

Plattform MX6

Softwareoption S110

CODESYS IOT Libraries

1 Identifikation

Identifikation	
Optionskennung	S110
Bestellnummer	S-05000317-0000
Kurzbezeichnung	CODESYS IOT Libraries
Kurzbeschreibung	Diese Softwareoption enthält zahlreiche Bibliotheken zur Unterstützung unterschiedlicher Kommunikationsprotokolle für IIOT und enthält Werkzeuge zum Kodieren und Dekodieren von Datenstrukturen
Revisionskennung Dokument	V1.0

2 Systemvoraussetzungen und Einschränkungen

Systemvoraussetzungen und Einschränkungen	
Unterstützte Plattformen oder Geräte	Berghof SPS Geräte der MX6 Plattform (z.B.: MCs, CCs, DCs). Weitere Informationen bezüglich Verfügbarkeit und Kompatibilität finden Sie im Produktkatalog im Abschnitt Optionen.
Firmware	MX6-SPS ab Version 1.23.1, CODESYS ab 3.5 SP15 Patch 4
Weitere Anforderungen	— IP Netzwerkschnittstelle mit Internetzugriff
Einschränkungen	—

3 Produktbeschreibung

IIoT Libraries

Das Produkt ‚IIoT Libraries‘ 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.

3.1 Bibliotheken zur Kommunikation

3.1.1 Web Client

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.

3.1.2 MQTT Client

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.

3.1.3 Mail Service

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.

3.1.4 SMS Service

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

3.1.5 SNMP Service

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

3.1.6 SNTP Service

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)

3.1.7 AWS IoT Core Client

„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.

3.1.8 Azure IoT Hub Client

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.

3.1.9 Google Cloud IoT Core Client

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.

3.1.10 JSON Web Token

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)
- Beispielproject JSON Web Token SL Example.project
- Programmierreferenz als integrierte CHM-Hilfe
- Produktdatenblatt (de und en)

3.2 Bibliotheken zum Lesen und Schreiben von Datenstrukturen

3.2.11 CSV Utility

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.

3.2.12 INI File Utility

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

3.2.13 JSON Utilities

Die Bibliothek ‚JSON Utilities SL‘ bietet folgende Funktionalitäten:

- 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. 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.

3.2.14 XML Utility

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)

Ihre Ansprechpartner erreichen Sie unter:

Vertriebsteam | T +49.7121.894-131 | controls@berghof.com

Berghof Automation GmbH | Arbachtalstrasse 26 | 72800 Eningen | www.berghof.com
SoftwareOption S110 CODESYS IIOT Libraries V1 DE.docx