

Ćwiczenie 9

Automatyzacja pomiarów

9.1. Wiadomości ogólne o interfejsach szeregowych

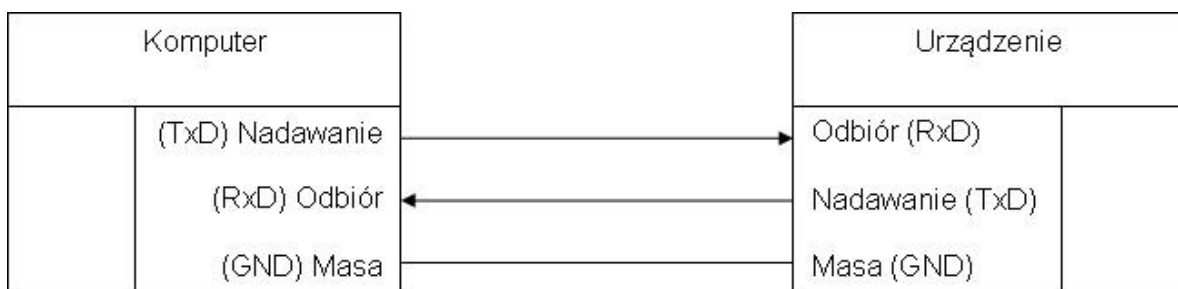
Ze względu na swoją prostotę oraz niewielkie wymagania sprzętowe (w porównaniu z interfejsami równoległymi), transmisja z wykorzystaniem łączy szeregowych jest szeroko stosowana w przemyśle elektronicznym. Popularnym standardem łączy szeregowych jest interfejs RS-232. Jego nazwa pochodzi od oznaczenia amerykańskiej normy branżowej, opracowanej przez Electronic Industries Alliance (EIA), zawartej w biuletynie „Recommended Standard” nr 232. Na jej bazie w roku 1972 powstały zalecenia międzynarodowe opisane w dokumentach V.24 oraz V.28 Międzynarodowego Komitetu Doradczego do spraw Telegrafii i Telefonii (CCITT).

Standard ten ma długą historię. W USA wprowadzony został w roku 1962 i był sześciokrotnie nowelizowany. Aktualnie obowiązująca wersja ma oznaczenie ANSI/TIA-232-F-1997 (został dodatkowo przyjęty przez Telecommunications Industry Association, TIA). Litera „F” w nazwie standardu wskazuje, że jest to jego siódma wersja.

Polska Norma opisująca to łącze ma oznaczenie PN-T-05052:1986.

9.2. Własności łączy szeregowych

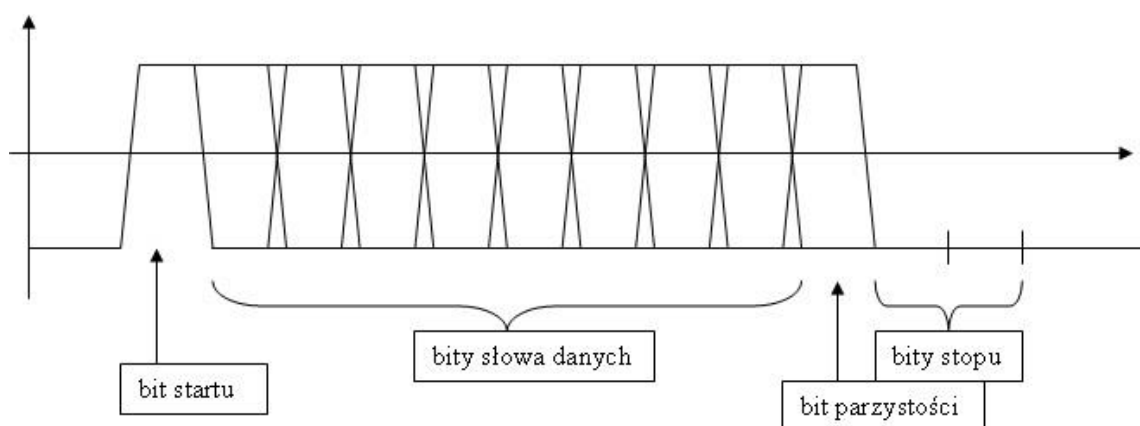
Połączenie komputera z przyrządem pomiarowym lub innym urządzeniem może być w tym standardzie w bardzo prostym przypadku zrealizowane następująco:



Każde z urządzeń przesyła bity danych zakodowane jako impulsy elektryczne, w których „0” odpowiada wysokiemu poziomowi napięcia, a „1” – niskiemu. Przesyłanie odbywa się w obu kierunkach z pomocą oddzielnych linii przy wykorzystaniu wspólnej linii masy. Użycie oddzielnych linii do przesyłania informacji w obu kierunkach umożliwia jednoczesną transmisję bez obawy o wystąpienie zakłóceń.

9.3. Format słowa danych

Przesłanie bitów słowa danych poprzedzone jest wysłaniem tzw. bitu startu (poziom napięcia wysoki), który sygnalizuje rozpoczęcie transmisji słowa. Następnie przesyłane są poszczególne bity w kolejności od najmniej znaczącego do najbardziej znaczącego. Po przesłaniu bitów słowa danych może być przesyłany bit kontroli parzystości, a następnie linia przechodzi w stan spoczynkowy (poziom napięcia niski) na czas trwania jednego lub dwóch bitów – są to tzw. bity stopu.



Dopuszczalne wartości napięcia dla odpowiednich stanów logicznych bitów są następujące:

Wartość bitu	Przy nadawaniu	Przy odbiorze
0	(+ 5 V ... + 15 V)	(+ 3 V ... + 15 V)
1	(- 5 V ... - 15 V)	(- 3 V ... - 15 V)

9.4. Parametry transmisji w łączy szeregowym

Norma RS-232 określa wiele szczegółów mechanicznych i elektrycznych odnoszących się do transmisji szeregowej. Aby urządzenie odbierające informację mogło określić, który bit transmitowanego słowa jest w danej chwili odbierany – unormowaniu musiał być poddany czas trwania pojedynczego bitu. Jeśli jest on równy np. 1/9600 s, to mówimy, że szybkość transmisji jest równa 9600 bitów na sekundę. Dopuszczalne przez normę szybkości transmisji to: 110, 150, 300, 600, 1200, 2400, 4800, 9600 oraz 19200 bitów na sekundę.

Aby transmisja odbywała się poprawnie, zarówno urządzenie nadające, jak i odbierające powinny być ustawione na tę samą szybkość przesyłania informacji. Należy w tym miejscu zaznaczyć, że wymienione powyżej szybkości transmisji odnoszą się do przesyłania bitów podczas transmisji pojedynczego słowa danych, nie zaś do transmisji jako całości.

Poszczególne słowa danych transmitowane są metodą start-stopową, odstępy czasu pomiędzy poszczególnymi słowami mogą być dowolnie duże i dlatego przeciętna szybkość całej transmisji z reguły bywa znacznie mniejsza. Słowo danych może zawierać 5, 6, 7 lub 8 bitów informacyjnych. W zależności od typu przesyłanej informacji można wybrać różną jego długość. W przypadku transmisji telegraficznych (dalekopis) wykorzystywane było słowo 5-bitowe; przy transmisji danych tekstowych – słowo 7-bitowe, a przy przesyłaniu danych binarnych – słowo 8-bitowe.

Bity stopu podczas transmisji stanowią swego rodzaju rezerwę czasową daną odbiorcy informacji, aby mógł w tym czasie odebrane słowo danych przesłać z rejestru odbiorczego do odpowiedniego buforu telekomunikacyjnego, mając zagwarantowane, że w tym czasie nadawca nie rozpocznie transmisji kolejnego słowa. W przypadku najniższej szybkości transmisji dopuszczalna jest wartość $1\frac{1}{2}$ bitu stopu.

W typowym łączy szeregowym w standardzie RS-232 poza liniami do transmisji danych (TxD oraz RxD) występują dodatkowe linie, którymi przesyłane są sygnały sterujące służące pierwotnie do komunikowania się komputera z modemem telefonicznym. W przypadku bezpośredniego połączenia komputera z przyrządem pomiarowym linie te można wykorzystać do sterowania przepływem danych między połączonymi urządzeniami. Sterowanie to polega na wstrzymywaniu transmisji od nadawcy danych, jeżeli urządzenie odbiorcze nie jest w stanie przetworzyć lub zapamiętać zbyt dużego strumienia danych, a także wznowieniu transmisji danych, kiedy urządzenie odbiorcze jest już gotowe na przyjęcie następnej ich porcji.

Sterowanie przepływem danych (ang. *flow control*) może być zrealizowane w sposób sprzętowy, przy wykorzystaniu linii sterujących portu szeregowego (ang. *hardware flagging*), lub też za pomocą przesyłania po liniach danych umówionych znaków sterujących wstrzymaniem (XOFF) lub wznowieniem (XON) transmisji strumienia danych. W tym drugim przypadku sterowanie transmisją nosi nazwę programowego (ang. *software flagging*) lub protokołu XON/XOFF.

Do realizacji transmisji szeregowej w standardzie RS-232 przez systemy mikroprocesorowe przygotowano specjalizowane układy sterowników takiej transmisji; powszechnie nazywane są one UART (ang. *Universal Asynchronous Receiver-Transmitter*). Są one produkowane w formie autonomicznych układów scalonych (np. MC 68681 firmy Motorola zastosowany w oscyloskopach Tektronix TDS-200) lub częściej stanowią element składowy chipsetu komputera, albo są funkcjonalnym składnikiem zintegrowanym w mikrokontrolerze.

9.5. Programy telekomunikacyjne

W celu ułatwienia wymiany danych między komputerami przy wykorzystaniu łączy szeregowych RS-232 przygotowane zostały specjalne programy komunikacyjne. Jednym z nich jest **HyperTerminal**, który wchodził w skład standardowego zestawu akcesoriów systemu MS Windows (do wersji Windows XP). Z jego pomocą możliwa jest wymiana informacji między komputerami lub komputerem i urządzeniem znajdującymi się blisko siebie – za pomocą specjalnego kabla dla łączy szeregowego (zwanego *null modem cable*) lub pomiędzy urządzeniami bardzo oddalonymi – za pomocą sieci telefonicznej. W tym drugim przypadku konieczne jest posiadanie odpowiedniego modemu telefonicznego.

Innym programem tego typu jest **Termite 3.1**, niewielki program opracowany przez firmę CompuPhase i dystrybuowany jako darmowy do użytku osobistego, ale także komercyjnego.

Aby wymiana informacji była możliwa, konieczne jest uzgodnienie ustawień parametrów transmisji obu komunikujących się ze sobą urządzeń, tzn. szybkości transmisji, liczby bitów danych w transmitowanym słowie, liczby bitów stopu, rodzaju zastosowanego sprawdzania parzystości, rodzaju sterowania przepływem.

9.6. Zestaw komend sterujących przyrządami pomiarowymi (SCPI)

Zdalne sterowanie przyrządem polega na przesyłaniu do niego odpowiednich komend sterujących i odbieraniu od przyrządu przesyłanej zwrotnie informacji o jego stanie, wynikach przeprowadzonych pomiarów itp. Aby ułatwić użytkownikom zdalne sterowanie przyrządami pomiarowymi, komendy do ich sterowania mają postać ciągów znaków alfanumerycznych (liter, cyfr, znaków pisarskich i znaków sterujących wydrukiem), które są z reguły wyrazami lub zwrotami języka angielskiego lub ich skrótami. Dzięki temu komendy te są łatwiejsze do zapamiętania i wygodniejsze w użyciu przez człowieka. Podczas transmisji z pomocą łączy szeregowego pojedyncze transmitowane słowo RS-232 zawiera informację o jednym znaku komendy (literze, cyfrze lub znaku sterującym), w związku z tym przesłanie komendy do przyrządu związane jest z reguły z transmisją od kilku do kilkunastu słów RS-232.

Jednolity sposób porozumiewania się z przyrządami (język komend przyrządów) pozwala istotnie ułatwić posługiwanie się nimi. W roku 1990 grupa największych producentów aparatury pomiarowej doszła do porozumienia i opracowała jeden wspólny zestaw komunikatów do zdalnego sterowania wszystkimi rodzajami przyrządów pomiarowych. Ten jednolity zestaw komunikatów sterujących został nazwany SCPI (ang. *Standard Commands for Programmable Instruments*).

Zastosowanie ujednoliconego języka SCPI jako standardu pozwala na osiągnięcie następujących korzyści:

1. Urządzeniami steruje się identycznie, jak innymi przyrządami tego samego typu, niezależnie od producenta. Np. wszystkie multimetry posiadają takie same polecenia dotyczące sterowania funkcją pomiarową, podzakresem, wyzwaniem itd. Oczywiście ich możliwości funkcjonalne mogą się różnić. Cecha ta jest nazywana kompatybilnością wertykalną.
2. Do programowania podobnych funkcji w przyrządach stosuje się takie same polecenia. Na przykład filtracja czy sposób sprzężenia wejścia przyrządu (stałoprądowe lub przemiennoprądowe) odbywa się przy użyciu tych samych poleceń niezależnie od rodzaju urządzenia (multimetr, częstotściomierz, oscyloskop). Nazywa się tę cechę kompatybilnością funkcjonalną.
3. Istnieje możliwość stosowania identycznych poleceń do wykonania pomiarów wybranych wielkości przez różne urządzenia stosujące nawet odmienne techniki pomiarowe. Przykładowo czas narastania zbocza sygnału można zmierzyć za pomocą oscyloskopu lub licznika, stosując identyczne polecenia. Uzyskuje się to za pomocą poleceń zorientowanych sygnałowo należących do specjalnego podsystemu **MEASURE**. Polecenia te są bardzo przyjazne dla użytkownika, ponieważ nie wymagają od niego znajomości szczegółów funkcjonowania przyrządu. Ta cecha jest określana jako kompatybilność horyzontalna.

W języku SCPI zdefiniowano standardowy zestaw rozkazów o otwartym charakterze, a więc możliwym do rozszerzenia. Upraszcza to lub wręcz eliminuje potrzebę modyfikacji oprogramowania systemu pomiarowego przy zmianie niektórych przyrządów na nieco inne modele.

Specyfikacja języka SCPI określa składnię i styl języka, zestaw rozkazów oraz format wymiany danych. Składnia i styl to definicje reguł tworzenia mnemoniki, struktury instrukcji, parametrów instrukcji, wyrażeń, informacji o stanie i sposobie zerowania urządzeń. Aby zapewnić wymienialność danych, zdefiniowano składnię, gramatykę, format danych i sposób opisu bloków danych. Pozwala to różnym przyrządom poprawnie wymieniać dane.

Język SCPI tworzy strukturę hierarchiczną drzewiastą, w której podobne funkcje programujące są zgrupowane w odrębnej gałęzi drzewa. Korzenie tych drzew komend noszą nazwy odpowiadających im podsystemów urządzenia pomiarowego (**SENSE**, **SOURCE**, **TRIGGER**, **CALIBRATION** itd.) i pod każdym z nich są zgrupowane polecenia programujące dany podsystem. Dalsze uszczegółowienie poleceń gwarantują węzły niższego poziomu grupujące funkcjonalnie podobne funkcje programujące. Dla przykładu **SOURCE** jest słowem kluczowym korzenia polecenia programującego, **FREQUENCY** oraz **SWEEP** są słowami kluczowymi drugiego poziomu, a **START** oraz **STOP** to słowa kluczowe trzeciego poziomu. Słowa kluczowe języka SCPI można zapisywać w postaci pełnej (rozwinętej), np. **FREQUENCY**, ale można je także zapisywać w postaci skrótów, z reguły czteroliterowych, np. **FREQ**. Do ich zapisu można

używać zarówno wielkich, jak i małych liter. W części wykonawczej ćwiczenia poszczególne słowa sterujące zapisywane będą w formie rozwiniętej, natomiast skrócona postać komendy będzie zapisywana dużymi literami, w tym przypadku **FREQUENCY**. Słowa kluczowe można zapisywać tylko w postaci rozwiniętej lub ww. skrótu, nie są dopuszczalne formy pośrednie.

Warto dodać, że drzewa hierarchii rozkazowej SCPI, tak jak to jest przyjęte w technice komputerowej, mają korzeń u góry i rosną w dół. Przykładowe drzewo podsystemu **:SENSE** (rys. 9.1) pokazuje reprezentatywne rozkazy poszczególnych poziomów, zaczynając od poziomu korzenia w dół.

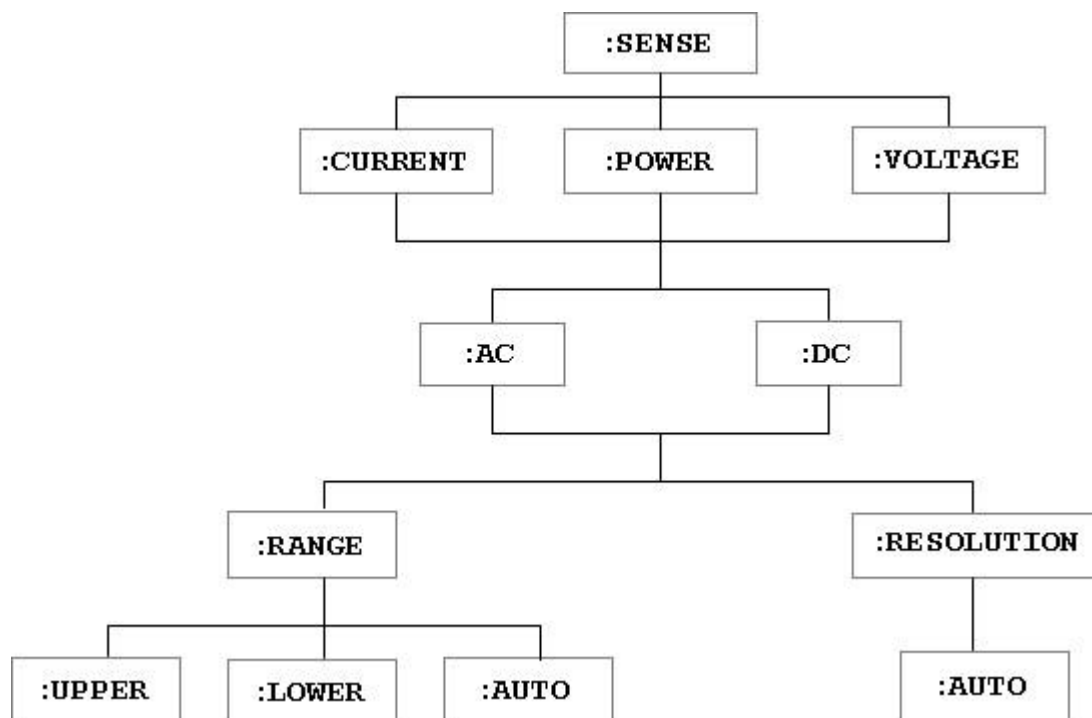
Komunikat programowy SCPI jest łańcuchem elementarnych rozkazów, rozpoczynającym się od rozkazu z korzenia podsystemu i schodzących w dół. Separatorem rozkazów poszczególnych poziomów jest dwukropek, np.:

:SENSE:VOLTAGE:AC:RANGE?

Ten przykładowy komunikat był zapytaniem o zakres pomiarowy multimetru przy pomiarach napięcia przemiennego.

Separatory niższego rzędu w języku SCPI to:

- spacja lub tabulacja, która oddziela parametry od rozkazu;
- przecinek oddzielający parametry wewnątrz ich listy;
- średnik, który oddziela polecenia różnych podsystemów (terminator) lub pozwala użyć kolejno dwóch różnych rozkazów z tego samego poziomu danego drzewa podsystemu.



Rys. 9.1. Przykładowe drzewo hierarchii rozkazowej języka SCPI

Przykładem nieco bardziej złożonego komunikatu programowego może być:

:SENSE:VOLTAGE:AC:RANGE:UPPER 5;LOWER 0.2

Przy czym pokazano tu sposób wykorzystania jako separatorów spacji i średnika.

Poza podsystemami rozkazów służącymi bezpośrednio do kontroli przyrządu, język ten zawiera też rozkazy o charakterze uniwersalnym, przerzucające ciężar konfiguracji sprzętu pomiarowego i pomiarów na sam sprzęt. Jednym z takich bardzo uniwersalnych rozkazów jest komunikat **:MEASURE** – zmierz, który może być użyty w sposób następujący:

:MEASURE:CURRENT:DC?

Multimetr sam dobierze tu właściwy zakres pomiarowy, przeprowadzi pomiar (alternatywnie niezbędne byłoby skorzystanie z podsystemu **:TRIGGER**) i prześle wynik pomiaru do komputera.

Zestaw poleceń o mniejszej uniwersalności, ale większej elastyczności, to na przykład para **:CONFIGURE** – **:READ**. Między te polecenia można wprowadzić rozkazy modyfikujące sprzętową konfigurację inicjowaną rozkazem **:CONFIGURE**, na przykład:

CONFIGURE:VOLTAGE:DC 10, 0.001

TRIGGER:SOURCE:EXTERNAL

READ?

Pierwszy komunikat programowy jest dość oczywistą konfiguracją sprzętu, drugi definiuje zewnętrzne wyzwalenie pomiaru, a trzeci powoduje przejście multimetru w stan oczekiwania na wyzwolenie.

Szczególną pozycję w języku SCPI zajmuje zestaw tzw. rozkazów wspólnych. Zasadniczym zadaniem tej specyficznej grupy rozkazów jest przede wszystkim odczyt i ustawienie rejestrów stanu przyrządu SCPI. Przykładowe rozkazy tej grupy to m.in.:

***CLS** – zerowanie rejestrów stanu,

***ESR** – odczyt rejestru zdarzeń,

***OPC** – ustawienie bitu „operacja zakończona”,

***RST** – ustawienie stanu początkowego (ang. *reset*),

***SRE <wartość>** – zapis maski blokującej,

***STB?** – odczyt bajtu stanu przyrządu,

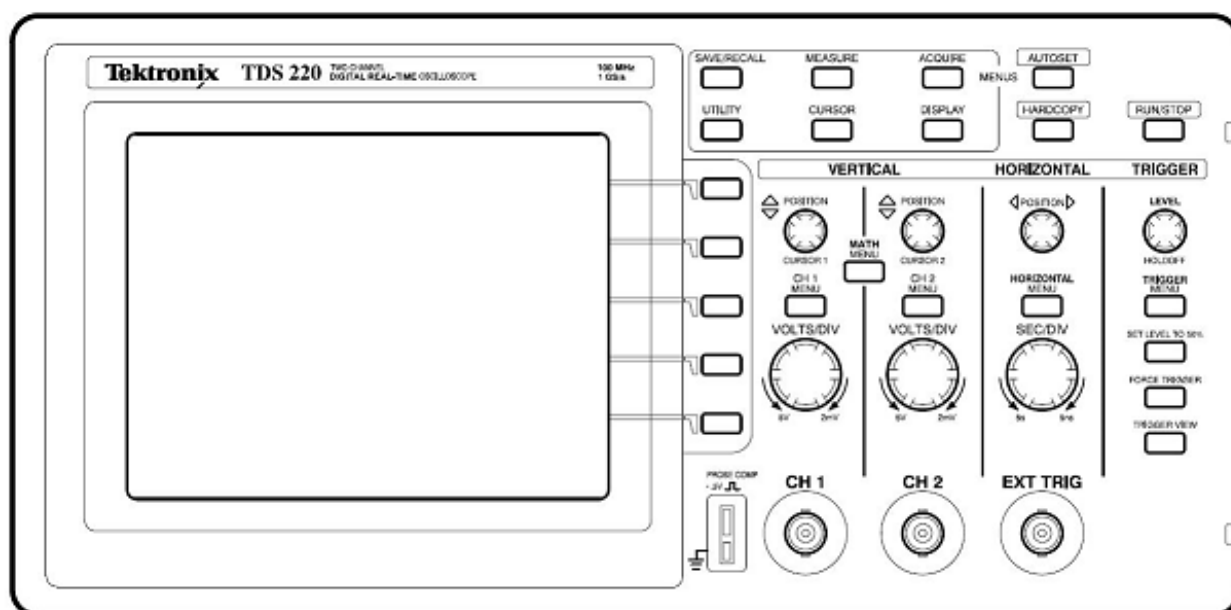
***TRG** – zainicjowanie (wyzwolenie) uprzednio zaprogramowanego pomiaru.

Komunikat w języku SCPI jest w istocie ciągiem znaków przesyłanych łączem komunikacyjnym, np. RS-232, i musi być uzupełniony o znacznik sygnalizujący jego koniec (ang. *End of Line*, EOL). Powszechnie w tym celu wykorzystuje się znaki specjalne ASCII sterujące wydrukiem, np. znak LF (przejście w wydruku do nowego wiersza), znak CR (powrót karetki drukarki na początek wiersza) albo sekwencję tych znaków – CR/LF lub LF/CR.

9.7. Przyrządy pomiarowe wykorzystywane w ćwiczeniu

Oscyloskopy cyfrowe firmy Tektronix rodziny TDS-200 oraz TDS-1000

Amerykańska firma Tektronix w roku 1996 wprowadziła do sprzedaży rodzinę oscyloskopów cyfrowych TDS-200. Są to przyrządy próbujące sygnał wejściowy z prędkością do 1 GSps. Rejestrowane przebiegi prezentowane są na monochromatycznym ekranie LCD o rozdzielczości 320×240 pikseli. Obraz tworzony jest na podstawie rekordu danych o długości 2500 próbek.



Rys. 9.2. Panel czołowy oscyloskopu Tektronix TDS-220

Są to przyrządy mikroprocesorowe sterowane za pomocą procesora MC 68000 firmy Motorola. W oscyloskopach tych zastosowano specyficzny sposób rejestracji sygnału, tzw. technikę FISO (ang. *Fast-In Slow-Out*). Polega ona na pobieraniu z bardzo dużą prędkością (1 GSps) analogowych próbek sygnału i zapamiętywaniu ich wartości w postaci ładunków zgromadzonych w komórkach matrycy CCD, a następnie sekwencyjnym wyprowadzaniu ich z takiego analogowego banku pamięci i przetwarzaniu do postaci cyfrowej za pomocą taniego, 8-bitowego przetwornika A/C, pracującego z częstotliwością rzędu 4 MHz.

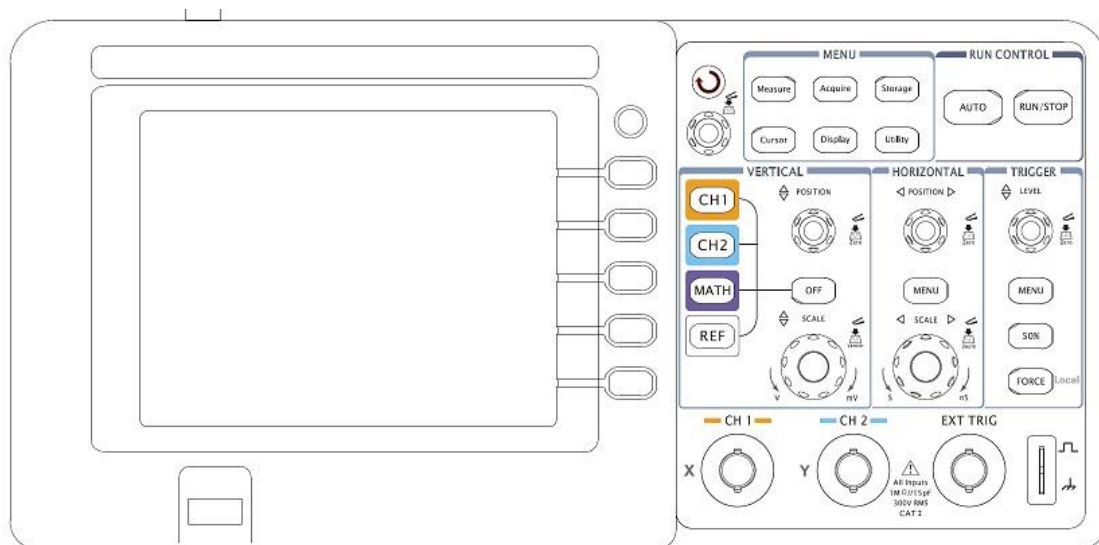


Rys. 9.3. Panel tylny oscyloskopu ze złączami interfejsów

Na panelu tylnym tych oscyloskopów (rys. 9.3) znajdują się złącza trzech interfejsów: szeregowego RS-232 oraz równoległych – Centronics (drukarkowy) i GPIB.

Oscyloskopy cyfrowe firmy Rigol rodziny DS-1000E

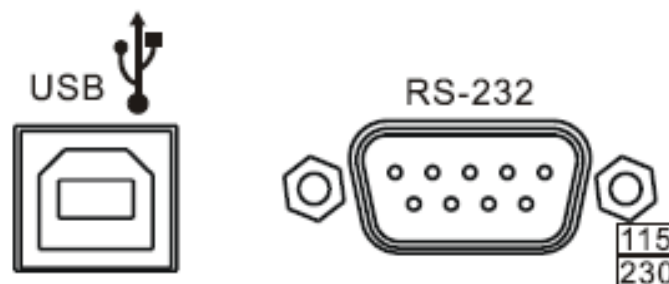
Chińska firma Rigol w roku 2008 wprowadziła do sprzedaży rodzinę oscyloskopów cyfrowych DS-1000E. Są to oscyloskopy próbkujące sygnał wejściowy z prędkością do 1 GSps. Przebiegi badane prezentowane są na kolorowym ekranie TFT LCD o rozdzielczości 320 × 234 piksele. W ćwiczeniu będzie wykorzystywany model DS-1052E o pasmie 50 MHz oraz dwóch kanałach wejściowych.



Rys. 9.4. Panel czolowy oscyloskopu Rigol DS-1052E

Przyrząd sterowany jest za pomocą 16-bitowego mikroprocesora Blackfin ADSP-BF531 firmy Analog Devices, zawierającego w swojej strukturze m.in. procesor sygnałowy oraz sterownik transmisji szeregowej (UART). Ponadto w skład systemu wchodzi 16 MB pamięci operacyjnej SDRAM, pamięć programu (ang. *firmware*) typu flash EPROM o pojemności 8 MB oraz 1 MB szybkiej pamięci FIFO na próbki sygnału. Przetwarzanie analogowo-cyfrowe jest realizowane za pomocą zespołu 10 przetworników A/C (5 podwójnych układów AD9288 firmy Analog Devices, o rozdzielczości 8 bitów, taktowanych sygnałem 100 MHz) pracujących „na zakładkę” – z przesunięciem czasowym 1 ns jeden względem drugiego. Dzięki temu osiągnięto efektywną częstotliwość próbkowania równą 1 GSps.

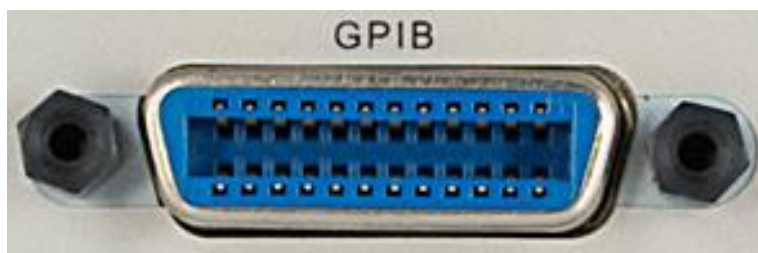
Na panelu tylnym tych oscyloskopów umieszczone są dwa standardowe porty transmisyjne przyrządu (rys. 9.5), za pomocą których mogą one być sterowane zdalnie: RS-232 oraz USB (*device*). W charakterze sterownika portów USB wykorzystano w tym przyrządzie układ ISP 1362 firmy Philips (obsługuje USB host na panelu czołowym oraz USB device – na tylnym). Transmisja przez łącze szeregowo RS-232 realizowana jest za pomocą układu UART zintegrowanego w strukturze głównego mikroprocesora.



Rys. 9.5. Łącza transmisyjne oscyloskopu RIGOL DS-1052E

9.8. Podstawowe informacje o systemie interfejsu GPIB

Okolo roku 1970 firma Hewlett-Packard opracowała cyfrowe łącze komunikacyjne, którego pierwotnym zadaniem było zdalne sterowanie przyrządami pomiarowymi. System ten szeroko upowszechnił się w świecie. Został opisany w amerykańskich normach branżowych o oznaczeniu IEEE-488.1 oraz IEEE-488.2. Powszechnie nazywa się go GPIB (ang. *General Purpose Interface Bus*). Złącze tego interfejsu przedstawione jest na rys. 9.6.



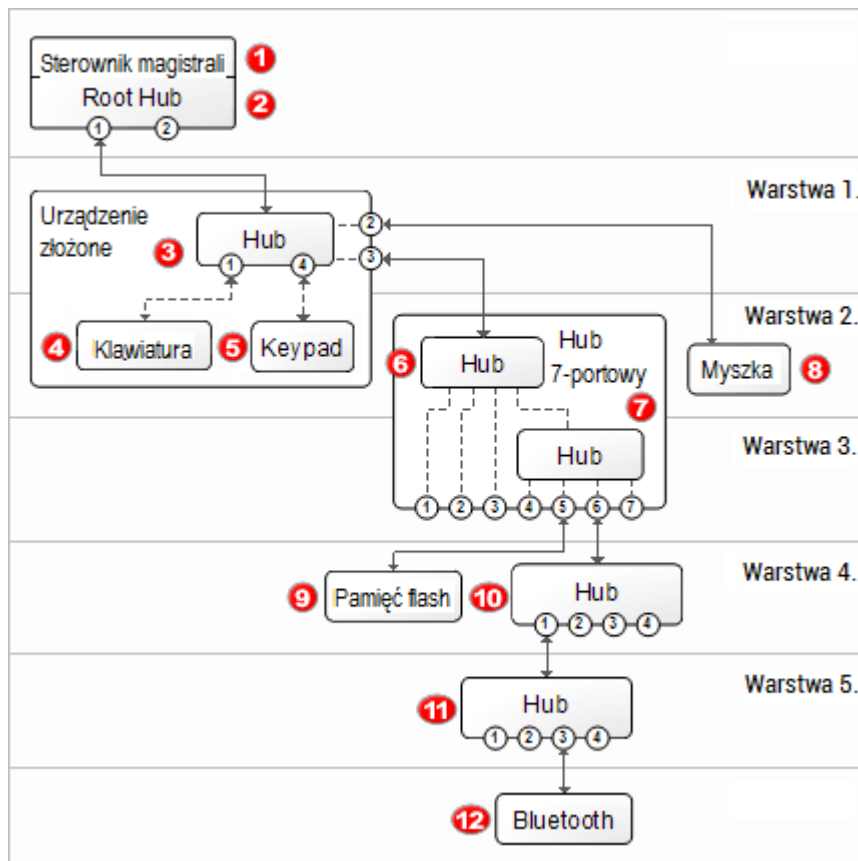
Rys. 9.6. Złącze systemu interfejsu GPIB

Jest to system magistralowy, równoległy, z 8-bitową szyną danych, umożliwiający jednocześnie podłączenie do 15 przyrządów (w tym również sterownika systemu, którym zwykle jest komputer). Maksymalna szybkość transmisji to 1,0 MB/s. Wszystkie urządzenia w systemie podłączone są równoległe do wspólnej magistrali, w której oprócz ośmiu linii danych występuje osiem linii sterujących: trzy linie wspomagające transmisję po liniach danych oraz pięć linii sterowania. Każde urządzenie w systemie jest identyfikowane przy pomocy unikatowego tzw. adresu. Adres GPIB jest liczbą całkowitą z zakresu od 0 do 30. Jest on zwykle przypisywany każdemu urządzeniu na etapie zastawiania (konfigurowania) systemu.

System interfejsu GPIB przez ostatnie kilkadziesiąt lat był szeroko stosowany do zdalnego sterowania przyrządami i systemami pomiarowymi, powstało bardzo dużo oprogramowania wykorzystującego ten interfejs, zarówno firmowego, jak i tworzonych samodzielnie przez użytkowników przyrządów pomiarowych.

9.9. Podstawowe informacje o systemie interfejsu USB

Uniwersalna magistrala szeregową (ang. *Universal Serial Bus*, USB) to rodzaj sprzętowego portu komunikacyjnego komputerów, opracowanego przez firmy Intel, Microsoft, Compaq, IBM i DEC w latach 1994-1996. Łączy to ma topologię asymetrycznego wielowarstwowego drzewa, składającego się ze sterownika magistrali (*hosta*), pewnej liczby portów (łączy transmisyjnych) oraz urządzeń peryferyjnych dołączonych do tych portów. Dodatkowo do portów mogą być dołączone koncentratory (ang. *hub*), które umożliwiają rozgałęzianie magistrali i dodawanie do jej struktury kolejnych warstw (poziomów). W pojedynczym, spójnym systemie dopuszczalne jest wykorzystanie do 5 warstw oraz do 127 urządzeń końcowych. Na rys. 9.7 przedstawiono przykładowy system urządzeń pracujących w tym standardzie, wykorzystujący pięć warstw połączeń.



Rys. 9.7. Przykład topologii systemu USB z punktu widzenia sterownika magistrali

Transmisja informacji odbywa się tu szeregowo po jednej parze przewodów, naprzemiennie – od sterownika magistrali do urządzenia końcowego, a następnie od urządzenia do sterownika. Przewody w magistrali (rys. 9.8) mają postać skrętki i oznaczone są jako „Data+” oraz „Data-”.



Rys. 9.8. Przewody transmisji danych magistrali USB

Transmisja odbywa się w postaci porcji (ciągów) bitów nazywanych pakietami. Pakiety mogą mieć różną długość i mogą realizować różne funkcje. Dozwolone są następujące prędkości transmisji bitów w pakiecie:

- 1,5 Mbita/s (praktycznie współcześnie niewykorzystywana),
- 12 Mbitów/s (tzw. Full-Speed),
- 480 Mbitów/s (tzw. High-Speed),
- 5 Gbitów/s (tzw. SuperSpeed),
- 10 Gbitów/s (tzw. SuperSpeed+).

Z punktu widzenia logiki działania USB jest systemem gwiazdowym, tzn. urządzenia końcowe komunikują się wyłącznie ze sterownikiem magistrali (bezpośrednio lub za pośrednictwem koncentratorów), natomiast nie mogą komunikować się ze sobą. Inicjatywa przy nawiązywaniu połączenia leży wyłącznie po stronie sterownika magistrali, urządzenie końcowe nie może próbować nawiązywać połączenia ze sterownikiem z własnej inicjatywy. W związku z tym w trakcie pracy systemu sterownik magistrali nieustannie wysyła do poszczególnych urządzeń końcowych pakiety zapytania, czy potrzebują one przeprowadzić jakąś transmisję. Zapytania przesyłane są do poszczególnych urządzeń po kolei lub według jakiegoś ustalonego priorytetu.

Ponieważ urządzenia końcowe, które można dołączyć do magistrali USB, mają różne wymagania odnośnie do częstości i charakteru realizowanych transmisji danych, zostały one podzielone na tzw. klasy USB. Klasy urządzeń końcowych są wymienione i opisane w dokumentach normatywnych standardu USB.

Po dołączeniu nowego urządzenia do portu USB kontroler magistrali samoczynnie wykrywa ten fakt, nawiązuje połączenie z urządzeniem, nadaje mu unikatowy identyfikator, tzw. adres USB (liczba całkowita z zakresu od 0 do 127), rozpoznaje klasę urządzenia, poznaje jego potrzeby transmisyjne. W dalszej kolejności powiadamia system operacyjny komputera nadrzędnego o potrzebie wgrania do pamięci operacyjnej sterownika odpowiedniej klasy urządzenia USB.

Na potrzeby przeprowadzania transmisji USB przez systemy mikroprocesorowe produkuje się specjalizowane układy scalone umożliwiające taką transmisję. Przykładem jest ISP 1362 firmy Philips. W jego strukturze zawarte są dwa niezależne sterowniki USB. Pierwszy z nich, który może pełnić wyłącznie rolę hosta USB, wykorzystywany bywa zwykle do zapisywania danych, np. wyników pomiarów, na dołączanej do przyrządu pamięci półprzewodnikowej (*pendrive*). Drugi sterownik, zawarty w układzie, może być programowo przełączany w tryb urządzenia peryferyjnego (USB device) albo w tryb USB host. Pracując jako urządzenie peryferyjne, układ umożliwia podłączenie przyrządu do komputera i zdalne sterowanie nim, natomiast jako USB host pozwala na bezpośrednie dołączenie drukarki i wydruk na niej danych w rozmaitych formatach (także graficznych).

W 2003 roku producenci przyrządów pomiarowych dodali do standardu USB opis dodatkowej, kolejnej klasy urządzeń końcowych o nazwie USBTMC (ang. *USB Test & Measurement Class*). Są to przyrządy pomiarowe wyposażone w interfejs USB, którymi można za pomocą tego łącza zdalnie sterować i odbierać od nich informacje o wynikach przeprowadzonych pomiarów. W praktyce nowe modele konstruowanych przyrządów pomiarowych są wyposażane w nowy interfejs USB w zastępstwie stosowanych wcześniej interfejsów RS-232 oraz GPIB albo jest on dokładany jako kolejny interfejs wraz z dotychczasowymi.

W związku z powszechną wśród programistów i użytkowników systemów pomiarowych znajomością standardu interfejsu GPIB, w dokumentach normatywnych klasy USBTMC dodano dodatkową podklasę urządzeń końcowych o nazwie USB488. Podklasa ta dotyczy urządzeń, które przy sterowaniu przez interfejs USB zachowują się tak, jakby to sterowanie odbywało się przez GPIB (opisany w normach IEEE-488.1 oraz IEEE-488.2, stąd oznaczenie podklasy). Zdecydowana większość produkowanych obecnie przyrządów pomiarowych sterowanych przez USB jest zgodna z podklasą USB488. Stanowi to znaczne ułatwienie dla użytkowników systemów pomiarowych oraz programistów, którzy potrzebują zastosować w istniejących systemach nowe przyrządy wyposażone w łącze USB. Nakład pracy na modyfikację istniejącego oprogramowania sterującego jest w tej sytuacji zdecydowanie mniejszy, a w pewnych przypadkach oprogramowania nie trzeba zmieniać wcale.

9.10. Ogólne właściwości multimetrów Rigol rodziny DM3000

Multimetry cyfrowe serii DM3000 zostały wprowadzone do sprzedaży przez chińską firmę Rigol Technologies w roku 2006. Są to przyrządy interfejsowe, wielofunkcyjne, o bardzo rozbudowanych w stosunku do klasycznych przyrządów możliwościach.

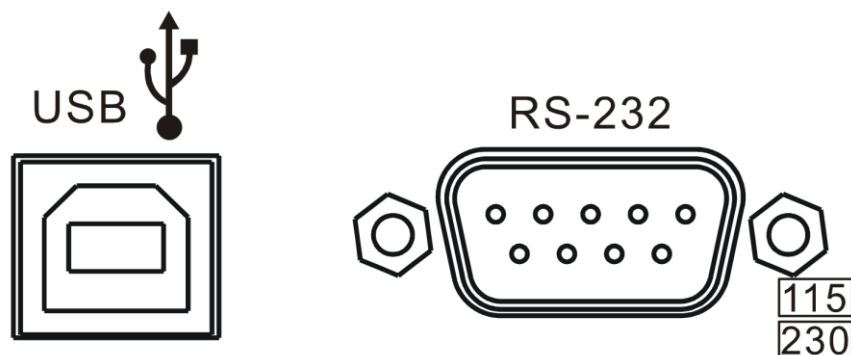
Na panelu czołowym miernika (rys. 9.9) użytkownik ma do dyspozycji zestaw podświetlanych przycisków oraz monochromatyczny wyświetlacz LCD o wymiarach 256×64 piksele.

Multimetry te to przyrządy mikroprocesorowe, sterowane są za pomocą 16-bitowego procesora Blackfin ADSP-BF531 firmy Analog Devices, zawierającego w swojej strukturze mikrokontroler typu RISC oraz procesor sygnałowy. Pomiar napięcia jest realizowany za pomocą 24-bitowego przetwornika analogowo-cyfrowego sigma-delta (Σ - Δ) typu ADS1256 firmy Texas Instruments.



Rys. 9.9. Panel czołowy multimetru Rigol DM3068

Wszystkie modele tej rodziny wyposażone są na panelu tylnym (rys. 9.10) w złącza interfejsów szeregowych RS-232 oraz USB, niektóre zaś posiadają ponadto interfejs równoległy GPIB oraz interfejs sieciowy LAN (Ethernet).



Rys. 9.10. Złącza interfejsów USB oraz RS-232 na panelu tylnym

Przyrządy z tej rodziny prezentują wynik pomiaru z rozdzielczością $5\frac{3}{4}$ albo $6\frac{1}{2}$ cyfry, niepewność podstawowa przy pomiarze napięć stałych jest równa $0,025\% \cdot U_x + 0,006\% \cdot U_{zakr}$, potrafią wykonywać do 50 tysięcy pomiarów na sekundę. Są wyposażone w wewnętrzne banki pamięci: pamięć ulotną (RAM) o pojemności 2 milionów wyników pomiarów oraz pamięć nielotną (EEPROM) na 512 wyników. Dodatkowo na panelu czołowym umieszczony jest port USB (host), do którego można dołączyć zewnętrzne urządzenie pamięciowe (np. *pendrive*) i przepisać na nie zarejestrowane wyniki pomiarów albo nastawy przyrządu. Multimetry te mają wbudowane 24 podstawowe funkcje pomiarowe, do których należą m.in.: pomiary napięcia i prądu – stałego i przemiennego, rezystancji, pojemności elektrycznej, testowanie diod, testowanie ciągłości przewodu, pomiary częstotliwości lub okresu przebiegu, pomiary z wykorzystaniem czujników wielkości nieelektrycznych. Posiadają także wiele wbudowanych funkcji matematycznych: odejmowanie stałej wartości (pomiar względny), zapamiętywanie wartości minimalnej, maksymalnej i średniej z serii pomiarów, wyliczanie poziomu sygnału w dB w stosunku do wybranego poziomu odniesienia, wyliczanie poziomu mocy sygnału w dBm, testowanie tolerancji wartości parametrów.

Ze względu na mnogość funkcji przyrządu i potrzebę ograniczenia liczby przycisków na panelu czołowym, programuje się go do pracy za pomocą rozbudowanego systemu menu ekranowego oraz sześciu przycisków tego menu umieszczonych pod ekranem (rys. 9.9).

9.11. Wiadomości ogólne o programowaniu w środowisku Agilent VEE

Program VEE (ang. *Visual Engineering Environment*) jest produktem firmy Hewlett-Packard i został wprowadzony na rynek w roku 1991, początkowo dla systemu operacyjnego Unix, a w dalszej kolejności został zaadaptowany do MS Windows. Od tego czasu program przeszedł wiele modyfikacji. Jest graficznym językiem programowania przeznaczonym do sterowania przyrządami przy pomocy łącz takich jak GPIB, RS-232, USB, LAN, VXI oraz PXI w trakcie samego procesu pomiarowego, jak i podczas dalszego przetwarzania danych uzyskanych z pomiarów. Pakiet ten nadaje się także do symulacji tych działań. Program VEE nie jest trudny do nauki, nie wymaga znajomości słów czy komend kluczowych, a poruszanie się po nim jest tak intuicyjne, że programowania w VEE można nauczyć się, mając jedynie ogólne pojęcie o pracy systemów pomiarowych.

9.12. Zapoznanie z oknem roboczym programu VEE

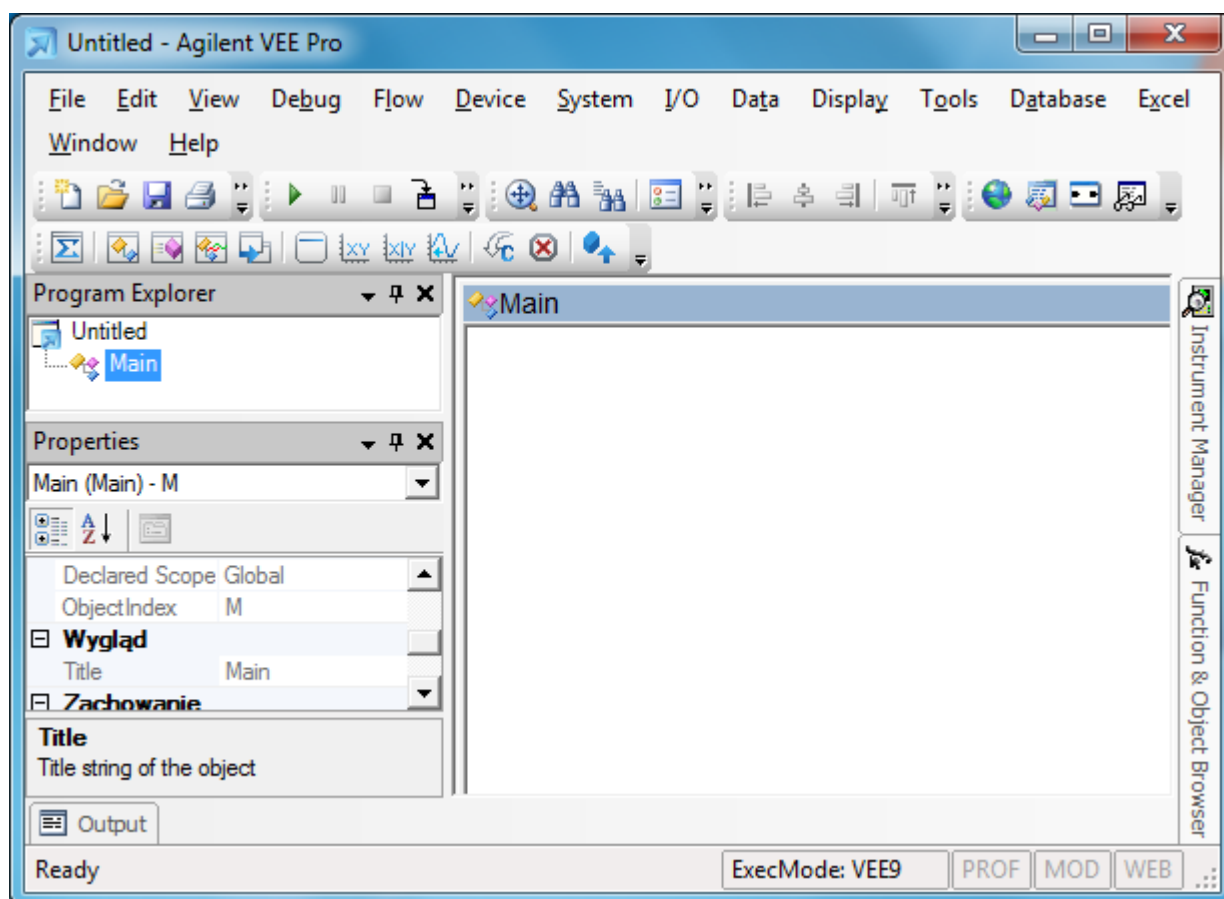
Rysunek 9.11 przedstawia okno robocze programu VEE. Składa się ono z czterech zasadniczych części:

- a) paska menu,
- b) paska narzędziowego,
- c) okna pola roboczego,
- d) okna eksploratora programu.

Pasek menu udostępnia szereg bibliotek obiektów graficznych używanych do budowy konkretnych programów.

Pasek narzędziowy składa się z klawiszy kontroli wykonywania programów oraz umożliwia szybki dostęp do często wykorzystywanych poleceń z menu **File**, **Edit**, **View**, **Debug**.

Pole robocze jest obszarem ekranu, który służy do konstruowania **grafu przepływu danych oraz uaktywnień** – tworzenia programu (graf ten jest odpowiednikiem kodu źródłowego w tekstowych językach programowania, takich jak: Pascal, C++, Java itp.).

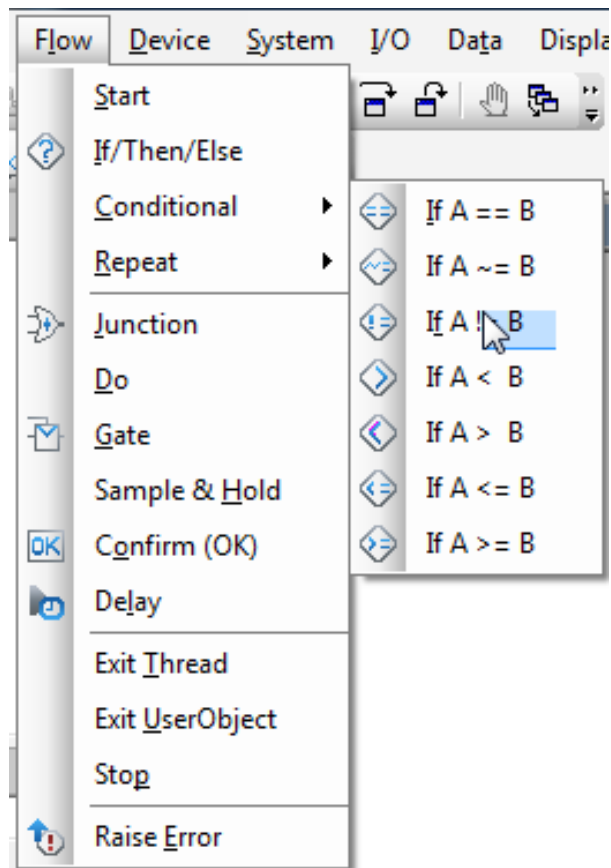


Rys. 9.11. Okno robocze programu

Środowisko programu VEE umożliwia tworzenie aplikacji zorganizowanych wielopoziomowo, tzn. aplikacje te mogą składać się z procedur, które mogą zawierać kolejne procedury itd. Procedury te, nazywane **obiektami użytkownika**, reprezentowane są przez specjalne okienka.

9.13. Zapoznanie z menu programu

Pasek menu umożliwia szybki dostęp do bogatych bibliotek obiektów oraz do pewnych funkcji dostępnych w programie – podobnie jak to jest w innych aplikacjach WINDOWS. Przykład organizacji menu **Flow** przedstawia rysunek 9.12. Pełny opis poszczególnych menu zawiera dokumentacja techniczna.



Rys. 9.12. Rozwinięte menu „Flow → Conditional → If...”

9.14. Operacje wykonywane na obiektach

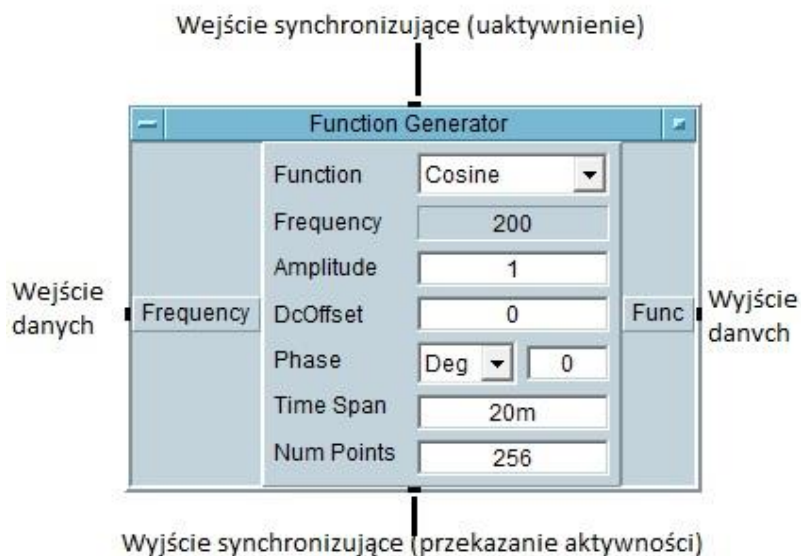
Obiektem nazywamy każdy blok lub jego ikonę uzyskaną z menu programu. Obiekty takie mogą reprezentować rzeczywisty przyrząd podłączony do komputera za pomocą interfejsu lub jakąś operację wykonywaną na dostarczonych danych.

Tworzenie programu w środowisku Agilent VEE polega na umieszczaniu, za pomocą myszki, w obszarze pola roboczego różnych obiektów i wykonywaniu między nimi połączeń, które obrazują przepływ informacji między obiektami i ustalają kolejność uaktywniania obiektów. Tworzony jest wtedy tzw. **graf przepływu danych i uaktywnień**.

Każdy obiekt w grafie może być zobrazowany w dwóch postaciach:

- w postaci rozwiniętej – otwartej,
- w postaci ikony.

9.15. Zaciski obiektu

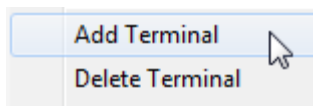


Rys. 9.13. Zaciski obiektu

Każdy obiekt w programie VEE jest wyposażony w wejścia i wyjścia – synchronizujące i danych. Rozmieszczenie tych wejść i wyjść przedstawia rysunek 9.13. Zaciski wejściowe dla danych umieszczane są zawsze z lewej strony okienka lub ikony obiektu, zaś zaciski wyjścia danych lub informacji o błędach – z prawej strony. Wejścia i wyjścia danych mogą występować w różnej liczbie, może ich także nie być wcale. W niektórych obiektach programista może tworzyć dodatkowe takie zaciski lub usuwać już istniejące. Do wejść danych muszą być doprowadzone dane z innego obiektu (wejście nie może pozostać niepodłączone).

Wejście synchronizujące może być w obiekcie tylko jedno i jest umieszczone u góry okienka. Służy ono do przekazywania obiektowi sygnału pobudzającego do działania od innego obiektu. Wyjście synchronizujące umieszczone jest na spodzie okienka i służy do wysłania sygnału pobudzenia do innego obiektu po zakończeniu działania przez dany obiekt. Przesyłanie pobudzeń za pomocą wejść i wyjść synchronizujących służy do ustalenia kolejności wykonywania operacji przez obiekty tworzące program, jeśli ma to dla aplikacji znaczenie. W przypadku gdy wejście synchronizujące nie jest podłączone (nie ma doprowadzonego sygnału pobudzającego z innego obiektu) zakłada się, że sygnał pobudzający jest stale obecny.

W menu niektórych obiektów znajdują się polecenia **Add Terminal** (dodaj zacisk) i **Delete Terminal** (usuń zacisk).

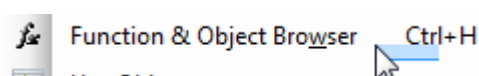


Po kliknięciu polecenia **Add Terminal** będzie można dołączyć do obiektu dodatkowe wejście umożliwiające sterowanie wybranym przez użytkownika parametrem (komponentem) obiektu.

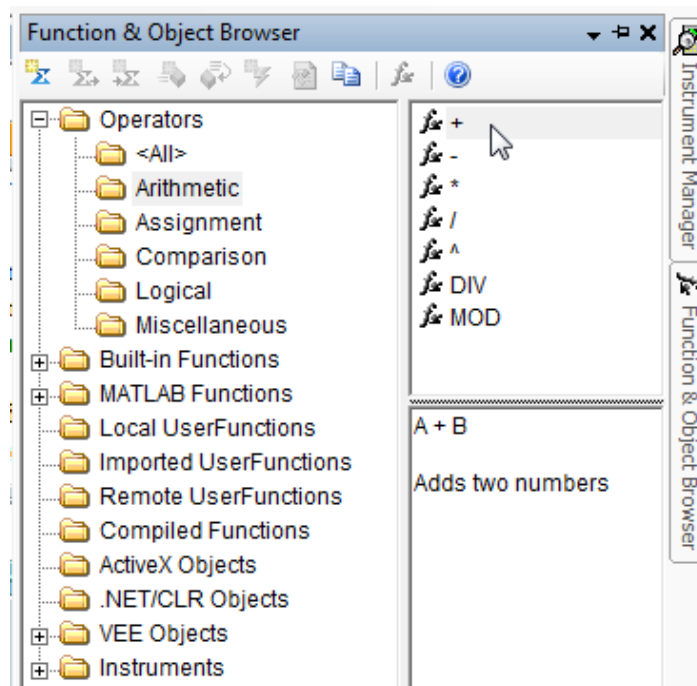
Zaciski można także usuwać. W tym celu wystarczy wybrać **Delete Terminal**. Klikając podwójnie istniejący zacisk, można uzyskać o nim pewne informacje, np. nazwę, typ oraz format przesyłanych danych.

9.16. Inne właściwości użytkowe środowiska programowego Agilent VEE

Ogromną liczbę wbudowanych funkcji zawiera **Function & Object Browser** w menu **Device** programu (rys. 9.14).



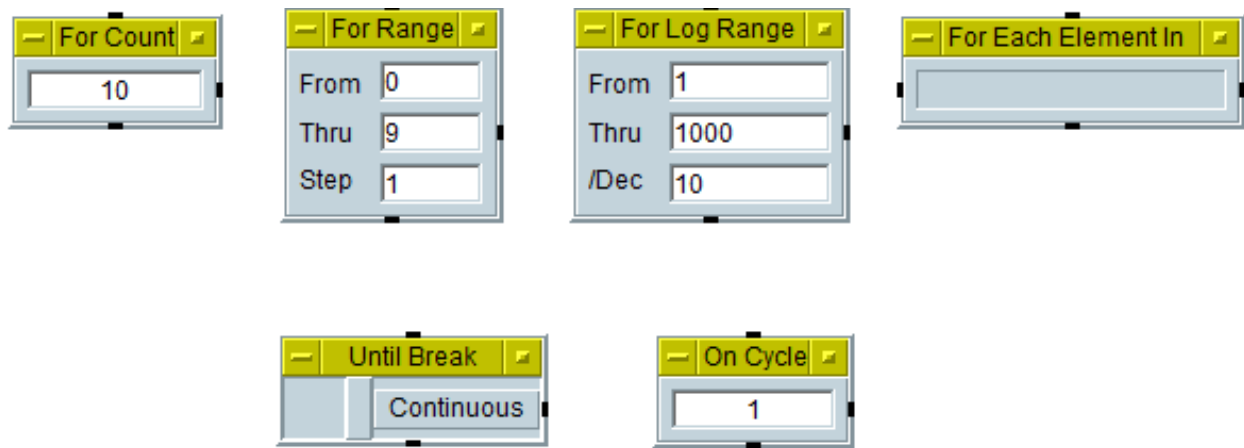
Przeglądarka tych funkcji dzieli je na typy i kategorie.



Rys. 9.14. Przeglądarka wbudowanych operatorów i funkcji

Pętle programowe

Do realizacji pętli programowych służy zestaw obiektów pętli umieszczonych w menu **Flow** opcja **Repeat**. Na rysunku 9.15 pokazano cały zestaw obiektów pętli, które mogą być użyte w programie.



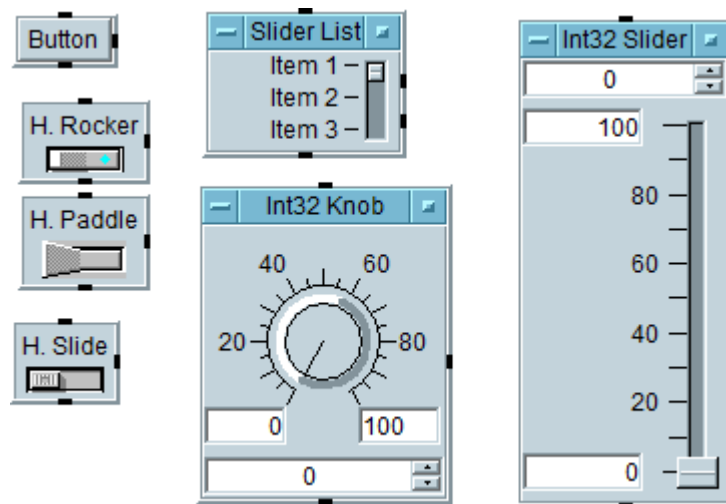
Rys. 9.15. Obiekty realizujące pętle programowe

Dostępnych jest tu sześć typów pętli:

- a) **For Count** – pętla realizująca pewną liczbę powtórzeń, określoną przez wartość wpisaną w polu edycyjnym obiektu,
- b) **For Range** – pętla zapewniająca zmianę wartości na swoim wyjściu danych w zakresie i z krokiem zadeklarowanym przez programistę (krok liniowy),
- c) **For Log Range** – pętla zapewniająca zmianę wartości na swoim wyjściu danych z krokiem zmiennym w sposób logarytmiczny,
- d) **For Each Element In** – pętla umożliwiająca powtarzanie operacji dla każdego elementu zbioru lub tablicy danych,
- e) **Until Breake** – pętla zapewniająca nieskończone powtarzanie, aż do momentu uaktywnienia obiektu **Stop** lub przerwania programu przez operatora,
- f) **On Cycle** – pętla zapewniająca okresowe powtarzanie pewnych operacji; okres powtarzania deklarowany jest w sekundach w okienku edycyjnym obiektu.

Obiekty sterujące

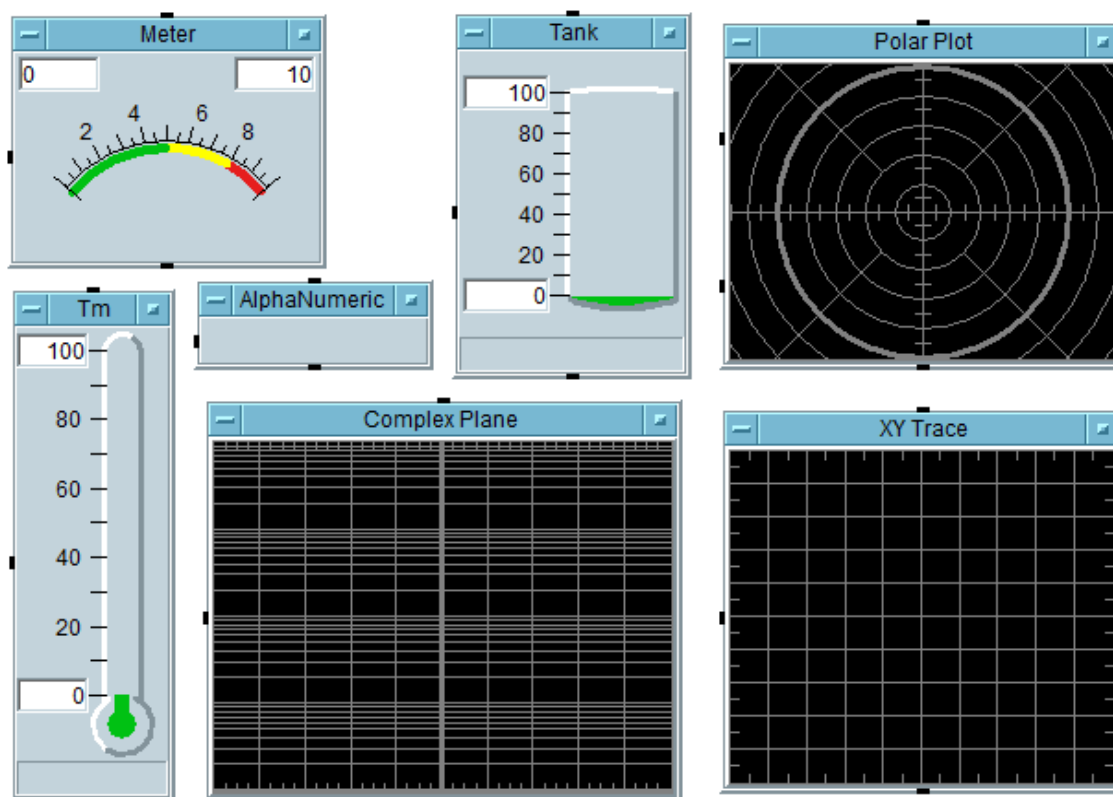
Niektóre obiekty sterujące przeznaczone do tworzenia graficznego interfejsu użytkownika przedstawione zostały na rysunku 9.16. Są one dostępne w menu **Data**, opcje: **Selection Control**, **Toggle Control** oraz **Continuous**.



Rys. 9.16. Przykładowe obiekty sterujące

Obiekty zobrazowujące

Duża różnorodność obiektów zobrazowujących umożliwia przedstawienie danych w sposób graficzny i bardzo wygodny dla użytkownika programu (rys. 9.17). Obiekty te dostępne są w menu **Display**, opcje: **Indicator**, **XY Trace**, **Strip Chart**, **Complex Plane**, **Polar Plot** i inne.



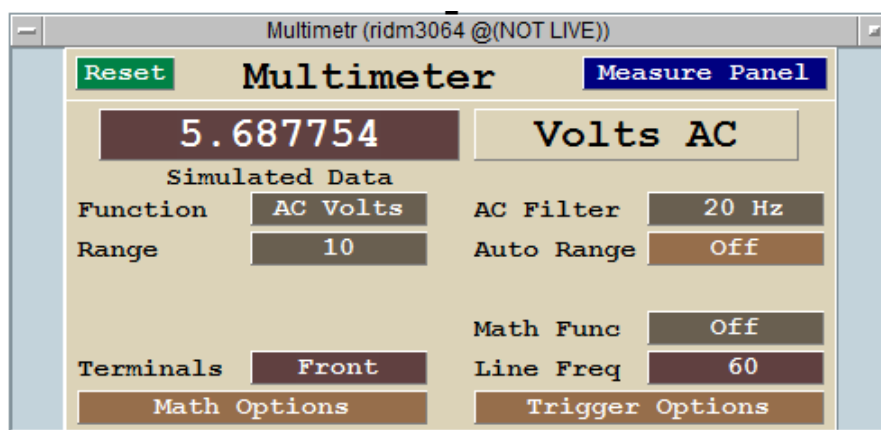
Rys. 9.17. Przykładowe obiekty prezentacyjne


9.17. Obiekty sterujące przyrządami pomiarowymi

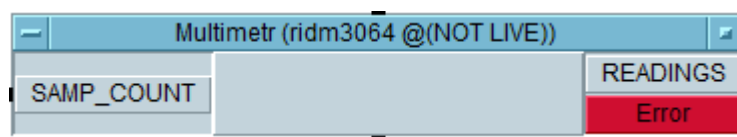
Program VEE dysponuje ponadto dużym zestawem obiektów, których zadaniem jest sterowanie przyrządami pomiarowymi dołączonymi do komputera za pomocą standardowych interfejsów, takich jak RS-232, GPIB, USB i LAN. Obiekty te dostępne są przez menu **I/O**, opcja **Instrument Manager...** Sterowanie przyrządami polega ogólnie na tym, że wszystkie dane doprowadzane do obiektu sterującego (ang. *driver*) są przez ten obiekt samoczynnie wysyłane do odpowiedniego przyrządu pomiarowego, natomiast próba odczytania danych z obiektu sterującego przyrządem związana jest każdorazowo z pobraniem stosownej informacji (wyniku pomiaru, wartości nastawy, informacji o stanie itp.) od podległego przyrządu pomiarowego. Dzieje się to samoczynnie, bez udziału operatora, i zwalnia programistę z potrzeby znajomości szczegółów protokołu komunikacji urządzeń ze sterującym komputerem.

Obiekty sterujące przyrządami mogą występować w trzech postaciach, różniących się wyglądem, łatwością użycia w programie i szybkością pracy. Są to:

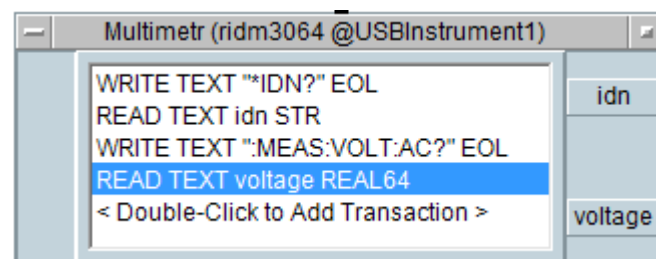
- sterowniki panelowe (*panel drivers*), 



- sterowniki komponentów obiektu (*component drivers*), 



- sterowniki bezpośrednie (*direct I/O*). 



W dalszej części ćwiczenia wykorzystywane będą głównie sterowniki bezpośrednie, jako najbardziej uniwersalne.

9.18. Badania laboratoryjne

Cel ćwiczenia:

- zdalne sterowanie przyrządem pomiarowym za pomocą komend języka SCPI przesyłanych przez interfejs szeregowy RS-232,
- programowe sterowanie innym przyrządem za pomocą prostych aplikacji tworzonych z wykorzystaniem środowiska Agilent VEE; w tym przypadku przyrząd pomiarowy połączony będzie z komputerem za pomocą interfejsu równoległego GPIB albo za pomocą łącza szeregowego USB.

Wykaz przyrządów na stanowisku pomiarowym

Lp.	Nazwa przyrządu
1	Mikrokomputer
2	Oscyloskop cyfrowy
3	Generator funkcyjny
4	Multimetr cyfrowy
5	Rezystor dekadowy

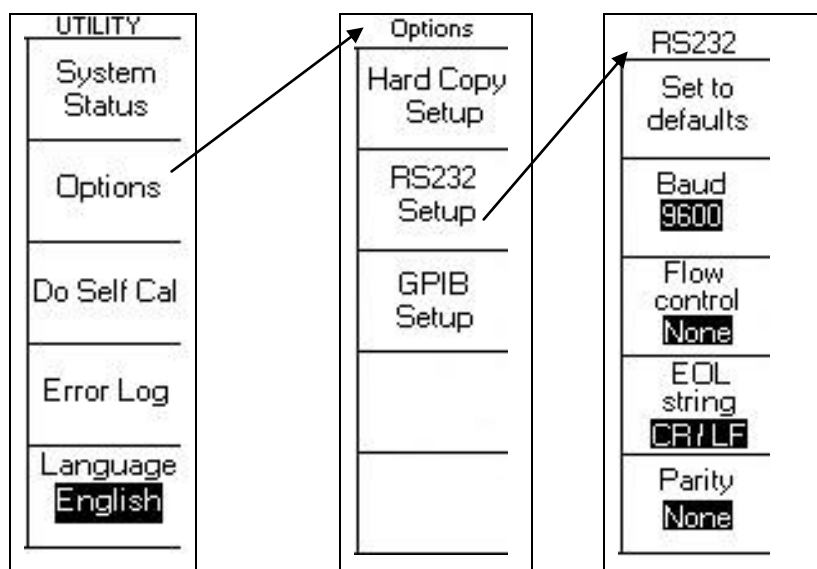
Przebieg ćwiczenia:

Na stanowisku z interfejsem szeregowym RS-232 ćwiczący zostaną praktycznie zapoznani ze sposobem przygotowania przyrządu pomiarowego oraz komputera do pracy: wybrania właściwych nastaw parametrów transmisji danych w przyrządzie pomiarowym oraz w komputerze sterującym tym przyrządem.

1. Przygotowanie przyrządu pomiarowego do pracy sterowanej zdalnie

Aby przygotować oscyloskop firmy Tektronix typu np. TDS-210 do pracy sterowanej zdalnie przez interfejs RS-232, należy:

- na płycie czołowej oscyloskopu wcisnąć klawisz **UTILITY**,
- wcisnąć klawisz programowalny **Options** (rys. 9.18), a następnie **RS232 Setup**,
- za pomocą klawiszy programowalnych, w oparciu o poniższą tabelę, ustawić właściwe wartości parametrów transmisji szeregowej.



Rys. 9.18. Menu ekranowe oscyloskopu przy ustawianiu parametrów portu RS-232

Klawisz programowalny	Rodzaj nastawianego parametru	Możliwe wartości parametru	Nastawiana wartość
Baud	Prędkość transmisji w bitach/s	300, 600, 1200, 2400, 4800, 9600, 19 200	9600
Flow Control	Sterowanie transmisją (<i>handshake</i>)	Hard Flagging (DTR/DSR, RTS/CTS) Soft Flagging (XON/XOFF) None	None
EOL String	Sekwencja znaków końca komunikatu	LF, CR, LF/CR, CR/LF	CR/LF
Parity	Sposób kontroli parzystości	Even Odd None	None

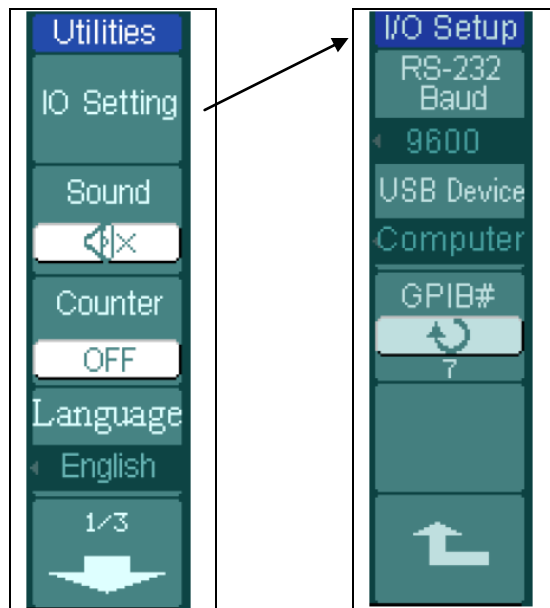
Pozostałe parametry transmisji są fabrycznie ustawione na stałe i nie można ich zmienić. Nastawy te są następujące:

- liczba bitów danych: 8,
- liczba bitów stopu: 1.

Oscyloskopy cyfrowe Tektronix serii TDS-1000 przygotowuje się do pracy sterowanej przez łącze szeregowe RS-232 w sposób analogiczny – za pomocą takiego samego menu ekranowego.

Przygotowanie oscyloskopu DS-1052E do pracy sterowanej zdalnie przez łącze szeregowe RS-232 polega jedynie na wyborze szybkości transmisji. W tym celu należy:

- na płycie czołowej oscyloskopu wcisnąć klawisz **Utility**,
- wcisnąć klawisz programowalny **IO Settings** (rys. 9.19), a następnie **RS-232 Baud**,
- za pomocą wielofunkcyjnego pokrętkła wybrać prędkość transmisji w bitach/s, a następnie zatwierdzić nastawę, przyciskając pokrętkło.



Rys. 9.19. Menu oscyloskopu wraz z wyborem typu interfejsu RS-232

Klawisz programowalny	Rodzaj nastawianego parametru	Możliwe wartości parametru	Nastawiana wartość
RS-232 Baud	Prędkość transmisji w bitach/s	300, 2400, 4800, 9600, 19 200, 38 400	9600

Pozostałe parametry są fabrycznie ustawione na stałe i nie można ich zmienić. Nastawy te są następujące:

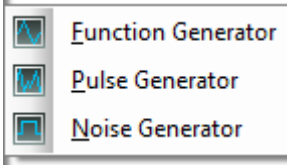
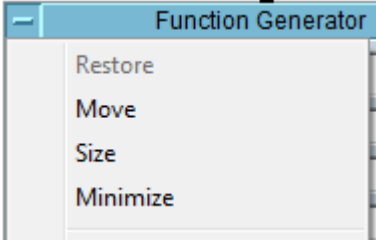
- liczba bitów danych: 8,
- liczba bitów stopu: 1,
- sposób kontroli parzystości: brak (ang. *none*),
- sterowanie transmisją: brak (ang. *none*),
- znak sygnalizujący koniec komunikatu: LF (ang. *Line Feed*).

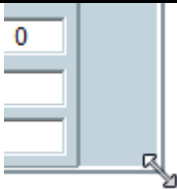
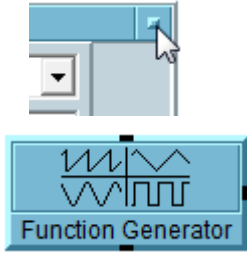

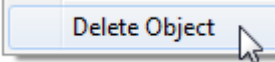
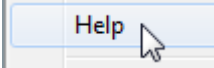
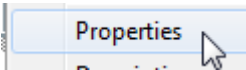
2. Przesyłanie do przyrządu poleceń w języku SCPI

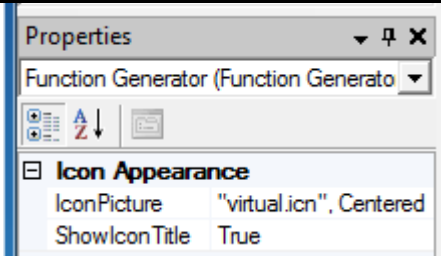
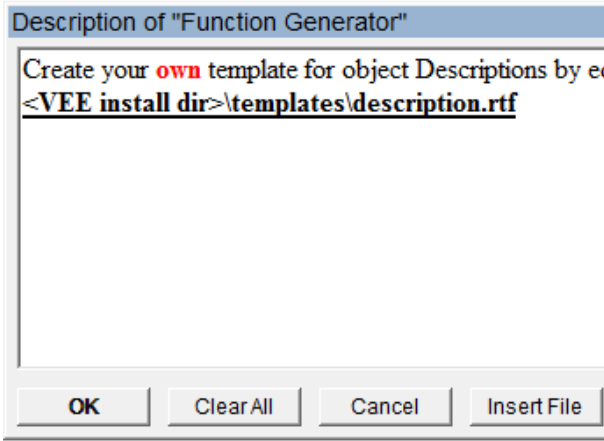
W dalszej kolejności z pomocą klawiatury komputera do przyrządu wysyłane będą polecenia konfiguracyjne miernik do różnorodnych trybów pracy, nakazujące wykonanie pomiarów i odsyłanie ich wyników oraz wartości rozmaitych nastaw zwrótnie do komputera sterującego. Zadaniem ćwiczącego będzie rejestrowanie sposobu reakcji przyrządu na polecenia sterujące, komunikatów pojawiających się na panelu czołowym oraz odpowiedzi przyrządu odsyłanych zwrótnie do urządzenia sterującego.

3. Zapoznanie z oknem roboczym programu VEE, jego menu oraz operacjami wykonywanymi na obiektach

Na stanowisku z programem Agilent VEE ćwiczący wstępnie zapoznani zostaną ze sposobem posługiwania się tym środowiskiem programistycznym, wyglądem okna roboczego, zawartością menu z komponentami, z których pomocą w sposób graficzny tworzone są programy.

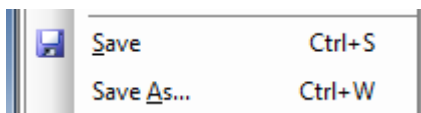
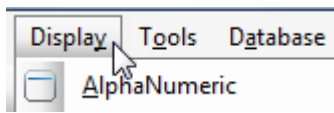
Lp.	Nazwa czynności	Sposób wykonania czynności
1.	Wybór obiektu i umieszczenie go w polu roboczym	<ul style="list-style-type: none"> • posługując się myszką, otworzyć menu Device, • z podmenu Virtual Source wybrać i kliknąć Function Generator,  <ul style="list-style-type: none"> • umieścić zmodyfikowany kursor konturu obiektu w wybranym miejscu pola roboczego i ponownie kliknąć myszką.
2.	Zapoznanie z menu obiektu	<ul style="list-style-type: none"> • kliknąć myszką w górnym lewym rogu obiektu (rozwinie się pasek menu tego obiektu),  <ul style="list-style-type: none"> • wybierać poszczególne polecenia z menu i zapoznać się z ich przeznaczeniem, • zamknąć menu przez kliknięcie w dowolnym miejscu ekranu.
3.	Przemieszczanie obiektu po polu roboczym	<ul style="list-style-type: none"> • najechać znacznikiem myszy na pasek tytułowy obiektu, • nacisnąć i przytrzymać lewy klawisz myszy, • przesunąć kontur obiektu w żądane miejsce ekranu, • zwolnić klawisz myszy.
4.	Zmiana rozmiaru obiektu	<ul style="list-style-type: none"> • wskaźnikiem myszy najechać na dolny lewy róg obiektu, aż do uzyskania znacznika w postaci ukośnej strzałki,

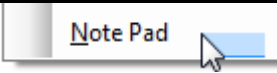
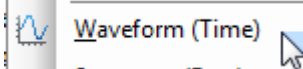
		 <ul style="list-style-type: none"> • przytrzymać lewy klawisz i przemieścić mysz do położenia definiującego nowy rozmiar obiektu, • zwolnić myszkę.
5.	Przejsięcie od zobrażenia otwartego do zamkniętego (ikonowego)	<ul style="list-style-type: none"> • kliknąć myszką w prawym górnym rogu obiektu,  <ul style="list-style-type: none"> • aby powrócić do zobrażenia tzw. otwartego, wystarczy podwójnie kliknąć w ikonę obiektu.
6.	Powielanie obiektu	<ul style="list-style-type: none"> • otworzyć menu obiektu; • wybrać polecenie Clone,  <ul style="list-style-type: none"> • kontur nowego obiektu umieścić w wybranym miejscu ekranu i ponownie kliknąć.
7.	Usuwanie obiektu	<ul style="list-style-type: none"> • otworzyć menu obiektu i wybrać polecenie Delete Object  <p>lub podwójnie kliknąć myszką w lewym górnym rogu obiektu.</p>
8.	Uzyskiwanie informacji o obiekcie	<ul style="list-style-type: none"> • otworzyć menu obiektu i kliknąć hasło Help,  <ul style="list-style-type: none"> • zapoznać się z zawartością okna Agilent VEE Help, • zamknąć okno pomocy.
9.	Zapoznanie z właściwościami obiektu	<ul style="list-style-type: none"> • z menu obiektu wybrać opcję Properties (właściwości),  <ul style="list-style-type: none"> • zapoznać się z zawartością okna <i>Properties</i>, klikając myszką w poszczególne zakładki,

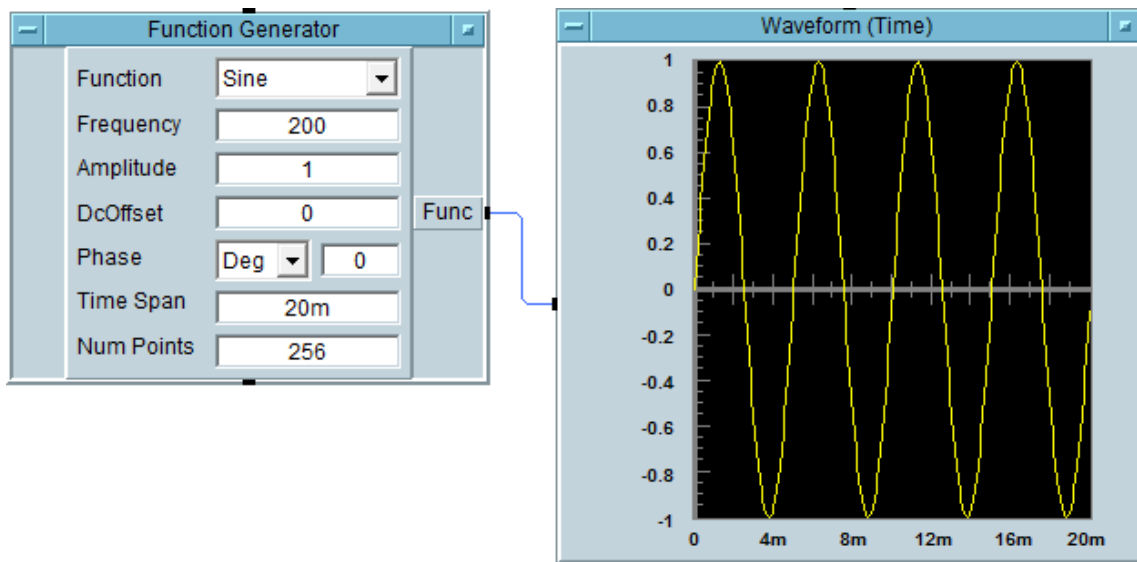
		 <ul style="list-style-type: none"> • zakończyć przyciskiem Cancel.
10.	Edycja opisu obiektu	<ul style="list-style-type: none"> • z menu obiektu wybrać opcję Description (opis),  <ul style="list-style-type: none"> • do pola edycyjnego wpisać wiersz tekstu, • zakończyć edycję przyciskiem OK.

4. Łączenie obiektów w oknie roboczym programu VEE

Sposób łączenia obiektów można prześledzić na prostym przykładzie. Przed przystąpieniem do tworzenia programu należy wyczyścić pole edycyjne. Można to zrobić w opisany wcześniej sposób (usuwając obiekt po obiekcie) lub wybierając w menu **File** opcję **New**.

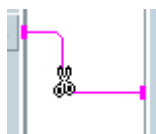
Lp.	Nazwa czynności	Sposób wykonania czynności
1.	Opisanie programu	<ul style="list-style-type: none"> • z menu File wybrać Save As...  <ul style="list-style-type: none"> • w okienku dialogowym w polu Nazwa pliku: wpisać „Zobrazowanie sygnału nieharmonicznego” i zaakceptować klawiszem Zapisz, • z menu Display  <p>wybrać Note Pad</p>

		 <p>i obiekt notatnika umieścić w górnej części pola roboczego,</p> <ul style="list-style-type: none"> • kliknąć na polu tekstowym i wpisać „Program zobrazowania sygnału. Powoduje generacje sygnału nieharmonicznego i wyświetlenie jego przebiegu czasowego”.
2.	Utworzenie programu	<ul style="list-style-type: none"> • z menu Device wybrać VirtualSource ⇒ Function Generator i umieścić w lewej części pola roboczego, • z menu Display wybrać Waveform[Time]  <p>i umieścić w prawej części pola edycyjnego,</p> <ul style="list-style-type: none"> • kliknąć tuż przy wyjściu danych obiektu FunctionGenerator, a następnie przenieść myszkę w pobliżu wejścia danych obiektu Waveform[Time] i kliknąć ponownie (powinno się uzyskać linię danych łączącą obydwa obiekty) – tak jak na rysunku 9.20, • w obiekcie Function Generator kliknąć rozwijalne menu Function i wybrać kształt sygnału, który ma być generowany, np. Square (przebieg prostokątny), • uruchomić tak utworzony program, wybierając z menu Debug opcję Run/Resume lub kliknąć ikonę Run na pasku narzędziowym. Jeśli wszystko zostało zrobione poprawnie, program powinien zacząć działać, można teraz w obiekcie Function Generator wprowadzać różne wartości parametrów generowanego przebiegu i obserwować efekty tych zmian, ponownie naciskając Run.



Rys. 9.20. Pierwszy program w języku graficznym VEE

Aby wykasować niepotrzebne lub błędne połączenie między obiektami, należy, przy wciśniętych klawiszach **Ctrl + Shift**, najechać myszką na linię (znacznik myszki zmieni kształt na nożyczki) i kliknąć ją.



W dalszej kolejności przedstawione zostanie wykorzystanie obiektów sterujących przyrządami pomiarowymi – najpierw interaktywnie, a następnie w sposób programowy. Ćwiczący będą przygotowywać aplikację przełączającą multimetr do pomiaru np. rezystancji, wykonania pojedynczego pomiaru i odesłania jego wyniku do komputera. Następnie przyrząd będzie programowany do wykonania serii pomiarów, po odesłaniu ich wyników do komputera będą one poddane prostemu przetwarzaniu z pomocą wbudowanych w środowisko VEE funkcyjnych (wyznaczenie średniej z serii oraz odchylenia standardowego pojedynczego pomiaru). W kolejnym kroku zaprezentowane zostaną możliwości prezentacji wyników (wykresy) oraz archiwizacji w postaci plików dyskowych.

Opracowanie sprawozdania:

1. Sformułować wnioski oraz spostrzeżenia wynikające z przeprowadzonych pomiarów i obserwacji odnośnie do zdalnego i programowego sterowania przyrządami pomiarowymi.

Przykładowe zagadnienia kontrolne:

1. Parametry techniczne systemu interfejsu RS-232.
2. Właściwości interfejsu szeregowego USB.

9.19. Literatura

1. W. Mielczarek, *Komputerowe systemy pomiarowe – standardy IEEE-488.2 i SCPI*, Wyd. Politechniki Śląskiej, Gliwice 2002.
2. W. Mielczarek, *Szeregowe interfejsy cyfrowe*, Wyd. Helion, Gliwice 1996.
3. W. Nawrocki, *Komputerowe systemy pomiarowe*, WKiŁ, Warszawa 2002.