

IM920

通信モジュール、IM920sLを使用するためのクラス.

enum IM920Bytes

- Bytes8 : 8バイトの送受信のために使う
- Bytes16 : 16バイトの送受信のために使う
- Bytes24 : 24バイトの送受信のために使う
- Bytes32 : 32バイトの送受信のために使う

class IM920

IM920::init(Pin tx_pin,Pin rx_pin,UartNumber uart_num)

初期化関数

[パラメータ]

送信ピン番号

受信ピン番号

シリアル番号

[戻り値]

なし

[サンプルコード]

オブジェクトを作成し、初期化する.

```
#include "stm32f4xx.h"
#include "stm32f4xx_nucleo.h"
#include "sken_library/include.h"

IM920 im920;

int main(void){
    sken_system.init();

    im920.init(A0,A1,SERIAL4);

    while(true){
    }
}
```

IM920::read(uint16_t* rx_id,uint8_t* rx_data,IM920Bytes bytes)

受信関数

[パラメータ]

受信先のノード番号を格納するための変数のアドレス

受信データを格納するための配列のアドレス

受信データの大きさ

[戻り値]

なし

[サンプルコード]

受信データを 8 バイトで取得

```
#include "stm32f4xx.h"
#include "stm32f4xx_nucleo.h"
#include "sken_library/include.h"

IM920 im920;
uint16_t RxID;
uint8_t RxData[8];

int main(void){
    sken_system.init();

    im920.init(A0,A1,SERIAL4);

    while(true){
        im920.read(&RxID,RxData,Bytes8);
    }
}
```

IM920::write(uint8_t* tx_data,IM920Bytes bytes,int node_num = 0x0000)

送信関数

[パラメータ]

送信データ配列のアドレス

送信データの大きさ

送信先のノード番号（省略すると全てのノードに送信される）

[戻り値]

なし

[サンプルコード]

8バイトのデータを送信

```
#include "stm32f4xx.h"
#include "stm32f4xx_nucleo.h"
#include "sken_library/include.h"

IM920 im920;
uint8_t TxData[8] = {0x91,0xA2,0xB3,0xC4,0xD5,0xE6,0xF7,0x08};

int main(void){
    sken_system.init();
```

```
im920.init(A0,A1,SERIAL4);  
im920.write(TxData,Bytes8);  
  
while(true){  
}  
}
```