ダウンロード・データあります

組み込みRustのライブラリ 便利クレート探偵団

第13回 組み込みRustの共通インターフェースembedded-halでPWM，I²C，SPIを使ってみよう

中林 智之

表1　SetDutyCycleトレイトのメソッド

メソッド	説　明
max_duty_cycle()	最大デューティ比を取得できる．取得できる値はデューティ比100%の値
set_duty_cycle()	デューティ比をduty/max_dutyに設定する
set_duty_cycle_fully_off()	デューティ比を0%，つまり常にOFFに設定する
set_duty_cycle_fully_on()	デューティ比を100%，つまり常にONに設定する
set_duty_cycle_fraction()	デューティ比を分数で指定して設定する
set_duty_cycle_percent()	デューティ比を%で指定して設定する

Rustは組み込みで使えるプログラミング言語として注目されています．本連載ではそんなRustの組み込み開発で役立つライブラリ（クレート）を紹介します．

前回はembedded-hal v1.0.0のdigital（GPIO）とdelayで定義されているトレイトの具体的な使い方，および，embedded-hal v1.0.0のエラーの設計について説明しました．今回はペリフェラル・デバイスの制御で頻繁に使用するPWM，I²C，SPIで定義されているトレイトの使い方を解説します．

PWMでデューティ比を設定できるSetDutyCycleトレイト

PWM（Pulse Width Modulation：パルス幅変調）は，LEDの明るさ制御やモータの速度制御など，組み込み開発では非常によく使われる出力方式です．

ある周期の中で信号を“H”にしている割合（デューティ比，デューティ・サイクル）を変化させることで，負荷に供給する平均電力を制御します．

例えば，周期を一定にしてデューティ比を50%にすると，信号が周期の半分だけ“H”になります．LEDの明るさを滑らかに変えたいときは，デューティ比を少しずつ変化させることで実現できます．

リスト1　SetDutyCycleトレイトの定義
[inline]が付いているものはデフォルト実装が提供されている

```
pub trait SetDutyCycle: ErrorType {
    fn max_duty_cycle(&self) -> u16;
    fn set_duty_cycle(&mut self, duty: u16)
        -> Result<(), Self::Error>;
    #[inline]
    fn set_duty_cycle_fully_off(&mut self)
        -> Result<(), Self::Error>;
    #[inline]
    fn set_duty_cycle_fully_on(&mut self)
        -> Result<(), Self::Error>;
    #[inline]
    fn set_duty_cycle_fraction(&mut self, num: u16,
            denom: u16) -> Result<(), Self::Error>;
    #[inline]
    fn set_duty_cycle_percent(&mut self,
            percent: u8) -> Result<(), Self::Error>;
}
```

● トレイト

embedded-hal v1.0.0では，PWMのデューティ比を設定するためのSetDutyCycleトレイトが用意されています．SetDutyCycleでは1つのチャネル（ピン）に対してデューティ比を設定します．

表1の6つのメソッドが定義されています．SetDutyCycleトレイトの定義を**リスト1**に示します．

● PWMを制御するプログラム

リスト2のプログラムでは，nRF52840（ノルディック・セミコンダクター）のPWMを使ってLED（P0.13）を徐々に明るくします．

nrf-halではPwmChannel構造体がSetDutyCycleを実装しています．main()関数ではGPIOピンとnRF52840のPWMを設定して，PwmChannel構造体のインスタンスを初期化しています．この部分は，HAL実装ごとに初期化の仕方が異なります．今回はGPIO P0.13を出力に設定し，500HzでPWM出力する設定にしています．

fade_in()関数は徐々にLEDを明るくする関数です．こちらの関数はembedded-halのトレイトのみを使っているため，nrf-hal以外でも使用できます．10msのループごとに，デューティ比を最大までイン

ダウンロードはこちらから．
https://github.com/tomoyuki-nakabayashi/interface-embedded-rust-crates/tree/main