

.NET Programmierung für industriennahe Anwendungen

Wintersemester 2022

(c) 2022

Roland J. Graf / Gerald Lochner

Labor 02: C#/.NET Assemblies und ModBus

1.1. Themen

- Vertiefung zu C#/.NET Programmierung
- Assemblies: Grundlagen, Erstellung & Verwendung
- Einbindung externer Assemblies in eigene Projekte
- Modbus: Grundlagen, Anwendung

1.2. Projektaufgaben

Im zweiten Schritt lernen Sie .NET Assemblies, sowie deren Erstellung und Verwendung kennen. Zudem implementieren Sie einen Modbus-Server. Hierfür schreiben Sie wieder einige neue Klassen und eine neue Applikation.

1.2.1. Erstellung und Verwendung einer Assembly

- Erstellung eines eigenen Assemblies
 - ⇒ die Basisklassen (**Würfel**, **Würfelsatz**,...) aus Übung 1 sollte in eine Assembly überführt werden
- Verwendung des neu erstellten Assemblies in der TestApplikation aus Übung 1
 - ⇒ Entfernen der Basisklassen aus der App, stattdessen Einbindung des obigen Assemblies in die App

1.2.2. Erstellung eines Factory-Simulator mit Modbus-Anbindung

- Erstellung eines Factory-Simulators
 - Sie sollten in einem Simulator **mindestens 5 Maschinen** simulieren, die jeweils ein bestimmtes *Produkt/Bauteil* (z.B. Motor, Schraube, Kabelbaum, Schalter, Getriebe,...) herstellen.
 - Jede Maschine sollte **mindestens 5 Produktions-/Maschinenparameter bzw. Kenngrößen** liefern (z.B. die Anzahl der jeweils produzierten *Produkte/Bauteile*, die aktuelle Geschwindigkeit, die Anzahl der produzierten Bauteile pro Minute, Fehlerrate, Herstellungstoleranzen, Temperaturen, Wartungsintervalle, Status der Maschine [processing, failed, maintenance,...] usw. ...); sehen Sie hierfür auch *variable* bzw. *zeitabhängig veränderliche* Kenngrößen vor, da wir in späteren Einheiten mit diesen Daten arbeiten und

auch einen Verlauf abbilden wollen (z.B. einen Temperaturverlauf);

- Jede Maschinen sollte fortlaufend produzieren können. Jede Maschine sollte **mindestens 3 Operationen** anbieten, mit der die Maschine gesteuert werden kann (z.B. Start, Stop, Speed,...).
- Erweiterung des Simulators zu einem ModbusTCP Server
 - Einbindung der EasyModbusTCP Bibliothek
⇒ siehe easymodbus tcp.net, [Modbus Specifications](#)



Die Einbindung des EasyModbusTCP Assemblies erfolgt innerhalb von Visual Studio über NuGet; zusätzlich muss auch noch das Assembly System.IO.Port eingebunden werden, das von EasyModbusTCP benötigt wird.

- Erweitern Sie Ihren Factory-Simulator zu einem **ModbusTCP-Server**. Die obigen Produktionsparameter/-kenngrößen und Operationen sollten nun über Modbus-Objekte externen Programmen zur Verfügung gestellt werden. Getestet werden kann mit dem ModbusClient von der EasyModbusTCP Seite (siehe [EasyModbusTCP.NET](#) ⇒ Downloads).
- Lesen Sie nun über den Modbus Client die Daten Ihrer simulierten Maschinen aus und machen Sie die Operationen der Maschinen über Modbus aufrufbar. Am Ende sollten die Maschinen über Modbus ansteuerbar sein und bestimmte Daten lese-/schreibbar sein.

1.2.3. Allgemeine Anforderungen:

- Es gelten die gleichen Regeln wie für Aufgabe 01.
- Gibt es vielleicht Softwareteile bzw. Klassen/Typen, die man in späteren Projekten wiederverwenden könnte? Sollten dies der Fall sein, dann legen Sie hierfür eigene Assemblies und Namensräume an.
- Achten Sie vor allem bei wiederverwendbaren Klassen auf die Fehlertoleranz Ihrer Implementierung. Keinesfalls sollte eine Methode abstürzen. Fehler werden ausschließlich über Exceptions signalisiert. Alle Parameter müssen überprüft bzw. plausibilisiert werden.
- Testen und Debuggen Sie Ihre Klassen und Methoden umfänglich.

1.3. Projektabgaben

- Abgabebedingungen und Formate: siehe Moodle-Kurs