

מחקר עם ג'מיני

השאלה:

יש לי מכשיר אנדרואיד עם מקלדת מקשים מובנית, אני מעוניין לשנות את הפונקציה של המקש השמאלי העליון, כרגע הוא מפנה למסך הבית אבל זה מיותר, כי מקש ההפעלה גם הוא מפנה למסך הבית. אני רוצה שהוא יפתח את תפריט האפשרויות בכל אפליקציה. איך עושים את זה

התשובה:

- (1) איתור דגמי אנדרואיד נפוצים עם מקלדות פיזיות
 - (a) מצא מידע על גרסאות מערכת ההפעלה של מכשירים אלו, מכיוון שאפשרויות מיפוי המקשים תלויות בהן.
 - (2) בדיקת הגדרות מערכת האנדרואיד
 - (a) חפש הגדרות נגישות או מקשים מובנות (בדרך כלל תחת "נגישות" או "מקשים") המאפשרות שינוי פונקציית מקשים פיזיים על ידי היצרן.
 - (3) זיהוי הקוד של המקש השמאלי העליון
 - (a) מצא כלים או שיטות (כגון אפליקציות בדיקת Keycode) שיכולים לזהות את הקוד של המקש הספציפי שאותו המשתמש מעוניין לשנות.
 - (4) מציאת אפליקציות צד שלישי למיפוי מחדש
 - (a) חפש אפליקציות פופולריות למיפוי מחדש של מקשים (כגון Button Mapper או Key Mapper) ובדוק את יכולתן לשנות מקשי חומרה במכשיר.
 - (5) בדיקת יכולת המיפוי לפונקציית "תפריט אפשרויות"
 - (a) ודא שהאפליקציות או השיטות שנמצאו תומכות בהקצאה מחדש של המקש לפונקציה של פתיחת "תפריט אפשרויות" (המתאים ל-KEYCODE_MENU ההיסטורי של אנדרואיד).
 - (6) חקירת פתרונות מתקדמים הדורשים גישת Root (כגון Magisk Modules)
 - (a) התמקד בבדיקת מודול Magisk למיפוי מחדש של מקשים שנמצא בקישור: [https://github.com/JefforDerp/Magisk-KeyboardRemaps`](https://github.com/JefforDerp/Magisk-KeyboardRemaps).
 - (b) חקור את הדרישות, תהליך ההתקנה וההתאמה של המודול הספציפי הזה להשגת פונקציית 'תפריט אפשרויות' (MENU) עבור המקש הפיזי.
 - (c) בנוסף, חקור את שיטת עריכת קובצי Key Layout (בנתיב /system/usr/keylayout) כאפשרות חלופית או משלימה, ובחן את הסיכונים הכרוכים בשיטות אלו.

(7) חיפוש בפורומים וקהילות טכנולוגיות

(a) מצא דיונים או מדריכים ספציפיים בפורומים (כגון XDA Developers) הנוגעים למיפוי מחדש של מקשים פיזיים במכשיר ספציפי בעל מקלדת מובנית.

תוצאות המחקר:

ניתוח מקיף ומיפוי מחדש של מקשי מערכת קשיחים במקלדות אנדרואיד פיזיות: הקצאת פונקציית תפריט (KEYCODE_MENU)

פרק 1: הבנת המקש הפיזי המוגן והצורך ב-Menu Key

1.1. הרקע לאתגר: כפילות פונקציית הבית במכשירים נישתיים

השאלתה מתייחסת לבעיה ארכיטקטונית נפוצה במכשירי אנדרואיד המשלבים מקלדת פיזית קשיחה, (QWERTY) כדוגמת סדרת) UniHertz Titan כמו Titan Pocket ששורר ב-2021¹ (או מכשירי BlackBerry הנישתיים². במכשירים אלו, מקשים פיזיים מסוימים ממופים לפונקציות מערכת קלאסיות.

הבעיה המדוברת נוגעת למקש השמאלי העליון, שכרגע מוגדר כ-Home וממופה, ככל הנראה, ל-KEYCODE_HOME (קוד 3). זהו מקש מוגן במערכת האנדרואיד. המשתמש מזהה כפילות מיותרת בפונקציונליות זו, שכן מקש ההפעלה (Power Key) כבר מבצע פעולת חזרה למסך הבית לאחר לחיצה או אתחול, ולכן הקצאת מקש ייעודי נוסף לפעולה זהה היא בזבזנית. נציין כי מכשירי UniHertz, כדוגמה, מציגים מספר מקשים הממופים לפונקציות מערכת קלאסיות כמו Home ו-Back Key בפריסות המקשים שלהם⁴.

1.2. ההיסטוריה והתפקיד של `KEYCODE_MENU` באנדרואיד המודרני

הפונקציה המבוקשת היא פתיחת תפריט האפשרויות, המקבילה לקוד המפתח הקלאסי `KEYCODE_MENU` (קוד 82). קוד זה נועד במקור להפעלת תפריט האפשרויות (Overflow Menu) של היישום הפעיל.

הבנה מעמיקה של הפונקציה מחייבת סקירה היסטורית: אנדרואיד הפסיקה לתמוך במקש Menu פיזי או וירטואלי כברירת מחדל החל מגרסת אנדרואיד 4.0 (Ice Cream Sandwich), תוך עידוד מפתחים לעבור לכפתור התפריט הווירטואלי (המופיע לרוב כ-3 נקודות) בתוך ה-Action Bar של האפליקציה. עם זאת, הקוד `KEYCODE_MENU` עדיין קיים ב-API של אנדרואיד⁵ ונותר רלוונטי עבור מספר תרחישים קריטיים: ראשית, יישומים עסקיים או תוכנות Legacy שפותחו עבור גרסאות אנדרואיד ישנות יותר או פלטפורמות דומות (כגון BlackBerry OS) עדיין עשויים להסתמך על אירוע קלט זה⁶. שנית, עבור משתמשים בעלי מקלדת פיזית, היכולת לפתוח תפריט בצורה מהירה ופיזית מספקת יתרון משמעותי בחווית המשתמש.

1.3. סקירת ארכיטקטורת הקלט של אנדרואיד (Input Handling Stack)

מיפוי מחדש של מקש מערכת מוגן דורש הבנה היכן בשרשרת הטיפול בקלט (Input Pipeline) מתרחש השינוי. ארכיטקטורת הטיפול בקלט באנדרואיד מחולקת למספר שלבים קריטיים:

1. **שלב החומרה (Hardware):** המקש הפיזי נלחץ ושולח Scan Code (קוד גולמי של הלחצן) למערכת ההפעלה.
2. **שלב ליבת המערכת (Kernel/HAL):** קובצי ה (.kl) Key Layout - הממוקמים במחיצת /system/usr/keylayout/ ממירים את ה Scan Code ל-Android Key Code (למשל, הופכים X Scan Code ל KEYCODE_HOME). זה הוא המקום האידיאלי והיציב ביותר לבצע שינוי מערכתי קבוע⁷.
3. **שלב שרת המערכת (System Server):** אירוע ה Key Code מטופל על ידי המערכת. בנקודה זו, מגבלות אבטחה מופעלות, והמערכת חוסמת אפליקציות צד שלישי (כמו שירותי נגישות רגילים) מליירט אירועי מקשים מוגנים כגון KEYCODE_HOME או KEYCODE_POWER.
4. **שלב האפליקציה (Application):** האירוע עובר לבסוף ליישום הפעיל. אפליקציות מיפוי (מחדש) כגון Button Mapper חייבות להתערב בשלב 4, לרוב באמצעות שירותי נגישות, אך חסימת האבטחה בשלב 3 מנמיכה את הצורך בהרשאות גבוהות יותר.

פרק 2: זיהוי ואבחון: מיפוי קוד המקש הפיזי (Keycode Identification and Structure)

כדי לבצע מיפוי מחדש יציב ברמת המערכת, חובה לדעת את ה Scan Code - הפיזי המדויק שהחומרה שולחת. ניתוח זה מחייב שימוש בכלים אבחון ספציפיים.

- 1.2 כלים לאבחון קודי מקשים (Keycode Decoders)
יש לוודא תחילה שקוד המפתח המדובר אינו חלק מצירוף מקשים (כגון Alt + המקש הנדון) שכבר ממופה לפונקציית Home בתוך קובץ ה (.kcm) Key Character Map של המקלדת⁴.

כדי לבודד את ה Key Code - הפיזי, יש להשתמש בכלי אבחון ייעודיים:

- **KeyEvent Display:** אפליקציה מומלצת המציגה, KeyEvents (KeyUp, KeyDown, KeyLongPress) כלי זה יכול להציג מידע מ LogCat - ואף מ-⁸ Kernel Log אם המכשיר בעל (Root) כדי לראות את הקוד הגולמי (Scan Code) הנשלח מהחומרה. בעת הפעלת האפליקציה, כל המקשים ה"קשיחים" למעט Home ומקש ההפעלה, יישלחו כ Keycodes - גולמיים⁸.
- **KeyDecoder:** כלי נוסף, זמין וחינמי, המשמש לפענוח קודי מקשים⁹.

השימוש בכלים אלה מאפשר לזהות את ה Scan Code - הייחודי שמזוהה על ידי הליבה כאשר המשתמש לוחץ על המקש השמאלי העליון.

- 2.2 פירוט קובצי Key Layout (.kl) ו Key Character Map (.kcm)
המיפוי המערכתי של מקשי חומרה מוגדר בקובצי Key Layout (.kl) המאחסנים את ההקצאה בין ה Scan Code - לבין ה Android Key Code - כגון KEYCODE_HOME). המקום של קבצים אלו הוא /system/usr/keylayout/.

האתגר המרכזי הוא איתור הקובץ הנכון לשינוי:

1. **מקלדות גנריות (HID):** אם המקלדת מחוברת באמצעות USB או בלוטות' ומוגדרת כהתקן ממשק אנושי גנרי (Generic HID), הקובץ הרלוונטי עשוי להיות Vendor_ffff_Product_ffff.kl. מודול Magisk KeyboardRemaps משתמש בנתיב זה כדי לשנות מקשים גנריים⁷.

2. **מקלדות מובנות (OEM/Proprietary):** במכשיר נישתי כמו, Unihertz Titan, היצרן משתמש ככל הנראה ב Vendor ID (VID) ו-Product ID (PID) ספציפיים משלו כדי להגדיר פריסת מקשים מותאמת. הצלחת המיפוי תלויה באיתור קובץ ה-kl. הספציפי שנוצר על ידי היצרן. ניתן לאתר קובץ זה על ידי ניתוח יומני מערכת (Logcat) בעת לחיצה על המקש באמצעות כלי האבחון⁸.

מטרת העריכה היא לאתר בתוך קובץ ה kl-את השורה הממפה את ה Scan Code-המזוהה ולשנות את ההקצאה מ HOME-(קוד 3) ל MENU-קוד 82).

הערות	פונקציה	ערך	שם Keycode
הקצאת המקור, מוגנת מפני יירוט אפליקטיבי.	חזרה למסך הבית	3	<code>\$\text{KEYCODE_HOME}\$</code>
ההקצאה המבוקשת (היעד).	פתיחת תפריט אפשרויות	82	<code>\$\text{KEYCODE_MENU}\$</code>

פרק 3: נתיב 1: פתרונות ברמת היישום (ללא Root) וגבולותיהם

הנתיב הקל ביותר ליישום הוא שימוש באפליקציות צד שלישי המשתמשות בשירותי נגישות (Accessibility Services) כדי ליירט ולמפות אירועי מקשים.

3.1. שימוש ביישומי מיפוי מקשים

Button Mapper (שפותחה על ידי flar2) היא האפליקציה הנפוצה והמומלצת ביותר למטרות אלו. האפליקציה פועלת על ידי ריצה כשירות נגישות ברקע, המיירט אירועי מקשים לפני שהם מגיעים ליישום היעד, ומחליף את הפעולה המוגדרת. Button Mapper תומכת במיפוי מחדש של לחצני חומרה שונים, כולל מקשי ווליום ואפילו מקשי קפסיטיביים מסוימים. האפליקציה תוכננה לתמוך במקלדות פיזיות, כפי שדווח על ידי משתמשי BlackBerry KEYone ו-KEY2.

האפליקציה מאפשרת להקצות פעולות שונות, כולל השקת יישומים, הפעלת פקודות Root, או הדמיית קודי מקשים (Simulate keycodes).

3.2. אתגר המקש המוגן (KEYCODE_HOME)

היישום הרגיל של Button Mapper ללא הרשאות מוגבהות (Root או ADB) אינו יכול לספק פתרון יציב או מלא לשינוי המקש המדובר, מכיוון שהוא ממופה ל-`KEYCODE\_HOME`.

אנדרואיד אוכפת מדיניות אבטחה מחמירה המונעת משירותי נגישות רגילים ליירט את אירועי המקשים הקריטיים למערכת, כגון `KEYCODE\_HOME` ו-`KEYCODE\_POWER`. חסימה זו נועדה להבטיח שיציאת חירום למסך הבית תמיד תישאר זמינה, גם אם יישום כלשהו או שירות צד שלישי קורס או תופס את השליטה במערכת.

כדי להתגבר על חסימה זו ולמפות את `KEYCODE\_HOME` או לבצע פעולות מורכבות יותר כמו "Menu button" או "Simulate keycodes", נדרשת הרשאת Root, או לפחות

הרשאת `WRITE\ _SECURE\ _SETTINGS` באמצעות ADB. במילים אחרות, מיפוי מחדש של מקש מערכת קשיח באמצעות אפליקציה בסיסית ללא Root ייכשל באופן עקבי, שכן ה-System Server מתייחס ללחיצה זו כאירוע פנימי שאין להעבירו לשירותי צד שלישי.

3.3. ניצול מקשים הניתנים לתכנות כפתרון עוקף

אם המכשיר המדובר מאפשר הקצאת מקשים דינמית על ידי היצרן, ניתן להקצות את הפונקציה המבוקשת למקש אחר שאינו מוגן. לדוגמה, במכשירי Unihertz Titan קיימים מקשים כמו "Magic Key" או "Subscreen Control". מקשים אלו מוגדרים לעיתים כניתנים לתכנות דרך ממשק המערכת עצמו, וניתן להקצות להם קיצור דרך המפעיל יישום או פונקציה המדמה את `KEYCODE\_MENU`. פתרון זה מציע חווית משתמש משופרת מבלי לפרוץ את המערכת, אך אינו עונה ישירות על דרישת המשתמש לשנות דווקא את המקש השמאלי העליון.

פרק 4: נתיב 2: מיפוי מתקדם באמצעות ADB (Android Debug Bridge)

נתיב הביניים משלב את הנוחות של אפליקציית צד שלישי עם הרשאות מוגבהות הניתנות באמצעות ADB. גישה זו נמנעת מהצורך ב-Root מלא, מה שהופך אותה לאטרקטיבית עבור משתמשים המעוניינים לשמור על אחריות המכשיר ככל האפשר.

4.1. הרשאות מתקדמות עבור אפליקציות צד שלישי באמצעות ADB

השלב הקריטי בנתיב זה הוא הענקת ההרשאה `WRITE\ _SECURE\ _SETTINGS` לאפליקציית המיפוי (כדוגמת Button Mapper Pro). הרשאה זו מאפשרת לאפליקציה לעקוף חלק ניכר ממגבלות האבטחה של אנדרואיד בנוגע לשינוי הגדרות מערכת ובקרת קלט.

התהליך דורש את השלבים הבאים:

1. הפעלת מצב Debugging USB במכשיר.
 2. חיבור המכשיר למחשב והפעלת ADB.
 3. הפעלת פקודת מענק הרשאה ספציפית דרך שורת הפקודה.
- לאחר מתן ההרשאה, Button Mapper מקבל את היכולת ליירט את `KEYCODE\_HOME` ולהמיר את פעולתו.

4.2. יישום מעשי (באמצעות Button Mapper Pro)

לאחר קבלת ההרשאות המתאימות, המשתמש יכול להיכנס להגדרות Button Mapper:

1. הגדר את המקש השמאלי העליון ככפתור למיפוי.
2. בחר באפשרות "Simulate keycodes" (שזמינה עם הרשאת ADB או Root).
3. הקצה לפעולת הלחיצה היחידה את הערך 82, המייצג את `KEYCODE\_MENU`.

Button Mapper מאפשר גם להגדיר פעולות נוספות לאותו מקש, כגון לחיצה כפולה או לחיצה ארוכה (Long Press), מה שמאפשר לשמר את פונקציית הבית המקורית (Home) כלחיצה ארוכה, אם המשתמש מעוניין בכך.

4.3. מגבלות ויציבות של פתרון ADB

שימוש ב-ADB להענקת הרשאות מהווה פתרון שימושי, אך הוא אינו יציב כמו שינוי ברמת הליבה. הבעיה העיקרית נובעת מניהול צריכת החשמל והאופטימיזציה של המערכת.

מכשירים נישתיים רבים, במיוחד אלה המשתמשים בערכות שבבים של Mediatek (כגון חלק ממכשירי Unihertz), משתמשים במנגנוני חיסכון בסוללה מותאמים אישית כמו DuraSpeed או App Blocker. מנגנונים אלו נועדו לסגור יישומי רקע בצורה אגרסיבית כדי לשפר את חיי הסוללה. אם אפליקציית Button Mapper (או שירות הנגישות שלה) נסגרת על ידי DuraSpeed, המיפוי המוגדר באמצעות ADB יפסיק לפעול באופן מיידי. המשתמש יאלץ לוודא שהאפליקציה "allowed to run in background" בהגדרות DuraSpeed ו- Intelligent assistance → App blocker, כדי להבטיח עקביות.

בנוסף, Button Mapper אינו פועל באופן כללי כשהמסך כבוי, אלא אם המכשיר מושרש (Rooted) או אם בוצעה פקודת ADB ספציפית עבור לחצני הווליום, אך לא בהכרח עבור מקשי מקלדת ייעודיים. מגבלה זו פוגעת ביכולת להשיג חוויה חלקה ועקבית.

פרק 5: נתיב 3: מיפוי ברמת מערכת ההפעלה (דרישת Root)

המיפוי המדויק, האמין והקבוע ביותר של מקש מערכת מוגן דורש גישה ישירה לקובצי הליבה של אנדרואיד באמצעות הרשאת Root. זהו הפתרון המומלץ למשתמשים מתקדמים המחפשים יציבות מרבית.

5.1. הצדקה להרשאת Root: יתרונות היציבות והשליטה המוחלטת

Root מאפשר שינוי קבוע של קובצי המערכת הקוראים את הקלט הפיזי (כלומר, קובצי ה-kl), ובכך משנה את ה-`Key Code` הפיזי המועבר לשכבות העליונות של אנדרואיד באופן מהותי. כשהמקש מוגדר כ-`KEYCODE\_MENU` כבר ברמת הליבה, ה-`System Server` מקבל אירוע Menu ולא Home, ואין צורך לעקוף חסימות אבטחה באמצעות שירותי נגישות.

פתרון זה מבטיח יציבות מוחלטת: הוא חסין בפני אופטימיזציות סוללה אגרסיביות (כמו אלו המוטמעות על ידי יצרנים מסוימים) ופועל לפני שכל חסימה אפליקטיבית או מערכתית מופעלת.

5.2. שימוש במנהל המערכת Magisk ומודולים Systemless

השיטה המודרנית והמועדפת לביצוע שינויי מערכת היא באמצעות "Root" Systemless דרך Magisk. גישה זו מאפשרת לבצע שינוי קבוע של קובץ פריסת המקשים הווירטואלי מבלי לשנות פיזית את מחיצת ה-`system`. זה שומר על יכולת קבלת עדכוני OTA (במקרים מסוימים) ומקל על תחזוקה.

מודול Magisk ייעודי, כמו `KeyboardRemaps`, מדגים מנגנון זה: הוא משתמש ב-Magisk כדי "להזריק" קבצי `Key Layout` מתוקנים לתוך סביבת המערכת הווירטואלית, בנתיב המקביל ל-`/system/usr/keychars/`. מנגנון זה נבדק ואושר לפעולה יציבה בגרסאות אנדרואיד 11, 12 ו-13.

מודול Magisk המבצע מיפוי מקשים ברמת המערכת מעניק למשתמש את האיזון הקריטי בין שליטה מלאה (Root) לבין קלות תחזוקה.

5.3. מדריך מפורט לשינוי קובץ Key Layout (.kl)

יישום הפתרון דורש ביצוע קפדני של השלבים הטכניים:

5.3.1. איתור קובץ ה-KL הרלוונטי

כפי שפורט בפרק 2, יש להשתמש בכלי אבחון כדי לזהות את ה-Scan Code ואת ה-VID/PID של המקלדת הפיזית. זה יאפשר לאתר את קובץ ה-.kl הספציפי שאותו יש לערוך בנתיב המערכת (/system/usr/keylayout/).

5.3.2. עריכת קובץ ה-KL

לאחר איתור הקובץ, יש לערוך אותו (לרוב באמצעות מנהל קבצים עם הרשאות Root):

1. אתר את שורת הקוד הממפה את ה-Scan Code הפיזי של המקש השמאלי העליון. בהנחה שה-Scan Code הוא 158 (מספר היפותטי):

```
key 158 HOME .2
```

3. שנה את ההקצאה המערכתית:

```
# Key 158 remapped to Menu .4  
key 158 MENU .5
```

שינוי זה מבטיח שכאשר המשתמש לוחץ על המקש, המערכת מזהה אותו אוטומטית כ-KEYCODE_MENU (82), עוד לפני שכל שירות אפליקטיבי נכנס לתמונה.

5.3.3. הטמעה באמצעות Magisk

אם השינוי מבוצע ישירות על קובץ המערכת (שאינו מומלץ), הוא עלול להימחק בעדכוני OTA. הדרך המועדפת היא להשתמש במודול Magisk:

1. צור מודול Magisk חדש (או השתמש במודול קיים כמו KeyboardRemaps).
2. הצב את קובץ ה-.kl המתוקן בתוך מבנה התיקיות של המודול כך שישקף את הנתיב /system/usr/keylayout/.
3. הפעל את המודול ודאג לאתחול מחדש של המכשיר.

הטמעת המיפוי ברמת הליבה מבטיחה שהשינוי יחול באופן עקבי בכל אתחול מחדש ובכל גרסאות האפליקציות, ומספקת את הביצועים המהירים והאמינים ביותר. יש לציין שנדרשת זהירות רבה בעת שינוי קבצים ברמה זו; טעות בקובץ .kl עלולה למנוע אתחול מלא של המערכת (Boot Loop), אם כי Magisk מאפשר הסרה קלה של המודול במצב שחזור.

פרק 6: סיכום והמלצות יישום מעשיות

הניתוח הטכני מצביע על שלושה נתיבים אפשריים להשגת מטרת המשתמש, עם רמות שונות של יציבות ואמינות. בהתחשב בכך שהמקש המקורי ממופה ל-KEYCODE_HOME, מדובר בשינוי של מקש מערכת מוגן, הדורש הרשאות מוגבהות.

ההמלצה האנליטית החד-משמעית היא לבחור ב**נתיב 3: שינוי קובץ Key Layout באמצעות Magisk Systemless**. נתיב זה מבטיח שהפונקציה תהיה עקבית, תעבוד גם כאשר שירותי צד שלישי מושעים (למשל על ידי DuraSpeed), ותימנע מחסימות אבטחה של אנדרואיד.

להלן סיכום השוואתי של הגישות השונות:

Table 3: השוואת פתרונות למיפוי מקש הבית (KEYCODE_HOME) לתפריט (KEYCODE_MENU)

שיטת מיפוי	יתרונות עיקריים	חסרונות עיקריים	דרישות קדם	המלצת מומחה
Button Mapper (לא Root)	קל ליישום, בטוח.	אינו יכול ליירט, KEYCODE_HOME חסר תועלת למקרה זה.	אפליקציה מותקנת.	לא רלוונטי.
Button Mapper (ADB)	נמנע מ-Root מלא.	יציבות נמוכה, מושפע מאופטימיזציית סוללה, דורש ADB לאחר אתחול.	ADB מופעל, Button Mapper Pro.	פתרון פשרה למשתמשים שאינם רוצים Root.
שינוי קובץ Key Layout	יציבות מקסימלית (רמת Kernel), Systemless.	מורכב טכנית, דורש Root וידע בזיהוי Keycodes.	Root (Magisk), Keycode Debugger.	הפתרון היסודי והמומלץ ביותר.

6.1. שלבי פעולה מיידיים

לצורך יישום הפתרון היסודי (נתיב 3), על המשתמש לבצע את הצעדים הבאים:

- הכנת המכשיר:** ודא שהמכשיר בעל הרשאת Root (באמצעות Magisk) ויש גישה לקבצי המערכת.
- זיהוי ה-Scan Code:** השתמש בכלי אבחון כגון KeyEvent Display כדי לזהות באופן מדויק את ה-Scan Code של המקש השמאלי העליון ואת קובץ ה-Key Layout (.kl) הרלוונטי למקלדת הפיזית.
- יישום השינוי:** צור מודול Magisk חדש או שנה מודול קיים (כגון KeyboardRemaps) כדי להחליף את ההקצאה של ה-Scan Code הספציפי מ-HOME ל-MENU בתוך קובץ ה-kl. המותאם.

בדרך זו, כל לחיצה על המקש השמאלי העליון תתורגם לאירוע `KEYCODE\_MENU` (82) ברמת המערכת, שיפתח את תפריט האפשרויות של היישום הפעיל, ויספק פתרון יציב ועקבי לדרישת המשתמש.

פרק 7: נספח: עבודה עם קודי מפתח מתקדמים

7.1. טיפול ב-Long Press ובצירופים

בעת שינוי ההקצאה המערכתית (נתיב 3), המשתמש מקריב את פונקציית הבית המקורית. אם יש צורך לשמר פונקציה זו, ניתן לבחור בפתרון ADB/Button Mapper (נתיב 2) ולהקצות את `KEYCODE\_HOME` ללחיצה ארוכה (Long Press) על המקש הנדון.

לחלופין, פריסת המקלדת הפיזית מאפשרת לעיתים קרובות צירופי מקשים. ניתן להגדיר דרך קובץ ה-`Key Character Map (kcm.)` או דרך הגדרות הנגישות המובנות של אנדרואיד (TalkBack), צירוף מקשים שיבצע פעולת Home (למשל, `Alt + Ctrl + H`) או פעולת Menu (למשל, `Alt + Shift + G`) להצגת תפריט TalkBack). ניתן להשתמש בפונקציות הנגישות הללו כדי להגדיר קיצור דרך חדש לפתיחת תפריט, אך מכיוון שהמשתמש מעוניין במיפוי מחדש של מקש בודד, הפתרון ברמת הליבה הוא העדיף.

7.2. הערות על תחזוקה ובטיחות

שינוי קבצי מערכת דורש זהירות מקסימלית. טעות תחביר בקובץ ה-`kl` עלולה להוביל לכשלים חמורים במערכת. יתרון השימוש במודול `Magisk Systemless` הוא שבמקרה של כשל, ניתן לנטרל את המודול ממצב שחזור, ובכך להימנע מאי יציבות מערכתית קבועה. יש לוודא כי המודול שהוטמע תואם לגרסת האנדרואיד הספציפית המותקנת במכשיר (כאשר דוגמאות קיימות פועלות עד אנדרואיד 13).