

PODŁĄCZANIE MIERNIKÓW DO KOMPUTERA Z WYKORZYSTANIEM PORTU RS232C

4.1. CEL I ZAKRES ĆWICZENIA

Celem ćwiczenia jest zapoznanie się ze sposobami podłączania miernika do komputera, nawiązywania transmisji danych i zapisu uzyskanych wyników pomiarów.

4.2. WYBRANE WIADOMOŚCI PODSTAWOWE

4.2.1. RS 232

Standard RS – 232C (*Recommended Standard*) został ustanowiony w 1969 r. przez Electronic Industries Association. Definiuje on sposób nawiązania i przeprowadzenia łączności między dwoma urządzeniami tzw. DTE (*Data Terminal Equipment*) za pośrednictwem modemów tzw. DCE (*Data Communication Equipment*) lub bez nich. Jest to obecnie standard łącza szeregowego do połączenia komputera PC z urządzeniami zewnętrznymi.



Rys. 4.1. Umiejscowienie RS232 w torze transmisyjnym.

Połączenia między urządzeniami (DTE – DCE lub DTE – DTE) dokonuje się za pomocą złączy 25 – stykowych lub 9 – stykowych.

Najważniejsze linie magistrali RS– 232C podane są w tabeli 4.1.

TABELA 4.1. Opis magistrali RS-232C

Numer przewodu i styku				Opis sygnału		Kierunek
Obwód	RS-232	Styk DB25	Styk DB9	Nazwa polska	Nazwa angielska	DTE - DCE
101	PG	1	-	Masa ochronna	Protective Ground	-
102	SG	7	5	Masa sygnałowa	Signal Ground	-
103	TxD	2	3	Dane nadawane	Transmitted Data	→
104	RxD	3	2	Dane odbierane	Received Data	←
105	RTS	4	7	Żądanie nadawania	Request to Send	→
106	CTS	5	8	Gotowość do nadawania	Clear to Send	←
107	DSR	6	6	Gotowość DCE	Data Set Ready	←
108	DTR	20	4	Gotowość DTE	Data Terminal Ready	→
109	DCD	8	1	Poziom sygnału	Carrier Detected	←

Opis linii

Linie Danych (obowiązuje logika negatywna):

TxD – dane nadawane.

RxD – dane odbierane

Linie sterujące (obowiązuje logika pozytywna):

RTS – żądanie nadawania danych zgłaszane przez terminal DTE

CTS – gotowość do nadawania zgłaszana przez modem DCE (przesyła potwierdzenie odebrania sygnału RTS)

DSR – gotowość modemu DCE do dalszej współpracy z DTE (aktywny przez cały czas trwania połączenia)

DTR – gotowość DTE do dalszej współpracy z DCE (aktywny przez cały czas trwania połączenia)

DCD – sygnał wykrycia przez modem fali nośnej (oznacza, że łączy się on z innym modemem)

Linie masy:

SG – masa sygnałowa

PG – masa ochronna połączona z obudową urządzenia

Istnieje wiele sposobów wykorzystania różnych podzbiorów tych linii. W najprostszym przypadku wykorzystuje się tylko 3 linie:

TxD – dane nadawane

RxD – dane odbierane

SG – masa sygnałowa

Jeżeli łączymy urządzenie typu DTE (komputer) z urządzeniem z DCE (modem) to styki łączymy wprost. Jeżeli komputer do komputera to używamy tzw. konfiguracji *null modem*. Na przykład dla wtyków 9-cio pinowych rys. 4.2.



Rys. 4.2. Łączenie styków przy komunikacji portem szeregowym 9-cio pinowym.

Linie DTR, DSR, RTS, CTS i DCD umożliwiają synchronizację transmisji tak, dane były wysyłane zawsze wtedy, gdy urządzenie po drugiej stronie połączenia jest gotowe do ich odbioru. Prosty przykład transmisji jednokierunkowej (simplex) może wyglądać następująco (rys. 4.3.):

1. Aby transmisja była możliwa oba urządzenia muszą ustawić linie gotowości na „1”
1. Gotowość komputera (DTR) i modemu (DSR) musi być utrzymywana na „1” przez całą transmisję.
2. Komputer ustawia RTS (żądanie nadawania) na „1”
3. W odpowiedzi na RTS modem ustawia linię CTS (gotowość do nadawania) na „1”
4. Widząc to komputer rozpoczyna nadawanie

symbol	obwód	stan linii	uwagi
DTR	108/2		komputer gotów
DSR	107		modem gotów
RTS	105		żądanie nadawania
CTS	106		gotowość do nadawania
TxD	103		transmisja danych do modemu

Rys. 4.3. Przykład transmisji jednokierunkowej.

Szybkość transmisji

Szybkość transmisji szeregowej określona jest ilością bitów przesyłanych w jednostce czasu. Jednostką szybkości transmisji jest Bod: 1 Bod = 1 Bit/s. Używane są różne szybkości transmisji w zależności od sposobu realizacji transmisji, typu urządzeń i rodzaju linii transmisyjnej. Najczęściej stosowane są szybkości transmisji: 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Bodów itd. aż do kilkuset MBodów. W komputerach kompatybilnych z IBM PC stosowane są także nietypowe prędkości transmisji : 57600 i 115200 bodów, a w interfejsie muzycznym MIDI szybkość : 31250 bodów.

Parametry elektryczne sygnałów

W opisie standardu używane są pojęcia stanów logicznych sygnałów sterujących interfejsu: .ON. i .OFF. oraz linii danych: .Mark. i .Space.. ON oznacza stan aktywny, OFF nieaktywny sygnału sterującego, Mark odpowiada stanowi .Logiczne 1., a Space stanowi .Logiczne 0. przesyłanych danych. Przyjęto następujące parametry elektryczne poszczególnych stanów linii interfejsu V.24:

ON : +3V..+25V

OFF : -3V..-25V

Mark : -3V..-25V

Space: +3V..+25V

Złącza RS232:

W interfejsie RS232C wykorzystuje się złącza DB, (D-sub, D-shell) w wersji 25-stykowej lub 9-stykowej. W urządzeniach DTE powinny być montowane złącza męskie (DB-9M, DB-25M) a w urządzeniach DCE złącza żeńskie (DB-9F, DB-25F).

DTE
Złącze DB-9M



DCE
Złącze DB-9F



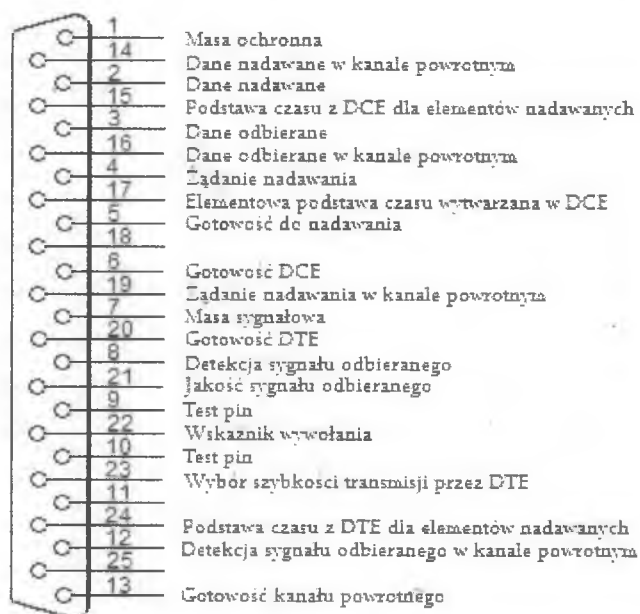
Złącze DB-25M



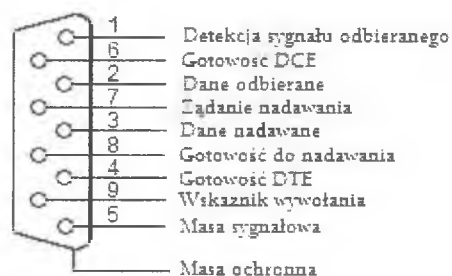
Złącze DB-25F



Opis poszczególnych sygnałów w złączach:



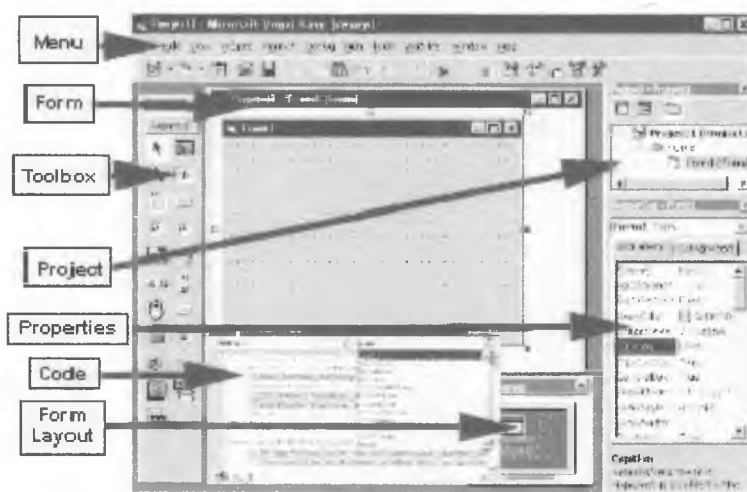
Złącze RS 232 DB25



Złącze RS 232 DB9

4.2.1. VISUAL BASIC

Dzisiejsze języki programowania znacznie się różnią od języków programowania sprzed kilku lat. Dzięki systemowi operacyjnemu Microsoft Windows, który używa graficznego interfejsu, powstało wiele narzędzi, bardziej zaawansowanych, które wykorzystują interfejs graficzny. Programy te, choć często spełniają taką samą rolę jak ich poprzednicy, są ze względu na swój wygląd, bardziej przyjazne dla użytkownika. Tworzenie programów z graficznym interfejsem i wykorzystujących zdarzenia generowane przez użytkownika stało się czasochłonne w językach strukturalnych. Wymusiło to rozwój języków programowania. Wprowadzenie wizualizacji do języków programowania znacznie uprościło proces tworzenia graficznego interfejsu aplikacji. Visual Basic jest aplikacją Windows i sam wykorzystuje interfejs graficzny. Platforma projektowa i środowisko projektowe Visual Basic nosi nazwę *Developer Studio*. Składa się ono z wielu okien i narzędzi, które pozwalają (i ułatwiają) tworzyć rozbudowane aplikacje działające w środowisku Microsoft Windows. W fazie projektowania programu, główne okno Visual Basic zawiera kilka różnych okien. Każde z nich może być ukryte w miarę potrzeby. Każde z nich ma inne zadanie i jest wykorzystywane do innych czynności niezbędnych przy tworzeniu aplikacji.



Rys. 4.4. Przykład okna aplikacji.

Pasek menu jest paskiem rozwijanych, wielopoziomych menu, z których można wybrać polecenia konieczne do ustawienia odpowiednich opcji zarówno tworzonej aplikacji, jak i samego Visual Basic.

Pasek narzędziowy **Toolbar**, jest szeregiem ikon, które tworzą skrót do odpowiednich pozycji menu. Zamiast rozwijać wielopoziomowe menu i wybierać odpowiednie elementy, można kliknąć ikonę, która spowoduje wykonanie tych samych czynności.

Okno **Form (Object)** jest głównym oknem aplikacji. W tym oknie tworzony jest cały wygląd aplikacji, rozmiar, kolor, rozmieszczenie elementów.

Okno **Project** jest oknem zawierającym informację o wszystkich częściach i plikach aktualnie tworzonej aplikacji.

Okno **Toolbox** jest zbiorem kontroltek, które można umieszczać na formie tworzonej aplikacji, aby wykonywały polecenia im zadania.

Okno **Properties** to okno, w którym można ustawić właściwości wykorzystywanych elementów, form, kontroltek, obiektów. Można tam ustawić cechy odpowiedzialne za wygląd, położenie, zachowanie poszczególnych elementów. Aby wybrać odpowiednią wartość właściwości, należy wpisać ją w odpowiednie okno lub wybrać jedną z pozycji rozwijanej listy.

Okno **Form Layout** pozwala określić położenie form na ekranie i sprawdzić położenie form przy innej rozdzielczości monitora.

Okno **Code** jest oknem edytora tekstu, w którym można wstawić wykonywalny kod programu. Kod ten to zbiór dostępnych rozkazów, które zostaną wykonane, gdy nastąpi określone zdarzenie, które spowoduje wykonanie odpowiedniej procedury i zawartej w niej kodu.

Okno **Property Page** jest oknem, które pozwala na stworzenie łatwej formy dostępu przez użytkownika do tworzonego elementu lub ułatwia ustawienie parametrów wykorzystywanych obiektów.

Okno **Components** jest oknem, które pozwala dołączyć dodatkowe elementy (oprócz standardowych elementów Visual Basic) do aplikacji.

Okno **Object Browser** pomaga używać obiekty stworzone w różnych aplikacjach.

Po zaprojektowaniu aplikacji, środowisko Visual Basic, pozwala wykonać aplikację, śledząc jej przebieg. Można sprawdzać działanie aplikacji, zachowanie obiektów, kontrolować wykonywanie kodu programu, sprawdzać wartości zmiennych. Dodatkowe okna platformy Visual Basic dostępne są tylko w fazie wykonywania programu i pozwalają kontrolować i zmieniać stan aplikacji.

Okno **Call Stack** wyświetla listę wszystkich działających procedur.

Okno **Immediate** pozwala wykonywać dodatkowe rozkazy, zmieniać wartości zmiennych, właściwości itp.

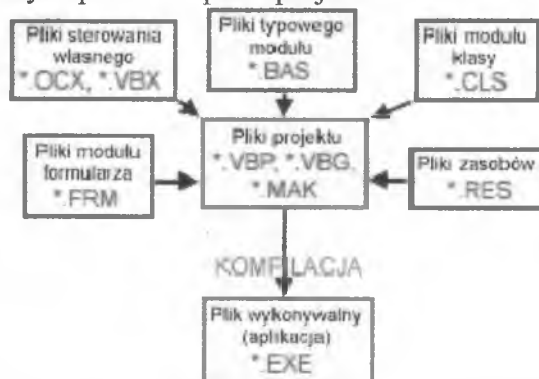
Okno **Locals** wyświetla wartość wszystkich zadeklarowanych zmiennych w bieżącej procedurze. Okno **Watch** pozwala śledzić stan poszczególnych zmiennych lub wyrażeń.

Wszystkie te elementy należą do platformy *Developer Studio*. Pomagają one tworzyć aplikację. To jak zostaną wykorzystane i jaką formę przybierze tworzona aplikacja zależy od programisty.

Pliki projektu

Przed rozpoczęciem tworzenia nowego projektu, dobrze jest utworzyć folder, w którym będą przechowywane wszystkie pliki projektu. Mogą one być przechowywane w różnych miejscach, ale umiejscowienie ich w jednym folderze ułatwia nawigację i zarządzanie plikami. Część plików projektu, jak pliki sterujące czy pliki modułów ogólnych, które mogą pracować w trybie współużytkowania, przechowywana jest w innych folderach.

Przy tworzeniu aplikacji, Visual Basic automatycznie tworzy główny plik projektu. Oprócz niego do projektu dołączany jest szereg innych plików, wchodzących w skład tworzonej aplikacji. Na przykład każde tworzone okno aplikacji, przechowywane jest w osobnym pliku formularza, a zasoby przechowywane są w osobnych plikach zasobów. Visual Basic automatycznie zapamiętuje położenie wszystkich niezbędnych plików w pliku projektu.



Plik (rozszerzenie)	Przeznaczenie i położenie pliku
Plik projektu (.VBP, .MAK, lub .VBG)	Zawiera nazwy wszystkich plików wykorzystywanych w projekcie i ich położenie. Zwykle przechowywany jest w folderze projektu
Plik modułu-formularza (.FRM)	Zawiera informacje o formularzu, zawartych na nim obiektach i kod programu związanym z tym formularzem lub tymi obiektami. Zwykle przechowywany jest w folderze projektu
Plik elementów sterujących (.OCX, lub .VBX)	Zawiera jeden lub kilka elementów sterujących. Przechowywany jest zwykle w podfolderze SYSTEM folderu WINDOWS i może być używany przez wiele różnych projektów
Plik typowego modułu (.BAS)	Plik zawiera kod programu, który jest dostępny dla wszystkich modułów projektu. Zwykle jest przechowywany w folderze projektu, choć może być przechowywany gdzie indziej i dostępny dla wielu projektów
Plik modułu klasy (.CLS)	Plik zawiera definicję klasy. Przechowywany jest tak jak pliki typowego modułu
Plik zasobów (.RES)	Zawiera dane projektu, teksty, obrazy, itp. Zwykle jest przechowywany w folderze projektu

Tworzenie aplikacji

Tworzenie aplikacji w Visual Basic składa się z wykonywania trzech podstawowych kroków:

- ustawienie interfejsu (wyglądu) aplikacji

- ustawienie właściwości
- pisanie kodu programu

Visual Basic jest narzędziem programistycznym, które umożliwia szybkie i łatwe tworzenie i modyfikowanie aplikacji. Zwykle wykonuje się wielokrotnie trzy powyższe kroki, aby sprawdzić i dopasować wygląd i zachowanie aplikacji.

Można dodać element sterujący do formularza, ustawić jego właściwości, dodać odpowiedni kod i uruchomić aplikację, aby sprawdzić jej działanie. Jeżeli trzeba zmienić działanie lub wygląd można zmienić właściwości i kod programu, po czym, sprawdzić działanie wprowadzonych zmian. Potem można dodać kolejny element sterujący i powtórzyć czynności. Można również dodać wszystkie potrzebne elementy interfejsu, ustawić wszystkie właściwości i dodać cały kod programu i dopiero wtedy sprawdzić działanie aplikacji.

Pierwszym działaniem przy tworzeniu aplikacji w Visual Basic jest ustawienie interfejsu. Typowe aplikacje działające pod kontrolą systemu Microsoft Windows mają wiele typowych i charakterystycznych cech i elementów. Standaryzacja interfejsów użytkownika Windows powoduje łatwość uczenia się działania aplikacji Windows. Ta standaryzacja ułatwia również zadanie programistom, ponieważ nie zachodzi konieczność definiowania nowego interfejsu dla każdej tworzonej aplikacji. Przy tworzeniu aplikacji w Visual Basic, większość pracy polegającej na projektowaniu interfejsu użytkownika jest już wykonana lub bardzo uproszczona. Formularze i elementy sterujące pozwalają na szybkie i łatwe dodawanie typowych składników interfejsu aplikacji. Elementy sterujące, zwane kontrolkami, umożliwiają stosowanie typowych składników aplikacji systemu Windows, wyglądających i zachowujących się zgodnie z ich przeznaczeniem. Elementy te mają zdefiniowane pewne typowe cechy i zachowania, np. okna tekstowe umożliwiają wprowadzanie tekstu i mają migający wskaźnik a przyciski poleceń mają charakterystyczny prostokątny wygląd i tekst identyfikujący działanie. Żadna ze zdefiniowanych cech nie wymaga pisania kodu programu, a jedynie wybrania lub ustawienia odpowiednich właściwości (większość z nich jest ustawiona na domyślne wartości).

Aby jednak te elementy wykonywały określone zadania należy dopisać kod programu.

Dodawanie kodu programu

Umieszczając na formularzach kontrolki i nadając im odpowiednie właściwości, można tworzyć efektowne aplikacje. Jednak nie będą one w pełni funkcjonalne, ponieważ kontrolki standartowo wykonują tylko podstawowe zadania. Np. okienko tekstowe pozwala wprowadzać i zaznaczać tekst a przycisk poleceń może wyglądać jak wciśnięty lub wyciśnięty, ale nie potrafi zakończyć aplikacji (nawet jeżeli takie jest jego docelowe znaczenie) dopóki nie zostanie dopisany odpowiedni kod programu, który to wykona. Funkcjonalność kontrolek zależy od programisty. To on musi stworzyć kod programu, który odpowiednio zareaguje na jakieś zachowanie użytkownika. Kod programu jest zbiorem instrukcji odpowiedzialnych za przetwarzanie danych, przyjmowania ich, wyprowadzanie i komunikowanie się z użytkownikiem.

Aby wyświetlić kod programu dla obiektu należy:

- kliknąć dwukrotnie na obiekcie
- wybrać obiekt i wcisnąć F7
- wybrać obiekt i wybrać pozycję *Code* z menu View
- wybrać obiekt i kliknąć przycisk *View Code* w oknie Project

Visual Basic zawsze pokazuje kod programu w oknie działającym jak edytor tekstu. Można w nim dokonywać podstawowych operacji takich jak wstawianie, usuwanie, kopiowanie, wklejanie, przenoszenie tekstu. Mimo, że Visual Basic jest raczej środowiskiem graficznym, wprowadzany kod programu jest tekstem.

4.3. PRZEBIEG ĆWICZENIA

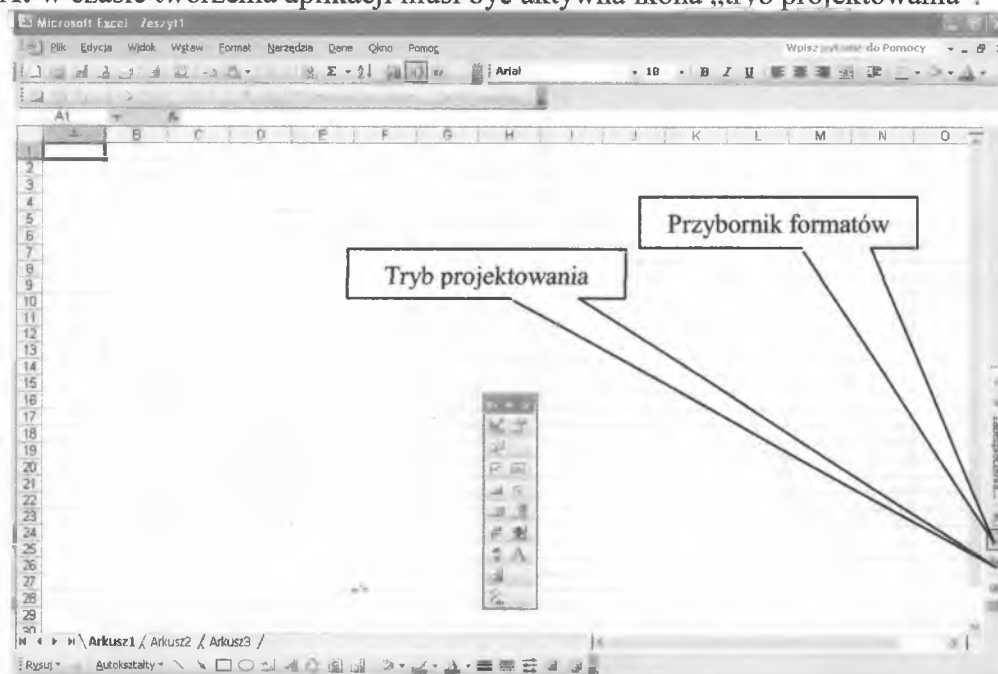
4.3.1. APLIKACJA W VISUAL BASICU DO ODCZYTU POJEDYŃCZEGO POMIARU Z MIERNIKA „METEX ME-32”

Przed przystąpieniem do tworzenia aplikacji, zapoznać się z instrukcją miernika METEX ME-32. Zapisać w brudnopisie wymagania miernika dotyczące komunikacji przez port RS-232. Instrukcja znajduje się załączniku 2.

Tworzenie prostej aplikacji.

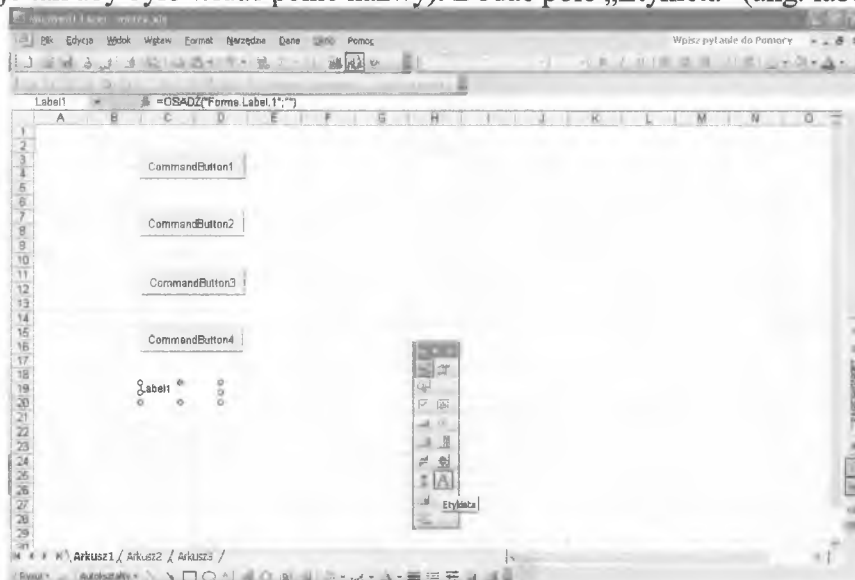
- a) Otworzyć arkusz kalkulacyjny. Zapisać aktywny zeszyt np. jako „Metex.xls” na pulpicie. Nacisnąć znajdującą się po lewej stronie ikonę „Przybornik formatów”. Pojawi się okienko z przyciskami do tworzenia aplikacji – *rys. 4.4.1.*

UWAGA: w czasie tworzenia aplikacji musi być aktywna ikona „tryb projektowania”.



Rys. 4.4.1.

- b) Wybrać „przycisk polecenia” (ang. CommandButton) i umieścić w arkuszu cztery takie przyciski (rozciągnąć je tak aby było widać pełne nazwy). Dodać pole „Etykieta” (ang. label). *Rys. 4.4.2.*

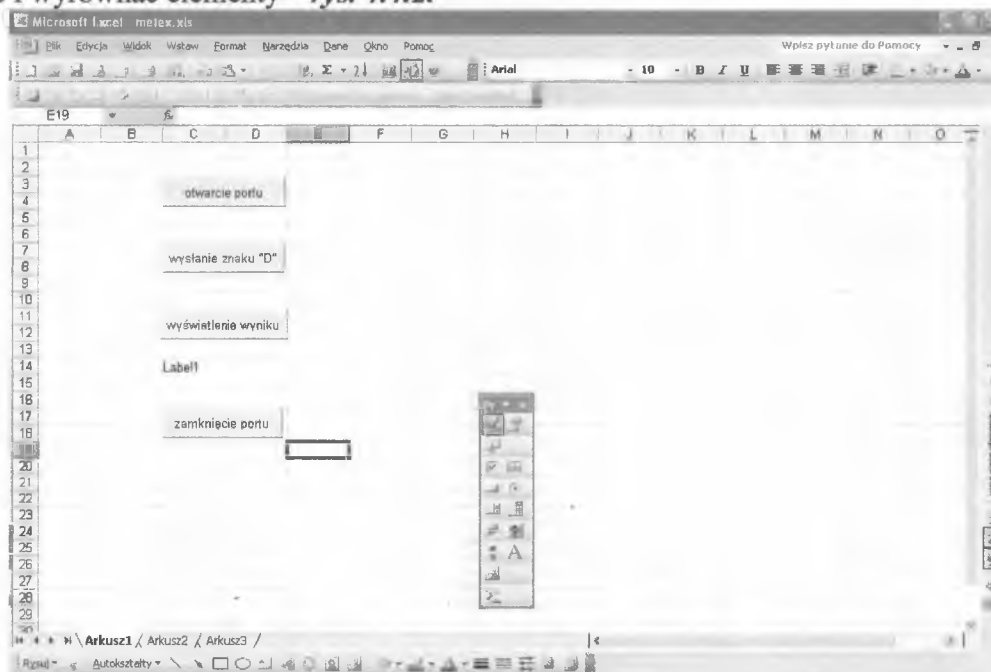


Rys. 4.4.2.

- c) Zmień nazwy poszczególnych przycisków, poprzez najechanie wskaźnikiem myszki na dany element, wciśnięcie lewego przycisku i wybranie obiektu **CommandButton**→**Edit**.
Wprowadzamy nazwy związane z obsługą aplikacji, np.:

- **CommandButton1** → otwarcie portu;
- **CommandButton2** → wysłanie znaku „D”;
- **CommandButton3** → wyświetlenie wartości z miernika;
- **CommandButton4** → zamknięcie portu.

Ułożyć i wyrównać elementy – **rys. 4.4.2.**



Rys. 4.4.3.

- e) Przez dwukrotne kliknięcie pierwszego przycisku przejść do Arkusza edycji kodu. Wpisać kod dla **CommandButton1** – kody znajdują się w **załączniku 1**. Czynność powtórzyć dla czterech przycisków.

UWAGA: w komendzie **NETCommX** X oznacza numer portu (domyślnie Com1) - sprawdzić używany numer portu com, zamienić X na konkretny numer.

- d) Dodać sterownik portu (MSComm lub NETComm). W przyborniku znaleźć przycisk „Więcej formatów”. Znaleźć i zaznaczyć opcję **NETCommOCX.NETComm**. Lewym klawiszem myszy edytować „właściwości” obiektu i ustawić transmisję zgodną z wymaganiami miernika – załącznik 4.

- c) Otworzyć okno „Propertis” obiektu **Label1**. Ustawić:

- kolor tła – np. czerwony ☺;
- utworzyć obramowanie wokół obiektu;
- zwiększyć czcionkę na min. 14pkt;
- wyśrodkować tekst napisu.

Przykładowy wygląd pokazano na **rys. 4.4.3.**

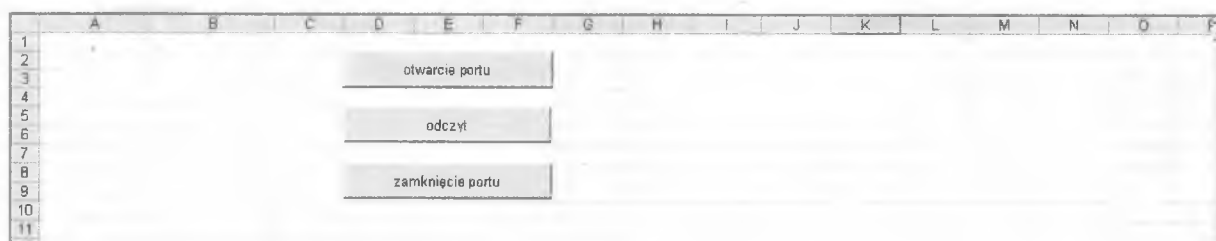
- f) Zakończyć tryb projektowania – ikona „Zakończ tryb projektowania” – **rys. 4.4.1**
zapisz jako „**miernik1**”

- g) Uruchomić ponownie Exel’a, zezwolić na uruchamianie Makr.
Przetestować program;

Odczytać wskazania miernika;
Zademonstrować program prowadzącemu zajęcia;

Sprawdzić czy utworzony program jest „idioto oporny”;
Co można poprawić w celu zwiększenia/polepszenia funkcjonalności programu?

- h) Popraw program wg rysunku 4.4.4 (kod w załączniku 2)
zapisz jako „**miernik2**”
Co się zmieniło?
Jak poprawić program aby działał właściwie (ewentualna odpowiedź w załączniku 3)?
Po poprawieniu zademonstrować program prowadzącemu zajęcia.
Co jeszcze można poprawić w funkcjonowaniu programu?
Jak zrobić wykres mierzonych wartości w czasie?



Rys. 4.4.4.

4.3.1. PROGRAM „ScopeView” do rejestracji pomiarów z mierników ME-32

Uruchomić program „ScopeView” – zapoznać się z jego funkcjonalnościami.

- jakie wielkości można ustawić w programie?
- jakie wielkości można mierzyć?
- co można zapisać?
- itp....

4.4. OPRACOWANIE SPRAWOZDANIA

Sprawozdanie powinno zawierać:

- dane charakterystyczne transmisji danych z miernika ME-32;
- przemyślane i umotywowane odpowiedzi na dwa ostatnie pytania – punkt 4.3.1. podpunkt g);
- opis funkcjonalności programu „ScopeView”;
- dokładność pomiaru i wyświetlacza miernika ME-32 (dla wybranych wielkości);

ZAŁĄCZNIK 1

```
Private Sub CommandButton1_Click()
'otwarcie portu
NETComm1.PortOpen = True
MsgBox "Port jest otwarty"
Beep
End Sub
```

```
Private Sub CommandButton2_Click()
'wysłanie znaku "D"
```

```
NETComm1.Output = "D"
```

```
Beep
```

```
End Sub
```

```
Private Sub CommandButton3_Click()
```

```
'wyświetlenie wartości odebranej z miernika w polu Etykieta (Label1)
```

```
Label1.Caption = NETComm1.InputData
```

```
Beep
```

```
End Sub
```

```
Private Sub CommandButton4_Click()
```

```
'zamknięcie portu
```

```
NETComm1.PortOpen = False
```

```
Beep
```

```
MsgBox "Port jest zamknięty"
```

```
End Sub
```

ZAŁĄCZNIK 2

```
Private Sub CommandButton1_Click()
```

```
'otwarcie portu
```

```
NETComm1.PortOpen = True
```

```
Beep
```

```
MsgBox "Port jest otwarty"
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
Cells(1, 2) = "rozpoczęcie pomiarów"
```

```
Dim i As Integer
```

```
Static j As Integer
```

```
i = i + 1
```

```
j = j + 1
```

```
'wysłanie znaku "D"
```

```
NETComm1.Output = "D"
```

```
Beep
```

```
'Czas
```

```
Cells(i, 1) = Time
```

```
'wyświetlenie wartości odebranej z miernika
```

```
Cells(j, 2) = NETComm1.InputData
```

```
End Sub
```

```
Private Sub commandButton4_Click()
```

```
'zamknięcie portu
```

```
NETComm1.PortOpen = False
```

```
Beep
```

```
MsgBox "Port jest zamknięty"
```

```
End Sub
```

ZAŁĄCZNIK 3

Zakres zmiennej czyli to, w jakich częściach programu jest ona dostępna, określamy podczas jej deklarowania. Zakres ten zależy od:

- miejsca, w którym zmienna jest zadeklarowana, w sekcji deklaracji modułu czy wewnątrz konkretnej procedury.
- za pomocą jakiego słowa kluczowego tj. Dim, Public, Private, lub Static, zmienna została zadeklarowana.

Poniżej przedstawiam kilka przykładów deklaracji. Tekst w kolorze zielonym zaczynający się od znaku ' jest komentarzem i nie ma wpływu na działanie przykładów.

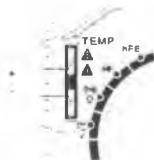
Dim X1 'Instrukcja ta może być umieszczona wewnątrz procedury, wówczas zostanie utworzona zmienna na poziomie procedury. Jeżeli natomiast deklaracja zostanie umieszczona na początku modułu, w sekcji deklaracji, utworzona będzie zmienna na poziomie modułu.

Private X2 'Stosowana na poziomie modułu do deklaracji zmiennych prywatnych oraz do przydziału pamięci. Zmienne te są dostępne tylko w tym module, w którym zostały zadeklarowane. Słowa kluczowego Private nie można użyć wewnątrz procedury.

Public X3 'Stosowana do deklarowania zmiennych publicznych na poziomie modułu. Zmienne zadeklarowane za pomocą instrukcji Public są dostępne dla wszystkich procedur we wszystkich modułach wszystkich projektów. Słowo kluczowe Public należy stosować wyłącznie w sekcji deklaracji modułu.

Static X4 'Wykorzystywana na poziomie procedury do deklaracji zmiennych i przydziału pamięci. Zadeklarowana w ten sposób zmienna zachowuje swoją wartość między wywołaniami procedury. Zmienne statyczne można deklarować tylko wewnątrz procedur.

ZAŁĄCZNIK 4 Instrukcja miernika Metex ME-32 (fragmenty)



Utwagi:

- Pomiary temperatury powinny być wykonywane bez jakichkolwiek przewodów pomiarowych.
- Dopuszczalne napięcie na wejściu sondy temperaturowej wynosi 30V AC (wartość skuteczna) lub 60V DC. Próba podłączenia wyższego napięcia grozi uszkodzeniem miernika.

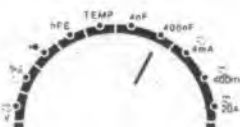
6.8. Pomiar pojemności (model ME-32)

OSTRZEŻENIA:

- ROZŁĄDOWAĆ KAŻDY KONDENSATOR PRZED POMIAREM PRZEZ ZWARCIE JEGO WYPROWADZENI! PRZESTROŻNOŚĆ TAKA JEST NIEZBEDNA, GDYŻ WŚRÓD PRZECHOWYWANYCH KONDENSATORÓW MOGĄ SIĘ ZNALEZĆ NAŁADOWANE NAWET ZNA CZĄCYM ŁADUNKIEM, A PROBA POMIARU NAŁADOWANEGO KONDENSATORA MOŻE SPOWODOWAĆ USZKODZENIE MIERNIKA!
- NIE DOTYKAĆ WYPROWADZEN KONDENSATORÓW, NA KTÓRYCH MOGĄ WYSTĄPIĆ NAPIĘCIA WIĘKSZE OD 35V_{DC} LUB 25V_{AC}.
- PRZY ZWIERNIANIU KONCÓWEK KONDENSATORA ZACHOWAĆ OSTROŻNOŚĆ I NIE WYKONYWAĆ TEGO W POMIESZCZENIACH, GDZIE WYSTĘPUJĄ GAZY WYBUCHOWE, GDYŻ ENERGIA ROZŁĄDOWANIA KONDENSATORA MOŻE BYĆ BARDZO DUŻA (POWSTAJE ISKRZENIE).

Procedura dokonywania pomiarów polemności:

1. Rozrządować każdy (I) kondensator przed pomiarem.
2. Ustawić przełącznik na wybranym zakresie pojemnościowym.



7.2.1. Uruchomienie oprogramowania w systemie MS-DOS

Uwaga: Opisana poniżej procedura zakłada podstawową znajomość przez użytkownika komend systemu DOS i posiadanie twardego dysku w komputerze. Oprogramowanie wymaga monitora VGA.

- 1 Włożyć dostarczoną dyskietkę do napędu dysków 3,5".
- 2 Na twardym dysku utworzyć katalog do przechowywania oprogramowania
Przykładowo, aby utworzyć katalog **METER**, należy wpisać komendy:

```
cd \ (ENTER)  
md METER (ENTER)
```
- 3 Wejść do utworzonego katalogu, który staje się teraz katalogiem bieżącym:

```
cd METER (ENTER)
```
- 4 Skoplować pliki z podkatalogu **GRAPHIC** z dyskietki na twardy dysk:

```
copy a:\GRAPHIC c:\
```
- 5 Uruchomienie programu następuje po wpisaniu nazwy **METEX** i potwierdzeniu klawiszem (ENTER). Na monitorze ukaze się menu **HELP**, prowadzące użytkownika po funkcjach programu.

Living:

- Uwagi eksploatacyjne dotyczące obsługi oprogramowania zamieszczono w pliku README na dyskietce.
- Przy braku twardego dysku w komputerze można uruchamiać oprogramowanie z dyskietki. W tym celu należy wejść do podkatalogu GRAPHIC i wpisać **METEX (ENTER)**.

7.2.2. Uruchomienie oprogramowania w środowisku Windows

Uwaga: Opisana poniżej procedura zakłada podstawową znajomość przez użytkownika środowiska Microsoft Windows. Oprogramowanie wymaga MS Windows w wersji 3.1 lub późniejszej i monitora VGA.

1. Włączyć komputer i uruchomić program Windows.
2. Włożyć dostarczoną dyskietkę do napędu.
3. W oknie **Menedżera Programów**, w menu Plik wybrać opcję Uruchom.
4. W oknie dialogowym wpisać, gdy dyskietka jest w napędzie A:
a:\scope\setup (ENTER).
5. Wykonywać polecenia ukazujące się na ekranie, aby zakończyć instalację.
6. Uruchomić program następując po podwójnym kliknięciu ikony SCOPEVIEW. Szczegółowych informacji na temat instrukcji i funkcji programu należy szukać w menu Help lub w pliku README w podkatalogu SCOPE na dyskietce.

7.3. Informacje techniczne

Parametry transmisji

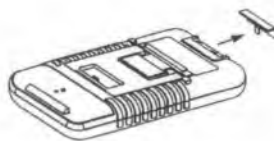
- szybkość transmisji : 600 bodów
- kod znakowy 7 bitowy ASCII
- kontrola parzystości : brak
- bity stopu : 2

3. Włożyć wyprowadzenia rozładowanego kondensatora do gniazda CAP + i -. Na wyświetlaczu ukaze się wartość jego pojemności. Mierzac kondensator polaryzowany, należy upewnić się, że jego ujemna końcówka znajduje się w gnieździe oznaczonym - (minus).

7. WSPÓŁPRACA MIERNIKA Z KOMPUTEREM PC

7.1. Sprzężenie miernika z komputerem klasy PC

1. Zdjąć osłone złącza RS-232C w górnej części obudowy.



2. Połączyć kablem z wtykiem D-sub gniazdo RS-232C miernika z portem szeregowym komputera.



- Przyciskiem **ON/OFF** włączyć młynik.
- Uruchomić komputer.

7.2. Oprogramowanie narzędziowe

Wraz z miernikiem producent dostarczył dwa programy do rejestracji i wyświetlania danych pomiarowych przekazywanych przez miernik. Program przeznaczony do pracy w systemie MS-DOS nazywa się **METEX** i znajduje się w podkatalogu **GRAPHIC** na łączącej dyskietce. Program do pracy w środowisku Windows nosi nazwę **Scopeview** i znajduje się w katalogu **SCOPE**.

Format danych

Format danych jest oparty na 14-bajtowej ramce jak w poniższych przykładach:

Bajt:	1	2	3	4	5	6	7	8	9	A	B	C	D	E
Przykład 1.:	D	C	-	3	9	9	.	9				m	V	CR
Przykład 2.:	O	H		3	9	.	9	9	M	o	h	m	CR	

7.4. Uwagi dla twórców własnego oprogramowania

Należy pamiętać podczas pisania własnego programu obsługi, że rozpoczęcie transmisji danych następuje po przesłaniu przez komputer sterujący do miernika komendy „D”. Poniżej przedstawiono przykładowy program w języku GW BASIC do uzyskiwania na komputerze wyniku pojedynczego pomiaru:

```

10 OPEN "COM1 : 2400, N, 7, 2, RS, CS, DS, CD" AS #2
20 AS="D"
30 PRINT #2, AS
40 IN$=INPUT$(14, #2)
50 PRINT IN$
60 CLOSE #2
70 END

```

B. UTRZYMANIE I KONSERWACJA

8.1. Wymiana bezpiecznika

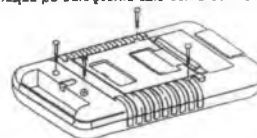
OSTRZEŻENIE: ZAWSZE ODŁĄCZAĆ PRZEWODY POMIAROWE PRZED WYMIANĄ BATERII LUB BEZPIECZNIKA. STOSOWAĆ TYLKO TAKI SAM TYP BATERII I BEZPIECZNIKA, JAK ORYGINALNY. WSZELKIE CZYNNOŚCI SERWISOWE POWINNY BYĆ WYKONYWANE JEDYNIEM PRZEZ WYKWA- LIFIKOWANĄ PERSONEL.

UWAGA: DLA ZAPEWNIENIA WŁAŚCIWEGO STOPNIA BEZPIECZEŃSTWA
PAMIĘTAĆ NALEŻY O DOBORZE PARAMETRÓW NAPIĘCIOWYCH I PRĄ-
DOWYCH WYMIENIANEGO BEZPIECZNIKA, TAKICH JAK BEZPIECZNIKA
ZAINSTALOWANEGO FABRYCZNIE

bezpiecznik F2 → FF 15A/250V
bezpiecznik F1 → F 0.8A/250V

Procedura wymiany bezpiecznika:

1. Wylączyć młernik przyciskiem **ON/OFF** i odłączyć przewody pomiarowe.
2. Zdjąć dolną pokrywę przyrządu po odkręceniu czterech śrub mocujących.



4. Włożyć nowy bezpiecznik w oprawkę.
5. Założyć pokrywę i zakręcić śruby.

Uwaga: Wymianę bezpiecznika F2 FF15A/250V należy zlecić wykwalifikowanej osobie.

OSTRZEŻENIE: NIE DOKONYWAĆ POMIARÓW PRZED ZAMKNIĘCIEM POKRYWY I ZAKRĘCENIEM ŚRUB ZABEZPIECZAJĄCYCH!

8.2. Uwagi eksploatacyjne

Omawiane mierniki są przykładem doskonałej konstrukcji, wzornictwa i wykonania. Poniższe uwagi pozwolą Państwu utrzymać przyrządy w dobrym stanie technicznym przez wiele lat.

- Nie dopuszczać do zalania miernika jakimkolwiek płynem, gdyż może on zawierać składniki powodujące korozję elementów elektronicznych. W przypadku takiego zdarzenia niezwłocznie wytrzeć przyrząd do sucha.
- Używać i przechowywać miernik w przeciwnych temperaturach otoczenia. Ekstremalne temperatury wpływają na szybsze starzenie się elementów elektronicznych i baterii, a nawet mogą uszkodzić plastikowe części urządzenia.
- Zachować ostrożność podczas transportu przyrządu. Jego upuszczenie może uszkodzić obudowę lub płytkę drukowaną.
- Chronić miernik przed kurzem i brudem, który może być przyczyną przedwczesnego zużycia jego podzespołów.
- Obudowę przyrządu można od czasu do czasu przetrzeć czystą, wilgotną szmatką. Nie używać rozpuszczalników, silnych detergentów i innych silnych środków czyszczących.
- Używać tylko nowych i odpowiedniego typu baterii. Zawsze wyjmować z miernika baterię zużytą lub starą, gdyż wydatek z baterii jest szkodliwy dla obwodu elektronicznego.

Modyfikacje obwodu wewnętrznego przyrządu i wymiana jego elementów powodują unieważnienie gwarancji, a także utratę dokładności miernika.

9. DANE TECHNICZNE

9.1. Dane podstawowe

Wyświetlacz LCD:	maks. wskazanie 3999 (3 ^{1/2} cyfry), automatyczny wakażnik polaryzacji
Maksymalny prąd wejściowy:	20 A (AC/DC)
Częstotliwość odczytu:	1-2 razy na sekundę
Temperatura pracy:	0°C do 40°C
Temperatura przechowywania:	-10°C do 50°C
Temperatura gwarantowanej klasy dokładności:	+23°C ±5°C
Zasilanie:	bateria 9V (NEDA 1604 lub 6F22)
Wymiary:	81 x 187 x 34mm
Waga:	350g ±10g (z baterią)
Wypożyczenie - podstawowe:	instrukcja obsługi, para przewodów pomiarowych, zapasowy bezpiecznik topikowy (800mA/250V)
- dodatkowe:	futerał, dyskietka z programem narzędziowym, sonda termoparowa, kabel połączeniowy z wtykiem D-sub

9.2. Parametry elektryczne

(format zapisu dokładności: ± % odczytu + wartość najmniej znaczącej cyfry)

MODEL	FUNKCJA	ZAKRES	DOKŁADNOŚĆ	Rozdzielczość
ME-31 ME-32	Napięcie stałe (DCV)	400 mV	± 0,5% + 3	100 µV
		4 V		1 mV
		40 V		10 mV
		400 V		100 mV
		1000 V		1 V
	Napięcie przemienne (ACV)	400 mV	± 1,0% + 5 (40~100Hz)	100 µV
		4 V	± 1,0% + 5 (40~400Hz)	1 mV
		40 V		10 mV
		400 V		100 mV
		750 V	± 1,5% + 5 (40~400Hz)	1 V
	Prąd stały (DCA)	4 mA	± 0,8% + 3	1 µA
		400 mA	± 1,2% + 3	100 µA
		20 A	± 2,0% + 5	10 mA
	Prąd zmienny (ACA)	4 mA	± 1,0% + 5 (40~400Hz)	1 µA
		400 mA	± 1,5% + 5 (40~400Hz)	100 µA
		20 A	± 3,0% + 5 (40~400Hz)	10 mA

ME-31 ME-32	Rezystancja (OHM)	400 Ω	± 0,5% + 3	0,1 Ω
		4 kΩ		1 Ω
		40 kΩ		10 Ω
		400 kΩ		100 Ω
		4 MΩ		1 kΩ
		40 MΩ	± 3,0% + 5	10 kΩ
ME-32	Test diod	Prąd testowy maks. 1,5mA na rezystancji 1kΩ (napięcie przewodzenia maks. 2V)		
	Test ciągłości	Sygnał akustyczny jest generowany, gdy rezystancja testowanego obwodu jest mniejsza niż 50Ω.		
	Pojemność	4 nF	± 3,0% + 10	1 pF
		400 nF	± 3,0% + 5	0,1 nF
	Temperatura	0~200°C	± 3,0% + 5	
		201~1000°C	± 3,0% + 3	
	h _{FE} tranzystorów	Pomiar przybliżonej wartości wzmacnienia tranzystorów (parametr h _{FE}) przy prądzie ok. 1000µA.		