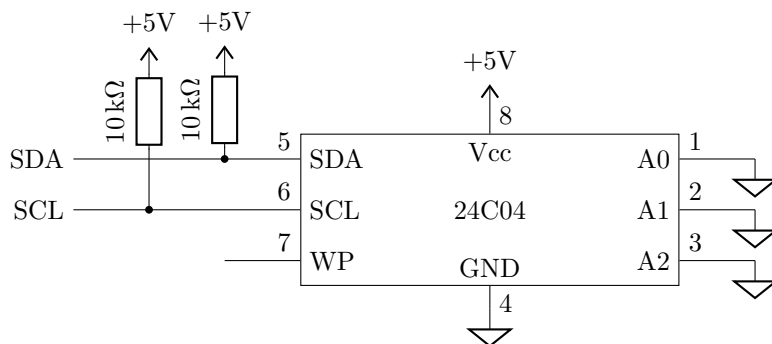


# Systemy wbudowane

## Lista zadań nr 7

2 i 5 grudnia 2019

1. Podłącz układ EEPROM I<sup>2</sup>C 24C04<sup>1</sup> z zestawu do pinów TWI zgodnie z poniższym schematem:



Zaimplementuj program umożliwiający zapisywanie i odczytywanie danych wewnątrz układu. Możesz wykorzystać przykładowe procedury obsługi I<sup>2</sup>C z książki „Make: AVR Programming” (pliki `i2c.h` i `i2c.c` na SKOS). Komunikacja z programem ma się odbywać przez UART, program powinien obsługiwać następujące polecenia:

- `read addr` – odczyt bajtu z adresu `addr`,
- `write addr value` – zapis pod adres `addr` wartości `value`.

2. Rozszerz program z poprzedniego zadania o następujące polecenia:

- `read addr length` – odczyt `length` bajtów od adresu `addr`, dane należy wypisać w formacie I8HEX<sup>2</sup>,
- `write` – zapis danych, kolejne wiersze powinny zawierać dane w formacie I8HEX.

Należy zminimalizować liczbę transakcji I<sup>2</sup>C, wykonując sekwencyjne odczyty i zapisy w ramach jednej transakcji.

3. Podłącz moduł zegara czasu rzeczywistego I<sup>2</sup>C DS3231<sup>3</sup> z zestawu do pinów TWI, łącząc: wejście + z linią +5V, wejście - z linią GND, wejście D z linią SDA, wejście C z linią SCL. Podłącz też rezystory pull-up o wartości 10 kΩ dla linii SDA i SCL.

Napisz program umożliwiający ustawianie i odczytywanie daty i czasu zapisanych w układzie DS3231. Program powinien obsługiwać następujące polecenia:

- `date` – odczyt daty (w formacie DD-MM-YYYY),
- `time` – odczyt czasu (w formacie HH:MM:SS),
- `set date DD-MM-YYYY` – ustawienie daty,
- `set time HH:MM:SS` – ustawienie czasu.

Moduł zegara posiada podtrzymanie baterijne, zapisany czas i data powinny być zachowane po odłączeniu kabla USB.

4. Sygnały magistrali SPI mogą być generowane programowo, przez „ręczne” sterowanie pinami GPIO (tzw. *bit-banging*). Połącz piny MISO, MOSI, SCK, SS z wybranymi czterema innymi pinami GPIO mikrokontrolera (np. PD4...PD7). Skonfiguruj wbudowany sprzętowy sterownik SPI jako slave. Napisz program, który będzie pracował jako master SPI przez *bit-banging* i komunikował się z sprzętowym sterownikiem SPI, np. wymieniając kolejne liczby 1, 2 itd. Za pomocą UART informuj o przebiegu komunikacji.

<sup>1</sup><https://datasheet.octopart.com/24C04WP-STMicroelectronics-datasheet-21183101.pdf>

<sup>2</sup>[http://en.wikipedia.org/wiki/Intel\\_HEX#Named\\_formats](http://en.wikipedia.org/wiki/Intel_HEX#Named_formats)

<sup>3</sup><https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>