

## ADLATUS® Software Suite

Automotive & Industry

ADLATUS® Software Suite

ADLATUS® Software Update

FlashCedere® Conformance Tester



2023



## Inhalt

### **ADLATUS® Software Suite (Embedded Produkte)**

ADLATUS® Base Software

#### **ADLATUS® Update**

ADLATUS® Bootmanager

ADLATUS® Flashloader

ADLATUS® Reprog

ADLATUS® Testloader

### **ADLATUS® Flasher (PC Tool)**

### **FlashCedere® Conformance Tester (PC Tool)**

12

SMART IN OVATION GmbH  
Meitnerstraße 10 70563 Stuttgart  
Telefon +49 - 711 255 212 - 0  
E-Mail [info@sio-gmbh.de](mailto:info@sio-gmbh.de)  
Web [www.sio-gmbh.de](http://www.sio-gmbh.de)





# ADLATUS® Software Suite

Der geniale Software Baukasten ist die Grundlage für die Embedded Produkte ADLATUS® Base Software, - Bootmanager, - Flashloader,- Reprog und -Testloader für Automotive und Industrie Elektronik



## Kurzbeschreibung

Die ADLATUS® Software Suite ist der geniale Software Baukasten, aus dem alle ADLATUS® Softwareprodukte hergestellt werden. Eine objektorientierte Software Architektur ermöglicht den einfachen Tausch der Software Komponenten.

Seit dem Jahr 2020 erfüllen alle aus diesem Baukasten geschöpften ADLATUS® Software Produkte das Prozesshaus ASPICE 3.1 CL2.

## Vorteil ADLATUS®

- Maximale Synergie durch Software "Gleichteilekonzept". Softwarequalität ASPICE V3.1 Capability Level 2.
- Software Sicherheit gemäß ISO26262 ASIL B/D.
- IT-Sicherheit gemäß TISAX Capability Level 2.
- Software Update Lösung und Basissoftware für alle Steuergeräte, für die eine AUTOSAR Lösung zu teuer, zu schwerfällig ist oder zu viel Speicher benötigt.
- OEM-spezifische Software-Anpassungen und Softwaretest (umfangreiche Testbibliotheken) und ASPICE-Assessments im Kundenauftrag.
- Schnell und flexibel: Software Support aus Deutschland.
- Nur genutzte Software "On Board".
- Saubere Trennung von generischem und hardwarespezifischem Code.
- ADLATUS® steht für einfachen und verständlichen Quellcode mit einem Minimum an Compilerschaltern.
- Basisplattform Software für Automotive Steuergeräte,
- Industrieelektronik und Multiprozessor-Lösungen.
- Auf vielen Prozessoren verfügbar.
- Größte OEM und Tier 1 als Referenzkunden.
- 25 Jahre ADLATUS in über 70 Millionen Steuergeräten.
- Software Buyout, General- oder Projektlizenz.

## ADLATUS® Base Software

- Schlankes, "lock free" CAN-Betriebssystem mit Task Verwaltung und Präemptives Task Scheduling.
- UDS-Server als generische „Middleware“.
- Diagnosekommunikation, ISO-TP ISO 15765-2.
- CAN-/CAN-FD Treiber und CAN-Signalverarbeitung,

## ADLATUS® Bootmanager

- Generischer "Bootverzweiger" nach Reset der Elektronik.
- Sicherheit ISO26262 ASIL D.

## ADLATUS® Flashloader

- Generischer Flashloader zur Software Update Programmierung von Steuergeräten.
- Einfache Anpassung an die OEM Flash-Protokolle.
- Sicherheit ISO26262 ASIL B.

## ADLATUS® Reprog

- Flashloader Austausch.
- Zugriffssicherheit wie bei der OEM Software Update.

## ADLATUS® Testloader

- Download und Ausführung von Testroutinen in der Produktion, -Entwicklung und -Befundung.
- Kommunikation über Diagnosekommunikation.

## Service und Wartung

- ASPICE-Consulting.
- Projektabwicklung inkl. ASPICE CL2 Assessment.
- Wartungsvertrag für ADLATUS und Projekte.



# ADLATUS® Base Software

Embedded Basissoftware für  
Automotive und Industrie Steuergeräte



## Kurzbeschreibung

Die ADLATUS® Basissoftware ist ein kostengünstiges schlankes, konfigurierbares Betriebssystem, das für Automotive Steuergeräte entwickelt wurde, die über wenig Speicherplatz verfügen.

Die Basissoftware kann für z. B. CAN-Steuergeräte eingesetzt werden und bringt neben dem Betriebssystem wesentlichen Grundfunktionen für ein Steuergerät mit:

- Schlankes Betriebssystem mit Task Verwaltung und Präemptives Task Scheduling.
- UDS-Server als generische „Middleware“.
- Diagnosekommunikation
- ISO-Transportprotokoll nach ISO 15765-2.
- CAN-/CAN-FD Treiber und CAN-Signalverarbeitung.
- Echtzeitfähiges Betriebssystem mit Lock free Algorithmen ohne Sperrung von Interruptroutinen.
- Eine Softwareprodukt aus der ADLATUS® Software Suite.
- Die Softwareentwicklung erfolgt gemäß ASPICE Prozesshaus V3.1 CL2 und ISO26262 (ASIL B/D) sowie TISAX CL2.

## Typische Einsatzgebiete

- Basissoftware Plattform inklusive Software Update Lösung.
- Für Steuergeräte, für die AUTOSAR und ähnliche Lösungen zu teuer oder für den Mikrocontroller zu groß sind.
- Basisplattform Software für Industrieelektronik, Automotive Steuergeräte, Multiprozessor-Lösungen etc..

## Nutzen

- ADLATUS® Softwarearchitektur angelehnt an Architektur AUTOSAR. Einfach zu konfigurieren.
- Softwarequalität gemäß ASPICE V3.1 CL2, ASIL B/D.
- Geringer Anpassungsaufwand für ein neues Projekt und hohe Software-Qualität durch saubere Trennung von generischem Code und hardwarespezifischem Code.
- Durch den modularen Aufbau ist ein einfacher Tausch von Softwarekomponenten möglich.
- Durch die OO-Software Architektur mit einer Implementierung in „C“ steht ADLATUS® für einfachen und verständlichen Quellcode mit einem Minimum an Compilerschaltern.
- Leichte Portierbarkeit auf marktgängige Automotive-Prozessoren und OEM-Protokolle.
- Referenzkunden u. a. Volkswagen AG, Endress + Hauser, Himmelwerk GmbH.

## Wartung und Support

Für unsere ADLATUS® Software Produkte sowie darauf aufbauende kundenspezifische Lösungen bieten wir umfassende Software Wartungs- und Support Verträge an.





# ADLATUS® Base Software

Embedded Basissoftware für  
Automotive und Industrie Steuergeräte



## Konfigurationstool CAN-Kommunikation

- Konfigurationstool zur Erstellung der Steuergeräte CAN Kommunikationsmatrix.
- Erzeugung der Konfigurationsdateien per Code-Generator für „C“.
- Die CAN-Kommunikation erfolgt konfigurierbar im Interrupt- oder im Polling-Betrieb..
- Alle Messageboxen des Controllers werden unterstützt.
- Konfiguration vergleichbar mit „CAN Interaction Layer“.
- Verteilung der Botschaften auf die CAN-Mailboxen (mit/ohne FIFO/Queue), CAN-Mailbox exklusiv einer CAN-Botschaft zuordenbar.
- Automatische Generierung der CAN Acceptance-Filter.
- Berechnung von Timing-Parameter, Baudrate und Abtastzeitpunkt.
- Mehrcontroller-Systeme - Für jede ECU-Variante kann eine Konfiguration erzeugt werden.
- Zur Laufzeit der ECU kann eine Variante der Konfiguration aktiviert werden.
- Umschaltung zur Laufzeit inkl. Acceptance-Filter.

## CAN-Message Kommunikation / Treiber

- Umfassende Funktionen für CAN-Botschaften und CAN-Signale.
- Optimierte für Mikrocontroller mit wenig CAN-Mailboxen.
- CAN-Botschaften werden softwaretechnisch mit Hilfe eines Algorithmus optimal priorisiert und über die zur Verfügung stehenden CAN-Mailboxen versandt.
- CanOnline CanOffline CanReset BusSleep.
- CanListenOnlyMode (Stummschaltung des CAN).
- CanTransmit, Canreset (Reset nach Bus-Off).

## Callback zu folgenden Ereignissen

### Callback RxIndication

- Info über den erfolgreichen Empfang einer Botschaft.
- Jeder Botschaft kann eine Funktion hinterlegt werden.
- Die Applikation kann die Daten direkt verarbeiten oder diese in den Empfangspuffer kopieren lassen.

### Callback Transmit Confirmation

- Info CAN-Botschaft erfolgreich versendet.
- Jeder Botschaft kann eine Funktion hinterlegt werden.

### Callback PreTransmit

- Dynamische Message-Inhalte (Botschaftszähler, CRC) können vor Versand eingetragen werden.
- Jeder Botschaft kann eine Funktion hinterlegt werden.

### "Wakeup-/ Bus Off" mit falschem DLC empfangen.

## Task Management

- Präemptives Task Scheduling mit konfig. Zykluszeiten.
- Unterbreungskriterium ist ausschließlich die Priorität.
- Definierte Start- und Stopp Sequenzen.
- Konfigurierbare Reihenfolge der zyklischen Tasks.
- Schutz vor Mehrfachaktivierung zyklischer Tasks.
- Meldungen zu Task-Überlauf / Restlaufzeit des Taskzyklus.

## Laufzeitmessung für jede zyklische Task und Messung der

- Genutzten Laufzeit je Taskzyklus.
- Messung Gesamt-Laufzeit für alle Module verfügbar.
- Beispiel-Applikation mit allen Softwarekomponenten ist Teil der Softwarelieferung.



# ADLATUS® Base Software

Embedded Basissoftware für  
Automotive und Industrie Steuergeräte



## CAN Signalzugriff

- Die Signal-Zugriffsschicht ist hardwareunabhängig.
- Zugriff auf alle Signale in einer CAN-Botschaft
  - (Rx und Tx).
- Direkter Zugriff auf alle Signale über Makros.
- Zyklischer Versand der Signale mit einstellbarer Zykluszeit.
- Es gibt verschiedene Sende-Modi wie „zyklisch senden“, „onChange senden“, „onWrite senden“ etc..
- Automatisches Multiplexen mehrerer logischer Signale in einer CAN-Nachricht durch sequenzielles Senden der Signale mit demselben CAN-Identifizier.
- Sende-Modi können für jedes Signal kombiniert werden.
- Für alle Sende-Modi wird ein einstellbarer Mindest-Sendeabstand („Inhibit-Zeit“) sichergestellt.
- Timeoutüberwachung des zyklischen Signalempfangs

## Transportprotokoll gemäß ISO 15765-2

- Voll- oder Halb-Duplex-Betrieb auswählbar.
- Unterstützung mehrerer Diagnosezugänge (UDS-Clients).
- Multi-Channel fähig auf einem oder mehreren physikalischen Zugängen.

## Over the Air Update Programming (OTA)

- Überwachung und Priorisierung gleichzeitiger Zugriffe verschiedener Werkstatt-/OTA Tester

## UDS-Server

- Diagnose Sessions werden vom UDS-Server verwaltet.
- Der Server stellt die gesamte „Hülle“ des "Session Handlings" sowie Freischaltsverwaltung der UDS-Services zur Verfügung.
- Das komplette Handling der Services wird vom Server erledigt, lediglich die spezifischen Diagnose Services müssen kundenseitig definiert ggfs. implementiert werden.
- Der Server verfügt über Timeout-Überwachungen.
- Kundenspezifische Handler pro UDS-Service, je eine C-Funktion.
- Handling von Tester Present, Response Pending etc..
- Multi-Client mit Abbruch-Priorisierung.
- Der UDS-Server entscheidet per Konfiguration, in welcher Diagnose Session welche Services ausgeführt werden dürfen.
- Die UDS-Services werden über eine einheitliche Schnittstelle aufgerufen.
- "Response Pending" Botschaften für Services, die viel Zeit zur Bearbeitung benötigen, werden automatisch vom Server im Hintergrund generiert.
- „Bypassing“ und automatisiertes „TesterPresent“ Handling.
- UDS Multi-Client Support mit Priorisierung der Verbindung.
- Der höchst-priorisiert anfragende TP-Kanal beendet alle niederpriorisierten TP-Kanäle automatisch.





# ADLATUS® Update

ADLATUS® für die Update Programmierung von Automotive und Industrie Steuergeräten



**ADLATUS® Bootmanager**  
Generischer Bootmanager

**ADLATUS® Flashloader**  
Generischer Flashloader

**ADLATUS® Testloader**  
Testsoftware Lösung für Hardwaretest ohne Anwendungssoftware

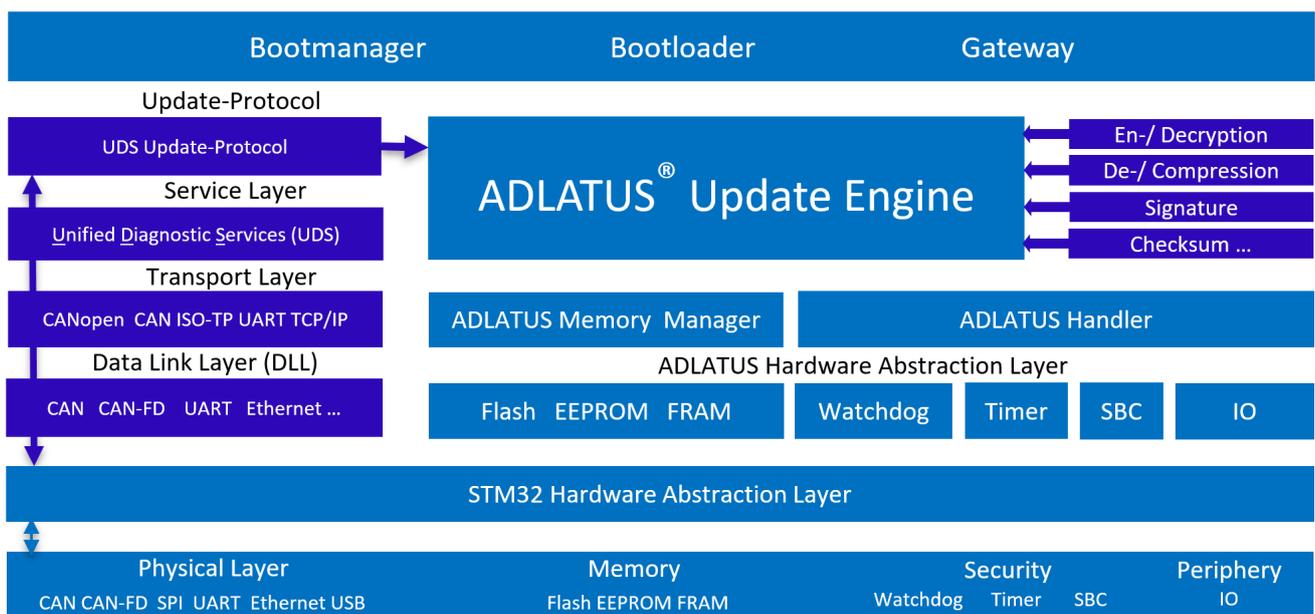
**ADLATUS® Software Komponenten**  
Verschlüsselung, Signatur, Kompression, Seed & Key, Gateway, ECU-Komponentenschutz uvm.

**ADLATUS® Flasher**  
PC-Tool zur Programmierung von Steuergeräten und DLL für die Programmierung in der Produktion

**FlashCedere® Conformance Tester, TesterRunner und Test Libraries**  
FlashCedere prüft den ADLATUS® Flashloader auf Einhaltung der Spezifikationen (Zeit-Inhalt-Sequenz)

## Nutzen

- Software Update-Komplettlösung für Automotive Steuergeräte.
- Erfahrung mit 70 Millionen „ADLATUS® Fahrzeugen“.
- Konfigurierbarer Flashloader mit minimaler Hardwareabhängigkeit.
- ANSI-C Code, der nur die tatsächlich verwendeten Software Komponenten enthält.
- FlashCedere Conformance Tester für automatisierten Test und Freigabe des ADLATUS®.



## ADLATUS® Struktur: Beispiel für STM32 Mikrocontroller





# ADLATUS® Bootmanager

Generischer Bootmanager für  
Automotive und Industrie Steuergeräte



## Kurzbeschreibung

Der ADLATUS® Bootmanager stellt nach dem Neustart eines Steuergerätes sicher, dass die im System befindlichen Anwendung sicher und korrekt aufgerufen werden. Der Bootmanager analysiert vorab die Ursache für den Neustart des Steuergerätes. Wurde der Neustart durch einen Werkstatttester (Service) ausgelöst, wird der Flashloader aufgerufen, um die Update Programmierung des Steuergerätes zu ermöglichen. Wurde der Neustart des Steuergerätes durch eine Störung ausgelöst, so wird die Anwendung des Steuergerätes gestartet. Der Bootmanager prüft vor Start der Zielanwendung, ob diese fehlerfrei im Speicher steht. Die Entscheidung, ob bzw. welche Softwareanwendung nach dem Neustart eines Steuergerätes gestartet wird, beinhaltet verschiedene Sicherheitsrisiken, die durch den ADLATUS® Bootmanager gemäß ASIL D reduziert werden. Wurde z. B.S der Neustart durch einen Speicherfehler der Standardapplikation ausgelöst, muss der Bootmanager prüfen, ob der Neustart dieser Applikation noch zulässig ist und prüft, ob diese Applikation fehlerfrei im Programmspeicher gespeichert ist. Ist dies nicht der Fall, darf der Bootmanager diese Applikation nicht starten. Die Überprüfungsverfahren sind umfassend konfigurierbar. Meist stehen mehrere Anwendungen zur Auswahl, z.B. die Kundenapplikation, der Flashloader oder ein End-Of-Line Test. Die Auswahl ist sicherheitsrelevant, da eine Fehlentscheidung u. U. dazu führt, dass das Steuergerät nach Neustart fehlerhaft funktioniert. Der Bootmanager unterstützt umfangreiche Sicherheits- und Logistik-Funktionen und einen "Secure Boot". Eine unvollständig programmierte Anwendung darf der Bootmanager nicht aufrufen! Der Bootmanager erhöht die Funktionssicherheit durch Berechnung von Prüfsummen der Anwendungen oder einen Secure Boot.

## Aufbau

Diese sicherheitskritischen Vorgänge sind im Bootmanager als Komponenten so strukturiert, dass die Sicherheitslevels ASIL B bzw. ASIL D erreicht werden. Die Architektur erlaubt eine umfassende Konfiguration der Funktionalität des Bootmanagers. Die "Boottabelle" in Verbindung mit einer verlinkten "Operationstabelle" bestimmt das Bootverhalten des Steuergerätes nach Reset. Hier wird festgelegt, welche Anwendung in Verbindung mit welchen "Aktionen" nach Reset des Steuergerätes wie und wohin geladen werden.

ADLATUS® Bootmanager „Boottabelle“

Aktionstabelle für Zielanwendungen	Operationstabelle Aktionen/Skripte
ADLATUS® Bootloader	Validierung: CRC, Hash, Pattern,...
ADLATUS® Testloader	App → RAM-Kopie
App 1	App → Dekompression
App 2	App → Entschlüsselung
...	Secure Boot: Hardware abhängig
	Schlüsseltausch Krypto
	...

Wesentlichen Aufgaben des Bootmanagers sind:

- Start des Mikrocontrollers/Prozessors.
- Initialisierung einer minimalen Hardware-Umgebung
- Überwachung von Speicherverletzungen.
- Auswahl der korrekten Zielanwendung
- Auswahl, Überprüfung und Aufruf von Anwendungen.
- Einhalten von Aufstartzeit, Speicherverbrauch etc..

## Nutzen

- Konfigurierbarer Bootmanager ASIL B oder ASIL D.
- Secure Boot.
- Referenzen: Volkswagen AG, Endress+Hauser uvm..
- Softwareentwicklung ASPICE V3.1 (CL2), ISO26262 .
- Unternehmen TISAX CL2.





## ADLATUS® Flashloader

Generischer Flashloader für  
Automotive und Industrie Steuergeräte



### Kurzbeschreibung

Der ADLATUS® Flashloader ist eine professionelle Lösung für das Softwareupdate von Steuergeräten über den gesamten Lebenszyklus.

Der ADLATUS® Flashloader bringt umfangreiche, standardisierte Sicherheits- und Logistik-Funktionen aus dem Bereich der Automotive und der Industrie Steuergeräte mit.

Er unterstützt unter anderem Bussystem wie CAN, CAN-FD, SPI, UART, LIN oder Ethernet.

Für die automatisierte Qualitätssicherung unserer Software kommt unser FlashCedere Conformance Tester mit generischen Test-Bibliotheken (Python Skript) zum Einsatz.

Der ADLATUS® Flasher kommt als Programmier-Tool (PC) in Entwicklung, Produktion und Service zum Einsatz.

Die ADLATUS® Flasher Software-DLL wird zur Integration der Programmierung in der Produktion eingesetzt (Programmierplatz).

Die Softwareentwicklung des ADLATUS® Flashloader erfolgt gemäß ASPICE Prozesshaus V3.1 (CL2) und ISO26262 (ASIL B, ASIL D Bootmanager).

### Hochflexibel und ASPICE CL2

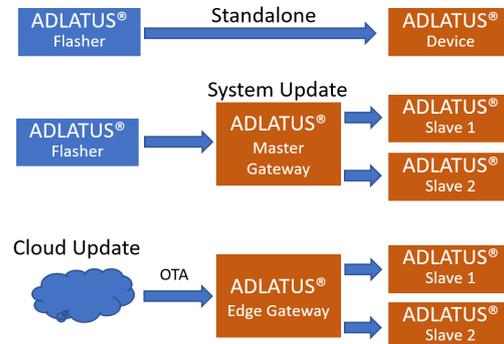
Der generische ADLATUS® Flashloader wird stetig auf neue Mikrocontroller portiert. Er unterstützt u. a. folgende externe Komponenten:

- FRAM, EEPROM und FLASH Speichermodule.
- Interne und externe Watchdog-Module.
- Slave-CPU's werden über das ADLATUS® Gateway programmierbar, zum Beispiel die "Cell-Controller" eines Batterie Management Systems.
- Externen DSPs, FPGAs oder ASICs werden über die ADLATUS Gateway-Komponente programmiert.

### Wartung und Support

Für unsere ADLATUS und FlashCedere Software Produkte sowie darauf aufbauende kundenspezifische Lösungen bieten wir umfassende Software Wartungs- und Support Verträge an.

### Typische Anwendungsfälle



### Nutzen

- 25 Jahre erprobte ADLATUS® Zuverlässigkeit.
- Programmieren von Master- und Slave-Systemen.
- Einfache Realisierung von kundenspezifischen
- Update-Protokollen über CAN/CAN-FD oder andere Kommunikationsbusse wie SPI, UART, LIN etc..
- Verschlüsselung der Flashware für externe Produktion.
- Signatur der Flashware.
- Flashware Datenkompression für kurze Update-Zeit
- Referenzen: Volkswagen AG, R. Bosch GmbH, Endress+Hauser SE+Co. KG, BMG GmbH, ...
- Softwareentwicklung ASPICE V3.1 (CL2), ASIL B/D .
- Unternehmen TISAX CL2.





# ADLATUS® Reprog

Flashloader Austausch für  
Automotive und Industrie Steuergeräte

ADLATUS® Flashloader



## Kurzbeschreibung

Softwarelösung zum Tausch eines vorhandenen Flashloaders in einem Steuergerät.

Der ADLATUS® Reprog ist die Lösung zum Austausch eines im Steuergerät vorhandenen ADLATUS® Flashloaders gegen eine neue Flashloader Softwareversion.

Die Programmierung eines neuen Flashloaders erfolgt vergleichbar der Programmierung einer neuen Anwendungssoftware. Hierbei kommen die gleichen Safety- und Sicherheitsmechanismen wie bei der Programmierung einer Anwendungssoftware zum Einsatz!

Die Softwareentwicklung des ADLATUS® Flashloader erfolgt gemäß ASPICE Prozesshaus V3.1 (CL2) und ISO26262 (ASIL B) und TISAX CL2.

## Nutzen

- 25 Jahre erprobte ADLATUS® Zuverlässigkeit.
- Standardisierter Flashloader Tausch.
- Referenzen: Volkswagen AG, R. Bosch GmbH, Endress+Hauser SE+Co. KG, BMG GmbH, ...
- Softwareentwicklung ASPICE V3.1 (CL2), ASIL B/D .
- Unternehmen TISAX CL2.

## Wartung und Support

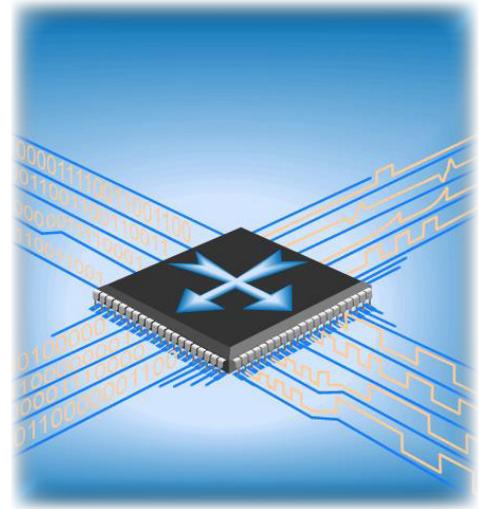
Für unsere ADLATUS® und FlashCedere® Software Produkte sowie darauf aufbauende kundenspezifische Lösungen bieten wir umfassende Software Wartungs- und Support Verträge an.





# ADLATUS® Testloader

Standardisierte Embedded Testsoftware Lösung  
für Hardwaretest ohne Anwendungssoftware  
für Automotive und Industrie Steuergeräte



## Kurzbeschreibung

Bei herkömmlichen End-of-Line Prüfungen werden entweder residente Prüffunktionen verwendet oder notwendige Testsoftware nach der Prüfung in der Produktion überschrieben, indem man sie durch die Applikationssoftware ersetzt. Beide Verfahren haben Nachteile. Der ADLATUS® Testloader ist fester Bestandteil des Steuergerätes, die Testfunktionen jedoch werden nur für die Prüfung des Steuergerätes vom Testloader in den RAM-Speicher geladen und dort ausgeführt. Alle Testroutinen können während des gesamten Lebenszyklus eines Steuergerätes modifiziert oder erweitert werden. Ein weiteres, wichtiges Einsatzgebiet des ADLATUS® Testloaders ist neben dem End-of-Line Test auch die Befundung defekter Steuergeräte. Die Testfunktionen eingesetzt werden, ohne das Steuergerät öffnen zu müssen. Speziell für die Befundung einwickelte oder angepasste Testschritte sind effizient. Im Gegensatz zu „mitfahrenden“ Testfunktionen benötigt der Testloader nur soviel Speicher wie die umfangreichste Testroutine. Neben der Ressourcenersparnis und der Möglichkeit einer nachträglichen Verfeinerung des Prüfverfahrens ist die Abkoppelung der Entwicklung der Testsoftware-Routinen von der Anwendungssoftware ein weiterer Vorteil. Da Testloader und Anwendungssoftware nur minimale Abhängigkeiten haben, kann der Test parallel unabhängig von der Anwendung entwickelt und gepflegt werden. Der Testloader verwendet für die Kommunikation mit dem Prüfautomaten vorhandenen Schnittstellen. Dies können sowohl CAN als auch andere Automotive- und Industrie-Bussysteme sein. Natürlich können auch steuergeräteinterne Schnittstellen genutzt werden, um zum Beispiel mit weiteren Mikrocontrollern oder Peripherieeinheiten zu kommunizieren.

Ein technisch zur Anwendungssoftware gleichwertiger Zugriffsschutz verhindert zuverlässig einen unautorisierten Zugriff auf die Elektronik. Mittels Signaturen und verschlüsselter Testfunktionen können bei erhöhtem Sicherheitsbedarf zusätzliche Schutzmechanismen integriert werden. Der Einsatz von Testfunktionen kann per Download einfach variiert werden. Die für alle Testroutinen einheitliche UDS-Service Aufrufchnittstelle ermöglicht dies. Damit ist die Anzahl der möglichen Prüfschritte unbegrenzt. Die Downloadzeit der Testfunktionen ist im Verhältnis zur Prüfzeit vernachlässigbar bzw. beeinflusst die Taktzeiten nur geringfügig.

## Einsatzgebiete

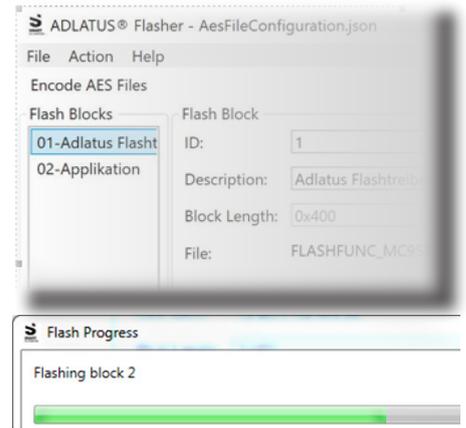
- Hardware Inbetriebnahme, Verifikation, Validierung und Erprobung.
- End-of-Line Hardware Funktionstest.
- Bedienung wie Seriennummern und Datensätze.
- Ausfall-Befundung.

## Nutzen

- Standardisierter und durchgängiger Hardwaretest auf Basis der ISO Standard 14229-1: Road vehicles-Unified diagnostic.
- Test unabhängig von Anwendungssoftware.
- Testsoftware sicher, da nicht permanent im Speicher.
- Signatur der Testware um Authentizität sicher zustellen.
- Testsoftware verbraucht keinen teuren Speicherplatz.
- Referenzen: R. Bosch GmbH, Huf Elektronik, ...



# ADLATUS® Flasher



Der ADLATUS® Flasher ist ein PC-Tool für die Update-Programmierung eines Steuergerätes.

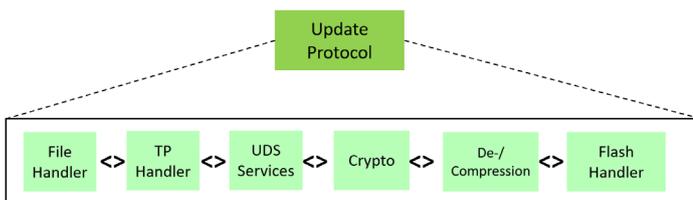
Die Programmierung erfolgt über ein Bussystem wie z.B. CAN, SPI oder UART und unterstützt u. a. Funktionen wie Verschlüsselung, Datenkompression und Signatur-Prüfung.

## Kurzbeschreibung

Der ADLATUS® Flasher ermöglicht die Programmierung eines Steuergerätes mit dem ADLATUS® Flashloader. Die Programmierung kann durch händische Auswahl der zur programmierenden Firmware über die grafische Bedienoberfläche erfolgen. Alternativ kann der Programmiervorgang über eine Batch-Datei mit der ADLATUS® Flasher DLL, zum Beispiel in einem Produktionsrechner erfolgen („Programmierstation“).

## Aufbau

Der ADLATUS Flasher arbeitet in einer Reihe von Verarbeitungsstufen. Die folgende Darstellung verdeutlicht die Verarbeitungsschritte:



Die Firmware wird von einer „Datenquelle“ zur Verfügung gestellt. Im Update-Protokoll (Flashprotokoll) ist der vom OEM vorgeschriebene Programmierablauf gefasst. Der ADLATUS® Flasher überträgt die (ggf. verschlüsselte) Firmware über das ausgewählte Transportprotokoll an das Steuergerät. Der „FileReader“ liest die zu programmierende Software im "Motorola S19" oder "Intel Hex-Format", ermittelt ihre Prüfsumme und überträgt sie komprimiert und verschlüsselt an den ADLATUS® Flashloader in der Elektronik. Die Funktionen der UDS-Service „Handler“ können leicht an kundenspezifische Programmieranforderungen angepasst werden.

Der CAN-TP Handler realisiert das CAN Transportprotokoll nach ISO15765-2 und wickelt dieses über einen CAN Device-Handler ab.

Der ADLATUS® Flasher arbeitet hoch konfigurierbar:

- Vorverarbeitung der Firmware, z. B. Einlesen verschiedener Formate, Berechnung der Firmware-Checksumme mit verschiedenen Algorithmen, Verschlüsselung und Kompression.
- Abarbeitung des Update-Protokolls (Flashprotokoll) mit Funktionen wie Autorisierung, Authentifizierung, Download etc.
- Kommunikation mit dem Steuergerät
- Der ADLATUS® Flasher protokolliert den gesamten Programmiervorgang in Form einer Log-Datei, die über Standwerkzeuge ausgewertet werden kann.

## Nutzen

- Ein Programmierwerkzeug für die Entwicklung, die Produktion und den Service (PC-Lizenz USB-Dongle)
- Vertrauliche Produktions-Programmierung mit verschlüsselter Firmware.
- Log-Datei des Programmiervorgangs





# FlashCedere®

Conformance Tester für das Kommunikationsverhalten  
von Automotive und Industrie Steuergeräten

Qualitätssicherung für alle  
ADLATUS® Software Produkte

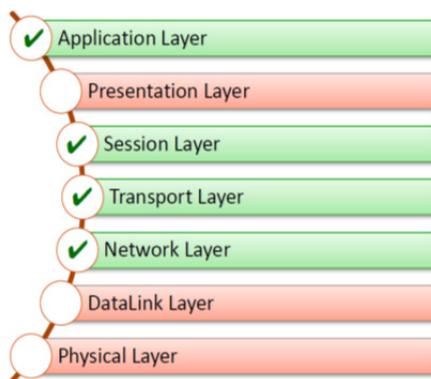


## Kurzbeschreibung

FlashCedere® ist eine professionelle Testlösung zum automatisierten Test der Basis-, Diagnose- und Update-Kommunikation von Automotive und Industrie Steuergeräten.

Der Testplatz besteht aus einem PC mit Kommunikationsinterface und FlashCedere®. FlashCedere® testet die Kommunikation und Funktion der Update-Programmierung von Steuergeräten automatisiert.

FlashCedere® testet folgende Kommunikations-Ebenen:



## Eigenschaften

Der FlashCedere® Conformance Tester ermöglicht die automatisierte Prüfung und Freigabe von Diagnose-, Flash- und Flashloader Software:

- Perfekte Reporterstellung, Nachweis der OEM-konformen Programmierbarkeit und Diagnosekommunikation von Steuergeräten.
- Aufzeichnung des gesamten Kommunikationsverlaufs, Inhalte inkl. Message-Trace mit Zeitstempeln.
- Report und Trace als HTML- und ASCII-File.
- Entwicklung von generischen Testfällen:  
FlashCedere Testfälle werden erst zur Laufzeit mit den Steuergeräte-spezifischen Parametern versorgt!

Einfache Verwaltung von Testfällen und "Testbundles". Die Steuergeräte-spezifische Parametrierung der Testfälle erfolgt über eine einfache Bedienung.

FlashCedere ermöglicht die automatische Bedienung aus FIBEX- oder ODX-Dateien.

Flexibler Export- und Import von Testergebnissen für die Sicherung und Wiederherstellung von Arbeitsergebnissen. Einfache Einbindung von Security-DLLs.

FlashCedere kann auf FlexRay beide „Coldstart Nodes“ simulieren um passive Steuergeräte testen zu können. CAN/CAN-FD bzw. FlexRay Produktlizenz für das Tool (USB Lizenz-Dongle). Produktlizenz für verschiedene Testbibliotheken (USB Lizenz-Dongle).

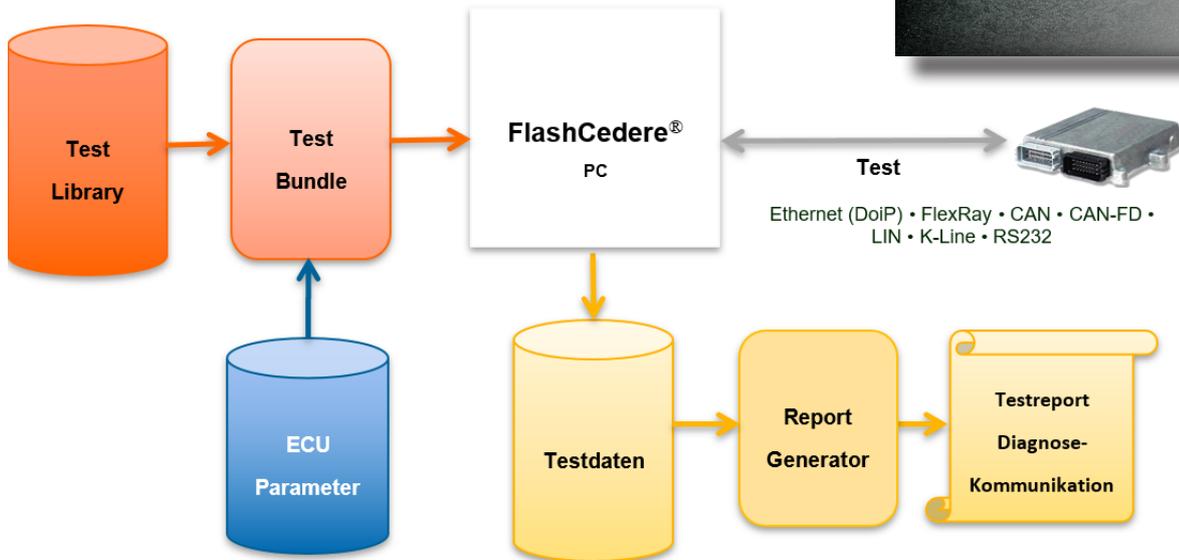
## Nutzen

- Automatisierter Flashloader Softwaretest für ASPICE Capability Level 2.
- Test der Requirement über Verifikationskriterien, Herstellung der Traceability zwischen Requirement, Verifikationskriterien und Testergebnis.
- Ein generisches Testbundle für alle Steuergeräte-Varianten, die der "UDS Update-Protokoll" Spezifikation genügen müssen.
- Nach Spezifikation zertifizierte Testbibliotheken als "Quality Gate" für Zulieferer.
- Das OEM-Test-Knowhow kann verschlüsselt werden im FlashCedere® TestRunner (Test only).
- In der Entwicklung wird durch automatisierten Test sichergestellt, dass die Diagnosekommunikation und die Update-Programmierung bei Lieferung gemäß OEM UDS-Spezifikation funktioniert.
- Stabile Testumgebung.



# FlashCedere®

Conformance Tester für das Kommunikationsverhalten von Automotive und Industrie Steuergeräten



## Testreport und Trace der Kommunikation bei der Update-Programmierung

Details:

V	ID	Data	Note
S	Tx	88 88 00 00 00 00 00 00	None
		00 00 00 00 00 00 00 00	
		00 00 00 00 00 00 00 00	
		00 00 00 00 00 00 00 00	
		00 00 00 00 00 00 00 00	
		00 00 00 00 00 00 00 00	
		00 00 00 00 00 00 00 00	

V	ID	Data	Note
Tx	7e5	31 01 02 03	RoutineControl CheckProgrammingPreConditions
		71 01 02 03	Expected
Rx	71	01 02 03	Pos. Response: RoutineControl CheckProgrammingPreConditions
Rx	700	85 82 ff ff	DTC Settings OFF
Rx	700	28 81 01	Communication control - Enable RX, Disable TX
			TesterPresent RBS started

Address	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18	Byte 19	Byte 20
6012.564	0x700	8	Tx	02	10	03	55	55	55	55	55	55	55	55	55	55	55	55	55	55
6014.044	0x7ed	8	Rx	06	50	03	00	32	01	F4	AA									
6023.224	0x7e5	8	Tx	04	31	01	02	03	55	55	55	55	55	55	55	55	55	55	55	55
6024.744	0x7ed	8	Rx	04	71	01	02	03	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA
6081.424	0x700	8	Tx	05	85	82	ff	ff	55	55	55	55	55	55	55	55	55	55	55	55
6135.174	0x700	8	Tx	03	28	81	01	55	55	55	55	55	55	55	55	55	55	55	55	55
6857.424	0x700	8	Tx	02	3e	80	aa	aa	aa	aa	aa	aa	aa	aa	aa	aa	aa	aa	aa	aa

Start prepare Testcase Details:

Consistency check. Details:

Test if bootloader may be accessed and wait if timelock is present Det

Step Sequence until Programming Session Details:

PreProgramming Sequence Details:

Diagnostic Session Control - Extended Session Details:

7000 10 03 Session Control - Extended Session

7000 04 Send delete error code

ECU did not answer with Pos. Response to delete error code: 50 03 00 32 01 F4

ECU answered with Pos. Response: Session Control - Extended Session: 50 03 00 32 01 F4

P2-Timing for this session is 50 ms

P2\*-Timing for this session is 5000 ms

PreConditions, DTC Settings and Communication Control Details:

Diagnostic Session Control - Programming Session Details:

Create a RBS with TesterPresent Details:

Read logistic data before continue Details:

Test the bootloader with the used CRCs in the downloadfiles Details:

Testing RecordDataIdentifier 0xF1 D4 Details:

6000 22 F1 D4 ServiceRequest: NotSupportedID

7F 22 31 Expected

4012 7F 22 31 Negative Response for NotSupportedID

Testing RecordDataIdentifier 0xF1 D5 Details:

6000 22 F1 D5 ServiceRequest: NotSupportedID

7F 22 31 Expected

4012 62 F1 D5 00 00 00 10 00 MessageMismatch: mismatching message payload

### PRODUKTMERKMALE

System	Features
Bussysteme	<ul style="list-style-type: none"> <li>FlexRay</li> <li>CAN</li> <li>LIN</li> <li>K-Line</li> <li>RS232</li> </ul>
FlexRay Interfaces	<ul style="list-style-type: none"> <li>Vector</li> <li>VN3600</li> <li>VN7600</li> <li>VN89xx</li> </ul>
CAN Interfaces	<ul style="list-style-type: none"> <li>Vector</li> <li>VN1600</li> <li>VN7600</li> <li>VN89xx</li> <li>VN5610</li> <li>CANcardXL(e)</li> <li>CANcaseXL</li> </ul>
System	<ul style="list-style-type: none"> <li>USB-CAN-modul/12 National Instruments</li> <li>NI-CAN Series 2</li> </ul>
LIN Interfaces	<ul style="list-style-type: none"> <li>Vector</li> <li>VN1600</li> <li>VN89xx</li> </ul>
K-Line Interfaces	<ul style="list-style-type: none"> <li>sartec</li> <li>HS+ Interface</li> </ul>
Protokolle	<ul style="list-style-type: none"> <li>ISO 14229</li> <li>ISO 14230</li> <li>ISO 15765-2</li> <li>ISO 10681-2</li> <li>LIN Protocol V2.1</li> </ul>
ECU Beschreibung	<ul style="list-style-type: none"> <li>FIBEX-File V2.0</li> <li>FIBEX-File V3.1</li> <li>AUTOSAR System Description V3.1 &amp; V3.2</li> </ul>
Systemvoraussetzungen	<ul style="list-style-type: none"> <li>Windows 7 32-Bit</li> <li>Windows 7 64-Bit</li> </ul>
Betriebssysteme	<ul style="list-style-type: none"> <li>Windows 7 32-Bit</li> <li>Windows 7 64-Bit</li> </ul>
CPU	Min. Dual Core 2.0 GHz
RAM	Min. 2 GByte

