RWC2100F Remote Control Commands

Version 1.030

May 18, 2022

CONTENTS

Remote control Programming

Users can control the RWC2100F remotely through Ethernet or RS232C interface using a comprehensive set of commands. This section provides the necessary information to operate the RWC2100F under Ethernet and RS232C control.

1. Introduction

The RWC2100F provides a remote control method using string-based control commands. This method is very easy to understand and provides flexible backward compatibility with new versions of FW.

1.1. Command Structure

1.1.1. Command Format

The format of command is like as the below:

```
CMD_TYPE:CATEGORY:FUNCTION (param1) (param2) (param3) EOL
```

CMD_TYPE

CMD_TYPE consist of *configuration* and read *command*

Commands consist of *set commands* and *query commands* (usually simply called commands and queries). Set commands change instrument settings or perform a specific action. Queries cause the RWC2100F to return data and information about its status. Most commands have both a set form and query form. The set form of the command is started with "CONF" and the query form of the command is started with "READ".

One of examples of the set commands and query commands is shown below respectively:

```
CONF:TX_FM:FREQUENCY 88.7
READ:TX_FM:FREQUENCY?.
```

CATEGORY

There are five categories such as SYSTEM, FM_TX, AM_TX, FM_RX, and AUDIO.

FUNCTION

The part is a command corresponding to an action. For example, it is a command such as frequency or power.

If it needs parameters, a *space* is used to separate parameters from commands. In the case of commands with more than one parameter, a space is required between the parameters.

EOL (End Of Line)

All commands should be finished with EOL character such as '\n' or '\r'.

1.1.2. Configuration Command Parameter and Response

RWC2100F responds "ACK" or "NAK" for all configuration commands. If the command and parameters are valid, it responds ACK, and if invalid, it responds NAK.

There are three types of commands with respect to the number of parameters.

Single Parameter

```
CONF:RX:PATHLOSS 0.5\n
ACK

CONF:AUDIO:ENABLE YES\n
ACK
```

Double Parameters

```
CONF:FM_TX:POWER_DBM 1 -15\n
ACK

CONF:AM_TX:AUDIO_SOURCE 2 SING_A_SONG.WAV\n
ACK
```

Triple Parameters

```
CONF:FM_TX:AF_FREQ 2 5 88.7\n
ACK

CONF:FM_TX:AF_VARIANT 1 1 91.1\n
ACK
```

1.1.3. Read Command Parameter and Response

There are three types of read commands with respect to the number of parameters. The response can be text, integer, or double values. But all responses are ASCII formatted

No Parameter

```
*IDN?\n
RWC2100F Analog Radio Tester, Ver=1.000, SN=RWC21000000000
```

Single Parameter

```
READ:FM_TX:FREQ? 1\n
88.7

READ:FM_TX:RT_HEADLINE 1\n
RedwoodComm
```

Double Parameters

```
READ:FM_TX:ERT_TAG_TYPE? 1 0\n
ITEM_TITLE

READ:FM_TX:AF_VARIANT 2 3\n
91.1
```

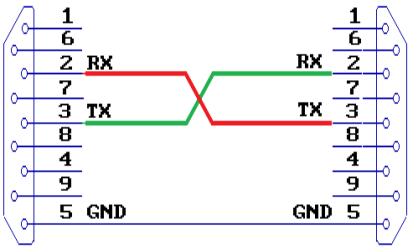
2. Interface

The RWC2100F supports RS232C and Ethernet interfaces on the rear panel for remote control operation. Ethernet is used for high speed and flexible interfaces. But socket programming is required to use Ethernet. RS232C is a slow serial interface, but it does not require any special equipment and is easy to use.

2.1. RS232

2.1.1. RS232 Port Connection of RWC2100F

RWX2100F has a DSUB9 connector for RS232 interface.
Users can use a DSUB-9 cross cable defined as the figure 1.



RWC2100F Remote PC

Figure 1. Cable connection of DSUB9 for RS232

2.1.2. Parameters of RS232

All parameters of RS232 of the RWC2100F are fixed as follows.

Parameter	Range	Description
BAUD RATE	115200	fixed
DATA BITS	8-bit	Length of Data bit
PARITY	Off	Error check bit
STOP BIT	1-bit	Stop bit

2.2. Ethernet Interface

2.2.1 The IP setup of the remote PC

RWC2100F has a RJ45 connector for LAN connection. Connect RWC2100F to your network hub with RJ45 LAN cable. If the PC and RWC2100F are connected directly, a cross cable must be used. For direct connection, set up the IP address static as follows.

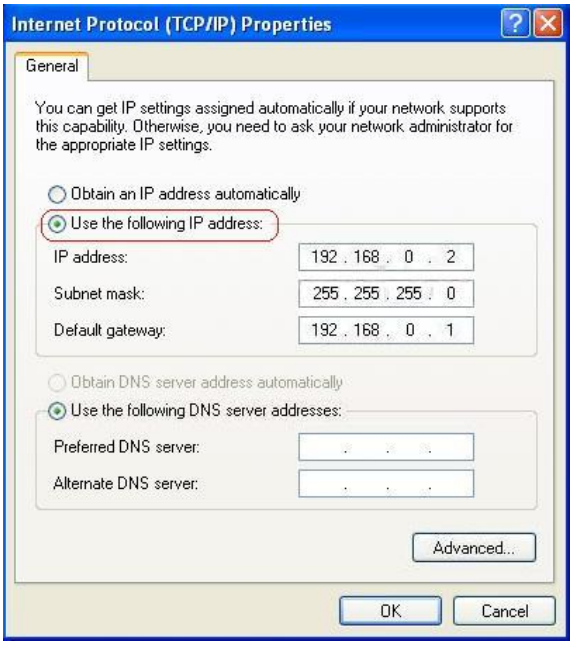


Figure 2. IP Configuration

2.2.2 Definition network socket in user application program.

RWC2100F uses the UDP protocol type. If you control with your own PC application or terminal program, you have to define your network protocol as a UDP.

```
Socket Rwc2100fSock;
String IP_ADDR_RWC2100F = “192.168.0.100”;
LocalAddress = IPAddress.Any;
LocalIpEndPoint = new IPEndPoint(LocalAddress, Convert.ToInt32(IP_ADDR_RWC2100F));
Rwc2100fSock = new Socket(LocalAddress.AddressFamily, SocketType.Dgram, ProtocolType.Udp);
```

Example code with C#

2.2.3 The IP setup of the RWC2100F

It supports only IP4 format. It supports both dynamic and static IP address setup. Whenever you toggle the button on the rear panel of RWC2100F, the address type will be toggled between dynamic and static IP address. In dynamic mode, the RWC2100F will get the IP address from your DHCP server. In static mode, it will keep the same IP address that was assigned by DHCP server when in dynamic mode.

2.2.4 Control example using the scripter function of the RWC2100F application program

Users can control the RWC2100F remotely using the scripter of RWC2100F PC application program. Controlling with LAN connection rather than RS232 is strongly recommended for more stable operation.

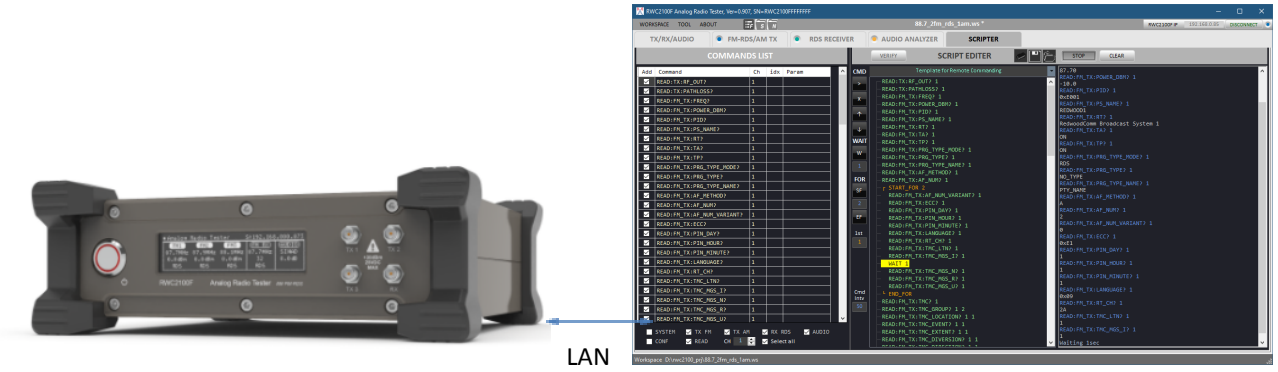


Figure 3. Controlling with the scripter of RWC2100F PC application

3. Command Tables

3.1. SYSTEM

Command	(param) Range
*IDN?	
READ:SYSTEM:SERIAL_NUM?	
READ:SYSTEM:IP_TYPE?	
CONF:SYSTEM:IP_TYPE (param)	DYNAMIC, STATIC
READ:SYSTEM:IP_ADDR?	
CONF:SYSTEM:IP_ADDR (param)	IP4 format 255.255.255.255
READ:SYSTEM:SW_VERSION?	

3.2. RF_COMMON

Command	(param1) Range	(param2) Range
READ:TX:AM_FM_SEL? (param1)	Channel # 1 ~ 3	
CONF:TX:AM_FM_SEL (param1) (param2)	Channel # 1 ~ 3	AM, FM
READ:TX:RF_OUT? (param1)	Channel # 1 ~ 3	
CONF:TX:RF_OUT (param1) (param2)	Channel # 1 ~ 3	ON, OFF
READ:TX:PATHLOSS? (param1)	Channel # 1 ~ 3	
CONF:TX:PATHLOSS (param1) (param2)	Channel # 1 ~ 3	0.0 ~ 60.0
READ:RX:PATHLOSS?		
CONF:RX:PATHLOSS (param1)	0.0 ~ 60.0	

3.3. FM_TX

Command	(param1) Range	(param2) Range	(param3) Range
READ:FM_TX:FREQ? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:FREQ (param1) (param2)	Channel # 1 ~ 3	76.0 ~ 107.9	
READ:FM_TX:POWER_UNIT? (param1)	Channel # 1 ~ 3		
READ:FM_TX:POWER_DBM? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:POWER_DBM (param1) (param2)	Channel # 1 ~ 3	-90 ~ 0	
READ:FM_TX:POWER_DBUV? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:POWER_DBUV (param1) (param2)	Channel # 1 ~ 3	17 ~ 107	
READ:FM_TX:PID? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PID (param1) (param2)	Channel # 1 ~ 3	0x0001~ 0xFFFF	
READ:FM_TX:PS_NAME? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PS_NAME (param1) (param2)	Channel # 1 ~ 3	String(8B)	
READ:FM_TX:RT? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:RT (param1) (param2)	Channel # 1 ~ 3	OFF, RT, RT+	
READ:FM_TX:TA? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TA (param1) (param2)	Channel # 1 ~ 3	ON, OFF	
READ:FM_TX:TP? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TP (param1) (param2)	Channel # 1 ~ 3	ON, OFF	
READ:FM_TX:PRG_TYPE_MODE?	Channel # 1 ~ 3		
CONF:FM_TX:PRG_TYPE_MODE (param1) (param2)	Channel # 1 ~ 3	RDS, RBDS	
READ:FM_TX:PRG_TYPE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PRG_TYPE (param1) (param2)	Channel # 1 ~ 3	NO_TYPE, NEWS, …	
READ:FM_TX:PRG_TYPE_NAME? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PRG_TYPE_NAME (param1)	Channel # 1 ~ 3	String(8B)	
READ:FM_TX:AF_METHOD? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:AF_METHOD (param1) (param2)	Channel # 1 ~ 3	A, B	

READ:FM_TX:AF_NUM? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:AF_NUM (param1) (param2)	Channel # 1 ~ 3	0 ~ 10	
READ:FM_TX:AF_NUM_VARIANT? (param1) (param2)	Channel # 1 ~ 3		
CONF:FM_TX:AF_NUM_VARIANT(param1) (param2) (param3)	Channel # 1~3	0 ~ 5	
READ:FM_TX:AF_FREQ? (param1) (param2)	Channel # 1 ~ 3		
CONF:FM_TX:AF_FREQ (param1) (param2) (param3)	Channel # 1 ~ 3	1 ~ 10	87.5 ~ 107.9
READ:FM_TX:ECC? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:ECC (param1) (param2)	Channel # 1 ~ 3	HEX string(1B)	
READ:FM_TX:PIN_DAY? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PIN_DAY (param1) (param2)	Channel # 1 ~ 3	1 ~ 31	
READ:FM_TX:PIN_HOUR? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PIN_HOUR (param1) (param2)	Channel # 1 ~ 3	0 ~ 23	
READ:FM_TX:PIN_MINUTE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:PIN_MINUTE (param1) (param2)	Channel # 1 ~ 3	0 ~ 59	
READ:FM_TX:LANGUAGE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:LANGUAGE (param1) (param2)	Channel # 1 ~ 3	0 ~ 0xFF	
READ:FM_TX:RT_CH? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:RT_CH (param1) (param2)	Channel # 1 ~ 3	2A, 2B	
READ:FM_TX:TMC_LTN? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_LTN (param1) (param2)	Channel # 1 ~ 3	0 ~ 63	
READ:FM_TX:TMC_MSG_AFI? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_MSG_AFI (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC_MSG_MODE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_MSG_MODE (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC_MSG_I? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_MSG_I (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC_MSG_N? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_MSG_N (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC_MSG_R? (param1)	Channel # 1 ~ 3		

CONF:FM_TX:TMC_MSG_R (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC_MSG_U? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC_MSG_U (param1) (param2)	Channel # 1 ~ 3	0, 1	
READ:FM_TX:TMC? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TMC (param1) (param2)	Channel # 1 ~ 3	ON, OFF	
READ:FM_TX:TMC_LOCATION? (param1) (param2)	Channel # 1 ~ 3	1,2,3,4,5	
CONF:FM_TX:TMC_LOCATION (param1) (param2) (param3)	Channel # 1 ~ 3	1,2,3,4,5	0 ~ 65535
READ:FM_TX:TMC_EVENT? (param1) (param2)	Channel # 1 ~ 3	1,2,3,4,5	
CONF:FM_TX:TMC_EVENT (param1) (param2) (param3)	Channel # 1 ~ 3	1,2,3,4,5	0 ~ 2047
READ:FM_TX:TMC_EXTENT? (param1) (param2)	Channel # 1 ~ 3	1,2,3,4,5	
CONF:FM_TX:TMC_EXTENT (param1) (param2) (param3)	Channel # 1 ~ 3	1,2,3,4,5	0 ~ 7
READ:FM_TX:TMC_DIVERSION? (param1) (param2)	Channel # 1 ~ 3	1,2,3,4,5	
CONF:FM_TX:TMC_DIVERSION (param1) (param2) (param3)	Channel # 1 ~ 3	1,2,3,4,5	0, 1
READ:FM_TX:TMC_DIRECTION? (param1) (param2)	Channel # 1 ~ 3	1,2,3,4,5	
CONF:FM_TX:TMC_DIRECTION (param1) (param2) (param3)	Channel # 1 ~ 3	1,2,3,4,5	0, 1
READ:FM_TX:TIME? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:TIME (param1) (param2)	Channel # 1 ~ 3	ON, OFF	
READ:FM_TX:RT_TAG_TYPE? (param1) (param2)	Channel # 1 ~ 3	0, 1	
CONF:FM_TX:RT_TAG_TYPE (param1) (param2) (param3)	Channel # 1 ~ 3	0, 1	DUMMY, ITEM_TITLE, …
READ:FM_TX:RT_TAG_START? (param1) (param2)	Channel # 1 ~ 3	0, 1	
CONF:FM_TX:RT_TAG_START (param1) (param2) (param3)	Channel # 1 ~ 3	0, 1	0 ~ 127
READ:FM_TX:RT_TAG_LENGTH? (param1) (param2)	Channel # 1 ~ 3	0, 1	
CONF:FM_TX:RT_TAG_LENGTH (param1) (param2) (param3)	Channel # 1 ~ 3	0, 1	0 ~ 127
READ:FM_TX:RT_MODE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:RT_MODE (param1) (param2)	Channel # 1 ~ 3	OFF, RT, RT+	
READ:FM_TX:EON? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:EON (param1) (param2)	Channel # 1 ~ 3	ON, OFF	
READ:FM_TX:EON_CH? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:EON_CH (param1) (param2)	Channel # 1 ~ 3	FM_RDS_0, FM_RDS_1, FM_RDS_2	
READ:FM_TX:EON_PID? (param1)	Channel # 1 ~ 3		

CONF:FM_TX:EON_PID (param1) (param2)	Channel # 1 ~ 3	0x0001 ~ 0xFFFF
READ:FM_TX:EON_PS_NAME? (param1)	Channel # 1 ~ 3	
~~CONF:FM_TX:EON_PS_NAME (param1) (param2)~~	~~Channel # 1 ~ 3~~	~~string~~
READ:FM_TX:EON_AF? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:EON_AF (param1) (param2)	Channel # 1 ~ 3	87.6 ~ 107.9
READ:FM_TX:EON_SWITCH? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:EON_SWITCH (param1) (param2)	Channel # 1 ~ 3	ON, OFF
READ:FM_TX:AUDIO_VOLUME? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AUDIO_VOLUME (param1) (param2)	Channel # 1 ~ 3	0 ~ 100
READ:FM_TX:FM_DEVIATION? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:FM_DEVIATION (param1) (param2)	Channel # 1 ~ 3	0 ~ 75
READ:FM_TX:PILOT_LEVEL_UNIT? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:PILOT_LEVEL_UNIT (param1) (param2)	Channel # 1 ~ 3	PERCENT, KHZ
READ:FM_TX:PILOT_LEVEL_KHZ? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:PILOT_LEVEL_KHZ (param1) (param2)	Channel # 1 ~ 3	0 ~ 15
READ:FM_TX:PILOT_LEVEL_PERCENT? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:PILOT_LEVEL_PERCENT (param1) (param2)	Channel # 1 ~ 3	0 ~ 15
READ:FM_TX:AUDIO_FREQ? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AUDIO_FREQ (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:FM_TX:AUDIO_SOURCE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AUDIO_SOURCE (param1) (param2)	Channel # 1 ~ 3	FILE, TONE_MONO, TONE_STEREO,SWEEP
READ:FM_TX:STEREO_MODE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:STEREO_MODE (param1) (param2)	Channel # 1 ~ 3	LEFT_AND_RIGHT, LEFT_ONLY,RIGHT_ONLY
READ:FM_TX:AUDIO_FREQ_R? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AUDIO_FREQ_R (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:FM_TX:AUDIO_FREQ_L? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AUDIO_FREQ_L (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:FM_TX:SWEEP_START? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:SWEEP_START (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:FM_TX:SWEEP_STOP? (param1)	Channel # 1 ~ 3	

CONF:FM_TX:SWEEP_STOP (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:FM_TX:SWEEP_TIME? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:SWEEP_TIME (param1) (param2)	Channel # 1 ~ 3	20 ~ 10000
READ:FM_TX:RDS_MODE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RDS_MODE (param1) (param2)	Channel # 1 ~ 3	ON, OFF
READ:FM_TX:PRE_EMPHASIS? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:PRE_EMPHASIS (param1) (param2)	Channel # 1 ~ 3	OFF, 50us, 75us
READ:FM_TX:WAVE_FILE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:WAVE_FILE (param1) (param2)	Channel # 1 ~ 3	String
READ:FM_TX:WAVE_PROG? (param1)	Channel # 1 ~ 3	
READ:FM_TX:YEAR? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:YEAR (param1) (param2)	Channel # 1 ~ 3	1900 ~ 2200
READ:FM_TX:MONTH? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:MONTH (param1) (param2)	Channel # 1 ~ 3	1 ~ 12
READ:FM_TX:DAY? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:DAY (param1) (param2)	Channel # 1 ~ 3	1 ~ 31
READ:FM_TX:HOUR? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:HOUR (param1) (param2)	Channel # 1 ~ 3	0 ~ 23
READ:FM_TX:MINUTE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:MINUTE (param1) (param2)	Channel # 1 ~ 3	0 ~ 59
READ:FM_TX:MODULATION? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:MODULATION (param1) (param2)	Channel # 1 ~ 3	ON, OFF
READ:FM_TX:LTO? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:LTO (param1) (param2)	Channel # 1 ~ 3	-24 ~ 24
READ:FM_TX:FM_DEVIATION? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:FM_DEVIATION (param1) (param2)	Channel # 1 ~ 3	0 ~ 75
READ:FM_TX:RDS_FILE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RDS_FILE (param1) (param2)	Channel # 1 ~ 3	String
READ:FM_TX:RDS_PROG? (param1)	Channel # 1 ~ 3	
READ:FM_TX:PS_NAME_HEX? (param1)	Channel # 1 ~ 3	

CONF:FM_TX:PS_NAME_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String(8B)
READ:FM_TX:RT_HEX? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RT_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String
READ:FM_TX:RT_HEADLINE_MODE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RT_HEADLINE_MODE (param1) (param2)	Channel # 1 ~ 3	ON, OFF
READ:FM_TX:RT_HEADLINE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RT_HEADLINE (param1) (param2)	Channel # 1 ~ 3	String
READ:FM_TX:ENCODING_FLAG? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:ENCODING_FLAG (param1) (param2)	Channel # 1 ~ 3	UTF_8, UCS_2
READ:FM_TX:RT_HEADLINE_HEX? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:RT_HEADLINE_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String
READ:FM_TX:EON_PS_NAME_HEX? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:EON_PS_NAME_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String
READ:FM_TX:TEXT_DIRECTION? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:TEXT_DIRECTION (param1) (param2)	Channel # 1 ~ 3	LTR, RTL
READ:FM_TX:AF_VARIANT? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AF_VARIANT (param1) (param2)	Channel # 1 ~ 3	0 ~ 5
READ:FM_TX:M_S? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:M_S (param1) (param2)	Channel # 1 ~ 3	MUSIC, SPEECH
READ:FM_TX:AF? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:AF (param1) (param2)	Channel # 1 ~ 3	0 ~ 10
READ:FM_TX:POWER_DBUV? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:POWER_DBUV (param1) (param2)	Channel # 1 ~ 3	17~97
READ:FM_TX:ERT_MODE? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:ERT_MODE (param1) (param2)	Channel # 1 ~ 3	OFF, eRT, eRT+
READ:FM_TX:ERT? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:ERT (param1) (param2)	Channel # 1 ~ 3	String
READ:FM_TX:ERT_HEX? (param1)	Channel # 1 ~ 3	
CONF:FM_TX:ERT_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String
READ:FM_TX:ERT_HEADLINE? (param1)	Channel # 1 ~ 3	

CONF:FM_TX:ERT_HEADLINE_HEX (param1) (param2)	Channel # 1 ~ 3	HEX String	
READ:FM_TX:ERT_TAG_TYPE? (param1) (param2)	Channel # 1 ~ 3	0,1	
CONF:FM_TX:ERT_TAG_TYPE (param1) (param2) (param3)	Channel # 1 ~ 3	0,1	DUMMY, ITEM_TITLE, …
READ:FM_TX:ERT_TAG_START? (param1) (param2)	Channel # 1 ~ 3	0, 1	
CONF:FM_TX:ERT_TAG_START (param1) (param2) (param3)	Channel # 1 ~ 3	0, 1	0 ~ 63
READ:FM_TX:ERT_TAG_LENGTH? (param1) (param2)	Channel # 1 ~ 3	0, 1	
CONF:FM_TX:ERT_TAG_LENGTH (param1) (param2) (param3)	Channel # 1 ~ 3	0, 1	0 ~ 63
READ:FM_TX:ERT_HEADLINE_MODE? (param1)	Channel # 1 ~ 3		
CONF:FM_TX:ERT_HEADLINE_MODE (param1) (param2)	Channel # 1 ~ 3	ON, OFF	

3.4. AM_TX

Command	(param1) Range	(param2) Range
READ:AM_TX:FREQ? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:FREQ (param1) (param2)	Channel # 1 ~ 3	500~1710kHz
READ:AM_TX:INDEX? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:INDEX (param1) (param2)	Channel # 1 ~ 3	0 ~ 100
READ:AM_TX:AUDIO_FREQ? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:AUDIO_FREQ (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:AM_TX:AUDIO_SOURCE? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:AUDIO_SOURCE (param1) (param2)	Channel # 1 ~ 3	FIEL, TONE, SWEEP
READ:AM_TX:POWER_UNIT? (param1)	Channel # 1 ~ 3	
READ:AM_TX:POWER_DBM? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:POWER_DBM (param1) (param2)	Channel # 1 ~ 3	-90 ~ -10
READ:AM_TX:POWER_DBUV? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:POWER_DBUV (param1) (param2)	Channel # 1 ~ 3	17~ 97
READ:AM_TX:SWEEP_START? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:SWEEP_START (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:AM_TX:SWEEP_STOP? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:SWEEP_STOP (param1) (param2)	Channel # 1 ~ 3	0 ~ 20
READ:AM_TX:SWEEP_TIME? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:SWEEP_TIME (param1) (param2)	Channel # 1 ~ 3	20 ~ 10000
READ:AM_TX:WAVE_FILE? (param1)	Channel # 1 ~ 3	
CONF:AM_TX:WAVE_FILE (param1) (param2)	Channel # 1 ~ 3	String
READ:AM_TX:WAVE_PROG? (param1)	Channel # 1 ~ 3	

3.5. FM_RX

Command	(param) Range
READ:FM_RX:FREQ?	
CONF:FM_RX:FREQ (param)	76.0 ~ 107.9
READ:FM_RX:ENABLE?	
CONF:FM_RX:ENABLE (param)	NO, YES
READ:FM_RX:RDS_STATUS?	
READ:FM_RX:RSSI?	
READ:FM_RX:SNR?	
READ:FM_RX:VOLUME?	
CONF:FM_RX:VOLUME (param)	0 ~ 10

3.6. AUDIO

Command	(param) Range
READ:AUDIO:ENABLE?	
CONF:AUDIO:ENABLE (param)	NO, YES
READ:AUDIO:TRIGGER?	
CONF:AUDIO:TRIGGER (param)	OFF, LEFT, RIGHT
READ:AUDIO:SINAD_L?	
READ:AUDIO:SINAD_R?	
READ:AUDIO:THDN_L?	
READ:AUDIO:THDN_R?	
READ:AUDIO:SNR_L?	
READ:AUDIO:SNR_R?	
READ:AUDIO:FREQ_R?	
READ:AUDIO:FREQ_L?	
READ:AUDIO:THD_L?	
READ:AUDIO:THD_R?	
READ:AUDIO:REF_FREQ_L?	
CONF:AUDIO:REF_FREQ_L (param)	0.4 ~ 4
READ:AUDIO:REF_FREQ_R?	
CONF:AUDIO:REF_FREQ_R (param)	0.4 ~ 4
READ:AUDIO:AVG_NUM?	
CONF:AUDIO:AVG_NUM (param)	1 ~ 50

Headquarters
#14008, OfficeSection Bldg, SK M-city, 195 Baengma-ro, Ilsandong-gu, Goyang-si, Gyeonggi-do, Korea
+82-70-7727-7011

Canada Branch
Suite 201, 132 15th Street West, North Vancouver, BC V7M 1R5, Canada
+1-640-770-2688

E-mail sales@RedwoodComm.com
Webpage http://RedwoodComm.com/