

SkenMdd

モータドライバドライバ（MDD）を使用するためのクラス

enum MddCommandId

- MOTOR_RPS_COMMAND_MODE : 目標回転数設定
- MOTOR_PWM_COMMAND_MODE : PWM設定
- MECANUM_MODE : メカナム動作モード
- OMNI3_MODE : 3輪オムニ動作モード
- OMNI4_MODE : 4輪オムニ動作モード
- M1_PID_GAIN_CONFIG : モータ1PIDゲイン設定
- M2_PID_GAIN_CONFIG : モータ2PIDゲイン設定
- M3_PID_GAIN_CONFIG : モータ3PIDゲイン設定
- M4_PID_GAIN_CONFIG : モータ4PIDゲイン設定
- ROBOT_DIAMETER_CONFIG : 車輪直径と旋回直径設定
- PID_RESET_COMMAND : PID制御リセット指令
- MOTOR_COMMAND_MODE_SELECT : PWM指令モードか回転数指令モードかの選択
- ENCODER_RESOLUTION_CONFIG : エンコーダ分解能設定

enum MddStdId

- None
- MDD_0
- MDD_1
- MDD_2
- MDD_3
- MDD_4
- MDD_5
- MDD_6
- MDD_7

class SkenMdd

void SkenMdd::init(Pin tx_pin,Pin rx_pin,UartNumber uart_num)

Uart通信を行う際の初期化関数

[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]

オブジェクトを生成し、初期化する

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

SkenMdd mdd;

int main(void)
{
    sken_system.init();
    mdd.init(B10,C5,SERIAL3);
    while (1) {

    }
}
```

void init(MddStdid stdid,Pin can_tx,Pin can_rx,CanSelect can_select)

CAN通信を行う際の初期化関数

[パラメータ]

MDDスタンダードID

送信ピン番号

受信ピン番号

CAN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

```
""C++
#include "stm32f4xx.h"
#include "sken_library/include.h"
#include "sken_mdd.h"

SkenMdd mdd;

int main(void)
{
    sken_system.init();
    mdd.init(MDD_0,A12,A11,CAN_1);
    while (1) {
```

```
}
```

```
}
```

```
## bool SkenMdd::tcp(uint8_t id,const float (&command_data)[4],unsigned int  
resend_time,unsigned int max_wait_time)
```

TCPモードで送信する

パラメータ設定など確実に通信する必要があるデータに対して使用する

[パラメータ]

指令ID

指令データ配列

再送信時間 [ms]

最大待機時間 [ms]

[戻り値]

成功か失敗

[サンプルコード]

モータ1のPIDのゲインを再送時間10 [ms]、最大待機時間2000 [ms]で設定する

```
```` C++  
#include "stm32f4xx.h"
#include "sken_library/include.h"
#include "sken_mdd.h"

const float MOTOR_PID_GAIN[4] = {1.5,0.1,0.1,20};

SkenMdd mdd;

int main(void)
{
 sken_system.init();
 mdd.init(B10,C5,SERIAL3);
 mdd.tcp(M1_PID_GAIN_CONFIG,MOTOR_PID_GAIN,10,2000);
 while (1) {

 }
}
```

## void SkenMdd::udp(uint8\_t id,const float (&command\_data)[4])

UDPモードで送信する

回転数指令など何度も送るため1度届かなくても大きな影響のないデータに対して使用する

[パラメータ]

指令ID

指令データ配列

[戻り値]

なし

[サンプルコード]

4つのモータの目標回転数を設定する.

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

```
#include "sken_mdd.h"

const float MOTOR_PID_GAIN[4] = {1.5,0.1,0.1,20};
float motor_target_rps[4] = {1,-0.5,1.5,-10};

skenMdd mdd;

int main(void)
{
 sken_system.init();
 mdd.init(B10,C5,SERIAL3);
 mdd.tcp(M1_PID_GAIN_CONFIG,MOTOR_PID_GAIN,10,2000);
 while (1) {
 mdd.udp(MOTOR_RPS_COMMAND_MODE,motor_target_rps);
 }
}
```