

Systemy wbudowane

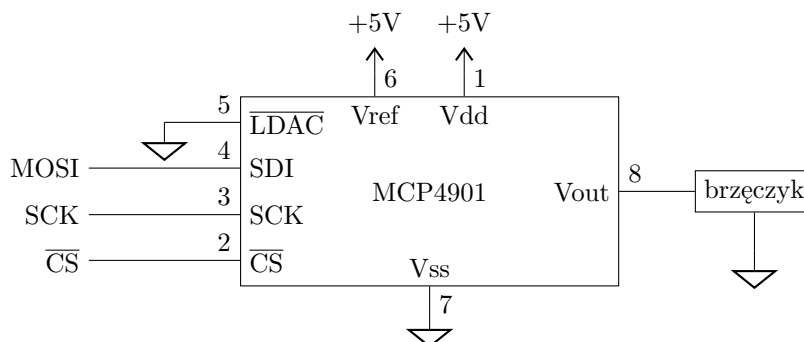
Lista zadań nr 6

17, 19 i 25 listopada 2020

1. Napisz jak najprostszy program, który będzie wykonywał za pomocą UART funkcję „echo”: każdy odebrany przez układ znak powinien zostać przesłany z powrotem do nadawcy. Nie używaj funkcji z biblioteki standardowej języka C (`printf`, `scanf`, `getchar` itp.). Wykorzystaj przerwania, pętla główna programu powinna wyłącznie usypiać mikrokontroler do stanu bezczynności (głębsze tryby uśpienia wyłączają układ UART).
2. Przykładowa implementacja wejścia/wyjścia przez UART z pliku `helloworld.c` wykorzystuje aktywne czekanie, co ma pewne wady. Podczas nadawania i odbierania procesor większość czasu spędza na sprawdzaniu bitu statusu, a odbieranie działa poprawnie tylko wtedy, gdy program wykonuje funkcje odczytywania. Popraw te wady, implementując rozwiązanie oparte na przerwaniach, wykorzystujące dwa bufony cykliczne: nadawania i odbierania.

Napisz procedury obsługi przerwań `USART_RX`, `USART_TX` i `USART_UDRE` oraz nowe implementacje funkcji `uart_transmit` i `uart_receive` unikające aktywnego czekania. Funkcja nadająca może czekać tylko wtedy, jeśli bufor nadawania jest pełny, zaś funkcja odbierająca tylko wtedy, gdy bufor odbierania jest pusty. Pamiętaj o prawidłowej synchronizacji funkcji z procedurami obsługi przerwań przy użyciu maskowania. Przetestuj swoje rozwiązanie.

3. Zbuduj następujący układ wykorzystujący układ DAC MCP4901. Zwróć uwagę na prawidłowe podłączenie układu: nóżka numer 1 jest oznaczona kropką, kolejne nóżki są liczone przeciwnie do kierunku wskazówek zegara. Nieprawidłowe podłączenie linii zasilających **uszkodzi** układ.



Zapisz w swoim programie krótki, jednokanałowy plik audio w formie ciągu próbek, używając niskiej częstotliwości próbkowania (np. 8 kHz); ze względu na rozmiar dane mogą być umieszczone wyłącznie w pamięci flash mikrokontrolera. Odtwórz ten plik używając układu DAC sterowanego transceiverem SPI.

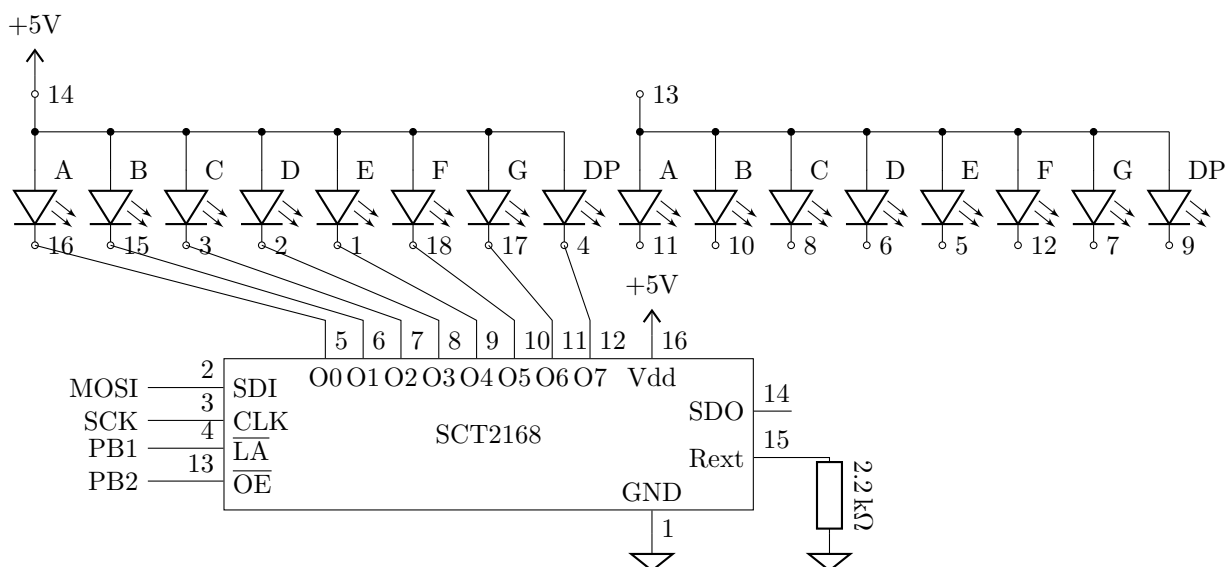
Wgranie nowej próbki do układu MCP4901 polega na: zmianie stanu pinu \overline{CS} na niski, przesłanie dwóch bajtów przy użyciu SPI (w trybie CPOL=0 CPHA=0) zgodnie z opisem w nocie katalogowej¹, zmianie stanu pinu \overline{CS} na wysoki.

4. Zbuduj następujący układ wykorzystujący wyświetlacz 7-segmentowy oraz układ sterownika LED SCT2168. Wyświetl na wyświetlaczu co sekundę kolejne cyfry od 0 do 9. Wykorzystaj transceiver SPI do sterowania układem sterownika LED. Sposób sterowania tym układem jest następujący:
- Stan wysoki pinu \overline{LA} oznacza załadowanie nowego stanu diod LED z rejestru przesuwne, zaś w stanie niskim rejestr przesuwany nie wpływa na diody LED.
 - Stan wysoki pinu \overline{OE} wyłącza świecenie diod LED.

¹<http://ww1.microchip.com/downloads/en/DeviceDoc/22248a.pdf>

Szczegóły są opisane w nocie katalogowej².

Zwróć uwagę na prawidłowe podłączenie układu. Nieprawidłowe podłączenie linii zasilających **uszkodzi** układ.



²http://www.starchips.com.tw/pdf/datasheet/SCT2168V01_01.pdf