NFC 读卡器扩充板

BMC11T001
Arduino Library V1.0.1 说明

版本：V1.00　日期：2023-08-30

www.bestmodulescorp.com

目录

简介

BMC11T001 是倍创推出的一款 NFC 读卡器扩充板，使用 UART 通信方式。本文档对 BMC11T001 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明；范例演示了对卡进行 UID 读取以及数据读写等功能。

Arduino Lib 函数

Arduino Lib 名称：BMC11T001		Lib 版本：V1.0.1
构造函数 & 初始化		
1	BMC11T001(HardwareSerial *theSerial = &Serial4)	
	描述	构造函数，使用 HW Serial 接口
	参数	*theSerial：选择 HW Serial 接口 (固定 Serial4 接口)
	返回值	—
	备注	—
2	void begin(uint32_t baud = BMC11T001_BAUD)	
	描述	对象初始化
	参数	baud：通信速率选择 115200(BMC11T001_BAUD)：固定为 115200bps
	返回值	void
	备注	—
ISO14443A 相关函数		
3	void_t begin_ISO14443A()	
	描述	初始化 ISO14443A 协议
	参数	—
	返回值	void
	备注	—
4	uint8_t getUID_ISO14443A(uint8_t buff[])	
	描述	获取 ISO14443A 卡的 UID
	参数	buff[]：存放含 UID 的字符串
	返回值	含 UID 字符串的长度
	备注	buff[] 的长度建议设为 50 字节
5	uint8_t readType2Tag_ISO14443A(uint8_t addr,uint8_t buff[])	
	描述	读取 Type 2 Tag 卡的 Memory 数据
	参数	addr：地址，范围：00h~2Ch buff[]：存放含 addr~addr+3 对应值的字符串
	返回值	含 addr~addr+3 对应值的字符串的长度
	备注	buff[] 的长度建议为 41 字节

6	uint8_t writeType2Tag_ISO14443A(uint8_t addr,long data)	
	描述	向 Type 2 Tag 卡的 Memory 写入数据
	参数	addr：地址，范围：04h~27h data：需写入的数据，长度固定为 4 字节
	返回值	执行情况 0x00：成功 0x01：失败
	备注	—
7	uint8_t readMifareBlock_ISO14443(uint8_t keyType[],uint8_t key[],uint8_t addr,uint8_t buff[])	
	描述	读取 Mifare 卡的 Memory 数据
	参数	keyType[]：密钥类型，可为“ka”或者“kb” key[]：密钥，“FFFFFFFFFFFF” addr：地址 Mifare 1K 的卡，范围：00h~3Fh Mifare 4K 的卡，范围：00h~FFh buff[]：存放含 addr 对应值的字符串
	返回值	含 addr 对应值的字符串的长度
	备注	buff[] 的长度建议设为 38 字节
8	uint8_t writeMifareBlock_ISO14443A(uint8_t keyType[],uint8_t key[],uint8_t addr,uint8_t data[])	
	描述	向 Mifare 卡的 Memory 写入数据
	参数	keyType[]：密钥类型，可为“ka”或者“kb” key[]：密钥，“FFFFFFFFFFFF” addr：地址 Mifare 1K 的卡，范围：01h~3Fh Mifare 4K 的卡，范围：01h~FFh data[]：需写入的数据，长度为 32 字节
	返回值	执行情况 0x00：成功 0x01：失败
	备注	—
ISO14443B 相关函数		
9	void begin_ISO14443B(void)	
	描述	初始化 ISO14443B 协议
	参数	void
	返回值	void
	备注	—
10	uint8_t getUID_ISO14443B(uint8_t buff[])	
	描述	获取 ISO14443B 卡的 UID
	参数	buff[]：存放含 UID 的字符串
	返回值	含 UID 字符串的长度
	备注	仅限标准 B 卡，buff[] 的长度建议设为 50 字节

11	uint8_t initiate_ST25TB512_ISO14443B(uint8_t buff[])	
	描述	初始化 ST25TB512 卡
	参数	buff[]：存放含随机数的字符串
	返回值	含随机数字符串的长度
	备注	buff[] 的长度建议设为 12 字节
12	uint8_t selectChipID_ST25TB512_ISO14443B(uint8_t random[],uint8_tbuff[])	
	描述	选择需操作的 ST25TB512 卡的 Chip ID
	参数	random[]：初始化 ST25TB512 卡后获取的含随机数的字符串 buff[]：存放含随机数的字符串
	返回值	含随机数字符串的长度
	备注	buff[] 的长度建议设为 12 字节
13	uint8_t getUID_ST25TB512_ISO14443B(uint8_t buff[])	
	描述	从已选择 Chip ID 的 ST25TB512 卡获取 UID
	参数	buff[]：存放含 UID 的字符串
	返回值	含 UID 字符串的长度
	备注	buff[] 的长度建议设为 26 字节
14	uint8_t readBlock_ST25TB512_ISO14443B(uint8_t addr,uint8_t buff[])	
	描述	从已选择 Chip ID 的 ST25TB512 中读取 Memory 数据
	参数	addr：地址，范围：00h~0Fh, FFh buff[]：存放含 addr 对应值的字符串
	返回值	含 addr 对应值字符串的长度
	备注	buff[] 的长度建议为 21 字节
15	void writeBlock_ST25TB512_ISO14443B(uint8_t addr,long data)	
	描述	从已选择 Chip ID 的 ST25TB512 中写入 Memory 数据
	参数	addr：地址，范围：07h~0Fh data：需写入的数据，长度为 4 字节
	返回值	void
	备注	—
ISO15693 相关函数		
16	void begin_ISO15693(void)	
	描述	初始化 ISO15693 协议
	参数	—
	返回值	void
	备注	—
17	uint8_t getSingleUID_ISO15693(uint8_t buff[])	
	描述	获取单张 ISO15693 卡的 UID
	参数	buff[]：存放含 UID 的字符串
	返回值	含 UID 字符串的长度
	备注	buff[] 的长度建议设为 23 字节

18	uint8_t getMultipleUID_ISO15693(uint8_t buff[])	
	描述	获取多张 ISO15693 卡的 UID
	参数	buff[]：存放含多个 UID 的字符串
	返回值	含 UID 字符串的长度
	备注	buff[] 的长度建议设为 120 字节，同时读取的卡的数量最多 4~5 张
19	uint8_t readBlock_ISO15693(uint8_t addr,uint8_t uid[],uint8_t buff[])	
	描述	读取 ISO15693 卡的 Memory 数据
	参数	addr：地址，范围：00h~1Bh uid[]：获取单个 ISO15693 卡 UID 后得到含 UID 的字符串 buff[]：存放含 addr 对应值的字符串
	返回值	含 addr 对应值字符串的长度
	备注	buff[] 的长度建议为 18 字节
20	uint8_t writeBlock_ISO15693(uint8_t addr,long data,uint8_t uid[])	
	描述	向 ISO15693 卡的 Memory 写入数据
	参数	addr：地址，范围：00h~1Bh data：需要写入的数据，长度为 4 字节 uid[]：获取单个 ISO15693 卡 UID 后得到含 UID 的字符串
	返回值	执行情况 0x00：成功 0x01：失败
	备注	—
Advanced 功能函数		
21	void turnoffRFField_ADVANCED(void)	
	描述	关闭 RF
	参数	void
	返回值	void
	备注	—
22	void resetRFField_ADVANCED(void)	
	描述	重启 RF
	参数	void
	返回值	void
	备注	—
23	void enableBuzzer_ADVANCED(void)	
	描述	蜂鸣器使能
	参数	void
	返回值	void
	备注	对卡操作成功时，蜂鸣器将会响一声
24	void disableBuzzer_ADVANCED(void)	
	描述	蜂鸣器除能
	参数	void
	返回值	void
	备注	—

25	uint8_t getSingleCardDetectResult_ADVANCED(void)	
	描述	检测是否有卡 (单卡)
	参数	void
	返回值	是否检测到卡 0：没检测到卡 1：检测到卡
	备注	使用该函数之前需根据检测卡的类型，进行初始化协议
26	uint8_t getMultiCardDetectResult_ADVANCED(void)	
	描述	检测是否有卡 (仅支持 ISO15693，可多卡)
	参数	void
	返回值	是否检测到卡 0：没检测到卡 1：检测到卡
	备注	使用该函数之前需使用函数 16 "void begin_ISO15693(void)" 进行初始化 ISO15693 协议
27	void setMCUPowerMode_ADVANCED(uint8_t powermode)	
	描述	设定 MCU 工作模式
	参数	powermode：MCU 工作模式 0x00(SLEEP)：Sleep 模式 0x01(DEEP_SLEEP)：Deep Sleep 模式 0x02(NORMAL)：Normal 模式
	返回值	void
	备注	仅卡检测模式可用
28	void setMCUWakeupTime_ADVANCED(uint8_t wakeuptime)	
	描述	设定 MCU 自动唤醒时间间隔
	参数	wakeuptime：MCU 自动唤醒时间间隔 0x00(Period_100MS)：间隔 100ms 0x01(Period_200MS)：间隔 200ms 0x02(Period_500MS)：间隔 500ms 0x03(Period_1000MS)：间隔 1000ms
	返回值	void
	备注	仅卡检测模式可用
29	void setCDCalibration_ADVANCED(void)	
	描述	射频校准
	参数	void
	返回值	void
	备注	仅卡检测模式可用，需确保天线附近没有 Tag 或者金属对象

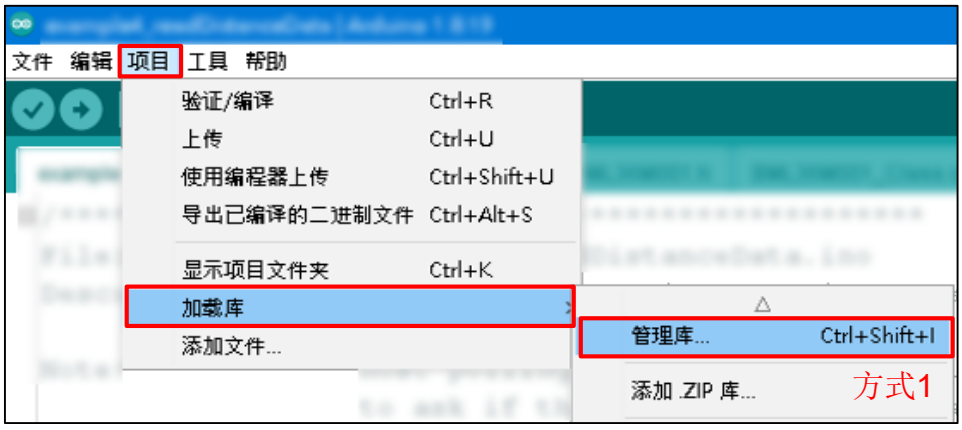
30	void startLowPowerCardDetection_ADVANCED(uint8_t cardType)	
	描述	启动卡检测模式
	参数	cardType：选择需检测卡的类型 0x01：NFC-A 0x02：NFC-B 0x03：NFC-A & NFC-B 0x04：NFC-V 0x05：NFC-A & NFC-V 0x06：NFC-B & NFC-V 0x07：NFC-A & NFC-B & NFC-V
	返回值	void
	备注	仅卡检测模式可用，需确保天线附近没有 Tag 或者金属对象
31	uint8_t scaningLowPowerCardDetection_ADVANCED(uint8_t buff[])	
	描述	卡检测模式下获取 UID
	参数	buff[]：存放检测到的卡的含 UID 值的字符串
	返回值	含 UID 值字符串的长度
	备注	仅卡检测模式可用。当检测到卡后，会返回 UID 的字符串；buff[] 的长度建议为 100 字节
32	void stopLowPowerCardDetection_ADVANCED(void)	
	描述	停止卡检测模式
	参数	void
	返回值	void
	备注	仅卡检测模式可用

Arduino Lib 下载及安装

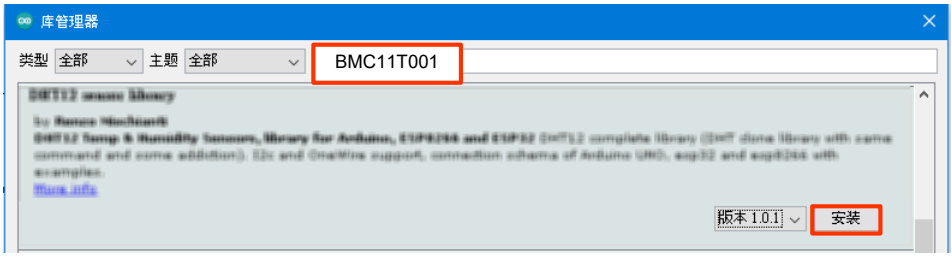
BMC11T001 Library：可参考下面两种方法安装 BMC11T001 的 Arduino Library

方式 1：搜索安装

搜索安装：Arduino IDE → 项目 → 加载库 → 管理库 → 搜索 BMC11T001 → 安装



搜索安装流程 1

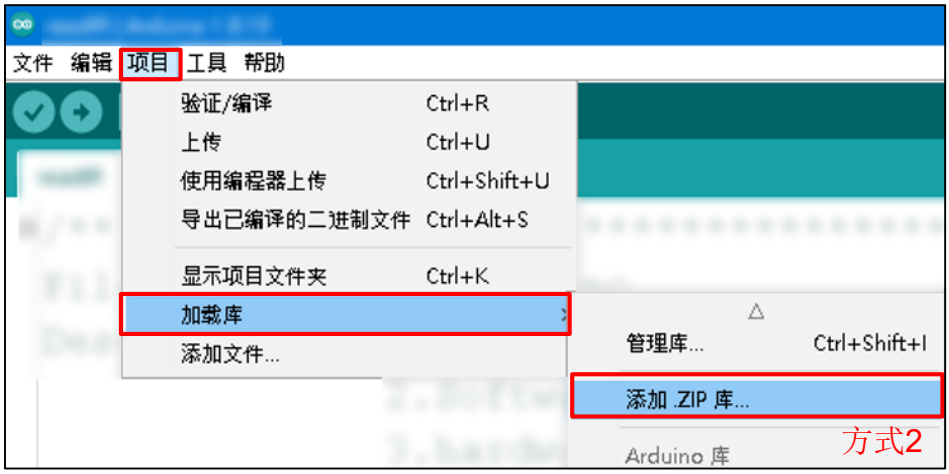


搜索安装流程 2

方式 2：添加 .ZIP 库，需提前下载 .ZIP 库

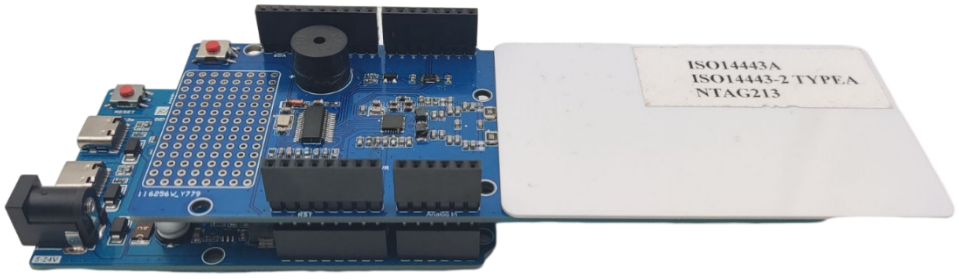
下载方法：打开倍创官方网站 (https://www.bestmodulescorp.com/bmc11t001.html) 文件目录下的 Arduino 范例程序 (BMC11T001 Library)。

添加 .ZIP 库：Arduino IDE → 项目 → 加载库 → 添加 .ZIP 库 ....



Arduino 范例

范例 1：ISO14443AType2Tag_Write_Read



实物连接示意图

范例实现功能：对 ISO14443A Type2 Tag 卡 UID 读取以及数据读写，并在串口监视器上显示。

1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMC11T001) → 选择范例 (ISO14443A_Type2Tag_Write_Read)
2. 示例说明：

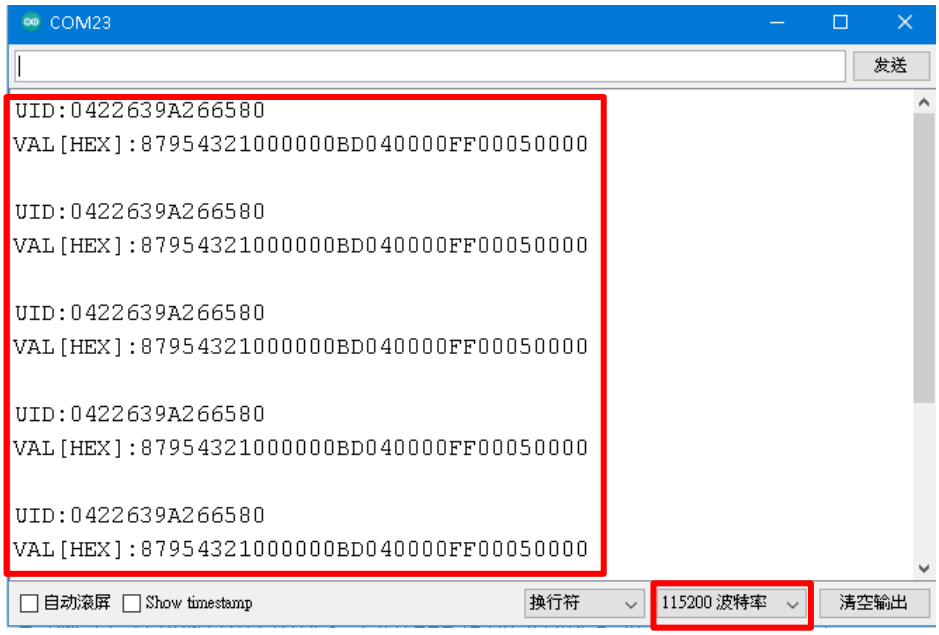
a. 创建对象 & 模块初始化及配置

```
#include "BMC11T001.h"
BMC11T001     BMC11(&Serial4);   // 创建对象，使用 Serial4 接口
int nlens;
uint8_t uid_buf[50] = {0};
uint8_t DATA[41] = {0};
void setup()
{
  delay(1000);                  // 等待扩充板上电初始化
  BMC11.begin(115200);          // 扩充板初始化
  Serial.begin(115200);         // 配置串口监视器
  BMC11.begin_ISO14443A();      // 初始化 ISO14443A 协议
}
```

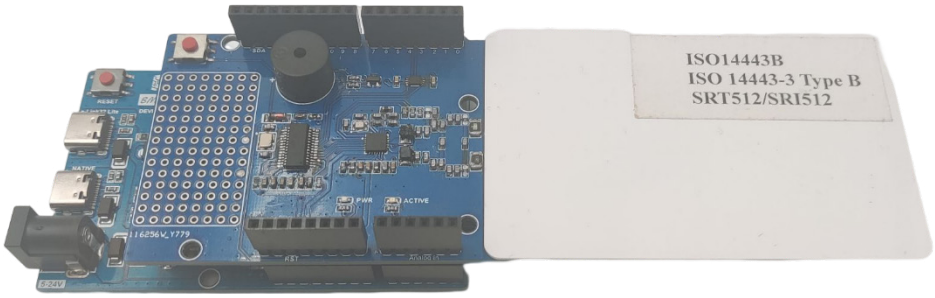
b. 获取 ISO14443A Type2 Tag 卡的 UID 并在串口监视器上显示，对卡的数据先进行写然后读取，获取的数据在串口监视器上显示。

```
void loop()
{
   /***** ISO14443A Type2 Tag 卡 UID 读取 *****/
    nlens = BMC11.getUID_ISO14443A(uid_buf); // 获取 UID
    Serial.write(uid_buf, nlens); // 获取到的 UID 显示在串口监视器上
    Serial.println(" ");
   /****ISO14443A Type2 Tag 卡数据读写 ****/
   if(!BMC11.writeType2Tag_ISO14443A(0x27,0x87954321)) // 写数据
   {
    nlens =  BMC11.readType2Tag_ISO14443A(0x27,data_buf); // 读数据
    Serial.write(data, nlens);  // 获取到的数据显示在串口监视器上
    Serial.println(" ");
   }
  Serial.println(" ");
  delay(2000);
}
```

3. 打开串口监视器，波特率选择 115200；串口监视器显示如下：



范例 2：SRT512_Write_Read



实物连接示意图

范例实现功能：SRT512 卡 UID 读取以及数据读写，并在串口监视器上显示。

1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMC11T001) → 选择范例 (SRT512_Write_Read)

2. 示例说明：

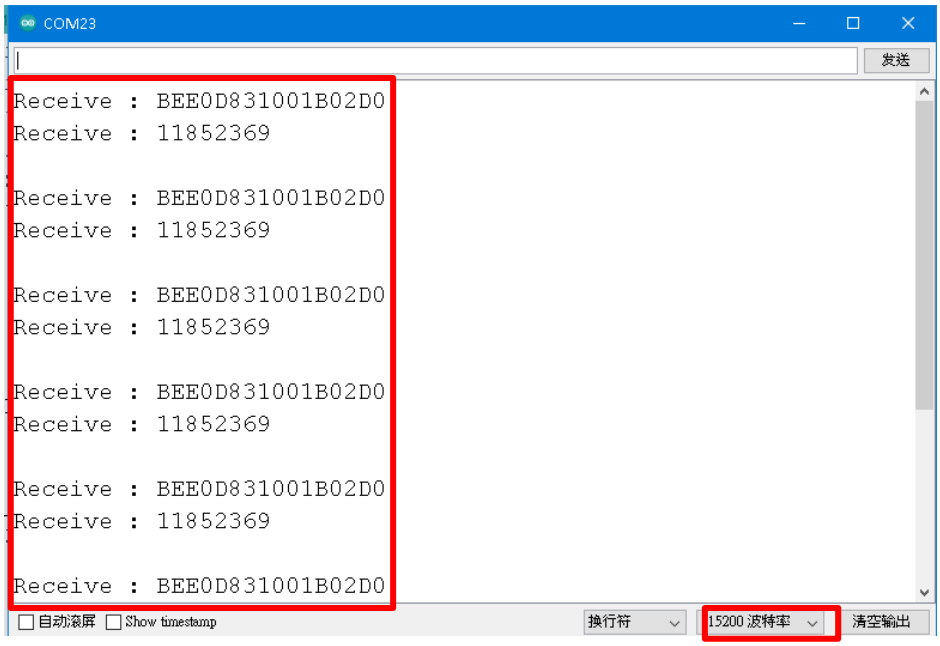
a. 创建对象 & 模块初始化及配置

```
#include "BMC11T001.h"
BMC11T001     BMC11(&Serial4);   // 创建对象，使用 Serial4 接口
int nlens;
uint8_t uid_buf[26] = {0};
uint8_t Random[12] = {0};
uint8_t Random1[12] ={0};
uint8_t data[21] = {0};
void setup()
{
  delay(1000);      // 等待扩充板上电初始化
  BMC11.begin(115200);     // 扩充板初始化
  Serial.begin(115200);       // 配置串口监视器
}
```

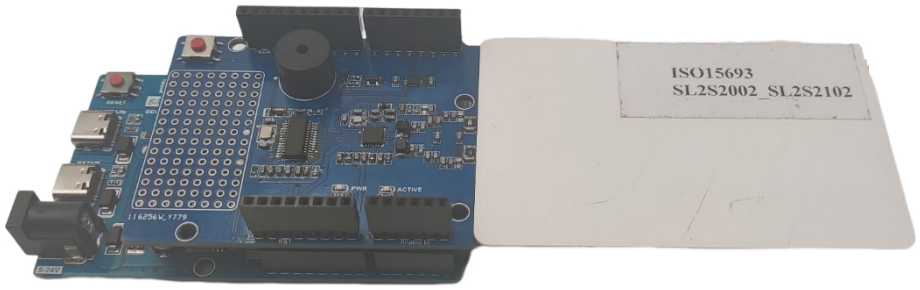
b. 获取 ST25TB512 卡的 UID 并在串口监视器上显示，对卡的数据先进行写然后读取，获取的数据在串口监视器上显示。

```
void loop()
{
    BMC11. begin_ISO14443B (); // 初始化 ISO14443B 协议
    BMC11.initiate_ST25TB512_ISO14443B(Random); // 向卡获取一个随机数
    BMC11.selectChipID_ST25TB512_ISO14443B(Random,Random1);
    // 把获取的随机数发给卡
   /***** SRT512 卡 UID 读取 *****/
    nlens = BMC11.getUID_ST25TB512_ISO14443B(uid_buf); // 获取 UID
    Serial.write(uid_buf, nlens);  // 获取到的 UID 显示在串口监视器上
    Serial.println(" ");
   /**** SRT512 卡数据读写 ****/
    BMC11.writeBlock_ST25TB512_ISO14443B(0x07,0x11852369);  // 写数据
    nlens = BMC11.readBlock_ST25TB512_ISO14443B(0x07,data); // 读数据
    Serial.write(data,nlens);  // 获取到的数据显示在串口监视器上
    Serial.println(" ");
    Serial.println(" ");
    delay(2000);
}
```

3. 打开串口监视器，波特率选择 115200；串口监视器显示如下：



范例 3：ISO15693_Write_Read



实物连接示意图

范例实现功能：ISO15693 卡 UID 读取以及数据读写，并在串口监视器上显示。

1. 范例打开方式：Arduino IDE → 文件 → 示例 → Lib 选择 (BMC11T001) → 选择范例 (ISO15693_Write_Read)

2. 示例说明：

a. 创建对象 & 模块初始化及配置

```
#include "BMC11T001.h"
BMC11T001     BMC11(&Serial4);  // 创建对象，使用 Serial4 接口
int nlens;
uint8_t uid_buf[23] = {0};
uint8_t data[18] = {0};
void setup()
{
  delay(1000);     // 等待扩充板上电初始化
  BMC11.begin(115200);      // 扩充板初始化
  Serial.begin(115200);     // 配置串口监视器
  BMC11.begin_ISO15693();  // 初始化 ISO15693 协议
}
```

b. 获取 ISO15693 卡的 UID 并在串口监视器上显示，对卡的数据先进行写然后读取，获取的数据在串口监视器上显示。

```
void loop()
{
   /***** ISO15693 卡 UID 读取 *****/
    nlens = BMC11.getSingleUID_ISO15693(uid_buf);  // 获取 UID
    Serial.write(uid_buf, nlens);  // 获取到的 UID 显示在串口监视器上
    Serial.println(" ");
   /****ISO15693 卡数据读写 ****/
   if(!BMC11.writeBlock_ISO15693(0x0A,0x22245678,uid_buf)) // 写数据
   {
    nlens = BMC11.readBlock_ISO15693(0x0A,uid_buf,data);  // 读数据
    Serial.write(data, nlens);  // 获取到的数据显示在串口监视器上
    Serial.println(" ");
   }
  Serial.println(" ");
  delay(2000);
}
```

3. 打开串口监视器，波特率选择 115200；串口监视器显示如下：