

Wojciech MIELCZAREK, Ryszard PAWŁOWSKI  
Politechnika Śląska, Instytut Informatyki

## MODEL URZĄDZENIA SCPI I JEGO IMPLEMENTACJA W JĘZYKU C

**Streszczenie.** W publikacji przedstawiono model urządzenia kompatybilnego ze standardami IEEE-488.2 i SCPI oraz oprogramowanie stanowiące implementację tego modelu w języku C. W modelu urządzenia uwzględniono: protokół wymiany komunikatów, interpreter rozkazów, system statusowy, mechanizmy synchronizacji z aplikacją sterującą oraz wyzwalanie operacji.

## THE SCPI INSTRUMENT MODEL AND ITS IMPLEMENTATION IN C

**Summary.** The paper presents the model of the instrument compatible with the IEEE-488.2 and SCPI standards, as well as the C program that implements this model. In the model are represented: message exchange protocol, command interpreter, status system, message synchronization techniques and operation triggering.

### 1. Wprowadzenie

Urządzenia pomiarowe i kontrolne wyposażane są w interfejsy umożliwiające zdalne sterowanie i odczyt danych w ramach systemu ATE (Automatic Test Equipment). Na poziomie połączenia fizycznego stosowane są najczęściej łącza znakowe RS-232 i GPIB, natomiast strona logiczna komunikacji zgodna jest ze standardami IEEE-488.2 oraz SCPI (Standard Commands for Programmable Instruments). Standard IEEE-488.2 określił:

- protokół wymiany komunikatów pomiędzy kontrolerem a urządzeniem,
- synchronizację działania kontrolera i urządzenia w celu zapewnienia ich poprawnej interakcji,
- strukturę systemu statusowego w urządzeniu i sposób jego obsługi,

- zestaw tzw. rozkazów wspólnych (common commands), które kontrolują podstawowe funkcje zarządzające przyrządem.

Standard SCPI unormował treść komunikatów przepływających pomiędzy kontrolerem a urządzeniem, co pozwala go uważać za język kontroli urządzeń.

W standardzie SCPI określono budowę i reguły syntaktyczne komunikatów programowych wysyłanych z kontrolera do urządzenia i odpowiedzi wysyłanych z urządzenia do kontrolera, a ponadto rozwinięto szereg elementów występujących już w normie IEEE-488.2, jak system statusowy urządzenia i obsługujące go rozkazy, rodzaje parametrów w komunikatach programowych oraz klasy urządzeń. Zdecydowaną większość normy SCPI zajmują rozkazy sterujące różnymi urządzeniami wykonawczymi. Istotne jest jednak, że te same funkcjonalności w różnych urządzeniach są kontrolowane przez te same rozkazy oraz że różni producenci tych samych funkcjonalnie urządzeń SCPI muszą implementować te same rozkazy do kontroli określonych funkcjonalności.

Standardy IEEE-488.2 i SCPI są ze sobą ściśle związane. Urządzenie zgodne z normą SCPI musi odpowiadać również wszystkim wymaganiom stawianym przez IEEE-488.2. W dalszej części artykułu takie urządzenie będziemy nazywać „urządzeniem SCPI”.

## 2. Elementy modelu urządzenia SCPI

Urządzenie SCPI musi odpowiadać pewnemu modelowi zdefiniowanemu w standardzie SCPI. Na model ten składa się: protokół wymiany komunikatów, model funkcjonalny, system statusowy oraz „mechanizmy” synchronizacji wykonywania rozkazów.

### 2.1. Protokół wymiany komunikatów

Zadaniem protokołu wymiany komunikatów jest zapewnienie poprawnego transferu wiadomości pomiędzy kontrolerem a urządzeniem i określenie reakcji urządzenia na naruszenia reguł protokołu, tak aby było ono w każdym przypadku gotowe do działania i w ściśle określonym stanie.

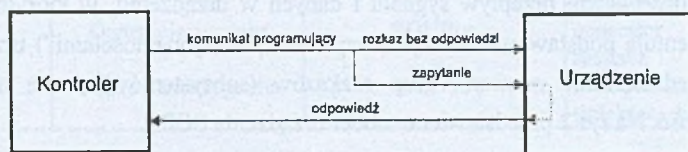
Najogólniej rzecz biorąc komunikacja pomiędzy kontrolerem a urządzeniem polega na:

- wysyłaniu do urządzenia komunikatów programujących (program messages), które można podzielić na rozkazy bez odpowiedzi (tzw. commands) i zapytania (queries),
- odbieraniu odpowiedzi z urządzenia na wysłane zapytania.

Rozkazy bez odpowiedzi inicjują jakąś akcję w urządzeniu lub ustawiają jego parametry, z czym nie wiąże się wysłanie żadnego komunikatu „zwrotnego” z urządzenia do kontrolera. Zapytania są natomiast komunikatami programującymi, na które urządzenie musi odesłać

odpowieź. W ten sposób można zażądać danych od urządzenia, jak np. wyników pomiarów, informacji o stanie przyrządu, wartości aktualnie ustawionych parametrów itp.

Na rys.1 przedstawiono zasadę wymiany komunikatów pomiędzy kontrolerem a urządzeniem.



Rys.1. Wymiana komunikatów pomiędzy kontrolerem a urządzeniem  
Fig.1. The message exchange between a device and the controller

Zapytania są wyróżnione znakiem '?' umieszczonym na końcu rozkazu i podlegają w urządzeniu innemu przetwarzaniu niż rozkazy bez odpowiedzi. Nadzór nad zgodnym z protokołem wykonaniem rozkazów i właściwym zachowaniem się przyrządu sprawuje sterowanie wymianą komunikatów.

Podstawowe zasady komunikacji kontrolera z urządzeniem są następujące:

1. Rozkazy są wykonywane przez urządzenie zgodnie z kolejnością ich wprowadzenia do przyrządu, poza nielicznymi ściśle określonymi wyjątkami.
2. Urządzenie jest zobowiązane odesłać odpowiedź tylko na wcześniej odebrane zapytanie (inaczej mówiąc nie zapytane urządzenie nie może wysłać odpowiedzi).
3. Odpowiedzi na zapytania są odsyłane w kolejności odpowiadającej odebranym zapytaniom.

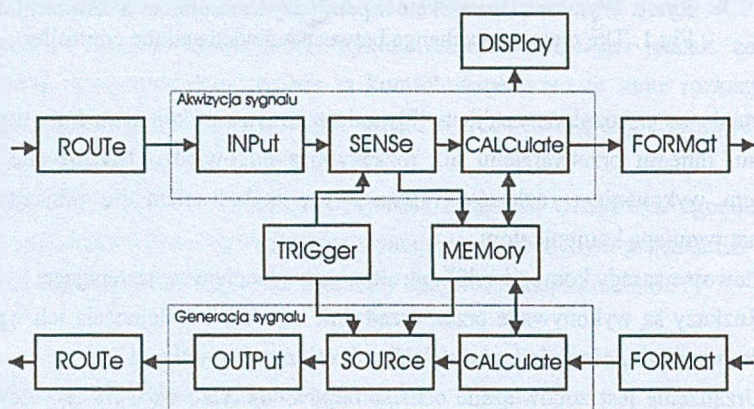
W protokole komunikacyjnym obowiązuje ponadto kilka prostych reguł:

- Reguła 1: Kontroler nie może odebrać odpowiedzi na zadane zapytanie, zanim nie zakończy rozpoczętego komunikatu programowego.
- Reguła 2: Kontroler nie może wysłać kolejnego komunikatu programowego, jeżeli nie odebrał w całości wszystkich odpowiedzi na zadane wcześniej zapytania.
- Reguła 3: Kontroler nie może podejmować próby odczytu odpowiedzi z urządzenia, jeżeli wcześniej nie wysłał żadnego zapytania.
- Reguła 4: Zapytanie powodujące odesłanie odpowiedzi o nieokreślonym rozmiarze musi być umieszczone jako ostatnie zapytanie w komunikacie programowym zawierającym kilka zapytań.
- Reguła 5: Aby uniknąć zablokowania przyrządu, zaleca się, by długość komunikatów programowych nie przekraczała rozmiaru bufora wejściowego.

Protokół wymiany komunikatów precyzyjnie określa zachowanie się urządzenia w przypadku naruszenia obowiązujących reguł komunikacyjnych (błąd zapytania), jak również w przypadku wystąpienia błędu syntaktycznego lub użycia nielegalnego rozkazu (błąd rozkazu).

## 2.2. Model funkcjonalny

Programowanie urządzenia kompatybilnego ze standardem SCPI odbywa się za pomocą rozkazów ściśle związanych z modelem urządzenia SCPI. Model ten jest schematem blokowym ilustrującym przepływ sygnału i danych w urządzeniu, w którym poszczególne bloki reprezentują podstawowe funkcje (nazywane „funkcjonalnościami”) urządzenia, a ich nazwy są jednocześnie nazwami grup rozkazów (subsystemów), które obsługują daną funkcjonalność. Na rys.2 przedstawiono model urządzenia SCPI.



Rys. 2. Model urządzenia SCPI

Fig. 2. The SCPI device model

W modelu urządzenia SCPI wyróżniono dwie części: akwizycję sygnału i generację sygnału, które określają dwa podstawowe typy urządzeń. Akwizycja sygnału jest pierwotną funkcją urządzenia typu „czujnik” (sense instrument), natomiast generacja sygnału jest pierwotną funkcją urządzenia typu „źródło” (source instrument). Trzecim typem urządzenia jest „przełącznik” (switch instrument), którego pierwotną funkcją jest routowanie sygnału.

W tabeli 1 zestawiono podstawowe typy urządzeń i podano ich przykłady. Bloki funkcjonalne określają przetwarzanie sygnału (lub danych) podczas jego przepływu przez urządzenie. Dla urządzenia typu „czujnik” funkcja ROUTe steruje układem podłączającym sygnał wejściowy do modułu akwizycji, funkcja INPut przekształcaniem sygnału (wzmianiem/tłumieniem, filtracją itp.), funkcja SENSE konwersją analogowo- cyfrową, funkcja

Tabela 1

## Podstawowe typy urządzeń

Typ urządzenia	Funkcja pierwotna	Podstawowy subsystem	Przykład
Czujnik	Akwizycja sygnału	SENSe	woltomierz, licznik, oscyloskop
Źródło	Generacja sygnału	SOURce	generator impulsów, zasilacz
Przełącznik	Routowanie sygnału	ROUTE	multiplexer, demultiplexer, skaner

CALCulate przetwarzaniem matematycznym wyników, funkcja DISPLAY prezentacją wyników, a FORMat ich przekształcaniem do formatu dogodnego do transmisji przez interfejs. Dla urządzenia typu „źródło” funkcja FORMat konwertuje dane z formatu obowiązującego na łączu interfejsowym do formatu dogodnego do przetwarzania w urządzeniu, funkcja CALCulate dokonuje wstępnego przetworzenia matematycznego danych, funkcja SOURce jest właściwym generatorem sygnału, funkcja OUTPut przekształca generowany sygnał (wzmocnia/tłumi, filtruje itp.), a ROUTe odpowiada za jego wyprowadzenie do odpowiednich zacisków wyjściowych. Oprócz wymienionych bloków, w modelu urządzenia występują jeszcze: funkcjonalność TRIGger zapewniająca synchronizację akwizycji lub generacji sygnału ze zdarzeniami zewnętrznymi lub wewnętrznymi oraz MEMory umożliwiające przechowanie (zapamiętanie) danych.

W modelu urządzenia SCPI występują tylko podstawowe funkcjonalności. W standardzie SCPI określono znacznie więcej subsystemów, które umożliwiają na przykład: ogólne zarządzanie urządzeniem (SYSTem), testowanie sprawności urządzenia (TEST), diagnostykę (DIAGnostic), kontrolę stanu (STATus), kalibrację (CALibration) itp.

### 2.3. System statusowy

W standardzie SCPI urządzenie przechowuje informację o stanie w tzw. „systemie rejestrów statusu”. System rejestrów statusu ma strukturę hierarchiczną, złożoną z pięciu grup rejestrowych:

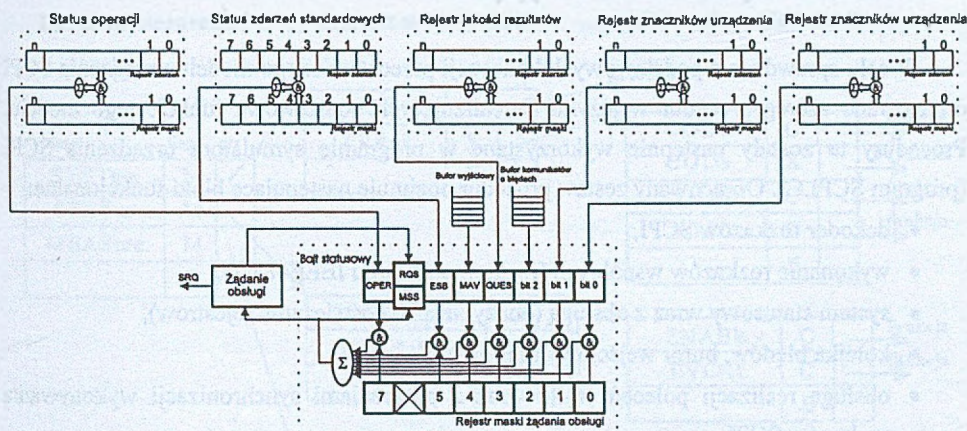
1. Bajt statusowy (Status Byte) – grupa umieszczona na najwyższym poziomie hierarchii, udostępniająca uogólnioną („zagregatyzowaną”) informację o stanie urządzenia. Znaczniki występujące w bajcie statusowym informują, czy nastąpiła zmiana stanu urządzenia oraz w której grupie rejestrowej (na niższym poziomie hierarchii) należy szukać informacji szczegółowej o wydarzeniu, które zmieniło stan przyrządu. Bajt statusowy występuje w systemie rejestrów statusu każdego urządzenia SCPI.

2. Rejestr zdarzeń standardowych (Standard Event Status) – 8-bitowy rejestr umieszczony na drugim poziomie hierarchii, informujący o wystąpieniu zdarzeń uważanych za „typowe” dla urządzenia SCPI. Są to zdarzenia niezależne od rodzaju i funkcji urządzenia i dlatego rejestr zdarzeń standardowych jest taki sam we wszystkich urządzeniach SCPI i musi wystąpić w ich systemie rejestrów statusu.
3. Rejestr znaczników urządzenia (Questionable Status) – 16-bitowy rejestr umieszczony na drugim poziomie hierarchii, informujący o „jakości” wyników pomiarów i sprawności ważniejszych układów w przyrządzie. Organizacja bitowa rejestru zależy od rodzaju urządzenia. Jeżeli wynik pomiaru jest „wątpliwy” (questionable), na przykład na skutek przekroczenia zakresu, to związany z nim bit zostaje ustawiony. W prostych przyrządach SCPI rejestr znaczników urządzenia może nie występować, natomiast w przyrządach złożonych z wymienionym rejestrem może być związanych nawet kilka kolejnych rejestrów na niższym poziomie hierarchii.
4. Rejestr operacji (Operation Status) – 16-bitowy rejestr umieszczony na drugim poziomie hierarchii informujący o stanie wykonywanych przez przyrząd operacji. Rejestr ten nie musi wystąpić w systemie rejestrów statusu urządzenia SCPI.
5. Rejestr statusu zależny od urządzenia (Device Dependent Status) – rejestr specyficzny dla urządzenia, co oznacza, że liczba i znaczenie bitów w rejestrze są ściśle związane z konkretnym urządzeniem. Rejestr ten nie musi wystąpić w systemie rejestrów statusu urządzenia SCPI.

Każda grupa rejestrowa składa się z rejestru zdarzeń i rejestru maski. Grupy rejestrowe na niższym poziomie hierarchii reprezentowane są przez odpowiadające im bity „zbiorcze” (summary bits) w rejestrze zbiorczym na wyższym poziomie. I tak na przykład bajt statusowy jest rejestrem zbiorczym dla grup: zdarzenie standardowe, jakość rezultatów, operacja i znaczniki urządzenia, co oznacza, że każda z wymienionych grup rejestrowych jest reprezentowana w bajcie statusowym przez przyporządkowany jej bit zbiorczy. Bit zbiorczy powstaje poprzez zsumowanie logiczne („OR”) nie zamaskowanych bitów rejestru zdarzeń danej grupy rejestrowej. Organizację systemu statusowego przedstawiono na rys.3.

#### 2.4. Synchronizacja wykonywania rozkazów

Sterowanie i akwizycja danych wymagają niejednokrotnie wykorzystania wielu urządzeń połączonych w system pracujący pod nadzorem aplikacji uruchomionej w kontrolerze. Sprawą zasadniczą jest zapewnienie właściwej sekwencji operacji urządzeń w systemie, co jest zadaniem synchronizacji. W języku SCPI rolę „mechanizmów” synchronizacyjnych pełnią trzy rozkazy: \*OPC, \*OPC? i \*WAI.



Rys.3. System statusowy urządzenia SCPI

Fig.3. The SCPI status system

Rozkazy wykonywane przez urządzenia SCPI należą do jednej z dwóch kategorii: rozkazy sekwencyjne i rozkazy nakładające się. Rozkazy sekwencyjne wykonywane są jeden po drugim w kolejności ich wprowadzenia do przyrządu i wykonanie rozkazu następnego rozpoczyna się dopiero po zakończeniu wykonania rozkazu poprzedniego. W przypadku rozkazów nakładających się rozpoczęcie wykonywania rozkazu następnego nie jest uzależnione od zakończenia wykonania rozkazów go poprzedzających, co umożliwia jednocześnie wykonywanie kilku rozkazów.

Rozkaz \*OPC (Operation Complete) powoduje ustawienie na 1 bitu OPC (bit 0) w rejestrze zdarzeń standardowych, ale tylko wtedy, gdy wszystkie rozpoczęte operacje urządzenia zostały zakończone. Testowanie bitu OPC po wysłaniu rozkazu \*OPC pozwala wykryć stan, w którym urządzenie zakończyło wykonanie wszystkich rozkazów wysłanych do urządzenia przed rozkazem \*OPC i nie jest aktualnie zajęte wykonywaniem żadnej operacji.

Rozkaz \*OPC? powoduje wprowadzenie do bufora wyjściowego znaku ASCII '1' (31hex) wtedy, gdy wszystkie operacje urządzenia zostały zakończone (no-pending flag = 1). \*OPC? należy umieszczać na końcu ciągu rozkazowego (terminowanego), którego wykonanie zamierzamy sprawdzić.

Rozkaz \*WAI (Wait) powoduje, że następujące za nim rozkazy nie zostaną wykonane przed zakończeniem wykonania rozkazów go poprzedzających. W ten sposób można zagwarantować, że pewne operacje (te po rozkazie \*WAI) nie rozpoczną się, zanim nie zakończą się wszystkie operacje już wykonywane (związane z rozkazami wysłanymi przed poleceniem \*WAI).

### 3. Model urządzenia SCPI w języku C

W celu sprawdzenia podstawowych koncepcji przedstawionego modelu urządzenia SCPI opracowano szereg procedur w języku C realizujących podstawowe funkcje tego modelu. Procedury te zostały następnie wykorzystane w programie symulatora urządzenia SCPI (program SCPI.C). Opracowany zestaw procedur obejmuje następujące bloki funkcjonalne:

- dekodér rozkazów SCPI,
- wykonanie rozkazów wspólnych (rozkazy standardu IEEE488.2),
- system statusowy wraz z obsługą (odczyt, zapis, odświeżanie rejestrów),
- kolejka błędów, bufor wejściowy i bufor wyjściowy,
- obsługa realizacji poleceń w tle wraz z poleceniami synchronizacji wykonywania rozkazów SCPI,
- wykonanie kilku wybranych rozkazów języka SCPI.

Program symulatora po uruchomieniu wyświetla: informację o wersji, stan rejestru statusowego (odświeżany po każdej zmianie stanu), bufor wyjściowy (wyprowadzany po wykonaniu każdego polecenia) oraz obszar przeznaczony na komunikaty różnych poleceń SCPI. Wprowadzone przez użytkownika polecenie w języku SCPI jest wyświetlane w dolnej części ekranu.

Drzewo aktualnie zaimplementowanych poleceń SCPI zostało zapisane w postaci powiązanych ze sobą tablic przedstawionych schematycznie na rysunku 4. Każdy element tablicy zawiera opis pojedynczego polecenia SCPI, znajdującego się na określonym poziomie hierarchicznej struktury rozkazów języka SCPI. Może to być zarówno rozkaz powiązany z określoną akcją interpretera, jak i element typu podmenu powiązany z tablicą rozkazów znajdujących się na niższym poziomie hierarchii. Przykładem elementu pierwszego rodzaju będzie element opisujący polecenie VPP, znajdujący się na drugim poziomie hierarchii w tablicy grupującej rozkazy podsystemu MEASure. Element opisujący polecenie STATus, znajdujący się na I poziomie hierarchii w tablicy opisującej polecenia SCPI, jest przykładem elementu typu podmenu. Jest on powiązany z tablicą zawierającą elementy opisujące polecenia QUESTionable i OPERation. Te elementy z kolei też są elementami typu podmenu powiązanymi ze „swoimi” tablicami. W ten sposób szereg powiązanych ze sobą tablic ostatecznie opisuje fragment poddrzewa rozkazów języka SCPI z przykładowymi poleceniami:

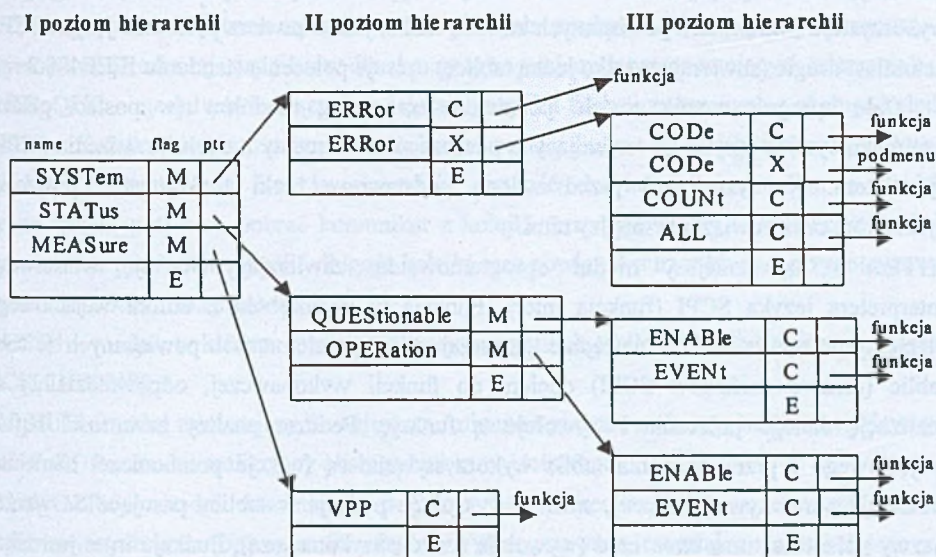
:STATus:QUESTionable:ENABLE

:STATus:QUESTionable:EVENT

:STATus:OPERation:ENABLE

:STATus:OPERation:EVENT





Rys.4. Struktura przykładowego drzewa rozkazów języka SCPI  
 Fig.4. The structure of the SCPI commands tree

Element opisujący pojedynczy rozkaz SCPI (właściwe polecenie lub podmenu) zawiera następujące pola:

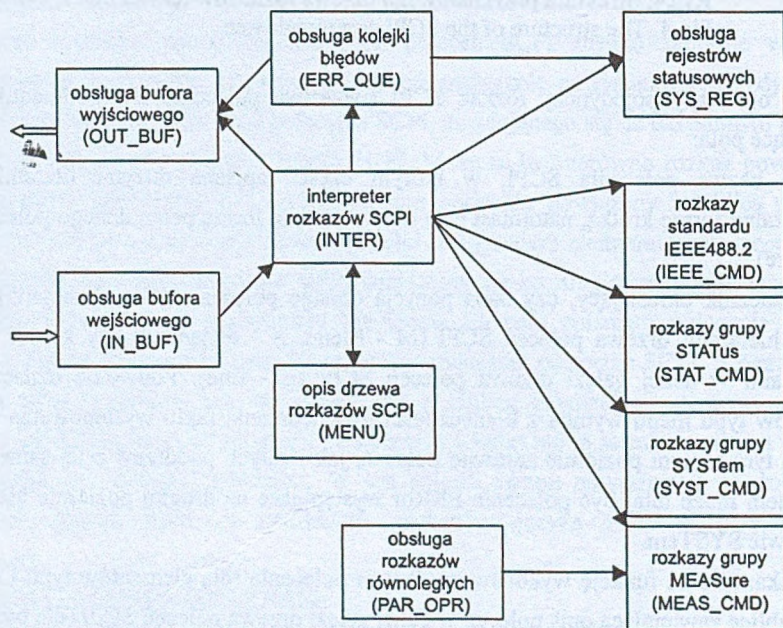
- **name** nazwa polecenia SCPI, w którym część zapisana dużymi literami określa dopuszczalną formę krótką, natomiast cała nazwa określa formę pełną danego polecenia (np. MEASure).
- **flag** znacznik określający, czy dana pozycja opisuje polecenie (C - Command), kolejny poziom hierarchii drzewa poleceń SCPI (M - Menu, X - eXtended) czy koniec tablicy z poleceniami w danej gałęzi drzewa poleceń SCPI (E - End). Podwójne oznaczenie dla elementów typu menu wynika z konieczności uwzględnienia faktu występowania w języku SCPI na tym samym poziomie zarówno poleceń, jak i całych poddrzew o tej samej nazwie. Przykładem może tutaj być polecenie ERRor występujące na drugim poziomie hierarchii w poddrzewie SYSTem.
- **ptr** wskazanie na funkcję wykonawczą danego polecenia (dla elementów typu Command) lub na tablicę zawierającą opis poleceń w danej gałęzi drzewa poleceń SCPI (dla pozycji typu Menu lub eXtended).

Przedstawione drzewo poleceń SCPI zaimplementowane w programie symulatora nie obejmuje poleceń na najwyższym poziomie hierarchii (poleceń: SENSE, SOURce oraz

ROUTE), które są opcjonalne dla określonych typów urządzeń pomiarowych. Program wykorzystuje dwa drzewa powiązanych ze sobą tablic: jedno zawiera polecenia języka SCPI, natomiast drugie (zawierające tylko jedną tablicę) opisuje polecenia standardu IEEE488.2.

Całe oprogramowanie zostało podzielone na szereg modułów (w postaci plików nagłówkowych w języku C) realizujących poszczególne elementy modelu urządzenia SCPI i symulatora. Na rysunku 5 przedstawiono podstawowe bloki funkcjonalne programu symulatora oraz powiązania między nimi.

**INTER** - najważniejszy moduł oprogramowania, zawierający funkcję właściwego interpretera języka SCPI (funkcja `inter`). Funkcja ta wyodrębnia z bufora wejściowego elementy polecenia SCPI, a następnie „poruszając się” po elementach powiązanych ze sobą tablic (drzewo rozkazów SCPI) dociera do funkcji wykonawczej, odpowiedzialnej za realizację danego polecenia i wywołuje tę funkcję. Podczas analizy zawartości bufora wejściowego i przeszukiwania tablic wykorzystywane są funkcje pomocnicze: `namecmp` (realizuje porównywanie nazw), `smenu` (wyszukuje pozycje w tablicy pasujące do wzorca nazwy polecenia) oraz `exec_cmd` (wywołuje funkcję wykonawczą). Funkcja `inter` jest także odpowiedzialna za realizację protokołu wymiany komunikatów zdefiniowanego w normie IEEE 488.2.



Rys.5. Struktura oraz powiązania pomiędzy modułami programu symulatora  
Fig.5. The structure and binding of the simulator's modules

**STAT\_REG** – moduł, w którym zebrano deklaracje zmiennych reprezentujących rejestry wchodzące w skład systemu rejestrów standardowych urządzenia SCPI (rejestry statusowe, rejestry masek oraz rejestry filtrów dla zbroczy wyzwalających reprezentujących zdarzenia).

**ERR\_QUE** – moduł zawierający definicje zmiennych i funkcji związanych z obsługą kolejki błędów. Kolejka błędów zorganizowana jest w postaci bufora FIFO i przechowuje numery błędów. Przygotowane funkcje pozwalają wstawić do kolejki komunikat o błędzie (`error_queue_put`) oraz pobrać komunikat z kolejki (`error_queue_get`). Dodatkowe funkcje pomocnicze pozwalają określić długość kolejki oraz powiązać numer błędu z przygotowanym wcześniej tekstem komentarza.

**MENU** – moduł, w którym zebrano definicje wszystkich tablic opisujących drzewa poleceń SCPI i IEEE488.2.

**IN\_BUF** – moduł zawierający funkcje oraz zmienne związane z obsługą bufora wejściowego.

**OUT\_BUF** – moduł realizujący obsługę bufora wyjściowego.

**PAR\_OPR** – moduł zawierający definicje funkcji realizujących mechanizm poleceń wykonywanych w tle (operacje równoległe). Wykorzystywana jest tutaj zmienna pełniąca rolę semafora. Funkcje wpływające na stan semafora to: `parallel_operation_start`, która powinna być wywoływana w momencie rozpoczęcia wykonywania jakiejś operacji w tle oraz `parallel_operation_stop`, która powinna z kolei kończyć tego rodzaju operację. Pomiędzy wywołaniami tych funkcji oprogramowanie (funkcja obsługująca odczyt polecenia SCPI do bufora wejściowego) wywołuje zarejestrowaną w funkcji `parallel_operation_start` funkcję wykonawczą. Funkcja ta działając na zasadzie „maszyny stanów” powinna symulować realizację operacji w tle. W ramach przygotowanego symulatora zaimplementowano przykładowy zestaw funkcji realizujących operację równoległą w celu przetestowania oprogramowania (niestandardowy rozkaz `:MEASure:VPP` w module `MEAS_CMD`).

Kolejne moduły zawierają funkcje wykonawcze dla przykładowych poleceń języka SCPI (oraz IEEE488.2):

**IEEE\_CMD** - moduł zawierający definicje funkcji realizujących polecenia IEEE488.2. W module zaimplementowano funkcje realizujące polecenia: `*CLS`, `*ESE`, `*ESE?`, `*ESR?`, `*IDN?`, `*OPC`, `*OPC?`, `*SRE`, `*SRE?`, `*STB?`, `*WAI`, `*RST` oraz `*TST?`.

**SYST\_CMD** – moduł, w którym zawarto definicje funkcji realizujących polecenia z grupy `:SYSTem`. Dotychczas opracowano funkcje realizujące polecenia:

`:SYSTem:ERRor?`, `:SYSTem:ERRor:COUNT?`, `:SYSTem:ERRor:ALL?`,

`:SYSTem:ERRor:CODE?` oraz `:SYSTem:ERRor:CODE:ALL?`.

**STAT\_CMD** – moduł zawierający definicje funkcji realizujących wybrane polecenia grupy `:STATus`. Obecnie zaimplementowano zestaw funkcji dla rejestrów `QUESTionable` i `OPERation`. Dodatkowo w module tym znalazła się definicja bardzo ważnej funkcji o nazwie

status\_generate, odpowiedzialnej za odświeżenie rejestru zbiorczego w bajcie statusowym na podstawie stanu pozostałych rejestrów systemu statusowego.

MEAS\_CMD – moduł, w którym zawarto przykładowe funkcje wykonawcze dla polecenia :MEASure:VPP, demonstrujący mechanizm pracy równoległej.

#### 4. Podsumowanie

Praca urządzenia pomiarowo-kontrolnego w systemie ATE wymaga wyposażenia go w interfejs umożliwiający połączenie do kontrolera oraz w stosowne oprogramowanie komunikacyjne. Realizacja zdalnych poleceń i odsyłanie odpowiedzi wymagają z kolei interpretacji i wykonania rozkazów, kontroli stanu urządzenia, formatowania odpowiedzi i zarządzania buforem wyjściowym, przez który urządzenie wysyła komunikaty do kontrolera. Bardzo ważne jest wykonywanie wymienionych operacji zgodnie z obowiązującymi standardami, tak aby sterowanie urządzeniami pomiarowymi było jednolite. Standardy obowiązujące w systemach ATE to: IEEE-488.2 oraz SCPI. We wprowadzeniu do niniejszej publikacji krótko przedstawiono funkcjonalności określone przez wspomniane normy.

Standardy IEEE-488.2 oraz SCPI przedstawiają pewną koncepcję urządzenia pomiarowo-kontrolnego i ilustrują ją za pomocą modelu urządzenia SCPI, który obejmuje: protokół wymiany komunikatów, model funkcjonalny, system statusowy, mechanizmy synchronizacji wykonywania rozkazów. Wymienione elementy opisano w punkcie 2.

W celu zaimplementowania modelu urządzenia SCPI opracowany został zestaw procedur w języku C realizujący podstawowe funkcje tego modelu. Procedury te zostały następnie wykorzystane w programie symulatora urządzenia SCPI (program SCPI.C), który umożliwia dokładne zrozumienie koncepcji i działania urządzenia zgodnego z normami IEEE.488.2 i SCPI. Wspomniane procedury nadają się do wykorzystania w projektowanych urządzeniach pomiarowych po ich uprzednim skompilowaniu na język procesora urządzenia. Urządzenie z zaimplementowanymi procedurami modelu SCPI opisanymi w rozdziale 3 publikacji można uważać za kompatybilne z normami IEEE-488.2 i SCPI.

## LITERATURA

1. IEEE Standard 488.2-1987: IEEE Standard Codes, Formats, Protocols and Common Commands", Institute of Electrical and Electronics Engineers.
2. IEEE Standard 488.2-1992: IEEE Standard Codes, Formats, Protocols and Common Commands, and Standard Commands for Programmable Instruments", Institute of Electrical and Electronics Engineers.
3. Mielczarek W.: „Urządzenia i systemy kompatybilne ze standardem SCPI", Helion, Gliwice 1999. [www.helion.pl](http://www.helion.pl).
4. Mielczarek W.: „Szeregowe interfejsy cyfrowe", Helion, Gliwice 1993.
5. Pieper John M.: „Automatic Measurement Control", Rohde & Schwarz GmbH & Co. KG Member of the SCPI Consortium. [www.rsd.de](http://www.rsd.de).

Recenzent: Dr inż. Ryszard Winiarczyk

Wpłynęło do Redakcji 4 kwietnia 2002 r.

### Abstract

The communication processes between a controller and a device in ATE systems (Automatic Test Equipment) are defined by several standards, among them the IEEE-488.2 defines message exchange protocol, data structures, common commands and status system, and the SCPI (Standard Commands for Programmable Instruments) defines the instrument programming language. These two standards are closely tied together and should be seen as a basic requirement for the exchange of instruments and the portability of application programs. They also increase the compatibility among instruments.

In the paper the C model of SCPI instrument was presented. This model covers all requirements of the IEEE-488.2 and SCPI standards: message exchange protocol, program messages parsing, status registers and error reporting system, trigger control, message synchronization techniques. The set of procedures in C can be used for simulating an SCPI device for example in a PC computer, or, after compiling to any microprocessor language, as the core of the program controlling an SCPI device behavior.