

# Do czego używamy komputera ?

Dla elektronika komputer jest niezastąpionym narzędziem, to każdy wie. Potrzebujemy go, aby połączyć się ze światem w poszukiwaniu ważnych informacji, projektować płytki drukowane, programować procesory, tworzyć dokumentację swoich konstrukcji, pisać listy itd.

Ja chciałbym Wam podsunąć jeszcze jeden pomysł na wykorzystanie pocziwego PC-ta, aby pomóc sobie w rozwiązywaniu nietypowych problemów. Najlepiej będzie jeśli posłużę się przykładami, aby wyjaśnić co mam na myśli.

Wielu elektroników zna jakiś język programowania. Inni na pewno jakiegoś się uczą, a pozostali - prędzej czy później się go nauczą. Swoje przykłady opieram na BASIC-u, ponieważ chyba największe grono Czytelników go zna, choćby z przygody z BASCOM-em. W każdym innym języku można podobnie rozwiązywać takie problemy. Można to zrobić nawet na arkuszu kalkulacyjnym, jeśli jest wyposażony w możliwość graficznego przedstawiania wyników.

## Przykład 1

Jako pierwszy przykład weźmy sobie układ ze „Skrzynki Porad” z numeru 1/2002 EdW. Chodzi o zmianę charakterystyki potencjometru liniowego przy użyciu dodatkowego rezystora. Na rysunkach przedstawione są tam charakterystyki wynikowe. Pokażę tu sposób szybkiego wykreślenia ich na monitorze dla oszacowania która z nich byłaby dla nas najbardziej korzystna. Napijemy kilka wierszy w BASICu:

```
screen 2
P=10
R=20
for X=.1 to P+.11 step .05
  Rx=1/(1/X+1/R)
  K=Rx/(P-X+Rx)
  pset (60*X,199-X*20)
  pset (60*X,199-200*K)
next X
```

Aby wykreślić na ekranie kolejne punkty obrazujące zmianę współczynnika podziału dzielnika napięcia jakim jest potencjometr z dołączonym rezystorem R (rysunek 1), trzeba napisać wzór na tenże współczynnik K:

$$K = R_x / (P - X + R_x),$$

gdzie:

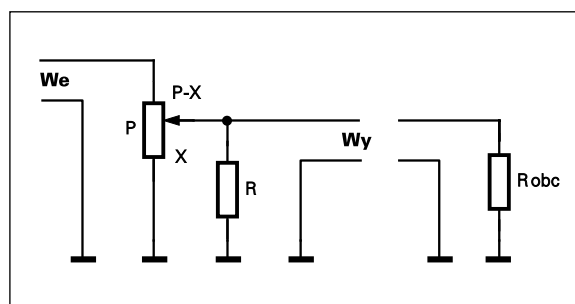
**P** - rezystancja całego potencjometru,

**X** - ustawienie potencjometru (rezystancja „dolnej” części - zmienia się od zera do P),

**R<sub>x</sub>** - rezystancja „dolnej” części potencjometru wraz z dołączonym rezystorem R

$$R_x = 1 / (1/X + 1/R) \text{ lub } R_x = (X * R) / (X + R)$$

Program oblicza w pętli *for - next* współczynnik podziału dla kolejnych ustawień potencjometru, co 0,05 kΩ. Wartość tego skoku możemy zmieniać tak, aby uzyskać czytelny wykres i nie czekać przy tym zbyt długo na wynik obliczeń.



Rys. 1 Schemat potencjometru z dodatkowym rezystorem do korekcji charakterystyki.

W Turbo Basicu komendą służącą do rysowania punktu na ekranie jest PSET (x,y). Rozdzielczość w trybie *screen2* wynosi 199 na 639 pixeli. Początek układu współrzędnych jest w górnym lewym narożniku ekranu, a zatem zamiast „y” należy pisać „199 - y” aby uzyskać normalne położenie wykresu. Współczynniki 60, 20 i 200 w tych komendach tak ustawiają skalę, aby nasz wykres zajął pełny ekran.

Jeszcze wyjaśnienie dlaczego jest *for X=.1 to P+.11 step .05* a nie *for X=0 to P step .05*.

Dlatego, aby nie miał miejsca przypadek dzielenia przez zero dla X=0. Nie zmienia to w żaden znaczący sposób uzyskanych wykresów, a uwalnia nas od kłopotów z działaniem programu.

Aby ostatnia wartość w pętli nie wychodziła poza zakres pętli - lepiej dać górną granicę nieco większą (P+.11). Brak tej jednej setnej czasami powoduje nieprzechodzenie pętli przez ostatnią wartość z powodu zaokrągleń BASICA.

Powyższa pętla *for - next* rysuje jednak tylko jeden wykres - dla wartości R=20 kΩ.

To już jest w jakiś sposób pomocne, ale my chcielibyśmy zobaczyć wpływ wartości R na kształt charakterystyki. Do tego posłużą druga pętla *for - next*, w której zmieniać się będzie R:

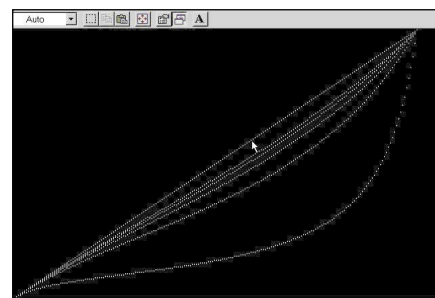
```
screen 2
P=10
for R=1 to 21.1 step 5
  for X=.1 to P+.11 step .05
```

```
Rx=1/(1/X+1/R)
K=Rx/(P-X+Rx)
pset (60*X,199-X*20)
pset (60*X,199-200*K)
next X
next R
```

Od tego momentu program rysuje po kolei charakterystyki potencjometru wraz z dołączonym coraz większym rezystorem dodatkowym R (rysunek 2). Zauważmy, że charakter tych krzywych nie jest dokładnie taki, jak na rysunkach z EdW 1/2002...

Wszystkie wykresy będą w dalszym ciągu rysowane na tle linii prostej będącej charakterystyką samego potencjometru:

$$pset (60 * X, 199 - X * 20).$$



Rys. 2 Charakterystyki wypadkowe dla P=10kΩ, zmiana R - od 1kΩ do 21kΩ - co 5kΩ.

Wprawdzie linia ta rysowana jest za każdym razem, czyli w tym przypadku 5 razy, ale nam to w żaden sposób nie przeszkadza, natomiast upraszcza program.

Widząc teraz wszystkie 5 krzywych możemy łatwo ustalić, którą wartość R najlepiej będzie wykorzystywać. Na szybkich komputerach krzywe będą rysowane tak szybko, że nie zdążymy zauważyć od której krzywej program zaczyna. Jeśli zależy nam na tym, aby widzieć jak postępuje rysowanie, trzeba wstawić w pętłę opóźnienie, czyli np. jeszcze jedną pustą pętlę ze zmienną t:

```
for t=1 to 1000:next t
```

Jeśli wstawimy ją powiedzmy między instrukcje PSET, czyli do wewnętrznej pętli, to cała krzywa będzie rysowana wolniej. Czasem jednak lepiej jest pozostawić wewnętrzną pętlę szybką, a większe opóźnienie wstawić w zewnętrzną pętlę. Wtedy między poszczególnymi wykresami mamy chwilę przerwy, co pozwala zorientować się o jakiej kolejności parametrów odnoszą się poszczególne krzywe.

Z powyższych wykresów widać na przykład, że wstawienie opornika dodatkowego  $6k\Omega$  powoduje dużą i podobną zmianę charakterystyki jak jego dalsze zmniejszenie do  $1k\Omega$ , a np. zmiana wartości z  $11k\Omega$  na  $16k\Omega$  daje tylko subtelną zmianę kształtu krzywej.

Jako dalsze ćwiczenie można przerobić nieco program i wykresić np. przebieg rezystancji wejściowej, takiego dzielnika w funkcji kąta obrotu gałki i to jeszcze przy kilku różnych rezystancjach obciążenia dzielnika. Znowy dwie pętle, kilka wierszy programu i już wszystko wiadomo:

screen 2

$P=10$

$R=20$

for Robc=1 to 101.1 step 20

for x=.1 to P+.11 step .05

$Rx=1/(1/x+1/R+1/robc)$

$Rwe=P-x+Rx$

pset (60\*x,199-Rwe\*15)

next x

for t=1 to 100000:next t

next Robc

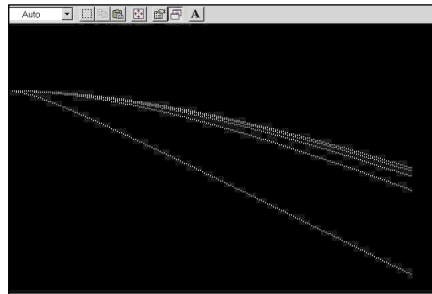
**Rysunek 3** pokazuje zmianę oporności wejściowej dzielnika z dodatkowym opornikiem  $20k\Omega$  w funkcji kąta obrotu przy obciążeniu zmieniającym się co  $20k\Omega$  od  $1k\Omega$  do  $101k\Omega$ .

Jeśli zacznie się już używać takich metod, szybko można się nauczyć jak postępować, aby jak najszybciej sprawdzić zachowanie się układu przy zmieniających się różnych parametrach.

W moim przypadku najczęściej zaczyna się od paru linijek programu a potem się to

rozrasta, bo chcę sprawdzić jeszcze to i tamto. W każdym razie polecam gorąco opatrzenie nawet każdej linijki komentarzami, abyście po roku również mogli z takiego - raz napisanego - programu skorzystać

**Rys. 3 Charakterystyki rezystancji wejściowej dzielnika dla  $R=20$  i Robc zmieniającej się od  $1k\Omega$  do  $101k\Omega$  - co  $20k\Omega$ .**



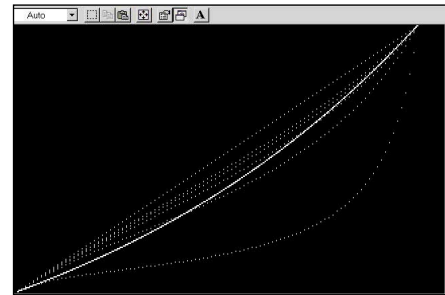
Dla oceny, jak dalece otrzymane krzywe przypominają prawdziwą charakterystykę logarytmicznego potencjometru, możemy przed właściwym programem narysować na ekranie po prostu krzywą wykładniczą (**rysunek 4**). Niewielu z Was się chyba nad tym zastanawiało, ale potencjometr logarytmiczny (B) ma charakterystykę wykładniczą, a wykładniczy (C) - logarytmiczną. Załatwia to następujący fragment:

for x=.1 to P+.11 step .01

pset (60\*x,199-(100\*1.116^x)+100)

next x

wstawiony przed właściwymi pętlami for - next.



**Rys. 4 Krzywe z rysunku 2 na tle krzywej wykładniczej (charakterystyki potencjometru logarytmicznego).**

Parametry komendy PSET należy dobrać tak, aby wykres został narysowany na całym ekranie, czyli aby rozpoczynał się i kończył w punktach skrajnych następujących wykresów.

Skok celowo proponuję mniejszy (0,01), aby krzywa ta wyróżniała się na tle pozostałych.

Po takim doświadczeniu widzimy, że ten opornik dodatkowy to jest jednak tylko półśrodek.

Gdybyśmy się uparli na ściśle logarytmiczną charakterystykę, to należałoby poszukać bardzo drogiego potencjometru spełniającego wysokie wymagania albo...

przeczytać drugą część artykułu, która ukaże się za miesiąc.

**Marek Klimczak**  
matik1@poczta.onet.pl