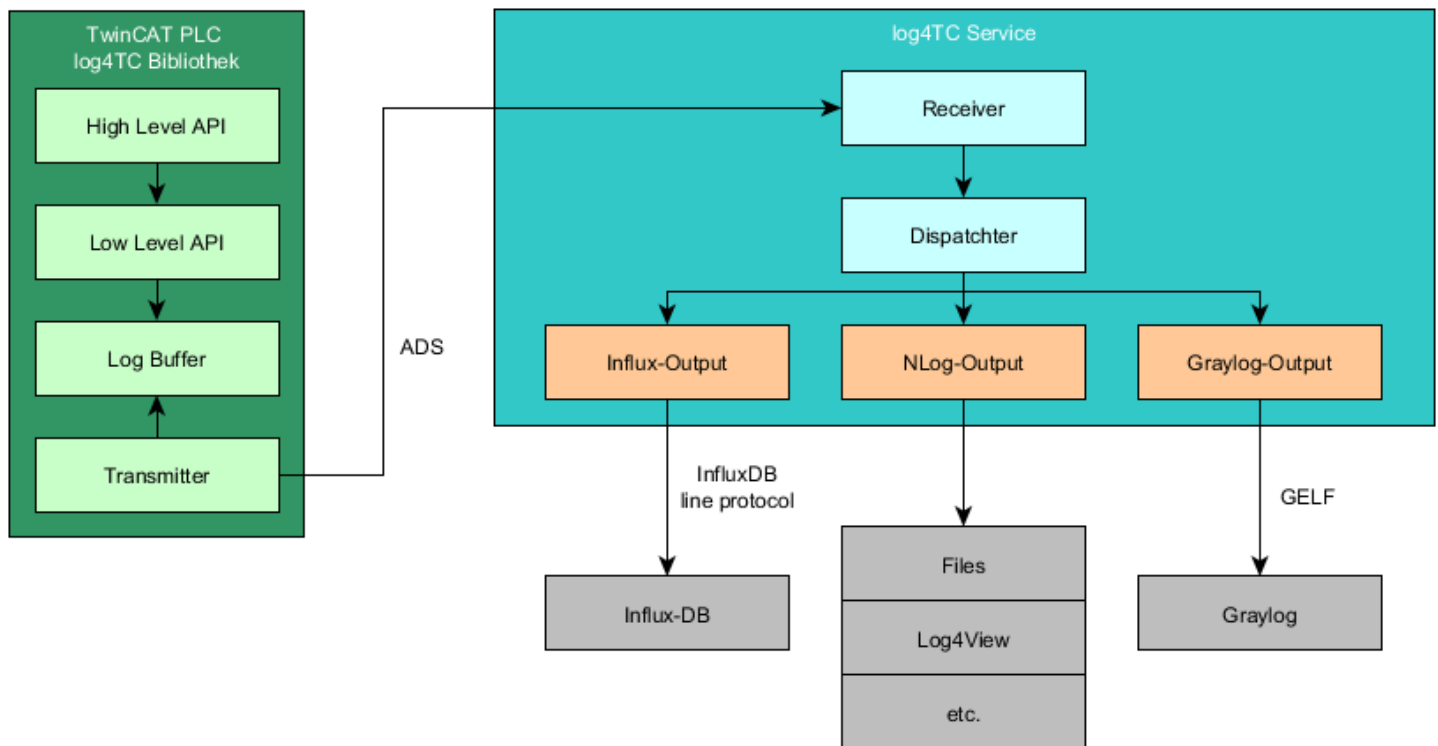


mbc log4TC

Log4TC ist eine Erweiterung für TwinCAT3 von Beckhoff, um direkt aus der SPS Logmeldungen erzeugen zu können. Die Meldungen können transferiert, gefiltert, ausgewertet und an verschiedene Ausgaben weitergeleitet werden.

Log4TC besteht aus zwei Teilen, einer SPS-Bibliothek und einem Windows-Service.

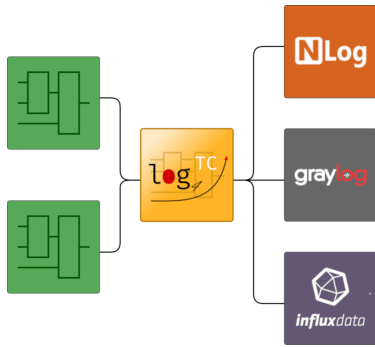


Der log4TC-Service wird normalerweise auf dem Rechner installiert, auf dem auch die SPS läuft, kann aber für bestimmte Einsatzzwecke auch auf einem anderen Rechner für mehrere Steuerung installiert werden.

Features

- Einfache API für die integration in die SPS
- Strukturiertes Logging (<https://message templates.org/>)
- Unterstützung von Context-Eigenschaften auf verschiedenen Ebenen
- Performant und Modular
- Kostenlose Testversion verfügbar
- Lizenzierung über Beckhoff-Mechanismus in Dongle, Klemme oder PC
- Unbegrenzte Ausgabemöglichkeiten (Textdatei, Datenbank, Cloud, usw.)

Ausgaben



Log4TC implementiert Ausgaben über ein Plugin-System. Standardmässig bei der Auslieferung ist die NLog-Ausgabe aktiv. Die Ausgabeplugins werden laufend erweitert, die nächsten Plugins sind Ausgaben für Graylog und InfluxDB.

Bei Bedarf können wir auch kundenspezifische Ausgaben erstellen.

Einführungsvideo

Log4TC - Setup und erste Schritte



Installation

Beim **Installieren** von log4TC wird eine Default-Konfigurationsdatei mit installiert und ist sofort zur Verwendung bereit. Folgen Sie der **Installationsanleitung**.

Typische Anwendungsfälle für log4TC

- Fehlertracking und Alarmierung
- Debugging von sporadischen Fehlern ohne Breakpoints
- Ablaufanalysen bei Problemen auch in Nachhinein
- Statistische Auswertungen, z.B. KPI

Nächste Schritte

Nächste Schritte

- [Download](#)[↗]
- [Erste Schritte](#)
- [Referenz](#)

Firewall

Der Log4TC Service erstellt einen eigenen ADS Server auf den die PLC sich verbindet. Der ADS Server läuft auf Port **16150**. Daher muss die Firewall entsprechend konfiguriert werden damit dieser Port erreichbar ist.

Table of Contents

Neues Einführungsvideo für log4TC	5
Neuer Release log4TC: 20.10.21	7
Neuer Release log4TC: 20.07.28	8

Neues Einführungsvideo für log4TC

Für den einfachen Einstieg in log4TC haben wir ein kurzes Einführungsvideo veröffentlicht. Es zeigt in unter vier Minuten wie man log4TC installiert, in ein SPS-Projekt einbindet und eine Log-Meldung ausgibt.

Log4TC - Setup und erste Schritte



Neuer Release log4TC: 20.10.21

Ausgabe der Log-Meldungen in SQL-Datenbanken

Dieses Release erweitert die log4TC Ausgaben um ein weiteres Plugin: Neu können Log-Meldungen in SQL-Datenbanken geschrieben werden. Log4TC wird dazu mit drei Treibern ausgeliefert: MySql/MariaDB, Postgres, MS-SQLServer. Für die Meldungen stehen zwei verschiedene Formate zur Verfügung.

Die Dokumentation der SQL-Ausgabe befindet sich [Hier](#).

Brauchen Sie Treiber für anderen Datenbanken? Wollen Sie ein anderes Schema? Zögern Sie nicht und fragen Sie uns!

Sie finden das log4TC Setup hier: [log4TC Release](#)

Fehlerbehebungen

Der Release beseitigt folgende Fehler:

- NLog-Ausgabe wird nicht mehr initialisiert, wenn die Ausgabe nicht konfiguriert wird.
- Fehler bei der Ausgabe werden jetzt in jeden Fall in das interne log4TC-Logfile geschrieben.

Sonstige Anpassungen

- Lizenzierung für Windows-CE möglich.

Neuer Release log4TC: 20.07.28

Neue Ausgaben für Log-Meldungen und Bugfixing

Dieses Release bringt die angekündigten Ausgaben für Influx und Graylog mit. Ausserdem kann der log4TC-Service jetzt über eine Datei konfiguriert werden.

Die neue Version steht auf [Hier](#) zum Download zur Verfügung.

Neue SPS-Library

Die neue Library hat die Versionsnummer 0.0.5. Neben diversen Fehlerbereinigungen wurde die SPS-API um einen neuen Baustein ergänzt: `FB_LoggerLAC`.

Mit diesen Baustein können Log-Meldungen in einem *FB* oder *PRG* abgesetzt werden, ohne das jedesmal der Loggername mitgegeben werden muss. Bei der Instanzierung hat man die Möglichkeit den Loggernamen automatisch ermitteln zu lassen (funktioniert nur bei *FBs* und *PRGs*) wenn man als Loggernamen die Konstante `Const.sLoggerFromInstance` angibt:

```
fbLog : FB_LoggerLAC(Const.sLoggerFromInstance);
```

Context-Eigenschaften können über die Eigenschaft `LoggerContext` einmalig hinzugefügt werden. Sie werden dann bei jeder Log-Messages dieses Instanz mitgeschrieben:

```
fbLog.LoggerContext.AddInt('orderno', 42);
```

Die eigentliche Ausgabe einer Meldung erfolgt über den direkten Aufruf:

```
fbLog(  
    eLogLevel := E_LogLevel.eInfo,  
    sMessage  := 'barbaz {bLog}',  
    aArg1     := bLog,  
);
```

Um eine bedingte Log-Meldung zu generieren, kann das Input-Argument `bExecute` benutzt werden. Nur wenn der Wert `TRUE` ist, wird eine Log-Meldung generiert.

```
fbLog(  
    bExecute := bError,  
    eLogLevel := E_LogLevel.eError,
```



```
sMessage := 'Fehler aufgetreten',  
);
```

Ausgabe nach Influx

[InfluxDB](#) ist eine Open-Source Datenbank, spezialisiert auf die Verwaltung von Zeitreihendaten. Kurz gesagt, lassen sich damit verschiedene Daten (Numerische Werte, Strings, etc.) mit einem Zeitstempel zeitlich geordnet speichern. Eine einfache Zeitreihe könnte z.B. eine Temperaturaufzeichnung sein:

- 09:00:00 -> 16°C
- 09:30:00 -> 18°C
- 11:00:00 -> 23°C

Die Abstände der Messung müssen dabei nicht regelmässig sein. Ebenso kann ein Zeitstempel auch mehrere Messwerte zugewiesen werden. Zusammengehörige Werte werden in einer Messung (*Measurement*) gespeichert.

Existieren mehrere Instanzen eines Messwerts - im Beispiel in etwas mehrere Sensoren, so können diese mit Tags gekennzeichnet werden. Ein umfangreiches Beispiel könnte damit so aussehen:

Klima:

- 09:00:00 -> Temperatur=16°C (Sensor=Aussen), Temperatur=21°C (Sensor=Innen)
- 09:30:00 -> Temperatur=18°C (Sensor=Aussen), Feuchtigkeit=30% (Sensor=Aussen)
- 11:00:00 -> Temperatur=23°C (Sensor=Innen)

Die gespeicherten Daten können mit einer SQL ähnlichen Abfragesprache selektiert werden. Um z.B. die maximale Aussentemperatur der letzten 24h abzufragen lautet die Abfrage:

```
SELECT MAX("Temperatur") FROM "Klima" WHERE "Sensor"="Aussen" AND timestamp >= now()  
- 24h
```

Mit der `influxdb`-Ausgabe können jetzt solche Daten direkt aus log4TC geschrieben werden. Wichtig dabei ist es, dass die zu schreibenden Daten als strukturierte Argumente in einer Log-Meldung vorliegen.

[Influx-DB Ausgabe](#)

Ausgabe nach Graylog

Für viele Anwendungen dürfte log4TC mit der NLog-Ausgabe, die lokale Dateien schreibt ausreichend sein. Sind aber mehrere Rechner mit TwinCAT im Einsatz wird das dezentrale Logging aber sehr schnell unübersichtlich und ineffizient bei der Überwachung und Fehlersuche. Für genau diesen Fall können jetzt Log-Meldungen an einen Graylog-Server zentralisiert weitergeleitet werden.

[Graylog](#) ist ein Log-Managementsystem, das Log-Meldungen - nicht nur von log4TC - empfängt, speichert und für Analysen bereitstellt. Durch die Indexierung können auch grosse Logdaten in kürzester Zeit durchsucht werden.

Da Graylog intern eine Log-Meldung strukturiert speichert, passt dieses System nahtlos zu log4TC.

[Graylog Ausgabe](#)