



Język VERILOG w przykładach (6)

Odczyt klawiatury matrycowej i PS/2

Klawiatura, czy jakiś przycisk, są bardzo ważnymi elementami wielu urządzeń. Pomijając te, które komunikują się z użytkownikiem wyłącznie poprzez podłączony do niego komputer (np. poprzez RS-232 czy USB) z odpowiednim oprogramowaniem, praktycznie każde urządzenie ma co najmniej jeden przycisk (sterujący, wyboru funkcji, zakresu itp.). O ile obsługa pojedynczych przycisków nie sprawia większych trudności (należy tylko wyeliminować lub przecześć zjawisko drgania styków), to już użycie klawiatur matrycowych lub z interfejsem PS/2 wymaga nieco większego wysiłku. W tej części kursu pokazemy, jak zastosować klawiatury matrycowe albo komputerowe PS/2 w projektach realizowanych w FPGA.

Połączenia elektryczne w klawiaturze matrycowej (klawiaturze XY) są ułożone w kolumnach i wierszach (rys. 6.1). Po wciśnięciu klawisz zwraca linię kolumny z linią

wiersza, w których jest umieszczony. W tym przykładzie linie kolumn są liniami sterującymi, a linie wierszy liniami kontrolnymi. Układ do którego jest dołączona klawiatura

ustawia kolejno na każdej kolumnie poziom wysoki, a następnie sprawdza linie kontrolne, poszukując na nich poziomu wysokiego oznaczającego wciśnięcie klawisza. Aby zabezpieczyć układ (np. FPGA) przed skutkami wciśnięcia jednocześnie dwóch klawiszy w jednym wierszu (zwarcie linii sterujących o różnym poziomie napięcia w danym czasie), zastosowano szeregowo włączone rezystory o wartości 1 k Ω . Można zrezygnować z tych rezystorów, ale wówczas przy występowaniu jednej kolumny pozostałe należy przełączać w tryb wysokiej impedancji.

Klawiaturę matrycową najczęściej stosuje się wtedy, gdy liczba wymaganych przycisków (klawiszy) przekracza 4. Pozwala zmniejszyć liczbę wyprowadzeń FPGA lub procesora w stosunku do prostej klawiatury

ry, w której każdemu przyciskowi musi być przyporządkowane jedno wyprowadzenie. Przykładowo, aby wykonać klawiaturę z 36 klawiszami w konfiguracji klasycznej (nie w macierzy), to należałoby przeznaczyć aż 36 wyprowadzeń układu. Natomiast w konfiguracji maciercowej należy utworzyć macrycę

6×6 i wystarczy już tylko 12 wyprowadzeń. Niestety w takim połączeniu detekcja wciśniętego klawisza jest nieco trudniejsza.

Na list. 6.1 przedstawiono opis kod modułu obsługi klawiatury maciercowej. Jest on tak sparametryzowany, że można go użyć do obsługi dowolnej klawiatury, przy czym

liczba kolumn i wierszy musi zawierać się w przedziale od 2 do 8 (klawiatury mogą mieć od 4 do 64 klawiszy).

Detekcja wciśniętego klawisza następuje wskutek skanowanie kolumn (ustawianie kolejno wysokiego poziomu napięcia) i wierszy (odczyt poziomu napięcia) w celu poszukiwania zwarcia między tymi liniami. Za skanowanie odpowiedzialne są dwa czterobitowe liczniki (*cnt_rows*, *cnt_cols*). Odliczają one aż do osiągnięcia liczby kolumn lub wierszy macrycy, przy czym licznik kolumn jest inkrementowany po każdym pełnym odliczeniu licznika wierszy. Licznikom tym towarzyszą dwa 8-bitowe rejestry przesuwne (*row_comp* i *cols*). Są one użyte w funkcji liczników liczących w kodzie pierścieniowym (1 z 8). Przesuwana jest w nich jedynka od początku (bitu LSB) aż do bitu odpowiadającego liczbie kolumn albo wierszy macrycy. Rejestr *cols* jest połączony z kolumnami klawiatury (liniami sterującymi) ustawiając kolejno na każdej z nich poziom wysoki napięcia. W czasie, gdy w danej kolumnie ustawiony jest poziom wysoki, odczytywany wejściowy wektor wierszy (z linii kontrolnych) jest porównywany ze zmieniającą się wartością rejestru *row_comp*. Dokonując na tych dwóch sygnałach bitową operacją AND można wykryć wciśnięty klawisz (zauważmy, że w przypadku wciśnięcia naraz dwóch klawiszy, wciśnięcie nie jest wykrywane). Gdy podczas skanowania nastąpi zgodność wartości 1 odpowiadających sobie bitów, to ustawiany jest rejestr *pressed*. Oznacza to, że wykryto wciśnięty klawisz. Po wykryciu wciśnięcia skanowanie nie jest kontynuowane, ale rozpoczyna się od początku. Rejestr ten jest zerowany dopiero wtedy, gdy po pełnym przeskanowaniu wszystkich kolumn i wierszy

List. 6.1 Kod źródłowy opisu modułu obsługi klawiatury maciercowej

```
module matrix_keyboard
#(parameter N=4, parameter M=4)
//N - liczba kolumn, M - liczba wierszy
//maksymalnie 8x8, minimalnie 2x2
//obsługa maksymalnie 1 wciśniętego przycisku
//pomiędzy wciśnięciami nie musi następowac przerwa!
(clk, clrn, dout, dav, cols, rows, impulse);

input clk, clrn;
output reg dav;
output reg [7:0] dout;
output reg [N-1:0] cols;
input [M-1:0] rows;
input impulse;

reg [3:0] cnt_col; //licznik kolumn
reg [3:0] cnt_row; //licznik rzędów
reg [M-1:0] rows_comp; //maska dla rzędów
reg pressed; //ciągłe wciśnięcie

reg [7:0] button; //aktualnie wciśnięty przycisk

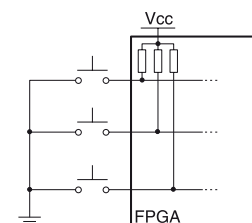
always@(posedge clk or negedge clrn)
if (!clrn) begin //reset
cnt_col <= 4'b0;
cnt_row <= 4'b0;
cols <= 8'b1;
button <= 8'b0;
pressed <= 1'b0;
rows_comp <= 8'b1;
end else if (impulse) begin
//skanowanie
if (rows_comp&rows) begin
//jeśli wykryto klawisz wciśnięty,
//reset i wysyłanie współrzędnych
cnt_row <= 8'b0;
rows_comp <= 8'b1;
cols <= 8'b1;
cnt_col <= 8'b0;
button <= {cnt_row, cnt_col};
pressed <= 1'b1;
end else begin
if (cnt_row<M-1) begin
//inkrementacja wierszy
cnt_row <= cnt_row + 1;
rows_comp <= rows_comp << 1;
end else begin
//reset wierszy i inkrementacja kolumn
cnt_row <= 4'b0;
rows_comp <= 8'b1;
if (cnt_col < N-1) begin
cnt_col <= cnt_col+1;
cols <= cols << 1;
end else begin
cnt_col <= 4'b0; //reset kolumn
cols <= 8'b1;
//jeśli przeskanowano wszystkie pozycje
//i nie wykryto wciśnięcia zresetuj
//"pressed"
//usuń znacznik wciśniętego przycisku
if ((~rows_comp|~rows))
pressed <= 1'b0;
end
end
end

reg pressed_prev; //poprzedni pressed
reg [7:0] button_prev; //poprzedni button
//wysyłanie kolejnych wciśnięć na zewnątrz

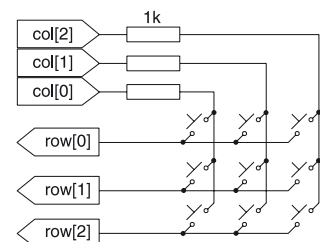
always@(posedge clk or negedge clrn)
if (!clrn) begin
pressed_prev <= 1'b0;
button_prev <= 8'b0;
dav <= 1'b0;
dout <= 1'b0;
end else begin
dout <= 8'b0;
dav <= 1'b0;
if (button_prev!=button || {pressed_prev,pressed}==2'b01) begin
dout <= button;
dav <= 1'b1;
end
end
button_prev <= button;
pressed_prev <= pressed;
end

endmodule
```

a) bezpośrednie podłączenie przycisków



b) klawiatura maciercowa 3x3



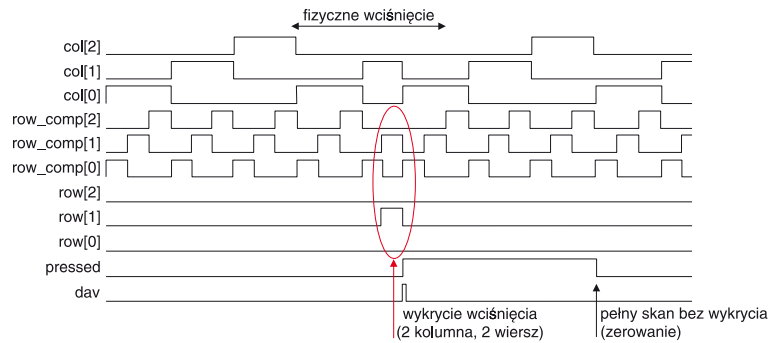
Rys. 6.1 Konfiguracje klawiatury: a) bezpośrednie dołączenie przycisków – klawiatura „klasyczna”, b) klawiatura maciercowa

szy nie wykryto wciśnięcia. Przebiegi czasowe sygnałów ilustrujących algorytm detekcji przedstawiono na **rys. 6.2**. Rejestr *button* przechowuje zawartość liczników *cnt_rows* i *cnt_cols* w momencie wykrycia wciśnięcia, a więc wskazuje na współrzędne (kolumna – wiersz) wciśniętego klawisza.

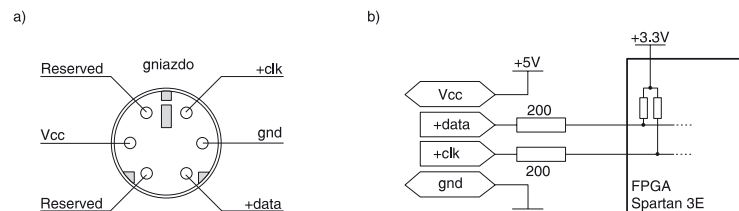
Aby umożliwić prostą współpracę tego modułu z pozostałą częścią systemu zaimplementowanego w FPGA należało opisać port wyjściowy tego modułu, składający się z 8-bitowego „adresu” wciśniętego przycisku, oraz sygnału *dav*, który sygnalizuje systemowi moment wciśnięcia przycisku. Rolą tego pierwszego spełnia rejestr *dout*. Druga pętla *always* odpowiedzialna jest za generację sygnału *dav*. Dzięki pomocniczym rejestrům *pressed_prev* i *button_prev* przechowującym wartości sygnałów *button* i *pressed* z poprzedniego cyklu zegara, możliwe jest wykrycie narastającego zbocza sygnału *pressed* oraz zmiana wartości wskazywanej przez *button*. Jeśli choć jeden z tych dwóch warunków jest spełniony, uaktualniony zostanie port wyjściowy *dout* i wygenerowany sygnał *dav*. Wskutek odpowiedniego spowolnienia próbkowania linii uzyskano efekt prostego *debouncer'a*, a więc układu omijającego fazę drgania styków klawiszy.

Należy zwrócić uwagę, że w module sparametryzowana jest tylko długość portów *cols* i *rows* oraz rejestru *rows_comp*. Długość portu wyjściowego *dout* jest 8-bitowa aby ułatwić łączenie modułu z innymi układami systemu. Również liczniki *cnt_col* i *cnt_row* mają stałą długość. Taka parametryzacja jest wystarczająca, ponieważ program syntezy automatycznie uprości model tak, aby wykorzystane zostały tylko te zasoby logiczne, które są niezbędne do jego prawidłowego działania. Po syntezie otrzymamy informację, które obwody generowania sygnałów i rejestry zostały uproszczone (skrócone lub części składowe zostały zastąpione stałymi przypisaniami).

Zastosowanie klawiatury komputerowej (zamiast własnej, matrycowej) jako części urządzenia ma szereg zalet. Po pierwsze jest to tani sposób na uzyskanie bardzo dużej liczby klawiszy w estetycznej obudowie. Po drugie, taka klawiatura ma zestandaryzowany interfejs. Kolejną zaletą jest łatwość jej obsługi. Interfejs PS/2 opiera się na liniach: zegarowej, danych, powrotnej (masy) oraz zasilania. Schemat połączeniowy oraz występowanie sygnałów na wtyczce PS/2 przedstawiono na **rys. 6.3**. Komunikacja w tym standardzie jest dwukierunkowa (linie: zegarowa i danych są typu otwarty kolektor). Zarówno komendy jak i odpowiedzi są opisane szeroko w Internecie. W tym przykładzie nie będziemy wysyłać żadnych komend do klawiatury – będziemy tylko odbierać i częściowo interpretować dane. Linie: danych i zegarowa mogą więc być traktowane jako wejście



Rys. 6.2 Przebiegi czasowe sygnałów ilustrujące działanie klawiatury matrycowej



Rys. 6.3 Gniazdo PS/2: a) rozmieszczenie sygnałów, b) sposób dołączenia do układu FPGA

```

List. 6.2 Kod źródłowy opisu modułu odbierającego dane przez interfejs PS/2
module ps2
    (clk, clrn, ps2_clock, ps2_data, dout, dav, impulse);

    input clk, clrn, ps2_clock, ps2_data, impulse;
    output reg dav;
    output reg [7:0] dout; //rejestr wyjściowy

    //obsługa transmisji interfejsu PS/2
    reg ps2_clock_prev;
    reg [10:0] sreg; //rejestr przesuwny
    reg [4:0] counter; //licznik przesunięc
    reg [12:0] timeout; //rejestr timeout'u

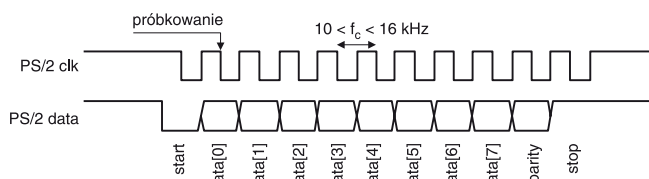
    always@(posedge clk or negedge clrn)
        if (!clrn) begin //reset
            ps2_clock_prev <= 1'b1;
            sreg <= 11'b0;
            counter <= 5'b0;
            dout <= 8'b0;
            dav <= 1'b0;
            timeout <= 13'b0;
        end else begin
            dav <= 1'b0;
            timeout <= timeout + 1;
            if (timeout >= 13'd8190) begin //timeout 125us
                ps2_clock_prev <= ps2_clock;
                sreg <= 11'b0;
                counter <= 5'b0;
                dout <= 8'b0;
                dav <= 1'b0;
                timeout <= 13'b0;
            end else begin
                //sampling wejścia
                if (impulse) begin
                    ps2_clock_prev <= ps2_clock;
                    if ({ps2_clock_prev, ps2_
clock})==2'b10) begin
                        sreg <= {ps2_
data, sreg[10:1]}; //przesuwanie rejestru
                    counter <= counter + 1'b1;
                    timeout <= 13'b0; //reset
                end
            end
            if (counter >= 5'd11) begin
                //po odliczeniu 11 przesunięc
                counter <= 5'd0; //zeruj licznik
                //kontrola poprawności (bit start,
                parzystość, bit stop)
                if (~sreg[0] & ^sreg[8:1]!=sreg[9]
& sreg[10]) begin
                    dav <= 1'b1;
                    dout <= sreg[8:1];
                end
            end
        end
    endmodule
    
```

we, wewnątrz podciągnięte do zasilania (rys. 6.3b). Standard elektryczny interfejsu PS/2 przewiduje zasilanie napięciem 5 V, ale

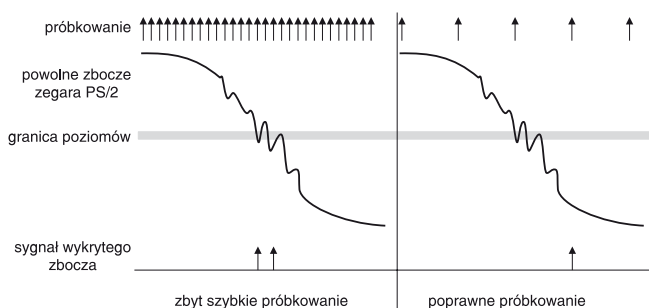
ponieważ linie: danych i zegarowa są typu otwarty kolektor, to w FPGA należy podciągnąć je do zasilania. Dodatkowo zastosowano

Tab. 6.1. Kody wybranych klawiszy

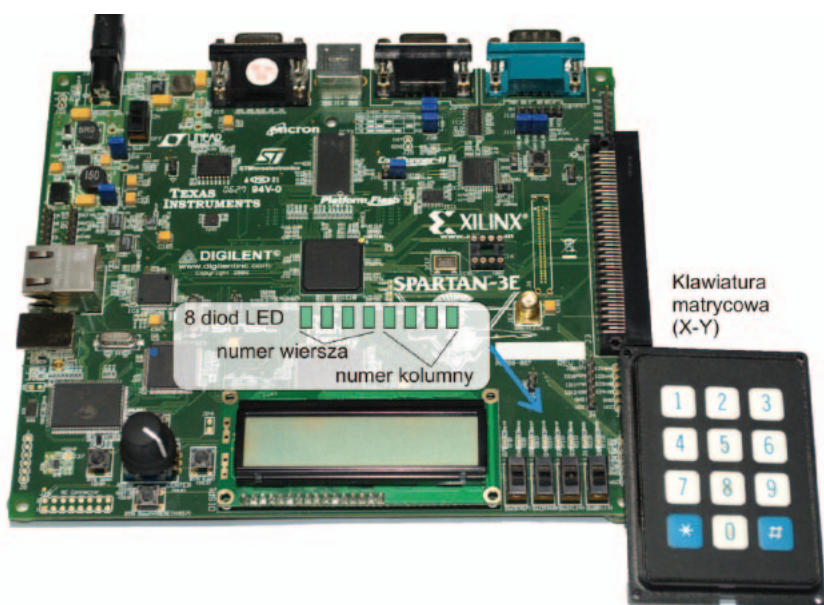
Klawisz	Kod wciśnięcia	Kod puszczenia	Klawisz	Kod wciśnięcia	Kod puszczenia	Klawisz	Kod wciśnięcia	Kod puszczenia
A	1C	F0,1C	S	1B	F0,1B	9	46	F0,46
B	32	F0,32	T	2C	F0,2C			
C	21	F0,21	U	3C	F0,3C	Shift lewy	12	F0,12
D	23	F0,23	V	2A	F0,2A	Ctrl lewy	14	F0,14
E	24	F0,24	W	1D	F0,1D	Alt lewy	11	F0,11
F	2B	F0,2B	X	22	F0,22	Enter	5A	F0,5A
G	34	F0,34	Y	35	F0,35	Spacja	29	F0,29
H	33	F0,33	Z	1A	F0,1A	Góra	E0,75	E0, F0,75
I	43	F0,43	0	45	F0,45	Lewo	E0,6B	E0, F0,6B
J	3B	F0,3B	1	16	F0,16	Prawo	E0,74	E0, F0,74
K	42	F0,42	2	1E	F0,1E	Dół	E0,72	E0, F0,72
L	4B	F0,4B	3	26	F0,26	F1	05	F0,05
M	3A	F0,3A	4	25	F0,25	F2	06	F0,06
N	31	F0,31	5	2E	F0,2E	Delete	E0,71	E0, F0,71
O	44	F0,44	6	36	F0,36	*	7C	F0,7C
P	4D	F0,4D	7	3D	F0,3D	+	79	F0,79
R	2D	F0,2D	8	3E	F0,3E	-	7B	F0,7B



Rys. 6.4 Ramka sygnałowa interfejsu PS/2 przy odbiorze danych z klawiatury



Rys. 6.5 Ilustracja problemu nadpróbkowania przy łagodnym, zakłóconym zboczach sygnału zegarowego



Rys. 6.6 Zestaw ewaluacyjny z dołączoną klawiaturą matrycową

szeregowo włączone rezystory zabezpieczające przed zwarciami o wartości ok. 100 Ω .

Ramka danych zastosowana w tym interfejsie jest bardzo podobna do ramki interfejsu RS-232. Składa się z 11 bitów (rys. 6.4). Są to kolejno: bit startu (0), 8 bitów danych, bit nieparzystości oraz bit stopu (1). Stan linii danych zmienia się przy narastających zboczach sygnału zegarowego, a więc ich odczyt powinien następować przy zboczach opadających. Po wciśnięciu klawisza, klawiatura wysyła co pewien krótki okres jego kod (jednobajtowy dla większości klawiszy). Gdy puszcza klawisz, przesłany jest jego kod poprzedzony bajtem F0 oznaczającym, że dany klawisz został puszczone. Kody wybranych klawiszy zestawiono w tab. 6.1.

Na list. 6.2 przedstawiono opis modułu odbierającego dane z klawiatury. Częstotliwość sygnału zegarowego wysyłanego przez klawiaturę może być bardzo różna (od 10 do 16 kHz) i jest to niestety sygnał asynchroniczny w stosunku do sygnału zegarowego w układzie odczytującym w FPGA. Wykorzystamy częstotliwościową przewagę zegara taktującego FPGA aby próbować sygnał zegarowy tak często, że wykrycie opadającego zbocza na tej linii będzie prawidłowe. Przy nadpróbkowaniu linii, której zbocze nie jest wystarczająco strome należy zachować ostrożność – może się bowiem zdarzyć, że zakłócenie na linii takiego zegarowego może być zinterpretowane jako zbocze, a wtedy mamy efekt podobny do efektu drgających styków klawisza. Na rys. 6.5 zilustrowano ten problem. Należy zatem nadpróbkować sygnał zegarowy, ale na tyle rzadko, że efekt związany z powolnym zboczem nie wystąpi. W tym projekcie próbkowanie zostało dobrane tak, aby sygnał zegarowy był próbowany ok. 500 razy na okres, jednak aby poprawnie zinterpretować zbocze wystarczy duże mniejsze nadpróbkowanie (np. kilkunastokrotne).

Opisywany moduł umożliwia podwójną kontrolę prawidłowości transmisji. Po pierwsze, po odliczeniu przesłanych bitów sprawdzane są bity startu i stopu oraz obliczany i sprawdzany jest bit nieparzystości. Jeśli którykolwiek wynik tych trzech kontroli jest nieprawidłowy, to transmisja jest ignorowana. Aby uniknąć błędów przy odłączaniu klawiatury „na gorąco”, skutkującej np. przerywaną w połowie transmisją, został wprowadzony licznik *timeout*, który sprawdza okres przychodzącego zegara. Jeśli w przeciągu 164 μ s (8190 okresów zegara) nie nastąpi zmiana poziomu na linii zegarowej, to moduł jest resetowany. Czas ten został dobrany arbitralnie (jest on w przybliżeniu 1.6

razy dłuższy niż najdłuższy przewidywany przez standard PS/2 okres zegara transmisji) Dzięki temu zabiegowi działanie modułu nigdy nie „zawiesza się”.

Projekt ISE (zamieszczony na płycie CD) umożliwia podłączenie zarówno klawiatury komputerowej jak i matrycowej (w tym przypadku 3x3). Po wciśnięciu klawisza jego kod jest wyświetlany na diodach led. W przypadku klawiatury matrycowej pierwsze 4 bity reprezentują numer kolumny, a kolejne 4 numer wiersza (rys. 6.6). Interpretacja danych napływających z klawiatury komputerowej została uproszczona. Ignorowany jest bajt F0 oznaczający puszczenie klawisza – stan diod jest uaktualniany z każdą poprawną trans-

misję. Konstruktor może zdecydować, czy chciałby wykorzystać tę komendę – z pewnością będzie ona potrzebna do interpretacji wciśniętych kombinacji klawiszy (np. ctrl + alt + ...). Pamiętajmy również, że kody niektórych klawiszy (np. strzałek oraz klawiszy specjalnych składają się z więcej niż jednego bajtu).

Przedstawiony projekt układu działa prawidłowo na płytce uruchomieniowej dla FPGA Spartan 3E firmy Xilinx HW-SPAR3E-SK-UNI-G.

Krzysztof Kasiński
krzysztof.kasinski@o2.pl
www.home.agh.edu.pl/kasinski

R E K L A M A

ZESTAWY LUTOWNICZE

GAS/SET
LUTOWNICA 10-60 W ZESTAW
cena: 136 zł

- * przenośna lutownica, zasilana gazem do zapalniczek
- * zapalarka wbudowana w osłonę grotu
- * czas nagrzewania grotu <40 s
- * zapas paliwa wystarcza na 90 min pracy
- * temperatura grotu (max.): 450°C
- * regulacja temperatury
- * w zestawie lutownica, groty, pojemnik na topnik, etui




GAS/PROSET
LUTOWNICA 25-125 W ZESTAW
cena: 212,50 zł

- * zapłonnik piezo
- * możliwość łatwej zmiany grotów
- * zbiornik paliwa ze stali
- * zapas paliwa na 2 godziny pracy
- * czas rozgrzewania grotu 30 s
- * maksymalna temperatura grotu 580°C
- * regulacja temperatury
- * w zestawie grot, pojemnik na topnik, etui




50 FREE
RONSON
Gaz do lutownic
RONSON
300ml
8 zł

TERAZ TANIEJ O 15%

WWW.SKLEP.AVT.PL