

Amtery RF Synthesizer Programming Guide

Content

1.	Overview	2
1.1.	Scope	2
1.2.	Getting started	2
1.3.	Programming methods.....	2
1.4.	Installed content.....	2
2.	API in DLL	3
2.1.	API List.....	3
3.	ASCII Commands for RF Synthesizer Control	8
3.1.	Using ASCII Commands	8
3.2.	ASCII Commands List.....	8

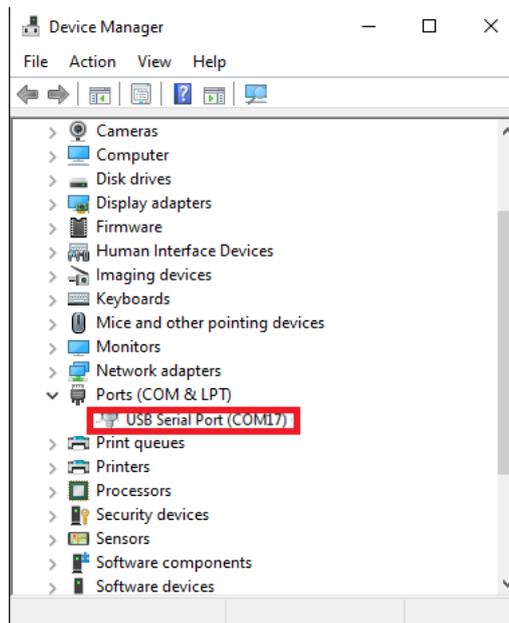
1. Overview

1.1. Scope

This guide is designed to help customers develop their own software for controlling Amtery RF synthesizer.

1.2. Getting started

After installing the Amtery synthesizer driver, connect your synthesizer 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 synthesizer, and then plug it back in again.



1.3. Programming methods

Amtery RF synthesizer offers three different interfaces for software control:

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

1.4. Installed content

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

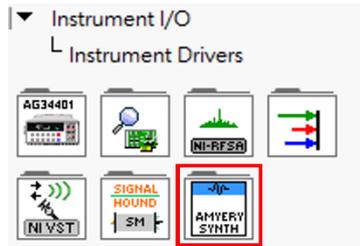
- A. A 32-bit (x86) DLL file, `Amtery_SynthDll`, and header file, `Amtery_SynthDll.h`, are in `“Bin”`.
- B. Software Panel Guide and Programming Guide are in `“Documents”`.
- C. Examples are in `“Examples”`.
 - i. C/C++ API DLL examples are in `“Examples\Cpp API DLL examples”`.
 - ii. LabVIEW API DLL examples are in `“Examples\LabVIEW API DLL examples”`.

- iii. C/C++ ASCII command examples are in “Examples\Cpp ASCII command examples”.
- iv. LabVIEW driver examples are in “Examples\ LabVIEW driver examples”.

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

Steps to install LabVIEW driver:

- A. Open folder “C:\Program Files (x86)\Amtery\Amtery Synthesizer\ Examples\LabVIEW driver Examples”.
- B. Copy the entire “Amtery Synthesizer” 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”.
- C. “Amtery Synthesizer” will appear in “LabVIEW instrument I/O -> instrument drivers” palette.



2. API in DLL

Amtery_SynthDll.dll library provides the API functions for programmatic configuration and control of the synthesizer.

2.1. API List

A) Reset Device

Command Syntax		Description
<code>int amtSynthReset(char* portName)</code>		Reset the device.
Arguments	Description	
portName	Virtual COM port number of the device.	
return	0: Success. -1: Fail.	

B) Query Device Information

Command Syntax		Description
<code>int amtSynthGetDeviceInfo(char* portName, char* deviceInfo)</code>		Read the device information.

Arguments	Description
portName	Virtual COM port number of the device.
deviceInfo	Device information. The return string is “model name” + “firmware version” + “SN”. Ex: Amtery SYN0020A-U ver1.3, SN:E0EE15FB1C00
return	0: Success. -1: Fail.

C) Start Continuous Wave

Command Syntax	Description
<code>int amtSynthStartCW(char* portName, int channel, double frequency, double power, char* response)</code>	Set the continuous wave parameters and start generation.
Arguments	Description
portName	Virtual COM port number of the device.
channel	Specify the channel number. Input “0” to control all channels.
frequency	Output frequency in MHz.
power	Output power level in dBm.
response	Device response: status or error information.
return	0: Success. -1: Fail.

D) Start Pulse Modulation

Command Syntax	Description
<code>int amtSynthStartPulseMod(char* portName, int channel, int source, double frequency, double power, double pulseWidth, double repetitionRate, char* response)</code>	Set the pulse modulation parameters and start generation.
Arguments	Description
portName	Virtual COM port number of the device.
channel	Specify the channel number. Input “0” to control all channels.
source	Source of pulse: 0: Use internal pulse. 1: Use external pulse.
frequency	Output frequency in MHz.
power	Output power in dBm.
pulseWidth	Pulse width in μ s. Ignored if using external source.
repetitionRate	Repetition rate in kHz. Ignored if using external source.

response	Device response: status or error information.
return	0: Success. -1: Fail.

E) Start Frequency List

Command Syntax		Description
<code>int amtSynthStartFreqList (char* portName, int channel, double startFrequency, double stopFrequency, double power, double step, int time, int loop, char* response)</code>		Set frequency list parameters and start generation.
Arguments	Description	
portName	Virtual COM port number of the device.	
channel	Specify the channel number. Input "0" to control all channels.	
startFrequency	The start frequency of the list in MHz.	
stopFrequency	The stop frequency of the list in MHz.	
power	Output power in dBm.	
step	The incremental frequency change in MHz.	
time	The dwelling time at each frequency in ms.	
loop	Number of repetitions of the list. Input "0" to loop infinitely.	
response	Device response: status or error information.	
return	0: Success. -1: Fail.	

F) Start Power List

Command Syntax		Description
<code>int amtSynthStartPwrList (char* portName, int channel, double startPower, double stopPower, double step, int time, int loop, char* response)</code>		Set power list parameters and start generation.
Arguments	Description	
portName	Virtual COM port number of the device.	
channel	Specify the channel number. Input "0" to control all channels.	
frequency	Output frequency in MHz.	
startPower	The start power of the list in dBm.	
stopPower	The stop power of the list in dBm.	
step	The incremental power change in dB.	
time	The dwelling time at each power value in ms.	
loop	Number of repetitions of the list. Input "0" to loop infinitely.	

response	Device response: status or error information.
return	0: Success. -1: Fail.

G) Get Generation Status

Command Syntax		Description
<code>int amtSynthGetGenStatus(char* portName, int channel)</code>		Get the generation status.
Arguments	Description	
portName	Virtual COM port number of the device.	
channel	Specify the channel number.	
return	0: Stopped. 1: Generating.	

H) Stop Generation

Command Syntax		Description
<code>int amtSynthStop(char* portName, int channel)</code>		Disable the internal frequency generation circuitry, effectively halting RF output.
Arguments	Description	
portName	Virtual COM port number of the device.	
channel	Specify the channel number. Input "0" to stop all channels.	
return	0: Success. -1: Fail.	

I) Set Reference Clock

Command Syntax		Description
<code>int amtSynthSetRefClk(char* portName, int source, double frequency, char* response)</code>		Set the reference clock.
Arguments	Description	
portName	Virtual COM port number of the device.	
source	Clock source: 0: Use internal reference clock. 1: Use external reference clock.	
frequency	Frequency of reference clock in MHz. Ignored if using internal source.	
response	Device response: status or error information.	
return	0: Success.	

	-1: Fail.
--	-----------

J) Get Reference Clock

Command Syntax		Description
<code>int amtSynthGetRefClk(char* portName, int source, double* frequency)</code>		Get the reference clock.
Arguments	Description	
portName	Virtual COM port number of the device.	
source	Clock source: 0: Internal reference clock. 1: External reference clock.	
frequency	Frequency of reference clock in MHz.	
return	0: Success. -1: Fail.	

3. ASCII Commands for RF Synthesizer Control

These ASCII commands directly control Amtery RF synthesizer. 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 15200.

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 RF synthesizer 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 the device.	*RST
Arguments	Description	
n/a	n/a	
Return String	Example	
”Status” or error information	”0”: Success. Negative value: 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 SYN0020B-U ver1.3, SN:E0EE15FB1C00.”	

C) Start Continuous Wave

Command Syntax	Description	Example
CW <CH> <FREQ> <PWR> CW <CH> <FREQ> <PWR>	Set the continuous wave parameters and start generation.	CW 1 1000 -10
Arguments	Description	
CH	Specify the channel number. Input ”0” to control	

	all channels.
FREQ	Output frequency in MHz.
PWR	Output power level in dBm.
Return String	Example
“Status” or error information	“0”: Success. Negative value: Fail.

D) Start Pulse Modulation

Command Syntax	Description	Example
PWM <CH> <SOURCE> <FREQ> <PWR> <WIDTH> <RATE>	Set the pulse modulation parameters and start generation.	PWM 1 0 1000 0 50 10
Arguments	Description	
CH	Specify the channel number. Input “0” to control all channels.	
SOURCE	Source of pulse: 0: Use internal pulse. 1: Use external pulse.	
FREQ	Output frequency in MHz.	
PWR	Output power in dBm.	
WIDTH	Pulse width in μ s. Ignored if using external source.	
RATE	Repetition rate in kHz. Ignored if using external source.	
Return String	Example	
“Status” or error information.	“0”: Success. Negative value: Fail.	

E) Start Frequency List

Command Syntax	Description	Example
LS_FREQ <CH> <STARTFREQ> <STOPFREQ> <PWR> <STEP> <TIME> <LOOP>	Set frequency list parameters and start generation.	LS_FREQ 1 100 1000 100 -10 20 1
Arguments	Description	
CH	Specify the channel number. Input “0” to control all channels.	
STARTFREQ	The start frequency of the list in MHz.	
STOPFREQ	The stop frequency of the list in MHz.	
STEP	The incremental frequency change in MHz.	

PWR	Output power in dBm.
TIME	The dwelling time at each frequency in ms.
LOOP	Number of repetitions of the list. Input "0" to loop infinitely.
Return String	Example
"Status" or error information.	"0": Success. Negative value: Fail.

F) Start Power List

Command Syntax	Description	Example
LS_PWR <CH> <FREQ> <STARTPWR> <STOPPWR> <STEP> <TIME> <LOOP>	Set power list parameters and start generation.	LS_PWR 1 1000 -5 0 1 20 1
Arguments	Description	
CH	Specify the channel number. Input "0" to control all channels.	
FREQ	Output frequency in MHz.	
STARTPWR	The start power of the list in dBm	
STOPPWR	The stop power of the list in dBm.	
STEP	The incremental power change in dB.	
TIME	The dwelling time at each power value in ms.	
LOOP	Number of repetitions of the list. Input "0" to loop infinitely.	
Return String	Example	
"Status" or error information.	"0": Success. Negative value: Fail.	

G) Get Generation Status

Command Syntax	Description	Example
RF? <CH>	Get the generation status.	RF? 1
Arguments	Description	
CH	Specify the channel number.	
Return String	Example	
"Status" or error information.	0: Stopped. 1: Generating.	

H) Stop Generation

Command Syntax	Description	Example
STOP <CH>	Disable the internal frequency generation circuitry, effectively halting RF output.	STOP 1
Arguments		Description
CH	Specify the channel number. Input "0" to control all channels.	
Return String		Example
"Status" or error information.		0: Success. Negative value: Fail.

I) Set Reference Clock

Command Syntax	Description	Example
REF <SOURCE> <FREQ>	Set the reference clock.	REF 0
Arguments		Description
SOURCE	Clock source: 0: Use internal reference clock. 1: Use external reference clock.	
FREQ	Frequency of reference clock in MHz. Ignored if using internal source.	
Return String		Example
"Status" or error information.		"0": Success. Negative value: Fail.

J) Get Reference Clock

Command Syntax	Description	Example
REF?	Get the reference clock.	REF?
Arguments		Description
n/a		n/a
Return String		Example
"Clock Source", "Frequency". Clock source: 0: Internal reference clock. 1: External reference clock.		"0, 100.0"