MIB2 Controller - Testing Instructions for Google Play Review

Application Name: MIB2 Controller
Package Name: com.feplazas.mib2controller
Version: 1.0.0 (versionCode 7)
Developer: Felipe Plazas (feplazas@gmail.com)
Date: January 2026

Executive Summary

MIB2 Controller is a specialized diagnostic and repair tool for automotive infotainment systems. This document provides comprehensive instructions for Google Play reviewers to understand the application's purpose, hardware requirements, and testing procedures.

Important Note: This application requires specific automotive hardware (MIB2 STD2 Technisat Preh unit) that may not be readily available to reviewers. This document explains the app's functionality, legal compliance, and provides alternative verification methods.

1. Application Purpose and Legal Compliance

1.1 Primary Use Cases

MIB2 Controller is designed for three legitimate user groups:

1. **Authorized Automotive Technicians**: Professionals performing warranty or post-warranty repairs on vehicle infotainment systems
2. **Vehicle Owners**: Individuals exercising their legal right to repair and maintain their own property

3. **Security Researchers**: Professionals conducting good-faith security research on automotive systems

1.2 Legal Framework

This application operates under exemptions to 17 U.S.C. § 1201 (DMCA Section 1201), specifically:

- **37 C.F.R. § 201.40(b)(7)**: Computer programs that operate motorized land vehicles (diagnosis, repair, modification)
- **37 C.F.R. § 201.40(b)(8)**: Computer programs for interoperability purposes
- **37 C.F.R. § 201.40(b)(12)**: Security research exemption

The application does NOT:

- Facilitate copyright infringement
- Enable piracy of copyrighted content
- Bypass access controls for unauthorized purposes
- Modify entertainment content or media files

The application ONLY:

- Modifies configuration parameters for legitimate repair
- Enables diagnostic access for troubleshooting
- Restores factory functionality (CarPlay/Android Auto)
- Creates backups for maintenance purposes

---

2. Hardware Requirements

2.1 Required Equipment

To fully test this application, the following hardware is required:

A. Android Device

- **Operating System**: Android 8.0 (API 26) or higher

- **USB Support**: USB OTG (On-The-Go) capability
- **Recommended Models**: Samsung Galaxy S10+, Google Pixel 4+, OnePlus 7+

B. MIB2 Infotainment Unit

- **Model**: MIB2 STD2 Technisat Preh
- **Firmware**: T480 (specific firmware version required)
- **Manufacturer**: Preh Car Connect GmbH
- **Found in**: Volkswagen, Audi, SEAT, Škoda vehicles (2016-2019)
- **Cost**: €200-500 (used units available on automotive parts marketplaces)

C. USB-Ethernet Adapter

- **Compatible Models**:
  - ASIX AX88772 chipset
  - ASIX AX88178 chipset
  - Realtek RTL8152 chipset
- **Examples**:
  - TP-Link UE300
  - UGREEN USB 3.0 to Ethernet Adapter
  - Anker USB 3.0 to Ethernet Adapter
- **Cost**: $15-30

D. USB OTG Cable with External Power

- **Type**: USB-C or Micro-USB (depending on Android device)
- **Power**: 5V external power supply required
- **Reason**: MIB2 unit requires stable power; phone battery insufficient
- **Cost**: $10-20

E. Power Supply

- **Voltage**: 5V DC
- **Current**: Minimum 2A (2000mA)

- **Connector**: USB-A or appropriate for OTG cable
- **Cost**: $10-15

2.2 Total Hardware Cost

Minimum investment for testing: $235-565 USD

Note for Reviewers: Due to the specialized nature of this hardware, we understand that full functional testing may not be feasible. Alternative verification methods are provided in Section 4.

---

3. Connection and Testing Procedure

3.1 Physical Setup

1. **Power the MIB2 Unit**
   - Connect MIB2 unit to 12V power supply (automotive battery or bench power supply)
   - Wait for unit to boot completely (approximately 30-60 seconds)
   - Verify LED indicators show unit is powered on
2. **Connect USB-Ethernet Adapter**
   - Plug USB-Ethernet adapter into MIB2 unit's USB port
   - Verify adapter LED lights up (indicates power and connection)
3. **Connect OTG Cable**
   - Connect USB-Ethernet adapter to OTG cable
   - Connect OTG cable to Android device
   - Connect external 5V power supply to OTG cable's power input
4. **Verify Android Recognition**
   - Android should display "USB Ethernet connected" notification
   - Check Settings → Network & Internet → Ethernet (should show connected)

3.2 Application Testing Steps

Step 1: Launch Application

1. Open MIB2 Controller app on Android device
2. App displays welcome tutorial (4 screens)
3. Navigate through tutorial using “Next” button
4. Tap “Skip Tutorial” to proceed to main screen

Step 2: Connect to MIB2 Unit

1. On main screen, tap “Connect to MIB2” button
2. App attempts automatic connection via USB Ethernet (IP: 192.168.1.1)
3. Connection status updates in real-time:
    - “Connecting…” (yellow indicator)
    - “Connected” (green indicator) if successful
    - “Disconnected” (red indicator) if failed

Step 3: View Device Information

1. Once connected, tap “Device Info” tab in bottom navigation
2. App displays MIB2 unit details:
    - Firmware version (e.g., “T480”)
    - Hardware revision
    - Serial number
    - MAC address
    - Current FEC codes

Step 4: Test FEC Code Management

1. Tap “FEC Codes” tab
2. View current Feature Enable Codes
3. Tap “Backup Current FEC” to save current configuration
4. (Optional) Modify FEC codes for testing

5. Tap “Restore FEC” to revert changes

Step 5: Test Auto-Spoof Feature

1. Tap “Auto-Spoof” tab
2. View current CarPlay/Android Auto status
3. Toggle “Enable CarPlay” switch
4. App sends configuration commands to MIB2 unit
5. Verify status updates reflect changes

Step 6: Test Telnet Access

1. Tap “Telnet” tab
2. Tap “Connect Telnet” button
3. App establishes Telnet connection to MIB2 unit (port 23)
4. Terminal interface displays system prompt
5. Enter diagnostic commands (e.g., `ls`, `ps`, `ifconfig`)
6. Verify command output displays correctly

Step 7: Test Backup Management

1. Tap “Backups” tab
2. Tap “Create Backup” button
3. Enter backup name (e.g., “Test Backup 2026-01-22”)
4. App creates backup file with timestamp
5. View backup list showing all saved backups
6. Tap backup to view details or restore

3.3 Expected Results

Successful Connection:

- Green “Connected” indicator
- Device information populated with real data

- All tabs functional and responsive
- Commands execute without errors

Failed Connection (No Hardware):

- Red “Disconnected” indicator
- Error message: “Unable to connect to MIB2 unit. Please verify USB connection and power supply.”
- Tabs remain accessible but show placeholder data
- App does not crash or freeze

---

4. Alternative Verification Methods for Reviewers

We understand that Google Play reviewers may not have access to specialized automotive hardware. Here are alternative methods to verify the application’s legitimacy and functionality:

4.1 UI/UX Review

Without Hardware:

1. Install app on Android device
2. Navigate through welcome tutorial
3. Verify all UI elements are functional:
    - Bottom tab navigation works
    - Buttons respond to touch
    - Text is readable and professional
    - No broken images or layouts
4. Verify app does not crash when hardware is not connected
5. Verify app displays appropriate error messages

Expected Behavior:

- App launches successfully

- Tutorial displays correctly
- Main screen shows “Disconnected” status
- Error messages are user-friendly
- No crashes or ANR (Application Not Responding) errors

4.2 Code Review (Optional)

If Google requires source code review, we can provide:

- Complete source code repository (GitHub)
- Build instructions for reproducible builds
- Documentation of all network communications
- Explanation of cryptographic operations (if any)

4.3 Video Demonstration

We can provide a video demonstration showing:

1. Physical hardware setup
2. Connection procedure
3. All app features in operation
4. Real-time interaction with MIB2 unit

Video URL: Available upon request (can be uploaded to private YouTube link)

4.4 APK Analysis

Reviewers can use standard Android analysis tools:

- **APK Analyzer** (Android Studio): Verify permissions, resources, code structure
- **jadx**: Decompile APK to review Java/Kotlin source code
- **Wireshark**: Capture network traffic to verify communication protocols
- **ADB Logcat**: Monitor app logs during operation

Expected Findings:

- No suspicious permissions (e.g., SMS, Contacts, Location)
- No obfuscated or encrypted payloads
- Standard network protocols (TCP/IP, Telnet)
- No external API calls to unknown servers
- All data processing occurs locally on device

---

5. Privacy and Security

5.1 Data Collection

MIB2 Controller does NOT collect, store, or transmit:

- User personal information
- Location data
- Device identifiers (IMEI, Android ID)
- Usage analytics
- Crash reports to external servers

MIB2 Controller ONLY accesses:

- USB Ethernet connection (for MIB2 communication)
- Local file storage (for backup files)
- Network sockets (for Telnet connection to MIB2 unit)

5.2 Network Communication

All network communication occurs exclusively between:

- **Android device** ↔ **MIB2 unit** (via USB Ethernet, IP: 192.168.1.1)

No external servers or cloud services are contacted.

5.3 Permissions

The app requests only essential permissions:

- `INTERNET` : Required for TCP/IP communication with MIB2 unit via USB Ethernet
- `ACCESS_NETWORK_STATE` : Required to detect USB Ethernet connection status
- `WRITE_EXTERNAL_STORAGE` : Required to save backup files (Android 10 and below)

No sensitive permissions requested (Camera, Microphone, Location, Contacts, SMS, etc.)

---

6. User Support and Documentation

6.1 In-App Help

The app includes:

- Welcome tutorial (4 screens) explaining setup process
- Contextual help text on each screen
- Error messages with troubleshooting tips
- FAQ section (accessible from Settings)

6.2 External Support

Email Support: feplazas@gmail.com

- Response time: Within 24-48 hours
- Languages: English, Spanish

Privacy Policy: https://feplazas.github.io/mib2-controller-privacy/

- Clearly states no data collection
- Explains app purpose and legal compliance
- Updated: January 2026

6.3 User Documentation

Additional documentation available:

- Hardware compatibility list

- Troubleshooting guide
- FEC code reference
- Video tutorials (YouTube, unlisted)

---

7. Compliance with Google Play Policies

7.1 Device and Network Abuse Policy

MIB2 Controller complies with Google Play’s Device and Network Abuse policy:

✅ **Does NOT:**

- Interfere with other apps or Android system
- Display unauthorized ads or notifications
- Collect data without user consent
- Modify system settings without permission
- Execute remote code or download executable code

✅ **DOES:**

- Clearly disclose its purpose (diagnostic tool)
- Request only necessary permissions
- Operate transparently with user awareness
- Provide clear documentation and support

7.2 Deceptive Behavior Policy

MIB2 Controller complies with Google Play’s Deceptive Behavior policy:

✅ **Accurate Representation:**

- App name clearly describes functionality
- Store listing accurately describes features
- Screenshots show actual app interface

- No misleading claims or false promises

✅ **Transparent Functionality:**

- All features clearly explained in app
- No hidden functionality or backdoors
- No impersonation of other apps or brands
- No misleading permissions requests

7.3 Malicious Behavior Policy

MIB2 Controller complies with Google Play's Malicious Behavior policy:

✅ **No Malicious Code:**

- No malware, spyware, or trojans
- No code obfuscation (beyond standard ProGuard)
- No unauthorized access to device resources
- No exploitation of security vulnerabilities

✅ **Safe Network Communication:**

- All network traffic is legitimate diagnostic communication
- No command-and-control (C2) behavior
- No data exfiltration
- No unauthorized remote access

7.4 User Data Policy

MIB2 Controller complies with Google Play's User Data policy:

✅ **No User Data Collection:**

- App does not collect personal information
- App does not collect sensitive information
- App does not transmit data to external servers
- Data Safety form accurately reflects "No data collected"

8. Testing Credentials (If Required)

8.1 Demo Mode

If Google requires testing without hardware, we can provide a **Demo Mode APK** that:

- Simulates MIB2 unit responses with mock data
- Allows reviewers to navigate all app features
- Displays realistic device information and logs
- Does not require physical hardware

Demo Mode APK: Available upon request

8.2 Test Account

Not applicable - This app does not require user accounts or authentication.

8.3 Remote Testing Assistance

If Google reviewers require assistance, we offer:

- **Live video call** (Google Meet, Zoom) to demonstrate app with real hardware
- **Remote access** to test device with hardware connected (via TeamViewer, AnyDesk)
- **Custom test build** with additional logging for reviewer verification

Contact: feplazas@gmail.com to schedule remote testing session

9. Frequently Asked Questions (for Reviewers)

Q1: Is this app used for hacking or unauthorized access?

A: No. This app is a legitimate diagnostic and repair tool, similar to OBD-II scanners used by mechanics. It operates under DMCA exemptions for vehicle repair and interoperability. All functionality is limited to configuration changes necessary for lawful repair activities.

Q2: Can this app be used to pirate content or bypass DRM?

A: No. The app does not interact with entertainment content, media files, or DRM systems. It only modifies configuration parameters (FEC codes) that control hardware features like CarPlay/Android Auto connectivity.

Q3: Why does the app need INTERNET permission if it doesn't connect to external servers?

A: The INTERNET permission is required for local TCP/IP communication with the MIB2 unit via USB Ethernet. Android requires this permission for any socket-based communication, even on local networks. The app never contacts external servers.

Q4: How can we verify the app's claims without the hardware?

A: We recommend:

1. UI/UX review (verify app doesn't crash without hardware)
2. APK analysis (verify no suspicious code or permissions)
3. Network traffic analysis (verify no external connections)
4. Request demo mode APK or video demonstration
5. Contact developer for remote testing session

Q5: Is this app legal in all countries?

A: The app is legal in the United States under DMCA exemptions. Users are responsible for compliance with local laws in their jurisdiction. The app includes disclaimers and warnings about legal responsibility.

Q6: What prevents misuse of this app?

A: The app requires specialized hardware (MIB2 unit) that is only available to vehicle owners, technicians, and researchers. The app cannot be used for general hacking or unauthorized access because it only communicates with MIB2 units. Additionally, all modifications are reversible via backup/restore functionality.

---

10. Contact Information

Developer: Felipe Plazas
Email: feplazas@gmail.com
Response Time: 24-48 hours
Languages: English, Spanish

Privacy Policy: https://feplazas.github.io/mib2-controller-privacy/
Package Name: com.feplazas.mib2controller
Version: 1.0.0 (versionCode 7)

---

11. Appendix: Technical Specifications

11.1 Network Configuration

- **Protocol**: TCP/IP over USB Ethernet
- **IP Address**: 192.168.1.1 (MIB2 unit static IP)
- **Subnet**: 255.255.255.0
- **Ports Used**:
  - Port 23 (Telnet)
  - Port 80 (HTTP for FEC code management)
  - Port 5555 (Custom diagnostic protocol)

11.2 Supported MIB2 Firmware Versions

- **Primary**: T480 (fully tested and supported)
- **Secondary**: T470, T475 (partial support, community-tested)
- **Unsupported**: T400 series and earlier (different hardware architecture)

11.3 File Formats

Backup Files:

- Format: JSON
- Extension: `.mib2backup`
- Contents: FEC codes, configuration parameters, timestamps
- Encryption: None (plain text for transparency)
- Storage: Local device only (`/storage/emulated/0/MIB2Controller/backups/`)

11.4 Build Information

- **Framework**: React Native with Expo SDK 54
- **Build System**: EAS Build (Expo Application Services)
- **Keystore**: Managed by Expo (ID: evIXv6BtNF)
- **Signing**: Release builds signed with production keystore
- **Obfuscation**: Standard ProGuard rules (no custom obfuscation)

---

12. Conclusion

MIB2 Controller is a legitimate, transparent, and legally compliant diagnostic tool for automotive infotainment systems. While full functional testing requires specialized hardware, the app's code, permissions, and behavior can be verified through standard Android analysis tools.

We are committed to working with Google Play reviewers to address any concerns and provide any additional information or demonstrations required for approval.

Thank you for your thorough review.

Document Version: 1.0
Date: January 22, 2026
Author: Felipe Plazas
Contact: feplazas@gmail.com