

# **Amtery Programmable Attenuator Programming Guide**

December 2022

# Content

1.	Overview .....	3
1.1.	Scope.....	3
1.2.	Introduction .....	3
1.3.	Getting started .....	3
1.4.	Programming methods .....	4
1.5.	Installed content .....	4
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.....	12

# 1. Overview

## 1.1. Scope

This programming guide is intended for customers to create their own programs to control Amtery programmable attenuators.

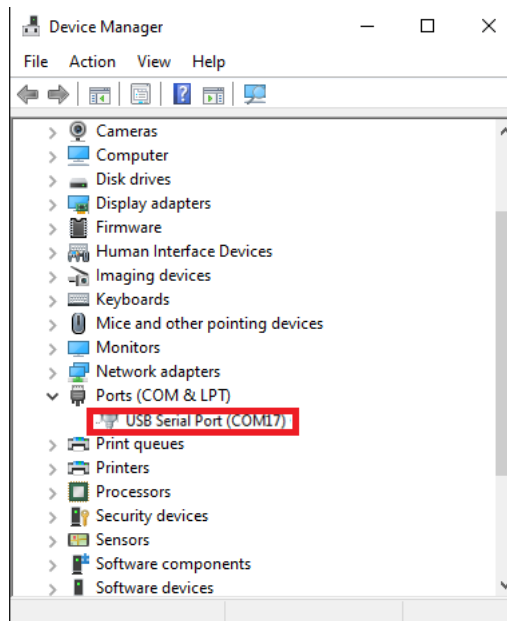
## 1.2. Introduction

Amtery programmable attenuator is a RF programmable attenuator controlled by a computer. It has built-in memory to store the inherent insertion loss and can compensate the insertion loss to provide the accurate total attenuation.

It has two operating modes: compensation ON and OFF. If the compensation mode is ON, the attenuator reads the insertion loss in its memory, calculates the insufficient attenuation, and set the remaining attenuation automatically to fill the gap. The user will get "set attenuation". (Please note the insertion loss increases with frequency, so the wanted frequency must be specified.) For example, a user sets 10 dB attenuation at 6 GHz, the attenuator will read the memory and get the insertion loss at 6 GHz, such as 4.5 dB. Then the attenuator will set the actual attenuation automatically to 5.5 dB to provide 10 dB attenuation totally. If the compensation is OFF, the attenuator disregards the insertion loss and just set the attenuation. The user will get "set attenuation + insertion loss".

## 1.3. Getting started

After Amtery attenuator driver installation is finished, plug in an Amtery programmable attenuator into a USB port, Windows device manager will show "USB Serial Port" in Ports (COM & LPT). If "USB Serial Port" doesn't pop up and anti-virus software is running, please close anti-virus software, unplug and plug in again.



#### 1.4. Programming methods

Amtery attenuators support three methods of programmatically device control:

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

#### 1.5. Installed content

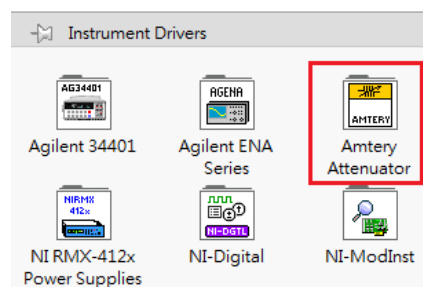
DLL, documents and examples are installed in three sub folders of “C:\Program Files (x86)\Amtery\Amtery Attenuator”:

- A. A 32-bit (x86) DLL file, amtery\_attenuator.dll, and header file, amtery\_attenuator.h, are in “Bin”.
- B. Software Panel Guide and Programming Guide are in “Documents”.
- C. Examples for LabVIEW and C/C++ are in “Programming Examples”.
  - i. C/C++ API DLL examples are in “Programming Examples\Cpp Examples”.
  - ii. LabVIEW API DLL examples are in “Programming Examples\LabVIEW (32-bit) Examples”.
  - iii. C/C++ ASCII command examples are in “Programming Examples\ASCII command Cpp Examples”.
  - iv. LabVIEW ASCII command examples are in “Programming Examples\ASCII command Labview 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.

Steps to install LabVIEW driver:

- A. Find folder “C:\Program Files (x86)\Amtery\Amtery Attenuator\Programming Examples\ASCII command Labview Examples”.
- B. Manually copy or move the folder “ASCII command Labview Examples” to “\LabVIEW\instr.lib”. For example, if a user has installed LabVIEW 2019 32-bit, the destination folder is “C:\Program Files (x86)\National Instruments\LabVIEW 2019\instr.lib”.
- C. “Amtery Attenuator” will appear in “LabVIEW instrument I/O -> instrument drivers” palette.





## 2. API in DLL




The DLL file amtery\_attenuator.dll contains API to control the attenuator.

### 2.1. API List








#### A) Reset device

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

C) Set attenuation








Command Syntax		Description	
<code>int amtAttSetAtt(char* portName, int channel, float attenuation, char* response)</code>		Set the attenuation on the specific channel and the compensation mode is OFF automatically.	
<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 <sup>1</sup> . 0: Disable compensation, set attenuation excluding insertion loss <sup>2</sup> .
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.



## 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>
Set attenuation for disables compensation mode	<code>int amtAttSetAtt(char* portName, int channel, float attenuation, , 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>

### 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".

#### 3.1. Using ASCII Commands

A terminal character 0x0D "carriage return" is needed at the end of all commands sent.

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

#### 3.2. ASCII Commands List

A terminal character 0x0D "carriage return" is needed at the end of all commands sent.

##### 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, compensation mode on the specific channel.	ATT 1,5,1000,1
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 <sup>1</sup> . 0: Disable compensation, set attenuation excluding insertion loss <sup>2</sup> .
<b>Return String</b>	<b>Example</b>
"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
<b>Arguments</b>	<b>Description</b>	
CH	Specify the channel number.	
<b>Return String</b>	<b>Example</b>	
"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."

3.3. ASCII Commands Quick Search Table

Function	Command Syntax
Reset device	*RST
Query device info	*IDN?
Set attenuation	ATT <CH>,<ATTN>,<FREQ>,<COMP>
Set attenuation for disables compensation mode	ATT <CH>,<ATTN>
Query device status	ATT? <CH>
Query device specification	SPEC?
Query device insertion loss	IL? <CH>,<FREQ>