

gpio

GPIOピンの入出力をあつかう

enum Mode

入出力の設定

```
OUTPUT
INPUT
INPUT_PULLUP
PWM_OUTPUT
```

enum PinState

入出力状態

```
LOW
HIGH
```

class Gpio

void Gpio::init(Pin pin, Mode pin_mode)

ピンの入出力設定

デジタル出力とデジタル入力の設定で使用する.

[パラメータ]

ピン番号

入出力の設定

[戻り値]

なし

[サンプルコード]

A5ピンをデジタル出力に設定する

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

Gpio gpio;

int main(void)
{
    sken_system.init();
    gpio.init(A5, OUTPUT);
    while(1)
    {

    }
}
```

void Gpio::init(Pin pin, Mode pin_mode, TimerNumber tim_num, TimerChannel tim_ch, int frequency = 16800)

ピンの入出力設定

PWM出力の設定で使用する.

[パラメータ]

ピン番号

入出力の設定

タイマー番号

タイマーチャンネル

PWM周波数

[戻り値]

なし

[サンプルコード]

A5ピンをPWM出力に設定する

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

Gpio gpio;

int main(void)
{
    sken_system.init();
    gpio.init(A5, PWM_OUTPUT, TIMER2, CH1);
    while(1)
    {

    }
}
```

void Gpio::write(PinState state)

出力に設定している場合, ピンからHIGHまたはLOWを出力することができる

[パラメータ]

出力状態

[戻り値]

なし

[サンプルコード]

A5ピンをデジタル出力に設定し, HIGHを出力する

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

Gpio gpio;

int main(void)
{
```

```

sken_system.init();
gpio.init(A5,OUTPUT);
while(1)
{
    gpio.write(HIGH);
}
}

```

void Gpio::write(double percentage)

PWM出力に設定されている場合矩形波が出力される。

ただし、同じタイマーにエンコーダやRCサーボ信号を使用している場合はたとえチャンネルが違ったとしても正しい出力が行われない。

[パラメータ]

出力[%]

[戻り値]

なし

[サンプルコード]

A5ピンから50%のPWMを出力する

```

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

Gpio gpio;

int main(void)
{
    sken_system.init();
    gpio.init(A5,PWM_OUTPUT,TIMER2,CH1);
    while(1)
    {
        gpio.write(50);
    }
}

```

bool Gpio::read(void)

入力に設定している場合、ピンの値を読み取ることができる

[パラメータ]

なし

[戻り値]

HIGHのときtrue, LOWのときfalse

[サンプルコード]

C13ピンに接続したスイッチを読む

```

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

Gpio gpio;

int main(void)

```

```
{
    sken_system.init();
    gpio.init(C13, INPUT_PULLUP);
    while(1)
    {
        if(!gpio.read()){

        }
    }
}
```