

Utils

Reset

TODO

Pamięć EEPROM

Popularnym sposobem przechowywania zmiennych, których wartości zostają zachowane po resecie urządzenia lub utracie zasilania, jest pamięć **EEPROM**. Najprostszą metodą jest użycie domyślnego kontrolera tej pamięci do zapisu oraz wczytywania zmiennych podczas inicjalizacji programu lub w momencie, kiedy będą one potrzebne.

W przykładzie wykorzystywane są 4 [wejścia cyfrowe DI](#). Operujemy na zmiennej `value`, której wartość jest inicjowana (*przed pętlą główną `while`*), przez odczyt z pamięci EEPROM. Jeśli odczyt się nie powiedzie (*zmienna nie zostanie znaleziona*), zostanie przypisana wartość domyślna. W pętli głównej program oczekuje na wciśnięcie jednego z przycisków podłączonych do wejść cyfrowych oraz realizuje ich obsługę:

- **DI1** - Inkrementacja zmiennej `value` (zwiększenie o 1)
- **DI2** - Zapisanie wartości zmiennej `value` do pamięci EEPROM
- **DI3** - Reset sterownika
- **DI4** - Wyczyszczenie pamięci EEPROM

```
// Wartość domyślna zmiennej `value`, używana, gdy nie ma jej w pamięci EEPROM
#define DEFAULT_VALUE 1
// Zmienna robocza, której wartość będzie przechowywana w pamięci EEPROM
uint32_t value;

void loop(void)
{
    if(EEPROM_Load(&value)) {
        // Obsługa, gdy zmienna nigdy nie była zapisana w pamięci EEPROM
        // Może być wykorzystana do wczytania wartości domyślnej
        value = DEFAULT_VALUE;
        DBG_String("EEPROM default value:");
    }
    else {
        // Wartość z pamięci EEPROM została znaleziona i przekopiiowana do zmiennej `value`
        DBG_String("EEPROM load value:");
    }
    DBG_uDec(value);
    DBG_Enter();

    while(1) {
        // Pojawienie się sygnału na wejściu cyfrowym DI1
        if(DIN_Rise(&DI1)) {
            value++; // Zwiększenie wartości zmiennej `value` o 1
            DBG_String("EEPROM inc value:");
        }
    }
}
```

```

        DBG_uDec(value);
        DBG_Enter();
    }
    // Pojawienie się sygnału na wejściu cyfrowym DI2
    if(DIN_Rise(&DI2)) {
        EEPROM_Save(&value); // Zapisz aktualną wartość zmiennej `value` w pamięci
EEPROM
        DBG_String("EEPROM save");
        DBG_Enter();
    }
    // Pojawienie się sygnału na wejściu cyfrowym DI3
    if(DIN_Rise(&DI3)) {
        DBG_String("PWR Reset");
        DBG_Enter();
        DBG_BeforeReset(); // Poczekaaj, aż wszystkie dane z DBG zostaną wysłane
        PWR_Reset(); // Zresetuj sterownik - program rozpocznie się od nowa
    }
    if(DIN_Rise(&DI4)) {
        EEPROM_Remove(); // Reset wszystkich zmiennych z pamięci EEPROM
        DBG_String("EEPROM remove");
        DBG_Enter();
        value = DEFAULT_VALUE; // Ustawienie wartości na domyślną
    }
    let();
}
}

```

 Kompletny przykład: [Podstawowa obsługa EEPROM](#)

Ale jest kilka haczyków. Zmienna, którą chcemy przechowywać musi zajmować 4 bajty, więc nadają się `uint32_t`, `float`, `int32_t`. Gdy typ jest różny od `uint32_t` należy dokonać rzutowania na niego. Gdy zmienna jest zajmuje mniej bajtów (`char`, `uint8_t`, `int8_t`, `uint16_t`, `int16_t`) należy zrobić wyjątek i zmienić jej deklarację na `uint32_t` lub `int32_t`, w zależności czy korzystamy zmiennej z znakiem czy bez

Zmienna, którą zapisujemy musi być zadeklarowana globalnie lub być zmienną statyczną `static`

W rzeczywistości na płytce nie ma dołączonej pamięci EEPROM

Plik konfiguracyjny FLASH

Backup registers BKPR

Gdy mamy naprawdę niewiele zmiennych konfiguracyjnych, zamiast pliku konfiguracyjnego lub nawet kontrolera EEPROM, możemy wykorzystać 4 rejestry **BKPR** podtrzymywane bateryjnie. Ich zaletą jest niewątpliwie szybkość dostępu, jednak rozładowanie lub wyciągnięcie baterii będzie skutkować utratą tych wartości.