

## Laboratorium 5.

### Magistrala szeregową RS232.

Ćwiczenie ma na celu zapoznanie użytkownika z magistralą szeregową RS232 oraz metodami jej obsługi w mikrokontrolerach AVR. Zadanie polega na napisaniu programów wykorzystujących wbudowane urządzenie UART do odbioru i transmisji komend sterujących i sprawdzenie ich działania wykorzystując komunikację poprzez RS232 z komputerem PC.

#### Część 1:

Napisać program, który będzie odbierał pojedyncze znaki wysyłane za pomocą magistrali RS232 z komputera PC. Jeśli odebrany znakiem będzie 'x' należy odesłać ciąg znaków „ok\n”.

#### Część 2:

Rozbudować program o obsługę komend wielo-znakowych „on” - włącz i „off” - wyłącz i za pomocą tych komend sterować włączaniem i wyłączaniem diody na płycie uruchomieniowej. W tym celu należy przygotować bufor odbieranych znaków. Dodatkowo jednocześnie z obsługą RS232 powinna zostać wykorzystana obsługa przycisku – podobnie jak w poprzednim ćwiczeniu, odczyt stanu naciśnięcia przycisku w przerwaniu licznika co 100ms. Przycisk SW0 należy podłączyć do linii portu PC1, a diodę LD1 do linii portu PC0. Zapalanie diody powinno odbywać się w pętli głównej w funkcji main na podstawie stanu zmiennej globalnej 'led\_status'. W rozwiązaniu zadania należy uwzględnić obsługę sytuacji błędnych typu niepełna przesłana instrukcja i kolejne znaki następnej. Można to zrealizować za pomocą czyszczenia bufora nadesłanych znaków po określonym czasie (w celu sprawdzenia wstępnie można posłużyć się dłuższym czasem np. 1s a następnie zejść do bezpiecznego minimum). Dekodowanie komendy może być zrealizowane poprzez odebranie określonej ilości znaków albo na podstawie wystąpienia znaku kończącego komendę np. '\n' lub innego.

#### Część 3:

Przygotować kalkulator RPN (Reverse Polish Notation) w ramach transmisji RS232. Użytkownik z podłączonego terminala wpisuje liczbę i zatwierdza znakiem nowej linii ('\n'). Mikrokontroler odczytuje liczbę i zapisuje do zmiennej y (przed tym zmienną y przepisuje do zmiennej x). Użytkownik wpisuje drugą liczbę i zatwierdza znakiem nowej linii ('\n'). Mikrokontroler przepisuje zmienną y do zmiennej x i zapisuje odczytaną z RS232 liczbę do zmiennej y. Użytkownik wpisuje działanie („+”, „-”, „\*”, „/”) i zatwierdza znakiem nowej linii ('\n'). Mikrokontroler oblicza wartość:  $x <\text{działanie}> y$  i wynik zapisuje do zmiennej y oraz przesyła w postaci tekstowej poprzez RS232. Należy obsłużyć sytuacje wyjątkowe ( np. przez wysłanie komunikatu o błędzie) typu dzielenie przez 0, niewłaściwa komenda itp.

#### UWAGA:

Port szeregowy w systemie GNU/Linux dostępne są jako pliki specjalne. W przypadku standardowych portów wbudowanych w płytę główną komputera (lub kartę rozszerzeń) są to /dev/ttyS0, /dev/ttyS1, /dev/ttyS2 (pod systemem MS Windows byłyby to odpowiednio COM1, COM2, COM3). W przypadku portów szeregowych podłączonych poprzez USB (za pomocą konwerterów) są to /dev/ttyUSB0, /dev/ttyUSB1..., ewentualnie /dev/ttyACM0, /dev/ttyACM1...

**W przypadku podłączenia płytki EVBAVR do komputera PC w Laboratorium Elektroniki, wykorzystywany jest port: /dev/ttyS2. Normalna praktyka podłączenia to nadajnik – odbiornik (TX – RX), jednakże na płycie EVBAVR należy podłączyć wyprowadzenie oznaczone RX z RX mikrokontrolera i odpowiednio wyprowadzenie TX z TX mikrokontrolera.**

Jako oprogramowanie komunikacyjne można wykorzystać oprogramowanie:

**cutecom** – polecany ze względu na osobne pola do wprowadzania instrukcji oraz prostą konfigurację, można także stosować terminalowe: **gtkterm** (trzeba skonfigurować port w menu), **minicom** (w terminalu uruchomić najpierw minicom -s , potem w ustawieniach portu szeregowego wpisać /dev/ttyS2), **screen** (poprzez komendę: screen /dev/ttyS2 115200 , wyjście przez Ctrl+a k)