ویژگی‌های نیم رساناها: باتری خورشیدی

گروه یک: آقایان سعید شیرانی، آبتین الماسی، امیرسهیل بلوچستان‌زاده

نگارنده: سعید شیرانی

۲۶ فروردین ۱۴۰۲

هدف آزمایش

بررسی تغییرات جریان اتصال کوتاه (I_{sc}) با شدت نورفرودی، محاسبه‌ی ولتاژ مدار باز(V_{oc})، محاسبه‌ی سازه‌ی پرشدگی(FF)بررسی تغییرات جریان اتصال کوتاه(I_{sc}) با زاویه‌ی فرود(زاویه‌ی میان‌خط عمود برسطح باتری خورشیدی و پرتوی‌فرودی)

ابزار آزمایش:

باتری خورشید، ولت‌سنج، آمپرسنج، جعبه مقاومت، چراغ شش ولتی، میزچه مدرج

چگونگی انجام آزمایش:

نور چشمه را به گونه‌ای یکنواخت روی باتری خورشیدی بیندازید.

۱. جریان اتصال کوتاه I_{sc} را با بستن آمپرسنج به دوسر باتری خورشیدی(بی‌مقاومت) برحسب فاصله‌ی چشمه از آن اندازه‌بگیرید و در جدول زیر یادداشت کنید. منحنی جریان اتصال کوتاه I_{sc}، برحسب فاصله‌ی چشمه از باتری خورشیدی رسم کنید. برای خطی شدن نمودار می‌توان از شدت برحسب یک توان مناسب از فاصله رسم کنید تا رابطه خطی بدست آید.

نکته: با تغییر فاصله‌ی چشمه از باتری خورشیدی، شدت نور تابیده به باتری تغییر می‌کند.

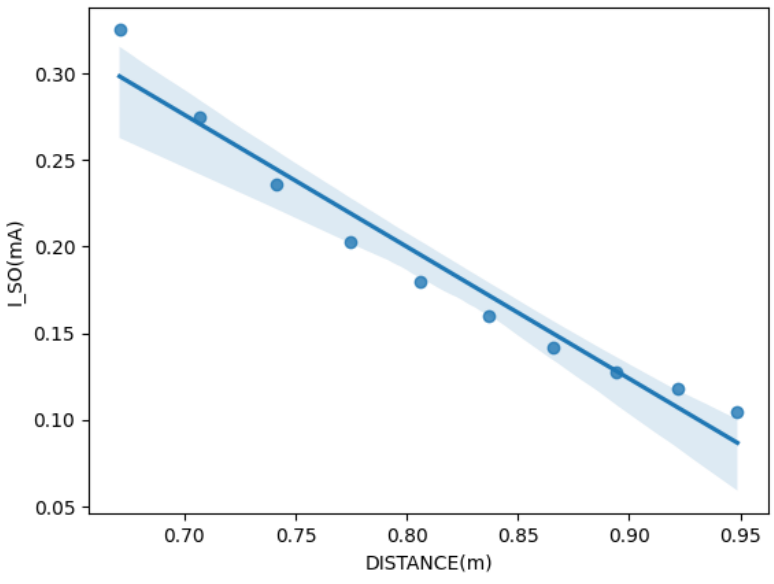
فاصله(cm) ±0.1 cm	$I_{sc}(mA)$
90	0.105 ± 0.004
85	0.118 ± 0.002
80	0.128 ± 0.002
75	0.142 ± 0.001
70	0.160 ± 0.001
65	0.180 ± 0.001
60	0.203 ± 0.001
55	0.236 ± 0.001
50	0.275 ± 0.001
45	0.325 ± 0.001

جدول ۱: تغییرات جریان اتصال کوتاه برحسب فاصله‌ی چشمه از باتری خورشیدی

I_{SC} - Distance

```
# draw chart I_sc - d
# d with respect to cm
# for get a linear chart we should insert distance in a upper power d^n
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import scipy
# create set of data
I_SO = np.array([0.105, 0.118, 0.128, 0.142, 0.160, 0.180, 0.203, 0.236, 0.275, 0.325]) # IN MILLI AMPER UNIT
DISTANCE = np.array([0.9, 0.85, 0.80, 0.75,0.70, 0.65, 0.60, 0.55, 0.50, 0.45])**0.5 # IN METER UNIT
df = pd.DataFrame({'DISTANCE(m)':DISTANCE, 'I_SO(mA)':I_SO})
#create regplot
p = sns.regplot(data=df, x=df['DISTANCE(m)'], y=df['I_SO(mA)'])
#calculate slope and intercept of regression equation we use slop and intercept in equation
slope, intercept, r, p, sterr = scipy.stats.linregress(x=p.get_lines()[0].get_xdata(),
                                                       y=p.get_lines()[0].get_ydata())
#add regression equation to plot
equation1 = 'y = ' + str(round(intercept,3)) + ' + ' + str(round(slope,3)) + 'x'

plt.legend()
plt.show()
```



برای آنکه نمودار به صورت خطی شود؛ شدت را برحسب جذر فاصله(توان $\frac{1}{2}$) رسم کردیم.

```
print(F"Eq1: {equation1}")
```

معادله‌ی خط فیت شده به شکل زیر است:

```
Eq1: y = 0.809 + -0.761x
```

۲. مدار آزمایش را مانند شکل ۳ در دستورکار ببندید.

باتری خورشیدی را در فاصله‌ی ثابت ۴۰ سانتی‌متری قرار می‌دهیم. مقدار جعبه‌ی مقاومت را به صورت لگاریتمی(۱، ۲، ۵، ۱۰، ۲۰، ۵۰، ۱۰۰، ۲۰۰، ۵۰۰، ۱۰۰۰، ۲۰۰۰، ۵۰۰۰) تغییر می‌دهیم تا حاصل ضرب IV بیشینه شود و مقادیر متناظر را در جدول ۲ یادداشت می‌کنیم. مقاومت را تاجایی بالا می‌بریم ک دیگر مقدار V تغییری نکند. این V_{oc} است. اینک پیرامون مقاومتی که IV بیشینه است، برای چند مقاومت دیگر (کوچکتر و بزرگتر) I و V را اندازه می‌گیریم. بیشینه‌ی $I * V$ را مشخص می‌کنیم. این مقادیر همان V_{mpp} و I_{mpp} است. نمودار$I - V$برحسب R را رسم کنید و یک منحنی مناسب در آن برازش کنید.

Changes in current and voltage according to the change in resistance in the resistance box			
R(Ω)	I(mA)	V	$I * v$
1	0.40 ± 0.01	$0.60mV \pm 0.1mV$	$2.4000e-04$
2	0.40 ± 0.01	$1.00mV \pm 0.1mV$	$4.0000e-04$
5	0.40 ± 0.01	$2.40mV \pm 0.1mV$	$9.6000e-04$
10	0.40 ± 0.01	$4.10mV \pm 0.1mV$	$1.6400e-03$
20	0.40 ± 0.01	$8.20mV \pm 0.1mV$	$3.2800e-03$
50	0.39 ± 0.01	$20.60mV \pm 0.2mV$	$8.0340e-03$
100	0.39 ± 0.01	$7.20mV \pm 0.1mV$	$2.8080e-03$
200	0.39 ± 0.01	$46.8mV \pm 0.5mV$	$1.8252e-02$
500	0.38 ± 0.01	$166.0mV \pm 0.5mV$	$6.3080e-02$
1000	0.39 ± 0.01	$0.39V \pm 0.01V$	$1.5210e-01$
2000	0.39 ± 0.01	$0.78V \pm 0.01V$	$3.0420e-01$
5000	0.36 ± 0.01	$1.81V \pm 0.01V$	$6.5160e-01$
10000	0.28 ± 0.01	$2.82V \pm 0.01V$	$7.8960e-01$
20000	0.16 ± 0.01	$3.43V \pm 0.01V$	$5.4880e-01$
50000	0.07 ± 0.01	$3.71V \pm 0.01V$	$2.5970e-01$
100000	0.03 ± 0.01	$3.80V \pm 0.01V$	$1.1400e-01$
200000	0.02 ± 0.01	$3.85V \pm 0.01V$	$7.7000e-02$
500000	0.01 ± 0.01	$3.87V \pm 0.01V$	$3.8700e-02$

Table 2:

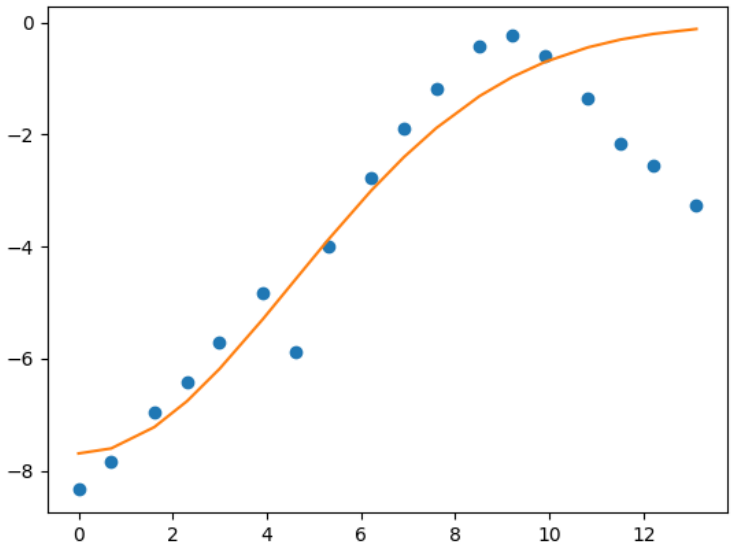
با دقت به جدول بالا و بررسی مقادیر می‌توان دریافت که بیش‌ترین مقدار ثبت‌شده برای $I * V$ در سطر سیزدهم گزارش شده‌است. به ازای $R = 10,000\Omega$ ثبت شده‌است. برای مقاومت $R = 10,000\Omega$، جریان 0.28 میلی‌آمپر و ولتاژ 0.282 ولت گزارش شده‌است. مقدار ماکسیموم $I * V$ برابر 0.7896 می‌باشد.
منحنی $I - V$ را رسم می‌کنیم و برای نمایش بهتر داده‌ها از داده‌ها لگاریتم می‌گیریم:

IV-R

now we are going to define our datasets

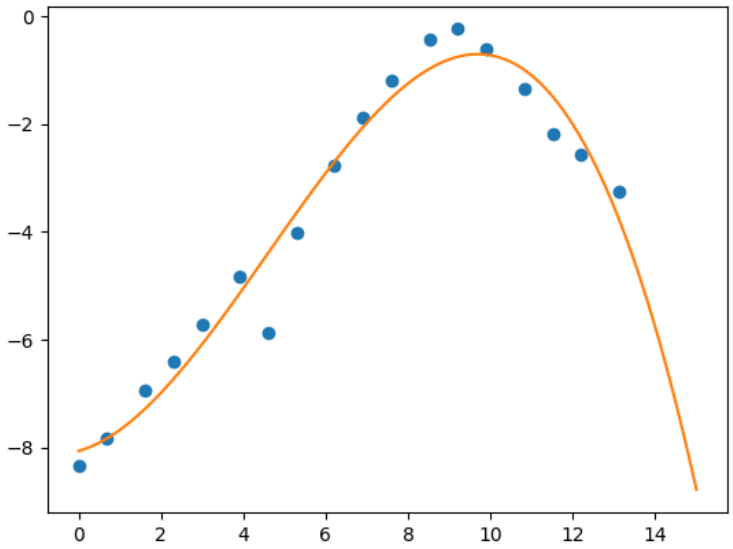
```
# IN MILLI AMPER UNIT
I = np.array([0.4, 0.4, 0.4, 0.4, 0.4, 0.39, 0.39, 0.39, 0.38, 0.39, 0.39, 0.36, 0.28, 0.16, 0.07, 0.03, 0.02, 0.01]
# IN Volt UNIT
V = np.array([0.6*0.001, 1.0*0.001, 2.4*0.001, 4.10*0.001, 8.20*0.001, 20.60*0.001, 7.20*0.001,
46.8*0.001, 166.0*0.001, 0.39, 0.78, 1.81, 2.82, 3.43, 3.71, 3.80, 3.85, 3.87])
R = np.array([1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000, 200000, 500000])
IV = I*V
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
## x-axis for the plot
xdata = np.log(R)
ydata = np.log(I*V)
# Recast xdata and ydata into numpy arrays so we can use their handy features
# Define the Gaussian function
def Gauss(x, A, B):
    y = A*np.exp(-1*B*x**2)
return y
parameters, covariance = curve_fit(Gauss, xdata, ydata)
fit_A = parameters[0]
fit_B = parameters[1]
fit_y = Gauss(xdata, fit_A, fit_B)
plt.plot(xdata, ydata, 'o', label='data')
plt.plot(xdata, fit_y, '-', label='fit')
plt.legend()
```



فیت کردن داده با تابع گاوسی ممکن نیست و خط رسم شده به داده‌ها فیت نشد. حال روش های دیگر را تست می‌کنیم. روش Cubic spline در این روش بین هر دو نقطه یک خط با معادله‌ی مرتبه دوم فیت می‌کنیم:

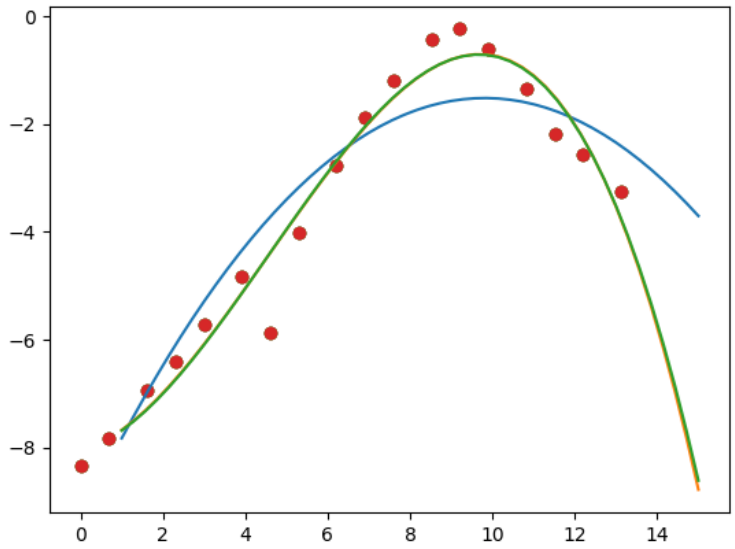
```
import matplotlib.pyplot as plt
from scipy.interpolate import UnivariateSpline
import numpy as np
xdata = np.log(R)
ydata = np.log(I*V)
s = UnivariateSpline(xdata, ydata, s=5)
xs = np.linspace(0, 15, 100)
ys = s(xs)
plt.plot(xdata, ydata, 'o')
plt.plot(xs, ys)
plt.show()
```



خط رسم شده به نمودار فیت شد.

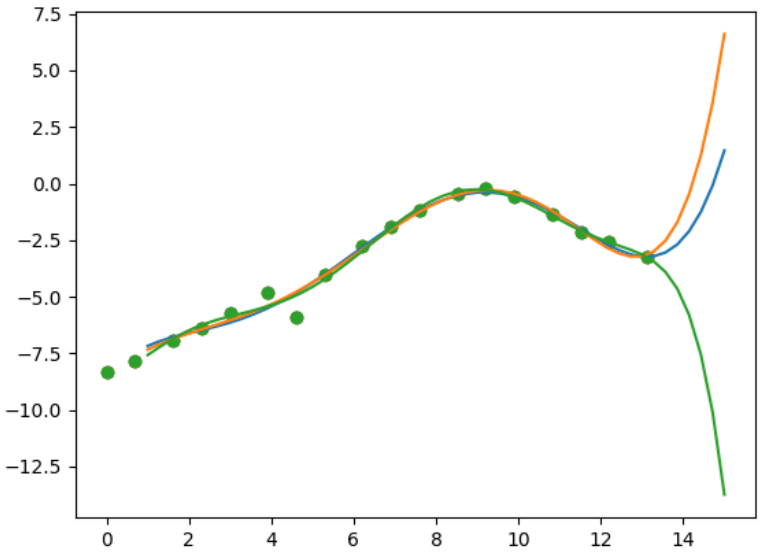
برای آنکه اطمینان بیشتری حاصل شود، توابع چند جمله‌ای از مراتب دو تا هفت را در دو پلات جداگانه به داده‌ها فیت می‌کنیم:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
## x-axis for the plot
xdata = np.log(R)
ydata = np.log(I*V)
plt.scatter(xdata, ydata)
model2 = np.poly1d(np.polyfit(xdata, ydata, 2))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model2(polyline),  '-', label='quadratic')
model3 = np.poly1d(np.polyfit(xdata, ydata, 3))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model3(polyline),  '-', label='cubic')
model4 = np.poly1d(np.polyfit(xdata, ydata, 4))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model4(polyline),  '-', label='Fourth degree')
plt.legend()
plt.show()
```



همانطور که در شکل بالا دیده می‌شود، هیچ یک از توابع مرتبه دوم و سوم و چهارم یک خط فیت شده به صورت رضایت بخش به ما ارائه نداده است.(خطوط مرتبه سوم و چهارم منطبق به یکدیگر است.)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
## x-axis for the plot
xdata = np.log(R)
ydata = np.log(I*V)
model5 = np.poly1d(np.polyfit(xdata, ydata, 5))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model5(polyline),  '-', label='Fifth degree')
model6 = np.poly1d(np.polyfit(xdata, ydata, 6))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model6(polyline),  '-', label='sixth degree')
model7 = np.poly1d(np.polyfit(xdata, ydata, 7))
polyline = np.linspace(1, 15, 50)
plt.scatter(xdata, ydata)
plt.plot(polyline, model7(polyline),  '-', label='seventh degree')
plt.legend()
plt.show()
```



خطوط رسم شده توسط هرسه معادله‌ی مرتبه پنجم و ششم و هفتم به بهترین شکل ممکن به داده‌ها فیت شد. فرمول هریک از خطوط زیر به شکل زیر است:

`quadratic Ep:`

$-0.08125x^2 + 1.595x - 9.341$

`cubic Eq:`

$-0.0139x^3 + 0.1913x^2 + 0.2119x - 8.063$

`Fourth degree Eq:`

$7.066e-05x^4 - 0.01576x^3 + 0.2066x^2 + 0.1702x - 8.043$

`Fifth degree Eq:`

$0.0006476x^5 - 0.02122x^4 + 0.2301x^3 - 0.9615x^2 + 2.139x - 8.558$

`sixth degree Eq:`

$4.977e-05x^6 - 0.001319x^5 + 0.007978x^4 + 0.02954x^3 - 0.335x^2 + 1.424x - 8.452$

`seventh degree Eq:`

$-2.781e-05x^7 + 0.001326x^6 - 0.02434x^5 + 0.2144x^4 - 0.9245x^3 + 1.773x^2 - 0.3043x - 8.306$

۳.با توجه به اندازه‌گیری‌های بخش‌های پیشین برای I_{sc} و V_{oc} و I_{mpp} و V_{mpp} سازه‌ی پرشدگی و خطای آن را محاسبه کنید.

خطاگیری در پیوست به صورت دست نویس قرار گرفته است.

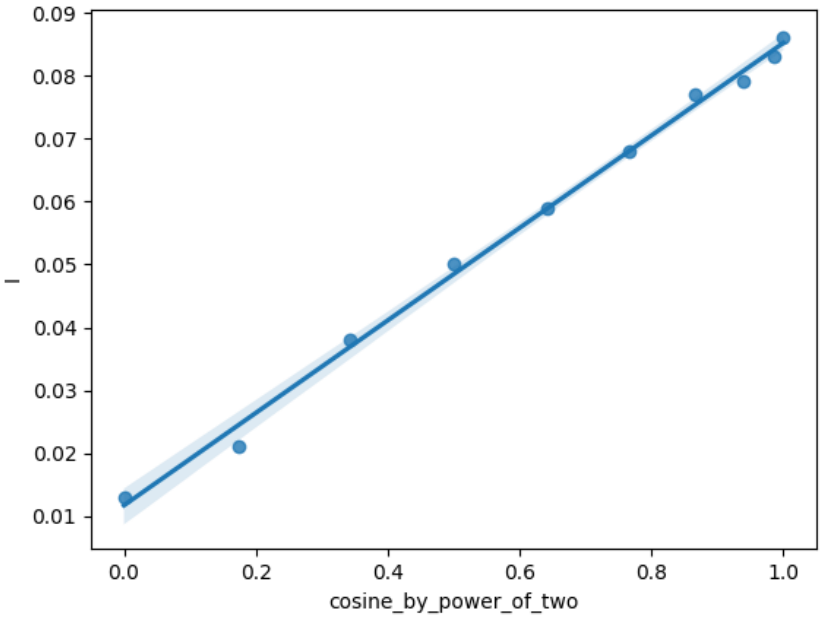
۴.با کمک یک میزچه‌ی مدرج تغییرات I را به صورت تابعی از زاویه‌ی فرود در فاصله‌ی ثابتی از لامپ اندازه‌گیری کنید. ونمودار تغییرات I بر حسب $\cos^2\theta$ را رسم کنید. آیا رابطه خطی است؟ علت آن را بنویسید.

Changes in current according to the change of landing angle										
θ	0	10	20	30	40	50	60	70	80	90
$I \pm 0.001mA$	0.086	0.083	0.079	0.077	0.068	0.059	0.050	0.038	0.021±0.003mA	0.013±0.003mA

Table 3:

I-cos^2(theta)

```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from math import pi
import pandas as pd
import scipy
I_o = np.array([0.086, 0.083, 0.079, 0.077, 0.068, 0.059, 0.050, 0.038, 0.021, 0.013])
theta = np.array([0, pi/18, pi/9, pi/6, 40*pi/180, 50*pi/180, pi/3, 70*pi/180, 80*pi/180, pi/2])
cos2 = np.cos(theta)**2
df = pd.DataFrame({'cosine_by_power_of_two':cos2, 'I':I_o})
#create regplot
p = sns.regplot(data=df, x=df.cosine_by_power_of_two, y=df.I)
#calculate slope and intercept of regression equation we use slop and intercept in equation
slope, intercept, r, p, sterr = scipy.stats.linregress(x=p.get_lines()[0].get_xdata(),
                                                       y=p.get_lines()[0].get_ydata())
#add regression equation to plot
equation2 = 'y = ' + str(round(intercept,3)) + ' + ' + str(round(slope,3)) + 'x'
# plt.plot(cos2, I_o)
plt.legend()
plt.show()
```



معادله‌ی خط فیت شده به صورت زیر است:

```
print(F"Eq2: {equation2}")
```

```
Eq2: y = 0.025 + 0.065x
```

پرسش‌ها

۱. چگونگی ساخت نیم‌رساناهای گونه‌ی n و گونه‌ی p را شرح دهید.

پاسخ برای تشکیل نیم‌رسانا نیاز به حداقل دوگونه اتم داریم. یک گونه از این اتم‌ها باید دارای t الکترون ظرفیت (A) و دیگری باید شامل $t+1$الکترون ظرفیت (B)باشد. در این صورت، اگر در یک شبکه از اتم‌های A و B، به طور تکراری، یک اتم B با t اتم A پیوند برقرار کند؛اتمB دارای یک الکترون ظرفیت آزاد می‌ماند. این شبکه‌ی تشکیل شده یک نیم‌رسانای نوع n است.

به طور مشابه می‌توان، یک گونه از این اتم‌ها باید دارای t الکترون ظرفیت (A) و دیگری باید شامل $t-1$الکترون ظرفیت (اتم C باشد. در این صورت، اگر در یک شبکه از اتم‌های C و A، به طور تکراری، یک اتم C با $t-1$ اتم A پیوند برقرار کند؛اتم C در شبکه، یک حفره تشکیل می‌دهد. شبکه‌ی تشکیل شده یک نیم‌رسانای نوع p است.

با ترکیب این سه مدل اتم با یکدیگر، این حفره‌ها و الکترون‌ها تشکیل یک نیم رسانا می‌دهند.

۳. خطاهای موجود در آزمایش را بیان کنید و در صورت امکان راه حلی برای کاهش آن‌ها بیابید.

پاسخ خطای آزمایشگر، خطای دستگاه ولت‌متر و آمپرسنج، تغییر مقاومت در سیم‌های بکار رفته.