Embedded and Device Software Testing Services

MOSCHIP®

Embedded Software Characteristics

- Designed for a specific device with a limited number of functions
- Tightly integrated with the device's hardware
  - Microcontroller or specific-purpose integrated circuit (IC)
- Resource-constrained, due to limited memory and processing
- Increasingly connected via multiple interfaces and protocols

Examples of devices

- Consumer Electronics:
  - Smart TVs, washing machines, digital watches.
- Automotive:
  - Automotive engine management and edge sensors/actuators
- IoT :
  - BLE/WiFi/ZigBee-based sensor integration platform, LoRaWAN-based Asset Monitoring
- Connected devices:
  - IoT-enabled air purifier

Software Test Services

- Board Support Package (BSP)
- Device drivers
- Protocol stack
- OS and file system validation
- Cross-platform
- Network
- Security

These can be tailored for consumer, automotive, connected devices, and IoT

Snapshot of Testing for Connected Devices

Wireless Audio for Android TV Testing

- Wireless Audio transmission from source to speaker
- Software solution part of Android TV Package
- Audio Quality, Latency testing

Integration

- VIM4 Khadas
- SEI Android TV Box
- Custom OEM Android TV OS

Solution

VIM4| SEI

Android TV

Audio Analysis

Web App for Wireless Audio Solution Testing

- Enterprise app for manufacturing testing used by speaker system OEMs
- Firmware programming and device configurations
- OTA Upgrade custom configurations

Integration

- Google Chrome over serial COM connected to device.
- Server APIs for custom client

Solution

Windows

Linux

Python

Firmware/OS Development & Test

Server App Development

Snapshot of Testing for Connected Devices & IoT

IoT Based Automation Framework

- End to End Test Automation Framework, CI/CD
- Wi-Fi BLE Co-ex testing
- BLE functionality testing, Stability testing
- Current/Power measurement tests
- Mobile, Web, API and End Device Automation

Integration

- End To End testing of IoT ecosystem
- Testlink, Jenkins, GitGub
- Post execution report generation

Solution

Mobile

Web

IoT Device

Framework Development

Testing for Automotive Domain

ECU & Embedded Software

- Python / CAPL / Robot Framework test automation
- HIL / SIL / bench validation of firmware
- Regression, timing, and resource profiling
- Fault injection and boundary condition tests

OS, Middleware & Frameworks

- Task profiling and memory / CPU monitoring
- HAL & driver validation with instrumentation
- Bootloader and OTA testing in sandboxed environments
- Embedded Linux / RTOS testing pipelines

Communication & Network

- Network simulation & monitoring tools (CANoe, Wireshark, custom sniffers)
- Latency, bus load, and error recovery validation
- Interoperability testing for multi-ECU setups
- Diagnostic sequence execution

Testing for Automotive Domain

Safety & Reliability

- Watchdog, memory, and fault injection tests
- Stress, soak, and boundary-condition testing
- Error handling and recovery validation
- FuSa: Safety concept & analysis (ISO 26262 compliant), Safety assessments & audits, Tool Qualification, Static & dynamic analysis, coverage (MC/DC), SEooC validation

Connectivity & Telematics

- Vehicle-to-cloud / mobile app integration tests
- OTA update simulation & rollback verification
- Data capture, logging, and replay testing
- MQTT / REST / telematics message validation

AI, Sensors & Perception

- Sensor fusion testbeds (camera, radar, LiDAR)
- Edge AI model inference profiling & accuracy tests
- Simulation-based scenario validation (ADAS/DMS)
- Drift detection and fallback scenario verification

Testing for Consumer Domain

Functional & Performance

- End-to-end device + app + cloud workflows
- UI/UX behavior and feature validation
- Boot time, responsiveness, latency, FPS, thermal profiling
- Power consumption and battery optimization

Connectivity & Protocols

- Wireless: Wi-Fi (802.11 a/b/g/n/ac/ax), BLE (GATT, GAP), Zigbee, Thread, Matter
- Wired: USB, HDMI-CEC, Ethernet
- Streaming: RTP/RTSP, HLS, DASH, WebRTC
- Interoperability: DLNA, UPnP, multi-device sync
- Protocol conformance, reconnection, failover, and coexistence testing

OS, Middleware & Frameworks

- Operating Systems: Android (AOSP / TV / Auto), Linux (Yocto / Ubuntu), FreeRTOS, …
- Layers Tested: HAL, middleware components, device drivers, system daemons
- Power state transitions, kernel event handling, app lifecycle validation
- OTA update mechanisms and rollback verification

Testing for Consumer Domain

Audio / Video & Multimedia

- Playback, recording, codec validation (AAC, MP3, H.264, H.265, VP9)
- AV synchronization (lip-sync, jitter, latency)
- DRM & content protection
- Resolution scaling, adaptive bitrate, buffer management
- Smart speaker / microphone array validation (voice trigger, beamforming)

Security, Compliance & Reliability

- Secure boot, encryption, authentication, key management
- Firmware & data protection (OTA, rollback, credential safety)
- Ecosystem integration: Alexa, Google Home, Apple HomeKit
- Compliance with HDMI-CEC, Bluetooth SIG, USB-IF, DLNA
- Reliability, stress, and soak testing across real-world use scenarios

AI, Data & Intelligence Testing

- On-device AI workloads (vision, audio, sensor analytics)
- ML model integration and inference accuracy testing
- AI-assisted UX: voice recognition, gesture control, personalization
- Edge–cloud model synchronization and fallback handling
- Data capture, drift detection, and privacy validation

MosChip Test Automation Framework (MTAF)

Achieve continuous integration and delivery(CI/CD) with **MTAF's end-to-end testing** across embedded devices, web, cloud and mobile.

Mobile	Cloud	Embedded	Web
Android mobile app UI testing	Cloud Application using REST API with HTTP/MQTT	IoT system end to end Functional Testing	Cross browser testing

Tools & Frameworks

Jenkins, Selenium, IBM Spectrum LSF, python, Locust, docker, Kubeflow, Appium, ZAP

Features

- Modular & Customizable
- Multi-Purpose
- Edge to Cloud testing
- Scalable
- Automatic Report Generation
- Parallel Execution

AI Driven Test Engineering Initiatives (Horizontal Enabler Example)

- **AI Copilots**
  - Test Case Generation from the requirements document
  - Unit test case generation from source code
  - Automated test script generation from Manual test cases
- **Test Document Intelligence**
  - Auto-links requirements ↔ test cases ↔ results for traceability
  - Detects outdated, redundant, or conflicting test scenarios
- **Continuous Learning from Test Outcomes**
  - Feedback loop: failed tests and defect logs used to retrain models
- **Process-Level AI Augmentation**
  - Enables early discovery of ambiguous or incomplete requirements
  - Lays the foundation for “AI-augmented QA lifecycle automation”