

Schrittmotoren am PC mit Ihren eigenen Anwendungsprogrammen

X4Step

Die Funktionen-Sammlung für C-Programme

Die Funktionen-Sammlung *X4Step* übernimmt die Ansteuerung von 1-4 Schrittmotoren in beliebigen C-Programmen. Der Realisierung von CNC-Aufgaben nach Ihren Vorstellungen steht somit nichts mehr im Weg. Ob Roboteranwendungen, Fräs- und Bohrtische, Dispensieranlagen, Handlingsysteme oder Vorschübe, alles ist in Verbindung mit *X4Step* möglich.

was *X4Step* leistet

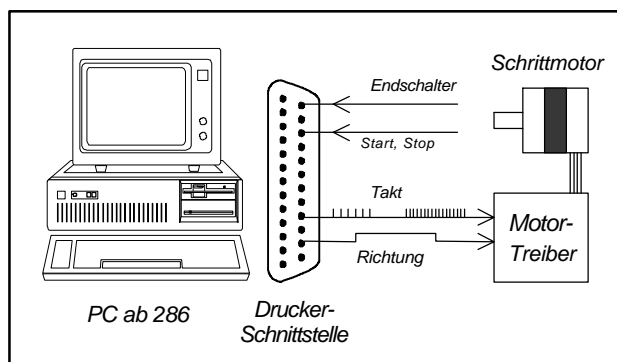
- Ansteuerung von max. 4 Schrittmotoren direkt über die parallele Schnittstelle
- läuft auf IBM kompatiblen PCs ab 286 und unter MS-DOS ab Version 5.0
- Motoren in allen Leistungsklassen möglich
- Schrittausgabe im Hintergrund
- Schrittfrequenzen von 10 Hz bis 20 kHz
- Rampenzeiten von 0 bis 1000 ms
- Wege bis 16 Mio. Schritten pro Positionierungsvorgang oder Endlosfahrt
- aktuell zurückgelegte Schritte jederzeit auslesbar
- alle Achsen fahren linear interpoliert
- unterstützt Turbo C, Borland C, Power C und Microsoft C
- 1 ms Timer verfügbar
- Einlesen von 5 Schnittstellen-Eingängen als externe Signale, z.B. START, STOP oder Endschalter
- Signale AKTIV und BOOST und zwei weitere Ausgänge zur freien Verwendung

keine Zusatz-Hardware nötig

X4Step steuert die Schrittmotor-Endstufen direkt über eine frei wählbare parallele Schnittstelle des PCs mit den nötigen Takt- und Richtungssignalen an.

Eine zusätzliche intelligente Steuerung oder Prozessorkarte ist also nicht mehr notwendig.

Mit den Signalen der Schnittstelle lassen sich alle gängigen Schrittmotor-Endstufen ansteuern, deren Eingänge mit +5 Volt (TTL-Pegel) betrieben werden können.



einfache Handhabung

Die Benutzung von *X4Step* ist nicht komplizierter als der Gebrauch von C-Funktionen aus der Standard-Bibliothek. *X4Step* wird zusammen mit Ihren anderen Projektdateien zu einem lauffähigen Programm gebunden.

Lieferumfang

Im Lieferumfang von *X4Step* befindet sich neben dem umfangreichen Handbuch ein Testprogramm im C-Quellcode, in dem die Verwendung aller Funktionen demonstriert wird.

Beispiele

Zur Veranschaulichung der einfachen Handhabung und des großen Leistungsumfangs dienen die folgenden Beispiele voll lauffähiger Programme.

Achsen 1, 2 und 3 interpoliert vor- und zurückfahren. Achse mit längstem Weg soll mit 500 Hz und 10 ms Rampenzeit fahren.

```
main()
{
  X4Init(1); // X4Step mit LPT1 starten
  X4Load(1000,400,100,0,500,10); // Achsen laden, Achsel fährt 500Hz in 10ms
  X4Start(); // interpolierte Fahrt starten
  while (X4GetState()) ; // warten, bis alle Achsen wieder stehen
  X4Load(-1000,-400,-100,0,500,10); // Achsen mit zweiten Wegen laden
  X4Start(); // interpolierte Fahrt starten
  while (X4GetState()) ; // warten, bis alle Achsen wieder stehen
  X4ReInit(); // X4Step und Programm beenden
}
```

Achse 3 mit 8500 Hz und 200 ms Rampenzeit endlos in positive Richtung fahren. Nach 2,5 Sekunden Fahrt abbrechen.

```
main()
{
  X4Init(1); // X4Step mit LPT1 starten
  X4MovPos(3,8500,200); // Achse 3 endlos in pos. Richtung fahren
  X4ClearTimer(); // lms Timer löschen
  while (X4GetTimer() < 2500) ; // 2,5 Sekunden warten
  X4Brake(); // Fahrt mit Rampe beenden
  while (X4GetState()) ; // warten, bis Achse 3 wieder steht
  X4ReInit(); // X4Step und Programm beenden
}
```

Achse 1 mit 10 kHz und 100 ms Rampenzeit endlos in negative Richtung fahren und stoppen, wenn am Eingang Pin 15 der Pegel HIGH wird.

```
main()
{
  X4Init(1); // X4Step mit LPT1 starten
  X4MoveNeg(1,10000,100); // Achse 1 endlos in neg. Richtung fahren
  while ((X4GetIn() & 0x01) == 0) ; // warten, bis Pin 15 HIGH Pegel hat
  X4Brake(); // Fahrt mit Rampe beenden
  while (X4GetState()) ; // warten, bis Achse 1 wieder steht
  X4ReInit(); // X4Step und Programm beenden
}
```

Achse 4 mit 5 kHz und 50 ms Rampenzeit 50000 Schritte fahren. Nach 10000 Schritten den Ausgang 1 auf HIGH und nach 30000 Schritten wieder auf LOW schalten.

```
main()
{
  X4Init(1); // X4Step mit LPT1 starten
  X4Load(0,0,0,50000,5000,50); // Achse 4 mit Weg, Frequenz, Rampe laden
  X4Start(); // Positionierung starten
  while (X4GetSteps(4) < 10000) ; // warten, bis 10000 Schritte erreicht
  X4SetOut(SIGNAL1ON); // Ausgang 1 setzen
  while (X4GetSteps(4) < 30000) ; // warten, bis 30000 Schritte erreicht
  X4SetOut(SIGNAL1OFF); // Ausgang 1 wieder zurücksetzen
  while (X4GetState()) ; // warten, bis Achse 4 wieder steht
  X4ReInit(); // X4Step und Programm beenden
}
```

Die Unterstützung von anderen Compilern und Compiler-Versionen ist möglich und kann auf Anfrage angeboten werden.

