



emccrc.dll(for .net platform) instruction

CONTENTS

1. COMMONMODULE.....	2
2. EMC BENCHTOP AMPLIFIER.....	3
3. CONTACT.....	6

1. commonModule

Common module for all USB+LAN products.

1.1 **string[]** getCOMnames()

Get all COM ports for this PC.
Return the COM ports.

1.2 **SerialPort** openConnectUSB(**string** comPort,**int** baudrate)

Connect the serial port with baudrate. StopBits = One, DataBits = 8, Parity = None.
Return the serial port of COMx.

Notes: The baudrate is 115200 by default or 9600, please refer to the product manual.

1.3 **void** closeConnectUSB(**SerialPort** usbPort)

Close the connection for serial port.

1.4 **string** getProductInformationUSB(**SerialPort** usbPort)

Get the product information by serial port.
Return: PN, SN, manufacturer, version.

1.5 **void** resetDeviceUSB(**SerialPort** usbPort)

Restart the product. No characters returned.

1.6 **string** getMAC_USB(**SerialPort** usbPort)

Query the MAC of device, return the MAC.

1.7 **string** setBaudrateUSB(**SerialPort** usbPort,**int** baudrate)

Set the baudrate for the serial port, return 'Baud:115200' or 'Baud:9600'.
This function will take effect after restart.
Parameters: baudrate, only be 0 or 1. 0-9600; 1-115200.

1.8 **string** queryErrorDescription(**string** errorCode)

Find the error description by error code.

Notes: The error code can be found on the product return string, and it will only contain: E1, E2, E3, E4, E5.

1.9 **string[]** getLocalIPs()

Get the IP address for all IPv4 in this PC.
Return IP address.

1.10 **Socket** openConnectTCP(**string** localIP, **string** productIP,**int** productPort)

Open the TCP connection. Return the handle of a TCP connection socket.
The default product IP address is: 192.168.1.225, and the default port is: 5100.
You can use 'EMC Amplifier Control Panel' to query or modify it. The range of port is: 2000-65535.

1.11 **void** closeConnectTCP(**Socket** TCPConnect)

Close the TCP connection.

1.12 **string** getProductInformationTCP(**Socket** TCPConnect)

Get the product information by TCP connect.
Return: PN, SN, manufacturer, version.

1.13 **void** resetDeviceTCP(**Socket** tcpConnect)

Restart the product. No characters returned.

1.14 **string** getMAC_TCP(**Socket** tcpConnect)

Query the MAC of this product, return the MAC.

2. EMC Benchtop Amplifier

2.1 **string** setOutputStatusUSB(**SerialPort** usbPort, **bool** status)

Set the output ON or OFF. String 'SS:x'.

Parameters: status-true means open the output, and false means close.

2.2 **string** setFrequencyUSB(**SerialPort** usbPort, **decimal** frequency)

Set the frequency. Return SF:x.

Parameters: frequency-the frequency value, and unit in GHz, 3 decimal places.

2.3 **string** setAttenuationUSB(**SerialPort** usbPort, **decimal** attenuation)

Set the attenuation value. Return SA:x.

Parameters: attenuation-the attenuation, unit in dB, 1 decimal place.

2.4 **string** setOutputPowerUSB(**SerialPort** usbPort, **decimal** outputPower)

Set the output Power. Return SOP:x.

Parameters: outputPower-the output power, unit in dBm, 1 decimal place.

Attention: If the ALC function closed, this command is invalid.

2.5 **string** setAlcStatusUSB(**SerialPort** usbPort, **bool** status)

Set the ALC function ON or OFF. Return SAS:x.

Parameters: status-true means open the output, and false means close.

2.6 **string** setAGCUSB(**SerialPort** usbPort, **bool** status, **decimal** value)

Set AGC(Auto Gain Control) status and value of the EMC product.

Parameters: the status, must be 0 or 1; 0 means close the AGC function, 1 means open the AGC function. Value unit is dB, keep one decimal place, valid only if status = 1.

2.7 **string** SetSweepUSB(**SerialPort** usbPort, **decimal** StartFreq, **decimal** StepFreq, **decimal** StopFreq, **decimal** delay, **decimal** level)

Set the parameters in the sweep frequency mode, including the starting frequency, frequency step, ending frequency, delay, and power.

Parameters: The starting frequency point, frequency step, and ending frequency point of the sweep mode are in GHz. The switching delay unit is in milliseconds. The output power is in dBm.

2.8 **string** SetSweepStatusUSB(**SerialPort** usbPort, **bool** status)

Set the status of the RF switch in the sweep frequency mode.

Parameters: status 1 indicates that the RF switch is turned on in the sweep frequency mode, while status 0 indicates that the RF switch is turned off in the sweep frequency mode.

2.9 **string** SetHopUSB(**SerialPort** usbPort, **decimal** num, **decimal** freq, **decimal** level, **decimal** delay)

Set the parameters for the frequency hopping mode, including the frequency point number, Frequency, power and delay.

Parameters: Serial number of the frequency hopping parameter table, frequency point, unit is GHz, output power, unit is dBm, frequency hopping delay time, unit is milliseconds.

2.10 **string** SetHopStatus(**SerialPort** usbPort, **bool** status)

Set the frequency hopping mode and the status of the radio frequency output switch.

Parameters: When the status is 1, it indicates that the radio frequency output is enabled in the frequency hopping mode; when the status is 0, it indicates that the radio frequency output is disabled in the frequency hopping mode.

2.11 **string** SetSigLevelUSB(**SerialPort** usbPort, **decimal** level)

Set the output power of the signal source at the EMC end.

Parameters: level represents the current power value output by the signal source, with the unit being dBm.

2.12 **string** SetSigModeUSB(**SerialPort** usbPort, **bool** status)

Set the internal and external signal sources of the EMC enclosure.

Parameters: When the status is 1, it indicates the use of the internal signal source; when the status is 0, it indicates the use of the external signal source.

2.13 **bool** getOutputStatusUSB(**SerialPort** usbPort)

Query the output status.

Return false or true; false means closed, and true means open.

2.14 **decimal** getFrequencyUSB(**SerialPort** usbPort)

Query the frequency.

Return the setting frequency, unit in GHz, 3 decimal places.

2.15 **decimal** getAttenuationUSB(**SerialPort** usbPort)

Query the attenuation.

Return the current attenuation, unit in dB, 1 decimal place.

2.16 **decimal** getInputPowerUSB(**SerialPort** usbPort)

Query the input power.

Return the current input power, unit in dBm, 1 decimal place.

2.17 **decimal** getOutputPowerUSB(**SerialPort** usbPort)

Query the output power.

Return the current output power, unit in dBm, 1 decimal place.

2.18 **bool** getAlcStatusUSB(**SerialPort** usbPort, **out decimal** alcPower)

Query the ALC function status.

Return ALC status false or true, false means closed, and true means open.

Parameters: The alcPower will return a decimal value, unit in dBm, 1 decimal place.

2.19 **decimal** getOutputVswrUSB(**SerialPort** usbPort)

Query the output VSWR.

Return the current output VSWR, unit in :1, 2 decimal place

2.20 **decimal** getTemperatureUSB(**SerialPort** usbPort)

Query the current temperature.

Return the current temperature, unit in degC, 1 decimal places.

2.21 **decimal[]** getFrequencyRangeUSB(**SerialPort** usbPort)

Query the product frequency range.

Return the minimum and maximum frequency range, unit in GHz, 3 decimal places.

2.22 **decimal[]** getAttenuationRangeUSB(**SerialPort** usbPort)

Query the product attenuation range.

Return the minimum and maximum attenuation range, unit in dB, 1 decimal places.

2.23 **bool[]** getAlarmUSB(**SerialPort** usbPort)

Query the alarm status.

Return the 6 alarms status. The order is:

- ★ Current (Exceed the maximum current or below the minimum current);
- ★ Current Imbalance;
- ★ Temperature (Exceed the maximum temperature);
- ★ Input Power (Exceed the maximum input power);
- ★ Output VSWR (Exceed the maximum output VSWR).
- ★ General (In addition to the above five kinds of alarm).

Each alarm status is defined by false or true, false means no alarm and true means alarm triggered.

2.24 **bool** GetAGCUSB(**SerialPort** usbPort)

Query AGC (Auto Gain Control) parameters of the EMC product.

Return AGC status, value and current actual gain of the EMC product.

2.25 **string** GetSweepUSB(**SerialPort** usbPort)

Query the parameters of the sweep frequency mode

Return to the initial frequency point, frequency step, termination frequency point, delay, and power.

2.26 **string** GetSweepStatusUSB(**SerialPort** usbPort)

Query the current status of the RF switch in the sweep frequency mode.

Return 1 or 0, 1 indicates the function is enabled, while 0 indicates the function is disabled.

2.27 **string** GetHopUSB(**SerialPort** usbPort, **decimal** num)

Query the parameters of the frequency hopping mode.

Return the frequency band number, frequency, power, and delay.

2.28 **string** GetHopStatus(**SerialPort** usbPort)

Query the current status of the RF switch in the hop frequency mode.

Return 1 or 0, 1 indicates the function is enabled, while 0 indicates the function is disabled.

2.29 **string** GetLevelRange(**SerialPort** usbPort)

Query the output power range of the signal source at the current frequency point.

Return the minimum and maximum values, and round to one decimal place.

2.30 **string** GetSigModeUSB(**SerialPort** usbPort)

Query the current source selection of the EMC chassis.

Return 1 or 0. 1 indicates using the internal signal source, while 0 indicates using the external signal source.

2.31 **string** setOutputStatusTCP(**Socket** tcpConnect, **bool** status)

Set the output ON or OFF. String 'SS:x'.

Parameters: status-True means open the output, and false means close.

2.32 **string** setFrequencyTCP(**Socket** tcpConnect, **decimal** frequency)

Set the frequency. Return SF:x.

Parameters: frequency-The frequency value, and unit in GHz, 3 decimal places.

2.33 **string** setAttenuationTCP(**Socket** tcpConnect, **decimal** attenuation)

Set the attenuation value. Return SA:x.

Parameters: attenuation-The attenuation, unit in dB, 1 decimal place.

2.34 **string** setOutputPowerTCP(**Socket** tcpConnect, **decimal** outputPower)

Set the output Power. Return SOP:x.

Parameters: outputPower-The output power, unit in dBm, 1 decimal places.

Attention: If the ALC function closed, this command is invalid.

2.35 **string** setAlcStatusTCP(**Socket** tcpConnect, **bool** status)

Set the ALC function ON or OFF. Return SAS:x.

Parameters: status-True means open the output, and false means close.

2.36 **string** SetAGCTCP(**Socket** tcpConnect, **bool** status, **decimal** value)

Set AGC(Auto Gain Control) status and value of the EMC product.

Parameters : the state, must be 0 or 1; 0 means close the AGC function; 1 means open the AGC function. Value unit is dB, keep one decimal place, valid only if status = 1.

2.37 **bool** getOutputStatusTCP(**Socket** tcpConnect)

Query the output status.

Return false or true; false means closed, and true means open.

2.38 **decimal** getFrequencyTCP(**Socket** tcpConnect)

Query the frequency.

Return the setting frequency, unit in GHz, 3 decimal places.

2.39 **decimal** getAttenuationTCP(**Socket** tcpConnect)

Query the attenuation.

Return the current attenuation, unit in dB, 1 decimal places.

2.40 **decimal** getInputPowerTCP(**Socket** tcpConnect)

Query the input power.

Return the current input power, unit in dBm, 1 decimal places.

2.41 **decimal** getOutputPowerTCP(**Socket** tcpConnect)

Query the output power.

Return the current output power, unit in dBm, 1 decimal places.

2.42 **bool** getAlcStatusTCP(**Socket** tcpConnect, **out decimal** alcPower)

Query the ALC function status.

Return false or true; false means closed, and true means open.

Parameters: The alcPower will return a decimal value, unit in dBm, 1 decimal places.

2.43 decimal getOutputVswrTCP(Socket tcpConnect)

Query the output VSWR.

Return the current output VSWR, unit in :1, 2 decimal places.

2.44 decimal getTemperatureTCP(Socket tcpConnect)

Query the current temperature.

Return the current temperature, unit in degC, 1 decimal places.

2.45 decimal[] getFrequencyRangeTCP(Socket tcpConnect)

Query the product frequency range.

Return the minimum and maximum frequency range, unit in GHz, 3 decimal places.

2.46 decimal[] getAttenuationRangeTCP(Socket tcpConnect)

Query the product attenuation range.

Return the minimum and maximum attenuation range, unit in dB, 1 decimal places.

2.47 bool[] getAlarmTCP(Socket tcpConnect)

Query the alarm status.

Return the 6 alarms status. The order is:

- ★ Current (Exceed the maximum current or below the minimum current);
- ★ Current Imbalance;
- ★ Temperature (Exceed the maximum temperature);
- ★ Input Power (Exceed the maximum input power);
- ★ Output VSWR (Exceed the maximum output VSWR).
- ★ General (In addition to the above five kinds of alarm).

Each alarm status is defined by false or true, false means no alarm and true means alarm triggered.

2.48 bool GetAGCTCP (Socket tcpConnect)

Query AGC (Auto Gain Control) parameters of the EMC product.

Return AGC status, value and current actual gain of the EMC product.

3. Contact

QOTANA TECHNOLOGIES C.O.,LTD

<http://www.qotana.com/>

E-mail: sales@qotana.com