Q：How to use the BMB22M210 module to communicate with other UART modules?

A：Since the Arduino Libraries for both the BMB22M210 module and other UART modules are designed for the BMduino development board, they are not directly compatible with each other. To use the BMB22M210 module for UART interface expansion, need to modify the underlying driver of the Arduino Library for the other UART module and replace its communication part with functions provided by the BMB22M210.

Using BMS31M001 as an example, here is the integration approach:

● Original setup：

→BMB22M210 module's Arduino Library: Handles communication between the BMduino development board and the BMB22M210 module, and controls sub-serial ports for data transmission/reception.

→BMS31M001 module's Arduino Library: Handles direct communication between the BMduino development board and the BMS31M001 module to read IR proximity state.

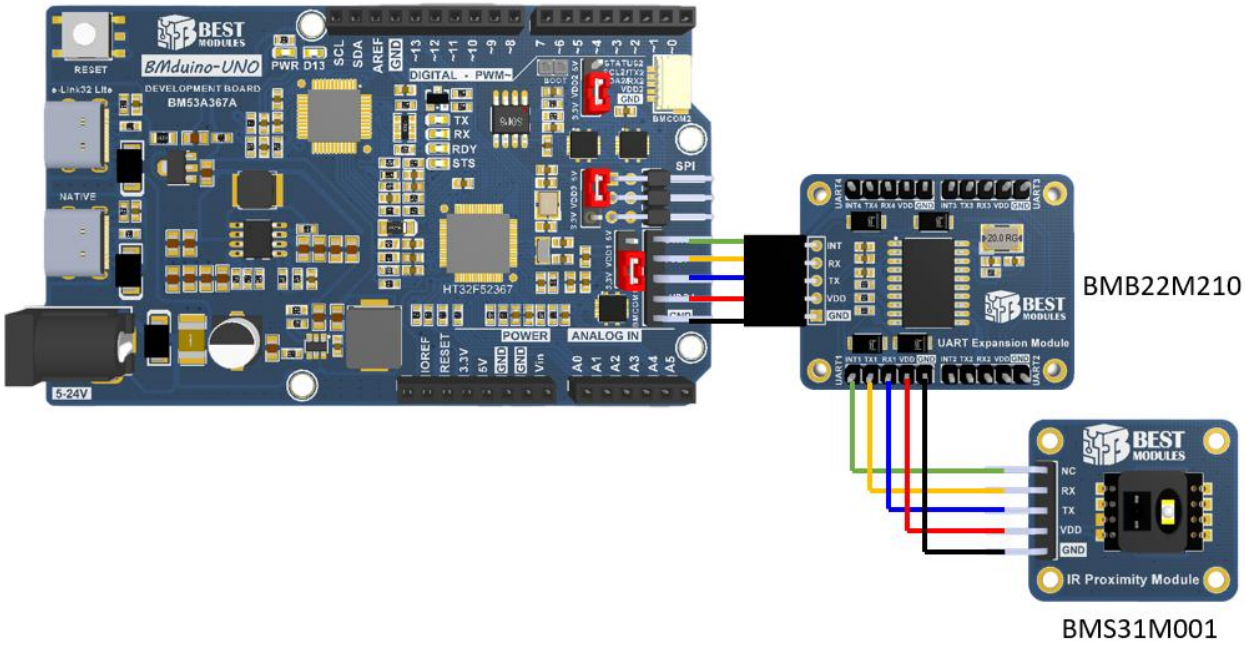
● After modifying BMS31M001’s Arduino Library to use BMB22M210 for communication:

→Sending data: The development board communicates with the BMB22M210, which then communicates with the BMS31M001 module.

→Receiving data: The BMS31M001 module sends data to the BMB22M210, which then forwards it to the development board.

◆ Example: Using the BMB22M210 module to communicate with the proximity sensing module BMS31M001.

● Connection Diagram: The BMB22M210 is connected to the BMCOM1 of the BMduino, and the BMS31M001 is connected to Port 1 of the BMB22M210.



● "BM32S2031-1.h" File Modifications

（1） Include Header File BMB22M210.h

→Required for enabling communication via BMB22M210 driver

```
#ifndef _BM32S2031_1_H_
#define _BM32S2031_1_H_

#include <Arduino.h>
#include <SoftwareSerial.h>
//*************** Change use BMB22M210,BY 20250506 ***************//
//************************************************************//
#include <BMB22M210.h>
//************************************************************//
```

（2） Refactor Constructor

→The original constructor created an object directly based on the interface of the development board. It is now revised to accept parameters related to the BMB22M210 interface: (intPin: the INT pin of the BMB22M210. *theSerial: the selected UART interface of the BMB22M210. port: the port of the BMB22M210 to which the BMS31M001 module is connected.)

```
class BM32S2031_1
{
    public:
    //*************** Change use BMB22M210,BY 20250506 ***************//
    //************************************************************//
    //BM32S2031_1(HardwareSerial *theSerial  = &Serial);
    //BM32S2031_1(uint8_t rxPin,uint8_t txPin);
    BM32S2031_1(uint8_t intPin , HardwareSerial *theSerial=&Serial , uint8_t port=1);
    //************************************************************//
    void begin(uint16_t baud = BM32S2031_1_BAUD);
```

（3） Create Private Variables

→A pointer to the BMB22M210 class, named mySerial, used to call BMB22M210 functions.

→A private variable _port, allowing the Port interface connected to BMS31M001 to be switched.

```
    private:
    uint8_t  readIRRef();
    uint8_t  readIREnv();
    void writeBytes(uint8_t wbuf[], uint8_t wlen);
    uint8_t readBytes(uint8_t rbuf[], uint8_t rlen, uint16_t timeOut = 10);
    uint16_t _rxPin;
    uint16_t _txPin;
    HardwareSerial *_hardSerial ;
    SoftwareSerial *_softSerial ;
    //*************** Change use BMB22M210,BY 20250506 ***************//
    //************************************************************//
    BMB22M210 *mySerial;
    uint8_t _port;
    //************************************************************//
};

#endif
```

- "BM32S2031-1.cpp" File Modifications

（1） Write New Constructor Content

→Use the constructor to create an instance of the BMB22M210 module, which is stored in the mySerial pointer. (intPin: the INT pin of the BMB22M210; *theSerial: the selected UART interface for BMB22M210)

→The _port variable stores the selected port number.

```
//*************** Change use BMB22M210,BY 20250506 ***************//
//****************************************************************//
BM32S2031_1::BM32S2031_1(uint8_t intPin,HardwareSerial *theSerial,uint8_t port)
{
  _softSerial = NULL;
  _hardSerial = theSerial;
  _mySerial = new BMB22M210(intPin,theSerial);
  _port = port;
}
//****************************************************************//
```

The original constructor content should be remark.

（2） Modify the begin Function

```
void BM32S2031_1::begin(uint16_t baud)
{
  //*************** Change use BMB22M210,BY 20250506 ***************//
  //****************************************************************//
    if(_softSerial != NULL)
    {
        _softSerial->begin(baud);
        getFWVer(); //  The UART mode is displayed (Any instruction)
    }
    else
    {
        //*************** Change use BMB22M210,BY 20250506 ***************//
        //****************************************************************//
        //_hardSerial->begin(baud);
        _mySerial->begin(baud);          //initial module
        _mySerial->beginPort(_port,baud);    //set uart port1 baud：9600
        //****************************************************************//
        getFWVer();  //  The UART mode is displayed (Any instruction)
    }

  //****************************************************************//
}
```

（3） Modify Low-Level Communication Functions writeBytes / readBytes

→Changed from direct data transmission and reception through the development board to data handling via BMB22M210.

```
void BM32S2031_1::writeBytes(uint8_t wbuf[], uint8_t wlen)
{
  /* Select SoftwareSerial Interface */
  if (_softSerial != NULL)
  {
    while (_softSerial->available() > 0)
    {
      _softSerial->read();
    }
    _softSerial->write(wbuf, wlen);
  }
  /* Select HardwareSerial Interface */
  else
  {
//*************** Change use BMB22M210,BY 20250506 ****************//
//*****************************************************************//
    // while (_hardSerial->available() > 0)
    // {
    //   _hardSerial->read();
    // }
    // _hardSerial->write(wbuf, wlen);
    while (_mySerial->available(_port) > 0)
    {
      _mySerial->read(_port);
    }
    _mySerial->writeBytes(_port,wbuf, wlen);
//*****************************************************************//
  }
}
```

```
uint8_t BM32S2031_1::readBytes(uint8_t rbuf[], uint8_t rlen, uint16_t timeOut)
{
  uint8_t i = 0, delayCnt = 0, checkSum = 0;
  // for (i = 0; i < rlen; i++)
  // {
  //   delayCnt = 0;
  //   while (_hardSerial->available() == 0)
  //   {
  //     if (delayCnt > timeOut)
  //     {
  //       return TIMEOUT_ERROR; // Timeout error
  //     }
  //     delay(1);
  //     delayCnt++;
  //   }
  //   rbuf[i] = _hardSerial->read();
  // }
  for (i = 0; i < rlen; i++)
  {
    delayCnt = 0;
    while (_mySerial->available(_port) == 0)
    {
      if (delayCnt > timeOut)
      {
        return TIMEOUT_ERROR; // Timeout error
      }
      delay(1);
      delayCnt++;
    }
    rbuf[i] = _mySerial->read(_port);
  }
*****************************************************************//
```

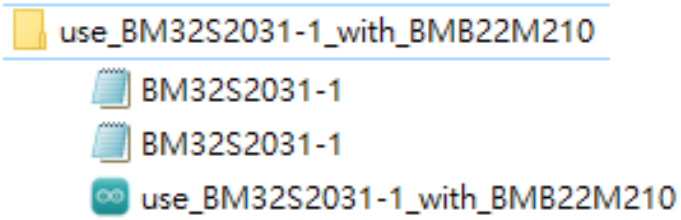
- “.ino” Example File Modifications

（1） When creating an object, select the constructor parameters based on the new constructor.

```
//*************** Change use BMB22M210,BY 20250506 ***************//
//****************************************************************//
//BM32S2031_1     BMS31(&Serial1);
BM32S2031_1 BMS31(22,&Serial1,1); //BMB22M210_intPin:22
                                  //BMB22M210_Serial:&Serial1
                                  //BM32S2031_1_ON_BMB22M210_port:Port1
//****************************************************************//
```

Notes:

1. Refer to the example project: use_BM32S2031-1_with_BMB22M210 for modification guidance.



2. Direct communication between the development board and the module offers higher communication efficiency. When using the BMB22M210 for UART expansion, communication performance will be relatively slower due to the following reasons:

→ Sending data: The development board first communicates with the BMB22M210, and then the BMB22M210 communicates with the module.

→ Receiving data: The module sends data to the BMB22M210, and then the development board reads it from the BMB22M210.