

Eagle w pigułce – płytko drukowana 3D Uzupełnienie

W odniesieniu do mojego artykułu w "Elektronice dla Wszystkich" nr 5/13 pod powyższym tytułem zauważyłem pewne usterki które mogą utrudniać tworzenie obrazu 3D płytki drukowanej programu Eagle. Usterki te ujawniły się na moim komputerze w systemie Windows 7 32 bit. Natomiast mój artykuł oparty był na systemie Windows XP – 32 bit. Dlatego powstało to uzupełnienie.

Usterka pierwsza: Dotyczy ona pliku **3dusrpac.dat** Plik ten zawiera przypisania obudów 3D do elementów płytkowych dokonane przez użytkownika. Dotyczy to elementów które nie są automatycznie kojarzone przez skrypt ULP. Skojarzenia elementów dokonane przez użytkownika zapisywane są w tym pliku. Jeśli błędnie przypiszemy rezystorowi znajdującemu się na płytce drukowanej np. obudowę 3D kondensatora, to błąd ten możemy usunąć poprzez edycję odpowiedniego wpisu w pliku 3dusrpac.dat. W Windows XP lokalizacja tego pliku jest u mnie w katalogu z programem Eagle czyli **C:\Program Files\EAGLE-6.2.0\ulp**.

W Windows 7 jednak tego pliku nie ma w podanej lokalizacji. Znajduje się on w zupełnie innej lokalizacji. Na moim komputerze z Windows 7 jest to lokalizacja:

C:\Users\Krzysiek\AppData\Local\VirtualStore\Program Files\EAGLE-6.2.0\ulp\3dusrpac.dat

Więc w razie konieczności korekty zawartości tego pliku należy go szukać w podanej powyżej lokalizacji. Może być też konieczne skopiowanie do tej lokalizacji katalogu **img** zawierającego obrazki podglądu 3D elementów płytkowych.

W podanej wyżej lokalizacji znajdziemy także pliki **3dcol_pl.dat** oraz **3dconf.dat**. Pierwszy plik zawiera przetłumaczone nazwy kolorów. Natomiast plik 3dconf.dat zawiera lokalizację plików do polskojęzycznej

nakładki do skryptu konwertującego ULP, co opisałem w moim artykule. Jednak w Windows 7 należy nieco zmienić tą lokalizację w pliku 3dconf.dat. U mnie w Windows 7 fragment tego pliku wygląda to jak poniżej:

```
C:/Users/Krzysiek/Desktop/  
C:/Program Files/EAGLE-6.2.0/ulp/3dcol_pl.dat  
C:/Program Files/EAGLE-6.2.0/ulp/3dlang_pl.dat
```

Plik ten znajdziemy również w lokalizacji **C:\Program Files\EAGLE-6.2.0\ulp** i zawartość tych dwóch plików w obydwóch lokalizacjach powinna być identyczna. Gdy na moim komputerze wyłączyłem systemowy **UAC** (User Account Control), to zauważyłem, że skrypt konwertujący ULP tworzy plik **3duserpac.dat** i korzysta z tego pliku oraz pliku **3dconf.dat** znajdującego się w lokalizacji:

C:\Program Files\EAGLE-6.2.0\ulp

Usterka druga: Dotyczy ona plików ***.inc** z których korzysta program POV-Ray. U mnie w Windows XP znajdowały się one w lokalizacji:

D:\Moje dokumenty\POV-Ray\v3.6\include

Jednak w Windows 7 program POV-Ray nie utworzył sobie katalogu „**include**” w lokalizacji

C:\Users\Krzysiek\Documents i nie skorzystał z plików w lokalizacji na dysku **D:** Konieczne okazało się więc skopiowanie plików ***.inc** dostarczonych z pakietem plików ***.inc** do konwersji rysunku płytki drukowanej do obrazu 3D do lokalizacji

C:\Program Files\POV-Ray\include Dopiero wówczas możliwe stało się renderowanie obrazu 3D płytki w programie POV-Ray.

Usterka trzecia: Dotyczy ona podglądu widoku 3D kojarzonych elementów. Podgląd ten ma znaczenie podczas ręcznego kojarzenia elementów kiedy skrypt konwertujący ULP nie potrafi

automatycznie skojarzyć elementów z płytki z odpowiednimi bibliotekami 3D. Na moim komputerze z Windows XP nie działał ten podgląd. Jak się okazało przyczyną tej usterki był zbędny wpis „**.pov**” w nazwie każdego pliku graficznego, na co zwróciłem uwagę w moim artykule. Pomimo tego w przypadku niektórych elementów podgląd ten i tak był niemożliwy. Natomiast w Windows 7 podgląd ten był niemożliwy pomimo usunięcia zwrotu „**.pov**” z nazwy pliku. Okazało się, że drugi błąd jest w skryptach konwertujących **3d41.ulp** i **3d50.ulp**. Pliki te trzeba otworzyć w darmowym edytorze np. PSPad editor i odszukać następujący zapis, który jest błędny (na przykładzie pliku 3d50.ulp). Wpis ten w pliku 3d50.ulp (3d41.ulp) jest zapisany w jednym wierszu.

```
dlgCell(0,0)    dlgTextView("<center><img  
src=\"'+mpd_show_pic[1]+'\"></center>");  
trzeba go skorygować do poprawnej  
postaci;
```

```
dlgCell(0,0)    dlgTextView("<center><img  
src=\"'+mpd_show_pic[1]+'\"></center>");  
Dokładniej chodzi o ten fragment skryptu:
```

```
""'+mpd_show_pic[1]+' - błędny  
""'+mpd_show_pic[1]+' - poprawny
```

Jak widzimy błąd występuje w stosowaniu znaków " - cudzysłowu i ' - apostrofu. Po skorygowaniu błędu dostępny jest podgląd widoku 3D elementów. Podobnej korekty wymaga plik 3d41.ulp. U mnie w PSPad editorze dla pliku 3d50.ulp wpis do skorygowania znajduje się w wierszu nr 2022, a dla pliku 3d41.ulp jest to wiersz nr 2019. Błąd ten pojawił się też we wcześniejszej wersji skryptu konwertującego ULP

Pakiet do tworzenia widoku 3D płytki drukowanej dla programu Eagle nie tylko jest słabo rozwijany, ale też zawiera drobne usterki. Pewne utrudnienia ujawniają się w nowszych systemach jak Windows 7, który jest pod wieloma względami bezpieczniejszy (UAC). Jednak czasami przedkłada się to na utrudnienia w korzystaniu z tego systemu lub programów zainstalowanych w tym systemie.

**Krzysztof Kawa
kaawa@wp.pl**

Eagle w pigułce – płytki drukowane 3D Metoda druga

Istnieje także druga metoda tworzenia obrazu 3D płytki drukowanej zaprojektowanej w programie Eagle, niż metoda opisana w moim artykule. Sposób ten znany jest od pewnego czasu i także bazuje on w zasadzie na darmowym oprogramowaniu. W porównaniu do poprzedniego opisu ma on pewne zalety jak łatwość i szybkość tworzenia elementów lepszą niż w programie POV-Ray. Możliwość obracania w czasie rzeczywistym widoku 3D płytki drukowanej oraz wadę podobną jak w przypadku programu POV-Ray czyli skromne biblioteki elementów.

Do utworzenia obrazu 3D płytki drukowanej z programu Eagle będziemy potrzebowali następującego oprogramowania:

Google SketchUP 8 do pobrania ze strony <http://www.sketchup.com/intl/en/download/gsu.html>

ImageMagick w wersji 6.8.5 Ja mam zainstalowaną wersję 6.7.7 do pobrania ze strony

<http://ftp.sunet.se/pub/multimedia/graphics/ImageMagick/binaries/>

Pakiet **EagleUP4.4.zip** do pobrania ze strony

<http://eagleup.wordpress.com/installation-and-setup/>

Skrypt eagleUp_export.ulp działa w programie Eagle od wersji 5.9.

Po pobraniu niezbędnego oprogramowania instalujemy program Google SketchUP 8 i Image Magic zgodnie z domyślnymi ustawieniami ich instalatorów. Następnie rozpakowujemy archiwum EagleUP4.4.zip. W archiwum tym interesują nas foldery **Eagle ULP**, **models** oraz **Sketchup Plugin**. Plik **eagleUp_export.ulp** z katalogu Eagle ULP kopiujemy do katalogu ulp programu Eagle. U mnie jest to lokalizacja **C:\Program Files\Eagle-6.2.0\ulp** Zawartość folderu **models** kopiujemy do katalogu models w katalogu głównym

Eagle czyli

C:\Program Files\Eagle-6.2.0\models

Natomiast plik **eagleUp_import.rb** z folderu Sketchup Plugin kopiujemy do lokalizacji

C:\Program Files\Google\Google SketchUp 8\Plugins

Dodatkowe modele 3D elementów znajdziemy w linku

<http://eagleup.wordpress.com/warehouse/> oraz na stronie

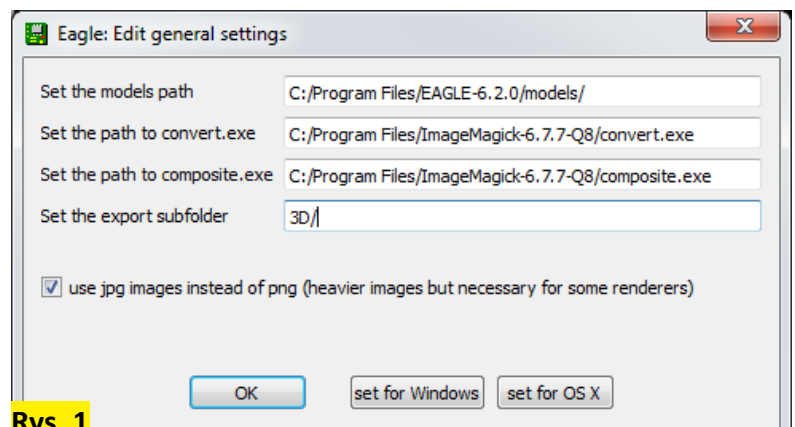
<http://sketchup.google.com/3dwarehouse/se/>

gdzie wyszukujemy

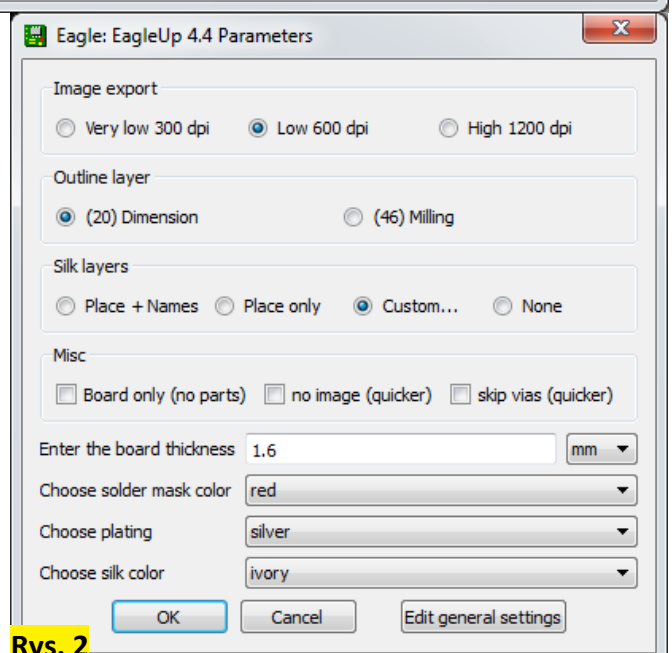
modele 3D elementów elektronicznych używając ich angielskich nazw. Po pobraniu i rozpakowaniu pobranych modeli 3D elementów kopiujemy je do katalogu models programu Eagle.

Teraz pora na wygenerowanie obrazu 3D przykładowej płytki z programu Eagle. Otwieramy wybrana płytkę drukowaną. Pozostańmy przy naszej płytce knight_rider.brd. Jeśli na płytce są wypełnienia trzeba je wytrasować poleceniem **Ratsnest**, po czym uruchamiamy skrypt **eagleUp_export.ulp** Otworzy się wówczas okno z **rysunku 1**, gdzie

w pierwszym polu podajemy ścieżkę dostępu do modeli 3D. W dwóch następnych polach podajemy ścieżkę dostępu do programów **convert.exe** i **composite.exe** z pakietu Image Magic. Ostatnie pole podaje lokalizację do katalogu w którym będą umieszczane pliki utworzone przez program. Dla wpisu 3D pliki wynikowe będą generowane w katalogu projektu dla którego tworzymy obraz 3D płytki, dzięki temu mamy od razu powiązane pliki danego projektu ze sobą w jednym folderze danego projektu. Jeśli zaznaczymy pole wyboru to domyślnym formatem plików graficznych będzie plik *.jpg który lepiej jest obsługiwany przez programy do renderingu. **Ważne jest aby ścieżka dostępu do modeli 3D i plików wynikowych była zakończona ukośnikiem w prawo „/” bo skrypt nie będzie działał prawidłowo.** Po zatwierdzeniu zmian ukaże się kolejne okno z **rysunku 2**. W oknie tym z ramce **Image export** ustawiamy rozdzielczość renderowanego obrazu. Kolejna ramka **Outline layer** pozwala na wybór która z

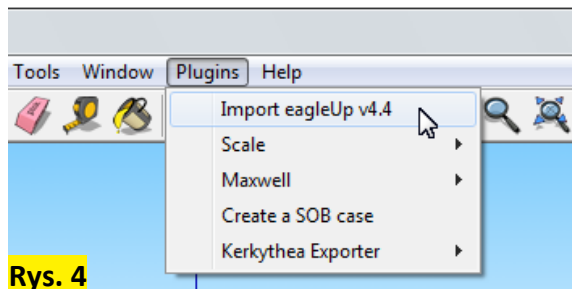


Rys. 1



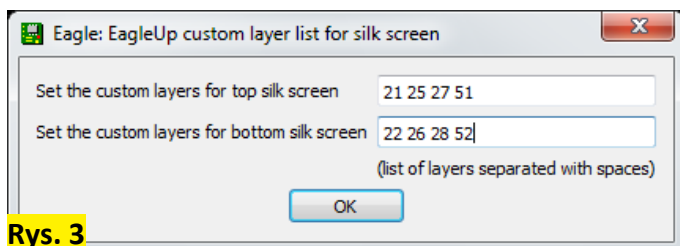
Rys. 2

warstw będzie stanowiła obrys (krawędzie) płytki w tworzonemu obrazie 3D płytki. Ramka **Silk layers** umożliwi wybór warstw opisowych, które będą widoczne na obrazie 3D płytki drukowanej, czyli obrysy elementów i ich oznaczenia (**Place + Names**), tylko obrysy elementów (**Place only**), brak opisów



Rys. 4

(**None**) oraz ustawienia własne (**Custom**). Wówczas pojawi się okno z rysunku 3 w którym będzie można wpisać numery potrzebnych warstw dodatkowych, które



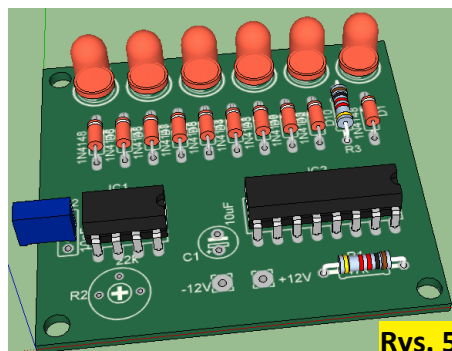
Rys. 3

mają być widoczne na warstwie górnej (Top) i dolnej (Bottom). W ramce **Misc** możemy wybrać renderowanie samej płytki – **Board only**, renderowanie bez ścieżek i opisów na płytce – **no image (quicker)** oraz renderowanie bez otworów w przelotkach – **skip vias (quicker)**. Pozostały nam jeszcze rozwijane listy i tak **Enter the board thickness** ustala grubość płytki również do wyboru w mm lub milsiach. **Choose solder mask color** – służy do wyboru koloru solder maski. **Choose plating** – wybór koloru pól lutowniczych. **Choose silk color** – wybór koloru wszystkich warstw opisowych. Po ustawieniu wszystkich parametrów płytki klikamy OK i po krótkiej chwili pojawi się komunikat, że proces generowania plików zakończył się sukcesem. Pliki te znajdują się w podkatalogu 3D w katalogu z naszym projektem. Są to pliki graficzne *.png odpowiadające poszczególnym warstwom płytki naszego projektu. Jest też plik **knight_rider.eup**, który wczytujemy do programu GoogleSketchUP. Po uruchomieniu programu GoogleSketchUP zaznaczamy na planszy projektowej sylwetkę człowieka i usuwamy ją klawiszem **Del**. Następnie z menu **Plugins** wybieramy opcję **Import eagleUp v4.4** jak na rysunku 4 i wczytujemy utworzony plik **knight_rider.eup**. Po krótkiej chwili zobaczymy widok płytki jak na rysunku 5. Jak widzimy widok ten jest nieco gorszy niż uzyskany za pomocą programu POV-Ray. Brakuje także niektórych elementów. Uzyskany obraz płytki można dowolnie

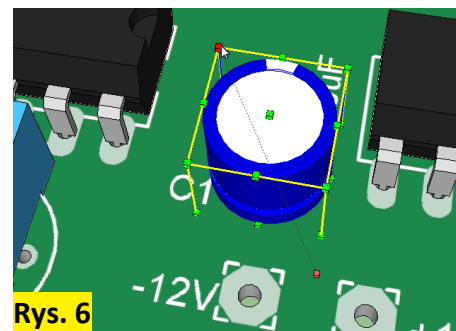
obracać i skalować. Także można obracać i skalować elementy jak na rysunku 6 bezpośrednio na płytce. Przy większym doświadczeniu możemy dorysować inne obiekty na naszej płytce czy nawet obudowę urządzenia. Jeśli wygląd płytki jest nieprawidłowy może to oznaczać, że na warstwie obrysu płytki **20 – Dimension** znajdują się dodatkowe zbędne linie lub obrys płytki nie jest zamkniętym prostokątem lub innym wielobokiem.

Sprawa brakujących elementów.

Jak widać na rysunku 5 na naszej płytce brakuje kilku elementów, kondensatora elektrolitycznego oraz potencjometru montażowego. Mianowicie po zaimportowaniu pliku *.eup do programu GoogleSketchUP otrzymujemy komunikat o brakujących elementach. Informacja ta zapisywana jest w pliku tekstowym z rozszerzeniem *.log. Wystarczy sprawdzić zawartość tego pliku i w katalogu z modelami odszukać potrzebne modele elementów 3D. Następnie zmieniamy im nazwy na nazwy brakujących elementów z pliku *.log i po kolejnym wczytaniu pliku *.eup elementy brakujące znajdują się na płytce co możemy zobaczyć na rysunku 7. Jak widać pojawił się brakujący kondensator elektrolityczny, i potencjometr montażowy, a położenie drugiego kondensatora na płytce zostało



Rys. 5

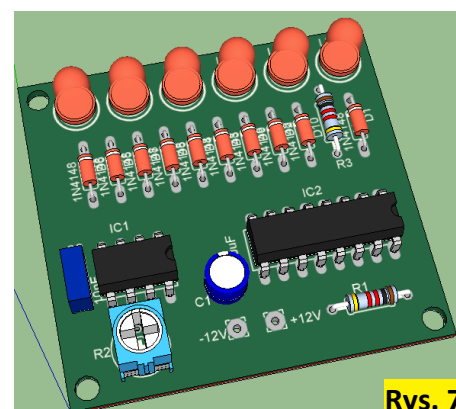


Rys. 6

ręcznie skorygowane. Odpowiednie opcje do korygowania wielkości i położenie elementów na płytce znajdziemy w menu **Tools** programu GoogleSketchUP.

Gdzie szukać brakujących elementów?

W pierwszej kolejności sprawdzamy pod hasłem „trimpot” na stronie <http://sketchup.google.com/3dwarehouse> czy nie ma potrzebnych elementów. Ja znalazłem cztery elementy i zapisałem je w katalogu „models”. Druga możliwość jest pośrednia. Mianowicie trzeba zarejestrować się na stronie <http://www.3dcontentcentral.com/default.aspx> gdzie na hasło "trimpot" znajdziemy czternaście elementów do pobrania. Tutaj jednak jest trochę pracy, gdyż nie znajdziemy na tej stronie elementów 3D w formacie GoogleSketchUP. Trzeba pobrać elementy z rozszerzeniem pliku *.stl. Następnie darmowym programem **MeshLab** ze strony <http://meshlab.sourceforge.net> importujemy plik *.stl i eksportujemy go do formatu pliku *.dae. Ten plik można natomiast zaimportować do programu GoogleSketchUP. Taki zaimportowany element to „szkielet” elementu bez kolorów, wypełnień i tekstur, który trzeba samodzielnie dopracować. Być może trzeba odpowiednio ustawić parametry konwersji plików, aby wygląd importowanego elementu był w pełni wartościowy. Osobiście tego nie sprawdzałem



Rys. 7

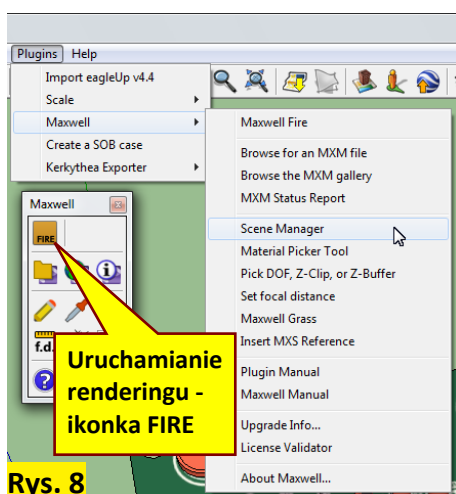
zadowolając się wspomnianymi wcześniej czterema potencjometrami montażowymi. W ostateczności osoby mające opanowaną obsługę programu SketchUP mogą utworzyć własne elementy.

Wyższa szkoła jazdy czyli wyższa jakość

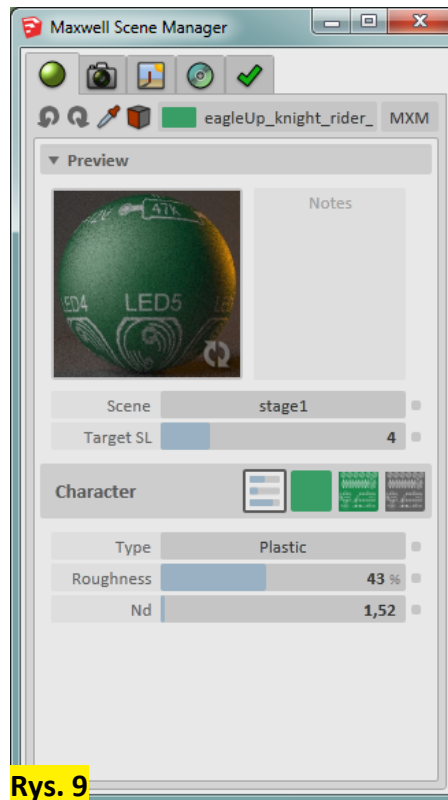
Jak wspomniałem wcześniej otrzymany widok 3D naszej płytki nie jest zbyt realistyczny. Można uzyskać lepsze efekty stosując zewnętrzne silniki renderujących dostępnych jako wtyczka do programu GoogleSketchUP. Jednym z nich jest silnik **Maxwell**, który dostępny jest w bezpłatnej wersji z pewnymi ograniczeniami, między innymi rozdzielczości renderowanego obrazu do 800 x 800 pikseli. Silnik ten pobieramy z linku: http://www.maxwellrender.com/index.php/maxwell_for_google_sketchup/download_buy i instalujemy.

Do jego poprawnego działania potrzebna jest jeszcze wtyczka Silverlight do bezpłatnego pobrania ze strony Microsoftu.

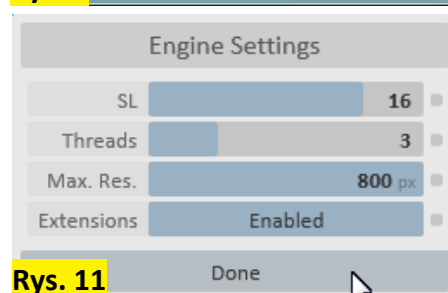
Po poprawnej instalacji silnika Maxwell i uruchomieniu programu GoogleSketchUP silnik ten dostępny jest w menu **Plugins** oraz dostępna jest paleta narzędzi tego silnika pokazana na **rysunku 8**. Na początek uruchamiamy Menadżer Scen i ustawiamy jego parametry. Klikamy pipetą na rysunku płytki drukowanej i parametry okna ustawiamy jak na **rysunku 9**. Dostępne jest też okno podglądu uzyskanego efektu. Następnie na palecie narzędzi silnika Maxwell klikamy na ikonkę **Maxwell Fire**. Otworzy się kolejne okno z **rysunku 10** gdzie można zobaczyć przebieg renderingu. Klikamy na ikonkę w kształcie koła zębatego i ustawiamy dostępne parametry jak na **rysunku 11**. Oczywiście możemy eksper-



Rys. 8



Rys. 9



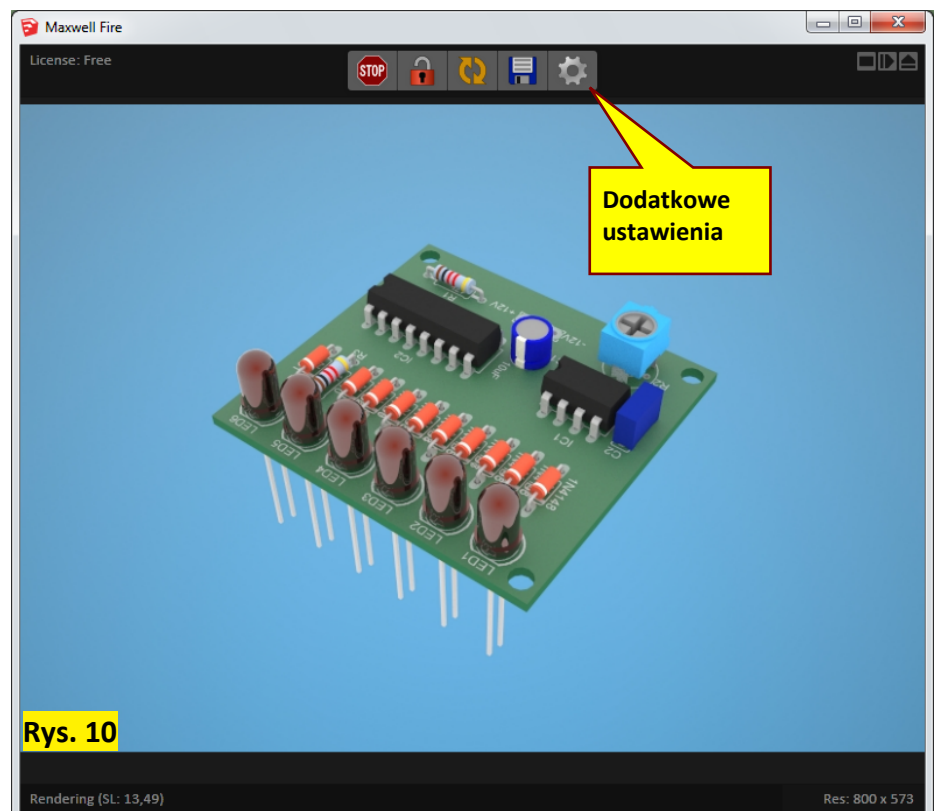
Rys. 11

mentować z ustawieniami silnika Maxwell, aby uzyskać jak najbardziej optymalny wygląd płytki drukowanej. Na moim zabytkowym komputerze klasy Pentium IV renderowanie obrazu płytki trwa kilkanaście minut. Utworzony obraz możemy zapisać w tym oknie do postaci pliku graficznego. Przy zapisywaniu pliku graficznego należy ręcznie wpisać rozszerzenie pliku *.jpg, gdyż program nie wstawia rozszerzenia pliku. Jak widać uzyskany efekt jest bardziej realistyczny niż ten z programu POV-Ray. Mimo to idealnego rozwiązania nie ma i zarówno jedna jak i druga metoda tworzenia obrazu 3D płytki drukowanej z programu Eagle ma swoje zalety i wady. Dostępny jest jeszcze drugi silnik renderujący dla programu GoogleSketchUP, o nazwie **Kerkythea** opisany w linku:

http://dangerousprototypes.com/docs/Rendering_your_3D_models_part2

Niestety na moim komputerze w systemie Windows 7 32 bit, nie wiem czemu program Kerkythea ma problemy z otwarciem pliku utworzonego w programie GoogleSketchUP i nie mam tym samym możliwości jego opisanie.

Krzysztof Kawa
kaawa@wp.pl



Rys. 10

