SYLVA
CONTROL SYSTEMS

IO-LCD - See Picture

The LCD connects 1 x 16 to 4 x 40 LCD displays to any RS485 network. The LCD also has 2 relay outputs that respond to received commands and 3 push button inputs that transmit ASCII codes out on the RS485. The received command set also includes cursor positioning and display functions.

Specifications

Input/Output

- three active low push button inputs
- two FORM A amp relay contacts
- 1 x 16 to 4 x 40 LCD displays via 16 header

Power

- 5V or 12 VDC operation

Communication

- RS485 at 9600 baud (half duplex)

Physical

- 2.0" x 3.0" x 1"

Features

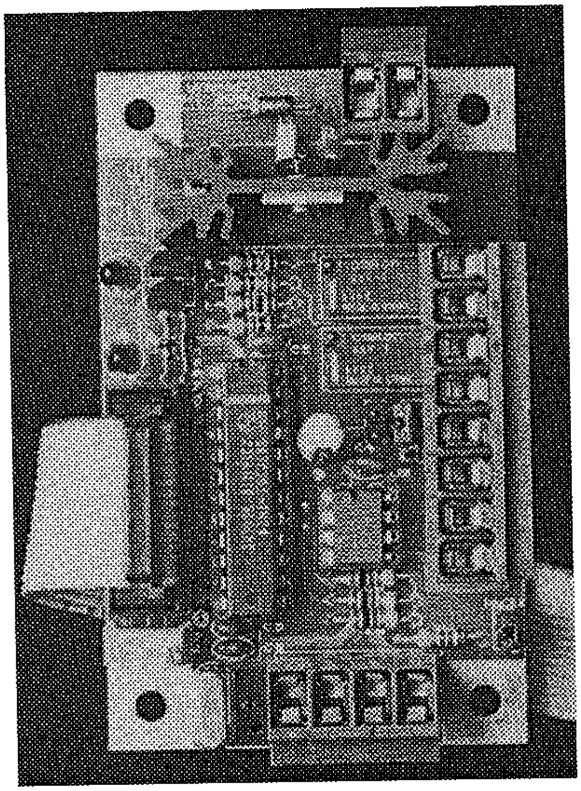
- Fully socketed circuit board
- Plug-on connector system
- LED indicators for relay status

[CATALOG] [ORDERING INFO] [PRICES&COSTS] [CONTACT US]

[TECH SUPPORT] [WHAT'S NEW]

LastEdited 08/10/97 by William H Mogk of ***WHM Software***

IO-LCD



Back to IO-LCD

SCextender

The SCEXTEND consists of a 82B715 chip, expansion bus connector, 3 pole cable connector and 3 pull-up resistor options. There are 2 factory installed pull-ups for each of the extended I2C lines and there is provision for a user installed pull-up. The formula for calculating the pull-up for each separate bus is as follows:

R= 1u sec/ (Cdevice+Cwiring)

Where: Cdevice= sum of device capacitances connected to the bus and Cwiring is the total wiring and stray capacitances on each bus.

The I2C buffered (extended) bus cannot exceed the maximum pull-up current of 30mA.
This is calculated by the formula:

30mA > (Vcc-0.4)/Rp

Where: Rp=scaled parallel combination of all pull-up resistors in the I2C system.

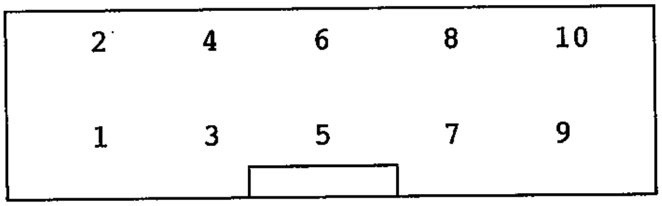
All SC552 controllers have 10K ohm pull-ups for the I2C bus. This 10K ohm resistance is scaled by multiplying by 0.1, resulting in 100K ohms. This value is considered to be negligible when in parallel with 330 ohms. When calculating Rp for a extended bus system, the parallel combination of all SCEXTENDers **cannot be less than approximately 160 ohms.**

The SCEXTEND is supplied with 330 ohm and 470 ohm pull-ups. If the user is connecting 2 SCEXTENDs then 330 in parallel with 330 yields 165 ohms. The 470 ohm resistors are for connecting 3 SCEXTENDs together. For any more SCEXTENDs the user can install their own pull-ups. It is recommended that for long cables that pull-ups be used at both ends.

Be sure to match arrows marking pin 1 on the expansion bus connectors. Improper connection may result in damage to the driver chip on the SCEXTEND.

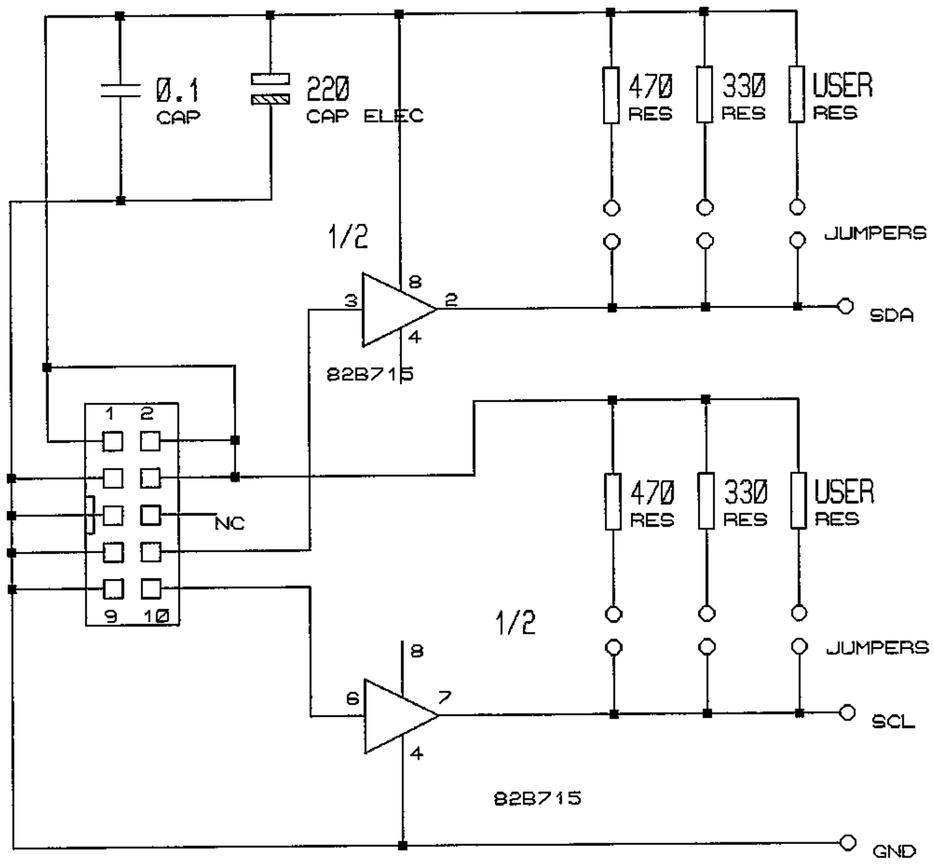
Users must satisfy themselves with the connection limits when using the extended bus. We have achieved a 1000 foot connection with 470 ohm bus pull-ups at each end using BELDEN 9451 shielded cable in a low noise environment. Users should select a low capacitance shielded cable for their applications.

EXPANSION IO CONNECTOR PIN-OUT (IIC BUS)



TOP VIEW

```
1,2,4 --- +5Vdc supply
3,5,7,9 - Ground
6 ------- RESET (active high from SC552)
8 ------- SDA (IIC serial data line 10K pull-up at SC552)
10 ------ SCL (IIC serial clock line 10K pull-up at SC552)
```



SYLVA CONTROL SYSTEMS

TITLE:
SCEXTEND
I2C BUS DRIVER

BY: LG

REV:1.0

DATE:
22-Oct-97

PAGE:
1/1

SYLVA
CONTROL SYSTEMS

AE-ADC - See Picture

The ACD analog expansion board provides 8 single analog inputs with 12 bit resolution. The ADC board can communicate in either local I2C expansion or RS 485 serial modes. the RS485 feature makes the ADC ideal for remote analog monitoring.

Specifications

Input

- 8 single ended 12 bit inputs
- Input range - 0-2.5V, 0-5V, 0-10V, 4-20mA.

Power

- 5V via I2C connection
- 8-18 VDC for for remote operation

Communication

- Philips I2C bus
- RS485 half duplex (4800 or 9600 baud)

Physical

- 4.5" x 4.5" x 1"

Features

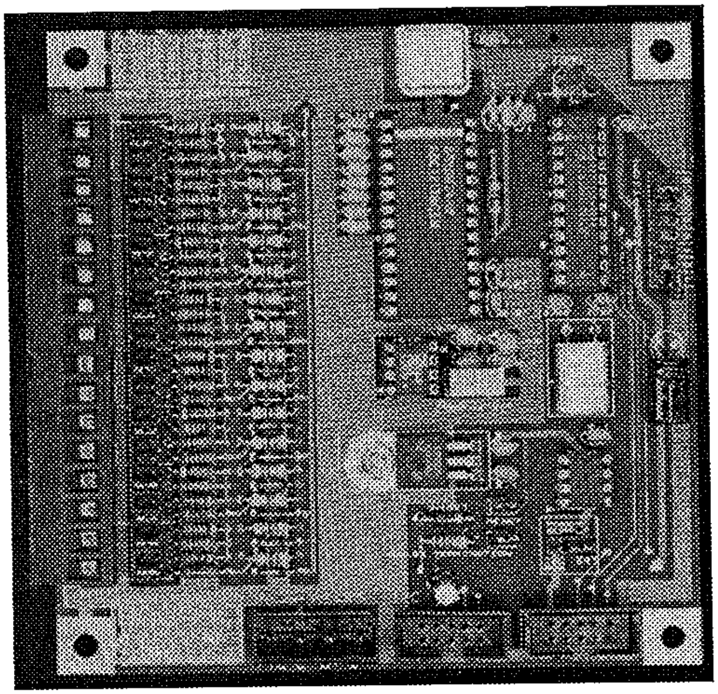
- Fully socketed circuit board
- Plug-on connector system
- Daily chain I2C bus connection
- Separately configured input ranges

[CATALOG] [ORDERING INFO] [PRICES&COSTS] [CONTACT US]

[TECH SUPPORT] [WHAT'S NEW]

LastEdited 08/10/97 by William H Mogk of ***WHM Software***

AE-ADC



[Return to AE-ADC](#)

READ VOLTAGE & CURRENT

GOSUB ___

GET VALUES FOR VOLTAGE

$$V = \text{SQRT}\left(\frac{\text{VRDG} * \text{VRDG} * \#\text{SMPLS}}{166464 \frac{\text{SMPLS}}{100}}\right) * \left(\frac{8}{1.4142}\right) \quad V$$

$$I = \frac{\text{IRDG} * \text{IRDG}}{\quad} * \frac{4}{1.4142} \quad A$$

$$\text{PRES} = V * I = \underline{\qquad} \text{ W.}$$

READ POWER

PRDG

$$32147 = 397.62$$

$$P = \frac{397.62}{32147} * 16074$$

$$= 198.48\text{W}$$

---

from Sylva A/D Rdg = AV

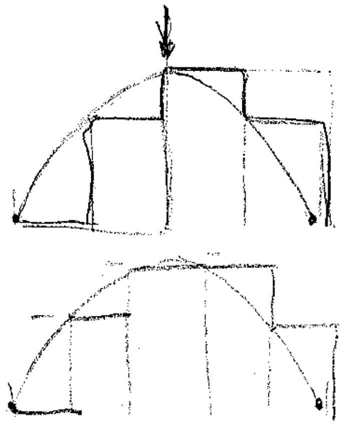
= .636 Vpk

from Zyn A/D Rdg = RMS

= .707 pk

$$\text{Sylva rdg} * \frac{.707}{.636} = \text{Zyn rdg}$$

1 * 2T



— 40V MAX —5V

— 20A MAX — 5V

VOLTAGE RDG

$$\mathrm{SQRT}\left(\frac{\mathrm{VRDG} * \mathrm{VRDG} * 128}{166464}\right) * \left(\frac{8}{1.4142}\right)$$

½ scale

$$\left(\frac{128 * 128 * 128}{166464}\right)^{1/2} \frac{* 8}{1.4142}$$

```
10  REM *amain03.BAS*

20  CLEAR R

30  COMINT #,62000: CIC #,1

100 string 1201,11
102 $(1)="PASS ":$(2)="FAIL ":$(3)="volts ":$(4)="amps ":$(5)="watts "
104 $(6)="Press 1 ":$(7)="Press 2 ":$(8)="to print ":$(9)="pass tests "
106 $(10)="fail tests ":$(11)="to run ":$(12)="manual "
108 $(13)="automatic ":$(14)="to go to ":$(15)="next test "
110 $(16)="redo test ":$(17)="min limit ":$(18)="max limit "
112 $(19)="test value "

300 gosub 63000
399 rem


400 rem decide what to print
410 gosub 63200
420 if k=1 then allp=1
430 if k=2 then falp=1
440 if allp=0.and.falp=0 then goto 420
450 print "allp= ",allp,"  ","falp= ",falp
460 rem xih 503: goto 460
470 k=0

500 rem decide manual or automatic
510 gosub 63300
520 if aut=0.and.man=0 then goto 520
550 print "aut= ",aut,"  ","man= ",man
560 rem xih 503: goto 560
570 k=0

600 rem display test "0" result page
610 gosub 63400
699 rem

700 if aut=1 then ~~man=0:~~ goto 800
710 if man=1 then ~~aut=0:~~ goto 1800
799 rem

800 rem sequence tests automatically
810 print "got to 810"
815 fwd=0
820 if man=1 then goto 1800
830 if fwd=0 then goto ~~830~~ ~~815~~ 820
840 for i=0to7
850 k=i
860 gosub 1900
865 rem
870 next i
880 goto 815
899 rem

1000 stop

1800 rem wait for test number input from keypad
1801 print "got to 1800"
1810 rem
```

```
1820 if aut=1 then goto 800
1830 rem
1840 if k>0.or.k<32 then gosub 1900
1850 goto 1810
1899 rem

1900 rem respond to keypad or automatic input
1901 if k=1 then gosub 10100:gosub 63400
1902 if k=2 then gosub 10200:gosub 63400
1903 if k=3 then gosub 10300:gosub 63400
1904 if k=4 then gosub 10400:gosub 63400
1905 if k=5 then gosub 10500:gosub 63400
1906 if k=6 then gosub 10600:gosub 63400
1907 if k=7 then gosub 10700:gosub 63400
1908 if k=8 then gosub 10800:gosub 63400
1909 if k=9 then gosub 10900:gosub 63400
1910 if k=10 then gosub 11000:gosub 63400
1911 if k=11 then gosub 11100:gosub 63400
1912 if k=12 then gosub 11200:gosub 63400
1913 if k=13 then gosub 11300:gosub 63400
1914 if k=14 then gosub 11400:gosub 63400
1915 if k=15 then gosub 11500:gosub 63400
1916 if k=16 then gosub 11600:gosub 63400
1917 if k=17 then gosub 11700:gosub 63400
1918 if k=18 then gosub 11800:gosub 63400
1919 if k=19 then gosub 11900:gosub 63400
1920 if k=20 then gosub 12000:gosub 63400
1921 if k=21 then gosub 12100:gosub 63400
1922 if k=22 then gosub 12200:gosub 63400
1923 if k=23 then gosub 12300:gosub 63400
1924 if k=24 then gosub 12400:gosub 63400
1925 if k=25 then gosub 12500:gosub 63400
1926 if k=26 then gosub 12600:gosub 63400
1927 if k=27 then gosub 12700:gosub 63400
1928 if k=28 then gosub 12800:gosub 63400
1929 if k=29 then gosub 12900:gosub 63400
1930 if k=30 then gosub 13000:gosub 63400
1931 if k=31 then gosub 13100:gosub 63400

1990 return
1999 rem goto 1900


10000 rem test routines

10100 rem * tst01.bas *
10110 tnum =01
10120 rem no loads
10130 otl 009: lio: rem select +5v supply
10132 for t=0to100:next t:rem let input settle
10140 rem no z4k setup
10150 trdg=adc5
10160 rem no z4k reset
10170 tval=trdg*(5/4)*(5/1024)
10180 limax=5.30:limin=4.70
10182 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10190 gosub 64100
10199 clear r:k=0:return

10200 rem * test02.bas *
```

```
10210 tnum =02
10220 rem no loads
10230 otl 008: lio: rem select P+12V supply
10232 for t=0to100:next t:rem let input settle
10240 rem no z4k setup
10250 trdg=adc5
10260 rem no z4k reset
10270 tval=trdg*(12/4)*(5/1024)
10280 limax=12.7:limin=11.3
10282 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10290 gosub 64100
10299 clear r:k=0:return

10300 rem * test03.bas *
10310 tnum =03
10320 rem no loads
10330 otl 008: otl 007: lio: rem select P+12V and P-12V supplies
10332 for t=0to100:next t:rem let input settle
10340 rem no z4k setup
10350 trdg=adc5
10360 rem no z4k reset
10370 tval=trdg*(5/1024)
10380 limax=2.15:limin=1.85
10382 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10390 gosub 64100
10399 clear r:k=0:return

10400 rem * test04.bas *
10410 tnum =04
10420 rem no loads
10430 otl 006: lio: rem select C+5V supply
10432 for t=0to100:next t:rem let input settle
10440 rem no z4k setup
10450 trdg=adc5
10460 rem no z4k reset
10470 tval=trdg*(5/4)*(5/1024)
10480 limax=5.05:limin=4.95
10482 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10490 gosub 64100
10499 clear r:k=0:return

10500 rem * test05.bas *
10510 tnum =05
10520 rem no loads
10530 otl 005: lio: rem select REF+2.5V supply
10532 for t=0to500:next t:rem let input settle
10540 rem no z4k setup
10550 trdg=adc5
10560 rem no z4k reset
10570 tval=trdg*(5/4)*(5/1024)
10580 limax=2.51:limin=2.49
10582 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10590 gosub 64100
10599 clear r:k=0:return

10600 rem * test06.bas *
10610 tnum =06
10620 rem no loads
10630 otl 004: lio: rem select RLY+12V supply
10632 for t=0to500:next t:rem let input settle
```

```
10640 rem no z4k setup
10650 trdg=adc5
10660 rem no z4k reset
10670 tval=trdg*(12/4)*(5/1024)
10680 limax=12.3:limin=11.7
10682 if tval>limax.or.tval<limin then $(40)=$(2)else $(40)=$(1)
10690 gosub 64100
10699 clear r:k=0:return


10700 print "10700":clear r:k=0:return
10800 print "10800":clear r:k=0:return
10900 print "10900":clear r:k=0:return
11000 print "11000":clear r:k=0:return
11100 print "11100":clear r:k=0:return
11200 print "11200":clear r:k=0:return
11300 print "11300":clear r:k=0:return
11400 print "11400":clear r:k=0:return
11500 print "11500":clear r:k=0:return
11600 print "11600":clear r:k=0:return
11700 print "11700":clear r:k=0:return
11800 print "11800":clear r:k=0:return
11900 print "11900":clear r:k=0:return
12000 print "12000":clear r:k=0:return
12100 print "12100":clear r:k=0:return
12200 print "12200":clear r:k=0:return
12300 print "12300":clear r:k=0:return
12400 print "12400":clear r:k=0:return
12500 print "12500":clear r:k=0:return
12600 print "12600":clear r:k=0:return
12700 print "12700":clear r:k=0:return
12800 print "12800":clear r:k=0:return
12900 print "12900":clear r:k=0:return
13000 print "13000":clear r:k=0:return
13100 print "13100":clear r:k=0:return



62000 cic#,0: a=xby(0d00h)
62010 if a<175.or.a>198 then cic#,1: reti

62020 a=a-128: rem strip bit 7

62030 rem look for an "F"
62035 if a=70 then fwd=1:print "fwd= ",fwd:cic#,1:reti

62040 rem look for a "D"
62045 if a=68 then dog=1:print "dog= ",dog:cic#,1:reti

62050 rem look for a "C"
62055 if a=67 then cat=1:print "cat= ",cat:cic#,1:reti

62060 rem look for a "B"
62065 if a=66 then man=1:aut=0:print "62065 man= ",man:cic#,1:reti

62070 rem look for a "A"
62075 if a=65 then aut=1:man=0:print "62075 aut= ",aut:cic#,1:reti


62100 rem handle numbers
```

```
62135 rem if get just "E" the force output to "0"
62140 if a=69.and.z=0 then xby(02e0h)=48:z=z+1

62145 rem when get an "E" go output the number
62150 if a=69 then xby(02e0h+z)=13: goto 62185
62155 rem add numbers pressed
62160 if a<58.and.a>47 then xby(02e0h+z)=a:z=z+1:print chr(a)
62165 cic#,1: reti

62185 call 35f0h: pop k:z=0
62190 rem print:print "The integer value input is ",k
62195 cic#,1:reti

63000 rem display at startup

63010 PRINT#CHR(27),CHR(3)
63020 print# chr(27),chr(5):rem backlight off
63030 for t=0to500:next t
63040 print# chr(27),chr(4):rem backlight on

63100 PRINT# CHR(27),CHR(3);: FOR T=1 TO 50: NEXT T
63105 PRINT#CHR(27),CHR(1),CHR(10);
63110 PRINT#"MTH ELECTRIC TRAINS";
63115 PRINT#CHR(27),CHR(1),CHR(50);
63120 PRINT#"Z-4000 TEST FIXTURE";
63125 PRINT#CHR(27),CHR(2),CHR(14);
63130 PRINT#"Revision 0.2";
63135 PRINT#CHR(27),CHR(2),CHR(42);: DATE #;
63140 PRINT#CHR(27),CHR(2),CHR(59);:PRINT# k;
63145 PRINT#CHR(27),CHR(2),CHR(70);: TIME #
63150 for t=0to2000:next t
63155 RETURN

63200 PRINT# CHR(27),CHR(3);: FOR T=1 TO 50: NEXT T
63205 PRINT#CHR(27),CHR(1),CHR(7);
63210 PRINT#"Press 1 to print all tests";
63215 PRINT#CHR(27),CHR(1),CHR(47);
63220 PRINT#"Press 2 to print fail tests";
63225 PRINT#CHR(27),chr(2),chr(7);
63230 print#"______________________________";
63235 PRINT#CHR(27),CHR(2),CHR(42);: DATE #;
63240 PRINT#CHR(27),CHR(2),CHR(59);:PRINT# k;
63245 PRINT#CHR(27),CHR(2),CHR(70);: TIME #
63250 for t=0to500:next t
63255 RETURN

63300 PRINT# CHR(27),CHR(3);: FOR T=1 TO 50: NEXT T
63305 PRINT#CHR(27),CHR(1),CHR(7);
63310 PRINT#"Press B to run manual";
63315 PRINT#CHR(27),CHR(1),CHR(47);
63320 PRINT#"Press A to run automatic";
63325 PRINT#CHR(27),chr(2),chr(7);
63330 print#"______________________________";
63335 PRINT#CHR(27),CHR(2),CHR(42);: DATE #;
63340 PRINT#CHR(27),CHR(2),CHR(59);:PRINT# k;
63345 PRINT#CHR(27),CHR(2),CHR(70);: TIME #
63350 for t=0to500:next t
63355 RETURN
```

```
63400 PRINT# CHR(27),CHR(3);: FOR T=1 TO 50: NEXT T

63405 PRINT#CHR(27),CHR(1),CHR(10);
63410 PRINT#"Test";:print# tnum;:print#"   ";:print#$(40);

63415 PRINT#CHR(27),CHR(1),CHR(40);
63420 PRINT#"Min limit = ";:print# limin;:print#"  ";
63422 print#"Max limit = ";:print# limax;

63425 PRINT#CHR(27),chr(2),chr(7);
63430 print#"Test value = ";:print# f(##.###),tval;:Print#"  ";:print# $(3)

63435 PRINT#CHR(27),CHR(2),CHR(42);: DATE #;
63440 PRINT#CHR(27),CHR(2),CHR(59);:PRINT# k;
63445 PRINT#CHR(27),CHR(2),CHR(70);: TIME #
63450 for t=0to500:next t
63455 RETURN

64000 rem print test results to screen
64100 print "test ",tnum,:print"  ",:print $(40)
64110 print "Min limit= ",limin,"   ","Max limit= ",limax
64120 print "test value= ",:print f(##.###),tval
64130 xih 503:goto 64130
64199 return
```