

# Practical: 7

**AIM:** What is BroadcastReceiver? Difference between service and BroadcastReceiver. Create an Alarm application by using service & BroadcastReceiver.

Submitted By: Jesika Pargi  
Enrollment number: 21012022030



**Ganpat  
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel  
College of  
Engineering**

Department of Computer  
Engineering/Information Technology

### What is BroadcastReceiver?

In Android, broadcasts are the system-wide events that occur due to changes in system state, like LOW\_BATTERY, incoming or outgoing calls, network connectivity change, device reboot, etc. These are some of the system events that require us to perform some action. Android's system sends broadcasts when various system events occur.

BroadcastReceiver	Service
Broadcast intents are usually delivered to all BroadcastReceivers registered for that intent.	Services are only sent to one matching Service instance.
BroadcastReceiver is used when you want to fire some stuff or code during some event.	Service is used when you want to do something in background, any long running process can be done using Service in Background.
Broadcast receivers are limited by maximum amount of time.	Service are run in background long period of time.
Usually system will send some info which can be received by your app.	If you want to do something in background. this will be running always in background even if the application closed.

### activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintTop_toTopOf="parent">

        <androidx.appcompat.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Alarm Application"
                android:textSize="18dp"
                android:textStyle="bold"/>

        </androidx.appcompat.widget.Toolbar>
```

## Practical: 7

---

```
</com.google.android.material.appbar.AppBarLayout>

<androidx.core.widget.NestedScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_constraintTop_toBottomOf="@id/appbar"
    app:layout_constraintStart_toStartOf="parent"
    android:layout_marginTop="60dp"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <com.google.android.material.card.MaterialCardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="20dp"
            android:layout_marginTop="30dp"
            android:layout_marginEnd="20dp"
            app:cardCornerRadius="15dp"
            app:cardElevation="10dp"
            android:layout_marginBottom="40dp">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical">

                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="200dp"
                    android:src="@drawable/alarm"
                    android:scaleType="fitXY"/>

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:backgroundTint="#EAEFEA"
                    android:orientation="vertical"
                    android:padding="20dp">
                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="Create Alarm Time"
                        android:textStyle="bold"
                        android:textSize="15dp"/>

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginTop="10dp"
                        android:layout_marginStart="10dp"
                        android:text=
                        "By pressing buttons, Alarm can be created and cancelled."/>
                </LinearLayout>
            </MaterialCardView>
        </LinearLayout>
    </NestedScrollView>
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:text=
    "Check Current time by looking below real digital clock."
    android:layout_marginTop="10dp"/>

<TextClock
    android:id="@+id/show_time_tv"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Show Time Here"
    android:textStyle="bold"
    android:layout_marginTop="10dp"
    android:textSize="15dp"/>

<com.google.android.material.button.MaterialButton
    android:id="@+id/btn_set_alarm"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:cornerRadius="25dp"
    android:layout_marginTop="10dp"
    app:icon="@drawable/ic_baseline_add_alarm_24"
    android:text="Create Alarm"
    android:textStyle="bold"
    android:textAllCaps="false"/>

</LinearLayout>

</LinearLayout>

</com.google.android.material.card.MaterialCardView>

<com.google.android.material.card.MaterialCardView
    android:id="@+id/cancel_alarm_card_view"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginEnd="20dp"
    app:cardCornerRadius="15dp"
    app:cardElevation="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="fitXY"
            android:src="@drawable/alarm" />
```

## Practical: 7

---

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:backgroundTint="#EAEFEA"
    android:orientation="vertical"
    android:padding="20dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Set Alarm Time"
        android:textStyle="bold"
        android:textSize="15dp"/>

    <TextView
        android:id="@+id/set_alarm_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="Time"
        android:textStyle="bold"
        android:textSize="15dp"/>

    <com.google.android.material.button.MaterialButton
        android:id="@+id/btn_cancel_alarm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cornerRadius="25dp"
        android:layout_marginTop="20dp"
        app:icon="@drawable/ic_baseline_alarm_off_24"
        android:text="Cancel Alarm"
        android:textStyle="bold"
        android:textAllCaps="false"/>

</LinearLayout>

</LinearLayout>

</com.google.android.material.card.MaterialCardView>
</LinearLayout>

</androidx.core.widget.NestedScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

### MainActivity.kt

```
package com.example.madpractical7_21012022030

import android.app.AlarmManager
import android.app.PendingIntent
import android.app.TimePickerDialog
import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.TextClock
import android.widget.TextView
import com.google.android.material.button.MaterialButton
import com.google.android.material.card.MaterialCardView
import java.util.*

class MainActivity : AppCompatActivity() {
    var mili:Long=0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val CanceAlarmCardView = findViewById<MaterialCardView>(
            R.id.cancel_alarm_card_view)
        val BtnCreateAlarm = findViewById<MaterialButton>(R.id.btn_set_alarm)
        val SetAlarmTime = findViewById<TextView>(R.id.set_alarm_time)
        val BtnCancelAlarm = findViewById<MaterialButton>(R.id.btn_cancel_alarm)
        val clockTC = findViewById<TextClock>(R.id.show_time_tv)

        clockTC.format12Hour = "hh:mm:ss a"

        CanceAlarmCardView.visibility=View.GONE

        BtnCreateAlarm.setOnClickListener {
            var cal:Calendar = Calendar.getInstance()
            var hour = cal.get(Calendar.HOUR_OF_DAY)
            var min = cal.get(Calendar.MINUTE)
            val tpd = TimePickerDialog(this,TimePickerDialog.OnTimeSetListener(function = {
                view, h, m ->
                    mili=getMillis(h,m)
                    setAlarm(getMillis(h,m),"Start")
                    CanceAlarmCardView.visibility=View.VISIBLE
                    SetAlarmTime.text=h.toString()+":"+m.toString()
            }),hour,min,false)
            tpd.show()
        }

        BtnCancelAlarm.setOnClickListener{
            setAlarm(mili,"Stop")
            CanceAlarmCardView.visibility=View.GONE
        }
    }

    fun setAlarm(millisTime: Long, str: String)
```

## Practical: 7

---

```
{
    val intent = Intent(this, AlarmBroadcastReceiver::class.java)
    intent.putExtra("Service1", str)
    val pendingIntent =
        PendingIntent.getBroadcast(applicationContext, 234324243, intent, 0)
    val alarmManager = getSystemService(ALARM_SERVICE) as AlarmManager
    if(str == "Start") {
        alarmManager.setExact(
            AlarmManager.RTC_WAKEUP,
            millisTime,
            pendingIntent
        )
    }else if(str == "Stop")
    {
        alarmManager.cancel(pendingIntent)
        sendBroadcast(intent)
    }
}

fun getMillis(hour:Int,min:Int):Long
{
    val setcalendar = Calendar.getInstance()
    setcalendar[Calendar.HOUR_OF_DAY] = hour
    setcalendar[Calendar.MINUTE] = min
    setcalendar[Calendar.SECOND] = 0
    return setcalendar.timeInMillis
}
}
```

### AlarmBroadcastReceiver.kt

```
package com.example.madpractical7_21012022030

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent

class AlarmBroadcastReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context, intent: Intent) {
        val str1 = intent.getStringExtra("Service1")
        if(str1 == "Start" || str1 == "Stop")
        {
            val intentService = Intent(context, AlarmService::class.java)
            intentService.putExtra("Service1", intent.getStringExtra("Service1"))
            if(str1 == "Start")
            {
                context.startService(intentService)
            }
            else if(str1=="Stop") {
                context.stopService(intentService)
            }
        }
    }
}
```



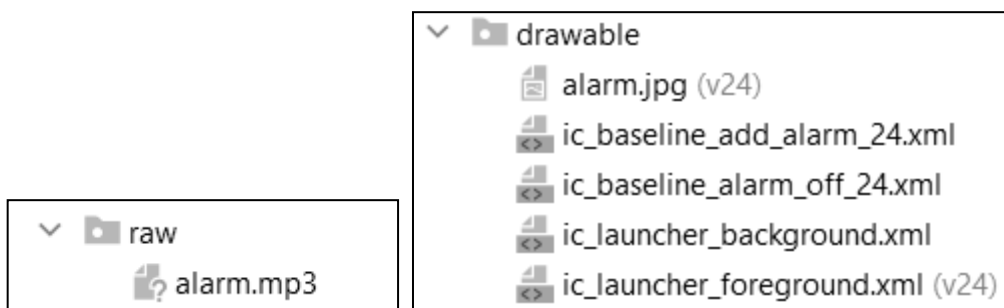
### AlarmService.kt

```
package com.example.madpractical7_21012022030

import android.app.Service
import android.content.Intent
import android.media.MediaPlayer
import android.os.IBinder

class AlarmService : Service() {
    var mp: MediaPlayer? = null
    override fun onBind(intent: Intent): IBinder {
        TODO("Return the communication channel to the service.")
    }
    override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {
        if(intent != null) {
            mp = MediaPlayer.create(this, R.raw.alarm);
            mp?.start()
        }
        return START_STICKY
    }
    override fun onDestroy() {
        mp?.stop()
        super.onDestroy()
    }
}
```

### Resource:



## Practical: 7

### Output:

