

IIoT ist nun einfach umsetzbar – denn Industrie 4.0 gibt's jetzt von der Stange!

RAMI: Theoretische Basis!

Das Thema Industrie 4.0 ist in aller Munde und nimmt so langsam Formen an.

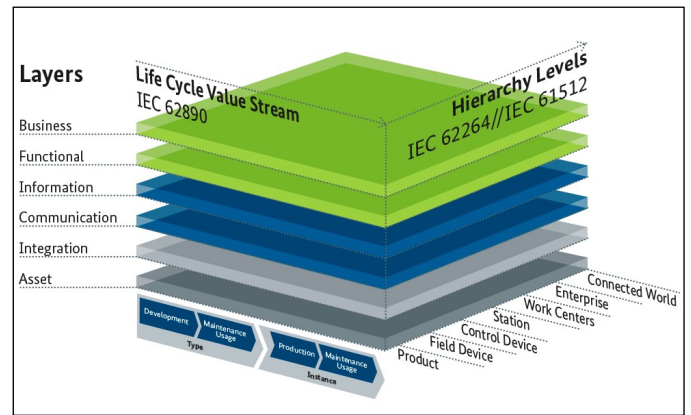
Das Referenzarchitekturmodell RAMI ist in eine dreidimensionale Architektur gegliedert, welche als Rahmenwerk fungiert und für unterschiedlichste Produktionsanforderungen die Zukunft bilden soll.

Dieses Modell soll auch die klassische ein- bis zweidimensionale Automationspyramide ersetzen. Während bei der bekannten Automationspyramide auf jeder Ebene konkrete Produkte, Protokolle oder Aufgabenstellungen problemlos eingefügt werden können, ist das RAMI Modell an vielen Stellen mehr als abstrakt. Hier muss noch intensiv an neuen Definitionen, Standards sowie Produkten gearbeitet werden.

Schaut man sich die verschiedenen Schichten des Modells genauer an, entsteht schnell der Eindruck, dass sich die Automationshersteller von den unteren Schichten nach oben arbeiten, während sich die IT entgegengesetzt von oben nach unten bewegt. Von daher ist es nicht verwunderlich, dass die drei Schichten „Functional“, „Information“ und „Communication“ bislang in Punkto Standards sowie Tools am wenigsten Substanz zu bieten haben. Doch denkt man an die Automationspyramide zurück, wird einem schnell klar, dass zwischen der Automationswelt und IT schon immer die größten Herausforderungen lagen. D.h. zwei Welten treffen aufeinander – und dieses Problem gilt es zu lösen!

Während das umfassende Kern-Know-how der Automatisierer im Verständnis des zu steuernden Prozesses liegen muss, sind ITler in erster Linie reine Informatiker. Wie aus der RAMI Grafik ersichtlich ist, fordert Industrie 4.0 gerade an dieser Schnittstelle eine Vervielfachung von Funktionen. Genau hier liegt der Knackpunkt des Referenzarchitekturmodells. Sollte die Industrie 4.0 scheitern, dann könnte es genau an dieser Stelle passieren.

Aus diesem Grund sollten wir uns jetzt Gedanken machen, wie die praktische Umsetzung von Industrie 4.0 Projekten auszusehen hat, welche Tools dafür benötigt werden und wie diese Werkzeuge ausgestattet sein müssen, um letztendlich eine erfolgreiche Industrie 4.0-fähige Lösung realisieren zu können.



Das Referenzarchitekturmodell Industrie 4.0 ©Plattform Industrie 4.0

Automationspyramide: nicht mehr zeitgemäß!

Was bedeutet Industrie 4.0 konkret für uns?

Kurz zusammengefasst:

Kunden möchten möglichst viele ihrer Prozesse digitalisieren. Und dies auf verschiedenen Ebenen: Betrieb, Beschaffung, Dokumentation, Logistik, Audit Trail, Vertrieb sowie in vielen weiteren Bereichen.

Damit dies alles umgesetzt werden kann, müssen im Vorfeld sämtliche Prozessanwendungen erfasst sowie analysiert werden. Wir möchten in diesem Artikel ein mögliches Szenario entwickeln, um diese Thematik besser beschreiben zu können: Digitalisierung ist am einfachsten, wenn ein Unternehmen, ausschließlich ein Produkt herstellt. In diesem Fallbeispiel kann die Automation für genau eine Aufgabe programmiert werden. Dies entspricht einer klassischen Aufgabenstellung jeder SPS-Programmierung. Für die weitere Digitalisierung werden beispielsweise Schnittstellen zur Produktionsplanung, Logistik oder Qualitätssicherung benötigt. Die Schnittstelle zur Produktionsplanung kann im einfachsten Fall daraus bestehen, dass die Anzahl der produzierten Einheiten ausgelesen und an die Auftragsbearbeitung übergeben wird. Auch die Logistik ist hier sehr einfach, da sich die Stückliste der produzierten Teile nie verändert.

So kann aus der Anzahl aller hergestellten Einheiten Rückschluss auf den Lagerbestand gezogen werden. Dadurch lassen sich wiederum Mindestbestände im Lager oder auch Materialeinkäufe steuern. Für die Qualitätssicherung ist es ausreichend, wenn relevante Parameter von diversen Automations- und Messtechnikkomponenten eingesammelt und über eine Schnittstelle nach oben weitergeleitet werden. Dieses Fallbeispiel zeigt auf, dass hier die traditionelle Automationspyramide für die Umsetzung der Aufgaben völlig ausreichend ist. Aus technischer Sicht werden die einzelnen Ebenen typischerweise mit dem Kommunikationsprotokoll Modbus oder OPC DA/UA verbunden.

Komplizierter wird es allerdings dann, wenn nicht immer nur das gleiche Produkt, sondern eine Palette unterschiedlicher Produkte in verschiedensten Varianten gefertigt wird. So wie es in den meisten Unternehmen der Fall ist.

Eine hochautomatisierte Produktion muss in der Lage sein, einen Auftragsbestand einer in der Zusammensetzung von Einzelteilen, Anzahl der zu produzierenden Einheiten sowie Art der Produktionsschritte hoch divergierenden Kombination abzuarbeiten. Dies macht die Aufgabenstellung äußerst komplex: Die Maschine muss für das zu fertigende Produkt jeweils gerüstet werden. Die benötigten Teile müssen in entsprechender Anzahl zur Verfügung stehen, der spezielle Produktionsablauf der Automation muss aktiviert werden und die passenden Schritte der Qualitätssicherung sowie Speicherung von Messungen müssen realisierbar sein. Bisher wurden diese komplexen Aufgaben von Facharbeitern durchgeführt. All diese Punkte wurden über Listen oder weitere Hilfsmittel abgearbeitet: das notwendige Material aus dem Lager geholt, die Maschine gerüstet und dann das Teil gefertigt. Hierfür wurde der Facharbeiter ausgebildet und hat sich bei der täglichen Arbeit das zusätzliche Know-how erarbeitet.

In der digitalisierten Welt werden die Aufgaben der Facharbeiter von einer Software übernommen. Doch warum scheitert gerade hier die klassische Automationspyramide? Auf der Automationsebene laufen nicht mehr nur lineare Prozesse ab, die über Datenpunkte gestartet und gestoppt werden können.

Ein einzelnes Produkt besteht nicht nur aus einer Stückliste, sondern verlangt auch spezifische Programmierung auf vielen unterschiedlichen Automationskomponenten, wie SPS, Robotik und CNC. Zudem verlangt die Herstellung dieses Produktes auch die Allokation von unterschiedlichen weiteren Produktionssystemen sowie deren Logistik. Durch die möglichen Kombinationen ergibt sich eine beliebige Komplexität.

Es stellt sich die Frage: Wo befindet sich die Intelligenz, welche weiß, wie ein Produkt hergestellt wird und welche Subsysteme und Komponenten dazu benötigt werden?

Die Automationspyramide sagt: Dies muss in der Leitebene erfolgen. Um das Problem über die klassische Automationspyramide zu lösen, könnte man so vorgehen, dass man auf den Steuerungen jede Produktvariante ausprogrammiert, d.h. jede zugehörige Materialförderung, jeder Test usw. Um die Produktion zu koordinieren, muss dann auf Betriebsleitebene (MES) der Ablauf für jedes zu produzierende Teil im Detail abgebildet werden, sodass die Vorgaben eine nach der anderen an die Steuerungen der Automationsebene geschickt werden können: Steuerung y, ich schreibe in dein Register, welches Programm du jetzt ausführen sollst, CNC m produziere jetzt Teil k usw. Diese Vorgehensart muss allerdings in der Lage sein, auf mögliche Unregelmäßigkeiten und Fehler schnell reagieren zu können, um einen kompletten Produktionsausfall zu vermeiden. Zwar wäre die beschriebene Umsetzung denkbar, stößt jedoch in der Praxis an ihre Grenzen: Denn einerseits muss man auf der Leitebene wesentliche Teile der Produktionsebene nachprogrammieren. Und: die Programmierer dieser Software benötigen hierfür ein umfassendes Wissen über die Programmierung der Automationsebene. Die Programmierung der Automationsebene erfordert nämlich die Definition von gewissen Standards, z.B. bei der Programmierung des SPS Codes, die im Sinne der Anbindbarkeit an die Leitebene eingehalten werden müssen. Um derartige Projekte realisieren zu können sind Ingenieure gefragt, die sich sowohl im Bereich der ERP, MES und IT auskennen und zugleich firm in Automation, SPS Programmierung sowie den Produktionsprozessen der jeweiligen Firma sind. Wie man weiß, sind solche Fachleute am Arbeitsmarkt eher selten zu finden.

Ferner hat diese Art von Architektur grundsätzliche Design-Fehler: Durch die teilweise Nachbildung der Automationsprozesse in der Leitebene entsteht an dieser Stelle eine Redundanz. Auf das alte Produktionsverfahren übertragen, wäre dies vergleichbar, wie wenn anstatt Facharbeitern ungelernete Hilfskräfte eingestellt werden würden. Jeder einzelne Arbeitsschritt müsste detailliert vorgegeben und kontrolliert werden. Und die Personen, welche diese Schritte kontrollieren müssen, kommen aus der Planung und Buchhaltung. Es entstünde so zu sagen sehr viel teurer Leerlauf.

Bezogen auf die Software-Architektur sowie praktische Umsetzung hat dies hier zwei Auswirkungen: Die Quasi-Nachbildung der Produktionsebene in der Leitebene findet auf einem recht abstrakten Niveau statt. Das Debugging ist enorm schwierig, weil weitestgehend im Blindflug programmiert wird. Dies führt zu hohen Entwicklungskosten. Noch kritischer wird es allerdings im weiteren Lebenszyklus dieser Lösung, da das System so komplex ist und Änderungen über die Jahre immer schwieriger werden. Verlassen dann noch Spezialisten mit wichtigem Know-how das Unternehmen, ist das Projekt nicht mehr wartbar. Aufgrund dessen wird es dann immer schwieriger, neue Produktionsverfahren sowie Produkte einzuführen. Damit wird das Gegenteil von dem erreicht, was Digitalisierung ursprünglich versprochen hat.

Die Lösung entsprechend der I4.0 Philosophie

Was ist anders bei I4.0?

Beginnen wir mit dem Thema Redundanz. Diese kann nur dann beseitigt werden, wenn die Produktionsprozesse ausschliesslich auf der Automationsebene programmiert werden. Die Leitebene legt fest, welche Aufträge bearbeitet und welche Produkte gefertigt werden sollen. Sie erteilt auch die Anforderungen an die Produktionsmittel, Produktionszeiten, Stücklisten, Testprotokolle und vieles mehr. Dies bedeutet auch, dass die Leitebene festlegt, was produziert werden soll - wie es aber produziert werden soll, bestimmt die Maschine.

Deshalb wird auf der Produktionsebene ein möglichst objektorientierter Ansatz verfolgt, im Sinne einer Orientierung an der Maschine als mechatronische Einheit. D.h. die jeweiligen Produktions- und Steuerungseinheiten holen sich die Aufträge von der Leitebene, kommunizieren mit den für die Produktfertigung notwendigen Subsystemen, allozieren diese und liefern ihnen den Link auf die Vorgaben aus der Leitebene. Der Materialbedarf bzw. -verbrauch wird an ein Logistiksystem gemeldet und der Produktionsstatus zurück an die Leitebene. Jedes Objekt beschäftigt sich ausschließlich mit seiner eigenen Aufgabe. Das bedeutet, dass zu jeder Steuerung nicht nur die I/O Ebene für die klassische Automation programmiert wird, sondern eben auch die Kommunikationsebene für die Einbindung in das I4.0 Netzwerk der Firma.

Für jedes anschließend darauf gefertigte Produkt wird möglichst „klassisch“ in IEC61131 auf der I/O Ebene der Automationssysteme programmiert. Also genau so, wie es bei einer Serienanfertigung eines einzelnen Produktes gemacht wird. Das garantiert eine effiziente Entwicklung, eine unkomplizierte Wartung und einen hohen Freiheitsgrad für zukünftige Erweiterungen. Eine Modularisierung von Produkten ist in den meisten Unternehmen längst Standard. Es ist nun zusätzlich ein Mechanismus gefordert, der das passende SPS Programm für zu fertigende Produkt aktiviert.

Die Steuerungseinheit führt Test, Messreihen oder Protokolle selbst aus. Auch Formulare oder Typenschilder werden automatisch erstellt. Sollte eine Arbeitsstation nicht vollautomatisch sein, so ist die Steuerungseinheit in der Lage, Arbeitsanweisungen für einen Arbeiter beispielsweise auf einem lokalen Display anzuzeigen.

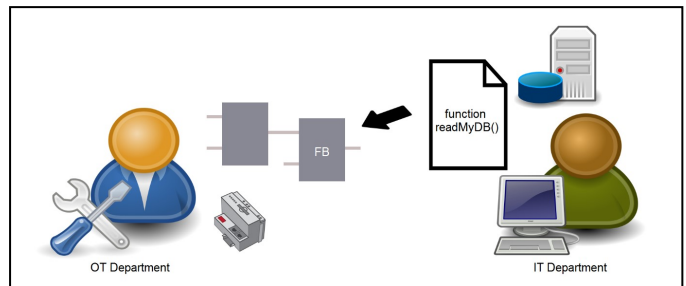
14.0 von der Stange: Die pragmatische Umsetzung

Wie kann diese Architektur umgesetzt werden?

Die Basis für den Aufbau der Steuerungseinheit bildet der SpiderControl SCADA Server. Diese performante Software ist nahezu für alle Betriebssysteme (Windows Standard, embedded Compact, Linux, embedded Linux, Raspian oder Android) verfügbar und wurde bereits auf aktuelle Geräten vieler namhafter SPS Hersteller portiert, die offene Steuerungen anbieten, wie z.B. Wago PFC, Phoenix PLCNext, Siemens IoT2040, industrielle RaspberryPi oder auch Industrie Router wie z.B. die MRX Serie von Insys. Ist der SCADA Server auf die SPS integriert, kann dieser direkt mit der lokalen SPS Applikation oder mit abgesetzten Automationskomponenten kommunizieren. In diesen SCADA Server ist die SpiderPLC eingebunden, welche eine einfache Programmierung von Funktionsplan über den Browser ermöglicht. Das Besondere daran: Die Funktionsbausteine führen nicht nur einfache Logik aus, sondern können zusätzlich Funktionen aus externen Scripten der wichtigsten aktuellen Hochsprachen, wie u.a. JavaScript (NodeJS, NodeRED), PHP, Python oder .Net aufrufen.

Hierzu wird zusätzlich eine Runtime für die Ausführung der benötigten Script-Sprachen auf der SPS installiert. Diese wird im Hintergrund gestartet und wartet darauf, dass sie eine Anfrage für die Ausführung einer Funktion aus Spider PLC empfängt.

Damit wird es möglich, den SPS Code direkt mit den Daten und Funktionen aus der Leitebene zu verknüpfen. Zur Anbindung an die IT Ebene können die Informatiker den Code zur Kommunikation mit deren Applikationen in der Hochsprache ihrer Wahl zur Verfügung stellen. Dieser Code wird über einen Funktionsbaustein gekapselt und in die Automation eingebunden.



Die vorher beschriebene Architektur kann beispielsweise wie folgt umgesetzt werden:

Um einen Auftrag an eine Produktionseinheit zu übergeben, kann das MES die notwendigen Informationen dazu z. B. in eine Datenbank schreiben, über ein REST API im Intranet ablegen, oder es gibt ein Software API zum Zugriff auf die entsprechende MES Software. Der Software Ingenieur, der sich im MES auskennt, hat nun mehrere Möglichkeiten. Er kann ein PHP Script erstellen, welches die entsprechenden Felder aus der Datenbank liest. Oder er liefert ein JavaScript, welches sein REST API auslesen kann. Oder er hat eine .Net Funktion, welche ein API direkt in seine MES Applikation implementiert.

Jetzt kann der SPS Programmierer die vom MES Spezialisten gelieferte Script-Funktion aus einem Funktionsbaustein in der Spider PLC aufrufen. Damit das Script die passenden Daten aus dem MES finden kann, werden alle notwendigen Parameter aus der SPS auf den Funktionsbaustein übertragen. Dazu muss die SPS angeben, für welches Produktionsmittel sie einen Auftrag abfragen will ('wer bin ich'). Das Script wiederum schreibt die Daten dann in den Namespace des SCADA Servers. Somit kann man auch größere Datenstrukturen mit einem Aufruf übergeben und auch wieder zurückbekommen. Die SPS weiss nun, was sie als nächstes produzieren soll. Über einen weiteren Funktionsbaustein kann Spider-PLC mit der Produkt-ID ein PHP Script aufrufen, welches aus einer SQL-Datenbank Produktinformationen zu diesem Objekt ausliest. Das Script erteilt Auskunft über Stücklisten, involvierte Subsysteme, Referenzen auf Dokumente, Testprotokolle, etc. und wird von einem Informatiker zur Verfügung gestellt, der mit der Wartung der Produkt-Datenbank vertraut ist.

Dieser Informatiker weiss, welche Parameter er von der SPS bekommen muss, damit er den gewünschten Datensatz findet. Den abgefragten Datensatz kann er als Antwort in ein XML Format packen, welches indessen vom SCADA Server verstanden und in seinen internen Namespace abgebildet wird.

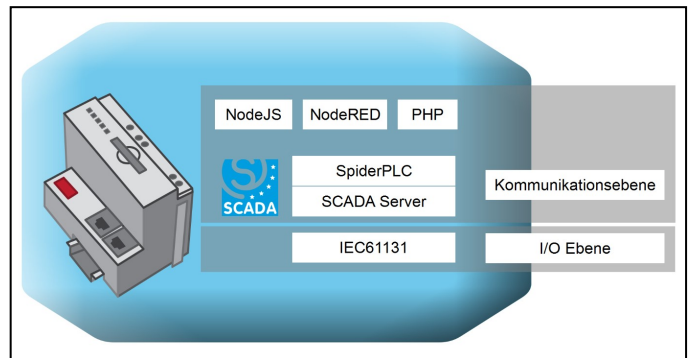
Danach alloziert die SPS die in den Produktionsprozessen involvierte Subsysteme, z. B. ein System welches für die Produktion das notwendige Material aus einem Lager entnimmt sowie für den Weitertransport konfektioniert. Die direkte Kommunikation kann durchaus über ein Feldbus Protokoll erfolgen. Förderlich wäre evtl., wenn die SPS eine Referenz auf die benötigten Datenbank Datensätze an das Subsystem weitergibt. Das Subsystem, das die benötigten Teile auf einen Transportträger konfektioniert, erstellt bei Bedarf einen Datensatz, wie die Teile angeordnet sind. Der Roboter, der dann die Teile vom Transportträger in die CNC Maschine fördert, kann dadurch mit sehr unterschiedlichen Konstellationen umgehen, weil die Spider PLC ihm mitteilt, welches Teil an welcher Position liegt.

Nun werden auf der SPS die für die Fertigung des gewünschten Teiles erforderlichen Programme (in IEC61131) aufgerufen. Diese geben über einen Datenpunkt ein Signal, wann die Sequenz abgeschlossen ist. Somit können die Teile ausgefördert und der Status des Auftrags im MES mutiert werden.

Testprotokolle werden zeitgleich an einer anderen Station in Empfang genommen und durchgeführt. Die Messungen werden in Registern der SPS abgelegt und über die Spider PLC wird ein PHP Script aufgerufen, welches diese Variablen als Parameter übergibt sowie beispielsweise mit der FPDF PHP Library ein PDF Dokument erzeugt.

Da die Daten parallel dazu in einer Datenbank abgelegt werden, ist die Produktion jederzeit zurück verfolgbar. Zudem wird stets eine aktuelle Statistik über die Qualität der Produktion geführt, welche Rückschlüsse über den Anlagenzustand und andere produktionsrelevante Parameter erlaubt. Da für diese Aufgaben z. B. oftmals Python verwendet wird, ergibt sich der Übergang zum Machine Learning nahezu von selbst. Python Funktionen können analog über Funktionsbausteine direkt aus Spider PLC aufgerufen werden.

Sollte an der Anlage u.a. eine Wartung notwendig sein, kann problemlos über Spider PLC direkt ein E-Mail an den zuständigen Subunternehmer gesandt werden.



Unser Fazit

Anhand dieses einfachen Fallbeispiels stellen wir fest, wie einfach gängige Abläufe auf Basis der Spider PLC / SCADA Infrastruktur umgesetzt werden können. Der entscheidende Vorteil besteht darin, dass die Programmierung von I4.0-Anwendungen unkompliziert mit dem eigenen Personal realisiert werden kann. Es bedarf keiner neuen, teuren Spezialisten – die zudem auf dem Arbeitsmarkt schwer zu finden sind.

Ohne große anfängliche Investitionen kann man an einer einzelnen Maschine damit beginnen, die Verbindung mit der Leitebene zu schaffen. Erfahrungsgemäß werden im Anschluss daran Schritt für Schritt weitere Komponenten – nach dem gleichen System – eingebunden. Das Risiko für das Scheitern des Projekts mit daraus resultierenden Folgekosten ist hierdurch weitestgehend eliminiert.

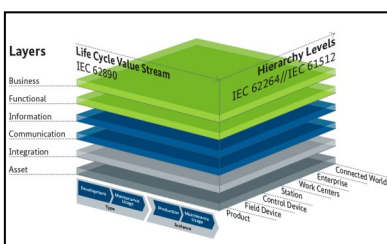
Das klassische Problem in Hinblick auf die Überwindung des Grabens zwischen der Automations- und IT-Ebene wird hier ebenfalls elegant gelöst, da beide Seiten in den technologischen Domänen bleiben können: Die IT liefert alle relevanten Schnittstellen in einer für sie bekannten Programmiersprache. Die Automations- und IT-Ebene verwendet diese als Funktionsbaustein in einer Funktionsplanprogrammierung, welche über das SCADA und seine Treiber Zugriff auf alle Automationskomponenten hat.

Weitere Vorteile liegen auf der Hand: Die Erfassung und Bewirtschaftung der Daten werden gleichzeitig für die I4.0 Funktionalität sowie für die Visualisierung an der Anlage oder im Leitsystem genutzt. Durch dieses einzige Tool werden enorme Kosten gespart sowie die Effektivität erheblich gesteigert.

Durch das Konzept der verteilten Intelligenz wird die Hierarchie der Automationspyramide aufgebrochen und durch eine dreidimensionale Vernetzung – entsprechend des RAMI Modells – ersetzt. Über dieselben Kommunikations- und Steuerungskomponenten werden zudem die Aspekte der Produktion abgedeckt. Auch andere Funktionen wie Life-Cycle-Management, für die Wartung des Anlagenparks und die Prozessoptimierung werden genutzt.

Das Thema I4.0 beginnt letztendlich mit der Zusammenführung der bestehenden Maschinen und Software. Sobald die Informationen in den Unternehmen frei fließen, kann auch die Cloud so verwendet werden, wie es der jeweilige Betrieb wünscht. Das Thema Cloud ist auf keinen Fall eine Bedingung für I4.0 – sondern kann individuell im Bedarfsfall und vor allem jederzeit ergänzt werden. Die hierfür benötigten Interfaces sind bereits vorhanden.

Bildmaterial / Titel / Copyright-Hinweise:



Titel- und Copyrightangabe:

Das Referenzarchitekturmodell Industrie 4.0 ©Plattform Industrie 4.0

Weitere Informationen unter:

https://www.plattform-i40.de/SiteGlobals/PI40/Forms/Listen/Online-Bibliothek/DE/Online-Bibliothek_Formular.html?re-sourceId=1105098&input_1=1105090&pageLocale=de&templateQueryString_Listen=&to=&from=&cl2Categories_TechnologieAnwendungsbereich_name=Standards&cl2Categories_TechnologieAnwendungsbereich_name.GROUP=1&selectSort=&selectSort.GROUP=1

Kontakt:

iniNet Solutions GmbH
Fichtenhagstr. 2
CH- 4132 Muttenz

Tel: 0041 61 716 96 26
Fax: 0041 61 716 96 17

E-Mail: info@ininet.ch

Web: www.ininet.ch