RS485 / UART to 433MHz RF Convertor

1 GENERAL DESCRIPTION

The CT-UR-433M is a RS485/UART to 433MHz RF Convertor. It can convert RS485 bus or UART data to 433MHZ RF signal for transmission. At the same time it can listen to and receive the 433MHz RF signal and convert it to RS485 or UART data.

It consists of a high-performance MCU processor, an OOK 433M RF receiver module and an RF transmitter module.

It is fully compatible (can receive or simulate sending) existing chip encoding format data such as PT2262/EV1527. What's more, it can also support Half-duplex wireless data transmission of 433MHz.

The encoding format of the air packet can be customized by command.

By integrating the intelligent recognition algorithm, it can obtain detailed format information of the received data (such as pulse width, sync header, 1 code, 0 code format).

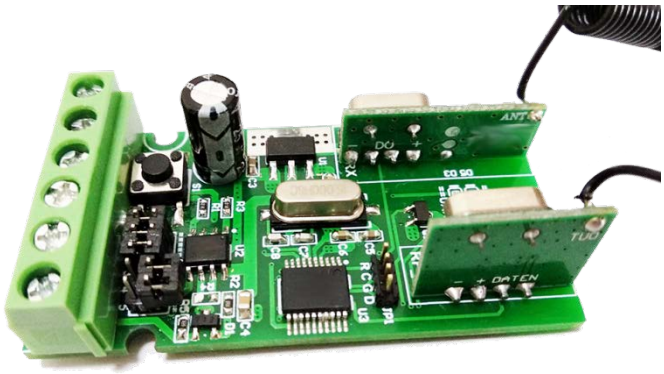
It can be easily applied to any 433MHz remote control system.

2 FEATURES

- Supports two command formats: HEX and ASCII
- Air packet format/speed/encoding can be customized
- Support half-duplex communication
- Data rate: up to 5kbps (single shot), up to 1.5kbps (half duplex)
- High receiving sensitivity: up to -108dBm
- Transmitting power: 10dbm
- Half-duplex communication distance: up to 80m

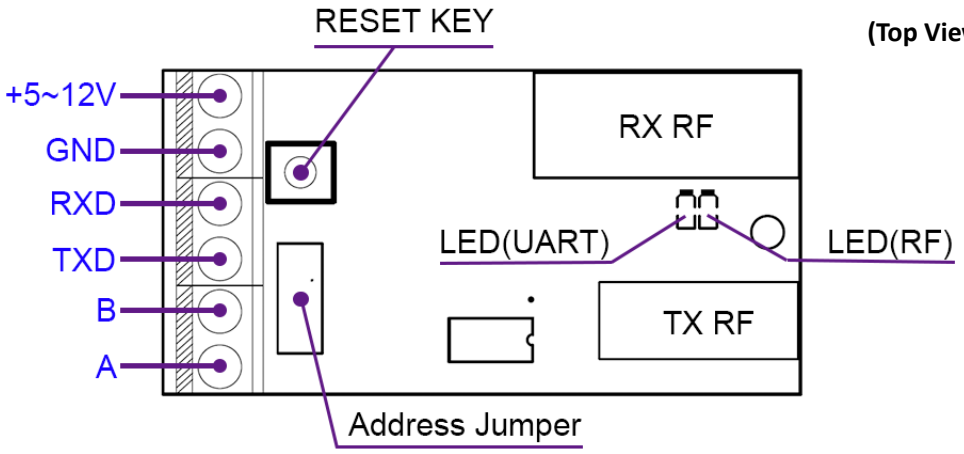
3 APPLICATIONS

- Remote control fan
- Remote control light
- Remote control door
- Remote keyboard
- Remote RFID
- Smart home controller
- Remote control toys
- Wireless encryption device receiver
- Short-distance remote control



RS485 / UART to 433MHz RF Convertor

4 Pin Configuration and Functions

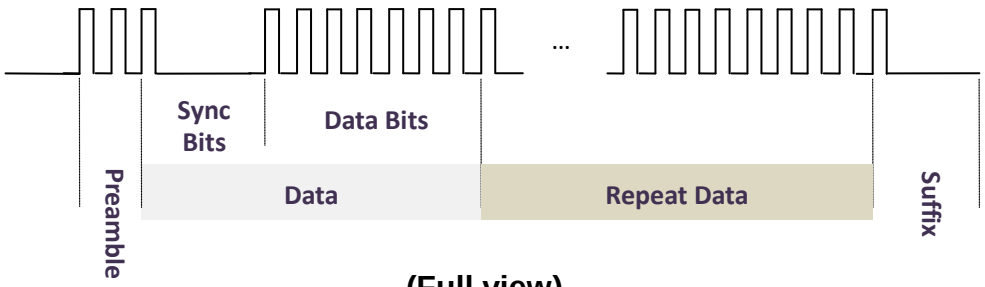


Symbol	Name and function
+5~12V	Power supply
GND	GND
RXD	RXD for UART communication
TXD	TXD for UART communication
B	B for RS485 communication
A	A for RS485 communication
RESET KEY	Reset key While short pressed, reset the communication baudrate to default (9600bps) While long pressed, reset all the system's parameters to default
Address Jumper	Select the address for RS485 BUS
LED(UART)	Indicator for UART data transition
LED(RF)	Indicator for RF data transition

5 Electrical Characteristics

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
Input Voltage	-	5	-	12	V
Operating Temperature	-	-20		85	℃
Supply current	VIN=5V, Idle mode	15.4	15.8	16.0	mA
	VIN=5V, Continuous receiving mode	16.1	16.2	16.3	mA
	VIN=5V,Continuous transmission mode	32	33.5	34	mA
High receiving sensitivity	VIN=5V	-108			dBm
Transmitting power	VIN=5V		10	13	dBm
Receiving distance	VIN=5V,24bits Data	50	80	100	m
Sending distance	VIN=5V,24bits Data	50	60	80	m

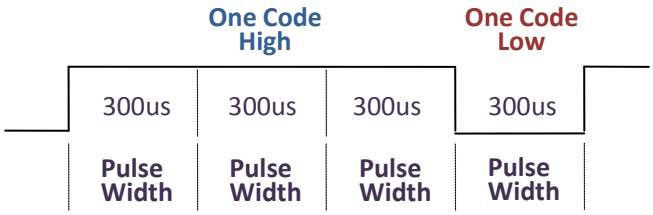
6 Air packet format



(Full view)

- **Preamble:**
  Since the receiving end cannot correctly receive the first few data bits sent by the transmitting end, the Preamble bit is sent in advance to prepare the receiving end for receiving.
- **Sync Bits:**
  The sync bit is used to indicate the start of the packet and prevent the packet from being misaligned.
- **Data Bits:**
  Data bits are valid data bits for transmission.
- **Data:**
  Data is composed of sync bits and data bits.
- **Repeat Data:**
  Repeat data is multiple repetitions of data to prevent loss of transmission.
- **Suffix:**
  Prevents the last valid data bit from being lost

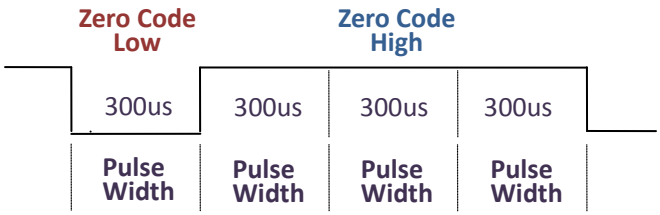
7 Pulse width & Pulse Count



Pulse width: 300us
One Code High pulse count: 3
One Code Low pulse count: 1

(One Code Example)

RS485 / UART to 433MHz RF Convertor



Pulse width: 300us
Zero Code High pulse count: 1
Zero Code Low pulse count: 3

(Zero Code Example)

Pulse width is the base time width. And the high/low level width of all sync headers, one code and zero code are specified by the pulse count. And the high/low level width equal: pulse width * pulse count.
For example, when the pulse width is 300us, the high-level pulse number of 1 code is 3, and the low-level pulse number is 1, then the 1 code high level width is 900us and its low level width is 300us.

8 Data Sending

There are three methods to send data using CT-UR-433M.

1) Method 1: ASCII code (decimal)

Using ASCII Command: Send Data (Decimal)
For example:

-1:1234\r\n

Send data 1234 to RS485 address 1.

2) Method 2: ASCII code (hex)

Using ASCII Command: Send Data (Hex)
For example:

-1:0x4D2\r\n

Send data 0x4d2 (Decimal: 1234) to RS485 address 1.

3) Method 3: HEX code

Using HEX Command: Send Data
For example:

3E 01 21 02 D2 04 0A (Note: all the data is hex byte)

Send data 0x4d2 (Decimal: 1234) to RS485 address 1.

9 Data Receiving

There are two methods to receiving data:

RS485 / UART to 433MHz RF Convertor

1) Method 1: Auto Report

1) By set Report Mode (Reg: 01) to 1, the ASCII report function enabled. Then once CT-UR-433M received any data it will report data like ***-1:1234\r\n*** (data 1234 from address 1).

2) By set Report Mode (Reg: 01) to 2, the ASCII report function enabled. Then once CT-UR-433M received any data it will report data like ***-1: 0x4D2\r\n*** (data 0x4d2, Decimal: 1234, from address 1).

3) By set Report Mode (Reg: 01) to 3, the ASCII report function enabled. Then once CT-UR-433M received any data it will report data like ***3C 01 01 02 D2 04 0A*** (data 0x4d2, Decimal: 1234, from address 1).

2) Method 2: Read Directly

By set Report Mode (Reg: 01) to 0, the directly read operation enabled. Once CT-UR-433M received any data it will temporarily save it to memory. Then you can read it out by the following 3 methods:

A. ASCII Command: **Read Data (Decimal)**

It response the data back as a decimal value string.

For example:

Send: ***-1:D?\r\n***

Response: ****1:1234\r\n*** (data 1234 from address 1).

And it response ****1:FALSE\r\n***(no data).

B. ASCII Command: **Read Data (Hex)**

It response the data back as a hex value string.

For example:

Send: ***-1:H?\r\n***

Response: ****1: 0X0004D2\r\n*** (data 0x4d2, Decimal: 1234, from address 1).

And it response ****1:FALSE\r\n***(no data).

C. HEX Command: **Read Data**

It response the data back as a hex byte value.

For example:

Send: ***3E 01 22 00 0A***

Response: ***3C 01 01 03 D2 04 00 0A*** (data 0x4d2, Decimal: 1234, from address 1).

And it response: ***3C 01 01 00 0A*** (no data).

10 Receive data's detail format

The detail format information of the received data can use the 'Read RX information' command to query back. The detail format contains: pulse width, sync high pulse count, sync low pulse count, one code high pulse count, one code low pulse count, zero code high pulse count, zero code low pulse count.

1) ASCII Command: Read RX information

For example:

Send: ***-1:RX?\r\n***

Response: ****1:300,1,31,3,1,1,3\r\n*** (address: 1, pulse width: 300us, sync high pulse count: 1, sync low pulse count: 31, one code high pulse count:3, one code low pulse count:1, zero code high pulse count:1, zero code low pulse count:3).

And it response ****1: 0,0,0,0,0,0,0 \r\n***(no data format detected).

2) HEX Command: Read RX information

For example:

Send: ***3E 01 23 00 0A***

Response: ***3C 01 01 08 2C 01 01 1F 03 01 01 03 0A*** (data 0x4d2, Decimal: 1234, from address 1).

And it response: ***3C 01 01 08 00 00 00 00 00 00 00 00 0A*** (no data).

Note: The return format information is related to the RX Tolerance (REG: 17), if the RX tolerance is to setting too low the format recognition may be wrong. The recommended tolerance value is 50.

11 Command Format

The CT-UR-433M support both ASCII commands and HEX commands.

1) ASCII Command

The ASCII command string begins with '–' ('*' for response), ends with a newline ('\r' + '\n').
ASCII Value: '-' = 0x2d; '*' = 0x2a
To simplify the expression the table below use '8' for any decimal data, 'A' for any hexadecimal data, "ADDR" for the address of the RS485.

Functions	Send/ Response	*Content*
Report Data(Decimal)	Report	***-ADDR:88888888\r\n***

RS485 / UART to 433MHz RF Convertor

	Report received data in decimal string. "*88888888*" is Fixed 24-bit report data, the value ranges from 0 to 16777215.	
Report Data(hex)	Report	***-ADDR:0XAAAAAA\r\n***
	Report received data in hex string. "*AAAAAA*" is variable length report data, the value ranges from 0x00 to 0XFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF.	
Send Data (Decimal)	Send	***-ADDR:88888888\r\n***
	Response	****ADDR:TRUE\r\n*** ****ADDR:FALSE\r\n***
	Send data in decimal string. "*88888888*" is Fixed 24-bit data, the value ranges from 0 to 16777215.	
Send Data (Hex)	Send	***-ADDR:0XAAAAAA\r\n***
	Response	****ADDR:TRUE\r\n*** ****ADDR:FALSE\r\n***
	Send data in hex string. "*AAAAAA*" is variable length data, the value ranges from 0x00 to 0XFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF.	
Read Data (Decimal)	Send	***-ADDR:D?\r\n***
	Response	****ADDR: 88888888\r\n*** ****ADDR:FALSE\r\n***
	Read the latest received data in decimal, only available at REPORT_MODE is 0. "*88888888*" is Fixed 24-bit data, the value ranges from 0 to 16777215. It'll return FALSE if no any data.	
Read Data (Hex)	Send	***-ADDR:H?\r\n***
	Response	**** ADDR:0XAAAAAA\r\n*** ****ADDR:FALSE\r\n***
	Read the latest received data in hex, only available at REPORT_MODE is 0. "*AAAAAA*" is variable length data, the value ranges from 0x00 to 0XFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF. It'll return FALSE if no any data.	
Read RX information	Send	***-ADDR:RX?\r\n***
	Response	****ADDR:WIDTH,SYNC_HIGH,SYNC_LOW,1_HIGH,1_LOW,0_HIGH,0_LOW\r\n*** ****ADDR:FALSE\r\n***
	Read the detail RX information, such as pulse width, high level pulse count, low level pulse count etc. *WIDTH – Pulse width* *SYNC_HIGH – Sync bits 's high level pulse count* *SYNC_LOW – Sync bits 's low level pulse count* *1_HIGH – One code's high level pulse count* *1_LOW – One code's low level pulse count* *0_HIGH – Zero code's high level pulse count*	

	0_LOW – Zero code's low level pulse count It'll return FALSE if command error.	
Set register	Send	***-ADDR:Rn=888\r\n***
	Response	****ADDR:TRUE\r\n*** ****ADDR:FALSE\r\n***
	Set the register's value 'Rn' express register n, n is the index of the register, please check the register table. '888' is the value of the register. It'll return TRUE if set success, otherwise return FALSE.	
Read register	Send	***-ADDR:Rn?\r\n***
	Response	****ADDR:R01=888\r\n*** ****ADDR:FALSE\r\n***
	Read the register's value 'Rn' express register n, n is the index of the register, please check the register table. '888' is the value of the register. It'll return FALSE if command error.	
Reset All	Send	***-ADDR:RST\r\n***
	Response	****ADDR:TRUE\r\n*** ****ADDR:FALSE\r\n***
	Reset all the register's value to default. It'll return TRUE if set success, otherwise return FALSE.	

2) HEX commands

Protocol packet format

Section	HEAD	ADDR	CMD	LEN	DATA	TAIL
Size (Byte)	1	1	1	1	0~16	1
Value	0x3E 0x3C	0x00~0Xff	See command list	0~16	--	0x0A(\n)

HEX Command List

Functions	Send/Response	CMD	DATA
Report Data	Send	**0x20**	Data 0~15: *Data bytes*
	—		
	Report received data. The data length specified by LEN section.		
Send Data	Send	**0x21**	Data 0~15: *Data bytes*
	Response	**0x01:** Success	

RS485 / UART to 433MHz RF Convertor

		0x02: Error	
	Send data. The data length specified by LEN section.		
Read Data	Send	**0x22**	
	Response	**0x01:** Success **0x02:** Error	Data 0~15: *Data bytes*
	Read data. Only available at REPORT_MODE is 0. The data length specified by LEN section.		
Read RX information	Send	**0x23**	
	Response	**0x01:** Success **0x02:** Error	Data 0~1: *pulse width* Data 2: *Sync bits 's high level pulse count* Data 3: *Sync bits 's low level pulse count* Data 4: *One code's high level pulse count* Data 5: *One code's low level pulse count* Data 6: *Zero code's high level pulse count* Data 7: *Zero code's low level pulse count*
	Read the detail RX information, such as pulse width, high level pulse count, low level pulse count etc.		
Set register	Send	**0x24**	Data 0: *Register Index* Data 1~4: *Register value*
	Response	**0x01:** Success **0x02:** Error	
	Set the register's value. Please check the register table for more detail.		
Read register	Send	**0x25**	Data 0: *Register Index*
	Response	**0x01:** Success **0x02:** Error	Data 0: *Register Index* Data 1~4: *Register value*
	Read the register's value Please check the register table for more detail.		
Reset All	Send	**0x26**	
	Response	**0x01:** Success **0x02:** Error	
	Reset all the registers 's value to default.		

12 Register table

Index	Function	Value	Default Value
1	**Report Mode**	*0: Close Report (Dedicated to RS485 multi-endpoint mode)* *1: ASCII (decimal)* *2: ASCII (hex)* *3: HEX mode*	*1*
2	**Preamble Enable**	*0: Disable* *1: Enable*	*0*
3	**Sync Enable**	*0: Disable* *1: Enable*	*1*
4	**Suffix Enable**	*0: Disable* *1: Enable*	*1*
5	**TX Pulse Width**	*100~1020*	*300*
6	**TX Sync High Pulse Count**	*1~127*	*1*
7	**TX Sync Low Pulse Count**	*1~127*	*31*
8	**TX 0 Code High Pulse Count**	*1~127*	*1*
9	**TX 0 Code Low Pulse Count**	*1~127*	*3*
10	**TX 1 Code High Pulse Count**	*1~127*	*3*
11	**TX 1 Code Low Pulse Count**	*1~127*	*1*
12	**TX Polarity**	*0: Positive polarity (high level send first)* *1: Reverse polarity (low level send first)*	*0*
13	**TX Repeat Count**	*0~15*	*3*
14	**TX Bit order**	*0: LSB* *1: MSB*	*1*
15	**TX Preamble Bits**	*0~16*	*4*
16	**TX Preamble Data**	*0x00~0xFFFF*	*0x55*

17	**RX Tolerance**	*1~50*	*20*
18	**RX Polarity**	*0: Positive polarity (high level send first)* *1: Reverse polarity (low level send first)*	*0*
19	**RX Receive Timeout**	*1~63 (unit: ms)*	*5*
20	**RX Bit Order**	*0: LSB* *1: MSB*	*1*
21	**RX Minimum Bits**	*16~128*	*8*
22	**Baud Rate**	*0x00: 9600bps* *0x01: 14400bps* *0x02: 19200bps* *0x03: 38400bps* *0x04: 56000bps* *0x05: 57600bps* *0x06: 115200bps*	*0*

13 Revision history

Date	Revision	Comment
August 1, 2018	V01	Initial Release