RFR100 433MHz TCPIP Reader

User Manual

Revision 02

User Manual

Revision	Date	Description
00	2009.1.15	Preliminary draft
01	2009.3.19	
02	2009.8.17	

Copyright

FCC Statement

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:
(1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation of the device.

FCC Statement
This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against h armful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications
Any changes or modification not expressly approved by the party responsible could void the user's authority to operate the device.

Introduction

RFR100 433MHz Reader by Champtek is a RFID monitoring system designed to receive all tags data through transmission frequency of 433.92 MHz. It was designed for easy setup, configuration, and usage.
RFR100 provides Ethernet interface to communicate with backend host. Host can easily search all available Champtek readers and setup related parameters. Besides, It made a significant achievement of the system throughput (Up to 160 tag-reports per second) to prevent from the data loss for the high-density implement.
RFR100 built in smart channel clear detection function indicates strength of channel interference that reduce the time and cost to setup the system. Reader feedbacks all the RSSI value of the packet received from the tags that is useful for the application of the real time positioning system. RFR100 uses cyclic redundancy check to ensure the data integrity and provide API to reduce the time to design the application software.

Features

- User configurable identification range (1m to 90m)
- Smart channel clear detection function
- Complete command set and API support
- Advanced two-layer anti-collision technique, identifying up to 160 tags per second

Specification

Reader Communication Protocol	
Ethernet	10/100M Fast Ethernet
Protocols	TCP/IP, UDP, ICMP, ARP, DHCP Client, Telnet
Operation frequency	433.92 MHz
Signal Strength	Default -30dbm
Range Control	Adjustable RSSI range (0 to -99dbm)
Modulation	GFSK

Read Range	
Omni Angle Antenna	90m

Physical Characteristics	
Length	4.89”
Width	3.09”
Height	0. 98”
Weight	120g
Case	Aluminum + ABS

Environmental	
Operation Temperature	-30℃ to 65℃
Storage Temperature	-40℃ to 80℃

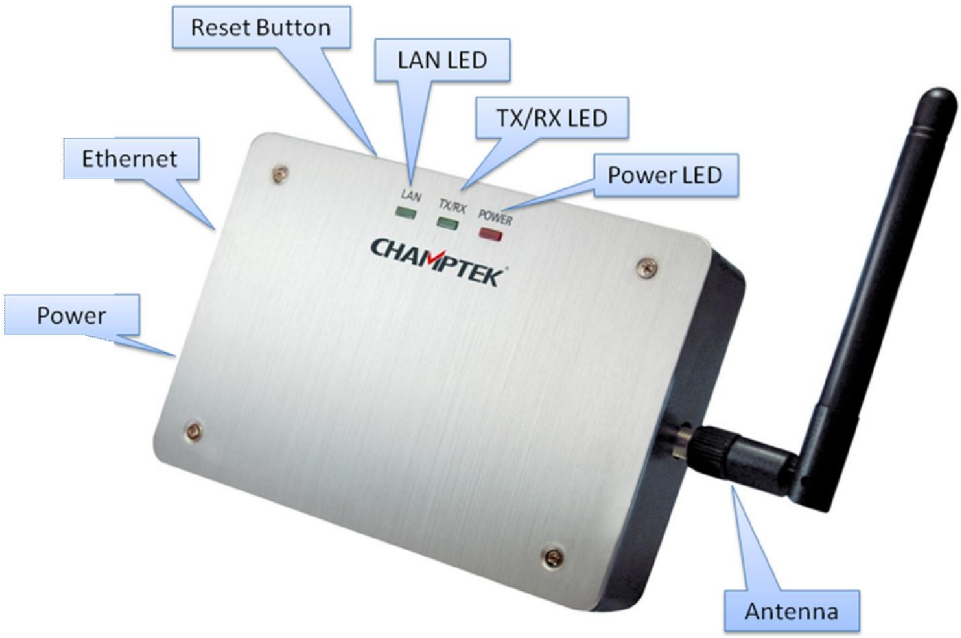
LED Indicators	
Power LED	Power on indicator
TX/RX LED	TX/RX indicator
Link LED	LAN indicator

Electrical	
Power	5Vdc, 1.2A
Power Consumption	1.2W

Connectors	
Ethernet	RJ-45 female to host
Antenna	SMA

Mechanicals

TOP View

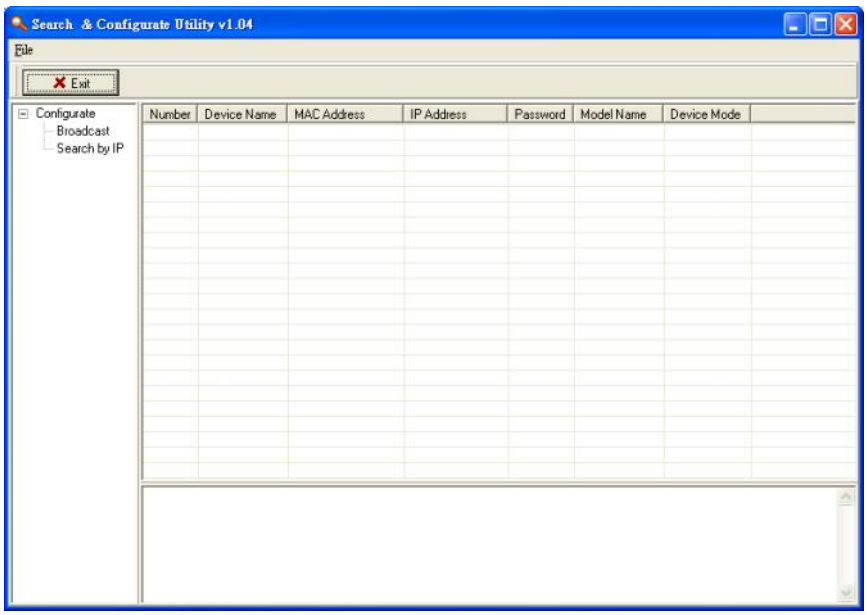


Power LED – On when reader is powered.
TX/RX LED – On while reader is transmitting or receiving a packet.
LAN LED – Blink when there are network activities.
Ethernet – RJ-45 Ethernet jack.
Antenna – SMA connector
Power – 5Vdc
Reset Button – Hardware Reset the Reader

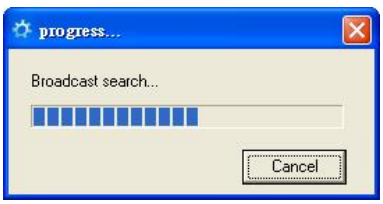
Network Configuration

This section will introduce how to configure the network connection of RFR100 by utility TCPIP.exe

- RFR100 default IP: 192.168.1.127
- Run Network Utility “TCPIP.exe”
- Homepage:

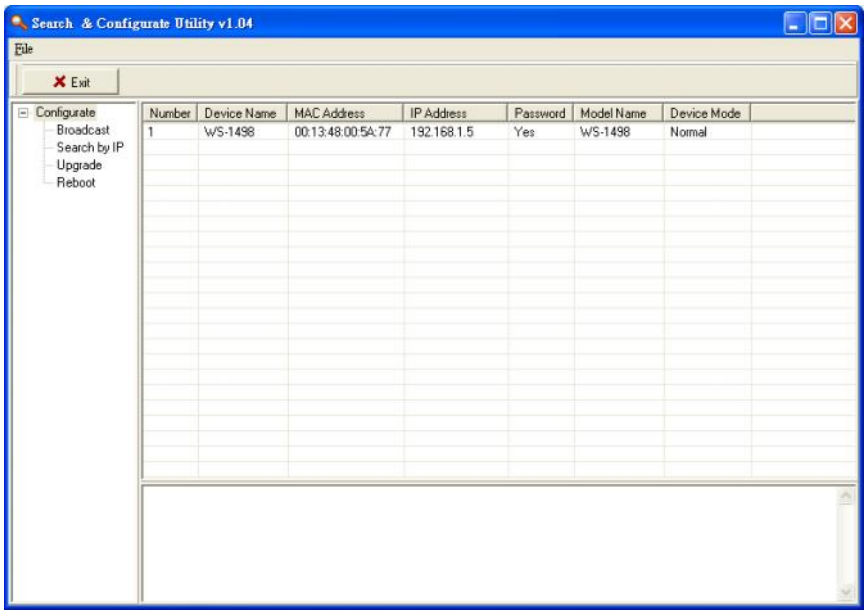


- Search devices
  1. Broadcast Search
     If don't know the IP of the device, can utilize broadcast to search all the devices in the LAN to configure the parameters.

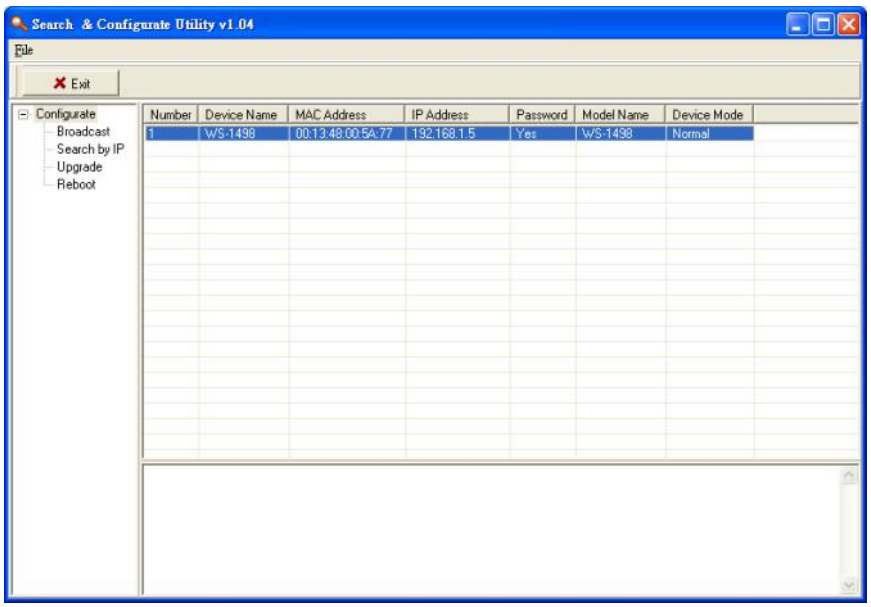


  2. Search by IP
     If know the IP of the device not in the same LAN, can search by IP to set the parameters.

- Search result



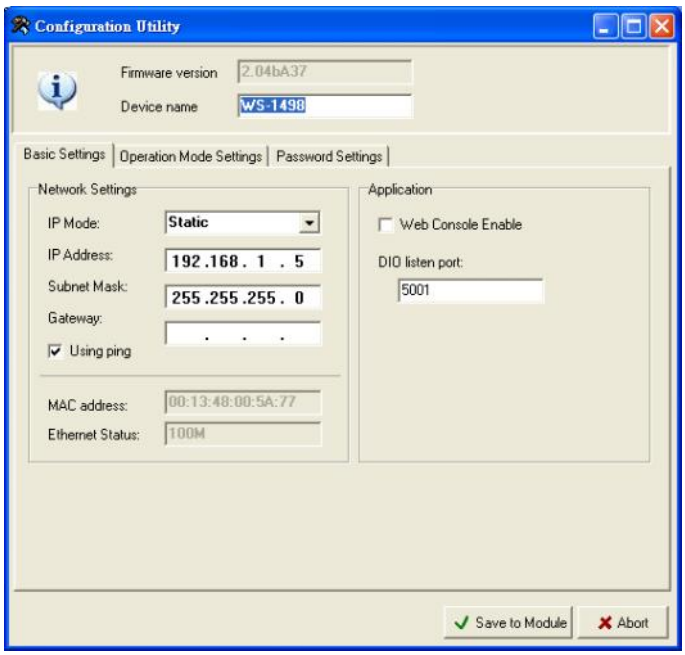
- Select anyone of the device and will pop up a dialog to check password.



- Check Password:
  Default Password: 00000000

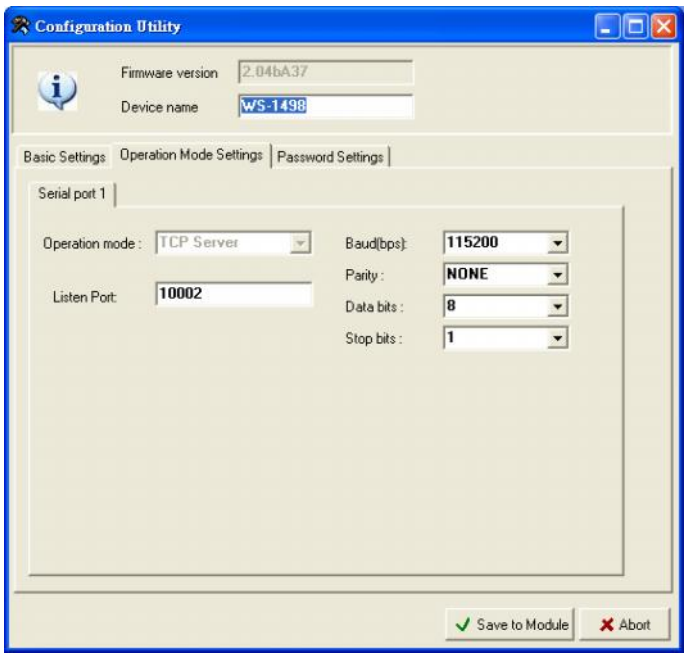
- Connect to the target device to configure the parameters.

- Configuration page 1
  1. Device Name
  2. Network Setting

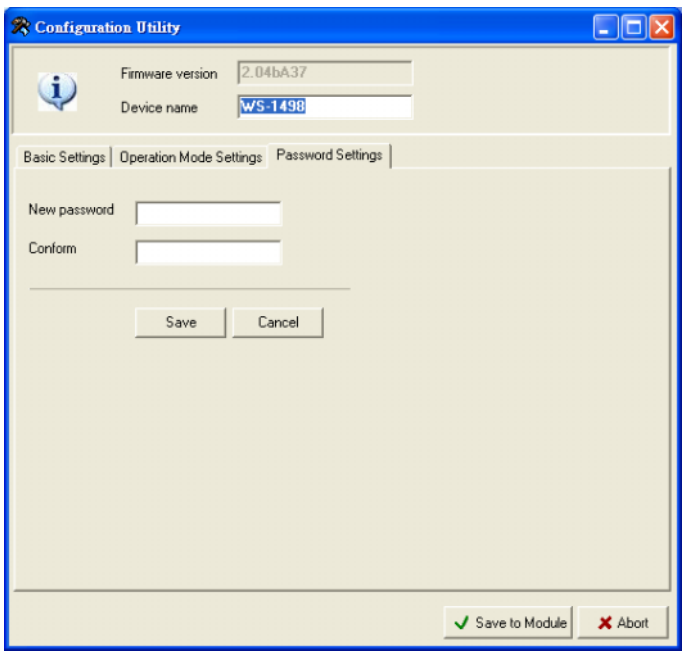


(Notice: Do not change the value the DIO listen port 5001.)

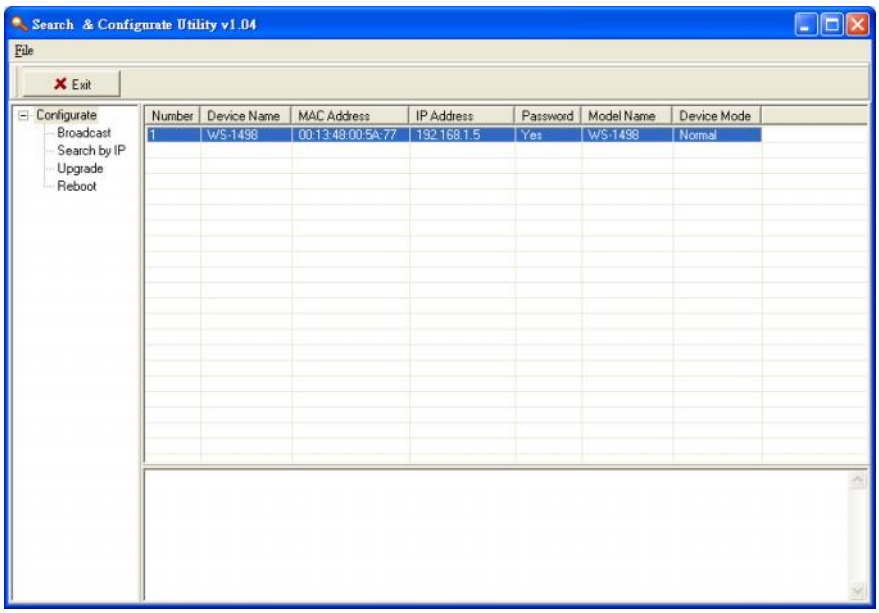
- Configuration page 2
  1. Reader Operation mode and listen port
  2. Serial Setting (Do not change the default setting)



- Configuration page 2
  1. Change password (Default password is 00000000)
  2. Change Device name



- Utility Upgrade (if needed)



Packet Format

Head	Reader ID	Group	Tag ID	Status	RSSI	Data	End Code
$	000000~999999	00~63	000000~999999	*	00~99	**	0x0d 0x0a

* “Status” indicates the tag’s condition such as moving, still, etc.
** The content of “Data” depends on different groups of tag.

Implementation Notice

- Keep away from potential RF radiators such as electronic / radio equipments for minimizing interference, i.e. engine, monitor etc, utilize the channel clear detection feature to pre-scan the interference and then choose the best place to setup the reader. And readers should be mounted at reasonable height for better performance.
- RFR100 supports wide range of antennas for different coverage areas, high quality SMA coaxial cable may be use to separate the antennas from Reader.

- Recommends CAT-5/5E or CAT-6 high quality Ethernet wire cable for RFR100.
- RFR100 is typical indoor used, for outdoor environment the Reader can be protected in a NEMA enclosure. The operation performance will be reduced the negatively impacted by nature weather conditions such as moisture, excessive temperature.

APIs

Champtek also provides efficient C ++ APIs to reduce the time of software development. This section describes how to use the APIs to setup and control RFR100.

Network Connection API

Prototype	DWORD con_connectdvc(char* ip, unsigned short port)
Description	Connect the remote reader with assigned IP and Port
Parameter	ip[input]: The address of the IP port[input]: The communication port of the reader
Return value	Connection ID (Program automatically assigns this connection ID to the reader) Null – Failed

Prototype	bool con_selectdvc(DWORD handle)
Description	Select a specific reader before send a command
Parameter	Handle[input]: Connection ID
Return value	True False – Failed

Prototype	bool con_reconnect()
Description	Re-connect the selected device (ref. bool con_selectdvc)
Parameter	None
Return value	True False

Command API

Prototype	void param_settimeout(int timevalue)
Description	Setting the threshold of command timeout
Parameter	Timevalue[in]: Command timeout
Return value	None

Prototype	int prtcl_startdvc()
Description	Start the reader to transfer the tag data back to the host through Ethernet
Parameter	None
Return Value	0 (Successful) – Successful operation 1 (Time out) – No response 2(Error) – Failed operation

Prototype	int prtcl_stopdvc()
Description	Stop the reader to transfer the tag data back to the host through Ethernet
Parameter	None
Return value	0: Successful - operation successful 1: Timeout – No response 2: Error - Reader received command but failed.

Prototype	int prtcl_resetmcu()
Description	Software reset the reader
Parameter	None
Return Value	0 (Successful) – Successful operation 1 (Time out) – No response 2 (Error) – Failed operation
Comment	After reset the reader, the Ethernet connection will be disconnected. Need to re-connect the reader for further operation.

Prototype	int prtcl_resettcp()
Description	Software reset the Ethernet module
Parameter	None
Return Value	0 (Successful) – Successful operation 1 (Time out) – No response 2 (Error) – Failed operation
Comment	After reset the Ethernet module, the Ethernet connection will be disconnected. Need to re-connect the reader for further operation.

Prototype	int prtcl_getidnumber(DWORD idnum)
Description	Inquire the ID of the reader
Parameter	Idnum[out]: 000000~999999(decimal)
Return value	0 (Successful) – Successful operation 1 (Time out) – No response 2 (Error) – Failed operation

Prototype	int prtcl_setidnumber(DWORD idnum)
Description	Set the ID of the reader
Parameter	Idnum[in]: 000000~999999(decimal)
Return value	0: Successful – Successful operation 1: Timeout – No response 2: Error – Failed operation 3: Illegal parameter - The parameter is illegal

Prototype	int prtcl_setRSSIrange(unsigned short range)
Description	Set the receiving RSSI range of the reader
Parameter	range[in]: 00~99(decimal) Ex. If want to reader only send back the packet above -80dbm. Set range[in] to 80
Return value	0: Successful – Successful operation 1: Timeout – No response 2: Error – Failed operation 3: Illegal parameter - The parameter is illegal

Prototype	int prtcl_getRSSIrange(unsigned short range)
Description	Inquire the RSSI range setting of the read er
Parameter	range[out]: The receiving RSSI range of the reader
Return value	0 (Successful) – Successful operation 1 (Time out) – No response 2 (Error) – Failed operation

Prototype	int prtcl_getfirmwareversion(char* version)
Description	Inquire the firmware version of the reader
Parameter	version[out]: [Data Format: XX.XX]
Return value	0 (Successful) – Successful operation 1 (Time out) – No response 2 (Error) – Failed operation

Data Handling API

Prototype	int prtcl_rssi_start()
Description	Get the channel RSSI information into buffer
Parameter	None
Return value	0: Successful - operation successful 1: Timeout – No response 2: Error - Reader received command but failed.

Prototype	int prtcl_rssi_stop()
Description	Stop receiving the channel RSSI information
Parameter	None
Return value	0: Successful - operation successful 1: Timeout – No response 2: Error - Reader received command but failed.

Prototype	DWORD data_pop_msg(char* lpbuf, DWORD nSize)
Description	Move a tag packet from the buffe r to specific memory
Parameter	lpbuf: Tag packet nSize[in]: memory size
Return value	Tag Packet real size 0: No data in the buffer

Prototype	DWORD data_pop_rssi()
Description	Get the RSSI message from the buffer
Parameter	None
Return value	RSSI message -1: No data in the buffer

Prototype	DWORD data_group (const char* cmsg, char* pdata)
Description	Get the group code from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Group code
Return value	The length of group code 0: Failed

Prototype	DWORD data_tagid (const char* cmsg, char* pdata)
Description	Get the tag id from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Tag ID
Return value	The length of tag ID 0: Failed

Prototype	DWORD data_tagstatus (const char* cmsg, char* pdata)
Description	Get the status from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Tag status
Return value	The length of tag status 0: Failed

Prototype	DWORD data_rssi (const char* cmsg,char* pdata)
Description	Get the RSSI from a specific packet
Parameter	cmsg[in]: Tag packet pdata: RSSI value
Return value	The length of RSSI value 0: Failed

Prototype	DWORD data_data (const char* cmsg,char* pdata)
Description	Get the data from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Tag data
Return value	The length of tag data 0: Failed

Prototype	DWORD data_readerid (const char* cmsg,char* pdata)
Description	Get the reader id id from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Reader ID
Return value	The length of reader ID 0: Failed

Prototype	DWORD data_routerid (const char* cmsg,char* pdata)
Description	Get the router id from a specific packet
Parameter	cmsg[in]: Tag packet pdata: Router ID
Return value	The length of router ID 0: Failed