

I2C (TWI) – Pamięć EEPROM AT24C32

Na zajęciach będzie wykorzystywana magistrala I2C (TWI ang. Two Wire Interface) do komunikacji z modułem zawierającym EEPROM AT24C32 oraz zegar czasu rzeczywistego DS1307. Schemat modułów można znaleźć się m.in. na stronie <http://electropark.pl/rtc-zegary-czasu-rzeczywistego/3422-modul-rtc-ds1307-z-at24c128-i-ds18b20.html>

W modułach tych nie ma baterijek.

Przed zajęciami proszę o:

przeczytanie krótkiego opisu najpierw po polsku

<http://pl.wikipedia.org/wiki/I%C2%B2C>

i ewentualne przejrzanie bardziej szczegółowych informacji po angielsku

<https://en.wikipedia.org/wiki/I%C2%B2C> (nie ma potrzeby analizowania tabelki).

Należy znać działanie funkcji `sprintf` opisaną na <http://www.cplusplus.com/reference/cstdio/sprintf/>

Na ćwiczenia będzie programowana komunikacja z pamięcią EEPROM AT24C32 o pojemności 4kB (32kb). Proszę przeczytać str. 1. i 8-12 noty katalogowej.

<https://ww1.microchip.com/downloads/en/DeviceDoc/doc0336.pdf>

Szczególnie uważnie proszę przeanalizować treść na str. 9-12.

DEVICE ADDRESS oznacza adres urządzenia I2C. Na podstawie analizy schematu modułu łatwo można zauważyć, że wyprowadzenia A1, A2 i A3 układu AT24C32 są zwarte do masy. Z tego powodu bity A1, A2, A3 przedstawione na rysunku 1. ze str. 11 mają wartość zero.

FIRST WORD ADDRESS i SECOND WORD ADDRESS oznaczają adres w pamięci EEPROM, od którego dane będą zapisywane lub odczytywane.

Do komunikacji z AT24C32 można wykorzystać następujące funkcje:

```
HAL_I2C_Master_Transmit(I2C_HandleTypeDef *hi2c, uint16_t DevAddress, uint8_t *pData,
uint16_t Size, uint32_t Timeout);
```

```
HAL_I2C_Master_Receive(I2C_HandleTypeDef *hi2c, uint16_t DevAddress, uint8_t *pData,
uint16_t Size, uint32_t Timeout);
```

```
HAL_I2C_Mem_Write(I2C_HandleTypeDef *hi2c, uint16_t DevAddress, uint16_t MemAddress,
uint16_t MemAddSize, uint8_t *pData, uint16_t Size, uint32_t Timeout);
```

```
HAL_I2C_Mem_Read(I2C_HandleTypeDef *hi2c, uint16_t DevAddress, uint16_t MemAddress,
uint16_t MemAddSize, uint8_t *pData, uint16_t Size, uint32_t Timeout);
```

(Funkcje te zwracają `HAL_StatusTypeDef`).

Przed przystąpieniem do ćwiczeń proszę zapoznać się z ich działaniem.

Zadania:

1. Znaleźć wyprowadzenia mikrokontrolera SDA i SCL. Następnie podłączyć nucleo z modułami (**proszę nie zapomnieć połączyć ze sobą mas makiety i modułu**).

1.) W pamięci EEPROM od adresu 1000 należy zapisać kolejne cyfry numerów indeksów wszystkich osób z grupy. Następnie należy je odczytać i wysłać przez UART do komputera.

2.) Napisać program, który będzie zliczał naciśnięcia niebieskiego przycisku na płytce nucleo. Zliczona liczba powinna być zapisywana w EEPROM (po wyłączeniu i powtórny włączeniu zasilania odliczanie powinno być kontynuowane od zapisanej wartości) Liczba naciśnięć powinna być wysyłana przez UART.

Na ocenę bardzo dobrą oraz celującą należy dodatkowo wykorzystać zegar czasu rzeczywistego DS1307:

Proszę przeczytać str. 1 i 8-13 noty katalogowej zegara czasu rzeczywistego DS1307 (RTC ang. Real-Time Clock)

<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf> Szczególnie uważnie proszę obejrzeć tabelę 2, przeczytać pierwszy akapit na str. 10 oraz przeanalizować rysunki 4-6. Word Address oznacza adres rejestrów DS1307 (pierwsza kolumna tab.2).

DS1307 będzie urządzeniem slave.

Dodatkowe zadanie na ocenę bardzo dobrą oraz celującą:

3. Należy napisać program, który najpierw ustawi przybliżoną godzinę w DS1307 zgodnie z rys. 4 (proszę wybrać tryb 24h), a następnie w nieskończonej pętli będzie odczytywać czas z DS1307 (według rys. 6) i przesyłać go przez UART.