



SpiderControl™ ist eine Marke von



Pressemeldung  
März 2021

## SpiderPLC auf einem MicroController Grafische Programmierung & online Debugging über USB/Seriell

Benötigen Sie eine ganz einfache Programmiermöglichkeit auf einem Single-Chip MicroController - ab 128kB Flash, 80kB RAM? Und möchten das Programmierwerkzeug auf Ihre Bedürfnisse anpassen, also ein Customizing machen?

Die vernetzte Architektur von SpiderPLC ermöglicht es, dass lediglich die Laufzeit zur Ausführung des Codes auf den MicroController kompiliert wird, während der Server mit der Programmiersoftware auf einem abgesetzten Rechner läuft (distributed an Stelle von integriert). Dies kann entweder ein normaler PC oder auch ein spezifisches embedded System sein, welches als Programmiergerät genutzt wird.

Das Konzept der SpiderPLC besteht aus verschiedenen Komponenten eines Baukastensystems, welches sich in verschiedenen unterschiedlichen Konstellationen einsetzen lässt.

### Integrierte Lösung auf Web-Controller (Integrierte Variante)

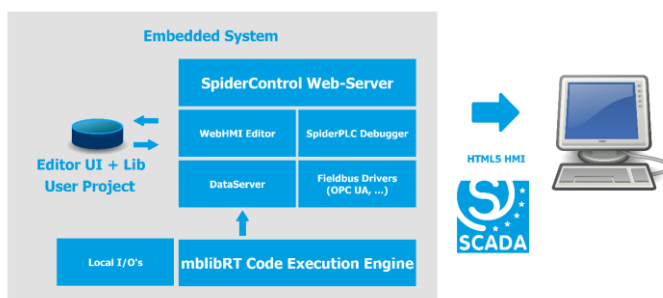


Bild 1

Das System basiert auf der SpiderControl HTML5 WebHMI Toolchain und besteht aus den Komponenten:

- Web-Server inkl. User Interfaces für die browserbasierte Fupla und HMI Editor Funktion
- Bibliothek der HMI und FB Objekte
- PC-basierter HMI Editor zum Design dieser Funktionen (nicht eingezeichnet)
- WebHMI Editor Business Logik zum Erzeugen und Speichern von HMI oder FB Objekten sowie zur Codegenerierung
- DataServer zur Verbindung zu den Datenpunkten lokal und ggf. über Feldbus Treiber
- Die Code Execution Engine mit Anbindung an die lokalen I/O's

Im obigen Bild ist eine Konfiguration gezeichnet, wie man sie auf ein embedded System mit Ethernet und hinreichend Ressourcen integrieren würde.



SpiderControl™ ist eine Marke von



## Distributed Web-Server für kleine uController

Für die Programmierung von kleinen MicroControllern, welche nicht über ein Ethernet Interface verfügen, kann man die einzelnen Module so aufteilen, dass man das klassische Konzept eines PC-basierten Programmiertools und einer separaten SPS umsetzen kann.

Die Komponenten des Web-Servers und der ganzen Programmierlogik werden in einer PC-Applikation integriert und lediglich die Code-Execution Engine (mblibRT) wird auf dem uController integriert. Zusätzlich wird eine Kommunikationsschicht, welche z.B. über eine serielle Verbindung integriert werden kann, verwendet. Dieser Layer übernimmt die folgenden Aufgaben:

- Deploy des ausführbaren Codes
- Datenkommunikation für die Kontrolle (Start, Stop...) und das Debugging der RT

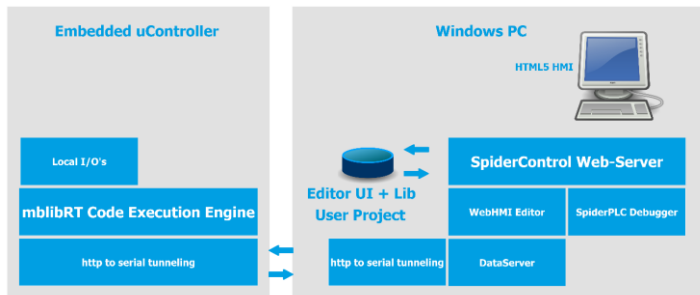


Bild 2

Das Tunneling bildet im Prinzip die relevanten http Funktionen (Web-Services) auf einen seriellen Transport Layer ab, sodass die bestehenden Funktionen, welche normalerweise im Web-Server drin aufgerufen werden, in analoger Weise angebunden sind.

## PC Applikation

Die PC-basierten Komponenten können in eine monolithische Applikation eingebunden werden, sodass ein einzelnes Exe gestartet wird, welches die weiteren Komponenten als DLL mit hochzieht und lokal auf das Filesystem zugreifen kann. Dieses EXE wird als Viewer ausgestaltet, d.h. besteht aus einem MicroBrowser oder einem HTML5 Client, sodass der Browser nicht separat gestartet werden muss.

Das Tool startet mit der bestehenden Webseite, welche alle zur Bedienung notwendigen Funktionen enthält. Zusätzlich wird noch eine Seite zur Auswahl des zu bearbeitenden Projektes zur Verfügung gestellt.

Die FUPLA Programmierung erfolgt lokal auf dem PC, über Build&Run wird der Code erzeugt und auf das Zielsystem übertragen. Zu diesem Zeitpunkt muss das Target über einen virtuellen COM-Port angeschlossen sein, damit die Code-Übertragung erfolgen kann. Das Zielsystem empfängt den Code und erzeugt daraus On-The-Fly das auf dem Target ausführbare Format, welches nach Ende des Downloads gestartet wird.

Die Debug Funktion auf dem HMI startet lokal die entsprechende View und beginnt über das Tunneling mit einer zyklischen Datenkommunikation, in welcher die aktuell angezeigten Variablen abgefragt und Schreib-Befehle übertragen werden.



SpiderControl™ ist eine Marke von



Bild 3

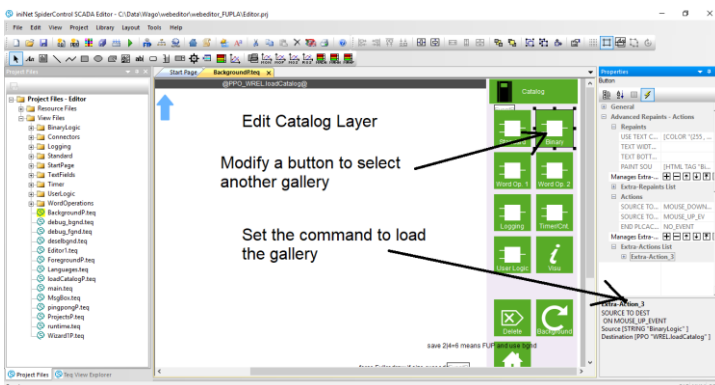
Das selbe Konzept kann auch auf ein dediziertes Programmiergerät angewendet werden, der einzige Unterschied besteht darin, dass der Client dann als regulärer Browser auf einem PC läuft, während die anderen Module - wie in Bild 2 gezeigt - auf einem als Programmiergerät ausgestatteten embedded System integriert sind.

## Baukastenkonzept OEM-Anpassungen leicht gemacht!

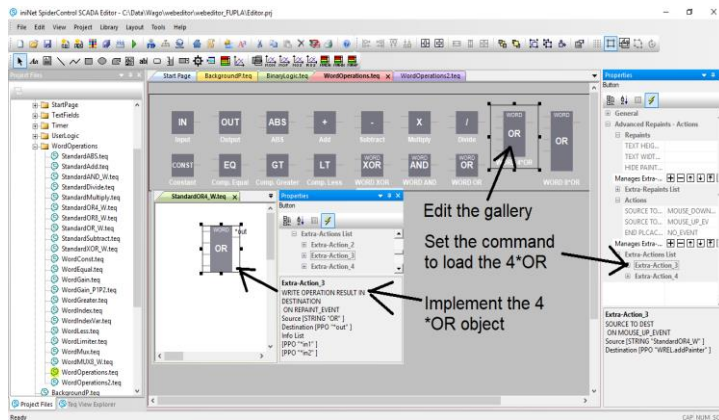
Der enorme Vorteil dieses Konzeptes besteht darin, dass alle Elemente des Web-HMI und FUPLA Editors, wie in einem Lego-Baukasten verwendet und kombiniert werden können.

Ein OEM-Kunde ist somit in der Lage, mit dem bestehenden PC-basierten SpiderControl Editor (PLC Edition) alle Eigenschaften und Funktionen des Web-HMI Editors zu programmieren. Das User-Interface des Editors wird damit im CI des OEM-Kunden gehalten, die ganzen HMI-Objekte werden spezifisch auf die Bedürfnisse des Produktes abgestimmt und können auf einfachste Weise erweitert werden.

Somit ist jedes einzelne FB-Objekt als einzelnes teq-File (SpiderControl Format) auf der HD der SPS abgelegt, das gleiche gilt für die UI's des Editors.



Bearbeitung der Web-HMI UI im PC-basierten Editor



## Bearbeitung der Objekt-Galerie des Web-HMI Editors im PC-basierten Editor

Ebenfalls ist es möglich, die im Target auszuführende Funktion eines FB's mittels des PC Editors zu programmieren.

Zusätzlich besteht die Möglichkeit, dass ein FB auch eine externe Funktion aufrufen kann. Auf einem uController kann dies eine C-Funktion der Firmware sein, welche einen spezifischen Algorithmus kapselt.

### Bildnachweis:

iniNet Solutions GmbH

### Pressekontakt

iniNet Solutions GmbH  
Fichtenhagstr. 2  
CH- 4132 Muttenz, Schweiz

Tel: 0041 61 716 96 26

Fax: 0041 61 716 96 17

E-Mail: [pr@ininet.ch](mailto:pr@ininet.ch)

Web: [www.ininet.ch](http://www.ininet.ch)