



SONDERAUSGABE ZUM THEMA

Intelligente Prozeßfunktionen in P-NET-Modulen

Die Themen

- 1. Gemeinsame Eigenschaften**
- 2. Prozeßfunktionen für Digital-Eingänge**
- 3. Prozeßfunktionen für Digital-Ausgänge**
- 4. Prozeßfunktionen für Analog-Eingänge**
- 5. Prozeßfunktionen für Analog-Ausgänge**
- 6. PID-Regler**
- 7. Kalkulator für den Programm-Download**
- 8. Sonstige Funktionen**

Warum diese Sonderausgabe ? Ein hoher praktischer und wirtschaftlicher Nutzen kann bei der Ausrüstung einer Anlage mit P-NET-Modulen gewonnen werden, indem man die modulinternen intelligenten Prozeßfunktionen ausnützt. Während bei „nackten“ I/O-Modulen - bisweilen auch als „intelligente Klemmen“ bezeichnet -, wie sie im Feldbusmarkt die Regel sind, lediglich Meßwerte und Signale eingelesen bzw. ausgegeben werden können, unterstützen übliche P-NET-Module die Signalvorverarbeitung vor Ort. Die vorliegende Sonderausgabe widmet sich diesem aktuellen Thema, indem sie die wichtigsten in P-NET-Modulen vorkommenden Prozeßfunktionen in ihrer Anwendung beschreibt. Sofern nicht anders vermerkt, sind die angeführten Registerspezifikationen den „Standardized General Purpose Channel Types“ entnommen und somit herstellerunabhängig.



1. Gemeinsame Eigenschaften

Wie allen P-NET-Anwendern bekannt, werden Modulfunktionen immer über standardmäßig definierte Modulregister angesprochen, die innerhalb eines P-NET-Telegramms adressiert werden können. Jeweils alle einem bestimmten Ein-/Ausgang zugehörigen Register sind in einem Channel zusammengefaßt. Gleiches gilt für alle Register, die einen gemeinsamen Funktionsblock betreffen, wie PID-Regler etc. Grundsätzlich werden alle Registerinhalte, welche die Konfiguration der Funktionen betreffen, nichtflüchtig im Modul abgespeichert - in der Regel in einem EEPROM - und sind somit nach Spannungsausfall unverändert vorhanden. Weiterhin wird in manchen Anwendungen die Eigenschaft von P-NET-Modulen ausgenutzt, auch ohne aktuellen P-NET-Anschluß nach anfänglicher Konfiguration autark beispielsweise als Regler arbeiten zu können. P-NET wird in diesem Fall ausschließlich zur (nichtflüchtigen !) Konfiguration benutzt.

Im folgenden wird bei der Nummerierung einzelner Register - der sogenannten Software Numbers SWNo - immer die Schreibweise $x\langle\text{reg}\rangle$ mit $\langle\text{reg}\rangle:0\dots F$ verwendet. x steht dabei für die modulspezifische Channelnummer, während $\langle\text{reg}\rangle$ eines von maximal 16 Registern eines Channels kennzeichnet. Die *symbolischen Namen* der Register werden kursiv wiedergegeben.

Allen die jeweiligen Prozeßfunktionen enthaltenden Channels gemeinsam ist die strukturelle Anordnung der Register. Dabei befindet sich im Register $x0$ immer der primär interessierende Prozeßwert wie Status des Digital-Eingangs, analoger Eingangswert etc. Im Register *ChConfig* ($x9$) werden immer die grundlegenden Konfigurationen vorgenommen. Register *Maintenance* (xD) dient der Ablage von Wartungsinformationen des angeschlossenen Sensors bzw. Aktors. Die Register *ChType* (xE) und *ChError* (xF) stehen für die Charakterisierung des Channel-Typs sowie für Fehlermeldungen. Auf die Bedeutung der einzelnen Register des allen P-NET-Modulen gemeinsamen Service Channels wird hier nicht eingegangen.

Nicht alle der im folgenden aufgeführten Prozeßfunktionen sind von den P-NET-Herstellern jeweils zu realisieren. Im Standard wird hierbei zwischen Pflichtfunktionen und optionalen Funktionen unterschieden. Insbesondere im Bereich der universellen Ein-/Ausgabemodule sind jedoch in der Regel alle Funktionen realisiert.



2. Prozeßfunktionen für Digital-Eingänge

Digital-Ein- und Ausgänge werden bei P-NET-Modulen häufig zusammen realisiert, so daß ein Klemmenanschluß je nach Konfiguration Ein- oder Ausgang sein kann. Bisweilen sind auch Kombinationen vorzufinden, beim dem vier kombinierte Ein-/Ausgänge sowie zwei reine Ausgänge vorhanden sind. In jedem Fall werden die dazugehörigen Prozeßfunktionen über einen kombinierten Digital-I/O-Channel gesteuert.

Wird das Konfigurationsregister *ChConfig* ($x9$) auf Input Only konfiguriert, so kann der Status des Eingangs direkt aus dem InFlag des *FlagReg* ($x0$) über P-NET ausgelesen werden. Zusätzliche Prozeßfunktionen, die der zugehörige Channel bietet, sind:

- Pulszählung, wobei auf 0-1-Flanken getriggert wird, über *Counter* ($x2$)
- Summation aller Zeiten, während denen der Eingang auf 1 steht, in *OperatingTime* ($x4$)

Die hier genannten zwei Register lassen sich, wie alle anderen unten aufgeführten Register, jederzeit im Betrieb löschen, sei es über P-NET vom Master verursacht oder beispielsweise über ein Programm im Kalkulator (vgl. 7. Abschnitt).

Zusätzlich kann für Simulationszwecke das Flag Input simulation im *ChConfig* gesetzt werden, wonach das InFlag direkt gesetzt werden kann und nicht den Zustand des von außen ggf. angeschalteten Signals übernimmt.

Ein Digital-Eingang kann alternativ aber auch als Feedbacksignaleingang für einen Digital-Ausgang dienen, wie im nächsten Abschnitt noch dargestellt wird.

3.

Prozeßfunktionen für Digital-Ausgänge

In *ChConfig* (x9) können neben der einfachsten Ausgabeart (genannt Output), bei der durch Setzen des Flags *OutFlag* in *FlagReg* (x0) direkt der Zustand am Digital-Ausgang festgelegt wird, folgende erweiterte Prozeßfunktionen konfiguriert werden:

- One shot output: Beim Setzen des Flags *Control* in *FlagReg* (x0) wird der Ausgang zunächst gesetzt, um nach einer in *OutPreset* (x8) definierbaren Zeit wieder zurückgesetzt zu werden. Diese Zeit kann dynamisch durch Beschreiben des Registers *OutTimer* (x1) verändert werden, da dieses Register für die Zeitüberwachung zuständig ist, indem es nach anfänglichem Setzen auf den Inhalt von *OutPreset* die Zeit herabzählt, um bei Null schließlich das Zurücksetzen des Ausgangs zu veranlassen.
- 50% Duty-Cycle output: Nach Setzen des Flags *Control* ändert der Ausgang periodisch seinen Status, wobei sowohl der Ein- als auch der Auszustand jeweils die in *OutPreset* spezifizierte Zeit umfassen. Einfach gesehen handelt es sich also um einen frequenzprogrammierbaren Rechteckgenerator.
- Output controlled by other channel: Ein Setzen des Flags *Control* - dieses ist nach einem Modulreset zunächst immer gesetzt ! - bewirkt, daß der Digital-Ausgang direkt von anderen Prozeßfunktionen wie zum Beispiel dem Kalkulator (siehe Abschnitt 7) gesetzt wird. Die Gesamtfunktionalität hängt somit von der Struktur dieser Prozeßfunktionen ab.

Unabhängig von der Art und Weise, wie ein Digital-Ausgang angesteuert wird, sind folgende zusätzliche Prozeßfunktionen für Überwachungszwecke vorhanden:

- Der Strom, der durch den Ausgang fließt, wird modulintern gemessen und in *OutCurrent* (x3) abgelegt. Entsprechende Überschreitungen gegenüber einem in *MaxCurrent* (xB) angebbaren Wert bzw. Unterschreitungen bezüglich *MinCurrent* (xA) generieren das Setzen von Fehlerflags.
- Ein bzw. zwei Digital-Eingänge können als Feedback-Eingänge für die Digital-Ausgabe dienen. Dazu werden die entsprechenden Channelnummern der Eingänge in *ChConfig* angegeben. Wenn nach Ablauf einer in *FBTimer* (x6) vorgebbaren Zeit die Feedback-Eingänge noch nicht eine dem ausgegebenen Signalzustand entsprechenden Zustand erreicht haben - Zustand erster Eingang = Ausgabezustand, Zustand zweiter Eingang = inverser Ausgabezustand -, wird ein Fehlerflag gesetzt. Diese Funktion ist beispielsweise für Stellglieder, die über Endlagenschalter kontrolliert werden, sinnvoll.

4.

Prozeßfunktionen für Analog-Eingänge

Das zum zugehörigen Channel gehörige *ChConfig* (x9) erlaubt neben der Auswahl des Eingangssignaltyps (Pt-100, 0/4..20 mA, 0-100 mV) vor allem die Spezifikation eines

- Eingangsfilters mit Zeitkonstanten von 1 ms bis 50 s.

Der vom Modul gemessene Analogwert wird zunächst

- entsprechend den Parametern in *FullScale* (xB) und *ZeroPoint* (xC) skaliert

und danach in *AnalogIn* (x0) bereitgestellt. Dies bedeutet, daß beispielsweise ein an das Modul angeschlossener Drucksensor mit 0...20 mA-Ausgang P-NET-seitig direkt in Druckeinheiten (z.B. bar) abgefragt wird. Der Meßwert wird dabei immer als Gleitpunktzahl dargestellt. Lediglich bei Pt-100-Betrieb wird *FullScale* nicht ausgewertet.

- *HighLevel* (x7) und *LowLevel* (x8) erlauben eine Grenzwertüberwachung, die ggf. zu entsprechendem Setzen von Fehlerflags führt.



5. ●

Prozeßfunktionen für Analog-Ausgänge

Typischerweise stellen P-NET-Module hier 0/4...20 mA-Signale zur Verfügung. Im allgemeinen lassen sich im Register *ChConfig* (x9) dann die vier Arbeitsbereiche 4-20 mA, 20-4 mA, 0-20 mA, 20-0 mA konfigurieren, wobei viele Module nur 4 mA als kleinsten Strom erlauben. Den ausgegebenen Strom in Prozent des Arbeitsbereiches gibt das Register *AnalogOut* (x0) wieder. Der eigentliche Sollwert wird in *Setpoint* (x6) eingeschrieben und zwar zunächst als beliebige Prozeßgröße, die dann über die Skalierung in ein Stromsignal umgerechnet wird.

Folgende Prozeßfunktionen sind üblich:

- *FullScale* (xB) und *ZeroPoint* (xC) dienen der Skalierung nach Eingabe eines Sollwertes.
- Kontrollmessung des ausgegebenen Stromes: Wird ein niedrigerer Strom als der gewünschte Sollwert festgestellt, so wird ein Fehlerflag gesetzt.
- *HighLevel* (x7) und *LowLevel* (x8) erlauben eine Grenzwertüberwachung, die ggf. zu entsprechendem Setzen von Fehlerflags führt. Dies ist insbesondere dann vorteilhaft, wenn der Sollwert von anderen Channels aus gesetzt wird, um beispielsweise einen bestimmten Arbeitsbereich des Stellgliedes nicht zu verlassen.

Das Setzen des Sollwertes von einem anderen Channel aus wird dadurch aktiviert, daß in *ChConfig* die SWNo eines Registers definiert wird, dessen Inhalt (der vom Typ Real sein muß !) den Sollwert bestimmen soll.



6. ●

PID-Regler

In den meisten Modulen, die über mindestens einen Analog-Eingang und einen Analog-Ausgang verfügen, findet sich auch diese Prozeßfunktion. In einem eigenen Channel können dazu mehrere Konfigurationen vorgenommen werden:

ChConfig (x9) legt die Reglerart - P/PD oder PID - sowie normalen („Heating“) oder inversen („Cooling“) Wirksinn fest. Für Istwert und Sollwert stehen die Register *Signal* (x1) sowie *Setpoint* (x6) zur Verfügung. An Stelle der manuellen Eingabe über P-NET können diese auch direkt von Registern anderer Channels aus angesteuert werden, wozu in *ChConfig* deren SWNo eingetragen werden muß. Eine häufig anzutreffende Kombination besteht darin, daß der Sollwert über P-NET von einem Master eingeschrieben wird, während der Istwert von einem Analogeingang desselben Moduls stammt.

Während diese beiden Eingangssignale des Reglers als Prozeßgrößen mit beliebigen Werten interpretiert werden, wird der Reglerausgangswert in *PIDOutput* (x0) als Prozentwert zwischen 0 % und 100 % abgelegt. Dies ist kompatibel zur Darstellung beim Analog-Ausgang (vgl. Abschnitt 5).

Die drei Reglerparameter eines PID-Reglers werden in den Registern *Xp* (xB), *Ti* (xC) sowie *Td* (xD) definiert. Entsprechend der obigen Prozentdarstellung muß in *Xp* der Wert für die Höhe eines Eingangssprunges des Sollwertes eingetragen werden, der zu einem *PIDOutput* von 100 % führen soll. Dies ist folglich nicht identisch mit der sonst üblichen Eingabe eines Verstärkungsfaktors, bewirkt aber genau dasselbe. *Ti* und *Td* werden mit der Einheit „s“ definiert und entsprechen den aus der Regelungstechnik bekannten Begriffen Nachstellzeit und Vorhaltezeit.

Durch Setzen des booleschen Registers *Manual* (x8) kann auf Handbetrieb umgeschaltet werden. Hierbei wird der Reglerausgang direkt durch Einschreiben in *PIDOutput* bestimmt.

Der Regler überwacht unter anderem, ob *PIDOutput* innerhalb der zulässigen Grenzen angesteuert wird und setzt notfalls entsprechende Fehlerflags.



Kalkulator für den Programm-Download

Der Kalkulator stellt von außen betrachtet einen programmierbaren Prozessor dar, der mit einer eigenen Assemblersprache programmiert werden kann. Strukturell besteht der Kalkulator aus zwei Akkumulatoren - einem für Gleitpunkt- und einem für boolesche Operationen -, zwei sogenannten Channel Pointern, zwei Indexregistern und einem Bitindexregister. Der sehr einfach gehaltene Befehlssatz umfaßt Move-, Compare-, Jump-, logische sowie arithmetische Operationen. Dabei kann auf beliebige Register aller Channel im Modul zugegriffen werden. Damit können beliebige neue Prozeßfunktionen kreiert werden, die per Download über P-NET in das Modul nichtflüchtig eingeschrieben werden können. Als Beispiele seien komplexe Regelalgorithmen (Fuzzy Control etc.) oder Ablaufsteuerungen genannt.

Unterschiede gibt es in der Struktur des zugehörigen Channels bzw. der Art des Programm-Downloads. In Modulen, die schon länger am Markt sind, findet man noch die ursprüngliche Konzeption eines Calculator Channels. In diesem dient das Register *ProgramStep* (xD), dessen Länge von Modul zu Modul variiert, der Aufnahme des Programms. Zusätzlich stehen in diesem Channel zahlreiche Universalregister sowie ein *UserTimer* (c8) zur Verfügung. Drei sogenannte LookUp-Register erlauben die Spezifikation nichtlinearer Kurven - z.B. Sensorkennlinien - durch Eingabe fester Wertepaare; zwischen diesen Wertepaaren interpoliert das Modul selbständig. Über *RunEnable* (x9) wird das Kalkulator-Programm gestartet. Das Fehlerregister *ChError* (xF) enthält im Gegensatz zu allen anderen Channel-Typen keine vordefinierten Fehlerflags; deren Bedeutung bestimmt allein das Programm, welches sie auch setzt.

Bei neueren Modulen ist eine MMS-kompatible Methode des Downloads und der Kontrolle des Programms implementiert. Dabei können von der Struktur des zuständigen sogenannten Program Channels her gleichzeitig mehrere unterschiedliche Programme im Modul, organisiert in einer Library, geführt werden. Heutige Module unterstützen allerdings nur die Verwaltung eines Programms. Der Download eines Programms erfolgt über das Register *IDAndCode* (x8), wobei dem eigentlichen Kalkulator-Code beim Download ein Header vorangestellt ist, der sich in identischer Form nach dem Download im Register *ProgramID* (x2) wiederfindet. Vor dem Download sind etliche andere Register wie *MemoryInfo* (x7) zu überwachen. Das Programm kann über *ProgramControl* (x0) in seinem Ablauf kontrolliert werden. Unter anderem sind hierbei Kommandos wie Start, Stop, Resume, Reset und Kill möglich. Module mit diesem neueren Program Channel weisen auch über einen zusätzlichen Data Channel auf, der unter anderem über mehrere Universalregister, zwei Timer und zwei LookUp-Register verfügt. Zusätzlich ist ein in seiner Funktion nicht fest definiertes *ChConfig* vorhanden, so daß dieser Data Channel in Abhängigkeit der Programmfunktion einem gewöhnlichen Channel nachgebildet werden kann.

Die Programme selbst werden mit einem am PC ablaufenden Calculator Assembler geschrieben und beispielsweise auch von dort über P-NET in das Modul geladen.

Die letztgenannte Variante des Programm-Downloads wird nicht nur bei Slave-Modulen angewandt sondern beispielsweise auch beim neuen Controller PD 5000. Dort dient er dem Download des Betriebssystems sowie von in Process-Pascal geschriebenen Controller-Programmen.



Sonstige Funktionen

Neben den bisher genannten häufig in P-NET-Modulen vorhandenen Prozeßfunktionen existieren insbesondere bei Modulen für spezielle Aufgaben hochentwickelte spezielle Prozeßfunktionen. Beispielhaft seien vier Funktionen kurz angesprochen:

- Wägeeingänge für DMS-Brücken realisieren unter anderem Funktionen wie Tarierung, Massenflußberechnung, Grenzwertüberwachung, Skalierung und Konfiguration der Wägegenauigkeit sowie der Ableseperiode, über die jeweils gemittelt wird. Befüllvorgänge z.B. über Förderbänder können über die Betriebsart Belt Weight automatisiert werden.
- Sogenannte Puls-Prozessoren in Modulen mit Digital-Ein-/Ausgängen dienen der äußerst schnellen Verknüpfung und Verarbeitung der digitalen Ein- und Ausgänge des Moduls. Damit können Funktionen wie Signalauswertung von Encodern, Pulserzeugung für Schrittmotoren, schnelle Pulszählung (z.B. bis zu 100 kHz) etc. effizient implementiert werden. Zur Programmierung wird der am PC ablaufende Pulse Processor Program Editor verwendet
- RS-232-Konverter enthalten sogenannte Communication Channels, die neben einer Konfiguration der Schnittstellenparameter insbesondere Eingangs- und Ausgangs-Puffer beinhalten. Dieser Channel wird bei allen neuen Modulen - ob Slave oder Master - standardmäßig für serielle Kommunikationsports herangezogen. So werden die P-NET- und RS-232-Ports des PD 5000 über exakt dieselbe Channelstruktur bedient.
- Analoges gilt für die Printer Channel bei P-NET-fähigen Druckern.



Impressum

Herausgeber:
Prof. Dr.-Ing. Jörg Böttcher c/o b-plus Meßsysteme GmbH
im Auftrag der
International P-NET User Organization ApS, Silkeborg
Anschrift:
Haslacher Str. 93, D-94469 Deggendorf
Tel.: +49 991 340 897, Fax: +49 991 340 447
Email: b-plus@t-online.de
P-NET_{lokal} wird ca. alle 2-3 Monate kostenlos an die Mitglieder der International P-NET User Organization ApS in Deutschland, Österreich und der Schweiz verteilt. Alle anderen Mitglieder können P-NET_{lokal} kostenlos auf Anfrage erhalten.

