



IIoT Libraries SL

Das Softwarepaket ‚IIoT Libraries SL‘ enthält Bibliotheken zum Thema ‚IIoT Kommunikation‘ und Bibliotheken zum Lesen und Schreiben von Datenstrukturen.

Produktbeschreibung

Lizenzierung:

Einzelgerätelizenz



Das Produkt ‚IIoT Libraries SL‘ enthält zahlreiche Bibliotheken zur Unterstützung unterschiedlicher Kommunikationsprotokolle (wie z.B. MQTT, HTTPS..) und enthält Werkzeuge zum Kodieren und Dekodieren von Datenstrukturen (wie z.B. JSON, XML..). Außerdem werden Bibliotheken mit Bausteinen zur direkten Kommunikation mit Cloud-Diensten einzelner Anbieter(AWS, Azure,...) bereitgestellt. Die im CODESYS Package enthaltenen Komponenten eignen sich hervorragend zur Kommunikation im IIoT Umfeld.

Bibliotheken zur Kommunikation

Web Client SL

Das CODESYS Software-Package enthält eine Bibliothek mit Funktionsbausteinen zur Kommunikation mit einem Webserver via http bzw. https.

- Mit http-GET können Daten von einem Webserver abgefragt werden.
- Mit http-POST und http-PUT können Daten von der Steuerung zu einem Webserver übertragen werden.
- Mit http-DELETE können Daten auf einem Webserver gelöscht werden.
- Mit http-HEAD kann der http-Header abgefragt werden.

Der http-Header einer Anfrage kann dabei beliebig erweitert werden. Zusätzlich sind Funktionsbausteine zur http-Basic-Authentifizierung und zur Authentifizierung über OAuth (Version 1a und 2) enthalten.

MQTT Client SL

MQTT (Message Queuing Telemetry Transport) ist ein offenes IoT Protokoll, das die Übertragung von Telemetriedaten in Form von Nachrichten zwischen Geräten ermöglicht. Die Kommunikation der Geräte erfolgt dabei immer über einen MQTT Broker (z.B. Mosquitto <https://mosquitto.org/>). Das Abonnieren und Verschicken von Nachrichten erfolgt auf Basis von ‚Topics‘. Ein ‚Topic‘ entspricht einem Pfad (z.B. device1/temperature). Das Abonnieren von Nachrichten erfolgt unter Angabe eines Topic-Filters, Wildcards sind hierbei auch erlaubt (+ für eine Ebene bzw. # für mehrere Ebenen) Das Format einer Nachricht ist nicht festgelegt, d.h. es kann ein JSON-String oder eine beliebige Datenstruktur übertragen werden.

Mit der Bibliothek ‚MQTT Client SL‘ können Nachrichten von einer CODESYS Steuerung an einen MQTT Broker geschickt werden und Nachrichten auf Basis von ‚Topics‘ abonniert werden.

Mail Service SL

Die Bibliothek enthält Funktionsbausteine zum Senden, Empfangen und Löschen von Emails über das SMTP- und POP3-Protokoll. Die Kommunikation mit dem Mailserver kann dabei verschlüsselt (TLS) oder unverschlüsselt erfolgen. Ein Beispielprojekt veranschaulicht die Verwendung der Funktionsbausteine.

SMS Service SL

Diese Bibliothek ermöglicht das Senden, Empfangen und Löschen einer SMS über ein GSM Modem. Zur Veranschaulichung wird ein Beispielprojekt mitgeliefert.

SNMP Service SL

Mit der SNMP Service SL Library können Informationen von Geräten, wie Drucker, Router usw. via SNMP von der Steuerung aus abgefragt werden. Die Bibliothek stellt Funktionsbausteine zum Versenden und Empfangen von SNMP Nachrichten bereit. Die Verwendung der Bibliothek wird anhand von mehreren Beispielen demonstriert.

Die Bibliothek unterstützt folgende SNMP Funktionen:

- SNMP GET: Abfrage eines Wertes (Steuerung ist Manager)
- SNMP GET_NEXT: Abfrage von Tabellen (Steuerung ist Manager)
- SNMP Agent: Abfrage von Werten der Steuerung (Steuerung ist Agent)
- SNMP TRAP: Senden und Empfangen von TRAP/INFORM-Telegrammen
- SNMP SET: Setzen von Werten

Das Package `SNMP Service SL.package` enthält folgende Komponenten:

- `SNMPServiceSL` (Namespace: `SNMP`)
- Projekt mit Beispielapplikationen
- CHM Hilfedatei
- Produktdatenblatt

SNTP Service SL

Die Bibliothek `SNTP Service SL` enthält Funktionsbausteine zum Aufbau von `SNTP Client` - und Serverapplikationen (`SNTP V3`, `SNTP V4`). Mit dem Funktionsbaustein `SNTPGetUTCtime` kann die UTC-Zeit von einem `NTP/SNTP`-Server abgefragt werden. Der Funktionsbaustein `SNTPServer` dient als einfacher `SNTP`-Server auf einer `CODESYS` Steuerung.

Folgende Funktionsbausteine (FBs) sind in der Bibliothek enthalten:

- `SNTPGetUTCtime`: FB zum Abfragen der Uhrzeit via `SNTP` (`SNTP Client`)
- `SNTPServer`: FB zum Verteilen der lokalen Uhrzeit via `SNTP` (`SNTP Server`)

AWS IoT Core Client SL

„AWS IoT Core“ ist eine verwaltende Cloud-Plattform von Amazon, mit der verbundene Geräte einfach und sicher mit Cloud-Anwendungen und anderen Geräten zusammenarbeiten können. Die Bibliothek „AWS IoT Core Client SL“ stellt Funktionsbausteine zum Senden und Empfangen von Nachrichten zur Verfügung. Die Kommunikation erfolgt dabei verschlüsselt über das Protokoll `MQTT`. Nachrichten werden im `AWS` Umfeld typischerweise im `JSON` Format übertragen. Zum Parsen und Erzeugen von `JSON` Dateien kann die Bibliothek „`JSON Utilities`“ verwendet werden.

Azure IoT Hub Client SL

Der Cloud-Dienst „`Azure IOT Hub`“ von Microsoft ist ein Service zur direkten Anbindung von `IoT`-Geräten (weitere Informationen siehe <https://azure.microsoft.com/de-de/services/iot-hub/>). Die Bibliothek „`Azure IoT Hub Client SL`“ stellt Funktionsbausteine zum Senden und Empfangen von Nachrichten zur Verfügung. Ein Beispielprojekt veranschaulicht die Verwendung der Bibliothek.

Google Cloud IoT Core Client SL

Der Cloud-Dienst „`Google Cloud IoT Core`“ von Google ist ein Service zur direkten Anbindung von `IoT`-Geräten (weitere Informationen siehe <https://cloud.google.com/iot-core?hl>).

Die Bibliothek „`Google Cloud IoT Core Client SL`“ stellt Funktionsbausteine zum Senden und Empfangen von Nachrichten zur Verfügung. Ein Beispielprojekt veranschaulicht die Verwendung der Bibliothek.

JSON Web Token SL

Die Bibliothek „`JSON Web Token SL`“ enthält einen Funktionsbaustein zum Erstellen von `JWT` (`JSON Web Token`) auf der Steuerung. Dabei werden die Algorithmen `HS256`, `HS384`, `HS512` und `RS256` unterstützt.

Das Package „`JSON Web Token SL`“ enthält folgende Komponenten:

- `IEC 61131-3` Bibliothek `JSON Web Token SL` (Namespace `JWT`)
- Beispielprojekt `JSON Web Token SL Example.project`
- Programmierreferenz als integrierte `CHM`-Hilfe
- Produktdatenblatt (`de` und `en`)

Web Socket Client SL

Das WebSocket-Protokoll ermöglicht eine bidirektionale Kommunikation zwischen einem Client und einem Web Socket Server über das Internet. Die Kommunikation erfolgt dabei in der Regel über die TCP-Portnummer 443 (oder 80 bei ungesicherten Verbindungen). Die Bibliothek ‚Web Socket Client SL‘ stellt Funktionsbausteine zur Kommunikation über das WebSocket-Protokoll zur Verfügung.

Bibliotheken zum Lesen und Schreiben von Datenstrukturen

CSV Utility SL

Die Bibliothek ‚CSV Utility SL‘ stellt Funktionsbausteine zum Lesen und Schreiben von CSV Dateien zur Verfügung. Die Bibliothek besteht aus den Komponenten `CSVReader` (Lesen von CSV Dateien) und `CSVWriter` (Schreiben von CSV Dateien). Sämtliche Funktionsbausteine können klassisch oder objektorientiert verwendet werden. In einem Beispielprojekt wird die Verwendung der Bibliothek ausführlich demonstriert.

INI File Utility SL

Das Produkt ‚INI File Utility SL‘ ist eine Bibliothek zum Lesen und Schreiben von INI-Dateien. Die Bibliothek enthält Funktionsbausteine zum Lesen, Schreiben und Suchen von Werten in INI-Dateien. Die Verwendung der Funktionsbausteine wird in einem Beispielprojekt demonstriert.

Inhalt des Packages:

- Bibliothek `INI File Utility SL` (Namespace: `INI_UTIL`)
- Beispielprojekt mit Visualisierung zum Lesen, Schreiben und Suchen von Werten
- CHM-Hilfedatei

JSON Utilities SL

Die Bibliothek ‚JSON Utilities SL‘ bietet folgende Funktionalitäten:

- Erstellen von komplexen JSON Dateien
- Lesen von JSON Dateien (UTF-8 und UTF-16)
- Lesen von JSON Bytearrays
- Schreiben von JSON Dateien (UTF-8 und UTF-16)
- Schreiben von JSON Bytearrays
- Suche nach Schlüssel, Werten, Kind-Elementen und Eltern-Elementen

Die Daten werden in einem Array von Strukturen gespeichert. Der Datenzugriff und die Suche nach Elementen erfolgt über Methoden des Funktionsbausteins `JSONData`. Das einfache erstellen einer JSON-Datei wird über den Funktionsbaustein `JSONBuilder` und seine Methoden realisiert. Die Funktionsbausteine `JSONByteArrayReader` und `JSONFileReader` lesen Daten im JSON-Format ein und speichern die Werte im Funktionsbaustein `JSONData`. Das Schreiben von Dateien und Bytearrays erfolgt über die Funktionsbausteine `JSONFileWriter` und `JSONByteArrayWriter`.

In Verbindung mit der Bibliothek ‚Web Client SL‘ (im Package enthalten) kann auf einfache Weise ein einfacher REST-Client implementiert werden.

XML Utility SL

Die Bibliothek ‚XML Utility SL‘ enthält Funktionsbausteine zum Lesen und Schreiben von XML-Dateien bzw. XML-Strings auf der Steuerung. Die einzelnen Elemente werden dabei in einem Array von Strukturen gespeichert. Darüber hinaus sind in der Bibliothek auch Funktionsbausteine zum Suchen von XML-Elementen enthalten. Die Verwendung der Funktionsbausteine wird anhand von zwei Beispielapplikationen demonstriert.

Das Package ‚XML Utility SL‘ enthält folgende Komponenten:

- IEC 61131-3 Bibliothek `XML utility SL` (Namespace `XML`)
- Beispielprojekt `XML Utility SL Example.project`
- Programmierreferenz als integrierte CHM-Hilfe
- Produktdatenblatt (de und en)

Die Produktdatenblätter der einzelnen Bibliotheken werden bei der Installation des CODESYS Packages in das Zielverzeichnis ‚IoT Libraries SL‘ installiert.

Allgemeine Informationen

Lieferant:

CODESYS GmbH
 Memminger Straße 151
 87439 Kempten
 Deutschland

Support:

<https://support.codesys.com>

Artikelname:

IIoT Libraries SL

Artikelnummer:

Vertrieb:


CODESYS Store

<https://store.codesys.com>

Lieferumfang:

- CODESYS Package mit Bibliotheken und Beispielprojekten zum Thema IIoT
- SL Lizenz

Systemvoraussetzungen und Einschränkungen

Programmiersystem	CODESYS Development System V3.5.16.10
Laufzeitsystem	CODESYS Control V3.5.15.0
Unterstützte Plattformen/ Geräte	Hinweis: Verwenden Sie das Projekt Device Reader, um die von der Steuerung unterstützten Funktionen zu ermitteln.
Zusätzliche Anforderungen	<p>Die Bibliotheken Web Client SL, MQTT Client SL, Mail Service SL, AWS IoT Core Client SL, Azure IoT Hub Client SL und Google Cloud IoT Core Client SL benötigen u.a. die Bibliotheken SysSocket2 3.5.16.0, Net Base Services, CmpCrypto, CmpTls 3.5.15.0 und CmpX509Cert.</p> <p>Die Bibliotheken CSV Utility SL, INI File Utility SL, JSON Utilities SL und XML Utility SL benötigen u.a. die Bibliothek SysFile.</p> <p>Die Bibliothek SMS Service SL benötigt u.a. die Bibliothek SysCom.</p>
Einschränkungen	-
Lizenzierung	<div style="text-align: center;">  <p>DEVICE</p> </div> <p>Einzelgerätelizenz: Die Lizenz kann auf der Steuerung, auf dem das CODESYS Laufzeitsystem installiert ist, verwendet werden.</p> <p>Die Lizenzaktivierung erfolgt auf einem softwarebasierten Lizenz-Container (Soft-Container), der fest an die Steuerung gebunden ist. Alternativ kann die Lizenz auf einem CODESYS Key (USB-Dongle) hinterlegt werden. Durch Umstecken des CODESYS Keys kann die Lizenz auf einer anderen Steuerung genutzt werden.</p> <p>Hinweis: Ohne Lizenz läuft die Software im Demomodus für 30 Minuten. Danach ist ein manueller Neustart erforderlich.</p>
Erforderliches Zubehör	-

Bitte beachten Sie: Nicht alle CODESYS-Funktionen sind in allen Ländern verfügbar. Weitere Informationen zu diesen länderspezifischen Einschränkungen erhalten Sie unter sales@codesys.com.

Bitte beachten Sie: Technische Änderungen, Druckfehler und Irrtümer vorbehalten. Es gilt der Inhalt der aktuellen Online-Version dieses Dokuments.