

Automatsko postavljanje elektroničkih komponenti u 3D elektromagnetskom alatu

Cirkveni, Andrej

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:176979>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1121

**AUTOMATSKO POSTAVLJANJE ELEKTRONIČKIH
KOMPONENTI U 3D ELEKTROMAGNETSKOM ALATU**

Andrej Cirkveni

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1121

**AUTOMATSKO POSTAVLJANJE ELEKTRONIČKIH
KOMPONENTI U 3D ELEKTROMAGNETSKOM ALATU**

Andrej Cirkveni

Zagreb, lipanj 2023.

ZAVRŠNI ZADATAK br. 1121

Pristupnik: **Andrej Cirkveni (0036532402)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: doc. dr. sc. Tomislav Marković

Zadatak: **Automatsko postavljanje elektroničkih komponenti u 3D elektromagnetskom alatu**

Opis zadatka:

Ansys HFSS je 3D elektromagnetski alat namijenjen rješavanju Maxwellovih jednadžbi na proizvoljnim geometrijama. Karakteristiku tiskane pločice definiraju geometrija pločice u kombinaciji s karakteristikama komponentata na pločici. Unos geometrije tiskane pločice je nativno riješen u alatu. Elektroničke komponente je potrebno dodati manualno što je vremenski zahtjevno i podložno greškama. Cilj projekta je napisati skriptu u Pythonu za automatsko postavljanje elektroničkih komponenti u HFSS-u na temelju PNP i BOM datoteka te usporedba sklopovskih i elektromagnetskih rezultata simulacija. Sastavni dijelovi projekta su napisati skriptu za čitanje, definirati biblioteku s modelima komponenti koje se nalaze na testnoj pločici, te napisati skriptu koja postavlja modele komponenti iz biblioteke na pločicu.

Rok za predaju rada: 9. lipnja 2023.

Sadržaj

1. Uvod	1
2. PNP datoteka	2
2.1 Čitanje PNP datoteke.....	2
3. Konfiguracija	3
3.1 Podaci u konfiguracijskoj datoteci.....	3
3.2 Nepostojeća konfiguracijska datoteka.....	4
4. Automatizacija	5
4.1 PyAEDT.....	5
4.2 Pokretanje AEDT-a.....	5
4.3 Dodavanje komponente u simulaciju.....	6
4.3.1 Preskakivanje komponenti.....	6
4.3.2 Postavljanje komponenti.....	7
5. Primjer	9
6. Zaključak	10
7. Literatura	11
8. Sažetak	12
9. Summary	13

1. Uvod

Ansyz Electronics Desktop (AEDT) simulacijski softver je 3D elektromagnetski alat namijenjen rješavanju Maxwellovih jednadžbi na proizvoljnim geometrijama, a njegova funkcionalnost uključuje i definiranje karakteristika tiskane pločice [1]. Karakteristike tiskane pločice određuju se kombinacijom geometrije pločice i karakteristika komponenata na pločici. Sama geometrija tiskane pločice može se jednostavno unijeti u AEDT, jer alat podržava nativno rješenje za unos geometrije, no elektroničke komponente na pločicu je potrebno dodati manualno što je vremenski zahtjevno i podložno greškama. Kako bi ubrzali proces, potrebno je napraviti skriptu za automatsko postavljanje elektroničkih komponenti na temelju PNP (Pick and Place) datoteka.

Cilj ovog projekta je napisati skriptu u Pythonu koja će omogućiti automatsko postavljanje elektroničkih komponenti u AEDT simulacijskom softveru čija će se implementacija temeljiti na čitanju PNP datoteke, koja sadrži, između ostalog, informacije o položaju i orijentaciji svake elektroničke komponente na pločici. Nakon što se podaci pročitaju, skripta će koristiti definiranu biblioteku modela komponenti čije će modele staviti na pločicu u AEDT.

Ovaj automatizirani pristup će značajno smanjiti vrijeme potrebno za postavljanje komponenti i minimizirati mogućnost ljudskih pogrešaka.

2. PNP datoteka

PNP su strukturirane kao .xlsx ili .xls file-ovi, gdje pojedini red opisuje jednu komponentu. U svakom stupcu se nalazi vrijednost za određeno obilježje čiji je naziv definiran u prvom redu. Nisu sva obilježja potrebna za uspješno postavljanje komponente, no ona koja jesu te koje će skripta koristiti su: “Designator”- jedinstveni naziv komponente/uređaja, “Ref-X(mm)” i “Ref-Y(mm)” - koordinate komponente x i y, “Rotation” - rotacija komponente, “Layer” - položaj komponente na z osi, može biti Top ili Bottom. Za dodavanje 3D komponente u simulaciju, potrebno je uvesti odgovarajući model, koji je implementiran kao .a3dcomp datoteka. Zato je još dodano obilježje “Simulation Model Name”. Ovo obilježje sadrži naziv .a3dcomp datoteke koja sadrži model 3D komponente za simulaciju.

2.1 Čitanje PNP datoteke

Za čitanje PNP datoteka, koje su tipa .xlsx, potrebno je koristiti xlrd Python biblioteku. Tu biblioteku je potrebno naknadno instalirati pomoću naredbe “pip install xlrd” u terminalu operacijskog sustava te se unutar koda uvozi sa “import xlrd”. Datoteka se sada može otvoriti sa metodom “open_workbook()”, a vrijednosti ćelija se dobivaju sa “cell_value(a,b)”, gdje su a i b položaj tražene ćelije u retku i stupcu. Podaci o komponenti će biti spremljeni u rječnik. U Pythonu, rječnik je podatkovna struktura koja se koristi za pohranu parova ključ-vrijednost. Kod spremanja podataka komponente, ključ će biti naziv obilježja, a vrijednost će biti vrijednost odgovarajuće ćelije. Svaka komponenta ima vlastiti rječnik kao bi se omogućio daljnji rad s njima.

```
{'Designator': 'C1',  
'Footprint': 'CAPC100X50X55L25N',  
'Center-X(mm)': '23.509',  
'Center-Y(mm)': '68.296',  
'Ref-X(mm)': '23.509',  
'Ref-Y(mm)': '68.296',  
'Pad-X(mm)': '23.971',  
'Pad-Y(mm)': '68.296',  
'Rotation': '180',  
'Layer': 'Bottom',  
'ERP Item ID': 'E120002483',  
'Manufacturer Part Number 1': 'GCM155R71H104KE02D',  
'Simulation Model Name': 'C'}
```

Slika 2.1: Primjer rječnika za jednu 3D komponentu

3. Konfiguracija

Da bi skripta znala koje komponente, varijable, PNP datoteku treba koristiti, potrebno je napraviti konfiguracijsku datoteku config. Informacije u konfiguracijskoj datoteci će biti slično organizirani kao i kod podataka komponente, tj. organizirat će se na parove ključ-vrijednost koje odvaja jedan točka zarez (;). Ti parovi će se spremirati u rječnik.

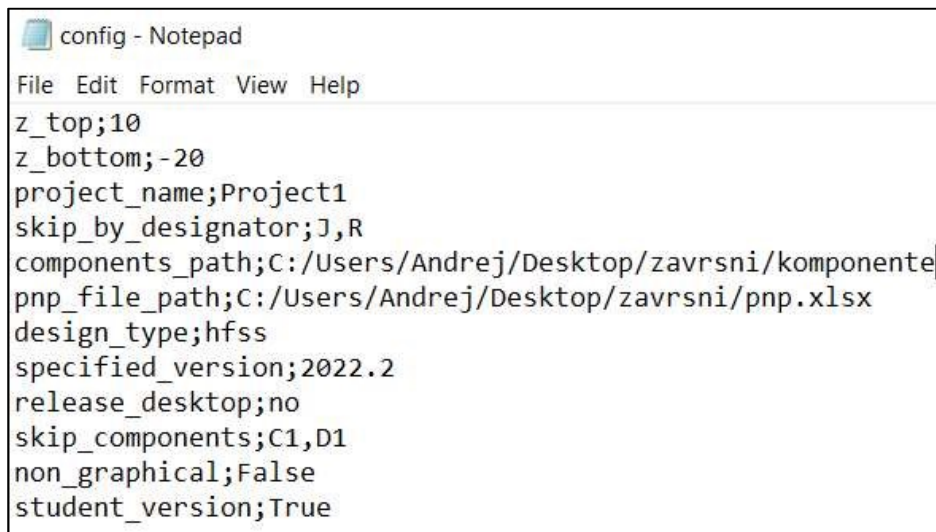
3.1 Podaci u konfiguracijskoj datoteci

Obavezni podaci koje korisnik mora unijeti su:

- Put (path) do pnp datoteke
- Put do foldera sa modelima 3D komponenti
- Ime AEDT projekta
- Verzija AEDT softvera
- Tip dizajna
 - Može biti samo hfss ili q3d.
- Pokreće li se AEDT u studentskoj verziji
 - Neobavezno ako se ne koristi studentska verzija.

Korisnik, ako želi, može dodati sljedeće konfiguracije:

- Imena komponenti koje korisnik želi da se preskoče
- Designatore komponenta koje korisnik želi da se preskoče
- Ako korisnik želi da se projekt spremi i AEDT ugasi nakon što skripta završi svoj posao
- Želi li korisnik da se AEDT uopće ne pali
 - U ovom slučaju će skripta dodati komponente u projekt bez da se AEDT pali.
- Varijable z_top i z_bottom



```
config - Notepad
File Edit Format View Help
z_top;10
z_bottom;-20
project_name;Project1
skip_by_designator;J,R
components_path;C:/Users/Andrej/Desktop/zavrsni/komponente
pnp_file_path;C:/Users/Andrej/Desktop/zavrsni/pnp.xlsx
design_type;hfss
specified_version;2022.2
release_desktop;no
skip_components;C1,D1
non_graphical;False
student_version;True
```

Slika 3.1: Primjer ispunjene konfiguracijske datoteke

3.2 Nepostojeća konfiguracijska datoteka

Ako ne postoji konfiguracijska datoteka, pri pokretanju skripte generirat će se nova konfiguracijska datoteka u kojoj su zapisane značajke koje korisnik treba ispuniti. Također će se ispisati upute o tome kako ispuniti konfiguraciju, te će skripta prekinuti rad.

4. Automatizacija

Kada su svi podaci, datoteke i biblioteke spremne, može se započeti sa automatizacijom postavljanja komponenti.

4.1 PyAEDT

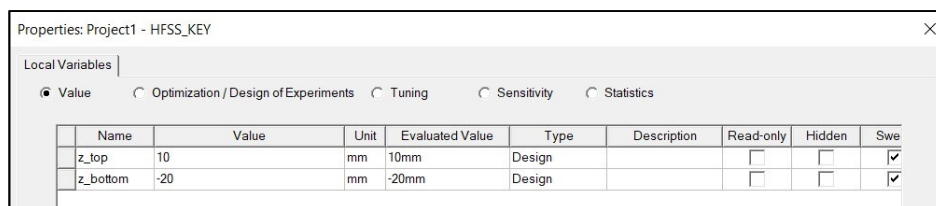
Za interakciju skripte sa AEDT softverom se koristi PyAEDT biblioteka [2]. PyAEDT je Python biblioteka koja izravno komunicira s API-jem za Ansys Electronics Desktop (AEDT) kako bi skriptiranje bilo jednostavnije. PyAEDT pruža mogućnosti skriptiranja u Ansys alatim za raspored kao što su HFSS 3D Layout i EDB. Strukture klasa i metoda PyAEDT pojednostavljaju rad uz ponovno korištenje informacija što je više moguće u API-ju. Biblioteka PyAEDT se instalira pomoću naredbe “pip install pyaedt”.

4.2 Pokretanje AEDT-a

Ovisno o tome koji tip dizajna je odabran, hfss ili q3d, simulacija će se pokrenuti inicijalizacijom klasa Hfss odnosno Q3d. Te klase za argumente primaju podatke definirane u konfiguracijskoj datoteci. To su: ime projekta, verzija AEDT softvera, pokreće li se softver u ne grafičkoj verziji i koristi li se studentska verzija. Ako je naveden krivi tip dizajna ili nije uopće, skripta prekida sa radom. Nakon toga, ako varijable z_top i z_bottom nisu definirane u AEDT-u, skripta će ih stvoriti po vrijednostima koje su definirane u konfiguracijskoj datoteci. Ako ni tamo nisu definirane, skripta će postaviti varijable z_top i z_bottom na vrijednost 0 mm..

```
app = Hfss(projectname=conf_dict["project_name"], specified_version=conf_dict["specified_version"],
non_graphical=conf_dict["non_graphical"]=="True", new_desktop_session=True,
close_on_exit=True, student_version=conf_dict["student_version"]=="True")
```

Slika 4.1: Inicijalizacija Hfss klase



Name	Value	Unit	Evaluated Value	Type	Description	Read-only	Hidden	Swe
z_top	10	mm	10mm	Design		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
z_bottom	-20	mm	-20mm	Design		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Slika 4.2: Varijable z_top i z_bottom unutar Design Properties

4.3 Dodavanje komponente u simulaciju

Za rad nad komponentama nam je potreban editor kojeg dobijemo sljedećom linijom koda [3]:

```
oEditor = app.odesign.SetActiveEditor("3D Modeler")
```

gdje je app instanca klase Hfss ili Q3d. Sljedeći koraci će se odvijati za svaku komponentu navedenu u PNP datoteci.

4.3.1 Preskakivanje komponenti

Sve komponente koje su navedene u konfiguracijskoj datoteci kao komponente koje treba preskočiti, neće se postavljati u simulaciju. Također ako komponenta već postoji u simulaciji, a nije na listi za preskok, bit će preskočena kako se ne bi pojavila dvaput. Međutim, ako komponenta postoji i nalazi se na listi za preskok, tj. ne želimo da ta komponenta bude u simulaciji, skripta će je izbrisati. Komponente za koje nije naveden Simulation Model Name ili je naveden, ali model komponente pod tim imenom ne postoji, također će biti preskočene.

Razlozi za preskakanje datoteka bit će zapisani u output datoteci koja će biti stvorena u istom direktoriju kao i skripta.



```
output - Notepad
File Edit Format View Help
Skipped designators: J, R
Skipped components: C1, D1
FID1 skipped:
  Simulation Model Name is not listed
FID2 skipped:
  Simulation Model Name is not listed
FID3 skipped:
  Simulation Model Name is not listed
FID4 skipped:
  Simulation Model Name is not listed
FL1 skipped:
  Component FL does not exist in folder: C:/Users/Andrej/Desktop/zavrzni/komponente
```

Slika 4.3: Primjer zapisa u output datoteci

4.3.2 Postavljanje komponenti

Prije umetanja komponente potrebno je stvoriti novi koordinatni sustav, jedinstven za tu komponentu, prema kojem će se ta komponenta orijentirati. Centar koordinatnog sustava bit će postavljen u odnosu na globalni koordinatni sustav, na vrijednostima Ref-X i Ref-Y iz PNP datoteke.

```
oEditor.CreateRelativeCS(  
  [  
    "NAME:RelativeCSParameters",  
    "Mode:=" , "Axis/Position",  
    "OriginX:=" , "{}mm".format(c["Ref-X (mm)"]),  
    "OriginY:=" , "{}mm".format(c["Ref-Y (mm)"]),  
    "OriginZ:=" , "0",  
    "XAxisXvec:=" , "1mm",  
    "XAxisYvec:=" , "0mm",  
    "XAxisZvec:=" , "0mm",  
    "YAxisXvec:=" , "0mm",  
    "YAxisYvec:=" , "1mm",  
    "YAxisZvec:=" , "0mm"  
  ],  
  [  
    "NAME:Attributes",  
    "Name:=" , "RelativeCS_{}".format(c["Designator"]),  
  ]  
)
```

Slika 4.4: Kreiranje koordinatnog sustava [3]

Nakon što je koordinatni sustav postavljen, komponenta će biti postavljena u odnosu na taj sustav, ovisno kako je model dizajniran, najčešće to je u središtu sustava.

```
oEditor.Insert3DComponent(  
  [  
    "NAME:{}".format(c["Designator"]),  
    "TargetCS:=" , "RelativeCS_{}".format(c["Designator"]),  
    "ComponentFile:=" , "{}/{}.a3dcomp".format(components_path, comp_file_name),  
    "IsLocal:=" , False,  
    "UniqueIdentifier:=" , "",  
    [  
      "NAME:InstanceParameters",  
      "GeometryParameters:=" , "",  
      "MaterialParameters:=" , "",  
      "DesignParameters:=" , ""  
    ]  
  ]  
)
```

Slika 4.5: Umetanje komponente u odnosu na novostvoreni koordinatni sustav [3]

Kada je komponenta postavljena na svoje mjesto, potrebno ju je pomaknuti za z_top ili z_bottom po z osi, ovisno o tome je li joj vrijednost obilježja "Layer" postavljen na Top ili Bottom. Ako je komponenta postavljena na Bottom, potrebno ju je zarotirati za 180° oko X-osi pa tek onda pomaknuti za z_bottom po z osi.

```

if(c["Layer"]=="Bottom"):
    oEditor.Rotate(
        [
            "NAME:Selections",
            "Selections:=",
            "NewPartsModelFlag:=",
        ],
        [
            "NAME:RotateParameters",
            "RotateAxis:=",
            "RotateAngle:=",
        ],
        c["Designator"],
        "Model",
        "X",
        "180deg"
    )
oEditor.Move(
    [
        "NAME:Selections",
        "Selections:=",
        "NewPartsModelFlag:=",
    ],
    [
        "NAME:TranslateParameters",
        "TranslateVectorX:=",
        "TranslateVectorY:=",
        "TranslateVectorZ:=",
    ],
    c["Designator"],
    "Model",
    "0mm",
    "0mm",
    "{}".format("z_bottom" if c["Layer"]=="Bottom" else "z_top")
)

```

Slika 4.6: Pomicanje komponente po z-osi i rotiranje u slučaju da je Layer postavljen na Bottom [3]

U konačnici, komponenta se rotira za vrijednost definiranom pod "Rotation" u PNP datoteci.

```

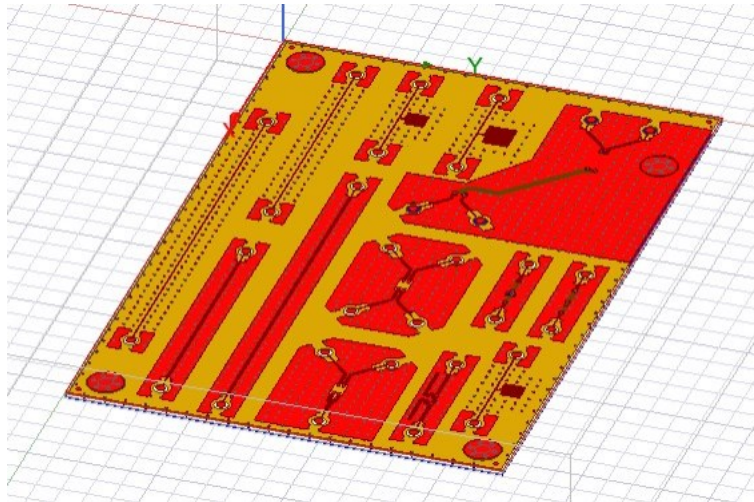
oEditor.Rotate(
    [
        "NAME:Selections",
        "Selections:=",
        "NewPartsModelFlag:=",
    ],
    [
        "NAME:RotateParameters",
        "RotateAxis:=",
        "RotateAngle:=",
    ],
    c["Designator"],
    "Model",
    "Z",
    "{}deg".format(c["Rotation"])
)

```

Slika 4.7: Rotiranje komponente [3]

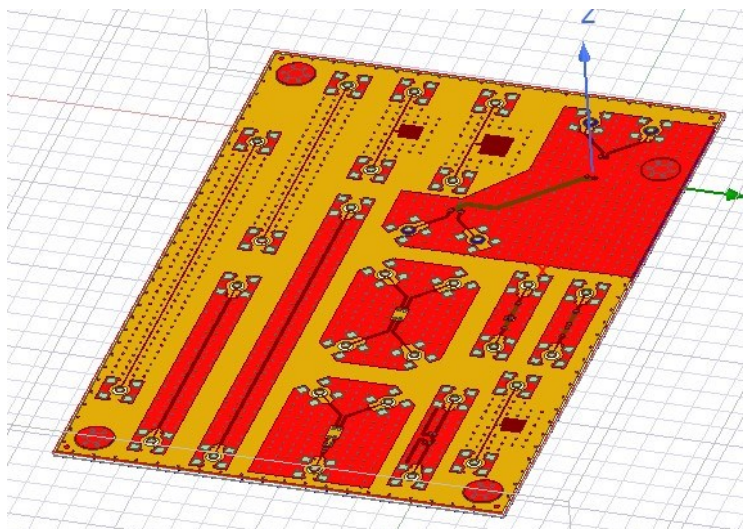
5. Primjer

Imamo pločicu na koju treba postaviti komponente (Slika 5.1).

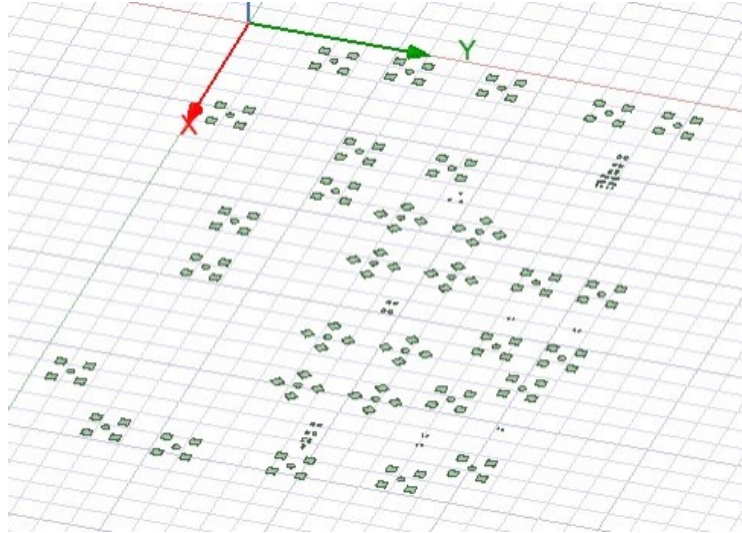


Slika 5.1: Pločica bez komponenti

Nakon pripreme potrebnih datoteka i pokretanja skripte, pločica će izgledati kao na slici 5.2.



Slika 5.2: Pločica sa automatski postavljenim komponentama



Slika 5.3: Komponente bez pločice

6. Zaključak

U ovom projektu je razvijena skripta u Pythonu za automatsko postavljanje elektroničkih komponenti u Ansys Electronics Desktop simulacijskom softveru. Skripta je implementirana na temelju čitanja PNP datoteke koja sadrži informacije o položaju i orijentaciji svake komponente na tiskanoj pločici. Automatizirani pristup postavljanju komponenti koristeći skriptu značajno smanjuje vrijeme potrebno za postavljanje komponenti i minimizira mogućnost ljudskih pogrešaka. Skripta koristi PyAEDT biblioteku za interakciju sa AEDT softverom i omogućava jednostavno skriptiranje u Ansys alatima. Implementacija automatskog postavljanja komponenti omogućava brže i pouzdanije stvaranje simulacija tiskane pločice u Ansys Electronics Desktop simulacijskom softveru, što je od velike koristi pri razvoju elektromagnetskih sustava i analizi njihovih karakteristika.

7. Literatura

[1] Electronic Design & Electromagnetics Simulation Software

Poveznica: <https://www.ansys.com/products/electronics>; pristupljeno 7. lipanj 2023.

[2] PyAEDT documentation 0.6.78 --- PyAEDT

Poveznica: <https://aedt.docs.pyansys.com/version/stable/>; pristupljeno 7. lipanj 2023.

[3] ANSYS Electronics Desktop™: Scripting Guide

Poveznica: <http://www.ece.uprm.edu/~rafaelr/inel6068/HFSS/scripting.pdf>;

pristupljeno 7. lipanj 2023.

Sažetak

Automatsko postavljanje elektroničkih komponenti u 3d elektromagnetskom alatu

U radu se razvija Python skripta za automatsko postavljanje elektroničkih komponenti u Ansys Electronics Desktop simulacijskom softveru. Skripta koristi PNP (Pick and Place) datoteku, koja sadrži informacije o položaju i orijentaciji komponenata na tiskanoj pločici. Na temelju tih podataka, skripta koristi biblioteku modela komponenata i postavlja modele na pločicu u AEDT. Skripta koristi PyAEDT biblioteku za komunikaciju s AEDT softverom. Ovaj automatizirani pristup smanjuje vrijeme postavljanja komponenata i minimizira ljudske pogreške.

Ključne riječi: Automatizacija, elektroničke komponente, 3D komponente, Ansys, AEDT, PNP, PyAEDT, Hfss, Q3d

Summary

Automatic placement of electronic components in a 3d electromagnetic tool

A Python script is being developed for automatic placement of electronic components in the Ansys Electronics Desktop simulation software. The script uses a PNP (Pick and Place) file that contains information about the position and orientation of the components on the printed circuit board. Based on this data, the script utilizes a library of component models and places the models on the board in AEDT. The script utilizes the PyAEDT library to communicate with the AEDT software. This automated approach reduces component placement time and minimizes human errors.

Keywords: Automation, electronic components, 3D components, Ansys, AEDT, PNP, PyAEDT, Hfss, Q3d