

## **1. Cel Ćwiczenia**







Celem ćwiczenia jest:

- zapoznanie się z budowa wirtualnych przyrządów pomiarowych, wykorzystujących złącze RS232C oraz środowiskiem LabVIEW;
- praktyczna nauka pisania prostych aplikacji w LabVIEW, wykorzystujących aplikacje VISA do obsługi portów oraz nabycie umiejętności konfigurowania i obsługiwanania portu szeregowego RS232C.

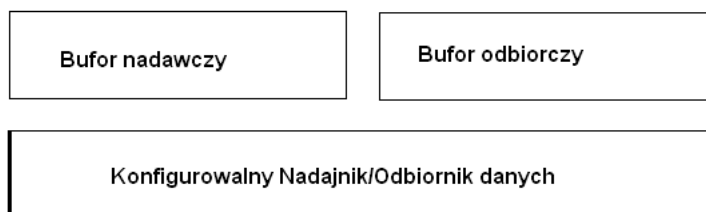
## **2. Obsługa portów szeregowych**

Prawie każde urządzenie oparte na mikroprocesorze bądź mikrokontrolerze ma wśród swoich zasobów układy asynchronicznej transmisji danych (UART). Fakt ten sprawia, że jest on najpowszechniej stosowanym medium komunikacji ze światem zewnętrznym. Po przetworzeniu sygnałów do poziomów określonych standardem RS232, możemy podłączyć urządzenie wprost do portu COM naszego komputera, jest to tzw. strona sprzętowa. Strona programowa jest trochę bardziej skomplikowana. Z punktu widzenia mikrokontrolera oprogramowanie układu UART nie przysparza wielu problemów, o tyle strona PC jest już bardziej niedostępna. Istnieją aplikacje terminalowe zdolne odbierać i nadawać dane poprzez porty szeregowo, jednak pozostawiają pewien niedosyt, bo często konstruktor chciałby mieć aplikacje do obsługi dopiero co zaprojektowanego urządzenia. Z Lab View nie ma z tym większego problemu. Poniżej przedstawiona jest tabela 1 zawierająca zestawienie sześciu komponentów pozwalających na kompletną obsługę portów szeregowych.

Tabela 1. Zawartość subpalety serial [2]

	<i>VISA Configure Serial Port</i> – dokonuje inicjalizacji portu szeregowego wskazanego przez zmienną typu <i>VISA Resource Name</i> (o wartości np. COM1, COM2). Funkcja ta dokonuje rezerwacji zasobu, która trwa aż do wywołania funkcji <i>VISA Close</i>
	<i>VISA Write</i> – funkcja zapisu do portu szeregowego. Dane podawane na terminal wejściowy <i>write buffer</i> mają postać łańcucha znakowego. Funkcja zwraca liczbę bajtów, które udało się zapisać
	<i>VISA Read</i> – funkcja odczytu bufora portu szeregowego. Jako argument przyjmuje liczbę bajtów, która ma zostać odczytana. W rezultacie działania zwraca dane w postaci łańcucha znakowego (zwraca też jego długość)
	<i>VISA Close</i> – zwalnia zasób zarezerwowany przez <i>VISA Configure Serial Port</i> (zamyka port szeregowy)
	<i>VISA Bytes At Serial Port</i> – komponent zwraca liczbę bajtów oczekujących w buforze odbiorczym portu szeregowego
	<i>VISA Serial Break</i> – przerywa transmisję wybranego portu szeregowego na 250 ms

Wysokopoziomowa architektura portu COM. Komponenty dostępne z poziomu Lab View operują na pokazanych poniżej blokach, rysunek 1, które wchodzi w skład każdego portu szeregowego zainstalowanego w komputerze.



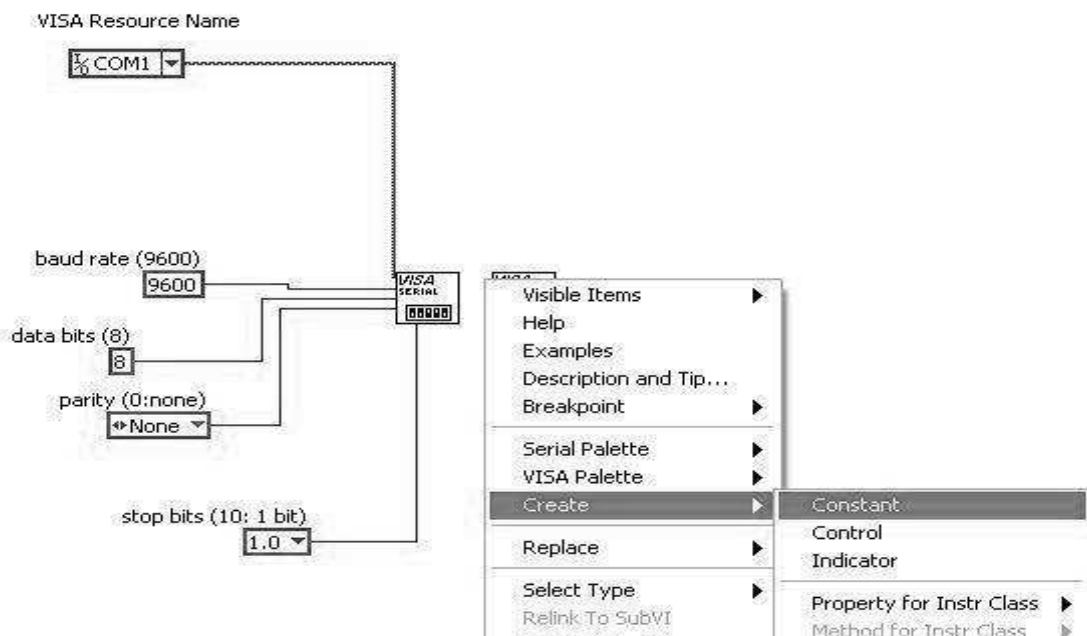
Rys. 1. Ogólna architektura portu szeregowego [2]

Komponent *VISA Configure Serial Port* dokonuje konfiguracji portu szeregowego. Operuje bezpośrednio na nadajniku/odbiorniku danych. Inicjalizacja portu szeregowego polega na określeniu parametrów jego pracy. Oprócz standardowych takich jak prędkość transmisji czy format danych, umożliwia zdefiniowanie wielu dodatkowych własności transmisji. Interfejs VI ma aż 10 terminali wejściowych. Najważniejszym z nich jest *VISA Resource Name*, przyjmuje referencję do portu szeregowego COM, który ma zostać zainicjalizowany. Kliknijmy w oknie block diagram na funkcje, przez wyszukanie(serach) znajdziemy blok *VISA Serial*, rysunek 2.



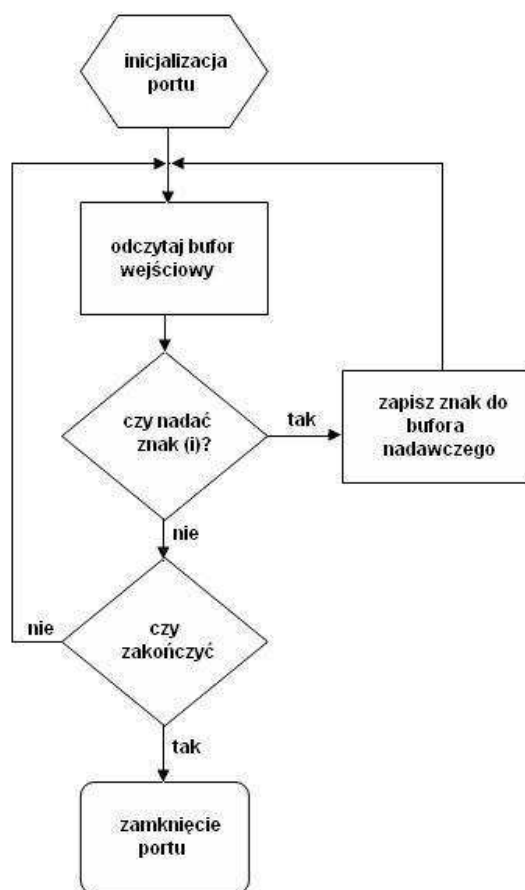
Rys. 2. Wyszukiwanie funkcji VISA serial

Klikając na terminal prawym przyciskiem myszy, używając narzędzia Connect Wire (szpulka) i stwórzmy stałą (Create>Constant). Tak utworzony obiekt zawiera w rozwijanej liście wszystkie porty I/O znajdujące się w systemie, przedstawione jest to na rysunku 3. Kolejne trzy terminale wejściowe są odpowiedzialne za podstawowe parametry transmisji szeregowej : baud rate – prędkość bodowa ; data bits – liczba bitów danych ; parity – parzystość. Na dole symbolu umieszczono terminal decydujący o liczbie bitów stopu – stop bits. W lewym dolnym rogu znajdziemy terminal error in. Ten zestaw parametrów jest potrzebny do nawiązania minimalnej wersji połączenia szeregowego. VISA Configure Serial Port ma dużo szersze możliwości konfiguracyjne np. kontrole przepływu .[2]



Rys. 3. Podstawowe parametry transmisji szeregowej

Na górnej krawędzi VISA Configure Serial Port znajduje się terminal wejściowy oznaczony jako timeout. Parametr ten określa maksymalny czas na wykonanie operacji zapisu lub odczytu bajtów. Jeśli do końca zadanego czasu w buforze nie znajdzie się wymagana liczba bajtów, wówczas komponent zgłosi błąd i zakończy działanie. Ostatnie dwa terminale są ze sobą powiązane. Pierwszy Enable Termination Char, przyjmuje wartości typu Boolean, która decyduje o tym, czy podczas odbioru danych będzie rozpoznawany znak terminacji, który podłączony jest do terminalu wejściowego termination char. Jeśli funkcja napotka w łańcuchu bajtów znak terminacji, wówczas zakończy działanie, zwracając odebrane do tej pory znaki. Komponenty zapisu i odczytu to VISA Write i Read . VISA Write jako parametr przyjmuje referencje do zainicjalizowania portu COM oraz łańcuch znaków do wysłania. Należy pamiętać, że samo wywołanie funkcji nie oznacza, iż bajty wysłano. Zostały one jedynie wpisane do bufora nadawczego i nie mamy żadnej pewności co do momentu w którym nastąpi transmisja. Ewentualne opóźnienia są znikome i często pomijalne. VISA Read oprócz standardowych parametrów przyjmuje liczbę bajtów, które mają zostać odebrane ( terminal byte count). Nie jest rzeczą oczywistą oszacowanie jaka to powinna być wartość. Z pomocą przychodzi inny komponent z palety Serial : Bytes At Serial Port. Jest to węzeł własności, jako parametr przyjmuje referencje do zainicjalizowania portu, a w rezultacie działania zwraca liczbę bajtów, która obecnie znajduje się w buforze odbiorczym. Uruchamiając ten komponent tuż przed wywołaniem VISA Rad, mamy pewność, że odczytamy wszystkie do tej pory odebrane dane oraz że nie będziemy starali się odczytywać znaków które jeszcze nie nadeszły. Komponent VISA Close zamyka port szeregowy, co skutkuje przywróceniem jego dostępności dla innych aplikacji Na rysunku 4 przedstawiony jest algorytm podstawowej aplikacji, która w bezpieczny sposób korzysta z komponentów. Działania na grafie wykonują się sekwencyjnie, co może sugerować, że algorytm nie będzie mógł jednocześnie nadawać i odbierać danych. W rzeczywistości jest jednak inaczej, gdyż z poziomu Lab View operujemy jedynie na buforach nadawczym i odbiorczym. Za transmisję full duplex odpowiedzialny będzie właściwy nadajnik/odbiornik danych. Dostęp do buforów rzeczywiście nie odbywa się w sposób równoległy. [2]



Rys. 4. Algorytm obsługi portu szeregowego

Kod aplikacji. Funkcjonalność, jaką program będzie oferował obejmuje ciągły odbiór danych oraz wysyłanie danych na żądanie użytkownika. W tym celu program będzie oparty na pojedynczej pętli, do detekcji akcji użytkownika wykorzystamy strukturę Event. Dane przesyłane łączem szeregowym umieszczane są w wejściowej pamięci buforowej. W przypadku przepełnienia tej pamięci komputer ignoruje nowe dane do chwili, gdy dane z pamięci buforowej zostaną odczytane. Przed utratą informacji zabezpiecza handshaking – odbiorca informuje nadawcę, że powinien przerwać wysyłanie danych do chwili, gdy zostaną odczytane dane przesłane wcześniej. W programie tym jest dostępny handshaking sprzętowy i programowy.