



Bilder: IBHsoftec

Zum Zugriff auf externe OPC-Server muss zuerst in der Client-Funktion ein externer Server eingerichtet werden

Mit Python komplexe IoT-Aufgaben einfach bewältigen

Dank der vielseitigen Programmiersprache Python ist es nun möglich, flexibel auf Maschinen- und Steuerungsdaten über OPC UA zuzugreifen. Dies wird durch die OPC-UA-Server/-Client-Baugruppe namens IBH Link UA ermöglicht. Die Verbindung dieser Baugruppe mit Python eröffnet eine flexible und skalierbare Plattform zur Datenverarbeitung. Wir haben darüber mit dem CTO Axel Hulsch und dem Senior Software Engineer Michael Heilmann von IBHsoftec gesprochen.

Text: Ronald Heinze

Um IoT-Anwendungen in der Produktion zu realisieren, ist der Zugriff auf Daten aus der OT-Ebene unerlässlich. Früher bestand der Ansatz darin, eine separate Infrastruktur aufzubauen, um die vorhandenen Steuerungsstrukturen in Brownfield-Anwendungen nicht zu gefährden. Doch mittlerweile hat sich die Erkenntnis durchgesetzt, dass es einfacher ist, Prozessdaten direkt aus der Steuerung auszulesen.

SPS

➔ IBHsoftec: Halle 5, Stand 140

Der IBH Link UA von IBHsoftec [1] ist eine etablierte OPC-UA-Server/-Client-Baugruppe mit integrierter Firewall, die genau für solche Aufgaben entwickelt wurde. Es handelt sich um ein kompaktes Gerät zur Hutschienmontage mit einer DC-24-V-Stromversorgung. „Unsere Baugruppe kann als zentraler Server Daten von verschiedenen Steuerungen verarbeiten“, betont A. Hulsch. Dadurch können Daten von unterschiedlichen Geräten und Systemen zusammengeführt werden, beispielsweise für Analyseaufgaben.

Durch die Integration von Python-Modulen wird die Flexibilität weiter gesteigert. Dies ermöglicht die Umsetzung



02 Axel Hulsch ist CTO bei IBHsofttec



03 Michael Heilmann ist Senior Software Engineer bei IBHsofttec

neuer Anwendungen auf individuelle und flexible Weise. Es eröffnen sich neue Möglichkeiten für die Analyse und die Automatisierung, indem Datenmodelle bzw. Companion Specs (Nodesets) eingelesen und mit Python-Modulen verknüpft werden können. Python ist eine weit verbreitete, normalerweise interpretierte Hochsprache, die für ihren gut lesbaren und prägnanten Programmierstil bekannt ist. „Python gewinnt immer mehr Anhänger, und viele Software-Entwickler setzen auf Python“, wie A. Hulsch ergänzt. „Man könnte sagen, Python ist der Nachfolger von Basic.“

Dank der Baugruppe IBH Link UA besteht nun die Möglichkeit, Datenmodelle bzw. Companion Specs in Form von Nodesets einzulesen und mit Python-Modulen zu verknüpfen. Zu beachten ist jedoch, so merkt A. Hulsch an, dass „diese Option nur im IBH Link UA mit Quadcore-Prozessor verfügbar ist, da hierfür mehr Speicher benötigt wird“. Der auf Linux basierende Softwarestack von IBHsofttec kann auch direkt auf den Mitsubishi-Electric-Steuerungen der Serie iQ-R (RD55) ausgeführt werden. „Auch hier wird die Integration von Python-Modulen möglich sein“, hebt A. Hulsch hervor.

Komplexe Aufgaben leicht automatisieren

Aber wozu ist das überhaupt nötig? Nun, Python ermöglicht die Automatisierung von komplexen Aufgaben, sei es die Überwachung von Maschinenparametern oder die Optimierung von Produktionsprozessen. Mithilfe dieser Programmiersprache können große Datenmengen analysiert und ausgewertet werden, um Trends und Muster zu identifizieren und Prozesse zu verbessern.

Durch die Integration von Python-Modulen für maschinelles Lernen wird es möglich, Maschinenparameter und -daten zu analysieren, um Vorhersagen zu treffen und Empfehlungen zur Optimierung der Maschinenleistung auszusprechen. Es ist sogar möglich, Daten aus verschiedenen Quellen und Systemen zu integrieren, um ein umfassendes Bild von der Maschinenleistung und -produktivität zu erhalten.

Ein einfaches praktisches Beispiel, das bereits von IBHsofttec umgesetzt wurde, ist ein Analysegerät mit einer OPC-Schnittstelle. „Das Messsystem misst die Viskosität von Flüssigkeiten. Eine Messung wird über eine OPC-Methode als SPS-Aufruf ausgelöst“, erklärt der Manager von IBHsofttec. „Abhängig vom Ergebnis greift die SPS dann erneut in den Prozess ein, und wir haben einen geschlossenen Regelkreis.“

Neben solchen Aufgaben zur Überwachung und Optimierung der Produktion können auch riesige Datenmengen analysiert werden, beispielsweise zur Erkennung von Trends. Es besteht auch die Möglichkeit, Python-Module für maschinelles Lernen zu integrieren. „Oder die Daten können in die Azure Cloud über-

tragen werden, um sie dort mit KI-Tools zu analysieren“, ergänzt der CTO. „Selbstverständlich ist auch die einfache Implementierung eines MQTT-Protokolls mit Python möglich.“ A. Hulsch hebt hervor, dass eine umfangreiche Python-Bibliothek zur Verfügung steht.

Zugriff auf Datenbanken

„Ein weiteres Beispiel betrifft den Zugriff auf Datenbanken“, führt A. Hulsch fort. „In der Datenbank können Prozessdaten, Alarmer und Ereignisdaten verwaltet werden. SPS-Systeme können direkt mit externen SQL-Datenbanken verbunden werden, ohne in das Steuerungsprogramm eingreifen zu müssen. Python ermöglicht den Zugriff auf die Datenbanken, sodass die SPS über den IBH Link UA Daten an die SQL-Datenbank senden kann. „Mithilfe des Python-Programms wird beispielsweise ein bestimmtes Bit in der SPS gesetzt, um einen Datensatz in die Datenbank zu schreiben“, erklärt A. Hulsch. „Interessant ist dabei, dass Daten aus verschiedenen Steuerungen gesammelt werden können.“

Sobald die Daten in der SQL-Datenbank vorliegen, können sie problemlos anderen Anwendungen zugeführt werden. SQL-Funktionen lassen sich in Verbindung mit OPC-Variablen ausführen. A. Hulsch erläutert, dass der aktuelle Wert einer OPC-Variablen abgerufen und in eine SQL-Abfrage einbezogen werden kann, um andere Variablenwerte zu aktualisieren oder Daten in eine Datenbank zu schreiben. Außerdem ist der Zugriff auf externe OPC-Server möglich. Zusätzlich können im externen Server auch Methoden ohne Parameter aufgerufen werden. „Es ist möglich, eine Methode in einem anderen OPC-Server aus einer SPS heraus aufzurufen“, freut sich A. Hulsch.

Datenmodelle der Companion Specs nutzen

Die Verwendung der Companion Specs der OPC Foundation bietet eine weitere Möglichkeit. M. Heilmann weist darauf hin, dass diese Companion Specs, die mittlerweile für verschiedene Branchen spezifiziert wurden, nun bearbeitet werden können. „Mit den Companion Specs wurden Datenmodelle definiert, die die Präsentation von Maschinen und Anlagen nach oben hin beschreiben“, erläutert er. Für die Industrieautomatisierung gibt es Datenmodelle für die Steuerung von Produktionsanlagen und Maschinen, Überwachung und Diagnose sowie die Verwaltung von Produktionsprozessen und Energiemanagement. „Die entsprechenden Nodesets sind im Github der OPC Foundation verfügbar“, fügt A. Hulsch hinzu.

Nun kommt es darauf an, wie diese Datenmodelle mit Leben gefüllt werden. Normalerweise müssten Maschinen von Grund auf neu entwickelt werden, um diese Datenmodelle nutzen zu können. Doch jetzt besteht die Möglichkeit, Companion Specs mit Python-Modulen auf der Baugruppe von IBHsoftec zu verbinden. „Wir haben dies bereits erfolgreich mit PackML, der Maschinensprache für die Verpackungsindustrie, getestet“, informiert M. Heilmann.

Eine einfache Vorgehensweise

Um Datenmodelle mit Python-Modulen auf dem IBH Link zu verknüpfen, gehen wir folgendermaßen vor: Zunächst

wird das gewünschte Companion Spec oder Datenmodell in das Projekt auf dem IBH Link UA integriert. Hierfür kann das entsprechende XML-File über den Webserver des IBH Link UA hochgeladen werden. Die Datenmodelle werden mithilfe spezieller Nodeset-Tools erstellt und beschreiben die Schnittstelle, über die die Daten ausgetauscht werden sollen. Beispiele für solche Tools sind UAModeler von Unified Automation, ein kostenpflichtiges Werkzeug, oder Siome von Siemens, eine kostenlose Variante. Auch Wago stellt einen Nodeset-Editor zur Verfügung. „Diese Nodesets werden dann direkt mit Funktionen verbunden“, berichtet A. Hulsch. „Ein solcher Editor erzeugt ein Python-Programmgerüst für den IBH Link“, fügt M. Heilmann an. „Es handelt sich sozusagen um eine Programmierhilfe, da es oft um komplexe Datenmodelle geht.“

Sobald das Datenmodell auf dem IBH Link UA verfügbar ist, wird ein Python-Programmgerüst aus dem Modell erstellt. Als nächster Schritt erfolgt das Schreiben des Python-Codes, mit dem auf die Daten des Datenmodells zugegriffen werden soll. „Die Client-Funktion wird in Python programmiert“, betont A. Hulsch. Der Code ermöglicht den Zugriff auf die Daten des Modells über die Schnittstelle. Abschließend wird der Python-Code in den IBH Link UA geladen. Dadurch wird es möglich, Daten aus dem Datenmodell in Python zu verarbeiten, wobei verschiedene Python-Module genutzt werden können.

Auch für Brownfield-Anwendungen geeignet

Die beschriebenen Möglichkeiten sind für alle Steuerungen verfügbar, die mit der IBH-Link-UA-Baugruppe ausgestattet werden können. Das umfasst derzeit S7- und S5-Steuerungen, Siemens-CNC sowie Rockwell- und Mitsubishi-Electric-Steuerungen und -Roboter. Somit können auch ältere Maschinen und Anlagen im Brownfield-Bereich, für die Digitalisierung ertüchtigt werden. Wenn die Baugruppe als OPC-UA-Server fungiert, kann eine „alte“ S5 also Daten an die Datenbank senden und Kommandos auslösen.

Unzählige Automatisierungsgeräte können über ein Python-Projekt Daten lesend und schreibend austauschen. „Dies eröffnet völlig neue Möglichkeiten für die Kommunikation innerhalb der OT-Ebene sowie zwischen IT und OT in der Welt des Industrial Internet of Things (IIoT)“, schließt A. Hulsch ab. (hz)

Literatur

- [1] IBHsoftec Gesellschaft für Automatisierungstechnik mbH, Beerfelden: www.ibhsoftec.com