Industrie-PC realisiert Sicherheitsfunktionen

Moderne Automatisierungssysteme werden immer häufiger mit leistungsfähigen Industrie-PCs zur Ausführung von Steuerungsaufgaben ausgestattet. Bisher fehlte jedoch die Möglichkeit, diese Leistungsfähigkeit auch zur Ausführung von Sicherheitsfunktionen zu nutzen. Die Komplexität der IPCs verhinderte Sicherheitsnachweise nach den international akzeptierten Sicherheitsnormen, wenn herkömmliche Steuerungslösungen einfach auf den IPC übertragen werden sollten. Durch den Einsatz mathematischer Codierungen konnte der IPC zur Bearbeitung von Sicherheitsfunktionen nach SIL 3 ertüchtigt werden.

Als Ende der achtziger Jahre die ersten PC-basierten Steuerungen eingesetzt wurden, war die Skepsis groß. Schließlich hatte jeder Anwender in seiner täglichen Arbeit Erfahrungen mit den üblichen Büro-PCs gemacht und wollte die ihm dabei begegnete (Un-) Zuverlässigkeit nicht auf seine Produktion übertragen. Obwohl schon damals die IPCs einen hohen Qualitätsstandard aufwiesen und seitdem immer besser und leistungsfähiger wurden, ist auch heute noch ein Hauch von Skepsis spürbar. Da erstaunt es manchen, dass nun sogar IPCs für Sicherheitsfunktionen eingesetzt werden können.

Warum IPC-basierte Sicherheit?

Die Grundlage für den Einsatz von IPCs für Sicherheitsfunktionen liegt natürlich in erster Linie bei der hohen Qualität der modernen IPCs und der auf sie zugeschnittenen Firmware. Die erfolgreiche Entwicklung von fehlersicheren Klemmen und der hohe Standard der fehlersicheren Kommunikation, wie Safety-over-EtherCAT, tragen ihren Teil zum erfolgreichen Einsatz bei.

Wie jedoch können nun Berechnungen auf einer PC-basierten Steuerungsplattform mit einer nachweisbaren Sicherheit erfolgen? Die

Aufgaben von fehlersicheren Automatisierungssystemen bestehen in der Erkennung von Fehlern mit nachweisbar hoher Wahrscheinlichkeit und einer sicheren Reaktion über die Peripherie. Um Fehler zu erkennen, werden üblicherweise redundante Hardware- und Softwarekanäle genutzt; nur wenn die Kanäle zueinander konsistente Daten liefern, werden die Ergebnisse als korrekt betrachtet und weiter verwendet. Man kann leicht erkennen, dass mehrere Kanäle auch höhere Produktkosten bedeuten.

Weiterhin muss für die Einhaltung internationaler Sicherheitsstandards auch nachgewiesen werden, dass es keinen singulären Fehler gibt, der sich auf mehrere Kanäle gleich auswirkt, so dass ein Vergleich erfolgreich verläuft und der Fehler dadurch nicht erkannt werden kann (sogenannter common cause). Auch bei der Nutzung der verschiedenen Prozessorkerne eines Multi-Core-Prozessors ist genau diese Eigenschaft sehr kritisch zu prüfen, denn viele Ressourcen werden von mehreren oder sogar von allen Kernen gemeinsam genutzt. Effizienter ist dagegen der Einsatz mathematischer Methoden, die ebenfalls die Verwendung von Multi-Core-Prozessoren unterstützen, bei denen jedoch die Aufteilung der Teilaufgaben auf die einzelnen Prozessoren und die Art der Nutzung gemeinsamer Ressourcen keine Sicherheitsmerkmale sind, die nachgewiesen werden müssen.

Die auf einem IPC leicht mögliche Kombination aus mehreren Softwarekanälen mit verschiedenen Codierungen erhöht die Güte der Fehleraufdeckung bzw. senkt die Restfehlerwahrscheinlichkeit beträchtlich (siehe Abb. 1). Zwar steigt der benötigte Speicherbedarf um ein Vielfaches (längere Datentypen, größere Anzahl von Operationen pro Kanal, redundante Kanäle), und die Bearbeitungszeit nimmt potentiell zu, da die Softwarekanäle tatsächlich nacheinander abgearbeitet werden. Wenn aber die leistungsfähigen IPCs zum Einsatz kommen, spielt dieser vermeintliche Nachteil nur noch eine untergeordnete Rolle.

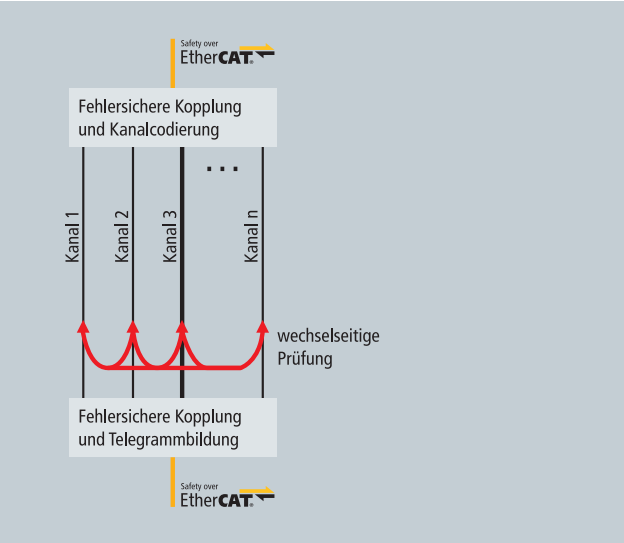

Abb. 1: Architektur des fehlersicheren IPCs

Grundprinzip der fehlersicheren Industrie-PCs

Neben der Redundanz – im Sinne von mehrfacher Ausführung gleicher oder ähnlicher Aufgaben – spielt die Datenredundanz eine immer größere Rolle. Diese kann auch als mathematische Codierung aufgefasst werden: Aus den Originaldaten werden – über eine Abbildungsvorschrift – Bilddaten in einem neuen, viel größeren Wertebereich erzeugt. Wenn die Daten Werte einnehmen, die zwar im Wertebereich liegen, aber nicht der Abbildungsvorschrift entsprechen, gehören sie nicht zum Code. Sie sind daher ungültig und können nur durch Fehler in der Datenhaltung oder -verarbeitung entstanden sein. Die hier verwendeten arithmetischen Codes basieren auf Primzahlen. Sie wurden erstmals ausführlich in [1] beschrieben. Dort werden die Originaldaten in einer vorgelagerten Einheit, wie z. B. in speziellen sicheren Sensoren, mit einer Primzahl multipliziert und mit einem spezifischen Offset beaufschlagt. Wenn nach Abzug des Offsets kein Vielfaches dieser Primzahl vorliegt, ist ein Fehler erkannt worden. In [1] wurde deshalb sogar eine einkanalige Lösung (einkanalige Hardware und einkanalige Software) beschrieben.

Der Wert einer codierten Variablen x_c lässt sich demnach als ein Produkt aus dem originalen Wert x_f und einer Primzahl A darstellen (vgl. [1, 5]):

$$x_c = A \cdot x_f$$

Weiterhin kann der codierte Wert mit einem variablenspezifischen Offset B_x beaufschlagt werden, so dass

$$x_c = A \cdot x_f + B_x$$

gilt. B_x wird auch statische Signatur genannt. Damit liegen im Speicher nicht einfach Vielfache der Primzahl A, sondern verschiedene Werte, von denen, je nach Speicherort, erst ein bestimmter Wert subtrahiert werden muss, bevor ein Vielfaches von A entsteht.

Ein weiterer Offset D_t, der den jeweiligen Prozessorzyklus beinhaltet, die sogenannte dynamische Signatur, vervollständigt die Codierung:

$$x_c = A \cdot x_f + B_x + D_t$$

Mit Hilfe der dynamischen Signatur wird nun auch die fehlerhafte Verwendung veralteter Werte erkennbar. Diese Fehlererkennung wird im Folgenden am Beispiel einer Addition erklärt.

Neben der codierten Variable x_c stehe auch die Variable y_c mit

$$y_c = A \cdot y_f + B_y + D_t$$

zur Verfügung. Die codierte Addition mit dem Ergebnis z_c, mit der entsprechenden Zusammensetzung,

$$z_c = A \cdot z_f + B_z + D_t$$

muss folgendermaßen ausgeführt werden:

$$z_c = x_c + y_c + (B_z - B_x - B_y) - D_t.$$

Beim Wert $(B_z - B_x - B_y)$ handelt es sich um eine Konstante, die Informationen über die beteiligten Variablen und die Operation enthält. Werden bei der Codierung beispielsweise die Offsets B_x, B_y und B_z mit 1093, 5012 und 8913 festgelegt, steht an der entsprechenden Stelle im codierten Programm eine 2808. Die dynamische Signatur muss hier subtrahiert werden, weil jede codierte Variable x_c und y_c ein D_t enthält, die korrekte codierte Summe z_c jedoch wieder nur ein D_t enthalten darf.

Prof. Dr.-Ing. Frank Schiller, Wissenschaftlicher Leiter Safety und Security, Beckhoff Automation

Eine einfache, einkanalige Prüfung der Fehlerfreiheit von z_c testet nur die Gültigkeit des Wertes:

$$(z_c - B_z - D_t) \bmod A == 0?$$

Werteverfälschungen von x_c und y_c, Auslesen von falschen Speicherstellen, Verwenden von veralteten Werten und Fehler des Rechenwerks können dadurch aufgedeckt werden. Die Wahrscheinlichkeit, dass ein Fehler nicht erkannt wird (also die Restfehlerwahrscheinlichkeit), berechnet sich aus $1/A$ (vgl. z. B. [3]). Der Abstand zwischen gültigen Werten beträgt A. Sämtliche dazwischen liegenden Werte werden als fehlerhaft erkannt (vgl. Abb. 2). A muss daher möglichst groß gewählt werden.

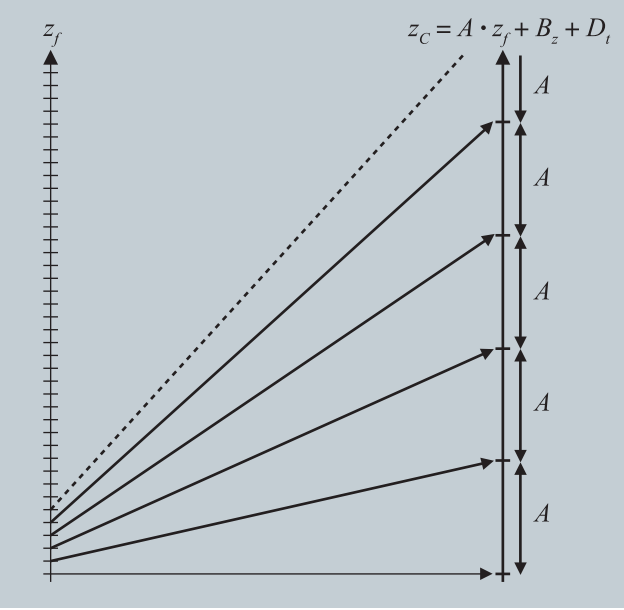

Abb. 2: Wertebereich codierter Variablen

Eine wesentliche Verbesserung der Fehleraufdeckung besteht in der Prüfung mit Hilfe von codierten Daten verschiedener Softwarekanäle [4]. Bei beispielsweise zwei codierten Kanälen lautet die Prüfung:

$$A_2 \cdot (z_{c1} - B_{z1} - D_{t1}) == A_1 \cdot (z_{c2} - B_{z2} - D_{t2})?$$

Die Restfehlerwahrscheinlichkeit sinkt beim Einsatz mehrerer Kanäle beträchtlich.

Vorteile der Nutzung mathematischer Codierungen

Die schnellen Innovationszyklen für die Mikroprozessoren lassen jeden hardwarebasierten Sicherheitsnachweis innerhalb kurzer Zeit hinfällig werden [5]. Daher wären ständig neue Nachweise notwendig. Durch die hier gewählte streng mathematische Grundlage muss der Nachweis der Sicherheit nicht auf den jeweiligen Prozessor und dessen Umgebung Bezug nehmen. Die Eigenschaften des mathematischen Codes bestimmen die Restfehlerwahrscheinlichkeit und, damit verbunden, den SIL nach IEC 61508.

Zudem ist die quasi gleichzeitige Ausführung von sicherheitsrelevanten und nicht-sicherheitsrelevanten Steuerungsprogrammen auf einem IPC möglich, so dass dessen verfügbare Performanz optimal ausgenutzt werden kann.

Fazit

Der heutige Stand der Entwicklung der Industrie-PCs erlaubt die sinnvolle Nutzung von arithmetischen Codes zur Abarbeitung von sicherheitsrelevanten Steuerungsprogrammen. Der erhöhte Speicherbedarf und die große Anzahl der umfangreichen, codierten Operationen stellen für einen IPC kein Problem dar. Die Qualität der Industrie-PCs ist auf einem solchen Niveau, dass ein kontinuierlicher Betrieb gewährleistet ist und die Fehleraufdeckung nicht ständig einen Übergang in den sicheren Betriebszustand (in der Regel einen nicht-produktiven) einleitet.

In der vorgestellten Lösung wird eine erweiterte Codierung genutzt, um eine hohe Fehleraufdeckung in der Datenverarbeitung eines Industrie-PCs zu erreichen. Dem Lösungsansatz wurde in einem Prüfbericht des TÜV SÜD die Zertifizierbarkeit nach IEC 61508 bestätigt. Die notwendige sichere Ankopplung an die Prozess-Peripherie wird durch Safety-over-EtherCAT hergestellt.

Literatur

[1] Forin, P.: Vital Coded Microprocessor Principles and Application for Various Transit Systems. IFAC Control, Computers, Communications, Paris, 1989, S. 79-84.

[2] Wappler, U., Fetzer, C.: Software Encoded Processing: Building Dependable Systems with Commodity Hardware. International Conference on Computer Safety, Reliability and Security, SAFECOMP 2007, LNCS 4680, München, Springer, 2007, S. 356-369.

[3] Ozello, P.: The Coded Microprocessor Certification. International Conference on Computer Safety, Reliability and Security, SAFECOMP 1992, München, Springer, 1992, S. 185-190.

[4] Oh, N., Mitra, S., McCluskey, E.J.: ED4I: Error Detection by Diverse Data and Duplicated Instructions. IEEE Transactions on Computers, 51, 2002, S. 180-199.

[5] Mottok, J., Schiller, F., Völkl, T., Zeitler, T.: A Concept for a Safe Realization of a State Machine in Embedded Automotive Applications. International Conference on Computer Safety, Reliability and Security, SAFECOMP 2007, LNCS 4680, München, Springer, 2007, S. 283-288.