

Celem czwartego zadania laboratoryjnego było zaprojektowanie, złożenie, zaprogramowanie i przetestowanie układu z mikrokontrolerem MSP430F16x tak aby umożliwiał on komunikację z komputerem PC za pomocą portu RS-232. Celem naszego projektu była korekcja linii tekstu zakończonego znakiem CR, która polegała na usuwaniu powtórzonych znaków i zbędnych spacji oraz konwersji wielkich liter.

\\\\Przykładowe wejście – wyjście programu:

```
\begin{center}
```

„ALA ma Kooota , 1123” -> „Ala ma kota, 123”

```
\end{center}
```

Wybrane zadanie pochodziło z puli zadań dodatkowych.

\\\\ UWAGI

\\ Podzielenie kodu na warstwę aplikacyjną i warstwę sprzętową. Gdyby pojawiła się taka potrzeba, proszę sformułować dodatkowe założenia zmieniające funkcjonalność z uwzględnieniem specyfiki transmitancji szeregowej. W programie należy zaimplementować i używać bufory cykliczne. Nadawanie i odbieranie danych należy realizować z największą dostępną szybkością i z użyciem przerwań. Funkcje systemu mogą zostać wzbogacone o dodatkową interakcję, należy zadbać o przemyślany sposób ich uruchamiania.

\\\\ OPIS SPRZĘTU

Dzięki strukturze mikrokontrolera MSP430f16x podłączenie urządzeń peryferyjnych nie było skomplikowanym zadaniem, jednak wymagało uwagi, aby odpowiednio połączyć ze sobą porty na PC (Tx i Rx) i mikrokontrolerze odpowiadające za komunikację.

% OBRAZEK %

Do portów P3.4 – Tx, P3.5 – Rx w mikrokontrolerze podłączyliśmy sygnały Rx i Tx z linii SV2 pochodzącej z modułu łącza szeregowego RS232. Moduł ten łączy się z komputerem za pomocą złącza DB9M.

\\\\ OPIS OPROGRAMOWANIA

\\ Do zmniejszenia poboru energii użyliśmy trybu LPM0. Kod programu napisaliśmy w języku C, ale w porównaniu do zadań z poprzednich laboratoriów warstwa programowa była zdecydowanie bardziej rozbudowana. Użycie języka C zamiast

assemblera pozwoliło na lepsze uporządkowanie funkcjonalności i zadań do wykonania w naszym programie. Kod stał się bardziej przejrzysty i zrozumiały.

#### \\\\ Transmisja danych

\\ Transmisję obsługujemy za pomocą przerwań generowanych przez USART w trybie UART. W przerwaniu od USART0\_TX dokonujemy kopiowania z bufora danych wejściowych txBuf do bufora TXBUF0. Zapisywanie danych do TXBUF0 inicjuje transmisję. Odbiór danych odbywa się w przerwaniu od USART0\_RX. Kiedy dane zostaną odebrane są przypisywane do UxRXBUF. Odbieranie danych z konsoli musi być zakończone znakiem nowej linii. Odebrany bufor jest poddawany obróbce w warstwie aplikacyjnej, w której to następuje m. in. zmiana dużych liter na małe lub usunięcie powtarzających się liter. Tak zmodyfikowany bufor pojawia się na wyświetlaczu.