

# Komponentenbasierte Entwicklung von Firmware für Sensorsysteme in der industriellen Praxis

Philipp Nenninger\*, Tilo Merlin<sup>+</sup>, Dirk John\*, Florian Kantz\*

\*ABB AG Forschungszentrum  
Wallstadterstr. 59  
68526 Ladenburg

<sup>+</sup>ABB Automation Products GmbH  
Borsigstraße 2  
63755 Alzenau

philipp.nenninger@de.abb.com  
tilo.merlin@de.abb.com  
dirk.john@de.abb.com  
florian.kantz@de.abb.com

**Abstract:** Um der steigenden Komplexität und Variantenanzahl in der Entwicklung von Transmittern für die industrielle Automatisierungstechnik zu begegnen wurde bei ABB bereits seit 2003 damit begonnen Transmitter mit Hilfe eines komponentenbasierten Ansatzes zu entwickeln. Dieser Artikel gibt einen Überblick über die Hintergründe und Grundlagen des Ansatzes sowie über Erfahrungen, welche bei dieser erfolgreichen Einführung von komponentenbasierter Entwicklung gemacht wurden.

## 1 Einleitung

Durch verschiedene Trends in der industriellen Automatisierungstechnik wurden einfache Sensoren für Größen wie beispielsweise Temperatur oder Druck im Laufe der vergangenen 20 Jahre deutlich komplexer. Kommunizierte ein einfaches Gerät bis dahin den Messwert über eine einfache Punkt-zu-Punkt Verbindung mit einer 4-20 mA Schleife zu dem angeschlossenen Leitsystem, so sind heute Transmitter mit Feldbusanschluss, aufwendiger und oft mehrsprachiger Benutzerschnittstelle (Human machine interface, HMI) mit mehrzeiliger Anzeige (LCD) sowie vielfältigen Diagnosemöglichkeiten für das Gerät selbst aber auch den angeschlossenen Prozess im Angebot aller namhaften Hersteller zu finden. Ein Großteil dieser Erweiterungen wurde erst durch Implementierung der Funktionalität in Software rentabel.

Die eingangs beschriebenen einfachen Geräte wurden aber nicht vollständig von neueren, komplexen Transmittern ersetzt; vielmehr koexistieren diese Geräte und bedienen verschiedene Teilmärkte. Durch diesen Umstand ist die Zahl der zu pflegenden Varianten deutlich gestiegen, was bereits gegen Ende der 1990er zu Problemen in der Entwicklung und vor allem der Wartung dieser Varianten führte. Im Fall von ABB kam zu dieser Zeit eine aggressive Akquisitionsstrategie hinzu, durch die namhafte Transmitterhersteller wie unter anderem Hartmann & Braun, Elsag Bailey und Fischer & Porter integriert wurden. Durch diese Akquisitionen wurde das Portfolio an Geräten deutlich erweitert und bildete den Grundstock für das Instrumentierungsportfolio, für das ABB bekannt ist.

Da diese Firmen aber jeweils eigene Entwicklungsprozesse und –werkzeuge verwendeten, ergab sich ein sehr heterogenes Bild. So existierten zum Beispiel zeitweise 40 verschiedene Elektronikplattformen, von denen wenige in relevanten Stückzahlen gefertigt wurden. Darüber hinaus ergab sich für den Benutzer keine einheitliche Bedienung verschiedener Geräte, selbst wenn sich die Funktionalität gleich, wie beispielsweise die Einstellung der Minimum- und Maximumwerte oder Alarmschwellen.

Vor diesem Hintergrund wurde bereits 2003 mit der Einführung eines komponentenbasierten Entwicklungsprozesses begonnen. Nicht zuletzt aufgrund der in der industriellen Instrumentierung üblichen langen Produktlebenszyklen von typischerweise bis zu 10 Jahren zeigten sich die gewünschten Effekte jedoch nicht sofort. In diesem Artikel wird in Kapitel 2 ein Überblick über typische Instrumentierungsgeräte sowie über den komponentenbasierten Entwicklungsansatz gegeben. Kapitel 3 beschäftigt sich mit den Erfahrungen, die durch den Einsatz des komponentenbasierten Entwicklungsansatzes in Projekten sowie der Nichtverwendung in einem Fall resultierten. Kapitel 4 gibt eine Zusammenfassung sowie einen Ausblick.

## **2 Komponentenbasierte Entwicklung von Transmittern**

### **2.1 Systembeschreibung der Transmitter**

Viele der in Kapitel 1 eingeführten Transmitter sind sich in Grundzügen ähnlich. Ein schematischer Aufbau eines solchen Transmitters ist in Abbildung 1 links zu sehen.

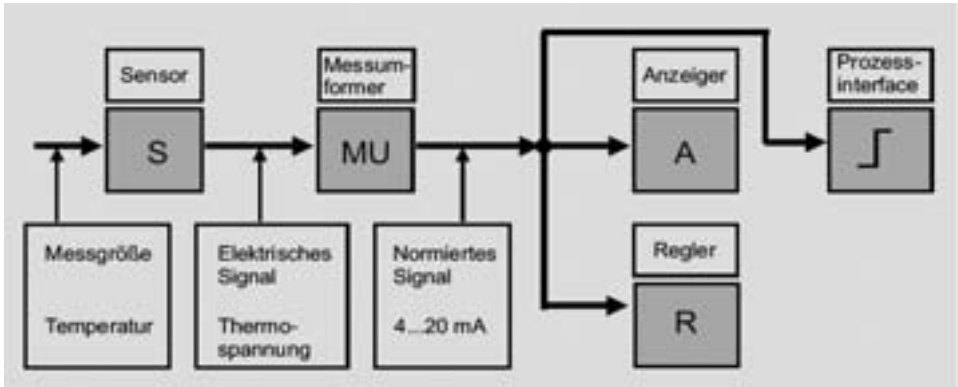


Abbildung 1: Komponenten der Temperaturmesskette (aus [EFG08])

Die eigentliche (physikalische) Messgröße, hier die Temperatur, wird mit Hilfe eines Sensors in ein elektrisches Signal umgewandelt. Typischerweise handelt es sich hier in industriellen Anwendungen um Widerstandsthermometer (Pt100/Pt1000), oder Thermoelemente. Diese ändern ihren Widerstand, beziehungsweise bei Thermoelementen die Ausgangsspannung, in Relation zu der angelegten Temperatur. Dieses elektrische Signal wird von einem Messumformer in ein normiertes Signal umgewandelt, hier 4–20 mA. Hierbei werden Nichtlinearitäten der Sensorelemente ausgeglichen und die Temperatur auf einen vorgegebenen Messbereich abgebildet, der nun in ein Gleichstromsignal überführt werden kann. Anzeiger, Regler und Prozessinterface, Abbildung 1 rechts, sind sowohl als separate Geräte als auch heute teilweise integriert erhältlich.

Diese Bausteine sind in den meisten am Markt erhältlichen Transmittern integriert, wobei je nach Messprinzip auch eine aktive Sensoranregung durch einen Regelkreis notwendig sein kann.

Handelt es sich um ein relativ einfaches Messprinzip, wie beispielsweise im Falle der Temperatur- oder Druckmessung, so kann der Aufbau der Elektronik des Messumformers in etwa der in Abbildung 2 gezeigten Struktur entsprechen.

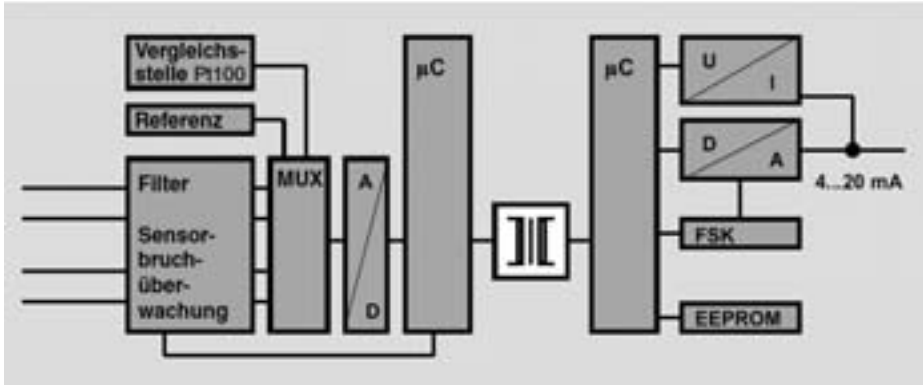


Abbildung 2: Blockschaltbild eines digitalen Transmitters (aus [EFG08])

Auffallend ist die galvanische Trennung der beiden Mikrocontroller ( $\mu\text{C}$ ) die unter anderem zur Verhinderung von Erdschleifen benötigt wird. In der hier gezeigten Anordnung wird auf beiden Seiten der galvanischen Trennung ein gewisses Maß an „Intelligenz“ in Form eines Mikrocontrollers benötigt.

Eine typische Aufgabenteilung würde beispielsweise Funktionalitäten wie Detektion von Leitungsbrüchen am Sensor und einfache Filterfunktionen auf dem dem Sensor zugewandten Mikrocontroller platzieren, während aufwendigere Signalverarbeitung sowie Kommunikationsprotokolle gegebenenfalls auf dem Mikrocontroller der anderen Seite angesiedelt werden.

## 2.2 Komponentenbasierte Systementwicklung

Die komponentenbasierte Entwicklung kann in eingebetteten Systemen nur selten von dem Gesamtsystem getrennt betrachtet werden, was auch durch Abbildung 2 deutlich wird. Daher wurde ein Ansatz gewählt, der Systeme in Plattformen gruppiert und in diesen eine höchstmögliche Wiederverwendung von Hard- und Softwarekomponenten anstrebt. Da sich die einzelnen zugekauften Transmitterhersteller oft auf bestimmte Messgrößen spezialisiert hatten, waren Plattformen, wenn sie denn vorhanden waren, entlang dieser Achse organisiert. Im Gegensatz zu der bis dahin üblichen vertikalen Gruppierung nach Messgrößen wurde jedoch eine horizontale Plattformstrategie gewählt, siehe auch Abbildung 3.

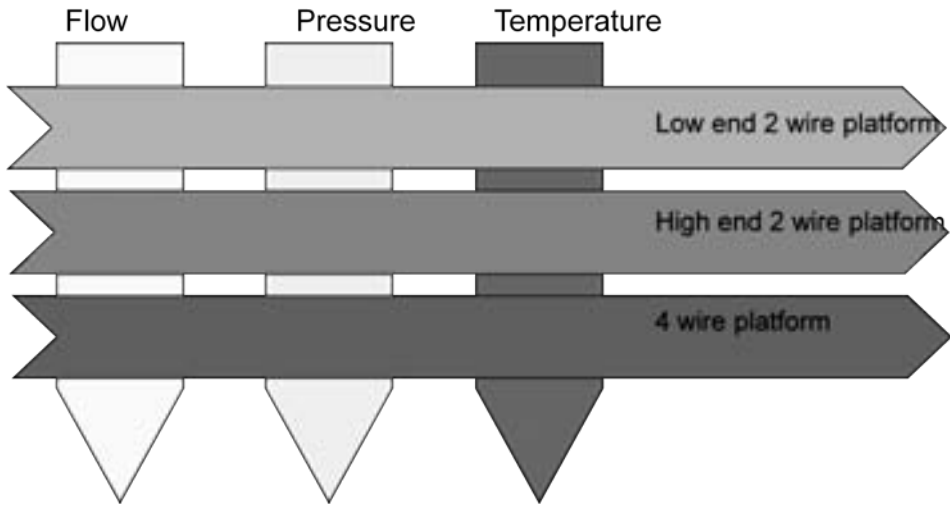


Abbildung 3: Plattformen bilden die Grundlage der komponentenbasierten Entwicklung

Über die Wiederverwendung in einer Plattform hinaus wurde auch die vertikale Wiederverwendung von Komponenten in verschiedenen Produktreihen<sup>1</sup> angestrebt. Hierfür eignen sich beispielsweise Komponenten, die die Linearisierung von Sensorkennlinien implementieren.

Um eine grundlegende Vereinheitlichung der Geräte vornehmen zu können, wurden im ersten Schritt Einschränkungen bezüglich der zu verwendenden Mikrocontroller durchgesetzt. Als Programmiersprache hatte sich C mit kleinen Anteilen Assembler als de-facto Standard durchgesetzt. Mit einem Standardcompiler sowie einem über alle Plattformen gemeinsamen Betriebssystem war also eine gemeinsame Basis vorhanden, auf der die komponentenbasierte Softwareentwicklung fußen konnte.

### 2.3 Komponentenbasierte Softwareentwicklung

Als Grundkomponente der Softwareentwicklung wurde das sogenannte Subsystem verwendet. Ein Subsystem wurde definiert als eine Gruppe verwandter Funktionen und Daten, die mit Subsystemen anderer Teile der Software über definierte und bekannte Interfaces kommuniziert. Dies entspricht der gängigen Definition von Komponenten, siehe auch [SVE07].

Die folgenden Abschnitte geben einen Überblick über die Kernpunkte des komponentenbasierten Ansatzes.

<sup>1</sup> Hierbei handelt es sich nicht um „product lines“ im Software-Engineering Sinne.

## **Vereinheitlichung der Struktur**

Um die Wiederverwendung zu vereinfachen, wurden grundlegende Teile der Interfaces standardisiert. Dies ermöglicht auch Entwicklern, die eine Komponente nicht kennen einen rudimentären Umgang mit dieser. Diese Standardisierung kann beispielsweise durch Code-Reviews sichergestellt werden. Um die Akzeptanz bei den beteiligten Entwicklern zu steigern, wurde jedoch auf einen Codegenerator gesetzt, der einerseits die Einheitlichkeit der Interfaces sicherstellt und den Entwickler andererseits von der Arbeit entlastet, eben diese Interfaces, oder wenigstens grundlegende Teile davon, zu implementieren.

## **Flexibilität und Robustheit**

Da die Komponenten mit dem Ziel implementiert werden, sie möglichst unverändert in verschiedenen Produktreihen weiterzuverwenden, stellt sich im Gegensatz zu herkömmlicher Entwicklung die Herausforderung, dass diese auch mit nicht vorhergesehener Benutzung umgehen können. Da Softwareteile bisher relativ aufgaben- und produktspezifisch entwickelt wurden und die Benutzung der Transmitter typischerweise wenige Variationen aufweist, ist dies ein nicht zu unterschätzender Unterschied in dem Entwicklungsansatz für Softwareteile und Softwarekomponenten. Im Unterkapitel 3.1 wird ein drahtloses Gerät als Beispiel für diese unvorhergesehenen Anforderungen näher beschrieben. Die Anforderung der „Robustheit“ schlägt sich in der Implementierung oft durch wiederholtes Überprüfen von Annahmen, besonders in der Umgebung des Interfaces nieder.

## **Erweiterte Testszenarien**

Mit dem Antizipieren neuer Verwendungen geht auch die Erweiterung des Testszenarios einher. Dies wird jedoch durch die Wiederverwendung existierender und damit auch ausführlich getesteter Komponenten mehr als wettgemacht. Dies zahlt sich besonders bei der Entwicklung und Integration neuer Komponenten aus, da existierende, robuste Komponenten als Fehlerquelle (quasi) ausgeschlossen werden können.

## **Vereinheitlichte Dokumentation**

Der Dreh- und Angelpunkt der Wiederverwendung ist die ausführliche Dokumentation der Komponenten und hier besonders der Interfaces. Erst durch gute Dokumentation ist es möglich, funktionierende und getestete Komponenten ohne deutlichen Mehraufwand zu integrieren. Zu diesem Zweck wurde die Dokumentation der Komponenten während des gesamten Entwicklungsprozesses vereinheitlicht. So ist gewährleistet, dass die Verwendung in einem antizipierten Bereich stattfindet und Auswirkungen, beispielsweise auf Laufzeit und Speicherbedarf auch bei Wiederverwendung früh abschätzbar sind.

Die Unterschiede zwischen dem komponentenbasierten Entwicklungsansatz und der herkömmlichen Methode lassen sich also wie folgt zusammenfassen:

- Vereinheitlichung der Struktur, Dokumentation und grundlegender Interfaces
- Design der Komponenten berücksichtigt Wiederverwendung in andersartigen Anwendungen

- Ausführliches Testen, berücksichtigt auch die Robustheit in andersartigen Anwendungen
- Erweiterte und standardisierte Dokumentation

### 3 Komponentenbasierte Entwicklung in der Praxis

#### 3.1 Wiederverwendung von Komponenten über Produktreihen und Produktgenerationen

Das für den Anwender der nach dem komponentenbasierten Ansatz entwickelten Geräte auffälligste Merkmal ist die Verwendung eines einheitlichen HMIs, siehe auch Abbildungen 4 und 5.



Abbildung 4: Auswahl aus dem Geräteportfolio vor (links) und nach der Entwicklung mit Hilfe des komponentenbasierten Ansatzes

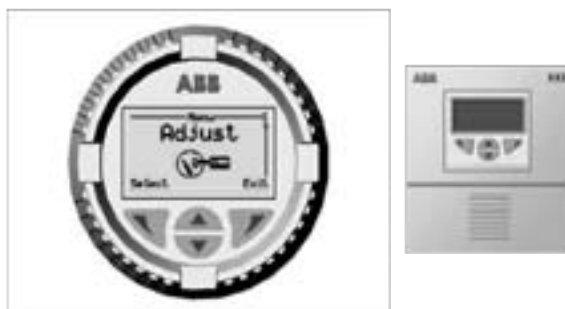


Abbildung 5: Wiederverwendung der HMI Komponente in verschiedenen Produkten

Abbildung 5 zeigt das HMI in zwei verschiedenen Bauformen. Diese Bauformen basieren jedoch intern auf der gleichen Software und ähnlichen Hardwarekomponenten und wurden lediglich im Formfaktor an die Geräte angepasst.

In Tabelle 1 ist ein Vergleich der ersten sechs Projekte, die mit dem komponentenbasierten Ansatz durchgeführt wurden, zu sehen. Dabei ist in der linken Spalte die Anzahl der Varianten aufgeführt, die ohne Zuhilfenahme des komponentenbasierten Ansatzes benötigt worden wären. Die rechte Spalte zeigt die tatsächlich mit dem komponentenbasierten Ansatz entstandenen Varianten.

| <b>6 Projekte mit komponentenbasierten Ansatz</b> | <b>6 Projekte ohne komponentenbasierten Ansatz</b> |
|---|--|
| 5 HMIs  | 2 HMIs   |
| 5 SW Architekturen                                | 1 SW Architektur                                   |
| 3 Hart HW/SW                                      | 1 Hart HW/SW                                       |
| 3 FF & Profibus HW/SW                             | 1 FF & Profibus HW/SW                              |
| 3 Gerätetreiber Technologien                      | 1 Gerätetreiber Technologie                        |
| 5 Diagnosesysteme                                 | 1 generisches Diagnosesystem                       |
| 5 Gehäuse   | 4 Gehäuse  |
| <b>Gesamt = 29</b>                                | <b>Gesamt = 11</b>                                 |

Tabelle 1: Vergleich der Varianten in sechs Projekten ohne (links) und mit dem komponentenbasierten Ansatz (rechts)

Besonders hervorzuheben ist hierbei, dass für sechs verschiedene Geräte lediglich eine Softwarearchitektur verwendet wurde. Obwohl dies nicht zwingend durch den komponentenbasierten Ansatz vorgegeben ist, lagen die Vorteile einer Wiederverwendung deutlich auf der Hand.

Bei diesen ersten sechs Projekten handelt es sich um Transmitter, die sich nach Abbildung 3 sowohl in der vertikalen als auch der horizontalen Dimension unterscheiden. Trotz dieser großen Unterschiede ist es dennoch gelungen, sowohl die Kommunikationskomponenten (HART, Foundation Fieldbus (FF) und Profibus) als auch die Diagnose<sup>2</sup> wiederzuverwenden. Allein hier wurde durch den komponentenbasierten Ansatz und die sich daraus ergebende Wiederverwendung die Entwicklung von sechs Komponenten eingespart.

Lediglich bei den HMIs wurden zwei Varianten entwickelt. Der Hauptgrund hierfür war, wie oben erwähnt, der Formfaktor. Nachfolgende Projekte zeigten gerade bei dem HMI jedoch neue Herausforderungen auf. Transmittern werden typischerweise an Stellen im Prozess verbaut, die nicht einfach zugänglich sind, oft auch in für Menschen gefährlichen Zonen. Das HMI wird daher normalerweise nur zur Konfiguration und Inbetriebnahme und seltenen Routinekontrollen verwendet. Die Werte des Transmitters werden am Leitsystem in der Leitwarte überwacht.

<sup>2</sup> Die Diagnosekomponente ist hierbei hauptsächlich für die Diagnose des Gerätes zuständig und macht keine Aussagen über den Zustand des Prozesses.



Andere Geräte wie beispielsweise Regler und Schreiber werden jedoch genau dort installiert und regelmäßig abgelesen. Das HMI spielt bei diesen Geräten daher verständlicherweise eine wichtigere Rolle, die nicht durch das einfache Transmitter-HMI wie in Abbildung 5 gezeigt ausgefüllt werden kann. Das aufwendige HMI eines Schreibers, welches beispielsweise ein 5“ Farb-TFT enthalten kann, ist jedoch für Transmitter ungeeignet. Hier ist also ein one-size-fits-all Ansatz ungeeignet.

In Unterkapitel 2.3 wurde betont, dass Komponenten so entwickelt werden sollten, dass sie auch unter nicht vorhergesehenen Randbedingungen wiederverwendet werden können. Diesem Test wurden einige Komponenten unterzogen, als ein erster Forschungsdemonstrator eines drahtlosen Temperaturtransmitters entwickelt werden sollte. Bisherige Geräte wurden entweder über die 4—20 mA Schleife oder eine separate Energieversorgungsleitung konstant mit Energie versorgt.

Bei drahtlosen Geräten ist aber im Batteriebetrieb entscheidend, dass die Energieaufnahme möglichst gering und die Lebensdauer der Batterie damit möglichst lang ist. Eine einfache Methode, Energie zu sparen, ist es, die Elektronikkomponenten in low-power Zustände zu versetzen und nur aktiv arbeiten zu lassen, wenn gerade ein Messwert benötigt wird. Hierzu wird das System quasi angehalten, ein Zustand, der in bisherigen Geräten nicht vorgesehen war.

Als weitere Herausforderung kam hinzu, dass die meisten Entwickler an dem Projekt weder an Temperaturtransmittern noch mit dem komponentenbasierten Ansatz gearbeitet hatten. Hier stellten sich die einheitliche Dokumentation sowie die einheitlichen Interfaces als deutlicher Vorteil heraus, da die Einarbeitungszeit verkürzt werden konnte. Darüber hinaus verbesserten sie auch die Kommunikation mit den Entwicklern der Komponenten, da eine gemeinsame Wissensbasis vorhanden war.

In diesem Projekt wurde deutlich, dass die mit dem komponentenbasierten Ansatz entwickelten Komponenten die Anforderung an eine Wiederverwendung in nicht vorhergesehenen Anwendungen erfüllen. Da die Komponenten bereits in ihrer ursprünglichen Verwendung getestet waren, wurde der zusätzliche Testumfang deutlich verkleinert.

Das Ergebnis des Projekts ist der in Abbildung 6 zu sehende Forschungsdemonstrator. Neben den oben beschriebenen Merkmalen wurde zusätzlich ein energy-harvesting Modul integriert, welches das Gerät durch die Temperaturdifferenz zwischen Prozess und Umgebungstemperatur speist.

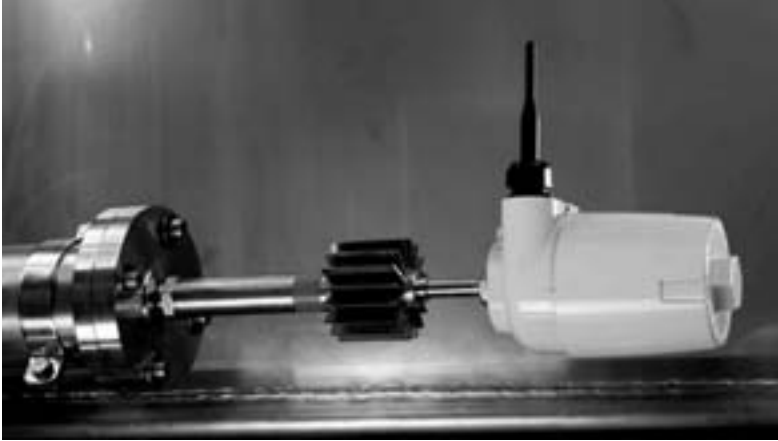


Abbildung 6: Forschungsdemonstrator eines autonomen Temperaturtransmitters, der mit dem komponentenbasierten Ansatz entwickelt wurde

### 3.2 Projekte außerhalb des komponentenbasierten Entwicklungsansatzes

Trotz der Vorteile des komponentenbasierten Ansatzes wurden auch immer wieder Systeme oder Teilsysteme ohne diesen entwickelt. So wurde beispielsweise ein Mikrocontroller, wie er in Abbildung 2 links zu sehen ist, nicht nach dem komponentenbasierten Ansatz neu entwickelt. Hier wurde die Kommunikation über die galvanische Trennung als Interface verwendet und Wiederverwendung war aufgrund der eingeschränkten Funktionalität quasi ausgeschlossen.

Als bei der Entwicklung eines anderen, ähnlichen Produktes durch eine Verlegung der galvanischen Trennung der zweite Mikrocontroller überflüssig wurde, wurde die Funktionalität in eine Komponente überführt und steht so für weitere Entwicklungen zur Wiederverwendung bereit.

Eine ähnliche Tendenz wurde bei einem Gerät festgestellt, welches von seinem Einsatzzweck her kein Transmitter im herkömmlichen Sinne ist, sondern sich auf bestimmte Aspekte der Kommunikation beschränkt. Auch hier wurde zu Beginn der Entwicklung bewusst auf die Verwendung des komponentenbasierten Ansatzes verzichtet, da eine Wiederverwendung nicht absehbar war. Aufgrund technischer Entwicklungen wurde im Laufe der Zeit eine Wiederverwendung von Teilen der Funktionalität aber opportun, weshalb diese Teile ebenfalls in Komponenten überführt wurden.

Es ist also festzustellen, dass Wiederverwendung, ob vorhergesehen oder nicht, eher die Regel als die Ausnahme darstellt. Dieser Trend wird natürlich durch die durch den komponentenbasierten Ansatz zur Verfügung gestellte Infrastruktur begünstigt.

## **4 Zusammenfassung und Ausblick**

Der in diesem Artikel beschriebene, komponentenbasierte Entwicklungsansatz wurde seit 2003 schrittweise eingeführt, sodass heute bis auf wenige Ausnahmen alle Produkte der entsprechenden Geschäftseinheit komponentenbasiert entwickelt werden. Die Hauptziele, nämlich die Vereinheitlichung von Produkten insbesondere der Benutzerschnittstellen und die Wiederverwendung von bestehenden Komponenten wurden erreicht und teilweise übertroffen. Hierbei existieren Beispiele von Komponenten, die für ihre Wiederverwendung in den komponentenbasierten Ansatz überführt wurden, obwohl sich bei der ursprünglichen Entwicklung explizit dagegen entschieden wurde.

Bei der Wiederverwendung von Komponenten, besonders in Anwendungsfällen, für die diese Komponenten nicht entworfen wurden, ist aber sorgfältig zu prüfen, ob eine direkte Wiederverwendung sinnvoll ist, oder Randbedingungen wie Speicherbedarf oder erforderliche Rechenzeit nicht das neu zu entwerfende System überfordern. In diesem Zusammenhang ist eine Kombination mit einem Produktlinienansatz vielversprechend.

Durch den komponentenbasierten Ansatz ist es aber heute schon möglich, einerseits bei steigender Komplexität die gleichbleibend hohe Qualität der entwickelten Transmitter zu sichern, und andererseits flexibel auf neue Kundenanforderungen zu reagieren. Daher war die Einführung des komponentenbasierten Ansatzes eine für die Steigerung der Wettbewerbsfähigkeit hilfreiche Maßnahme.

### **Literaturverzeichnis**

- [EFG08] Ehinger, K.; Flach, D.; Gellrich, L.; Horlebein E.; Huck, R.; Ilgner, H.; Kayser, T.; Müller, H.; Schädlich, H.; Schüssler, A.; Staab, U.: Praxis der industriellen Temperaturmessung, ABB, 2008
- [SVE07] Stahl, T.; Völter, M.; Efftige, S.; Haase, A.: Modelgetriebene Softwareentwicklung, dpunkt.verlag, 2007