

บทที่ 6
การติดต่อสื่อสาร
(COMMUNICATIONS)

บนข้อบิบายคำสั่งต่าง ๆ ของ GWBASIC ที่ต้องการเพื่อที่จะ support RS232 asynchronous communication กับคอมพิวเตอร์และอุปกรณ์รอบนอก (peripherals) ต่าง ๆ

การเปิดคอมมิวนิเคชันไฟล์ (Opening a Communications File)

OPEN "COM..." จัดสรรบัฟเฟอร์สำหรับ I/O ในรูปแบบเดียวกันกับที่ OPEN จัดสรรสำหรับคิสเกตต์ไฟล์ต่าง ๆ

ข้อมูลเข้า/ข้อมูลออกของการติดต่อสื่อสาร (Communication I/O)

เนื่องจาก port ของการติดต่อสื่อสารถูกเปิดในฐานะเป็นไฟล์ คำสั่งข้อมูลเข้า/ข้อมูลออกทุก ๆ คำสั่งที่ใช้ได้สำหรับคิสเกตต์ไฟล์ต่าง ๆ จะใช้ได้กับ COM ด้วย

คำสั่งข้อมูลเข้าต่าง ๆ แบบเรียงลำดับของ COM (COM sequential input statements)

เป็นเช่นเดียวกับกับคำสั่งข้อมูลเข้าแบบเรียงลำดับสำหรับคิสเกตต์ไฟล์ต่าง ๆ คำสั่งเหล่านั้นคือ :

INPUT #<file number>

LINE INPUT #<file number>

INPUT\$

คำสั่งข้อมูลออกต่าง ๆ แบบเรียงลำดับของ COM (COM sequential output statements)

เป็นเช่นเดียวกับกับคำสั่งข้อมูลออกแบบเรียงลำดับสำหรับคิสเกตต์ไฟล์ต่าง ๆ คำสั่งเหล่านั้นคือ :

PRINT #

PRINT # USING

WRITE #

อ้างอิงไปยังหัวข้อต่าง ๆ ของ INPUT และ PRINT สำหรับรายละเอียดต่าง ๆ ของรูปแบบการลงรหัส (coding format) และการใช้

ฟังก์ชันต่าง ๆ ของ I/O ของ COM (COM I/O Functions)

ลักษณะที่ช้าที่สุดของ asynchronous communication คือความสามารถที่จะปฏิบัติกับอักขระต่าง ๆ อย่างเร็วที่สุดที่จะเร็วได้ตามที่อักขระต่าง ๆ ถูกรับมา ณ อัตรา (rates) ต่าง ๆ เนื้อ 2400 ไบต์ต่อวินาที (bps) จึงมีความจำเป็นที่จะต้องส่งการเคลื่อนย้ายอักขระจาก host นานพอเพียงที่จะ "catch up" กระบวนการเหล่านี้สามารถถูกกระทำโดยทำการส่ง XOFF (Ctrl-S) ไปยัง host และทำการส่ง XON (Ctrl-Q) เมื่อพร้อมที่จะดำเนินการต่อไป ฟังก์ชันต่าง ๆ 3 ฟังก์ชันที่จะนำมาเพื่อที่จะช่วยเหลือในการตัดสินใจเมื่อเงื่อนไข "over-run" เกิดขึ้น :

LOC(x)

ส่งจำนวนของอักขระต่าง ๆ ในบัฟเฟอร์ข้อมูลเข้า (input buffer) ที่กำลังคอยเพื่อที่จะถูกอ่าน ถ้าจำนวนนั้นมีค่ามากกว่า 255 แล้ว LOC จะส่งค่าเป็น 255

LOF(x)

ส่งจำนวนของเนื้อที่ว่าง (free space) ในบัฟเฟอร์ข้อมูลเข้า ฟังก์ชันนี้จะเหมือนกับ $n - \text{LOC}(x)$ เมื่อ n เป็นขนาดของบัฟเฟอร์ของการติดต่อสื่อสารตามที่กำหนดโดย /C: option บนคำสั่ง GWBASIC ค่าที่เป็น default สำหรับ n คือ 256

EOF(x)

ส่งค่าจริง (-1) ถ้าบัฟเฟอร์ข้อมูลเข้าว่างเปล่า (empty) และส่งค่าเท็จ (0) ถ้าอักขระใด ๆ กำลังคอยที่จะถูกอ่าน

"Communications buffer overflow" error นี้ อาจเกิดขึ้นถ้าการอ่าน (read) ได้ถูกพยายามที่จะกระทำหลังจากที่บัฟเฟอร์ข้อมูลเข้าเต็ม (full) แล้ว (นั่นคือ loc(x) ส่งค่า 0)

ฟังก์ชัน INPUT\$ สำหรับไฟล์ต่าง ๆ ของ COM (INPUT\$ Function for COM Files)

ฟังก์ชัน INPUT ควรจะนำมาใช้มากกว่าคำสั่ง INPUT และคำสั่ง LINE INPUT เมื่อทำการอ่าน

ไฟล์ต่าง ๆ ของ COM เนื่องจากอักขระแอสกี (ASCII characters) ทุก ๆ อักขระอาจจะมีนัยสำคัญในการส่งต่อข้อมูล INPUT ถูกต้องการน้อยที่สุดเนื่องจากข้อมูลเข้าจะหยุดลงเมื่อเครื่องหมายจุลภาคหรือ carriage return ถูกพบ LINE INPUT สิ้นสุดลงเมื่อ carriage return ถูกพบ

INPUT\$ อนุญาตให้อักขระทุก ๆ อักขระที่ถูกอ่านถูกกำหนดค่าไปยังสตริง INPUT\$(n,y) และจะส่งค่าอักขระต่าง ๆ n ตัวจากไฟล์ y คำสั่งต่าง ๆ ต่อไปนี้เป็นสิ่งเพียงพอสำหรับการอ่านไฟล์

ของ COM :

```
10 WHILE NOT EOF(1)
20 A$=INPUT$(LOC(1),#1)
```

.
.
.
.

(process data returned in A\$)

.
.

```
60 WEND
```

คำสั่งต่าง ๆ ข้างบนนี้จะส่งจำนวนอักขระต่าง ๆ ในบัฟเฟอร์และเก็บอักขระต่าง ๆ เหล่านี้ใน A\$ ถ้ามีอักขระต่าง ๆ มากกว่า 255 อักขระ เฉพาะอักขระ 255 อักขระเท่านั้นที่จะถูกส่งไปในแต่ละครั้งเพื่อป้องกันการเกิด "String overflow" ขึ้น ยิ่งไปกว่านั้น ถ้าเกิดเป็นกรณีที่ EOF(1) เป็นเท็จแล้ว ข้อมูลเข้าดำเนินการต่อไปจนกว่าบัฟเฟอร์จะว่างเปล่า รหัสนี้ทำให้เกิดการปฏิบัติการที่รวดเร็ว (fast processing)

GET และ PUT สำหรับไฟล์ต่าง ๆ ของ COM (GET and PUT for COM Files)

GET และ PUT สำหรับไฟล์ต่าง ๆ ของ COM แตกต่างกันเล็กน้อยจาก GET และ PUT ของคัส-

ไฟล์ต่าง ๆ และจะถูกใช้สำหรับส่งหรือรับ fixed-length I/O จากไฟล์ของ COM แทนที่จะทำการระบุหมายเลขของเร็คคอร์ด โดยการระบุจำนวนไบต์ต่าง ๆ ที่จะถูกเคลื่อนย้ายไปสู่หรือเคลื่อนย้ายออกจากบัฟเฟอร์ของไฟล์ หมายเลขนี้ต้องมีค่าเกินค่าที่ถูกกำหนดโดย LEN option บนคำสั่ง OPEN "COM..." อ้างอิงไปยัง GET และ PUT ในบทที่ 7

โปรแกรมตัวอย่าง (Sample Program)

โปรแกรมต่อไปนี้จะแสดงการใช้คอมพิวเตอร์ในฐานะเป็น conventional terminal นอกจากการติดต่อสื่อสารแบบ full duplex communications กับ host แล้วโปรแกรมอนุญาตให้ข้อมูล "down-loaded" ไปยังไฟล์ได้ และในทางกลับกัน ไฟล์อาจจะถูก "up-loaded" (transmitted) ไปยังเทอร์มินัลอื่นเช่นกัน

ระบบของ asynchronous communication โปรแกรมนี้จะเป็นประโยชน์ในการเคลื่อนย้ายโปรแกรมต่าง ๆ ของ GWBASIC และข้อมูลจาก PC ไปสู่ host และจาก host มาสู่ PC

ข้อสังเกต : โปรแกรมนี้ถูกสร้างขึ้น (set up) เพื่อที่จะติดต่อกับ DEC-20 โดยการใช้ XON และ XOFF ท่านอาจจะปรับปรุงแก้ไขโปรแกรมให้เหมาะสมตามความต้องการได้

โปรแกรม TTY

```

10 SCREEN 0,0:WIDTH 80
15 KEY OFF:CLS:CLOSE
20 DEFINT A-Z
25 LOCATE 25,1
30 PRINT STRING$(60," ")
40 FALSE=0:TRUE= NOT FALSE
50 MENU=5 ' Value of MENU Key zctrl-E)
60 XOFF$=CHR$(19):XON$=CHR$(17)
100 LOCATE 25,1:PRINT "Async TTY Program"

```

```
110 LOCATE 1,1:LINE INPUT "Speed? ";SPEED$
120 COMFIL$="COM1:"+SPEED$+",E,7,1,cs,ds,cd,lf"
130 OPEN COMFIL$ AS #1
140 OPEN "SCRN:" FOR OUTPUT AS #2

200 LOCATE 25,1: PRINT "ASYNC TTY PROGRAM";SPC(5);:COLOR 0,7:
    PRINT "TERMINAL EMULATION MODE";:COLOR 7,0:PRINT SPC(5);
    "TYPE CTRL-E FOR MENU":LOCATE 2,1

205 PAUSE=FALSE

210 A$=INKEY$: IF A$="" THEN 230
220 IF ASC(A$)=MENU THEN 300 ELSE PRINT #1,A$;
230 IF EOF(1) THEN 210
240 IF LOC(1)>128 THEN PAUSE=TRUE: PRINT #1,XOFF$;
250 A$=INPUT$(LOC(1),#1)
260 PRINT #2,A$;:IF LOC(1)>0 THEN 240
270 IF PAUSE THEN PAUSE=FALSE:PRINT #1MXON$;
280 GOTO 210

300 LOCATE 25,1: PRINT "ASYNC TTY PROGRAM";SPC(5);:COLOR 0,7:
    PRINT "FILE TRANSFER MODE";:COLOR 7,0:PRINT STRING$(30," "):
    LOCATE 2,1

305 LOCATE 1,1:PRINT STRING$(30," "):LOCATE 1,1

310 LINE INPUT "File? ";DSKFIL$
```

```
400 LOCATE 1,1:PRINT STRING$(30," "):LOCATE 1,1
410 LINE INPUT"(T)ransmit or (R)eceive? ";TXRX$
420 IF TXRX$="T" THEN OPEN DSKFIL$ FOR INPUT AS #3:GOTO 1000
430 OPEN DSKFIL$ FOR OUTPUT AS #3
440 PRINT #1,CHR$(13);

500 IF EOF(1) THEN THEN GOSUB 600
510 IF LOC(1)>128 THEN PAUSE=TRUE: PRINT #1,XOFF$;
520 A$=INPUT$(LOC(1),#1)
530 PRINT #3,A$;:IF LOC(1)>0 THEN 510
540 IF PAUSE THEN PAUSE=FALSE: PRINT #1,XON$;
550 GOTO 500

600 FOR I=1 TO 5000
610 IF NOT EOF(1) THEN I=9999
620 NEXT I
630 IF I>9999 THEN RETURN
640 CLOSE #3:CLS:LOCATE 1,1:PRINT "* Download complete *"
650 GOTO 200

1000 WHILE NOT EOF(3)
1010 A$=INPUT$(1,#3)
1020 PRINT #1,A$;
1030 WEND

1040 PRINT #1,CHR$(26); 'CTRL-Z to make close file.
```

1050 CLOSE #3:CLS: LOCATE 1,1:PRINT "** Upload complete **"

1060 GOTO 200

9999 CLOSE:KEY OFF

รายละเอียดต่าง ๆ บนโปรแกรม TTY

<u>ไลน์ัมเบอร์</u>	<u>คำอธิบาย</u>
10	กำหนดจอภาพไปที่ black and white alpha mode และกำหนดความกว้างเป็น 80
15	ปิดการพิมพ์ข้อพท์คีย์ เคลียร์จอภาพและทำให้แน่ใจว่าไฟล์ทุก ๆ ไฟล์ถูกปิดลง หมายเหตุ : Asynchronous บังบอก character I/O ในลักษณะตรงกันข้ามกับ line I/O หรือ block I/O ดังนั้นคำสั่ง PRINT ทุก ๆ คำสั่ง (ไปยังคอมมูนีเคชัน ไฟล์หรือ ไม่ก็ ไปยังจอภาพ) จะถูกทำให้สิ้นสุดลงด้วยเครื่องหมายเซมิโคลอน สิ่งนี้ทำให้ carriage return และ line feed โดยปกติถูกปฏิบัติที่จุดจบของคำสั่ง PRINT ข้างลง
20	บังบอกตัวแปรจำนวนเลขทุก ๆ ตัวเป็นจำนวนเต็ม สิ่งนี้เป็นความเหมาะสมแรกเริ่มกับสับวูทที่ไลน์ 600 ถึงไลน์ 620 เพื่อที่จะทำให้ความเร็วสูงสุด โดยปกติจะใช้ตัวนับที่เป็นจำนวนเต็ม (integer counters) ต่าง ๆ ในรูปแบบต่าง ๆ เท่าที่จะทำได้
25-30	เคลียร์ไลน์ที่ 25 เริ่มต้นที่สคีมที่ 1
40	บังบอกบูลีน TRLJE และ FALSE
50	บังบอกมูลค่าแอสกี (ASC value) ของเมนูคีย์
60	บังบอก ASCII XON, XOFF characters
100-200	พิมพ์ชื่อโปรแกรม ถามหา baud rate (speed) เปิดคอมมูนีเคชันไฟล์ไปยังไฟล์หมายเลข 1, even parity, 7 data bit

แบบฝึกหัดสำหรับโปรแกรมเมอร์ : ปรับปรุงแก้ไขส่วนนี้ให้เหมาะสมเพื่อที่จะ
ตรวจสอบ buad rates ที่ใช้ได้ก่อนที่จะดำเนินการต่อไป

205-280 ส่วนนี้กระทำ full duplex I/O ระหว่างจอภาพและอุปกรณ์ที่ถูกเชื่อมต่อกับ
RS232 connector ดังต่อไปนี้ :

1. อ่านอักขระจากแป้นคีย์ไปยัง A\$ สังเกตว่า INKEY\$ ส่งค่าสริงที่ว่างเปล่า (null string) ถ้าไม่มีอักขระรอคอยอยู่
2. ถ้าไม่มีอักขระรอคอยอยู่แล้ว ตรวจสอบว่าอักขระต่าง ๆ ตัวใดที่จะถูกรับ
3. ถ้าอักขระที่แป้นคีย์เป็นเมนูคีย์แล้ว ไฟล์สามารถถูก downloaded อ่านชื่อไฟล์
4. ถ้าอักขระ (A\$) ไม่ใช่เมนูคีย์แล้วส่งมันไปโดยทำการบันทึกไปยังคอมมูนิตีชันไฟล์ (PRINT #1,A\$)
5. ณ โล่ 230 ตรวจสอบว่ามีอักขระต่าง ๆ รอคอยอยู่ในบัฟเฟอร์หรือไม่ ถ้าไม่มีอักขระใด ๆ รอคอยอยู่ ให้หันกลับไปแป้นคีย์และตรวจสอบแป้นคีย์
6. ณ โล่ 240 ถ้ามีอักขระมากกว่า 128 อักขระรอคอยอยู่แล้ว กำหนดของสัญญาณกำหนดยุค (pause flag) เพื่อที่จะหยุด (suspend) ข้อมูลเข้าชั่วคราว ส่ง XOFF เพื่อที่จะหยุดการเคลื่อนย้ายที่ทางไกลออกไปมากกว่านั้น (further transmission)
7. ณ โล่ 250 ถึงโล่ 260 อ่านและพิมพ์เนื้อหา (contents) ต่าง ๆ ของบัฟเฟอร์บนจอภาพจนกว่าบัฟเฟอร์จะว่างเปล่า ดำเนินการต่อไปเพื่อที่จะดูกระทำ (monitor) ขนาดของบัฟเฟอร์ (ในโล่ 240) หยุดการส่งถ่าย (transmission) ชั่วคราวถ้า falling behind
8. ทำให้การส่งถ่ายกลับคืนมาโดยการส่ง XON ก็ต่อเมื่อการส่งถ่ายถูกทำให้หยุดโดย XOFF ก่อนหน้า กระทำซ้ำกระบวนการจนกว่าเมนูคีย์จะถูกกด

300-310

อ่านชื่อของไฟล์จากคิสเกตต์เพื่อที่จะ download เปิดไฟล์หมายเลข 2

- 400-430 ถามชื่อไฟล์ที่จะถูกส่งถ่าย (uploaded) หรือถูกรับ (download)
- 440 ส่ง carriage return ไปยัง host เพื่อที่จะเริ่มต้นการทำ download
โปรแกรมนี้กำหนดว่าคำสั่งล่าสุดที่ถูกส่งไปยัง host จะเริ่มทำการเคลื่อนย้าย
และถูกทำให้สูญเสียบ้างเพียง terminating carriage return เท่านั้น
ถ้า DEC system เป็น host แล้วคำสั่งอาจเป็น :
- COPY TTY:=MANUAL.MEM<MENU Key>
- เมื่อเมนูนี้ถูกกดแทน RETURN
- 500 เมื่อไม่มีอักขระต่าง ๆ เพิ่มเติมที่จะถูกรับ (LOC(x) จะส่งค่า 0) กระทำที่
time-out (ถูกอธิบายในไลน์ 600)
- 510 ถ้าอักขระต่าง ๆ มากกว่า 128 อักขระรอคอยอยู่ ส่งสัญญาณการหยุด (pause)
และส่ง XOFF ไปยัง host
- 520-530 อ่านอักขระทุก ๆ อักขระในบัฟเฟอร์ (LOC(x)) และบันทึกลงสู่คิสต์
(PRINT #3..) จนกว่าจะถูก caught up
- 540-550 ถ้าการหยุด (pause) ถูกปฏิบัติ เริ่มต้น host ใหม่โดยการส่ง XON ไปและ
เคลียร์ธงสัญญาณการหยุด (pause flag) ดำเนินกระบวนการต่อไปจนกระทั่ง
ไม่มีอักขระถูกรับ ณ เวลาที่ถูกกำหนดไว้แล้วล่วงหน้า
- 600-650 สิ้นที่ time-out ตัวนี้ของลูป FOR ถูกกำหนดโดยภาพทดลอง นั่นคือ ถ้า
ไม่มีอักขระถูกรับจาก host ในเวลา 17 ถึง 20 วินาทีแล้ว การส่งถ่ายจะถูก
ว่าเสร็จสิ้นสมบูรณ์ ถ้าอักขระใด ๆ ถูกรับในระหว่างเวลานี้ (ไลน์ 610)
กำหนดค่า I ให้อยู่เหนือช่วงของลูป FOR เพื่อที่จะออกจากลูปและหันกลับไปยัง
ตัวเรียก (caller) ถ้าการส่งถ่ายของ host เสร็จสิ้นสมบูรณ์แล้ว ปิด
คิสต์เกตต์ ไฟล์และหันกลับ ไปอยู่ที่เทอามินัล
- 1000-1060 รoutines การส่งถ่าย (transmit routine) อ่านอักขระ 1 อักขระไปยัง A\$
ด้วยคำสั่ง INPUT\$ ส่งอักขระ ไปยังอุปกรณ์ในไลน์ 1020 ส่ง Ctrl Z

ในไลน์ 1040 ในการต่ออุปกรณ์ที่จะทำการรับต้องก้ามันเพื่อที่จะปิดไฟล์ของมัน
ท้ายที่สุด ในไลน์ 1050 และไลน์ 1060 ปิดคสเกตต์ไฟล์พิมพ์ข้อความที่เสร็จ-
สิ้นสมบูรณ์และหันกลับไปยัง conversation mode ในไลน์ 200

9999 ในเวลานี้ไม่ได้ถูกปฏิบัติ ไลน์เปิดคอมมูนิตี้นไฟล์ที่ถูกเปิดไว้และทำให้ soft
key display กลับคืนสู่สภาพเดิม

แบบฝึกหัดของโปรแกรมเมอร์ : เพิ่มไลน์ต่าง ๆ บางไลน์ไปยังรูทีนในไลน์

400 ถึงไลน์ 420 เพื่อที่จะออกจากโปรแกรมผ่านไลน์ 9999
