

## CMT211xA & CMT221Xa Communication Example

This chapter will guide the user to carry out the communication experiment on a pair of single transmitting and single receiving chips CMT211xA and CMT221xA. CMT211xA is the OOK modulated single transmitting chip belonging to HoperF's CMOSTEK wireless product line below. The specific models are CMT2110A and CMT2117A. CMT221xA is the corresponding OOK modulated single receiving chip. The specific models are CMT2210A and CMT2217A.

### 1. Tools and software needed to be prepared

- Arduino IDE version 1.0.5
- HopeDuino board(2 pieces)  
(If you have not used the HopeDuino board, please refer to the 《AN0002-HopeDuino Platform Construction Guideline》 )
- USB cable(Type A to Type B)
- CMOSTEK USB Programmer
- CMOSTEK RFPDK V1.38 ( Pay attention to using the latest version. The latest version is V1.38 in the paper )
- Module RFM110 (Based on chip CMT2110A) , module RFM210 (Based on chip CMT2210A) and the matching conversion board



Or module RFM117 (Based on chip CMT2117A), module RFM217 (Based on chip CMT2217A) and the matching conversion board



### 2. CMT211xA and CMT221xA parameter configuration and recording

Chip CMT211xA and chip CMT221xA parameters are configured through CMOSTEK RFPDK software and recorded through CMOSTEK USB Programmer. The specific refer to the CMOSTEK application documentation 《AN102 CMT2110-17A Configuration Guideline》 and 《AN108 CMT2210-17A Configuration Guideline》 . This article is no longer a cumbersome description.


#### ⚠ Notice:

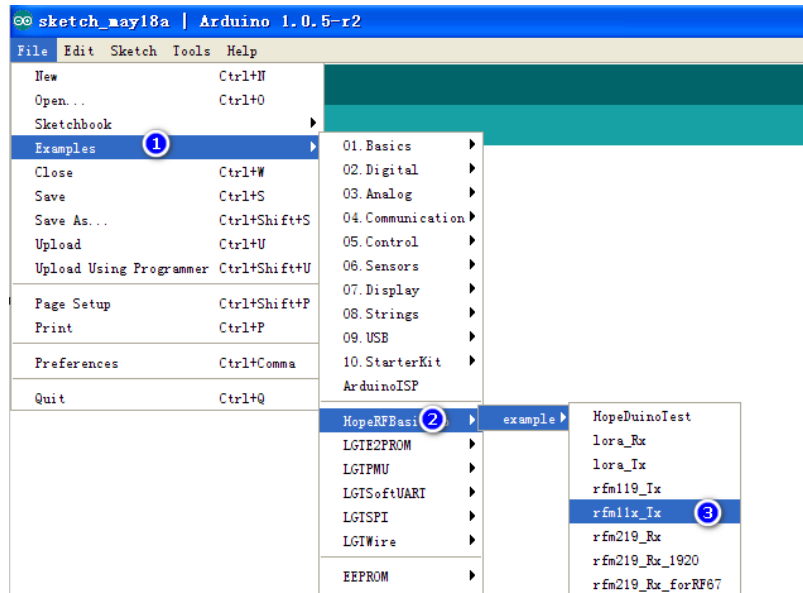
- a) A pair of single transmitting and single receiving modules needs the same operation frequency. For example, CMT2110A is 433.92MHz, and the corresponding CMT2210A is also 433.92MHz.
- b) If the experimental object is a module, such as RFM110 and RFM210, you need to pay attention to the screen printing logo on the module - matching frequency band. If the identity is 433, it is the best to debug in 433M frequency band. It is best to use 433M similar frequency points instead of 315M.

### 3. Hands-on experiment

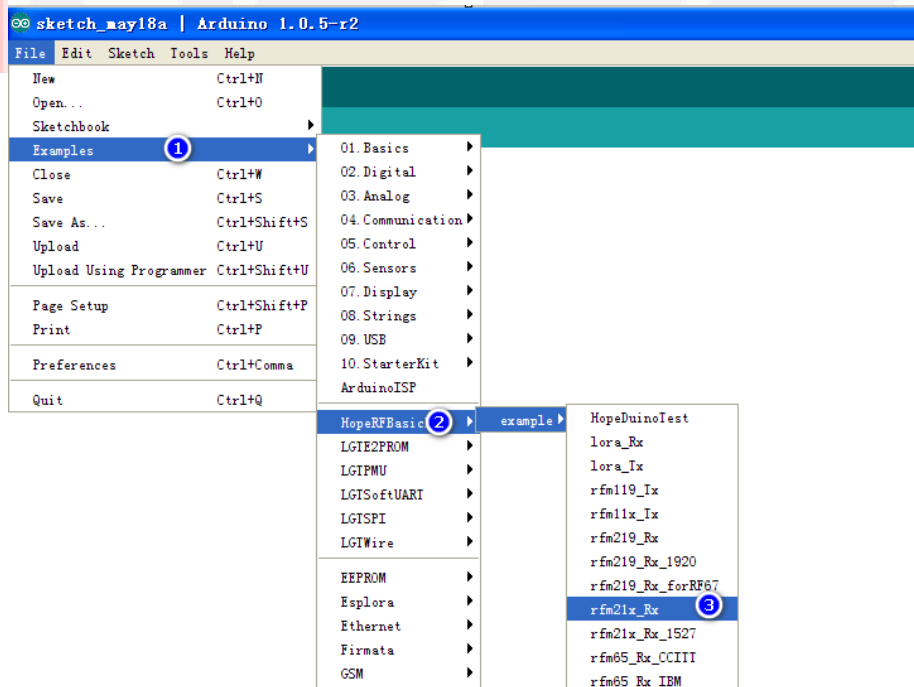
- Module RFM11x and module RFM21x (with conversion board) are respectively inserted into the corresponding HopeDuino board.
- Connect the two HopeDuino boards to the PC through the USB cable;
- Open Arduino IDE interface, click **【File】** → **【Examples】** → **【HoperFBasicLib】** → **【example】** → **【rfm11x\_Tx】** ,as

shown below:

-  Notice: If you did not find [HopeRFBasicLib] in [Examples], it is because the HSP provided by HopeRF is not installed. The specific operation refer to 《AN0002-HopeDuino Platform Construction Guideline》.



- Open Arduino IDE interface, click **【File】** → **【Examples】** → **【HopeRFBasicLib】** → **【example】** → **【rfm21x\_Rx】**, as shown below:



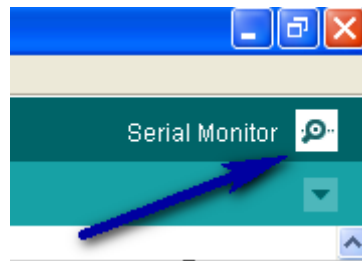
- At this time you have opened a Tx program and a Rx program, please compile the download programs according to the corresponding COM port.

-  Notice:

1. Do not know how to compile the download code, please refer to the 《AN0002-HopeDuino Platform Construction Guideline》
2. HopeDuino platform support multiple boards connected to the same PC. But you need to modify the parameters manually. So you need to pay special attention to which COM port when you download the program every time. COM port is in the lower right corner of Arduino interface, as shown below:



- After the two programs are downloaded, the Tx board will transmit a packet of data through module RFM11x. The Rx board will receive a packet of data through module RFM21x periodically and upload the data to PC through UART (USB). At this point, you can set the COM of Arduino IDE as the port connected with Rx HopeDuino board. Open the “Serial Monitor”, as shown below:



- Click the “Serial Monitor”, pop up the serial port assistant interface, as shown below. Window will display the received data message.



 Notice:

- a) The transmitted data according to program should be “CMOSTEK”. The image above is designed to change the data to “CMT2110”. Also, you can modify the program. Let it show you what you need or other more complex features.
- b) The receiving program enables UART library function. On the description of library function UART, please refer to the "HopeDuino\_UART" library file. It is also stored in the HopeRFLib.

## 4.Program Explanation

- rfm11x\_Tx.ino Case explanation

```
#include <HopeDuino_CMT211xA.h>           // Call corresponding library file
```

```
cmt211xaClass radio; //Define variable radio for CMT211xA

byte str[7] = {'C', 'M', 'O', 'S', 'T', 'E', 'K'}; //Define the message to be transmitted.
//Users can modify their own, add the fun of operation.

void setup() //Setup function is necessary in Arduino software architecture.
{
  radio.Chipset = CMT2110A; //Define chip as CMT2110A
  radio.SymbolTime = 416; // Symbol time is 2.4Ksps (that is 416us)
  radio.vCMT211xAInIt(); //Initialize radio
  radio.vEncode(str, 7, E527); // Execute radio encoding function,
//the source data is srt, the length is 7 bytes.
//Encoding uses 527 encoding format.
}

void loop() //Loop function is necessary in Arduino software architecture.
{
  radio.vTxPacket(); //Transmit a packet of data
  _delay_ms(100); //Delay 100ms.
//That means it will transmit a packet of data automatically every 100ms.
}
```

## ➤ rfm21x\_Rx.ino Case explanation

```
#include <HopeDuino_CMT221xA.h> // Call the corresponding library file.
//Calling UART is added because of using UART.

#include <Hopeduino_UART.h>

cmt221xaClass radio; //Define variable radio for CMT221xA
uartClass uart; //Define variable uart for UART
byte getstr[21]; //Define pending data buffer


void setup()
{
  radio.Chipset = CMT2210A; //Define chip as CMT2210A
  radio.SymbolTime = 416; //Define symbol time as 2.4Ksps(that is 416us).
//This needs to correspond.
  radio.Decode = E527; //Define decoding format as E527.
  radio.vCMT221xAInIt(); //Initialize CMT221xA
  uart.vUartInIt(9600, _8N1); //Initialize UART, parameters are 9600 baud rate and 8N1 format.
}

void loop()
{
  byte i;
```

```

if(radio.bGetMessage(getstr, 7)!=0)    //Check radio whether to receive data function,
                                        //analyze data received.
{
    uart.vUartPutNByte(getstr, 7);    // Output the received data to PC via UART
    uart.vUartNewLine();              //UART newline is easy to read.
    for(i=0; i<7; i++)                //Clear the buffer to receive the data next time.
        getstr[i] = 0x00;
}
}

```

 Notice: In order to deepen the understanding of this case, users can attempt to modify the rate and data content. You need to pay attention to the correspondence with the transmitting and receiving.

## 5. CMT211xA Library Function Description

“CMT211xA.h”and“CMT211xA.cpp”library files are stored in the Arduino IDE file\libraied\HopeRFLib.

### ➤ chipsetType

**Type:** enumeration type

**Function:** Select chip model

**Contents:** CMT2110A、CMT2113A、CMT2117A、CMT2119A

### ➤ encodeType

**Type:** enumeration type

**Function:** Select encoding type

**Contents:** ENRZ、E527、E201

ENRZ——Represents NRZ( Non-Return-Zero )encoding format. Simple explanation is not coding. “1” is the high level. “0” is the low level.

E527——Represents 1527/527 encoding format. Each bit consists of 4 unit time, as shown below:



E201——Represents a code consisting of 3 unit time, shown as below:



### ➤ Chipset

**Type:** enumeration type

**Function:** Define chip model

### ➤ SymbolTime

**Type:** unsigned int

**Function:** Define rate, which is symbol time (SYM for short). The unit is microsecond (us). The setting range is

10~4000.

➤ **vCMT211xAInit**

**Type:** Function

**Input:** None

**Output:** None

**Function:** The initialization is suitable for module CMT2110A, CMT2113A and CMT2117A. Call it at the beginning of program. The initialization is mainly for the IO configuration and reset operation for MCU. It cannot configure the working parameters of the CMT211xA. These parameters must be made by RFPDK software through USB Programmer.

➤ **vCMT211xASleep**

**Type:** Function

**Input:** None

**Output:** None

**Function:** Configure chip to enter low power consumption state for CMT2110A, CMT2113A and CMT2117A. It is suitable for the situation with reducing electricity consumption and getting the chip (module) into hibernation.

➤ **vEncode**

**Type:** Function

**Input:** **ptr[ ]**, unsigned char, pointer, the calling entrance (pointer) of array to be encoded.

**Length**, unsigned char, the data length to be encoded. The unit is byte.

**etype**, encode type, enumeration type, select encoding format (ENRZ, E1527 and E201).

**Output:** None

**Function:** Encode the data to be transmitted according to the specified encoding format.

➤ **vTxPacket**

**Type:** Function

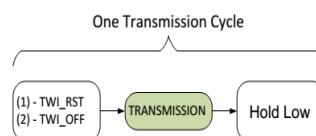
**Input:** None

**Output:** None

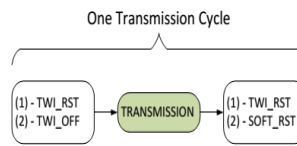
**Function:** Transmit the coded data. You must call vEncode before calling the function. Two functions must be used successively. Call once, and transmit a packet of data. And then return to sleep. If you need to transmit multiple times, you need to repeat the cycle call.

⚠ Notice:

1. The library function control way for CMT211xA is TWI. It is two wire control mode. It is mainly to facilitate the chip to reset and fast sleep. If the user needs to use a single control mode, can only reference encoding (vEncode) and transmitting (vTxPacket) part of the code.
2. The library function vTxPacket uses the transmission process shown as below



The transmission process recommended by CMT211xA specification is shown as below.



The main difference is the end of the final process. The actual effect is the same. If you use HoldLow way, the time keeping the DAT port low level is longer than StopTime time set by CMT211xA. Chip automatically enters low power consumption state. If you use the combination way of TWI\_RST and SOFT\_RST, chip can be more quickly into the low power consumption state. That means the chip does not wait for StopTime time after the execution of the command. Note that the execution of the SOFT\_RST command will enable the chip to reset. After the chip is reset, it will enter the low power consumption state.

3. This paper mainly describes the use of CMT211xA and CMT221xA in HopeDuino. It does not explain the operation process of CMT211xA and CMT221xA. If you have any questions, please refer to the corresponding specification description of CMT211xA CMT221xA.

## 6. CMT221xA Library Function Description

“CMT221xA.h”and“CMT221xA.cpp”library files are stored in the Arduino IDE file\libraries \ HoperFLib;

### ➤ chipsetType

**Type:** enumeration type

**Function:** Select chip model

**Contents:** CMT2210A、CMT2213A、CMT2217A(Due to the reasons for the version design, CMT2219A is packaged as another library file)

### ➤ decodeType

**Type:** enumeration type

**Function:** Select decoding type

**Contents:** E527、E201(Due to the reasons of the version design, the corresponding decoding process is only aimed at E527 and E201.)

### ➤ Chipset

**Type:** enumeration type

**Function:** Define chip model

### ➤ Decode

**Type:** enumeration type

**Function:** Select decoding format

### ➤ SymbolTime

**Type:** unsigned int

**Function:** Define rate,that is symbol time (SYM for short). The unit is microsecond (us). The setting range is 10~4000.

### ➤ vCMT221xAInit

**Type:** Function

**Input:** None

**Output:** None

Function: The initialization is suitable for module CMT2210A、CMT2213A、CMT2217A. Call it at the beginning of program. The initialization is mainly for the IO configuration of MCU not to configure the working parameters of the CMT221xA. These parameters must be burned by USB Programmer through RFPDK software.

➤ **bReadRssi**

**Type:** Function

**Input:** None

**Output:** Signal intensity range is from 0 to 255, the stronger signal, the greater value.

**Function:** Carry out a RSSI (signal strength) conversion and read the results.

➤ **bGetMessage**

**Type:** Function

**Input:** `msg[ ]`, **pointer variable, the buffer entrance of received data.**

**Pklen**, **unsigned char, the length of received data. The unit is byte.**

**Output:** Returns the length of the received data content. The unit is byte. 0 indicates that there is no data received

**Function:** Detect whether to receive data. The function is to query the data flow mechanism and not to interrupt processing mechanism. It needs to be embedded in the loop function when the software calls it. At the same time, if there is other process function in the repeated cycle, and the process function takes up more time, it will affect the reception decoding effect. So you need to ensure that this function is the highest priority call.



## 7. Pin Assignment Table:

HopeDuino	MCU	CMT2110/3/7/9A	CMT2210/3/7A
13	PB5		
12	PB4	CLK	
11	PB3		
10	PB2		
9	PB1		
8	PB0		
7	PD7		CSB
6	PD6	DAT	DOUT
5	PD5		SDA
4	PD4		SCL



## 8. Version Records:

Version	Revised Contents	Date
1.0	Initial version	2016-03-29
1.1	Add watermarks, program explanations and descriptions	2016-04-06

HOPERF