

activity\_main:

```
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2   xmlns:app="http://schemas.android.com/apk/res-auto"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:id="@+id/main"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   android:orientation="vertical"
8   tools:context=".MainActivity"
9   android:padding="50dp"
10  android:background="#2D142C">
11    <Chronometer
12      android:id="@+id/textTime"
13      android:layout_width="wrap_content"
14      android:layout_height="wrap_content"
15      android:textSize="80sp"
16      android:layout_gravity="center"
17      android:textColor="#C72C41"/>
18
19    <androidx.appcompat.widget.AppCompatButton
20      android:id="@+id/btnStart"
21      android:layout_width="wrap_content"
22      android:layout_height="wrap_content"
23      android:text="Start"
24      android:layout_gravity="center"
25      android:layout_margin="25dp"
26      android:background="#EE4540"/>
27
28    <androidx.appcompat.widget.AppCompatButton
29      android:id="@+id/btnPause"
30      android:layout_width="wrap_content"
31      android:layout_height="wrap_content"
32      android:text="Pause"
33      android:layout_gravity="center"
34      android:background="#EE4540"/>
35
36    <androidx.appcompat.widget.AppCompatButton
37      android:id="@+id/btnReset"
38      android:layout_width="wrap_content"
39      android:layout_height="wrap_content"
40      android:text="Reset"
41      android:layout_gravity="center"
42      android:layout_margin="25dp"
43      android:background="#EE4540"/>
44
```

MainActivity:

```
12 class MainActivity : AppCompatActivity() {
13     lateinit var chronometer: Chronometer
14     var running: Boolean = false
15     var offset: Long = 0
16
17     val OFFSET_KEY = "offset"
18     val RUNNING_KEY = "running"
19     val BASE_KEY = "base_key"
20
21     chronometer = findViewById(R.id.textTime)
22     val btnStart: Button = findViewById(R.id.btnStart)
23     val btnPause: Button = findViewById(R.id.btnPause)
24     val btnReset: Button = findViewById(R.id.btnReset)
25
26     btnStart.setOnClickListener {
27         if (!running) {
28             setBaseTime()
29             chronometer.start()
30             running = true
31         }
32     }
33
34     btnPause.setOnClickListener {
35         if (running) {
36             saveOffset()
37             chronometer.stop()
38             running = false
39         }
40     }
41
42     btnReset.setOnClickListener {
43         offset = 0
44         setBaseTime()
45         running = false
46     }
47
48     if (savedInstanceState != null) {
49         offset = savedInstanceState.getLong(OFFSET_KEY)
50         running = savedInstanceState.getBoolean(RUNNING_KEY)
51
52         if (running) {
53             chronometer.base = savedInstanceState.getLong(BASE_KEY)
54             chronometer.start()
55         } else setBaseTime()
56     }
57 }
58
59
60
61
62
63
64
65
66
67
```

```
69  override fun onSaveInstanceState(savedInstanceState: Bundle) {
70      super.onSaveInstanceState(savedInstanceState)
71
72      savedInstanceState.putLong("offset",offset)
73      savedInstanceState.putBoolean("running",running)
74      savedInstanceState.putLong("base_key",chronometer.base)
75  }
76
77  private fun setBaseTime() {
78      chronometer.base = SystemClock.elapsedRealtime() - offset
79  }
80
81  private fun saveOffset() {
82      offset = SystemClock.elapsedRealtime() - chronometer.base
83  }
84  }
```

result:

