# GPS\_Rollover\_Fix — פרויקט מקור לבנייה

**דגם מטרה:** AC8227L (ה-Build ששלחת: android-trunk-m0.AC8227L-V1.0, Android 9.1) **מעמד:** המכשיר מושתש (root) — נבנה גרסה שמנסה לשנות את זמן המערכת אוטומטית.

המסמך הזה מכיל את כל קבצי המקור הדרושים כדי לבנות APK ב‑Android Studio. העתק/הורד את התוכן לפרויקט חדש ותבצע Build → Generate APK.

## מבנה הפרויקט (קבצים עיקריים)

GPS\_Rollover\_Fix/  
├─ README.md # הוראות נוספות (תוכן זה)  
├─ build.gradle # project-level  
└─ app/  
 ├─ build.gradle # app-level  
 ├─ src/main/  
 │ ├─ AndroidManifest.xml  
 │ ├─ java/com/example/gpsfix/  
 │ │ ├─ MainActivity.kt  
 │ │ ├─ BootReceiver.kt  
 │ │ └─ GpsFixService.kt  
 │ └─ res/... # מינימלי (אייקון, ערכות נדרשות)

## README (הוראות מהירות)

**מה הקוד עושה** - מאזין ל‑GPS (LocationManager) ולקבלת מיקום/זמן (Location.getTime()). - מיישם תיקון ל־GPS Week Rollover על ידי הוספת בלוקים של 1024 שבועות במידת הצורך. - אם המכשיר מושתש (root): מריץ פקודת su -c "date -s YYYYMMDD.HHMMSS && hwclock -w" לשינוי זמן המערכת. - רץ כ‑Foreground Service ומופעל גם ב־BOOT כדי לפעול בהדלקת המערכת.

**דרישות לפיתוח** - Android Studio (מומלץ 3.5+) - JDK מתאים (8+) ואמצעי בנייה gradle - targetSdkVersion / compileSdkVersion: 28–29

**הרצת בדיקה** 1. פתח את הפרויקט ב‑Android Studio. 2. वBuild → Make Project 3. Generate Signed APK או Build APK (debug) לצורך ניסוי. 4. התקן על המכשיר עם adb install -r app-debug.apk. 5. תן הרשאת מיקום ידנית בהגדרות אם לא התבקש. 6. הרץ את האפליקציה ולחץ Start. 7. בדוק לוג ב‑adb logcat (חפש GpsFixService) כדי לראות את שורות ה־GPS raw ו־corrected.

**להפעיל שינויים במערכת (דרוש root או התקנה כ‑system)** - אם המכשיר מושתש: הפונקציה setSystemTimeRoot(...) תריץ פקודת su לשינוי זמן. - להתקנה כ‑system app (כדי לקבל הרשאת SET\_TIME רשמית): יש להעביר את ה‑APK ל־/system/priv-app/ או להשתמש במודול Magisk.

**פקודות לדוגמה (ADB/root)**

# אם יש adb root / remount  
aadb root  
adb remount  
adb push app-release.apk /system/priv-app/GpsFix/GpsFix.apk  
adb shell chmod 644 /system/priv-app/GpsFix/GpsFix.apk  
adb shell chown root:root /system/priv-app/GpsFix/GpsFix.apk  
adb reboot

**אזהרות** - שינוי זמן מערכת יכול לפגוע בתעודות, אפליקציות, לוגים ושירותים. יש לבצע בגיבוי ועם מודעות לסיכונים. - התקנה כ‑system או שימוש ב‑su עלולה לבטל אחריות. - אל תתקין קבצי firmware או ROM שלא מותאמים לדגם.

## קבצי מקור (להעתקה ישירה)

**שימו לב:** להלן התוכן של הקבצים העיקריים. העתיקו את כל הבלוקים לקבצים התואמים בפרויקט Android Studio.

### build.gradle (project-level)

buildscript {  
 repositories { google(); mavenCentral() }  
 dependencies { classpath "com.android.tools.build:gradle:3.5.4" }  
}  
allprojects { repositories { google(); mavenCentral() } }

### app/build.gradle

apply plugin: 'com.android.application'  
  
android {  
 compileSdkVersion 29  
 defaultConfig {  
 applicationId "com.example.gpsfix"  
 minSdkVersion 24  
 targetSdkVersion 28  
 versionCode 1  
 versionName "1.0"  
 }  
 buildTypes { release { minifyEnabled false } }  
}  
  
dependencies {  
 implementation "org.jetbrains.kotlin:kotlin-stdlib:1.3.72"  
 implementation 'androidx.core:core-ktx:1.0.2'  
 implementation 'androidx.appcompat:appcompat:1.0.2'  
}

### AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest package="com.example.gpsfix"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
 <uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED" />  
 <!-- לשימוש כ-system app בלבד -->  
 <uses-permission android:name="android.permission.SET\_TIME" tools:ignore="ProtectedPermissions" />  
  
 <application  
 android:allowBackup="false"  
 android:label="GPS Rollover Fix"  
 android:icon="@mipmap/ic\_launcher"  
 android:theme="@style/Theme.AppCompat.Light.NoActionBar" >  
  
 <activity android:name=".MainActivity" android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN"/>  
 <category android:name="android.intent.category.LAUNCHER"/>  
 </intent-filter>  
 </activity>  
  
 <receiver android:name=".BootReceiver" android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.BOOT\_COMPLETED"/>  
 </intent-filter>  
 </receiver>  
  
 <service android:name=".GpsFixService" android:exported="false" />  
 </application>  
</manifest>

### MainActivity.kt

package com.example.gpsfix  
  
import android.Manifest  
import android.app.Activity  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.os.Bundle  
import android.widget.Button  
import androidx.core.app.ActivityCompat  
import androidx.core.content.ContextCompat  
  
class MainActivity: Activity() {  
 private val REQ\_PERM = 123  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 val btn = Button(this).apply { text = "Start GPS Fix Service" }  
 setContentView(btn)  
  
 btn.setOnClickListener {  
 if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION)  
 != PackageManager.PERMISSION\_GRANTED) {  
 ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.ACCESS\_FINE\_LOCATION), REQ\_PERM)  
 } else {  
 startService(Intent(this, GpsFixService::class.java))  
 }  
 }  
 }  
  
 override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>, grantResults: IntArray) {  
 if (requestCode == REQ\_PERM && grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  
 startService(Intent(this, GpsFixService::class.java))  
 }  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
 }  
}

### BootReceiver.kt

package com.example.gpsfix  
  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
import android.util.Log  
  
class BootReceiver: BroadcastReceiver() {  
 override fun onReceive(context: Context, intent: Intent) {  
 if (intent.action == Intent.ACTION\_BOOT\_COMPLETED) {  
 val i = Intent(context, GpsFixService::class.java)  
 context.startForegroundService(i)  
 Log.i("GpsFix", "BootReceiver started service")  
 }  
 }  
}

### GpsFixService.kt

package com.example.gpsfix  
  
import android.app.\*  
import android.content.Context  
import android.content.Intent  
import android.location.Location  
import android.location.LocationListener  
import android.location.LocationManager  
import android.os.Build  
import android.os.Bundle  
import android.os.IBinder  
import android.util.Log  
import java.text.SimpleDateFormat  
import java.util.\*  
  
class GpsFixService : Service() {  
  
 private lateinit var lm: LocationManager  
 private val TAG = "GpsFixService"  
  
 override fun onCreate() {  
 super.onCreate()  
 lm = getSystemService(Context.LOCATION\_SERVICE) as LocationManager  
 startForegroundCompat()  
 startListening()  
 }  
  
 private fun startForegroundCompat() {  
 val channelId = "gpsfix\_channel"  
 if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {  
 val nm = getSystemService(Context.NOTIFICATION\_SERVICE) as NotificationManager  
 val ch = NotificationChannel(channelId, "GPS Fix", NotificationManager.IMPORTANCE\_LOW)  
 nm.createNotificationChannel(ch)  
 }  
 val notif = Notification.Builder(this, channelId)  
 .setContentTitle("GPS Rollover Fix")  
 .setContentText("מנסה לתקן תאריך GPS")  
 .setSmallIcon(android.R.drawable.ic\_menu\_mylocation)  
 .build()  
 startForeground(1, notif)  
 }  
  
 private fun startListening() {  
 try {  
 lm.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 2000L, 0f, object: LocationListener {  
 override fun onLocationChanged(loc: Location) {  
 handleLocation(loc)  
 }  
 override fun onStatusChanged(provider: String?, status: Int, extras: Bundle?) {}  
 override fun onProviderEnabled(provider: String) {}  
 override fun onProviderDisabled(provider: String) {}  
 })  
 } catch (e: SecurityException) {  
 Log.e(TAG, "Missing permissions: ${e.message}")  
 }  
 }  
  
 private fun handleLocation(loc: Location) {  
 val gpsMillis = loc.time  
 val corrected = correctGpsWeekRollover(gpsMillis)  
 val sdf = SimpleDateFormat("yyyy-MM-dd HH:mm:ss", Locale.getDefault())  
 sdf.timeZone = TimeZone.getDefault()  
 Log.i(TAG, "GPS raw: ${sdf.format(Date(gpsMillis))} -> corrected: ${sdf.format(Date(corrected))}")  
  
 if (isDeviceRooted()) {  
 setSystemTimeRoot(corrected)  
 } else {  
 Log.i(TAG, "Device not rooted - cannot set system time automatically")  
 }  
 }  
  
 private fun correctGpsWeekRollover(gpsMillis: Long): Long {  
 val cal = Calendar.getInstance(TimeZone.getTimeZone("UTC"))  
 cal.timeInMillis = gpsMillis  
 val year = cal.get(Calendar.YEAR)  
 if (year >= 2015) return gpsMillis  
  
 val WEEK\_MS = 7L \* 24 \* 60 \* 60 \* 1000  
 val BLOCK\_MS = 1024L \* WEEK\_MS  
 var corrected = gpsMillis  
 for (i in 0 until 5) {  
 corrected += BLOCK\_MS  
 val c = Calendar.getInstance(TimeZone.getTimeZone("UTC"))  
 c.timeInMillis = corrected  
 if (c.get(Calendar.YEAR) >= 2015) return corrected  
 }  
 return gpsMillis  
 }  
  
 private fun isDeviceRooted(): Boolean {  
 val paths = arrayOf("/sbin/su","/system/bin/su","/system/xbin/su","/data/local/xbin/su","/data/local/bin/su","/system/sd/xbin/su")  
 for (p in paths) if (java.io.File(p).exists()) return true  
 return false  
 }  
  
 private fun setSystemTimeRoot(timeMillis: Long) {  
 val sdf = SimpleDateFormat("yyyyMMdd.HHmmss", Locale.getDefault())  
 sdf.timeZone = TimeZone.getDefault()  
 val timestr = sdf.format(Date(timeMillis))  
 val cmd = arrayOf("su", "-c", "date -s $timestr && hwclock -w")  
 try {  
 val p = Runtime.getRuntime().exec(cmd)  
 val exit = p.waitFor()  
 Log.i(TAG, "setSystemTimeRoot exit=$exit, time=$timestr")  
 } catch (e: Exception) {  
 Log.e(TAG, "Failed to set time: ${e.message}")  
 }  
 }  
  
 override fun onBind(intent: Intent?): IBinder? = null  
 override fun onDestroy() { super.onDestroy() }  
}

## התאמות המיוחדות ל־AC8227L

* הפרויקט בנוי באופן כללי; אם אחרי בדיקות ראשוניות עדיין מתגלים פורמטים NMEA לא תקינים בנגזרת Location.getTime(), אפשר להוסיף ניתוח NMEA ישירות (לשמוע RMC/GPZDA) ולהשתמש בשדה התאריך משם.
* אם תרצה — אוכל להוסיף קוד פרסינג ל‑NMEA בשלב הבא.

## מה אני עושה עכשיו בשבילך

* הכנתי את חבילת המקור המלאה (התוכן שלמעלה). אתה יכול:
  1. להעתיק את הקבצים לפרויקט ב‑Android Studio ולבנות APK בעצמך, או
  2. להעביר את החבילה לטכנאי/מפתח שיבנה / יתקין עבורך כ‑system app.

**אם תרצה** אכין גרסה עם פרסינג NMEA נוסף, או הסבר מפורט לשלבי Magisk module להתקנה אוטומטית כ‑system.

**שאלות נוספות?** כתוב לי — ואני אתאים עבורך פירוט/סקריפטים נוספים.

תודה ובהצלחה בבדיקה!