Systemy Mikroprocesorowe

Dokumentacja Projektu

Opracowanie komunikacji UART za pomocą konwertera FT232 i mikroprocesora ATMEGA328PB

Projekt wykonali:
Leonid Kulakov
Jakub Sędłak

Prowadzący: **dr inż. Jacek Ostrowski**

Możliwości programu

Program realizuje prostą, interaktywną obsługę terminala znakowego przez UART. Oferuje następujące funkcje:

- Wysyłanie wiadomości z pamięci programu Flash (ROM) za pomocą subrutyny send_str (lub send_char do wysyłania pojedynczego znaku) .
- Odbiór danych przesłanych od np. laptopa i buforowanie odebranych znaków w pamięci ulotnej RAM za pomocą subrutyny recieve_str (lub pojedynczego znaku za pomocą recieve_char).
- Odczyt otrzymanej wiadomości z pamięci ulotnej i odsyłanie jej do nadawcy za pomocą subrutyny echo.

Opis działania programu

Inicjalizacja i konfiguracja

Konfiguracja UARTa polega na ustawieniu:

- *baudrate* (9600)
- BPS $\left(\frac{f_{CPU}}{8*baudrate} - 1\right)$
    - UCSR0B tak aby włączyć odbieranie i wysyłanie danych
      ldi r16 , (1<<RXEN0) | (1<<TXEN0)
      sts UCSR0B,r16

Rezerwacja pamięci

- Pamięć nieulotna

  W pamięci nieulotnej zostały zainicjalizowane znak zachęty i przejście do nowej linii

- Pamięć ulotna

  W pamięci ulotnej jest zainicjalizowany buffer X, w którym są zapisywane i odczytywane wiadomości otrzymywane poprzez komunikację UART. Służy on do ponownego odczytu otrzymanych wiadomości w celu wysłania ich echa.

Połączenia elektroniczne

Aby umożliwić komunikację, należy podłączyć pin PD0 (Rx0) mikrokontrolera Atmega328PB do pinu Tx konwertera FT232, a pin PD1 (Tx0) do pinu Rx.

Main Loop programu

Główna pętla programu ma na celu zautomatyzowanie wywołań subrutyn tworząca interfejs w terminalu np. PuTTy (COM6) oraz ułatwienie w implementacji wysyłania gotowych wiadomości z pamięci nieulotnej.

main_loop:

rcall send_cursor

rcall recieve_str

rcall send_newline

jmp main_loop

Opis Subrutyn

- Odbieranie ciągu znaków wprowadzonych przez użytkownika do momentu naciśnięcia Enter (CR).
-
- Echo (zwrotne wysłanie) odebranego tekstu na terminal po naciśnięciu Enter.
- Przejście do nowej linii po każdym echo.
- Obsługa pojedynczego znaku (tryb testowy, niewykorzystywany w pętli głównej).
- Podstawowe opóźnienie programowe (delay).

    - Wyświetlanie znaku zachęty ('>') jako kursora.
    - Przejście do nowej linii za pomocą ciągu znaków (0x0A,0x0D)
    - Wyświetlenie dowolnej wcześniej zaprogramowanej wiadomości np. 'TEST' czy 'msg1'

Grupa subrutyn send_str

send_char

Wysyła pojedynczy znak zapisany w rejestrze r18 przez UART. Czeka na gotowość bufora nadawczego (bit UDRE0)..

Grupa subrutyn send_str

Wczytuje do rejestru z pamięci Flash (ROM) kolejne znaki z pamięci nieulotnej po poprzednim wpisaniu wiadomości do pointera Z, do momentu napotkania znaku null ($00). Po wczytaniu znaku do rejestru zapisywany jest on do UDR0, w celu jego przesłania. Taka procedura umożliwia przesłanie wiadomości w całości.

receive_char

Odbiera pojedynczy znak przez UART (z UDR0) i wywołuje send_char w celu natychmiastowego jego odesłania .

Grupa subrutyn receive_str

Odbiera cały ciąg znaków z UART do momentu odebrania znaku CR (Enter). Każdy znak zapisuje w buforze X w pamięci ulotnej RAM. Po odebraniu CR wywoływana jest subrutyna echo w celu przesłania echa otrzymanej wiadomości.

Grupa subrutyn echo

Ustawia wskaźnik X na początek bufora pamięci ulotnej X i rozpoczyna procedurę echa Wczytuje do rejestru z pamięci ulotnej kolejne znaki do momentu napotkania znaku CR. Po wczytaniu znaku do rejestru zapisywany jest on do UDR0, w celu jego przesłania. Taka procedura umożliwia odesłanie otrzymanej wiadomości w całości.

send_cursor

Zapisuje do pointera Z znak ">",$00z pamięci nieulotnej i wywołuje subrutyne send_str w celu wyświetlenia kursora na terminalu.

send_newline

Zapisuje do pointera Z 0x0A,0x0D,$00 z pamięci nieulotnej i wywołuje subrutyne send_str w celu wyświetlenia przejścia do nowej linii (LF + CR) na terminalu.

delay

Wykonuje krótkie, programowe opóźnienie poprzez pętle dekrementacyjne.