too long to receive, your code may send the next command in Step 3 before Step 2 completes. In this case, you may need to insert a delay between steps 2 and 3.

You can determine an optimal delay based on your setup, but Airgain suggests starting with 50ms and increasing or decreasing as needed.

3.4 Data Protocol Path

Just as you can run a point-to-point (PPP) stack on a desktop OS, you can implement the same protocol in a real-time operating system (RTOS) environment to send data. PPP is the most commonly implemented protocol in an RTOS if you're not using socket dials.

Airgain does not provide support for external protocol stack implementations in RTOS environments. However, we have application notes for running PPP on a Linux platform located on each modem's webpage. Additionally, Airgain maintains working PPP scripts on the following GitHub page:

https://github.com/NimbeLink/skywire-ppp-scripts

Note: While these application notes and PPP scripts are designed for Linux, they can serve as a foundation for integrating PPP in your RTOS environment.

3.5 Firmware-Over-the-Air (FOTA) Updates

Airgain recommends, and cellular carriers require, that you have a process to update both the firmware on your microcontroller and the firmware on your modem using firmware-over-the-air (FOTA) updates.

Note: Some carriers require FOTA updates for the modem to maintain carrier certifications.

FOTA-update capability for your microcontroller also enables you to remotely fix bugs and add new features.

Locate the web page that corresponds to your Skywire modem for the FOTA update procedure.

3.6 SSL/TSL Stacks

As security becomes more important in end-device applications, customers are encouraged to use secure data connection methods. Some modems have built in SSL/TLS stacks, and some do not.

For the modems that have the SSL/TLS stack built-in, see their associated application note and AT command manual for more information about supported protocol versions.

If your modem does not have a built-in SSL/TLS stack, or if your application requires a version of TLS that is not supported by your modem, you must use a stack on your host processor (e.g., Mbed TLS or wolfSSL).

Airgain does not provide support for implementing external SSL/TLS stacks.

3.7 Hardware Flow Control

Airgain recommends you implement hardware flow control into your microcontroller application, especially if you're sending large amounts of data and/or sending data very quickly.

Note: This data transition rate varies by microcontroller. If you noticed that bytes are dropping without using flow control, turn on hardware flow control to see if that resolves it.

Skywire modems are data circuit-terminating equipment (DCE), and need to be connected as follows:

- CTS on the modem connected to CTS on the host processor
- RTS on the modem connected to RTS on the host processor

If you're using UART as the primary communication method, Airgain recommends you thoroughly test your device in the prototype/development stage to see if flow control should be incorporated into the design before going to production.

3.8 The Skywire as a State Machine

It can be helpful to interact with and treat your Skywire modem as a state machine. Your host processor must know the modem's state, and take action if it's in an unknown state.

For example, if you cannot open a socket:

1. Verify that your package data protocol (PDP) context is open and you have an IP address.
2. If you don't have an IP address, verify that your APN is correct.
3. If your APN is correct, verify you are registered on the network and have a good signal.

For more considerations, see Section 6: Monitoring and Recovery.

4 Windows Platform

You can use the modem with both embedded and desktop versions of Windows.

The following subsections cover usage of the modem with Windows:

- Connections
- Data Protocol Path
- Firmware-Over-the-Air (FOTA) UpdatesSSL/TLS Stacks
- Hardware Flow Control
- Modem Monitoring

4.1 Connections

The most common data paths to set up a Skywire modem network interface on a Windows device include:

- PPP
- Network Configuration Management data (NCM)
- Mobile Broadband Interface Model (MBIM)

You can also use a direct AT communications path. For more information about AT command usage, see sections 3.3 and 3.4.

4.2 Data Protocol Path

There may be multiple data paths available for your specific modem.

- **PPP:** Set up a dial-up modem connection to use PPP. If supported by your modem, you can use a UART or USB communications path.
- **USB:** Additionally, your modem may have a USB data path available as well.

  Windows requires an installed USB driver before it can recognize the modem. Each modem has a different USB driver and the download files are provided in the documentation section for each specific modem.

- **NDIS/MBIM:** Some modems have native interfaces like NDIS or MBIM available, while others require setup and management in order to implement.

Note: The data paths available for your modem can vary based on your version of Windows. For example, MBIM might be supported in Windows 8 and 10, but not Windows 7.

Using a data path such as PPP or MBIM where Windows handles the device like any other network connection, lets you take advantage of the built-in protocol stacks on your device. Additionally, it may be easier to install external stacks (e.g., MQTT) onto Windows using a standard software installation procedure, compared to trying to integrate it into an RTOS.

Airgain does not provide any application notes specifically for setting up PPP on Windows. Application notes for running PPP on Linux are located on each modem's webpage. Additionally, Airgain maintains working PPP scripts on the following GitHub page:

https://github.com/NimbeLink/skywire-ppp-scripts

Note: Although these application notes and PPP scripts are designed for Linux, they can serve as a foundation for integrating PPP into your Windows environment.

For Windows communications options, visit the web page for your Skywire modem and review its AT command manual.

4.3 Firmware-Over-the-Air (FOTA) Updates

In addition to using FOTA via AT commands, Windows enables you to implement your own FOTA update process using the Windows update utilities provided by Microsoft.

For example, you can download the update file and use the command line tool to update your modem. This could be automated within your Windows device. See the firmware update application notes specific for your selected modem for more details.

For more information on FOTA and when it is required, see Section 3.5.

4.4 SSL/TLS Stacks

One advantage of using the Skywire modem as a network interface, is that you can use the built-in security stacks in Windows. This saves development time integrating external stacks.

4.5 Hardware Flow Control

If you plan to use the modem's UART communications path with your Windows device, Airgain recommends you implement hardware flow control. For more information on flow control, see Section 3.7.

If you are using the USB data path, you do not need to set up hardware flow control because it's included as part of the USB standard.

4.6 Modem Monitoring

Any sort of connection and network monitoring must be built into your end device to maintain a reliable connection. See Section 6 for information on monitoring the modem.

5 Linux Platform

Linux is the most commonly used and well-supported OS for Skywire modems on both embedded and desktop flavors of the OS.

The following sub sections cover usage of the modem in a Linux environment:

- Connections
- Data Protocol Path
- Firmware-Over-the-Air (FOTA) Updates
- SSL/TLS Stacks
- Hardware Flow Control
- Monitoring

5.1 Connections

Many customers use the UART data path and/or USB data path on both pre-built and custom Linux distributions. Most Skywire modems can use PPP, and many have USB data paths available as well.

You can also use AT commands to control the modem as described in sections 3.3 and 3.4.

For more information on the data paths available to you, see the web page for your Skywire modem.

5.2 Data Protocol Path

You may have multiple data paths available to you depending on your specific Skywire modem. PPP is available on Linux, and most Skywire modems that support PPP can use either UART or USB.

Some modems have native interfaces like USB Communication Device Class Ethernet Control Model (CDCECM) available, while others require some set up and management to implement.

Using a data path such as PPP or a USB data path enables Linux to handle the device like any other network connection so you can use the built-in protocol stacks on your device. Additionally, it may be easier to install external stacks (e.g., MQTT) onto Linux via a standard software installation procedure, instead of trying to integrate it into an RTOS.

Most Skywire modems have written guides for using PPP on Linux. Additionally, we host example PPP scripts as a starting point for your application in the following GitHub repo:

https://github.com/NimbeLink/skywire-ppp-scripts

Each modem's product page also provides guides on using the USB data path.

For Linux data paths, see your modem's AT command manual and application notes.

5.3 Firmware-Over-the-Air (FOTA) Updates

In addition to using FOTA via AT commands, Linux enables you to implement your own FOTA update process using the Linux update utilities provided with the modem. For example, you can download the update file and use the command line tool to update your modem. This could be automated within your Linux device.

For more information on FOTA and when it is required, see Section 3.5.

5.4 SSL/TLS Stacks

One advantage of using the Skyewire modem as a network interface, is the ability to use the built-in security stacks in Linux. This saves development time integrating external stacks.

5.5 Hardware Flow Control

If you plan on using the modem's UART communications path with your Linux device, Airgain recommends you implement hardware flow control. For more information on flow control, see Section 3.7.

If you are using the USB data path, you do not need to set up hardware flow control because it's included in the USB standard.

5.6 Monitoring

You must include connection and network monitoring into your end device to maintain a reliable connection.

See Section 6 for information on monitoring the modem.

6 Monitoring and Recovery

When designing a robust and reliable system using a cellular modem, additional software for the host processor, regardless of its type, must be written for modem management.

The following subsections cover considerations for this management software:

Note: Not all aspects covered are required for each customer's application, but most are necessary for consistent performance.

- Network Registration and Status
- Modem Recovery
- Modem Monitoring
- Logging
- Persistent Connections
- ERROR Results from AT Commands

6.1 Network Registration and Status

Check and monitor that the modem has a cellular signal and network connection.

Regardless of how you plan to integrate the modem into your system, you must register your application on the network and have a cellular signal, otherwise the cellular modem won't function.

Airgain recommends you integrate checks for the network status and cellular signal strength into your recovery software.

6.2 Modem Recovery

There may be times when you need to recover the modem from an unknown state to a known state. For example, if you are unable to open a socket, go back and check that the modem has an IP address. If it doesn't have an IP address, verify that the context is activated and that it is registered on the network.

You can also issue AT commands to perform a hardware reset of the modem. These commands vary by modem – see your modem's AT command manual for more information.

Finally, each Skywire modem has a reset pin which Airgain recommends you only use as a last resort. The pin removes the modem from the network, which can cause damage to the modem (e.g., firmware corruption that *bricks* the device) and could potentially corrupt the flash contents of the modem.

See your Skywire modem's datasheet for information about using the reset pin.

6.3 Modem Monitoring

If you are using the Skywire modem as a data connection for a host OS (e.g., as a PPP or CDC-ECM connection on Linux), you must use a network manager to monitor the connection and take action if it goes down to bring it back up. This can be a watchdog timer that you create, or you can modify an existing software project for your needs.

6.4 Logging

Airgain recommends that your application log output from the Skywire to assist with debugging efforts and general monitoring. This can involve:

- a constant log of AT command outputs stored on the host processor or sent to a centralized logging system, or
- a set of AT commands to check basic functionality when things are not working as expected.

Note: Airgain Support is limited in assisting customers who have devices in the field that are no longer responding and transmitting logs. Additionally, it's difficult for the designers of the application to perform debugging without proper logging and reporting. In these situations it may be necessary to bring the device in from the field for diagnostics.

6.5 Persistent Connections

If a persistent connection is required by your application, you must implement monitoring and *keep-alive* software to ensure you have a persistent connection. Carriers remove idle devices from their networks to prioritize active devices, so your host processor must check and re-establish the connection if this happens.

6.6 ERROR Results from AT Commands

There may be times when an AT command returns `ERROR` as a response and your host processor must be able to handle these scenarios. Airgain recommends retrying the command several times.

Note: The general number of times to retry, varies depending on the operation you're performing.

If the error persists, step through other possible causes to verify functionality.