

Amtery Programmable Attenuator Programming Guide

May 2026

Content

1.	Overview	2
1.1.	Scope.....	2
1.2.	Introduction	2
1.3.	Getting started	3
1.4.	Programming methods	3
1.5.	Installed content	3
2.	API in DLL	5
2.1.	API List.....	5
2.2.	API Quick Search Table.....	9
3.	ASCII Commands for Programmable Attenuator Control	10
3.1.	Using ASCII Commands	10
3.2.	ASCII Commands List.....	10
3.3.	ASCII Commands Quick Search Table.....	13

1. Overview

1.1. Scope

This guide is designed to help customers develop their own software for controlling Amtery programmable attenuators.

1.2. Introduction

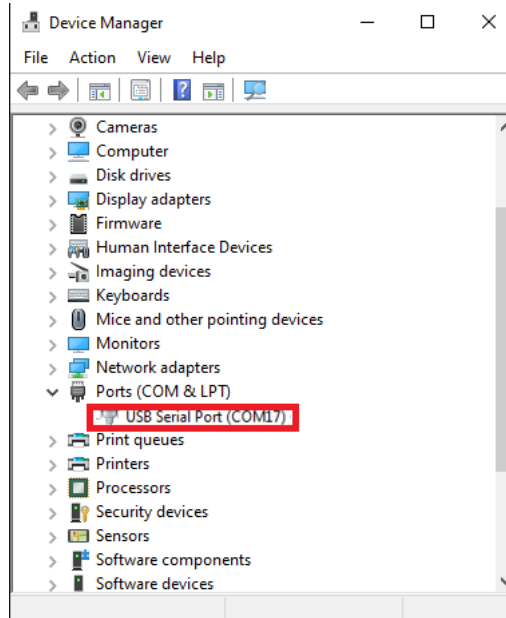
The Amtery programmable attenuator is an RF device that you control via a computer. It features built-in memory that stores its inherent insertion loss versus frequency and can compensate for that loss to deliver precise total attenuation.

It operates in two modes: **Compensation ON** and **Compensation OFF**. When compensation is **ON**, the attenuator looks up the stored insertion loss at the specified frequency, subtracts that from your target attenuation, and applies the remaining amount so that the *total* attenuation equals your setpoint. For example, if you ask for 10 dB of attenuation at 6 GHz and the stored insertion loss at 6 GHz is 4.5 dB, the attenuator will actually set itself to 5.5 dB so that you still get a total of 10 dB.

When compensation is **OFF**, the device ignores its insertion loss and simply applies the exact attenuation you request, resulting in the sum of your set attenuation plus the inherent insertion loss.

1.3. Getting started

After installing the Amtery attenuator driver, connect your attenuator to a USB port. Windows will list it as “**USB Serial Port**” under **Ports (COM & LPT)** in Device Manager. If it doesn’t appear—and you have antivirus software running—temporarily disable your antivirus, unplug the attenuator, and then plug it back in again.



A. Troubleshooting:

If the device is not detected in the COM port list, manually install the FTDI driver.

- i. Navigate to the software installation directory. (default: C:\Program Files (x86)\Amtery\Amtery Attenuator)
- ii. Open Scripts\USB_Driver.
- iii. Run dpinst-amd64.exe (for 64-bit systems).
Run dpinst-x86.exe (for 32-bit systems).

1.4. Programming methods

Amtery programmable attenuators offer three different interfaces for software control:

- A. DLL API
- B. ASCII command
- C. LabVIEW driver

1.5. Installed content

DLL, documents and examples are installed in three sub folders of “C:\Program Files (x86)\Amtery\Amtery Attenuator”:

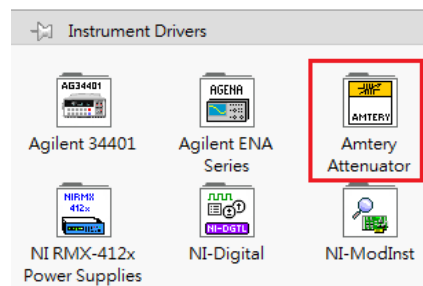
- A. A 32-bit (x86) DLL file, AmteryAtt.dll, and header file, AmteryAtt.h, are in “Bin”.
- B. Software Panel Guide and Programing Guide are in “Documents”.

- C. Examples for LabVIEW and C/C++ are in “Examples”.
- C/C++ API DLL examples are in “Examples\Cpp API DLL examples”.
 - LabVIEW API DLL examples are in “Examples\LabVIEW API DLL examples”.
 - C/C++ ASCII command examples are in “Examples\Cpp ASCII command examples”.
 - LabVIEW driver examples are in “Examples\LabVIEW driver examples”.

The 64-bit (x64) DLL is in “C:\Program Files (x86)\Amtery\Amtery Attenuator\Bin\x64”. C/C++ API DLL and LabVIEW API DLL examples are written with 32-bit (x86) DLL. If 64-bit development environment is used, please replace the DLL with 64-bit (x64) DLL.

LabVIEW Driver Installation Steps:

- Open the folder “C:\Program Files (x86)\Amtery\Amtery Attenuator\Examples\LabVIEW driver examples”.
- Copy the entire “Amtery Attenuator” folder into “\LabVIEW\instr.lib”. For a 32-bit LabVIEW 2020 installation, that would be “C:\Program Files (x86)\National Instruments\LabVIEW 2020\instr.lib”.
- “Amtery Attenuator” will appear in “LabVIEW instrument I/O -> instrument drivers” palette.



2. API in DLL

The **AmteryAtt.dll** library provides the API functions for programmatic configuration and control of the attenuator.

2.1. API List

A) Reset device

Command Syntax			Description
int amtAttReset(char* portName)			Reset device.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char*		Virtual COM port number that the device connects to the PC.
return	int		0: Success. -1: Fail.

B) Query device information

Command Syntax			Description
<code>int amtAttGetDeviceInfo(char* portName, char* deviceInfo)</code>			Read the device information.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	<code>char*</code>		Virtual COM port number that the device connects to the PC.
deviceInfo	<code>char*</code>		Device information. The return string is “model name” + “firmware version” + “SN”. Ex: Amtery AT10006B-30-U ver0.98, SN:E0EE15FB1C00
return	<code>int</code>		0: Success. -1: Fail.

C) Set attenuation

Command Syntax			Description
<code>int amtAttSetAtt(char* portName, int channel, float attenuation, int frequency, int compensation, char* response)</code>			Set the attenuation, frequency, compensation mode on the specific channel.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char*		Virtual COM port number that the device connects to the PC.
channel	int		Specify the channel number, or 0 is controlling all channels.
attenuation	float		Attenuation value in dB.
frequency	int		If compensation is ON, set working frequency in MHz. If compensation is OFF, can be any frequency within specification.
compensation	int		1: Enable compensation, set attenuation including insertion loss ¹ . 0: Disable compensation, set attenuation excluding insertion loss ² .
response	char*		Device response: status or error information.
return	int		0: Success. -1: Fail.



Attention

1. If compensation mode is ON, since the attenuation includes insertion loss, the acceptable minimum attenuation setting is insertion loss. The real attenuation = the set attenuation.
2. If compensation mode is OFF, the attenuation can be set from 0 to the maximum value, the real attenuation = insertion loss + the set attenuation. And the frequency setting will not affect the attenuation, but user still needs to set a frequency within the specification.

D) Query device status

Command Syntax			Description
<code>int amtAttGetAtt(char* portName, int channel, float* attenuation, int* frequency, int* compensation, char* response)</code>			Read device status, including attenuation, frequency, and compensation setting.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char*		Virtual COM port number that the device connects to the PC.
channel	int		Specify the channel number.
attenuation	float*		Read attenuation value in dB.
frequency	int*		Read frequency in MHz.
compensation	int*		Read compensation mode: 1: Compensation enabled. 0: Compensation disabled.
response	char*		Device response includes device status or error information.
return	int		0: Success. -1: Fail.

E) Query device Specification

Command Syntax			Description
<code>int amtAttGetDeviceSpec(char* portName, int* totalChannel, float* attenuationStep, float* attenuationMax, int* minFrequency, int* maxFrequency)</code>			Read the device specification, including step of attenuation, maximum attenuation, number of all channels, minimum frequency, and maximum frequency.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char *		Virtual COM port number that the device connects to the PC.
totalChannel	int *		Read the number of all channels.
attenuationStep	float *		Read the attenuation step in dB.
attenuationMax	float *		Read the maximum attenuation in dB.
minFrequency	int *		Read the minimum frequency in MHz.
maxFrequency	int *		Read the maximum frequency in MHz.
return	int		0: Success. -1: Fail.

F) Query device insertion loss

Command Syntax			Description
<code>int amtAttGetInsertionLoss(char* portName, int channel, int frequency, float* insertionLoss)</code>			Read the insertion loss at specific channel and frequency.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char*		Virtual COM port number that the device connects to the PC
channel	int		Specify the channel number.
frequency	int		Specify the frequency in MHz.
insertionLoss	float*		Read the insertion loss at the specific channel and frequency.
return	int		0: Success. -1: Fail.

G) Save as default settings

Command Syntax			Description
<code>int amtAttSaveAsDefault(char* portName)</code>			Save the current settings as default used at startup.
Arguments	Data type		Description
	C/C++	LabVIEW	
portName	char*		Virtual COM port number that the device connects to the PC.
return	int		0: Success. -1: Fail.

2.2. API Quick Search Table

Function	Syntax
Reset device	<code>int amtAttReset(char* portName)</code>
Query device information	<code>int amtAttGetDeviceInfo(char* portName, char* deviceInfo)</code>
Set attenuation	<code>int amtAttSetAtt(char* portName, int channel, float attenuation, int frequency, int compensation, char* response)</code>
Query device status	<code>int amtAttGetAtt(char* portName, int channel, float* attenuation, int* frequency, int* compensation, char* response)</code>
Query device specification	<code>int amtAttGetDeviceSpec(char* portName, int* totalChannel, float* attenuationStep, float* attenuationMax, int* minFrequency, int* maxFrequency)</code>
Query device insertion loss	<code>int amtAttGetInsertionLoss(char* portName, int channel, int frequency, float* insertionLoss)</code>
Save as default settings	<code>int amtAttSaveAsDefault(char* portName)</code>

3. ASCII Commands for Programmable Attenuator Control

These ASCII commands directly control Amtery programmable attenuators. All commands are case sensitive. Users can use either DLL or ASCII to control the devices. ASCII commands can be used in common programming languages or serial port communication tools such as "PuTTY" , configured with a baud rate of 115200.

3.1. Using ASCII Commands

When sending ASCII commands, append a CRC16 checksum and the 0x0A terminator to each message. For example, the *RST command becomes (in hexadecimal) : 2A52 5354 00AA 160A, Here, 0xAA16 is the CRC16 checksum and 0x0A marks the end of the command.

Amtery programmable attenuator ASCII commands are SCPI-like but not standard SCPI. Therefore, users cannot cascade multiple commands. Every single command needs to be sent separately.

3.2. ASCII Commands List

A) Reset device

Command Syntax	Description	Example
*RST	Reset device.	*RST
Arguments	Description	
n/a	n/a	
Return String	Example	
If reset successfully.	"0": Success. Any value but 0: Fail.	

B) Query device information

Command Syntax	Description	Example
*IDN?	Read device information.	*IDN?
Arguments	Description	
n/a	n/a	
Return String	Example	
"Model name" + "Firmware version" + "SN".	"Amtery AT10006B-30-U ver0.98, SN:E0EE15FB1C00."	

C) Set attenuation

Command Syntax	Description	Example
ATT <CH>,<ATTN>	Set the attenuation on the specific channel and the compensation is OFF automatically.	ATT 1,5
ATT <CH>,<ATTN>,<FREQ>,<COMP>	Set the attenuation, frequency,	ATT 1,5,1000,1

	compensation mode on the specific channel.	
Arguments	Description	
CH	Specify the channel number, or 0 is controlling all channels.	
ATTN	Attenuation value in dB.	
FREQ	If compensation is ON, set working frequency in MHz. If compensation is OFF, can be any frequency within specification.	
COMP	1: Enable compensation, set attenuation including insertion loss ¹ . 0: Disable compensation, set attenuation excluding insertion loss ² .	
Return String	Example	
"Channel", "Attenuation", "Frequency", "Compensation Mode".	"1,5.00,1000,1"	
Error information.	"Error, the min frequency is 15 MHz."	



Attention

1. If compensation mode is ON, since the attenuation includes insertion loss, the acceptable minimum attenuation setting is insertion loss. The real attenuation = the set attenuation.
2. If compensation mode is OFF, the attenuation can be set from 0 to the maximum value, the real attenuation = insertion loss + the set attenuation. And the frequency setting will not affect the attenuation, but user still needs to set a frequency within the specification.

D) Query device status

Command Syntax	Description	Example
ATT? <CH>	Read device status, including attenuation, frequency, and compensation setting.	ATT? 1
Arguments	Description	
CH	Specify the channel number.	
Return String	Example	
"Channel", "Attenuation", "Frequency", "Compensation Mode". Frequency in MHz. Attenuation in dB. Compensation mode: 1 is enabled. 0 is disabled.	"1,5.00,1000,0"	
Error information.	"Error, invalid ch:2"	

E) Query device Specification

Command Syntax	Description	Example
SPEC?	Read the device specification, including number of all channels, step of attenuation, maximum attenuation, minimum frequency, and maximum frequency.	SPEC?
Arguments	Description	
n/a	n/a	
Return String	Example	
"Total Number of Channels","Attenuation Step","Maximum Attenuation","Minimum Frequency","Maximum Frequency".	"1,0.250,63.500,200,6000"	

F) Query device insertion loss

Command Syntax	Description	Example
IL? <CH>,<FREQ>	Read the insertion loss at specific channel and frequency.	IL? 1,2520
Arguments	Description	
CH	Specify the channel number.	
FREQ	Specify the frequency in MHz.	
Return String	Example	
Insertion loss in dB.	"1.359"	
Error information.	"Error, invalid frequency input value."	

G) Save settings as default

Command Syntax	Description	Example
*SAV	Save the current settings as default used at startup.	*SAV
Arguments	Description	
n/a	n/a	
Return String	Example	
If save successfully.	"0": Success. Any value but 0: Fail.	

3.3. ASCII Commands Quick Search Table

Function	Command Syntax
Reset device	*RST
Query device info	*IDN?
Set attenuation	ATT <CH>,<ATTN>,<FREQ>,<COMP>
Query device status	ATT? <CH>
Query device specification	SPEC?
Query device insertion loss	IL? <CH>,<FREQ>
Save as default settings	*SAV