

Messen Steuern Regeln
Leistungstest von Prozeßrechnersystemen
Zeitmessungen

Testfunktion: Interrupt-Behandlung mit Programm-Start

DIN
19 242
 Teil 7

Measurement and control; performance tests on process computer systems, time measurement; function tested: interrupt handling with program start

Für den Anwendungsbereich dieser Norm bestehen keine entsprechenden regionalen oder internationalen Normen.

Inhalt

	Seite		Seite
1 Anwendungsbereich und Zweck	1	6 Testumgebung	2
2 Allgemeines	1	7 Testprotokoll	2
3 Testfunktion	1	Anhang A	3
4 Testvorgang	1	Anhang B	17
4.1 Vorbereitung	1	Anhang C	24
4.2 Durchführung	2	Zitierte Normen	31
5 Testergebnisse	2	Erläuterungen	31

1 Anwendungsbereich und Zweck

Mit dieser Testfunktion soll die Bearbeitungszeit für einen zyklischen, zeitgesteuerten Interrupt mit Task-Start/-Ende nach dem Testverfahren DIN 19242 Teil 1 gemessen werden.

Um die Programmierung der Testfunktionen zu erleichtern, sind in den Anhängen A, B und C Programmbeispiele in FORTRAN, PEARL und PASCAL beigelegt, die mit geringfügigen Änderungen auf allen Rechnern ablaufen können. Diese Beispiele sind auch beim Beuth Verlag auf Datenträger (z. B. Magnetband) erhältlich.

2 Allgemeines

Aus Aufwandgründen wird in dieser Testfunktion der Uhr-Interrupt verwendet. Dieser wird softwaremäßig verwaltet und führt parametergesteuert zu einem zyklischen Task-Start.

3 Testfunktion

Für n Tasks wird je eine Beauftragung an das Betriebssystem zum zyklischen Start gegeben.

Die Tasks müssen in getrennten Laufbereichen parallel ablaufen und in einem globalen Kommunikationsbereich, der allen Programmen zugänglich ist, taskspezifische Zähler Z_n hochzählen können (Beispiele siehe Anhänge).

Diese Testfunktion umfaßt je Task:

- eine einmalige Beauftragung für den zyklischen Start
- Z_n -mal Unterbrechung des laufenden Hintergrundprogramms bzw. einer Task

- Z_n -mal Task-Start
- Z_n -mal Task-Ausführungen (Zähler erhöhen)
- Z_n -mal Task-Ende
- Z_n -mal Fortsetzung des unterbrochenen Hintergrundprogramms oder einer anderen Task

- ein einmaliges Beenden der Beauftragung

Durch die eingestellte Dauer des Hintergrundprogramms und die gewählten Zykluszeiten wird die Anzahl der Task-Starts gesteuert. Die Zykluszeiten müssen sinnvoll gewählt werden.

4 Testprogramm

4.1 Vorbereitung

Es müssen n Tasks eingerichtet und lauffähig in den Hauptspeicher mit höherer Priorität als der des Hintergrundprogramms geladen sein.

Es ist möglich, die Prioritäten der Tasks untereinander zu variieren.

Ein globaler Kommunikationsbereich muß für die Zähler der n Tasks angelegt sein. Im Vorlauf des Hintergrundprogramms werden die Parameter

- Identifikation der zu startenden Task
 - Zykluszeit je Task
- eingelassen.

Die Laufzeit des Hintergrundprogramms muß so eingestellt sein, daß für jede Task eine genügend große Anzahl Starts möglich ist.

Fortsetzung Seite 2 bis 31

Deutsche Elektrotechnische Kommission im DIN und VDE (DKE)

Jede Art der Vervielfältigung, auch auszugsweise, nur mit Genehmigung des DIN Deutsches Institut für Normung e. V., Berlin, gestattet.

4.2 Durchführung

Im Hintergrundprogramm werden direkt nach der Abfrage der Anfangszeit und vor der festgelegten Funktion die Beauftragungen für n Tasks gegeben und die Beendigungen dieser Beauftragungen nach der festliegenden Funktionsfolge und direkt vor der Abfrage der Endzeit durchgeführt. Jede Task erhöht nach ihrem Start ihren eigenen Zähler im Kommunikationsbereich und beendet sich.

Im Nachlauf des Hintergrundprogrammes wird der Zählerstand je Task ausgedrückt.

Um die Zeiten für die Testfunktion zu ermitteln, muß der Testvorgang in zwei Schritten durchgeführt werden:

Schritt 1: Testlauf des Hintergrundprogrammes ohne Beauftragungen an das Betriebssystem

Schritt 2: Testlauf des Hintergrundprogrammes mit Beauftragungen an das Betriebssystem.

5 Testergebnisse

Z_n Anzahl der durchgeführten Starts der Task n

andere Formelzeichen: siehe DIN 19 242 Teil 1

Z ist die Summe ($\sum Z_n$) der Zählerstände und damit die Anzahl der ausgeführten Task-Starts usw. (siehe Abschnitt 3)).

Belegungszeit des Zentralprozessors (t_{ZP})

$$t_{ZP} = [(t_{EH2} - t_{AH2}) \cdot (t_{EH1} - t_{AH1})] / Z$$

6 Testumgebung

Grundkonfiguration nach DIN 19 242 Teil 1/01.87, Abschnitt 7.

7 Testprotokoll

Zum Testergebnis muß die Testumgebung (siehe DIN 19 242 Teil 1) im Protokoll hinreichend genau beschrieben werden, damit der Test reproduzier- und vergleichbar wird.

Das Testprotokoll muß eine Aussage über

- die verwendeten Tasks
- die Zykluszeiten
- die Prioritäten
- die Zählerstände

enthalten.

Anhang A

FORTRAN-Programm

```

C*****
C****
C**** PROGRAMMIER-SPRACHE:...FORTRAN IV
C****
C****
C****
C****
C****
C****
C****
C*****
C****
C**** DER TESTRAHMEN BESTEHT NACH DIN 19242 TEIL 1 AUS DEN
C**** PROGRAMMABSCHNITTEN:
C****
C****      1). VORSPANN (EINLESEN VON PARAMETERN, INITIALISIEREN,
C****              STARTEN)
C****      2). BEREICH FUER TESTFUNKTION BZW. TESTHILFSFUNKTION
C****              (FUNKTION, FUNKTIONSFOLGE, ANWENDERPROFIL)
C****      3). NACHSPANN (BEENDEN, AUSWERTEN, PROTOKOLLIEREN)
C****
C**** DER TESTRAHMEN KANN VOM ANWENDER ERWEITERT WERDEN,
C**** MUSS ABER FUER EINE TESTSERIE UNVERAENDERT BLEIBEN.
C****
C**** DIESER TEIL 7 IST IDENTISCH MIT TEIL 1.
C**** TEIL 7 IST ERWEITERT UM DIE SYSTEM-AUFRUFE
C****      CALL CYCLE UND CALL CANCL
C****      ZUM ZYKLISCHEN STARTEN BZW. BEENDEN DES ZYKLUS.
C****
C**** DIE TASKS SIND SELBSTAENDIGE UEBERSETZUNGS-EINHEITEN
C**** ANGEFUEGT.
C****
C*****
C*****
C****      == SYSTEM - ROUTINEN ==
C*****
C****
C****      TIME:   TAGESZEIT (STUNDEN:MINUTEN:SEKUNDEN)
C****      SECNDS: AUSFUEHRUNGSZEIT (SEKUNDEN)
C****
C****      CYCLE:  BEAUFTRAGUNG ZUM START VON TASK-STARTS
C****      CANCL:  STORNIEREN DER BEAUFTRAGUNG
C****
C*****
C*****

```

```

C*****
C****      === D A T E N - D E K L A R A T I O N E N ===      ****
C*****
C****      ****
C**** PN = PROGRAMM-IDENTIFIKATIONS-NUMMER (01 ... 99)      ****
C****      FESTE ZUORDNUNG ZU JEDEM TESTPROGRAMM (KONSTANTE)      ****
C**** PI = PROGRAMM-INDEX ZUR STEUERUNG VON PROGRAMMABSCHNITTEN      ****
C****      (PI = -1: PROGRAMM-ABBRUCH)      ****
C****      (PI = 0: OHNE FUNKTIONSTEST)      ****
C****      (PI = 1: MIT FUNKTIONSTEST)      ****
C**** NI = ANZ. INNERE SCHLEIFE (1 ... 32760)      ****
C**** NA = ANZ. AEUSSERE SCHLEIFE (1 ... 32760)      ****
C**** EIN = LOG.-NR.-EINGABE-GERAET (TERMINAL)      ****
C**** AUS = LOG.-NR.-AUSGABE-GERAET (TERMINAL)      ****
C**** PRINT = LOG.-NR.-PROTOKOLL-GERAET (DRUCKER/FERNSCHREIBER)      ****
C**** TA = ZEIT-ANFANG (STD:MIN:SEK)      ****
C**** TE = ZEIT-ENDE (STD:MIN:SEK)      ****
C**** TD = ZEIT-DIFFERENZ (SEK)      ****
C**** IR = VARIABLE FUER DIE DIVISION (TESTHILFS-FUNKTION)      ****
C****      ****
C**** TANZ = ANZAHL DER TASKS (VORBESETZUNG = 4)      ****
C**** TAKT = AKTUELLE TASK (1 ... 4)      ****
C**** NT1 = ZAEHLER FUER DIE TASK1 IM COMMON-BEREICH      ****
C**** NT2 = ZAEHLER FUER DIE TASK2 IM COMMON-BEREICH      ****
C**** NT3 = ZAEHLER FUER DIE TASK3 IM COMMON-BEREICH      ****
C**** NT4 = ZAEHLER FUER DIE TASK4 IM COMMON-BEREICH      ****
C**** PNR1 = PROZESSNUMMER FUER DIE TASK1 (1 ... 99)      ****
C**** PNR2 = PROZESSNUMMER FUER DIE TASK2 (1 ... 99)      ****
C**** PNR3 = PROZESSNUMMER FUER DIE TASK3 (1 ... 99)      ****
C**** PNR4 = PROZESSNUMMER FUER DIE TASK4 (1 ... 99)      ****
C**** PNRX = EINGABE-FELD FUER PNR1 ... PNR4      ****
C**** CYCL1 = START-ZYKLUS FUER DIE TASK1 (1 ... 999)      ****
C**** CYCL2 = START-ZYKLUS FUER DIE TASK2 (1 ... 999)      ****
C**** CYCL3 = START-ZYKLUS FUER DIE TASK3 (1 ... 999)      ****
C**** CYCL4 = START-ZYKLUS FUER DIE TASK4 (1 ... 999)      ****
C**** CYCLX = EINGABE-FELD FUER CYCL1 ... CYCL4      ****
C**** EINH1 = ZEITEINHEIT FUER TASK1 (1 ... 9)      ****
C**** EINH2 = ZEITEINHEIT FUER TASK2 (1 ... 9)      ****
C**** EINH3 = ZEITEINHEIT FUER TASK3 (1 ... 9)      ****
C**** EINH4 = ZEITEINHEIT FUER TASK4 (1 ... 9)      ****
C**** EINHx = EINGABE-FELD FUER EINH1 ... EINH4      ****
C**** ST1 = STATUS-VARIABLE FUER AUFRUF TASK1 (1 = FEHLERFREI)      ****
C**** ST2 = STATUS-VARIABLE FUER AUFRUF TASK2 (1 = FEHLERFREI)      ****
C**** ST3 = STATUS-VARIABLE FUER AUFRUF TASK3 (1 = FEHLERFREI)      ****
C**** ST4 = STATUS-VARIABLE FUER AUFRUF TASK4 (1 = FEHLERFREI)      ****
C**** DUMMY = HILFSGROESSE      ****
C****      ****
C*****

```

```
C*****
C****
COMMON /TASK1/NT1(2)
COMMON /TASK2/NT2(2)
COMMON /TASK3/NT3(2)
COMMON /TASK4/NT4(2)

C****
INTEGER PN,PI,NI,NA,EIN,AUS,PRINT
INTEGER TA(4),TE(4)
INTEGER IR(3)
INTEGER DUMMY
REAL TD
INTEGER TANZ,TAKT,PNRX(4),CYCLX(4),EINHX(4)
INTEGER PNR1,CYCL1,EINH1,ST1
INTEGER PNR2,CYCL2,EINH2,ST2
INTEGER PNR3,CYCL3,EINH3,ST3
INTEGER PNR4,CYCL4,EINH4,ST4
DATA EIN/5/,AUS/6/,PRINT/8/
DATA PN/7/
DATA TANZ/4/
DATA PNR1/0/,ST1/1/
DATA PNR2/0/,ST2/1/
DATA PNR3/0/,ST3/1/
DATA PNR4/0/,ST4/1/

C****
C*****
```

```

C*****
C**** ABSCHNITT 1: === V O R S P A N N === ****
C*****
C****
C****          EINLESEN DER PARAMETER ****
C****          ****
100 WRITE(AUS,101) PN
101 FORMAT(' PN=',I2,' GIB PARAMETER:',
1      /,' =====',
2      /,' PI = -1: PROGRAMM-ABBRUCH',
3      /,'      0: OHNE FUNKTIONSTEST',
4      /,'      1: MIT FUNKTIONSTEST',
5      /,' NA = ANZAHL AEUSSERE SCHLEIFE (1 ... 32760)',
6      /,' NI = ANZAHL INNERE SCHLEIFE (1 ... 32760)',
7      /,' =====',
8      /,' PI      NA      NI',
9      /,' II IIIII IIIII')
      READ(EIN,102) PI,NA,NI
102 FORMAT(I2,X,I5,X,I5)
C****          ****
C****          PRUEFEN DER PARAMETER ****
C****          ****
      IF (PI .EQ. -1)      GOTO 90000
      IF (PI .LT. 0)      GOTO 100
      IF (PI .GT. 1)      GOTO 100
      IF (NA .LT. 1)      GOTO 100
      IF (NA .GT. 32760) GOTO 100
      IF (NI .LT. 1)      GOTO 100
      IF (NI .GT. 32760) GOTO 100
C****          ****
C****          PROTOKOLLIEREN DER PARAMETER ****
C****          ****
      WRITE(AUS,103) PN,PI,NA,NI
      WRITE(PRINT,103) PN,PI,NA,NI
103 FORMAT(' PN=',I2,' PI=',I2,' NA=',I5,' NI=',I5,
1      /,' =====')
      IF(PI .EQ. 0) GOTO 160
C****          ****
C****          EINLESEN DER TASK-PARAMETER ****
C****          ****
      WRITE(AUS,111) PN
111 FORMAT(' PN=',I2,' TASK-PARAMETER:',
1      /,' PNR = PROZESSNUMMER (1 ... 99)',
2      /,'      (0 = KEIN START)',
3      /,' CYCL= START-ZYKLUS (1 ... 999)',
4      /,' EINH= ZEIT-EINHEIT (1 ... 9)',
5      /,' =====')
      DO 150 IT = 1,TANZ
      TAKT = IT
120 WRITE(AUS,121) PN,TAKT
121 FORMAT(' PN=',I2,' EINGABE FUER TASK',I1,':',
1      /,' PNR CYCL EINH',
2      /,' II III I')
      READ(EIN,122)PNRX(TAKT),CYCLX(TAKT),EINHx(TAKT)
122 FORMAT(1X,I2,2X,I3,4X,I1)
      WRITE(AUS,123)
      WRITE(PRINT,123)
C*****

```

```

C*****
123 FORMAT(' =====')
    IF(PNRX(TAKT) .EQ. 0)      GOTO 130
    IF(PNRX(TAKT) .LT. 1)      GOTO 120
    IF(PNRX(TAKT) .GT. 99)      GOTO 120
    IF(CYCLX(TAKT) .LT. 1)      GOTO 120
    IF(CYCLX(TAKT) .GT. 999)    GOTO 120
    IF(EINHX(TAKT) .LT. 1)      GOTO 120
    IF(EINHX(TAKT) .GT. 9)      GOTO 120
    WRITE(AUS,124)PN,TAKT,PNRX(TAKT),CYCLX(TAKT),EINHX(TAKT)
    WRITE(PRINT,124)PN,TAKT,PNRX(TAKT),CYCLX(TAKT),EINHX(TAKT)
124 FORMAT(' PN=',I2,' TASK',I1,'-PARAMETER:',
1      /,' PNR=',I2,' CYCL=',I3,' EINH=',I1,
2      /,' =====')
    GOTO 140
130 WRITE(AUS,131)PN,TAKT
    WRITE(PRINT,131)PN,TAKT
131 FORMAT(' PN=',I1,' TASK',I1,' : KEIN START',
1      /,' =====')
140 CONTINUE
150 CONTINUE
    PNR1 = PNRX(1)
    PNR2 = PNRX(2)
    PNR3 = PNRX(3)
    PNR4 = PNRX(4)
    CYCL1 = CYCLX(1)
    CYCL2 = CYCLX(2)
    CYCL3 = CYCLX(3)
    CYCL4 = CYCLX(4)
    EINH1 = EINHX(1)
    EINH2 = EINHX(2)
    EINH3 = EINHX(3)
    EINH4 = EINHX(4)
160 CONTINUE
C****
C****          ZAEHLER FUER TASK-STARTS LOESCHEN
C****
    NT1(1) = 0
    NT1(2) = 0
    NT2(1) = 0
    NT2(2) = 0
    NT3(1) = 0
    NT3(2) = 0
    NT4(1) = 0
    NT4(2) = 0
C****
C****          START FUNKTIONS-TEST
C****
    WRITE(AUS,190) PN
    WRITE(PRINT,190) PN
190 FORMAT(' PN=',I2,' *** START ***')
C****
C****          ABFRAGE DER STARTZEIT
C****
    CALL TIME (TA)
    TD = SECNDS(0.0)
C****
C*****

```



```

C*****
C**** ABSCHNITT 2:  == T E S T F U N K T I O N      B Z W.      ==      ****
C****              == T E S T H I L F S , F U N K T I O N ==      ****
C*****
C****
C**** TESTHILFSFUNKTION
C**** =====
C****
C**** BESCHREIBUNG: ALS TESTHILFSFUNKTION WIRD EINE FESTLIEGENDE
C**** FOLGE VON ARITHMETISCHEN AUSDRUECKEN BENUTZT,
C**** MIT DEM DIE FREIE ZENTRALPROZESSOR-ZEIT
C**** AUFGENOMMEN WIRD (SIEHE TESTVERFAHREN).
C**** DIESE TESTHILFSFUNKTION IST IN DIESEM HAUPT-
C**** PROGRAMM (PN=7) EINGEFUEGT. DAMIT WIRD DAS
C**** STARTEN EINES WEITEREN HINTERGRUNDPROGRAMMES
C**** FUER DIESEN ZWECK UEBERFLUESSIG.
C****
C**** TESTFUNKTION
C**** =====
C****
C**** BESCHREIBUNG: DIE TESTFUNKTION BESTEHT IM BEAUFTRAGEN DES
C**** BETRIEBSSYSTEMS (CYCLE) ZUM ZYKLISCHEN START
C**** VON TASKS IN KONSTANTEN, VORGEGBAREN
C**** ZEITINTERVALLEN.
C****
C**** DAS ZEITINTERVALL MUSS SO GEWAHLT WERDEN, DASS
C**** DIE BETROFFENE TASK SICHER ABLAUFEN KANN.
C**** DIE ZU STARTENDEN TASKS LIEGEN STARTBEREIT
C**** (RESIDENT) IM HAUPTSPICHER.
C**** DIE TASKS MUESSEN EINE HOEHERE PRIORITAET ALS
C**** DAS HINTERGRUND-PROGRAMM (PN = 7) HABEN.
C**** ZUR KONTROLLE DER ANZAHL DER TASK-STARTS
C**** ERHOEHET JEDE TASK SEINEN ZAEHLER IM
C**** 'COMMON'-BEREICH.
C****
C**** VORBEREITUNG:
C**** =====
C****
C**** IM VORSPANN WERDEN DIE PARAMETER FUER DEN ZYKLISCHEN START
C**** EINGELESEN. IST DER PARAMETER PNR = 0, SO WIRD DER START
C**** DIESER TASK NICHT AUSGEFUEHRT.
C****
C**** DAS LADEN UND PRIORISIEREN DER TASKS MUSS VOR DEM BEGINN DES
C**** TESTES EXTERN (UEBER 'LADE-KOMMANDO') VORGENOMMEN SEIN.
C**** JE NACH TESTABSICHT KOENNEN DIE PRIORITAETEN ZWISCHEN DEN
C**** TASKS VARIERT WERDEN.
C****
C**** NACHBEREITUNG:
C**** =====
C****
C**** AM ENDE VON ABSCHNITT 2 WERDEN DIE AUFTRAEGE AN DAS BETRIEBS-
C**** SYSTEM ZUM ZYKLISCHEN START DER TASKS ABGEBROCHEN (CANCL).
C**** IM NACHSPANN WERDEN ZUSAETZLICH DIE ZAEHLER-STAEENDE AUS DEM
C**** COMMON PROTOKOLLIERT.
C****
C*****
C****
C**** DIESES TESTPROGRAMMN IST AUSGELEGT FUER 4 TASKS.
C****
C*****

```

```

C*****
C****
    IR(1) = 1
    IR(2) = 1
    IR(3) = 1
C****
C****          LEERSCHLEIFE BEI PI = 0
C****
    IF (PI .EQ. 1) GOTO 300
    DO 200 IA=1,NA
    DO 200 II=1,NI
    DO 200 IX=1,3
    IR(IX) = IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)
1          /IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)
200 CONTINUE
    GOTO 900
C****
C****          FUNKTIONSTEST BEI PI = 1
C****
300 CONTINUE
C****
C****          AUFTRAG ZUM ZYKLISCHEN START VON TASK1
C****
    IF (PNR1 .EQ. 0) GOTO 320
    CALL CYCLE (PNR1,CYCL1,EINH1,ST1)
    IF (ST1 .NE. 1) PNR1 = 0
    WRITE(AUS,311) PN,PNR1,ST1
    WRITE(PRINT,311) PN,PNR1,ST1
311 FORMAT(' PN=',I2,' CYCLE(TASK',I2,'): ST=',I6)
C****
C****          AUFTRAG ZUM ZYKLISCHEN START VON TASK2
C****
320 CONTINUE
    IF (PNR2 .EQ. 0) GOTO 330
    CALL CYCLE (PNR2,CYCL2,EINH2,ST2)
    IF (ST2 .NE. 1) PNR2 = 0
    WRITE(AUS,311) PN,PNR2,ST2
    WRITE(PRINT,311) PN,PNR2,ST2
C****
C****          AUFTRAG ZUM ZYKLISCHEN START VON TASK3
C****
330 CONTINUE
    IF (PNR3 .EQ. 0) GOTO 340
    CALL CYCLE (PNR3,CYCL3,EINH3,ST3)
    IF (ST3 .NE. 1) PNR3 = 0
    WRITE(AUS,311) PN,PNR3,ST3
    WRITE(PRINT,311) PN,PNR3,ST3
C****
C****          AUFTRAG ZUM ZYKLISCHEN START VON TASK4
C****
C*****

```

```

C*****
340 CONTINUE
  IF (PNR4 .EQ. 0) GOTO 350
  CALL CYCLE (PNR4,CYCL4,EINH1,ST4)
  IF (ST4 .NE. 1) PNR4 = 0
  WRITE(AUS,311) PN,PNR4,ST4
  WRITE(PRINT,311) PN,PNR4,ST4
350 CONTINUE
C****
C****          TESTHILFSFUNKTION
C****
  IR(1) = 1
  IR(2) = 1
  IR(3) = 1
  DO 600 IA = 1,NA
  DO 600 II = 1,NI
  DO 600 IX = 1,3
  IR(IX) = IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)
1      /IR(IX)/IR(IX)/IR(IX)/IR(IX)/IR(IX)
600 CONTINUE
C****
C****          BEENDEN DER ZYKLISCHEN STARTS DER TASK1
C****
  IF (PNR1 .EQ. 0) GOTO 620
  CALL CANCL (PNR1,ST1)
C****
C****          BEENDEN DER ZYKLISCHEN STARTS DER TASK2
C****
620 CONTINUE
  IF (PNR2 .EQ. 0) GOTO 630
  CALL CANCL (PNR2,ST2)
C****
C****          BEENDEN DER ZYKLISCHEN STARTS DER TASK3
C****
630 CONTINUE
  IF (PNR3 .EQ. 0) GOTO 640
  CALL CANCL (PNR3,ST3)
C****
C****          BEENDEN DER ZYKLISCHEN STARTS DER TASK4
C****
640 CONTINUE
  IF (PNR4 .EQ. 0) GOTO 900
  CALL CANCL (PNR4,ST4)
C****
900 CONTINUE
C****
C*****

```