SILICON LABS

AN844

ASHv3-UART Host Interfacing Guide

This document describes how to set up and test UART communication between a host and Network Co-Processor (NCP) using ASHv3-UART. It assumes that you have a Raspberry Pi, USB cable (for UART communication), and a development board. This document applies to Silicon Labs Thread version 1.0 or later.

Contents

1 Software Overview

A Network Co-Processor (NCP) runs the Ember Thread stack and is controlled by the host processor through ASHv3-UART commands. The NCP must be a chip in the Ember EM35XX family. Two applications are needed for testing: ip-modem-app and ash-v3-test-app.

1.1 ip-modem-app

The image ip-modem-app.s37 is loaded onto your NCP. Table 1 summarizes the four versions of this application and their file locations.

Table 1. ip-modem-app.s37 images

Description	**File location**
357 with NULL bootloader	build/ip-modem-app-cortexm3-iar-em357-em3xx-dev0680-null_btl/ip-modem-app.s37
357 with no built-in bootloader	build/ip-modem-app-cortexm3-iar-em357-em3xx-dev0680/ip-modem-app.s37
3588 with NULL bootloader	build/ip-modem-app-cortexm3-iar-em3588-em3xx-dev0680-null_btl/ip-modem-app.s37
3588 with no built-in bootloader	build/ip-modem-app-cortexm3-iar-em3588-em3xx-dev0680/ip-modem-app.s37

1.2 ash-v3-test-app

ash-v3-test-app runs on your host and provides tests to verify that your UART + ASHv3 implementation is working. This test program is much simpler than a full-fledged gateway sample application. As a result, it can more efficiently pinpoint ASHv3-UART interface problems.

ash-v3-test-app has two modes of execution—interactive and non-interactive. In interactive mode, an interactive console operates the application. In non-interactive mode, the application is operated by command-line arguments. By default, ash-v3-test-app runs in interactive mode. Invoking ash-v3-test-app with the arguments `--test-echo`, `--test-bootstrap` or `--test-xon-xoff` forces it to run in non-interactive mode. (See section 2, **Testing** for a detailed explanation of these arguments.)

Note: ash-v3-test-app does not enable any wireless functionality because its purpose is to validate proper operation of the ASHv3 interface between the host and the NCP.

1.3 ash-v3-test-app Command Line Options

ash-v3-test-app takes the following arguments:

`--uart <uart_file>`: Specify the UART file, such as /dev/ttyUSB0. This option is required.

`--test-echo <echo_string>`: Run an echo test. For more information, see section 2.1, **Bootstrap Testing.**

`--test-bootstrap`: Run a bootstrap test. For more information, see section 2.2, **Echo Testing**.

`--test-xon-xoff`: Run an XON/XOFF test. For more information, see section 2.3, **XON/XOFF Testing**.

`--verbose`: Display all TX'd and RX'd ASHv3 packets.

2 Testing

ash-v3-test-app provides tests to verify that your UART + ASHv3 implementation is working properly. It verifies that an ASHv3 handshake can be performed and it also provides an ‘echo’ command. To build the application, execute the following command in the base release directory:

```
make –f app/ip-ncp/ash-v3-test-app.mak
```

Before executing ash-v3-test-app, identify your USB TTY driver device. The TTY driver device is usually `/dev/ttyUSB0`.

2.1 Bootstrap Testing

The bootstrap test verifies that ASHv3 can perform an initial handshake between the host and NCP. Invoke ash-v3-test-app with the `-u` and `--test-bootstrap` arguments as follows:

```
sudo build/ash-v3-test-app-unix-host/ash-v3-test-app -u /dev/ttyUSB0 --test-bootstrap
```

The text ‘ASHv3 is up’ will be displayed when ASHv3 successfully performs an initialization handshake. If a handshake cannot be performed, the application displays ‘Failure’.

2.2 Echo Testing

The echo test sends a string from the host to the NCP, and the NCP then sends the string back to the host. The host verifies that the received string matches what was sent. Invoke ash-v3-test-app with the –u and `--test-echo` arguments as follows:

```
sudo build/ash-v3-test-app-unix-host/ash-v3-test-app -u /dev/ttyUSB0 --test-echo
“this is my string”
```

If the host receives a correctly formatted string back from the NCP, it will print ‘Success’. Otherwise, it will print ‘Failure’.

2.3 XON/XOFF Testing

The NCP implements a one-way version of XON/XOFF as follows:

- The NCP tells the host to stop sending it data by sending an XOFF.
- The NCP sends a series of XONs when it can accept serial data again.
- The host does not have the same ability; the NCP ignores any XON or XOFF bytes sent by the host.

The NCP sends an XOFF when its receive buffer fills up to a threshold. It sends a series of five XONs, with 100 milliseconds between each, when the buffer drains down to a second lower threshold. The NCP also sends the same sequence of XONs after it is reset. These additional XONs prevent the UART from hanging if an XON is lost or corrupted by noise, or if another byte sent by the NCP is corrupted into an XOFF. XON/XOFF flow control is enabled in the application when it is compiled with this global #define:

```
EMBER_APPLICATION_SUPPORTS_SOFTWARE_FLOW_CONTROL
```

The XON/XOFF test verifies that the ip-modem-app sends XONs when either of its RX buffers is full, and sends a series of XONs after it services its RX buffers. Invoke ash-v3-test-app with the –u and `--test-xon-xoff` arguments as follows:

```
sudo build/ash-v3-test-app-unix-host/ash-v3-test-app -u /dev/ttyUSB0 --test-xon-xoff
```

ash-v3-test-app displays ‘Passed’ upon success, and prints an error upon failure.

3 Debugging

If the steps in section 2, **Testing** fail, try these debugging tips:

- Some problems can be debugged by viewing the ASH frames that are sent between the host and NCP. To do this, invoke ash-v3-test-app with the `--verbose` argument:

  ```
  sudo build/ash-v3-test-app-unix-host/ash-v3-test-app -u /dev/ttyUSB0 --verbose
  ```

- Verify that a tty device exists in `/dev`, such as `/dev/ttyUSB0`. It might have a different name, such as `/dev/ttyUSB1`.
- If ash-v3-test-app is completely unable to communicate with the NCP, attach a logic analyzer to the UART TX and RX lines, and verify that traffic is being sent.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

For additional information please visit the Silicon Labs Technical Support page:
http://www.silabs.com/support/Pages/default.aspx