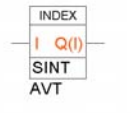
Realizer cz. 5

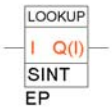
Krzysztof Górski

W tej części poznamy możliwości jakie dają nam takie elementy biblioteczne, jak tabele. Z obserwacji i rozmów z fanami Realizera wynika że wykorzystanie tabel w projektach sprawiało trochę kłopotów.
Aby rozwiać wszelkie wątpliwości postaram się wyjaśnić zastosowanie tabelek na przykładzie prostego projektu.
ST-Realizer oferuje w swojej bibliotece dwa rodzaje tabel

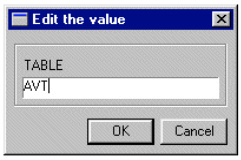
- indextable



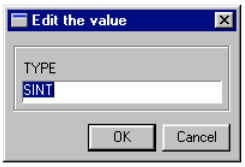
- lookuptable



IndexTable daje nam możliwość konwersji wartości wejściowej na nieliniową wartość wyjściową, którą definiujemy w tabeli .
W **LookupTable** wartość wejściowa jest wykorzystywana do znalezienia i umieszczenia wartości stałej z tabeli na jej wyjściu.
Opis może wydawać się trochę zawiły ale spróbujemy go sprowadzić do prostszej postaci w praktycznym zastosowaniu.
Uruchamiamy Realizera tworzymy nowy projekt nazywamy go np. **EPKURS**. Z biblioteki wybieramy elementy **INDEXTABLE** oraz **INDEXLOOKUP** i umieszczamy na planszy schematu. Podczas umieszczania na planszy schematu elementu z biblioteki otworzy się najpierw okno edycji nazwy tabeli Rys.3 oraz okno edycji zmiennej Rys.4.



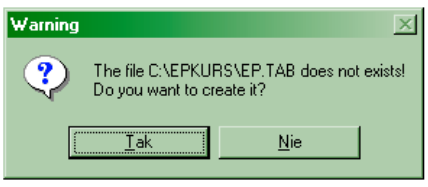
Rys.3



Rys4

Aby wejść w edycję zawartości którejkolwiek z tabel należy na wybrany symbol tabeli dwukrotnie kliknąć.

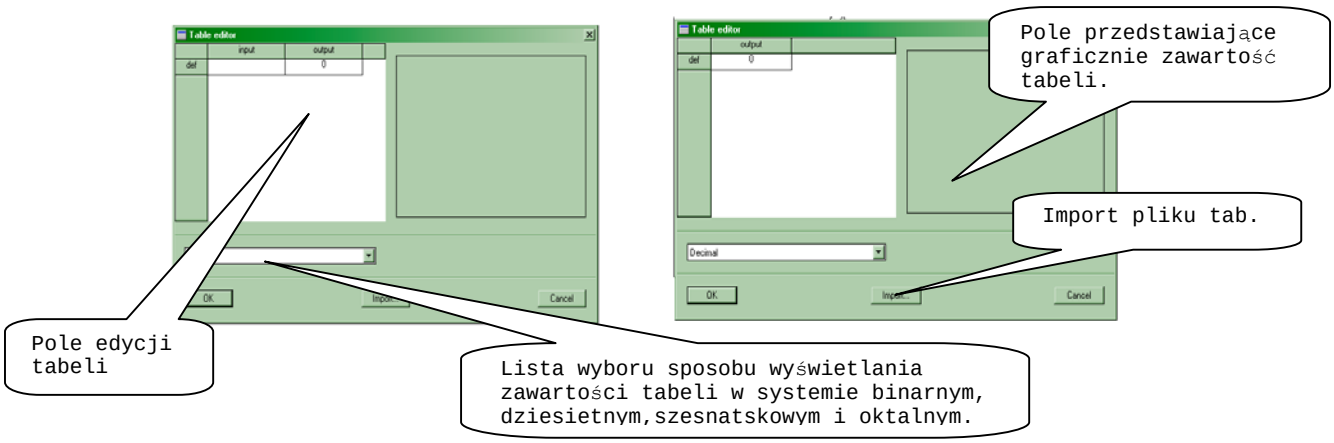
- Otworzy się okno z pytaniem czy utworzyć plik TAB Rys.5 wciskamy Ok.



Rys.5

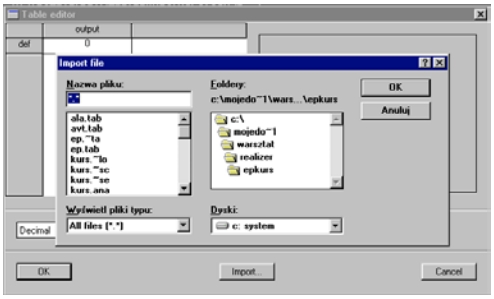
- Zostaje utworzony plik Tab o takiej samej nazwie jak tabela

Ważne jest to aby nazwy tabel nie powtarzały się gdyż może to doprowadzić do błędnej pracy programu. Po utworzeniu pliku otwiera się okno edytora tabeli TableEditor Rys.6



Rys.6

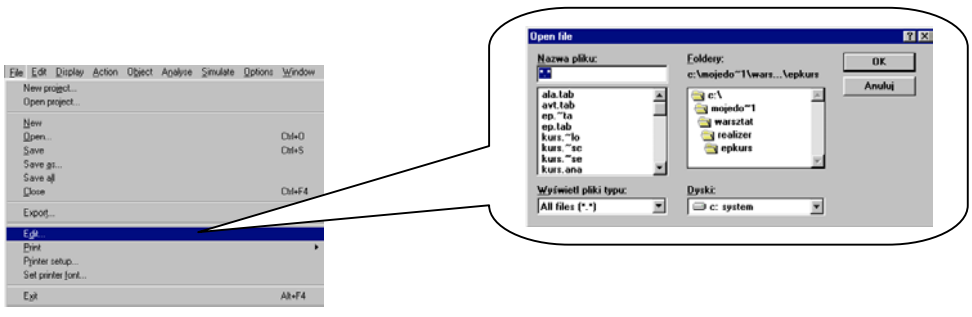
Po prawej stronie widzimy okno edytora dla tabeli Indextable natomiast po lewej dla tabeli Lookuptable.
Oba Okna edytora tabel po utworzeniu pliku tab zawierają tylko jedną pozycję def definiującą na wyjściu tabel wartość w przypadku obecności na wejściu wartości nie ustalonej w tabeli. Aby zwiększyć ilość pozycji do definiowania w tabeli należy dokonać importu gotowej tabeli klikając przycisk import. Otwiera się okno Rys.7 w którym dokonujemy wyboru pliku do importu.



Rys.7

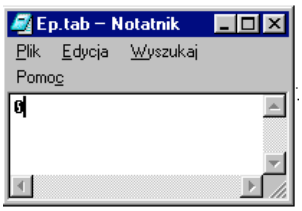
Utworzone pliki TAB możemy otworzyć ręcznie w notatniku i dokonać ich edycji. Ładnie to brzmi ale jak to zrobić? Otwarcia pliku tab dokonujemy w następujący sposób:

- w menu File wybieramy opcję Edit... otworzy się okno open file Rys.8 w nim dokonujemy wyboru interesującego nas pliku TAB



Rys.8

- wybrany przez nas plik zostaje otwarty w systemowym notatniku Rys.9



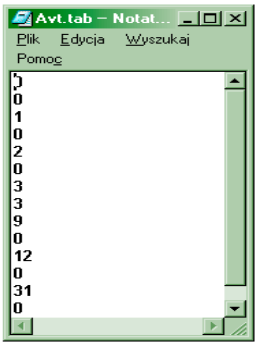
Rys.9

Gdy już mamy otwarty plik TAB widzimy że plik ten zawiera tylko jedną cyfrę 0, która określa wspomnianą pozycję def w tabeli.
Bardzo ważną sprawą jest sposób zapisu podczas edycji tabelki w notatniku. W związku z tym że mamy dwa rodzaje tabel będziemy mieli dwa sposoby zapisu danych:

- Dla tabelki lookuptable zapisu będziemy dokonywać w dwóch kolumnach oddzielonych przecinkiem , pierwsza kolumna to wartości wejściowe druga to wartości wyjściowe Rys.10.

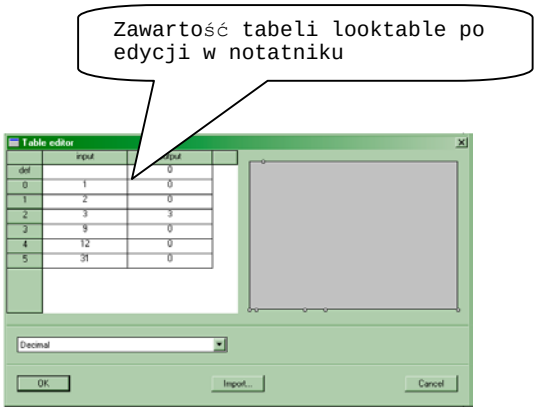
Rys.10

- Natomiast dla tabelki indextable zapisu dokonamy tylko w jednej kolumnie, która przedstawia wartości wyjściowe Rys.11.

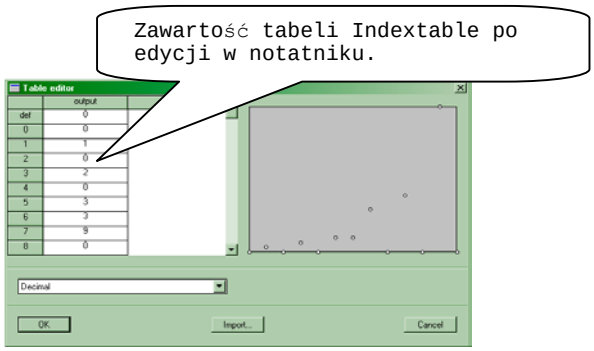


Rys.11

Na pewno zauważyliście że ilość kolumn w pliku TAB jest ściśle powiązana z ilością kolumn w Table Editor. Po dokonaniu odpowiedniego zapisu w plikach Tab należy zapisać zmiany i zamknąć notatnik. Efekty swojej pracy możemy sprawdzić dwukrotnie klikając na edytowaną tabelę po czym otworzy się okno edytora tabeli Rys 12 i 13.

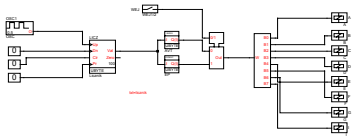


Rys.12



Rys.13

Opisaną metodę edycji zawartości tabeli należy tylko stosować przy tworzeniu tabeli po raz pierwszy. Później gdy będziemy mieli już kilka tabel stworzonych w różnych projektach możemy tylko dokonywać importu do tworzonej nowej tabeli. I w tym momencie nasuwa się pytanie czy jeżeli będziemy chcieli zmienić którąkolwiek wartość w tablicy to musimy za każdym razem edytować ją w notatniku?
Otóż nie, wystarczy wejść w Table Editor danej tabelki kursorem najechać na interesującą nas daną, kliknąć dwukrotnie po czym otworzy się okno w którym zmieniamy wartość. Proste prawda!
Po tej ilości teorii przystąpmy do praktycznego zastosowania tabelek. Przy okazji tworzenia projektu poznamy inny ciekawy element biblioteczny jakim jest licznik ***countf*** . Na Rys. 14 widzimy schemat programu stworzonego na potrzeby lekcji. Przedstawia on kilka elementów połączonych razem w celu sprawdzenia działania tabelek.

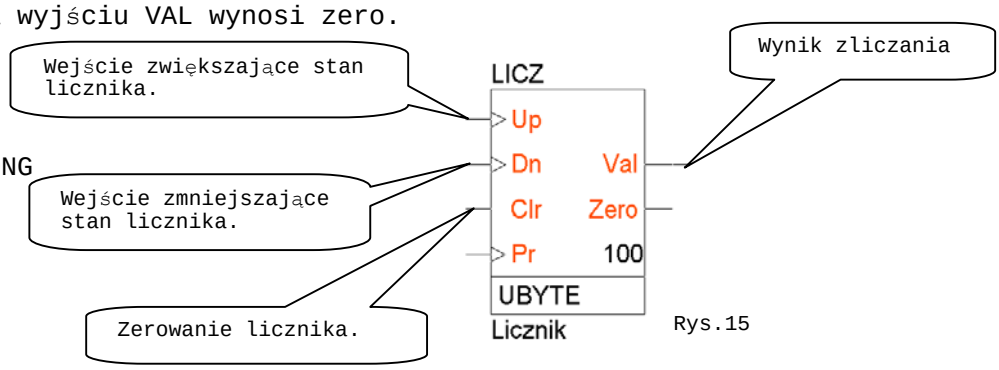


Rys.14

Jak widzimy na schemacie nie jest to skomplikowana konstrukcja licznik (countf) zlicza impulsy pochodzące z generatora (oscy) do wartości *value* zależnej od wpisanego rodzaju zmiennej w liczniku (countf).
UBYTE: unsigned byte, 0 .. 255
SBYTE : signed byte, -128 .. 127
UINT: unsigned integer, 0 .. 65535
SINT : signed integer, -32768 .. 32767
LONG: signed long, -2147483648 ..2147483647
Wyjście oscylatora połączone jest z wejściem zliczającym w górę licznika UP. Z pozostałych wyprowadzeń licznika nie będziemy korzystać. Wejścia te jednak nie mogą pozostać w „powietrzu”, najprościej do nich dołączyć elementy *constb* z wartością 0.
Wyjście z licznika val połączone jest z wejściami tabel, natomiast wyjścia tabel poprzez element multipleksera mux1 na wejście ***bunpack*** którego wyjścia dołączone są do ***digout***.

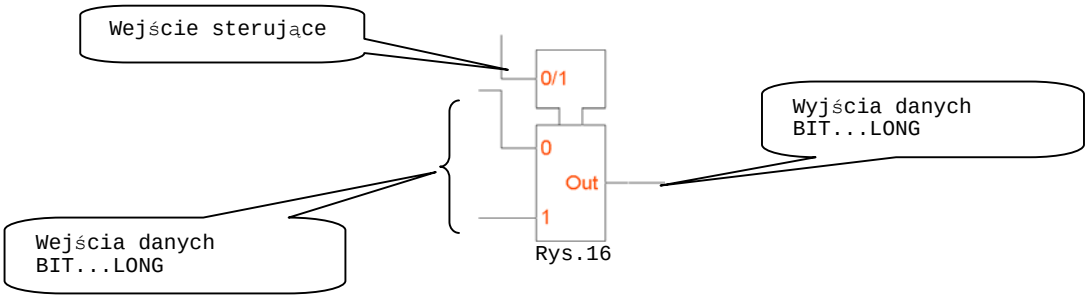
Program działa w następujący sposób: impulsy z oscylatora zliczane są przez licznik ***countf*** podczas zliczania na jego wyjściu pojawia się słowo które następnie po podaniu na wejście tabelki wywołuje przypisaną wartość która pojawia się na wyjściu tabeli. Wartość ta podana na wejście bunpack powoduje wyświetlenie odpowiedniej kombinacji stanów logicznych na wyjściach B0-B7. Aby dokładnie prześledzić działanie tabelek najlepiej obserwację przeprowadzić w symulatorze.
W przykładowym projekcie użyliśmy kilka elementów bibliotecznych niektóre z nich warte są opisania.
Do nich należą m.in. licznik ***countf*** , multiplekser ***mux.1***, oscylator ***OSC***, ***Countf*** jest licznikiem zliczającym impulsy do wartości na stałe określonej. Wejście UP licznika reaguje na zbocze wschodzące zliczanego impulsu, jest to wejście powiększające wartość licznika. Natomiast każdy impuls na wejściu DN pomniejsza zawartość licznika, reaguje ono na zbocze wschodzące każdego impulsu.
Pojawienie się wysokiego stanu na wejściu licznika CLR powoduje wyzerowanie licznika do zera. Dlaczego tak zaznaczyłem że do zera? Dlatego że podanie bitu na wejście Pr licznika powoduje jego wyzerowanie do wartości zapisanej w liczniku. Jest to wartość zależna od rodzaju zmiennej dla jakiej został zadeklarowany licznik. I tak na przykład dla UBYTE może to być wartość od 0...255. Na wyjściu licznika Val podawany jest wynik zliczania w postaci słowa UBYTE...LONG . Na wyprowadzeniu Zero pojawia się BIT w chwili gdy stan licznika na wyjściu VAL wynosi zero.

Up = BIT.
Dn = BIT.
Clr = BIT.
Pr = BIT.
Val = UBYTE...LONG
Zero = BIT



Rys.15

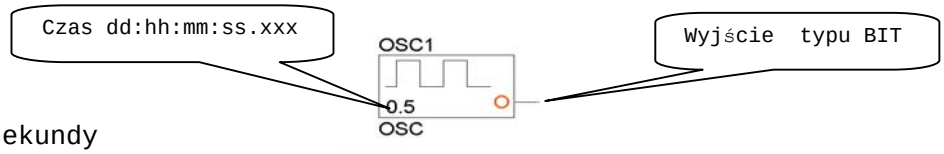
Multiplekser *mux.1* jest przełącznikiem dołączającym jedno z dwóch wejść do wyjścia Out pod wpływem sygnału sterującego podanego na wejście 0/1.



Rys.16

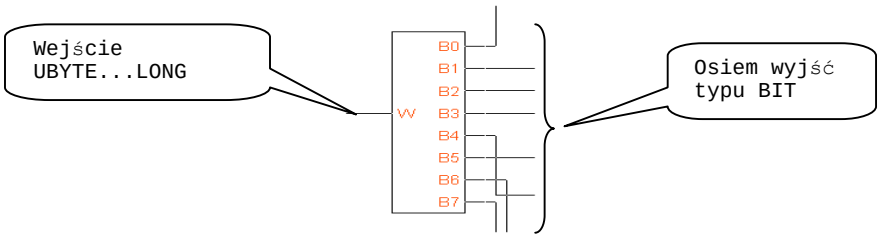
Do wejść mikrokontrolera 0 i 1 możemy dołączyć zmienne każdego typu od BIT....LONG. Na wyjściu multipleksera pojawi się zmienna o takiej samej wartości jaka została dołączona do wybranego wejścia. Należy pamiętać że wejście sterujące jest typu BIT. Jeżeli na wejściu tym jest stan 0 dołączone do wyjścia jest wejście 0 .
Oscylator OSC jest generatorem o stałej częstotliwości. Czas ustawiany jest poprzez zmianę atrybutu TIME w symbolu, zapis w następującym formacie:
dd:hh:mm:ss.xxx gdzie

dd - dni
hh - godziny
mm - minuty
ss - sekundy
xxx - części sekundy



Rvs.17

Częstotliwość impulsów generowanych przez generator równa się f(Hz) = 1/(2*time). Aby nie zapisywać czasu w tak długi formacie dd:hh:mm:ss.xxx możemy dokonać skrótu tak jak to widzimy na rysunku przedstawiającym generator. I tak np. dla zapisania czasu 1.25sekundy dłuższy zapis wygląda następująco, 00:00:00:01.25 a krótszy zapis to tylko końcówka dłuższego(1.25).
Bunpack jest elementem przetwarzającym jednobajtowe słowo na osiem bitów. Na wejście W możemy podać zmienną typu UBYTE...LONG, wyjścia są typu BIT.



Rys.18

W kolejnej części kursu Realizera zapoznamy się ze sterowaniem wyświetlacza alfanumerycznego spod Realizera.

Krzysztof Górski
Krzysztof.gorski@ep.com.pl