

Android Application for an RC Charging and Discharging Circuit



K. BASKAR

An RC circuit is an electric circuit consisting of a resistor and a capacitor. This article describes the simple RC charging and discharging circuit with an Android application as shown in Fig. 1. By using this Android application, you can calculate capacitor charging and discharging voltages and currents, and capacitor charging and discharging times with graphical outputs. The program is written in Java programming language. Achartengine-1.1.0

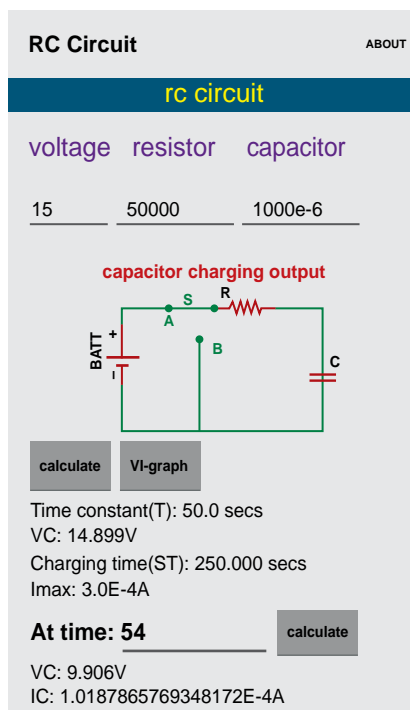


Fig. 1: RC Android application

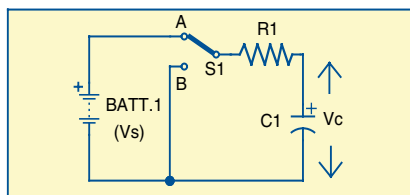


Fig. 2: Capacitor charging circuit

graph library is used to simulate the graph output.

RC charging circuit

The RC charging circuit is shown in Fig. 2. In this circuit, the resistor and capacitor are connected across the battery through mechanical switch S1. When switch S1 is closed, current from the battery passes through the resistor and, gradually, charges the capacitor until the capacitor voltage equals battery voltage ($V_c = V_s$).

At time $t = 0$, the voltage gradually builds up across the capacitor as shown in Fig. 3 and the current

Calculation of various parameters

Time constant $\tau = R \times C$

Where τ = time constant, $R = R1$ = resistor and $C = C1$ = capacitor

Voltage across the charging capacitor $V_c = V_s(1 - e^{-t/RC})$

Where V_c = voltage across the charging capacitor, V_s = supply voltage, t = elapsed time

Voltage across the discharging capacitor $V_c = V_s(e^{-t/RC})$

RC = Time constant of the charging circuit

Current through the capacitor $I_c = \frac{V_s}{R} e^{-t/RC}$

Maximum current in the capacitor $I_{max} = \frac{V_s}{R}$

Where I_c = current through the capacitor, R = resistor, t = elapsed time and RC = time constant of the charging circuit

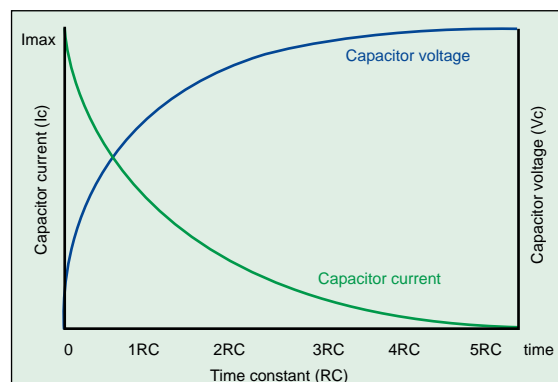


Fig. 3: A typical capacitor charging output graph

gradually decreases due to decrease in movement of electrons as shown in Fig. 3. The rate of charging is described in terms of time constant, denoted by $\tau = R \times C = R1 \times C1$. In the application, V_c is capacitor voltage, I_{max} is maximum current in the capacitor and I_c is capacitor current.

Formulae to calculate various parameters are given in the box.

For time $t = 1\tau$ (one time constant), $V_c = V_s(1 - e^{-1}) = 0.63V_s$

That is, the capacitor has been charged to 63 per cent of its final value. After 5τ (or five times constant), the capacitor will be fully charged.

For example,

If $R1 = R = 50k\text{-ohms}$, $V_s = 15V$, $C1 = C = 1000\mu F$, time constant $\tau = 50k \times 1000e^{-6} = 50.0$ seconds as shown in Fig. 1.

For time $t = 5\tau$, capacitor is fully charged. We need to calculate 5τ to know charging time t of the capacitor.

Using the formula given in the box, time $t = 5(50k \times 1000e^{-6}) = 250$ seconds, we get $V_c = 14.899V$, $I_{max} = 0.003A$ and $I_c = 2.0213\mu A$

At $t = 54$ seconds, $V_c = 9.906V$ and $I_c = 1.0187865e^{-4}$

RC charging output graph in the Android application for the example is shown in Fig. 4.

In the output graph, the white line indicates the gradual decrease in current across the capacitor, and the yellow line indicates the gradual increase in voltage across the capacitor. This output waveform has one X axis and two Y axes. Left-side Y axis indicates the capacitor current

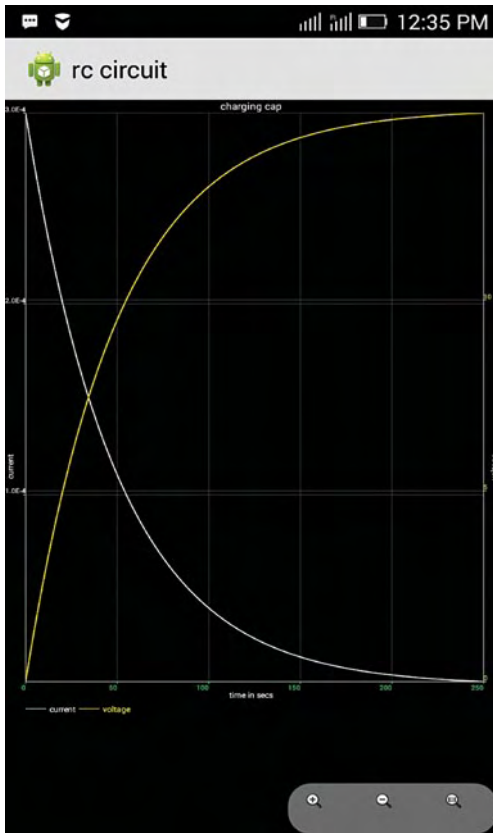


Fig. 4: Charging capacitor output graph on Android

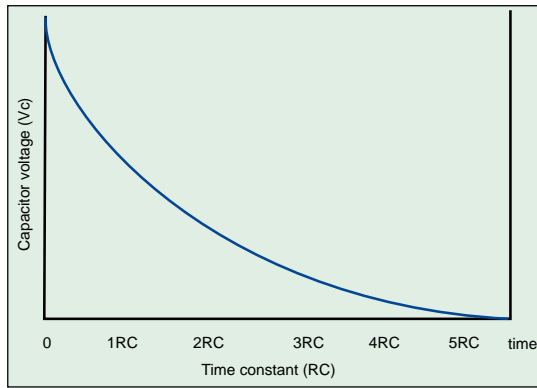


Fig. 6: Capacitor discharging voltage graph

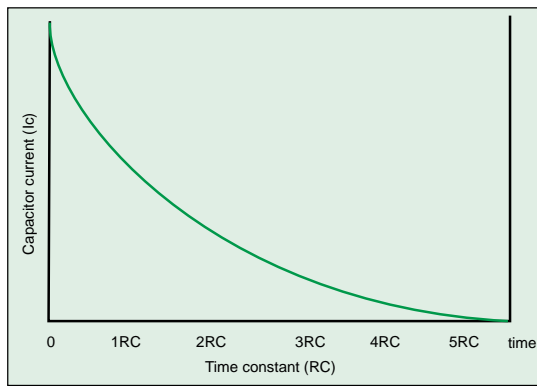


Fig. 7: Capacitor discharging current graph

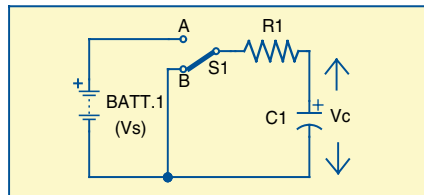


Fig. 5: Capacitor discharging circuit

and the right-side Y axis denotes the capacitor voltage. X axis indicates the charging time of the capacitor or time constant of the capacitor.

RC discharging circuit

In the discharging circuit, mechanical switch S1 is turned to position B and the battery is left open as shown in Fig. 5. During this time, the charge stored in the capacitor starts to discharge through resistor R1 until charge in the capacitor equals zero (Q=0).

From Fig. 6, at time t=0, Vc gradually decreases because accumulation of the positive charge and the negative charge freely moves away from the plates of the capacitor. So, the potential difference between the

plates decreases.

As shown in Fig. 7, from time t=0, current across the capacitor gradually decreases. In other words, at t=0, the battery is disconnected. So, stored electrostatic charges tend to freely move from the plates through the resistor. Therefore voltage (potential difference between two plates) across the two plates decreases. In the discharging circuit, both voltage and current gradually decrease.

Voltage across the capacitor and current in the capacitor are calculated using the formulae given in the box.

Time constant of the discharging capacitor is calculated using:

$$\tau = R \times C = R1 \times C1 = 50k \times 1000e^{-6} = 50 \text{ seconds}$$

For time t=5τ (five times constant), the capacitor is fully charged. So, we need to calculate 5τ to know the discharging time of the capacitor; time t=5(50k × 1000μ) = 250 seconds.

Capacitor maximum current Imax = 0.003A

Voltage across the discharging

capacitor is calculated using:

At time t = 0, capacitor voltage = 15V; after 250 seconds, the capacitor has almost zero voltage:

$$V_c = V_s(e^{-t/RC}) = 15(e^{\frac{-250}{(50k \times 1000\mu)}}) = 0.101V$$

Current across the discharging capacitor Ic = 2.0213μA

At 54 seconds, the capacitor discharging voltage and capacitor discharging current are:

$$V_c = 5.093V \text{ and } I_c = 1.0187865e^{-4}$$

Voltage and current graphs of a discharging capacitor on Android are shown in Figs 8 and 9, respectively.

Development of Android application

First, you need to install Java Development Kit (JDK) version 1.7.0 or above on your Windows computer. We have tested this on Windows 8.1, 64-bit. Configuration of JDK is included in this month's EFY DVD accompanying EFY Plus magazine.

Next, you need to install Android Studio version 1.1.0. Details for setting up Android Studio, configuration of Software Development Kit (SDK) manager and Android Virtual Device have been included in the DVD.

Importing the Android project. To import the Android project, click file tab→Import as shown in the Fig. 10.

Click Android→Existing Android Code into Workspace as shown in Fig. 11.

Click Browse and navigate your Android project (rcckt1). Click OK as shown in Fig. 12.

Program compilation. Click Android project folder; right-click on src folder→select run→. Click Run as Android application and select your phone from the dialog box or select the An-

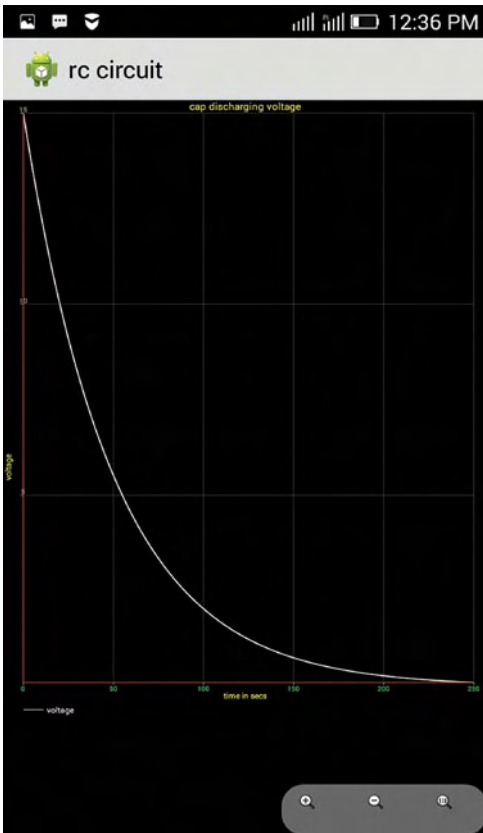


Fig. 8: Capacitor discharging voltage graph

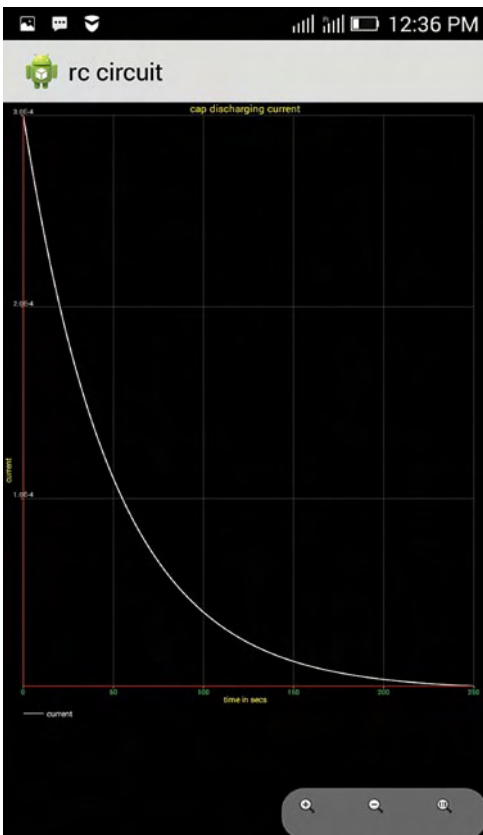


Fig. 9: Capacitor discharging current graph

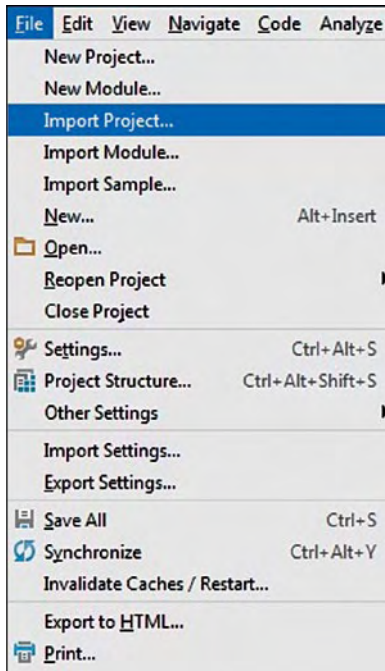


Fig. 10: Importing the project

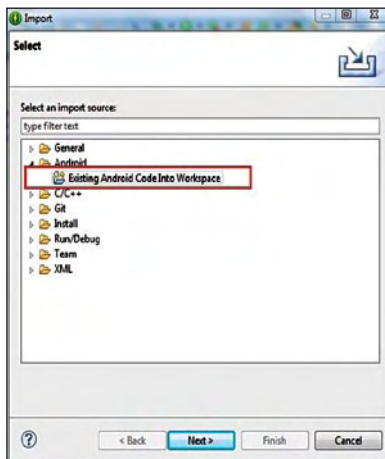


Fig. 11: Selecting the existing code

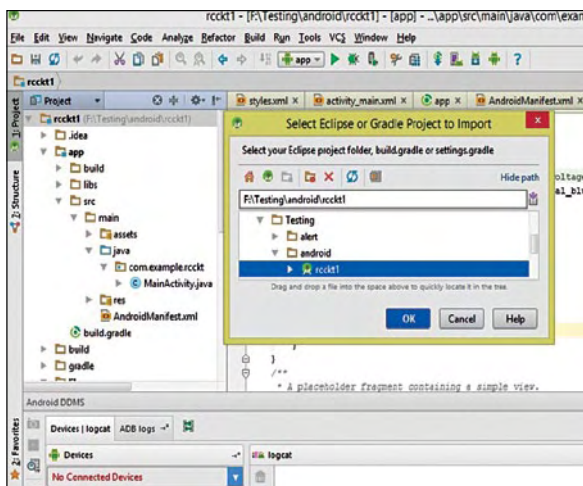


Fig. 12: Selecting directory

EFY Note

The source code of this project is included in this month's EFY DVD and is also available for free download at source.efymag.com

droid emulator that you have created. To run the emulator, you need to install Intel x86 emulator accelerator (HAXM installer) from SDK manager.

Compile the program by clicking Rebuild Project under Build menu. Once it is compiled successfully, you can connect your Android phone to your computer. Before connecting, select USB Debugging under Developer Options on your device. After connecting, you will see the mobile device model on the bottom-left side of the screen. Run the application by selecting Run app under Run menu. You can see the app on your Android device. This app is tested on 4.1.2 Jelly Bean and 4.4 Kitkat Android operating systems.

By now your app would have transferred to your Android phone, so you can disconnect from the computer and run the app. app-debug.apk is automatically generated under rcckt1/app/build/apk folder. Copy and transfer this apk file to another Android device.

apk installer software is available in almost all latest Android smartphones. So, you can install it to run it. To publish the app on Google Play Store, you should select Generate Signed APK... under Build menu.

Minimum required operating system is Android 3.0 Honeycomb (API level 11), and it works up to Android 5.0 Lollipop (API level 21).



K. Baskar is ME from J.J. College of Engineering, Tiruchirappalli. He is working as software developer at Netsec Solution, Chennai. His interests include space science, computer programming and digital art