Pocket A^{2}B® Bus Monitor User Guide

1.2 April 2025

This document is for information and instruction purposes. Flextech AKT LLC reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Flextech AKT to determine whether any changes have been made.

The terms and conditions governing the sale and licensing of Flextech AKT products are set forth in written agreements between Flextech AKT and its customers. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Flextech AKT whatsoever.

FLEXTECH AKT MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

FLEXTECH AKT SHALL NOT BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF FLEXTECH AKT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

U.S. GOVERNMENT LICENSE RIGHTS: The software and documentation were developed entirely at private expense and are commercial computer software and commercial computer software documentation within the meaning of the applicable acquisition regulations. Accordingly, pursuant to FAR 48 CFR 12.212 and DFARS 48 CFR 227.7202, use, duplication and disclosure by or for the U.S. Government or a U.S. Government subcontractor is subject solely to the terms and conditions set forth in the license agreement provided with the software, except for provisions which are contrary to applicable mandatory federal laws.

TRADEMARKS: The trademarks, logos and service marks ("Marks") used herein are the property of Flextech AKT LLC or other parties. No one is permitted to use these Marks without the prior written consent of Flextech AKT or the owner of the Mark, as applicable. The use herein of a third- party Mark is not an attempt to indicate Flextech AKT as a source of a product, but is intended to indicate a product from, or associated with, a particular third party.

Flextech AKT LLC
24613 S 220th St.
Queen Creek, AZ 85142

Website: www.flextechakt.com

Contents

Contents **3**
Chapter 1. Pocket A2B Bus Monitor Overview **7**
Capabilities 7
Delivery Parts List 8
A2B Network System Components 9
Chapter 2. Interfaces **11**
Pocket A2B Ports 11
Front View 11
Back View 11
Side View 11
Bottom View 11
LEDs 12
Status LED 12
USB LED 12
I/O LED (Input / Output) 12
Chapter 3. Graphical User Interface (GUI) **13**
Initial Setup 13
Install the Graphical User Interface (GUI) 13
Graphical User Interface (GUI) Tour 14
Info Pane 15
Event Trace 15
Chapter 4. Working with the Pocket A2B Bus Monitor **16**
Initial Setup 17
Installing the Flextech AKT USB device driver 17
Insert the Pocket Bus Monitor into an A2B Network 18
Position 1: Between the Main Node and the First Downstream Sub Node 18
Position 2: Between Two Sub Nodes 18
General Notes 19
Constructing a Test Network (optional) 20
Connecting the Bus Monitor to your Host PC 21
Prerequisites 21
Steps to Connect 21
Programming the DATCTL Register 23
Starting a Capture 24
Routing Audio Channels from A2B to USB 25

Routing Audio Using the Bus Monitor GUI 25
Routing Table 25
Recording Audio 27
Bus Monitor GUI 27
Digital Audio Workstation 28
Listening to Audio 29
Recording a Trace 29
Importing a Trace 29
Reading an Event Trace 30
I2C Transactions 31
SPI Transactions 31
Using the Pocket Bus Monitor Command Line 32
Download and Install a Terminal Emulator Utility (Tera Term) 32
Hookup 32
Using the Command Line 33
General Commands 33
File Management 34
File Commands 34
File Transfers 35
SDCARD 35
Edit Command 35
X/YMODEM 36
USB Sound Card Re-Configuration 36
Bus Monitor Commands 37
Connecting to a live A2B Bus 38
Step 1: Discover the A2B network with the Bus Monitor Attached 38
Step 2: Dump the settings to a file 38
Step 3: Connect to a live bus 38
Step 4: Begin monitoring 39
Qwiic Peripheral Configuration 40
I2C Commands 40
Qwiic Examples 40
Audio 41
General 41
A2B 41
USB Audio 41
WAV Files 42
Circular Audio Buffer (CBUF) 42
VU Meters 42
ASIO 43
Audio Routing 43
AKT Automation 45
Command Scripts 45

Running commands at startup....46
AKT Automation with Lua....46
Bus Monitor GUI Plugins....47
Pre-defined Constants....48
Pre-defined Callbacks....48
plugin_trace()....48
Parameters....48
Return Values....49
Required Plugin Methods....49
plugin_loaded()....49
plugin_unloaded()....49
Parameters....49
Return Values....49
plugin_start()....49
plugin_stop()....49
Parameters....50
Return Values....50
plugin_event()....50
Parameters....50
Return Values....50
A2B Bus Monitor Plugin Enumerations and Events....50
BM_GUI_PLUGIN....50
BM_I2C_TYPES....51
BM_I2C_CONDITIONS....51
BM_EVENTS....51
Updating the Pocket Bus Monitor's Firmware....54
Methods to Update the Firmware....54
Command-line Initiated AKT Flasher Over USB....54
Command-line via SD card file....54
Bootloader initiated AKT Flasher over USB....54
Chapter 5. Connectors & Cables....56
Included Connectors....56
Bus Monitor Connections....56
Analog Devices Evaluation Platform Connectors....57
Fabricating Custom Cables....57
Chapter 6. Specifications....59
Environmental....59
Ordering Information....59
Chapter 7. Troubleshooting....60

Chapter 1. Pocket A²B Bus Monitor Overview

Observe, capture, and analyze the data flowing on a live A²B network with the Pocket A²B Bus Monitor.

The Pocket A²B Bus Monitor system consists of a hardware device and a host-based Graphical User Interface (GUI). The hardware device enables non-intrusive, real-time monitoring of the traffic on an A²B bus. The GUI controls the device and provides tools for capturing and inspecting data. The Pocket A²B Bus Monitor can passively log data between any two nodes on an A²B bus

NOTE: *If the DATCTL register on the main node transceiver is not properly configured, the A²B Bus Monitor won't be able to capture audio from the bus. For more information, see* *Programming the DATCTL Register.*

Capabilities

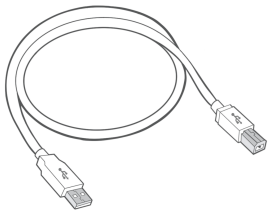
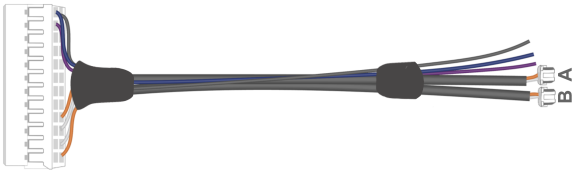
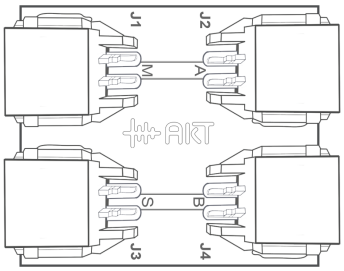
Use the Pocket A²B Bus Monitor as a bench tool, validation tool, or in-vehicle test tool.

The Pocket A²B Bus Monitor can:

- Capture data from the A²B bus **non-intrusively.**
- **Continuously** log live A²B bus data over USB.
- Record A²B events for timing **analysis.**
- **Integrate** into larger test and production frameworks.
- **AKT Automation**

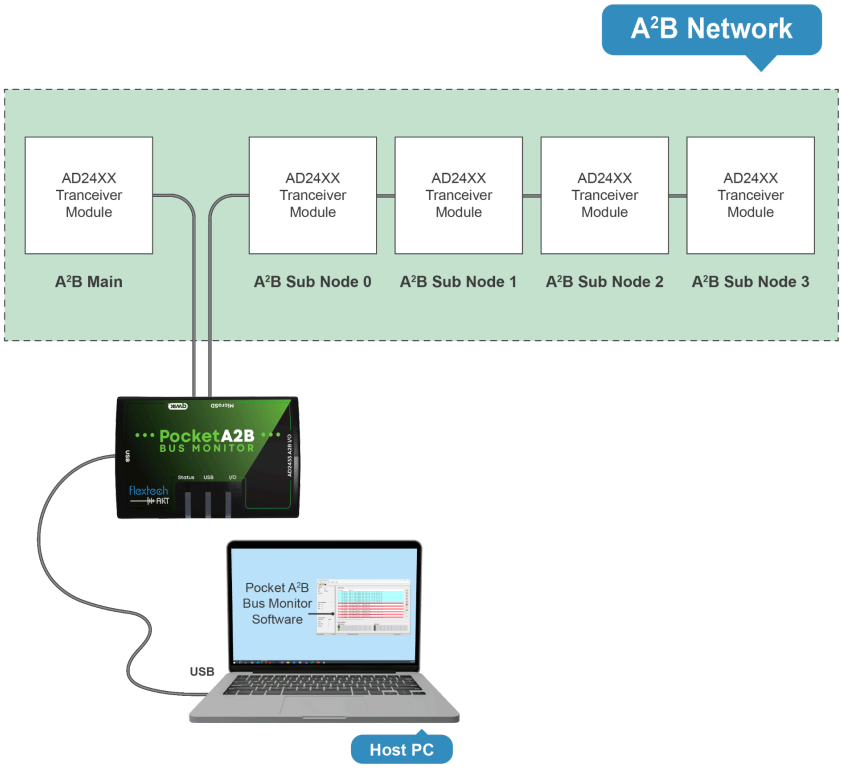
Delivery Parts List

The following parts are included and shipped with the Pocket A²B Bus Monitor:

1 A²B Pocket Bus Monitor device	
1 USB cable, for connecting the Bus Monitor to your host PC	
1 Header Connector with cross-over cable for connecting the Bus Monitor to the A²B network, A and B bus, and optional GPIO pins.	
1 Crossover Board	
1 SanDisk Industrial microSD Card	

A²B Network System Components

The following diagram shows how to integrate the Pocket A²B Bus Monitor into an A²B network.
For maximum visibility, insert the device between the main node and the first sub node.



Host PC	System that runs the Pocket A²B Bus Monitor GUI software. In this setup, the Bus Monitor device is connected to Host PC over USB.
GUI software	App for configuring the A²B Bus Monitor device, observing live data from the A²B bus, and recording and analyzing traces. The app runs on the host PC and is supported on both Windows and Linux platforms.
USB	Connection used to configure and control the device. It can also be used to view and capture data from the A²B bus.
Pocket A²B Bus Monitor	Hardware device that non-intrusively captures data from a live A²B bus and streams it in real-time over USB to the Pocket A²B Bus Monitor GUI software.
A²B network	High bandwidth (50 Mbps) digital bus that transports I2S audio, I²C control data, along with clock and power over a single, 2-wire, unshielded, twisted pair cable. A²B networks consist of a single main node and multiple sub nodes in a daisy-chained configuration.
Main node	Main node must properly configure its transceiver to grant permission to the Bus Monitor to capture audio data from the bus. For more information, see Programming the DATCTL Register. If you use the **A²B Bridge** device to simulate the main node, you can dynamically configure the network definition to properly configure the DATCTL register. The **A²B Bridge**, however, is NOT a required component. The Bus Monitor can interoperate with any main node.

Chapter 2. Interfaces

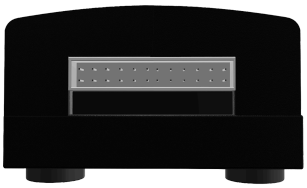
This chapter introduces the ports and LEDs found on the device.

Pocket A²B Ports

Front View

1 USB 2.0 high speed type B interface. Main connection to PC.

NOTE: *The Pocket Bus Monitor is powered over the USB connection.*



Back View

24 Pin Multi I/O Connector. Includes A-side and B-side A²B connections; two independent, bi-directional general purpose 3.3V logic-level GPIO lines with interrupt capability; access to the A²B transceiver I²C and SPI data pins.

Side View

SD Card Slot for file storage and **AKT Automation**.

[Sparkfun Qwiic Expansion Connector]

Bottom View

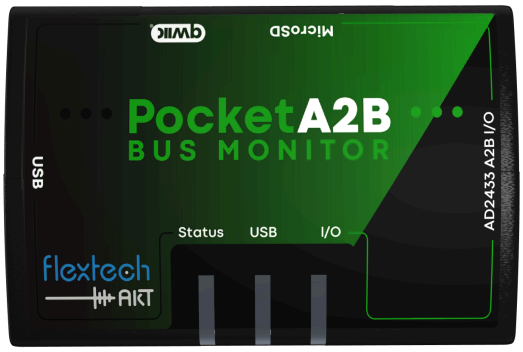
Safe-Boot Recovery button

HW ID and **Serial Number**

For further details on cables and connectors, see Chapter 5. Connectors & Cables

LEDs

LEDs on the top indicate the status of the device when it's powered on.



Each LED reports status two ways:

- **Background color** – Main display color.
- **Blink color** — Cycles on and off, temporarily overriding the background color.

Status LED

LED State	Color	Description
Fast Flash	Green	System OK
	Red	Invalid / missing feature key file. See Troubleshooting.
Slow Blink	Yellow	Bootloader mode active

USB LED

LED State	Color	Description
Blink	Blue	Activity on the USB bus
Steady	Blue	Active USB Audio stream

I/O LED (Input / Output)

LED State	Color	Description
Steady	Green	A^2B Bias OK
	Red	A^2B Bias Reversed. See Troubleshooting.

Chapter 3. Graphical User Interface (GUI)

The **Graphical User Interface (GUI)** interfaces with the Pocket A^{2}B Bus Monitor to capture and analyze data from the A^{2}B bus.

There are two classes of traffic on the A^{2}B bus: **control data** and **audio data**. The GUI displays the control data along with VU meters for audio. Real-time audio is transferred through the Pocket A^{2}B Bus Monitor USB sound card.

With the GUI Software You Can:

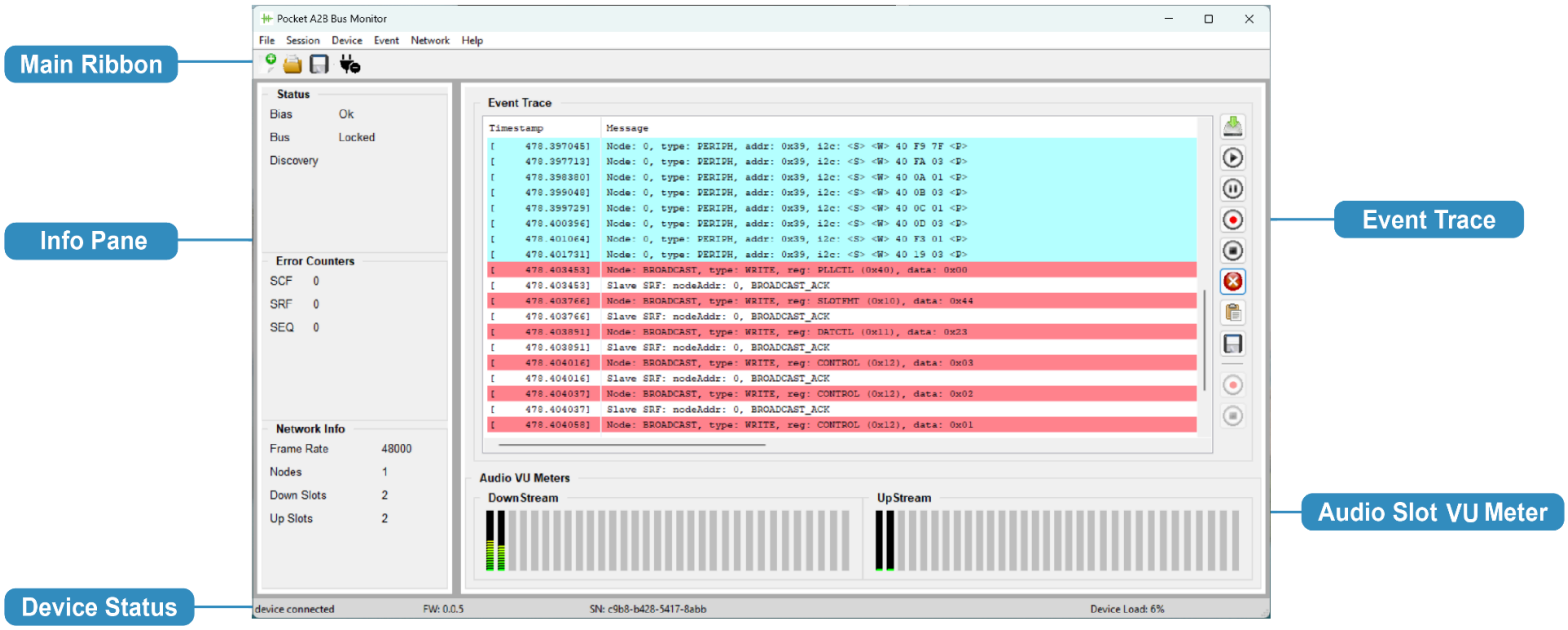
- Configure the device including USB audio routing.
- Record audio to WAV files over USB.
- Visualize control data on the A^{2}B bus.
- Visualize audio data on the A^{2}B bus.
- Store captures to a file.
- Analyze traces.

Initial Setup

Install the Graphical User Interface (GUI)

Unzip the GUI zip file. Double click the installer executable to start the installation process. Follow the on-screen instructions to complete the installation.

Graphical User Interface (GUI) Tour



Main Ribbon

- **Sessions** — Stores information about the app's layout (window size, window position) and the device configuration. Use session files to quickly restore preferred settings when you start the Bus Monitor GUI. You can create and save sessions to files on your host PC.
- **Connections** — Connect to the Bus Monitor device via USB.
- **Record** — Record multichannel A^{2}B audio over USB.

Info Pane — General information about the A^{2}B network.

Device Status Bar — Information about the A^{2}B Bus Monitor, including its hardware ID, firmware version and CPU load.

Event Trace Pane — Real-time stream of the control traffic on the A^{2}B bus. Use the controls on the right side of the pane to record captures.

Audio VU Slot Meters Pane — Real-time view of the levels of all available A^{2}B audio slots. VU meters provide a quick visual indicator that audio is present on a slot. The Bus Monitor automatically determines the active A^{2}B audio slots. Audio slots reserved for SPI over distance are indicated in Blue.

Info Pane

The **Network** heading shows:

- Bias Status, Bus Lock, and Discovery Status
- Detected sample / frame rate (Fs).
- Number of downstream nodes relative to the Bus Monitor's position on the network.
- Number of downstream slots relative to the Bus Monitor's position on the network.
- Number of upstream slots relative to the Bus Monitor's position on the network.

The Bus Monitor also tracks:

- A count of Synchronization Control Frame (SCF) errors.
- A count of Synchronization Response Frame (SRF) errors.
- A count of sequence (SEQ) errors seen so far. A^2B frames have a simple 2-bit sequence number. The Bus Monitor reports missing sequence numbers.

Event Trace

A timestamp is recorded for every event logged in the trace window. Timestamps have microsecond resolution. Time starts when the Bus Monitor powers up.

Start, pause, stop, record, save, and clipboard buttons can be used to control visible and saved events.

Chapter 4. Working with the Pocket A^{2}B Bus Monitor

A typical workflow is covered in this chapter:

- Connect the Pocket Bus Monitor to the A^{2}B network between the main node and the first downstream sub node.
- Connect the Bus Monitor to the host PC using the supplied USB cable.
- Start the Pocket Bus Monitor GUI on the host PC.
- Optionally configure audio routes to route A^{2}B audio over USB.
- Perform an A^{2}B bus discovery. Discovery is the process by which the main node initializes the A^{2}B network and all the nodes on it.
- Capture and store network data.
- Analyze your captures.
- Using the command line

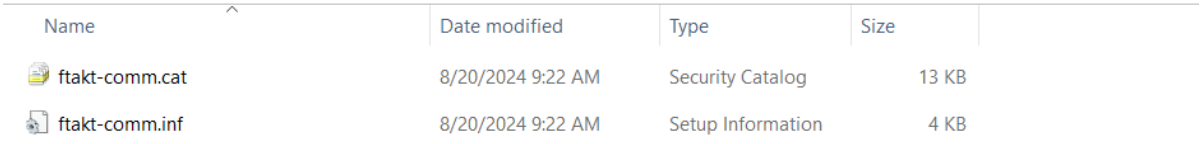
NOTE: *The DATCTL register in the main node must be configured to capture audio or SPI over distance data. For more information, see* *Setting the DATCTL Register*

Initial Setup

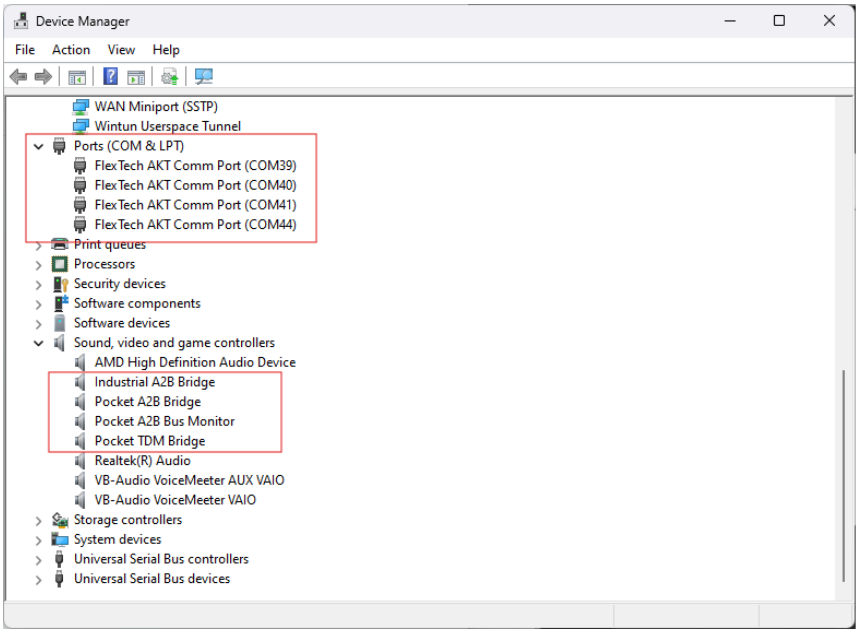
Installing the Flextech AKT USB device driver

To install the USB serial device driver, perform the following steps:

1. Download the ftakt-comm driver from www.flextechakt.com



2. Unzip the supplied zip file
3. Right Click on the *ftakt-comm.inf setup file* and select *install.*
4. After installation you will see the FlexTech AKT Comm Port and sound card listed on Windows Device Manager under Ports (Com & LPT).



Insert the Pocket Bus Monitor into an A²B Network

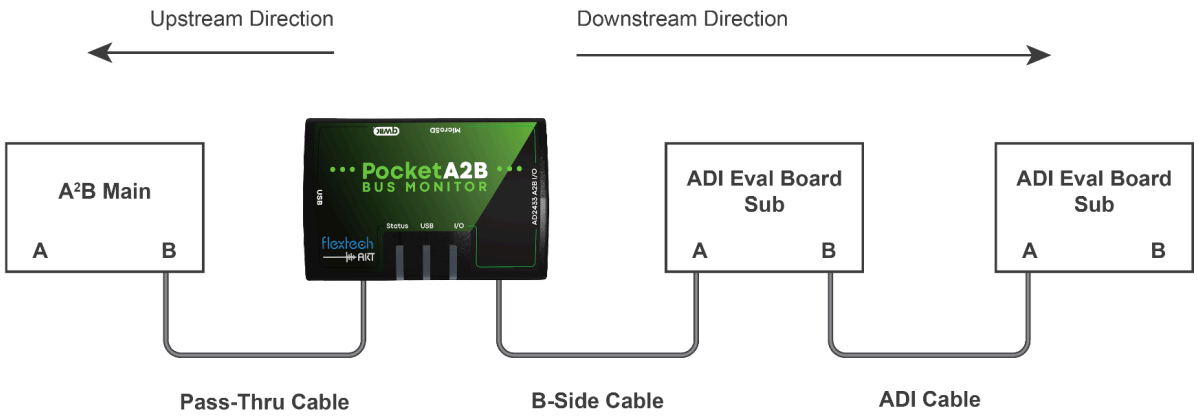
The following sections describe two positions the Bus Monitor can be connected in an A²B network and procedures for establishing the connection.

> For further details on cables and connectors, see Chapter 5. Connectors & Cables

Position 1: Between the Main Node and the First Downstream Sub Node

The most common way to position the Bus Monitor is between the main node and the first downstream sub node. From this perspective, the Bus Monitor can **see all data on the network**, except sub-to-sub audio traffic. From this position the A²B Bus Monitor can capture the entire network discovery sequence.

The following diagram shows the connections for this setup:



Position 1 Connection Setup:

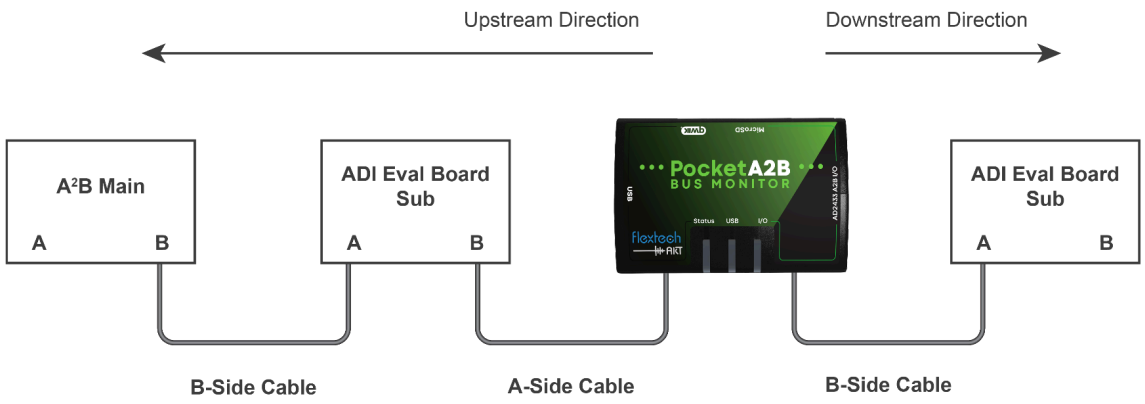
1. Connect the **B-side** of the main node (or A²B Pocket Bridge) **to the A-side** of the Bus Monitor.
2. Connect the **B-side** of the Bus Monitor to the **A-side** of the first downstream sub node.

Position 2: Between Two Sub Nodes

From this perspective, the Bus Monitor has a narrowed view of the network. The Bus Monitor can only monitor traffic destined for nodes downstream from this position. Use this position when you need to

capture sub-to-sub audio or SPI data. Sub-to-sub audio is invisible to the main node and cannot be observed when the Bus Monitor is positioned immediately after the main node.

The following diagram shows the connections for this setup:



Position 2 Connection Setup:

1. Connect the **B-side** of the upstream node (or A^{2}B Pocket Bridge) to the **A-side** of the Bus Monitor.

2. Connect the **B-side** of the Bus Monitor to the **A-side** of the downstream sub node.

General Notes

NOTE: *If you are using the Flextech AKT Pocket Bridge as a Main node, and Analog Devices evaluation boards as sub nodes, use the supplied DuraClik crossover board to insert the bus monitor between nodes. Connect the Bus Monitor "A" and "B" to the crossover headers labeled "A" and "B". Connect the upstream node to the "M" header and downstream node to the "S" header.*

NOTE: *If the I/O LED illuminates Red (Bias Reversed), but the bus discovers normally, swap the A-Side and B-Side connections if using the DuraClik crossover board. Reverse the polarity of both the A-Side and B-Side if using custom cables.*

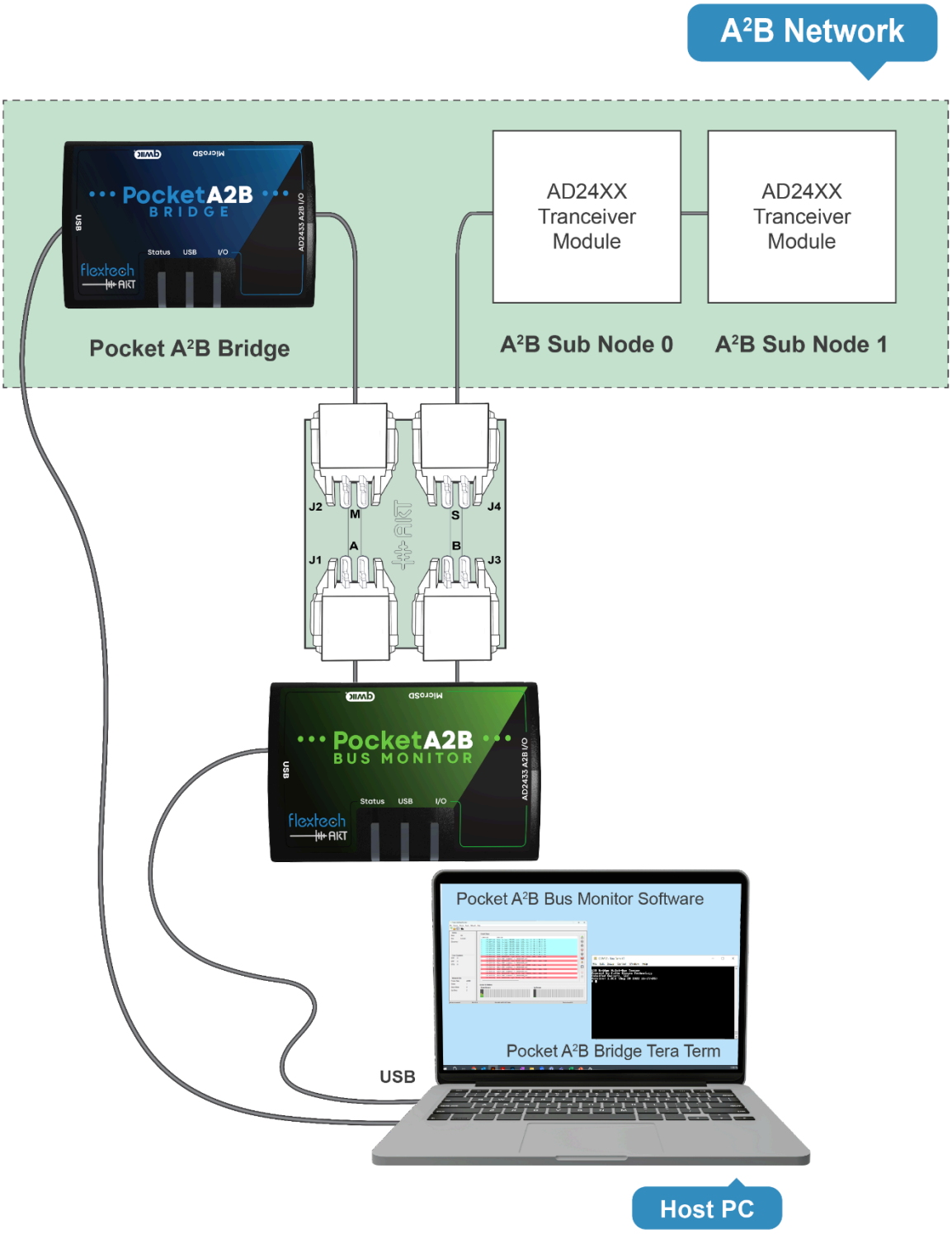
NOTE: *If the I/O LED illuminates Red (Bias Reversed), and bus discovery fails, reverse the polarity of only the A-Side.*

Constructing a Test Network (optional)

Consider using a Pocket A²B Bridge as the main node and an Analog Devices evaluation board as a Sub node to quickly test the Pocket Bus Monitor.

NOTE: *Use of the Pocket A²B Bridge is not a requirement. For further instructions on setting up the A²B networking using the Pocket A²B Bridge, see the Pocket A²B Bridge User Guide.*

Example Test Network Configuration using Pocket Bridge and Pocket Bus Monitor:



Connecting the Bus Monitor to your Host PC

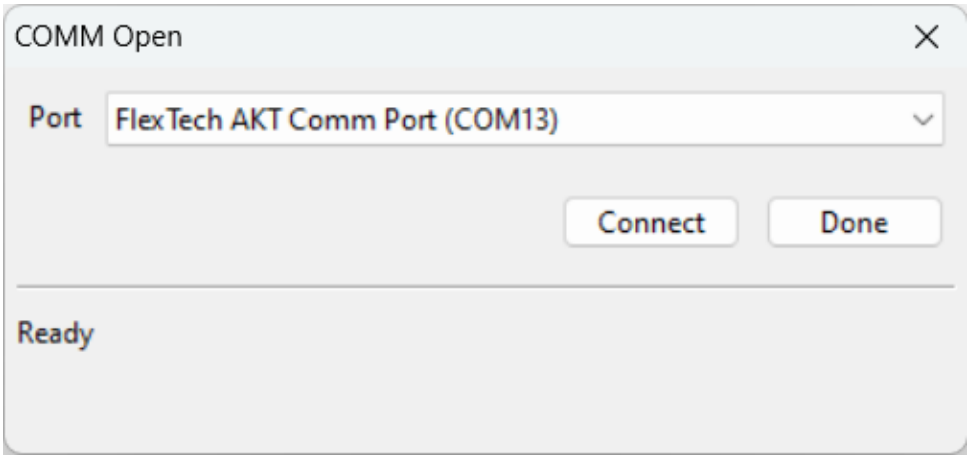
Prerequisites

- A dedicated High Speed USB 2.0 or faster USB port for the Bus Monitor.
- AKT Pocket A²B Bus Monitor GUI installed.

Steps to Connect

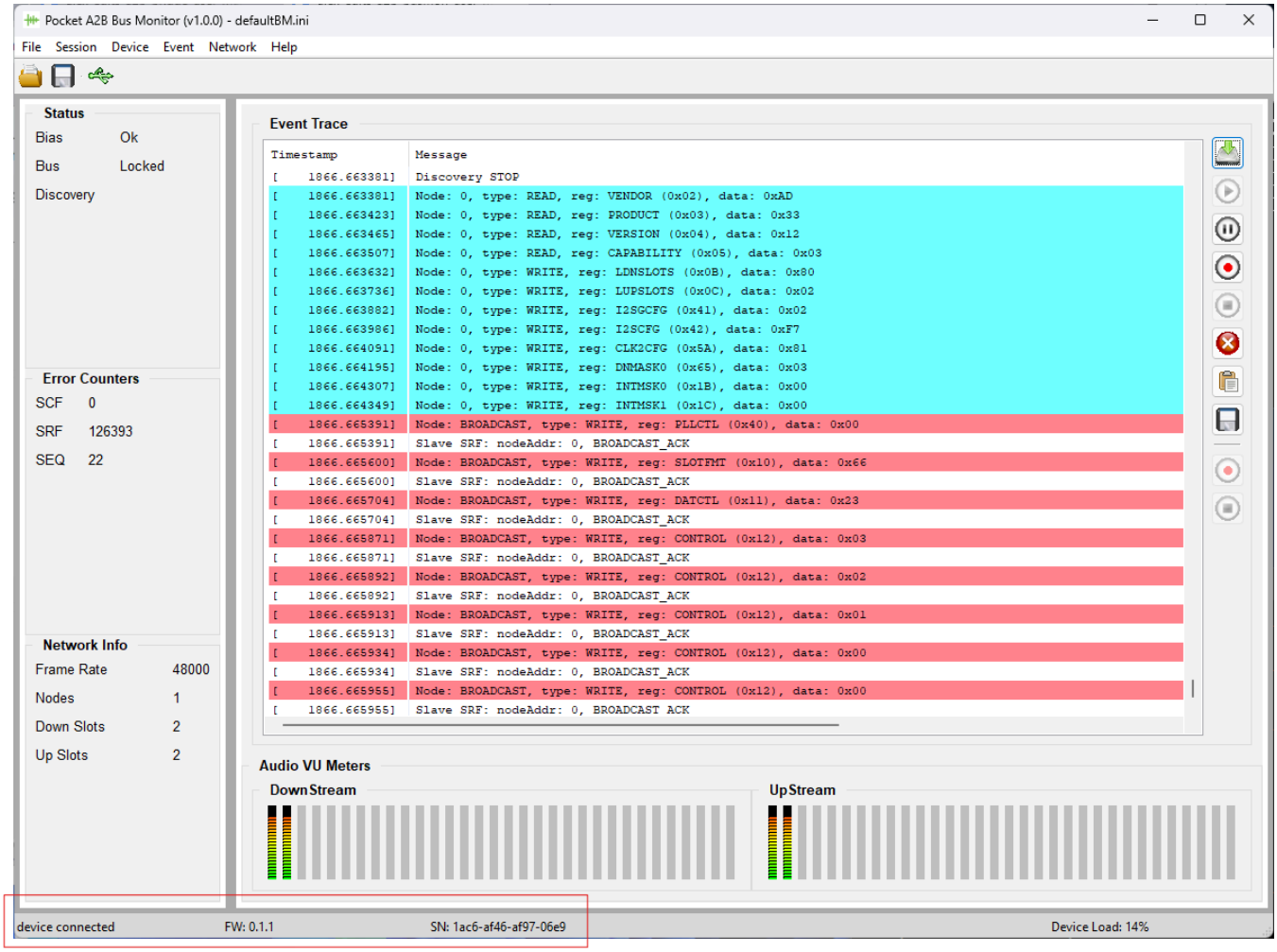
1. Connect the supplied USB cable from the Bus Monitor to the host PC USB port.
2. Start the Pocket A²B Bus Monitor GUI on the host PC. 

3. On the main ribbon bar click the USB icon to connect to the Pocket A²B Bus Monitor. Select the FlexTech AKT Comm Port assigned to Bus Monitor.



NOTE: *All FlexTech AKT equipment has a FlexTech AKT Comm Port. If more than one FlexTech AKT device is plugged in, and the Device Status does not show connected after clicking Connect, select a different comm port.*

When the connection is established, the **status bar** at the bottom of the GUI window **displays the device's firmware version and serial number**.



At this point, the A^{2}B Bus Monitor is ready to start monitoring A^{2}B bus activity as shown above.

Programming the DATCTL Register

In order for the Bus Monitor to capture audio or SPI data from the A^{2}B bus, the main node must properly configure the DATCTL register.

The A^{2}B main node must set bit 5 (0x20, ENDSNIFF) to 1 to enable audio data sniffing by the A^{2}B Bus Monitor. Setting this bit automatically triggers the main node to broadcast the setting to any attached bus monitor.

The ENDSNIFF bit should be set during discovery and sent periodically throughout the life cycle of the A^{2}B network. As soon as the Bus Monitor sees the ENDSNIFF bit, it will start capturing audio data. For more information, refer to the *Bus Monitor Support* section in the AD2428 Technical Reference Manual.

Starting a Capture

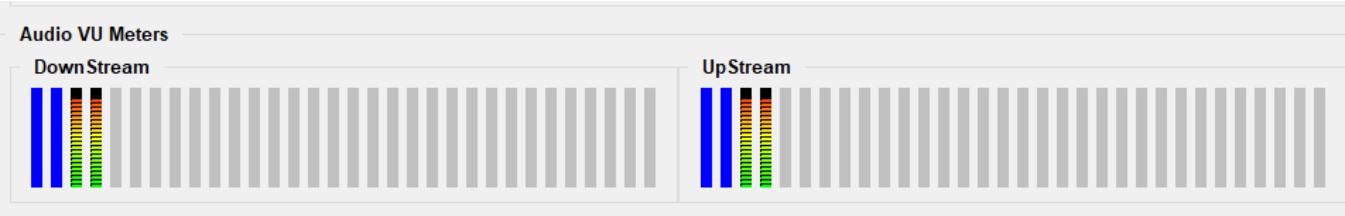
Prior to A^{2}B discovery you should have already:

- Inserted the Bus Monitor into your A^{2}B network.
- Launched the GUI on the host PC and connected to the Bus Monitor.
- Configured the DATCTL register in the main node to allow audio capture.

The Bus Monitor **immediately starts capturing data** from the A^{2}B network as soon as it locks onto the bus during discovery. The Bus Monitor continues to capture all I^{2}C, SPI, and audio data at its position in the network after discovery.

I^{2}C traffic can be captured and inspected regardless of the DATCTL setting. Both SPI and audio monitoring require proper configuration of the DATCTL register.

When the DATCTL register is properly configured, the Bus Monitor captures audio and SPI data. The VU meters are activated as shown below.



If the DATCTL register is *not* properly configured, the Bus Monitor cannot capture audio or SPI data, and all Audio Slot VU Meters are grayed out.

Audio slots reserved for SPI over distance are indicated in *BLUE*.

NOTE: *If you are not seeing any data in the GUI, see Troubleshooting to validate the settings.*

Routing Audio Channels from A^{2}B to USB

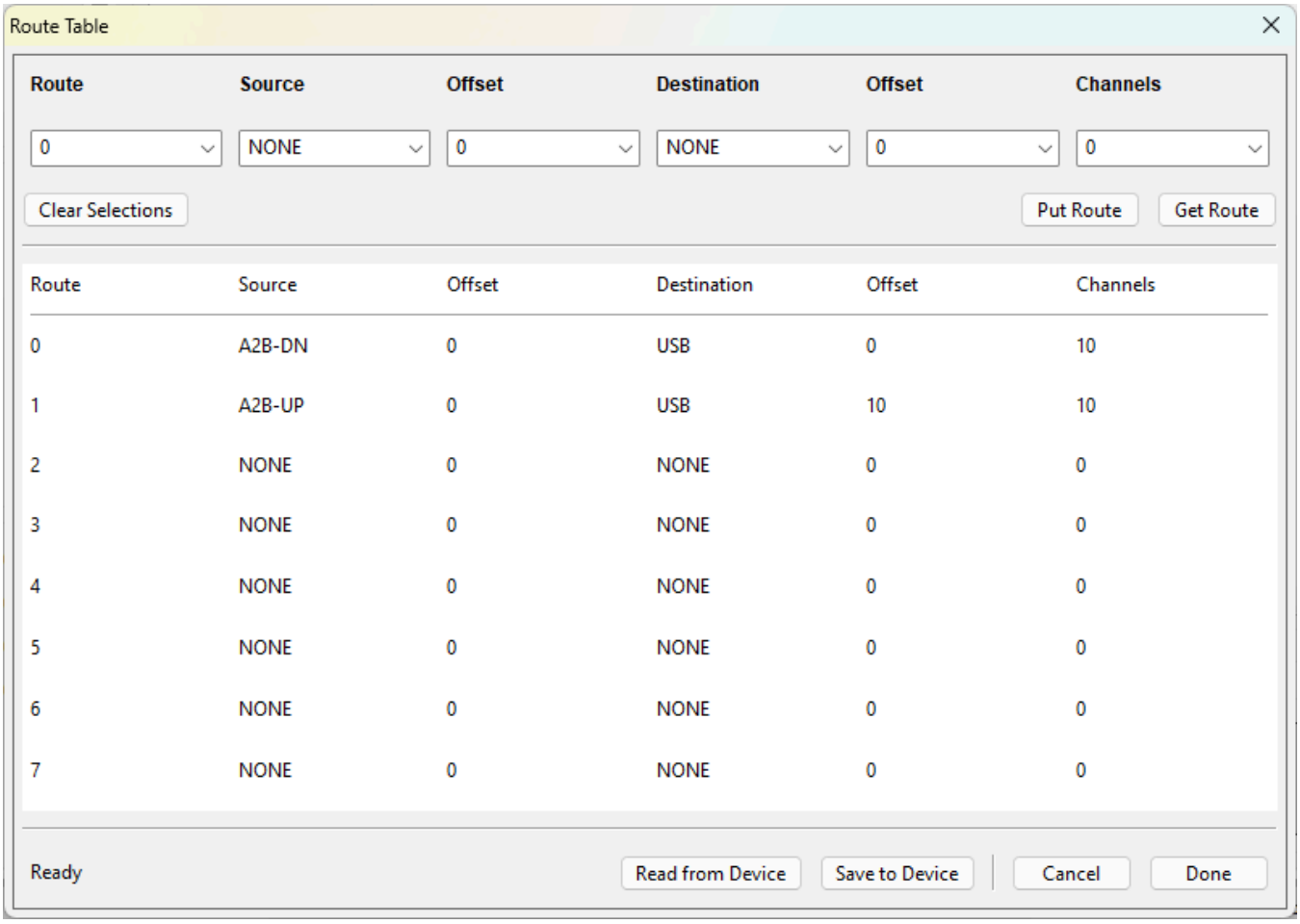
This section demonstrates how to route audio from A^{2}B to USB using the GUI. A more detailed explanation of the audio routing engine can be found in the Audio Routing section.

Routing Audio Using the Bus Monitor GUI

- Navigate to the Bus Monitor GUI top menu
- Select ***Device***
- Select ***Edit Routing Table*** from the drop down menu

Routing Table

This example below configures a variety of routes to illustrate the flexibility of the routing engine.



To configure additional routes, select the Route number, Source + Offset, Destination + Offset, and Channels in the upper section of the dialog box and press the **Put Route** button.

To edit an existing route, select the Route number and press the **Get Route** button.

To quickly clear the Route selections, press the **Clear Selections** button.

To update the active routes in the A^{2}B Bus Monitor, press **Save to Device**.

To read active routes from the A^{2}B Bus Monitor, press **Read from Device**.

To discard changes, press **Cancel**.

To store changes in the current session press **Done**.

Recording Audio

Bus Monitor GUI

The Bus Monitor GUI version can record multi-channel audio directly to a WAV file from USB.

The main ribbon bar tools shown below and the “Audio” menu options control the recording

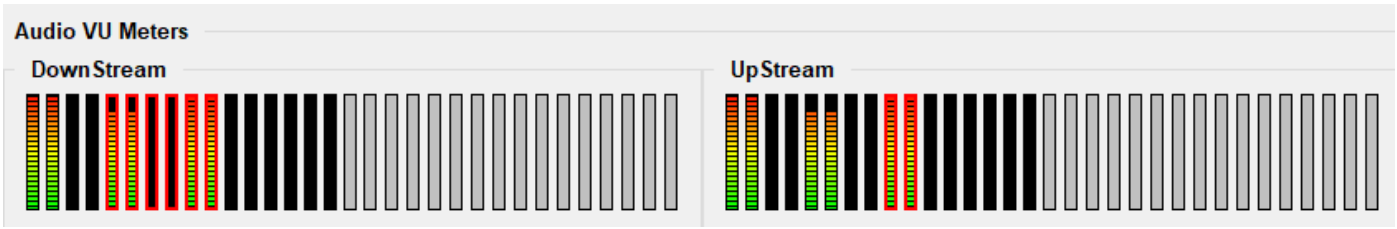
The tools from left to right are “Select Audio Device” , “Set WAV File Name”, “Record Audio”, “Pause Recording” and “Stop Recording”. To record audio:

1. Connect to the Bus Monitor.
2. Configure the Audio Routing Table to route the desired A2B slots to USB
3. Press “Select Audio Device” and select the Pocket A2B Bus Monitor sound card
4. Press “Set WAV File Name” to select the WAV file
5. Press the “Record Audio” button to start recording
6. Press the “Pause Recording” button to pause recording
7. Press the “Stop Recording” button to stop recording

The GUI uses the routing table to determine which A^{2}B slots to record. Channels recorded in the WAV file will start at the first routed USB channel and end at the last routed USB channel. Any empty USB channels in between the first and last routed USB channels will also be recorded. For example, the following routes will record six downstream A^{2}B slots and two upstream A^{2}B slots for a total of 8 channels of audio:

Route	Source	Offset	Destination	Offset	Channels
0	A2B-DN	4	USB	0	6
1	A2B-UP	8	USB	6	2

The GUI highlights the VU meters of the A^{2}B slots being recorded while a recording is active.



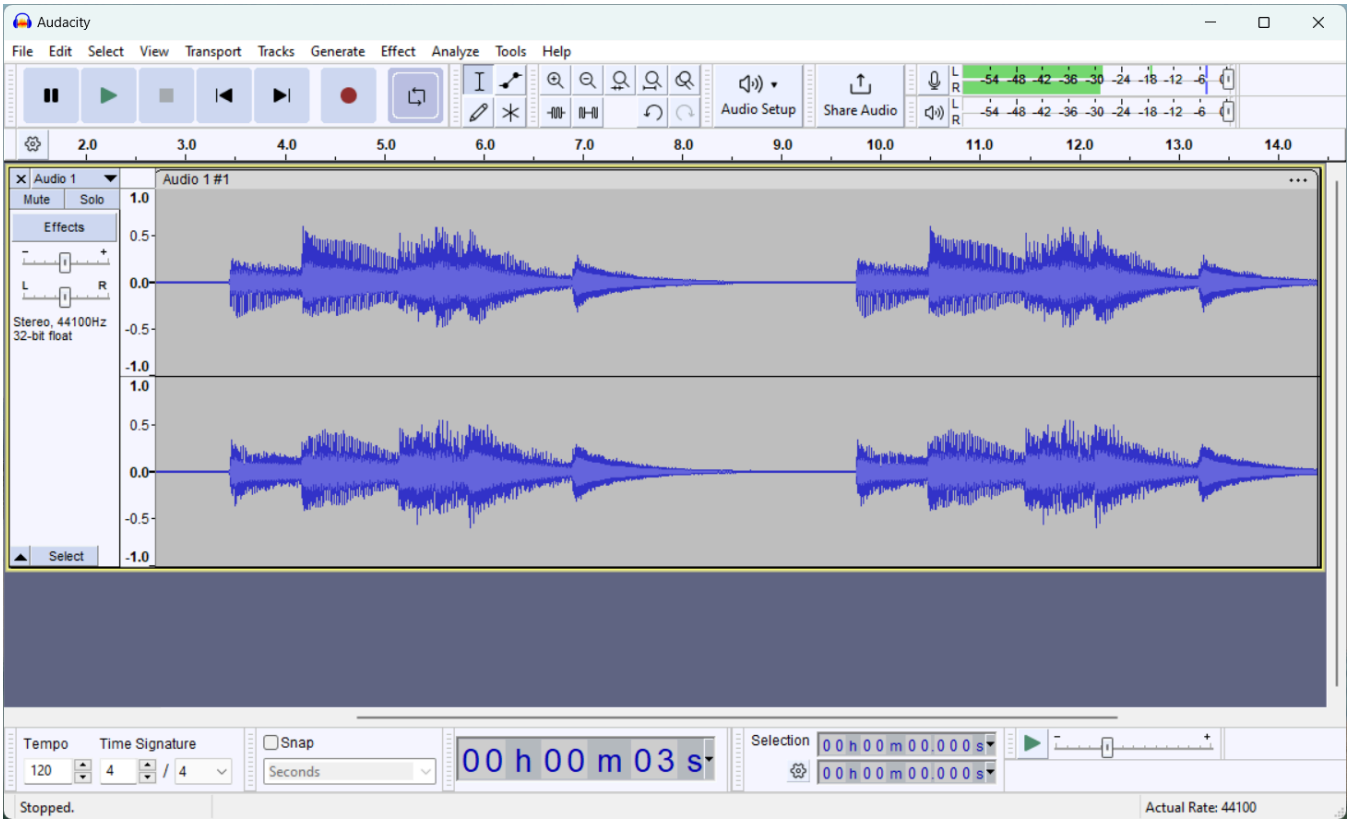
***NOTE**: The Bus Monitor GUI will automatically stop recording once the 4GB WAV file size limit has been reached.*

***NOTE**: Do not confuse WAV route destinations with the Bus Monitor GUI WAV recording. The Bus Monitor GUI records audio from routes with a USB destination. The WAV route destination is for recording directly to the Pocket Bus Monitor's SD card and used for autonomous mode audio recording.*

Digital Audio Workstation

After A^{2}B audio slots are routed to USB **you can use a Digital Audio Workstation (DAW) software,** such as Audacity, Reaper, and more to capture, record, analyze and playback the A^{2}B audio channels.

Audacity is a free Audio Workstation program. Click here to access the latest version and user guide.



Example of Audacity recording the two A2B-DN channels that were routed to USB in the previous section, Audio Routing in the Bus Monitor GUI. In the Audio Setup, select the Microphone (Flextech AKT Bus Monitor) as the recording device.

Listening to Audio

To listen directly to USB audio from the A^{2}B Bus Monitor on Windows, do the following.

1. "Navigate to Control Panel -> Hardware and Sound -> Sound" or search and launch "mmsys.cpl" on Windows 11.
2. Select the "Recording" tab and locate the Pocket A^{2}B Bus Monitor
3. Select the Pocket A^{2}B Bus Monitor then click Properties
4. Select the "Listen" tab and check "Listen to this device"

A short YouTube video demonstrating this process can be found here.

Recording a Trace

Recording an Event Trace

All control data flowing on the bus, including I^{2}C and SPI, can be recorded to a file. The ENDSNIFF bit in the DATCTL register needs to be set to view SPI but does not need to be set to capture I^{2}C.

The buttons for recording control data are located on the right side of the event trace window.

You can:

- Start, stop, and pause recordings.
- Save the current contents of the event trace pane to a file
  OR
- Select a range of events from the pane and copy them to the clipboard.

Captures can span multiple discoveries. For example, you could click **Record**, then run a discovery, reset the network and run another discovery, and data for both discoveries will be captured.

Importing a Trace

Importing an Event Trace

Saved event traces can be imported back into the Bus Monitor GUI for offline display or plugin processing. Supported formats are Flextech AKT .txt format and Mentor / Siemens .etf format. Use the “Import Event Trace” button or the “Event –> Import Event Trace” menu.

Reading an Event Trace

Events are color-coded to make reading and interpreting traces easier. All events with a background color are I²C or SPI transactions.

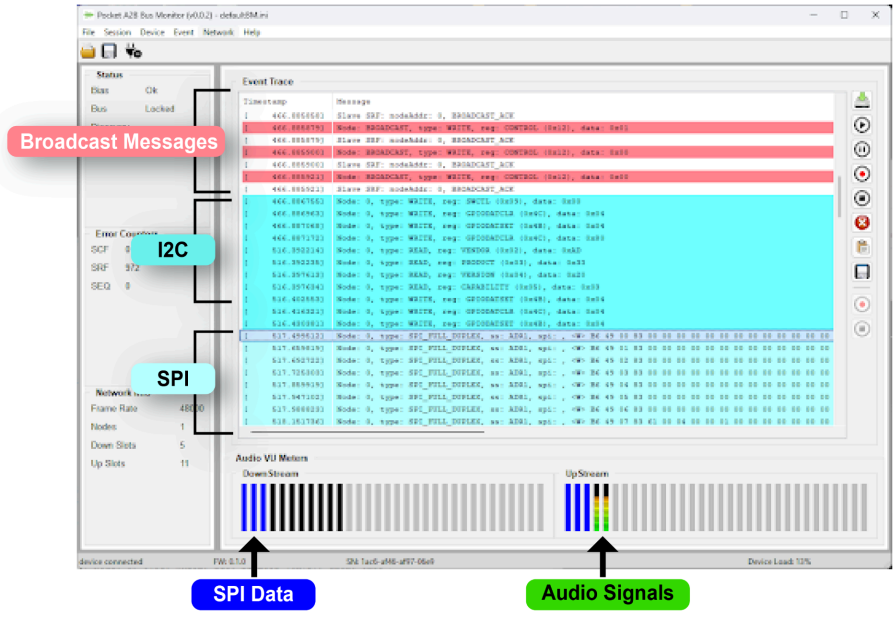
- Each node is individually color coded.
- **A dark shade** indicates an A²B transceiver register read or write.
- **A light shade** indicates a sub node peripheral read or write. The following table shows how nodes are mapped to colors:

Node	Color
Node 1	Green
Node 2	Blue
Node 3	Orange
Node 4	Turquoise
Node 5	Purple
Node 6	Cyan
Node 7	Brown
Node 8	Red-orange
Node 9	Yellow

Additional Colors:

Salmon	Events with a salmon color background indicate A²B broadcast messages.
White	Events with a white color background indicate other bus events, such as bus lock status, bus bias status, discovery mode start/stop, and so on.

Example Event Trace with I²C and SPI Peripheral Transactions



I²C Transactions

I²C transactions have the following fields:

- [seconds.microseconds] — Event timestamp in microseconds.
- Node — Node position
- type — I²C transaction type.
- addr — Address of the target register or peripheral I²C address.
- i2c — Details about the I²C transaction. The following codes are used:
  - <S> — Start condition
  - <P> — Stop condition
  - <Sr> — Repeat Start condition
  - <R> — Read
  - <W> — Write
  - <?> — Unknown
  - <E> — Error

I²C timestamps are relative to the beginning of the transaction.

SPI Transactions

SPI transactions have the following fields:

- [seconds.microseconds] — Event timestamp in microseconds
- Node — Node position
- type — SPI transaction type.
- ss — SPI Slave select.
- spi — Details about the SPI transaction. The following codes are used:
  - <W> — Write data
  - <R> — Read data

SPI timestamps are relative to the beginning of the transaction.

NOTE: *Some events are longer than what can be displayed in the event window. Right click on the event of interest to show the full content.*

Using the Pocket Bus Monitor Command Line

In addition to the Bus Monitor GUI, the Bus Monitor has a command line interface for setup and monitoring.

Download and Install a Terminal Emulator Utility (Tera Term)

To access the command line interface, you must install a terminal emulator such as Tera Term, Putty, GTKTerm, or screen.

Tera Term under Windows is recommended because it has built in support for X/YMODEM that can be used to transfer files over USB.

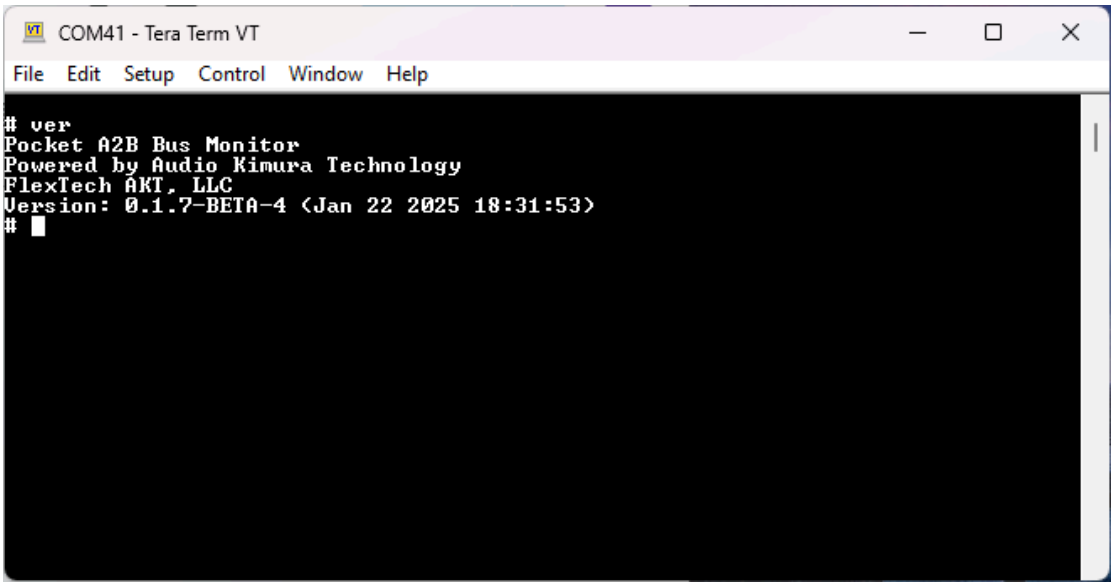
Click Here to Access the latest version of **Tera Term**

Hookup

To access the command line, disconnect the GUI from the serial port and start a terminal emulator on the "FlexTech AKT Comm Port" USB UART. Since this is a virtualized USB serial port, the terminal settings are not critical, but suggested to be 115200 Baud, No parity, 8 data bits, 1 stop bit (115200,N,8,1).

Once connected, press `<ENTER>` a few of times until a `'#'` prompt appears. Type `'ver'` and press `<ENTER>`.

A welcome/version message similar to the following will be displayed:



***NOTE**: The Bus Monitor GUI communicates with the A²B Bus Monitor through the same serial port as the*

command line. Therefore, the GUI and command line cannot be used at the same time.

Using the Command Line

The A²B Bus Monitor has a number of useful commands for setup, debug, automation, and maintenance activities.

Type `'help'` at the command prompt to see the full list of available commands. Additional help specific to each command can be accessed by typing `'help <command>'`.

There are many commands available on the A²B Bus Monitor so the contents of the `'help'` command varies depending on the Command Level.

- **Command Level 0** - Available at startup and only shows the most common commands.
- **Command Levels 1 and 2** - Display additional system maintenance and troubleshooting commands.

***NOTE**: All commands can be run at all levels to facilitate scripting. Only the help is filtered by the command level.*

Use the `'shell'` command to change command levels. For example to switch to command level 2, enter the following command:

```
# shell level 2
```

General Commands

Command	Purpose
`edit`	Simple text editor that can be used to modify small text files directly on the A²B Bus Monitor.
`hwid`	Show device HW ID
`reset`	Resets system components. A “soft” returns the A²B Bus Monitor to its power on reset state.
`resize`	Resize or Sync the terminal window size. Run this command after the Tera Term command window is resized to synchronize the new terminal size or specify a size to resize the terminal window.
`ver`	Show version information
`help`	Shows specific help for commands

For additional help on any command use the built-in help command:

```
# help help
help - shell help
Usage: help [<command>]
  [<command>] - the command to get help on.
Without arguments it shows a summary of all the shell commands.
```

File Management

It is often necessary to install files onto the A²B Bus Monitor to enable automation features. The most direct method is to simply copy files to or from a PC on the SD card. The SD card can be safely removed and reinstalled when not in use.

NOTE: *Always remember to eject the card from the PC prior to removing it.*

In addition to the SD card, the A²B Bus Monitor has a small internal "Flash" file system. This file system is meant for system files that must be maintained even when the SD card is ejected and for files used early at startup. Such files include:

File	Purpose
`XXXX-XXXX-XXXX-XXXX.key`	This file is the product activation key and required for proper operation. ***Never modify or erase this file. It is recommended to make a backup of this file and store it somewhere safe.***
`shell.cmd`	Commands in this file are automatically run at system startup.
`cfg.ini`	Accessed early in the boot process to override system defaults such as USB audio channels or Ethernet settings.

Wherever file names are mentioned in this document, prefix the actual file name with `'sf:'` to access files on the internal Flash file system or `'sd:'` to access files on the SD card. File names with no prefix will default to the SD card.

NOTE: *The maximum file name length on the internal Flash filesystem is 30 characters*

File Commands

The following commands are available on the A²B Bus Monitor for manipulating files. Type `'help <cmd>'` on the command line for detailed usage instructions.

Command	Purpose

`cat`	Show the contents of a text file. Do not use this command with binary files. Use the `'dump'` command to display binary files.
`cp / copy`	Copy a file
`df`	Show the drive full status
`drive`	Show or set the default drive
`dump`	Show the contents of a file in hex
`format`	Format a drive. ***WARNING:*** *Formatting the internal flash file system will erase the activation key rendering the A²B Bus Monitor inoperable. It should never be necessary to format the internal sf: filesystem.*
`fsck`	Check the integrity of a drive
`edit`	Edit a text file
`ls / dir`	Show a directory listing of a drive
`recv`	Receive a file via XMODEM. If no file name is given, receive multiple files via YMODEM.
`send`	Sends one or more files via YMODEM
`rm / del`	Delete a file
`run`	Run a command script
`tail`	Show the last <n> lines of a text file

File Transfers

The A²B Bus Monitor supports a variety of methods to transfer or create files for system setup.

SDCARD

The most direct method is to simply copy files to or from a PC using the SD card. The SD card can be freely removed and reinstalled in the A²B Bus Monitor when not in use.

NOTE: *Always remember to eject the card from the PC prior to removing it.*

Edit Command

Simple text files can be created or modified directly from the command line using the `'edit'` command. Press `<CTRL-S>` to save the file. Press `<CTRL-Q>` to quit editing.

X/YMODEM

Files can be downloaded to the A²B Bus Monitor using the XMODEM or YMODEM protocols via the `'recv'` command. XMODEM is used to transfer a single file and used when a file name is provided. YMODEM is used to transfer multiple files when no file name is specified.

TeraTerm supports both XMODEM and YMODEM file transfer protocols. After issuing the `'recv'` command, select `File -> Transfer -> [X][Y]MODEM -> Send` to initiate a transfer.

Files can be uploaded from the A²B Bus Monitor using the YMODEM protocol via the `'send'` command. Multiple files can be sent in a single transfer.

Select the 1k file transfer option for faster transfers.

USB Sound Card Re-Configuration

By default the A²B Bus Monitor is configured as a 20 Speaker OUT x 20 Microphone IN x 16 bit USB sound card.

To modify the default values, create a text file called `cfg.ini` with the contents below. If you already have a `cfg.ini` file, add a `[usb-audio]` section to it.

```
[usb-audio]
out-channels = <2-32 channels>
in-channels = <2-32 channels>
word-size-bits = <16 or 32>
```

Set the values as required for the application. The example below configures 32 OUT channels (from the PC to the A²B Bus Monitor), 2 IN channels (from the A²B Bus Monitor to the PC) with a bit-depth of 16-bits:

```
[usb-audio]
out-channels = 32
in-channels = 2
word-size-bits = 16
```

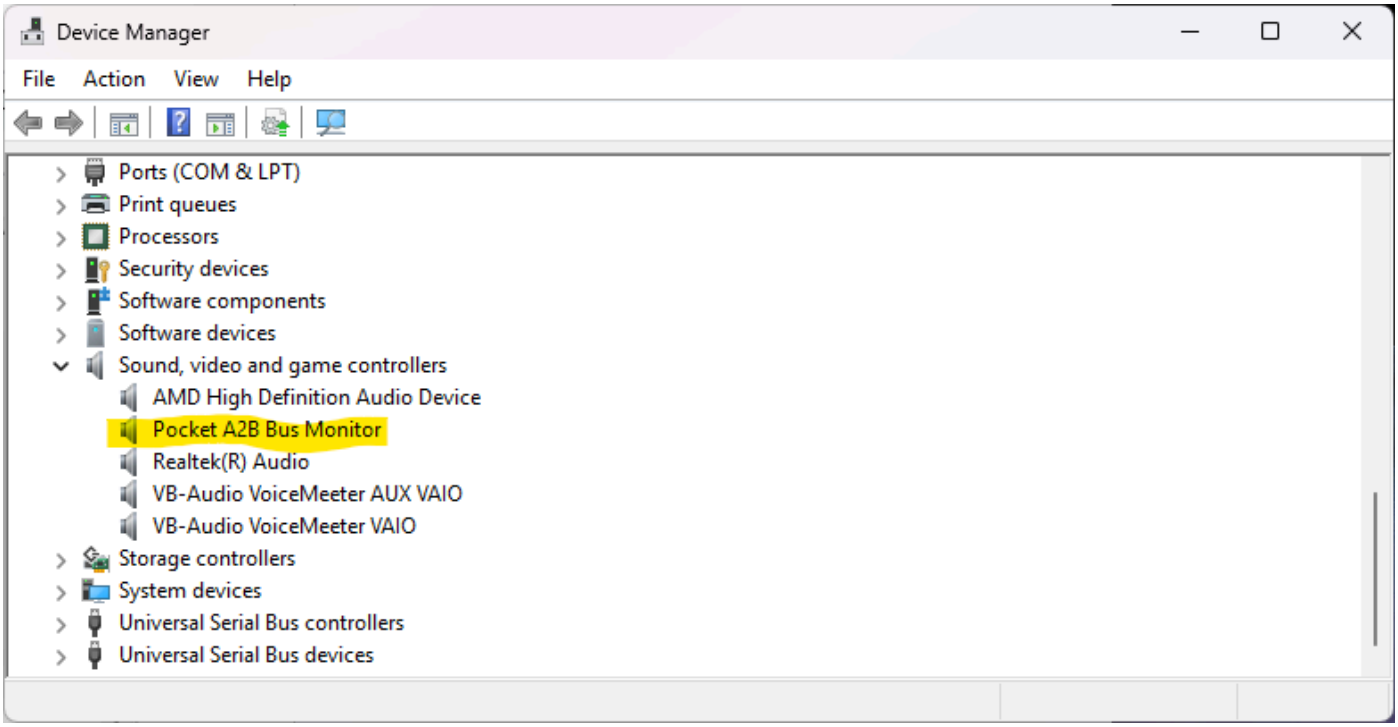
Copy this file onto the SD card, insert the SD card into the A²B Bus Monitor, then copy the file from the SD card to the internal flash filesystem with the following command:

```
cp cfg.ini sf:cfg.ini
```

The `sf:cfg.ini` file can also be created or modified directly on the A²B Bus Monitor using the `'edit'` command.

For Windows 10/11, you must uninstall the existing A²B Bus Monitor audio driver in the Device Manager after changing any of the USB audio settings.

With the A²B Bus Monitor powered and connected, right click on the "A²B Bus Monitor" sound card and select uninstall:



Reset the A²B Bus Monitor and Windows will apply the new settings.

NOTE: *Be very careful when configuring this file. Incorrect settings can result in a boot failure that may require a Safe-Boot Recovery of the A²B Bus Monitor.*

NOTE: *Do not set the number of channels or bit-depth higher than required by the application. Unused channels still consume CPU cycles on the A²B Bus Monitor and USB bandwidth on the PC.*

Bus Monitor Commands

Most common bus monitor activities are supported directly on the command line. Below are commands specific to bus monitoring.

Command	Purpose
`bm`	Display A²B bus control traffic. The output of this command mirrors the Event Trace in GUI. Press any key to exit the command.

`bmm`	Set the bus monitor mode. Used to configure and connect to a live A²B bus. See Connecting to a live A2B Bus
`bmspi`	Save captured A²B SPI over distance data to a file
`bmspidump`	Dumps A²B SPI over distance data from a captured file

Connecting to a live A²B Bus

The A²B Bus Monitor normally configures itself by monitoring the A²B discovery process. During discovery, the A²B Bus Monitor synchronizes a number of internal settings with those of the next downstream node.

The following process can be used to monitor an A²B bus when the A²B Bus Monitor is connected after discovery. This mode is called “no discovery” mode.

Step 1: Discover the A²B network with the Bus Monitor Attached

Begin by monitoring a normal A²B discovery to allow the A²B Bus Monitor to acquire the A²B network configuration.

Step 2: Dump the settings to a file

After discovery, issue the `'bmm dump <file>'` command where `<file>` is the name of a file to dump the settings. This command will create a simple command script that can be used to configure the A²B Bus Monitor for “no discovery” monitoring. Use the `'cat <file>'` command to view the contents of the file.

NOTE: *The settings depend heavily on the A²B bus configuration and where the A²B Bus Monitor is positioned on the bus. If the network configuration changes, or the A²B Bus Monitor is moved to a new position, this step will need to be repeated. Any number of setting dumps can be stored on the A²B Bus Monitor to allow easy attachment to any number of A²B networks or positions.*

Step 3: Connect to a live bus

After powering up or connecting the A²B Bus Monitor to a live bus, issue the following sequence of commands from the command line. If progressing from Step 2, issue the `'reset'` command to reset the A²B Bus Monitor.

Seq	Command	Action
1	`run <file>`	Run the command script created in Step 2 to load the A²B configuration settings
2	`bmm nodiscover`	Places the A²B Bus Monitor in “no discover” mode
3	`bmm connect`	Electrically connect to the A²B bus

4	`delay 100`	Delay to allow the A^{2}B Bus Monitor to lock to the bus
5	`bmm configure`	Apply the A^{2}B bus configuration

These commands can be combined into a single command script and executed using the `'run'` command.

Step 4: Begin monitoring

After Step 3 the I/O LED should be a steady green and the A^{2}B bus can be monitored as usual. To monitor audio or SPI data, the A^{2}B Bus Monitor must see a broadcast write of the DATCTL (0x12) register with the ENDSNIFF (0x20) bit set.

To monitor with the GUI, disconnect the terminal from the serial port and use the GUI as usual.

From the command line, use the `'bm'` command for command line monitoring of control traffic and the `'route'` command to route audio from A^{2}B. The `'bmspi'` command can be used to save high bandwidth SPI data to the SD card. The `'bmspidump'` command can be used to dump the SPI data stored in the file to the terminal.

The A^{2}B Bus Monitor will automatically exit “no discovery” and return to normal mode when it detects a new A^{2}B discovery.

***NOTE**: Do not disconnect or break the network when attaching the Bus Monitor to a live A^{2}B Bus. The A^{2}B Bus Monitor should have been wired into the network prior to connecting. While it is possible to “tap” onto a live network, the act of physically attaching the A^{2}B Bus Monitor to the A^{2}B bus might disrupt the A^{2}B bus or introduce errors. If tapping onto the bus is the only option, make the stub length as short as physically possible. Attach either the “A” or “B” side of the Bus Monitor to the A^{2}B bus being sure to maintain proper polarity.*

Qwiic Peripheral Configuration

Qwiic peripheral configuration via I²C on the qwiic connector is supported through the command line and AKT Automation scripts.

I²C Commands

Command	Purpose
`qwiic_i2c`	Performs an I²C write, read, or write/read transaction on the qwiic connector
`qwiic_scan`	Scans the I²C bus on the qwiic connector for active devices

NOTE: *The I²C bus speed is fixed at 400KHz*

Qwiic Examples

Scan the qwiic connector for I²C devices

```
# qwiic_scan
Probing I2C port 2:
 Found device 0x68
```

1 byte write / 16 byte read I²C transaction to device 0x68 on the qwiic connector

```
# qwiic_i2c 0x68 "0" 16
I2C Device 0x68, Read Bytes 16 (0x10)
00000000: 34 07 20 02 06 11 24 00 00 00 00 00 00 00 1c 88
```

Audio

General

The A²B Bus Monitor processes audio in blocks of 128 samples. Latency through the A²B Bus Monitor for all sources and destinations, except for USB, is 2.67uS. USB audio latency is approximately 960 samples, or 20mS. Latency through USB is generally controlled to within +/- 10 samples of nominal.

Internal audio samples are all 32-bits wide. Audio is up / down converted as required to 32-bits from 16-bit sources / destinations. Samples down-converted from 32-bit to 16-bit are truncated.

All audio within the A²B Bus Monitor is fundamentally clocked by the A²B bus. Audio does not flow through the A²B Bus Monitor when the A²B bus is idle.

Audio routing is “bit perfect” from source to destination.

A²B

All A²B audio is 32-bits regardless of the A²B network settings. The A²B Bus Monitor can capture the first 30 slots of both upstream and downstream audio for a total of 60 slots.

USB Audio

By default the A²B Bus Monitor supports 20 IN (to PC), 20 OUT (from PC), by 16-bit audio. These settings can be modified by the `cfg.ini` file as necessary. The `'usb'` command can be used to view USB audio statistics.

View the audio statistics during long-term USB audio recording or playback when testing is sensitive to gaps in audio. Windows 10/11 are not real-time operating systems and can fail to transfer USB audio in a timely manner during periods of high system load.

To listen directly to USB audio from the A²B Bus Monitor on Windows, do the following.

1. "Navigate to Control Panel -> Hardware and Sound -> Sound" or search and launch "mmsys.cpl" on Windows 11.
2. Select the "Recording" tab and locate the Pocket A²B Bus Monitor
3. Select the Pocket A²B Bus Monitor then click Properties
4. Select the "Listen" tab and check "Listen to this device"

A short YouTube video demonstrating this process can be found here.

WAV Files

The A^{2}B Bus Monitor can play and record 16-bit or 32-bit multi-channel WAV files. WAV files up to thirty two 32-bit channels are supported assuming the SD card has sufficient bandwidth. Use the `'sdtest'` command to confirm SD card bandwidth.

A single WAV file can be the source or destination of multiple routes. If an audio route (offset plus channels) extends beyond the number of channels available in a WAV file destination the extra channels are dropped. Empty WAV destination channels are zero filled. If a route extends beyond the channels available in a WAV file source, the missing channels are zero-filled.

Use the `'wav'` command to start and stop WAV file playback or recording.

NOTE: *Always use a minimum Class 10 or UHS Class 1 SD card if the card will be used for WAV file audio. Use a freshly formatted SD card when recording high bit-rate WAV files to reduce the risk of audio drops. Use the `'sdtest'` command to confirm acceptable SD card bandwidth.*

Circular Audio Buffer (CBUF)

The A^{2}B Bus Monitor has a programmable circular memory buffer (CBUF) that can be used to record a rolling window of audio. The audio inside the CBUF can be dumped to a WAV file on the SD card at any time. This feature can be used to capture transient audio events without having to record a continuous audio WAV file.

Use the `'cbuf'` command to configure and control the CBUF.

VU Meters

The A^{2}B Bus Monitor has a set of simple built in VU meters. The VU meters are very useful for locating audio on A^{2}B. Any captured upstream or downstream A^{2}B slot can be routed to the VU meters.

VU meters can be the destination of multiple routes.

Use the `'vu'` command to view the VU meters.

NOTE*: There is a known issue where the VU meter screen is blank with TeraTerm version 5. TeraTerm Version 4.108 is recommended. The rendering character for the VU meters can also be set to a `'*'` with the following command `'vu 32 *'`.*

ASIO

The A^{2}B Bus Monitor supports ASIO through the ASIO4ALL project. Please refer to the ASIO4ALL website for more information https://asio4all.org/.

Audio Routing

The `'route'` command is the key command for transferring audio between audio streams on the A^{2}B Bus Monitor. The A^{2}B Bus Monitor supports up to eight simultaneous multi-channel routes enabling very sophisticated audio routing schemes. The routing engine is a full crossbar between any source and any destination.

The table below describes the routing capabilities of the A^{2}B Bus Monitor

Stream	Src / Dest	Notes
usb	Dest	
a2b-dn	Src	
a2b-up	Src	
wav	Both	
cbuf	Dest	
vu	Dest	

An audio route starts with a source stream. The source stream is the base stream, like 'a2b-dn' or 'a2b-up'.

A specific channel within a source stream is identified with the source channel offset. For A^{2}B the source channel offset refers directly to the A^{2}B bus audio slot. For other streams, the offset refers to the channel offset.

The route source must then connect to a destination. Destination streams are identified by stream and channel offset exactly like source streams.

A route copies a defined number of channels from the source to the destination. Audio can be optionally attenuated and mixed during this copy.

Route command arguments are :

```
  route [ <idx> <src> <src offset> <dst> <dst offset> <channels> [attenuation]
[mix|set] ]
```

Below are some examples:

Route the first 2 slots from A^{2}B downstream to USB with no attenuation

```
route 0 a2b-dn 0 usb 0 2
```

Route the first two slots from A^{2}B upstream to USB with a channel offset of 2

```
route 1 a2b-up 6 usb 2 2
```

Route 16 channels from both a2b-dn and a2b-up to the VU meters to find active slots

```
route 16 a2b-dn 0 vu 0 0 16
route 16 a2b-up 0 vu 0 16 16
```

A route that exceeds the number of source channels (i.e. source offset + channels > source channels) will zero fill the missing channels. Routes that exceed the number of destination channels (destination offset + channels > destination channels) will terminate after the last destination channel.

***NOTE**: A^{2}B SPI over distance slots are routed just like audio slots.*

Peak Detectors

The A^{2}B Bus Monitor supports real-time peak detectors on all captured upstream and downstream slots. These peak detectors continuously monitor A^{2}B audio. The peak detectors also indicate SPI data slots with the words 'Active' and 'Idle'. The peak detectors can be viewed or cleared using the `'peaks'` command.

Command Scripts

Any series of commands can be grouped together into a command script and executed using the `'run'` command. Command scripts are simple text files containing one command per line. Lines starting with a semicolon or hash mark are ignored and treated as comments. Any valid file name can be used for command scripts.

Some commands are especially useful in command script processing

Command	Purpose
`delay`	Delays script execution for a specified number of milliseconds
`echo`	Displays a line of text
`shell redirect`	Silences or redirects output of a script to the syslog
`reset`	Reset various subsystems to power on reset values. No arguments performs a full hardware reset.

Be careful scripting interactive commands as the script will not proceed until the interactive command has completed. Interactive commands include `'syslog', 'vu'`, and interactive `'lua'` scripts.

Below is an example command script that captures 8 slots of downstream and 8 slots of upstream A2B audio to a 16 channel WAV file for 5 seconds.

```
# Perform a soft reset
reset soft
# Route 8 slots of downstream audio to a WAV file
route 0 a2b-dn 0 wav 0 8
# Also route 8 slots of upstream audio to the WAV file
route 1 a2b-up 0 wav 8 8
# Start a 16 channel WAV file recording
wav sink on sink.wav 16
# Delay 5 seconds
delay 5000
# Stop recording
wav sink off
```

Running commands at startup

If present, the A²B Bus Monitor runs `'sf:shell.cmd'` at startup. This feature allows for autonomous configuration of the A²B Bus Monitor at startup.

NOTE: *The SD card takes some time after startup to initialize. Be sure to add a delay of at least 500ms at the beginning of any startup script that uses files on the SD card.*

NOTE: *Be careful including interactive commands in the startup script. For example, launching a Lua script that never terminates will lock out the command line requiring a Safe Boot Recovery to correct.*

AKT Automation with Lua

Fully autonomous operation on the A²B Bus Monitor is made possible through the on-board Lua based AKT Automation environment. This feature is an optional add-on for the Pocket A²B Bus Monitor.

Lua is a fully-featured open source scripting language. More information on Lua, including programmer reference manuals, can be found at https://lua.org/.

The pairing of Lua with the A²B Bus Monitor command and control APIs results in an extremely rich interactive A²B automation environment.

User interfaces can be created using the 'term' module. Low-level operations are possible using the 'rtos' and 'system' modules. Qwiic compatible peripherals, such as RTCs, displays, pushbuttons, etc., are accessible via the ‘qwiic’ module.

A full complement of A²B bus monitor events are available through the 'bm' module.

For more detailed information on the API, refer to the ***FlexTech AKT Automation API Guide***.

Example AKT Automation scripts can be found in the **AKT-Automation** repository on Github.

When developing Lua scripts, the following techniques can speed up script development:

1. Use XMODEM to transfer scripts to the A²B Bus Monitor instead of the SD card
2. Running Lua with no arguments starts an interactive Lua interpreter. Lua code "chunks" can be copied and pasted from the PC into the interpreter for quick prototyping of logic or code blocks.
3. Use the on-board `'edit'` command for quick bug fixes or script modifications.

Bus Monitor GUI Plugins

Event processing by the Bus Monitor GUI can be extended through the use of Plugins. Like the Flextech AKT Automation environment, the Bus Monitor GUI plugin environment is based on the Lua programming language. A series of pre-defined Lua methods, callbacks, and constants allow plugins to process A^2B events received by the Bus Monitor GUI and output color coded text to the Event Trace window.

Plugins can be loaded, reloaded, unloaded, started and stopped at any time through the “Plugin” menu.

Plugins can access third party Lua Modules such as those provided by the Penlight Lua Library and many others. The Plugin path is included in the default Lua module search path. Third party modules should be placed in the same folder as the Plugin.

Bus Monitor GUI Plugins can be found in the **AKT-Automation** repository on Github.

The script below demonstrates a minimal Bus Monitor GUI plugin. This Plugin echoes all events back to the Event Trace window in purple. The script requires the Penlight Lua Library. Place the Penlight `'pl'` folder in the same folder as the Plugin.

```
pretty = require('pl.pretty')

PURPLE = {177,156,217}

--[[
-- Bus Monitor GUI plugin functions
--]]
function plugin_loaded()
    plugin_trace(0, 'Plugin Loaded')
end

function plugin_unloaded()
    plugin_trace(0, 'Plugin Unloaded')
end

function plugin_stop()
    plugin_trace(0, 'Plugin Stop')
end

function plugin_start()
    plugin_trace(0, 'Plugin Start')
end

function plugin_event(e)
    e.event = BM_EVENTS[e.event] or e.event
    plugin_trace(e.timeStamp, pretty.write(e, ''), PURPLE)
end
```

Plugins can be developed within the GUI or outside of it. When developing within the GUI, use the Reload menu option to iteratively reload the current Plugin. Parsing and loading errors are output to the Event Trace window.

When developing outside of the GUI, use the BM_GUI_PLUGIN global variable to enable standalone code or create a Plugin compatible environment.

A standalone example that creates a Plugin compatible environment and parses register reads and writes from an Event Trace dump can be found in the **AKT-Automation** repository on GitHub in the `examples/bm_gui_plugin/standalone` folder.

Pre-defined Constants

The following constants are provided in the Lua environment by the Bus Monitor GUI

Name	Type	Notes
BM_EVENTS	Table	Event table. Use this table to map event identifiers to or from readable strings.
BM_I2C_TYPES	Table	I2C event type table. Use this table to map A^{2}B I2C event type identifiers to or from readable strings.
BM_I2C_CONDITIONS	Table	I2C peripheral condition table. Use this table to map A^{2}B I2C peripheral condition identifiers to or from readable strings.
BM_GUI_PLUGIN	Boolean	This constant is set to true when the plugin is running within the Flextech Bus Monitor GUI environment.

Pre-defined Callbacks

plugin_trace()

ok = plugin_trace(timestamp, text [,color])

Output text to the Bus Monitor GUI Event Trace window.

***NOTE**: The Event Trace window is only active when the Bus Monitor GUI is attached to the Bus Monitor or while an event trace is being imported.*

Parameters

Parameter	Type	Optional	Description
timestamp	number	no	Event Trace timestamp in seconds. Normally the same as an associated event timestamp.
text	string	no	Event Trace text

Parameter	Type	Optional	Description
color	table number	yes	Custom RGB values can be defined as a table { R, G, B } The standard node color will be used when ‘color’ is a node number. No color will be used when omitted

Return Values

Return	Type	Optional	Description
ok	Boolean	N/A	True for success otherwise false

Required Plugin Methods

The following methods must be implemented within a Plugin for proper operation.

plugin_loaded()

plugin_unloaded()

Called when the plugin is loaded or unloaded

Parameters

Parameter	Type	Optional	Description
N/A	N/A	N/A	This method has no parameters.

Return Values

Return	Type	Optional	Description
N/A	N/A	N/A	This method has no return value.

---

plugin_start()

plugin_stop()

Called when the Bus Monitor GUI ‘Plugin Start’ or ‘Plugin Stop’ menu item is selected.

***NOTE**: The Bus Monitor GUI continues to send events regardless of the start / stop state. The Plugin is free to define the behavior of these states internally.*

Parameters

Parameter	Type	Optional	Description
N/A	N/A	N/A	This method has no parameters.

Return Values

Return	Type	Optional	Description
N/A	N/A	N/A	This method has no return value.

---

plugin_event()

plugin_event(event)

Called when the Bus Monitor GUI has a new event to process

Parameters

Parameter	Type	Optional	Description
event	table	no	<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>event</td><td>number</td><td>Event enumeration</td></tr><tr><td>timeStamp</td><td>number</td><td>Event timestamp</td></tr><tr><td>EVENT SPECIFIC</td><td></td><td>See BM_EVENTS</td></tr></table>

Return Values

Return	Type	Optional	Description
N/A	N/A	N/A	This method has no return value.

A2B Bus Monitor Plugin Enumerations and Events

BM_GUI_PLUGIN

This boolean is set to true when running in the Bus Monitor GUI plugin environment.

BM_I2C_TYPES

Name	Value	Description
I2C_REG	0	Register transaction
I2C_PERIPHERAL	1	Peripheral transaction
I2C_PERIPHERAL_CONDITION	2	Peripheral condition

BM_I2C_CONDITIONS

Name	Value	Description
I2C_RPTSTART	0	start
I2C_RPTSTART	1	repeat start
I2C_ACK	2	ack
I2C_NACK	3	nack
I2C_NORMAL	4	data
I2C_STOP	5	stop
I2C_ERROR	6	error
I2C_UNKNOWN	7	unknown

BM_EVENTS

<table>
<tr><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td>I2C</td><td>2</td><td>I2C register or peripheral event
<table>
<tr><th>Parameter</th><th>Type</th><th>Value</th></tr>
<tr><td>tid</td><td>number</td><td>I2C transaction ID</td></tr>
<tr><td>type</td><td>number</td><td>I2C event type. See BM_I2C_TYPES.</td></tr>
<tr><td>nodeAddr</td><td>number</td><td>Sub node address</td></tr>
<tr><td>addr</td><td>number</td><td>Register or peripheral I2C address</td></tr>
<tr><td>rw</td><td>boolean</td><td>True for read, false for write</td></tr>
<tr><td>data</td><td>number</td><td>I2C data byte</td></tr>
<tr><td>condition</td><td>number</td><td>I2C peripheral condition.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td></td><td></td><td>See BM_I2C_CONDITIONS</td></tr></table></td></tr>
<tr><td>SPI</td><td>3</td><td>SPI event<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>respNode</td><td>number</td><td>SPI node</td></tr><tr><td>error</td><td>boolean</td><td></td></tr><tr><td>type</td><td>string</td><td>SPI transaction type
"ATOMIC_READ"
"ATOMIC_WRITE"
"FULL_DUPLEX"
"BULK_WRITE"</td></tr><tr><td>ss</td><td>string</td><td>Slave select
"ADR1"
"SIO2"
"ADR2"</td></tr><tr><td>wLen</td><td>number</td><td>Write data length</td></tr><tr><td>rLen</td><td>number</td><td>Read data length</td></tr><tr><td>wr</td><td>number array</td><td>Write data</td></tr><tr><td>rd</td><td>number array</td><td>Read data</td></tr></table></td></tr>
<tr><td>BUS_LOCK_LOCKED</td><td>4</td><td>Bus Locked event</td></tr>
<tr><td>BUS_LOCK_UNLOCKED</td><td>5</td><td>Bus Unlocked event</td></tr>
<tr><td>BIAS_OK_DETECTED</td><td>6</td><td>Bias OK detected event</td></tr>
<tr><td>BIAS_OK_REMOVED</td><td>7</td><td>Bias OK removed event</td></tr>
<tr><td>BIAS_REV_DETECTED</td><td>8</td><td>Bias Reverse detected event</td></tr>
<tr><td>BIAS_REV_REMOVED</td><td>9</td><td>Bias Reverse removed event</td></tr>
<tr><td>DISCOVERY_MODE</td><td>10</td><td>Discovery mode event<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>status</td><td>string</td><td>"START"
"STOP"</td></tr><tr><td>respCycles</td><td>number</td><td>Response cycles</td></tr></table></td></tr>
<tr><td>IRQ</td><td>11</td><td>IRQ event<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>nodeAddr</td><td>number</td><td>Sub node address. First submode is zero.</td></tr><tr><td>status</td><td>number</td><td>IRQ status</td></tr></table></td></tr>
</table>

<table>
<tr><td>DOWNSTREAM_SCF_ERROR
UPSTREAM_SRF_ERROR</td><td>12
13</td><td>Downstream SCF error event
<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>type</td><td>string</td><td>"SCF"
"SRF"</td></tr><tr><td>missed</td><td>number</td><td>Control frames missed in last block.</td></tr><tr><td>hdcnt</td><td>number</td><td>Headers missed in last block</td></tr><tr><td>crc</td><td>number</td><td>Bad CRCs in last block</td></tr></table></td></tr>
<tr><td>SLAVE_ERROR</td><td>14</td><td>Slave error / ack event
<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>nodeAddr</td><td>number</td><td>Sub node address. First submode is zero.</td></tr><tr><td>error</td><td>number</td><td>Error number</td></tr><tr><td>errorStr</td><td>string</td><td>"SRFMISSED_ERROR"
"BROADCAST_ACK"
"DISCOVERY_ERROR"
"DOWNSTREAM_CRC_ERROR"
"UNSPECIFIED_ERROR"</td></tr></table></td></tr>
<tr><td>SEQUENCE_ERROR</td><td>15</td><td>Sequence error event
<table><tr><th>Parameter</th><th>Type</th><th>Value</th></tr><tr><td>seq</td><td>Number</td><td>Sequence errors detected in last block</td></tr></table></td></tr>
</table>

Updating the Pocket Bus Monitor's Firmware

WARNING*:** *Prior to updating the firmware, always be sure to disable any custom* ***cfg.ini *and* ***sf:shell.cmd*** *startup files. These may interfere with the proper startup of the new firmware causing the update to fail unexpectedly.*

Download the AKT Flasher Utility from www.flextechakt.com and install.

Follow one of the methods below to update the Bus Monitor Firmware.

Methods to Update the Firmware

Command-line Initiated AKT Flasher Over USB.

To start this update, issue the following commands from the Tera Term command line:

```
# bootmode 1
# reset
```

Proceed to update with AKT Flasher utility. Once the `'reset'` command is issued, the unit must be updated using the AKT Flasher.

Command-line via SD card file

To initiate this update, the desired firmware binary must be on the SD card. One can simply copy it from the PC or use the `'recv'` command to transfer it through TeraTerm via XMODEM over USB.

Once the file is on the SD card, issue the following command:

```
# update <file>
```

Where `<file>` is the firmware binary. Reset the unit once the update is complete using the `'reset'` command or cycling power.

Bootloader initiated AKT Flasher over USB

Press and hold the "Boot Recovery" button while powering on the unit. The unit will immediately enter the bootloader mode and can be updated using the AKT Flasher utility. If the unit is reset before updating the firmware, it will boot normally. Once an update is initiated with AKT Flasher, the update must complete.

Bootloader mode is indicated by a slow yellow blink of the Status LED on the Pocket products.

NOTE: *The "Boot Recovery" button is located under the small hole on the underside of the Pocket A²B Bus Monitor.*

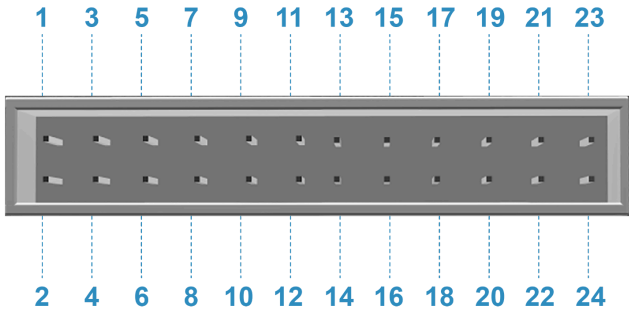
Included Connectors

USB

The USB connector is a standard USB 2.0 Type B receptacle.

24 Pin Multi I/O Connector

JST S24B-PHDSS



Pin	1	3	5	7	9	11	13	15	17	19	21	23
Description	GND	GPIO0							A-	A+	B+	B-
Color	Black	Blue							Brown			Brown

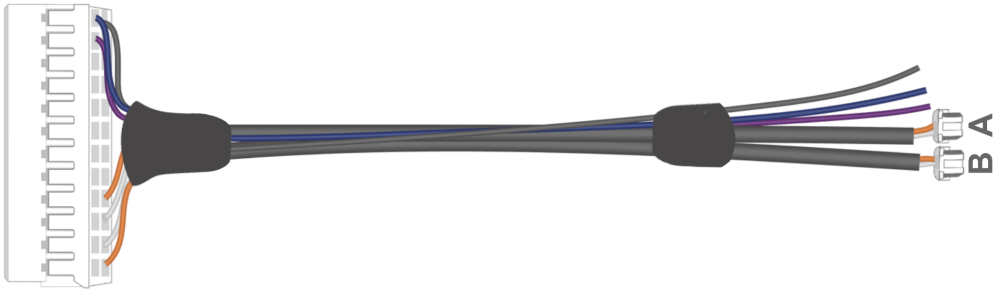
Pin	2	4	6	8	10	12	14	16	18	20	22	24
Description	3.3V	GPIO0										
Color	Black	Purple										

Pin Diagram.

NOTE: *Color shown match harness provided. Gray sections represent unused pins.*

Bus Monitor Connections

The I/O Header on the Pocket Bus Monitor accepts the JST PHDR-24VS Socket Housing. [Click here for the full data sheet for the JST PHD connector series](#).



Analog Devices Evaluation Platform Connectors

The A-side and B-side on most Analog Devices evaluation boards use Molex DuraClik connectors (Molex part number 502352-0200).

- When looking into the connectors on the evaluation platform, pin 1 is on the left side and pin 2 is on the right side.
- On the ADI evaluation boards, the polarity of the A^2B signals is flipped between the A-side and the B-side.
- On the A-side, which faces the main node, pin 1 is positive and pin 2 is negative.
- On the B-side, which faces the next sub node, pin 1 is negative and pin 2 is positive.

After completing the steps in this procedure, you will have A-side and B-side cables that connect the Bus Monitor to an Analog Devices evaluation platform.

Fabricating Custom Cables

After completing the steps in this procedure, you will have A-side and B-side cables that connect the Pocket Bus Monitor to an Analog Devices evaluation platform.

You might need to craft your own custom cables to connect the Pocket Bus Monitor to your A^2B network. This section describes how to fabricate A^2B cables with plugs for the Pocket Bus Monitor on one side and connectors for Analog Devices evaluation platforms on the other.

NOTE**: **Adjust these instructions for interconnects used by your own hardware.

***NOTE:** When connecting the A-side or B-side of an A^2B Pocket Bus Monitor to an ADI evaluation platform, you can start with the provided cable harness with DuraClik connectors.*

Bill of Materials

Item	Qty	Mfg.	Mfg. P/N	Dist.	Dist. P/N
DuraClik plug	2	Molex	502351-0200	Mouser	538-502351-020
Duraclik Crimp Terminal	4	Molex	50212-8100	Mouser	538-50212-8100
Duraclik Crimp Tool	1	Molex	63823-5100	Mouser	538-63823-5100
JST Socket Housing	1	JST	JST PHDR-24VS	Digikey	455-1177-ND
JST Terminal	7	JST	SPHD-001T-P0.5	DigiKey	455-1325-1-ND
Alternate Crimp Tool		Engineer	PA-09	Amazon	PA-09
Wire, Brown, 24AWG		Any	UL1061 24AWG		
Wire, White, 24 AWG		Any	UL106124AWG		

Cable, 2 Cond, 24 AWG, Black		Belden	1353A 010	DigiKey	BEL1253-1000-ND

Procedure

1. If using discrete wires, cut the white and brown wires to the desired length considering that the wires must be twisted pairs. If using the Belden cable, cut to the desired length. Ensure no copper is exposed while separating the conductors.

2. If using discrete wires for the A^2B cables, twist the wires at 0.4 twist per cm or 1 twist per inch.

3. Strip and crimp DuraClik terminals onto one end of the wires or cable using a Molex compatible crimp tool.

4. Insert the terminals into the DuraClik plug, carefully noting the polarity of the cables and the placement of the conductors.

5. For the A-side cable, insert the white wire into pin 1 (left side) of the DuraClik plug. Insert brown wire into pin 2. For the B-side cable, insert the brown wire into pin 1 (left side) of the DuraClik plug. Insert the white wire into pin 2.

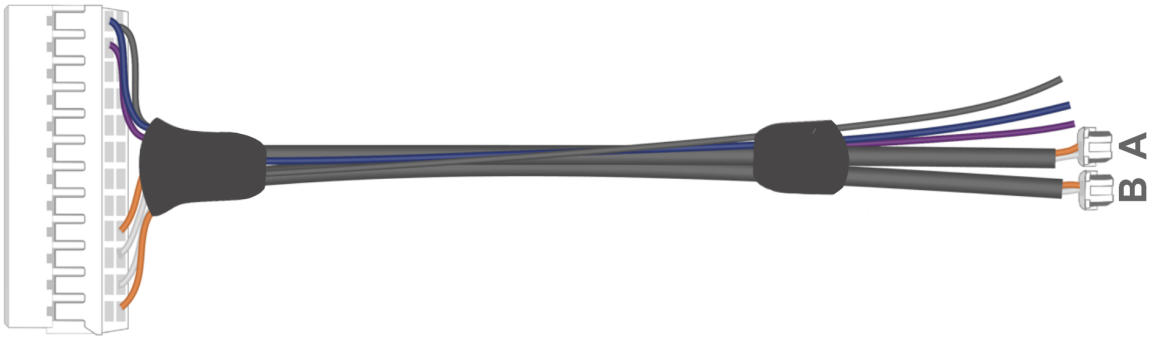
6. (Optional) Apply shrink tubing to dress the end of the cable.

7. Label your A-side and B-side cables.

8. Crimp JST terminals onto the opposite end using a JST compatible crimp tool.

9. Insert terminals into the JST plug, carefully selecting the terminal slots with polarity as shown in the JST 24 Pin I/O Header Pinout.

10. The following figure shows the finished cables.



Chapter 6. Specifications

This chapter provides technical specifications for the Pocket A²B Bus Monitor.

Environmental

DC Characteristics
Target Power: USB +5V, 350mA max
GPIO Signal: 3.3V, 10 mA

Dimensions (W x D x L)
68 x 43 x 25 mm (2.7” x 1.8 x 1”)

Weight
64 g (0.14 lbs)

Operating Temperature
0° C to 70° C (32° C to 158° F)

Ordering Information

A²B Bus Monitor
Part Number: AKT-1500
Country of Origin: USA
HTS: 8473.30.1180
ECCN: EAR99

Chapter 7. Troubleshooting

Issue	Possible Cause	Solution
Status LED Flashing Red	The device has an invalid or missing feature license key	Check to see if you have a copy of the license file and re-install it on the internal sf: filesystem. If not, contact support at flextechakt.com
I/O LED Steady Red	Bias reversed on A^{2}B	See Connect the Pocket Bus Monitor into an A^{2}B Network
No USB Audio in Windows	Windows Audio Enhancement is On for the A^{2}B Bus Monitor	Open Windows Sound Settings, Select the A^{2}B Bus Monitor audio device and confirm Audio Enhancements is Off.
No audio or SPI	**Has your DATCTL register been properly configured?** If not, the Bus Monitor cannot capture audio from the bus	Confirm bit 5 (0x20) of the DATCTL register is set during discovery
A^{2}B Bus Monitor not showing up as an audio device	Windows versions prior to Windows 10, release 1703 do not support the UAC2 USB audio protocol	Use a newer release of Windows
USB Audio dropouts	Windows 10/11 are not real-time operating systems and can fail to transfer USB audio in a timely manner.	Use a different host PC platform, like Linux or Mac, if USB audio is critical.
Extremely high bit errors detected though A^{2}B appears to function normally	TXLEVEL set too low in the upstream node A2B_TXCTL register.	The A^{2}B Bus Monitor input circuit is only compatible with the High Transmit Power Setting (0b00)