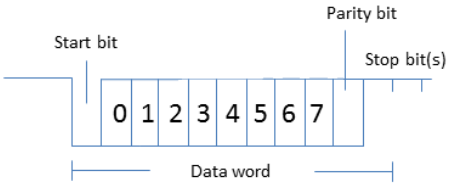
Analiza protokołów komunikacyjnych na przykładzie USART

Większość inżynierów systemów wbudowanych zna interfejs UART: uniwersalny asynchroniczny odbiornik/nadajnik. Jest to urządzenie peryferyjne mikrokontrolera, które konwertuje przychodzące i wychodzące bajty danych na strumień szeregowy. Bit początkowy rozpoczyna szeregowy strumień, a bit lub dwa stopu kończy słowo danych (ramkę). UART ma również możliwość dodania bitu parzystości do ramki, aby pomóc w wykryciu, jeżeli podczas transmisji wystąpił błąd. **Rysunek 1** pokazuje standardowy przykład ramki, jaką można zobaczyć oglądając dane przesyłane przez UART.



Rys.1 Standardowa ramka UART

Z kolei USART jest to uniwersalny synchroniczny/asynchroniczny odbiornik/nadajnik. Jest to urządzenie peryferyjne mikrokontrolera, które przekształca przychodzące i wychodzące bajty danych w szeregowy strumień bitów... Definicja USART jest identyczna z definicją UART, ale w skrócie pojawia się dodatkowe słowo - "synchroniczny". Muszą zatem istnieć jakieś różnice, w przeciwnym razie USART byłby tym samym co UART i nikt nikogo nie musiałby poprawiać.

Pierwszą różnicą między USART a UART jest sposób, w jaki dane szeregowe mogą być taktowane. UART generuje swój własny zegar danych wewnętrznie, w mikrokontrolerze i synchronizuje go ze strumieniem danych wykrywając początkowe przejście - bit startu. Nie ma tutaj przychodzącego sygnału zegarowego związanego z danymi, więc aby prawidłowo odbierać strumień takich danych, odbiornik musi wiedzieć z wyprzedzeniem, jaka powinna być prędkość transmisji.

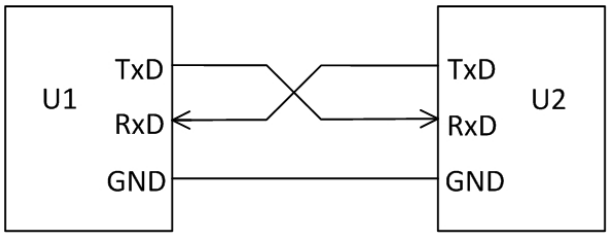
Z kolei USART, można skonfigurować do pracy w trybie synchronicznym. W tym trybie urządzenie peryferyjne wysyła sygnał zegarowy na zewnątrz, a odbierające urządzenie peryferyjne może odzyskać go ze strumienia danych, nie znając przy tym wcześniej szybkości tej transmisji. Alternatywnie, niektóre łącza używać mogą wręcz osobnej linii zegarowej do taktowania całej transmisji. Użycie zewnętrznego zegara pozwala na znacznie większą szybkość transmisji danych z wykorzystaniem interfejsu USART niż standardowego UART - tego rodzaju przesyły osiągają prędkość do 4 Mbps. Drugą istotną różnicą pomiędzy interfejsami USART a UART jest liczba protokołów obsługiwanych przez urządzenie peryferyjne. UART jest prosty i oferuje tylko kilka opcji podstawowego formatu danych. Konfigurować można liczbę bitów stopu oraz występowanie i sposób obliczania bitu parzystości ramki. USART jest dużo bardziej złożony i może generować dane w formie odpowiadającej wielu różnym protokołom, takim jak IrDA, LIN, Smart Card oraz Driver Enable dla interfejsów takich jak RS-485 czy Modbus. Lista ta oczywiście nie wyczerpuje pełnych możliwości interfejsu USART. Dodatkowo, ma również takie same

możliwości realizowania transmisji asynchronicznej jak UART, to znaczy USART może generować ten sam typ danych szeregowych, jak pokazano na **rysunku 1**.

RS232 / UART

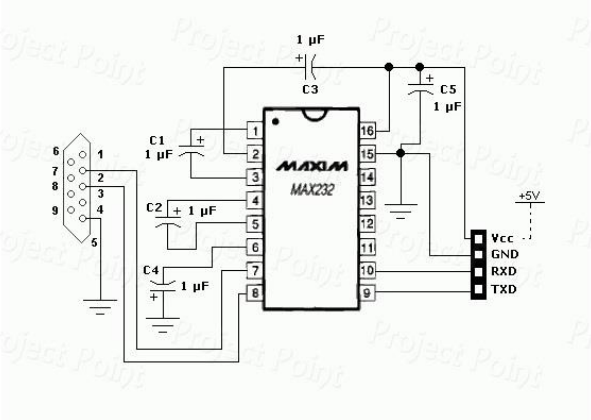
Jest to jeden z najstarszych interfejsów komunikacyjnych opracowany przez firmę Gordon Bell. Znalazł on zastosowanie w urządzeniach przemysłowych, medycznych oraz wojskowych. Popularność RS232 spadła po pojawieniu się interfejsu USB, lecz ze względu na jego niezawodność ciągle jest stosowany, nawet w nowoczesnych układach. Warto zaznaczyć, że w specyfikacjach różnych urządzeń można spotkać inną nazwę interfejsu RS232, a mianowicie jest to UART (Universal Asynchronous Receiver and Transmitter). RS232 i UART są to te same interfejsy a różnica w ich działaniu polega na pracy z różnymi poziomami napięć (od -15 V do 15 V dla RS232 i od 0 V do 5V dla UART w technologii bramek logicznych TTL).

Do realizacji komunikacji za pomocą interfejsu UART wystarczą dwie linie sygnałowe, gdzie jedna linia to TxD, odpowiedzialna za nadawanie oraz druga RxD, odpowiedzialna za odbieranie. Linie te muszą być skrzyżowane co oznacza, że wejście RxD pierwszego urządzenie połączone jest z wyjściem TxD drugiego urządzenia, analogicznie wyjście TxD pierwszego urządzenia jest połączone z wejściem RxD urządzenia drugiego. Sposób połączenia został przedstawiony na rys. 2.



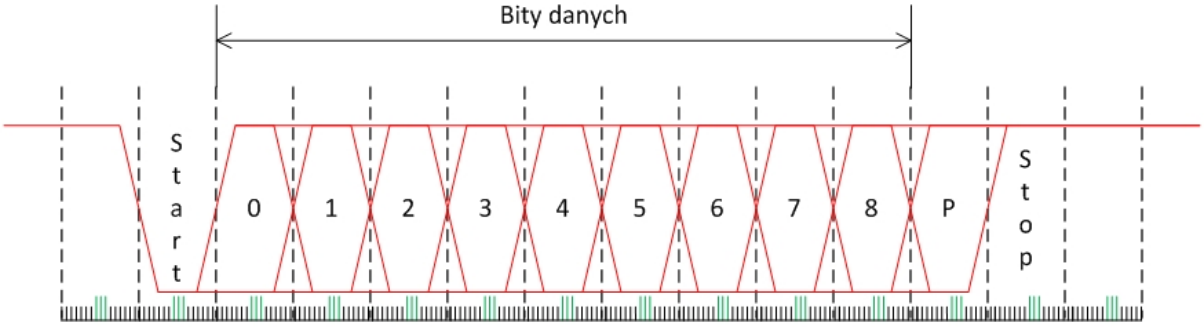
Rys.2 Połączenie dwóch urządzeń w standardzie RS232 / UART / USART

Jeśli do komunikacji z mikrokontrolerem chcemy wykorzystać standardowy interfejs RS-232, to należy dokonać konwersji poziomów napięć do poziomu akceptowalnego przez mikrokontroler. Można do tego celu zastosować popularny układ MAX232. Na rys.3 został przedstawiony przykładowy schemat konwertera.



Rys.3 Konwerter poziomów napięć standardu RS-232/UART

Transmisja interfejsu UART jest asynchroniczna, dlatego każdy wysłany znak poprzedzony jest bitem startu (logiczne zero). Następnie w ramce przesyłane są dane, ramka może pomieścić od 5 do 9 bitów. Po bitach danych opcjonalnie możliwe jest ustawienie bitu parzystości. Na koniec transmisji wysyłany jest bit stopu (logiczna jedynka), o wartości 1, 1,5 lub 2 bitów. Wewnętrzne układy odbiornika próbkują sygnał z częstotliwością 16 razy szybszą niż ustawiane są bity. Odczytywanie bitów polega na zebraniu próbki 8, 9 i 10 a następnie porównanie ich ze sobą. W przypadku kiedy dwa kolejne bity posiadają tą samą wartość, wówczas jest ona wpisywana do rejestru przesuwnego. W przypadku, kiedy bit po 9 będzie posiadał inny stan niż bit 8 i 10, wówczas oznacza to, że wystąpił błąd komunikacji. Prędkości transmisji nie zostały ujęte w specyfikacji interfejsu, dlatego należy ją ustawić własnoręcznie. Typowe wartości prędkości transmisji jakie aktualnie się przyjmuje to: 2400, 4800, 9600, 19200, 38400, 57600, 115200 bodów/bitów na sekundę. W celu poprawnej komunikacji, prędkość transmisji nadajnika i odbiornika musi być taka sama. Ramkę interfejsu UART została przedstawiona na rys.4.



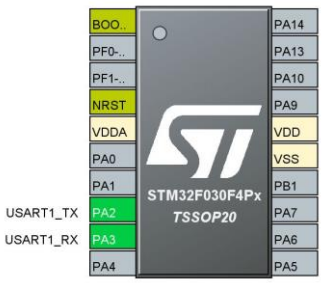
Rys. 4 Ramka danych w protokole UART

Ćwiczenie 1.

Należy zaprojektować system komunikacji USART z wykorzystaniem układów STM32 w taki sposób, aby było możliwe bezpośrednie przesyłanie komunikatów z jednego układu do drugiego.

Z uwagi na zgodność napięć po obydwu stronach łącza, nie zachodzi potrzeba dokonywania konwersji poziomów napięć stanów logicznych, jedynie wymagane jest odpowiednie połączenie linii nadawczej TX i odbiorczej RD obydwu układów.

Komunikacja odbywa się z wykorzystaniem pinów PA2 i PA3.



Rys. 5 Łącze komunikacyjne USART mikrokontrolera STM32F030F4

Aby była możliwe komunikacja między jednostkami, połączenia powinny zostać wykonane jak zostało to przedstawione na rys. 2.

Do zaprojektowanego układu sprzętowego, należy napisać program sterujący, który będzie wysyłał danej z jednego mikrokontrolera do drugiego.

Należy przedstawić rozwiązania w oparciu o następujące podejścia:

- Tryb odpytywania
- Tryb z wykorzystaniem przerwań
- Tryb z wykorzystaniem DMA

Ćwiczenie 2.

Należy napisać program dla nadajnika, który zawiera trzy komendy sterujące:

- włącz
- wyłącz
- generuj przerywany sygnał

Następnie należy napisać program dla klienta, który na podstawie odebranych i zdekodowanych komend, wykonuje następujące działania:

- dla komendy „włącz”, zostaje włączona wbudowana dioda LED na płytce prototypowej,
- dla komendy „wyłącz”, zostaje wyłączona wbudowana dioda LED na płytce prototypowej,
- Dla komendy „generuj …”, wbudowana dioda LED na płytce prototypowej przechodzi w tryb migania z ustaloną częstotliwością

Ćwiczenie 3.

Dla przedstawionych w ćwiczeniach rozwiązaniach, należy sprawdzić ich bezpieczeństwo w zakresie poprawności komunikacji w zależności od czynników zakłócających (słabej jakości połączenie kablowe, pasożytnicze pojemności i indukcyjności, i inne). Innym kryterium do zbadania, jest weryfikacja maksymalnej dopuszczalnej prędkości komunikacji, która zapewnia bezbłędną transmisję danych, w zależności od długości przewodów łączących.

Ćwiczenie 3.

Należy przygotować serwis, którego zadaniem będzie obliczanie sumy przesłanej tablicy bajtów, a uzyskany wynik powinien być odsyłany do klienta.