

Lista zadań nr 5

1. Benchmarking. Zbadaj czas działania następujących operacji (czas można mierzyć np. licznikiem 16-bitowym TC1):
 - o Operacji arytmetycznych (dodawania, mnożenia, dzielenia) dla liczb całkowitych o różnych rozmiarach (int8_t, int16_t, int32_t) oraz liczb zmiennoprzecinkowych (float, double).
 - o Operacji matematycznych dla liczb zmiennoprzecinkowych (sin, log, exp).
 - o Zmiany stanu pinów za pomocą Arduino API i bezpośredniej manipulacji rejestrami.

Czas operacji należy podać w cyklach zegara i w mikrosekundach, należy odjąć czas poświęcony na manipulowanie licznikiem. [3]

2. Zaimplementuj grę typu zgadnij liczbę od 0-99. Wprowadzanie danych powinno się odbywać za pomocą impulsatora (inaczej enkodera, jest w zestawie), wybór ma być zatwierdzany przyciskiem. Informację o tym, czy szukana liczba jest większa, czy mniejsza od wybranej ma być pokazana za pomocą diod LED. Obecnie wybrana liczba ma być wyświetlona na wyświetlaczu 8-segmentowym przy pomocy układu drivera SCT2168. [3] Ładowanie danych do układu można zrealizować programowo lub przy pomocy sprzętowego SPI. [1] Dodatkowe punkty za wersję używającą dwóch Arduino, komunikujących się przez SPI: pierwsze ma pracować jako interfejs dla użytkownika jako SPI master, drugie jako slave ma losować liczbę do zgadnięcia, odbierać zapytania z wybraną liczbą od mastera i odpowiadać (również po SPI) w jakiej relacji jest wybrana liczba od liczby wylosowanej. [2]
3. Zbuduj przedwzmacniacz dla mikrofonu elektretowego zgodnie z załączonym schematem (wszystkie elementy potrzebne do konstrukcji są w zestawie). Zaprogramuj detektor częstotliwości (np. dźwięku A4 - 440 Hz) używając [algorytmu Goertzela](#). [3] Częścią poprawnego rozwiązania jest obliczenie offsetu DC (wartości napięcia która pojawia się na wyjściu wzmacniacza, gdy mikrofon nie odbiera żadnego dźwięku) przez uśrednienie wartości wejścia. Polecam użyć częstotliwości próbkowania ok. 8 kHz. Aby częstotliwość próbkowania była precyzyjna, można użyć wyzwalania ADC przez licznik (bity ADTSn rejestru ADCSRB). [1] Dla efektywności warto unikać niepotrzebnych obliczeń (np. funkcję sinus można stabilizować, można też unikać obliczeń zmiennoprzecinkowych używając reprezentacji stałoprzecinkowej). [2]
4. Korzystając z noty katalogowej układu [24C01](#) napisz programator pamięci EEPROM korzystających z szyny I²C. Uwaga: komunikację z układem pamięci należy napisać samemu z użyciem biblioteki [Wire](#). Twój program ma komunikować się z komputerem poprzez łącze szeregowo i obsługiwać następujące polecenia:
 - o read addr: gdzie addr to adres czytanego bajtu,
 - o read addr length: gdzie addr to pierwszy adres czytanego bloku, a length to jego długość; odczytane dane należy wypisać w formacie [I8HEX](#) (należy wygenerować sumę kontrolną).
 - o write addr value: gdzie addr to adres pisanego bajtu o wartości value,

- write: którego następnymi liniami będą zapis tekstowy formatu [I8HEX](#) (należy sprawdzać sumę kontrolną).

Zauważ, że istnieją dwa tryby pracy EEPROMu - bajtowy i stronicowy - wykorzystaj tą właściwość. **[4]**

5. Skonstruuj choinkę z kolorowymi diodami (zielony kartonik z wpiętymi diodami), której lampki będą świecić w różny sposób a dołączony głośnik/brzęczyk będzie grać kolędę lub inną świąteczną piosenkę. **[3]**
6. **[Drukarka 3d]** Zapoznaj się z formatami danych używanymi w druku 3D (STL, G-code) oraz metodami "plasterkowania" modelu 3d (np. program Slic3r). Zrealizuj sterowanie lusterkami dla wyświetlania przekroju obiektu zadanego przez wejście G-code. **[4]**



[preamp.png](#)